

# การเชื่อมต่อกับอุปกรณ์อินพุท (สวิตช์)

## สวิตช์แบบต่างๆ

สวิตช์ (switch) SPST (ON OFF Switch หรือ Single Pole Sing Throw (SPST))



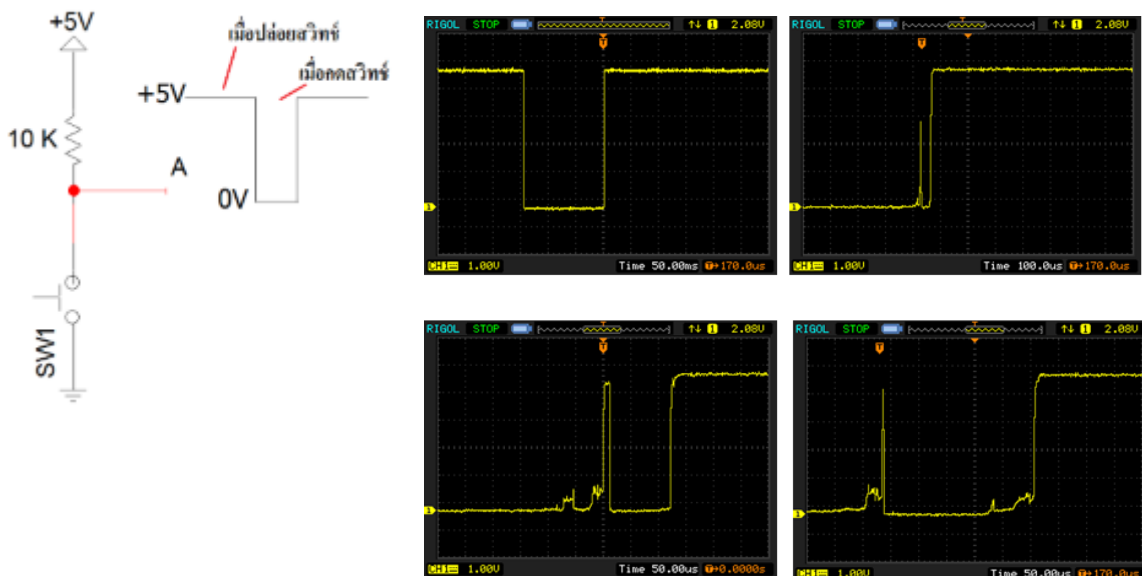
สวิตช์ (switch) SPDT (ON OFF Switch แบบ 2 ทาง หรือ Single Pole Double Throw (SPDT))



สวิตช์แบบกดติดปล่อยดับ (push button switch)



สัญญาณที่ได้จากสวิตช์ Push Button เมื่อกดและปล่อย

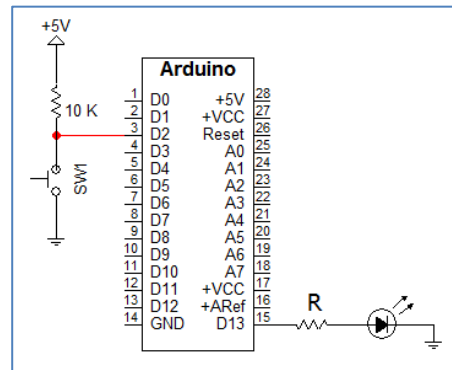


## การเขียนโปรแกรมอ่านสวิตช์ แบบที่ 1

โปรแกรมอ่านสถานะสวิตช์ต้องแก้ปัญหาการกระเด็น (Bounce) ของหน้าสัมผัสสวิตช์ได้

```
#define sw1 2
#define readsw1 digitalRead(sw1);

if(readsw1 == 0)
{
    ....งานที่ต้องทำเมื่อกดสวิตช์
    do{ //ตรวจสอบว่าปล่อยสวิตช์หรือยัง
        delay(10);
        readsw1 = digitalRead(SW1);
    }while(readsw1==0);
}
```



### การทดลอง

เมื่อกด SW1 ครั้งแรกให้ LED ติด

เมื่อกด sw1 ครั้งที่ 2 ให้ LED ดับ

การทำงานเมื่อกดสวิตช์ SW1 จะสลับกันไปเช่นนี้

### อัลกอริทึม (Algorithm)

Algorithm คือ กระบวนการแก้ปัญหาที่สามารถอธิบายออกมาเป็นขั้นตอนที่ชัดเจน วิธีการในการอธิบาย Algorithm มีหลายวิธีได้แก่

- 1) Natural Language อธิบายแบบใช้ภาษาที่เราสื่อสารกันทั่วไป
- 2) Pseudocode อธิบายด้วยคำสังเทียม
- 3) Flowchart อธิบายด้วยแผนผังภาพ

อัลกอริทึมการควบคุมการติด-ดับของ LED 1 ดวงด้วย Push button switch 1 อัน

1. กำหนดค่าเบื้องต้น
  - ตำแหน่งสวิตช์
  - ตำแหน่ง LED
  - สถานะ LED ให้เป็นดับ
  - กำหนดสถานะของพอร์ต พอร์ตอินพุต และพอร์ตเอาต์พุต
2. อ่านค่าสวิตช์
3. ตรวจสอบว่ามีการกดสวิตช์หรือไม่
4. ถ้ากดสวิตช์ ให้เปลี่ยนสถานะของ LED เป็นตรงกันข้าม
5. ส่งค่าออกเอาต์พุต สั่งให้ LED ทำงานตามสถานะ
6. ย้อนกลับไปข้อ 2

## โปรแกรมควบคุมการติด-ดับของ LED 1 ดวง

```
#define sw1 2
#define readsw1 digitalRead(sw1)
#define led1 11

int ledState = LOW;

void setup() {
  pinMode(sw1, INPUT);
  pinMode(led1, OUTPUT);
}
void loop()
{
  if(readsw1 == 0)
  {
    ledState = !ledState; //toggle led state
    do
    {
      delay(10); //delay time 10 mS
    }while(readsw1==0);
  }
  digitalWrite(led1, ledState); //turn on/off LED
}
```

## การเขียนโปรแกรมอ่านสวิตช์ แบบที่ 2

ใช้การจับเวลาของการกดสวิตช์ ถ้านานกว่าค่าๆหนึ่งจึงถือว่าเป็นการกดสวิตช์

การจับเวลาใช้การอินเตอร์รัพท์จาก Timer1

\*\*ดูการติดตั้ง Timer 1 ได้จาก <https://sites.google.com/site/eplearn/arduino-project/03-led-8x8> \*\*

```
#include <TimerOne.h>

#define sw1 2
#define sw1_on digitalRead(sw1)==0
#define led1 11
#define time_press 5
#define on true
#define off false
unsigned char swtime1;
unsigned char sw1_press;
int ledState = LOW;

void setup() {
  pinMode(sw1, INPUT);
  pinMode(led1, OUTPUT);
  Timer1.initialize(10000); // set a timer of length 10 milliseconds
  Timer1.attachInterrupt( timerIsr ); // attach the service routine here
  swtime1 = 0;
  sw1_press = off;
  digitalWrite(led1, ledState);
}
void loop() {
  if(sw1_press)
  {
    ledState = !ledState; //toggle led state
    digitalWrite(led1, ledState); //turn on/off LED
    while(sw1_on) swtime1 = 0;
  }
}
```

```

// -----
// Scan keyboard
// -----
void timerIsr()
{
  if(sw1_on) ++swtime1;
  else swtime1 = 0;
  if(swtime1 > time_press){
    sw1_press = on;
    swtime1 = time_press+1;
  }else{
    sw1_press = off;
  }
}
}

```

โปรแกรมอยู่ที่ <https://sites.google.com/site/eplearn/arduino-project/lab-02>

### งานมอบหมาย 1 ควบคุมการติด-ดับของ LED 3 ดวง

ให้เขียนโปรแกรมควบคุมการติด-ดับของ LED1 ถึง

LED3 ด้วยสวิตช์ SW1 ถึง SW3 ตามลำดับ โดย

SW1 ควบคุมการทำงานของ LED1

SW2 ควบคุมการทำงานของ LED2

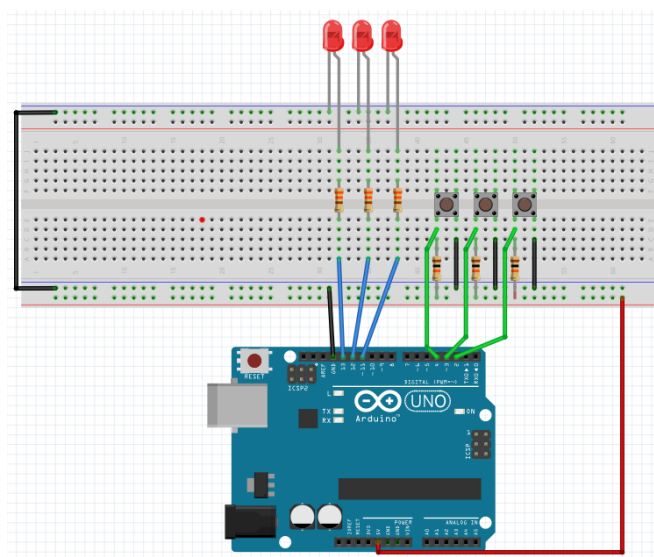
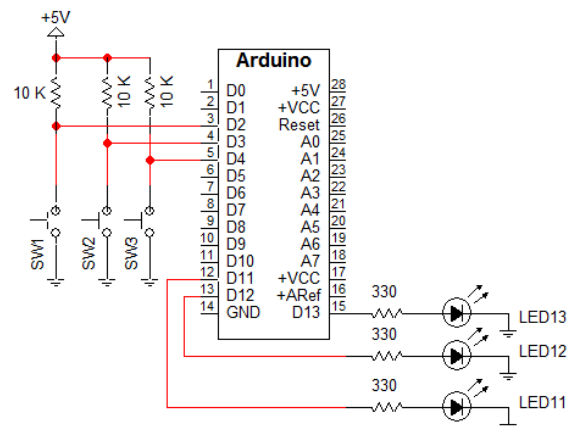
SW3 ควบคุมการทำงานของ LED3

การควบคุมให้ทำเหมือน Lab 2

หมายเหตุ LED3 ใช้ที่มีอยู่แล้วในบอร์ด ส่วน LED1

และ LED2 ต้องต่อเพิ่ม เช่นเดียวกับ SW1, SW2, และ

SW3



## งานมอบหมาย 2 ใช้สวิตช์ SW1 ควบคุมการติด-ดับของ LED 3 ดวง

เมื่อกด sw1 ครั้งแรกให้ LED1 ติด

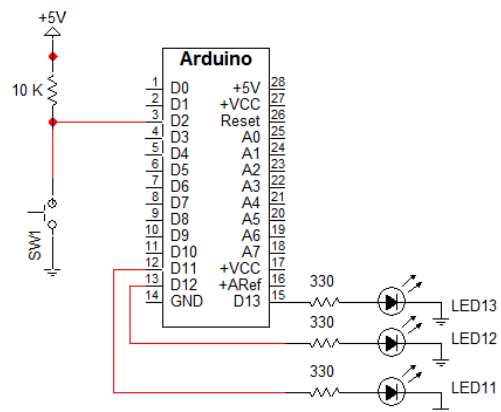
เมื่อกด sw1 ครั้งที่ 2 ให้ LED1 และ LED2 ติด

เมื่อกด sw1 ครั้งที่ 3 ให้ LED1 LED2 และ LED3 ติด

เมื่อกด sw1 ครั้งที่ 4 ให้ LED ทั้ง 3 ดวงดับหมด

เมื่อกด sw1 ครั้งที่ 5 จะเป็นเหมือนการกดครั้งที่ 1

การทำงานจะเป็นเช่นนี้ตลอดไป



### อัลกอริทึมการควบคุมการติด-ดับของ LED 3 ดวงด้วย Push button switch

1. กำหนดค่าเบื้องต้น
  - ตำแหน่งสวิตช์
  - ตำแหน่ง LED
  - กำหนดตัวแปรจำนวนครั้งการกดสวิตช์ ค่าเริ่มต้นให้เป็น 0
  - กำหนดสถานะของพอร์ต พอร์ตอินพุต และพอร์ตเอาต์พุต
2. อ่านค่าสวิตช์
3. ตรวจสอบว่ามีการกดสวิตช์หรือไม่
4. ถ้ากดสวิตช์ ให้เพิ่มตัวแปรจำนวนครั้งการกดสวิตช์
  - ถ้าจำนวนครั้งเท่ากับ 4 ให้เปลี่ยนค่าเป็น 0
5. ส่งค่าออกเอาต์พุต ตามค่าจำนวนครั้งการกดสวิตช์
  - ถ้าเท่ากับ 0 ให้ LED ดับหมด
  - ถ้าเท่ากับ 1 ให้ LED ติด 1 ดวง
  - ถ้าเท่ากับ 2 ให้ LED ติด 2 ดวง
  - ถ้าเท่ากับ 3 ให้ LED ติด 3 ดวง
6. ย้อนกลับไปข้อ 2