

HIT 센서를 이용한 충격 감지

- 기초 학습
 - 인터럽트
- Hit 센서를 이용한 충격 감지
 - 개 요
 - 예 제
- 응용 실습



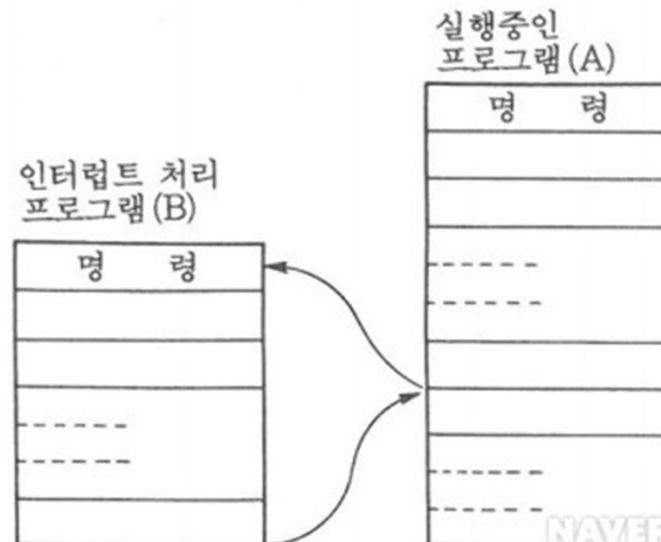
엣지아이랩

Basic learning

기초 학습

인터럽트

- 인터럽트란?
 - 실행 중인 프로그램을 일시 중단하고 다른 프로그램을 끼워 넣어 실행시키는 것
 - 인터럽트 요인이 되는 조건 발생
 - 실행 중인 프로그램(A)을 중단하여 강제적으로 특정한 주소로 제어를 이동
 - 준비되어 있는 인터럽트 처리 프로그램(B)을 실행
 - 처리가 끝나면 원래의 프로그램(A)으로 복귀 후 실행
 - 프로그램 처리의 효율화, 입출력 장치의 동시 동작 처리의 효율화
 - 즉, loop() 함수의 내용과 관계없이 지정한 핀의 상태가 변경 되었을 때 사용자가 정의한 기능의 함수를 실행하도록 하는 것



외부 인터럽트

- attachInterrupt(pin_interrupt, ISR, mode)
 - 인터럽트가 발생할 때 호출할 인터럽트 서비스 루틴(ISR)을 지정
 - pin_interrupt은 인터럽트 번호(핀 번호와 다름)
 - ISR은 인터럽트 발생시 호출될 함수(ISR : Interrupt Service Routines)
 - mode는 LOW, HIGH, CHANGE, RISING, FALLING 으로 설정

```
int pin = 2;
int pinInterrupt = 0;
int state = LOW;
void setup()
{
    pinMode(pin, OUTPUT);
    attachInterrupt(pinInterrupt, blink, CHANGE);
}
void loop() {}
void blink()
{
    state = !state;
    digitalWrite(pin, state);
    // pin 핀이 state 값에 따라 HIGH/LOW
}
```

외부 인터럽트

- detachInterrupt(pin)
 - 지정된 인터럽트를 해제
 - pin의 핀번호에 해당하는 인터럽트를 해제

- 아두이노 인터럽트 핀

Board	Int.0	Int.1	Int.2	Int.3	Int.4	Int.5
Uno	2	3				
Mega2560	2	3	21	20	19	18
Leonardo	3	2	0	1	7	

- 주의사항
 - 인터럽트 callback 함수는 파라미터를 전달하거나 리턴할 수 없음(예 : void xx())
 - ISR 안에서는 delay() 함수 사용할 수 없음
 - ISR 안에서 Serial data를 읽을 경우 값이 소실됨

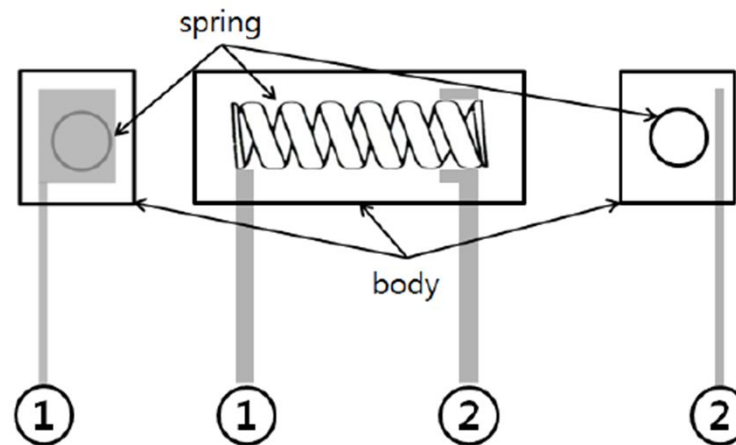
인터럽트

- interrupts() 함수
 - noInterrupts() 함수에 의해 금지된 인터럽트의 발생을 허용
 - 인터럽트를 허용하게 되면 인터럽트에 의한 작업이 우선 처리되어 작업의 순서와 시간에 변화가 발생
 - 중요한 작업을 위해 인터럽트는 일시적으로 금지 가능
- noInterrupts() 함수
 - 인터럽트의 발생을 금지
 - interrupts() 함수를 통해 인터럽트 발생을 다시 허용 가능

HIT 센서를 이용한 충격 감지

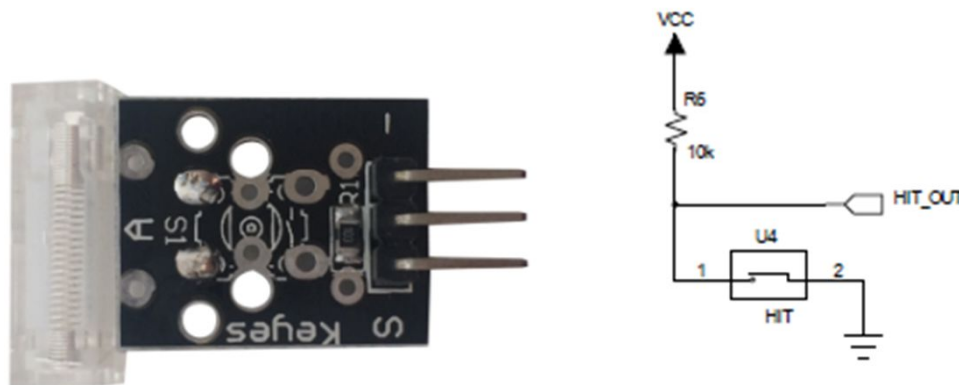
개요

- 목적
 - 디지털 출력
 - Hit 센서는 큰 충격에 반응
 - 유사센서로는 Shock sensor
- 관련이론
 - 1번과 스프링은 연결되어 있고 2번과 스프링은 약간 떨어져 있음
 - Hit 모듈을 두드리면 스프링의 움직임에 의해 2번과 순간적으로 연결되어 1번과 2번이 연결



개요

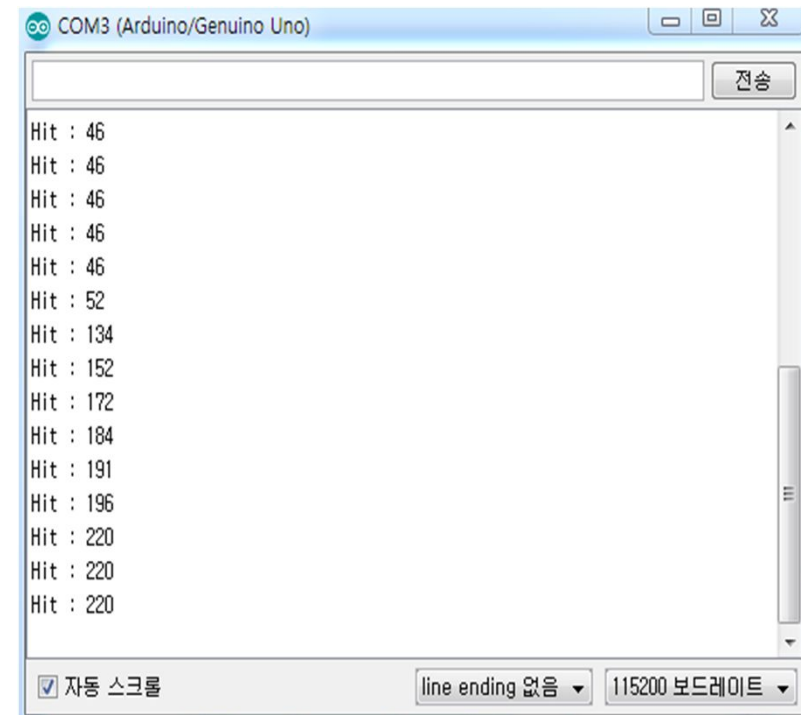
- 하드웨어 설명
 - 전원을 인가 후 Hit 모듈을 두드리면 모듈의 내부에 스프링이 접점과 연결
 - 풀업저항으로 기본 'High' 유지



[모듈과 구성회로]

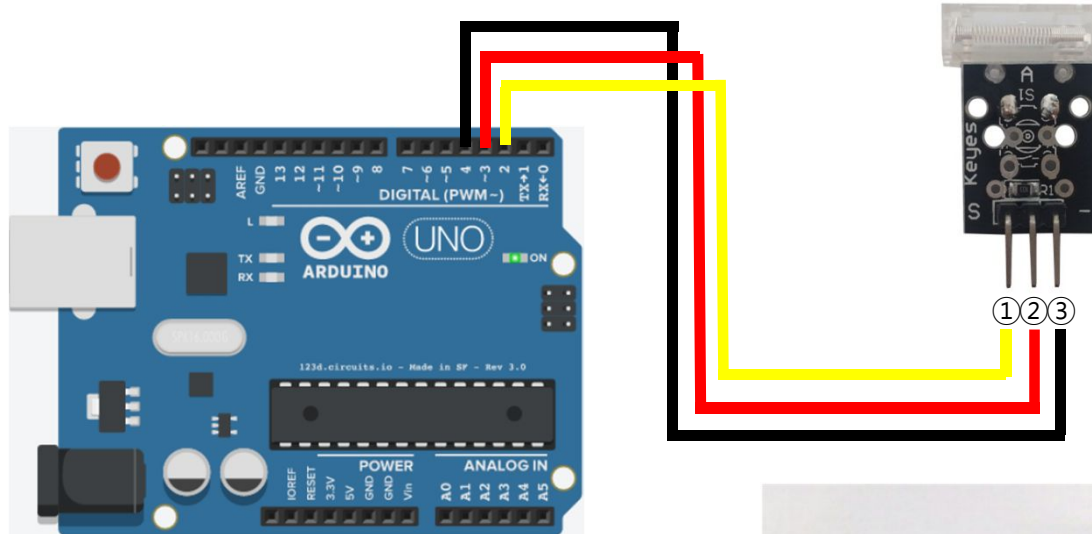
예제

- 예제
 - Hit 센서를 두드려서 모듈 내부의 스프링이 접점과 연결된 수를 카운트하여 시리얼 모니터에 출력하시오.



결선

- Uno 보드와 Hit 센서 모듈 연결
 ※ 센서의 Pin이 그림과 다를 수 있으니 확인 후 연결



Name	Hit Sensor Pin Number	Arduino Pin Number
HIT	①	2
VCC	②	3
GND	③	4



예제

- 프로그램 설명

- 사용할 핀 정의

```
int pinGnd = 4;  
int pinVcc = 3;  
int pinHit = 2;
```

- 초기화 구문

Serial port(UART 0)를 전송속도 115200, 데이터 비트 8, 패리티 없음, 스톱비트 1로 설정

GND, VCC 핀을 출력, Hit 핀을 입력으로 설정, GND 핀에 LOW, VCC 핀에 HIGH를 출력

```
Serial.begin(115200);  
  
pinMode(pinGnd, OUTPUT);  
pinMode(pinVcc, OUTPUT);  
pinMode(pinHit, INPUT);  
digitalWrite(pinGnd, LOW);  
digitalWrite(pinVcc, HIGH);
```

예제

- 프로그램 설명

- 초기화 구문

외부 인터럽트를 설정

외부소스는 핀 2번(INT0 – pinHit), ISR은 hitISR, mode는 RISING(상승엣지)으로 설정

```
// External interrupt setup
// source : pinHit, ISR : hitISR, mode : RISING
attachInterrupt(0, hitISR, RISING);
```

- hitISR 함수

외부 인터럽트 서비스 루틴으로 인터럽트가 발생하면 count 값이 1 증가

```
void hitISR()
{
    count++;    // 1 increase
}
```

- 전체 소스코드

```
1. int pinGnd = 4;
2. int pinVcc = 3;
3. int pinHit = 2;
4. int count = 0;
5.
6. void setup() {
7.   Serial.begin(115200);
8.
9.   pinMode(pinGnd, OUTPUT);
10.  pinMode(pinVcc, OUTPUT);
11.  pinMode(pinHit, INPUT);
12.  digitalWrite(pinGnd, LOW);
13.  digitalWrite(pinVcc, HIGH);
14.
15.  // External interrupt setup
16.  // source : pinHit, ISR: hitISR, mode : RISING
17.  attachInterrupt(0, hitISR, RISING);
18. }
```

예제

- 전체 소스코드

```
19. void loop() {  
20.   Serial.print("Hit : ");  
21.   Serial.println(count);  
22.   delay(500);  
23. }  
24.  
25. // External ISR function  
26. void hitISR(void)  
27. {  
28.   count++; // 1 increase  
29. }
```

Application practice

응용 실습

응용 실습

- 응용 문제
 - LED0~LED7까지 1초마다 순서대로 켜지다가 Hit 센서에 충격을 가하면 LED가 모두 꺼지게 하고 다시 LED0부터 켜지게 하시오.
- 구성
 - Arduino Uno
 - LED/Switch모듈
 - Hit 센서

