

無人機探索與應用

組別:第二組

授課老師:范崇碩老師

組員:

410260050 化生二 陳俞方

410415025 電機二 潘弘恩

410736016 運競二 彭郁芯

411410070 資工一 王俐晴

411410097 資工一 黃崇信

411422021 機械一 何智桓

目錄:

- 一、Codrone 概述
- 二、Windows 和 Mac OS 安裝 python
- 三、安裝與設定 MacOS 版本的 PyCharm
- 四、Codrone 介紹影片
- 五、編程
- 六、飛行動作
- 七、變數
- 八、條件式
- 九、迴圈
- 十、飛行關卡設計與撰寫程式

一、 CoDrone 概述

1. CoDrone 的組件

- (1) CoDrone 板*1
- (2) 螺旋槳防護罩*2
- (3) CoDrone 外殼*2
- (4) 螺旋槳*4
- (5) 馬達*4

2. 組合步驟

- (1) 把 CoDrone 板與底部外殼接合
- (2) 頂殼與底殼上的接腳對齊並推在一起
- (3) 手臂插入馬達且固定完全
- (4) 主體連結螺旋槳 (前螺旋槳向內旋轉，後螺旋槳向外旋轉)
- (5) 接上防護罩



3. 安裝驅動程式

- (1) 先進入到 <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers> 下載與系統相符的驅動程式
- (2) 檔案存於桌面
- (3) 將解壓縮檔放入新的資料夾，再安裝 'CP210xVCPInstaller'

二、 Windows 和 Mac OS 安裝 python

1. Windows 版本

- (1) 先進入 <https://www.python.org/> 下載 windows 版本的 python
- (2) Download >> View the full list of downloads. >> 選擇最新版 Python
>> 下載 Python.exe 資料檔案 >> 開啟
- (3) 若是 win 10 版本，Install anyway >> Install now >> Add Python (your version here) to PATH >> Close >> 完成下載

2. Mac OS 版本

- (1) 先進入 <https://www.python.org/> 下載 Mac OS 版本的 python
- (2) Download >> View the full list of downloads. >> 選擇最新版 Python
>> 下載 Python.exe 資料檔案 >> 開啟
- (3) 下載的資料夾 >> python-(你下載的版本)-macosx10.9.pkg >>
點兩下開啟>>一直按 Continue >> Install >> close

三、安裝與設定 MacOS 版本的 PyCharm

1. 安裝 PyCharm Edu

- (1) 搜尋 PyCharm Edu >> Download >> 轉換至 applications >> 選擇 Learners>>完成

2. PyCharm Edu 環境設定

- (1) Create new project >> 檔名:Test >> 選擇 python3.8
- (2) 左上角 Pycharm >> Preferences >> Project:Test >> Project Interpreter
>> + >> 搜尋 "codrone" >> install >> close >> ok

3. 飛行測試

- (1) 建立一個新的 Python File，檔名命為 take off >> 輸入以下程式

```
import CoDrone
drone = CoDrone.CoDrone()
drone.pair()
drone.takeoff()
drone.hover()
drone.land()
drone.close()
```

四、CoDrone 介紹影片

1. 安全性

- (1) 拿取無人機時不要由頂部抓它 > 螺旋槳會碰到手指
- (2) 操作前要先把頭髮綁起來 > 避免頭髮捲進馬達
- (3) 小心使用藍芽晶片
- (4) 避免無人機碰到水

2. 在教室操作無人機的注意事項

- (1) 在課前預習操作步驟
- (2) 準備器材(無人機、電腦、USB、電池)
- (3) 確認 code 是否正確
- (4) 可以在手上測試無人機的飛行

五、 編程

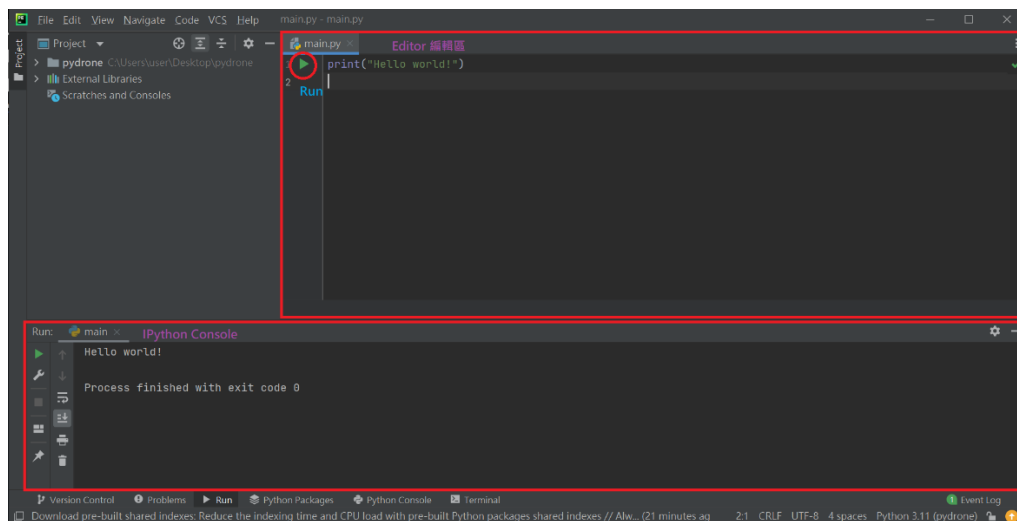
1. 簡介

本次無人機課程使用的是 **PyCharm Edu** 這個程式，而此程式使用的直譯器即是 **Python**，以下便會簡單介紹 **Python** 的程式編成。

首先是基本上每位程式設計初學者都會看到的：

```
print("Hello world")
```

這是 **Python** 初學者最基本的程式碼，其意義即是將 “Hello world” 這行文字"印"(print)出來，這裡的印並不是真的印出來，而是顯示在下方的顯示框內。



如圖所示，由上的方框為編輯區，一般都是由此輸入程式碼的，再來可以看到下方的方框，這裡便會顯示程式運行的結果，而在 **PyCharm Edu** 這個軟體只要按下綠色三角形 **Run** 即會執程式。

2. Print 的用法

Print 可以印出任意的文字、句子

```
print("輸入想要的文字")
```

```
print("CoDrone is good")
```

```
print("Maybe I can't edit videos")
```

同樣的，只要在編輯區輸入程式碼，按下 **Run** 鍵，即會一行一行的執程式 (**Python** 會幫你自動換行)。

接著我們介紹一些 **print** 函式支援的程式碼

```
print("\t 空三格")      #\t==空三格
```

```
print("這是\n 斷句")    #\n==enter(斷句)
```

此外 **#** 和 **'** 是 **Python** 寫留言的方式，一個是單行留言，一個是多行留言，**Python** 並不會執行留言

```
#print("Python 不會執行這行")
print("Python 會執行這行")
'''
print("A")
print("B")
'''
print("C")
```

3. 進入 Codrone

在了解 Python 的基本後，我們要開始進入 Codrone 的世界。

```
import CoDrone
```

這行代表將 CoDrone(請注意大小寫)的資料庫帶進這個程式，我們將可以用 CoDrone 資料庫裡的所有函式

加入 CoDrone 資料庫後，請創建一個使用在數據集結變數與函式的新目標。

請創建“drone(無人機)”如以下編寫內容：

```
drone = CoDrone.CoDrone()
```

這行代表

(1)建立一個變數 drone

(2)將 CoDrone 資料庫裡的 CoDrone 函式加入到 drone

drone 就像是個溝通的橋樑，接下來就會以 drone 這個變數來跟 CoDrone 溝通

現在將無人機與 BLE 板(低功耗藍芽板)連接上

```
drone.pair(drone.Nearest)
```

這會讓板子與無人機連接上，這樣我們便能執行一連串的動作

接下來請加入一些特別的列印語句到程式碼來練習！

```
import CoDrone
print("Creating drone object")
drone = CoDrone.CoDrone()
print("Getting ready to pair")
drone.pair(drone.Nearest)
print("Paired!")
```

如果配對上，你的 BLE LED 燈和 CoDrone 尾部 LED 燈會出現穩定不閃爍綠燈。

4. Take off 起飛囉

在配對完成後，我們便可以開始執行動作(請確保無人機周圍是安全的)

寫下 `drone.takeoff()`指令可以讓 CoDrone 起飛並盤旋

```
drone.takeoff()
```

接下來，我們可以設定盤旋的時間

函式 `drone.hover()`的參數是你希望無人機在空中盤旋的時間。

```
drone.hover(4)    #盤旋 4 秒
```

最後別忘記要著陸

```
drone.land()    #著陸
```

在運作完整的程式碼前:

(1)請確認 BLE 板有閃爍紅燈，如否請按下 LED 燈下的重置鈕。

(2)CoDroen 進入配對模式後請中斷電池連線並按下藍芽板上的重置鈕。

若想要編輯程式碼，無須與無人機斷開連線，只要在編輯區更改程式碼，再按 Run 鍵即可讓無人機執行新的指令。

六、 飛行動作

1. 簡介

(1) Roll(橫滾)->左右移動

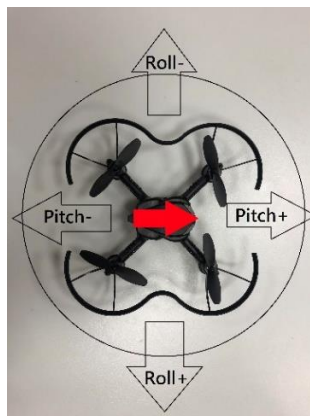
Positive roll(正向橫滾)是把 CoDrone 向右移動，反之，negative roll(負向橫滾)是把 CoDrone 向左移動

```
drone.set_roll(power)    #power can be between -100~100
```

Pitch(俯仰)->前後傾斜

Positive pitch(正向俯仰)是讓 CoDrone 向前傾斜，negative pitch(負向俯仰)是讓 CoDrone 向後傾斜

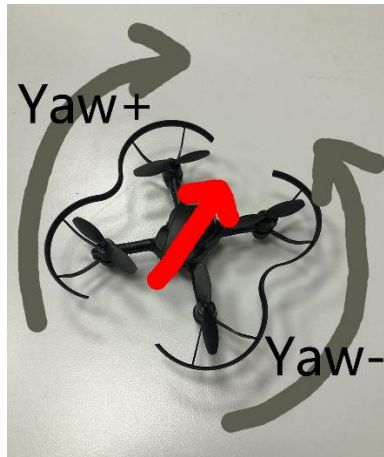
```
drone.set_pitch(power)    #power can be between -100~100
```



(2) Yaw(偏航)->左右旋轉

Positive yaw(正向偏航)讓 CoDrone 向右轉；而 negative yaw (負向偏航)是向左旋轉

```
drone.set_yaw(power)    # power can be between -100~100
```



(3) Throttle(油門)->上升下降

Positive throttle(催油門)會讓 CoDrone 往上飛高；而 negative throttle(收油門)讓 CoDrone 往下飛低

```
drone.set_throttle(power)    # power can be between -100 ~ 100
```



2. 使用飛行指令

如第一節所述，首先需要準備動作

```
import CoDrone
```

```
drone = CoDrone.CoDrone()
```

```
drone.pair(drone.Nearest)
```

```
drone.takeoff()
```

接下來就可以插入函式

```
import CoDrone
```

```
drone = CoDrone.CoDrone()
```



```

drone.pair(drone.Nearest)
# drone.pair() if paired to CoDrone before

drone.takeoff()          # takeoff for 2 seconds

drone.set_pitch(30)      # Set positive pitch to 30% power
drone.move(2)            # forward for 2 seconds

drone.set_yaw(-50)       # Set negative yaw to 30% power
drone.move(2)            # turn left for 2 seconds

drone.land()             # lands the CoDrone
drone.close()            # disconnects CoDrone

```

每個動作都包含兩行程式碼，一個是"準備"朝特定方向與速度移動，而 `drone.move()` 是讓 CoDrone 實際在空中移動

3. 同時使用多組指令

在指令改回 0 之前，CoDrone 在接收指令後會永久為該方向設定這個值，這也是為什麼 CoDrone 可以同時向多個方向飛行的原因

我們來看個範例

```

import CoDrone

drone = CoDrone.CoDrone()
drone.pair(drone.Nearest)
# drone.pair() if paired to CoDrone before

drone.takeoff()          # takeoff for 2 seconds
#two movement at the same time
drone.set_pitch(30)      # Set positive pitch to 30% power
drone.set_roll(-30)      # Set negative roll to 30% power
drone.move(2)            # forward and right for 2 seconds

drone.land()             # lands the CoDrone
drone.close()            # disconnects CoDrone

```

4. 函式 drone.go()

drone.go(direction, duration, power)

三個變數分別代表:方向(direction)、持續時間(duration)、動力(power)

(1) direction

可以用[FORWARD(向前)、BACKWARD(向後)、LEFT(向左)、RIGHT(向右)、UP(向上),和DOWN(向下)]表示，但請記得如上要輸入大寫

(2) duration

以秒為單位去控制無人機，可以留白、設定 0 或其他正值

0=無限持續運作

沒有輸入數字=1 秒

任意正值=持續時間

(3) power

無人機的飛行速度，可以輸入 0~100，若留白則會預設 50

再來要注意的是，使用 drone.go()需要在 import CoDrone 加入以下文字

```
from CoDrone import Direction
```

以下是範例

```
drone.go(UP, 3, 75)    # go up for 3 seconds at 75% power
drone.go(DOWN)         # go down for 1 second at 50% power
drone.go(LEFT, 6)      # go left for 6 seconds at 50% power
```

七、變數

1. 簡介

前面我們有提到變數這個詞，這在每個程式都是極度重要的一項元素

變數就像是一個儲存框，首先要設定或宣告一個變數，並初始化他的值，之後便可以在編成部分呼叫使用

```
abc = 10 #Declaring variable abc and initializing the value to 10
print(abc) #Calling the variable abc
```

若在後面改變變數的值，則變數的新值會將舊值覆蓋，因此請注意所有變數的取名

```
x=2      #x=2
print(x) #print 2
x=3      #x=3
print(x) #print 3
```

變數之間也可以互相溝通，支援加減乘除等運算

```
x=2
y=2+x #y=4
```

```
z=x+y #z=6
```

2. 變數的種類

基本可以分為幾個值

```
w = 4                # this is an int
x = 3.2              # this is a float
phrase = "Hello"     # this is a string
is_flying = True     # this is a boolean
```

int:正負整數

Float:十進制的浮點數(在表現上類似於小數，但又不全然是小數)

String:字串，一排的字符

Boolean:可以設定' True' 和' False'

Python 會自動設定該值的類別，因此我們無須指定數據的類型

接著我們可以列印看看變數

```
a=1
b=1.5
c=9487
phrase = CoDrone is good ?
print(a)
print(b)
print(c)
print(phrase)
```

3. 變數加入 CoDrone

在了解變數的用途後，我們便能將變數加入 CoDrone 編程了

```
duration=5
drone.takeoff()
drone.hover(duration)    # in this case, hover for 5 seconds
drone.set_pitch(50)      # Set positive pitch to 50% power
drone.move(duration)     # forward for 5 seconds
drone.land()
drone.close()
```

每一個由數字或字符組成的變數都可被函式所使用，例如前面 `drone.go()` 的 FORWARD，如果不想每次打都這麼長一串，可以給他一個新的代表名稱

```
f=FORWARD    #f represent FORWARD
drone.go(f)
```

要注意的是，變數一定要在使用變數前宣告，不存在使用後才宣告。此外，強烈建議變數使用有意義的名稱，不然當變數一多會不好處理

```
#variable represent speed
```

```
a = 10 X
```

```
speed = 10 O
```

八、條件式

1. 簡介

如果…就…是條件式的精神，接下來看一下 if 語句如何呈現

```
if alarm goes off:
```

```
    I should wake up
```

```
else:
```

```
    I can stay asleep
```

```
#這只是範例，電腦跑不動這幾行程式碼
```

請注意在每個條件後都需加上：且欲執行的程式碼盡量縮排處理

條件式讓電腦來分辨是 **true**(真)與 **false**(否)，若為 **true** 則執行之下的語句，若為 **false** 則不會執行之下的語句。**"Else"**則是上述條件皆不滿足，則會執行 **Else** 下面的語句，當然也可以不使用 **else**，執行時會執行該行以下所有有縮排的程式碼

```
drones = 100
```

```
bugs = 55
```

```
if drones > bugs:
```

```
    print("Squish the bugs!")
```

```
if drones + bugs > 90:
```

```
    print("Run away!")
```

這時有人可能會問：如果我想要同時對一件事執行多重條件判斷該怎麼辦。這時就是 **"elif"**登場的時候了

```
if Today is Saturday:
```

```
    I don't have to go to school
```

```
elif Today is Sunday:
```

```
    I don't have to go to school
```

```
else :
```

```
    I have to go to school
```

```
#這只是範例，電腦跑不動這幾行程式碼
```

2. 比較算符

條件式常用的是一些兩兩比較的運算符，像是><，若條件符合是 True，反之則為 False
常見的比較算符：

‘>’ 大於
‘<’ 小於
‘>=’ 大於或等於
‘<=’ 小於或等於
‘==’ 等於
‘!=’ 不等於

由於單等號在程式的世界被用作"將左方的資料存入右方"，因此在條件判斷的部分是用雙等號來代表。

大家可以帶入以下範例來實際操作

```
x = 58
y = 12
z = 90
if x > y:
    print("jelly")
if z > x:
    print("peanut butter")
if y+z > x:
    print("bread")
if y > z:
    print("toast")
else :
    print("best sandwich ever")
```

3. 巢狀 if

其實就是在 if 內又使用 if，當需要同一個目的，並遇到多重條件式或是其之下的問題，使用巢狀式 if 語句就很方便。

```
if it is sunny outside:
    if it is hot:
        I will go swimming
    else:
        #it is NOT hot
        I will play baseball
else:
    #it is NOT sunny
    I will play inside
```

九、迴圈

迴圈便是重複執行一段程式碼，讓同樣的東西不需要寫第二遍，以下會介紹常用的 While、for 迴圈

1. While

當 While 迴圈的條件是 True，則會執行他之下所有縮排的程式碼，反之則會略過

```
battery = 100
while battery > 0: #battery > 0 是 while 迴圈運作的條件
    print("Using...")
    if battery == 10:
        print("Low battery")
        battery = battery-10
print("Dead battery")
```

若條件為 1 或 True，則會無限執行 while(請盡量不要這麼做，因為程式會一直跑)

接下來可以看 while loop 在 CoDrone 的應用

```
#讓 CoDrone 做
import CoDrone 伏地挺身。只有變數超過 10 才讓條件式為 true(是)

drone = CoDrone.CoDrone()
drone.pair(drone.Nearest)

pushups = 0
while pushups < 10:
    drone.takeoff()
    drone.hover(1)
    drone.land()
    pushups+=1 # increment pushups by 1
drone.close()
```

2. For loops

和 While loop 類似，但可以在起始便設定初始值、結束值、與增減值

```
for variable in range (minimum, maximum, increment or decrement):
    do something
```

看個範例應該就能理解 for loop 的原理了

```
for x in range (0,10,1):
    print(x)
```

變數 x 會從 0 開始，以 1 遞增，直到 x 變為 10。因此其輸出應為

```
0
1
2
3
4
5
6
7
8
9
```

可以發現 10 並沒有被印出來，因為 10 是結束的條件，一旦達到最大值他就不會被列印出或超出限制。

接下來可以看看 CoDrone 使用 For loop，讓 CoDrone 在方形中飛行

```
import CoDrone
drone = CoDrone.CoDrone()
drone.pair(drone.Nearest)

drone.takeoff()

for x in range(0, 4, 1):
    drone.set_pitch(50)
    drone.move(2)
    drone.set_yaw(-50)
    drone.move(2)

drone.land()
drone.close()
```

十、飛行關卡設計與撰寫程式

1. 海盜船

```
import CoDrone
v = 50
i = 5
drone = CoDrone.CoDrone()
drone.pair(drone.Nearest)
# drone.pair() if paired to CoDrone before
drone.takeoff()    # takeoff for 2 seconds
while i > 0:
    while v > -1: #v=-1 為 while 的終止值
        drone.set_roll(v)    # 向右移動
        drone.set_throttle(v)    # 向上移動
        drone.move(0.01)
        v -= 1
    #此時無人機向右上方向作減速度運動直到 V = 0，跳出迴圈並進入下一個迴圈
    v = 0    #此時將 v 改為 0，以免使下一個迴圈多跑一次
    while v < 51: #v=51 為 while 的終止值
        drone.set_roll(-v)    # 向左移動
        drone.set_throttle(-v)    # 向下移動
        drone.move(0.01)
        v += 1
    #此時無人機向左下方向作加速度運動直到 V = 51，跳出迴圈並進入下一個迴圈
    v = 50    #此時將 v 改為 50，以免使下一個迴圈多跑一次
    while v > 0: #此 v=0 為 while 的終止值
        drone.set_roll(-v)    # 向左移動
        drone.set_throttle(v)    # 向上移動
        drone.move(0.01)
        v -= 1
    #此時無人機向左上方向作減速度運動直到 V = 0，跳出迴圈並進入下一個迴圈
    v = 0    #此時將 v 改為 0，以免使下一個迴圈多跑一次
    while v < 51: #v=51 為 while 的終止值
        drone.set_roll(v)    # 向右移動
        drone.set_throttle(-v)    # 向下移動
```



```

        drone.move(0.01)
        v += 1
#此時無人機向右下方向作加速度運動直到 V = 51，跳出迴圈並進入下一個迴圈
        v = 50    #此時將 v 改為 50，以免使下一個迴圈多跑一次
        i -= 1
#i 每次減 1，共跑五次
drone.land()          # lands the CoDrone
drone.close()         # disconnects CoDrone

```

2. BALLET DANCE

```

import CoDrone

void setup() {
    CoDrone.begin(115200);
    CoDrone.pair();          #連接至無人機

    CoDrone.takeoff();       #起飛

    CoDrone.setThrottle(25);  #向上飛
    CoDrone.move(1.5);        #1.5 秒後在執行下一個動作

    for x in range(0, 3, 1):  #執行舞步三次
        CoDrone.setPitch(30);  #前行
        CoDrone.setyaw(100);   #旋轉

        CoDrone.setPitch(30);  #前行
        CoDrone.setThrottle(25); #向上飛
        drone.hover(0.5);      #停留 0.5 秒
        CoDrone.setThrottle(0); #回到遠本高度
        CoDrone.land();        #降落
    }

#關卡描述:
#讓無人機前進幾步做一次旋轉，並重複動作三次，以一次跳躍做為結尾，結束表演，
而後向下降落

```