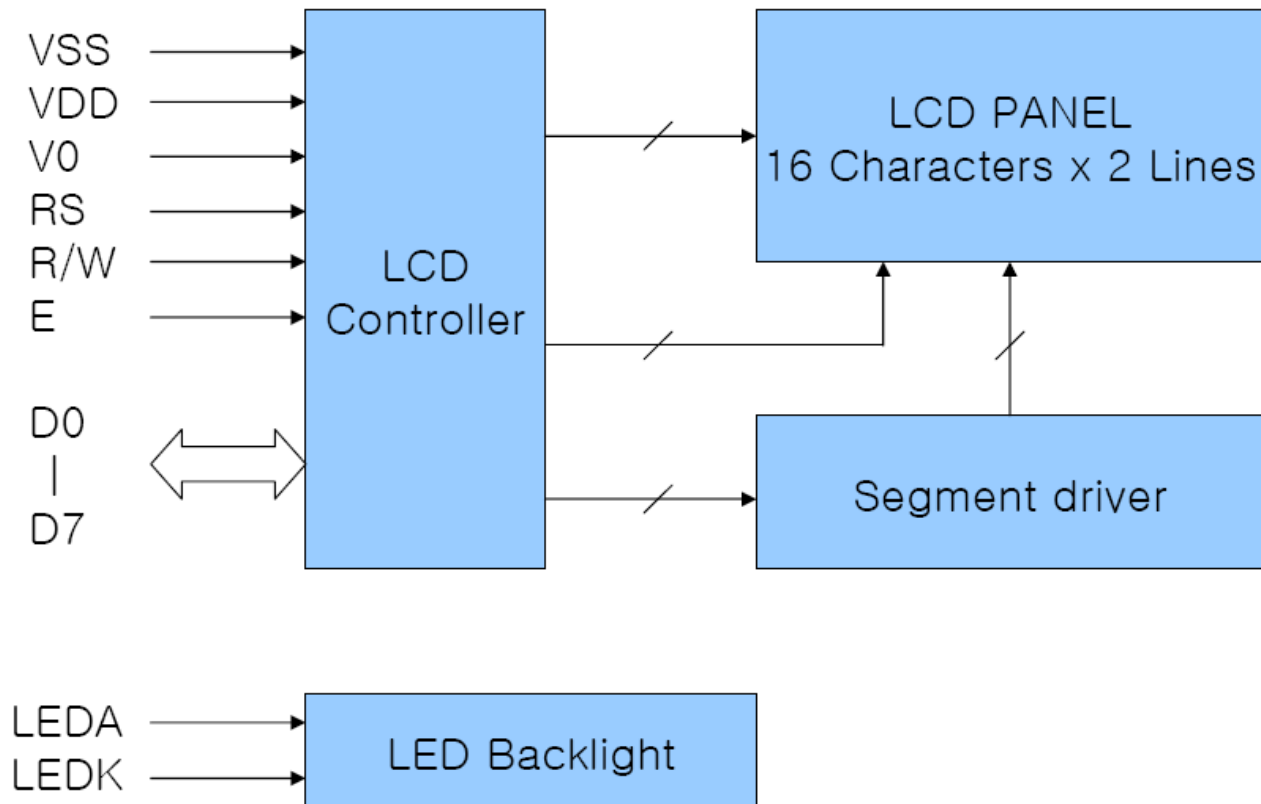

Lab5

Character LCD
Using Memory Mapped I/O

Character LCD Block Diagram

LCD Controller, LCD Panel, Segment Driver, Backlight



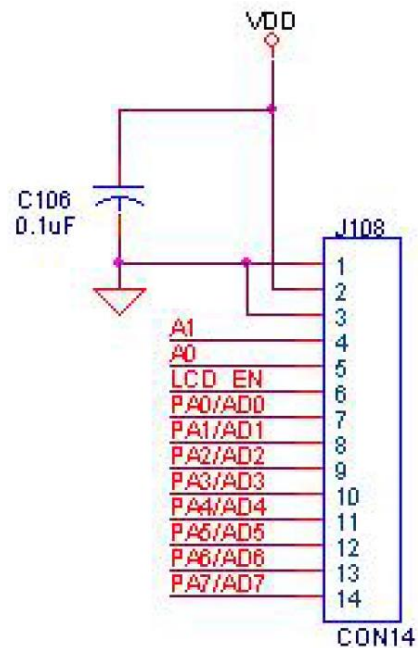
Registers

- IR (Instruction Register)
- DR (Data Register)

Table 1. Various Kinds of Operations according to RS and R/W Bits

RS	R/W	Operation
L	L	Instruction Write operation (MPU writes Instruction code into IR)
L	H	Read Busy Flag (DB7) and address counter (DB0 - DB6)
H	L	Data Write operation (MPU writes data into DR)
H	H	Data Read operation (MPU reads data from DR)

LCD Connector



LCD Connector

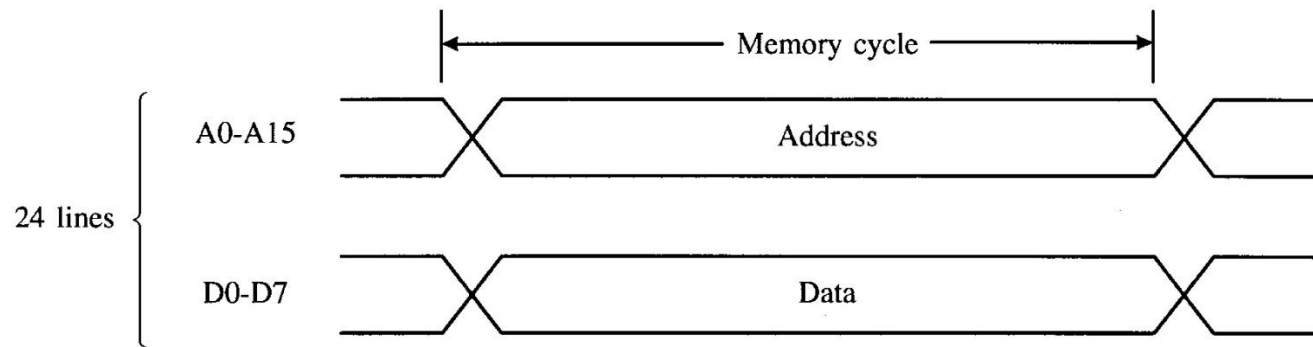
Pin no.	Symbol	External connection	Function
1	V _{SS}	Power supply	Signal ground for LCM
2	V _{DD}		Power supply for logic for LCM
3	V ₀		Contrast adjust
4	RS	MPU	Register select signal
5	R/W	MPU	Read/write select signal
6	E	MPU	Operation (data read/write) enable signal
7~10	DB0~DB3	MPU	Four low order bi-directional three-state data bus lines. Used for data transfer between the MPU and the LCM. These four are not used during 4-bit operation.
11~14	DB4~DB7	MPU	Four high order bi-directional three-state data bus lines. Used for data transfer between the MPU
15	LED+	LED BKL power supply	Power supply for BKL
16	LED-		Power supply for BKL

Addressing of Registers

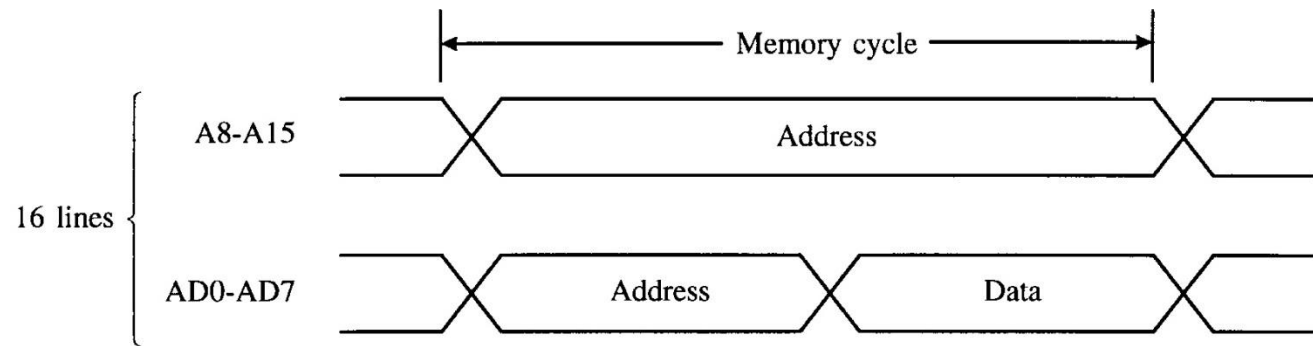
Table 1. Various Kinds of Operations according to RS and R/W Bits

RS [A1]	R/W [A0]	Operation
L 0	L 0	Instruction Write operation (MPU writes Instruction code into IR)
L	H	Read Busy Flag (DB7) and address counter (DB0 - DB6)
H 1	L 0	Data Write operation (MPU writes data into DR)
H	H	Data Read operation (MPU reads data from DR)

External SRAM Interface



(a) Nonmultiplexed (24 lines)



(b) Multiplexed (16 lines)

External SRAM Interface

- MCUCR : B7, B6

Bit	7	6	5	4	3	2	1	0	
	SRE	SRW10	SE	SM1	SM0	SM2	IVSEL	IVCE	MCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

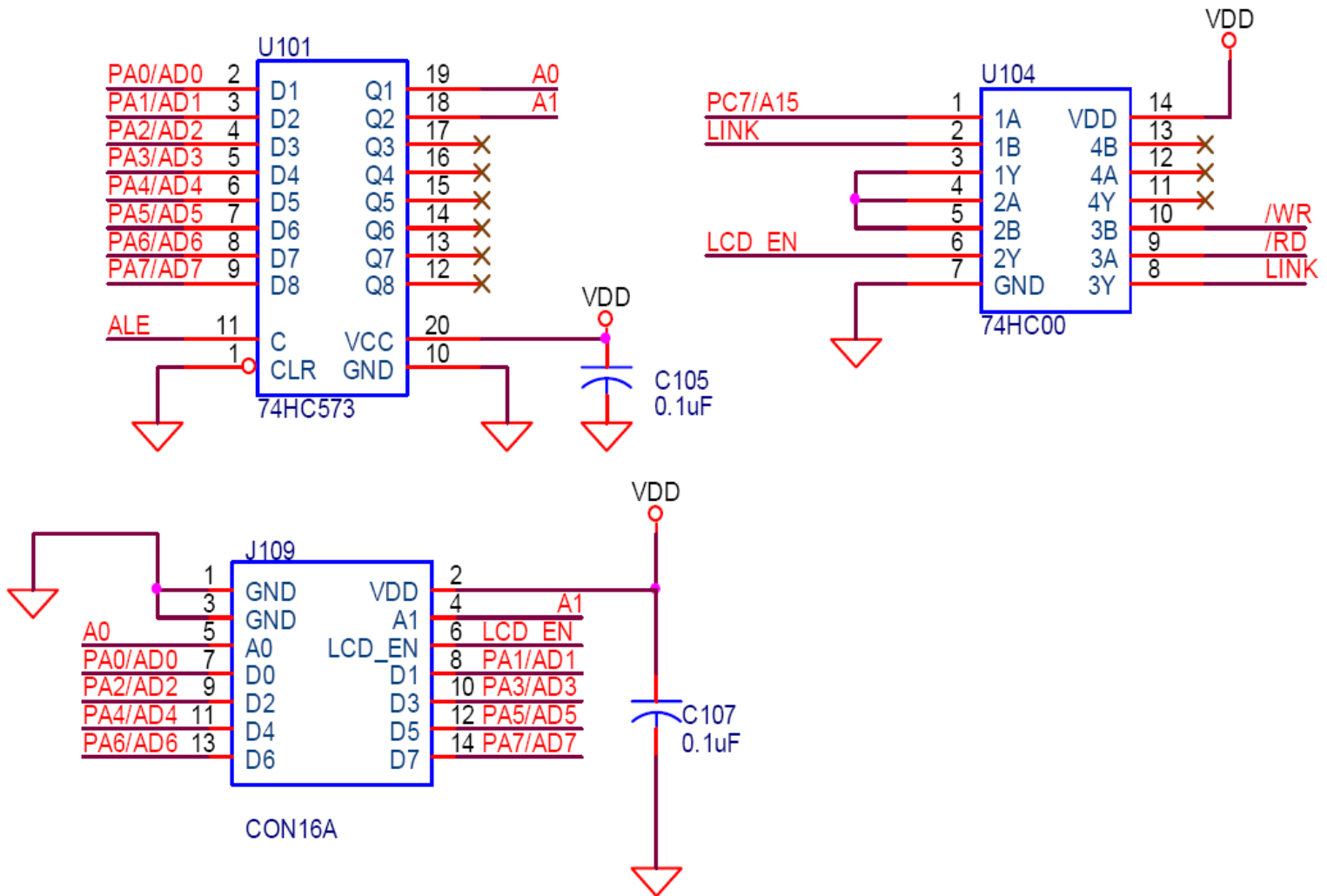
- Bit 7 – SRE: External SRAM/XMEM Enable**

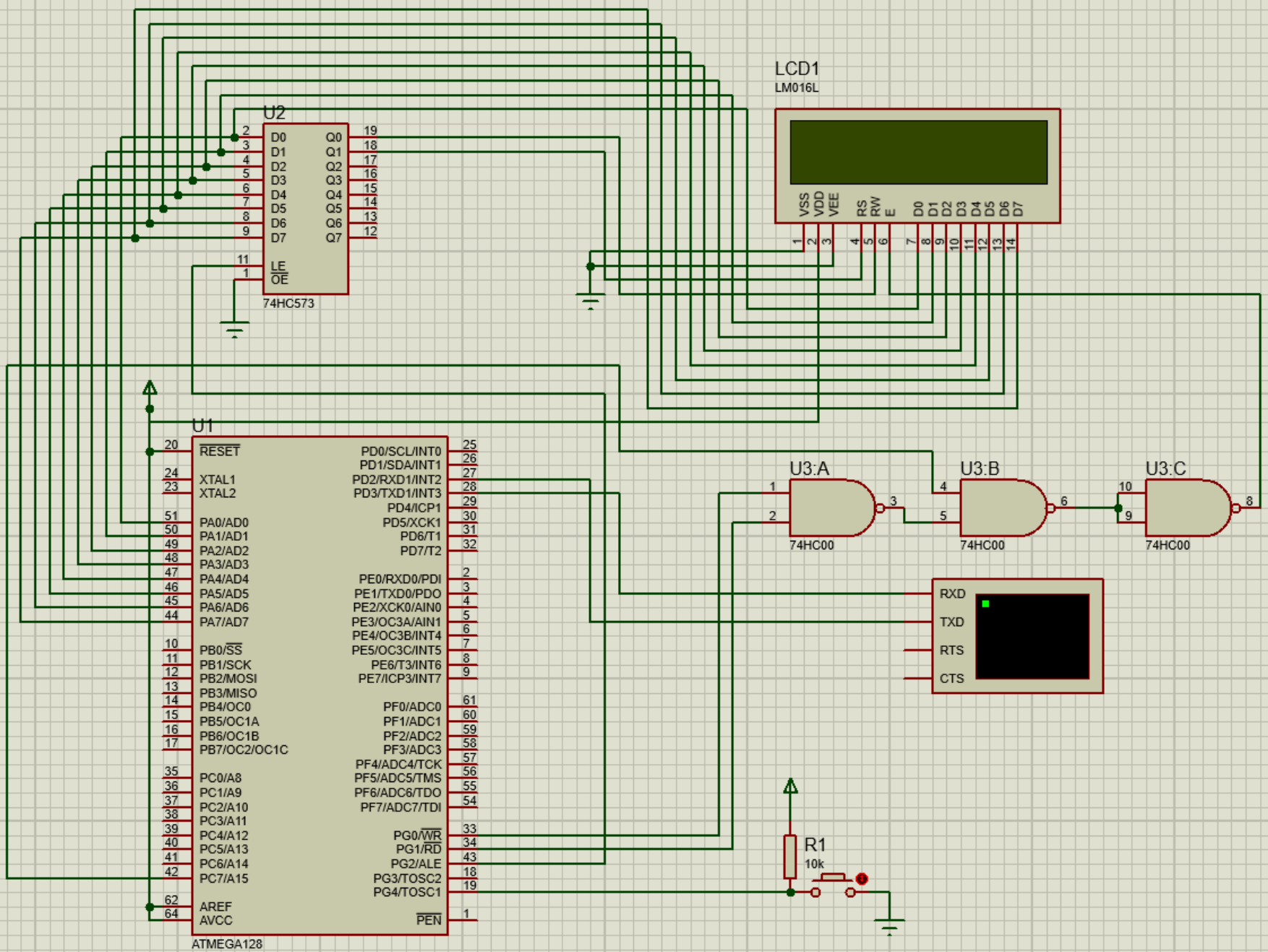
Writing SRE to one enables the External Memory Interface. The pin functions AD7:0, A15:8, ALE, WR, and RD are activated as the alternate pin functions. The SRE bit overrides any pin direction settings in the respective data direction registers. Writing SRE to zero, disables the External Memory Interface and the normal pin and data direction settings are used.

- Bit 6 – SRW10: Wait-state Select Bit**

For a detailed description in non-ATmega103 compatibility mode, see common description for the SRWn bits below (XMCRA description). In ATmega103 compatibility mode, writing SRW10 to one enables the wait-state and one extra cycle is added during read/write strobe as shown in [Figure 14](#).

Addressing of Registers





Addressing of Registers

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
														RS	R/W
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8				0				0				0			

[표 8-3] IR(Instruction Register)을 사용하기 위한 주소

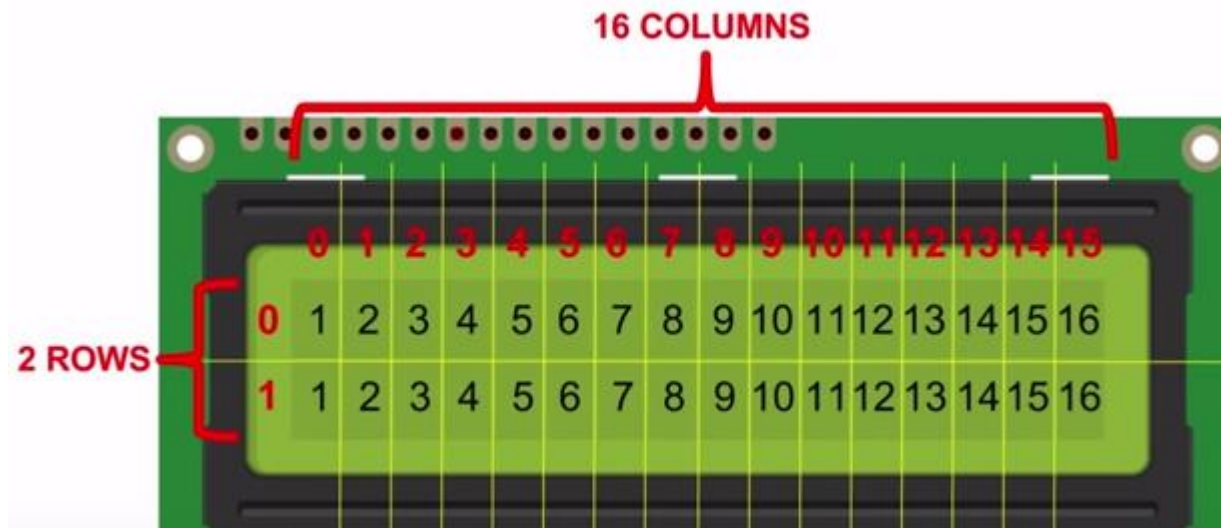
A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
														RS	R/W
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
8				0				0				2			

[표 8-4] DR(Data Register)을 사용하기 위한 주소

Instruction	Instruction Code										Description	Execution time (fosc=270KHz)
	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM and set DDRAM address to "00H" from AC	1.52ms
Return Home	0	0	0	0	0	0	0	0	1	-	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.52ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Assign cursor moving direction and enable the shift of entire display	38μs
Display ON/ OFF Control	0	0	0	0	0	0	1	D	C	B	Set display(D), cursor(C), and blinking of cursor(B) on/off control bit.	38μs
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	-	-	Set cursor moving and display shift control bit, and the direction, without changing of DDRAM data.	38μs
Function Set	0	0	0	0	1	DL	N	F	-	-	Set interface data length (DL: 8-bit/4-bit), numbers of display line (N: 2-line/1-line) and, display font type (F:5x10 dots/5x8 dots)	38μs
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	38μs
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in counter	38μs
Read Busy Flag and Address Counter	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	38μs
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	38μs

DDRAM Address

Display position	Column	1	2	- - -	15	16
DD RAM Address (Hex-Decimal)	1-Line	00H	01H	- - -	0EH	0FH
	2-Line	40H	41H	- - -	4EH	4FH



LCD Command

- 화면 클리어 : LCD 의 화면을 깨끗이 지우고 어드레스 카운터의 값을 '0'으로 만드어 1열 첫 번째 번지부터 글자를 나타낼 수 있다. (실행시간 : 1.53ms)

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	1

- 커서 홈 : 화면을 지우지 않고 어드레스 카운터를 '0'으로 만들어 주고 디스플레이 이 쉬프트 되었던 것을 본래의 위치로 복귀시킨다. (실행시간 : 1.53ms)

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	1	*

- 엔트리 모드 셋 : 커서의 움직이는 방향을 설정하거나, 글자가 표기된 부분을 쉬프트 할 것인지를 결정한다. (실행시간 : 39 μ s)

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	1	I/D	S

- I/D = '1' -> 자동으로 어드레스 증가
- I/D = '0' -> 자동으로 어드레스 감소
- S = '0' 으로 설정하는 것이 정상이고 '1'로 설정하면 화면이 쉬프트 되면서 글자가 쓰여지는 매우 혼란스러운 동작을 한다.

LCD Command

- 디스플레이 ON/OFF : 전체 화면의 ON/OFF, 커서의 ON/OFF, 커서의 위치가 있는 부분의 글자를 깜박이는 것을 설정한다.(실행시간 : 39 μ s)

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	1	0	C	B

- D = '1' 이면, 전체화면이 ON 되어 글자가 나타난다.
- D = '0' 이면, 전체화면이 OFF 되어 어떤 글자도 나타나지 않는다.
- C = '1' 이면, 커서가 나타난다.
- C = '0' 이면, 커서가 나타나지 않는다.
- B = '1' 이면, 커서는 글자위치에서 깜박이게 된다.
- B = '0' 이면, 커서는 깜박이지 않는다.

LCD Command

- 커서 디스플레이 쉬프트 : 현재 LCD 창에 표시된 내용을 좌우로 쉬프트 시키거나 커서를 좌우로 이동시킨다.(실행시간 : $39\mu s$)

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	1	S/C	R/L	*	*

S/C	R/L	동 작 상 태
0	0	커서는 왼쪽으로 쉬프트 된다.
0	1	커서는 오른쪽으로 쉬프트 된다.
1	0	LCD 창에 표시된 내용과 커서가 왼쪽으로 쉬프트 된다.
1	1	LCD 창에 표시된 내용과 커서가 오른쪽으로 쉬프트 된다.

(단, 디스플레이의 쉬프트만 행해질 때 어드레스 카운터는 변하지 않는다.)

LCD Command

- LCD 기능설정 : LCD 가 CPU 와 연결되는 데이터 라인, 디스플레이 라인, 글자 폰트 등을 설정한다.(실행시간 : 39 μ s)

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	DL	N	F	*	*

- DL = '1' 일 때 8비트(D0~D7)의 데이터 라인을 사용한다.
- DL = '0' 일 때 4비트(D0~D7)의 데이터 라인을 사용한다.
- N 은 LCD 디스플레이 라인 수를 설정한다.
- F는 글자 폰트를 결정한다.

N	F	라 인	폰트	비고
0	0	1	5*8	
0	1	1	5*10	
1	0	2	5*8	5*10 도트는 2 Line 디스플레이 불가능

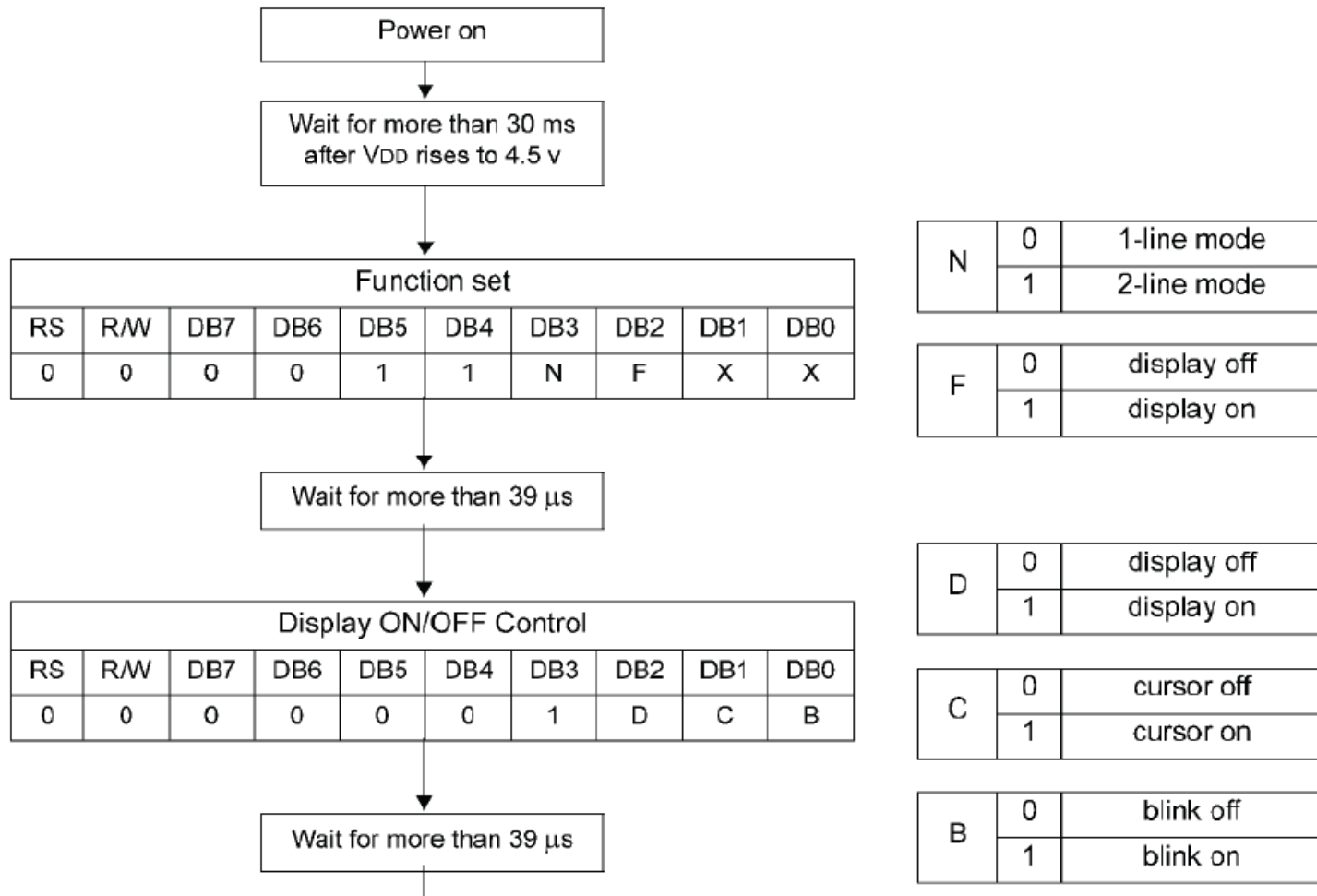
LCD Command

- D.D RAM 어드레스 설정 : 자신이 원하는 곳에 문자를 디스플레이 하기 위해서는 먼저 'Set D.D RAM Address' 명령으로 위치를 지정해 주어야 한다. 램으로 비유하면 데이터가 들어갈 번지를 미리 지정하는 것이 된다. D.D RAM 어드레스는 디스플레이 라인을 설정하는 N 값에 따라 N='0' 이면 D.D RAM 어드레스는 0x00~0x47이다. N='1' 이면 D.D RAM 어드레스는 첫 번째 라인이 0x00~0x27 이고, 두 번째 라인은 0x40~0x67 이다. (실행시간 : 39 μ s)

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0*

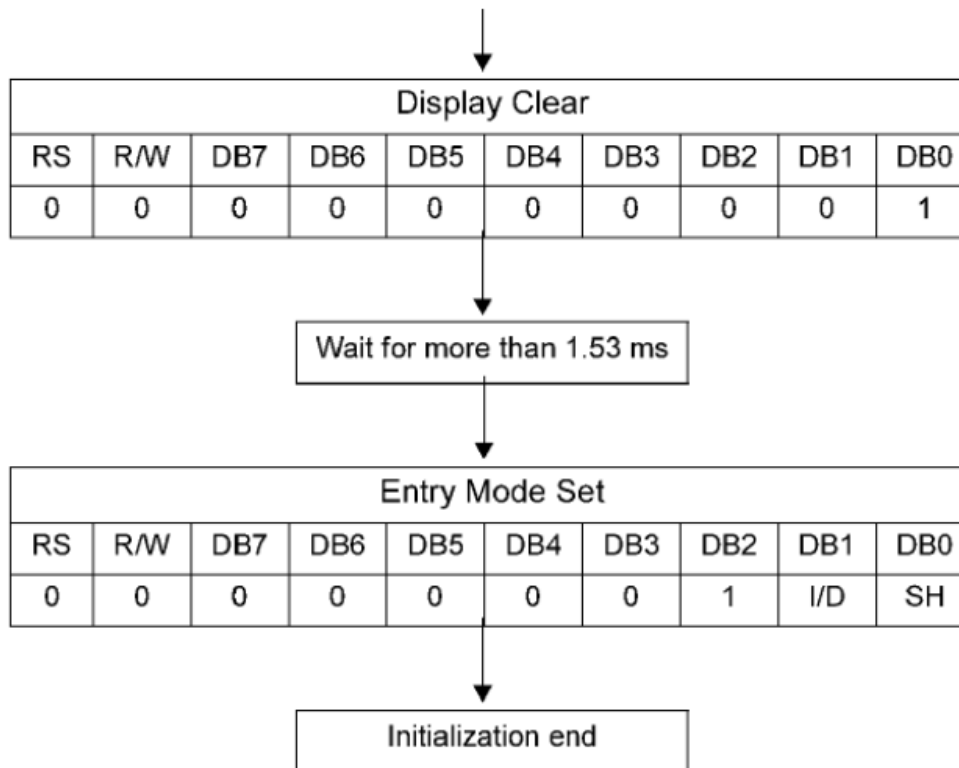
(단, D.D RAM 어드레스 설정은 7비트까지 가능하다.)

Initialization(1)



Initialization(2)

(Continued)



I/D	0	decrement mode
	1	increment mode

SH	0	entire shift off
	1	entire shift on

Sample Code(1)

```
#define F_CPU 16000000U
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#define LCD_INST (*(volatile unsigned char *)0x8000)
#define LCD_DATA (*(volatile unsigned char *)0x8002)

void LCD_data(char ch)
{
    LCD_DATA = ch; // Write data
    _delay_us(50);
}
void LCD_comm(char ch)
{
    LCD_INST=ch; // Write instruction
    _delay_ms(5);
}
void LCD_CHAR(char c) // Display one character
{
    LCD_data(c);
}
void LCD_STR(char *str) // Display a string
{
    while(*str)
        LCD_CHAR(*str++);
}
```

Sample Code(2)

```
void LCD_pos(char col, char row) // Start position
{
    LCD_comm(0x80|(col+row*0x40));
}
void LCD_clear(void)
{
    LCD_comm(1);
}
void LCD_init(void)
{
    LCD_comm(0x38); // Function Set: Data line 8bit, 5X8 dot, 2 Line
    LCD_comm(0x0C); // Display ON/OFF
    LCD_comm(0x06); // Entry Mode Set:
    LCD_clear();
}
int main(void)
{
    char str[20]="LCD test..  ";
    MCUCR = 0x80; // Address and data bus
    LCD_init();
    LCD_pos(0,0);
    sprintf(str,"Hello");
    LCD_STR(str);
    LCD_pos(0,1);
    sprintf(str,"World");
    LCD_STR(str);
}
```

Sample Code

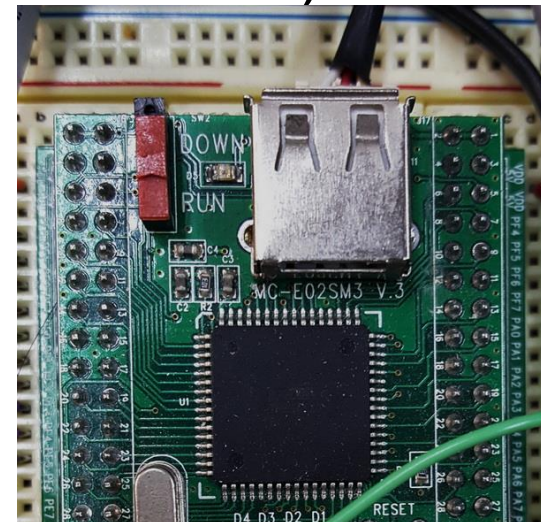
- 배포된 `lcd.c`를 실행하여 LCD에 나타나는 문자를 확인한다.
- 만약 LCD의 문자가 잘 안보일 경우 LCD 옆의 밝기 조절용 `potentiometer`를 `screw driver`로 조절하여 문자가 잘 보이도록 조정한다.
- **중요: 버튼 키 입력을 연결한 PA1의 전선을 제거해야 정상 동작함.(Why?)**

Exercise 1

- LCD의 윗줄에 “Hello World” 나오도록 프로그램을 수정하여 실행한다. LCD의 한 줄은 8자만 보이므로 마지막 3글자가 잘리는 것을 알 수 있다.
- 전광판처럼 LCD의 문자들이 왼쪽으로 흐르도록 한다. 즉, “Hello World Hello World Hello World Hello World Hello World” 와 같이 글자들이 연속적으로 왼쪽으로 흐르도록 한다. 흐르는 속도는 보기 좋을 만큼 적절하도록 정한다.
- LCD의 커서 디스플레이 쉬프트 기능을 사용하지 않는다.

Exercise 2

- 브레드보드에 장착된 버튼을 이용하여, 버튼을 한 번 누를 때 마다, 문자가 흐름이 정지되거나 계속되도록 만든다.
- 주의: **PA1**은 스위치 입력으로 사용할 수 없으므로, 스위치 입력은 **PG4**로 바꾼다. **PG4**단자와 버튼 키를 연결 한다. **PG4**를 사용할 경우 **DOWN/RUN** 스위치를 **RUN**으로 설정해야 함(**PG4**를 공유함).



PG4-Button Key

