

# Arduino Workshop



# Outline

- Arduino Basics
- Project 1
  - Serial connection
  - Parallel connection



# Part I

Basics

# Micro Processor

- Single-Chip Microcomputer
- also called Microcontroller
- A small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals.





# http://www.arduino.cc/

www.arduino.cc

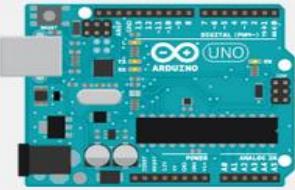
生活資訊 線上問卷 Temp Google 研究英文 8Cyber Google 日曆 Mecanim Dexpot miesocket TANET 修齊 LouisD

ARDUINO

Search the Arduino Website

Home Buy Download Products Learning Reference Support Blog LOG IN SIGN UP

## WHAT IS ARDUINO?



BUY AN ARDUINO



LEARN ARDUINO



ARDUINO ATHEART™

Designed for makers and companies wanting to make their products easily recognizable as based on the Arduino technology.



BLOG

ANNOUNCING A WEARABLE  
COLLABORATION WITH  
ADAFRUIT: ARDUINO

ARDUINO ESPLORA,  
MAKE YOUR PERSONAL  
VIDEOGAME!  
SHOP NOW



Maker Faire®  
THE EUROPEAN EDITION

MAKE IT  
HAPPEN  
DISCOVER.  
INVENT.  
MAKE.

MAKERFAIREROME.EU  
ROME ■ 2014 ■ OCTOBER 3-5



BLOG



3/51

# Arduino

- Arduino started in 2005 as a project for students at the Interaction Design Institute Ivrea in Ivrea, Italy. At that time program students used a "BASIC Stamp" at a cost of \$100, considered expensive for students. Massimo Banzi, one of the founders, taught at Ivrea.
- A hardware thesis was contributed for a wiring design by Colombian student Hernando Barragan. After the wiring platform was complete, researchers worked to make it lighter, less expensive, and available to the open source community. The school eventually closed down, so these researchers, one of them David Cuartielles, promoted the idea.
- The current prices run around \$30 and related "clones" as low as \$9.



# Arduino (cont.)

- Arduino is open source hardware: the Arduino hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino Web site. Layout and production files for some versions of the Arduino hardware are also available. The source code for the IDE is available and released under the GNU General Public License, version 2.
- Although the hardware and software designs are freely available under copyleft licenses, the developers have requested that the name "Arduino" be exclusive to the official product and not be used for derivative works without permission. The official policy document on the use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product.

# Arduino (cont.)

- Open-source electronics prototyping platform based on flexible, easy-to-use hardware and software.
- Cross-platform application written in Java, and is derived from the IDE for the Processing programming language and the Wiring projects.
- Intended for artists, designers, hobbyists and anyone interested in creating interactive objects or environments.
- Can communicate with software running on a computer (e.g. Flash, Processing, MaxMSP).

# 學習資源

- [Arduino – Home Page](#)
- [Arduino.tw](#)
- [Cooper Maa 部落格](#)
- Google it!



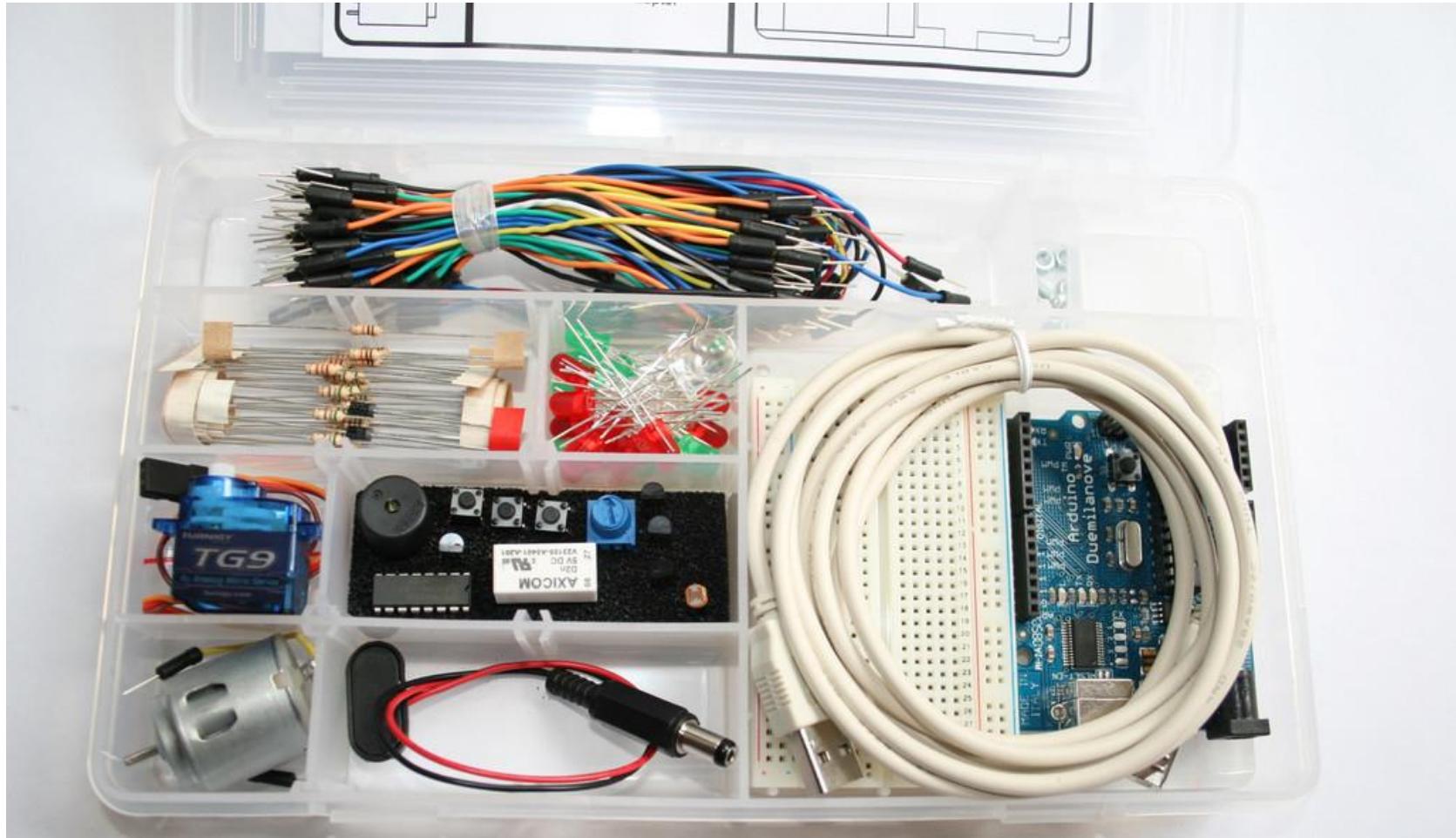
# Arduino Basics

Acknowledgements: Some slides in the following section is created based on the slides from “Harvard Technical Session” and “Arduino Projects Book” as well as various materials from the web.

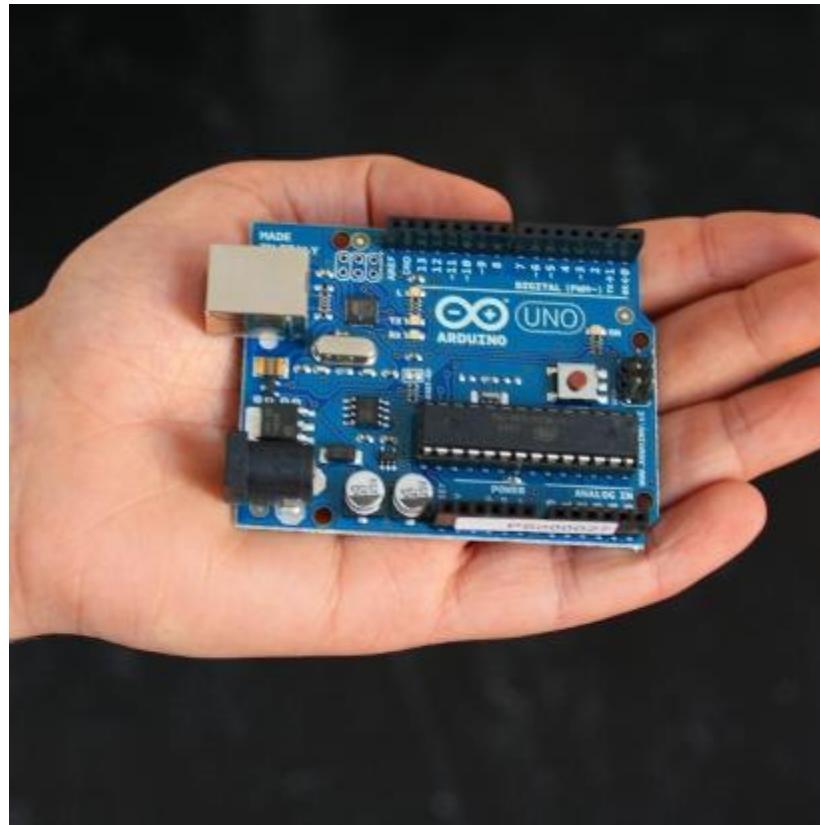
# Open the kit!!!



# Open the kit!!!

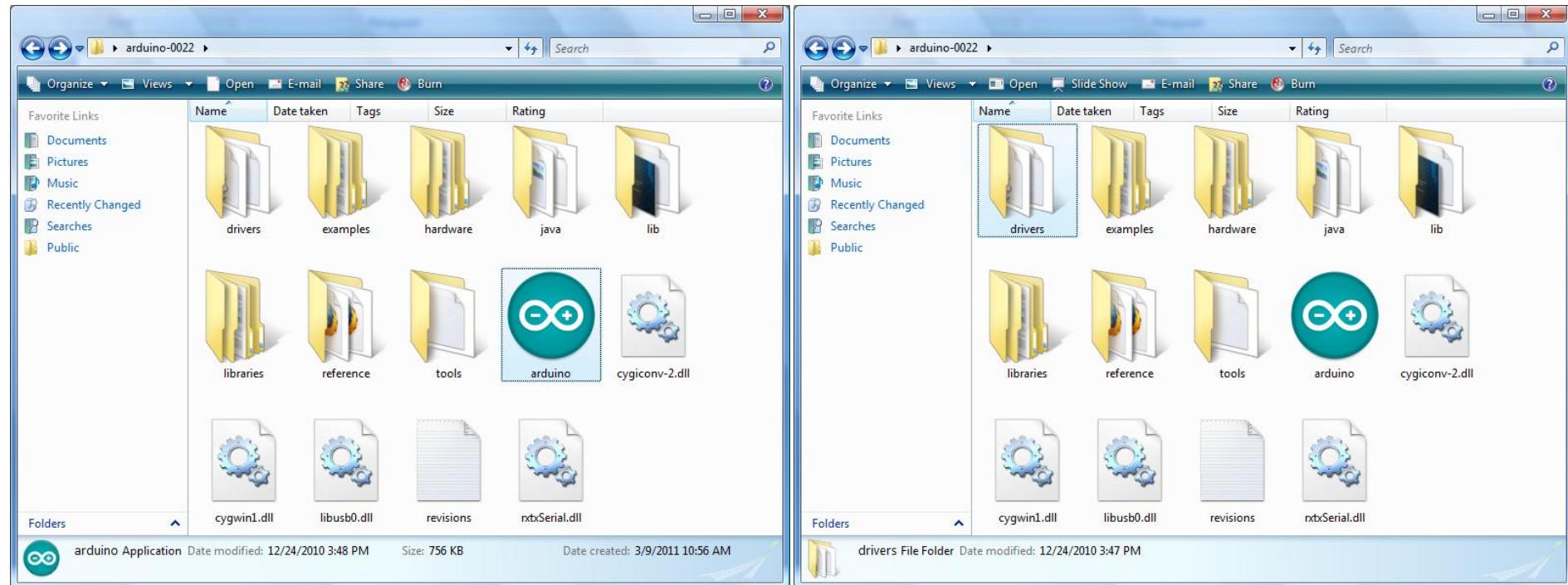


# Arduino Board: Arduino UNO



Arduino Programming Environment Download @  
<http://arduino.cc/en/Main/Software>

# IDE installation



\*Download Arduino Software from Arduino.cc and unzip the folder to your computer. A file within the folder called Arduino, allows you to launch the programming environment.

\*You need to install a driver that comes with Arduino to be able to communicate with the board

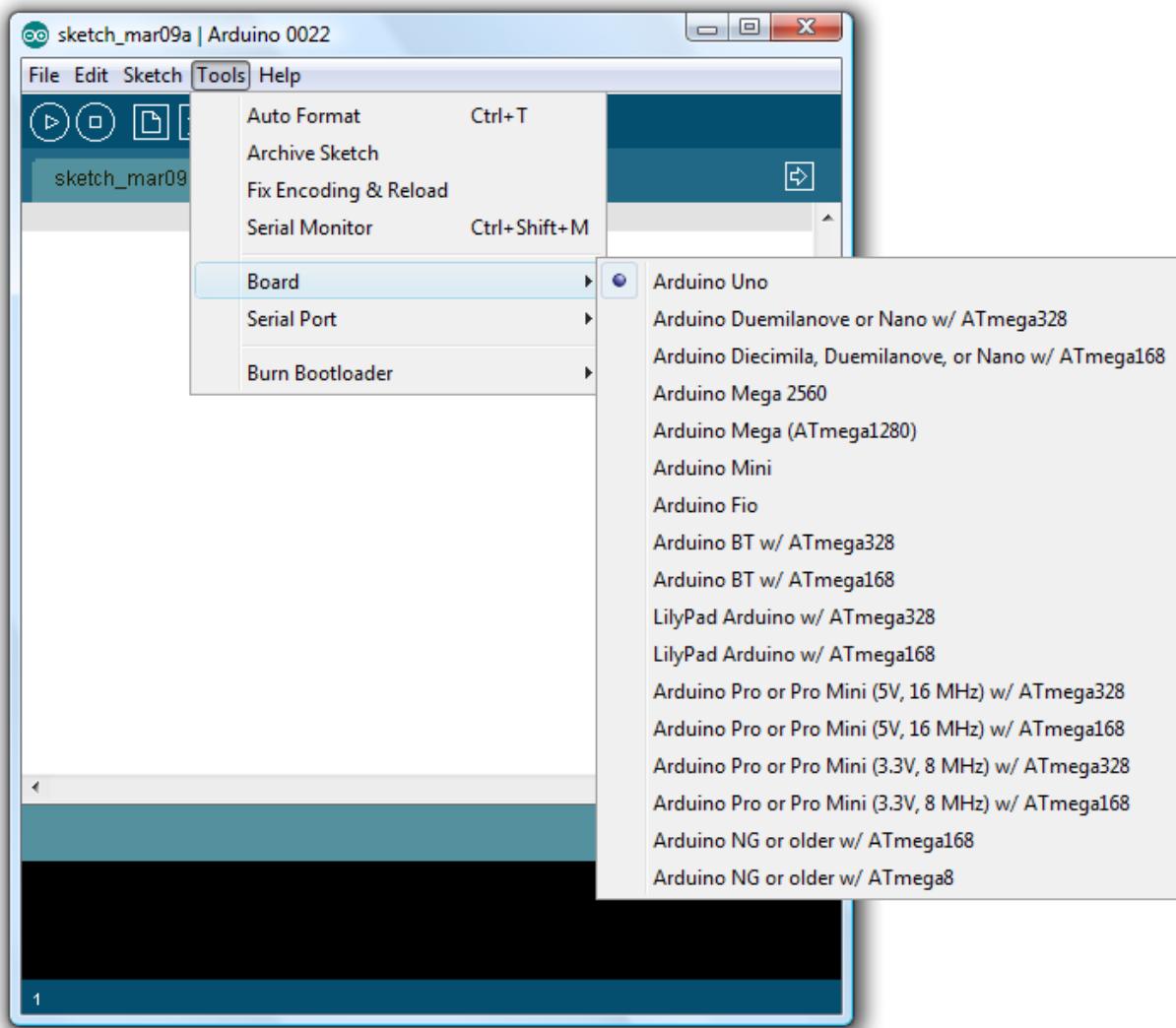


# Driver Installation

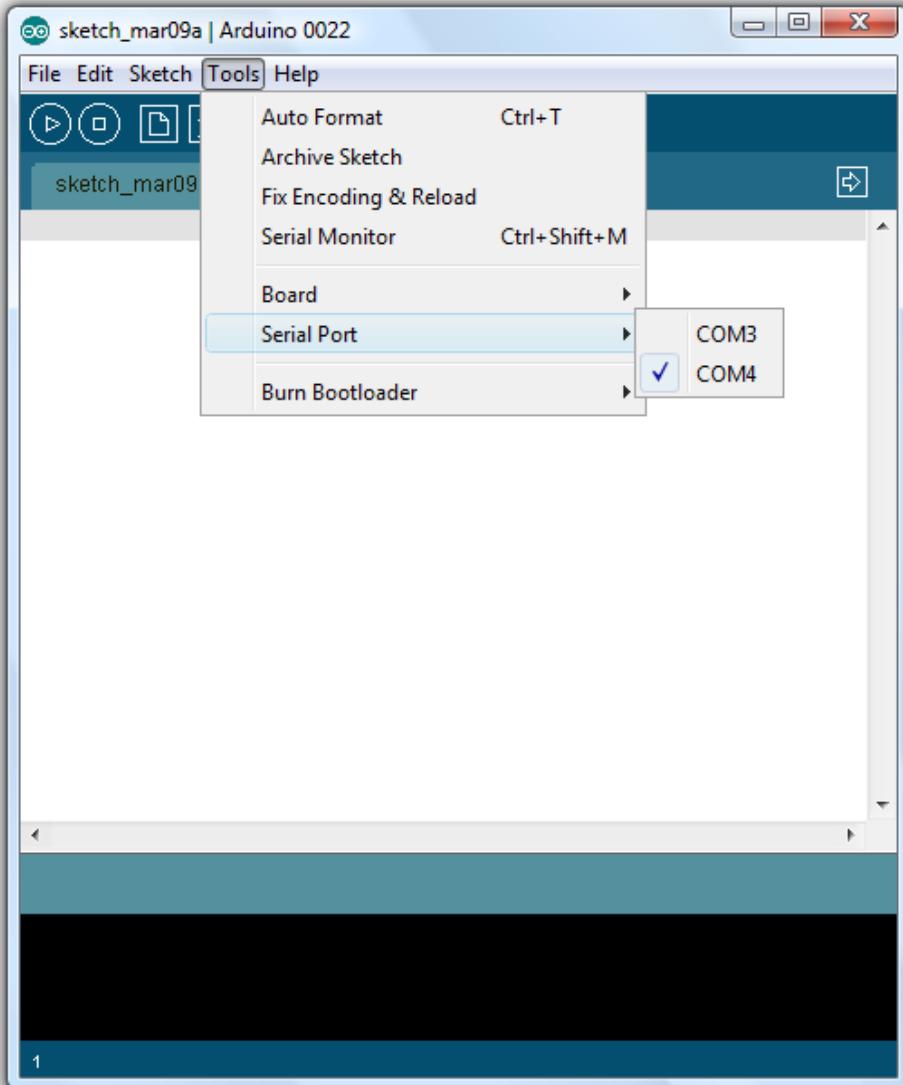
- You will be asked whether to install automatically or "with a path". Choose the second option, "with a path". On Windows Vista proceed directly to the next step.
- Select the "Browse my computer for Driver software" option.
- Navigate to the folder you unzipped in the earlier step. Locate and select the "**Drivers**" folder in the main Arduino folder (not the "FTDI USB Drivers" sub-directory). Press "OK" and "Next" to proceed. - If you are prompted with a warning dialog about not passing Windows Logo testing, click "Continue Anyway".
- Windows now will take over the driver installation. In the Device Manager, you should now see a port listing similar



# Arduino-Checking the Right Board



# Arduino-Checking the Right Port



# Arduino-Compiling and Uploading Code

```
void setup(){
//Code that runs once
}
void loop(){
//Code that runs repeatedly
}
```

Done Saving.  
Binary sketch size: 1104 bytes  
(of a 14336 byte maximum)

```
void setup(){
pinMode(13, OUTPUT);
}
void loop(){
digitalWrite(13, HIGH);
delay(1000);
digitalWrite(13, LOW);
delay(1000);
}
```

Done uploading.  
Binary sketch size: 1104 bytes  
(of a 14336 byte maximum)

```
void setup(){
pinMode(13, OUTPUT);
}
void loop(){
digitalWrite(13, HIGH);
delay(1000);
digitalWrite(13, LOW);
delay(1000);
}
```

Uploading to I/O Board...

```
void setup(){
pinMode(13, OUTPUT);
}
void loop(){
digitalWrite(13, HIGH);
delay(1000);
digitalWrite(13, LOW);
delay(1000);
}
```

Done uploading.  
Binary sketch size: 1104 bytes  
(of a 14336 byte maximum)

1. Write the code
2. Compile the code
3. Check Arduino Port Connection
4. Upload the Code
5. The Arduino and Connected Circuits start to show behavior based on the uploaded code



# Layout

Digital Input/Output Pins

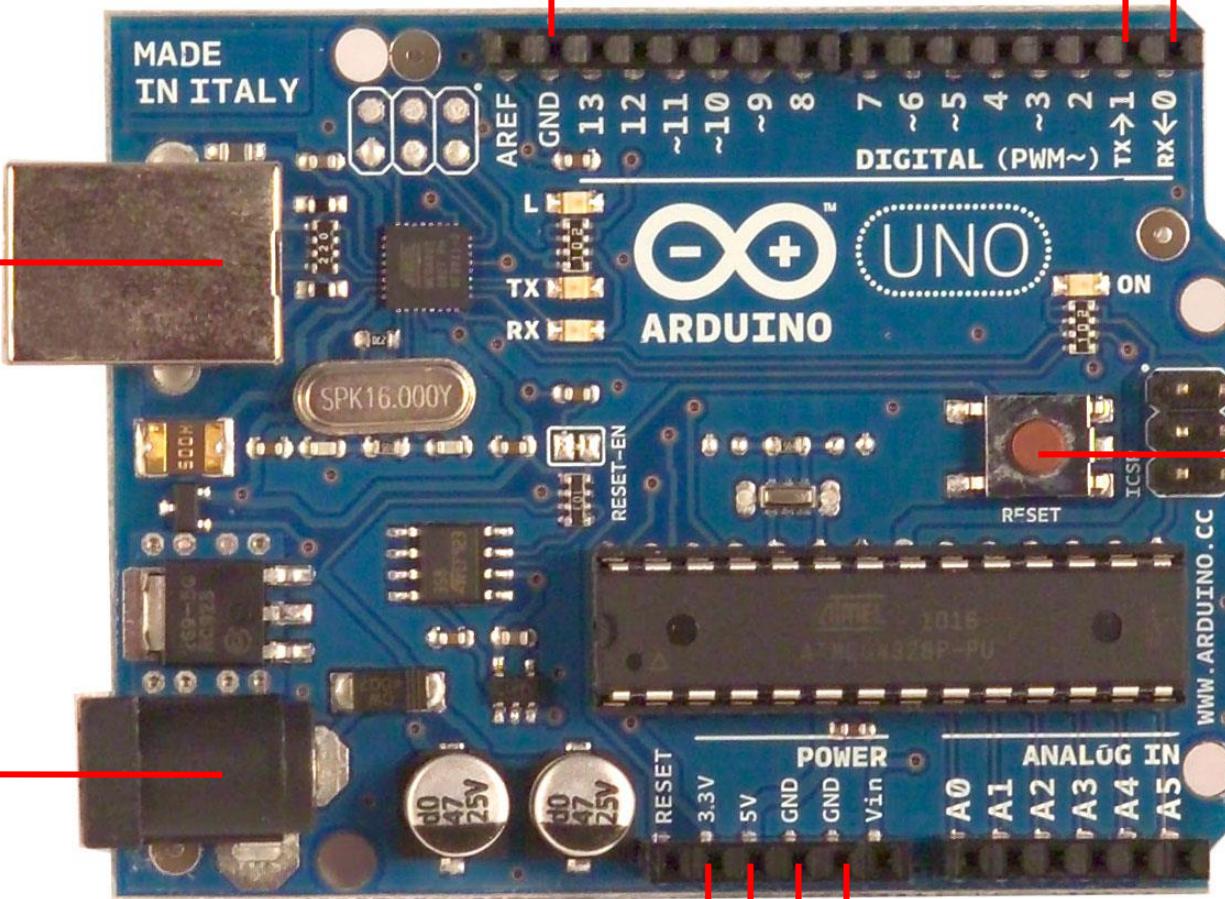
Pins with ~ are PWM

[Analog Output]

GND

Transmitter/Receiver  
Serial Connection

USB



- Microcontroller ATmega328
- Operating Voltage 5V
- Input Voltage (recommended) 7-12V
- Input Voltage (limits) 6-20V
- Digital I/O Pins 14
- (of which 6 provide PWM output)
- Analog Input Pins 6
- DC Current per I/O Pin 40 mA
- DC Current for 3.3V Pin 50 mA

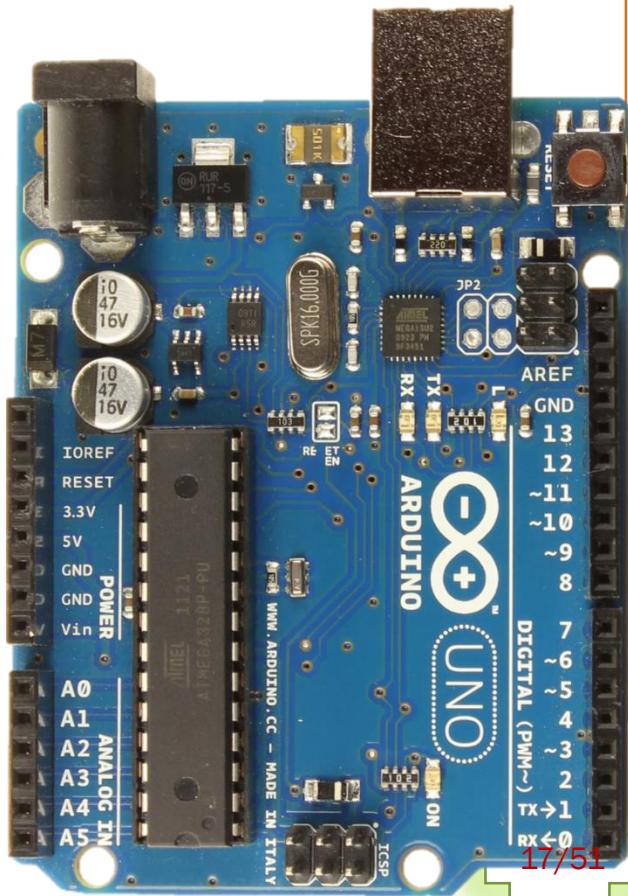
Reset

GND

Analog Input Pins

# Arduino Hardware

- Analog Read : A0 ~ A5
- Digital Write/Read : 0 ~ 13
- Analog Write (PWM) : ‘~X’
- Serial : 0 (RX) and 1 (TX)
- External Interrupts : 2 and 3
- SPI :
  - 10(SS), 11(MOSI), 12(MISO), 13(SCK)
- TWI (I<sup>2</sup>C) : A4(SDA) and A5(SCL)
- IOREF
- RESET
- AREF
- ICSP



# Signal Types

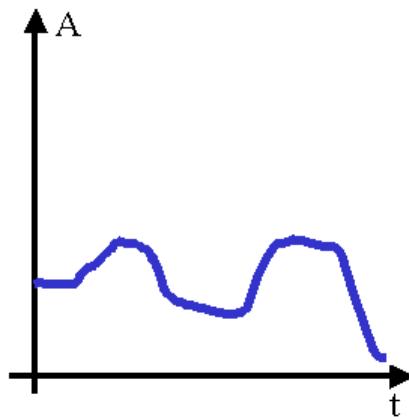
- Digital signal
  - Discrete time signals generated by digital modulation.
  - Less affected since noise response are analog in nature.



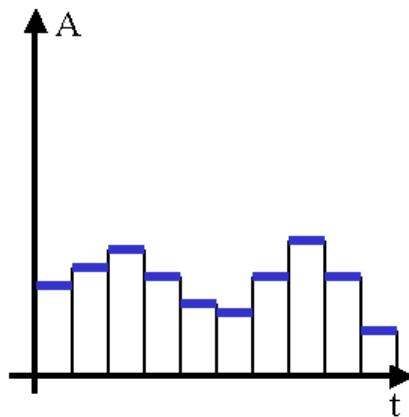
- Analog signal
  - Continuous signal which represents physical measurements.
  - More likely to get affected reducing accuracy.



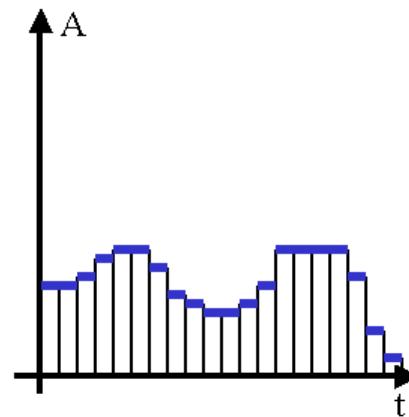
# Signal Types (cont.)



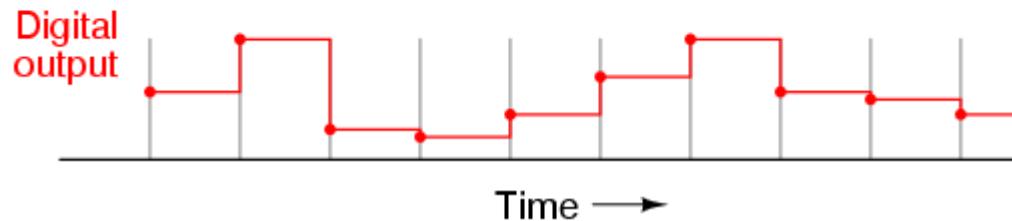
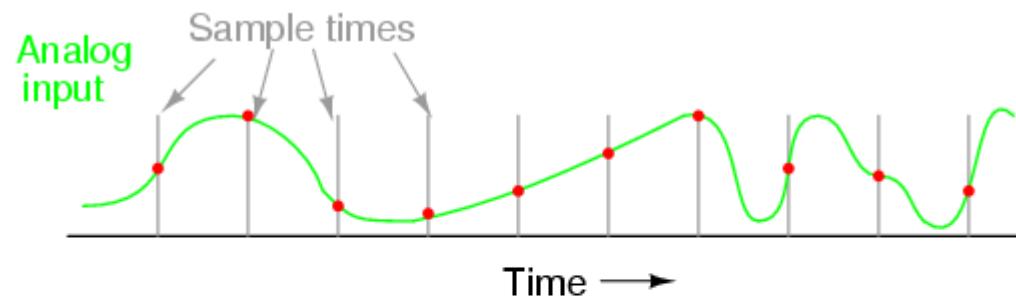
Analog signal – continuously varying



Digital signal – large time divisions



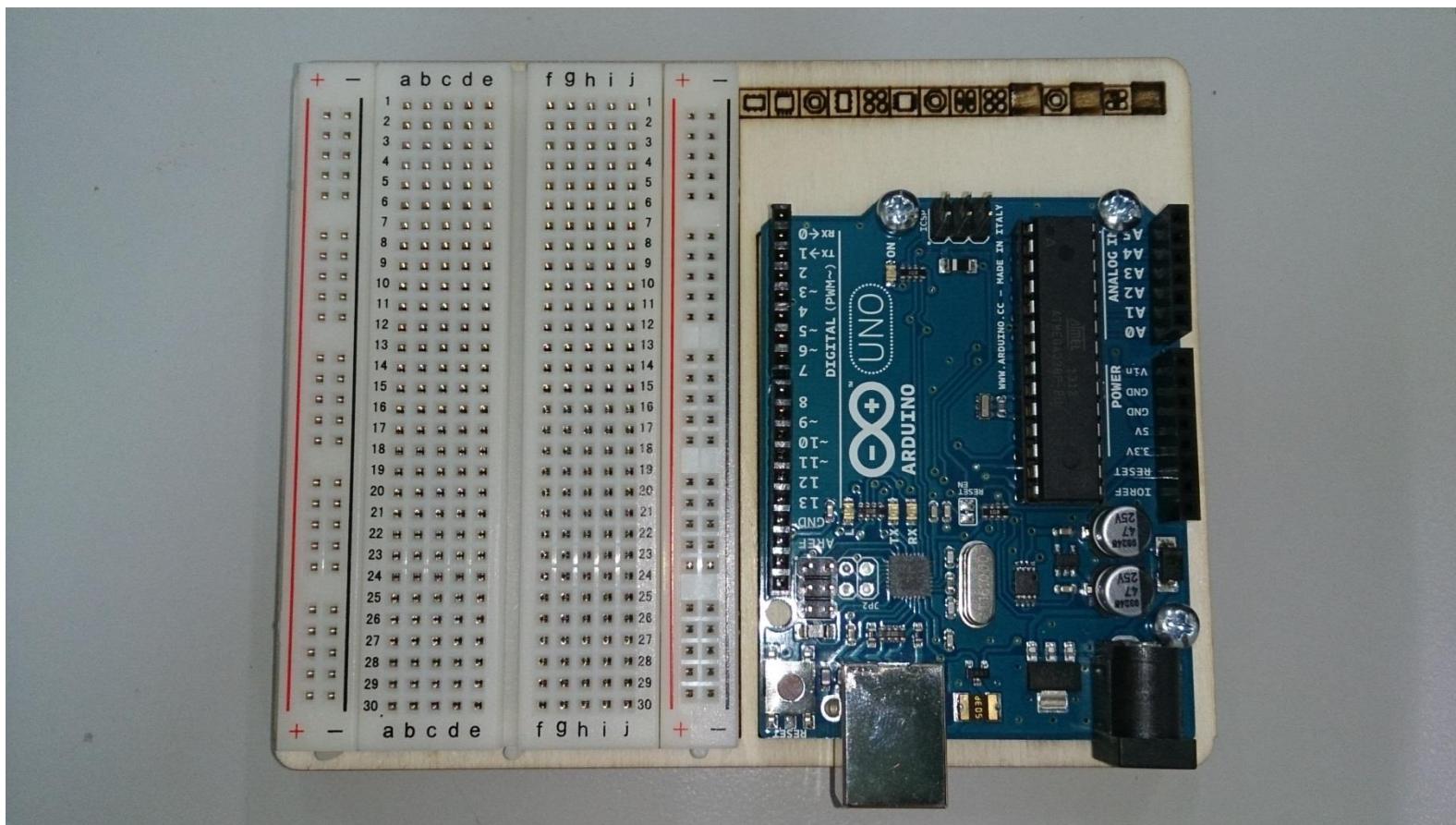
Digital signal – small time divisions



# Project #1

basic electrical theory, how a breadboard works,  
components in series and parallel

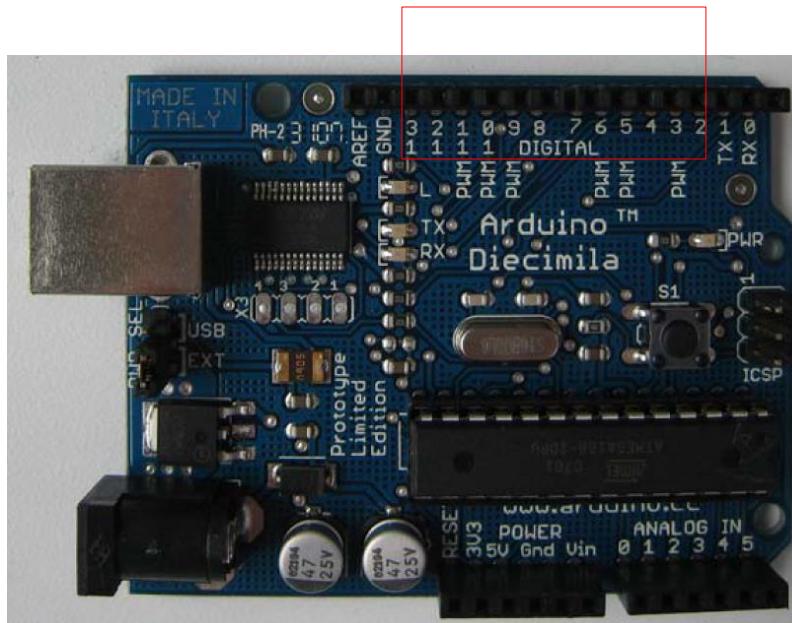
# Arduino套件板子裝法



# Arduino-Digital Output

Digital Output is defined as sending **on/off** or **0/1** signals from one of the digital pins on the Arduino board (**pin 2-13**) to the electronic actuator that recognize on/off or 0/1 signal.

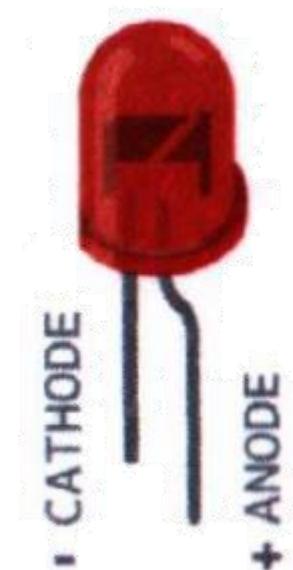
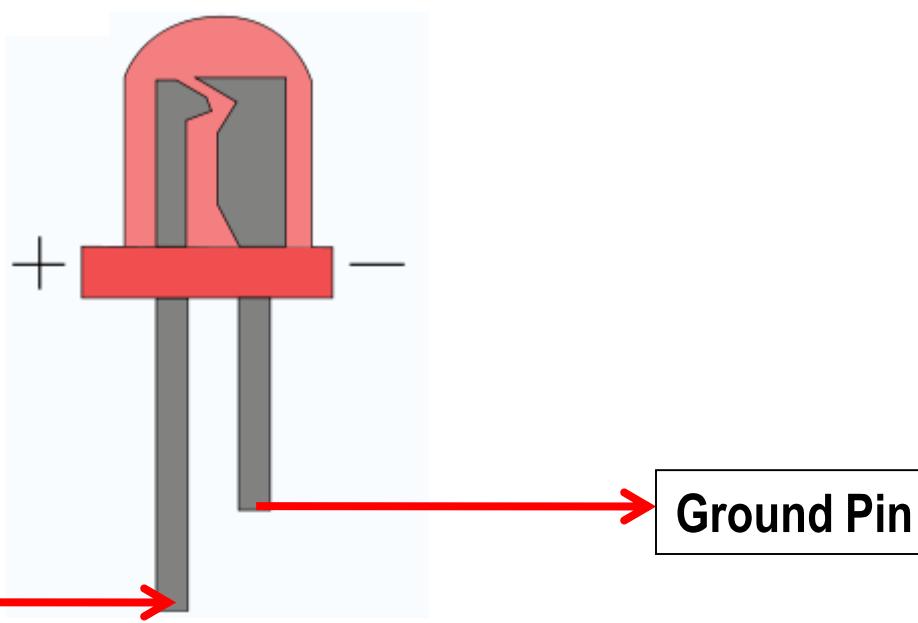
The so-called digital pins are highlighted here.



# Arduino-Digital Output-LED

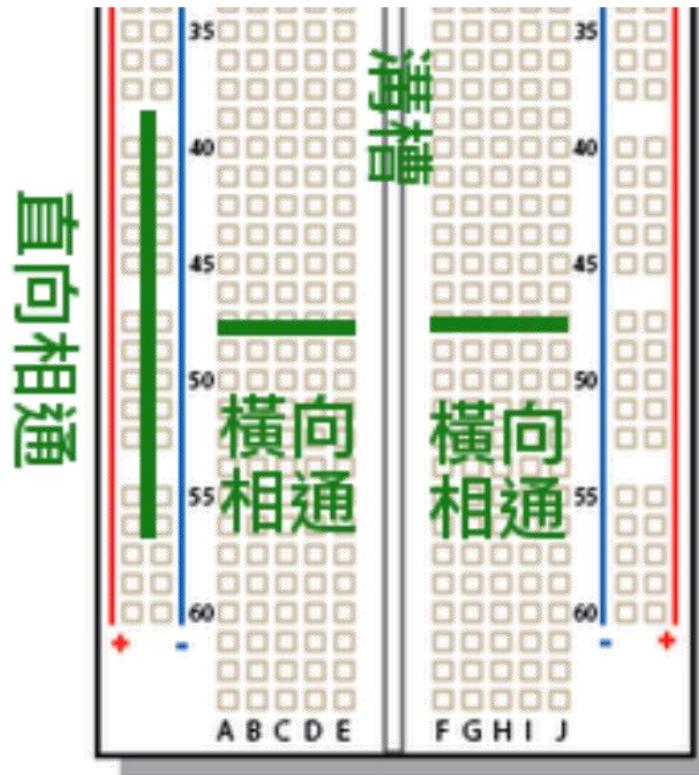
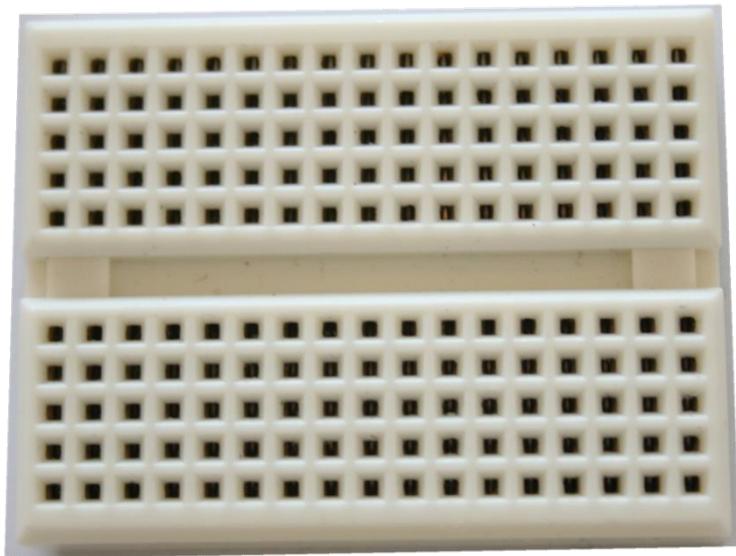
LED (Light Emitting Diode) is a light feature that can be used as an actuator of the space.

Being a Diode, an LED is a directional piece meaning that it is activated only if it is placed in the circuit in the right direction



# Arduino-Using Solderless Breadboard

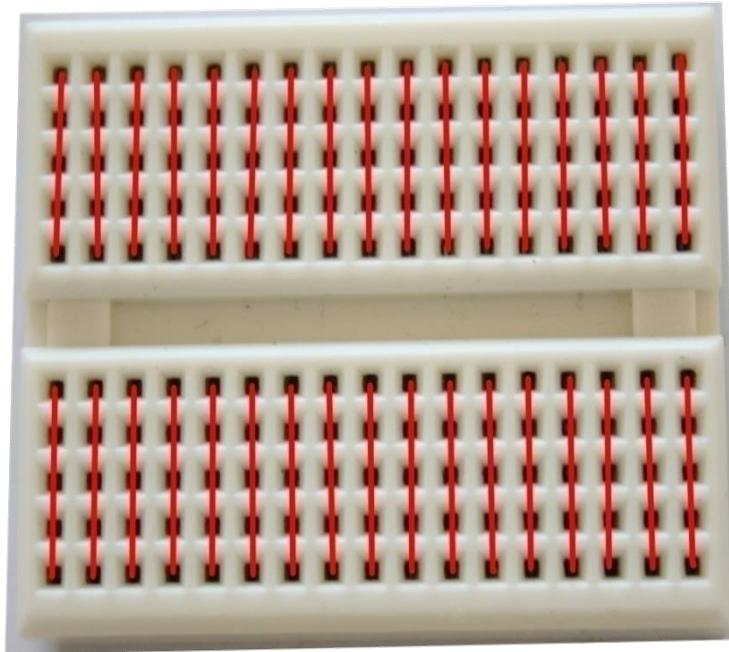
Solderless Board is useful to build prototypes, as fast as possible without going through tiresome and time consuming process of soldering parts together to make connections



# Arduino-Using Solderless Breadboard

Solderless Board is useful to build prototypes, as fast as possible without going through tiresome and time consuming process of soldering parts together to make connections

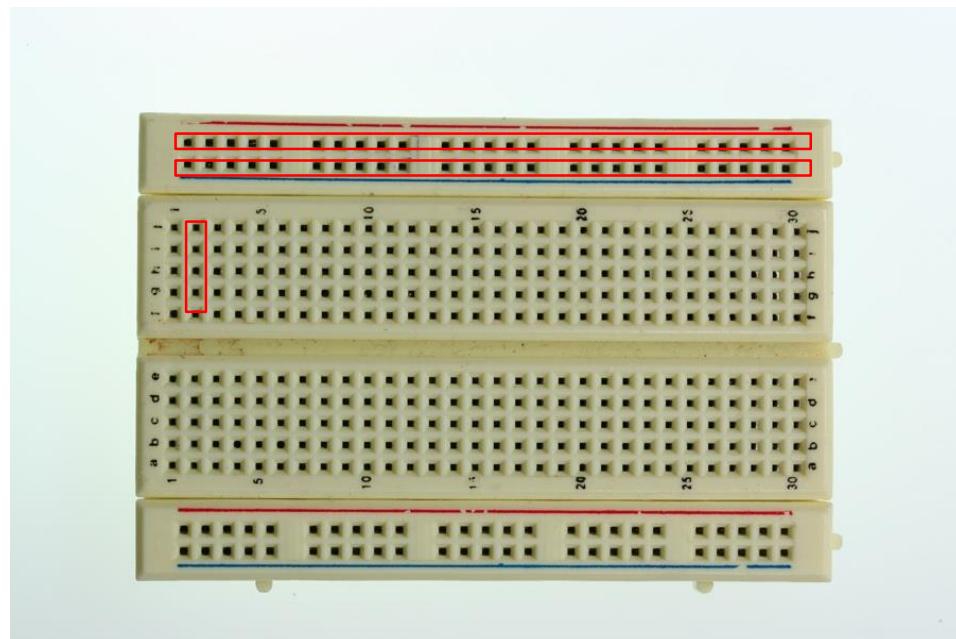
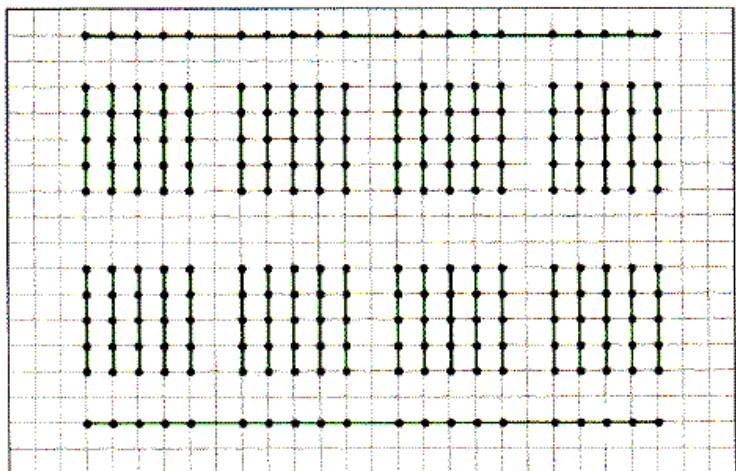
Most important thing in using a solderless breadboard in understanding its connections and wiring underneath the white cover to be able to connect parts in a way that complete and flawless lines are provided for electricity flow



# Arduino-Using Solderless Breadboard

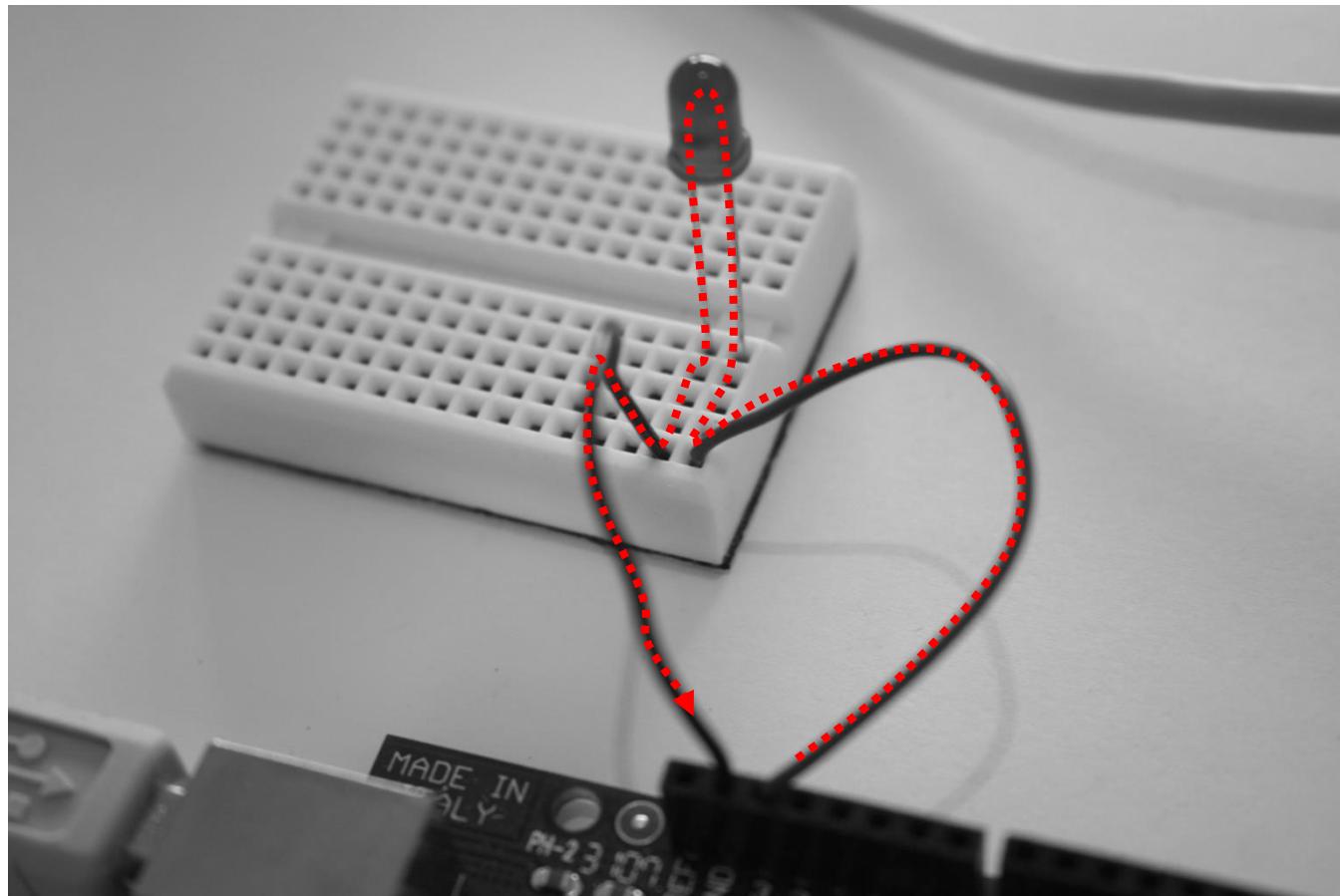
Solderless Board is useful to build prototypes, as fast as possible without going through tiresome and time consuming process of soldering parts together to make connections

Most important thing in using a solderless breadboard in understanding its connections and wiring underneath the white cover to be able to connect parts in a way that complete and flawless lines are provided for electricity flow



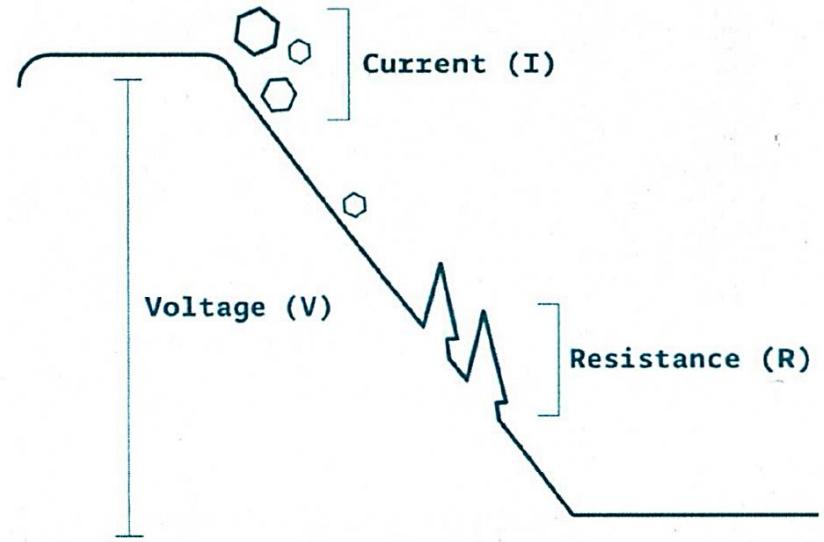
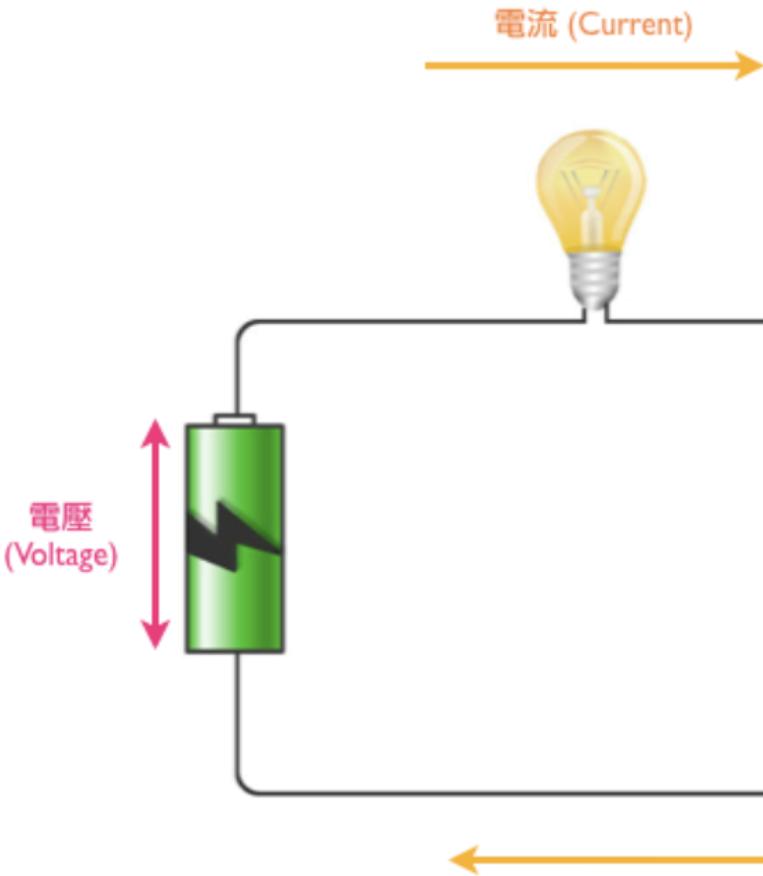
# Arduino-Using SolderlessBreadboard

For example this is how an LED can be connected to an Arduino board using a solderless breadboard. The red dotted line shows the flow of electricity from the digital output pin to LED and then ground pin.



# Circuitry Introduction

# 簡單的電路

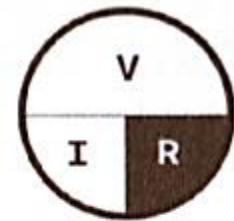
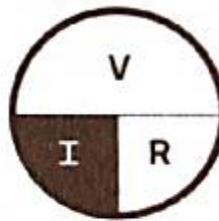
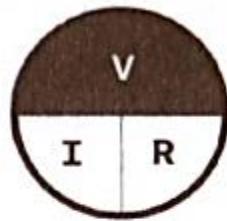
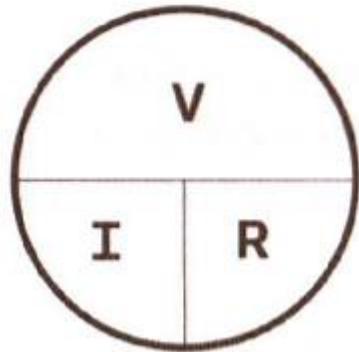


# OHM's Law

$$V = I * R$$

$$I = V / R$$

$$R = V / I$$



**Current, voltage, and resistance are all related.** When you change one of these in a circuit, it affects the others. The relationship between them is known as Ohm's Law, named for Georg Simon Ohm, who discovered it.

**VOLTAGE (V) = CURRENT (I) \* RESISTANCE (R)**

When measuring amperage in the circuits you'll be building, values will be in the millamp range. That's thousandths of one amp.



# Practice Time

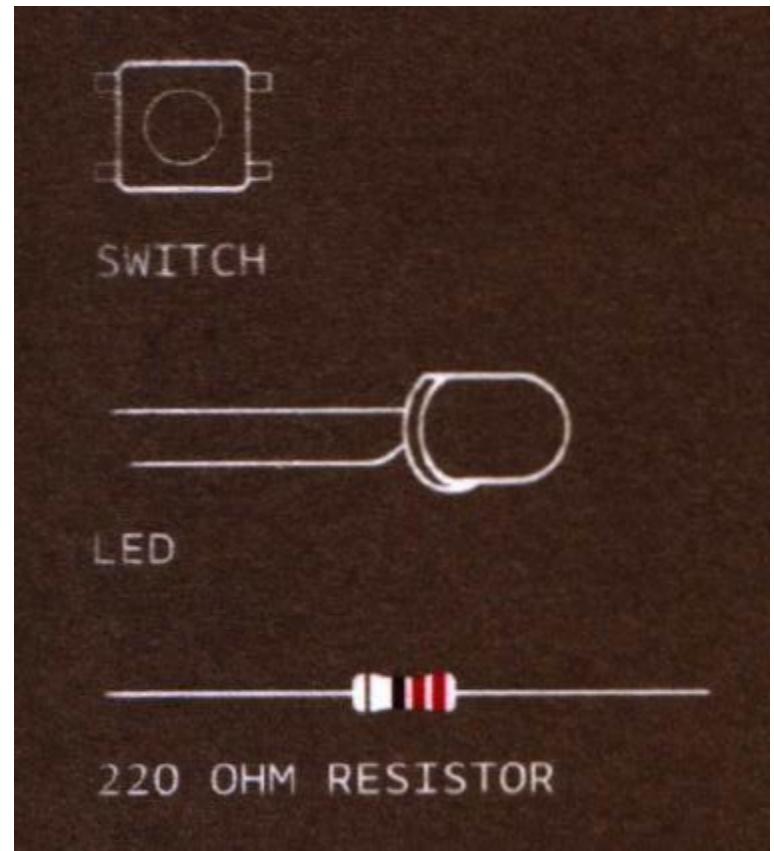
# Project #1

Get to know your tools

basic electrical theory, how a breadboard works,  
components in series and parallel

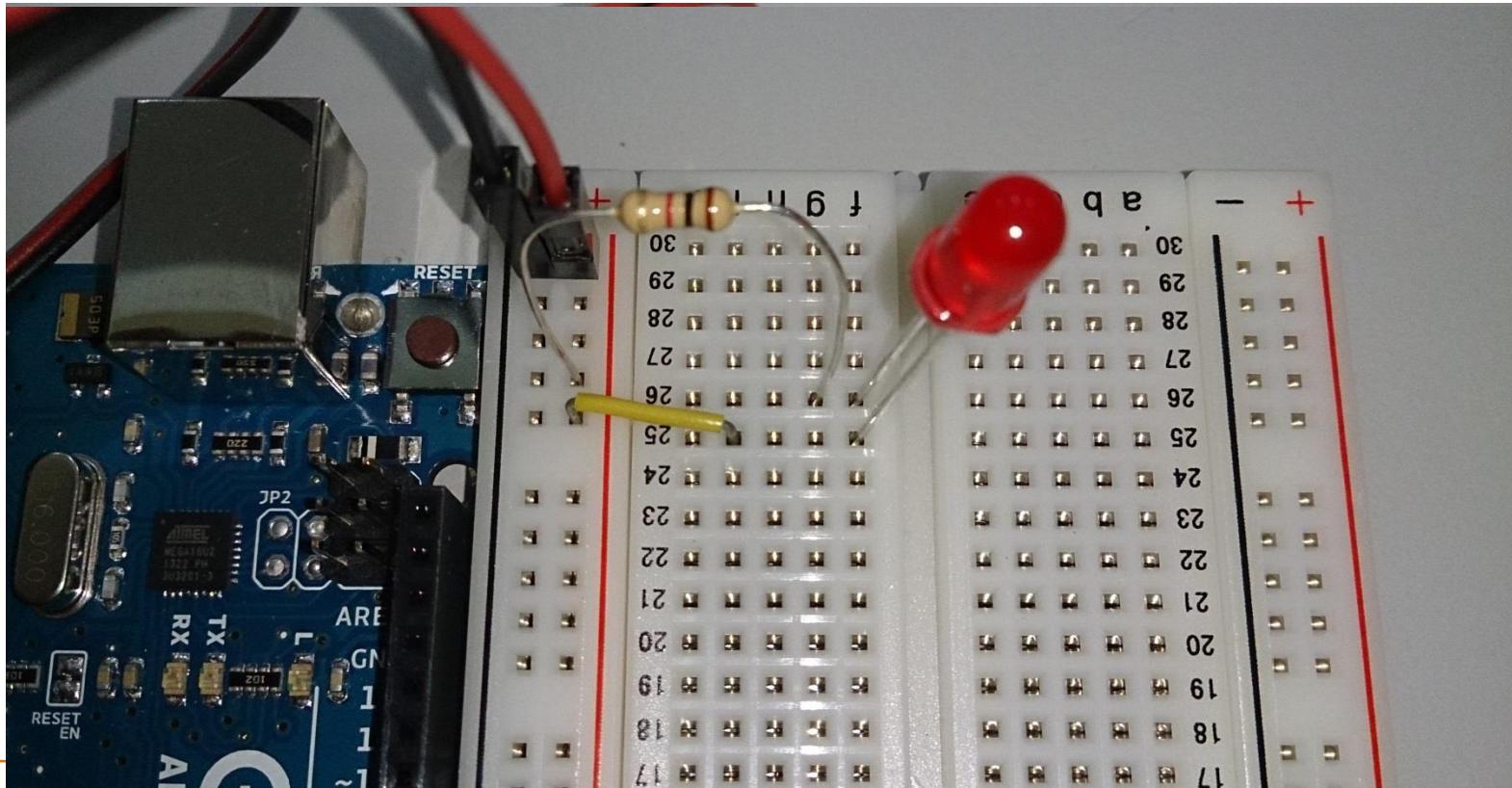
# Ingredients

- 3個LED燈
- 3個220Ω電阻
- 3個開關



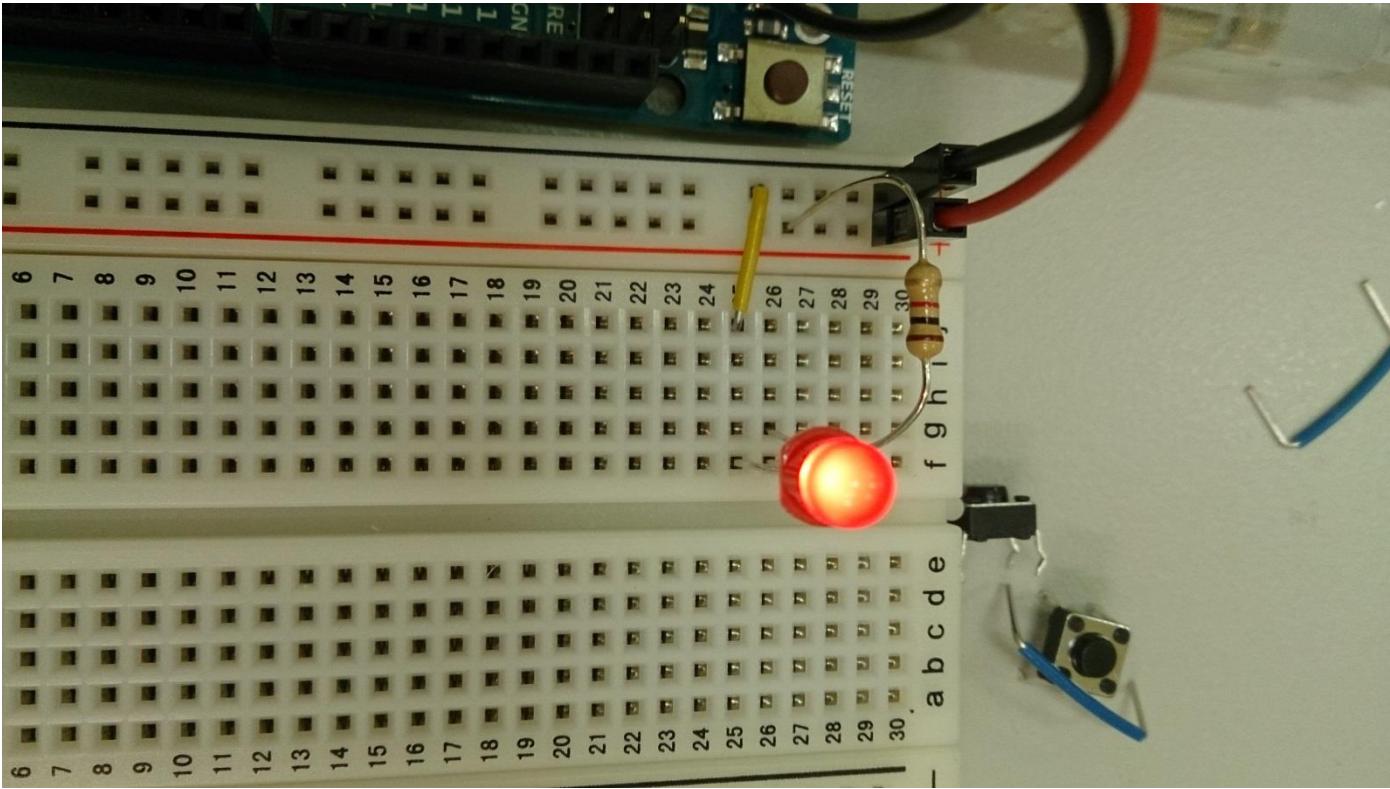
# 電路接法

- 紅線從5V接到麵包版上+ 黑線從GND接到麵包版上-
- LED接腳長的接25短的接26
- 電阻從+接到LED接長腳長 黃線從-接到LED短接腳

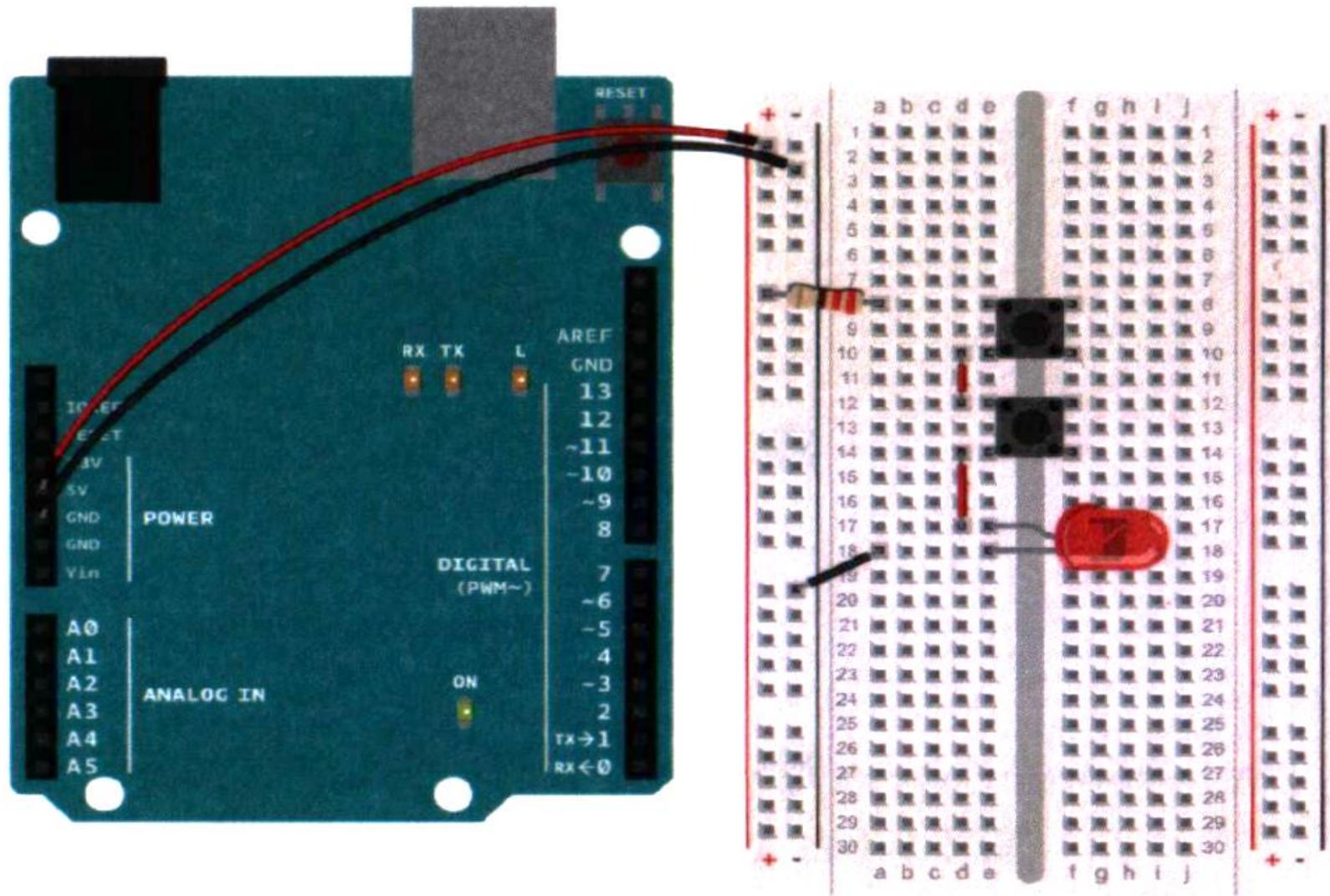


# 電路接法

- Arduino接電腦，LED會發亮
- 電從紅線出來通過電阻在到LED回到黑線GND



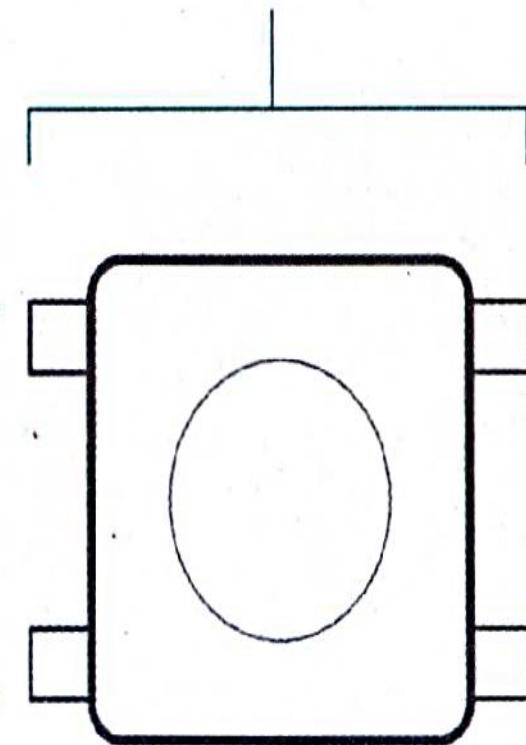
# The Top View of the Circuit



# The switch

## SWITCH CONNECTIONS

These two pins of a switch are connected to each other



These two are not.  
They form the switch

## SWITCH SCHEMATIC VIEW



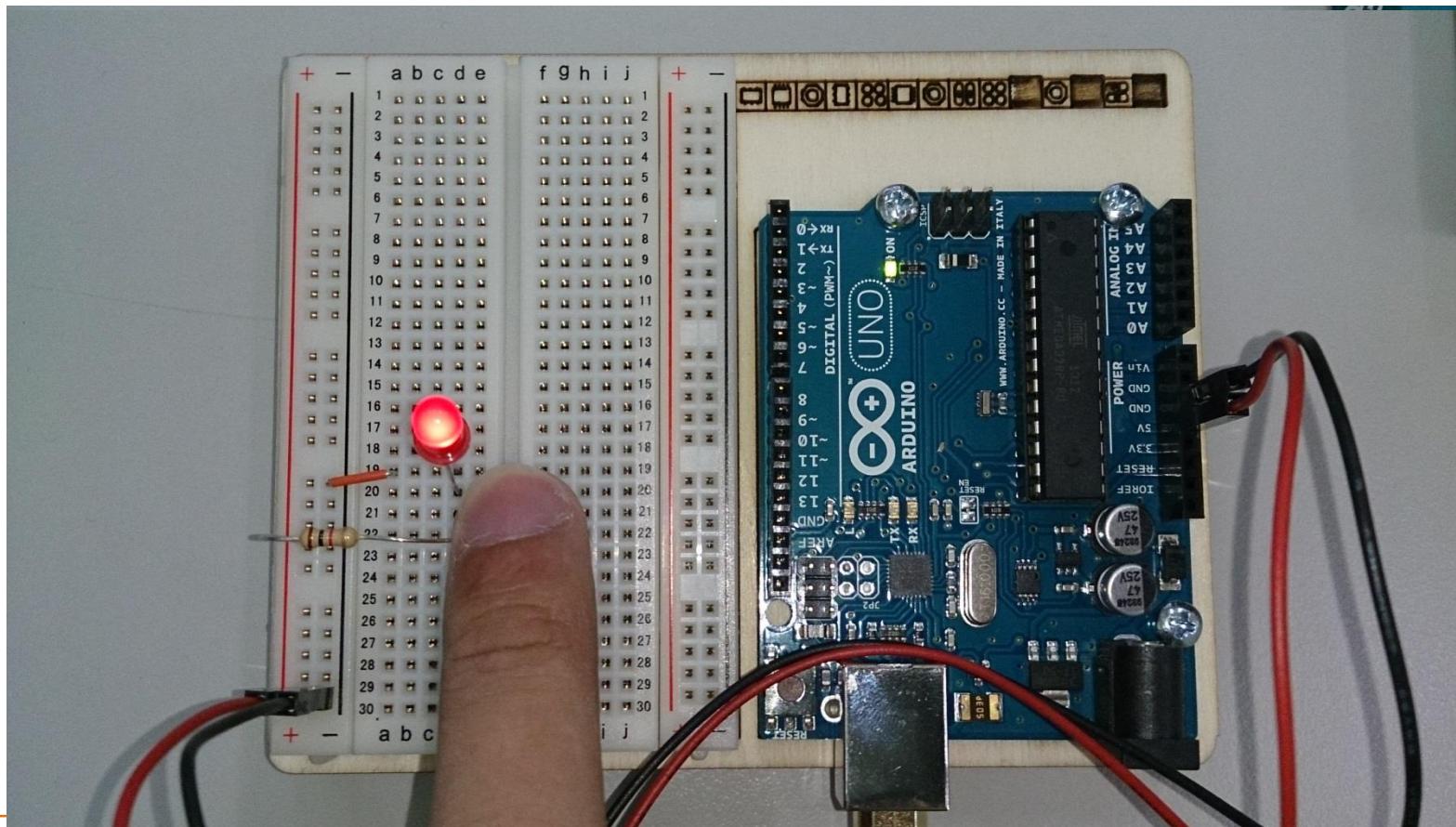
A - Toggle switch symbol



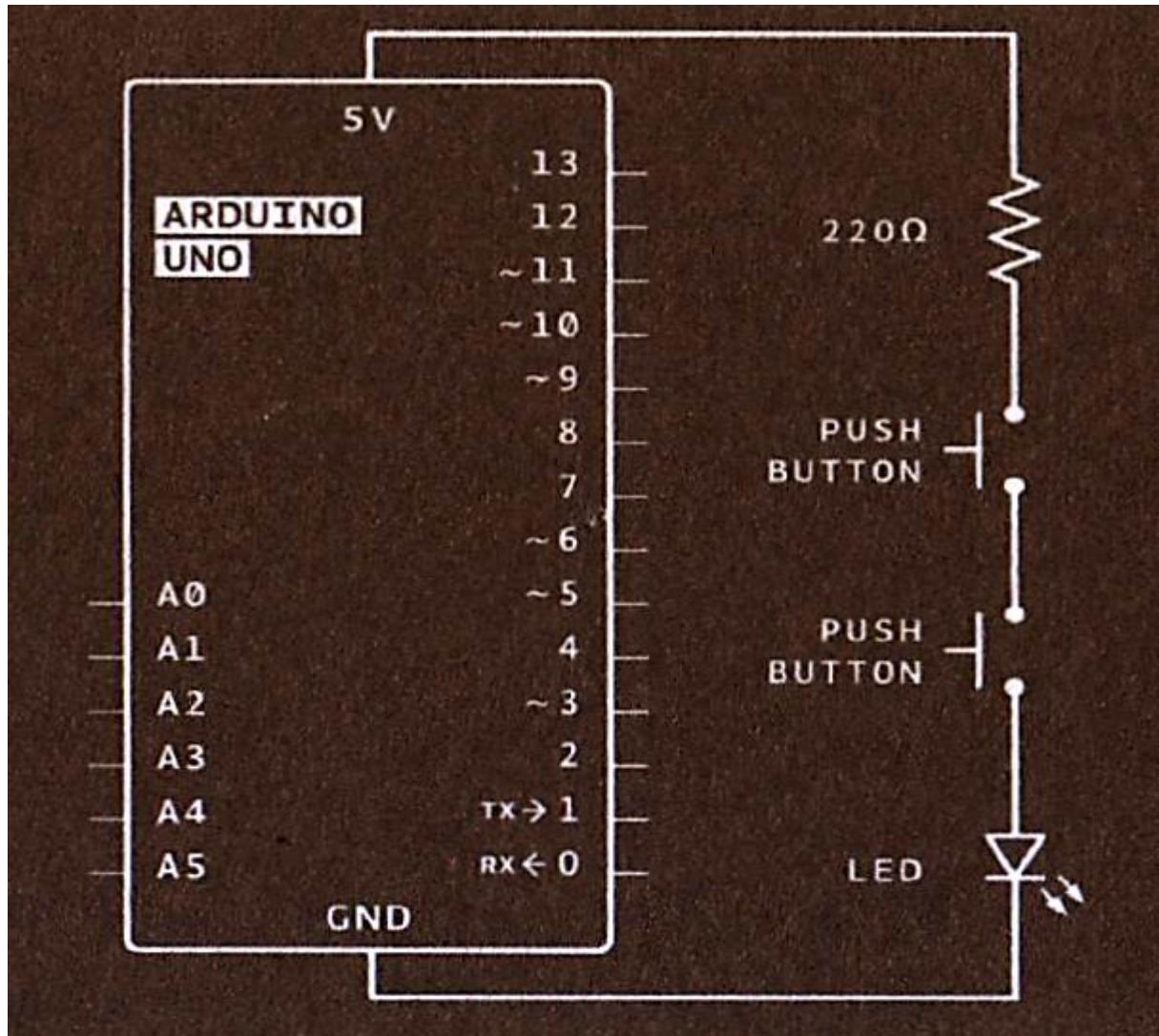
B - Pushbutton symbol

# 按鈕接法

- 橘線接到負在接到LED短的街腳
- 電阻接到正在接到LED長的街腳

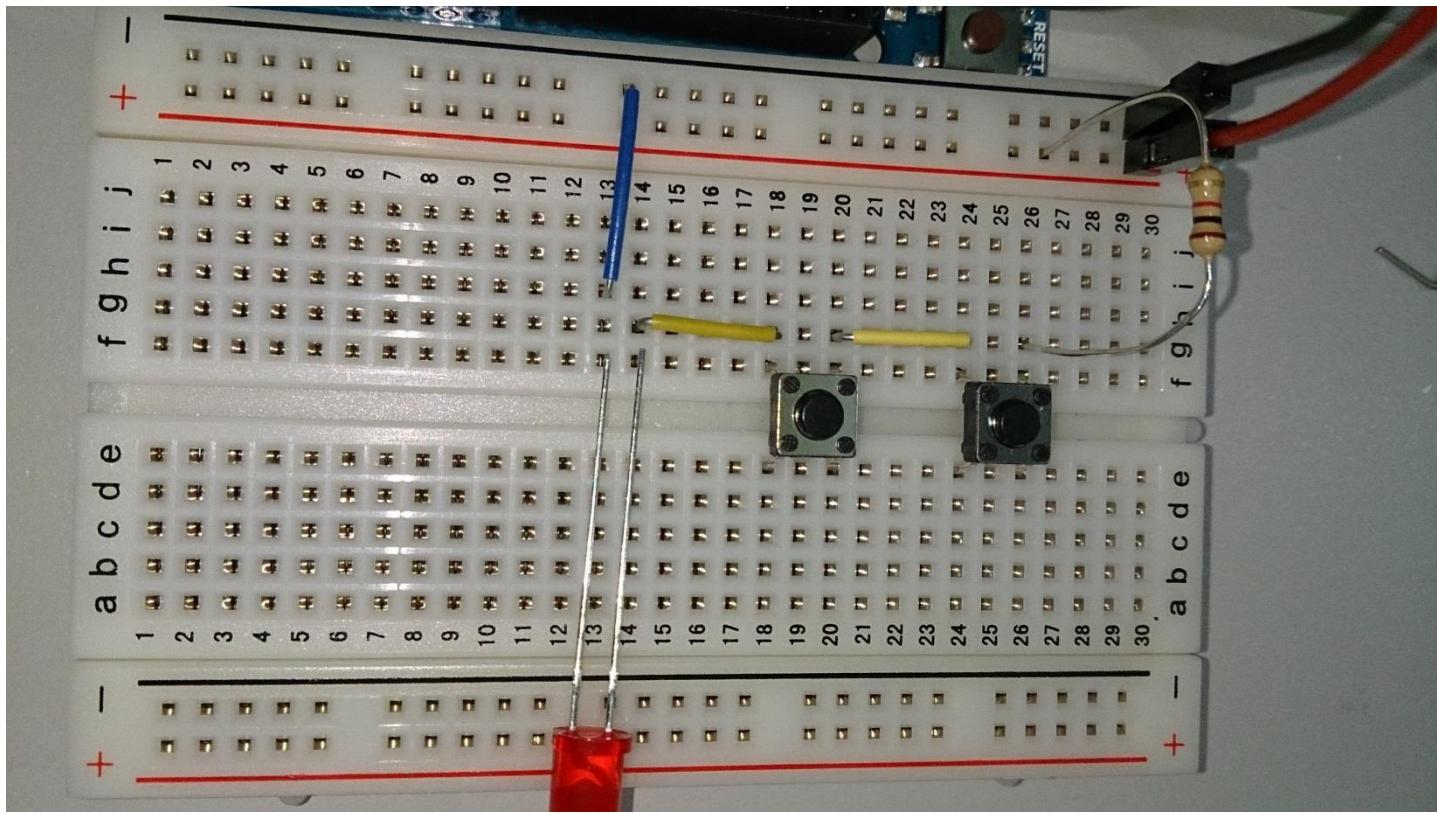


# Practice 2



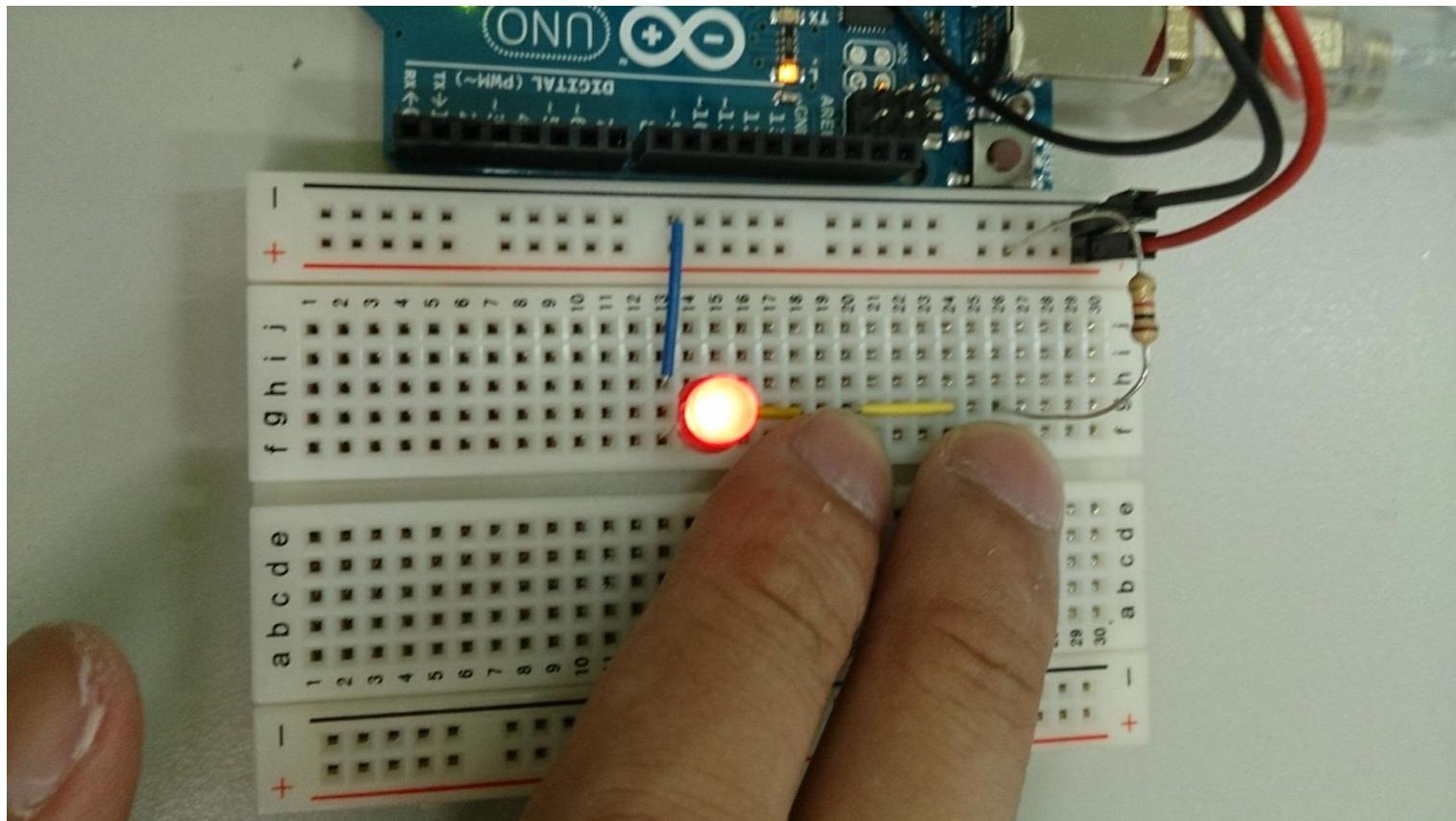
# 串聯電路

- 圖中有2個按鈕開關，當按鈕按下時才會通電
- 任意按下1鈕燈是不會亮，必須同時按下才會亮



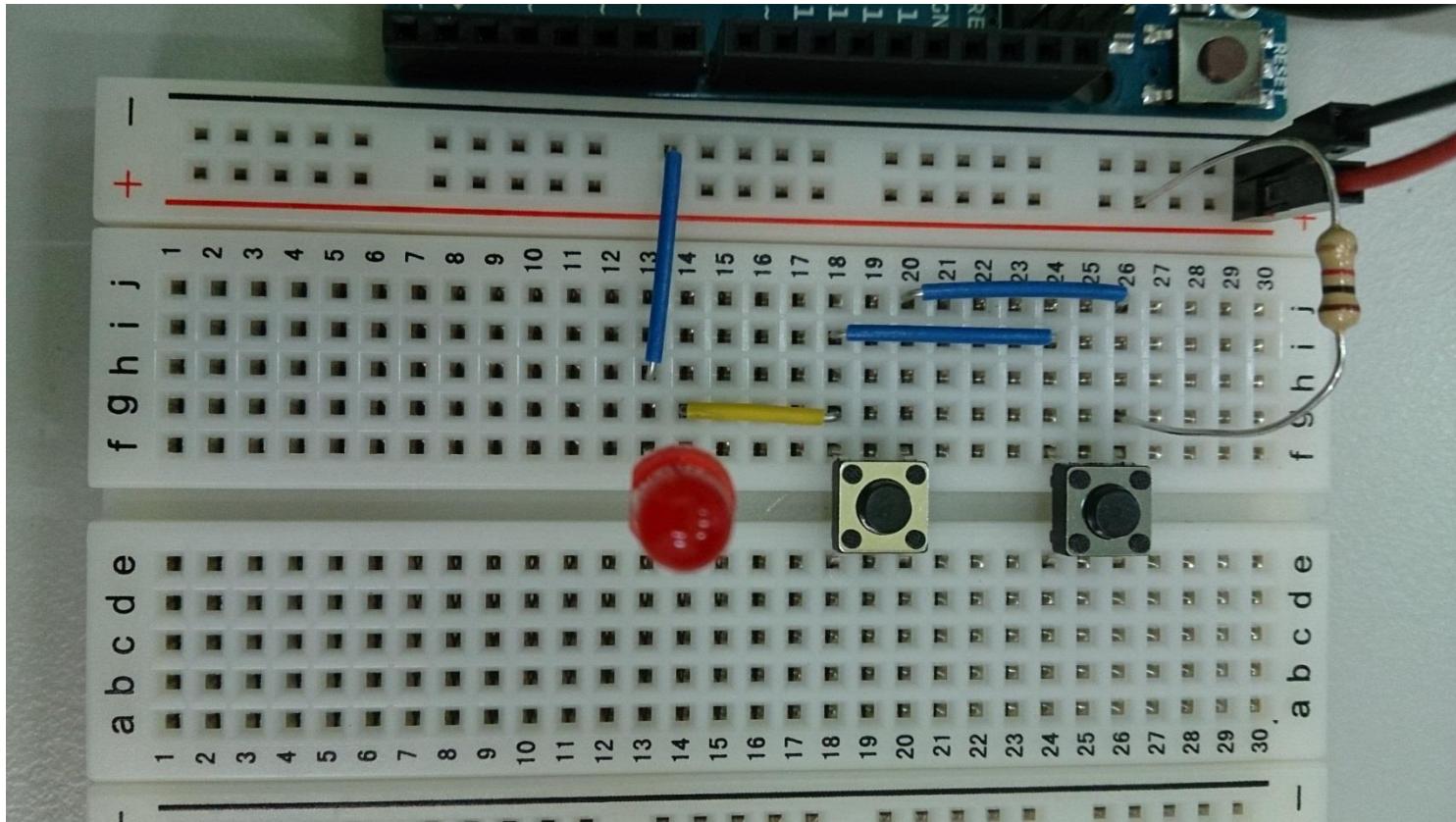
# 串聯電路

- 串聯電路是指電流會通過每一個元件，當按鈕沒按時電就流不過去，LED自然就不會亮



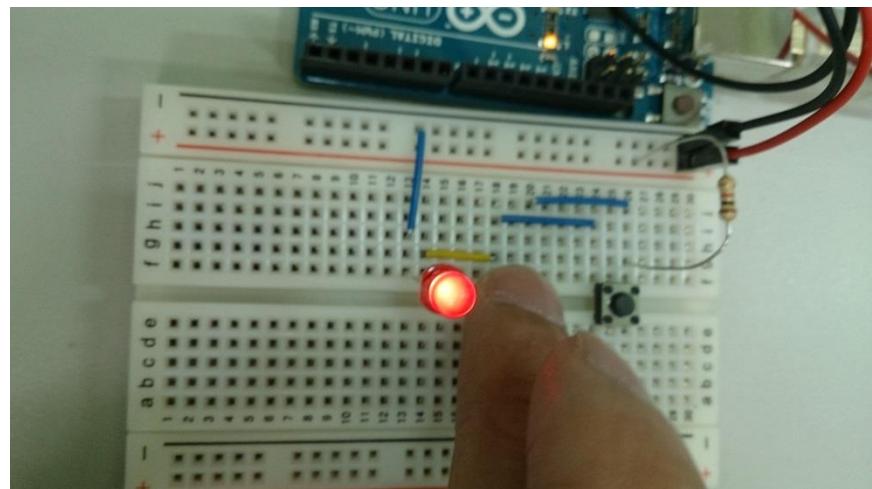
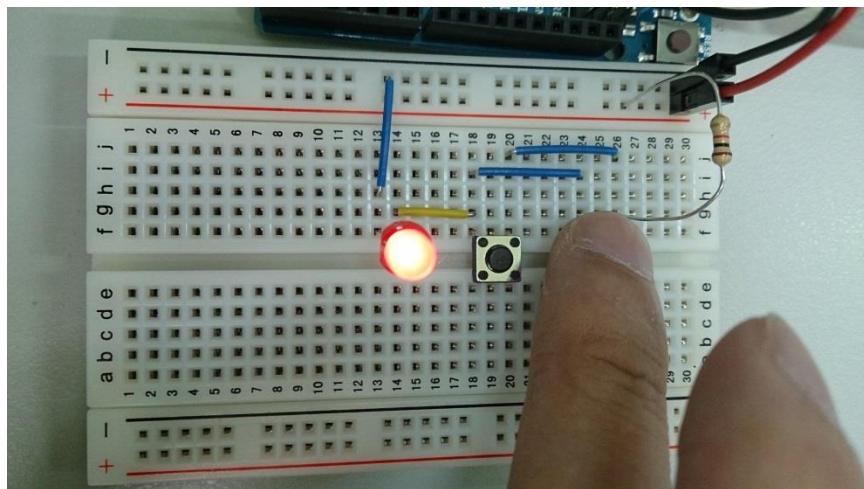
# 並聯電路

- 圖中有2個按鈕開關，任意按下1鈕LED燈就會亮



# 並聯電路

- 並聯電路中通往LED的路不只有一條，所以只要有一按鈕是按著就會亮



# Project #2

Spaceship Interface

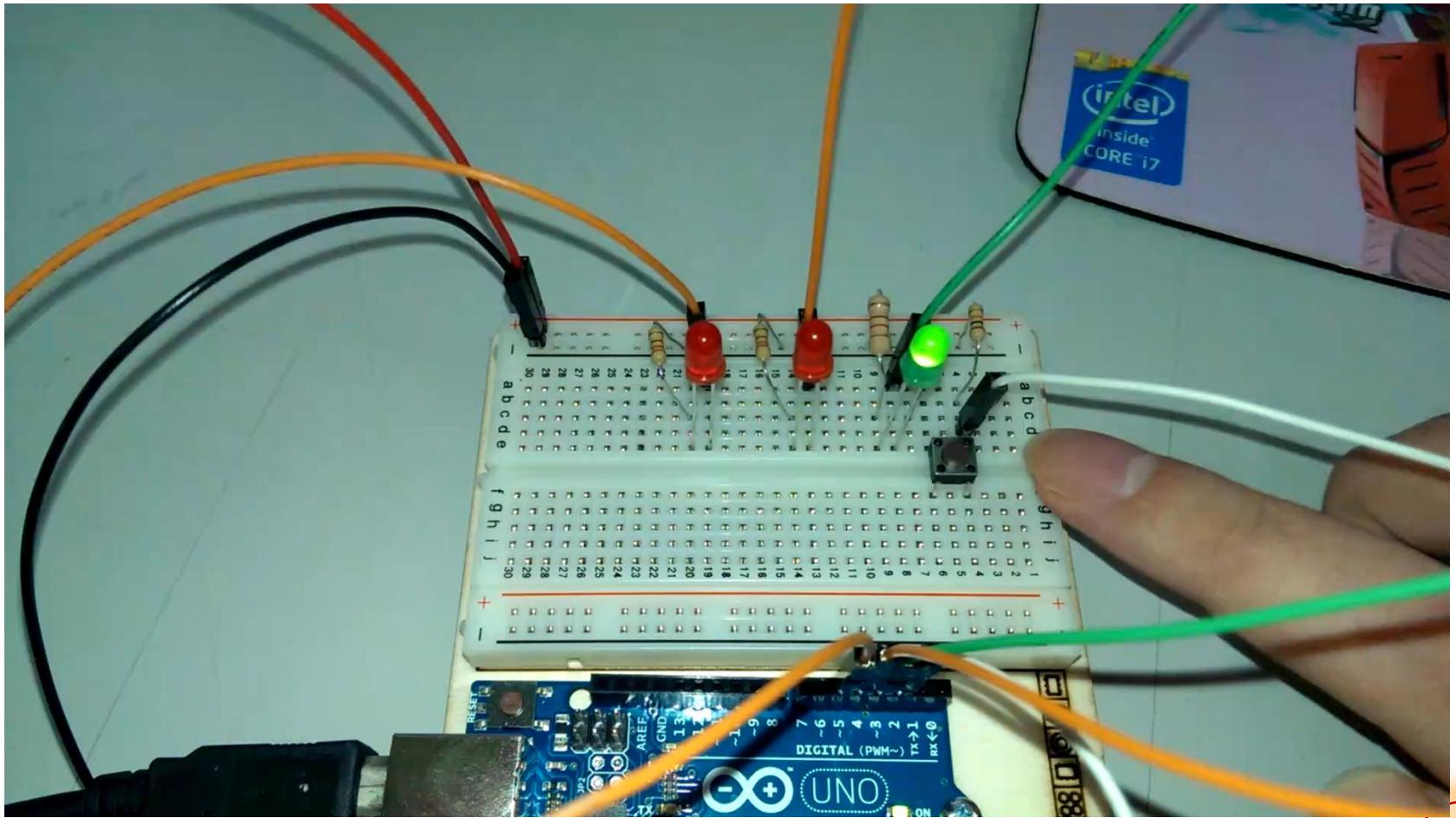
Discover: digital input and output, your first  
program, variables

# Spaceship Interface

- In this project, you'll be building something that could have been a spaceship interface in a 1970 science fiction movie.
- You'll make a cool control panel with a switch and lights that turn on when you press the switch.
- You can decide whether the lights mean "Engage Hyperdrive" or "Fire the lasers!".
- A green LED will be on, until you press a button. When the Arduino gets a signal from the button, the green light will turn off and 2 other lights will start

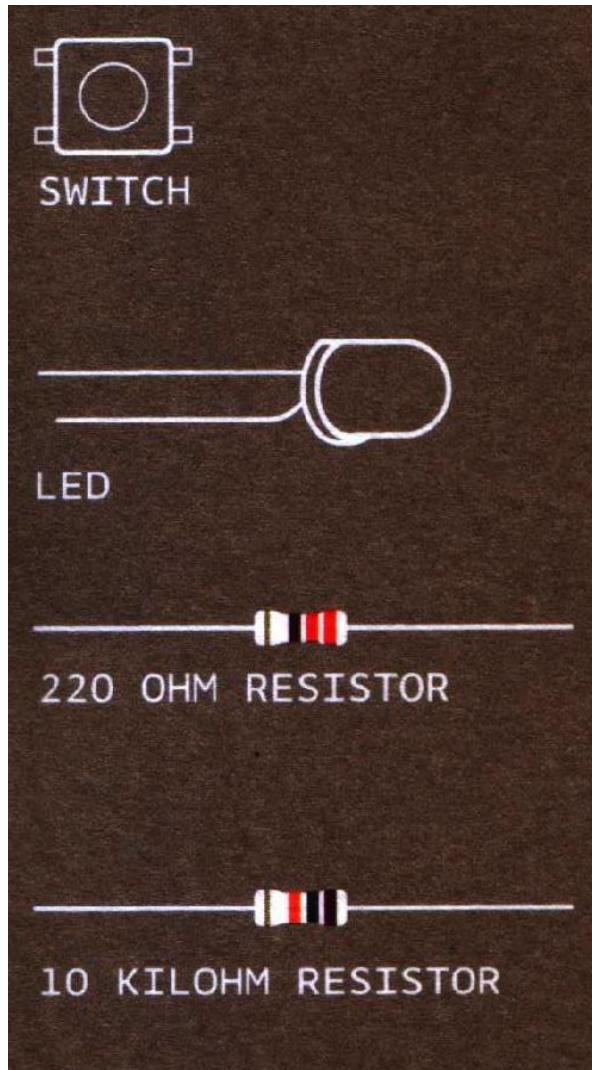


# Demo



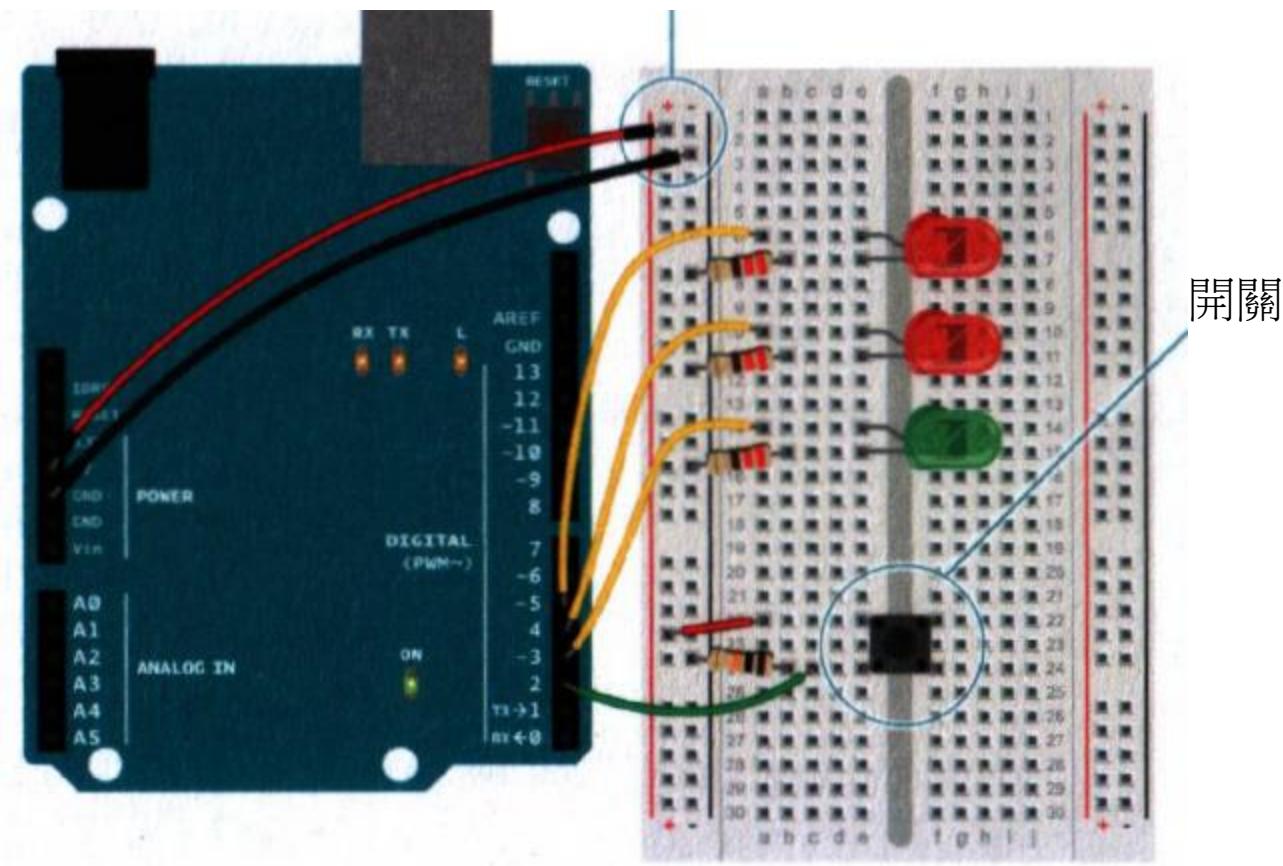
# Ingredients

- 1個LED燈(綠)
- 2個LED燈(紅)
- 1個10K電阻
- 3個220Ω電阻
- 1個開關

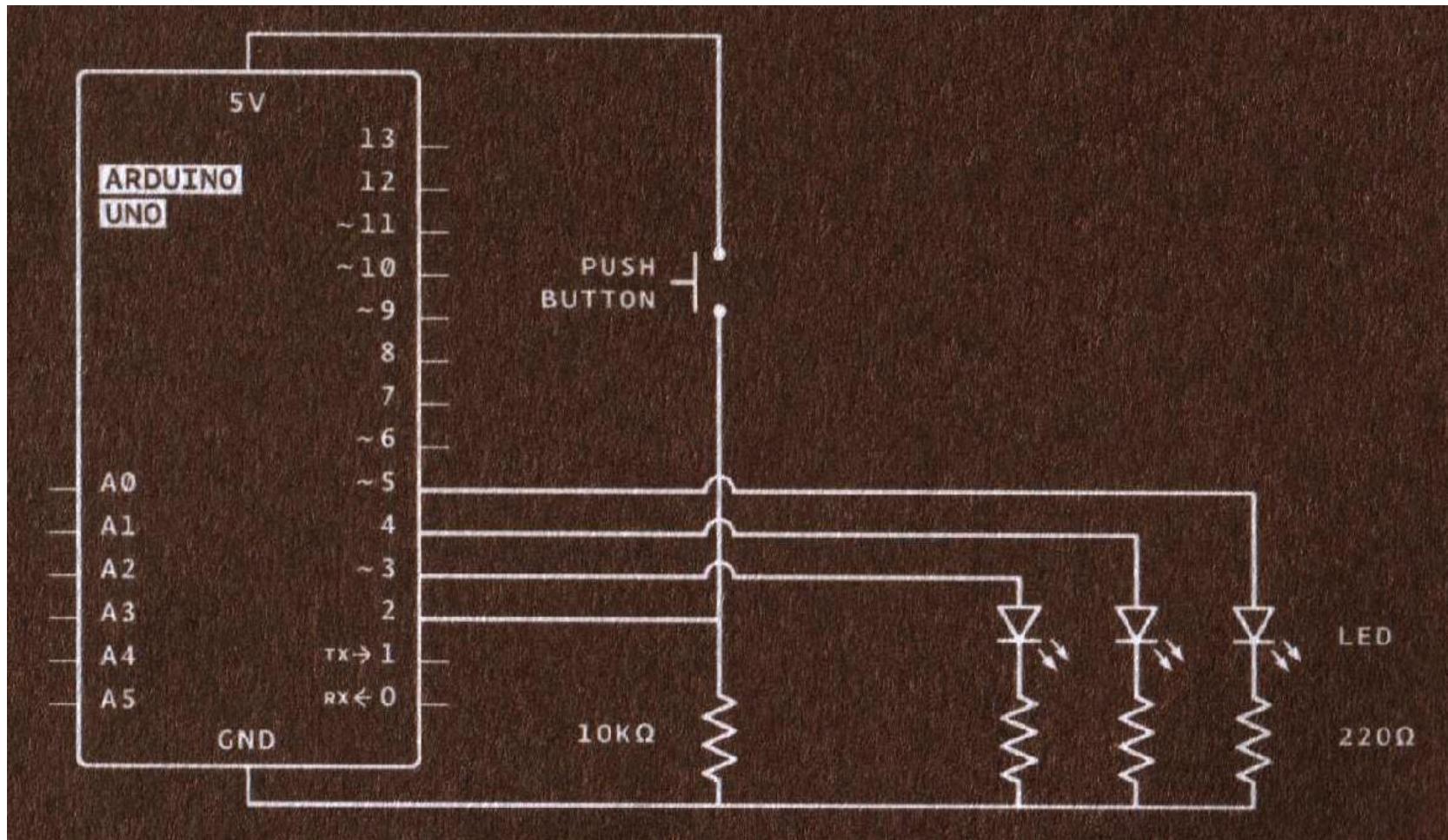


# The Top View of the Circuit

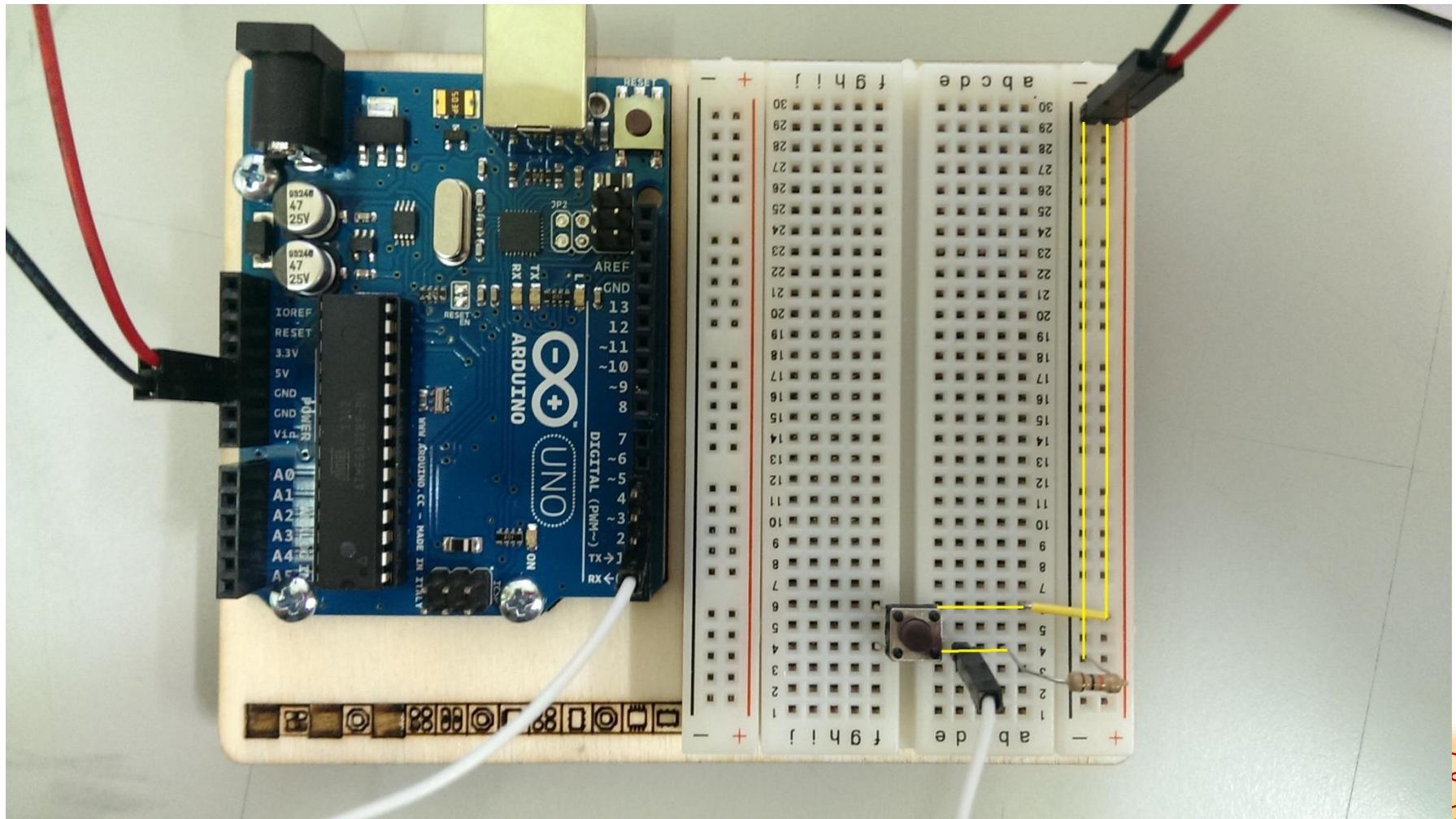
電源



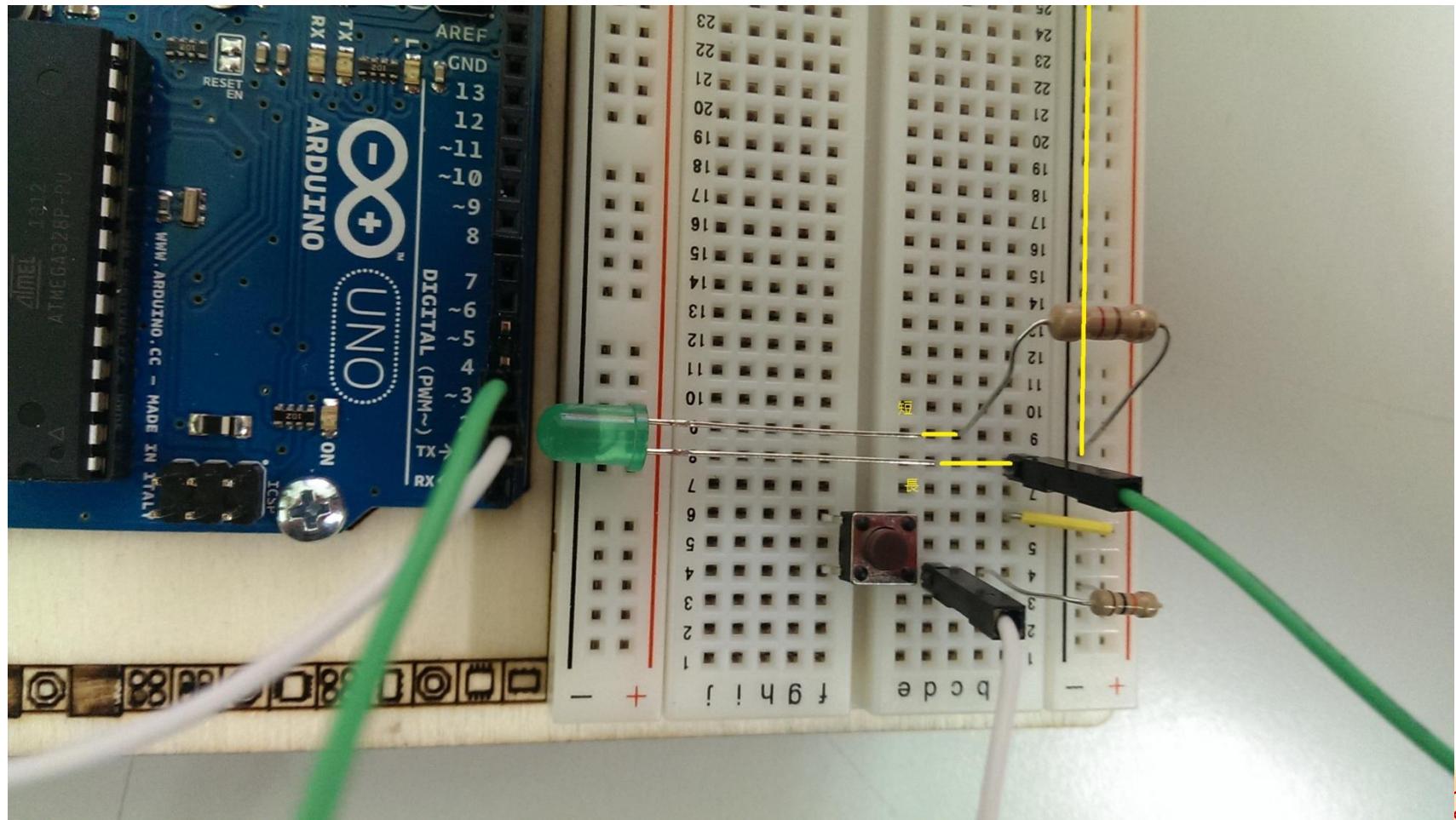
# Schematic Diagram



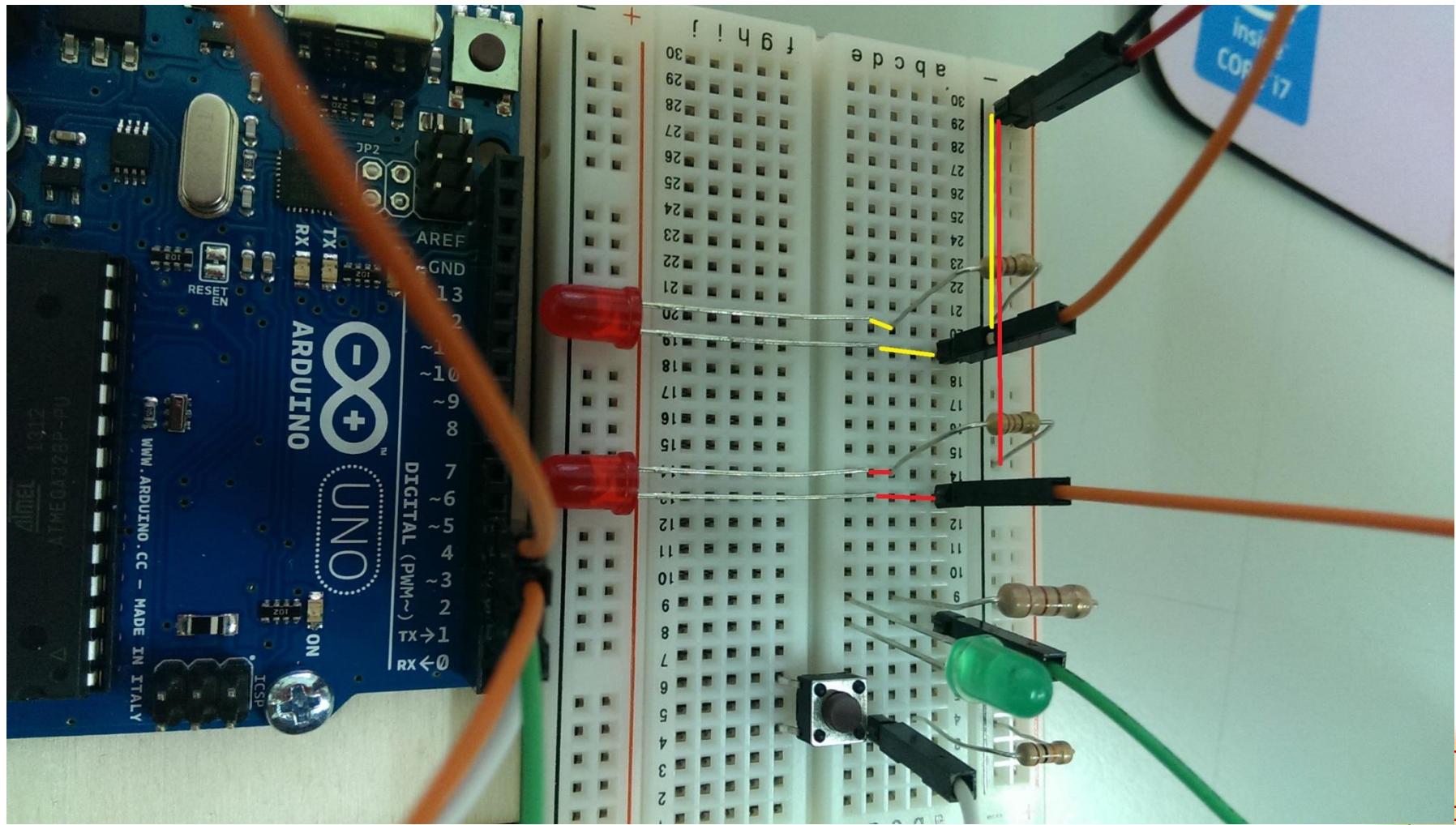
# Build The Circuit



# Build The Circuit



# Build The Circuit



# The Codes

//This is a comment and Arduino code is case sensitive

```
int switchState = 0;
```

```
void setup() {
```

```
    pinMode (2, INPUT);
```

```
//This pin is the button as INPUT.
```

```
    pinMode (3, OUTPUT);
```

```
    pinMode (4, OUTPUT);
```

```
    pinMode (5, OUTPUT);
```

```
//Those pins are the LED as OUTPUT.
```

```
}
```



p02\_SpaceshipInterface.ino



48/51

# The Codes (cont.)

```
void loop() { //this loop will be call thousand times a second  
    switchState = digitalRead( 2 );  
    if ( switchState == LOW ) {  
        digitalWrite ( 3, HIGH );  
        //green LED will be on.  
        digitalWrite ( 4, LOW );  
        digitalWrite ( 5, LOW );  
        //red LEDs will be off.  
    }  
}
```

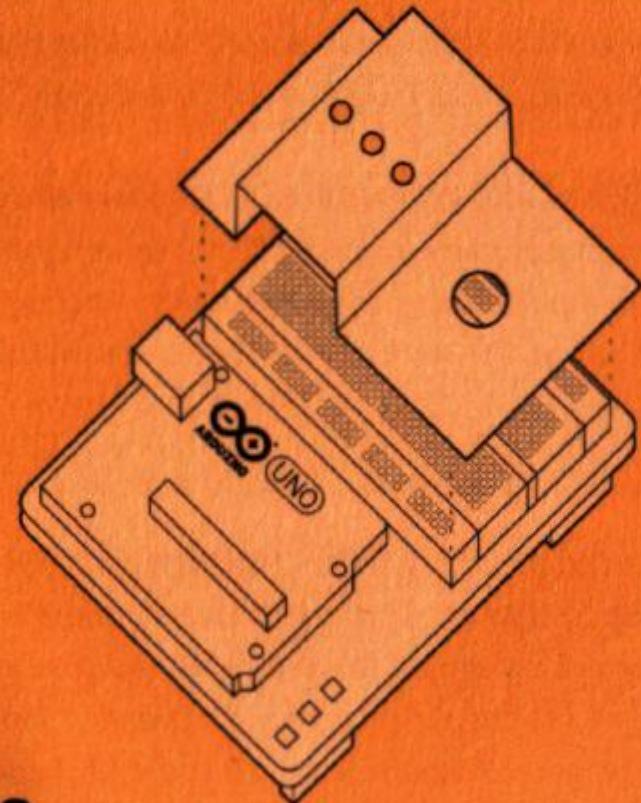


# The Codes (cont.)

```
else {  
    digitalWrite( 3, LOW );  
    digitalWrite( 4, LOW );  
    digitalWrite( 5, HIGH );  
    delay( 250 ); //It will completely halt for 250 msec  
    digitalWrite ( 4, HIGH );  
    digitalWrite ( 5, LOW );  
    delay ( 250 );  
}  
} // end of loop()
```

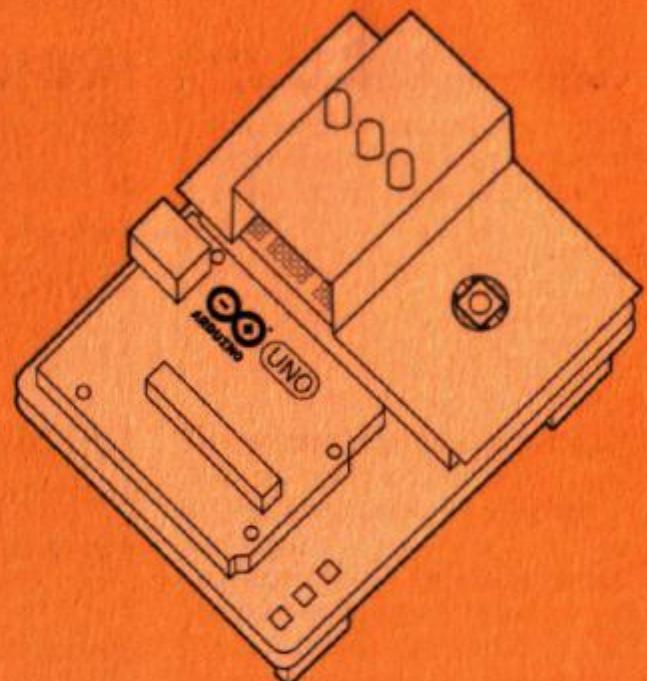


# Final Enhancement



1

Fold the pre-cut paper as shown.



2

Place the folded paper over the breadboard. The three LEDs and pushbutton will help keep it in place.