**Rabboni with python**

# Python安裝與前置作業

1. 請先至Python官方網站下載最新版python(windows)

<https://www.python.org/downloads/>

1. 使用python 安裝rabboni專屬的library
	* 1. 使用系統管理員開啟命令提示字元(CMD)
		2. 輸入: pip install rabboni==1.73 (目前最新版本1.73)



# C:\Users\Sui\Desktop\raboni\圖片2.pngC:\Users\Sui\Desktop\raboni\ico\icon.png Rabboni介紹

1. 內含6 軸sensor，3-axis加速度與3-axis陀螺儀
	* 1. 規格 : ICM-20689
		2. 可使用Python 藍芽與USB API將值讀出來
2. 內置counter計數功能
	* 1. 可使用Python 藍芽與USB API將值讀出來
		2. 可使用Python 藍芽與USB API 將counter歸零
3. 內建兩種連接模式
	* 1. 藍芽模式 :
			1. 將dongle插上電腦USB
			2. 開啟rabboni，按下藍芽廣播鍵
			3. 執行python藍芽相關code(如example code)
		2. USB模式 :
			1. 將rabboni使用USB連上電腦
			2. 執行python USB相關code(如example code)

# Rabboni 藍芽連接 ─ Python API Example code

1. 以下為藍芽的example code，會做到以下結果
	* 1. 掃描藍芽裝置並連上裝置(會自動檢查裝置是否為Rabboni)
		2. 將連上後的六軸資料印出來
		3. 依照設定的條件重置counter
		4. 將結果儲存成圖檔與csv檔
		5. 斷開連接
2. 相關API請看下一頁 “Rabboni 藍芽連接 ─ python API”

# -\*- coding: UTF-8 -\*-

from rabboni import \*

rabbo = Rabboni(mode = BLE) #先宣告一個物件

rabbo.scan() #掃描所有藍芽Device

rabbo.print\_device() # 列出所有藍芽Device

rabbo.connect("D1:FA:F0:F2:12:29")#依照MAC連接

rabbo.discover\_characteristics()#掃描所有服務 可略過

rabbo.print\_char()#列出所有服務 可略過

# print (rabbo.characteristics)

# print (rabbo.Status)

rabo.set\_sensor\_scale(acc\_scale = 16, gyr\_scale = 2000) ## 設定加速度跟陀螺儀的最大範圍 建議先設定完離線後再連線進行檔案錄製

rabo.set\_count\_threshold(threshold = 2500)## 設定threshold = 2500mg

rabbo.read\_data()#讀取資料 必跑

try:

 while True:#一直打印資料 直到結束程式

 rabbo.print\_data()#print資料

 if rabbo.Cur\_Cnt == 100:

 rabbo.rst\_count(mode = “Both”)

except KeyboardInterrupt:#結束程式

 print('Shut done!')

 print (rabbo.Accx\_list)#印出到結束程式時的所有Accx值

 rabbo.stop()#停止dongle

 rabbo.write\_csv(data = rabbo.Accx\_list,file\_name ="AccX")#將Accx寫出csv檔

 rabbo.plot\_pic(data = rabbo.Accx\_list,file\_name = "AccX",show = True)#將Accx畫出圖案並存檔

finally:

 rabbo.stop()

Rabboni 藍芽連接 ─ python API

|  |
| --- |
| 前置API |
| API名稱 | 解釋 | Return值 |
| Rabboni(mode = “BLE”) | Class 名稱 一開始先宣告一個物件給它 並宣告模式 | None |
| Rabboni.scan() | 掃描附近藍芽device | 所有device的名稱與相關資訊 |
| Rabboni.print\_device() | 印出所有 | None |
| Rabboni.connect(MAC) | 填入MAC，連接device | Device |
| Rabboni.discover\_characteristics() | 抓出device所有的服務 | None |
| Rabboni.print\_char() | 列出所有服務(需先discover) | None |
| Rabboni.disconnect() | 斷開聯絡 | None |
| Rabboni.stop() | 關掉dongle，建議最後都要關掉，以免下次開啟出錯 | None |
| 操作、讀取API |
| API名稱 | 解釋 | 參數解釋 |
| Rabboni.read\_data() | 讀取所有sensor資料，必須先跑此行，所有的參數才會出來 | None |
| Rabboni.print\_data() | 列出read\_data()所讀到的資料Acc\_x,y,zGyr\_x,y,zCount | None |
| Rabboni.rst\_count(mode) | 重置裝置紀錄的count | mode: “Both”,”Store\_cnt”,”Cur\_Cnt”將不同的count進行reset，Default 為 Both |
| Rabboni.set\_sensor\_scale(acc\_scale,gyr\_scale) | 決定sensor量取範圍(+-範圍) | acc\_scale:必填,2,4,8,16gyr\_scale:必填, 250,500,100,2000 請將設定與錄製分開進行，設定完後離線再進行錄製。 |
| Rabboni.set\_count\_threshold(threshold) | 決定Counter的threshold | threshold: int, 單位為mg，ex:2500->2.5g，為3軸平方開根號的threshold請將設定與錄製分開進行，設定完後離線再進行錄製。 |
| Rabboni.write\_csv(data,file\_name) | 將data(list)傳進去寫出csv檔案，並以file\_name命名 | data:必填，資料file\_name:必填，檔名 |
| Rabboni.plot\_pic(data,file\_name,show) | 將data(list)傳進去畫出圖片，並畫圖存起來 | data:必填，資料file\_name:選填，檔名，若沒傳入值則不會存檔show : 預設為True會顯示圖片，False則不會顯示圖片 |
| 參數API |
| API名稱 | 解釋 | 參數解釋 |
| Rabboni.Status | 是否連接上 | 0:disconnected1:connected |
| Rabboni.Hex\_data | 當下sensor data的16進位值 | string |
| Rabboni.Accx | 當下Accx的值 | float |
| Rabboni.Accy | 當下Accy的值 | float |
| Rabboni.Accz | 當下Accz的值 | float |
| Rabboni.Gyrx | 當下Gyrx的值 | float |
| Rabboni.Gyry | 當下Gyry的值 | float |
| Rabboni.Gyrz | 當下Gyrz的值 | float |
| Rabboni.Cur\_Cnt | 當下Cur\_Cnt的值 | int |
| Rabboni.Store\_Cnt | 裝置中紀錄的Store\_Cnt | Int |
| Rabboni.Acc\_char | 目前Acc (量測)的範圍 | int |
| Rabboni.Gyr\_char | 目前Gyro (量測)的範圍 | int |
| Rabboni.Accx\_list | 紀錄連接後直到結束的Accx  | list |
| Rabboni.Accy\_list | 紀錄連接後直到結束的Accy  | list |
| Rabboni.Accz\_list | 紀錄連接後直到結束的Accz  | list |
| Rabboni.Gyrx\_list | 紀錄連接後直到結束的Gyrx  | list |
| Rabboni.Gyry\_list | 紀錄連接後直到結束的Gyry  | list |
| Rabboni.Gyrz\_list | 紀錄連接後直到結束的Gyrz  | list |
| Rabboni.Cnt\_list | 紀錄連接後直到結束的Cur\_cnt  | list |

# Rabboni USB 連接 ─ Python API Example code

1. 以下為USB的example code，會做到以下結果
	* 1. 連上裝置
		2. 將連上後的六軸資料印出來
		3. 依照設定的條件重置counter
		4. 斷開連接
2. 相關API請看下一頁 “Rabboni USB連接 ─ python API”

from rabboni import \*

rabbo = Rabboni(mode = "USB") #先宣告一個物件

rabbo.connect()#連結上rabboni，若沒插上會報錯

print ("Status:",rabbo.Status)

rabbo.set\_sensor\_scale(acc\_scale = 4, gyr\_scale = 2000) ## 設定加速度跟陀螺儀的最大範圍 建議先設定完離線後再連線進行檔案錄製

rabo.set\_count\_threshold(threshold = 2500)## 設定threshold = 2500mg

try:

 rabbo.read\_data()

 while True:#一直打印資料 直到結束程式

 rabbo.print\_data()#print資料

 print (rabbo.data\_num)

 if rabbo.Cur\_Cnt > 10:

 rabbo.rst\_count() #重置count 會delay一下

 if rabbo.data\_num>100:

 rabbo.stop()#停止運作

 break

except KeyboardInterrupt:#結束程式

 print('Shut done!')

 # print (rabbo.Accx\_list)#印出到結束程式時的所有Accx值

 rabbo.stop()#停止運作

Rabboni USB連接 ─ python API

|  |
| --- |
| 前置API |
| API名稱 | 解釋 | Return值 |
| Rabboni(mode = “USB”) | Class 名稱 一開始先宣告一個物件給它 並宣告模式 | None |
| Rabboni.connect() | 透過USB連接device，如果抓不到裝置會報錯 | None |
| Rabboni.disconnect() | 斷開聯絡，關掉USB | None |
| Rabboni.stop() | 斷開聯絡，關掉USB | None |
| 操作、讀取API |
| API名稱 | 解釋 | 參數解釋 |
| Rabboni.read\_data() | 讀取所有sensor資料，必須先跑此行，所有的參數才會出來 | None |
| Rabboni.print\_data() | 列出read\_data()所讀到的資料Acc\_x,y,zGyr\_x,y,zCount | None |
| Rabboni.rst\_count(mode) | 重置裝置紀錄的count | mode: “Both”,”Store\_cnt”,”Cur\_Cnt”將不同的count進行reset，Default 為 Both |
| Rabboni.set\_sensor\_scale(acc\_scale,gyr\_scale) | 決定sensor量取範圍(+-範圍) | acc\_scale:必填,2,4,8,16gyr\_scale:必填, 250,500,100,2000 請將設定與錄製分開進行，設定完後離線再進行錄製。 |
| Rabboni.set\_count\_threshold(threshold) | 決定Counter的threshold | threshold: int, 單位為mg，ex:2500->2.5g，為3軸平方開根號的threshold請將設定與錄製分開進行，設定完後離線再進行錄製。 |
| Rabboni.write\_csv(data,file\_name) | 將data(list)傳進去寫出csv檔案，並以file\_name命名 | data:必填，資料file\_name:必填，檔名 |
| Rabboni.plot\_pic(data,file\_name,show) | 將data(list)傳進去畫出圖片，並畫圖存起來 | data:必填，資料file\_name:選填，檔名，若沒傳入值則不會存檔show : 預設為True會顯示圖片，False則不會顯示圖片 |
| 參數API |
| API名稱 | 解釋 | Return值 |
| Rabboni.Status | 是否連接上 | 0:disconnected1:connected |
| Rabboni.Hex\_data | 當下sensor data的16進位值 | string |
| Rabboni.Accx | 當下Accx的值 | float |
| Rabboni.Accy | 當下Accy的值 | float |
| Rabboni.Accz | 當下Accz的值 | float |
| Rabboni.Gyrx | 當下Gyrx的值 | float |
| Rabboni.Gyry | 當下Gyry的值 | float |
| Rabboni.Gyrz | 當下Gyrz的值 | float |
| Rabboni.Cur\_Cnt | 當下Cur\_Cnt的值 | int |
| Rabboni.Store\_Cnt | 裝置中紀錄的Store\_Cnt | Int |
| Rabboni.Acc\_char | 目前Acc (量測)的範圍 | int |
| Rabboni.Gyr\_char | 目前Gyro (量測)的範圍 | int |
| Rabboni.Accx\_list | 紀錄連接後直到結束的Accx  | list |
| Rabboni.Accy\_list | 紀錄連接後直到結束的Accy  | list |
| Rabboni.Accz\_list | 紀錄連接後直到結束的Accz  | list |
| Rabboni.Gyrx\_list | 紀錄連接後直到結束的Gyrx  | list |
| Rabboni.Gyry\_list | 紀錄連接後直到結束的Gyry  | list |
| Rabboni.Gyrz\_list | 紀錄連接後直到結束的Gyrz  | list |
| Rabboni.Cnt\_list | 紀錄連接後直到結束的Cur\_Cnt  | list |
| Rabboni.data\_num | 上述的list總共有幾筆(單位為sampling數) | int |