

Chapter 14. 서보 모터

모터는 전기를 이용하여 회전력을 얻는 기기로 움직이는 장치를 만들기 위해 필수적인 부품 중 하나이다. 마이크로컨트롤러와 함께 사용되는 모터에는 여러 종류가 있지만 DC 모터, 서보 모터, 스텝핑 모터 등이 흔히 사용된다. 각각의 모터들은 그 특징과 용도가 조금씩 다르므로 사용하고자 하는 목적에 맞는 모터를 선택하는 것이 중요하다.

- DC (Direct Current) 모터 : 축이 연속적으로 회전하는 형태의 모터로, 전원이 연결되면 회전하고 전원이 끊어지는 경우에 정지하는 간단한 구조로 되어 있어 제어가 간단한 장점이 있다. 정지 시에는 관성으로 인해 정확한 정지 위치를 지정하기 어렵다.
- 서보 (servo) 모터 : DC 모터의 한 종류로 DC 모터에 귀환 제어 회로를 추가하여 위치 제어가 가능하도록 구성된 모터이다. 제어 회로로 가격이 비싼 단점은 있지만 정밀 제어가 가능하며, 속도 면에서는 스텝핑 모터에 비해 빠른 장점이 있다.
- 스텝 (step) 모터 : 전원이 공급되면 축은 일정 각도를 회전하고 멈춘다. 축을 연속적으로 회전시키기 위해서는 모터로 펄스열이 전달되어야 하며 하나의 펄스에 반응하여 회전하는 양을 분할각(step angle)이라고 한다. 스텝 모터는 원하는 각도만큼 회전시킬 수 있는 장점은 있지만 분할각 단위로만 제어되어 미세한 제어는 불가능하다.

1. 서보 모터

서보 모터는 DC 모터의 한 종류로 PWM (Pulse Width Modulation) 신호를 통해 모터를 특정 위치로 움직일 수 있는 모터이다. 서보 모터는 일반적으로 $0^{\circ} \sim 180^{\circ}$ 범위에서만 회전할 수 있다.

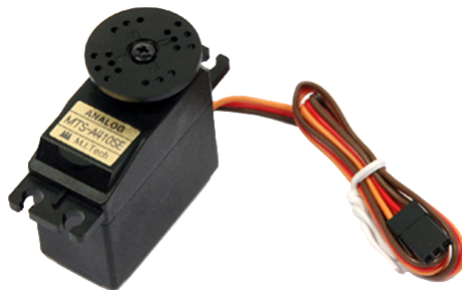


그림 1. 표준 서보 모터¹⁾

서보 모터는 일반적으로 3개의 연결선을 (VCC, GND 그리고 제어선) 가지며 VCC는 붉은색, GND는 검정색이나 갈색, 제어선은 노란색, 주황색 또는 흰색으로 구별된다. DC 모터의 경우 VCC와 GND의 구별이 없으며 반대로 연결하는 경우 반대 방향으로 회전하지만, 서보 모터의 경우에는 전원을 반대로 연결하면 모터가 파손될 수 있으므로 주의하여야 한다.

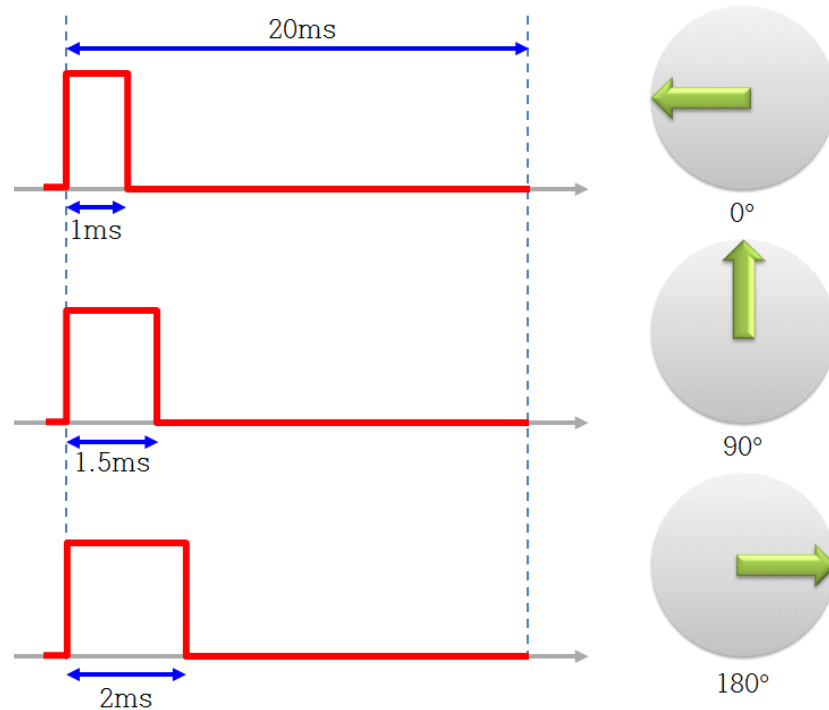


그림 2. 서보 모터 제어

서보 모터의 위치 제어는 PWM 신호에 의해 이루어진다. 서보 모터가 PWM 신호를 받으면 현재 위치와 입력된 신호를 비교하여 입력된 PWM 신호에 맞는 위치로 모터를 회전시킨다. 서보 모터는 50Hz의 PWM 주파수, 20ms 주기의 PWM 신호를 필요로 한다. 20ms 중 서보 모터의 위치를 결정하는 구간은 1~2ms로, 1ms에서는 반시계 방향으로 최대 회전 상태(0°)이며 2ms에서는 시계 방향으로 최대 회전 상태(180°)에 있게 된다.

2. Servo 라이브러리

서보 모터 제어를 위해 아두이노에는 기본 라이브러리 중 하나로 Servo 라이브러리가 포함되

1) 이미지 출처: <http://www.newtc.co.kr>

어 있다. Servo 라이브러리를 사용하면 아두이노 우노에서는 최대 12개까지, 아두이노 메가 2560에서는 최대 48개까지 서보 모터를 제어할 수 있다. Servo 라이브러리를 사용하기 위해서는 먼저 각 서보 모터 제어를 위한 객체를 생성하고 attach 함수를 통해 서보 모터 제어 핀과 서보 모터를 연결시켜 주어야 한다.

```
uint8_t attach(int pin)
uint8_t attach(int pin, int min, int max)
- 매개변수
  pin : 서보의 데이터 선이 연결된 핀 번호
  min : 0°에 해당하는 마이크로초 단위의 펄스 폭 (디폴트값은 544)
  max : 180°에 해당하는 마이크로초 단위의 펄스 폭 (디폴트값은 2400)
- 반환값 : 현재 서보 모터가 연결된 채널
```

attach 함수는 서보 모터가 연결된 채널을 반환하며, 연결에 실패하면 255를 반환한다. 서보 모터가 연결되면 write 함수를 통해 각도 제어가 가능하다. write 함수의 매개변수는 0에서 180 사이의 값을 지정하여 위치를 제어할 수 있다.

```
void write(int value)
- 매개변수
  value : 서보 모터의 제어 값
- 반환값 : 없음
```

서보 모터의 제어선을 디지털 11번 핀에 연결하자. 학습보드에도 서보 보터는 포함되어 있지 않지만 서보 모터 연결을 위한 커넥터가 준비되어 있으므로 간단하게 연결할 수 있다.



그림 3. 학습보드의 서보 모터 커넥터

스케치 1은 서보 모터를 0°에서 180°로 회전시키고, 다시 0°로 회전시키는 스케치의 예이다.

스케치 1. 서보 모터 제어

```
#include <Servo.h>

Servo myServo;
int servoPin = 11;

void setup(){
    myServo.attach(servoPin);    // 서보 모터 연결
}

void loop(){
    int angle;                  // 축 제어 값

    for(angle = 0; angle < 180; angle++){    // 0도에서 180도로
        myServo.write(angle);
        delay(5);
    }
}
```

```
    }  
  
    for(angle = 180; angle > 0; angle--){    // 180도에서 0도로  
        myServo.write(angle);  
        delay(5);  
    }  
}
```

3. 가변저항으로 서보 모터 제어

아날로그 0번 A0 핀에 연결된 가변저항을 사용하여 서보 모터의 위치를 제어하는 스케치를 작성해 보자. 가변저항은 0에서 1023 사이의 값을 반환하므로 이를 0°에서 180° 사이의 값으로 변환하고, 변환된 각도 값으로 서보 모터를 제어하면 된다. 스케치 2는 가변저항으로 서보 모터를 제어하는 스케치의 예이다.

스케치 2. 가변저항으로 서보 모터 제어

```
#include <Servo.h>  
  
Servo myServo;  
int servoPin = 11;  
  
void setup(){  
    myServo.attach(servoPin);    // 서보 모터 연결  
}  
  
void loop(){  
    int value = analogRead(A0);    // 가변저항 값 읽기  
    // 가변저항 값을 각도 값으로 변환  
    int angle = map(value, 0, 1023, 0, 180);  
  
    myServo.write(angle);  
}
```