

# 사물인터넷

블루투스, 릴레이, 터치, 초음파센서

101111000101001010011000101001111010101010111110011111010100001111100001000010101011001111101010000111110000100001010101111100111110101000011111000010000101011001111101010000111110000100001010110011111010100001111100001000010101010000101

# 블루투스 통신이란?

---

- 1994년 에릭슨이 최초로 개발한 개인 근거리 무선통신을 위한 산업 표준 통신방식.
- 1999년 5월 20일 공식적으로 발표
- 현재 블루투스는 1.0버전을 시작으로 5.2 버전까지 업데이트 됨.

# 블루투스 프로파일 (Bluetooth Profile)

## ■ 프로파일 (Profile)이란?

- ❖ 블루투스의 목적에 맞게 프로토콜을 정의한 것
- ❖ 블루투스 프로토콜은 여러 계층의 스택으로 구성되어 있으며, 프로파일에 따라 각 계층의 설정과 연결이 변경될 수 있다.

## ■ 흔히 볼 수 있는 프로파일

- ❖ A2DP (Advanced Audio Distribution Profile)
  - 오디오 통신을 위한 프로파일
- ❖ BPP (Basic Printing Profile)
  - 프린트를 연결하여 출력하기 위한 프로파일
- ❖ FTP (File Transfer Profile)
  - 파일을 전송하기 위한 프로파일
- ❖ SPP (Serial Port Profile)
  - 두 개의 장치가 직렬 포트 (Serial Port) 생성하여 블루투스 통신을 한다.

# 블루투스 모듈 사용해보기

## ■ HC-06 블루투스 모듈

❖ 블루투스를 이용한 시리얼 통신 가능

■ 페어링 이란 과정 후 사용 할 수 있다.

■ 페어링 시 모듈의 고유 이름을 통해서 식별 가능하다.



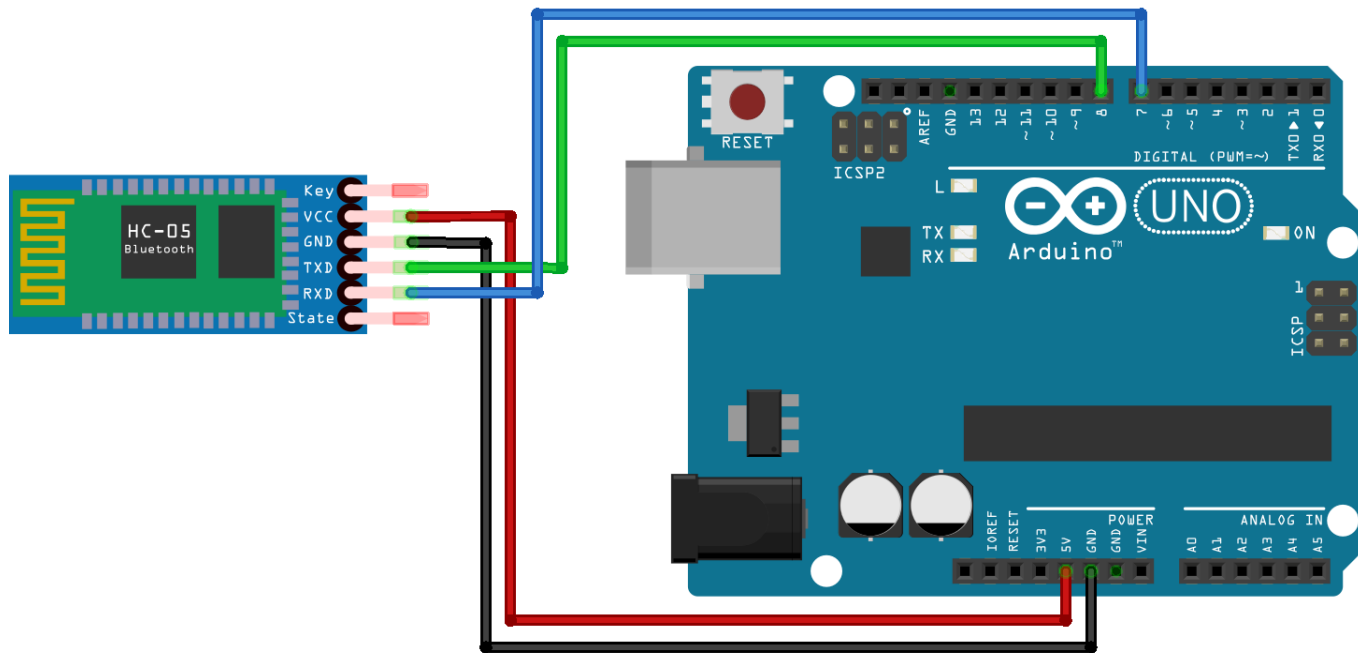
# 블루투스 모듈 핀 아웃

## ■ 다음과 같이 연결

- ❖ GND – GND
- ❖ VCC – 5V
- ❖ RXD – 8
- ❖ TXD – 7



# 연결도



## 예제 코드

```
#include <SoftwareSerial.h>

#define BT_RXD 8
#define BT_TXD 7

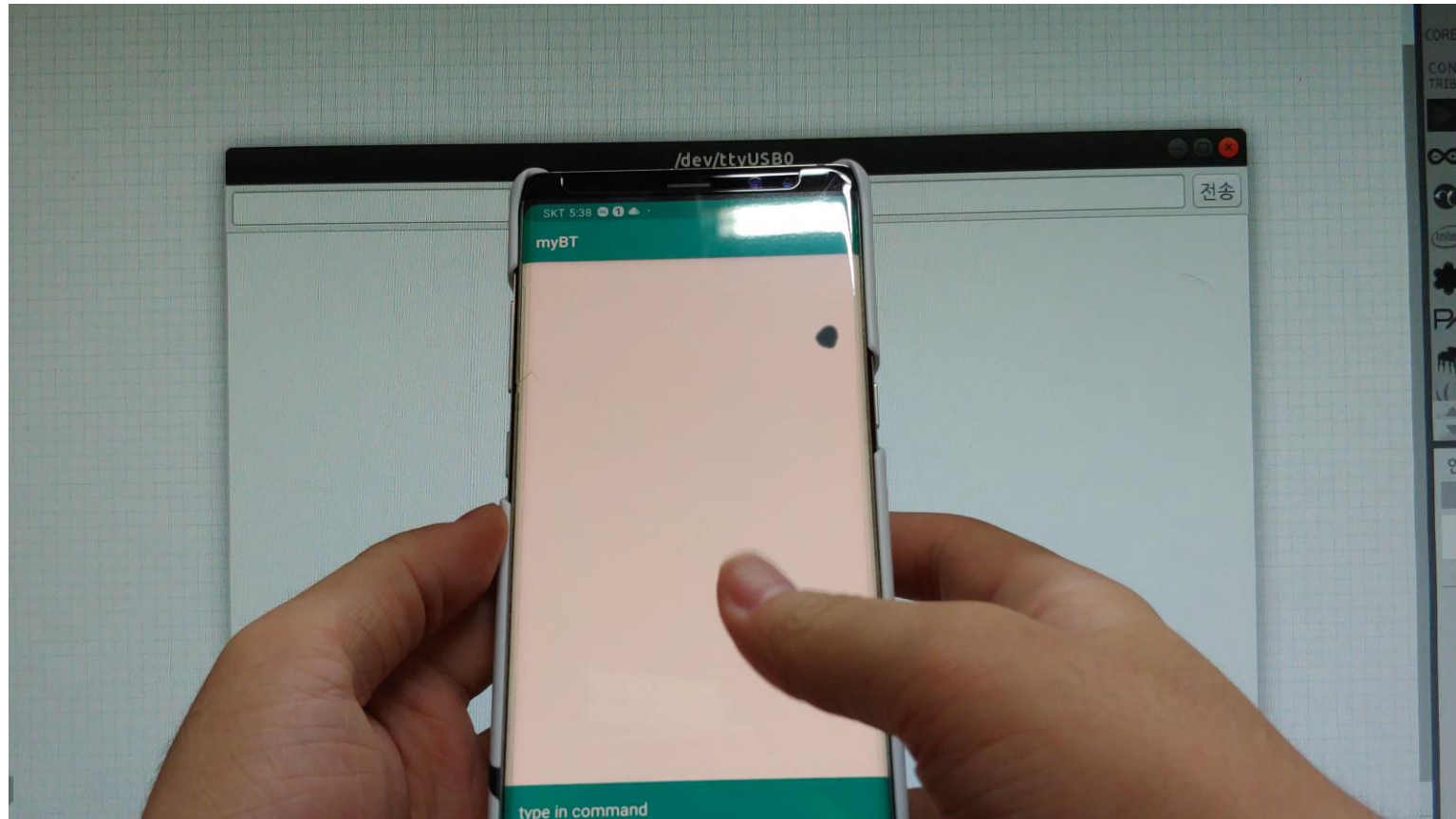
SoftwareSerial bluetooth(BT_RXD, BT_TXD);

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

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



# 동작 확인



# 아두이노 소프트웨어 시리얼

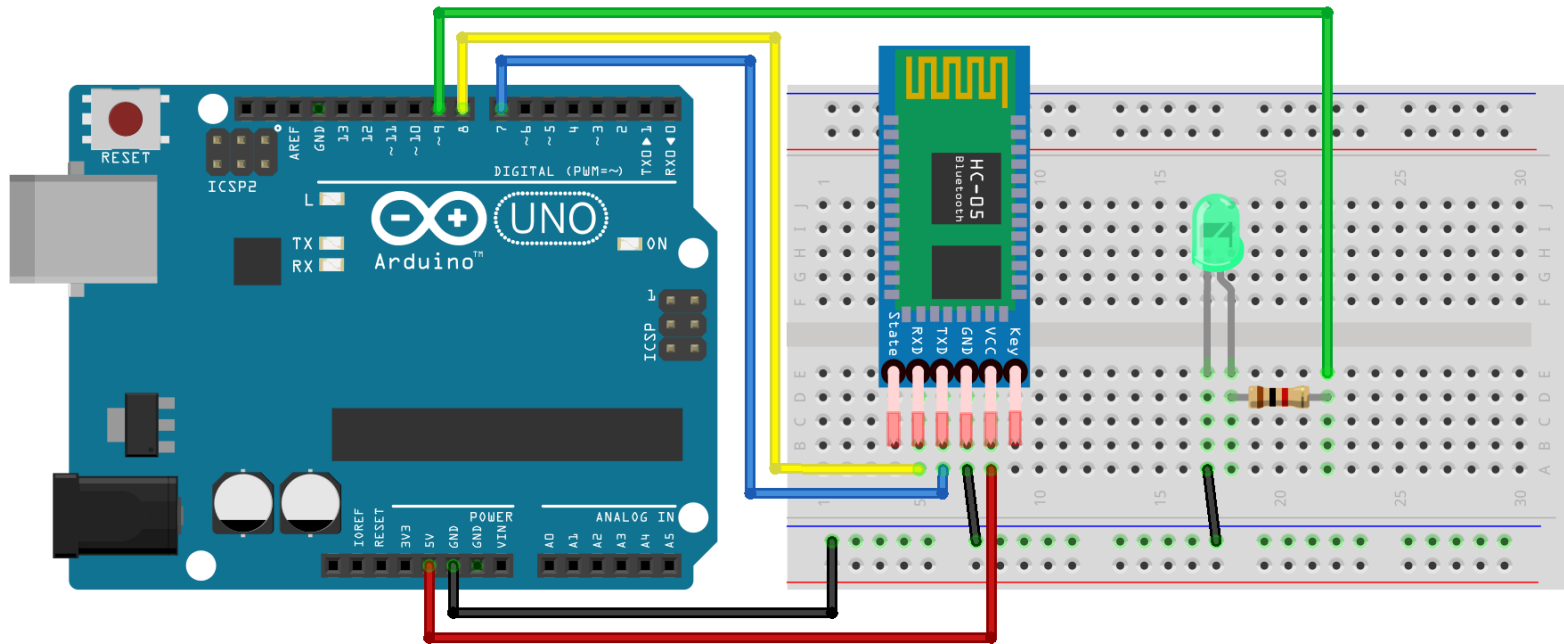
## ■ 소프트웨어 시리얼 (SoftwareSerial) 이란?

- ❖ GPIO 를 사용하여 uart 통신을 소프트웨어적으로 구현한 라이브러리.
- ❖ 특정 GPIO를 시리얼통신용 TX, RX로 지정할 수 있다.

## ■ 아두이노의 시리얼모니터 기능은 GPIO(핀) 0, 1을 이용하고 있고 블루투스 모듈을 사용한 통신을 위해 시리얼 (UART) GPIO가 추가적으로 필요하다.

- ❖ 만약 0, 1핀을 사용하게 되면 업로드시 정상 동작을 하지 않게 된다
- ❖ 시리얼 (UART) 통신은 1:1 통신이지만 1:2 통신으로 연결되기 때문이다.

# 연결도



# 예제 코드

```
#include <SoftwareSerial.h>

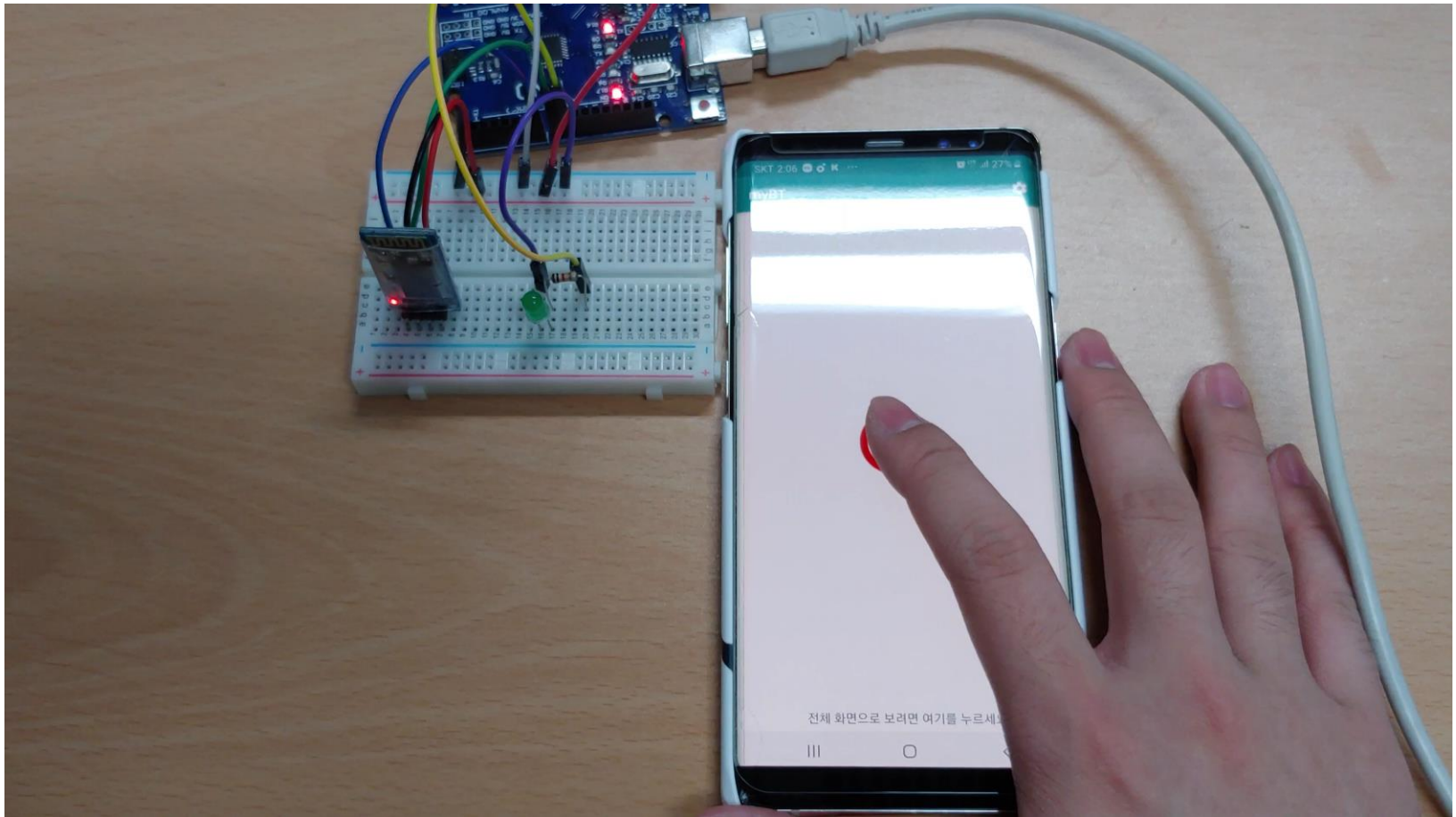
#define BT_RXD 8
#define BT_TXD 7
#define LED_PIN 9

SoftwareSerial bluetooth(BT_RXD, BT_TXD);

void setup(){
  bluetooth.begin(9600);
  pinMode(LED_PIN, OUTPUT);
}

void loop(){
  if(bluetooth.available() > 0){
    int state = bluetooth.read() - 48;
    if(state == 1){
      digitalWrite(LED_PIN, HIGH);
    }
    else{
      digitalWrite(LED_PIN, LOW);
    }
  }
}
```

# 동작 확인



# 아두이노 블루투스 앱

## ■ Arduino bluetooth controller

- ❖ 아두이노 블루투스 모듈과 연결 후 데이터를 전송하기 위한 안드로이드 앱.
- ❖ 이 앱 외에도 블루투스 통신 앱 변경 및 보이스 기능 추가 있다.
- ❖ 일부 기능들의 전송할 값...



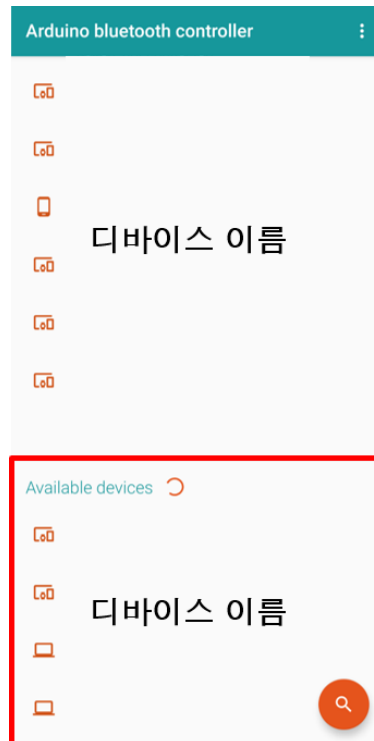
Ar  
Giumig

제거

열기

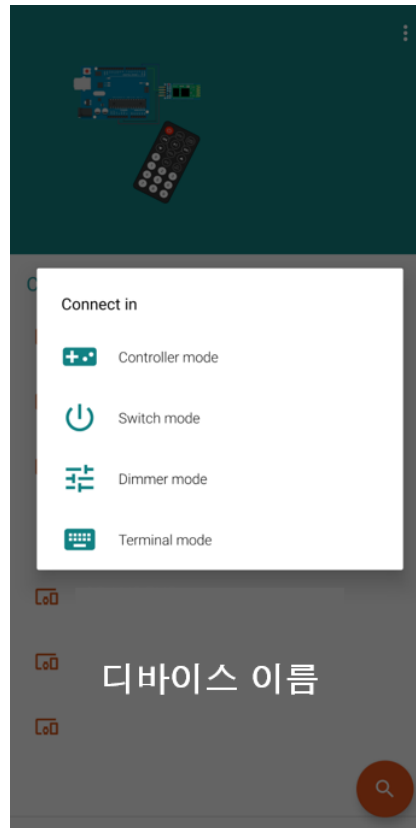
# 아두이노 블루투스 앱 연결과 사용방법

- ❖ 처음 실행 화면에서 스크롤을 내리면 연결 가능한 디바이스들을 찾을 수 있다.
- ❖ 우선 블루투스 모듈에 전원을 공급한 후 여기서 디바이스 이름 “myBT” 에 페어링을 한다.



# 아두이노 블루투스 앱 연결과 사용방법

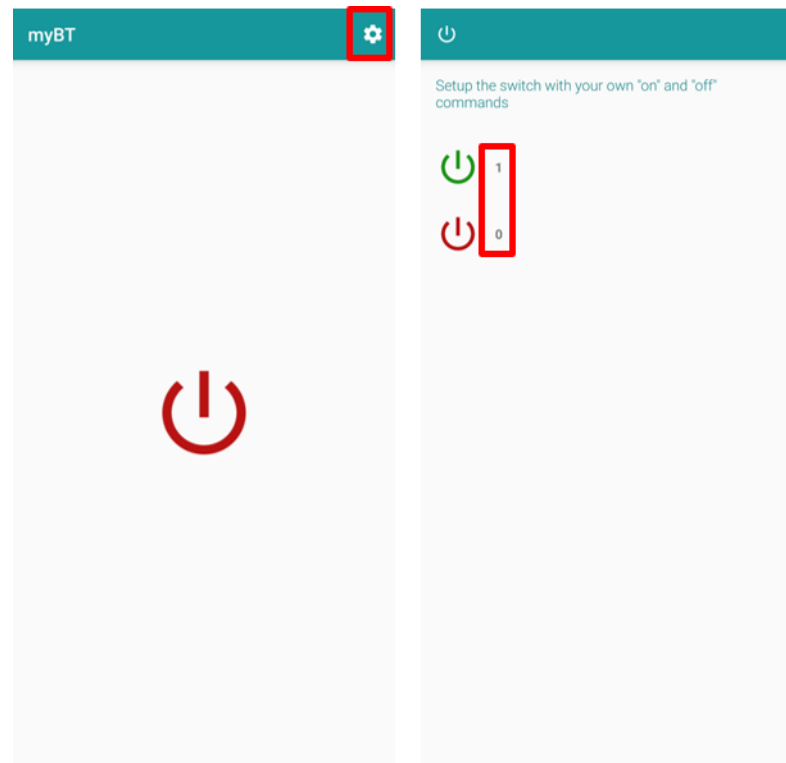
- ❖ 연결을 시도하면 다음과 같이 여러가지 기능들을 선택할 수 있는 메뉴가 나타난다.
- ❖ 텍스트 전송 예제는 “Terminal mode” 를 클릭하면 된다.





# 아두이노 블루투스 앱 연결과 사용방법

- ❖ “Switch mode” 를 사용하기 위해서는 전송 값 설정이 필요하다.
- ❖ 우측상단의 톱니바퀴를 클릭하고 나오는 화면에서 버튼이 초록색일 때 1, 빨간색일 때 0을 전송하도록 입력해주면 스위치모드를 사용할 수 있는 세팅이 완료된다.



101111000101001010011000101001111010101010111110011111010100001111100001000010101011001111101010000111110000100001010101111100111110101000011111000010000101011001111101010000111110000100001010110011111010100001111100001000010101010000101

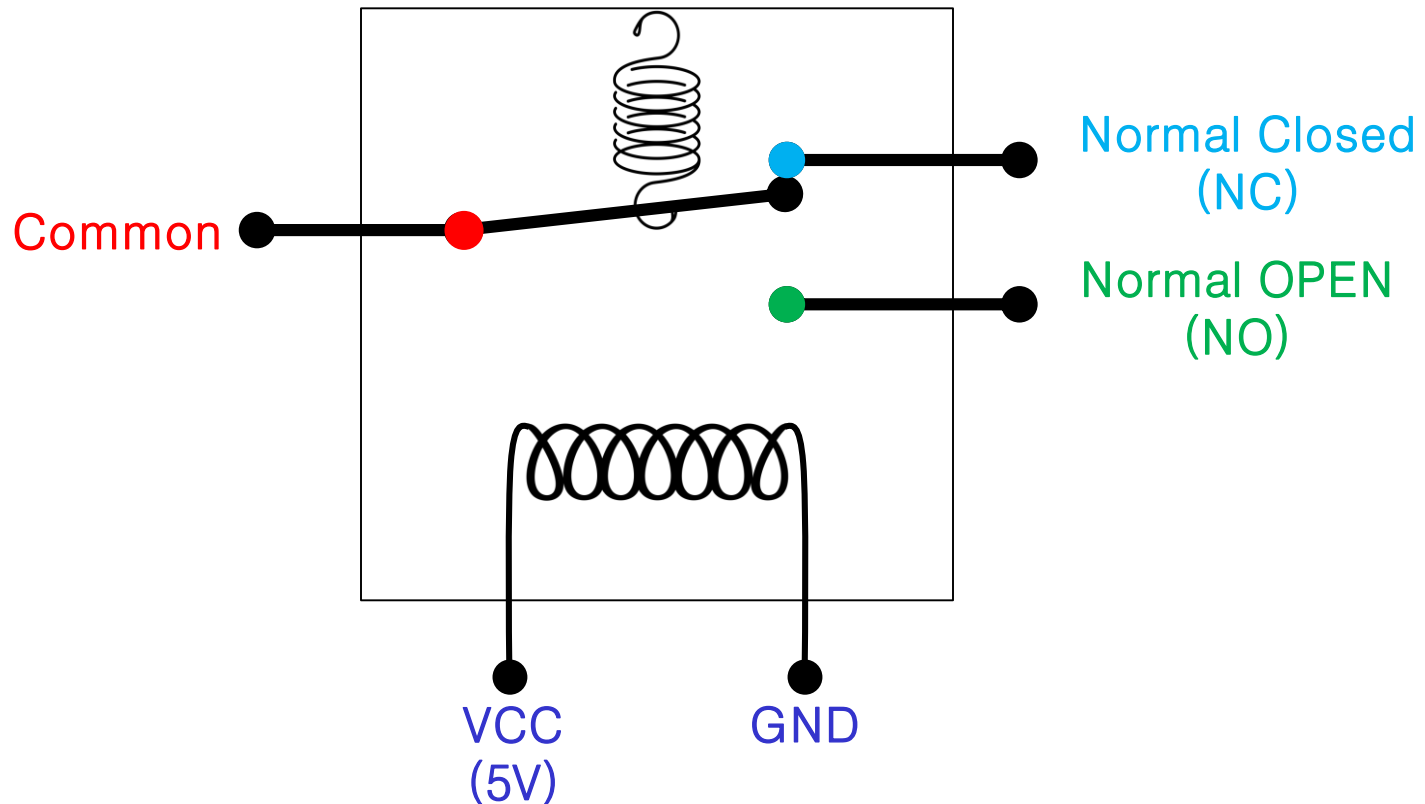
# 릴레이 란?

- 작은 전압으로 큰 전압을 컨트롤 하기 위한 전자석 스위치.
- 아두이노에서 선풍기, 전등 등을 켜고 끌 수 있게 함.
- 평상시에 스위치가 열려있는 NormalOpen(NO) 모드와  
평상시에 스위치가 닫혀있는 NormalClose(NC) 모드가 있  
음.



# 릴레이 원리

- 전자석의 전자기유도원리를 이용.
- 아두이노는 전자석 코일만 컨트롤 한다.



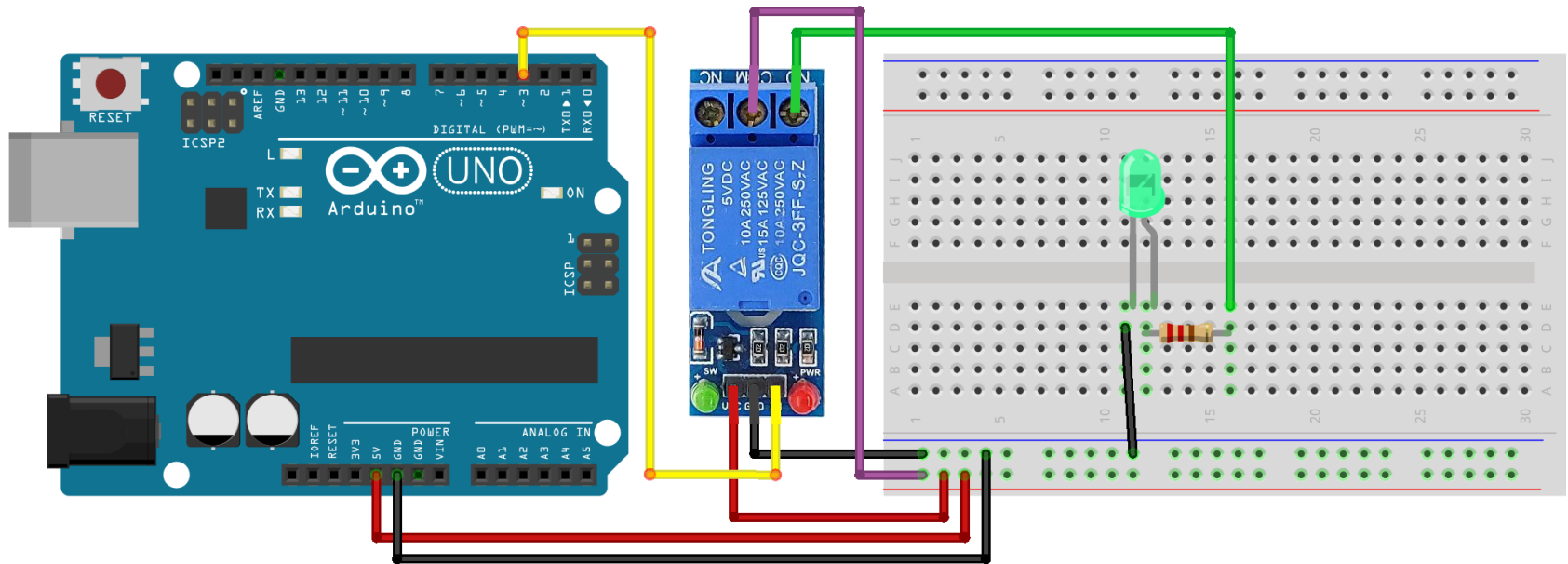
# 릴레이 핀 아웃

■ 다음과 같이 연결

- ❖ GND – GND
- ❖ VCC – 5V
- ❖ IN – 3



# 연결도



## 예제 코드

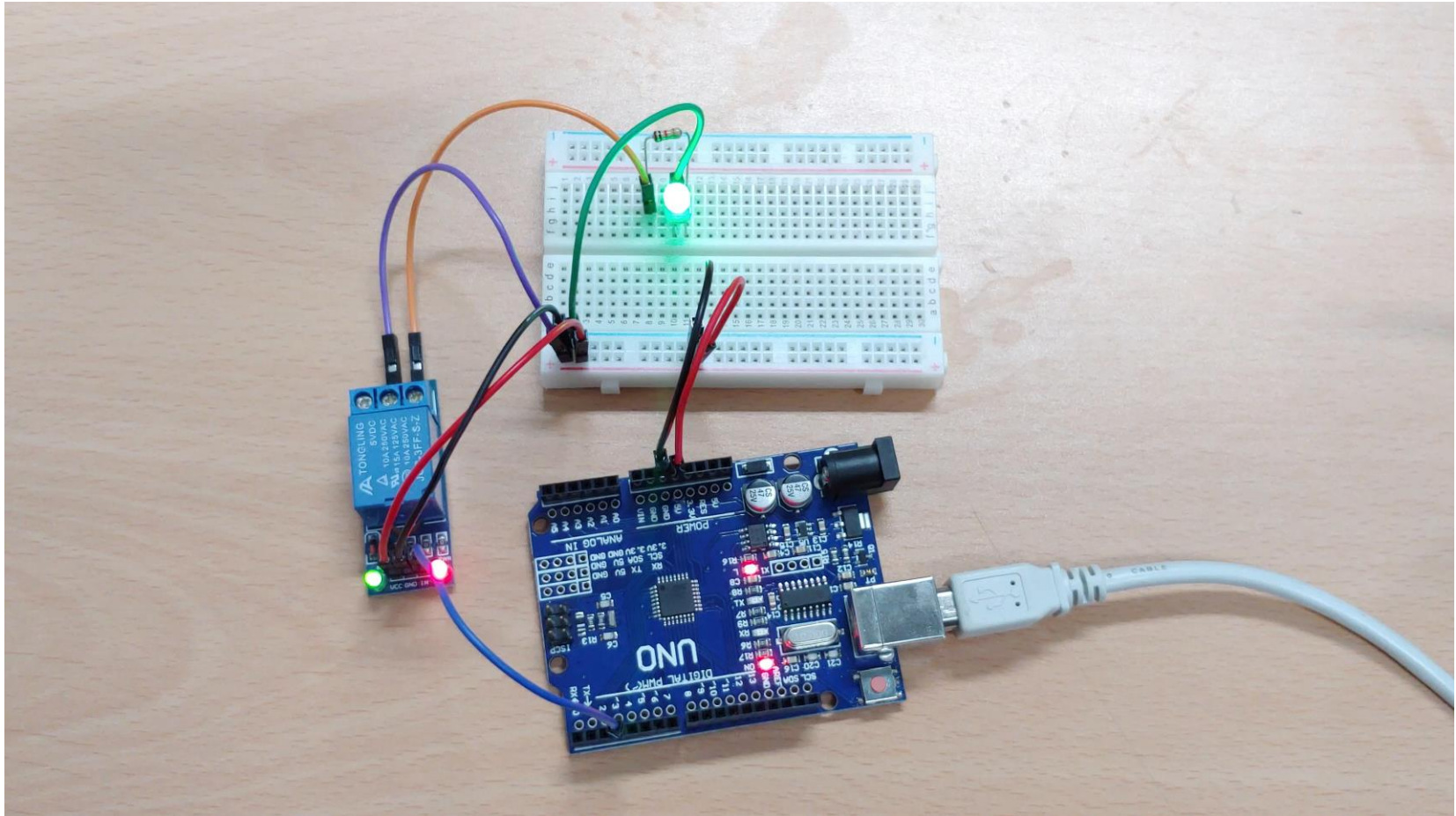
```
#define RELAY_PIN 3

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

void loop() {
  digitalWrite(RELAY_PIN, HIGH);
  delay(1000);
  digitalWrite(RELAY_PIN, LOW);
  delay(1000);
}
```



# 동작 확인





**LED 회로도 결선도 및 설명**

1011110001010010100110001010011110101010101111100111110101000011111000010000101010110011111010100001111100001000010101011111001111101010000111110000100001010101111100111110101000011111000010000101011001111101010000111110000100001010101010000101

# 사용 사례

## ■휴대 폰이나 테블릿의 버튼

## ■장점

- ❖기계적 움직임이 없어 내구성이 우수
- ❖방수 또는 방진에 강함

## ■단점

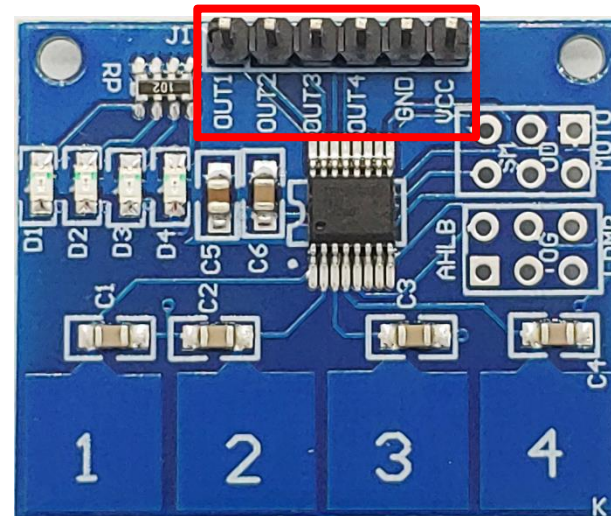
- ❖누르는 키감이 없음



# 터치센서 모듈 핀 아웃

## ■ 다음과 같이 연결

- ❖ OUT1 - 4
- ❖ OUT2 - 5
- ❖ OUT3 - 6
- ❖ OUT4 - 7
- ❖ GND - GND
- ❖ VCC - 5V





## 예제 코드

```
#define TOUCH_PIN1 4
#define TOUCH_PIN2 5
#define TOUCH_PIN3 6
#define TOUCH_PIN4 7
#define LED_PIN1 8
#define LED_PIN2 9
#define LED_PIN3 10
#define LED_PIN4 11

void setup() {
  pinMode(TOUCH_PIN1, INPUT);
  pinMode(TOUCH_PIN2, INPUT);
  pinMode(TOUCH_PIN3, INPUT);
  pinMode(TOUCH_PIN4, INPUT);
  pinMode(LED_PIN1, OUTPUT);
  pinMode(LED_PIN2, OUTPUT);
  pinMode(LED_PIN3, OUTPUT);
  pinMode(LED_PIN4, OUTPUT);
}
```

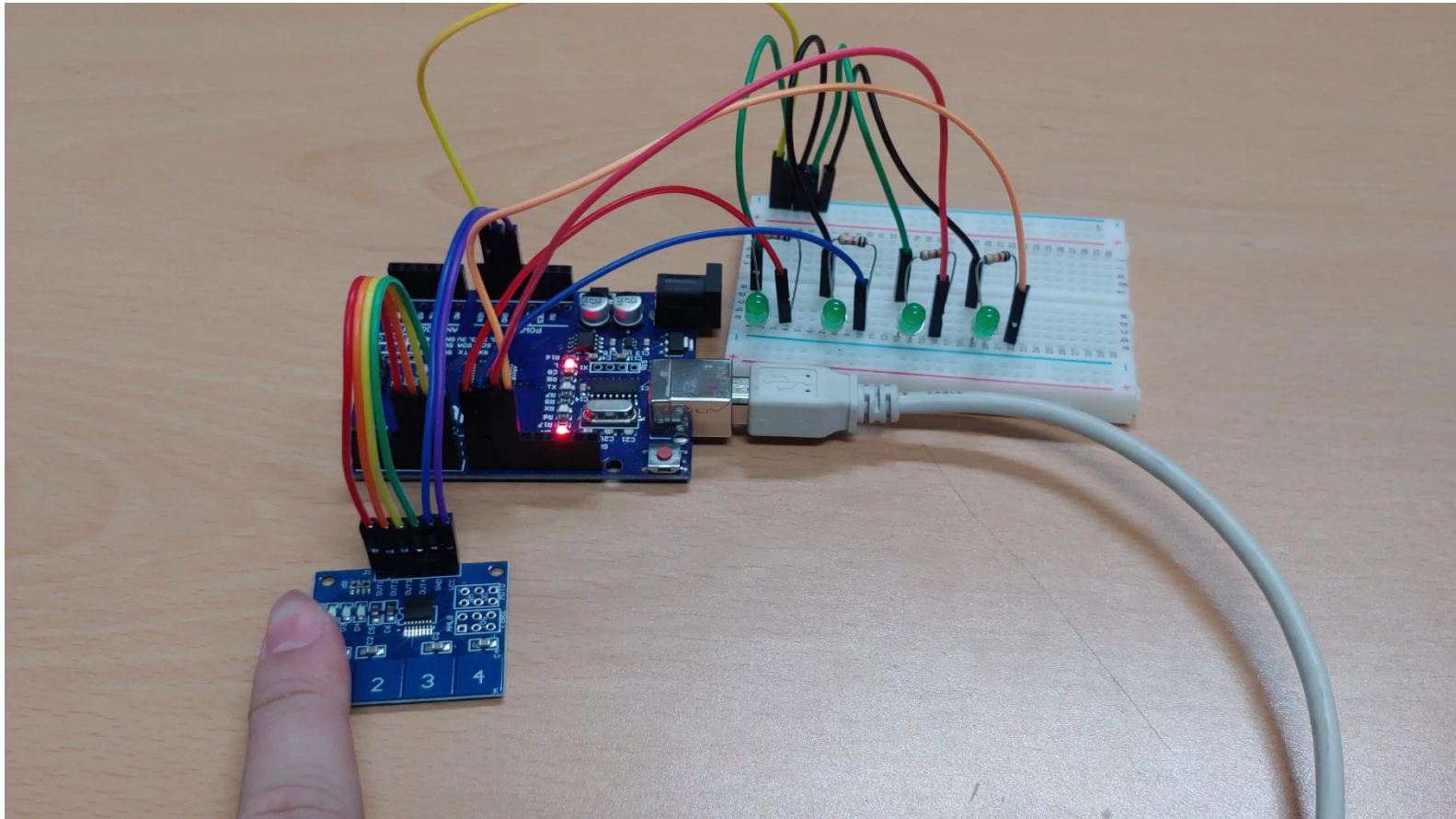
```
void loop() {
  digitalWrite(LED_PIN1, LOW);
  digitalWrite(LED_PIN2, LOW);
  digitalWrite(LED_PIN3, LOW);
  digitalWrite(LED_PIN4, LOW);

  if(digitalRead(TOUCH_PIN1) == HIGH){
    digitalWrite(LED_PIN1, HIGH);
  }
  if(digitalRead(TOUCH_PIN2) == HIGH){
    digitalWrite(LED_PIN2, HIGH);
  }
  if(digitalRead(TOUCH_PIN3) == HIGH){
    digitalWrite(LED_PIN3, HIGH);
  }
  if(digitalRead(TOUCH_PIN4) == HIGH){
    digitalWrite(LED_PIN4, HIGH);
  }

  delay(50);
}
```



# 결과 확인



---

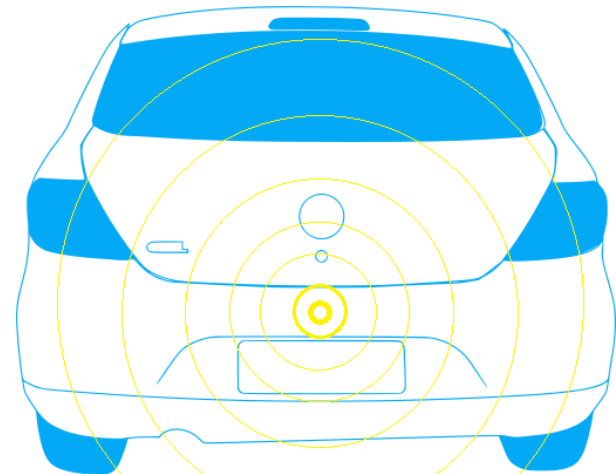
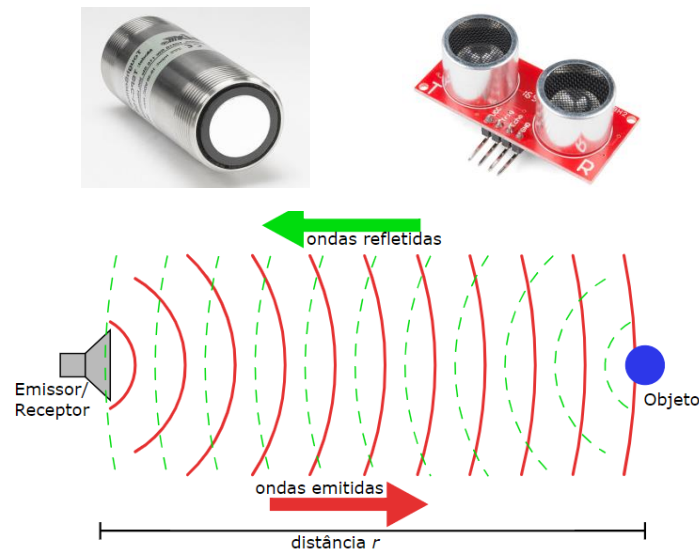
## ■터치와 릴레이 연결하기



101111000101001010011000101001111010101010111110011111010100001111100001000010101011001111101010000111110000100001010101111100111110101000011111000010000101011001111101010000111110000100001010110011111010100001111100001000010101010000101

# 사용 사례

- 자동차 후방감지 센서 등
- 센서 전면에 있는 물체의 거리를 측정할 수 있다.
- 음파를 출력해 반사되어 돌아오는 음파를 측정하여 거리를 계산한다.



자동차 그림: <https://svgsilh.com/ko/03a9f4/image/1029941.html>

초음파 센서: [https://commons.wikimedia.org/wiki/File:Senix ToughSonic 30 Ultrasonic Sensor.jpg](https://commons.wikimedia.org/wiki/File:Senix_ToughSonic_30_Ultrasonic_Sensor.jpg)

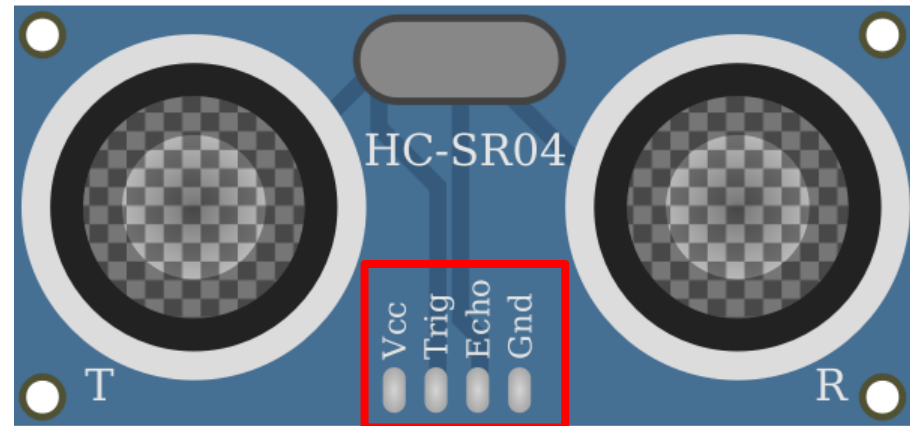
원리: [https://upload.wikimedia.org/wikipedia/commons/a/af/Sonar Principle\\_pt-BR.svg](https://upload.wikimedia.org/wikipedia/commons/a/af/Sonar_Principle_pt-BR.svg)

초음파센서: [https://commons.wikimedia.org/wiki/File:SparkFun HC-SR04 Ultrasonic-Sensor 13959-01a.jpg](https://commons.wikimedia.org/wiki/File:SparkFun_HC-SR04_Ultrasonic-Sensor_13959-01a.jpg)

# 초음파센서 모듈 핀 아웃

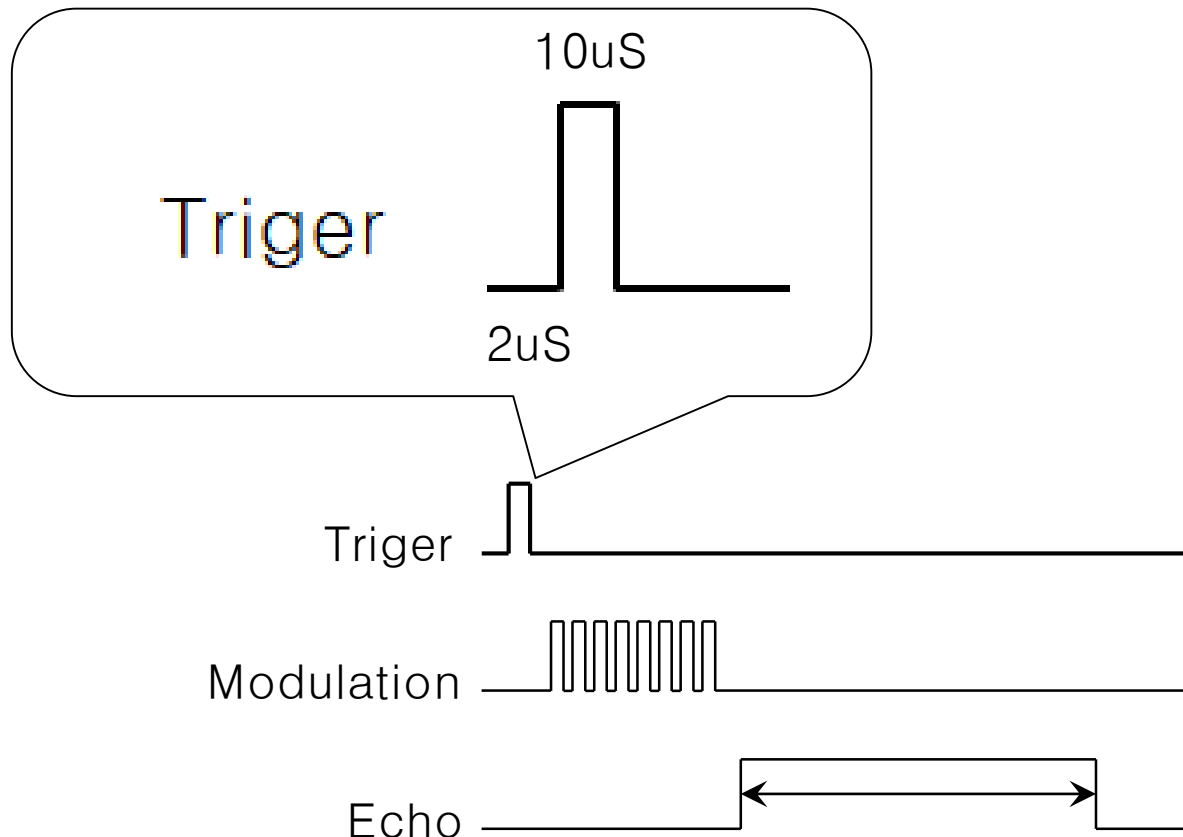
## ■ 다음과 같이 연결

- ❖ VCC – 5V
- ❖ Trig – 8
- ❖ Echo – 9
- ❖ GND – GND

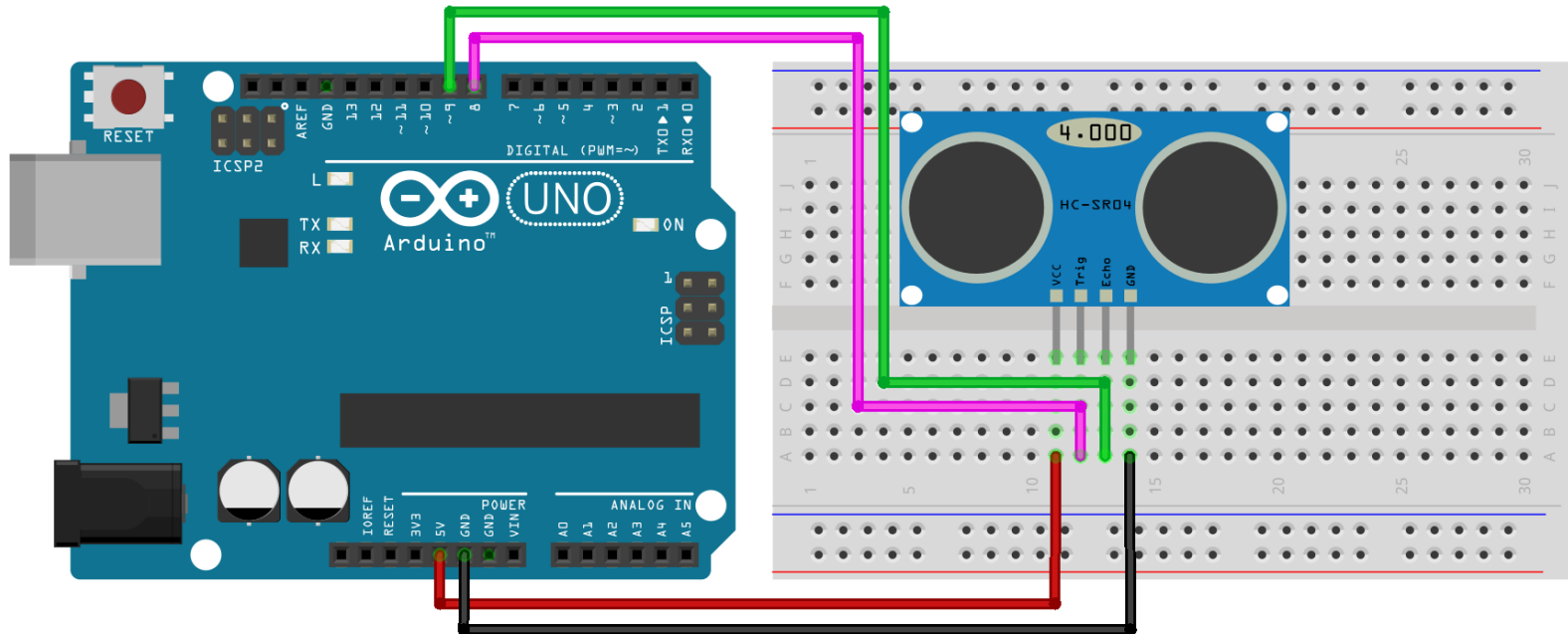


# Triger 핀제어

- HC-SR04는 초음파 발진을 위해 Triger 핀을 타이밍에 맞춰 제어를 해주어야 한다.



# 연결도



## 예제 코드

```
#define TRIG_PIN 8
#define ECHO_PIN 9

int distance;

void setup() {
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  distance = pulseIn(ECHO_PIN, HIGH) / 58;

  if(distance <= 2000){
    Serial.print("Distance(cm) = ");
    Serial.println(distance);
  }
  else{
    Serial.println("ERROR!!");
  }

  delay(1000);
}
```

# 결과 확인

