ARDUINO起源

ARDUINO起源

- 2005 年 1 月由當時米蘭互動設計學院的教授 David Cuartielles 和 Massimo Banzi（下圖）所設計出來。它使用了 Atmel AVR 單片機，是一個 開放原始碼 的軟硬體平台，構建於開放原始碼 simple I/O 介面板，使用類似 Java，C 語言的 Processing / Wiring 開發環境。

這五人創辦了ARDUINO

為何要做ARDUINO?

- 先從 Arduino 被發明的起源開始說起，Arduino 是共同創辦然 Banzi 還在意大利北部的一個設計學校教書時，因為學生沒有合適的學習的硬體（太貴，當時一片要接近 3000 元台幣），所以才開發出的開源電路板。
- ARDUINO共同創辦人的見解
  https://www.bilibili.com/video/av3577498/

ARDUINO TED

- 影片來源TUYTUBE
  https://www.youtube.com/watch?v=UoBUXOOdLXY&list=PLIkcMSIMWI7Rr_P4IVf9ktMOtTIIN7YhP

開源

- 「開源」是一種去中心化的創造，讓我們可以自由地發揮來滿足創造欲，理想的開源計劃應該是沒有領導者、沒有核心的，甚至開源可以不是一個計劃，它可以是一團 Code 或者一個隨意發想的機器人的半成品，就像自助餐桌上的菜餚一樣讓人隨時搭配享用，或者被另外一個「廚師（玩家）」二次烹飪，有的只是我們需要「吃（玩）」的這種需求。

案例分享

- MAKER
  https://makezine.com/category/technology/arduino/
  逗貓器
  生態養殖系統
  土壤警報缺水自動給水
  自動平衡車
  機器手臂

官方網站

- https://www.arduino.cc/

ARDUINO相關網站--使用者上傳作品

- https://www.thingiverse.com/
  http://makezine.com/?s=arduino
- http://www.instructables.com/tag/type-id/category-technology/channel-arduino/

ARDUINO 開發介面

- 線上模擬平台

https://www.tinkercad.com

- 官方下載位置
- https://www.arduino.cc/en/Main/Software

ARDUINO UNO

ARDUINO UNO硬體介紹

重新執行(RESET)

數位接腳

LED燈顯示13號接腳
是高位還是低位

USB控制晶片

USB接頭傳輸資料及供電

TX 發送資料LED

RX 接收資料LED

穩壓器

輸入電壓5V

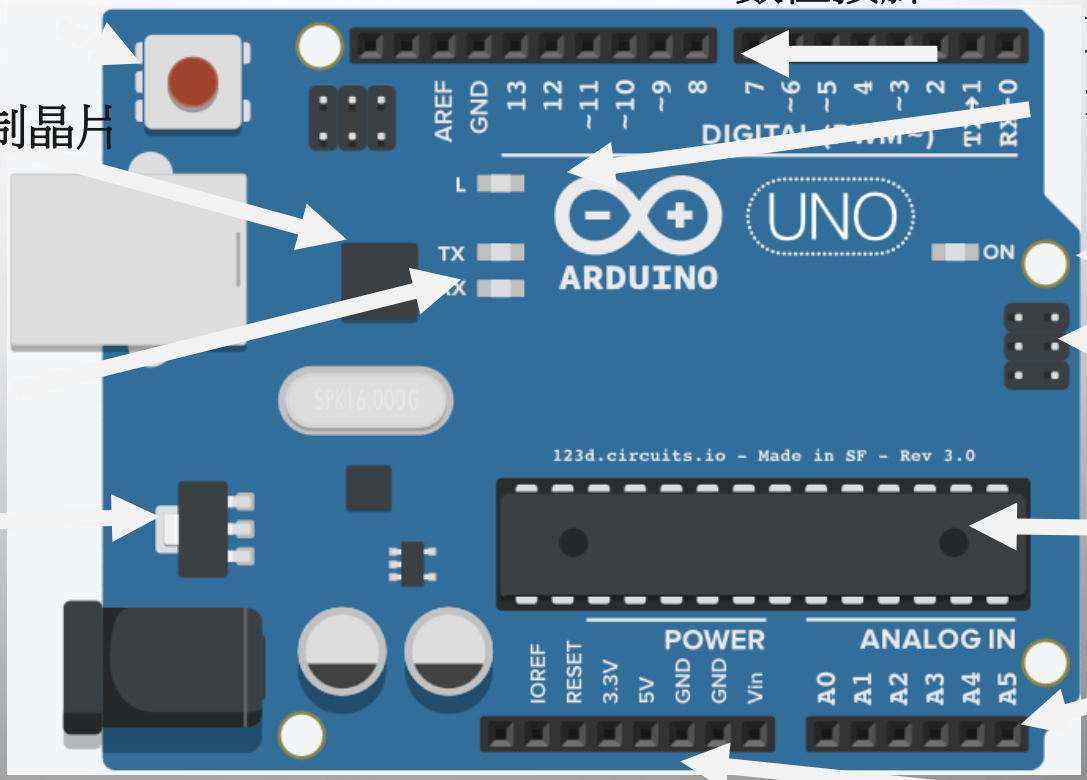
電源燈

ICSP燒錄接腳

ATMAGE328P-PU

類比接腳

電源控制

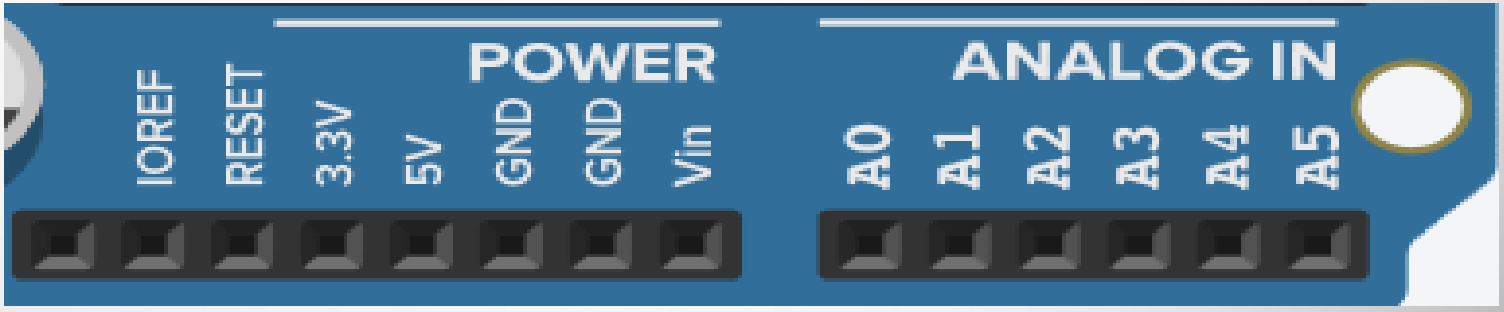


ARDUINO UNO 數位接腳

0-RX 接收資料用的接腳
1-TX 發送資料用的接腳
2到13 為數位接腳可接收(INPUT)也可發送(OUPUT)
3、5、6、9、10、11 接腳前方有一個'~'表示可接收PWM資訊
GND-接地接腳
AREF-模擬參考

*PWM:Pulse-width Modulation 即脈波寬度調整或稱頻寬調整

ARDUINO UNO 類比及電源接腳

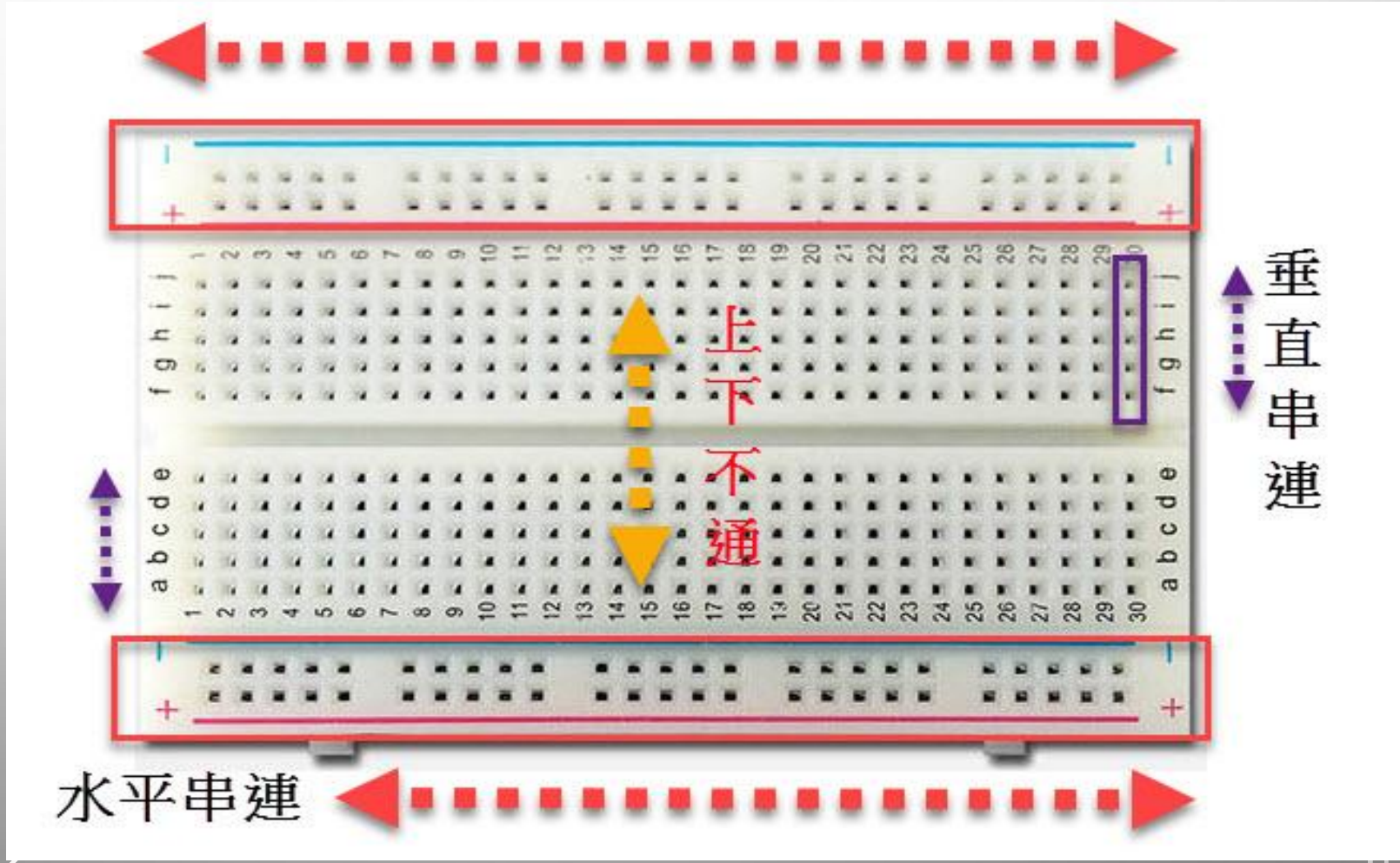


Analog IN-六組類比接腳(A0、A1、A2、A3、A4、A5)
POWER-
Vin 電源輸入
GND 接地
5V　5V電源輸出
3.3V　3.3電源輸出
RESET　重新執行
IOREF 取得目前工作電壓

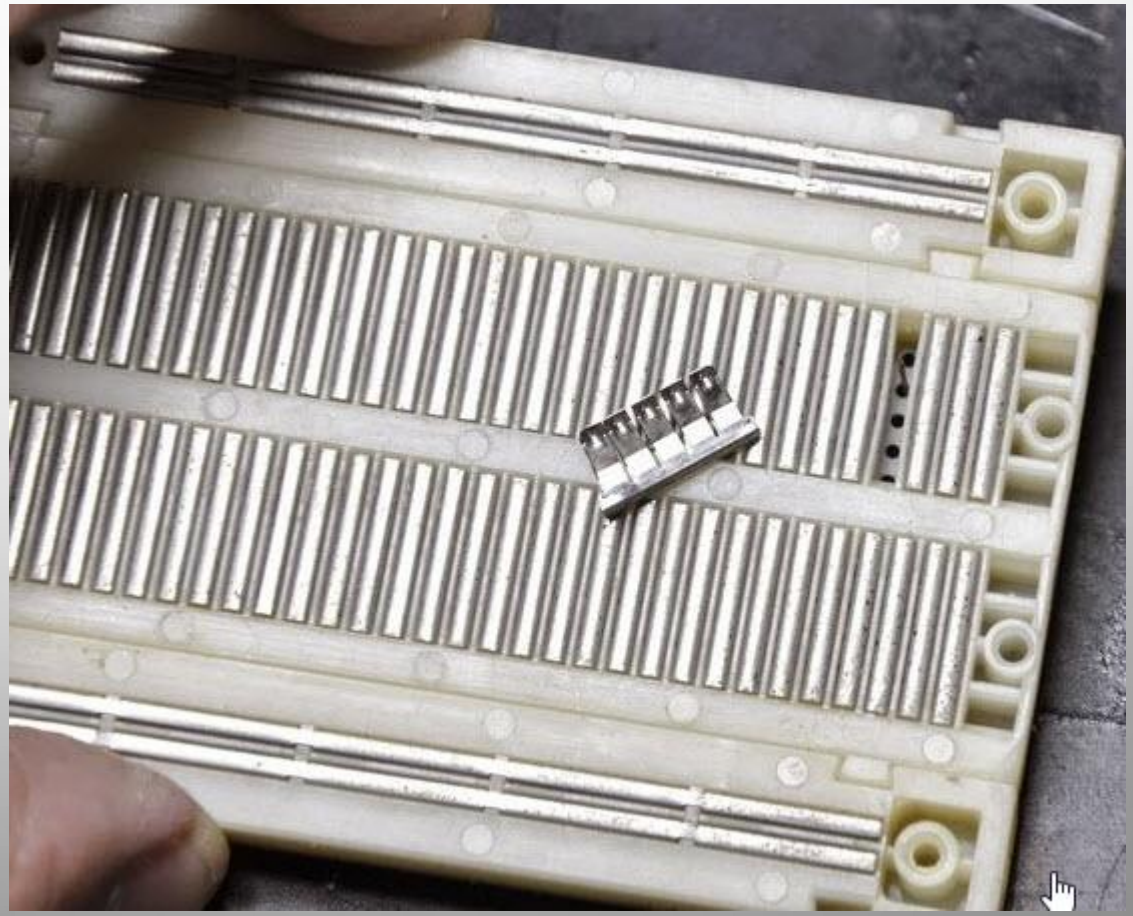
認識麵包板

BREADBOARD

認識麵包板



麵包板背面架構



HTTPS://WWW.TINKERCAD.COM

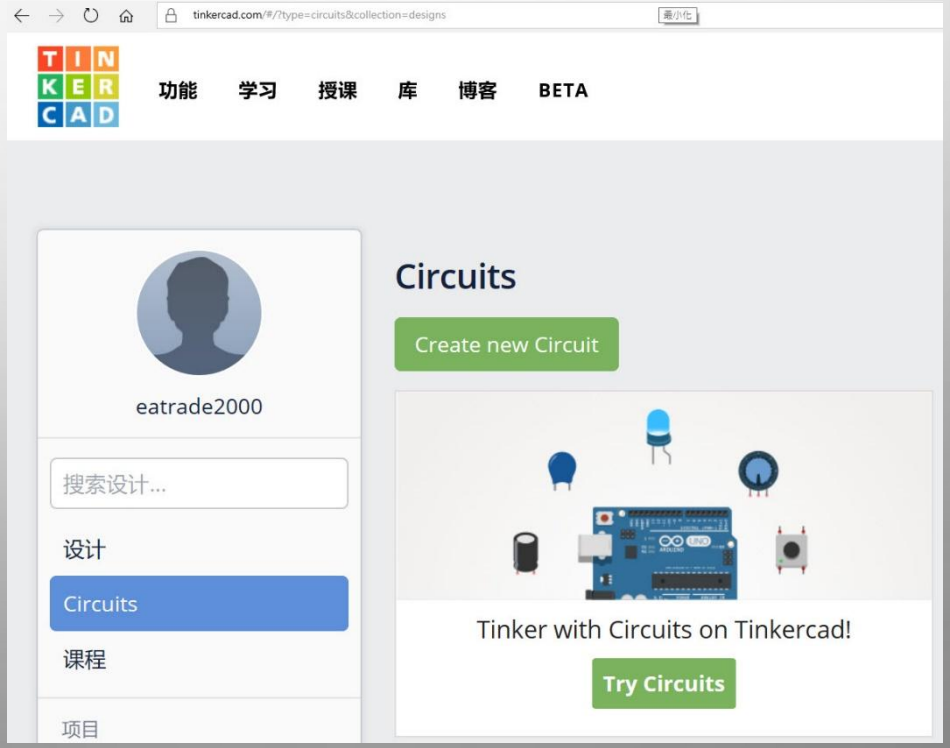
線上開發平台

登入線上開發平台 (HTTPS://WWW.TINKERCAD.COM)

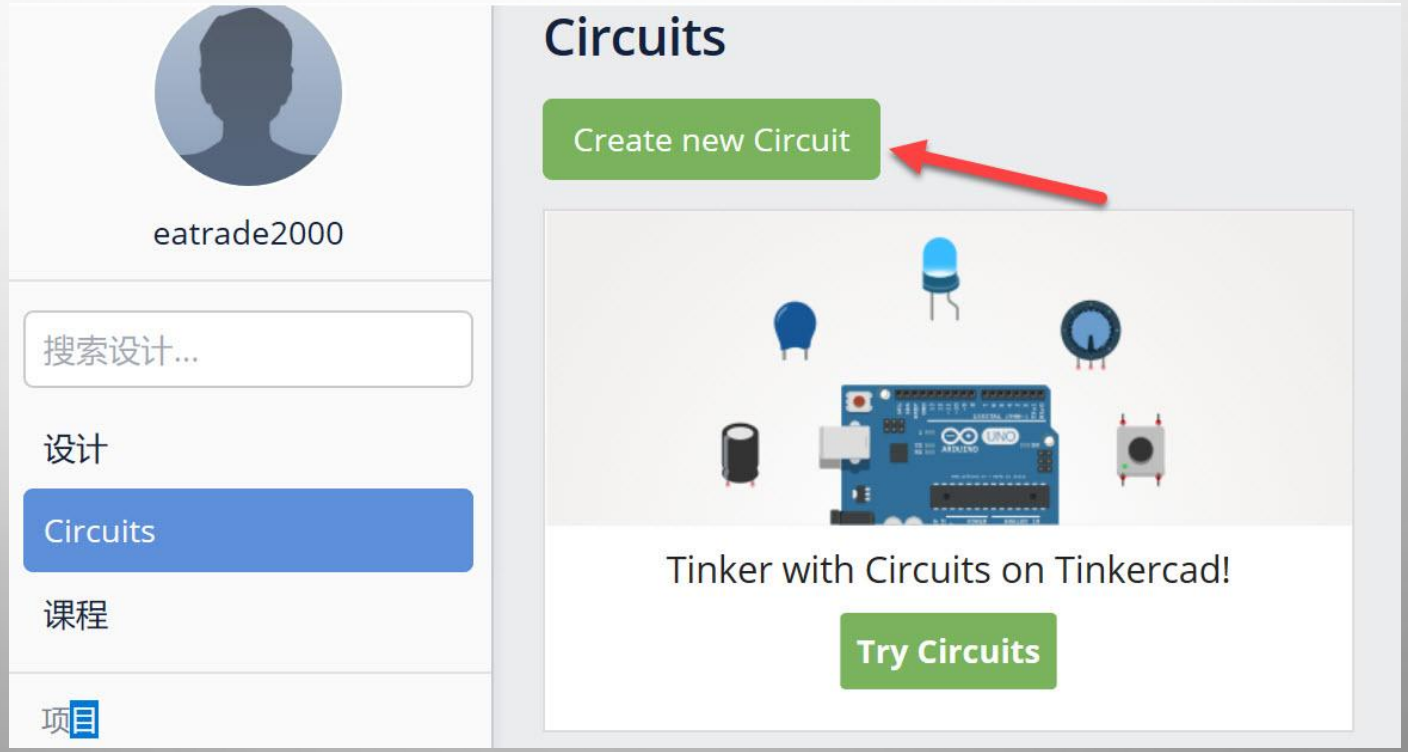
- 申請帳號

登入後需要再重新連結

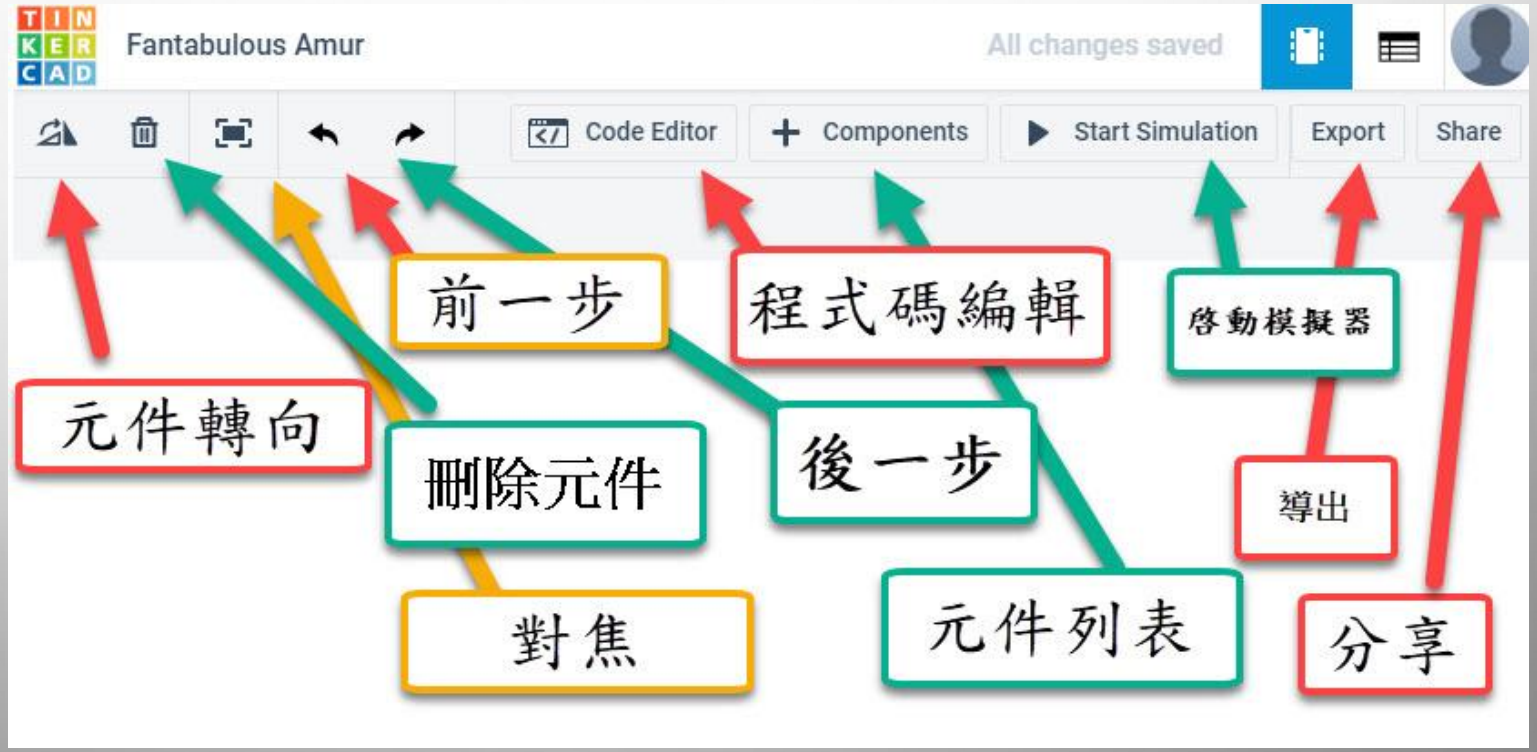
- https://www.tinkercad.com/#/?type=circuits&collection=designs



新增一個方案

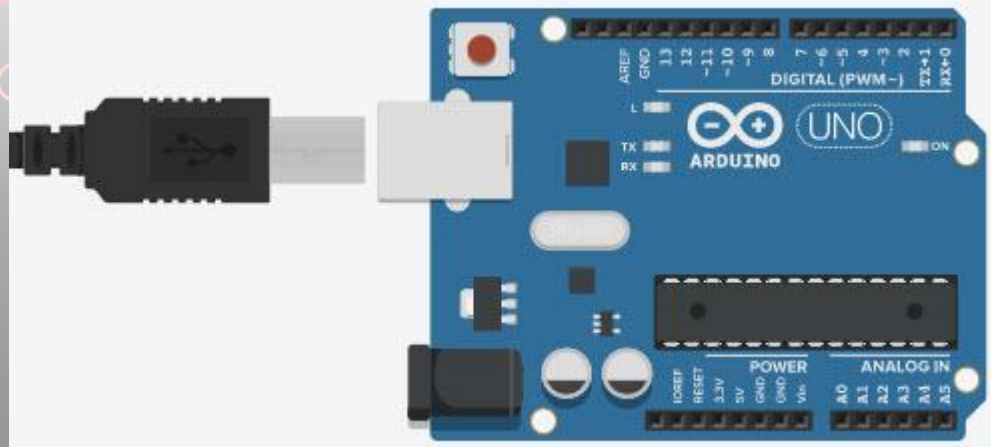

eatrade2000
搜索设计...
设计
Circuits
课程
项目
Circuits
Create new Circuit
Tinker with Circuits on Tinkercad!
Try Circuits

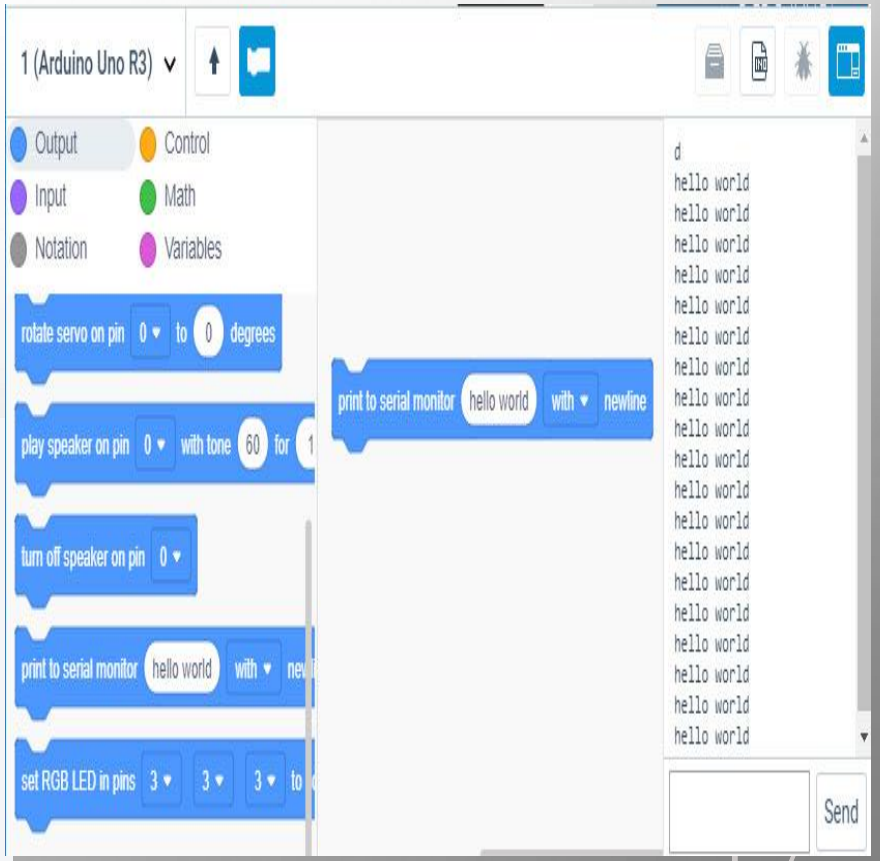
CIRCUIT介面


Fantabulous Amur
All changes saved
Code Editor
Components
Start Simulation
Export
Share
元件轉向
前一步
刪除元件
對焦
程式碼編輯
後一步
元件列表
啓動模擬器
導出
分享

我的第一個程式HELLO WORLD

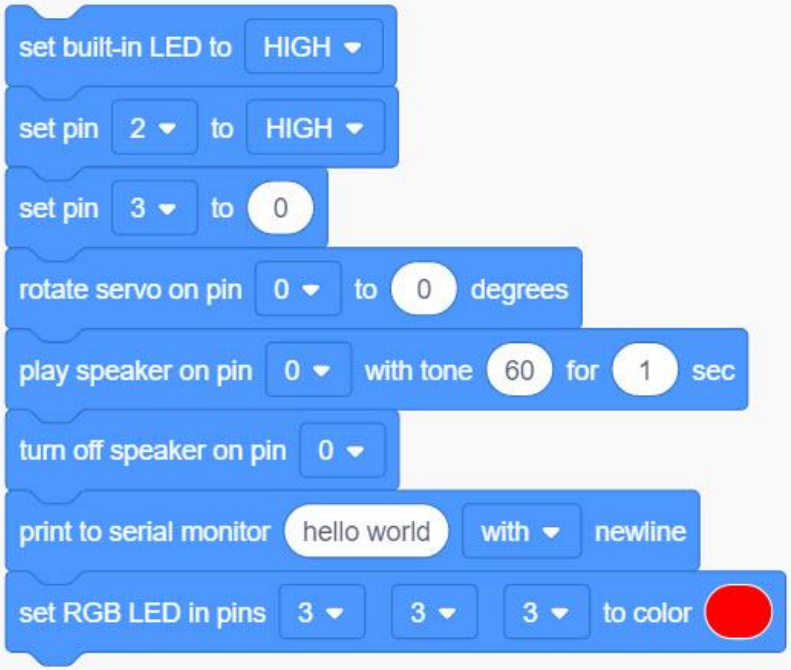
- 元件:ARDUINO UNO R3
- 程式方塊:輸出PRINTLN



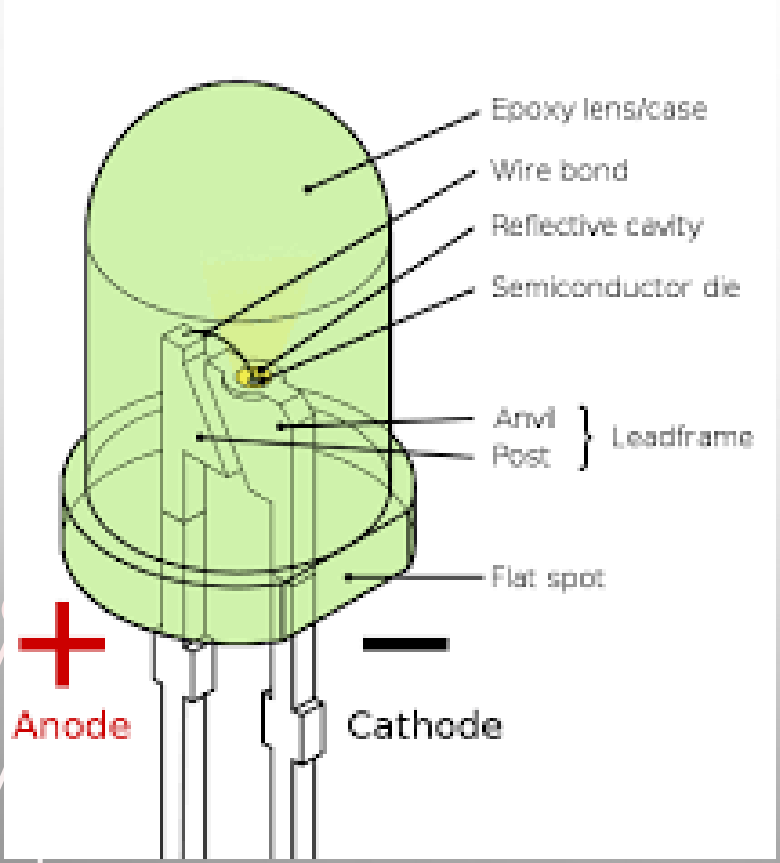


輸出(OUTPUT)

- 數位輸出　HIGH　LOW
- 類比輸出　0　1023



LED 燈(發光二極體)


Epoxy lens/case
Wire bond
Reflective cavity
Semiconductor die
Anvil
Post
Leadframe
Flat spot
Anode
Cathode

發光二極體規格

顏色	λ波長（nm）	順向偏壓（V）
紅外線	>760	< 1.9
紅	760至610	1.63-2.03
橙	610至590	2.03-2.10
黃	590至570	2.10-2.18
綠	570至500	2.18-4
藍	500至450	2.48-3.7
紫	450至380	2.76-4

- LED電流使用約在20ma

  R=V/I

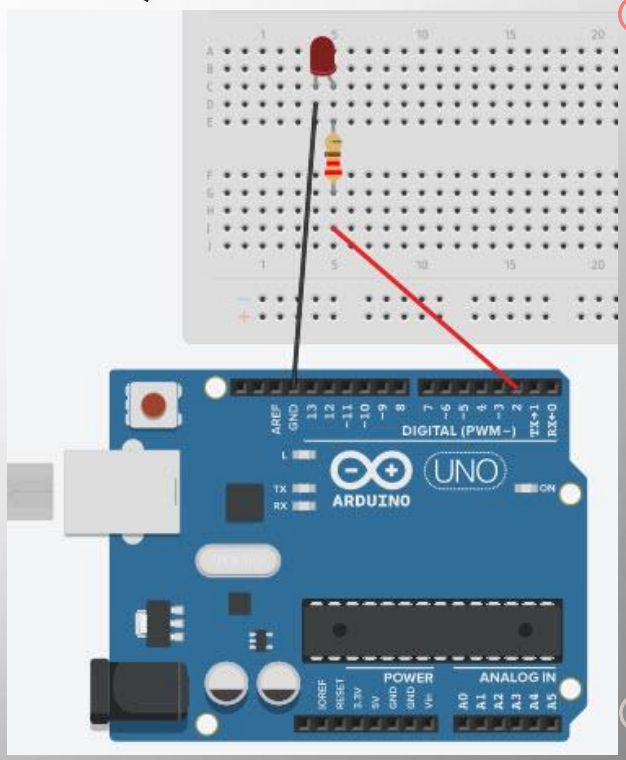
  V=5V-2V

  R=3/0.002=150Ω

  為保險起見一般使用220Ω電阻

LED 數位輸出實驗(MY_LED1)

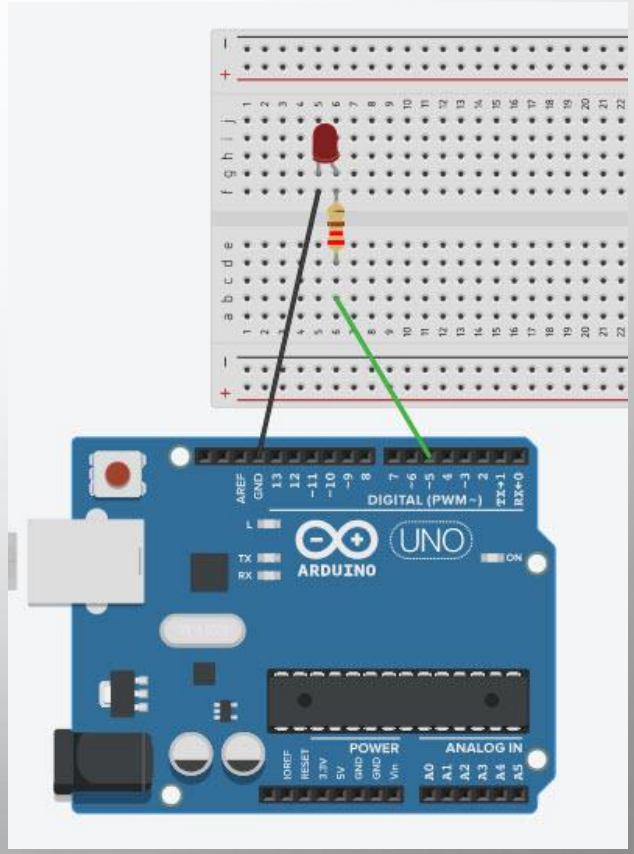
材料:
ARDUINO UNO R3
麵包板
220Ω電阻
LED



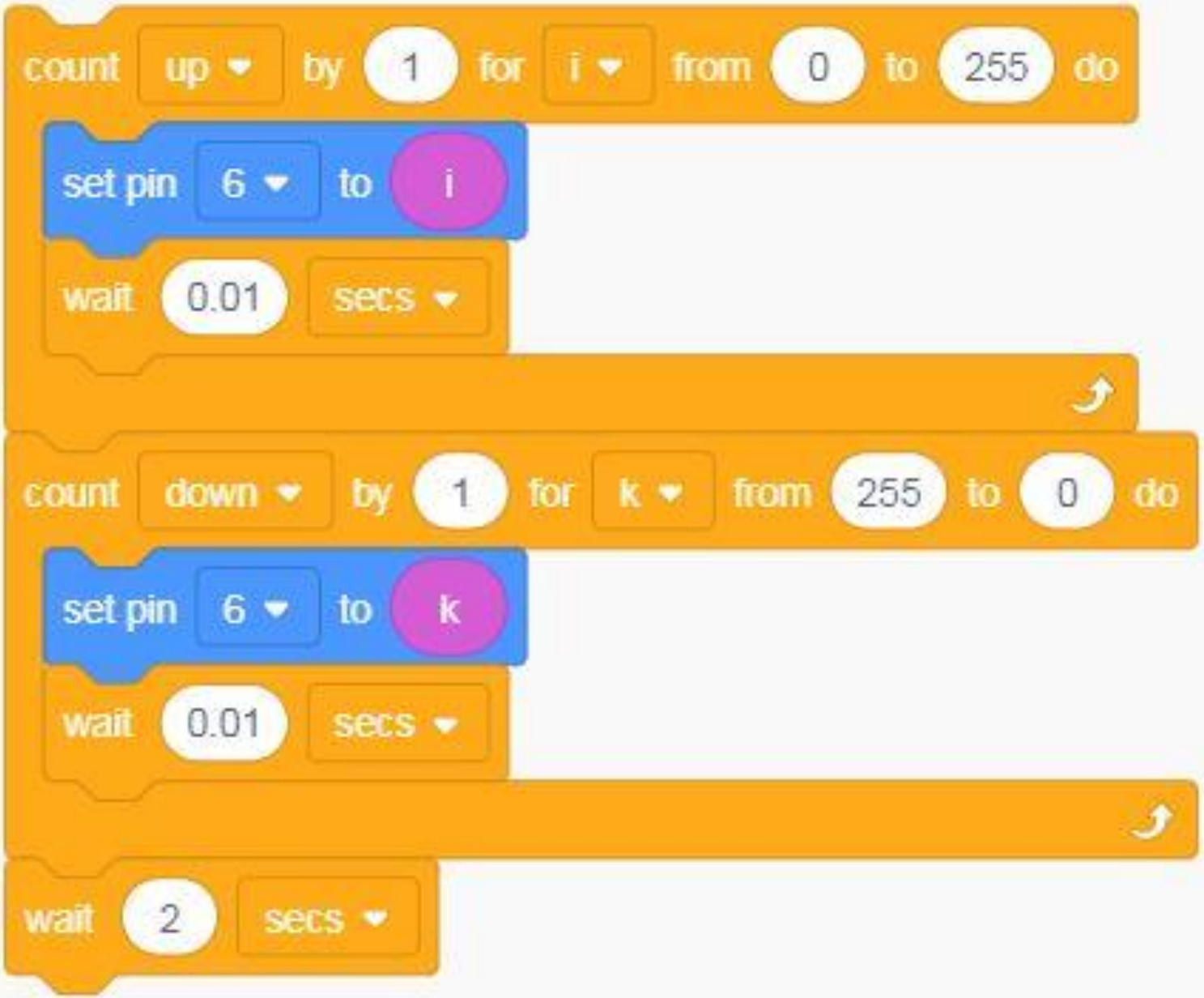
LED PWM輸出實驗(MY_LED2)

材料:
ARDUINO UNO R3
麵包板
200Ω電阻
LED

實作: 利用數位接腳輸出PWM數值點亮
LED 燈，LED燈範圍0~255



課堂練習:利用循環
從0一直加到255
再從255一直減回0
(MY_LED3)


count up by 1 for i from 0 to 255 do
set pin 6 to i
wait 0.01 secs
count down by 1 for k from 255 to 0 do
set pin 6 to k
wait 0.01 secs
wait 2 secs

輸入 INPUT

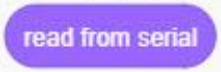
讀取數位接腳數值

讀取類比接腳數值

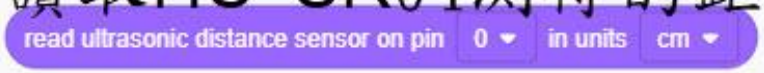
讀取伺服馬達接腳數值

是否有數列傳回

從數列取回數值

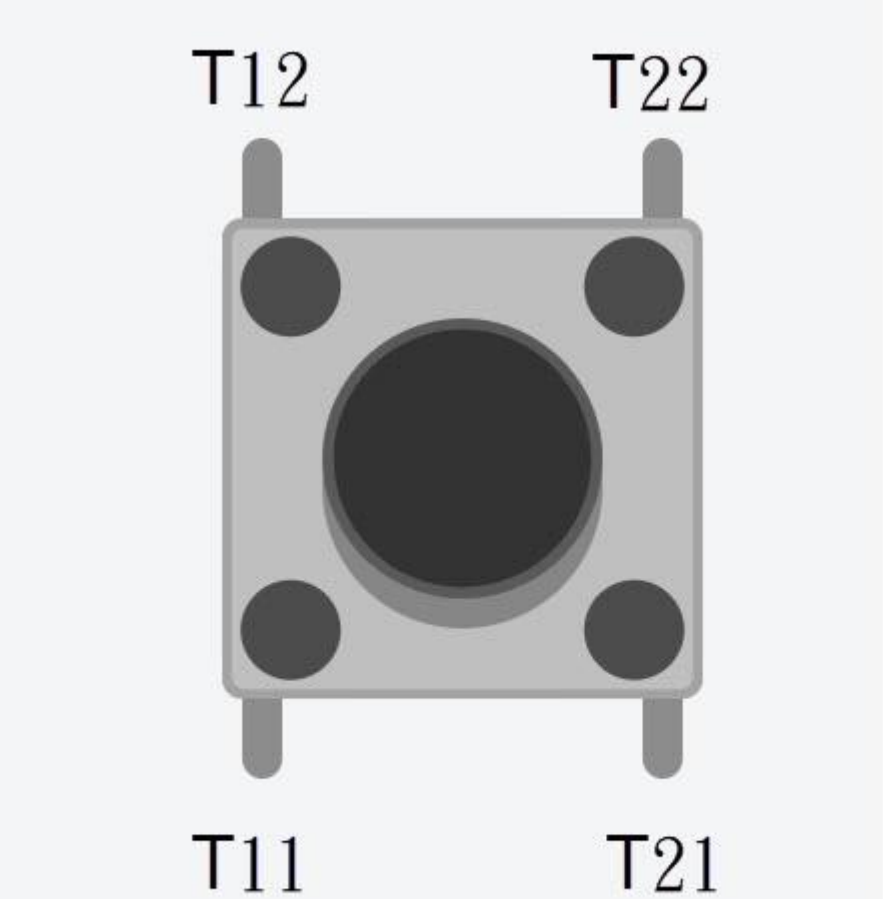


讀取HC-SR04測得的距離

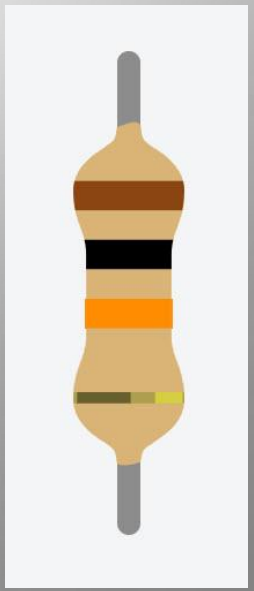


讀取温度感測器的度數

PUSHBUTTON 按鍵開關



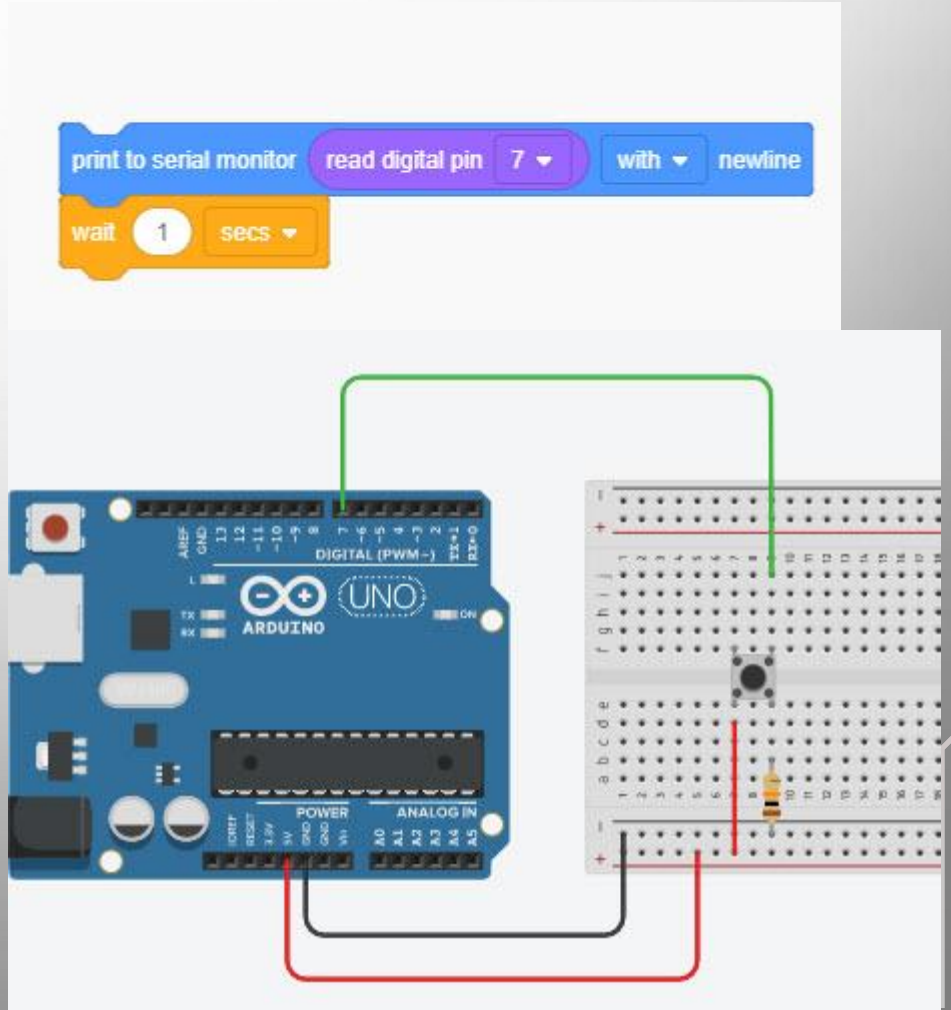
為順利取得偏向電壓，會在接地端加上10K歐姆電阻



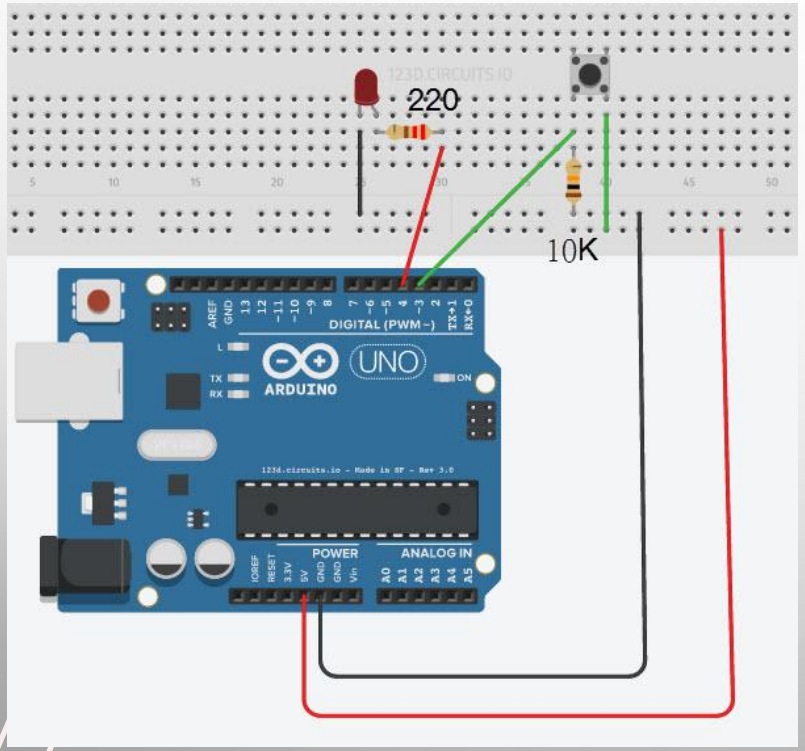
PUSHBUTTON(MY_PUSHBUTTON) 讀取按鍵開關訊號

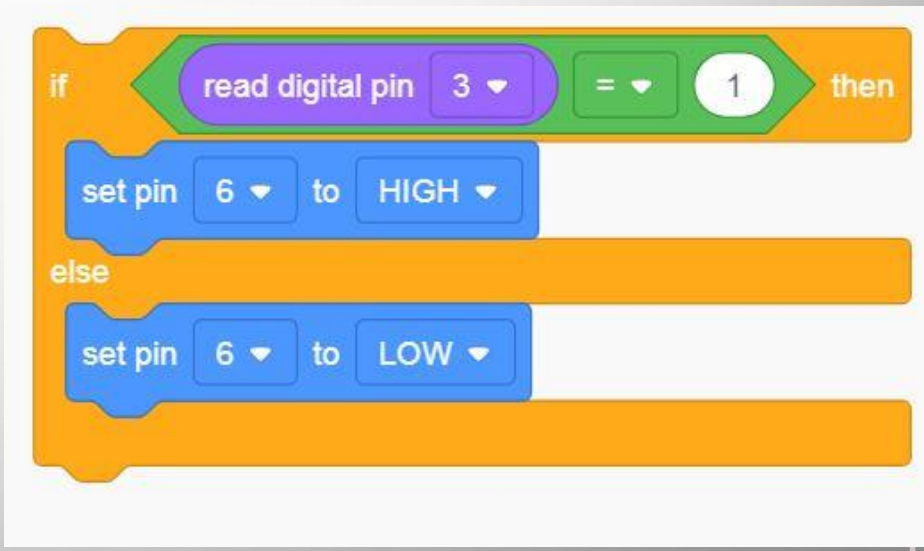
- 材料:

Quantity	Component
1	Arduino Uno R3
1	Pushbutton
1	10 kohm Resistor



實作 DIGITAL READ WRITE
讀取按鍵開關送5V點亮LED





作業練習

- 控制LED開一秒關一秒(閃爍)

- 三個LED燈及三個200Ω電阻
  控制由左向右陸續開停2秒後再由右向左陸續關

數位訊號與類比訊號的差異

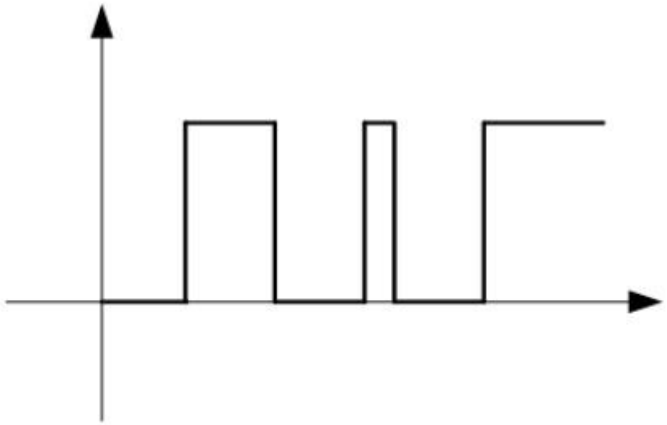


圖　數位訊號波形

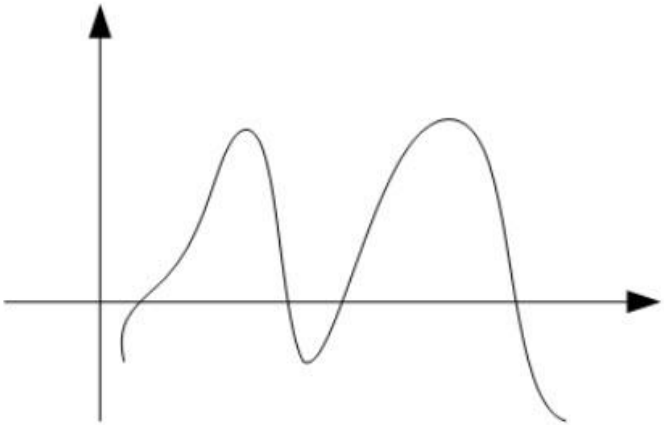
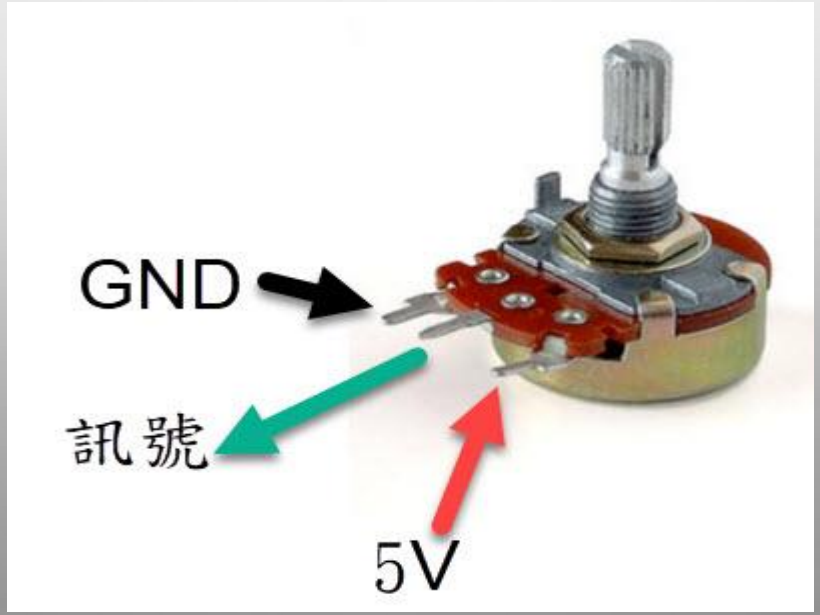


圖　類比訊號波形

可變電阻(電位器) POTENTIOMETER

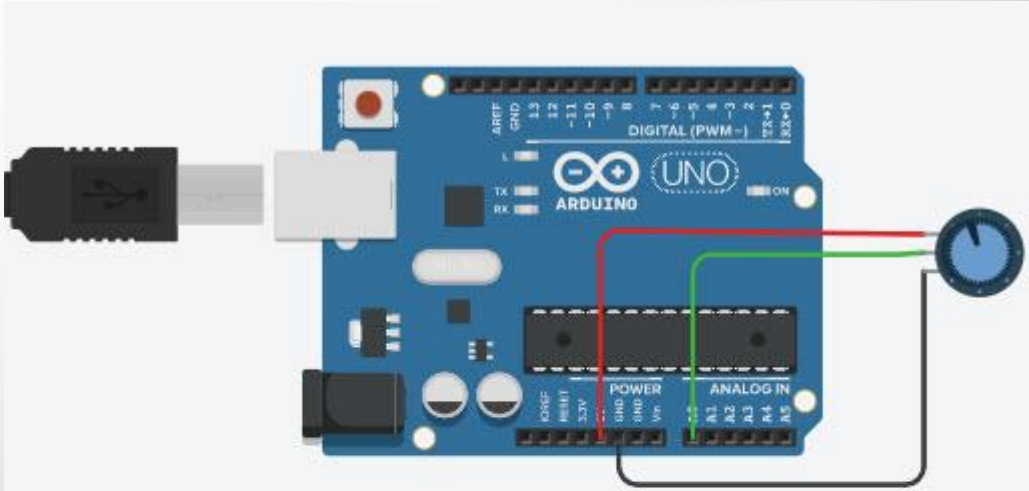
- 音量控制用的電位器附開關，可兼作音量與電源開關的功能，此時通常是在音量最小的一端附帶關閉電源。

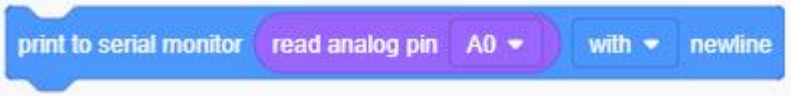


可變電阻(MY_POTENT)
讀取可變電阻的數值(類比接腳)

- 材料:

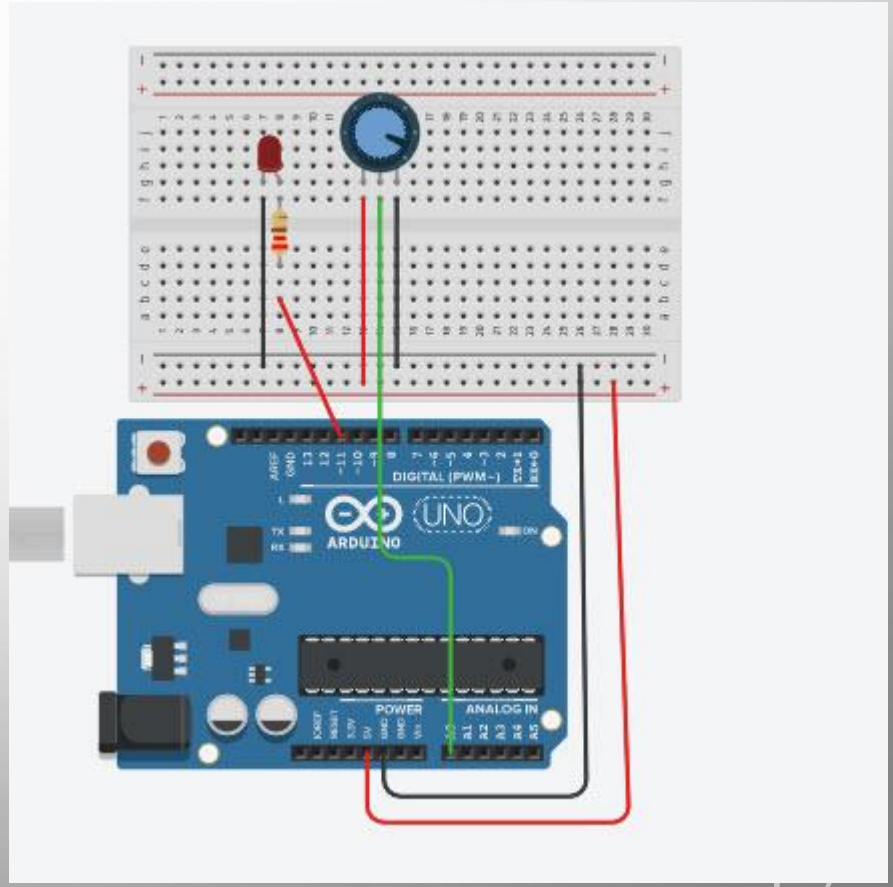
Quantity	Component
1	Arduino Uno R3
1	250 kOhm, Potentiometer



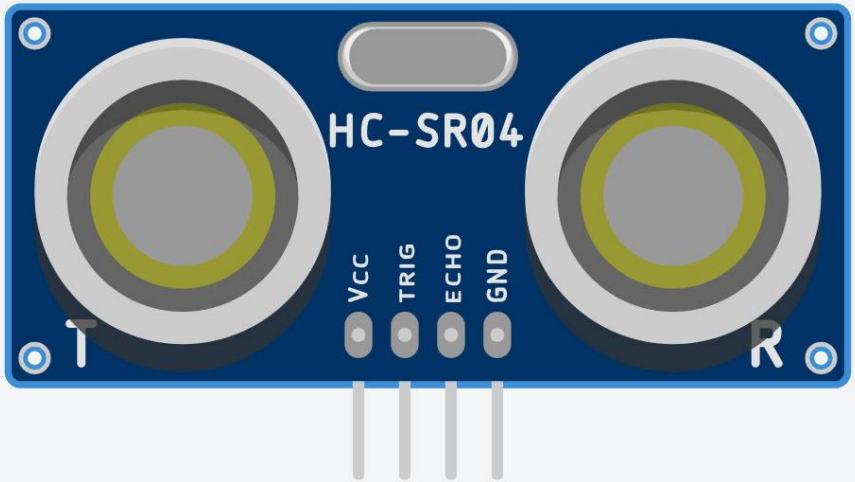


控制燈光明暗(MY_LEDCONTROL)

Quantity	Component
1	Red LED
1	220 ohm Resistor
1	10 kOhm, Potentiometer
1	Arduino Uno R3



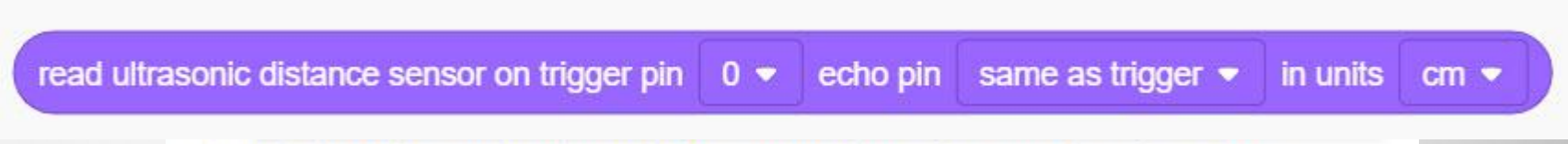
HC SR04



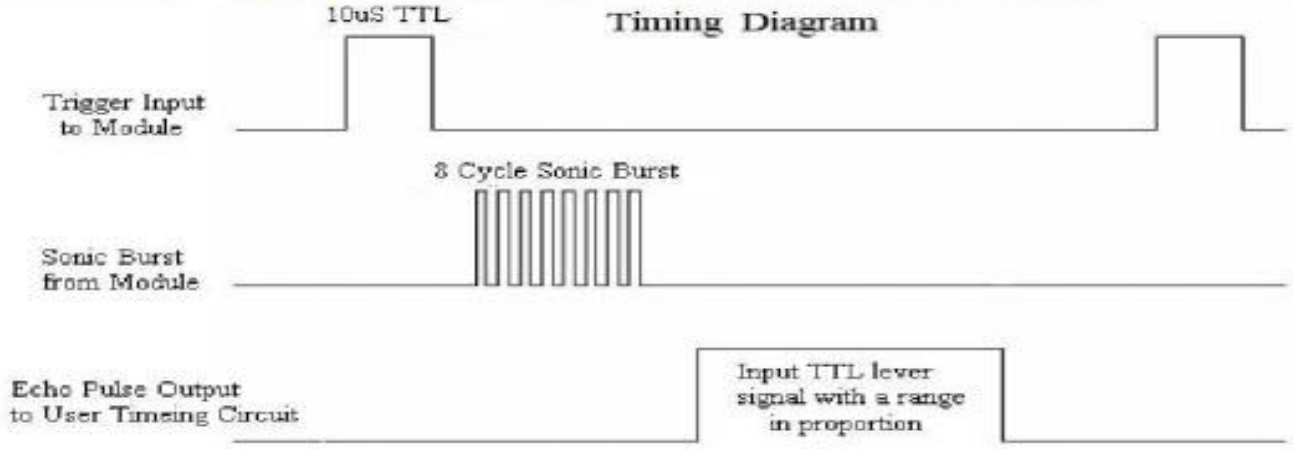
Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

HC SR 04運作方式

- uS=微秒=10的-6次方秒
  距離=ECHOPIN讀的訊號的時間*340M/S/2=levelTime*((1/170)*10000) CM=ledelTime*58.8

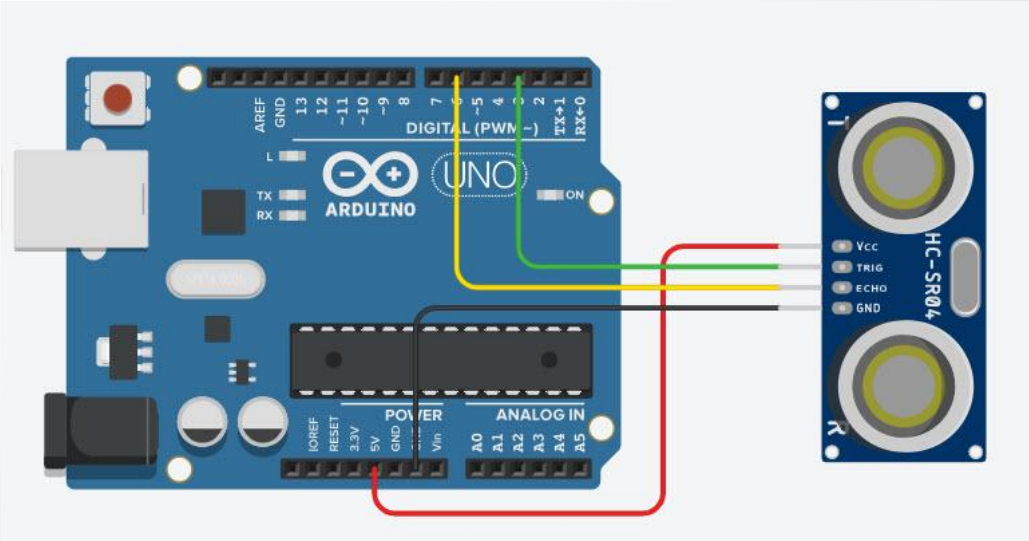


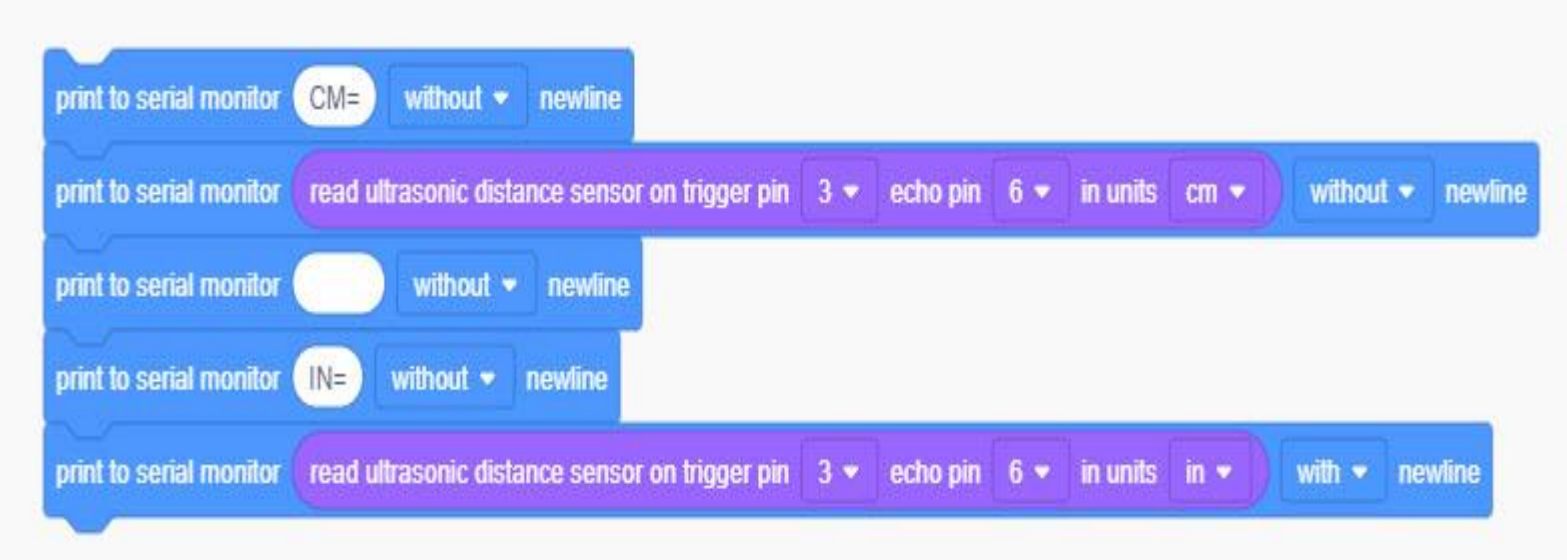
The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: uS / 58 = centimeters or uS / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



HC-SR04测距

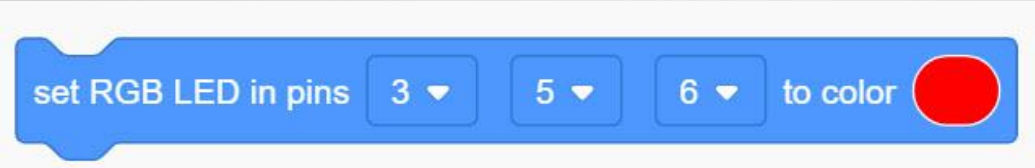
- 材料:
- ARDUINO UNO R3
- HC-SR04

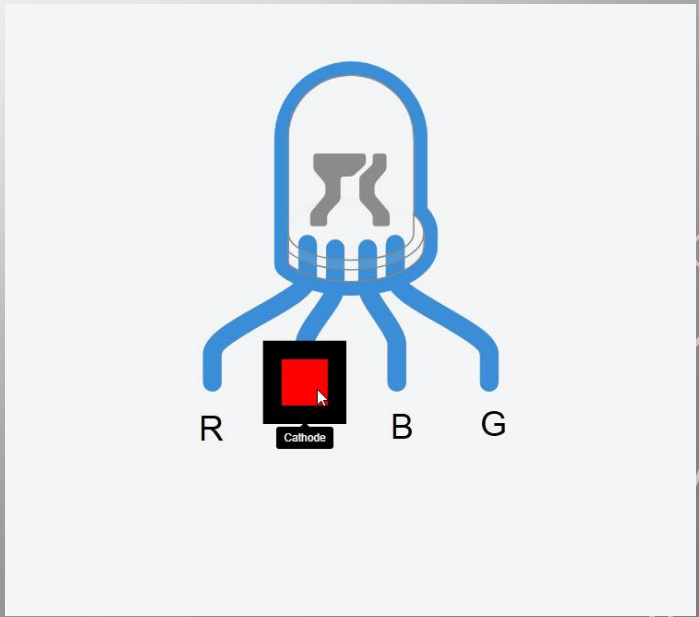




全彩LED 共用陰極

- 通過R 、G 、B 三個引腳的PWM 電壓輸入可以調節三種基色（紅/ 藍/ 綠）的強度從而實現全彩的混色效果。用Arduino 對模組的控制可實現酷炫的燈光效果。

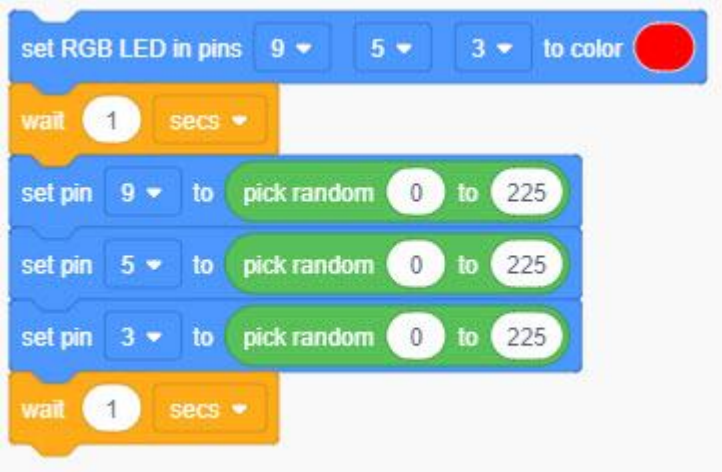


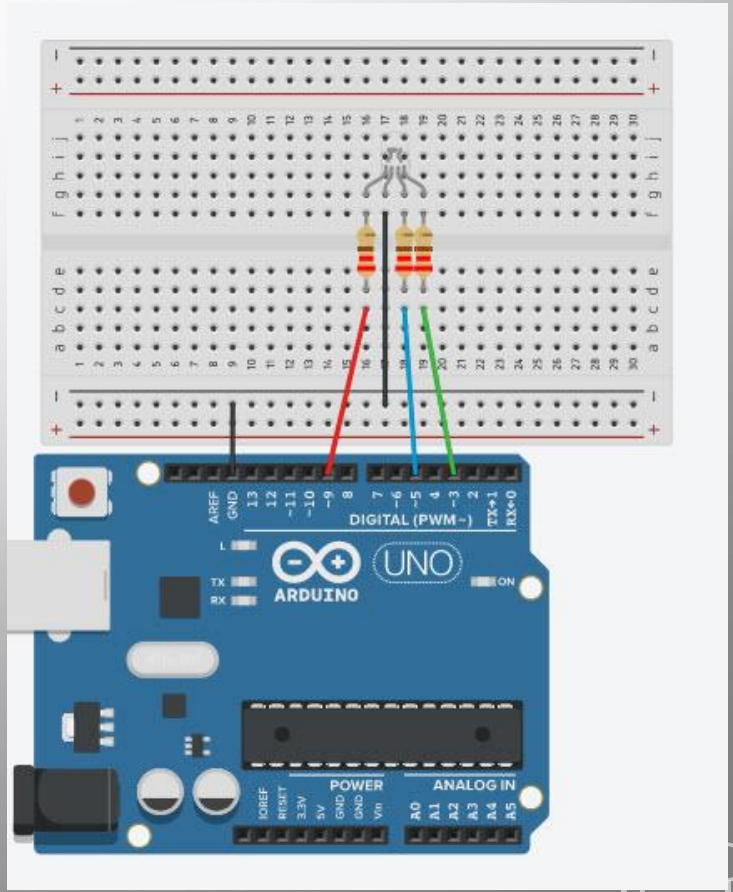


全彩LED (MY_LEDRGB)

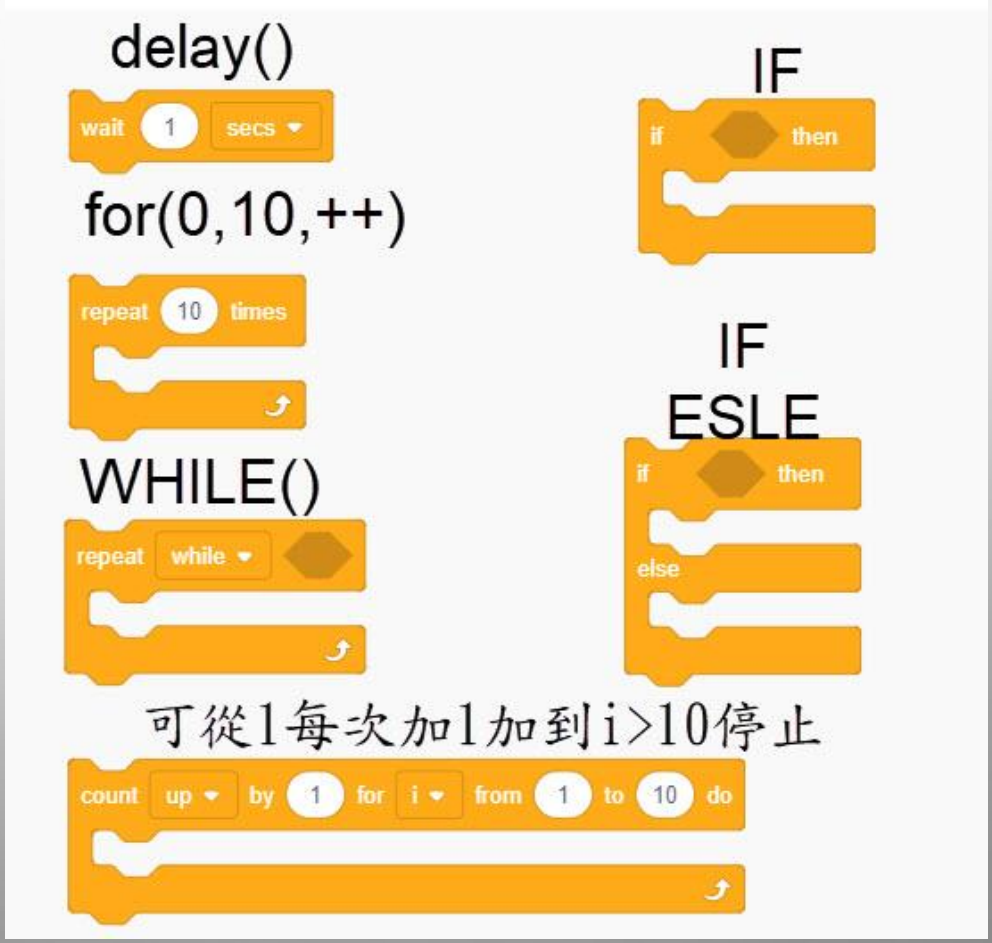
使用全彩LED利用亂數改變顏色

Quantity	Component
1	Arduino Uno R3
1	LED RGB
3	220 ohm Resistor



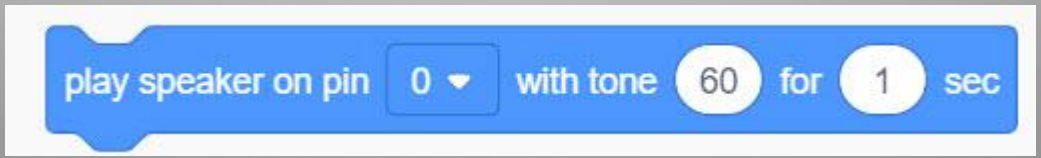


控制 CONTROL


delay()
wait 1 secs
for(0,10,++)
repeat 10 times
WHILE()
repeat while
IF
if then
IF
ESLE
if then
else
可從1每次加1加到i>10停止
count up by 1 for i from 1 to 10 do

蜂鳴器

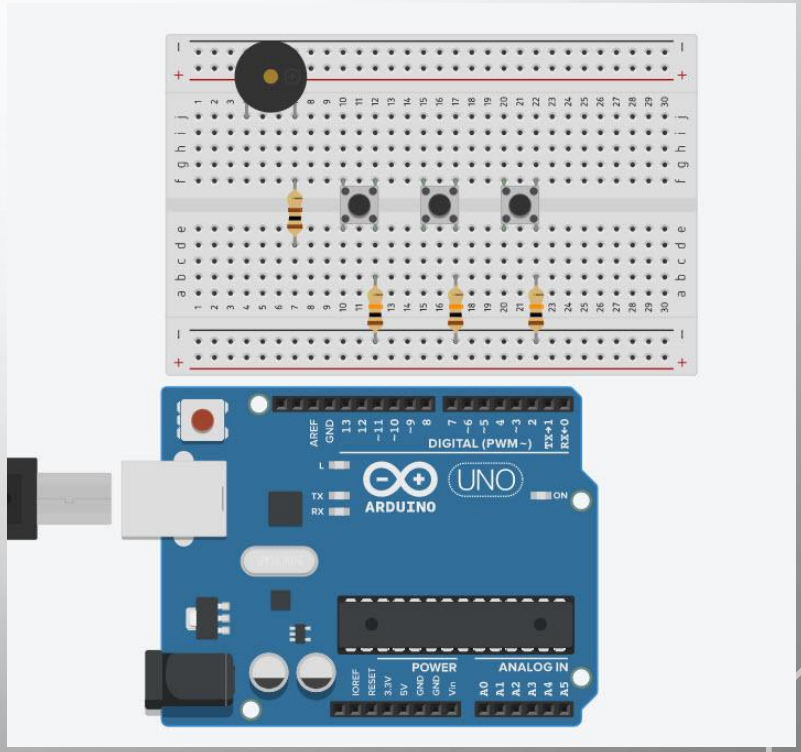
- 蜂鳴器（Buzzer）是產生聲音的信號裝置，有機械型、機電型及壓電型。蜂鳴器的典型應用包括警笛，報警裝置，火災警報器，防空警報器，防盜器，定時器。
- 接上電阻可控制蜂鳴器聲音大小。



按鍵音樂(MY_PUSHTONE)

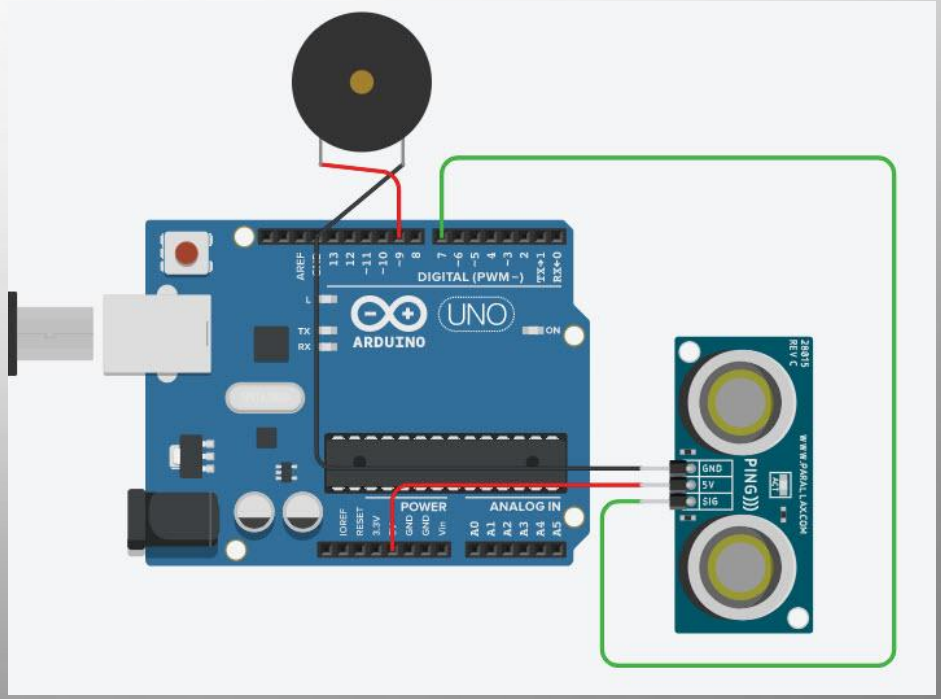
利用三個pushbutton分別發出不同的tone音
如 58 59 60 三個Tone

Quantity	Component
1	Arduino Uno R3
3	Pushbutton
3	10 kohm Resistor
1	100 ohm Resistor
1	Buzzer [Piezo small]



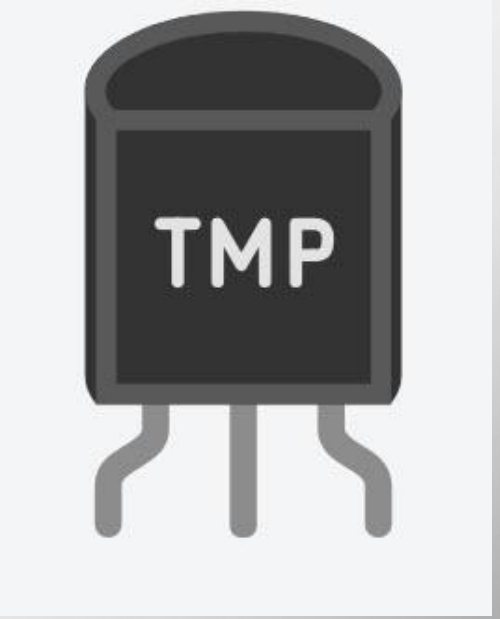
作業練習

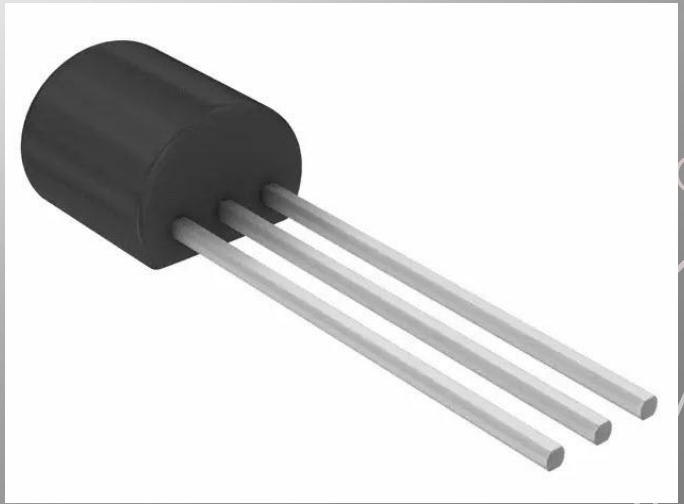
- 利用HC-SR04作一個警報器當物體接近30公分內發出聲音



温度感測器

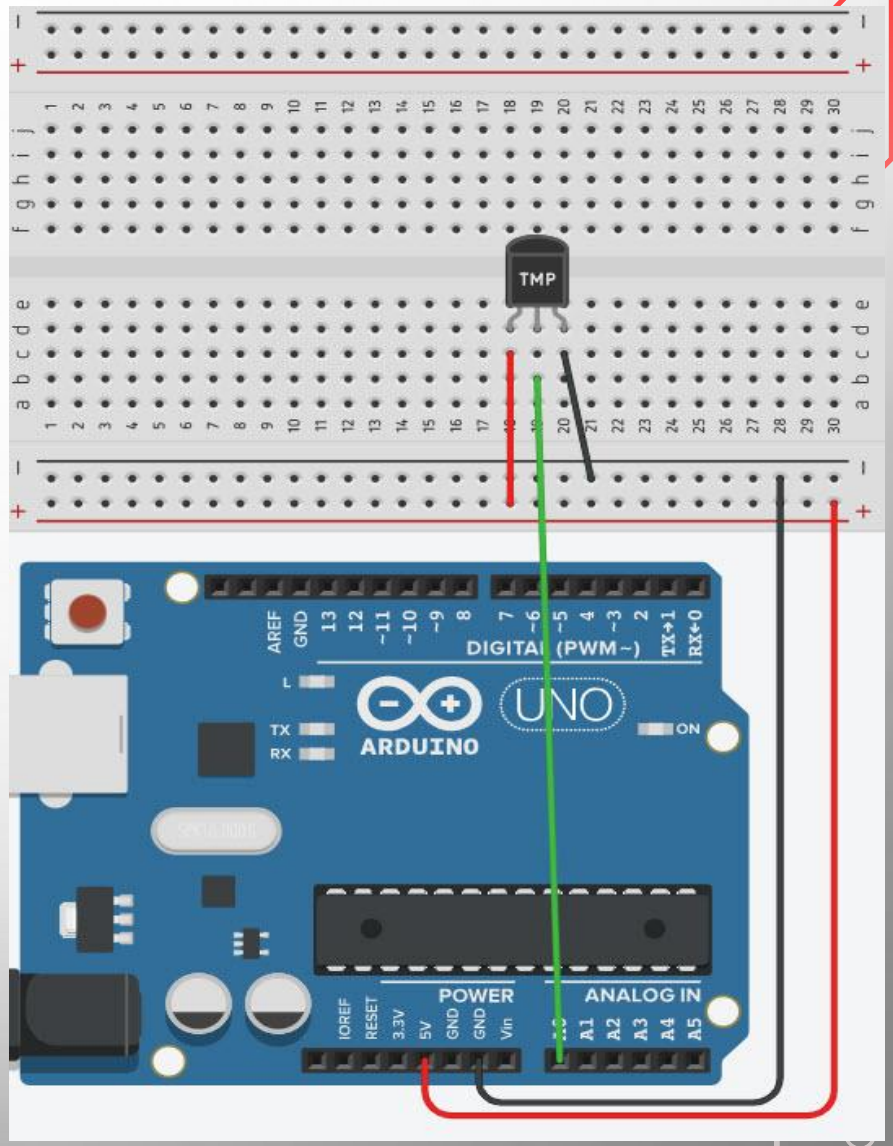
- 温度感測器
  低電壓操作：2.7 V 至 5.5 V
- 以 °C 單位直接校準
- 10 mV/°C 標度因數（TMP37 為 20 mV/°C）
- 在溫度範圍內準確度為 ±2°C（典型）
- 線性度為 ±0.5°C（典型）
- 較大電容負載下保持穩定
- 指定溫度介於 -40°C 至 125°C，工作溫度高達 150°C
- 靜態電流小於 50 µA
- 關斷電流為 0.5 µA（最大）
- 低自體發熱

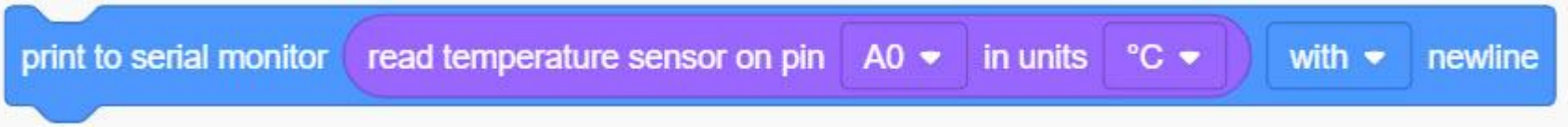




TMP36 温度感測器

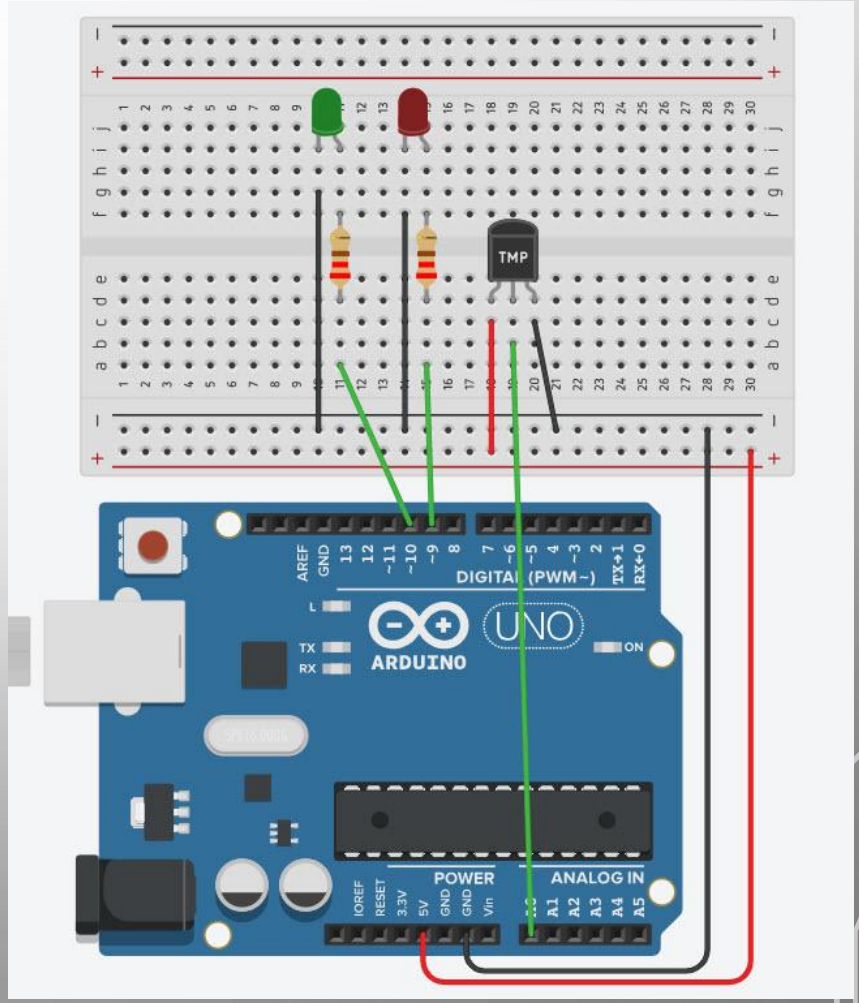
- Arduino UNO
- 麵包板
- TMP36 温度感測器





TMP36 温度感測器温度 高於40度亮紅燈小於20度亮綠燈

- Arduino UNO
- 麵包板
- TMP36 温度感測器
- LED 紅 綠各1個
- 220Ω電阻*2

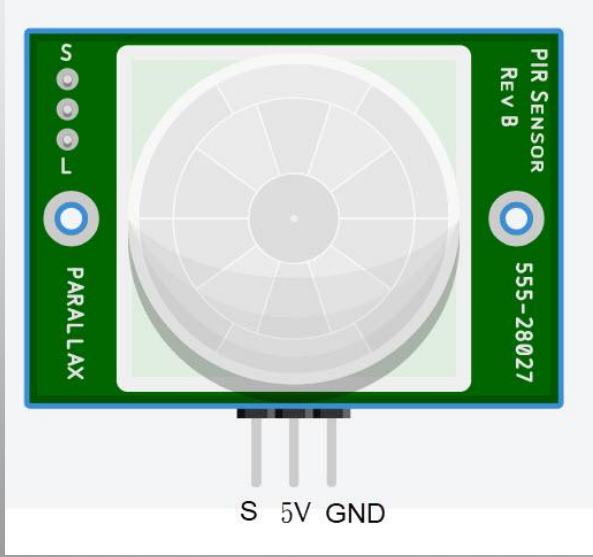


PIR紅外線人體感測器

- 電源需求: 3 to 6 V
  DC: 12 mA @ 3 V
  23 mA @ 5 V
- 適合用在走廊及路口轉角的亮等警示, 警報, 或機器人移動檢測。

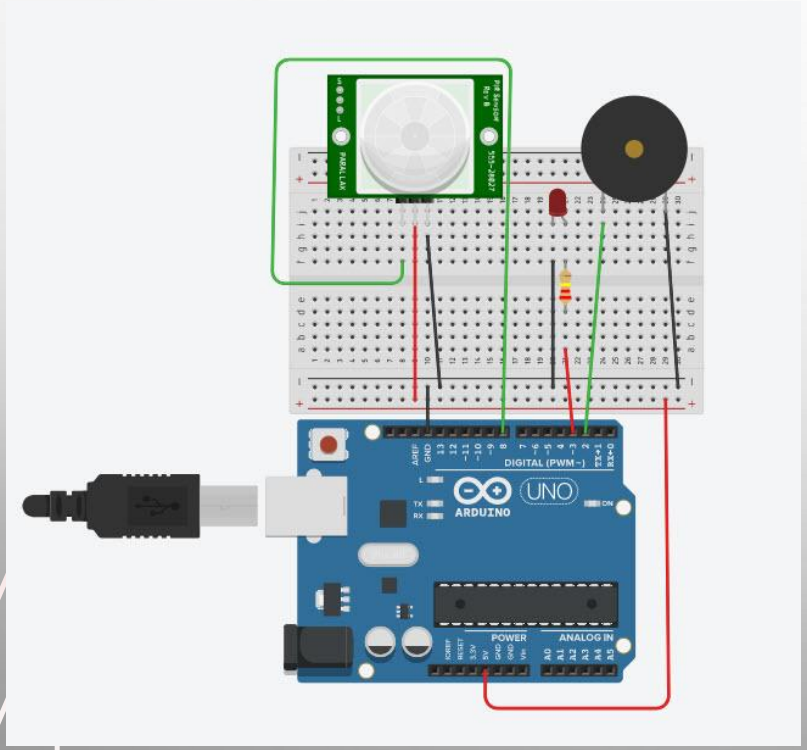
人體感測器(PIR)
利用數位接腳讀取PIR訊號

- 人體接近時LED亮 離開時暗
- 人體接近時BUZZER出聲音人體遠離時BUZZER無聲



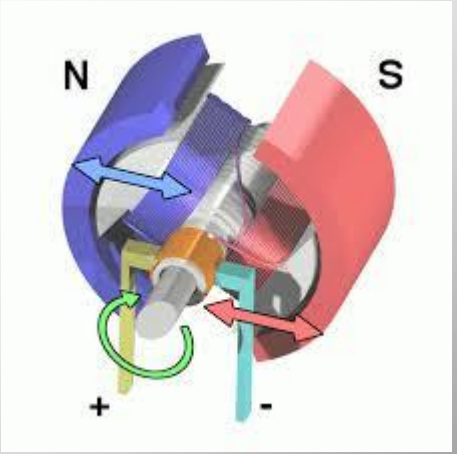
Quantity	Component
1	PIR SENSOR
1	Arduino Uno R3
1	220 kohm Resistor
1	Red LED
1	Piezo

PIR 測試(PIR SENSOR)



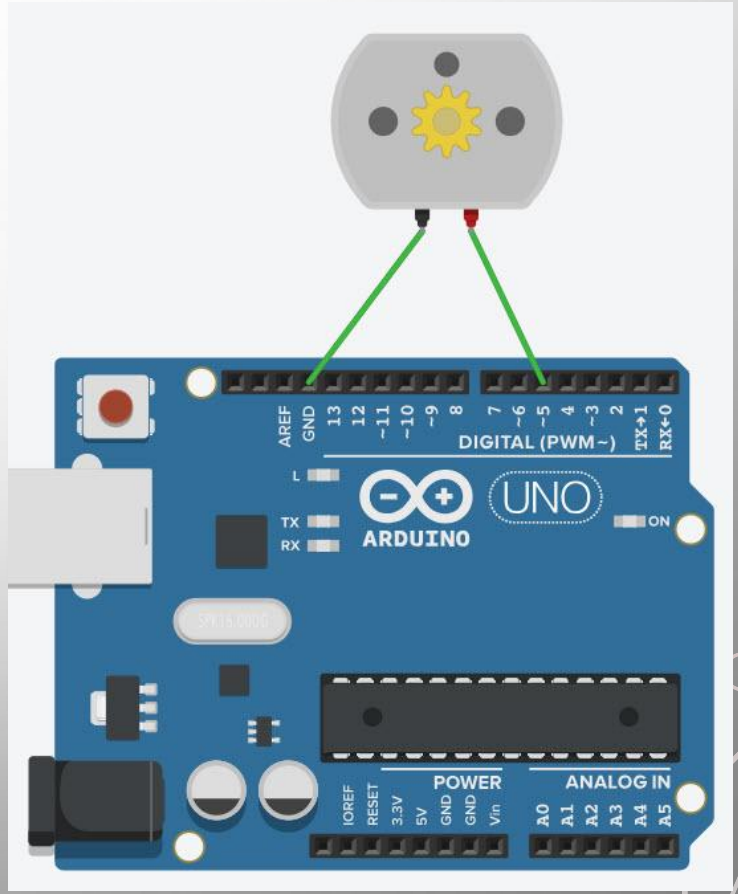
DC MOTOR 直流馬達

- 直流馬達是依靠直流電驅動的馬達，在小型電器上應用較為廣泛。



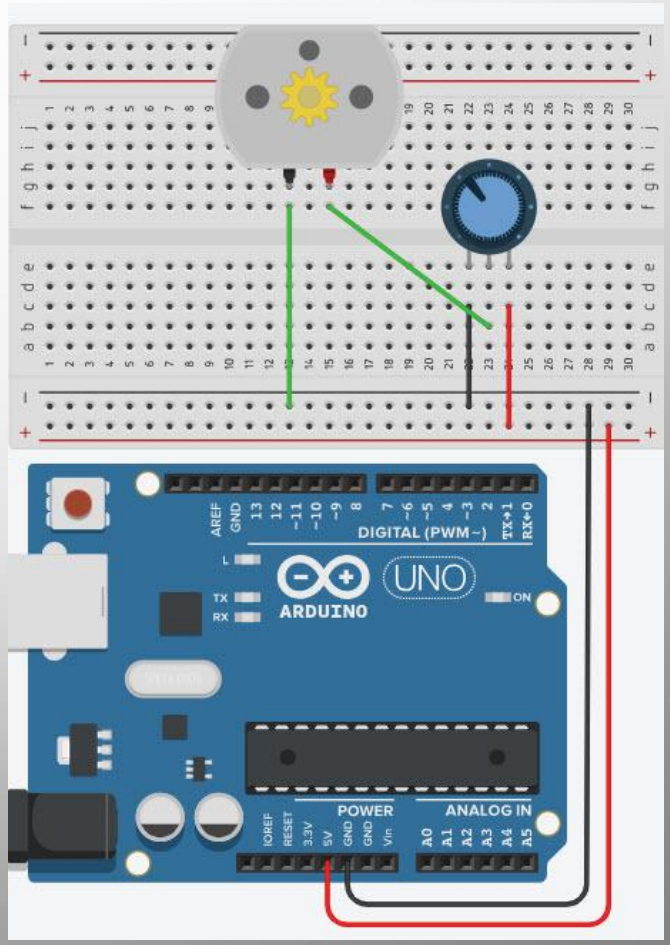
利用PWM控制直流馬達轉速

- Arduino UNO
- 麵包板
- DC MOTOR



利用可變電阻電流量控制直流馬達

- Arduino UNO
- 麵包板
- DC MOTOR
- 可變阻

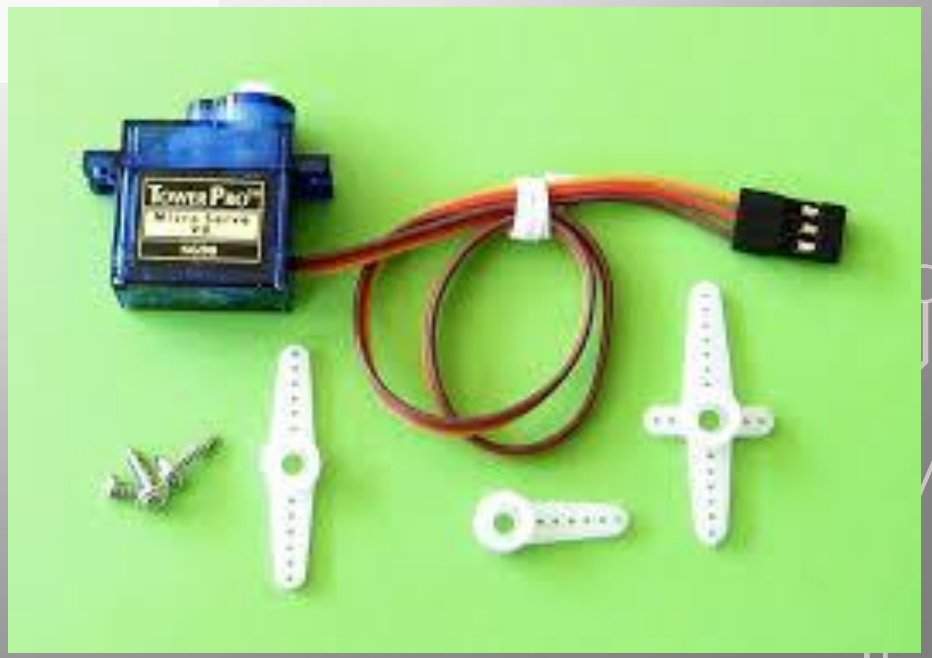


伺服馬達(SERVO)

- 伺服馬達（Servomotor）是對用於使用伺服機構的馬達（電動機）總稱。伺服（Servo）一詞來自拉丁文"Servus"，本為奴隸（Slave）之意，此指依照命令動作的意義。所謂伺服系統，就是依照指示命令動作所構成的控制裝置，應用於馬達的伺服控制，將感測器裝在馬達與控制對象機器上，偵測結果會返回伺服放大器與指令值做比較。由此可知，因為伺服馬達是以回饋訊號控制，與藉由輸入脈波訊號控制的步進馬達有所區別。

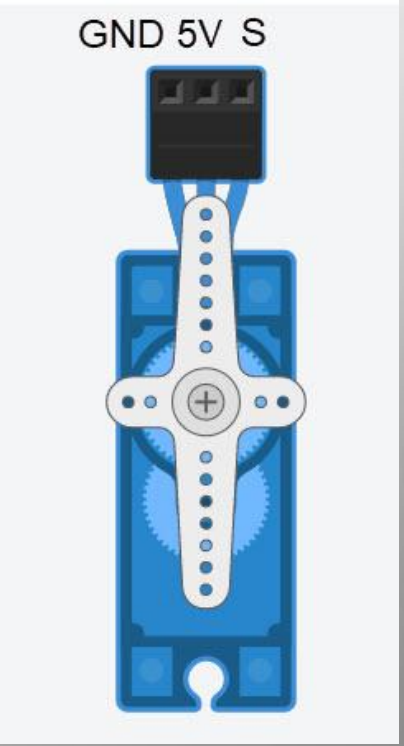
伺服馬達運作

- 利用類別庫送出PWM訊號，控制馬達角度。



SERVO 伺服馬達(SERVOTEST)

1. 伺服馬達指向指定度數(60 90 120 180)
2. 伺服馬達從0度走到180 再從180回到0度
3. 控制三個伺服馬達同時從0到180再從180回到0

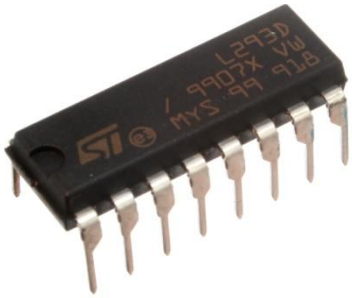


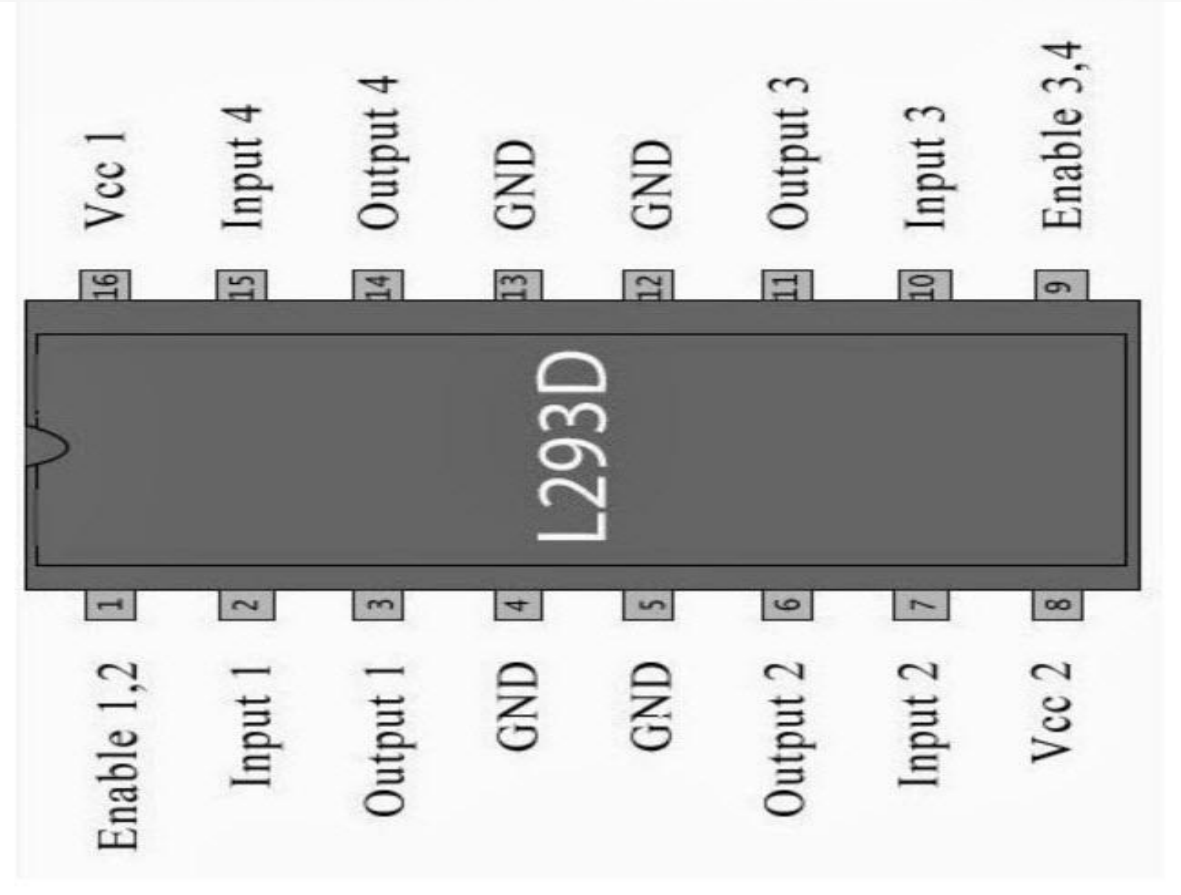
課堂線習

- 利用三個可變電阻修改三個不同的伺服馬達方向

- 每一個馬達記錄下三個角度，同一時間三個馬達同時作動作，並停二秒重覆作3次停止，等按下啓動開關再重來

L293D 馬達趨動IC

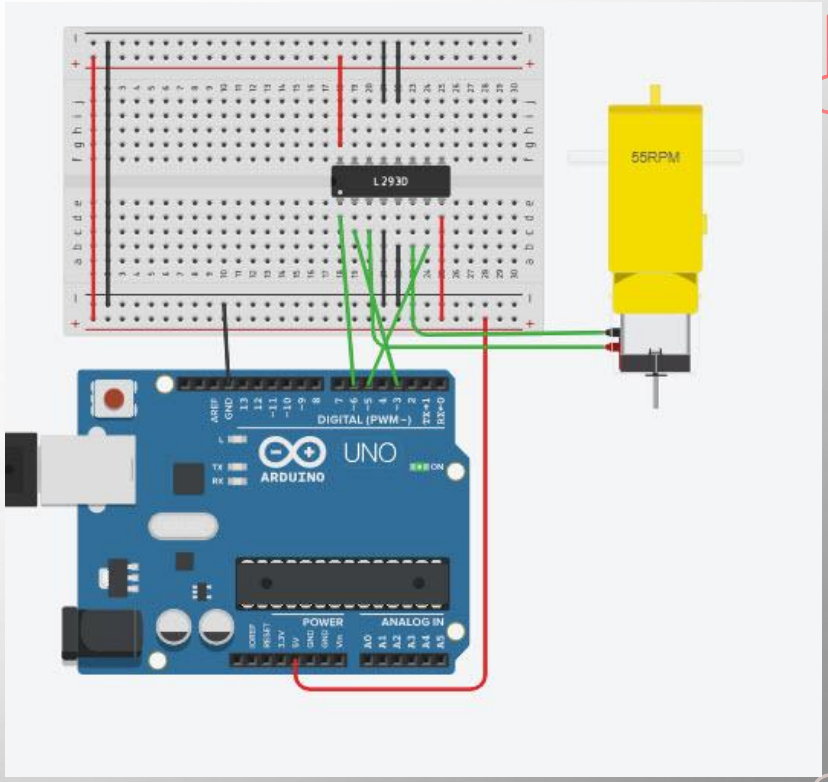




- Enable1,2　Enable3,4 PWM訊號控制馬達速度，0~255
- INPUT 數位PWM接腳
- OUTPUT 馬達接腳
- VCC1、VCC2接5V

單馬達運作

- Arduino UNO
- 麵包板
- L293D 馬達趨動IC
- 步進馬達

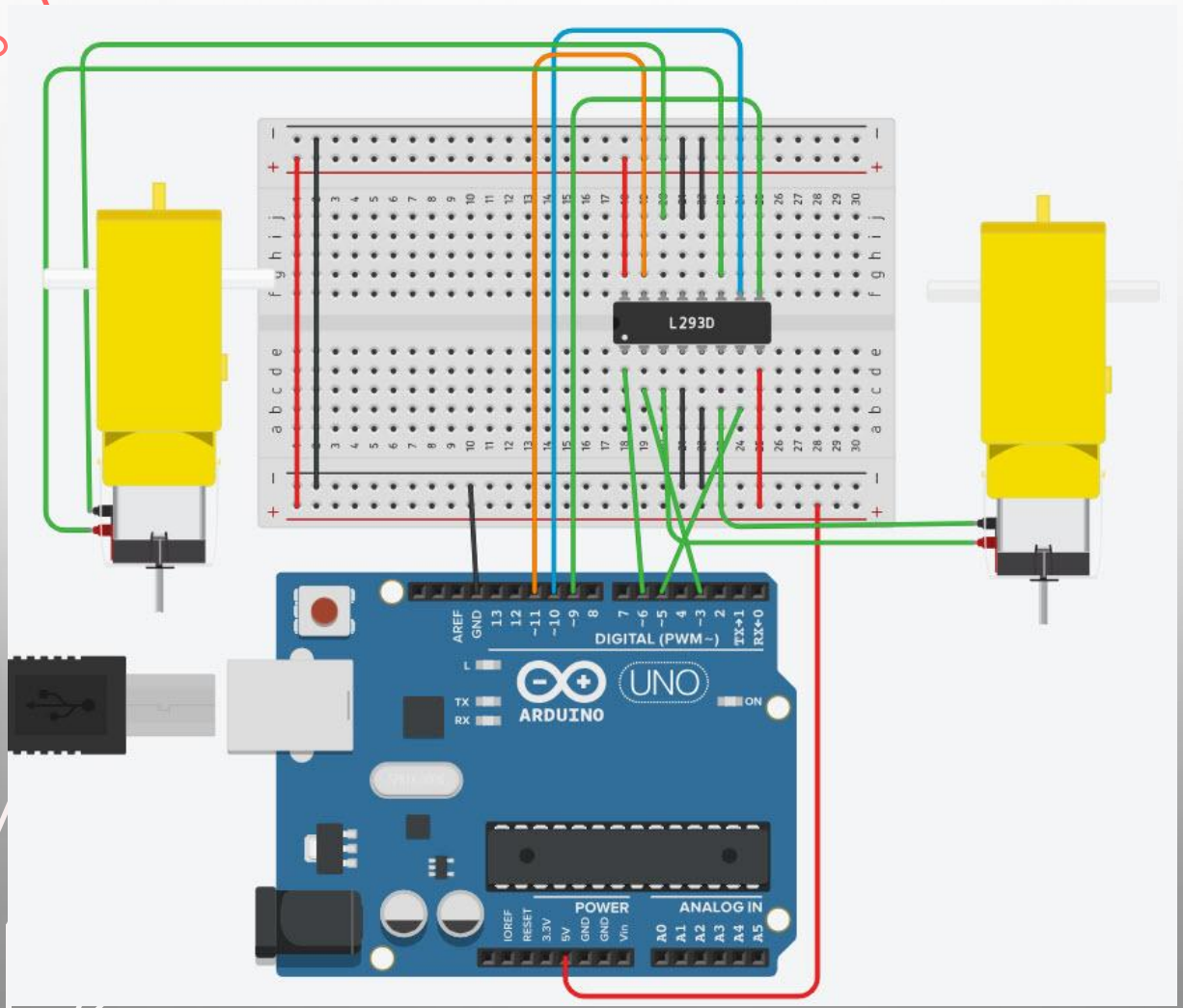


雙馬達運作

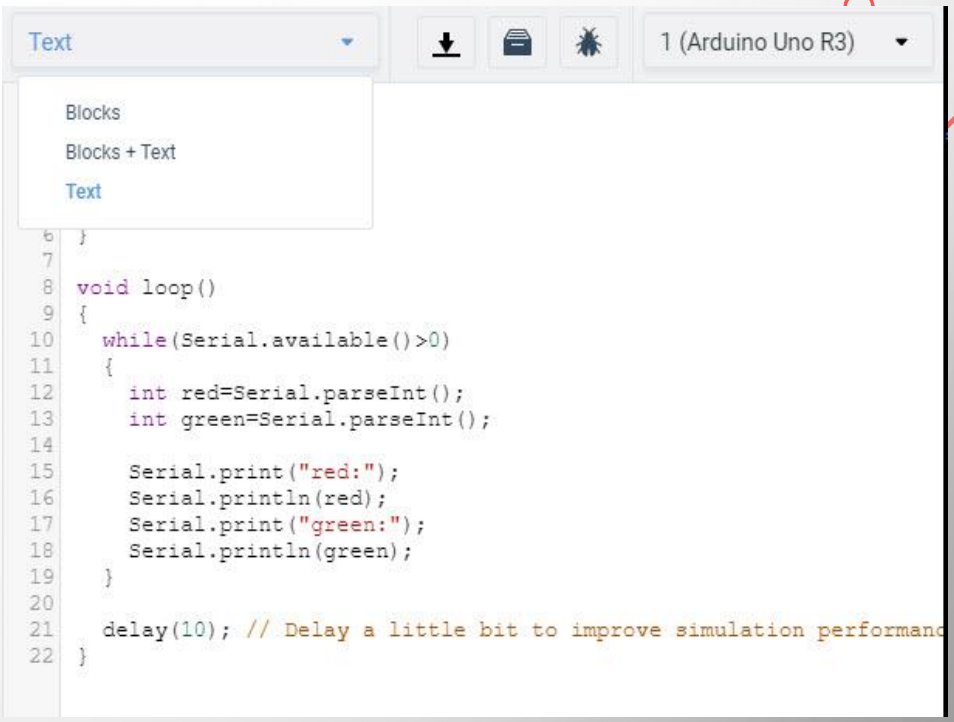
- Arduino UNO
- 麵包板
- L293D 馬達趨動IC
- 步進馬達*2

L293D	ARDUINO
Enable1	D6
INPUT1	D3
OUTPUT1	MOTOR1 +
GND	GND
GND	GND
OUTPUT2	MOTOR1 -
INPUT2	D5
VS	5V
Enable2	D9
INPUT3	D10
OUTPUT3	MOTOR3 +
GND	GND
GND	GND
OUTPUT4	MOTOR4 -
INPUT4	D11
VS	5V

接線圖及程式區塊


L293D
ARDUINO
UNO
DIGITAL (PWM~)
POWER
ANALOG IN


set pin 3 to HIGH
set pin 5 to LOW
set pin 6 to 100
set pin 10 to HIGH
set pin 11 to LOW
set pin 9 to 100
wait 0.1 secs



ARDUINO 文字化程式碼

ARDUINO主程式碼架構

- setup()
  {
    設定及初始化
  }

loop()
{
  重覆執行程式碼片段
}

學習內容

- SETUP()　　//設定開始初始宣告
- LOOP()　　//執行命令
- Serial.begin(9600)　//設訂波特率為9600
- Serial.print(“HELLO WORLD!!”)　//輸出序列文字內容
- Delay(10000)　　//等待10000毫秒

程式碼內容

```
void setup() {
  // put your setup code here, to run once:
Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
Serial.print("Hello World");
delay(10000);
}
```

PINMODE()

- pinMode(pin腳,模式)

  模式= INPUT　OUPUT

  pinMode(pin, INPUT_PULLUP);　//輸入高電壓

  pinMode(pin, INPUT_PULLUP);　//輸入低電壓

  EX:pinMode(13,OUTPUT)

  指定第13PIN腳為輸出腳位(可點亮LED)

  pinMpde(10,INPUT)

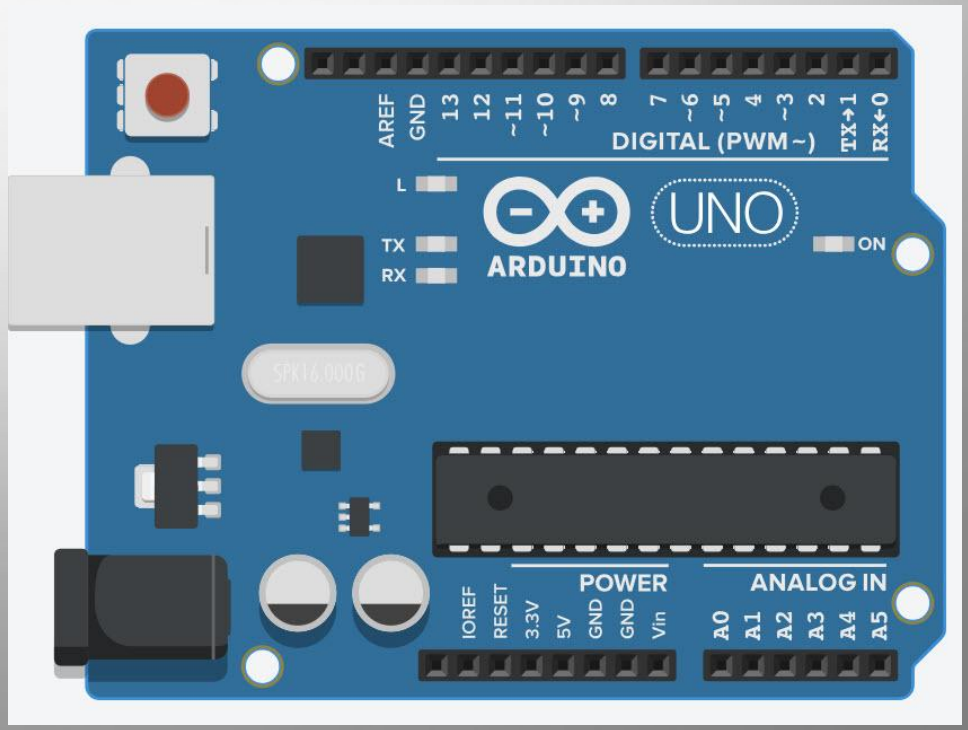
  指定第10PIN腳為接收腳位(可接收元件傳來的訊號)

ARDUINO 內鍵關鍵字

- HIGH　1　輸出高電壓
- LOW　0　輸出低電壓
- LED_BUILTIN　13　常數

LED BLINK

- 利用Arduino主板的LED燈作一亮一暗
- 數位第13支腳HIGH　LOW
- 亮一秒暗一秒



SETUP()

```
void setup()
{
  pinMode(13, OUTPUT);
}
```

LOOP()

```
void loop()
{
  digitalWrite(13, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(13, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

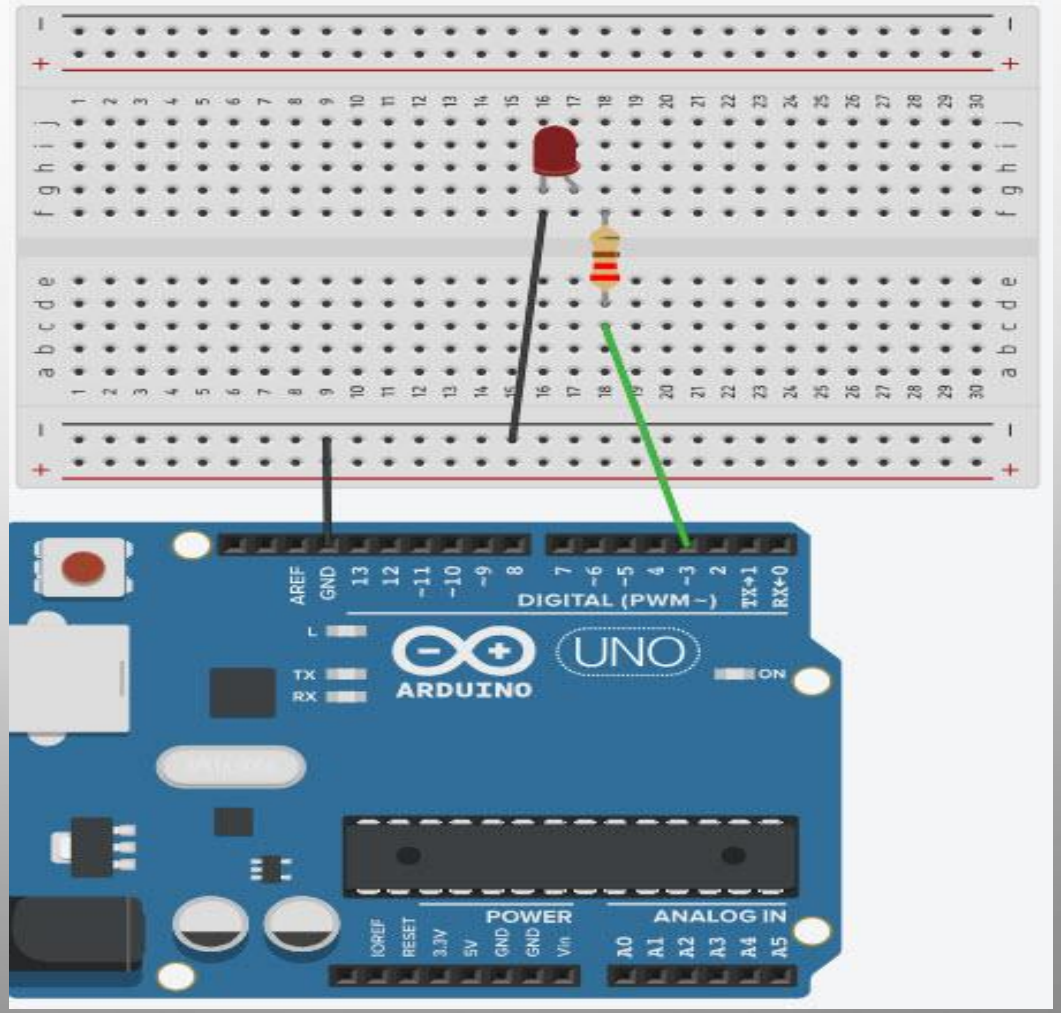
ARDUINO 變數型態

數值、字元、字串、真假值、陣列

ARDUINO 變數型態數值

- Int 整型變數　int led1=13;
- double 雙浮點 double pi=3.1416159;
- float　浮點　float sen=1.117;
- long　長整型　介於 -2,147,483,648 to 2,147,483,647
- unsigned long 無符號的長整型
  0 to 4,294,967,295

INT 型態練習

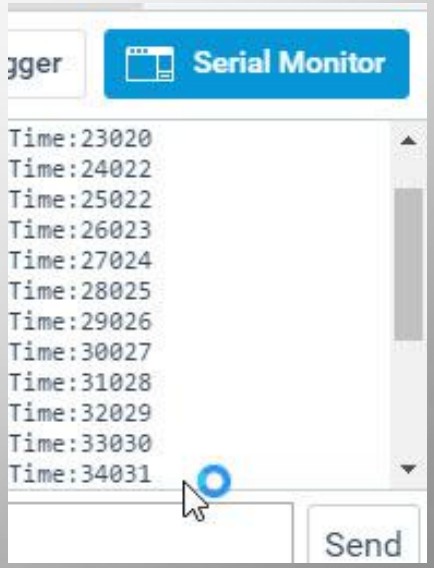


INT 型態

```
int led1=3;//宣告LED1接在數位接腳3
void setup()
{
  pinMode(led1, OUTPUT);
}
void loop()
{
  digitalWrite(led1, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(led1, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

UNSIGNED LONG
MILLIS() 程式開始執行以來的時間毫秒

```
unsigned long  time=0;

void setup()

{

   Serial.begin(9600);

}

void loop()

{

   Serial.print("Time: ");

  time = millis();

Serial.println(time);

delay(1000);

}
```



陣列宣告

- int ar[3]={2,3,4};
- char ch[8]="arduino";
- char ch[7]={'a','r','d','u','i','n','o'};

陣列範例

```
char ch[8]="arduino";
void setup()
{
   Serial.begin(9600);
   for(int i=0;i<7;i++)
   {
      Serial.println(ch[i]);
   }
}
void loop()
{
}
```

字元與字串

- char ch='a';
- char* myStrings[]={"This is string 1", "This is string 2", "This is string 3", "This is string 4", "This is string 5","This is string 6"};

STRING()

- String stringOne = "Hello String";
- String stringOne = String(stringTwo + " with more");

真假值BOOL

- bool running = false;
- bool st = true ;

計算、比較、註解

比較

- == 等於
- != 不等於
- > 大於
- >= 大於等於
- < 小於
- <= 小於等於

程式碼註解

- /*……*/　　區段註解
- //　　單行註解

邏輯比較

- ! 否(反之) not
- && 且(同時) and
- || 或(或者) or

流程控制

IF WHILE SWITCH FOR

流程控制IF 條件語句

- 簡單條件語句

```
if(true)
  {
    陳述句內容;
  }
```

```
if(millis()<5000)
 {
   Serial.println(millis());
 }
```

流程控制IF 條件語句

- 多條件語句

```
if(true)
  {
    陳述句內容;
  }
  else if(true)
  {
   陳述句內容;
  }
  else
  {
   陳述句內容;
  }
```

```
if(millis()<5000)
  {
    Serial.println(millis());
  }
  else
  {
    Serial.println("Go~~");
  }
```

WHILE

```
while(true)
{
   陳述句內容;
}
```

```
Serial.begin(9600);
 while(millis()<5000)
{
  Serial.println(millis());
}
```

SWITCH CASE

```
switch (var)
    {
   case label1:
   陳述語21
   break;
   case label2:
     陳述語2
    break;
   default:
   }
```

FOR 循環語句

- 遞增

```
for(int i=0;i<N;i++)
{
  循環N次;
}
```

```
for(int i=0;i<255;i++)
{
  analogWrite(led1,i);
  delay(300);
}

//利用FOR使LED燈慢慢變亮
```

FOR 循環語句

- 遞減

```
for(int i=N;i>=0;i--)
{
  循環N次;
}
```

```
 for(int i=255;i>=0;i--)
{
   analogWrite(led1,i);
   delay(300);
}

//利用FOR使LED燈慢慢變暗
```

數位接腳應用

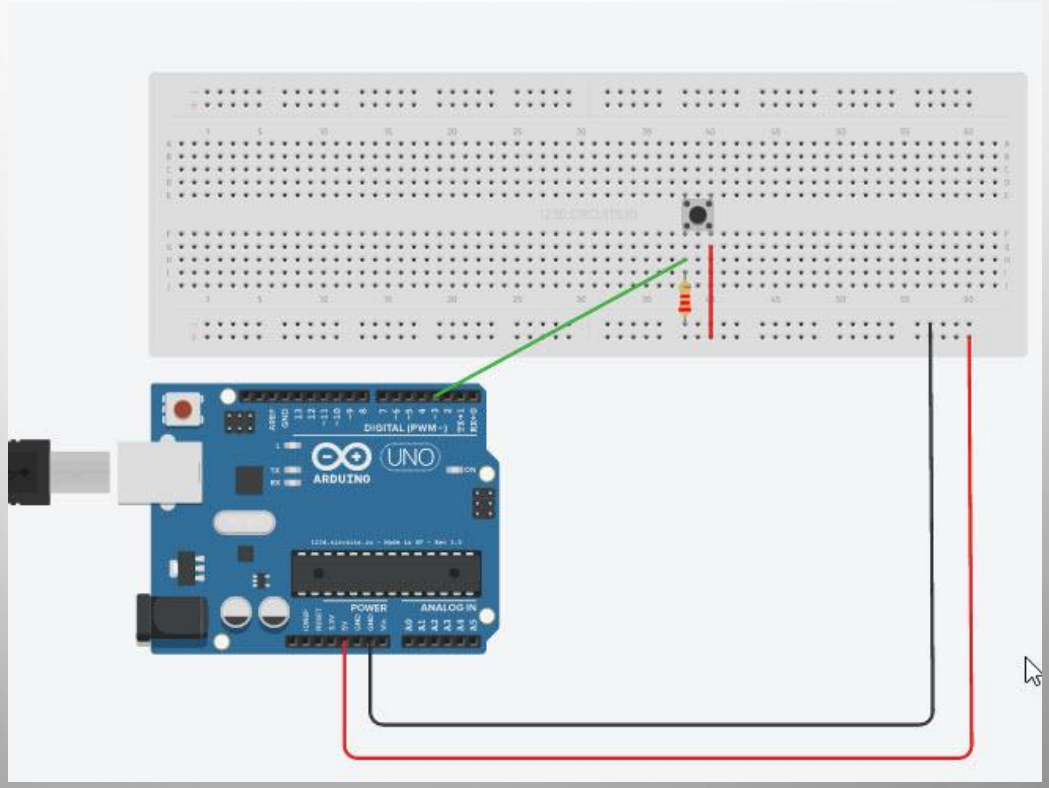
DIGITALREAD **SERIAL**

數位接腳接收數位訊號(HIGH　LOW)

材料

- ARDUINO UNO
- 按鍵開關一個
- 10KΩ電阻
- 面包板

PUSHBUTTON



程式碼

```
Int pin1=3;
int value=0;
void setup()
{
  pinMode(pin1,INPUT);
  Serial.begin(9600);
}
```

```
void loop()
{
  value=digitalRead(pin1);
  if(value==HIGH)
  {
   Serial.println("PUSH BUTTON");
    delay(1000);
  }
  else
  {
    Serial.println("BUTTON OPEN");
    delay(1000);
  }
}
```

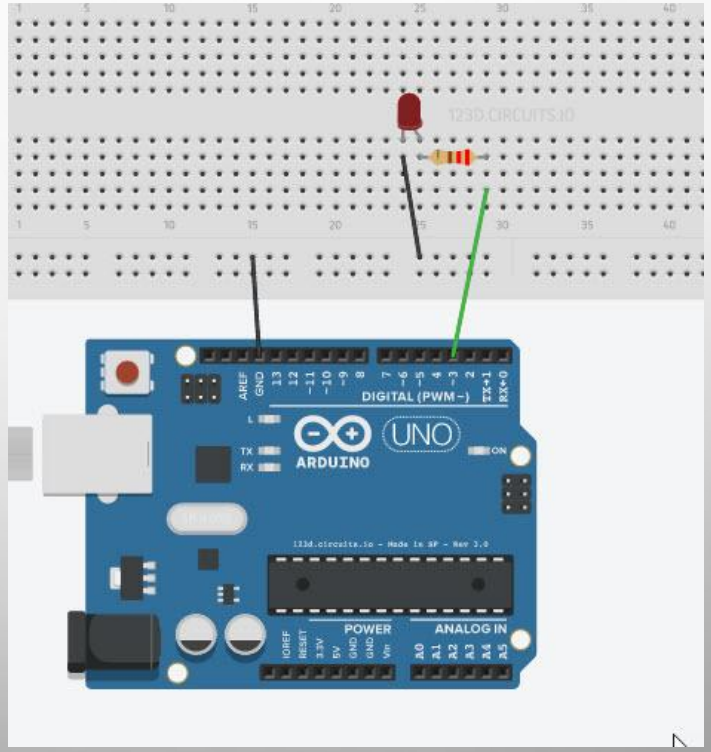
DIGITALWRITE **SERIAL**

數位接腳發送數位訊號(HIGH　LOW)

材料

- ARDUINO UNO
- LED 燈炮
- 220Ω電阻
- 面包板

DIGITALWRITE



程式碼

```
int led=3;
void setup(){
  pinMode(led,OUTPUT);
}
```

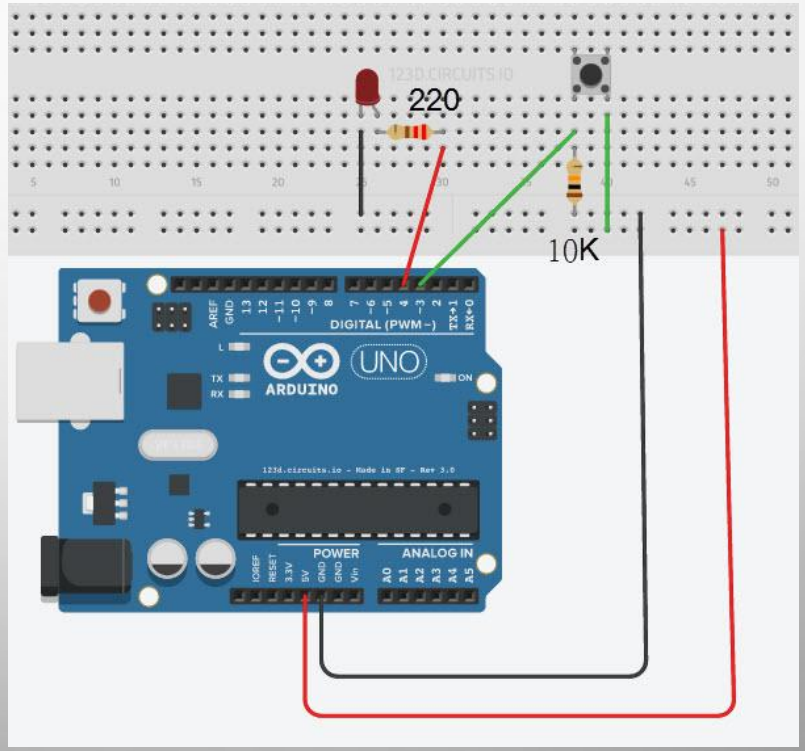
```
void loop()
{
digitalWrite(led,HIGH);
  delay(1000);
  digitalWrite(led,LOW);
  delay(1000);
}
```

DIGITAL READ WRITE 整合應用

讀取按鍵開關控制LED燈
材料

- ARDUINO UNO
- LED 燈炮
- 220Ω電阻
- 10Ω電阻
- 按鍵開關
- 面包板

DIGITAL READ WRITE



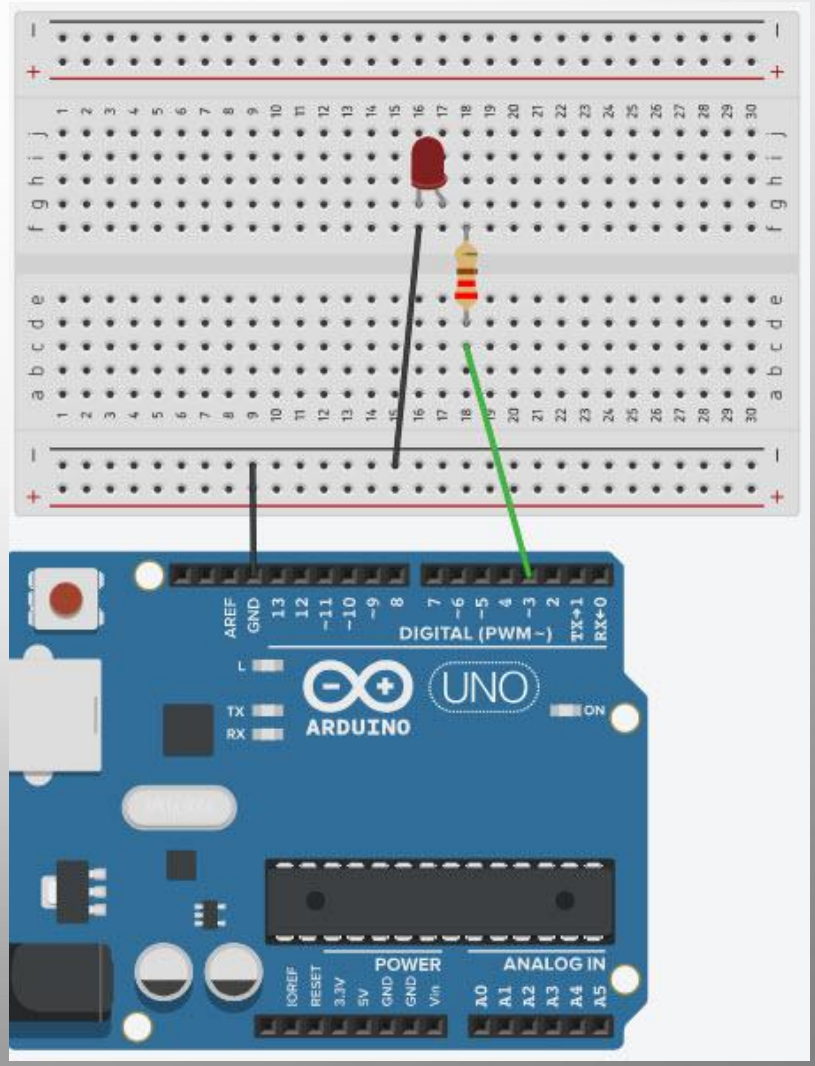
程式碼

```
int btpin=3;
int led=4;
int value=0;

void setup()
{
  pinMode(btpin,INPUT);
  pinMode(led,OUTPUT);
}
```

```
void loop()
{
  value=digitalRead(btpin);
  if(value==HIGH)
  {
    digitalWrite(led,HIGH);
  }
  else
  {
    digitalWrite(led,LOW);
  }
}
```

實現按鍵開關一開一關的功能
材料

- ARDUINO UNO
- LED 燈炮
- 220Ω電阻
- 10Ω電阻
- 按鍵開關
- 面包板



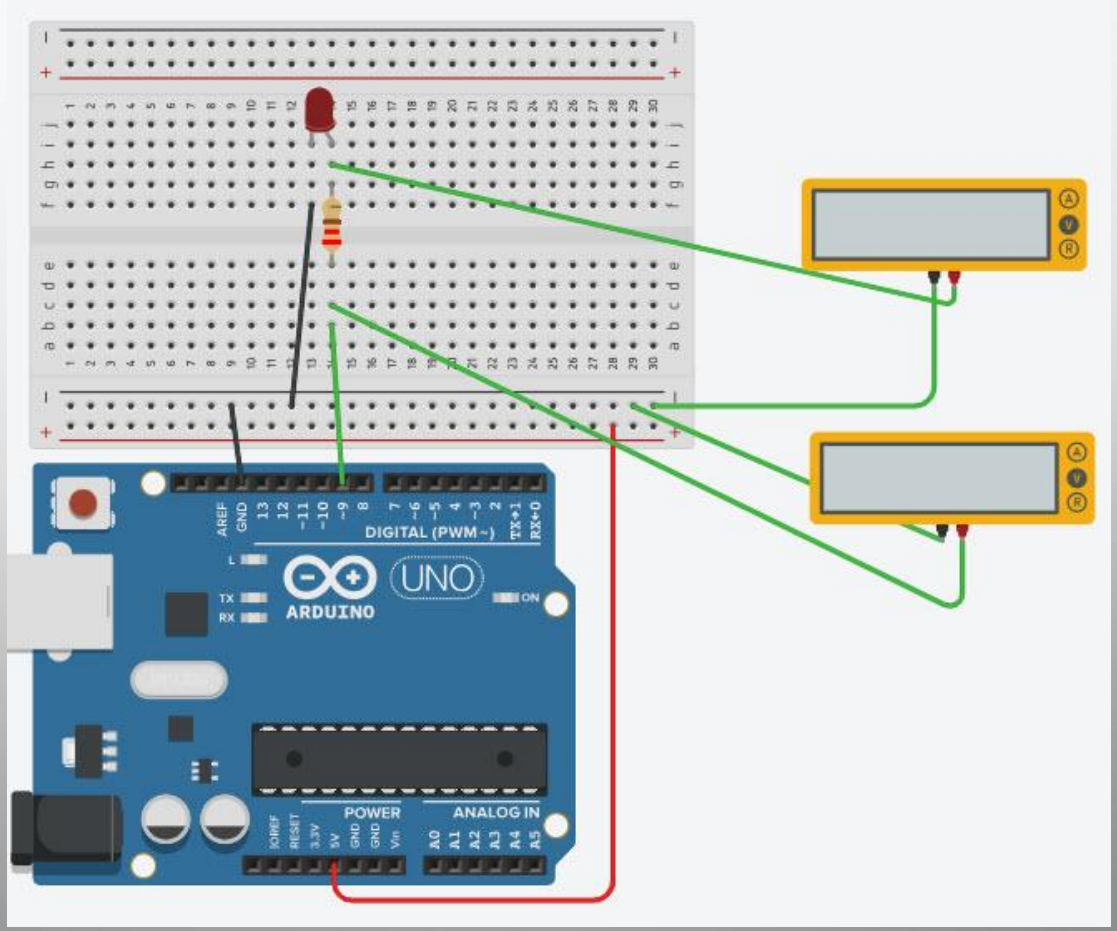
程式碼

```
int led1=7;
int pushbutton1=2;
void setup()
{
  pinMode(led1, OUTPUT);

    pinMode(pushbutton1,INP
    UT);
}
```

```
void loop()
{
  if(digitalRead(pushbutton1)==HIGH)
  {
   digitalWrite(led1,!digitalRead(led1));

  }
   delay(1000);
}
```

數位接腳的ANALOGWRITE

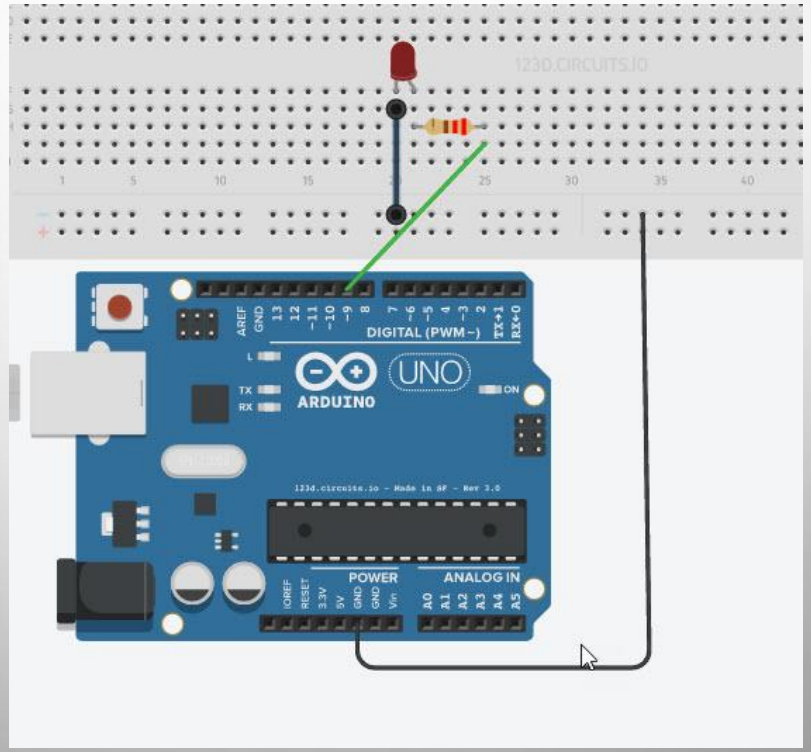
利用電表了解PWM運作原理



使用PWM接腳輸出模擬類比訊號
材料

- ARDUINO UNO
- LED 燈炮
- 220Ω電阻
- 面包板

DIGITPIN　ALANALOGWRITE (數位接腳PWM 輸出)



呼吸燈制作程式碼

```
int led = 9;
int brightness = 0;
int fadeAmount = 5;

void setup() {

  pinMode(led, OUTPUT);
}

void loop() {

  analogWrite(led, brightness);

  brightness = brightness + fadeAmount;

  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }
   delay(30);
}
```

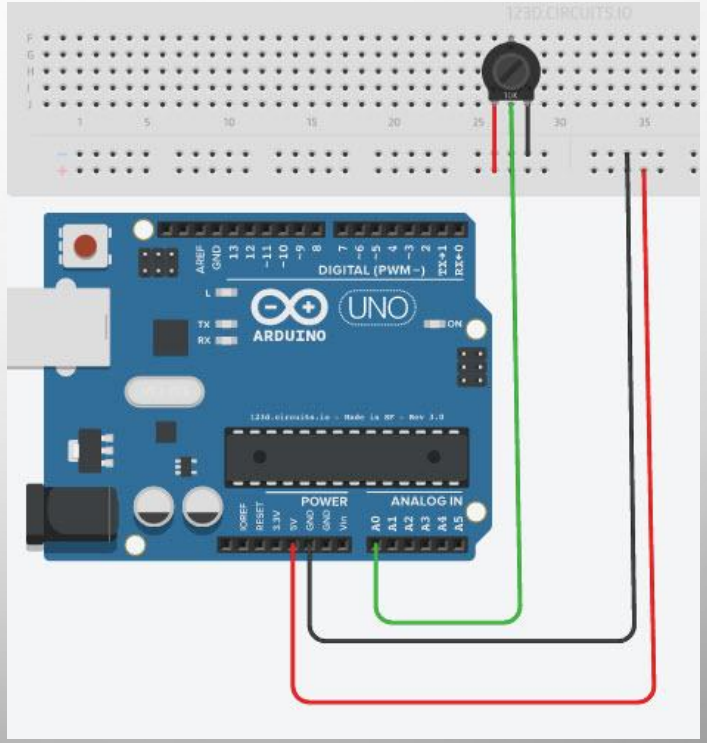
類比接腳應用

類比接收訊號

使用A0接腳讀取可變電阻
材料

- ARDUINO UNO
- 10K可變電阻
- 面包板

ANALOGREAD



程式碼

```
int sensorPin = A0;
int sensorValue = 0;
void setup() {
 Serial.begin(9600);
}
```

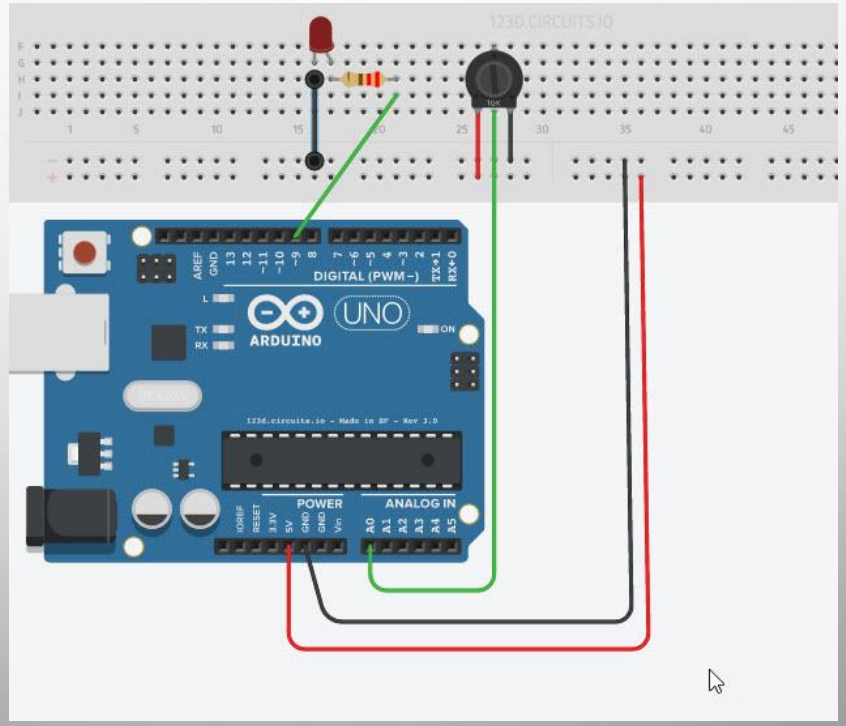
```
void loop()
{

sensorValue=analogRead(sensorPin);
  Serial.println(sensorValue);
  delay(500);
}
```

類比接腳接收 數位接腳發送訊號

利用可變電阻控制LED燈亮度
材料

- ARDUINO UNO
- LED 燈炮
- 220Ω電阻
- 10K可變電阻
- 面包板

ANALOGINOUT



MAP函數說明

map(value, fromLow, fromHigh, toLow, toHigh)
value : 當下的值

fromLow:來原最小值

fromHigh:來原最大值

toLOW:對應最小值

toHigh:對應最大值

LED燈範圍:0~255
可變電阻:0~1023

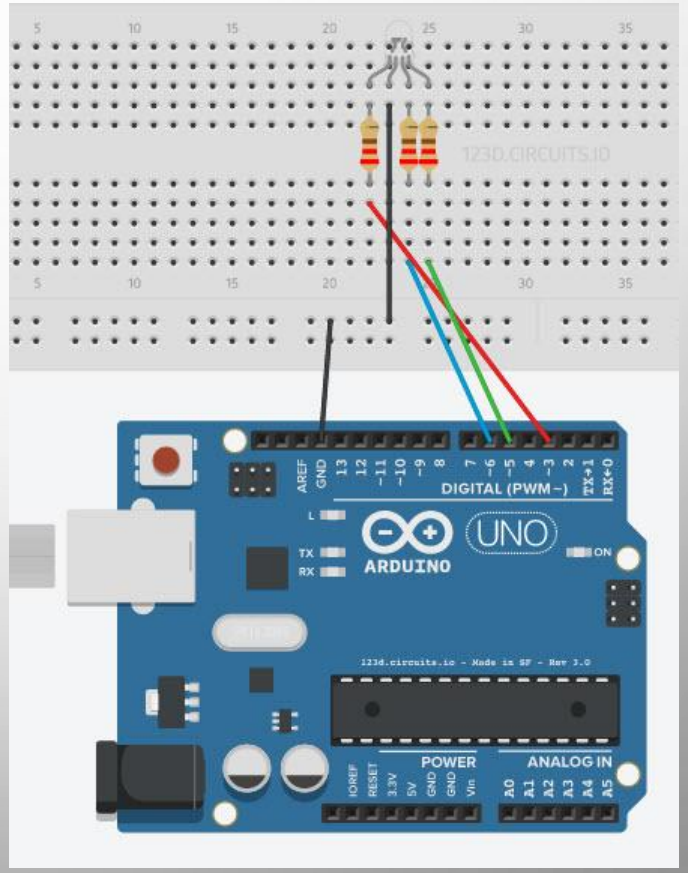
程式碼

```
int analogInPin = A0;
int analogOutPin = 9;
int sensorValue = 0;
int outputValue = 0;
void setup() {
  pinMode(analogOutPin,OUTPUT);
  Serial.begin(9600);
}
```

```
void loop() {
  sensorValue = analogRead(analogInPin);
  outputValue = map(sensorValue, 0, 1023, 0, 255);
  analogWrite(analogOutPin, outputValue);
  Serial.print("sensor = ");
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);
  delay(2);
}
```

全彩LED

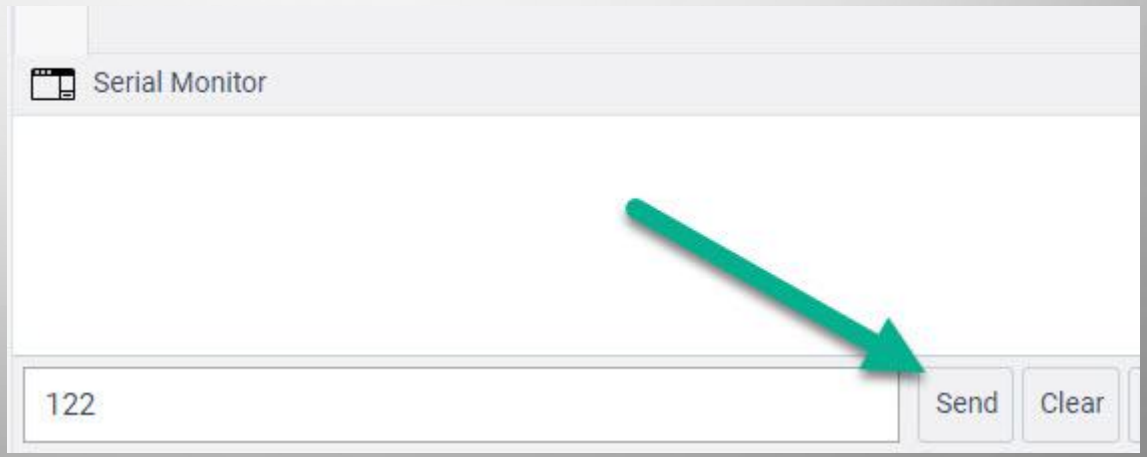
- ARDUINO UNO
- RGB LED
- 220Ω 電阻三個
- 面包板



SERIAL AVAILABLE() 串口傳回訊息

當串口發送訊息時會傳回一字節數，利用

Serial.available() 會反回字節數量>0



程式碼

```
const int redPin = 3;
const int greenPin = 5;
const int bluePin = 6;

void setup()
{
  Serial.begin(9600);
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}
```

```
void loop() {
  while (Serial.available() > 0) {
 int red = Serial.parseInt();
  int green = Serial.parseInt();
   int blue = Serial.parseInt();
        Serial.println(red);
        Serial.println(green);
        Serial.println(blue);
      red = 255 - constrain(red, 0, 255);
      green = 255 - constrain(green, 0, 255);
      blue = 255 - constrain(blue, 0, 255);
     analogWrite(redPin, red);
      analogWrite(greenPin, green);
      analogWrite(bluePin, blue);
     Serial.println(red, DEC);
      Serial.println(green, DEC);
      Serial.println(blue, DEC);
  }
}
```

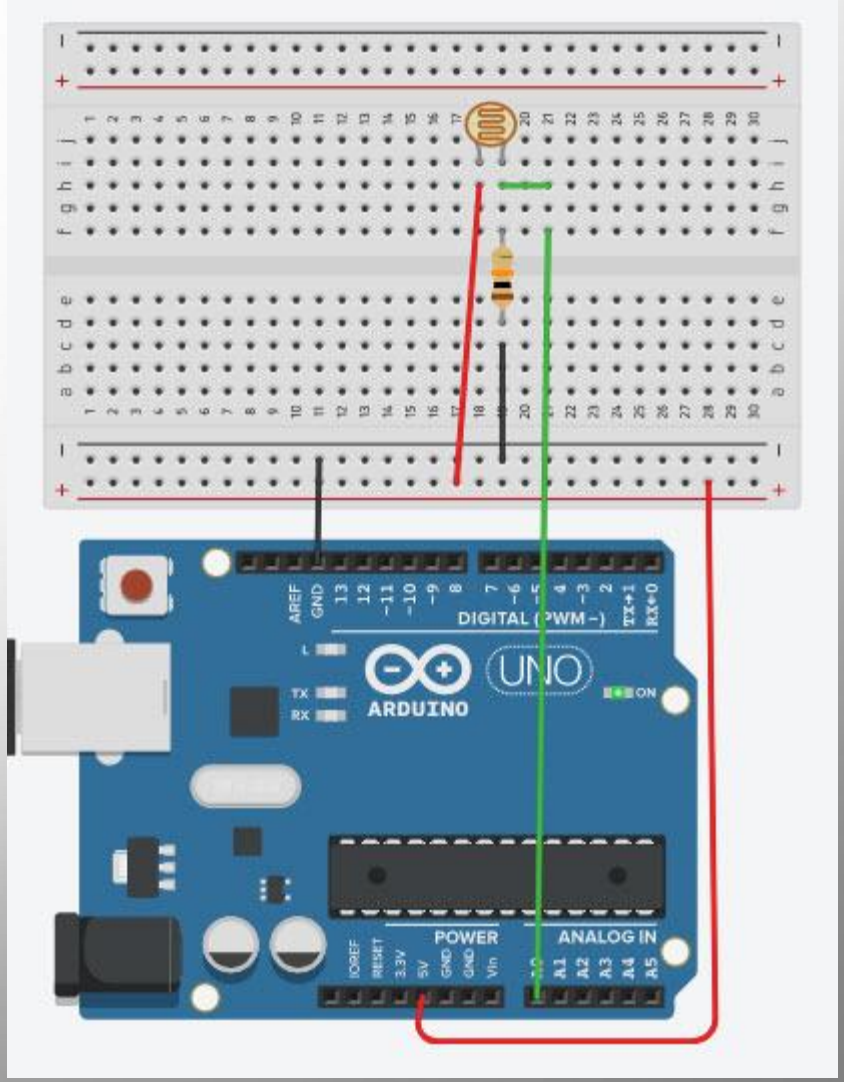
光敏電阻(LDR)

- 光敏電阻是利用光電導效應的一種特殊的電阻，簡稱光電阻，又名光導管。

- 它的電阻和光線的強弱有直接關係。光強度增加，則電阻減小；光強度減小，則電阻增大。

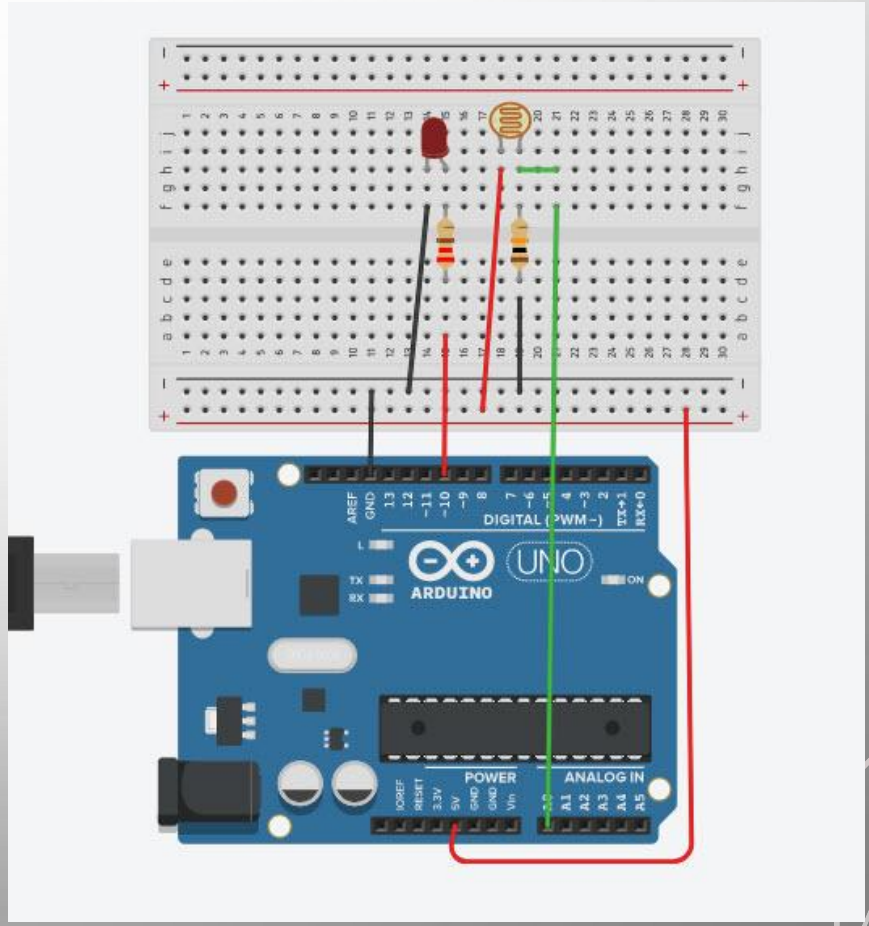
光敏電阻值

材料:
ARDUINO UNO
面包板
光敏電阻
10KΩ電阻



小夜燈制作

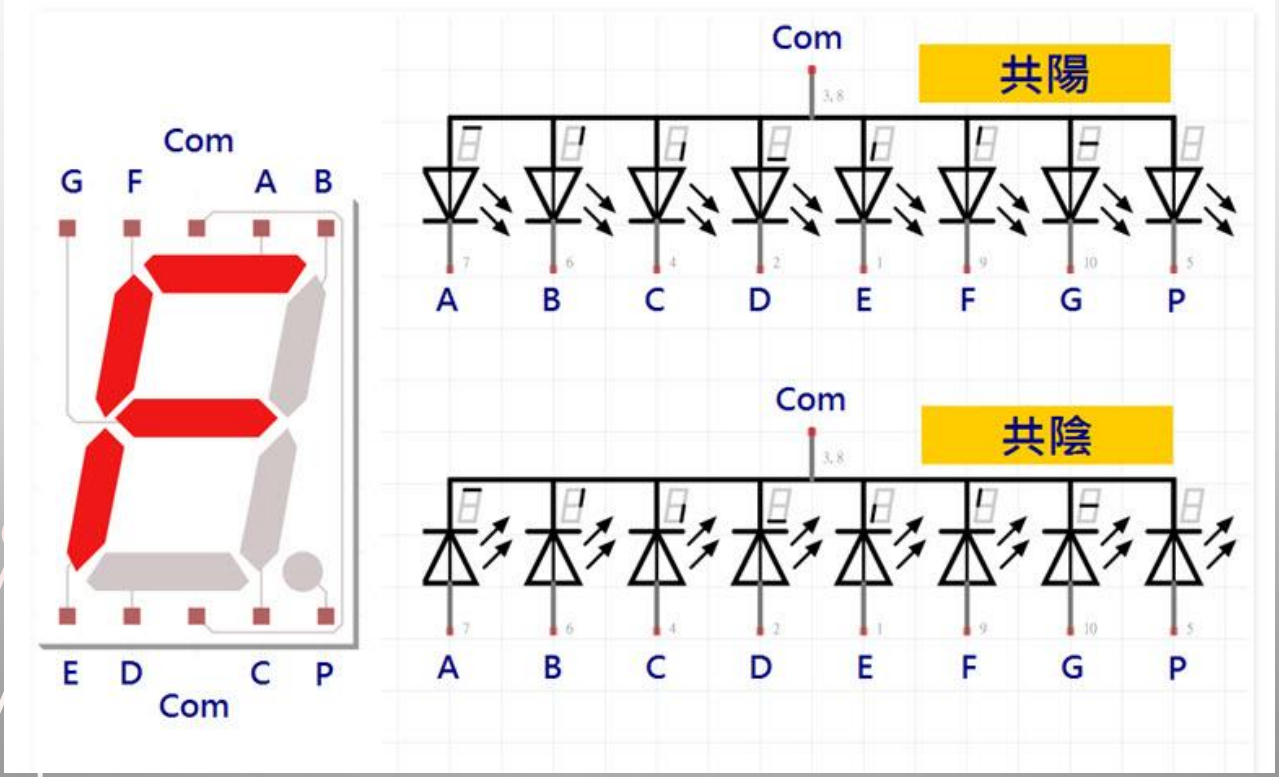
- ARDUINO UNO
- 面包板
- 光敏電阻
- 10KΩ電阻
- LED 燈泡
- 220Ω電阻

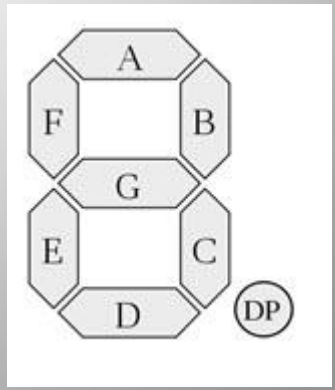


七段數字顯示器(共陽)

- 七段顯示器（英語：Seven-segment display）為常用顯示數字的電子元件。因為藉由七個發光二極體以不同組合來顯示數字，所以稱為「七劃管」、「七段數碼管」、「七段顯示器」，由於所有燈管全亮時所表示的是「8」，所以又稱「8字管」、「8字顯示器」。
- 多數七段顯示器還會在右下角附加一個表示小數點的燈管，因此也稱八段管。

七段數字腳位及排列

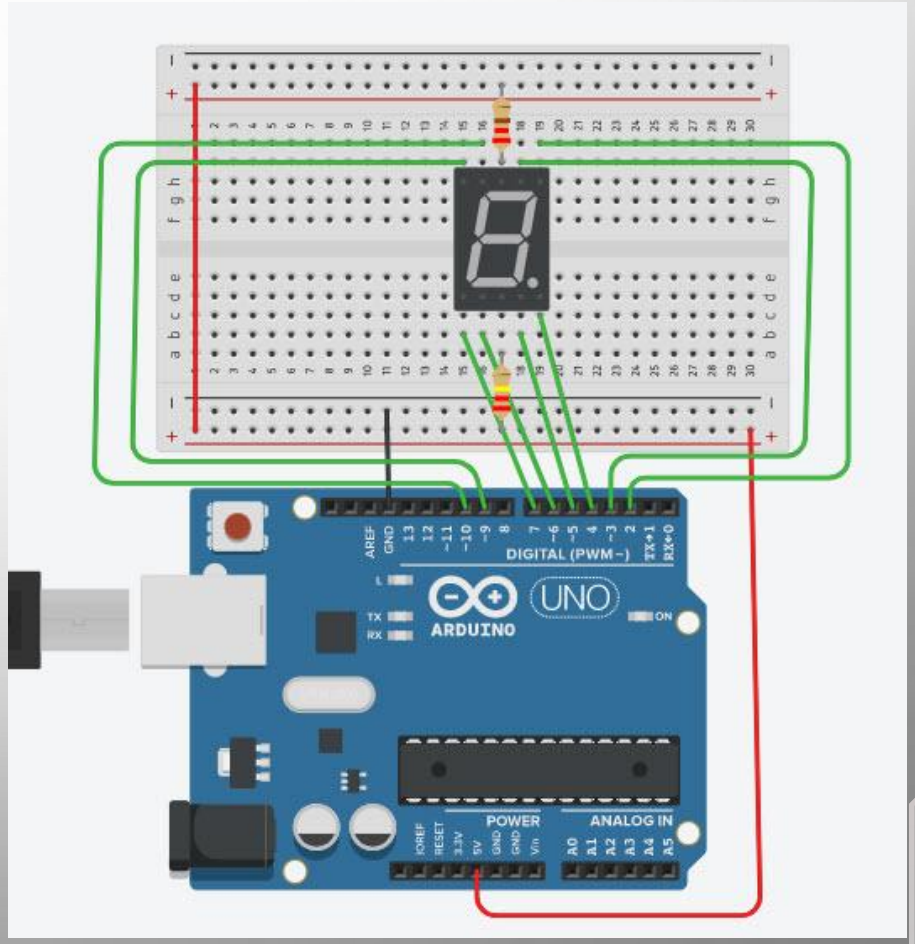




七段數字顯示器

材料

- ARDUINO UNO
- 面包板
- 220Ω電阻 2個
- 七段數字顯示器



數字代碼表

0 –0000001

1 –1001111

2 –0010010

3 --0000110

4 --1001100

5 --0100100

6 --0100000

7 --0001111

8 –0000000

9 --0000100

H --1001000
E --0110000
L --1110001
O--0000001

自訂義函數

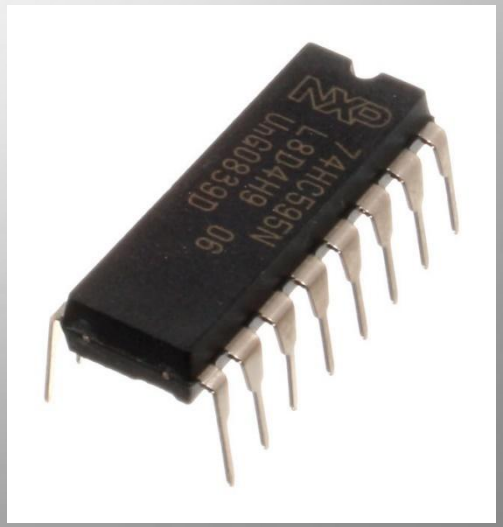
自訂義函數

```
void 函式名(參數)  //無反回值函數
 {
    陳述句;
 }
```

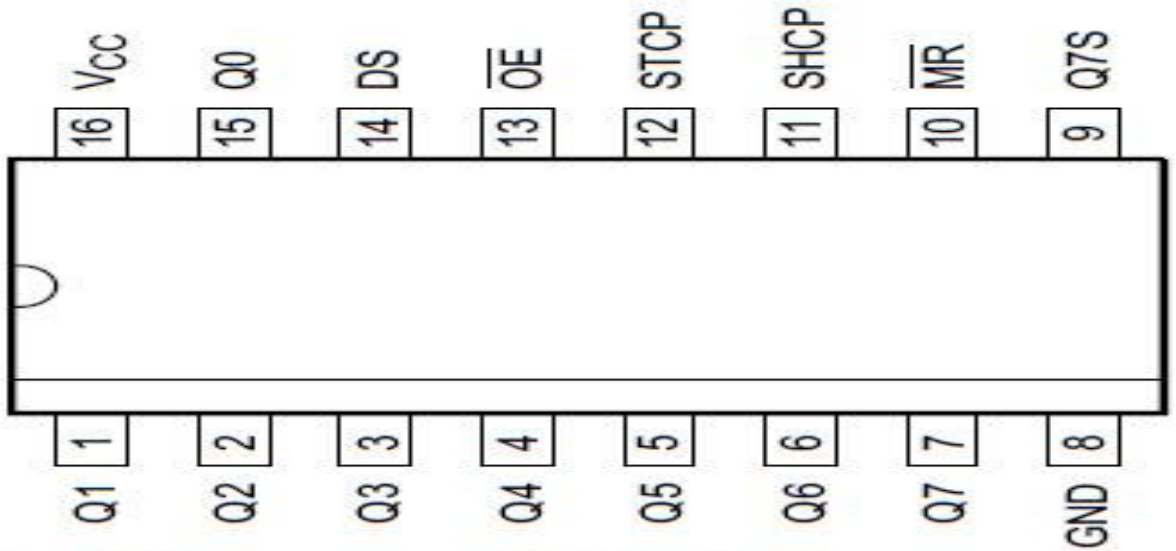
自訂義函數

```
int  函式名(參數)  //有反回值函數
 {
    陳述句;
    return  x;
 }
```

IC 74HC595



74HC595腳位介紹



VCC 5V GND 接地

Q0~Q7 並行輸出接腳

Q7S串口資料輸出可串接下一個74HC595

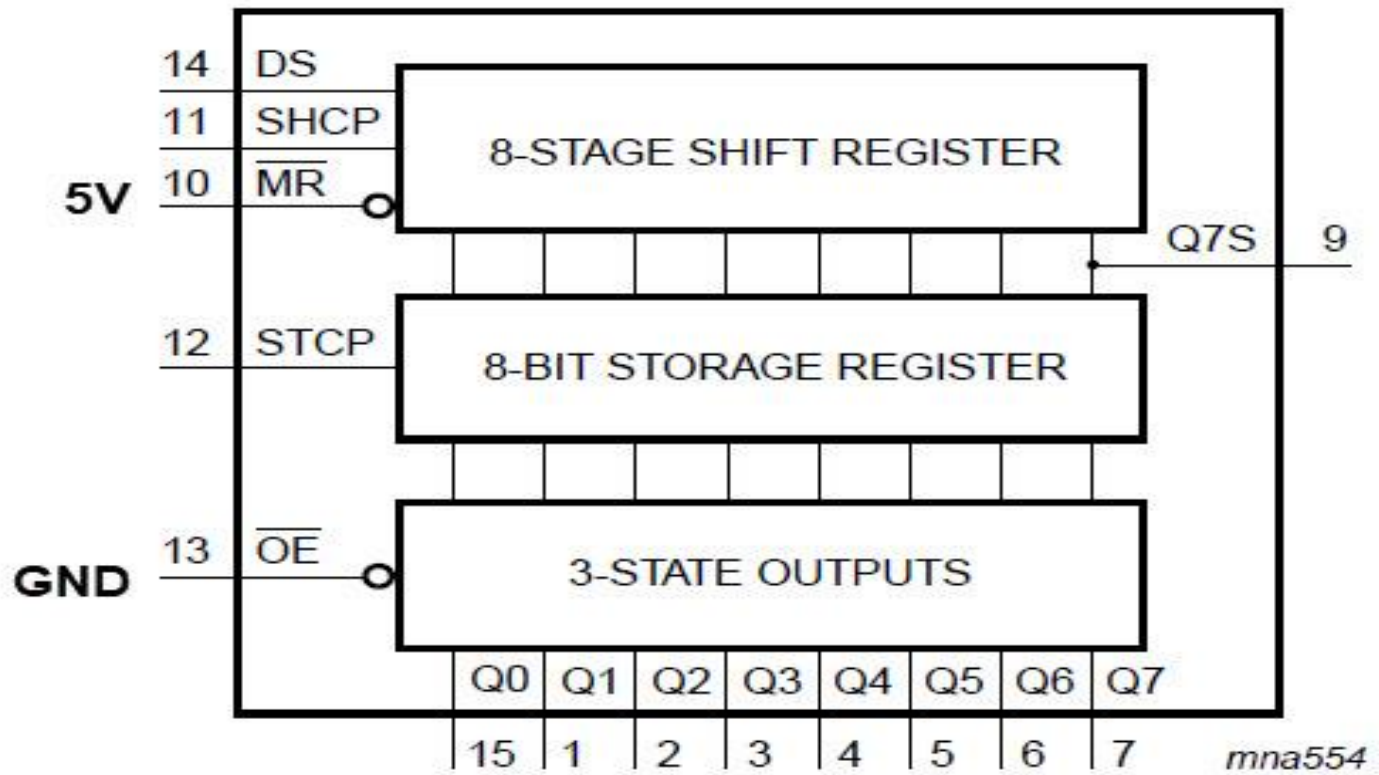
MR Reset (LOW清空數據 一般為HIGH)

OE OUTPUT ENABLE (LOW為輸出)

DS 數據HIGH LOW

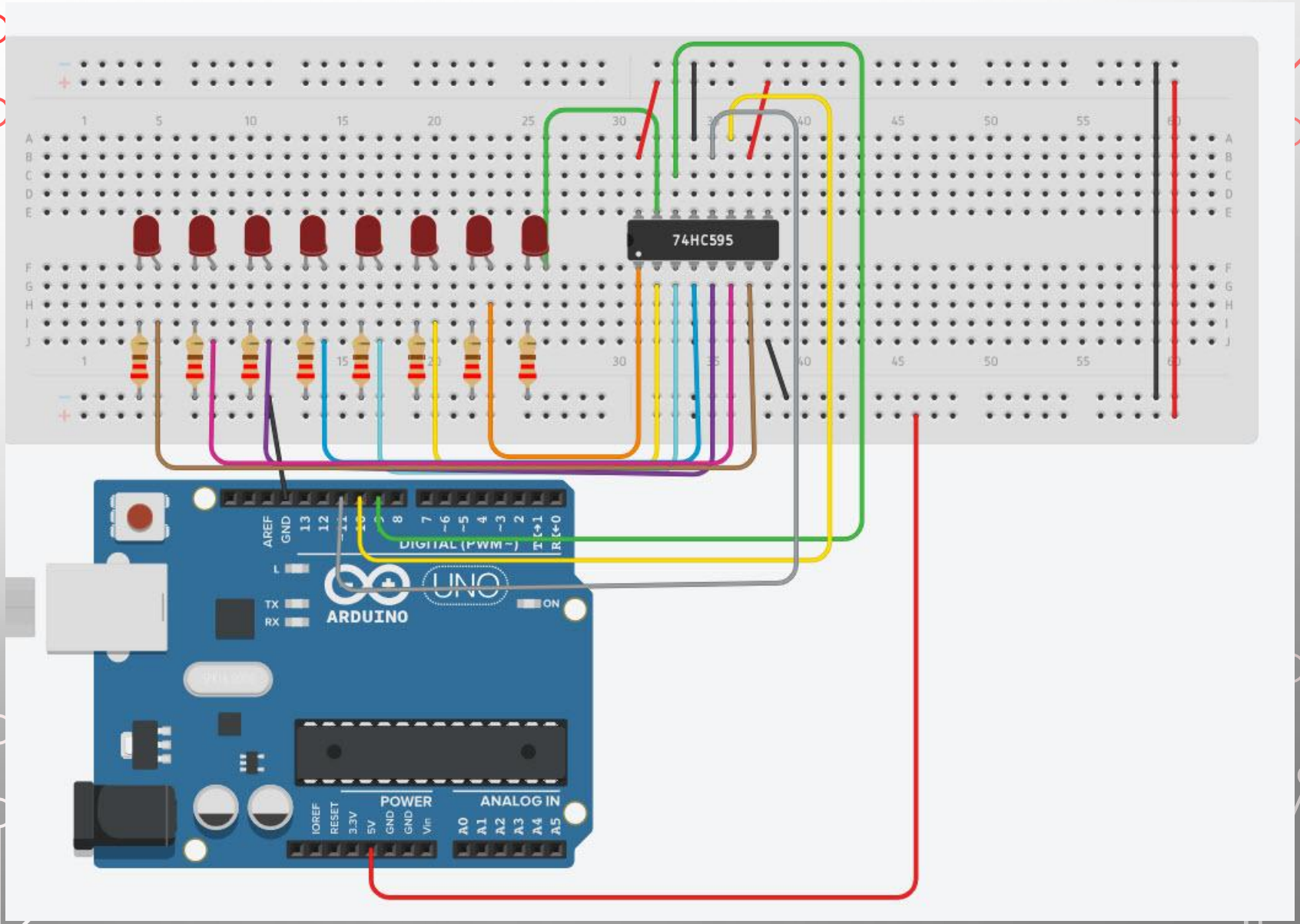
SHCP PUSH DS (資料推送器)

運作原理


14
DS
11
SHCP
5V
10
MR
8-STAGE SHIFT REGISTER
Q7S
9
12
STCP
8-BIT STORAGE REGISTER
GND
13
OE
3-STATE OUTPUTS
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
15
1
2
3
4
5
6
7
mna554

實作:IC　74HC595

- Arduino UNO
- 面包板
- LED *8
- 220Ω*8
- 74HC595


74HC595
ARDUINO
UNO
DIGITAL (PWM~)
POWER
ANALOG IN

```
int data=9;
int push=10;
int upload=11;
void setup()
{
  pinMode(data, OUTPUT);
  pinMode(push, OUTPUT);
  pinMode(upload, OUTPUT);
}
```

```
int value[8];
void pled()
{
  digitalWrite(upload,LOW);
  for(int i=0;i<8;i++)
  {
    digitalWrite(push,LOW);
    digitalWrite(data,value[i]);
    digitalWrite(push,HIGH);
  }
  digitalWrite(upload,HIGH);
}
```

```
void loop()
{
  for(int i=0;i<8;i++)
  {
    value[i]=1;
    pled();
    delay(300);
  }

}
```

引用類別庫

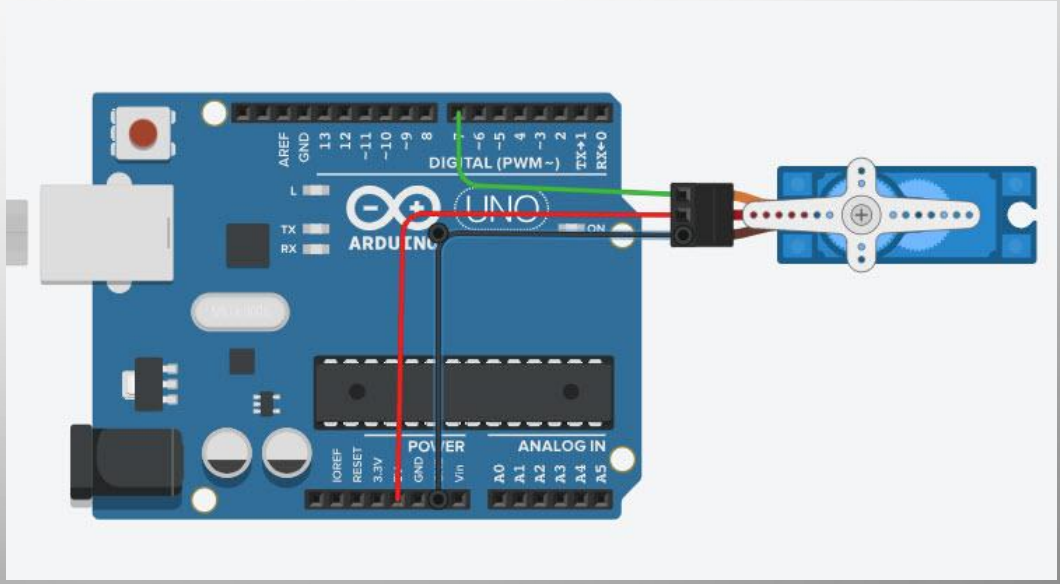
#INCLUDE

引用內建類別庫(#INCLUDE)

- 實作伺服馬達#include <Servo.h>
- LCD1602 #include <LiquidCrystal.h>

利用串口資訊傳送伺服馬達角度

- ARDUINO UNO
- 伺服馬達



```
#include <Servo.h>
Servo s1;  //實例化一個伺服馬達
int angle=0;//初始化角度為0
void setup()
{
 s1.attach(7);  //宣告馬達接腳
 Serial.begin(9600);//宣告串口波特率
}
void loop()
{
  //檢查串口是否有數據傳來
  if(Serial.available()>0)
  {
    //將數據轉為數值
    angle=Serial.parseInt();
    s1.write(angle);//指定伺服馬達角度
   }
}
```

CONSTRAIN(X, A, B)限制X 只能在A與B之間

X 值如果介於A 與B之間X=X

X值如果小於A 則X=A

X值如果大於B 則X=B

```
angle=constrain(angle,0,180);
```

LCD 1602

- 本次實驗使用arduino直接驅動1602液晶顯示字母
  1602液晶在應用中非常廣泛，最初的1602液晶使用的是HD44780控制器，現在各個廠家的1602模組基本上都是採用了與之相容的IC，所以特性上基本都是一致的。

- **1602LCD主要技術參數**
  顯示容量為16×2個字元;
  晶片工作電壓為4.5～5.5V;
  工作電流為2.0mA（5.0V）;
  模組最佳工作電壓為5.0V;
  字元尺寸為2.95×4.35（W×H）mm。

1602採用標準的16腳介面，其中:

第1腳：VSS為接地電源

第2腳：VDD接5V正電源

第3腳：VO為液晶顯示器對比度調整端，接5V電源時對比度最弱，接地電源時對比度最高，對比度過高時會產生“殘影”，使用時可以通過一個10KΩ的電位器調整對比度

第4腳：RS為寄存器選擇，高電平時選擇資料寄存器、低電平時選擇指令寄存器。

第5腳：R/W為讀寫信號線，高電平時進行讀操作，低電平時進行寫操作。當RS和RW共同為低電平時可以寫入指令或者顯示位元元元址，當RS為低電平RW為高電平時可以讀信號，當RS為高電平RW為低電平時可以寫入資料。

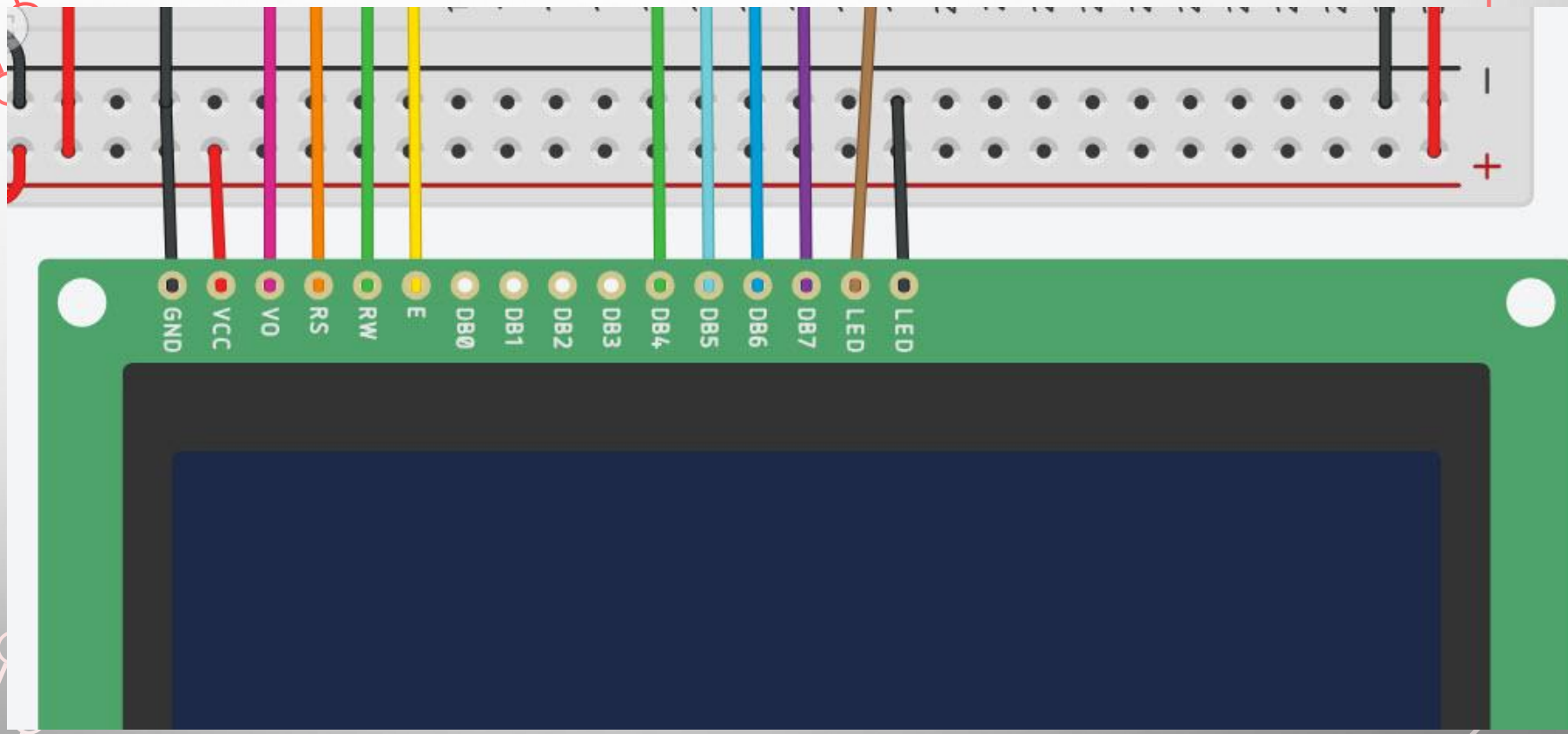
第6腳：E端為使能端，當E端由高電平跳變成低電平時，液晶模組執行命令。

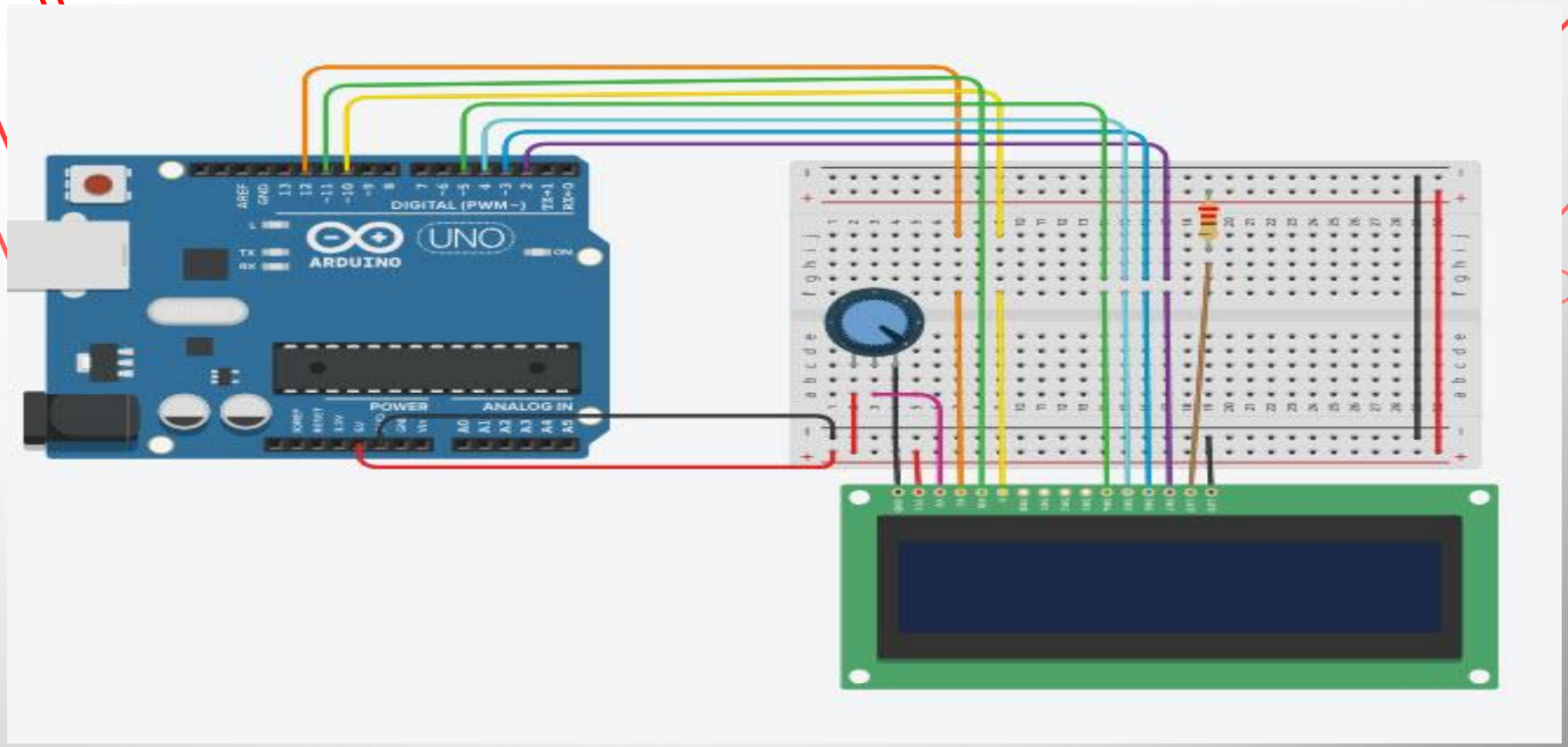
第7～14腳：D0～D7為8位元雙向資料線。

第15腳：背光電源正極接220Ω電阻

第16腳：背光電源負極

5V





ARDUINO	12	11	10	5	4	3	2
	↓	↓	↓	↓	↓	↓	↓
LCD 1602	RS	RW	EN	D4	D5	D6	D7

- 在Arduino的安装目錄下\libraries\LiquidCrystal可以查看到函數的原型
  - LiquidCrystal()——定義你的LCD的介面：各個引腳連接的I/O口編號
  - LiquidCrystal(rs, rw, enable, d2, d3, d4, d5)

- begin()——定義LCD的長寬（n列×n行），格式lcd.begin(cols, rows)
- clear()——清空LCD，格式lcd.clear()
- home()——把游標移回左上角，即從頭開始輸出，格式lcd.home()
- setCursor()——移動遊標到特定位置，格式lcd.setCursor(col, row)
- write()——在螢幕上顯示內容（必須是一個變數，如”Serial.read()”），格式lcd.write(data)
- print()——在螢幕上顯示內容（字母、字串，等等），格式lcd.print(data)

cursor()——顯示遊標（一條底線），格式lcd.cursor()

noCursor()——隱藏遊標，格式lcd.noCursor()

blink()——閃爍遊標，格式lcd.blink()

noBlink()——遊標停止閃爍，格式lcd.noBlink()

display()——（在使用noDisplay()函數關閉顯示後）打開顯示（並恢復原來內容），格式lcd.display()

noDisplay()——關閉顯示，但不會丟失原來顯示的內容，格式為lcd.noDisplay()

scrollDisplayLeft()——把顯示的內容向左滾動一格，格式lcd.scrollDisplayLeft()

scrollDisplayRight()——把顯示的內容向右滾動一格，格式為lcd.scrollDisplayRight()

autoscroll()——打開自動滾動，這使每個新的字元出現後，原有的字元都移動一格：如果字元一開始從左到右（默認），那麼就往左移動一格，否則就向右移動，格式lcd.autoscroll()

noAutoscroll()——關閉自動滾動，格式lcd.noAutoscroll()

leftToRight()——從左往右顯示，也就是說顯示的字元會從左往右排列（預設），但螢幕上已經有的字元不受影響，格式lcd.leftToRight()

rightToLeft()——從右往左顯示，格式lcd.rightToLeft()

定義及宣告

```
#include <LiquidCrystal.h> //申明1602液晶的函式程式庫
LiquidCrystal lcd(12,11,10,5,4,3,2); //4資料口模式連線聲明(RS,RW,E,D4,D5,D6,D7)
int i;
```

定義及宣告

```
void setup()
{
  lcd.begin(16,2);          //初始化1602液晶工作模式
While(1)
{
    lcd.home();
    lcd.print("Hello World");
}
```

SETCURSOR() 設定游標位置

```
//游標回到第一行第零列
    lcd.setCursor(0,1);
    delay(1000);  //暫停一下看才能看到游標
    //顯示游標
    lcd.cursor();
```

NODISPLAY　DISPLAY

```
//LCD字幕閃三下
    for(int i=0;i<3;i++)
    {
      lcd.noDisplay();
      delay(500);
      lcd.display();
      delay(500);
      }
```

SCROLLDISPLAYLEFT() 字向左動

```
for(i=0;i<24;i++)
  {
    lcd.scrollDisplayLeft();
    delay(500);
  }
```

CLEAR() 清空内容

```
lcd.clear();
```

重設新內容

lcd.setCursor(0,0);　　//把游標移回左上角，即從頭開始輸出

lcd.print("Hi,"); //顯示

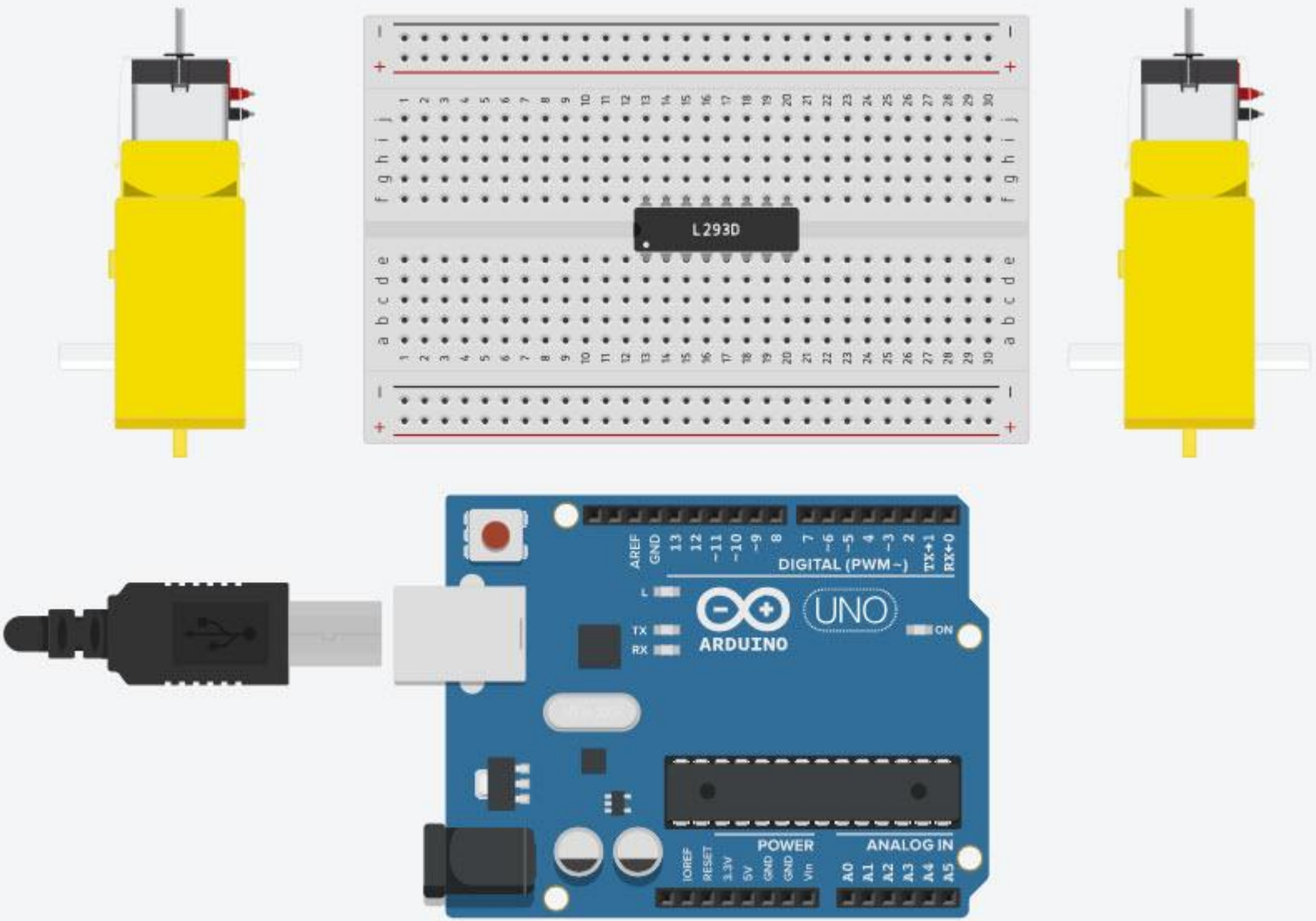
lcd.setCursor(0,1);　//把遊標定位在第1行，第0列

lcd.print("Arduino is fun");　　//顯示

delay(2000);

課當練習:模擬自走車
前進、後退、向左、向右

- ARDUINO UNO
- 面包板
- L293D
- 自走車電機*2


L293D
DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN

函數

- go(){} //向前
- back(){} //向後
- left(){} //向左
- Right(){}//向右