



01076001

วิศวกรรมคอมพิวเตอร์เบื้องต้น

Introduction to Computer Engineering

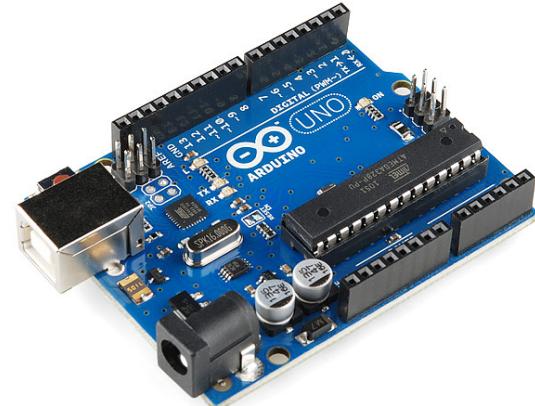
Arduino #1

LED, Digital Output, Digital Input, Switch

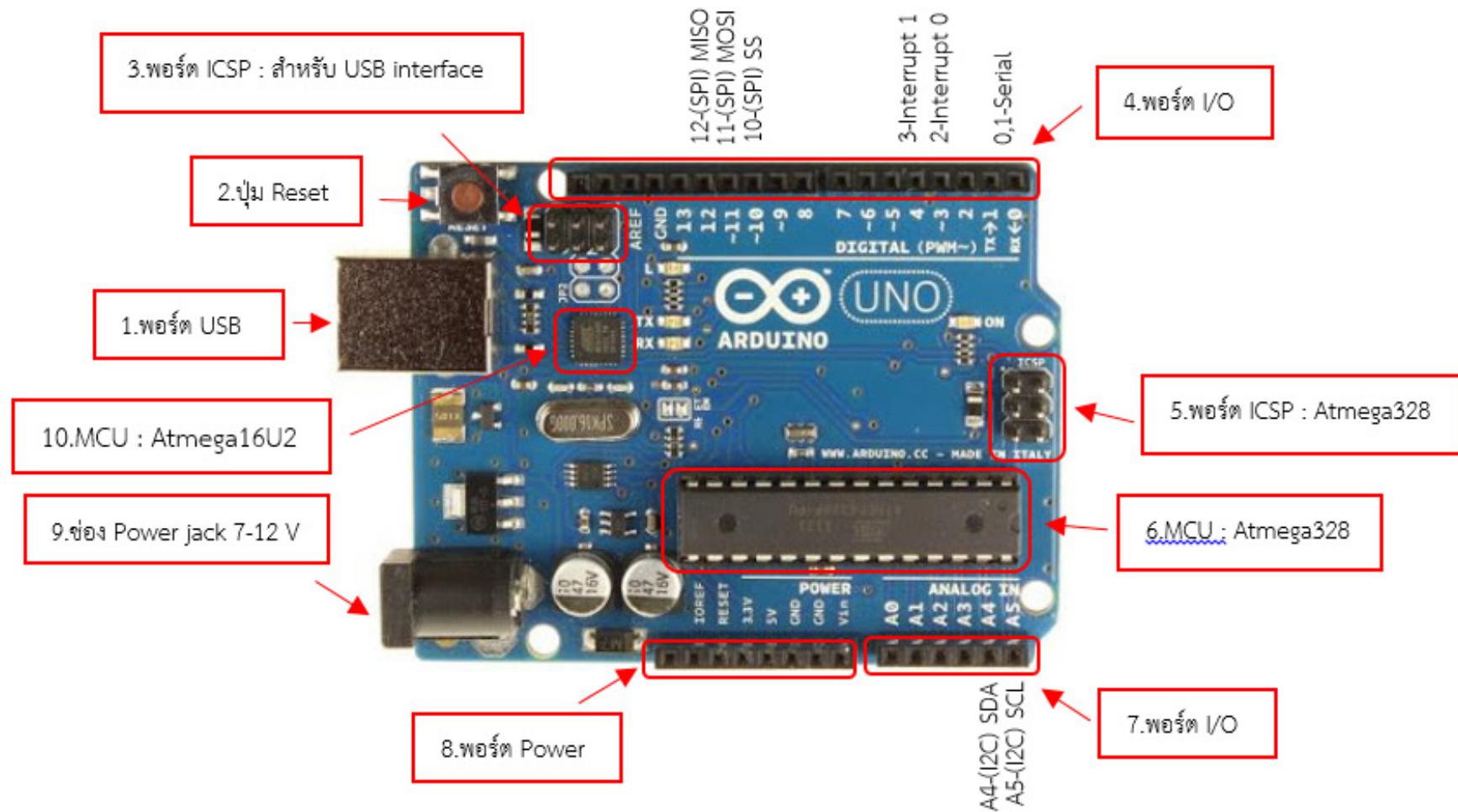
# บอร์ด Arduino



- อ่านว่า (อา-ดู-อิ-โน่ หรือ อาดูยโน่)
- เป็นบอร์ด ไมโครคอนโทรลเลอร์
- มีการใช้งานที่ง่าย
- มี Library มาก
- มีอุปกรณ์ที่นำไปเชื่อมต่อมาก
- ทำให้ได้รับความนิยมอย่างมาก



# องค์ประกอบของ Arduino



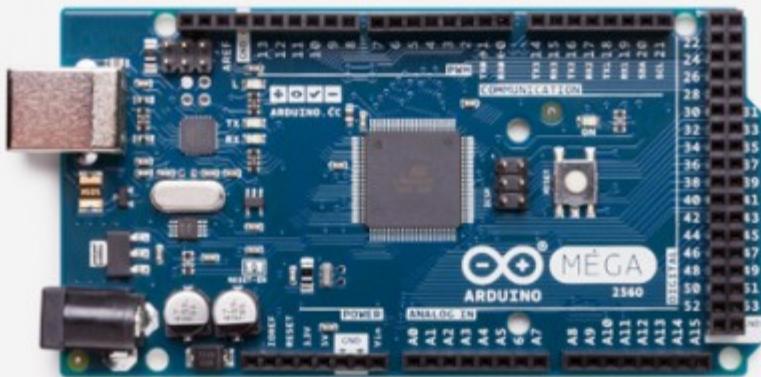
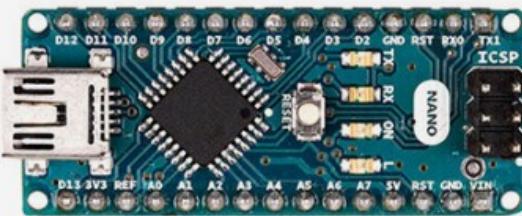


# Arduino Technical Specs

## 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

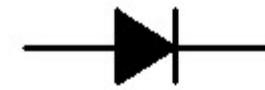
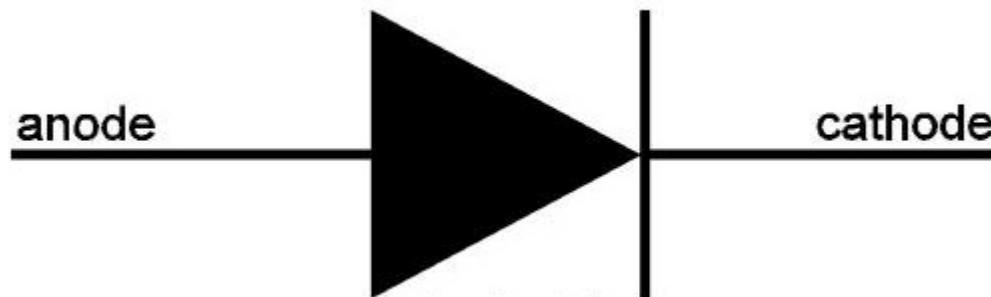
# Arduino Board



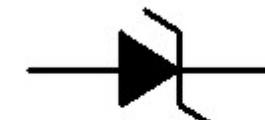
# ໄດໂອດ



- ใช้ในการควบคุมทิศทางการไหลของกระแสไฟฟ้า คือ ໄດໂອດ จะยอมให้ กระแสไฟฟ้า ไหลผ่าน ในทิศทางเดียว (ไปอีกทาง) และกันการไหลในทิศทางตรงกันข้าม (ไปอีกย้อนกลับ)



Diode



Zener Diode

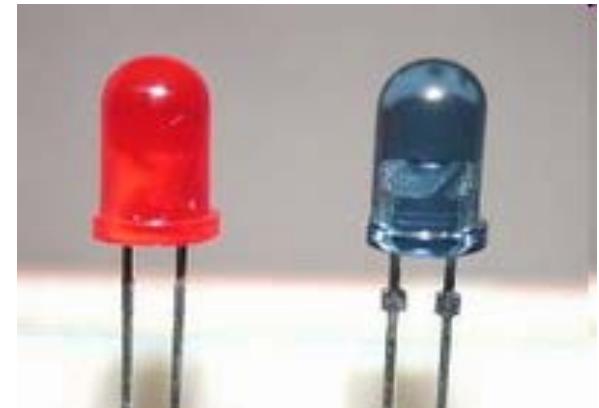


Light Emitting Diode - LED



# ประเภทไอดีโอด (บางส่วน)

- ไอดีโอดเรกติฟาย (Rectifier Diode) ใช้ในวงจรเรียงกระแส (Rectifier) หรือวงจรแปลงไฟสลับเป็นไฟตรง
- ไอดีโอดเปล่งแสง (Light Emitting Diode) มักเรียกว่า แอลอีดี (LED) เป็นไอดีโอดที่ออกแบบมาเป็นพิเศษโดยเมื่อได้รับแรงดันไฟอัลตร้าจะเปล่งแสงออกมากได้ซึ่งมีทั้งสีแดง เขียว ฟ้า ส้ม เหลือง ฯลฯ



# ประเภทไดโอด (บางส่วน)



- ซีเนอร์ไไดโอด (Zener Diode) จะยอมให้กระแสไฟลนสภาวะไปอัลล์บ หากมีแรงดันมากกว่าค่าที่กำหนด ใช้ในการรักษาแรงดันแรงดันไฟฟ้าต่ำคร่อมตัวมันให้มีค่าคงที่
- ไดโอดรับแสง (Photo Diode) จะยอมให้กระแสไฟลนสภาวะไปอัลล์บ ตามแสงที่มากระทบ ซึ่งอาจเป็นแสงสว่างธรรมชาติหรือแสงอินฟารेड มักใช้กับวงจรตรวจจับสัญญาณ หรือในวงจรควบคุมระยะใกล้ หรือที่เรียกว่า รีโมท

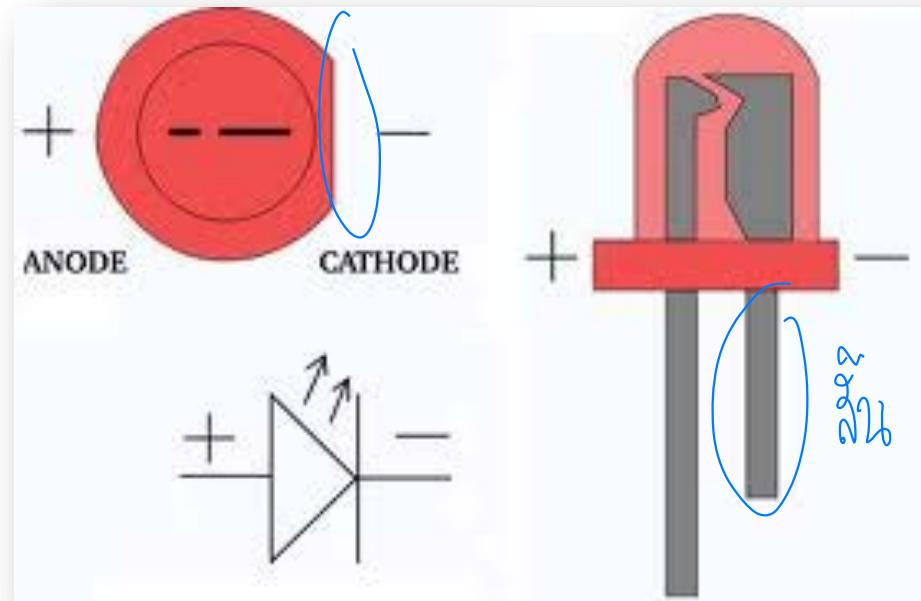


# ໄດໂອດເປົ່າງແສງ ແອລອືດີ (LIGHT EMITTING DIODES)



- LED เป็นไดโอดที่ใช้สารประเภทแกลเลียมอาร์เซนิคในต์ฟอสไฟต์ (Gallium Arsenide Phosphide ; GaAsP) หรือสารแกลเลียมฟอสไฟต์ (Gallium Phosphide ; GaP) มาทำเป็นสารกึ่งตัวนำชนิด P และ N แทนสาร Si และ Ge สารเหล่านี้มีคุณลักษณะพิเศษ คือ สามารถเรืองแสงได้เมื่อได้รับใบอัสตรอง การเกิดแสงที่ตัว LED นี้เราเรียกว่า อิเล็กโทรลูมินิเซนต์ (Electroluminescence) ปัจจุบันนิยมใช้ LED แสดงผลในเครื่องมืออิเล็กทรอนิกส์ เช่น เครื่องคิดเลข, นาฬิกา เป็นต้น

# การดูขา LED





# Forward Voltage

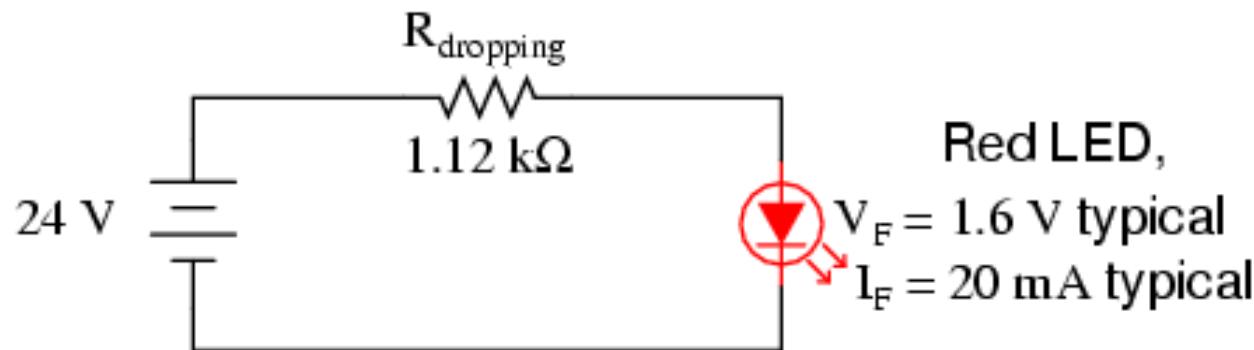
- เป็นแรงดันที่เหมาะสมในการทำงานของ LED
- LED แต่ละชนิด จะไม่เท่ากัน
- ในการใช้งานต้องสร้างแรงดันตามที่ LED ต้องการ
- ทั่วไปใช้ประมาณ 2V

Color	Material	Wavelength (nm)	V-forward
Super Red	GaAlAs	660	1.8
Green	GaP	565	2
Red	GaAsP	635	2
Red	AlInGaP	636	2
Orange	AlInGaP	610	2
Yellow	AlInGaP	590	2
Amber	GaAsP	605	2.1
Red	GaP	700	2.1
Green	GaP	555	2.1
Green	AlInGaP	574	2.2
Blue	SiC	430	3.5
Green	InGaN	505	3.5
Blue	InGaN	470	3.5
White	InGaN		3.5
Green	InGaN	525	3.7
Green	InGaN	525	4
Blue	SiC	430	4.5



# การหาค่า R สำหรับต่อ LED

- เป็น R อนุกรมที่ใช้จำกัดกระแสที่จะไหลผ่าน LED โดยหัวไปกระแสสูงสุดที่ใช้จะไม่เกิน 20 mA หากเกินหลอดอาจชำรุดได้
- โดยใช้หลักการแบ่งแรงดันจะใช้สูตร  $R = \frac{V - V_{LED}}{I}$



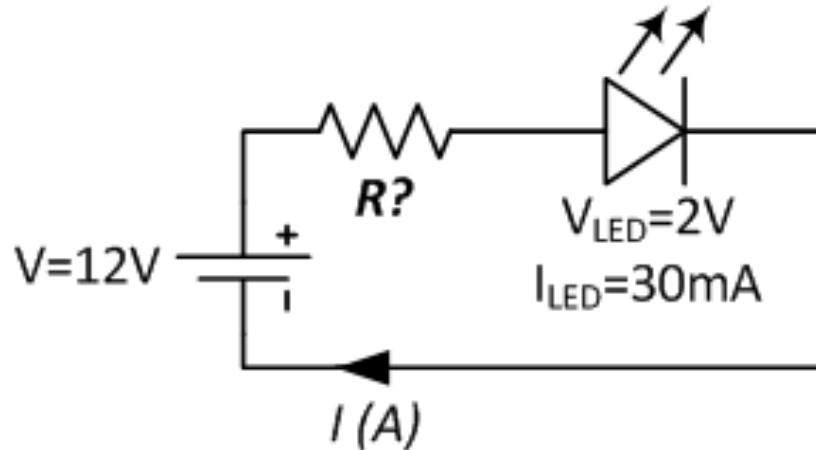
- $R = (24 - 1.6) / 0.02 = 1.12 \text{ k}\Omega$  แต่ R ค่านี้ หาซื้อยาก จึงใช้ค่า 1.2 kΩ แทน

# Common Resistor Value



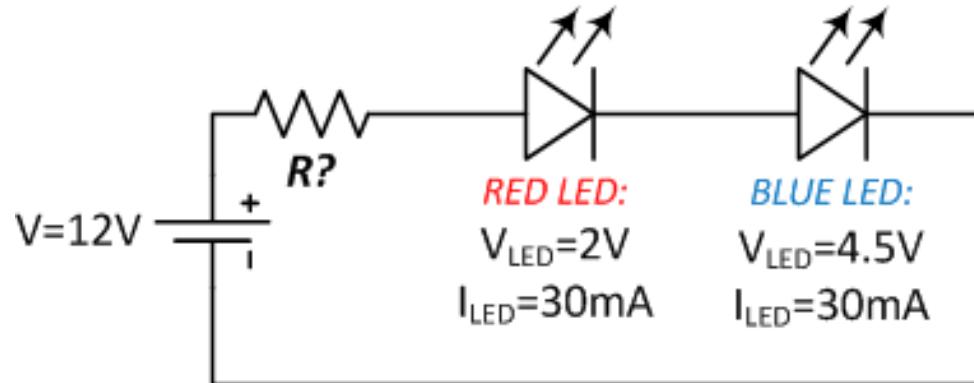
Standard Resistor Values ( $\pm 5\%$ )						
1.0	10	100	1.0K	10K	100K	1.0M
1.1	11	110	1.1K	11K	110K	1.1M
1.2	12	120	1.2K	12K	120K	1.2M
1.3	13	130	1.3K	13K	130K	1.3M
1.5	15	150	1.5K	15K	150K	1.5M
1.6	16	160	1.6K	16K	160K	1.6M
1.8	18	180	1.8K	18K	180K	1.8M
2.0	20	200	2.0K	20K	200K	2.0M
2.2	22	220	2.2K	22K	220K	2.2M
2.4	24	240	2.4K	24K	240K	2.4M
2.7	27	270	2.7K	27K	270K	2.7M
3.0	30	300	3.0K	30K	300K	3.0M
3.3	33	330	3.3K	33K	330K	3.3M
3.6	36	360	3.6K	36K	360K	3.6M
3.9	39	390	3.9K	39K	390K	3.9M
4.3	43	430	4.3K	43K	430K	4.3M
4.7	47	470	4.7K	47K	470K	4.7M
5.1	51	510	5.1K	51K	510K	5.1M
5.6	56	560	5.6K	56K	560K	5.6M
6.2	62	620	6.2K	62K	620K	6.2M
6.8	68	680	6.8K	68K	680K	6.8M
7.5	75	750	7.5K	75K	750K	7.5M
8.2	82	820	8.2K	82K	820K	8.2M
9.1	91	910	9.1K	91K	910K	9.1M

# Activity



- คำนวณค่า  $R$  ที่เหมาะสมและหาซึ่งได้สำหรับวงจรข้างต้น
- หากเปลี่ยน  $V=5V$  ค่า  $R$  ที่เหมาะสมเป็นเท่าไร

# Activity



- คำนวณค่า  $R$  ที่เหมาะสมสำหรับวงจรข้างต้น
- หาก LED เป็นสีแดงทั้งหมด จะใช้ LED กี่ตัว จึงไม่ต้องต่อ  $R$  เลย

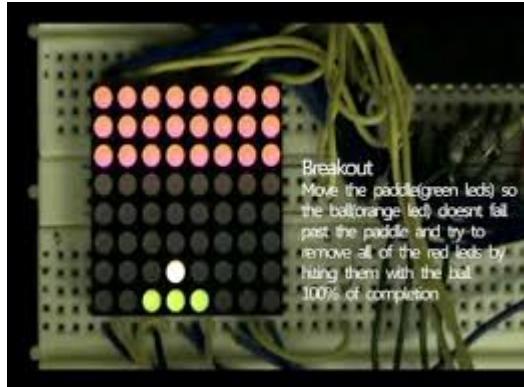
# Activity



ห้องน้ำ

- ถ้าจะทำป้ายไฟโดยมี LED ประมาณ 100 ดวง จะต่อวงจรอย่างไร

# LED Dot Matrix



# ติดตั้ง Arduino IDE



- ดาวน์โหลด Arduino จากเว็บ

The screenshot shows the official Arduino website's header. The top navigation bar includes links for HOME, BUY, SOFTWARE (which is circled in red), PRODUCTS, LEARNING, FORUM, SUPPORT, and BLOG. The main content area features a section titled "WHAT IS ARDUINO?" with an image of an Arduino Uno board. Below this are three buttons: "BUY AN ARDUINO", "LEARN ARDUINO", and "DONATE". To the right, there's a "BLOG" section showing a close-up of electronic components on a breadboard. Further right is the "ARDUINO EDUCATION" section, which includes the text "REDEFINING THE LEARNING EXPERIENCE ONE CLASSROOM AT A TIME" and a decorative graphic of colored squares.

# ติดตั้ง Arduino IDE



## Download the Arduino IDE



### ARDUINO 1.8.5

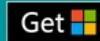
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

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 \(sha512\)](#)

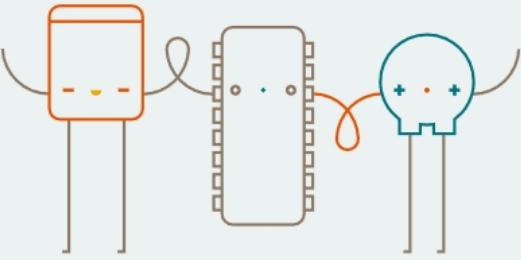
# ติดตั้ง Arduino IDE



- 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**

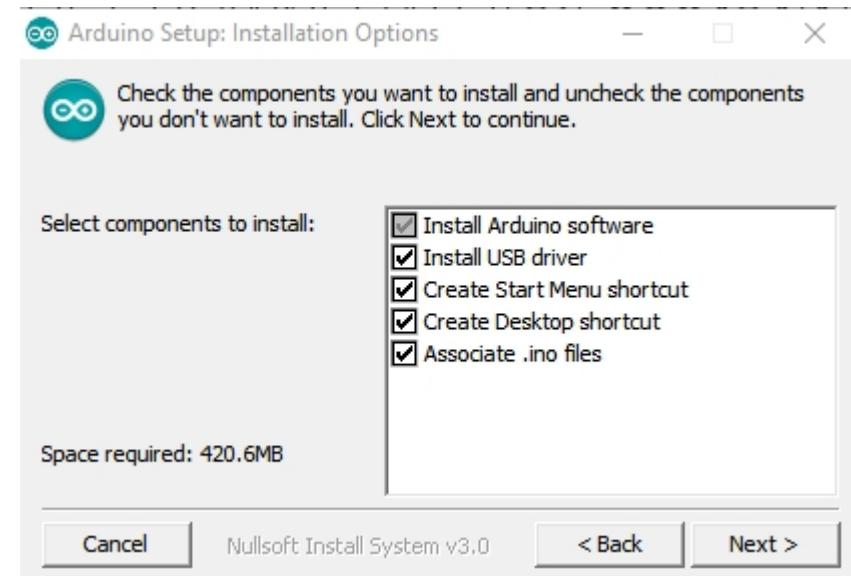
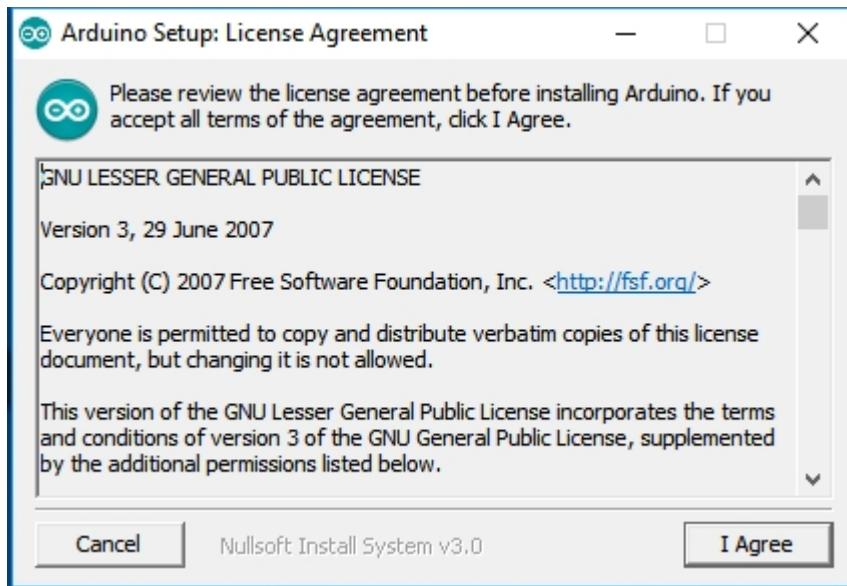
JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

# ติดตั้ง Arduino IDE



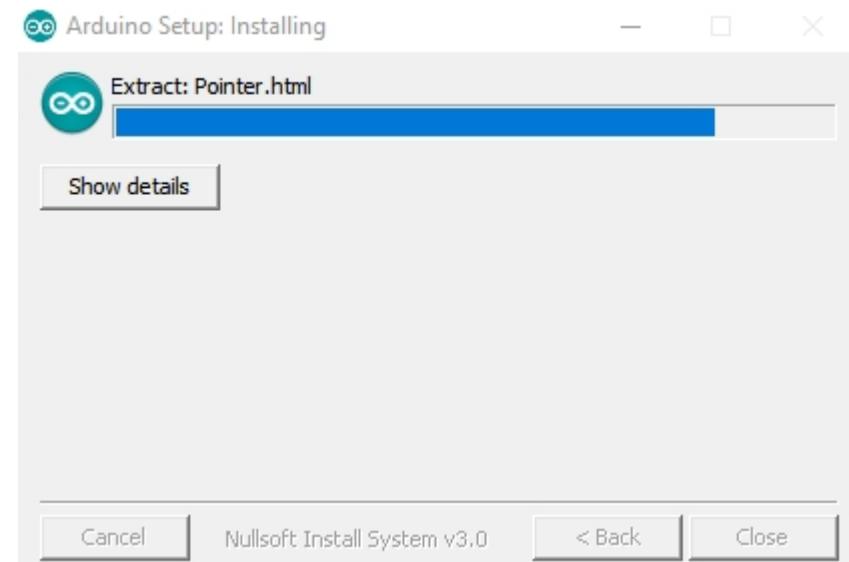
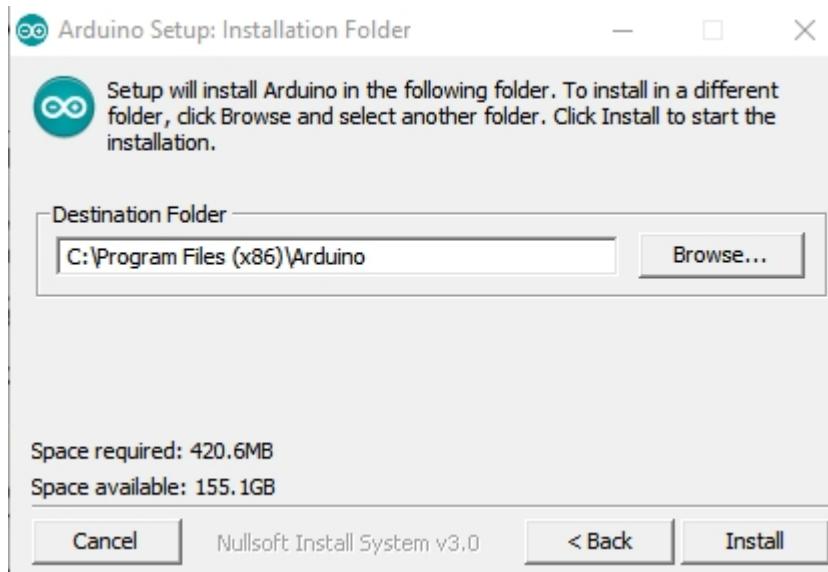
- คลิก I Agree และ Next ตามลำดับ



# ติดตั้ง Arduino IDE



- คลิก Install



# ติดตั้ง Arduino IDE



- เรียกขึ้นมาทำงาน

The screenshot shows the Arduino IDE interface. The title bar reads "sketch\_sep08a | Arduino 1.8.2". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for save, upload, and other functions. The main area displays the following code:

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

}

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

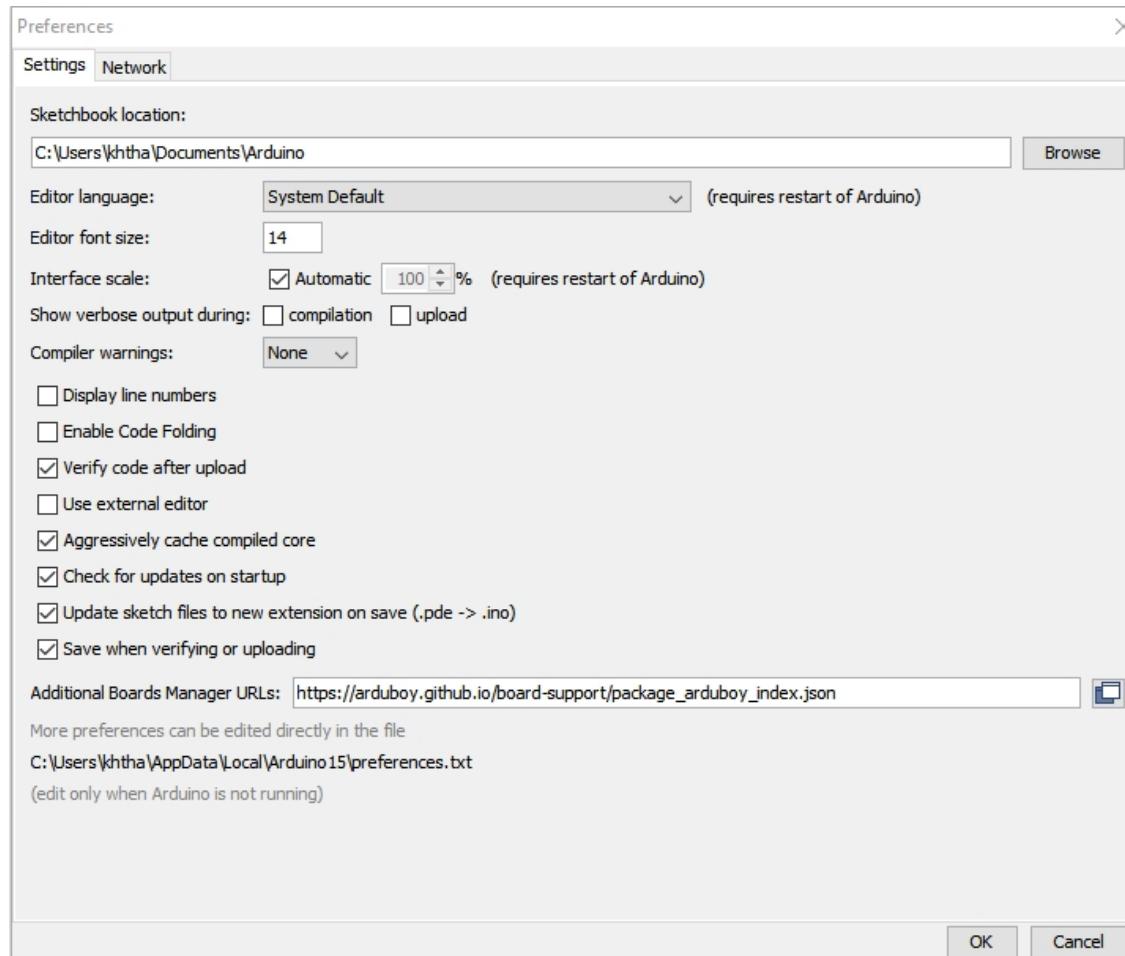
}
```

At the bottom of the IDE window, there is a status bar with the text "Arduino/Genuino Uno on COM10".

# តែងគោ preferences



- គ្រឿងទី File -> Preferences



# ตั้งค่า preferences

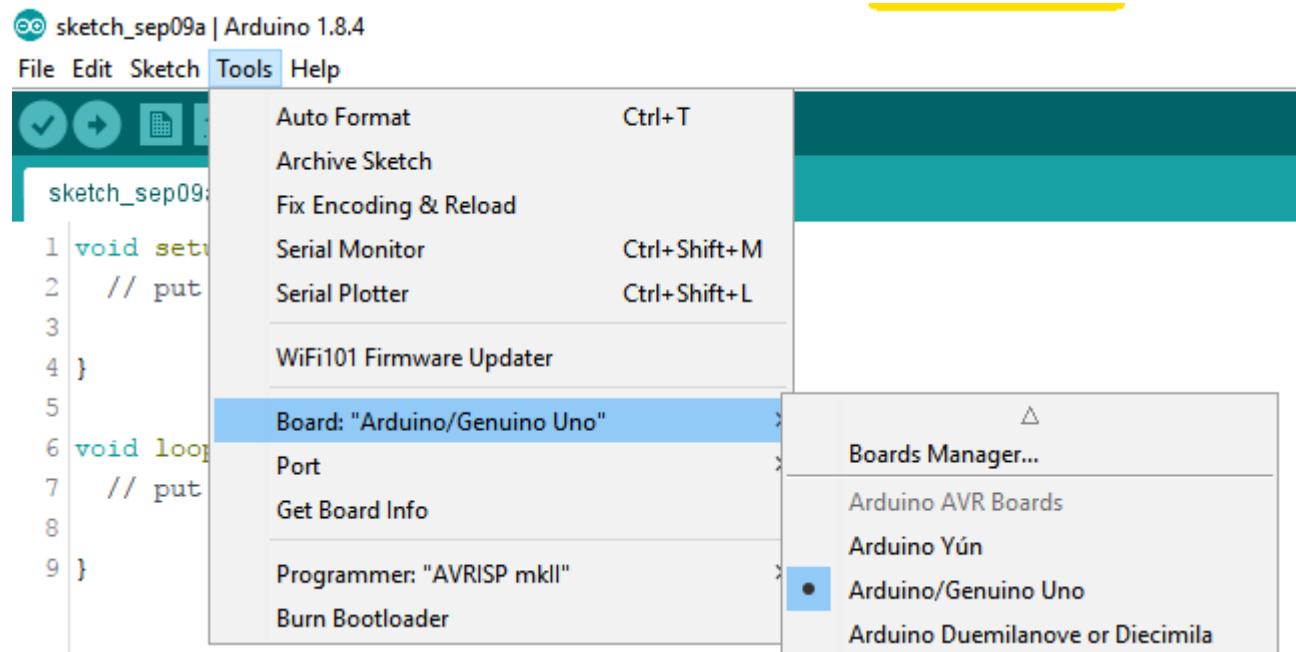


- กำหนดจุดที่ใช้เก็บโปรแกรม
- ขนาดพอนต์
- Show verbose output during จะให้แสดงรายละเอียดระหว่างที่ Compile หรือ Upload หรือไม่
- แสดง Warning หรือไม่
- แสดงเลขที่บรรทัด
- Check for updates on startup
- Save when verifying or uploading

# กำหนด Board และ Port



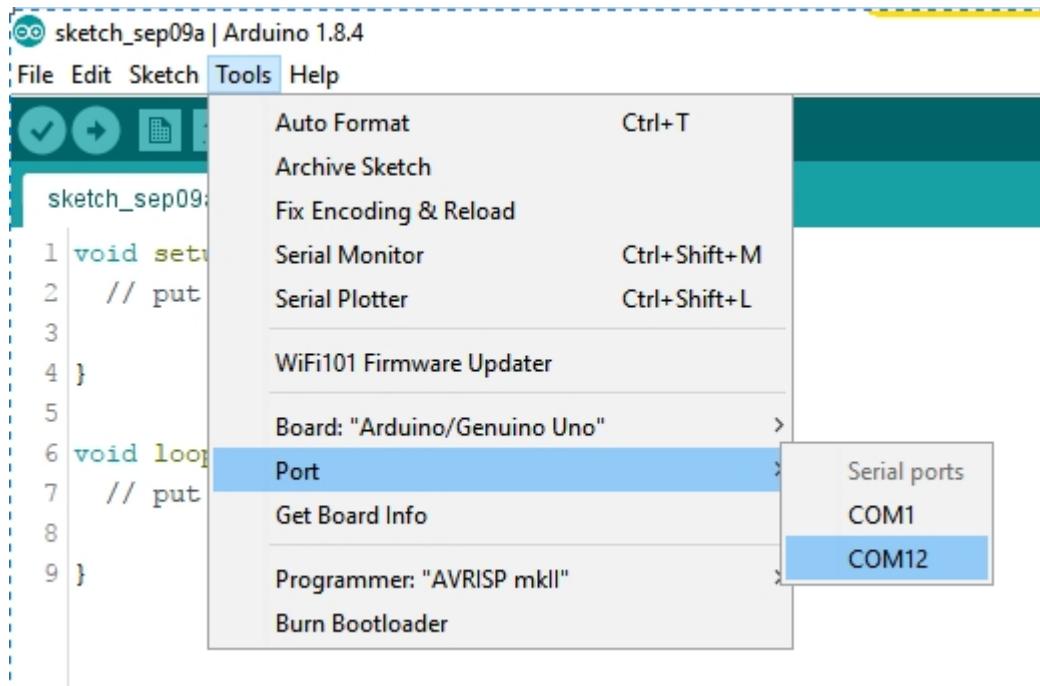
- เสียบ USB ระหว่างเครื่องคอมพิวเตอร์กับบอร์ด
- ไปที่ Tools -> Board เลือก Arduino/Genuino Uno



# กำหนด Board และ Port



- เสียบ USB ระหว่างเครื่องคอมพิวเตอร์กับบอร์ด
- เลือกพอร์ต (ขึ้นอยู่กับเครื่อง)



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

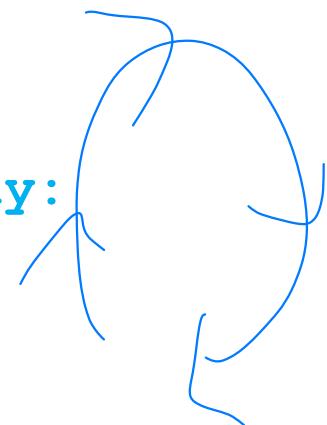
# Arduino Sketch



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

}

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





# 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.

↳ ๑๕(๑)



# 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 ()** พังก์ชันที่ส่งค่าจำนวน มิลลิวินาที นับจากที่โปรแกรมเริ่มรัน

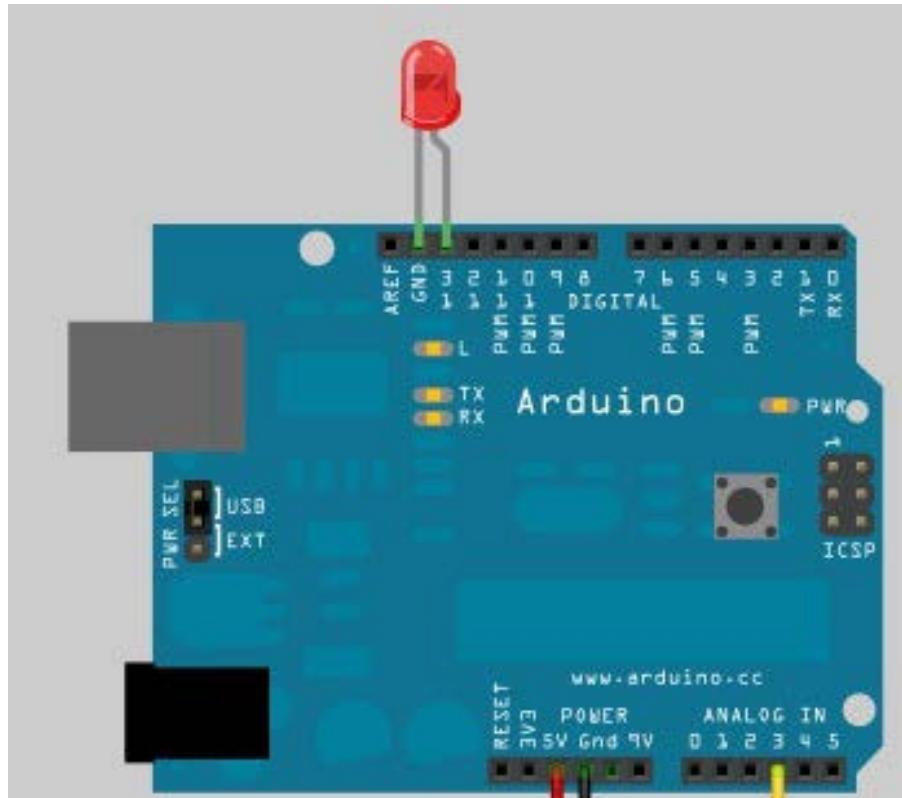
```
unsigned long time;

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);
}
```

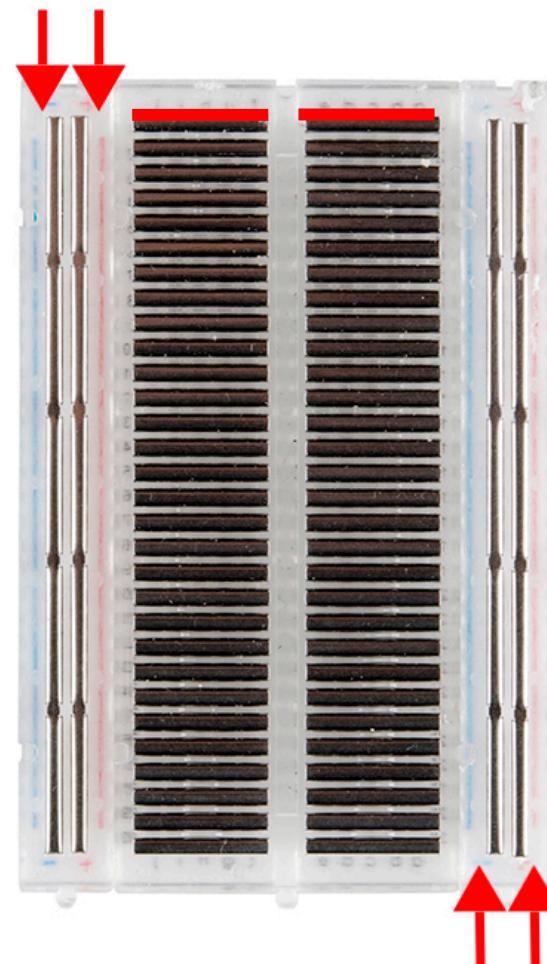
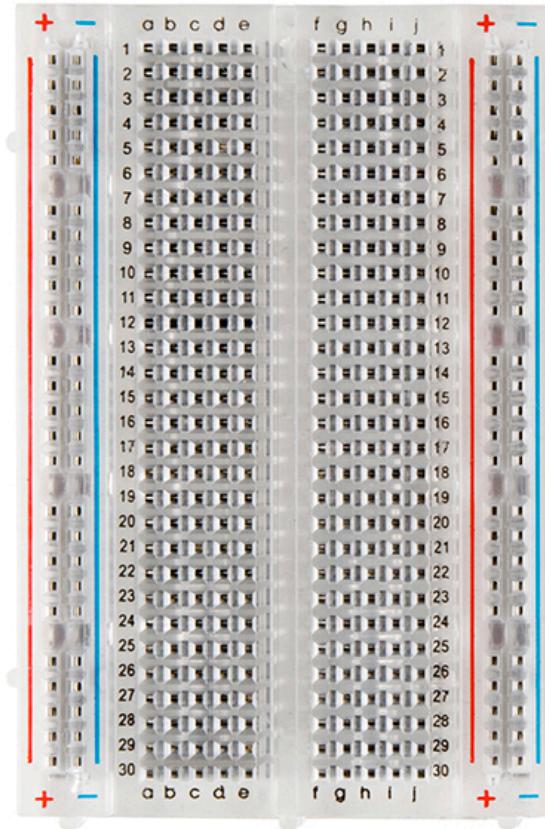
# Activity



- ให้หน้าหลอด LED มาต่อ กับบอร์ด โดยให้ขา + ต่อกับขา 13 และขาลบต่อกับ GND ดังรูป แล้วรันโปรแกรมอีกครั้ง



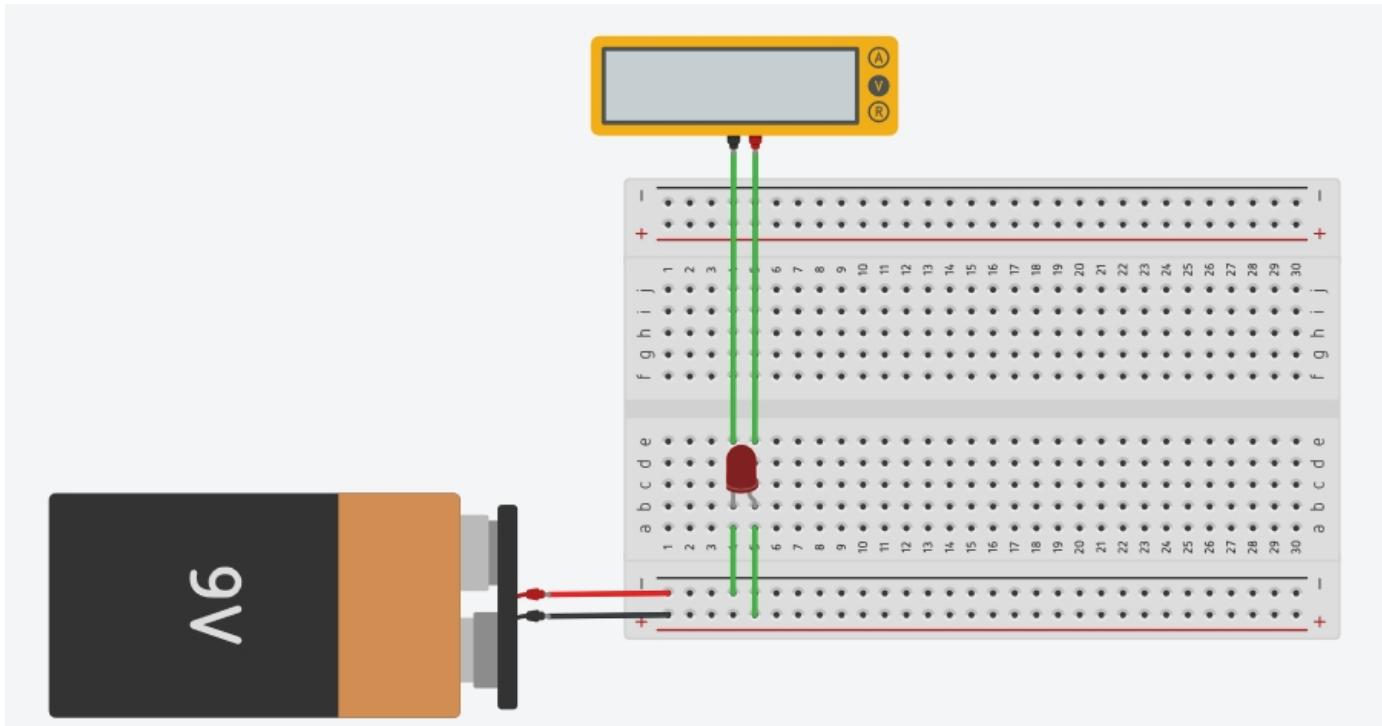
# Protoboard หรือ Breadboard



# Activity



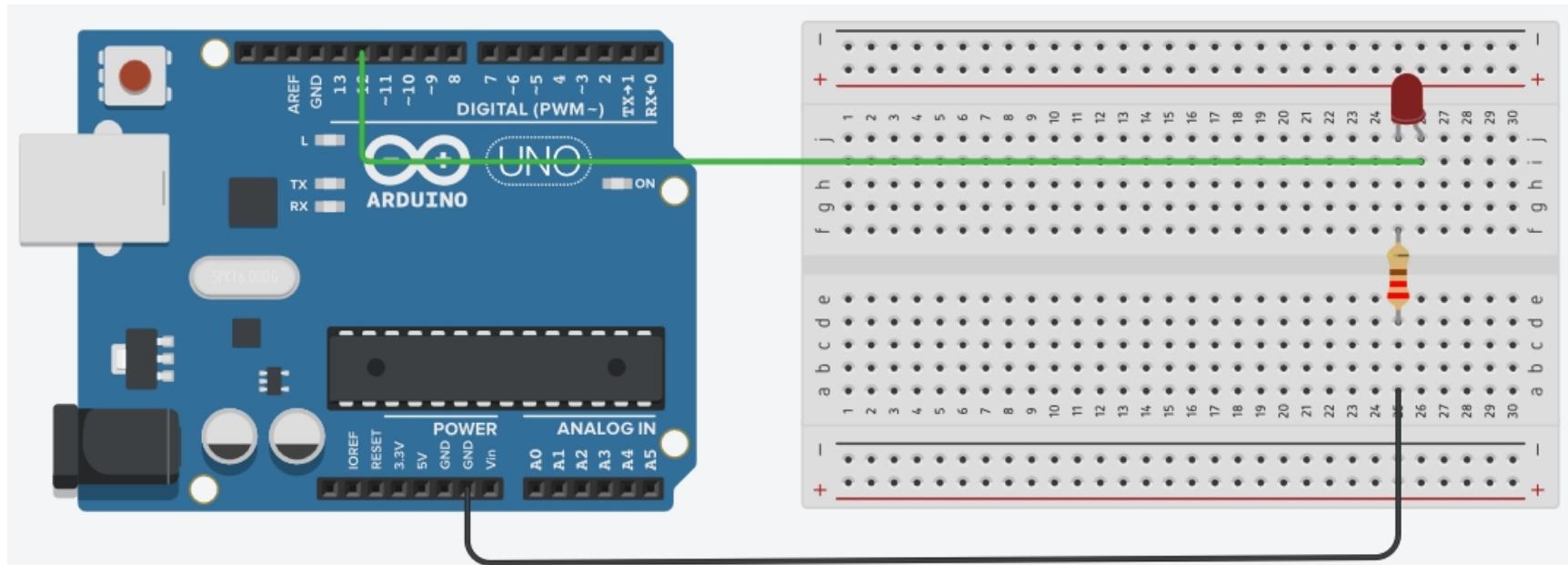
- ทดลอง Protoboard บน [www.tinkercad.com](http://www.tinkercad.com)
- LED สว่างหรือไม่ ให้แก่ไขให้ LED สว่าง



# Activity



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





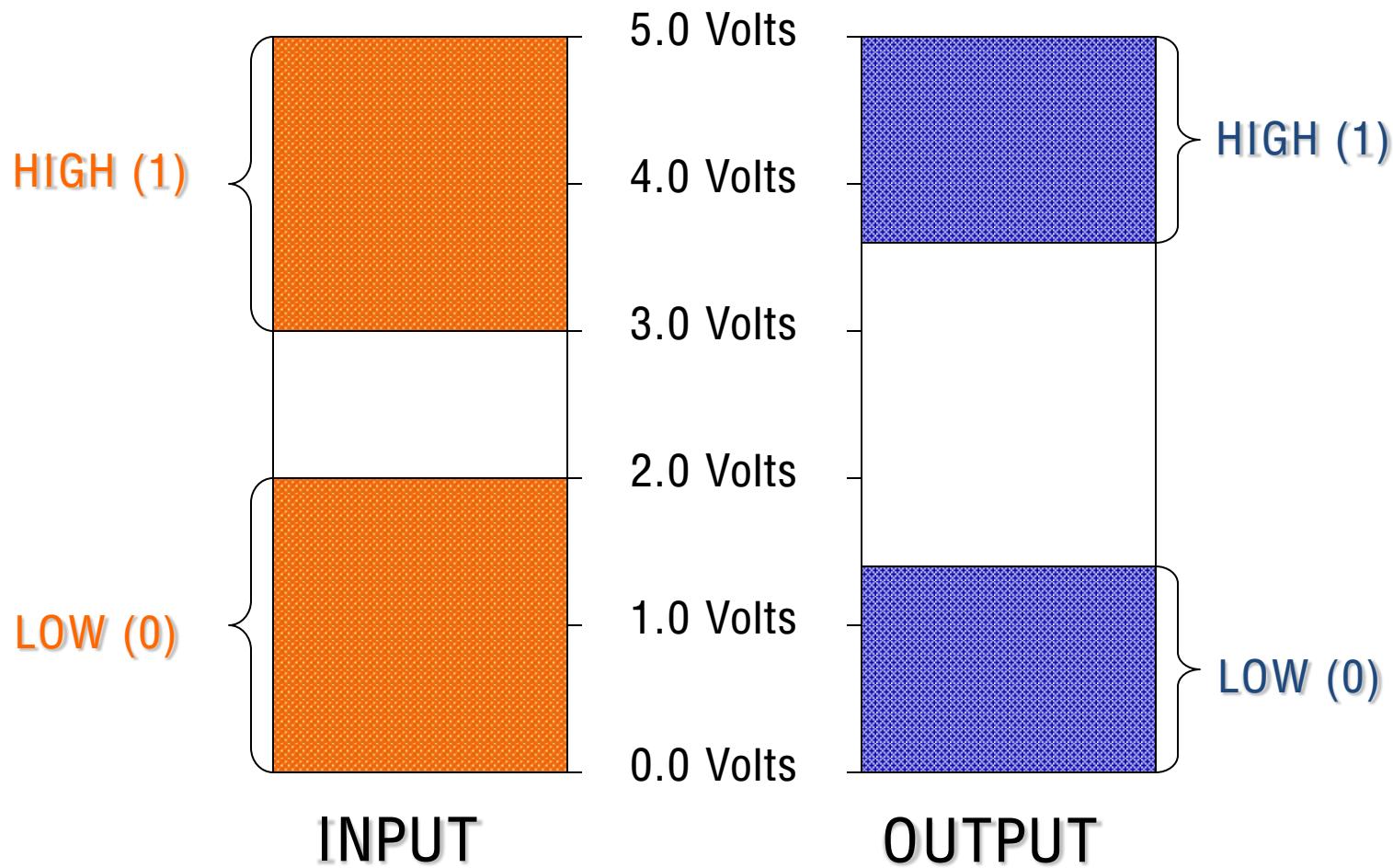
## คำถาม

- Logic 1 มีค่ากี่โวลท์ และ Logic 0 มีค่ากี่โวลท์

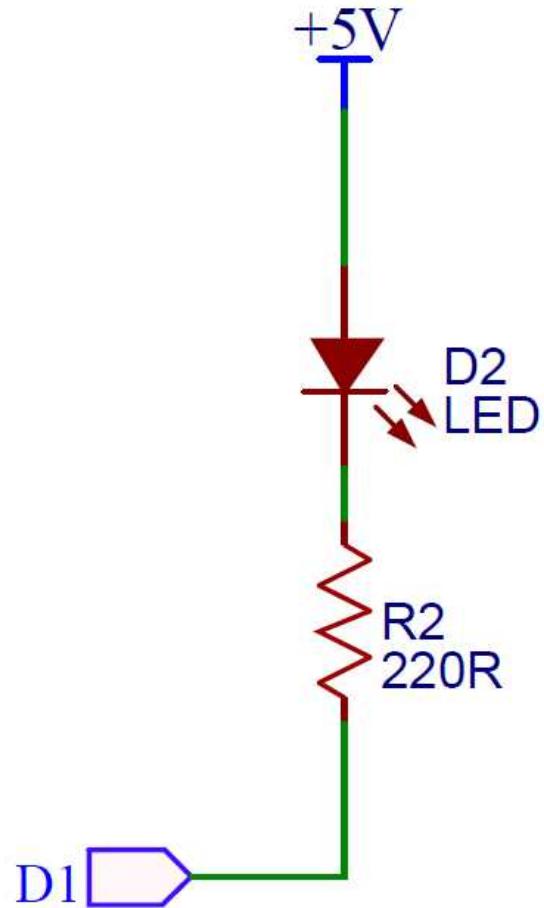
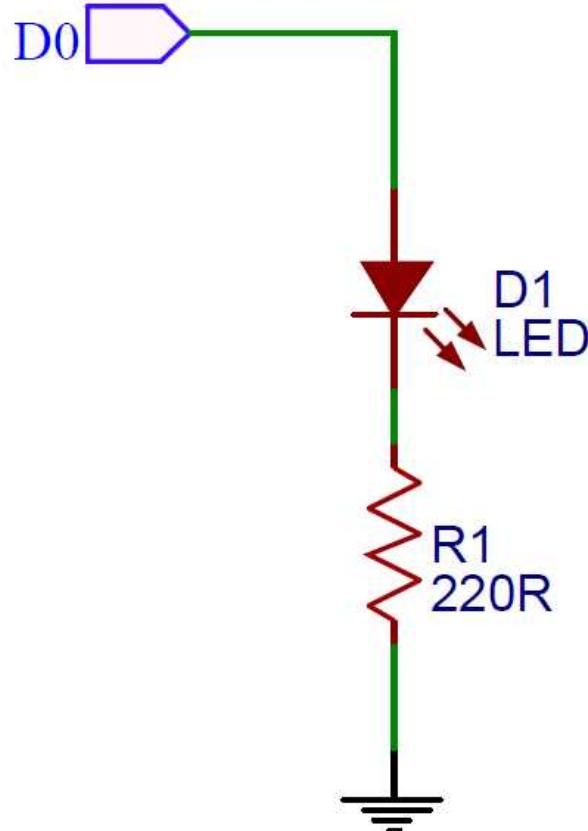




# Voltage Range of Binary Signals



# Positive Logic Interface & Negative Logic Interface



# Activity

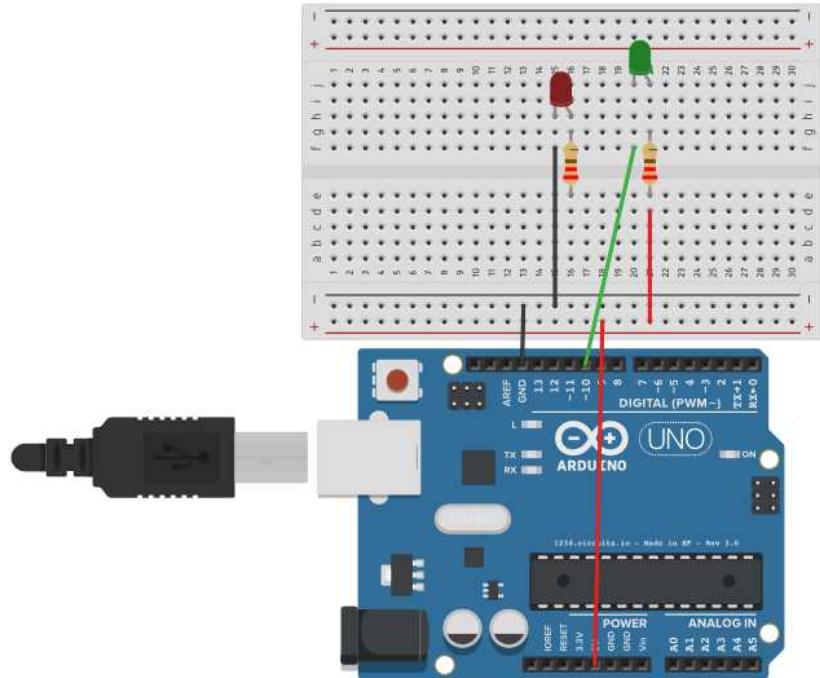


- ให้ทดลองต่อ LED แบบ Positive Logic และ Negative Logic
- ทดลองต่อ LED นาน, อนุกรม เพิ่ม
- ตอบคำถามว่า ในการใช้งานควรใช้แบบ Positive Logic หรือ Negative Logic

# Activity



- ต่อวงจรโดยใช้ LED สีแดง (positive logic) และสีเขียว (negative logic) อย่างละ 1 ดวง
- เขียนโปรแกรม ให้แสดงดังนี้
  - เริ่มต้นดับทั้งหมด
  - สีแดงติด 3 ครั้ง 3 วินาที
  - สีเขียวติด 2 ครั้ง 2 วินาที
  - ติดสลับกัน 1 ครั้ง
  - กลับไปวน loop ใหม่





# Switch and Pullup Pulldown



Top



Front



Side



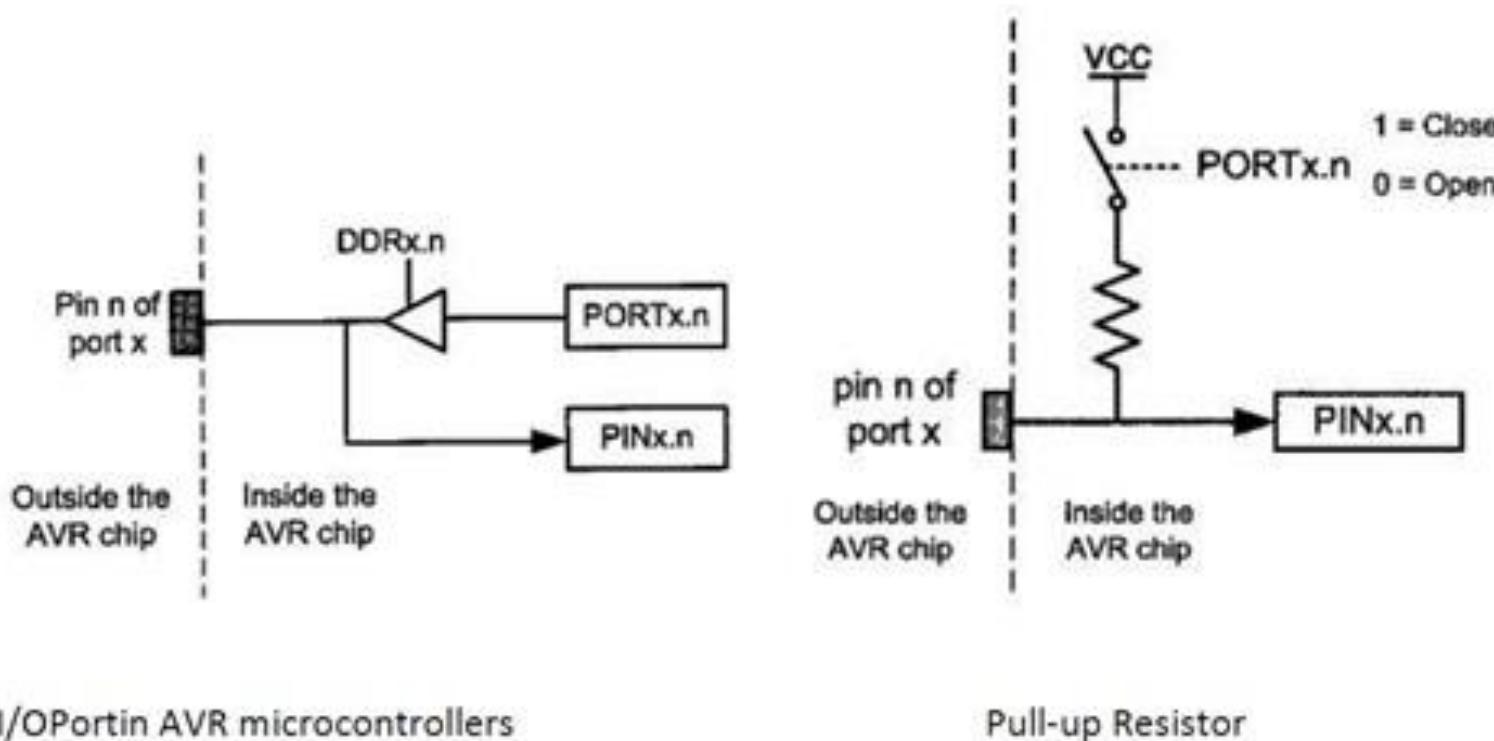
Released



Pressed



# Digital Input



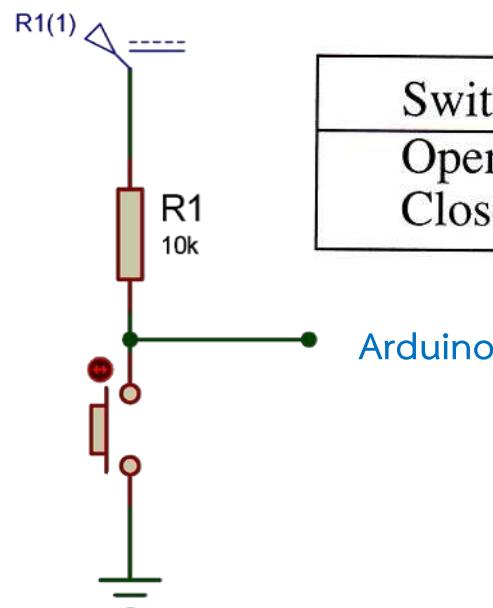
I/O Port in AVR microcontrollers

Pull-up Resistor

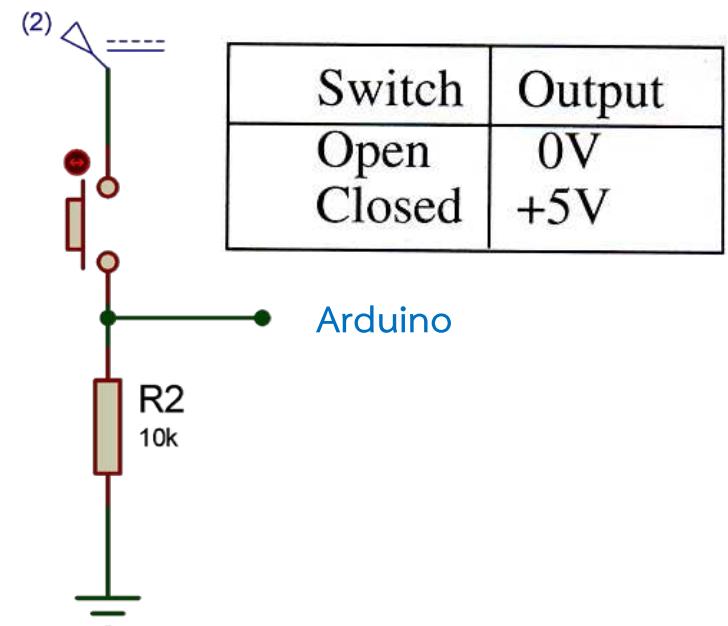


# Digital Input

## Pull Up



## Pull Down





# Digital Input

## SETUP PINMODE

Syntax:

pinMode(pin, mode)

Parameter:

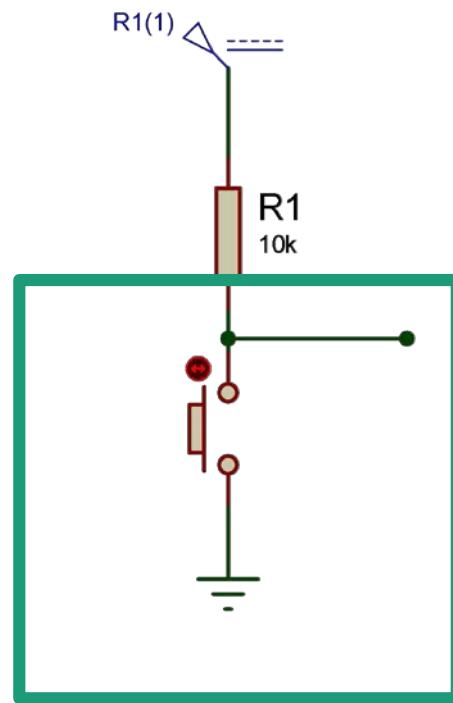
pin: the number of the pin whose mode you wish to set

mode: INPUT, OUTPUT or INPUT\_PULLUP.



# Digital Input

## INPUT\_PULLUP





# 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:

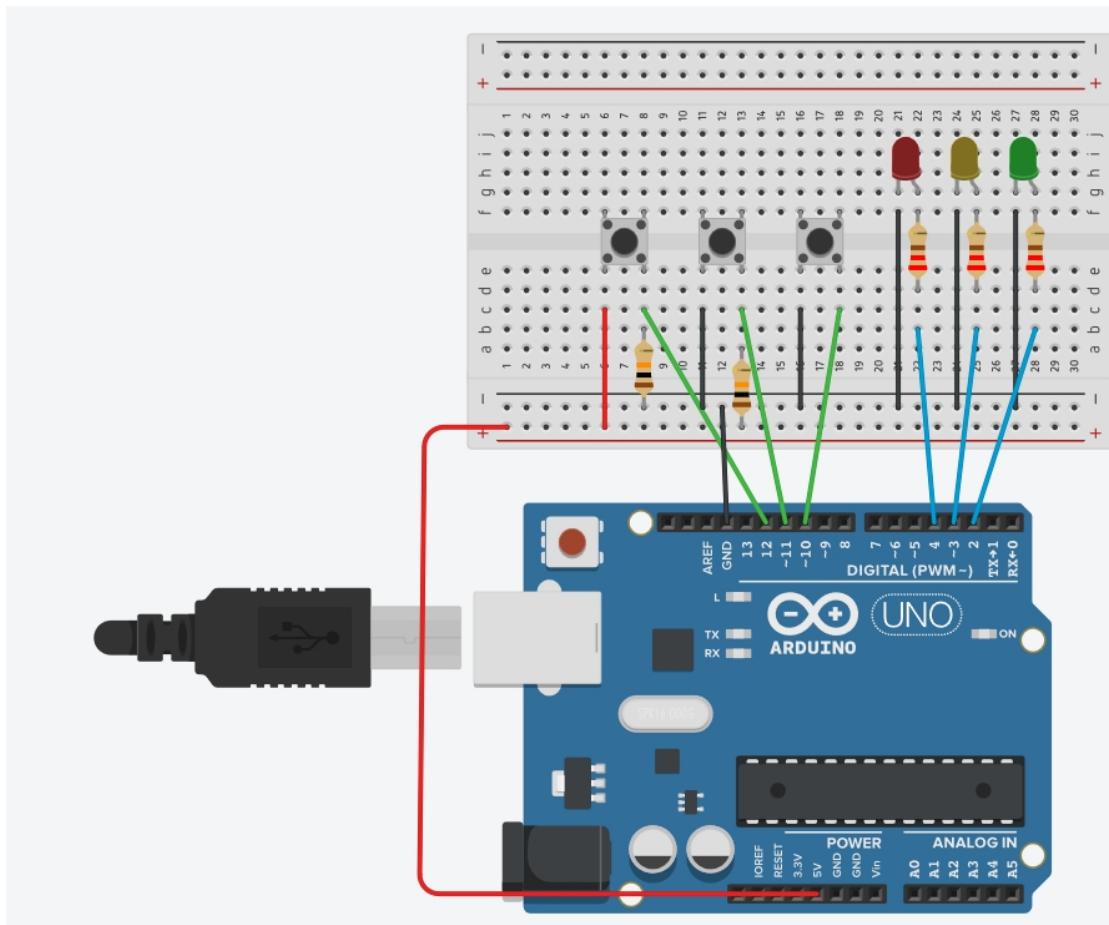
```
#define button 2      // switch input Active Low  
#define pressed LOW
```

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

# Activity



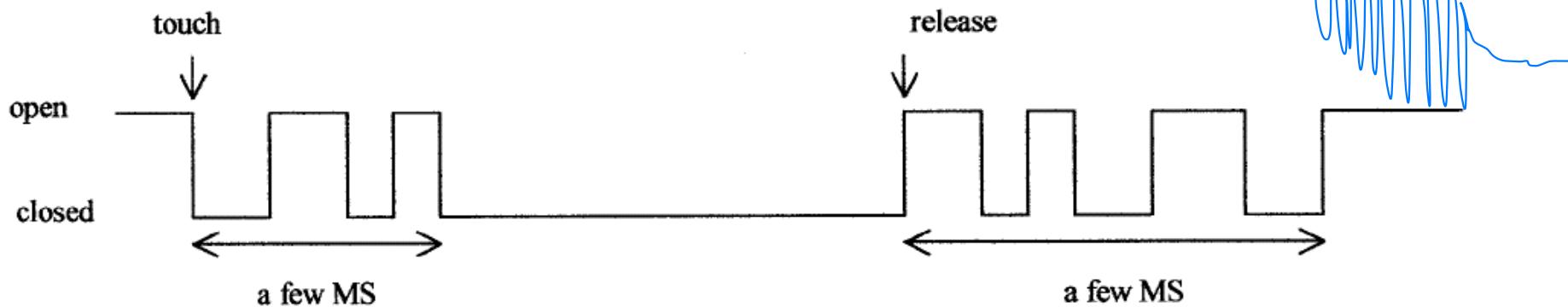
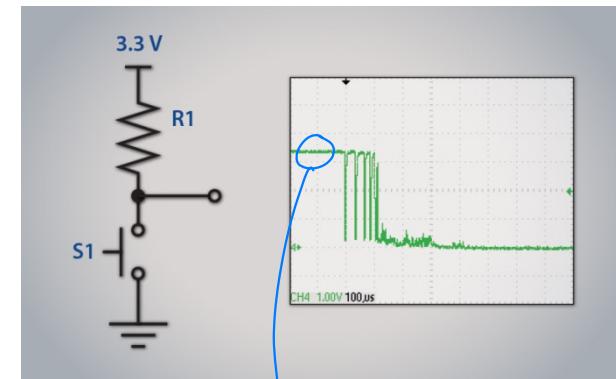
- ต่อวงจรโดยใช้ Switch
    - Pull Up
    - Pull Down
    - Internal Pullup
  - กดปุ่มในให้ไฟติด  
(ไม่ต้องกดค้าง)





# Bounce Problem

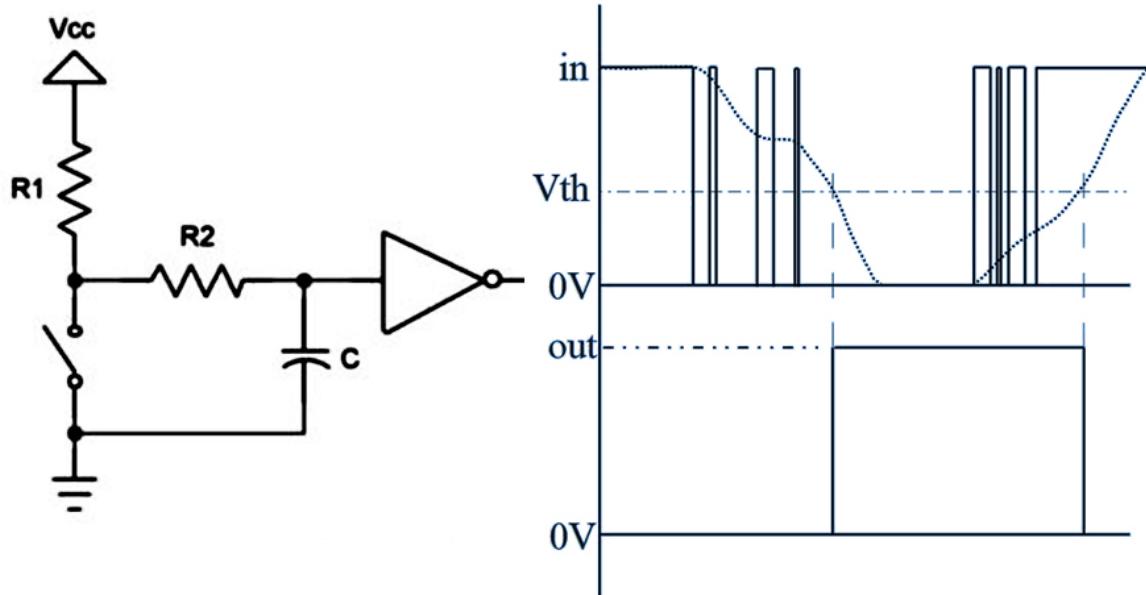
- ในการกดสวิตซ์ 1 ครั้ง จะมีช่วงเวลาสั้นๆ ที่เกิดสัญญาณคล้ายกับการกดสวิตซ์หลายครั้ง เนื่องจากหน้าสัมผัส
- การแก้ไข
  - สวิตซ์บางตัวที่มีราคาแพงจะไม่เกิด Bounce
  - แก้ไขโดยวิธีการทางฮาร์ดแวร์
  - แก้ไขโดยวิธีการทางซอฟต์แวร์





# Bounce Problem

- **Hardware De-bouncing**
  - ใช้วงจรยาร์ดแวร์ในการกำจัดการ 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

ด้วย `delay()` (delay การทำซ้ำ)



# 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:

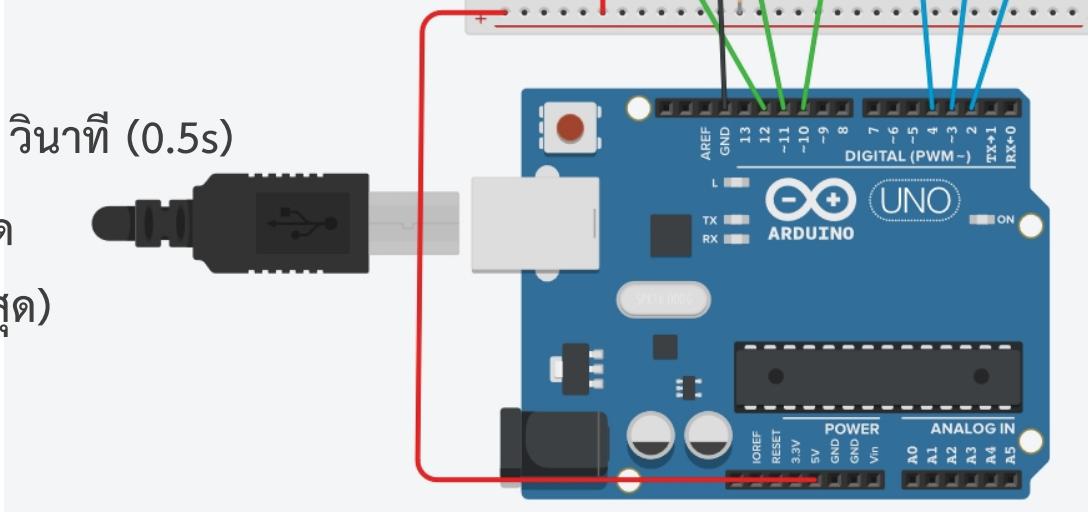
        if (reading != buttonState) {
            buttonState = reading;
        }
    }
    lastButtonState = reading;
}
```

ms  $\approx$  millis

# Assignment #1



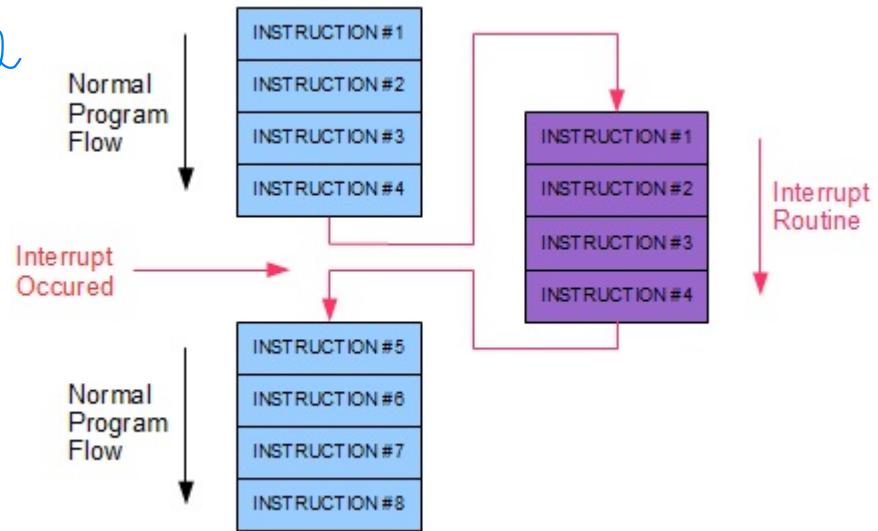
- เขียนโปรแกรม ให้แสดงดังนี้
  - กดปุ่มขวา เขียวติด 3 วินาทีแล้วดับ
  - กดปุ่มซ้าย แดงติด 3 วินาทีแล้วดับ
  - ถ้ากดปุ่มแดงหรือเขียวซ้ำ ให้ดับ
  - แม้จะเขียว แต่ถ้ากดซ้าย แดงต้องติด
  - ถ้าแดงอยู่ กดขวา ไม่มีผล
  - กดปุ่มกลาง เหลืองกระพริบ 2 วินาที (0.5s)
  - ปุ่มกลางจะมีผลเมื่อไฟอื่นไม่ติด  
(ปุ่มกลางมีความสำคัญน้อยที่สุด)





# 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)
```

↓ ពីរការស្នើសោរណ៍

mode និងតម្លៃរបៀប

## Parameter:

**interrupt** : the number of the interrupt

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

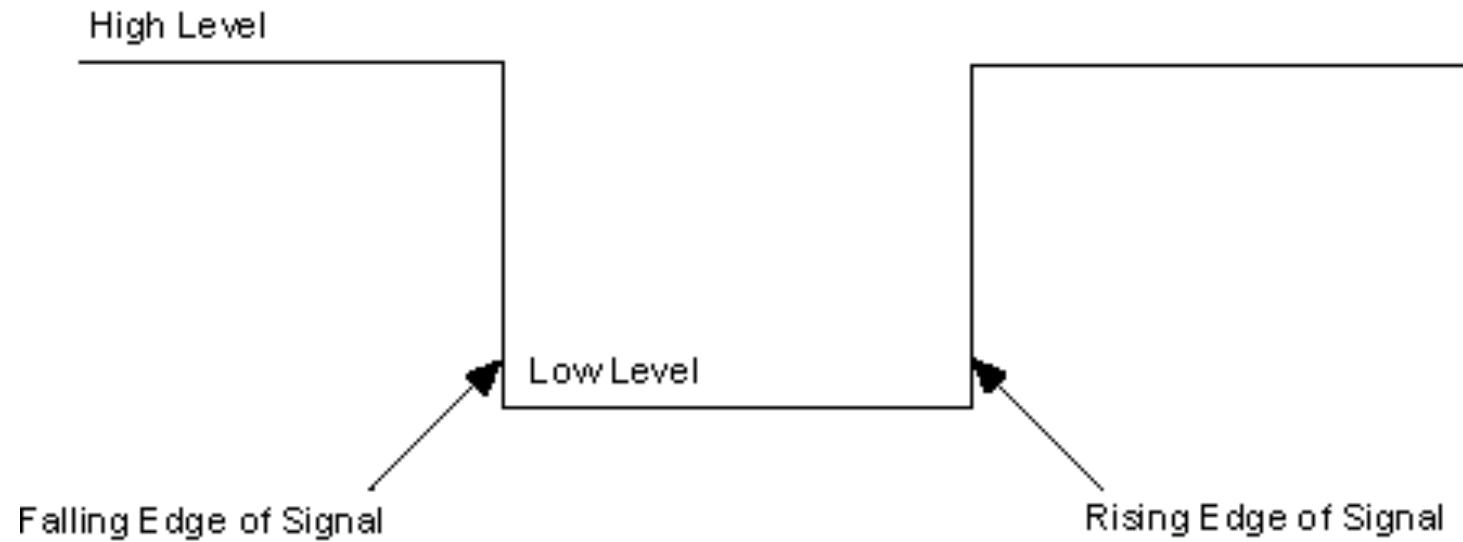
arduino មិនមែន 2 pin ទេ  
interrupt

**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, EXTI0_ISR, FALLING);
}

void loop()
{

}

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

# Exercise



- ให้นำโปรแกรมใน Assignment 4 มาเขียนโดยใช้ Interrupt
- ตอบคำถามว่า ในกรณีที่ให้เลือกใช้งาน นักศึกษาจะเลือกการเขียนโปรแกรมแบบใด เพื่ออะไร



*for your attention*