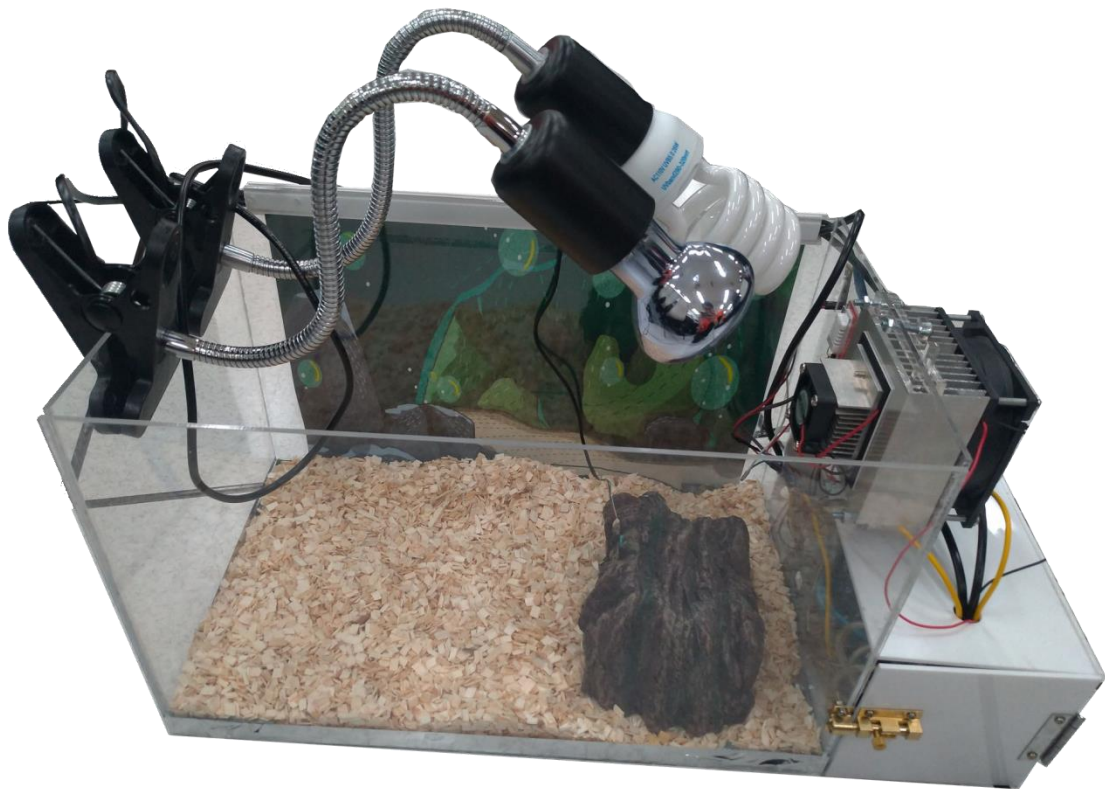


臺北市立大安高級工業職業學校專題製作競賽
「專題組」作品說明書



群別：電機與電子群

作品名稱：爬爬的溫馨之家

關鍵詞：影像辨識、環境控制

目錄

壹、摘要.....	1
貳、研究動機.....	1
參、主題與課程之相關性或教學單元之說明.....	2
一、雷射切割.....	2
二、3D 列印.....	3
肆、研究方法.....	3
一、研究流程.....	3
(一)、研究步驟.....	3
(二)、操作步驟.....	5
二、用材料及工具.....	6
(一)、零件介紹.....	6
(二)、軟體介紹.....	11
伍、研究結果.....	14
一、溫濕度感測.....	14
二、溫溼度控制.....	15
三、手機介面.....	15
四、影像辨識.....	15
陸、討論.....	17
一、溫溼度感測.....	17
二、影像辨識.....	17
三、元件擺放.....	17
柒、結論.....	18
捌、參考資料及其他.....	19

圖目錄

圖 1 影像辨識及爬蟲箱.....	1
圖 2 雷雕機.....	2
圖 3 RDWorks V8 logo.....	2
圖 4 Tinkercad 及元件 3D 圖.....	3
圖 5 研究步驟.....	4
圖 6 環境設定流程圖.....	5
圖 7 元件動作流程圖.....	6
圖 8 電源供應器.....	6
圖 9 ESP8266.....	7

圖 10 紫外線感測器(UV 光譜感測器).....	7
圖 11 取暖 UVA 汝燈泡.....	8
圖 12 UVB 燈	8
圖 13 DHT22 溫溼度感測器	9
圖 14 繼電器.....	9
圖 15 霧化噴霧片.....	10
圖 16 攝像頭 W2FN5-CB11B.....	11
圖 17 App Inventor logo	11
圖 18 Arduino 程式撰寫.....	12
圖 19 Azure Custom Vision logo	12
圖 20 Rhino6 Azure Custom Vision TF 檔輸出	13
圖 21 TensorFlow 程式.....	13
圖 22 Python logo	14
圖 23 辨識條件.....	14
圖 24 溫溼度感測位置.....	15

表目錄

表 1 時間分配表.....	4
表 2 電源供應器規格.....	6
表 3 ESP8266 規格.....	7
表 4 紫外線感測器(UV 光譜感測器)規格	7
表 5 取暖 UVA 汝燈泡規格	8
表 6 UVB 燈	8
表 7 DHT22 規格	9
表 8 繼電器規格.....	9
表 9 霧化噴霧片規格.....	10
表 10 攝像頭 W2FN5-CB11B 規格	10
表 11 元件比較.....	17

【爬爬的溫馨之家】

壹、摘要

在現在養寵物的風潮下，飼養爬蟲類逐漸變得熱門，也因為這個關係，照顧爬蟲類的商品也出現在市面上，光是爬蟲箱就有好幾種可以選。因此，為了給爬蟲類充分的照顧，本專題希望能結合現在熱門的影像辨識技術(如圖 1 左)，做出能感測並自動控制環境溫濕度和紫外線的爬蟲箱(如圖 1 右)。箱子主要的功能是透過攝像頭拍攝放入爬蟲箱的爬蟲類，透過樹莓派上放入的影像辨識系統，來辨識物種的不同，研究中測試三種物種，有蛇、守宮及烏龜，再透過序列埠傳送辨識到的資訊，通知 ESP8266，並且在手機上顯示出當下環境的溫溼度、紫外線光照強度以及被辨識成功的物種，還有它的環境設定值，自動控制箱內的溫濕度及紫外線，在幾乎全自動操作下，讓爬蟲類有適合居住的環境。

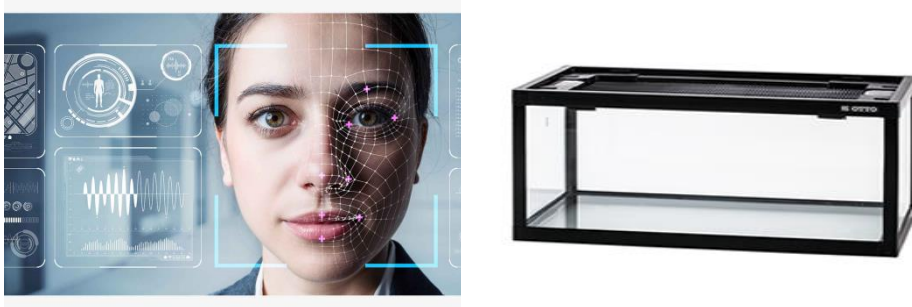


圖 1 影像辨識及爬蟲箱

貳、研究動機

很多人在家裡都有養寵物，最常見的就是貓和狗，而爬蟲類也是現代人的熱門選擇之一。在飼養方面爬蟲類相較於其他寵物，對環境有較嚴格的要求，如果是在同時需要飼養很多不同種類的場所，例如：寵物醫院、爬蟲店、生物研究所等等，都需要針對不同的物種來調整他們的生活環境，而這樣繁雜的照顧程序，卻必須要不斷重複。我們有組員在爬蟲店打工，雖然他每天都需要花大把時間來照顧他們，但對他來說，照顧它們依然是很開心的事，但久了之後，他因為課業壓力，需要壓縮照顧的時間，也因此照顧起來不像之前那麼游刃有餘，也沒辦法好好的培養感情，讓他非常苦惱。所以我們就想透過這次的專題，做一個可以透過影像辨識，辨識爬蟲類，然後自動控制環境溫溼度和紫外線光的爬蟲箱，除了可以減少照顧起來的辛苦，箱內的寵物也能得到良好的照顧。

參、主題與課程之相關性或教學單元之說明

一、雷射切割

雷射切割系統通過計算機實現對激光數控機床的有效控制，能根據使用者不同的需求完成不同任務。我們使用的軟體是 RDWorks V8，能藉由不同顏色的線條分別設計出對於材料不同的要求，有切割、掃描以及打孔三種功能。在專題中我們使用此軟體的切割和打孔來設計硬體架構，完成並調整適當的外加功率。最後在雷雕機(如圖 2)定位好，裁切我們硬體架構的主體(如圖 3)。



圖 2 雷雕機

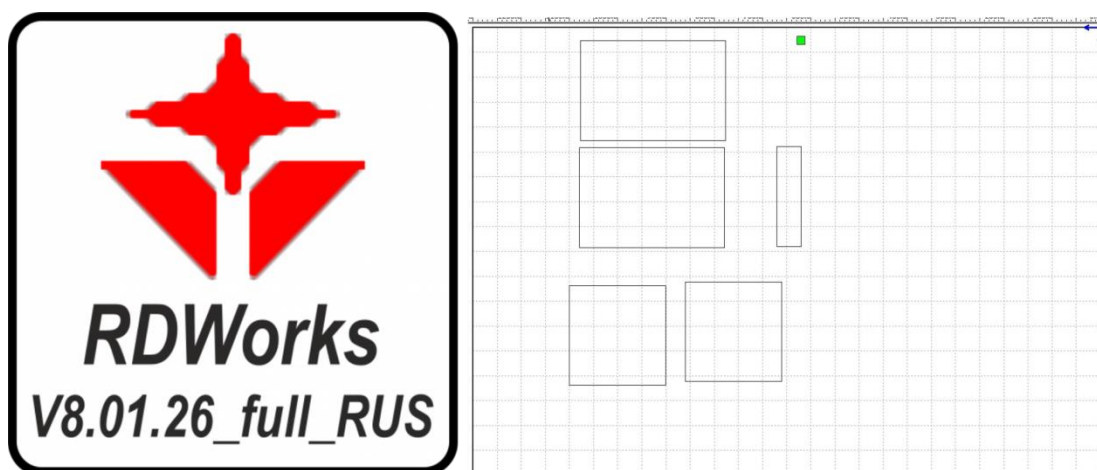


圖 3 RDWorks V8 logo

二、3D 列印

高三的專題製作課當中我們學會使用 Tinkercad 來建立 3D 模型(如圖 4)，以及學習 3D 列印機的使用，因為 3D 列印機幾乎什麼都能夠列印所以運用範圍十分的廣泛，並且能在短暫時間內就影印完，我們看中了他的高可塑性和製作時間短，於是使用他來製造些比較精細的小零件，是非常便捷的硬體建構方式。

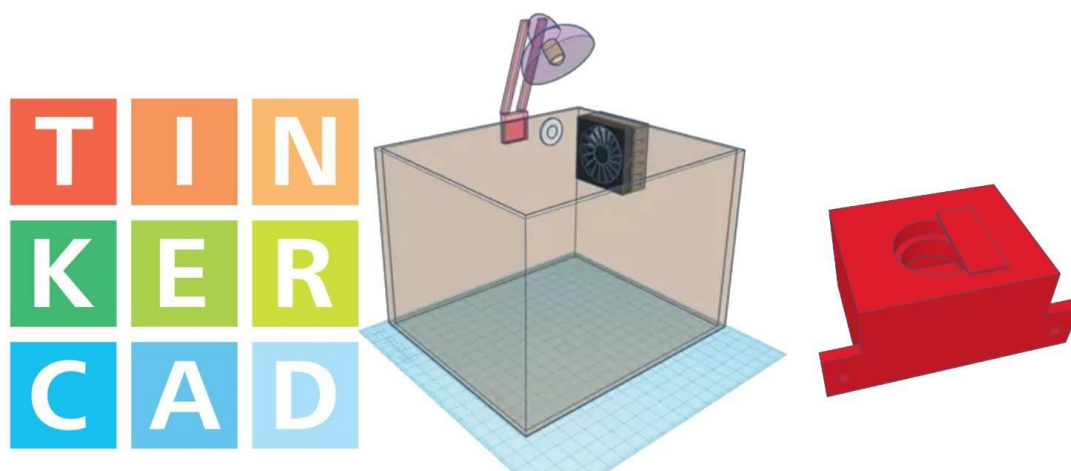


圖 4 Tinkercad 及元件 3D 圖

肆、研究方法

一、研究流程

(一)、研究步驟

在七月初定下專題題目後，且與指導老師討論過後便開始我們的製作流程。我們首先從影像辨識的部分開始著手，須先建立好影像辨識所需的資料庫以及安裝環境，同時也撰寫環境控制的程式，接著開始製作爬蟲箱，著手為爬蟲類打造一個良好的環境，最後將程式與硬體做最終整合以及外觀的修飾，完成專題成品。專題的時間分配及研究步驟分別如下表 1 及圖 5：

表 1 時間分配表

	7 月	8 月	9 月	10 月	11 月	12 月	1 月
1.購買材料							
2.蒐集資料							
3.程式設計							
4.操控介面							
5.爬蟲箱製作							
6.箱內裝飾							
7.成品測試							

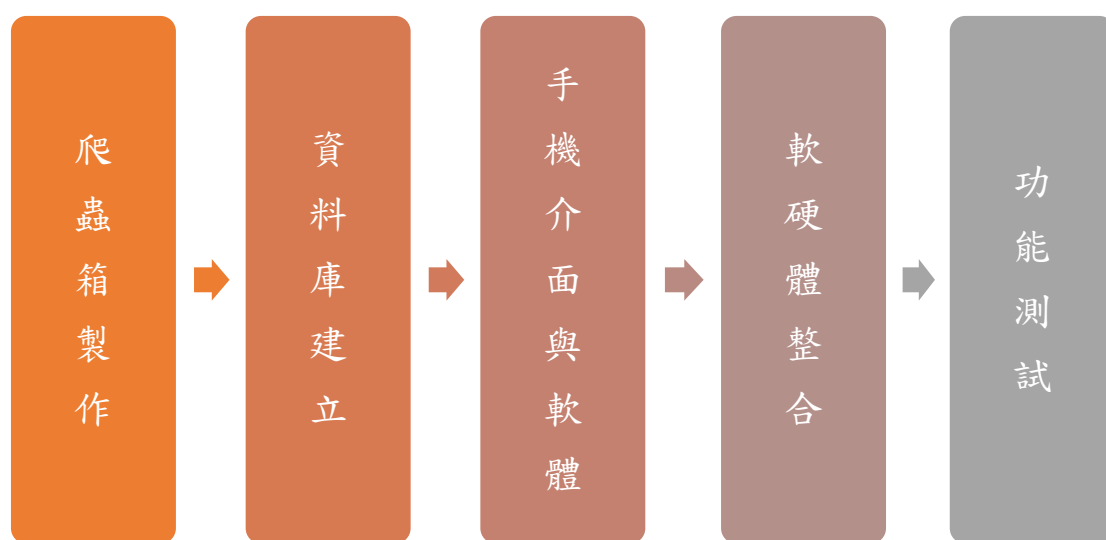


圖 5 研究步驟

(二)、操作步驟

1、環境設定程式

透過影像辨識來控制箱內環境，其動作流程如下圖 6 所示。

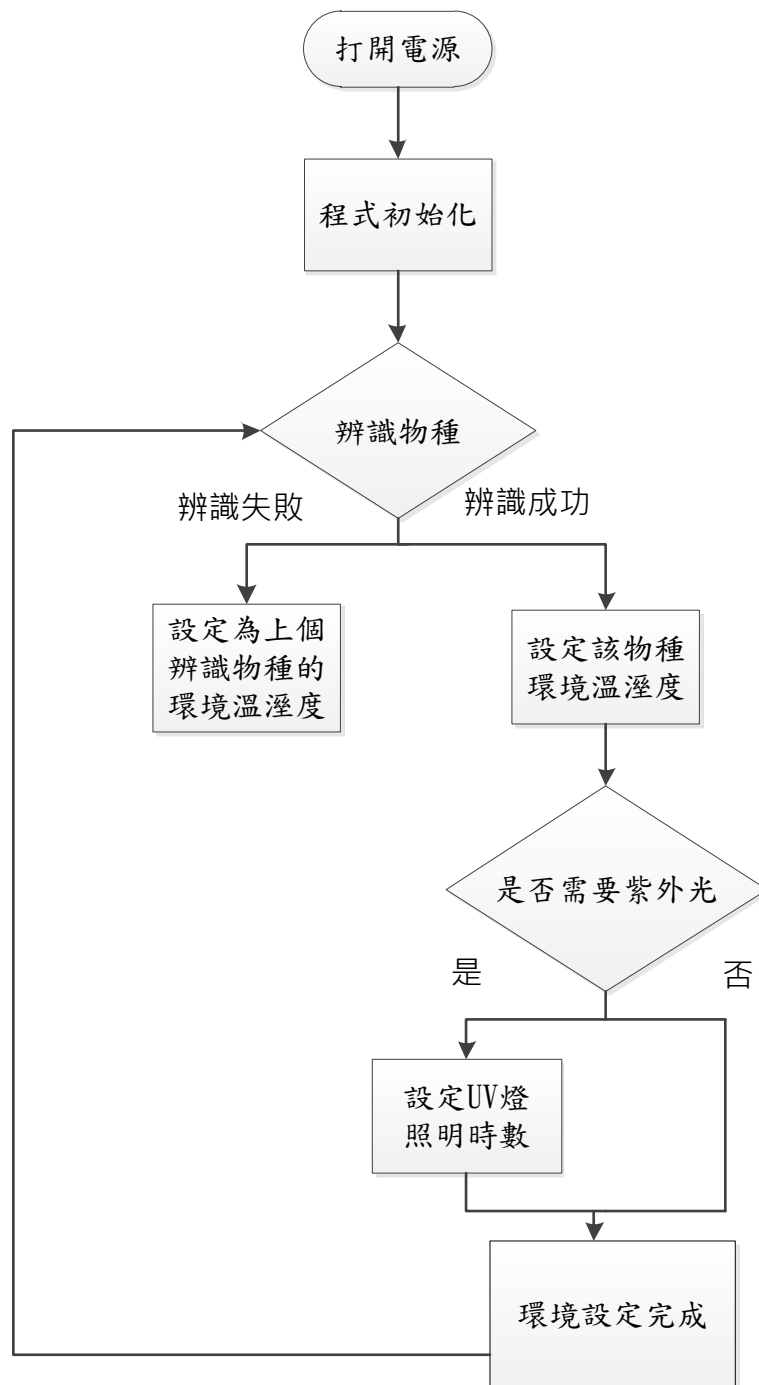


圖 6 環境設定流程圖

2、元件動作程式

環境設定完之後，元件將會根據設定的值進行動作，如圖 7 所示。

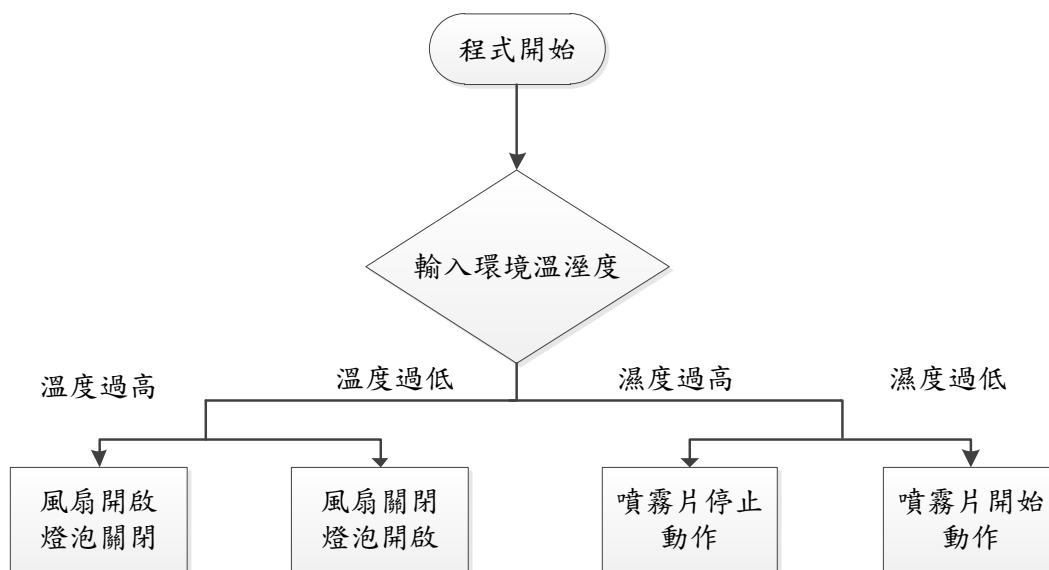


圖 7 元件動作流程圖

二、用材料及工具

(一)、零件介紹

1、電源供應器

電源供應器(如圖 8)，是電能轉換類的電源，主要功能為將標準交流電轉成低壓穩定的直流電，或直接將直流電轉成低壓穩定的直流電給某些特殊的器材使用。我們使用它來提供給 ESP8266 及繼電器所需的電源。電源供應器之規格如下表 2：

表 2 電源供應器規格

產品尺寸	198mm x 98mm x 41mm
輸入電壓	100-150V AC 170-240V AC
輸入電流	3A 115V AC，1.3A 230V
輸入頻率	60Hz
輸出電壓	DC 12V
輸出功率	120W
輸出電流	0-10A



圖 8 電源供應器

2、ESP8266

ESP8266(如圖 9)具有 WIFI、藍芽、價格便宜、提供 micro USB 介面等優點。ESP8266 之規格如下表 3：



圖 9 ESP8266

表 3 ESP8266 規格

產品尺寸	50.8×25.4mm
重量	7g
工作電壓	5V
輸入電壓	5v
PWM 腳位	10 個

3、紫外線感測器(UV 光譜感測器)

在本專題中因為有爬蟲類須適度照射太陽光，因此我們使用了 ultraviolet radiation b(UVB)燈，且為了提供爬蟲類良好的環境，我們裝設了紫外線感測器(UV 光譜感測器)來感測環境紫外線強度以便有效控制紫外線強度(如圖 10)。紫外線感測器(UV 光譜感測器)之規格如下表 4：

表 4 紫外線感測器(UV 光譜感測器)規格

產品尺寸	11mmx27mm
檢測範圍寬	240nm-370nm
工作電壓	2.5~5V
大角度	130 度



圖 10 紫外線感測器(UV 光譜感測器)

4、取暖 ultraviolet radiation a(UVA)燈泡

爬蟲類比起其他寵物有著較為嚴苛的溫度需求，因此我們使用取 UVA 燈泡(如圖 11)來提供爬蟲類需要的溫度。此燈泡除了提供 UVA 光源可用作取代太陽日光外，也能增進爬蟲活力與食慾保持健康。如下表 5：



表 5 取暖 UVA 汝燈泡規格

工作電壓	AC 100V~120V
瓦數	50W
燈座	E27

圖 11 取暖 UVA 汝燈泡

5、UVB 燈

除了蛇類之外，幾乎所有的爬蟲都需要由陽光來獲取必需的营养，因此我們加裝了 UVB 燈泡(如圖 12)，不僅能維持骨骼的強健也能幫助牠們生成維生素 D3 促進鈣質吸收，此外，紫外線也有助於讓爬蟲類的體色維持鮮明。如下表 6：



表 6 UVB 燈

產品尺寸	長 16 CM .直徑 6CM
瓦數	26W
工作電壓	110V
燈座	E27

圖 12 UVB 燈

6、溫溼度感測器(DHT22)

在本專題中因為要隨時感測爬蟲箱的溫濕度，我們選擇使用 DHT22(如圖 13)，他有超小的體積、極低的功耗，信號傳輸距離可達 20 米以上，並且 DHT22 的誤差值比 DHT11 更為精準，這讓爬蟲類可以過得更為舒適。 DHT22 之規格如下表 7：

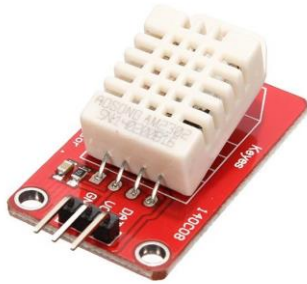


表 7 DHT22 規格

產品尺寸	40x23mm
重量	4g
工作電壓	5V
溫度範圍	-40~80°C $\pm 0.5^{\circ}\text{C}$
濕度範圍	20~90%RH $\pm 2\%\text{RH}$

圖 13 DHT22 溫溼度感測器

7、繼電器

繼電器(如圖 14)具有控制系統和被控制系統，通常運用於自動控制電路中，他實際上是用較小的電流去控制較大電流的一種「自動開關」。故在電路中有著自動調節、安全保護、轉換電路等作用。因此我們用它來控制風扇運轉跟加熱燈與 UV 燈的開關。繼電器之規格如下表 8：

表 8 繼電器規格

工作電壓	5V
尺寸	73×50×18.5mm
重量	55g
觸點容量	250V 10A(AC)或 30V 10A(DC)

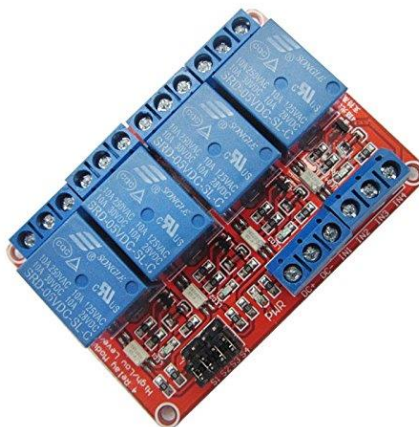


圖 14 繼電器

8、霧化噴霧片

微孔霧化噴霧片(如圖 15)，是有超聲波晶片貼合經過鐳射打孔的鋼片而成的，因為鋼片上有很多微小的孔徑而因此得名。而且它具有驅動電壓低、功率小、阻抗低、波形穩定、高轉換效率等優點。微孔霧化噴霧片之規格如下表 9：

表 9 霧化噴霧片規格

尺寸	直徑 16mm；孔數 740，孔徑 5um
諧振頻率	113KHz
輸入電壓	5V
輸入電流	300mA
功率	2W



圖 15 霧化噴霧片

9、攝像頭

我們透過攝像頭(如圖 16)來拍攝爬蟲類，並將照片傳到樹莓派後，進行影像辨識。我們將攝影機放到洞穴的頂部，加大物種被拍到的範圍。攝像頭的規格如表 10：

表 10 攝像頭 W2FN5-CB11B 規格

工作電壓	5V
解析度	640*480
鏡片組成	3 塑膠鏡片+1 紅外線過濾鏡片
聚焦範圍	60cm~無限
像素	2.0M pixel

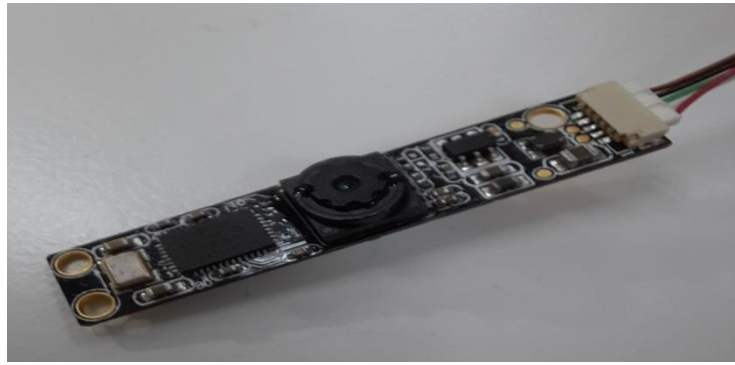


圖 16 攝像頭 W2FN5-CB11B

一、軟體介紹

1. App Inventor

App Inventor(如圖 17)是一款可以簡單設計 Android 手機程式的工具。書寫方式採用了類似積木堆疊的方式呈現，不是傳統複雜的程式碼，因次適合初學者；且介面外觀簡單扼要，使用方始簡易，在雲端即可完成設計。我們透過手機面板，可以看到目前箱內的環境狀態。



圖 17 App Inventor logo

2. Arduino

Arduino 的開發環境是以 AVR-GCC 和其他一些開源軟體為基礎，採用 Java 編寫的，軟體可在 Arduino 的網站上免費下載。Arduino 開發環境使用的語法與 C/C++相似，非常容易使用。Arduino 程式以 setup()及 loop()兩個函式為主體(如圖 18)，隨編寫者需求能夠加上副程式及使用所需之函式庫。

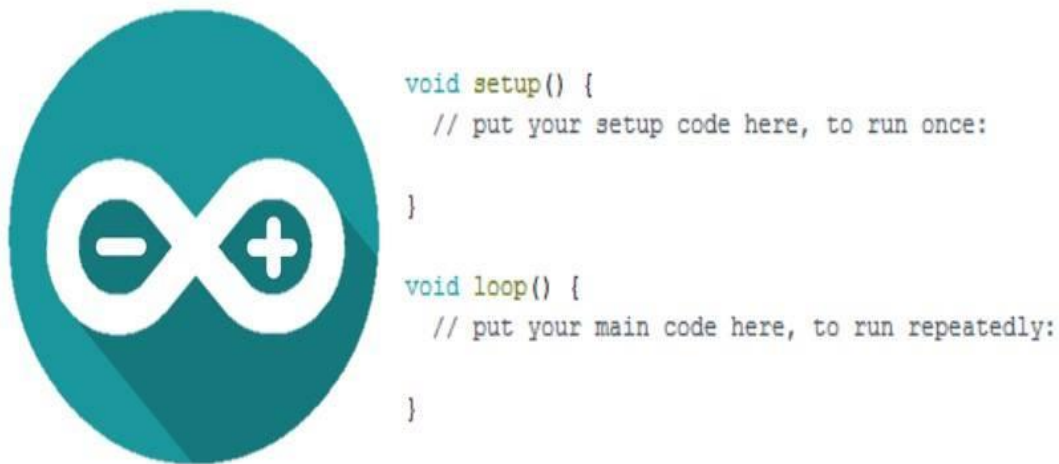


圖 18 Arduino 程式撰寫

3. Azure Custom Vision

Custom Vision(如圖 19)是一個可以使用者簡單上傳圖片、標記，即可達成訓練模型的網站功能，使用者不需有相關機器學習背景也可以自行訓練模型。只需要透過上傳圖片再進行 label 的分類，就可以快速地訓練出模型，照片越多，訓練出來的辨識度也會成正比的上升。再從訓練好的模型把 TensorFlow(如圖 20)的程式檔導出來進行程式的修改(如圖 21)。



圖 19 Azure Custom Vision logo

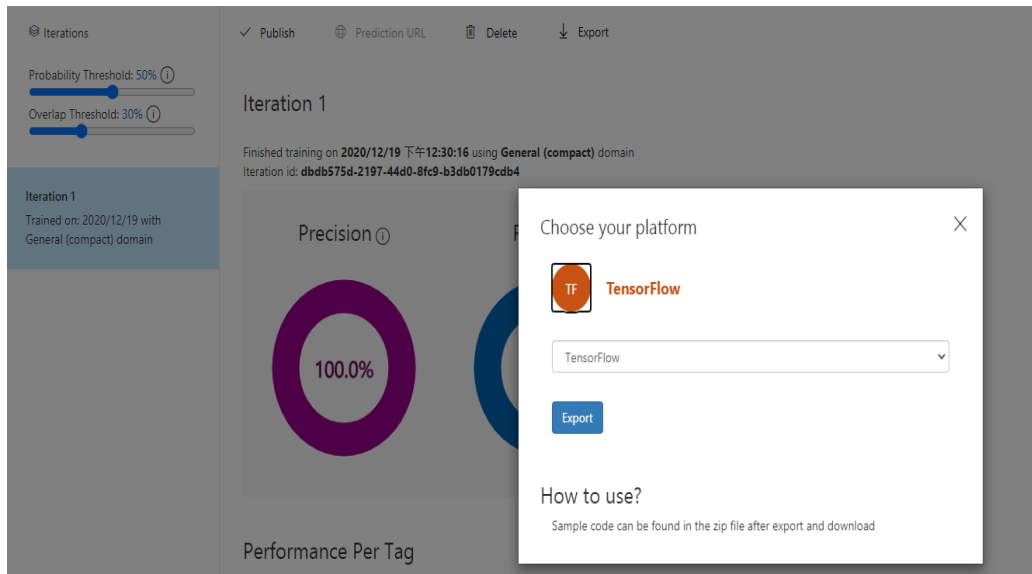


圖 20 Rhino6 Azure Custom Vision TF 檔輸出

```
# The steps implemented in the object detection sample code:
# 1. for an image of width and height being (w, h) pixels, resize image to (w', h'), where w/h = w'/h' and w' < h'
# 2. resize network input size to (w', h')
# 3. pass the image to network and do inference
# 4. If inference speed is too slow for you, try to make w' < h' smaller, which is defined with DEFAULT_INPUT_SIZE
"""Sample prediction script for TensorFlow i.x."""
import sys
import tensorflow as tf
import numpy as np
from PIL import Image
from object_detection import ObjectDetection

MODEL_FILENAME = 'model.pb'
LABELS_FILENAME = 'Labels.txt'

class TFObjectDetection(ObjectDetection):
    """Object Detection class for TensorFlow"""
    def __init__(self, graph_def, labels):
        super(TFObjectDetection, self).__init__(labels)
        self.graph = tf.Graph()
        with self.graph.as_default():
            input_data = tf.placeholder(tf.float32, [1, None, None, 3], name='Placeholder')
            tf.import_graph_def(graph_def, input_map={'Placeholder:0': input_data}, name='')

    def predict(self, preprocessed_image):
        inputs = np.array(preprocessed_image, dtype=np.float)[0, :, :, (2, 1, 0)] # RGB -> BGR
        with tf.Session(graph=self.graph) as sess:
            output_tensor = sess.graph.get_tensor_by_name('model_outputs:0')
            outputs = sess.run(output_tensor, {'Placeholder:0': inputs[np.newaxis, ...]})
            return outputs[0]

def main(image_filename):
    # Load a TensorFlow model
    graph_def = tf.compat.v1.GraphDef()
    with tf.compat.v2.io.gfile.GFile(MODEL_FILENAME, 'rb') as f:
        graph_def.ParseFromString(f.read())

    # Load labels
    with open(LABELS_FILENAME, 'r') as f:
        labels = [l.strip() for l in f.readlines()]

    od_model = TFObjectDetection(graph_def, labels)

    image = Image.open(image_filename)
    predictions = od_model.predict_image(image)
    print(predictions)
```

圖 21 TensorFlow 程式

4. Python

Python(如圖 22)是一種易於學習、功能強大且被廣泛使用的高階程式語言，屬於通用型程式語言，具有高效能的資料結構和簡單又有效的物件導向方法，以及對象編程的簡單而有效的方法，相比於 C++ 或 Java，Python 讓使用者能夠用更少的程式碼來表達想法。我們將導出的 TF 檔進行修改，利用分類時 label(如圖 23)的不同來當作辨識的條件，達成我們辨識不同爬蟲類的目的，再透過序列埠來傳送訊號以及接收回傳的確認訊號。



圖 22 Python logo

```
def main(image_filename):  
    # Load a TensorFlow model  
    graph_def = tf.compat.v1.GraphDef()  
    with tf.io.gfile.GFile(MODEL_FILENAME, 'rb') as f:  
        graph_def.ParseFromString(f.read())  
  
    # Load labels  
    with open(LABELS_FILENAME, 'r') as f:  
        labels = [l.strip() for l in f.readlines()]  
  
    od_model = TFObjectDetection(graph_def, labels)  
  
    image = Image.open(image_filename)  
    predictions = od_model.predict_image(image)  
    for i in predictions:  
        a=(i.get('tagName'))  
        print(a)  
        if len(a) < 6:  
            print("1")  
            #UART.write(a)#gecko  
        elif len(a) > 6:  
            print("2")  
            #UART.write(b)#turtles  
        else:  
            print("3")  
            #UART.write(c)#snakes
```

圖 23 辨識條件

伍、研究結果

一、溫濕度感測

溫溼度感測透過 DHT22 來感測環境溫度，說明如下：

根據我們多次溫溼度感測實驗，發現放在箱子偏高的地方，跟環境的相似度最為一致，如下圖 24，我們將它放置於風扇的左側。

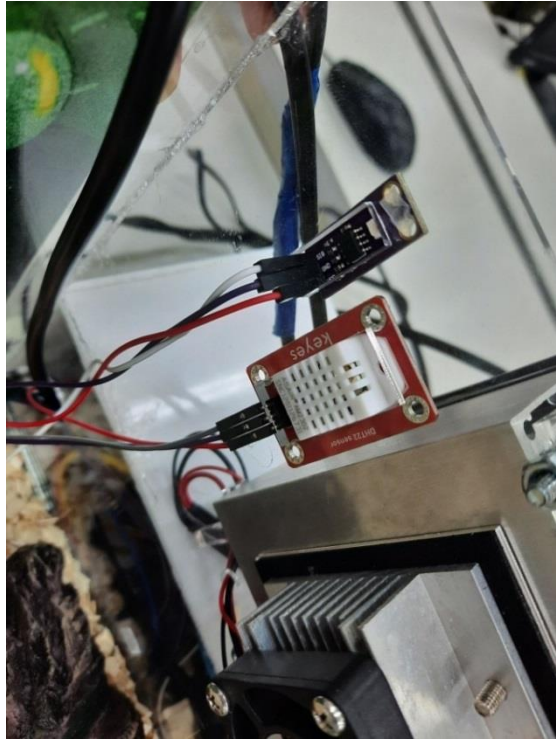


圖 24 溫溼度感測位置

二、溫溼度控制

溫濕度控制 我們根據先前影像辨識的結果，設定完環境溫濕度及 UV 燈照明時數後，透過 Arduino 程式來控制環境狀態。當溫度大於該物種的適合溫度時，致冷風扇開啟、加熱燈關閉；溫度低於條件時，致冷風扇關閉、加熱燈開啟。濕度過低時加濕晶片將會動作、到達一定水平後會自動關閉。UV 燈只會在該種爬蟲類需要紫外線時才會開啟，像是烏龜需要每天約 4 小時的照明，守宮跟蛇則不用。

三、手機介面

手機介面透過 wifi 傳送溫溼度的變化狀況，以及被辨識成功的物種，如圖 25。





圖 25 手機介面及辨識物種

四、影像辨識

我們透過攝像鏡頭拍攝辨識的物種(如圖 26)，根據爬蟲類的身體特徵來判斷，看該辨識物是屬於哪個已建立好的模組。啟動後影像辨識會重複執行，當沒有偵測到任何物種，不會傳送辨識資訊，直到辨識成功才會傳送辨識到的物種設定的溫溼度值，等待下一次偵測到不同物種，才會改變辨識內容。



圖 26 模擬辨識狀況

陸、討論

一、溫溼度感測

在溫溼度感測部分，我們原本使用 DHT11 來感測，但在測試階段發現，偵測出來的數值跟環境溫溼度有蠻大的誤差，感測的範圍不夠大，跟整個箱內環境的實際狀況有出入。最後我們改使用 DHT22，相較之下，它對整個環境的感測較好，誤差也有大幅下降，感測的溫溼度範圍也較大。在顯示方面，溫溼度以及紫外線光都透過網路用 App Inventor 在手機上呈現，但可能是因為跟巴法雲的一些連接有問題，當 esp8266 通電後，必須要再斷一次電，手機上的數值才會隨著環境而改變，我們最後也沒能找出問題，但我們另外利用 Arduino UNO R3 在通電後自動對供電進行一次斷電，來解決需要手動斷電的問題。

表 11 元件比較

	DHT11	DHT22
價格	<NT\$100	NT\$150~ NT\$250
濕度範圍	20%~80%，誤差 5%	0%~100%，誤差 2%~5%
溫度範圍	0°C~50°C，誤差±2°C	-40°C~125°C，誤差 ±0.5°C
取樣平率	每 1 秒不超過一次	每 2 秒不超過一次

二、影像辨識

一開始選用樹莓派 pi 3b+開發板來處理影像辨識的程式，考慮到程式執行時的速度以及可能因運作量太大，而造成卡頓的問題，為了讓專題在辨識部分有較好的動作呈顯，改用了最新的樹莓派 pi 4，在辨識的速度上有很大的加分，也沒有卡頓現象。而在辨識成功度上，因為資源有限的關係，成功度沒有辦法像市面一樣那麼高，為了盡可能的增加成功度，從一開始物體的全身改成針對物體個別的特徵來進行判斷的根據，經過測試，發現成功度真的有些許的提升，如果想要再進一步加強，必須增加每個物種的資料量，連結大數據，才能有很高的辨識成功率。

三、元件擺放

在最終測試時我們發現，根據 DHT22 溫濕度感測器與風扇、加熱燈、噴霧片的相對位置，我們得到的溫濕度值會與實際箱子溫度有誤差。一開始我們沒有考慮到這個問題，導致風扇吹出來的風會直接吹到溫濕度感測器上，使得溫度不管怎麼測都會比實際溫度低。為了修正這個誤差，我們把感測元件放在和風扇同一側，避開噴霧片噴霧的位置，並且裝設在箱子較短的一面上，讓我們的感測元件感測到的是氣體經過整個箱內循環後較接近實際溫溼度的數值。經過實驗，我們發現這個方法確實有效。

柒、 結論

經過不懈努力，我們把最初的想法以專題呈現出來。影像辨識結合環境溫濕度自動控制，完美達到了更加方便照顧爬蟲類的目的。UV 燈能在固定時間下，提供爬蟲類需要的紫外線，不僅可以維持骨骼強健，也能讓爬蟲動物體內製造維生素 D3，幫助鈣質吸收。加熱燈更是爬蟲類的必需品之一，許多的生理或行為功能，都依賴溫度的維持才得以順利進行。以及加裝了風扇防止天氣太熱時把寵物悶壞、在乾燥的天氣加濕來讓爬蟲類過得更舒適。

研究一份專題就像在工作上處理一件專案，每個人都像是一顆螺絲釘，缺一不可。從討論、分作、或偶有的想法分歧，到後來達成共識、互助合作。研究過程中也會遇到許多問題，與預期成果不相符，甚至幾度想放棄、改用其他簡單的方法。我們上網查閱了無數個網站，不斷再錯誤中嘗試，最後我們同心協力，終於將成品做出。這些，都超越了教科書與老師所能帶給我們的知識。



圖 27 爬爬的溫馨之家

捌、參考資料及其他

- 電源供應器規格。2020/8/23。取自
<https://www.ruten.com.tw/item/show?22038433700215>
- 取暖 UVA 鈹燈泡。2020/9/3。取自
<https://24h.pchome.com.tw/prod/DXAM61-A9009XEOY?fq=/S/DXAM61>
[1](https://24h.pchome.com.tw/prod/DXAM61-A9009XEOY?fq=/S/DXAM61)
- UVB 燈。2020/9/3。取自
<https://shopee.tw/110V-UVB-5.0-10.0-26W-%E8%A3%9C%E9%88%A3%E7%87%88%E7%B4%AB%E5%A4%96%E7%B7%9A%E7%87%88%E6%B3%A1%E7%94%9F%E6%88%90D3%E5%B9%AB%E5%8A%A9%E5%90%B8%E6%94%B6%E9%88%A3%E7%B2%89%E7%88%AC%E8%9F%B2%E5%85%A9%E6%A3%B2%E9%99%B8%E9%BE%9C%E7%83%8F%E9%BE%9C%E5%AE%88%E5%AE%AE%E8%9C%A5%E8%9C%B4%E5%A4%9A%E8%82%89%E6%A4%8D%E7%89%A9%E7%BC%BA%E9%88%A3%E6%9B%AC%E8%83%8C-i.4842952.1402798175>
- UV 光譜感測器規格。2020/9/11。取自
<https://www.waveshare.com/uv-sensor.htm>
- ESP8266 規格。2020/9/11。取自
<https://pcbdesign88.wordpress.com/2018/04/11/node-mcu%E9%96%8B%E7%99%BC%E6%9D%BFesp8266-wifi%E6%A8%A1%E7%B5%84%E4%BB%8B%E7%B4%B9/>
- UV 光譜感測器規格。2020/9/11。取自
<https://www.waveshare.com/uv-sensor.htm>
- DHT22 溫溼度感測器規格。2020/9/11。取自
<https://www.raspberrypi.com.tw/23140/dht22-temperature-and-humidity-sensor/>
- 四路繼電器。2020/10/3。取自
<https://www.playrobot.com/relay/1771-4-channel-5v-relay-module.html>
- 微孔噴霧片規格。2020/11/4。取自
<https://www.ruten.com.tw/item/show?21836607199619>
- Arduino Uno 規格。2020/11/4。取自
<https://shop.cpu.com.tw/product/40500/pdf2/>
- How to install opencv on RaspBerryPi4。2020/12/13。取自
<https://www.pyimagesearch.com/2019/09/16/install-ope>

[ncv-4-on-raspberry-pi-4-and-raspbian-buster/](#)

- Costom Vision 。 2020/12/13 。 取自
[https://blog.cavedu.com/2019/09/30/azure-custom-vision/](#)
- Tinkercad logo 。 2020/12/13 。 取自
[https://images.app.goo.gl/PS5eGbDXdSNPTWpc7](#)
- RDWorks V8 logo 。 2020/12/13 。 取自
[https://images.app.goo.gl/NdGu9SaR8Ev7fhe79](#)
- Install Tensorflow and Keras on the Raspberry Pi 。 2020/12/13 。 取自
[https://www.teknotut.com/en/install-tensorflow-and-keras-on-the-raspber
y-pi/](#)
- 爬蟲箱 。 2020/12/23 。 取自
[https://24h.pchome.com.tw/prod/DXAM59-A900AXWZ9?gclid=CjwKC
AiA9bmABhBbEiwASb35V8Muz1aN3bTaC50QPx7t-u5urf9ibUMjy LS
7ZqLlZjfYXNNOkosURoCMMwQAvD BwE](#)
- 影像辨識 。 2021/1/25 。
[https://futurecity.cw.com.tw/article/1529?fbclid=IwAR0E 8yLC12xS4Si7J
_4qmXfgGEzTywACGFyFKKcI12r 9 -IDHMe5F B9o](#)