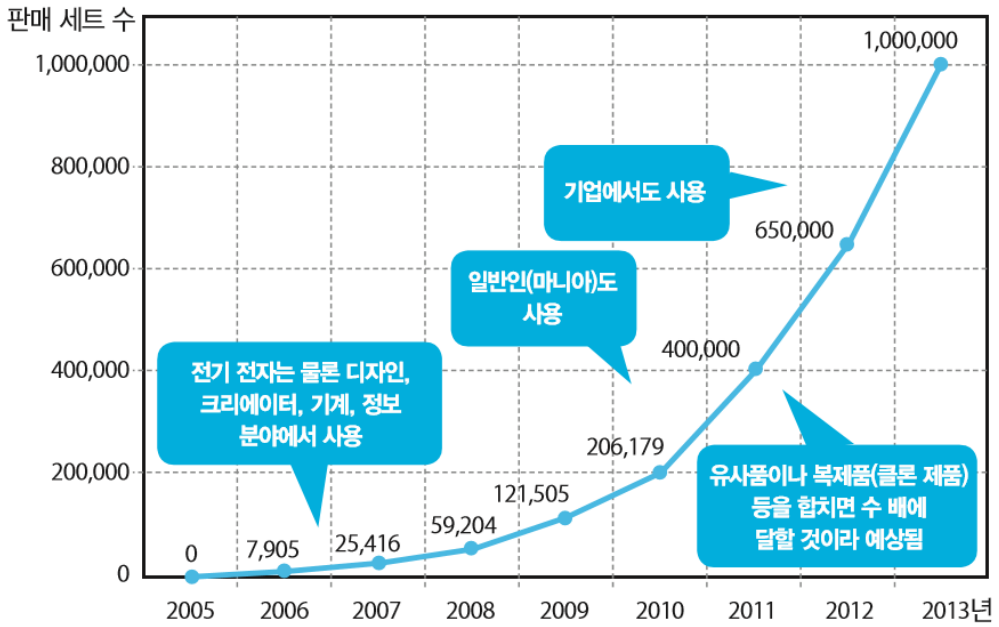
IoT 응용소프트웨어 기획

경남대학교 전하용

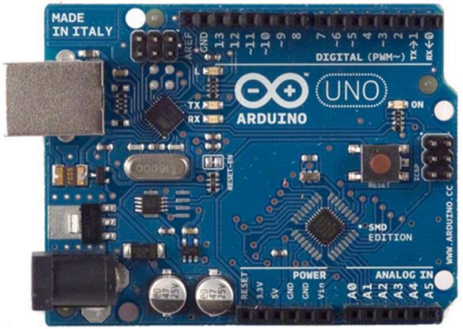
아두이노(Arduino)란?

- 최초의 2005년 이탈리아 이브레아(Ivrea)에서 시작
- 마시모 반지(Massimo Banzi) 교수와 데이비드 쿠아르티에예스(David Cuartielles) 교수는 인터랙션 디자인 전문학교(IDII)에서 공부하는 하드웨어 미숙련자 및 비전공 학도들을 위해 **기초적인 지식만으로도 쉽게 프로그램 작성이 가능**하고, 또한 **저렴하게 구입 가능한 마이크로컨트롤러 보드**를 개발
- 아두이노의 **하드웨어와 소프트웨어가 오픈소스**로 개방
- 수많은 개인과 기업들이 아두이노를 기반으로 다양한 모양과 성능의 아두이노 및 아두이노 호환 보드를 개발

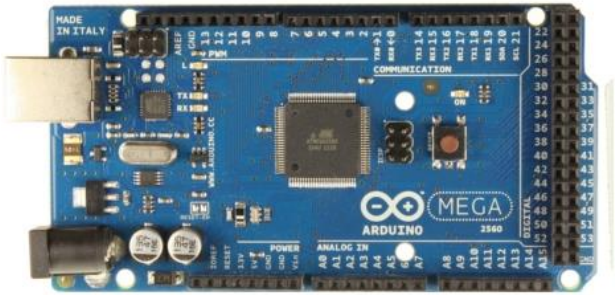
- 아두이노는 기술의 장벽이 매우 낮아서 지금은 정보 처리나 기계, 디자인, 크리에이터 등 문·이과를 막론하고 다양한 학생들이 사용
- 이러한 보급 기세는 세계적으로 퍼져, 학생과 일반인은 물론 여러 기업의 기술자들도 사용



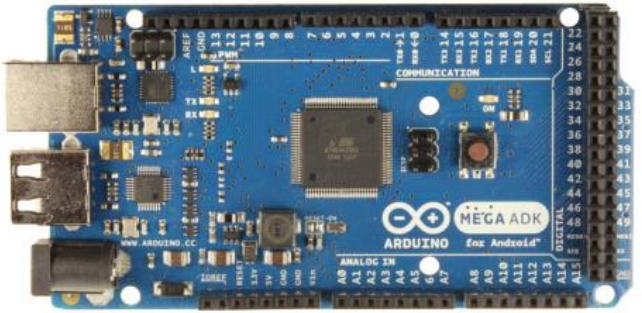
아두이노 종류



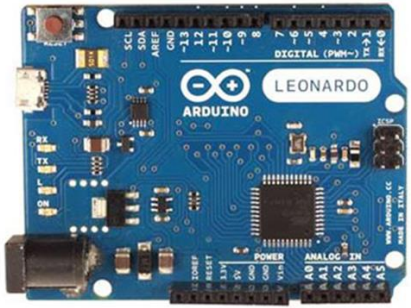
[UNO]



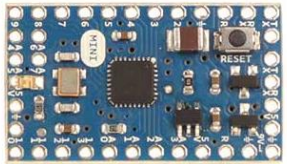
[MEGA]



[MEGA ADK]

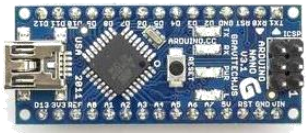


[Leonardo]

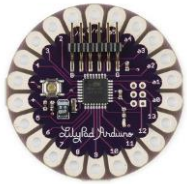


[MINI]

[MICRO]



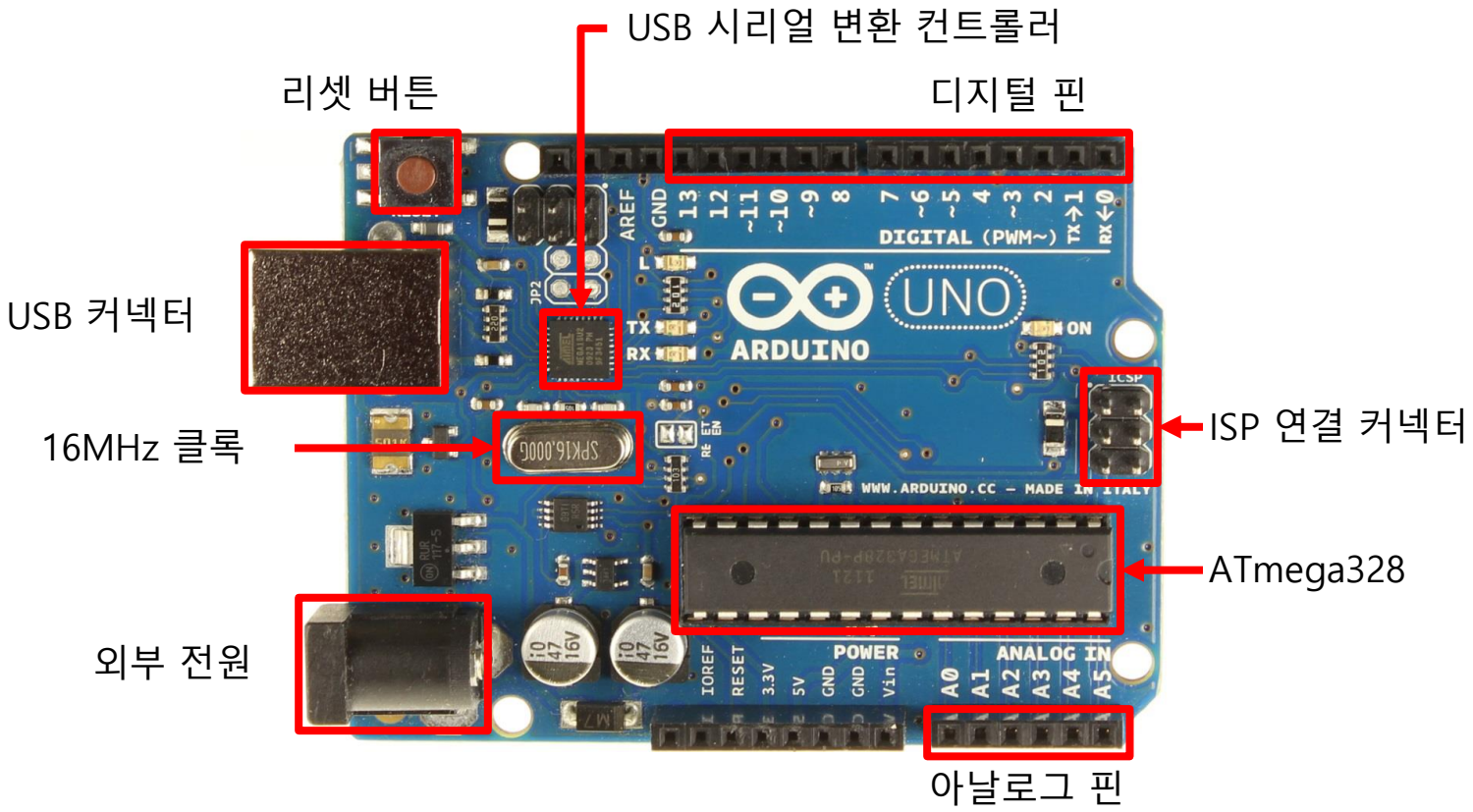
[NANO]



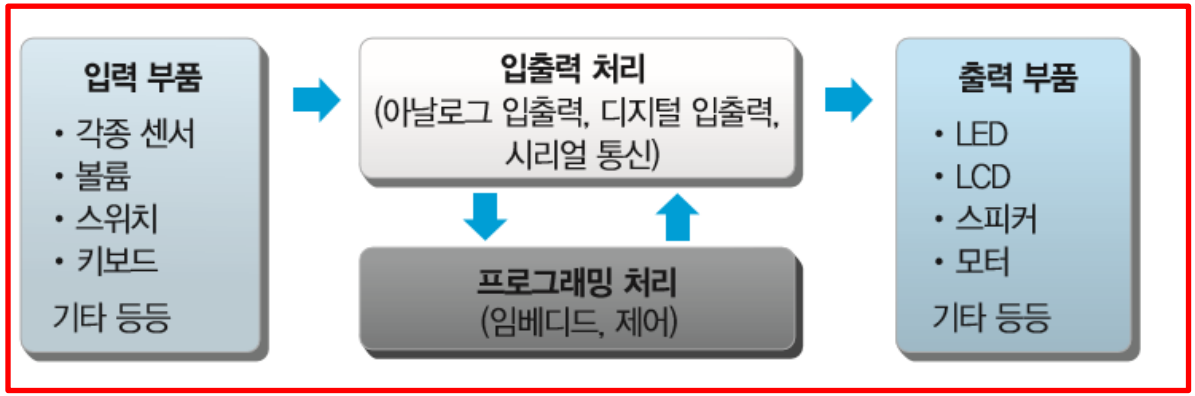
[LilyPad]

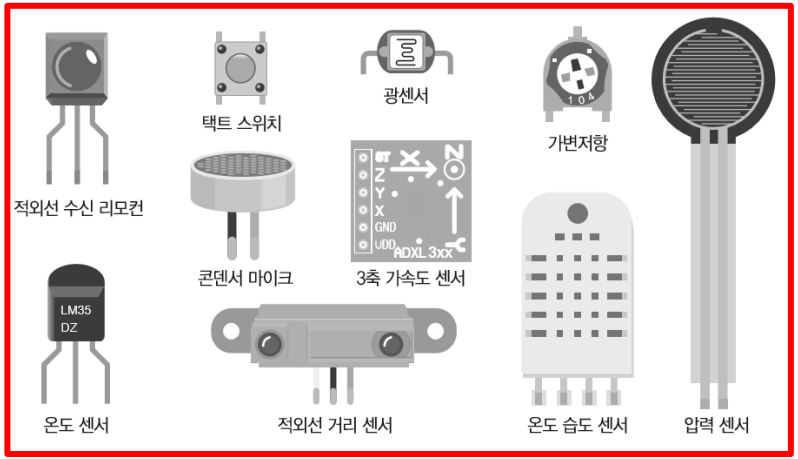
[Esplora]

아두이노 Uno 살펴보기

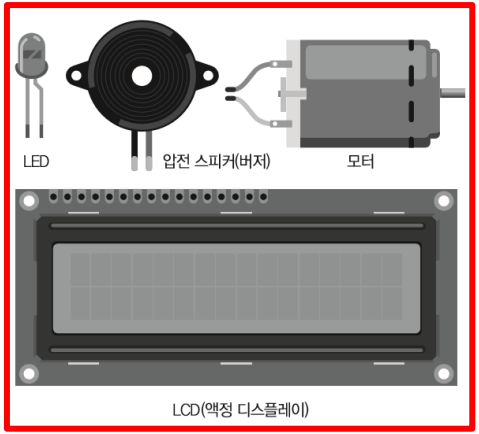


아두이노 할수 있는 일





[입력부품]

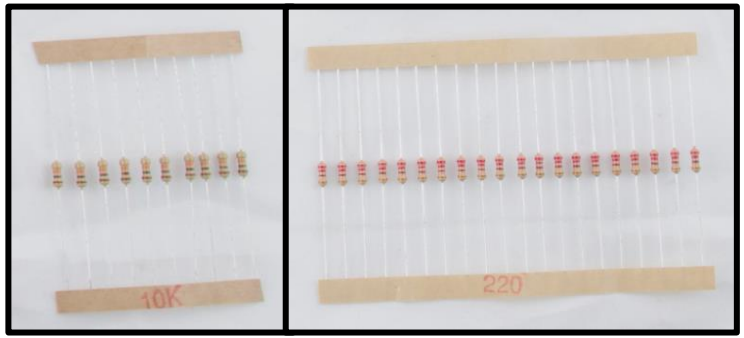


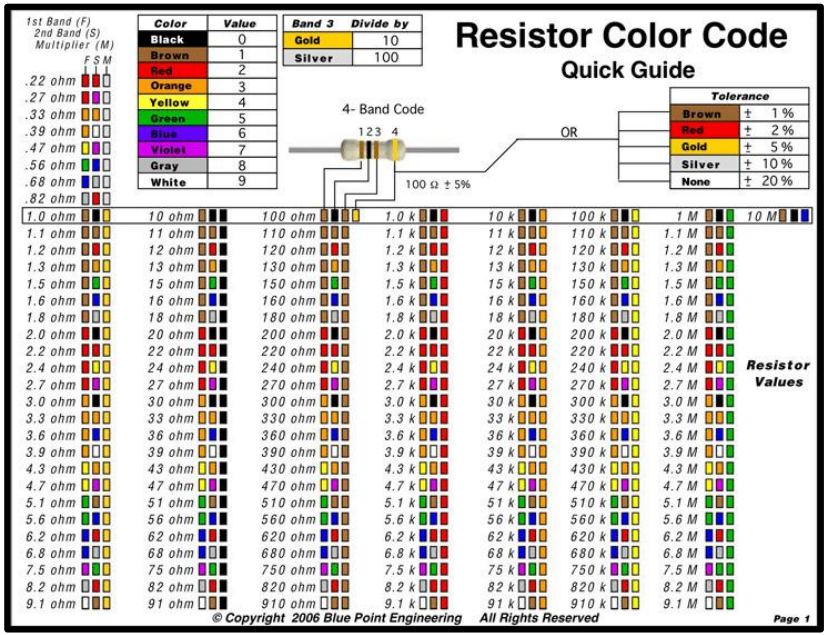
[출력부품]

기본 부품

★ 저항(Resistor)

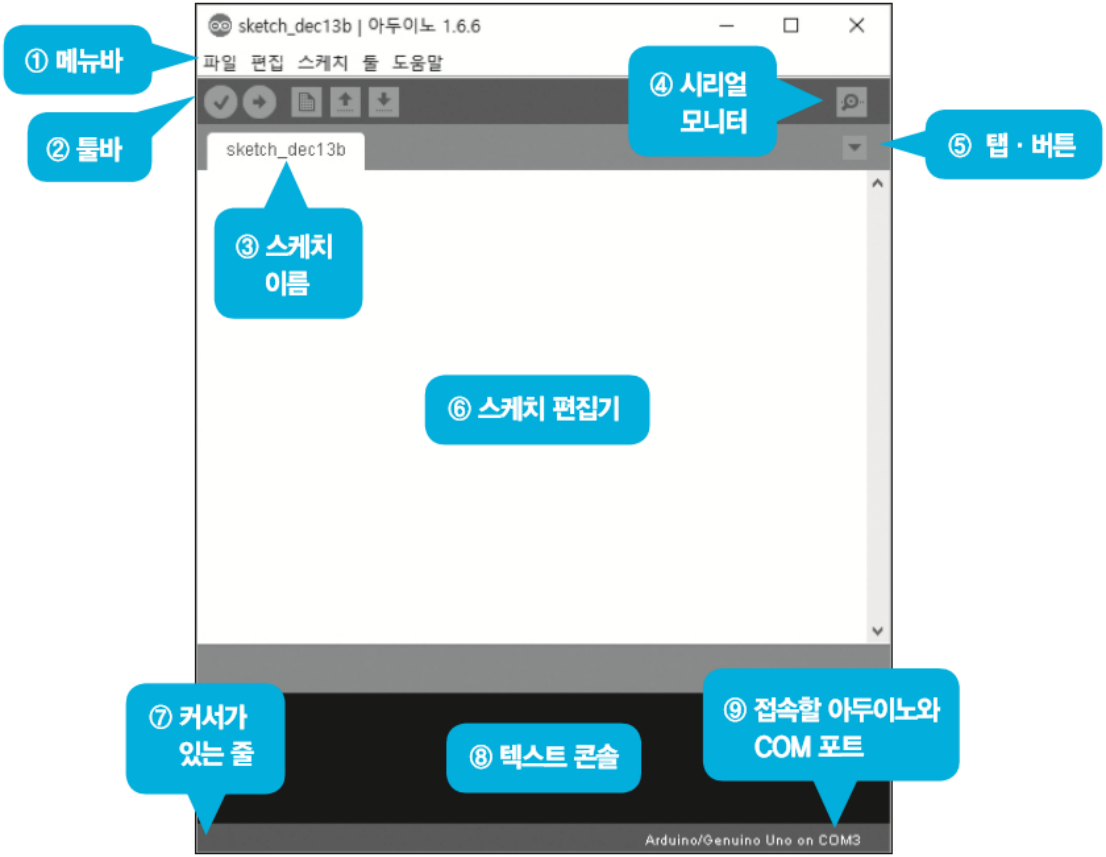
- 전기적에너지를 열에너지로 전환해서 전류를 낮추는 역할.
- 부품이 파손되지 않도록 전류 조절을 목적으로 사용.
- 극성이 없음.
- 띠의 색상 배열로 저항값을 알 수 있음





[4 band]

아두이노C 기반 개발 환경

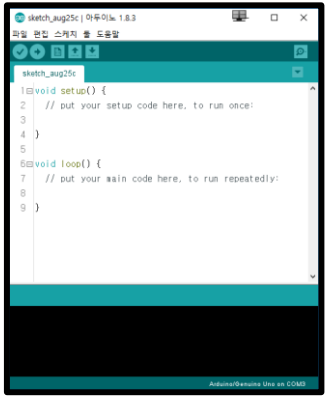


아두이노 개발 환경

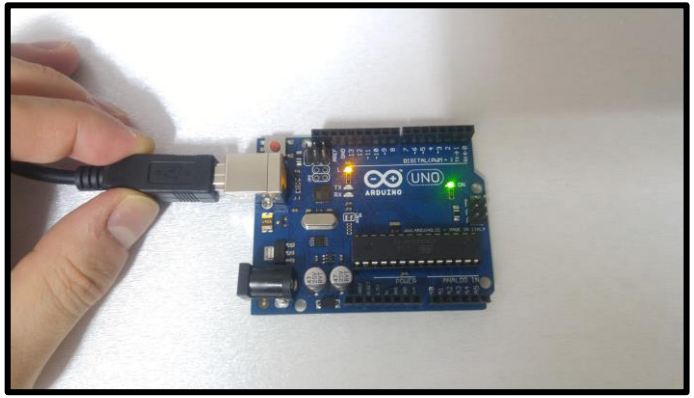
●소프트웨어 다운로드 설치/실행하기

- ✓ 아두이노 소프트웨어를 사용하여 스케치(소스 코드)를 작성하고 아두이노에 업로드 할 수 있습니다.
- ✓ 소프트웨어를 다운받고 사용하는 방법을 따라해보도록 합니다.

1.소프트웨어를 다운로드 후 실행하여 설치합니다.
링크(바로가기):
https://www.arduino.cc/en/Main/Donate

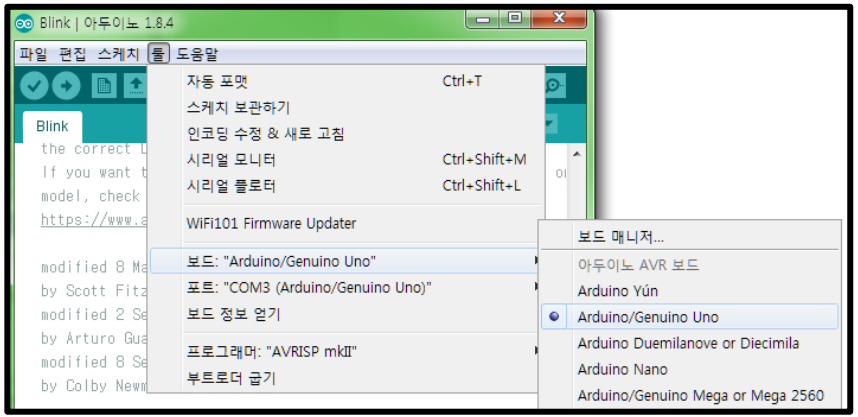


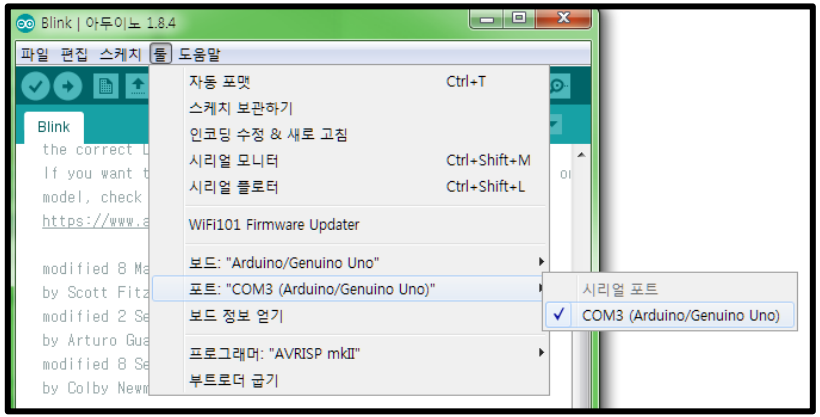
2.프로그램을 설치후 소프트웨어를 실행합니다.



3.아두이노를 usb케이블을 이용하여 pc와 연결합니다

- 아두이노보드 포트 설정하기
  - ✓ 소프트웨어가 아두이노 우노 보드 를 인식할 수 있도록 설정합니다.

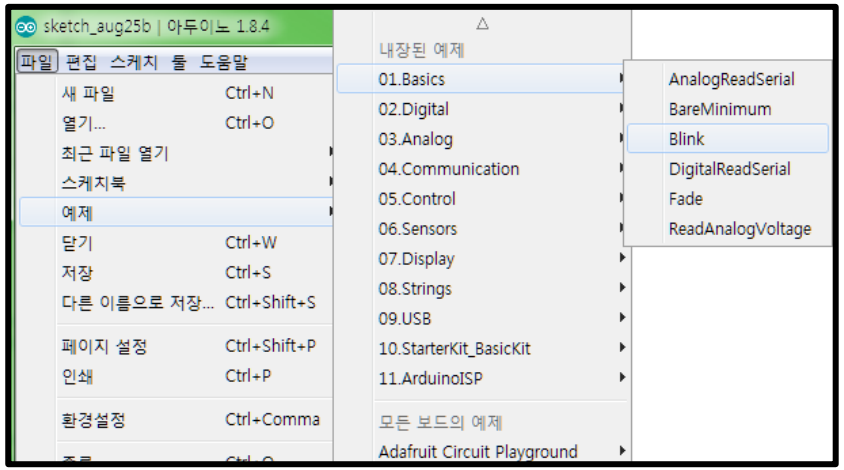




4.`툴->보드`메뉴에서 보드를 `Arduino Uno` 로 설정합니다

5.`툴 -> 포트` 메뉴에서 아두이노가 연결된 포트를 선택합니다.

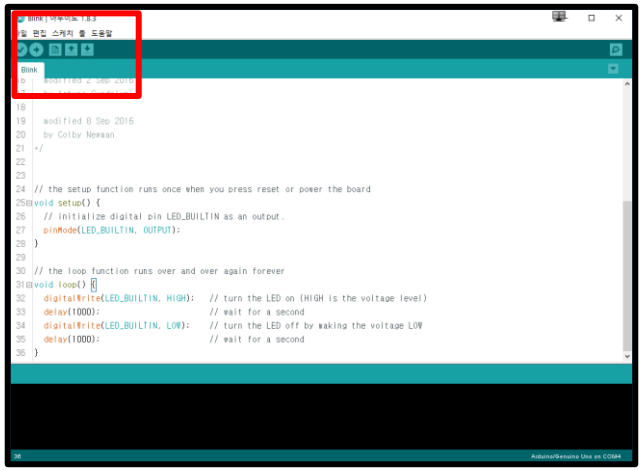
- 예제 스케치 활용하기
  - ✓ 아두이노는 오픈소스로 운영됩니다. 모든 자료가 공개되어 있고 자주 사용하는 기본적인 기능은 아래의 '예제'에서 확인할 수 있습니다.
  - ✓ 예제 스케치를 불러오고 아두이노에 업로드하는 것을 실습해 봅니다.



6.`파일->예제->Blink`메뉴에서 예제 스케치를 불러옵니다.
해당 스케치는 아두이노의 내장 LED를 1초 간격으로 점멸합니다.

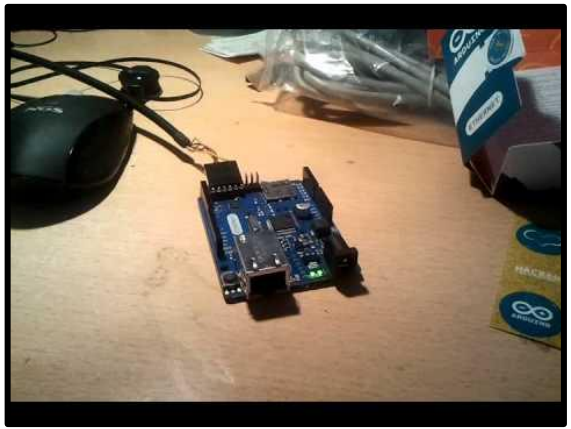
```
void setup()
{
 pinMode(13, OUTPUT);
}
void loop()
{
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

7.코드 수정



8.불러온 스케치를 `업로드`버튼을 클릭하여 아두이노에 업로드 합니다. 자동으로 컴파일되어 업로드됩니다.

- 아두이노 동작확인



9.동작을 확인합니다

- 소스코드 설명

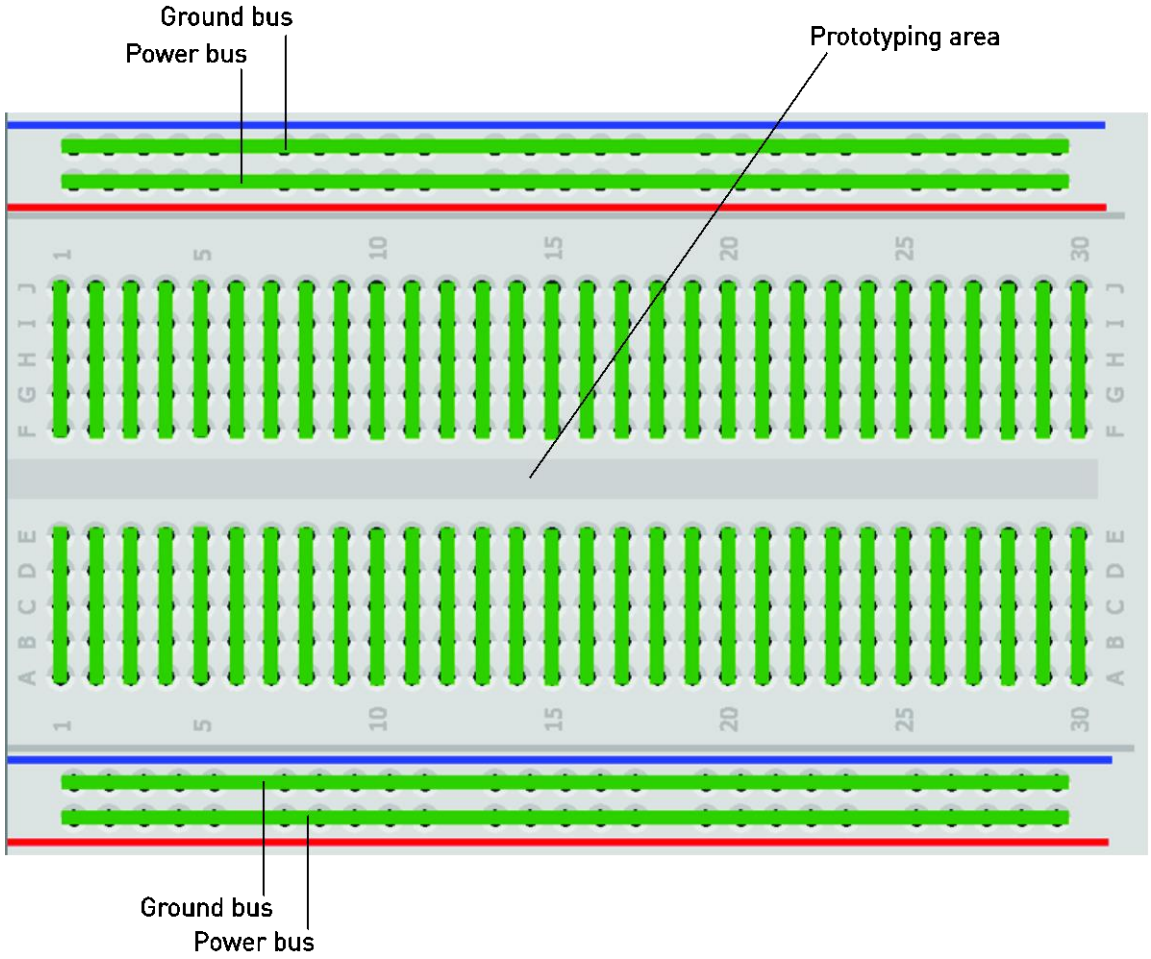
- 파일/예제/0.1Basic/Blink 수정

소스코드 설명

```
void setup()
{
 pinMode(13, OUTPUT);
}
void loop()
{
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

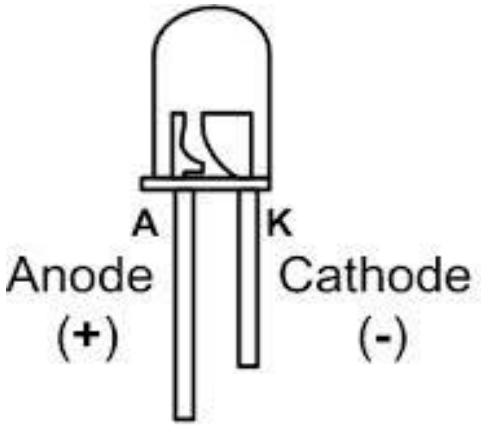
최초 상태 설정()
{
 아두이노 보드의 핀이 어떤 상태인지 설정(내장LED = 13, PIN을 출력모드로 설정);
}
명령이 반복되는 구간 설정()
{
  디지털핀의 출력여부 입력(디지털 핀 번호, HIGH = 1 = 아두이노 보드가 출력을 실행함 = 5V 가압);
  입력이 유지되는 시간 설정(1000 milliseconds = 1초 입력);
  디지털핀의 출력여부 입력(디지털 핀 번호, LOW = 0 = 아두이노 보드가 출력을 실행하지않음);
  입력이 유지되는 시간 설정(1000 milliseconds = 1초 입력);
}

브레드보드 사용법


Ground bus
Power bus
Prototyping area
1
5
10
15
20
25
30
A B C D E
F G H I J
Ground bus
Power bus

LED 제어하기

- LED(Light Emitting Diode)의 약자로 빛을 뿜어내는 반도체란 뜻
- 다른 다이오드와 마찬가지로 애노드에서 캐소드 방향으로만 전류가 흐름
- 따라서 +, - 방향을 고려하여 연결
- 애노드와 캐소드
  - +극은 애노드(anode), -극은 캐소드(cathode)



옴(Ω)의 법칙

- 일반적인 LED는 2V 전압, 20mA 전류를 사용
- 아두이노에는 5V 전압 입력 됨은 모두 사용
- LED가 2V를 사용하여 3V가 남으므로(5V-2V=3V),
- 이 3V의 전압은 저항으로 낮춰야 하므로 옴의 법칙에 이것을 적용하면,

저항값 = (공급전압 - LED동작전압) / LED

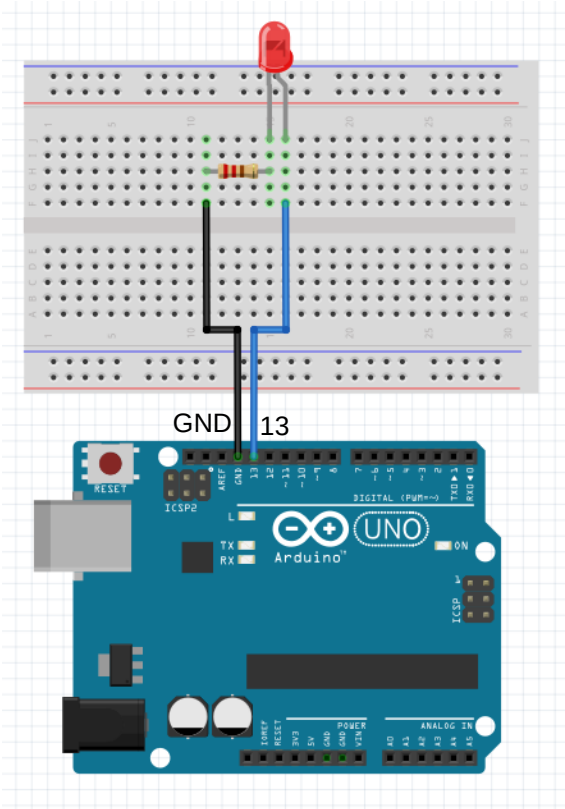
$$V = IR$$

$$R = V/I$$

$$R = 3V / 0.02A = 3 / 0.02 = 150\ \Omega$$

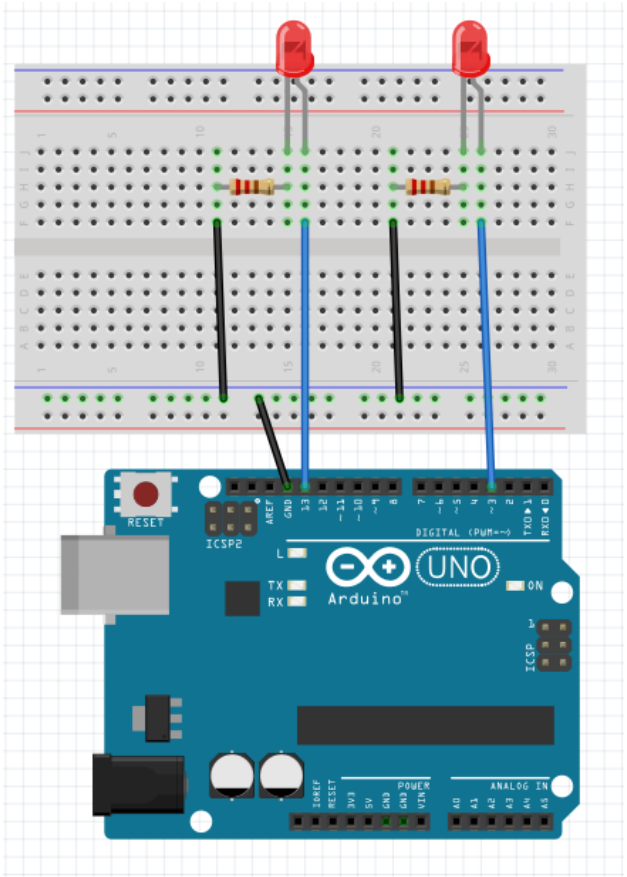
디지털 출력

- 마이크로컨트롤러에서 외부로 데이터를 내 보내는데, 결국은 마이크로 컨트롤러의 특정 핀의 전압을 HIGH, LOW로 설정하는 것



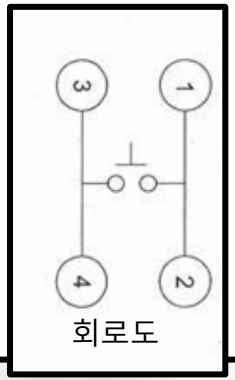

```
void setup() {
  pinMode(13, OUTPUT);
}
void loop() {
  digtalWrite(13, HIGH);
  delay(1000);
  digtalWrite(13, LOW);
  delay(1000);
}
```

LED 2개 연결하기



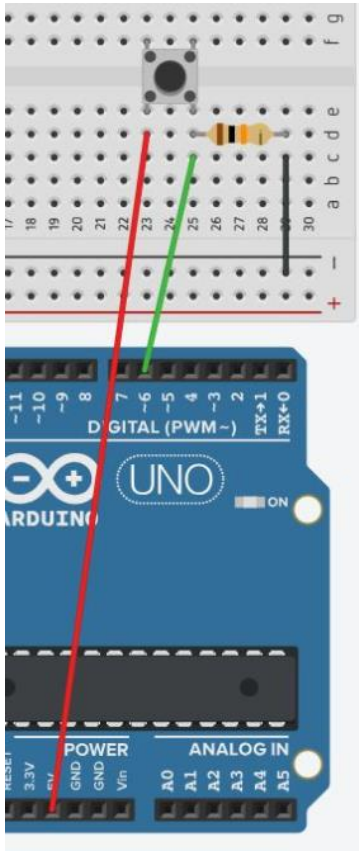

```
int led1=3;
int led2=13;

void setup() {

  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
}

void loop() {
  digtalWrite(led1, HIGH);
  digtalWrite(led2, HIGH);
  delay(500);
  digtalWrite(led1, LOW);
  digtalWrite(led2, LOW);
  delay(500);
}
```

버튼을 이용한 LED 제어



회로도

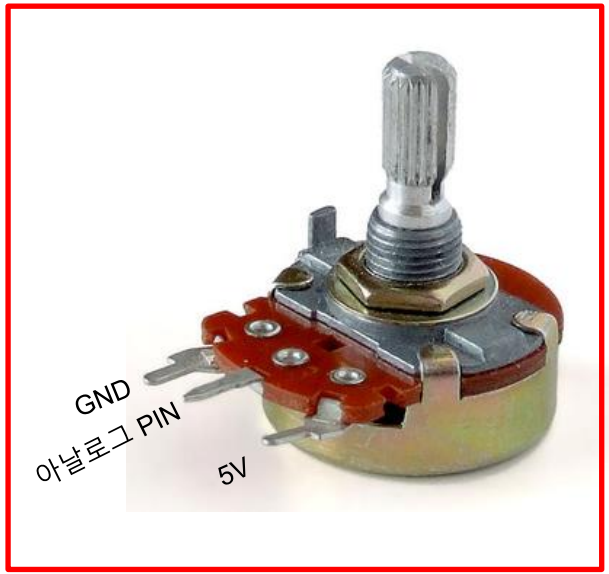
10kΩ 저항 필요




```
int led1=13;
int key1=6;

void setup() {
  pinMode(led1, OUTPUT);
  pinMode(key1, INPUT);
}

void loop() {
  if(digitalRead(key1)==HIGH)
  {
       digitalWrite(led1, HIGH);
   }
  else
  {
     digitalWrite(led1, LOW);
  }
}
```

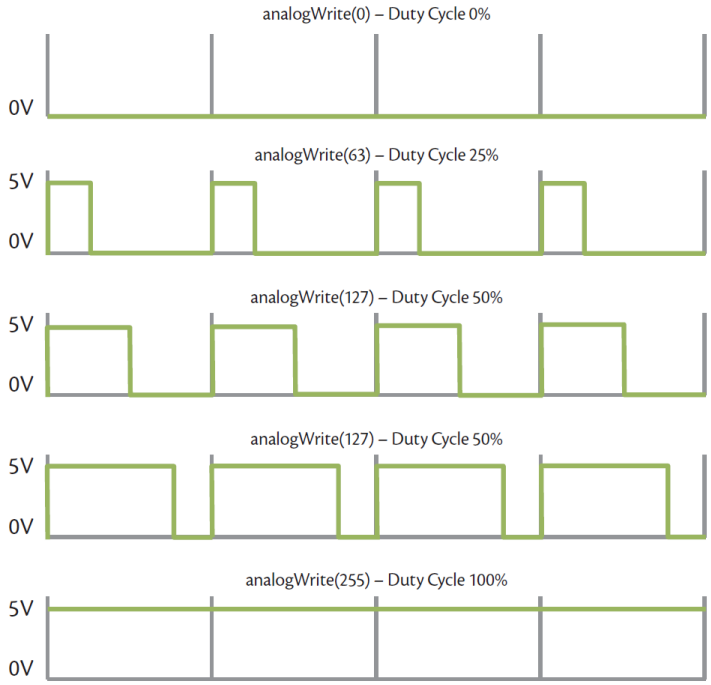
아날로그 입력 -가변저항

- 전자회로에서 저항을 임의로 바꿀 수 있는 저항
- 저항을 바꾸게 되면 전류의 크기도 바뀌게 됨
- 전압을 가하면 설정에 따라 전압 일부를 전달하는 역할
- 주로 오디오 장비나 동작 감지기 같은 세서에서 감도, 밸런스, 입력, 출력 등을 조절하는데 사용



펄스(PWM)로 LED 밝기 조절하기(P68)

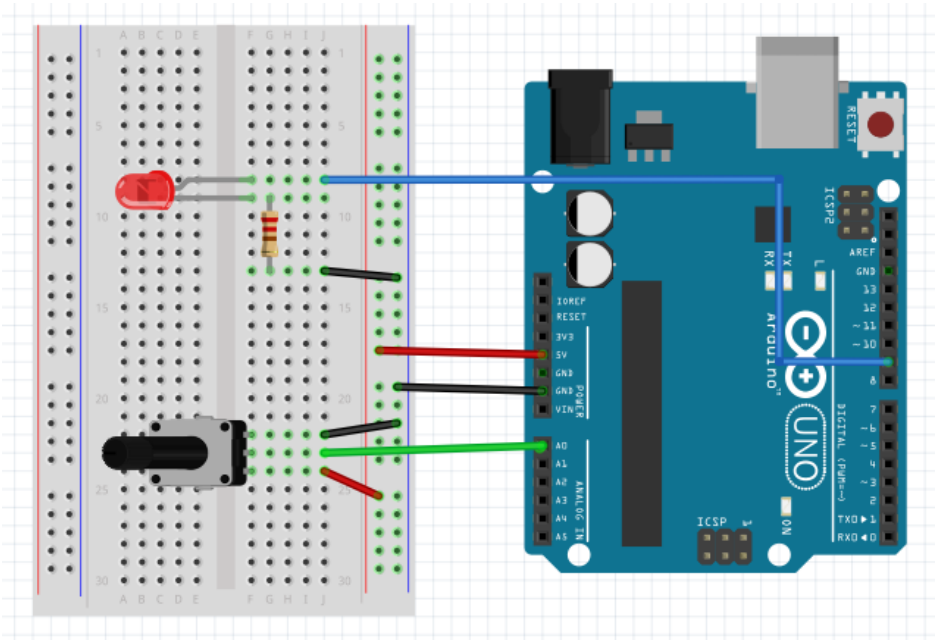
- PWM(Pulse Width Modulation: 펄스폭 변조)
- 펄스 폭을 가변시킬 수 있는 파형
- 파형 주기 중 On과 Off 비율(듀티비)를 조절함으로써 평균 전압을 조절할 수 있음.
- 듀티비(duty ratio) : 파형 주기에서 On되는 비율
- 펄스 폭을 가변함으로써 아날로그 출력의 진폭을 조절하는 효과

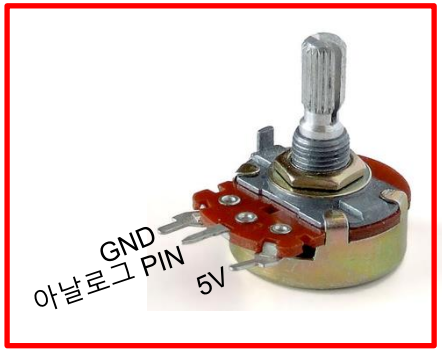


- 파일메뉴/예제/0.1 Basics/Fade

```
int led = 9;
int brightness = 0;
int fadeAmount = 5;

void setup()  {
   pinMode(led, OUTPUT);
}

void loop()
{
  analogWrite(led, brightness);
  brightness = brightness + fadeAmount;

  if (brightness == 0 || brightness == 255) {
     fadeAmount = -fadeAmount ;
  }
  delay(30);
}
```

가변저항으로 LED 제어하기

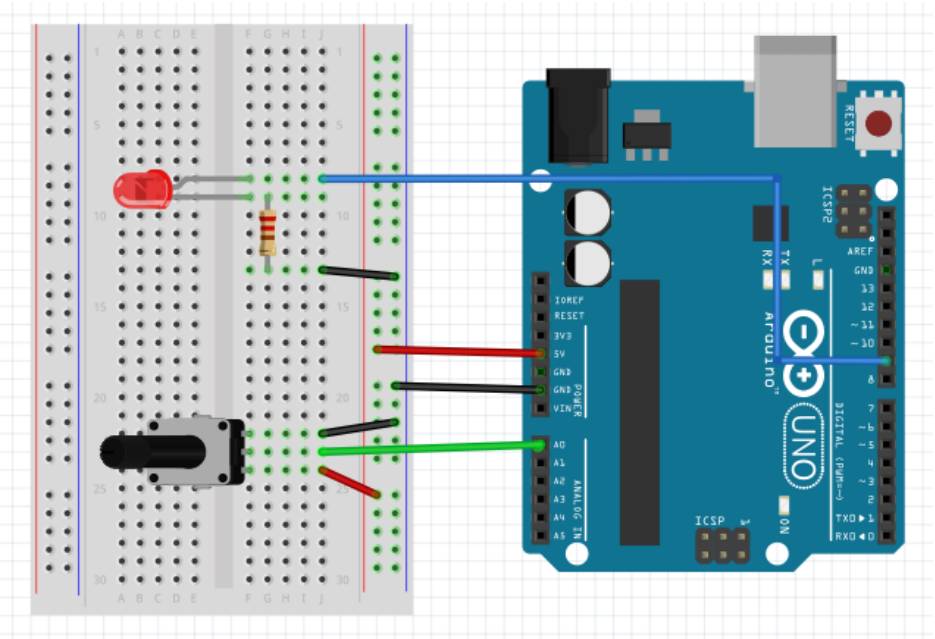




- 파일/예제/0.1basic/AnalogReadSerial

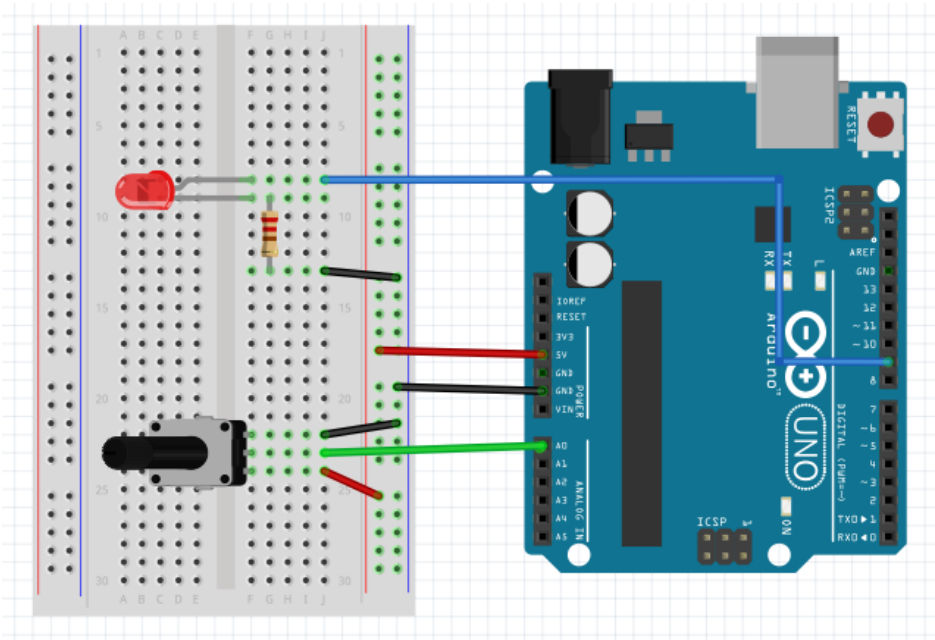
```
void setup()
{
   Serial.begin(9600);
}

void loop()
{
   int value = analogRead(A0);

   Serial.println(value);

   delay(10);
}
```

가변저항으로 LED 제어하기



```
void setup()
{
 pinMode(9, OUTPUT);
 Serial.begin(9600);
}

void loop()
{

  int value = analogRead(A0);

  analogWrite(9, value);
  Serial.println(value);

  delay(10);
}
```

가변저항으로 LED 제어하기(고급)



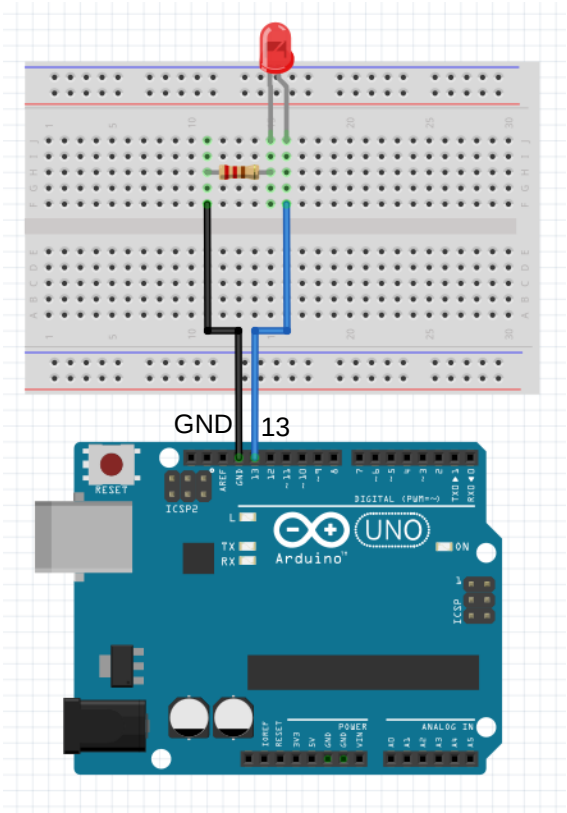
```
 void setup()
 {
  pinMode(9, OUTPUT);
  Serial.begin(9600);
 }

void loop()
{

  int value = map(analogRead(A0), 0, 1023,0,255);

  analogWrite(9, value);
  Serial.println(value);

  delay(10);
}
```

시리얼(RS232) 통신

- 아두이노와 외부기기(컴퓨터)가 서로 대화할 수 있게 하는 기능
- 아두이노 IDE로부터 데이터를 수신하는 작업
  - ✓ 아두이노로부터 전송된 데이터는 버퍼에 자동 저장

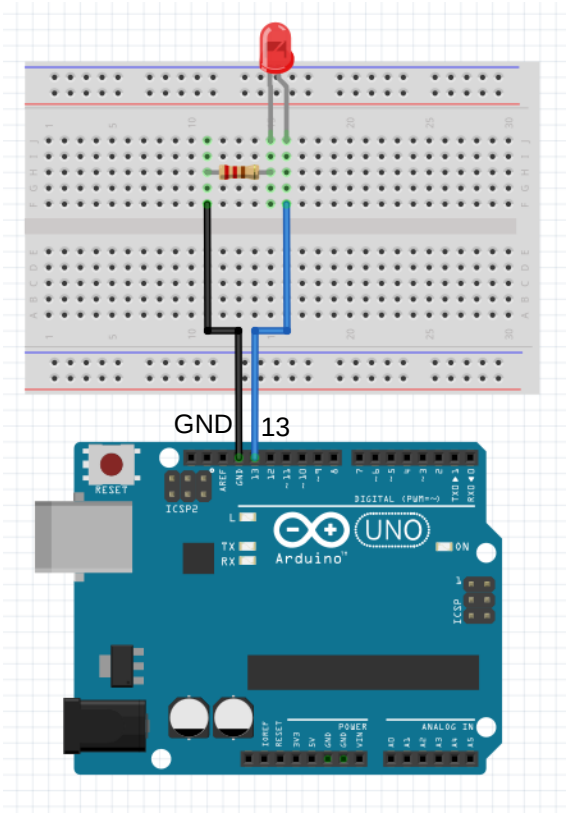
```
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  char read_data;
  if (Serial.available())
  {
    read_data = Serial.read();
    Serial.print(read_data);
  }
  delay(10);
}
```

시리얼통신으로 LED 제어하기




```
void setup()
{
   Serial.begin(9600);
}
void loop()
{
  char read_data;
  if (Serial.available())
  {
     read_data = Serial.read();
     if( read_data == '1')
     {
        Serial.println("LED1 ON");
     }
     else if( read_data == '0')
     {
        Serial.println("LED1 OFF");
     }
  }
  delay(10);
}
```

시리얼통신으로 LED 제어하기




```
int led1 = 13;
void setup()
{
  pinMode(led1, OUTPUT);
  digitalWrite(led1, LOW);
  Serial.begin(9600);
}
void loop()
{
  char read_data;
  if (Serial.available())
  {
    read_data = Serial.read();
    if( read_data == '1')
    {
      digitalWrite(led1, HIGH);
      Serial.println("LED1 ON");
    }
    else if( read_data == '0')
    {
      digitalWrite(led1, LOW);
      Serial.println("LED1 OFF");
    }
  }
  delay(10);
}
```

부저

액티브 부저 (active Buzzer)

액티브 부저(능동부저), 피에조 부저

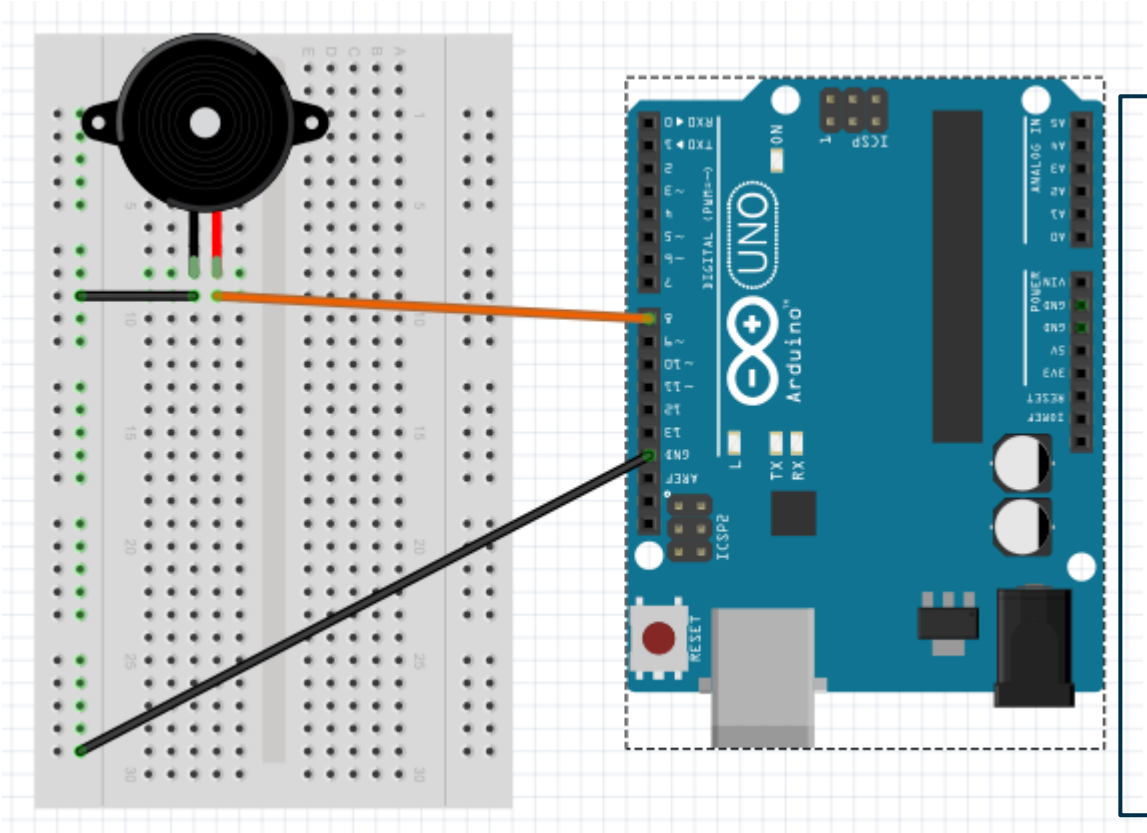
- 내장된 회로가 있어 외부 전원만 인가하면 소리 출력
- 단일음만 가능
- 보통 양극과 음극이 스티커로 표시가 되어 있음

패시브 부저 (Passive Buzzer)

패시브 부저(수동부저)

- 내장된 회로 없음
- 주파수 파형(tone 함수)으로 제어

능동 부저 동작



```
int buzzer = 8;
void setup()
{
  pinMode(buzzer, OUTPUT);
}
void loop()
{
  digitalWrite(buzzer, HIGH);
  delay(1000);
 digitalWrite(buzzer, LOW);
  delay(1000);
}
```

수동 부저 동작

```
int buzzer = 8;
int numTones = 8;
int tones[] = {261, 294, 330, 349, 392, 440, 494, 523};
void setup()
{
}
void loop()
{
  for (int i = 0; i < numTones; i++)
  {
    tone(buzzer, tones[i]);
    delay(500);
  }
  noTone(buzzer);

  delay(1000);
}
```

(단위 : Hz)

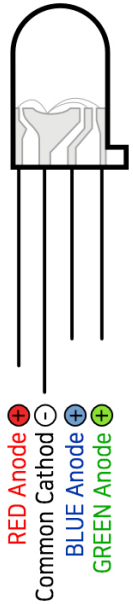
옥타브 / 음계	1	2	3	4	5	6	7	8
C(도)	32.7032	65.4064	130.8128	261.6256	523.2511	1046.502	2093.005	4186.009
C#	34.6478	69.2957	138.5913	277.1826	554.3653	1108.731	2217.461	4434.922
D(레)	36.7081	73.4162	146.8324	293.6648	587.3295	1174.659	2349.318	4698.636
D#	38.8909	77.7817	155.5635	311.1270	622.2540	1244.508	2489.016	4978.032
E(미)	41.2034	82.4069	164.8138	329.6276	659.2551	1318.510	2637.020	5274.041
F(파)	43.6535	87.3071	174.6141	349.2282	698.4565	1396.913	2793.826	5587.652
F#	46.2493	92.4986	184.9972	369.9944	739.9888	1479.978	2959.955	5919.911
G(솔)	48.9994	97.9989	195.9977	391.9954	783.9909	1567.982	3135.963	6271.927
G#	51.9130	103.8262	207.6523	415.3047	830.6094	1661.219	3322.438	6644.875
A(라)	55.0000	110.0000	220.0000	440.0000	880.0000	1760.000	3520.000	7040.000
A#	58.2705	116.5409	233.0819	466.1638	932.3275	1864.655	3729.310	7458.620
B(시)	61.7354	123.4708	246.9417	493.8833	987.7666	1975.533	3951.066	7902.133

수동 부저 동작(학교종이 땡땡땡)

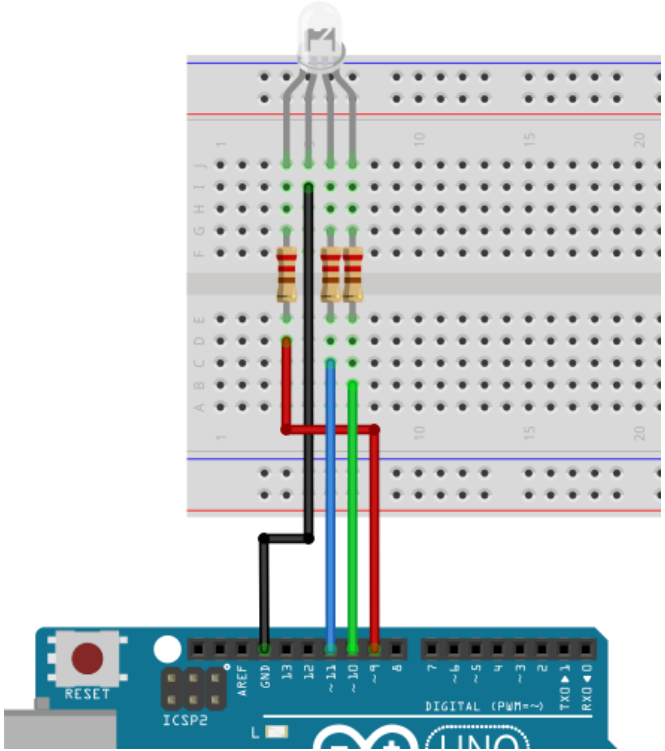
- 파일/예제/0.2Digital/toneMelody 수정

```
int buzzer = 8;
int noteDuration =4;
int melody[] = {261, 294, 330, 349, 392, 440, 494, 523};

int noteDurations[]={4,4,5,5,4,4,2,  4,4,2,2,1,  4,4,5,5,4,4,2,  4,2,1,2,0};

void setup(){
 for (int thisNote =0 ; thisNote <sizeof(noteDurations)/sizeof(int) ; thisNote++){
    int noteLength = 1000/ noteDuration;
    tone(buzzer, melody[noteDurations[thisNote]], noteLength);
    int pauseBetweenNotes = noteLength * 1.3;
    delay(pauseBetweenNotes);
    noTone(buzzer);
  }
}

void loop(){
}
```

RGB LED(Cathod Type)



R - D9
(-) - GND
B - D11
G - D10




```
int R = 9 ;
int G = 10 ;
int B = 11 ;

void setup()
{
 pinMode(R,OUTPUT);
 pinMode(G,OUTPUT);
 pinMode(B,OUTPUT);
}

void loop()
{
 analogWrite(R,255);
 analogWrite(G,0);
 analogWrite(B,0);
 delay(1000);

 analogWrite(R,0);
 analogWrite(G,255);
 analogWrite(B,0);
 delay(1000);

 analogWrite(R,0);
 analogWrite(G,0);
 analogWrite(B,255);
 delay(1000);
}
```

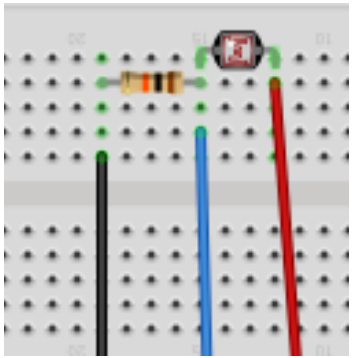
CDS 조도센서

- 빛의 양에 따라 특성이 변화하는 소자로는
  포토레지스터(photo resistor), 포토다이오드(photo diode),
  포토트랜지스터(photo transistor) 등이 있음

- 빛의 양에 따라 저항 값이 변하는 포토레지스터, 일반적으로
  광센서라 불림

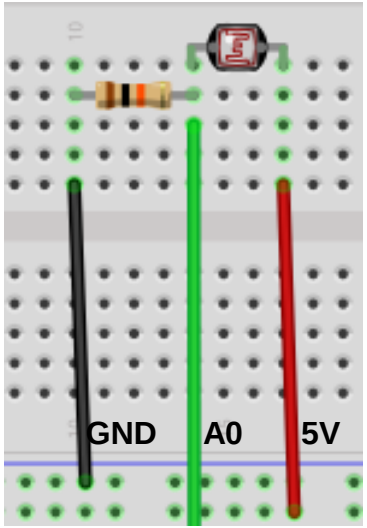
- 광센서는 조도센서, 광전도셀, 포토셀 등으로도 불리며, CdS
  광센서는 카드뮴(Cd)과 황(S)을 화합하여 만들어지는
  황화카드뮴 결정에 금속 다리를 결합하여 만들어짐

  10kΩ 저항 필요



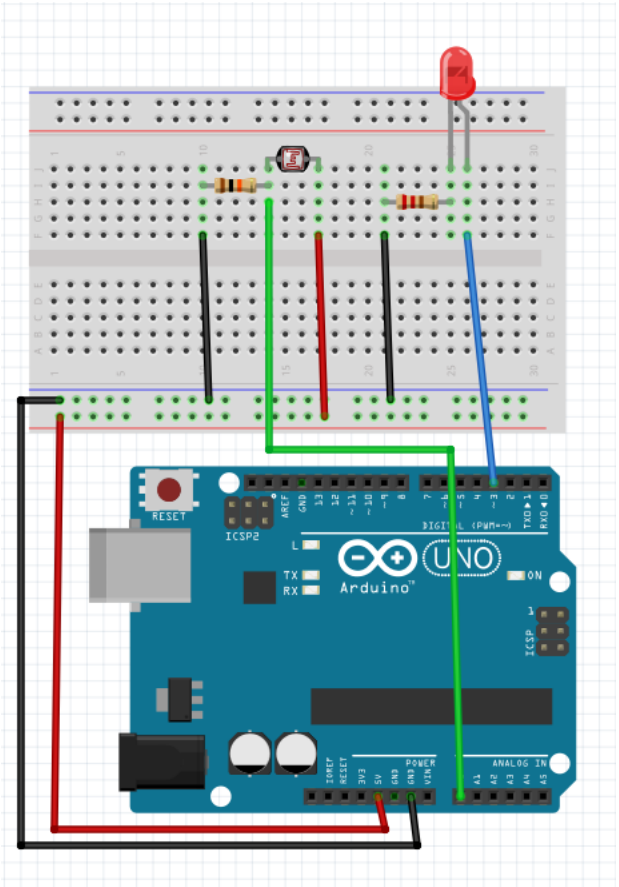
GND 아날로그 5V
PIN

CDS로 LED 제어하기




```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
   int value = analogRead(A0);
   Serial.println(value);
}
```

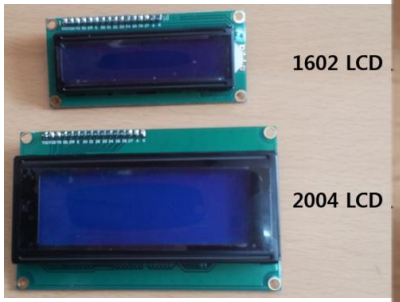
CDS로 LED 제어하기

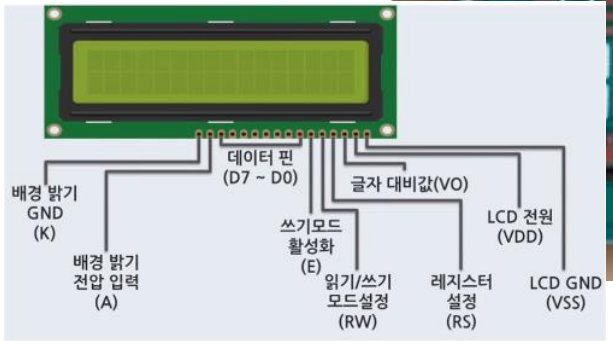



```
void setup()
{
    Serial.begin(9600);
    pinMode( 3, OUTPUT );
}
void loop()
{
   int value = analogRead(A0);
   Serial.println(value);
   if(value > 500)
   {
      digitalWrite(3, HIGH);
   }
   else
   {
       digitalWrite(3,  LOW);
  }
}
```

LCD 사용하기

- LCD(liquid crystal display)
- 라이브러리 필요





- 파일/예제/LiquidCrystal-I2C-library-master/HelloWorld

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x3F, 16, 2);

void setup()
{
  lcd.begin();
  lcd.backlight();
  lcd.clear();
  lcd.print("Hello, world!");
}
void loop()
{
  lcd.setCursor(5,1);
  lcd.print("ABC");
}
```

- LCD 기능

Function	Description
begin(int column, int row)	LCD 화면의 열, 행의 개수를 지정
clear()	LCD 화면을 클리어 한다.
home()	문자 표시위치를 왼쪽 상단으로 지정
setCursor(int column, int row)	문자 표시위치를 열과 행으로 지정
write()	현재 커서 위치에 한 문자를 표시
print()	현재 커서 위치에 문자열을 표시
cursor()	현재 커서 위치에 커서('_') 모양 표시
noCursor()	커서 문자를 숨긴다
blink()	커서 모양을 깜빡이도록 한다
noBlink()	커서 Blinking disable
display()	LCD Display On
noDisplay()	LCD Display Off
scrollDisplayLeft()	왼쪽으로 한문자 스크롤
scrollDisplayRight()	오른쪽으로 한문자 스크롤
autoScroll()	문자를 LCD 디스플레이 범위를 넘어서면 왼쪽에서 오른쪽으로 자동 스크롤
noAutoScroll()	문자를 LCD 디스플레이 범위를 넘어도 자동으로 스크롤 하지 않음
leftToRight()	문자 표시 방향을 왼쪽에서 오른쪽으로 표시
rightToLeft()	문자 표시 방향을 오른쪽에서 왼쪽으로 표시

●글자 이동

```
while(1) {
   lcd.scrollDisplayLeft();
   delay(500);
  }
```

●글자 깜빡이기

```
lcd.noDisplay();
 delay(1000);
lcd.display();
 delay(1000);
```

실습

- LCD에 CDS 센서 측정 값 나타내기

실습

1. D13번에 연결된 LED가 CDS 센서 측정 값을 기준으로 ON/OFF

2. ON/OFF 결과가 LCD에 디스플레이 되어야 함.

   ("CDS ON" or "CDS OFF")

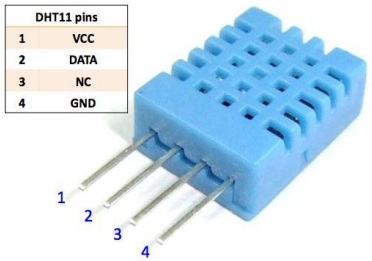
온·습도 센서(DHT11)

- 라이브러리 필요

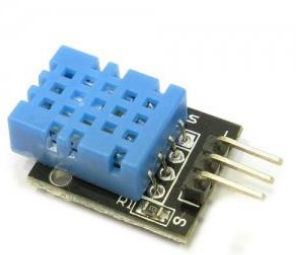
●스케치메뉴/라이브러리 포함하기/라이브러리 관리/dht11 검색

●SimpleDHT 추가

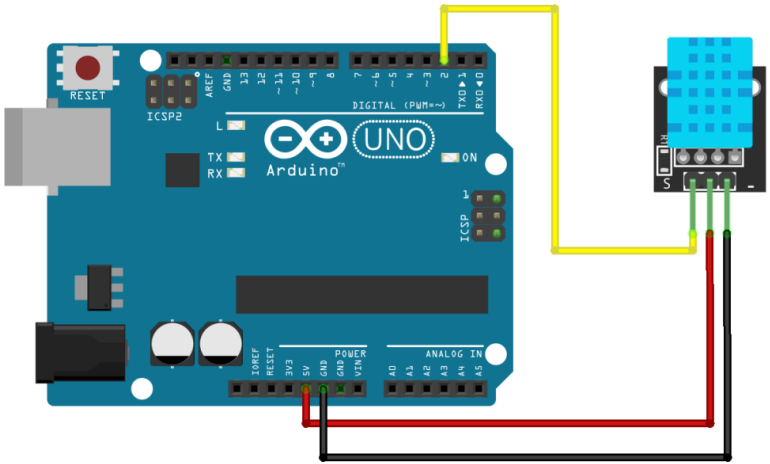
측정 범위	20 ~ 90% RH(상대습도), 0 ~ 50℃
습도 정확도	±5%RH
온도 정확도	±2℃



DHT22



DHT11



- 파일메뉴/예제/SimDHT/DHTDefault 정리

```
#include <SimpleDHT.h>

int pinDHT11 = 2;
SimpleDHT11 dht11(pinDHT11);

void setup() {
  Serial.begin(9600);
}

void loop() {
  byte temperature = 0;
  byte humidity = 0;
  int err = SimpleDHTErrSuccess;
  dht11.read(&temperature, &humidity, NULL);
  Serial.print("Sample OK: ");
  Serial.print((int)temperature);
  Serial.print(" *C, ");
  Serial.print((int)humidity);
  Serial.println(" H");

  delay(1500);
}
```

실습

- 온도가 2도 이상이면 LED가 켜지는 동작을 위해 조립 및 코딩 하기

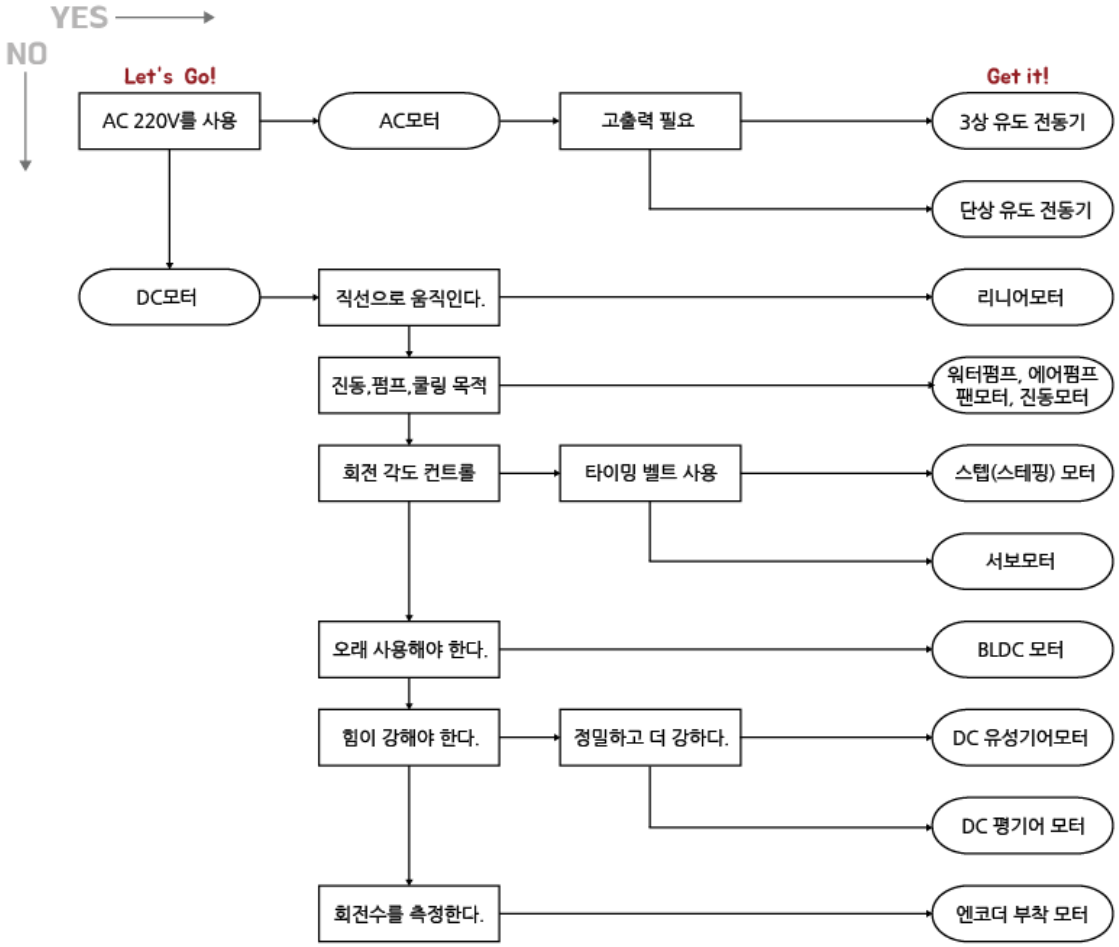
모터

<DC모터>

<서보모터>

<스텝모터>

<기어박스모터>



• 모터 기본 용어 정리

작동 전압 (예: 3~6V)
모터가 작동하는 최적의 전압. 더 작은 전압을 입력해도 동작함 (큰 전압은 모터의 고장 원인임)

적정 전압 (예: 6V)
모터의 성능을 최대한 활용하면서 모터에 부담을 주지 않는 전압. 또한 적정 전압을 기준으로 속도 및 힘 등을 표시

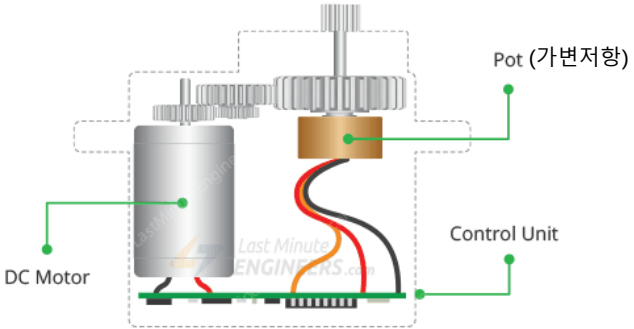
전류 (예: 200mA)
모터가 사용하는 전류(에너지). 모터의 전류보다 더 큰 전원을 입력해야 함 (1A = 1000mA)

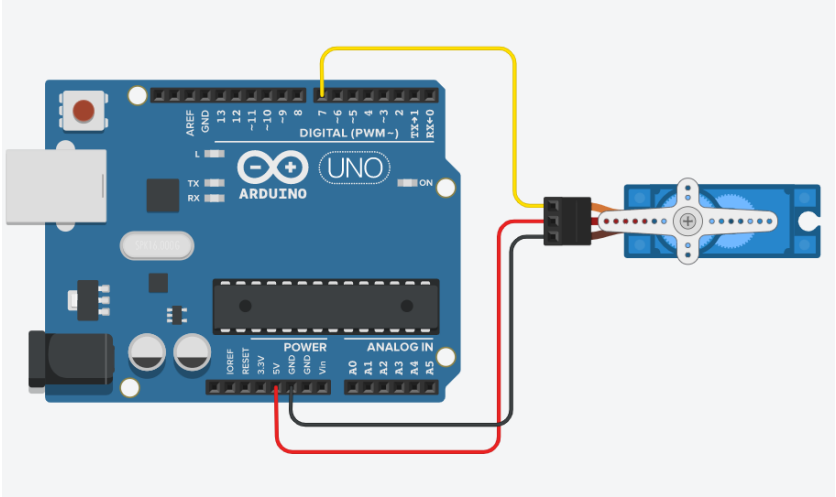
RPM (예: 300RPM)
Revolutions Per Minute의 약자이며, 분당 회전수. 300RPM하면 1분에 300바퀴 회전

토크 (예: 1kg/cm)
회전력(힘). 1kg/cm라하면 모터의 축(돌아가는 부분)을 중심으로 1cm 떨어진 점에서 1kg의 힘을 버틸수 있다는 의미

서보모터(SG90)

- 로봇 관절, 차량의 방향타 등에 사용
- 0~180도 사이의 각도 제어
- 내부에 DC모터와 저항, 모터 드라이버로

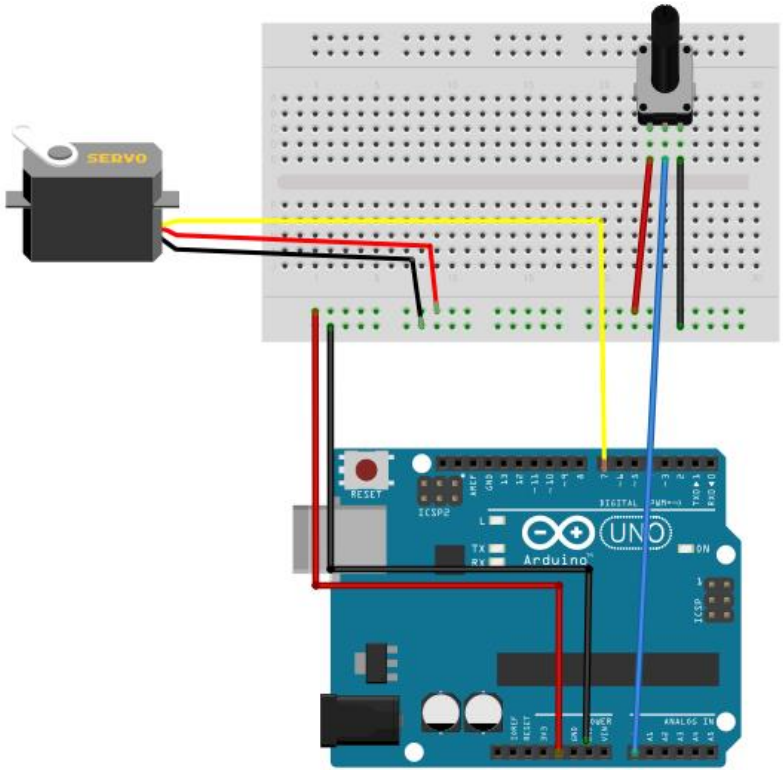





```ino
#include <Servo.h>

int servoPin= 7;

Servo servo;

void setup() {
    servo.attach(servoPin);
}

void loop() {
  for (int i=0; i<180; i++){
    servo.write(i);
    delay(100);
  }
}
```

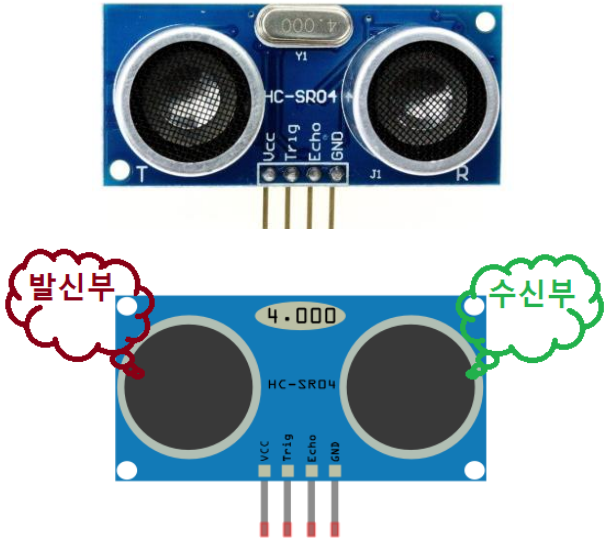
- 파일메뉴/예제/Servo/Knob

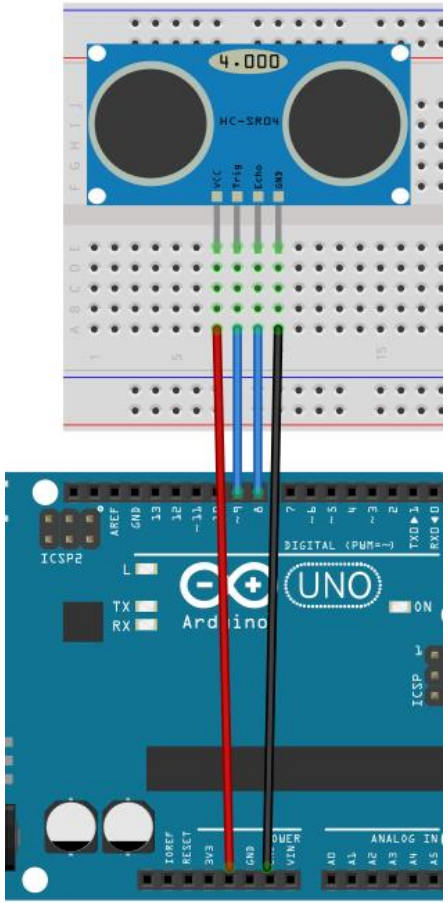
```
#include <Servo.h>

Servo myservo;

int potpin = A0;
int val;

void setup() {
  myservo.attach(7);
}

void loop() {
  val = analogRead(potpin);
  val = map(val, 0, 1023, 0, 180);
  myservo.write(val);
  delay(15);
}
```



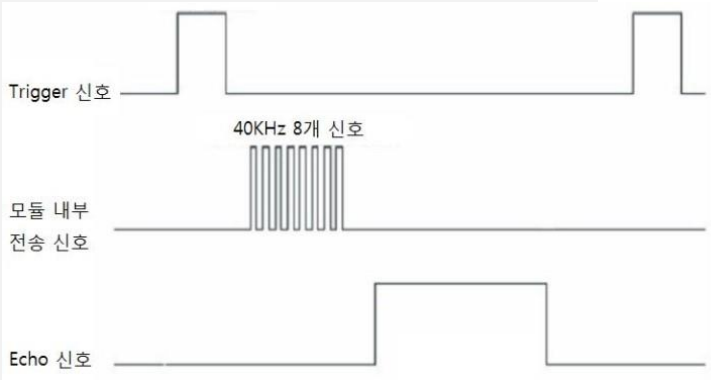
초음파센서(UltraSonic)

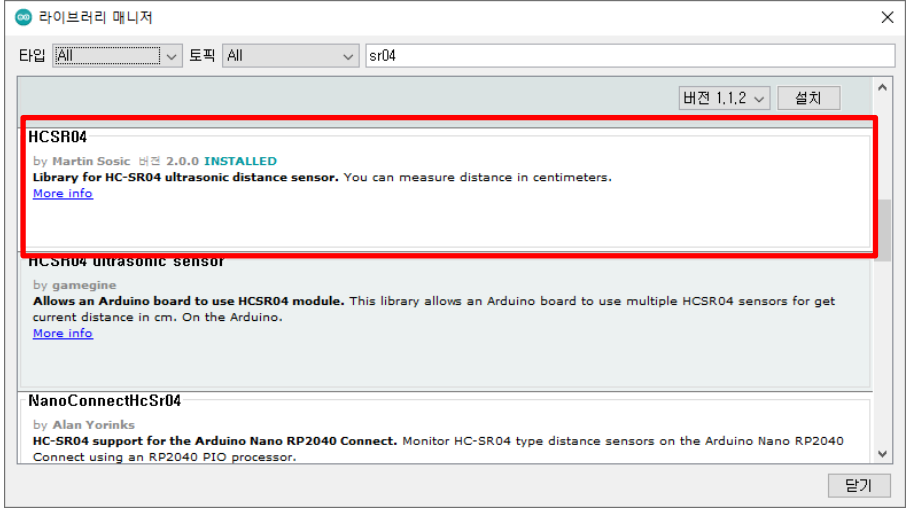
- 약 20kHz이상의 높음 주파수의 음파가 물체에 부딪힌 다음 반사파가 되어 돌아오는 시간을 측정하여 물체까지의 거리를 계산
- VCC, Trig, Echo, GND의 4개의 핀
- Trig는 VCC에 신호가 들어오면 초음파을 발신
- 발신된 초음파가 장애물이나 벽에 부딪혀 다시 돌아오게 되면, Echo가 받아들임
- Trig를 통해 초음파는 340m/s의 속도로 발신






```
float duration;
float distance;

void setup( )
{
  Serial.begin(9600);
  pinMode(8,INPUT);//echo 입력
  pinMode(9,OUTPUT);//trig 출력
}

void loop( )
{
  digitalWrite(9, HIGH); //trig 5V 출력
  delay(10);
  digitalWrite(9, LOW);//trig 0V 출력

  //pulseIn함수의 단위는 ms(마이크로 세컨드)
  duration = pulseIn(8, HIGH);
  distance = ((34000*duration)/1000000)/2;//편도거리계산
  Serial.print("distance:");
  Serial.println(distance);
  delay(100);
}
```





- 파일메뉴/예제/HCSR04/simple

```
#include <HCSR04.h>

UltraSonicDistanceSensor distanceSensor(9, 8);  // Initialize sensor that uses digital pins 13 and 12.

void setup () {
    Serial.begin(9600);  // We initialize serial connection so that we could print values from sensor.
}

void loop () {
    // Every 500 miliseconds, do a measurement using the sensor and print the distance in centimeters.
    Serial.println(distanceSensor.measureDistanceCm());
    delay(500);
}
```

실습

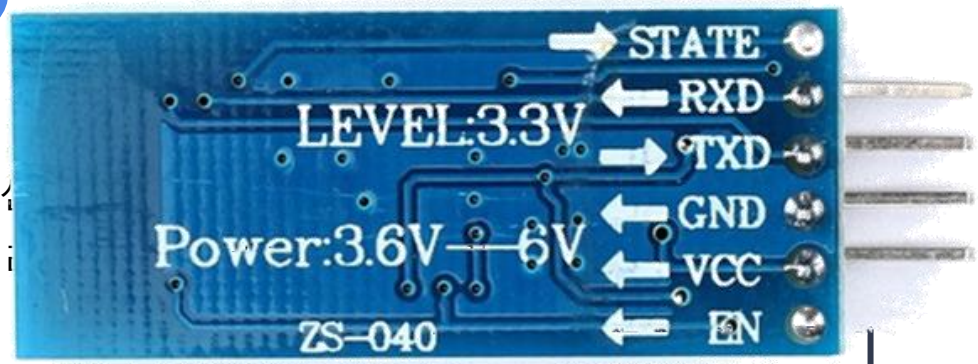
- 거리가 10Cm 이내이면 LED가 켜지는 동작을 위해 조립 및 코딩 하기

블루투스(BT06)

★블루투스(Bluetooth)

○ 블루투스는 디지털 무

10미터 안팎의 초단거

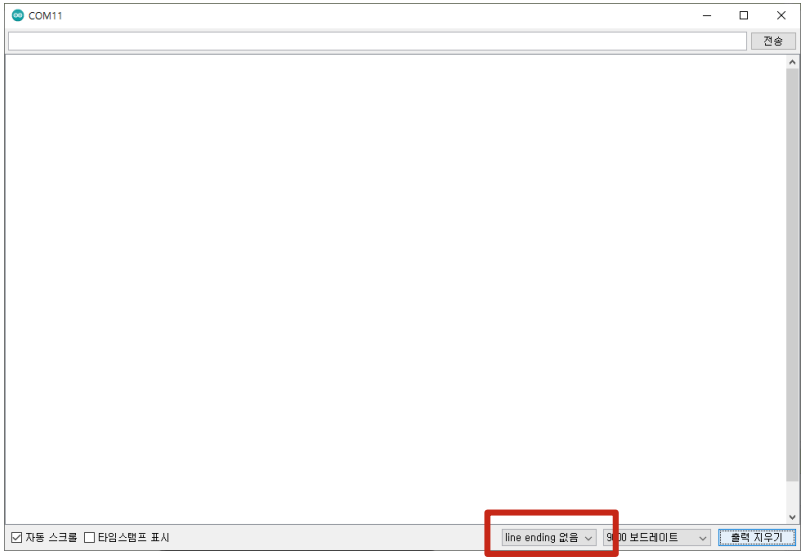


- 블루투스 모듈 검사 및 설정
  파일/예제/SoftwareSerial/ SoftwareSerialExample 수정

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(10, 11); //(Tx,Rx)

void setup()
{
  Serial.begin(9600);
  mySerial.begin(9600);
 }

void loop(){
  if(mySerial.available()){
     Serial.write(mySerial.read());
   }
   if (Serial.available()){
    mySerial.write(Serial.read());
  }
}
```

시리얼 모니터에
AT 입력
OK 나오면 정상 연결

AT+NAME영문이름



NL(Line Feed) : 커서를 아랫 줄(다음 줄)로
CR(Carriage Return) : 커서를 그줄의 맨 앞으로
Both NL & CR : 엔터키(Enter)

블루투스로 LED 제어하기

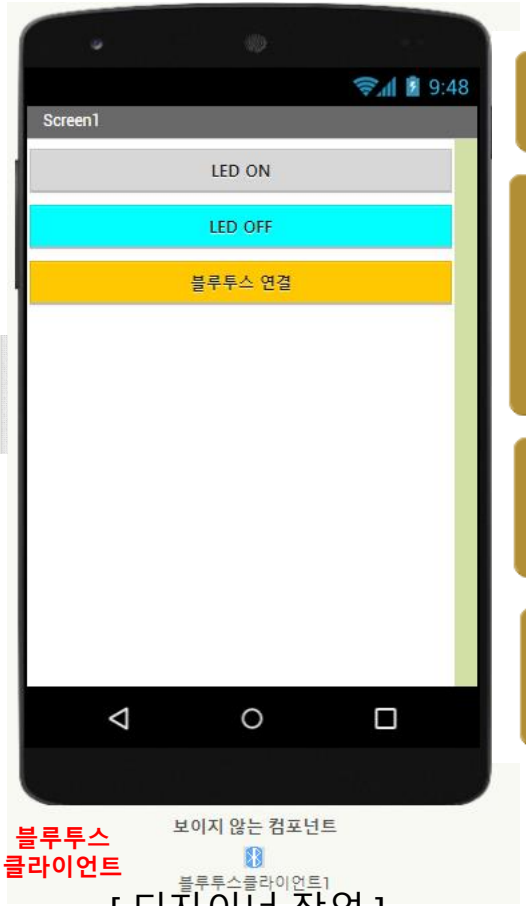
- APP Inventor 이용하여 휴대폰으로 아두이노 LED 제어하기

아두이노 소스 작성

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(10, 11);

void setup()
{
  Serial.begin(9600);
  mySerial.begin(9600);
  pinMode(9, OUTPUT);
}
```

```
void loop()
{
  char cmd;
  if(mySerial.available() )
  {
    cmd = mySerial.read();

    if( cmd == '1' )digitalWrite(9, HIGH);

    if( cmd == '0' )digitalWrite(9, LOW);
  }
}
```

●APP Inventor 앱 제작하기

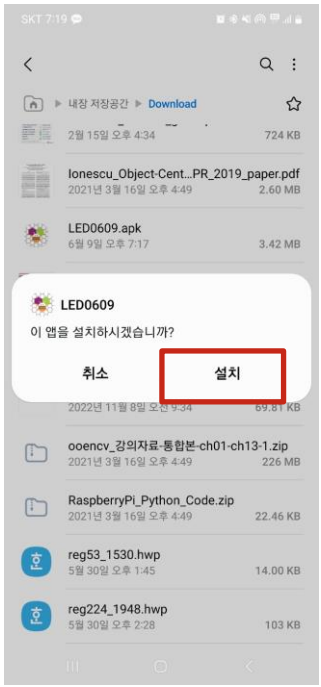


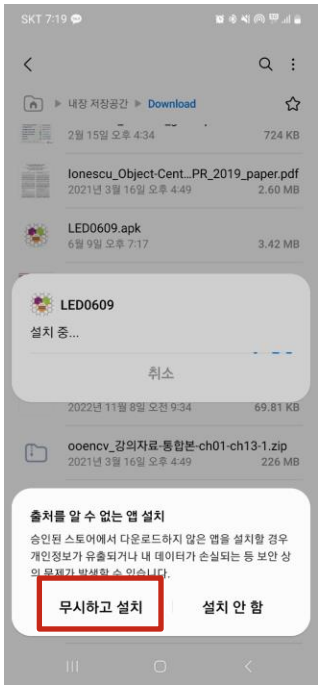
[디자이너 작업]

[블록 작업]

앱 다운로드 및 설치 방법

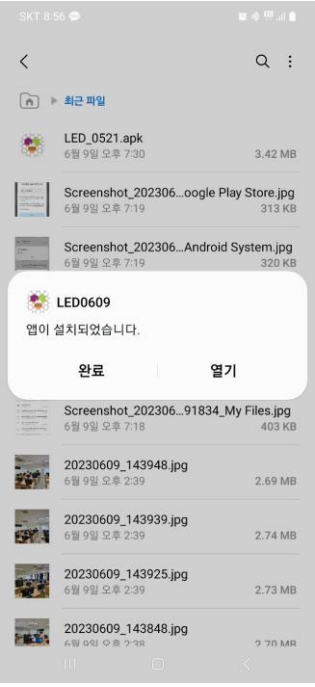

내장 저장공간 ▸ Download
2월 15일 오후 4:34 724 KB
Ionescu_Object-Cent...PR_2019_paper.pdf
2021년 3월 16일 오후 4:49 2.60 MB
LED0609.apk
6월 9일 오후 7:17 3.42 MB
MelDIR+Sample+Pr...32onMDK(E) (2).pdf
2022년 1월 19일 오후 2:55 1.59 MB
METAVERSE 2.0_20...03_compressed.pdf
2021년 7월 4일 오후 4:02 5.15 MB
Notes_221108_093428.jpg
2022년 11월 8일 오전 9:34 69.81 KB
ooencv_강의자료-통합본-ch01-ch13-1.zip
2021년 3월 16일 오후 4:49 226 MB
RaspberryPi_Python_Code.zip
2021년 3월 16일 오후 4:49 22.46 KB
reg53_1530.hwp
5월 30일 오후 1:45 14.00 KB
reg224_1948.hwp
5월 30일 오후 2:28 103 KB


LED0609
이 앱을 설치하시겠습니까?
취소
설치

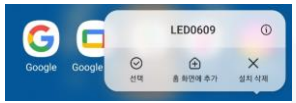

LED0609
설치 중...
취소
출처를 알 수 없는 앱 설치
승인된 스토어에서 다운로드하지 않은 앱을 설치할 경우 개인정보가 유출되거나 내 데이터가 손실되는 등 보안 상
무시하고 설치
설치 안 함


최근에 추가된 파일 1개
Google Play 프로텍트
안전하지 않은 앱 차단됨
CDS_0609
Play 프로텍트에서 앱 개발자를 인식할 수 없습니다. 알 수 없는 개발자의 앱은 안전하지 않을 수 있습니다.
세부정보 더보기
확인
OneDrive
Google 드라이브
네트워크 저장공간

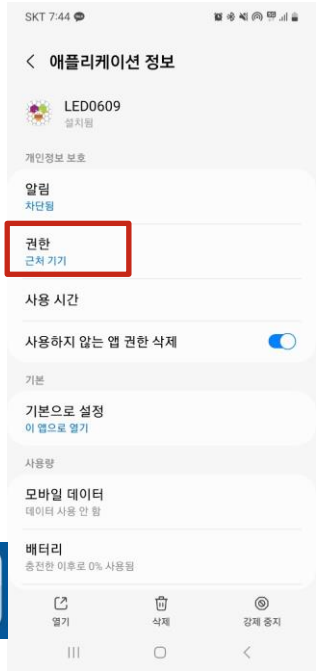

Google Play 프로텍트
안전하지 않은 앱 차단됨
LED0609
Play 프로텍트에서 앱 개발자를 인식할 수 없습니다. 알 수 없는 개발자의 앱은 안전하지 않을 수 있습니다.
이 앱을 설치하면 기기가 위험에 노출될 수 있습니다. Play 프로텍트에 관해 자세히 알아보세요.
무시하고 설치하기
확인

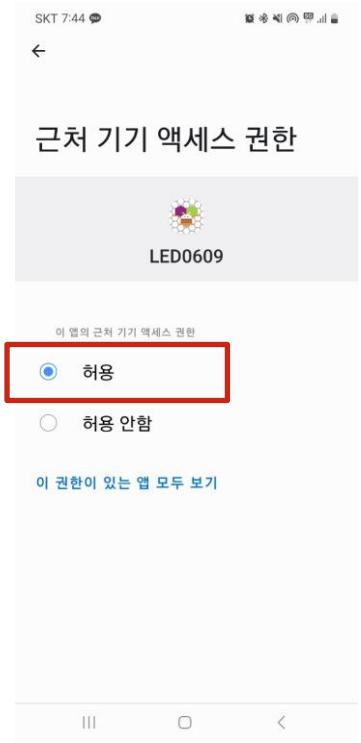

LED_0521.apk
Screenshot_202306...oogle Play Store.jpg
Screenshot_202306...Android System.jpg
LED0609
앱이 설치되었습니다.
완료
열기
Screenshot_202306...91834_My Files.jpg
20230609_143948.jpg
20230609_143939.jpg
20230609_143925.jpg
20230609_143848.jpg

앱 권한 변경 방법



앱 우클릭
느낌표 클릭





앱
블루투스확인