matrox
electronic systems ltd.
5800 ANDOVER AVE., T.M.R., QUE., H4T 1H4, CANADA
TEL.: 514—735-1182 TELEX: 05-825651

MTX-ALPHA
SOFTWARE

ALT-2480 SOFTWARE

User's Manual

ALT-2480 SOFTWARE PACKAGE

INTRODUCTION

The ALT-2480 Software Package provides the user the full flexibility of a software driven video display with the implementation ease of a stand-alone terminal. The Software Package has been designed explicitly to support easy and reliable modification to meet varying user requirements. Wherever possible, parameters and definitions are not tested until run time to permit maximum flexibility without requiring user written code modifications. Although the input routines are set up to run using 'skip' I/O, the display routines (OUTCHR and ECHOCH) are explicitly written to be useable at interrupt level.

As supplied, the package will fully emulate the popular Lear Siegler, inc. ADM-3A and Digital Equipment Corp. DECSCOPE VT-52 interactive display terminals. In addition, line at a time and text block input modes are available to provide the powerful text preparation features of an intelligent terminal.

USER'S GUIDE

This section explains the keyboard functions available under the ALT-2480 Software Package. All key codes are interpreted by software, so the ASCII code(s) associated with any function(s) can be changed as desired (see Software Interfacing Guide). The input key codes for ATTN, XON, XOFF, and block mode ESC may also be changed dynamically under keyboard or program control.

Input Modes

There are three basic input modes that can be used depending on the degree of input processing desired. The least sophisticated mode is the full duplex (FDX) mode. In this mode no processing is performed on input. For a typed in character to appear on the display, it must be echoed by the user program. (If the ALT-2480 Software Package is being used as part of a system monitor, that monitor is considered the user program.) Characters are passed on to the user program as soon as they are input, exactly as they are input. The only exceptions are the input control codes SETC (^B), ATTN (^C), XOFF (^S), and XON (^Q) used to set configuration switches, return to monitor level, stop output, and resume output respectively.

Half duplex mode buffers characters as they are input until a full line is typed. A full line consists of either 80 characters or 0 through 79 characters followed by a CR, ESC, or LF. All characters are echoed as they are input. Carriage return echoes as CR-LF and both CR and LF are passed to the user program. Rubout will delete the last character

in the buffer (and on the screen) while ΛU will cancel the entire line. Once a full line has been entered, no further input will be accepted until the entire line has been read by the user program and the first character on the next line requested. Control characters other than SETC, ATTN, XOFF, XON, RO, Line Cancel, CR, LF, and HT are echoed as Λ<char> and have no other effect on the display. They will be passed to the user program when requested exactly as typed, not as Λ<char>. *

In this release, RO and ΛU may not update the display correctly if tabs are erased or the input line exceeds one display line. Regardless of what appears on the display, RO and ΛU always have the correct effect on the input line buffer.

The third input mode is block mode. In this mode, the user can generate an entire block of data using all the editing capabilities of the intelligent terminal system. By inputing the XMIT (End of Text) code (ΛD), all data on the screen entered since the last XMIT code is sent to the user program. This can be particularly effective in such applications as filling in the entries on a computer generated form. When in block mode, no control characters are passed on to the user program except the implied carriage returns at the end of each line of data, horizontal tabs to indicate a field of protected data, and the EOT to mark the end of the transmission.

* EX.: When Typed
A^H B CR

^H = Backspace
CR = Carriage Return
LF = Line Feed

When Echoed

B CR LF

Keyboard Commands

Except as noted in Appendix I, all commands can also be executed by the user program through calls to OUTCHR.

The notation ^<char> is used to indicate the ASCII code generated by holding down the control key while the <char> key is depressed. Some control characters such as ESC (^[) may require using both the control and shift keys. Many keyboards include separate keys for some of the frequently typed control codes. For example, virtually every keyboard has a CR (or Return) key, which generates the same code as control M. Appendix I is a list of all the commands, their assigned control characters, and equivalent letter codes. In the definitions which follow, only the letter code is given to avoid confusion.

Cursor Controls. The following commands move the cursor about the screen. To retain compatibility with the LSI ADM-3A, vertical tab and form feed require preceding ESC characters. All cursor controls are nondestructive (i.e., they do not affect any of the data on the display).

Backspace (^H). Each time a backspace is executed, the cursor moves one position to the left. Cursor action when the cursor is already in the leftmost column is determined by OFFLFT.

Horizontal Tab (^I). Each time a horizontal tab is executed, the cursor moves right to the next tab stop. Tab stops are set at every eighth column. Cursor action when the next tab stop is beyond the right end of the line is determined by OFFRT.

Linefeed (^J). Each time a line feed is executed the cursor moves down to the same position on the next line. If the cursor is already on the bottom line, either the cursor will wrap around to the top line or the entire display will scroll up one line (losing the contents of the top line) as determined by OFFBOT.

Vertical Tab (^[, ^K). Each time a vertical tab is executed the cursor moves down to the next vertical tab stop. These stops are set every eight lines. If the next tab stop is off the bottom of the display, cursor action is determined by OFFBOT. This is a two character command because the VT character is used for the upline command.

Upline (^K). Each time an upline is executed the cursor moves to the same character position in the line immediately above the current one. If the cursor is already on the top line, display action is determined by OFFTOP.

Forespace (^L). Each time a forespace is executed the cursor moves to the next character position. If the cursor is on the last position on a line, the next character position is determined by the OFFRT switch.

Return (^M). This code moves the cursor to the first character position of the present line. When input from the keyboard in half duplex or block mode, a line feed is automatically appended and executed.

Home (^^). The cursor is moved to the upper left display position; line 1, column 1.

Load Cursor (^[, '=', <Y>, <X>). The next two characters following the

^[, '=' sequence represent the absolute line and column (Y and X) coordinates which are used to position the cursor. The upper left cursor position is line 1, column 1. The characters required are calculated by adding 31 (decimal) to the desired line (or column) number. The Home Command is equivalent to the Load Cursor sequence ^[, '=', SP, SP.

Editing Commands

The following commands are used to manipulate data on the screen. They may be output by the user program at any time. However, they are executable from the keyboard only when in block input mode. The half duplex input mode editing command Rubout is described in the Input Mode section.

Form Feed (^[, ^L). The form feed command sequence clears the screen and moves the cursor to the first position on the top line. This is a two character command because the FF character is used for the Forespace Command.

Clear Screen (^Z). This deletes all data on the screen. The cursor position is not changed.

Line Insert (^W). The line containing the cursor and all following lines move down one line. The bottom line on the screen is lost.

Line Delete (^U). The line containing the cursor is deleted. All lines below the cursor are moved up one line and a blank line is moved into the bottom line. In half duplex modes, the entire line buffer is deleted.

Char Insert (∧V). The character indicated by the cursor and succeeding characters on the same line are shifted right one character. The cursor position is set to a blank. This function will not operate if the last position on the line contains data.

Char Delete (∧X). The character indicated by the cursor is removed. Characters to the right of the cursor on the same line are moved one position to the left.

Insert Mode (∧[, 'I'). This command simplifies insertion of long strings of data. Logically precedes each succeeding character with an insert character command. The Insert Mode is terminated by any control character, which is otherwise ignored.

EX : ∧[I FFFF = ∧F∧F∧F∧F

Transmit Block (∧D). Transmit all screen data from the last transmit command (from line 1, column 1 if not on screen) up to the current cursor position, to the user program. Trailing blanks on each line (unless explicitly entered by the user) are ignored. Individual lines are separated by carriage return line feed sequences. End of Text character (∧D) is appended to the end of the transmission to signify end of text block. Protected fields are replaced by Horizontal Tabs (∧I). No other control characters are transmitted. *

Set XMIT Start (∧[, ∧D). Changes the cursor position associated with the last Transmit Block command to the current cursor position. This allows the user to select a command from a menu or repeatedly input the same string (as long as it stays on the screen).

*EX:AAAA■■■■ BBBB is transmitted as AAAA∧I∧I∧I∧I BBBB
■ Background display = protected field ∧I is only one character long.

Special Commands

Four special command codes are implemented to maximize system utility. The first one, SETC, is used to change the terminal configuration switches. The other three are normally system monitor functions and can be deleted if the monitor (if any) in use provides the same function.

These switches are always detected and acted upon while the software is in use. Any keyboard input other than these commands while output is being processed is ignored.

Set Configuration (^B, <CMND>, <PARM>). The SETC command allows the user or program complete control over the terminal configuration. Each configuration switch change requires a complete three character sequence. The ^B causes the following two characters to be interpreted as the switch to change and the desired value. The switch must be an upper or lower case letter, the value can be any character other than a control character. When executed from the keyboard, the user will be prompted on the top line. Any response other than 'Y' to the question mark will cause the request to be aborted. The 'Y' should not be included when setting configuration switches from the user program.

The key codes for different setting values are given in Appendix II. Switches which can be set are:

<A> Use the ALT-2480 display at the address indicated.

<B> Set OFFBOT to determine whether to wrap around to the top line or scroll the screen up when the cursor is moved below the bottom line.

<C> Select cursor character. The character input becomes the new cursor character.

<D> Display lower case as lower case (normal).

<E> Select escape character for block mode keyboard input.

<F> Select XOFF character.

<G> Display lower case characters using the greek symbol set.

<H> Display lower case as upper case. (This switch should always be used with the 2480-C option.)

<I> Select ATTN character.

<J> Set display line length. If length is forty or less, the display generated will be compatible with the ALT-2480 low resolution option.

<K> Reserved.

<L> Set OFFLFT to determine whether to back up to the previous line, wrap around on the same line, or remain in the first position on the line when the cursor is moved past the left edge of the display.

<M> Select input mode.

<N> Select XON character.

<O>, <P>, <Q> Reserved

<R> Set OFFRT to determine whether to start a new line, wrap around on the same line, or remain in the last position when the cursor is moved past the last position on a line.

<S> Set or clear "TTY lock." The TTY lock shifts all lower case characters to upper case on input. It does not affect program output.

<T> Set OFFTOP to determine whether to wrap around to the bottom line or scroll the screen down when the cursor is moved above the top line.

<U> through <Z> Unused.

Attention (ΛC). This command returns control to a user specified address. Normally this would be the monitor restart or breakpoint trap address.

Stop Output (ΛS). This command stops all output processing until a Resume output command is given. This allows the user to stop the program long enough to read the output and then resume processing. Only special commands may be entered while this command is in effect.

Resume Output (ΛQ). This nullifies the output freeze caused by a stop output command.

Additional Commands

Six additional commands are provided for additional flexibility.

Auto Answer Back (ΛE). In response to the ENQ command from the program, the software will respond with a short HERE IS message. This can be convenient for identifying specific versions of the 2480 software which have been specially modified for a given application.

Bell Subroutine (ΛG). Since the ALT-2480 does not provide an acoustic warning tone, a special routine is provided. This routine can either be modified to ring a user provided bell interface or left as is to flash the screen once.

Keyboard Lock and Unlock (^O and ^N). These command codes disable and enable keyboard input respectively. An attempt to input from the keyboard while it is locked will trap to the monitor entry point specified by MONLVI.

Select Foreground Display (^[, ^_). Display all following characters in normal video.

Select Background Display (^[, ^Y). Display all following characters in inverse video (and/or blink as strapped in hardware). Note that fields in background mode are not input in block mode but are replaced by horizontal tabs.

Software Interfacing Guide

There are only three primary entry points in the MATROX 2480 Software Package. There is one routine call to output to the display, one to read the next available keyboard input, and one to see if any keyboard input is available. The same three routine calls are used regardless of the input mode in use or the style of output desired. A fourth entry point is also provided to allow independent, noninterfering output. Local storage for this routine is totally independent of that used for program output facilitating adaptation of the package to interrupt driven keyboard input.

All four routines obey the following register conventions: 1) All registers except the PSW are preserved. 2) Values are returned in register A with the flags set to match. 3) Output routines expect the argument to be in register C. In accordance with convention 2, this argument is returned in register A as well.

Primary Entry Points

OUTCHR

The character in register C is displayed at the current cursor position and the cursor is advanced to the next character position. Characters with numerical values less than 32 (blank) are assumed to be control characters (parity is ignored). The action taken for any particular control character is determined by the lookup table at the address in CONAT. If a control character is not in the referenced table, it is displayed as ^<char>. Lower case characters may be optionally shifted to their upper case or greek equivalents. The parity bit will be set or cleared to match the current display mode (background or foreground respectively).

INCHRW

The next available input character is returned in register A. If no input data is available (e.g. a line terminator has not yet been typed in half duplex input mode), the cursor character is flashed at the current cursor location to prompt the user. Only one character is returned with each call to this routine. However, once an input line or block is terminated, there is no delay in subsequent calls as long as buffered data is available.

There is no requirement that all available data be input before processing any output, but be careful with the block mode, as any input data shifted off screen before being input will be irrecoverably lost. Half duplex line storage is limited to 80 bytes. If this limit is reached before a line terminator is entered, the entire line buffer will be released to the user program and no further keyboard data entry will be accepted until the entire buffer has been read by the user program. Care must also be exercised when changing input modes

to avoid undesired loss of buffered input data.

When using block mode input, keep in mind that no distinction is made between displayed program output and displayed keyboard input. This distinction can be maintained by using foreground mode for keyboard input and background mode for program output.

TSTIN

This routine allows the user program to check if any data is available for input. If calling INCHRW would result in a delay (i.e. a character, line, or block is not available), this routine returns with register A set to zero. If data is currently available, register A is set to FF(hex). Flags are set to match the contents of register A.

ECHOCH

This routine is similar to OUTCHR but is modified to simplify keyboard echo, specially in interrupt driven systems. Its functioning is identical to OUTCHR with the following exceptions.

1) Output is independent of the XOFF command.
2) The control table referenced by ECONAT is used.
3) Parity is not ignored. Characters with the parity bit zero are treated the same as in OUTCHR. However, all characters with parity bit set are considered control characters and searched for in the control table.
4) Escape and other multiple character sequences are maintained independent of any in progress in OUTCHR.

Required User Supplied Subroutines

To interface with the user supplied keyboard, this package requires two user defined routines. These routines may use any registers desired, the only requirement is that they return their value in register A.

INKBS

This routine must return the status of the user keyboard. Register A should be zero if a character is not available. Any other value implies a character is available immediately by calling INKBD. For compatibility with potential user programs, there should not be any response time requirement between a positive response to INKBS and the subsequent call to INKBD. INKBS is called at address STFDX + 3.

INKBD

This routine should return in register A the ASCII character input by the user. It is called only after a positive response to INKBS is received. (Note that more than one positive response to INKBS may be required before a call to INKBD depending on the user program.) The parity bit may be set or clear as desired. It is ignored by the package but is provided to the user in full duplex and half duplex input modes. INKBD is called at address INFDD + 1.

Lookup Tables and Variables

Most of the power and flexibility of this package are due to the extensive use of run time interpretation of critical parameters and control character definitions along with strict segregation of program code (ROMable) and program data (RAM only). By appropriate use of the SETC command, a single copy of this package can independently control multiple MTX -2480 displays.

The use and allowable values of all variables are documented in the source listing. Some of the more powerful or unusual ones are:

CPTRS

To maintain the identity of specific points on the display as characters and lines are added or deleted and as scrolling occurs, these character pointers are updated by all routines which move data about the screen. Each pointer requires two bytes. The low byte is the column and the high byte is the line. The total number of pointers is determined by the compilation switch CPNUM, currently set to three. The first two pointers are used in block input mode to keep track of which character to transmit next and when to stop transmitting. The third pointer is available for other uses.

CURSAT

This pointer is the current cursor position. It is tested before displaying any data to verify that it is on screen. Action taken when off screen is determined by the variables OFFBOT, OFFLFT, OFFRT, and OFFTOP which are interpreted by the routine TSTCUR. Note that TSTCUR modifies only the cursor data supplied in registers H,L and if necessary, the display. It does not modify the contents of CURSAT.

MTXAT

This word contains the base address to use in all references to the MATROX display memory. It can also be used to provide a left margin by increasing the address by the desired value and decreasing the line width accordingly.

BLKEND

This byte defines both the terminate block (XMIT) character and the second character of the set block start command.

MONLVL

This defines the address to call if the ATTN character is detected on input. It is also called if an attempt is made to input a character while the keyboard is locked. Both conditions are ignored if the address is zero.

ECONAT

This word defines the control character lookup table used by the routine ECHOCH. It must contain the address of a valid control character lookup table. CONAT performs the same function for OUTCHR. In this package ECONAT and CONAT are the same. If wanted a new lookup table can be created for routine ECHOCH.

INTRAP

This word is tested before each attempt to get a character from the keyboard. If it is not zero, the address contained is jumped to. A RET instruction will return the value in register A as if it had been input from the keyboard. A JMP to INFDK will proceed with normal acquisition of keyboard input. Useful to control input data, and output data or commands from the user program when in block of half-duplex mode.

MULJMP

This word is tested by OUTCHR after registers B and C have been set up but before any processing is begun. If not zero, the contents are considered the address of a routine and called. If output has been inhibited by an XOFF command, it will not be tested until output is permitted to resume. The routine called should return with the CY

flag clear if output processing of the contents of register C is desired. CY flag set squelches further processing. Only the contents of register B must be preserved. IMULJM performs the same function for ECHOCH. The same use as INTRAP but in output controlling.

CONTAB

This is an ECHOCH and OUTCHR control character definition table. The table is built of three byte entries consisting of the value of the control character and the address of the routine to execute it. By convention, a character with the parity bit set is equivalent to the same character preceded by the escape character. If a match is found, the associated routine is called with register B positive if from OUTCHR, negative if from ECHOCH (guaranteed not to change sign if incremented less than 100 times). Register C contains the character matched and registers H and L contain the line and column of the current cursor position respectively. The routine called may use any registers desired, including register B.

Table entries may be for any eight bit value. However, the table is only searched for characters from 0 through 31 and 128 through 255. The entries may be in any order with the exception of the null control character. The last entry in the table must be zero in order to terminate the search. The table is linearly searched and only the first occurrence of a character is detected. This is utilized to redefine the carriage return in block input mode without duplicating the entire table.

APPENDIX I

Control Codes

Code		ASCII	Function	Input FDX	Input HDX	Input BLK	Output
^@	00	NUL					
^A	01	SOH					
^B	02	STX	SETC: Set Configuration Switches	x	x	x	x
^C	03	ETX	ATTN: Return to Monitor	x	x	x	
^D	04	EOT	Transmit Block			x	
^E	05	ENQ	Auto Answer Back				x
^F	06	ACK					
^G	07	BEL	Bell Subroutine			x	x
^H	08	BS	Backspace			x	x
^I	09	HT	Horizontal Tab		x	x	x
^J	0A	LF	Line Feed		x	x	x
^K	0B	VT	Upline			x	x
^L	0C	FF	Forespace			x	x
^M	0D	CR	Carriage Return		x	x	x
^N	0E	SO	Unlock Keyboard			x	x
^O	0F	SI	Lock Keyboard			x	x
^P	10	DLE					
^Q	11	DC1	XON - Resume Output	x	x	x	
^R	12	DC2	Reserved				
^S	13	DC3	XOFF - Stop Output	x	x	x	
^T	14	DC4					
^U	15	NAK	Delete Line		x	x	x
^V	16	SYN	Insert Character			x	x
^W	17	ETB	Insert Line			x	x
^X	18	CAN	Delete Character			x	x
^Y	19	EM	Reserved				
^Z	1A	SUB	Clear Screen			x	x
^	1B	ESC	Escape Char for Multi Char Commands			x	x
^	1C	FS					
^	1D	GS					
^^	1E	RS	Home			x	x
^_	1F	US					
Rubout	7F	DEL	Delete Character		x		

Escape Character Sequences

Code	Function	Input FDX	Input HDX	Input BLK	Output
ESC '=' <X> <Y>	Direct Cursor Addressing			x	x
ESC 'I'	Insert Mode			x	x
ESC FF	Form Feed			x	x
ESC VT	Vertical Tab			x	x
ESC EOT	Set Start of XMIT Block			x	
ESC US	Select Foreground Display			x	x
ESC EM	Select Background Display			x	x

APPENDIX II

Configuration Switches

^B, <CMND>, <PARM>

<CMND>	Function	<PARM>	Set to
A	Set 2480 Base Address	0	0000H
		1	1000H
		:	:
		9	9000H
		:	A000H
		;	B000H
		<	C000H
		=	D000H
		>	E000H
		?	F000H
B	Set OFFBOT Switch	1	Wrap around to top
		H	Scroll up
C	Select Cursor Char	<char>	Cursor becomes the char
D	Display Lower Case as Lower	0	
E	Select Escape Char	<char>	Escape becomes the char
F	Select XOFF Char	<char>	XOFF becomes the char
G	Display LC as Greek	0	
H	Display Lower Case as Upper	0	
I	Select ATTN Char	<char>	ATTN becomes the char
J	Set Display Line Length	X	40 wide
		x	72 wide
		SP	80 wide
K	Reserved		
L	Set OFFLFT Switch	SP	Back up to previous line
		1	Overwrite first char on line
		0	Wrap around to end of line
M	Set Input Mode	0	Half duplex
		1	Full duplex
		2	Block mode
N	Select XON Char	<char>	XON becomes the char
O			
P	Reserved		
Q			
R	Set OFFRT Switch	SP	First char of next line
		1	Wrap around on same line
		0	Overwrite last char
S	Set TTY Upper Case Lock	1	On
		0	Off (normal)
T	Set OFFTOP Switch	1	Scroll down
		H	Wrap around to bottom
U through Z	Undefined		

APPENDIX III

The Demonstration Program 1

To permit evaluation of this software package, a simple demonstration program is included as part of the package. To run the demonstration, load the object paper tape using a standard Intel format hex loader. The program loads starting at address 0100 hex and requires less than 3K bytes of memory. Once loaded, manually patch the address of your INKBS routine into the JMP at location 0103 hex, the address of your INKBD routine into the JMP at location 0106 hex (see Software Interfacing Guide for the definitions of the INKBD and INKBS routines), the address of a routine to read your current console device (value returned in register A) into the JMP at location 0109 hex and the address of a routine to output the character in register C or A on your console device into the JMP at location 010C hex. The console I/O routines are not required if only the first phase of the demonstration is executed. If desired, the location MONLVL (address 0A5A hex) may be patched to the breakpoint or restart address of your monitor.

Display software parameters are initially set to the following values. They may be modified as desired using the Set Configuration Switch command.

-ALT-2480 addressed at E000 hex.

-Line length is 40 characters (low resolution).

-Input mode is full duplex.

-Input upper case shift lock is off.

-Output displays lower case as upper case.

-Cursor character is inverse video underline.

-OFFBOT set to scroll display.

-OFFTOP set to wrap around to bottom line.

-OFFRT set to start a new line.

-OFFLFT set to overwrite the first character on the line.

-Control characters are defined to correspond with the User's Guide.

To run the demonstration, start execution at location 0100 hex. If your ALT-2480 is addressed in the memory block starting at E000 hex, a flashing cursor will appear in the upper left corner of the display. If your ALT-2480 is not addressed at E000 hex, type in the command sequence ^B, A, n, Y where n is the character 0 through 9 or :, ;, <, =, or ?. See Appendix II for the correct value to use. This command sequence will reset the software to use the ALT-2480 at the specified address.

The first phase of the demonstration is a simple loop where a character is read by INCHRW and displayed by OUTCHR. The input mode is initially set to full duplex so that characters are displayed by OUTCHR exactly as typed in. By changing to half duplex input mode, (type ^B, M, Ø, Y) it is possible to see the line at a time editing ability of the package. When a line is terminated by either CR, LF, or ESC, the entire line is redisplayed by OUTCHR. (If the line is terminated by an ESC, the first character provided by the next input from INCHRW will be processed by OUTCHR as the second character of an escape sequence, so use care). Similarly, the full editing power of the [illegible] may be [illegible] selecting block mode (type ^B, M,

2,Y). When using block mode for the very first time, clear the screen first. This will initialize the line fill table and character pointers from the "random" contents left from the loading process. When changing from half duplex or block mode, type the terminator character immediately after executing the mode change to exit the input buffer fill code and permit the mode change to take effect.

The second phase of the demonstration program is an independent test of OUTCHR. This phase is entered by typing the control character FS (^ \) during phase one. Note that this will not change test phases if input mode is block mode, nor will the phase change in half duplex mode until the line is terminated and the FS character is received by the demonstration program. Phase two accepts characters from the console and displays them using OUTCHR. The routines INCHRW and INSTS are not involved. This permits extensive evaluation of display output characteristics without interference from input restrictions. This phase is exited by typing a US (^ _) on the console.

The final phase of the test program is an independent test of the various INCHRW modes. Characters are output to the console exactly as they would have been received by a user program. Typing an RS (^^) will return the demonstration program to the initial phase.

Application Notes

This package is provided to permit the user to experiment with various system capabilities and differing applications with a minimum of programming effort. While this section discusses various applications using the MTX2480 software package, it is important to keep in mind that this software package is not production level software. Efficiency, size, and speed of execution are all deliberately sacrificed to provide a wide range of capabilities and maximum flexibility.

A sophisticated intelligent terminal can be assembled from a minimum number of parts; display, keyboard, CPU, serial port, ROM, and a little RAM. The MTX2480 software package demonstrates many of the capabilities found in the popular Hazeltine 2000 intelligent terminal. In this case, however, many terminal characteristics can be modified by simple keyboard commands. Even production models could be radically modified simply by changing the ROM program, greatly simplifying last minute specification changes or custom variations.

In mini and micro computer based systems, the display can be integrated directly into the system, eliminating the need for extra I/O ports and utilizing idle processor time and memory. This also permits highly interactive, real time control of the display, which is often not practical over typical communication lines.

When used as the operator's console, system parameters can be displayed and updated by the operating system in real time with a minimum of overhead. For example, to display a status message on a PDP-11, an eight word routine is sufficient:

```
        MOV     R1, #MESSAGE        ;Address of message text
        MOV     R2, #DISPLAY        ;Address of display area to use
        MOV     R3, #LENGTH         ;Length of message
LOOP:   MOVB    (R2)+, (R1)+        ;Transfer the message
        SOB     R3, LOOP            ;Repeat until done
```

On a Z-80, only ten bytes are required!

```
LD      BC, LENGTH          ;Message length
LD      DE, DISPLAY         ;Display area desired
LD      HL, MESSAGE         ;Message desired
LDIR                        ;Display it.
```

Integrated into a small business system, the display can significantly enhance thruput and accuracy, especially with unsophisticated users. For example, order forms can be filled in by displaying the appropriate blank form and guiding the user through the required entries one step at a time using the line at a time input mode provided by MTX2480. Entries can be checked by the applications software for validity and consistency at the time of entry, allowing immediate interactive correction.

Considering the display can do anything a CRT terminal can do, only far faster; the possibilities are limitless. Except for operations requiring mass data movement (e.g. scrolling), even the MTX2480 software package can display several thousand characters a second. In general, the primary limitation on display update speed is the time required to generate or retrieve the data. This capability to read or write any data on the display almost instantly makes practical applications not even contemplated with conventional terminals.

Demonstration Program 2

This program sets the page mode, clears the screening, sets the cursor at home, and line length in 80 characters. The INTRAP location points to the address of a routine that test the column numbers. If it is 75, the bell is outputted using the OUTCHR routine.

When a block is terminated, it is outputted on the screem without blanks.

ADDRESS	CONTENT		MNEMONIC	COMMENT
ØCØØ	Ø6	BEGIN	MOVI B,Ø8	/Load B with number of
ØCØ1	Ø8			/Codes
ØCØ2	21		LXI, H,L	/Load H,L with first
ØCØ3	ØØ			/Address of codes
ØCØ4	ØD			
ØCØ5	4E	LOOP	MOV C,M	/Get one code and
ØCØ6	CD		CALL	/Output it
ØCØ7	ØC		OUTCHR	
ØCØ8	Ø2			
ØCØ9	23		INX H	/Point to next code
ØCØA	Ø5		DCR B	/Decrement counter and
ØCØB	C2		JNZ LOOP	/Test for all done
ØCØC	Ø5			
ØCØD	ØC			
ØCØE	CD	HERE	CALL	/Get a block or
ØCØF	6B		INCHRW	/Buffer character if block
ØC1Ø	Ø2			/Terminated
ØC11	FE		CPI ' '	/If it is a blank
ØC12	2Ø			/Get next character
ØC13	CA		JZ HERE	/
ØC14	ØE			
ØC15	ØC			
ØC16	CD		CALL	/If not output it
ØC17	ØC		OUTCHR	
ØC18	Ø2			
ØC19	C3		JMP HERE	/Get next character or
ØC1A	ØE			/Block
ØC1B	ØC			

The bell call is made with the following routine. Manually patch the address of this routine (ØC4Ø) in the INTRAP location (ØAB5).

ØC4Ø	2A	LHLD	/Load H,L with
ØC41	32	CURSAT	/Cursor position
ØC42	ØA		
ØC43	7D	MOV A,L	
ØC44	FE	CPI 75D	/Test if it is
ØC45	4B		/Equal to 75
ØC46	C2	JNZ INFOK	/If not, get the
ØC47	86		/Next character
ØC48	Ø5		

ADDRESS	CONTENT	MNEMONIC	COMMENT
ØC49	ØE	MOVI CØ7H	/If yes
ØC4A	Ø7		
ØC4B	CD	CALL	/Output the
ØC4C	ØC	OUTCHR	/Bell command
ØC4D	Ø2		
ØC4E	C3	JMP INFOK	/Then get the
ØC4F	86		/Next character
ØC5Ø	Ø5		

```
02AF C9                 RET


                ;***********************************************
                ;
                ;       OUTPUT SUBROUTINES
                ;
                ;
                ;THESE ROUTINES ALL MUST PRESERVE REGISTERS B AND C
                ;
                ;***********************************************

                ;ROUTINE CONTRL (C=CHAR, B=LEVEL)
                ;       CONTROL CHARACTER PROCESSING ROUTINE.
                ;       SCANS TABLE CONTAB OR ECONTAB AS DETERMINED BY
                ;          LEVEL FOR THE CHARACTER.
                ;       IF A MATCH IS FOUND, THE INDICATED ROUTINE IS
                ;          CALLED WITH B=LEVEL, C=CHAR, H=LINE AND
                ;          L=COLUMN (OF CURRENT CURSOR POSITION).
                ;       CALLED ROUTINES MAY UTILIZE ANY REGISTERS.
                ;       CONTROL CHARACTER TABLES (ADDRESS IN CONAT OR EC
                ;          MUST CONCLUDE WITH THE NULL CHARACTER (00HEX)
                ;       REGISTERS A, D, E, FLAGS, H AND L MODIFIED.
                ;       CY FLAG IS SET IF TNE CHARACTER IS NOT FOUND.

02B0 C5         CONTRL: PUSH    B       ;SAVE VITALS
02B1 2AC70A             LHLD    CONAT   ;ASSUME OUTPUT MODE
02B4 04                 INR     B       ;  IS IT?
02B5 F2BB02             JP      CNTRO   ;    IT IS
02B8 2AB30A             LHLD    ECONAT  ;GET ECHO CONTROL TABLE
02BB 7E         CNTRO:  MOV     A,M     ;GET TABLE ENTRY
02BC 23                 INX     H       ;ON TO ADDRESS
02BD B9                 CMP     C       ;DESIRED CHARACTER?
02BE CACA02             JZ      CNTR1   ;  YES.  DO IT
02C1 23                 INX     H       ;STEP TO NEXT ENTRY
02C2 23                 INX     H
02C3 B7                 ORA     A       ;BUT CHECK FOR END OF TABLE
02C4 C2BB02             JNZ     CNTRO   ;  BEFORE CONTINUING
02C7 37                 STC             ;FLAG AS FAILURE TO FIND
02C8 C1                 POP     B       ;RESTORE VITAE
02C9 C9                 RET

                ;EXECUTE THE DESIRED CONTROL FUNCTION

02CA 5E         CNTR1:  MOV     E,M     ;LOW BYTE OF ADDRESS
02CB 23                 INX     H
02CC 56                 MOV     D,M     ;AND HI BYTE
02CD 21D602             LXI     H,CNTRB ;FAKE A CALL
02D0 E5                 PUSH    H
02D1 D5                 PUSH    D       ;CALL ADDRESS
02D2 2A320A             LHLD    CURSAT  ;CURSOR POSITION
02D5 C9                 RET             ;WOULD YOU BELIEVE 'CALL'?
02D6 C1         CNTRB:  POP     B       ;RESTORE VITALS
02D7 AF                 XRA     A       ;CLEAR CARRY
```

```
02D8 C9                 RET             ;  AND RETURN SUCCESSFULLY

                ;SUBROUIINE TSTCUR (H=LINE, L=COLUMN)
                ;       ADJUST H,L TO THE NEAREST ON SCREEN POINT
                ;       H AND L ARE TREATED AS SIGNED 8 BIT INTEGERS
                ;       ACTION TAKEN ON OFF SCREEN POINTS IS DETERMINED
                ;        BY THE SWITCHES OFFLFT, OFFRT, OFFTOP AND
                ;        OFFBOT.
                ;       IF ORIGINAL POINT IS ON SCREEN IT IS NOT MODIFIE
                ;       A, D, E, FLAGS AND HL MODIFIED.

02D9 AF         TSTCUR: XRA     A       ;TEST FOR OFF LEFT FIRST
02DA B5                 ORA     L
02DB F2EB02             JP      TST10   ;OK SO FAR, TEST RIGHT SIDE
02DE 3AC50A             LDA     OFFLFT  ;OFF THE LEFT, WHAT TO DO?
02E1 3D                 DCR     A
02E2 F2E702             JP      TST05   ;USE A, LINE # IS OK
02E5 25                 DCR     H       ;UP ONE LINE
02E6 2F                 CMA             ;  AND CORRECT COL NUMBER
02E7 6F         TST05:  MOV     L,A     ;SET NEW COLUMN
02E8 C3FD02             JMP     TST20   ;AND TEST LINE #

02EB 3ABC0A     TST10:  LDA     WIDTH   ;TEST FOR OFF RIGHT SIDE
02EE 3D                 DCR     A       ;MAX LEGAL IS WIDTH-1
02EF BD                 CMP     L       ;STILL ON?
02F0 D2FD02             JNC     TST20   ;YES.  CHECK LINE
02F3 3AC60A             LDA     OFFRT   ;OFF THE RIGHT, NOW WHAT?
02F6 3D                 DCR     A
02F7 F2FC02             JP      TST15   ;ADJUST COL ONLY
02FA 24                 INR     H       ;DOWN ONE LINE
02FB AF                 XRA     A       ;  AND 1ST COLUMN
02FC 6F         TST15:  MOV     L,A     ;SET CORRECT COLUMN

                ;COLUMN IS NOW OK.  CHECK THE LINE.

02FD AF         TST20:  XRA     A       ;TEST FOR OFF TOP
02FE B4                 ORA     H
02FF F20D03             JP      TST30   ;TOP OK, CHECK BOTTOM
0302 3AC30A             LDA     OFFTOP  ;OFF TOP.  NOW WHAT?
0305 3D                 DCR     A
0306 67                 MOV     H,A     ;NEW LINE NUMBER
0307 E5                 PUSH    H       ;SAVE CURSOR
0308 CC8F03             CZ      SCRLDN  ;SCROLL IF REQUIRED
030B E1                 POP     H       ;RETRIEVE CURSOR
030C C9                 RET             ;ALL DONE

030D FE18       TST30:  CPI     24D     ;TEST FOR OFF BOTTOM
030F D8                 RC              ;A-OK.
0310 3AC40A             LDA     OFFBOT  ;DOWN TOO FAR, SO FIX
0313 3D                 DCR     A
0314 67                 MOV     H,A     ;NEW LINE NUMBER
0315 E5                 PUSH    H       ;SAVE CURSOR
0316 C44303             CNZ     SCRLUP  ;SCROLLING AS REQUIRED
0319 E1                 POP     H
031A C9                 RET

                ;SUBROUTINE PUTUP (C=CHAR, H=LINE, L=COLUMN)
```

```
Z

                ;       MATROX 2480 SUBROUTINE PACKAGE

                ;       VERSION 2.05  <>  JAN 21, 1978

                ;             COPYRIGHT (C) 1978
                ;             DR VINCENT C JONES
                ;             25B NORTH MAGNOLIA
                ;             SATELLITE BCH, FLA
                ;                   32937


                ;
                ;       COMPILATION SWITCHES
                ;
0000                    FALSE   EQU     0
FFFF                    TRUE    EQU     NOT FALSE
0000                    SALONE  EQU     FALSE   ;STAND ALONE VERSION
FFFF                    DEMO    EQU     TRUE    ;COMPILE AS DEMONSTRATIO
0003                    CPNUM   EQU     3       ;CURSOR POINTERS
0050                    LINSIZ  EQU     80D     ;LINE BUFFER SIZE

                            IF DEMO

                        ;STAND ALONE DEMONSTRATION PROGRAM  1
0100                        ORG     100H    ;WORK UNDER CP/M
0100 C30F01                 JMP     BEGIN
0103 C312F0     INKBS:  JMP     CSTS    ;KEYBOARD STATUS
0106 C303F0     INKBD:  JMP     CI      ;KEYBOARD DATA
0109 C303F0     CILOC:  JMP     CI      ;READ CONSOLE
010C C309F0     COLOC:  JMP     CO      ;WRITE CONSOLE
010F 310002     BEGIN:  LXI     SP,STACK
                    ;SELF CONTAINED TEST
0112 CD6B02     LOOP0:  CALL    INCHRW  ;GET A CHAR
0115 4F                 MOV     C,A
0116 CD0C02             CALL    OUTCHR  ;DISPLAY IT
0119 FE1C               CPI     FS      ;TIME FOR NEXT TEST?
011B C21201             JNZ     LOOP0   ;NOT YET
                    ;OUTPUT TEST
011E CD0901     LOOP:   CALL    CILOC
0121 4F                 MOV     C,A
0122 CD0C02             CALL    OUTCHR
0125 FE1F               CPI     US      ;SHIFT TIME?
0127 C21E01             JNZ     LOOP
                    ;INPUT TEST
012A CD6B02     LOOP2:  CALL    INCHRW
012D F5                 PUSH    PSW
012E 4F                 MOV     C,A
012F CD0C01             CALL    COLOC
0132 F1                 POP     PSW
0133 FE1E               CPI     RS
0135 C22A01             JNZ     LOOP2
0138 C31E01             JMP     LOOP

0200                    ORG     200H

F003                    CI      EQU     0F003H  ;DEFINE FOR ZAPPLE
```

```
F009                    CO      EQU     0F009H
F012                    CSTS    EQU     0F012H

0200                    STACK   EQU     $

                            ENDIF


                    ;
                    ;       ****    TOP LEVEL ROUTINES    ****
                    ;
                    ;EXCEPT AS NOTED ALL REGISTERS ARE PRESERVED.
                    ;
                    ;
                    ;ROUTINE OUTCHR (C=CHAR)
                    ;    DISPLAY THE ASCII CHARACTER IN C AT THE
                    ;CURRENT CURSOR POSITION AND ADVANCE THE CURSOR
                    ;TO THE NEXT CHARACTER POSITION.
                    ;
                    ;    CHARACTERS WITH NUMERICAL VALUES LESS THAN
                    ;32 (SPACE) ARE ASSUMED TO BE CONTROL CHARACTERS.
                    ;
                    ;
                    ;ROUTINE INCHRW
                    ;    RETURNS THE NEXT AVAILABLE INPUT CHARACTER
                    ;IN REGISTER A (FLAGS ARE SET TO MATCH).
                    ;IF NO CHARACTER IS AVAILABLE, THIS ROUTNE WILL WAIT
                    ;UNTIL ONE IS.
                    ;    THIS ROUTINE IS USED FOR ALL INPUT MODES.
                    ;IF IN A BUFFERED MODE (HALF DUPLEX OR BLOCK)
                    ;NO CHARACTERS WILL BE RETURNED UNTIL A COMPLETED
                    ;BUFFER IS AVAILABLE.  ONCE THE BUFFER IS RELEASED
                    ;BY THE KEYBOARD, EACH SUCCESSIVE CALL TO INCHRW
                    ;WILL RETURN THE NEXT CHARACTER IN THE BUFFER.
                    ;
                    ;
                    ;ROUTINE INSTS
                    ;    RETURNS THE ACCUMULATOR SET TO TRUE (FF HEX)
                    ;IF A CHARACTER IS AVAILABLE FOR INPUT FROM INCHR.
                    ;OTHERWISE, A IS CLEARED TO FALSE (00 HEX).
                    ;    FLAGS ARE SET TO MATCH.
                    ;
                    ;
                    ;ROUTINE ECHOCH (C=CHAR)
                    ;    SAME AS OUTCHR EXCEPT THAT MULTIPLE CHARACTER SEQUE
                    ;ARE MAINTAINED INDEPENDENTLY TO ALLOW NONCONFLICTING
                    ;ECHOING CONCURRENTLY WITH PROGRAM OUTPUT.
                    ;


                    ;**** INTERRUPT LEVEL (ECHO) ENTRY POINT

0200 E5             ECHOCH: PUSH    H       ;SAVE THE WORLD
0201 D5                     PUSH    D
0202 C5                     PUSH    B
0203 F5                     PUSH    PSW
0204 0680                   MVI     B,80H   ;SET INTERRUPT LEVEL FLAG
```

```
0206 2AC10A            LHLD    IMULJM  ;CHECK FOR MULTICHAR
0209 C31F02            JMP     OUTCO   ;REST OF PROCESSING IS
                                       ;  COMMON WITH OUTCHR

                ;**** NORMAL ENTRY POINT FOR PROGRAM OUTPUT

020C E5        OUTCHR: PUSH    H       ;SAVE THE WORLD
020D D5                PUSH    D
020E C5                PUSH    B
020F F5                PUSH    PSW
0210 79                MOV     A,C     ;CLEAR PARITY
0211 E67F              ANI     7FH
0213 4F                MOV     C,A
0214 0600              MVI     B,00H   ;SET NORMAL OUTPUT FLAG
0216 CD3A08    OUTCE:  CALL    XOFFED  ;OUTPUT THROTTLED?
0219 C21602            JNZ     OUTCE   ;YES.  KEEP TRYING
021C 2ABF0A            LHLD    MULJMP  ;MULTI CHARACTER SEQUENCE?
021F 7C        OUTCO:  MOV     A,H     ;CHECK IF ZERO
0220 B5                ORA     L
0221 CA2C02            JZ      OUTC2   ;NOTHING TO IT
0224 112902            LXI     D,OUTC1 ;FAKE A CALL TO IT
0227 D5                PUSH    D       ;RETURN ADDRESS
0228 E9                PCHL            ;'CALL' ROUTINE
0229 DA3902    OUTC1:  JC      OUTC9   ;CY SET MEANS ALL DONE
022C 79        OUTC2:  MOV     A,C     ;PROCESS THE CHARACTER
022D FE20              CPI     ' '     ;CONTROL CHARACTER?
022F DA3E02            JC      OUTCC   ;  YES
0232 B7                ORA     A       ;PARITY BIT SET?
0233 FA3E02            JM      OUTCC   ;  YES.  TREAT AS CONTROL
0236 CD5D02            CALL    OUTCX   ;NORMAL PRINTING CHAR,
                                       ;  DISPLAY IT
0239 F1        OUTC9:  POP     PSW     ;ALL DONE, RESTORE
023A C1                POP     B       ;  REGISTERS
023B D1                POP     D       ;  AND RETURN
023C E1                POP     H
023D C9                RET

                ; CONTROL CHARACTER PROCESSING

023E CDB002    OUTCC:  CALL    CONTRL  ;SEE IF LEGITIMATE CONTROL
0241 D23902            JNC     OUTC9   ;YES, ALL DONE
0244 79                MOV     A,C     ;UNRECOGNIZED CONTROL CHAR
0245 FEA0              CPI     ' ' OR 80H ;PRINTING CHAR?
0247 D25702            JNC     OUTC3   ;  YES, PRINT IT
024A C640              ADI     '@'     ;SHIFT TO UC ALPHA
024C 4F                MOV     C,A
024D C5                PUSH    B       ;  AND SAVE FOR LATER
024E E680              ANI     PARON   ;SAVE FOR/BACK BIT
0250 C65E              ADI     '^'     ;AND INDICATE CONTROL CHAR
0252 4F                MOV     C,A
0253 CD5D02            CALL    OUTCX   ;  BY LEADING UP-ARROW
0256 C1                POP     B       ;GET THE CHARACTER BACK
0257 CD5D02    OUTC3:  CALL    OUTCX   ;DISPLAY CHARACTER
025A C33902            JMP     OUTC9   ;  AND RETURN

                ;  INTERNAL SUBROUTINE OUTCX (C = CHAR)
                ;    ADJUST CURSOR TO LIE ON THE SCREEN.
```

```
                ;     DISPLAY THE CHARACTER AT THE ADJUSTED
                ;             CURSOR POSITION.
                ;     MOVE THE CURSOR TO THE NEXT COLUMN
                ;               (MAY BE OFF SCREEN).

025D 2A320A     OUTCX:  LHLD    CURSAT  ;GET CURRENT CURSOR
0260 CDD902             CALL    TSTCUR  ;CHECK AND ADJUST
0263 CD1B03             CALL    PUTUP   ;DISPLAY IT
0266 2C                 INR     L       ;NEXT COLUMN
0267 22320A             SHLD    CURSAT  ;SAVE NEW CURSOR
026A C9                 RET

                ;
                ;****** NORMAL ENTRY POINT FOR PROGRAM INPUT
                ;

026B E5         INCHR:  PUSH    H       ;SAVE THE WORLD
026C D5                 PUSH    D
026D C5                 PUSH    B
026E 3A540A             LDA     FDUX    ;WHAT INPUT MODE?
0271 3C                 INR     A
0272 CA8402             JZ      INCHF   ;  FULL DUPLEX.
0275 F27E02             JP      INCHH   ;  HALF DUPLEX
0278 CDA806             CALL    INBLK   ;BLOCK MODE
027B C38A02             JMP     INCHX   ; AND RETURN
027E CDF005     INCHH:  CALL    INHDX   ;GET A LINE BUFFERED CHAR
0281 C38A02             JMP     INCHX
0284 CD7D05     INCHF:  CALL    INFDX   ;GET NEXT KEYSTROKE
0287 DA8402             JC      INCHF   ;CY SET SO TRY AGAIN
028A B7         INCHX:  ORA     A       ;SET FLAGS
028B C1                 POP     B
028C D1                 POP     D
028D E1                 POP     H
028E C9                 RET

                ;
                ;****** NORMAL ENTRY POINT FOR INPUT STATUS
                ;

028F E5         INSTS:  PUSH    H       ;SAVE THE WORLD
0290 D5                 PUSH    D
0291 C5                 PUSH    B
0292 3A540A             LDA     FDUX    ;WHAT INPUT MODE?
0295 3C                 INR     A
0296 CAA802             JZ      STCHF   ;FULL DUPLEX.
0299 F2A202             JP      STCHH   ;  HALF DUPLEX
029C CDB707             CALL    STBLK   ;BLOCK MODE
029F C3AB02             JMP     STCHX   ; AND RETURN
02A2 CDB107     STCHH:  CALL    STHDX   ;GET A LINE BUFFERED CHAR
02A5 C3AB02             JMP     STCHX
02A8 CDA507     STCHF:  CALL    STFDX   ;GET NEXT KEYSTROKE
02AB B7         STCHX:  ORA     A       ;SET FLAGS
02AC C1                 POP     B
02AD D1                 POP     D
02AE E1                 POP     H
```

```
                ;       BARE MINIMUM ALT-2480 DRIVER
                ;       VERSION 1.00  <>  OCT 20,1977
                ;
                ;            COPYRIGHT (C) 1977
                ;           DR VINCENT C. JONES
                ;           11017 BENNINGTON AVE
                ;           KANSAS CITY MO 64134
                ;
                ;       BARE MINIMUM DRIVER ROUTINE FOR MATROX ALT-2480
                ;    DISPLAY.  EMULATES A SIMPLE SCROLLING TERMINAL.
                ;    THE ONLY CONTROL CHARACTERS RECOGNIZED ARE LINE
                ;    FEED AND CARRIAGE RETURN.
                ;       THE USING PROGRAM MUST DEFINE THE BASE ADDRESS
                ;    OF THE 2480 IN USE (MTXAD) AND A ONE BYTE LOCATION
                ;    IN RAM (CURSOR).
                ;       CHARACTER TO BE DISPLAYED MUST BE IN REGISTER C.
                ;    A AND FLAGS ARE MODIFIED.
                ;
                ;
0000 =                  FALSE   EQU     0
FFFF =                  TRUE    EQU     NOT FALSE
FFFF =                  DEMO    EQU     TRUE    ;SUBROUTINE OR DEMO?
FFFF =                  W40     EQU     TRUE    ;40 WIDE OR 80?
000D =                  CR      EQU     0DH     ;DEFINE CARRIAGE RET
000A =                  LF      EQU     0AH     ;  AND LINE FEED
                ;
                ;
                            IF DEMO

                ;       DEMONSTRATION DRIVER

0100                            ORG 100H

0100 310002     BEGIN:  LXI     SP,STACK
0103 DB00       KBIN:   IN      0       ;WAIT FOR INPUT
0105 E601               ANI     1
0107 CA0301             JZ      KBIN
010A DB01               IN      1       ;GET THE CHARACTER
010C E67F               ANI     7FH     ;DELETE PARITY
010E 4F                 MOV     C,A     ;MOVE INTO POSITION
010F CD1501             CALL    MTXOUT  ;AND ECHO IT
0112 C30301             JMP     KBIN    ;AND KEEP ON DOING IT

0200 =                  STACK   EQU     200H    ;DEFINE SOME STACK
0200 =                  CURSOR  EQU     200H    ;AND A BYTE OF RAM

E000 =                  MTXAD   EQU     0E000H  ;MATROX IS HERE

                ;       END OF DEMONSTRATION DRIVER

                            ENDIF
                ;
                ;       START OF ACTUAL 2480 ROUTINE
                ;
0115 79         MTXOUT: MOV     A,C     ;CHECK IF CONTROL
0116 FE0D               CPI     CR
0118 CA3E01             JZ      OUTCR   ;CARRIAGE RETURN
```

```
011B FE0A               CPI     LF
011D CA4301             JZ      OUTLF   ;LINEFEED
0120 3A0002             LDA     CURSOR  ;DISPLAY AT NEXT LOCATION
                        IF W40          ;40 WIDE
0123 FE28               CPI     40D
                        ENDIF
                        IF NOT W40
                        CPI     80D     ;80 WIDE
                        ENDIF
0125 DA2C01             JC      OUT20   ;OK AS IS
0128 CD4301             CALL    OUTLF   ;SCROLL
012B AF                 XRA     A       ;CARRIAGE RETURN
012C E5         OUT20:  PUSH    H       ;SAVE WORK REGS
012D D5                 PUSH    D
012E 6F                 MOV     L,A     ;DESIRED COLUMN
012F 3C                 INR     A       ;ADVANCE CURSOR
0130 320002             STA     CURSOR  ;  FOR NEXT TIME
0133 2600               MVI     H,0     ;CALCULATE ADDRESS
0135 1180EB             LXI     D,MTXAD+128D*23D ;1ST ON LAST LINE
                        IF W40
0138 29                 DAD     H       ;40 WIDE IS EVEN ONLY
                        ENDIF
0139 19                 DAD     D       ;ADDRESS OF CHAR
013A 71                 MOV     M,C     ;DISPLAY IT
013B D1                 POP     D
013C E1                 POP     H
013D C9                 RET
                ;
                ;LOCAL ROUTINES FOR MTXOUT
                ;
013E AF         OUTCR:  XRA     A       ;BEGINNING OF LINE
013F 320002             STA     CURSOR
0142 C9                 RET
                ;
0143 E5         OUTLF:  PUSH    H       ;SCROLL UP ONE LINE
0144 D5                 PUSH    D
0145 C5                 PUSH    B
0146 2180E0             LXI     H,MTXAD+128D ;SOURCE
0149 1100E0             LXI     D,MTXAD ;DESTINATION
014C 01500B             LXI     B,22D*128D+80D ;BYTE COUNT
014F 7E         OUT80:  MOV     A,M     ;Z-80 LDIR
0150 12                 STAX    D
0151 23                 INX     H
0152 13                 INX     D
0153 0B                 DCX     B
0154 78                 MOV     A,B
0155 B1                 ORA     C
0156 C24F01             JNZ     OUT80
0159 012050             LXI     B,80D*256D+' ' ;ZAP LAST LINE
015C 71         OUT85:  MOV     M,C
015D 2B                 DCX     H
015E 05                 DCR     B
015F C25C01             JNZ     OUT85
0162 C1                 POP     B
0163 D1                 POP     D
0164 E1                 POP     H
0165 C9                 RET
```

```
                        ;       DISPLAY THE CHARACTER IN C AT THE SCREEN
                        ;          POSITION INDICATED.
                        ;       ADJUST THE CHARACTER TO CORRESPOND WITH
                        ;          OUTPUT SWITCHES GREEK AND FORBAK.
                        ;       H AND L MUST CONTAIN A VALID, ON SCREEN POINT.
                        ;       REGISERS A, D, E AND FLAGS MODIFIED

031B E5         PUTUP:  PUSH    H       ;SAVE H,L FOR LATER
031C 5C                 MOV     E,H
031D 1600               MVI     D,0
031F 7D                 MOV     A,L     ;COLUMN WITH NEW CONTENTS
0320 21340A             LXI     H,LINFIL
0323 19                 DAD     D       ;ADDRESS OF PREVIOUS MAX
0324 BE                 CMP     M       ;HAVE MORE NOW?
0325 DA2B03             JC      PUTU0   ;  NO
0328 3C                 INR     A       ;FILL IS COL + 1
0329 77                 MOV     M,A     ;NEW MAXIMUM
032A 3D                 DCR     A       ;BACK TO COLUMN
032B 63         PUTU0:  MOV     H,E     ;RETRIEVE POSITION
032C 6F                 MOV     L,A
032D CDFA03             CALL    MTXAD   ;CALCULATE ADDRESS
0330 EB                 XCHG            ;PUT IN D,E
0331 21BD0A             LXI     H,GREEK ;NOTE:  GREEK AND FORBAK
0334 79                 MOV     A,C     ;  MUST BE CONSECUTIVE
0335 E660               ANI     60H     ;LOWER CASE?
0337 FE60               CPI     60H
0339 79                 MOV     A,C     ;GET FRESH COPY
033A C23E03             JNZ     PUTU1   ;NOT LC SO OK AS IS
033D A6                 ANA     M       ;CONVERT TO GREEK OR UC AS REQ
033E 23         PUTU1:  INX     H       ;SAME FOR FORGROUND/BACK
033F B6                 ORA     M       ;SET INVERT/BLINK AS REQ
0340 12                 STAX    D       ;DISPLAY IT
0341 E1                 POP     H       ;RESTORE HL
0342 C9                 RET

                        ;SUBROUTINE DELINE (H=LINE)
                        ;       SCROLL THE LINE INDICATED AND ALL LINES BENEATH
                        ;         IT UP ONE LINE.
                        ;       THE LINE INDICATED BY H IS LOST.
                        ;       H MUST CONTAIN A VALID LINE NUMBER BETWEEN
                        ;         0 AND 23 INCLUSIVE (NOT CHECKED).
                        ;       THE TOP LINE IS LOST.
                        ;       CP*L AND LINFIL ARE UPDATED AS REQUIRED.
                        ;       A, D, E, H, L AND FLAGS MODIFIED.


0343 2600       SCRLUP: MVI     H,0     ;DO THE WHOLE SCREEN
0345 C5         DELINE: PUSH    B       ;SAVE THE SACRED
0346 2E17               MVI     L,23D
0348 E5                 PUSH    H       ;AND CURSOR FOR LATER
0349 7D                 MOV     A,L     ;FIRST CORRECT LINFIL TABLE
034A 94                 SUB     H
034B CA8803             JZ      SCRLST
034E 214B0A             LXI     H,LINFIL+23D
0351 1E00               MVI     E,0     ;BOTTOM LINE GETS 0
0353 56         SLUP0:  MOV     D,M     ;GET CURRENT CONTENTS
0354 73                 MOV     M,E     ;SET TO NEW
```

```
0355 5A                 MOV     E,D     ;SET NEW TO CURRENT
0356 2B                 DCX     H       ;NEXT ENTRY
0357 3D                 DCR     A
0358 C25303             JNZ     SLUP0

                ; UPDATE CP*L POINTERS

035B C1                 POP     B       ;RETRIEVE CURSOR
035C 212D0A             LXI     H,CP1L  ;LINE POINTER
035F 0E03               MVI     C,CPNUM ;POINTER COUNT
0361 78                 MOV     A,B     ;LINE MOVED
0362 BE         SLUP1:  CMP     M       ;WAS POINTER MOVED
0363 D26703             JNC     SLUP2   ;NO
0366 35                 DCR     M       ;MOVE UP ONE LINE
0367 23         SLUP2:  INX     H       ;ON TO NEXT POINTER
0368 23                 INX     H
0369 0D                 DCR     C       ;IF ANY
036A C26203             JNZ     SLUP1
            ;FINALLY DO THE ACTUAL SCROLL
036D 60         SLUP4:  MOV     H,B     ;FIND ADDRESS
036E 2E00               MVI     L,0
0370 CDFA03             CALL    MTXAD
0373 3E17               MVI     A,23D   ;HOW MANY LINES?
0375 90                 SUB     B
0376 47                 MOV     B,A
0377 EB                 XCHG            ;DESTINATION IN DE
0378 05         SLUP5:  DCR     B       ;DONE YET?
0379 FDE03              JM      SLDN4   ;REST IS COMMON
037C 218000             LXI     H,128D  ;OFFSET TO SOURCE
037F 19                 DAD     D
0380 E5                 PUSH    H       ;SAVE FOR NEXT TIME
0381 CDED03             CALL    MOV80   ;COPY LINE UP
0384 D1                 POP     D       ;NEW DESTINATION
0385 C37803             JMP     SLUP5
0388 C1         SCRLST: POP     B       ;CLEAN STACK
0389 324B0A             STA     LINFIL+23D ;BOTTOM LINE IS EMPTY
038C C36D03             JMP     SLUP4

                ;SUBROUTINE SCRLDN (H = LINE)
                ;       SCROLL INDICATED LINE AND ALL LINES BENEATH IT
                ;         DOWN ONE LINE.
                ;       H MUST CONTAIN A VALID LNE NUMBER (0 - 23).
                ;       CP*L AND LINFIL ARE UPDATED AS REQUIRED.
                ;       A, FLAGS, D, E, H & L MODIFIED.

038F C5         SCRLDN: PUSH    B       ;SAVE THE SACRED
0390 2E17               MVI     L, 23D
0392 E5                 PUSH    H       ;SAVE ARGUMENTS FOR LATER
0393 7D                 MOV     A,L     ;HOW MANY LINES?
0394 94                 SUB     H
0395 CA8803             JZ      SCRLST  ;BOTTOM LINE IS SPECIAL
            ;UPDATE LINFIL TABLE
0398 214B0A             LXI     H,LINFIL+23D
039B 2B         SLDN0:  DCX     H       ;PICK UP NEW VALUE
039C 56                 MOV     D,M
039D 23                 INX     H       ; AND PUT IT WHERE
039E 72                 MOV     M,D     ; IT BELONGS
```

```
039F 2B                 DCX     H       ;NEXT ENTRY
03A0 3D                 DCR     A       ;ANY LEFT?
03A1 C29B03             JNZ     SLDN0   ;  YES.
03A4 3600               MVI     M,0     ;CLEAR LAST ENTRY
                    ;UPDATE CHARACTER POINTERS
03A6 D1                 POP     D       ;RETRIEVE PARAMETERS
03A7 D5                 PUSH    D
03A8 212D0A             LXI     H,CP1L  ;LINE POINTER
03AB 0E03               MVI     C,CPNUM ;LINE POINTER COUNT
03AD 7E         SLDN1:  MOV     A,M     ;GET POINTER
03AE BA                 CMP     D       ;ABOVE TOP LINE?
03AF DAB703             JC      SLDN2   ;YES, NO CHANGE REQ
03B2 BB                 CMP     E       ;ON OR BELOW BOTTOM LINE?
03B3 D2B703             JNC     SLDN2   ;  YES, NO CHANGE REQ
03B6 34                 INR     M       ;MOVE POINTER DOWN 1 LINE
03B7 23         SLDN2:  INX     H       ;NEXT POINTER
03B8 23                 INX     H
03B9 0D                 DCR     C       ;ANY LEFT?
03BA C2AD03             JNZ     SLDN1   ;YES

                    ;DO THE ACTUAL SCROLL
03BD C1                 POP     B       ;RETRIEVE PARAMETERS
03BE 79                 MOV     A,C
03BF 90                 SUB     B
03C0 47                 MOV     B,A     ;LINE COUNT IN B
03C1 AF                 XRA     A       ;MOV A,C W/ CY CLEAR
03C2 B1                 ORA     C
03C3 1F                 RAR             ;MULTIPLY LINE BY 128
03C4 57                 MOV     D,A
03C5 3E00               MVI     A,0
03C7 1F                 RAR
03C8 5F                 MOV     E,A     ;FINAL LINE OFFSET
03C9 2A4C0A             LHLD    MTXAT   ;BASE ADDRESS
03CC 19                 DAD     D       ;DESTINATION ADDRESS
03CD EB                 XCHG            ;GOES IN DE
03CE 05         SLDN3:  DCR     B       ;DONE YET?
03CF FADE03             JM      SLDN4   ;YES
03D2 2180FF             LXI     H,-128D ;UP A LINE
03D5 19                 DAD     D
03D6 E5                 PUSH    H       ;SAVE FOR NEXT
03D7 CDED03             CALL    MOV80   ;COPY A LINE
03DA D1                 POP     D       ;RETRIEVE LAST SOURCE
03DB C3CE03             JMP     SLDN3   ;AND TRY AGAIN

                    ;BLANK THE LINE
03DE 214F00     SLDN4:  LXI     H,79D   ;ALSO USED BY SCRLUP
03E1 19                 DAD     D
03E2 112050             LXI     D,80D*100H+' '
03E5 73         SLDN5:  MOV     M,E     ;BLAANK IT
03E6 2B                 DCX     H       ;NEXT
03E7 15                 DCR     D       ;IF ANY
03E8 C2E503             JNZ     SLDN5
03EB C1                 POP     B       ;RESTORE SACRED
03EC C9                 RET

                ;SUBROUTINE MOV80 (DE=DESTINATION, HL=SOURCE)
                ;       MOVE 80 CHARACTERS FROM ADDRESS IN HL TO THE ADD
```

```
                    ;          IN DE.
                    ;       A, D, E, FLAGS, H AND L MODIFIED.

03ED C5         MOV8O:  PUSH    B
03EE 0650               MVI     B,80D
03F0 7E         MOV8L:  MOV     A,M
03F1 12                 STAX    D       ;COPY TO GOAL
03F2 23                 INX     H       ;NEXT SOURCE
03F3 13                 INX     D       ; AND DEST
03F4 05                 DCR     B       ;ANY MORE?
03F5 C2F003             JNZ     MOV8L   ;  YES
03F8 C1                 POP     B
03F9 C9                 RET


                    ;SUBROUTINE MTXAD (H=LINE, L=COLUMN)
                    ;       CALCULATE ADDRESS FOR LINE/COLUMN IN H,L.
                    ;       ALL REGISTERS EXCEPT B & C MODIFIED.
03FA 3ABC0A     MTXAD:  LDA     WIDTH   ;40 OR 80 WIDE?
03FD FE29               CPI     41D     ;IS IT 40 OR LESS?
03FF D20504             JNC     MTXA0   ;  NO.
0402 7D                 MOV     A,L     ;MULTIPLY COLUMN BY 2
0403 87                ADD     A
0404 6F                 MOV     L,A
0405 7C         MTXA0:  MOV     A,H     ;TAKE LINE NUMBR
0406 29                 DAD     H       ;COLUMN * 2
0407 B7                 ORA     A       ;CLEAR CY SO CAN GET
0408 1F                 RAR             ;  (LINE+COL*2)/2
0409 67                 MOV     H,A
040A 7D                 MOV     A,L
040B 1F                 RAR
040C 6F                 MOV     L,A
040D EB                 XCHG            ;DE = LINE*128D + COLUMN
040E 2A4C0A             LHLD    MTXAT   ;ADR OF LINE 0, COL 0
0411 19                 DAD     D       ;DESIRED ADDRESS
0412 C9                 RET

                    ;SUBROUTINE PUTIN (H=LINE, L=COLUMN)
                    ;       MAKE ROOM ON A LINE FOR A NEW CHARACTER
                    ;       IF LAST CHARACTER POSITION ON THE LINE IS
                    ;         NOT A SPACE, HAS NO EFFECT.
                    ;       LINFIL AND CP*C ARE UPDATED AS REQUIRED.
                    ;       A, D, E, FLAGS, H & L MODIFIED.

0413 0E20       PUTSPC: MVI     C,' '   ;INSERT  A SPACE
0415 C5         PUTIN:  PUSH    B
0416 E5                 PUSH    H
0417 E5                 PUSH    H       ;SAVE EXTRA FPOR LATER
0418 11340A             LXI     D,LINFIL        ;CHECK IF ROOM FOR ANOT
041B 6C                 MOV     L,H
041C 2600               MVI     H,0
041E 19                 DAD     D       ;LOOK UP CURRENT FILL
041F D1                 POP     D       ;GET LINE AND COL
0420 3ABC0A             LDA     WIDTH   ;FULL LINE SIZE
0423 BE                 CMP     M       ;IS THERE ROOM?
0424 CA3504             JZ      PUTIO   ;  NO
0427 34                 INR     M       ;UPDATE LINFIL
```

```
0428 7E                 MOV     A,M     ;INSERT AFTER LINE END?
0429 BB                 CMP     E
042A D23804             JNC     PUTIA   ;  NO
042D 73                 MOV     M,E     ;THIS CHAR IS LAST
042E E1                 POP     H       ;WHERE DOES IT GO?
042F CDFA03             CALL    MTXAD
0432 71                 MOV     M,C     ;STUFF IT
0433 C1                 POP     B
0434 C9                 RET
                    ;CAN'T BE DONE
0435 C1         PUTIO:  POP     B       ;CLEAN UP STACK
0436 C1                 POP     B
0437 C9                 RET
                    ;SHIFT THE LINE OVER ONE COLUMN
0438 93         PUTIA:  SUB     E       ;NOW MANY COL NEED SHIFTING
0439 47                 MOV     B,A     ;SAVE COUNTER IN B
043A EB                 XCHG            ;GET IN POSITION
043B CDFA03             CALL    MTXAD   ;PHYSICAL ADDRESS OF INSERT
043E 7E         PUTID:  MOV     A,M     ;GET OLD
043F 71                 MOV     M,C     ;STUFF WITH NEW
0440 4F                 MOV     C,A     ;MAKE OLD NEW FOR NEXT
0441 23                 INX     H       ;NEXT COLUMN
0442 3ABC0A             LDA     WIDTH   ;LOW RESOLUTION MODE?
0445 FE29               CPI     41D
0447 D24B04             JNC     PUTIC   ;NO, OK AS IS
044A 23                 INX     H       ;EVERY OTHER
044B 05         PUTIC:  DCR     B       ;ANY LEFT?
044C F23E04             JP      PUTID   ;YES
                    ;FIX CURSOR POINTERS AFFECTED
044F C1                 POP     B       ;GET CURSOR
0450 212D0A             LXI     H,CP1L  ;CHECK IF ON LINE
0453 1603               MVI     D,CPNUM ;INIT COUNTER
0455 7E         PUTIE:  MOV     A,M     ;WHICH LINE?
0456 B8                 CMP     B       ;THE ONE MOVED?
0457 CA6204             JZ      PUTIF   ;YES, CHECK IT OUT
045A 23         PUTIH:  INX     H       ;MMOVE TO NEXT
045B 23                 INX     H
045C 15                 DCR     D       ;ANY LEFT TO CHECK?
045D C25504             JNZ     PUTIE   ;  YES
0460 C1                 POP     B       ;RESTORE SACRED
0461 C9                 RET
0462 2B         PUTIF:  DCX     H       ;BACK UP TO COL
0463 7E                 MOV     A,M
0464 B9                 CMP     C       ;TO LEFT OF INSERT?
0465 D27204             JNC     PUTIG   ;  YES, NO CORRECTION REQ
0468 34                 INR     M       ;MOVE OVER ONE
0469 3ABC0A             LDA     WIDTH   ;CHECK FOR IN RANGE
046C BE                 CMP     M
046D C27204             JNZ     PUTIG   ;OK
0470 36FF               MVI     M,0FFH  ;FLAG AS OFF (SHOULD NEVER HAPPE
0472 23         PUTIG:  INX     H       ;BACK TO LINE
0473 C35A04             JMP     PUTIH   ;DO NEXT


                    ;SUBROUTINE GETCON (B=COMMAND, C=PARAMETER)
                    ;       PROCESSOR FOR CONFIGURATION SWITCHES.
                    ;       USES TABLE SETTAB FOR DEFINITIONS.
```

```
                ;       CALLS SETTAB ROUTINES WITH
                ;          B=COMMAND          C=PARAMETER
                ;          D=PARAMETER-'0'
                ;          E=PARAMETER-'0'+ 60 HEX IF D NEGATIVE
                ;       ALL REGISTERS MODIFIED

0476 78         SETCON: MOV     A,B     ;VERIFY VALID COMMAND
0477 E65F               ANI     5FH     ;INSURE UC
0479 D641               SUI     'A'
047B F8                 RM              ;TOO SMALL
047C FE14               CPI     (SETEND-SETTAB)/2
047E D0                 RNC             ;TOO LARGE
047F 5F                 MOV     E,A     ;CALCULATE ENTRY
0480 1600               MVI     D,0
0482 215505             LXI     H,SETTAB
0485 19                 DAD     D
0486 19                 DAD     D       ;2 BYTES/ENTRY
0487 5E                 MOV     E,M     ;PICK UP ADDRESS
0488 23                 INX     H
0489 56                 MOV     D,M
048A EB                 XCHG
048B 79                 MOV     A,C     ;CALCULATE PARAMETER
048C E67F               ANI     7FH     ;  VARIANTS
048E D630               SUI     '0'
0490 57                 MOV     D,A
0491 F29604             JP      SETC0
0494 C660               ADI     60H
0496 5F         SETC0:  MOV     E,A
0497 E9                 PCHL            ;GO TO IT
0498 C9         SETNOT: RET             ;UNDEFINED, IGNORE

                ;
                ;       SETCON CALL ROUTINES
                ;

                    ;SET ALT-2480 BASE ADDRESS
0499 7A         SETADR: MOV     A,D
049A FE10               CPI     10H     ;MAKE SURE VALID
049C D0                 RNC             ;IT ISN'T
049D 07                 RLC             ;TIMES 16
049E 07                 RLC
049F 07                 RLC
04A0 07                 RLC
04A1 324D0A             STA     MTXAT+1 ;NEW HIGH ADDRESS BYTE
04A4 C9                 RET

                    ;SET OFFBOT SWITCH
04A5 7A         SETBOT: MOV     A,D
04A6 FE19               CPI     25D
04A8 D0                 RNC             ;ILLEGAL
04A9 32C40A             STA     OFFBOT
04AC C9                 RET

                    ;SET OFFTOP SWITCH
04AD 7A         SETTOP: MOV     A,D
04AE FE19               CPI     25D
04B0 D0                 RNC
```

```
04B1 32C30A             STA     OFFTOP
04B4 C9                 RET

                    ;SET OFFRT SWITCH
                    ;   1 SETS TO 1, 0 SETS TO WIDTH
                    ;   2 + SETS TO 1-WIDTH
04B5 7A         SETRT:  MOV     A,D
04B6 FE01               CPI     1
04B8 CAC404             JZ      SETR0
04BB 3ABC0A             LDA     WIDTH
04BE DAC404             JC      SETR0
04C1 2F                 CMA
04C2 3C                 INR     A
04C3 3C                 INR     A       ;1-WIDTH
04C4 32C60A     SETR0:  STA     OFFRT
04C7 C9                 RET

                    ;SET OFFLFT SWITCH
                    ;   SAVE ACTION AS SETRT
04C8 7A         SETLFT: MOV     A,D
04C9 FE01               CPI     1
04CB CAD704             JZ      SETL0
04CE 3ABC0A             LDA     WIDTH
04D1 DAD704             JC      SETL0
04D4 2F                 CMA
04D5 3C                 INR     A
04D6 3C                 INR     A
04D7 32C50A     SETL0:  STA     OFFLFT
04DA C9                 RET

                    ;SET GREEK FOR GREEK TRANSLATION
04DB 3E9F       SETGRK: MVI     A,9FH
04DD 32BD0A             STA     GREEK
04E0 C9                 RET

                    ;SET GREEK FOR UPPER CASE ONLY
04E1 3EDF       SETUC:  MVI     A,0DFH
04E3 32BD0A             STA     GREEK
04E6 C9                 RET

                    ;SET GREEK FOR NORMAL LOWER CASE
04E7 3EFF       SETLC:  MVI     A,0FFH
04E9 32BD0A             STA     GREEK
04EC C9                 RET

                    ;SET DISPLAY WIDTH (LINE LENGTH)
04ED 7B         SETWDH: MOV     A,E
04EE 32BC0A             STA     WIDTH
04F1 C9                 RET

                    ;DEFINE A NEW CURSOR CHARACTER
04F2 79         SETCUR: MOV     A,C
04F3 32550A             STA     CURSOR
04F6 C9                 RET

                    ;DEFINE A NEW ESCAPE, ATTN, XON OR XOFF
04F7 21570A     SETESC: LXI     H,ESCAPE
```

```
04FA C30F05             JMP     SETCHR
04FD 21500A   SETATT:   LXI     H,ATTN
0500 C30F05             JMP     SETCHR
0503 21510A   SETXFF:   LXI     H,XOFF
0506 C30F05             JMP     SETCHR
0509 21520A   SETXN:    LXI     H,XON
050C C30F05             JMP     SETCHR

050F 79       SETCHR:   MOV     A,C       ;NEW CHARACTER
0510 E67F               ANI     7FH       ;NO PARITY ALLOWED
0512 77                 MOV     M,A
0513 C9                 RET

                ;SET INPUT TTY LOCK
                ;   OFF IF 0,  OTHERWISE ON.
0514 7A       SETTTY:   MOV     A,D
0515 B7                 ORA     A
0516 2F                 CMA               ;ASSUME OFF
0517 CA1C05             JZ      SETLK
051A 3EDF               MVI     A,0DFH    ;TURN ON
051C 32530A   SETLK:    STA     UCLOCK
051F C9                 RET

                ;SET INPUT MODE
                ;   0 = FULL DUPLEX, 1= HALF DUPLEX
                ;   2 = BLOCK MODE
0520 7A       SETMODE:  MOV     A,D
0521 B7                 ORA     A
0522 F8                 RM
0523 FE03               CPI     3
0525 D0                 RNC
0526 2F                 CMA
0527 3C                 INR     A
0528 32540A             STA     FDUX      ;SET FLAG
052B 3C                 INR     A         ;SET ECONTAB TO MATCH
052C 21D10A             LXI     H,CONTAB  ;ASSUME FDUX
052F CA3B05             JZ      SETMF
0532 21CE0A             LXI     H,CONBLK  ;MAYBE BLOCK?
0535 FA3B05             JM      SETMF
0538 211C0B             LXI     H,HDCON   ;MUST BE HALF DUPLEX
053B 22B30A   SETMF:    SHLD    ECONAT
053E 2A320A             LHLD    CURSAT    ;XMIT POINTER
0541 222C0A             SHLD    CP1C
0544 21610A             LXI     H,LINBUF  ;LINE BUFFER POINTER
0547 225E0A             SHLD    LBPTR
054A AF                 XRA     A         ;LINE BUFFER FILL COUNT
054B 32600A             STA     LBCNT
054E 325C0A             STA     LDONE     ;NO LINE AVAIL
0551 325D0A             STA     BDONE     ;NO BLOCK AVAIL
0554 C9                 RET


            ;LOOKUP TABLE SETTAB
;               CONFIGURATION SWITCH SETTING DEFINITIONS
;               FORMAT IS
;                   ADDRESS     ;FOR SWITCH 'A'
;                   ADDRESS     ;FOR SWITCH  'B'
```

```
                        ;              ETC.
                        ;        SETEND MUST BE DEFINED TO SET TABLE LENGTH.

0555 9904       SETTAB: DW      SETADR  ;A=SET 2480 ADDRESS
0557 A504               DW      SETBOT  ;B=SET OFFBOT SWITCH
0559 F204               DW      SETCUR  ;C=SET CURSOR CHARACTER
055B E704               DW      SETLC   ;D=DISPLAY LOWER CASE AS LC
055D F704               DW      SETESC  ;E=DEFINE ESCAPE CHARACTER
055F 0305               DW      SETXFF  ;F=DEFINE XOFF CHARACTER
0561 DB04               DW      SETGRK  ;G=DISPLAY LOWER CASE AS GREEK
0563 E104               DW      SETUC   ;H=DISPLAY LOWER CASE AS UPPER
0565 FD04               DW      SETATT  ;I=DIFINE ATTN CHARACTER
0567 ED04               DW      SETWDH  ;J=SET DISPLAY WIDTH
0569 9804               DW      SETNOT  ;K=
056B C804               DW      SETLFT  ;L=SET OFFLFT SWITCH
056D 2005               DW      SETMODE ;M=SET INPUT MODE
056F 0905               DW      SETXN   ;N=DEFINE XON CHARACTER
0571 9804               DW      SETNOT  ;O=
0573 9804               DW      SETNOT  ;P=
0575 9804               DW      SETNOT  ;Q=
0577 B504               DW      SETRT   ;R=SET OFFRT SWITCH
0579 1405               DW      SETTTY  ;S=SET/RESET TTY LOCK
057B AD04               DW      SETTOP  ;T=SET OFFTOP SWITCH
057D            SETEND: DS      0       ;END OF TABLE

                        ;**********************

                        ;
                        ;       INPUT SUBROUTINES
                        ;


                        ;SUBROUTINE INFDX
                        ;       BASIC KEYBOARD READ ROUTINE
                        ;       RETURNS NEXT USER KEYSTROKE IN A
                        ;       FLASHES CURSOR TO PROMPT USER.
                        ;       CHECKS FOR SPECIAL CHARACTERS
                        ;          SETC-SET CONFIGURATION SWITCHES
                        ;          ATTN-RETURN TO MONITOR
                        ;          XOFF-HALT OUTPUT.
                        ;          XON -RESUME OUTPUT
                        ;       A, D, E, FLAGS, H AND L MODIFIED.

057D 2AB50A     INFDX:  LHLD    INTRAP  ;CHECK FOR SPECIALS
0580 7C                 MOV     A,H
0581 B5                 ORA     L
0582 CA8605             JZ      INFDK   ;PROCEED
0585 E9                 PCHL            ;CHECK IT OUT
0586 2A320A     INFDK:  LHLD    CURSAT  ;GET CURSOR
0589 CDD902             CALL    TSTCUR  ;MAKE SURE ON SCREEN
058C CDFA03             CALL    MTXAD   ;CONVERT TO ADDRESS
058F 4E                 MOV     C,M     ;SAVE CURRENT CONTENTS
0590 3A550A             LDA     CURSOR  ;  IN C, CURSOR CHAR
0593 47                 MOV     B,A     ;  IN B
                    ;WAIT FOR USER INPUT
                        ;REGISTERS ARE SET UP AS FOLLOWS:
                        ;    B=CURSOR CHARACTER
```

```
                        ;     C=ORIGINAL SCREEN CONTENTS AT CURSOR POS
                        ;     HL=ADDRESS OF SCREEN CHAR CORRESPONDING
                        ;        TO CURRENT CURSOR POSITION.
0594 CDA507   INFD0:    CALL    STFDX    ;ANYTHING AVAILABLE?
0597 C2B405             JNZ     INFDD    ;FINALLY
059A 3E80               MVI     A,80H    ;KILL SOME TIME
059C 3D       INFD3:    DCR     A
059D C29C05             JNZ     INFD3
05A0 3A590A             LDA     FLASH    ;BUMP FLASH COUNTER
05A3 3C                 INR     A
05A4 32590A             STA     FLASH
05A7 C2AB05             JNZ     INFD1    ;TIME FOR CURSOR?
05AA 70                 MOV     M,B      ;YES
05AB FE80     INFD1:    CPI     80H      ;TIME FOR CURRENT?
05AD C2B105             JNZ     INFD2    ;  NO
05B0 71                 MOV     M,C
05B1 C39405   INFD2:    JMP     INFD0    ;KEEP TRYING

                    ;GET A CHARACTER AND CHECK IT OUT
05B4 71       INFDD:    MOV     M,C      ;RESTORE DISPLAY
05B5 CD0601             CALL    INKBD    ;GET USER INPUT
                                         ;  ALL REGISTERS EXPENDABLE
05B8 4F                 MOV     C,A      ;SAVE USER INPUT
05B9 214F0A             LXI     H,SETC   ;CHECK IF SPECIAL
05BC E67F               ANI     7FH      ;REMOVE PARITY BEFORE CHECKING
05BE BE                 CMP     M        ;SETC?
05BF CAD207             JZ      KSETC    ;  YES
05C2 23                 INX     H
05C3 BE                 CMP     M        ;ATTN?
05C4 CAC007             JZ      BREAK    ;  YES
05C7 23                 INX     H
05C8 BE                 CMP     M        ;XOFF?
05C9 C2D105             JNZ     INFDE    ;  NO
05CC 3EFF               MVI     A,0FFH   ;  YES
05CE C3D705             JMP     INFDF
05D1 23       INFDE:    INX     H
05D2 BE                 CMP     M        ;XON?
05D3 C2DD05             JNZ     INFDZ    ;  NO
05D6 AF                 XRA     A        ;CLEAR FLAG
05D7 324E0A   INFDF:    STA     XOFFD    ;SET XOFFED AS REQ.
05DA AF       INFDG:    XRA     A        ;RETURN NULL
05DB 37                 STC              ;  BUT WITH CY SET
05DC C9                 RET

05DD FE7F     INFDZ:    CPI     DEL      ;SPECIAL CASE
05DF CAEE05             JZ      INFDU    ;  NOT REALLY LC BUT IS.
05E2 E660               ANI     60H      ;LOWER CASE?
05E4 EE60               XRI     60H      ;CLEAR CY REGARDLESS
05E6 C2EE05             JNZ     INFDU    ;OK AS IS
05E9 3A530A             LDA     UCLOCK   ;FIX UP AS REQUIRED
05EC A1                 ANA     C
05ED 4F                 MOV     C,A
05EE 79       INFDU:    MOV     A,C      ;SET UP FOR RETURN
05EF C9                 RET              ;NOTE: CY MUST BE CLEAR


;
```

```
                        ;HALF DUPLEX INPUT PROCESSING
                        ;

05F0 3A5C0A     INHDX:  LDA     LDONE   ;GOT A LINE YET?
05F3 B7                 ORA     A
05F4 CC1006             CZ      INHDY   ;GO READ ONE
05F7 2A5E0A             LHLD    LBPTR   ;BUFFER POINTER
05FA 7E                 MOV     A,M     ;GET NEXT CHARACTER
05FB 23                 INX     H       ;UPDATE POINTER
05FC 225E0A             SHLD    LBPTR   ;NEW POINTER
05FF 21600A             LXI     H,LBCNT ;CHARACTER COUNT
0602 35                 DCR     M       ;  IS ONE LESS
0603 C0                 RNZ             ;DONE IF NOT LAST
0604 215C0A             LXI     H,LDONE ;RESET TO EMPTY
0607 3600               MVI     M,0
0609 21610A             LXI     H,LINBUF
060C 225E0A             SHLD    LBPTR
060F C9                 RET

                    ;FILL UP THE LINE BUFFER

0610 CD7D05     INHDY:  CALL    INPDX   ;GET A CHAR
0613 47         INHD0:  MOV     B,A     ;SAVE ORIGINAL
0614 E67F               ANI     7FH     ;CLEAR PARITY
0616 4F                 MOV     C,A     ; AND SAVE A COPY
0617 FE7F               CPI     DEL     ;RUBOUT?
0619 CA5D06             JZ      RUBOUT  ;  YES
061C FE15               CPI     NAK     ;LINE CANCEL?
061E CA8206             JZ      CANCEL  ;  YES
                    ;ADD CHAR TO BUFFER AND ECHO IT
0621 CD0002             CALL    ECHOCH  ;ECHO IT
0624 2A5E0A             LHLD    LBPTR   ;BUFFER POINTER
0627 70                 MOV     M,B     ;PUT IN BUFFER
0628 23                 INX     H       ;NEXT
0629 225E0A             SHLD    LBPTR   ;NEW POINTER
062C 21600A             LXI     H,LBCNT ;CHAR COUNTER
062F 7E                 MOV     A,M
0630 34                 INR     M       ;BUMP IT
0631 FE4F               CPI     LINSIZ-1 ;FULL LINE?
0633 CA5206             JZ      INHD1   ;YES
0636 79                 MOV     A,C     ;GET COPY WITHOUT PARITY
0637 FE0D               CPI     CR      ;CARRIAGE RETURN?
0639 CA5806             JZ      INHDC   ;  APPEND A LINEFEED
063C FE1B               CPI     ESC     ;ESCAPE?
063E CA4606             JZ      INHDZ   ;  END THE LINE
0641 FE0A               CPI     LF      ;LINE FEED?
0643 C21006             JNZ     INHDY   ;  NO, GET ANOTHER CHARACTER
0646 3EFF       INHDZ:  MVI     A,0FFH  ;SET LINE COMPLETE FLAG
0648 325C0A             STA     LDONE
064B 21610A             LXI     H,LINBUF        ;RESET BUFFER POINTER
064E 225E0A             SHLD    LBPTR   ;  TO SCAN BUFFER
0651 C9                 RET
0652 79         INHD1:  MOV     A,C     ;TEST FOR CR
0653 FE0D               CPI     CR      ;  WHIH IS A SPECIAL
0655 C24606             JNZ     INHDZ   ;  CASE IN LINE OVERFLOW
0658 3E0A       INHDC:  MVI     A,LF    ;APPEND A LINEFEED
065A C31306             JMP     INHD0
```

```
                    ;RUBOUT LAST TYPED CHARACTER

065D 21600A   RUBOUT: LXI     H,LBCNT ;FIX CHAR COUNT
0660 7E               MOV     A,M
0661 B7               ORA     A       ;ANYTHING TO DELETE?
0662 CA1006           JZ      INHDY   ;NO
0665 35               DCR     M       ;ONE LESS CHAR IN BUFFER
0666 2A5E0A           LHLD    LBPTR   ;FIX POINTER
0669 7E               MOV     A,M     ;CHECK WHAT IS GETTING ZAPPED
066A 2B               DCX     H
066B 225E0A           SHLD    LBPTR
066E 21320A   INHDR:  LXI     H,CURSC ;BACK UP ONE SPACE
0671 35               DCR     M       ;NOTE: THIS ALGORITHM IS NOT
0672 0E20             MVI     C,' '   ;  GOOD WITH TABS
0674 CD0002           CALL    ECHOCH
0677 35               DCR     M
0678 E660             ANI     60H     ;WAS IT A CONTROL CHAR?
067A 3EFF             MVI     A,0FFH  ;  IF SO, REPEAT TO DELETE
067C CA6E06           JZ      INHDR   ;  THE PRECEEDING UP ARROW
067F C31006           JMP     INHDY   ;PROCESS NEXT

                    ;CANCEL THE ENTIRE LINE TYPED
                    ;   NOTE:  IF LINE HAS OVERFLOWED ONTO NEXT LINE,
                    ;     THIS ALGORITHM WILL NOT CLEAN UP PREVIOUS PHYS

0682 21320A   CANCEL: LXI     H,CURSC ;CURRENT COLUMN
0685 0E20             MVI     C,' '   ;FILL WITH BLANKS
0687 35       INHDD:  DCR     M       ;BACK UP
0688 FA9206           JM      INHDE   ;ALL DONE
068B CD0002           CALL    ECHOCH  ;BLANK IT
068E 35               DCR     M
068F C38706           JMP     INHDD   ;REPEAT TO COL 0
                    ;RESET POINTERS AND COUNTERS
0692 34       INHDE:  INR     M       ;FIX UP CURSOR
0693 AF               XRA     A
0694 32600A           STA     LBCNT   ;NO BUFFER CONTENTS
0697 23               INX     H       ;WHAT LINE DID WE ZAP?
0698 4E               MOV     C,M
0699 47               MOV     B,A     ;SET APPROPRIATE ENTRY
069A 21340A           LXI     H,LINFIL ;  IN LINFIL TABLE TO
069D 09               DAD     B       ;   ZERO
069E 77               MOV     M,A
069F 21610A           LXI     H,LINBUF
06A2 225E0A           SHLD    LBPTR   ;RESET POINTER
06A5 C31006           JMP     INHDY   ;NEXT CHARACTER


                    ;
                    ;BLOCK MODE INPUT ROUTINE
                    ;

06A8 3A5D0A   INBLK:  LDA     BDONE   ;GOT A BLOCK YET?
06AB B7               ORA     A
06AC CC4B07           CZ      INBLN   ;  NO, GET ONE
06AF 2A2C0A           LHLD    CP1C    ;GET 'AT' POINTER
06B2 7C               MOV     A,H     ;CHECK IF ON SCREEN
```

```
06B3 FE18           CPI     24D     ;OFF BOTTOM OR TOP?
06B5 D2CB06         JNC     INBLZ   ;YES.  FATAL ERROR
06B8 B5             ORA     L       ;IS COLUMN POSITIVE?
06B9 F2C206         JP      INBL0   ;SEEMS TO BE
06BC 210000         LXI     H,00    ;START AT COL 0 LINE 0
06BF 222C0A         SHLD    CP1C
06C2 EB       INBL0:XCHG            ;PUT AT IN DE
06C3 2A2E0A         LHLD    CP2C    ;GET LAST POINTER
06C6 7C             MOV     A,H     ;IS IT OFF SCREEN?
06C7 B5             ORA     L
06C8 F2E106         JP      INBL1   ;STILL ON SCREEN
06CB 2A2E0A   INBLZ:LHLD    CP2C    ;COPY END POINTER TO
06CE 7C             MOV     A,H     ;  AT POINTER UNLESS
06CF B5             ORA     L       ;  WOULD BE OFF SCREEN,
06D0 3C             INR     A       ;  IN WHICH CASE GO BACK
06D1 C2D706         JNZ     INBLU   ;  TO LINE 0 COL 0
06D4 210000         LXI     H,0000H
06D7 222C0A   INBLU:SHLD    CP1C
06DA 215D0A         LXI     H,BDONE ;RESET BDONE
06DD 7E             MOV     A,M     ;WHILE PICKING
06DE 3600           MVI     M,0     ;  UP TERMINATOR.
06E0 C9             RET
06E1 7A       INBL1:MOV     A,D     ;TEST FOR END OF TEXT
06E2 BC             CMP     H       ;HOW DO LINES COMPARE
06E3 DAEE06         JC      INBL2   ;NOT DOWN TO LAST YET
06E6 C2CB06         JNZ     INBLZ   ;WENT TOO FAR!
06E9 7B             MOV     A,E     ;COLUMN?
06EA BD             CMP     L
06EB D2CB06         JNC     INBLZ   ;THAT'S ALL FOLKS
06EE 4A       INBL2:MOV     C,D     ;CURRENT LINE
06EF 0600           MVI     B,0     ;ANY DATA LEFT ON IT?
06F1 21340A         LXI     H,LINFIL
06F4 09             DAD     B
06F5 7E             MOV     A,M     ;CHARACTRS ON LINE
06F6 EB             XCHG            ;MEANTIME...
06F7 2C             INR     L       ;   SET POINTERS FOR
06F8 222C0A         SHLD    CP1C    ;   NEXT ITERATION
06FB 2D             DCR     L       ;BACK TO PRESENT
06FC BD             CMP     L       ;PAST END OF LINE?
06FD CA0C07         JZ      INBL3   ;SEND CR
0700 DA0F07         JC      INBL4   ;SEND LF
0703 CDFA03         CALL    MTXAD   ;SEND NEXT CHARACTER
0706 7E             MOV     A,M
0707 B7             ORA     A       ;BACKGROUND?
0708 FA1807         JM      INBLF   ;YES.
070B C9             RET
070C 3E0D     INBL3:MVI     A,CR    ;CARRIAGE RETURN
070E C9             RET
070F 24       INBL4:INR     H       ;DOWN 1 LINE
0710 2E00           MVI     L,0     ;FIRST COLUMN
0712 222C0A         SHLD    CP1C    ;NEW AT POINTER
0715 3E0A           MVI     A,LF    ;RETURN LF
0717 C9             RET
0718 212C0A   INBLF:LXI     H,CP1C  ;BACK UP TO CHECKED
071B 35             DCR     M
071C 2A2C0A   INBL5:LHLD    CP1C    ;SKIP TO NEXT FOREGROUND
071F 2C             INR     L       ;  FIELD
```

```
0720 222C0A             SHLD    CP1C    ;TRY NEXT CHARACTER
0723 3E17               MVI     A,23D   ;CHECK FOR OFF BOTTOM
0725 BC                 CMP     H
0726 3E09               MVI     A,HT    ;ASSUME OFF
0728 D8                 RC
0729 4C                 MOV     C,H     ;HOW LONG IS CURRENT LINE?
072A 0600               MVI     B,0
072C EB                 XCHG
072D 21340A             LXI     H,LINFIL
0730 09                 DAD     B
0731 7B                 MOV     A,E     ;LINE LENGTH
0732 BE                 CMP     M       ;TRIED THEM ALL?
0733 EB                 XCHG
0734 D24207             JNC     INBL6   ;YES.  NEXT LINE
0737 CDFA03             CALL    MTXAD   ;CONVERT TO ADDRESS
073A 7E                 MOV     A,M     ;FORGROUND?
073B B7                 ORA     A
073C FA1C07             JM      INBL5   ;NO, TRY NEXT
073F 3E09               MVI     A,HT
0741 C9                 RET
                ;MOVE DOWN TO NEXT LINE
0742 24         INBL6:  INR     H
0743 2E00               MVI     L,0     ;RESET COL
0745 222C0A             SHLD    CP1C
0748 C31807             JMP     INBLF

                ;ACCEPT A BLOCK FROM THE KEYBOARD

074B 2A2C0A     INBLN:  LHLD    CP1C    ;IF START POINTER IS
074E 7C                 MOV     A,H     ;  OFF SCREEN, RESET TO HOME
074F B5                 ORA     L       ;  BEFORE PROCEEDING
0750 3C                 INR     A
0751 C25A07             JNZ     INBLL   ;OK AS IS
0754 210000             LXI     H,0000H ;RESET REQUIRED
0757 222C0A             SHLD    CP1C
075A CD7D05     INBLL:  CALL    INFDX   ;GET A CHARACTER
075D E67F               ANI     7FH     ;CLEAR PARITY
075F 4F                 MOV     C,A     ;SAVE A COPY
0760 3A560A             LDA     FIXUP   ;WAS PREVIOUS AN ESCAPE?
0763 B7                 ORA     A
0764 3E00               MVI     A,0     ;CLEAR FLAG REGARDLESS
0766 32560A             STA     FIXUP
0769 CA7007             JZ      INBLM   ;NO FIXUP REQUIRED
076C 79                 MOV     A,C     ;PATCH IT UP
076D F680               ORI     80H
076F 4F                 MOV     C,A
0770 79         INBLM:  MOV     A,C     ;GET  A COPY
0771 21570A             LXI     H,ESCAPE ;GOT AN ESCAPE?
0774 BE                 CMP     M
0775 CA8907             JZ      INBLE   ;YES
0778 23                 INX     H       ;GOT AN END OF TEXT?
0779 BE                 CMP     M
077A CA9A07             JZ      INBLT   ;  YES
077D EE80               XRI     80H     ;GOT SET START?
077F BE                 CMP     M       ;  =ESC EOT
0780 CA9107             JZ      INBLV   ;YES
0783 CD0002             CALL    ECHOCH  ;ECHO IT
```

```
0786 C35A07             JMP     INBLL   ;AND GET ANOTHER

0789 3EFF       INBLE:  MVI     A,0FFH  ;SET FIXUP FLAG
078B 32560A             STA     FIXUP
078E C35A07             JMP     INBLL   ;GET ANOTHER CHARACTER

0791 2A320A     INBLV:  LHLD    CURSAT  ;SET START MARKER
0794 222C0A             SHLD    CP1C
0797 C35A07             JMP     INBLL   ;GET THE NEXT
079A 79         INBLT:  MOV     A,C     ;SET BUFFER READY FLAG
079B 325D0A             STA     BDONE
079E 2A320A             LHLD    CURSAT  ;SET END MARKER
07A1 222E0A             SHLD    CP2C
07A4 C9                 RET


                ;INPUT STATUS ROUTINES
                ;       RETURN A=0 (Z=1) IF NO CHAR AVAILABLE
                ;       RETURN A=FF(HEX) (Z=0) IF A CHAR IS AVAILABLE
                ;       A AND FLAGS MODIFIED.

                    ;FULL DUPLES INPUT MODE
07A5 C5         STFDX:  PUSH    B       ;EXTERNAL INTERFACES ARE
07A6 D5                 PUSH    D       ;  INHERENTLY UNTRUSTWORTHY
07A7 E5                 PUSH    H
07A8 CD0301             CALL    .INKBS  ;KEYBOARD READY?
07AB E1                 POP     H
07AC D1                 POP     D
07AD C1                 POP     B
07AE C3BA07             JMP     STHDB   ;TEST FLAGS

                    ;HALF DUPLEX INPUT MODE
07B1 3A5C0A     STHDX:  LDA     LDONE   ;SEEN TERMINATOR YET?
07B4 C3BA07             JMP     STHDB   ;SET FLAGS ACCORDINGLY

                    ;BLOCK INPUT MODE
07B7 3A5D0A     STBLK:  LDA     BDONE   ;'END OF TEXT' ENTERED?
07BA B7         STHDB:  ORA     A       ;RETURN Z=1, A=0 IF A=0
07BB C8                 RZ              ;       Z=0, A=-1 IF A NOT 0
07BC 3EFF               MVI     A,0FFH
07BE B7                 ORA     A
07BF C9                 RET

                ;SUBROUTINE BREAK
                ;       JUMP TO MONITOR ENTRY POINT
                ;       WILL NOT JUMP IF ADDRESS IS FFFF HEX.

07C0 21DA05     BREAK:  LXI     H,INFDG
07C3 E5                 PUSH    H       ;RETURN ADDRESS
07C4 2A5A0A             LHLD    MONLVL  ;BREAK ADDRESS
07C7 7C                 MOV     A,H     ;CHECK IF SPECIFIED
07C8 A5                 ANA     L
07C9 3C                 INR     A
07CA CACE07             JZ      BRK00   ;NONE SPECIFIED
07CD E9                 PCHL            ;JMP TO IT
07CE 3A500A     BRK00:  LDA     ATTN    ;RESTORE A
07D1 C9                 RET             ;AND GO TO INFDG
```

```
                ;SUBROUTINE KSETC
                ;       TRAP NEXT TWO KEYSTROKES TO SET CONFIGURATION
                ;       PROMPTS USER TO AVOID CONFUSION.

07D2 2A320A     KSETC:  LHLD    CURSAT  ;SAVE CURSOR
07D5 E5                 PUSH    H
07D6 2A4C0A             LHLD    MTXAT   ;DISPLAY ADDRESS
07D9 1610               MVI     D,10H   ;SAVE TOP LINE
07DB 46         KSET0:  MOV     B,M
07DC 3620               MVI     M,' '   ;CLEAR SOME ECOJHO SPACE
07DE 23                 INX     H
07DF 4E                 MOV     C,M
07E0 3620               MVI     M,' '
07E2 23                 INX     H
07E3 C5                 PUSH    B
07E4 15                 DCR     D       ;ANY MORE LEFT?
07E5 C2DB07             JNZ     KSET0   ;  YES
07E8 E5                 PUSH    H       ;SAVE LAST ADDRESS
07E9 210000             LXI     H,0     ;SET CURSOR
07EC 22320A             SHLD    CURSAT
07EF 213608             LXI     H,KSETM ;MESSAGE
07F2 0604               MVI     B,KSETN-KSETM ;  AND LENGTH
07F4 4E         KSET1:  MOV     C,M
07F5 CD0002             CALL    ECHOCH
07F8 23                 INX     H
07F9 05                 DCR     B
07FA C2F407             JNZ     KSET1
                    ;ACCEPT NEW PARAMETERS
07FD CD7D05             CALL    INFDX   ;COMMAND
0800 F5                 PUSH    PSW     ;SAVE FOR LATER
0801 4F                 MOV     C,A
0802 CD0002             CALL    ECHOCH  ;ECHO IT
0805 0E20               MVI     C,' '   ;SPAC OVER
0807 CD0002             CALL    ECHOCH
080A CD7D05             CALL    INFDX   ;GET VALUE
080D 4F                 MOV     C,A
080E CD0002             CALL    ECHOCH  ;ECHO IT
0811 F1                 POP     PSW     ;RETRIEVE COMMAND
0812 47                 MOV     B,A
0813 C5                 PUSH    B       ;SAVE ENTIRITY
0814 0E3F               MVI     C,'?'   ;VERIFY CORRECT
0816 CD0002             CALL    ECHOCH
0819 CD7D05             CALL    INFDX
081C E65F               ANI     5FH     ;TAKE CARE OF LC
081E FE59               CPI     'Y'     ;YES???
0820 C1                 POP     B       ;RETRIEVE ARGUMENTS
0821 CC7604             CZ      SETCON  ;DO IT IF OK
                    ;CLEAN UP THE RESULTS
0824 E1                 POP     H       ;RESTORE ADDRESS
0825 1610               MVI     D,10H   ;COUNT
0827 C1         KSET9:  POP     B       ;GET 2 CHARS
0828 2B                 DCX     H
0829 71                 MOV     M,C     ;AND RESTORE THEM
082A 2B                 DCX     H
```

```
082B 70                 MOV     M,B
082C 15                 DCR     D           ;ANY LEFT?
082D C22708             JNZ     KSET9       ;KEEP TRUCKING
0830 E1                 POP     H           ;RESTORE CURSOR
0831 22320A             SHLD    CURSAT
0834 37                 STC                 ;FLAG AS IGNORABLE
0835 C9                 RET                 ;AND DONE
0836 5345542D KSETM:    DB      'SET-'      ;PROMPT
083A          KSETN:    DS      0           ;END OF PROMPT


083A C5       XOFFED:   PUSH    B           ;SAVE SACRED
083B CD0301             CALL    INKBS       ;ANY USER ACTION?
083E B7                 ORA     A
083F C47D05             CNZ     INFDX       ;CHECK IT OUT
0842 3A4E0A             LDA     XOFFD       ;CHECK FLAG
0845 B7                 ORA     A
0846 C1                 POP     B           ;RESTORE SACRED
0847 C9                 RET


                ;       **********************************

                ;
                ;       CONTROL CHARACTER  PROCESSING ROUTINES
                ;

                ;ESCAPE CHARACTER PROCESSING
                ;       TRAP TO ADD PARITY BIT TO NEXT CHARACTER
0848 215708     ESCCHR: LXI     H,ESCTRP ;SET UP TRAP
084B 04         TRPSET: INR     B           ;WHICH ONE?
084C F25308             JP      ESCA0       ;OUTPUT
084F 22C10A             SHLD    IMULJM      ;ECHO
0852 C9                 RET
0853 22BF0A     ESCA0:  SHLD    MULJMP
0856 C9                 RET
                    ;ESCAPE TRAP
0857 79         ESCTRP: MOV     A,C         ;TURN ON PARITY BIT
0858 F680               ORI     80H         ; AND CLEAR CY
085A 4F                 MOV     C,A
085B 210000     TRPCLR: LXI     H,0         ;RESET TRAP VECTOR
085E C34B08             JMP     TRPSET

                ;LOCK KEYBOARD

0861 2A5A0A     LCKKB:  LHLD    MONLVL      ;TRAP ATTEMPT TO ACCESS
0864 22B50A             SHLD    INTRAP
0867 C9                 RET

                ;UNLOCK KEYBOARD

0898                    UNLKKB  SET     CLRTRP

                ;RETURN HEREIS MESSAGE
```

```
0868 217A08    HEREIS: LXI     H,HRISM ;MESSAGE ADDRESS
086B 22B90A            SHLD    HRISP   ;GOES INPOINTER
086E 3E11              MVI     A,HRISN-HRISM ;COUNTER
0870 32BB0A            STA     HRISC
0873 218B08            LXI     H,HERETO ;HEREIS TRAP
0876 22B50A            SHLD    INTRAP
0879 C9                RET

087A 4D617472 HRISM:   DB      'MATROX ALT-2480',CR,LF
087E 6F782041
0882 4C542D32
0886 3438300D
088A 0A
088B          HRISN:   DS      0

088B 2AB90A    HERETO: LHLD    HRISP   ;MESSAGE POINTER
088E 7E                MOV     A,M     ;GET A CHAR
088F 23                INX     H       ;SET FOR NEXT
0890 22B90A            SHLD    HRISP
0893 21BB0A            LXI     H,HRISC ;COUNTER
0896 35                DCR     M       ;ANY LEFT?
0897 C0                RNZ             ;YES
0898 210000    CLRTRP: LXI     H,0     ;COMMON CODE SEGMENT
089B 22B50A            SHLD    INTRAP
089E C9                RET

               ;BACKSPACE

089F 2D        BACKSP: DCR     L       ;BACK ONE
08A0 CDD902    TESTIT: CALL    TSTCUR  ;STAY ON PAGE
08A3 22320A            SHLD    CURSAT
08A6 C9                RET

               ;HORIZONTAL TAB

08A7 7D        HORTAB: MOV     A,L     ;NEXT COL MOD 8
08A8 C608              ADI     08H     ;OVER 8
08AA E6F8              ANI     0F8H    ;AND BACK TO LAST MOD 8
08AC 6F                MOV     L,A
08AD C3A008            JMP     TESTIT

               ;CARRIAGE RETURN AND LINE FEED
               ;WARNING***  LINE FEED MUST FOLLOW IMMEDIATELY

08B0 2E00      CRLF:   MVI     L,0     ;RESET COLUMN

               ;LINE FEED

08B2 24        LNFEED: INR     H       ;DOWN ONE LINE
08B3 C3A008            JMP     TESTIT

               ;VERTICAL TAB

08B6 7C        VERTAB: MOV     A,H     ;NEXT LINE MOD 8
08B7 C608              ADI     08H
08B9 E6F8              ANI     0F8H
08BB 67                MOV     H,A
```

```
08BC C3A008             JMP     TESTIT

                ;FORM FEED

08BF 210000     FORMFD: LXI     H,0000  ;UPPER LEFT
08C2 222C0A             SHLD    CP1C
08C5 22320A             SHLD    CURSAT
08C8 C3F009             JMP     CLEAR   ;CLEAR THE SCREEN TOO

                ;CARRIAGE RETURN

08CB 2E00       CARRET: MVI     L,00    ;COLUMN ZERO
08CD 22320A             SHLD    CURSAT
08D0 C9                 RET

                ;UPLINE

08D1 25         UPLINE: DCR     H       ;UP ONE LINE
08D2 C3A008             JMP     TESTIT

                ;FORESPACE

08D5 2C         FORSPC: INR     L       ;NEXT COLUMN
08D6 C3A008             JMP     TESTIT
                ;DIRECT CURSOR ADDRESSING

08D9 21DF08     DCACOM: LXI     H,DCAY
08DC C34B08             JMP     TRPSET  ;SET TRAP VECTOR
                    ;READ LINE DESIRED
08DF 21FE08     DCAY:   LXI     H,DCAX
08E2 CD4B08             CALL    TRPSET
08E5 79                 MOV     A,C
08E6 D620               SUI     ' '
08E8 FE18               CPI     24D
08EA DAEF08             JC      STATMP  ;OK AS IS
08ED 3E00               MVI     A,0
08EF E5         STATMP: PUSH    H       ;SAVE FOR GP ROUTINE
08F0 21C90A             LXI     H,TEMP  ;ASSUME OUTPUT MODE
08F3 04                 INR     B
08F4 F2FA08             JP      DCAY2
08F7 21B70A             LXI     H,TEMPE ;WRONG
08FA 77         DCAY2:  MOV     M,A     ;STORE IT
08FB E1                 POP     H
08FC 37                 STC             ;INHIBIT FURTHER PROCESSING
08FD C9                 RET
                    ;READ COLUMN AND SET CURSOR
08FE 210000     DCAX:   LXI     H,0
0901 CD4B08             CALL    TRPSET
0904 79                 MOV     A,C
0905 D620               SUI     ' '
0907 21BC0A             LXI     H,WIDTH
090A BE                 CMP     M
090B DA0F09             JC      DCAX1   ;OK AS IS
090E 7E                 MOV     A,M
090F 6F         DCAX1:  MOV     L,A
0910 CD1909             CALL    LDATMP  ;GET STORED ARG
0913 67                 MOV     H,A
```

```
0914 22320A             SHLD    CURSAT
0917 37                 STC
0918 C9                 RET

                ;ROUTINES LDATMP AND STATMP TO LOAD AND
                ;     STORE THE TEMPORARY VARIABLE AS PER ECHO OR OUTPUT.

0919 E5         LDATMP: PUSH    H       ;SAVE H
091A 21C90A             LXI     H,TEMP  ;ASSUME UTPUT MODE
091D 04                 INR     B
091E F22409             JP      LDATP   ;GOOD ASSUMPTION
0921 21B70A             LXI     H,TEMPE
0924 7E         LDATP:  MOV     A,M     ;FETCH IT
0925 E1                 POP     H
0926 C9                 RET

                ;INSERT A STRING OF CHARACTERS

0927 212D09     PUTSTR: LXI     H,PUTSO ;SET UP TRAP
092A C34B08             JMP     TRPSET

092D 3E1F       PUTSO:  MVI     A,' '-1 ;VALID CHARACTER?
092F B9                 CMP     C       ;QUIT IF CONTROL
0930 3F                 CMC             ;RETURN FLAG VALUE
0931 DA5808             JC      TRPCLR  ;END OF INSERT
0934 2A320A             LHLD    CURSAT  ;GOES HERE
0937 C5                 PUSH    B       ;SAVE VITAE
0938 CD1304             CALL    PUTSPC  ;MAKE ROOM
093B C1                 POP     B
093C AF                 XRA     A       ;ALLOW FURTHER PROCESSING
093D C9                 RET

                ;DELETE THE CHARACTER SPECIFIED BY H,L

093E EB         DECHAR: XCHG            ;MAKE ROOM IN HL
093F 4A                 MOV     C,D     ;CHECK LINE FILL
0940 0600               MVI     B,0
0942 21340A             LXI     H,LINFIL
0945 09                 DAD     B
0946 7E                 MOV     A,M     ;NUMBER OF CHARS ON LINE
0947 0C                 INR     C       ;  PLUS ONE
0948 B9                 CMP     C       ;CHAR TO BE DELETED
0949 D8                 RC              ;NOTHING THERE
094A 35                 DCR     M       ;ONE LESS NOW
                    ;FIX UP CURSOR POINTERS
094B 0603               MVI     B,CPNUM ;COUNTER
094D 212D0A             LXI     H,CP1L  ;LINE POINTER
0950 7A         DECH1:  MOV     A,D     ;CHECK LINE
0951 BE                 CMP     M       ;SAME ONE?
0952 C25D09             JNZ     DECH2   ;NO, CAN IGNORE
0955 2B                 DCX     H       ;CHECK COLUMN
0956 7B                 MOV     A,E
0957 BE                 CMP     M       ;TO THE RIGHT?
0958 D25C09             JNC     DECH3   ;  NO, NOT AFFECTED
095B 35                 DCR     M       ;FIX IT UP
095C 23         DECH3:  INX     H       ;BACK TO LINE POINTER
095D 23         DECH2:  INX     H       ;ON TO NEXT
```

```
095E 23                 INX     H
095F 05                 DCR     B       ;ANY LEFT?
0960 C25009             JNZ     DECH1   ;  DO IT
                    ;FINALLY, DELETE THE CHARACTER
0963 EB                 XCHG            ;SET UP FOR MTXAD
0964 E5                 PUSH    H       ;SAVE FOR LATER
0965 CDFA03             CALL    MTXAD   ;CONVERT TO ADDRESS
0968 E3                 XTHL            ;SAVE AND RETRIEVE
0969 3ABC0A             LDA     WIDTH   ;LAST ON LINE
096C 0E00               MVI     C,0     ;HIGH OR LOW RESOLUTION?
096E FE29               CPI     41D     ;ASSUME HIGH
0970 D27509             JNC     DECH6   ;  IT IS
0973 0E80               MVI     C,80H   ;LOW RESOLUTION FLAG
0975 6F         DECH6:  MOV     L,A     ;ADDRESS OF LAST POSITION
0976 2D                 DCR     L       ;CONVERT TO COL NUM
0977 CDFA03             CALL    MTXAD   ; AND THEN TO ADDRESS
097A D1                 POP     D       ;STARTING AT
097B EB                 XCHG
097C 7B                 MOV     A,E     ;FINAL ADDRESS
097D BD         DECH4:  CMP     L       ;DONE YET?
097E CA9809             JZ      DECH5   ;YES
0981 23                 INX     H       ;MOVE A CHAR
0982 0C                 INR     C       ;CHECK FOR LOW RES
0983 F28709             JP      DECH7
0986 23                 INX     H       ;EVERY OTHER ADDRESS
0987 46         DECH7:  MOV     B,M     ;GET A CHAR
0988 2B                 DCX     H       ;BACK UP ONE SPACE
0989 0C                 INR     C       ;CHECK RESOLUTION
098A F28E09             JP      DECH8
098D 2B                 DCX     H       ;LOW
098E 70         DECH8:  MOV     M,B     ;STUFF IT
098F 23                 INX     H       ;SET UP FOR NEXT
0990 0D                 DCR     C       ;CHECK RESOLUTION
0991 F27D09             JP      DECH4   ;HIGH
0994 23                 INX     H
0995 C37D09             JMP     DECH4
0998 3620       DECH5:  MVI     M,' '   ;CLEAR LAST COLUMN
099A C9                 RET

                    ;SET CONFIGURATION SWITCHES

099B 04         CONSET: INR     B       ;ONLY VALID ON OUTPUT
099C F8                 RM
099D 21A409             LXI     H,CONS1 ;SET UP TRAP
09A0 22BF0A             SHLD    MULJMP
09A3 C9                 RET
09A4 21AE09     CONS1:  LXI     H,CONS2 ;SET NEW TRAP
09A7 22BF0A             SHLD    MULJMP
09AA 79                 MOV     A,C     ;SAVE COMMAND
09AB C3EF08             JMP     STATMP  ;SAVE AND RETURN
09AE 210000     CONS2:  LXI     H,0     ;RESET TRAP
09B1 22BF0A             SHLD    MULJMP
09B4 CD1909             CALL    LDATMP  ;RETRIEVE COMMAND
09B7 47                 MOV     B,A
09B8 CD7604             CALL    SETCON
09BB 37                 STC             ;ALL DONE
09BC C9                 RET
```

```
                ;BELL (FLASH SCREEN) ROUTINE

09BD 2A4C0A     BELL:   LHLD    MTXAT
09C0 E5                 PUSH    H
09C1 CDCF09             CALL    BELL0
09C4 1B         BELLK:  DCX     D       ;KILL SOME TIME
09C5 7A                 MOV     A,D
09C6 B3                 ORA     E
09C7 C2C409             JNZ     BELLK
09CA E1                 POP     H
09CB CDCF09             CALL    BELL0
09CE C9                 RET

09CF 11000C     BELL0:  LXI     D,128D*24D      ;TOTAL OF CHARS
09D2 7E         BELL1:  MOV     A,M
09D3 EE80               XRI     80H
09D5 77                 MOV     M,A
09D6 1B                 DCX     D
09D7 23                 INX     H
09D8 7A                 MOV     A,D
09D9 B3                 ORA     E
09DA C2D209             JNZ     BELL1
09DD C9                 RET

                ;FOREGROUND FOLLOWS

09DE AF         OUTFOR: XRA     A
09DF 32BE0A             STA     FORBAK
09E2 C9                 RET

                ;BACKGROUND FOLLOWS

09E3 3E80       OUTBAK: MVI     A,80H
09E5 32BE0A             STA     FORBAK
09E8 C9                 RET

                ;HOME CURSOR

09E9 210000     HOMEIT: LXI     H,0000
09EC 22320A             SHLD    CURSAT
09EF C9                 RET

                ;SUBROUTINE CLEAR
                ;       SET ALL DISPLAY POSITIONS TO BLANK
                ;       RESET LINFIL TABLE TO ZERO
                ;       ALL REGISTERS EXCEPT B & C ARE MODIFIED.

09F0 C5         CLEAR:  PUSH    B
09F1 21340A             LXI     H,LINFIL        ;RESET LINFIL 1ST
09F4 111800             LXI     D,24D   ;BYTE COUNT
09F7 0E00               MVI     C,00    ;STUFF WITH ZEROS
09F9 CD090A             CALL    FILLUP  ;SET THEM ALL
09FC 2A4C0A             LHLD    MTXAT   ;DISPLAY ADDRESS
09FF 11010C             LXI     D,24D*128D+1 ;ERASE BETWEEN LINES TOO
0A02 0E20               MVI     C,' '   ;FILL WITH BLANKS
```

```
0A04 CD090A             CALL    FILLUP
0A07 C1                 POP     B
0A08 C9                 RET

                ;SUBROUTINE FILLUP (C=VALUE, DE=BYTE COUNT, HL=ADDR
                ;       SET DE POINTS TO C STARTING AT H
                ;       RETURNS WITH A=0, DE=0, HL=NEXT ADDRES
                ;       WARNING***  DE = 0 DOES 64K
                ;       A, D, E, FLAGS, H & L MODIFIED

0A09 71         FILLUP: MOV     M,C     ;STUFF ONE
0A0A 23                 INX     H       ;NEXT ADR
0A0B 1B                 DCX     D       ;ONE LESS TO DO
0A0C 7A                 MOV     A,D
0A0D B3                 ORA     E
0A0E C2090A             JNZ     FILLUP  ;ANY LEFT?
0A11 C9                 RET

0A12 04         ANULL:  INR     B       ;ECHO OR PRINT '^0'
0A13 FA210A             JM      ANULE   ;ECHO IT
0A16 0E5E               MVI     C,'^'
0A18 CD0C02             CALL    OUTCHR
0A1B 0E30               MVI     C,'0'
0A1D CD0C02             CALL    OUTCHR
0A20 C9                 RET
0A21 0E5E       ANULE:  MVI     C,'^'
0A23 CD0002             CALL    ECHOCH
0A26 0E30               MVI     C,'0'
0A28 CD0002             CALL    ECHOCH
0A2B C9                 RET


                ;***********************************************
                ;
                ;       VARIABLES AND SWITCHES
                ;
                ;***********************************************

                ;GLOBALS


                ;CHARACTER POINTERS
                ;       THESE POINTERS ARE MAINTAINED TO ALWAYS POINT
                ;         TO THE SAME CHARACTER.  IF THE CHARACTER
                ;         AT THAT POSITION IS DELETED OR OVERWRITTTEN
                ;         THEY POINT TO ITS REPLACEMENT.  IF THE CHARACT
                ;         IS MOVED OFF THE SCREEN, ONE OF ITS COORDINATE
                ;         WILL BE SET TO -1.

0A2C FFFF       CPTRS:  DW      OFFFFH          ;CP1 IS XMIT POINTER
0A2E FFFF               DW      OFFFFH          ;CP2 IS END POINTER
0A2C                    CP1C    EQU     CPTRS ;INIT TO OFF SCREEN
0A2D                    CP1L    EQU     CP1C+1
0A2E                    CP2C    EQU     CPTRS+2
0A2F                    CP2L    EQU     CP2C+1
0A30                    DS      CPNUM*2 - 4 ;EXTRAS
```

```
                ;THE CURRENT CURSOR POSITION
                ;       NOTE: MAY DRIFT PAST EDGES

0A32 0000       CURSAT: DW      0000H   ;START AT UPPER LEFT

0A33                    CURSL   EQU     CURSAT+1  ;LINE NUMBER
0A32                    CURSC   EQU     CURSAT+0  ;COLUMN NUMBER

                ;TABLE OF RIGHTMOST COLUMN IN EACH LINE CONTAINING EXPLI
                ;       (I.E. NOT BLANK FROM CLEAR TYPE FUNCTION)
                ;       INDEXED BY LINE NUMBER

0A34            LINFIL: DS      24D

                ;BASE ADDRESS OF THE 4K MEMORY SPACE USED BY THE 2480

0A4C 00E0       MTXAT:  DW      0E000H


                ;               **********************

                ;INPUT CONTROL SWITCHES

0A4E 00         XOFFD:  DB      0       ;INHIBIT OUTCHR OUTPUT IF -1

0A4F 02         SETC:   DB      STX     ;CONFIGURATION CHANGE CHARACTER

0A50 03         ATTN:   DB      ETX     ;BREAK CHARACTER

0A51 13         XOFF:   DB      DC3     ;INPUT CHAR TO STOP OUTPUT

0A52 11         XON:    DB      DC1     ;INPUT CHAR TO RESUME OUTPUT

0A53 FF         UCLOCK: DB      0FFH    ;-1: NORMAL
                                        ;DFHEX: CONVERT LC TO UC ON INPU

0A54 FF         FDUX:   DB      0FFH    ;00: HALF DUPLEX
                                        ;-1: FULL DUPLEX
                                        ;-2: BLOCK MODE

0A55 DF         CURSOR: DB      '_'+PARON ;CHAR TO USE FOR CURSOR

0A56 00         FIXUP:  DB      0       ;-1 IF PREVIOUS CHAR WAS ESCPE C

0A57 1B         ESCAPE: DB      ESC     ;INPUT ESCAPE CHARATER

0A58 04         BLKEND: DB      EOT     ;INPUT BLOCK TERMINATE CHARACTER

0A59 00         FLASH:  DB      0       ;COUNTER FOR TIMING CURSOR FLASH

0A5A 0000       MONLVL: DW      0000H   ;MONITOR TRAP ADDRESS
                                        ;  DISABLED IF 0000

0A5C 00         LDONE:  DB      0       ;0 = NOT YET
```

```
                                        ;-1=LINE AVAILABLE

0A5D 00     BDONE:  DB      0           ;0 = BLOCK BEING FILLED
                                        ;NON-ZERO = BLOCK COMPLETE

0A5E 610A   LBPTR:  DW      LINBUF      ;LINE BUFFER POINTER

0A60 00     LBCNT:  DB      0           ;LINE BUFFER FILL COUNT

0A61        LINBUF: DS      LINSIZ+2    ;LINE BUFFER

0AB3 D10A   ECONAT: DW      CONTAB      ;ECHO CONTROL TABLE TO USE

0AB5 0000   INTRAP: DW      0000H       ;INPUT TRAP VECTOR

0AB7        TEMPE:  DS      5           ;TEMPORARY VARIABLE SPACE

0ABB            HRISC   EQU     TEMPE+4
0AB9            HRISP   EQU     TEMPE+2

            ;*************************************

            ;               OUTPUT SWITCHES

0ABC 28     WIDTH:  DB      40D         ;COLUMNS PER LINE

0ABD 5F     GREEK:  DB      5FH         ;FF: NORMAL
                                        ;5F: LC DISPLAYS AS UPPER CASE
                                        ;1F: LC DISPLAYED AS GREEK

0ABE 00     FORBAK: DB      0           ;00: NORMAL
                                        ;80: INVERSE VIDIO

0ABF 0000   MULJMP: DW      0           ;0000: NORMAL
                                        ;ADDR: ADDRESS OF ROUTINE TO CAL
                                        ;       BEFORE OUTPUTING.

0AC1 0000   IMULJM: DW      0000H       ;SAME AS MULJMP EXCEPT FOR ECHO

            ;THE FOLLOWING 4 PARAMETERS DETERMINE DISPLAY ACTION
            ;  WHEN THE CURSOR IS MOVED OFF SCREEN

0AC3 18     OFFTOP: DB      24D         ; 1: SCROLL DOWN
                                        ;24: WRAPAROUND TO BOTTOM

0AC4 18     OFFBOT: DB      24D         ; 1: WRAPAROUND TO TOP LINE
                                        ;24: SCROLL UP

0AC5 01     OFFLFT: DB      1           ; 1: OVERWRITE 1ST CHAR ON LINE
                                        ;WIDTH: WRAPAROUND TO END OF SAM
                                        ;1-WIDTH: BACK UP TO PREVIOUS LI

0AC6 00     OFFRT:  DB      0           ; 0: START NEXT LINE
                                        ; 1: WRAP AROUND TO SAME LINE
                                        ;WIDTH: OVERWRITE LAST CHAR ON L
```

```
0AC7 D10A      CONAT:  DW      CONTAB  ;OUTPUT CONTROL CHAR TABLE

0AC9           TEMP:   DS      5       ;TEMPORARY VARIABLE STORAGE


               ;TABLE CONTAB
               ;       CONTROL CHARACTER FUNCTIONS
               ;       FORMAT IS SETS OF THREE BYTES
               ;         BYTE N    = CHARACTER TO RECOGNIZE
               ;         BYTE N+1  = LOW BYTE OF ADDRESS
               ;         BYTE N+2  = HIGH BYTE OF ADDRESS
               ;       ADDRESS IS OF THE ROUTINE TO CALL TO EXECUTE
               ;         THE REQUIRED FUNCTION.
               ;       LAST ENTRY MUST BE NULL (00HEX).

0ACE 0D        CONBLK: DB      CR              ;SPECIAL ECHO FOR BLOCK
0ACF B008              DW      CRLF    ;  MODE

0AD1 0D        CONTAB: DB      CR              ;CARRIAGE RETURN
0AD2 CB08              DW      CARRET
0AD4 0A                DB      LF              ;LINE FEED
0AD5 B208              DW      LNFEED
0AD7 09                DB      HT              ;HORIZONTAL TAB
0AD8 A708              DW      HORTAB
0ADA 8C                DB      FF+PARON        ;FORM FEED
0ADB BF08              DW      FORMFD
0ADD 08                DB      BS              ;BACK SPACE
0ADE 9F08              DW      BACKSP
0AE0 8B                DB      VT+PARON        ;VERTICAL TAB
0AE1 B608              DW      VERTAB
0AE3 1B                DB      ESC             ;ESCAPE
0AE4 4808              DW      ESCCHR
0AE6 07                DB      BEL             ;BELL DING
0AE7 BD09              DW      BELL    ;  FLASHER
0AE9 02                DB      STX             ;SET CONFIG SWITCHES
0AEA 9B09              DW      CONSET
0AEC 0B                DB      VT              ;UPLINE
0AED D108              DW      UPLINE
0AEF 0C                DB      FF              ;FORESPACE
0AF0 D508              DW      FORSPC
0AF2 BD                DB      '='+PARON       ;DIRECT CURSOR ADDRESSIN
0AF3 D908              DW      DCACOM
0AF5 9F                DB      US+PARON        ;FORGROUND FOLLOWS
0AF6 DE09              DW      OUTFOR
0AF8 99                DB      EM+PARON        ;BACKGROUND FOLLOWS
0AF9 E309              DW      OUTBAK
0AFB 1A                DB      XUB             ;CLEAR SCREEN
0AFC F009              DW      CLEAR
0AFE 1E                DB      RS              ;HOME CURSOR
0AFF E909              DW      HOMEIT
0B01 05                DB      ENQ             ;HERE IS MESAGE
0B02 6808              DW      HEREIS
0B04 16                DB      SYN             ;INSERT CHARACTER
0B05 1304              DW      PUTSPC
0B07 18                DB      CAN             ;DELETE CHARACTER
0B08 3E09              DW      DECHAR
0B0A 17                DB      ETB             ;INSERT LINE
```

```
0B0B 8F03                       DW      SCRLDN
0B0D 15                 DB      NAK             ;DELETE LINE
0B0E 4503                       DW      DELINE
0B10 0F                 DB      SI              ;LOCK KEYBOARD
0B11 6108                       DW      LCKKB
0B13 0E                 DB      SO              ;UNLOCK KEYBOARD
0B14 9808                       DW      UNLKKB
0B16 C9                 DB      'I'+FARON       ;STRING INSERT
0B17 2709                       DW      PUTSTR
                    ;END OF LIST MUST BE NUL <<<<<<<<<<<<
0B19 00                 DB      NUL             ;NULL
0B1A 120A                       DW      ANULL


                    ;
                    ;CONTAB FOR HALF DUPLEX ECHOING
                    ;

0B1C 0D         HDCON:  DB      CR      ;CARRIAGE RETURN
0B1D CB08                       DW      CARRET
0B1F 0A                 DB      LF              ;LINE FEED
0B20 B208                       DW      LNFEED
0B22 09                 DB      HT              ;HOR TAB
0B23 A708                       DW      HORTAB
0B25 00                 DB      NUL
0B26 120A                       DW      ANULL


                    ;*****************************************************
                    ;
                    ;       CONTROL CHARACTER DEFINITIONS
                    ;
                    ;*****************************************************

0000                    NUL     EQU     00H     ;NULL
0001                    SOH     EQU     01H     ; ^A
0002                    STX     EQU     02H     ;
0003                    ETX     EQU     03H     ;
0004                    EOT     EQU     04H     ;
0005                    ENQ     EQU     05H     ;
0006                    ACK     EQU     06H     ;
0007                    BEL     EQU     07H     ;
0008                    BS      EQU     08H     ;BACK SPACE
0009                    HT      EQU     09H     ;HORIZONTAL TAB
000A                    LF      EQU     0AH     ;LINE FEED
000B                    VT      EQU     0BH     ;VERTICAL TAB
000C                    FF      EQU     0CH     ;FORM FEED
000D                    CR      EQU     0DH     ;CARRIAGE RE
000E                    SO      EQU     0EH
000F                    SI      EQU     0FH
0010                    DLE     EQU     10H
0011                    DC1     EQU     11H
0012                    DC2     EQU     12H
0013                    DC3     EQU     13H
0014                    DC4     EQU     14H
0015                    NAK     EQU     15H
0016                    SYN     EQU     16H
```

```
0017                    ETB     EQU     17H
0018                    CAN     EQU     18H
0019                    EM      EQU     19H
001A                    XUB     EQU     1AH     ;^X (SUB)
001B                    ESC     EQU     1BH
001C                    FS      EQU     1CH
001D                    GS      EQU     1DH
001E                    RS      EQU     1EH
001F                    US      EQU     1FH
007F                    DEL     EQU     7FH

                ;*****************************************
                ;
                ;       BIT DEFINITIONS
                ;
                ;*****************************************

0080                    PARON   EQU     80H     ;PARITY BIT
0040                    ALPHA   EQU     40H     ;ALPHA CHARACTER BIT
0020                    LC      EQU     20H     ;LOWERCASE = LC + ALPHA
001F                    CTRL    EQU     1FH     ;'X' AND CTRL = ^X

0000                    END
```