

Chapter 07

따라 하면서 배우는

아두이노

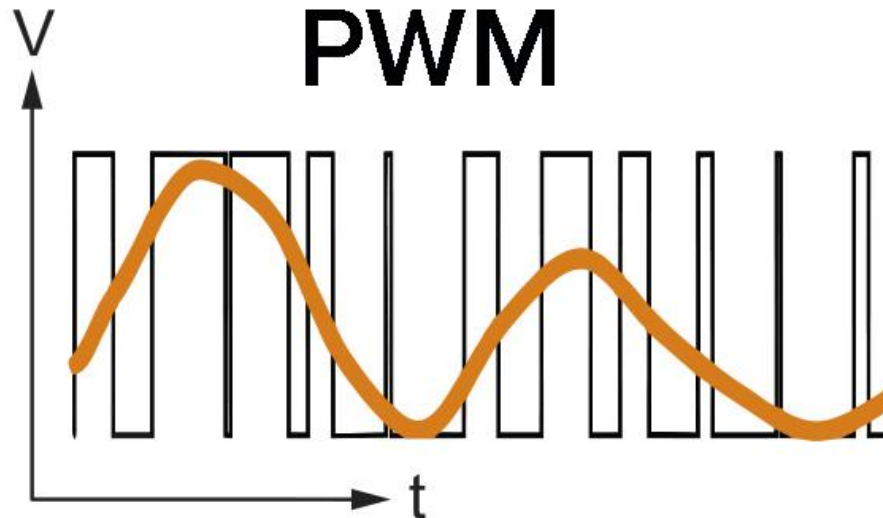
아두이노 메가2560으로 마이크로컨트롤러 시작하기

아날로그 데이터 입출력

아날로그 데이터 출력

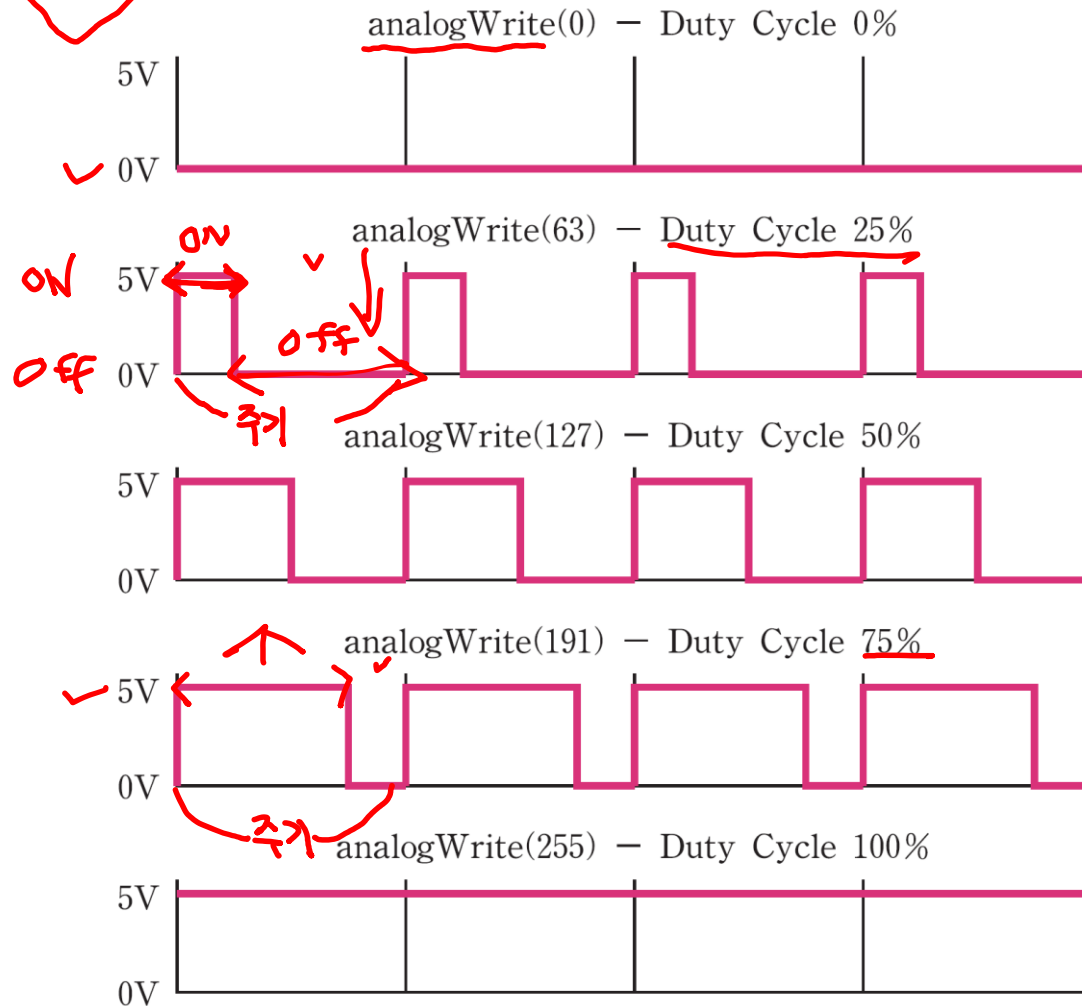
■ 아날로그 데이터 출력

- 아두이노에서 아날로그 데이터 출력은 불가능
- 펄스 폭 변조(PWM) 신호를 통해 아날로그 데이터 출력과 유사한 효과를 얻을 수 있음
 - PWM 신호는 디지털 신호의 일종임
 - PWM 신호 출력 함수가 `analogWrite`이므로 흔히 아날로그 데이터 출력으로 불림



PWM 신호

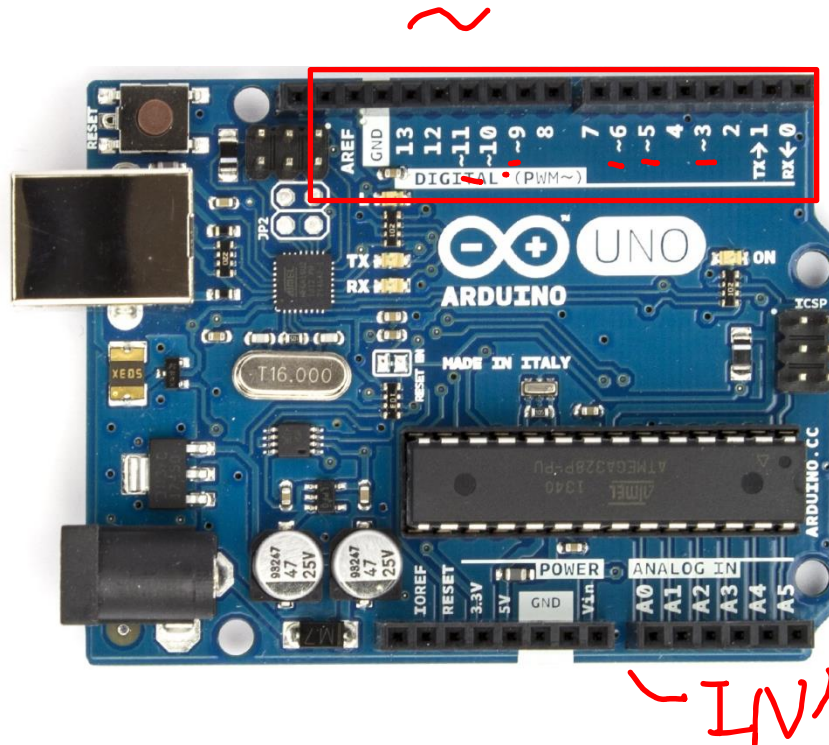
Pulse Width Modulation



아두이노 우노 아날로그 출력핀

■ 아날로그 데이터 출력

- 아두이노 우노에는 PWM 핀이 정해져 있음.
- ~3, ~5, ~6, ~9, ~10, ~11 - 물결모양은 PWM핀(아날로그 출력핀)



스케치 7-3 : RGB LED 밝기 제어

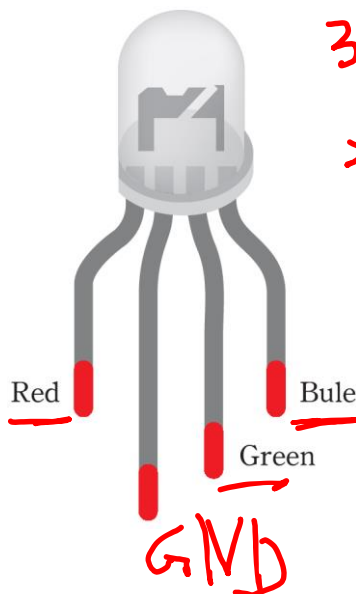
■ RGB LED

- 공통 핀과 3개의 R, G, B 제어핀으로 구성
- 공통 양극 방식의 경우 제어핀에 GND를 출력하면 켜짐

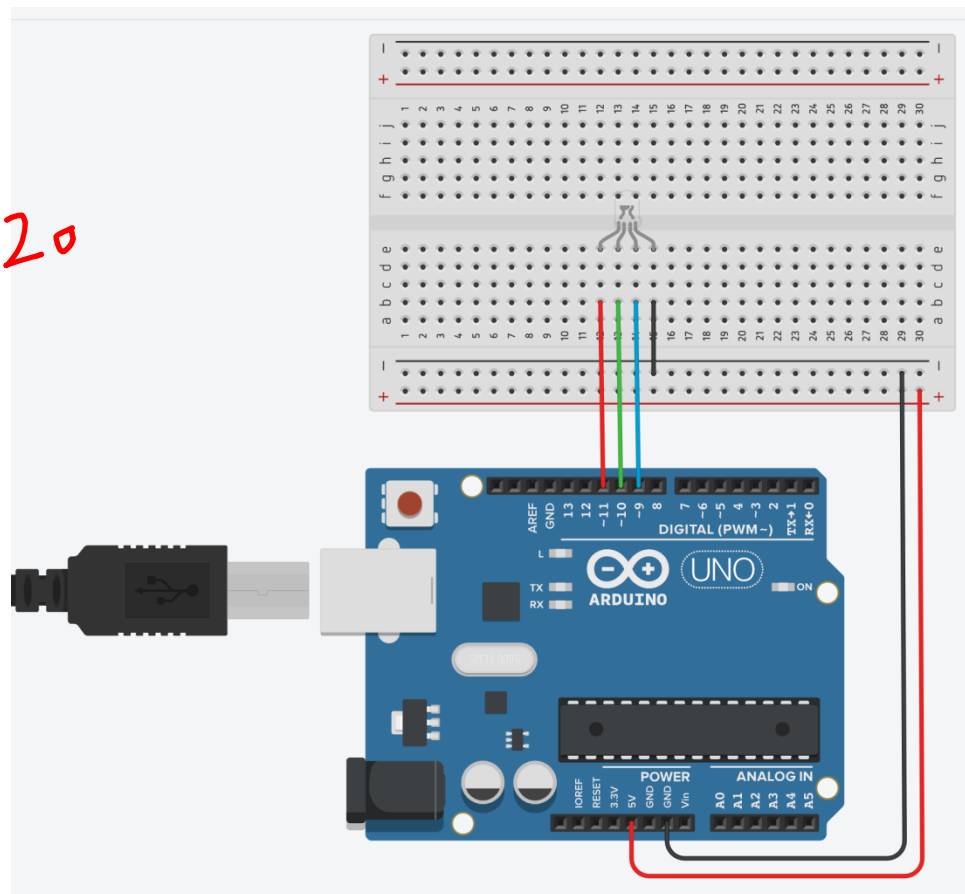
10비트
출력 0 ~ 1023
4x
0 ~ 255

■ 밝기 제어 : 공통 양극 방식

- 최대 밝기 : `analogWrite(255)`
- 최소 밝기 : `analogWrite(0)` 120



3가지
개장 밝은 색



스케치 7-4 : 조도센서로 밝기 제어

```
void loop() {
```

```
  ✓ int ADC_value = analogRead(A0);
```

```
  int PWM_value = ADC_value >> 2;
```

```
  Serial.print(String("ADC value : ") + ADC_value);
```

```
  Serial.println(String(", PWM value : ") + PWM_value);
```

```
  for(int i = 0; i < 4; i++){
```

```
    analogWrite(pins_LED[i], PWM_value);
```

```
  }
```

```
  delay(1000);
```

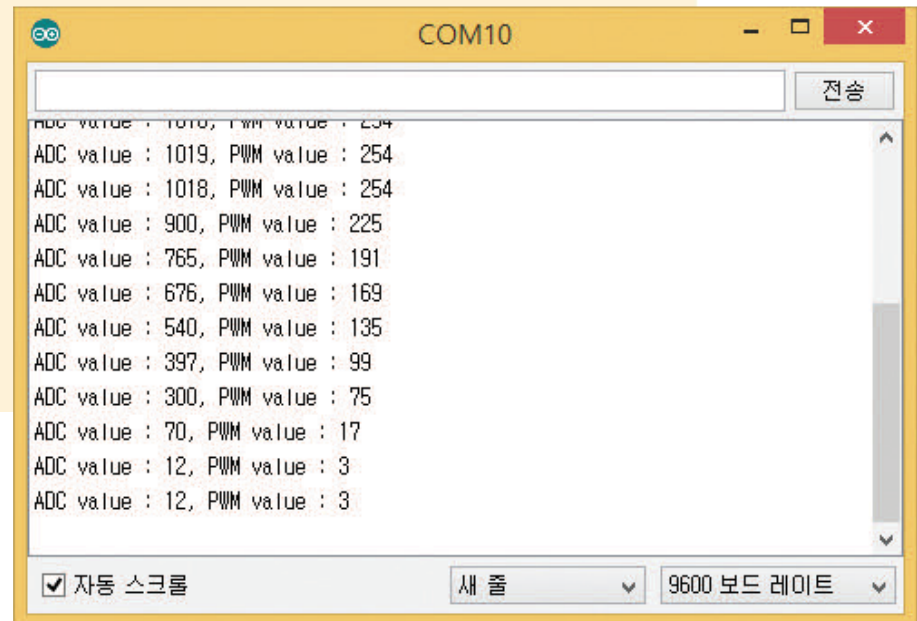
```
}
```

Handwritten notes illustrating the bit shift operation:

$0000\ 0100 \div 4 = 0000\ 0001$

$4 \div 4 \rightarrow 1$

$\rightarrow 2\text{bit} = \div 4$



스피커 출력

■ PWM 신호를 사용하여 단음 재생

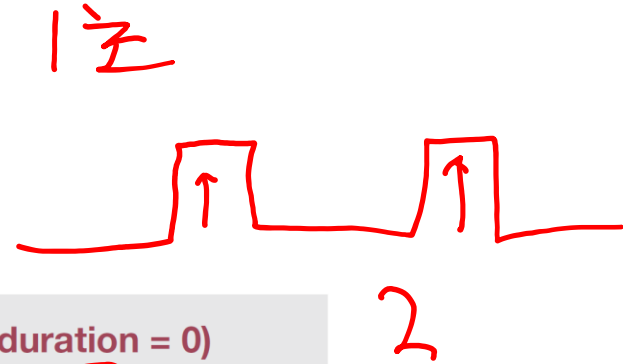
- tone 함수에서 음의 주파수를 지정하여 사용

`void tone(uint8_t pin, unsigned int frequency, unsigned long duration = 0)`

- 매개변수
 - pin : 핀 번호
 - frequency : 출력 주파수 (unsigned int)
 - duration : 출력 지속 시간 (밀리초 단위) (unsigned long)
- 반환값 : 없음

`void noTone(uint8_t pin)`

- 매개변수
 - pin : 핀 번호
- 반환값 : 없음



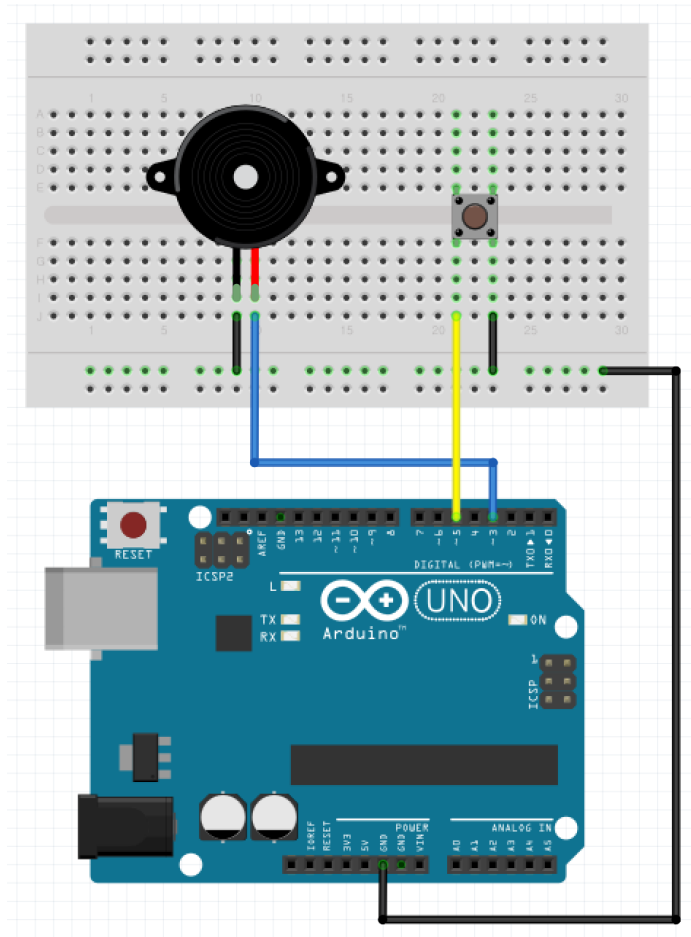
스피커 출력

■ 부저 Frequency

(단위 : Hz) ✓

| 음계 \ 옥타브 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|----------|---------|----------|----------|----------|----------|----------|----------|----------|
| C(도) | 32.7032 | 65.4064 | 130.8128 | 261.6256 | 523.2511 | 1046.502 | 2093.005 | 4186.009 |
| C# | 34.6478 | 69.2957 | 138.5913 | 277.1826 | 554.3653 | 1108.731 | 2217.461 | 4434.922 |
| D(레) | 36.7081 | 73.4162 | 146.8324 | 293.6648 | 587.3295 | 1174.659 | 2349.318 | 4698.636 |
| D# | 38.8909 | 77.7817 | 155.5635 | 311.1270 | 622.2540 | 1244.508 | 2489.016 | 4978.032 |
| E(미) | 41.2034 | 82.4069 | 164.8138 | 329.6276 | 659.2551 | 1318.510 | 2637.020 | 5274.041 |
| F(파) | 43.6535 | 87.3071 | 174.6141 | 349.2282 | 698.4565 | 1396.913 | 2793.826 | 5587.652 |
| F# | 46.2493 | 92.4986 | 184.9972 | 369.9944 | 739.9888 | 1479.978 | 2959.955 | 5919.911 |
| G(솔) | 48.9994 | 97.9989 | 195.9977 | 391.9954 | 783.9909 | 1567.982 | 3135.963 | 6271.927 |
| G# | 51.9130 | 103.8262 | 207.6523 | 415.3047 | 830.6094 | 1661.219 | 3322.438 | 6644.875 |
| A(라) | 55.0000 | 110.0000 | 220.0000 | 440.0000 | 880.0000 | 1760.000 | 3520.000 | 7040.000 |
| A# | 58.2705 | 116.5409 | 233.0819 | 466.1638 | 932.3275 | 1864.655 | 3729.310 | 7458.620 |
| B(시) | 61.7354 | 123.4708 | 246.9417 | 493.8833 | 987.7666 | 1975.533 | 3951.066 | 7902.133 |

부저 연결



부저 실습 코드



The image shows a screenshot of the Arduino IDE interface. The window title is "sketch_may07a | 아두이노 1.8.7". The menu bar includes "파일", "편집", "스케치", "툴", and "도움말". The toolbar contains icons for checking, running, uploading, and downloading. The sketch name "sketch_may07a" is shown in the top bar. The code is as follows:

```
int piezo = 3;
int sw = 5;

int numTones = 8;
int tones[] = {261, 294, 330, 349, 392, 440, 494, 523};

void setup() {
  pinMode(piezo, OUTPUT);
  pinMode(sw, INPUT_PULLUP);
}

void loop() {
  if (digitalRead(sw) == LOW) {
    for (int i = 0; i < numTones; i++) {
      tone(piezo, tones[i]);
      delay(500);
    }
    noTone(piezo);
  }
}
```

모터의 종류

■ DC 모터

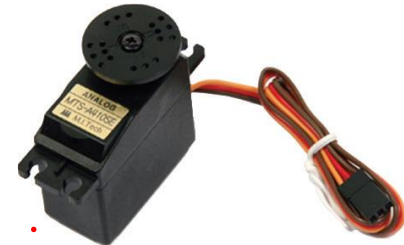
- 연속 회전 모터
- 정지 시 관성으로 정확한 위치 제어 어려움



■ 서보 모터 ✓

- DC 모터 + 귀환 제어 회로
- 정밀 제어 가능

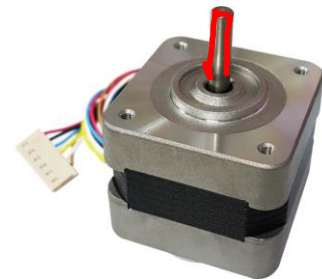
PWM



■ 스텝 모터 ✓

- 분할각 단위로만 회전 가능
- 하나의 펄스가 주어지면 분할각 만큼 회전하고 멈춤

정밀한 위치 제어를 위한 펄스 제어 방식



서보 모터

■ 표준 서보 모터 : 0~180도 사이만 회전

- 연속 회전 서보 모터는 무한 회전 가능

■ PWM 신호로 위치 제어

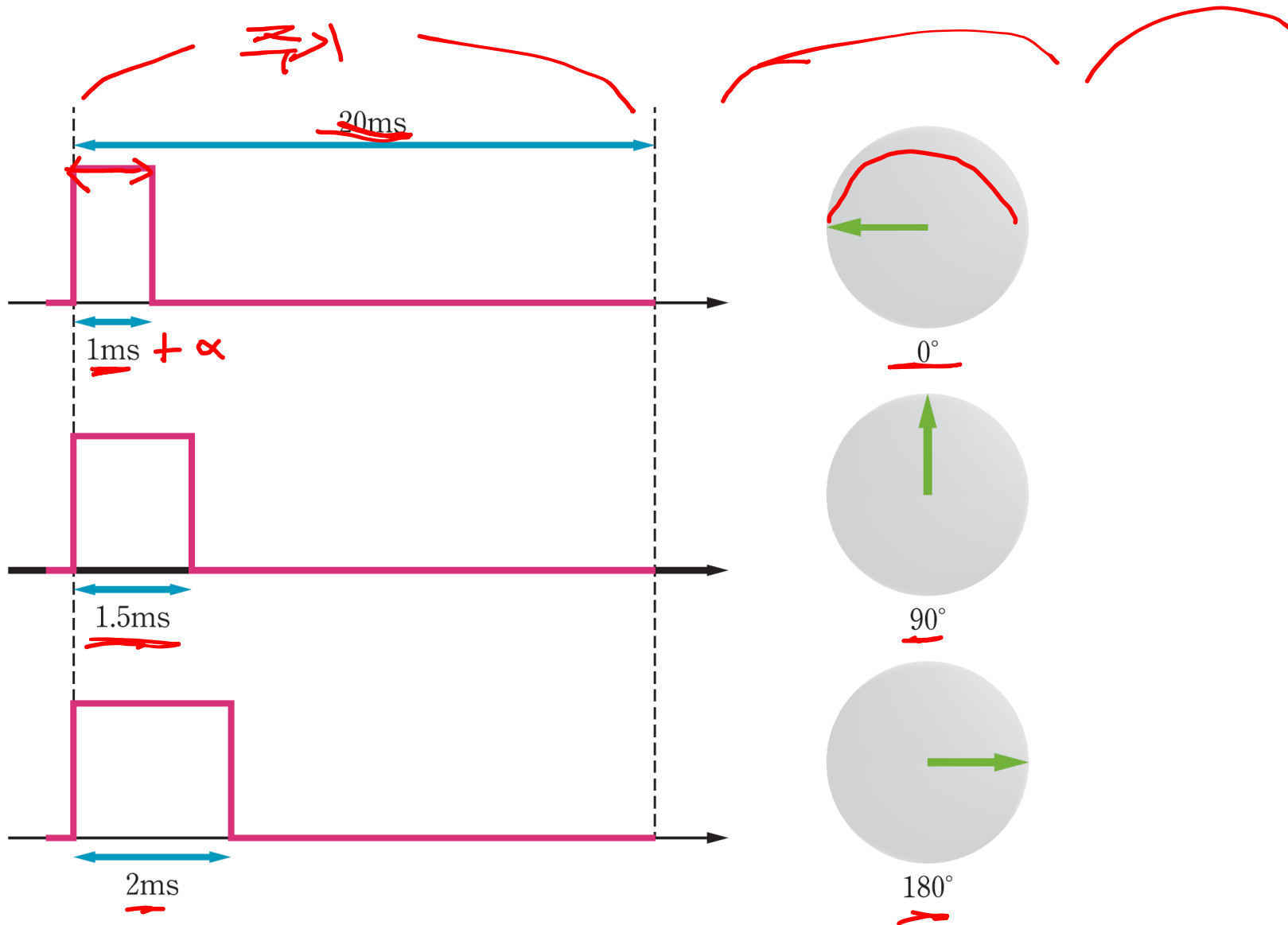
- 50Hz PWM 신호 사용 : 20ms 주기
- 1ms ($1/20 = 5\%$ 듀티 사이클) 펄스에서 0도 회전
- 2ms ($2/20 = 10\%$ 듀티 사이클) 펄스에서 180도 회전

■ 3개의 연결선

- VCC : 붉은색 ✓
- GND : 검정색 또는 갈색 ✓
- (위치 설정을 위한) 제어선 : 노란색, 주황색 또는 흰색 ✓

• PWM 출력이 가능한 핀에 연결하여야 함

서보 모터 제어



서보 모터 제어 라이브러리

■ Servo 라이브러리

- 표준 라이브러리 중 하나
- 우노의 경우 최대 12개, 메가2560의 경우 최대 48개 서보 모터 제어 가능

■ 사용 방법

- 라이브러리 포함

```
• #include <Servo.h>
```

- 객체 생성

```
• Servo myServo;
```

- 제어핀 연결

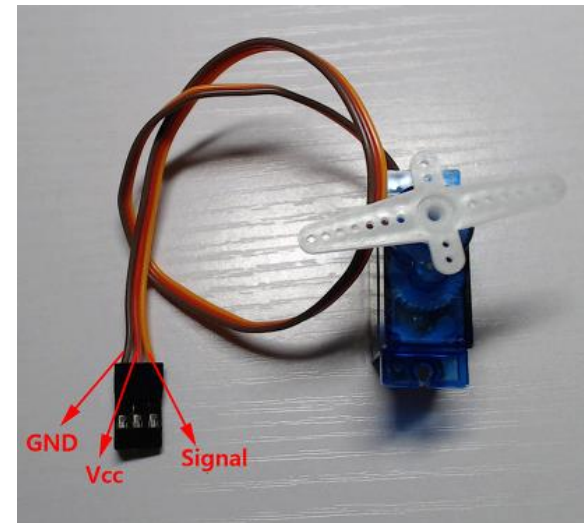
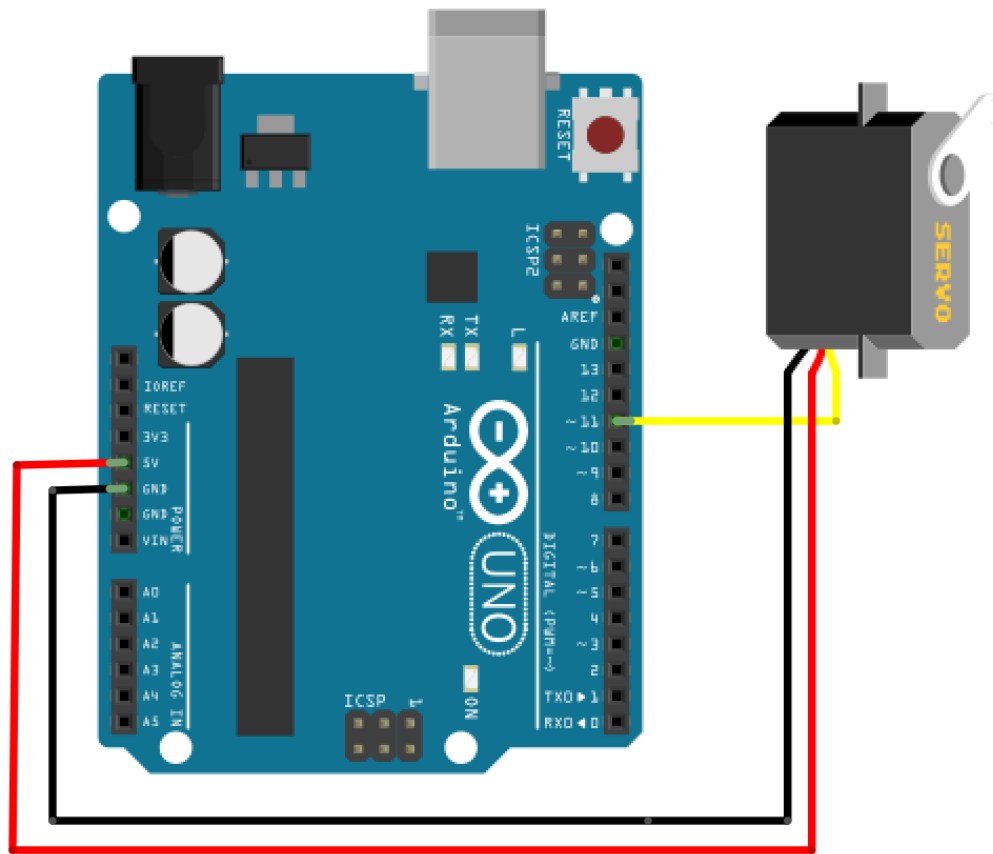
```
• myServo.attach(servoPin);
```

- 각도 지정

```
• myServo.write(angle);
```

0 ~ 180

서보 모터 연결



```
#include <Servo.h>

Servo servo;
int servoPin = 11;
int angle = 0; // servo position in degrees
void setup()
{
    servo.attach(servoPin);
}

void loop()
{
    // scan from 0 to 180 degrees
    for(angle = 0; angle < 180; angle++)
    {
        servo.write(angle);
        delay(15);
    }
    // now scan back from 180 to 0 degrees
    for(angle = 180; angle > 0; angle--)
    {
        servo.write(angle);
        delay(15);
    }
}
```


■ 아날로그 데이터 출력

- ATmega2560에는 DAC가 없으므로 아날로그 데이터 출력은 불가능
- 디지털 신호의 일종인 펄스 폭 변조 신호를 통해 아날로그 데이터와 유사한 효과를 얻을 수 있음
- LED 밝기 제어, 모터 속도 제어 등에 PWM 신호가 사용

Thank you!!

