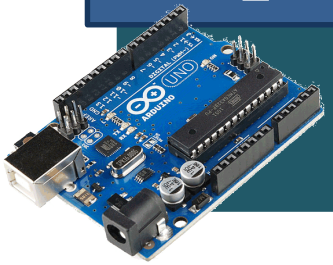
2022년 IoT기반 스마트 솔루션 개발자 양성과정



Firmware [펌웨어]

17-Servo Motor

담당 교수 : 유근택

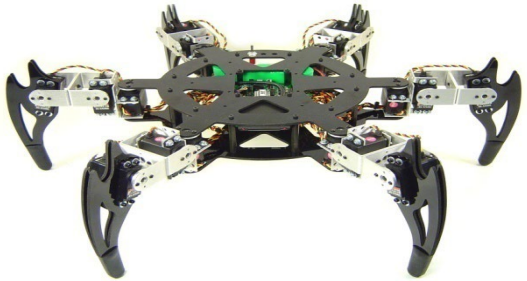
010-5486-5376

rgt3340@naver.com

Servo Motor (서보 모터)

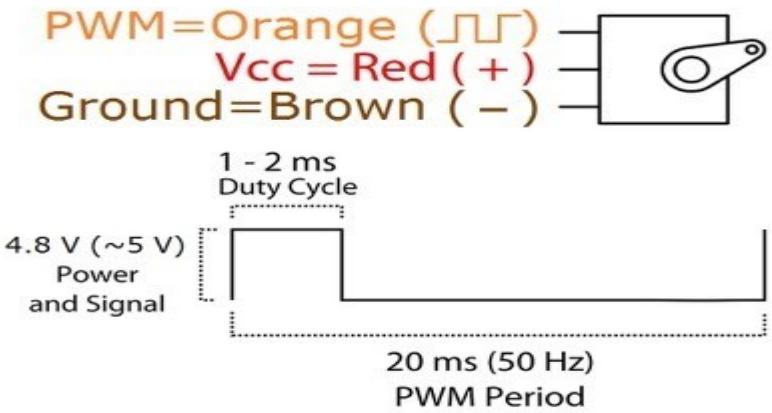
- Servo : 추종하다, 따라하다
- DC Servo Motor
  - 구성이 간단, 고 정밀도
  - 수명이 짧다
- AC Servo Motor
  - 대형 시스템 구성 가능, 수명이 안정
  - 구성이 복잡, 고가

Servo Motor의 활용



소형 DC Servo Motor

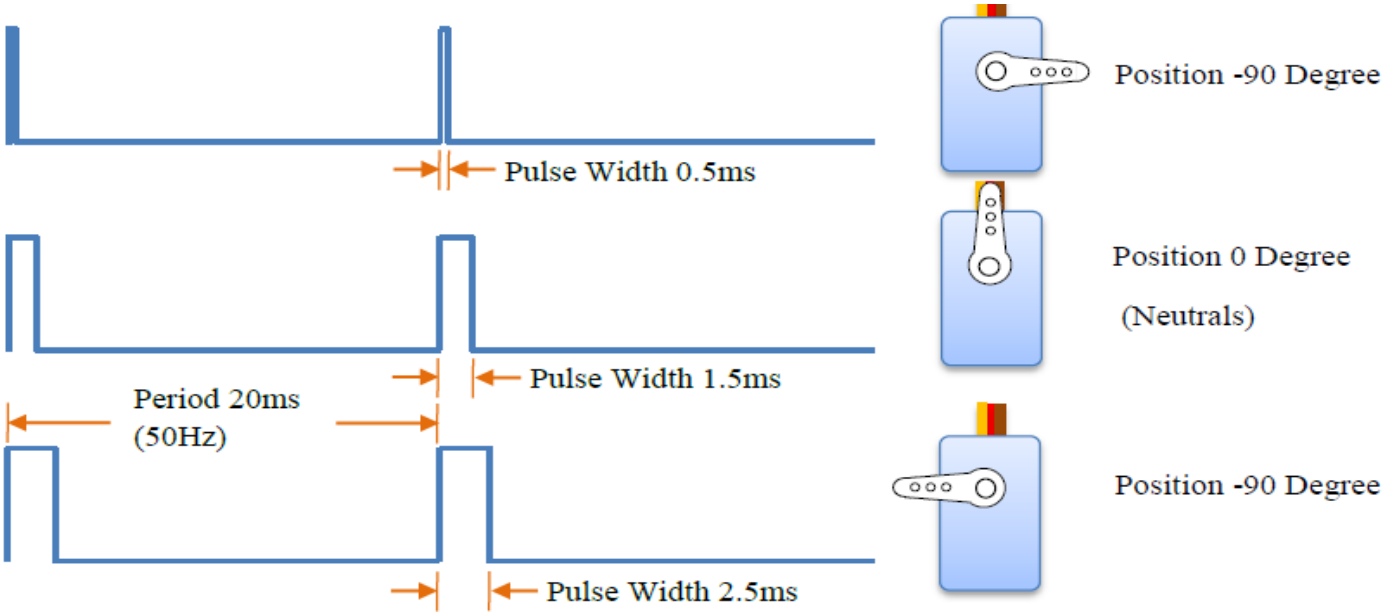
- DC Motor의 한종류
- 궤환회로로 구성되어 각도 조절이 가능함
- PWM 가능한 포트 사용



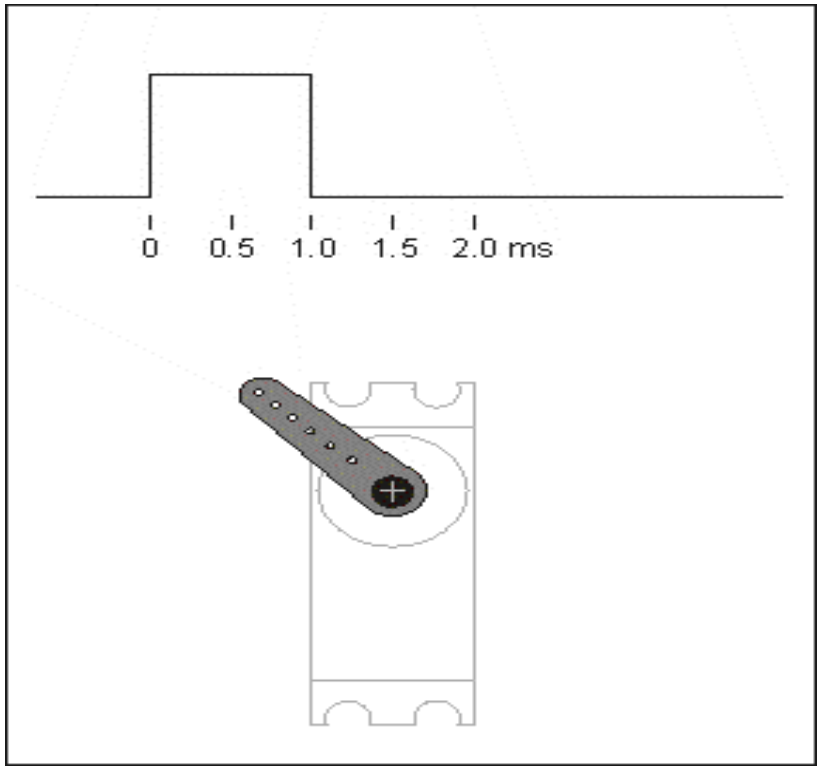
Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right, "-90" (~1 ms pulse) is all the way to the left.

Servo Motor의 동작원리

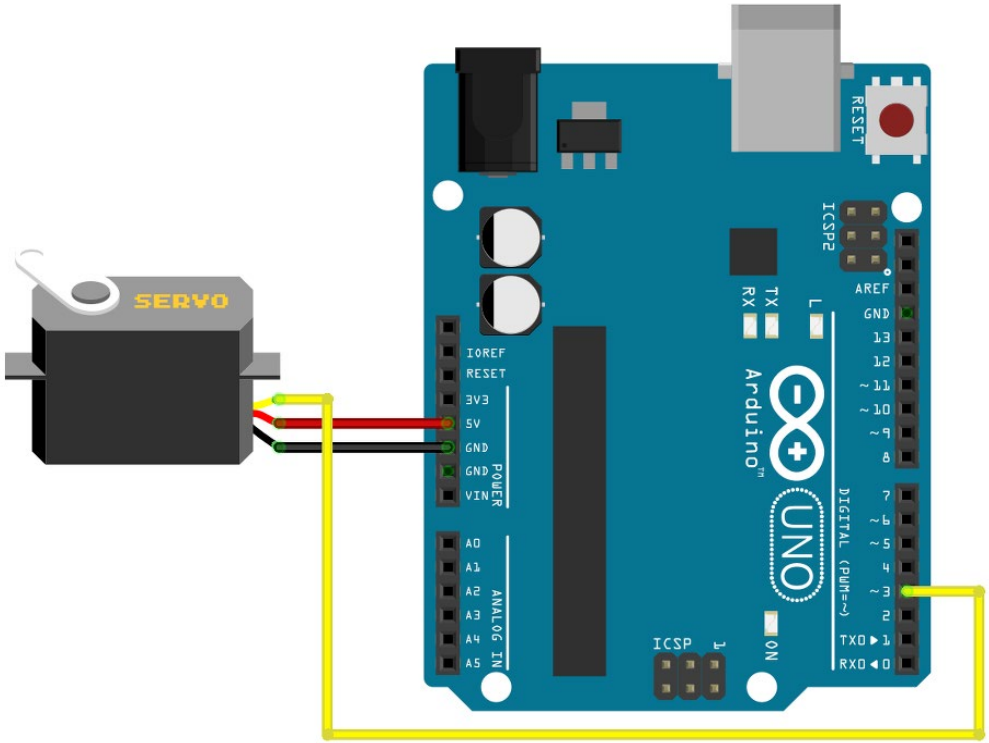
- 펄스의 폭이 **0.5ms 이면 0도, 1.5ms 이면 90도, 2.5ms 이면 180도**로 제어가 된다.



Servo Motor의 동작

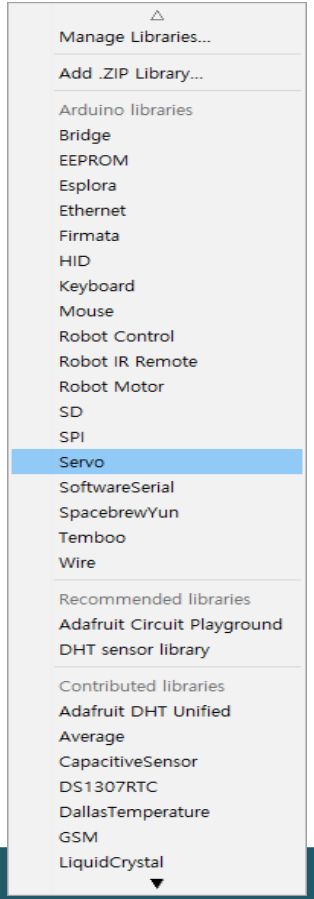


Servo Motor Test 결선1



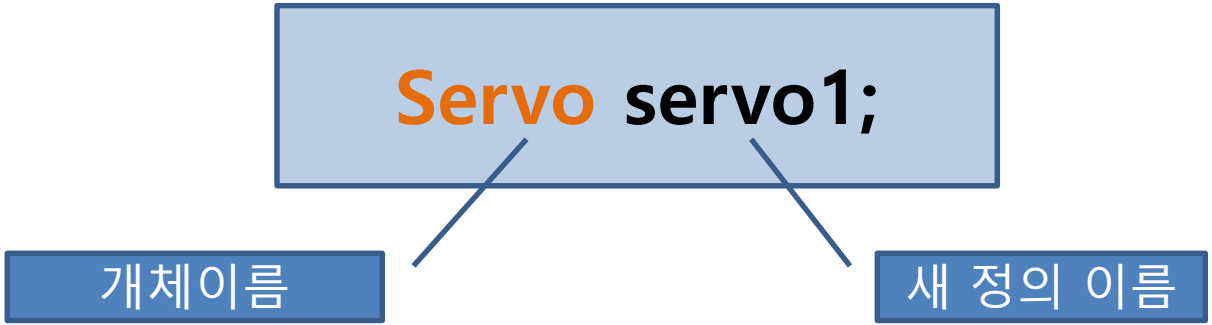
servo 라이브러리

- attach(pin)
  - pin : 서보가 연결된 핀번호
- write(angle)
  - angle : 0 ~ 180
  - 연속 서보일 경우는 속도
- 라이브러리의 사용
  - Menu-Sketch-include Library->Servo
  - #include <Servo.h>



Servo 라이브러리 개체 선언

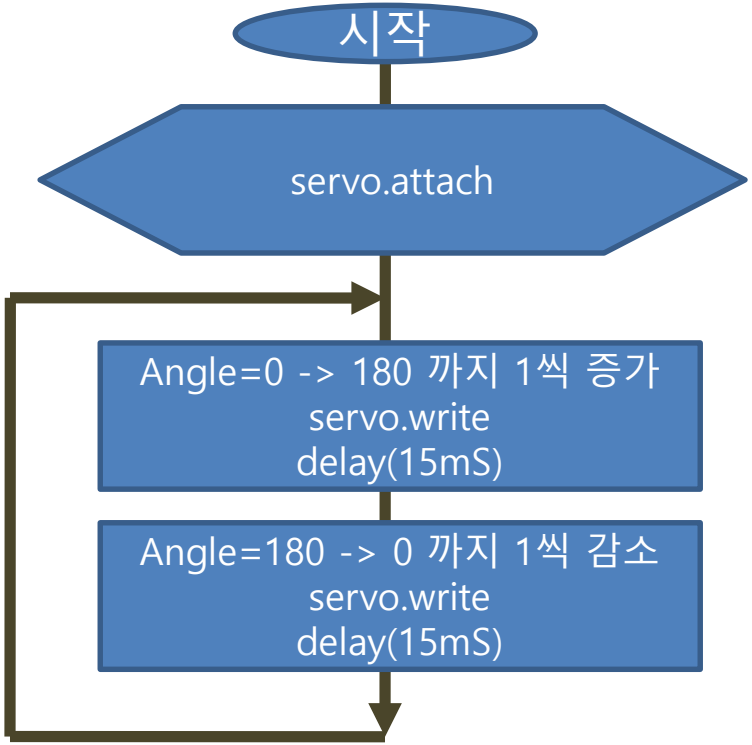
- 최대 12개 선언 가능



- 새 정의 이름
  - 영문자로 시작(변수명 정의 규칙)
  - 대소문자 구분
- Servo 모터를 여러 개 사용하면 별도 전원을 사용함

ServoMotorTest-1

- 우에서 좌로, 좌에서 우로 servo motor 동작시키기




```
#include <Servo.h>
#define ServoPin 3
Servo servo;

int Angle = 0;   //Servo Position in Degrees

void setup( ) {
  servo.attach( ServoPin );
}

void loop( ) {
for(Angle = 0; Angle <= 180; Angle++)  {
    servo.write(Angle);
    delay(15);
  }
  for(Angle = 180; Angle >= 0; Angle--)  {
    servo.write(Angle);
    delay(15);
  }
}
```

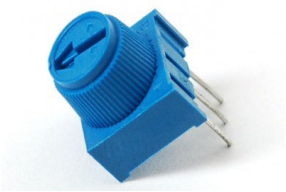
ServoMotorTest-2

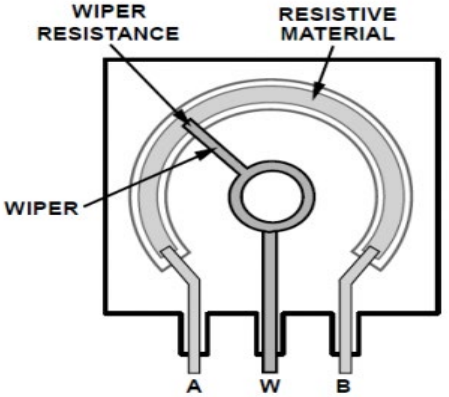
- Serial 통신을 이용하여 Servo Motor를 제어하자

```
#include <Servo.h>
#define ServoPin 3
Servo servo;

int Angle = 0;   //Servo Position in Degrees

void setup( ) {
  Serial.begin(9600);
  servo.attach( ServoPin );
}

void loop( ) {
  if ( Serial.available( ) > 0 ) {
    Angle = Serial.parseInt( );

    servo.write ( Angle );
    Serial.print ( "Angle = " );
    Serial.println ( Angle );
  }
}
```

Potentiometer

- 가변저항, 볼륨
- 저항값을 변화하여 제어장치의 설정값이나 신호값을 변화시킴

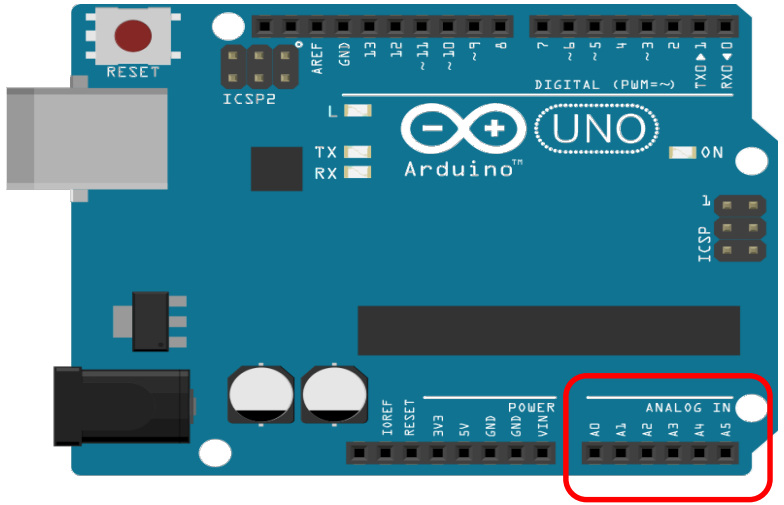




analogRead( ) 함수

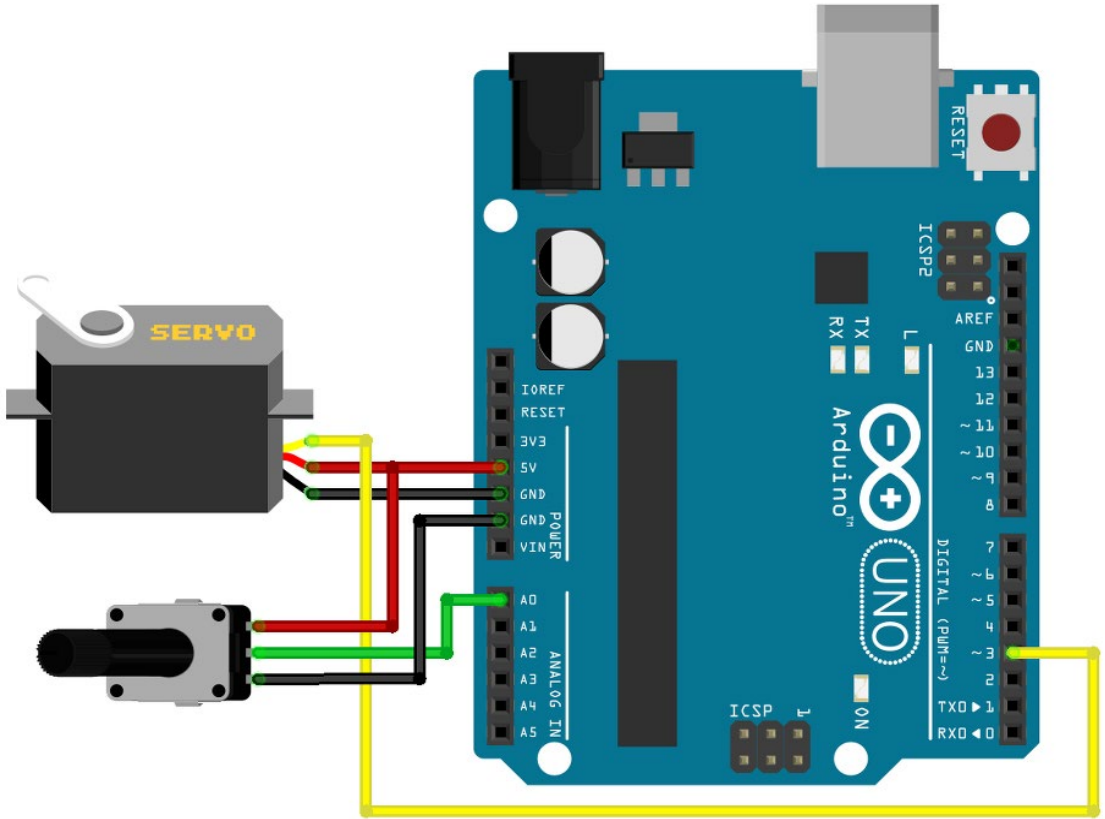
- 구문

  analogRead(**pin**)
- 파라메터
  - **Pin :** 아날로그 입력 핀 번호
  - A0-A5
- 리턴 값
  - 정수(0-1023)
- 예제




```
int  val =  analogRead( 0 );
```

Servo Motor Test 결선2



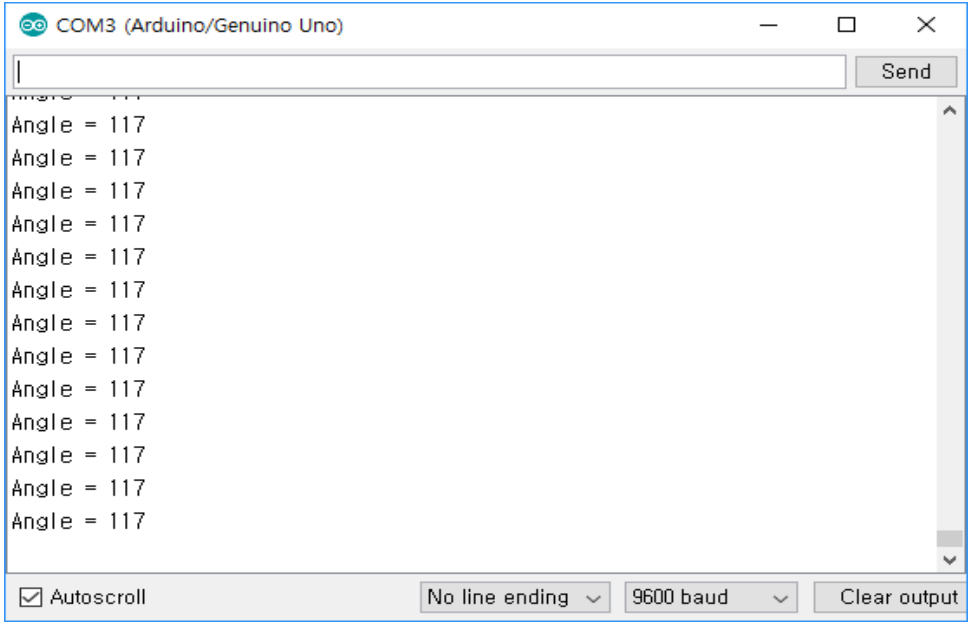
ServoMotorTest-3

- 볼륨(Volume) 값으로 서보모터 제어하기

```
#include <Servo.h>
#define ServoPin 3
Servo servo;

#define VolumePin A0

int Angle = 0;   //Servo Position in Degrees

void setup( ) {
  Serial.begin(9600);
  servo.attach( ServoPin );
}
```

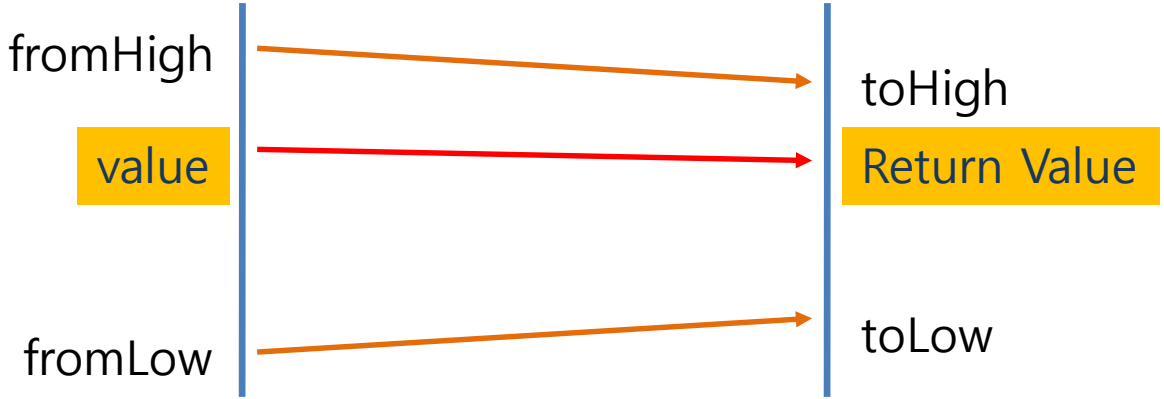
```
void loop( ) {

  int Volume = analogRead( VolumePin);

  Angle = Volume / 6;  //0~1023 -> 0~180 으로 변환

  servo.write ( Angle );
  Serial.print ( "Angle = " );
  Serial.println ( Angle );
  delay(100);

}
```

결과


COM3 (Arduino/Genuino Uno)
Send
Angle = 117
Angle = 117
Angle = 117
Angle = 117
Angle = 117
Angle = 117
Angle = 117
Angle = 117
Angle = 117
Angle = 117
Angle = 117
Angle = 117
Angle = 117
Autoscroll
No line ending
9600 baud
Clear output

map 함수

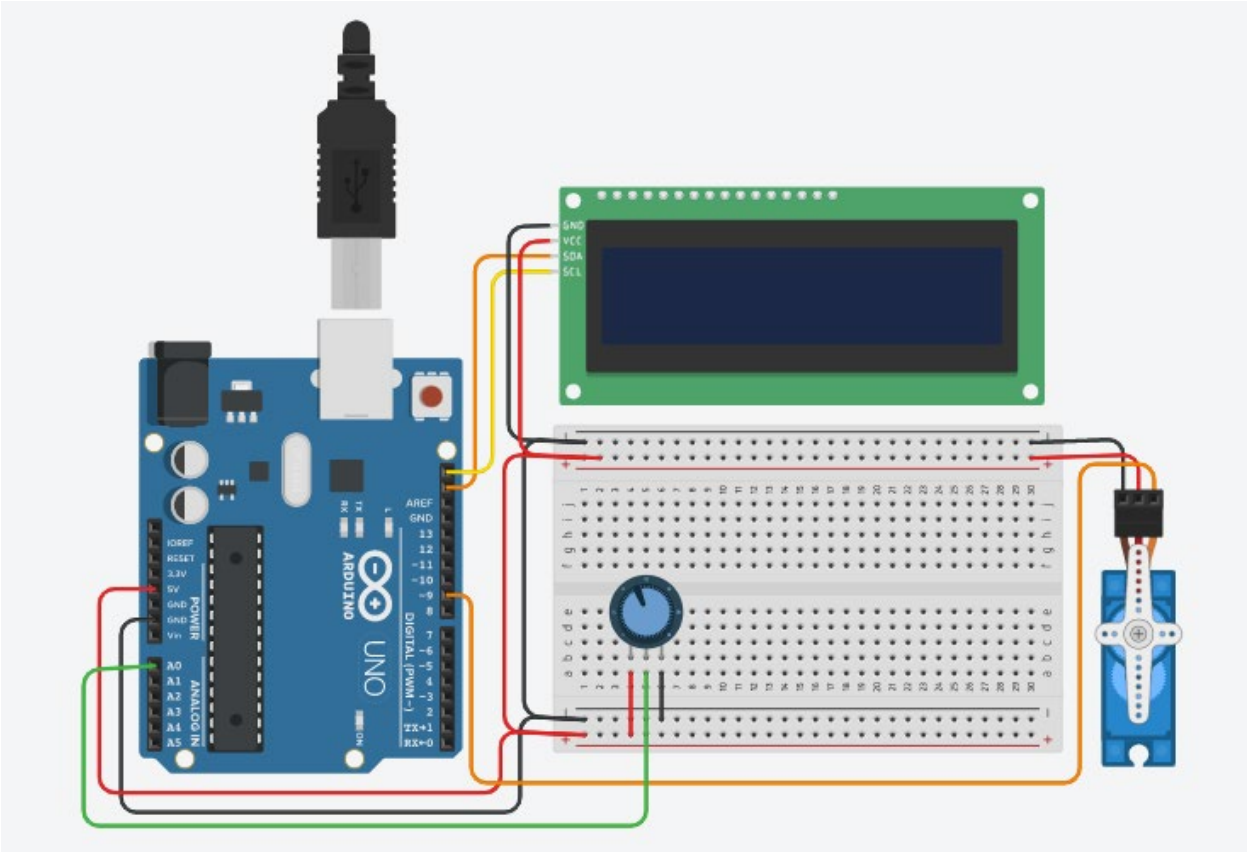
- map(value, fromLow, fromHigh, toLow, toHigh)
  - value : 변환 대상 값
  - fromLow : 변환 대상 값 중 최소값
  - fromHigh : 변환 대상 값 중 최대값
  - toLow : 변환 목표 값 중 최소값
  - toHigh : 변환 목표 값 중 최대값
  - **Return Value : (long) 변환값**



ServoMotorTest-4

```
#include <Servo.h>
#define ServoPin 3
Servo servo;

int Angle = 0;   //Servo Position in Degrees

void setup( ) {
  Serial.begin(9600);
  servo.attach( ServoPin );
}

void loop( ) {

  int Volume = analogRead( A0 );

  Angle = map ( Volume, 0, 1023, 0, 180 );

  servo.write ( Angle );
  Serial.print ( "Angle = " );
  Serial.println ( Angle );
}
```

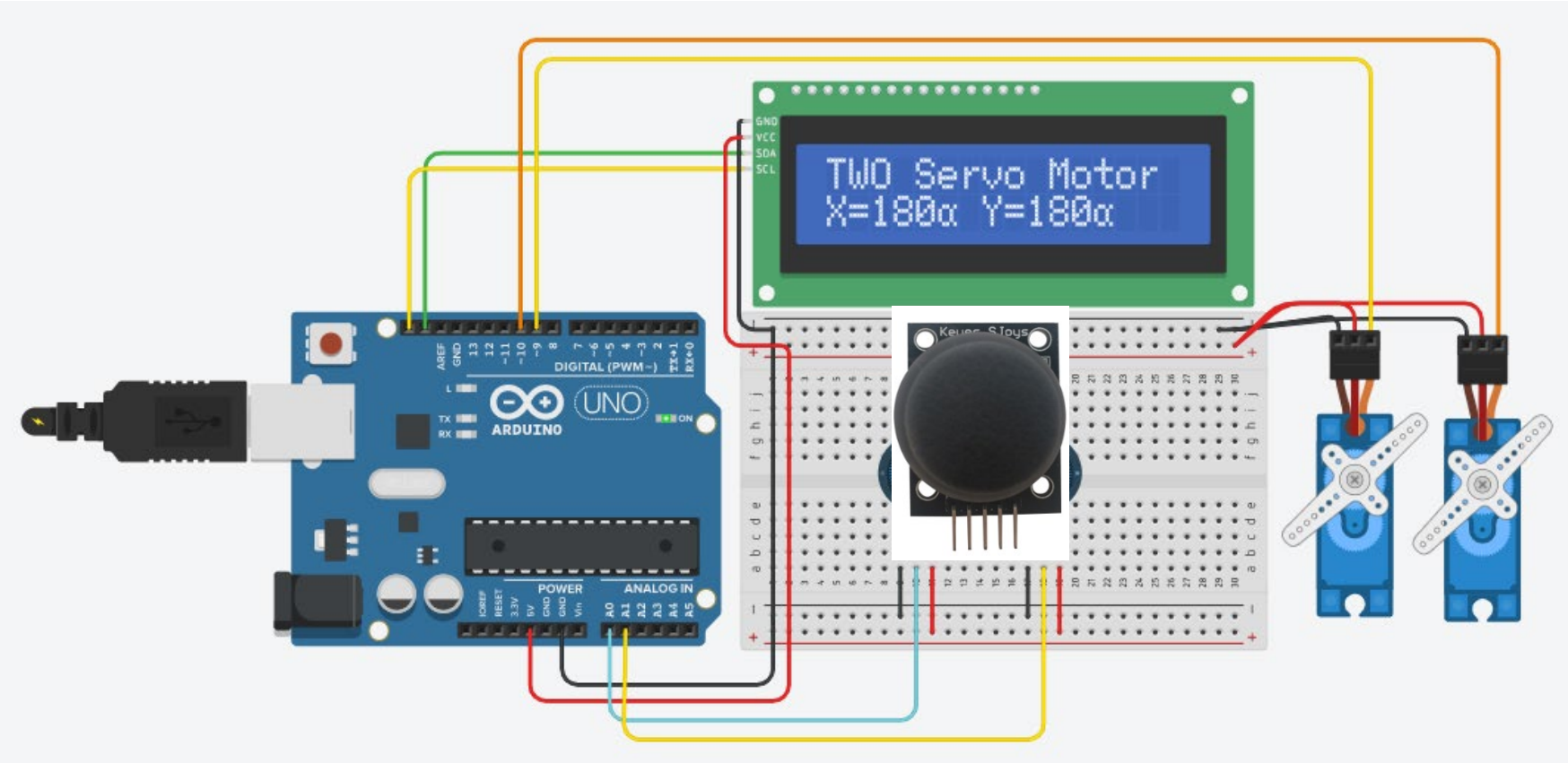
ServoMotor - i2c lcd



coding

```
#include <LiquidCrystal_I2C.h>
#include <Servo.h>

#define ServoPin 3
Servo Myservo;
LiquidCrystal_I2C lcd(0x27,16,2);

#define VolumePin A0

int Angle = 0;   //Servo Position in Degrees

void setup( ) {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  Myservo.attach( ServoPin );
  lcd.setCursor(1,0);
  lcd.print("Servo Motor");
}

void loop( ) {
 int  Volume = analogRead( VolumePin);
  Angle = map(Volume,0,1023,0,180);
  Myservo.write ( Angle );
  delay(15);

  // serial and lcd display call func.
   SelcdDisplay(Angle);

}
void SelcdDisplay(int Angle){
  Serial.print ( "Angle = " );
  Serial.println ( Angle );

  lcd.setCursor(2,0);
  lcd.print("Servo Motor");
  lcd.setCursor(4,1);
  lcd.print("Angle=");
  lcd.print(String(Angle) + char(0xdf));
  delay(100);
  lcd.clear();
}
```

Joystick- SeMotor



Coding

```
#include <LiquidCrystal_I2C.h>
#include <Servo.h>

LiquidCrystal_I2C lcd(0x27,16,2);

Servo servo1;
Servo servo2;

int joyX = 0;
int joyY = 1;

int servoVal1 = 0;
int servoVal2 = 0;

void setup()
{
  lcd.init();
  lcd.backlight();

  servo1.attach(9);
  servo2.attach(10);
}

void loop()
{

  servoVal1 = analogRead(joyX);
  servoVal1 = map(servoVal1, 0, 1023, 0, 180);
  servo1.write(servoVal1);

  servoVal2 = analogRead(joyY);
  servoVal2 = map(servoVal2, 0, 1023, 70, 180);
  servo2.write(servoVal2);
  SelcdDisplay(servoVal1,servoVal2);
  delay(15);
}

void SelcdDisplay(int Angle1,int Angle2){

  lcd.setCursor(2,0);
  lcd.print("Servo Motor");
  lcd.setCursor(0,1);
  lcd.print("A=");
  lcd.print(String(Angle1) + char(0xdf));
  lcd.setCursor(6,1);
  lcd.print("B=");
  lcd.print(String(Angle1) + char(0xdf));
  delay(100);
  lcd.clear();
}
```