

## Chapter 07

따라 하면서 배우는

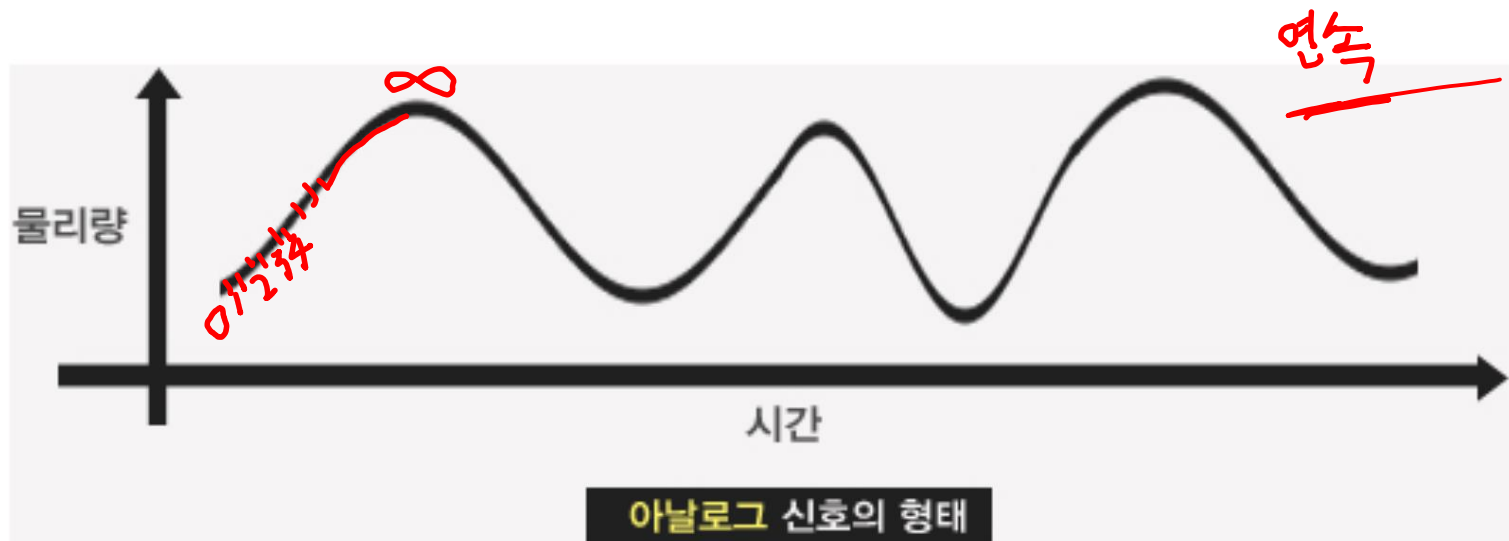
# 아두이노

아두이노 메가2560으로 마이크로컨트롤러 시작하기

# 아날로그 데이터 입출력

# 아날로그 데이터 처리

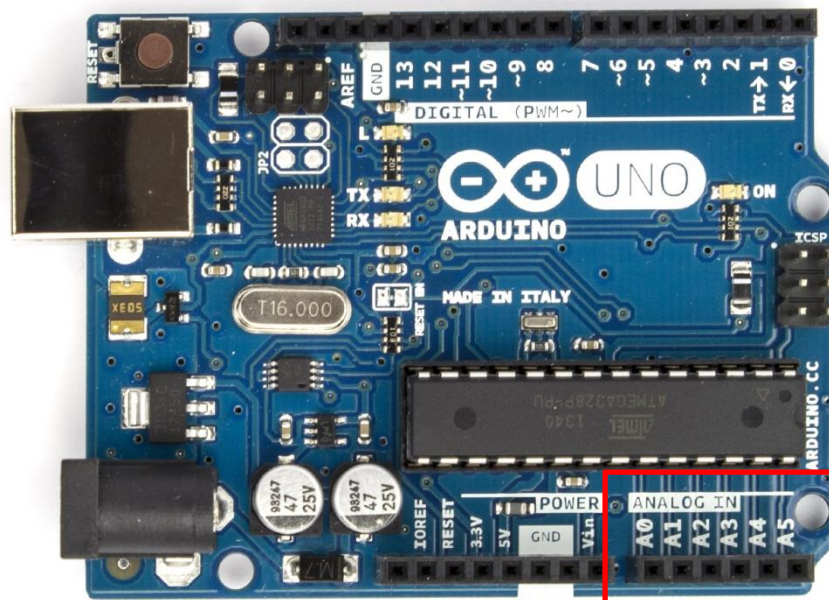
## ■ 아날로그 신호와 디지털 신호



# 아날로그 데이터 처리

## ■ 아날로그 데이터 입력

- 마이크로컨트롤러에는 16채널의 10비트 해상도 아날로그-디지털 변환기(ADC)가 포함되어 있음 Analog to Dig Conv
- 각 채널에는 'A0'에서 'A5'까지의 핀이 할당



Digital

6개 채널

# 아날로그 데이터 입력

## ■ 6 채널의 ADC

- 하나의 아날로그-디지털 변환기를 공유하므로 동시에 여러 채널 사용은 불가능

센서 6개

## ■ 10비트 해상도

- 0에서 1023 사이의 정수값 반환
- 1023에 해당하는 기준 전압은 아두이노 우노의 동작 전압인 5V가 일반적으로 사용됨
- $5V / 1024 \approx 4.9mV$  전압 차이 인식 가능

전류

5/2

0V

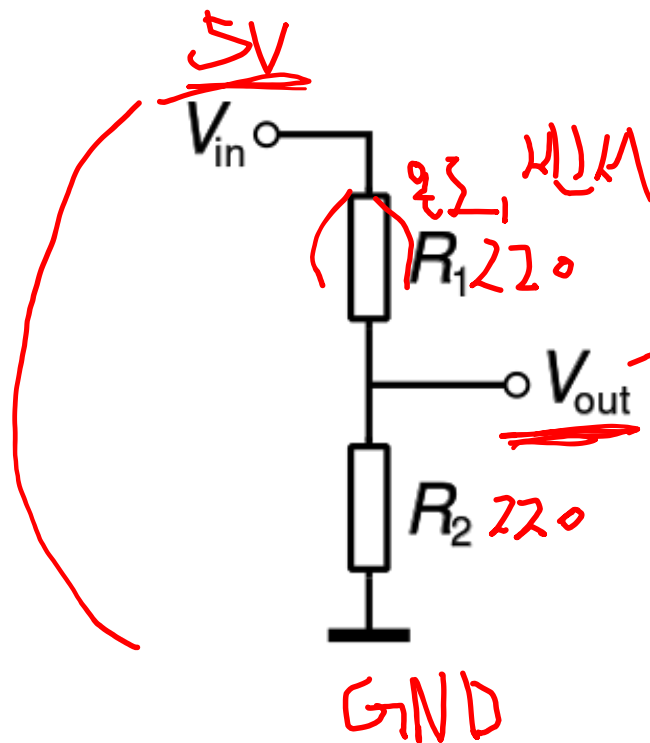
2.5V

# 전압 분배법칙

## ■ 전압 분배

- $V_{in}$ 에 비례하는 전압( $V_{out}$ )을 만들기 위해 사용
- $R_1, R_2$ 의 비율로 전압을 분배

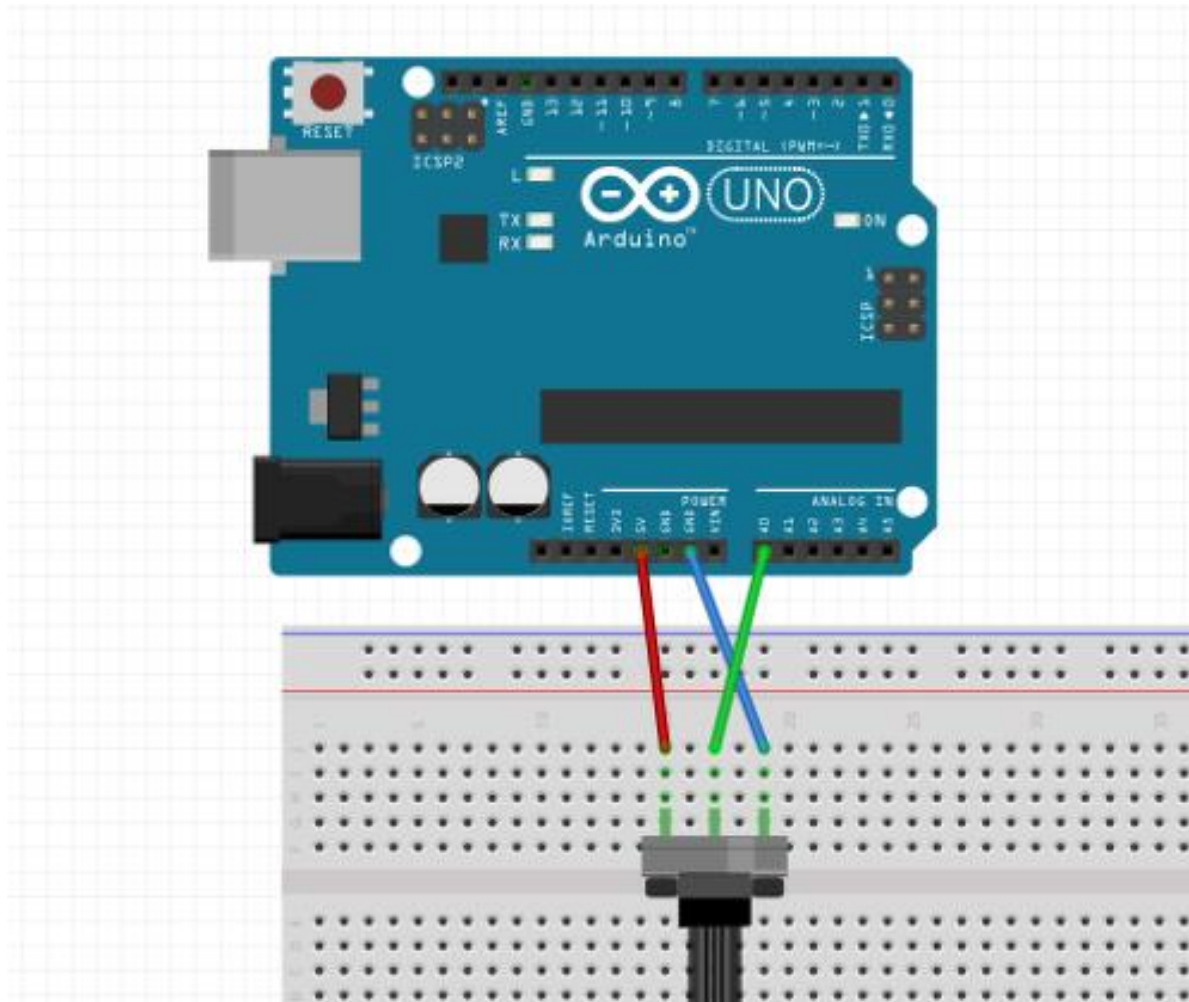
전압 ↓  
5V (↓) 5V (↓)  
5V (↓) 5V (↓)  
GND  
기존 전선 온도, 조도 ↑ ↓ 저항값



$$V_{out} = \frac{R_2}{R_1 + R_2} \cdot V_{in}$$

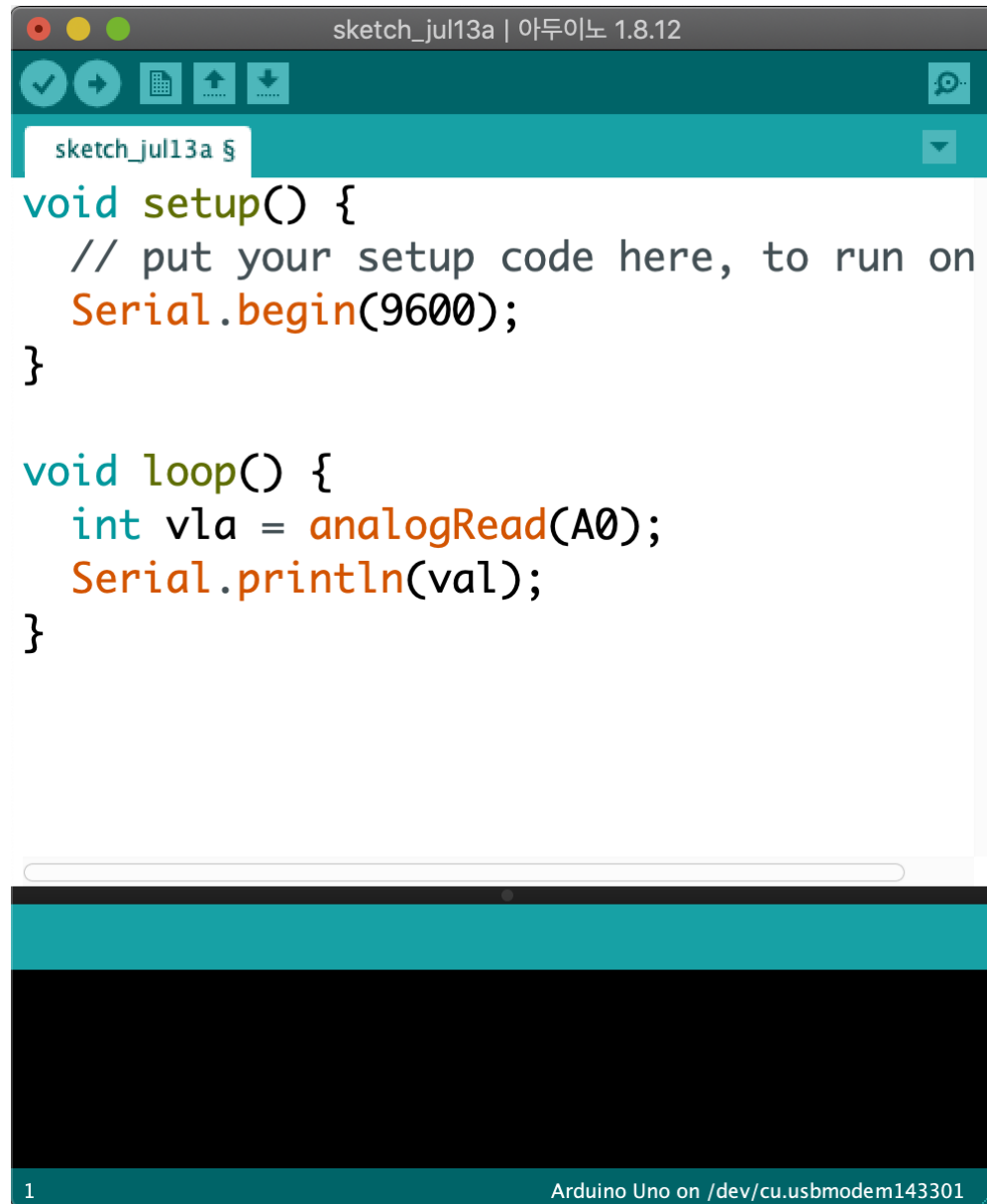
2.5V  
220 + 220  
240  
240

# 가변저항 실습



회전형 가변저항 (rotary potentiometer)

# 가변저항 실습



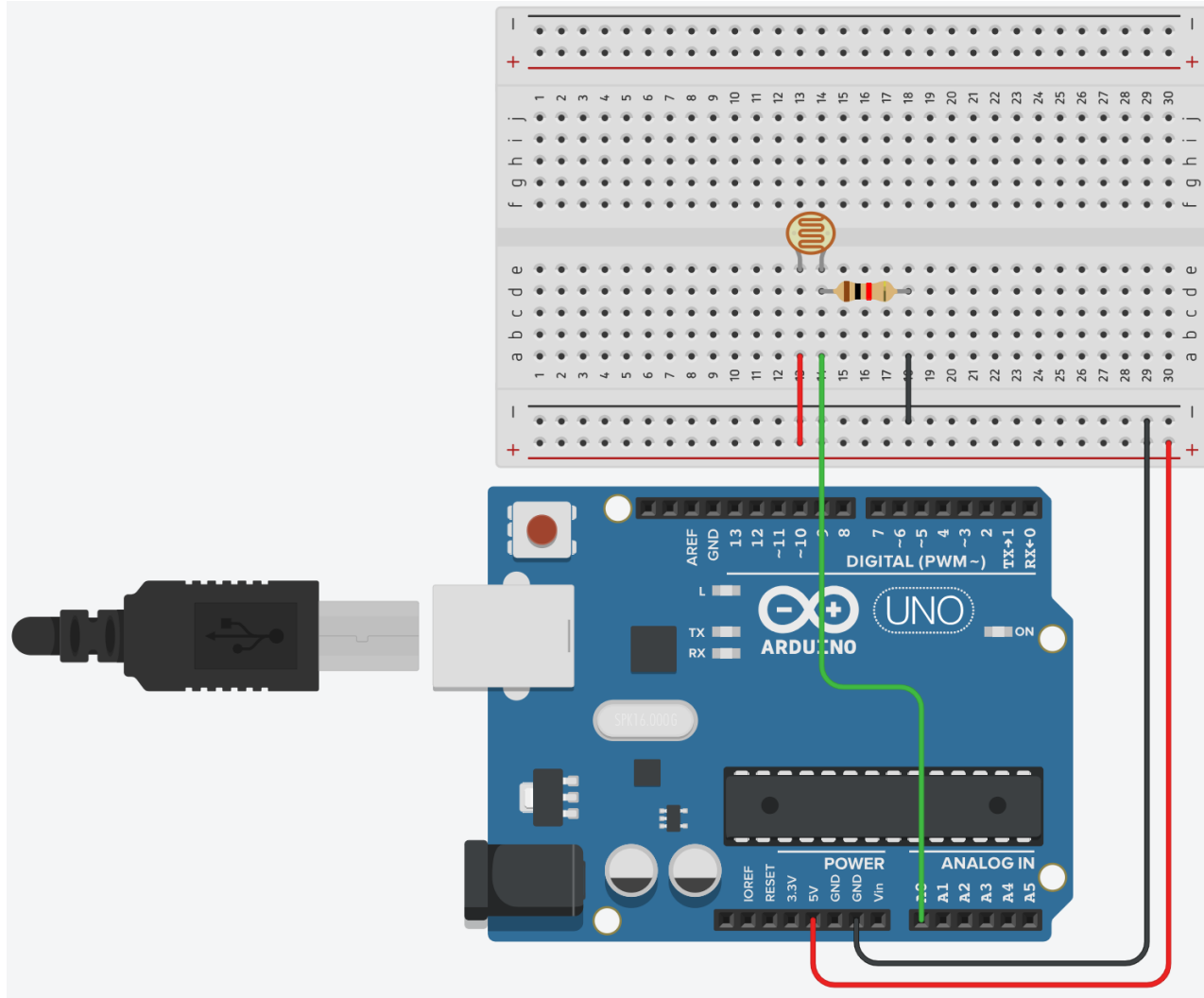
The image shows a screenshot of the Arduino IDE interface. The title bar at the top reads "sketch\_jul13a | 아두이노 1.8.12". Below the title bar is a toolbar with icons for checking, running, serial monitor, and file operations. The main text area contains the following C++ code:

```
sketch_jul13a §
void setup() {
  // put your setup code here, to run on
  Serial.begin(9600);
}

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

At the bottom of the IDE, there is a status bar. On the left, it shows the line number "1". On the right, it shows the connection status "Arduino Uno on /dev/cu.usbmodem143301".

# 조도센서(CDS) 실습





# 조도센서(CDS) 실습

플러터

```
sketch_apr30a | 아두이노 1.8.7
파일 편집 스케치 툴 도움말

sketch_apr30a $
int cds = A0;

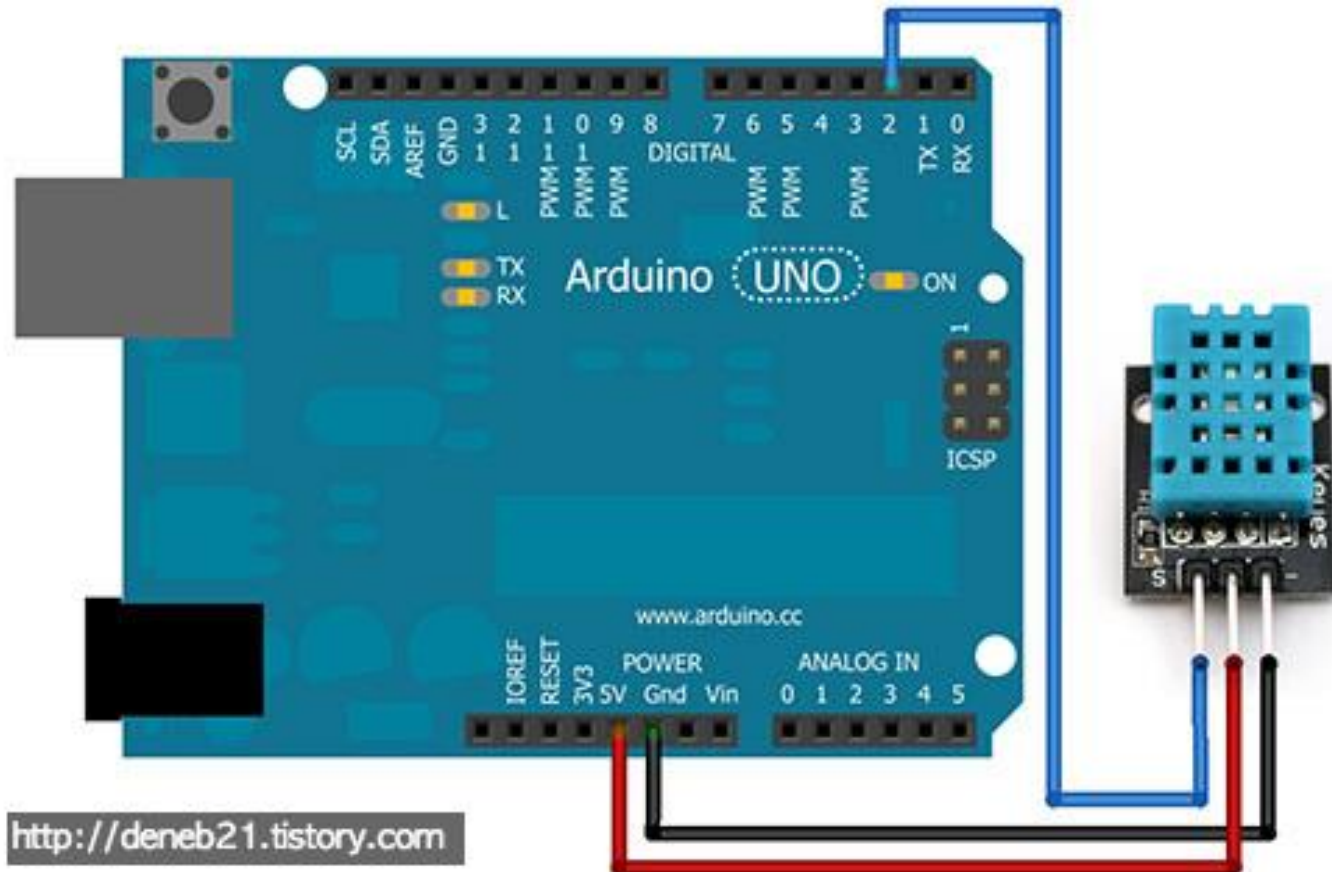
void setup() {
  Serial.begin(9600);
}

void loop() {
  int val= analogRead(cds);
  Serial.println(val);
  delay(1000); x
}

Arduino Version: 1.8.7

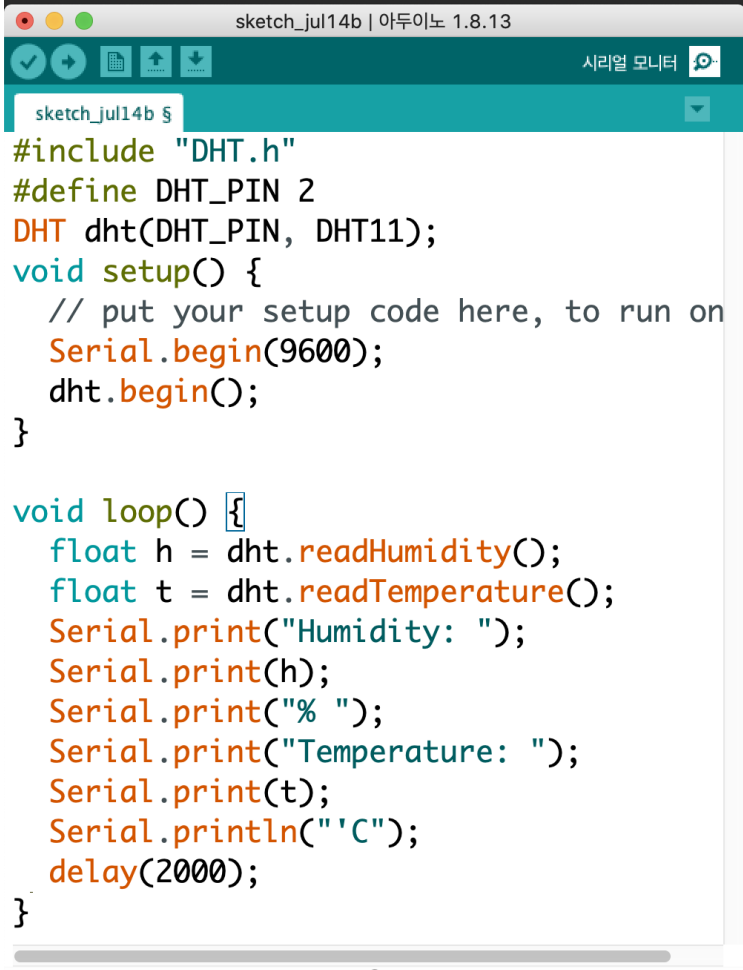
14 Arduino/Genuino Uno on COM13
```

# 온습도 센서(DHT11) 실습



# 온습도 센서(DHT11) 실습

- DHT11 라이브러리 다운로드 필수
- 툴 -> 라이브러리관리
- 검색 -> DHT by adafruit



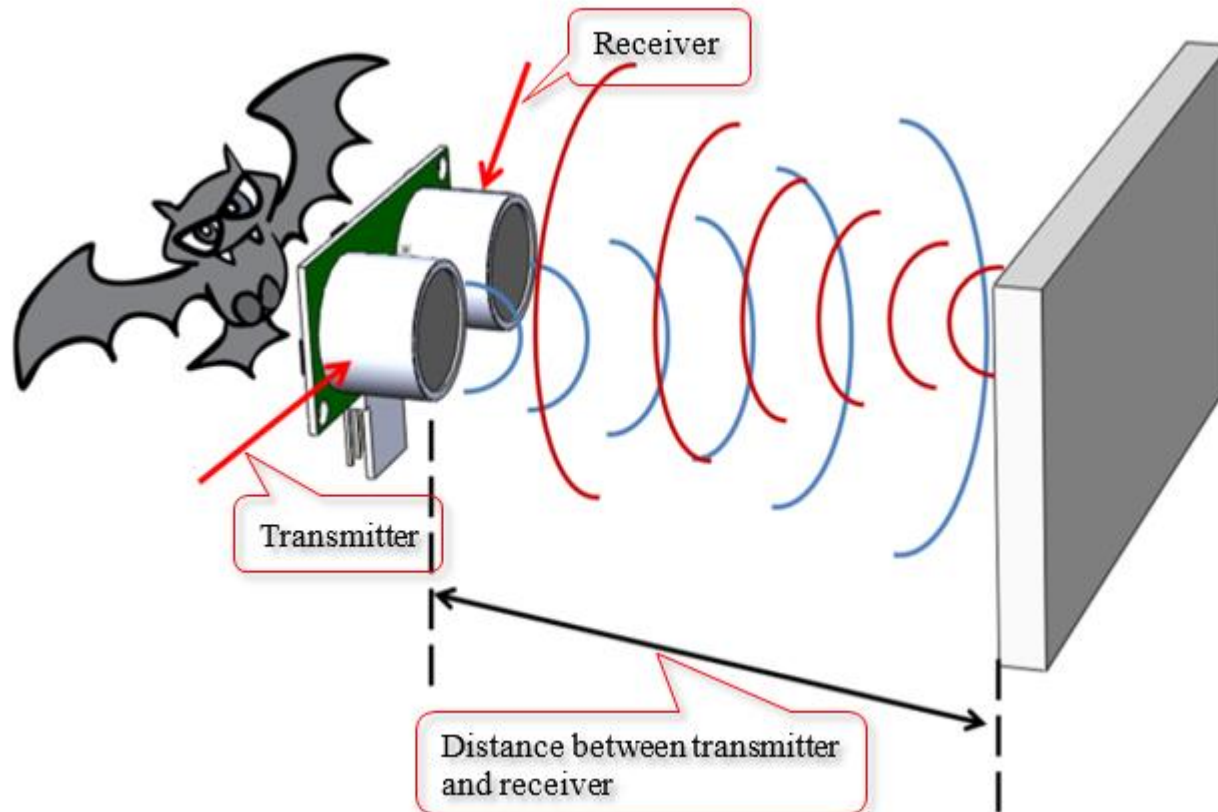
```
sketch_jul14b | 아두이노 1.8.13
시리얼 모니터
sketch_jul14b §
#include "DHT.h"
#define DHT_PIN 2
DHT dht(DHT_PIN, DHT11);
void setup() {
    // put your setup code here, to run on
    Serial.begin(9600);
    dht.begin();
}

void loop() {
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    Serial.print("Humidity: ");
    Serial.print(h);
    Serial.print("% ");
    Serial.print("Temperature: ");
    Serial.print(t);
    Serial.println("'C");
    delay(2000);
}
```

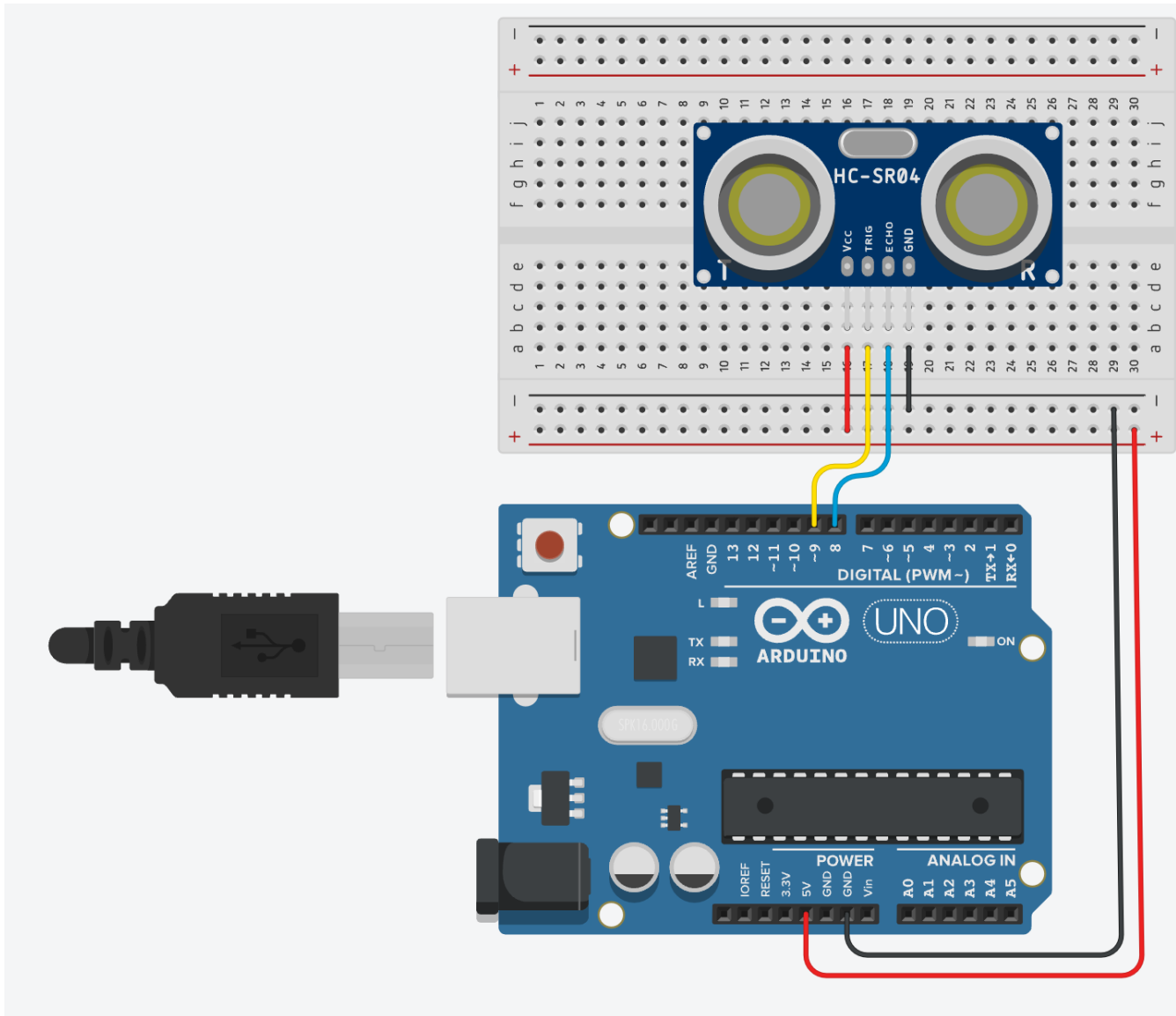
스케치는 프로그램 저장 공간 5054 바이트(15%)를 사용.  
전역 변수는 동적 메모리 249바이트(12%)를 사용, 1799

20 Arduino Uno on /dev/cu.usbmodem144301

# 초음파센서(Ultra Sonic) 원리



# 초음파센서(Ultra Sonic) 실습



TRIG : 송신부  
ECHO : 수신부

# 초음파센서(Ultra Sonic) 실습

```
//초음파 센서의 핀번호를 설정한다.
int echoPin = 8;
int trigPin = 9;

void setup() {
  Serial.begin(9600);
  // trig를 출력모드로 설정, echo를 입력모드로 설정
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

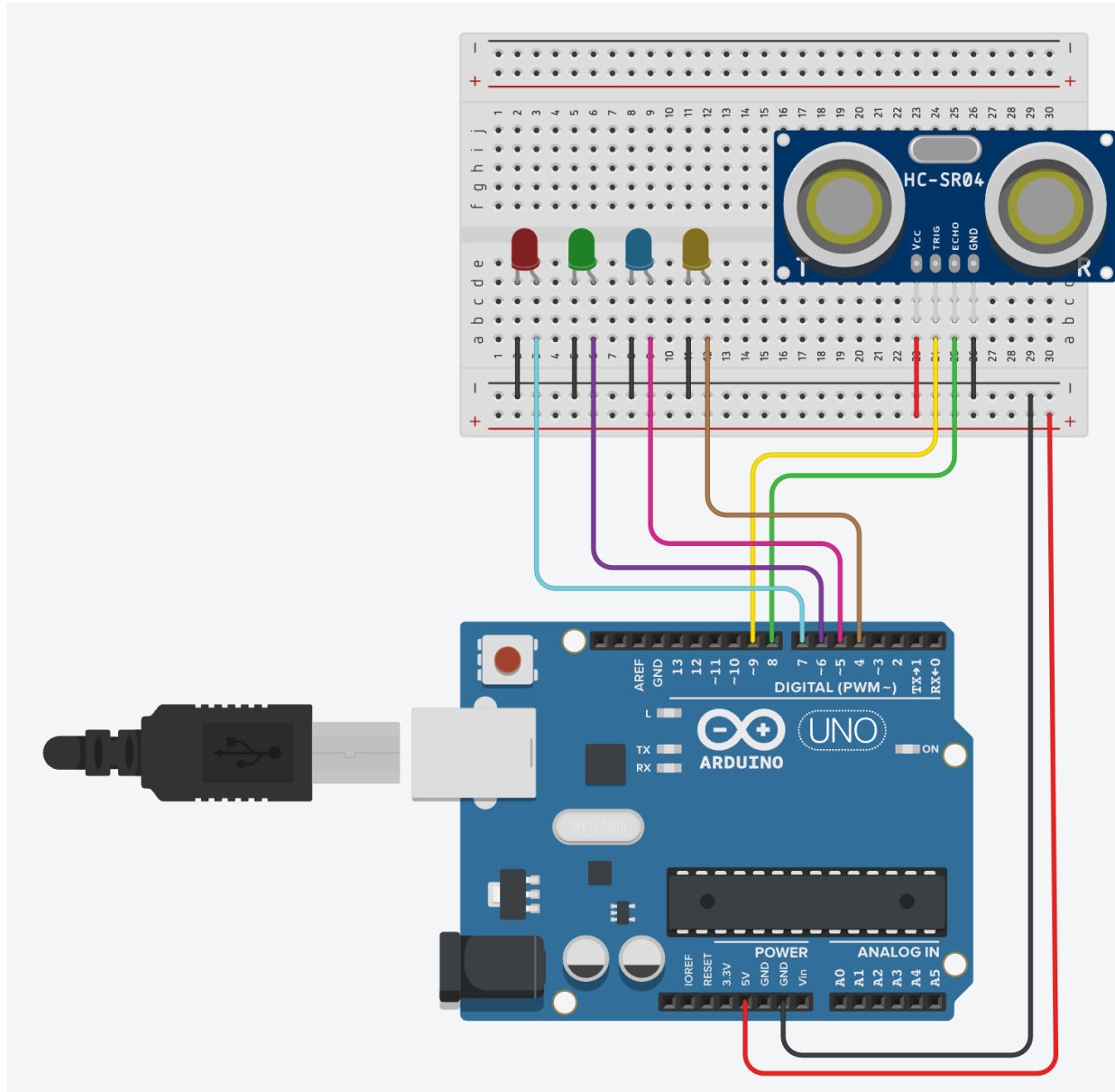
void loop() {
  // 초음파를 보낸다. 다 보내면 echo가 HIGH 상태로 대기하게 된다.
  digitalWrite(trig, LOW);
  digitalWrite(echo, LOW);
  delayMicroseconds(2);

  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);

  // echoPin 이 HIGH를 유지한 시간을 저장 한다.
  unsigned long duration = pulseIn(echoPin, HIGH);
  // HIGH 였을 때 시간(초음파가 보냈다가 다시 들어온 시간)을 가지고 거리를 계산 한다.
  float distance = ((float)(340 * duration) / 10000) / 2;

  Serial.print(distance);
  Serial.println("cm");
  delay(500);
}
```

# 실습1 초음파센서로 LED 표현



## 실습 2 거리에 따른 LED 켜지기

- 초음파센서와 물체가 가까워지면,
  - LED의 수가 많이 켜지고 (4개)
- 반대로 초음파센서와 물체가 멀어지면,
  - LED의 수가 적게 켜지도록 (1개)



# Thank you!!

