

## Interfacing LCD with 8088 CPU to Print a Text Message

### By: Usman Rafiq (CE Department, CUI Lahore Campus)

LCD is interfaced with 8088 CPU through PPI. Addresses of PPI are as under:

40H -> PortA  
 41H -> PortB  
 42H -> PortC  
 43H -> CW

```
.DATA
; -----
; MESSAGE TO DISPLAY
; -----
MESSAGE DB 'HELLO', 00H

.CODE
START:
    ; INITIALIZE 8255: PORT A OUTPUT, PORT C OUTPUT
    MOV     AL, 80H           ; 10000000B => MODE 0, PORT A OUT, PORT C OUT
    OUT     43H, AL           ; WRITE TO CONTROL WORD REGISTER

    CALL    INIT_LCD          ; INITIALIZE THE LCD

    ; DISPLAY "HELLO"
    MOV     AX, @DATA
    MOV     DS, AX
    LEA     SI, MESSAGE        ; LOAD OFFSET OF MESSAGE STRING INTO SI
    CALL    DISPLAY_STRING

    HLT                      ; HALT EXECUTION

; -----
; LCD INITIALIZATION SUBROUTINE
; -----
INIT_LCD:
    MOV     AL, 38H           ; FUNCTION SET: 8-BIT, 2-LINE, 5x7 FONT
    CALL    LCD_CMD
    CALL    DELAY

    MOV     AL, 0CH           ; DISPLAY ON, CURSOR OFF
    CALL    LCD_CMD
    CALL    DELAY

    MOV     AL, 01H           ; CLEAR DISPLAY
    CALL    LCD_CMD
```

```

CALL DELAY

MOV AL, 06H          ; ENTRY MODE SET: INCREMENT CURSOR
CALL LCD_CMD
CALL DELAY

RET

; -----
; SEND COMMAND TO LCD
; -----
LCD_CMD:
    OUT    40H, AL      ; SEND COMMAND TO PORT A (LCD DATA)

    MOV    AL, 00H      ; RS=0, RW=0, EN=0
    OUT    42H, AL      ; WRITE CONTROL LINES TO PORT C

    OR     AL, 04H      ; SET EN=1 (PC2)
    OUT    42H, AL      ; ENABLE HIGH

    CALL   SHORT_DELAY  ; SHORT DELAY FOR PULSE WIDTH

    AND    AL, 0FBH     ; CLEAR EN (BIT 2 = 0)
    OUT    42H, AL      ; ENABLE LOW

    RET

; -----
; SEND DATA TO LCD
; -----
LCD_DATA:
    OUT    40H, AL      ; SEND DATA TO LCD

    MOV    AL, 01H      ; RS=1, RW=0, EN=0
    OUT    42H, AL      ; CONTROL LINES

    OR     AL, 04H      ; EN=1
    OUT    42H, AL      ; ENABLE HIGH

    CALL   SHORT_DELAY

    AND    AL, 0FBH     ; EN=0
    OUT    42H, AL      ; ENABLE LOW

    RET

```

```

; -----
; DISPLAY STRING ON LCD
; -----
DISPLAY_STRING:
NEXT_CHAR:
    LODSB                ; LOAD BYTE FROM [DS:SI] TO AL AND INCREMENT SI
    CMP     AL, 00H      ; CHECK FOR NULL TERMINATOR
    JE      DONE        ; END IF NULL

    CALL    LCD_DATA     ; SEND CHARACTER TO LCD
    CALL    DELAY        ; WAIT

    JMP     NEXT_CHAR

DONE:
    RET

; -----
; SHORT DELAY SUBROUTINE
; -----
SHORT_DELAY:
    MOV     CX, 1000
DELAY1:
    NOP
    LOOP    DELAY1
    RET

; -----
; MAIN DELAY SUBROUTINE
; -----
DELAY:
    MOV     CX, 0FFFFH
DELAY_LOOP:
    NOP
    LOOP    DELAY_LOOP
    RET

```