01076101
วิศวกรรมคอมพิวเตอร์เบื้องต้น
Introduction to Computer Engineering

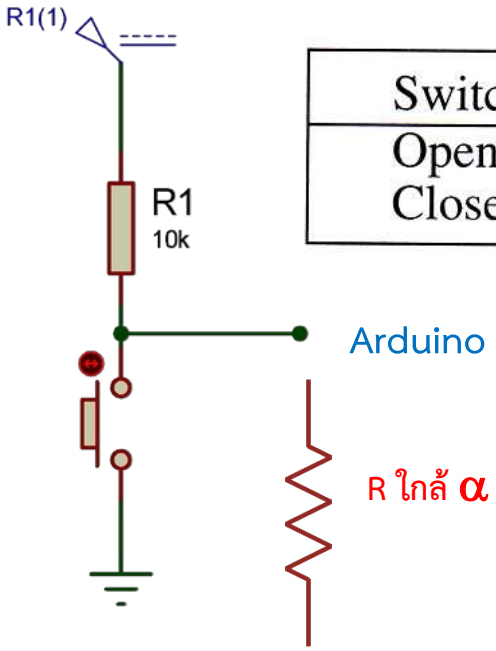
Arduino #2

Digital Input, Switch, Interrupt

Switch and Pullup Pulldown

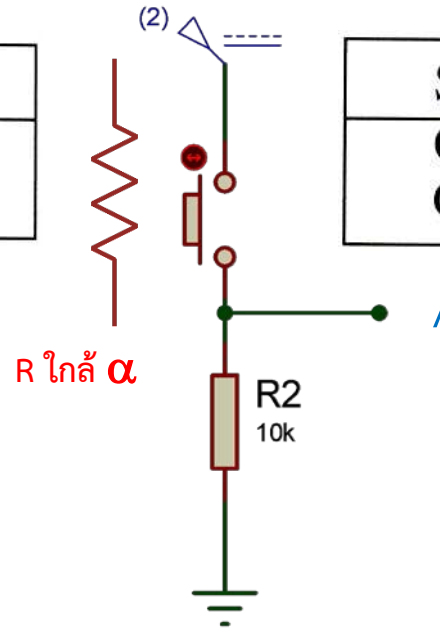
Digital Input

Pull Up



Switch	Output
Open	+5V
Closed	0V

Pull Down



Switch	Output
Open	0V
Closed	+5V

$$R_{\text{total}} = R_1 || R_2 = \frac{R_1 \cdot R_2}{R_1 + R_2}$$

Digital Input

SETUP PINMODE

Syntax:

pinMode(pin, mode)

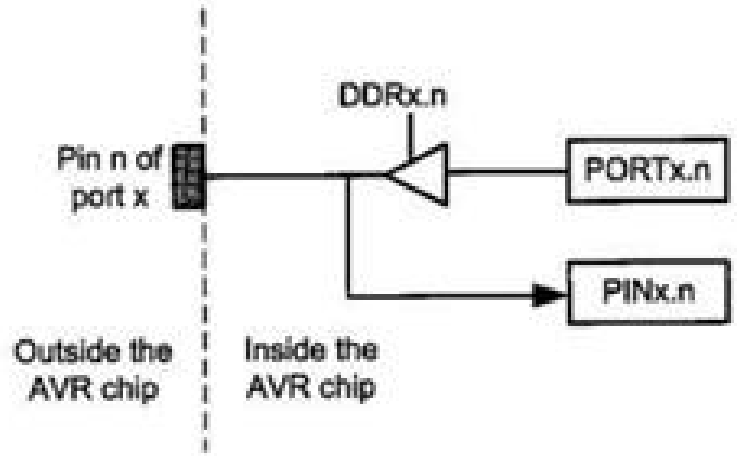
Parameter:

pin: the number of the pin whose mode you wish to set

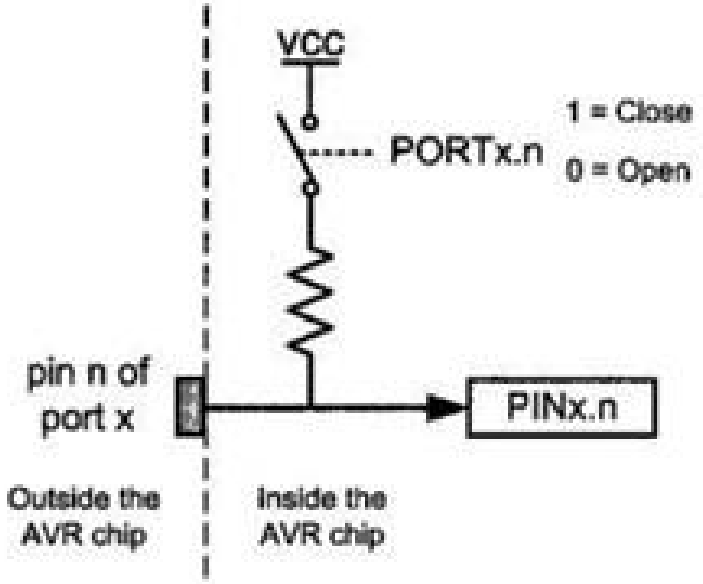
mode: INPUT, OUTPUT or INPUT_PULLUP.

Digital Input

INPUT_PULLUP



I/OPortin AVR microcontrollers



Pull-up Resistor

Digital Input

DIGITAL INPUT PROGRAMMING (ON-OFF)

Syntax:

digitalRead(pin)

Parameter:

pin: the number of the pin whose mode you wish to set

Return:

HIGH: when the logic is HIGH

LOW: when the logic is LOW

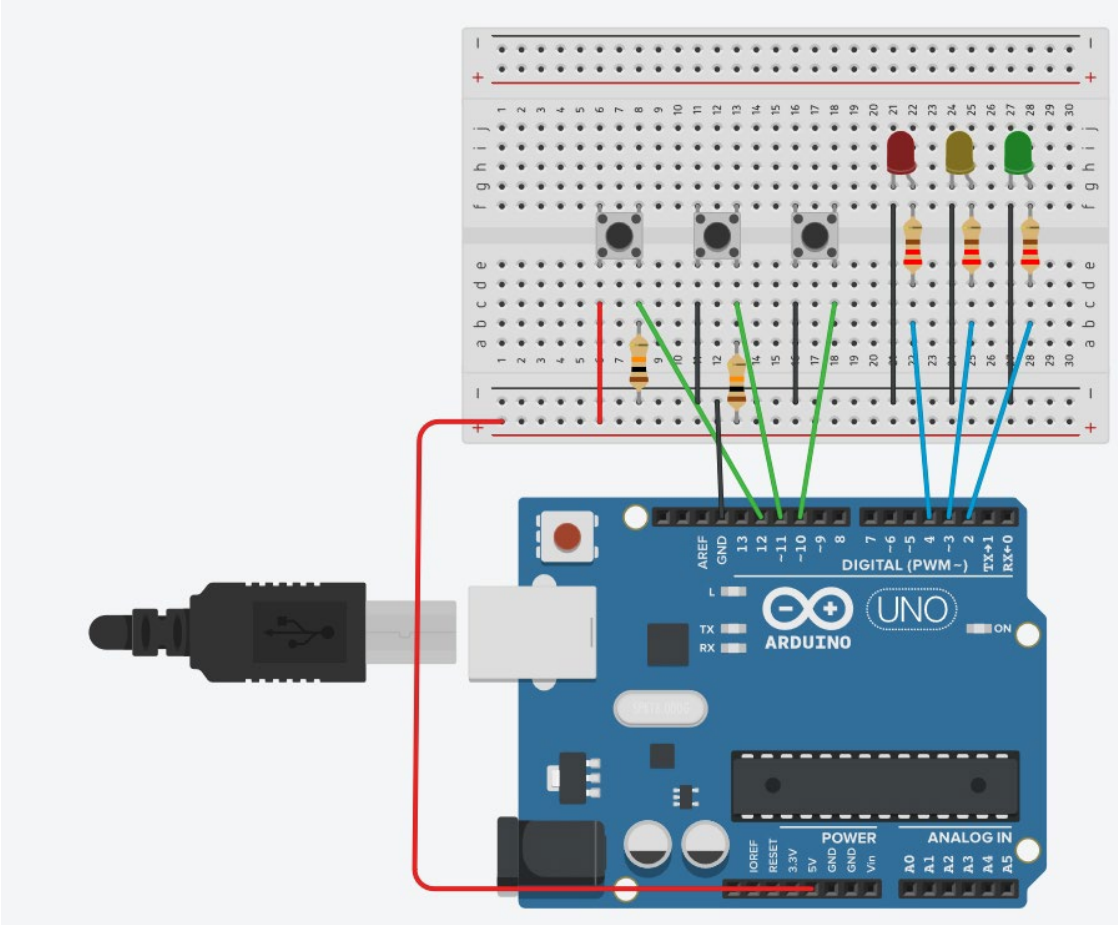
Digital Input

Example:

```
#define button 2        // switch input Active Low
#define pressed LOW

void setup()
{
  Serial.begin(9600);
  pinMode(button,INPUT_PULLUP);
}
void loop()
{

  int ReadSwitch = digitalRead(button);

  if(ReadSwitch == pressed)
  {
    Serial.println("Pressed Switch."); delay(500);
  }

}
```

Activity

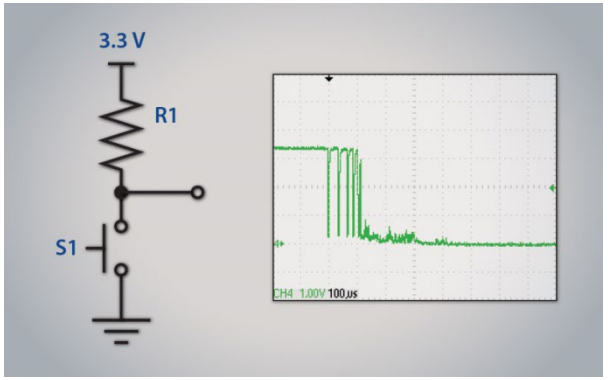
- ต่อวงจรโดยใช้ Switch
  - Pull Up
  - Pull Down
  - Internal Pullup
- กดปุ่มไหน ให้ไฟติด (ไม่ต้องกดค้าง)
- กดอีกทีให้ดับ

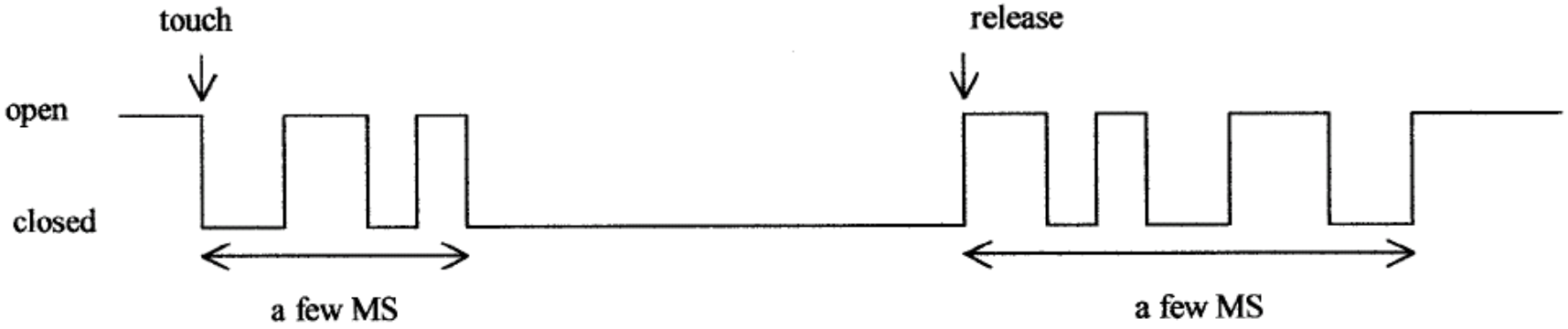


Bounce Problem

- ในการกดสวิตซ์ 1 ครั้ง จะมีช่วงเวลาสั้นๆ ที่เกิดสัญญาณคล้ายกับการกดสวิตซ์หลายครั้ง เนื่องจากหน้าสัมผัส

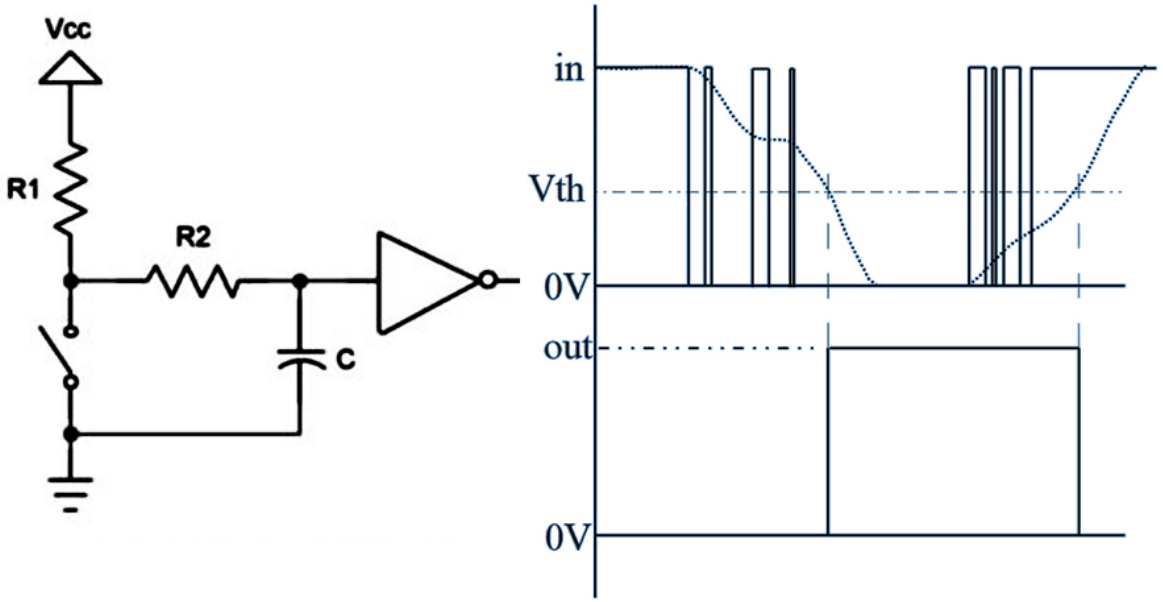
- การแก้ไข
  - สวิตซ์บางตัวที่มีราคาแพงจะไม่เกิด Bounce
  - แก้ไขโดยวิธีการทางฮาร์ดแวร์
  - แก้ไขโดยวิธีการทางซอฟต์แวร์





Bounce Problem

- Hardware De-bouncing
  - ใช้วงจรฮาร์ดแวร์ในการกำจัดการ Bounce
  - Bounce จาก Button/Switch bounce สามารถลดหรือกำจัดได้โดยใช้ตัวเก็บประจุ และใช้ Schmitt Trigger ในการสร้าง Logic Level



Bounce Problem

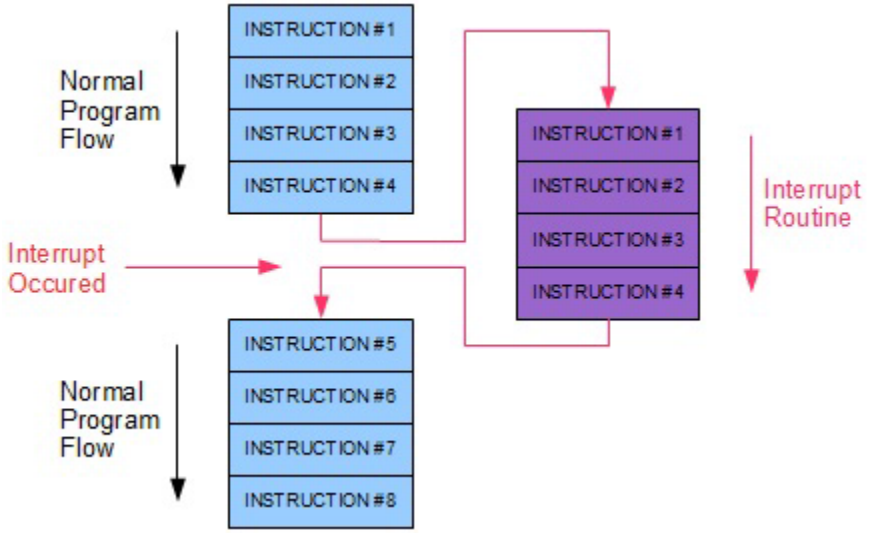
- Software De-bouncing
  - ลดผลของ Bounce โดยใช้โปรแกรม โดยไม่ต้องใช้วงจรฮาร์ดแวร์เพิ่มเติม
- Steps:
  - รอการกดคีย์
    - หน่วงเวลา 10 ms (หรือมากกว่า) เพื่อข้ามช่วงเวลาที่เกิดการ Bounce
  - รอการปล่อยคีย์
    - หน่วงเวลา 10 ms

Software Debounce

```
int buttonState;
int lastButtonState = LOW;
long lastDebounceTime = 0;
long debounceDelay = 50;

loop(){
  int reading = digitalRead(buttonPin);

  // If the switch changed, due to noise or pressing:
  if (reading != lastButtonState) {
    // reset the debouncing timer
    lastDebounceTime = millis();
  }

  if ((millis() - lastDebounceTime) > debounceDelay) {
    // whatever the reading is at, it's been there for longer
    // than the debounce delay, so take it as the actual current state:

    if (reading != buttonState) {
      buttonState = reading;
    }
  }
  lastButtonState = reading;
}
```

External Interrupt

- คือการขัดจังหวะการทำงานระหว่างที่กำลังทำงานบางอย่างอยู่
- เมื่อถูก Interrupt แล้ว จะต้องมาทำงานที่กำหนดไว้ (เรียกว่า Interrupt Service Routine หรือ ISR) เมื่อเสร็จแล้วจึงจะกลับไปทำงานเดิมต่อ
- ข้อดีของ Interrupt คือ ไม่ต้อง polling ข้อเสีย คือ debug ยาก



External Interrupt

```
#define button 2 // switch input Active Low
#define pressed LOW

void setup()
{
     Serial.begin(9600);
     pinMode(button,INPUT_PULLUP);
}
void loop()
{
     bool ReadSwitch = digitalRead(button);

     if(ReadSwitch == pressed)
      {
          Serial.println("Pressed Switch.");
          delay(500);
      }
}
```

External Interrupt

Syntax:

attachInterrupt(interrupt, ISR, mode)

Parameter:

interrupt : the number of the interrupt

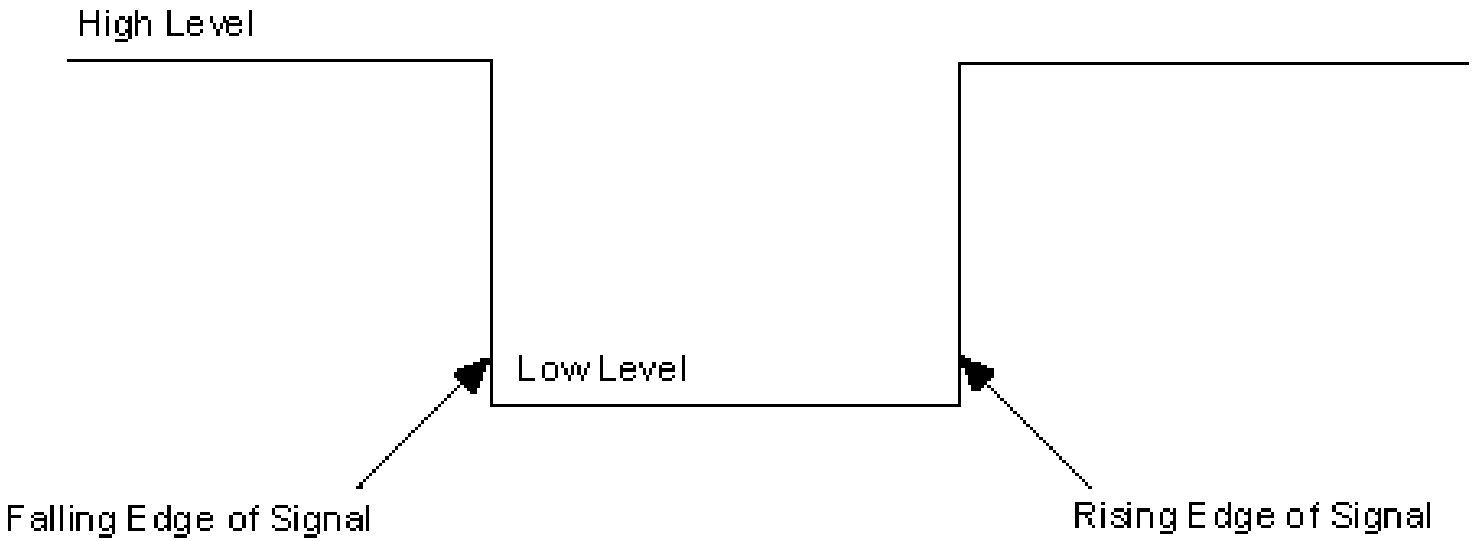
interrupt -> 0(pin2) , interrupt -> 1(pin3)

ISR: the ISR to call when the interrupt occurs; this function must take no parameters and return nothing. This function is sometimes referred to as an *interrupt service routine*.

mode: defines when the interrupt should be triggered.

- LOW to trigger the interrupt whenever the pin is low.
- CHANGE to trigger the interrupt whenever the pin changes value.
- RISING to trigger when the pin goes from low to high.
- FALLING for when the pin goes from high to low.

External Interrupt

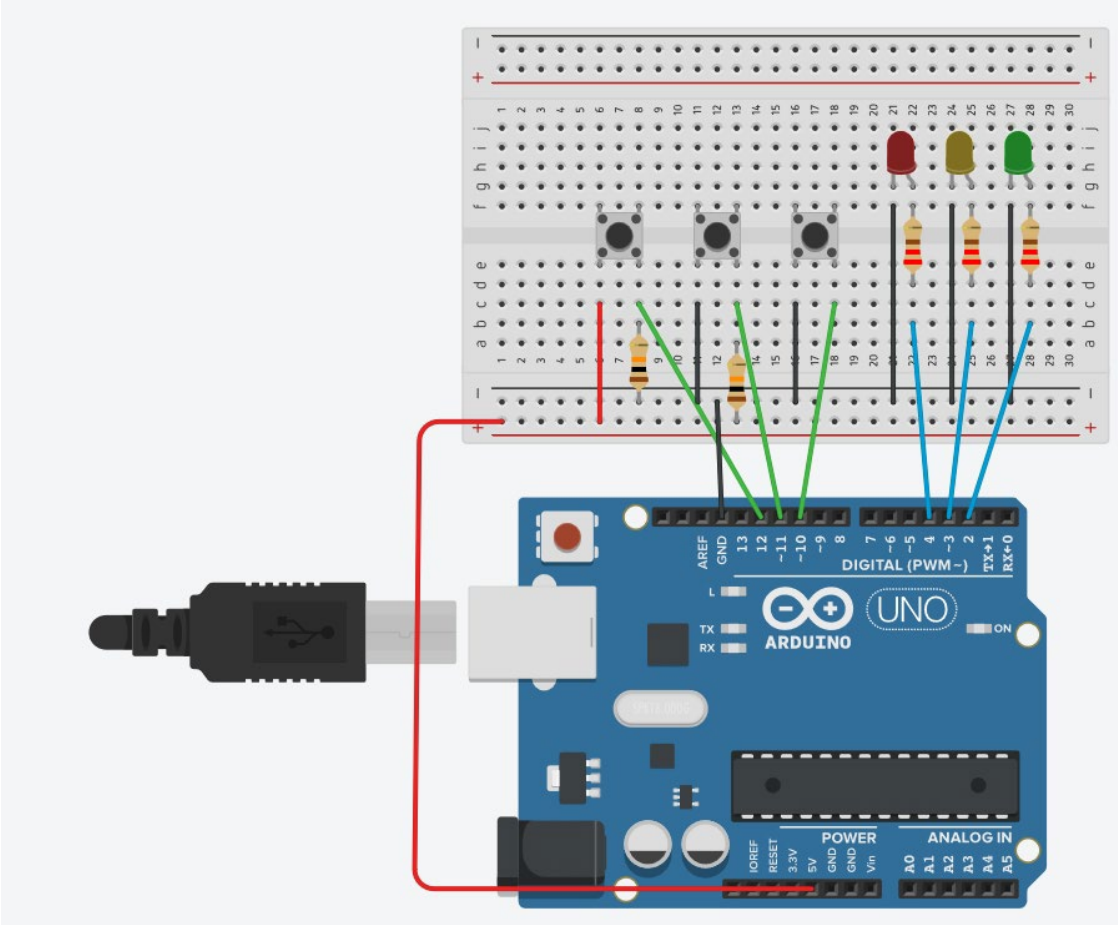


External Interrupt

```
#define button 2
#define ledPin 12

void setup()
{
    pinMode(button,INPUT_PULLUP);
    pinMode(ledPin, OUTPUT);
    attachInterrput(0, EXTI0_ISR, FALLING);
}
void loop()
{

}

void EXTI0_ISR()
{
    digitalWrite(ledPin, !digitalRead(ledPin); // Toggle LED
    delay(150);
}
```

Activity

- จากวงจรใน Slide หน้า 8 ให้ดัดแปลงวงจร และโปรแกรม โดยให้ใช้ Interrupt กับสวิตซ์ปุ่มที่ควบคุม LED สีเขียว

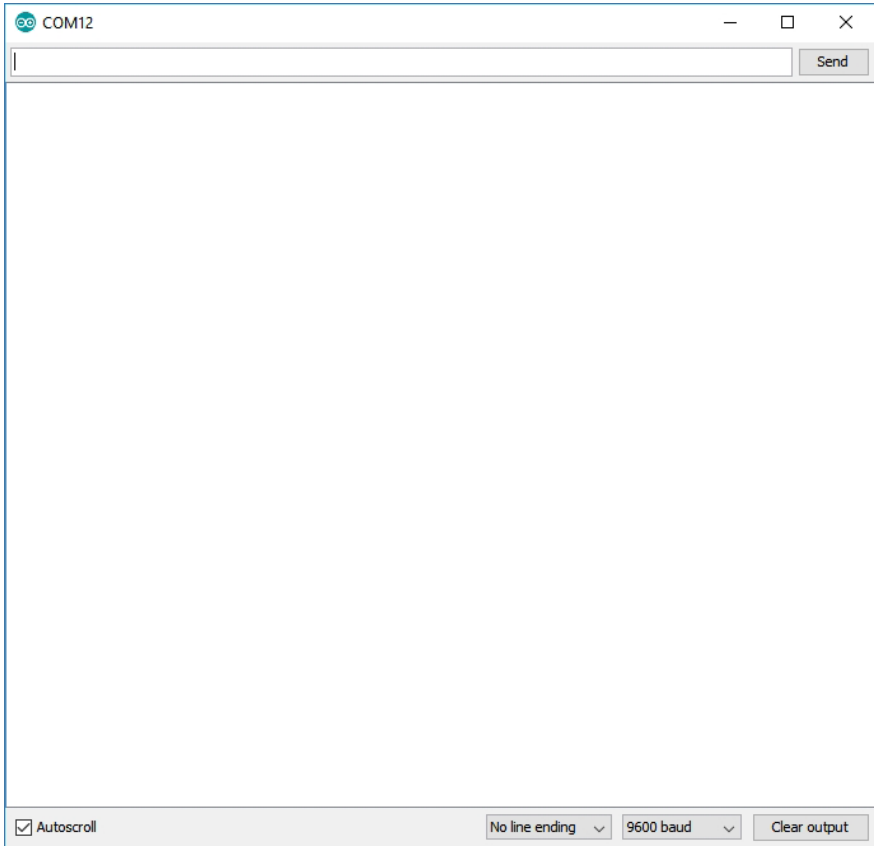


การติดต่อระหว่าง Arduino IDE กับบอร์ด

- ในบางครั้งเราต้องการติดต่อระหว่าง Arduino IDE กับ บอร์ด
  - กรณีที่ต้องการส่งค่าจากคีย์บอร์ดไปที่บอร์ด Arduino
  - กรณีที่ต้องการส่งค่าจากบอร์ด Arduino ไปแสดงผล เช่น กรณี debug โดยการแสดงตัวแปร
  - กรณีต้องการ plot กราฟข้อมูล

การติดต่อระหว่าง Arduino IDE กับบอร์ด

- ใน Arduino IDE จะมีหน้าต่าง Serial Monitor
- Tools -> Serial Monitor
- ช่องด้านบนสำหรับส่งข้อมูลจาก PC -> Arduino Board
- หน้าต่างด้านล่างสำหรับแสดงผลข้อมูลที่ส่งจาก Arduino Board
- Note : ต้องเลือก baud rate ให้ตรงกับโปรแกรม
- ข้อควรระวัง : หากเปิด Serial Monitor จะ Upload ไม่ได้



Serial Function - Begin

- ใช้สำหรับกำหนดว่ามีการใช้งาน Serial และกำหนดค่าความเร็ว (ต้องเท่ากัน)

Syntax:

Serial.begin(speed)

Parameter:

speed: in bits per second(baud)

300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, or 115200

Serial Function - Begin

- สั่งเริ่มต้นการใช้ Serial

Example:

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

Serial Function - Print

- สั่งให้บอร์ดส่งข้อมูลไปแสดงผลใน Serial Monitor

Syntax:

```
Serial.print(val)
Serial.print(val, format)
Serial.println(val)
Serial.println(val, format)
```

Parameter:

```
val: the value to print - any data type
format: specifies the number base
        or number of decimal places (floating point)
```

Serial Function - Print

Example:

- Prints data to the serial port as human-readable ASCII text

  Serial.print(78) gives "78"

  Serial.print(1.23456) gives "1.23"

  Serial.print('N') gives "N"

- An optional second parameter specifies the base (format) to use

  Serial.print(78, BIN) gives "1001110"

  Serial.print(78, OCT) gives "116"

  Serial.print(78, DEC) gives "78"

  Serial.print(78, HEX) gives "4E"

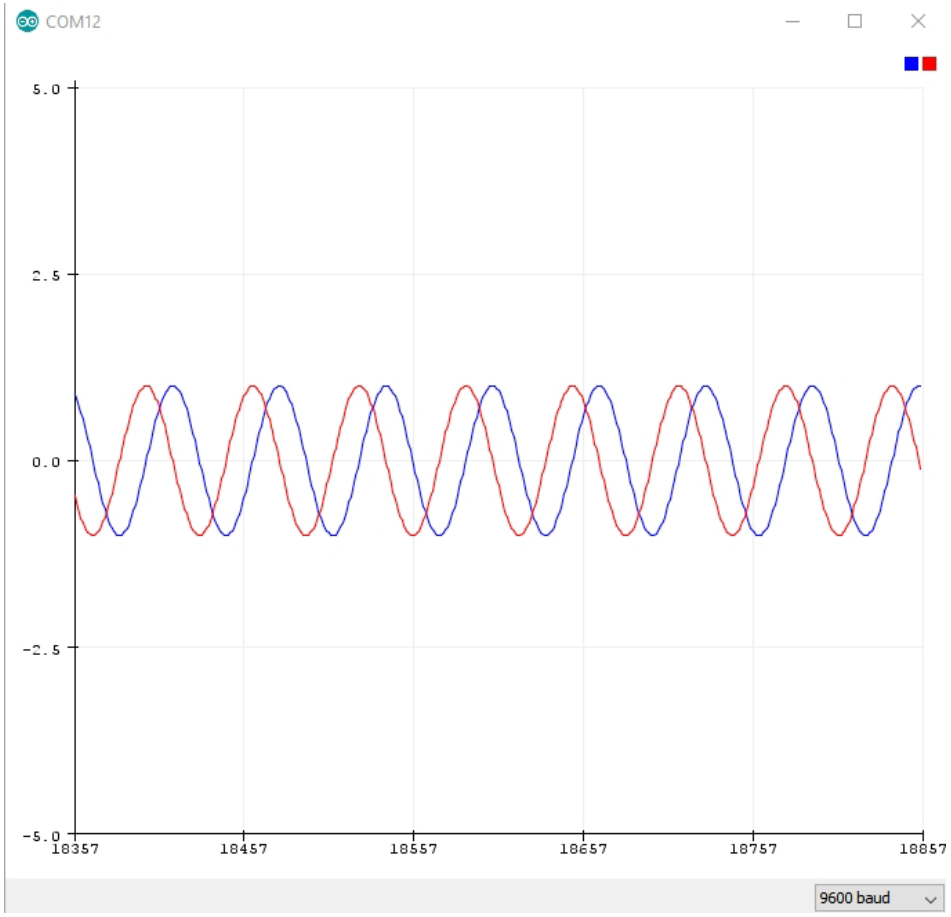
Activity

- ให้นักศึกษา นำ Switch ต่อกับบอร์ด Arduino ที่ขา 2 รันโปรแกรมและเปิด Serial Monitor ดู

```
#define button 2        // switch input Active Low
#define pressed LOW

void setup()
{
  Serial.begin(9600);
  pinMode(button,INPUT_PULLUP);
}
void loop()
{
   bool ReadSwitch = digitalRead(button);
   if(ReadSwitch == pressed)
   {
       Serial.println("Pressed Switch."); delay(500);
   }
}
```

Serial Plotter

- นอกจากจะแสดงเป็นข้อความแล้ว ยังสามารถแสดงเป็นกราฟได้อีกด้วย



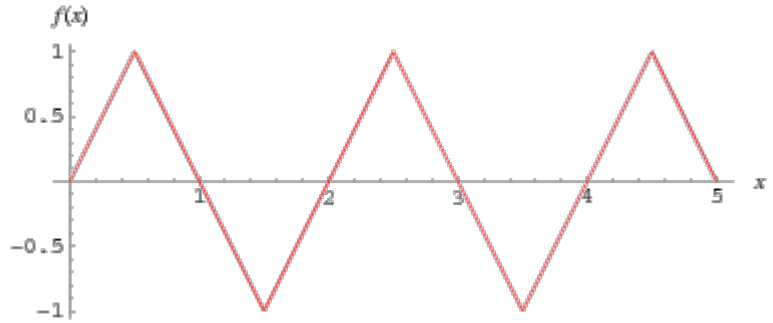
Activity

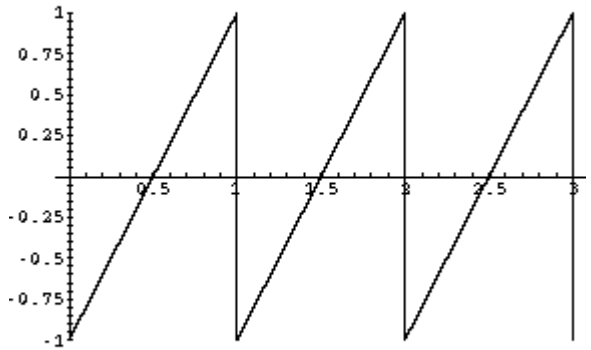
- ให้นำโปรแกรมต่อไปนี้ไปรัน และดูผลใน Serial Plotter

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

void loop()
{
  //Sine Wave & Cosine Wave
  float angle=0;
  for(angle=0;angle<=90;angle=angle+0.1)
  {
    float sina=sin(angle);
    float cosa=cos(angle);
    Serial.print(sina);
    Serial.print(" ");
    Serial.println(cosa);
    delay(1);
  }
}
```

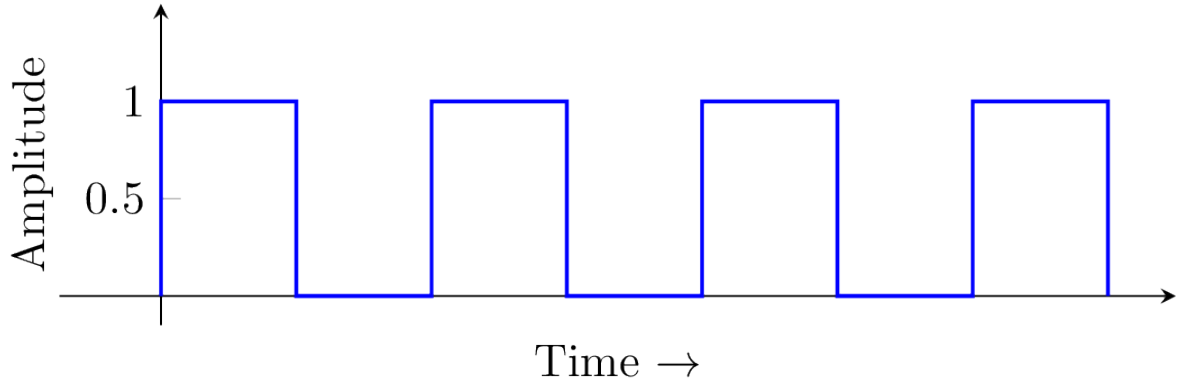
Activity

- ทดลองสร้างคลื่น Triangle, Saw tooth และ Square



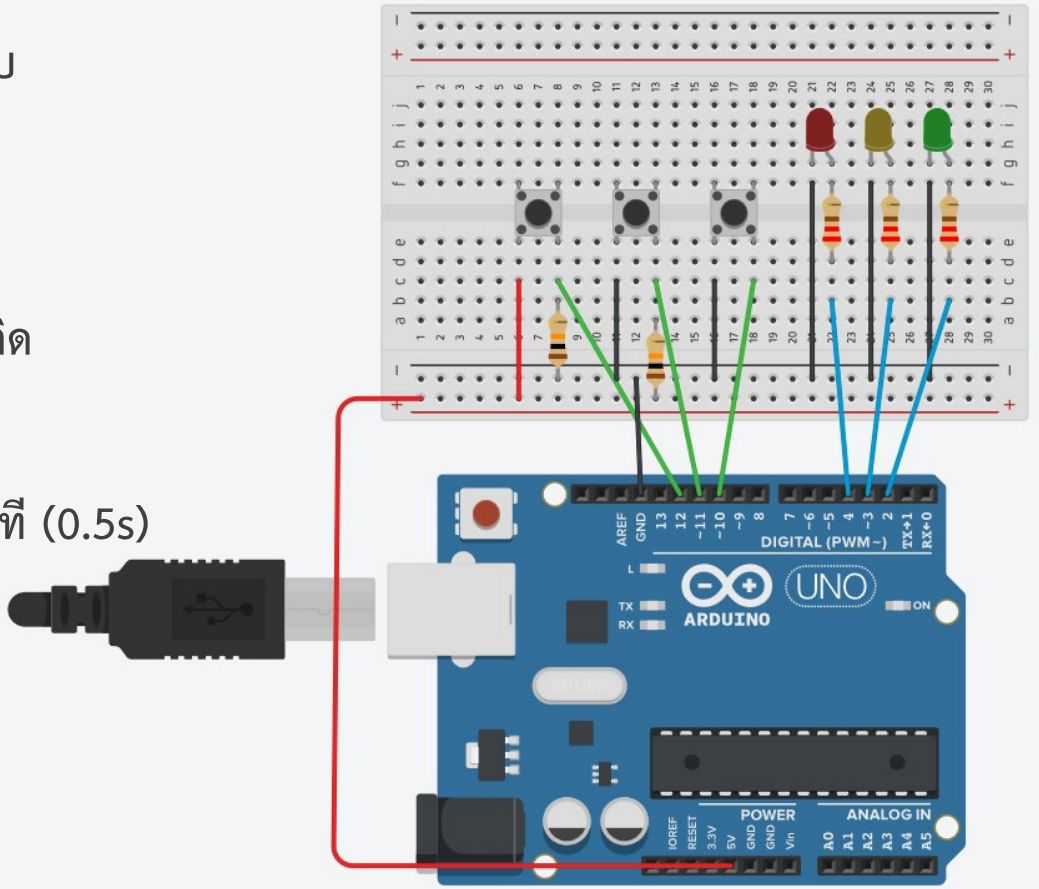


Square wave



Assignment #2

- เขียนโปรแกรม ให้แสดงดังนี้
  - กดปุ่มขวา เขียวติด 3 วินาทีแล้วดับ
  - กดปุ่มซ้าย แดงติด 3 วินาทีแล้วดับ
  - ถ้ากดปุ่มแดงหรือเขียวซ้ำ ให้ดับ
  - แม้จะเขียว แต่ถ้ากดซ้าย แดงต้องติด
  - ถ้าแดงอยู่ กดขวา ไม่มีผล
  - กดปุ่มกลาง เหลืองกระพริบ 2 วินาที (0.5s)
  - ปุ่มกลางจะมีผลเมื่อไฟอื่นไม่ติด (ปุ่มกลางมีความสำคัญน้อยที่สุด)



For your attention