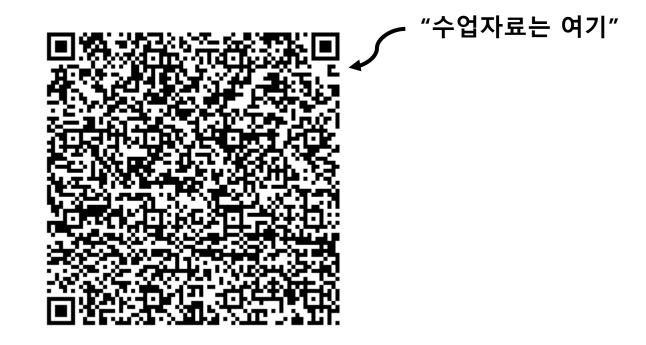
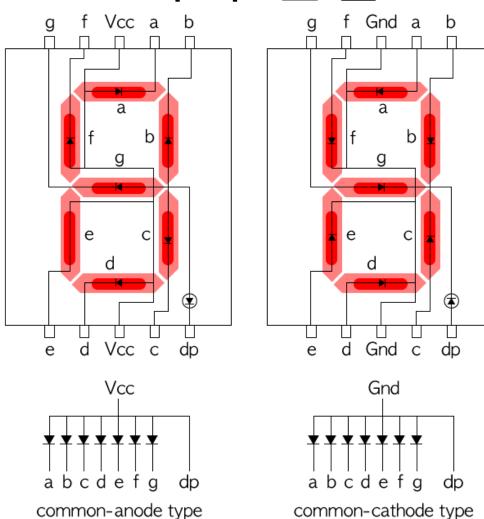
# 아두이노 LCD 실험



## 8bit data 전달 실험 LED를 이용한 포트 제어 실험

• 7-segment 실험

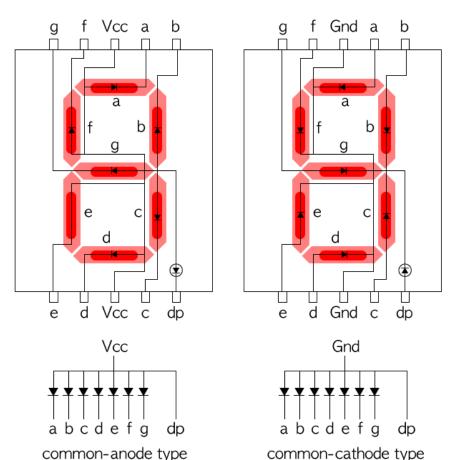




## 8bit data 전달 실험 LED를 이용한 포트 제어

• 7-segment 실험

common-anode type

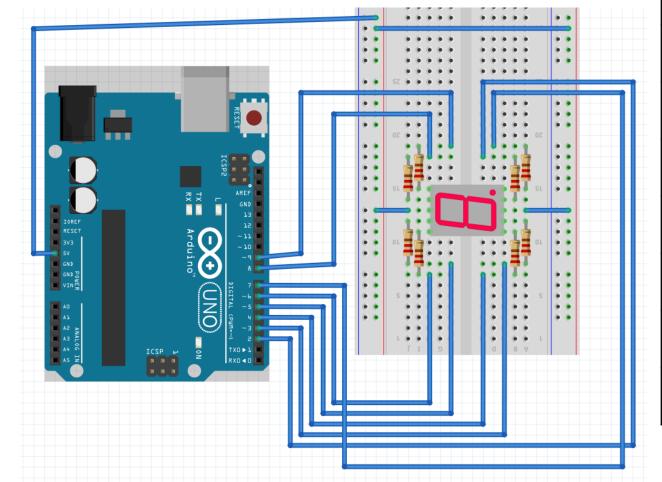


#### **Common-anode type**

$$0 = a(0) b(0) c(0) d(0) e(0) f(0) g(1) DP(1)$$
  
 $1 = a(1) b(0) c(0) d(1) e(1) f(1) g(1) DP(1)$   
 $2 = a(0) b(0) c(1) d(0) e(0) f(1) g(0) DP(1)$   
 $3 = a(0) b(0) c(0) d(0) e(1) f(1) g(0) DP(1)$   
 $4 = a(1) b(0) c(0) d(1) e(1) f(0) g(0) DP(1)$   
 $5 = a(0) b(1) c(0) d(0) e(1) f(0) g(0) DP(1)$   
 $6 = a(0) b(1) c(0) d(0) e(0) f(0) g(0) DP(1)$   
 $7 = a(0) b(0) c(0) d(1) e(1) f(0) g(1) DP(1)$   
 $8 = a(0) b(0) c(0) d(0) e(0) f(0) g(0) DP(1)$   
 $9 = a(0) b(0) c(0) d(0) e(1) f(0) g(0) DP(1)$   
 $0 = a(0) b(0) c(0) d(0) e(1) f(0) g(0) DP(1)$ 

## 8bit data 전달 실험 LED를 이용한 포트 제어 실험

• 7-segment 실험



a	$\rightarrow$	PB0
b	$\rightarrow$	PB1
С	<b>&gt;</b>	PD2
d	$\rightarrow$	PD3
е	$\rightarrow$	PD4
f	$\rightarrow$	PD5
g	$\rightarrow$	PD6
DP	$\rightarrow$	PD7

## 8bit data 전달 실험 LED를 이용한 포트 제어 실험

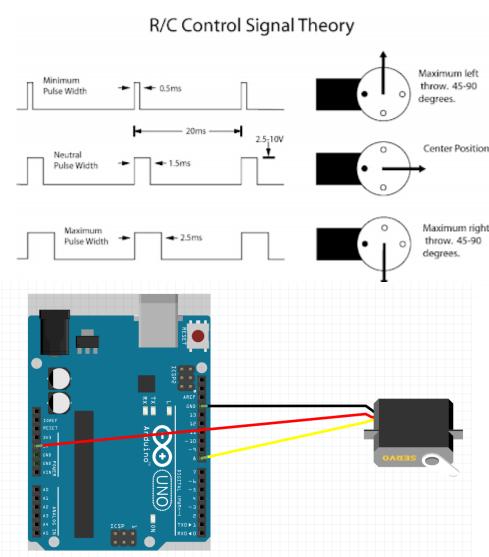
• 7-segment 실험

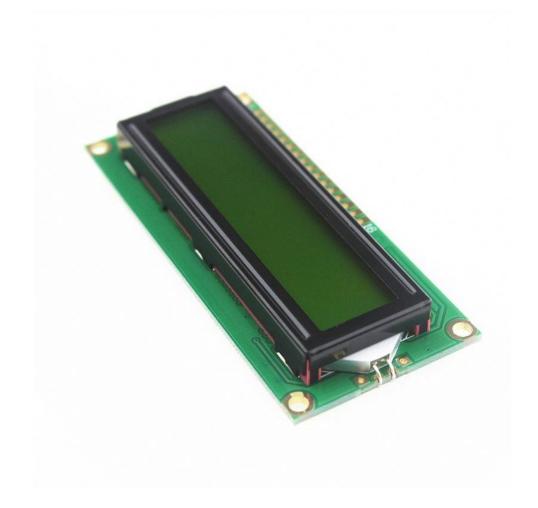
```
0 = a(0) b(0) c(0) d(0) e(0) f(0) g(1) DP(1)
1 = a(1) b(0) c(0) d(1) e(1) f(1) g(1) DP(1)
2 = a(0) b(0) c(1) d(0) e(0) f(1) g(0) DP(1)
3 = a(0) b(0) c(0) d(0) e(1) f(1) g(0) DP(1)
4 = a(1) b(0) c(0) d(1) e(1) f(0) g(0) DP(1)
5 = a(0) b(1) c(0) d(0) e(1) f(0) g(0) DP(1)
6 = a(0) b(1) c(0) d(0) e(0) f(0) g(0) DP(1)
7 = a(0) b(0) c(0) d(1) e(1) f(0) g(1) DP(1)
8 = a(0) b(0) c(0) d(0) e(0) f(0) g(0) DP(1)
9 = a(0) b(0) c(0) d(0) e(1) f(0) g(0) DP(1)
. = DP(0)
```

		а	b	С	d	е	f	g	DP
		PB0	PB1	PD2	PD3	PD4	PD5	PD6	PD7
0	<b>&gt;</b>	0	0	0	0	0	0	1	1
1	<b>→</b>	1	0	0	1	1	1	1	1
2	<b>→</b>	0	0	1	0	0	1	0	1
3	<b>→</b>	0	0	0	0	1	1	0	1
4	<b>→</b>	1	0	0	1	1	0	0	1
5	<b>→</b>	0	1	0	0	1	0	0	1
6	$\rightarrow$	0	1	0	0	0	0	0	1
7	<b>→</b>	0	0	0	1	1	0	1	1
0	<b>→</b>	0	0	0	0	0	0	0	1
9	$\rightarrow$	0	0	0	0	1	0	0	1

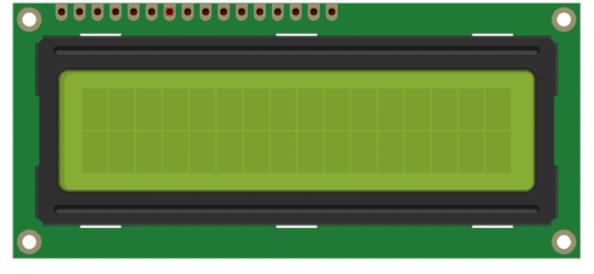
# Digital Timing Level 실험(PWM)

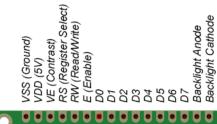
```
void setup() {
 pinMode(8, OUTPUT);
void loop() {
 digitalWrite(8, HIGH);
 delayMicroseconds(1400);
 digitalWrite(8, LOW);
 delayMicroseconds(20000-1400);
```

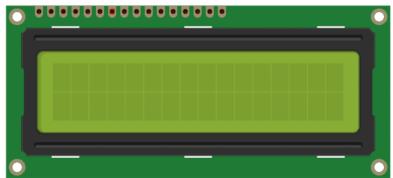




VDD (5V)
VE (Contrast)
RS (Register Select)
RW (Read/Write)
D0
D1
D2
D3
D4
D5
D6
D7
Backlight Anode
Backlight Cathode

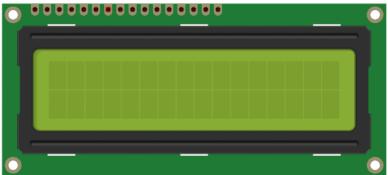






- VSS(Ground): LCD에 전원을 인가하는 단자로, 0V(GND)에 연결.
- VDD(5V): LCD에 전원을 인가하는 단자로, +5V에 연결.
- **VE(Contrast)** : LCD의 밝기를 조절하는 단자로서, 10kΩ의 가변 저항을 연결하여 밝기를 조정. 밝기 조정을 하지 않으려면 GND에 연결.
- RS(Register Select) : 레지스터 종류를 선택
  - 0: 명령 레지스터
  - 1: 데이터 레지스터
- RW(Read/Write) : 데이터 혹은 명령을 읽는지 쓰는지 설정
  - 0 : 레지스터의 데이터를 씀. (Write : 아두이노 → LCD)
  - 1 : 레지스터에 데이터를 읽음. (Read : 아두이노 ← LCD)
- **E(Enable)**: Enable 신호(LCD동작 허가)
  - 0 : LCD 동작 X
  - 1: LCD 동작 O
- **D0~D7(Data Bus)**: 아두이노와 LCD 사이에 데이터를 주고받기 위한 데이터 핀. 만약 4비트를 사용할 경우에는 D4~D7만 사용.
- Backlight Anode: 백 라이트의 전원 단자로서 보통 저항과 IN4001(다이오드)을 연결
- Backilight Cathode : 백 라이트의 전원 단자로서 GND에 연결.





#### • 2개의 8-bit 레지스터

- RS(Register Select) 신호로 어떤 레지스터를 사용할 것인지 선택
- 명령 레지스터 (Instruction Register, IR)
  - D.D.RAM, C.G.RAM에 대한 주소정보, 클리어, 커서 이동등 LCD 제어 명령
- 데이터 레지스터 (Data Register, DR)
  - D.D.RAM, C.G.RAM에 써넣은 데이터나 읽어낸 데이터를 일시적으로 저장

#### D.D.RAM(Display Data RAM)

- 80x8비트 용량으로 80개의 8비트 아스키(ASCII)코드를 저장할 수 있다.
- 0x00~0F 주소가 LCD의 1행의 1~16째.
- 0x40~4f 주소가 LCD의 2행의 1~16번째 문자로 표시 된다.
- 빈 주소에는 자유롭게 RAM 데이터 메모리로 사용이 가능하다.

#### C.G.RAM(Charactor Generator RAM)

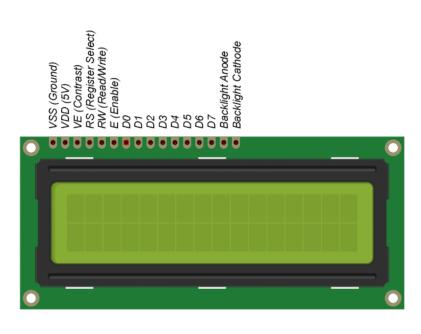
- 사용자가 문자를 만들 때 사용하는 메모리로 5x7은 8개, 5x10은 4개를 만들어서 저장이 가능.
- C.G.ROM((Charactor Generator ROM): 5x7, 5x10의 도트문자를 내장하고 있음. 아스키코드와 일치

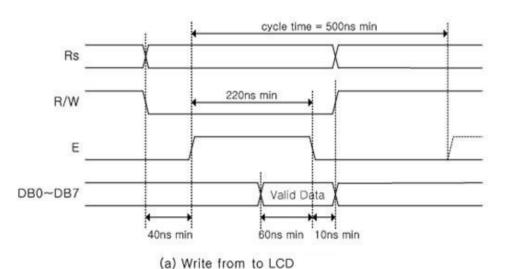
#### Address Counter (AC)

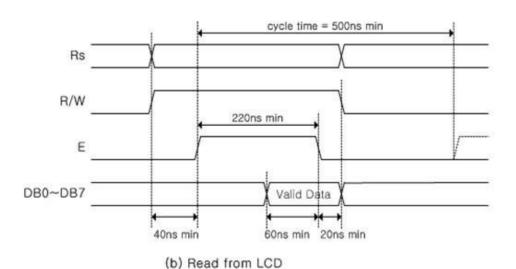
• 주소를 저장하는 레지스터로 D.D.RAM과 C.G.RAM의 주소를 지정할 때 사용, 명령레지스터에 주소정보를 지정하면 AC로 주소정보가 전달 되고, D.D.RAM에 데이터를 쓰면 AC는 자동으로 (설정 값에 따라) 1증가하거나 1감소한다.

#### Busy Flag (BF)

- Busy flag가 '1' → 내부에서 LCD가 명령을 처리 중으로 명령을 받을 수 없음.
- Busy flag가 '0' → 명령 가능.
- Busy flag를 확인 하려면? RS=0, R/W=1일 때, D7핀으로 출력









0x38 = **001**(기능셋)1 1000

명명	명명						Glole	:1				설명	실행	
08	08	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	20	시간	
	화면지우기	0	0	0	0	0	0	0	0	0	1	화면지우기, 커서홈	1,52ms	
	커서홈	0	0	0	0	0	0	0	0	1	-	커서 처음 위치로 이동	1,52ms	
	엔트리	0	0	0	0	0		0	1	I/D	s	어드레스자동증가/감소(I/D)	27.10	
	모드 셋	ľ	"	"	ľ	Ů	0	ľ	'	1/0	3	표시 쉬프트(S)	37us	
	표시 On/Off	0	0	0	0	n	n	1	2	c	В	디스플레이(D), 커서(C),	37us	
	제어	Ů	Ů	0 0 0 0 0 1 D C B						)	깜박임(B) On/Off	3103		
명령	표시, 커서	0	0	0	0	0	1	s/c	B/I	_	_	표시, 커서 이동	37us	
쓰기	쉬프트		Ľ	Ŭ	Ŭ	Ŭ	Ţ,	0, 0	.,, _			40, 710 018	0,00	
	기능 셋	0	0	0	0	1	DL	N	F	_	_	인터페이스라인(DL),	37us	
	710 X		Ŭ	O O I DL N P					Ċ		라인수(N), 문자폰트(F)			
	CGRAM	0	0	0	0 1 CGRAM 어드레스(ACG)						CGRAM 어드레스 설정	37us		
	어드레스				O T CONAM OLDINACO									
	DDRAM	0	0	1 DDRAM 어드레스(ADD)								DDRAM 어드레스 설정	37us	
	어드레스			bertain of Eq. (1999)										
명령	비지체크,	0	1	BF		оч⊆	레스	카운6	H(AC	)		비지플래그 읽기	Ous	
윍기	어드레스											어드레스 카운터 읽기		
데이터	데이터 쓰기	1	0			wi	rite d	ata				CGRAM 또는 DDRAM에	37us	
쓰기												데이터 쓰기		
데이터	데이터 읽기	1	1			ге	ad d	ata				CGRAM 또는 DDRAM에서	37us	
윍기	511 51 51 51 F	·	·									데이터 읽기		
I/D=1:0	I드레스 자동증	가		I/D=	0 : 어	드레스	스 자용	동감소				DDRAM : 표시 데이터 RAM		
S=1 : 전기	S=0 : 쉬프트 하지 않음								CGRAM : 폰트 제작 RAM					
S/C=1 : 표시 쉬프트				S/C	=0:5	9 M F	I동				ACG : CGRAM 어드레스			
R/L=1 : 오른쪽으로 쉬프트					: 0 : 온	<u>!쪽으</u>	로쉬.	ᄪᇀ			ADD : DDRAM 어드레스			
DL=1:88  <b>⊆</b>					O:46	IJ트					AC : 어드레스 카운터			
N=1 : 2강	·인			N=0	: 1감	인					(DDRAM, CGRAM 어드레스)			
F=1 : 5x1	0 dots			F=0:5x8 dot										
BF=1 : LH	부 동작중			BF=0 : 명령/데이터 받기 가능										

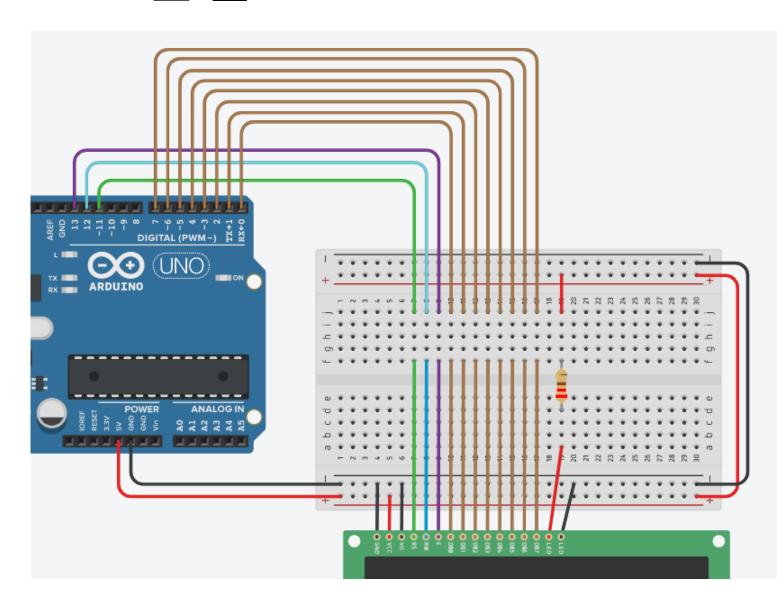
#### • 회로 구성

• RS: 11번핀(PORTB3)

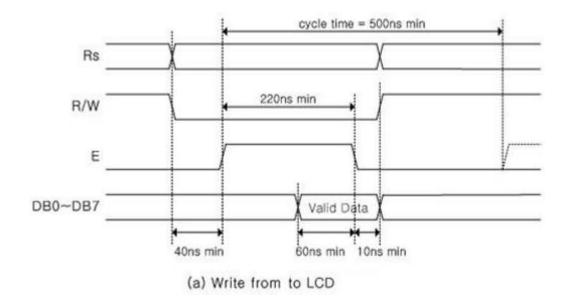
• RW: 12번핀(PORTB4)

• E: 13번핀(PORTB5)

• D0~D7: 0번핀~7번핀)PORTD



• LCD 초기화(명령 Write to LCD)



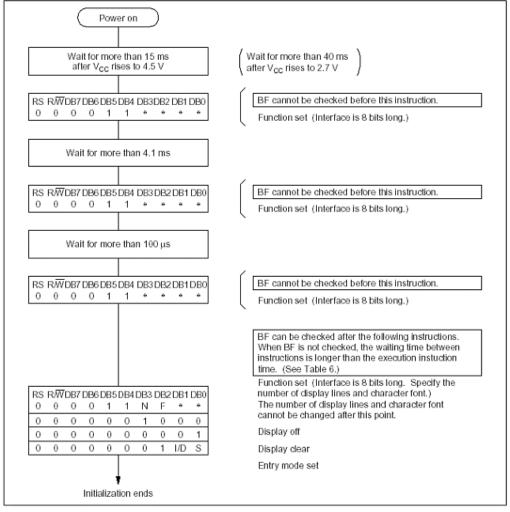
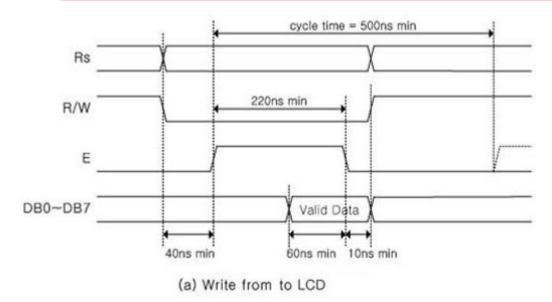


Figure 23 8-Bit Interface

• LCD 초기화

RS(Register Select): 0 → 명령 레지스터



Power on Wait for more than 15 ms Wait for more than 40 ms after V<sub>CC</sub> rises to 2.7 V after V<sub>CC</sub> rises to 4.5 V BF cannot be checked before this instruction. WDB7 DB6 DB5 DB4 DB3DB2 DB1 DB0 0 0 1 1 + + + + Function set (Interface is 8 bits long.) Wait for more than 4.1 ms RS RWDB7DB6DB5DB4DB3DB2DB1DB0 BF cannot be checked before this instruction. 0 0 0 0 1 1 . . . . Function set (Interface is 8 bits long.) Wait for more than 100 us RS RWDB7DB6DB5DB4DB3DB2DB1DB0 BF cannot be checked before this instruction. 0 0 0 0 1 1 \* \* \* \* Function set (Interface is 8 bits long.) BF can be checked after the following instructions. When BF is not checked, the waiting time between instructions is longer than the execution instuction time. (See Table 6.) Function set (Interface is 8 bits long. Specify the RS RWDB7DB6DB5DB4DB3DB2DB1DB0 number of display lines and character font.) The number of display lines and character font 0 0 0 0 1 1 N F + + cannot be changed after this point. 0 0 0 0 0 0 1 0 0 0 Display off 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 I/D S Display clear Entry mode set Initialization ends

Figure 23 8-Bit Interface

• LCD 초기화  $RW(Read/Write) : 0 \rightarrow Write$ cycle time = 500ns min 220ns min R/W DB0~DB7 Valid Data 40ns min 60ns min 10ns min (a) Write from to LCD

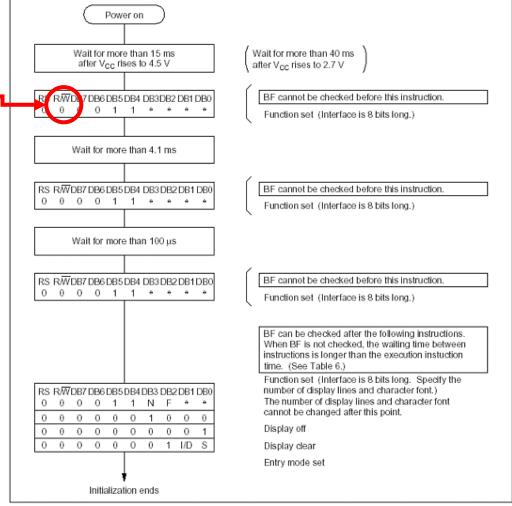
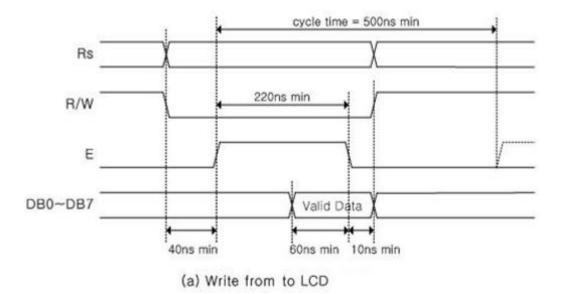


Figure 23 8-Bit Interface

• LCD 초기화

#### DB5 bit가 1이면 "기능셋 " 설정



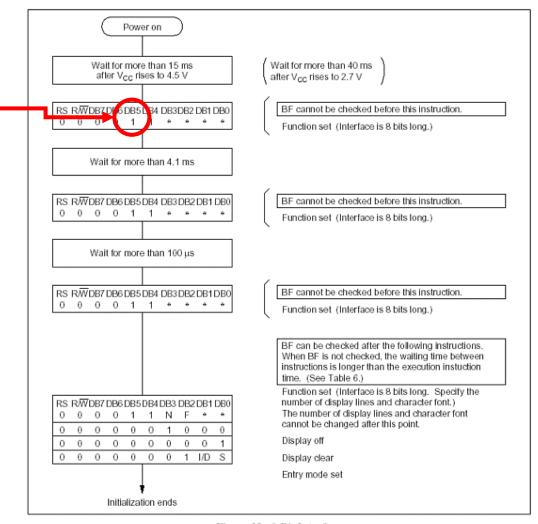
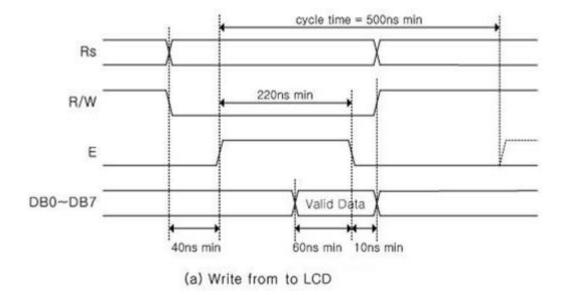


Figure 23 8-Bit Interface

• LCD 초기화

DL(DB4 bit) : Data Line : 1 → 8Bit



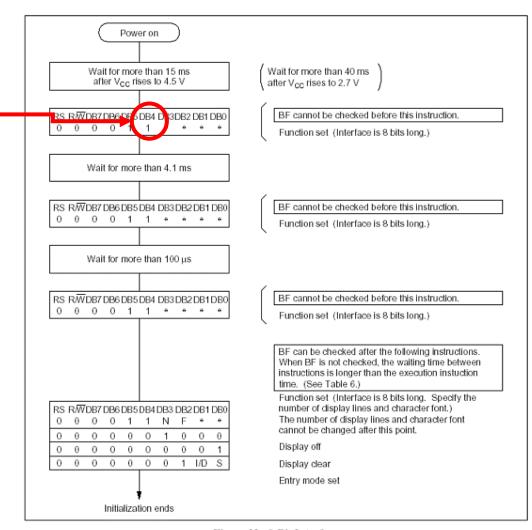
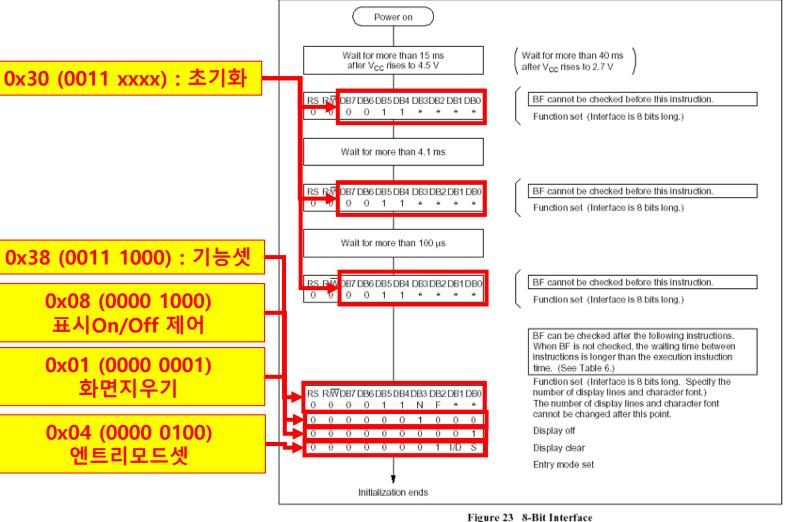
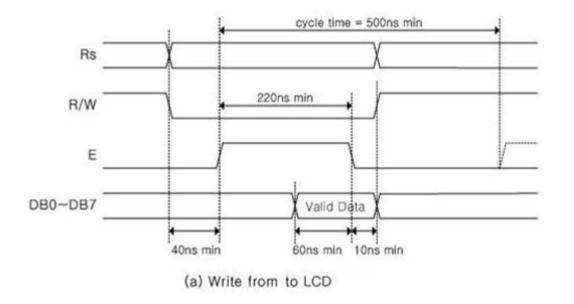


Figure 23 8-Bit Interface

• LCD 초기화

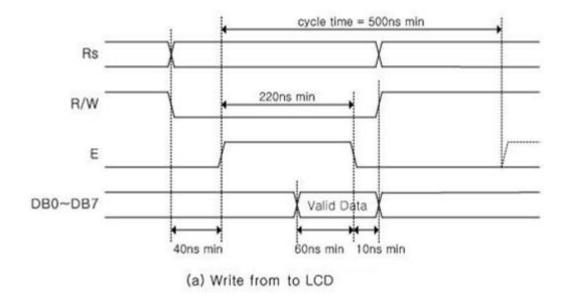
함께 전체	명령	명령		데이터								설명	실행		
	9.9	88	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DBO	28	시간	
변호의		화면지우기	0	0	0	0	0	0	0	0	0	1	화면지우기, 커서홈	1,52ms	
모든 셋 이 이 이 이 이 이 이 이 이 이 이 이 이 이 이 이 이 이		커서홈	0	0	0	0	0	0	0	0	1	-	커서 처음 위치로 이동	1,52ms	
평형			0	0	0	0	0	0	0	1	I/D	s		37us	
지하는			0	0	0	0	0	0	1	D	С	В		37us	
지형 생 이 이 이 이 이 이 이 이 이 이 이 이 이 이 이 이 이 이		.,	0	0	0	0	0	1	s/c	R/L	-	-	표시, 커서 이동	37us	
어 등 해 수 명		기능 셋	0	0	0	0	1	DL	N	F	-	-		37us	
명명			0	0	0 1 CGRAM 메드레스(ACG)							CGRAM 어드레스 설정	37us		
왕기 어드레스 이 1 BF 어드레스 카운터(AC) 어드레스 카운터 왕기 이상 어드레스 카운터 왕기 이상 어드레스 카운터 왕기 이상 어드레스 카운터 왕기 이 어드레스 카운터 왕기 의장에 어드레스 기 이 이 어드레스 카운터 왕기 이 어드레스 자동강소 어디에서 경기 이 어드레스 자동강소 어디에서 공기 이 어드레스 자동강소 어디에서 유제에 무료를 하지 않을 어른 하지 않을 어른 이 어드레스 자동에 어드레스 무시크를 모르는 이 어드레스 자동강소 이 어드레스 자동 이 어드레스 자동 이 어드레스 자동 이 어드레스 가장이 어드레스 가장이 어드레스 이 어드레스 가운터 이 어드레스 카운터 이 어드레스 아들레스 아들레스 아들레스 아들레스 아들레스 아들레스 아들레스 아들			0	0	1 DDRAM 에드레스(ADD)								DDRAM 어드레스 설정	37us	
변기 에이터 쓰기 1 0 write data 데이터 쓰기 37Us 데이터 쓰기 1 1 1 1 read data 데이터 쓰기 37Us 데이터 함기 1 1 1 read data GIOI터 쓰기 37Us 데이터 함기 1 1 1 read data GIOI터 쓰기 37Us 데이터 함기 37Us 대한 전체 쉬프트 S=0: 쉬프트 하지 않음 CGRAM : 폰트 제작 RAM CGRAM 아드레스 R/L=1: 오른쪽으로 쉬프트 R/L=0: 왼쪽으로 쉬프트 ADD: DDRAM 아드레스 DL=1: 8번트 DL=0: 4번트 AC: 어드레스 카운터 AC: 어드레스 카운터 (DDRAM, CGRAM 어드레스)			0	1	BF 어드레스 카운터(AC)									0us	
위기   에이터 위기   1   1   read data   에이터 위기   370s   에이터 위기   370s   에이터 위기   370s   에이터 위기   370s   370s		데이터 쓰기	1	0	write data									37us	
S=1 : 전체 쉬프트 S=0 : 쉬프트 하지 않음 CGRAM : 폰트 제작 RAM S/C=1 : 표시 쉬프트 S/C=0 : 커서 이동 ACG : CGRAM 어드레스 R/L=1 : 오른쪽으로 쉬프트 R/L=0 : 왼쪽으로 쉬프트 ADD : DDRAM 어드레스 DL=1 : 8비트 DL=0 : 4비트 AC : 어드레스 카운터 N=1 : 2라인 N=0 : 1라인 (DDRAM, CGRAM 어드레스)		데이터 왕기	1	1	read data									37us	
S/C=1 : 표시 쉬프트 S/C=0 : 커서 이동 ACG : CGRAM 어드레스 R/L=1 : 오흔쪽으로 쉬프트 R/L=0 : 왼쪽으로 쉬프트 ADD : DDRAM 어드레스 DL=1 : 8비트 DL=0 : 4비트 AC : 어드레스 카운터 N=1 : 2라인 N=0 : 1라인 (DDRAM, CGRAM 어드레스)	I/D=1:0	I드레스 자동증	가		I/D=	0 : 어	드레:	스 자용	동감소				DDRAM : 표시 데이터 RAM		
R/L=1 : 오른쪽으로 쉬프트 R/L=0 : 왼쪽으로 쉬프트 ADD : DDRAM 어드레스 DL=1 : 8비트 DL=0 : 4비트 AC : 어드레스 카운터 N=1 : 2라인 N=0 : 1라인 (DDRAM, CGRAM 어드레스)	S=1 : 전기	해 쉬프트			S=0	: 쉬프	Œ ŧ	티지않	음				CGRAM : 폰트 제작 RAM		
DL=1 : 8비트 DL=0 : 4비트 AC : 어드레스 카운터 N=1 : 2각인 N=0 : 1각인 (DDRAM, CGRAM 어드레스)	S/C=1:.	표시 쉬프트		s/c	=0:5	4M 0	I동					ACG : CGRAM 어드레스			
N=1: 2단인 N=0:1단인 (DDRAM, CGRAM 어드레스)	R/L=1 : 5	모른쪽으로 쉬프		R/L=0 : 왼쪽으로 쉬프트								ADD : DDRAM 어드레스			
	DL=1:8	비트		DL=0:48  <b>⊆</b>								AC : 어드레스 카운터			
E 0. E 0. 4.	N=1:22	인			N=0 : 1라인								(DDRAM, CGRAM 어드레스	:)	
F=1: 5x10 dots F=0: 5x8 dot	F=1 : 5x1	0 dots			F=0:5x8 dot										
BF=1 : 내부 동작중 BF=0 : 명령/데이터 받기 가능	BF=1 : LH	부 동작중			BF=0 : 명령/데이터 받기 가능										





```
void setup()
 //포트 방향 설정
 pinMode(11, OUTPUT);
 pinMode(12, OUTPUT);
 pinMode(13, OUTPUT);
 //명령 1개 전달
 digitalWrite(11, LOW); // RS = 0, 명령
 digitalWrite(12, LOW) ; // RW = 0, 쓰기
 digitalWrite(13, HIGH); // E = 1
 PORTD = 0x38; // 데이터 출력
 delayMicroseconds(1);
 digitalWrite(13, LOW); // 데이터 쓰기 동작 끝
 delayMicroseconds(1);
void loop()
```

• LCD 초기화 코드 작성



 $0x38 = 0011 \ 1000$ 

```
#define RS
                    11
#define RW
                    12
#define EN
void setup()
 //포트 방향 설정
 pinMode(RS, OUTPUT);
 pinMode(RW, OUTPUT);
 pinMode(EN, OUTPUT) ;
 //명령 1개 전달
 digitalWrite(RS, LOW); // RS = 0, 명령
 digitalWrite(RW, LOW) ; // RW = 0, 쓰기
 digitalWrite(EN, HIGH); // E = 1
 PORTD = 0x38; // 데이터 출력
 delayMicroseconds(1);
 digitalWrite(EN, LOW); // 데이터 쓰기 동작 끝
 delayMicroseconds(1);
void loop()
```

```
Rs

R/W

220ns min

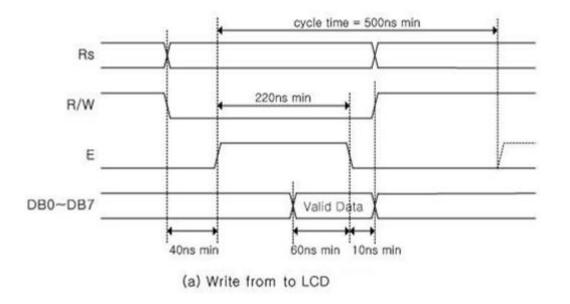
Valid Data

40ns min 10ns min

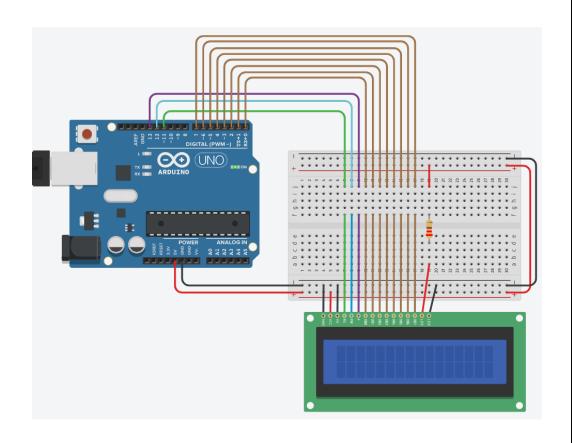
(a) Write from to LCD
```

```
#define RS
              11
#define RW
              12
#define EN
              13
void LCD_Command_Write(char cmd)
 delayMicroseconds(100000);
                                    //100msec
 digitalWrite(RS, LOW); //8 - RS
 digitalWrite(RW, LOW); //9 - RW
 digitalWrite(EN, HIGH); //10 - Enable
 PORTD = cmd;
 delayMicroseconds(1);
 digitalWrite(EN, LOW); //10 - Enable
 delayMicroseconds(1);
void setup()
 //포트 방향 설정
 pinMode(RS, OUTPUT);
 pinMode(RW, OUTPUT);
 pinMode(EN, OUTPUT);
 //명령 1개 전달
LCD Command Write(0x38);
void loop()
```

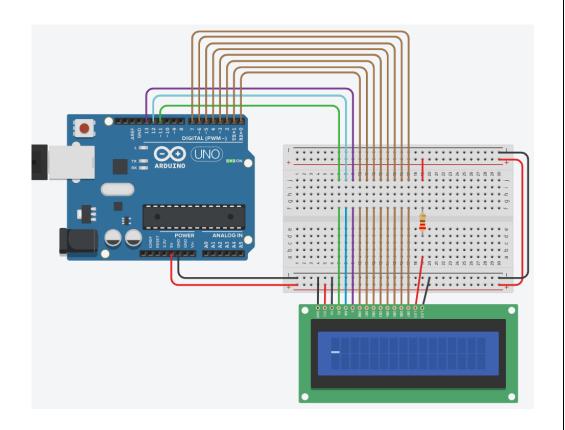




```
void LCD Command Write(char cmd)
void setup()
 //LCD초기화
 delay(150);
 LCD Command_Write(0x38); //0x38 = 0011 1000
 delayMicroseconds(4100);
 LCD Command Write(0x38);
 delayMicroseconds(100);
 LCD Command Write(0x38);
 delayMicroseconds(100);
 LCD Command Write(0x38);
 LCD Command Write(0x0E);
 LCD Command Write(0x01);
 LCD Command Write(0x04);
void loop()
```



```
#define RS
             11
#define RW
              12
             13
#define EN
void LCD_Command_Write(char cmd)
void setup()
 //포트 방향 설정
 pinMode(RS, OUTPUT);
 pinMode(RW, OUTPUT);
 pinMode(EN, OUTPUT);
 //LCD초기화
 delayMicroseconds(150000);
 LCD_Command_Write(0x38); //0x38 = 0011 1000
 delayMicroseconds(4100);
 LCD Command Write(0x38);
 delayMicroseconds(100);
 LCD Command Write(0x38);
 delayMicroseconds(100);
 LCD Command Write(0x38);
 LCD Command Write(0x08);
 LCD Command Write(0x01);
 LCD_Command_Write(0x04);
```



```
#define RS
              11
#define RW
              12
#define EN
              13
void LCD_Command_Write(char cmd)
void setup()
 //포트 방향 설정
 pinMode(RS, OUTPUT);
 pinMode(RW, OUTPUT);
 pinMode(EN, OUTPUT);
 //LCD초기화
 delayMicroseconds(150000);
 LCD_Command_Write(0x38); //0x38 = 0011 1000
 delayMicroseconds(4100);
 LCD Command Write(0x38);
 delayMicroseconds(100);
 LCD Command Write(0x38);
 delayMicroseconds(100);
 LCD Command Write(0x38);
 LCD_Command_Write(0x0E);
                                   //Display On, 커서표시
 LCD Command Write(0x01);
 LCD_Command_Write(0x04);
```

• LCD 문자표시 코드 작성

명령	명령						GIOLE	1				설명	실행
8.8	88	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DBO	20	시간
	화면지우기	0	0	0	0	0	0	0	0	0	1	화면지우기, 커서홈	1,52ms
	커서홈	0	0	0	0	0	0	0	0	1	-	커서 처음 위치로 미동	1,52ms
	엔트리 모드 셋	0	0	0	0	0	0	0	1	I/D	s	어드레스자동증가/감소(I/D) 표시 쉬프트(S)	37us
	표시 On/Off 제어	0	0	0	0	0	0	1	D	С	В	디스플레이(D), 커서(C), 깜박임(B) On/Off	37us
명령 쓰기	표시, 커서 쉬프트	0	0	0	0	0	1	s/c	R/L	-	-	표시, 커서 이동	37us
	기능 셋	0	0	0	0	1	DL	N	F	-	-	인터페이스라인(DL), 라인수(N), 문자폰트(F)	37us
	CGRAM	0	0	0	1 CGRAM 어드레스(ACG)					ACG)		CGRAM 어드레스 설정	37us
	DDRAM 어드레스	0	0	1	D (DDA)스바리에 MARDD							DDRAM 어드레스 설정	37us
명령	नामभाग	0	-	DE		OLC.	OII A	2100	1/ 40			비치표계고 있기	0
윍기	어드레스	0	1	BF		어느	ଘ≏	카운	:1(AC	,		어드레스 카운터 읽기	Ous

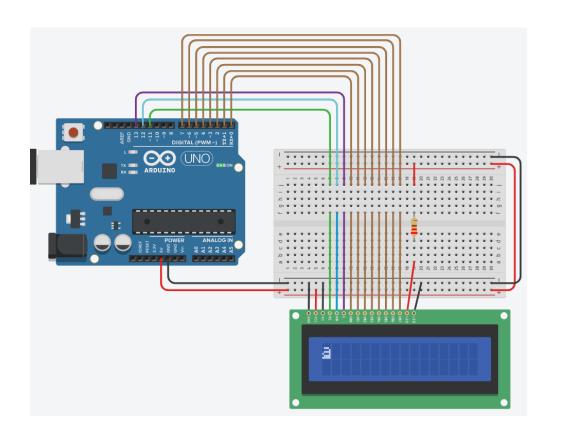
#### D.D.RAM(Display Data RAM)

- 80x8비트 용량으로 80개의 8비트 아스키(ASCII)코드를 저장할 수 있다.
- 0x00~0F 주소가 LCD의 1행의 1~16째.
- 0x40~4f 주소가 LCD의 2행의 1~16번째 문자료 표시 된다.
- 빈 주소에는 자유롭게 RAM 데이터 메모리로 사용이 가능하다.

```
DL=1 : 8비트 DL=0 : 4비트 AC : 어드레스 카운터
N=1 : 2간인 N=0 : 1간인 (DDRAM, CGRAM 어드레스)
F=1 : 5x10 dots F=0 : 5x8 dot
BF=1 : 내부 동작중 BF=0 : 명령/데이터 받기 가능
```

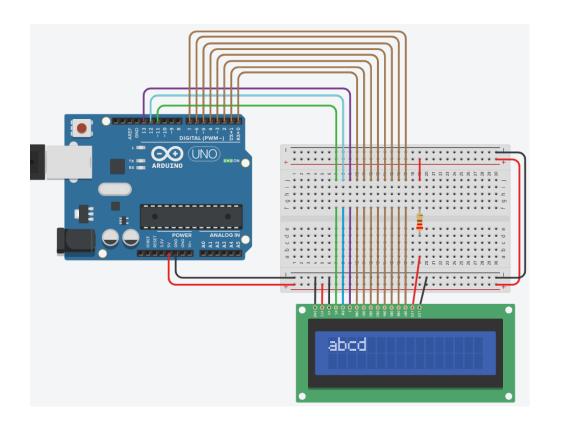
```
void LCD Command Write(char cmd)
void setup()
//포트 방향 설정
 //LCD초기화
void loop()
LCD_Command_Write(0x80 | 0x00); // DDRAM Address = 0 설정
 digitalWrite(RS, HIGH); // 0번 비트 설정, RS = 1, 데이터
 digitalWrite(RW, LOW); // 1번 비트 클리어, RW = 0, 쓰기
 digitalWrite(EN, HIGH); // 2번 비트 설정, E = 1
 PORTD = 'a'; // 데이터 출력
 delayMicroseconds(1);
 digitalWrite(EN, LOW); // 데이터 쓰기 동작 끝
 delayMicroseconds(1);
```

• LCD 문자표시 코드 작성



```
void LCD Command_Write(char cmd) {
void LCD_Data_Write(char data)
 digitalWrite(RS, HIGH); // 0번 비트 설정, RS = 1, 데이터
digitalWrite(RW, LOW) ; // 1번 비트 클리어, RW = 0, 쓰기
 digitalWrite(EN, HIGH); // 2번 비트 설정, E = 1
 PORTD = data;
                    // 데이터 출력
 delayMicroseconds(1);
 digitalWrite(EN, LOW); // 데이터 쓰기 동작 끝
 delayMicroseconds(1);
void setup() {
 //포트 방향 설정
 //LCD초기화
void loop()
 LCD_Command_Write(0x80 | 0x00); // DDRAM Address = 0 설정
 LCD_Data_Write('a');
```

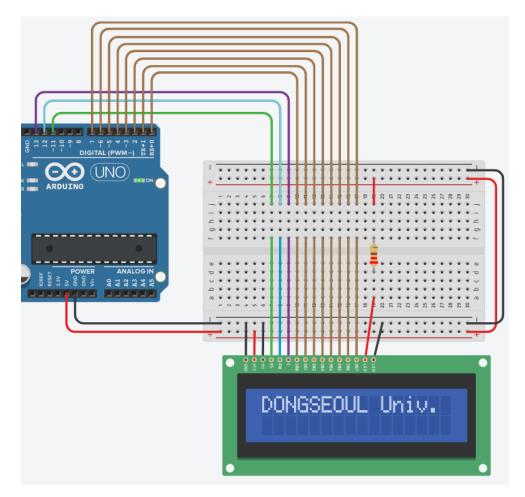
• LCD 문자표시 코드 작성



```
void LCD Command_Write(char cmd) {
void LCD_Data_Write(char data) {
void setup() {
 //포트 방향 설정
 //LCD초기화
void loop()
 LCD_Command_Write(0x80 | 0x00); // DDRAM Address = 0 설정
 LCD_Data_Write('a');
 LCD_Command_Write(0x80 | 0x01); // DDRAM Address = 1 설정
 LCD_Data_Write('b');
 LCD_Command_Write(0x80 | 0x02); // DDRAM Address = 2 설정
 LCD_Data_Write('c');
 LCD_Command_Write(0x80 | 0x03); // DDRAM Address = 3 설정
 LCD_Data_Write('d');
```

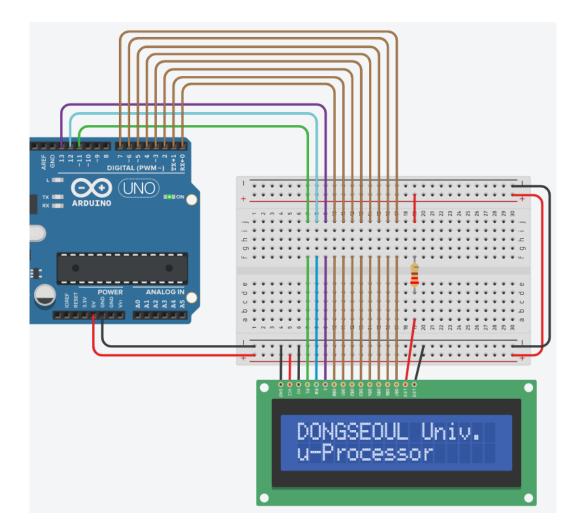
## 16x2 Character LCD 실험 - quiz1

• Example\_32를 참고하여 아래의 그림과 같이 LCD에 문자를 출력 하시오

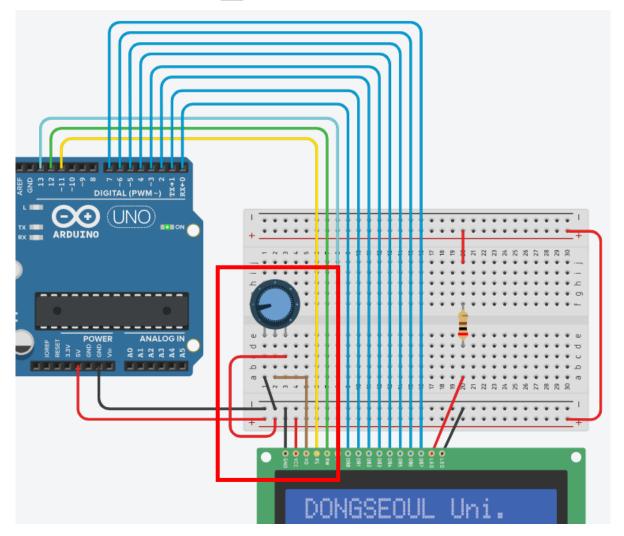


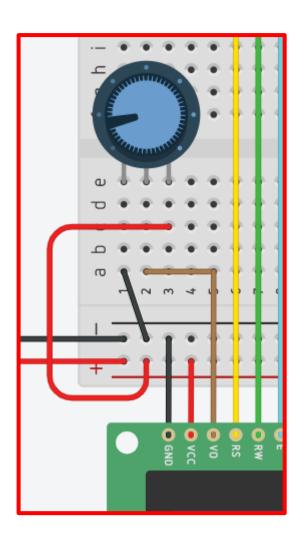
## 16x2 Character LCD 실험 - quiz2

• Example\_32를 참고하여 아래의 그림과 같이 LCD에 문자를 출력 하시오

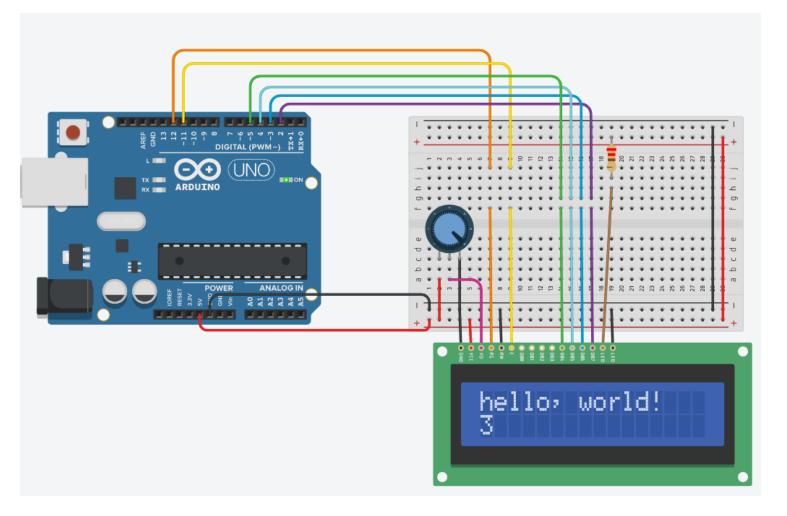


• Contrast 조절





• 아두이노 라이브러리 사용한 LCD실험



```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
 lcd.begin(16, 2);
 lcd.print("hello, world!");
void loop() {
 lcd.setCursor(0, 1);
 lcd.print(millis() / 1000);
```

• 아두이노 라이브러리 사용한 LCD실험, Backlight 포트 제어

