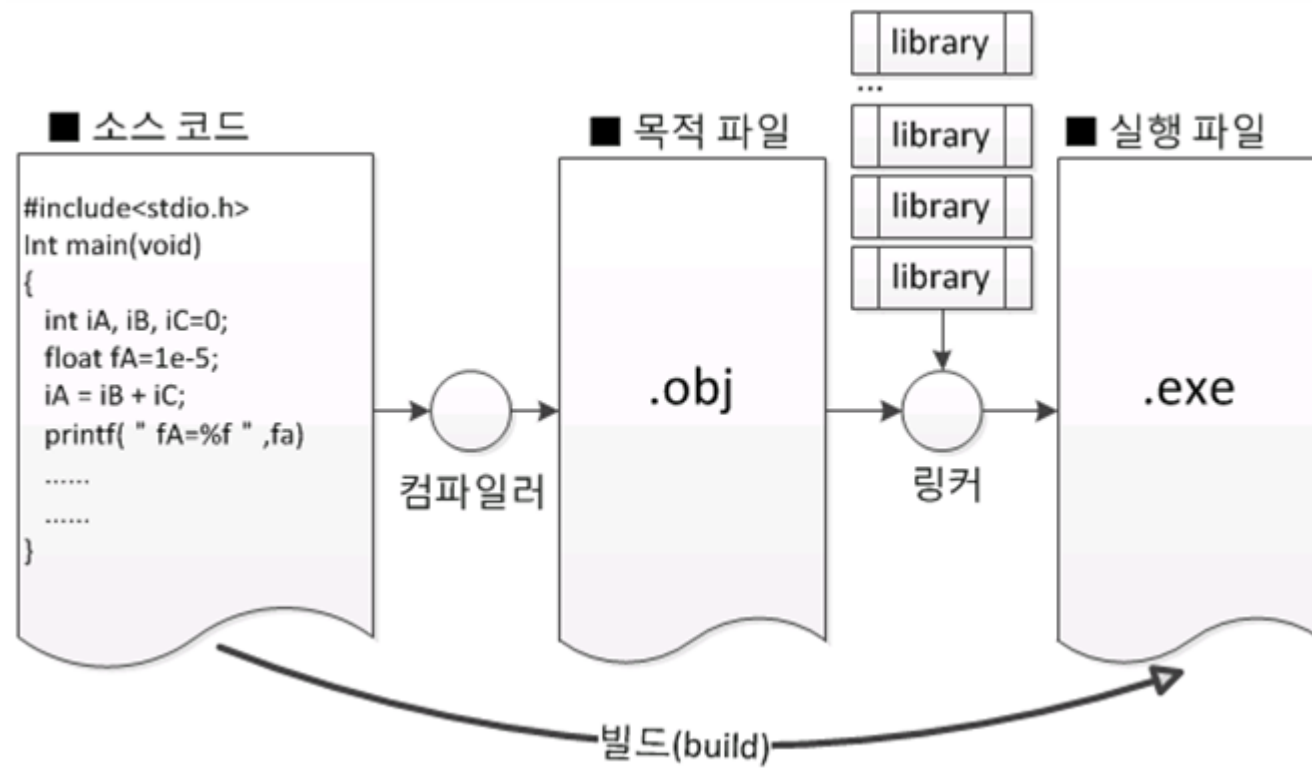


아두이노

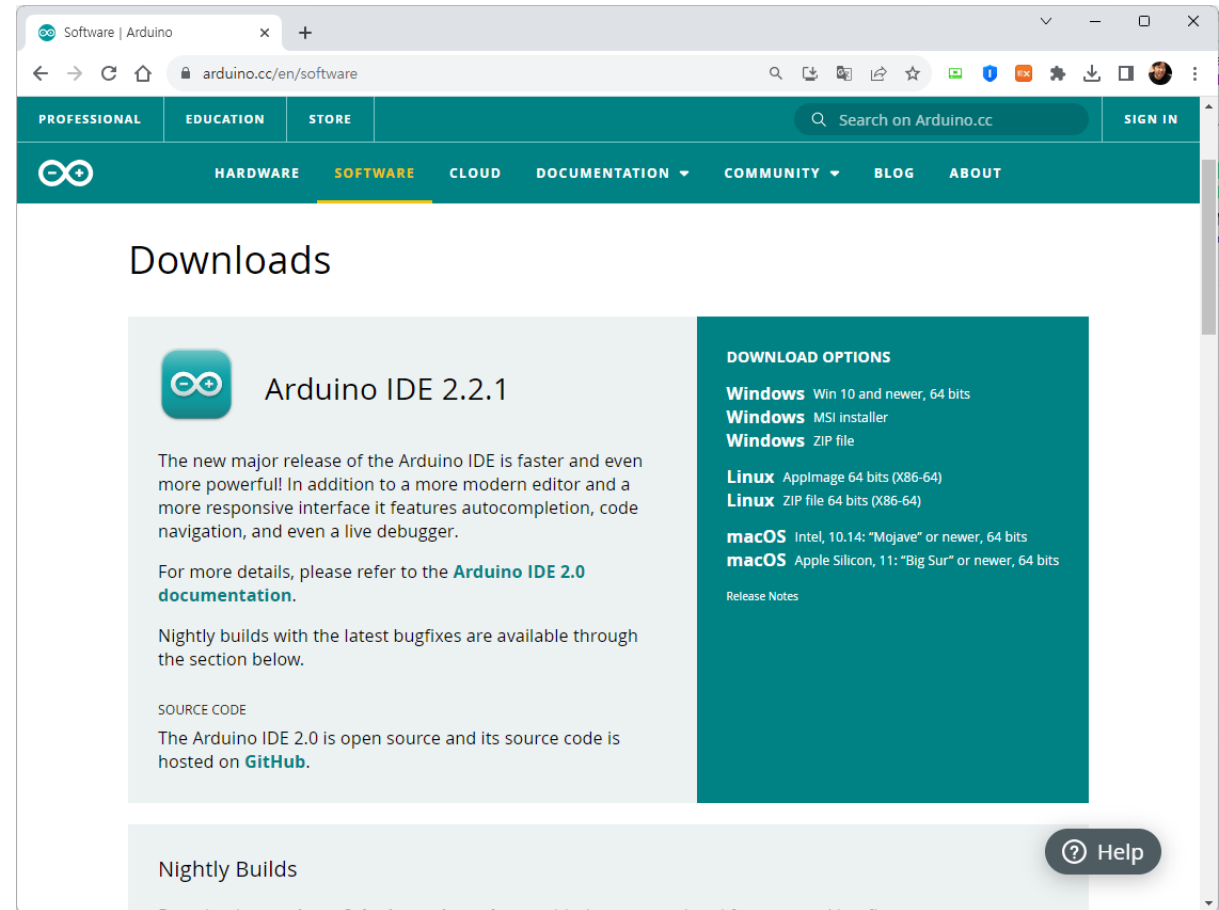
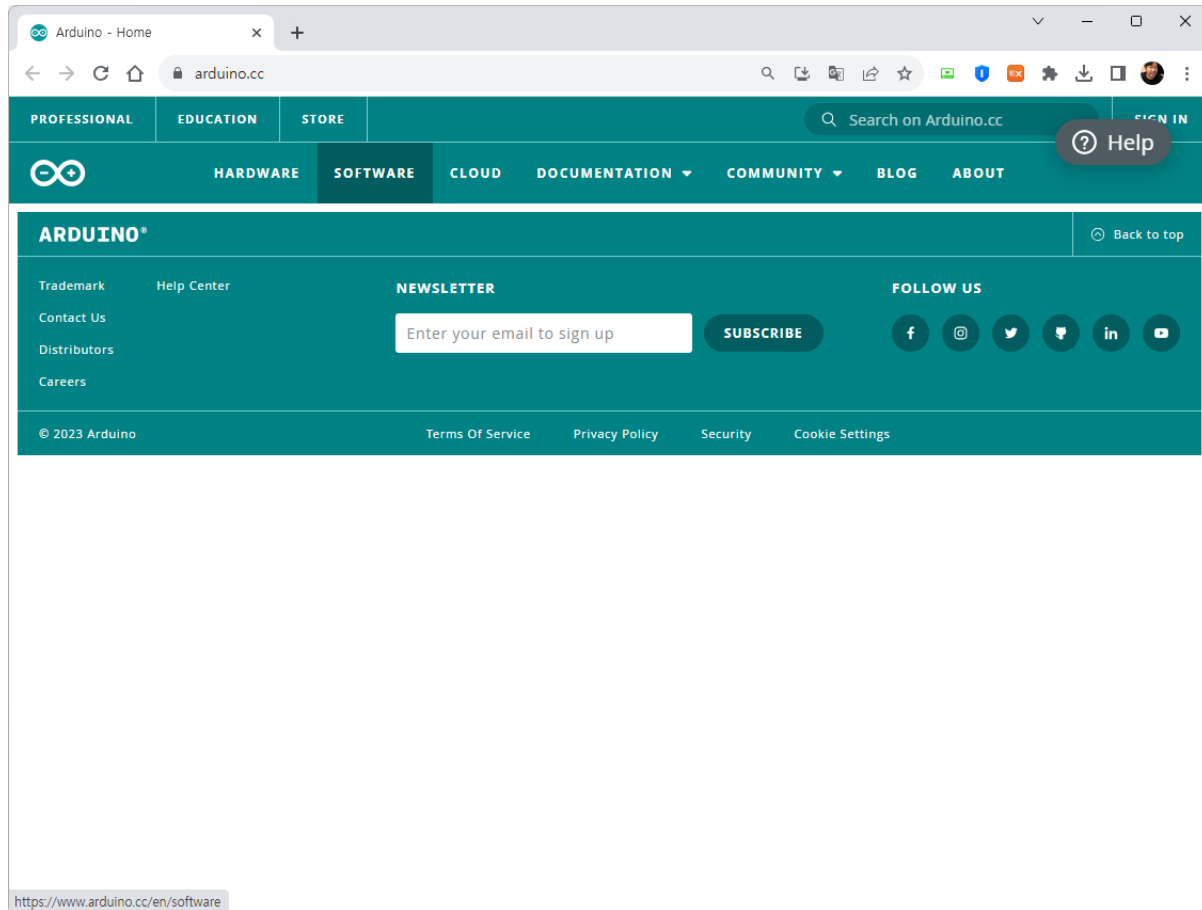


# 아두이노 SW 프로그래밍 과정



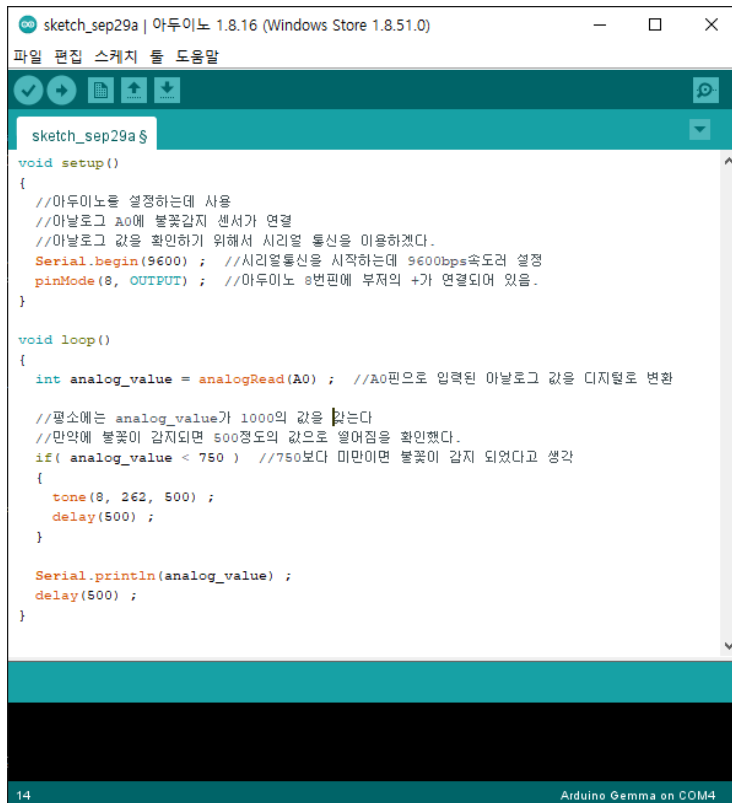
# 아두이노 개발툴 다운로드

- <https://www.arduino.cc/>



# 아두이노 프로그래밍 과정

- 소스코드 작성



```
sketch_sep29a $
void setup()
{
  //아두이노를 설정하는데 사용
  //아날로그 A0에 불꽃감지 센서가 연결
  //아날로그 값을 확인하기 위해서 시리얼 통신을 이용하겠다.
  Serial.begin(9600); //시리얼통신을 시작하는데 9600bps속도려 설정
  pinMode(8, OUTPUT); //아두이노 8번핀에 부저의 +가 연결되어 있음.
}

void loop()
{
  int analog_value = analogRead(A0); //A0핀으로 입력된 아날로그 값을 디지털로 변환

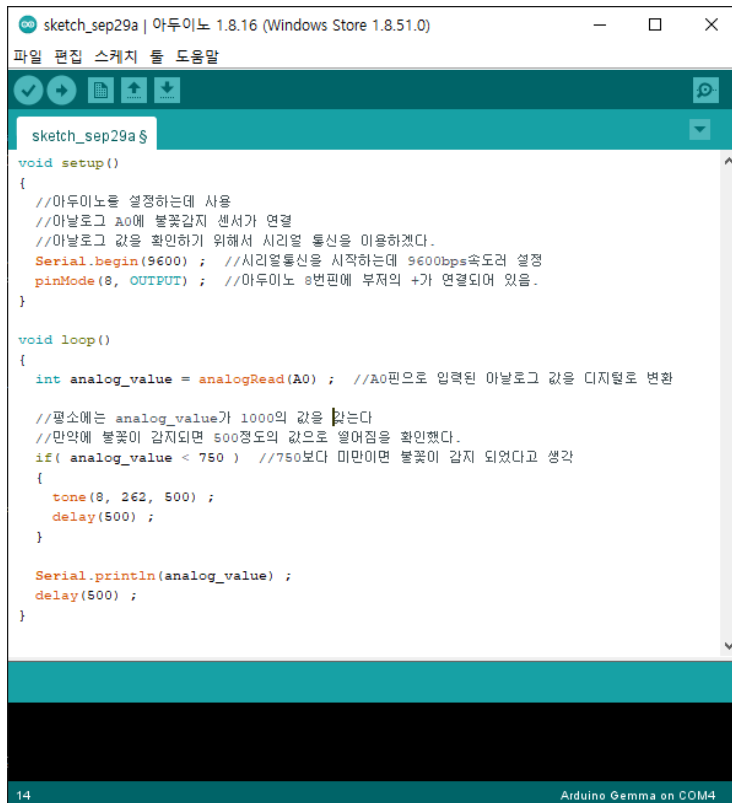
  //평소에는 analog_value가 1000의 값을 갖는다
  //만약에 불꽃이 감지되면 500정도의 값으로 떨어짐을 확인했다.
  if( analog_value < 750 ) //750보다 미만이면 불꽃이 감지 되었다고 생각
  {
    tone(8, 262, 500);
    delay(500);
  }

  Serial.println(analog_value);
  delay(500);
}
```

14 Arduino Gemma on COM4

# 아두이노 프로그래밍 과정

- 컴파일

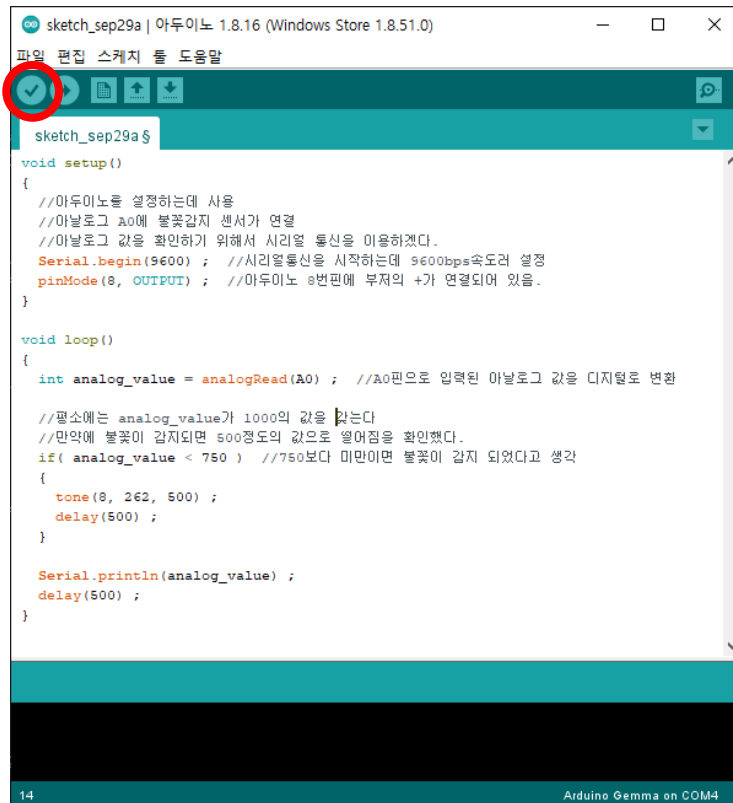


```
sketch_sep29a $
void setup()
{
  //아두이노를 설정하는데 사용
  //아날로그 A0에 불꽃감지 센서가 연결
  //아날로그 값을 확인하기 위해서 시리얼 통신을 이용하겠다.
  Serial.begin(9600); //시리얼통신을 시작하는데 9600bps속도려 설정
  pinMode(8, OUTPUT); //아두이노 8번핀에 부저의 +가 연결되어 있음.
}

void loop()
{
  int analog_value = analogRead(A0); //A0핀으로 입력된 아날로그 값을 디지털로 변환

  //평소에는 analog_value가 1000의 값을 갖는다
  //만약에 불꽃이 감지되면 500경도의 값으로 떨어짐을 확인했다.
  if( analog_value < 750 ) //750보다 미만이면 불꽃이 감지 되었다고 생각
  {
    tone(8, 262, 500);
    delay(500);
  }

  Serial.println(analog_value);
  delay(500);
}
```



```
sketch_sep29a $
void setup()
{
  //아두이노를 설정하는데 사용
  //아날로그 A0에 불꽃감지 센서가 연결
  //아날로그 값을 확인하기 위해서 시리얼 통신을 이용하겠다.
  Serial.begin(9600); //시리얼통신을 시작하는데 9600bps속도려 설정
  pinMode(8, OUTPUT); //아두이노 8번핀에 부저의 +가 연결되어 있음.
}

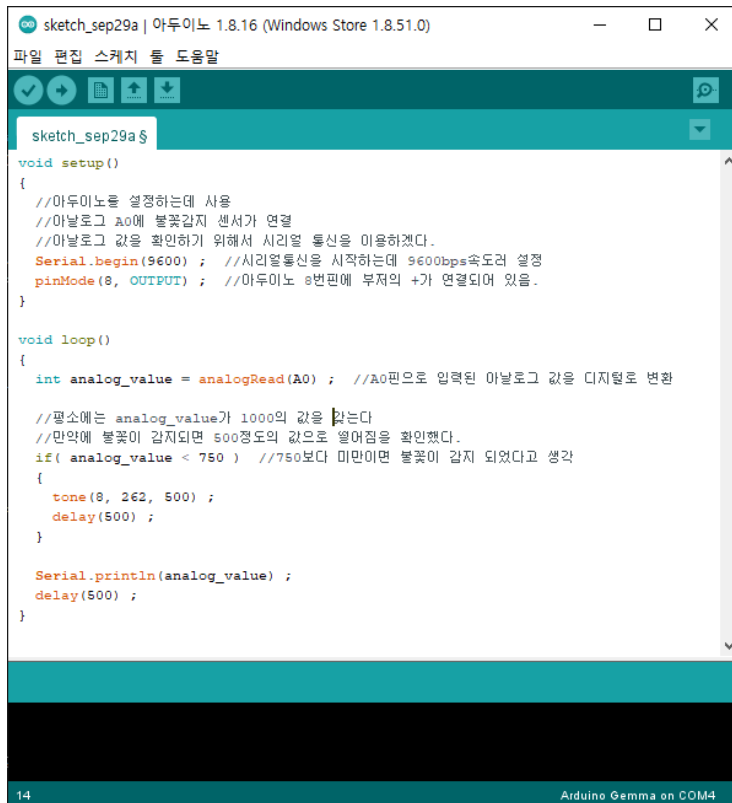
void loop()
{
  int analog_value = analogRead(A0); //A0핀으로 입력된 아날로그 값을 디지털로 변환

  //평소에는 analog_value가 1000의 값을 갖는다
  //만약에 불꽃이 감지되면 500경도의 값으로 떨어짐을 확인했다.
  if( analog_value < 750 ) //750보다 미만이면 불꽃이 감지 되었다고 생각
  {
    tone(8, 262, 500);
    delay(500);
  }

  Serial.println(analog_value);
  delay(500);
}
```

# 아두이노 프로그래밍 과정

## • 업로드 & 실행

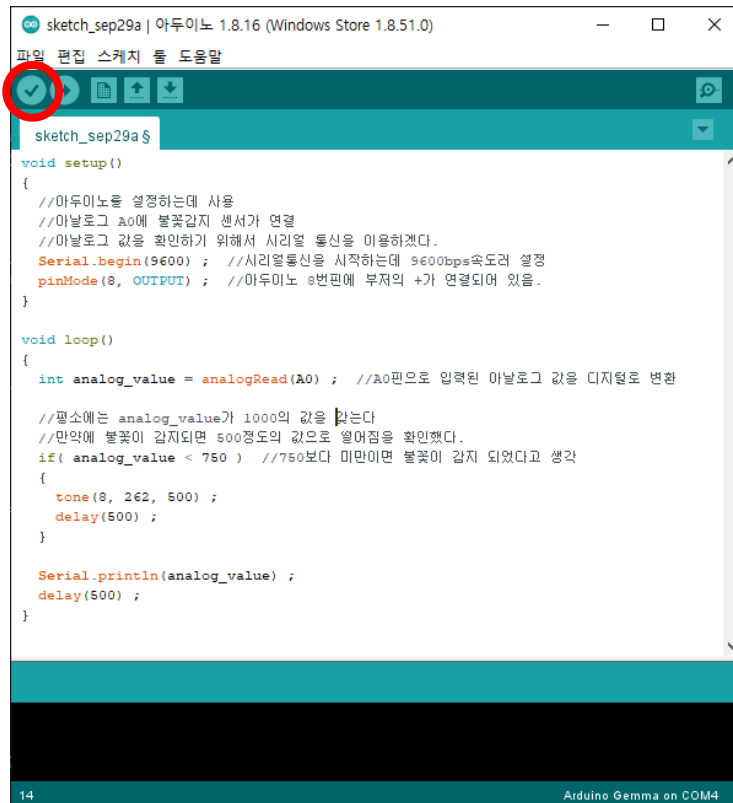


```
sketch_sep29a $
void setup()
{
  //아두이노를 설정하는데 사용
  //아날로그 A0에 불꽃감지 센서가 연결
  //아날로그 값을 확인하기 위해서 시리얼 통신을 이용하겠다.
  Serial.begin(9600); //시리얼통신을 시작하는데 9600bps속도려 설정
  pinMode(8, OUTPUT); //아두이노 8번핀에 부저의 +가 연결되어 있음.
}

void loop()
{
  int analog_value = analogRead(A0); //A0핀으로 입력된 아날로그 값을 디지털로 변환

  //평소에는 analog_value가 1000의 값을 갖는다
  //만약에 불꽃이 감지되면 500정도의 값으로 떨어짐을 확인했다.
  if( analog_value < 750 ) //750보다 미만이면 불꽃이 감지 되었다고 생각
  {
    tone(8, 262, 500);
    delay(500);
  }

  Serial.println(analog_value);
  delay(500);
}
```

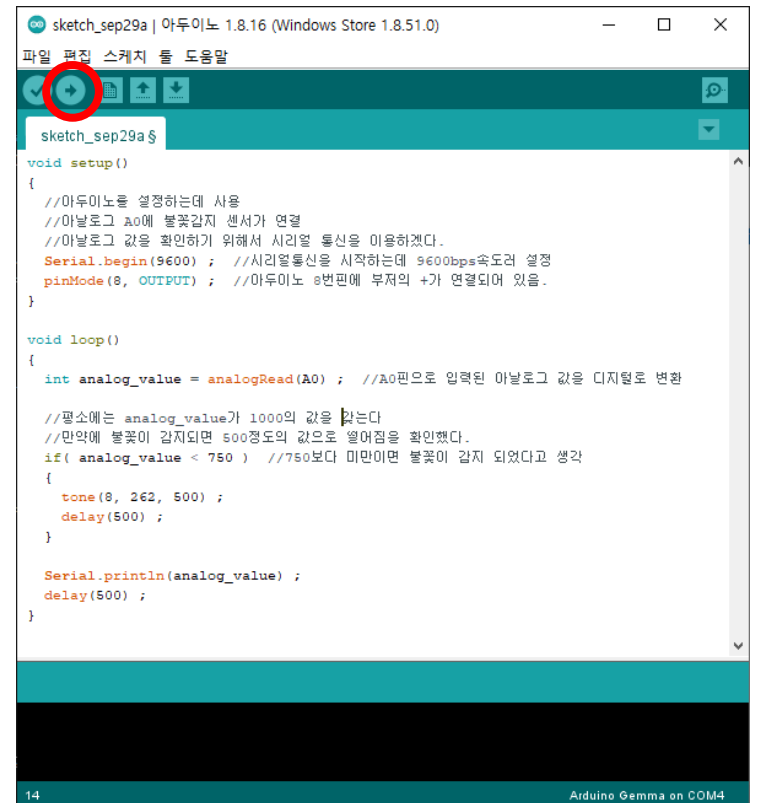


```
sketch_sep29a $
void setup()
{
  //아두이노를 설정하는데 사용
  //아날로그 A0에 불꽃감지 센서가 연결
  //아날로그 값을 확인하기 위해서 시리얼 통신을 이용하겠다.
  Serial.begin(9600); //시리얼통신을 시작하는데 9600bps속도려 설정
  pinMode(8, OUTPUT); //아두이노 8번핀에 부저의 +가 연결되어 있음.
}

void loop()
{
  int analog_value = analogRead(A0); //A0핀으로 입력된 아날로그 값을 디지털로 변환

  //평소에는 analog_value가 1000의 값을 갖는다
  //만약에 불꽃이 감지되면 500정도의 값으로 떨어짐을 확인했다.
  if( analog_value < 750 ) //750보다 미만이면 불꽃이 감지 되었다고 생각
  {
    tone(8, 262, 500);
    delay(500);
  }

  Serial.println(analog_value);
  delay(500);
}
```



```
sketch_sep29a $
void setup()
{
  //아두이노를 설정하는데 사용
  //아날로그 A0에 불꽃감지 센서가 연결
  //아날로그 값을 확인하기 위해서 시리얼 통신을 이용하겠다.
  Serial.begin(9600); //시리얼통신을 시작하는데 9600bps속도려 설정
  pinMode(8, OUTPUT); //아두이노 8번핀에 부저의 +가 연결되어 있음.
}

void loop()
{
  int analog_value = analogRead(A0); //A0핀으로 입력된 아날로그 값을 디지털로 변환

  //평소에는 analog_value가 1000의 값을 갖는다
  //만약에 불꽃이 감지되면 500정도의 값으로 떨어짐을 확인했다.
  if( analog_value < 750 ) //750보다 미만이면 불꽃이 감지 되었다고 생각
  {
    tone(8, 262, 500);
    delay(500);
  }

  Serial.println(analog_value);
  delay(500);
}
```

# 아두이노 ADC 실험과 광 (빛)센서 실험

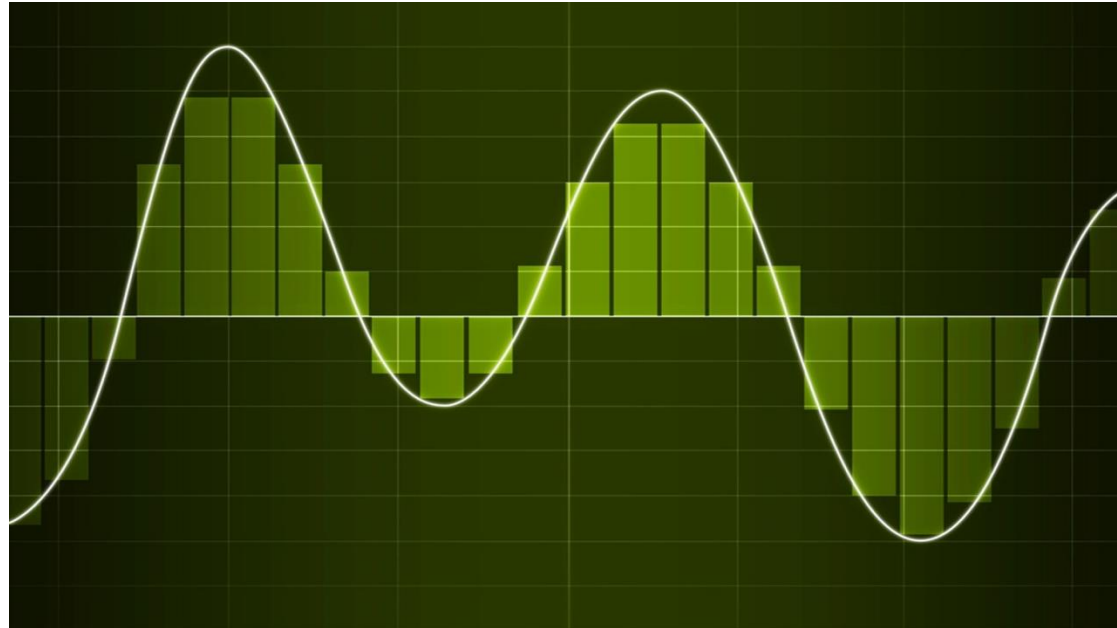
# 센서란?

- 인간의 5감 대신에 그 역할을 하는 기기
- 우리 주위 환경에 대한 정보를 전기적 신호로 수집하는 계측기
  - 예) 기온, 습도, 압력, 빛, 소리 등
- 전기적 신호로 계측된 정보를 확인하려면 컴퓨터에서 그 값을 읽어야 함.
- 컴퓨터는 아날로그 값을 읽지 못함 → 디지털 값으로 변환 시켜줘야 함. 이때 사용하는 것이 ADC

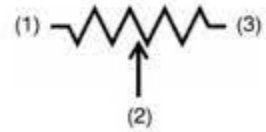
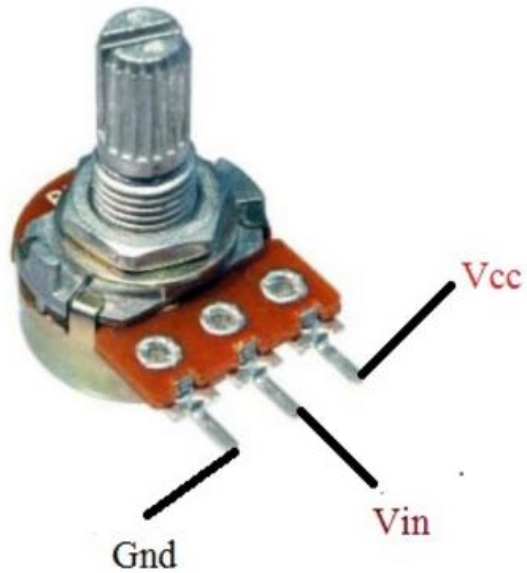


# ADC(Analog-to-digital converter)

- 입력 된 아날로그 전기 신호를 디지털 신호로 변환하는 장치
- 센서로 수집된 전기 신호를 디지털 신호로 변환하면 컴퓨터에서 센서의 데이터를 확인 가능

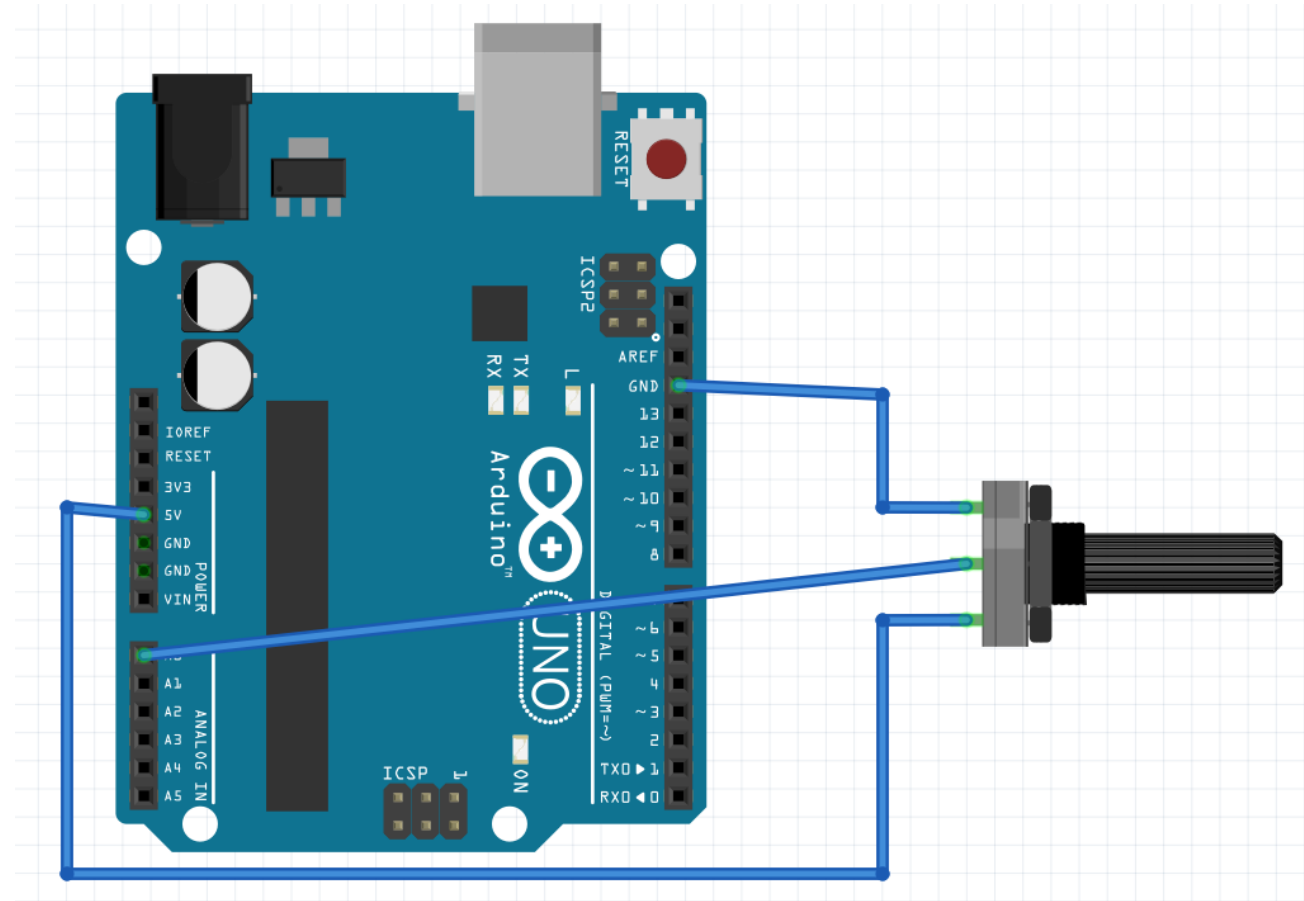


# 아두이노 ADC 실험

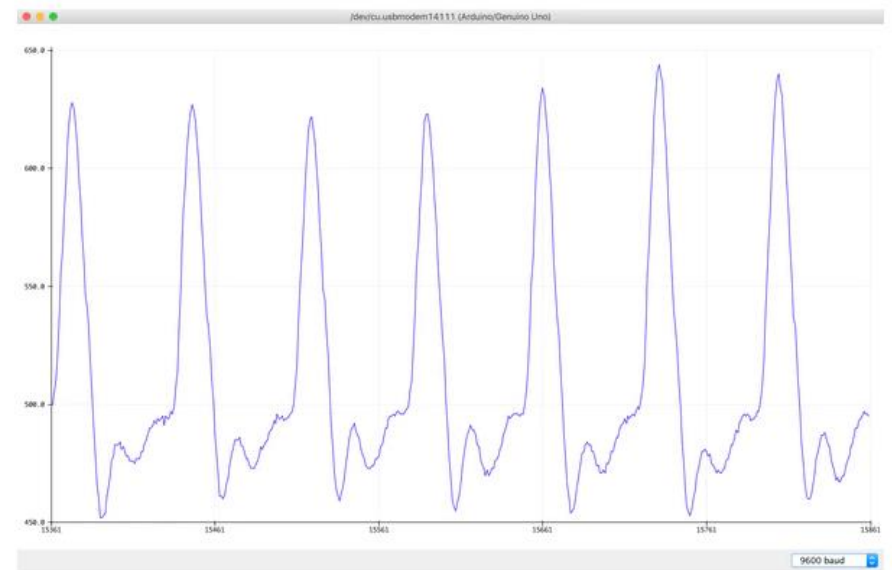
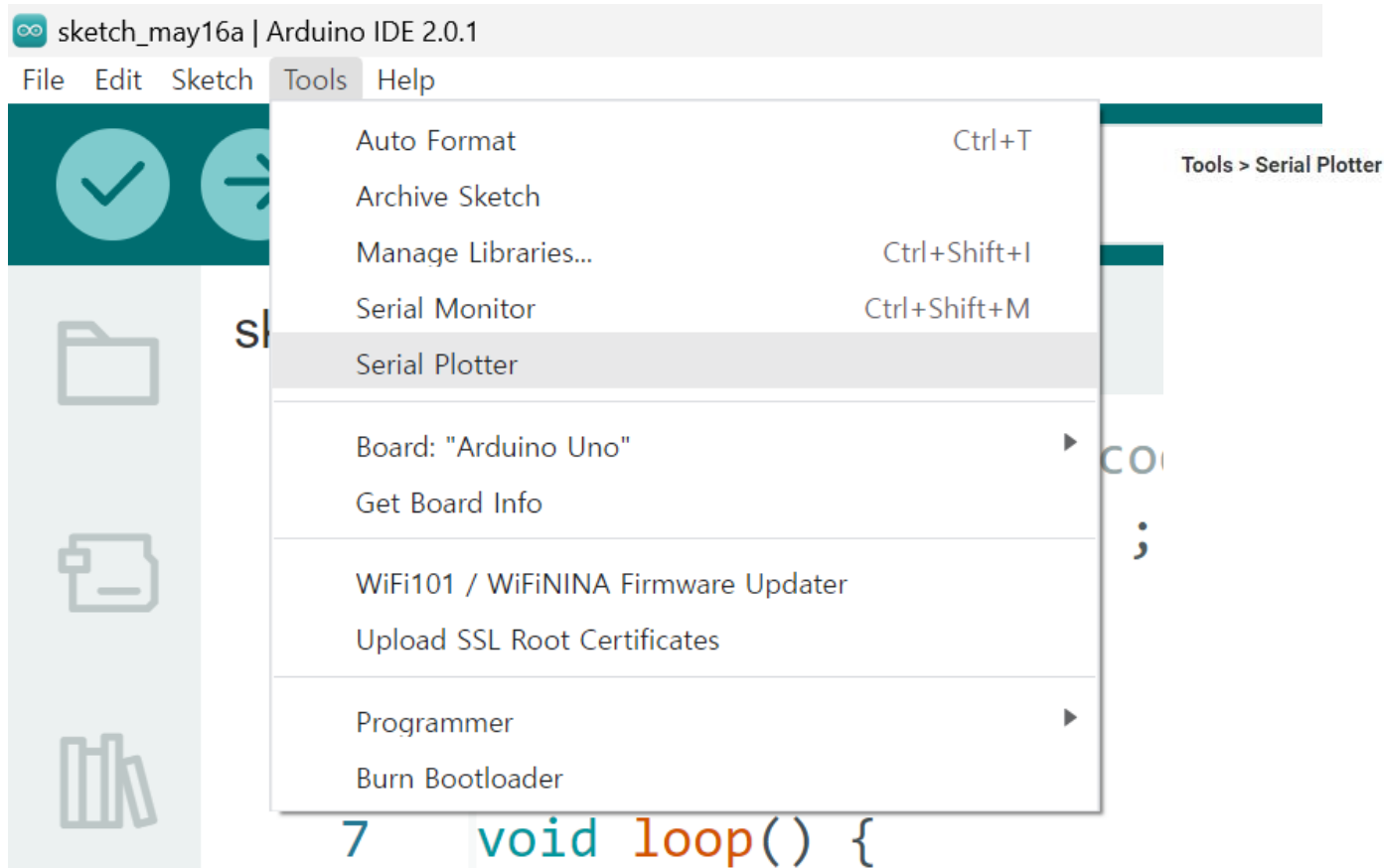


# 가변저항(Potentiometer, 볼륨)

```
void setup (){  
  Serial.begin(9600);  
}  
  
void loop(){  
  int val = analogRead(A0);  
  Serial.print("Analog : ");  
  Serial.println(val);  
}
```

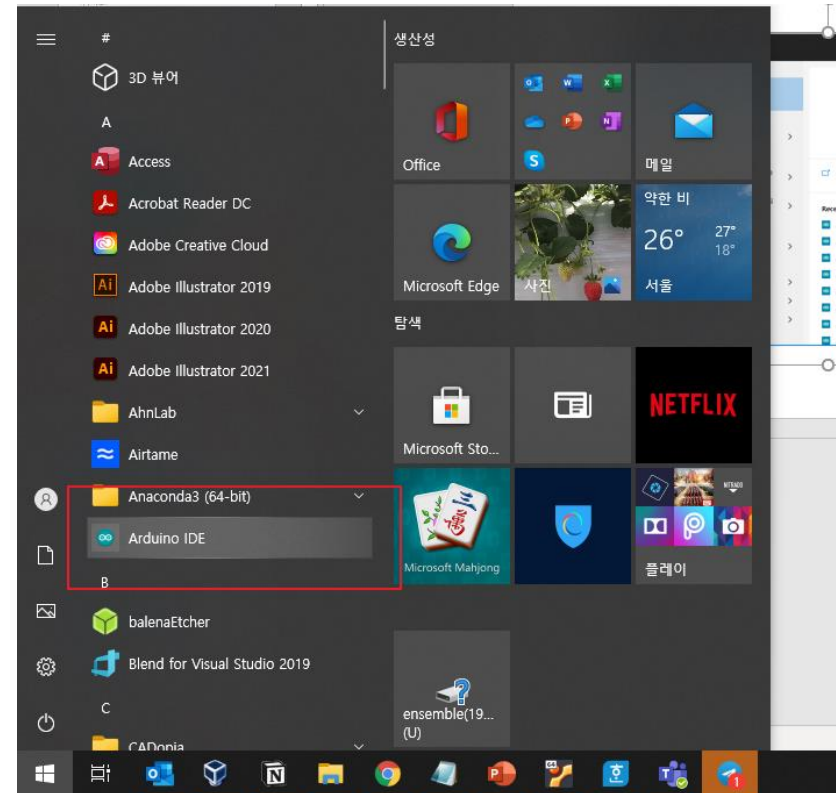
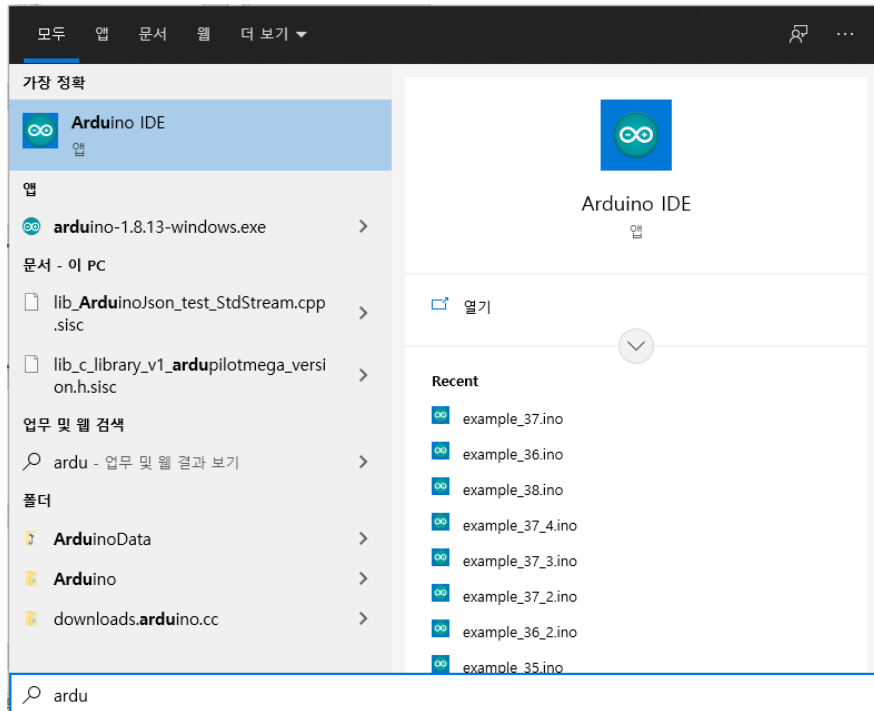


# 센서값을 그래프로 확인

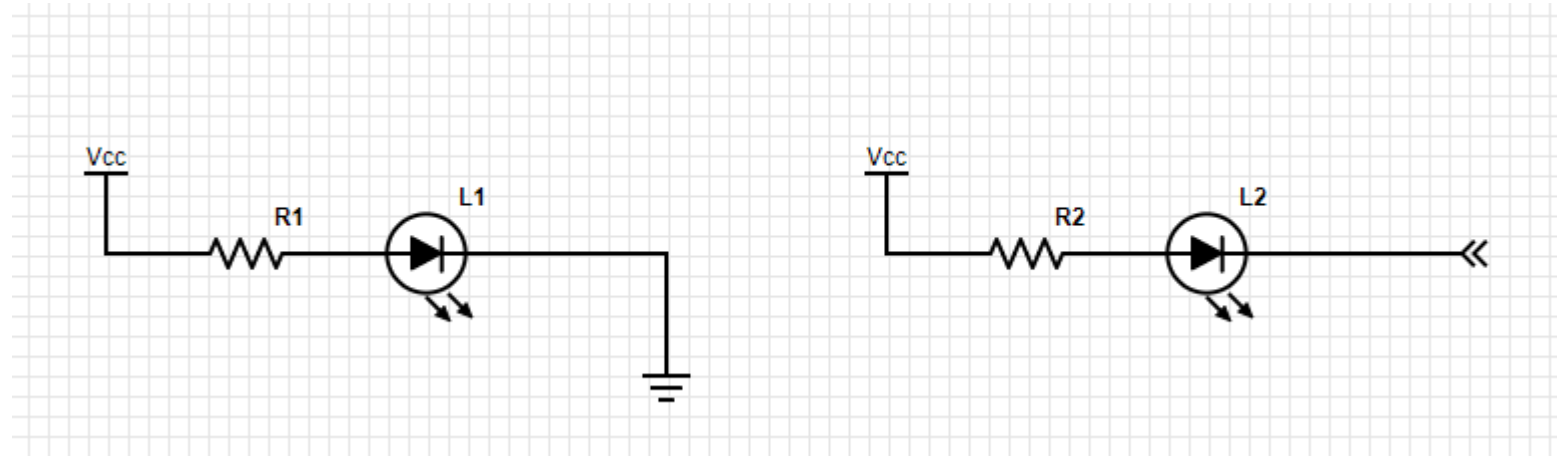
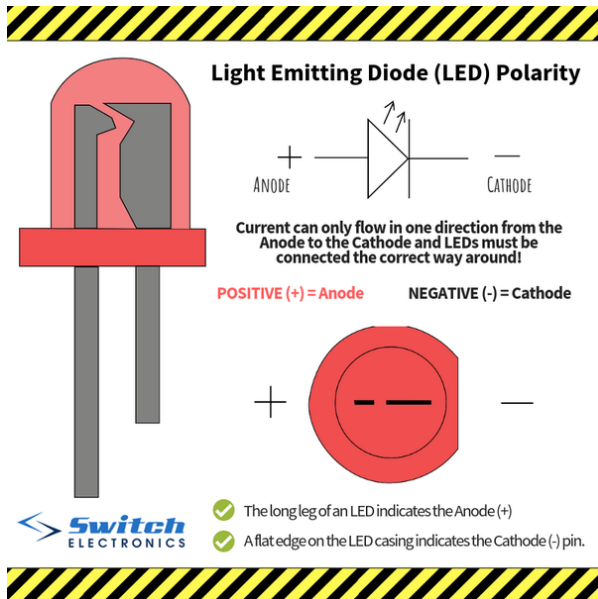


# 아두이노 개발 환경 구성 및 LED 테스트

- Arduino IDE 실행
  - Arduino IDE를 검색하거나 Arduino IDE 실행 아이콘을 클릭한다.

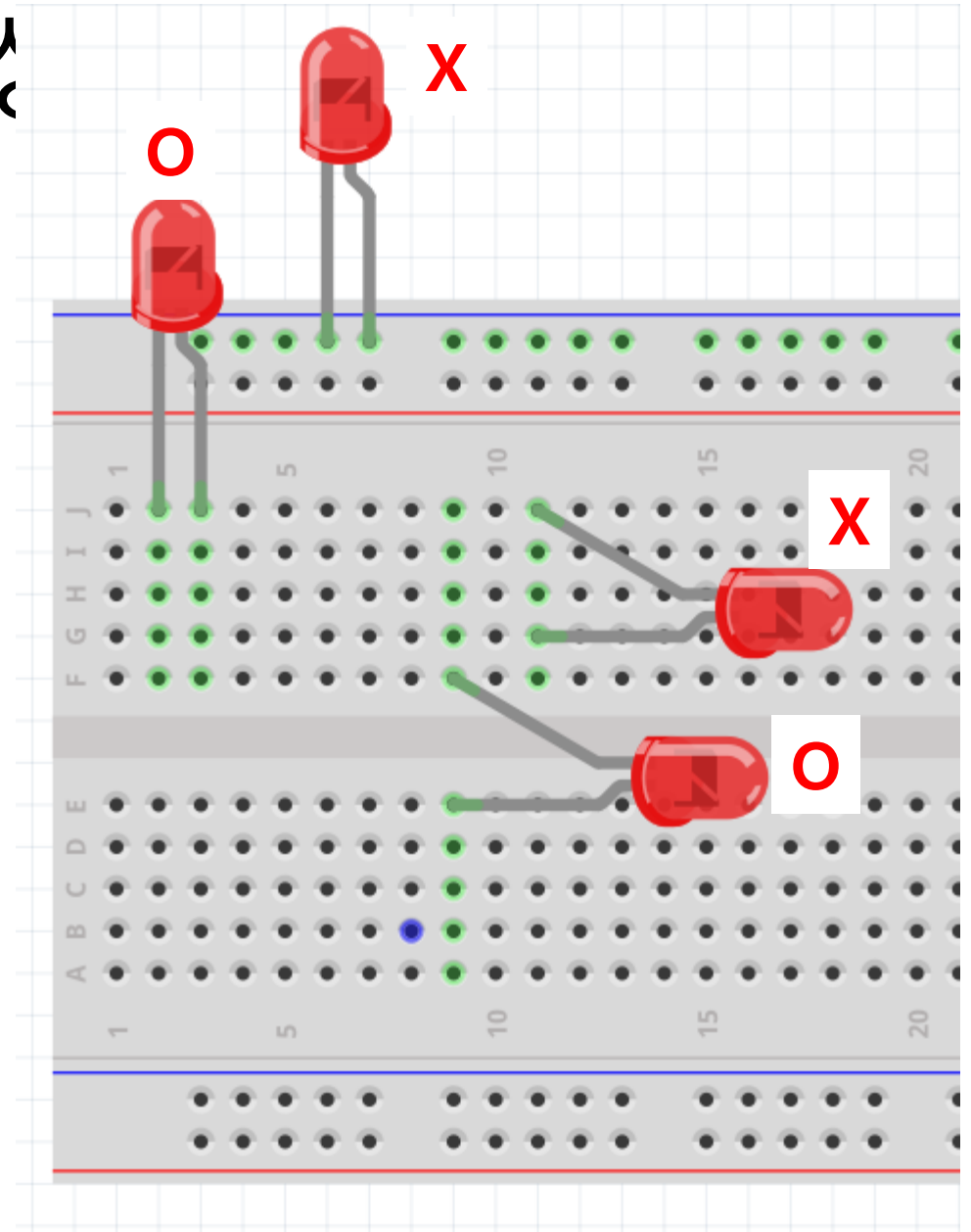
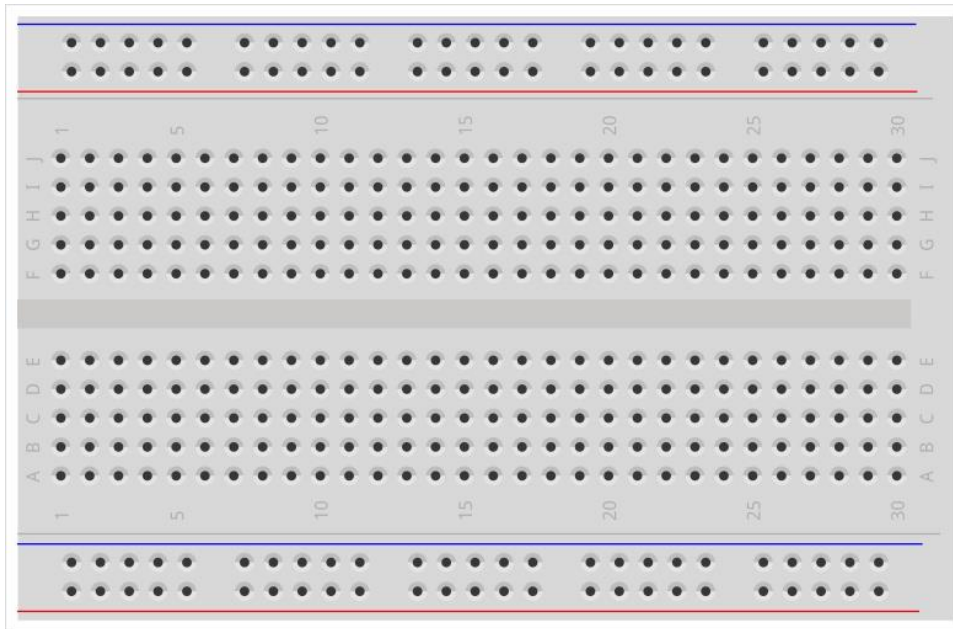


# 아두이노 개발 환경 구성 및 LED 테스트



# 아두이노 개발 환경 구성

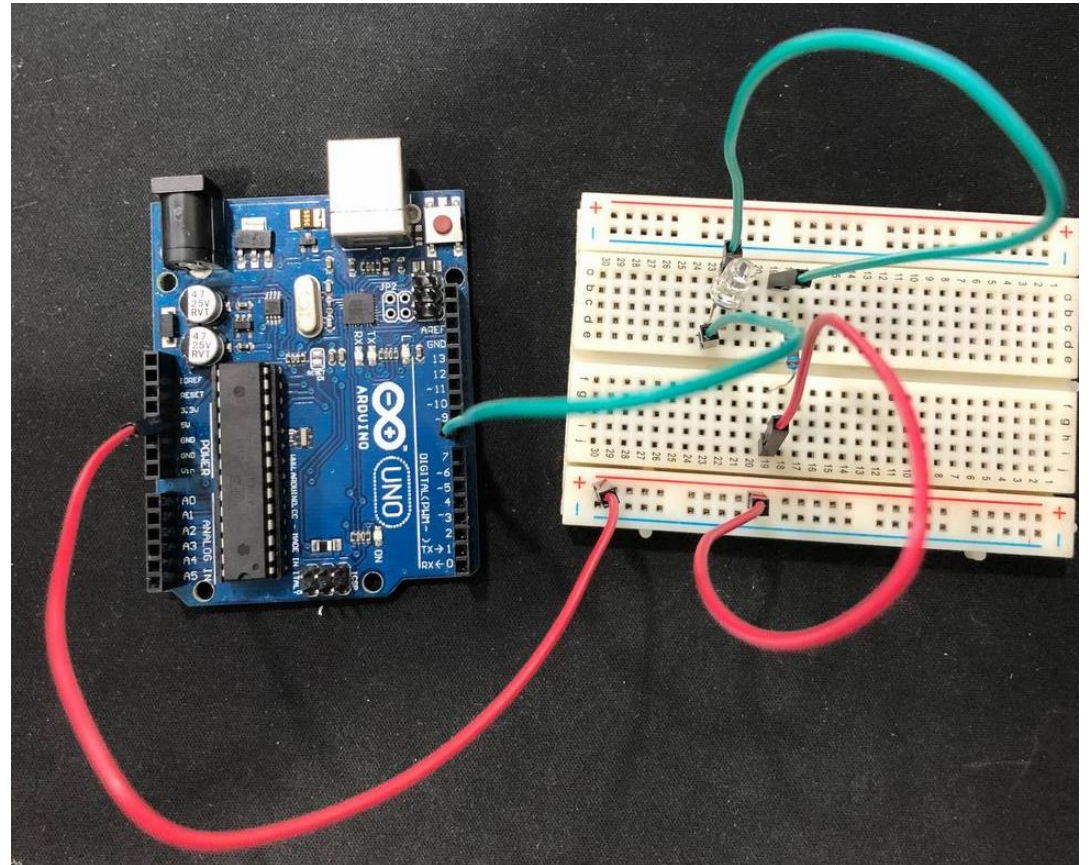
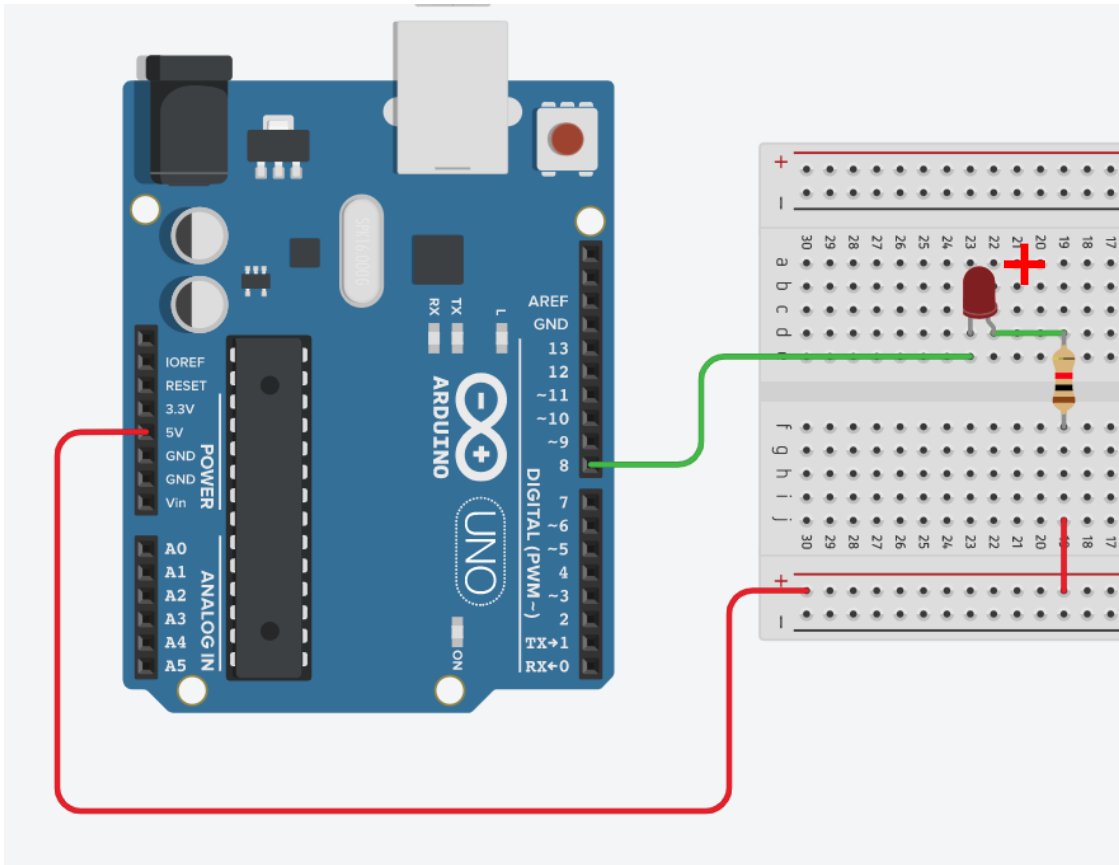
- 빵판 사용법(브레드보드)





# 아두이노 개발 환경 구성 및 LED 테스트

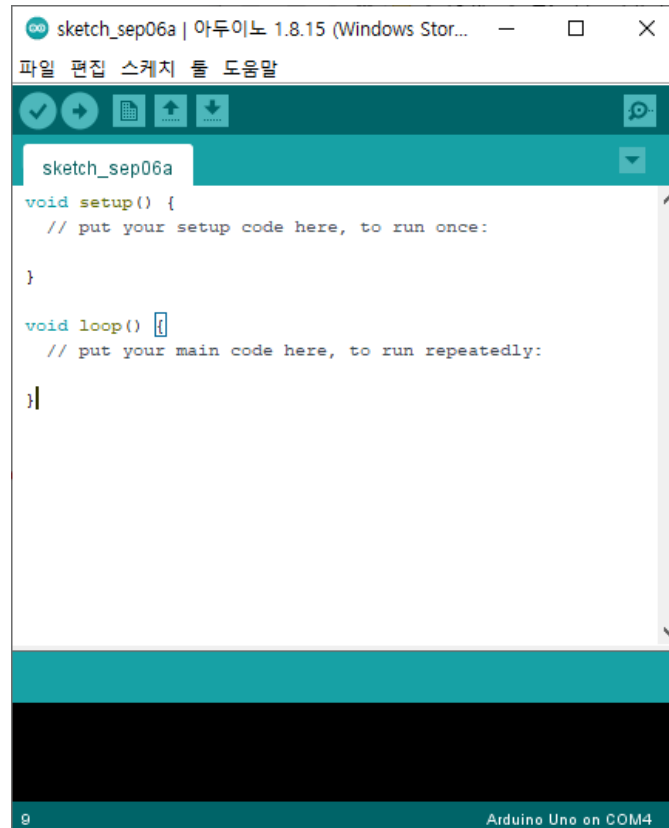
- Arduino LED 회로 구성





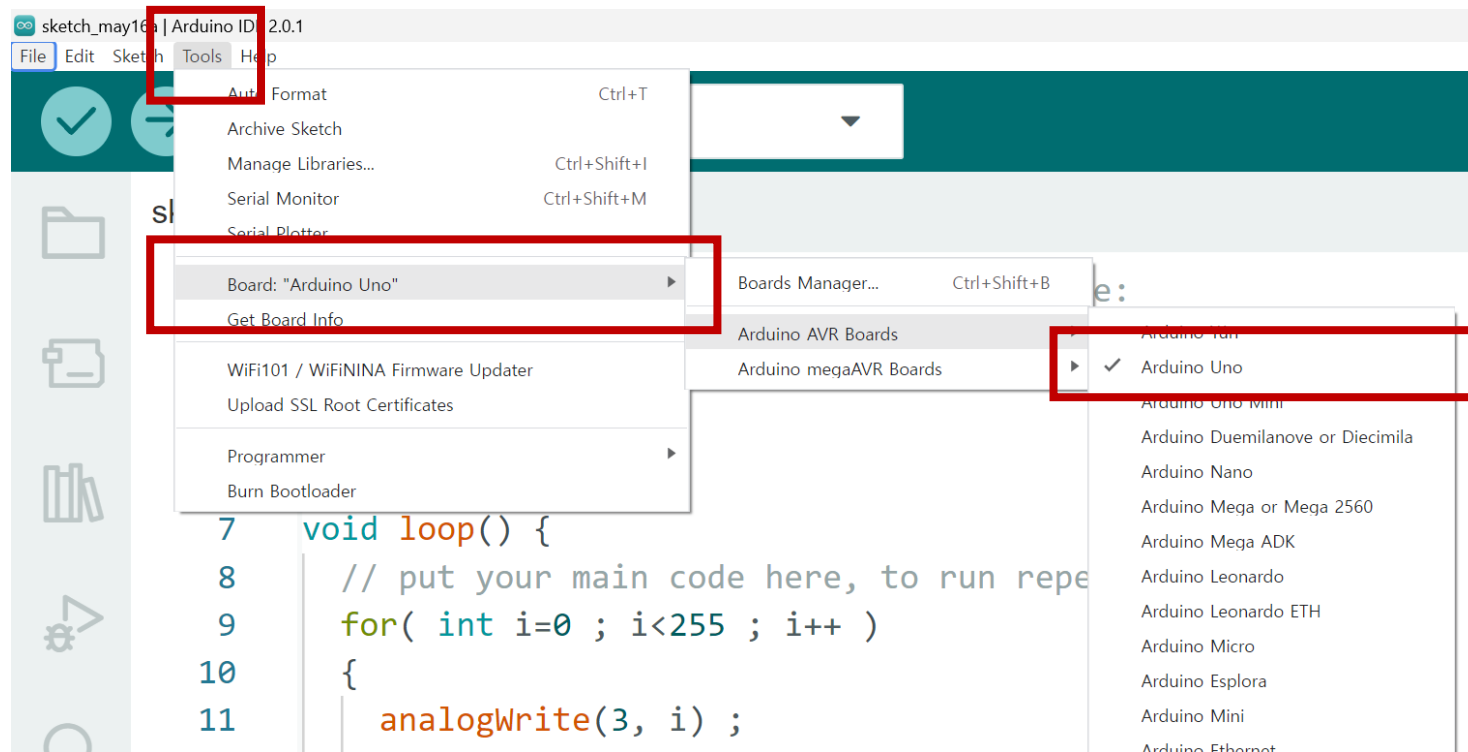
# 아두이노 개발 환경 구성 및 LED 테스트

- Arduino LED ON/OFF 코드 작성
  - 아두이노 IDE 실행



# 아두이노 개발 환경 구성 및 LED 테스트

- Arduino LED ON/OFF 코드 작성
  - 아두이노 IDE에서 보드 선택 : **Menu → Tools → Board → Arduino Uno**



# 아두이노 개발 환경 구성 및 LED 테스트

- Arduino LED ON/OFF 코드 작성
  - 코드 작성

```
// C++ code
//
void setup()
{
  pinMode(8, OUTPUT);
}

void loop()
{
  digitalWrite(8, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(8, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

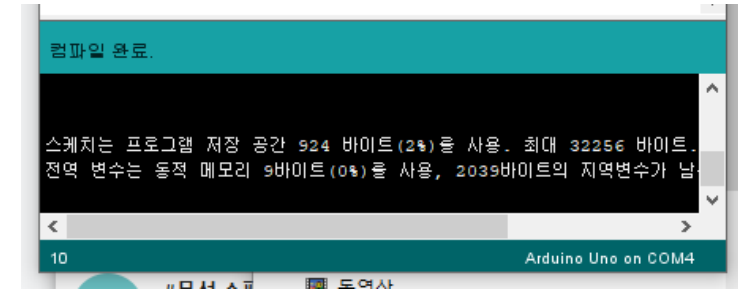
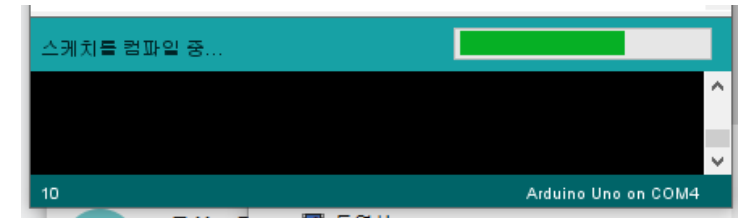
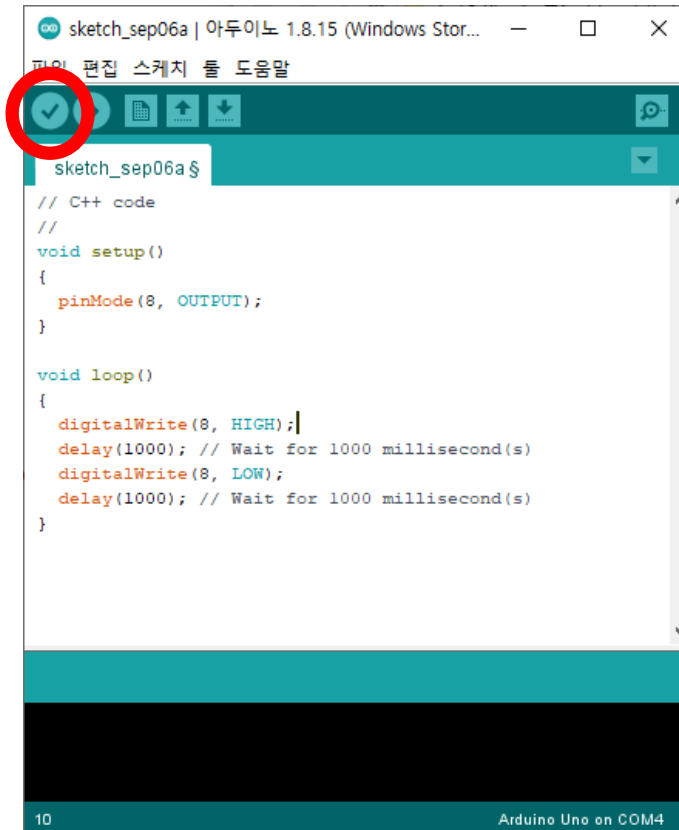


# 아두이노 개발 환경 구성 및 LED 테스트

- Arduino LED ON/OFF 코드 작성
  - 컴파일

```
// C++ code
//
void setup()
{
  pinMode(8, OUTPUT);
}

void loop()
{
  digitalWrite(8, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(8, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

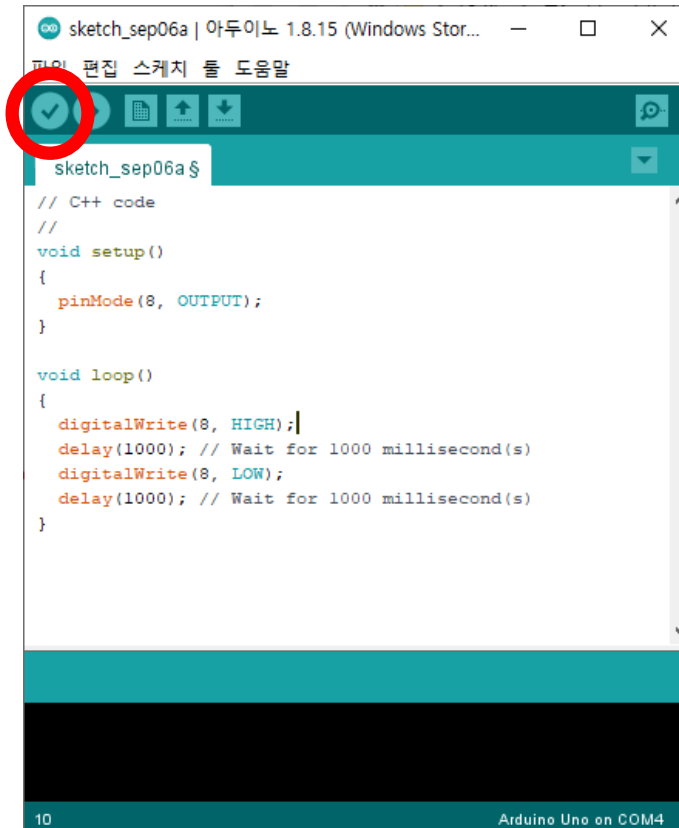


# 아두이노 개발 환경 구성 및 LED 테스트

- Arduino LED ON/OFF 코드 작성
  - 컴파일

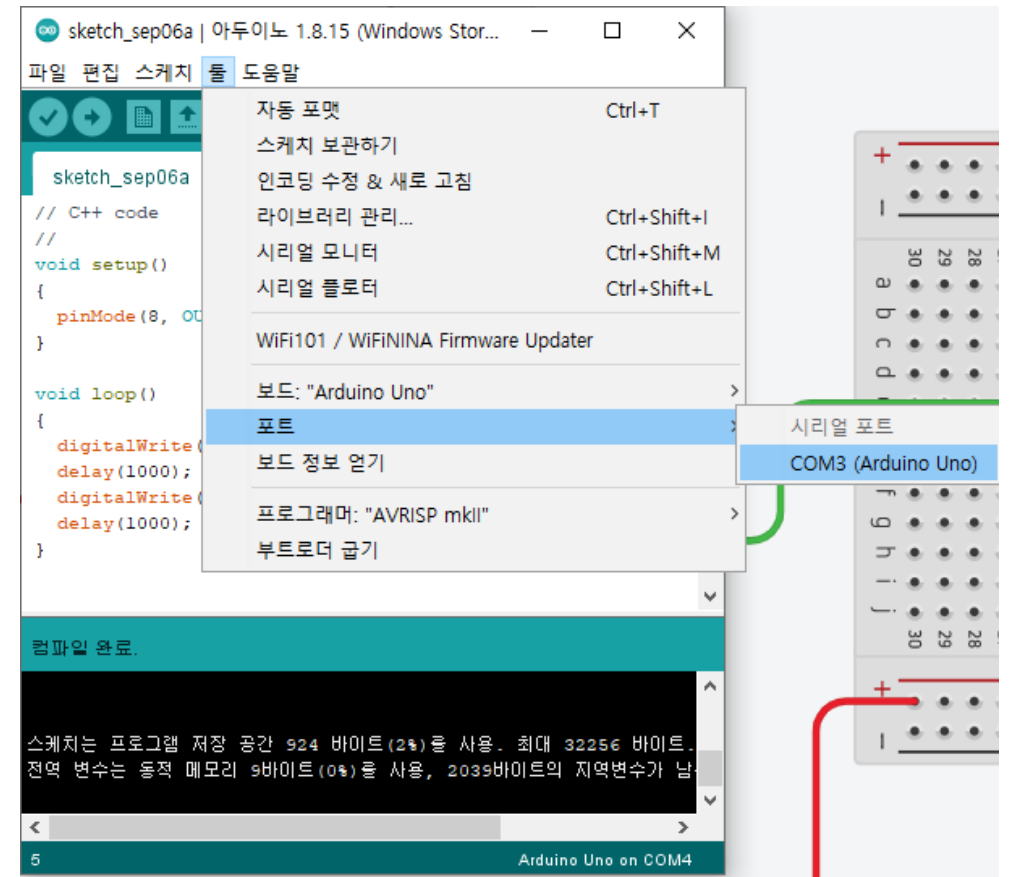
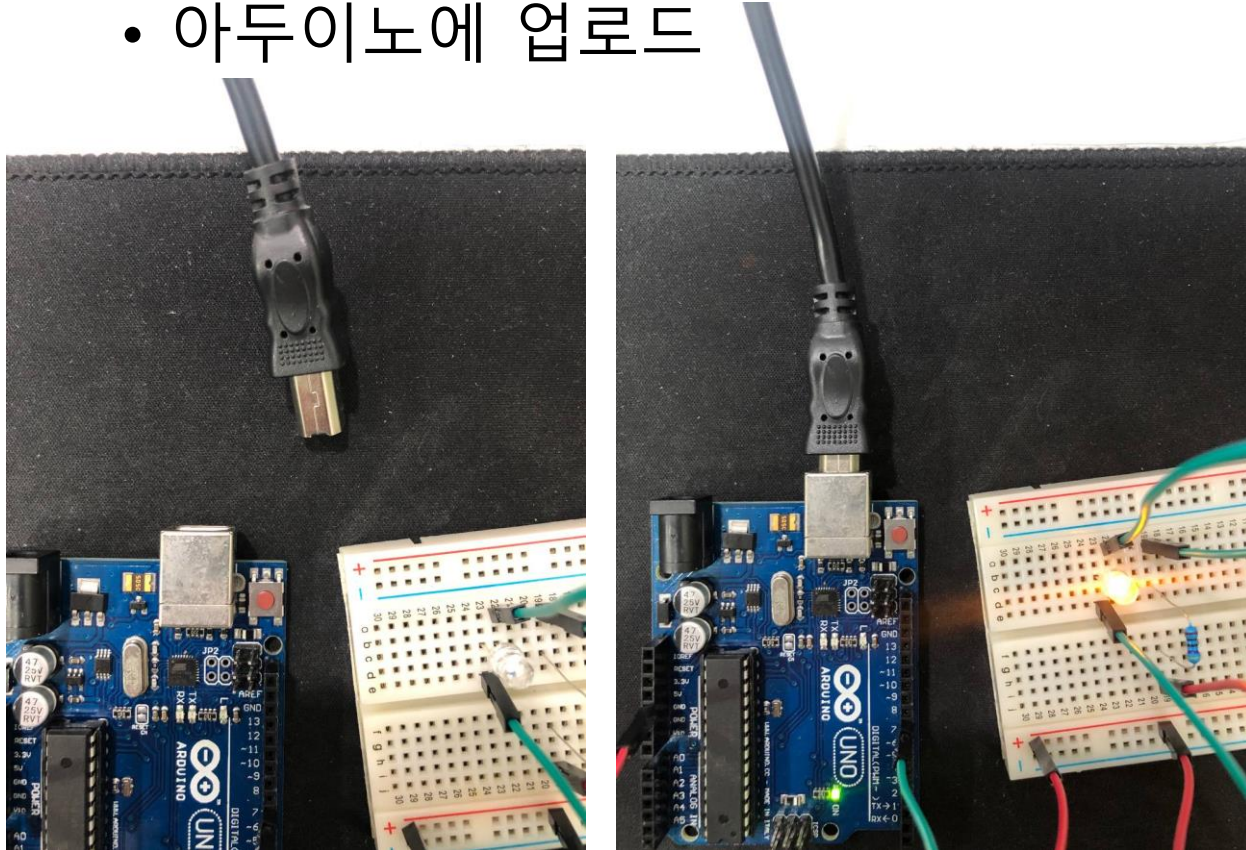
```
// C++ code
//
void setup()
{
  pinMode(8, OUTPUT);
}

void loop()
{
  digitalWrite(8, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(8, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```



# 아두이노 개발 환경 구성 및 LED 테스트

- Arduino LED ON/OFF 코드 작성
  - 아두이노에 업로드



# 아두이노 개발 환경 구성 및 LED 테스트

- Arduino LED ON/OFF 코드 작성
  - 아두이노에 업로드

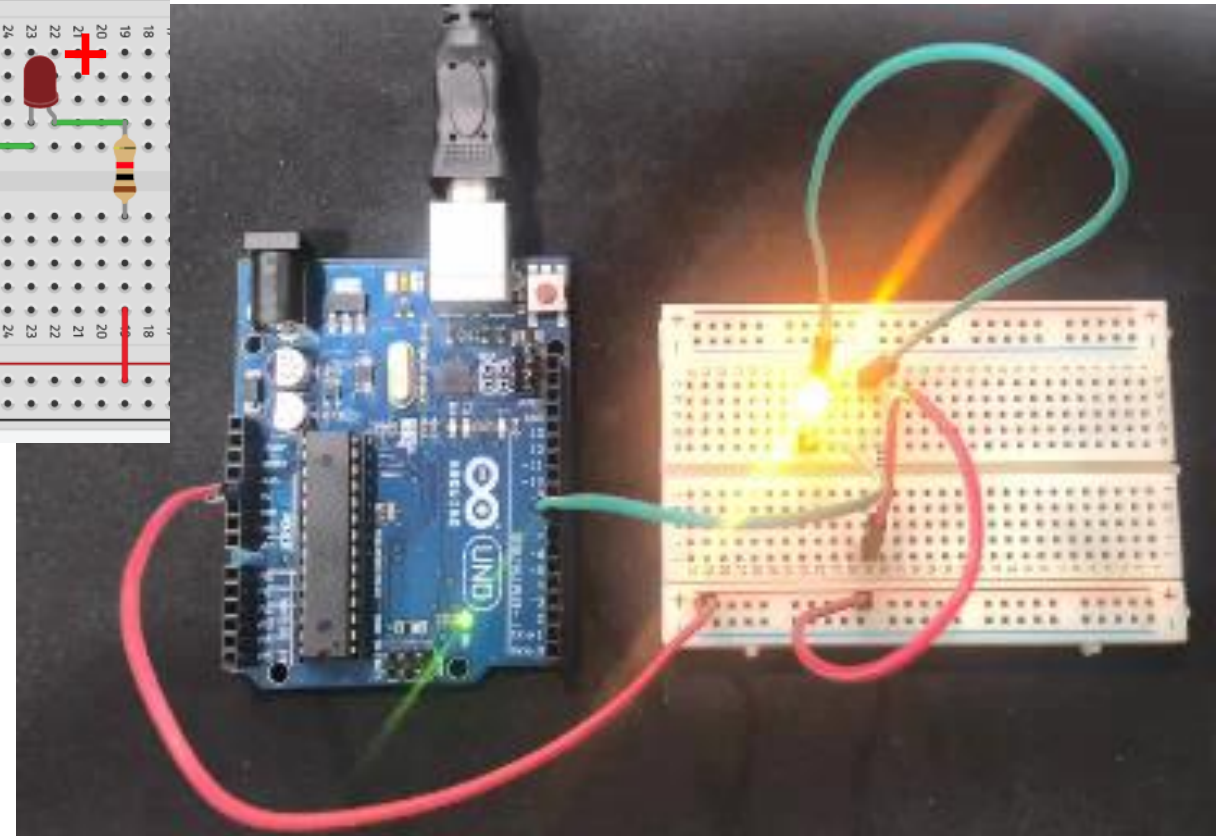
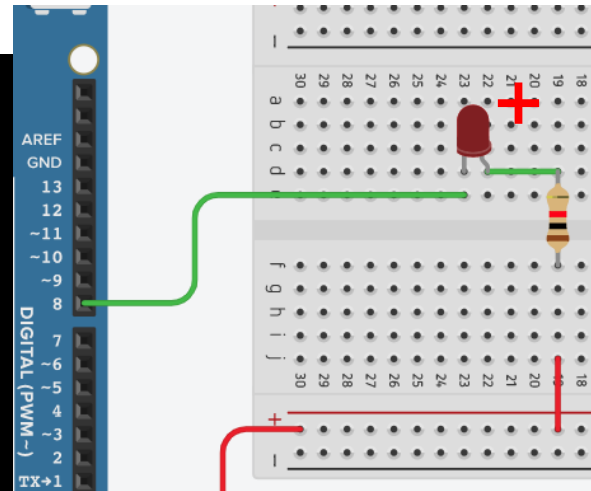


# 아두이노 개발 환경 구성 및 LED 테스트

- Arduino LED ON/OFF 실행

```
// C++ code
//
void setup()
{
  pinMode(8, OUTPUT);
}

void loop()
{
  digitalWrite(8, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(8, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```





# 아두이노 C언어 기본 함수, 기본 문법

- setup 함수

- 아두이노 보드에 전원이 인가되거나 리셋버튼이 눌리면 처음 실행될때 처음 호출되는 함수로서 아두이노 혹은 주변장치의 초기화용 코드를 작성한다.

```
void setup()
```

```
{  
}
```

```
void loop()
```

```
{  
}
```

- loop 함수

- setup함수 호출 이후에 주기적으로 반복 호출하는 함수로서 아두이노가 처리해야 하는 메인 제어 코드를 작성한다.

```
void setup()
```

```
{  
}
```

```
void loop()
```

```
{  
}
```

# 시리얼 통신

- 아두이노 <-> PC와 정보를 주고 받기 위한 가장 편리한 방법

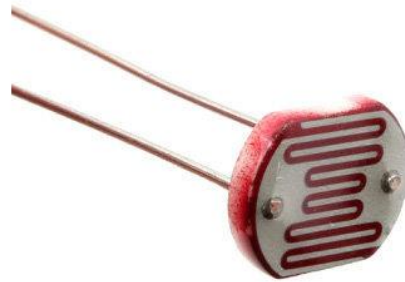
```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  Serial.println("Hello, World");
  delay(1000);
}
```



# 아두이노 광(빛)센서 실험

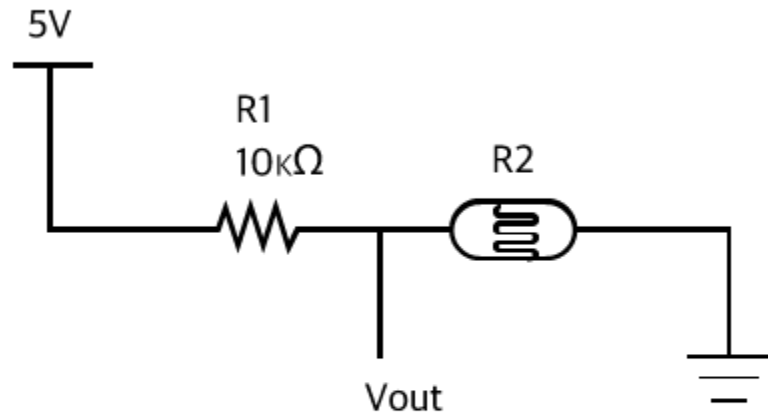
# 조도센서(CDS cell)



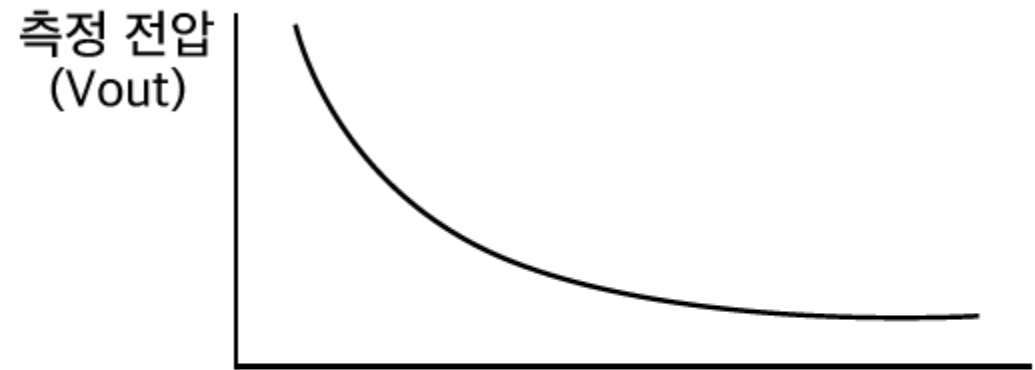
# 조도센서(CDS cell)

- 특징
  - 빛의 양(조도)에 따라 저항값이 변화(밝기값과 저항값은 반비례)
  - 극성이 없음(+,-가 없음)
  - 아날로그 입력
- 조도센서를 이용하여
  - 스마트 가로등 : 가로등을 자동으로 On/Off

# 조도 센서(CDS cell)

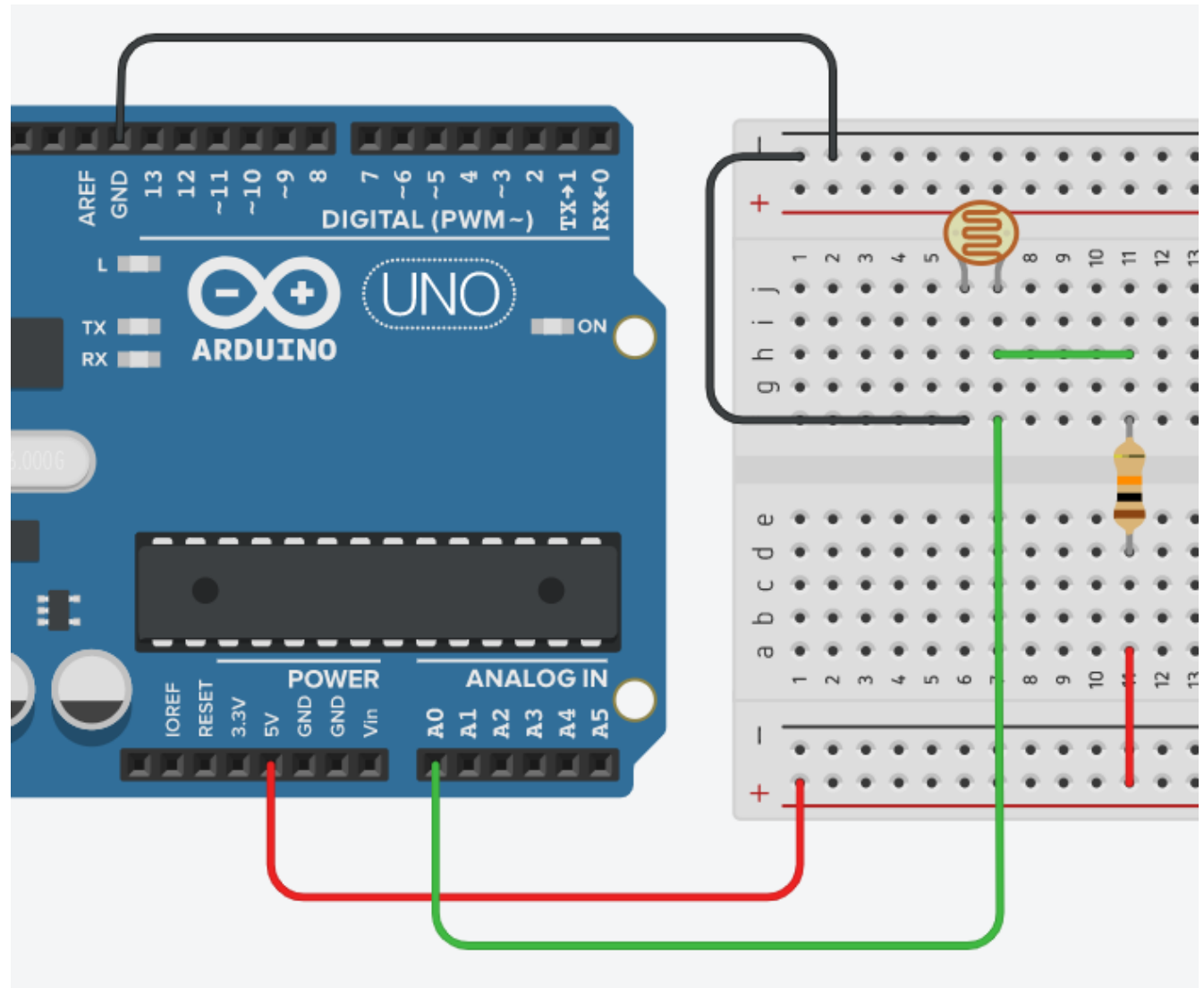
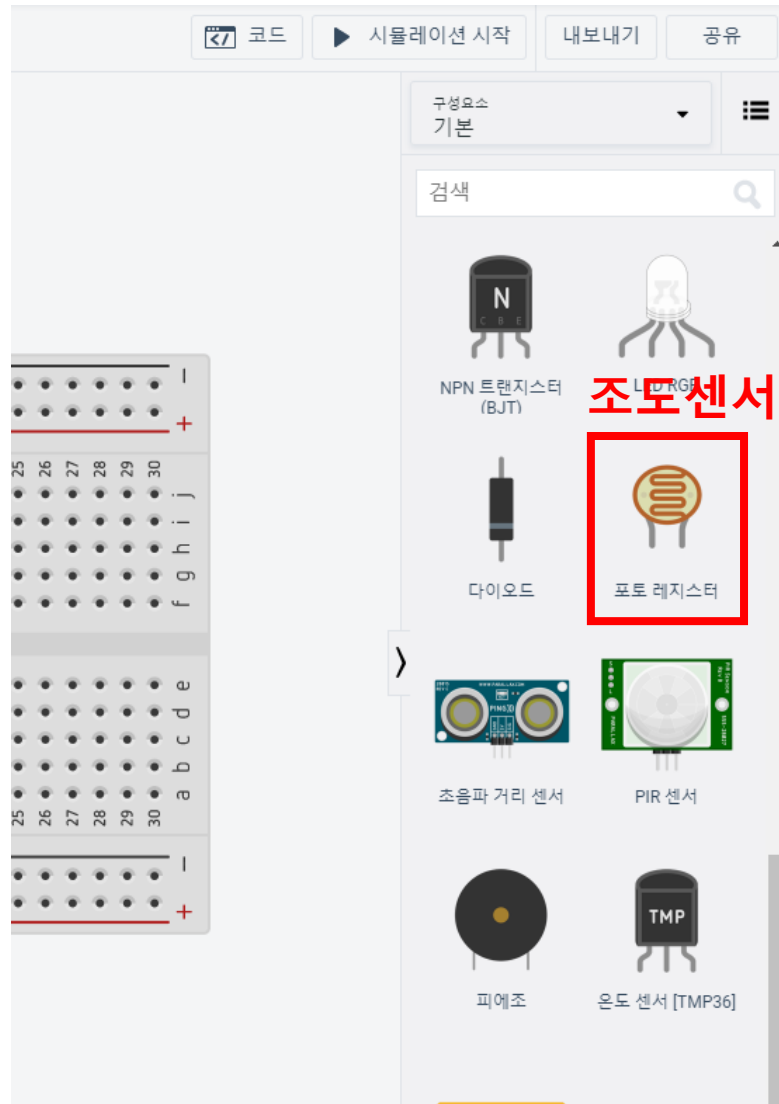


풀업 저항 사용



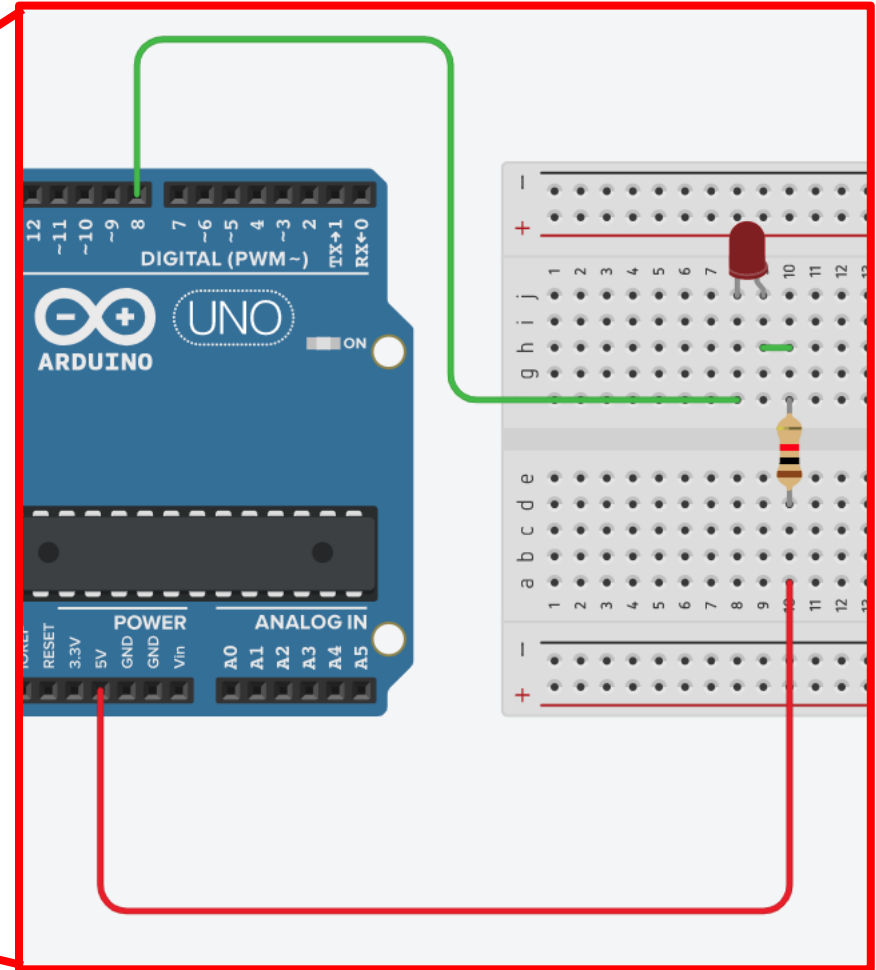
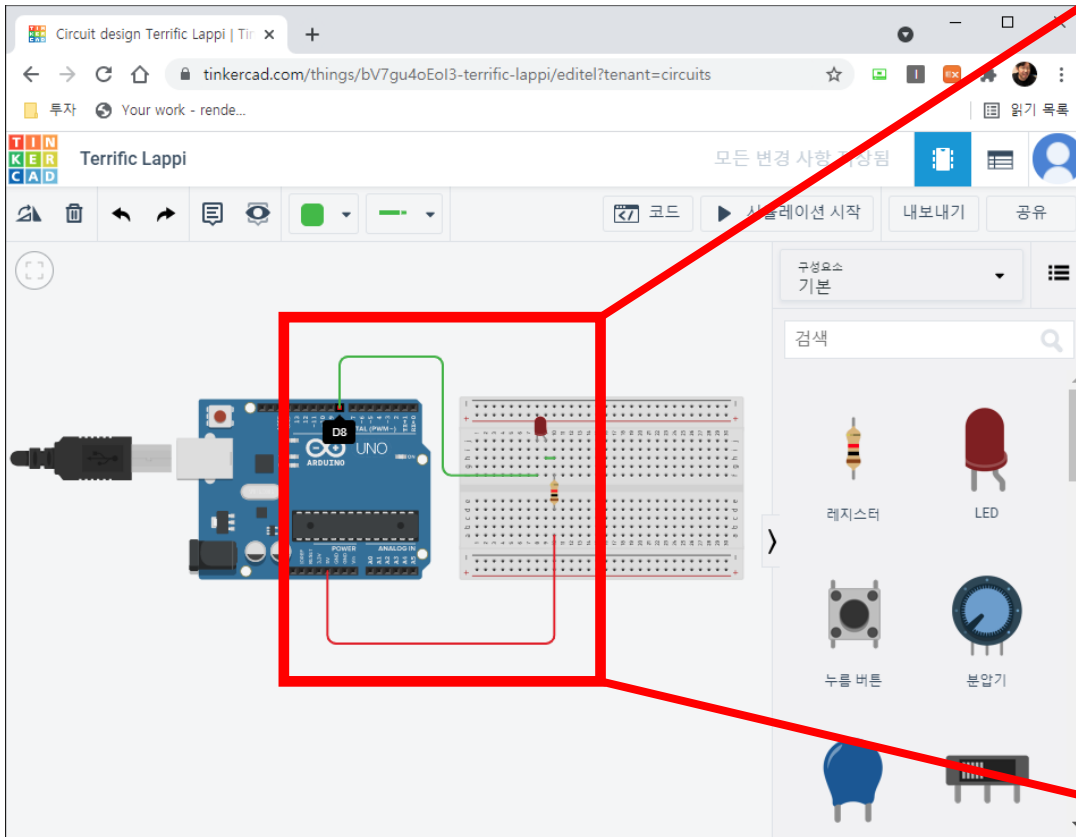
풀업 저항 사용시 밝기에 대한 측정 전압

# 아두이노를 활용한 조도센서 실험



# 아두이노를 이용한 LED 실험

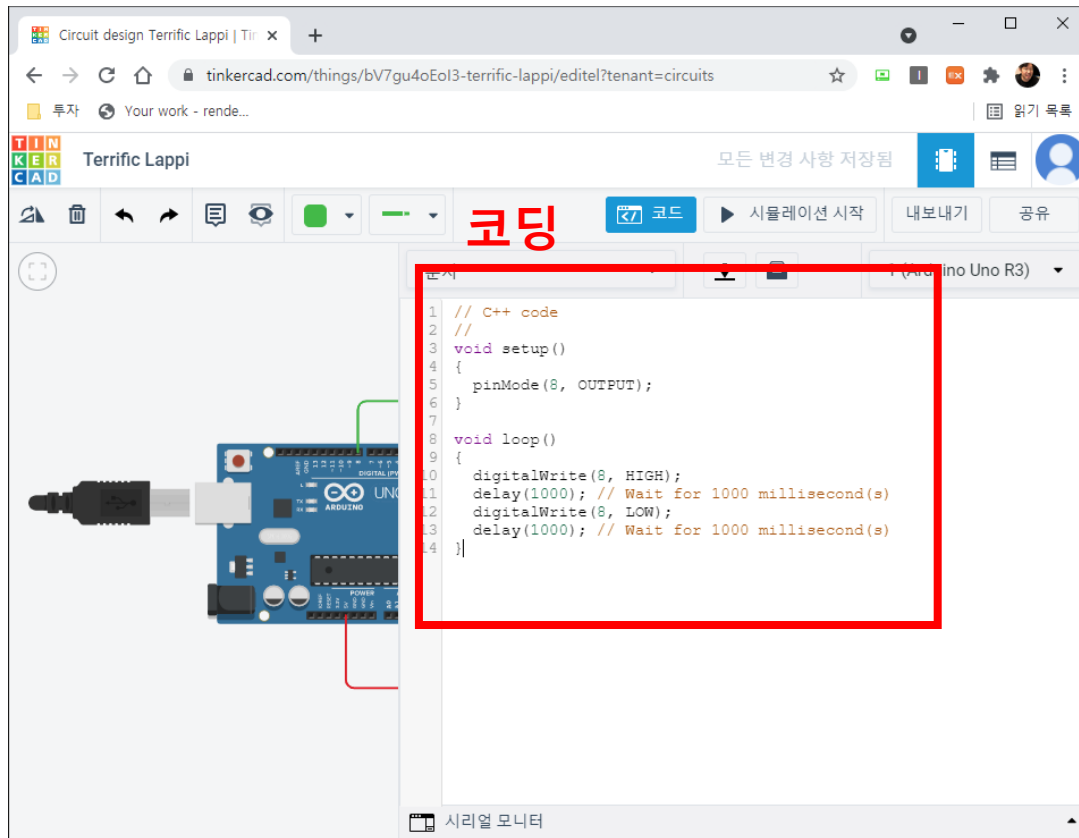
- tinkercad.com : 전체 회로 구성





# 아두이노를 이용한 LED 실험

- tinkercad.com : 코드 작성

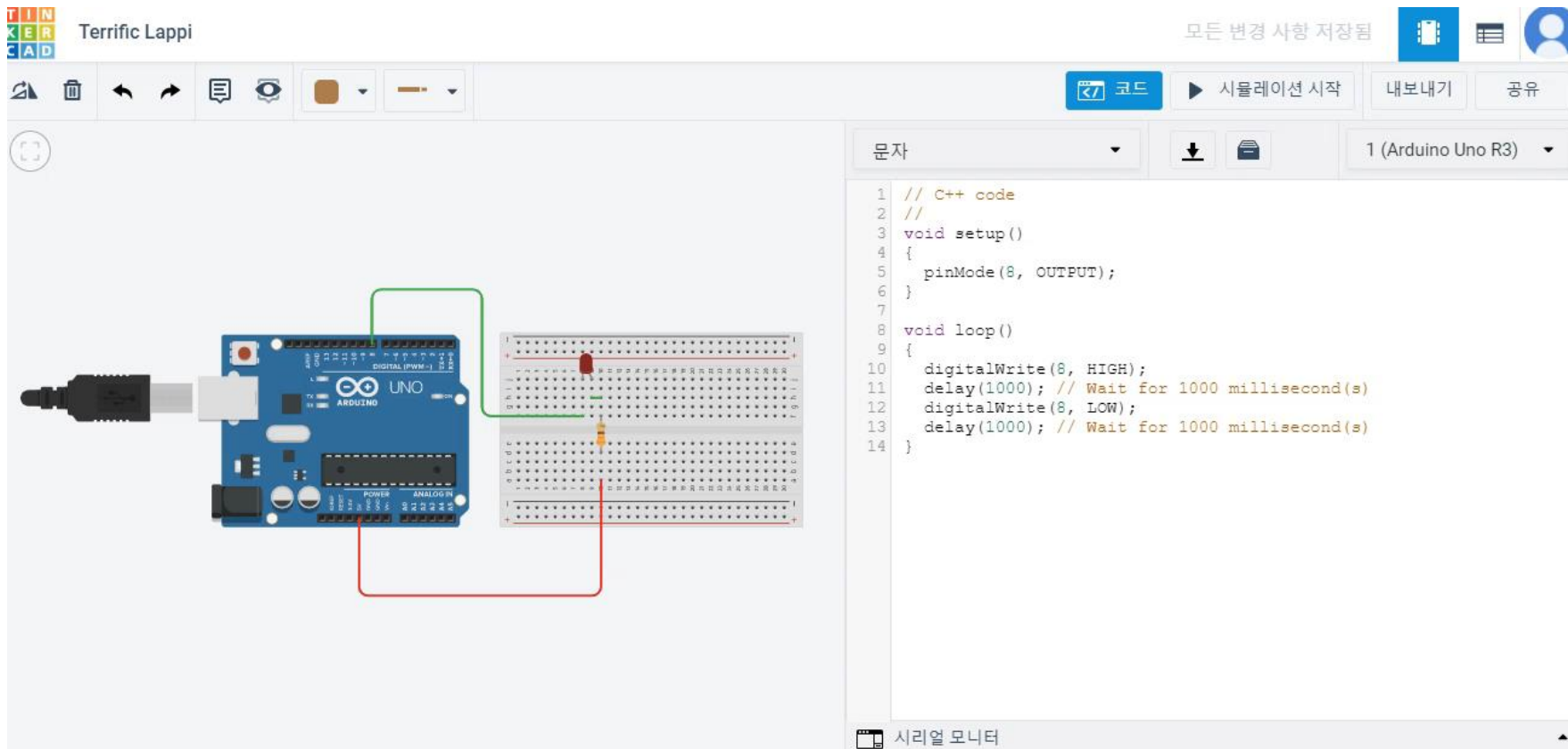


```
// C++ code
//
void setup()
{
  pinMode(8, OUTPUT);
}

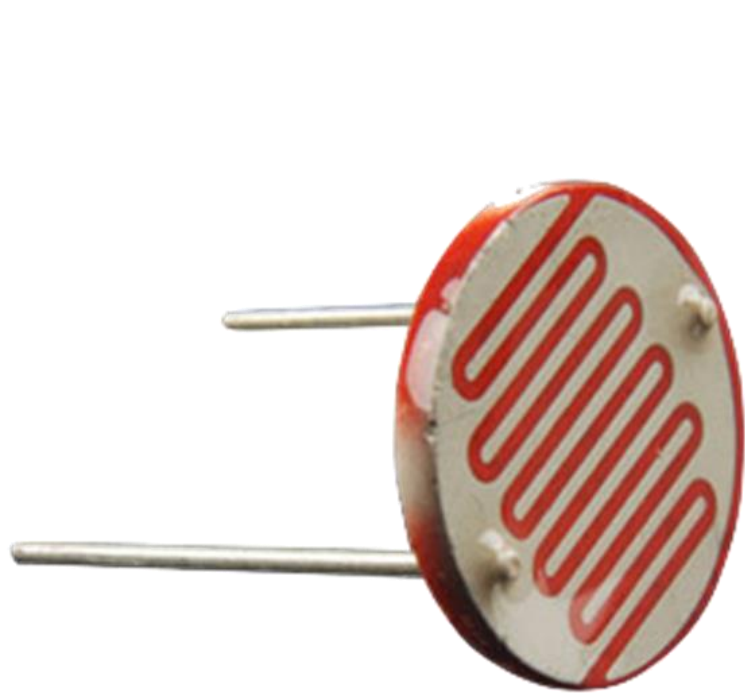
void loop()
{
  digitalWrite(8, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(8, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

# 아두이노를 이용한 LED 실험

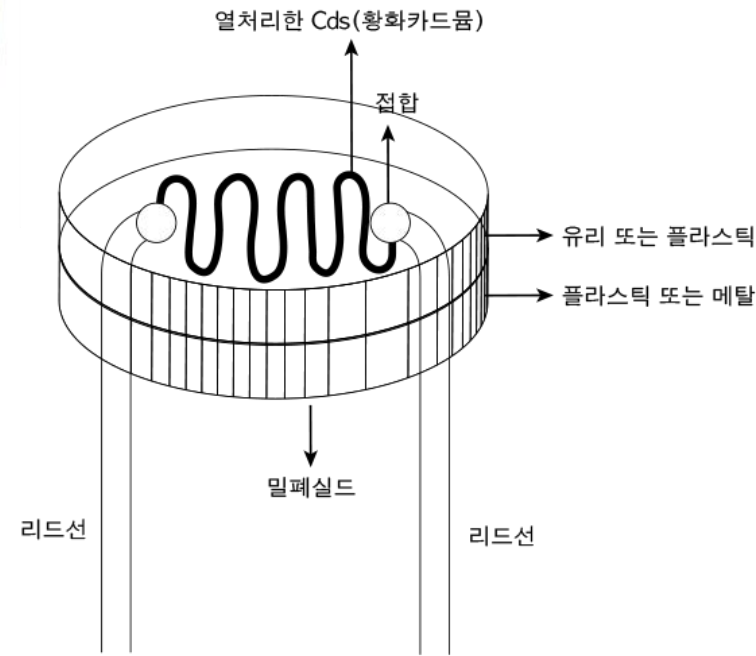
- tinkercad.com : 시뮬레이션 시작



# 스마트 가로등 제작



조도센서(Cds)



조도센서(Cds) 구조도