



開放式自學教育的學校！

STEAM程式教育

USE 3D/Arduino/Python/雷切/電子電路

1

Outline

- 14 : 00 ~ 14 : 30 STEAM教育總覽
- 14 : 30 ~ 15 : 00 組裝電子徽章
- 15 : 00 ~ 16 : 00 程式實作

2

STEAM教育總覽

3

天闊科技 趣學校教育品牌

先思考一個具體的問題

20年後的職業（產業）
應是如何的樣貌？

4

未來學習

我們籌劃著，孩子未來教育學習的方式！

我們希望未來學習的課程，是**理論與實踐並存**！

譬如著，我們學習天平的計算，卻不了解天平的結構，更奇妙的是，我們還發展一系列的計算公式與題型。

等等！我們還不知到天平長得如何？如何運作？

5

該如何改革？

如果每次段考所驅動孩子的進步，但**缺乏思考**的重要性！那我們所培養的，稱為名之"考試學霸"的一種機器，這種機器看似可以被未來的大數據，機器人，AI人工智慧所取代，屆時，人類又剩下什麼？

6

我們僅剩下，創新思考！

這類的能力培養，在於我們熟知"不流血的革命"中，特別顯著！許多科學家，爆炸性思考，探索，對於當時的未來！不乏許多，必須**依賴實驗操作，產生具體結果，進而下了結論！**

即使是理論科學家的代表"愛因斯坦"，也必須依賴科學家愛丁頓出發到非洲西海岸-普林西比島 (Principe) 觀察日全蝕驗證廣義相對論！

7

我們的教育呢？

離實驗越來越遠了！ 升學主義的興起，造就死背答案，以答案為主臬，減少思考！孩子認為背了！有分數，下課後很辛苦，不用在花費任何時間進行學習！造就惡性循環！他們越沈迷遊戲，這原因來自於逃避與速成背誦，減少思考所造成！

8

這是一個物質優渥的時代！

我們處於科技便利的時代，但卻利用30年前的方法，進行教育！
這十分可惜與浪費！浪費了最佳思想培養的時間背景！

這時代擁有，高處理速度的微型電腦，手機，筆電，單晶片電
路板，3D列印機，雷射切割機，AR，VR，XR...等。

這些優質的工具，我們都撇乾淨，還是採用傳統訓練學生的方
式進行著，便是"考試卷，考試卷，考試卷..."。

9

我們希望打造未來教室

在未來，學習每個單元時，我們都有搭配一項實體教
材！這實體教材，可能是手機APP，可能是3D列印，
可能是雷射切割...等，我們採用課綱對應單元內容，
引導學習學科內的數理項目，讓這些艱深的學問，在
既有的學理基礎，發展適當對應的實做！

10

我們希望打造未來教室

而學習單位，可以是**自學，補教，正規學校，親子互動學習，社區有志一同產生的班級**，我們希望都能帶領這樣的方式，讓更多人一同參與！

這是**這世紀，發展到第20年後**，應扎根的重要項目！
是翻轉，帶領人類前往下個世紀重要的基石！

11

程式架構

12

程式開發流程

- 在做微處理器的程式開發時，最關鍵的一步就是將程式編譯成單晶片看得懂的機械語言，而這部份的工作多半是交由電腦上的相關程式來執行。
- 目前常用於單晶片系統的程式碼為組合語言 (Assembly)、C/C++ 等種類，設計完系統所要的執行流程，最後再編譯成特殊格式的檔案，也就是能讓微處理器看的懂的語言，常見的檔案是副檔名為 .hex。

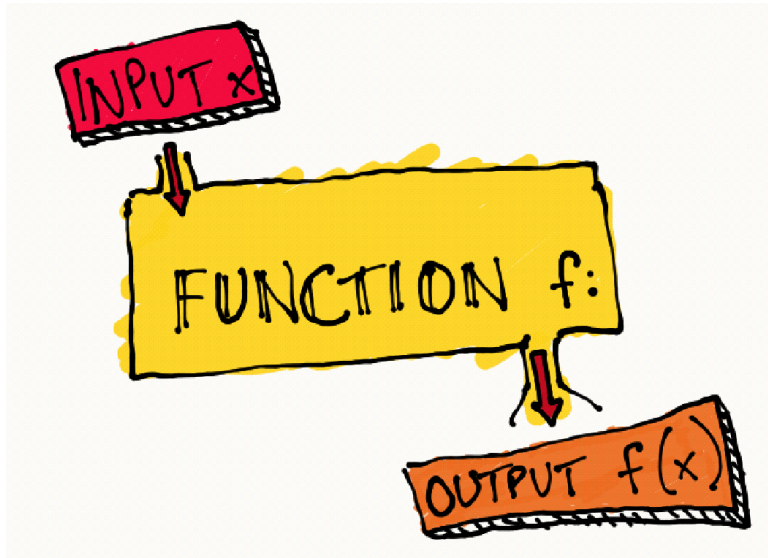
<#>

何謂程式？



14

何謂程式？



15

何謂程式？

Input (x)	Output (y)
1	2
2	4
3	6
4	8

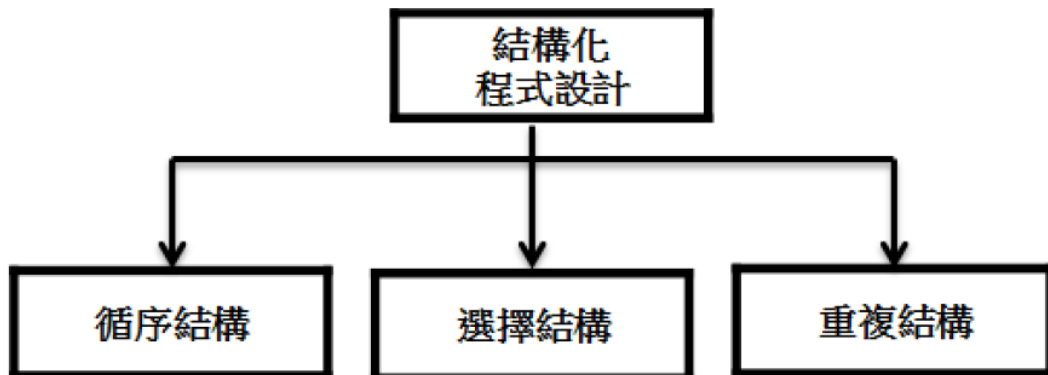
16

何謂程式？

Input (x)	Processing =>	Output (y)
1	* 2	2
2	* 2	4
3	* 2	6
4	* 2	8

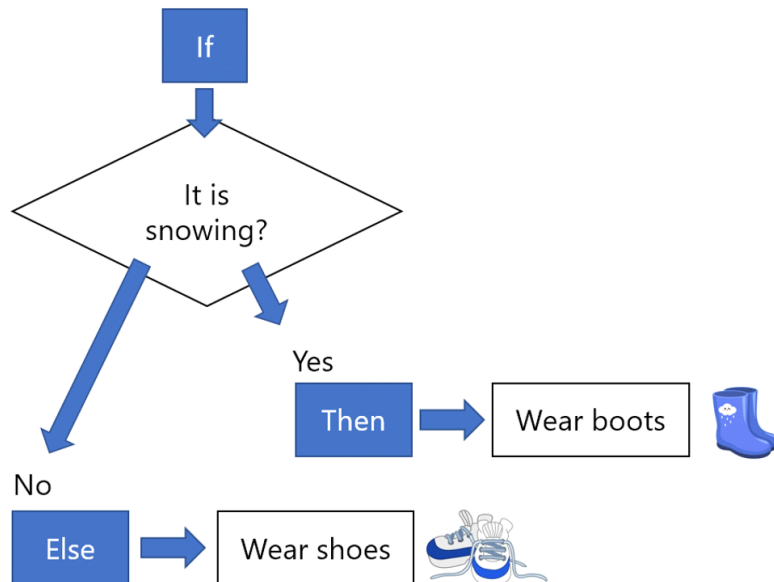
17

結構化程式設計



18

何謂程式？



19

何謂程式？ Lather, rinse, repeat

The **programmer** got stuck in the shower because the instructions on the shampoo bottle said,

Lather, Rinse, Repeat.

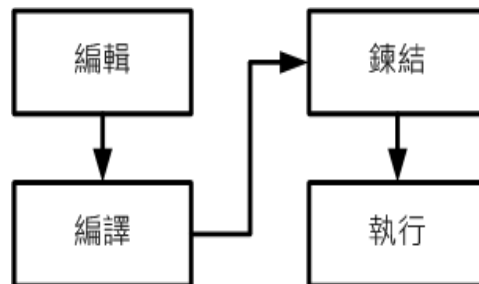


DBWebSolutions.com

20

程式開發 - 編輯

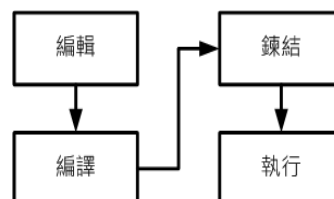
- 這是開發的第1個步驟：產生程式碼。
- 一個專屬的程式開發環境，還可以幫你管理專案內的多個程式，也可以利用顏色來區分程式碼類的內容。



<#>

程式開發 - 編譯

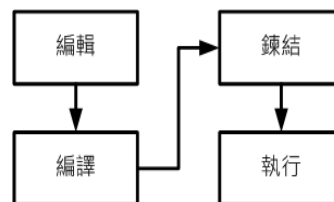
- 這個階段的工作就是將我們編輯完的文字檔轉換成為機器碼，這個步驟會幫你檢查程式上的錯誤，並提出警告。因此這個步驟完後通常會回到前一步驟將語法錯誤、邏輯錯誤的地方進行修正，直到編譯器沒有產生錯誤為止。



<#>

程式開發 - 鍊結

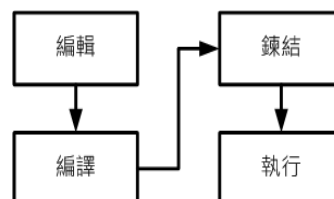
- 當我們的專案程式越來越大時，為了方便管理，通常會分功能的儲存成不同的檔案，鍊結的功能就是將程式當中所有用到的功能模組、內建函式庫找尋原始程式的位置，再與主程式結合成為一個可以執行的檔案。



<#>

程式開發 - 執行

- 能夠執行程式的話，恭喜已經成功的完成前面的步驟，終於可以看到程式所跑出來的結果了，不過能執行不一定表示結果是符合預期的，還是有可能有各種不同的錯誤情形會發生。



<#>

IDE

- 編輯、編譯、鍊結和執行等過程在某些單晶片系統上是需要分作不同的應用程式來完成不同的步驟，可能先經由A程式產生出燒錄檔，再透過B程式將檔案燒錄到單晶片中，這樣分段式的開發流程不甚方便，故系統商或是第三方(the third party)的軟體商會整合所有的過程，統整為單一的環境，稱為IDE(Integrated Development Environment)，整合式開發環境。

<#>

編譯器

- 一個現代編譯器的主要工作流程如下：源代碼 (source code) → 預處理器(preprocessor) → **編譯器 (compiler)** → 彙編程式 (assembler) → 目標代碼 (object code) → 鏈接器(Linker) → 可執行文件

26

Arduino程式架構

(一)宣告:

Arduino宣告方式與C相同,例如:int , float...

(二)初始化 Setup():

使Arduino板子裝置妥當的指令

EX:

```
int ledPin=7;    /*宣告Arduino7號腳為輸入腳*/  
Setup()  
{  
    pinMode(ledPin,INPUT);  
}
```

27

(三)執行Loop():

為程式的主要內容,這程式內容會一直重複被執行

EX:

```
Loop()  
{  
    .....  
}
```

28

(四)函式:

1. pinMode(7,INPUT)

//將腳位7設定為輸入模式

2. digitalWrite(8,HIGH) (數位腳專用)

//將腳位8設定輸出高電位

3. val=digitalRead(7) (數位腳專用)

//讀出腳位7的值並指定給
val變數

29

4.analogWrite(9,128) (數位訊號專用所設計的函式)

//將擁有PWM的數位腳位9
設定輸出電位2.5V對應值大約為128

5. val=analogRead(0) (類比腳專用)

//讀出腳位0的值並指定給val變數(且analogRead可讀取
範圍0(0V)~1023(5V))

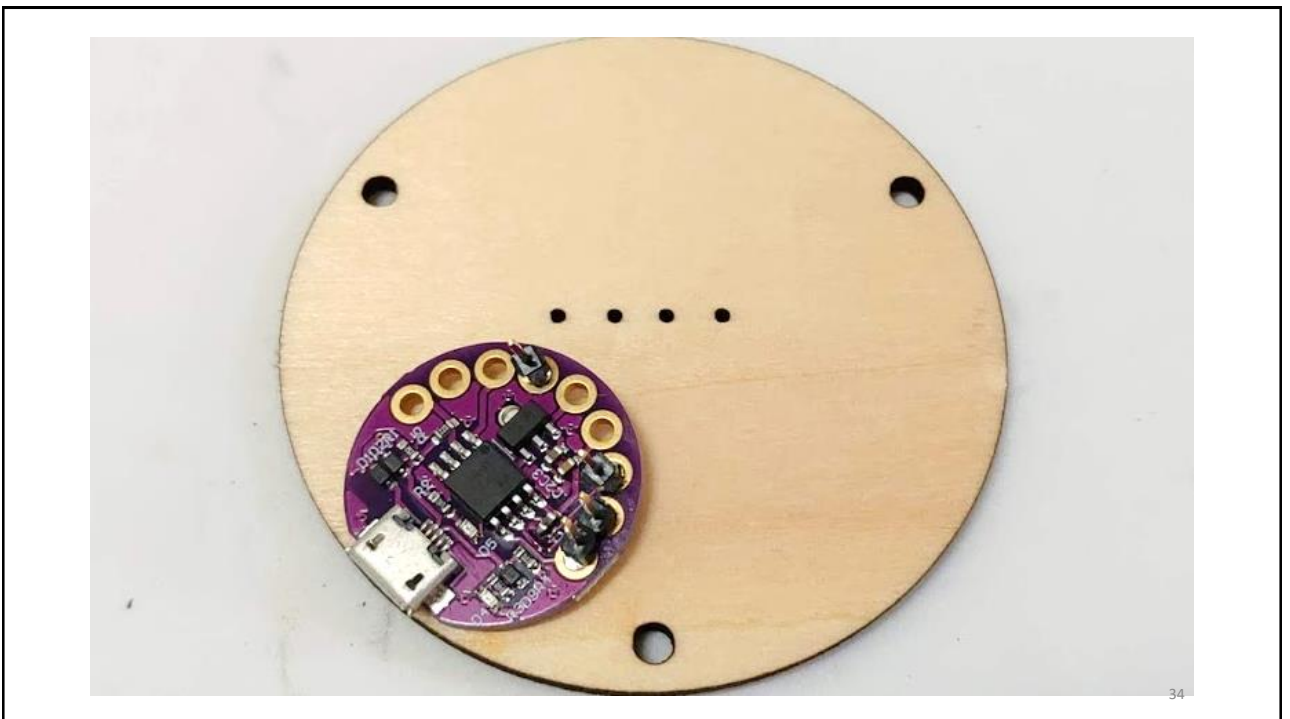
30

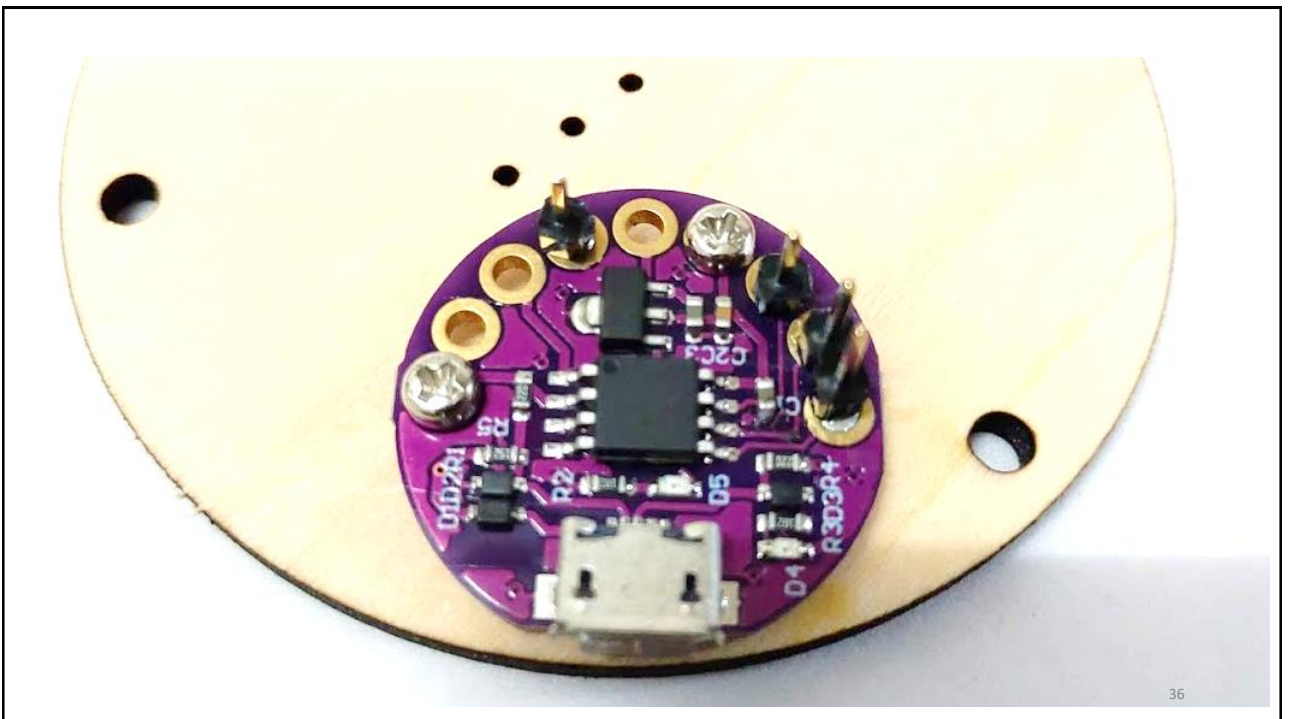
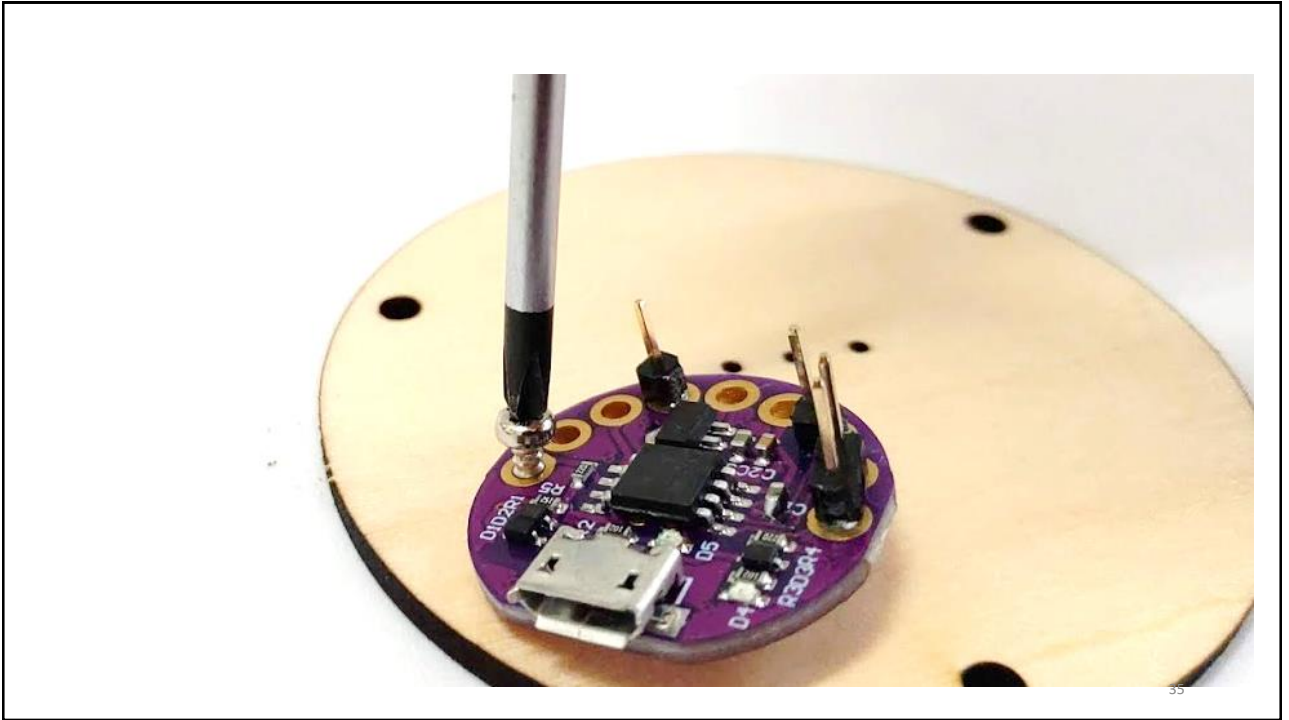
組裝徽章

31



32



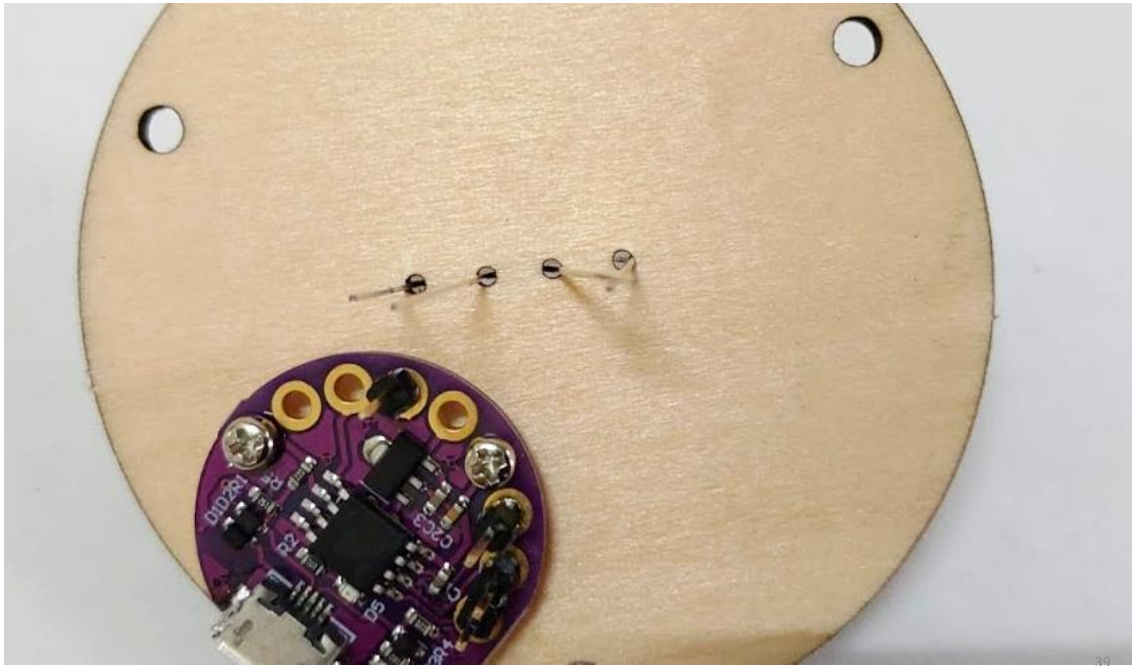




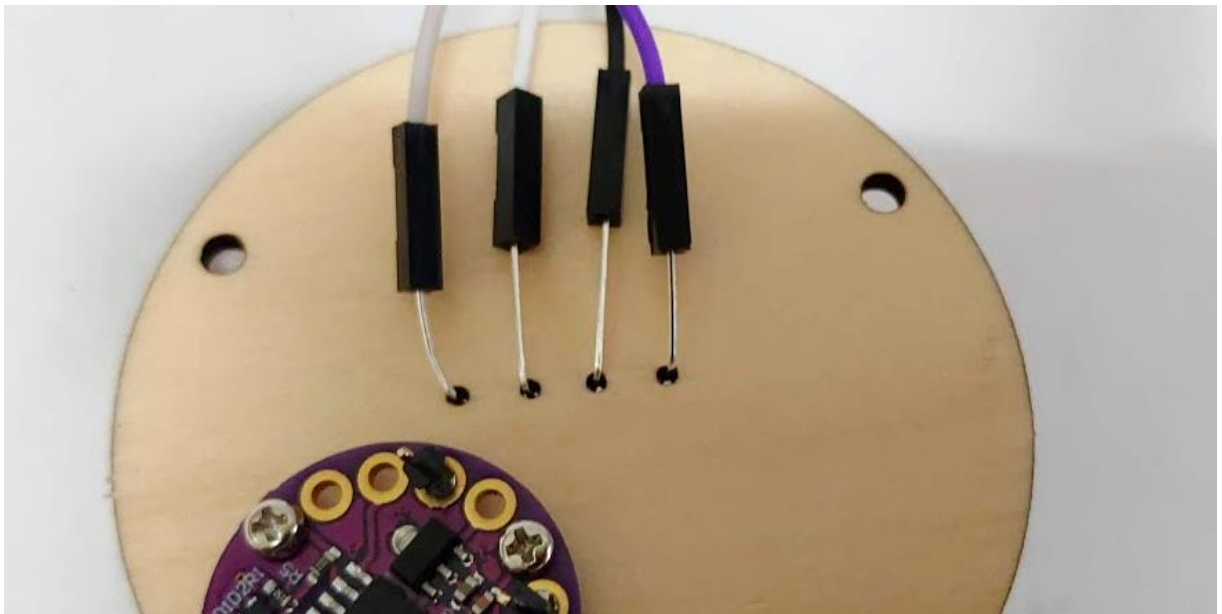
37



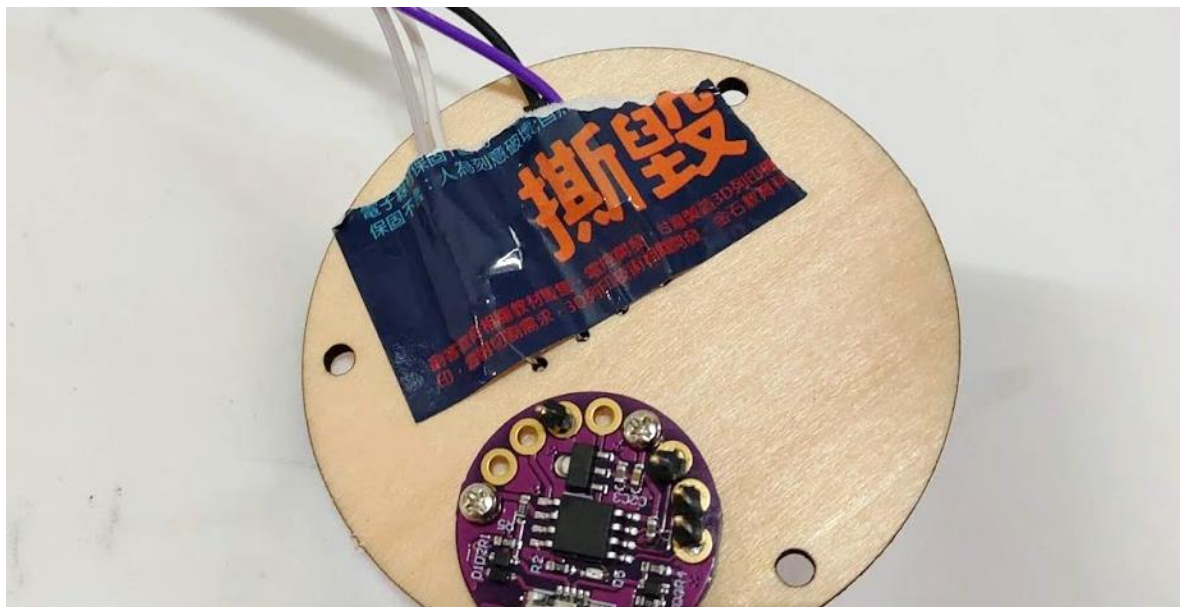
38



39



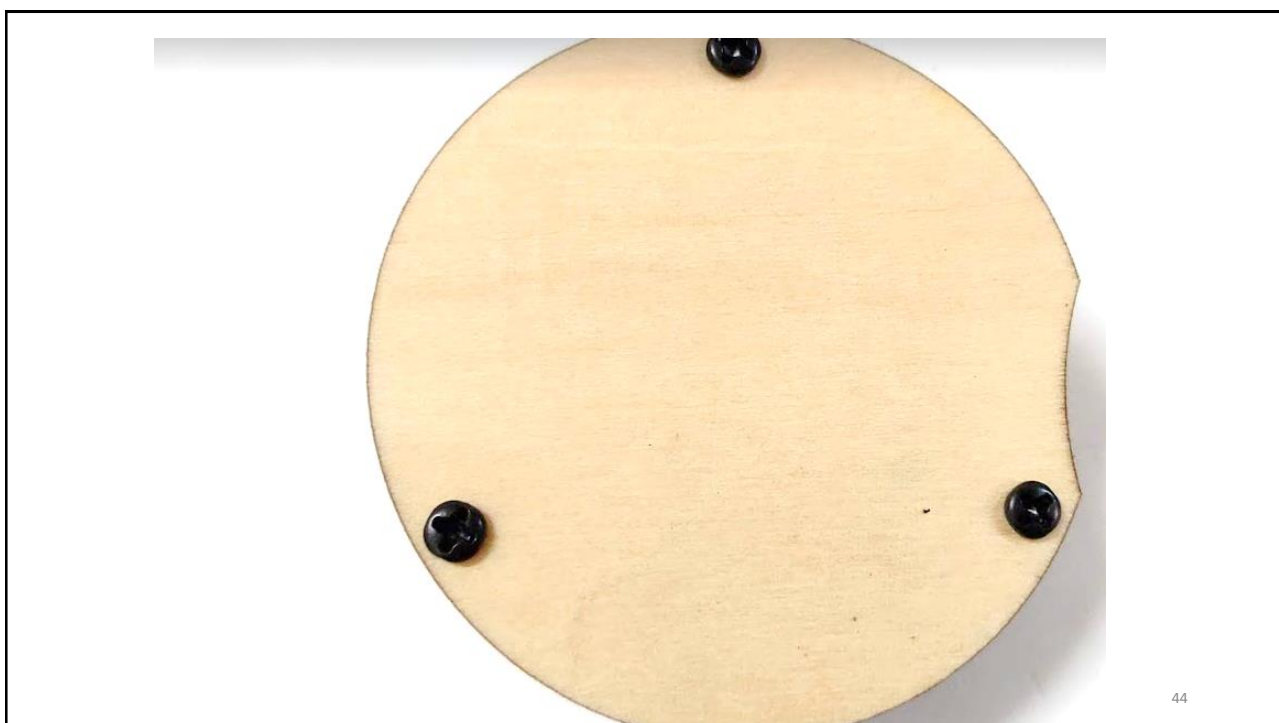
40



41



42





45

下載編輯程式

46

www.austone.org



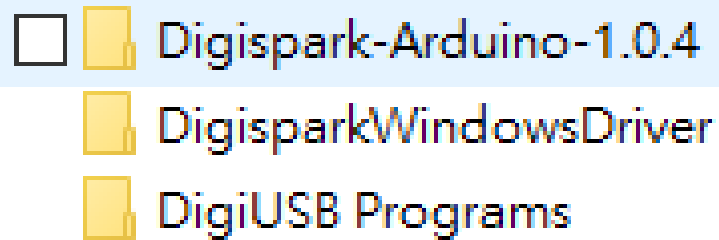
47

下載文件



48

下載後解壓縮有三個資料夾



49

1Down > Digispark-Arduino-1.0.4

☐ 名稱

- drivers
- examples
- hardware
- java
- lib
- libraries
- reference
- tools
- arduino.exe ✓
- cygconv-2.dll
- cygwin1.dll
- libusb0.dll
- revisions.txt
- rtxSerial.dll

開啟程式

50

Down > DigisparkWindowsDriver

☐ 名稱

- amd64
- x86
- _DriverFiles.7z
- 7zDP_LZMA.cfg
- 7ZDP_LZMA.sfx
- Digispark_Bootloader.inf
- dpinst.xml
- dpinst32.exe ✓
- dpinst64.exe ✓✓
- dpscat.exe
- InstallDriver.exe
- README.md

安裝Digipark

51

安裝DigiUSB

Down > DigiUSB Programs

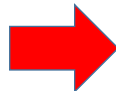
☐ 名稱

- ☒ DigiUSB Windows Driver
- source
- digiusb.exe
- pdccurses.dll
- readme.txt
- receive.exe
- send.exe

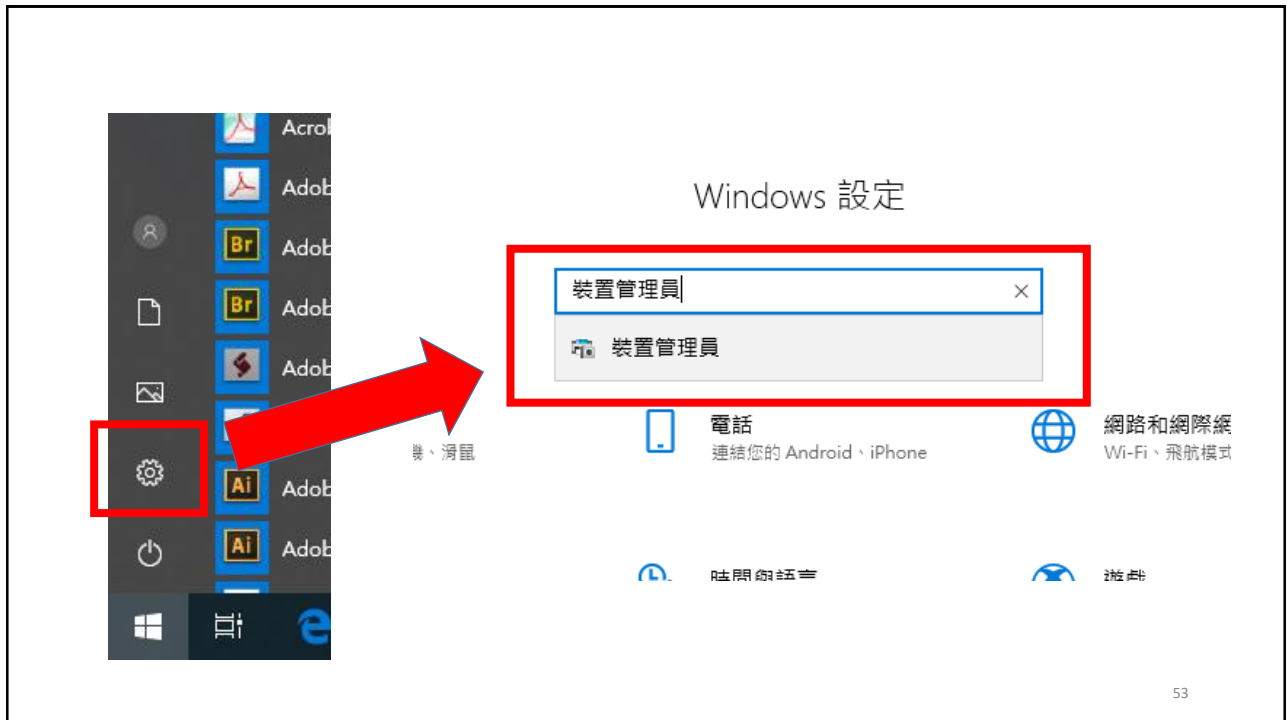
> DigiUSB Windows Driver

名稱

- amd64
- x86
- _DriverFiles.7z
- 7zDP_LZMA.cfg
- 7ZDP_LZMA.sfx
- DigiUSB.inf
- dpinst.xml
- dpinst32.exe
- dpinst64.exe
- dpscat.exe
- InstallDriver.exe
- Instructions.txt
- re-pack-files.cmd



52



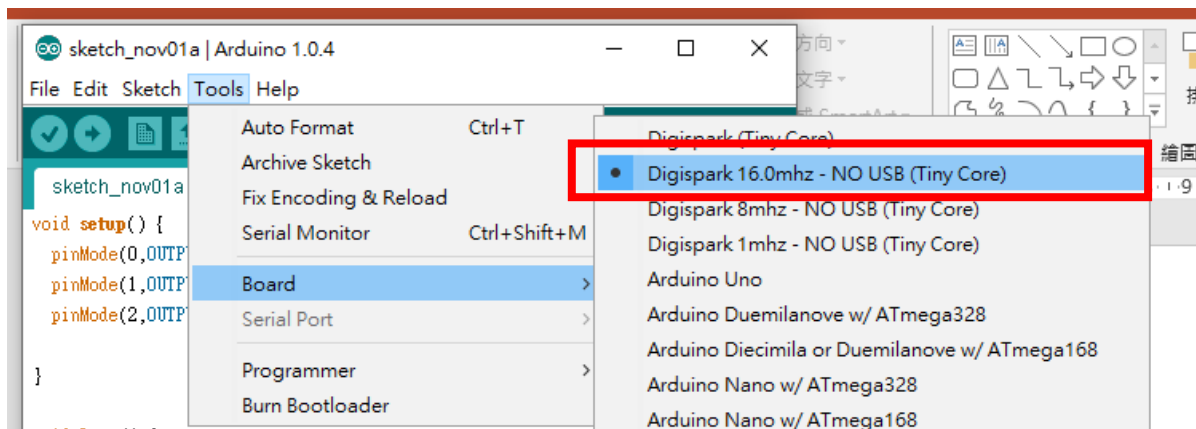
安裝完成會有出現板子已經安裝完成



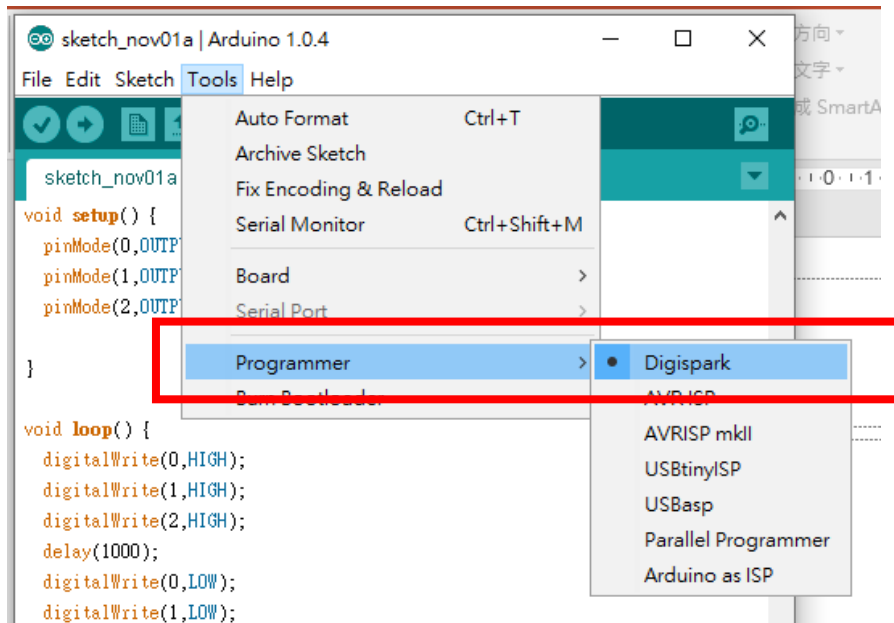
程式演練

55

選板子



56



57

第一支程式
循序燈

```
void setup() {  
  pinMode(0,OUTPUT);  
  pinMode(1,OUTPUT);  
  pinMode(2,OUTPUT);  
}  
  
void loop() {  
  digitalWrite(0,HIGH);  
  digitalWrite(1,HIGH);  
  digitalWrite(2,HIGH);  
  delay(1000);  
  digitalWrite(0,LOW);  
  digitalWrite(1,LOW);  
  delay(3000);  
}
```

59

第二支程式 呼吸燈

```
int LED=0;
int i=0;
void setup() {
  pinMode(LED, OUTPUT);
}
void loop() {
  for(i=0;i<256;i=i+32){
    analogWrite(LED, i);
    delay(200);
  }
}
```

61

非常感謝