



01076001

วิศวกรรมคอมพิวเตอร์เบื้องต้น

Introduction to Computer Engineering

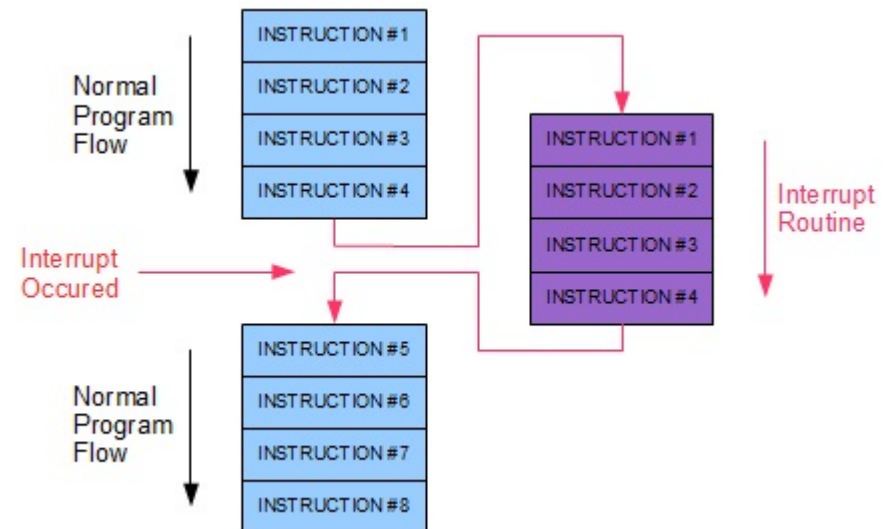
Arduino #2

Serial, Analog Read, LDR, 7 Segment



# 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

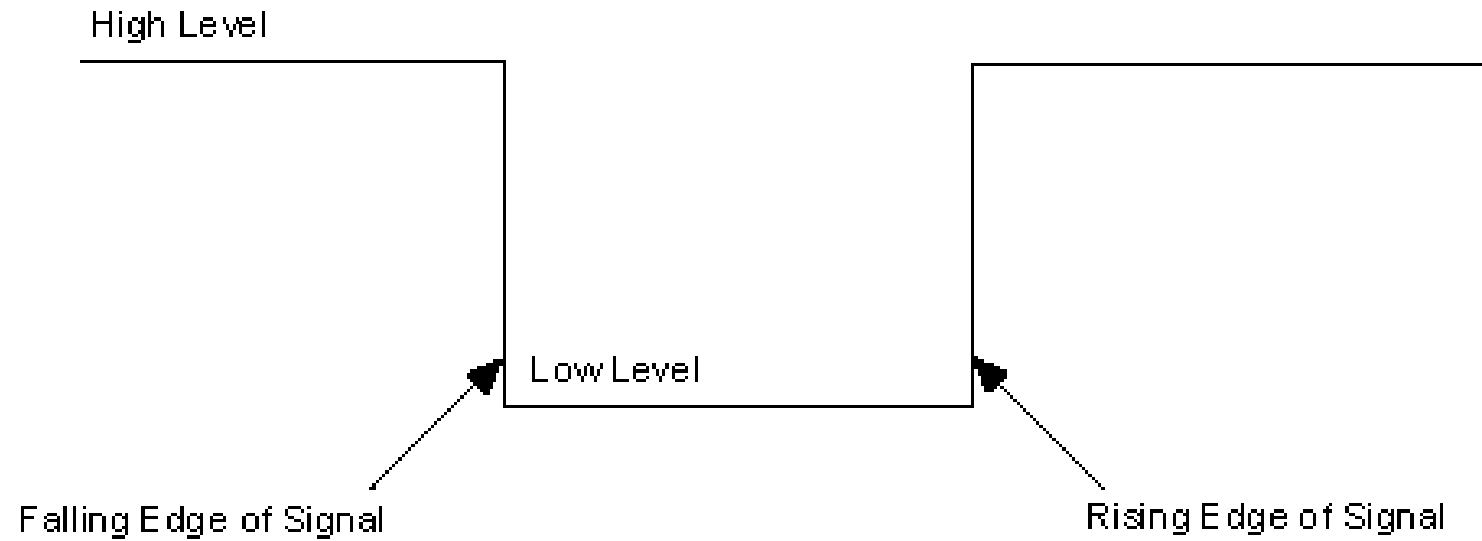
interrupt -> 0(pin2) , interrupt -> 1(pin3)

**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 13

void setup()
{
    pinMode(button, INPUT_PULLUP);
    pinMode(ledPin, OUTPUT);
    attachInterrupt(0, EXTIO_ISR, FALLING);
}

void loop()
{
}

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



# Exercise

- ให้นำโปรแกรมใน Assignment 4 มาเขียนโดยใช้ Interrupt
- ตอบคำถามว่า ในกรณีที่ให้เลือกใช้งาน นักศึกษาจะเลือกการเขียนโปรแกรมแบบใด เพราะอะไร



# การติดต่อระหว่าง Arduino IDE กับบอร์ด

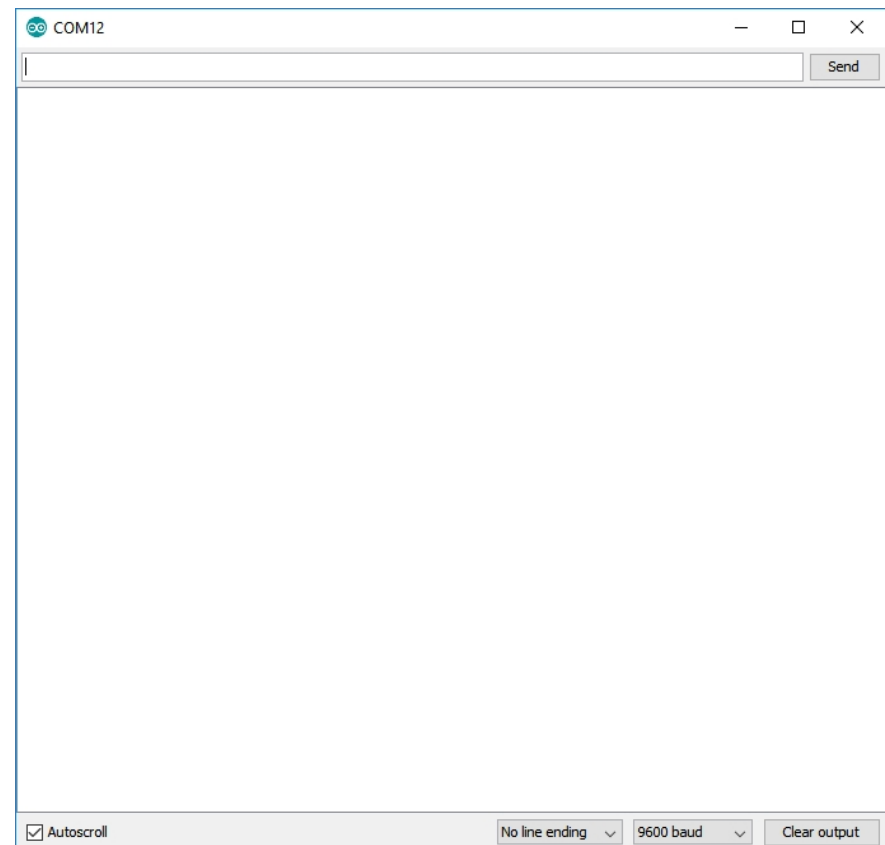
- ในบางครั้งเราต้องการติดต่อระหว่าง Arduino IDE กับ บอร์ด
  - กรณีที่ต้องการส่งค่าจากคีย์บอร์ดไปที่บอร์ด Arduino
  - กรณีที่ต้องการส่งค่าจากบอร์ด Arduino ไปแสดงผล เช่น กรณี debug โดยการแสดงตัวแปร
  - กรณีต้องการ plot กราฟข้อมูล





# 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

- เริ่มต้นการใช้ 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);
  }
}
```



# Analog Read

- นอกจาก Arduino จะสามารถอ่านค่าในแบบ Digital แล้ว ยังให้ขาสำหรับอ่านค่าแบบ Analog (ไม่ใช่แค่ 0,1) มาด้วย จำนวน 6 ขา คือ A0-A5 โดยค่าที่อ่านจะอยู่ระหว่าง 0-1023 โดย 0=0v และ 1023=5v

**Syntax:**

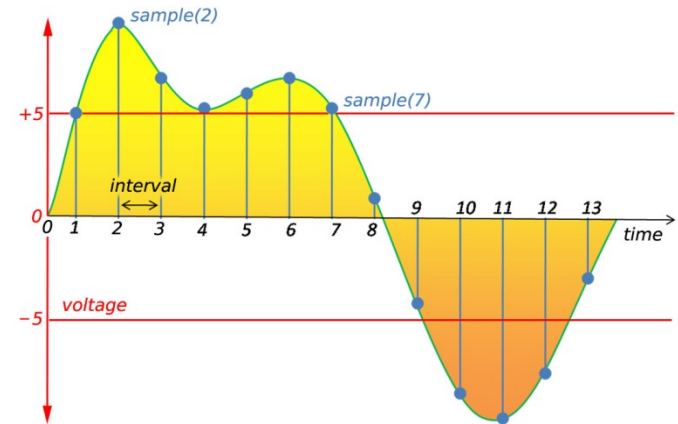
`analogRead(pin)`

**Parameter:**

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

**Return:**

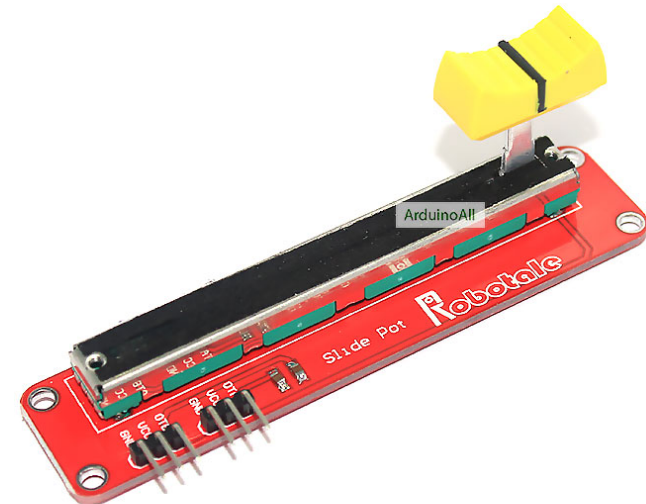
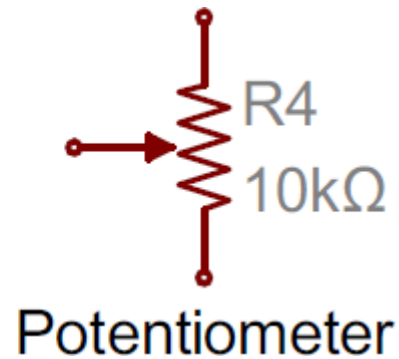
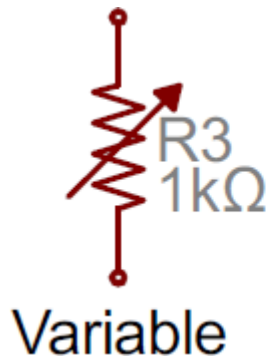
Integer : 0-1023 (0-5V)



# Potentiometer



- คือ ตัวต้านทานแบบปรับค่าได้

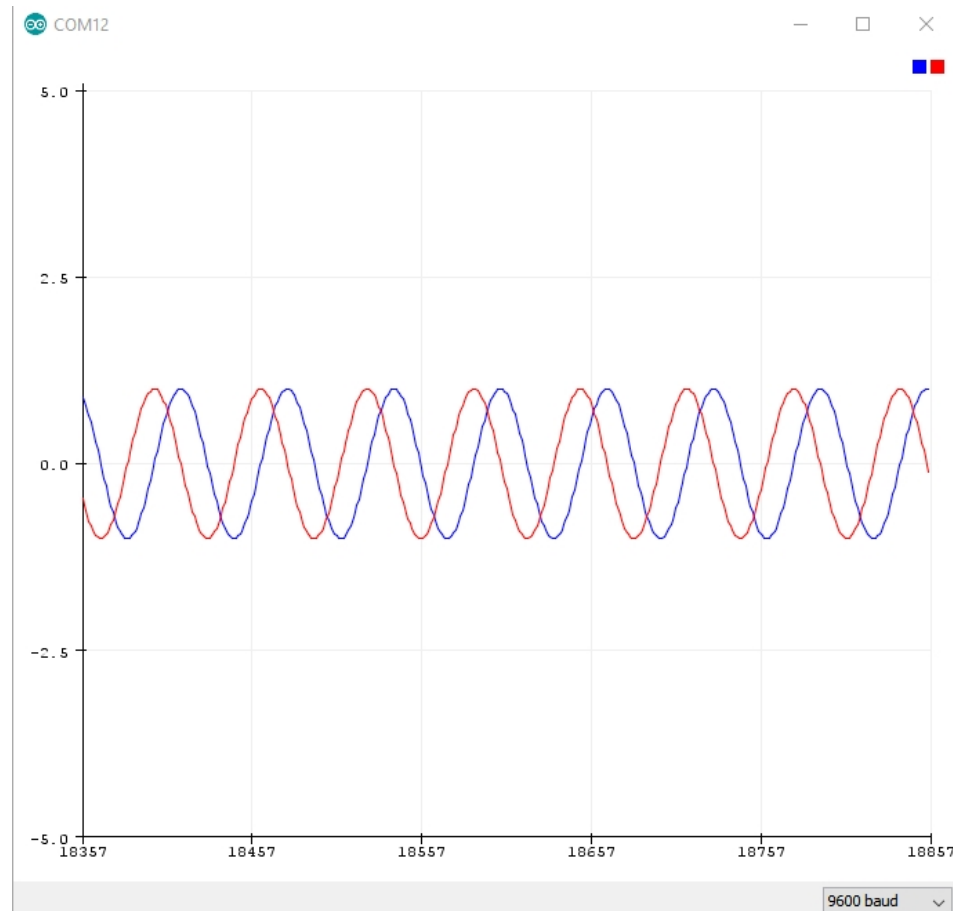




# Serial Plotter



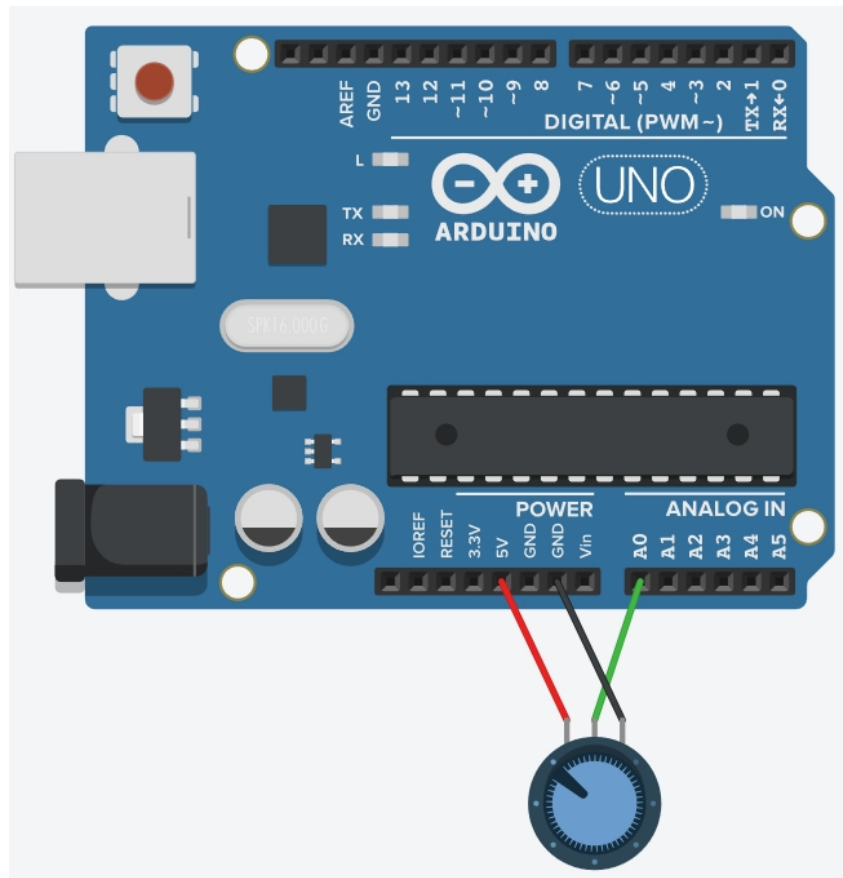
- นอกจากจะแสดงเป็นข้อความแล้ว ยังสามารถแสดงเป็นกราฟได้อีกด้วย



# Activity



- ให้ต่อวงจรตามรูป และเขียนโปรแกรมเพื่ออ่านค่ามาแสดงใน Serial Plotter





# Activity

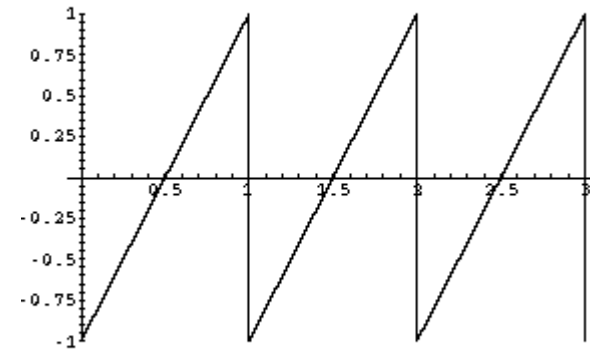
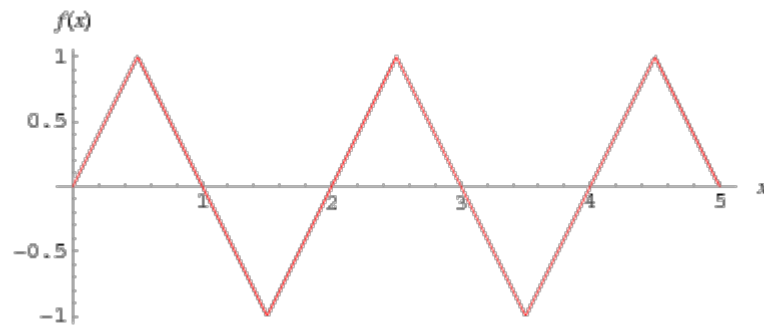
- ให้นำโปรแกรมต่อไปนี้ไปรัน และดูผลใน Serial Plotter

```
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);  
    }  
}
```

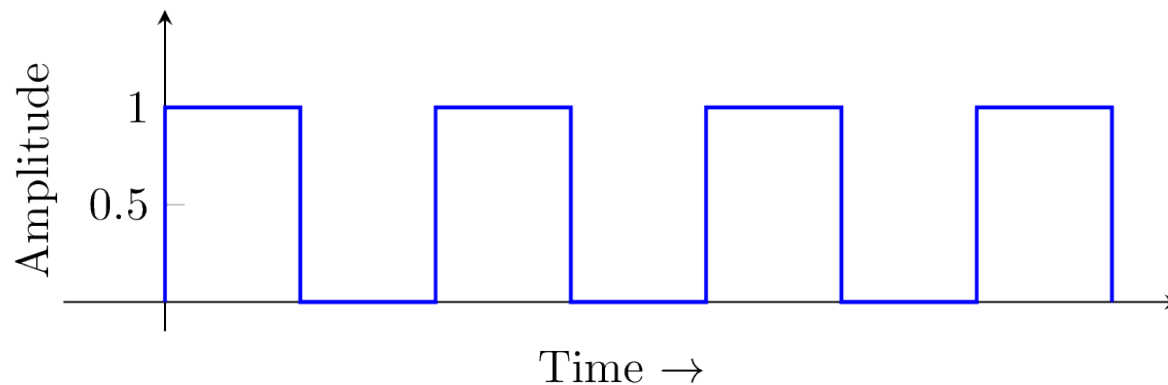
# Activity



- ทดลองสร้างคลื่น Triangle, Saw tooth และ Square



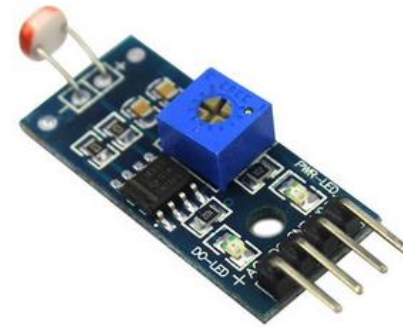
Square wave





# LDR module

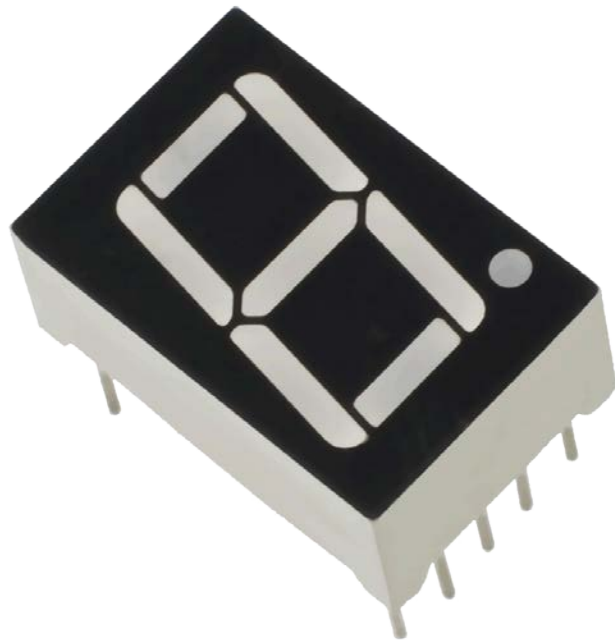
- เป็นโมดูลสำหรับใช้วัดความสว่างของแสง
- LDR ย่อมาจาก Light Detector Resister
- LDR จะเปลี่ยนค่าความต้านทานไปตามความสว่างของแสง
- โมดูลจะมี 4 ขา คือ
  - Vcc ต่อกับ 5V เพื่อเลี้ยงวงจร, Gnd ต่อกับ Ground ของ Arduino
  - AO (Analog Out) จะให้ Output เป็น Analog (0-1023)
  - DO (Digital Out) จะให้ Output เป็น **HIGH** เมื่อความสว่างมากกว่าที่กำหนด (สามารถกำหนดโดย R ปรับค่าได้ (สไลด์))



# Activity

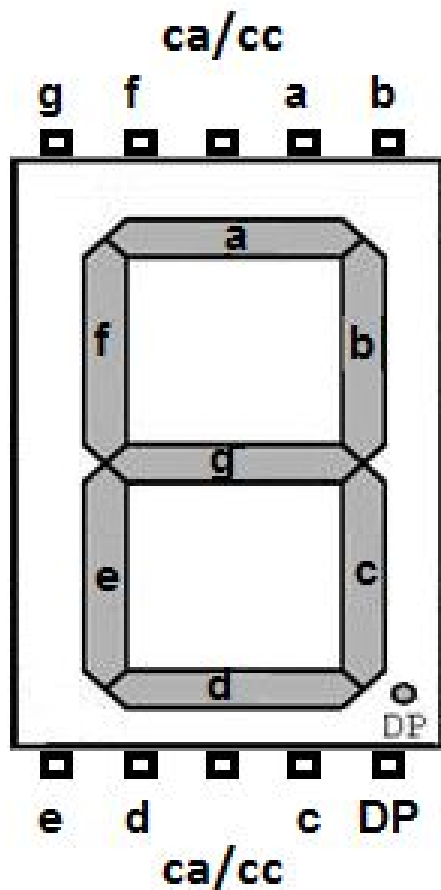


- ให้นำ LDR ต่อกับ Arduino
- ให้ต่อขาทั้ง Digital และ Analog
- เขียนโปรแกรมแสดงค่าที่อ่านได้ ทั้งสองขา
- เมื่อปรับ Trimpot (สีฟ้า) แล้วเกิดอะไรขึ้น

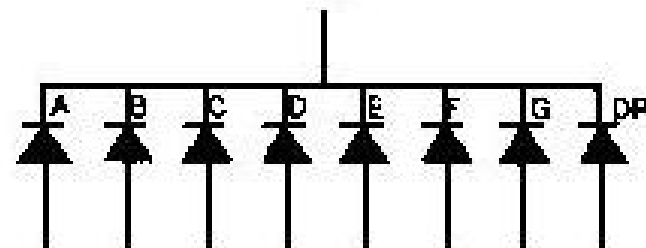


# 7-Segments

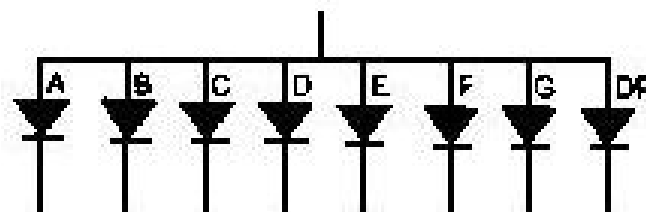
# รายละเอียดแต่ละ Segment ใน 7 Segment



Common Cathode

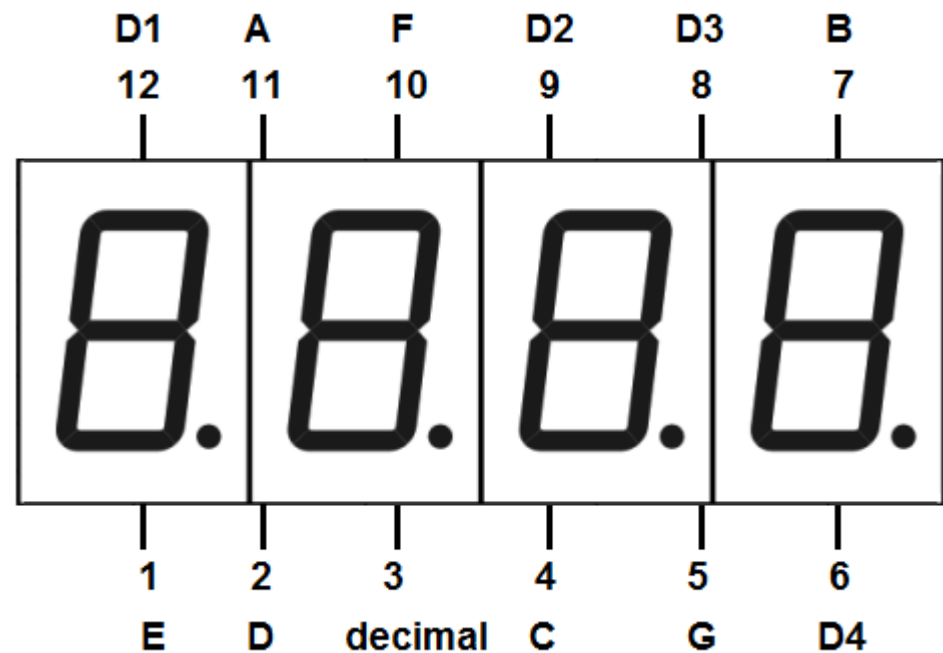


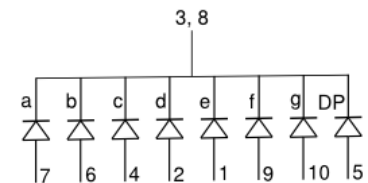
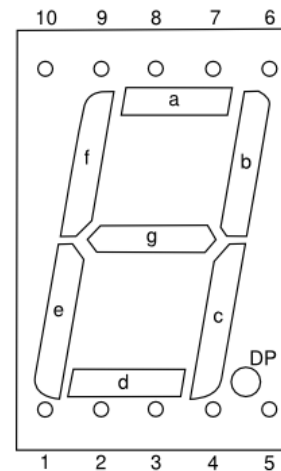
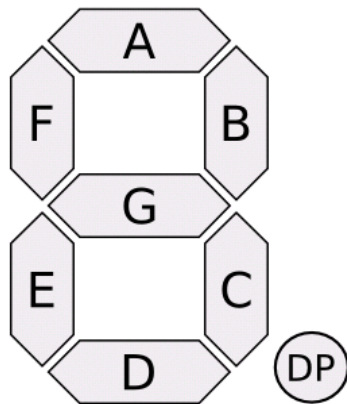
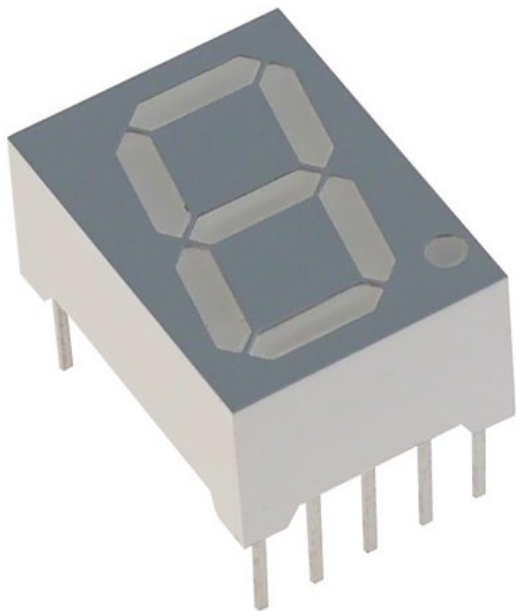
Common Anode





# LED 7 Segment ชนิดหลายหลัก



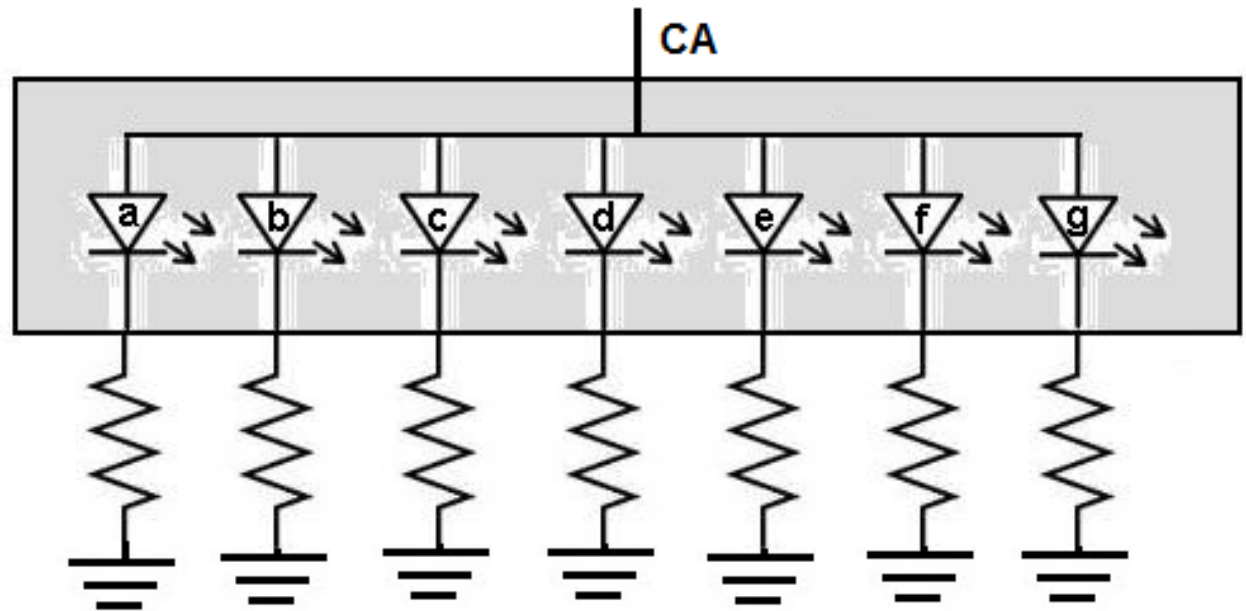


อยากให้ 7-Segments แสดงเลข 3 ค่า a-g จะต้องมามีค่าเป็น  
เท่าใดตามลำดับ

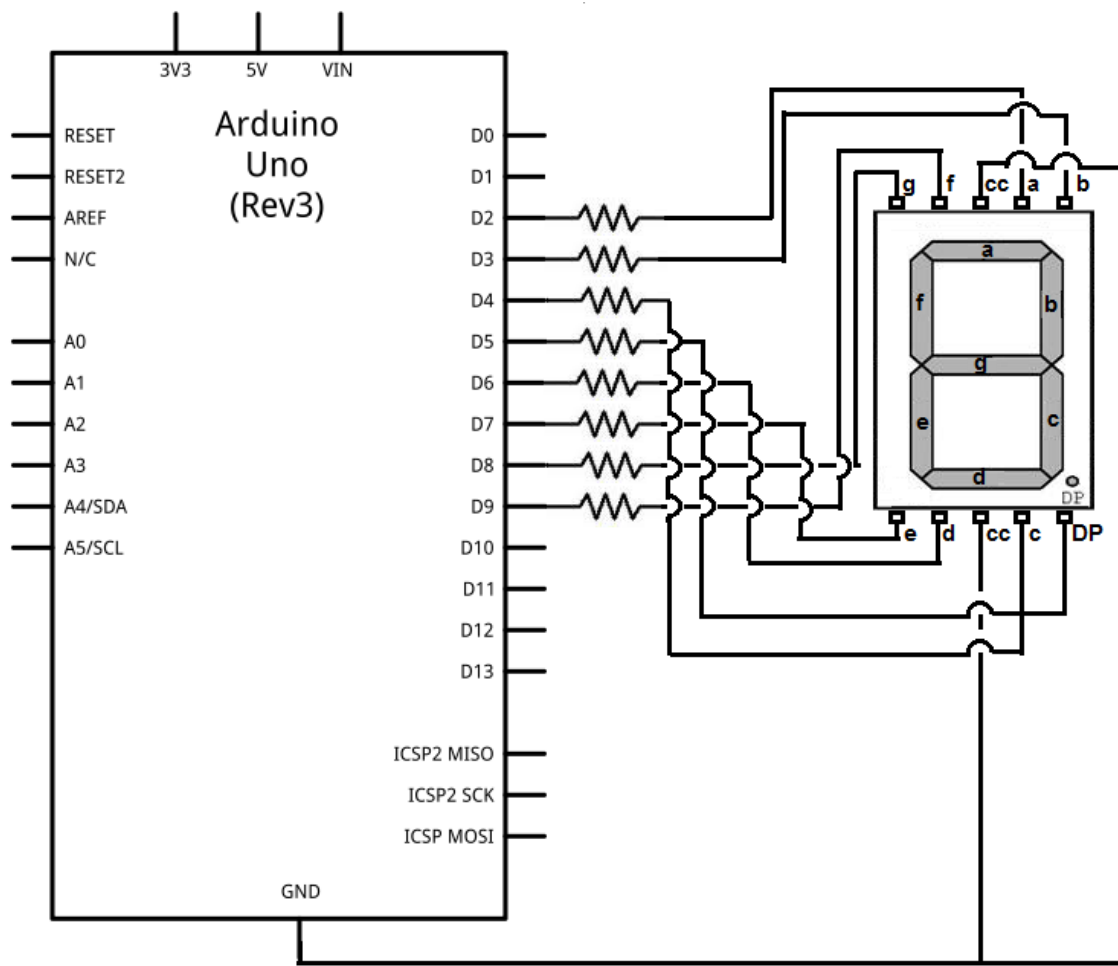
# การคำนวณตัวต้านทานสำหรับ 7-Segments



- Input voltage จากวงจร 5 Volts
- 7-Segment รับกระแส 15mA และ forward voltage drop ที่ 2 Volts
- ต้องใช้ R เท่าไหร่ - สีอะไร

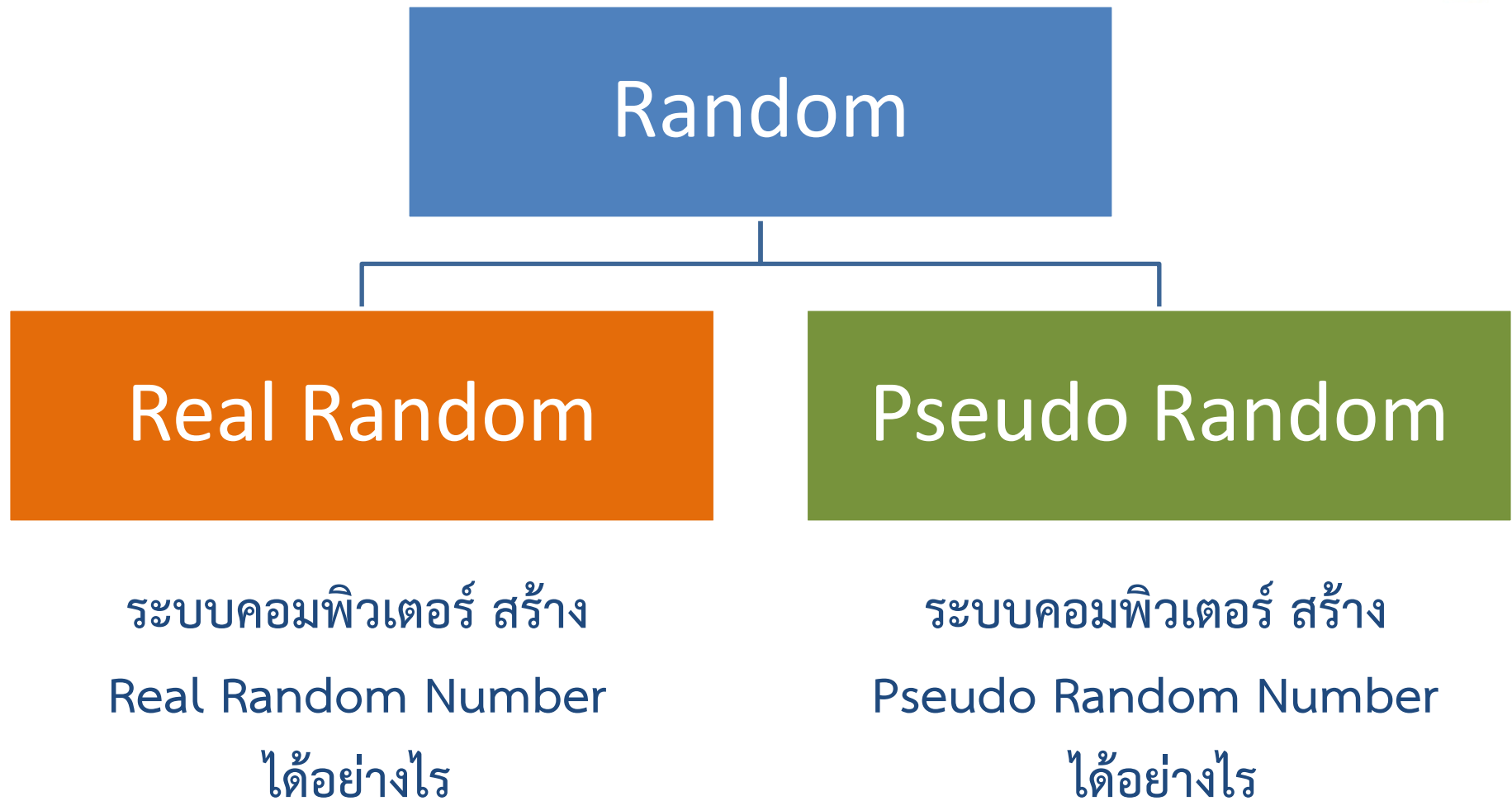


# ตัวอย่างการต่อ 7 Segment กับ Arduino





ตัวเลขสุ่ม (Random Number) มีลักษณะอย่างไร





# Random

## Real Random

วัดค่าจากแหล่งภายนอก  
เช่น Key Stroke , Voltage ที่มีการ  
สุ่มค่าจริงๆ

## Pseudo Random

Pseudo Random Number  
Generator Algorithm โดยใช้  
สูตรคณิตศาสตร์ มาสร้างตาราง  
ที่มีชุดตัวเลขที่เดาค่าได้ยาก



# random()

- Generate pseudo-random number
- Syntax
  - random(max)
  - random(min,max)
- Return
  - A random number between min and max-1 (long)





# randomseed()

- Initial random number generator
- Start point of random sequences
- Parameter
  - Long int : parameter to generate the seed



# bitRead()

## Description

Reads a bit of a number.

## Syntax

```
bitRead(x, n)
```

## Parameters

**x**: the number from which to read

**n**: which bit to read, starting at 0 for the least-significant (rightmost) bit



## การเลือกบิตในไบต์มาใช้งาน

- โดยใช้ `bitRead()` เช่น `bitRead(5,2)` จะ return ค่า บิตที่ 2 นับจากขวาสุดของ `00000101` นั่นก็คือ 1
- โดยใช้การ shift bit เช่น `((5) >> (2)) & 0x01`
  - หมายถึง การนำเลข 5 มาเลื่อนไปทางขวา 2 ครั้ง (`101 -> 10 -> 1`)
  - จากนั้นนำไปทำ Logical AND กับ `0x01` (`00000001b`)
  - ถ้าบิตที่ 2 (3 นับจากขวา) มีค่าเป็น 1 จะได้ผลเป็น 1
  - ถ้าบิตที่ 2 (3 นับจากขวา) มีค่าเป็น 0 จะได้ผลเป็น 0
- ใน arduino หากต้องการกำหนดข้อมูลเป็นฐาน 2 ให้ใช้ `B00000101`



# Assignment #2

- ต่อวงจรโดยใช้สวิตช์ 2 ตัว A และ B และต่อ 7 Segment จำนวน 1 ตัว
- ให้สร้าง Dice Game โดยเมื่อกดสวิตช์ A ให้แสดงผลใน 7 Segment เพิ่มครั้งละ 1 ถ้าเกิน 6 ให้กลับมาเริ่มที่ 1 ใหม่
- ถ้ากดสวิตช์ B ให้ทำการสุ่มและแสดงผล แล้วถ้าตรงกันให้แสดง ?? ที่แสดงว่าชนะ ถ้าไม่ตรงกันให้แสดง ?? ที่แสดงว่าแพ้
- ให้กำหนดตัวเลขใน Array และการแสดงผลให้ทำเป็นฟังก์ชันเดียว ห้ามทำเป็นฟังก์ชัน แสดงเลข 0,1,2,3,4,5,6,7,8,9 แยกกันไป ห้ามทำเป็น if หรือ case
- คะแนน 2 คะแนน
- มีคะแนนความคิดสร้างสรรค์ เพิ่มต่างหาก 0.5 คะแนน



# Dice Game

```
void setup() {  
    //setup pin mode and randomseed  
}  
  
loop(){  
    handle_guess_button();  
    handle_start_button();  
}  
  
void handle_guess_button() {  
    //increment guess number in pressed.  
}  
  
void handle_start_button() {  
    // if start pressed :  
    //     random and show  
    //     if guess == random : hooray() else boo()  
}
```



*For your attention*