STEM 教育知識增益系列：
「製作多功能迷你控制台」學與教資源

引言

在中小學的 STEM 學習中，很多時也需利用微控制器，配合多過輸入及輸出元件，設計和製作不同類型的習作，過程中訓練學生的運算思維和編程技巧。這個工作坊能讓學習者如何利用開源的微控制器 Arduino，製作一個多功能的迷你控制台，例如一個可以收集科學實驗數據的儀器或一個音樂節拍器，你亦可以加上你的創意和運用編程技巧，編寫一個復古遊戲，製作一個迷你的遊戲機。

學習目標

1. 微控制器在 STEM 專題習作的應用。
2. 運用不同的感測器在製作 STEM 專題習作。
3. 電腦編程在電子控制中的角色。

學習內容

1. 簡介常用微控制器在 STEM 專題習作的應用。
2. 應用微控制器 Arduino 設計及製作多功能迷你控制台(節拍器/彈球遊戲) 。
3. 製作多功能迷你控制台工以進行 STEM 教育。
4. 微控制器在 STEM 專題研習的延伸使用。
5. 整合應用平面及立體設計軟件於鐳射切割和 3D 打印技術，製作多功能迷你控制台。

學習計劃

本專題習作需時大約 8 個課節，以下為一個學習計劃的參考例子，在這例子裏，每一節為 40 分鐘，而每週則設有兩個課節。

週數	課節	學習內容
1	1-2	1. 專題研習簡介 2. Arduino 的認識 輸入處理輸出 閃燈 按鈕 類比輸入
2	3-4	1. 結合 OLED 螢幕與 Arduino 的使用 2. 在 OLED 螢幕中顯示動畫和數據
3	5-6	利用 Arduino 及程式編寫，製作簡單電腦遊戲和節拍機
4	7-8	綜合使用平面及立體軟件，設計及製作多功能迷你工作枱的外殼 1. 平面軟件 2. 立體軟件

學習活動

<u>背景</u>

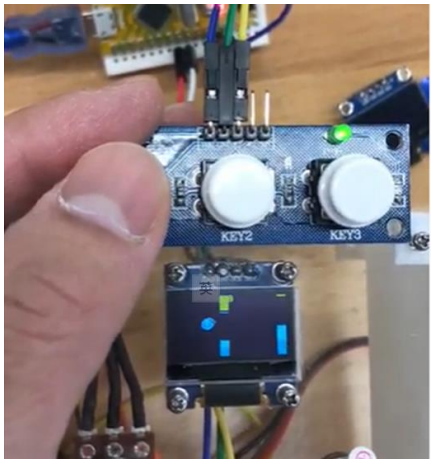
本學與教資源目的主要教授如何利用 Arduino 微處理器製作多功能迷你控制台，並以 Arduino IDE 編程，並將編寫好的程式上載到控制台，以有趣的方法學習編寫程式及運用電子感應器。

<u>建立系統所需的材料</u>

編號	元件	數量
1	Arduino Nano 電路板連 30cm USB 連接線	1 塊
2	0.96 寸 OLED IIC 通信 OLED 屏幕	1 塊
3	三位按键模組	1 塊
4	10K 電位器旋轉按鈕	1 粒
5	3V 有源蜂鳴器	1 粒
6	40 條 10cm 杜邦線(公公、公母、母母)	各一
7	小麵包板	1 塊
8	3M 螺絲、螺絲等	多枚
9	4 × AA 或 2 × 3.7V 或 9V 電池盒	一盒
10	LED	一顆
11	電阻 10K	一顆

學習成果

透過 Arduino 微處理器及不同的傳感器，配合編程技巧，同學可以製作出各種不同的設備，例如一個可以收集科學實驗數據的儀器，音樂節拍器，或編寫一個復古遊戲，製作一個迷你的遊戲機。



迷你的遊戲機

科學實驗數據儀器

延伸學習

相關的學與教資源，可參考由教育局開發的「以專題研習方式於 STEM 教育中教授機械人」學與教資源科技科目。詳情可參照下列網址：https://www.edb.gov.hk/tc/curriculum-development/kla/technology-edu/resources/tech-subjects/resources.html。

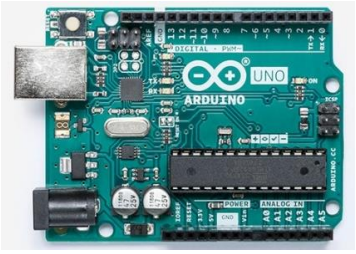
其他相關的參考網站：

- https://hackaday.io/project/19717-gbaa-game-boy-advance-arduino
- https://www.instructables.com/id/DIY-Cheap-Arduino-Gameboy/
- https://www.techbang.com/posts/23764-the-arduino-into-game-boy-play-smart-and-lovely-arduboy
- https://www.instructables.com/id/Build-an-ArduinoBoy/
- https://arduboy.com/

單元一：常用微控制器的應用和編程

在這階段裡，我們會介紹 Arduino 的特性，並且會為 Arduino Nano 作基本的編程。

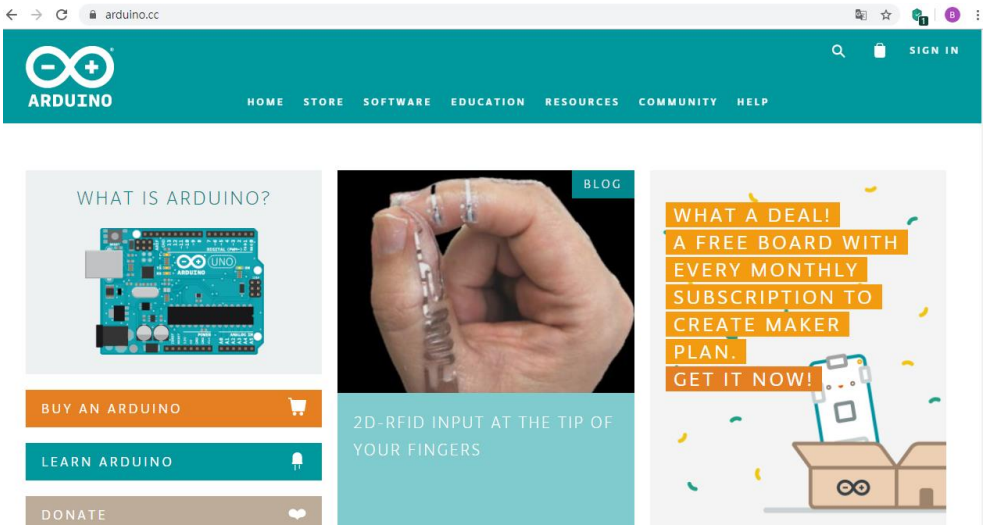
1.1 Arduino 簡介



Arduino 的特點：

- 開放源碼 Open source
- 硬體(Arduino I/O board)
- 軟體(Arduino IDE)
- 用感應器(sensor)能感應周遭環境變化(如：紅外線感應器、溫濕度感應器等)
- 能控制周遭裝置(如：燈、馬達)
- 電路可獨自運作
- 也可以搭配電腦運作，跟電腦溝通

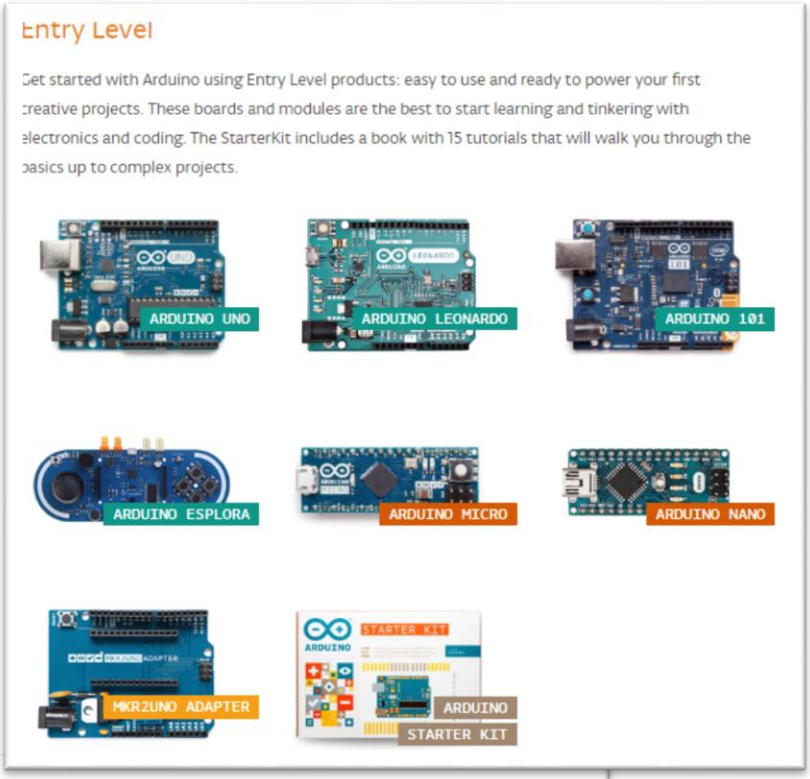
有關 Arduino 的學習資源，可瀏覽網址 https://www.arduino.cc/

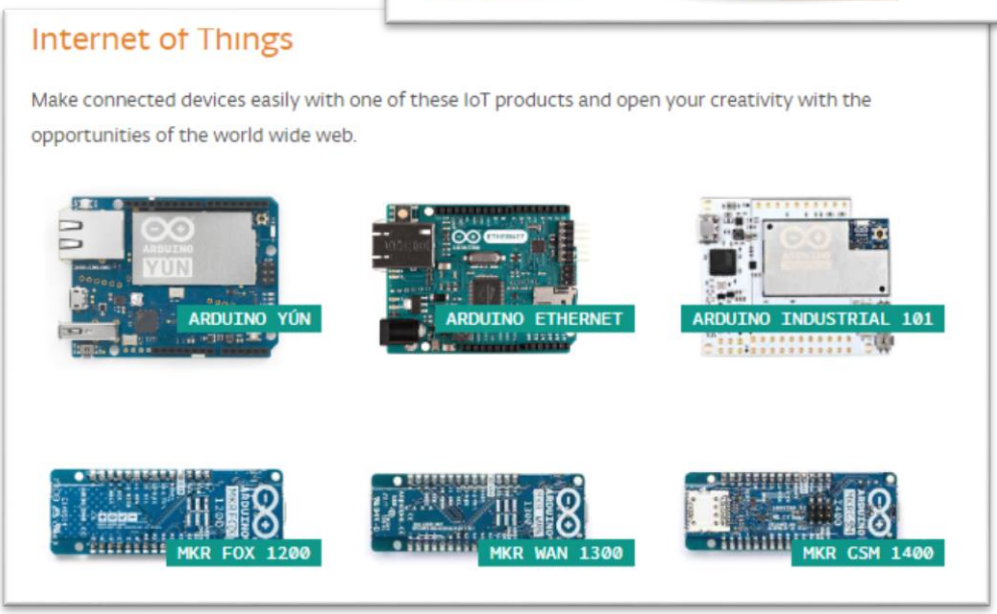


1.2 Arduino 家族

Arduino 微處理器有多個不同的版本，每個也有不同的外型，運算能力，輸入及輸出接口的數目與其他不同的功能。例如 Arduino UNO 有 14 個數碼輸入輸出接口，6 個模擬接口，適合一般中小學生用來製作 STEM 習作，而 Arduino Nano 和 UNO 有相同數目的接口，但體積較為細小，適合收藏於體積較小的裝置。如果想開展一些物聯網應用的習作，可以選擇 Arduino Yun，或使用一些 Arduino WiFi Shield，便可以將習作與互聯網連接起來。下列是 Arduino 家族中的一些例子：

- Arduino UNO
- Arduino Mega
- Arduino Nano
- Arduino Mini
- Arduino Pro Mini
- Arduino LilyPad

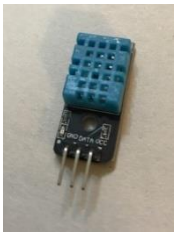




1.3 輸入輸出元件

Arduino 微處理器與不同的輸入輸出元件配合使用，便可以製作不同類型的習作。下列是一些常用的輸入及輸出元件：

<u>輸入元件</u>



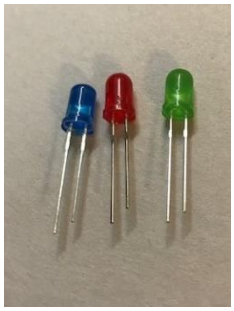
溫濕度傳感器

超音波感測器

土壤濕度傳感器

聲音傳感器

<u>輸出元件</u>



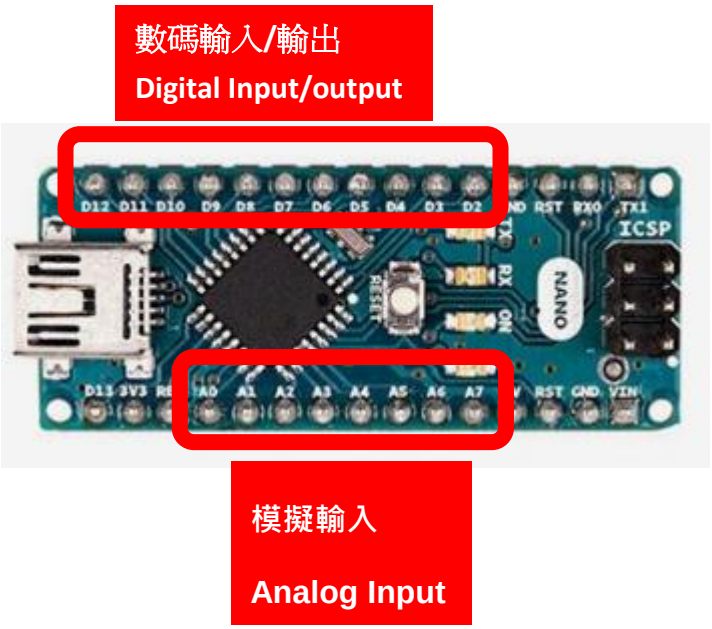
LED 小燈

蜂鳴器

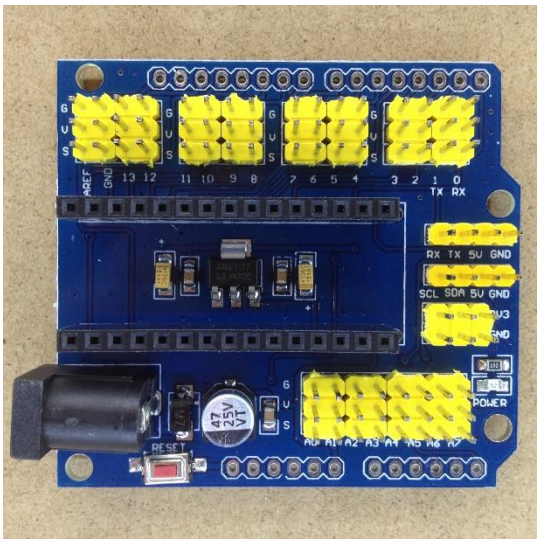
伺服馬達

1.4 Arduino 的結構

Arduino 微處理器中包含多個輸入及輸出的接口，下列 Arduino Nano 的結構。



配合擴展板的使用，同學在接駁線材的時候會更方便。



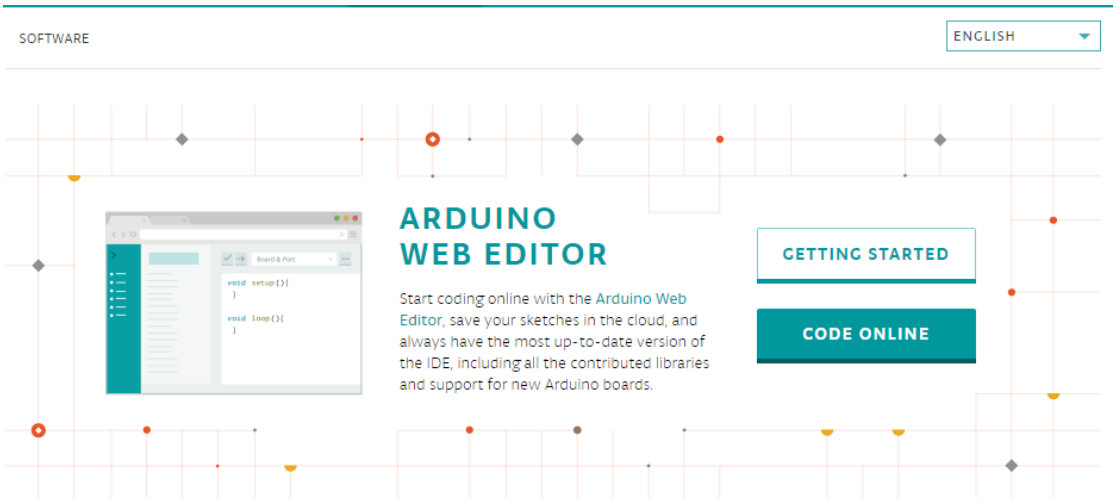
Nano 擴展板(Shield)

1.5 Arduino 編程軟件

除了需要預備 Arduino 開發版，我們還需要安裝 Arduino 開發工具軟體 IDE，有關軟件可到下列網址下載最新的版本。

https://www.arduino.cc/en/Main/Software

同學也可以選擇在線版本(需要繳付費用)

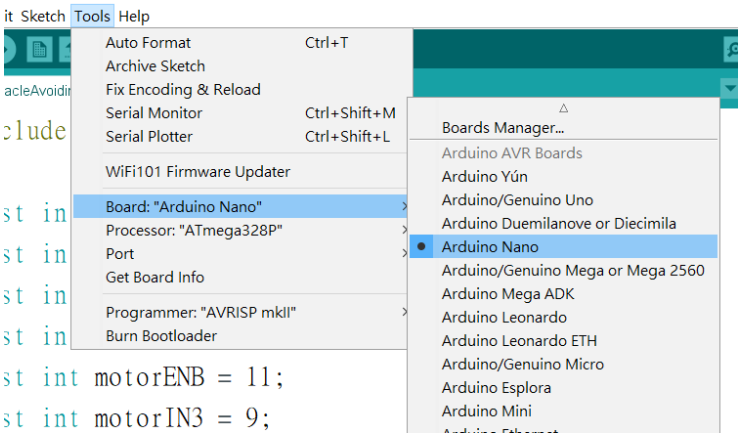


使用 mblock 編程工具，可以利用圖像化的介面進行程式編寫，有關軟件可到 www.mblock.cc 下載。

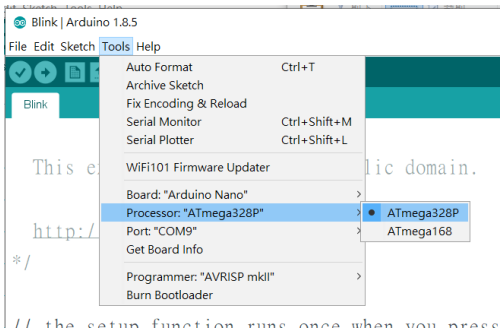
1.6 為編寫程式作好準備

同學在進行 Arduino 學習活動前，請按下列步驟預備：

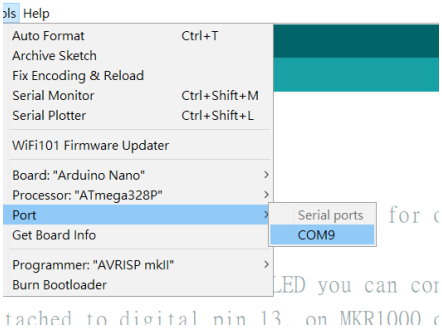
1. 將 Arduino Nano 插入小麵包板
2. 用 USB 線將 Arduino Nano 和電腦連接
3. 打開 Arduino IDE 軟件
4. 在 tools/Board 選”Arduino Nano”
5. 在 tools/Processer 選”ATmega328P”
6. 在 tools/Port 選 Arduino Nano 連接的 com



在 tools/Board 選”Arduino Nano”



選”ATmega328P”



選 Arduino Nano 連接的 com

單元二：Arduino 編程

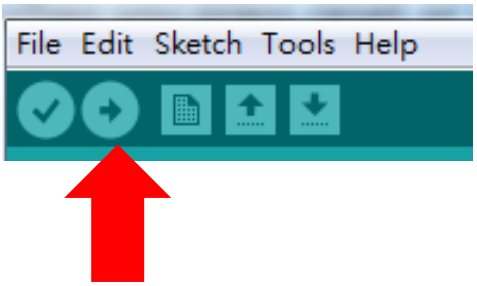
2.1 基本閃燈

在 Arduino IDE 開啟一個新的檔案，複製下列內容到檔案中。

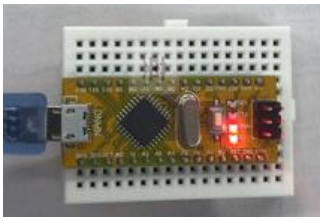
```
// the setup function runs once when you press reset or power the board

void setup() {
   pinMode(LED_BUILTIN, OUTPUT);   //  設定 digital pin LED_BUILTIN  作為輸出
}

// the loop function runs over and over again forever
void loop() {
   digitalWrite(LED_BUILTIN, HIGH);          //亮著 LED (HIGH)
   delay(1000);                              //等候 1 秒(1000 毫秒)
   digitalWrite(LED_BUILTIN, LOW);           //熄 LED (LOW)
   delay(1000);                              //等候 1 秒(1000 毫秒)
}
```

注意：在上述的程式碼中，包含了兩個循環，分別為 void setup()和 void loop()。在 void setup()中為程式設定，循環內的指令只會執行一次；而在 void loop()中，循環內的指令會無限循環。

```
void setup() {
            指令只會執行一次
}

void loop() {
            指令會無限循環
}
```

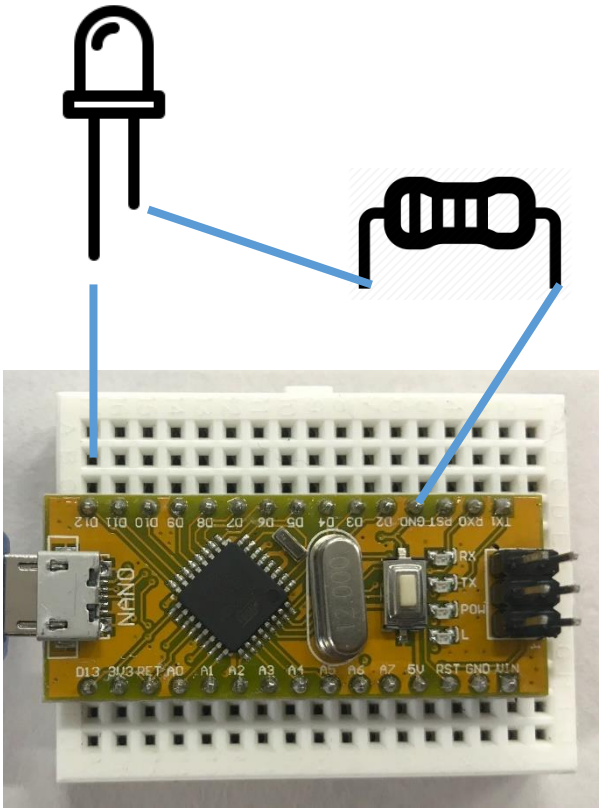


按下「上載」按鈕，便會將程式上載到 Arduino Nano 開發版中，Arduino 開發版的 LED 循環地亮一秒，熄一秒。



2.2 進階閃燈

按下面的線路圖將 LED 小燈和 10K 電阻器接駁起來。



在 Arduino IDE 開啟一個新的檔案，複製下列內容到檔案中。

```
void setup() {
    pinMode(12, OUTPUT);
}

void loop() {
    digitalWrite(12, HIGH);
    delay(1000);
    digitalWrite(12, LOW);
    delay(1000);
}
```

當程式上載到 Arduino Nano 開發版後，LED 小燈便會循環地亮一秒，熄一秒。

2.3 按鈕開關 LED 小燈

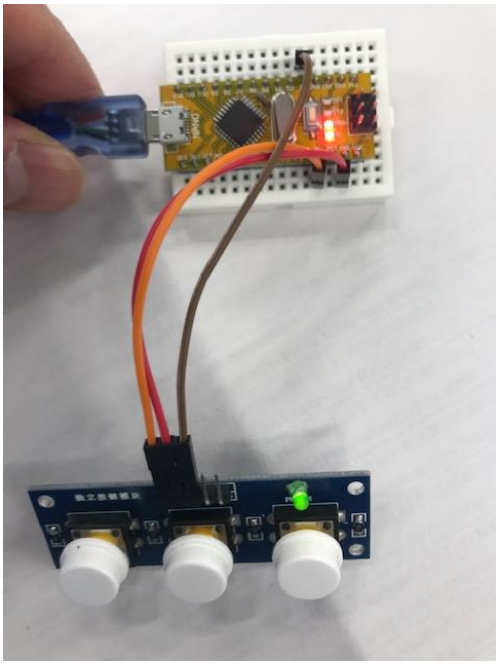
按下面的指示將按鈕模組和 Arduino Nano 開發版接駁起來。

VCC → 5V
GND → GND
K1 → D2

在 Arduino IDE 開啟一個新的檔案，複製下列內容到檔案中。

```
// constants won't change. They're used here to set pin numbers:
const int buttonPin = 2;        // the number of the pushbutton pin
const int ledPin =    13;        // the number of the LED pin
// variables will change:
int buttonState = 0;             // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}
void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed. If it is, the buttonState is LOW:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  } else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

當程式上載到 Arduino Nano 開發版後，Arduino 開發版的 LED 小燈，會隨著按鈕的狀態，亮起來或熄滅。

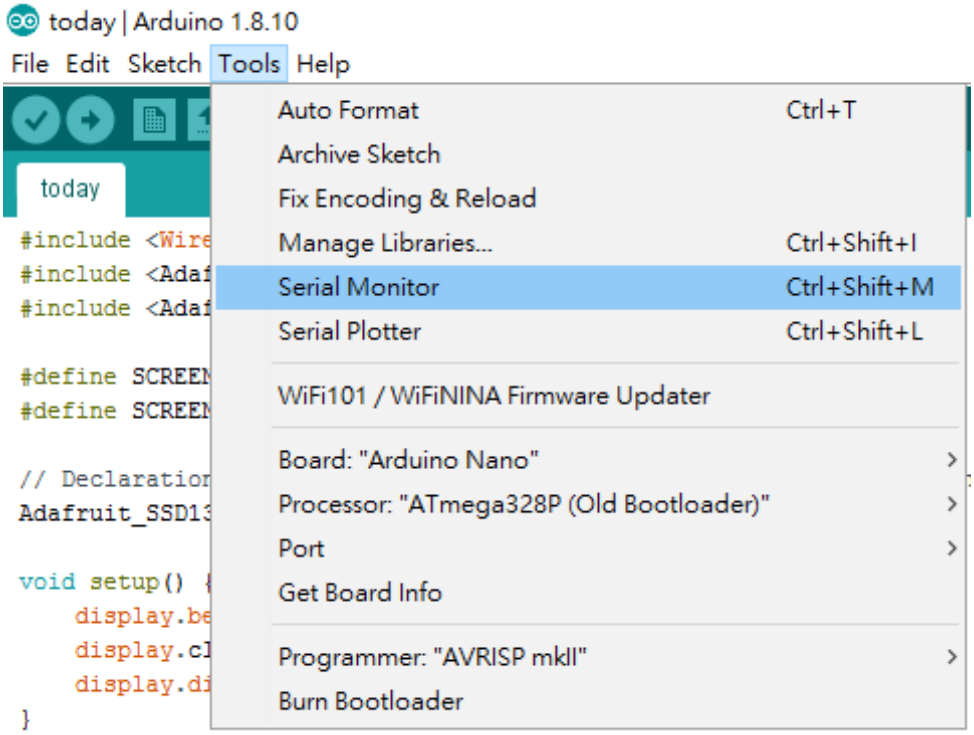


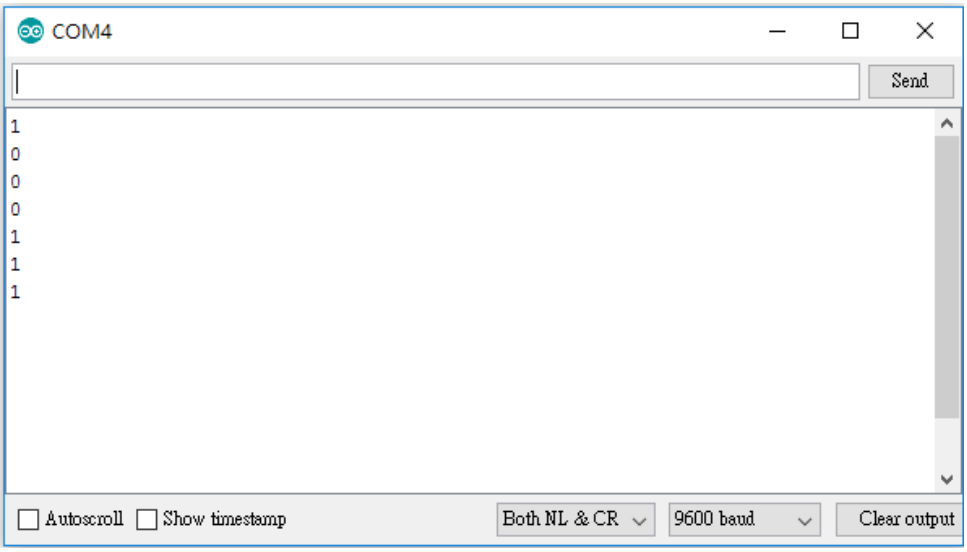
2.4 Serial Print 序列埠輸出

在 Arduino IDE 開啟一個新的檔案，複製下列內容到檔案中。

```
// digital pin 2 has a pushbutton attached to it. Give it a name:
int pushButton = 2;
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // make the pushbutton's pin an input:
  pinMode(pushButton, INPUT);
}
// the loop routine runs over and over again forever:
void loop() {
  // read the input pin:
  int buttonState = digitalRead(pushButton);
  // print out the state of the button:
  Serial.println(buttonState);
  delay(1);          // delay in between reads for stability
}
```

當程式上載到 Arduino Nano 開發版後，按下 Arduino IDE 的「Serial Monitor」，便可以看到按鈕的狀態。





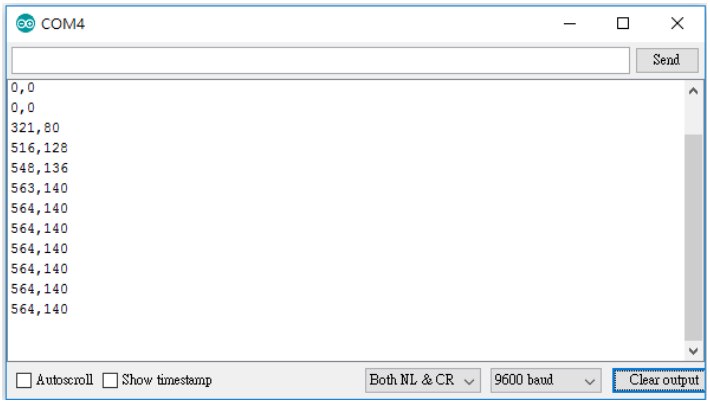
2.5 Analog Input 類比輸入

按下面的指示將電位器和 Arduino Nano 開發版接駁起來。

在 Arduino IDE 開啟一個新的檔案，複製下列內容到檔案中。

```
int sensor = A0;                                    //設定變數 sensor，由 A0 腳輸入
int sensorRead = 0;                                 //設定變數 sensor 的初始值為 0
int newdata = 0;
void setup() {
   Serial.begin(9600);                              //開啟序列埠
}

void loop() {
   sensorRead = analogRead(sensor);                 //讀取 A0 的訊號
   newdata = map(sensorRead, 0, 1023, 0, 255);      //將輸出的數值轉為 0-255
   Serial.print(sensorRead);
   Serial.print(",");
   Serial.println(newdata);                         //將讀到訊號傳到序列埠
   delay(500);                                      //停滯時間
}
```

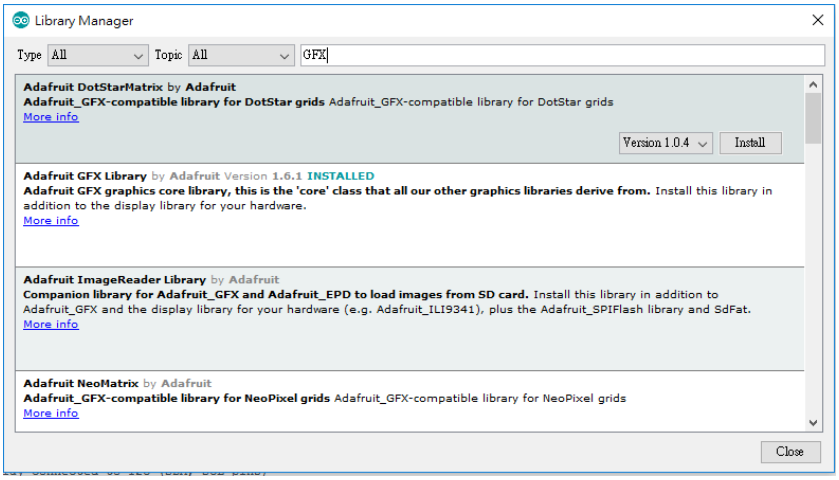
當程式上載到 Arduino Nano 開發版後，旋轉電位器的位置，輸入的電壓數據便會顯示在 Arduino IDE 的「Serial Monitor」中。

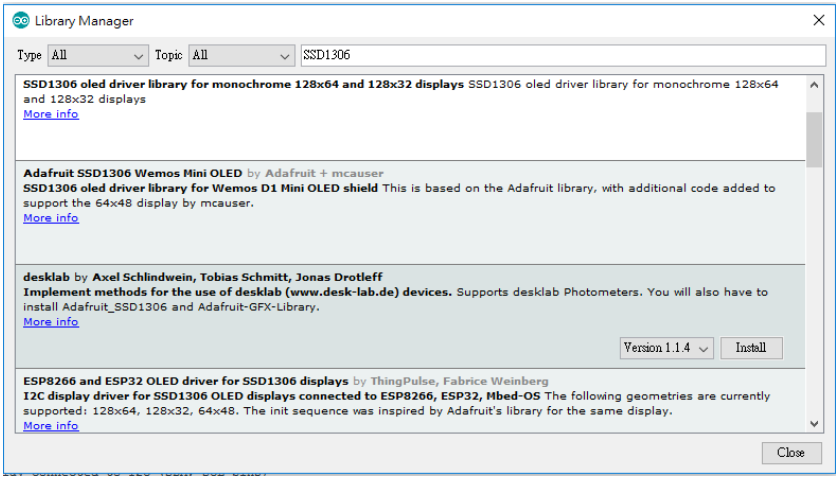


單元三：使用 OLED 螢幕

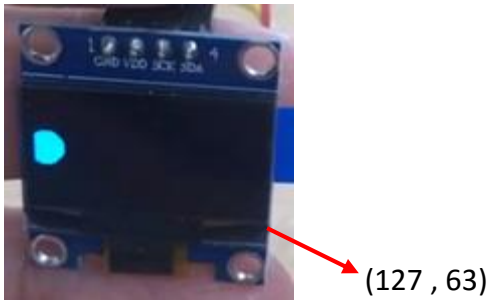
OLED 一款細小的螢幕顯示屏，解像度為 128X64，方便顯示一些數據或圖像。
使用前我們需在 Arduino IDE 中下載及安裝下列兩個 Library：

- Adafruit GFX Library
- SSD 1306 oled driver library for monochrome 128x64 displays





OLED 螢幕上的位置以坐標系統方式定位，左上角的座標為(0,0)，右下角的座標位(127, 63)，詳情可見下圖：

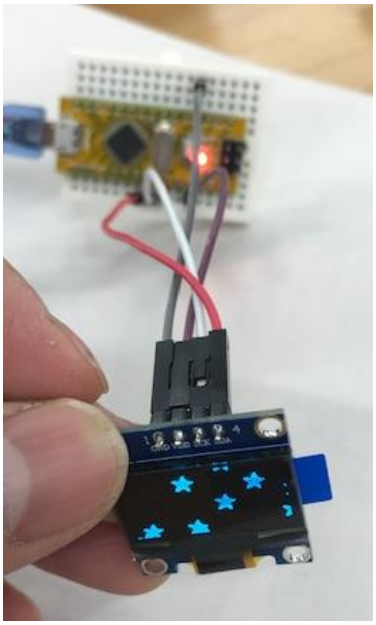


3.1 OLED 示範程式

按下面的指示將 OLED 螢幕和 Arduino Nano 開發版接駁起來。

OLED	→	Arduino
Vin	→	5V
GND	→	GND
SCL	→	A5
SDA	→	A4

當有關 OLED 螢幕的程式上載到 Arduino Nano 微處理器後，OLED 螢幕便會展示相關的動畫或數據。



OLED 螢幕展示動畫

如果同學想學習更多關於如何在 OLED 屏幕上，繪畫不同的形狀，可參照下列網址：

https://www.instructables.com/id/Fun-With-OLED-Display-and-Arduino/

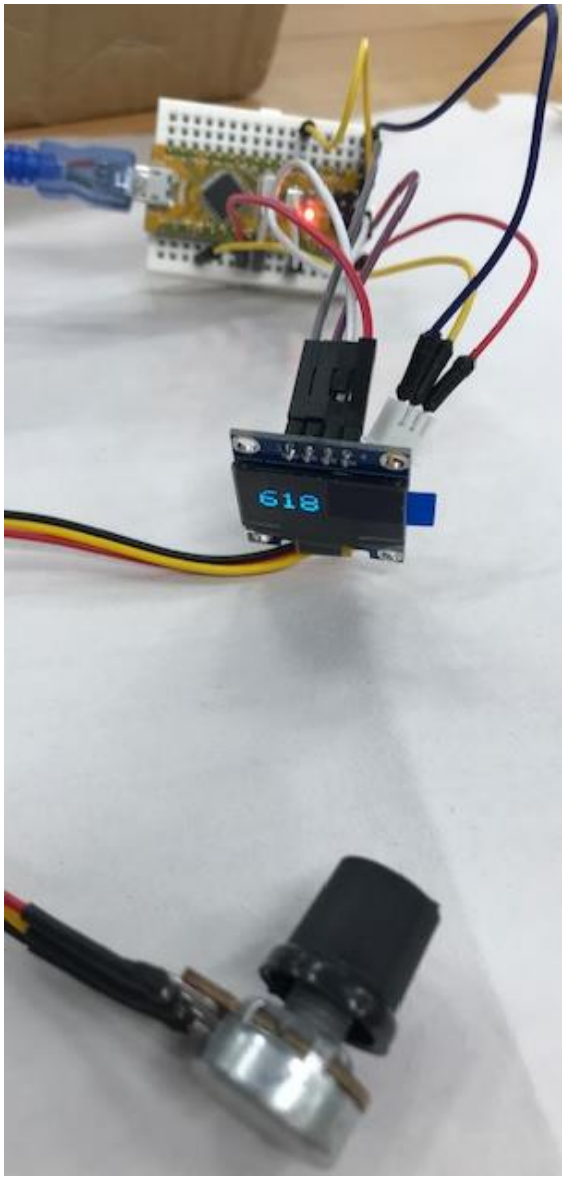
3.2 OLED 屏幕顯示即時數據

在 Arduino IDE 開啟一個新的檔案，複製下列內容到檔案中。

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

void setup() {
  Serial.begin(115200);

  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
  }
  delay(2000);
  display.clearDisplay();
  display.setTextSize(3);
  display.setTextColor(WHITE);
  display.setCursor(12, 20);
  display.println("ATEC");
  display.display();
  delay(1000);
}

void loop() {
  int sensorValue = analogRead(A0);
  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
  float voltage = sensorValue * (5.0 / 1023.0);
  // print out the value you read:
  Serial.println(voltage);
  display.setCursor(12, 20);
  display.print(sensorValue);
```

```
    display.display();
    display.clearDisplay();
    delay(1000);
}
```

當程式上載到 Arduino Nano 開發版後，OLED 螢幕便會展示電位器所輸入的數據。(參照影片：https://youtu.be/TCx-yYv4cxo)

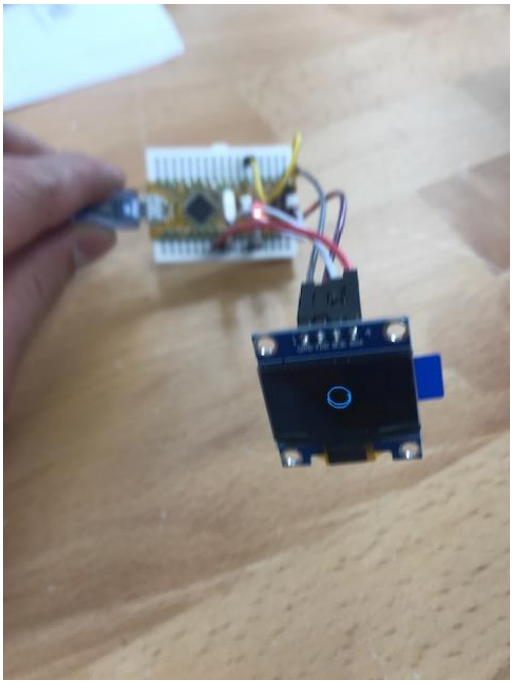


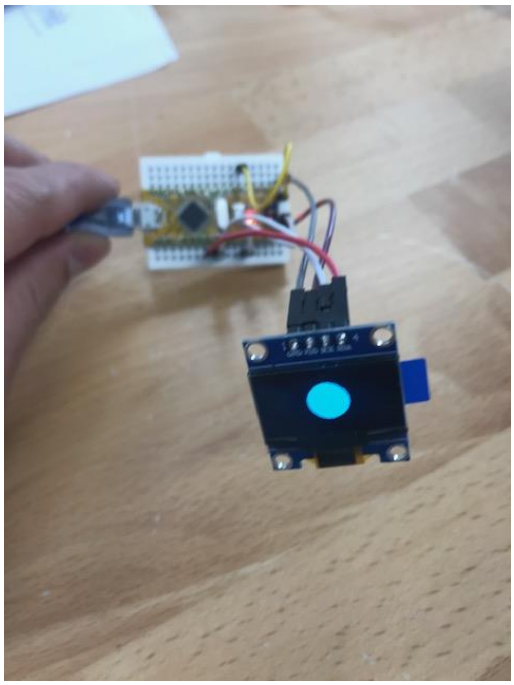
3.3 顯示閃動圓形

在 Arduino IDE 開啟一個新的檔案，複製下列內容到檔案中。

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

void setup() {
  Serial.begin(115200);

  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
  }
  delay(2000);
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0, 0);
  }

void loop() {
  int sensorValue = analogRead(A0);
  display.drawCircle(64, 32, 10, WHITE);
  display.display();
  delay(sensorValue);
  display.clearDisplay();
  display.fillCircle(64, 32, 20, WHITE);
  display.display();
  display.clearDisplay();
  delay(500);
}
```

當程式上載到 Arduino Nano 開發版後，OLED 螢幕便會顯示一個會閃動圓形，閃動時間為 0.5 秒。(參照影片：https://youtu.be/ouN0e6HquQQ)





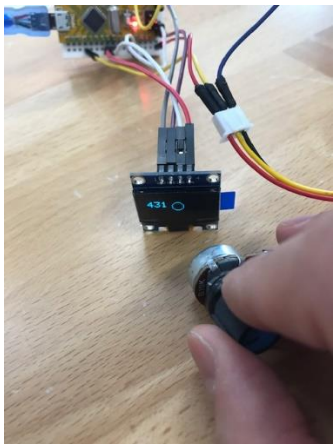
3.4 顯示閃動圓形及數據

在 Arduino IDE 開啟一個新的檔案，複製下列內容到檔案中。

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

void setup() {
  Serial.begin(115200);

  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
```

```
    }
    delay(2000);
    display.clearDisplay();
    display.setTextSize(2);
    display.setTextColor(WHITE);
    display.setCursor(0, 0);
}
void loop() {
    int sensorValue = analogRead(A0);
    display.drawCircle(64, 32, 10, WHITE);
    display.display();
    delay(sensorValue);
    display.clearDisplay();
    display.fillCircle(64, 32, 20, WHITE);
    display.display();
    display.clearDisplay();
    display.setCursor(12, 20);
    display.print(sensorValue);
    display.display();
}
```

當程式上載到 Arduino Nano 開發版後，OLED 螢幕便會顯示一個閃動圓形，其閃動時間會隨著電位器的輸入而改變，並會將電位器的數據顯示出來，類似一個節拍器。(參照影片：https://youtu.be/pnEyDUeWEUo)



同學可以嘗試配合使用蜂鳴器，製作出一個會發出聲音的節拍器。

3.5 左右移動的圓球

在 Arduino IDE 開啟一個新的檔案，複製下列內容到檔案中。

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

void setup() {
	display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
	display.clearDisplay();
	display.display();
}

int xpos=64;
int xstep=2;
void loop() {
	display.clearDisplay();
	display.fillCircle(xpos, 32, 10, WHITE);
	display.display();
	xpos=xpos+xstep;
	if (xpos>128 || xpos<0){
	xstep=xstep*-1;
	}
}
```

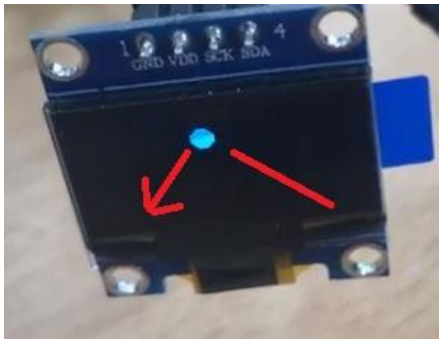
當程式上載到 Arduino Nano 開發版後，OLED 螢幕便會顯示一個圓球，向右方向移動，圓球到達螢幕的右邊邊緣，便會反彈，然後向左方移動，動作會不斷循環。(參照影片：https://youtu.be/b_Vnx_fZ3Bc)

3.6 上下左右移動的圓球

在 Arduino IDE 開啟一個新的檔案，複製下列內容到檔案中。

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
void setup() {
	display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
	display.clearDisplay();
	display.display();
}
int xpos=64;
int ypos=32;
int xstep=2;
int ystep=2;
void loop() {
	display.clearDisplay();
	display.fillCircle(xpos, ypos, 5, WHITE);
	display.display();
	xpos=xpos+xstep;
	ypos=ypos+ystep;
	if (xpos>128 || xpos<0){
	xstep=xstep*-1;
	}
	if (ypos>64 || ypos<0){
	ystep=ystep*-1;
	}
}
```

當程式上載到 Arduino Nano 開發版後，OLED 螢幕便會顯示一個圓球，向右下方移動，圓球到達螢幕的邊緣，便會反彈，動作會不斷循環。(參照影片：https://youtu.be/9aOBLbShFPg)



同學可以嘗試以「向量」的方法，編程程式。

3.7 加入隨機變數的移動圓球

在 Arduino IDE 開啟一個新的檔案，複製下列內容到檔案中。

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
void setup() {
        display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
        display.clearDisplay();
        display.display();
}
float angle = 0.785;
int xpos=64;
int ypos=32;
int rx=5;
int ry=5;
void loop() {
    float xstep=rx*cos(angle);
    float ystep=ry*sin(angle);
```

```
        display.clearDisplay();
        display.fillCircle(xpos, ypos, 5, WHITE);
        display.display();
        xpos=xpos+xstep;
        ypos=ypos+ystep;
        if (xpos>128 || xpos<0){
        rx=rx*-1;
        angle=0.785+0.001*random(700);
        }
        if (ypos>64 || ypos<0){
        ry=ry*-1;;
        angle=0.785+0.001*random(700);
        }
}
```

當程式上載到 Arduino Nano 開發版後，OLED 螢幕便會顯示一個圓球，向著上下左右的邊緣反彈移動，因為程式中加入隨機變數，令每次反彈的角度有所不同。

(參照影片：https://youtu.be/TcMaPCLtI8o)

3.8 加入球板的圓球遊戲

在 Arduino IDE 開啟一個新的檔案，複製下列內容到檔案中。

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

void setup() {
        display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
        display.clearDisplay();
        display.display();
}
float angle = 0.785;
int xpos=64;
int ypos=32;
int rx=2;
int ry=2;
int potvalue=512;
int batpos=64;

void loop() {
    display.clearDisplay();
    ball();
    bat();

    if(ypos>62){
      if(xpos>potvalue-15 && xpos<potvalue+15 ){
      ry=ry*-1;;
      angle=0.785+0.001*random(700);
      }
        else{
        xpos=64;
```

```
        ypos=32;
        delay(1000);
        }
    }
display.display();
}

void ball(){
        float xstep=rx*cos(angle);
        float ystep=ry*sin(angle);
        display.fillCircle(xpos, ypos, 5, WHITE);
        xpos=xpos+xstep;
        ypos=ypos+ystep;
        if (xpos>128 || xpos<0){
        rx=rx*-1;
        angle=0.785+0.001*random(700);
        }

        if (ypos<0){
        ry=ry*-1;;
        angle=0.785+0.001*random(700);
        }
}

void bat(){
    potvalue = analogRead(A0);
    display.fillRect(potvalue, 60, 15, 5, WHITE);
    }
```

當程式上載到 Arduino Nano 開發版後，OLED 螢幕便會顯示一個圓球，向著上下左右的邊緣反彈移動，玩家可以控制電位器，移動球拍的左右位置，當球拍能夠接住墜下來的圓球，圓球便會反彈，否則遊戲會重新開始。
(參照影片：https://youtu.be/Pqsml-qlnXc)

3.9 科學實驗收集數據 - 溫度和濕度

按下面的指示將溫度和濕度感測器 DHT11 和 Arduino Nano 開發版接駁起來。

DHT11	→	Arduino
GND	→	GND
DATA	→	D8
VCC	→	5V

在 Arduino IDE 開啟一個新的檔案，複製下列內容到檔案中。

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
#include "DHT.h"
#define dhtPin 8          //讀取 DHT11 Data
#define dhtType DHT11 //選用 DHT11
// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
DHT dht(dhtPin, dhtType); // Initialize DHT sensor
void setup() {
  Serial.begin(115200);//設定鮑率 9600
  dht.begin();//啟動 DHT
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
  }
  delay(2000);
  display.clearDisplay();
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.setCursor(12, 20);
  display.println("ATEC");// Display static text
  display.display();
  delay(1000);
```

```
}

void loop() {
  float h = dht.readHumidity();//讀取濕度
  float t = dht.readTemperature();//讀取攝氏溫度
  display.setCursor(12, 20);
  display.print(h);
  display.println("%");
  display.setCursor(12, 40);
  display.print(t);
  display.println("C");
  display.display();
  display.clearDisplay();
  delay(1000);
}
```

當程式上載到 Arduino Nano 開發版後，OLED 螢幕便會顯示溫度和濕度感測器 DHT11 的數據。

(參照影片：https://youtu.be/fBH0cESPTc0)

3.10 科學實驗收集數據 - 量度距離

按下面的指示將超音波感測器 HC-SR04 和 Arduino Nano 開發版接駁起來。

HC-SR04	→	Arduino
VCC	→	5V
GND	→	GND
Trig	→	D9
Echo	→	D10

在 Arduino IDE 開啟一個新的檔案，複製下列內容到檔案中。

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// defines pins numbers
const int trigPin = 9;
const int echoPin = 10;

// defines variables
long duration;
int distance;

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  Serial.begin(115200);

  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
```

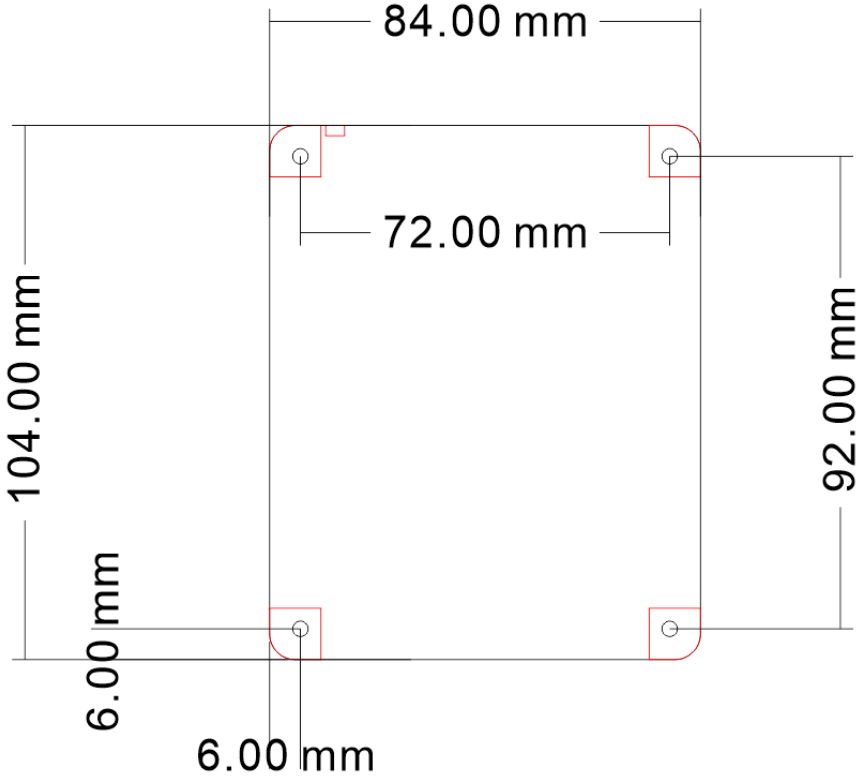
```
    }
    delay(2000);
    display.clearDisplay();
    display.setTextSize(3);
    display.setTextColor(WHITE);
    display.setCursor(0, 0);
}

void loop() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    // Sets the trigPin on HIGH state for 10 micro seconds
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    // Reads the echoPin, returns the sound wave travel time in microseconds
    duration = pulseIn(echoPin, HIGH);
    // Calculating the distance
    distance= duration*0.034/2;
    display.clearDisplay();
    display.display();
    display.setCursor(30, 20);
    display.print(distance);
    display.display();
}
```

當程式上載到 Arduino Nano 開發版後，OLED 螢幕便會顯示超音波感測器與前面的障礙物距離的數據。(參照影片：https://youtu.be/UodHEJJd-ZA)

單元四：設計及製作多功能迷你控制台的外殼

在這學與教資源中，我們將會利用平面與立體設計軟件，配合鐳射切割機和 3D 立體打印機，製作多功能迷你控制台的外殼。我們將會利用 CorelDRAW 平面設計軟件來繪畫面板和底板，完成後的圖樣將會輸出至鐳射切割機，並以 3mm 的亞加力膠板切割出來。另外，我們會以 Tinkercad 立體設計軟件，繪畫出外殼的側面，完成後的立體檔案將會輸出至 3D 立體打印機，並打印出來。最後將兩個成品合併起來，製作這個多功能迷你控制台的外殼。

外殼的設計，是基於所製作的迷你控制台的功能，在這裏我們以一個最簡單長方體，作為這個多功能迷你控制台的外殼。下列為面板的尺寸圖。



面板的尺寸圖

4.1 以 CorelDRAW 繪畫底板及面版

CorelDRAW 是一套簡單易學的向量繪圖軟體，提供多元化的創意工具。下圖顯示 CorelDRAW 的介面及一些常用的圖表功能。

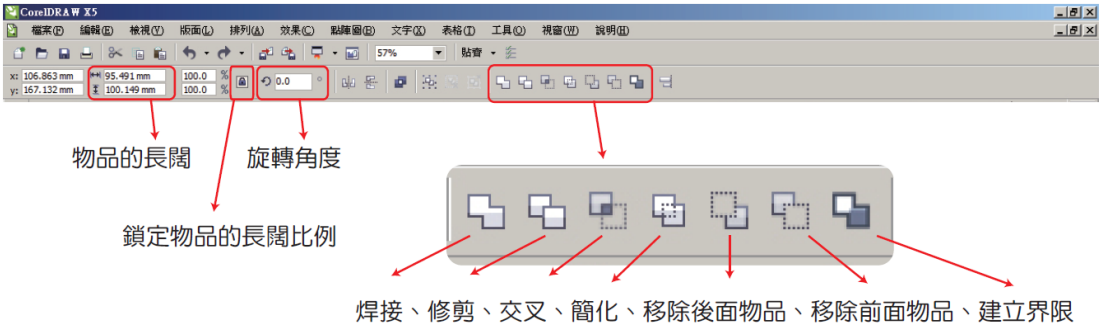
選取工具 Pick tool （ 快速鍵 空白 ）
選取工具用於選取物件，並可對物件進行移動、旋轉、縮放的編輯。

形狀工具
形狀工具用於編輯物件上的節點，選取節點後拖曳貝茲線控點，可調整節點兩側的曲線，或是透過屬性列改變節點的類型（尖角節點、平滑節點、對稱節點）。此外，也可以調整文字的字元位置。

顯示比例工具
更變文件視窗的放大縮小倍率

矩形工具 Rectangle tool
在繪製視窗中拖曳以繪製矩形或正方形，按著鍵盤的 Fn / Ctrl 鍵可繪製正方形。

多邊工具
輸入想要的邊數，在繪製視窗中拖曳以繪製多邊形，按著鍵盤的 Fn / Ctrl 鍵可繪製等邊圖形。



焊接
焊接是將幾個形結合一個圖形，新的圖形輪廓由被焊接的圖形邊界組成，被焊接圖形的交叉線將會消失。

焊接圖形前後的對比效果

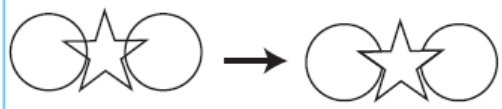
修剪
修剪是將目標對象與來源對象的相交部份裁掉，使目標圖像的形狀被更改。修剪前後的目標對象保留其填充和輪廓屬性。

修剪圖形前後的對比效果

交叉
交叉是將2個或2個以上對象的相交部份保留，使相交的部份成為一個新的圖像。新的圖像的填充和輪廓屬性將與目標對象相同。

交叉圖形前後的對比效果

簡化
簡化是減去後面圖形中和前面圖形的重疊部份，並保留前面圖形和後面圖形的狀態。



簡化圖形前後的對比效果

移除後面物品
移除後面物品是指減去後面圖形，並減去前後圖形的重疊部份，保留前面圖形的剩餘部份。

移除後面物品圖形前後的對比效果

移除前面物品
移除前面物品是指減去前面圖形，並減去前後圖形的重疊部份，保留後面圖形的剩餘部份。

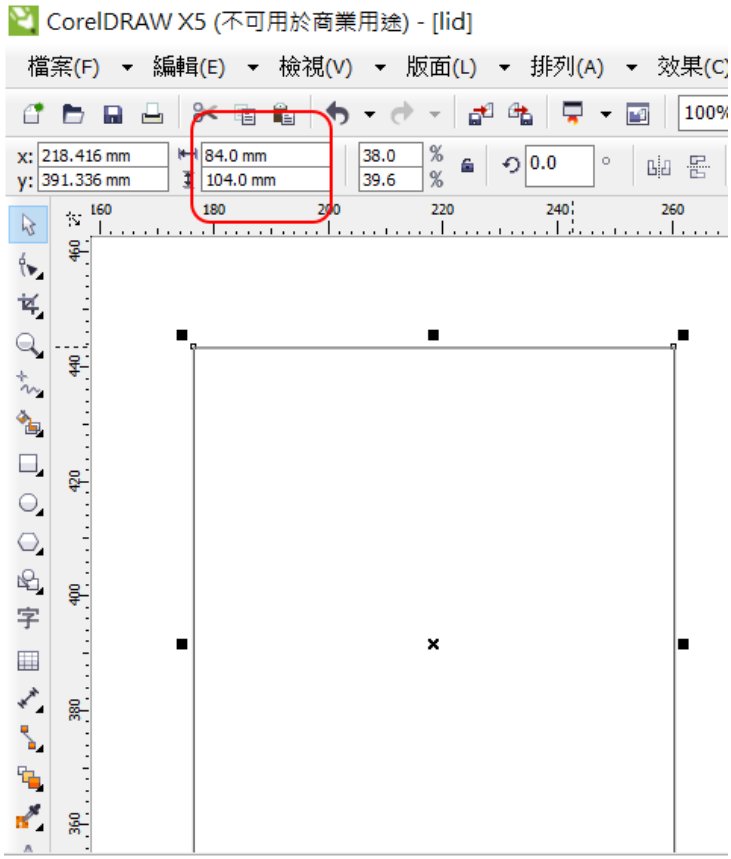
移除前面物品圖形前後的對比效果

建立界限
通過應用建立界限按鍵，可以快速創建一個所選圖形的共同邊界。

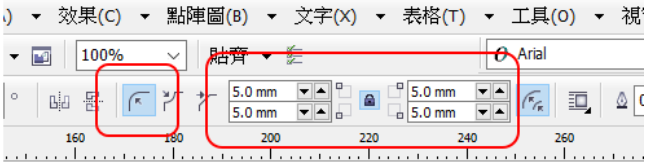
移除前面物品圖形前後的對比效果

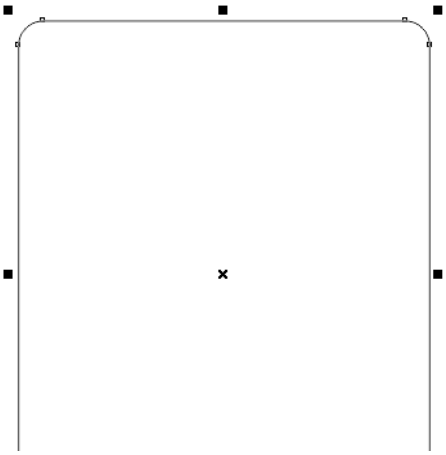
有了基本的認識後，我們便可以開始繪畫面板。

步驟一：畫一個闊 84mm 及高 104mm 的長方形

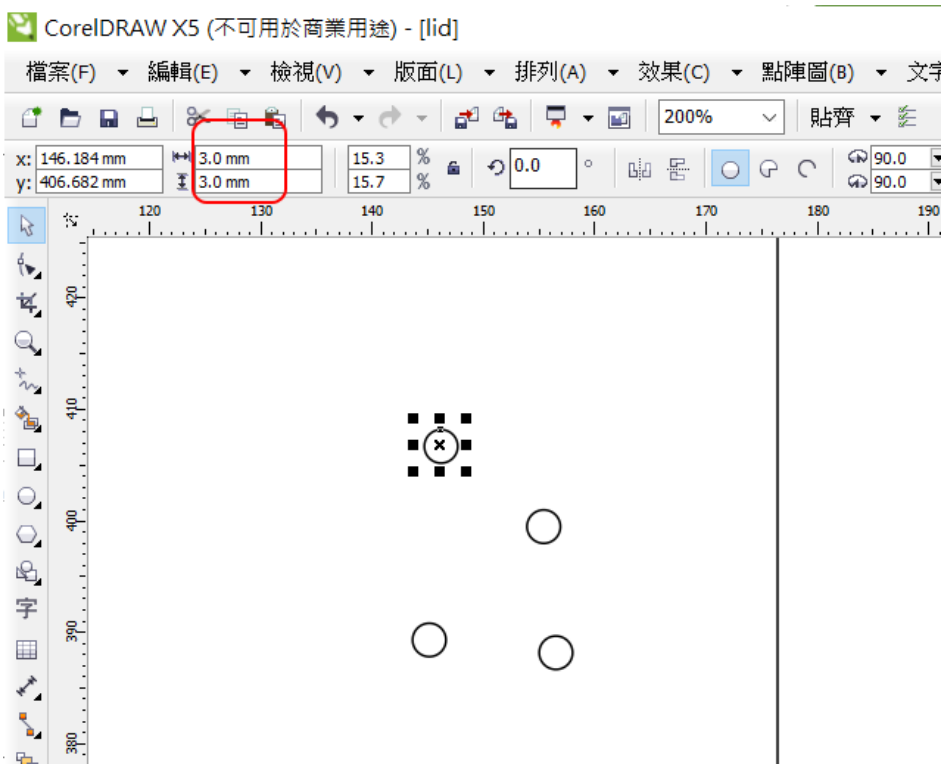


步驟二：利用「圓角」功能改變長方形的形狀

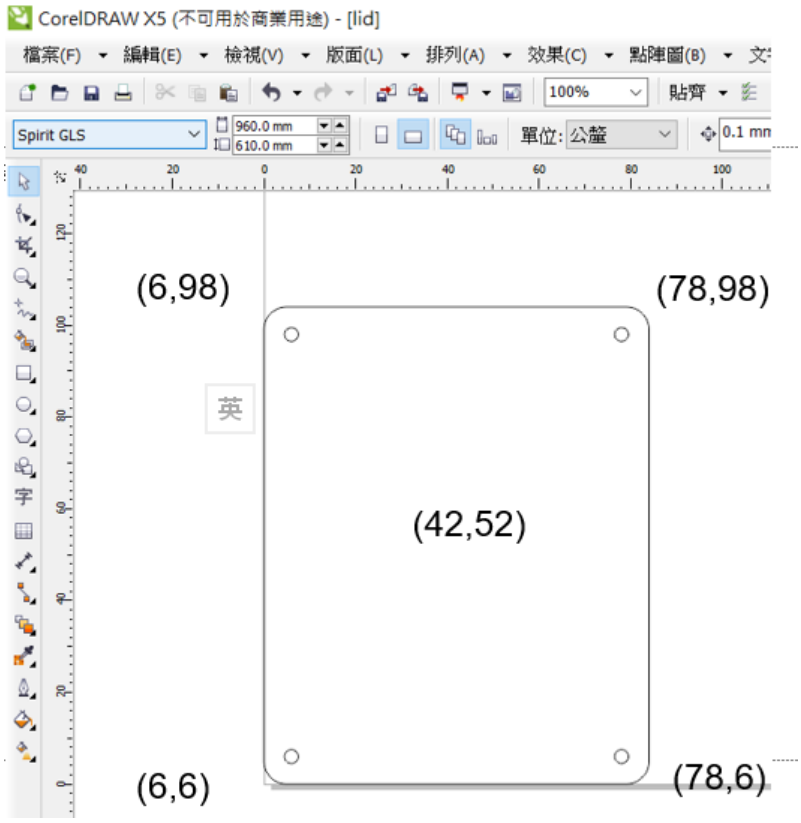




步驟三：畫 4 個直徑為 3mm 的圓形

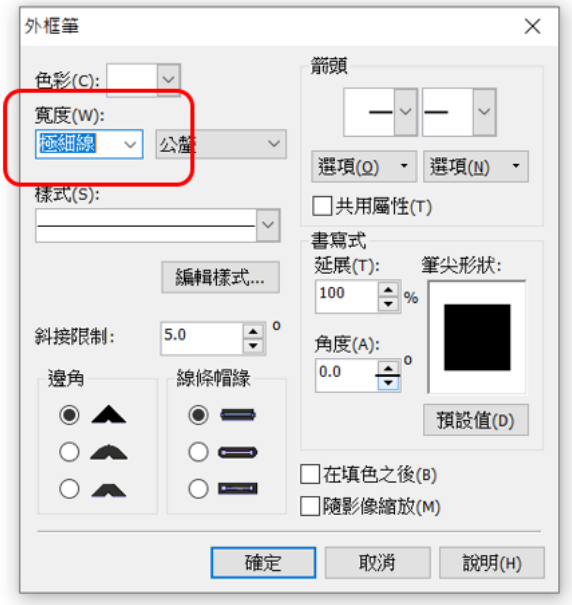


步驟四：將長方形和圓形放在適當的位置(括弧中的數字為各個物件的 x 及 y 坐標)

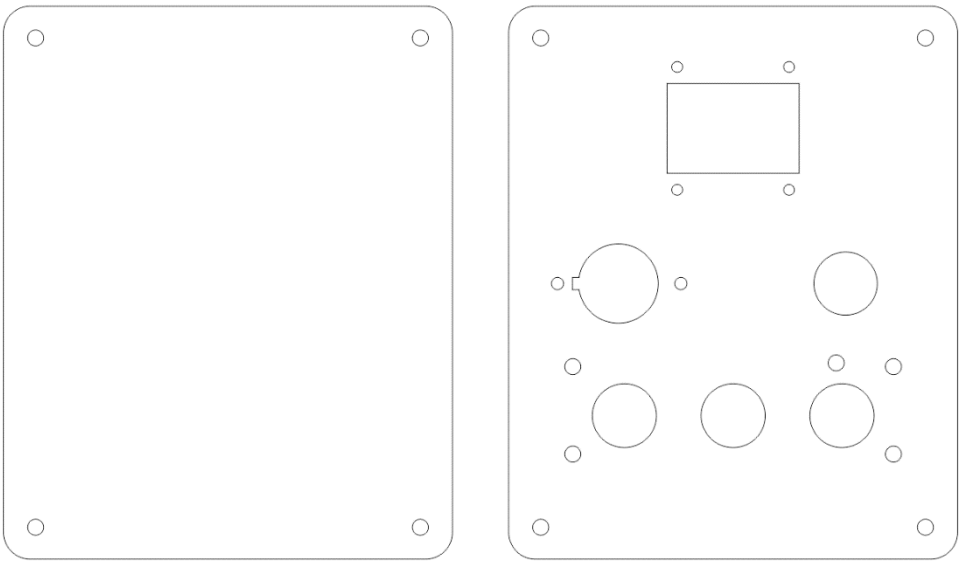


步驟五：將線條設置黑色，表示該線條的切割為穿透。

步驟六：將所有線變成極細線

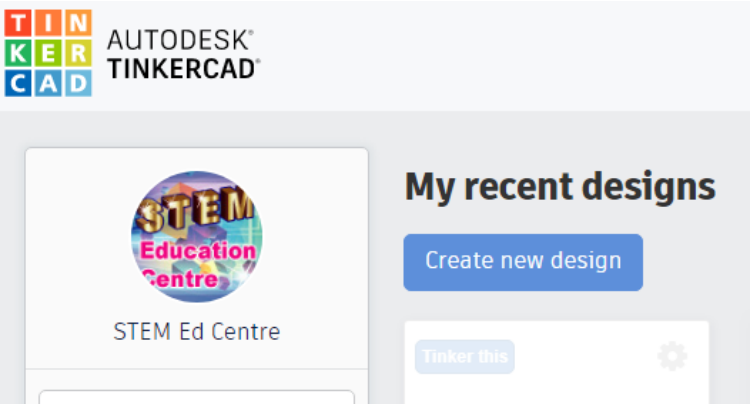


步驟七：將其它元件建立及拖拉到正確的位置

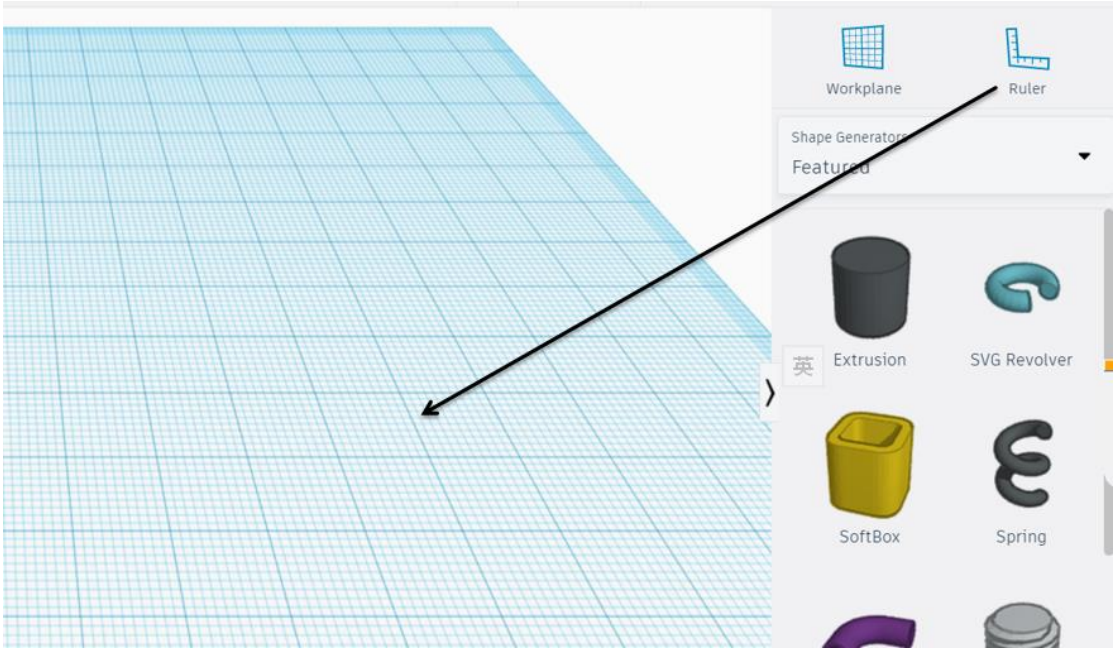


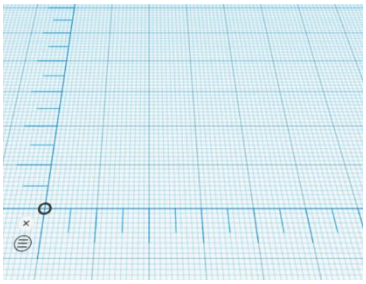
4.2 以 Tinkercad 繪畫外殼的側面

使用網頁瀏覽器進入 Tinkercad 網站(網頁地址為 www.tinkercad.com)。登入後，開啟一個新的立體設計，便可以開始繪畫外殼的側面。

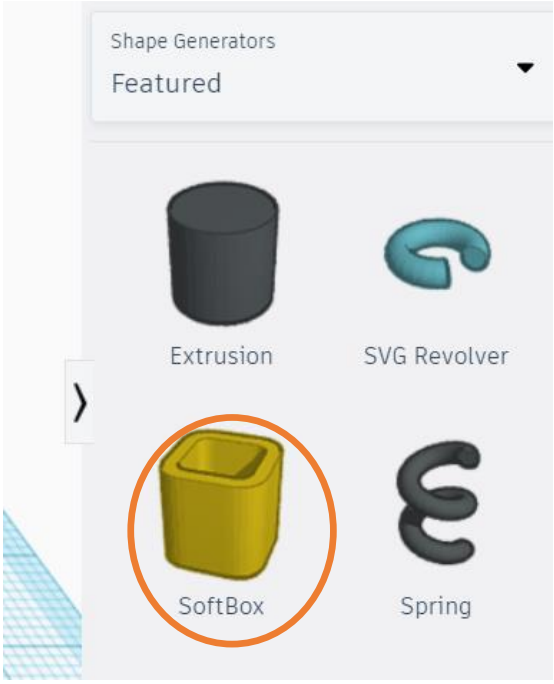


步驟一：先將尺規拖動到格仔紙上 作為物件尺寸的參考

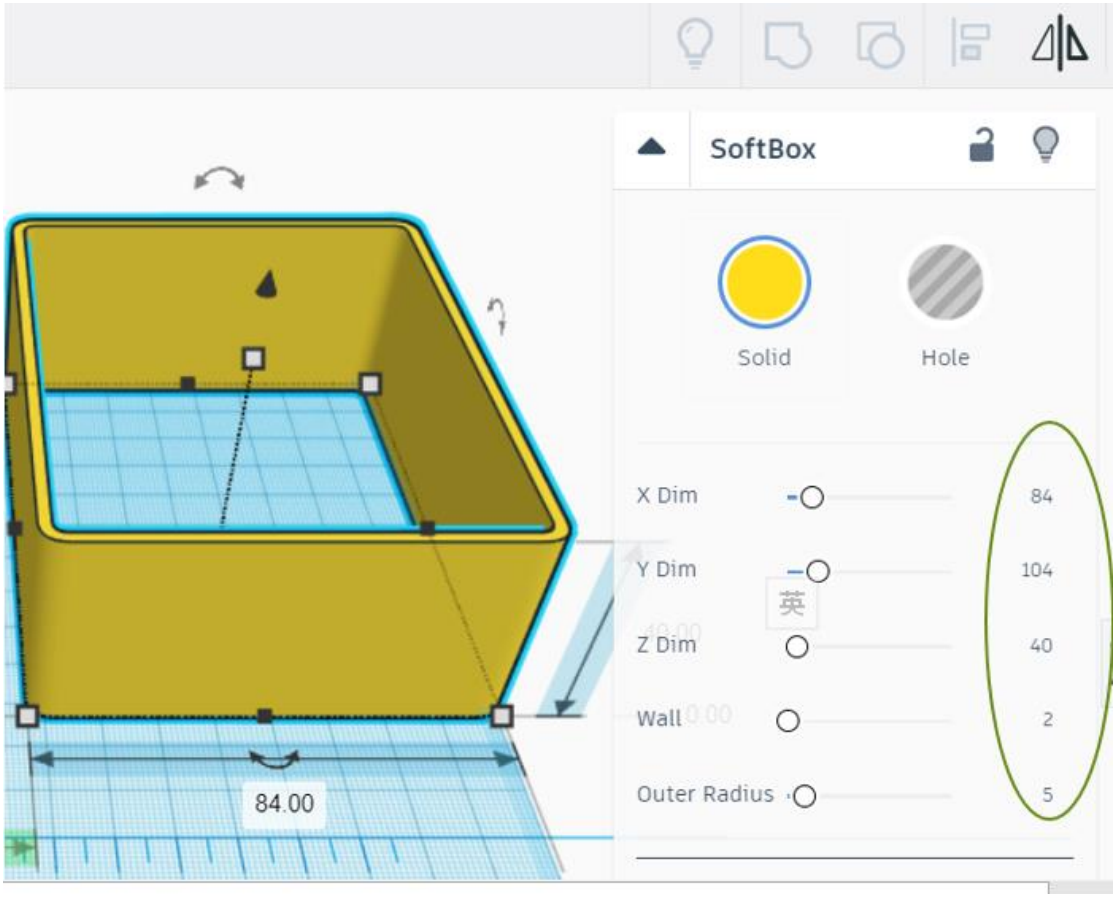




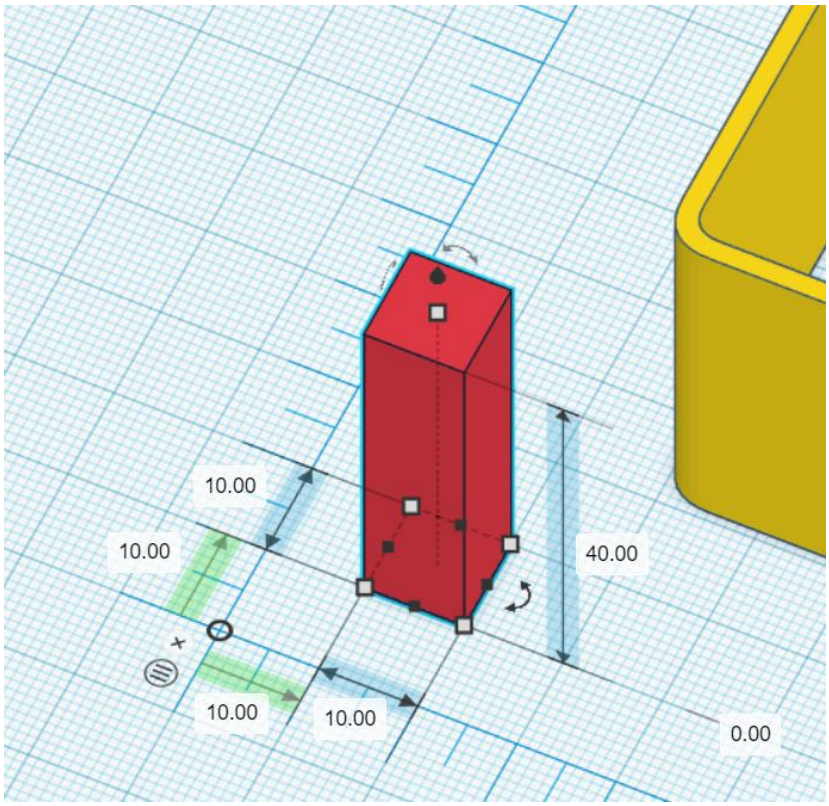
步驟二：選擇 Featured(特色) → SoftBox (柔光箱)



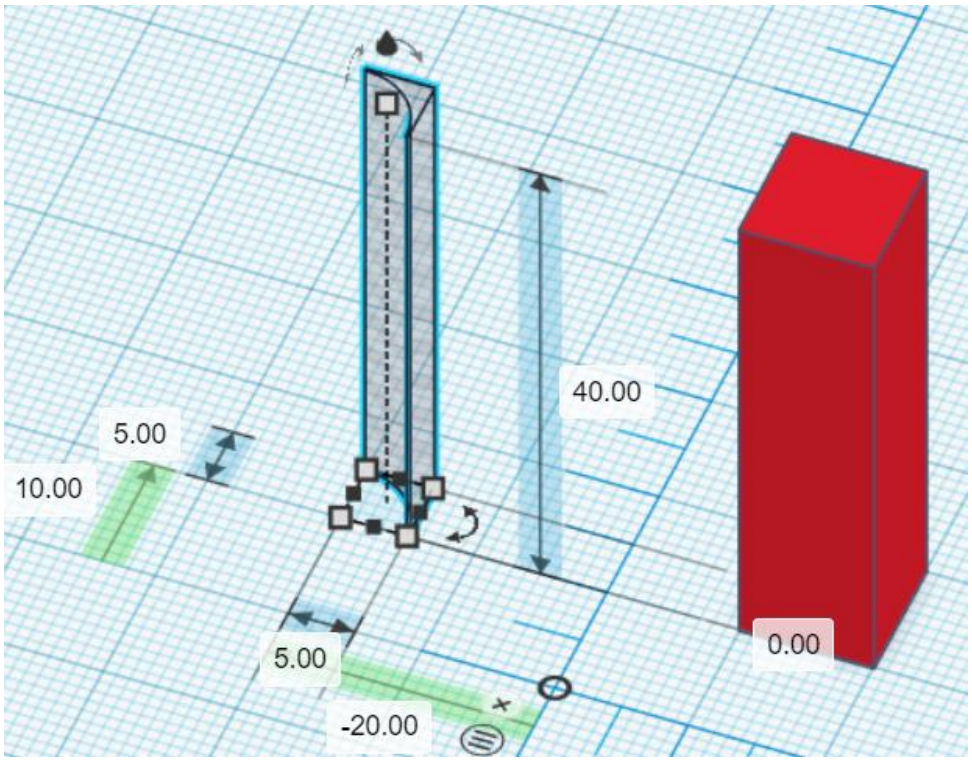
步驟三：設定柔光箱的數據，並將它放到(0,0)的位置



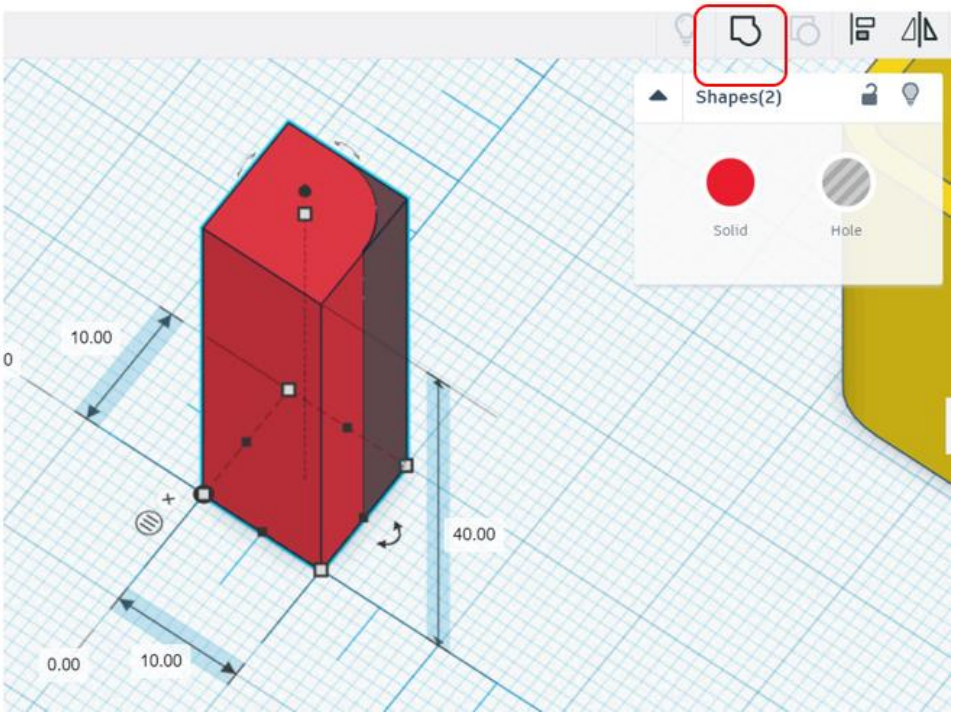
步驟四：畫一個長為 10mm、闊為 10mm 及高為 40mm 的長方體



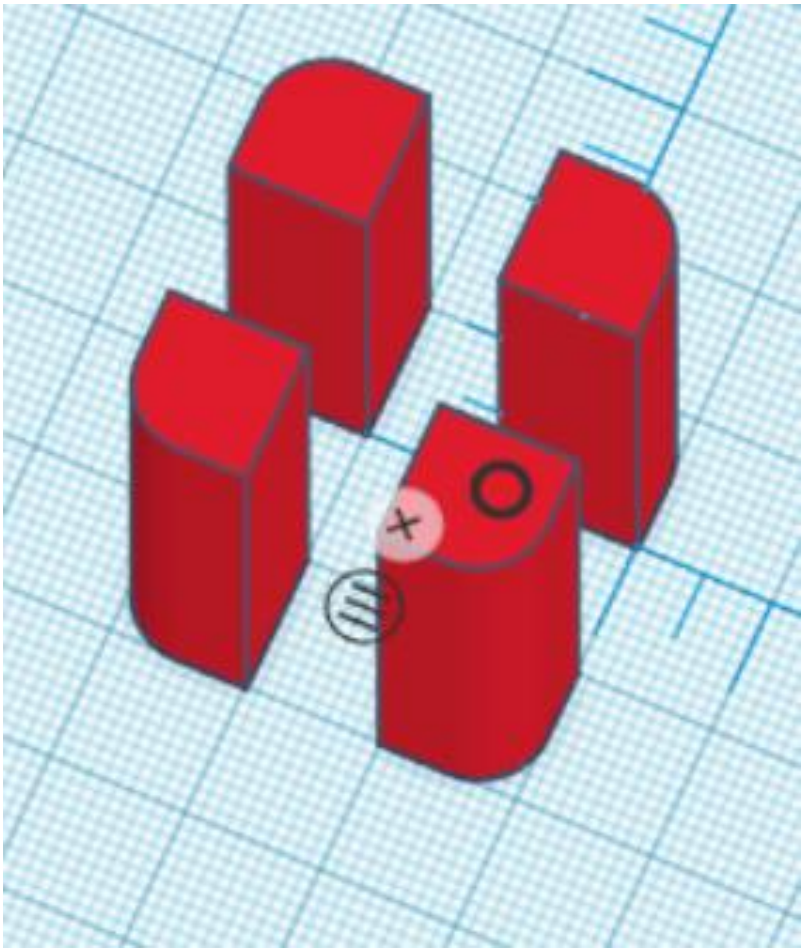
步驟五：畫一個 Metafillet (Shape Generators → All →Metafillet)，並設定長為 5mm、 闊為 5mm 及高為 40mm，然後將物件變成 Hole (灰色)



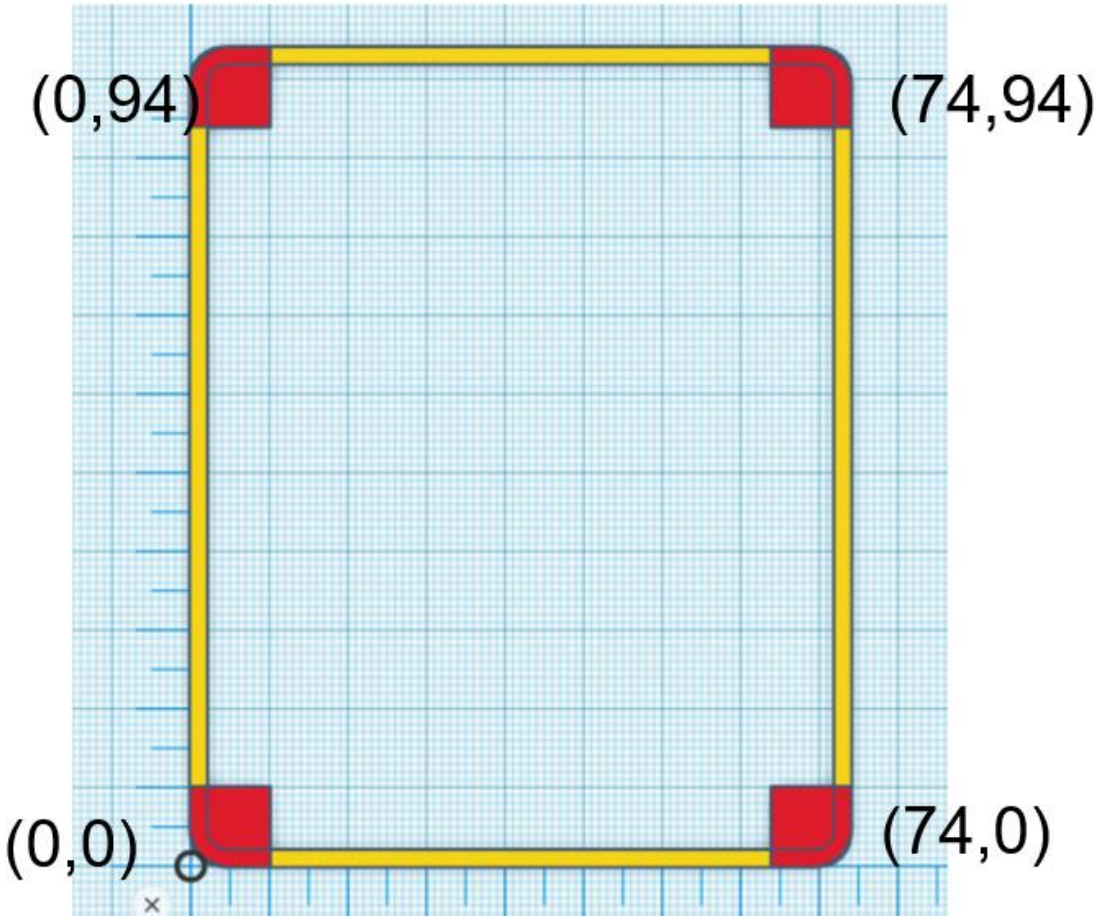
步驟六：將其中一隻角變成圓形



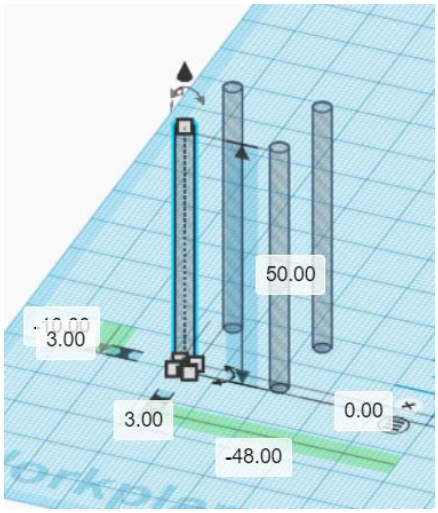
步驟七：複製多三個並旋轉至適當的方向

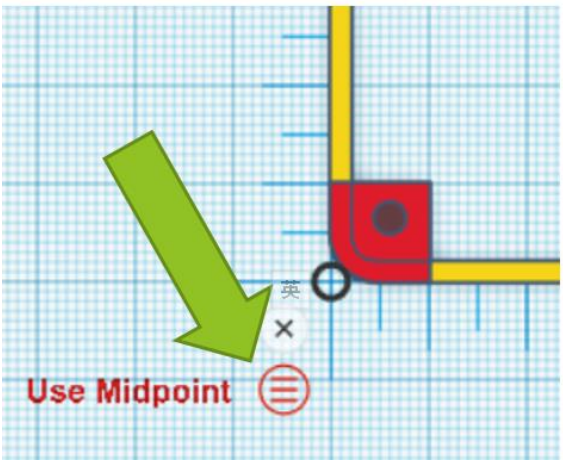


步驟八：放到適當的位置 (括弧中的數字為各個物件的 x 及 y 坐標)

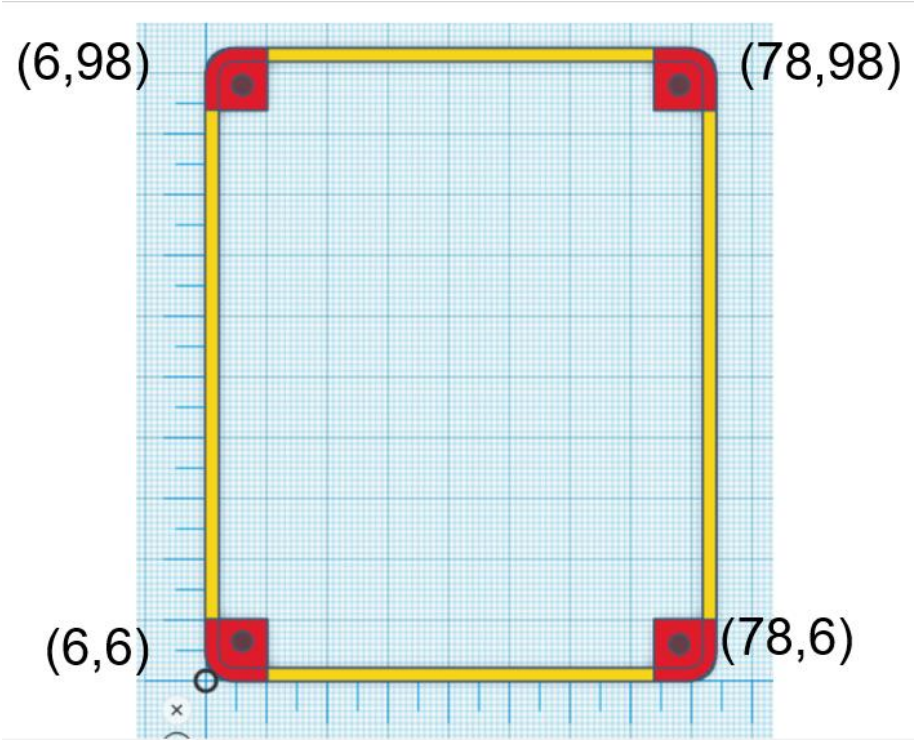


步驟九：畫出四條長 3mm、闊 3mm 及高 50mm 的圓柱體，並將物件變成 Hole (灰色)。按下原點附近的 [使用中點]，利用物件的終點作對位。

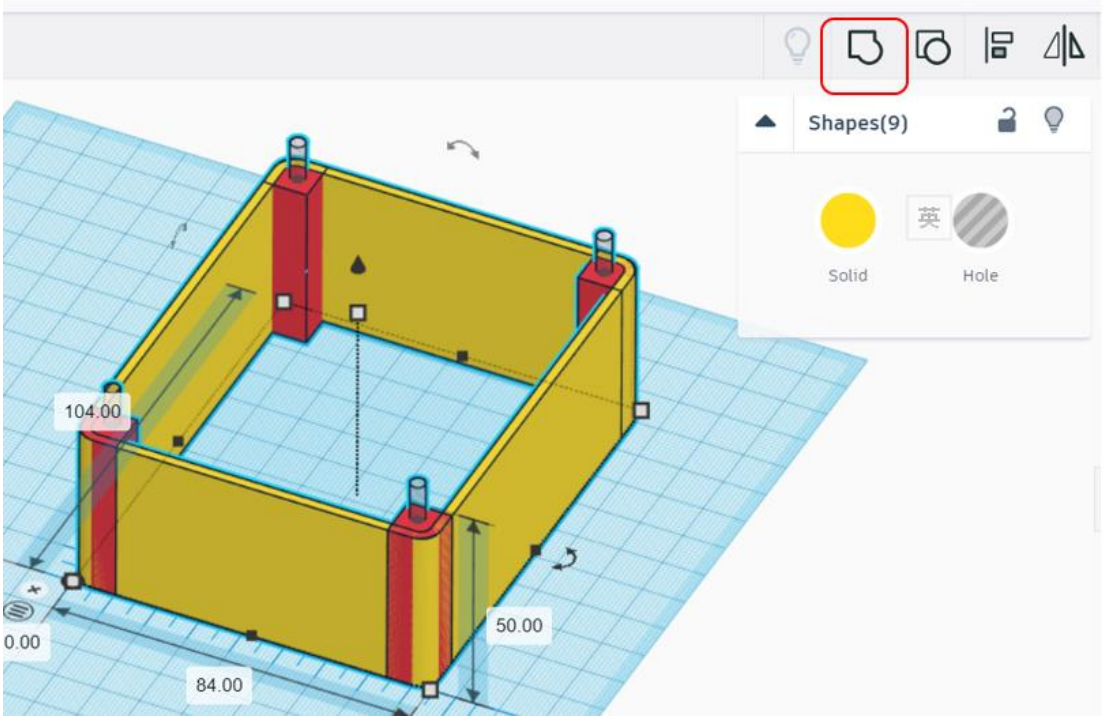


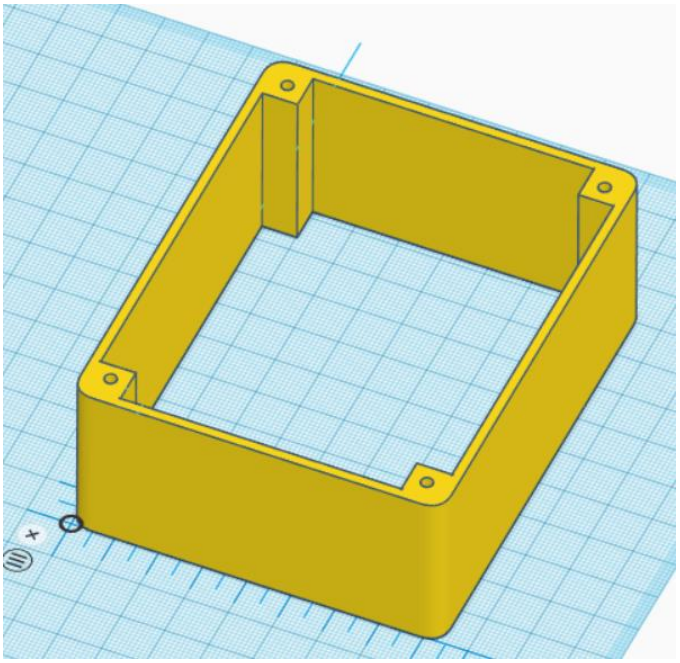


步驟十：放到適當的位置(括弧中的數字為各個物件的 x 及 y 坐標)

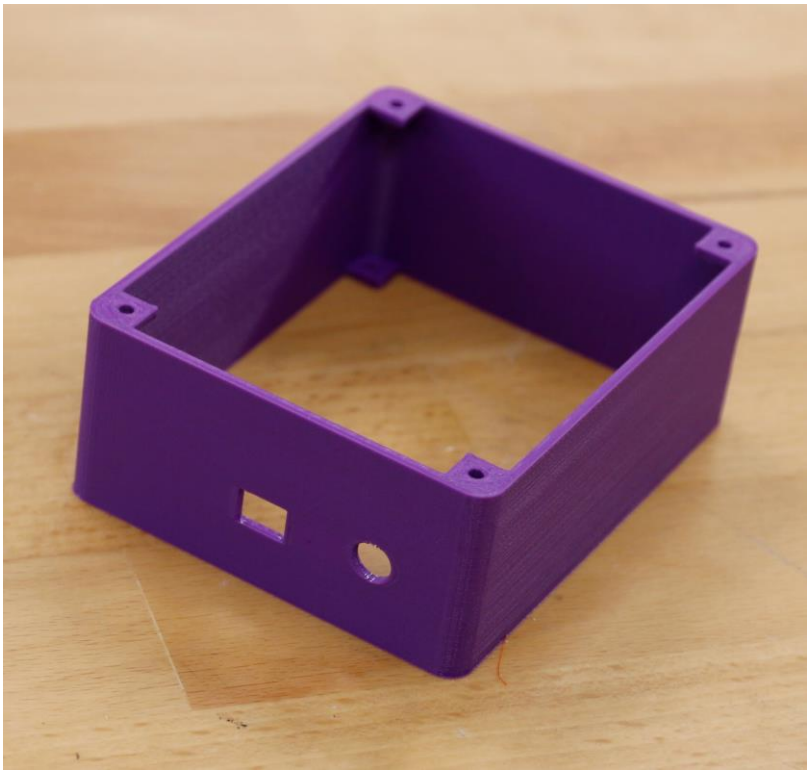


步驟十一：利用「組合」功能將圓柱的空間移去便可完成

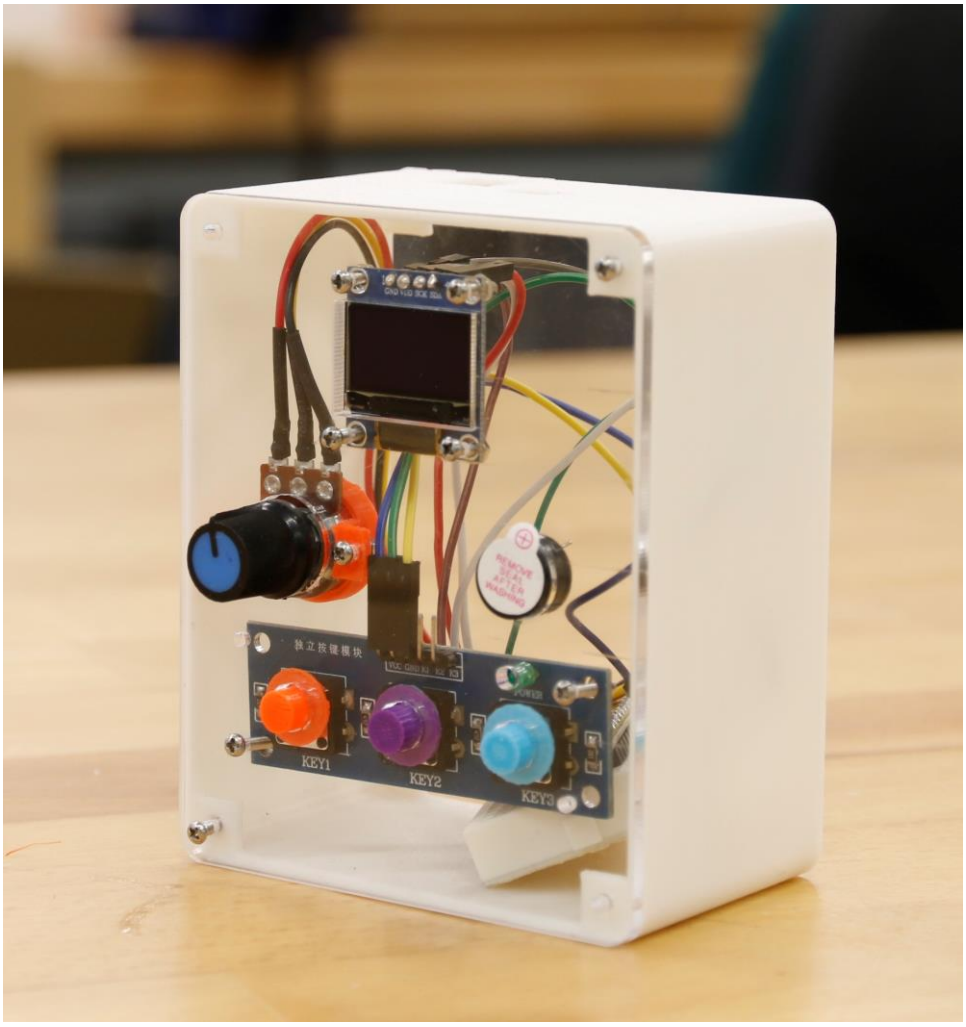




外殼側面完成圖



外殼側面實物圖



多功能迷你控制台製成品