前往下載教學資源 www.geniopy.com.tw







### 為什麼要選Python程式語言呢?

#### 簡單好學

Python 的基本語法規則很 簡單, 學習曲線短, 而且語法 設計上已經預防你寫錯, 可以 減少初學時的時間浪費!

#### 現成功能多

有全世界眾多善心人士寫好 的功能 (稱為**模組**)可以套用, 要做任何事都可以省去自己 從頭打造的時間!

#### 大家都在用

現在最火熱的資料科學、人 工智慧、機器學習等都是以 Python 程式語言為主流, 學了到哪裡都會用得到!



Python 可以唸拍嬸 (英國) 也可以唸拍賞 (美國) 這個單字是蟒蛇, 所以 Python 的 logo 就是蟒蛇 來源是英國喜劇 Monty Python's Flying Circus







#### 軟硬體介紹

1

2

3

● 認識 Tinka 開發板

2 MicroPython 開發環境安裝及操作

**3**更新開發板韌體

### MicroPython 基礎應用

MicroPython 建立物件、變數
利用 GPIO 腳位控制LED燈
利用 GPIO 腳位讀取按鍵

#### MicroPython 進階應用

● MicroPython 自訂函式

2 計時器應用









# MIT? Made In Taiwan!!

行政院自去在即业。	viace in ranvaria
關及三級機關的資通安全實地 核重點對象,檢視其設備汰除; 該官員進一步說,已通令全國包括中央 使用或採購的中國廠牌資通產品,各機 情況,回報期限在本月底,不少機關因 設備及中國品牌投影機,也將面臨全面 設備及中國品牌投影機,也將面臨全面 資安處指出,已責請各機關資; (含軟體、硬體及服務)相關 有經費需求應循程序爭取資源, 進行汰換。	凌通科技股份有限公司("凌通"),是一家台灣企業位於台灣 300 新竹市新竹科學 工業園區工業東四路 19 號,在此聲明 GP329_STEM-KIT-20 晶片由凌通設計, 委由 SMT 廠: <u>宸宥科技股份有限公司</u> 製造,是一家台灣企業位於台灣 300 新竹市 香山區埔前路 240 號,及 PCB 廠: <u>百為實業有限公司</u> 製造,是一家台灣企業位於 台灣 238 新北市樹林區三俊街 81 巷 42 號。           Generalplus Technology In 凌通科技股份有限公司           廣轄代表人: 賈懿行 Kevin Chia 職稱:總經理 President

# 2 PART

# **安裝與測試環境** 動手做:安裝環境







## 連接 USB 傳輸線並檢查連接埠編號











		▶ 系統自動	偵測代入USB連接埠
G+MassProductionTool for GPL	329x: v1.0.6.4		
Script: gp3297xxa	a_spi download.conf Open Map Port	Start <u>D</u> ownload	Enum Device Device Number: 1
🖨 [Hub 1][Port 1]:	Ready!> SN: 62f263f36 <u>0</u> #{53		
🖕 [Hub 1][Port 5]:			
🖕 🤤 [Hub 1][Port 8]:			
G+MassProductionTool for GPL	329xx v1.0.6.4		
G+MassProductionTool for GPL:	329xx v1.0.6.4		
Script: gp3297xxa	a_spi_download.conf Open Map Port	Stop Download	Enum Device Device Number: 1
🚓 [Hub 1][Port 1]:	100% Download Completed!		
~ [Hub 1][Port 5]:			
🖕 🤄 [Hub 1][Port 8]:			





燒錄完成後記得拔除 jumper, 不然開發板會一直在燒錄模式, 無法進行其他操作!





## 安裝Thonny開發環境



# 設定 Thonny









## 3 Python 基礎入門 動手做:寫出最簡單的程式 PART

# Python 的核心 ★ 物件



同一個物件有很多不同的方法			
互動環境(Shell) ×			
<pre>&gt;&gt;&gt; 'abc'.find('b') ^</pre>			
1	劫行 'abc' 的		
>>> 万動環境(Shall) ×	括號内填入的 Python 中順序	参考資料 'b' 稱為參數 序編號都是從 0 開始算	
>>> 'abc'.replace('b', 'z') ^			
'azc' 執行 'abc' 的 把 'b' 換成 'z' 方法			
>>>			
互動環境(Shell)× >>> "abc".upper() 'ABC'		'abc' 這種叫做 <b>字串</b> 物件 可以用 'abc' 或是 "abc" 但 <b>頭尾引號</b> 一定要一致	
<pre>&gt;&gt;&gt; "it's a book".upper() "IT'S A BOOK"</pre>			
>>>			

物件有各種不同的類別







# Python内建函式

**函式**可以提供特定的功能 用法跟物件的方法很相似 但是不需要『**物件.**』開頭

Python 提供隨時可用的稱為內建 (built-in) 函式



# **4** PART

# **點亮人生的光明燈** 數位輸出 (Digital Output)



這裡有 2 顆可控 LED





























### 沒有防洪就會變成這樣!



空中攝影/齊柏林 圖片版權/台灣阿布電影股份有限公司



不同攔阻能力的攔砂壩,限流降壓的能力會不同









### GPIO 腳位就是魔術水池



General Purpose Input/Output 中文稱為通用目的的輸入/輸出

這些 GPIO 腳位/插槽可透過程 式控制瞬間變化**輸出高/低電位** 





## 呼叫外部支援 - 控制IO





gpb 模組中的 LED 類別可產生幫你控制LED.







想想看! 怎麼將LED熄滅呢?





### 使用模組提供的函式

Python 本身沒有控制時間的功能 再次呼叫 外部功能 刷剛用到的 gpb 模組 內含有與控制時間的功能





### 寫程式控制時間狀態



但就必須額外冠上模組名稱



執行後請注意看板子上的 LED, 若沒看到可以重新再執行一次



## 點亮後0.5秒熄滅



## 5 **閃爍 LED** Python 的 while 迴圈 PART

### 把前個範例改成閃爍 LED





閃爍LED燈















可參考sample code\_3.py



1	<pre>from gpb import LED</pre>	# 匯入LED類別
2	<pre>from gpb import delay</pre>	# 進人delay函式
3 4	L1 = LED(0)	# 建立物件
5	L2 = LED(1)	
6		
7	while True:	# True代表成立
8	L1.on()	# 點亮L1
9	L2.off()	# 熄滅L2
10	delay(500)	
11	L1.off()	# 熄滅L1
12	L2.on()	# 點亮L2
13	delay(500)	



## 執行後好像沒有反應?

<b>陙</b> Thonny - D:\mee\Dropbox (個人)\Flag\projects\GPlus\Mi		
檔案 編輯 檢社	視 執行 Device 工具 說明	
🗋 💕 🔙 ( 🤅	₩ 選擇直譯器	
demo3.pv ×	de ♥ 執行目前程式	F5
	除錯目前的程式(較佳)	Ctrl+F5
1 +	▶ 除錯目前的程式(較快)	Shift+F5
2 <b>f</b>	▶ 除錯目前程式(birdseye)	Ctrl+Shift+B
3	🥋 跳下一步(Step Over)	F6
	🚴 跳入(Step into)	F7
4 1	e .☆ 跳出(Step out)	
5 b	★ 🗈 繼續	F8
	┉┊執行至游標	Ctrl+F8
6 W		なまませった
7	一 在約4000000000000000000000000000000000000	义里儿 空 市 小 化
8		
9	✔ Pygame Zero模式	
	🔤 停止 / 重新啟動 後端程式	Ctrl+F2
	中斷程式執行	Ctrl+C
	送出EOF / 軟開機	Ctrl+D
	中斷連接	





# 6 PART

# 用按鈕控制LED 數位輸入(Digital Input)

























## 寫程式觀察按鈕狀態

可參考sample code: 4.py

Thonny - C:\Users\LynnLin\Documents\14_GenioPy\06_Project	t\TINKA\V1.2正式\SampleCode\教學資源\4.p 🗖 💷 🗮 🍋
檔案 編輯 檢視 執行 工具 說明	
📑 🖻 🛃 💿 🕸 😳 3et 🗈 🚭	
1.py * × 2.py × 3.py * × 4.py * ×	
1 <b>from</b> machine <b>import</b> Pin	# 匯入Pin類別
2 from gpb import delay	# 匯入delav函式
3	
4 btn = $Pin(2, Pin.IN)$	# 建立Pin物件
5	
6 while True:	# True代表成立
<pre>7 print(btn.value())</pre>	# 印出按鈕的值
8 delay(50)	
互動環境(Shell) ×	
0	
◎ → 沒有按下按扣	
1 按下按钮个队	
	MicroPython (一般)





#### 可參考sample code: 05\_btn.py

沒按

按鈕

不成立

點亮 LED



else:

■■■ 不成立時要做的動作 ■■■ ... ②== 會判斷左邊的值是否等於右邊 判斷式依此得到 True 或是 False







可參考sample code: 07\_btn.py

Ν

Ν















可參考sample code\_btn05.py

```
while True:
   from machine import Pin
                                                                 curr = btn.value()
                                            # 匯入計時器類別
   from gpb import delay,timer
                                                                 if prev == 0 and curr == 1:
3
                                                                    delay(20)
                                            # 建立控制LED的腳位
   led = Pin('C13',Pin.OUT)
4
                                                                    curr = btn.value()
                                            # 熄滅LED
   led.value(0)
                                                                    if curr == 1:
                                              建立讀取按鈕的腳位
   btn = Pin('E7',Pin.IN)
                                            #
 6
                                                                        if led_state == 1:
                                            # 上一次的按鈕狀態
   prev = 0
                                            # 目前按鈕狀態
                                                                            led state = 0
   curr = 0
8
                                                                        else:
9
                                                                            led state = 1
                                              計時喚醒時要執行的
10
   def switch led(t):
                                            #
                                                                        led.value(led state)
                                              使用外面的 prev 彳
11
       global prev
                                              使用外面的 curr 彳
                                                                 prev = curr
12
      global curr
                                            #
                                            # 讀取目前按鈕狀態
13
     curr = btn.value()
      if prev == 0 and curr == 1:
                                            # 之前沒按現在按了
14
                                            # 切換 LED 亮按
15
          led.value(1 - led.value())
                                            # 記錄按鈕狀態
16
       prev=curr
17
                                            # 建立 1 號計時器
18
   led_timer = timer(1)
   led_timer.init(freq=50,callback=switch_led) # 指定每秒唤醒 50 次,也就是每 20 毫秒 1 次
19
20
21
   while True:
                                            # 要 delay 計時器才能運作
22
       delay(1)
```

