

- WELCOME TO -

賽馬會科藝共融計劃

JC-Project-IDEA

Day01

上午

有關於物料選用



Arduino UNO



SD CARD



導電銅箔膠帶



導電墨水



麵包板



皺紋膠紙



耳機



USB SD卡讀卡器



USB Mirco-B to USB A 數據線



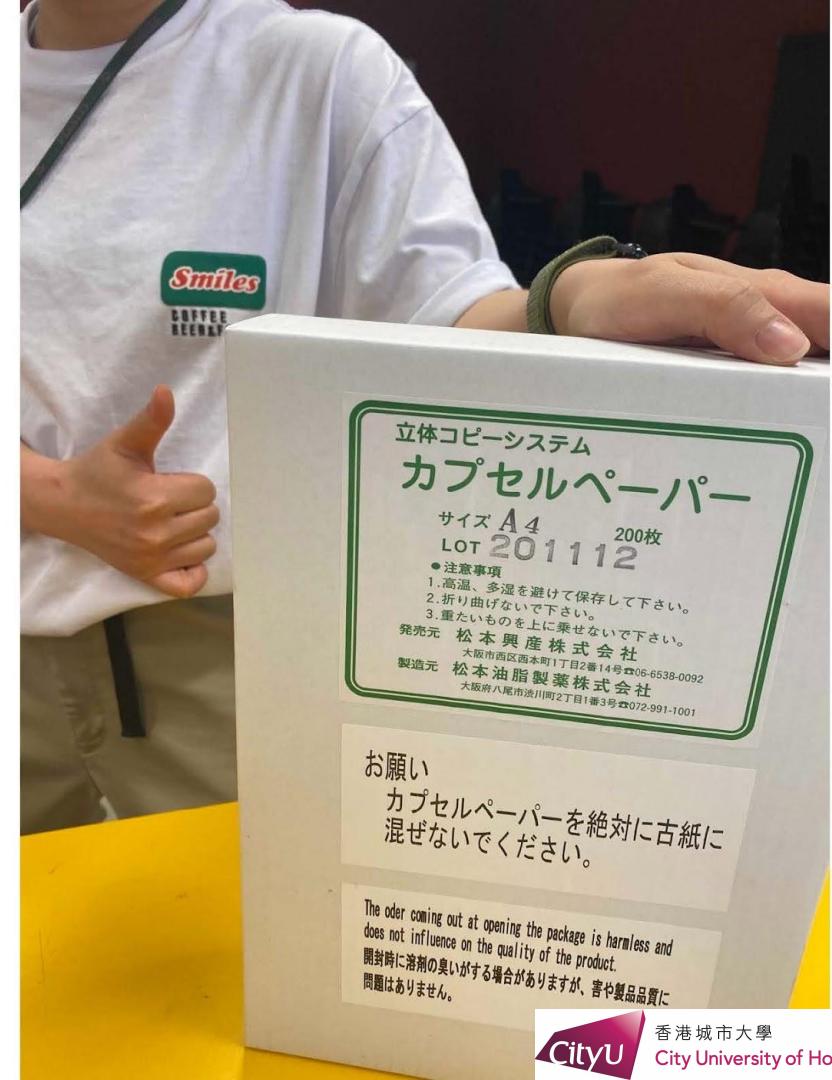
電夾



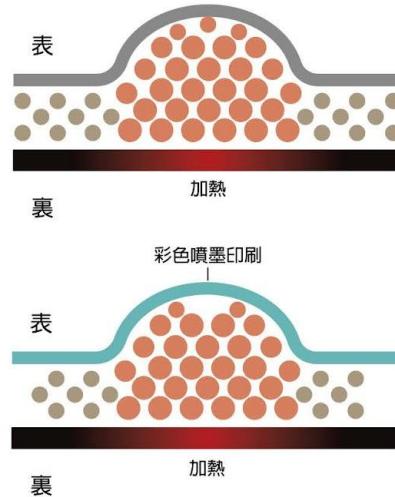
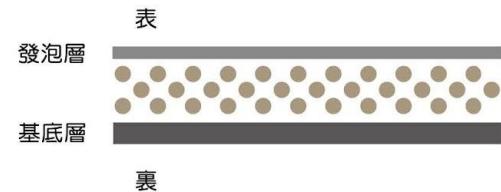
跳線

小知識

- 我們將會使用的紙張名叫「熱敏式專用紙」
專用紙張的表面有一層遇熱即會發泡的塗層。印刷時，熱敏印刷頭會從專用紙的背面(基底層)將印字的所在位置進行加熱，使發泡層膨脹，紙張的表面即製造出各種精細的立體線條及圖形。

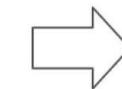
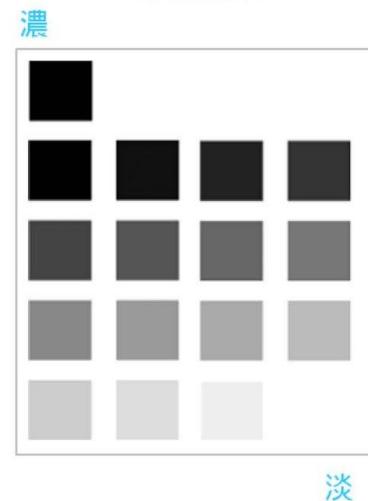


藉於發泡紙內面加熱，可印製出精細的立體圖形。



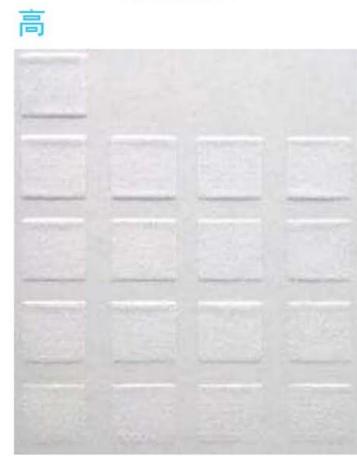
與彩色噴墨式印刷機搭配使用，
可呈現真實性更高、更美觀的圖形。
※ 彩色印刷機需另備

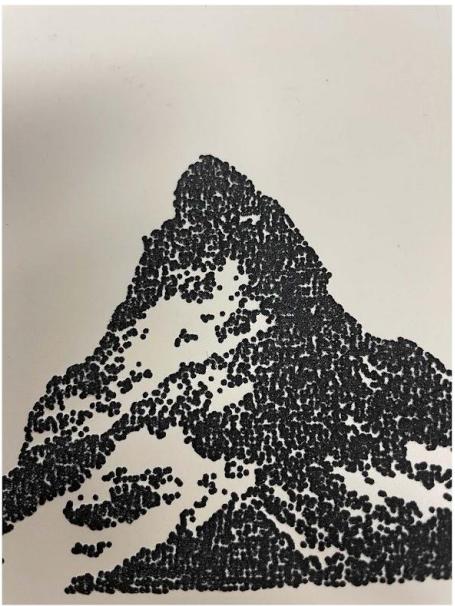
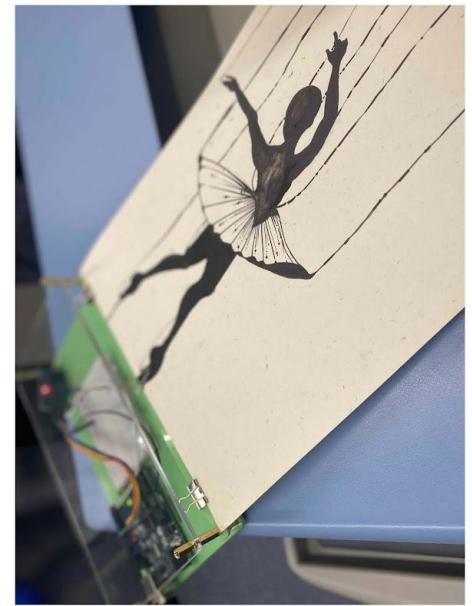
印刷檔案

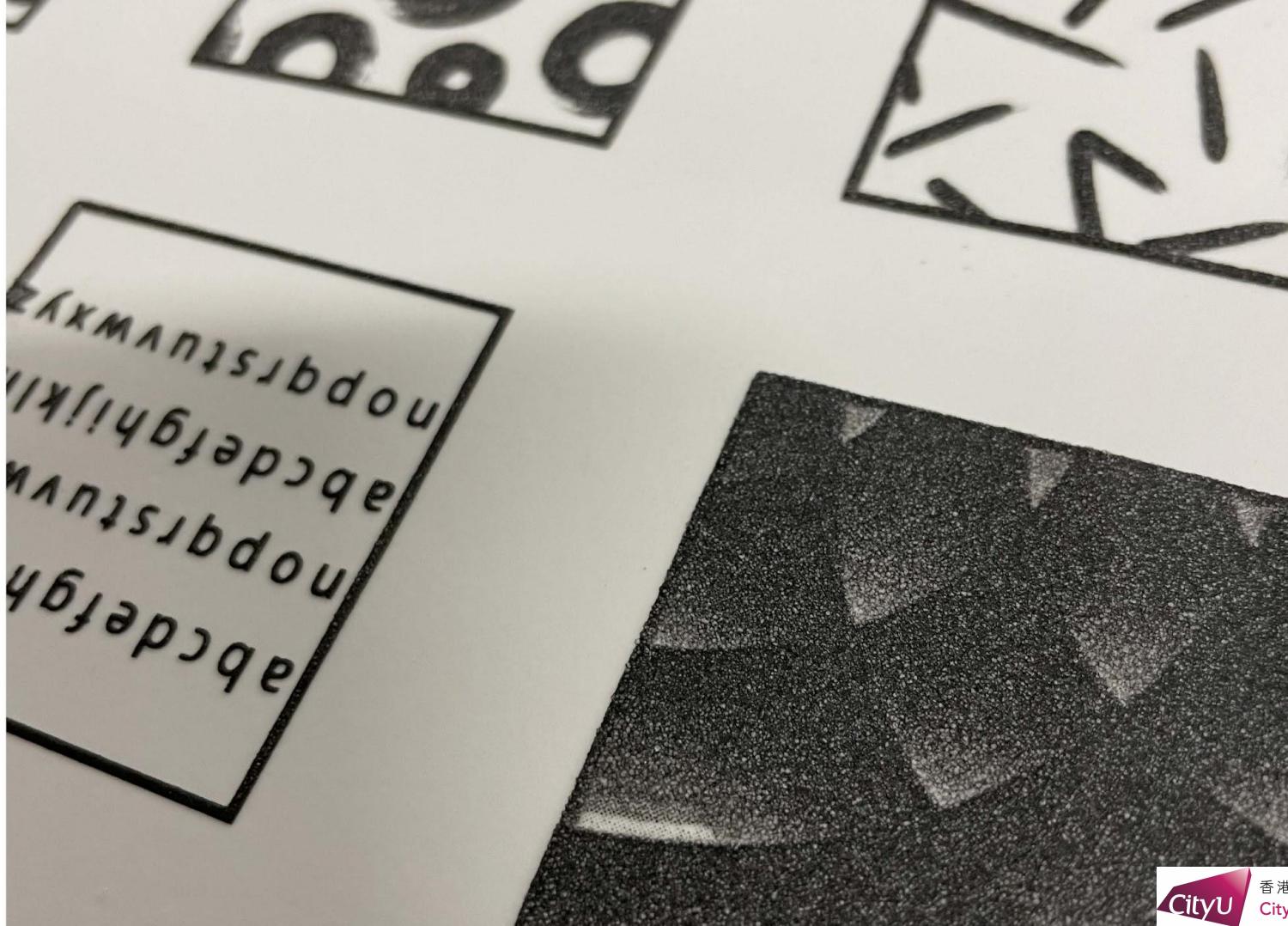


立體化

印刷結果

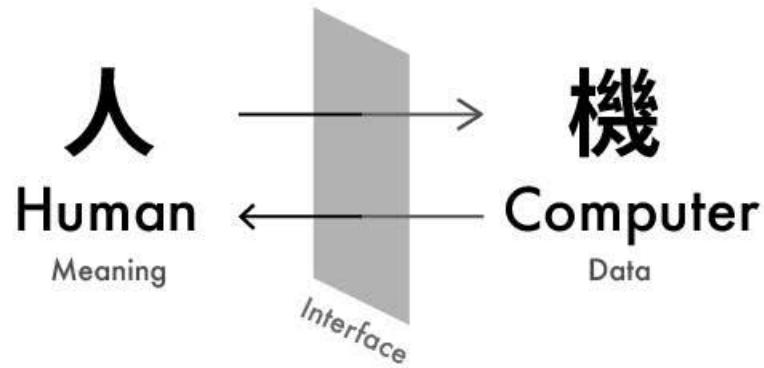






動手繪畫前先了解 ~

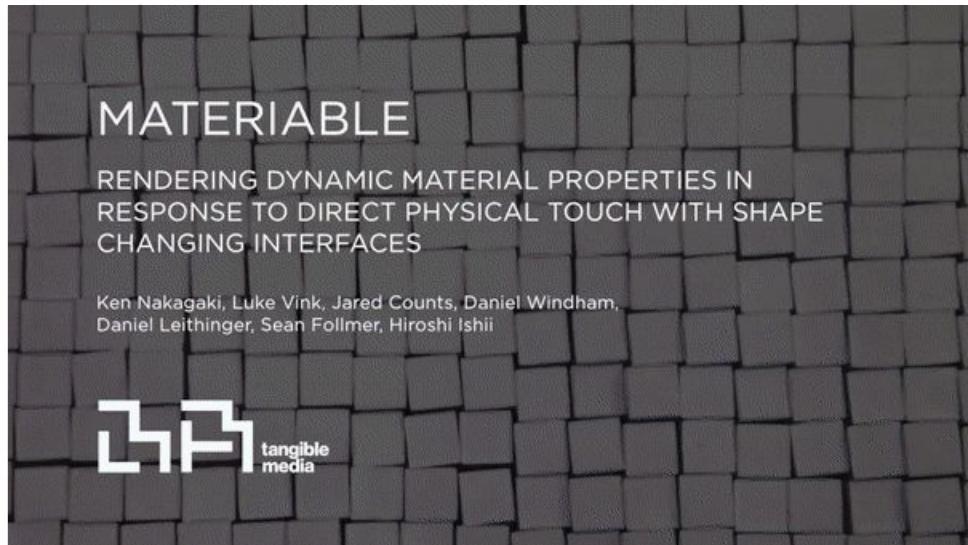
使用者體驗 (User Experience) 與
人機互動介面 (Human–Computer
Interface)



Materiable

美國 MIT Media Lab 的 Tangible Media 實體媒介實驗室，嘗試各種材料與方法，超越平面的螢幕，讓人們可以接觸實體三維的物件，就像是立體的顯示器，直接操作其代表的數位資訊。

https://vimeo.com/165798784?embedd ed=true&source=video_title&owner=11720996



MATERIABLE

RENDERING DYNAMIC MATERIAL PROPERTIES IN
RESPONSE TO DIRECT PHYSICAL TOUCH WITH SHAPE
CHANGING INTERFACES

Ken Nakagaki, Luke Vink, Jared Counts, Daniel Windham,
Daniel Leithinger, Sean Follmer, Hiroshi Ishii



tangible
media



香港城市大學
City University of Hong Kong

Please Touch!

這是一個包容性的藝術界面，挑戰觀眾與藝術品的傳統關係。它打破禁忌，邀請觀眾觸摸藝術品。觀眾參與其中，通過數字製作的觸感浮雕、計算機視覺和多媒體剪輯，探索藝術家的世界。

<https://www.youtube.com/watch?v=uDP10ZNTfaM>





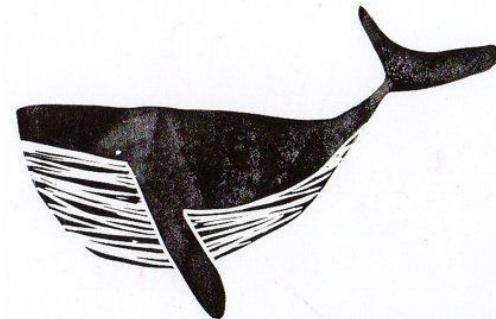
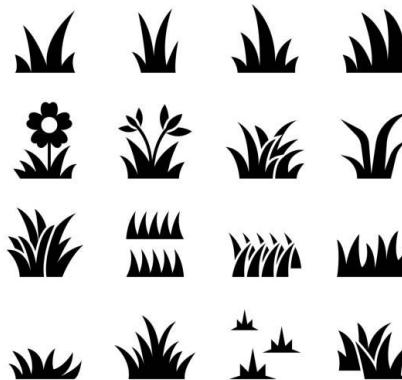
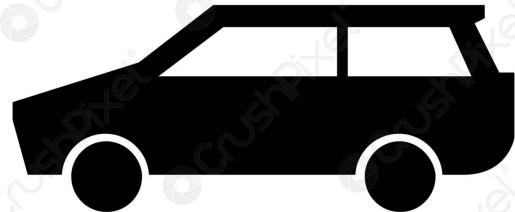
有關裝置結構

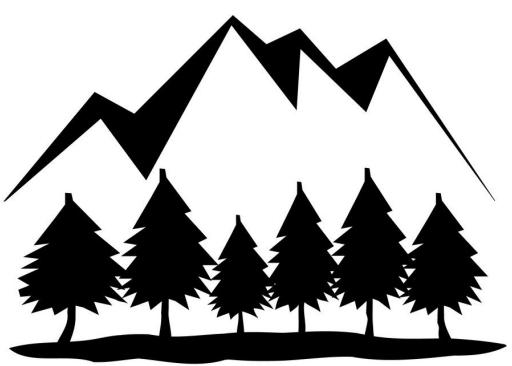
先完成畫面部份

繪畫時候都要顧及以後會加入的聲音元素 !!
以及經過加熱後的質感 !!

構圖考慮 : 1. 內容主體容易分別，以面積形狀區分
2. 簡單構圖，突顯內容主體特徵
3. 觸摸時的次序
4. 後加聲音元素 (如表達火車，可以畫火車軌表達)

圖案、形狀與主體特徵





網上參考圖關聯詞彙：

Lino print, Icon, Symbol + 主體詞彙





畫面



電路

完成畫面後，建議老師可以把聲音內容先標記好。

城市聲

歡呼聲

交通工具聲

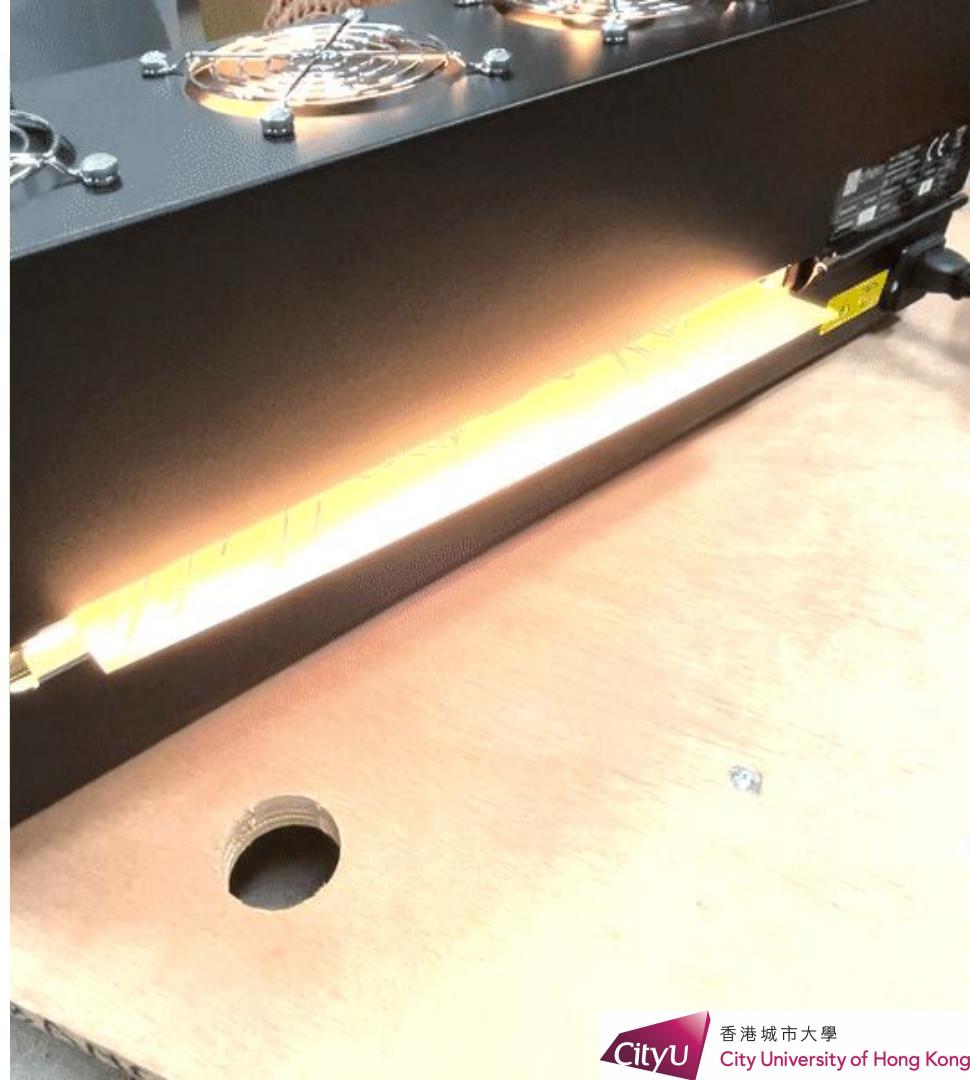
歌聲

小鼓聲



**完成畫面後，請老師把
作品交給助理。**

我們會先用印刷機把作品複製，下午
再放到加熱機器。



上午部份完成

下午部份會有關

- 使用導電油墨製作電路
- 聲音主題 擬聲音效 (Foley) 及 現場錄音 (Field Recording)

需要準備任何可發聲的日常用品

例如：保鮮紙、紙杯、各種文具

- WELCOME TO -

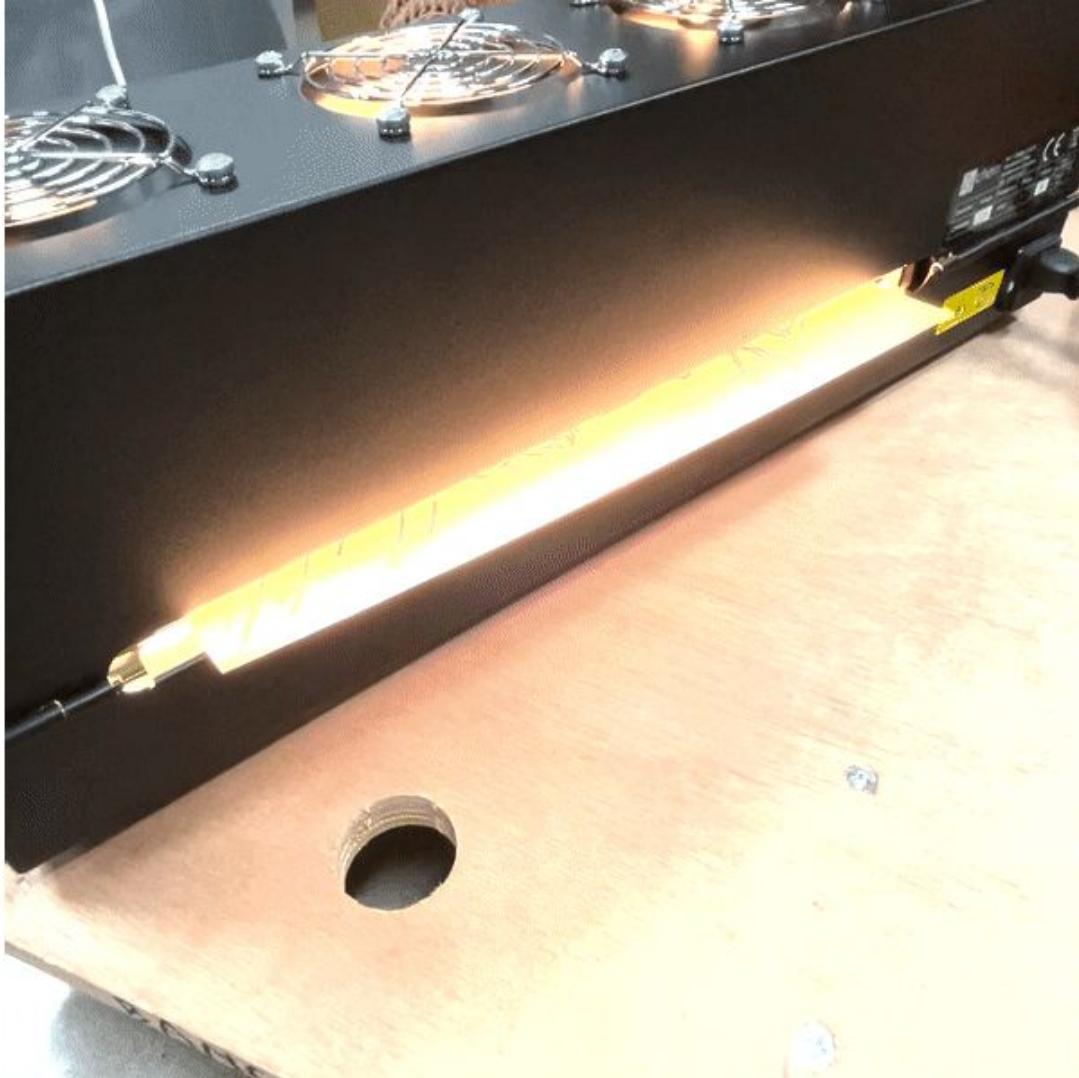
賽馬會科藝共融計劃

JC-Project-IDEA

Day01

下午

準備加熱!!!!

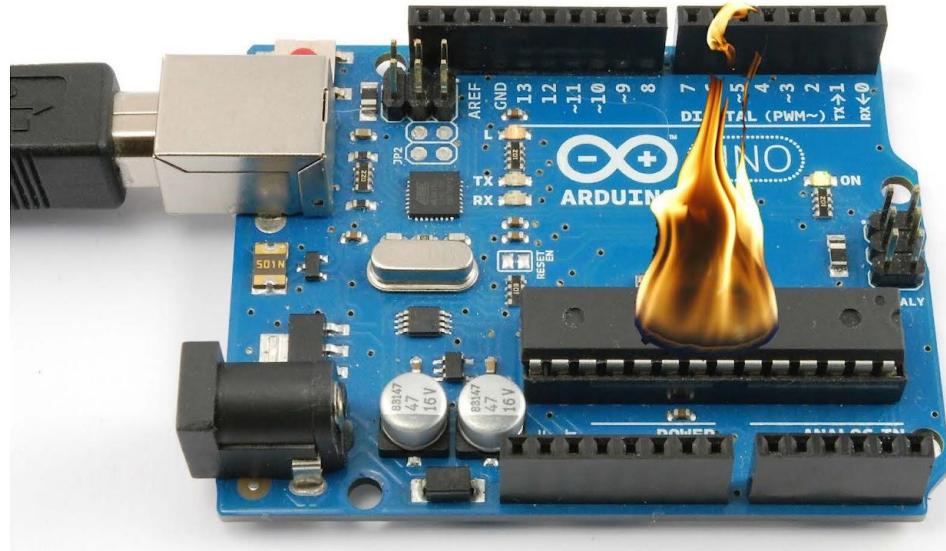


簡介 - Conductive Ink (導電油墨)

導電油墨是一種具有導電性能的特殊墨水，通常由導電顆粒（如銀、碳）和樹脂基質組成。它可用於創建彈性、可印刷的電子電路，擴展應用領域。

在開始前，先補充基本知識。

以免



Circuit 電路

電路是由電子元件組成的路徑，讓電流能夠流通，用於控制、處理、傳輸和存儲電信號和電能。是電子技術中的基礎。

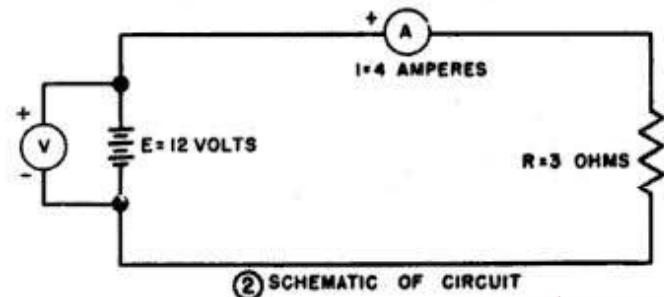
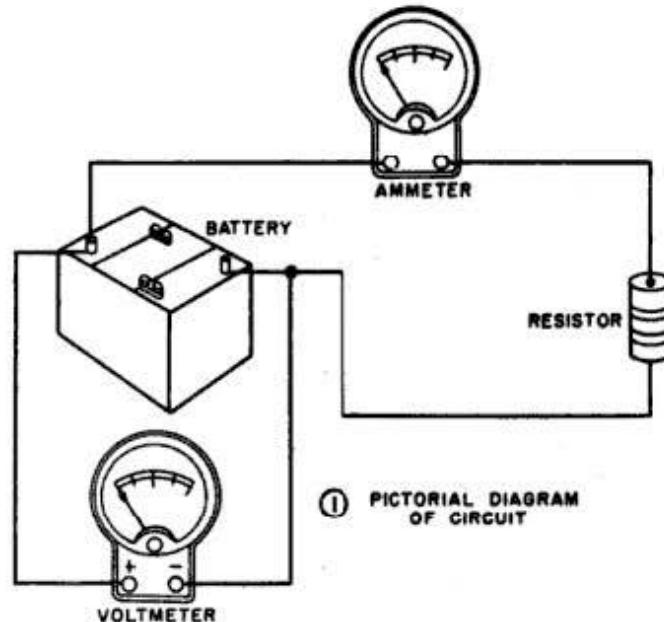
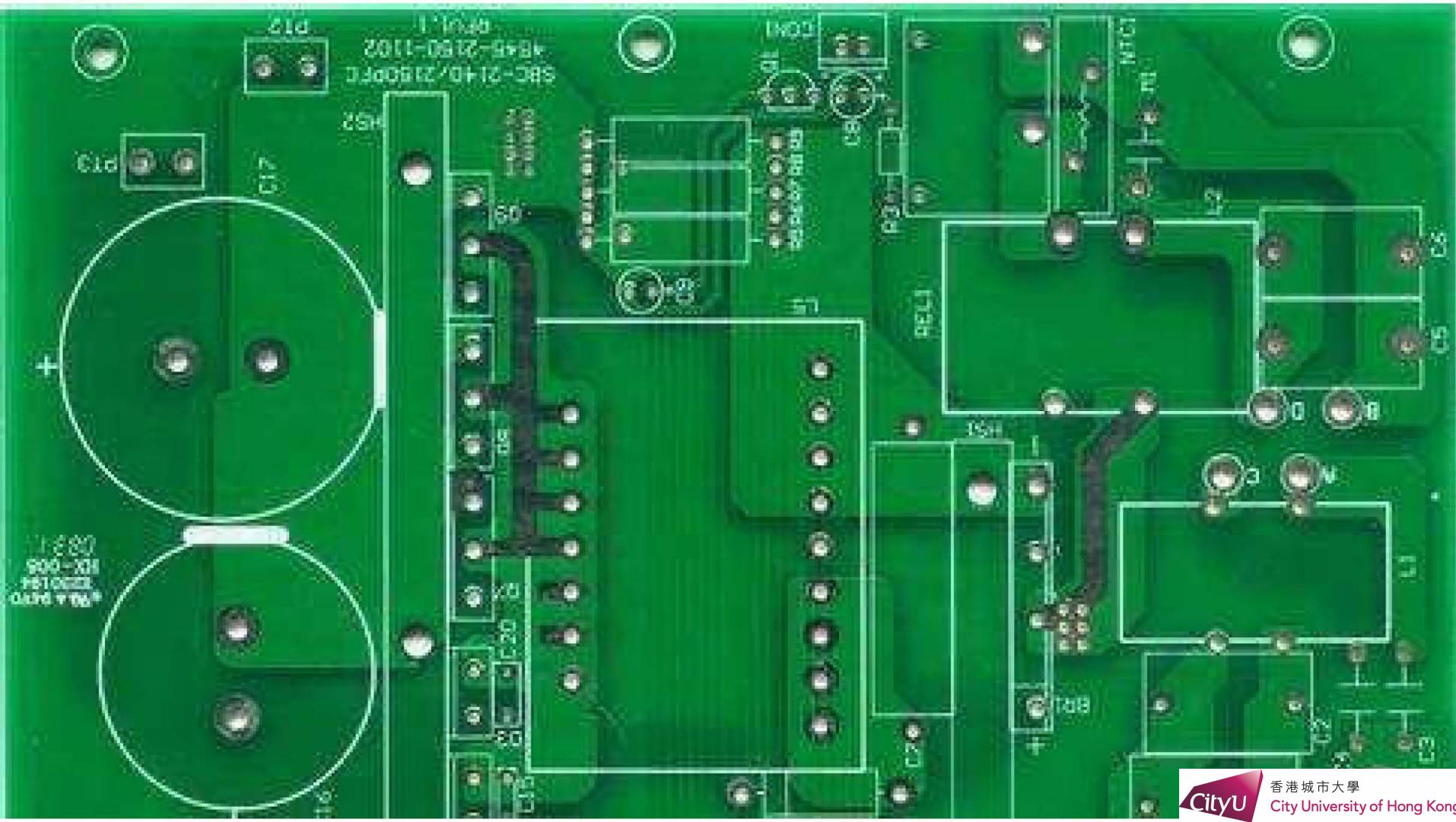


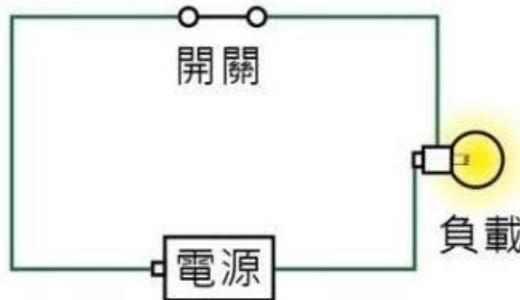
Figure 48. Diagram of a basic circuit



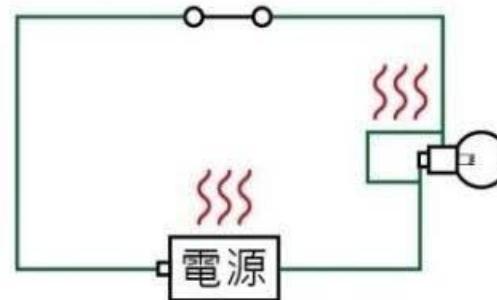


你所不知道的秘密

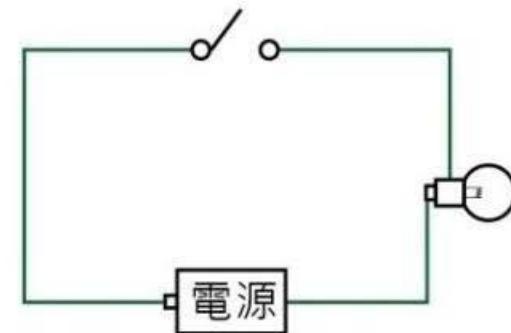
通路就是按下開關(on)，線路上通電，電器就能運作；短路就是在電路中直接讓正、負極接觸，產生的電流極高，會發熱甚至著火；斷路是電路呈斷開狀態(off)，此時電力系統無法構成迴路不能通電，電器就無法運作。



通路



短路



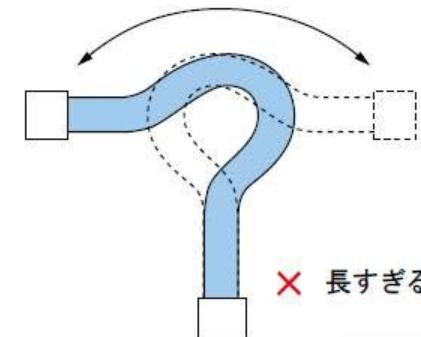
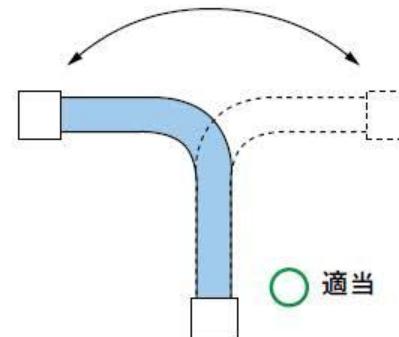
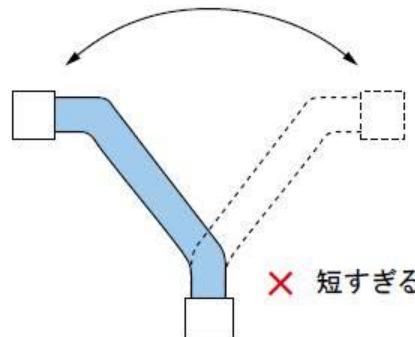
斷路

在接下來需要留意的事項：

- 插電前檢查電路是否已經接通
- 小心正負極接反導致電子元件損壞



- 不要過度屈曲電線



先在另一張紙上做好電路分佈計劃

城市聲

歡呼聲

交通工具聲

歌聲

小鼓聲



Now, Draw the Circuit !

現在檢查電路

現在工作人員會幫大家簡單測試一下電路運作

為明日課堂準備聲音素材

城市聲

歡呼聲

交通工具聲

歌聲

小鼓聲



freesound.org

The screenshot shows the homepage of freesound.org. At the top, there's a navigation bar with links for Home, Sounds, Forums, People, and Help. A search bar is prominently displayed. Below the navigation, a "Random sound of the day" is shown: "RETRO LASER SHOT_02.wav" by Matix. The description says it's a retro laser shot sound used in games. To the right of the sound preview, there are links for "More Sounds" (Browse latest comments, Browse tags, Browse geotags, Browse packs, Browse remix-groups, Give me a random sound!). Below this, a "Latest sounds" section features a recording of lions roaring, uploaded by eardeer. The description includes details like the recording location (Ukutula Game Reserve, South Africa) and equipment used (Rode NT1A stereo pair). There's also a link to more sounds from the same user.

網上聲音資源

<https://freesound.org/>

聲音圖書館 by soundpocket

The screenshot shows the homepage of "The Library by soundpocket". The header features the title "THE LIBRARY" and "by soundpocket 聲音圖書館". Below the header, there's a call to action: "sound out your support · click to make a sound donation 賽援我們 按此作聲音捐獻". The main content area is divided into several sections: "Project info 本館簡介", "Listen 來聽" (with a play button icon), "editorial 編輯室", "sound scoop 細聲公", and "meanderings 隅尺". The background has a large, stylized yellow shape resembling a book or a wave.

<https://www.thelibrarybysoundpocket.org.hk/>

小森平的免費下載音效

生活上的動作1

敲·門·門口 / 浴室·洗臉池·廁所 / 滾轉·壞 /
揮 / 工具 / 掃除

生活上的動作2

文具 / 道具 / 傘 / 擦 / 房間中的這個那個地

放置·舉起

雜物 / 餐具 / 放置·舉起

家用設備

電氣化產品·電燈·電話·手機 / PC·外部設備
·照相機·衛生·掃除

烹飪·廚房

烹飪 / 烘烤·煎炸·燉煮 / 他 / 蛋 / 廚房·櫥·
桌子 / 打掃乾淨

飯菜·吃飯

食物 / 喝 / 在食用前 / 飲料

遊戲·按鈕

按鈕 / 正確的解答·廢除 / 掠硬幣 / 跳躍 / 能
量提升·減少 / 貓按鈕

遊戲·動畫片

場面轉換·開始·結束 / 噟頭·動畫片風格(異怪
的聲音)

魔術·幻想

魔術 / 幻想

SF·空想機械

SF·空想機械 / 機器人·大型機械 / 宇宙

武俠·古裝劇

切的·刺 / 劍·刀 / 勇士的吶喊 / 宗教·禮儀 /
各種

格鬥·打擊

打的·踢 / 損壞 / 風切·空揮 / 科幻小說 / 他



樂器·玩具·桌面遊戲

樂器 / 桌面遊戲

活動·祭祀·宴會

演奏會 / 宴會 / 祭祀·活動 / 煙火 / 職業棒球 / 相撲 / 遊樂
園

http://taira-komori.jpn.org/freesoundtw.html

小森平的免費下載音效

[https://taira-komori.jpn.org/freesoundtw.html](http://taira-komori.jpn.org/freesoundtw.html)

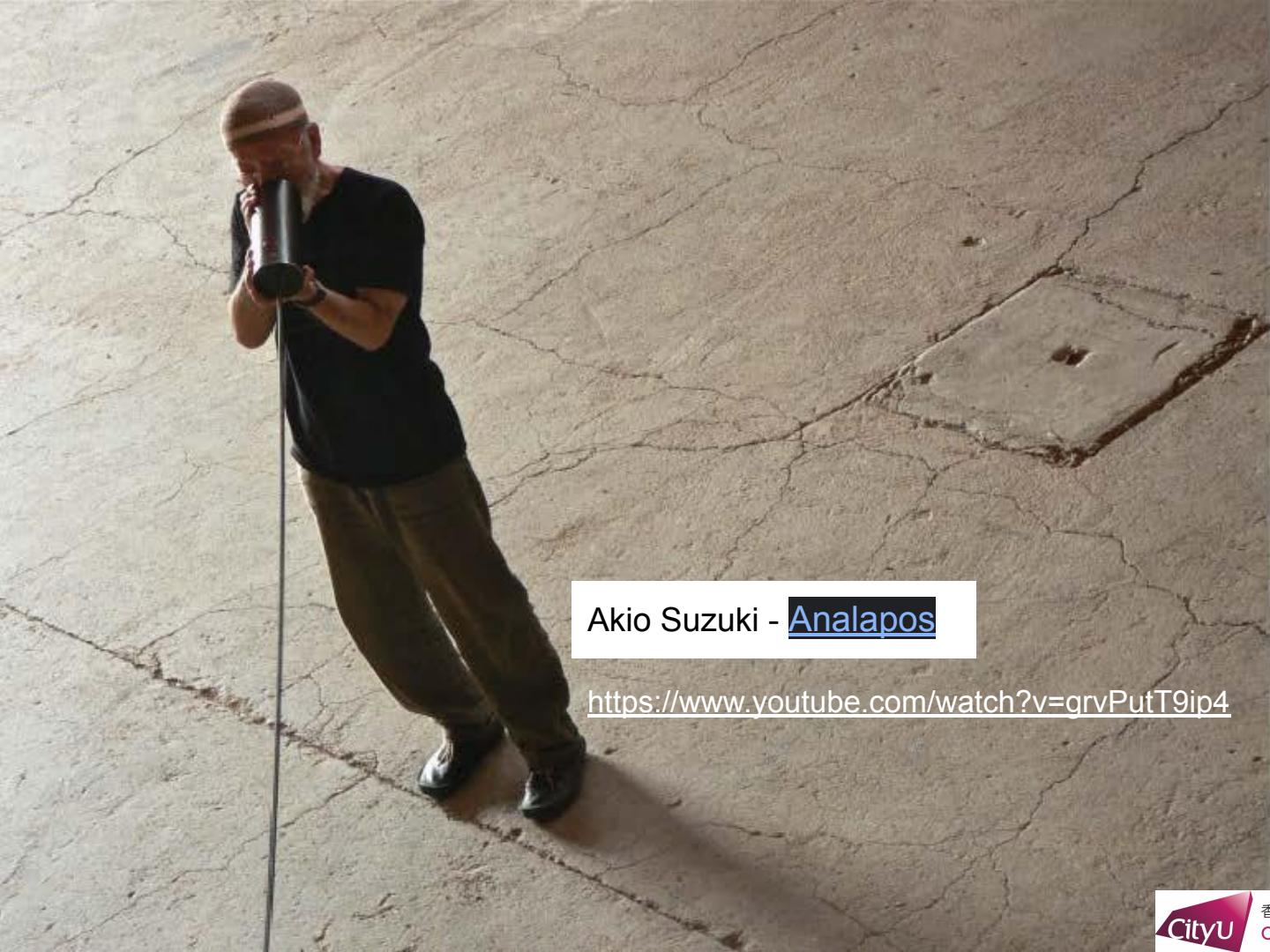
除了於 網上下載聲音資料 或 在城市錄音之外

還有甚麼方法取得聲音？

Live Demo Section

Circuit, Sonic, Effect, Sound





Akio Suzuki - [Analapos](#)

<https://www.youtube.com/watch?v=grvPutT9ip4>

甚麼是 擬聲音效 (Foley) ?

擬聲音效 (Foley) 是指使用人聲或物品模擬出特定聲音，用於影片、遊戲、廣告等多媒體製作，營造更真實的聲音效果，增加觀眾的身臨其境感受。

https://www.youtube.com/watch?v=_6tkhCNv9Ek







Foley Artist 的工具

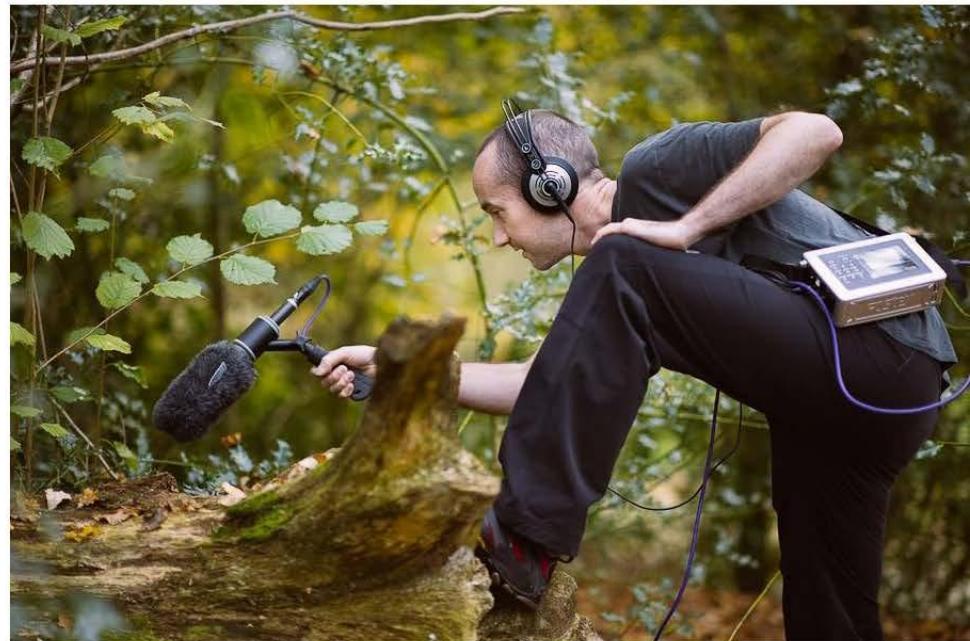


Foley Artist 的工具

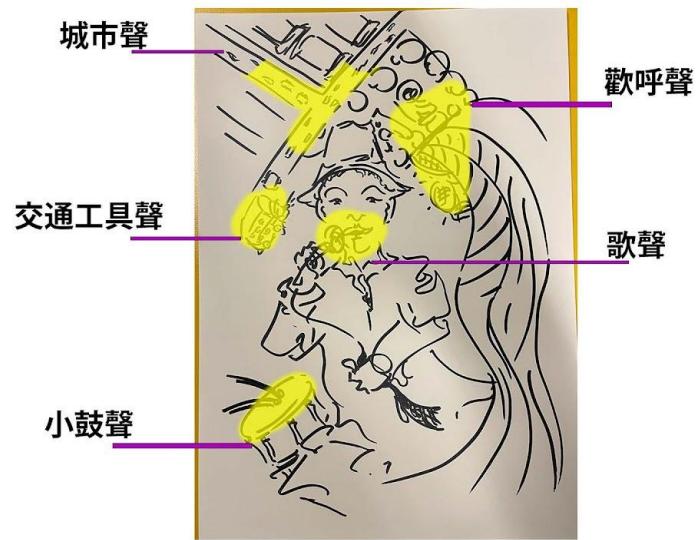
甚麼是 現場錄音 (Field Recording) ?

現場錄音 (Field Recording) 是指在戶外或室內環境中使用錄音設備進行錄音，捕捉自然或人造聲音，用於音樂、電影、電視、遊戲等製作，增加真實感和氛圍。

<https://www.youtube.com/watch?v=G0nhh6Jkb1U>



現在為你的作品創作聲音



集齊聲音檔案後，我們可以做一些後製～

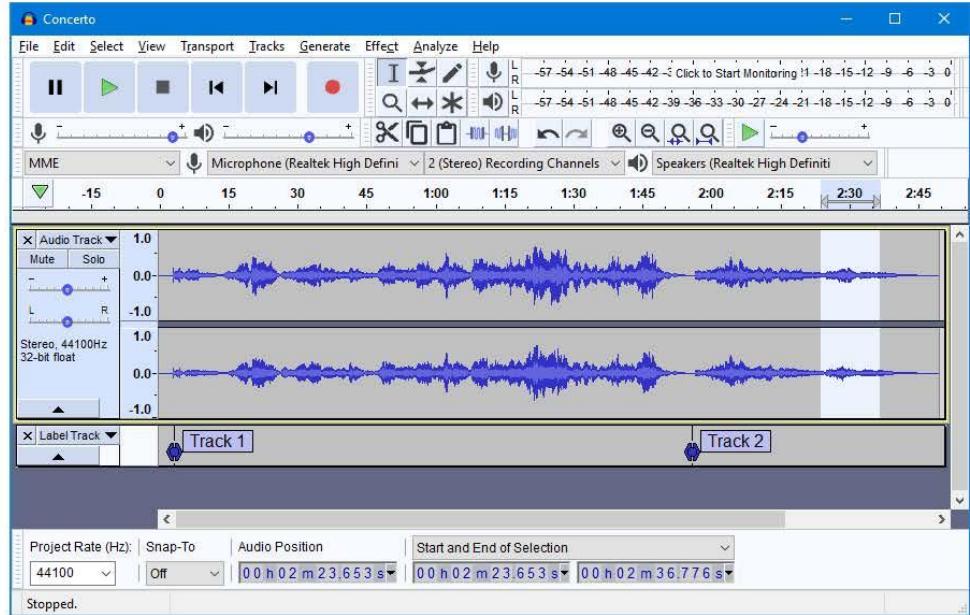
甚麼是 數位音訊工作站 (Digital Audio Workstation DAW) ?

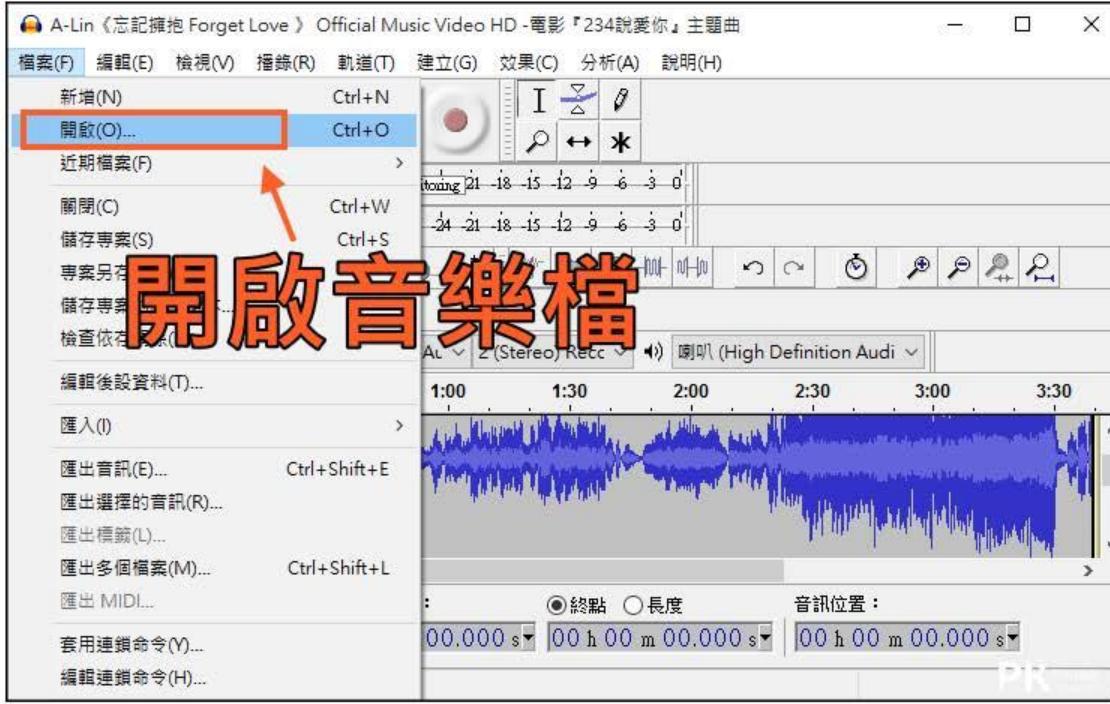
音訊工作站是一種專業數位音頻製作軟體，可用於錄音、混音、編曲、編輯及後製處理等音樂製作工作，適用於聲音藝術家、音樂製作人、錄音師、唱片製作人等。

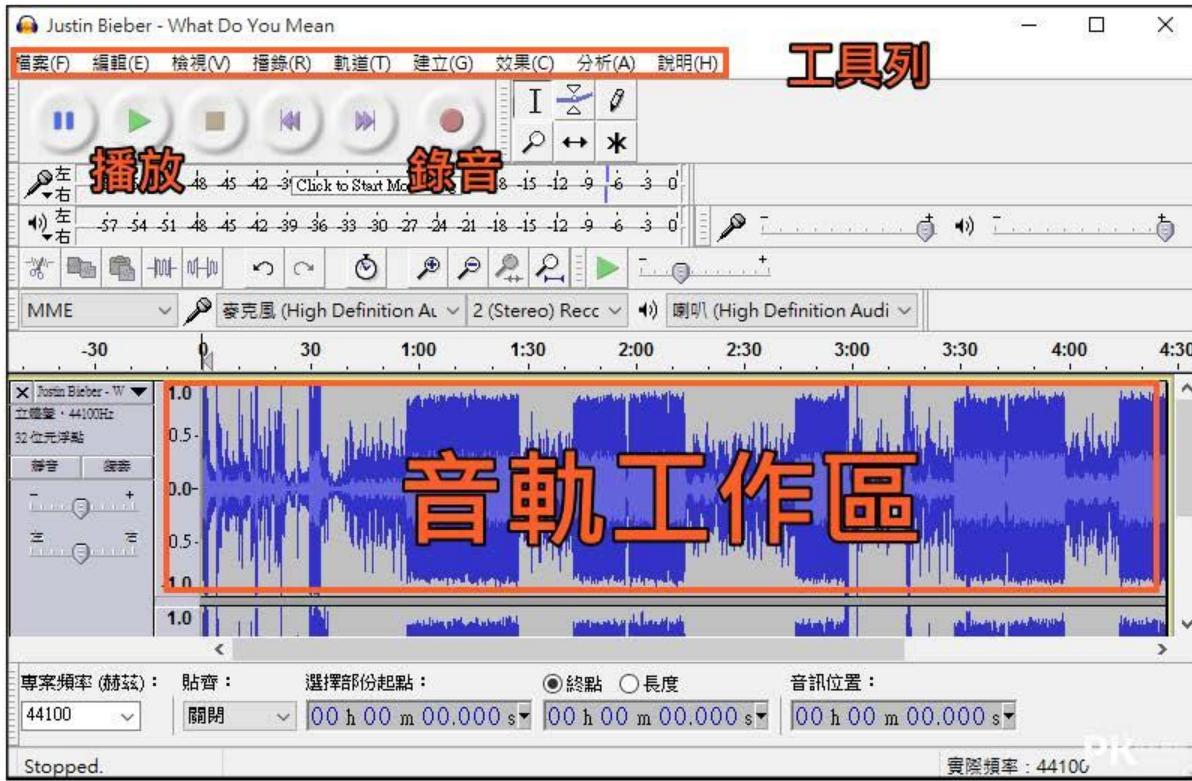


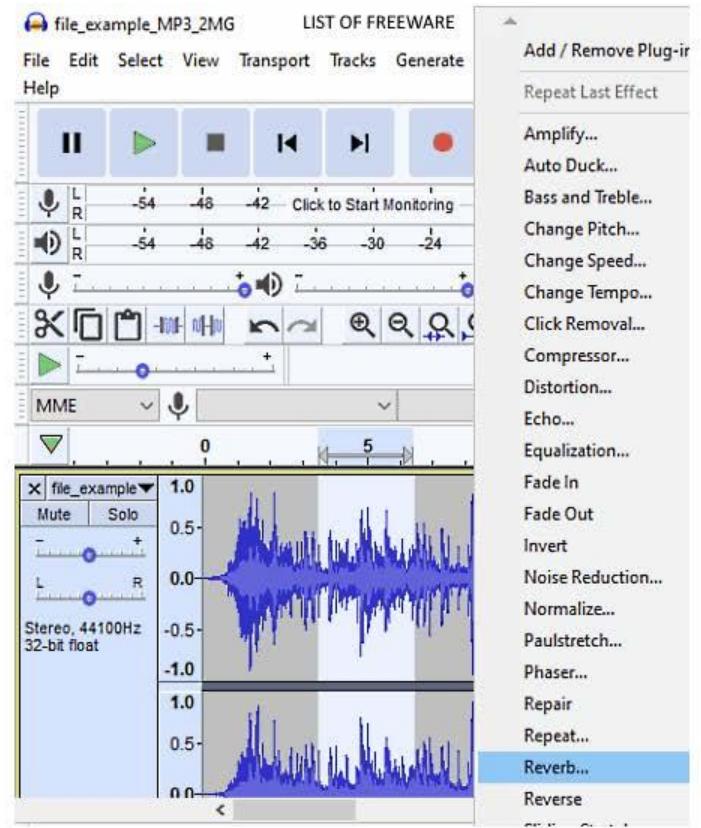
我們會使用開源軟件

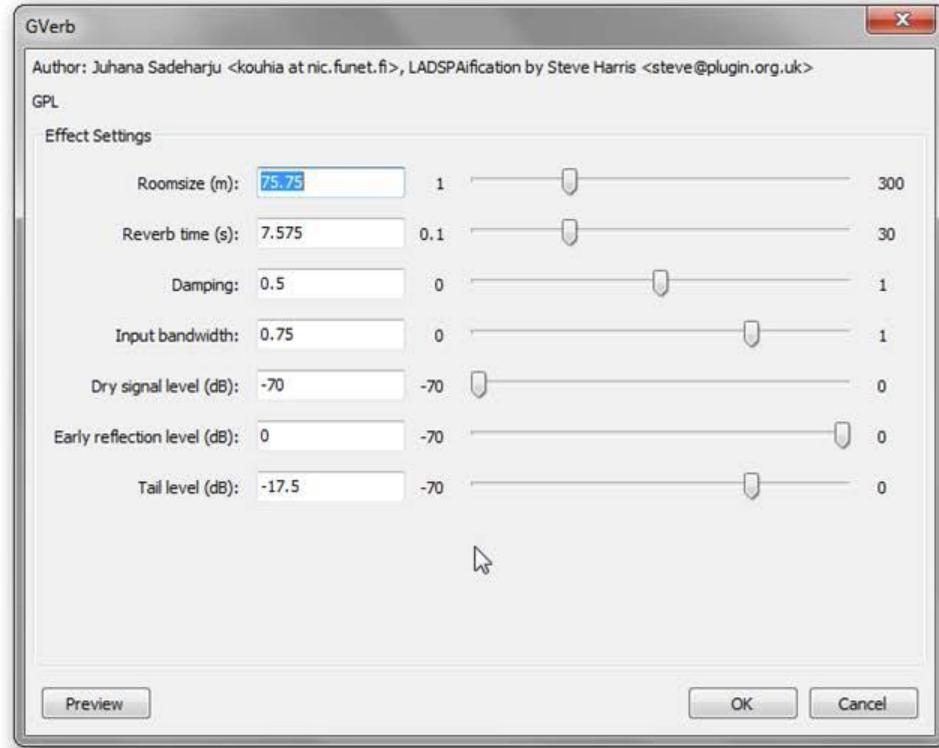
Audacity

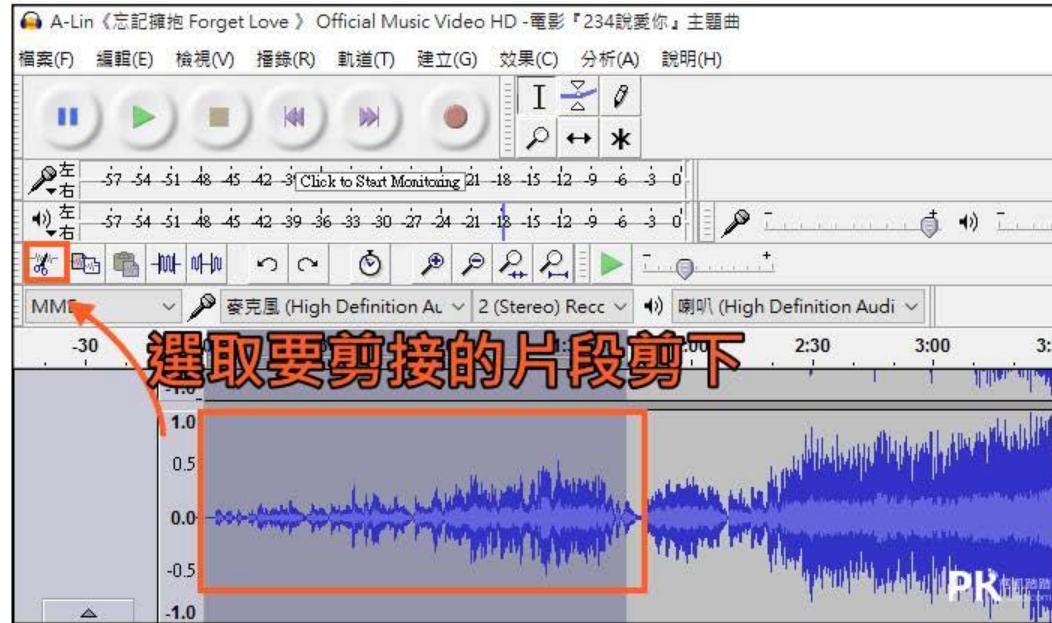














另存成MP3等音樂檔

	Name	Date modified	Type	Size
1	TRACK000	4/29/2023 3:16 PM	MP3 File	59 KB
2	TRACK001	4/29/2023 3:16 PM	MP3 File	77 KB
3	TRACK002	1/6/2023 8:15 AM	MP3 File	48 KB
4	TRACK003	1/6/2023 8:14 AM	MP3 File	74 KB

把聲音檔案依次序修改名稱

TRACK000.mp3、TRACK001.mp3、TRACK002.mp3，以此類推。

下午部份完成

明日課堂部份

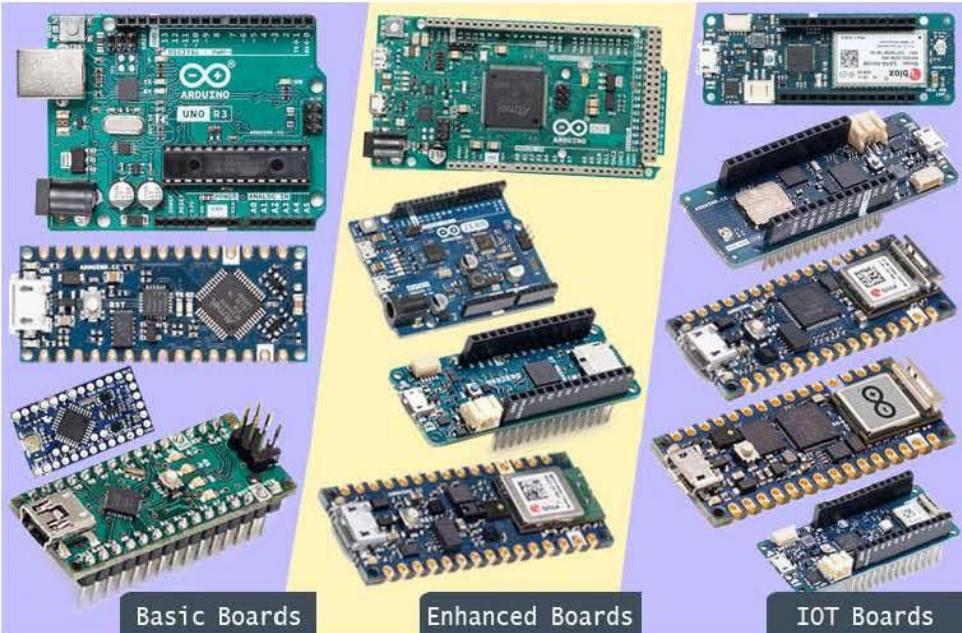
- 基礎編程、更多有關聲音藝術的例子

Q&A

附加

甚麼是 Arduino?

Arduino 是一個開源電子原型平台，允許用戶輕鬆創建互動式電子項目。它基於用戶友好的硬件和軟件，為創客、工程師和學生提供了一個靈活且易於使用的開發環境，促進創新和學習。源於2005年，Arduino 由一個意大利團隊開發，旨在為非專業人士提供低成本、易於使用的工具。

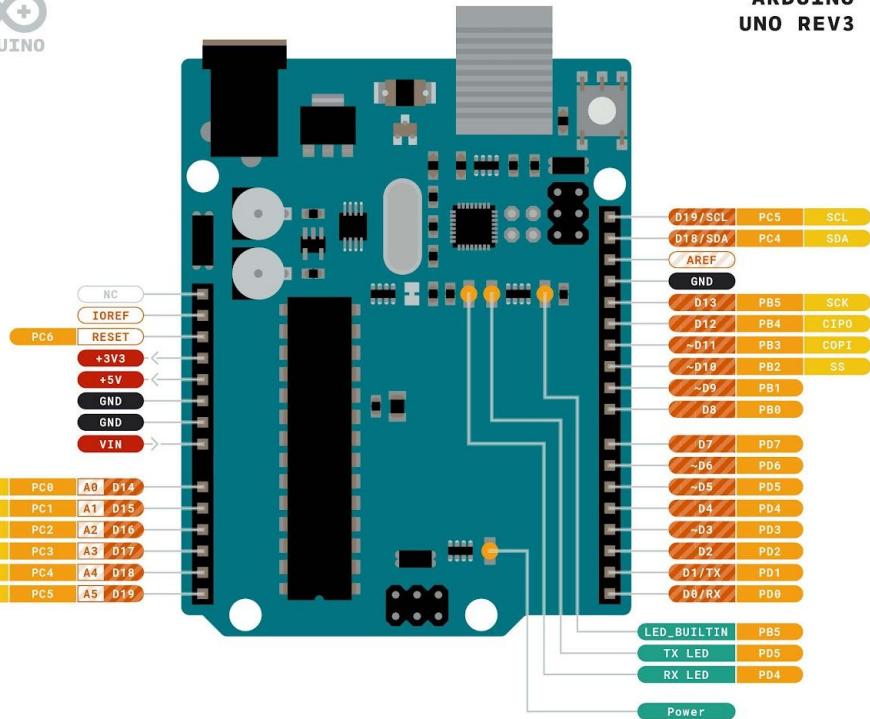




ARDUINO
UNO REV3

Basic Pinout Diagram

https://content.arduino.cc/assets/Pinout-UNOrev3_latest.pdf?_gl=1*jm7fcz*_ga*MTQ2MTMwNzk2OC4xNjgyMjc0MjYx*_ga_NEXN8H46L5*MTY4MjI3NDI2MC4xLjAuMTY4MjI3NDI2MC4wLjAuMA..



ARDUINO.CC

■ Ground ■ Internal Pin ■ Digital Pin ■ Microcontroller's Port
■ Power ■ SWD Pin ■ Analog Pin ■ Default
■ LED ■ Other Pin



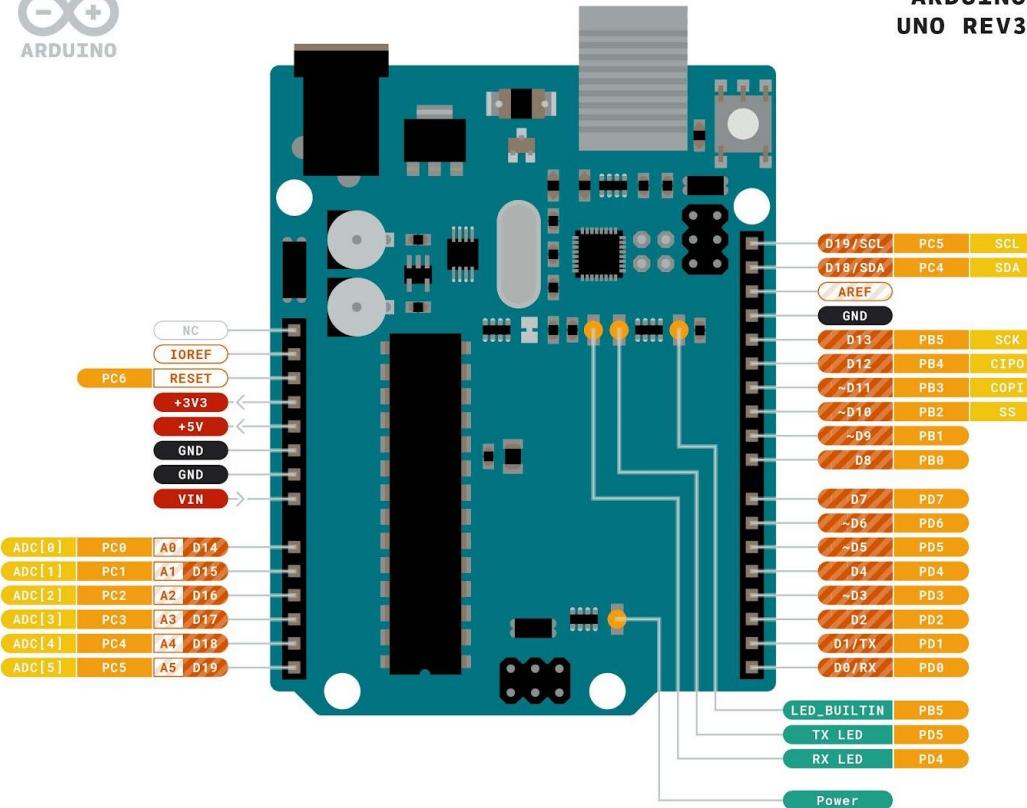
This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1822, Mountain View, CA 94032, USA.



香港城市大學
City University of Hong Kong



ARDUINO
UNO REV3



- Ground
- Power
- LED
- Internal Pin
- SWD Pin
- Other Pin
- Digital Pin
- Analog Pin
- Microcontroller's Port
- Default



This work is licensed under the Creative Commons
Attribution-ShareAlike 4.0 International License. To view
a copy of this license, visit creativecommons.org/licenses/by-sa/4.0/ or send a letter to Creative
Commons, PO Box 1866, Mountain View, CA 94042, USA.

安裝 Arduino IDE 1.8.19

link to download Arduino IDE 1.8

<https://www.arduino.cc/en/software#future-version-of-the-arduino-ide>



Arduino IDE 1.8.19

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

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

SOURCE CODE

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

DOWNLOAD OPTIONS

Windows Win 7 and newer

Windows ZIP file

Windows app Win 8.1 or 10 [Get](#)

Linux 32 bits

Linux 64 bits

Linux ARM 32 bits

Linux ARM 64 bits

Mac OS X 10.10 or newer

[Release Notes](#)

[Checksums \(sha512\)](#)

File Edit Sketch Tools Help



sketch_sep06a

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

Arduino IDE

Arduino IDE 1.8.19 是一個的開發環境，提供代碼編輯器、實時錯誤檢查。



使用介面簡介



```
sketch_sep06a
void setup() {
  // put your setup code here, to run once:

}

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

}
```



Arduino/Genuino Uno on COM1

啟動 Arduino IDE：您將看到主界面，包括編輯區域、工具欄和菜單欄。



sketch_may30a | Arduino 1.8.19

檔案 編輯 儲存碼 工具 說明

儲存專案

開啟範例或舊檔

開啟新專案

寫入到 arduino 上

編譯程式碼

目前選擇的 arduino 硬體

Arduino Nano, ATmega328P (Old Bootloader) 於 COM4

```
sketch_may30a
1 void setup() {
2 // put your setup code here, to run once:
3 }
4
5 void loop() {
6 // put your main code here, to run repeatedly:
7 }
```

File Edit Sketch Tools Help



Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater

Board: "Arduino Uno"

Port

Get Board Info

Programmer: "AVRISP mkII"

Burn Bootloader

Boards Manager...

Arduino AVR Boards

Bare Conductive Boards (in sketchbook) >

run repeatedly:

Arduino Yún

• Arduino Uno

Arduino Duemilanove or Diecimila

Arduino Nano

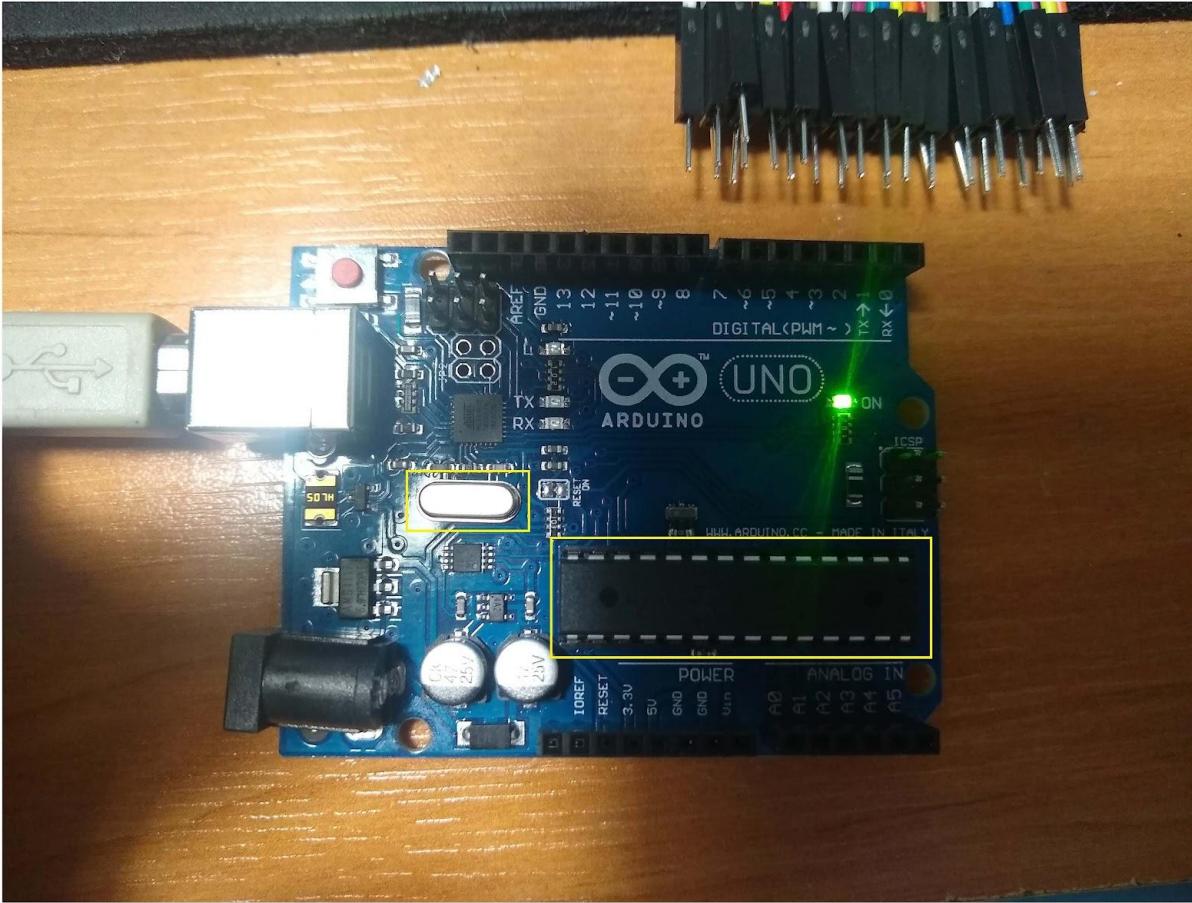
Arduino Mega or Mega 2560

Arduino Mega ADK

Arduino Leonardo

Arduino Leonardo ETH

選擇開發板：在菜單欄，選擇「工具」>「開發板」，然後選擇您所使用的 Arduino 開發板。



先確定 LED 燈亮起

檔案 編輯 草稿碼 工具 說明



sketch_may30a

```
void setup()
// put your
}
```

```
void loop()
// put your
}
```

自動格式化

Ctrl+T

封存草稿碼

修正編碼並重新載入

管理程式庫...

Ctrl+Shift+I

序列埠監控視窗

Ctrl+Shift+M

序列繪圖家

Ctrl+Shift+L

WiFi101 / WiFiNINA Firmware Updater

開發板: "Arduino Nano"

>

處理器: "ATmega328P (Old Bootloader)"

>

序列埠: "COM4"

>

取得開發板資訊

>

燒錄器: "BusPirate as ISP"

>

燒錄Bootloader

>

序列埠

COM1

COM2

COM3

COM4

選擇連接埠：再次點擊「工具」，選擇「連接埠」，然後選擇 Arduino 開發板所連接的串口。

```
sketch_may30a | Arduino 1.8.19
檔案 編輯 草稿碼 工具 說明
sketch_may30a
1 void setup() {
2   // put your setup code here, to run once:
3 }
4 } 初始化 arduino 硬體的程式碼

5 void loop() {
6   // put your main code here, to run repeatedly:
7
8 }
9 } 實際運作的程式碼
```

1 Arduino Nano, ATmega328P (Old Bootloader) 於 COM4

編寫程式碼：在編輯區域編寫您的 Arduino 程式碼，使用 `setup()` 和 `loop()` 函數組織程式結構。

先按這裡做編譯

這裡會顯示編譯的結果！
若有錯誤，就得要修訂才行

編譯完畢。

```
22 https://www.arduino.cc/en/Tutorial/BuiltinExamples/Blink
23 */
24
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27     // initialize digital pin LED_BUILTIN as an output.
28     pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33     digitalWrite(LED_BUILTIN, HIGH);      // turn the LED on (HIGH is the voltage level
34     delay(1000);                      // wait for a second
35     digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW
36     delay(1000);                      // wait for a second
37 }
```

草稿碼使用了 924 bytes (3%) 的程式儲存空間。上限為 30720 bytes。
全域變數使用了 9 bytes (0%) 的動態記憶體，剩餘 2039 bytes 給區域變數。上限為 2048 bytes。

1 Arduino Nano, ATmega328P (Old Bootloader) 於 COM4

驗證程式碼：點擊工具欄上的「✓」按鈕，Arduino IDE 將對您的程式碼進行編譯和錯誤檢查。

先按下上傳按鈕之後

會顯示上傳的結果在這裡！

有問題時，參考這裡的方案

```
22 // https://www.arduino.cc/en/Tutorial/BuiltinExamples/Blink
23 */
24
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27     // initialize digital pin LED_BUILTIN as an output.
28     pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33     digitalWrite(LED_BUILTIN, HIGH);      // turn the LED on (HIGH is the voltage level
34     delay(1000);                      // wait for a second
35     digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW
36     delay(1000);                      // wait for a second
37 }
```

上傳完畢。

avrdude: 924 bytes of flash verified

avrdude done. Thank you.

Arduino Nano, ATmega328P (Old Bootloader) 於 COM4

上傳程式碼：確保程式碼無誤後，點擊工具欄上的「→」按鈕，將程式碼上傳到 Arduino 開發板。

完成以上所有步驟 = 電子零件正常運作

- WELCOME TO -

賽馬會科藝共融計劃

JC-Project-IDEA

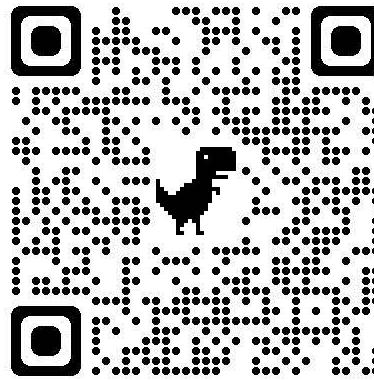
Day02

上午



Starter Project 1 - Graphic Sensors

<https://www.youtube.com/watch?v=29ZIiJJQTFI>



Arduino IDE

Arduino IDE 1.8.19 是一個的開發環境，提供代碼編輯器、實時錯誤檢查。



使用介面簡介

File Edit Sketch Tools Help



sketch_sep06a



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

Arduino/Genuine Uno on COM1

啟動 Arduino IDE：您將看到主界面，包括編輯區域、工具欄和菜單欄。

sketch_may30a | Arduino 1.8.19

檔案 編輯 儲存碼 工具 說明

儲存專案

開啟範例或舊檔

開啟新專案

寫入到 arduino 上

編譯程式碼

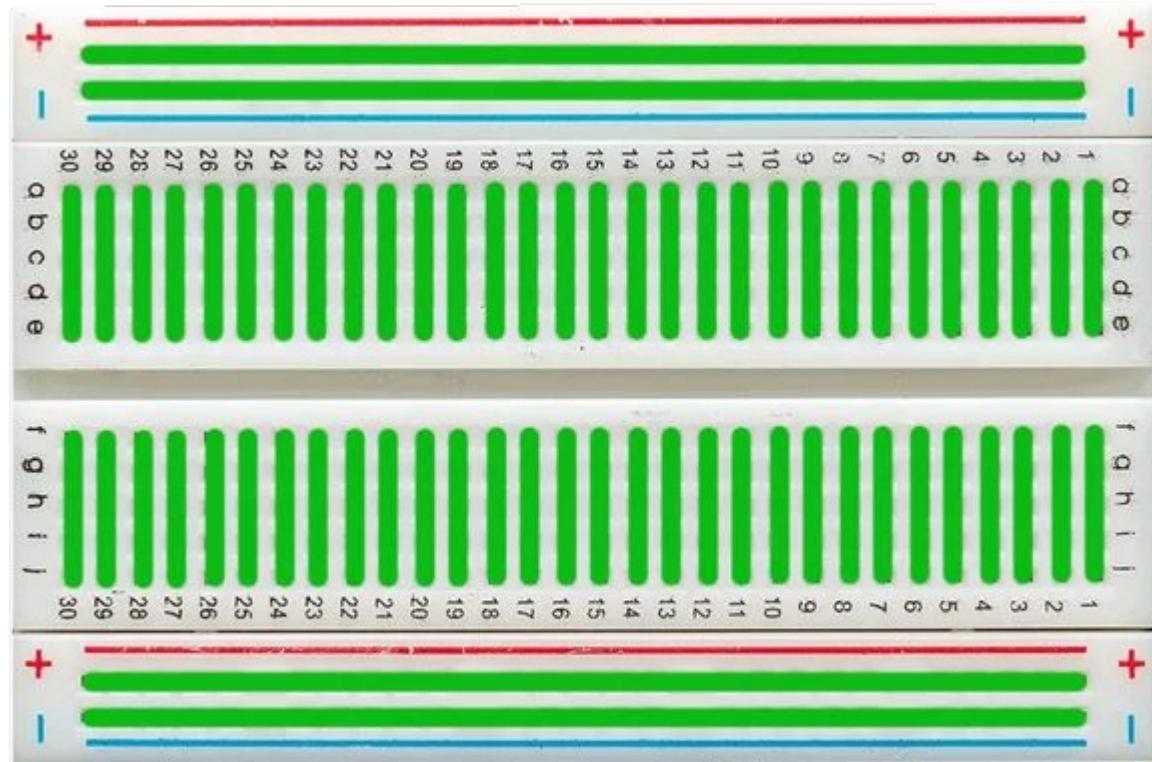
目前選擇的 arduino 硬體

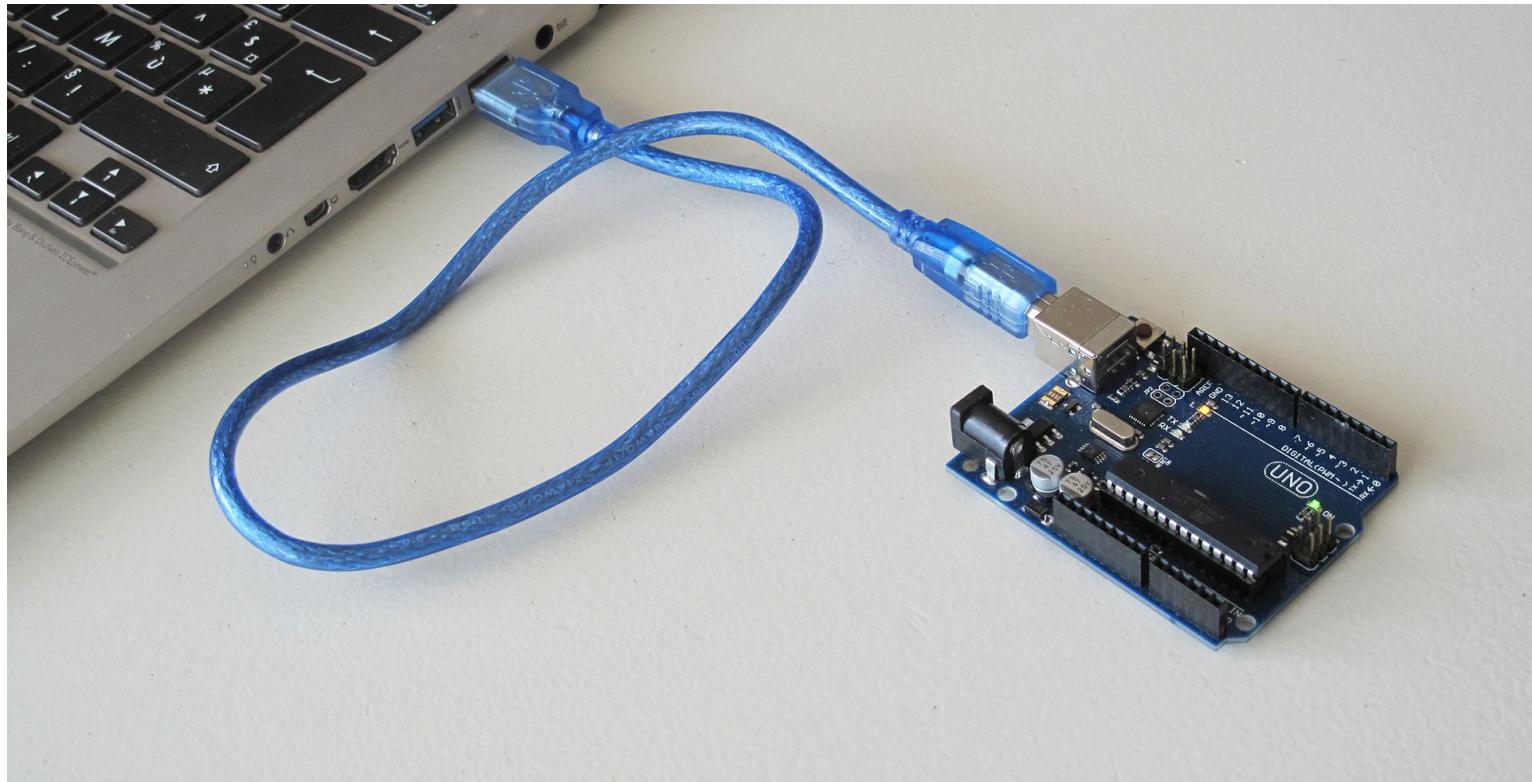
Arduino Nano, ATmega328P (Old Bootloader) 於 COM4

在使用 開發板 做到影片中的效果前

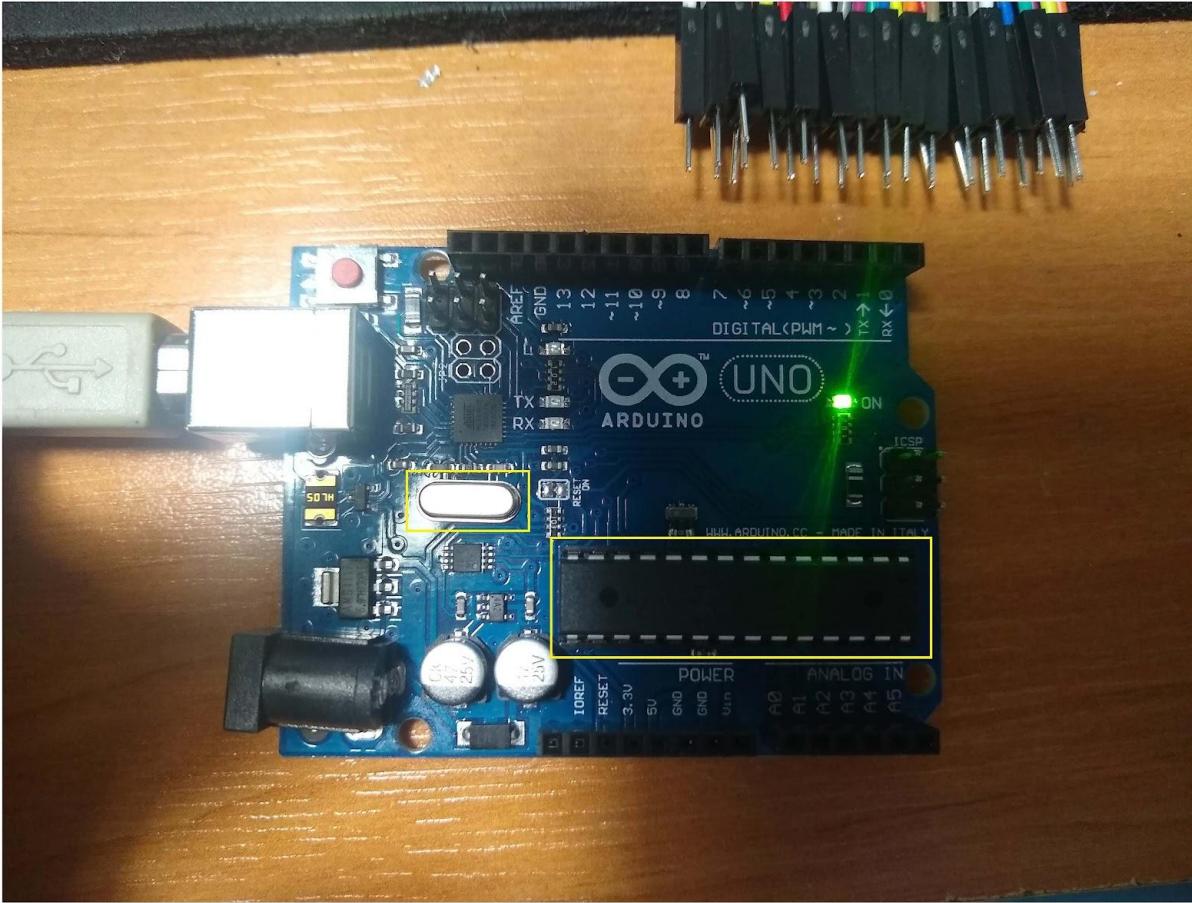
要做的準備工作

麵包板是電子原型設計中常見的工具，用於在無需焊接的情況下建立和測試電路。麵包板上的孔洞按照特定的方式連接，讓使用者能夠插入電子元件並使用導線將它們連接起來。每一列的五個孔洞是相互連接的，可以用來連接元件，而兩側的長條則是用來提供電源。在使用麵包板進行電路連接時，請確保將所有元件和導線正確無誤地插入相應的孔洞，並確保電源和接地線的正確連接，以避免任何可能的短路或故障。





USB連接電腦及開發板



先確定 LED 燈亮起

File Edit Sketch Tools Help

```
sketch_sep22a
void setup() {
    // put your setup code here, to run once:
}

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

Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater

Board: "Arduino Uno" Boards Manager...
Port Arduino AVR Boards
Get Board Info Bare Conductive Boards (in sketchbook)
Programmer: "AVRISP mkII" Burn Bootloader

run once:

run repeatedly:

- Arduino Yún
- Arduino Uno
- Arduino Duemilanove or Diecimila
- Arduino Nano
- Arduino Mega or Mega 2560
- Arduino Mega ADK
- Arduino Leonardo
- Arduino Leonardo ETH

選擇開發板：在菜單欄，選擇「工具」>「開發板」，然後選擇您所使用的 Arduino 開發板。

檔案 編輯 草稿碼 工具 說明



sketch_may30a

```
void setup()
// put your
}
```

```
void loop()
// put your
}
```

自動格式化

Ctrl+T

封存草稿碼

修正編碼並重新載入

管理程式庫...

Ctrl+Shift+I

序列埠監控視窗

Ctrl+Shift+M

序列繪圖家

Ctrl+Shift+L

WiFi101 / WiFiNINA Firmware Updater

開發板: "Arduino Nano"

>

處理器: "ATmega328P (Old Bootloader)"

>

序列埠: "COM4"

>

取得開發板資訊

>

燒錄器: "BusPirate as ISP"

>

燒錄Bootloader

>

序列埠

COM1

COM2

COM3

COM4

選擇連接埠：再次點擊「工具」，選擇「連接埠」，然後選擇 Arduino 開發板所連接的串口。

The screenshot shows the Arduino IDE interface. The title bar reads "sketch_may30a | Arduino 1.8.19". The menu bar includes "檔案", "編輯", "草稿碼", "工具", and "說明". Below the menu is a toolbar with icons for save, upload, and other functions. The code editor contains the following sketch:

```
sketch_may30a

1 void setup() {
2   // put your setup code here, to run once:
3
4 } 初始化 arduino 硬體的程式碼
5

6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 } 實際運作的程式碼
```

The code is divided into two sections: "setup()" and "loop()". The "setup()" section is annotated with "初始化 arduino 硬體的程式碼" (Initialization code for initializing the Arduino hardware). The "loop()" section is annotated with "實際運作的程式碼" (Actual running code).

編寫程式碼：在編輯區域編寫您的 Arduino 程式碼，使用 `setup()` 和 `loop()` 函數組織程式結構。

編碼例子：Blink

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

Blink | Arduino 1.8.19

檔案 編輯 草稿碼 工具 說明

先按這裡做編譯

```
22 https://www.arduino.cc/en/Tutorial/BuiltinExamples/Blink
23 */
24
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27     // initialize digital pin LED_BUILTIN as an output.
28     pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33     digitalWrite(LED_BUILTIN, HIGH);      // turn the LED on (HIGH is the voltage level
34     delay(1000);                      // wait for a second
35     digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW
36     delay(1000);                      // wait for a second
37 }
```

這裡會顯示編譯的結果！
若有錯誤，就得要修訂才行

編譯完畢。

草稿碼使用了 924 bytes (3%) 的程式儲存空間。上限為 30720 bytes。
全域變數使用了 9 bytes (0%) 的動態記憶體，剩餘 2039 bytes 給區域變數。上限為 2048 bytes。

Arduino Nano, ATmega328P (Old Bootloader) 於 COM4

驗證程式碼：點擊工具欄上的「✓」按鈕，Arduino IDE 將對您的程式碼進行編譯和錯誤檢查。

先按下上傳按鈕之後

會顯示上傳的結果在這裡！

有問題時，參考這裡的方案

```
22 https://www.arduino.cc/en/Tutorial/BuiltinExamples/Blink
23 */
24
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27     // initialize digital pin LED_BUILTIN as an output.
28     pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33     digitalWrite(LED_BUILTIN, HIGH);      // turn the LED on (HIGH is the voltage level
34     delay(1000);                      // wait for a second
35     digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW
36     delay(1000);                      // wait for a second
37 }
```

上傳完畢。

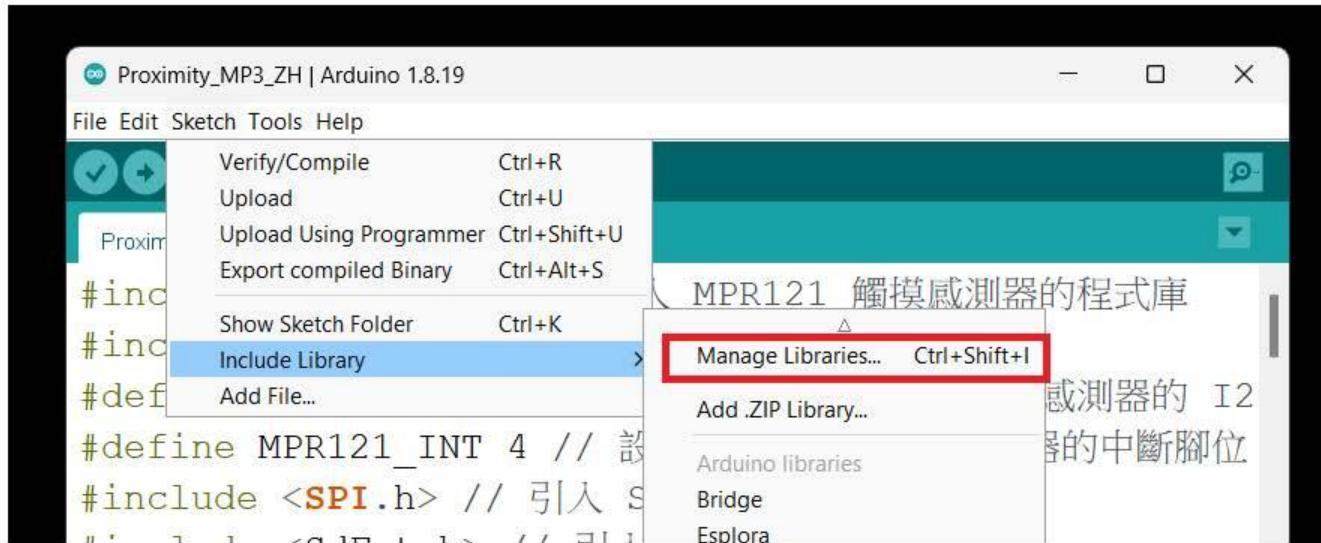
avrdude: 924 bytes of flash verified

avrdude done. Thank you.

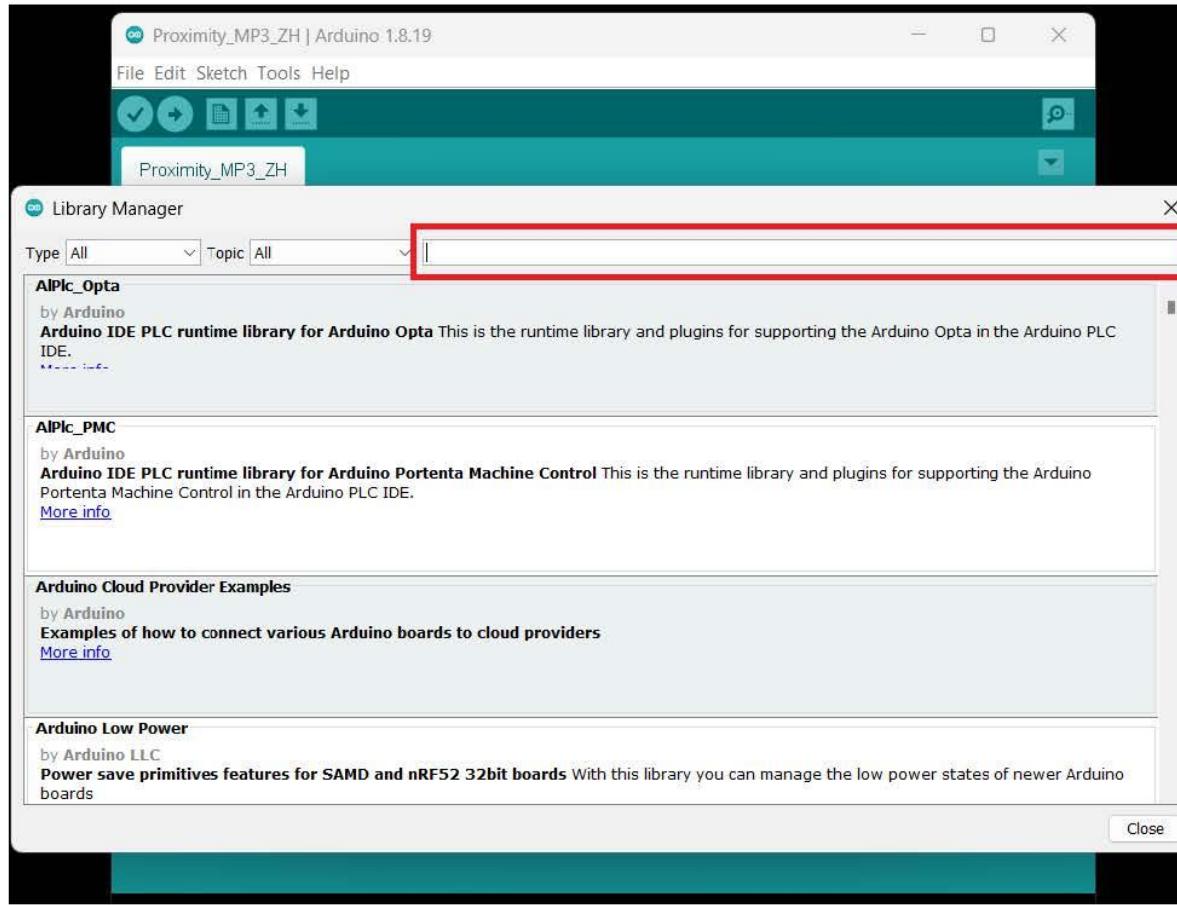
Arduino Nano, ATmega328P (Old Bootloader) 於 COM4

上傳程式碼：確保程式碼無誤後，點擊工具欄上的「→」按鈕，將程式碼上傳到 Arduino 開發板。

完成以上所有步驟 = 電子零件正常運作



在 Sketch 工具欄下, 把滑鼠移到 Inculde Library,
選取 Manage Libraries



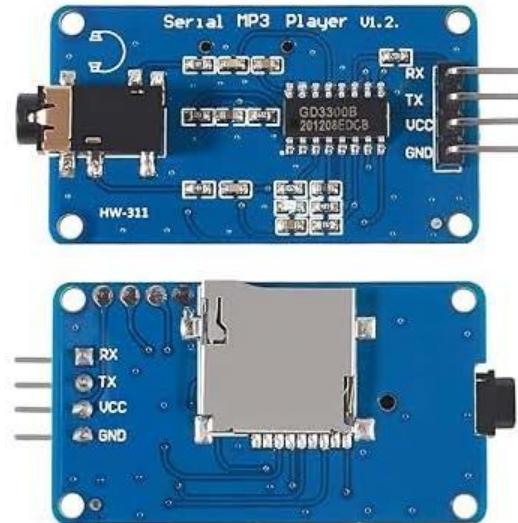
輸入 SerialMp3Player 搜索編碼庫



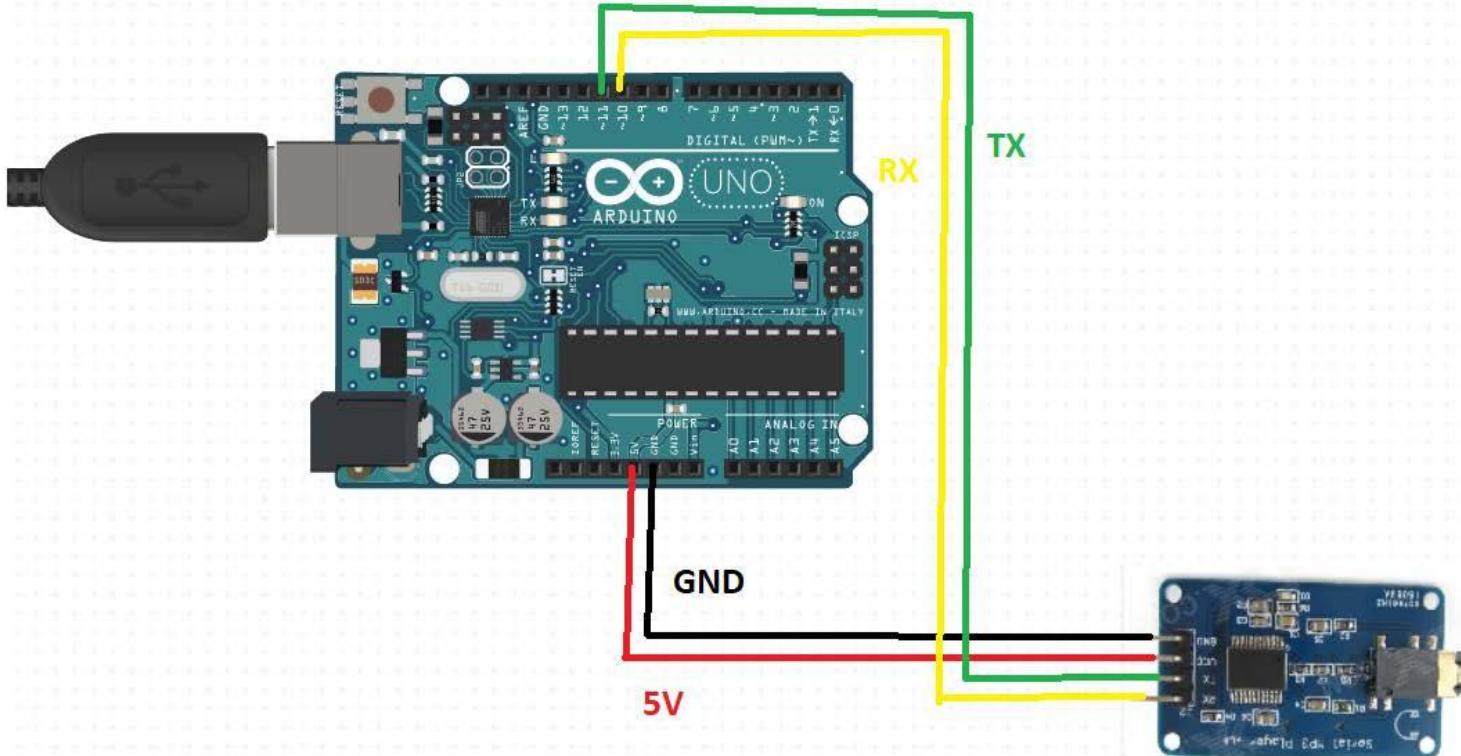
按 Install 安裝編碼庫



另外再安裝 CapacitiveSensor 編碼庫



把 Serial MP3播放模組YX5300/YX6300 接到 Arduino UNO





編程時間

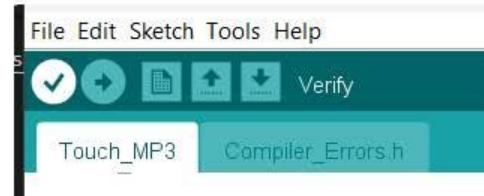
將以下的編碼複製到編輯區域

```
#include "SerialMP3Player.h" // 使用MP3版的編碼庫library
#define TX 10 //to MP3 board RX //定義ARDUINO TX到MP3 RX引腳連接
#define RX 11 //to MP3 board TX //定義ARDUINO RX到MP3 TX引腳連接

SerialMP3Player mp3(RX, TX); // 定義起動MP3相關的TX, RX

//設定：有電源起動時執行一次的程序
void setup() {
    Serial.begin(9600); // 起動serial介面
    mp3.begin(9600); // 開始MP3版的連接
    delay(500); // 等待起動
    mp3.sendCommand(CMD_SEL_DEV, 0, 2); //選取 sd-card
    delay(500); // 等待起動
    mp3.setVol(15); // 設定音量
    mp3.play(1); //歌曲於SD CARD內的次序
}

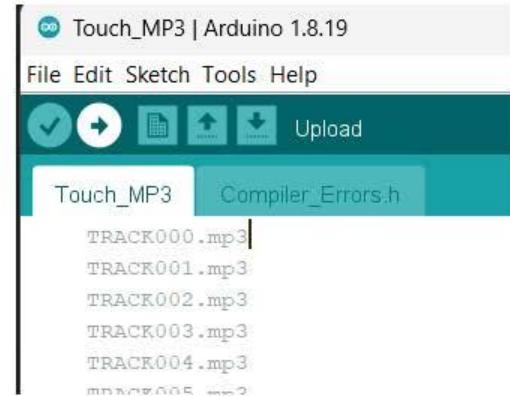
//迴圈：處理器不停執行的程序
void loop() {
}
```



先編譯程式碼

```
Done compiling.  
  
Sketch uses 24038 bytes (83%) of program storage space. Maximum is 28672 bytes.  
Global variables use 1459 bytes (56%) of dynamic memory, leaving 1101 bytes for local variables  
  
16 Bare Conductive Touch Board on COM3
```

如果系統控制台 (Console) 內沒有出現問題, 就可以做下一步。



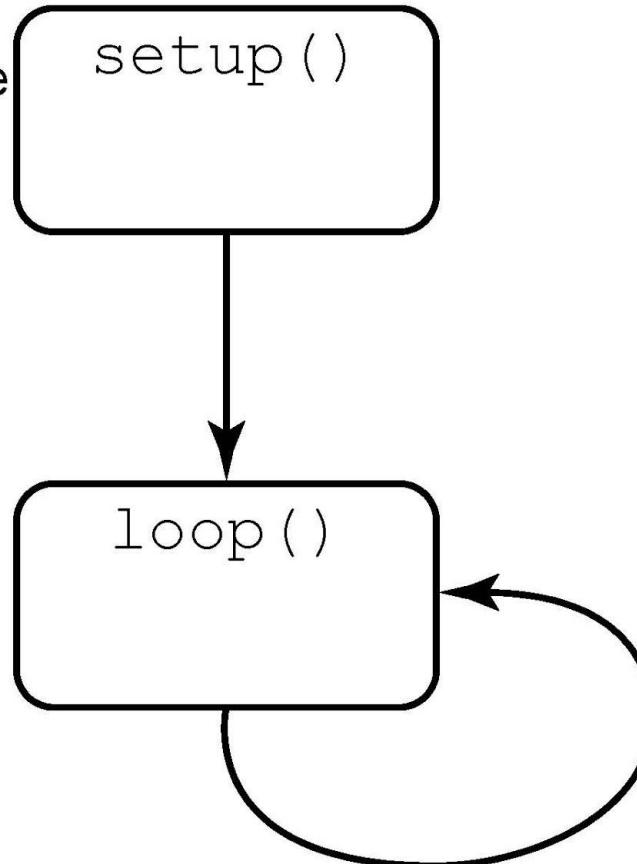
上傳程式碼

有關 void setup() 及 void loop()

void setup() 是 Arduino 程式的一個函式，用於初始化設定，例如設定序列通訊、腳位模式、變數初值等。當 Arduino 板子啟動後，會**自動執行一次** *setup()*。

void loop() 是 Arduino 程式的另一個函式，用於執行主要的程式邏輯，例如讀取腳位數值、判斷條件、控制輸出等。*loop()* **會一直執行**，直到板子關機或重置。

Execution starts here
Execute once.

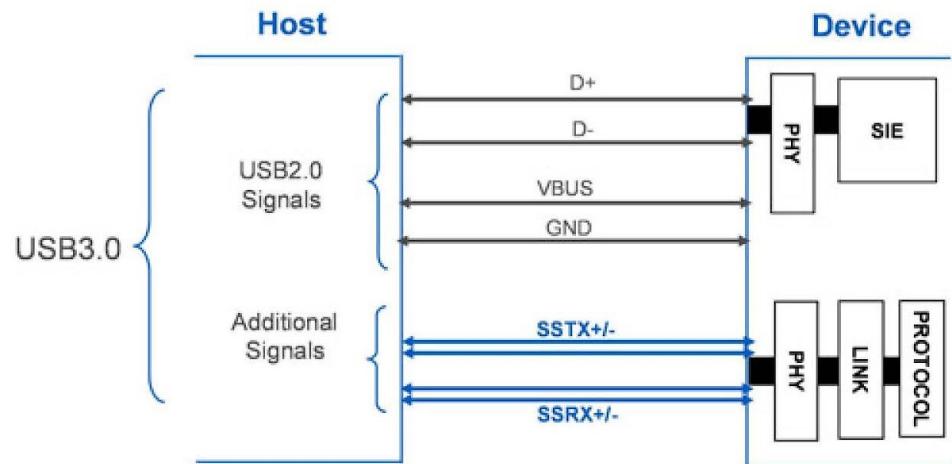


Execute repeatedly.

```
#include "SerialMP3Player.h">// 使用MP3版的編碼庫library  
  
#include <CapacitiveSensor.h>// 使用可感應導電墨水的CAP SENSE編碼庫library  
  
#define TX 10 //to MP3 board RX //定義ARDUINO TX到MP3 RX引腳連接  
#define RX 11 //to MP3 board TX //定義ARDUINO RX到MP3 TX引腳連接
```

Arduino程式碼包含兩個庫：SerialMP3Player和CapacitiveSensor。SerialMP3Player用於控制MP3播放板，CapacitiveSensor用於讀取導電墨水的觸碰輸入。

#define指令創建了兩個名為TX和RX的常量，分別設為10和11，用於表示Arduino與MP3播放板之間串行通訊的接口引腳。



```
//設定：有電源起動時執行一次的程序
void setup() {
    Serial.begin(9600);          // 起動serial介面
    mp3.begin(9600);            // 開始MP3版的連接
    delay(500);                 // 等待起動
    mp3.sendCommand(CMD_SEL_DEV, 0, 2); //選取 sd-card
    delay(500);                 // 等待起動
    mp3.setVol(15);             // 設定音量
    mp3.play(1);                //歌曲於SD CARD內的次序
}
```

setup()函數初始化串口和MP3播放器，並設定音量和播放歌曲。

Serial.begin(9600) 和 *mp3.begin(9600)* 開始串口和MP3播放器的通訊，傳輸速率為9600位元/秒。

mp3.sendCommand(CMD_SEL_DEV, 0, 2) 選擇SD卡作為存儲設備。

mp3.setVol(15) 設定音量為15, *mp3.play(1)* 播放SD卡中的第一首歌曲。

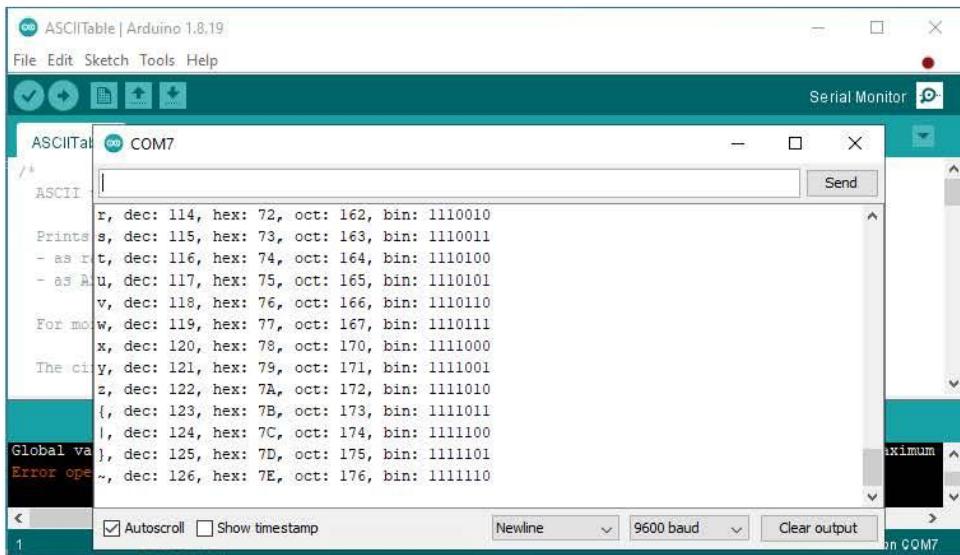
delay(500) 在相關操作之間提供必要的等待時間，以確保操作的成功執行。

編碼內容

- *Serial.begin();*

Serial.begin() 是 Arduino 程式語言的一個函式，用於啟動序列通訊介面，在程式執行時可以透過該介面與外部設備或電腦進行資料傳輸。

- 右圖內的是 Serial Monitor



有關上一段編碼的補充

- *pinMode();*

pinMode() 是 Arduino 程式語言的一個函式，用於設定腳位的輸入或輸出模式，以便與外部電路或設備進行數位或類比訊號的傳輸。

- *Serial.println();*

Serial.println() 是 Arduino 程式語言的一個函式，用於透過序列通訊介面輸出文字或數字訊息，並在最後自動換行。

- WELCOME TO -

賽馬會科藝共融計劃

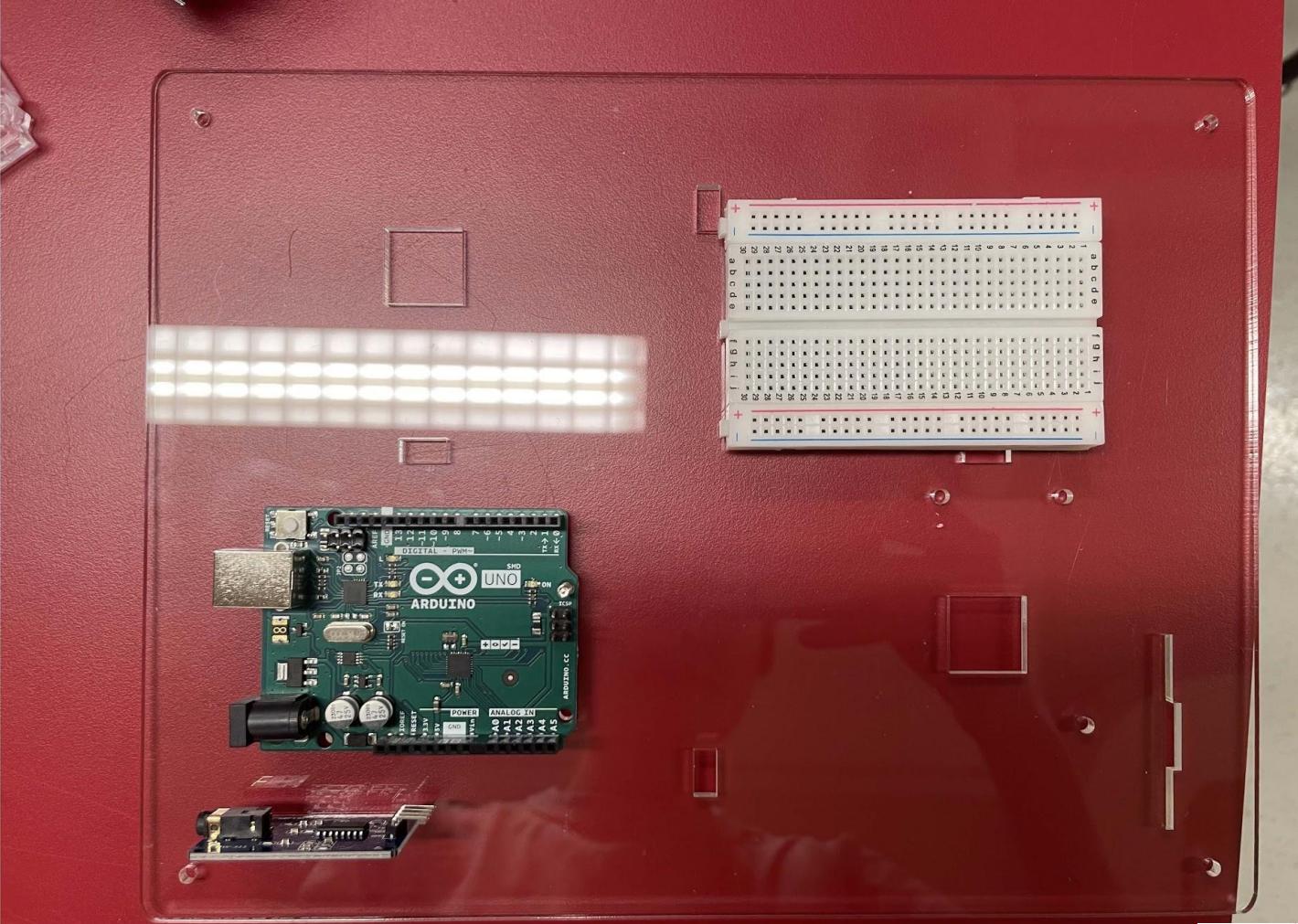
JC-Project-IDEA

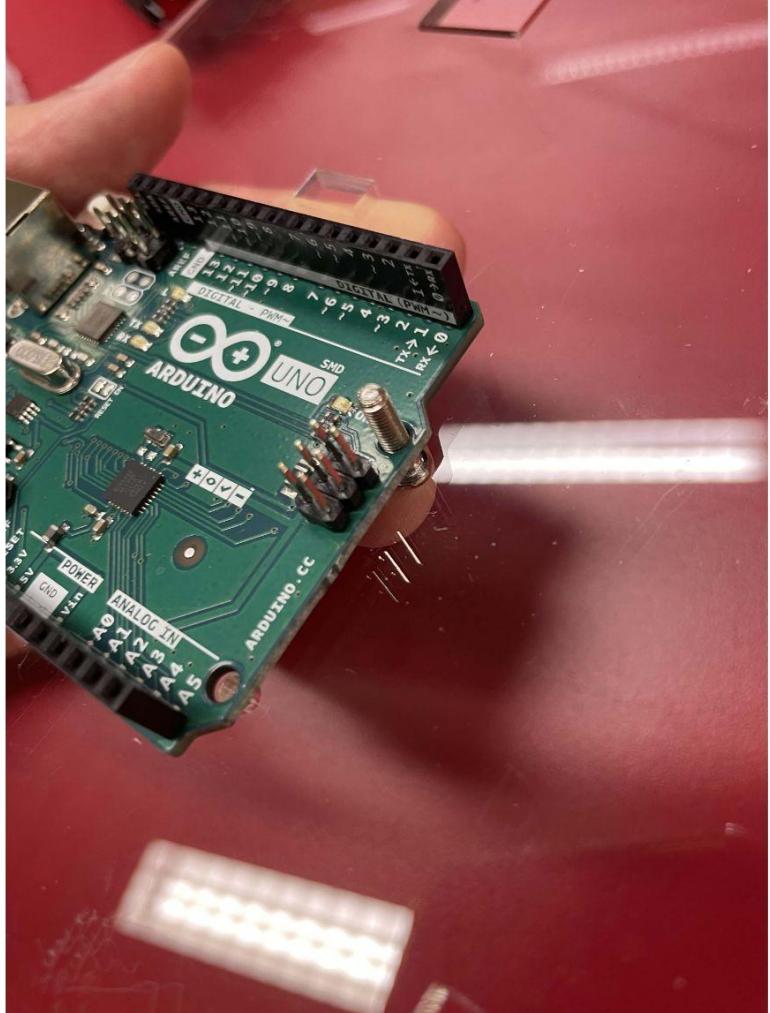
Day02

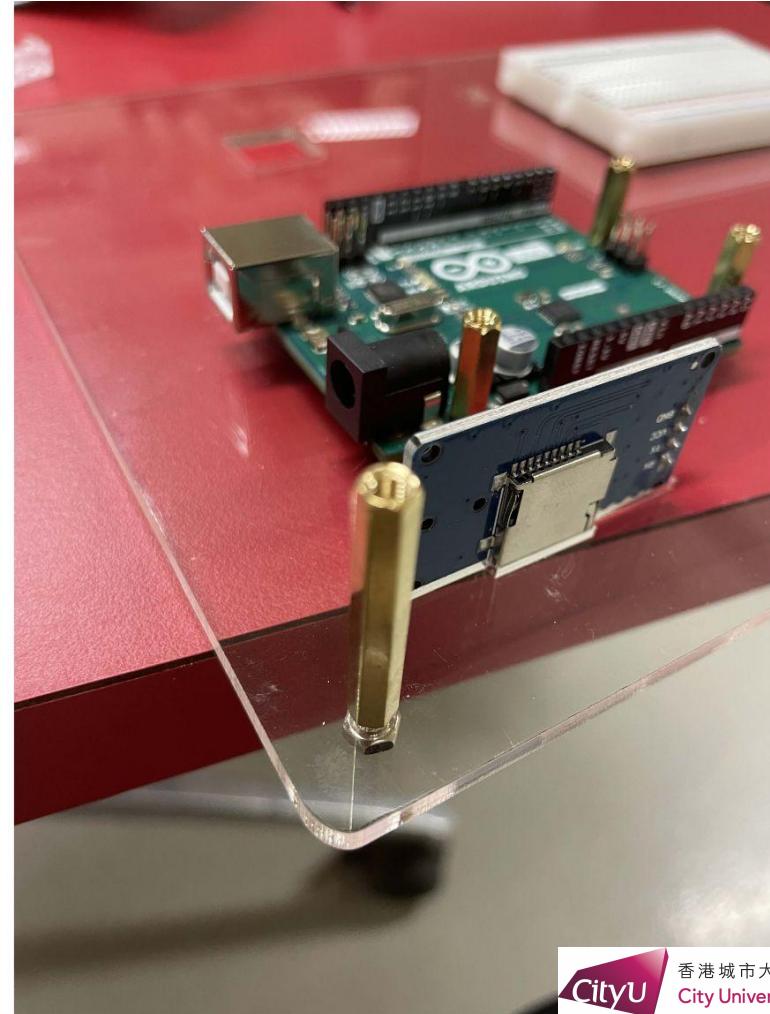
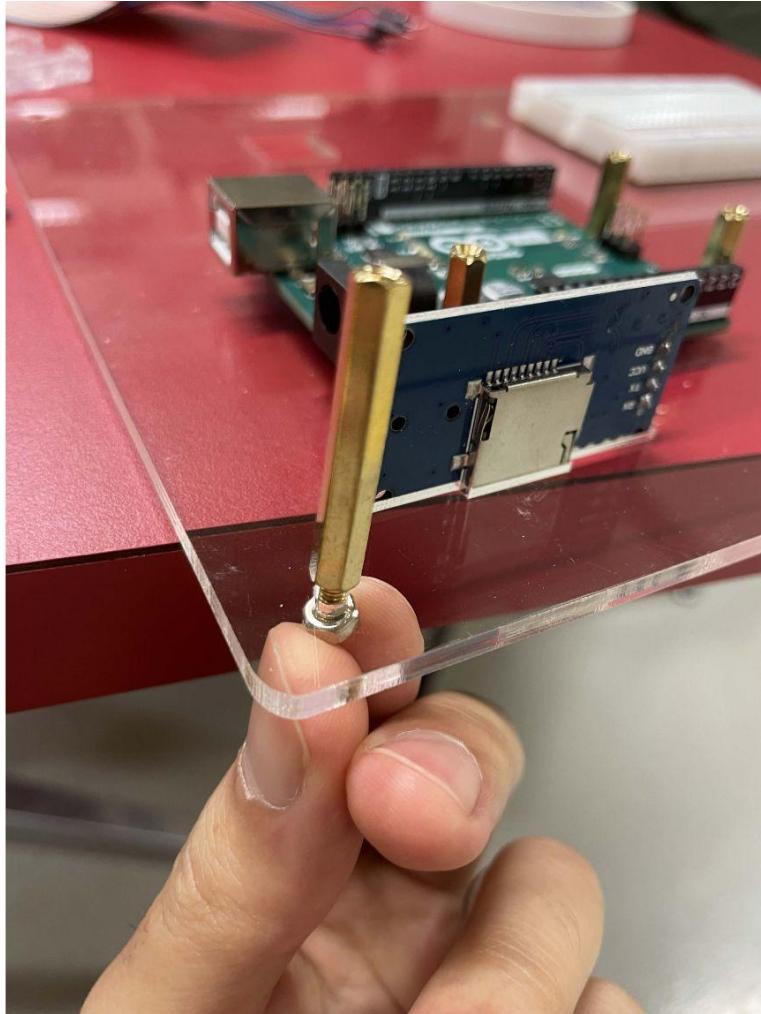
下午

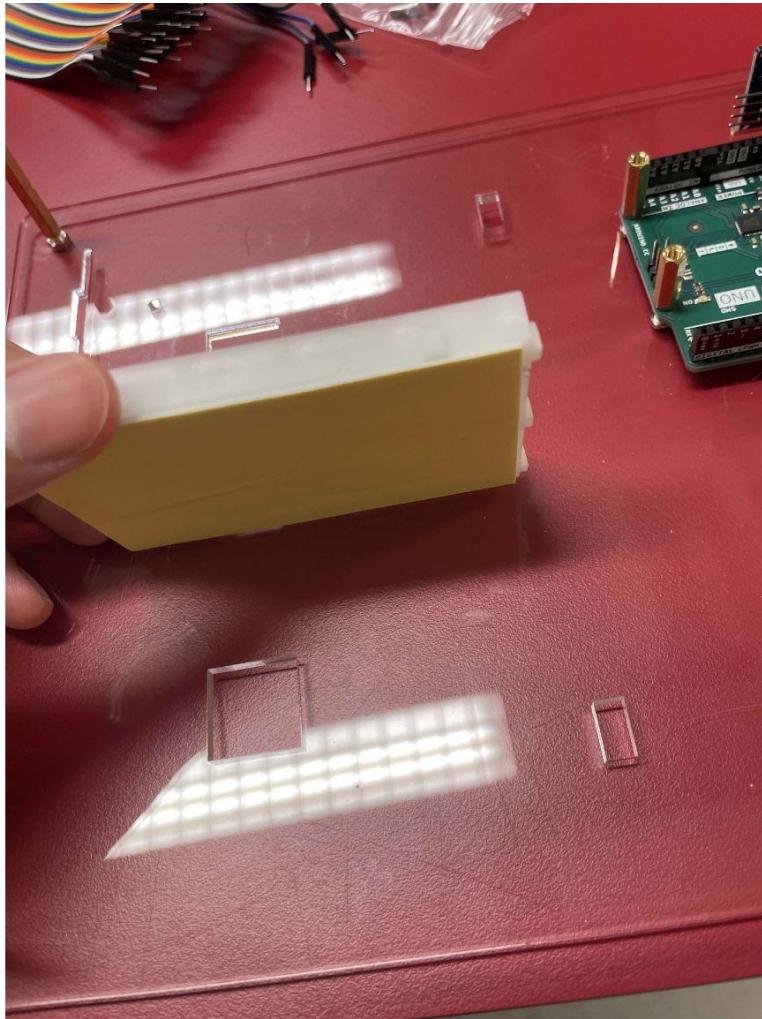
硬件組裝

[https://github.com/JC-Project-IDEA/PHASE-2-
Student-Workshop/blob/main/%E7%B5%84%E8%A3%9D%E6%8C%87%E5%8D%97.pdf](https://github.com/JC-Project-IDEA/PHASE-2-Student-Workshop/blob/main/%E7%B5%84%E8%A3%9D%E6%8C%87%E5%8D%97.pdf)









組裝完成效果



sketch_apr29c §

```
#include "Compiler_Errors.h"

#include <MPR121.h>
#include <MPR121_Datastream.h>
#include <Wire.h>

#include <SPI.h>
#include <SdFat.h>
#include <FreeStack.h>
#include <SFEMP3Shield.h>

const uint32_t BAUD_RATE = 115200;
const uint8_t MPR121_ADDR = 0x5C;
const uint8_t MPR121_INT = 4;

const bool WAIT_FOR_SERIAL = false;

const bool MPR121_DATASTREAM_ENABLE = false;

uint8_t result;
uint8_t lastPlayed = 0;

SFEMP3Shield MP3player;

const bool REPLAY_MODE = true;

SdFat sd;

void setup() {

}

void loop() {

}
```

打開新的 Arduino File (File -> New / Ctrl + N) 將以下的編碼複製到編輯區域

```
#include "SerialMP3Player.h">// 使用MP3版的編碼庫library  
  
#include <CapacitiveSensor.h>// 使用可感應導電墨水的CAP SENSE編碼庫library  
  
#define TX 10 //to MP3 board RX //定義ARDUINO TX到MP3 RX引腳連接  
#define RX 11 //to MP3 board TX //定義ARDUINO RX到MP3 TX引腳連接
```

將以下的編碼複製到編輯區域

```
SerialMP3Player mp3(RX, TX); // 定義起動MP3相關的TX, RX  
  
CapacitiveSensor sensor1 = CapacitiveSensor(3, 2);  
CapacitiveSensor sensor2 = CapacitiveSensor(5, 4);  
CapacitiveSensor sensor3 = CapacitiveSensor(7, 6);  
CapacitiveSensor sensor4 = CapacitiveSensor(9, 8);  
CapacitiveSensor sensor5 = CapacitiveSensor(13, 12);  
//定義CAP SENSE導電感應引腳連接，兩者使用ARDUINO的DIGITAL引腳，並配合電阻達到感應運作  
//前者為SEND PIN，後者為RECIEVE PIN要連接到紙上
```

這段程式碼建立了一個名為 mp3 的 SerialMP3Player 對象，並使用先前定義的 RX 和 TX 引腳。同時，它還創建了 5 個 CapacitiveSensor 對象（sensor1 到 sensor5），用於讀取不同引腳上的觸控輸入。每個對象都接收兩個引腳號碼，一個用於發送脈衝，一個用於接收脈衝，以測量導電墨水的觸控。

將以下的編碼複製到編輯區域

```
void setup() {  
    Serial.begin(9600);          // 起動serial介面  
    mp3.begin(9600);            // 開始MP3版的連接  
    delay(500);                 // 等待起動  
    mp3.sendCommand(CMD_SEL_DEV, 0, 2); // 選取 sd-card  
    delay(500);                 // 等待起動  
    mp3.setVol(50); // 設定音量  
}
```

這段Arduino **setup()** 函數與先前的版本相似，但音量設置有所不同。

Serial.begin(9600) 和 **mp3.begin(9600)** 開始串口和MP3播放器的通訊，傳輸速率為 9600位元/秒。

delay(500) 提供操作之間的必要等待時間。**mp3.sendCommand(CMD_SEL_DEV, 0, 2)** 選擇SD卡作為存儲設備。

最後，**mp3.setVol(50)** 將音量設定為 50，這是與以前版本的主要區別。

將以下的編碼複製到編輯區域

```
void loop() {  
    long measurement1 = sensor1.capacitiveSensor(10); // 讀取SENSOR的數值  
    long measurement2 = sensor2.capacitiveSensor(10); // 讀取SENSOR的數值  
    long measurement3 = sensor3.capacitiveSensor(10); // 讀取SENSOR的數值  
    long measurement4 = sensor4.capacitiveSensor(10); // 讀取SENSOR的數值  
    long measurement5 = sensor5.capacitiveSensor(10); // 讀取SENSOR的數值  
    Serial.print(measurement2); // SERIAL PRINT SENSOR的數值以方便MAPPING  
    Serial.println("\t");  
    if (measurement1 >= 60) { // 決定觸發起動歌曲的條件 (值)  
        mp3.play(4); // 歌曲於SD CARD內的次序  
    }  
    if (measurement2 >= 60) { // 決定觸發起動歌曲的條件 (值)  
        mp3.play(2); // 歌曲於SD CARD內的次序  
    }  
    if (measurement3 >= 60) { // 決定觸發起動歌曲的條件 (值)  
        mp3.play(5); // 歌曲於SD CARD內的次序  
    }  
    if (measurement4 >= 60) { // 決定觸發起動歌曲的條件 (值)  
        mp3.play(1); // 歌曲於SD CARD內的次序  
    }  
    if (measurement5 >= 60) { // 決定觸發起動歌曲的條件 (值)  
        mp3.play(3); // 歌曲於SD CARD內的次序  
    }  
    delay(50); // 迴圈再執行的中間位  
}
```

```
long measurement1 = sensor1.capacitiveSensor(10); //讀取SENSOR的數值  
long measurement2 = sensor2.capacitiveSensor(10); //讀取SENSOR的數值  
long measurement3 = sensor3.capacitiveSensor(10); //讀取SENSOR的數值  
long measurement4 = sensor4.capacitiveSensor(10); //讀取SENSOR的數值  
long measurement5 = sensor5.capacitiveSensor(10); //讀取SENSOR的數值
```

loop() 函數讀取五個觸控感應器的值，並根據每個感應器的讀數決定是否播放特定的歌曲。每個感應器的讀數使用 *capacitiveSensor(10)* 方法獲取，結果儲存在 *measurement1* 到 *measurement5* 變數中。然後，*measurement2* 的值透過串口列印出來供調試。

```
...
if (measurement1 >= 60) { //決定觸發起動歌曲的條件(值)
    mp3.play(4);        //歌曲於SD CARD內的次序
}
...
delay(50);
```

接著，若任一感應器的讀數大於或等於60，將播放對應的歌曲。例如，如果 *measurement1* 大於或等於60，則播放SD卡中的第四首歌曲。同樣，如果 *measurement2* 大於或等於60，則播放第二首歌曲，依此類推。

最後，*delay(50)* 保證在每次迴圈迭代之間有短暫的暫停，以避免過度使用處理器資源。

sketch_apr29c §

```
#include "Compiler_Errors.h"

#include <MPR121.h>
#include <MPR121_Datastream.h>
#include <Wire.h>

#include <SPI.h>
#include <SdFat.h>
#include <FreeStack.h>
#include <SFEMP3Shield.h>

const uint32_t BAUD_RATE = 115200;
const uint8_t MPR121_ADDR = 0x5C;
const uint8_t MPR121_INT = 4;

const bool WAIT_FOR_SERIAL = false;

const bool MPR121_DATASTREAM_ENABLE = false;

uint8_t result;
uint8_t lastPlayed = 0;

SFEMP3Shield MP3player;

const bool REPLAY_MODE = true;

SdFat sd;

void setup() {

}

void loop() {

}
```

除錯階段

有機會遇上的問題

1. 在編碼中遺漏了特定符號 如 : (), ,
2. 編碼不在 {} 之內

留意!!! Syntax 句法

- 指一門語言裡支配句子結構, 直到組成句子的規則或過程。

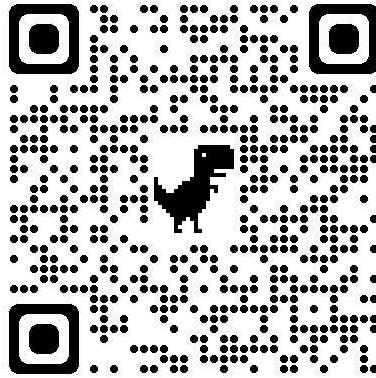
<https://en.wikipedia.org/wiki/Syntax>

如發現編碼排位混亂 可以使用 CTRL + T 自動整理編碼功能



Physics of AUM/OM mantra made
visible - CYMATICS - Sound of
Creation

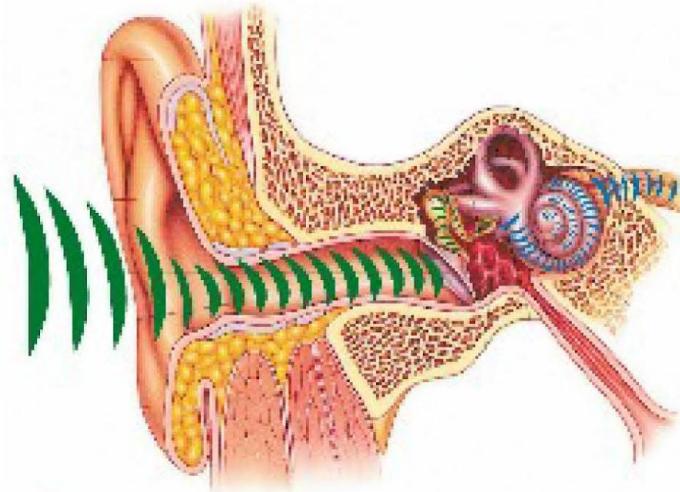
<https://www.youtube.com/watch?v=fozlNrtdzrQ>



聲音是？

聲音是振動產生的聲波，通過介質（氣體、固體、液體）傳播並能被人或動物聽覺器官所感知的波動現象。

聲音的頻率一般會以赫茲表示，記為 Hz，指每秒鐘週期性震動的次數。而分貝是用來表示聲音強度的單位，記為 dB。



綠色代表聲波以震動能傳遞到耳膜及聽小骨

藍色代表由內耳轉換為電波能

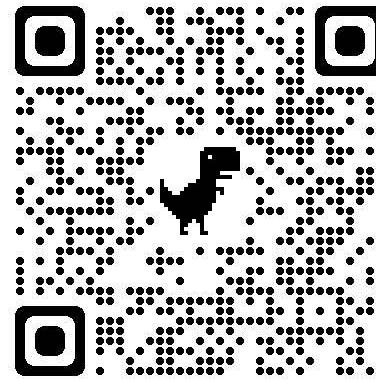
再經由聽神經傳達至大腦聽覺區

Oscillator 振盪器

Oscillator 是指能夠產生定期波形的電子電路或元件，可用於產生時鐘訊號、調整頻率和相位等。在電子學和通訊系統中，oscillator 常用於產生高頻信號，如無線電訊號，以及在數位電路中作為時序控制器。

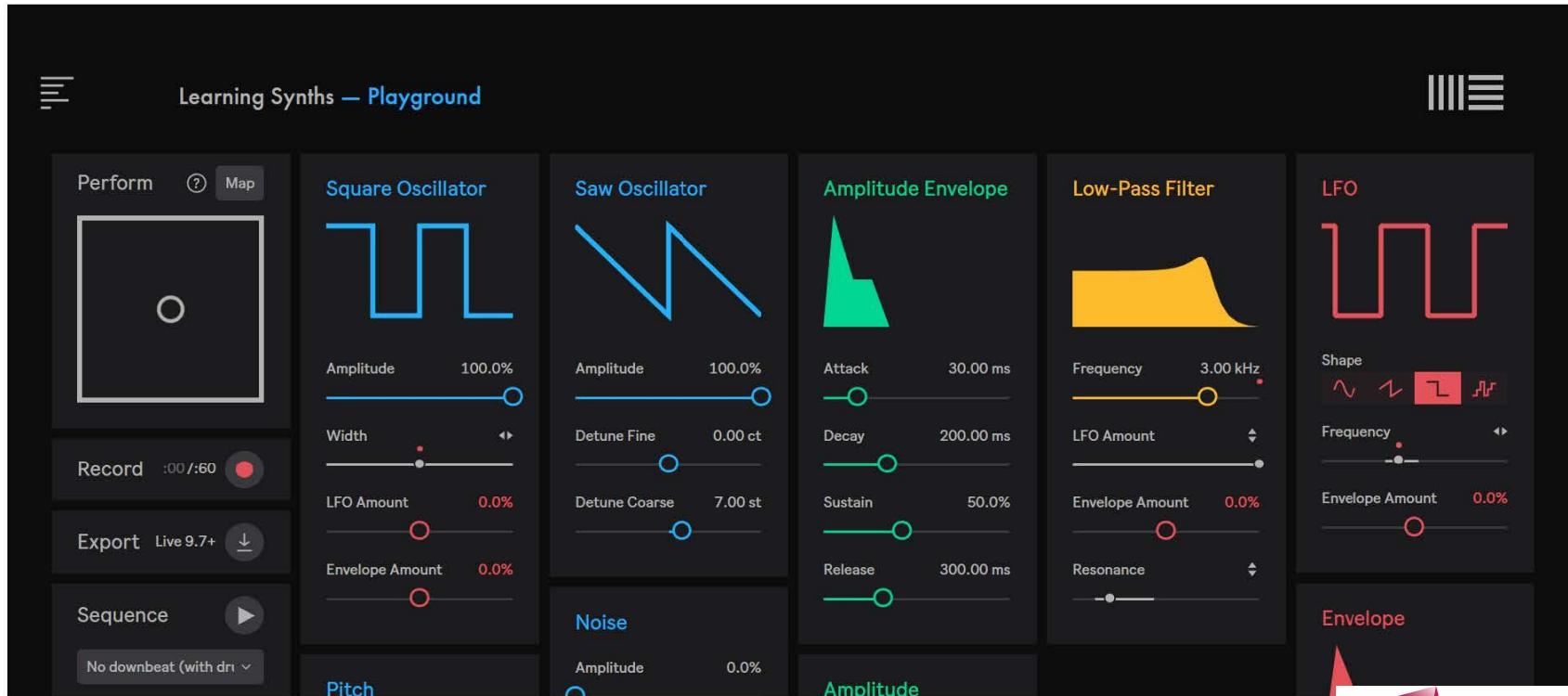
A Brief History of Synthesizers

[https://www.youtube.com/watch?v=5sjr
eF6H_rY](https://www.youtube.com/watch?v=5sjreF6H_rY)

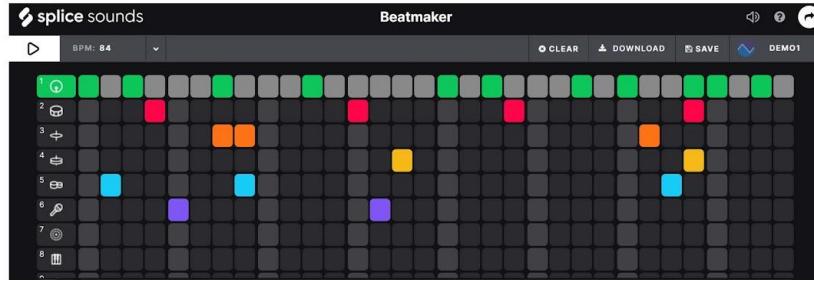


學一下 Synthesizer (合成器)

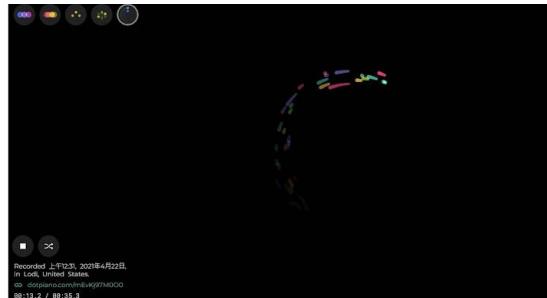
<https://learningsynths.ableton.com/en/playground>



有趣的網上音樂創作工具



<https://splice.com/sounds/beatmaker/026f92d3afda>



<https://dotpiano.com/mEvKj97M0O0>

除錯時段

作品在硬件上或編碼上有任何問題的話，請告知身邊的工作人員及老師。

簡報時間

現在邀請大家分享自己的作品

(*今まで)
こうじやなくて



賽馬會科藝共融計劃

Jockey Club Project IDEA

Inclusive Digital and Experimental Art





香港城市大學
City University of Hong Kong

賽馬會科藝共融計劃

Jockey Club Project IDEA

Inclusive Digital and Experimental Art

專業 創新 胸懷全球
Professional · Creative
For The World





香港城市大學
City University of Hong Kong

賽馬會科藝共融計劃

Jockey Club Project IDEA

Inclusive Digital and Experimental Art

專業 創新 胸懷全球
Professional · Creative
For The World