

# 소프트웨어 프로젝트 2

## 서보모터, Part I

2024년 2학기

국민대학교  
소프트웨어학부/인공지능학부  
주용수, 최진우, 한재섭, 허대영  
{ysjoo, jaeseob, jnwochoi, dyheo}@kookmin.ac.kr

# 서보 모터 (Servo Motor)

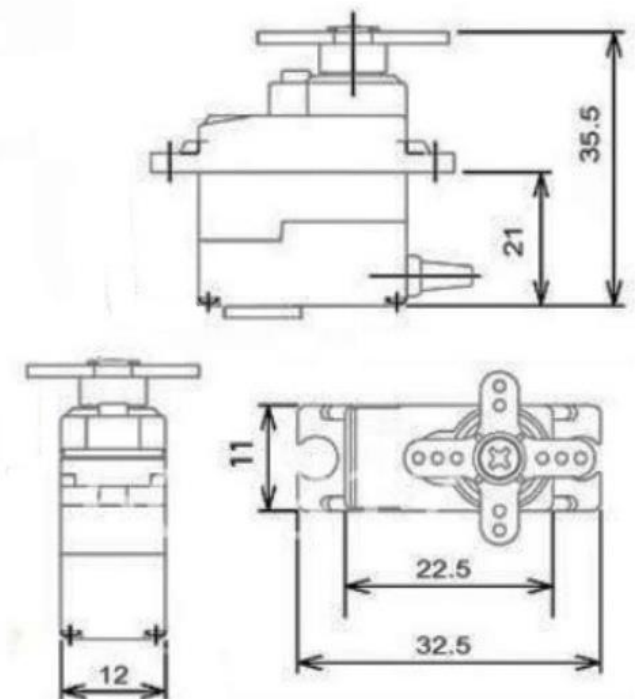
- 액추에이터 (Actuator)의 한 종류
  - 입력된 제어량에 따라 회전각을 조절하여 위치제어를 수행
  - (참고) 리니어 액추에이터 (Linear Actuator) :  
모터 회전운동이 직선운동으로 변환됨
- 회전각
  - $0^{\circ} \sim 180^{\circ}$  ( $360^{\circ}$  연속회전 가능한 서보도 있음)
- 제어 인터페이스
  - 펄스 폭 변조(PWM) 방식으로 제어
  - 아두이노 IDE에는 서보 제어를 위한 전용 PWM 라이브러리가 제공됨
    - analogWrite()는 PWM 제어 방식이지만  
서보 제어를 위한 주기 및 duty가 호환되지 않음

# 서보 모터

[https://www.electronicoscaldas.com/datasheet/MG90S\\_Tower-Pro.pdf](https://www.electronicoscaldas.com/datasheet/MG90S_Tower-Pro.pdf)

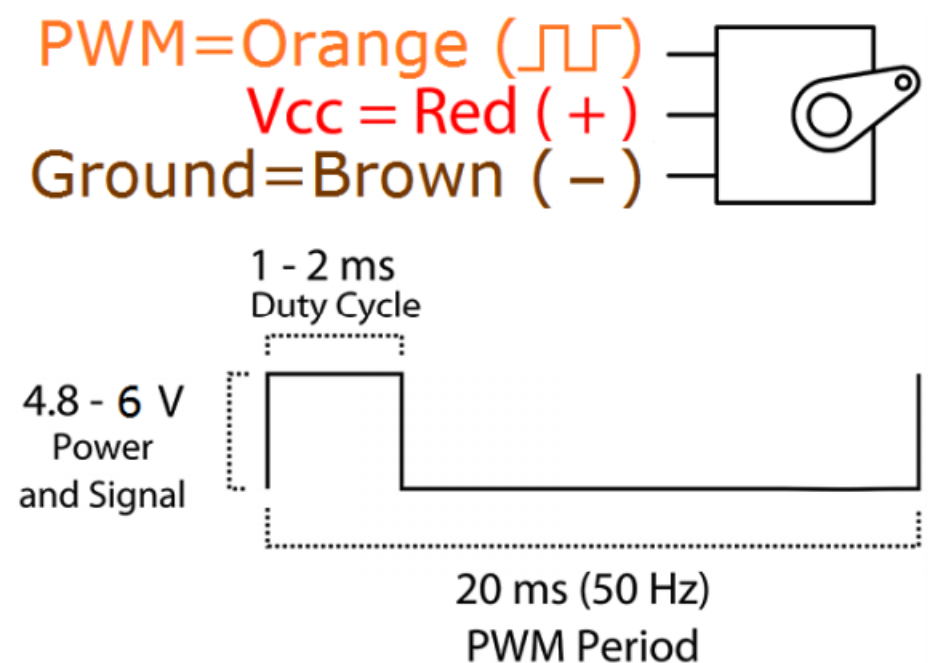
- MG90S 기술적 사양
  - 무게: 13.4g
  - 크기: 22.5x12x35.5 (mm)
  - 속도: 0.1s/60° @4.8V
  - Stall torque: 1.8kg • cm @4.8V
    - 4.8V 전압 공급시 축으로부터 1cm 길이의 서보암에 달린 1.8kg의 무게추를 버틸 수 있음
  - 입력 전압: 4.8 ~ 6.0V
  - Dead band width: 5us
    - 다음 장에 설명

**MG90S**  
Metal Gear Servo



# 제어 인터페이스

- 펄스 폭 변조 (PWM: Pulse Width Modulation)
  - 20ms 주기 (=50 Hz)
    - 위치제어 명령은 초당 50회 갱신 가능
  - 1ms: 반시계방향 최대회전 ( $0^{\circ}$ )
  - 2ms: 시계방향 최대회전 ( $180^{\circ}$ )
  - 1.5ms: 중립 위치 ( $90^{\circ}$ )
- Dead band width
  - 5us 이내의 입력변화에는 반응하지 않음
- PWM 인터페이스의 장단점
  - 아날로그 vs. 디지털
  - 병렬 vs. 직렬통신
  - PWM 파형 생성 오버헤드



# 서보 동작전류

- 대기전류
  - 서보에 전원이 연결되어 있는 상태
  - MG90S: 10mA
- 동작전류
  - 서보에 적정 부하가 가해진 상태에서 구동하는 상태
  - MG90S: 120-250mA
- 정동전류 (stall current)
  - 서보에 큰 부하가 걸려 모터가 정지해 있는 상태
  - MG90S: 700mA
- 시스템 전원공급 설계를 위해 정확한 이해 필요

# 아두이노 서보 라이브러리

- 아두이노 서보제어 API
  - **attach()**: 디지털 I/O핀을 서보 객체에 연결
  - **write()**: 정수형의 각도 값으로 서보 위치 제어
    - 입력범위: 0, 1, 2, ..., 180 (예: 12.7 => 12로 인식)
  - **writeMicroseconds(pulse\_width)**: 펄스 폭으로 서보 위치 제어
    - 최소값: 1,000 (us): 반시계방향 최대 회전
    - 최대값: 2,000 (us): 시계방향 최대 회전
    - 중간값: 1,500 (us): 중립
    - ➔ 서보 모터의 품질, 조립 상태 등에 따라 **전혀** 정교하지 않을 수 있음
  - **read()**: 현재 서보 각도 읽기 (**write()** 한 값을 리턴)
  - **attached()**: 서보 객체에 I/O핀이 연결되어 있는지 확인
  - **detach()**: 서보 객체에서 해당 I/O핀 분리

<https://www.arduino.cc/en/reference/servo>

# 실습

- 서보+초음파센서 회로 구성
- 서보 동작 확인
  - $0^{\circ}$ ,  $90^{\circ}$ ,  $180^{\circ}$  각도 제어 테스트
- 서보 회전속도 측정

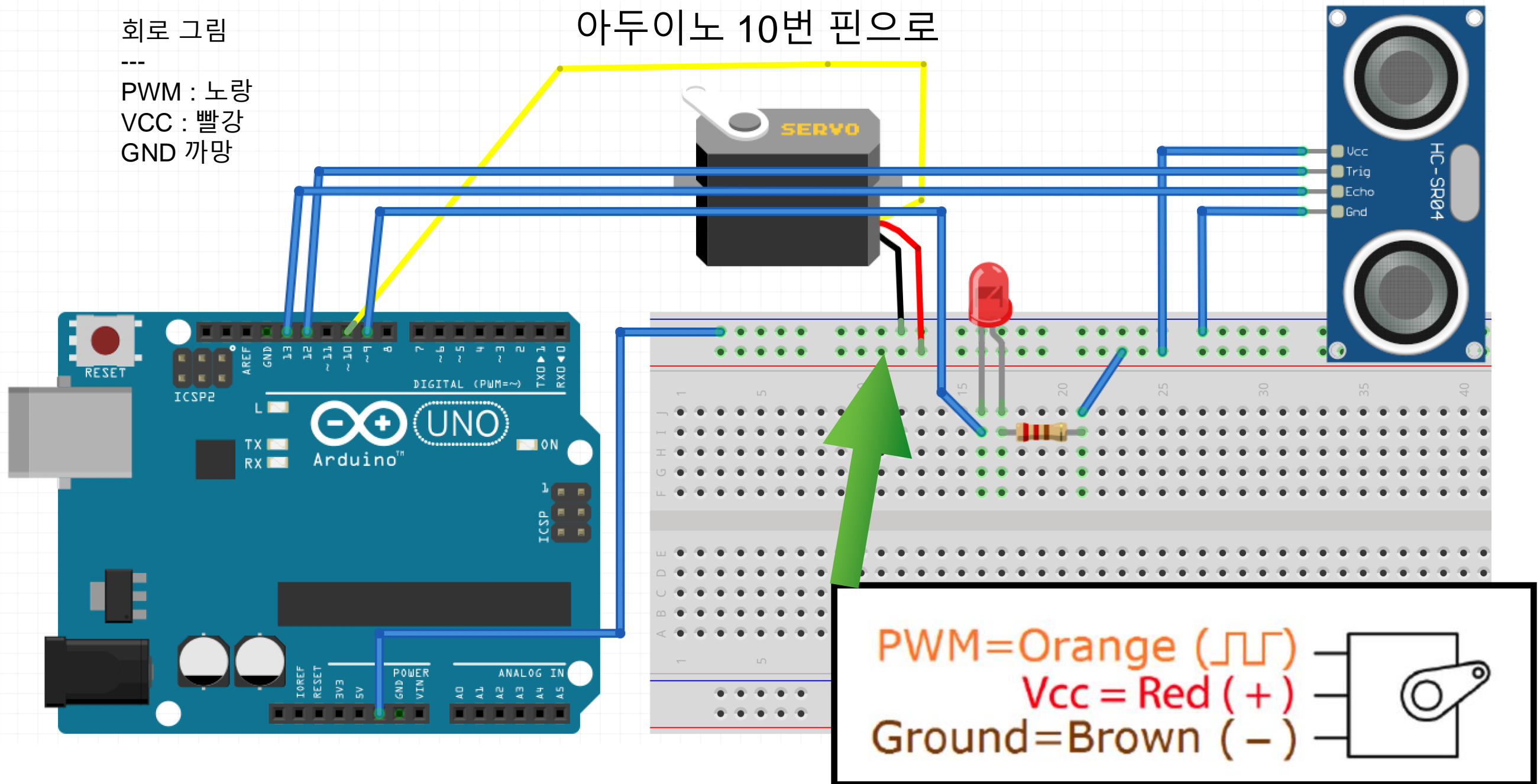
# 실습 1: 서보+초음파센서 회로 구성

회로 그림

---

PWM : 노랑  
VCC : 빨강  
GND 까망

아두이노 10번 핀으로



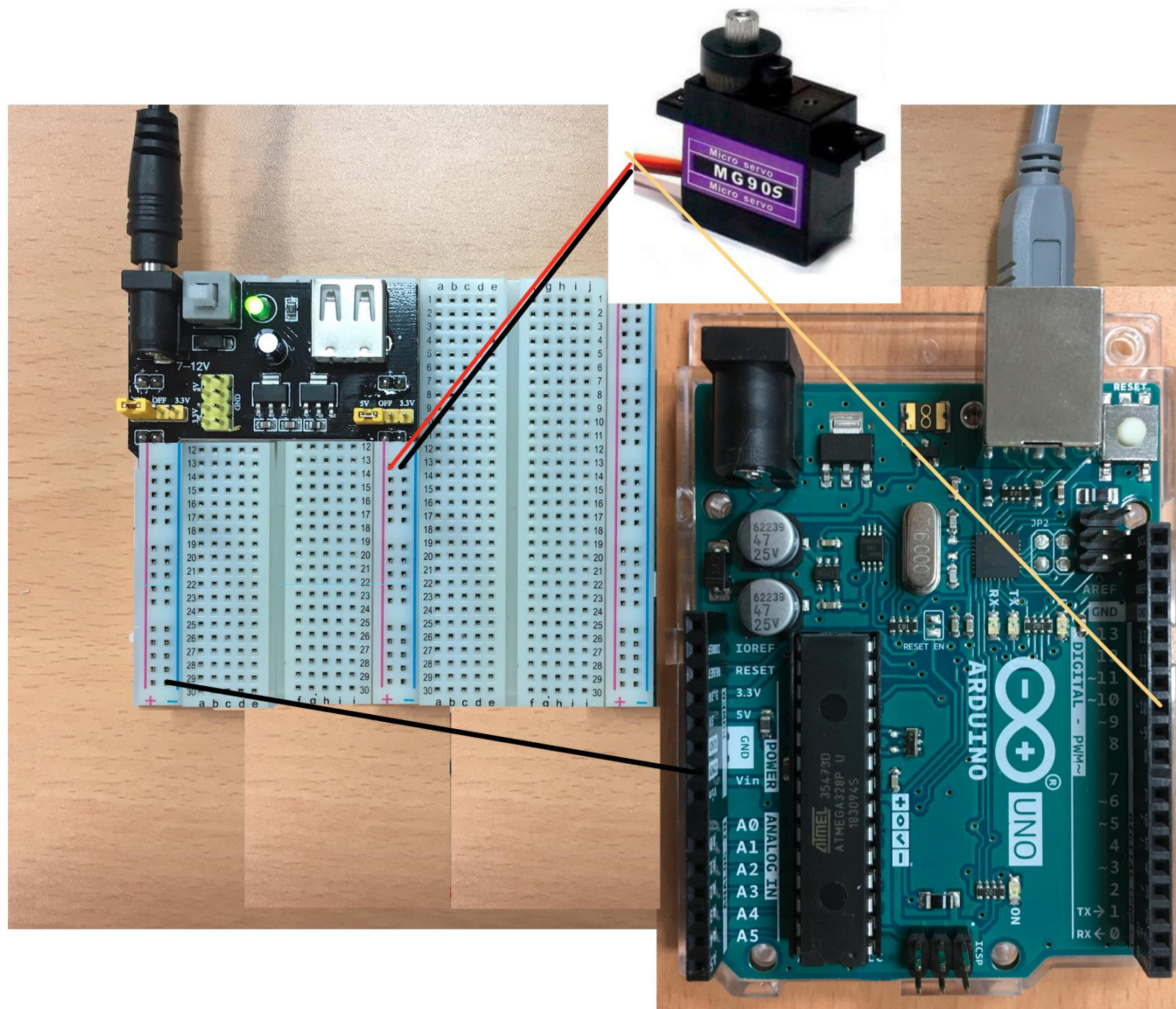
주의: 위 회로도에는 전원공급모듈이 생략되어 있음.

모든 주변장치의 전원은 브레드보드의 파워레일을 거쳐 전원공급모듈로부터 제공받도록 연결할 것.



# 전원모듈 - 서보 결선도

- 초음파 센서 및 LED 회로는 직전 슬라이드 참고



# 실습 2: 서보 동작 확인

- 각도 제어 테스트
  - **write()** 사용
  - $0^{\circ} - 90^{\circ} - 180^{\circ} - 90^{\circ} - 0^{\circ}$ 의 순서를 반복
  - **delay()** 함수로 각 위치에서 500ms 동안 정지 후 다음 위치로 이동
- 테스트 코드
  - 10\_example\_2.ino
  - [https://www.dropbox.com/scl/fi/433exyip0w42xwjcrw1cv/10\\_example\\_2.ino?rlkey=tn07nujl7cm0a4dbo3c1x78kc&dl=0](https://www.dropbox.com/scl/fi/433exyip0w42xwjcrw1cv/10_example_2.ino?rlkey=tn07nujl7cm0a4dbo3c1x78kc&dl=0)

# 실습 2: 서보 동작 확인

- 테스트 코드: 10\_example\_2.ino

```
1  #include <Servo.h>
2  #define PIN_SERVO 10
3
4  Servo myservo;
5
6  void setup() {
7      myservo.attach(PIN_SERVO); // 10번 PIN을 서보 제어용으로 초기화
8      myservo.write(90);         // 서보 모터를 '90°'로 제어 !!!
9      delay(1000);
10 }
11
12 void loop() {
13     // add code here.
14 }
```

// 90 대신 0 ~ 180 입력하여  
// 원하는 각도로 제어 가능

# 서보 회전속도 측정

- 서보 회전속도 측정?
  - x degree per second: 1초 동안 몇 도 회전하는지
  - 서보 모터를 제작할 당시에 최대 속도가 결정됨
  - 측정이 필요하다면, 측정을 위한 전략을 직접 고안해야 함
    - 각도를 바꾸는 시간 간격을 조절하면서 서보의 회전속도를 추정할 수는 있음
  - 서보에 입력 가능한 제어량은 희망 위치(각도)뿐임
  - 명시적으로 속도를 제어하는 인터페이스는 제공되지 않음
- 실제 환경에서는 encoder를 이용한 광학적 방법 또는 리졸버 장치를 이용하여 모터의 회전 속도를 측정
  - <https://www.digikey.kr/ko/articles/how-to-precisely-determine-motor-angular-position-and-velocity-with-a-resolver>



# 실습 3: 서보 회전속도 제어

- 속도제어 샘플 코드를 다운로드하여 돌려볼 것
  - 10\_example\_3.ino
    - [https://www.dropbox.com/scl/fi/58e4qh1u9uog393bty2hn/10\\_example\\_3.ino?rlkey=zu2xn bqzs1vx7bx09d6xugg9v&dl=0](https://www.dropbox.com/scl/fi/58e4qh1u9uog393bty2hn/10_example_3.ino?rlkey=zu2xn bqzs1vx7bx09d6xugg9v&dl=0)
    - **map()** 함수의 동작을 확인할 것
      - <https://www.arduino.cc/reference/en/language/functions/math/map/>

```
14 unsigned long MOVING_TIME = 3000;  
15 unsigned long moveStartTime;  
16 int startAngle = 30; // 30°  
17 int stopAngle = 90; // 90°  
18
```

- 이 예제 코드의 한계
  - **MOVING\_TIME**을 줄이면 속도가 빨라지지만 한계가 있음
    - Dead-band, Servo의 물리적 한계

# 도전과제 1: 서보 움직임 개선

- 실습3 코드를 수정하여 서보의 움직임을 sigmoid 함수로 제어할 것
  - 검색 keyword: servo, sigmoid
  - 서보암에 긴 막대(빨대나 아이스바 막대 등)를 달아 움직임 관찰을 용이하게 할 것
- 과제 제출
  - 과제코드: 10C1
  - 제출물
    - Github link, YouTube link
    - 코드 참조 webpage 주소
  - 제출마감: 7주 1강

## Sigmoid Function

