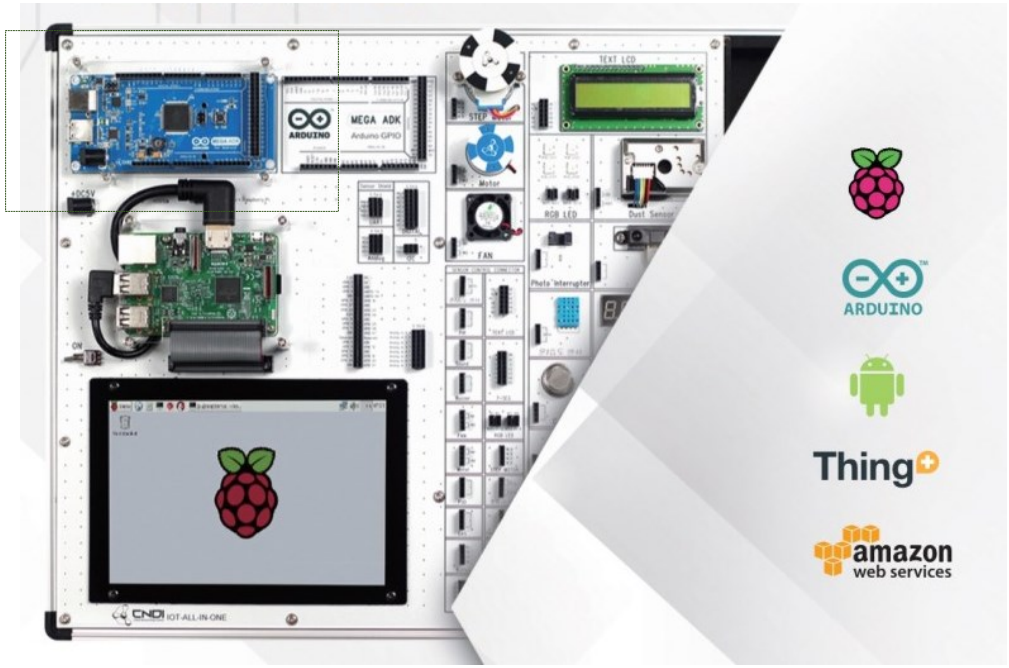
마이크로프로세서
(강의자료 #4)



교과목명 : 마이크로프로세서 (01)

담당교수 : 이 수 형

E-mail : soohyong@uu.ac.kr

교재명 : 스마트기기 개발을 위한 아두이노, 이수형. (LINC+ 사업단 배포)

수강생 안내

- 본 교과목은 기본적으로 “대면”수업입니다.
- ‘코로나바이러스감염증-19’ 등의 이유로 대면수업을 받지 못하는 학생들을 위해서 ‘비대면 실시간 수업’을 병행합니다. 기본적으로 대면 수업이므로 대면 수업에 참석한 학생들은 교실에서 출석을 체크하면 되며, 참석하지 못하고 ‘비대면 수업’을 듣는 학생들은 ‘실시간 온라인 강의’에 참석하면 출석이 반영됩니다.
- 수강생 여러분들은 강의 자료를 온라인 배포 및 게재 등의 행위를 할 시 저작권 문제가 발생할 수 있사오니 이에 유념하여 강의 자료를 공유하는 행위는 삼가 바랍니다.
- 교재는 “스마트기기 개발을 위한 아두이노”이며, 나누어준 키트는 각자 잘 관리하면서 실습실(대면) 및 집(비대면)에서 실험/실습을 수행할 수 있도록 진행합니다.

지난시간에는?

아두이노란

- 아두이노란
  - 1995년 이탈리아에서 만든 초보자를 위하여 제작한 하드웨어 제작용 마이크로 컨트롤러 보드
  - 하드웨어 및 소프트웨어를 오픈소스화 ⇒ 전세계적으로 확산, 다양한 모듈
- 아두이노의 종류
  - Uno : 가장 기본적인 보드
  - Mega : Uno의 확장판 (많은 입/출력 포트 및 늘어난 메모리)
  - Leonardo : 키보드/마우스 제품을 위한 보드
  - Due : 32비트 MPU
  - Nano / Pro mini : 소형화된 버전, 제품 개발용

아두이노 개발

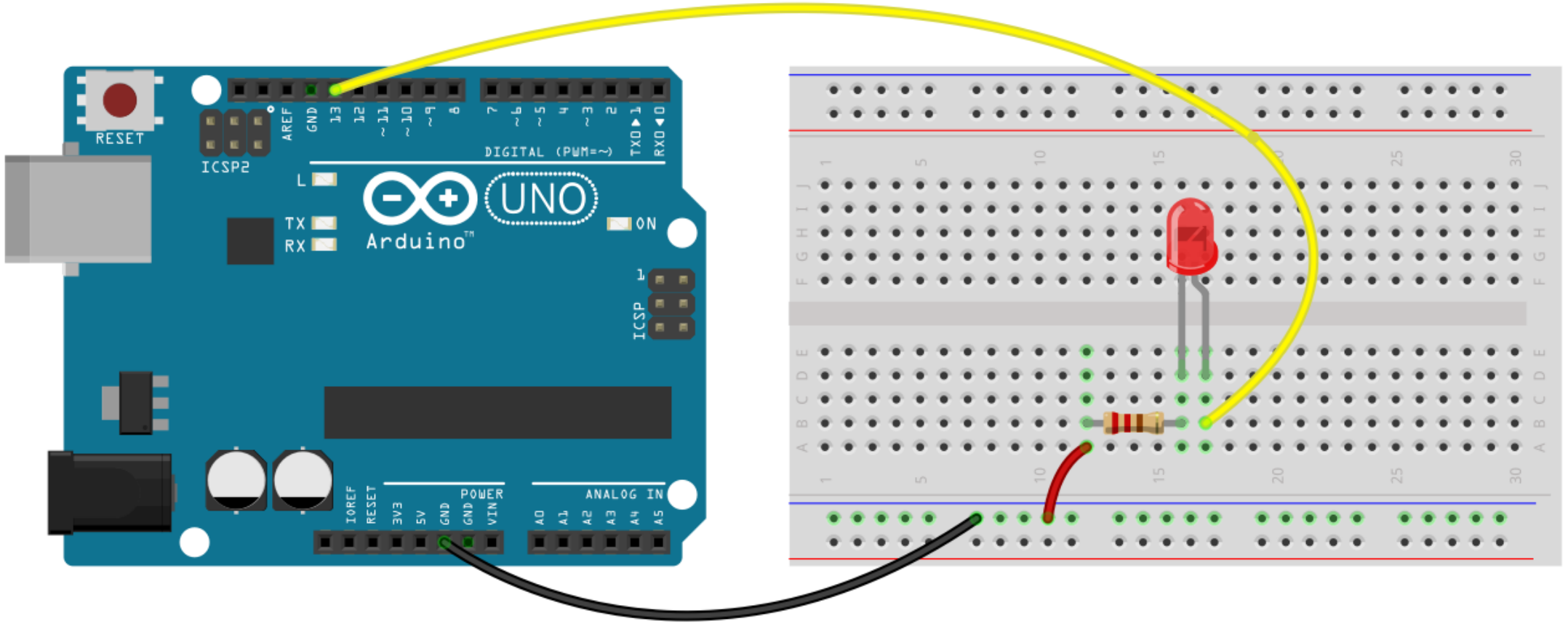
- 아두이노 소프트웨어 개발환경
  - C/C++을 이용한 통합개발환경(IDE) 제공

- 스케치 구성
  - 스케치 : 아두이노의 소스 프로그램
  - 일반적인 C++과 다르게 setup() 과 loop()의 두 함수로 구성
    - ➢ setup() : 처음 한번만 수행, loop() : 무한히 반복해서 수행
  - 디지털 입/출력 설정 : pinMode()
  - 디지털 출력 : digitalWrite()
  - 시간 지연 (기다리기) : delay()

4.1 LED를 이용한 디지털 신호 출력

Review

- 회로 구성



- 스케치

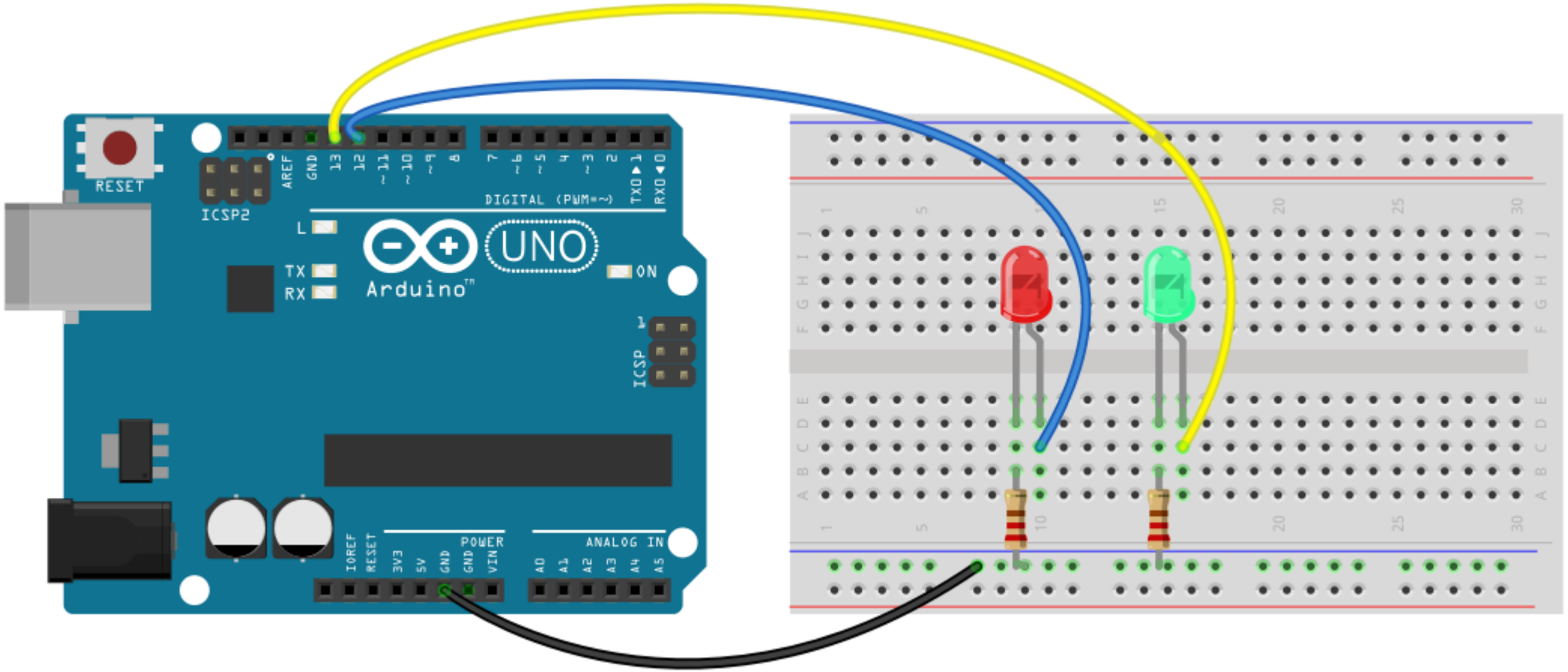
예제 4-1. 첫번째 스케치

```
int led = 13;

// 처음 시작시 한번만 수행하는 함수, 설정을 담당한다
void setup() {
    pinMode(led, OUTPUT);      // 13번 핀을 출력으로 설정한다
}

// 반복해서 호출되는 함수
void loop() {
    digitalWrite(led, HIGH);  // 13번 핀으로 5V 디지털 신호를 출력한다
    delay(1000);              // 1000 ms = 1초를 기다린다.
    digitalWrite(led, LOW);    // 13번 핀으로 0V 디지털 신호를 출력한다
    delay(1000);              // 1초를 기다린다.
}
```

4.2 두개의 LED를 제어하기

Review

- 회로 구성



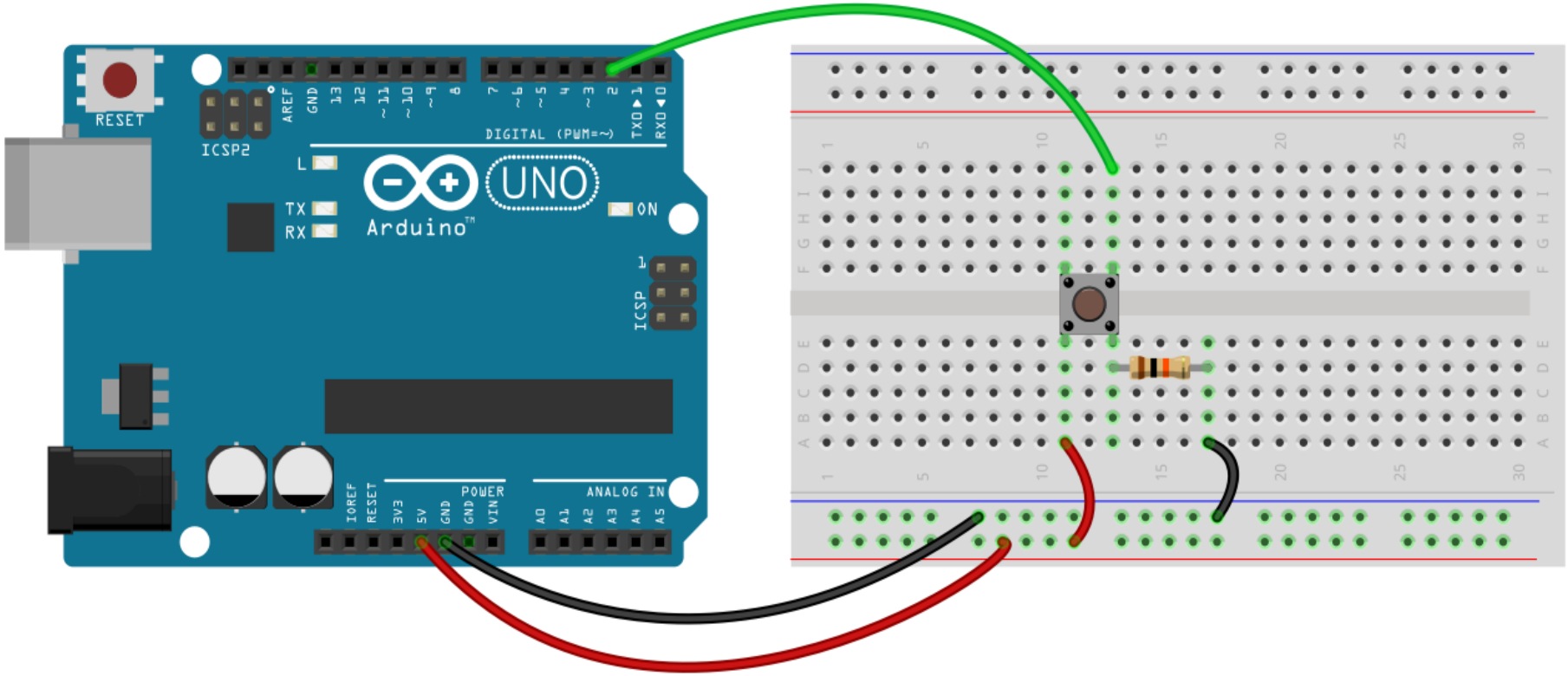
예제 4-2. 두 개의 LED 순차 점등

```
int ledA = 13;
int ledB = 12;

// 처음 시작시 한번만 수행하는 함수, 설정을 담당한다
void setup() {
    pinMode(ledA, OUTPUT);      // 13번 핀을 출력으로 설정한다
    pinMode(ledB, OUTPUT);      // 12번 핀을 출력으로 설정한다
}

// 반복해서 호출되는 함수
void loop() {
    // LED 1만 켜기
    digitalWrite(ledA, HIGH);   // LED 1 켜기
    digitalWrite(ledB, LOW);    // LED 2 끄기
    delay(1000);                // 1초를 기다린다.
    // LED 2만 켜기
    digitalWrite(ledA, LOW);    // LED 1 끄기
    digitalWrite(ledB, HIGH);   // LED 2 켜기
    delay(1000);                // 1초를 기다린다.
}
```

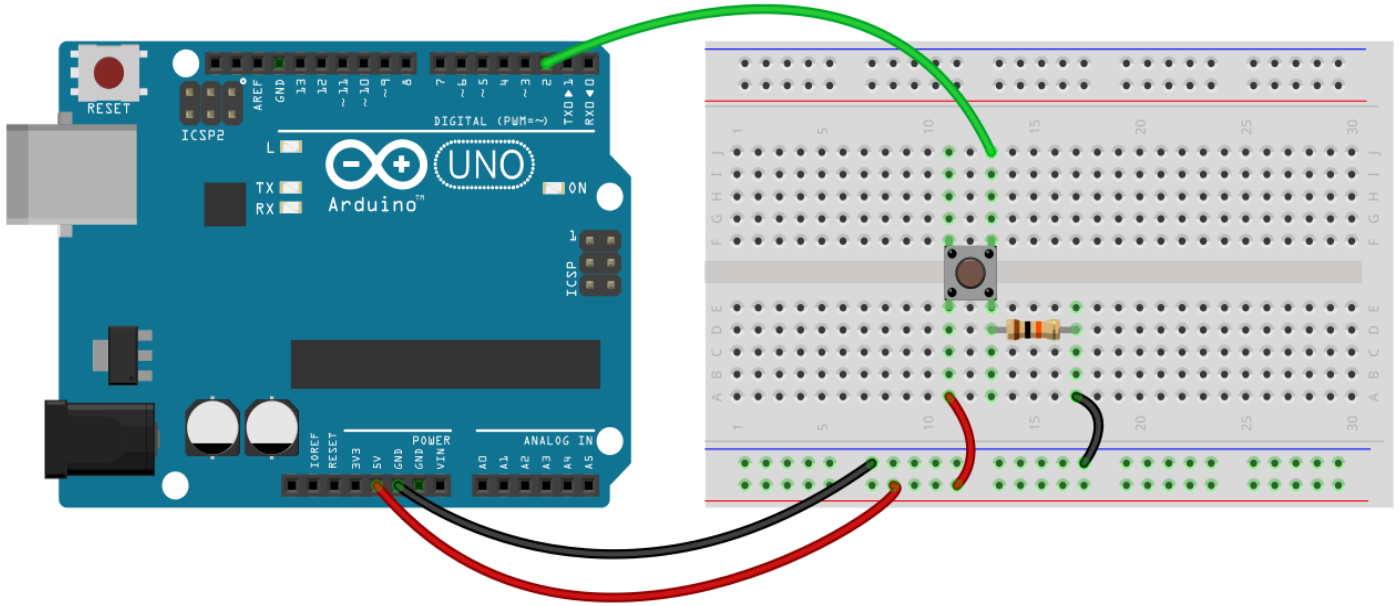
4.3 스위치를 이용한 디지털 입력

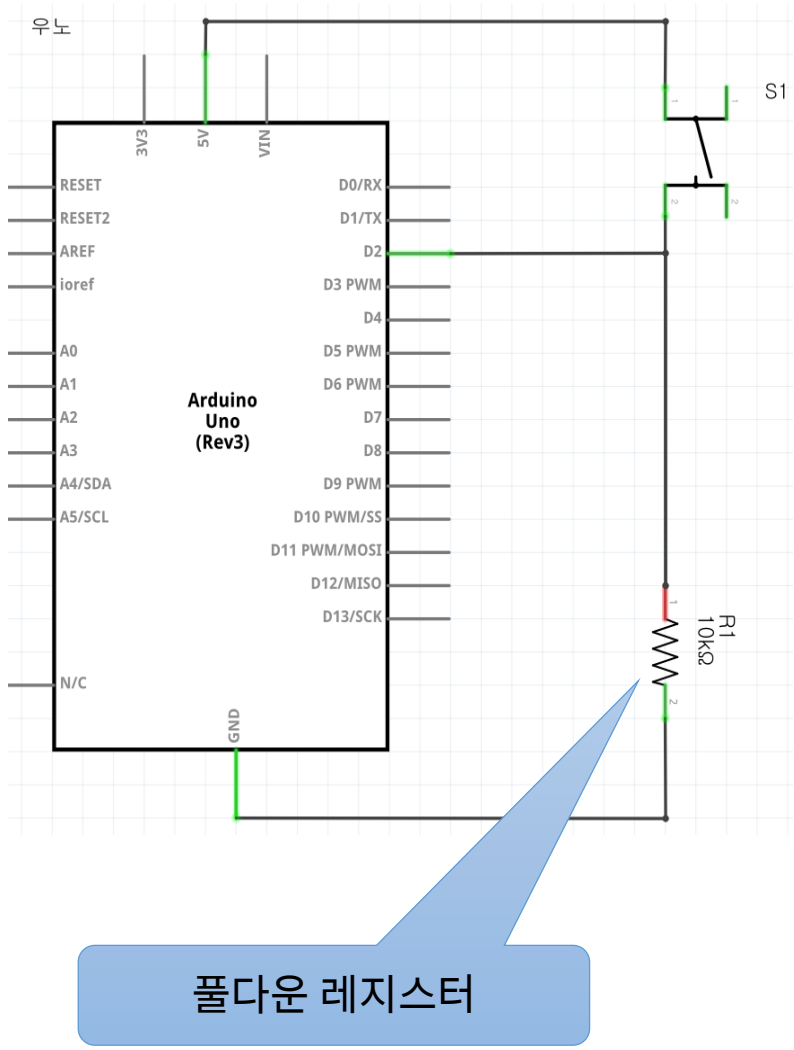
Review

- 택트 스위치를 이용하여 입력받음



- 회로도
  - 저항의 역할 : 풀다운(pull-down) 저항
  - 스위치가 Off인 경우에 GND로 연결
  - 스위치 On인 경우에는 5V 인가

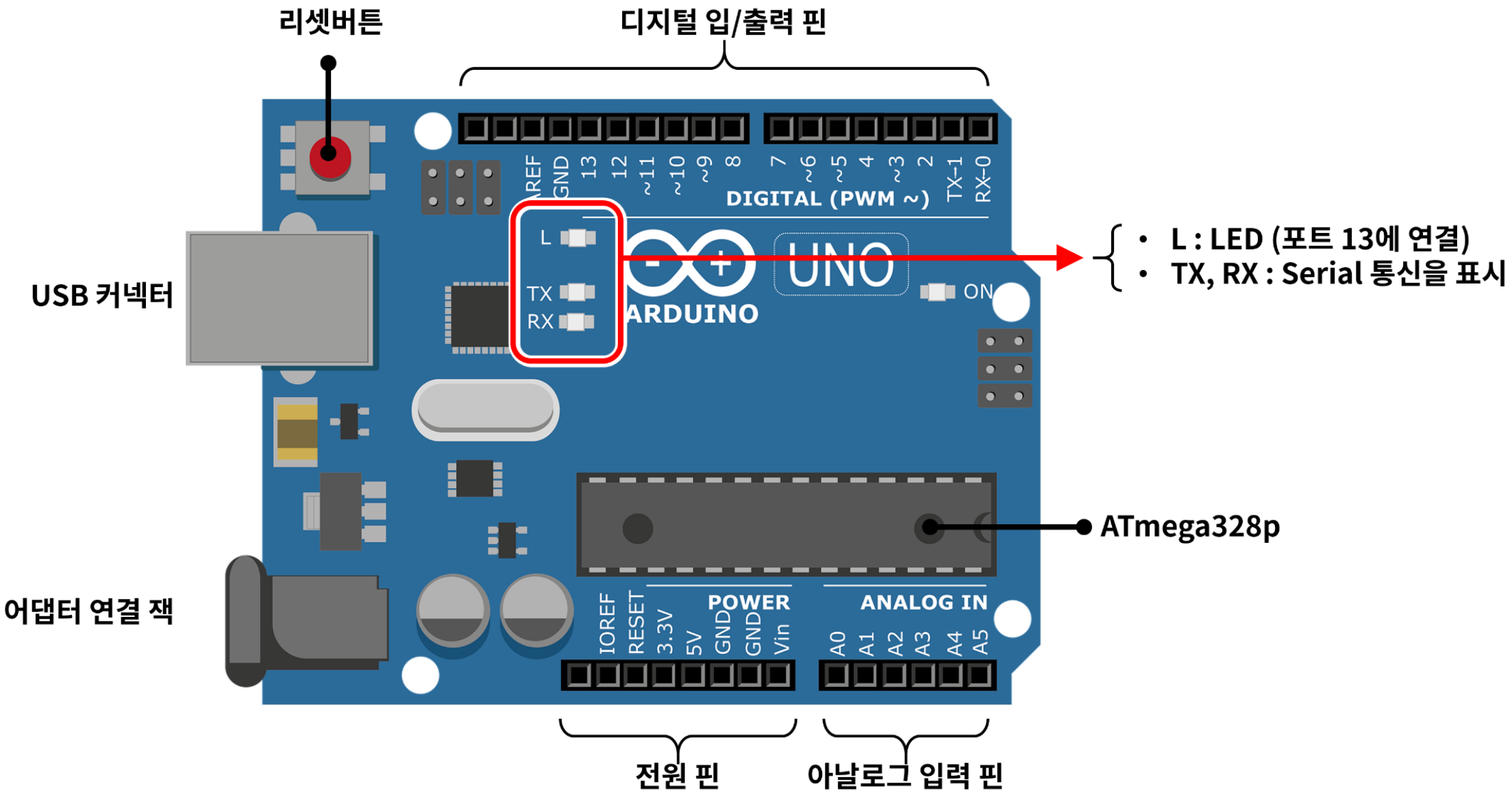




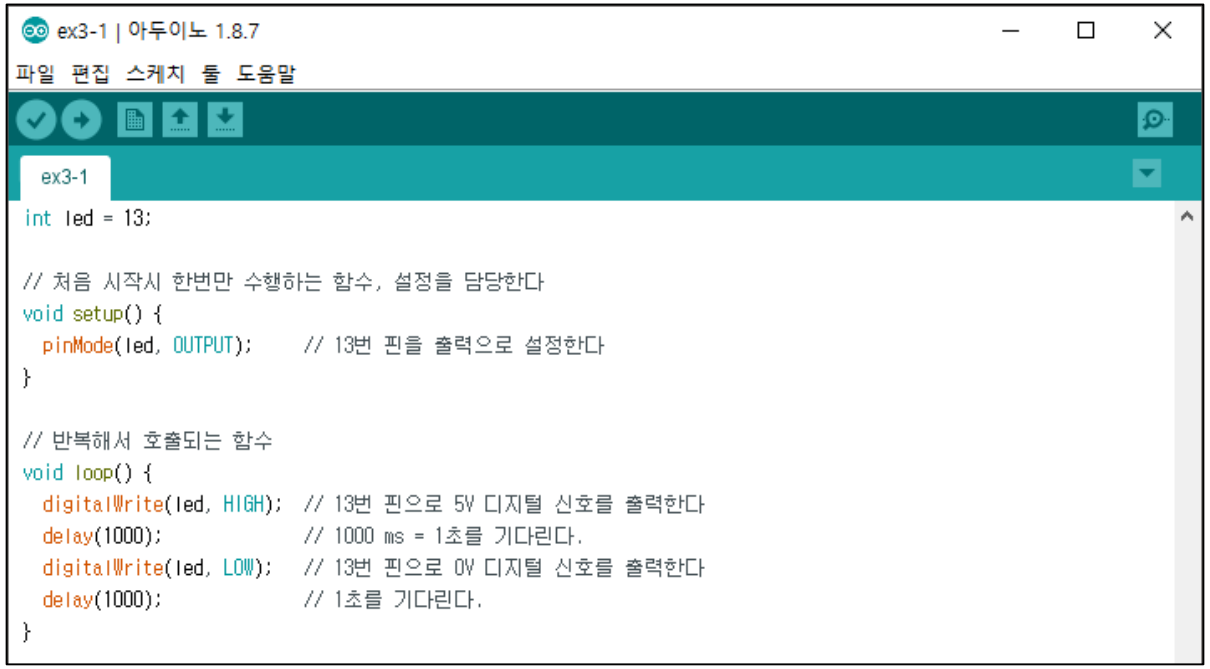
예제 4-3. 디지털 입력 프로그램

```
// 디지털 입/출력 핀 2번을 스위치의 입력으로 설정한다.
int pushButton = 2;

void setup() {
  // 시리얼 통신을 위하여 초기화하는 과정, 9600보레이트(baud-rate)로 설정한다.
  Serial.begin(9600);
  // 버튼 스위치가 연결된 2번 핀을 입력 모드로 설정한다.
  pinMode(pushButton, INPUT);
}

void loop() {
  // 현재 버튼이 연결되어 있는 핀의 값을 읽어들인다.
  int buttonState = digitalRead(pushButton);
  // 입력받은 값을 시리얼 통신을 통해서 PC로 전송한다.
  Serial.println(buttonState);
    // 입력 상태를 안정화하기 위하여 1미리초 동안 대기한다
  delay(1);
}
```

아두이노 우노 보드


리셋버튼
디지털 입/출력 핀
USB 커넥터
어댑터 연결 잭
• L : LED (포트 13에 연결)
• TX, RX : Serial 통신을 표시
ATmega328p
전원 핀
아날로그 입력 핀
DIGITAL (PWM ~)
POWER
ANALOG IN
UNO
ARDUINO
ON
L
TX
RX

4. 디지털 입출력의 기초 #2

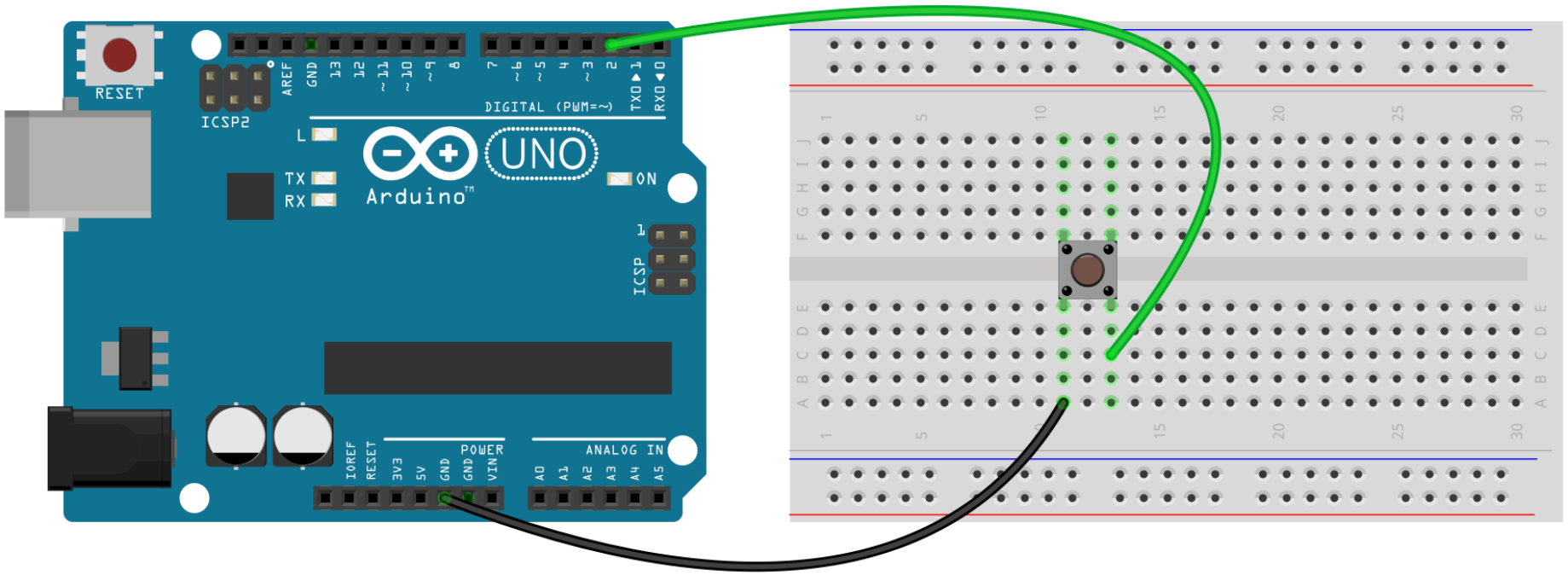


풀업/풀다운

- 풀업 (pull-up) / 풀다운 (pull-down)
  - 없는 경우 : floating 상태 (High impedance) ⇒ 전압이 결정되지 않음
  - 풀업 저항 : 스위치 오픈시에 5V로 입력 전압 유지
  - 풀다운 저항 : 스위치 오픈시에 0V(GND)로 입력 전압 유지

Internal pull-up register

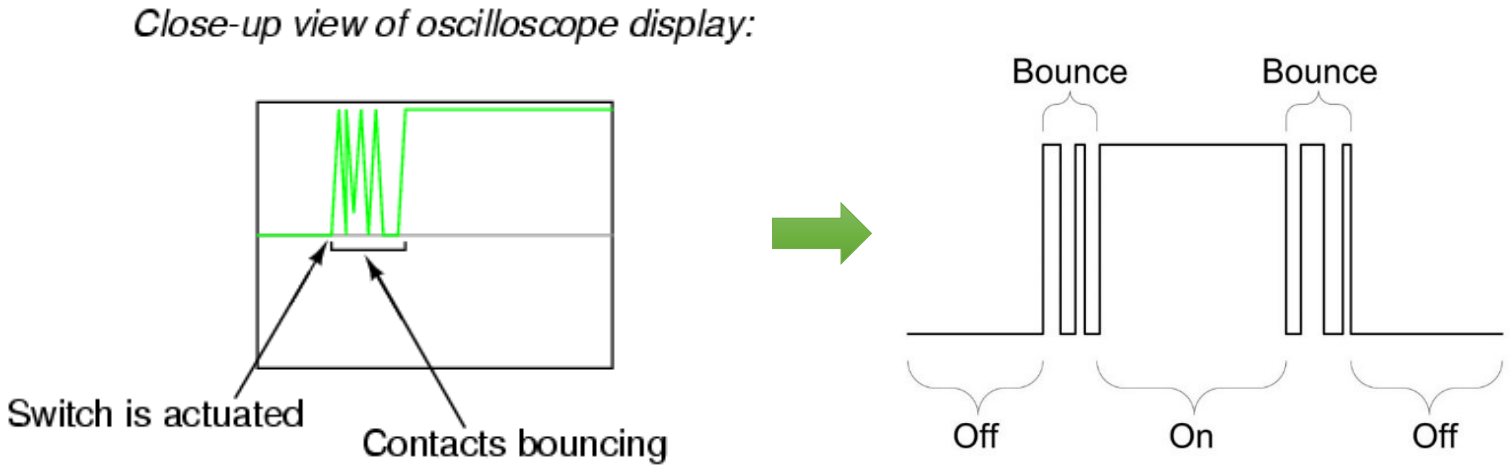
- 아두이노 내부에 존재하는 풀업 저항
  - `pinMode(INPUT)` 대신 `pinMode(INPUT_PULLUP)` 사용
  - 외부에 별도의 풀업 저항이 필요 없음 ⇒ 회로가 단순해짐
  - 풀다운 → 풀업 : 스위치 On/Off시에 digitalRead()의 결과가 바뀜
    - ➢ On → LOW, Off → HIGH



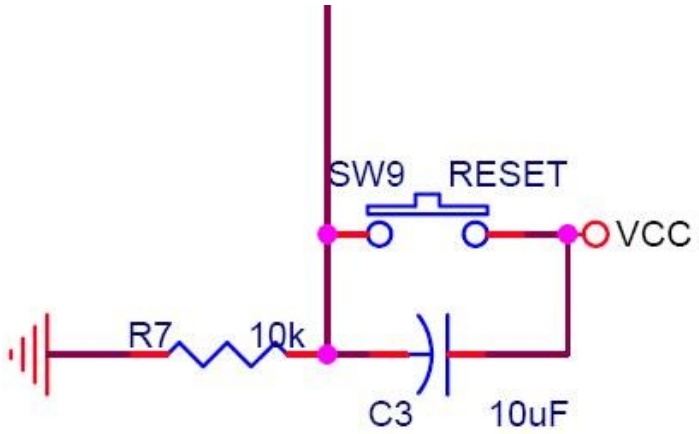
예제 4-3. 디지털 입력 프로그램 (pull-up)

```
// 디지털 입/출력 핀 2번을 스위치의 입력으로 설정한다.
int pushButton = 2;

void setup() {
  // 시리얼 통신을 위하여 초기화하는 과정, 9600보레이트(baud-rate)로 설정한다.
  Serial.begin(9600);
  // 버튼 스위치가 연결된 2번 핀을 입력 모드로 설정한다.
  pinMode(pushButton, INPUT_PULLUP);
}

void loop() {
  // 현재 버튼이 연결되어 있는 핀의 값을 읽어들인다.
  int buttonState = digitalRead(pushButton);
  // 입력받은 값을 시리얼 통신을 통해서 PC로 전송한다.
  Serial.println(buttonState);
    // 입력 상태를 안정화하기 위하여 1미리초 동안 대기한다
  delay(1);
}
```

- 채터링(chattering)
  - 스위치의 물리적인 접촉이 일어나는 경우 잡음이 발생

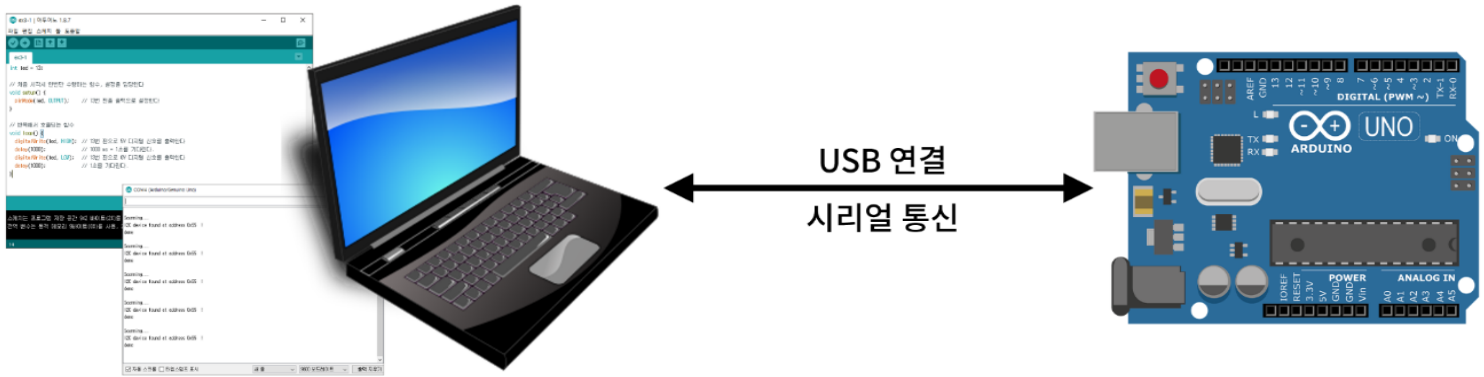


  - 채터링 방지 회로의 예
    - H/W : Capacitor를 사용한 회로 → 상태가 천천히 변함
      시간지연 발생 → 논리 gate를 사용한 회로
    - S/W 로 해결 : delay()함수 사용



UART 통신

- 시리얼 통신
  - 아두이노와 PC와의 통신
  - UART(Universal Asynchronous Receiver and Transmittor)
  - RS-232, RS-422, RS-485 등의 표준방식이 존재
  - 아두이노에는 디지털 포트 0번과 1번이 준비되어 있음
  - 추가 시리얼 통신은 소프트웨어로 해결

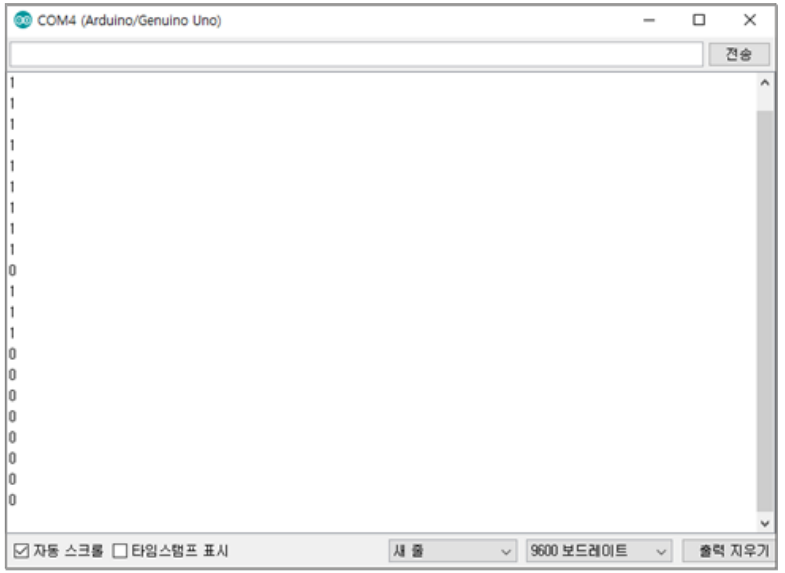


아두이노 IDE & 시리얼 모니터 Serial 객체 사용

- Serial 객체
  - PC와의 시리얼통신을 위해 사용함
  - setup()함수에서 초기화
    - `Serial.begin(9600);`
    - 9600 보레이트(baud-rate)로 설정
      - ✓ Baud-rate : 300, 1200, 2400, 4800, 9600, 19200, 38400, …
  - Serial.println(값);
    - 주어진 값을 PC로 전송 print + line (줄넘김 포함)
      - ✓ `Serial.println("Hello");`
      - ✓ `Serial.print(3.14);`
      - ✓ `Serial.print('A');`
    - 읽기 : Serial.available() 함수와 Serial.read() 함수를 사용

시리얼 입력

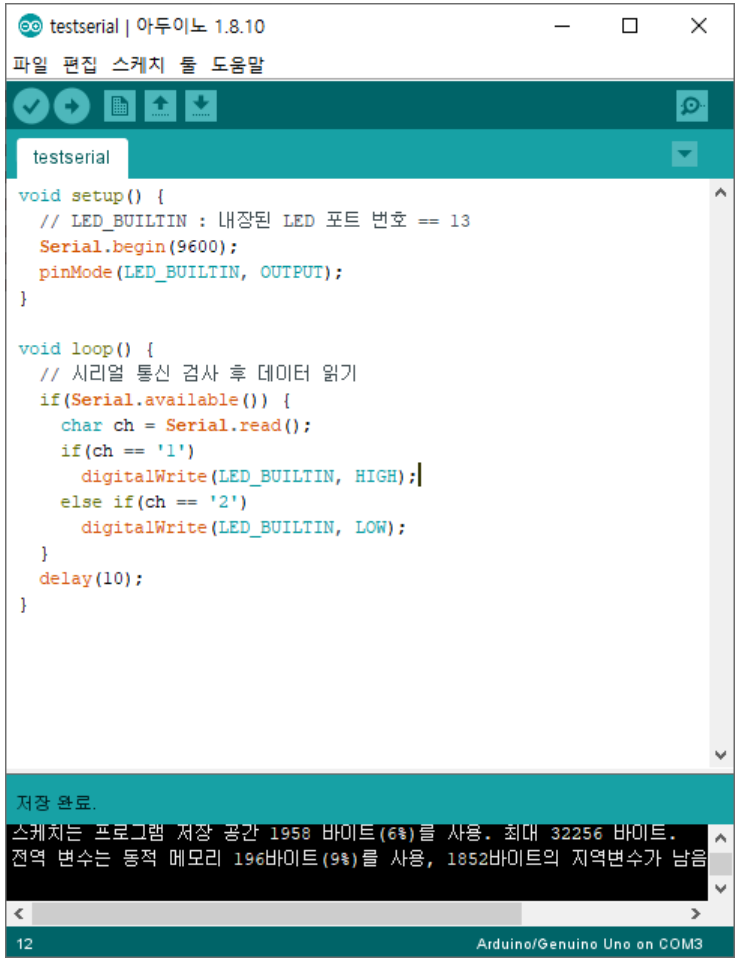
- Serial.available()
  - 현재 시리얼 포트에 새로운 값이 전송되었는지 판단하는 함수
  - 이 함수를 이용하여 새로운 값이 있는 경우만 읽어 들임
- Serial.read()
  - 시리얼 포트에 저장되어 있는 값을 읽어 들임

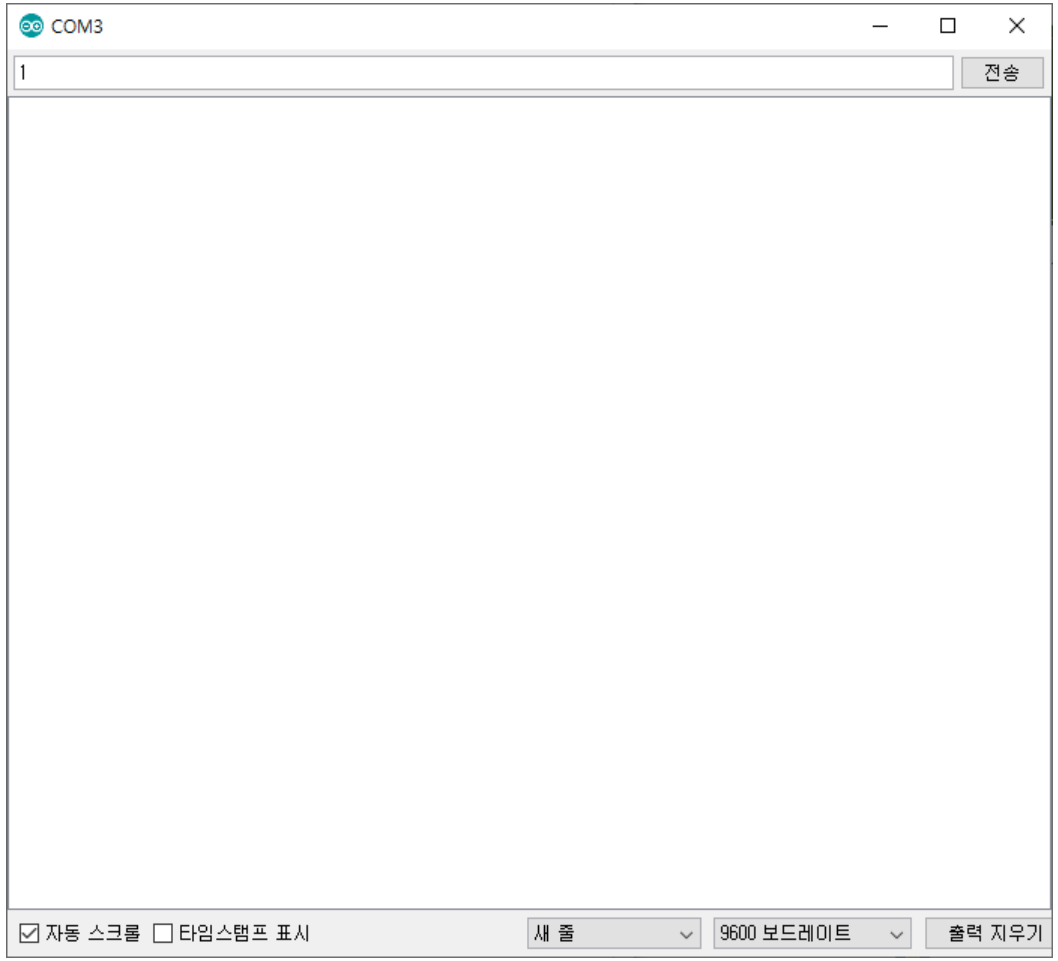


예제: 시리얼 입력을 통한 LED 제어

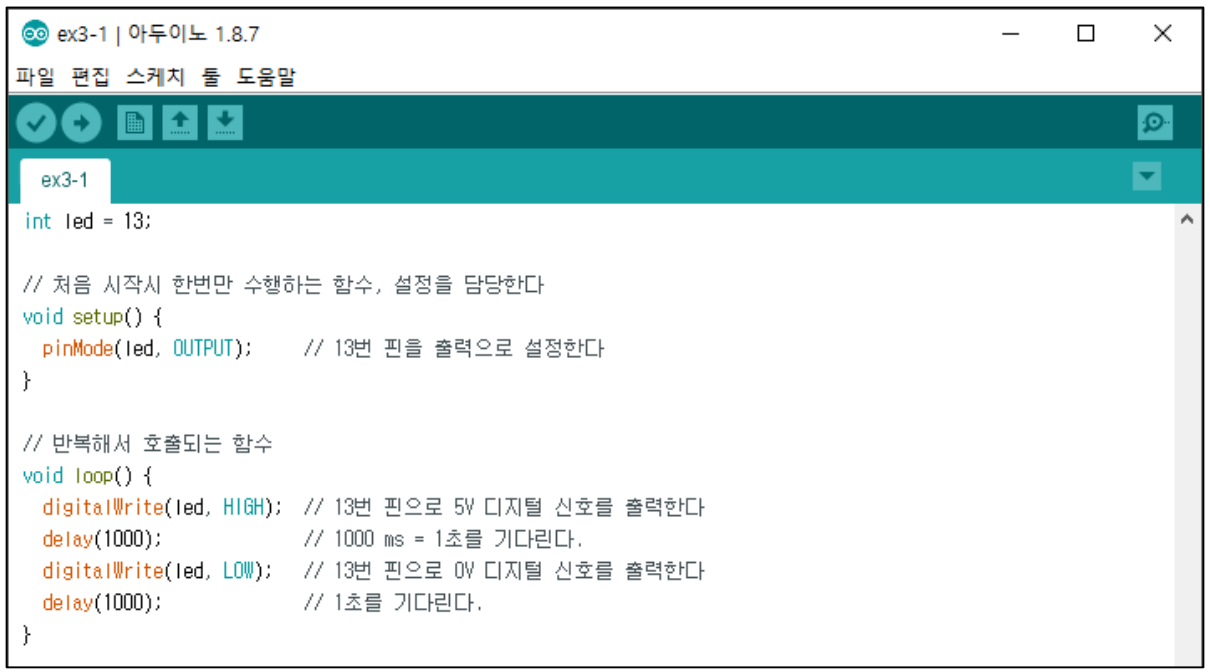
```
void setup() {
  // LED_BUILTIN : 내장된 LED 포트 번호 == 13
  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  // 시리얼 통신 검사 후 데이터 읽기
  if(Serial.available()) {
    char ch = Serial.read();
    if(ch == '1')
      digitalWrite(LED_BUILTIN, HIGH);
    else if(ch == '2')
      digitalWrite(LED_BUILTIN, LOW);
  }
  delay(10);
}
```

- 시리얼 모니터를 이용하여 전송





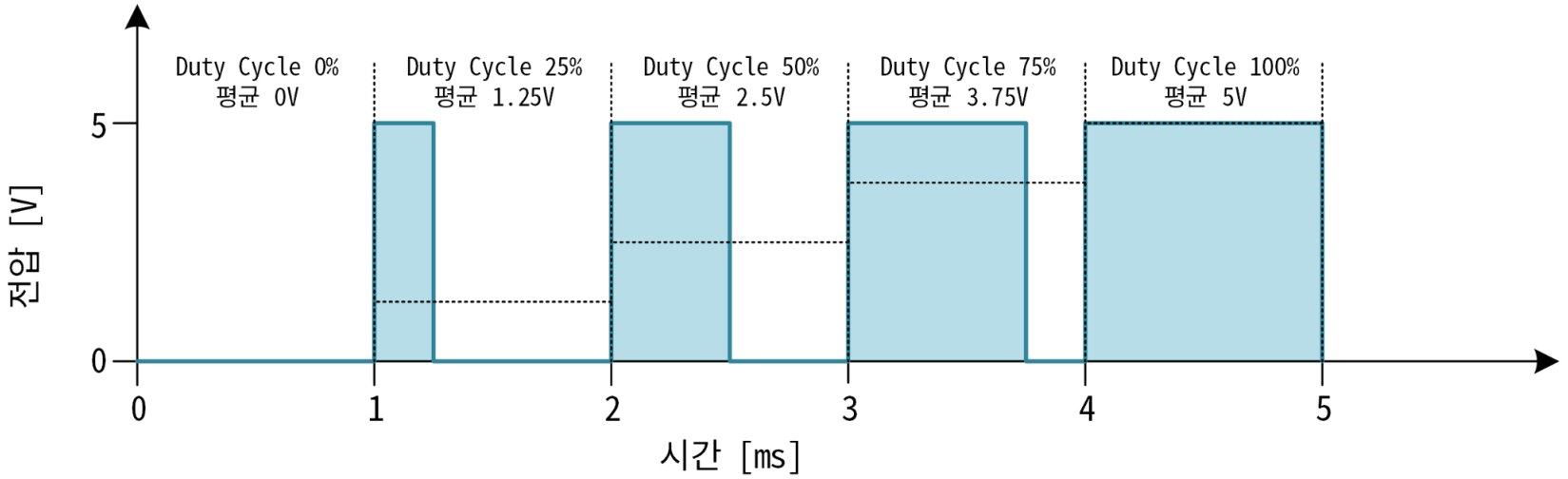
5. 아날로그 입출력의 기초



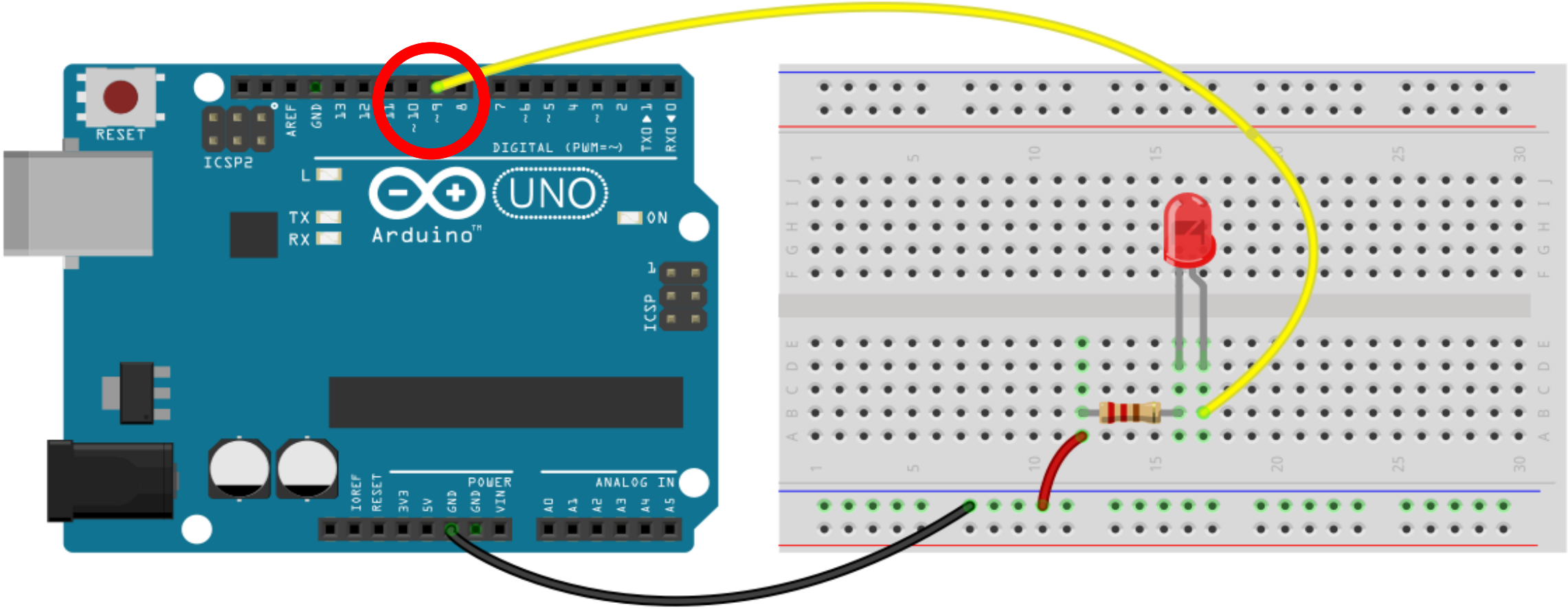
5.1 아날로그 출력

- 아두이노 우노의 입/출력 기능
  - 14개의 디지털 입/출력 기능
  - 6개의 아날로그 입력 기능
  - 디지털 출력 : HIGH → 5V, LOW → 0V
  - 아날로그 출력 기능은 없음
    → PWM(Pulse Width Modulation) 출력으로 해결

- PWM(Pulse Width Modulation)
  - 사각형 펄스를 주기적으로 발생시키면서 5V, 0V의 발생 시간을 조절하면서 평균 전압을 제어
  - 5V, 0V의 발생 시간의 비 : 듀티 사이클 (duty cycle)
  - 아두이노 : 6개의 디지털 출력핀이 가능 (~기호로 표시)
  - analogWrite(9, 127); → 9번 핀으로 50%듀티 사이클 출력



- 회로 구성


RESET
ICSP2
AREF
GND
13
12
11
10
9
8
7
6
5
4
3
2
TXO 1
RXO 0
DIGITAL (PWM=~)
L
TX
RX
UNO
Arduino™
ON
ICSP
POWER
IOREF
RESET
3V3
5V
GND
GND
VIN
ANALOG IN
A0
A1
A2
A3
A4
A5

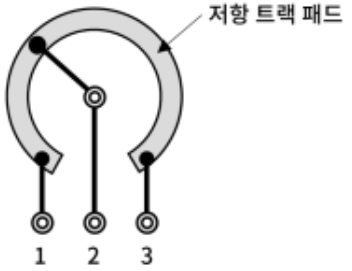
- 스케치

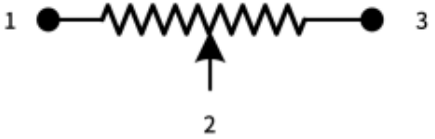
예제 5-1. 아날로그 출력 예제

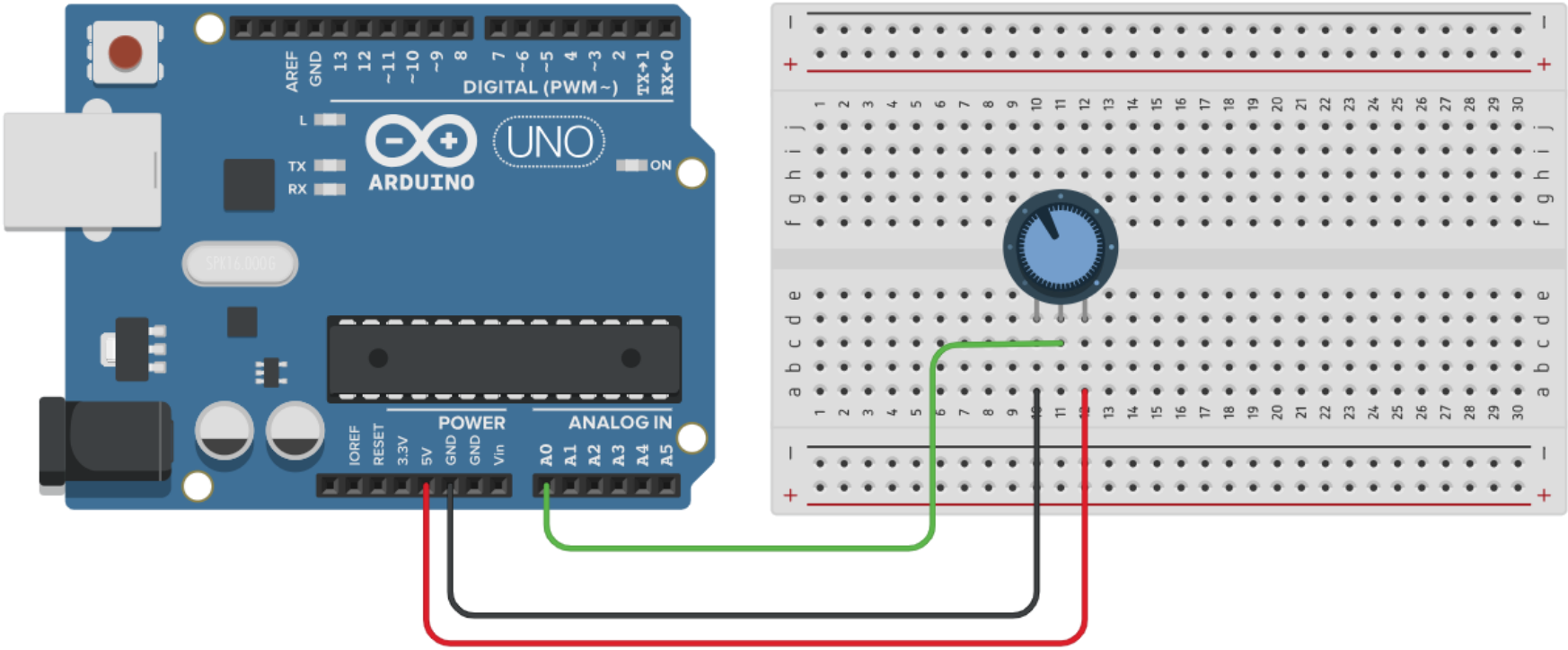
```
void setup()
{
  // PWM 신호를 사용하기 위하여 9번 핀을 출력으로 설정
  pinMode(9, OUTPUT);
}

void loop()
{
  // 변수 정의 : 밝기를 저장함 (0 ~ 255)
  int br;
  // for문을 사용하여 변수 br의 값을 0에서 255까지 5씩 증가
  for (br = 0; br <= 255; br += 5) {
    analogWrite(9, br);
    delay(30); // 30 미리초 대기
  }
  // for문을 사용하여 변수 br의 값을 255에서 0까지 5씩 감소
  for (br = 255; br >= 0; br -= 5) {
    analogWrite(9, br);
    delay(30); // 30 미리초 대기
  }
}
```

5.2 포텐쇼미터를 이용한 아날로그 입력

- 포텐쇼미터(potentiometer)
  - 가변저항
- 회로구성







- 스케치

예제 5-1. 포텐쇼미터를 이용한 아날로그 입력

```
// 아날로그 입력 핀(A0~A5) 중에서 A0를 입력으로 사용
int sensor = A0;

void setup()
{
  // 시리얼 통신 초기화
  Serial.begin(9600);
  // 아날로그 센서 포트를 입력으로 사용
  pinMode(sensor, INPUT);
}

void loop()
{
  // 아날로그 센서를 통해서 값을 읽어들임
  int value = analogRead(sensor);
  // 시리얼 통신을 이용하여 PC로 전송
  Serial.println(value);
  // 100ms 대기
  delay(100);
}
```

- 아날로그 센서 읽기 (예: A0)
  - 초기화 : `setup()`함수에서 `pinMode(A0, INPUT);`
  - 값 읽어들이기 : `int value = analogRead(A0);`
  - 입력 : 0 ~ 5V 전압 측정
  - 값의 범위 : 0 ~ 1023 (아두이노에서는 10 bit ADC를 사용)
    0000000000_2 (0_{10}) ~ 1111111111_2 (1023_{10})
  - 입력전압의 계산 : 입력전압 $= \frac{\text{아날로그입력값}}{1024} \times 5\ [\text{V}]$