제11강

Char. LCD 제어

CLCD 개요

CLCD 내부 구성

제어명령

CLCD 초기화

실습과제

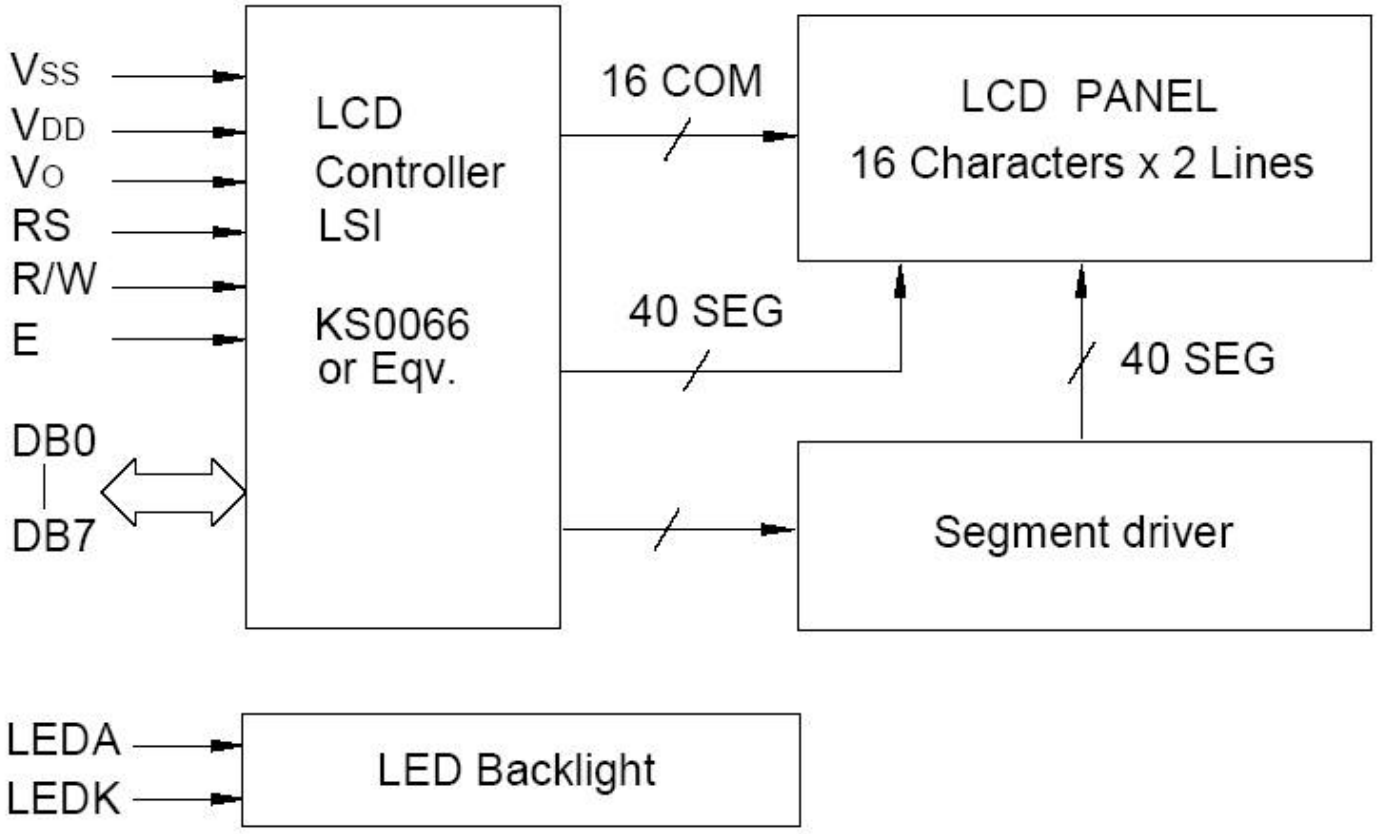
Ref.) Chapter 11

CLCD 개요

- 5V의 단일 전원으로 구동
- 5x7, 5x10 도트 크기의 문자 표현 가능
- 8비트 혹은 4비트의 데이터버스 방식으로 제어
- CG(character generator) ROM(예약된 문자패턴),
  CG RAM(사용자 문자패턴),
  DD(display data) RAM을 내장

CLCD 개요(계속)

* Char. LCD 블록도



CLCD 개요(계속)

핀번호	기 호	신호레벨	기 능
1	Vss	–	Gnd
2	Vdd	–	+5V
3	V0		밝기조절
4	RS register select	H/L	L : IR 선택(제어명령) H : DR 선택(데이터)
5	R/W	H/L	L : Write(MCU로부터 CLCD로) H : Read(CLCD로부터 MCU로)
6	E	H	Enable Signal
7	DB0	H/L	데이터버스 8비트모드 : 하위니블 4비트모드 : 사용않됨
8	DB1	H/L	
9	DB2	H/L	
10	DB3	H/L	
11	DB4	H/L	데이터버스 8비트모드 : 상위니블 4비트모드 : 상/하위니블
12	DB5	H/L	
13	DB6	H/L	
14	DB7	H/L	
15	LEDA	+5V	LED 백라이트용 전원
16	LEDK	0V	LED 백라이트용 전원

CLCD의 내부 구성

* IR(Instruction), DR(Data) 레지스터(각각 8비트)

  : RS(register select) 신호로 선택

RS	R/W	동 작
0	0	IR Write
0	1	Busy 플래그와 주소카운터 Read
1	0	DR로 Write(DR->DDRAM, CGRAM)
1	1	DR을 Read(DR<-DDRAM, CGRAM)

- IR 레지스터

  : 화면클리어, 커서이동 등의 인스트럭션 정보

  : DD RAM과 CG RAM의 주소정보 기억

- DR 레지스터

  : DD RAM, CG RAM으로 기록할 데이터 및

  DD RAM, CG RAM에서 읽어낸 데이터의 일시 기억

CLCD의 내부 구성(계속)

* BF(Busy Flag) 비트

  : CLCD가 다음 인스트럭션을 받을 수 있는 상태인가 여부

  : 명령판독(RS=0, R/W=1)인 조건에서
  IR 레지스터를 읽어 DB7 비트를 검사함으로서 판별

  : BF가 1이면 CLCD가 내부 동작중으로 다음 명령 수신불가
  BF가 0이면 다음 인스트럭션을 수신 가능 상태

* AC(Address Counter) x2

  : 주소 계수기로 DD RAM용 AC와 CG RAM용 AC 제공

  : DD RAM 혹은 CG RAM에 있는 데이터를 기록/판독할 때
  주소를 계수하는 용도로 사용

CLCD의 내부 구성(계속)

* DD(Display Data) RAM

: DD RAM은 최대 80x8비트(8비트 문자 80개)를 기억

▪ DD RAM의 주소와 CLCD의 표시라인과의 관계

: 첫 라인과 둘째 라인에 할당된 DD RAM 주소는 불연속임

: 7비트의 유효 주소 길이

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		40
1행	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	...	27
2행	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	...	67

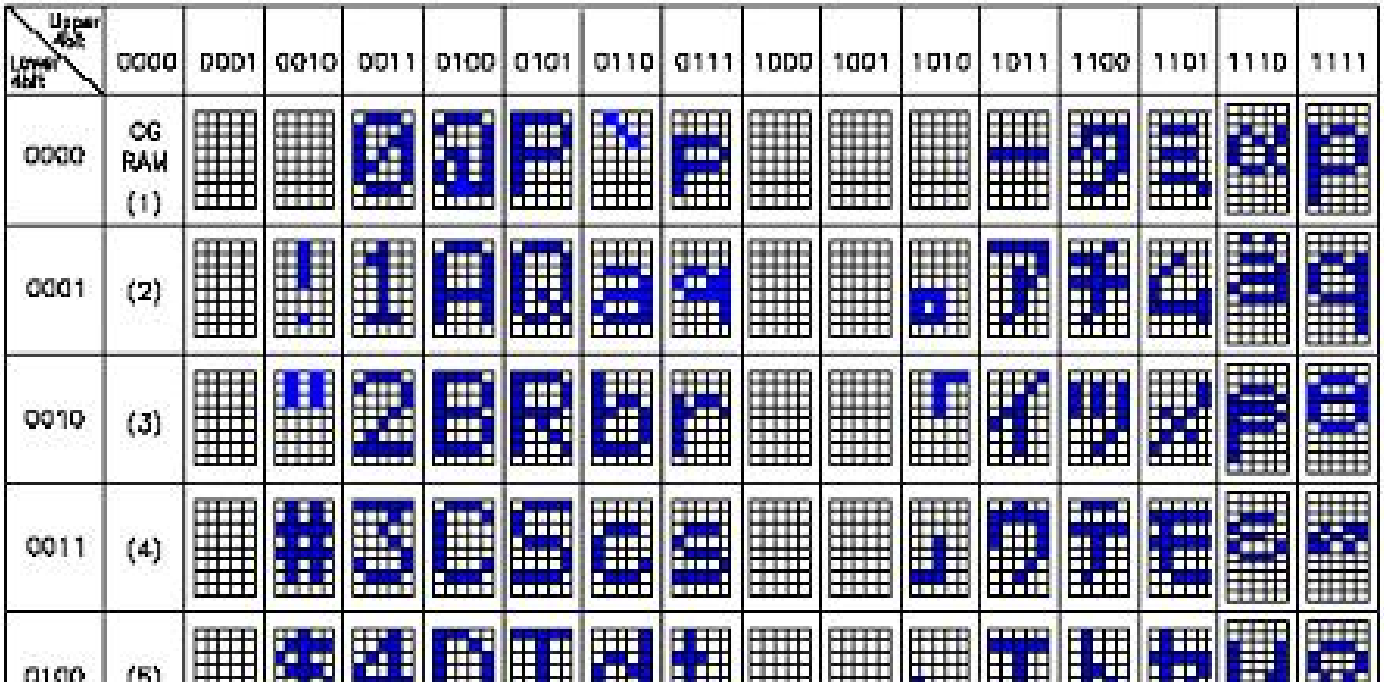
CLCD의 내부 구성(계속)

* CG ROM(Character Generator ROM)

: CG ROM에는 8비트 문자코드 192개 정의

: 5x7 도트의 문자 패턴을 저장

예) 주소 0011_0000B(30H) 에는 '0' 문자 패턴 정의



CLCD의 내부 구성(계속)

* CG RAM(Character Generator RAM)

: 사용자가 문자패턴을 정의하여 저장할 수 있는 RAM

: 8개의 5x7 혹은 4개의 5x10 크기의 문자패턴 정의 가능

문자코드(DD RAM 데이터)								CG RAM 주소						CG RAM 데이터								패턴 번호
D7	D6	D5	D4	D3	D2	D1	D0	A5	A4	A3	A2	A1	A0	P7	P6	P5	P4	P3	P2	P1	P0	
0	0	0	0	x	0	0	0	0	0	0	0	0	0	x	x	x	1	1	1	1	0	패턴1
								0	0	0	0	0	1	x	x	x	1	0	0	1	0	
								0	0	0	0	1	0	x	x	x	1	1	1	1	0	
								0	0	0	0	1	1	x	x	x	0	0	0	0	0	
								0	0	0	1	0	0	x	x	x	0	0	1	0	0	
								0	0	0	1	0	1	x	x	x	0	0	1	1	1	
								0	0	0	1	1	0	x	x	x	0	0	1	0	0	
								0	0	0	1	1	1	x	x	x	0	0	1	0	0	
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
0	0	0	0	x	1	1	1	1	1	1	0	0	0	x	x	x	1	1	1	1	1	패턴8
								1	1	1	0	0	1	x	x	x	0	0	1	0	0	
								1	1	1	0	1	0	x	x	x	0	0	1	0	0	
								1	1	1	0	1	1	x	x	x	0	0	1	0	0	
								1	1	1	1	0	0	x	x	x	0	0	1	0	0	
								1	1	1	1	0	1	x	x	x	0	0	1	0	0	
								1	1	1	1	1	0	x	x	x	0	0	1	0	0	
								1	1	1	1	1	1	x	x	x	0	0	0	0	0	

CLCD 제어명령(11개)

기 능	제어신호		제어명령(빨강:OP코드)								실행시간
	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
Clear Display	0	0	0	0	0	0	0	0	0	1	1.52ms
Return Home	0	0	0	0	0	0	0	0	1	x	1.52ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	37us
Display On/Off control	0	0	0	0	0	0	1	D	C	B	37us
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	x	x	37us
Function Set	0	0	0	0	1	DL	N	F	x	x	37us
Set CG RAM Address	0	0	0	1	CG RAM Address(6bit)						37us
Set DD RAM Address	0	0	1	DD RAM Address(7bit)							37us
Read BF and Address	0	1	BF	Address Counter							0us
Write Data to CG/DD RAM	1	0	Data								37us
Read Data from CG/DD RAM	1	1	Data								37us

CLCD 제어명령(계속)

* 비트별 기능

비트명	설 명	
	0	1
I/D	Decrement cursor position	Increment cursor position
S	No display shift	Display shift
D	Display OFF	Display ON
C	Cursor OFF	Cursor ON
B	Cursor blink OFF	Cursor blink ON
S/C	Move cursor	Shift display
R/L	Shift left	Shift right
DL	4-bit interface	8-bit interface
N	1 line	2 lines
F	5x7 dots	5x10 dots
BF	can accept instruction	Internal operation in progress
x	don't care	don't care

CLCD 제어명령(계속)

1) Clear Display

: 전체화면을 지우고(DD RAM 내용 소거) 커서를 Home으로 위치

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

```
*pLCD_C_WR = 0x01;
delay(1.52ms);
```

2) Return Home

: DD RAM의 내용 유지한채 커서를 Home 위치로 이동

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	x

```
*pLCD_C_WR = 0x02;        // 0000_0010
delay(1.52ms);
```

CLCD 제어명령(계속)

3) Entry Mode Set

: 데이터를 R/W시 커서 위치를 증가 혹은 감소(I/D), 화면 쉬프트(S) 여부설정

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

```
*pLCD_C_WR = 0x06;        // 0000_0110 : inc, no-shift
delay(37us);
```

4) Display On/Off control

: 화면을 On/Off(D), 커서를 On/Off(C), 커서를 깜빡이게(B) 여부 지정

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

```
*pLCD_C_WR = 0x0C;        // 0000_1100 : dispON, cursOFF, BlinOFF
delay(37us);
```

CLCD 제어명령(계속)

5) Cursor or Display Shift

: 화면 혹은 커서(S/C)를 우측/좌측 쉬프트(R/L) 여부 설정

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	x	x

```
*pLCD_C_WR = 0x18;        // 0001_1000 : screen left shift
delay(37us);
```

6) Function Set

: 인터페이스 길이, 화면 표시행수, 폰트 크기 설정

: Busy 플래그 명령을 제외한 다른 명령보다 우선 실행되어야 함

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	x	x

```
*pLCD_C_WR = 0x38;        // 0011_1000 : 8bit, 2line, 5x7dot
delay(37us);
```

CLCD 제어명령(계속)

7) Set CG(Character Generator) RAM Address

: CG RAM의 주소 지정

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	CG address					

```
*pLCD_C_WR = 0x40 + addr;          // 주소길이 6bit
delay(37us);
```

8) Set DD(Display Data) RAM Address

: DD RAM의 주소 지정

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	DD address						

```
*pLCD_C_WR = 0x80 + DDRAM addr;     // 주소길이 7bit
delay(37us);
```

CLCD 제어명령(계속)

9) Read Busy Flag and Address

: Busy Flag(BF) 및 주소 카운터 내용을 read

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	Address counter						

```
data = *pLCD_C_RD;
delay(0);
```

10) Write Data to CG or DD RAM

: CG RAM 혹은 DD RAM에 8비트 데이터를 기록

: 사용에 앞서 CG RAM 혹은 DD RAM 주소 지정 필요

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	data							

```
*pLCD_D_WR = data;
delay(37us);
```

CLCD 제어명령(계속)

11) Read Data from CG or DD RAM

: CG RAM 혹은 DD RAM으로부터 8비트 데이터를 판독

: 사용에 앞서 CG RAM 혹은 DD RAM 주소 지정 필요

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	data							

```
data = *pLCD_D_RD;
delay(37us);
```

사용자 패턴 정의

CG RAM 주소						CG RAM 데이터								패턴값	패턴 번호
A5	A4	A3	A2	A1	A0	P7	P6	P5	P4	P3	P2	P1	P0		
0	0	0	0	0	0	x	x	x	1	1	1	1	0	1EH	패턴1
0	0	0	0	0	1	x	x	x	1	0	0	1	0	12H	
0	0	0	0	1	0	x	x	x	1	1	1	1	0	1EH	
0	0	0	0	1	1	x	x	x	0	0	0	0	0	00H	
0	0	0	1	0	0	x	x	x	0	0	1	0	0	04H	
0	0	0	1	0	1	x	x	x	0	0	1	1	1	07H	
0	0	0	1	1	0	x	x	x	0	0	1	0	0	04H	
0	0	0	1	1	1	x	x	x	0	0	1	0	0	04H	
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:

1) 문자코드 00H의 첫 번째 라인(주소 00H)을 지정하기 위해,
Set CG RAM address명령으로 첫 라인(주소 00_0000B, 6bit) 주소 지정

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	0	0	0	0	0	0

2) Write Data to CG or DD RAM 명령을 반복 사용(AC 자동 증가)

RS	R/W	DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0
1	0	1EH

RS	R/W	DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0
1	0	12H

⋮

사용자 정의 패턴 표시

* 정의된 패턴을 2행 1열 위치(주소40H)에 표시 가정

1) Set DD RAM address 명령으로 DD RAM의 AC에 주소 40H 설정

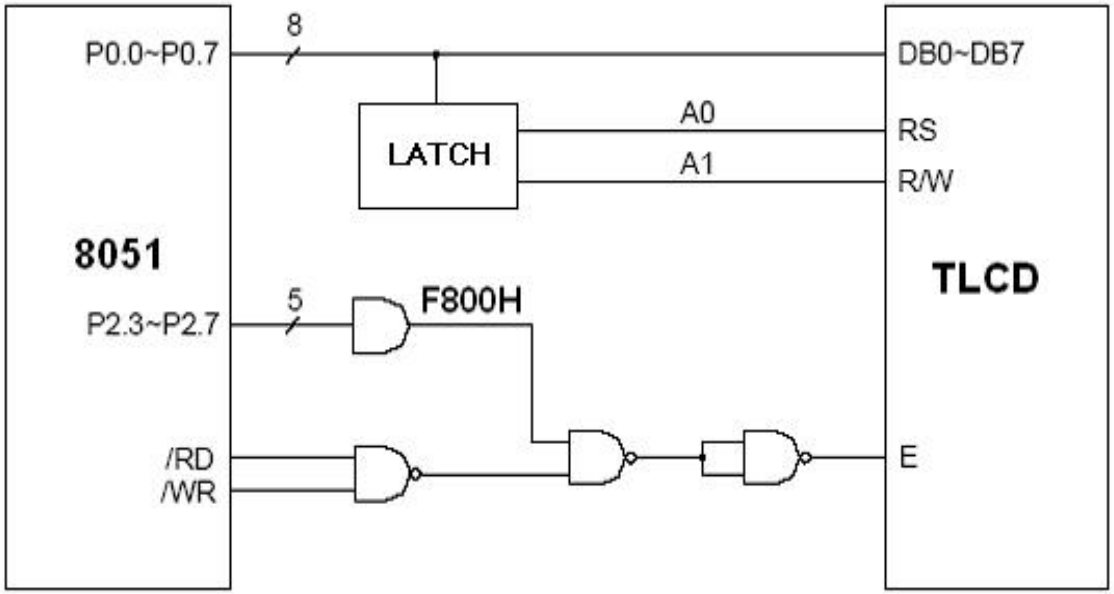
(100_0000B)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	1	0	0	0	0	0	0

2) Write Data to CG or DD RAM 명령을 통해 문자패턴 주소 지정

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	00H							

KUT51 보드에서의 제어



- CLCD 모듈의 메모리 맵(1111_1xxx_xxxx_xxR/wRs)

주 소	R/W(A1)	RS(A0)	기 능
0F800H ... F80_0000	0	0	LCD_COMMAND_WR
0F801H ... F80_0001	0	1	LCD_DATA_WR
0F802H ... F80_0010	1	0	LCD_COMMAND_RD
0F803H ... F80_0011	1	1	LCD_DATA_RD

CLCD의 초기화

* 내부 RESET 회로에 의한 초기화(H/W적으로)

■ Clear Display

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

■ Function Set

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	0	0	x	x

DL=1, N=0, F=0(8비트모드, 1라인, 5x7 dots)

■ Display ON/OFF Control

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	0	0	0

D=0, C=0, B=0(Display OFF, Cursor OFF, Blink OFF)

■ Entry Mode Set

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	1	0

I/D=1, S=0(increment, no shift)

: 내부 reset 타이밍 부적합시 초기화되지 않을 수 있음.

CLCD의 초기화(계속)

* 8비트 모드 초기화(S/W적으로)

■ Function Set

(37uSec이상 대기)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	1	0	x	x

■ Display ON/OFF Control

(37uSec이상 대기)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

■ Clear Display

(1.52mSec이상 대기)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

■ Entry Mode Set

(37uSec이상 대기)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

CLCD의 초기화(계속)

```
;INITIALIZE THE LCD(8비트 모드, 시간지연)
LCD_INIT_8:
    MOV A,#38H                  ; SET 8-BIT, 2-LINE, 5X8 FONT
    CALL LCD_COMMAND_WRITE_8
    MOV A,#0EH                  ; DISPLAY ON, CURSOR ON, BLINK OFF
    CALL LCD_COMMAND_WRITE_8
    MOV A,#01H                  ; CLEAR DISPLAY, HOME CURSOR
    CALL LCD_COMMAND_WRITE_8
    MOV A,#16                   ; REQUIRE 1.53MS
    CALL DELAY100US
    MOV A,#06H                  ; CURSOR INC.(MOVES RIGHT), NO SHIFT
    CALL LCD_COMMAND_WRITE_8
    RET
```

* LCD_COMMAND_WRITE_8 서브루틴 참조

  : 40us 시간지연포함

CLCD의 초기화(계속)

* 4비트 모드 초기화(S/W적으로)

:모든 명령 및 데이터는 상위 4비트만 이용(하위니블 비사용)

■ Function Set

(37uSec이상 대기)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	0	x	x	x	x
0	0	0	0	1	0	x	x	x	x
0	0	1	0	x	x	x	x	x	x

■ Display ON/OFF Control

(37uSec이상 대기)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	x	x	x	x
0	0	1	D	C	B	x	x	x	x

■ Clear Display

(1.52mSec이상 대기)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	x	x	x	x
0	0	0	0	0	1	x	x	x	x

■ Entry Mode Set

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	x	x	x	x
		0	1	I/D	S	x	x	x	x

CLCD의 초기화(계속)

```
;INITIALIZE TLCD(4비트 모드, 시간지연)
LCD_INIT_4:
    MOV A,#00100000B          ; *Function Set : #0010_xxxx
    CALL LCD_COMMAND_WRITE_4   ; SET 4-BIT Mode
    MOV A,#00101000B          ; Function Set : #0010_10xx
    CALL LCD_COMMAND_WRITE_4   ; SET 4-BIT, 2-LINE, 5X8 FONT
    MOV A,#00001100B          ; Display ON/OFF : #0000_1DCB
    CALL LCD_COMMAND_WRITE_4
    MOV A,#00000001B          ; Display Clear : #0000_0001
    CALL LCD_COMMAND_WRITE_4
    MOV A,#16                 ; REQUIRE 1.53MS
    CALL DELAY100US
    MOV A,#00000110B          ; Entry mode : #0000_0 1 I/D S
    CALL LCD_COMMAND_WRITE_4 ; 니블단위로 쪼개 상위니블 통해 전송 구현
    RET
```

* LCD_COMMAND_WRITE_4 : 40us 시간지연 포함, 루틴 참조

예제실습

[실습1] 문자열 출력 I

: 8비트 모드, 시간지연 RTN에 의한 방법

```
;=======================================
; CLCD_01.ASM
;  8bit Mode & Delay
;=======================================

; PORT DEFINITION
LCD_COMMAND_WR     EQU  0F800H
LCD_DATA_WR        EQU  0F801H
LCD_COMMAND_RD     EQU  0F802H
LCD_DATA_RD        EQU  0F803H

STACK              EQU  40H

          ORG 8000H
START:    MOV SP,#STACK

          CALL LCD_INIT_8

          MOV A,#01H           ; CLEAR DISPLAY, HOME CURSOR
          CALL LCD_COMMAND_WRITE_8
```

```
        MOV A,#16           ; REQUIRE 1.53MS
        CALL DELAY100US

        MOV DPTR,#MSG1      ; DISPLAY 1ST LINE
        CALL LCD_PRINT

        MOV A,#0C0H         ; CURSOR TO SECOND LINE HOME
        CALL LCD_COMMAND_WRITE_8

        MOV DPTR,#MSG2      ; DISPLAY 2ND LINE
        CALL LCD_PRINT

        JMP $

MSG1:   DB ' KUT51-LCD-8/D  ',00H          ; 00H EndOf...
MSG2:   DB 'WELLCOME TO KUT!',00H


; INITIALIZE TLCD, 8bit mode
LCD_INIT_8:
   MOV A,#38H           ; SET 8-BIT, 2-LINE, 5X8 FONT
   CALL LCD_COMMAND_WRITE_8

   MOV A,#0EH           ; DISPLAY ON, CURSOR ON, BLINK OFF
   CALL LCD_COMMAND_WRITE_8

   MOV A,#01H           ; CLEAR DISPLAY
   CALL LCD_COMMAND_WRITE_8
```

```
    MOV A,#16             ; REQUIRE 1.53MS(SPEC. SHEET)
    CALL DELAY100US

    MOV A,#06H            ; CURSOR INCRE.(MOVES RIGHT), NO SHIFT
    CALL LCD_COMMAND_WRITE_8
    RET

; SEND DATA TO LCD
; INPUT : A = OUTPUT CHARACTER
LCD_DATA_WRITE_8:
        PUSH DPL
        PUSH DPH

        MOV DPTR,#LCD_DATA_WR
        MOVX @DPTR,A

        CALL DELAY40US

        POP DPH
        POP DPL
        RET

; SEND COMMAND TO LCD
; INPUT : A = OUTPUT COMMAND
LCD_COMMAND_WRITE_8:
        PUSH DPL
        PUSH DPH
```

```
        MOV DPTR,#LCD_COMMAND_WR
        MOVX @DPTR,A

        CALL DELAY40US          ; 40usec delay

        POP DPH
        POP DPL
        RET

; SEND A STRING TO LCD
; INPUT: DPTR = STRING ADDRESS
LCD_PRINT:
        MOV A,#0                ; MOVC INDEX = 0
        MOVC A,@A+DPTR          ; READ CHAR.
        JZ LCD_PRINT01          ; STRING END '00'

        CALL LCD_DATA_WRITE_8

        INC DPTR
        SJMP LCD_PRINT          ; REPEAT LOOP
LCD_PRINT01:
        RET

; 40 MICROSECOND DELAY
; 11.0592MHZ = 1 MACHINE CYCLE = 1.085 US
; 2(CALL) + 35 = 37 (40.145 US)
DELAY40US:
        PUSH 1                  ; (2) SAVE REGISTER R1
```

```
        MOV R1,#13          ; (1) 20 DJNZ INSTR = 40US
LL4:    DJNZ R1, LL4        ; (2 X 14) = 28
        POP 1               ; (2) RESTORE REGISTER R1
        RET                 ; (2)

; ACC*100US DELAY
; INPUT: ACC = ? (100 US APPROX.)
; 2(CALL) + 100 = 102 (110 US)
DELAY100US:
        PUSH 1                      ; (1) SAVE REGISTER R1
DELAY100US01:
        MOV R1,#46                  ; (1) 46X2 DJNZX1.085=99.82us
LL1:    DJNZ R1,LL1                 ; (2 X 46) = 92
        DJNZ ACC,DELAY100US01       ; (2)
        POP 1                       ; (2)
        RET                         ; (2)

        END
```

예제실습(계속)

[실습2] 문자열 출력 II

: 8비트 모드, BF 검사에 의한 방법

```
;=======================================
; CLCD_02.ASM
;  8bit Mode & BF check
;=======================================

; PORT DEFINITION
LCD_COMMAND_WR    EQU  0F800H
LCD_DATA_WR       EQU  0F801H
LCD_COMMAND_RD    EQU  0F802H
LCD_DATA_RD       EQU  0F803H

STACK             EQU  40H

        ORG 8000H
START:  MOV SP,#STACK

        CALL LCD_INIT_8

        MOV A,#01H          ; CLEAR DISPLAY, HOME CURSOR
        CALL LCD_COMMAND_WRITE_8
```

```
        MOV DPTR,#MSG1      ; DISPLAY 1ST LINE
        CALL LCD_PRINT

        MOV A,#0C0H         ; CURSOR TO SECOND LINE HOME
        CALL LCD_COMMAND_WRITE_8
        MOV DPTR,#MSG2      ; DISPLAY 2ND LINE
        CALL LCD_PRINT

        JMP $

MSG1:   DB ' KUT51-LCD-8/BF ',00H
MSG2:   DB 'WELLCOME TO KUT!',00H

; INITIALIZE CLCD, 8bit mode
LCD_INIT_8:
        MOV A,#38H          ; SET 8-BIT, 2-LINE, 5X8 FONT
        CALL LCD_COMMAND_WRITE_8

        MOV A,#0EH          ; DISPLAY ON, CURSOR ON, BLINK OFF
        CALL LCD_COMMAND_WRITE_8

        MOV A,#01H          ; CLEAR DISPLAY, HOME CURSOR
        CALL LCD_COMMAND_WRITE_8

        MOV A,#06H          ; CURSOR INCRE.(MOVES RIGHT), NO SHIFT
        CALL LCD_COMMAND_WRITE_8
        RET
```

```
; SEND DATA TO LCD
; INPUT : A = OUTPUT CHARACTER
LCD_DATA_WRITE_8:
        PUSH DPL
        PUSH DPH

        CALL LCD_READY      ; BF Check

        MOV DPTR,#LCD_DATA_WR
        MOVX @DPTR,A

        POP DPH
        POP DPL
        RET

; SEND COMMAND TO LCD
; INPUT : A = OUTPUT COMMAND
LCD_COMMAND_WRITE_8:
        PUSH DPL
        PUSH DPH

        CALL LCD_READY      ; BF Check

        MOV DPTR,#LCD_COMMAND_WR
        MOVX @DPTR, A

        POP DPH
        POP DPL
```

```
        RET

; WAIT UNTIL LCD READY BIT SET
LCD_READY:
        PUSH ACC
        MOV DPTR,#LCD_COMMAND_RD
LCD_READY0:
        MOVX A,@DPTR
        JB ACC.7,LCD_READY0      ; LOOP IF BF=1
        POP ACC
        RET

; SEND A STRING TO LCD
; INPUT: DPTR = STRING ADDRESS
LCD_PRINT:
        MOV A,#0                 ; MOVC INDEX = 0
        MOVC A,@A+DPTR           ; READ CHAR.

        JZ LCD_PRINT01           ; STRING END '00'

        CALL LCD_DATA_WRITE_8

        INC DPTR
        SJMP LCD_PRINT           ; REPEAT LOOP
LCD_PRINT01:
        RET

        END
```

예제실습(계속)

[실습3] 문자열 출력 III

: 4비트 모드, 시간지연 루틴활용

```
;=========================================
; CLCD_03.ASM
;  4bit Mode & Delay
;=========================================
$MOD51

; PORT DEFINITION
LCD_COMMAND_WR     EQU  0F800H
LCD_DATA_WR        EQU  0F801H
LCD_COMMAND_RD     EQU  0F802H
LCD_DATA_RD        EQU  0F803H

STACK              EQU  40H

        ORG 8000H
START:  MOV SP,#STACK

        CALL LCD_INIT_4

        MOV A,#00000001B    ; Display Clear : #0000 0001
```

```
        CALL LCD_COMMAND_WRITE_4
        MOV A,#16               ; REQUIRE 1.52MS
        CALL DELAY100US

        MOV DPTR,#MSG1          ; DISPLAY 1ST LINE
        CALL LCD_PRINT

        MOV A,#0C0H             ; #0C0H, CURSOR TO 2nd LINE
        CALL LCD_COMMAND_WRITE_4

        MOV DPTR,#MSG2          ; DISPLAY 2ND LINE
        CALL LCD_PRINT

        JMP $

MSG1:   DB ' KUT51-LCD-4/D ',00H
MSG2:   DB 'WELLCOME TO KUT!',00H


; INITIALIZE LCD, 4bit mode
LCD_INIT_4:
        MOV A,#00100000B    ; *Function Set  #0010 xxxx
        CALL LCD_COMMAND_WRITE_4 ; SET 4-BIT

        MOV A,#00101000B    ; Function Set  #0010 10xx
        CALL LCD_COMMAND_WRITE_4 ; SET 4-BIT, 2-LINE, 5X8 FONT

        MOV A,#00001100B    ; Display ON/OFF  #0000 1DCB
```

```
        CALL LCD_COMMAND_WRITE_4

        MOV A,#00000001B    ; Display Clear  #0000 0001
        CALL LCD_COMMAND_WRITE_4
        MOV A,#16
        CALL DELAY100US

        MOV A,#00000110B    ; Entry mode  #0000 0 1 I/D S
        CALL LCD_COMMAND_WRITE_4

        RET

; SEND COMMAND TO LCD
; INPUT : A = OUTPUT COMMAND
LCD_COMMAND_WRITE_4:
        PUSH DPL
        PUSH DPH

        MOV DPTR,#LCD_COMMAND_WR
        MOVX @DPTR,A            ; higher nibble
        SWAP A                  ;
        MOVX @DPTR, A           ; lower nibble

        CALL DELAY40US

        POP DPH
        POP DPL
        RET
```

```
; SEND DATA TO LCD
; INPUT : A = OUTPUT CHARACTER
LCD_DATA_WRITE_4:
        PUSH DPL
        PUSH DPH
        MOV DPTR,#LCD_DATA_WR
        MOVX @DPTR,A            ; higher nibble
        SWAP A                  ;
        MOVX @DPTR, A           ; lower nibble

        CALL DELAY40US

        POP DPH
        POP DPL
        RET

; SEND A STRING TO LCD
; INPUT: DPTR = STRING ADDRESS
LCD_PRINT:
        MOV A,#0            ; MOVC INDEX = 0
        MOVC A,@A+DPTR      ; READ CHAR.
        JZ LCD_PRINT01      ; STRING END '00'

        CALL LCD_DATA_WRITE_4

        INC DPTR
        SJMP LCD_PRINT      ; REPEAT LOOP
```

```
LCD_PRINT01:
        RET

; 40 MICROSECOND DELAY
; 11.0592MHZ = 1 MACHINE CYCLE = 1.085 US
; 2(CALL) + 35 = 37 (40.145 US)
DELAY40US:
        PUSH 1
        MOV R1,#13
LL4:    DJNZ R1,LL4
        POP 1
        RET

; ACC*100US DELAY
; INPUT: ACC = ? (100 US APPROX.)
; 2(CALL) + 100 = 102 (110 US)
DELAY100US:
        PUSH 1
DELAY100US01:
        MOV R1,#46
LL1:    DJNZ R1,LL1
        DJNZ ACC,DELAY100US01
        POP 1
        RET

        END
```

예제실습(계속)

[실습4] 사용자 문자패턴 출력

: 문자패턴을 정의하여 출력하는 프로그램 작성

```
;=========================================
; CLCD_04.ASM
;  8bit Mode & BF Check & User Pattern
;=========================================
$MOD51

; PORT DEFINITION
LCD_COMMAND_WR     EQU  0F800H
LCD_DATA_WR        EQU  0F801H
LCD_COMMAND_RD     EQU  0F802H
LCD_DATA_RD        EQU  0F803H

STACK              EQU  40H

        ORG 8000H
START:  MOV SP,#STACK

        CALL LCD_INIT_8

        CALL REG_PATTERN
```

```
        MOV A, #01H          ; #01H, CLEAR DISPLAY, HOME CURSOR
        CALL LCD_COMMAND_WRITE_8

        MOV DPTR, #MSG1      ; DISPLAY 1ST LINE
        CALL LCD_PRINT

        MOV A, #0C4H         ; #0C4H, CURSOR TO 2ND LINE(2행 중간위치)
        CALL LCD_COMMAND_WRITE_8

        CALL PRT_PATTERN     ; DISPLAY USER PATTERN

        JMP $

MSG1:   DB  '  USER PATTERN  ',00H

PAT0:   DB 1EH, 12H, 1EH, 00H, 04H, 07H, 04H, 04H    ; 마
PAT1:   DB 09H, 15H, 09H, 01H, 01H, 01H, 01H, 01H    ; 이
PAT2:   DB 1FH, 01H, 07H, 01H, 00H, 1FH, 00H, 00H    ; 크
PAT3:   DB 1FH, 01H, 1FH, 10H, 1FH, 00H, 04H, 1FH    ; 로
PAT4:   DB 1FH, 0AH, 0AH, 1FH, 00H, 1FH, 00H, 00H    ; 프
PAT5:   DB 1FH, 01H, 1FH, 10H, 1FH, 00H, 04H, 1FH    ; 로
PAT6:   DB 04H, 0AH, 11H, 00H, 05H, 1DH, 05H, 05H    ; 세
PAT7:   DB 04H, 0AH, 11H, 00H, 01H, 1FH, 01H, 01H    ; 서


; INITIALIZE TLCD
LCD_INIT_8:
```

```
        MOV A,#38H          ; SET 8-BIT, 2-LINE, 5X8 FONT
        CALL LCD_COMMAND_WRITE_8

        MOV A,#0CH          ; DISPLAY ON, CURSOR OFF, BLINK OFF
        CALL LCD_COMMAND_WRITE_8

        MOV A,#06H          ; CURSOR INC.(MOVES RIGHT), NO SHIFT
        CALL LCD_COMMAND_WRITE_8
        RET

; Save CGRAM defined Pattern (매번 CG RAM 주소 지정하는 방식으로 )
REG_PATTERN:
        MOV DPTR,#PAT0
        MOV R5,#64
        MOV R6,#40H         ; CG RAM address, 0100_0000
        MOV R7,#0           ; R7 <- OFFSET
LP_R:   MOV A,R6
        ADD A,R7            ; A <- R6 + R7
        CALL LCD_COMMAND_WRITE_8          ; CG RAM 주소 설정
        INC R7              ; OFFSET +1
        ;
        CLR A
        MOVC A,@A+DPTR
        CALL LCD_DATA_WRITE_8             ; 패턴 등록
        INC DPTR
        DJNZ R5,LP_R
        RET
```

```
; Display Pattern "마이크로프로세서"
PRT_PATTERN:
        MOV R2,#08H
        MOV A,#00H              ; CG RAM address
LP_P:   CALL LCD_DATA_WRITE_8
        INC A
        DJNZ R2,LP_P
        RET

; WAIT UNTIL LCD READY BIT SET
LCD_READY:
        PUSH ACC
        MOV DPTR,#LCD_COMMAND_RD
LCD_READY0:
        MOVX A,@DPTR
        JB ACC.7,LCD_READY0     ; LOOP IF BF=1
        POP ACC
        RET

; SEND DATA TO LCD
; INPUT : A = OUTPUT CHARACTER
LCD_DATA_WRITE_8:
        PUSH DPL
        PUSH DPH

        CALL LCD_READY

        MOV DPTR,#LCD_DATA_WR
```

```
        MOVX @DPTR,A

        POP DPH
        POP DPL
        RET

; SEND COMMAND TO LCD
; INPUT : A = OUTPUT COMMAND
LCD_COMMAND_WRITE_8:
        PUSH DPL
        PUSH DPH

        CALL LCD_READY

        MOV DPTR,#LCD_COMMAND_WR
        MOVX @DPTR,A

        POP DPH
        POP DPL
        RET

; SEND A STRING TO LCD
; INPUT: DPTR = STRING ADDRESS
LCD_PRINT:
        MOV A,#0            ; MOVC INDEX = 0
        MOVC A,@A+DPTR      ; READ CHAR.
        JZ LCD_PRINT01      ; STRING END '00'
```

```
        CALL LCD_DATA_WRITE_8

        INC DPTR
        SJMP LCD_PRINT      ; REPEAT LOOP
LCD_PRINT01:
        RET

        END
```

실습과제

[과제1] CLCD 제어 I

: [실습4]를 활용하여
문자열이 좌, 우로 반복 이동하도록 하는 기능 포함

[과제2] CLCD 제어 II

: 4비트 인터페이스모드, BF 검사 활용

실습과제(계속)

[과제3] 사용자 패턴 등록 및 표시 I

: [실습4] 참조

: 4비트 인터페이스 모드로 제어

: 패턴등록시 랜덤순으로("로세크마로이프서"),
표시할 때는 순서에 맞게("마이크로프로세서")

[과제4] 사용자 패턴 등록 및 표시 II

: 원하는 패턴을 설계하여 표시

: 기타 사항은 각자 정의

실습과제(계속)

[과제5] 수신 문자열을 CLCD에 표시

: 시리얼통신을 통해 수신된 문자열을 CLCD에 표시

: 기타 제반사항은 각자 정의