

ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น
188 460 ไมโครคอนโทรลเลอร์

Experiment MCL-02: LCD Module

วัตถุประสงค์

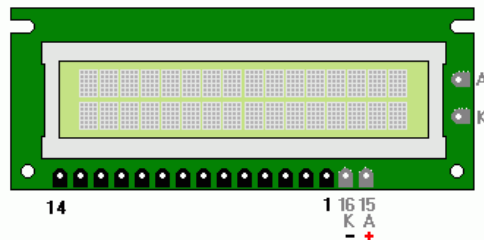
เพื่อศึกษาการทำงานและวิธีการควบคุม Character LCD module โดยใช้ PIC Microcontroller

อุปกรณ์และเครื่องมือ

1. เครื่องคอมพิวเตอร์พร้อมโปรแกรม MPLAB สำหรับการเขียนโปรแกรมภาษาแอสเซมบลีและ WinPIC800 สำหรับการบันทึกโปรแกรมลงในไมโครคอนโทรลเลอร์
2. เครื่องบันทึกโปรแกรมลงในไมโครคอนโทรลเลอร์ ETT EMPIC V2.0 พร้อม power supply และสายเชื่อมต่อกับเครื่องคอมพิวเตอร์
3. PIC MCU 16F628A
4. Crystal Oscillator 4.0 MHz.
5. Ceramic capacitor 30 pF x 2
6. Digital Experiment Box
7. 2 Lines, 16 Character/Line LCD Module
8. Potentiometer 10K

หลักการทำงานของ LCD Module

LCD Module ชนิดตัวอักษร (Alphanumeric LCD) ที่ใช้ Hitachi HD44780 เป็น controller จะมีหลักการเดียวกันในการควบคุม



Pin No.	Symbol	Description	Level	Function	
1	VSS	Ground			Ground
2	VDD	Power Supply	-	+5V	ต่อกับแรงดันไฟเลี้ยง +5V
3	VO	LCD Contrast	-	-	ต่อกับแรงดันเพื่อปรับความเข้มของการแสดงผล
4	RS	Register Select	H/L	RS = 0 เป็นการติดต่อกับรีจิสเตอร์คำสั่ง (Instruction Register) RS = 1 เป็นการติดต่อกับรีจิสเตอร์ข้อมูล (Data Register)	
5	R/W	Read/Write	H/L	R/W = 0 หมายถึงต้องการเขียนข้อมูลไปยัง LCD โมดูล R/W = 1 หมายถึงต้องการอ่านข้อมูลจาก LCD โมดูล	
6	E	Enable	H, H->L	Enable Signal	
7 - 14	DB0-DB7	Data Bus	H/L	Data Bus Line	
15	A	Back Light A	-	ไฟแสงสว่างส่องหน้าจอ +5V (สำหรับรุ่นที่มี Back Light)	
16	K	Back Light K	-	ไฟแสงสว่างส่องหน้าจอ 0V (สำหรับรุ่นที่มี Back Light)	

ไม่ว่าหน้าจอ LCD จะแสดงผลได้กี่บรรทัดหรือกี่ตัวอักษรต่อบรรทัดก็ตาม โดยผู้ใช้งานจะต้องทำการส่งคำสั่งเพื่อ initialize ตัว controller ให้ถูกต้องก่อนที่จะใช้งาน ซึ่งมีหลักการดังต่อไปนี้

- ชุดคำสั่งต่างๆ สำหรับการปรับตั้งค่าของ LCD module จะส่งไปยัง Command Register ของตัว controller
- ข้อมูลที่จะส่งให้ไปปรากฏบนหน้าจอ LCD จะส่งไปยัง Data Register ของตัว controller

โดยข้อมูลจะส่งเข้าไปยัง controller ผ่านขาสัญญาณ DB0-DB7 ซึ่งผู้ใช้สามารถที่จะเลือกต่อสายสัญญาณได้ทั้ง 8 เส้น (8-bit mode) หรือจะใช้เพียง 4 เส้น (4-bit mode) ก็ได้ แต่จะต้องใช้คำสั่งเพื่อ initialize ตัว LCD controller ให้ถูกต้อง

เนื่องจาก LCD controller จะมีระยะเวลาตอบสนองคำสั่ง ซึ่งผู้ใช้อาจจะเลือกวิธีการอ่านสถานะจาก controller เพื่อตรวจสอบความพร้อมของ controller ในการที่จะรับคำสั่งต่อไปก็ได้ หรืออาจจะใช้วิธีการชลอ (delay) การส่งคำสั่งเข้าไปยัง controller ให้เป็นไปตามข้อกำหนดทางเวลาก็ได้ สำหรับการทดลองนี้จะเลือกการเชื่อมต่อแบบ 4 bit เพื่อเป็นการประหยัด I/O ของ PIC และใช้วิธีการชลอการส่งคำสั่งเพื่อให้ไม่ต้องควบคุมสถานะของขา R/W ทำให้ลดการใช้ I/O ไปได้อีก

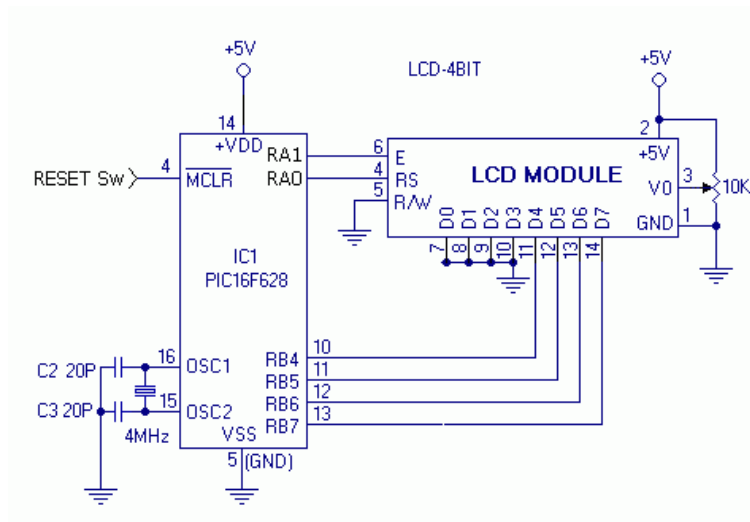
เนื่องจากชุดคำสั่งของ LCD controller มีขนาด 8 บิต แต่การเชื่อมต่อมีช่องสัญญาณเพียง 4 บิต ดังนั้นชุดคำสั่งและข้อมูลที่จะส่งไปยัง controller จะต้องทำการส่ง 2 ครั้ง ต่อข้อมูลหรือคำสั่ง 1 ไบต์ สำหรับชุดคำสั่งพื้นฐานบางส่วน ของ LCD controller มีดังนี้

Instruction	Code										Description	Execution Time (max) (when f_{op} or f_{osc} is 270 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.	
Return home	0	0	0	0	0	0	0	0	1	—	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.	1.52 ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	37 μ s
Display on/off control	0	0	0	0	0	0	1	D	C	B	Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).	37 μ s
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	—	—	Moves cursor and shifts display without changing DDRAM contents.	37 μ s
Function set	0	0	0	0	1	DL	N	F	—	—	Sets interface data length (DL), number of display lines (N), and character font (F).	37 μ s

I/D = 1: Increment
 I/D = 0: Decrement
 S = 1: Accompanies display shift
 S/C = 1: Display shift
 S/C = 0: Cursor move
 R/L = 1: Shift to the right
 R/L = 0: Shift to the left
 DL = 1: 8 bits, DL = 0: 4 bits
 N = 1: 2 lines, N = 0: 1 line
 F = 1: 5 $\times$ 10 dots, F = 0: 5 $\times$ 8 dots

ขั้นตอนการทดลอง

1. ต่อวงจรดังต่อไปนี้บนกล่องทดลอง Digital box



2. ป้อนและแปลโปรแกรมภาษาแอสเซมบลีต่อไปใน MPLAB

```
list p=16f628
#include <p16f628.inc>
#define RS PORTA,0
#define E PORTA,1

COM          equ 0x20          ;LCD Command storage
DAT          equ 0x21          ;LCD Data storage
count1      equ 0x22
count2      equ 0x23
count3      equ 0x24

org 0x000

        movlw 0x07
        movwf CMCON            ;Turn comparator off
        call  init_lcd
        movlw "C"              ;Send char "A" to LCD
        call  wr_data          ;write data to LCD
        movlw "o"
        call  wr_data
        movlw "E"
        call  wr_data
        goto  $                ;halt

init_lcd ;**** Initialize LCD ****
        banksel TRISA          ;bank 1
        movlw 0x00
        movwf TRISA           ;PortA = output
        movwf TRISB           ;PortB = output
        banksel PORTA         ;bank 0
        clrf PORTA
        clrf PORTB
        call  delay
        movlw b'00110011'     ;Set LCD status
        call  wr_ins          ;write command to LCD
        movlw b'00110010'     ;Set LCD status
        call  wr_ins
        movlw b'00101000'     ;4-bit mode,2 lines, 5x7 char
        call  wr_ins
        movlw b'00001100'     ;turn on LCD, cursor off
        call  wr_ins
        movlw b'00000110'     ;entry mode, incr addr
        call  wr_ins
        movlw b'00000001'     ;clear DDRAM
        call  wr_ins
        return
```

```

wr_ins      ;**** WRITE COMMAND to LCD ****
bsf    RS      ;RS = 0
bsf    E      ;E = 1
movwf  COM      ;get parameter from W
andlw  0xf0     ;mask low nibble
movwf  PORTB    ;send to LCD
call   enable   ;enable LCD
swapf  COM,W    ;swap low-high nibble
andlw  0xf0     ;mask low nibble
movwf  PORTB    ;send to LCD
call   enable   ;enable LCD
return

wr_data      ;**** WRITE DATA TO LCD ****
bsf    RS      ;RS = 1
bsf    E      ;E = 1
movwf  DAT      ;get parameter from W
andlw  0xf0     ;mask low nibble
movwf  PORTB    ;send to LCD
call   enable   ;enable LCD
swapf  DAT,W    ;swap low-high nibble
andlw  0xf0     ;mask low nibble
movwf  PORTB    ;send to LCD
call   enable   ;enable LCD
return

enable       ;**** generate LCD enable signal ***
bsf    E      ;E = 0
call   delay
bsf    E      ;E = 1
return

delay        ;**** delay subroutine ****
movlw  .50      ;W = 50 (decimal)
movwf  count1
loop1      clrf  count2
loop2      decfsz count2      ;inner loop
           goto  loop2
           decfsz count1
           goto  loop1
           return
end

```

3. ทำการ download hex file ที่ได้จากขั้นตอนที่ 1 ลงในไมโครคอนโทรลเลอร์
4. ทดสอบการทำงานและสังเกตการแสดงผลบน LCD
5. จัดเก็บ source code สำหรับใช้เป็น library สำหรับการทดลองต่อไป ที่ต้องใช้การแสดงผลบนจอ LCD

คำถามท้ายการทดลอง

เพิ่มเติมวงจรโดยต่อสวิทช์เข้าที่ RB0, RB1, RB2 และ RB3 จากนั้นให้แก้ไขโปรแกรมในข้อ 2 เพื่อให้แสดงผลสถานะของ สวิทช์ทั้ง 4 ตัวเป็นเลขฐาน 16 บนจอ LCD