```
; This is a "Hello World" program for Z80 and TMS9918 / TMS9928 /
; TMS9929 /  V9938 or V9958 VDP.
; That means that this should work on SVI, MSX, Colecovision, Memotech,
; and many other Z80 based home computers or game consoles.
;
; Because we don't know what system is used, we don't know where RAM
; is, so we can't use stack in this program.
;
; This version of Hello World was written by Timo "NYYRIKKI" Soilamaa
; 17.10.2001
;
; Converted to Z80ASM by Leon Byles 25 Feb 2012 for testing
; N8VEM-S100 Computers S-100 VDP board
;
; <<< NOTE CONSOLE I/O HARDWARE DEPENDENT ROUTINES ZCI: and ZCO BELOW >>>
; Converted to Dumb terminal display mode by John Monahan 6/23/2013
;
;      V1.0         ;Entering CR from keyboard repositions "cursor"
;
;
;-----------------------------------------------------------------------
; Configure this part:

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


DATAP:        EQU    98H    ; VDP Data port
CMDP:         EQU    99H    ; VDP Command port (99 works on all MSX models)
LATCH:        EQU    9CH    ; LATCH-WR* port

ramdac_address_wr EQU      90H
ramdac_address_rd EQU      93H
ramdac_palette_ram EQU     91H
ramdac_pixel_read_mask EQU 92H

ramdac_over_wr      EQU    94H
ramdac_over_rd      EQU    97H
ramdac_over_ram EQU 95H
ramdac_do_not_use EQU      96H    ;Listed as reserved

ramdac_latch  EQU    9CH    ;<<< May not be correct

LATCH_RAMDAC_256  EQU    80H     ; 256 color mode
LATCH_OVERLAY_MASK EQU   0FH    ; 4 overlay bits

;-----------------------------------------------------------------------
; Program starts here:

       ORG 0100H           ;Z80 starts always from here when power is turned on

       DI                  ;We don't know, how interrupts works in this system,
                           ;so we disable them.

begin: LD     SP,STACK
       LD     HL,SIGN$ON   ;Print a welcome message on Console
       CALL   PSTRING

       LD     A,010H       ;reset VDP by
       OUT    (LATCH),A    ;toggling  SYNTMS* bit
       NOP
```

```
       NOP
       NOP
       NOP
       NOP
       LD     A,00H
       OUT    (LATCH),A

                          ;Let's set VDP write to address register with value of 0000H
       XOR    A           ;Send LSB (00000000)
       OUT    (CMDP),A
       LD     A,40H       ;nets MSB (01XXXXXX) (Note range is 0 to 3FFFH)
       OUT    (CMDP),A    ;Bits 7 & 6 = 00 for reads, 01 for writes

                          ;Now let's clear first 64Kb of VDP memory
       LD     B,0
       LD     HL,0FFFFH
       LD     C,DATAP
CLEAR1:
       OUT    (C),B
       DEC    HL
       LD     A,H
       OR     L
       NOP                ;Let's wait 16 clock cycles just in case VDP is not quick enough.
       NOP
       NOP
       JR     NZ,CLEAR1
                          ;---------------------------------------
                          ;Now it is time to set up VDP registers:
                          ;Register 0 set to 0H
                          ;
                          ;Set mode selection to TEXT2 MODE (80X24 Characters/line)
                          ;Disable external video & horizontal interrupt
       LD     A,04        ;00000100 (DG, NO LP, No Horz Int, MODE 5,4,3 = 010, 0)
       LD     C,CMDP      ;[C] will contain the command port# for all register adjustments below
       LD     E,80H       ;Register #0

       OUT    (C),A       ;Data = 0H
       OUT    (C),E       ;E = Register = 0 (+128)


                          ;---------------------------------------
                          ;Register 1 set to 50H  (01010000B) (4K, Enable Blank, DI, MODE 1, MODE 2, 0,
8X8, Normal Size)
                          ;Select 80 column mode, enable screen and disable vertical interrupt
       LD     A,50H
       INC    E           ;To register #1
       OUT    (C),A       ;Date = 50H   (80X24, TEXT2 Mode)
       OUT    (C),E       ;E = Register = 1 (+128)


                          ;---------------------------------------
                          ;Register 2 to 0H
                          ;
                          ;Set Name Table base address to 0000H
       XOR    A
       INC    E           ;To register #2
       OUT    (C),A       ;Date = 0
       OUT    (C),E       ;E = Register #2 (+128)


                          ;---------------------------------------
                          ;Register 3 is ignored as TEXT2 column mode does not need
                          ;a Color Table base address
                          ;
       INC    E           ;To Register #3


                          ;---------------------------------------
                          ;Register 4 to 1H
                          ;Set Pattern Generator Table base address to to 800H
```

```
       INC    A             ;Not clear why 800H <<< Manual, p29, says 1000H, cannot get it to work there
>>>
       INC    E             ;To Register #4

       OUT    (C),A         ;Data = 1  (00000001)
       OUT    (C),E         ;E = Register #4 (+128)
                            ;Reg #4 = 0,0,A16,A15,A14,A13,A12,A11 (A11 = 1 for 800H)


       LD     A,09H         ;(00001001) = no mouse,no LP, set color code,color bus input mode,
                            ;             VRAM = 64K, 0, Sprite disable, Set to B&W
       OUT    (C),A         ;Send 00001000
       LD     A,48H
       OUT    (C),A         ;To Reg #8 (no mouse, no LP, Color code, color bus,64KX1 RAM refresh,

                            ;Let's next copy character set

       PUSH   BC            ;Save C
       LD     HL,CHARS
       LD     B,(CHARS_END-CHARS)/8      ;Number of 8 byte characters to move
COPYCHARS:
       LD     C,8           ;8 bytes per character
COPY1: LD     A,(HL)
       OUT    (DATAP),A
       INC    HL
       NOP                  ;Let's wait 8 clock cycles just in case VDP is not quick enough.
       NOP
       NOP
       NOP
       DEC    C
       JP     NZ,COPY1
       DJNZ   COPYCHARS
       POP    BC            ;Return C

                            ;----------------------------------------
                            ;Registers 5 (Sprite attribute) & 6 (Sprite pattern) are ignored
                            ;as 40 column mode does not have sprites
       INC    E             ;To Register #5
       INC    E             ;To Register #6

                            ;----------------------------------------
                            ;Register 7 to F0H
                            ;Set colors to white on black
       LD     A,0F1H        ;Bits 7-4 = Text color (White), Bits 3-0 Background color (Black)
       INC    E             ;To Register #7
       OUT    (C),A         ;Data = 0F1H  (white on black)
       OUT    (C),E         ;E = Register #7 (+128)

       CALL   INIT_RAMDAC   ;Now Let's setup the RAMDAC (see below)

                            ;======  We are done setting up registers! =======

                            ;Let's set write address to start of "name table" (CRT Display)
       LD     IY,0          ;[IY] Will ALWAYS hold the line number on screen (0 - 23)
       LD     IX,0          ;[IX] Will ALWAY  hold X position on current line (0 - 79)
       XOR    A
       OUT    (C),A         ;send 0 to CMD port
       LD     A,40H
       OUT    (C),A

       LD     HL,VH_SIGNON  ;Send VPD Board signon message to VGA display
       CALL   VPSTRING

NEXTCHAR:
       CALL   ZCI           ;Stay here until keyboard hit
       CP     A,ESC         ;ESC to abort
       JP     Z,DONE
       LD     C,A
       CALL   ZCO           ;Char will still be in [A] on return
       CP     A,CR          ;If CR then special treatment
       JP     NZ,NO_CR
```

```
       CALL   DO_CR         ;CR+LF
       JP     NEXTCHAR
NO_CR: SUB    A,1FH         ;Adjust ASCII to table offset
       OUT    (DATAP),A     ;Send character to DATA PORT
       CALL   UPDATE_POSITION     ;Update cursor positioning data
       JP     NEXTCHAR

DONE:  LD     C,00H         ; CP/M SYSTEM RESET CALL
       CALL   0005H         ; RETURN TO PROMPT


UPDATE_POSITION:
       PUSH   HL
       INC    IX            ;Line position
       PUSH   IX
       POP    HL
       LD     A,H
       CP     A,50H         ;50H = 80 if at end of line
       POP    HL
       RET    NZ            ;If not at end of line then return
       LD     IX,0          ;X back to start of line
       INC    IY            ;Normally the chip will position cursor on next line
       RET                  ;<<<< Scroll on bottom line not done yet >>>


DO_CR: PUSH   HL            ;Do a CR+LF on screen (80 characters/line)
       PUSH   BC
       LD     IX,0
       INC    IY            ;Next line
       PUSH   IY
       POP    HL            ;Now calculate position for chip (0 - 3FFF) from line count
       LD     A,L           ;Line count will be 0 to 23
       LD     BC,80         ;80 characters/line
       LD     HL,0
DO_CR1:       ADD    HL,BC         ;0,80,160,240,320,,,,
       DEC    A             ;line count
       JP     NZ,DO_CR1
       CALL   SET_CURSOR    ;Update V9938
       POP    BC
       POP    HL
       RET

SET_CURSOR:                 ;Set cursor at position [A]+[HL]
       RLC    H
       RLA
       RLC    H
       RLA
       SRL    H
       SRL    H
       OUT    (CMDP),A      ;(Note Control Port)
       LD     A,14+128
       OUT    (CMDP),A      ;Send A16, A15, A14 to register #14 (Note Control Port)
       LD     A,L
       NOP
       NOP
       OUT    (CMDP),A      ;(Note Control Port)
       LD     A,H
       OR     64
       OUT    (CMDP),A      ;(Note Control Port)
       RET
                            ; The end                                        ;


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

INIT_RAMDAC:
       LD     E,0                  ; ramdac_init();
       CALL   ramdac_overlay

       LD     E,1                  ;for (i=1; i<=15; ++i)
       LD     HL,default_palette   ; ramdac_set_overlay_color(i, default_palette[i]);
```

```
NEXT_COLOR:
       CALL   ramdac_set_over_color
       LD     A,16
       CP     A,E                 ; Do 1,2,3,...,15
       JP     NZ,NEXT_COLOR

       LD     E,8                 ;ramdac_overlay(8)
       CALL   ramdac_overlay

       LD     E,0FH               ;ramdac_set_read_mask(0x0F);
       CALL   ramdac_set_read_mask

       LD     E,0                 ;for (i=0; i<=15; ++i)
       LD     HL,default_palette  ;  ramdac_set_palette_color(i, default_palette[i]);
NEXT_PALLET:
       CALL   ramdac_set_palette_color
       LD     A,16
       CP     A,E                 ; Do 1,2,3,...,15
       JP     NZ,NEXT_PALLET

       LD     E,0FH               ;ramdac_set_read_mask(0x0F)
       CALL   ramdac_set_read_mask

       LD     E,0                 ;ramdac_overlay(0);
       CALL   ramdac_overlay
       RET


ramdac_overlay:                   ;Output value in E
       LD     A,E
       AND    A,LATCH_OVERLAY_MASK
       OR     A,LATCH_RAMDAC_256
       OUT    (ramdac_latch),A
       RET


ramdac_set_read_mask:             ;Output value in E
       LD     A,E
       OUT    (ramdac_pixel_read_mask),A
       RET

                                  ;Output RGB values in E
ramdac_set_over_color:
       LD     A,E
       OUT    (ramdac_over_wr),A  ;overlay
       LD     A,(HL)
       OUT    (ramdac_over_ram),A ;RED
       INC    HL
       LD     A,(HL)
       OUT    (ramdac_over_ram),A ;GREEN
       INC    HL
       LD     A,(HL)
       OUT    (ramdac_over_ram),A ;BLUE
       INC    HL
       INC    E
       RET

ramdac_set_palette_color:
       LD     A,E
       OUT    (ramdac_address_wr),A      ;overlay
       LD     A,(HL)
       OUT    (ramdac_palette_ram),A     ;RED
       INC    HL
       LD     A,(HL)
       OUT    (ramdac_palette_ram),A     ;GREEN
       INC    HL
       LD     A,(HL)
       OUT    (ramdac_palette_ram),A     ;BLUE
       INC    HL
       INC    E
       RET
```

```
                        ;Return keyboard character in [A]
ZCI:   IN     A,0H     ;Get Character in [A]
       AND    02H
       JP     Z,ZCI
       IN     A,01H
       RET

ZCO:   PUSH   AF       ;Write character that is in [C]
ZCO1:  IN     A,0H     ;Show Character
       AND    04H
       JP     Z,ZCO1
       LD     A,C
       OUT    01H,A
       POP    AF
       RET


PSTRING:LD    A,(HL)   ;Print a string in [HL] up to '$' on CPM/Console
       CP     A,'$'
       RET    Z
       LD     C,A
       CALL   ZCO
       INC    HL
       JP     PSTRING

VPSTRING:              ;Print a string in [HL] up to '$' on V9938 VGA Display
       LD     A,(HL)
       CP     A,'$'
       RET    Z
       LD     C,A
       CALL   ZCO      ;Show on CPM Console first (A is returned unchanged)
       SUB    A,1FH    ;Adjust ASCII to table offset
       OUT    (DATAP),A
       CALL   UPDATE_POSITION     ;Update cursor positioning data
       INC    HL
       DJNZ   VPSTRING



; DISPLAY CURRNT VALUE IN [A] (For Debugging Only)
; All registers unchanged

LBYTE:PUSH    AF
       PUSH   BC
       PUSH   AF
       RRCA
       RRCA
       RRCA
       RRCA
       CALL   CONV
       CALL  ZCO
       POP    AF
SF598: CALL   CONV
       CALL   ZCO      ;Send to consol
       POP    BC
       POP    AF
       RET
;
CONV:  AND    0FH      ;CONVERT HEX TO ASCII
       ADD    A,90H
       DAA
       ADC    A,40H
       DAA
       LD     C,A
       RET


default_palette:
```

```
       DB     0,0,0
       DB     0,0,128
       DB     0,128,0
       DB     0,128,128
       DB     128,0,0
       DB     128,0,128
       DB     128,64,0
       DB     128,128,128
       DB     64,64,64
       DB     0,0,255
       DB     0,255,0
       DB     0,255,255
       DB     255,0,0
       DB     255,0,255
       DB     255,255,0
       DB     255,255,255


SIGN$ON:DB    CR,LF,'AW9938 on VDP S-100 Board Test Program 6/21/2013 (V0.1) '
       DB     CR,LF,'Enter characters to appear on screen. ESC to abort.',CR,LF,LF,'$'

VH_SIGNON: DB'VH9938, VIDEO DISPLAY  (80 X 24 Characers/Line, TEXT2 Mode). $'


CHARS:                                                      ;This is the IBM-PC Character ROM +20H
              DB     000H,000H,000H,000H,000H,000H,000H,000H          ;SP
              DB     030H,078H,078H,030H,030H,000H,030H,000H          ;!
              DB     06CH,06CH,06CH,000H,000H,000H,000H,000H          ;"
              DB     06CH,06CH,0FEH,06CH,0FEH,06CH,06CH,000H          ;#
              DB     030H,07CH,0C0H,078H,00CH,0F8H,030H,000H          ;$
              DB     000H,0C6H,0CCH,018H,030H,066H,0C6H,000H          ;%
              DB     038H,06CH,038H,076H,0DCH,0CCH,076H,000H          ;&
              DB     060H,060H,0C0H,000H,000H,000H,000H,000H          ;'
              DB     018H,030H,060H,060H,060H,030H,018H,000H          ;(
              DB     060H,030H,018H,018H,018H,030H,060H,000H          ;)
              DB     000H,066H,03CH,0FFH,03CH,066H,000H,000H          ;*
              DB     000H,030H,030H,0FCH,030H,030H,000H,000H          ;+
              DB     000H,000H,000H,000H,000H,030H,030H,060H          ;'
              DB     000H,000H,000H,0FCH,000H,000H,000H,000H          ;-
              DB     000H,000H,000H,000H,000H,030H,030H,000H          ;.
              DB     006H,00CH,018H,030H,060H,0C0H,080H,000H          ;/

              DB     07CH,0C6H,0CEH,0DEH,0F6H,0E6H,07CH,000H          ;30, 0
              DB     030H,070H,030H,030H,030H,030H,0FCH,000H          ;31, 1
              DB     078H,0CCH,00CH,038H,060H,0CCH,0FCH,000H          ;32, 2
              DB     078H,0CCH,00CH,038H,00CH,0CCH,078H,000H          ;33, 3
              DB     01CH,03CH,06CH,0CCH,0FEH,00CH,01EH,000H          ;34, 4
              DB     0FCH,0C0H,0F8H,00CH,00CH,0CCH,078H,000H          ;35, 5
              DB     038H,060H,0C0H,0F8H,0CCH,0CCH,078H,000H          ;36, 6
              DB     0FCH,0CCH,00CH,018H,030H,030H,030H,000H          ;37, 7
              DB     078H,0CCH,0CCH,078H,0CCH,0CCH,078H,000H          ;38, 8
              DB     078H,0CCH,0CCH,07CH,00CH,018H,070H,000H          ;39, 9

              DB     000H,030H,030H,000H,000H,030H,030H,000H          ;:
              DB     000H,030H,030H,000H,000H,030H,030H,060H          ;;
              DB     018H,030H,060H,0C0H,060H,030H,018H,000H          ;<
              DB     000H,000H,0FCH,000H,000H,0FCH,000H,000H          ;=
              DB     060H,030H,018H,00CH,018H,030H,060H,000H          ;>
              DB     078H,0CCH,00CH,018H,030H,000H,030H,000H          ;?
              DB     07CH,0C6H,0DEH,0DEH,0DEH,0C0H,078H,000H          ;@
              DB     030H,078H,0CCH,0CCH,0FCH,0CCH,0CCH,000H          ;A
              DB     0FCH,066H,066H,07CH,066H,066H,0FCH,000H          ;B
              DB     03CH,066H,0C0H,0C0H,0C0H,066H,03CH,000H          ;C
              DB     0F8H,06CH,066H,066H,066H,06CH,0F8H,000H          ;D
              DB     0FEH,062H,068H,078H,068H,062H,0FEH,000H          ;E
              DB     0FEH,062H,068H,078H,068H,060H,0F0H,000H          ;F
              DB     03CH,066H,0C0H,0C0H,0CEH,066H,03EH,000H          ;G
              DB     0CCH,0CCH,0CCH,0FCH,0CCH,0CCH,0CCH,000H          ;H
              DB     078H,030H,030H,030H,030H,030H,078H,000H          ;I
              DB     01EH,00CH,00CH,00CH,0CCH,0CCH,078H,000H          ;J
              DB     0E6H,066H,06CH,078H,06CH,066H,0E6H,000H          ;K
```

```
        DB      0F0H,060H,060H,060H,062H,066H,0FEH,000H         ;L
        DB      0C6H,0EEH,0FEH,0FEH,0D6H,0C6H,0C6H,000H         ;M
        DB      0C6H,0E6H,0F6H,0DEH,0CEH,0C6H,0C6H,000H         ;N
        DB      038H,06CH,0C6H,0C6H,0C6H,06CH,038H,000H         ;O
        DB      0FCH,066H,066H,07CH,060H,060H,0F0H,000H         ;P
        DB      078H,0CCH,0CCH,0CCH,0DCH,078H,01CH,000H         ;Q
        DB      0FCH,066H,066H,07CH,06CH,066H,0E6H,000H         ;R
        DB      078H,0CCH,0E0H,070H,01CH,0CCH,078H,000H         ;S
        DB      0FCH,0B4H,030H,030H,030H,030H,078H,000H         ;T
        DB      0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0FCH,000H         ;U
        DB      0CCH,0CCH,0CCH,0CCH,0CCH,078H,030H,000H         ;V
        DB      0C6H,0C6H,0C6H,0D6H,0FEH,0EEH,0C6H,000H         ;W
        DB      0C6H,0C6H,06CH,038H,038H,06CH,0C6H,000H         ;X
        DB      0CCH,0CCH,0CCH,078H,030H,030H,078H,000H         ;Y
        DB      0FEH,0C6H,08CH,018H,032H,066H,0FEH,000H         ;Z
        DB      078H,060H,060H,060H,060H,060H,078H,000H         ;[
        DB      0C0H,060H,030H,018H,00CH,006H,002H,000H         ;\
        DB      078H,018H,018H,018H,018H,018H,078H,000H         ;]
        DB      010H,038H,06CH,0C6H,000H,000H,000H,000H         ;^
        DB      000H,000H,000H,000H,000H,000H,000H,0FFH         ;_

        DB      030H,030H,018H,000H,000H,000H,000H,000H         ;'
        DB      000H,000H,078H,00CH,07CH,0CCH,076H,000H         ;a
        DB      0E0H,060H,060H,07CH,066H,066H,0DCH,000H         ;b
        DB      000H,000H,078H,0CCH,0C0H,0CCH,078H,000H         ;c
        DB      01CH,00CH,00CH,07CH,0CCH,0CCH,076H,000H         ;d
        DB      000H,000H,078H,0CCH,0FCH,0C0H,078H,000H         ;e
        DB      038H,06CH,060H,0F0H,060H,060H,0F0H,000H         ;f
        DB      000H,000H,076H,0CCH,0CCH,07CH,00CH,0F8H         ;g
        DB      0E0H,060H,06CH,076H,066H,066H,0E6H,000H         ;h
        DB      030H,000H,070H,030H,030H,030H,078H,000H         ;i
        DB      00CH,000H,00CH,00CH,00CH,0CCH,0CCH,078H         ;j
        DB      0E0H,060H,066H,06CH,078H,06CH,0E6H,000H         ;k
        DB      070H,030H,030H,030H,030H,030H,078H,000H         ;l
        DB      000H,000H,0CCH,0FEH,0FEH,0D6H,0C6H,000H         ;m
        DB      000H,000H,0F8H,0CCH,0CCH,0CCH,0CCH,000H         ;n
        DB      000H,000H,078H,0CCH,0CCH,0CCH,078H,000H         ;o

        DB      000H,000H,0DCH,066H,066H,07CH,060H,0F0H         ;p
        DB      000H,000H,076H,0CCH,0CCH,07CH,00CH,01EH         ;q
        DB      000H,000H,0DCH,076H,066H,060H,0F0H,000H         ;r
        DB      000H,000H,07CH,0C0H,078H,00CH,0F8H,000H         ;s
        DB      010H,030H,07CH,030H,030H,034H,018H,000H         ;t
        DB      000H,000H,0CCH,0CCH,0CCH,0CCH,076H,000H         ;u
        DB      000H,000H,0CCH,0CCH,0CCH,078H,030H,000H         ;v
        DB      000H,000H,0C6H,0D6H,0FEH,0FEH,06CH,000H         ;w
        DB      000H,000H,0C6H,06CH,038H,06CH,0C6H,000H         ;x
        DB      000H,000H,0CCH,0CCH,0CCH,07CH,00CH,0F8H         ;y
        DB      000H,000H,0FCH,098H,030H,064H,0FCH,000H         ;z
        DB      01CH,030H,030H,0E0H,030H,030H,01CH,000H         ;{
        DB      018H,018H,018H,000H,018H,018H,018H,000H         ;|
        DB      0E0H,030H,030H,01CH,030H,030H,0E0H,000H         ;}
        DB      076H,0DCH,000H,000H,000H,000H,000H,000H         ;~
        DB      000H,010H,038H,06CH,0C6H,0C6H,0FEH,000H         ;DEL
;
CHARS_END:

        DS      40H
STACK:  DW      0

        END
```