

SWITCH 읽기

- 기초 학습
 - 제어구조
- Switch 읽기
 - 개 요
 - 예 제(1)
 - 예 제(2)
- 응용 실습



엣지아이랩

Basic learning

기초 학습

제어 구조

- if
 - 조건부에 의해 참이면 동작
 - if(조건부)의 형태로 되어 있으며 조건부가 참이면 동작

```
if(x>50)
{
    // 조건부가 참일 때 실행
}
```

제어 구조

- if - else
 - 조건부가 참이면 if 문을 실행
 - 거짓이면 else 문을 실행

```
if(x>50)
{
    // 조건부가 참일 때 실행
}
else
{
    // 조건부가 거짓일 때 실행
}
```

제어 구조

- 조건부가 2개 이상인 경우
 - 조건A를 판단하여 참이면 action A를 실행
 - 거짓인 경우에는 다시 조건B를 판단
 - 참이면, action B를 실행
 - 거짓이면, action C를 실행
 - “else if” 문은 여러 개가 반복적 사용 가능

```
if(조건A)
{
    // action A
}
else if(조건B)
{
    // action B
}
else
{
    // action C
}
```

제어 구조

- for
 - for문은 중괄호로 묶인 블록을 반복하는데 사용
 - 증분 카운터는 루프의 증가 및 종료하기 위한 조건에 사용
 - 횟수가 결정된 반복 작업에 유용
 - 종종 데이터/핀의 집합에서 작동하도록 배열과 함께 사용

```
for(initialization; condition; increment or decrement)
{
    // action
}
```

제어 구조

- initialization은 처음 한 번만 실행
 - 반복할 때마다 condition으로 루프 실행 여부 판단
 - 판단된 조건이 참이면 increment or decrement에 의해 증가/감소 실행
 - 다음 조건을 다시 판단
 - 조건이 거짓이면 루프가 종료
- for문이 사용되는 예
 - i를 0으로 초기화하고 i가 255보다 작거나 같으면 계속 실행
 - 실행할 때마다 i값을 증가
 - i 값이 255보다 커지면 루프 종료

```
for(i=0; i<=255; i++)  
{  
    analogWrite(PWMpin, i);  
}
```

제어 구조

- switch-case
 - if문과 유사한 프로그램의 흐름을 제어문
 - case에 지정된 값을 switch 문의 변수 값과 비교
 - case의 값이 변수와 일치하면 해당 case 문에서 코드가 실행
 - break 키워드
 - switch 문을 종료하는 역할
 - 일반적으로 각각의 case 문의 끝부분에 사용
 - break 키워드가 없다면 switch 문이 중단 될 때까지 다음을 실행하거나 switch 문의 끝부분에 도달

```
switch(var)
{
    case label1:
        // statements
        break;
    case label2:
        // statements
        break;
    default:
        // statements
}
```


제어 구조

- while
 - 괄호 안의 조건이 거짓이 될 때까지 무한히 계속 루프를 반복 실행
 - expression을 판단하여 거짓이 되기 전까지 루프 반복

```
while(expression)
{
    // statements
}
```

제어 구조

- do - while
 - do-while은 while 문과 유사
 - 다른 점은 조건식 판단을 언제 하느냐의 차이
 - while 문은 먼저 조건식을 판단하여 루프가 실행
 - do-while 문은 처음 한번 루프 실행
 - 이후는 while 문과 동일

```
do
{
    // statements
}while(expression);
```

제어 구조

- break
 - do-while, while, for문의 루프 조건 종료
 - switch문 종료
- continue
 - 반복문(do-while, while, for)의 현재 반복의 나머지 부분을 건너 뛴
 - 루프의 처음으로 이동
 - 루프의 조건식을 검사 후 반복 진행 계속
- return
 - 함수에서 사용
 - 함수를 종료하고 결과 값 반환

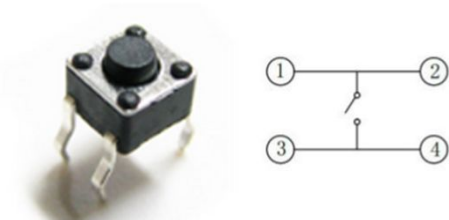
```
return;  
or  
return value;
```

Read a switch

SWITCH 읽기

개요

- 목적
 - 디지털 입력(Digital In)의 개념 이해
 - 푸쉬버튼 스위치 동작 이해
 - Uno Board는 $0.6 \times V_{cc}(0.6 \times 5 = 3V) \sim V_{cc} + 0.5$ 까지 'High'로 인식
 - $0.5 \sim 0.3 \times V_{cc}(0.3 \times 5 = 1.5V)$ 까지 'Low'로 인식
- 관련이론
 - Tact switch : "버튼" 또는 "Key"라고 불리고, 1회로 1접점의 누르면 "ON", 손을 떼면 "OFF" 방식의 회로구성
 - Push button switch : 누름 스위치라고 불리고, PC의 전원, 자동차 등 누름방식의 스위치로써 Locking Type과 Nonlock Type으로 분류



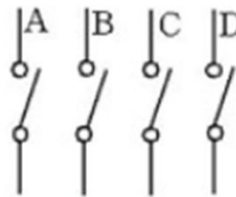
Tact switch



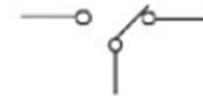
Push button switch

개요

- Slide switch
 - 사이즈와 용량의 따라 작은사이즈는 MP3의 Hold용으로 사용되고, 큰사이즈는 자동, 수동 변환 또는 회로 변환 등으로 사용
- Toggle switch
 - 레버를 움직임으로써 ON/OFF 상태를 변경할 수 있는 방식



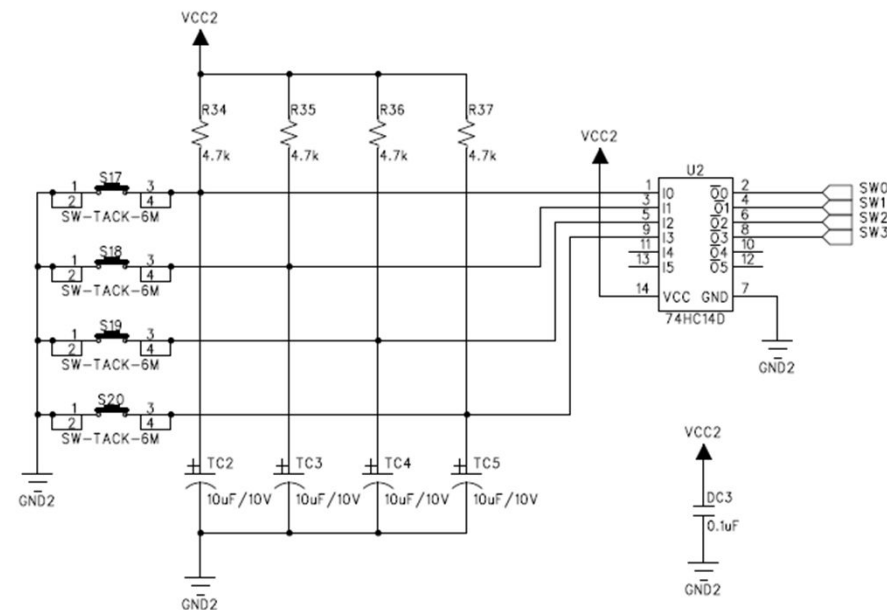
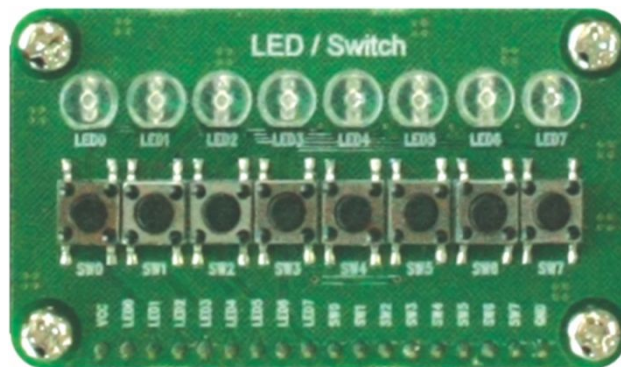
Slide switch



Toggle switch

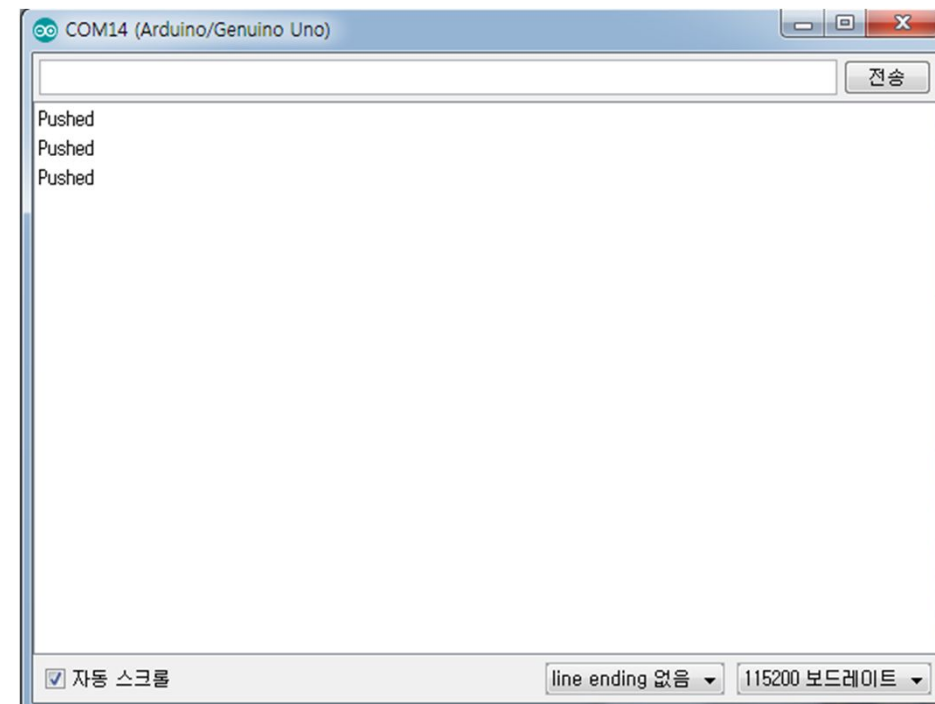
개요

- 하드웨어 설명
 - 아래 회로의 동작을 살펴보면 평상시에는 U2(인버터) 입력상태가 PullUP 되어 HIGH상태의 입력이 LOW상태로 반전되어 출력
 - 스위치를 누르면 'VCC2' 전원이 'GND'2로 흐르게 되고 U2(인버터)의 입력이 'LOW'가 되어 인버터의 출력은 'HIGH'



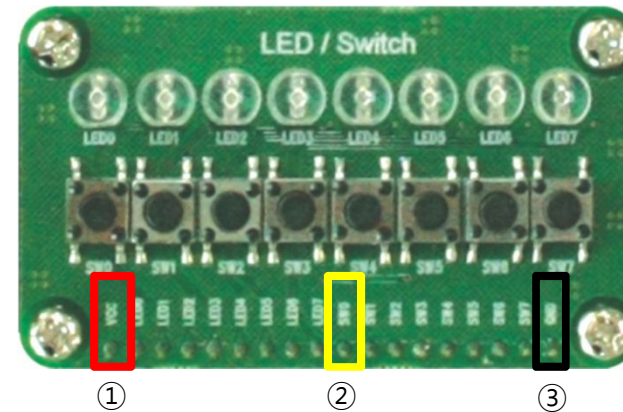
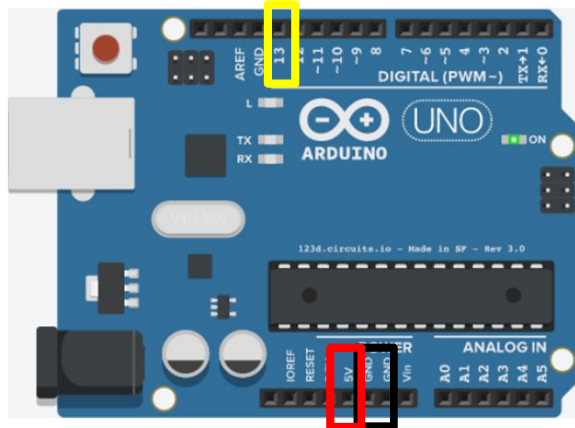
예제(1)

- 예제
 - SW0 버튼을 누르면 시리얼 모니터에 "Pushed"를 출력하시오.



예제(1)

- Uno보드와 스위치 모듈 결선



Name	Switch Pin Number	Arduino Pin Number
VCC	①	5V
SW0	②	13
GND	③	GND



예제(1)

- 프로그램 설명

- 사용할 핀 정의

```
int pinSwitch = 13;
```

- 초기화 구문

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

Switch 핀을 입력으로 설정

```
Serial.begin(115200);
```

```
pinMode(pinSwitch, INPUT);
```

예제(1)

- 프로그램 설명
 - loop 구문
Switch 값을 읽어 1이면 PUSH 출력하고 0.5초 지연

```
if(digitalRead(pinSwitch))  
{  
    Serial.println("Pushed");  
    delay(500);  
}
```

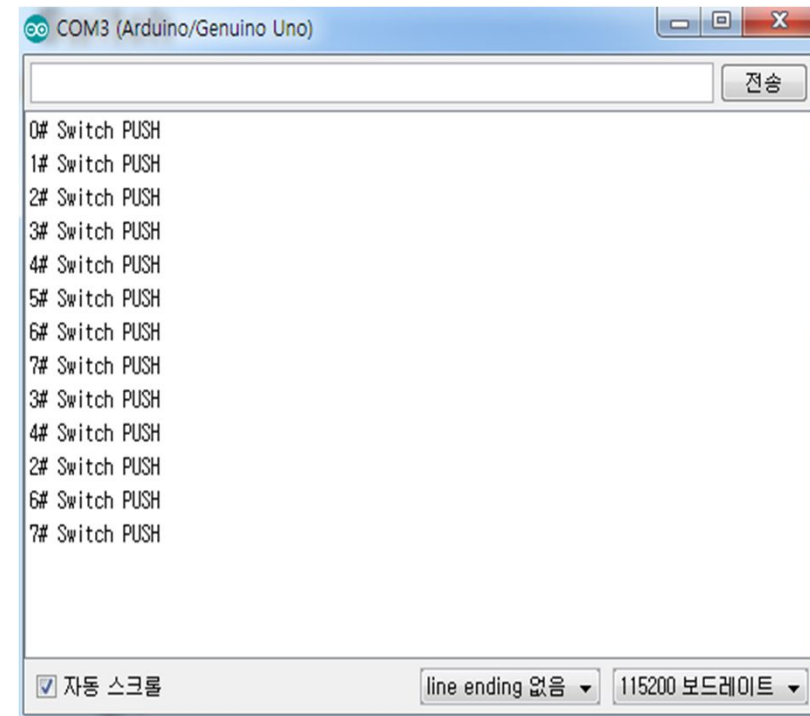
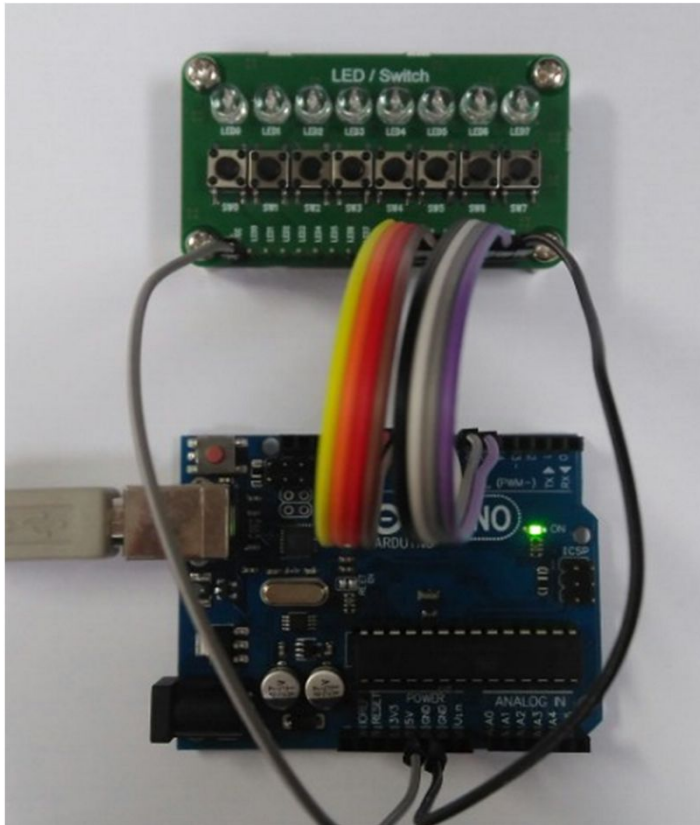
예제(1)

- 전체 소스코드

```
1. int pinSwitch = 13;
2.
3. void setup() {
4.   Serial.begin(115200);
5.   pinMode(pinSwitch, INPUT);
6. }
7.
8. void loop() {
9.   if(digitalRead(pinSwitch)) {
10.    Serial.println("Pushed");
11.    delay(500);
12.   }
13. }
```

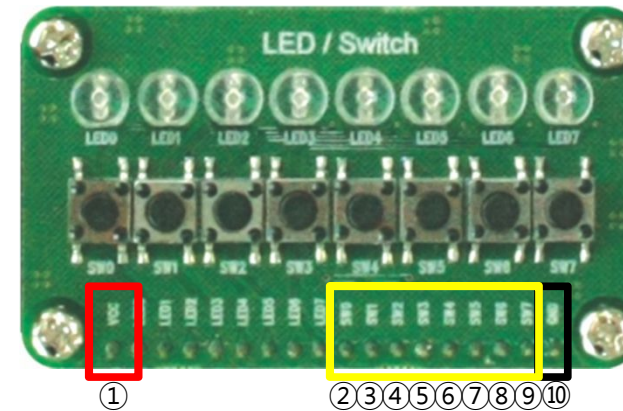
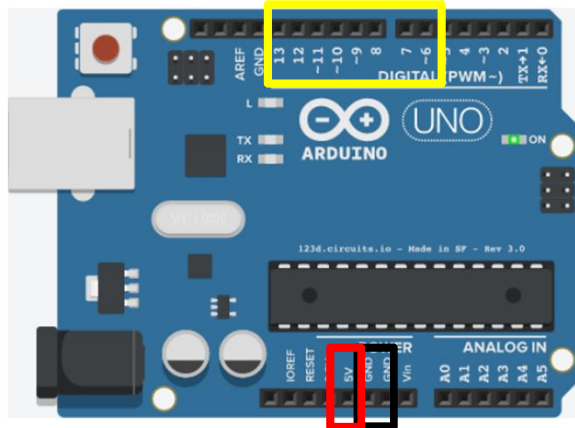
예제(2)

- 예제
 - 임의의 버튼을 누르면 시리얼 모니터에 해당 switch의 번호와 "Switch PUSH"를 출력하시오.

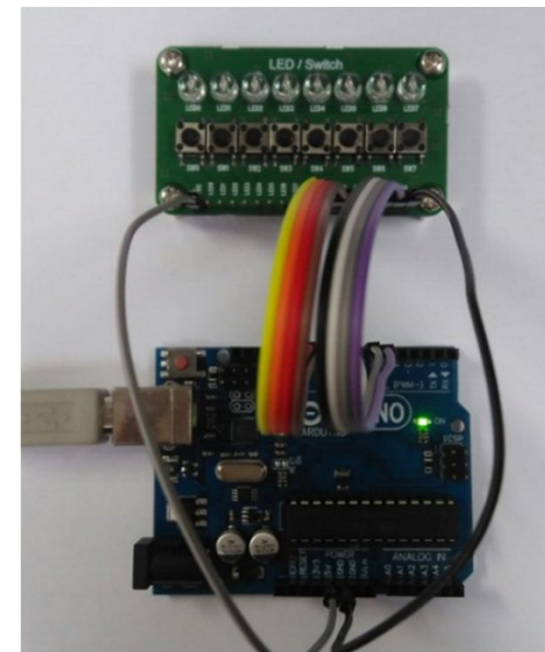


예제(2)

- Uno보드와 스위치 모듈 결선



Name	Switch Pin Number	Arduino Pin Number
VCC	①	5V
SW0	②	13
SW1	③	12
SW2	④	11
SW3	⑤	10
SW4	⑥	9
SW5	⑦	8
SW6	⑧	7
SW7	⑨	6
GND	⑩	GND



예제(2)

- 프로그램 설명

- 사용할 핀 정의

```
int aPinSwitch[8] = {13, 12, 11, 10, 9, 8, 7, 6};
```

- 초기화 구문

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

Switch 핀을 입력으로 설정

```
Serial.begin(115200);

for(int i=0; i<8; i++)
{
    pinMode(aPinSwitch[i], INPUT);
}
```

예제(2)

- 프로그램 설명
 - loop 구문
Switch 값을 읽어 1이면 PUSH 출력하고 0.5초 지연

```
for(int i=0; i<8; i++)  
{  
    if(digitalRead(aPinSwitch[i]))  
    {  
        Serial.print(i);  
        Serial.print("#");  
        Serial.println("Switch PUSH");  
    }  
}  
delay(500);
```


예제(2)

- 전체 소스코드

```
1. int aPinSwitch[8] = {13,12,11,10,9,8,7,6};
2.
3. void setup() {
4.   Serial.begin(115200);
5.   for(int i=0; i<8; i++) {
6.     pinMode(aPinSwitch[i], INPUT);
7.   }
8. }
9.
10. void loop() {
11.   for(int i=0; i<8; i++) {
12.     if(digitalRead(aPinSwitch[i])) {
13.       Serial.print(i);
14.       Serial.print("# ");
15.       Serial.println("Switch PUSH");
16.     }
17.   }
18.   delay(500);
19. }
```

Application practice

응용 실습

응용 실습

- 응용 문제
 - 스위치를 한번 누르면 번호에 맞는 LED를 켜고, 다시 누르면 끄게하여 이를 시리얼 모니터에 출력하시오.(0~3번 LED와 Switch 이용)
- 구성
 - Arduino Uno
 - LED/Switch 모듈

