

**[아두이노 강좌] 40. 서보 모터 (1) - 서보 모터 동작 방식** 강좌

2015. 4. 30. 9:42

<https://blog.naver.com/yuyyulee/220345769854>

모터 시리즈 중 마지막인 서보 모터. 뭔가 힘든 여정이었음.



서보 모터는 정확한 각도 회전을 위해 사용된다는 점은 스텝 모터와 비슷하지만 구동되는 방식은 전혀 다르다. 우선 서보 모터가 무엇이나.

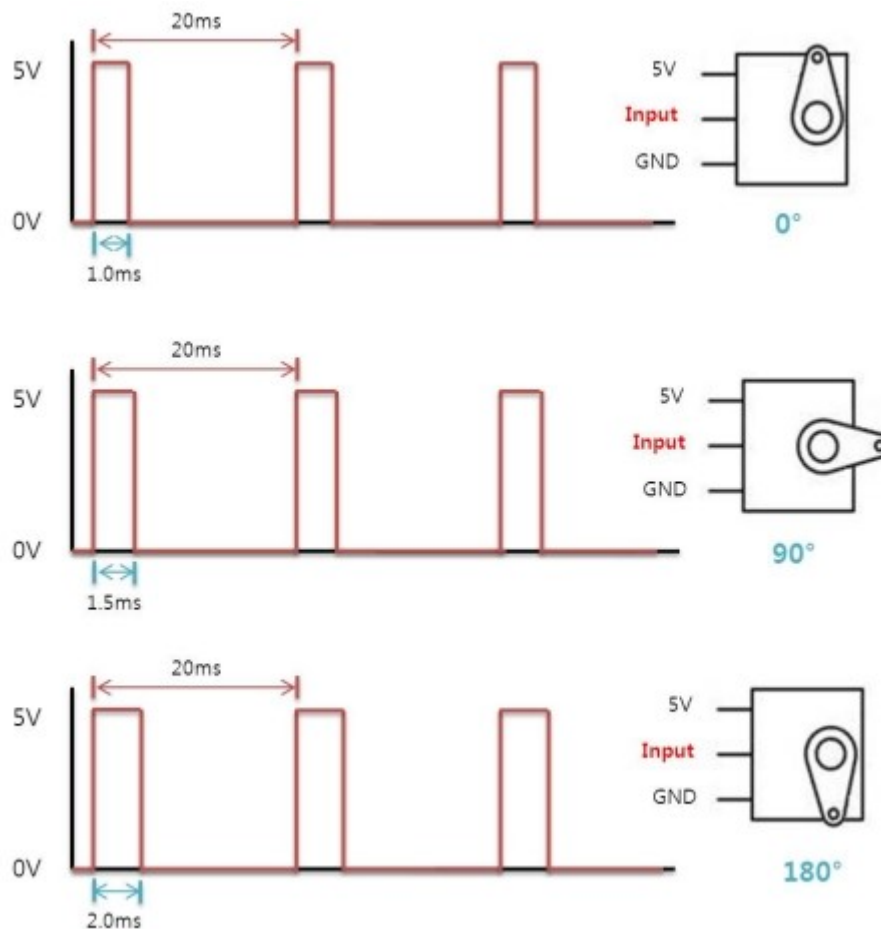
서보 모터는 일반적으로 모터 하나만을 가지고 서보라고 하지는 않는다. 서보(Servo)라는 말 자체가 ‘하인’이라는 말에서 유래되어 따르다, 추종하다 등의 의미를 담고 있다고 하여 제어 명령에 따라 정확한 위치와 속도를 맞출 수 있는 모터를 서보 모터라고 한다. (구글 검색 했어용. 데헛.)

일반적인 의미는 그렇고, 서보 모터의 종류도 여러 가지지만 다 설명하기는 어렵기 때문에(나도 모르고), 우리가 아두이노로 제어할 서보 모터만을 설명하자. 쉽게 말해서 원하는 각도로 모터를 위치시키는 용도로 사용한다고 생각하면 됨. 모터를 ‘위치시킨다’라는 말이 스텝 모터와의 차이를 나타내는데, 스텝 모터의 경우 원하는 각도만큼 “회전”하기 위해서, 서보는 “위치”시키기 위해서.

잉? 서보는 회전이 안된다규. 보통의 경우 최대 180°까지만 돌 수 있다.

제어 방법은 DC 모터와 같은 PWM이지만, 서보 모터의 PWM은 주파수가 정해져 있으며, 듀티비라기보다는 신호의 유지 시간으로 회전 각도가 결정된다.

그림으로 설명하자. 대부분의 서보 모터는 3개의 선으로 이루어져 있다. 전원, GND, 입력 신호. 입력 신호에 다음과 같은 신호가 입력되면 신호에 맞는 각도로 모터가 회전한다.

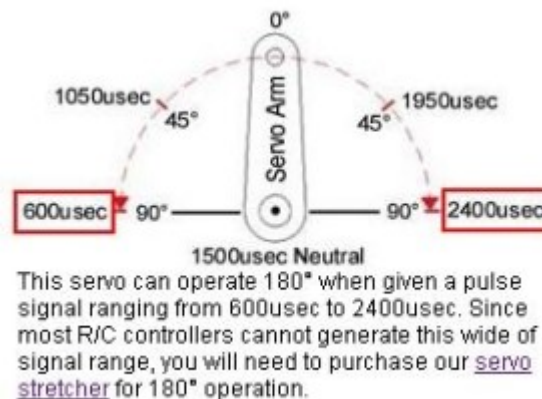


서보 모터를 제어하기 위해서는 50Hz의 주파수를 가지는 신호가 입력되어야 한다. 즉, 한 주기 당 20ms의 시간을 가진다는 것. 20ms의 주기 안에서 HIGH 신호의 폭(=시간)이 얼마인가에 따라 서보 모터의 회전 각도가 결정된다.

서보 모터의 종류에 따라 약간씩 다를 수는 있겠지만 보통은 1.0ms에서 0°, 1.5ms에서 90°, 2.0ms에서 180°의 각도를 가진다. 이 시간은 서보 모터의 데이터 시트에 정확히 명시되어 있을 것이다. 모든 것은 데이터 시트에.

사실 1.5ms에서 90°, 정중앙에 위치하는 것이고 1.5ms보다 작을 때 시계 반대 방향으로 이동, 1.5ms보다 클 때 시계 방향으로 이동하는 거라고 생각하면 된다. 모터가 각 방향으로 최대한 이동했을 때의 신호 시간 폭은 모터마다 다를 수 있다. 물론 최대한 이동했을 때의 각도도 0°나 180°보다 크거나 작을 수 있다.

내가 가진 서보 모터는 약 0.6 밀리초에서 0°(-90°), 2.4 밀리초에서 180°(+90°)로 이동한다,고 되어 있네. 다음은 데이터 시트에 명시된 부분.



서보 모터는 위치를 유지하기 위해서는 신호를 계속해서 보내야 한다. 신호를 주지 않으면 모터는 각도를 유지하지 못하고 흐물 흐물 돌아가버린다. 사실 1.5ms를 준다고 해서 정확히 90°로 회전하는 것은 아니고 약간의 편차가 있을 수 있기 때문에 기준 펄스를 계산하기 위해 약간의 튜닝 작업이 필요하다는 단점이 있다.

서보 모터의 경우 DC 모터나 스텝 모터처럼 큰 전압을 요구하지는 않지만, 큰 전류를 요구하기 때문에 역시 외부 전압을 아두이노 보드에 연결해주는 것이 좋다. 게다가 서보는 회전하지 않아도 각도를 유지하기 위해 전류를 계속 소모하게 되므로, 만일 건전지를 이용한다면 건전지의 소모 시간이 예상 소모 시간보다 훨씬 짧을 수 있다는 마음의 준비를 하는 것이 좋다.

그럼 신호를 소스로 구현해볼까. 저거 은근히 까다롭...

```
ServoTest01.ino

int servoPin = 6;
int freq = 20000; // 20 milliseconds (50Hz)
int minPulse = 600; // 600 microseconds
int maxPulse = 2400; // 2400 microseconds

void setup() {
  pinMode(servoPin, OUTPUT);
  digitalWrite(servoPin, LOW);
}

void loop() {
  for(int i=0; i<=180; i++) {
    setServo(i);
  }

  for(int i=180; i>=0; i--) {
    setServo(i);
  }
}

void setServo(int degree) {
  int hTime = 0;
  int lTime = 0;

  if(degree < 0) degree = 0;
  if(degree > 180) degree = 180;

  hTime = (int)(minPulse + ((maxPulse-minPulse) / 180.0 * degree));
  lTime = freq - hTime;

  digitalWrite(servoPin, HIGH);
  delayMicroseconds(hTime);
  digitalWrite(servoPin, LOW);
  delayMicroseconds(lTime);
}
```

우선 서보 모터에 입력될 신호 핀을 9번 핀으로 저장한다. 디지털 출력 기능만을 사용할 것이므로 원하는 핀 아무거나 사용해도 된다.

그리고 서보 모터의 스펙에 맞는 주기와 최소 펄스, 최대 펄스 값을 저장한다. 내가 가진 모터 스펙에 맞게 설정해주면 된다. 각도 계산에 사용된다.

setup() 함수에서 신호 핀을 출력 모드로 설정하고 LOW 값으로 초기화 해준다.

loop() 함수에서는 for() 반복문을 이용하여 서보 모터의 각도를 제어하는 함수를 호출한다.

### void setServo(degree)

서보 모터를 지정한 각도로 위치시키는 함수다. 단, 반복해서 신호를 출력하지는 않고 1번만 출력하는 기능을 하므로 만일 계속해서 신호를 발생시키고 싶다면 이 함수를 계속 호출해야 한다. 반복 출력되도록 하려면 타이머나 뭐 그런걸 쓰면 되긴 한데, 그냥 서보 모터 동작을 이해하기 위한 예제이므로 간단하게 구현했다.

#### 매개 변수

degree : 각도 값을 지정한다. 각도는 0~180 사이의 값만 허용된다.

내가 만든 함수면서 마치 아두이노 함수마냥 설명하고 있억ㅋㅋㅋㅋㅋㅋ



함수 내용을 보면 우선 HIGH 신호의 폭(=시간)을 조절하기 위해 딜레이 시간을 저장하기 위한 변수를 선언한다.

만일 매개 변수로 받은 값이 0보다 작거나 180보다 클 경우 degree 값을 0 또는 180으로 변경한다.

hTime 변수는 신호가 HIGH로 유지되기 위한 딜레이 타임으로, maxPulse(180°일 때의 신호 폭)에서 minPulse(0°일 때의 신호 폭)를 뺀 값을 180으로 나누어 1° 회전을 위한 시간 간격을 구하고, 이 값에 degree 값을 곱해줘서 원하는 각도로 위치시키기 위한 시간을 구한다.

lTime 변수는 HIGH 출력 후 LOW가 출력될 시간으로, 한 주기의 시간인 freq(20ms)에서 hTime을 뺀 만큼의 시간을 저장한다.

그리고는 digitalWrite() 함수를 이용하여 hTime만큼 HIGH 신호를, lTime만큼 LOW 신호를 출력하면 된다.

소스를 업로드 한 후 실행하면 모터가 -90°에서 90°까지 왔다 갔다 하는 것을 볼 수 있다.

짹짹!

다음 강좌에서는 서보 모터 제어를 위해 아두이노에서 제공하는 라이브러리 'Servo'에 대해 알아보도록 하자. 오늘은 이만. 안녕!