



AVR – 8bit LCD

임베디드스쿨2기

Lv1과정

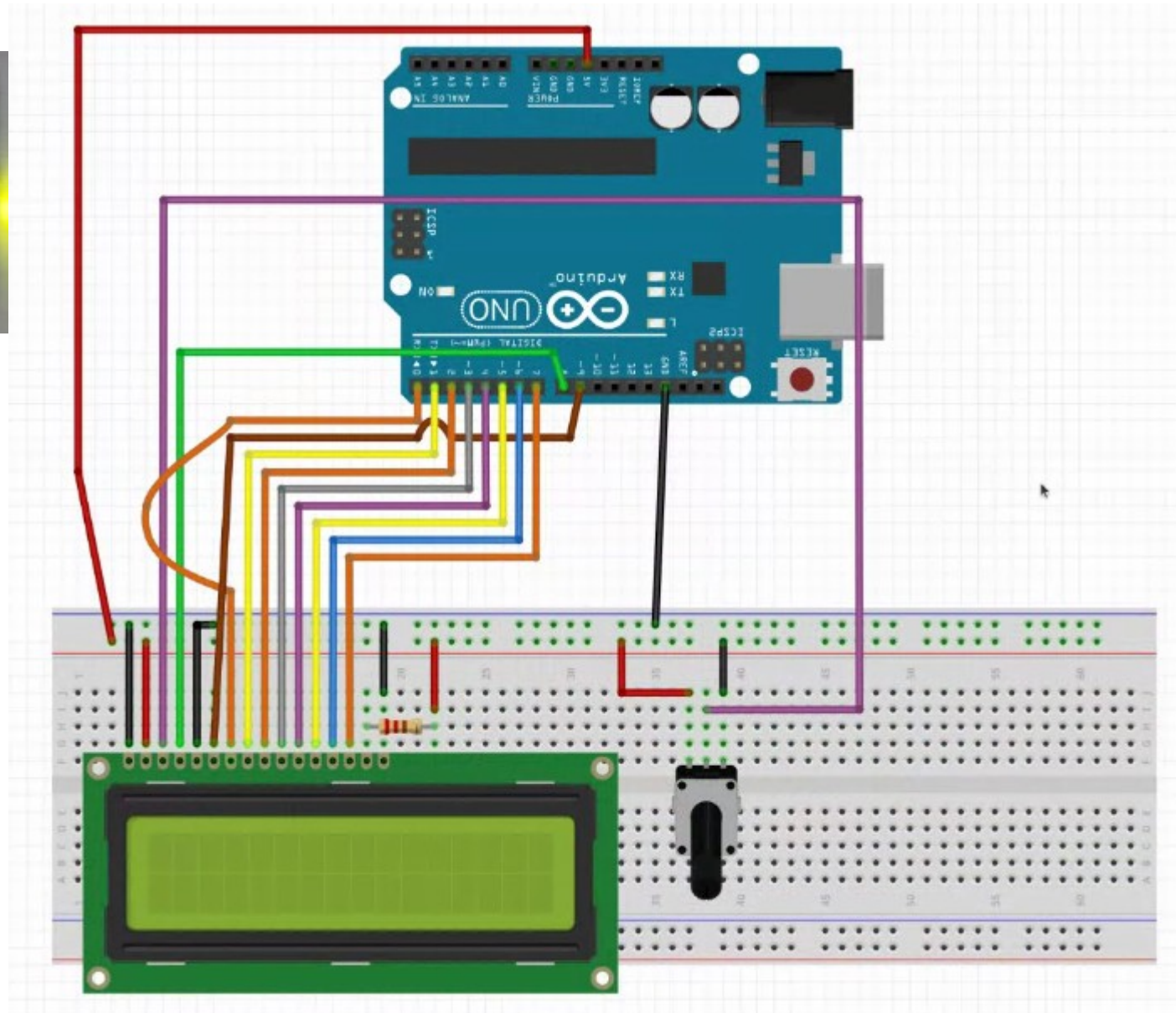
2021. 07. 23

박태인

LCD H/W 연결



- 1602a LCD
- 2 x 16 LCD
- 2줄 표현 방식의 캐릭터 LCD



LCD 코딩(header 선언)

main.c

```
1  /*
2  */
3  #define F_CPU 16000000UL
4  #include <avr/io.h>
5  #include <util/delay.h>
6
7  #include <string.h>
8
9  // 데이터 연결 포트
10 #define PORT_DATA PORTD
11 // 데이터 제어 포트
12 #define PORT_CONTROL PORTB
13 // 데이터의 방향
14 #define DDR_DATA DDRD
15 // 제어 핀의 방향
16 #define DDR_CONTROL DDRB
17 // RS 제어 핀의 비트 번호
18 #define RS_PIN 0
19 // E 제어 핀의 비트 번호
20 #define E_PIN 1
21
22 #define COMMAND_CLEAR_DISPLAY 0x01
23 // 8비트 2라인 5x8 폰트
24 #define COMMAND_8_BIT_MODE 0x38
25
26 #define COMMAND_DISPLAY_ON_OFF_BIT 2
27 #define COMMAND_CURSOR_ON_OFF_BIT 1
28 #define COMMAND_BLINK_ON_OFF_BIT 0
29
```

커서 Option 관련

- 데이터의 방향 DDRD
- 데이터 연결 포트 PORTD
- 제어 핀의 방향 DDRB
- 제어 포트 PORTB
- RS 제어 핀 비트 '0'
- E 제어 핀 비트 '1'

화면 clear 명령 : 0x01, function set : 0x38 [8bit, 2 Line, 5x8 폰트]

Instruction	Instruction Code										Description	Execution time (fosc=270KHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "00H" to DDRAM and set DDRAM address to "00H" from AC	1.53ms
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:5×11 dots/5×8 dots)	39 μs

LCD 코딩(Write)

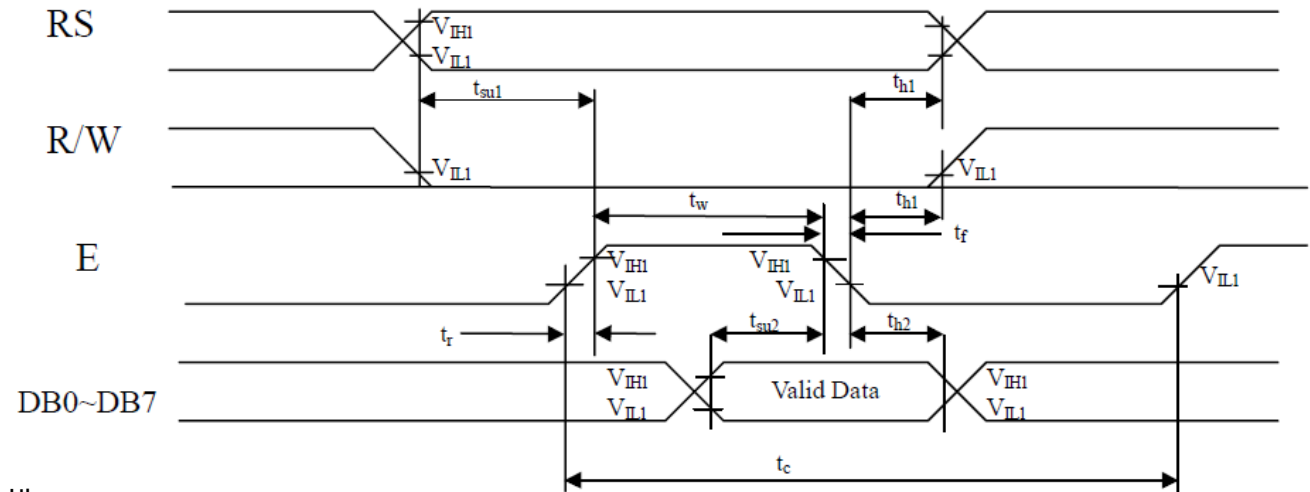
Lcd_pulse_enable

```

30 // Falling edge 에서 동작
31 void lcd_pulse_enable(void)
32 {
33     // E 를 HIGH
34     PORT_CONTROL |= (1 << E_PIN);
35     _delay_ms(1);
36     // E를 LOW 로 다시 떨어줌
37     PORT_CONTROL &= ~(1 << E_PIN);
38     _delay_ms(1);
39 }
40

```

E 제어 핀을 High -> Low 동작함으로써 동작실행 (Valid Data, 유효 데이터)



Lcd_write_data

```

41 void lcd_write_data(uint8_t data)
42 {
43     // 문자 출력에서 RS는 1
44     PORT_CONTROL |= (1 << RS_PIN);
45     // 출력 할 내용
46     PORT_DATA = data;
47     // 문자 출력!
48     lcd_pulse_enable();
49 }
50

```

RS 제어 핀을 High 로 놓은 상태에서 data write!

Instruction	Instruction Code										Description	Execution time (fosc=270Khz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43 μ s

LCD 코딩(Write)

Lcd_write_command

```

51 void lcd_write_command (uint8_t command)
52 {
53     //명령 실행시 RS 는 0
54     PORT_CONTROL &= ~(1 << RS_PIN);
55     // 데이터 핀에 명령어 전달
56     PORT_DATA = command;
57     // 명령 실행
58     lcd_pulse_enable();
59 }

```

Command 명령어 실행 시 RS 값이 Low(0) 임을 확인.

Lcd_clear

```

void lcd_clear(void)
{
    lcd_write_command(COMMAND_CLEAR_DISPLAY);
    _delay_ms(2); // 데이터 시트상 1.52ms 소요
}

```

클리어 명령 0x01 진행.
Delay 시간 우측 데이터 시트 정보 활용.

Instruction	Instruction Code										Description	Execution time (fosc=270KHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "00H" to DDRAM and set DDRAM address to "00H" from AC	1.53ms
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.	1.53ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction and enable the shift of entire display.	39 μ s
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.	39 μ s
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.	39 μ s
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:5 \times 11 dots/5 \times 8 dots)	39 μ s
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39 μ s
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.	39 μ s
Read Busy Flag and Address	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.	0 μ s
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43 μ s
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43 μ s

LCD 코딩(lcd_init)

Lcd_init

```
void lcd_init(void)
```

```
{
```

```
_delay_ms(50);
```

초반 딜레이

```
// 연결 핀을 출력으로 설정 !
```

```
DDR_DATA = 0xFF;
```

```
DDR_CONTROL |= (1<< RS_PIN) | (1<< E_PIN);
```

연결핀을 출력.
RS, E 제어핀 모두 High

```
// 8비트 모드 (0x38)
```

```
// 8비트, LCD 표시 2 라인, 5x7 스타일
```

```
lcd_write_command(COMMAND_8_BIT_MODE);
```

Command 함수로 LCD 출력 옵션 설정.

```
// display 모드(ON/OFF) 제어
```

```
// 화면 on. 커서 off. 커서 깜빡임 off
```

```
// 8 이랑 4 가 or 해서 c 가 된다. 1 << COMMAND(2) 는 2를 쉬프트 한거니 4가 된다.
```

```
uint8_t command = 0x08 | (1<< COMMAND_DISPLAY_ON_OFF_BIT);
```

```
lcd_write_command(command);
```

COMMAND_DISPLAY를 define 한 것을 활용 하였고,
이것을 설정들에 맞는 값(0x0c)을 만들기 위해
0x08 에 2를 쉬프트 한 값(4)를 OR 하면 0x0C 완성!

```
// 화면 지움
```

```
lcd_clear();
```

↳ 이렇게 함으로써 코드의 가독성이 생길 수 있다.

```
// Entry Mode Set
```

```
// 출력 후 커서를 옮김
```

```
// 주소가 증가하며 화면 이동은 없음
```

```
lcd_write_command(0x06);
```

```
// 이 것을 설정 했기에 다음 커서로 넘어가는 명령어 없이도 순서대로 출력 된다.
```

→ 이것을 설정 함으로써 커서의 다른 명령 없이도 순서대로 출력 시킴

Instruction	Instruction Code										Description	Execution time (fosc=270KHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "00H" to DDRAM and set DDRAM address to "00H" from AC	1.53ms
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.	1.53ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction and enable the shift of entire display.	39 μs
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.	39 μs
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.	39 μs
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:5×11 dots/5×8 dots)	39 μs
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39 μs
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.	39 μs
Read Busy Flag and Address	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.	0 μs
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43 μs
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43 μs

LCD 코딩(문자열 출력)

Lcd_write_string

```
94
95 void lcd_write_string(char *string)
96 {
97     uint8_t i;
98
99     // 종료 문자 까지
100    for(i=0; string[i]; i++)
101    {
102        // 문자 단위 출력
103        lcd_write_data(string[i]);
104    }
105 }
```

한 글자 씩 출력하는 write 함수를

Char 포인터 인자로 값을 받아

For문으로 문자열 형태로 출력.

Lcd_write_animation_string

```
106
107 void lcd_write_animation_string(char *string)
108 {
109     uint8_t i, j;
110     int len = strlen(string);
111
112     // 종료 문자 까지
113     for(i=0; i<len; i++)
114     {
115         for(j=i; string[j]; j++)
116         {
117             // 문자 단위 출력
118             lcd_write_data(string[j]);
119         }
120         _delay_ms(500);
121         lcd_clear();
122     }
123 }
124
```

문자열 출력하는 함수에서

For 문을 하나 더 실행 시켜 글자가 흘러가듯 실행 시킴

Len 변수에 문자열 길이 return.

For문 진행을 차근차근 생각해 보자.

Ex) len = 3 이라면.

초기 I = 0 , 초기 j = 0, string[0] 출력, j++ j=1;

string[1] 출력, j++ j =2;

String[2] 출력, j++ j =3;

String[4] = NULL 출력, 내부 for문 종료

0.5초간 현 상태 유지 후, lcd 클리어.

그리고 다시 i++ i=1이 되고 위의 과정을 lcd의 글자수 만큼 반복.

결과적으로 자리를 옮겨 가며,

글자를 글자 수 만큼 지웠다 생겼다 를 반복 시킴.

LCD 코딩(커서 좌표 이동)

Lcd_goto_xy

```

126 void lcd_goto_xy(uint8_t row, uint8_t col)
127 {
128     // [0 15]
129     col %= 16; // 나머지 연산
130     // [0 1]
131     row %= 2;
132
133     // 첫라인 시작 주소는 0x00
134     // 두번째 라인 시작 주소는 0x40
135     uint8_t address = (0x40 * row) + col;
136     uint8_t command = 0x80 + address;
137
138     // 커서 이동
139     lcd_write_command(command);
140 }
141

```

2 x 16
즉, 2 Line 16 글자의 LCD 시스템 좌표이다.
좌표를 지정하기 위한 col, row 값



커서 위치의 주소 값 확인

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

2-Line by 16-Character Display

Write를 하기 위한 커서 이동

Instruction	Instruction Code										Description	Execution time (fosc=270KHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43 μ s

LCD 코딩(main)

main

```
143 int main(void)
144 {
145     lcd_init();           Lcd init
146
147     lcd_write_string("Hello World");
148     _delay_ms(5000);      Hello World 초기 글씨 5초 유지
149
150     lcd_clear();          Lcd Clear
151
152     lcd_goto_xy(0, 0);
153     lcd_write_data('1');
154
155     lcd_goto_xy(0, 5);
156     lcd_write_data('2');
157
158     lcd_goto_xy(1, 0);
159     lcd_write_data('3');
160
161     lcd_goto_xy(1, 5);
162     lcd_write_data('4');
163     // Insert code
164
165     while(1)
166     {
167         lcd_write_animation_string("Embedded Master Lv1!")
168         ;
169     }
170
171     return 0;
172 }
173
```

각 goto 좌표에 따라

숫자 1, 2, 3 ,4 를 출력

Embedded Master Lv1! 글씨를
좌측으로 글씨가 움직이는
Animation 효과 출력 반복.