```
;
; Test Program to interact with LAVA-10 VGA Board
;===============================================================================
;
;      V0.1   4/27/2012    ;Initial Program


LavaStat      EQU    090H   ;Status Port
LavaData      EQU    091H   ;Data port

                            ;Equates for display on SD Systems Video Board (Used In CPM Debugging mode only)
SCROLL        EQU    01H    ;Set scrool direction UP.
LF            EQU    0AH
CR            EQU    0DH
BS            EQU    08H    ;Back space (required for sector display)
BELL          EQU    07H
SPACE         EQU    20H
QUIT          EQU    11H    ;Turns off any screen enhancements (flashing, underline etc).
NO$ENHANCEMENT       EQU    17H    ;Turns off whatever is on
FAST          EQU    10H    ;High speed scrool
TAB           EQU    09H    ;TAB ACROSS (8 SPACES FOR SD-BOARD)
ESC           EQU    1BH
CLEAR         EQU    1CH    ;SD Systems Video Board, Clear to EOL. (Use 80 spaces if EOL not available
                            ;on other video cards)

;LAVA Commands:-
COPY$MEMORY   EQU    010H
WRITE$MEMORY  EQU    020H
READ$MEMORY   EQU    030H
DRAW$TEXT     EQU    040H
READ$CSR      EQU    036H
WRITE$CSR     EQU    022H


RDCON         EQU    1      ;For CP/M I/O
WRCON         EQU    2
PRINT         EQU    9
CONST         EQU    11     ;CONSOLE STAT
BDOS          EQU    5

FALSE  EQU    0
TRUE   EQU    NOT FALSE



;-------------------------------------------------------------------------------------------
       ORG    100H          ;<--- For CPM

begin:
       LXI    SP,STACK
```

```
       LXI     D,SIGN$ON   ;print a welcome message
       CALL    PSTRING
       MVI     A,00000000B ;Set to read mode, no strobes etc
       OUT     LavaStat    ;Send to lava status port (91H)

NEXT$CHAR:
       CALL    ZCI         ;Get a keyboard character
       MOV     C,A
       CPI     ESC
       JZ      ABORT
       CPI     '@'
       JNZ     SKIP

       LXI     D,2000
       MVI     C,'1'       ;Test Rapid sending of characters
StrTest:PUSH   B
       PUSH    D
       CALL    LOUT        ;Only [A] is changed
       CALL    printparm$DE
       POP     D
       POP     B

       DCX     D
       MOV     A,E
       ORA     D
       JNZ     StrTest
       LXI     H,0
       CALL    PUT$X
       LXI     H,0
       CALL    PUT$Y
       INR     C           ;Next character
       LXI     D,2000
       JMP     StrTest


SKIP:  CALL    LOUT        ;Only [A] is changed
       CALL    ZCO         ;Print Char in [C]
       JMP     NEXT$CHAR

ABORT: JMP     0           ;Hopefully CPM is still OK



                           ;Write character (in [C]) to Lava-10 Board
LOUT:  MVI     A,DRAW$TEXT ;Send Draw Text Command
       CALL    PULSE$WR
       MVI     A,1         ;Send 2 characters
       CALL    PULSE$WR
       MOV     A,C
       CALL    PULSE$WR    ;Send Ascii 2X
```

```
       MOV    A,C
       CALL   PULSE$WR      ;Send Ascii (So we have an even number of bytes sent)
       CALL   NextPosition  ;Advance the cursor one position
       RET

NextPosition:               ;Advance cursor one position
       PUSH   H             ;Because [HL] & [DE] will be changed
       PUSH   D
       CALL   GET$X         ;Get X position into HL
       MOV    A,H
       CPI    02            ;> 512 pixels
       JC     SAME$LINE

       CALL   GET$Y         ;Start a new line
       LXI    D,16
       DAD    D             ;Add DE to HL for next line
       CALL   PUT$Y
       LXI    H,0           ;Start of line always
       JMP    LINE2
SAME$LINE:
       LXI    D,8
       DAD    D             ;Add DE to HL
LINE2: CALL   PUT$X
       POP    D             ;Get back Origional [DE]
       POP    H             ;Get back Origional [HL]
       RET



GET$X: MVI    A,READ$CSR    ;READ Register command
       CALL   PULSE$WR      ;send
       MVI    A,0
       CALL   PULSE$WR
       MVI    A,0
       CALL   PULSE$WR
       MVI    A,01H
       CALL   PULSE$WR      ;Point to CSR_FONT_X
       CALL   PULSE$RD      ;Read 2 bytes, into [HL]
       RET

GET$Y: MVI    A,READ$CSR    ;READ Register command
       CALL   PULSE$WR      ;send
       MVI    A,0
       CALL   PULSE$WR
       MVI    A,0
       CALL   PULSE$WR
       MVI    A,02H
       CALL   PULSE$WR      ;Point to CSR_FONT_Y
       CALL   PULSE$RD      ;Read 2 bytes, into [HL]
       RET
```

```
PUT$X: MVI    A,WRITE$CSR  ;WRITE Register command
       CALL   PULSE$WR     ;send
       MVI    A,0
       CALL   PULSE$WR
       MVI    A,0
       CALL   PULSE$WR
       MVI    A,01H
       CALL   PULSE$WR     ;Point to CSR_FONT_X
       MOV    A,H
       CALL   PULSE$WR
       MOV    A,L
       CALL   PULSE$WR
       RET

PUT$Y: MVI    A,WRITE$CSR  ;WRITE Register command
       CALL   PULSE$WR     ;send
       MVI    A,0
       CALL   PULSE$WR
       MVI    A,0
       CALL   PULSE$WR
       MVI    A,02H
       CALL   PULSE$WR     ;Point to CSR_FONT_Y
       MOV    A,H
       CALL   PULSE$WR
       MOV    A,L
       CALL   PULSE$WR
       RET

;------------------------ LAVA WRITE ROUTINE -------------------------

PULSE$WR:
       OUT    LavaData     ;Send [A] to lava data port (91H)
       MVI    A,00000001B  ;Enable U10 Output to LAVA data bus, strobe HIGH, and LAVA set to WRITE mode
       OUT    LavaStat     ;Send to lava status port (90H)
       MVI    A,10000001B  ;Pulse LAVA strobe bit LOW
       OUT    LavaStat     ;Send to lava status port (90H)
WR$NOT$RDY:
       IN     LavaStat     ;Wait until JAVA "Done" signal clears U12A. Then we are done
       ANI    80H          ;This will set strobe bit back HIGH. Note still in WRITE LAVA mode
       JZ     WR$NOT$RDY
       RET

;------------------------ LAVA READ ROUTINE -------------------------

PULSE$RD:
       MVI    A,00001000B  ;Set strobe HIGH, set to READ MODE, Disable U10
       OUT    LavaStat     ;Send to lava status port (90H)

       MVI    A,10001000B  ;Pulse strobe bit LOW
```

```
       OUT    LavaStat      ;Send to lava status port (90H)
RD$NOT$RDY:
       IN     LavaStat      ;Wait until JAVA "Done" signal clears U12A. Then we are done
       ANI    80H           ;This will set strobe bit back HIGH. Note still in READ LAVA mode
       JZ     RD$NOT$RDY

       IN     LavaData      ;Port (91H), Data [15:8]
       MOV    H,A

       MVI    A,10001000B   ;Pulse strobe bit LOW
       OUT    LavaStat      ;Send to lava status port (90H)
RD$NOT$RDY1:
       IN     LavaStat      ;Wait until JAVA "Done" signal clears U12A. Then we are done
       ANI    80H           ;This will set strobe bit back HIGH. Note still in READ LAVA mode
       JZ     RD$NOT$RDY1

       IN     LavaData      ;Port (91H),Data [7:0]
       MOV    L,A
       RET



 ;------------- SUPPORT ROUTINES -----------------------------------------------------

 ZCRLF:
       PUSH   PSW
       MVI    C,CR
       CALL   ZCO
       MVI    C,LF
       CALL   ZCO
       POP    PSW
       RET

ZEOL:                        ;CR and clear current line
       MVI    C,CR
       CALL   ZCO
       MVI    C,CLEAR              ;Note hardware dependent, (Use 80 spaces if necessary)
       CALL   ZCO
       RET

ZCSTS:
       IN     0H            ;Get Character in [A]
       ANI    02H
       RZ
       MVI    A,01H
       ORA    A
       RET


ZCO:                         ;Write character that is in [C]
```

```
       PUSH   PSW
ZCO1:  IN     0H            ;Show Character
       ANI    04H
       JZ     ZCO1
       MOV    A,C
       OUT    1H
       POP    PSW
       RET

ZCI:                         ;Return keyboard character in [A]
       IN     0H            ;Get Character in [A]
       ANI    02H
       JZ     ZCI
       IN     01H
       RET


;      Print a string in [DE] up to '$'

PSTRING:
       PUSH   B
       PUSH   D
       PUSH   H
       XCHG
PSTRX: MOV    A,M
       CPI    '$'
       JZ     DONEP
       MOV    C,A
       CALL   ZCO
       INX    H
       JMP    PSTRX
DONEP: POP    H
       POP    D
       POP    B
       RET


; Print a 16 bit number in [DE] +SPACE

printparm$DE:
       PUSH   B
       MOV    A,D
       CALL   PHEX
       MOV    A,E
       CALL   PHEX
       MVI    C,' '
       CALL   ZCO
       POP    B
       RET
```

```
; Print an 8 bit number, located in [A]
PHEX:  PUSH   PSW
       PUSH   B
       PUSH   PSW
       RRC
       RRC
       RRC
       RRC
       CALL   ZCONV
       POP    PSW
       CALL   ZCONV
       POP    B
       POP    PSW
       RET

ZCONV: ANI    0FH          ;HEX to ASCII and print it
       ADI    90H
       DAA
       ACI    40H
       DAA
       MOV    C,A
       CALL   ZCO
       RET

;DISPLAY BIT PATTERN IN [A]
ZBITS: PUSH   PSW
       PUSH   B
       PUSH   D
       MOV    E,A
       MVI    B,8
BQ2:   DB     0CBH,23H
       SLAR   E            ;Z80 Op code for SLA A,E
       MVI    A,18H
       ADC    A
       MOV    C,A
       CALL   ZCO
       DJNZ   BQ2
       POP    D
       POP    B
       POP    PSW
       RET


SIGN$ON:      DB     CR,LF,'LAVA-10 Test Program 4/27/2012 (V0.4) '
              DB     CR,LF,'Type characters from keyboard, ESC to abort.',CR,LF,'$'
TestString:   DB     '   >>> This is a test for rapidly sending '
              DB     'characters 1234567890. <<<< $'

              DS     100H
STACK         DW     0H
```

```
;END
```