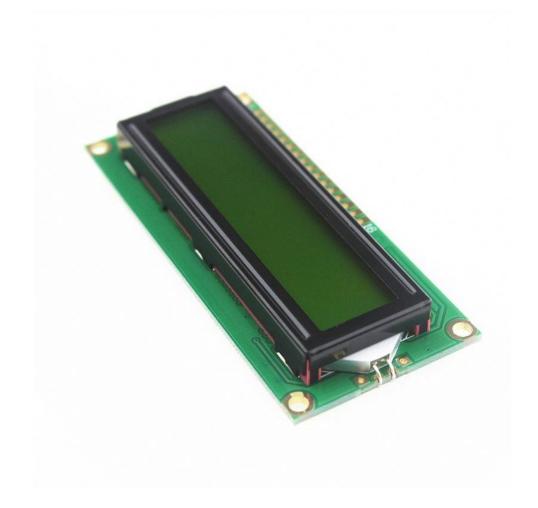
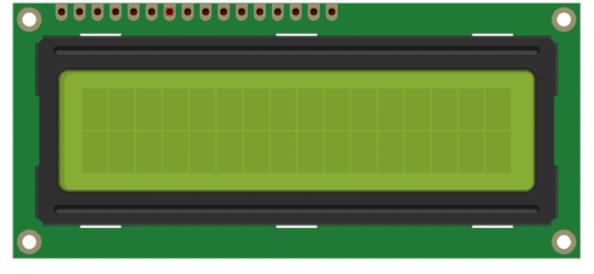
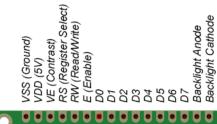
아두이노 LCD 실험 & 시뮬레이터 실험

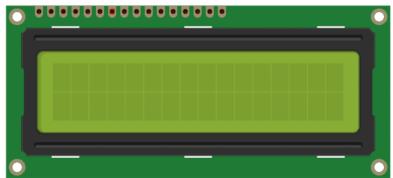
"수업자료는 여기"



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

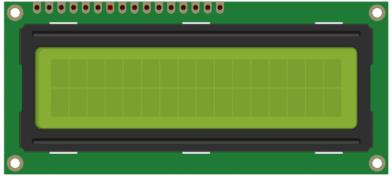






- VSS(Ground): LCD에 전원을 인가하는 단자로, 0V(GND)에 연결.
- VDD(5V): LCD에 전원을 인가하는 단자로, +5V에 연결.
- **VE(Contrast)** : LCD의 밝기를 조절하는 단자로서, 10kΩ의 가변 저항을 연결하여 밝기를 조정. 밝기 조정을 하지 않으려면 GND에 연결.
- RS(Register Select) : 레지스터 종류를 선택
 - 0: 명령 레지스터
 - 1: 데이터 레지스터
- RW(Read/Write) : 데이터 혹은 명령을 읽는지 쓰는지 설정
 - 0 : 레지스터의 데이터를 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 문자료 표시 된다.
- 빈 주소에는 자유롭게 RAM 데이터 메모리로 사용이 가능하다.

C.G.RAM(Character Generator RAM)

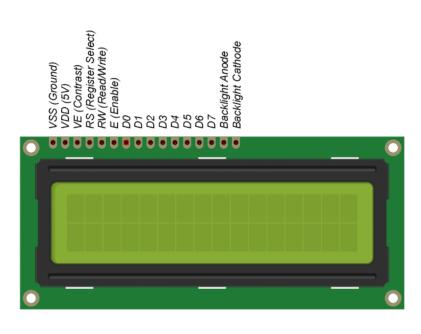
- 사용자가 문자를 만들 때 사용하는 메모리로 5x7은 8개, 5x10은 4개를 만들어서 저장이 가능.
- C.G.ROM((Charactor Generator ROM) : 5x7, 5x10의 도트문자를 내장하고 있음. 아스키코드와 일치

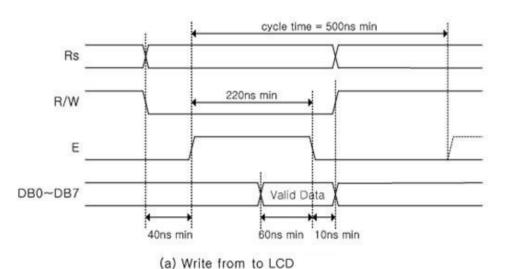
Address Counter (AC)

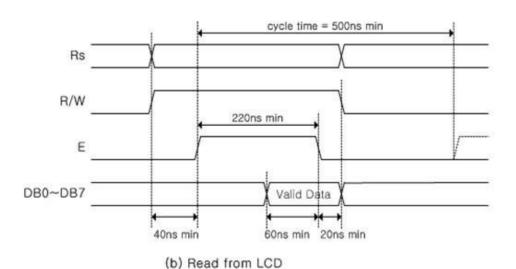
• 주소를 저장하는 레지스터로 D.D.RAM과 C.G.RAM의 주소를 지정할 때 사용, 명령레지스터에 주소정보를 지정하면 AC로 주소정보가 전달 되고, D.D.RAM에 데이터를 쓰면 AC는 자동으로 (설정 값에 따라) 1증가하거나 1감소한다.

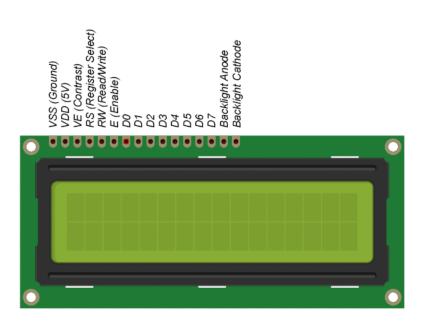
Busy Flag (BF)

- 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









명령	ප ලි						GIOLE	3				설명	실행
98	08	RS	R/W	DB7	DB6	DB5	DB4	DВЗ	DB2	DB1	DBO	80	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	Ø	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 어드레스	0	0	0	1 CGRAM 어드레스(ACG)							CGRAM 머드레스 설정	37us
	DDRAM 어드레스	0	0	1		DDRA	чм ОН	드레=	∆(AD	D)		DDRAM 어드레스 설정	37us
명령 읽기	비지체크, 어드레스	0	1	BF		ОЧ⊆	레스	카운6	H(AC)		비지플래그 읽기 어드레스 카운터 읽기	0us
데이터 쓰기	데이터 쓰기	1	0			WI	rite d	ata				CGRAM 또는 DDRAM에 데이터 쓰기	37us
데이터 읽기	데이터 읽기	1	1			ге	ad d	ata				CGRAM 또는 DDRAM에서 데이터 읽기	37us
I/D=1:0	1드레스 자동증	가		I/D=	o : O	드레:	스 자용	동감소	:			DDRAM : 표시 데이터 RAM	
C_1 · 전체 시교론 C_0 · 시교론 원자 아이											CCDAM · 포트 제자 DAM		

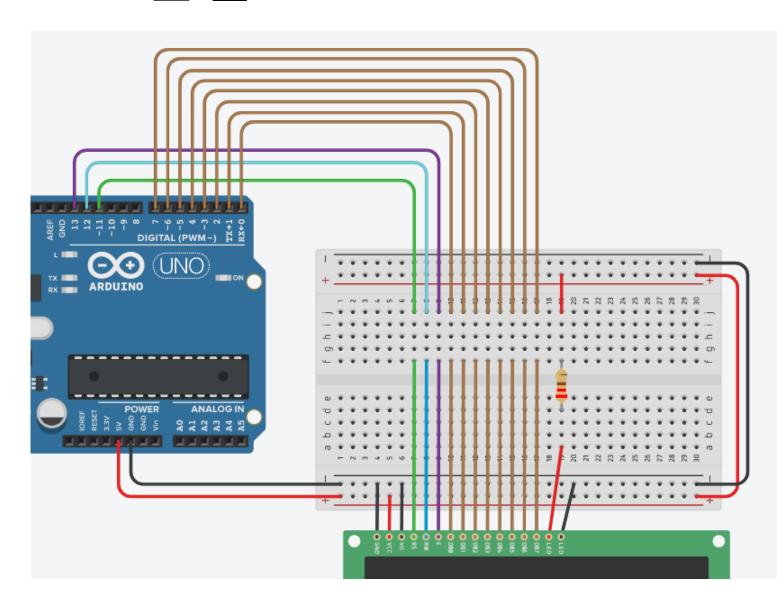
• 회로 구성

• RS: 11번핀(PORTB3)

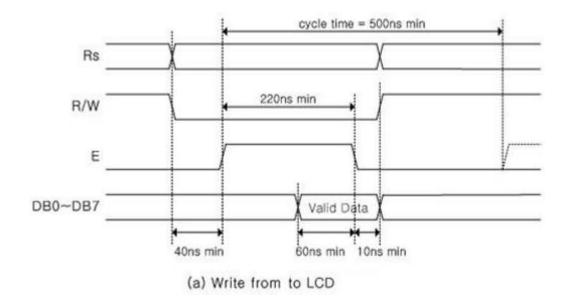
• RW: 12번핀(PORTB4)

• E: 13번핀(PORTB5)

• D0~D7: 0번핀~7번핀)PORTD



• LCD 초기화(명령 Write to LCD)



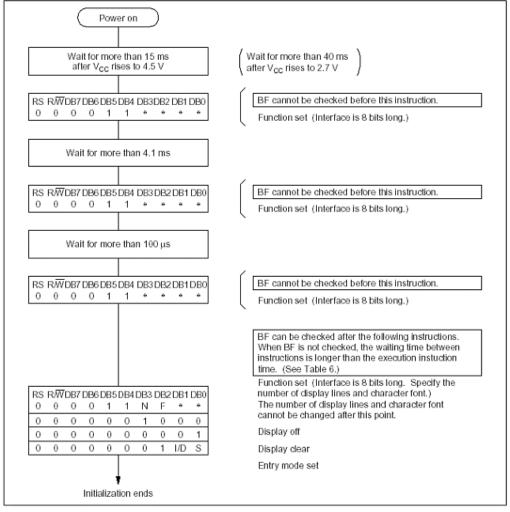
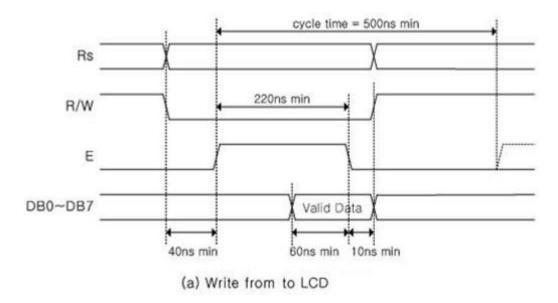


Figure 23 8-Bit Interface

• LCD 초기화

RS(Register Select): 0 → 명령 레지스터



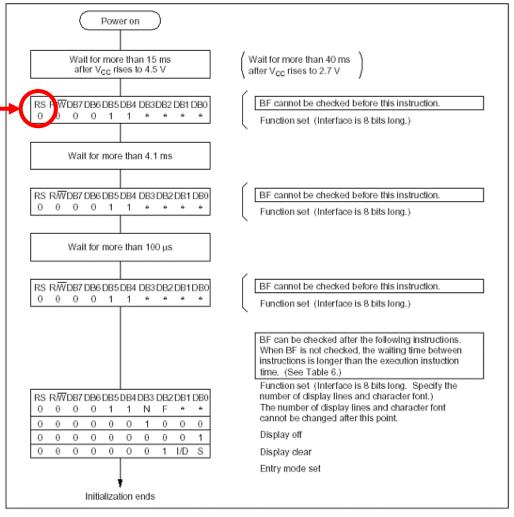


Figure 23 8-Bit Interface

• 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

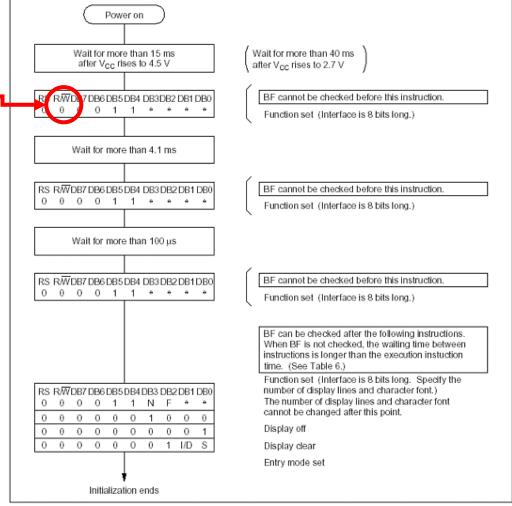
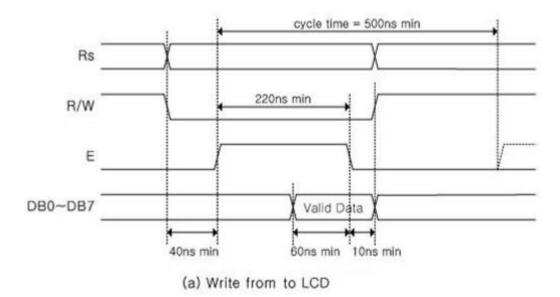


Figure 23 8-Bit Interface

• LCD 초기화

DB5 bit가 1이면 "기능셋 " 설정



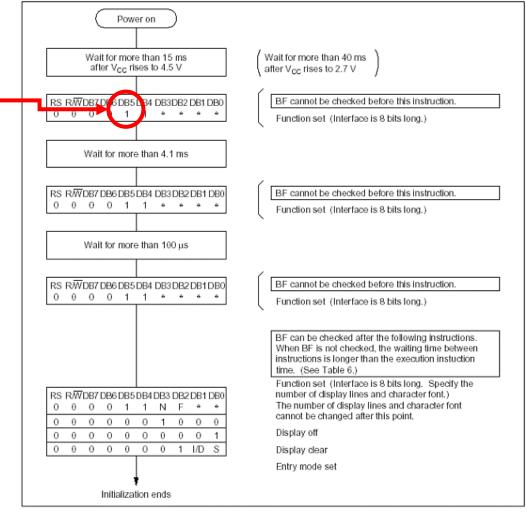
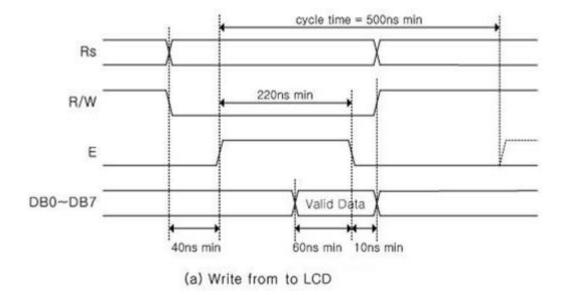


Figure 23 8-Bit Interface

• LCD 초기화

DL(DB4 bit) : Data Line : 1 → 8Bit



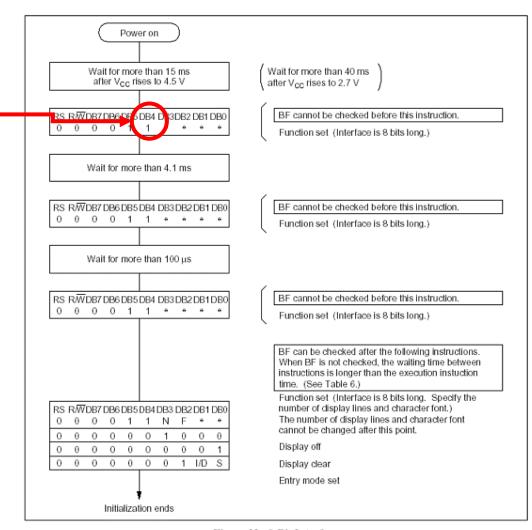
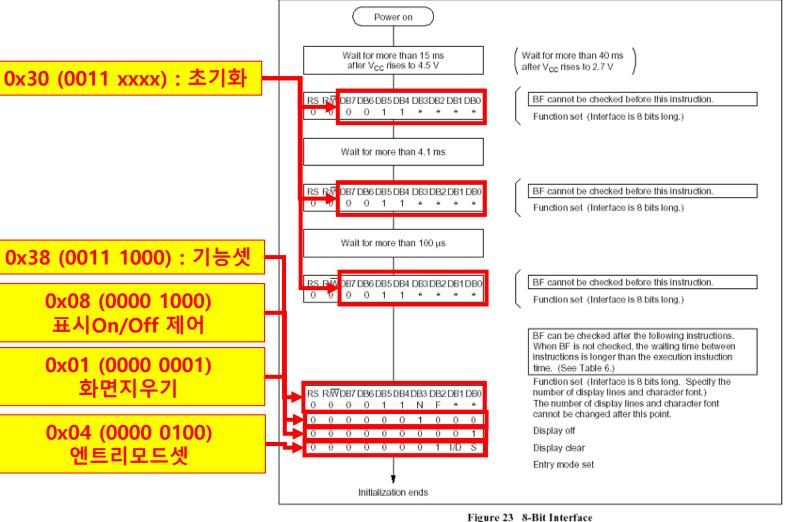
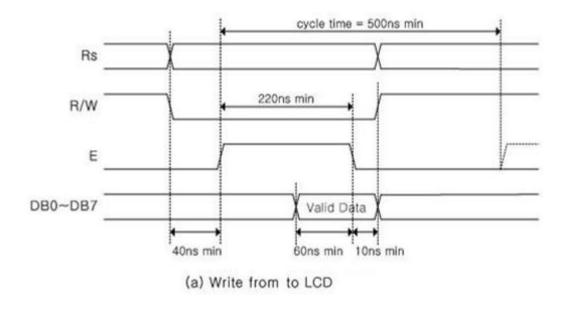


Figure 23 8-Bit Interface

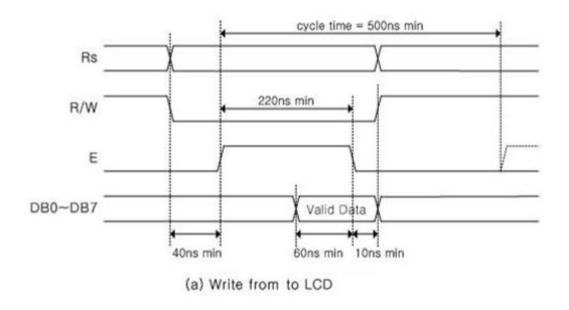
• LCD 초기화

함께 전체	명령	명령						GIOLE	4				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					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	CG	RAM	어드	레스(.	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 어드레스 ACG: CGRAM 어드레스 ACG: CGRAM 어드레스 ACG: CGRAM 어드레스 ACG: OHEAM 어드레스 ACG: OHEAM 어드레스 카운터 N=1: 2만인 (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			WI	rite d	ata					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			ге	ad d	ata					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 : 온	!쪽으	로 쉬.	≖∈				ADD : DDRAM 어드레스		
	DL=1:8	비트			DL=I	D:46	IE						AC : 어드레스 카운터		
E 0. E 0. 4.	N=1:22	인			N=0	: 1라	인						(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=I	0:명	령/데	이터	받기:	가능					



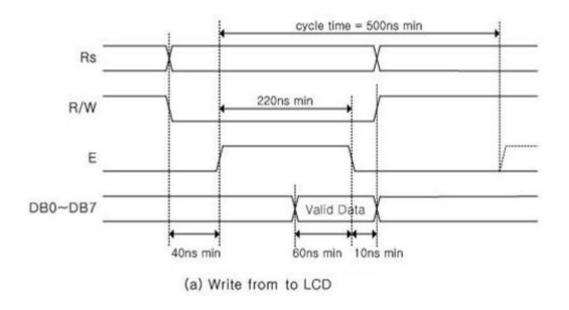


```
void setup()
 //포트 방향 설정
 pinMode(11, OUTPUT);
 pinMode(12, OUTPUT);
 pinMode(13, OUTPUT);
 //명령 1개 전달
 digitalWrite(11, LOW); // RS = 0, 명령
 digitalWrite(12, LOW) ; // RW = 0, 쓰기
 digitalWrite(13, HIGH); // E = 1
 PORTD = 0x38; // 데이터 출력
 delayMicroseconds(1);
 digitalWrite(13, LOW); // 데이터 쓰기 동작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

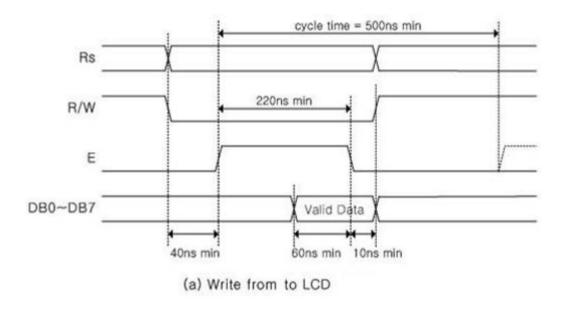


```
#define RS
                    11
#define RW
                    12
                    13
#define EN
void setup()
 //포트 방향 설정
 pinMode(RS, OUTPUT) ;
 pinMode(RW, OUTPUT);
 pinMode(EN, OUTPUT);
 //명령 1개 전달
 digitalWrite(RS, LOW); // RS = 0, 명령
 digitalWrite(RW, LOW) ; // RW = 0, 쓰기
 digitalWrite(EN,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

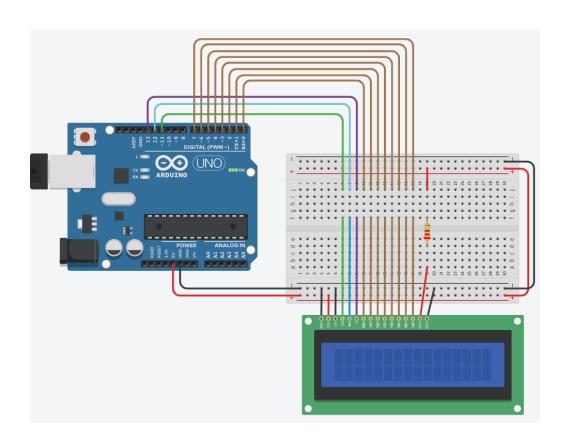
#define RS



```
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()
```

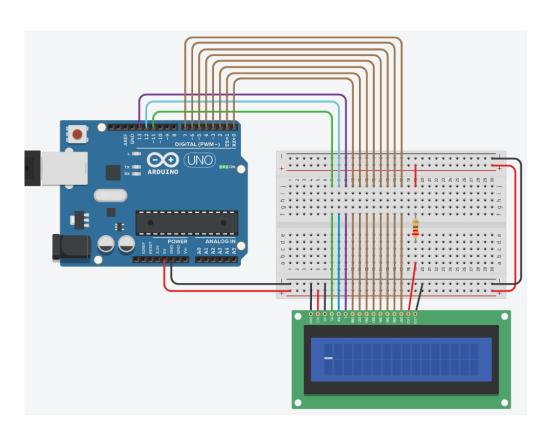


```
#define RS 11
#define RW 12
#define EN 13

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(0x08);
 LCD Command Write(0x01);
 LCD_Command_Write(0x04);
```

• LCD 초기화 코드 작성



```
#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	:				설명	실행	
88	9.8	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DBO	28	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	1
	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	CG	BAM	어드	레스(ACG))	CGRAM 머드레스 설정	37us	
	DDRAM 어드레스	0	0	1		DDRAM 어드레스(ADD)						DDRAM 어드레스 설정	37us	
명령 읽기	비판체크 어드레스	0	1	BF	어드레스 카운터(AC)							비자품에고 있다. 어드레스 카운터 읽기	Ous	

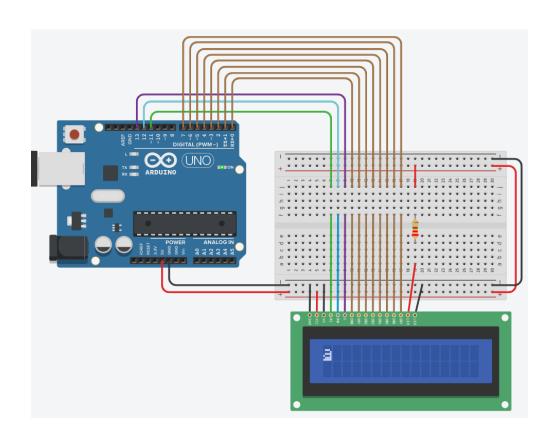
D.D.RAM(Display Data RAM)

- 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
```

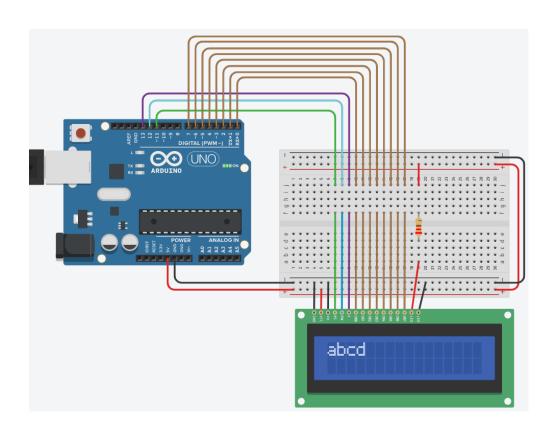
```
void LCD Command Write(char cmd)
void setup()
 //포트 방향 설정
 //LCD초기화
void loop()
 LCD_Command_Write(0x80 | 0x00); // DDRAM Address = 0 설정
 digitalWrite(RS, HIGH); // 0번 비트 설정, RS = 1, 데이터
 digitalWrite(RW, LOW); // 1번 비트 클리어, RW = 0, 쓰기
 digitalWrite(EN, HIGH); // 2번 비트 설정, E = 1
 PORTD = 'a'; //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

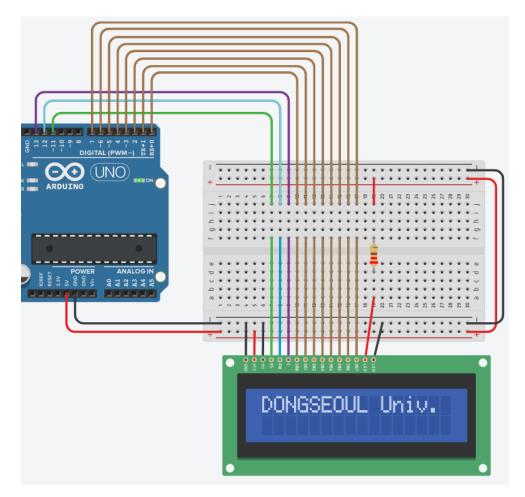
• LCD 문자표시 코드 작성



```
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');
```

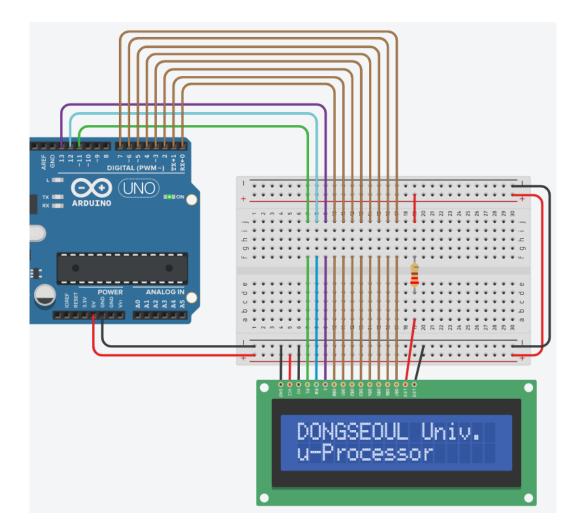
16x2 Character LCD 실험 - quiz1

• Example_32를 참고하여 아래의 그림과 같이 LCD에 문자를 출력 하시오

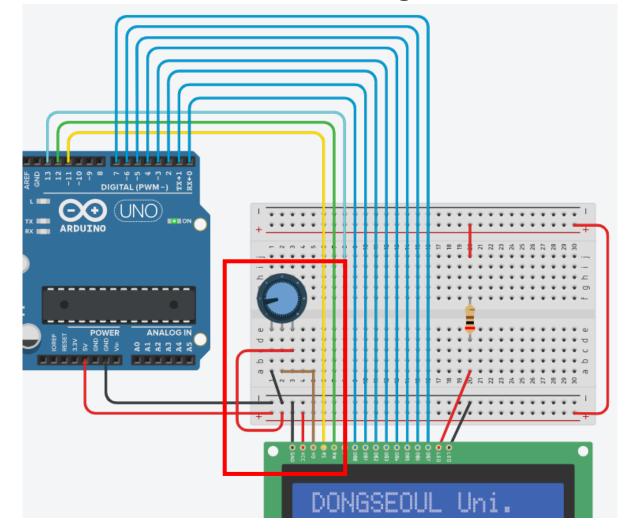


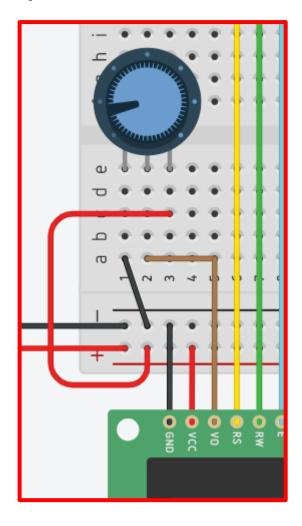
16x2 Character LCD 실험 - quiz2

• Example_32를 참고하여 아래의 그림과 같이 LCD에 문자를 출력 하시오

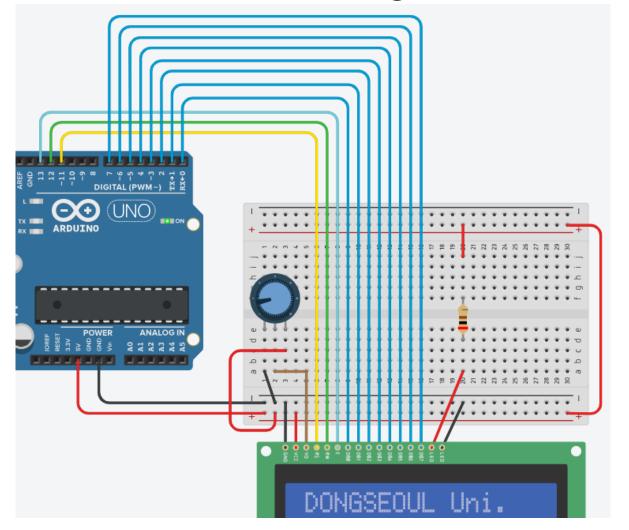


• Contrast 조절 및 String 예제 실험(example_36)



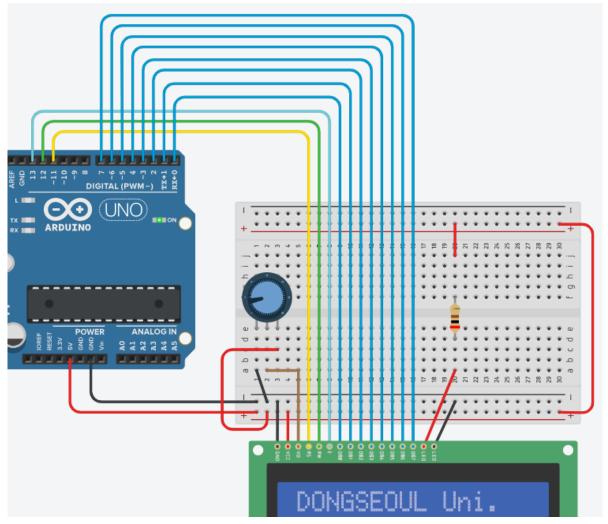


• Contrast 조절 및 String 예제 실험(example_36_2)

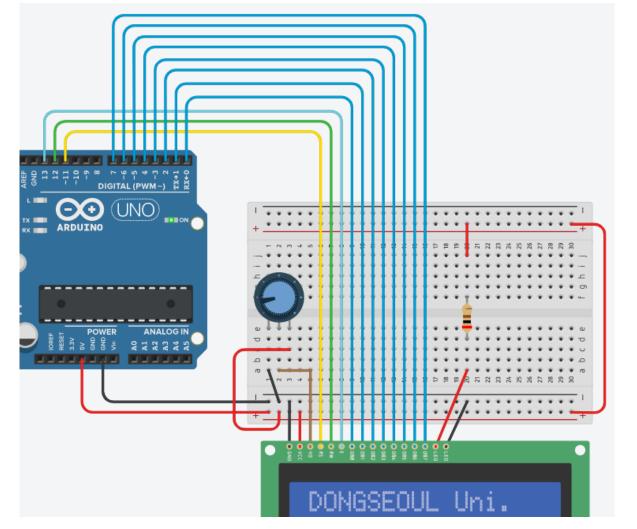


DI SI	DI SI						GIOLE	3				мп	실행
명령	명령	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DBO	설명	시간
	화면지우기	0	0	0	0	0	0	0	0	0	1	화면지우기, 커서홈	1,52ms
	크개주	0	0	0		0	0	0	0	1		과서 되는 이태를 미토	1 50ma
	엔트리 모드 셋	0	0	0	0	0	0	0	1	I/D	s	어드레스자동증가/감소(I/D) 표시 쉬프트(S)	37us
		0	0	0	0	0	0	1	D	С	В		37us
	제어											깜박임(B) On/Off	
명령 쓰기	표시, 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 어드레스	0	0	0	1	CG	RAM	어드	레스(.	ACG)		CGRAM 어드레스 설정	37us
	DDRAM 어드레스	0	0	1	1	DDRA	чм ОН	드레스	∆(AD	D)		DDRAM 어드레스 설정	37us
명령 읽기	비지체크, 어드레스	0	1	BF 어드레스 카운터(AC)								비지플래그 읽기 어드레스 카운터 읽기	0us
데이터 쓰기	데이터 쓰기	1	0			W	rite d	ata				CGRAM 또는 DDRAM에 데이터 쓰기	37us
데이터 윍기	데이터 읽기	1	1			ге	ad d	ata				CGRAM 또는 DDRAM에서 데이터 윍기	37us
I/D=1:0	서드레스 자동증	가		I/D=	0 : 어	드레:	스 자동	동감소			DDRAM : 표시 데이터 RAM		
S=1 : 전#	헤 쉬프트			S=0	: 쉬프	Œ ē	IXI &	음				CGRAM : 폰트 제작 RAM	
S/C=1:	표시 쉬프트			s/c	=0:5	444 O	l 동					ACG : CGRAM 어드레스	
R/L=1 : 5	모른쪽으로 쉬프		R/L=	=0 : 온	!쪽으	로 쉬	ΞΞ				ADD : DDRAM 어드레스		
DL=1:8	비트		DL=I	0 : 4b	IE						AC : 어드레스 카운터		
N=1 : 2김	P인			N=0	: 1라	인						(DDRAM, CGRAM 어드레스	<u>≥</u>)
F=1 : 5x1	IO dots			F=0	: 5x8	dot							
BF=1 : 나	부 동작중			BF=I	0 : 명	령/데	이터	받기:	가능				

• String Shift 예제 실험(example_37)

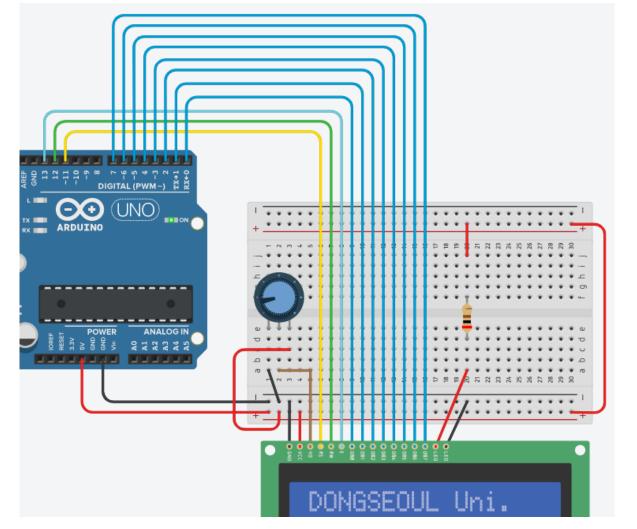


• String Shift 예제 실험(example_37_2)



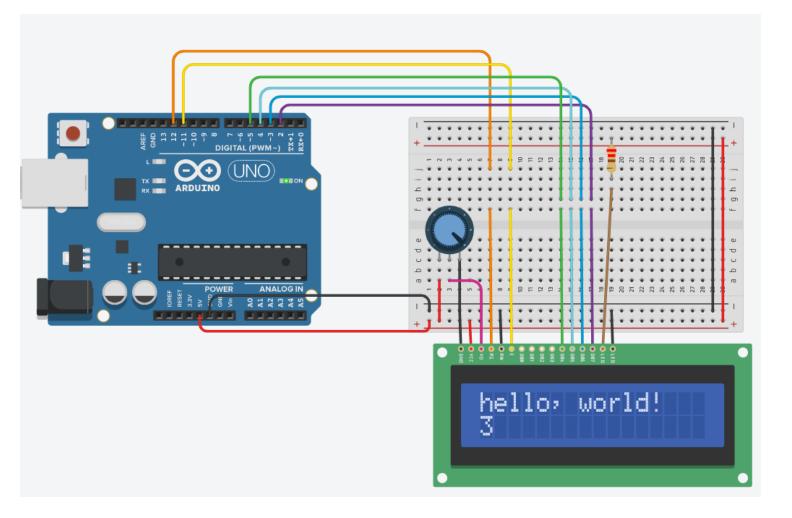
	T _							GIOLE	3					실행	
명령	P	병령	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DBO	설명	시간	
	화면	지우기	0	0	0	0	0	0	0	0	0	1	화면지우기, 커서홈	1,52ms	
•	∌.	서홈	0	0	0	0	0	0	0	0	1	-	커서 처음 위치로 이동	1,52ms	
	-	트리 드 셋	0	0	0	0	0	0	0	1	I/D	s	어드레스자동증가/감소(I/D) 표시 쉬프트(S)	37us	
		On/Off IIO	0	0	0	0 0 0 1 D C B					С	В	디스플레이(D), 커서(C), 깜박임(B) On/Off	37us	
명령 쓰기		, 커서 프트	0	0	0	0 0 1 S/C R/L				R/L	-	-	표시, 커서 이동	37us	
	기	능 셋	0	0	0	0	1 DL N F						인터페이스라인(DL), 라인수(N), 문자폰트(F)	37us	
	CGRAM 어드레스 0 0					1	CG	RAM	Η⊆	레스(.	ACG)	•	CGRAM 어드레스 설정	37us	
		BAM E레스	0	0	1 DDRAM 에드레스(ADD)								DDRAM 어드레스 설정	37us	
명령 읽기		체크, 트레스	0	1	BF		어드	레스	카운	H(AC)		비지플래그 읽기 어드레스 카운터 읽기	0us	
데이6 쓰기	ં લાગા	러 쓰기	1	0			W	rite d	ata				CGRAM 또는 DDRAM에 데이터 쓰기	37us	
데이6 왕기	· G 0 6	터 읽기	1	1			ге	ad d	ata				CGRAM 또는 DDRAM에서 데이터 읽기	37us	
I/D=1	: 어드레스	: 자동증	가		I/D=	0 : 어	드레:	스 자용	동감소	:			DDRAM : 표시 데이터 RAM		
S=1:	전체 쉬프	⊑			S=0	: 쉬프	Œŧ	티지않	음				CGRAM : 폰트 제작 RAM		
S/C=1	:표시 쉬	프트			s/c	=0:9	HM 0	I동					ACG : CGRAM 어드레스		
R/L=1	: 오른쪽9		R/L=	0 : 온	!쪽으	로쉬:	≖∈				ADD : DDRAM 어드레스				
DL=1	:88 ⊑				DL=0	o: 48	ļΞ						AC : 어드레스 카운터		
N=1:	2라인				N=0	: 1라	인						(DDRAM, CGRAM 어드레스)		
F=1:	5x10 dots	3			F=0	: 5x8	dot								
BF=1	: 내부 동작	악중			BF=0): 명	령/데	이터 1	받기:	가능					

• String Shift 예제 실험(example_37_4)



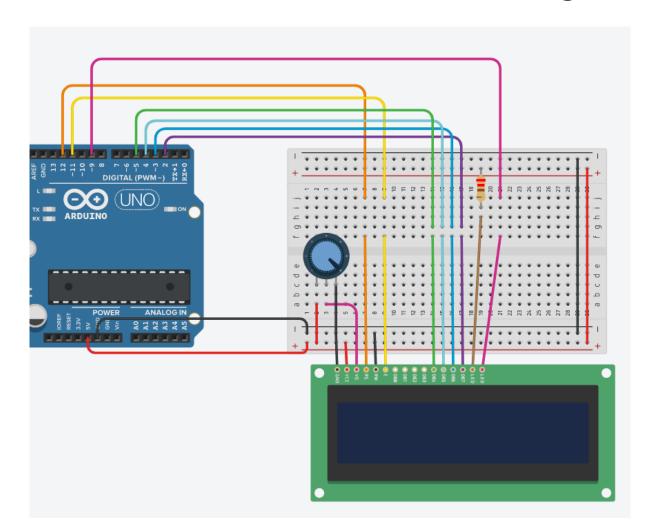
							GIOLE	:					실행
명령	명령	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	설명	시간
	화면지우기	0	0	0	0	0	0	0	0	0	1	화면지우기, 커서홈	1,52ms
	커서홈	0	0	0	0	0 0 0 0 1 -					-	커서 처음 위치로 이동	1,52ms
	엔트리 모드 셋	0	0	0	0	0	0	0	1	I/D	s	어드레스자동증가/감소(I/D) 표시 쉬프트(S)	37us
	표시 On/Off 제어	0	0	0 0 0 0 1 D C B				디스플레이(D), 커서(C), 깜박임(B) On/Off	37us				
명령 쓰기	표시, 커서 쉬프트	0	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 어드레스	0	0	0	1	CG	BAM	Η⊆	레스(ACG)	CGRAM 어드레스 설정	37us	
	DDRAM 어드레스	0	0	1	1 DDRAM 어드레스(ADD)							DDRAM 어드레스 설정	37us
명령 읽기	비지체크, 어드레스	0	1	BF		어드	레스	카운6	H(AC)		비지플래그 읽기 어드레스 카운터 읽기	0us
데이터 쓰기	데이터 쓰기	1	0			W	rite d	ata				CGRAM 또는 DDRAM에 데이터 쓰기	37us
데이터 왕기	데이터 윍기	1	1			ге	ad d	ata				CGRAM 또는 DDRAM에서 데이터 왕기	37us
I/D=1 : 0	I드레스 자동증	가		I/D=	0 : 어	드레:	스 자용	동감소	:		DDRAM : 표시 데이터 RAM		
S=1 : 전호	해 쉬프트			S=0	: 쉬프	Œέ	티지않	음				CGRAM : 폰트 제작 RAM	
S/C=1:3	표시 쉬프트			S/C	=0:5	144 O	용					ACG : CGRAM 어드레스	
R/L=1 : 9	모른쪽으로 쉬프	≝		R/L=	0 : શ	!쪽으	로 쉬.	ΞE				ADD : DDRAM 어드레스	
DL=1 : 8t	비트			DL=0) : 4b	IE						AC : 어드레스 카운터	
N=1 : 2라	인			N=0	: 1감	인						(DDRAM, CGRAM 어드레스	:)
F=1 : 5×1	0 dots			F=0	: 5x8	dot							
BF=1: LH	부 동작중			BF=0):명	령/데	이터	받기:	가능				

• 아두이노 라이브러리 사용한 LCD실험



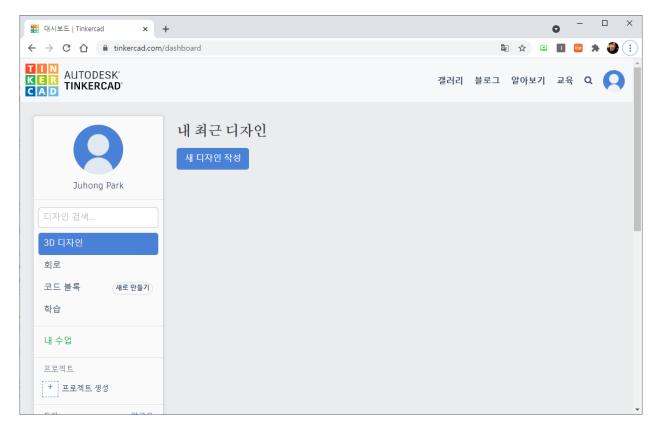
```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
 lcd.begin(16, 2);
 lcd.print("hello, world!");
void loop() {
 lcd.setCursor(0, 1);
 lcd.print(millis() / 1000);
```

• 아두이노 라이브러리 사용한 LCD실험, Backlight 포트 제어

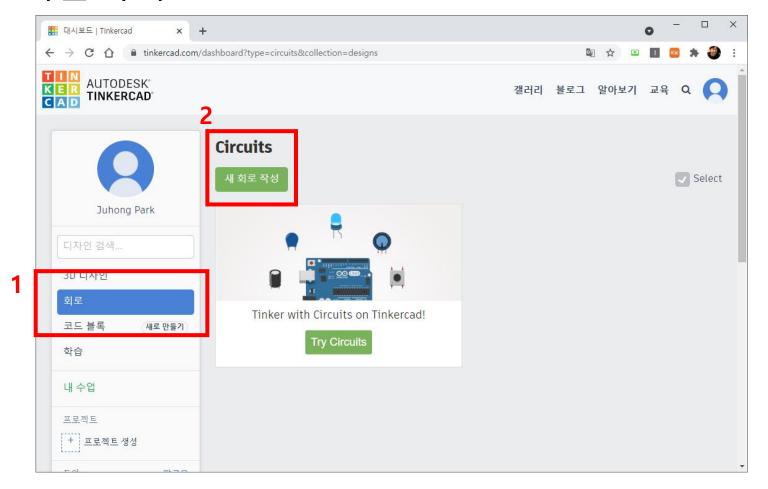


tikercad

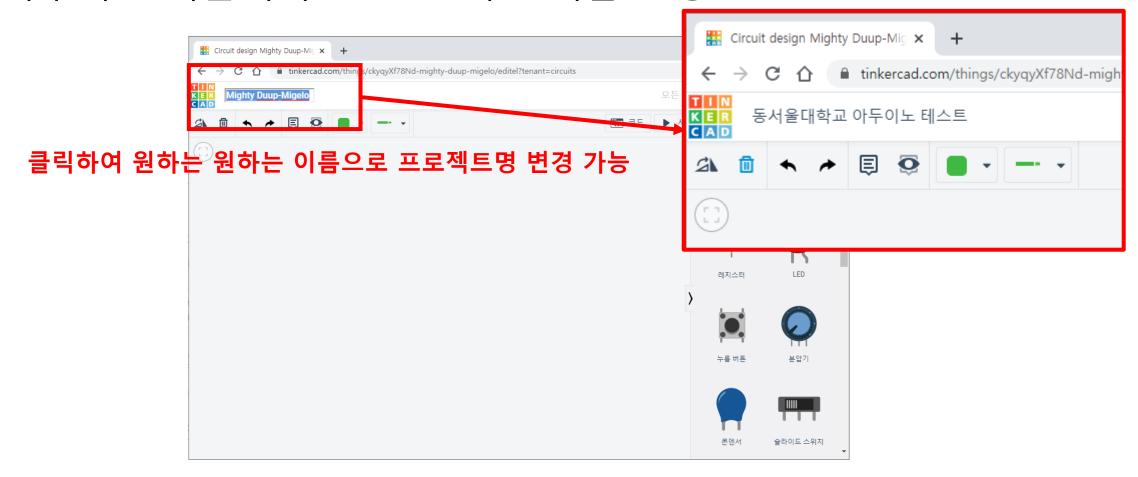
- 아두이노를 시뮬레이션 할 수 있는 사이트
- https://www.tinkercad.com/



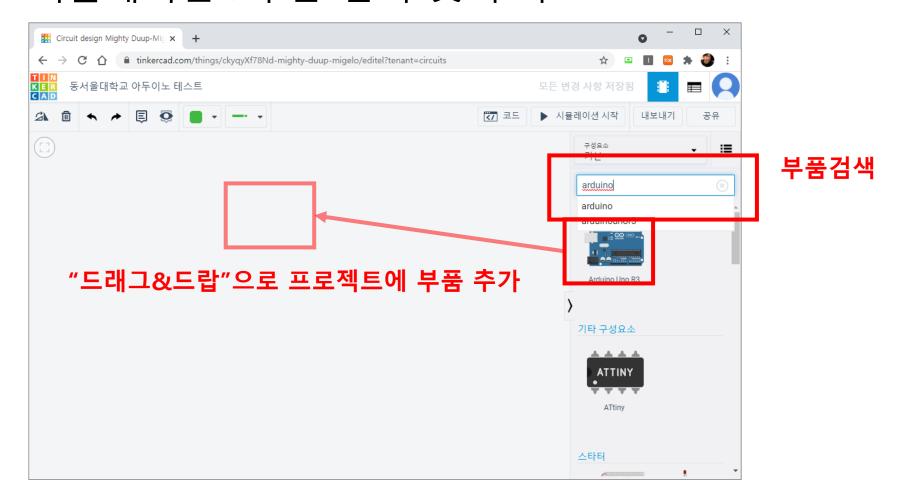
• 아두이노 시뮬레이션



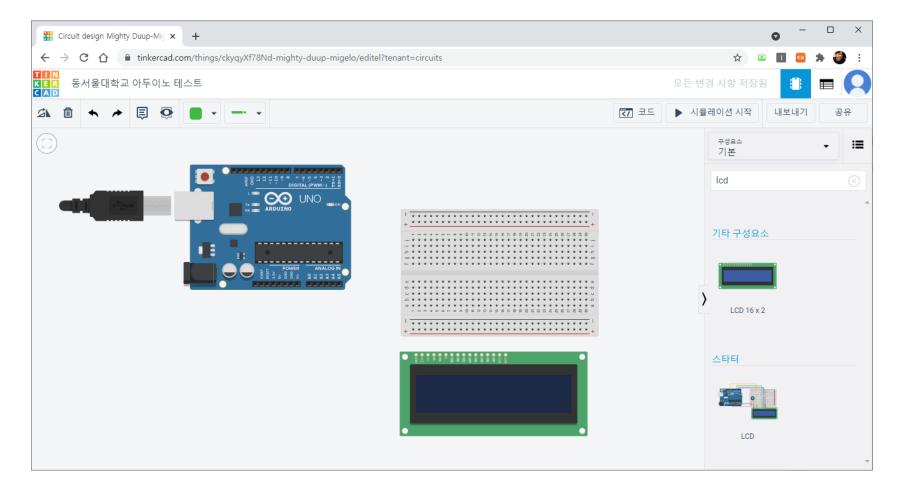
• 아두이노 시뮬레이션 : 프로젝트 이름 변경



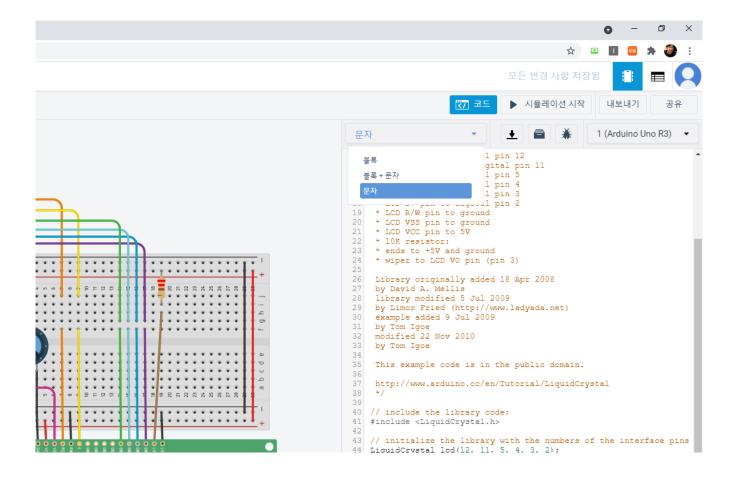
• 아두이노 시뮬레이션 : 부품 검색 및 추가



• 아두이노 시뮬레이션 : 실험에 필요한부품 추가



• 아두이노 시뮬레이션 : 코드 작성 및 시뮬레이션



• 아두이노 시뮬레이션 : 코드 작성 및 시뮬레이션

