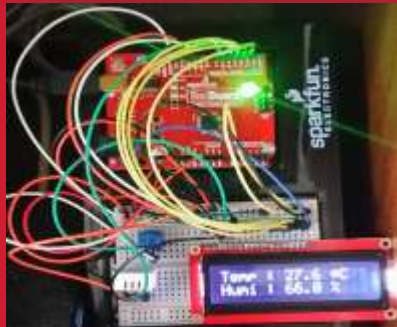




Arduino-basic

[wk03]

LCD & LED



Learn how to code Arduino from scratch

Comsi, INJE University

2nd semester, 2019

Email : chaos21c@gmail.com



My ID (ARnn)

AR01	염현제
AR02	강민수
AR03	구병준
AR04	김종민
AR05	박성철
AR06	이승현
AR07	이창호
AR08	변성현
AR09	손성빈
AR10	안예찬
AR11	유종인
AR12	이석민
AR13	이주원
AR14	정재영
AR15	차요신

AR16	하태성
AR17	강현이
AR18	신종원
AR19	최진솔
AR20	김경미
AR21	김경영
AR22	김규년
AR23	김민재
AR24	김영록
AR25	송다은
AR26	정지환
AR27	김종건



[Review]

◆ [wk02]

- Serial comm.
- Complete your project
- upload folder: Arnn_Rpt01
in github.com repo “arnn”

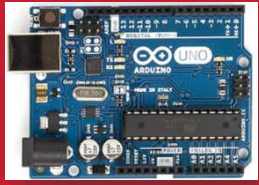
◆ [Target of this week]

- Complete your works
- Save your outcomes and upload 4 figures in github.com

제출폴더명 : **ARnn_Rpt01**

- 제출할 파일들

- ① **ARnn_blink.png**
- ② **ARnn_sawtooth.png**
- ③ **ARnn_loop_escape.png**
- ④ **ARnn_sum100.png**



Hello

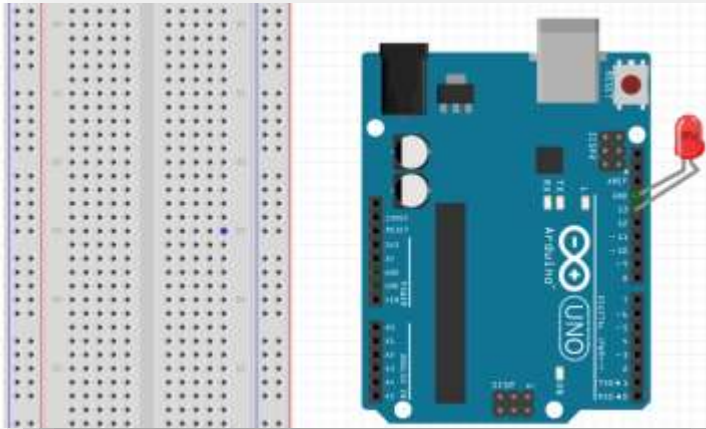


Blink

a LED



Blink a LED!



```
Blink$
1 /*
2  Blink
3  Turns on an LED on for one second, then off for one second, repeatedly.
4  */
5
6
7 // the setup function runs once when you press reset or power the board
8 void setup() {
9   // initialize digital pin 13 as an output.
10  pinMode(13, OUTPUT);
11 }
12
13 // the loop function runs over and over again forever
14 void loop() {
15   digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
16   delay(1000);            // wait for a second
17   digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
18   delay(1000);            // wait for a second
19 }
```



2. Serial comm. monitor & plotter

2. Serial comm.

시리얼 통신

- 2.1 Arduino에서 컴퓨터로 데이터 전송하기
- 2.2 변수 유형별로 컴퓨터에 전송하기
- 2.3 Arduino에서 시리얼 통신을 이용하여
 데이터 수신하기

2.1.3 Arduino에서 컴퓨터로 데이터 전송하기

ex_2_1 | 아두이노 1.8.2

↔

—

□

×

파일 편집 스케치 툴 도움말

✓

→

📄

⬆

⬇

ex_2_1

```

1 /*
2  예제 2.1
3  Arduino에서 컴퓨터로 변수와 문자열 전송하기
4  */
5
6  int number = 0;           // -32768~32767 범위의 변수 number 설정, 초기값은 0
7
8  void setup() {
9    Serial.begin(9600);     // 9600bps로 시리얼 통신 설정
10
11
12  void loop() {
13    Serial.print(number);   // number 변수값 출력
14    Serial.println(" sec"); // " sec"를 출력 후 줄 바꿈
15    delay(1000);           // 1초동안 지연시킨다.
16    number++;              // number 변수값을 하나 증가시킨다.
17  }
18

```

업로드 완료.

스케치는 프로그램 저장 공간 1862 바이트(5%)를 사용, 최대 32256 바이트.

전역 변수는 동적 메모리 192바이트(9%)를 사용, 1856바이트의 지역변수가 남음. 최대는 2048 바이트.

COM3 (Arduino/Genuino Uno)

36 sec

37 sec

38 sec

39 sec

40 sec

41 sec

42 sec

43 sec

44 sec

45 sec

46 sec

47 sec

48 sec

49 sec

50 sec

51 sec

52 sec

53 sec

54 sec

55 sec

56 sec

57 sec

58 sec

59 sec

60 sec

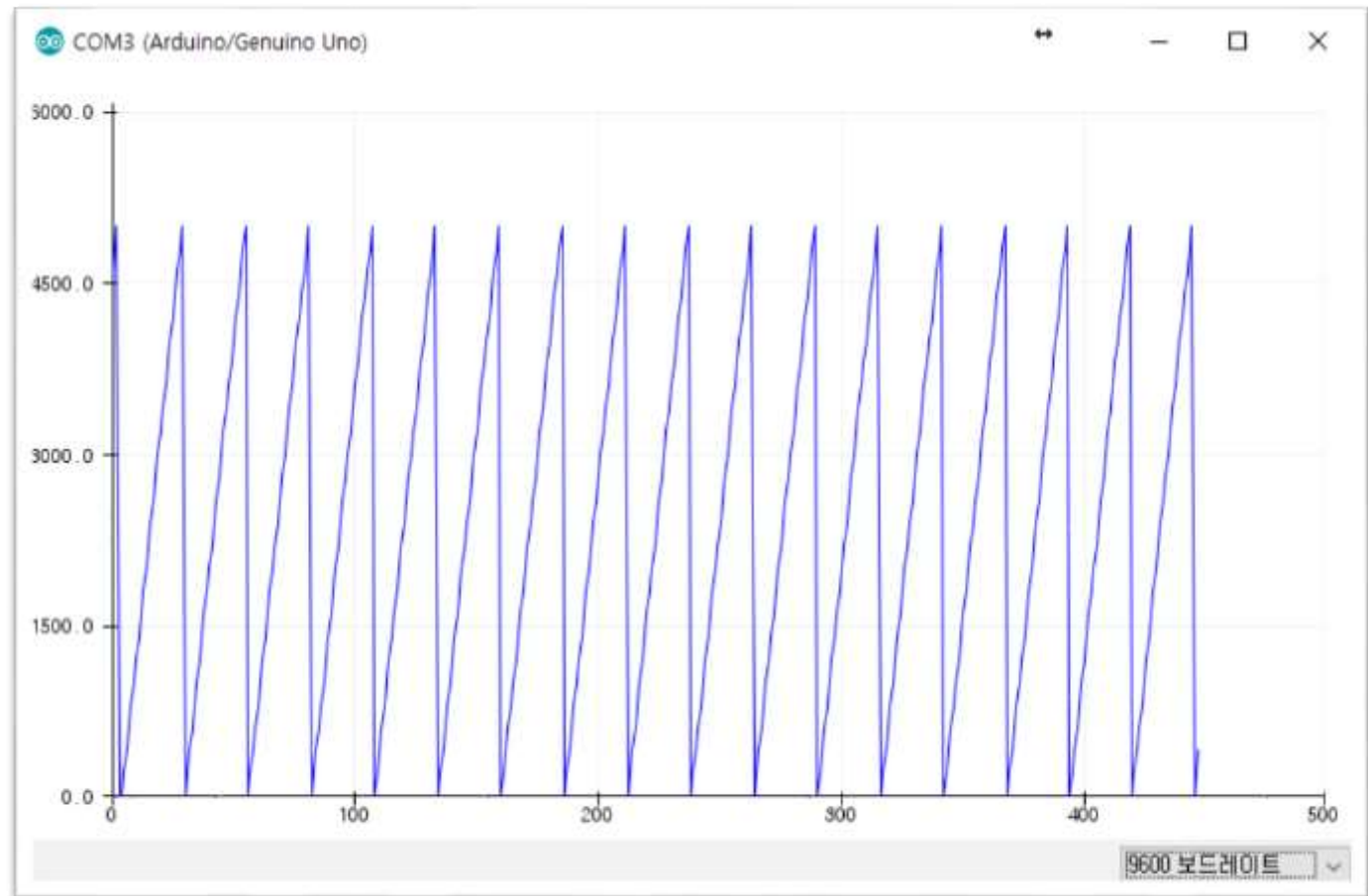
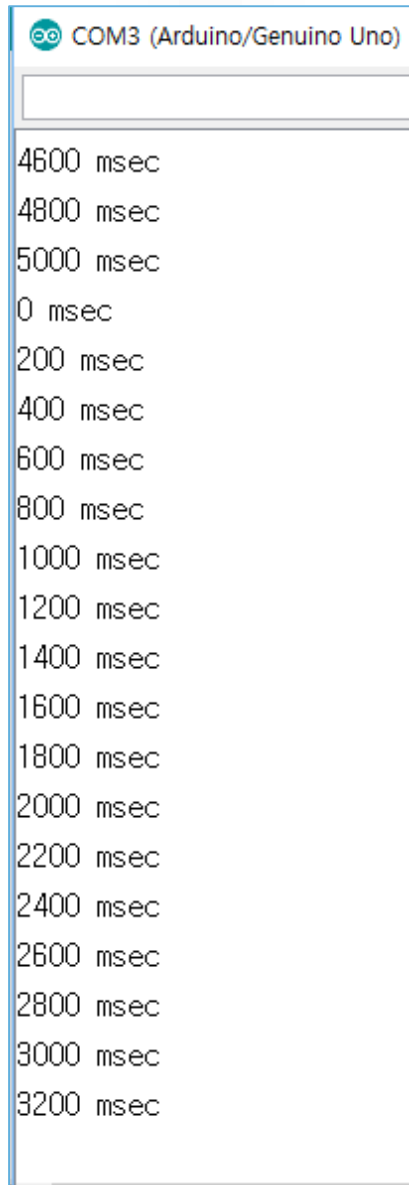
☒ 자동 스크롤

8

Arduino/Genuino Uno on COM3



DIY-1. sawtooth signal



Save ARnn_sawtooth.png



2. 시리얼 통신 (Serial comm.)

2.2

변수 유형별로 컴퓨터에 전송하기

```
*** Hello Arduino ***
```

```
*** char Value ***
```

```
Binary:1000001
```

```
Decimal:65
```

```
Hexadecimal:41
```

```
ASCII:A
```

```
*** int Value ***
```

```
int Value:65
```

```
char(intValue):A
```

```
*** float Value ***
```

```
float Value:65.00
```



DIY-2. Escape from loop()

응용 문제 [DIY-2] 0~15까지 10진수를 2진수와 16진수로 출력하는 스케치를 작성해보자

```
COM3 (Arduino/Genuino Uno)

Number = 0, Binary:0, Hexadecimal:0
Number = 1, Binary:1, Hexadecimal:1
Number = 2, Binary:10, Hexadecimal:2
Number = 3, Binary:11, Hexadecimal:3
Number = 4, Binary:100, Hexadecimal:4
Number = 5, Binary:101, Hexadecimal:5
Number = 6, Binary:110, Hexadecimal:6
Number = 7, Binary:111, Hexadecimal:7
Number = 8, Binary:1000, Hexadecimal:8
Number = 9, Binary:1001, Hexadecimal:9
Number = 10, Binary:1010, Hexadecimal:A
Number = 11, Binary:1011, Hexadecimal:B
Number = 12, Binary:1100, Hexadecimal:C
Number = 13, Binary:1101, Hexadecimal:D
Number = 14, Binary:1110, Hexadecimal:E
Number = 15, Binary:1111, Hexadecimal:F
Mission completed!
```

[Hint]

1. `int number = 0; // starting number`
2. `loop()`에서 1초 간격으로 `number`를 1씩 증가
3. 옆의 방식으로 결과 출력
4. `number`가 15를 초과하면 `loop()` 탈출
`exit(0); // loop 탈출 함수`

ARnn_loop_escape.png



DIY-2. Escape from loop() – code

응용 문제 [DIY-2] 0~15까지 10진수를 2진수와 16진수로 출력하는 스케치를 작성해보자

AR00_loop_escape

```
1 /*
2  DIY-2
3  */
4
5 // start number
6 int number = 0;
7
8 // 문자열 세가지를 설정한다.
9 String stringValue[]{"Binary:", "Hexadecimal:"};
10
11 void setup() {
12   // 9600bps로 시리얼 통신 설정
13   Serial.begin(9600);
14 }
```

```
16 void loop() {
17
18   // 'char Value'를 출력하고 문자열과 숫자를 변수 유형별로 출력한다.
19   Serial.print("Number = ");
20   Serial.print(number);
21   Serial.print(", ");
22   Serial.print(stringValue[0]); // stringValue 중 첫 번째 문자열 출력
23   Serial.print(number, BIN); // 2진수 형태로 출력
24   Serial.print(", ");
25   Serial.print(stringValue[1]); // stringValue 중 첫 번째 문자열 출력
26   Serial.print(number, HEX); // 16진수 형태로 출력
27   // 줄바꿈
28   Serial.println();
29
30   number++; // number 1 증가
31
32   if(number > 15) {
33     Serial.print("Mission completed!");
34     delay(1000);
35     exit(0);
36   }
37
38   delay(1000); // 1초동안 지연시킨다.
39 }
```



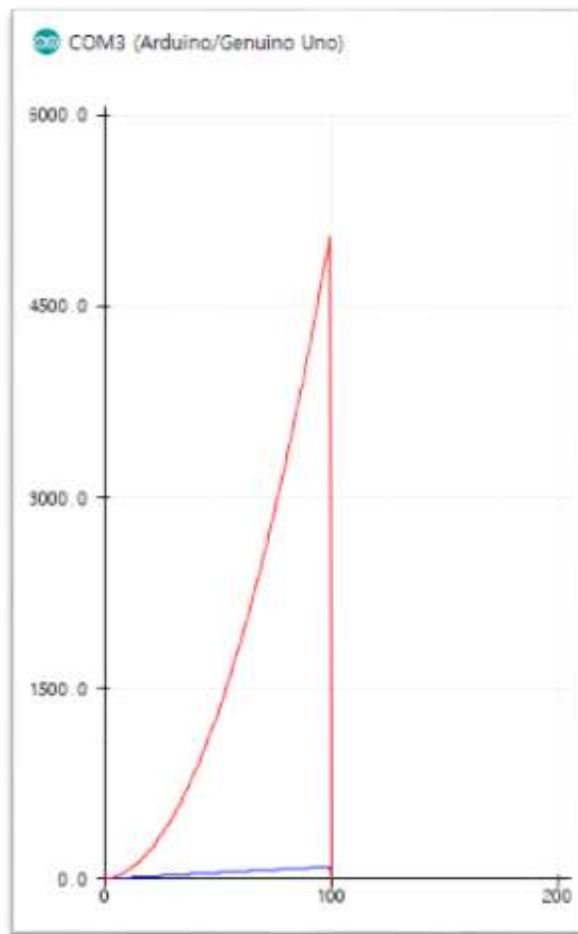
DIY-3. sum from 1 to 100

응용 문제 [DIY-3] Results on serial monitor and plotter

COM10 (Arduino/Genuino Uno)

Number = 86,	Sum = 3741
Number = 87,	Sum = 3828
Number = 88,	Sum = 3916
Number = 89,	Sum = 4005
Number = 90,	Sum = 4095
Number = 91,	Sum = 4186
Number = 92,	Sum = 4278
Number = 93,	Sum = 4371
Number = 94,	Sum = 4465
Number = 95,	Sum = 4560
Number = 96,	Sum = 4656
Number = 97,	Sum = 4753
Number = 98,	Sum = 4851
Number = 99,	Sum = 4950
Number = 100,	Sum = 5050

ARnn: $1 + 2 + \dots + 100 = 5050$



ARnn_sum100.png



DIY-3. sum from 1 to 100 - code

응용 문제 [DIY-3] Results on serial monitor and plotter

```
AR00_sum100
1 /*
2  DIY-3
3  */
4
5 // start number
6 int number = 0;
7 int sum = 0;
8
9 void setup() {
10  // 9600bps로 시리얼 통신 설정
11  Serial.begin(9600);
12 }
```

```
9 void setup() {
10  // 9600bps로 시리얼 통신 설정
11  Serial.begin(9600);
12 }
13
14 void loop() {
15
16  number++;
17  sum += number;
18  Serial.print("Number = ");
19  Serial.print(number);
20  Serial.print(", Sum = ");
21  Serial.println(sum);
22
23  if(number == 100) {
24    Serial.println();
25    Serial.print("ARnn: 1 + 2 + ... + 100 =");
26    Serial.println(sum);
27    delay(1000);
28    exit(0);
29  }
30
31  delay(100); // 0.1초동안 지연시킨다.
32 }
```



2.3 Serial monitor & plotter

2.3

시리얼 통신을 이용하여 데이터 수신하기



EX 2.3

변수 유형별 Arduino에서 컴퓨터로 전송하기 (1/3)

- 실습목표**
1. 컴퓨터에서 Arduino로 0~9의 숫자를 전송한다.
 2. Arduino에서는 전송 받은 숫자만큼 Arduino 보드의 LED를 점멸시킨다.

Hardware Arduino와 PC를 USB 케이블로 연결한다.

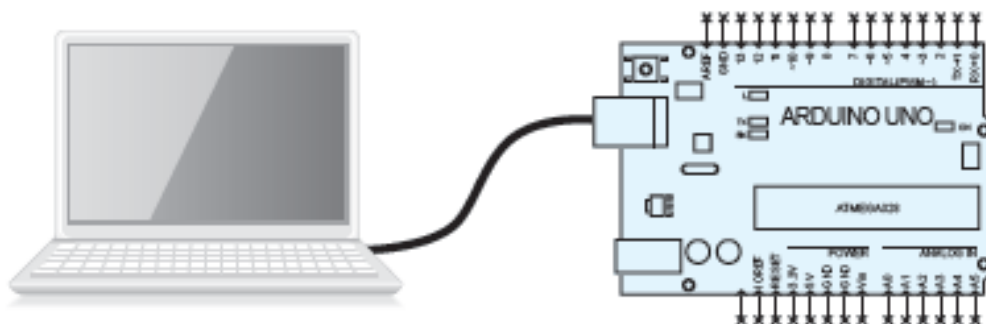


그림 2.1 Arduino와 PC와의 연결



3.3.2 시리얼 통신을 이용하여 데이터 수신하기

EX 2.3

변수 유형별 Arduino에서 컴퓨터로 전송하기 (2/3)

Commands • Serial.[available\(\)](#)

시리얼 통신에 수신된 데이터가 있는지 확인한다. 있을 경우 참(true)의 값을 갖는다.

• Serial.[read\(\)](#)

시리얼 통신을 통하여 수신된 값을 읽는다.

• [isDigit](#)(변수)

변수의 값이 ASCII 코드의 0~9의 숫자 범위에 있는지 여부를 판단. 범위에 있을 경우 참(true)의 값을 갖는다.

• [pinMode](#)(핀번호, 설정)

핀의 입출력 모드를 설정한다. '핀번호'에는 설정하고자 하는 핀의 번호와 '설정'에는 입력으로 사용하기 위해선 'INPUT', 출력으로 사용하기 위해선 'OUTPUT', 입력이며 풀업 사용시 'INPUT_PULLUP'을 적는다.

• [digitalWrite](#)(핀번호, 값)

핀에 디지털 출력(High or Low)을 한다. '핀번호'에는 출력하고자 하는 핀의 번호를, '값'에는 'HIGH' 혹은 'LOW'를 설정하여 High 혹은 Low 출력을 한다.



2.3.3 시리얼 통신을 이용하여 데이터 수신하기

EX 2.3

변수 유형별 Arduino에서 컴퓨터로 전송하기 (3/3)

- Sketch 구성**
1. 13번 핀에 연결된 내장 LED를 이용한다.
 2. 시리얼 통신 상태를 감시한 후 시리얼 통신으로 입력되는 데이터가 있을 때 이를 저장한다.
 3. 전송된 값은 ASCII 코드값이므로 이를 숫자로 변경한다.
 4. 숫자만큼 LED를 0.2초 간격으로 점멸시킨다.

실습 결과 IDE의 시리얼 모니터를 실행시켜 전송란에 0~9의 값을 입력한 후 Arduino의 LED가 입력한 값 만큼 점멸하는지를 확인해 본다..

- 응용 문제**
1. 0~9의 입력 값에 따라 점멸 주기가 변화하는 스케치를 작성해 보자.
 2. 0~9의 숫자를 전송하면 전송된 수의 2진수와 16진수를 컴퓨터로 전송하는 스케치를 만들어보자. (hint: 예제 2.2를 참고하자)



2.3.3 시리얼 통신을 이용하여 데이터 수신하기

EX 2.3

변수 유형별 Arduino에서 컴퓨터로 전송하기 (code)

ex_2_3_final

```
1 /*
2  예제 2.3
3  컴퓨터로부터 시리얼 통신을 통하여 데이터 수신하기
4 */
5
6 // LED 출력을 할 핀 번호 설정
7 const int ledPin = 13;
8
9 // 점멸횟수 변수 설정
10 int blinkNumber = 0;
11
12 void setup() {
13   // 9600bps로 시리얼 통신 설정
14   Serial.begin(9600);
15   // 13번 핀을 출력으로 설정
16   pinMode(ledPin, OUTPUT);
17 }
18
```

```
19 void loop() {
20   // 시리얼 통신으로 입력 받은 데이터가 있는지를 검사하여
21   // 데이터가 있을 경우에 if문 안의 명령어를 실행
22   if (Serial.available()) {
23     // val 변수에 시리얼 통신값 읽어오기
24     char val = Serial.read();
25     // 입력된 값이 0~9의 숫자인지를 판단
26     if (isdigit(val)) {
27       // val은 ASCII 코드값이므로 숫자로 바꿔주기 위하여
28       // '0'의 아스키 코드값을 빼줌
29       // blinkNumber에는 실제 숫자가 저장된다.
30       blinkNumber = (val - '0');
31     }
32     Serial.print("입력한 수:");
33     Serial.println(blinkNumber);
34     // Serial.println();
35     delay(2000);
36
37     // blinkNumber 만큼 LED의 켜짐상태를 길게 유지.
38     for (char i = 0; i < blinkNumber; i++) {
39       digitalWrite(ledPin, HIGH);
40       delay(100);
41       digitalWrite(ledPin, LOW);
42       delay(100);
43     }
44   }
45   // 점멸 횟수를 리셋함
46   blinkNumber = 0;
47 }
```



DIY-4. 점멸 주기가 변화

응용 문제 [DIY-4] 0~9의 입력 값에 따라 점멸 주기가 변화하는 스케치를 작성해 보자.

- 시리얼모니터에 입력한 수를 표시
- 입력한 수에 비례해서 LED 켜 상태를 길게 유지.



완성된 스케치 code를

ARnn_period.ino

로 저장해서 제출.



DIY-5. 입력된 수를 변환하여 출력

응용 문제 [DIY-5] 0~9의 숫자를 전송하면 입력 값에 따라 점멸 주기를 변화시키며, 전송된 수의 2진수와 8진수를 컴퓨터로 전송하는 스케치를 만들어보자.
(hint: 예제 2.2를 참고하자).

- 아래 출력 참조.



완성된 스케치 code를

ARnn_number.ino

로 저장해서 제출.



3. LCD

Liquid Crystal Display



핀에 직접 연결
7 핀



I²C 통신
2 핀

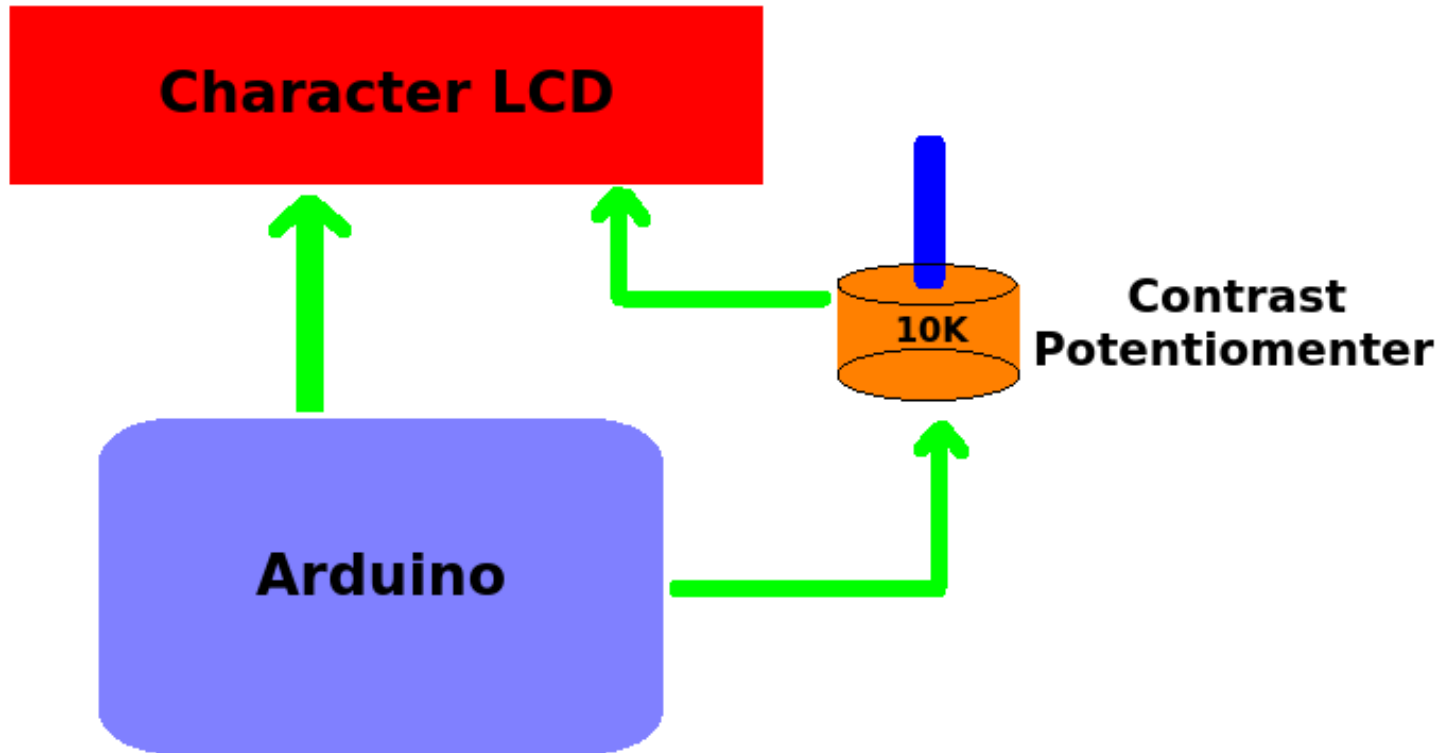


얇은 액정판 아래 조명을 비추는 장치로서 액정판의 전류 흐름을 제어하여 문자나 그림을 표시

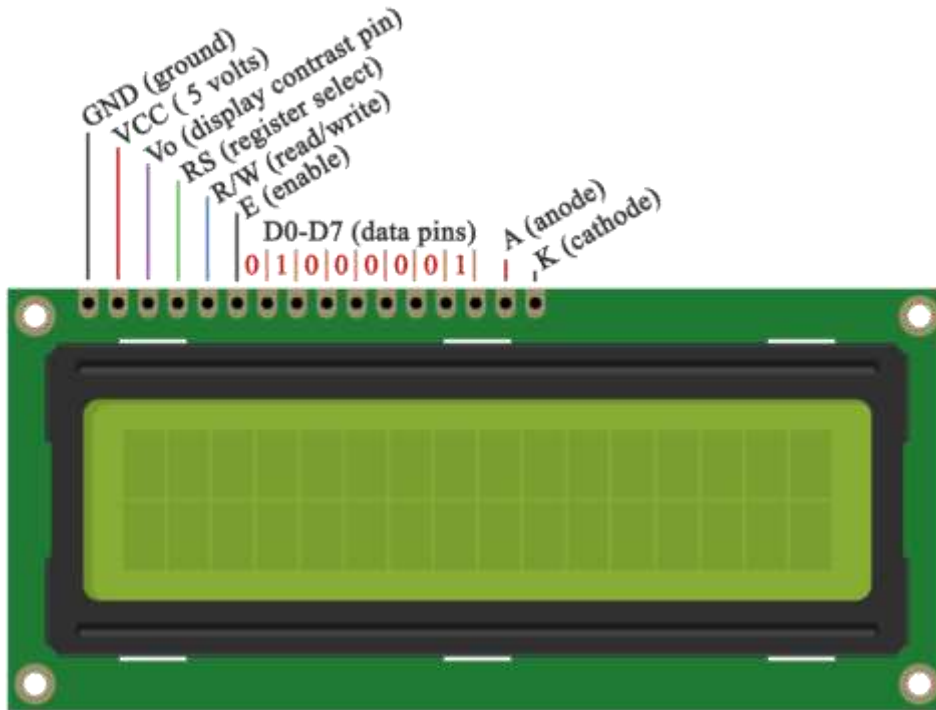
Liquid crystal display

- 3.1 입출력 핀을 이용하여 LCD 모듈에 표시하기
- 3.2 I²C를 이용한 LCD 출력

3.1.1 Introduction to LCD Module



LCD (Liquid Crystal Display, 16 X 2)

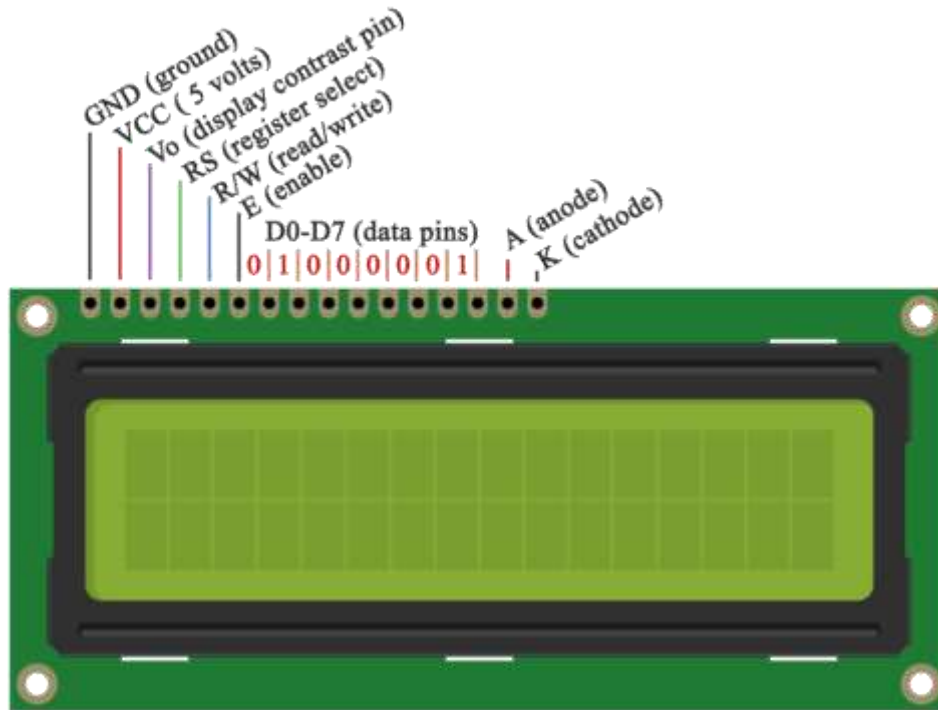


(1,2, ... 15,16)

1. GND
2. VCC (+5V)
3. Vo (contrast, 가변저항기 연결)
4. RS
5. R/W
6. E
- D0 ~ D7 (data, 7~14)
- A (15, Backlight+, 220 or 330 Ω)
- K (16, Backlight-)

3.1.3 Introduction to LCD Module

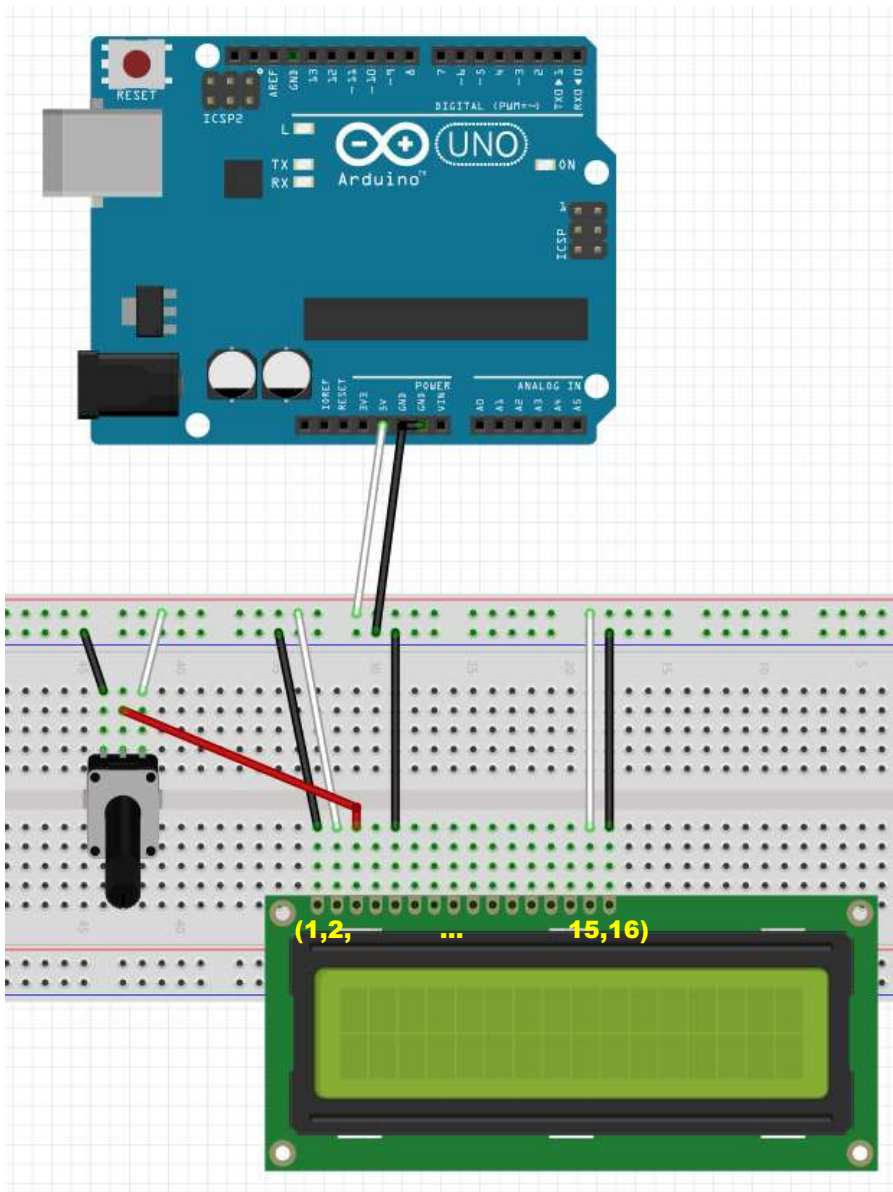
LCD (Liquid Crystal Display, 16 X 2)



(1,2, ... 15,16)

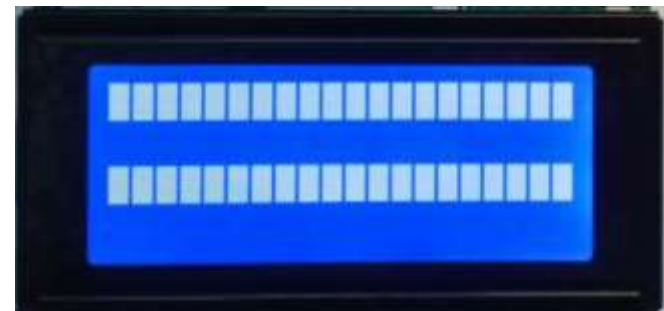
Pin 1 to Arduino GND
 Pin 2 to Arduino +5V
 Pin 3 to wiper
 Pin 4 to Arduino pin D12
 Pin 5 to Arduino GND
 Pin 6 to Arduino pin D11
 Pin 11 to Arduino pin D5
 Pin 12 to Arduino pin D4
 Pin 13 to Arduino pin D3
 Pin 14 to Arduino pin D2
 Pin 15 to +5V (with 220 or 330 Ω)
 Pin 16 to GND

3.1.4 LCD 초기화 (pin-1, 2, 3, 5, 15,16)

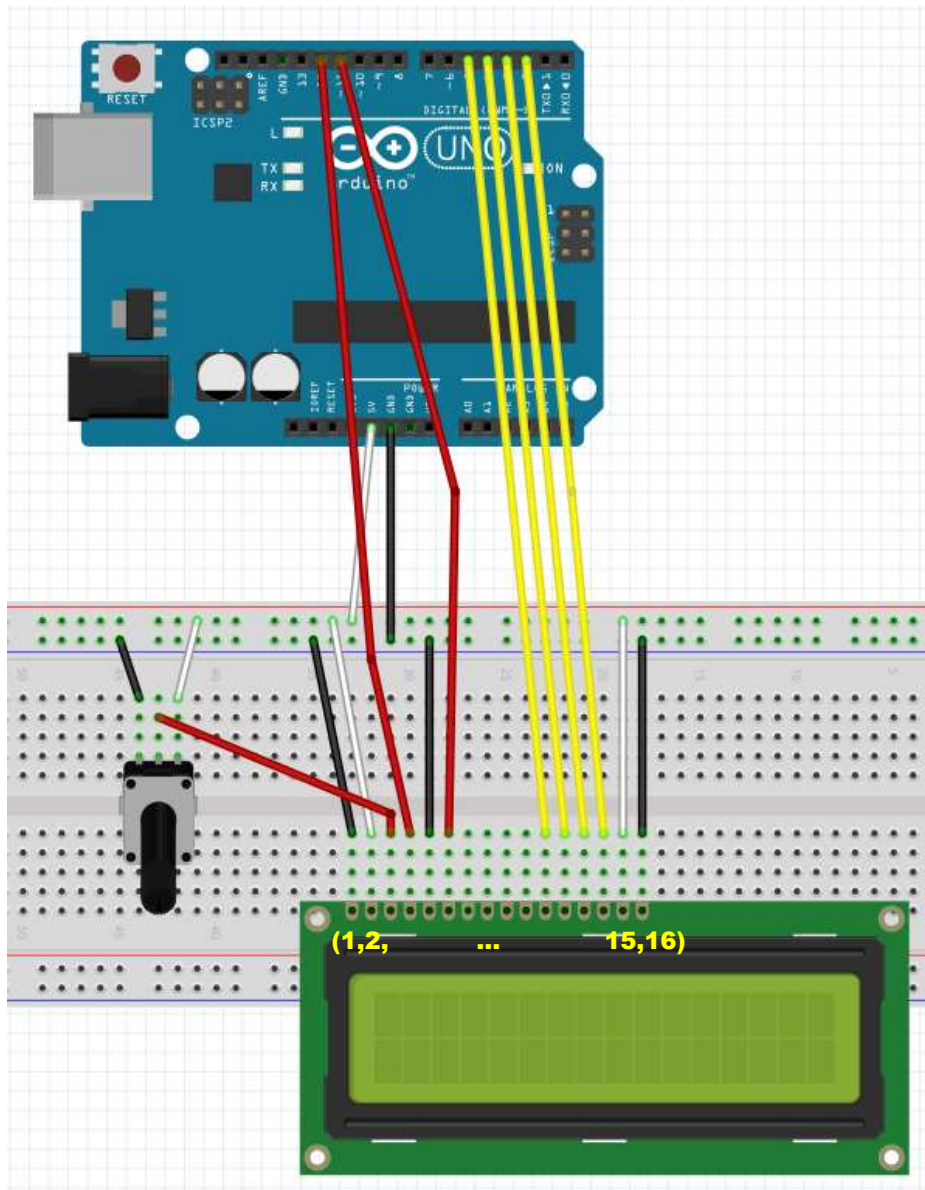


Pin 1 to Arduino GND
Pin 2 to Arduino +5V
Pin 3 to wiper (potentiometer)
Pin 5 to Arduino GND
Pin 15 to +5V
Pin 16 to GND

전원 연결 후
LCD 초기화



3.1.5 데이터 입력 초기화 (pin-4, 6, 11,12,13,14)



Pin 1 to Arduino GND

Pin 2 to Arduino 5V

Pin 3 to wiper

Pin 4 to Arduino pin D12

Pin 5 to Arduino GND

Pin 6 to Arduino pin D11

Pin 11 to Arduino pin D5

Pin 12 to Arduino pin D4

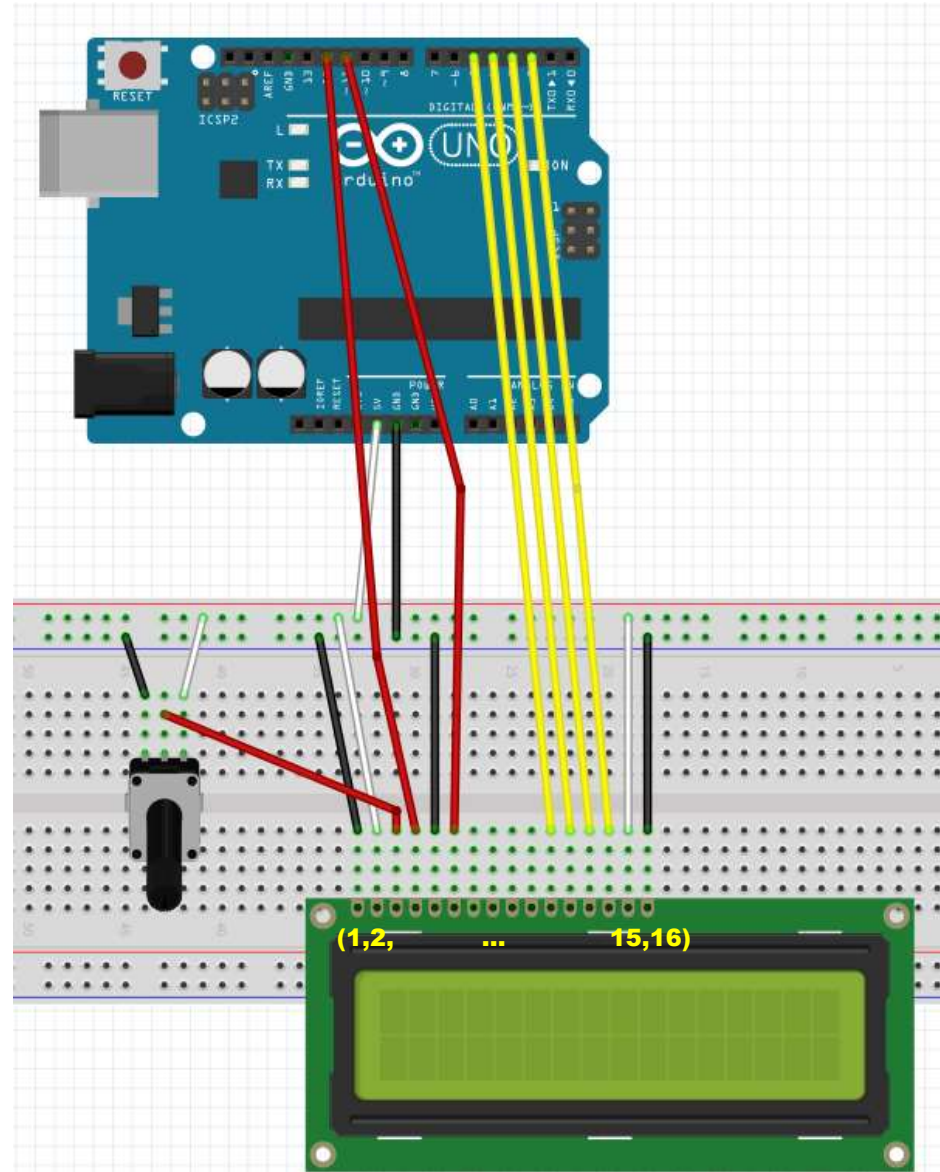
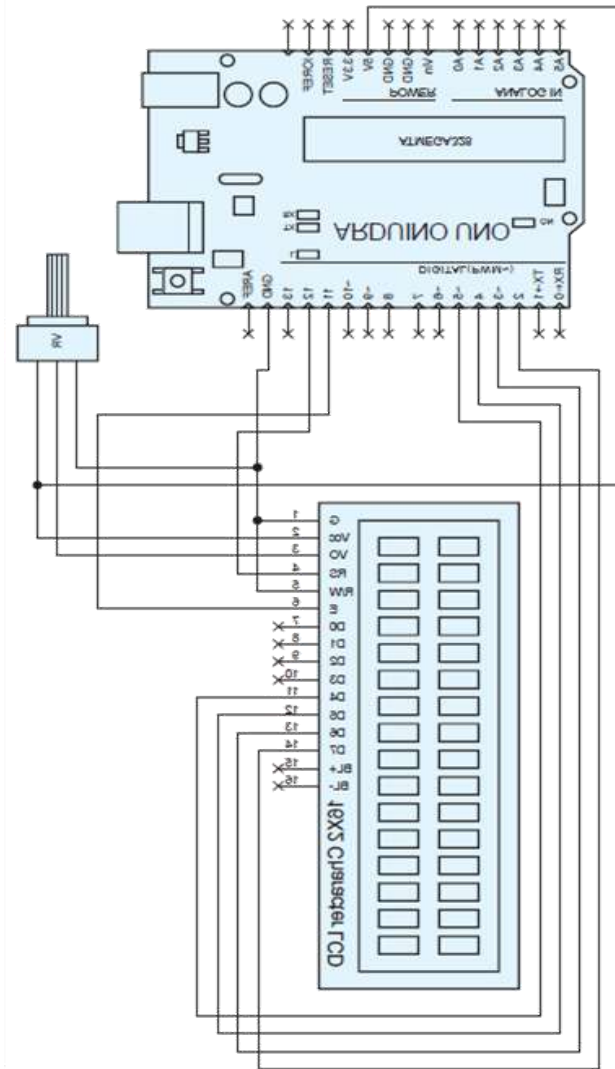
Pin 13 to Arduino pin D3

Pin 14 to Arduino pin D2

Pin 15 to +5V

Pin 16 to GND

DIY-6. LCD module circuit



3.1.6 Introduction to LCD - code “Hello ARnn”

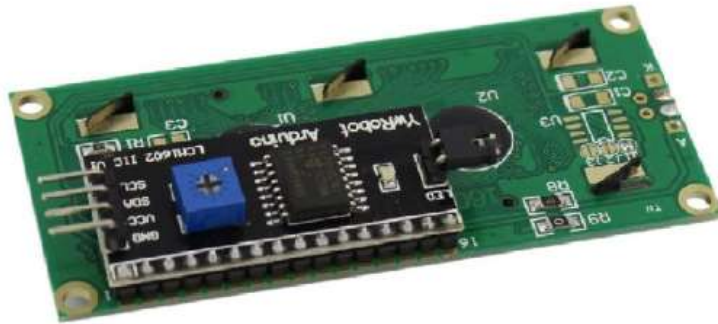
- `LiquidCrystal lcd(rs, en, d4, d5, d6, d7)`
lcd란 이름으로 I2C에 연결된 LCD 모듈 객체.
- `lcd.begin(행, 열)`
lcd란 이름의 LCD 모듈의 크기를 정의한다.
- `lcd.clear()`
lcd란 이름의 LCD 모듈의 화면의 모든 표시를 지우고 커서를 왼쪽 위로 옮긴다.
- `lcd.home()`
lcd란 이름의 LCD 모듈의 커서를 왼쪽 위로 옮긴다.
- `lcd.setCursor(행, 열)`
lcd란 이름의 LCD 모듈의 커서를 원하는 위치로 이동시킨다.
- `lcd.print(데이터)`
lcd란 이름의 LCD 모듈에 데이터를 출력한다.
- `lcd.noBacklight();`
lcd란 이름의 LCD 모듈의 백라이트를 소등한다.
- `lcd.backlight();`
lcd란 이름의 LCD 모듈의 백라이트를 점등한다.

3.1.7 Introduction to LCD - code “Hello ARnn”

hello_LCD

```
7 // include the library code:
8 #include <LiquidCrystal.h>
9
10 // initialize the library with the numbers of the interface pins
11 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
12
13 void setup() {
14   // set up the LCD's number of columns and rows:
15   lcd.begin(16, 2);
16   // Print a message to the LCD.
17   lcd.print("Hello, ARnn!");
18 }
19
20 void loop() {
21   // set the cursor to column 0, line 1
22   lcd.setCursor(0, 1); // second line, first column
23   // print the number of seconds since reset:
24   lcd.print(millis() / 1000);
25   lcd.print(" sec");
26 }
```


3.2 I²C를 이용한 LCD 출력



I²C(아이스퀘어드시, **Inter-Integrated Circuit**)는 필립스에서 개발한 직렬 버스이다. 마더보드, 임베디드 시스템, 휴대 전화 등에 저속의 주변 기기를 연결하기 위해 사용된다.

I²C 는 풀업 저항이 연결된 직렬 데이터(**SDA**)와 직렬 클럭(**SCL**)이라는 두 개의 양 방향 오픈 컬렉터 라인을 사용한다. 최대 전압은 **+5 V** 이며, 일반적으로 **+3.3 V** 시스템이 사용되지만 다른 전압도 가능하다.

<https://ko.wikipedia.org/wiki/I%C2%B2C>

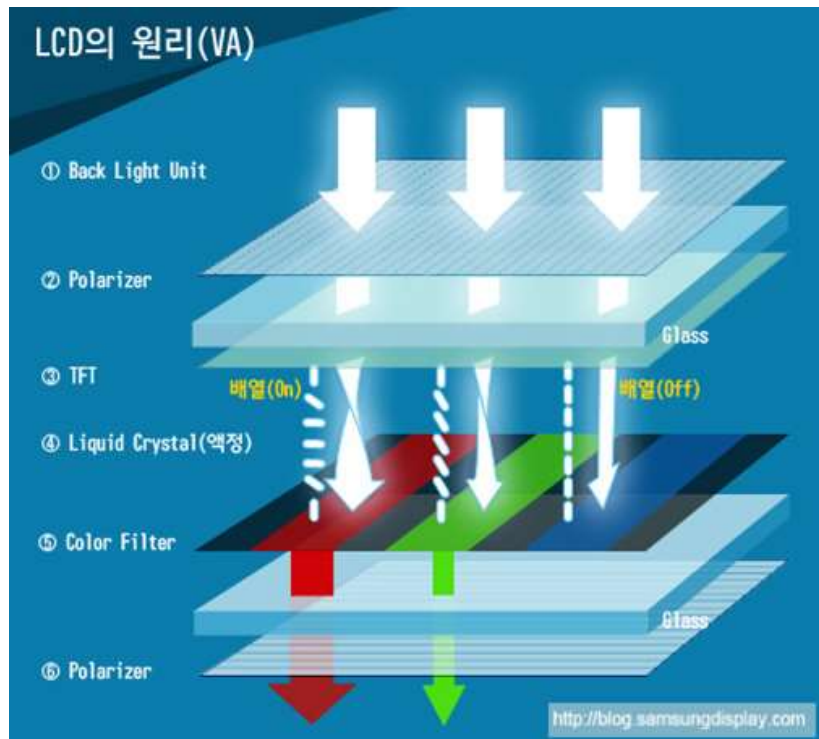
<http://www.ifuturetech.org/product/16x2-lcd-i2c-lcd/>



특수문자나 기호는
LCD 모듈 문자표를 참고하여 출력

표 3.1 LCD 모듈 문자표

Upper 4 Bits	Lower 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			00P`P										一タミαp			
xxxx0001	(2)			!1AQa4										。アチムäq			
xxxx0010	(3)			"2BRbr										「イツメβθ			
xxxx0011	(4)			#3CScs										」ウテモe∞			
xxxx0100	(5)			\$4DTdt										、エトヤμΩ			
xxxx0101	(6)			%5EUeu										・オナユσÜ			
xxxx0110	(7)			&6FVfv										ヲカニヨρΣ			
xxxx0111	(8)			'7GWgw										フキヌラgπ			
xxxx1000	(1)			<8HXhx										イクネリJX			
xxxx1001	(2)			>9IYiy										ウケノル-y			
xxxx1010	(3)			*:JZjz										エコハレjチ			
xxxx1011	(4)			+;K[k{										オサヒロ*万			
xxxx1100	(5)			,<L¥ll										ヤシフワφ円			
xxxx1101	(6)			-=M]m}										ユスヘンも÷			
xxxx1110	(7)			.>N^n÷										ヨセホ"ん			
xxxx1111	(8)			/?O_o+										ッソマ°ö			



3.2.2 I²C를 이용한 LCD 출력

I²C (Inter Integrated Circuit)

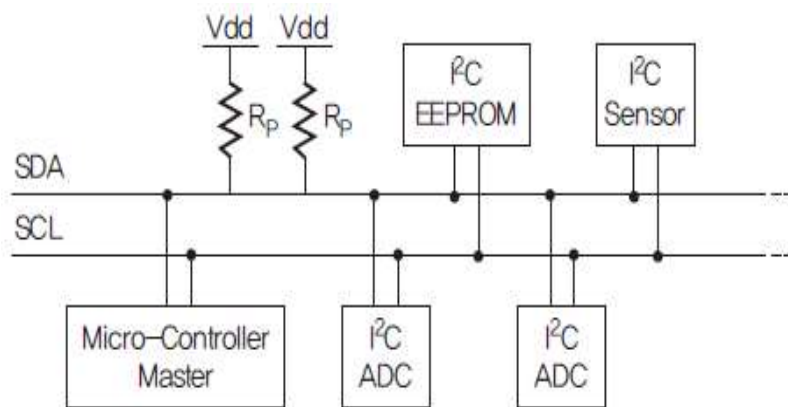
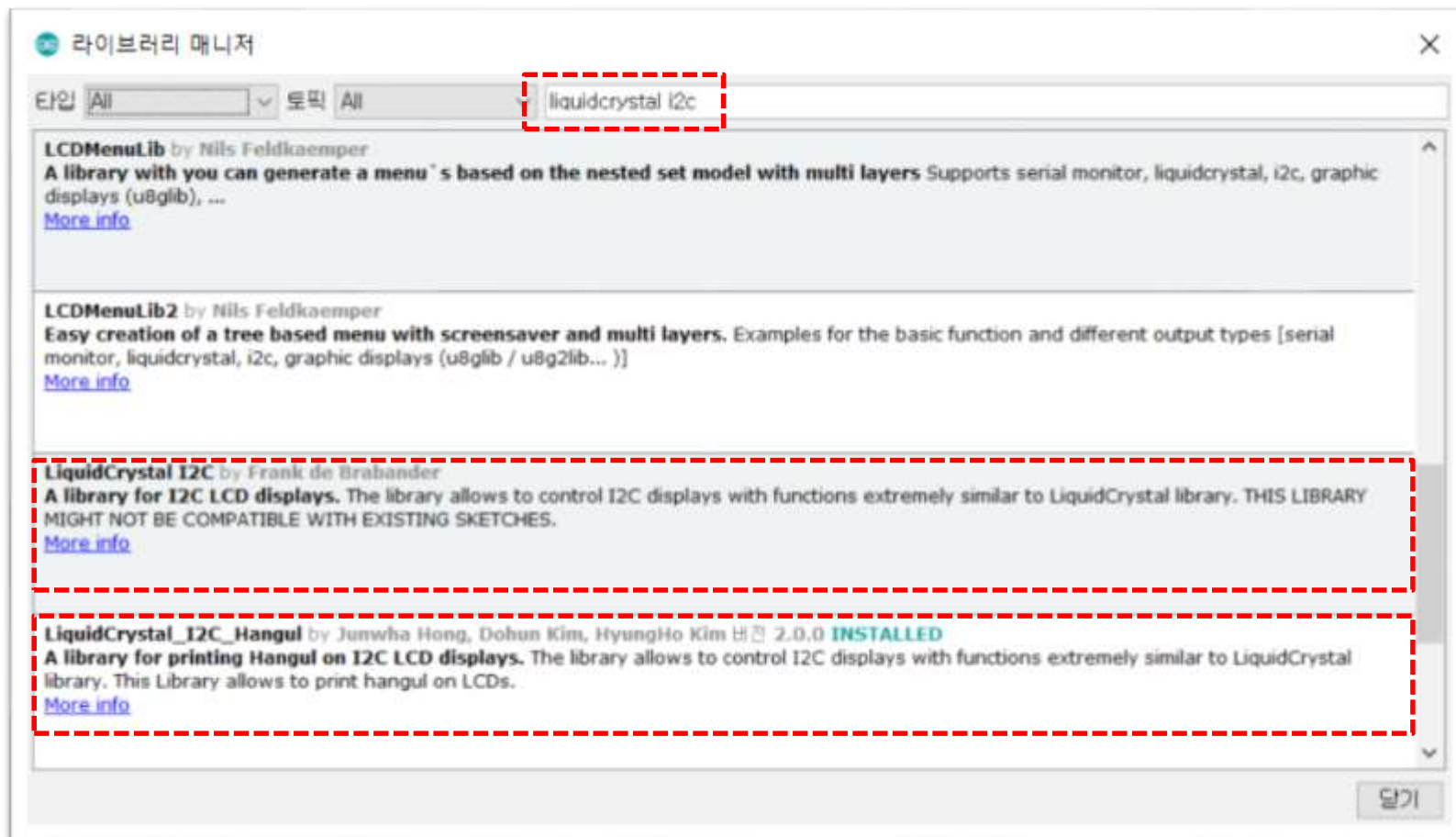


그림 3.2 I²C를 이용한 네트워크

- ✓ Phillips사에서 개발된 규격이며 TWI라고도 함.
- ✓ SDA(Serial Data line), SCL(Serial Clock Line) 두 선으로 통신
- ✓ Master와 Slave로 구분되어 Master에서 통신을 주관
- ✓ 최대 112개의 노드를 연결 가능하고 최고 3.4Mbps의 속도
- ✓ LCD 모듈을 I²C 통신으로 제어하기 위해선 PCF8574 IC를 사용
- ✓ SDA, SCL 두 개의 입출력 핀만 필요

라이브러리 매니저를 이용하여 I²C LCD용 라이브러리(LiquidCrystal I2C)를 설치

스케치 > 라이브러리 포함하기 > 라이브러리 관리



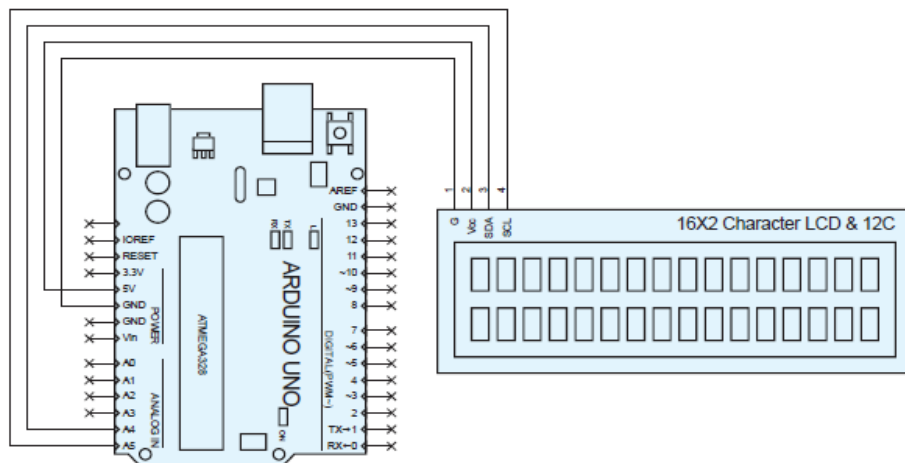
3.2.4 I²C를 이용한 LCD 출력

EX 3.2

I²C를 이용한 LCD 출력 (1/3)

- 실습목표**
1. 16X2 도트매트릭스 LCD를 I²C를 이용하여 제어한다.
 2. 'Welcome' 메시지와 함께 백라이트를 점멸시킨다.
 3. 시리얼 포트에 입력 받은 값을 LCD에 출력한다.

- Hardware**
1. I²C LCD 모듈과 Arduino는 전원핀 Vcc, GND와 I²C 통신핀 SDA, SCL이 연결되어야 한다.
 2. I²C LCD 모듈의 Vcc와 GND를 Arduino의 5V와 GND에 연결한다.
 3. SDA는 A4에, SCL은 A5에 연결한다.



3.2.5 I²C를 이용한 LCD 출력

EX 3.2 I²C를 이용한 LCD 출력 (2/3)

Commands

- LiquidCrytral_I2C(I2C 주소, 가로 글자수, 세로 글자수)
LCD 모듈이 연결된 I2C 주소와 LCD의 가로, 세로 글자수를 설정한다.
- lcd.init(); LCD 모듈을 설정한다.
- lcd.clear(): lcd란 이름의 LCD 모듈의 화면의 모든 표시를 지우고 커서를 왼쪽 위로 옮긴다.
- lcd.home(): lcd란 이름의 LCD 모듈의 커서를 왼쪽 위로 옮긴다.
- lcd.setCursor(행, 열): lcd란 이름의 LCD 모듈의 커서를 원하는 위치로 이동시킨다.
- lcd.print(데이터): lcd란 이름의 LCD 모듈에 데이터를 출력한다.
- lcd.noBacklight(): lcd란 이름의 LCD 모듈의 백라이트를 소등한다.
- lcd.backlight(): lcd란 이름의 LCD 모듈의 백라이트를 점등한다.

Sketch 구성

1. I²C 방식의 LCD 모듈을 사용하기 위해 앞서 다운받은 라이브러리를 추가해 준다.
2. 라이브러리의 함수를 이용하여 LCD를 설정해 준다.
3. `setup()`에서 'Welcome'메시지와 백라이트를 점멸시킨다.
4. 시리얼 통신으로 데이터를 입력받기위해서 시리얼 통신 설정을 해 준다.
5. 데이터 입력이 있을 때 이를 LCD에 출력해 준다.

EX 3.2 I²C를 이용한 LCD 출력 (code)

```
ex_3_2_final
6 // I2C 통신 라이브러리 설정
7 #include <Wire.h>
8 // I2C LCD 라이브러리 설정
9 #include <LiquidCrystal_I2C.h>
10
11 // LCD I2C address 설정 PCF8574:0x27, PCF8574A:0x3F
12 LiquidCrystal_I2C lcd(0x3F,16,2); // LCD address:0x27,
```



```
14 void setup()
15 {
16   // 9600 bps로 시리얼 통신 설정
17   Serial.begin(9600);
18   lcd.init(); // LCD 설정
19   lcd.clear(); // LCD를 모두 지운다.
20   lcd.backlight(); // 백라이트를 켜다.
21   // Arduino LCD, Welcome 표시
22   lcd.setCursor(0,0);
23   lcd.print("Arduino LCD");
24   delay(3000);
25   lcd.setCursor(0,1);
26   lcd.print("Welcome");
27   delay(250);
28
29   // LCD 백라이트를 두 번 점멸
30   lcd.noBacklight();
31   delay(250);
32   lcd.backlight();
33   delay(250);
34   lcd.noBacklight();
35   delay(250);
36   lcd.backlight();
37   delay(3000);
38
39   // Open Serial Monitor, Type to display 표시
40   lcd.clear();
41   lcd.setCursor(0,0); //Start at character 0 on line 0
42   lcd.print("Open Serial Mntr");
43   lcd.setCursor(0,1);
44   lcd.print("Type to display");
45 }
```

3.2.6 I²C를 이용한 LCD 출력 (code-2)

EX 3.2 I²C를 이용한 LCD 출력 (code)



```

47 void loop()
48 {
49     // 시리얼 통신 수신 값이 있을 때
50     if (Serial.available()) {
51         delay(100);
52         // 모두 삭제
53         lcd.clear();
54         // 커서를 좌측 상단으로
55         lcd.setCursor(0,0);
56         // "Message from PC" 출력
57         lcd.print("Message from PC");
58         // 커서를 두 번째 줄로
59         lcd.setCursor(0,1);
60
61         // LCD에 PC에서 전송된 데이터를 출력
62         while (Serial.available() > 0) {
63             lcd.write(Serial.read());
64         }
65     }
66 }

```


EX 3.2 I²C를 이용한 LCD 출력 (3/3)

- 실행 결과
1. Arduino LCD 표시 후 백라이트가 2회 점멸한다.
 2. 시리얼 모니터를 실행 시킨 후 메시지를 입력하여 보자. → “Hello ARnn”
 3. 메시지가 LCD에 출력되는지를 확인해 보자.

Take a photo of LCD screen.

**Save photo as
ARnn_LCD_hello.png**



DIY-7

시리얼 통신으로 입력 받은 1~9의 숫자에 대하여 LCD의 백라이트가 입력된 숫자만큼 점멸하고 점멸 횟수를 LCD에 표시하는 스케치를 작성해 보자.

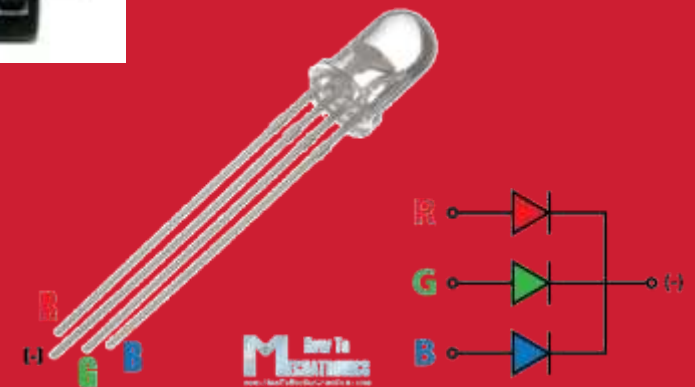
(hint: 예제 2.3을 참고하자)

Save ARnn_LCD.ino



4. LED

Light Emitting Diode

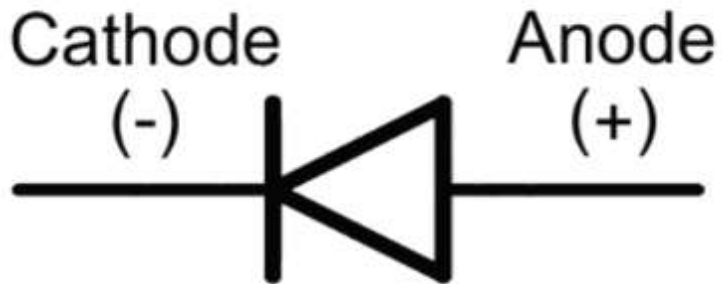


LED (Light Emitting Diode)

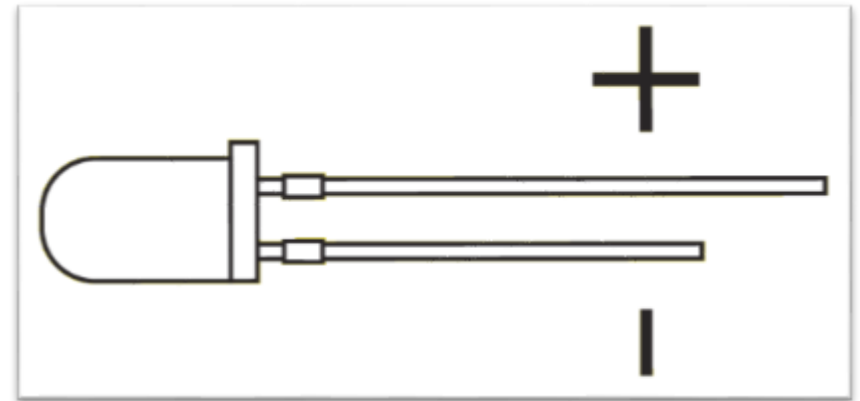
- ✓ 전기 신호를 빛으로 출력하는 반도체 소자
- ✓ 고효율, 반영구적 수명
- ✓ 가정용 실내등, 산업용 특수등, 자동차용 전조등 및 실내등에 사용



Polarity of Diode and LED



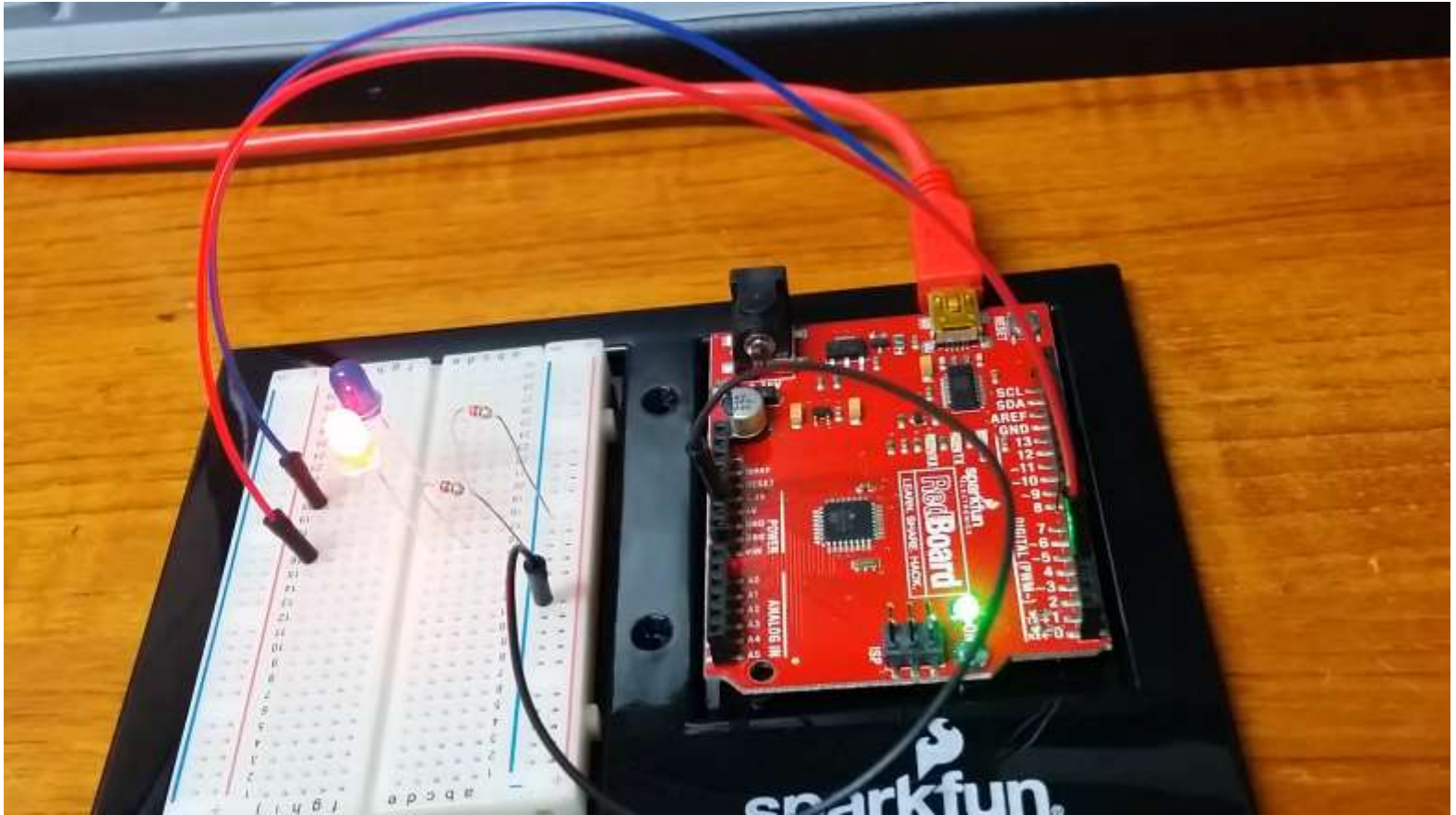
The diode circuit symbol, with the anode and cathode marked.



Find the longer leg, which should indicate the positive, anode pin.

<https://learn.sparkfun.com/tutorials/polarity/diode-and-led-polarity>

4.1 LED 교차 점멸

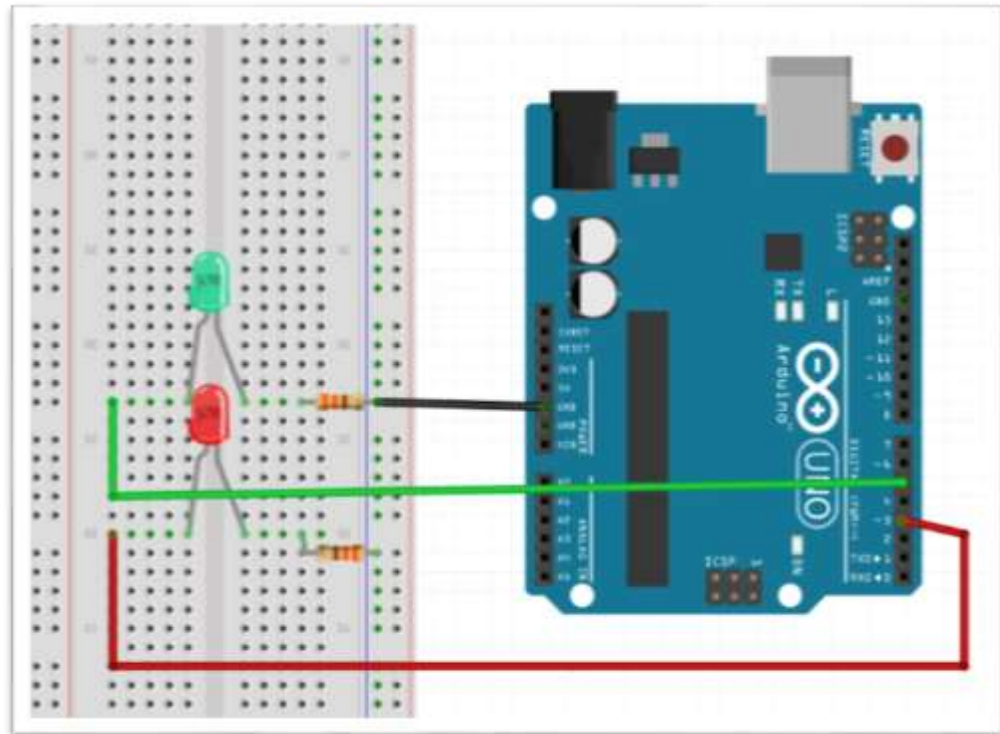
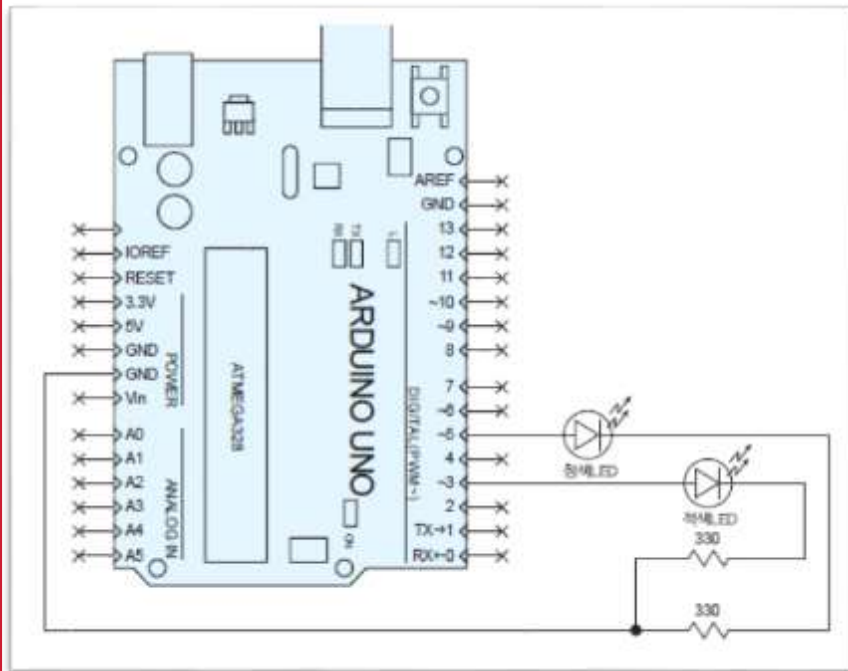


4.1.1 LED control - 교차 점멸

EX 4.1 LED 교차 점멸 (1/3)

실습목표 두 개의 LED를 0.1초 간격으로 교차하여 점멸시키자.

Hardware



Save ARnn_2LED.fzz

4.1.2 LED control - 교차 점멸

EX 4.1

LED 교차 점멸 (2/3)

Commands

- `pinMode`(핀번호, 설정)

핀의 입출력 모드를 설정한다. '핀번호' 에는 설정하고자 하는 핀의 번호와 '설정'에는 입력으로 사용하기 위해선 'INPUT', 출력으로 사용하기 위해선 'OUTPUT', 입력이며 풀업 사용시 'INPUT_PULLUP'을 설정한다.

- `digitalWrite`(핀번호, 값)

핀에 디지털 출력 (High or Low) 을 한다. '핀번호' 에는 출력하고자 하는 핀의 번호를, '값'에는 'HIGH' 혹은 'LOW' 를 설정하여 High 혹은 Low 출력을 한다.

Sketch 구성

1. LED의 핀 번호를 설정한다.
2. `setup()`에서는 LED 출력으로 사용할 핀을 출력핀으로 설정한다.
3. `loop()`에서는 하나의 LED를 켜 후 일정시간이 지난 후에 소등하고, 다른 LED를 켜다.

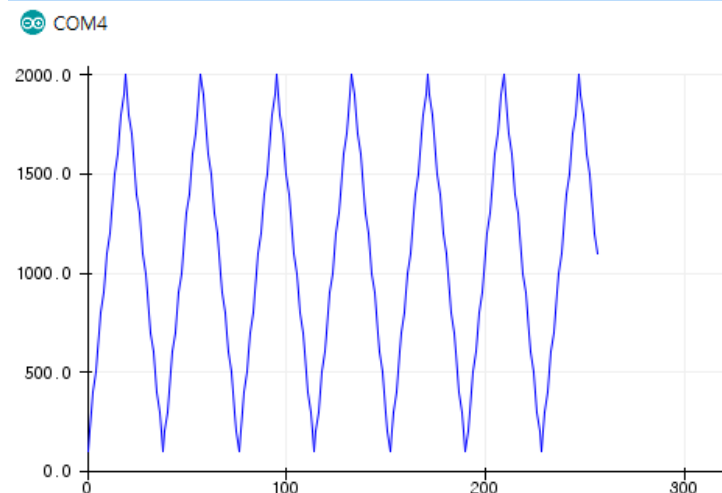
4.1.3 LED control - 교차 점멸

EX 4.1 LED 교차 점멸 (3/3)

실습 결과 LED A와 B가 0.1초 단위로 교차하며 점멸한다.

응용 문제 점멸 주기가 0.1초부터 2초로 0.1초 단위로 증가하였다가 다시 반대로 2초부터 0.1초까지 감소하는 동작을 반복하는 스케치를 작성해 보자.
(hint: delay 명령어의 괄호 안의 숫자를 증감시킨다.)

delay = 1600 msec	delay = 500 msec
delay = 1700 msec	delay = 400 msec
delay = 1800 msec	delay = 300 msec
delay = 1900 msec	delay = 200 msec
delay = 2000 msec	delay = 100 msec
delay = 1900 msec	delay = 200 msec
delay = 1800 msec	delay = 300 msec
delay = 1700 msec	delay = 400 msec
delay = 1600 msec	delay = 500 msec



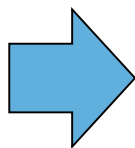
4.1.4 LED control - 교차 점멸 (code)

ex_4_1

```

1  /*
2  예제 4.1
3  LED 점멸
4  */
5
6  const int ledA   = 3;
7  const int ledB   = 5;
8
9  void setup()
10 {
11    pinMode(ledA, OUTPUT);
12    pinMode(ledB, OUTPUT);
13 }
14
15 void loop()
16 {
17    digitalWrite(ledA,HIGH);
18    digitalWrite(ledB,LOW);
19    delay(100);
20    digitalWrite(ledA,LOW);
21    digitalWrite(ledB,HIGH);
22    delay(100);
23 }

```



```

6  const int ledA   = 3;
7  const int ledB   = 5;
8
9  int number = 1;
10 boolean flag = true;
11
12 void setup()
13 {
14    Serial.begin(9600);
15    pinMode(ledA, OUTPUT);
16    pinMode(ledB, OUTPUT);
17 }

```

완성된 스케치 code를
ARnn_2led.ino
로 저장해서 제출.

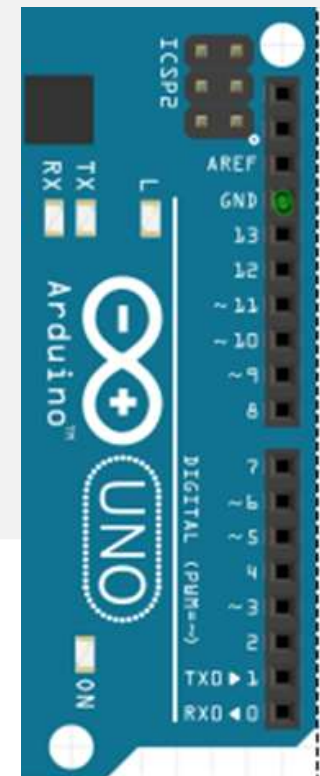
```

19 void loop()
20 {
21    digitalWrite(ledA, HIGH);
22    digitalWrite(ledB, LOW);
23    delay(100 * number);
24    digitalWrite(ledA, LOW);
25    digitalWrite(ledB, HIGH);
26    Serial.print("delay = ");
27    Serial.print(100 * number);
28    Serial.println(" msec");
29    delay(100 * number);
30
31    if (flag) {
32        number++;
33    } else {
34        number--;
35    }
36
37    if (number >= 20) {
38        Fill in your code!
39    }
40    else if (number == 1) {
41        flag = true;
42    }
43 }

```

밝기 조절 : 디밍 (Dimming)

- ✓ LED에 입력되는 전력은 **PWM (Pulse Width Modulation)**을 이용하여 조절.
- ✓ PWM : 고속의 스위칭으로 High와 Low 신호의 비율을 조절하여
LED의 밝기, 모터의 회전 등을 조절하는 방법
- ✓ Arduino에서는 **analogWrite()** 명령어로 구현
- ✓ Arduino UNO의 경우 **3, 5, 6, 9, 10, 11 번 핀이 PWM을 지원한다.**



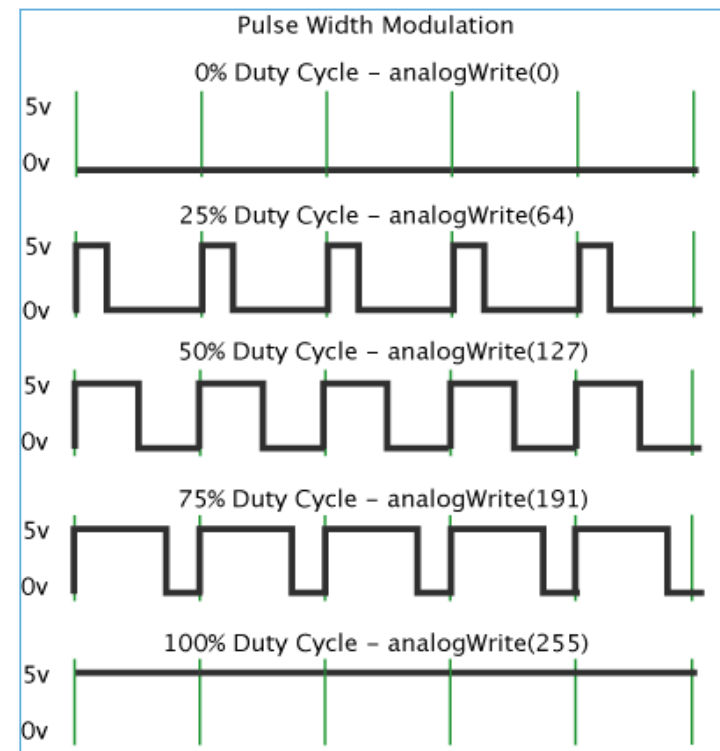
4.2.1 LED control - 밝기 조절

PWM (Pulse Width Modulation)

Using [analogWrite\(pin, pwm_value\)](#) function in fading an LED off and on. AnalogWrite uses [pulse width modulation \(PWM\)](#), turning a digital pin on and off very quickly with different ratio between on and off, to create a fading effect.

A call to [analogWrite\(\)](#) is on a scale of **0 - 255**, such that `analogWrite(255)` requests a 100% duty cycle (always on), and `analogWrite(127)` is a 50% duty cycle (on half the time)

PWM frequency = 500 Hz

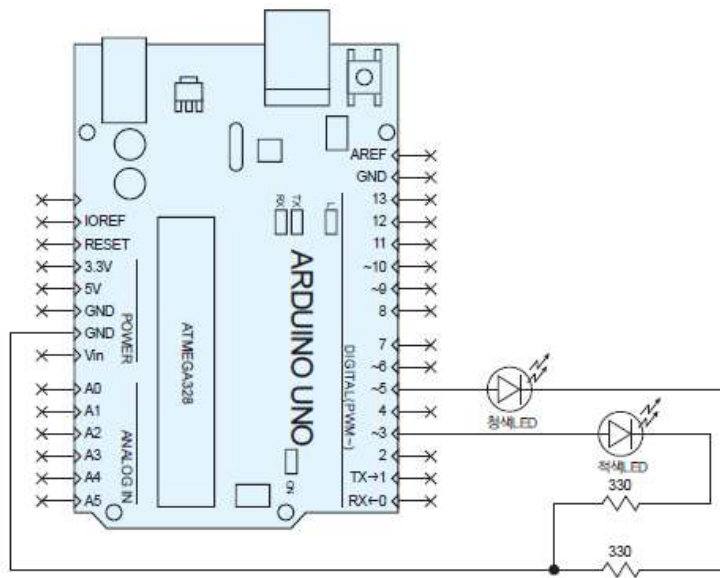


<https://www.arduino.cc/en/Tutorial/PWM>

EX 4.2 LED 밝기 조절 (1/2)

- 실습목표**
1. 두 개의 LED의 밝기를 조절하자.
 2. 각각의 LED가 교차하며 밝아졌다 어두워 졌다를 반복하도록 하자.

- Hardware**
1. 청색과 적색 LED의 Anode핀을 Arduino의 3번 5번 핀에 연결한다.
 2. Cathode핀에 330 Ω 저항을 연결하여 저항의 반대쪽은 Arduino의 GND에 연결한다.
 3. LED가 연결된 핀에 HIGH신호가 출력될 때 LED가 점등된다.



EX 4.2

LED 밝기 조절 (2/2)

Commands • analogWrite(핀번호, 값)

정해진 핀에 아날로그 출력을 한다. '값' 에는 0~255의 값을 넣는다.

Sketch 구성

1. LED의 핀 번호를 설정한다.
2. setup()에서는 LED 출력으로 사용할 핀을 출력핀으로 설정한다.
3. 밝기를 저장할 변수를 설정한다.
4. 하나의 LED가 밝아질 때 다른 LED는 어두워져야 하므로 이를 조절할 변수를 설정한다.
5. loop()에서는 밝기와 밝기 변수 증감을 위한 변수를 조절하여 두 개의 LED를 교차 점멸시키는 동작을 반복한다.

실험 결과 LED A와 B가 밝기가 변화하며 점멸한다.

응용 문제

1. 네개의 다른 색깔의 LED를 Arduino에 연결한다.
2. 네개의 LED가 순서대로 디밍하는 스케치를 작성해보자.

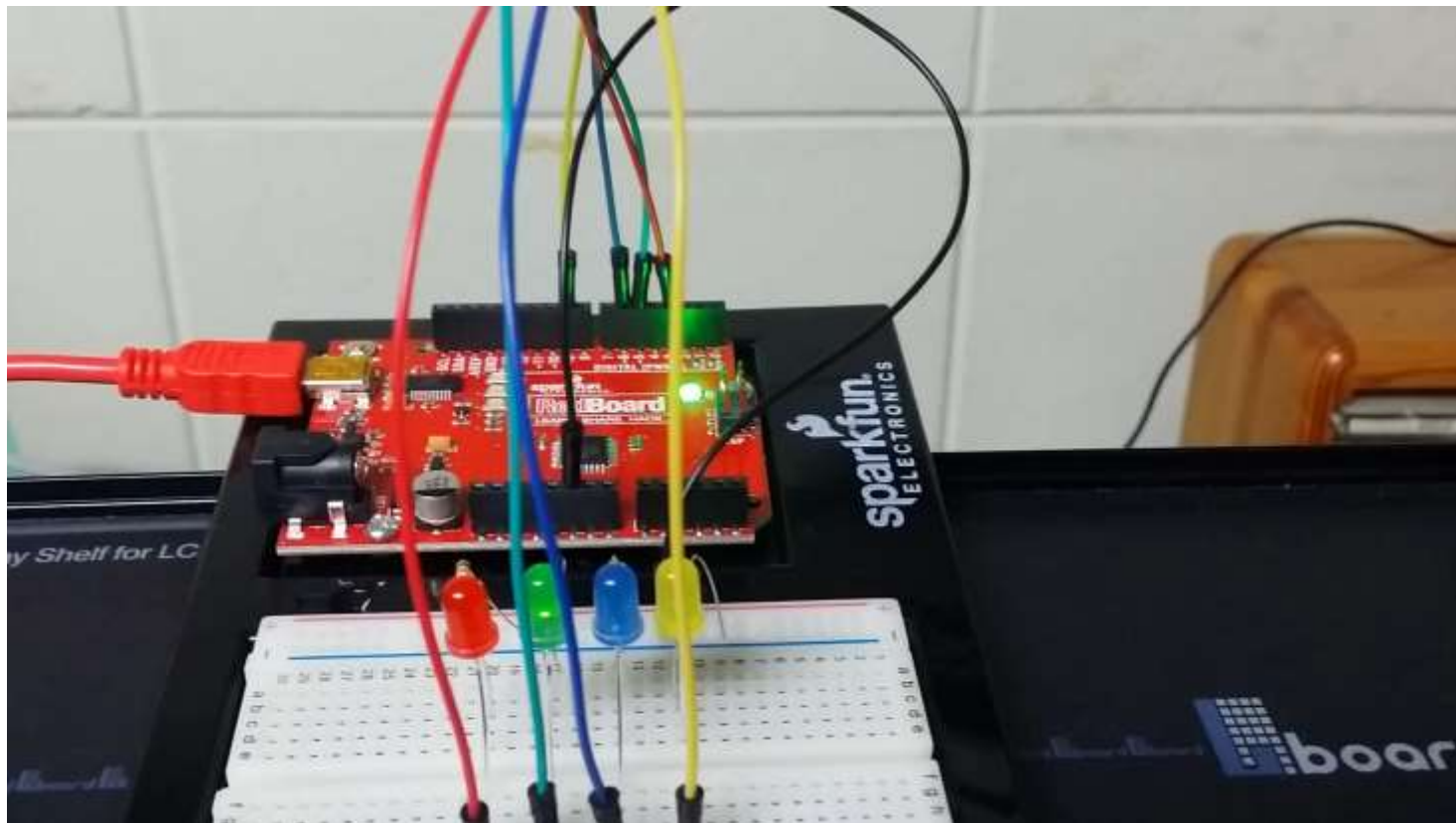
4.2.4 LED control - 밝기 조절 (code)

```

ex_4_2_start
1  /*
2   예제 4.2
3   LED 밝기 조절
4  */
5
6  const int ledA   = 3;    //LED A를 3번핀에 연결
7  const int ledB   = 5;    //LED B를 5번핀에 연결
8  int brightness = 0;      //밝기를 조절하기 위한 변수
9  int increment = 1;       //밝기 변수 증감을 위한 변수
10
11 void setup()
12 {
13  // analogWrite 핀에는 별도의 설정이 불필요하다.
14 }
15
16 void loop()
17 {
18  analogWrite(ledA,brightness); // LED A 밝기 조절
19  analogWrite(ledB,255-brightness); // LED B 밝기 조절
20
21  brightness = brightness + increment; // 밝기 조절
22  if((brightness >= 255)|| (brightness <= 0)) increment = -increment; // 밝기 변수 증감 방향 변경
23  delay(10);    // 0.01 초간 지연
24 }
  
```

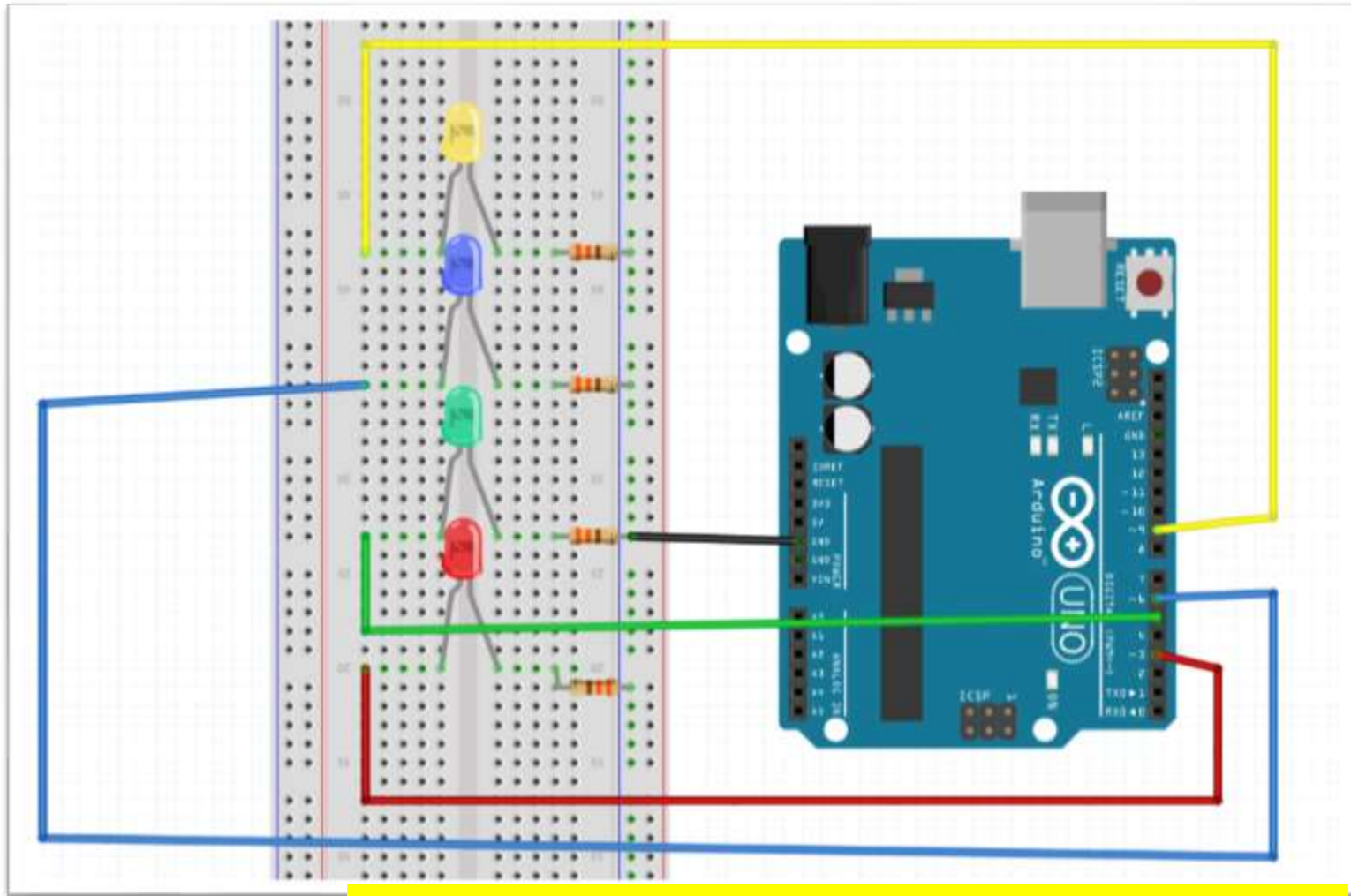
4.2.5 LED control - DIY

- DIY. 1. 네개의 다른 색깔의 LED를 Arduino에 연결한다.
2. 네개의 LED가 순서대로 디밍하는 스케치를 작성해보자.



4.2.5 LED control - DIY

DIY. 1. 네개의 다른 색깔의 LED를 Arduino에 연결한다. (pwm pin: 3,5,6,9)



완성된 회로를 [ARnn_4led.fzz](#)
로 저장해서 제출.

4.2.5 LED control - DIY: code-1

ARnn_4_led_start.ino

```

1  /*
2  Dimming 4 leds
3  */
4
5  int ledR = 3; // LED connected to digital pin 3
6  int ledG = 5;
7  int ledB = 6;
8  int ledY = 9;
9
10 int dimTime = 20;
11
12 void setup() {
13   // nothing happens in setup
14 }

```

```

16 void loop() {
17   // fade in from min to max in increments of 5 points:
18   for(int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5) {
19     // sets the value (range from 0 to 255):
20     analogWrite(ledR, fadeValue);
21     // wait for 30 milliseconds to see the dimming effect
22     delay(dimTime);
23   }
24
25   // fade out from max to min in increments of 5 points:
26   for(int fadeValue = 255 ; fadeValue >= 0; fadeValue -=5) {
27     // sets the value (range from 0 to 255):
28     analogWrite(ledR, fadeValue);
29     // wait for 30 milliseconds to see the dimming effect
30     delay(dimTime);
31   }

```

각 led에 동일한 dimming code 적용

```

33 for(int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5) {
34   // sets the value (range from 0 to 255):
35   analogWrite(ledG, fadeValue);
36   // wait for 30 milliseconds to see the dimming effect
37   delay(dimTime);
38 }

```

4.2.5 LED control - DIY: code-2

```

1  /*
2  Dimming 4 leds
3  */
4
5  int ledR = 3; // LED connected to digital pin 3
6  int ledG = 5;
7  int ledB = 6;
8  int ledY = 9;
9
10 int dimTime = 20;
11
12 void setup() {
13   // nothing happens in setup
14 }

```

완성된 스케치 code를

ARnn_4led.ino

로 저장해서 제출.

```

16 void loop() {
17   // fade ledR
18   dimLed(ledR);
19   // fade ledG
20   dimLed(ledG);
21   // fade ledB
22   dimLed(ledB);
23   // fade ledY
24   dimLed(ledY);
25 }
26 void dimLed(int led) {
27   // fade in from min to max in increments of 5 points:
28   for(int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5) {
29     // sets the value (range from 0 to 255):
30     analogWrite(led, fadeValue);
31     // wait for 20 milliseconds to see the dimming effect
32     delay(dimTime);
33   }
34   // fade out from max to min in increments of 5 points:
35   for(int fadeValue = 255 ; fadeValue >= 0; fadeValue -=5) {
36     // sets the value (range from 0 to 255):
37     analogWrite(led, fadeValue);
38     // wait for 20 milliseconds to see the dimming effect
39     delay(dimTime);
40   }
41 }

```

각 led에 동일한 dimming code 적용

dimLed(int led) 반복 사용



[Practice]

◆ [wk03]

- **Arduino LCD & LED-I.**
- **Complete your project**
- **Submit folder : ARnn_Rpt02**

wk03 : Practice-02 : ARnn_Rpt02

◆ [Target of this week]

- Complete your works
- Save your outcomes
- Upload all in github.

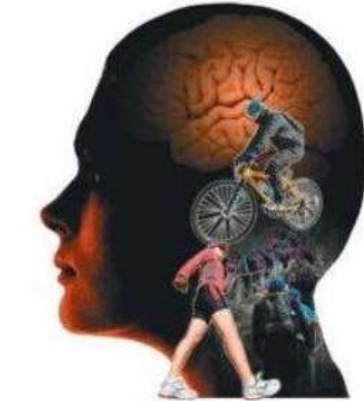
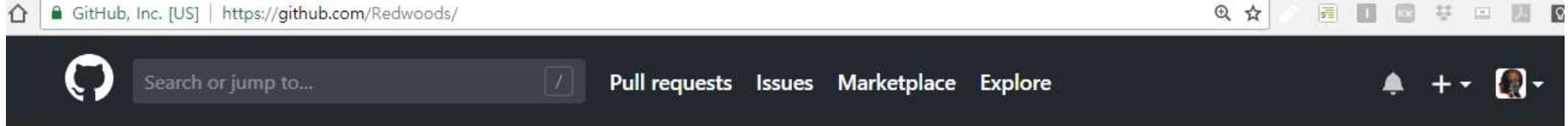
제출폴더명 : **ARnn_Rpt02**

■ 압축할 파일들

- ① **ARnn_period.ino**
- ② **ARnn_number.ino**
- ③ **ARnn_LCD_hello.png**
- ④ **ARnn_LCD.ino**

● References & good sites

- ✓ <http://www.arduino.cc> Arduino Homepage
- ✓ <http://www.github.com> GitHub
- ✓ <http://www.google.com> Googling
- ✓ <https://www.youtube.com> Youtube



Redwoods Yi

Redwoods

Add a bio

GimHae, Republic of Korea

chaos21c@gmail.com

Overview

Repositories 7

Stars 2

Followers 1

Following 0

Pinned repositories

Customize your pinned repositories

Py

Lectures on coding python from scratch to the advanced level.

Jupyter Notebook

Arduino

Lectures on learning Arduino from scratch to the advanced level in iot environment.

Lec

All lectures by Redwoods in Inje University

Jupyter Notebook

hw-coding

Resource for lecture of Hardware Programming (2017, Inje university)


Arduino

171 contributions in the last year

Contribution settings

Redwoods/Arduino: Lect

GitHub, Inc. [US] | https://github.com/Redwoods/Arduino






Search or jump to...

Pull requests

Issues

Marketplace

Explore



Redwoods / Arduino

Unwatch

1

Star

0

Fork

0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

Lectures on learning Arduino from scratch to the advanced level in iot environment.

Edit

Add topics

2 commits

1 branch

0 releases

1 contributor

Branch: master


New pull request


Create new file


Upload files


Find file

Clone or download


 Redwoods 2018 start Latest commit 38ca9e0 28 minutes ago

 ar-basic 2018 start 28 minutes ago

 ar-iot 2018 start 28 minutes ago

 README.md Initial commit 43 minutes ago

README.md



Arduino

Lectures on learning Arduino from scratch to the advanced level in iot environment.

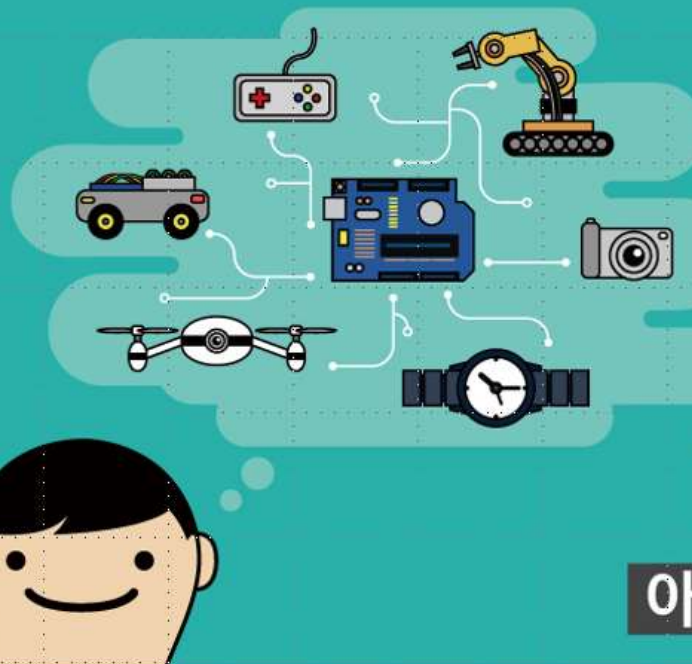




아두이노 고급 키트

High-Level Kit for Arduino

LEVEL 03



당신의 상상을
실현해줄
아두이노스토리

http://arduinostory.com/goods/goods_view.php?goodsNo=1000000306

상급키트 구성품

1 1EA 아두이노 우노 R3 DIP 아두이노 우노 R3 (DIP) 호환보드 기본 메인보드입니다.	2 1EA 9V 배터리 홀더 9V 배터리를 연결하여 아두이노에 외부전원을 공급할 수 있습니다.	3 1EA 7세그먼트 4채널 7세그먼트가 4개 연결된 형태의 부품입니다. 총 12개의 핀을 사용합니다.	4 1EA 7세그먼트 1채널 공통 음극 7세그먼트 시계나 점수 등의 숫자를 표현 할 때 많이 사용됩니다.
5 1EA 74HC595N 기본 메인보드입니다. 74HC595N LED, 드로메트릭스, NFD 제어 IC 입니다.	6 1EA 65핀 점퍼 와이어 브레드보드에 연결할 때 사용하는 65핀 점퍼와이어 입니다.	7 1EA 무지개 점퍼선 F-M 20cm M타입과 F타입이 양쪽으로 달린 무지개 점퍼선입니다.	8 1EA 투명 부품 케이스 대,소 키트 구성품을 담을 수 있는 투명 부품 케이스입니다.
9 1EA 가변저항10K 물리변 저항값이 바뀝니다. (0~10KΩ)	10 1EA 1602 I2C LCD 아두이노 16x2 I2C LCD 모듈입니다. LCD입니다.	11 1EA 저항 100, 220, 330, 1K, 2K, 4.7K, 10K, 47K, 100K	12 1EA 브레드 보드 830홀 브레드 보드 830홀(봉무형) 센서 테스트나, 회로 프로토타입을 작성할 때 사용됩니다.

13 1EA 수동부저 아두이노의 tone함수를 통해 소리를 내는 부저입니다.	14 6EA 택트스위치 (12x12x7) 스위치를 누르고 있을 경우만 ON됩니다.	15 1EA 택트스위치 컵 (파랑, 노랑, 초록, 빨강, 하양) 택트스위치를 사용할 때 스위치간의 구분을 할 수 있습니다.	16 3EA 조도센서 빛을 감지하거나 빛의 밝기를 아날로그로 출력해주는 CDS 센서입니다.
17 1EA LED 5mm (빨강, 노랑, 초록, 하양, 파랑) 기본으로 사용되는 LED입니다. 동작전압 : 2.2~2.4V 사용전류 : 20mA 미만	18 1EA 헤더핀 1x40/2.54mm 핀 간격은 2.54mm이며 헤더핀의 길이는 약 1.15cm입니다.	19 1EA USB케이블 50cm PC와 아두이노 우노 보드를 연결하여 프로그램을 다운로드 할 때 사용합니다.	20 1EA 저항값 카드 저항값을 쉽게 확인 할 수 있는 카드입니다. 사이즈 : 60mm x 50mm
21 1EA 능동부저 Signal 단자가 HIGH 일 때 약 2.5kHz의 음이 발생됩니다.	22 1EA 5V 1채널 릴레이 모듈 아두이노의 디지털 핀과 모듈 하단의 IN 핀들을 연결해 릴레이를 제어할 수 있는 모듈입니다.	23 1EA 8x8 도트 매트릭스 모듈 LED로 다양한 연출을 할 수 있습니다.	24 1EA 4x4 16 키패드 모듈 16개의 버튼을 사용할 수 있습니다.

아두이노 키트(Kit) : Part-2

<p>25 1EA</p> <p>무선 리모콘 키트</p> <p>핵파선을 사용해서 리모콘 기능을 구현할 수 있습니다.</p>	<p>26 2EA</p> <p>가열기 센서 스위치</p> <p>센서의 가열기에 따라 스위치 역할을 합니다.</p>	<p>27 1EA</p> <p>사운드 센서 모듈</p> <p>아두이노와 호환되는 사운드센서 모듈입니다.</p>	<p>28 1EA</p> <p>불꽃 센서</p> <p>근거리 화재, 불꽃을 감지하는 센서입니다.</p>	<p>37 1EA</p> <p>DC 5V 스텝 모터</p> <p>28BYJ48 스텝 모터 중 저렴한 편에 속하는 모델입니다. 5개의 핀을 사용합니다.</p>	<p>38 1EA</p> <p>DS1302 RTC 모듈</p> <p>아두이노 등 마이크로컨트롤러에서 사용이 가능합니다.</p>	<p>39 1EA</p> <p>아두이노 우노 프로토 쉴드</p> <p>UNO 보드에서 회로를 간단히 짜기 위해 보드 위에 얹어 사용하는 쉴드입니다.</p>	<p>40 1EA</p> <p>3축 가속도 센서 모듈</p> <p>가속도를 측정할 수 있는 센서입니다.</p>
<p>29 1EA</p> <p>모터 드라이버 모듈</p> <p>ULN2003 스텝 모터 드라이버 모듈 5V ~ 12V를 사용할 수 있습니다.</p>	<p>30 1EA</p> <p>LM35 온도 센서</p> <p>온도를 마닐로그 값으로 출력합니다.</p>	<p>31 1EA</p> <p>수위 센서 모듈</p> <p>센서 역할에 잠긴 정도를 마닐로그 값으로 출력합니다.</p>	<p>32 1EA</p> <p>SG90 서보모터</p> <p>Vcc, GND, 신호선, 총 3개의 핀이 있습니다. 로봇팔이나 자동차, 비행기 조종에 사용됩니다.</p>	<p>41 1EA</p> <p>5V DC모터</p> <p>5V DC모터</p>	<p>42 1EA</p> <p>인체 감지 센서 모듈</p> <p>핵파선을 이용해 움직임 감지하는 센서입니다. 오선이 감지되면 HIGH 신호를 출력합니다.</p>	<p>43 5EA</p> <p>다이오드 1N4001</p> <p>다이오드 1N4001</p>	<p>44 5EA</p> <p>세라믹 캐패시터 (22pF)</p> <p>세라믹 캐패시터 (22pF)</p>
<p>33 1EA</p> <p>초음파 거리 센서 모듈</p> <p>5V를 사용하여 만직 거리는 2cm에서 500cm입니다.</p>	<p>34 1EA</p> <p>조이스틱 모듈</p> <p>기본적으로 조이스틱 모듈은 두개의 가변저항이 서로 수직으로 회전하는 형태로 되어 있습니다.</p>	<p>35 1EA</p> <p>온습도 센서 모듈</p> <p>아두이노 온습도 센서중 가장 대중적으로 사용되는 DHT11 디지털 센서입니다.</p>	<p>36 1EA</p> <p>RGB LED 모듈</p> <p>RGB LED 모듈로 RGB LED 세개를 하나로 묶은 상품입니다.</p>	<p>45 5EA</p> <p>세라믹 캐패시터 (1uF)</p> <p>세라믹 캐패시터 (1uF)</p>	<p>46 5EA</p> <p>트랜지스터 2N2222</p> <p>트랜지스터 2N2222</p>	<p>47 5EA</p> <p>트랜지스터 BC547</p> <p>트랜지스터 BC547</p>	<p>48 5EA</p> <p>트랜지스터 BC557</p> <p>트랜지스터 BC557</p>
<p>49 2EA</p> <p>전해 캐패시터 (50V 10uF)</p> <p>전해 캐패시터 (50V 10uF)</p>	<p>50 2EA</p> <p>전해 캐패시터 (50V 100uF)</p> <p>전해 캐패시터 (50V 100uF)</p>						