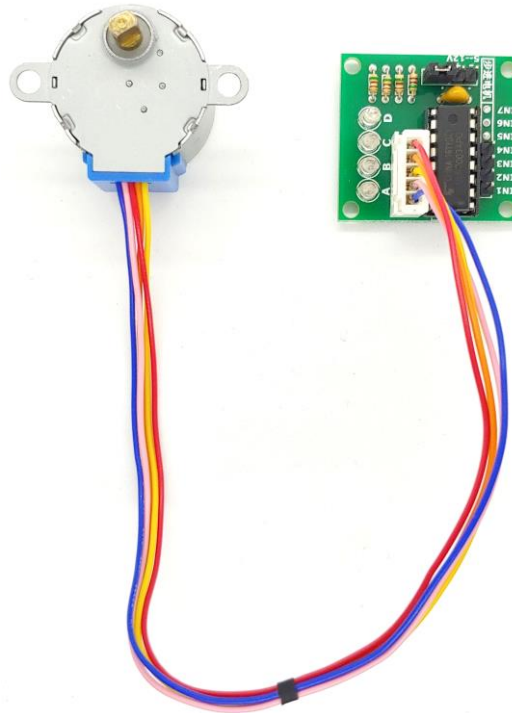


# 1. 스텝 모터

# 스텝 모터 란?

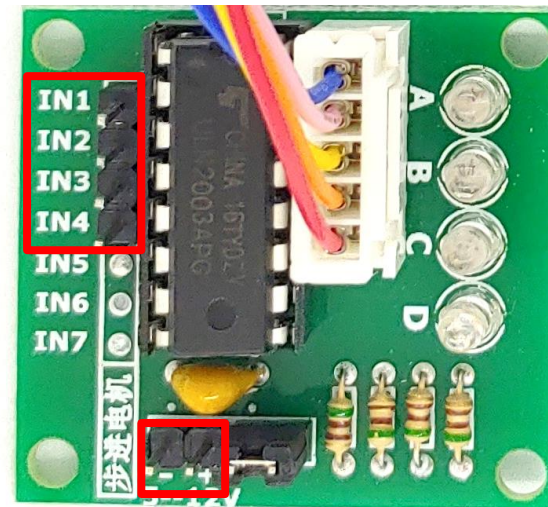
- 특정 각도 단계(step)단위로 회전하는 모터.
- 펄스신호로 인해 동작함.
- Resolution(해상도)에 따라 한 펄스에 회전하는 각도가 다름.
  - ❖ 해상도에 높을 수록 세밀하게 제어가 된다.



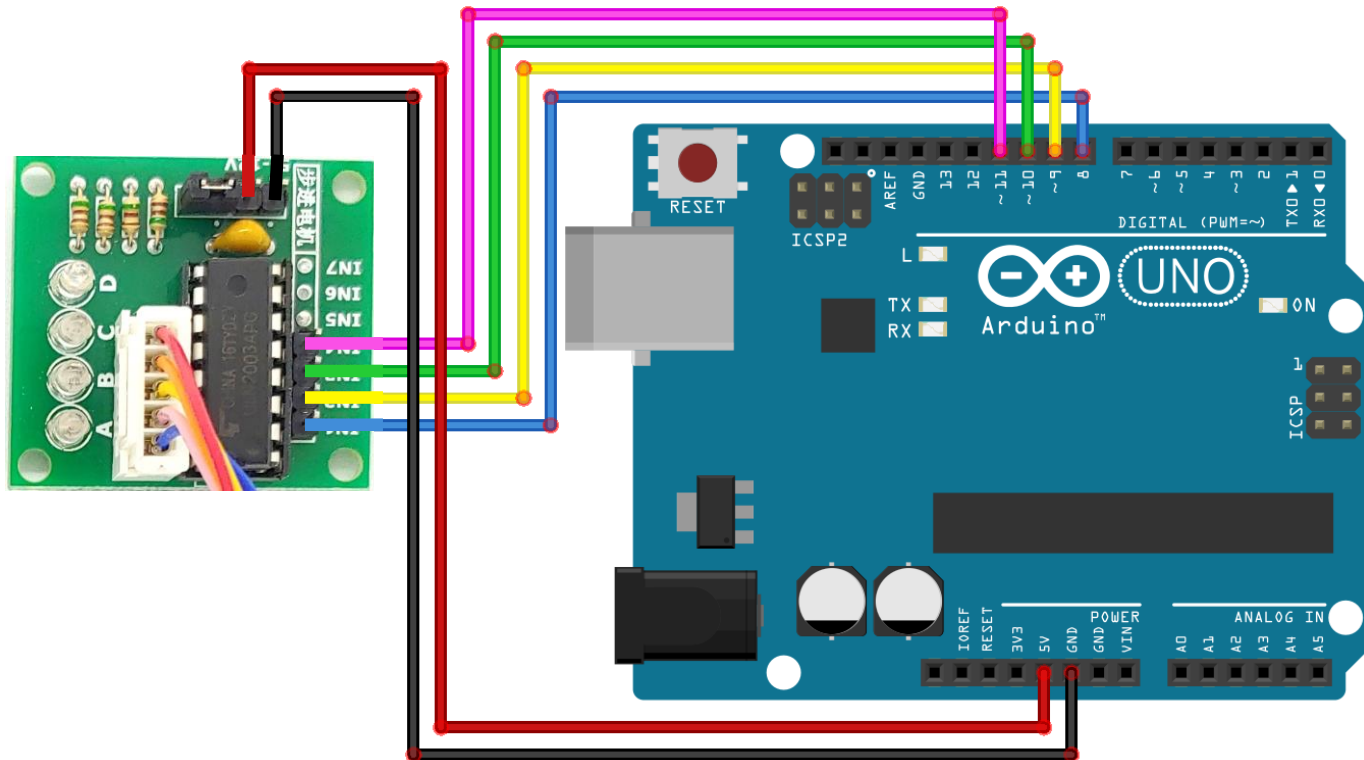
# 스텝 모터 드라이버 핀 아웃

## ■ 다음과 같이 연결

- ❖ IN1 - 8
- ❖ IN2 - 9
- ❖ IN3 - 10
- ❖ IN4 - 11
- ❖ + - 5V
- ❖ - - GND



# 연결도



# 예제 코드

```
#include <Stepper.h>

#define IN1 8
#define IN2 9
#define IN3 10
#define IN4 11

//360 ° 회전
const int stepsPerRevolution = 2048;

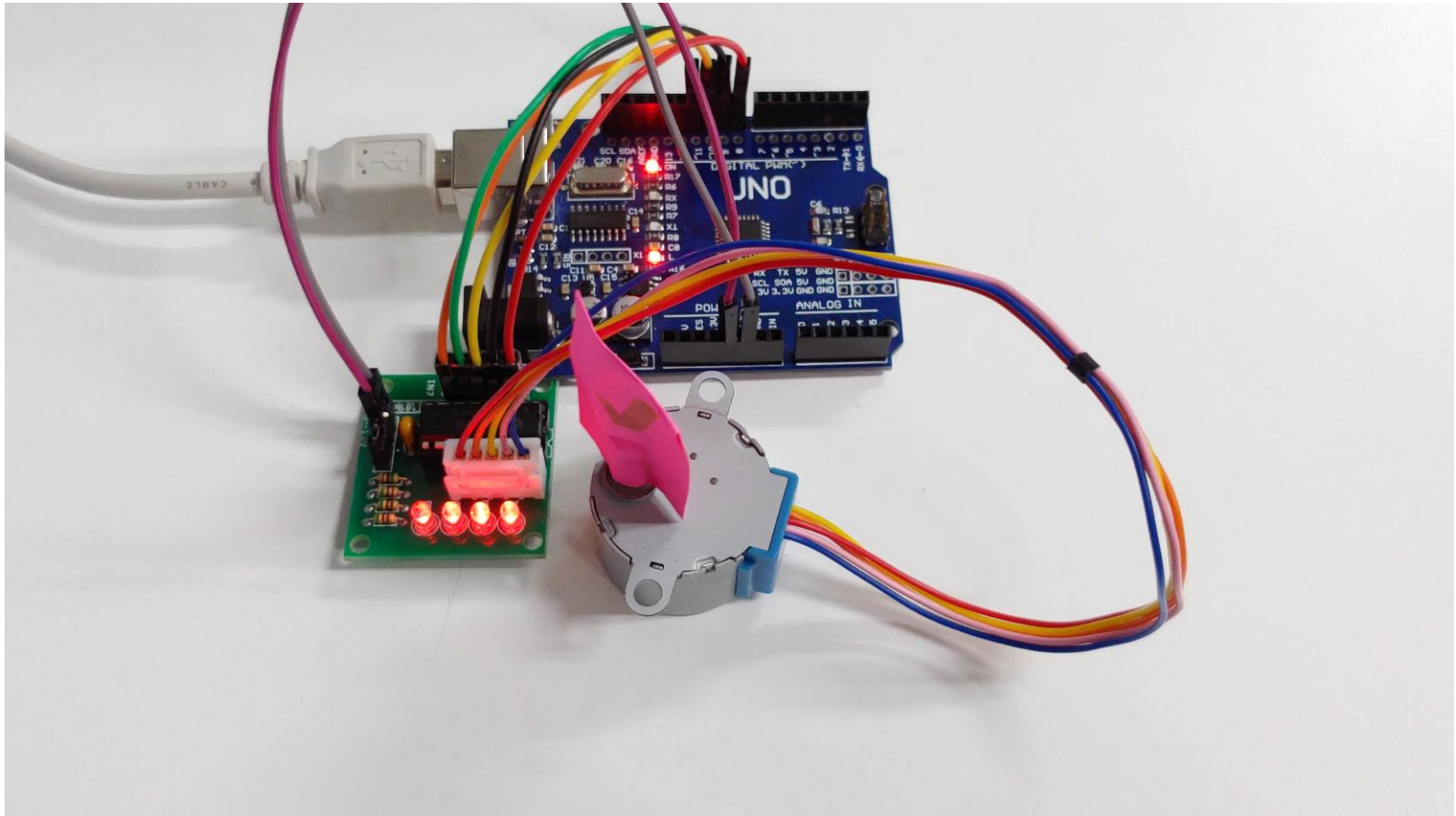
Stepper myStepper(stepsPerRevolution,IN4,IN2,IN3,IN1);

void setup() {
  myStepper.setSpeed(14);
}

void loop() {
  myStepper.step(stepsPerRevolution); //시계 반대방향
  delay(500);

  myStepper.step(-stepsPerRevolution); //시계 방향
  delay(500);
}
```

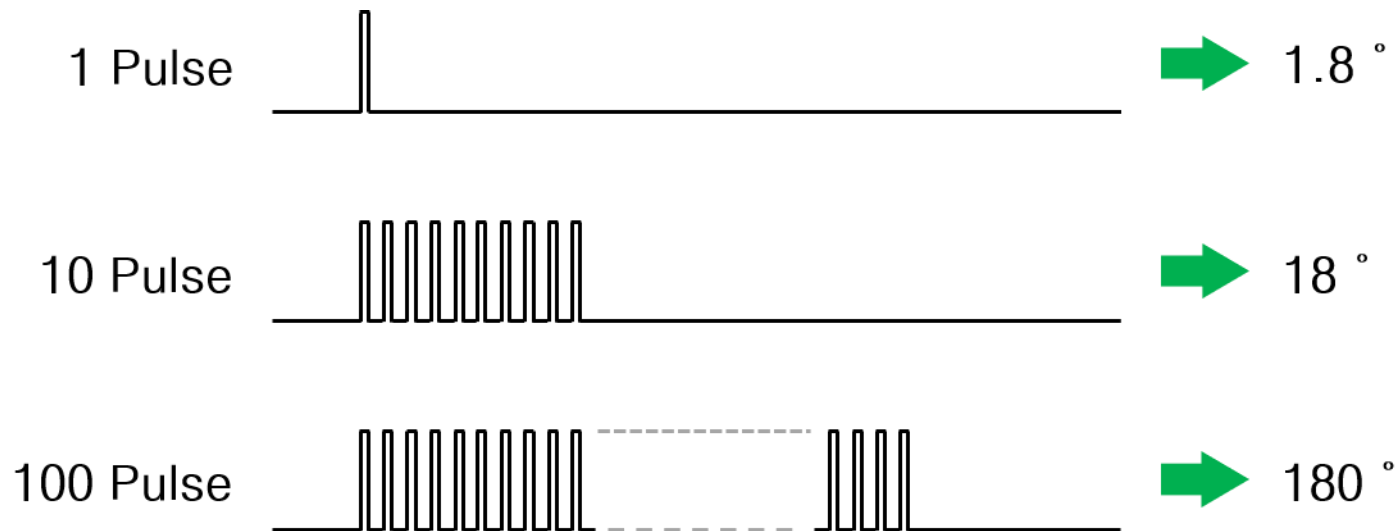
# 동작 확인



# 스텝 모터의 원리

## ■펄스신호를 받으면 한 단계씩 움직임

- ❖일반적으로 많이 사용되는 스텝모터는 2상 또는 5상이다.
- ❖만약 2상의 스텝모터가  $1.8^{\circ}/\text{step}$  이면 한 바퀴( $360^{\circ}$ )는 200step 필요하다.

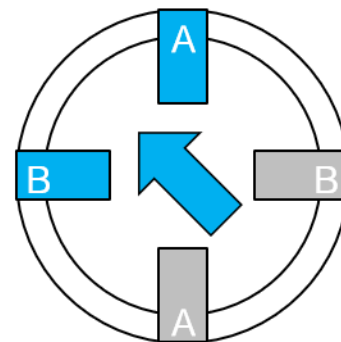
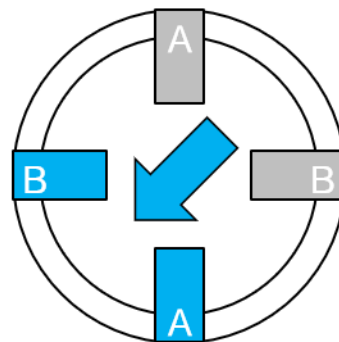
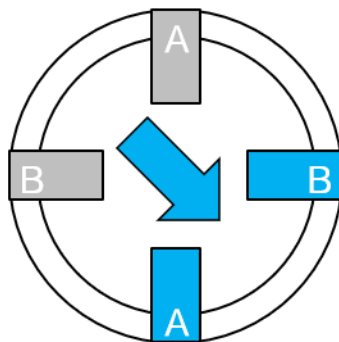
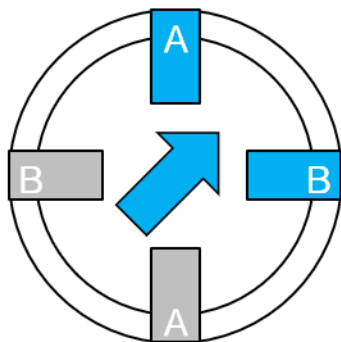


# 스텝 모터의 원리

## ■ 구동 방식

### ❖ 2상 여자방식 – Full Step

METHOD	PHASES	1	2	3	4	5	6	7	8
2상 여자방식 Full Step	BLUE	ON	ON	OFF	ON	ON	ON	ON	ON
	PINK	ON	ON	ON	OFF	ON	ON	ON	ON
	YELLOW	ON	ON	ON	ON	ON	ON	ON	ON
	ORANGE	ON	ON	ON	ON	ON	ON	ON	ON

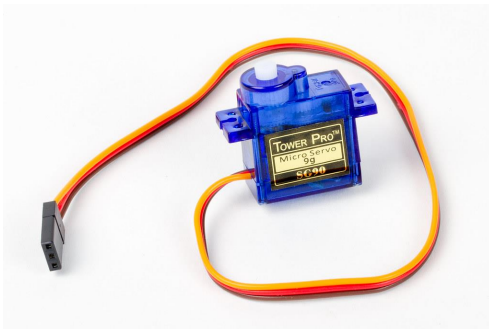




1011110001010010100110001010011110101010101111100111110101000011111000010000101010110011111010100001111100001000010101011111001111101010000111110000100001010101111100111110101000011111000010000101011001111101010000111110000100001010101010000101

# 서보모터 란?

- 지정된 각도만큼 회전할 수 있는 모터.
- 자동화 생산 시스템, 로봇, 장난감, 가전제품 등 광범위하게 쓰임.
- 회전 반경이 약  $-90 \sim 180$ 도 범위로 정해져 있음.



예시1. 서보모터

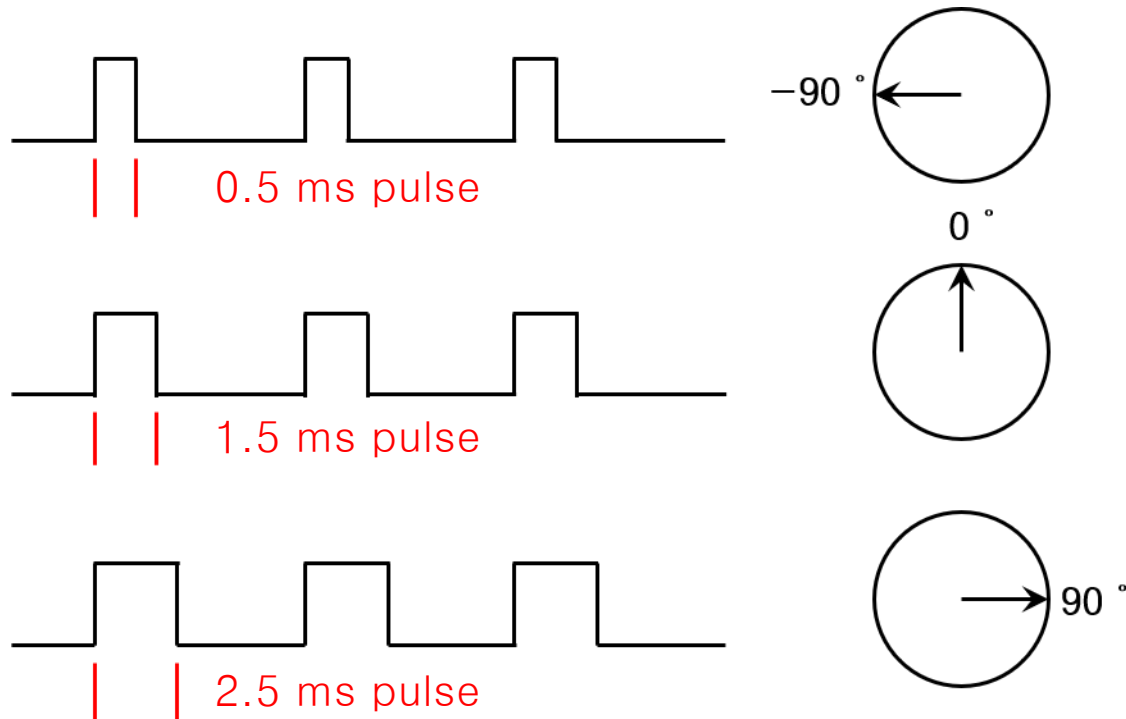


예시2. RC카의 앞 바퀴 각도 조절

# 서보모터의 원리

## ■ PWM(펄스폭변조) 방식으로 제어

- ❖ 펄스폭에 따라 모터의 회전 각도가 달라진다.
- ❖ 모터의 축을 특정 각도로 이동시킬 수 있다.



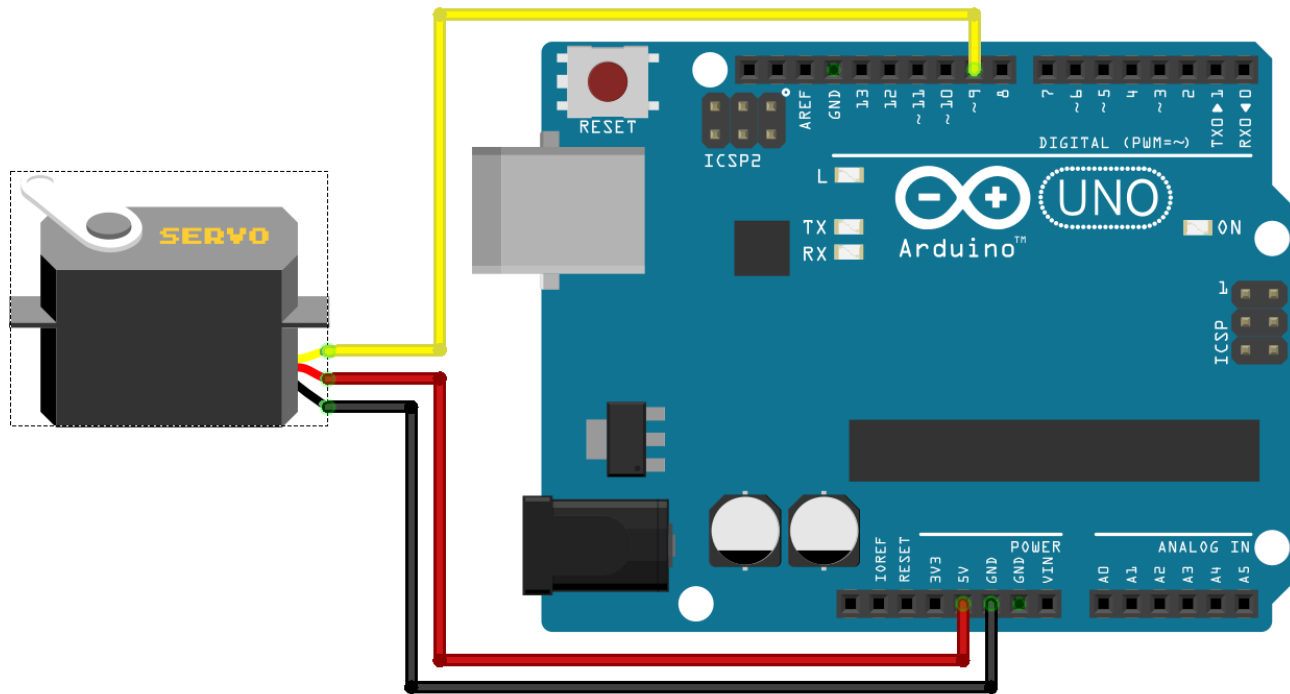
# 서보모터 핀 아웃

## ■ 다음과 같이 연결

- ❖ 갈색 - GND
- ❖ 빨간색 - 5V
- ❖ 노란색 - 9



# 연결도



# 예제 코드

```
#include <Servo.h>

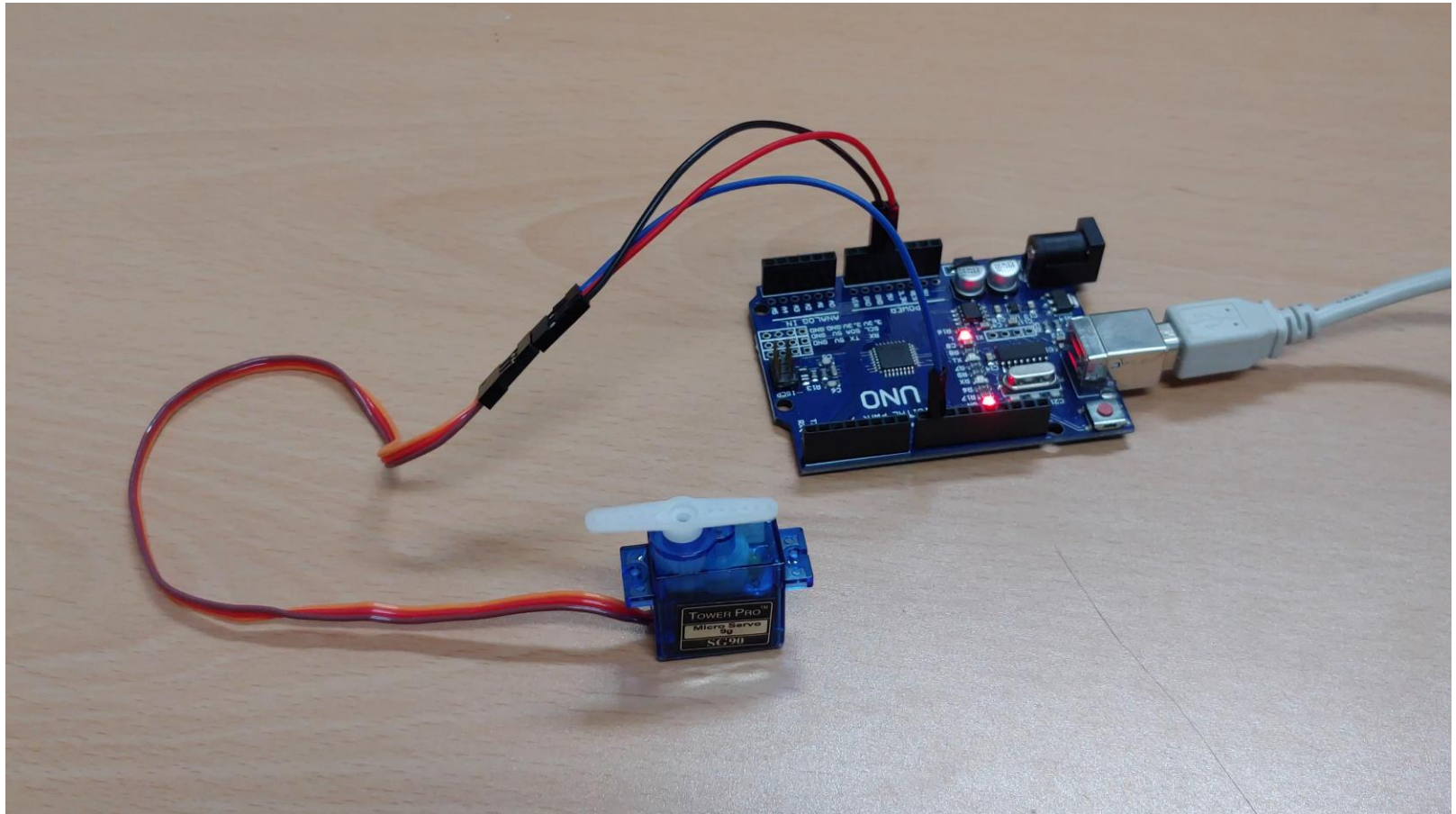
#define SERVO_PIN 9

Servo servo;
int pos = 0;

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

void loop() {
  for(pos = 0; pos < 180; pos += 1) {
    servo.write(pos);
    delay(5);
  }
  for(pos = 180; pos >= 1; pos -= 1) {
    servo.write(pos);
    delay(5);
  }
}
```

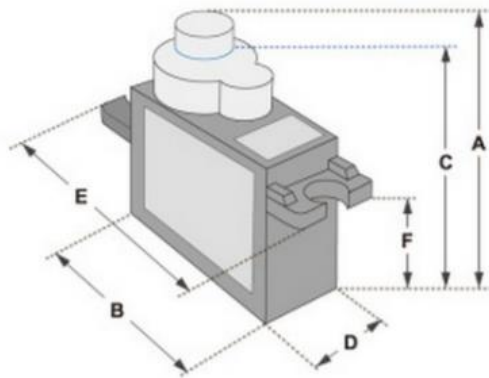
# 동작 확인





# SG-90 서보모터

■ 180도의 움직임을 할 수 있다.

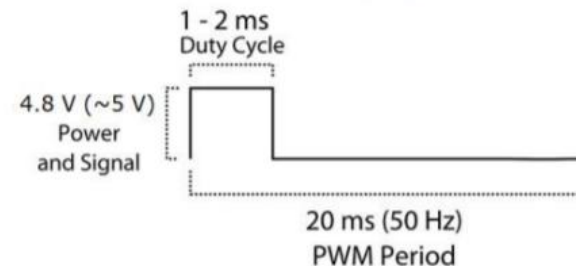


Dimensions & Specifications	
A (mm) :	32
B (mm) :	23
C (mm) :	28.5
D (mm) :	12
E (mm) :	32
F (mm) :	19.5
Speed (sec) :	0.1
Torque (kg-cm) :	2.5
Weight (g) :	14.7
Voltage :	4.8 - 6

Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is middle, is all the way to the right, "-90" (~1ms pulse) is all the way to the left.

Datasheet를 보면 펄스폭의 범위가 1ms~2ms라고 나와 있지만 실제로는 0.5ms~2.5ms이다.

PWM=Orange (⌋⌋)  
Vcc = Red (+)  
Ground=Brown (-)





## 과제 - 스위치를 연결해서 서보모터를 제어해보자

---

- GPIO 9번에 서보모터, GPIO 8번에 스위치를 연결한다.
- 스위치를 누르면 90도 위치로 이동하고 스위치를 떼면 원래 위치로 이동하도록 수정해보자.

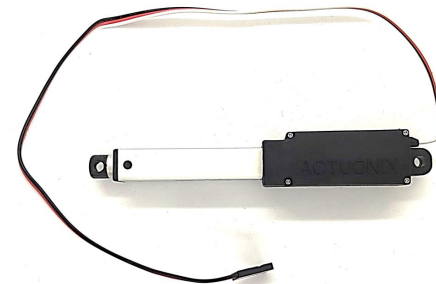
### 3. 리니어 서보 액추에이터

# 리니어 액추에이터 란?

- 액추에이터 (actuator): 제어 기기에서 출력된 신호를 바탕으로 대상에 물리적인 움직임을 주는 기계 장치를 일컫는 용어. 앞서 나온 모터들도 액추에이터에 속함.
- 리니어 액추에이터는 모터의 회전운동을 직선운동으로 변화시켜 push/pull 동작을 수행하는 장치이다.



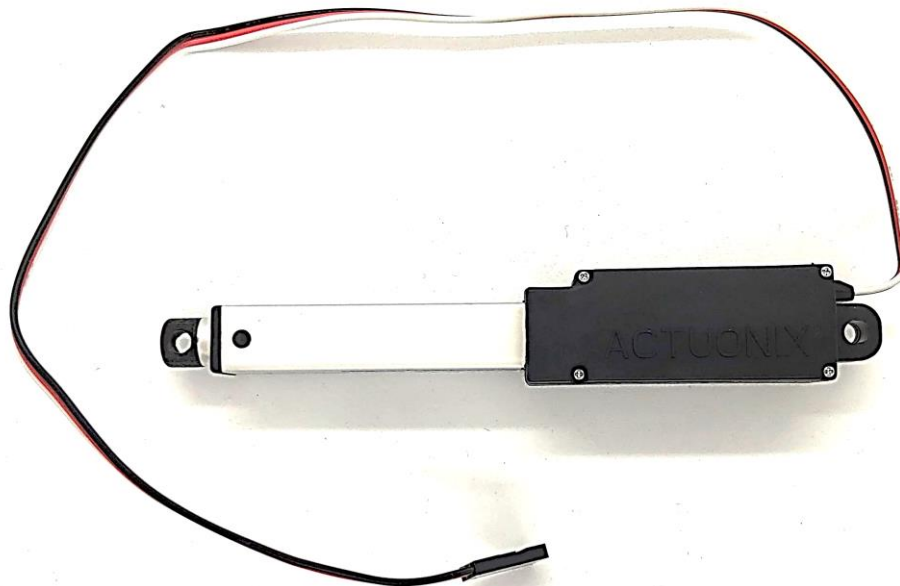
예시1. 수중 리니어 액추에이터



예시2. 리니어 서보 액추에이터

# L12-R 리니어 서보 액추에이터

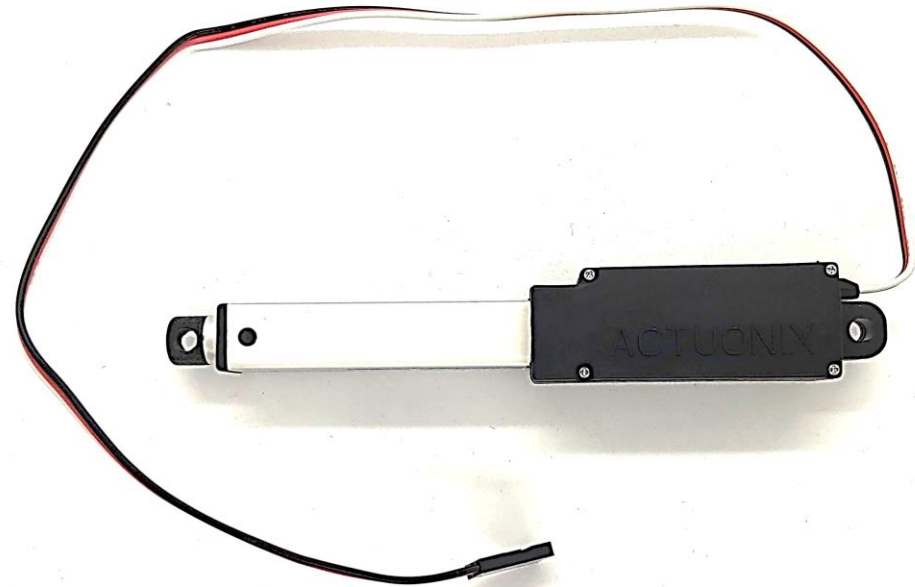
- 일반적인 회전운동을 하는 서보 인터페이스와 인터페이스가 동일하기 때문에 3선 서보모터(SG-90)처럼 연결이 가능하다.



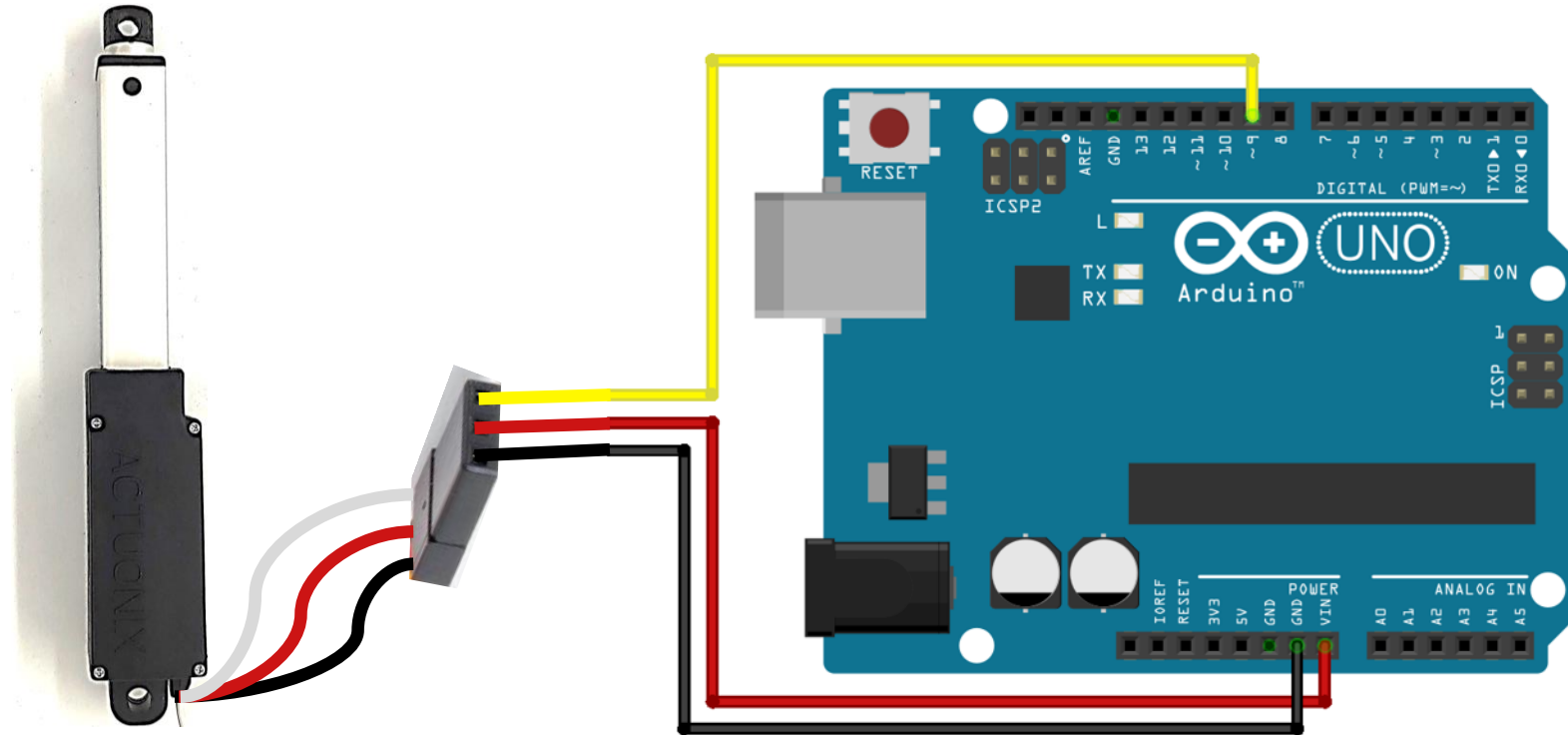
# 리니어 서보 액추에이터 핀 아웃

## ■ 다음과 같이 연결

- ❖ 검정색 - GND
- ❖ 빨간색 - VIN
- ❖ 흰색 - 9



# 연결도



# 예제 코드

```
#include <Servo.h>

#define SERVO_PIN 9

void SetStrokePerc(float strokePercentage);

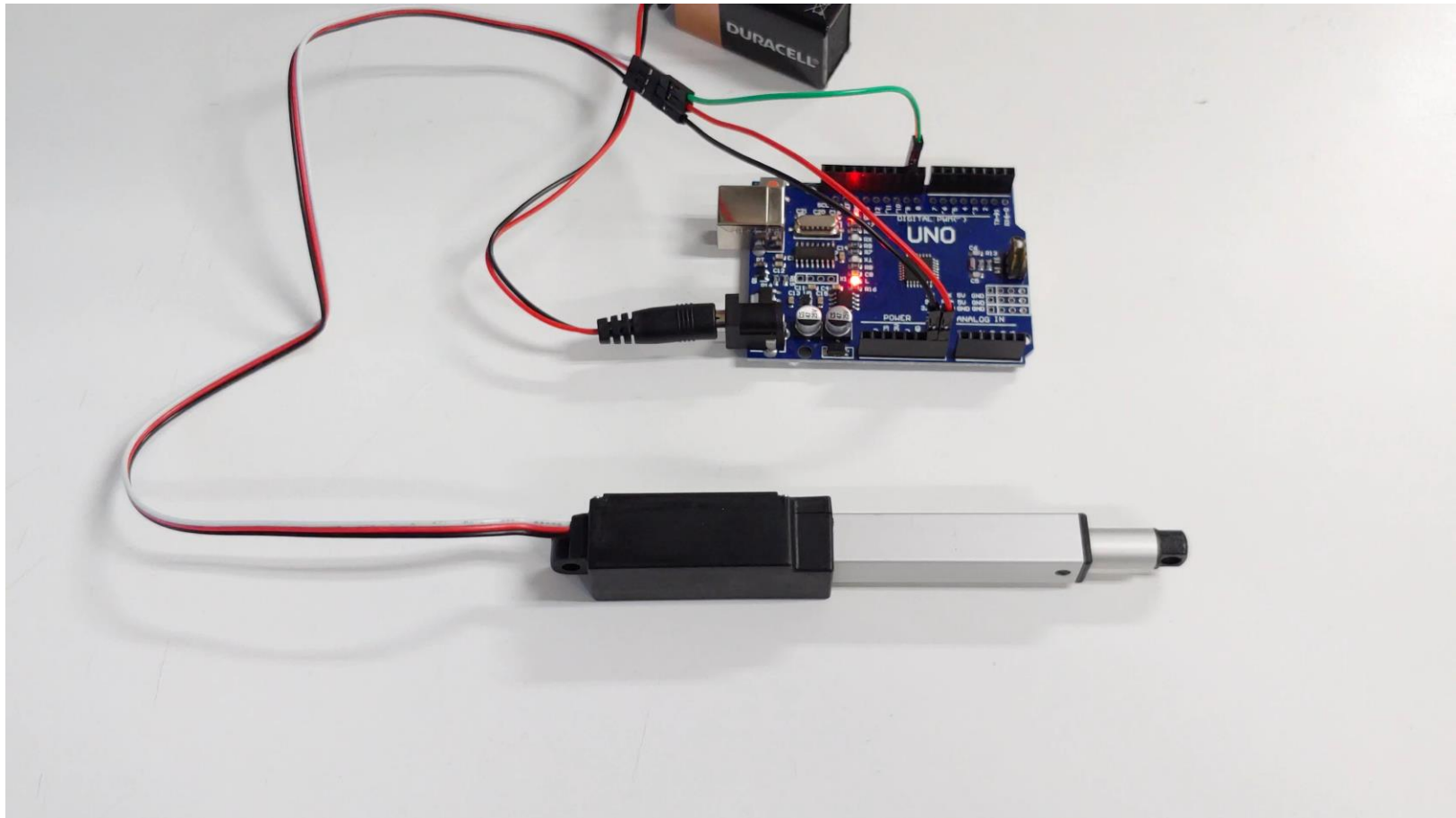
Servo myServo;
int delayMS = 1500;

void setup()
{
    myServo.attach(SERVO_PIN);
}

void loop()
{
    SetStrokePerc(1);
    delay(delayMS);
    SetStrokePerc(25);
    delay(delayMS);
    SetStrokePerc(50);
    delay(delayMS);
    SetStrokePerc(75);
    delay(delayMS);
    SetStrokePerc(99);
    delay(delayMS);
}

void SetStrokePerc(float strokePercentage)
{
    if ( strokePercentage >= 1.0 && strokePercentage <= 99.0 )
    {
        int usec = 1000 + strokePercentage * ( 2000 - 1000 ) / 100.0 ;
        myServo.writeMicroseconds( usec );
    }
}
```

# 동작 확인





## 주의사항

---

- 아두이노의 5V 전원으로도 동작하는 가벼운 마이크로 서보 모터(SG-90)와는 달리 L12-R 서보는 전류를 많이 사용.
- 코드를 보드에 업로드 한 후 USB를 제거하고 9V DC를 인가해야 제대로 된 동작을 확인할 수 있음.