



01076001

วิศวกรรมคอมพิวเตอร์เบื้องต้น

Introduction to Computer Engineering

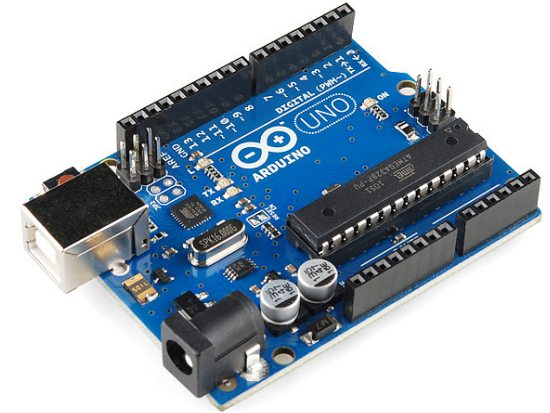
Arduino #1

LED, Digital Output, Digital Input, Switch

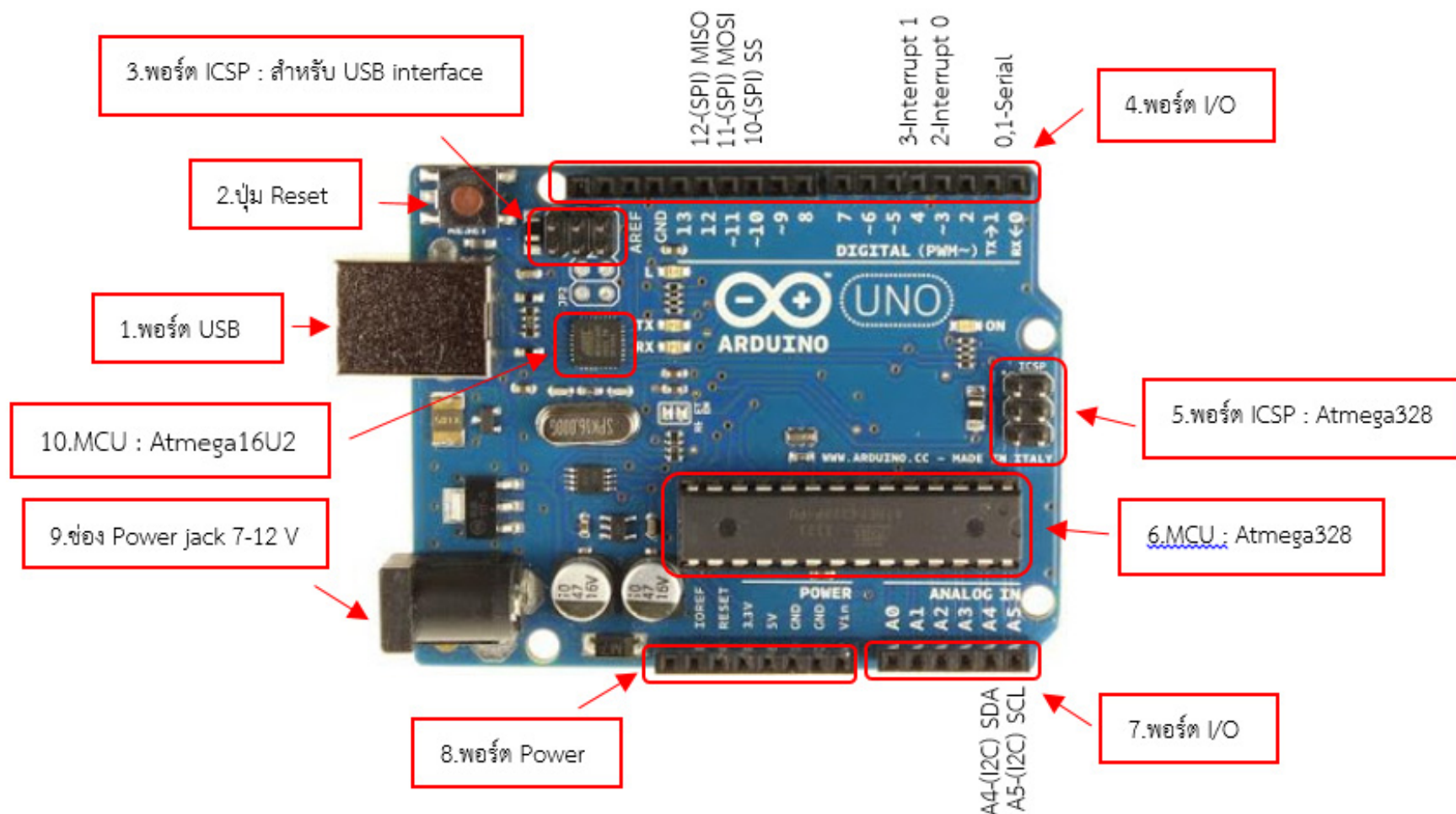
บอร์ด Arduino



- อ่านว่า (อา-ดู-อี-โน้ หรือ อาดูยโน้)
- เป็นบอร์ด ไมโครคอนโทรลเลอร์
- มีการใช้งานที่ง่าย
- มี Library มาก
- มีอุปกรณ์ที่นำไปเชื่อมต่อมาก
- ทำให้ได้รับความนิยมอย่างมาก



องค์ประกอบของ Arduino





Arduino Technical Specs

Technical specs

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Arduino Board



ติดตั้ง Arduino IDE



- ดาวน์โหลด Arduino จากเว็บ

The screenshot shows the Arduino website homepage. The top navigation bar is teal with the Arduino logo on the left and links for HOME, BUY, SOFTWARE (highlighted with a red circle), PRODUCTS, LEARNING, FORUM, SUPPORT, and BLOG on the right. Below the navigation bar, there are three main sections: 1. 'WHAT IS ARDUINO?' featuring an image of an Arduino Uno board and buttons for 'BUY AN ARDUINO', 'LEARN ARDUINO', and 'DONATE'. 2. 'BLOG' featuring an image of an RGB LED color detector circuit and the text 'BUILD A SIMPLE RGB LED COLOR DETECTOR WITH ARDUINO'. 3. 'ARDUINO EDUCATION' featuring the Arduino logo and the text 'REDEFINING THE LEARNING EXPERIENCE ONE CLASSROOM AT A TIME'.

ติดตั้ง Arduino IDE



Download the Arduino IDE



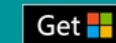
ARDUINO 1.8.5

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10



Mac OS X 10.7 Lion or newer

Linux 32 bits

Linux 64 bits

Linux ARM

[Release Notes](#)

[Source Code](#)

[Checksums \(sha512\)](#)

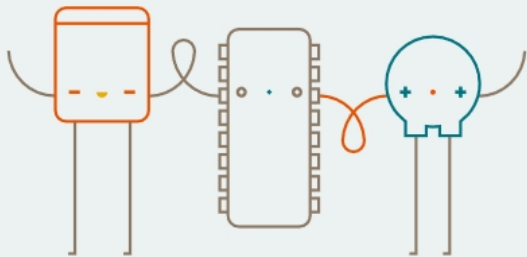
ติดตั้ง Arduino IDE



- เลือก **JUST DOWNLOAD** และ ติดตั้งลงในเครื่อง

Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **17,842,652** TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

\$3

\$5

\$10

\$25

\$50

OTHER

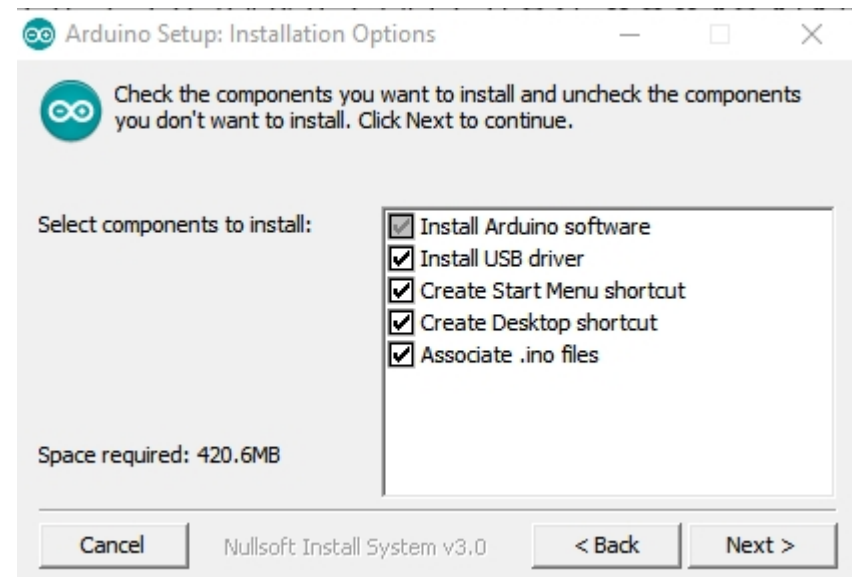
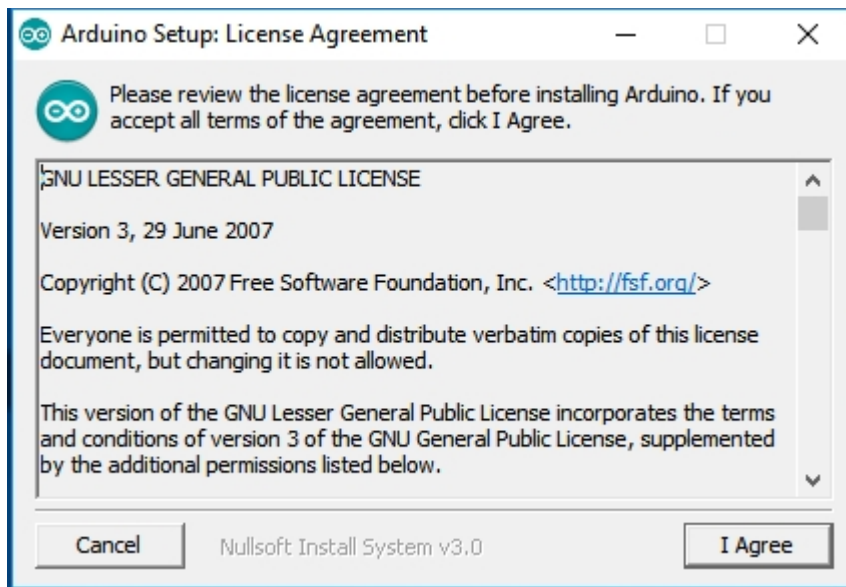
JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

ติดตั้ง Arduino IDE



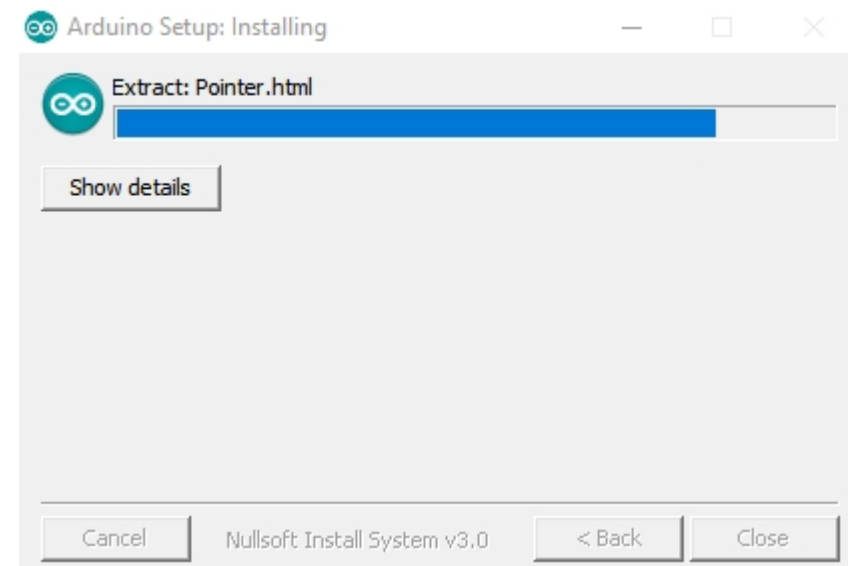
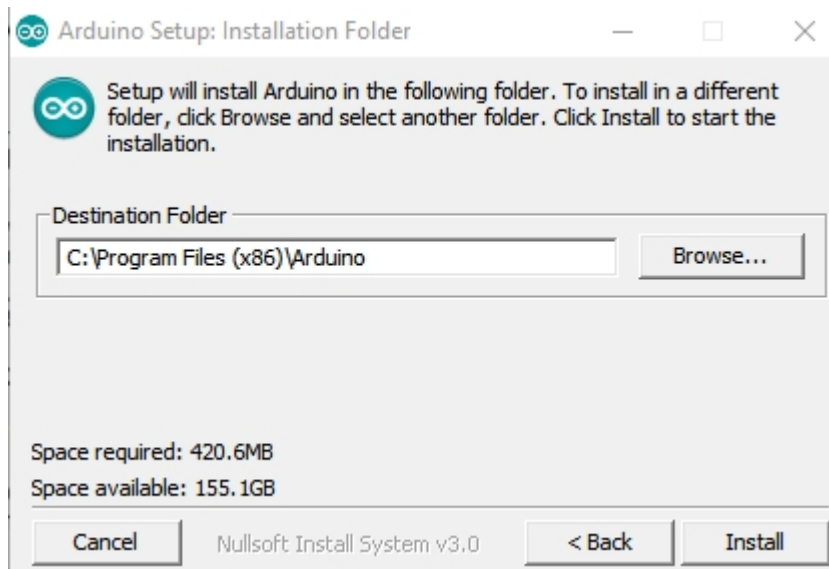
- คลิก I Agree และ Next ตามลำดับ



ติดตั้ง Arduino IDE



- คลิก Install



ติดตั้ง Arduino IDE



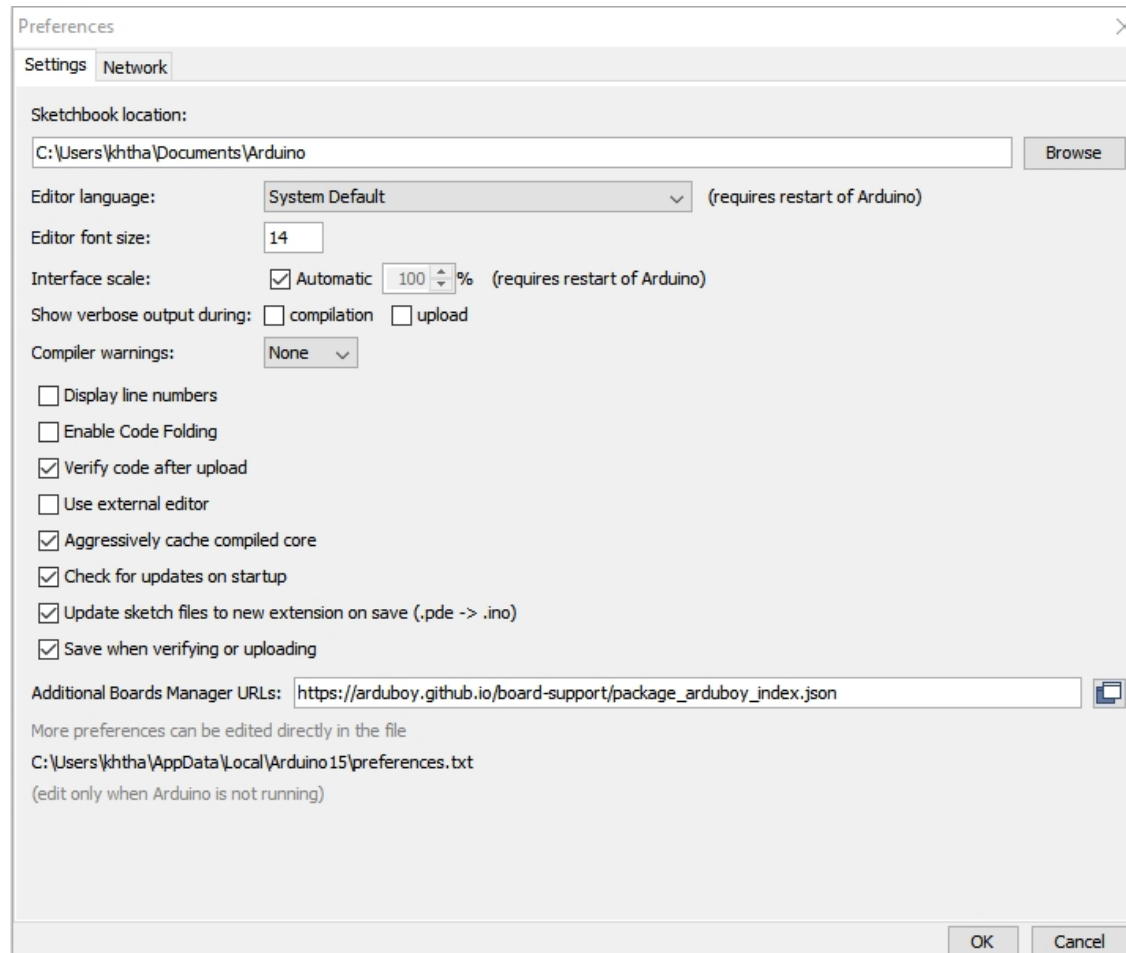
- เรียกขึ้นมาทำงาน



ตั้งค่า preferences



- คลิกที่ File -> Preferences



ตั้งค่า preferences

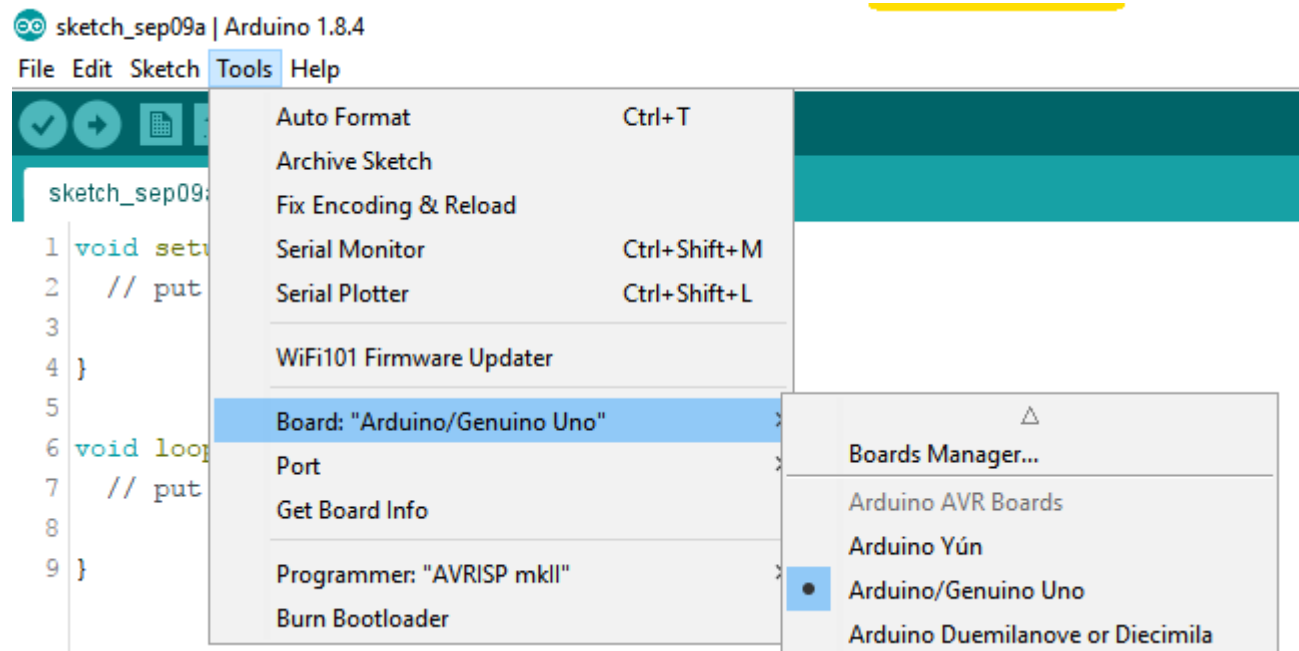


- กำหนดจุดที่ใช้เก็บโปรแกรม
- ขนาดฟอนต์
- Show verbose output during จะให้แสดงรายละเอียดระหว่างที่ Complier หรือ Upload หรือไม่
- แสดง Warning หรือไม่
- แสดงเลขที่บรรทัด
- Check for updates on startup
- Save when verifying or uploading



กำหนด Board และ Port

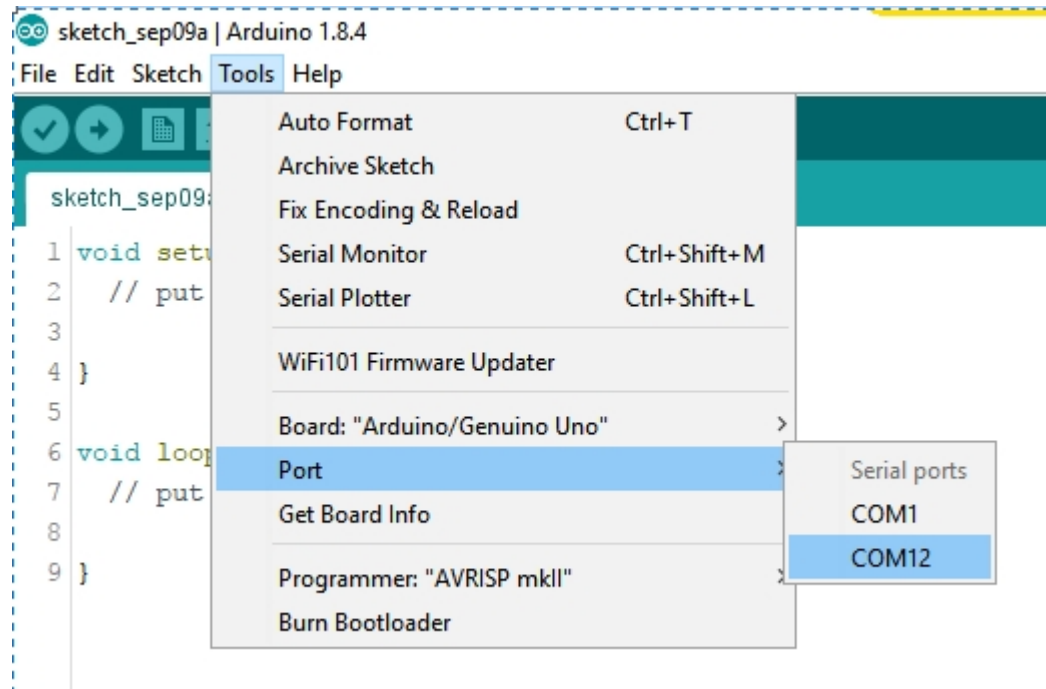
- เสียบ USB ระหว่างเครื่องคอมพิวเตอร์กับบอร์ด
- ไปที่ Tools -> Board เลือก Arduino/Genuino Uno





กำหนด Board และ Port

- เสียบ USB ระหว่างเครื่องคอมพิวเตอร์กับบอร์ด
- เลือกพอร์ต (ขึ้นอยู่กับเครื่อง)



<https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>



Arduino Sketch

```
void setup()  
{  
    // put your setup code here, to run once:  
  
}  
  
void loop()  
{  
    // put your main code here, to run repeatedly:  
  
}
```

Activity

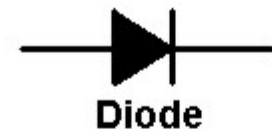
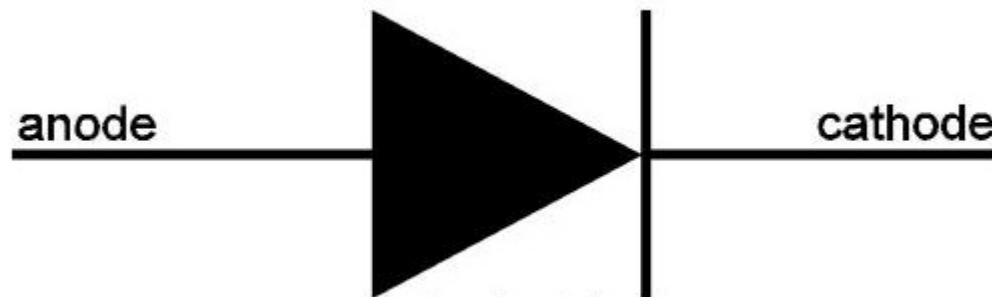


- เลือก Files -> Examples -> Basics -> Blink
- จากนั้น Upload ขึ้นบอร์ด
- หากไฟบนบอร์ดกระพริบ แสดงว่า การติดตั้งสำเร็จ

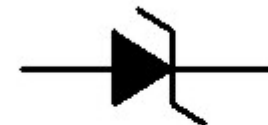
ไดโอด



- อุปกรณ์ที่ใช้ในการควบคุมทิศทางการไหลของกระแสไฟฟ้า คือ ไดโอด จะยอมให้กระแสไฟฟ้าไหลผ่าน ในทิศทางเดียว (ไบอัสตรง) และกั้นการไหลในทิศทางตรงกันข้าม (ไบอัสย้อนกลับ)



Diode



Zener Diode



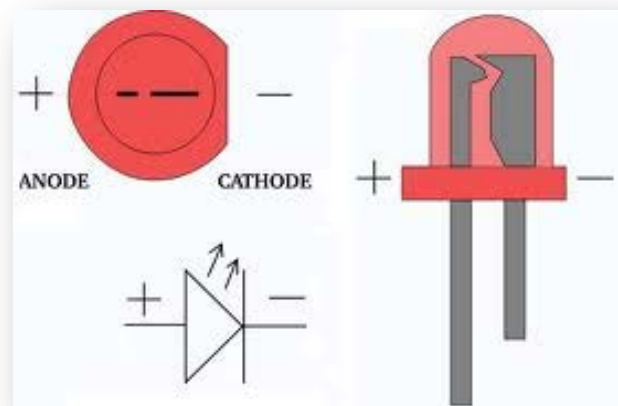
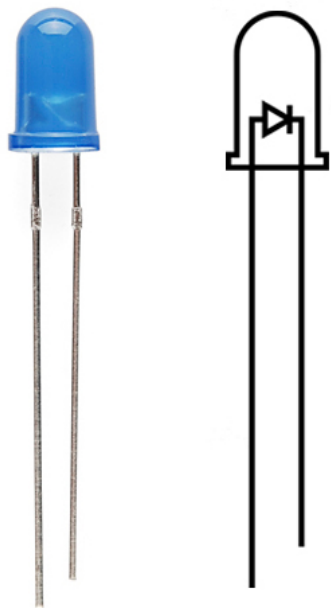
Light Emitting Diode - LED

ไดโอดเปล่งแสง แอลอีดี (LIGHT EMITTING DIODES)



- ไดโอดเปล่งแสง (Light Emitting Diode) มัก เรียกว่า แอลอีดี (LED) เป็นไดโอดที่ออกแบบมาเป็นพิเศษโดยเมื่อได้รับแรงดัน

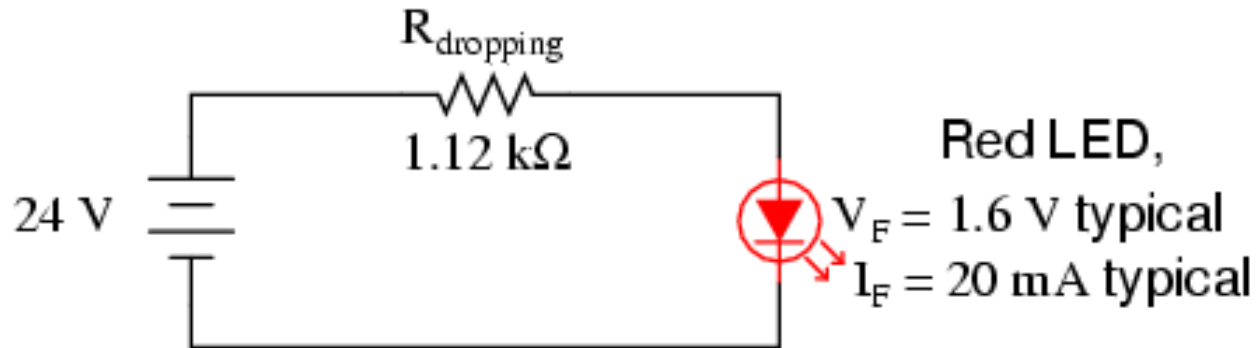
ไบอัสตรง จะเปล่งแสงออกมาได้ ซึ่งมีทั้ง สีแดง เขียว ฟ้าม ส้ม เหลือง ฯลฯ





การหาค่า R สำหรับต่อ LED

- เป็น R อนุกรมที่ใช้จำกัดกระแสที่จะไหลผ่าน LED โดยทั่วไปกระแสสูงสุดที่ใช้จะไม่เกิน 20 mA หากเกิน **หลอดอาจจะขาดได้**
- โดยใช้หลักการแบ่งแรงดันจะใช้สูตร $R = \frac{V - V_{LED}}{I}$



- $R = (24 - 1.6) / 0.02 = 1.12 \text{ k}\Omega$
- กรณี 5V $R = (5 - 1.6) / 0.02 = 170 \Omega$ แต่โดยทั่วไปจะใช้ $R = 220 \Omega$ สำหรับไฟ 5V (มักเพื่อให้มากเอาไว้นิดหน่อย)

Digital Output



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 Output

DIGITAL OUTPUT PROGRAMMING (ON-OFF)

Syntax:

```
digitalWrite(pin, logic)
```

Parameter:

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

logic : HIGH or LOW.



Digital Output

Example:

```
#define LED_on_Arduino 13
```

```
void setup()
```

```
{  
  pinMode(LED_on_Arduino, OUTPUT); // setup output  
}
```

```
void loop()
```

```
{  
  digitalWrite(LED_on_Arduino, HIGH); // Pin13 is HIGH  
  delay(250);  
  digitalWrite(LED_on_Arduino, LOW); // Pin13 is LOW  
  delay(250);  
}
```



Delay

- **delay(x)** หมายถึงให้หน่วงเวลา เป็นระยะเวลาเท่ากับ x มิลลิวินาที
- **millis()** ฟังก์ชันที่ส่งค่าจำนวน มิลลิวินาที นับจากที่โปรแกรมเริ่มรัน

```
unsigned long time;

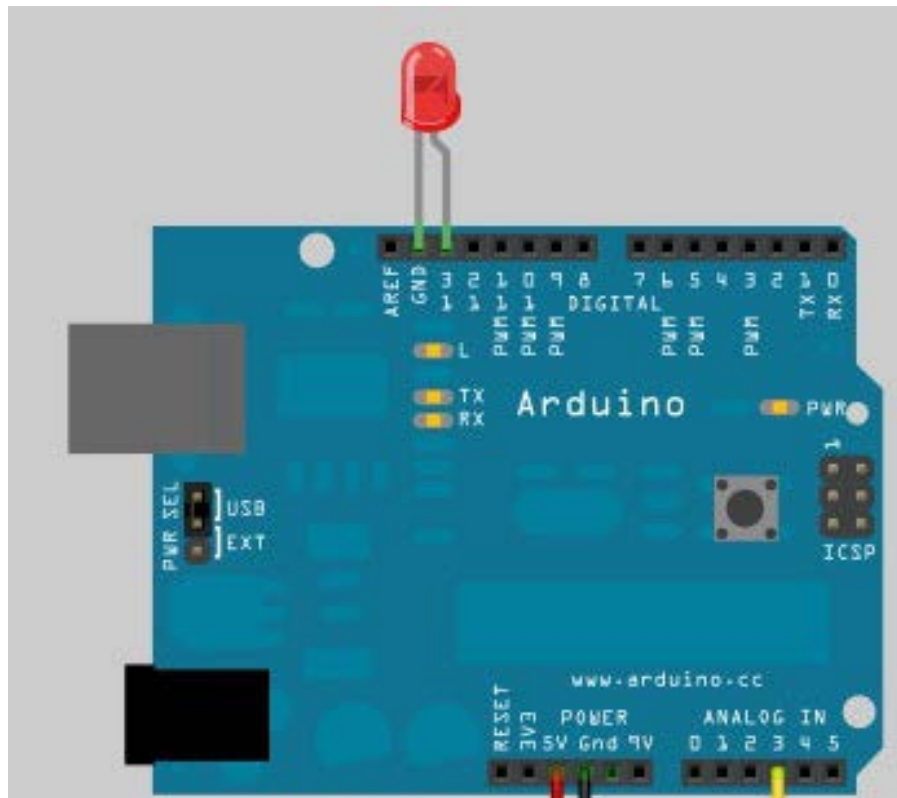
void setup(){
  Serial.begin(9600);
}

void loop(){
  Serial.print("Time: ");
  time = millis();
  //prints time since program started
  Serial.println(time);
  // wait a second so as not to send massive amounts of data
  delay(1000);
}
```

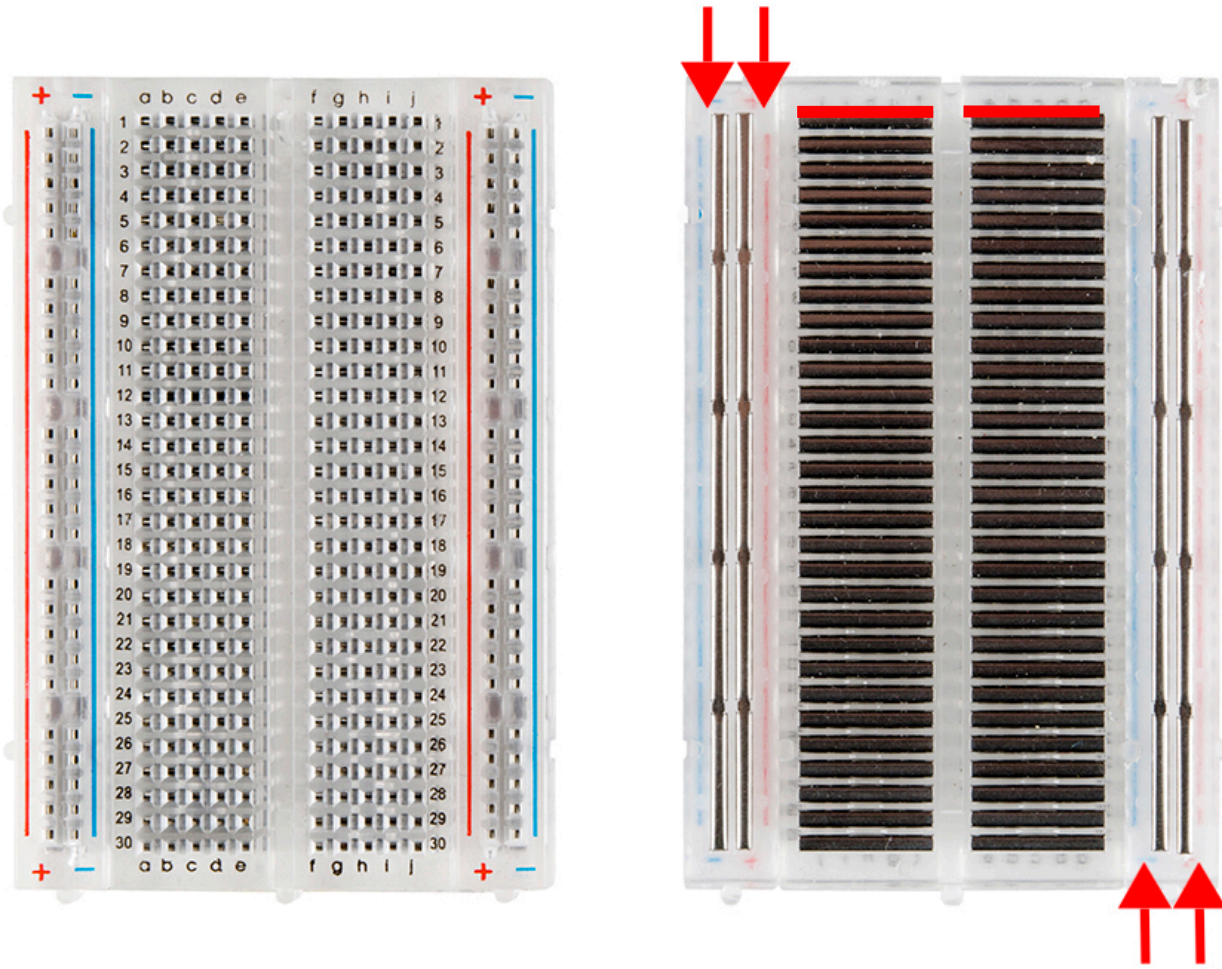


Activity

- ให้นำหลอด LED มาต่อกับบอร์ด โดยให้ขา + ต่อกับขา 13 และขาลบต่อกับ GND ดังรูป แล้วรันโปรแกรมอีกครั้ง



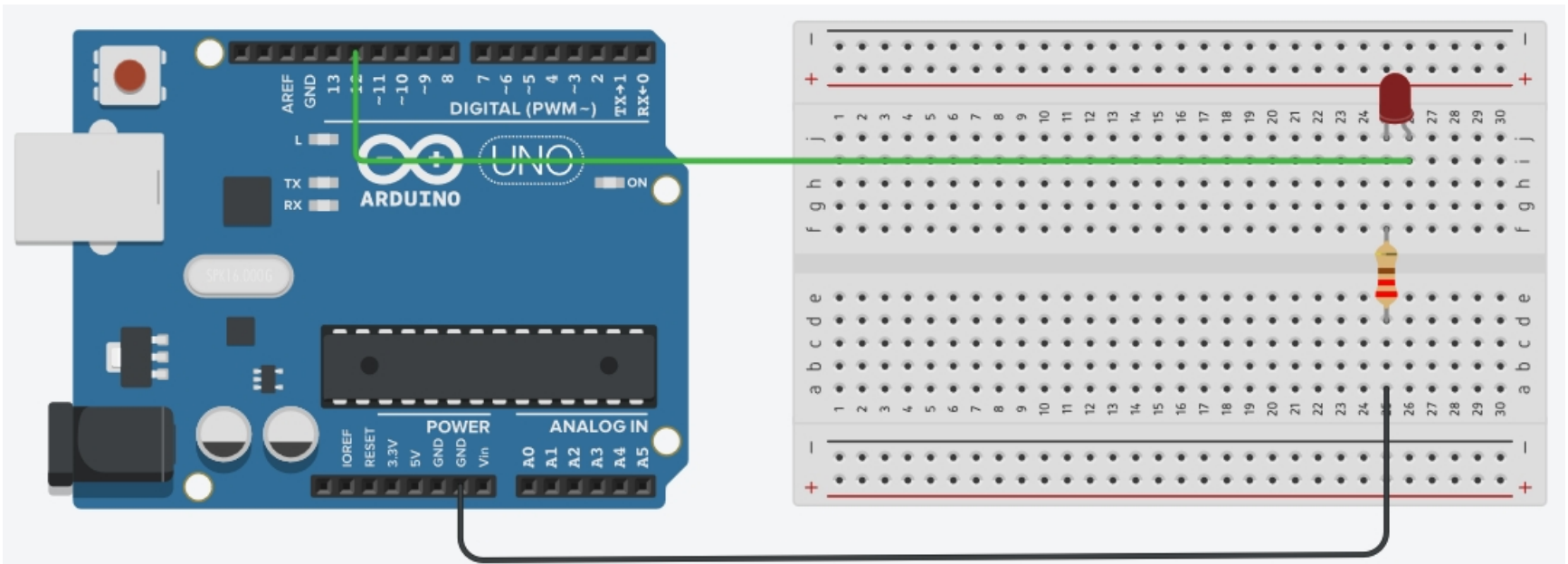
Protoboard หรือ Breadboard



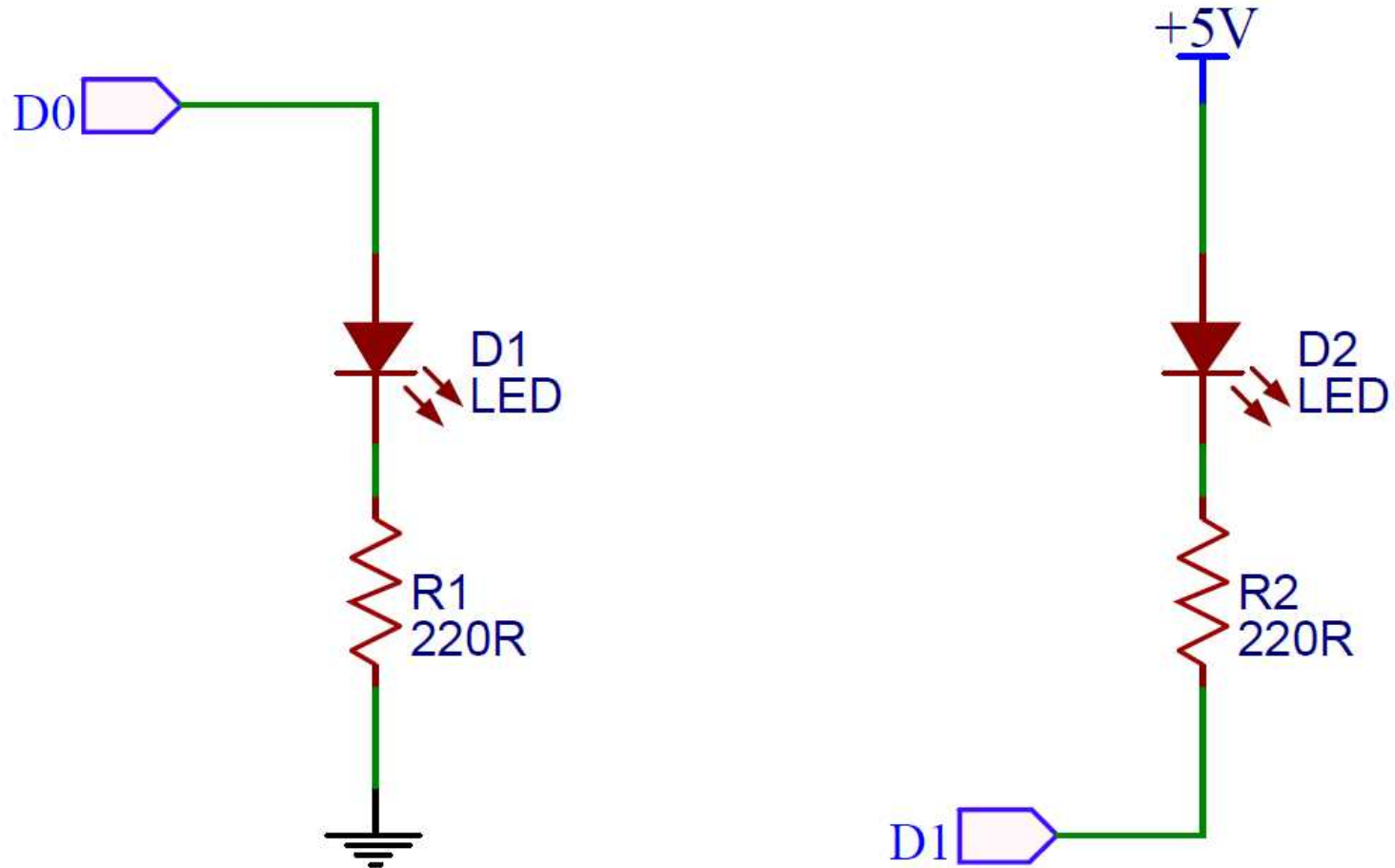


Activity

- ถ้าจะต่อ LED กับไฟ +5V จะต้องใช้ R อนุกรมค่าเท่าไร?
(กำหนดให้ LED มีค่ากระแส ณ จุดทำงาน $2V = 15mA$)
- ให้ต่อ LED บน Protoboard ตามรูปด้านล่าง (ขา 12) ทดสอบการทำงาน และ
ตอบคำถามว่า ทำไมต้องต่อ R



Positive Logic Interface & Negative Logic Interface



Activity

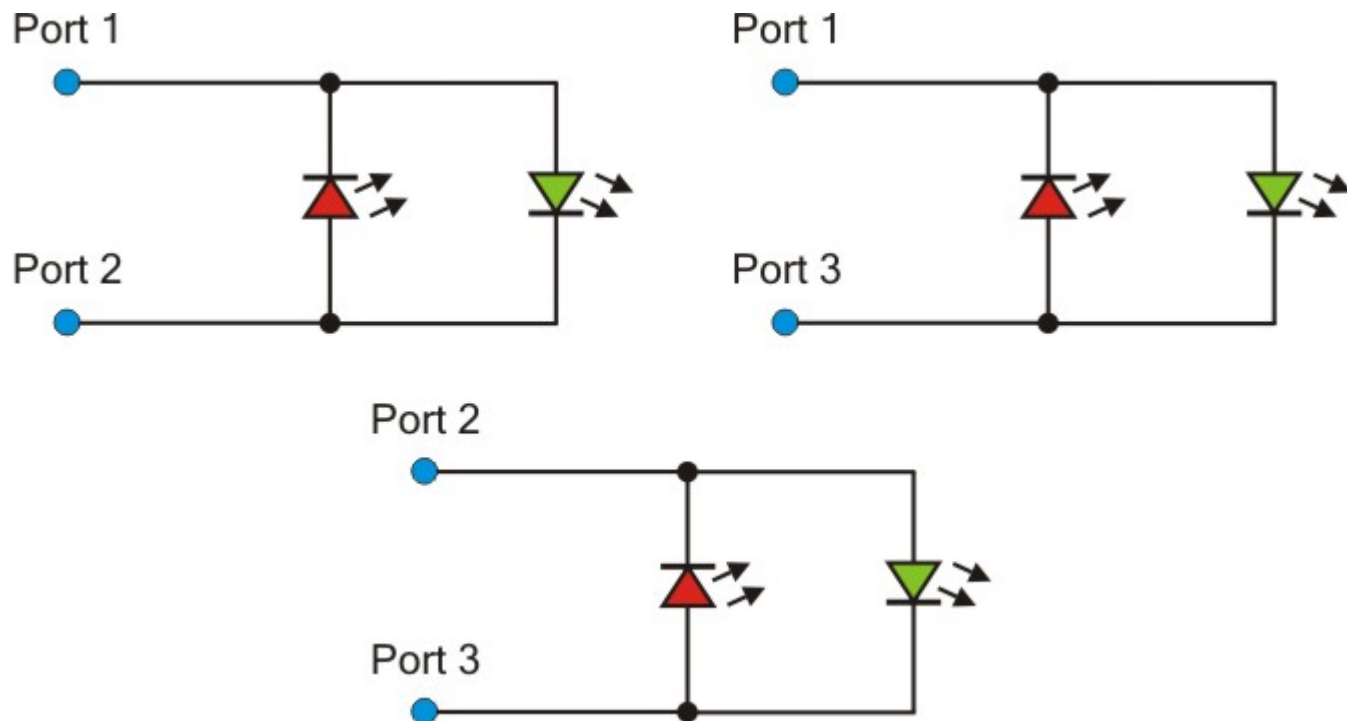


- ให้ต่อ LED กับบอร์ด Arduino จำนวน 4 ดวง ในแบบ Positive Logic
- เขียนโปรแกรมให้แสดง OFF-OFF-OFF-OFF -> OFF-OFF-OFF-ON -> OFF-OFF-ON-ON -> OFF-ON-ON-ON -> ON-ON-ON-ON โดยเว้นจังหวะละ 500 ms
- ให้วนซ้ำแสดงตามข้อก่อนหน้าไปเรื่อยๆ

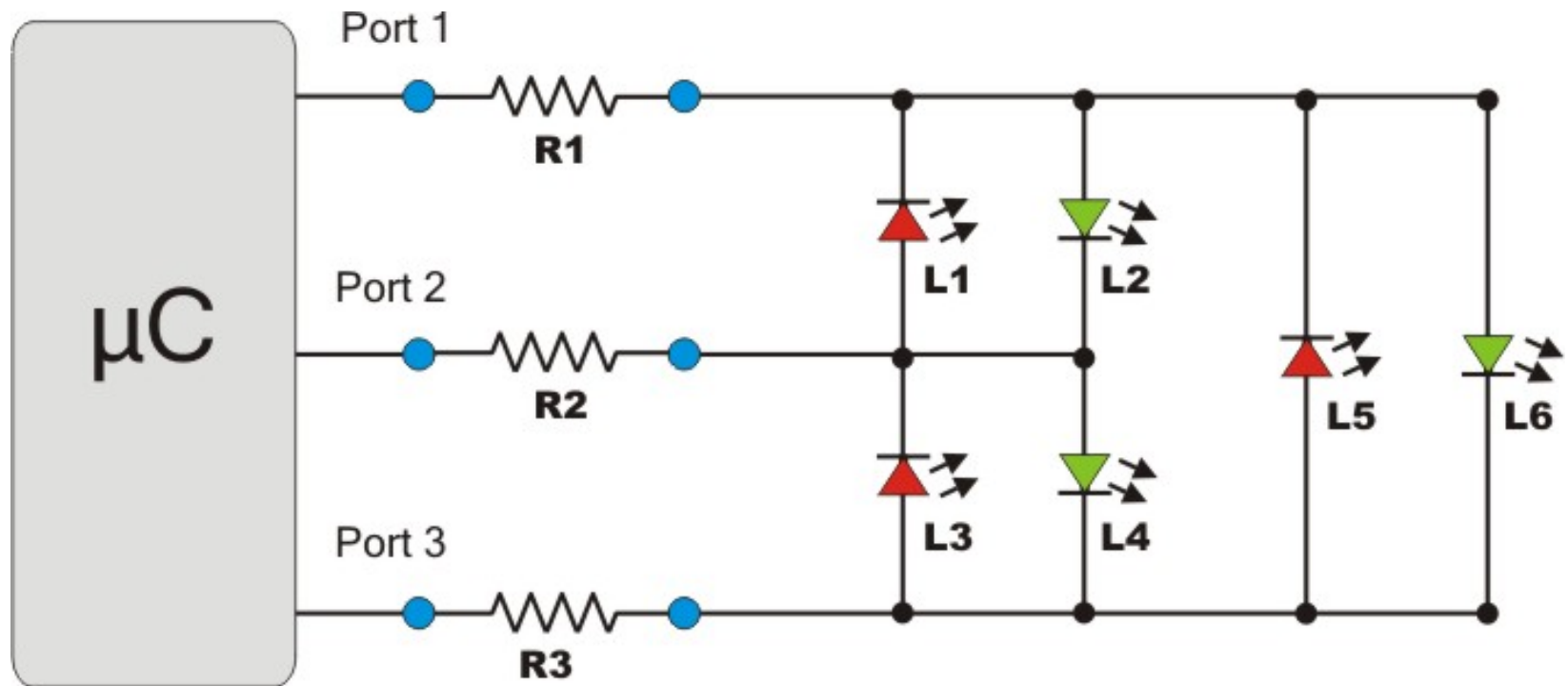


Charlieplexing

- เป็นเทคนิคในการลดขาที่ต้องใช้ในการขับ LED เช่นจากรูปด้านล่างนี้ จะใช้ขาจำนวน 3 ขา ในการขับ LED จำนวน 6 ดวง



Charlieplexing





Exercise

- ให้ต่อวงจร LED ในแบบ Charlieplexing จำนวน 3 ขา
- เขียนโปรแกรมให้แสดงไฟวิ่ง 1 ดวงไปกลับ



Switch and Pullup Pulldown



Top

Connected



Connected

Front



Side



Released

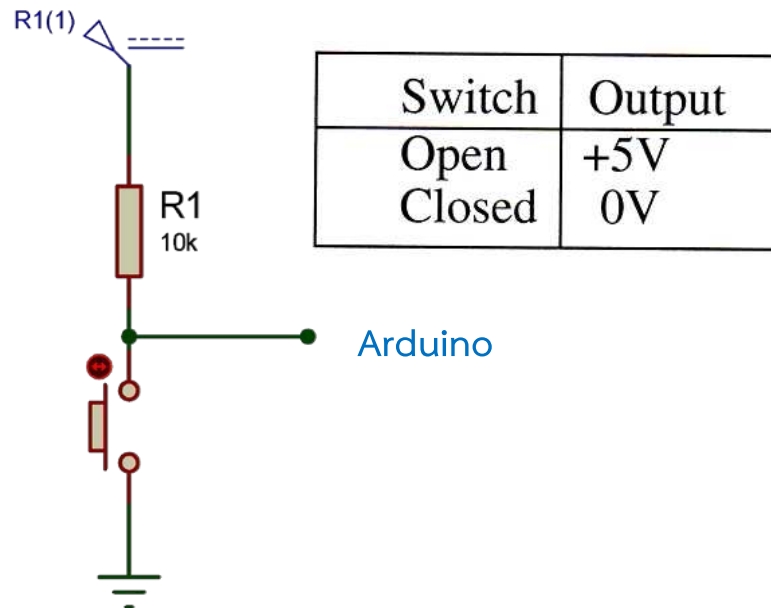


Pressed

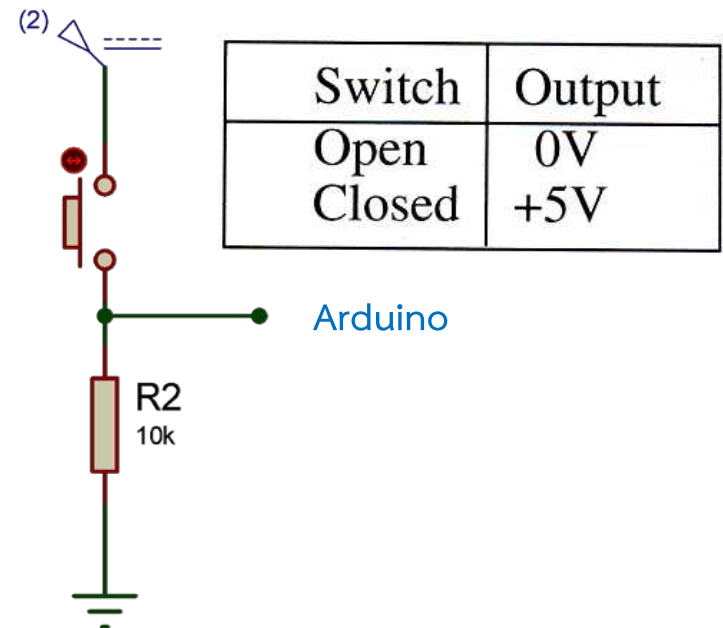
Digital Input



Pull Up



Pull Down



Digital Input



SETUP PINMODE

Syntax:

```
pinMode(pin, mode)
```

Parameter:

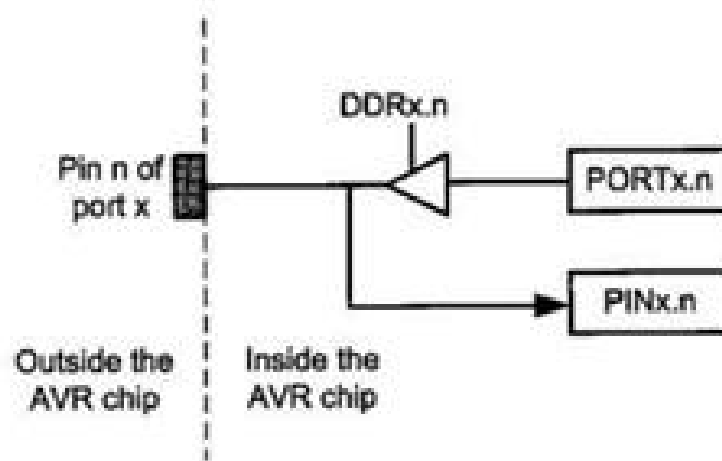
pin: the number of the pin whose mode you wish to set

mode: INPUT, OUTPUT or INPUT_PULLUP.

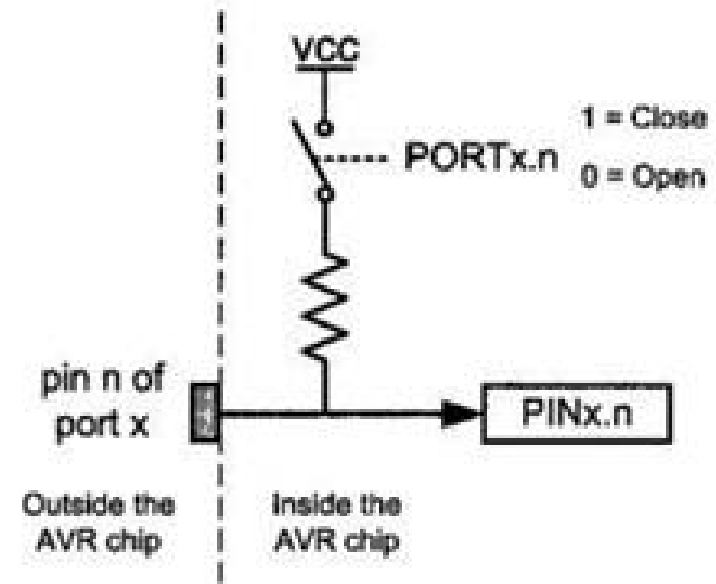
Digital Input



INPUT_PULLUP



I/O Port in 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:

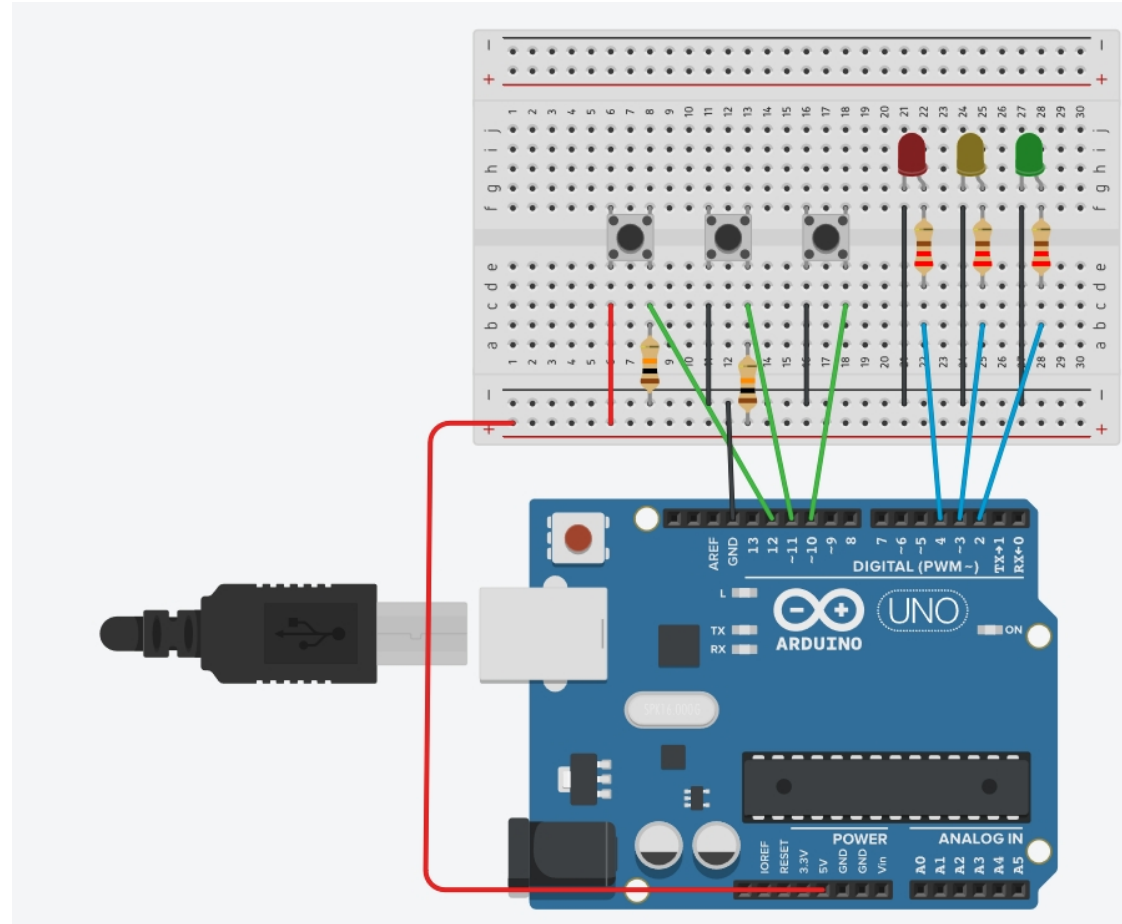
```
#define button 2    // switch input Active Low  
#define pressed LOW
```

```
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



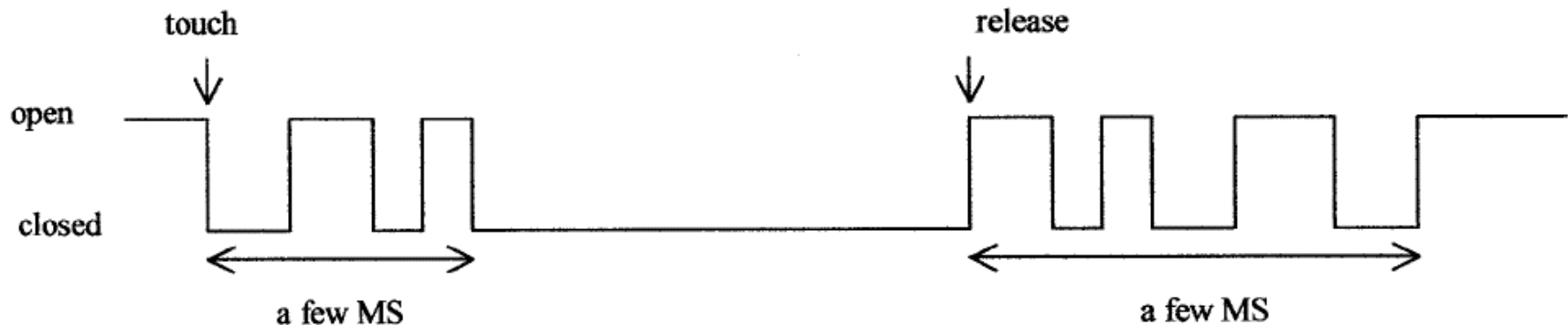
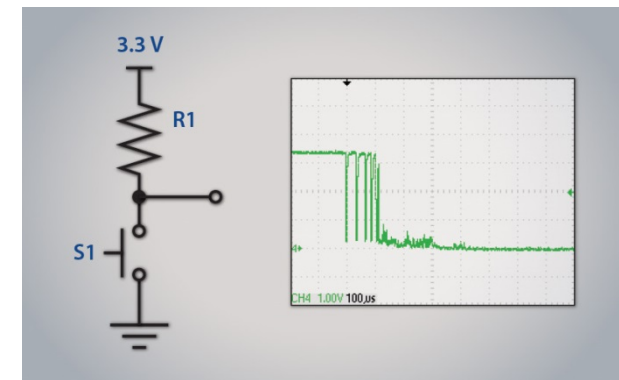
- ต่อวงจรโดยใช้ Switch
 - Pull Up
 - Pull Down
 - Internal Pullup
- กดปุ่มไหน ให้ไฟติด
(ไม่ต้องกดค้าง)





Bounce Problem

- ในการกดสวิตช์ 1 ครั้ง จะมีช่วงเวลาสั้นๆ ที่เกิดสัญญาณคล้ายกับการกดสวิตช์หลายครั้ง เนื่องจากหน้าสัมผัส
- การแก้ไข
 - สวิตช์บางตัวที่มีราคาแพงจะไม่เกิด Bounce
 - แก้ไขโดยวิธีการทางฮาร์ดแวร์
 - แก้ไขโดยวิธีการทางซอฟต์แวร์

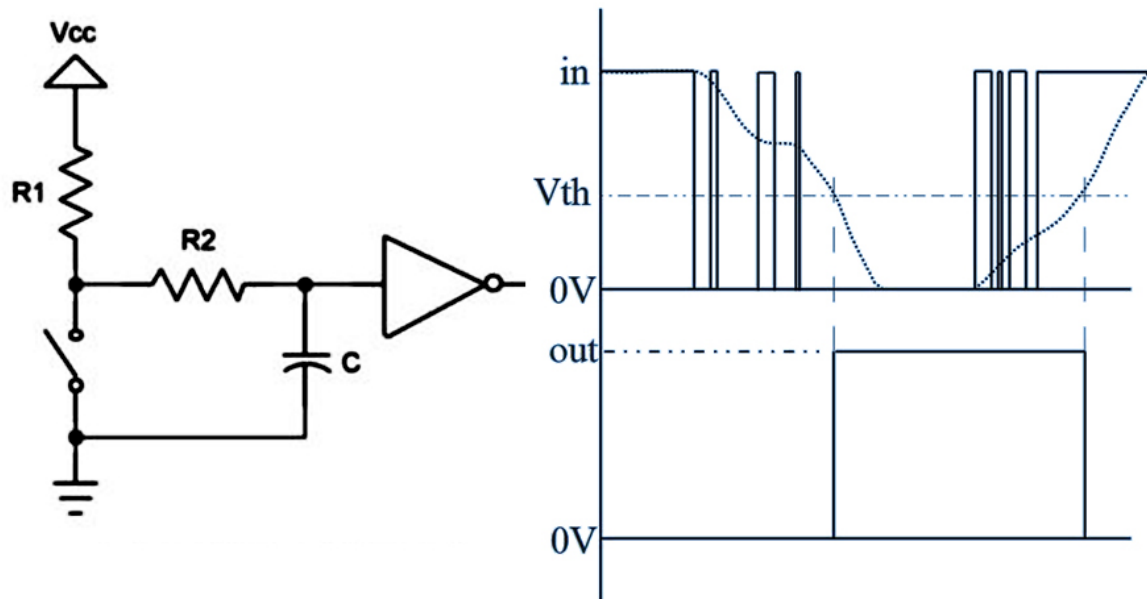




Bounce Problem

- 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();
    }

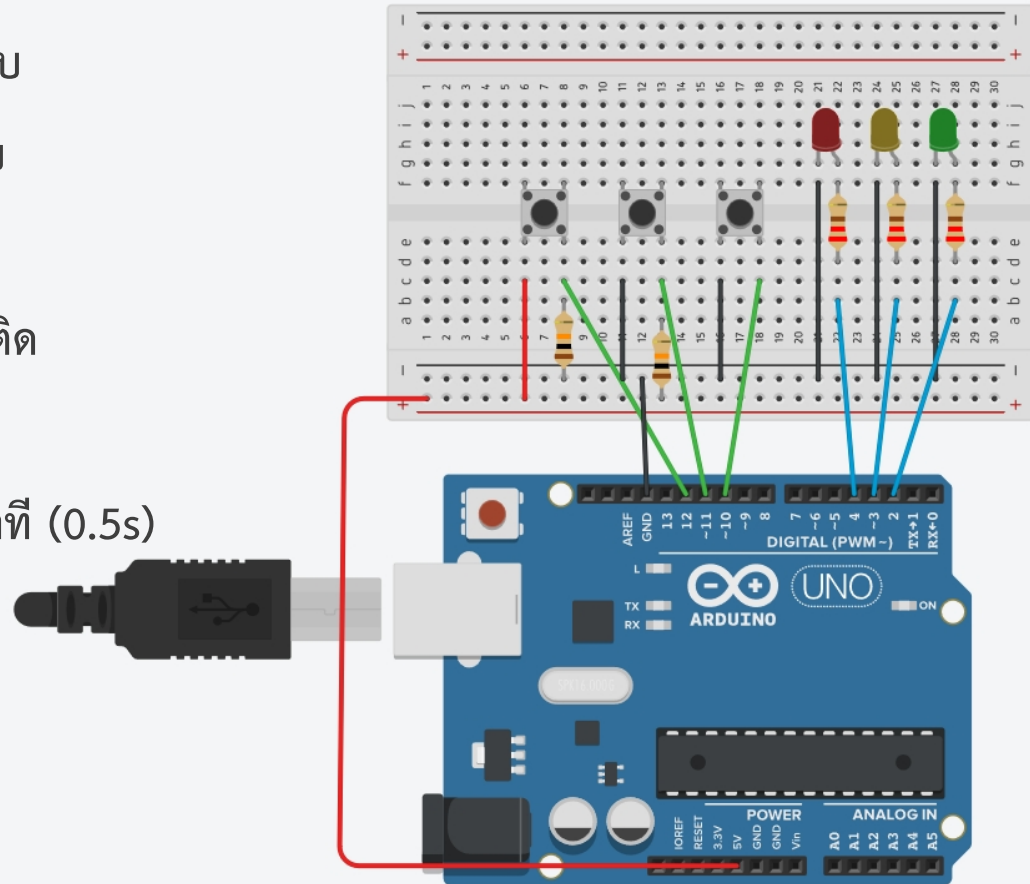
    if ((millis() - lastDebounceTime) > debounceDelay) {
        // whatever the reading is at, it's been there for longer
        // than the debounce delay, so take it as the actual current state:

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

Assignment #1



- 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