**3주차. PWM 실험**

# **예제1. PWM을 이용한 LED 밝기 조절**

|  |  |  |
| --- | --- | --- |
| **구분** | **설명** | **코드** |
| 핀 정의  핀 세팅 | 핀 번호를 LED\_BULITIN으로 정의합니다.  빌트인 LED를 출력으로 사용합니다. | Int LED\_BULITIN = 12;  Void setup(){  pinMode(LED\_BUILTIN, OUTPUT)  } |
| 루프 | 프로그램의 동작을 루프문 안에 기술합니다. | void loop(){ |
|  |  | digitalWrite(LED\_BUILTIN, HIGH);  delay(1000); |
|  |  | digitalWrite(LED\_BUILTIN, HIGH);  delay(1000); |
| analogWrite | LED\_BULITIN을 analog형식으로 값을 출력합니다. | }  For(int i=10;i<=255;i+=10)  {  analogWrite(LED\_BULITIN,i)  } |

**예제2. PWM을 이용한 모터 각도 조절**

|  |  |  |
| --- | --- | --- |
| **구분** | **설명** | **코드** |
| 라이브러리 호출  핀 정의  핀 세팅 | Servo라이브라리를 호출하여 servo의 함수를 사용합니다.  핀 번호를 servoPin 으로 정의합니다.  Servor객체를servoPin으로 초기화 합니다. | #include<Servo.h>  Servo servo  Int servoPin = 5;  Void setup(){  Servo.attach(servoPin)  } |
| 루프 | 프로그램의 동작을 루프문 안에 기술합니다. | void loop(){  for(Level = 0; Level<180;Level++){ |
|  |  | Servo.write(Level);  Delay(10); |
|  |  | }  For(Level=180;Level>0;Level—)  {  Servo.write(Level);  Delay(10);  }  } |
| Servo.write | Servo객체의 write함수를 이용하여 위치를 지정한다. | Servo.write(level) |

**1.서보모터의 날개가 0,45,90,135,180도 방향지시를 반복하는 프로그램 만들기**

**소스 코드**

=========================================================================================

#include<Servo.h>

int servoPin = 5;

Servo servo;

int Level =0;

int ledPin = 11;

void setup() {

servo.attach(servoPin);

}

void loop() {

for(Level =0;Level <45;Level++)

{

servo.write(Level);

delay(10);

}

for(Level =45;Level <90;Level++)

{

servo.write(Level);

delay(10);

}

delay(1000);

for(Level = 90;Level <135;Level++)

{

servo.write(Level);

delay(10);

}

delay(1000);

for(Level =135;Level <180;Level++)

{

servo.write(Level);

delay(10);

}

delay(1000);

for(Level = 180;Level >0;Level--)

{

servo.write(Level);

delay(10);

}

delay(1000);

}

=========================================================================================

**2.LED를 부착하여 각도를 표현하는 프로그램을 작성**

**소스 코드**

=========================================================================================

#include<Servo.h>

int servoPin = 5;

Servo servo;

int Level =0;

int ledPin = 6;

void setup() {

servo.attach(servoPin);

pinMode(ledPin,OUTPUT);

}

void loop() {

for(Level =0;Level <45;Level++)

{

servo.write(Level);

analogWrite(ledPin,Level);

delay(10);

}

delay(1000);

for(Level =45;Level <90;Level++)

{

servo.write(Level);

analogWrite(ledPin,Level);

delay(10);

}

delay(1000);

for(Level = 90;Level <135;Level++)

{

servo.write(Level);

analogWrite(ledPin,Level);

delay(10);

}

delay(1000);

for(Level =135;Level <180;Level++)

{

servo.write(Level);

analogWrite(ledPin,Level);

delay(10);

}

delay(1000);

for(Level = 180;Level >0;Level--)

{

servo.write(Level);

analogWrite(ledPin,Level);

delay(10);

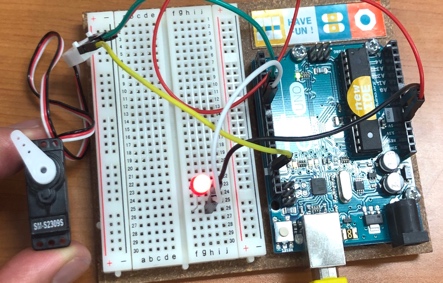
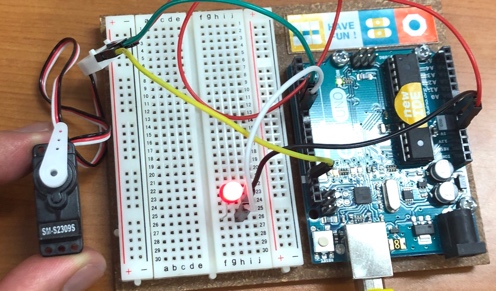
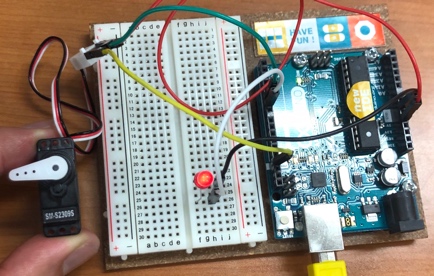
}

delay(1000);

}

}

=========================================================================================



0도 45도 90도

|  |
| --- |
| 135도 180도  **고찰 및 실험 평가**  **이번 실험은 펄스폭 변조 방식인 pwm을 이용하여 led를 아날로그 신호처럼 바꾸어 사용하는 것과, 서보 모터를 사용하여, 각도를 움직이는 것을 이용하여 실험을 하였다. 여기서 서보 모터란 정밀하게 제어 할 수 있는 기어와 샤프트를 가지고 있고 원하는 각도로 이동시키는 모터를 말한다. DC모터와는 다르게 서보모터는 0~180도 범위의 회전각을 가지고 있고, 180도 이상으로 가게 되면 손상이 된다. 또한 서보모터 또한 pwm방식을 사용하여 제어하는데 신호의 유지시간으로 회전각도를 결정한다. 이때 전체 20ms의 pmw주기중에서 1~2ms사이의 파형을 이용하여 각도를 제어한다.그러므로 1ms의 high 신호를 주게 되면 0도 1.5ms만큼의 high신호를 주면 90도 2ms만큼의 high신호를 준다면 180도를 가르킨다.**  **이때 servo라는 라이브러리 함수를 이용하면 서보모터를 동작할 수있는 다양한 라이브러리 함수를 사용할 수 있다. 여기서 주로 사용한 것은 attach함수와 write함수 이다. 첫번째 실험에서 0도 45도 90도 135도 180도는 for문을 이용하여, 0~45도, 45~90도 90~135도 135~180도의 사이에 delay를 사용하여 구분을 하였다 그런데, 프로그램을 돌려도 계속 실행이 안되어서 몇번이고 알아보았는데, 입력한 코드에서 입력이 들어가는 pin이 제대로 연결되지 않아서 실행이 되지 않았다. 그래서 선언된 pin으로 다시 실행하니 제대로 돌아가는 것을 확인 할 수 있었다. 다음 실험 부터는 코드를 짠 후 입력한 코드와 맞는 아두이노 핀번호에 연결되어있는지 확인 해야겠다. 두번째 실험은 각도가 돌아가는 것을 led로 나타내는 것이였는데 우리는 각도에 맞게 for문안에 led핀에 pwm을 이용하여 아날로그 신호를 같이 넣어서 표현 하였다. 두번째 실험에서는 led핀에서 불이 들어오지 않아 확인 해보니 아두이노 판에서 핀에 동작하지 않는 곳이 많았다. 많은 사람들이 사용하다 보니 그런지, 다음실험 부터는 실험전, 핀이 제대로 동작하는지 확인하고 난 후 실험 해야겠다.**  **추가) 라이브러리를 사용하지 않고 write 함수 사용하는 법**  **먼저 servo 라이브러리 경로에 들어가서 cpp 파일을 열어보면**  **write 함수가 다음과 같이 선언 된 것을 확인 할 수 있다.**  void Servo::write(int degrees) {  degrees = constrain(degrees, this->minAngle, this->maxAngle);  this->writeMicroseconds(ANGLE\_TO\_US(degrees));  }  **여기에서 writeMicroseconds 함수가 또 호출 되므로,**  void Servo::writeMicroseconds(uint16 pulseWidth) {  if (!this->attached()) {  ASSERT(0);  return;  }  pulseWidth = constrain(pulseWidth, this->minPW, this->maxPW);  analogWrite(this->pin, US\_TO\_COMPARE(pulseWidth));  }  **그러므로 라이브러리 함수를 사용하지 않고 사용하기 위해서는 write 함수와 writemicroseconds함수를 새로 만들어 주면 된다.**  **constrain함수와 analogwrite 함수를 사용하여 만들고,**  **int minpw, int maxpw 선언, int pin, int Minangle, int maxangle 을 선언하면 된다.** |