



01076101

วิศวกรรมคอมพิวเตอร์เบื้องต้น

Introduction to Computer Engineering

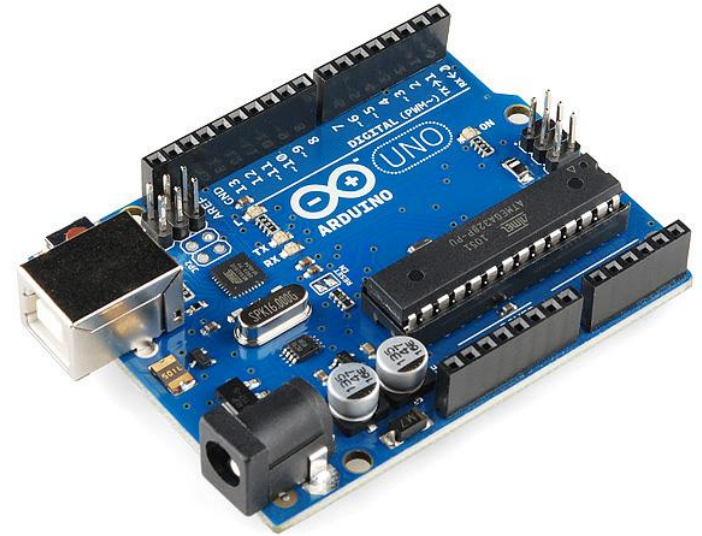
Arduino #1

LED, Digital Output

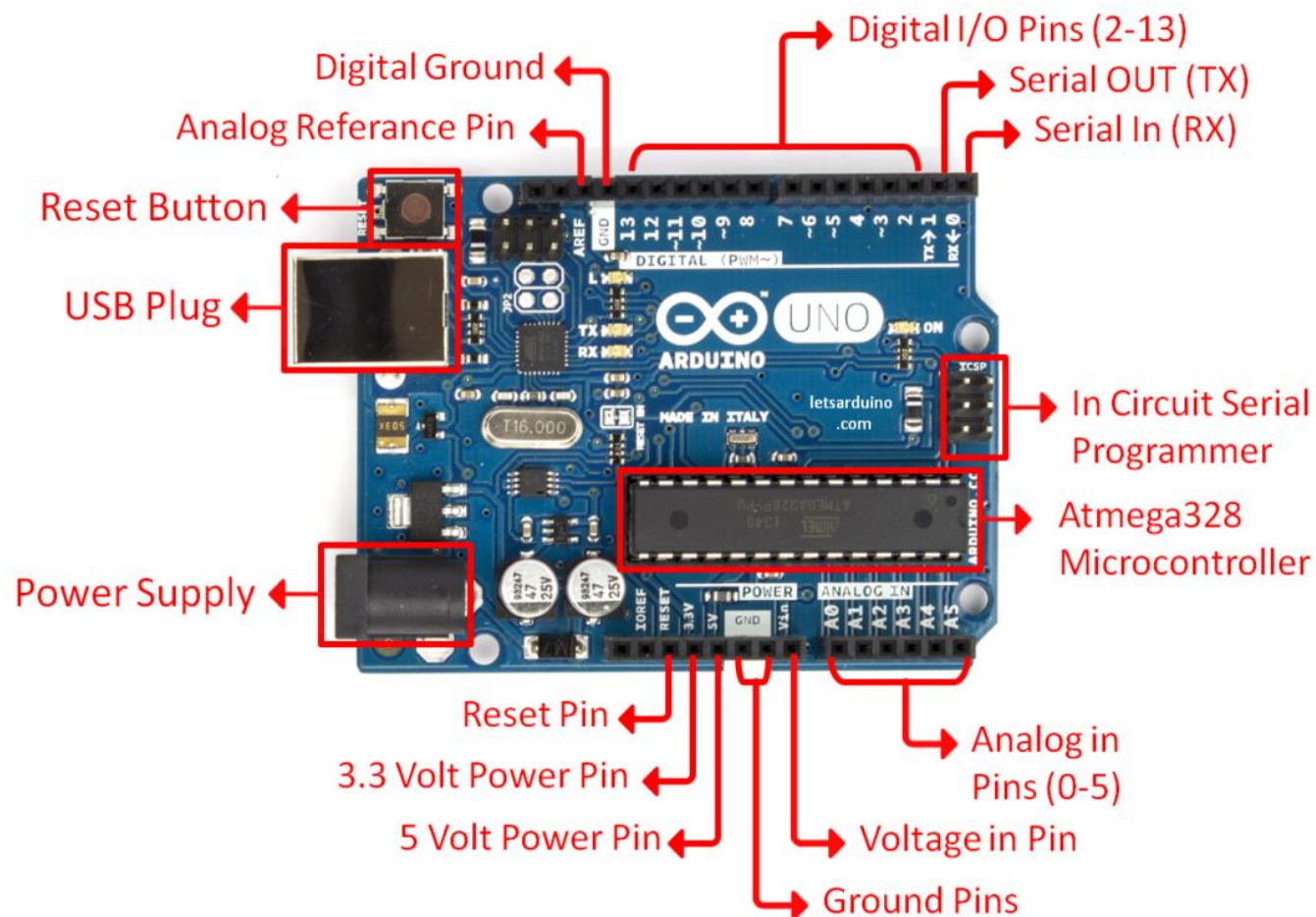


# บอร์ด Arduino

- อ่านว่า (อา-ดู-อิ-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 <https://www.arduino.cc/en/software>

## Downloads



### Arduino IDE 1.8.15

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

#### SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

#### DOWNLOAD OPTIONS

**Windows** Win 7 and newer

**Windows** ZIP file

**Windows app** Win 8.1 or 10



**Linux** 32 bits

**Linux** 64 bits

**Linux** ARM 32 bits

**Linux** ARM 64 bits

**Mac OS X** 10.10 or newer

[Release Notes](#) [Checksums \(sha512\)](#)



# ติดตั้ง Arduino IDE




- เลือก **JUST DOWNLOAD** และ ติดตั้งลงในเครื่อง

**Support the Arduino IDE**

Since the release 1.x release in March 2015, the Arduino IDE has been downloaded **53,522,788** times — impressive! Help its development with a donation.

[JUST DOWNLOAD](#) [CONTRIBUTE & DOWNLOAD](#)

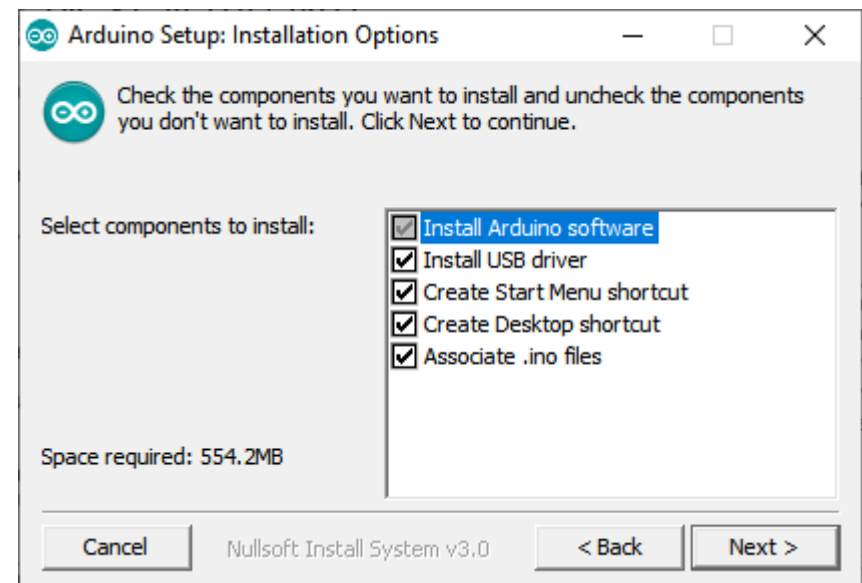
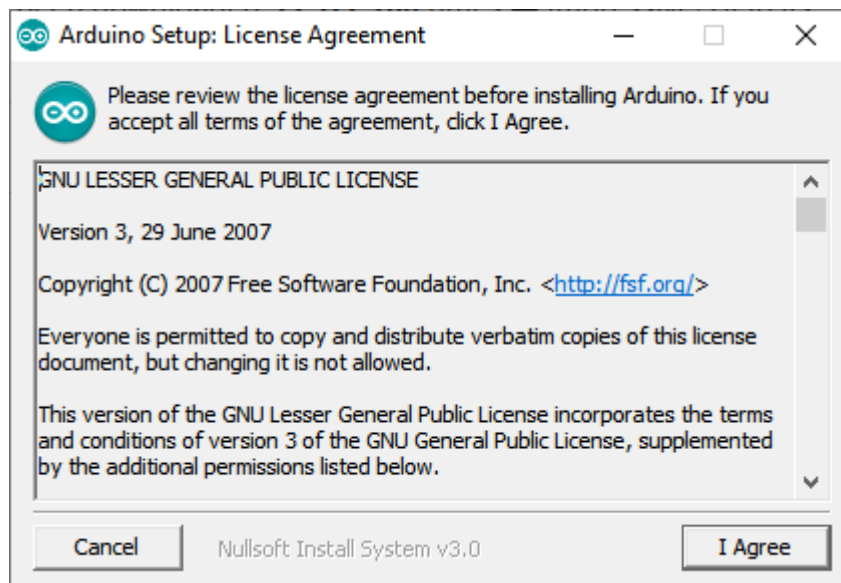


[Learn more about donating to Arduino.](#)

# ติดตั้ง Arduino IDE



- คลิก I Agree และ Next ตามลำดับ

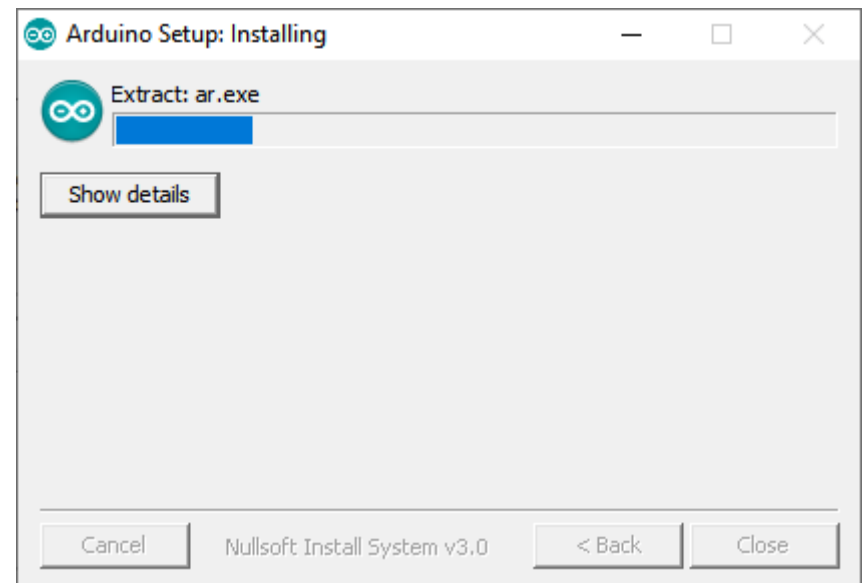
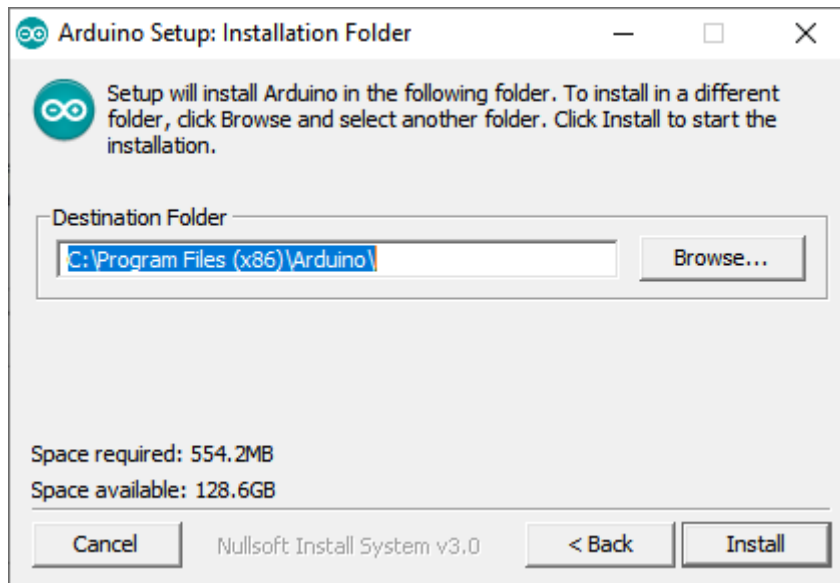




# ติดตั้ง Arduino IDE



- คลิก Install



# ติดตั้ง Arduino IDE



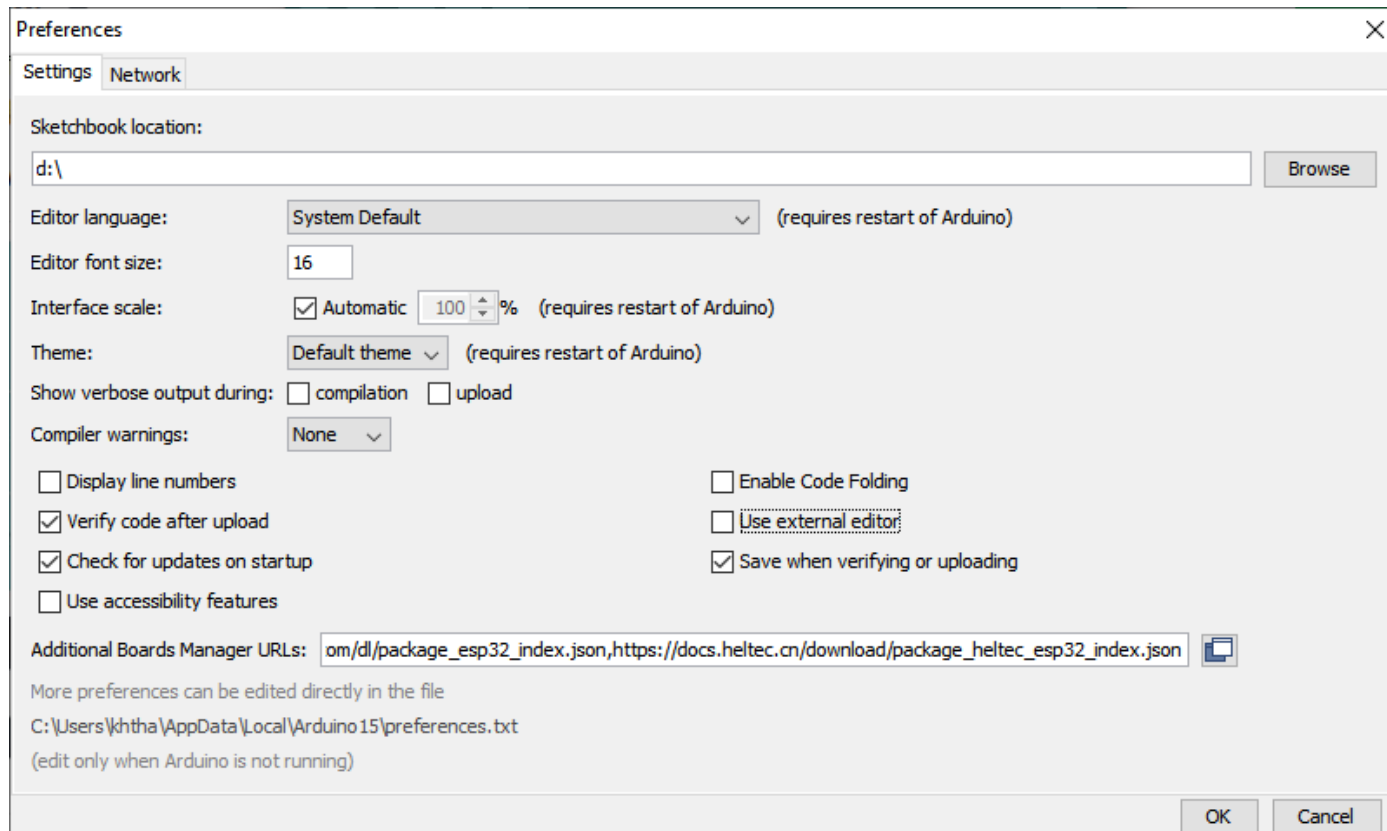
- เรียกขึ้นมาทำงาน



# ตั้งค่า preferences



- คลิกที่ File -> Preferences แล้วปรับแต่งได้ตามสะดวก เช่น theme, ขนาด font, แสดงเลขบรรทัด



# ตั้งค่า preferences

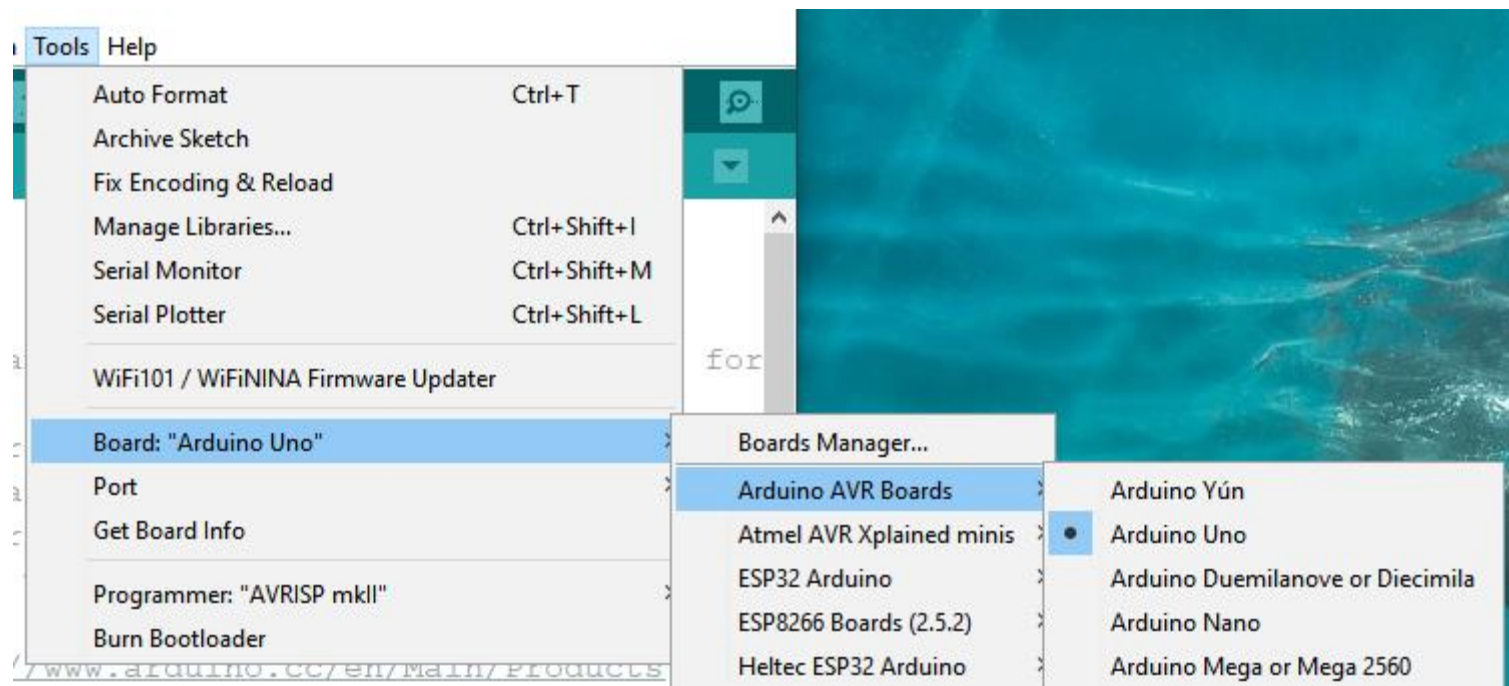


- กำหนดจุดที่ใช้เก็บโปรแกรม
- ขนาดฟอนต์
- Show verbose output during จะให้แสดงรายละเอียดระหว่างที่ Compile หรือ Upload หรือไม่
- แสดง Warning หรือไม่
- แสดงเลขที่บรรทัด
- Check for updates on startup
- Save when verifying or uploading



## กำหนด Board และ Port

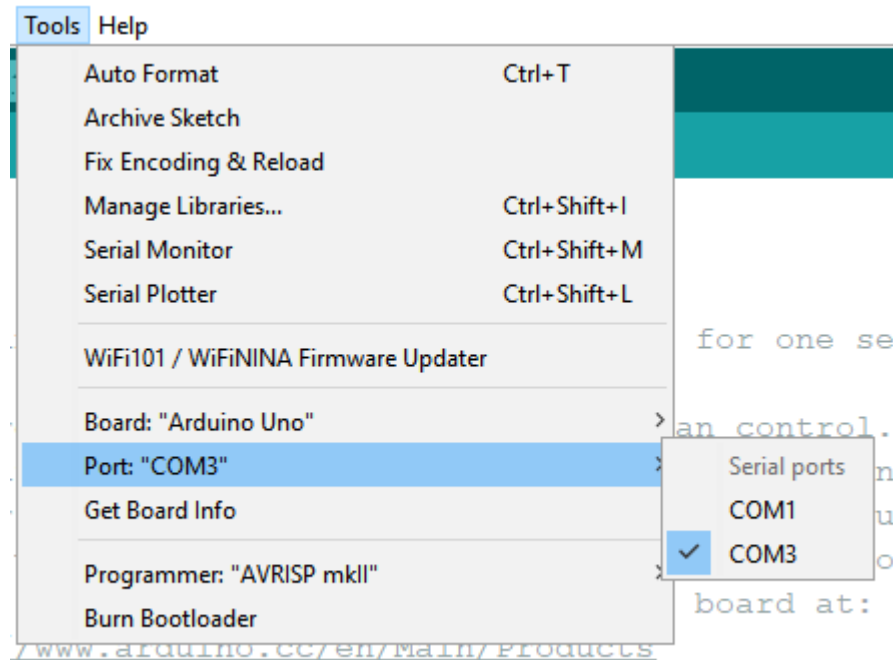
- เสียบ USB ระหว่างเครื่องคอมพิวเตอร์กับบอร์ด
- ไปที่ Tools -> Board เลือก Arduino Uno (ถ้าเป็น Nano เลือก Nano)





## 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:  
  
}
```





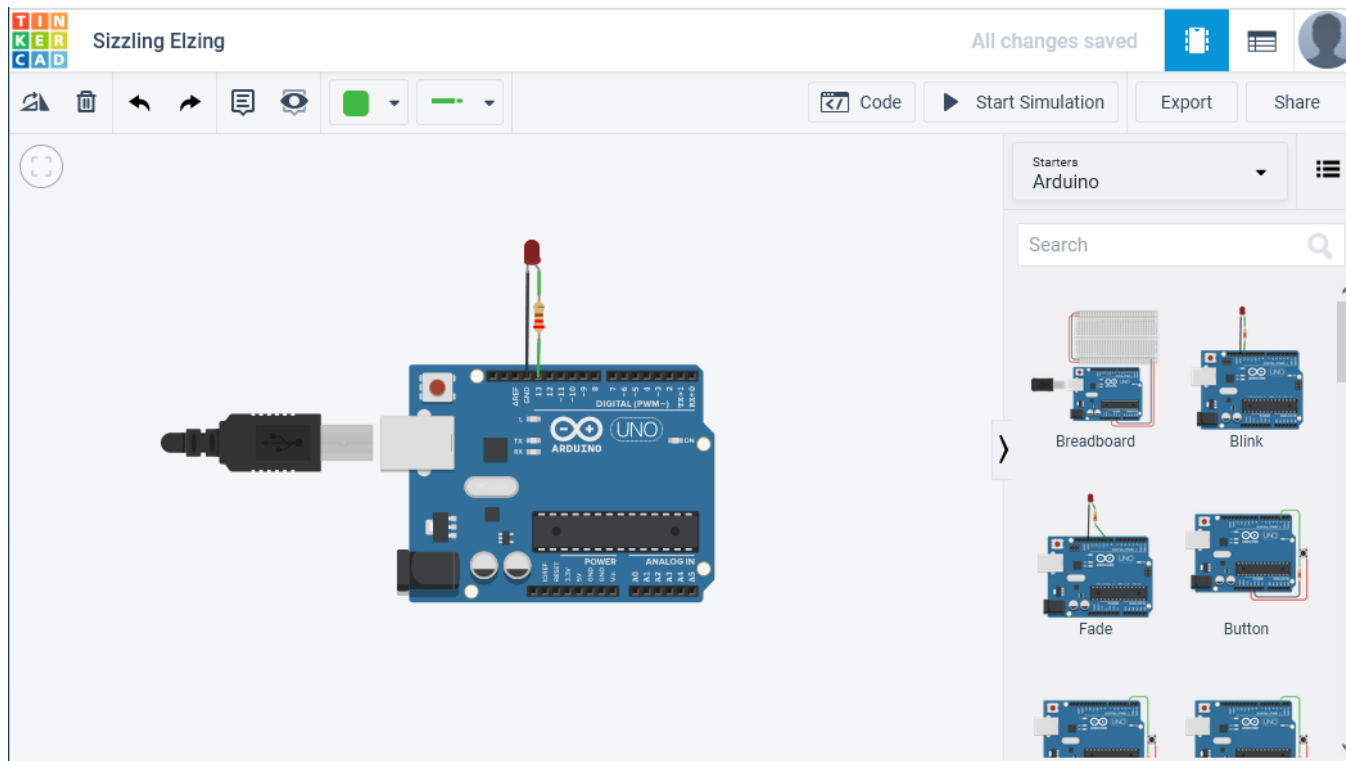
## Activity

- เลือก Files -> Examples -> Basics -> Blink
- จากนั้น Upload ขึ้นบอร์ด 
- หากไฟบนบอร์ดกระพริบ แสดงว่า การติดตั้งสำเร็จ
- กรณีใช้บอร์ด Arduino Nano แล้ว upload ไม่ได้ให้ลองเปลี่ยนเป็น Tools -> Processor ATmega328P (Old Bootloader) แล้ว Upload ใหม่อีกครั้ง



# Tinkercad

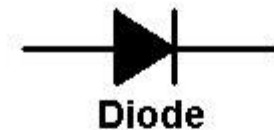
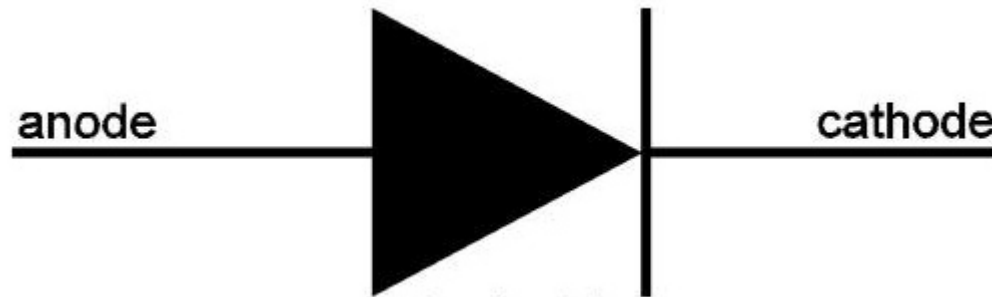
- ไปที่เว็บ <https://www.tinkercad.com/>
- สร้าง account โดยเลือก personal account ในช่อง starters เลือก Arduino แล้วลากตามรูปมาวาง จากนั้นกดที่ Code แล้วเลือก Text แล้วกด Start Simulation





# ไดโอด

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

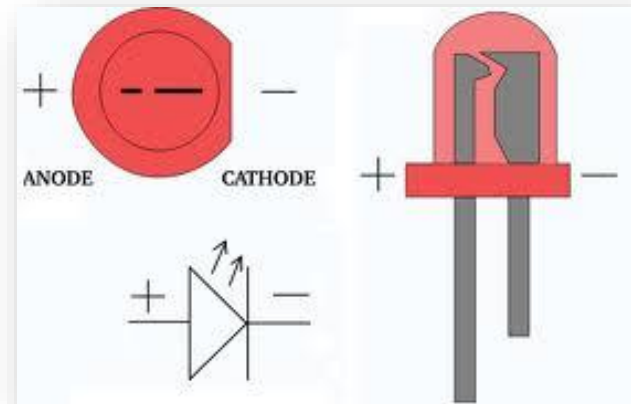
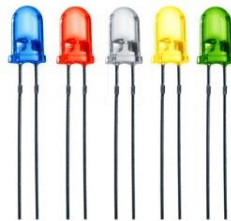




# ไดโอดเปล่งแสง แอลอีดี (LIGHT EMITTING DIODES)

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

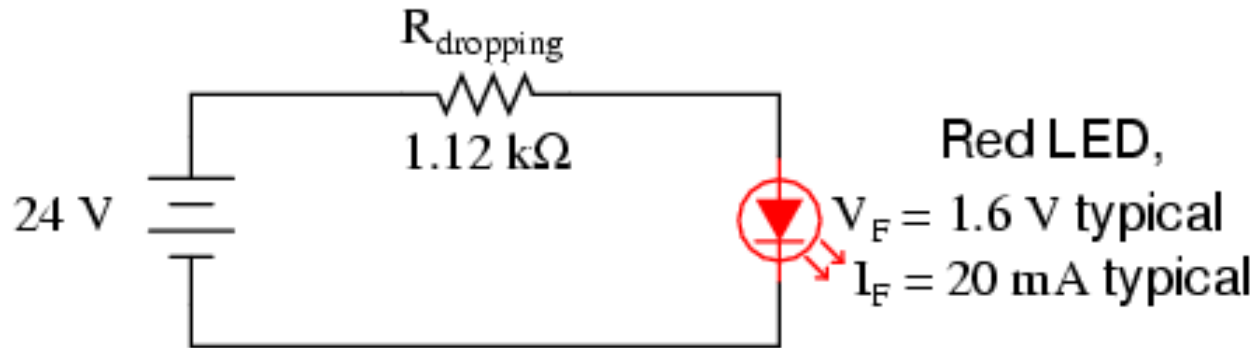
**ไบอัสตรง** จะเปล่งแสงออกมาได้ ซึ่งมีทั้ง สีแดง เขียว ฟ้าม ส้ม เหลือง ฯลฯ





## 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;

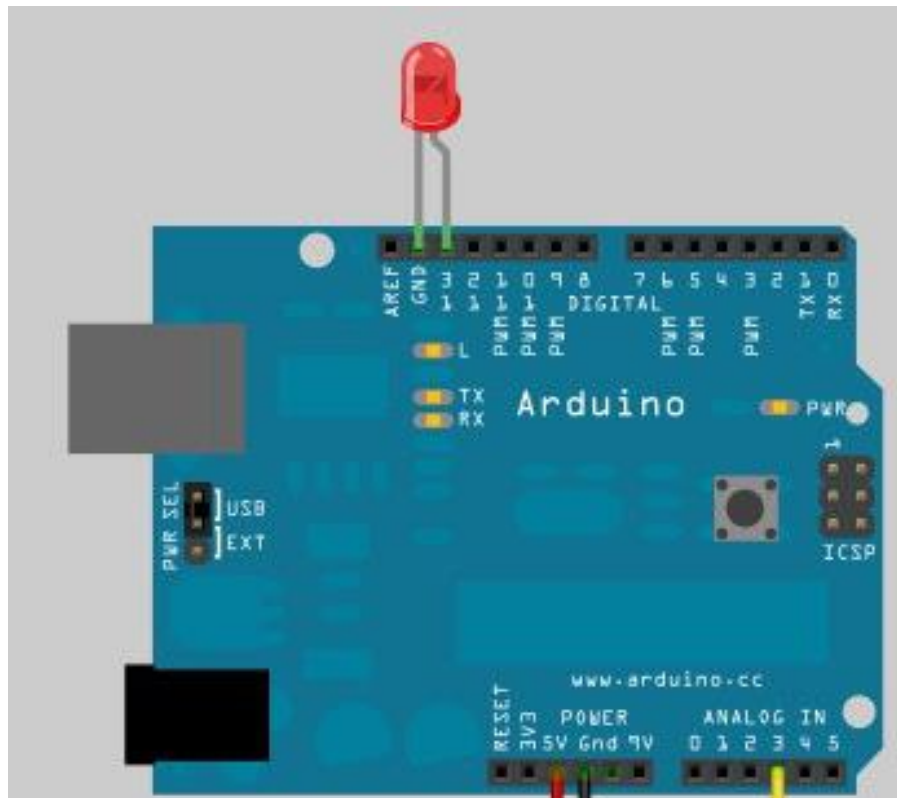
void setup(){
  Serial.begin(9600);
}

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);
}
```

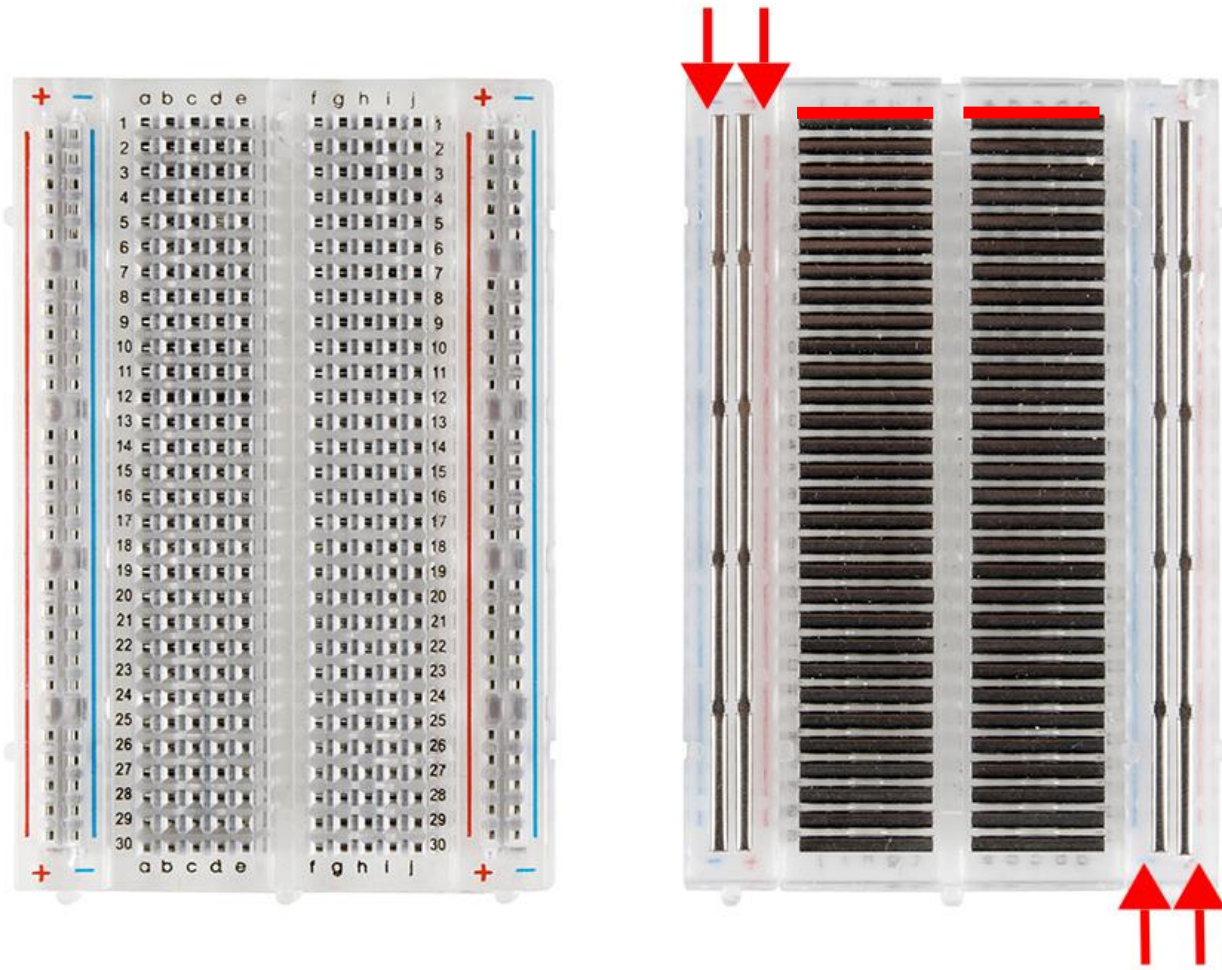


## Activity

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



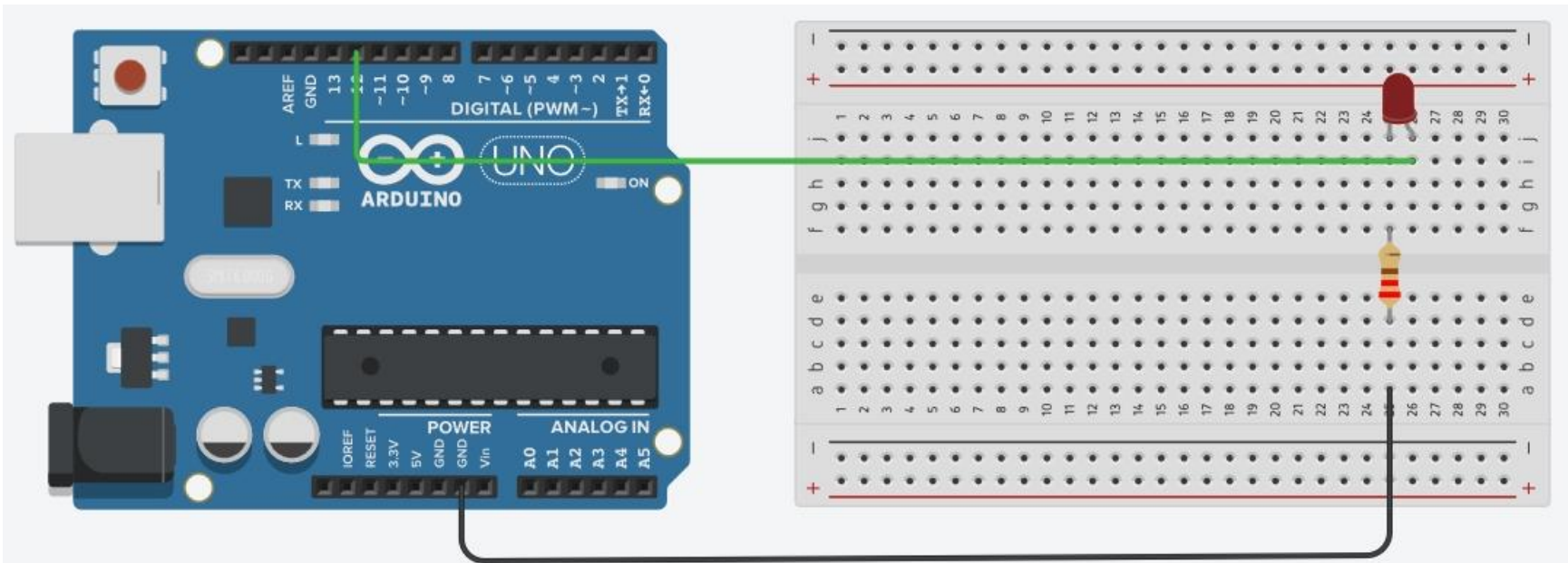
# Protoboard หรือ Breadboard



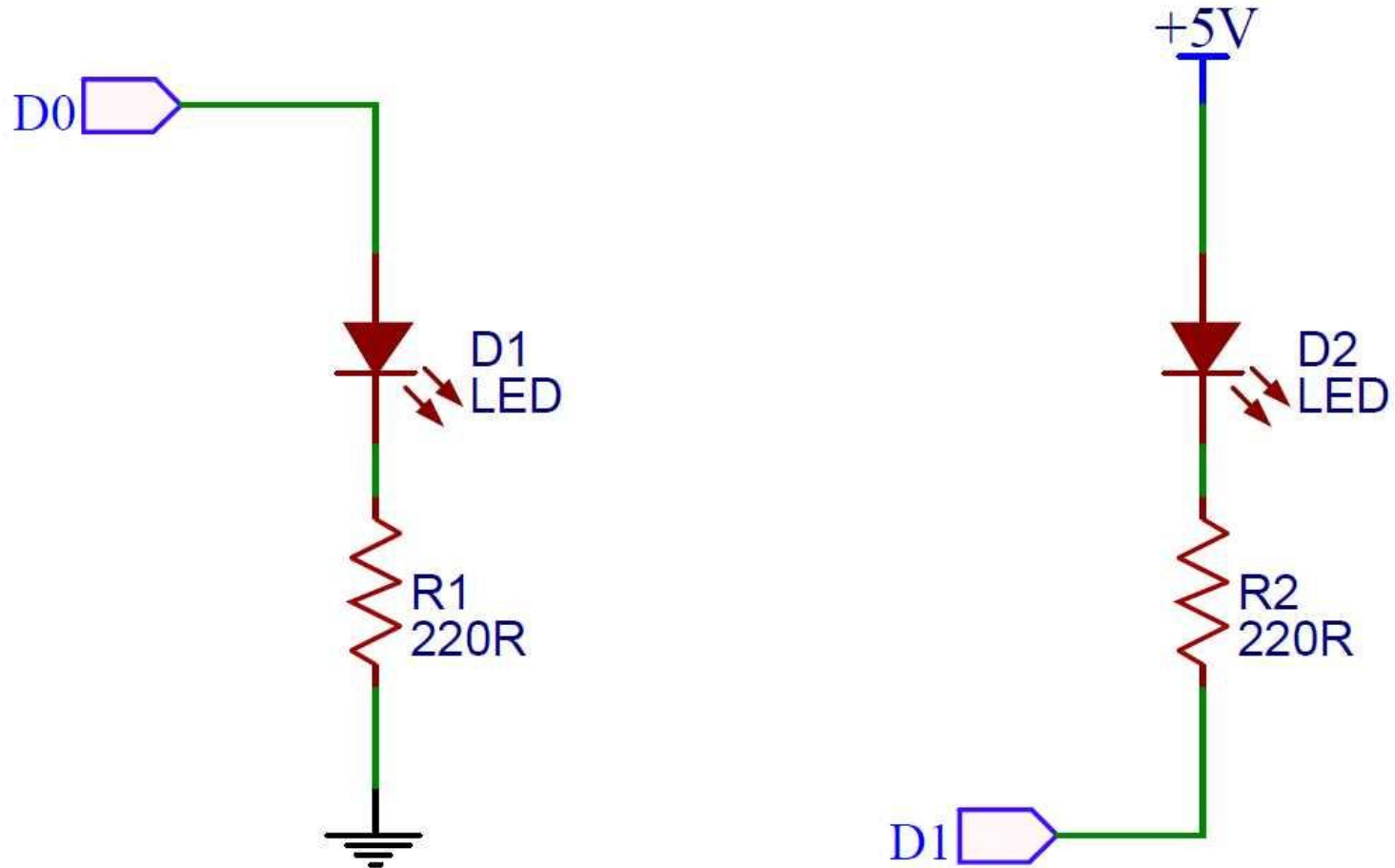


## Activity

- 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

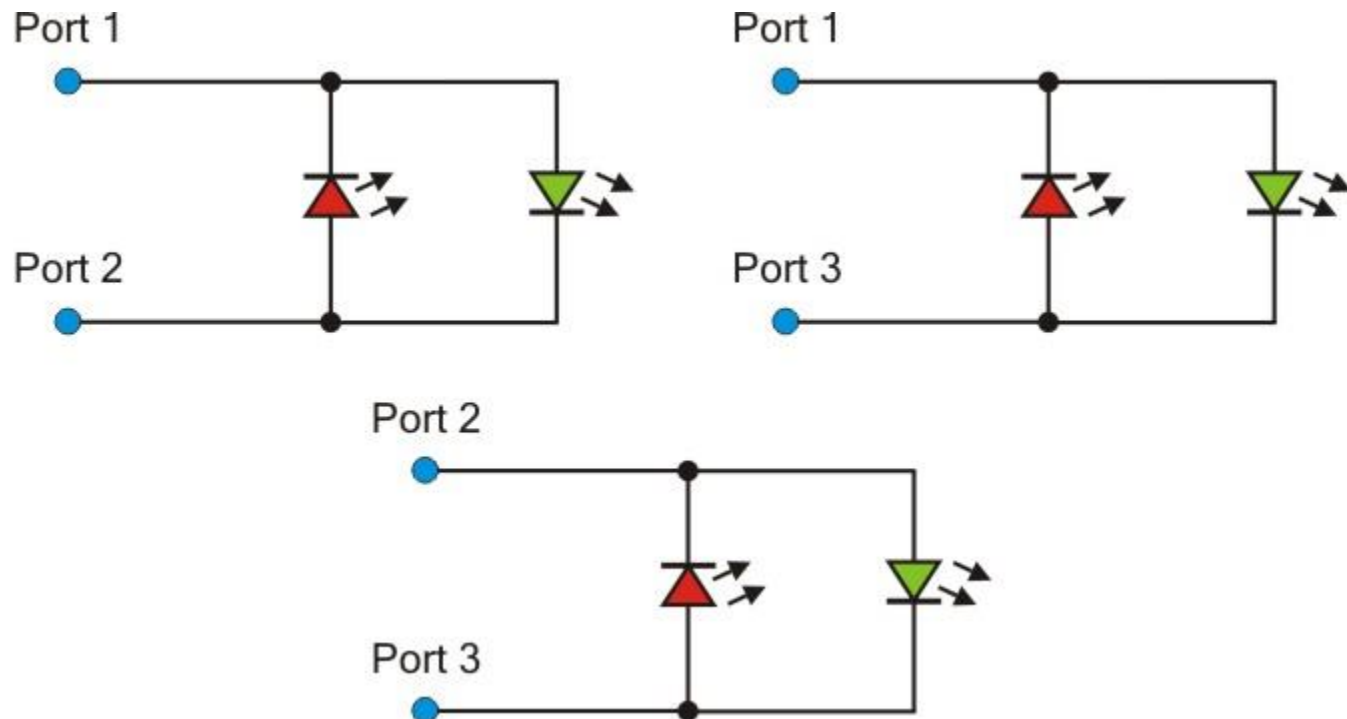
- ให้ต่อ LED กับบอร์ด Arduino จำนวน 4 ดวง ในแบบ Positive Logic
- เขียนโปรแกรมให้แสดง OFF-OFF-OFF-OFF -> OFF-OFF-OFF-ON -> OFF-OFF-ON-ON -> OFF-ON-ON-ON -> ON-ON-ON-ON โดยเว้นจังหวะละ 500 ms
- ให้วนซ้ำแสดงตามข้อก่อนหน้าไปเรื่อยๆ



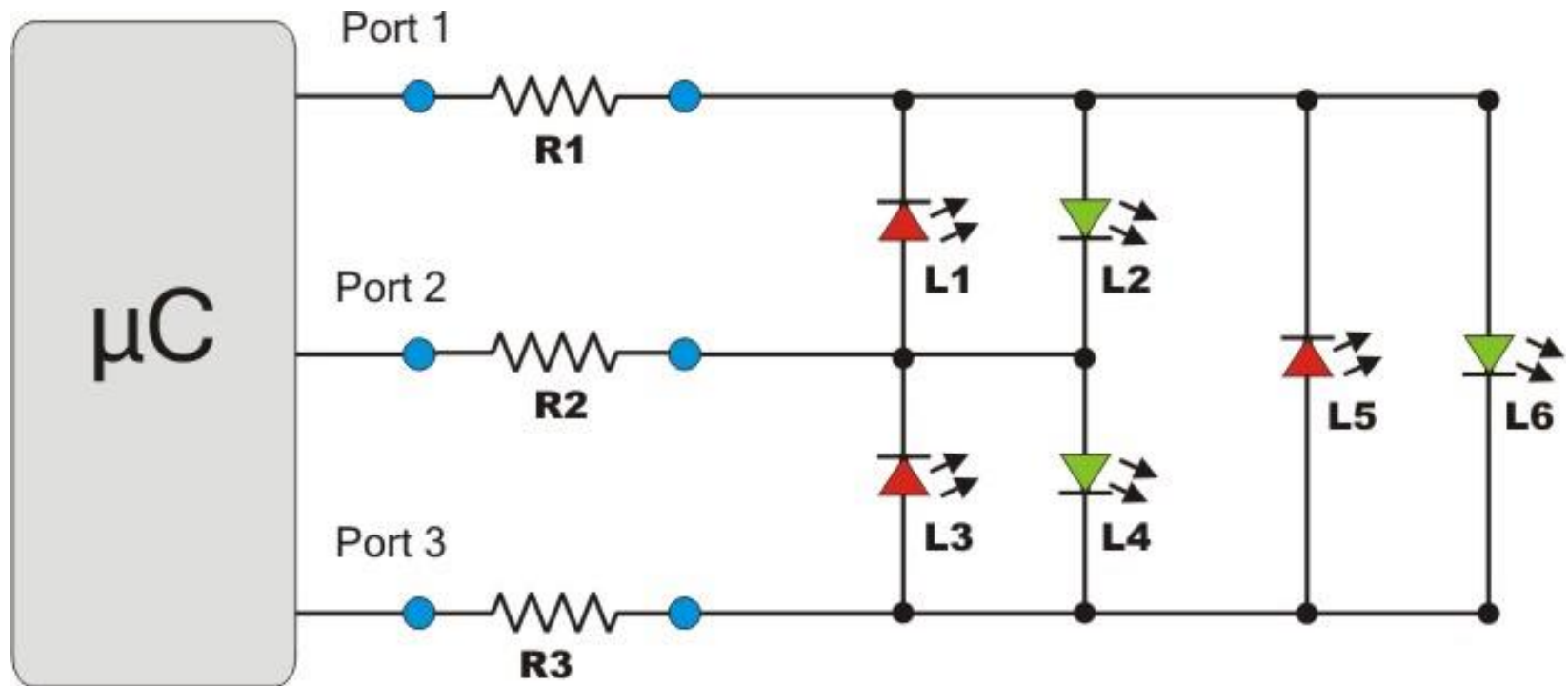


# Charlieplexing

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



# Charlieplexing





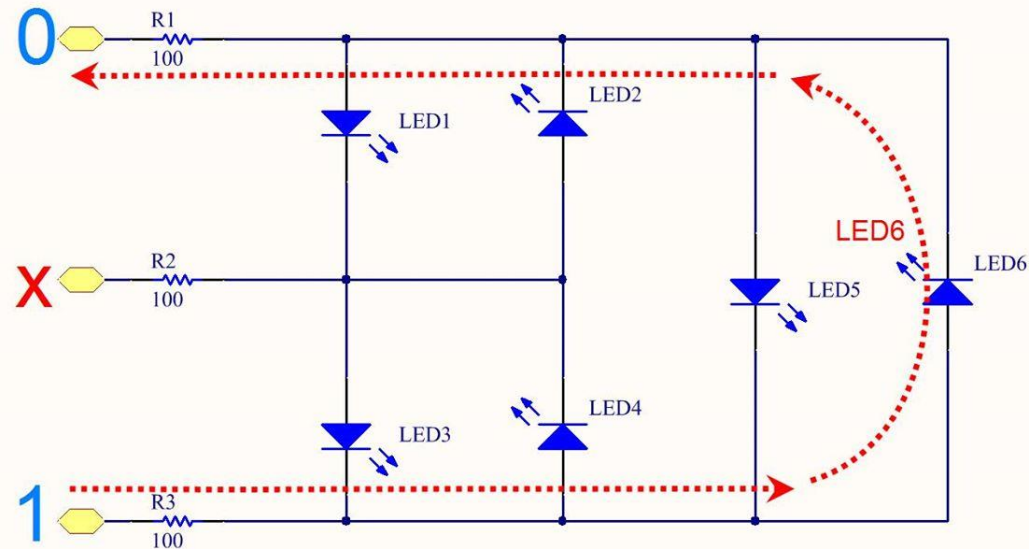
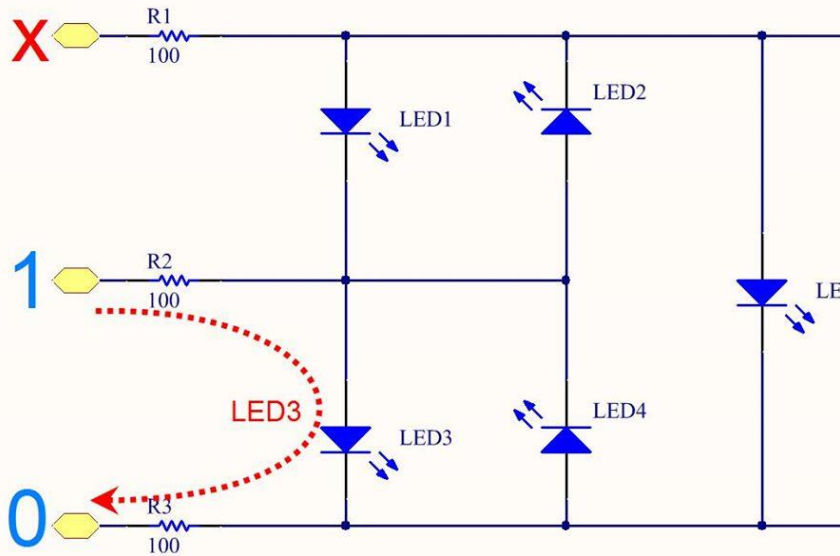
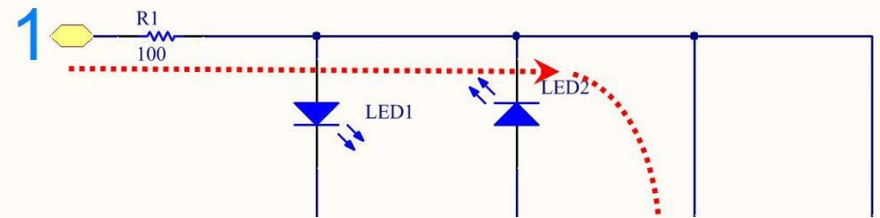
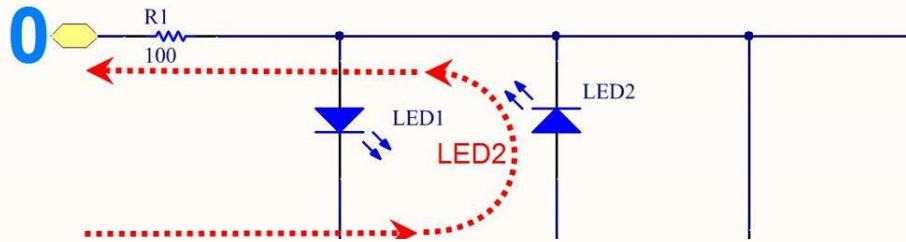
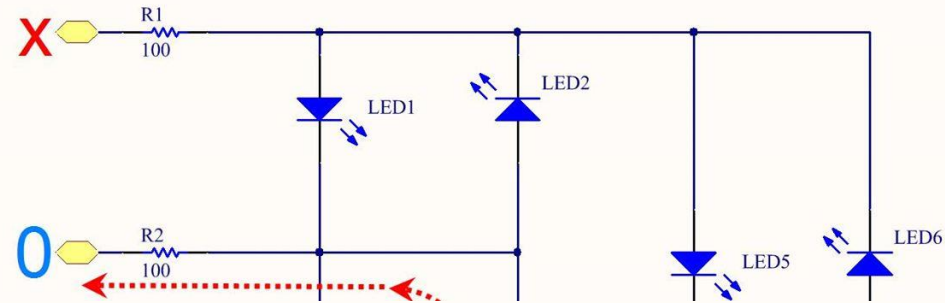
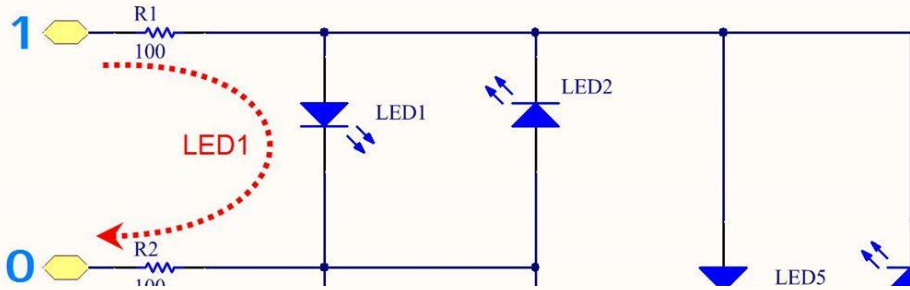
# Charlieplexing

- การเขียนโปรแกรมสำหรับ Charlieplexing ต้องมีเทคนิคเล็กน้อย เนื่องจากเราจะต้องใช้ขาเพียง 2 ขาเท่านั้นที่จะทำงานได้ มิฉะนั้นจะติด 2 ดวง จึงต้องเขียนในลักษณะนี้

```
void setup()
{
    pinMode(2, OUTPUT);
    digitalWrite(2, HIGH);
    pinMode(3, OUTPUT);
    digitalWrite(3, LOW);
    pinMode(4, INPUT);
}

void loop()
{

```







# Assignment #1

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



*For your attention*