

# JOYSTICK 사용하기

- 기초 학습
  - Math 함수
- Joystick 사용하기
  - 개 요
  - 예 제
- 응용 실습

Basic learning

---

# 기초 학습

- `min()` 함수
  - 두 숫자 중에서 최소 값 반환
  - `min(x, y)`의 형태로 사용
- `max()` 함수
  - 두 숫자 중에서 최대 값 반환
  - `max(x, y)`의 형태로 사용
- `abs()` 함수
  - 숫자의 절대 값을 계산하여 반환
  - `abs(x)`의 형태로 사용
- `constrain()` 함수
  - 지정된 범위의 값을 반환
  - `constrain(x, a, b)`의 형태로 사용
    - x값을 비교하여 a에서 b까지 범위의 값을 반환
    - x가 a 값보다 작으면 a
    - b보다 크면 b
    - a와 b사이의 값이면 x

- map() 함수
  - 입력 값을 현재 지정한 범위에서 다시 지정된 범위의 값으로 환산하여 반환
  - 입력 값은 현재 지정한 범위를 제한하지 않음
  - map(value, fromLOW, fromHIGH, toLOW, toHIGH)의 형태로 사용
    - fromLOW에서 fromHIGH의 값을 toLOW에서 toHIGH의 값으로 변환
    - value값이 fromLOW와 fromHIGH값의 범위를 벗어나도 그 벗어난 크기에 비례하게 변환되어 반환
- pow() 함수
  - 거듭제곱한 숫자 값을 계산
  - pow(x, exponent)의 형태로 사용
    - $x^{\text{exponent}}$  로 계산되어 반환
- sqrt() 함수
  - 숫자의 제곱근을 계산
  - sqrt(x)의 형태로 사용
    - $\sqrt{x}$ 로 계산되어 반환

- `sin()` 함수
  - 라디안 각도의 사인 값을 계산
  - 결과는 -1에서 1사이
  - `sin(rad)`의 형태로 사용
    - rad값은 radians의 값을 입력, -1에서 1사이의 결과 값을 반환
- `cos()` 함수
  - 라디안 각도의 코사인 값을 계산
  - 결과는 -1에서 1사이
  - `cos(rad)`의 형태로 사용
    - rad값은 radians의 값을 입력, -1에서 1사이의 결과 값을 반환
- `tan()` 함수
  - 라디안 각도의 탄젠트 값을 계산
  - 결과는  $-\infty$ 에서  $\infty$ 사이
  - `tan(rad)`의 형태로 사용
    - rad값은 radians의 값을 입력,  $-\infty$ 에서  $\infty$ 사이의 결과 값을 반환

---

# JOYSTICK 사용하기

- 목적
  - 게임의 컨트롤러라고 부르며 컴퓨터의 입력장치들 가운데 하나
  - 상하좌우 방향으로 레버를 움직이는 장치
- 관련이론
  - 조이스틱의 축은 위치에 따라 가변된 전압을 출력하는 디바이스
  - 동작 형태에 따라 포텐셔미터 타입, HFX 타입, 스위치 타입등으로 구분
    - 포텐셔미터 타입: 가변저항을 이용하는 형태
    - HFX 타입: Hall 센서를 사용하여 Hall Effect 효과를 이용하는 형태
    - 스위치 타입: 각 축마다 +, - 2개의 스위치가 달려있는 형태



포텐셔미터



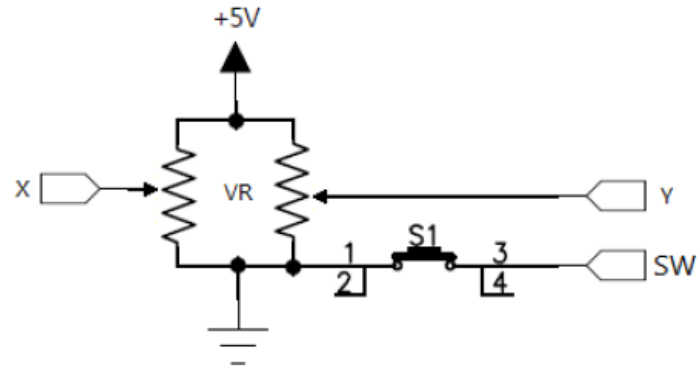
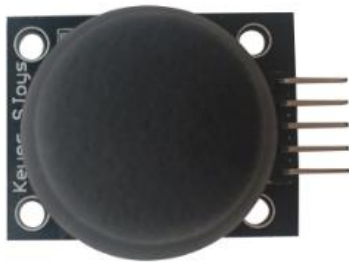
HFX



스위치



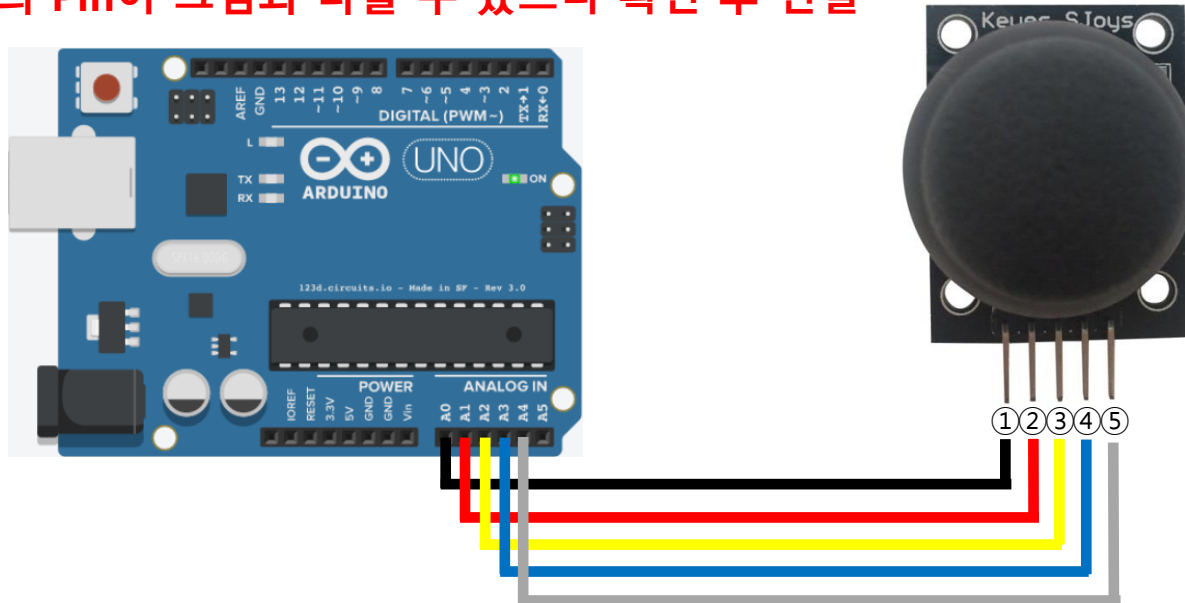
- 하드웨어 설명
  - Joystick 레버를 상하좌우 방향에 따라 움직이면 VR 값에 따라 전압값이 변동됨
  - S1(Tact switch)이 센터에 장착되어 레버를 누르게 되면 'Low'값이 출력 내부에 Pull up저항이 달려 있지 않기 때문에 프로그램 할 때 Pull up 설정을 해줘야 함



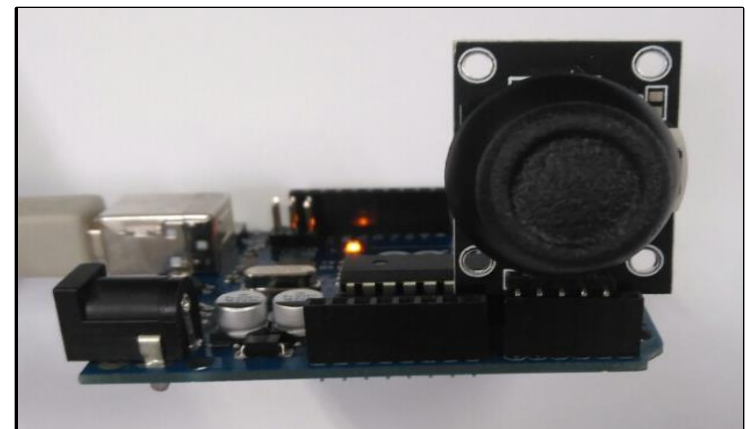
- 예제
  - Joystick을 움직여서 시리얼 모니터에 방향 값(-5~5까지 표현)과 Switch 값을 출력하시오.



- Uno Board와 Joystick 모듈 연결  
※ 센서의 Pin이 그림과 다를 수 있으니 확인 후 연결



Name	Touch Sensor Pin Number	Arduino Pin Number
GND	①	A0
VCC	②	A1
X	③	A2
Y	④	A3
SW	⑤	A4



- 프로그램 설명

- 초기화 구문

스위치 핀을 INPUT\_PULLUP으로 설정

플로팅 상태일때, 랜덤하지않고 HIGH로 고정시키는 역할

```
pinMode(pinGnd, OUTPUT);  
pinMode(pinVcc, OUTPUT);  
pinMode(pinSwitch, INPUT_PULLUP);  
digitalWrite(pinGnd, LOW);  
digitalWrite(pinVcc, HIGH);
```

- loop 구문

Joystick 모듈의 X, Y 값을 ADC하고, map()함수를 이용하여 값의 범위를 0~1023에서 -5~+5로 변경하여 출력

```
int x = map(analogRead(pinX), 0, 1023, -5, 5);  
int y = map(analogRead(pinY), 0, 1023, -5, 5);
```

- 전체 소스코드

```
1. int pinGnd = A0;
2. int pinVcc = A1;
3. int pinX= A2;
4. int pinY= A3;
5. int pinSwitch= A4;
6.
7. void setup()
8. {
9.   Serial.begin(115200);
10.  pinMode(pinGnd, OUTPUT);
11.  pinMode(pinVcc, OUTPUT);
12.  pinMode(pinX, INPUT);
13.  pinMode(pinY, INPUT);
14.  pinMode(pinSwitch, INPUT_PULLUP);
15.  digitalWrite(pinGnd, LOW);
16.  digitalWrite(pinVcc, HIGH);
17.}
```

- 전체 소스코드

```
18. void loop()
19. {
20.   int x = map(analogRead(pinX), 0, 1023, -5, 5);
21.   int y = map(analogRead(pinY), 0, 1023, -5, 5);
22.   Serial.print("X : ");
23.   Serial.print(x);
24.   Serial.print(" \tY : ");
25.   Serial.print(y);
26.   Serial.print(" \tSW : ");
27.
28.   if(!digitalRead(pinSwitch))
29.   {
30.     Serial.println("ON");
31.   }
32.   else
33.   {
34.     Serial.println("OFF");
35.   }
36.   delay(500);
37. }
```

Application practice

---

**응용 실습**

- 응용 문제
  - 6-Digit FND에 좌우 방향에 따라 숫자를 +, - 하고 클릭하면 Reset되도록 출력하시오. (FND 두자리만 사용)
- 구성
  - Arduino Uno
  - 6-Digit FND
  - Joystick

