

ASSEMBLED 6/9/82

Bob McKen

MACLIB Z80
ORG 0000H

I/O PORTS

```

0000 = PIO1D EQU 00H ;00 = PIO PORT 1. DATA
0001 = PIO1C EQU 01H ;01 = PIO PORT 1. CONTROL
0002 = PIO2D EQU 02H ;02 = PIO PORT 2. DATA
0003 = PIO2C EQU 03H ;03 = PIO PORT 2. CONTROL
0020 = DIP1 EQU 20H ;20-3F= DIPSWITCH 1.
; 40-5F= UART
0040 = UARTDL EQU 40H ; DATA / DIVISOR (LOW)
0041 = UARTDH EQU 41H ; INT. ENABLE / DIVISOR (HIGH)
0042 = UARTID EQU 42H ; INTERRUPT IDENT.
0043 = UARTLC EQU 43H ; LINE CONTROL
0044 = UARTMC EQU 44H ; MODEM CONTROL
0045 = UARTLS EQU 45H ; LINE STATUS
0046 = UARTMS EQU 46H ; MODEM STATUS
; 60-7F= CRTC
0060 = CRTCWA EQU 60H ; WRITE REG. SELECT
0061 = CRTCWD EQU 61H ; WRITE REG. DATA
0063 = CRTCRD EQU 63H ; READ REG. DATA
0064 = CRTCTE EQU 64H ; INTERRUPT ENABLE
0068 = CRTCAUX EQU 68H ; AUX. CHAR. SET
0080 = KEYCON EQU 80H ;80-9F= KEYBOARD ENCODER
00A0 = KEYINT EQU 0A0H ;A0-BF= SPECIAL KEYS & INT. ID.
00C0 = CLICK EQU 0C0H ;C0-DF= CLICK
00E0 = BEEP EQU 0E0H ;E0-FF= BEEP

```

MEMORY LAYOUT

```

; 0 - 3FFFH ;PROM
; 4000H - 7FFFH ;RAM 256 OR 1024 BYTES
; 8000H - BFFFH ;NOT USED
; C000H - FFFFH ;DISPLAY

```

AC7
53D
C7C

```

000D = CR EQU 0DH
000A = LF EQU 0AH
000C = FF EQU 0CH
0009 = TAB EQU 09H
0007 = BEL EQU 07H
0008 = BSPACE EQU 08H
001B = ESC EQU 1BH
;
0400 = RAMSIZE EQU 0400H ;CPU RAM SIZE 1K
0050 = LINELENGTH EQU 0050H ;CHAR./LINE
0018 = NUMBEROFLINES EQU 24 ;NUMBER OF LINES TO DISPLAY
1000 = DMEMSIZE EQU 1000H ;DISPLAY MEMORY SIZE 4K
0780 = SCREENSIZE EQU LINELENGTH * NUMBEROFLINES
;
0087 = INITCONFIG EQU 87H ;POWER UP OPTIONS
; XXXX XXXI CURSOR
; 0=UNDERLINE
; 1=BLOCK

```

```

; XXXX XX1X CLICK
;           0=NO CLICK
;           1=CLICK
; XXXX X1XX END OF LINE
;           0=DISCARD
;           1=WRAP
; XXXX 1XXX AUTO LINEFEED
;           0=NO
;           1=YES
; XXX1 XXXX AUTO CRETURN
;           0=NO
;           1=YES
; XX1X XXXX MODE
;           0=HEATH
;           1=ANSI
; X1XX XXXX KEYPAD
;           0=NORMAL
;           1=SHIFTED
; 1XXX XXXX AUTO REPEAT
;           0=NO
;           1=YES

```

```

;
0018 = KEYDELAY EQU 24 ; INITIAL REPEAT KEY DELAY
0004 = REPRATE EQU 4 ; REPEAT KEY RATE
004E = SCROLLKEY EQU 4EH ; SCROLL KEY COORDINATES
0052 = BREAKKEY EQU 52H ; BREAK KEY COORDINATES
;
0001 = MATRIXBREAK EQU 1 ; BREAK KEY 0=SEPERATE
; 1=MATRIX
0001 = MATRIXLOCAL EQU 1 ; ON/OFF LINE 0=SEPERATE
; 1=MATRIX
0001 = MATRIXSHIFTLOCK EQU 1 ; SHIFT LOCK 0=SEPERATE
; 1=MATRIX
0000 = ENABDISKBOARD EQU 0 ; ENABLE/DISABLE
; KEYBOARD 0= NO.
; 1= YES.
0006 = ENABDISKEY EQU 06H ; ENAB/DIS. KEYBOARD TOGGLE
; KEY COORDINATES
;
0001 = ANSI EQU 1 ; ANSI MODE 0=NO.
; INCLUDED 1=YES.
;
0001 = CPUCLOCK EQU 1 ; CPU CLOCK RATE
; 0=1.536 MHZ
; 1=3.072 MHZ

```

Handwritten annotations: **4F** and **62** are written next to the EQU values 4EH and 52H respectively. A red circle is drawn around the value 1 in the MATRIXSHIFTLOCK line.

```

; *****
;

```

```

0000 C34C07 JMP INIT
MAINLOOP:
0003 210300 LXI H,MAINLOOP ;SET RETURN ADDR.
0006 E5 PUSH H
0007 CD5D0B CALL RBUFEMP ;RECIEVE BUFFER EMPTY?
000A D29902 JNC DISPLAY
000D CDAD0B LB0002: CALL KBUFEMP ;KEY INT. BUFFER EMPTY?
0010 D26D01 JNC KEYBOARD

```

```

0013 CD6A0C      CALL    UPDATECRTC      ;UPDATE CRTC CURSOR & ENABLE CRTC INT.
0016 3AC940      LDA     MODEFLAG2
0019 E608        ANI     08H            ;OFF LINE?
001B C8          RZ              ;YES.-->
ENTRANSINT:
001C F3          DI
001D 3AA340      LDA     TBUF CNT
0020 B7          ORA     A
JRZ     LB0004
0021+280A       DB     28H, LB0004-$-1
0023 DB41        IN      UARTDH
BIT     1,A      ;TRANSMIT INT. ENABLED?
0025+CB4F       DB     0CBH, 1*8+A+40H
JRNZ    LB0004  ;YES.-->
0027+2004       DB     20H, LB0004-$-1
0029 F602       ORI     02H      ;ENABLE TRANSMIT INT.
002B D341        OUT     UARTDH
002D FB          LB0004: EI
002E C9          RET

```

```

;
;*****
;

```

```

0038            ORG     0038H      ;INTERRUPT VECTOR
MSKINT: EXAF
0038+08         DB     08H
EXX
0039+D9         DB     0D9H
003A DBA0        IN      KEYINT
003C EEF6        XRI     0F6H      ;TOGGLE BITS
003E 4F          MOV     C,A

```

```

;
IF     NOT MATRIXLOCAL
LXI    H,MODEFLAG2
ANI    01H      ;OFF LINE KEY?
JRZ    SETOFFLINE
SETB   3,M     ;ON-LINE BIT
JR     CHKBREAK
SETOFFLINE:
RES    3,M     ;RESET FLAG
CHKBREAK:
ENDIF

```

```

;
IF     NOT MATRIXBREAK
;
ANI    04H     ;BREAK KEY?
CNZ    SETBREAKSEP ;YES.-->
ENDIF

```

```

003F+1872       JR     RECEIVEINT
DB     18H, RECEIVEINT-$-1

```

```

;*****
;
NMI VECTORS HERE
;

```

```

0066            ORG     0066H

```

PATCH AREA.

*patch to
MAKE ANSI
WORK*

*0041 ~~7A~~ 7A
0042 2123 0F
0045 C9*

*patch to
clear screen
ON Power-UP*

*0046 CD 1B 08
0049 CD FC 04
004C C9*

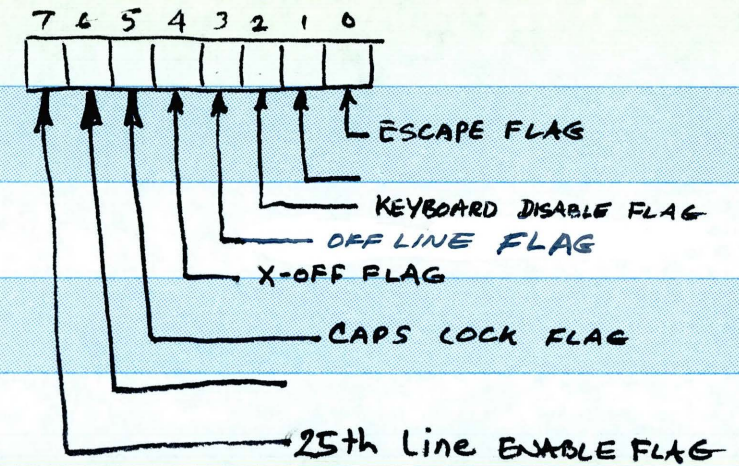
```
;
0066 F5      NMIINT: PUSH   PSW
0067 C5      PUSH   B
0068 D5      PUSH   D
0069 E5      PUSH   H
006A 3E06    MVI     A,06H          ;# ROWS REG.
006C D360    OUT     CRTCWA
006E 3ABE40  LDA     CRTCROWS
0071 D361    OUT     CRTCWD
0073 2ABF40  LHLD   CURSORTYPE
0076 3E0A    MVI     A,0AH          ;CURSOR START SCAN
0078 D360    OUT     CRTCWA
007A 7C      MOV     A,H
007B D361    OUT     CRTCWD
007D 3E0B    MVI     A,0BH          ;CURSOR END SCAN
007F D360    OUT     CRTCWA
0081 7D      MOV     A,L
0082 D361    OUT     CRTCWD
0084 2AC140  LHLD   TOPCRTC
0087 3E0C    MVI     A,0CH          ;TOP OF PAGE (HIGH)
0089 D360    OUT     CRTCWA
008B 7C      MOV     A,H
008C D361    OUT     CRTCWD
008E 3E0D    MVI     A,0DH          ;TOP OF PAGE (LOW)
0090 D360    OUT     CRTCWA
0092 7D      MOV     A,L
0093 D361    OUT     CRTCWD
0095 2AC340  LHLD   CURSORCRTC
0098 3E0E    MVI     A,0EH          ;CURSOR (HIGH)
009A D360    OUT     CRTCWA
009C 7C      MOV     A,H
009D D361    OUT     CRTCWD
009F 3E0F    MVI     A,0FH          ;CURSOR (LOW)
00A1 D360    OUT     CRTCWA
00A3 7D      MOV     A,L
00A4 D361    OUT     CRTCWD
;
00A6 3ACD40  LDA     TIMER
00A9 B7      ORA     A          ;CHECK REPEAT KEY TIMER
00AA C43801  CNZ    TIMEOUT
;
00AD E1      POP     H
00AE D1      POP     D
00AF C1      POP     B
00B0 F1      POP     PSW
RETN
00B1+ED45   DB      0EDH,45H
;
;
;                                     ;END OF NMI ROUTINE.
; *****
;
RECEIVEINT:
00B3 DB42    IN      UARTID
00B5 FE04    CPI     04H          ;RECEIVE DATA AVAILABLE?
                JRNZ   TBUFEMPINT      ;NO.-->
```

```

00B7+202C DB 20H,TBUFEMPINT-$-1
00B9 DB40 IN UARTDL
00BB E67F ANI 7FH
1 JRZ INTRETURN ;= NUL? YES.-->
2 00BD+284F DB 28H,INTRETURN-$-1
3 BIT 3,C ;OFF LINE?
4 00BF+CB59 DB 0CBH,3*8+C+40H
5 JRZ INTRETURN ;YES.-->
6 00C1+284B DB 28H,INTRETURN-$-1
7 00C3 57 MOV D,A
8 00C4 3AA140 LDA RBUFCNT
9 00C7 FE70 CPI 70H ;RECEIVE BUFFER ALMOST FULL?
10 JRC NOXOFF
11 00C9+3812 DB 38H,NOXOFF-$-1
12 00CB DB45 LB0012: IN UARTLS
13 00CD E620 ANI 20H ;TRANSMIT BUFFER EMPTY?
14 JRZ LB0012 ;NO.-->
15 00CF+28FA DB 28H,LB0012-$-1
16 00D1 3E13 MVI A,13H ;DC3 (X-OFF)
17 00D3 D340 OUT UARTDL
18 00D5 3AC940 LDA MODEFLAG2
19 00D8 F610 ORI 10H ;SET X-OFF FLAG
20 00DA 32C940 STA MODEFLAG2
21 00DD 7A NOXOFF: MOV A,D
22 00DE CDD50B CALL RECBUFIN ;PUT BYTE IN RECEIVE BUFFER
23 JRC BEEPRET ;BUFFER FULL -->
24 00E1+3852 DB 38H,BEEPRET-$-1
25 JR INTRETURN
26 00E3+1829 DB 18H,INTRETURN-$-1
27 TBUFEMPINT:
28 00E5 FE02 CPI 02H ;TRANS. BUFF. EMPTY INTERRUPT?
29 JRNZ KBOARDINT ;NO.-->
30 00E7+2015 DB 20H,KBOARDINT-$-1
31 BIT 3,C ;OFF-LINE?
32 00E9+CB59 DB 0CBH,3*8+C+40H
33 JRZ NOTRANS ;YES.-->
34 00EB+2809 DB 28H,NOTRANS-$-1
35 00ED CD800B CALL TBUFGET ;ANYTHING TO TRANSMIT?
36 JRC NOTRANS ;NO.-->
37 00F0+3804 DB 38H,NOTRANS-$-1
38 00F2 D340 OUT UARTDL
39 JR INTRETURN
40 00F4+1818 DB 18H,INTRETURN-$-1
41 NOTRANS:
42 00F6 DB41 IN UARTDH
43 00F8 E6FD ANI 0FDH ;DISABLE TRANS. INTERRUPTS
44 00FA D341 OUT UARTDH
45 JR INTRETURN
46 00FC+1810 DB 18H,INTRETURN-$-1
47 KBOARDINT:
48 00FE 79 MOV A,C
49 00FF E680 ANI 80H ;KEYBOARD INTERRUPT?
50 JRZ INTRETURN ;NO.-->
51 0101+280B DB 28H,INTRETURN-$-1
52 0103 DB80 IN KEYCON
53 0105 47 MOV B,A

```

MODEFLAG2



FE62
0106 FE52
0108 CC6601

see next page

Wrong in EQUATES

```

;
IF MATRIXBREAK
CPI BREAKKEY ;SHOULD WE BREAK?
CZ SETBREAK ;YES.-->
ENDIF
;
IF ENABDISKBOARD
CPI ENABDISKEY ;PROPER KEY?
LDA MODEFLAG2
JRNZ LB0018
XRI 04H ;TOGGLE KEYBOARD DISABLE FLAG
STA MODEFLAG2
JR LB0019
LB0018: ANI 04H ;KEYBOARD DISABLED?
JRNZ INTRETURN ;YES.-->
ENDIF

```

010B CD1201

```

;
LB0019: CALL KEYSAVE
;
INTRETURN:

```

010E+08

```

EXAF
DB 08H

```

010F+D9

```

EXX
DB 0D9H

```

0110 FB

```

EI

```

0111 C9

```

RET

```

;*****

```

;
KEYSAVE:

```

0112 3AC840

```

LDA DIP2SAVE

```

0115 E602

```

ANI 02H ;CLICK ENABLED?

```

0117+2802

```

JRZ LB0020 ;NO.-->

```

0119 D3C0

```

DB 28H, LB0020-$-1
OUT CLICK ;CLICK

```

011B 2AB440

```

LB0020: LHLD KBUFPNT

```

011E 7D

```

MOV A,L

```

011F FEB4

```

CPI LOW(KBUFPNT) ;KEYBOARD BUFFER FULL?

```

0121+2812

```

JRZ BEEPRET ;YES.-->

```

0123 70

```

DB 28H, BEEPRET-$-1

```

0124 23

```

MOV M,B

```

0125 71

```

INX H

```

0126 23

```

MOV M,C

```

0127 22B440

```

INX H
SHLD KBUFPNT

```

012A+ED43

```

SBCD KSAVE ;SAVE FOR REPEAT KEY

```

012C+CE40

```

DB 0EDH, 43H

```

012E 3E18

```

DW KSAVE
MVI A,KEYDELAY ;SET REPEAT KEY TIMER

```

0130 32CD40

```

STA TIMER

```

0133+1802

```

JR KSAVERET
DB 18H, KSAVERET-$-1

```

```

BEEPRET:

```

0135 D3E0

```

OUT BEEP

```

```

KSAVERET:

```

0137 C9

```

RET

```

;*****

; TIMEOUT:

```

0138 3D          DCR      A
0139 32CD40      STA      TIMER
013C C0          RNZ
013D CD5F01      CALL     RESBREAK ;NOT TIME YET.
0140 DBA0        IN       KEYINT ;JUST IN CASE BREAK WAS SET
                BIT      6,A    ;KEYDOWN?
0142+CB77       DB      0CBH,6*8+A+40H
0144 C8          RZ
                BIT      7,A    ;NEW KEY INTERRUPT?
0145+CB7F       DB      0CBH,7*8+A+40H
0147 C8          RZ ;YES.-->
                LBCD     KSAVE  ;GET LAST KEY
0148+ED4B       DB      0EDH,4BH
014A+CE40       DW      KSAVE

```

; IF

```

MATRIXBREAK
014C 78          MOV      A,B ;GET KEY COORDS.
014D FE52        CPI      BREAKKEY ;BREAK KEY?
014F C8          RZ ;YES.-->

```

ENDIF

;

```

0150 3AC840      LDA      DIP2SAVE
                BIT      7,A    ;REPEAT KEY DISABLED?
0153+CB7F       DB      0CBH,7*8+A+40H
0155 C8          RZ ;YES.-->
0156 CD1201      CALL     KEYSAVE
0159 3E04        MVI      A,REPRATE
015B 32CD40      STA      TIMER ;RESET TIMER
015E C9          RET

```

; RESBREAK:

```

015F DB43        IN       UARTLC
0161 E6BF        ANI      0BFH ;RESET BREAK
0163 D343        OUT      UARTLC
0165 C9          RET

```

;

IF NOT MATRIXBREAK

SETBREAKSEP:

```

IN       KEYCON ;CLEAR BREAK INTERRUPT

```

ENDIF

;

SETBREAK:

```

0166 DB43        IN       UARTLC
0168 F640        ORI      40H ;SET BREAK
016A D343        OUT      UARTLC
016C C9          RET

```

;

END OF INTERRUPT ROUTINES.

;

;

TERMINAL KEYBOARD ROUTINES 10/27/80

;

;

;

;

;

;

;

;

;

;

wrong in EQUATES

Fix these & break works OK

62

```

; LOOKUP TABLE FORMAT
;
; MAIN TABLE
; 0XXX XXXX PRINTABLE
; 1XXX XXXX FUNCTION KEY
; 11XX XXXX CHECK KEYPAD SHIFTED FLAG
; 1X1X XXXX CHECK ALT. MODE FLAG
;
; FUNCTION TABLE
; 0XXX XXXX TYPE 0. (PRINTABLE)
; 100X XXXX TYPE 1. (ESC ? OR ESC O)
; 101X XXXX TYPE 2. (ESC OR ESC O)
; 110X XXXX TYPE 3. (ESC OR ESC [)
; 111X XXXX TYPE 4. (SPECIAL ROUTINE LOOKUP)
;
; FUNCTION DATA TABLE
; 1XXX XXXX LAST BYTE OF THIS SEQUENCE
; (MSB IS RESET BEFORE OUTPUT)
;
; **** IMPORTANT ****
; ANY CHANGES TO THE TABLE AREA MUST NOT ALTER
; THE LENGTH. 'HOLES' IN THE TABLE MAY
; BEEN FILLED WITH ROUTINES AND PADDED TO
; MAINTAIN ALIGNMENT.
;
; *****

```

KEYBOARD:

```

016D 7A MOV A,D
016E E67F ANI 7FH
;
; BIT 5,E ;SHIFT?
0170+CB6B DB 0CBH,5*8+E+40H
JRZ NOSHIFT ;NO.-->
0172+2802 DB 28H,NOSHIFT-$-1
0174 F680 ORI 80H

```

NOSHIFT:

```

0176 4F MOV C,A
0177 0600 MVI B,0
0179 21440D LXI H,KTABLE
017C 09 DAD B
017D 7E MOV A,M
017E B7 ORA A
017F CA9B03 JZ BELL ;INVALID KEY.
;PRINTABLE?
;
; BIT 7,A
0182+CB7F DB 0CBH,7*8+A+40H
JRZ OUTPRINT ;YES.-->
0184+2855 DB 28H,OUTPRINT-$-1
;CHECK KEYPAD SHIFT?
;
; BIT 6,A
0186+CB77 DB 0CBH,6*8+A+40H
JRZ PADOK ;NO.-->
0188+280F DB 28H,PADOK-$-1
018A 3AC840 LDA DIP2SAVE
018D E640 ANI 40H ;KEYPAD SHIFTED?
;NO.-->
;
; BIT 4,A
018F+2808 DB 28H,PADOK-$-1

```


0191 79	MOV	A,C	
0192 EE80	XRI	80H	;TOGGLE SHIFT BIT
0194 4F	MOV	C,A	
0195 21440D	LXI	H,KTABLE	; KTABLE BASE ADDRESS KEY LOOKUP TABLE BASE ADDRESS
0198 09	DAD	B	
0199 7E	MOV	A,M	;GET FUNCTION CODE
019A 21440E	LXI	H,FUNTAB	
	BIT	7,C	;SHIFT?
019D+CB79	DB	0CBH,7*8+C+40H	
	JRZ	NOTSHIFT	;NO.-->
019F+2803	DB	28H,NOTSHIFT-\$-1	
01A1 21610E	LXI	H,FUNSHIFTTAB	
	NOTSHIFT:		
	BIT	5,A	;CHECK ALT. MODE?
01A4+CB6F	DB	0CBH,5*8+A+40H	
	JRZ	GETFUNCT	;NO.-->
01A6+280C	DB	28H,GETFUNCT-\$-1	
01A8 4F	MOV	C,A	;SAVE IN C
01A9 3AC740	LDA	MODEFLAG1	
01AC E604	ANI	04H	;ALT. MODE?
01AE 79	MOV	A,C	;NEED CODE IN A
	JRZ	GETFUNCT	;NO.-->
01AF+2803	DB	28H,GETFUNCT-\$-1	
01B1 21710E	LXI	H,ALTTAB	
	GETFUNCT:		
01B4 E61F	ANI	1FH	;MAX. TABLE SIZE
01B6 4F	MOV	C,A	
01B7 09	DAD	B	
01B8 7E	MOV	A,M	;GET BYTE FROM TABLE
01B9 B7	ORA	A	;VALID CODE?
01BA CA9B03	JZ	BELL	;NO.-->
	BIT	7,A	;PRINTABLE?
01BD+CB7F	DB	0CBH,7*8+A+40H	
	JRZ	OUTA	;YES.-->
01BF+283E	DB	28H,OUTA-\$-1	
01C1 E61F	ANI	1FH	;MAX. FUNCTION #
01C3 4F	MOV	C,A	;SAVE FUNCTION #
01C4 7E	MOV	A,M	
01C5 E660	ANI	60H	;MASK ALL BUT TYPE BITS
	JRZ	TYPE1	
01C7+2840	DB	28H,TYPE1-\$-1	
01C9 FE20	CPI	20H	
	JRZ	TYPE2	
01CB+284C	DB	28H,TYPE2-\$-1	
01CD FE40	CPI	40H	
	JRZ	TYPE3	
01CF+2854	DB	28H,TYPE3-\$-1	
	TYPE4:		
01D1 21980E	LXI	H,TYPE4TAB	;TABLE OF ROUTINE ADDR.
01D4 09	DAD	B	
01D5 09	DAD	B	;WORD VALUE
01D6 7E	MOV	A,M	
01D7 23	INX	H	
01D8 66	MOV	H,M	
01D9 6F	MOV	L,A	
01DA E9	PCHL		;GO TO IT.

```

;
;
OUTPRINT:
01DB 47      MOV      B,A          ;SAVE KEY
;
;
01DC 3AC940  LDA      MODEFLAG2
01DF E620    ANI      20H          ;CAPS. LOCK FLAG BIT?
ELSE
01E1 78      MOV      A,B
JRZ         OUTP1          ;NO.-->
01E2+2813   DB      28H,OUTP1-$-1
01E4 FE41    CPI      41H
JRC         OUTP1
01E6+380F   DB      38H,OUTP1-$-1
01E8 FE5B    CPI      5BH
01EA FAF501  JM       FLIPIT
01ED FE61    CPI      61H
JRC         OUTP1
01EF+3806   DB      38H,OUTP1-$-1
01F1 FE7B    CPI      7BH
JRNC        OUTP1
01F3+3002   DB      30H,OUTP1-$-1
01F5 EE20    FLIPIT: XRI     20H          ;SWAP UPPER & LOWER CASE
OUTP1:      BIT      4,E          ;CONTROL KEY?
01F7+CB63   DB      0CBH,4*8+E+40H
JRZ         ASIS          ;NO.-->
01F9+2802   DB      28H,ASIS-$-1
01FB E61F    ANI      1FH
ASIS:       JR       OUTIT
01FD+1807   DB      18H,OUTIT-$-1
;
OUTA:
01FF E67F    ANI      7FH
BIT      4,E          ;CONTROL KEY?
0201+CB63   DB      0CBH,4*8+E+40H
0203 C2D50B  JNZ      RECBUFIN        ;YES.--> LOCAL ONLY.
0206 C32B0C  OUTIT:   JMP      OUTPUTA    ;ON/OFF LINE.
;FULL/HALF DUPLEX.
;
;
;
0209 217D0E  TYPE1:   LXI     H,TYPE1TAB
020C 09      DAD      B
020D E5      OUTPUT1: PUSH   H          ;THIS OUTPUTS ESC ? OR ESC O
020E CD3E02  CALL    ESCOUT
0211 3E3F    MVI     A,'?'
JRZ         TYPE3LOOP      ;HEATH
0213+281C   DB      28H,TYPE3LOOP-$-1
0215 3E4F    OUT1A:  MVI     A,'O'
JR         TYPE3LOOP      ;ANSI
0217+1818   DB      18H,TYPE3LOOP-$-1
;

```

01DC 00 00 00

01DF CB4B

To change from Matrix shift lock to separate shift lock key installed 7/9/82

```

;
0219 21890E      TYPE2: LXI      H,TYPE2TAB
021C 09          DAD        B
021D E5          OUTPUT2:PUSH H                ;THIS OUTPUTS ESC OR ESC O
021E CD3E02      CALL      ESCOUT
                JRZ        HEATH3              ;HEATH
0221+2811        DB        28H,HEATH3-$$-1
                JR         OUT1A              ;ANSI
0223+18F0        DB        18H,OUT1A-$$-1

```

```

;
0225 21910E      TYPE3: LXI      H,TYPE3TAB      ;ESC OR ESC [
0228 09          DAD        B
0229 E5          OUTPUT3:PUSH H
022A CD3E02      CALL      ESCOUT
022D 3E5B        MVI        A,['[
                JRZ        HEATH3              ;HEATH
022F+2803        DB        28H,HEATH3-$$-1
                TYPE3LOOP:                    ;LOOP UNTIL MSB = 1.
0231 CDFE01      CALL      OUTA
0234 E1          HEATH3: POP     H
0235 7E          MOV        A,M
                BIT        7,A                ;LAST ONE?
0236+CB7F        DB        0CBH,7*8+A+40H
                JRNZ      OUTA                ;YES.-->
0238+20C5        DB        20H,OUTA-$$-1
023A 23          INX        H                ;NEXT.
023B E5          PUSH     H
023C+18F3        JR         TYPE3LOOP         ;CONTINUE
                DB        18H,TYPE3LOOP-$$-1

```

```

;
023E 3E1B        ESCOUT:                    ;OUTPUT ESC & CHECK MODE
0240 CDFE01      MVI        A,ESC
                CALL      OUTA
0243 3AC840      MODETEST:
0246 E620        LDA        DIP2SAVE
0248 C9          ANI        20H              ;HEATH MODE?
                RET

```

```

;
;*****
;
; TYPE 4 ROUTINES START HERE.
;

```

```

INSERTKEY:
0249 21A40E      LXI        H,INSERTAB
024C 0E00        MVI        C,0
024E 3AC740      LDA        MODEFLAG1
0251 E640        ANI        40H              ;INSERT ON?
                JRZ        TURNON            ;NO.-->
0253+2803        DB        28H,TURNON-$$-1
0255 0C          INR        C
0256 0C          ADDTWO: INR     C
0257 0C          INR        C
0258 3AC840      TURNON: LDA     DIP2SAVE
025B E620        ANI        20H              ;HEATH?
                JRZ        NOTANSI          ;YES.-->
025D+2801        DB        28H,NOTANSI-$$-1

```

```

025F 0C          INR    C
NOTANSI:
0260 09          DAD    B
                JR     OUTPUT3 ;HL => STRING.
0261+18C6        DB     18H,OUTPUT3-$-1
;
; SCROLKEY:
0263 21CB40      LXI    H,HOLDCNT ;LF COUNTER
                BIT    5,E ;SHIFT?
0266+CB6B        DB     0CBH,5*8+E+40H
                JRZ   ONELF ;NO.-->
0268+2802        DB     28H,ONELF-$-1
026A 3617        MVI    M,17H ;23 LF'S
026C 34          ONELF: INR    M
                BREAKRET: ;BREAK HANDLED IN INTERRUPT
                                ;ROUTINE, BUT VALID FUNCTION
                                ;NEEDED TO AVOID BEEP.
026D C9          RET
;
; ERASEKEY:
026E 21AA0E      LXI    H,ERATAB
0271 48          MOV    C,B ;ZERO INTO C
                BIT    5,E ;SHIFT?
0272+CB6B        DB     0CBH,5*8+E+40H
                JRZ   TURNON ;NO.-->
0274+28E2        DB     28H,TURNON-$-1
                JR     ADDTWO
0276+18DE        DB     18H,ADDTWO-$-1
;
; DELCHARKEY:
0278 21AF0E      LXA    H,DELTAB
027B 48          MOV    C,B ;ZERO INTO C
                JR     TURNON
027C+18DA        DB     18H,TURNON-$-1
;
;
; IF MATRIXLOCAL OR MATRIXSHIFTLOCK
OFFLINE:
027E 21C940      LXI    H,MODEFLAG2 ;ON/OFF LINE,ON/OFF CAPS. LOCK
                BIT    5,E ; DUAL FUNCTION
                ;SHIFT?
0281+CB6B        DB     0CBH,5*8+E+40H
                JRZ   LOC1 ;NO.-->
0283+280A        DB     28H,LOC1-$-1
                BIT    4,E ;CONTROL?
0285+CB63        DB     0CBH,4*8+E+40H
                JRZ   LOC2 ;NO.-->
0287+2803        DB     28H,LOC2-$-1
                SETB  5,M ;SET CAPS. LOCK FLAG
0289+CBEE        DB     0CBH,5*8+M+0C0H
028B C9          RET
;
; LOC2:
028C+CBDE        SETB  3,M ;ON-LINE
028E C9          DB     0CBH,3*8+M+0C0H
                RET
;
; LOC1:
                BIT    4,E ;CONTROL?

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

028F+CB63      DB      0CBH,4*8+E+40H
                JRZ     LOC3      ;NO.-->
0291+2803      DB      28H,LOC3-$-1
                RES     5,M      ;RESET CAPS. LOCK FLAG
0293+CBAE      DB      0CBH,5*8+M+80H
0295 C9        RET
                ;
                LOC3:  RES     3,M      ;OFF-LINE
0296+CB9E      DB      0CBH,3*8+M+80H
0298 C9        RET
                ENDIF
                ;
                ;
                ;*****
                ;
                DISPLAY: ;PROCESS KEYS FROM RECEIVE BUFFER
0299 57        MOV     D,A
029A 3AC940     LDA     MODEFLAG2
                BIT     4,A      ;DID WE SEND X-OFF?
029D+CB67      DB      0CBH,4*8+A+40H
                JRZ     LB0062     ;NO.-->
029F+2813      DB      28H,LB0062-$-1
02A1 5F        MOV     E,A
02A2 3AA140     LDA     RBUFCNT
02A5 FE60      CPI     60H      ;TIME TO SEND X-ON?
02A7 7B        MOV     A,E
                JRNC    LB0062     ;NO.-->
02A8+300A      DB      30H,LB0062-$-1
02AA E6EF      ANI     0EFH
02AC 32C940     STA     MODEFLAG2   ;RESET X-OFF FLAG
02AF 5F        MOV     E,A
02B0 CD3103     CALL    OUTPUTXON   ;DC1 (X-ON)
02B3 7B        MOV     A,E
                LB0062: BIT     0,A      ;WAS PREVIOUS BYTE ESC?
02B4+CB47      DB      0CBH,0*8+A+40H
02B6 7A        MOV     A,D
                JRZ     LB0065     ;NO.-->
02B7+2824      DB      28H,LB0065-$-1
02B9 3AC940     LDA     MODEFLAG2
02BC E6FE      ANI     0FEH
02BE 32C940     STA     MODEFLAG2   yes; RESET ESC FLAG (turnoff 0 bit.)
                ;
                IF     ANSI
02C1 CD4302     CALL    MODETEST   ;HEATH MODE?
                JRZ     LB0063     ;YES.-->
02C4+2809      DB      28H,LB0063-$-1
                ;
                ANSI LOOKUP
02C6 21230F     LXI     H,ANSITAB
02C9 1604      MVI     D,04H      ;# OF ENTRIES
02CB 1E03      MVI     E,03H      ;TABLE INCREMENT
02CD+1808      JR      LB0064
                DB      18H,LB0064-$-1
                ENDIF
                ;
                ;
                HEATH LOOKUP
02CF 7A        LB0063: MOV     A,D

```

yes, keep on going ↓

it seems to me mov A, D is needed in here!

02C6 CD4100 patched to do this (works) 7/14/82

```

02D0 21B10E      LXI      H,HEATHTAB
02D3 162C        MVI      D,2CH          ;# OF ENTRYS
02D5 1E03        MVI      E,03H        ;TABLE INCREMENT
02D7 CD580C      LB0064: CALL     LOOKUP
02DA+306D        JRNCR   LB0073        jump NO carry
02DC C9          DB      30H,LB0073-$-1
02DD FE0A        RET          ;NOT FOUND.-->
;
;
02DD FE0A        LB0065: CPI      LF          COMMON DISPLAY
02DF+2055        JRNZ     LB0071        ;LINEFEED?
02E1 3AC740      DB      20H,LB0071-$-1 ;NO.-->
02E4 E601        LDA      MODEFLAG1
02E6+2829        ANI      01H          ;HOLD SCREEN MODE?
02E8 3ABB40      JRZ      HOLD1        ;NO.-->
02EB FE17        DB      28H,HOLD1-$-1
02ED+2022        LDA      ROW
02EF 21CB40      CPI      17H          ;LAST LINE?
02F2 35          JRNZ     HOLD1        ;NO.-->
02F3+201C        DB      20H,HOLD1-$-1
02F5 3E13        LXI      H,HOLD1CNT
02F7 CD2B0C      DCR      M            ;DISPLAY NEXT LINE?
02FA CD1C00      JRNZ     HOLD1        ;YES.-->
02FD CDAD0B      MVI      A,13H        ;DC3 (X-OFF)
0300+3012        CALL    OUTPUTA
0302 CD6A0C      CALL    ENTRANSINT
0305 3AA140      LB0066: CALL    KBUFEMP ;ANY KEYS?
0308 FE80        JRNCR   LB0068        ;YES.-->
030A+20F1        DB      30H,LB0068-$-1
030C 3E01        CALL    UPDATECRTC
030E 32CB40      LDA      RBUFCNT
0311 C3D803      CPI      80H          ;RECEIVE BUFFER FULL?
0314 7A          JRNZ     LB0066        ;NO.-->
0315 E67F        DB      20H,LB0066-$-1
0317 FE4E        MVI      A,01H
0319+2808        STA     HOLD1CNT     ;DISPLAY A LINE
031B CD6D01      JMP     LINEFEED
031E CD1C00      ;
0321+18E2        ;
0323+CB6B        LB0068: MOV     A,D
0325 3E01        ANI      7FH
0327+2802        CPI      SCROLLKEY   WRONG IN EQUATES - WORKS WHEN FIXED
0329+2808        JRZ     LB0069        ;YES.-->
032B CD6D01      DB      28H,LB0069-$-1
032D CD1C00      CALL    KEYBOARD
032F+2802        CALL    ENTRANSINT
0331+18E2        JR      LB0067
0333+18E2        DB      18H,LB0067-$-1
;
;
;
0323+CB6B        ;
0325 3E01        ;
0327+2802        ;
0329+2802        ;
032B CD6D01      ;
032D CD1C00      ;
032F+2802        ;
0331+18E2        ;
0333+18E2        ;
0335+18E2        ;
0337+18E2        ;
0339+18E2        ;
033B CD6D01      ;
033D CD1C00      ;
033F+2802        ;
0341+18E2        ;
0343+18E2        ;
0345+18E2        ;
0347+18E2        ;
0349+18E2        ;
034B CD6D01      ;
034D CD1C00      ;
034F+2802        ;
0351+18E2        ;
0353+18E2        ;
0355+18E2        ;
0357+18E2        ;
0359+18E2        ;
035B CD6D01      ;
035D CD1C00      ;
035F+2802        ;
0361+18E2        ;
0363+18E2        ;
0365+18E2        ;
0367+18E2        ;
0369+18E2        ;
036B CD6D01      ;
036D CD1C00      ;
036F+2802        ;
0371+18E2        ;
0373+18E2        ;
0375+18E2        ;
0377+18E2        ;
0379+18E2        ;
037B CD6D01      ;
037D CD1C00      ;
037F+2802        ;
0381+18E2        ;
0383+18E2        ;
0385+18E2        ;
0387+18E2        ;
0389+18E2        ;
038B CD6D01      ;
038D CD1C00      ;
038F+2802        ;
0391+18E2        ;
0393+18E2        ;
0395+18E2        ;
0397+18E2        ;
0399+18E2        ;
039B CD6D01      ;
039D CD1C00      ;
039F+2802        ;
03A1+18E2        ;
03A3+18E2        ;
03A5+18E2        ;
03A7+18E2        ;
03A9+18E2        ;
03AB CD6D01      ;
03AD CD1C00      ;
03AF+2802        ;
03B1+18E2        ;
03B3+18E2        ;
03B5+18E2        ;
03B7+18E2        ;
03B9+18E2        ;
03BB CD6D01      ;
03BD CD1C00      ;
03BF+2802        ;
03C1+18E2        ;
03C3+18E2        ;
03C5+18E2        ;
03C7+18E2        ;
03C9+18E2        ;
03CB CD6D01      ;
03CD CD1C00      ;
03CF+2802        ;
03D1+18E2        ;
03D3+18E2        ;
03D5+18E2        ;
03D7+18E2        ;
03D9+18E2        ;
03DB CD6D01      ;
03DD CD1C00      ;
03DF+2802        ;
03E1+18E2        ;
03E3+18E2        ;
03E5+18E2        ;
03E7+18E2        ;
03E9+18E2        ;
03EB CD6D01      ;
03ED CD1C00      ;
03EF+2802        ;
03F1+18E2        ;
03F3+18E2        ;
03F5+18E2        ;
03F7+18E2        ;
03F9+18E2        ;
03FB CD6D01      ;
03FD CD1C00      ;
03FF+2802        ;

```

AF

WRONG IN EQUATES - WORKS WHEN FIXED

WORKS OK!

new page

```

0329 3E18      MVI      A,18H
032B 32CB40    LB0070: STA      HOLDCNT
032E CDD803    CALL     LINEFEED
                OUTPUTXON:
0331 3E11      MVI      A,11H      ;DC1 (X-ON)
0333 C32B0C    JMP      OUTPUTA
;
0336 FE7F      LB0071: CPI      7FH
                JRZ      LB0072
0338+2804     DB      28H, LB0072--$-1
033A FE20      CPI      20H
                JRNC     LB0074
033C+3016     DB      30H, LB0074--$-1
033E 21800F    LB0072: LXI      H,COMMONTAB ;LOOKUP STANDARD CONTROL CODES
0341 1607      MVI      D,07H      ;# OF ENTRIES
0343 1E03      MVI      E,03H      ;TABLE INCREMENT
0345 CD580C    CALL     LOOKUP
0348 D8        RC          ;NOT FOUND.-->
0349 5F        LB0073: MOV      E,A
034A 23        INX      H
034B 56        MOV      D,M
034C EB        XCHG
034D 01C840    LXI      B,DIP2SAVE
0350 11C740    LXI      D,MODEFLAG1
0353 E9        PCHL
;
0354 3AC740    LB0074: LDA      MODEFLAG1
0357 47        MOV      B,A      ;SAVE IN B
0358 E602     ANI      02H      ;GRAPHICS?
035A 7A        MOV      A,D
                JRZ      LB0076 ;NO.-->
035B+280D     DB      28H, LB0076--$-1
035D FE5E     CPI      5EH      ;SPECIAL GRAPHIC CHAR.
035F DA6A03    JC      LB0076
                JRNZ     LB0075
0362+2004     DB      20H, LB0075--$-1
0364 3E7F     MVI      A,7FH
                JR      LB0076
0366+1802     DB      18H, LB0076--$-1
;
0368 E61F     LB0075: ANI      1FH
036A 57     LB0076: MOV      D,A
036B 78     MOV      A,B      ;GET MODEFLAG1
036C E680    ANI      80H      ;REVERSE VIDEO?
036E B2     ORA      D
036F F5     PUSH     PSW
0370 78     MOV      A,B      ;GET MODEFLAG1
0371 E640    ANI      40H      ;INSERT MODE?
0373 C4310C  CNZ      CINSERTSCROLL ;LINE SCROLL
0376 F1     POP      PSW
0377 2ABC40  LHLD     CURSORABS
037A 77     MOV      M,A
037B 23     INX      H
037C 7C     MOV      A,H
037D F6F8   ORI      0F8H
037F 67     MOV      H,A

```

(They're in reverse order in mem)

*dip2save into BC
modeflag1 into DE
goto function*

→ E6

0F8H

FD

- DMEM SIZE

```

0380 3ABA40 LDA COLUMN
0383 FE4F CPI 4FH
JRNZ LB0077
0385+200C DB 20H, LB0077-$-1
0387 3AC840 LDA DIP2SAVE
038A E604 ANI 04H ;WRAP?
038C C8 RZ ;NO.-->
038D CD8F04 CALL DOLFEEED 4BF
0390 C3EC03 JMP DOCRETURN

```

```

;
0393 3C LB0077: INR A
0394 32BA40 STA COLUMN
0397 22BC40 SHLD CURSORABS
039A C9 RET

```

```

;
;*****
;
; COMMON ROUTINES START HERE.
;

```

```

039B D3E0 BELL: OUT BEEP
039D C9 RET

```

```

; BACKSPACE:

```

```

039E 3ABA40 LDA COLUMN
03A1 B7 ORA A ;BEGINNING OF LINE?
03A2 C8 RZ ;YES.-->
03A3 3D DCR A
03A4 32BA40 STA COLUMN
03A7 2ABC40 LHLD CURSORABS
03AA 2B DCX H

```

(can't backspace from column 0)

```

; LAB2:

```

```

03AB 7C MOV A, H
03AC F6F8 ORI 0F8H
03AE 57 MOV H, A
03AF 22BC40 SHLD CURSORABS ;UPDATE CURSOR ABS.
03B2 C9 RET

```

```

; HORIZTAB:

```

```

03B3 3ABA40 LDA COLUMN ;HORIZ TAB (09H)
03B6 C608 ADI 08H
03B8 E6F8 ANI 0F8H
03BA FE50 CPI 50H
JRNZ LB0216
03BC+200E DB 20H, LB0216-$-1
03BE 3ABA40 LDA COLUMN
03C1 FE4F CPI 4FH
JRZ LB0215
03C3+2803 DB 28H, LB0215-$-1
03C5 3C INR A
JR LB0216
03C6+1804 DB 18H, LB0216-$-1
03C8 3E4F LB0215: MVI A, 4FH
JR LB0216
03CA+1800 DB 18H, LB0216-$-1
03CC 32BA40 LB0216: STA COLUMN
03CF 5F MOV E, A

```

NOT DONE




```

03D0 1600      MVI    D,00H
03D2 2AB840    LHL    BEGLINABS
03D5 19        LAB5:  DAD    D
                JR     LAB2
03D6+18D3      DB     18H,LAB2-$-1
                ;
                LINEFEED:
03D8 CD8F04    CALL   DOLFEED
03DB 3AC840    LDA    DIP2SAVE
03DE E610      ANI    10H      ;AUTO CR?
03E0 C2EC03    JNZ    DOCRETURN  ;YES.-->
03E3 C9        RET

                ;
                CRETURN:
03E4 3AC840    LDA    DIP2SAVE
03E7 E608      ANI    08H      ;AUTO LF?
03E9 C48F04    CNZ    DOLFEED  ;YES.-->
                DOCRETURN:
03EC 3E00      MVI    A,00H
03EE 32BA40    STA    COLUMN
03F1 2AB840    LHL    BEGLINABS
03F4 22BC40    SHLD   CURSORABS
03F7 C9        RET

                ;
                SETESCFLAG:
03F8 3AC940    LDA    MODEFLAG2  ;SET ESC FLAG
03FB F601      ORI    01H
03FD 32C940    STA    MODEFLAG2
0400 C9        RET

                ;
                *****
                ;
                HEATH ROUTINES START HERE.
                ;
                IF     ANSI
                ENTERANSI:      ;ENTER ANSI MODE.
0401 0A        LDAX   B
0402 F620      ORI    20H
0404 02        STAX   B
0405 C9        RET

                ;
                ANSITRANS25:
0406 78        MOV    A,B      ;TRANSMIT 25TH LINE
0407 B7        ORA    A      ;ANSI ESC [ Q
0408 C0        RNZ

                ENDIF
                ;
                TRANS25TH:      ;TRANSMIT 25TH LINE
0409 3AC940    LDA    MODEFLAG2
040C E680      ANI    80H      is top bit set?
                JRZ    LB0227
040E+2811     DB     28H,LB0227-$-1
0410 2AB640    LHL    TOPABS
0413 118007    LXI    D,SCREENSIZE
0416 19        DAD    D
0417 7C        MOV    A,H

```

is top bit set?

780H

OK. WORKS



Fφ?

ex DE,HL

```

0418 F6F8      ORI      0F8H
041A 67        MOV      H,A
041B EB        XCHG
041C 0600      MVI      B,00H
041E CD970C    CALL     LINETOCOMM
0421 3E0D      MVI      A,CR      ;OUTPUT A CR
0423 CD8C0C    CALL     BYTETOCOMM
0426 D3E0      OUT      BEEP
0428 C9        RET

;
IF      ANSI
ANSITRANS PAGE:
0429 78        MOV      A,B      ;TRANSMIT PAGE
042A B7        ORA      A      ;ANSI ESC [ P
042B C0        RNZ

ENDIF

;
TRANSPAGE:      ;HEATH TRANSMIT PAGE
042C 2AB640    LHLD     TOPABS
042F EB        XCHG
0430 0600      MVI      B,00H
0432 2618      MVI      H,18H
0434 E5        LB0243: PUSH   H
0435 CD970C    CALL     LINETOCOMM
0438 E1        POP      H
0439 25        DCR      H
043A+20F8     JRNZ     LB0243
043B          DB      20H,LB0243-$-1
043C+18E3     JR      LB0227      ;OUTPUT CR THEN BEEP
043D          DB      18H,LB0227-$-1

;
INSERTCHARON:  ;ENTER INSERT CHARACTER MODE
043E EB        XCHG
043F+CBF6     SETB     6,M
0441 C9        DB      0CBH,6*8+M+0C0H
0442          RET

;
CURSUP:  LDA      ROW
0445 B7        ORA      A      ;TOP LINE?
0446 C8        RZ      ;YES.-->

REVLFEED:
0447 3ABB40    LDA      ROW
044A FE18     CPI      18H      ;25TH LINE?
044C C8        RZ      ;YES.-->
044D 11B0FF    LXI      D,-LINELENGTH
0450 CD7D04    CALL     LAB3
0453 3ABB40    LDA      ROW
0456 B7        ORA      A
0457+201E     JRNZ     LB0125
0458          DB      20H,LB0125-$-1
0459 3AC940    LDA      MODEFLAG2
045C E680     ANI      80H      ;25TH LINE ENABLED?
045D          JRZ     LB0124      ;NO.-->
045E+280F     DB      28H,LB0124-$-1

;
0460 2AB640    LHLD     TOPABS      ;MOVE 25TH LINE DATA TO NEW

```

OK

HL -> SCREEN BASE ADD.

```

0463 113007      LXI      D,SCREENSIZE-LINELENGTH      ;LOCATION.
0466 19          DAD      D
0467 EB         XCHG
0468 215000      LXI      H,LINELENGTH
046B 19          DAD      D
046C CDCD0A     CALL     SCROLL80
;
046F 2AB640     LB0124: LHLD   TOPABS      ;BACKUP ONE LINE
0472 11B0FF     LXI      D,-LINELENGTH
;
0475+1847      JR      LAB4
DB      18H,LAB4-$-1
;
0477 3D         LB0125: DCR    A
0478 32BB40     STA    ROW
;
047B+1856      JR      LB0117
DB      18H,LB0117-$-1
;
047D 2AB840     LAB3:  LHLD   BEGLINABS
0480 19          DAD      D
0481 7C         MOV     A,H
0482 F6F8      ORI     0F8H      ;DISPLAY MEM. ABS.
0484 67         MOV     H,A
0485 22B840     SHLD   BEGLINABS
0488 C9         RET
;
CURSDOWN:      ;CURSOR DOWN
0489 3ABB40     LDA    ROW
048C FE17      CPI    17H      ;LAST LINE?
048E D0        RNC     ;YES.-->
DOLFEED:
048F 3ABB40     LDA    ROW
0492 FE18      CPI    18H      ;25TH LINE?
0494 C8        RZ     ;YES.-->
0495 115000     LXI    D,LINELENGTH
0498 CD7D04     CALL   LAB3
049B 3ABB40     LDA    ROW
049E FE17      CPI    17H      ;LAST LINE?
;NO.-->
04A0+202D     DB     20H,LB0116-$-1
04A2 3AC940     LDA    MODEFLAG2
04A5 E680      ANI    80H      ;25TH LINE ENABLED?
;NO.-->
04A7+280C     DB     28H,LB0113-$-1
;
04A9 E5        PUSH   H      ;MOVE 25TH LINE DATA TO NEW
04AA 115000     LXI    D,LINELENGTH ; LOCATION.
04AD 19          DAD    D
04AE EB         XCHG
04AF E1        POP    H
04B0 E5        PUSH   H
04B1 CDCD0A     CALL   SCROLL80
04B4 E1        POP    H
;
04B5 CDC80A     LB0113: CALL   SPACE80      ;CLEAR ONE LINE AHEAD
04B8 2AB640     LHLD   TOPABS
04BB 115000     LXI    D,LINELENGTH

```

F0?

something missing here? or just two unnecessary inst.?

maybe not! reloads HL and stores them stack.

F0?

shouldn't this be 0FH?

7/20/80



```

; LAB4:
04BE 19      D
04BF 7C      MOV     A,H
04C0 F6F8    ORI     0F8H
04C2 67      MOV     H,A
04C3 22B640  SHLD   TOPABS      ;UPDATE TOP OF PAGE ABS.
04C6 7C      MOV     A,H
04C7 E607    ANI     07H
04C9 67      MOV     H,A
04CA 22C140  SHLD   TOPCRTC     ;UPDATE TOP OF PAGE CRTIC.
04CD+1804    JR     LB0117
                    DB     18H,LB0117-$-1

```

```

;
04CF 3C      LB0116: INR     A
04D0 32BB40  STA     ROW
04D3 2ABC40  LB0117: LHLD   CURSORABS
04D6 CDD503  CALL   LAB5      ;UPDATE CURSOR ABS.
04D9 C3870C  JMP    ENABSYNCINT

```

```

; CURSFORWARD:      ;CURSOR FORWARD
04DC 3ABA40  LDA     COLUMN
04DF FE4F    CPI     LINELENGTH-1 ;END OF LINE?
04E1 C8      RZ      ;YES.-->
04E2 3C      INR     A
04E3 32BA40  STA     COLUMN
04E6 2ABC40  LHLD   CURSORABS
04E9 23      INX     H
04EA C3AB03  JMP    LAB2

```

```

; IF ANSI
ANSIERASEDIS:
04ED 78      MOV     A,B      ;ERASE IN DISPLAY
04EE B7      ORA     A      ;ANSI ESC [ J
                    JRZ     ERASEEOP
04EF+2833    DB     28H,ERASEEOP-$-1
04F1 EB      XCHG
04F2 7E      MOV     A,M
04F3 B7      ORA     A
                    JRZ     ERASEEOP
04F4+282E    DB     28H,ERASEEOP-$-1
04F6 3D      DCR     A
04F7 CA3706  JZ     ERASEBEGDIS
04FA 3D      DCR     A
04FB C0      RNZ

```

ENDIF

```

; CLEARHOME:      ;ERASE SCREEN
04FC 3ABB40  LDA     ROW
04FF FE13    CPI     18H      ;25TH LINE?
0501 CA9E06  JZ     ERASELINE ;YES.-->
0504 CD1405  CALL   CURSHOME
0507 0678    MVI     B,LOW(SCREENSIZ/16)
0509 C3340B  JMP    SPACE16

```

```

; SETGRAPHICS:
050C EB      XCHG      ;ENTER GRAPHICS MODE

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```
050D+CBCE      SETB      1,M
050F C9         DB          0CBH,1*8+M+0C0H
                RET
```

```
;
RESETGRAPHICS:
```

```
0510 EB        XCHG                      ;EXIT GRAPHICS MODE
                RES          1,M
0511+CB8E      DB          0CBH,1*8+M+80H
0513 C9        RET
```

```
;
CURSHOME:      ;HOME CURSOR
```

```
0514 210000    LXI          H,0
0517 22BA40    SHLD         COLUMN
051A 2AB640    LHL         TOPABS
051D 22BC40    SHLD         CURSORABS
0520 22B840    SHLD         BEGLINABS
0523 C9        RET
```

```
;
ERASEEOP:      ;ERASE TO END OF PAGE
```

```
0524 3ABB40    LDA          ROW
0527 FE18      CPI          18H          ;25TH LINE?
                JRZ          ERASEEOL   ;YES.-->
0529+281B     DB          28H,ERASEEOL-$-1
052B 2AB640    LHL         TOPABS
052E 117F07    LXI          D,SCREENSIZE-1
0531 19        DAD          D
0532 37        STC
0533 3F        CMC
```

```
                LDED         CURSORABS
0534+ED5B     DB          0EDH,5BH
0536+BC40     DW          CURSORABS
                DSBC         D
0538+ED52     DB          0EDH,D*8+42H
```

```
053A 23        INX          H
053B EB        XCHG
053C 7A        MOV          A,D
053D E607     ANI          07H
053F 57        MOV          D,A
0540 2ABC40    LHL         CURSORABS
0543 C31A0B    JMP          SPACEDE
```

```
;
ERASEEOL:      ;ERASE TO END OF LINE
```

```
0546 2ABC40    LHL         CURSORABS
0549 3ABA40    LDA          COLUMN
                NEG
054C+ED44     DB          0EDH,44H
054E C650     ADI          50H
0550 5F        MOV          E,A
0551 1600     MVI          D,00H
0553 C31A0B    JMP          SPACEDE
```

```
;
INSERTLINE:    ;INSERT A LINE
```

```
0556 3ABB40    LDA          ROW
0559 FE18      CPI          18H          ;25TH LINE?
055E CA9E06    JZ            ERASELINE   ;YES.-->
                NEG
```

How about 0F here? 7/20/84

```

055E+ED44      DB      0EDH,44H
0560 C617      ADI      17H
                JRZ      LB0189
0562+2828      DB      28H,LB0189-$-1
0564 4F        MOV      C,A
0565 81        ADD      C
0566 81        ADD      C
0567 81        ADD      C
0568 81        ADD      C
0569 4F        MOV      C,A
                LDED     TOPABS
056A+ED5B      DB      0EDH,5BH
056C+B640      DW      TOPABS
056E 217F07    LXI      H,SCREENSIZE-1
0571 19        DAD      D
0572 E5        PUSH     H
0573 212F07    LXI      H,SCREENSIZE-LINELENGTH-1
0576 19        DAD      D
0577 D1        POP      D
0578 C5        LB0188: PUSH     B
0579 CD9205    CALL    LAB6
                LDDR     TOPABS
057C+EDB8      DB      0EDH,0B8H
057E C1        POP      B
057F 0D        DCR      C
                JRNZ     LB0188
0580+20F6      DB      20H,LB0188-$-1
0582 7A        MOV      A,D
0583 F6F8      ORI      0F8H
0585 57        MOV      D,A
0586 7C        MOV      A,H
0587 F6F8      ORI      0F8H
0589 67        MOV      H,A
                LDD      A
058A+EDA8      DB      0EDH,0A8H
058C CDEC03    LB0189: CALL    DOCRETUR
058F C3C80A    JMP      SPACE80
;
0592 7A        LAB6:  MOV      A,D
0593 F6F8      ORI      0F8H
0595 57        MOV      D,A
0596 7C        MOV      A,H
0597 F6F8      ORI      0F8H
0599 67        MOV      H,A
059A 0E10      MVI     C,10H
059C 0600      MVI     B,00H
059E C9        RET
;
DELLINE:      ;DELETE A LINE
059F CDEC03    CALL    DOCRETUR
05A2 EB        XCHG
05A3 215000    LXI      H,LINELNGTH
05A6 19        DAD      D
05A7 3ABB40    LDA      ROW
05AA FE18      CPI      18H ;25TH LINE?
05AC CA9E06    JZ      ERASELINE ;YES.-->

```



Handwritten annotations: Circles around 0F8H in ORI instructions and F0 symbols next to them.

Vertical line numbers on the right margin: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76

05AF+ED44	NEG	DB	0EDH,44H
05B1 C617	ADI	17H	
	JRZ	LB0184	
05B3+2810	DB	28H, LB0184-\$-1	
05B5 4F	MOV	C,A	
05B6 81	ADD	C	
05B7 81	ADD	C	
05B8 81	ADD	C	
05B9 81	ADD	C	
05BA 4F	MOV	C,A	
05BB C5	LB0183: PUSH	B	
05BC CD9205	CALL	LAB6	
	LDIR		
05BF+EDB0	DB	0EDH,0B0H	
05C1 C1	POP	B	
05C2 0D	DCR	C	
	JRNZ	LB0183	
05C3+20F6	DB	20H, LB0183-\$-1	
05C5 EB	LB0184: XCHG		
05C6 C3C80A	JMP	SPACE80	
; DELCHAR: ;DELETE A CHARACTER			
05C9 3ABA40	LDA	COLUMN	
	NEG		
05CC+ED44	DB	0EDH,44H	
05CE C64F	ADI	4FH	
05D0 4F	MOV	C,A	
05D1 0600	MVI	B,00H	
	LDED	CURSORABS	
05D3+ED5B	DB	0EDH,5BH	
05D5+BC40	DW	CURSORABS	
	JRZ	LB0181	
05D7+2812	DB	28H, LB0181-\$-1	
05D9 2ABC40	LHLD	CURSORABS	
05DC 23	INX	H	
05DD 7A	LB0180: MOV	A,D	
05DE F6F8	ORI	0F8H	F0
05E0 57	MOV	D,A	
05E1 7C	MOV	A,H	
05E2 F6F8	ORI	0F8H	F0
05E4 67	MOV	H,A	
	LDI		
05E5+EDA0	DB	0EDH,0A0H	
05E7 78	MOV	A,B	
05E8 B1	ORA	C	
	JRNZ	LB0180	
05E9+20F2	DB	20H, LB0180-\$-1	
05EB 3E20	LB0181: MVI	A,20H	
05ED 12	STAX	D	
05EE C9	RET		
; INSERTCHAROFF: ;EXIT INSERT CHARACTER MODE			
05EF EB	XCHG		
	RES	6,M	
05F0+CBB6	DB	0CBH,6*8+M+80H	OK

05F2 C9

RET

; IDENTVT52: ;HEATH IDENTIFY AS VT52

05F3 CD290C CALL OUTESC
 05F6 3E2F MVI A, '/'
 05F8 CD2B0C CALL OUTPUTA
 05FB 3E4B MVI A, 'K'
 05FD C32B0C JMP OUTPUTA

OK

; CURSADDR: ;DIRECT CURSOR ADDR

0600 CD9E0B CALL GETRECBYTE
 0603 FE18 CPI 18H
 0605 C8 RZ
 0606 FE20 CPI 20H
 JRC LB0169
 0608+3814 DB 38H, LB0169-\$-1
 060A FE38 CPI 38H
 JRC LB0168
 060C+380B DB 38H, LB0168-\$-1
 JRNZ LB0169
 060E+200E DB 20H, LB0169-\$-1
 0610 47 MOV B, A
 0611 3AC940 LDA MODEFLAG2
 0614 E680 ANI 80H ; 25TH LINE ENABLED?
 JRZ LB0169 ; NO.-->

OK

0616+2806 DB 28H, LB0169-\$-1
 0618 78 MOV A, B
 0619 D620 LB0168: SUI 20H
 061B 32BB40 STA ROW
 061E CD9E0B LB0169: CALL GETRECBYTE
 0621 FE18 CPI 18H
 JRZ LB0172
 0623+280F DB 28H, LB0172-\$-1
 0625 FE20 CPI 20H
 JRC LB0170
 0627+3804 DB 38H, LB0170-\$-1
 0629 FE70 CPI 70H
 JRC LB0171
 062B+3802 DB 38H, LB0171-\$-1
 062D 3E6F LB0170: MVI A, 6FH
 062F D620 LB0171: SUI 20H
 0631 32BA40 STA COLUMN
 0634 C37306 LB0172: JMP CURSTOSAVED

subtract from A data

; ERASEBEGDIS: ;ERASE FROM BEGINNING OF PAGE

0637 3ABB40 LDA ROW
 063A FE18 CPI 18H ; 25TH LINE?
 JRZ ERASEBOL ; YES.-->
 063C+2813 DB 28H, ERASEBOL-\$-1
 063E B7 ORA A ; TOP LINE?
 JRZ ERASEBOL ; YES.-->
 063F+2810 DB 28H, ERASEBOL-\$-1
 0641 6F MOV L, A
 0642 5F MOV E, A
 0643 2600 MVI H, 00H
 0645 57 MOV D, A

Jump if zero

subtract from A, initial

0646	19	DAD	D	
0647	19	DAD	D	
0648	19	DAD	D	
0649	19	DAD	D	
064A	45	MOV	B,L	
064B	2AB640	LHLD	TOPABS	
064E	CD340B	CALL	SPACE16	
		ERASEBOL:		;ERASE FROM BEGINNING OF LINE
0651	2AB840	LHLD	BEGLINABS	
0654	3ABA40	LDA	COLUMN	
0657	3C	INR	A	
0658	47	MOV	B,A	
0659	3E20	MVI	A,20H	
065B	77	LB0135: MOV	M,A	
065C	23	INX	H	
		DJNZ	LB0135	
065D+10FC		DB	10H,LB0135-\$-1	
065F	C9	RET		
		;		
		IF ANSI		
		ANSISAVECURS:		
0660	78	MOV	A,B	;SAVE CURSOR POSITION
0661	B7	ORA	A	;ANSI ESC [S
0662	C0	RNZ		
		ENDIF		
		;		
		SAVECURS:		
0663	2ABA40	LHLD	COLUMN	;HEATH SAVE CURSOR POSITION
0666	22C540	SHLD	CURSORSAVE	OK
0669	C9	RET		
		;		
		IF ANSI		
		ANSICURSTOSAVE:		
066A	78	MOV	A,B	;SET CURSOR TO SAVED POSITION
066B	B7	ORA	A	;ANSI ESC [U
066C	C0	RNZ		OK
		ENDIF		
		;		
		SETCURS:		
066D	2AC540	LHLD	CURSORSAVE	;HEATH SET CURSOR TO SAVED
0670	22BA40	SHLD	COLUMN	
		CURSTOSAVED:		
0673	3ABB40	LDA	ROW	
0676	CDBA0A	CALL	ATIMES80	Result in HL Now in DE
0679	EB	XCHG		
067A	2AB640	LHLD	TOPABS	ADD DE TO HL
067D	19	DAD	D	OK
067E	7C	MOV	A,H	
067F	F6F8	ORI	0F8H	rand?
0681	67	MOV	H,A	
0682	22B840	SHLD	BEGLINABS	
0685	3ABA40	LDA	COLUMN	
0688	5F	MOV	E,A	
0689	1600	MVI	D,00H	
068B	C3D503	JMP	LAB5	
		;		

```

IF ANSI
ANSIERASELINE: ;ERASE IN LINE
;ANSI ESC [ K
068E 78 MOV A,B
068F B7 ORA A
0690 CA4605 JZ ERASEEOL
0693 EB XCHG
0694 7E MOV A,M
0695 B7 ORA A
0696 CA4605 JZ ERASEEOL
0699 3D DCR A
JRZ ERASEBOL
069A+28B5 DB 28H,ERASEBOL-$-1
069C 3D DCR A
069D C0 RNZ

```

OK (Heath)

ENDIF

```

;
ERASELINE: ;ERASE CURSOR LINE
069E 2AB840 LHLD BEGLINABS
06A1 C3C80A JMP SPACE80

```

OK

; CURSPOSREPORT: ;CURSOR POSITION REPORT

```

06A4 CD290C CALL OUTESC
06A7 3E49 MVI A,'I'
06A9 CD2B0C CALL OUTPUTA
06AC 3ABB40 LDA ROW
06AF CDB506 CALL ADIOUT
06B2 3ABA40 LDA COLUMN
06B5 C620 ADIOUT: ADI 20H ;MAKE PRINTABLE
06B7 C32B0C JMP OUTPUTA

```

OK

; SETREVVVIDEO: ;ENTER REVERSE VIDEO MODE

```

06BA EB XCHG
06BB 7E MOV A,M
06BC F680 ORI 80H
06BE 77 MOV M,A
06BF C9 RET

```

OK

; RESETREVVVIDEO: ;EXIT REVERSE VIDEO MODE

```

06C0 EB XCHG
RES 7,M
06C1+CBBE DB 0CBH,7*8+M+80H
06C3 C9 RET

```

OK

; MODBAUDRATE: ;MODIFY BAUD RATE

```

06C4 CD9E0B CALL GETRECBYTE
06C7 FE41 CPI 41H
06C9 D8 RC
06CA FE50 CPI 50H
06CC D0 RNC
06CD E60F ANI 0FH

```

BAUDMOD1:

```

06CF 47 MOV B,A
06D0 3ACA40 LDA DIP1SAVE
06D3 E6F0 ANI 0FH
06D5 B0 ORA B
06D6 32CA40 STA DIP1SAVE

```

OK

SETBAUD:

```

06D9 3E80      MVI    A,80H
06DB D343      OUT    UARTLC
06DD 3ACA40    LDA    DIP1SAVE
06E0 4F        MOV    C,A
06E1 E60F      ANI    0FH
                JRZ    LB0157
06E3+2801     DB    28H, LB0157-$-1
06E5 3D        DCR    A
06E6 47        LB0157: MOV   B,A
06E7 07        RLC
06E8 21950F    LXI    H,BAUDTAB ;BAUD RATE DIVISOR TABLE
06EB 85        ADD    L
06EC 6F        MOV    L,A
06ED 7E        MOV    A,M
06EE D340      OUT    UARTDL
06F0 23        INX    H
06F1 7E        MOV    A,M
06F2 D341      OUT    UARTDH
06F4 AF        XRA    A
06F5 B0        ORA    B
06F6 0600      MVI    B,00H
                JRNZ   LB0158
06F8+2002     DB    20H, LB0158-$-1
06FA 0604      MVI    B,04H
06FC 79        LB0158: MOV   A,C
06FD E670      ANI    70H
06FF 0F        RRC
0700 B0        ORA    B
0701 47        MOV    B,A
0702 E608      ANI    08H
0704 3E02      MVI    A,02H
                JRNZ   LB0159
0706+2002     DB    20H, LB0159-$-1
0708 3E03      MVI    A,03H
070A B0        LB0159: ORA    B
070B D343      OUT    UARTLC
070D 3E01      MVI    A,01H
070F D341      OUT    UARTDH
0711 3E03      MVI    A,03H
0713 D344      OUT    UARTMC
0715 C9        RET

```

; SETLINEWRAP: ;WRAP AT END OF LINE

```

0716 0A        LDAX   B
0717 F604      ORI    04H
0719 02        STAX   B
071A C9        RET

```

; SETNOWRAP: ;DISCARD AT END OF LINE

```

071B 0A        LDAX   B
071C E6FB      ANI    0FBH
071E 02        STAX   B
071F C9        RET

```

; RESETMODE:

sets bit 3 to zero

OK

OK

```

0720 21C90F      LXI      H,RESETTAB      ;HEATH RESET MODE
                  JR          LB0203
0723+1803       DB          18H,LB0203-$-1

```

SETMODE:

```

0725 21B30F      LXI      H,SETTAB      ;HEATH SET MODE
0728 CD9E0B      LB0203: CALL    GETRECBYTE
072B FE18        CPI          18H
072D C8          RZ
072E FE30        CPI          30H
0730 DA9902      JC          DISPLAY
0733 FE3B        CPI          3BH
0735 D29902      JNC         DISPLAY
0738 E60F        ANI          0FH
073A 07          DOMODE: RLC
073B 85          ADD          L
073C 6F          MOV          L,A
073D 7E          MOV          A,M
073E 23          INX          H
073F 66          MOV          H,M
0740 6F          MOV          L,A
0741 01C840      LXI      B,DIP2SAVE
0744 11C740      LXI      D,MODEFLAG1
0747 E9          PCHL

```

ANSIREINIT:

```

; IF ANSI
ANSIREINIT:      ;RESET TO POWER UP CONFIG.
0748 78          MOV          A,B      ;ANSI ESC [ Z
0749 B7          ORA          A
074A C0          RNZ

```

ENDIF

REINIT:

```

; REINIT: DI      ;RESET TO POWER UP CONFIG.
074B F3 uH      INIT:  LXI      H,RECBUF      CPU RAM STARTS @ RECBUF (400H)
074C 210040      LXI      D,RECBUF+1
074F 110140      LXI      B,RAMSIZE      ;CPU RAM SIZE TO CLEAR
0752 010004      MVI      M,00H
0755 3600      LDIR

```

disable interrupts
sets CPU RAM to zeroes
move 00 into Recbuf

LOOP

```

0757+EDB0       DB          0EDH,0B0H
0759 21A440      LXI      H,KEYBUF=400A load
075C 22B440      SHLD     KBUFPT=40BA store
075F 3E18        MVI      A,18H
0761 32BE40      STA      CRTCROWS
0764 DB20        IN       DIP1=0020
0766 F601        ORI      01H
0768 32CA40      STA      DIP1SAVE
076B 3E87        MVI      A,INITCONFIG
076D 32C840      STA      DIP2SAVE
0770 3E08        MVI      A,08H
0772 32C940      STA      MODEFLAG2
0775 310041      LXI      SP,STACK
0778 2100F0      LXI      H,-DMEMSIZE
077B 22B640      SHLD    TOPABS
077E 22B840      SHLD    BEGLINABS
0781 22BC40      SHLD    CURSORABS
0784 110010      LXI      D,DMEMSIZE

```

store 24 in CRTROWS

get value @ DIP1

OR the dipswitch w/01H (sets LSB)
and store in 40CA (save)
INITIAL OPTIONS = 87
store in DIP2SAVE

SET ON-LINE FLAG
modeflag2 = 08 now
load stack pointer with STACK ADDR (=4100)
negate Display mem size & put in HL
store in TOPABS (=40B6)
and in BegLinABS (=40B8)
and in cursor ABS (=40BC)
load DE with Disp. Size (=1000)



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76

```

0787 CD1A0B      CALL  SPACEDE
078A CD9F0A      CALL  INITCRTC
078D CDD906      CALL  SETBAUD
0790 CD1B08      CALL  CURSON
                                ;SET INTERRUPT MODE 1.
0793+ED56        DB      0EDH,56H
0795 3E00        MVI   A,00H
0797 D364        OUT   CRTICIE      ;ENABLE SYNC INTERRUPT
0799 FB          EI
079A C30300      JMP   MAINLOOP

```

call clearhome *CD 46 00* *jump to patch area for clear screen*

```

;
;
IF ENABDISKBOARD
ENABKBOARD:      ;ENABLE KEYBOARD
                  LDA   MODEFLAG2
                  ANI   0FBH
                  STA   MODEFLAG2
                  RET
DISABKBOARD:     ;DISABLE KEYBOARD
                  LDA   MODEFLAG2
                  ORI   04H
                  STA   MODEFLAG2
                  RET
;
ENDIF

```

```

;
;*****
;
; SET/RESET MODE ROUTINES
;
SETFULLDUP:     ;SET FULL DUPLEX
079D 3ACA40      LDA   DIP1SAVE
07A0 F680        ORI   80H
07A2 32CA40      SETDUP: STA  DIP1SAVE
07A5 C9          RET

```

```

;
RESETFULLDUP:   ;SET HALF DUPLEX
07A6 3ACA40      LDA   DIP1SAVE
07A9 E67F        ANI   7FH
                  JR    SETDUP
07AB+18F5        DB    18H,SETDUP-$-1

```

```

;
ENAB25TH:
07AD 3AC940      LDA   MODEFLAG2      ;ENABLE 25TH LINE
07B0 47          MOV   B,A
07B1 E680        ANI   80H      ;ALREADY ENABLED?
07B3 C0          RNZ      ;YES.-->
07B4 3E80        MVI   A,80H
07B6 B0          ORA   B      ;SET FLAG
07B7 32C940      STA   MODEFLAG2
07BA 2AB640      LHLD  TOPABS
07BD 118007      LXI   D,SCREENSIZE
07C0 19          DAD   D
07C1 CDC80A      CALL  SPACE80      ;CLEAR 25TH LINE
07C4 3E19        MVI   A,19H      ;UPDATE # OF ROWS

```

```

SETROWS:
07C6 32BE40      STA   CRTCROWS

```

```

07C9 C9          RET
;
DISAB25TH:
07CA 3AC940     LDA    MODEFLAG2    ;DISABLE 25TH LINE
07CD E67F       ANI    7FH
07CF 32C940     STA    MODEFLAG2
07D2 3E18       MVI    A,18H
                JR     SETROWS      OK
07D4+18F0      DB     18H,SETROWS-$-1

```

```

;
DISABCLICK:
07D6 0A        LDAX   B          ;DISABLE CLICK
07D7 E6FD      ANI    0FDH
07D9 02        STAX   B          OK
07DA C9        RET

```

```

;
ENABCLICK:
07DB 0A        LDAX   B          ;ENABLE CLICK
07DC F602      ORI    02H
07DE 02        STAX   B          OK
07DF C9        RET

```

```

;
SETHOLDSCREEN:
07E0 EB        XCHG                ;ENTER HOLD SCREEN MODE
                SETB   0,M
07E1+CBC6      DB     0CBH,0*8+M+0C0H
07E3 3E01      MVI    A,01H      OK
07E5 32CB40    STA    HOLDCNT
07E8 C9        RET

```

```

;
RESETHOLDSCREEN:
07E9 EB        XCHG                ;EXIT HOLD SCREEN MODE
                RES    0,M
07EA+CB86      DB     0CBH,0*8+M+80H      OK
07EC C9        RET

```

```

;
SETBLOCKCURS:
07ED 0A        LDAX   B          ;SET BLOCK CURSOR
07EE F601      ORI    01H
07F0 02        STAX   B          OK
07F1 1A        LDAX   D
07F2 E610      ANI    10H
07F4 C0        RNZ

```

```

BLOCKCURS:
07F5 2660      MVI    H,60H
07F7 2E09      MVI    L,09H
07F9 22BF40    BLOC1: SHLD   CURSORTYPE
07FC C9        RET

```

```

;
SETULINECURS:
07FD 0A        LDAX   B          ;SETUP UNDERLINE CURSOR
07FE E6FE      ANI    0FEH
0800 02        STAX   B          OK
0801 1A        LDAX   D
0802 E610      ANI    10H
0804 C0        RNZ

```

↑
↓

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76

```

ULINECURS:
0805 2649      MVI      H,49H
0807 2E09      MVI      L,09H
0809+18EE     JR        BLOC1
                DB        18H,BLOC1-$-1
;
DISABCURS:
080B 3AC740    LDA      MODEFLAG1      ;DISABLE CURSOR
080E F610      ORI      10H
0810 32C740    STA      MODEFLAG1
0813 2620      MVI      H,20H
0815 2E00      MVI      L,00H
0817 22BF40    SHLD     CURSORTYPE
081A C9        RET
;
CURSON:
081B 3AC740    LDA      MODEFLAG1
081E E6EF      ANI      0EFH
0820 32C740    STA      MODEFLAG1
0823 3AC840    LDA      DIP2SAVE
0826 E601      ANI      01H            ;BLOCK CURSOR?
0828 C2F507    JNZ      BLOCKCURS     ;YES.-->
082B C30508    JMP      ULINECURS
;
SETPADSHIFT:
082E 0A        LDAX     B              ;ENTER KEYPAD SHIFTED MODE
082F F640      ORI      40H
0831 02        STAX     B
0832 C9        RET
;
RESETPADSHIFT:
0833 0A        LDAX     B              ;EXIT KEYPAD SHIFTED MODE
0834 E6BF      ANI      0BFH
0836 02        STAX     B
0837 C9        RET
;
SETALTPAD:
0838 1A        LDAX     D              ;ENTER ALT. KEYPAD MODE
0839 F604      ORI      04H
083B 12        STAX     D
083C C9        RET
;
RESETPALTPAD:
083D 1A        LDAX     D              ;EXIT ALT. KEYPAD MODE
083E E6FB      ANI      0FBH
0840 12        STAX     D
0841 C9        RET
;
AUTOLFON:
0842 0A        LDAX     B              ;AUTO LF ON CR
0843 F608      ORI      08H
0845 02        STAX     B
0846 C9        RET
;
AUTOLFOFF:
0847 0A        LDAX     B              ;RESET AUTO LF ON CR
0848 E6F7      ANI      0F7H

```

```

084A 02      STAX  B
084B C9      RET

;
AUTOCRON:    ;AUTO CR ON LF
084C 0A      LDAX  B
084D F610    ORI   10H
084F 02      STAX  B
0850 C9      RET

;
AUTOCROFF:   ;NO AUTO CR
0851 0A      LDAX  B
0852 E6EF    ANI   0EFH
0854 02      STAX  B
0855 C9      RET

;
AUTOREPEAT:  ;SET REPEAT KEY FLAG
0856 0A      LDAX  B
0857 F680    ORI   80H
0859 02      STAX  B
085A C9      RET

;
NOAUTOREPEAT: ;RESET REPEAT KEY FLAG
085B 0A      LDAX  B
085C E67F    ANI   7FH
085E 02      STAX  B
085F C9      RET

;
;*****
IF ANSI
;
; ANSI ROUTINES START HERE.
;
ANSIBRACKET:
0860 CD530A  CALL  ANSIPARAMEVAL ;ANSI "ESC [" LOOKUP
0863 C5      PUSH  B
0864 1617    MVI  D,17H
0866 1E03    MVI  E,03H
0868 212F0F  LXI  H,ANSITAB2
086B CD580C  CALL  LOOKUP
086E C1      POP   B
086F D8      RC
0870 23      INX  H
0871 66      MOV  H,M
0872 6F      MOV  L,A
0873 11CC40  LXI  D,ANSISAVE
0876 E9      PCHL

;
SETRESETMODE:
0877 78      MOV  A,B ;ANSI ESC [ >
0878 B7      ORA  A
0879 C0      RNZ
087A CD530A  CALL  ANSIPARAMEVAL
087D FE48    CPI  'H'
JRZ ANSISSET
087F+2805    DB  28H,ANSISSET-$-1
0881 FE4C    CPI  'L'

```

OK

OK

OK

OK

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57


```

0883+2813      JRZ   ANSIRESET
0885 C9        DB    28H,ANSIRESET-$-1
               RET

```

```

;
ANSISET:

```

```

0886 78      MOV   A,B
0887 B7      ORA   A
0888 C8      RZ
0889 21B30F  LB0087: LXI  H,SETTAB      ;SET MODE TABLE
088C 1A      LDAX D
088D FE0B    CPI   0BH      ;CHECK RANGE
088F D0      RNC
0890 D5      PUSH D
0891 CD3A07  CALL  DOMODE
0894 D1      POP  D
0895 13      INX  D
               JR   LB0087
0896+18F1    DB    18H,LB0087-$-1

```

or Accumulator with Reg A. store results in Acc.

```

;
ANSIRESET:

```

```

0898 78      MOV   A,B
0899 B7      ORA   A
089A C8      RZ
089B 21C90F  LB0089: LXI  H,RESETTAB      ;RESET MODE TABLE
089E 1A      LDAX D
089F FE0B    CPI   0BH      ;CHECK RANGE
08A1 D0      RNC
08A2 D5      PUSH D
08A3 CD3A07  CALL  DOMODE
08A6 D1      POP  D
08A7 13      INX  D
               JR   LB0089
08A8+18F1    DB    18H,LB0089-$-1

```

```

;
CHANGEMODE:

```

```

08AA 78      MOV   A,B      ;ANSI ESC [ ?
08AB B7      ORA   A
08AC C0      RNZ
08AD CD530A  CALL  ANSIPARAMEVAL
08B0 FE48    CPI   'H'
               JRZ  LB0082
08B2+2805    DB    28H,LB0082-$-1
08B4 FE4C    CPI   'L'
               JRZ  LB0084
08B6+281B    DB    28H,LB0084-$-1
08B8 C9      RET

```

```

;
LB0082:

```

```

08B9 78      MOV   A,B
08BA B7      ORA   A
08BB C8      RZ
08BC EB      XCHG
08BD 01C840  LXI  B,DIP2SAVE
08C0 7E      LB0083: MOV  A,M
08C1 FE02    CPI   02H
08C3 CC960A  CZ   SETHEATHMODE
08C6 7E      MOV  A,M

```

```

08C7 FE07      CPI      07H
08C9 CC1607    CZ        SETLINEWRAP
08CC 7E        MOV        A,M
08CD FE48      CPI        'H'
08CF C8        RZ
08D0 23        INX        H
                JR        LB0083
08D1+18ED      DB        18H,LB0083-$$-1
                ;
08D3 78        LB0084: MOV      A,B
08D4 B7        ORA        A
08D5 C8        RZ
08D6 EB        XCHG
08D7 01C840    LXI        B,DIP2SAVE
08DA 7E        LB0085: MOV      A,M
08DB FE07      CPI        07H
08DD CC1B07    CZ        SETNOWRAP
08E0 7E        MOV        A,M
08E1 FE4C      CPI        'L'
08E3 C8        RZ
08E4 23        INX        H
                JR        LB0085
08E5+18F3      DB        18H,LB0085-$$-1
                ;
                ANSICURSUP:
08E7 78        MOV        A,B          ;CURSOR UP
08E8 B7        ORA        A          ;ANSI ESC [ A
08E9 CA4204    JZ        CURSUP
08EC 1A        LDAX       D
08ED B7        ORA        A
08EE CA4204    JZ        CURSUP
08F1 F5        LB0095: PUSH     PSW
08F2 CD4204    CALL     CURSUP
08F5 F1        POP        PSW
08F6 3D        DCR        A
                JRNZ     LB0095
08F7+20F8      DB        20H,LB0095-$$-1
08F9 C9        RET
                ;
                ANSICURSDOWN:
08FA 78        MOV        A,B          ;CURSOR DOWN
08FB B7        ORA        A          ;ANSI ESC [ B
08FC CA8904    JZ        CURSDOWN
08FF 1A        LDAX       D
0900 B7        ORA        A
0901 CA8904    JZ        CURSDOWN
0904 F5        LB0090: PUSH     PSW
0905 CD8904    CALL     CURSDOWN
0908 F1        POP        PSW
0909 3D        DCR        A
                JRNZ     LB0090
090A+20F8      DB        20H,LB0090-$$-1
090C C9        RET
                ;
                ANSICURSFOR:
090D 78        MOV        A,B          ;CURSOR FORWARD

```


0958 C37306 JMP CURSTOSAVED

; ANSIINSERTLINE:

```

095B 78 MOV A,B ;INSERT LINE(S)
095C B7 ORA A ;ANSI ESC [ L
095D CA5605 JZ INSERTLINE
0960 1A LDAX D
0961 B7 ORA A
0962 CA5605 JZ INSERTLINE
0965 F5 LB0103: PUSH PSW
0966 CD5605 CALL INSERTLINE
0969 F1 POP PSW
096A 3D DCR A
JRNZ LB0103
096B+20F8 DB 20H,LB0103-$-1
096D C9 RET

```

; ANSIDELLINE:

```

096E 78 MOV A,B ;DELETE LINE(S)
096F B7 ORA A ;ANSI ESC [ M
0970 CA9F05 JZ DELLINE
0973 1A LDAX D
0974 B7 ORA A
0975 CA9F05 JZ DELLINE
0978 F5 LB0102: PUSH PSW
0979 CD9F05 CALL DELLINE
097C F1 POP PSW
097D 3D DCR A
JRNZ LB0102
097E+20F8 DB 20H,LB0102-$-1
0980 C9 RET

```

; ANSIDELCHAR:

```

0981 78 MOV A,B ;DELETE CHARACTER(S)
0982 B7 ORA A ;ANSI ESC [ P
0983 CAC905 JZ DELCHAR
0986 1A LDAX D
0987 B7 ORA A
0988 CAC905 JZ DELCHAR
098B F5 LB0101: PUSH PSW
098C CDC905 CALL DELCHAR
098F F1 POP PSW
0990 3D DCR A
JRNZ LB0101
0991+20F8 DB 20H,LB0101-$-1
0993 C9 RET

```

; ANSIMODMODE: ;CHANGE MODES

```

0994 78 MOV A,B ;ANSI ESC [ H
0995 B7 ORA A
0996 C8 RZ
0997 EB XCHG
0998 01C840 LXI B,DIP2SAVE
099B 11C740 LB0110: LXI D,MODEFLAG1

```

; IF ENABDISKBOARD

```
MOV A,M
CPI 02H
CZ DISABKBOARD
```

```
ENDIF
```

```
;
```

```
099E 7E MOV A,M
099F E5 PUSH H
09A0 FE04 CPI 04H
09A2 CC3E04 CZ INSERTCHARON
09A5 E1 POP H
09A6 7E MOV A,M
09A7 FE14 CPI 14H
09A9 CC4C08 CZ AUTOCRON
09AC 7E MOV A,M
09AD FE48 CPI 'H'
09AF C8 RZ
09B0 23 INX H
JR LB0110
09B1+18E8 DB 18H, LB0110-$-1
```

```
;
```

```
ANSIMODMODE2: ;CHANGE MODES
```

```
09B3 78 MOV A,B ;ANSI ESC [ L
09B4 B7 ORA A
09B5 C8 RZ
09B6 EB XCHG
09B7 01C840 LXI B,DIP2SAVE
09BA 11C740 LB0104: LXI D,MODEFLAG1
```

```
;
```

```
IF ENABDISKBOARD
```

```
MOV A,M
CPI 02H
CZ ENABKBOARD
```

```
ENDIF
```

```
;
```

```
09BD 7E MOV A,M
09BE FE04 CPI 04H
09C0 E5 PUSH H
09C1 CCEF05 CZ INSERTCHAROFF
09C4 E1 POP H
09C5 7E MOV A,M
09C6 FE14 CPI 14H
09C8 CC5108 CZ AUTOCROFF
09CB 7E MOV A,M
09CC FE4C CPI 'L'
09CE C8 RZ
09CF 23 INX H
JR LB0104
09D0+18E8 DB 18H, LB0104-$-1
```

```
;
```

```
ANSIDISPLAYMODE: ;CHANGE DISPLAY MODE
```

```
09D2 78 MOV A,B ;ANSI ESC [ M
09D3 B7 ORA A
JRZ LB0107
09D4+2809 DB 28H, LB0107-$-1
09D6 D5 PUSH D
09D7 D1 LB0106: POP D
```

```

09D8 1A          LDAX  D
09D9 13          INX   D
09DA D5          PUSH  D
09DB 21D709     LXI   H, LB0106
09DE E5          PUSH  H
09DF 21740F     LB0107: LXI   H, ANSITAB3
09E2 1604       MVI   D, 04H
09E4 1E03       MVI   E, 03H
09E6 CD580C     CALL  LOOKUP
                   JRC   LB0108
09E9+3807      DB    38H, LB0108-$-1
09EB 23         INX   H
09EC 66         MOV   H, M
09ED 6F         MOV   L, A
09EE 11C740     LXI   D, MODEFLAG1
09F1 E9         PCHL
                   ;
09F2 E1         LB0108: POP  H
09F3 D1         POP  D
09F4 C9         RET
                   ;
ANSICURSREPORT:
09F5 78         MOV   A, B           ;REPORT CURSOR POSITION
09F6 B7         ORA   A           ;ANSI ESC [ H
09F7 C8         RZ
09F8 1A         LDAX  D
09F9 FE06       CPI   06H
09FB C0         RNZ
09FC CD290C     CALL  OUTESC
09FF 3E5B       MVI   A, '['
0A01 CD2B0C     CALL  OUTPUTA
0A04 3ABB40     LDA   ROW
0A07 3C         INR   A
0A08 CD3A0A     CALL  BINDEC
0A0B 7A         MOV   A, D
0A0C B7         ORA   A
                   JRZ   LB0092
0A0D+2805      DB    28H, LB0092-$-1
0A0F F630       ORI   30H
0A11 CD2B0C     CALL  OUTPUTA
0A14 7B         LB0092: MOV  A, E
0A15 F630       ORI   30H
0A17 CD2B0C     CALL  OUTPUTA
0A1A 3E3B       MVI   A, ';'
0A1C CD2B0C     CALL  OUTPUTA
0A1F 3ABA40     LDA   COLUMN
0A22 3C         INR   A
0A23 CD3A0A     CALL  BINDEC
0A26 7A         MOV   A, D
0A27 B7         ORA   A
                   JRZ   LB0093
0A28+2805      DB    28H, LB0093-$-1
0A2A F630       ORI   30H
0A2C CD2B0C     CALL  OUTPUTA
0A2F 7B         LB0093: MOV  A, E
0A30 F630       ORI   30H

```

```
0A32 CD2B0C CALL OUTPUTA
0A35 3E52 MVI A,'R'
0A37 C32B0C JMP OUTPUTA
```

```
1
2 0A3A 1600 BINDEC: MVI D,00H
3 0A3C FE0A LB0138: CPI 0AH
4 JRC LB0139
5 0A3E+3805 DB 38H, LB0139-$-1
6 0A40 D60A SUI 0AH
7 0A42 14 INR D
8 JR LB0138
9 0A43+18F7 DB 18H, LB0138-$-1
10 0A45 5F LB0139: MOV E,A
11 0A46 C9 RET
```

```
12 ;
13 ;
14 ANSIMODBAUD:
```

```
15 0A47 78 MOV A,B ;MODIFY BAUD RATE
16 0A48 B7 ORA A ;ANSI ESC [ R
17 JRZ LB0105
18 0A49+2805 DB 28H, LB0105-$-1
19 0A4B EB XCHG
20 0A4C 7E MOV A,M
21 0A4D FE10 CPI 10H
22 0A4F D0 RNC
23 0A50 C3CF06 LB0105: JMP BAUDMOD1
```

```
24 ;
25 ;*****
26 ;
27 ANSIPARAMEVAL: ;EVALUATE ANSI PARAMETERS
```

```
28 0A53 21CC40 LXI H,ANSISAVE
29 0A56 010000 LXI B,0
30 0A59 70 MOV M,B
31 0A5A CD9E0B CALL GETRECBYTE
32 0A5D FE3B CPI ';'
33 0A5F C8 RZ
34 0A60 FE30 LB0191: CPI '0'
35 JRC LB0195
36 0A62+3826 DB 38H, LB0195-$-1
37 0A64 FE3A CPI ':'
38 JRNC LB0192
39 0A66+300F DB 30H, LB0192-$-1
40 0A68 E60F ANI 0FH
41 0A6A 57 MOV D,A
42 0A6B 7E MOV A,M
43 0A6C 07 RLC
44 0A6D 77 MOV M,A
45 0A6E 07 RLC
46 0A6F 07 RLC
47 0A70 86 ADD M
48 0A71 82 ADD D
49 0A72 77 MOV M,A
50 0A73 04 INR B
51 0A74 0C INR C
52 JR LB0194
53 0A75+180E DB 18H, LB0194-$-1
```

*POT 0 in ANSISAVE
get next byte*

IT WAS A ';' Ret

```

0A77 FE3B      LB0192: CPI      ', '
                JRNZ     LB0195
0A79+200F     DB      20H, LB0195-$-1
0A7B 7D       MOV      A, L
0A7C FEDA     CPI      0DAH
                JRZ      LB0193
0A7E+2801     DB      28H, LB0193-$-1
0A80 23       INX      H
0A81 0E00     LB0193: MVI     C, 00H
0A83 3600     MVI     M, 00H
0A85 CD9E0B   LB0194: CALL    GETRECBYTE
                JR      LB0191
0A88+18D6     DB      18H, LB0191-$-1
0A8A 57       LB0195: MOV     D, A
0A8B AF       XRA      A
0A8C B1       ORA      C
                JRZ     LB0196
0A8D+2801     DB      28H, LB0196-$-1
0A8F 23       INX      H
0A90 72       LB0196: MOV     M, D
0A91 7A       MOV     A, D
0A92 11CC40   LXI     D, ANSISAVE
0A95 C9       RET

```

```

;
; SETHEATHMODE: ; ENTER HEATH MODE

```

```

0A96 3AC840   LDA      DIP2SAVE
0A99 E6DF     ANI     0DFH
0A9B 32C840   STA     DIP2SAVE
0A9E C9       RET

```

```

;
; ENDIF ; END OF ANSI ROUTINES

```

```

;
; *****

```

```

; INITIALIZE CRTC

```

```

; INITCRTC:

```

```

0A9F DB20     IN      DIP1
0AA1 E601     ANI     01H
0AA3 21EF0F   LXI     H, CRTCTAB2 ; CRTC TABLE 2.
                JRNZ     LB0163
0AA6+2003     DB      20H, LB0163-$-1
0AA8 21DF0F   LXI     H, CRTCTAB1 ; CRTC TABLE 1.
0AAB 0610     LB0163: MVI     B, 10H
0AAD 0E00     MVI     C, 00H
0AAF 79       LB0164: MOV     A, C
0AB0 D360     OUT     CRTCWA
0AB2 7E       MOV     A, M
0AB3 D361     OUT     CRTCWD
0AB5 0C       INR     C
0AB6 23       INX     H
                DJNZ    LB0164
0AB7+10F6     DB      10H, LB0164-$-1
0AB9 C9       RET

```

*all but
TURN OFF / low order BIT and check for zero result.*

Jump = 2 flag #0

*writes 00H to CRTCWA 1st byte, set reg in CRTC
get CRT TAB into A
write to CRTC Reg.*

*incr. ptr to
& mem*

] loop until B=0 (dec B, test)

```

; ATIMES80:

```

```

0ABA 6F       MOV     L, A

```


80x A :
Result in
H2

OK

Fφ
Fφ

```

0ABB 2600      MVI    H,00H
0ABD 29        DAD     H
0ABE 29        DAD     H
0ABF 29        DAD     H
0AC0 29        DAD     H
0AC1 54        MOV     D,H
0AC2 5D        MOV     E,L
0AC3 19        DAD     D
0AC4 19        DAD     D
0AC5 19        DAD     D
0AC6 19        DAD     D
0AC7 C9        RET

```

```

; SPACE80:
0AC8 0605      MVI    B,05H      ;FILL 80 SPACES AT HL.
0ACA C3340B    JMP     SPACE16

```

```

; SCROLL80:
0ACD 0605      MVI    B,05H      ;SCROLL 80 BYTES FROM HL TO DE

```

```

; SCROLL16:

```

```

0ACF 7A        MOV     A,D
0AD0 F6F8      ORI     0F8H
0AD2 57        MOV     D,A
0AD3 7C        MOV     A,H
0AD4 F6F8      ORI     0F8H
0AD6 67        MOV     H,A
0AD7 7E        MOV     A,M
0AD8 12        STAX   D
0AD9 23        INX   H
0ADA 13        INX   D
0ADB 7E        MOV     A,M
0ADC 12        STAX   D
0ADD 23        INX   H
0ADE 13        INX   D
0ADF 7E        MOV     A,M
0AE0 12        STAX   D
0AE1 23        INX   H
0AE2 13        INX   D
0AE3 7E        MOV     A,M
0AE4 12        STAX   D
0AE5 23        INX   H
0AE6 13        INX   D
0AE7 7E        MOV     A,M
0AE8 12        STAX   D
0AE9 23        INX   H
0AEA 13        INX   D
0AEB 7E        MOV     A,M
0AEC 12        STAX   D
0AED 23        INX   H
0AEE 13        INX   D
0AEF 7E        MOV     A,M
0AF0 12        STAX   D
0AF1 23        INX   H
0AF2 13        INX   D
0AF3 7E        MOV     A,M

```

```

0AF4 12 STAX D
0AF5 23 INX H
0AF6 13 INX D
0AF7 7E MOV A,M
0AF8 12 STAX D
0AF9 23 INX H
0AFA 13 INX D
0AFB 7E MOV A,M
0AFC 12 STAX D
0AFD 23 INX H
0AFE 13 INX D
0AFF 7E MOV A,M
0B00 12 STAX D
0B01 23 INX H
0B02 13 INX D
0B03 7E MOV A,M
0B04 12 STAX D
0B05 23 INX H
0B06 13 INX D
0B07 7E MOV A,M
0B08 12 STAX D
0B09 23 INX H
0B0A 13 INX D
0B0B 7E MOV A,M
0B0C 12 STAX D
0B0D 23 INX H
0B0E 13 INX D
0B0F 7E MOV A,M
0B10 12 STAX D
0B11 23 INX H
0B12 13 INX D
0B13 7E MOV A,M
0B14 12 STAX D
0B15 23 INX H
0B16 13 INX D
0B17+10B6 DJNZ SCROLL16
0B19 C9 DB 10H,SCROLL16-$-1
RET

```

15

SPACEDE:

```

0B1A 7D MOV A,L
0B1B E60F ANI 0FH
0B1D+2807 JRZ LB0221
0B1F 0E20 DB 28H,LB0221-$-1
0B21 71 MVI C,20H
0B22 23 MOV M,C
0B23 1B INX H
DCX D
JR SPACEDE
0B24+18F4 DB 18H,SPACEDE-$-1
0B26 0604 LB0221: MVI B,04H
0B28 7A LB0222: MOV A,D
0B29 1F RAR
0B2A 57 MOV D,A
0B2B 7B MOV A,E
0B2C 1F RAR

```

? How does this test work? it just turns off upper half byte & checks for zero Result.
 Turn off upper half and masked (set lower upper half to 0)
 don't jump if non-zero result
 is zero flag set?, yes=jump

move L into A (L had 00; see line 0778)
 move a 20 into C (20H=space)
 move it into memory as pointed to by H
 dec DE (the memory size) by 1
 Jump back & Loop
 move 04H into B

this writes blanks to
 but H must point to start of DISP.MEM!!
 I don't see that.

what is the purpose of these rotates?

```

0B2D 5F      MOV      E,A
              DJNZ     LB0222
0B2E+10F8    DB       10H, LB0222-$-1
0B30 7B      MOV      A,E
0B31 B7      ORA      A
0B32 C8      RZ
0B33 43      MOV      B,E

SPACE16:
0B34 0E20    MVI      C,20H
0B36 7C      LB0224: MOV     A,H
0B37 F6F8    ORI      0F8H
0B39 67      MOV     H,A
0B3A 71      MOV     M,C
0B3B 23      INX     H
0B3C 71      MOV     M,C
0B3D 23      INX     H
0B3E 71      MOV     M,C
0B3F 23      INX     H
0B40 71      MOV     M,C
0B41 23      INX     H
0B42 71      MOV     M,C
0B43 23      INX     H
0B44 71      MOV     M,C
0B45 23      INX     H
0B46 71      MOV     M,C
0B47 23      INX     H
0B48 71      MOV     M,C
0B49 23      INX     H
0B4A 71      MOV     M,C
0B4B 23      INX     H
0B4C 71      MOV     M,C
0B4D 23      INX     H
0B4E 71      MOV     M,C
0B4F 23      INX     H
0B50 71      MOV     M,C
0B51 23      INX     H
0B52 71      MOV     M,C
0B53 23      INX     H
0B54 71      MOV     M,C
0B55 23      INX     H
0B56 71      MOV     M,C
0B57 23      INX     H
0B58 71      MOV     M,C
0B59 23      INX     H
              DJNZ     LB0224
0B5A+10DA    DB       10H, LB0224-$-1
0B5C C9      RET

```

Zact? yes, Ret

HL = F800 ; start of Disp Mem = C000 + 1? Hardware however ignores A_{13,12}
This is probably OK for 2K screen mem. so F8 → becomes C8

*since B reg = 0 at start
 this thing loops 16 times*

437
B37

F800 + 16
HL = F816.
DEC B, JUMP NOT ZERO

```

;
;
RBUFEMP:      ;GET BYTE FROM RECEIVE BUFFER

```

```

0B5D 3AA140  LDA     RBUFCNT
0B60 B7      ORA     A
0B61 37      STC
0B62 C8      RZ
0B63 F3      DI
0B64 3AA140  LDA     RBUFCNT

```

```

0B67 3D      DCR      A
0B68 32A140  STA      RBUFCNT
0B6B 3AA040  LDA      RECBUFPNTR
0B6E 47      MOV      B,A
0B6F 210040  LXI      H,RECBUF
0B72 85      ADD      L
0B73 6F      MOV      L,A
0B74 78      MOV      A,B
0B75 3C      INR      A
0B76 E67F    ANI      7FH
0B78 32A040  STA      RECBUFPNTR
0B7B 7E      MOV      A,M
0B7C 37      STC
0B7D 3F      CMC
0B7E FB      EI
0B7F C9      RET

```

```

;
TBUFGET: ;GET BYTE FROM TRANSMIT BUFFER

```

```

0B80 3AA340  LDA      TBUFCNT
0B83 B7      ORA      A
0B84 37      STC
0B85 C8      RZ
0B86 3D      DCR      A
0B87 32A340  STA      TBUFCNT
0B8A 3AA240  LDA      TBUFPNTR
0B8D 47      MOV      B,A
0B8E 218040  LXI      H,TRANSBUF
0B91 85      ADD      L
0B92 6F      MOV      L,A
0B93 78      MOV      A,B
0B94 3C      INR      A
0B95 E61F    ANI      1FH
0B97 32A240  STA      TBUFPNTR
0B9A 7E      MOV      A,M
0B9B 37      STC
0B9C 3F      CMC
0B9D C9      RET

```

```

;
GETRECBYTE: ;WAIT FOR BYTE FROM RECEIVE BUFFER

```

```

0B9E C5      PUSH     B
0B9F D5      PUSH     D
0BA0 E5      PUSH     H
0BA1 CD0D00  LB0151: CALL    LB0002
0BA4 CD5D0B  CALL    RBUFEMP
          JRC    LB0151
0BA7+38F8  DB      38H,LB0151-$$-1
0BA9 E1      POP      H
0BAA D1      POP      D
0BAB C1      POP      B
0BAC C9      RET

```

```

;
KBUFEMP: ;GET KEY FROM KEY INTERRUPT BUFFER

```

```

0BAD C5      PUSH     B
0BAE E5      PUSH     H
0BAF F3      DI
0BB0 2AB440  LHLD    KBUFPNT

```

```

0BB3 7D      MOV      A,L
0BB4 FEA4    CPI      0A4H      ;BUFFER EMPTY?
0BB6 37      STC      ;YES.-->
                JRZ      LB0154
0BB7+2818    DB      28H,LB0154-$-1
0BB9 2B      DCX      H
0BBA 2B      DCX      H
0BBB 22B440  SHLD     KBUFPNT
0BBE 21A440  LXI      H,KEYBUF
0BC1 56      MOV      D,M
0BC2 23      INX      H
0BC3 5E      MOV      E,M
0BC4 23      INX      H
0BC5 D5      PUSH     D
0BC6 11A440  LXI      D,KEYBUF
0BC9 010E00  LXI      B,KBUFPNT-KEYBUF-2      ;KEY BUFFER SIZE-2
                LDIR
0BCC+EDB0    DB      0EDH,0B0H
0BCE 37      STC
0BCF 3F      CMC
0BD0 D1      POP      D
0BD1 E1      LB0154: POP      H
0BD2 C1      POP      B
0BD3 FB      EI
0BD4 C9      RET

                ;
                ; RECBUFIN: ; PUT BYTE IN RECEIVE BUFFER
0BD5 4F      MOV      C,A
0BD6 3AA140  LDA      RBUFCNT
0BD9 FE80    CPI      80H      ;BUFFER FULL?
0BDB 37      STC      ;YES.-->
0BDC CA9B03  JZ       BELL
0BDF 47      MOV      B,A
0BE0 3AA040  LDA      RECBUFPNTR
0BE3 80      ADD      B
0BE4 E67F    ANI      7FH
0BE6 210040  LXI      H,RECBUF
0BE9 85      ADD      L
0BEA 6F      MOV      L,A
0BEB 71      MOV      M,C
0BEC 78      MOV      A,B
0BED 3C      INR      A
0BEE 32A140  STA      RBUFCNT
0BF1 37      STC
0BF2 3F      CMC
0BF3 79      MOV      A,C
0BF4 C9      RET

                ;
                ; LB0174:
0BF5 4F      LB0174: MOV      C,A
0BF6 3AC940  LDA      MODEFLAG2
0BF9 E608    ANI      08H      ;ON-LINE?
                ;YES.-->
0BFB+2005    DB      20H,LB0175-$-1
0BFD 79      MOV      A,C
0BFE CDD50B  CALL     RECBUFIN
0C01 C9      RET

```

```

;
0C02 3ACA40 LB0175: LDA DIP1SAVE
0C05 E680 ANI 80H ;FULL DUPLEX?
0C07 79 MOV A,C
0C08 CCD50B CZ RECBUFIN ;NO.-->
0C0B 3AA340 LDA TBUFCNT
0C0E FE20 CPI 20H ;BUFFER FULL?
0C10 37 STC
JRZ LB0176 ;YES.-->
0C11+2815 DB 28H, LB0176-$-1
0C13 47 MOV B,A
0C14 3AA240 LDA TBUFPNTR
0C17 80 ADD B
0C18 E61F ANI 1FH
0C1A 218040 LXI H, TRANSBUF
0C1D 85 ADD L
0C1E 6F MOV L,A
0C1F 71 MOV M,C
0C20 78 MOV A,B
0C21 3C INR A
0C22 32A340 STA TBUFCNT
0C25 37 STC
0C26 3F CMC
0C27 79 MOV A,C
0C28 C9 LB0176: RET
;
0C29 3E1B OUTESC: MVI A,ESC
OUTPUTA: ;OUTPUT A TO COMM.
;ON/OFF LINE, HALF/FULL DUPLEX
0C2B F3 DI
0C2C CDF50B CALL LB0174
0C2F FB EI
0C30 C9 RET
;
CINSERTSCROLL: ;CHAR. INSERT SCROLL
0C31 3ABA40 LDA COLUMN
NEG
0C34+ED44 DB 0EDH, 44H
0C36 C64F ADI 4FH
0C38 C8 RZ
0C39 4F MOV C,A
0C3A 0600 MVI B,00H
0C3C 2AB840 LHLD BEGLINABS
0C3F 114F00 LXI D, LINELENGTH-1
0C42 19 DAD D
0C43 54 MOV D,H
0C44 5D MOV E,L
0C45 2B DCX H
0C46 7A LB0186: MOV A,D
0C47 F6F8 ORI 0F8H
0C49 57 MOV B,A
0C4A 7C MOV A,H
0C4B F6F8 ORI 0F8H
0C4D 57 MOV H,A
LDD
0C4E+EDA8 DB 0EDH, 0A8H
0C50 78 MOV A,B

```

0F0H

0C47 F6F8
0C4B F6F8

1/8
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76

```

0C51 B1      ORA      C
0C52+20F2   JRNZ     LB0186
0C54 2ABC40 LHL     CURSORABS
0C57 C9      RET
;
0C58 4F     LOOKUP: MOV    C,A      ;LOOKUP A IN TABLE
0C59 AE     LB0211: XRA   M      ;HL = TABLE
                                JRZ    LB0213   ;D = # OF ENTRYS
0C5A+280B   DB      28H, LB0213-$-1
0C5C 43     MOV    B,E      ;E = TABLE INCREMENT
0C5D 79     MOV    A,C      ;CARRY = 1 IF NOT FOUND.
0C5E 23     LB0212: INX   H
                                DJNZ   LB0212
0C5F+10FD   DB      10H, LB0212-$-1
0C61 15     DCR    D
                                JRNZ   LB0211
0C62+20F5   DB      20H, LB0211-$-1
0C64 79     MOV    A,C
0C65 37     STC
0C66 C9     RET
0C67 23     LB0213: INX   H
0C68 7E     MOV    A,M
0C69 C9     RET

```

does it match? yes jump

no

dec B, jump if Breg not zero no match increments H by contents of B

```

;
UPDATECRTC: ;UPDATE CRTC CURSOR SAVE
;AND ENABLE SYNC. INTERRUPT

```

```

0C6A 3ABB40 LDA    ROW
0C6D CDBA0A CALL  ATIMES80
0C70 3ABA40 LDA    COLUMN
0C73 5F     MOV    E,A
0C74 1600   MVI   D,00H
0C76 19     DAD   D
0C77 EB     XCHG
0C78 2AB640 LHL   TOPABS
0C7B 7C     MOV    A,H
0C7C E607   ANI   07H
0C7E 57     MOV    H,A
0C7F 19     DAD   D
0C80 7C     MOV    A,H
0C81 E63F   ANI   3FH
0C83 67     MOV    H,A
0C84 22C340 SHLD  CURSORCRTC

```

07H? 7/20/82

```

ENABSYNCINT:
0C87 3E00   MVI   A,00H
0C89 D364   OUT  CRTCIE ;ENABLE SYNC INTERRUPT
0C8B C9     RET

```

```

;
;
BYTETOCOMM:
0C8C F5     PUSH  PSW
0C8D DB45   LB0229: IN   UARTLS
0C8F E620   ANI   20H
                                JRZ   LB0229
0C91+28FA   DB      28H, LB0229-$-1
0C93 F1     POP   PSW
0C94 D340   OUT  UARTDL

```

0C96 C9

RET

; LINETOCOMM:

```

1 0C97 2650 MVI H,50H
2 0C99 E5 LB0231: PUSH H
3 0C9A 1A LDAX D
4 0C9B 4F MOV C,A
5 0C9C 13 INX D
6 0C9D 7A MOV A,D
7 0C9E F6F8 ORI 0F8H
8 0CA0 57 MOV D,A
9 0CA1 D5 PUSH D
10 0CA2 79 MOV A,C
11 0CA3 E67F ANI 7FH
12 0CA5 FE20 CPI 20H ;GRAPHIC CHAR?
13 JRC LB0232 ;YES.-->
14 0CA7+3808 DB 38H, LB0232-$-1
15 0CA9 FE7F CPI 7FH ;SPECIAL GRAPHIC?
16 JRNZ LB0236 ;NO.-->
17 0CAB+2027 DB 20H, LB0236-$-1
18 0CAD 3E5E MVI A,5EH
19 0CAF+180A JR LB0234
20 DB 18H, LB0234-$-1
21 ;
22 0CB1 FE1F LB0232: CPI 1FH ;SPECIAL CASE?
23 JRNZ LB0233 ;NO.-->
24 0CB3+2004 DB 20H, LB0233-$-1
25 0CB5 3E5F MVI A,5FH ;MAKE PRINTABLE
26 JR LB0234
27 0CB7+1802 DB 18H, LB0234-$-1
28 ;
29 0CB9 F660 LB0233: ORI 60H ;MAKE PRINTABLE
30 0CBB F5 LB0234: PUSH PSW
31 BIT 1,B ;GRAPHIC FLAG?
32 0CBC+CB48 DB 0CBH, 1*8+B+40H
33 JRNZ LB0238 ;YES.-->
34 0CBE+202B DB 20H, LB0238-$-1
35 SETB 1,B ;SET FLAG
36 0CC0+CBC8 DB 0CBH, 1*8+B+0C0H
37 ;
38 0CC2 CD4302 IF ANSI
39 CALL MODETEST ;HEATH MODE?
40 JRZ LB0235 ;YES.-->
41 0CC5+2807 DB 28H, LB0235-$-1
42 0CC7 CD250D CALL STRINGCOMMONLY
43 0CCA 5B3130ED DB '[10','M' OR 80H ;ANSI ENTER GRAPHICS
44 ENDIF
45 ;
46 0CCE CD250D LB0235: CALL STRINGCOMMONLY
47 0CD1 C6 DB 'F' OR 80H ;ENTER GRAPHICS MODE
48 JR LB0238
49 0CD2+1817 DB 18H, LB0238-$-1
50 ;
51 0CD4 F5 LB0236: PUSH PSW
52 BIT 1,B ;GRAPHIC FLAG?
53 0CD5+CB48 DB 0CBH, 1*8+B+40H

```

F0H



0F8H


```

0CD7+2812      JRZ      LB0238      ;NO.-->
                DB      28H,LB0238-$-1
                RES      1,B      ;RESET FLAG
0CD9+CB88      DB      0CBH,1*8+B+80H
                ;
                IF      ANSI
0CDB CD4302    CALL      MODETEST      ;HEATH MODE?
                JRZ      LB0237      ;YES.-->
0CDE+2807      DB      28H,LB0237-$-1
0CE0 CD250D    CALL      STRINGCOMMONLY
0CE3 5B3131ED DB      '[11','M' OR 80H      ;ANSI EXIT GRAPHICS
                ENDIF
                ;
0CE7 CD250D    LB0237: CALL      STRINGCOMMONLY
0CEA C7        DB      'G' OR 80H      ;EXIT GRAPHICS MODE
                LB0238: BIT      7,C      ;REVERSE VIDEO?
0CEB+CB79      DB      0CBH,7*8+C+40H
                JRZ      LB0240      ;NO.-->
0CED+2817      DB      28H,LB0240-$-1
                BIT      2,B      ;REVERSE VIDEO FLAG?
0CEF+CB50      DB      0CBH,2*8+B+40H
                JRNZ      LB0242      ;YES.-->
0CF1+2027      DB      20H,LB0242-$-1
                SETB      2,B      ;SET FLAG
0CF3+CBD0      DB      0CBH,2*8+B+0C0H
                ;
                IF      ANSI
0CF5 CD4302    CALL      MODETEST      ;HEATH MODE?
                JRZ      LB0239      ;YES.-->
0CF8+2806      DB      28H,LB0239-$-1
0CFA CD250D    CALL      STRINGCOMMONLY
0CFD 5B37ED    DB      '[7','M' OR 80H ;ANSI ENTER REVERSE VIDEO
                ENDIF
                ;
0D00 CD250D    LB0239 CALL      STRINGCOMMONLY
0D03 F0        DB      'P' OR 80H      ;ENTER REVERSE VIDEO MODE
                JR      LB0242
0D04+1814      DB      18H,LB0242-$-1
                ;
                LB0240: BIT      2,B      ;REVERSE VIDEO FLAG?
0D06+CB50      DB      0CBH,2*8+B+40H
                JRZ      LB0242      ;NO.-->
0D08+2810      DB      28H,LB0242-$-1
                RES      2,B      ;RESET FLAG
0D0A+CB90      DB      0CBH,2*8+B+80H
                ;
                IF      ANSI
0D0C CD4302    CALL      MODETEST      ;HEATH MODE?
                JRZ      LB0241      ;YES.-->
0D0F+2805      DB      28H,LB0241-$-1
0D11 CD250D    CALL      STRINGCOMMONLY
0D14 5BED      DB      '[','M' OR 80H ;ANSI EXIT REVERSE VIDEO
                ENDIF
                ;
0D16 CD250D    LB0241: CALL      STRINGCOMMONLY
0D19 F1        DB      'Q' OR 80H      ;EXIT REVERSE VIDEO MODE

```

```

0D1A F1      LB0242: POP      PSW
0D1B CD8C0C  CALL      BYTETOCOMM
0D1E D1      POP      D
0D1F E1      POP      H
0D20 25      DCR      H
0D21 C2990C  JNZ      LB0231
0D24 C9      RET

```

```

;
STRINGCOMMONLY:                ;ESC STRING TO COMM. ONLY
                                ;STRING FOLLOWS CALL AND
                                ;ENDS WITH MSB=1.

```

```

0D25 3E1B    MVI      A,ESC
0D27 CD8C0C  CALL      BYTETOCOMM
0D2A D1      POP      D
0D2B 1A      LB0245: LDAX   D
0D2C 13      INX      D
0D2D B7      ORA      A
0D2E FA360D  JM       LB0246
0D31 CD8C0C  CALL      BYTETOCOMM
JR          LB0245
0D34+18F5    DB      18H,LB0245-$-1

```

```

;
0D36 E67F    LB0246: ANI      7FH
0D38 CD8C0C  CALL      BYTETOCOMM
0D3B D5      PUSH   D
0D3C C9      RET

```

```

;*****

```

```

; J H E T E ; 52 4D
0D3D 4A48435445 DB 'JHCTERM'

```

What this USED FOR?

```

4000 =      RECBUF      EQU      4000H
4080 =      TRANSBUF   EQU      4080H
40A0 =      RECBUFPNTR EQU      40A0H
40A1 =      RBUFCNT    EQU      40A1H
40A2 =      TBUFPNTR   EQU      40A2H
40A3 =      TBUFCNT    EQU      40A3H
40A4 =      KEYBUF     EQU      40A4H
40B4 =      KBUFPNT    EQU      40B4H
40B6 =      TOPABS     EQU      40B6H
40B8 =      BEGLINABS  EQU      40B8H
40BA =      COLUMN     EQU      40BAH
40BB =      ROW        EQU      40BBH
40BC =      CURSORABS  EQU      40BCH
40BE =      CRTCROWS   EQU      40BEH
40BF =      CURSORTYPE EQU      40BFH
40C1 =      TOPCRTC    EQU      40C1H
40C3 =      CURSORCRTC EQU      40C3H
40C5 =      CURSORSAVE EQU      40C5H
40C7 =      MODEFLAG1  EQU      40C7H
40C8 =      DIP2SAVE   EQU      40C8H
40C9 =      MODEFLAG2  EQU      40C9H
40CA =      DIP1SAVE   EQU      40CAH
40CB =      HOLDCNT    EQU      40CBH
40CC =      ANSISAVE   EQU      40CCH
;
40CD =      TIMER      EQU      40CDH
40CE =      KSAVE      EQU      40CEH

```

```

4100 =      ;
            ; STACK          EQU      4100H
            ;
            ; *****
            ;
            ;          TABLES START HERE.          11/7/80
            ;
            ; *****
            ;
            ;          KEYBOARD TABLES START HERE.
KTABLE:
            ;          THIS TABLE IS FOR THE CHERRY (ADDS) KEYBOARD
            ;
            ;          START OF UN-SHIFTED TABLE.
0D44 34      DB      '4'      ;00
0D45 5F      DB      5FH      ;01 - UNDERLINE
0D46 75      DB      'U'      ;02
0D47 E9      DB      0E9H     ;03 - 9 KP
0D48 7A      DB      'Z'      ;04
0D49 2E      DB      '.'      ;05
0D4A 90      DB      90H      ;06 - F1
0D4B E1      DB      0E1H     ;07 - 1 KP
0D4C 5B      DB      '['      ;08
0D4D 00      DB      0        ;09 - BEL
0D4E E6      DB      0E6H     ;0A - 6 KP
0D4F 96      DB      96H      ;0B - F7
0D50 2D      DB      '-'      ;0C - '-' KP
0D51 6C      DB      'L'      ;0D
0D52 00      DB      0        ;0E
0D53 00      DB      0        ;0F
0D54 36      DB      '6'      ;10
0D55 67      DB      'G'      ;11
0D56 00      DB      0        ;12
0D57 00      DB      0        ;13
0D58 00      DB      0        ;14
0D59 5C      DB      5CH      ;15 - BACK SLASH
0D5A 8C      DB      8CH      ;16 - ERASE
0D5B 9A      DB      9AH      ;17 - RT. ARROW FUNCT.
0D5C 6A      DB      'J'      ;18
0D5D EA      DB      0EAH     ;19 - '.' KP
0D5E 93      DB      93H      ;1A - F4
0D5F 95      DB      95H      ;1B - F6
0D60 5E      DB      5EH      ;1C - UP ARROW CHAR.
0D61 69      DB      'I'      ;1D
0D62 E5      DB      0E5H     ;1E - 5 KP
0D63 78      DB      'X'      ;1F
0D64 68      DB      'H'      ;20
            ;
0D65 00      DB      0        ;21
0D66 00      DB      0        ;22
0D67 00      DB      0        ;23
0D68 00      DB      0        ;24
0D69 00      DB      0        ;25
0D6A 00      DB      0        ;26
0D6B 00      DB      0        ;27
0D6C 00      DB      0        ;28

```

```

0D6D 00      DB      0      ;29
0D6E 00      DB      0      ;2A
0D6F 00      DB      0      ;2B

;
0D70 9B      DB      9BH     ;2C - UP ARROW FUNCT.
0D71 35      DB      '5'     ;2D
0D72 70      DB      'P'     ;2E
0D73 77      DB      'W'     ;2F
0D74 38      DB      '8'     ;30
0D75 73      DB      'S'     ;31
0D76 E4      DB      0E4H    ;32 - 4 KP
0D77 5D      DB      'I'     ;33
0D78 6E      DB      'N'     ;34
0D79 3B      DB      ';'     ;35
0D7A 20      DB      20H     ;36 - SPACE BAR
0D7B 72      DB      'R'     ;37
0D7C 39      DB      '9'     ;38
0D7D 31      DB      '1'     ;39
0D7E 65      DB      'E'     ;3A
0D7F 37      DB      '7'     ;3B
0D80 2F      DB      '/'     ;3C
0D81 00      DB      0      ;3D
0D82 8E      DB      8EH     ;3E - LOCAL (OFF-LINE)
0D83 00      DB      0      ;3F
0D84 62      DB      'B'     ;40

;
0D85 00      DB      0      ;41
0D86 00      DB      0      ;42
0D87 00      DB      0      ;43
0D88 00      DB      0      ;44
0D89 00      DB      0      ;45
0D8A 00      DB      0      ;46
0D8B 00      DB      0      ;47
0D8C 00      DB      0      ;48
0D8D 00      DB      0      ;49
0D8E 00      DB      0      ;4A

;
0D8F 71      DB      'Q'     ;4B
0D90 6F      DB      'O'     ;4C
0D91 63      DB      'C'     ;4D
0D92 E8      DB      0E8H    ;4E - 8 KP
0D93 8F      DB      8FH     ;4F - AUX (SCROLL)
0D94 2C      DB      ','     ;50 - ',' KP

;
0D95 00      DB      0      ;51
0D96 00      DB      0      ;52
0D97 00      DB      0      ;53
0D98 00      DB      0      ;54
0D99 00      DB      0      ;55
0D9A 00      DB      0      ;56
0D9B 00      DB      0      ;57
0D9C 00      DB      0      ;58
0D9D 00      DB      0      ;59
0D9E 00      DB      0      ;5A

;
0D9F 09      DB      TAB     ;5B - TAB

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57


```

0DD4 26      DB      '&'      ;10
0DD5 47      DB      'G'      ;11
0DD6 00      DB      0         ;12
0DD7 00      DB      0         ;13
0DD8 00      DB      0         ;14
0DD9 7C      DB      7CH      ;15 - BROKEN VERT. BAR
0DDA 8C      DB      8CH      ;16 - ERASE
0DDB 00      DB      0         ;17 - RT. ARROW FUNCT.
0DDC 4A      DB      'J'      ;18
0DDD 2E      DB      '.'      ;19 - '.' KP
0DDE 00      DB      0         ;1A - F4
0DDF 00      DB      0         ;1B - F6
0DE0 7E      DB      7EH      ;1C - TILDA
0DE1 49      DB      'I'      ;1D
0DE2 E5      DB      0E5H     ;1E - 5 KP
0DE3 58      DB      'X'      ;1F
0DE4 48      DB      'H'      ;20
;
0DE5 00      DB      0         ;21
0DE6 00      DB      0         ;22
0DE7 00      DB      0         ;23
0DE8 00      DB      0         ;24
0DE9 00      DB      0         ;25
0DEA 00      DB      0         ;26
0DEB 00      DB      0         ;27
0DEC 00      DB      0         ;28
0DED 00      DB      0         ;29
0DEE 00      DB      0         ;2A
0DEF 00      DB      0         ;2B
;
0DF0 00      DB      0         ;2C - UP ARROW FUNCT.
0DF1 25      DB      '%'      ;2D
0DF2 50      DB      'P'      ;2E
0DF3 57      DB      'W'      ;2F
0DF4 28      DB      '('      ;30
0DF5 53      DB      'S'      ;31
0DF6 E4      DB      0E4H     ;32 - 4 KP
0DF7 7D      DB      7DH      ;33 - CLOSE BRACKET
0DF8 4E      DB      'N'      ;34
0DF9 2B      DB      '+'      ;35
0DFA 20      DB      20H      ;36 - SPACE BAR
0DFB 52      DB      'R'      ;37
0DFC 29      DB      ')'      ;38
0DFD 21      DB      21H      ;39 - EXCLAMATION
0DFE 45      DB      'E'      ;3A
0DFF 27      DB      27H      ;3B - ""
0E00 3F      DB      '?'      ;3C
0E01 00      DB      0         ;3D
0E02 8E      DB      8EH      ;3E - LOCAL (ON-LINE)
0E03 00      DB      0         ;3F
0E04 42      DB      'B'      ;40
;
0E05 00      DB      0         ;41
0E06 00      DB      0         ;42
0E07 00      DB      0         ;43
0E08 00      DB      0         ;44

```

1/8
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
70
71
72
73
74
75
76

0E09 00	DB	0	;45
0E0A 00	DB	0	;46
0E0B 00	DB	0	;47
0E0C 00	DB	0	;48
0E0D 00	DB	0	;49
0E0E 00	DB	0	;4A
;			
0E0F 51	DB	'Q'	;4B
0E10 4F	DB	'O'	;4C
0E11 43	DB	'C'	;4D
0E12 E8	DB	0E8H	;4E - 8 KP
0E13 8F	DB	8FH	;4F - AUX
0E14 00	DB	0	;50 - ', ' KP
;			
0E15 00	DB	0	;51
0E16 00	DB	0	;52
0E17 00	DB	0	;53
0E18 00	DB	0	;54
0E19 00	DB	0	;55
0E1A 00	DB	0	;56
0E1B 00	DB	0	;57
0E1C 00	DB	0	;58
0E1D 00	DB	0	;59
0E1E 00	DB	0	;5A
;			
0E1F 09	DB	TAB	;5B - TAB
0E20 00	DB	0	;5C - F8
0E21 00	DB	0	;5D - DOWN ARROW FUNCT.
0E22 0D	DB	CR	;5E - ENTER KP
0E23 00	DB	0	;5F - F5
0E24 00	DB	0	;60
0E25 46	DB	'F'	;61
0E26 8D	DB	8DH	;62 - BREAK
0E27 4B	DB	'K'	;63
0E28 0A	DB	LF	;64 - LINE FEED
0E29 1B	DB	ESC	;65 - ESCAPE
0E2A 3C	DB	'<'	;66
0E2B 7F ⁰⁸	DB	7FH ^{08H}	;67 - DELETE <i>Backspace</i>
0E2C E2	DB	0E2H	;68 - 2 KP
0E2D 00	DB	0	;69 - 0 KEY
0E2E 59	DB	'Y'	;6A
0E2F 2A	DB	'*'	;6B
0E30 23	DB	'#'	;6C
0E31 56	DB	'V'	;6D
0E32 0D	DB	CR	;6E - NEW LINE
0E33 22	DB	'"'	;6F
0E34 41	DB	'A'	;70
0E35 44	DB	'D'	;71
0E36 4D	DB	'M'	;72
0E37 00	DB	0	;73 - HOME
0E38 60	DB	60H	;74
0E39 54	DB	'T'	;75
0E3A 00	DB	0	;76 - LEFT ARROW FUNCT.
0E3B E7	DB	0E7H	;77 - 7 KP
0E3C E3	DB	0E3H	;78 - 3 KP
0E3D 30	DB	'0'	;79 - 0 KP

E22

will probably assign a separate key for delete

shifted del, bs = actual 08 H. 10/6/84

0E3E 00 DB 0 ;7A - F3

0E3F 00 DB 0 ;7B

0E40 00 DB 0 ;7C

0E41 00 DB 0 ;7D

0E42 00 DB 0 ;7E

0E43 00 DB 0 ;7F - F2

;
;
; END OF MAIN TABLE.
; *****
; FUNCTION KEY TABLES
; UN-SHIFTED

FUNTAB: ;MSB = 0 IF PRINTABLE.
; MAIN TABLE VALUE B4-B0

0E44 30 DB '0' ;KEYPAD 0 00

0E45 31 DB '1' ; 1 01

0E46 32 DB '2' ; 2 02

0E47 33 DB '3' ; 3 03

0E48 34 DB '4' ; 4 04

0E49 35 DB '5' ; 5 05

0E4A 36 DB '6' ; 6 06

0E4B 37 DB '7' ; 7 07

0E4C 38 DB '8' ; 8 08

0E4D 39 DB '9' ; 9 09

0E4E 2E DB '.' ; . 0A

0E4F 0D DB CR ; ENTER 0B

0E50 E2 DB 0E2H ;ERASE 0C

0E51 E4 DB 0E4H ;BREAK 0D

0E52 E5 DB 0E5H ;LOCAL 0E

0E53 E1 DB 0E1H ;AUX SCROLL 0F

0E54 A0 DB 0A0H ;F1 10

0E55 A1 DB 0A1H ;F2 11

0E56 A2 DB 0A2H ;F3 12

0E57 A3 DB 0A3H ;F4 13

0E58 A4 DB 0A4H ;F5 14

0E59 A5 DB 0A5H ;F6 15

0E5A A6 DB 0A6H ;F7 16

0E5B A7 DB 0A7H ;F8 17

0E5C C1 DB 0C1H ;DOWN ARROW 18

0E5D C3 DB 0C3H ;LEFT ARROW 19

0E5E C5 DB 0C5H ;RIGHT ARROW 1A

0E5F C6 DB 0C6H ;UP ARROW 1B

0E60 C4 DB 0C4H ;HOME 1C

;
;
; SHIFTED FUNCTIONS
; FUNSHIFTTAB:

0E61 00 DB 0 ;KEYPAD 0

0E62 C0 DB 0C0H ; 1 INSERT LINE

0E63 C1 DB 0C1H ; 2 CURSOR DOWN

0E64 C2 DB 0C2H ; 3 DELETE LINE

0E65 C3 DB 0C3H ; 4 CURSOR LEFT

0E66 C4 DB 0C4H ; 5 CURSOR HOME

0E67 C5 DB 0C5H ; 6 CURSOR RIGHT

0E68 E0 DB 0E0H ; 7 INSERT CHAR. TOGGLE


```
0E69 C6 DB 0C6H ; 8 CURSOR UP
0E6A E3 DB 0E3H ; 9 DELETE CHAR.
0E6B 00 DB 0 ; .
0E6C 00 DB 0 ; ENTER
```

```
0E6D E2 DB 0E2H ; ERASE
0E6E E4 DB 0E4H ; BREAK
0E6F E5 DB 0E5H ; LOCAL
0E70 E1 DB 0E1H ; AUX SCROLL
; DB 0 ; F1 PRESENTLY UNASSIGNED
; DB 0 ; F2
; DB 0 ; F3
; DB 0 ; F4
; DB 0 ; F5
; DB 0 ; F6
; DB 0 ; F7
; DB 0 ; F8
; DB 0 ; DOWN ARROW
; DB 0 ; LEFT ARROW
; DB 0 ; RIGHT ARROW
; DB 0 ; UP ARROW
; DB 0 ; HOME
```

ALTERNATE FUNCTIONS

ALTTAB:

```
0E71 80 DB 80H ; KEYPAD 0
0E72 81 DB 81H ; 1
0E73 82 DB 82H ; 2
0E74 83 DB 83H ; 3
0E75 84 DB 84H ; 4
0E76 85 DB 85H ; 5
0E77 86 DB 86H ; 6
0E78 87 DB 87H ; 7
0E79 88 DB 88H ; 8
0E7A 89 DB 89H ; 9
0E7B 8A DB 8AH ; .
0E7C 8B DB 8BH ; ENTER
```

```
; DB 0 ; ERASE
; DB 0 ; BREAK
; DB 0 ; LOCAL
; DB 0 ; AUX
; DB 0 ; F1 PRESENTLY UNASSIGNED
; DB 0 ; F2
; DB 0 ; F3
; DB 0 ; F4
; DB 0 ; F5
; DB 0 ; F6
; DB 0 ; F7
; DB 0 ; F8
; DB 0 ; DOWN ARROW
; DB 0 ; LEFT ARROW
; DB 0 ; RIGHT ARROW
; DB 0 ; UP ARROW
; DB 0 ; HOME
```

16
0001 0110

0110 1000

```

; TYPE 1 FUNCTIONS DATA TABLE
;
TYPE1TAB: ;ALT. KEYPAD CODES
0E7D F0 DB 'P' OR 80H ;0
0E7E F1 DB 'Q' OR 80H ;1
0E7F F2 DB 'R' OR 80H ;2
0E80 F3 DB 'S' OR 80H ;3
0E81 F4 DB 'T' OR 80H ;4
0E82 F5 DB 'U' OR 80H ;5
0E83 F6 DB 'V' OR 80H ;6
0E84 F7 DB 'W' OR 80H ;7
0E85 F8 DB 'X' OR 80H ;8
0E86 F9 DB 'Y' OR 80H ;9
0E87 EE DB 'N' OR 80H ;.
0E88 CD DB 'M' OR 80H ;ENTER

```

```

; TYPE 2 FUNCTIONS DATA TABLE
;
TYPE2TAB: ;F1 - F8 CODES
0E89 D3 DB 'S' OR 80H
0E8A D4 DB 'T' OR 80H
0E8B D5 DB 'U' OR 80H
0E8C D6 DB 'V' OR 80H
0E8D D7 DB 'W' OR 80H
0E8E D0 DB 'P' OR 80H
0E8F D1 DB 'Q' OR 80H
0E90 D2 DB 'R' OR 80H

```

```

; TYPE 3 FUNCTIONS DATA TABLE
;
TYPE3TAB: ;SHIFTED PAD (EXCEPT TYPE 4)
0E91 CC DB 'L' OR 80H ;1
0E92 C2 DB 'B' OR 80H ;2
0E93 CD DB 'M' OR 80H ;3
0E94 C4 DB 'D' OR 80H ;4
0E95 C8 DB 'H' OR 80H ;5
0E96 C3 DB 'C' OR 80H ;6
0E97 C1 DB 'A' OR 80H ;8

```

```

; TYPE 4 FUNCTIONS ROUTINE ADDR. TABLE
;
TYPE4TAB:
0E98 4902 DW INSERTKEY ;PAD 7 SHIFT
0E9A 6302 DW SCROLKEY ;AUX
0E9C 6E02 DW ERASEKEY ;ERASE
0E9E 7802 DW DELCHARKEY ;PAD 9 SHIFT
0EA0 6D02 DW BREAKRET ;BREAK (RETURN ONLY)

;
IF MATRIXLOCAL OR MATRIXSHIFTLOCK
0EA2 7E02 DW OFFLINE ;LOCAL (CAPS. LOCK)
ENDIF ;DUAL FUNCTION

```

```

; TYPE 4 ROUTINES DATA TABLES
;
INSERTAB:
0EA4 C0 DB 40H OR 80H ;HEATH INSERT ON

```

```

0EA5 34E8      DB      '4','H' OR 80H      ;ANSI
0EA7 CF        DB      'O' OR 80H        ;HEATH INSERT OFF
0EA8 34EC      DB      '4','L' OR 80H      ;ANSI
;
0EAA CA        ERATAB: DB      'J' OR 80H      ;HEATH ERASE EOP
0EAB CA        DB      'J' OR 80H        ;ANSI
0EAC C5        DB      'E' OR 80H        ;HEATH CLEAR & HOME
0EAD 32CA      DB      '2','J' OR 80H      ;ANSI
;
0EAF CE        DELTAB: DB      'N' OR 80H      ;HEATH DELETE CHAR.
0EB0 D0        DB      'P' OR 80H        ;ANSI
;
;          END OF KEYBOARD TABLES
;
;*****
;
;          HEATH MODE TABLE
HEATHTAB:
0EB1 1B        DB      ESC
0EB2 F803      DW      SETESCFLAG
;
IF      ANSI
0EB4 3C        DB      '<'          ;ENTER ANSI MODE
0EB5 0104      DW      ENTERANSI
ENDIF
;
0EB7 23        DB      '#'          ;TRANSMIT PAGE
0EB8 2C04      DW      TRANSPAGE
0EBA 40        DB      '@'          ;ENTER INSERT CHAR. MODE
0EBB 3E04      DW      INSERTCHARON
0EBD 41        DB      'A'          ;CURSOR UP
0EBE 4204      DW      CURSUP
0EC0 42        DB      'B'          ;CURSOR DOWN
0EC1 8904      DW      CURSDOWN
0EC3 43        DB      'C'          ;CURSOR FORWARD
0EC4 DC04      DW      CURSFORWARD
0EC6 44        DB      'D'          ;CURSOR BACK (BACKSPACE)
0EC7 9E03      DW      BACKSPACE
0EC9 45        DB      'E'          ;CLEAR SCREEN & HOME CURSOR
0ECA FC04      DW      CLEARHOME
0ECC 46        DB      'F'          ;ENTER GRAPHICS MODE
0ECD 0C05      DW      SETGRAPHICS
0ECF 47        DB      'G'          ;EXIT GRAPHICS MODE
0ED0 1005      DW      RESETGRAPHICS
0ED2 48        DB      'H'          ;HOME CURSOR
0ED3 1405      DW      CURSHOME
0ED5 49        DB      'I'          ;REVERSE LINEFEED
0ED6 4704      DW      REVLFEED
0ED8 4A        DB      'J'          ;ERASE TO END OF PAGE
0ED9 2405      DW      ERASEEOP
0EDB 4B        DB      'K'          ;ERASE TO END OF LINE
0EDC 4605      DW      ERASEEOL
0EDE 4C        DB      'L'          ;INSERT LINE
0EDF 5605      DW      INSERTLINE
0EE1 4D        DB      'M'          ;DELETE LINE
0EE2 9F05      DW      DELLINE

```

```

0EE4 4E      DB      'N'          ;DELETE CHARACTER
0EE5 C905    DW      DELCHAR
0EE7 4F      DB      'O'          ;EXIT INSERT CHAR. MODE
0EE8 EF05    DW      INSERTCHAROFF
0EEA 59      DB      'Y'          ;DIRECT CURSOR ADDRESS
0EEB 0006    DW      CURSADDR
0EED 5A      DB      'Z'          ;IDENTIFY AS VT52
0EEE F305    DW      IDENTVT52
0EF0 5B      DB      '['          ;ENTER HOLD SCREEN MODE
0EF1 E007    DW      SETHOLDSCREEN
0EF3 5C      DB      '\ '        ;EXIT HOLD SCREEN MODE
0EF4 E907    DW      RESETHOLDSCREEN
0EF6 5D      DB      ']'          ;TRANSMIT 25TH LINE
0EF7 0904    DW      TRANS25TH
0EF9 62      DB      'B'          ;ERASE BEGINNING OF SCREEN
0EFA 3706    DW      ERASEBEGDIS
0EFC 6A      DB      'J'          ;SAVE CURSOR POSITION
0EFD 6306    DW      SAVECURS
0EFF 6B      DB      'K'          ;SET CURSOR TO SAVED POSITION
0F00 6D06    DW      SETCURS
0F02 6C      DB      'L'          ;ERASE LINE
0F03 9E06    DW      ERASELINE
0F05 6E      DB      'N'          ;CURSOR POSITION REPORT
0F06 A406    DW      CURSPOSREPORT
0F08 6F      DB      'O'          ;ERASE FROM BEGINNING OF LINE
0F09 5106    DW      ERASEBOL
0F0B 70      DB      'P'          ;ENTER REVERSE VIDEO MODE
0F0C BA06    DW      SETREVVIDEO
0F0E 71      DB      'Q'          ;EXIT REVERSE VIDEO MODE
0F0F C006    DW      RESETREVVIDEO
0F11 72      DB      'R'          ;MODIFY BAUD RATE
0F12 C406    DW      MODBAUDRATE
0F14 76      DB      'V'          ;WRAP AT END OF LINE
0F15 1607    DW      SETLINEWRAP
0F17 77      DB      'W'          ;DISCARD AT END OF LINE
0F18 1B07    DW      SETNOWRAP
0F1A 78      DB      'X'          ;SET MODE
0F1B 2507    DW      SETMODE
0F1D 79      DB      'Y'          ;RESET MODE
0F1E 2007    DW      RESETMODE
0F20 7A      DB      'Z'          ;RESET TO POWER UP CONFIG.
0F21 4B07    DW      REINIT

;
IF      ENABDISKBOARD
      DB      7BH          ;OPEN BRACKET
      DW      ENABKBOARD    ;ENABLE KEYBOARD
      DB      7DH          ;CLOSE BRACKET
      DW      DISABKBOARD   ;DISABLE KEYBOARD
ENDIF

;
;      END OF HEATH MODE TABLE
;
;*****
IF      ANSI
;
;      ANSI MODE TABLE

```

ANSITAB:

0F23	3D	DB	'='	;ENTER ALT. KEYPAD MODE
0F24	3808	DW	SETALTPAD	
0F26	3E	DB	'>'	;EXIT ALT. KEYPAD MODE
0F27	3D08	DW	RESETALTPAD	
0F29	4D	DB	'M'	;REVERSE LINEFEED
0F2A	4704	DW	REVLFEED	
0F2C	5B	DB	'['	;STRING LEAD-IN CHAR.
0F2D	6008	DW	ANSIBRACKET	

;
; ANSI "ESC [" LOOKUP TABLE
ANSITAB2:

0F2F	3E	DB	'>'	
0F30	7708	DW	SETRESETMODE	
0F32	3F	DB	'?'	
0F33	AA08	DW	CHANGEMODE	
0F35	41	DB	'A'	
0F36	E708	DW	ANSICURSUP	
0F38	42	DB	'B'	
0F39	FA08	DW	ANSICURSDOWN	
0F3B	43	DB	'C'	
0F3C	0D09	DW	ANSICURSFOR	
0F3E	44	DB	'D'	
0F3F	1A09	DW	ANSICURSBACK	
0F41	48	DB	'H'	
0F42	2C09	DW	ANSICURSADDR	
0F44	4A	DB	'J'	
0F45	ED04	DW	ANSIERASEDIS	
0F47	4B	DB	'K'	
0F48	8E06	DW	ANSIERASELINE	
0F4A	4C	DB	'L'	
0F4B	5B09	DW	ANSIINSERTLINE	
0F4D	4D	DB	'M'	
0F4E	6E09	DW	ANSIDELLINE	
0F50	50	DB	'P'	
0F51	8109	DW	ANSIDELCHAR	
0F53	66	DB	'F'	
0F54	2C09	DW	ANSICURSADDR	
0F56	68	DB	'H'	
0F57	9409	DW	ANSIMODMODE	
0F59	6C	DB	'L'	
0F5A	B309	DW	ANSIMODMODE2	
0F5C	6D	DB	'M'	
0F5D	D209	DW	ANSIDISPLAYMODE	
0F5F	6E	DB	'N'	
0F60	F509	DW	ANSICURSREPORT	
0F62	70	DB	'P'	
0F63	2904	DW	ANSITRANS PAGE	
0F65	71	DB	'Q'	
0F66	0604	DW	ANSITRANS25	
0F68	72	DB	'R'	
0F69	470A	DW	ANSIMODBAUD	
0F6B	73	DB	'S'	
0F6C	6006	DW	ANSISAVECURS	
0F6E	75	DB	'U'	
0F6F	6A06	DW	ANSICURSTOSAVE	

```

0F71 7A      DB      'Z'
0F72 4807    DW      ANSIREINIT

```

```

;
;
;          ANSI ESC [ M TABLE
;
ANSITAB3:

```

```

0F74 00      DB      0
0F75 C006    DW      RESETREVVIDEO
0F77 07      DB      07H
0F78 BA06    DW      SETREVVIDEO
0F7A 0A      DB      0AH
0F7B 0C05    DW      SETGRAPHICS
0F7D 0B      DB      0BH
0F7E 1005    DW      RESETGRAPHICS

```

```

;
;          END OF ANSI TABLES
;
ENDIF
;
; *****
;

```

```

;
;          COMMON TABLE
;
COMMONTAB:

```

```

0F80 07      DB      BEL
0F81 9B03    DW      BELL
0F83 08      DB      BSPACE
0F84 9E03    DW      BACKSPACE
0F86 09      DB      TAB
0F87 B303    DW      HORIZTAB
0F89 0A      DB      LF
0F8A D803    DW      LINEFEED
0F8C 0C      DB      FF
0F8D FC04    DW      CLEARHOME
0F8F 0D      DB      CR
0F90 E403    DW      CRETURN
0F92 1B      DB      ESC
0F93 F803    DW      SETESCFLAG

```

```

;
;          END OF COMMON TABLE
;
; *****
;
;          BAUD RATE TABLE          1.536 OR 3.072 MHZ.
;

```

```

BAUDTAB:
IF          CPUCLOCK          ;          DIPSWITCH          MODIFY
          3.072 MHZ TABLE

```

```

0F95 D106    DW      1745      ;110      0          A
0F97 0005    DW      1280      ;150      -          B
0F99 8002    DW      640       ;300      1          C
0F9B 8002    DW      640       ;300      -          D
0F9D 4001    DW      320        ;600      2          E
0F9F 6B00    DW      107         ;1800     -          F
0FA1 A000    DW      160         ;1200     3          G
0FA3 6000    DW      96          ;2000     -          H

```

```

0FA5 5000      DW      80      ;2400    4      I
0FA7 3500      DW      53      ;3600    -      J
0FA9 2800      DW      40      ;4800    5      K
0FAB 1B00      DW      27      ;7200    -      L
0FAD 1400      DW      20      ;9600    6      M
0FAF 1400      DW      20      ;9600    -      N
0FBI 0A00      DW      10      ;19200   7      O

```

```

ELSE
;

```

1.536 MHZ TABLE

```

DW      873     ;110     0      A
DW      640     ;150     -      B
DW      320     ;300     1      C
DW      320     ;300     -      D
DW      160     ;600     2      E
DW      53      ;1800    -      F
DW      80      ;1200    3      G
DW      48      ;2000    -      H
DW      40      ;2400    4      I
DW      27      ;3600    -      J
DW      20      ;4800    5      K
DW      13      ;7200    -      L
DW      10      ;9600    6      M
DW      10      ;9600    -      N
DW      5       ;19200   7      O

```

```

ENDIF
;
;

```

SET MODE TABLE

```

SETTAB: DW      SETFULLDUP      ;0
0FB3 9D07      DW      ENAB25TH              ;1
0FB5 AD07      DW      ENABCLICK        ;2
0FB7 DB07      DW      SETHOLDSCREEN      ;3
0FB9 E007      DW      SETBLOCKCURS       ;4
0FBB ED07      DW      CURSON              ;5
0FBD 1B08      DW      SETPADSHIFT      ;6
0FBF 2E08      DW      SETALTPAD        ;7
0FC1 3808      DW      AUTOLFON         ;8
0FC3 4208      DW      AUTOCRON         ;9
0FC5 4C08      DW      AUTOREPEAT      ;:
0FC7 5608

```

```

;
;

```

RESET MODE TABLE

```

RESETTAB:

```

```

0FC9 A607      DW      RESETFULLDUP      ;0
0FCB CA07      DW      DISAB25TH        ;1
0FCD D607      DW      DISABCLICK      ;2
0PCF E907      DW      RESETHOLDSCREEN   ;3
0FD1 FD07      DW      SETULINECURS     ;4
0FD3 0B08      DW      DISABCURS        ;5
0FD5 3308      DW      RESETPADSHIFT    ;6
0FD7 3D08      DW      RESETALTPAD     ;7
0FD9 4708      DW      AUTOLFOPF       ;8
0FDB 5108      DW      AUTOCROFF       ;9
0FDD 5B08      DW      NOAUTOREPEAT    ;:

```

```

;

```

12/22/84 change CRT to use ext Horiz syne

```

;
; CRTC INITIALIZATION TABLE 1.
CRTCTAB1:
0FDF 5D DB 5DH ;R0 = HORIZ. TOTAL-1 94
0FE0 50 DB 50H ;R1 = HORIZ. DISPLAYED 20
0FE1 51 52 56 DB 51H 52 ;R2 = HORIZ. SYNC. POSITION
0FE2 08 05 09 DB 08H 05 ;R3 = HORIZ. SYNC. WIDTH
0FE3 19 DB 19H ;R4 = VERT. TOTAL-1
0FE4 18 0B 05 DB 10H 0B ;R5 = VERT. SYNC. ADJUST
0FE5 18 DB 18H ;R6 = VERT. DISPLAYED
0FE6 18 19 19 DB 18H ;R7 = VERT. SYNC. POSITION
0FE7 00 DB 0 ;R8 = INTERLACE MODE
0FE8 09 DB 9 ;R9 = MAX. SCAN LINE-1
0FE9 48 DB 48H ;R10= CURSOR START SCAN/BLINK RATE
0FEA 08 DB 8 ;R11= CURSOR END SCAN
0FEB 00 DB 0 ;R12= TOP OF PAGE (HIGH)
0FEC 00 DB 0 ;R13= TOP OF PAGE (LOW)
0FED 00 DB 0 ;R14= CURSOR ADDR. (HIGH)
0FEE 00 DB 0 ;R15= CURSOR ADDR> (LOW)

```

```

;
; CRTC TABLE 2.
CRTCTAB2:
0FEF 5D DB 5DH ;R0 = HORIZ. TOTAL-1
0FF0 50 DB 50H ;R1 = HORIZ. DISPLAYED
0FF1 51 DB 51H ;R2 = HORIZ. SYNC. POSITION
0FF2 08 DB 08H ;R3 = HORIZ. SYNC. WIDTH
0FF3 19 DB 19H ;R4 = VERT. TOTAL-1
0FF4 10 DB 10H ;R5 = VERT. SYNC. ADJUST
0FF5 18 DB 18H ;R6 = VERT. DISPLAYED
0FF6 18 DB 18H ;R7 = VERT. SYNC. POSITION
0FF7 03 DB 3 ;R8 = INTERLACE MODE
0FF8 09 DB 9 ;R9 = MAX. SCAN LINE-1
0FF9 48 DB 48H ;R10= CURSOR START SCAN/BLINK RATE
0FFA 08 DB 8 ;R11= CURSOR END SCAN
0FFB 00 DB 0 ;R12= TOP OF PAGE (HIGH)
0FFC 00 DB 0 ;R13= TOP OF PAGE (LOW)
0FFD 00 DB 0 ;R14= CURSOR ADDR. (HIGH)
0FFE 00 DB 0 ;R15= CURSOR ADDR> (LOW)

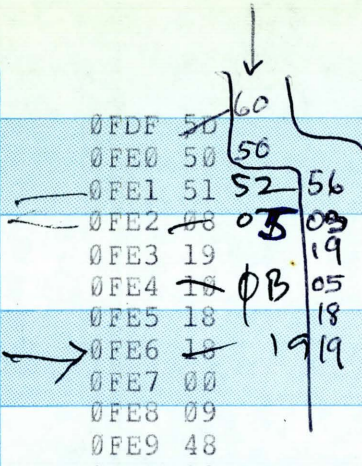
```

0FFF ; END

changed to 0B 8/31/82

This gets the vert. freq a lot closer to 60Hz

This fixes 25th line problem



0A8A LB0195	0A90 LB0196	0728 LB0203	0C59 LB0211	0C5E LB0212
0C67 LB0213	03C8 LB0215	03CC LB0216	0B26 LB0221	0B28 LB0222
0B36 LB0224	0421 LB0227	0C8D LB0229	0C99 LB0231	0CB1 LB0232
0CB9 LB0233	0CBB LB0234	0CCE LB0235	0CD4 LB0236	0CE7 LB0237
0CEB LB0238	0D00 LB0239	0D06 LB0240	0D16 LB0241	0D1A LB0242
0434 LB0243	0D2B LB0245	0D36 LB0246	000A LF	03D8 LINEFEED
0050 LINELENGTH	0C97 LINETOCOMM	028F LOC1	028C LOC2	0296 LOC3
0C58 LOOKUP	0003 MAINLOOP	0001 MATRIXBREAK		
0001 MATRIXLOCAL		0001 MATRIXSHIFTLOCK		
06C4 MODBAUDRATE		40C7 MODEFLAG1	40C9 MODEFLAG2	0243 MODETEST
0038 MSKINT	0066 NMIINT	085B NOAUTOREPEAT		0176 NOSHIFT
0260 NOTANSI	00F6 NOTRANS	01A4 NOTSHIFT	00DD NOXOFF	
0018 NUMBEROFLINES		027E OFFLINE	026C ONELF	0215 OUT1A
01FF OUTA	0C29 OUTESC	0206 OUTIT	01F7 OUTP1	01DB OUTPRINT
020D OUTPUT1	021D OUTPUT2	0229 OUTPUT3	0C2B OUTPUTA	0331 OUTPUTXON
0199 PADOK	0001 PIO1C	0000 PIO1D	0003 PIO2C	0002 PIO2D
0400 RAMSIZE	40A1 RBUFNCNT	0B5D RBUFEMP	0BD5 RECBUFIN	4000 RECBUF
40A0 RECBUFNTR	00B3 RECEIVEINT	074B REINIT	0004 REPRATE	015F RESBREAK
083D RESETALTPAD		07A6 RESETFULLDUP		
0510 RESETGRAPHICS		07E9 RESETHOLDSCREEN		0720 RESETMODE
0833 RESETPADSHIFT		06C0 RESETREVVIDEO		0FC9 RESETTAB
0447 REVLFEED	40BB ROW	0663 SAVECURS	0780 SCREENSIZE	0263 SCROLKEY
0ACF SCROLL16	0ACD SCROLL80	004E SCROLLKEY	0838 SETALTPAD	06D9 SETBAUD
07ED SETBLOCKCURS		0166 SETBREAK	066D SETCURS	07A2 SETDUP
03F8 SETESCFLAG	079D SETFULLDUP	050C SETGRAPHICS		
0A96 SETHEATHMODE		07E0 SETHOLDSCREEN		
0716 SETLINEWRAP		0725 SETMODE	071B SETNOWRAP	
082E SETPADSHIFT		0877 SETRESETMODE		
06BA SETREVVIDEO		07C6 SETROWS	0FB3 SETTAB	
07FD SETULINECURS		0B34 SPACE16	0AC8 SPACE80	0B1A SPACEDE
4100 STACK	0D25 STRINGCOMMONLY	0009 TAB	40A3 TBUFNCNT	
00E5 TBUFEMPINT	0B80 TBUFGET	40A2 TBUFNTR	0138 TIMEOUT	40CD TIMER
40B6 TOPABS	40C1 TOPCRTC	0409 TRANS25TH	4080 TRANSBUF	042C TRANSPAGE
0258 TURNON	0209 TYPE1	0E7D TYPE1TAB	0219 TYPE2	0E89 TYPE2TAB
0225 TYPE3	0231 TYPE3LOOP	0E91 TYPE3TAB	01D1 TYPE4	0E98 TYPE4TAB
0041 UARTDH	0040 UARTDL	0042 UARTID	0043 UARTLC	0045 UARTLS
0044 UARTMC	0046 UARTMS	0805 ULINECURS	0C6A UPDATECRTC	

1/6
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76