

中華民國第 60 屆中小學科學展覽會 作品說明書

高級中等學校組 工程學(一)科

佳作

052313

智慧型滅火系統

學校名稱：國立屏東高級工業職業學校

作者： 職二 張良旭 職二 包峯任 職二 林桔慶	指導老師： 方勇傑 辜正同
---	-----------------------------

關鍵詞：樹莓派 Python、手機 App 程式、
Intelligent

摘要

人們用火於烹飪、生熱、照明及推進…等等，由此看的出來火在人類社會上是不可或缺的，但同『水可載舟，亦可覆舟』的道理一樣，使用不當也會造成危險的火災，使人、財務及住處的傷亡及損失，市面上有許多種的火災灑水器，而傳統的灑水器是以”煙”或”溫度”來做偵測，並大範圍灑水，但灑水器偵測方式過於簡易，容易發生誤判，及大範圍灑水的方式造成不必要的損失，為解決此問題，我們以機器視覺來追蹤火點，並以定點噴水的方式取代大範圍灑水。

本小組製作的自動偵測火源灑水系統以 Raspberry Pi3 板、Arduino UNO 板、做為偵測火源及噴水的主體，並以 Webduino Smart 板 及 App inventor2 建立起物聯網，萬一發生火災也能及時通知用戶，以便提早通報火警，儘可能的減少火災造成的損失。

壹、 研究動機

為預防火災，現今社會的部份住戶都會在家中裝上火災警報器及灑水器，不過傳統灑水器的觸發方式過於簡略，會有發生誤觸的問題，從而導致另一種災情發生，又因為以大範圍灑水的方式，不免會有無辜的家具、家電因泡水而報廢。

相較於簡易的溫度觸發和大範圍灑水，我們認為需要使用更多的判斷方式和定點灑水，以解決上述傳統灑水器帶來的種種問題。

我們也察覺到近年來物聯網的蓬勃發展，因此將灑水器的狀態透過網路傳送給用戶，讓使用者在外也能立即知道房子的狀況。

貳、 研究目的

一、本研究以讓灑水器更智慧、科技化為主旨，研究目的如下

1. 增加其火源判斷方式;
2. 增加自動尋找火源程式;
3. 增加自動火源追蹤程式;
4. 變更以大範圍灑水滅火方式;
5. 建立起物聯網。

參、 研究設備及器材

一、零件與器材簡介

(一) 如(表一) 是本專題使用到的器材及零件

(二) 應用方面:

1. 樹莓派主要應用於影像辨識，作為偵測火源主體。
2. Arduino UNO 控制板主要應用在接收樹莓派傳送的字元，配合字元去做伺服馬達的增減。
3. Webduino Smart 板主要應用於傳送樹莓派的偵測狀態。
4. 伺服馬達為調整攝影機位置及水柱發射方向的重要工具
5. 抽水馬達替代傳統灑水器，可以對需要的位置提供更集中的水源
6. 網路攝影機可為樹莓派提供所需影像
7. 火焰燈作為模擬火源使用
8. 感熱 IC 是偵測火災是否發生(65°C)的工具
9. 8*8 熱像儀做為偵測火源的雙重判斷及修正
10. 壓力感測器偵測射出水壓，依不同水壓回饋給噴水頭角度

(表一)零件與器材表

零件名稱:	規格:	數量:
Raspberry Pi 3b (樹莓派 3B)	1.2 GHz 64-bit quad-core ARM Cortex-A53	1 個
Arduino UNO	Atmel ATmega328P AU 單 晶片	1 個
Webduino smart	ESP8266	1 個
伺服馬達	SG90	4 個
抽水馬達	365 直流微型隔膜泵	1 個
網路攝影機	Logi C310 HD 720p	1 個
火焰燈	5v	1 個
8*8 熱像儀	SparkFun Grid-eye AMG8833	1 個
壓力感測器	Z0052	1 個
三色 LED	串列式	64 個

肆、 研究方法(過程)

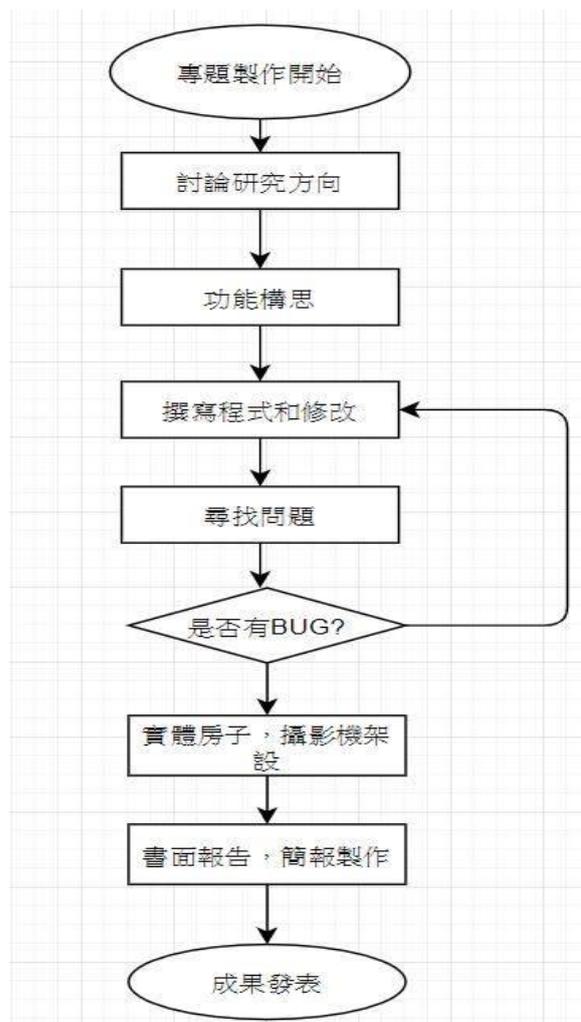
詢問老師並參考書本上及網路上有關於 Raspberry、Python、Arduino、Webduino Smart 板、App inventor 2 及其他配備相關的程式知識及使用方式，並應用於我們小組的自動偵測火源灑水系統程式，進而完成此完全自動化系統。

一、製作流程:如圖一，說明如下

(一) 開始專題製作;

(二) 小組討論關於此作品應朝向哪方面去發展

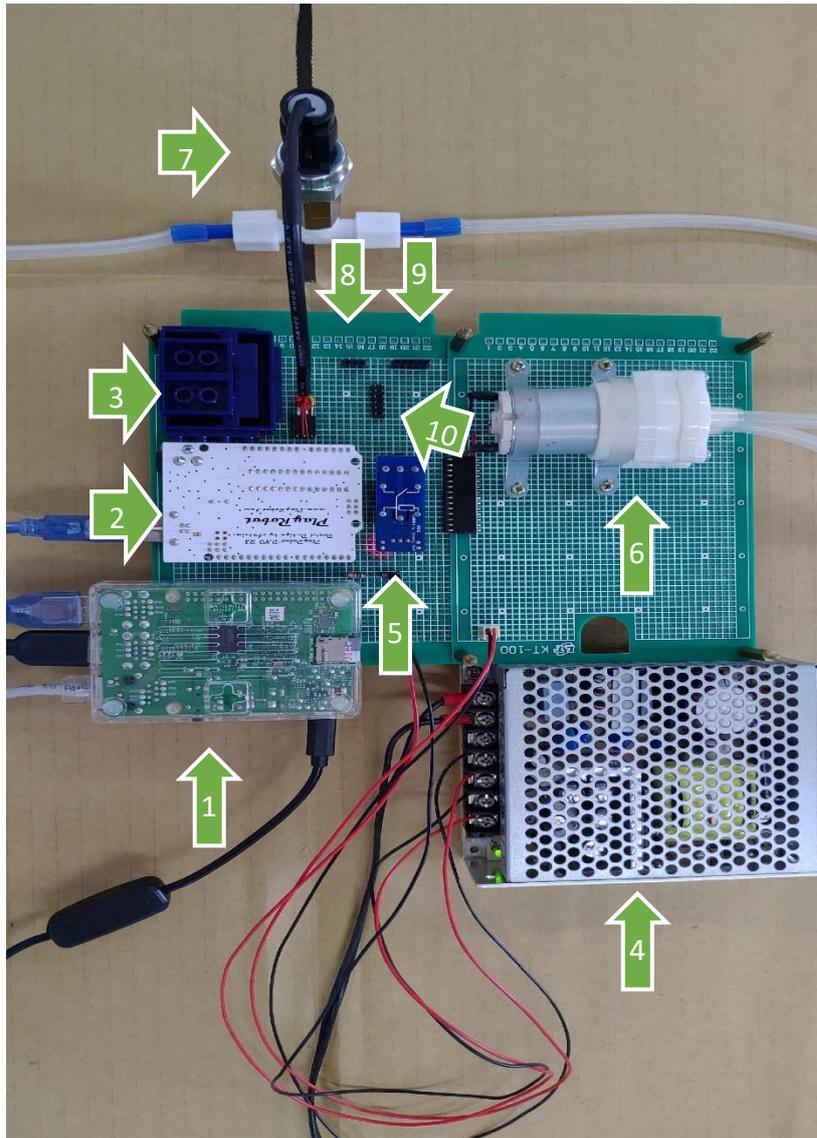
- (三) 組員討論此作品所需功能;
- (四) 小組利用課程及學校資源開始撰寫程式;
- (五) 遭遇到的困難並解決問題：
 - 1. 程式編譯錯誤
 - (1) 語法不對，參考書本資料做更正;
 - 2. 伺服馬達沒動作
 - (1) 配線是否有誤，檢查程式定義是否有誤
- (六) 尋找是否有 BUG，並將以更正;
- (七) 按照設計圖開始搭建模擬小屋，並架設攝影機;
- (八) 開始著手製作書面資料、簡報;
- (九) 成果發表。



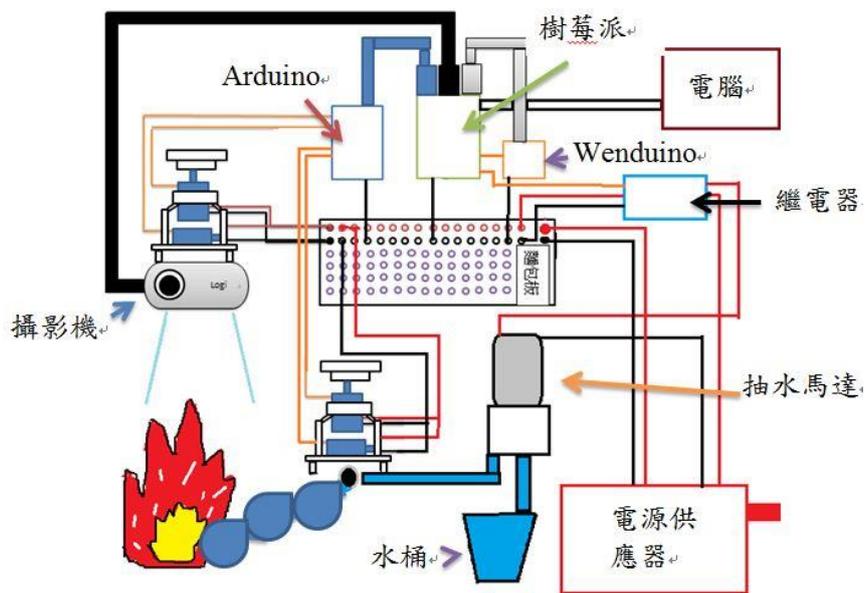
(圖一)製作流程圖

二、實體及虛擬電路圖

- (一) 箭頭 1 為樹莓派，為此系統之主體，使用攝影機做影像辨識掃描、偵測，追蹤火源，鄰近 Arduino Uno 板(箭頭 2)及 Webduino 板(箭頭 3)，以方便板子間互相的配合。
- (二) 箭頭 4 為電源供應器，負責提供 5V 及 12V 分別給電路上的板子和抽水馬達(箭頭 6)使用。
- (三) 箭頭 5 為繼電器，讓樹莓派利用小信號控制使用 12V 的抽水馬達之啟動及停止。
- (四) 箭頭 7 為壓力感測器，判斷抽水馬達送給噴水端的水壓，並回傳資料給 Arduino UNO 做噴水端的角度補償，從而角度修正。
- (五) 箭頭 8 為 8*8 熱像儀使用的接腳，將所讀取資料由 I2C 的 SDA、SCL 傳送給 Arduino UNO 做火源的第二判斷。
- (六) 箭頭 9 為 8*8 三色 LED 的接腳，接收由 Arduino UNO 傳送出來的資料，用於讓我們看到熱像儀的現在狀態。
- (七) 箭頭 10 為 4 個伺服馬達的接角，接收由 Arduino UNO 傳送出來的資料，分別帶動攝影機及噴水端口的方向。



(圖二) 實體電路圖



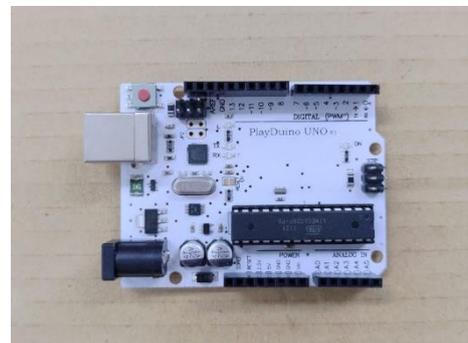
(圖三) 虛擬電路圖

三、主要開發板及環境介紹:

- (一) 樹莓派(Raspberry Pi) 如圖四是只有火柴盒大小的微型單晶片電腦，使用開源的 Linux 系統，配有瀏覽器、文字處理、學習程式設計的軟體，所以只要搭配螢幕、鍵盤、滑鼠，就能當一台電腦使用，Raspberry Pi 3B 照片如圖三，即是樹莓派的第三代，只有信用卡般的大小，跟 Raspberry Pi 2 的尺寸別無二致，需使用 5V,3A 電源，配備了 1GB 的 RAM 和 400MHz VideoCore IV GPU，內建藍牙 4.0 和 802.11n Wi-Fi，因架構上改善與處理器升級讓它的時脈增加了 33%。
- (二) Arduino UNO 如圖五，是個只有手掌大小的電路控制板，Arduino 用於連接數位裝置和互動式物件，以便在物理和數位世界中感知和控制物件，且允許任何人製造 Arduino 板和軟體分發。電路板設計使用各種微處理器和控制器，這些電路板配有一組數字和類比 I/O 引腳，可以連接各種擴充板或麵包板和其他電路。這些電路板具有串列埠，包括某些型號上的通用串列匯流排（USB），也用於從個人電腦載入程式。通常使用 C/C++ 程式語言。除了使用傳統的編譯工具鏈之外，Arduino 專案還提供了一個基於 Processing 語言專案的整合式開發環境。
- (三) Webduino 如圖六，如其名就是 Web 和 Arduino 的結合，慶奇科技於 2015 年終於可以使用純粹 Web 的方式來做操控，除了可以讓原本的 Arduino 上網，還能做「雲端更新韌體程式」、「用各種程式語言開發」以及「使用 Wi-Fi 控制」，使物連網的名字實至名歸，常見的物聯網連線控制的方法有四種，分別有網際網路控制（W-Fi）、區域網路控制（WebSocket）、藍芽（Bluetooth）和序列埠（Serial Port）。



(圖四)Raspberry PI3 板



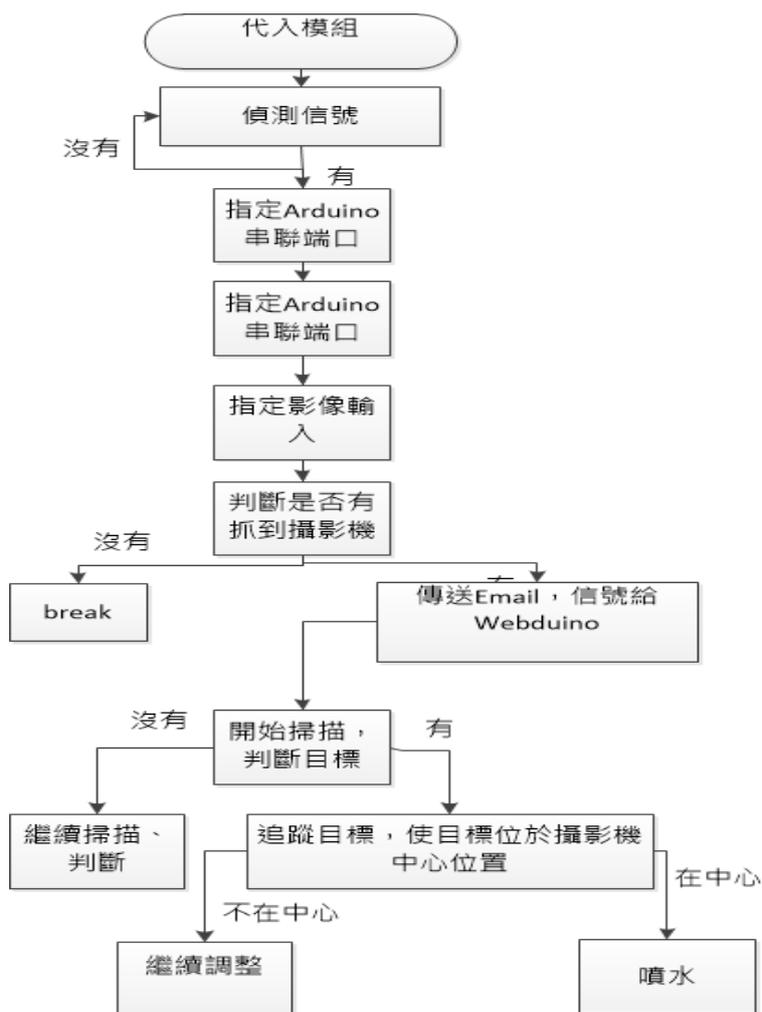
(圖五)Arduino UNO 板



(圖六)Webduino smart

(四) App Inventor 是一個在線上開發的 Android 程式環境，不使用複雜的程式碼而採取積木式的堆疊法來完成 Android 程式，這樣的程式結構即便是初學者也能輕易上手，程式完成後只要將手機與電腦連線，剛做好的程式馬上就能在手機上使用。

(五) Python 是一種廣泛的高階語言，作為一種直譯語言,Python 的設計哲學強調代碼的可讀性和簡潔的語法，使用空格縮排來取代大括號劃分程式區塊，相較於 C++和 Java 都能使用較少的程式來表達想法，使程式結構更清楚明瞭。



(圖七) Raspberry pi 程式流程圖

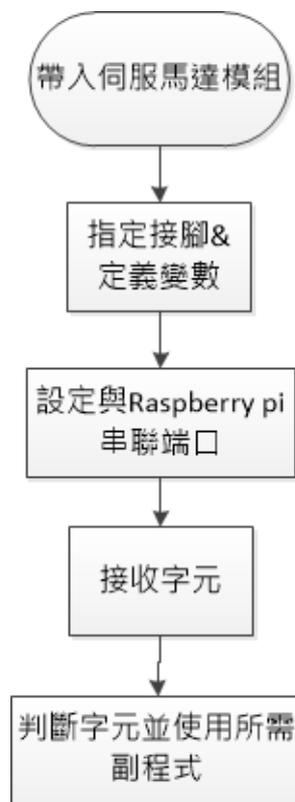
四、程式流程

(一) Raspberry PI3 的程式流程如圖(七)

程式啟動後，樹莓派會先帶入模組，並快速偵測環境，如果沒有發生火災則會持續偵測，反之則會開始一系列動作。首先會指定 Arduino 的串聯端口，並指定影像輸入，接著判斷是否有抓到攝影機，如果沒有又會回到偵測信號的地方重新載入，如果有就會傳送 e-mail 及信號至 Webduino，並且繼續判斷目標的動作，找到目標後會調整攝影機的位置，直到目標位於中心點。

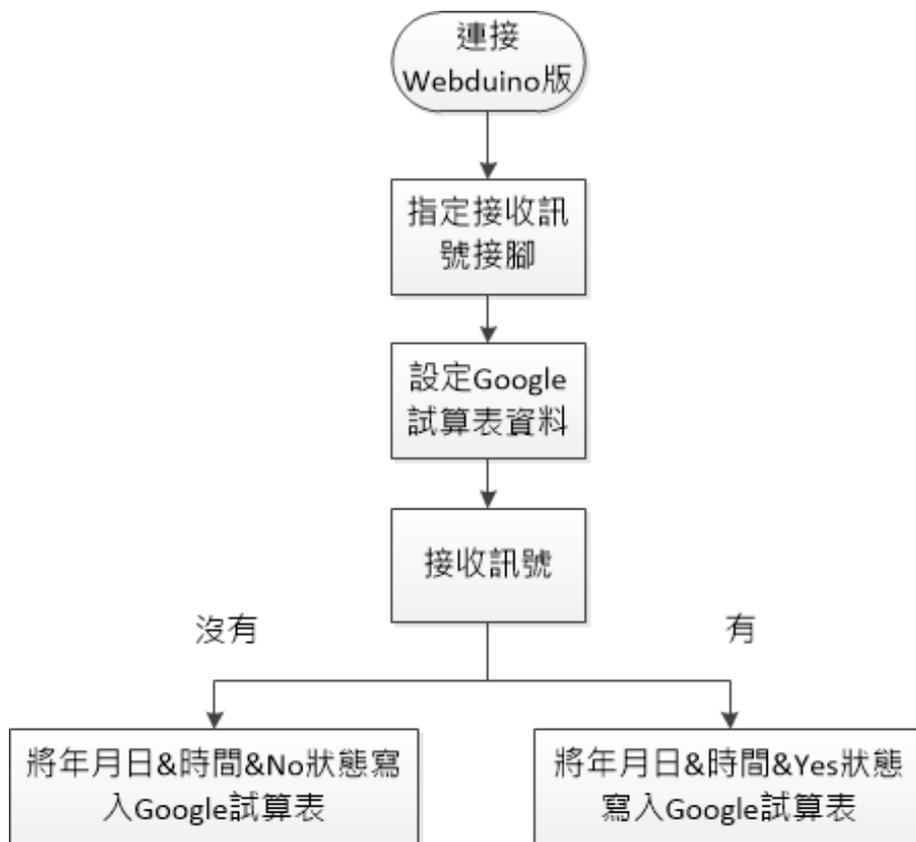
(二) Arduino 的程式流程如圖(八)

接收到樹莓派傳送來的信號後，即開始帶入模組、指定接腳、定義變數、設定與樹莓派的串聯接口，之後樹莓派的訊息會透過串聯接口傳送給 Arduino，再自行判斷需要使用的副程式。



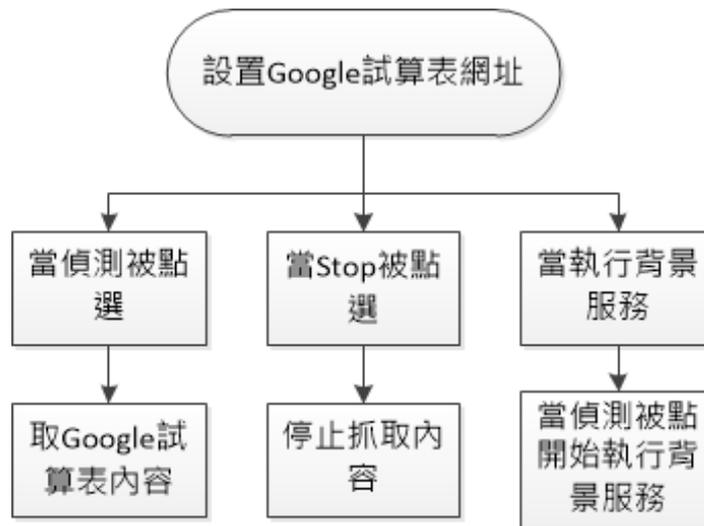
(圖八) Arduino 程式流程圖

(三) Webduino 的程式流程如圖(九)收到樹莓派的訊息後，會指定接收訊號的接腳，接著設定 google 試算表的資料，之後由接腳接收的訊息決定將在試算表內寫入年月日&時間&Yes 狀態 亦或是 No 狀態



(圖九) Webduino 程式流程圖

(四) App inventor2 的程式流程如圖(十)與 google 試算表的網址相連，點擊偵測時可以取得 google 試算表的內容，點擊 stop 即可停止接收試算表的內容，也可以使用背景服務



(圖十) App inventor2 程式流程圖

五、主要程式說明

(一)、Raspberry pi 主要程式

1、設置 E-mail 傳輸基本設定及內容

```

smtp=smtplib.SMTP("smtp.gmail.com", 587)

smtp.ehlo()

smtp.starttls()

smtp.login("scorpion897657@gmail.com","ujtddtogkyfhizcs")

from_addr="Raspberry.fire.test"

to_addr="scorpion897657@gmail.com"

msg="Subject:Gmail sent by Python scripts\nFIRE!FIRE!"

status=smtp.sendmail(from_addr, to_addr, msg)

if status=={}:

    print("郵件傳送成功!")

else:

    print("郵件傳送失敗!")

smtp.quit()
  
```

2、將 BGR 轉換成 HSV

```

hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
  
```

```

hsvb = hsv.copy()

# define range1 of Red color in HSV

lower_red = np.array([0,50,225])

upper_red = np.array([50,150,255])

# Threshold the HSV image to get only Red colors

mask0 = cv2.inRange(hsv, lower_red, upper_red)

mask3 = mask0.copy()

#偵測二值化後的圖像，比較哪一個圖像較大

contours = cv2.findContours(mask3,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)

contours1 = sorted(contours[1],key = cv2.contourArea,reverse = True)

```

3、如果圖像的水平位子小於 LOW，輸出 A 給 Arduino

```

if left < Low:

ser.write('A'.encode())

ser.flush()

```

(二)、Arduino 主要程式

1、偵測字元

```

if(Serial.available())

#定義 cmd 為端口讀到的值

int cmd = Serial.read();

#如果端口為下列的字元則使用配合的副程式

if(cmd == 'A') DR();

else if(cmd == 'B') DL();

else if(cmd == 'C') UR();

else if(cmd == 'D') UL();

else if(cmd == 'E') roll();

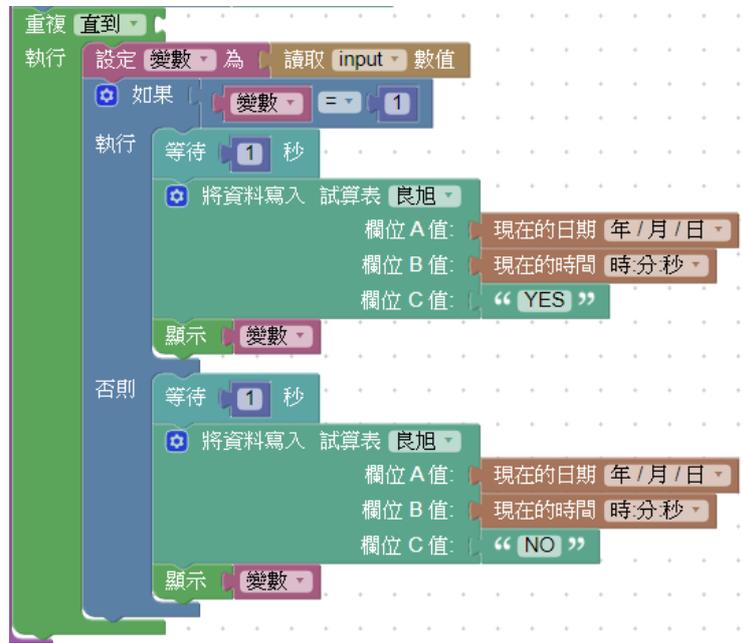
else if(cmd == 'F') Receivedegree();

```

}

(三)、Webduino 主要程式

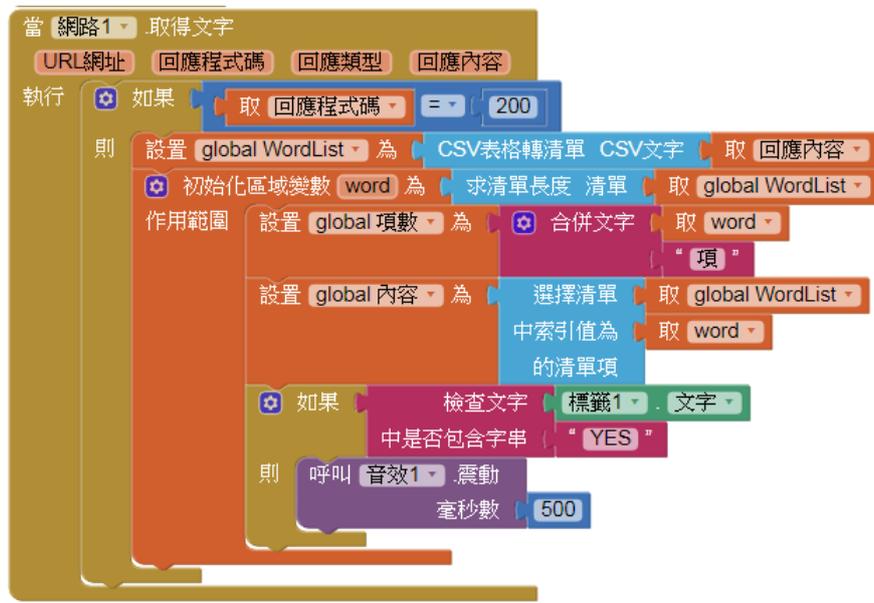
1.如圖十一所示，判斷變數為 1 或是 0，將 YES 或 NO 填進 Google 試算表



(圖十一)Webduino 主要程式圖

(四)、App inventor 主要程式

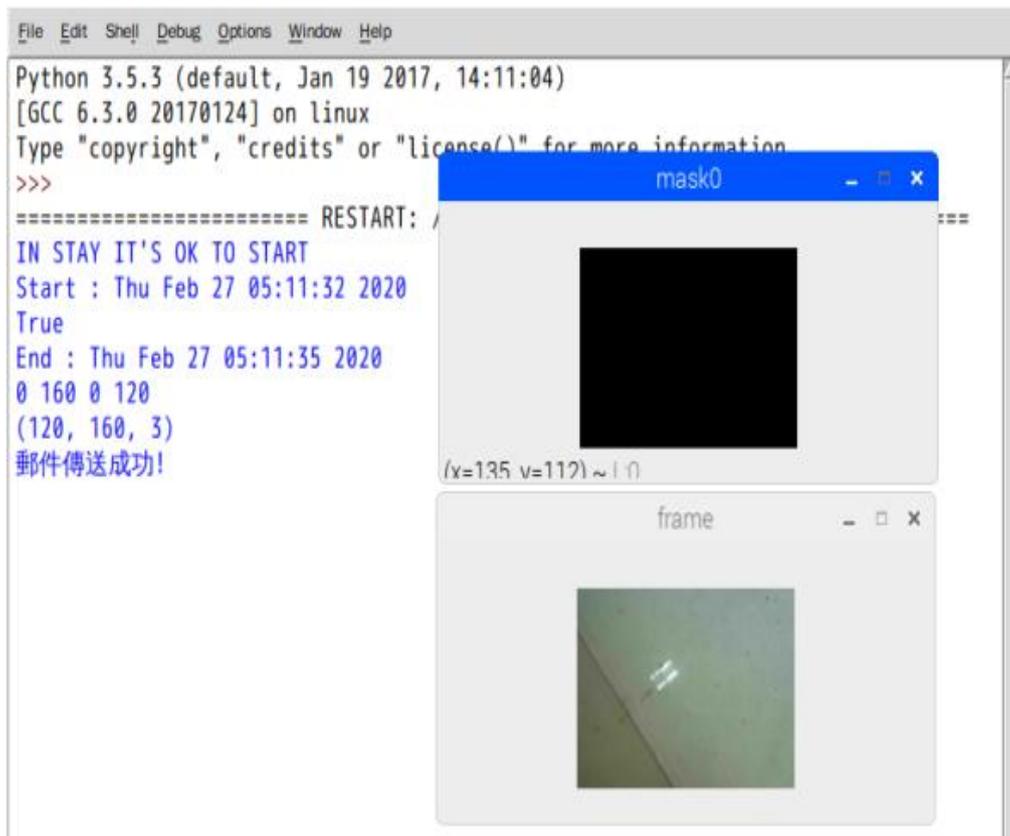
1、如圖十二所示，取完內容後，檢查是否含有 YES，有的話就震動



(圖十二) App inventor2 主要程式圖

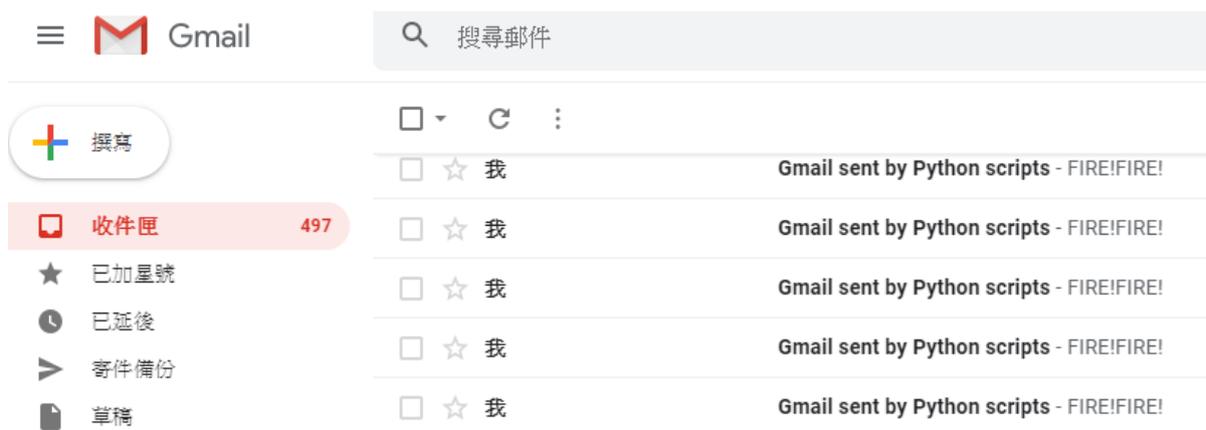
伍、 研究結果

一、 假定房間內有一處發生火災，溫度到達 65°C，樹莓派啟動如圖十三



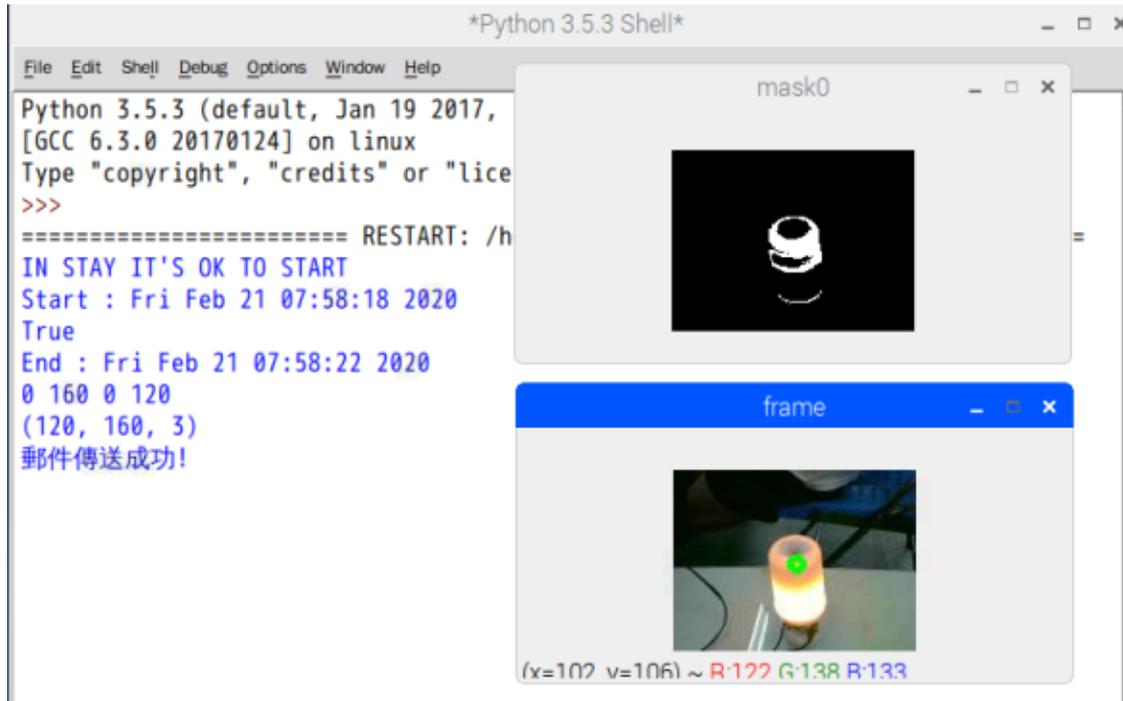
(圖十三)因溫度到達 65°C 後程式啟動

二、 用戶會在程式啟動時同時接收到由 Raspberry PI3 板傳來的 Email 通知



(圖十四)組員收到 Email 的火災通知

三、攝影機偵測到目標後，傳送信號給 Arduino UNO 板及 Webduino smart 板。



(圖十五)攝影機成功辨識出目標，並傳送信號

四、 Webduino 在程式啟動時便每秒將現在日期、時間及到偵測狀態傳送到 Google 試算表

<i>fx</i>	A	B	C
4904	2020/2/2	20:31:34	YES
4905	2020/2/2	20:31:35	YES
4906	2020/2/2	20:31:36	YES
4907	2020/2/2	20:31:37	YES
4908	2020/2/2	20:31:38	YES
4909	2020/2/2	20:31:39	YES
4910	2020/2/2	20:31:40	YES
4911	2020/2/2	20:31:41	YES
4912	2020/2/2	20:31:42	YES
4913	2020/2/2	20:31:43	YES
4914	2020/2/2	20:31:44	YES
4915	2020/2/2	20:31:45	YES
4916	2020/2/2	20:31:46	YES
4917	2020/2/2	20:31:47	YES
4918	2020/2/2	20:31:48	YES
4919	2020/2/2	20:31:49	YES
4920	2020/2/2	20:31:50	YES
4921	2020/2/2	20:31:51	YES
4922	2020/2/2	20:31:52	YES
4923	2020/2/2	20:31:53	YES
4924	2020/2/2	20:31:54	YES
4925	2020/2/2	20:31:55	NO

(圖十六)Webduino smart 每秒上傳資料中

五、用戶如有開啟 App inventor2 程式，則程式會每秒抓取 Google 試算表的資料

(一)、狀態顯示 YES 的話手機會震動，NO 則不會

(二)、執行背景服務時，只要不把此程式移掉，就會繼續執行



(圖十七)組員點選偵測後，讀取資料為 YES，開始震動



(圖十八) 組員點選偵測後，讀取資料為 NO 讀取資料_NO



(圖十九) 組員點選 STOP 後，停止程式



(圖二十)組員點選背景服務後，程式在背景時仍可繼續服務

六、追蹤目標

如圖二十一至圖二十二，移動火源後，攝影機開始進行追蹤



(圖二十一)攝影機追蹤右側起火



(圖二十二)攝影機追蹤左側起火

七、定位噴水

(一)、如圖二十三、二十四之繼電器動作，抽水馬達啟動

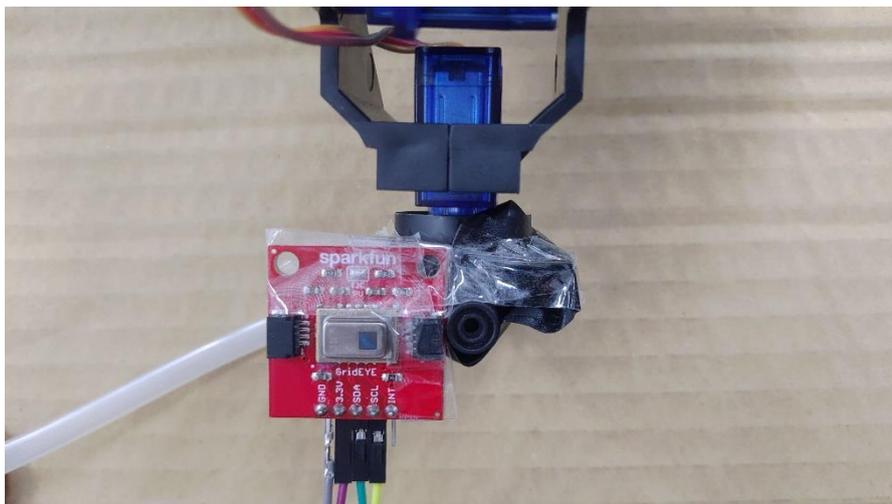


(圖二十三)繼電器亮的時候呈現常開，抽水馬達不會動作

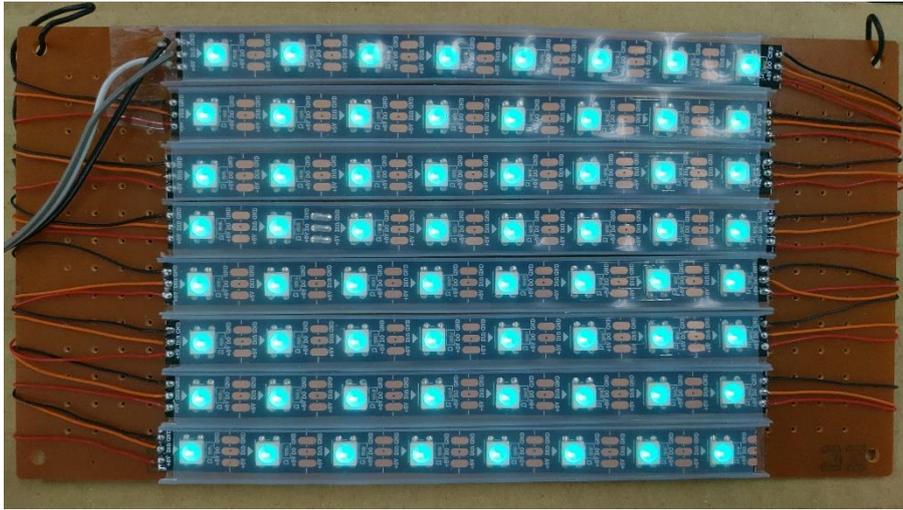


(圖二十四)繼電器不亮的時候呈現常閉，抽水馬達開始動作

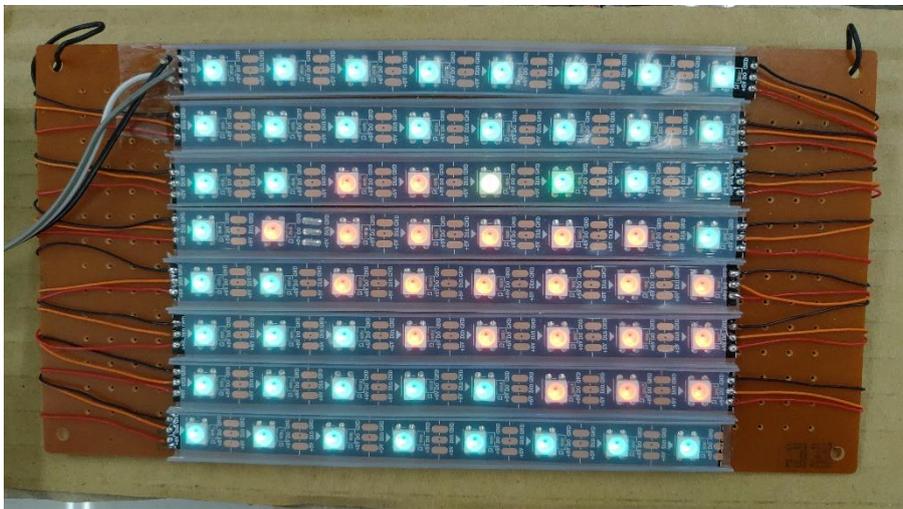
(二)、如圖二十五架設於噴水端的熱像儀會做第二段的熱源追蹤、判斷並導正噴水端口的噴水角度，圖二十六為熱像儀未偵測到熱源之反應，圖二十七為偵測到熱源之反應



(圖二十五)架設於噴水端口的熱像儀，負責第二段的熱源追蹤、判別及導向

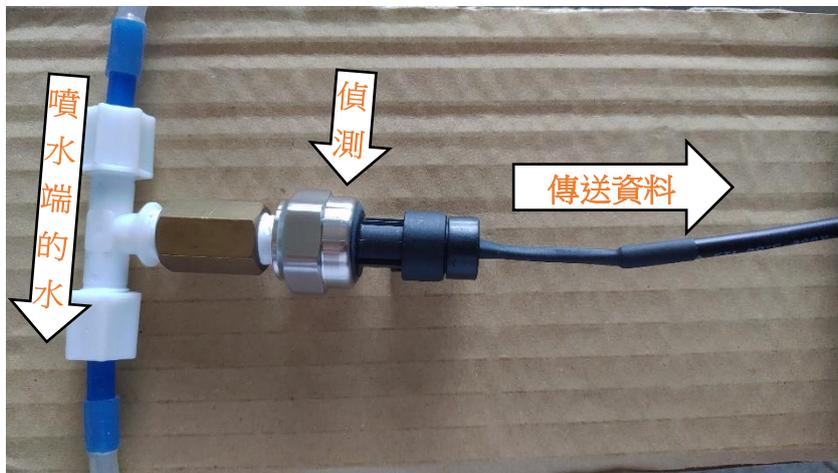


(圖二十六)熱像儀未偵測到 60 度 C 以上熱源



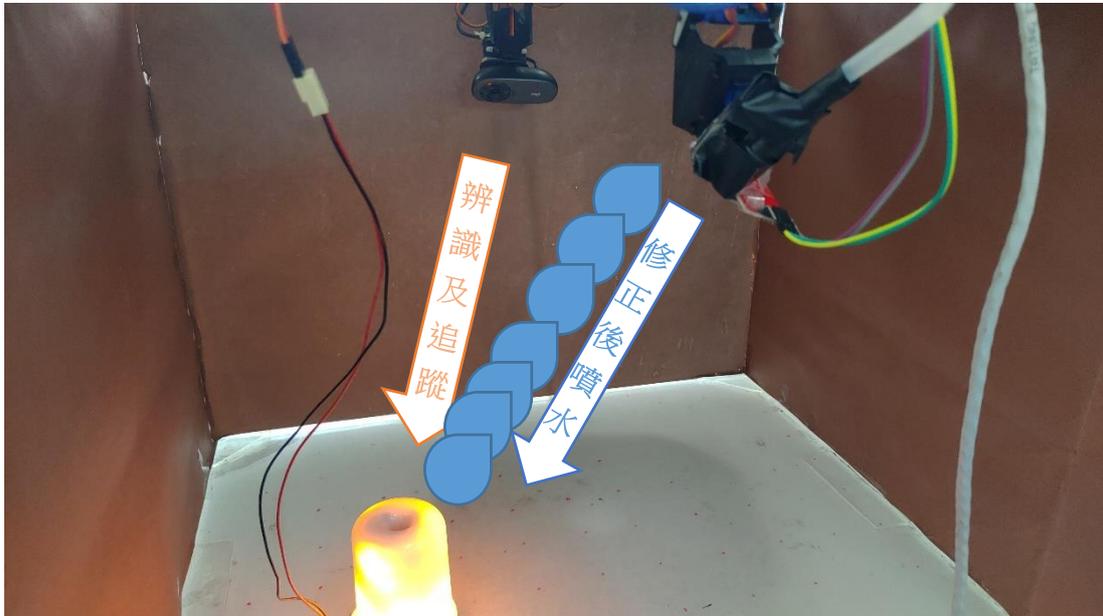
(圖二十七)熱像儀偵測到熱源，圖中測試物為使用中的電烙鐵

(三)、如圖二十八水壓感測器會判斷噴水端水壓狀況並回傳給 Arduino UNO 做噴水端口的噴水角度補償功能



(圖二十八)水壓感測器會讀取噴水端水壓並回傳資料

(四)、如圖二十九，偵測到目標後，程式開始判斷目標位置，使另一馬達會朝向目標再經由熱像儀及水壓感測器的噴水角度修正後，向目標開始噴水



(圖二十九)噴水的馬達會朝向目標噴水

陸、 討論

此作品為我們小組的第一次作品，對組員來說都是一個特別的經驗，從最起出的不知所云，慢慢的學習，察覺到分工合作的重要，到完成其一整套程式系統、書面資料，一路過來得到許多收穫，以下是我們的小組討論：

一、Q:『此作品的使用的辨識方法會不會有誤差?』

A:『有的，此作品採用顏色作為辨識方法，誤差會取決於攝影機、背景亮度而有所影響，但是因有加裝熱像儀做雙重的火源判斷，能有效的降低誤差的比率。』

二、Q:『作品中採用定位噴水方式，可否詳細說明?』

A:『作品中的定位方式是依安裝攝影機及噴水端的位置，去進行推算，取出大約的方向後，在使用噴水端的熱像儀做第二段的火源追蹤，判斷及導向』

三、Q:『此作品能市售嗎?請詳細講解狀況』

A:『此作品仍有需多不及之處，可能需要更多判斷火源及穩定的噴水方式，減少其誤差，才能用於市售』

四、Q:『在不同的房子類型，如透天厝、平屋或是高樓大廈，安裝的方式會不同嗎?』

A:『此系統因採取影像作為辨識火源的主體，所以安裝地點需要在較寬廣、無障礙物及死角的地方，如果房子裡有較多房間或隔間的區域，建議每個房間或隔間應都需分別安裝一個系統』

五、Q:『為甚麼不把攝影機、熱像儀跟噴水端口做結合，這樣不就少了更多麻煩了嗎?』

A:『這個想法在我們小組一開始討論研究方向時有討論到的點，這樣好處是攝影機，熱像儀能同步動作增加火源判斷準確度及只需要追蹤火源就好，省掉了第二顆馬達的定位步驟，但所有原件擠在一起怕被水用濕後短路之外，像是在寬闊的倉庫裡一顆噴水馬達噴出水的範圍一定不夠涵蓋整個倉庫，若照個每一個都要配有攝影機及熱像儀的噴水端口增加數量，所需成本高，所以我們才使用一組攝影機，搭配多組噴水端，這樣成本能有效降低』

柒、 結論

發生火災是不幸的事情，傳統灑水器作動後更又造成了水災，想必家中是無一倖免，能以此作品的功能減少火災及灑水的損失，或許是不幸中的大幸。

此作品有別於傳統灑水器的偵測及灑水方式，使用顏色辨識判斷火源，以定位噴水噴向火源，減少因大範圍灑水的損失，能夠自動掃描及追蹤火源也是其一大特點，此外，我們也建立起物聯網，使用戶能夠及時收到通知。

作品整題及程式部分在製作途中也遭遇許多瓶頸，但是組員們的不屈不饒，藉由本學校電子科的大量資源、課程內容、及老師的大力指導和組員的熱忱，一一突破關卡，製作而成。

捌、 參考資料及其他

[1] Rasperry Pi 樹莓派從不懂，到玩上手 陳會安著

旗標科技股份有限公司

[2] Arduino 微電腦應用實習 施士文 著 台科大圖書股份公司

[3] 手機應用程式設計超簡單 APP Iventor2 零基礎入門班

鄧文淵、文淵閣工作室編著 碁峰資訊股份公司

【評語】 052313

本作品整合最新的機電、視覺、感測及物聯網等技術，開發製作自動偵測火源灑水系統，以機器視覺及熱像儀來追蹤火點，並以定點噴水的方式取代大範圍灑水，有別於市面上火災灑水器以”煙”或”溫度”來做偵測及大範圍灑水，非常有創意，成品具進步性，作者研發過程說明清楚，以強化系統功能驗證的實驗說明，展示系統的可靠性。另外對於可能的應用場域，火災的可能情境，可再深入了解思考，對於作品實際應用會有助益。

摘要

火是一把鋒利且危險的雙面刃，市面上灑水器的傳統偵測方式容易誤判，且大範圍灑水造成不必要的損失，因此採用影像辨識來追蹤火點，以定點噴水來改善，系統中以Raspberry Pi(樹莓派)、Arduino為智慧型滅火器主體，用Webduino、App inventor2結合手機建立輔助物聯網，即時通知用戶以減少火災造成的損失。

研究動機

家中的火災警報、灑水器觸發方式簡略，有誤觸問題，又因大範圍灑水，不免有家具、電器因泡水報廢，我們認為需精準灑水、火源判斷，以解決傳統灑水器問題，也因物聯網蓬勃發展，用Webduino將狀態透過網路傳給用戶，使用戶能立即接收房內危急環境狀況。

研究設備及器材

主要零件、器材簡介如表一

1. 樹莓派影像辨識為偵測火源主體。
2. Arduino UNO配合樹莓派調整馬達。
3. Webduino smart板傳送當下狀態。
4. 火焰燈為模擬火源。
5. 熱像儀輔助樹莓派達成雙重辨識。
6. 水壓感測器使Arduino對噴水端角度補償，彌補水壓不足。

(表一)主零件、器材表

主要零件名稱:
Raspberry Pi 3b
Arduino UNO
Webduino smart
伺服馬達x4
抽水馬達
網路攝影機
火焰燈
熱像儀
水壓感測器

研究方法

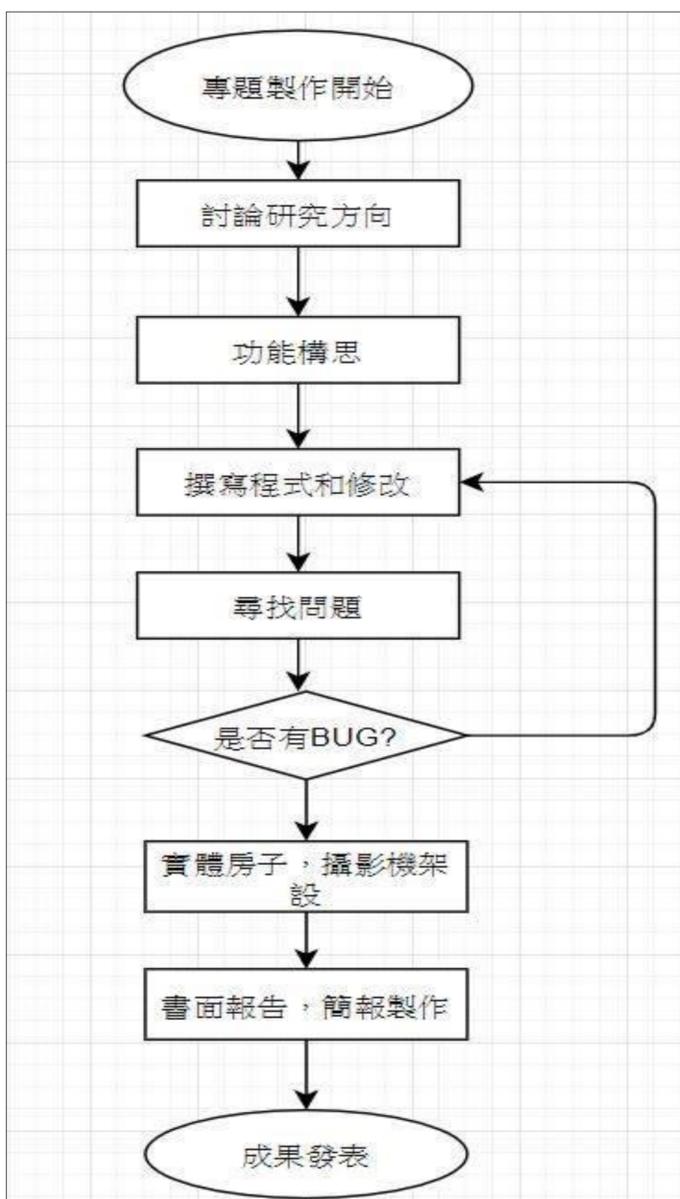
參考書本、網路上程式知識，進而完成智慧型滅火系統。

一、製作流程

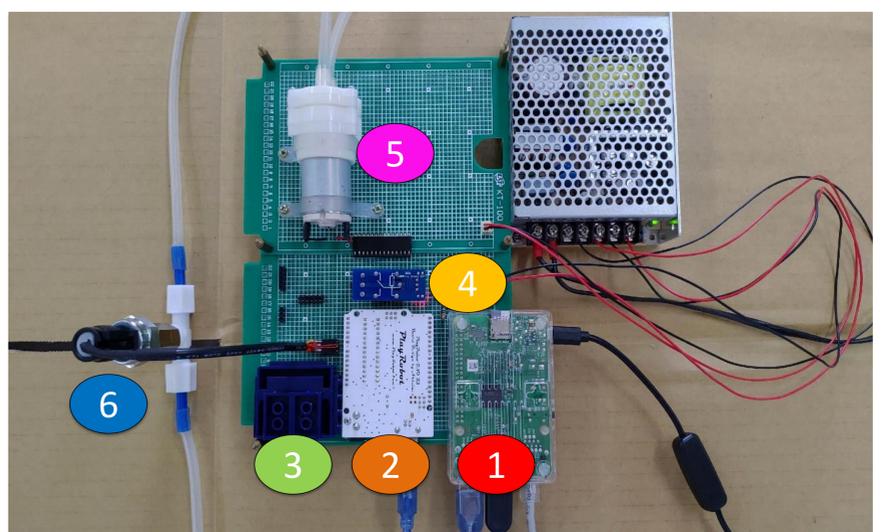
如圖一，研究作品方向、功能，著手程式的開發，解決錯誤，建立模型屋，以模擬現實狀況。

二、實體電路及模擬圖

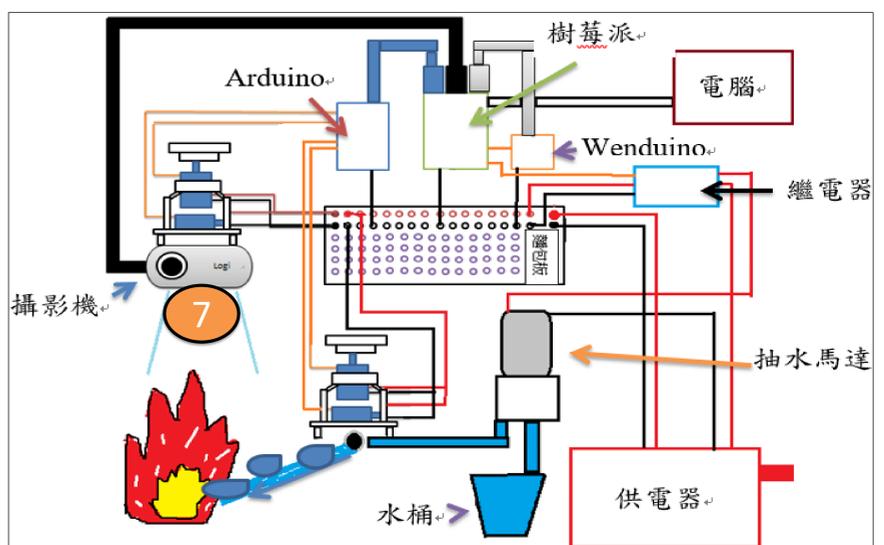
如圖二，樹莓派控制板①與Arduino板②、Webduino smart板③及圖三的網路攝影機⑦互相連接；繼電器④幫助樹莓派(小電流)控制抽水馬達(大電流)⑤；⑥為水壓感測器



(圖一)製作流程圖



(圖二)實體電路



(圖三)模擬電路圖

三、主要開發板及環境介紹：

樹莓派	使用Linux系統的微型單晶片電腦。
Arduino UNO	Arduino用於連接數位裝置和互動式物件。
Webduino	Web和Arduino的結合，可用網路或藍芽連接並控制周邊元件。
AI2	是使用積木堆疊法的Android程式，將手機與電腦連線程式就能在手機使用。
Python	是一廣泛的高階語言，相較C++和Java，程式結構更清楚明瞭。

四、程式流程：

(一)樹莓派程式流程如圖四

樹莓派做基本設定，傳送e-mail及信號至Webduino後，偵測、判斷目標，一發現就追蹤目標，確定位置調整噴水端後噴水。

(二)Arduino UNO程式流程

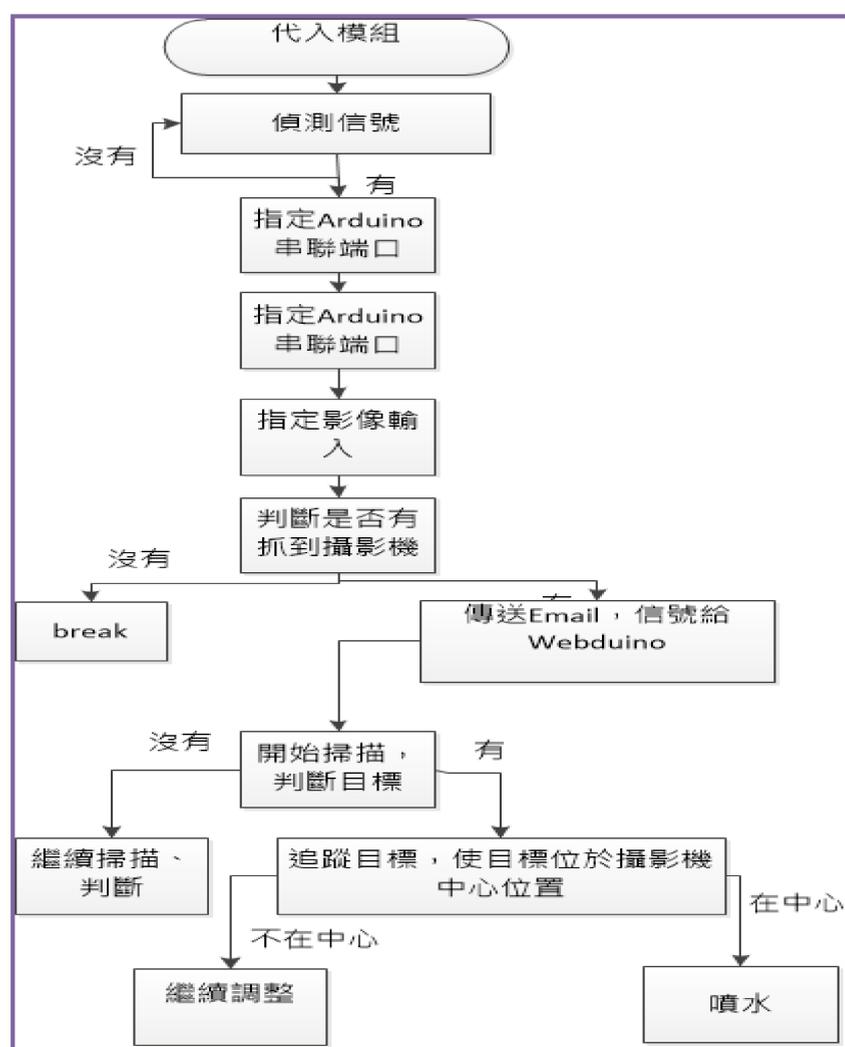
接收樹莓派傳送信號，自行判斷需要使用的副程式。

(三)、Webduino的程式流程如圖五

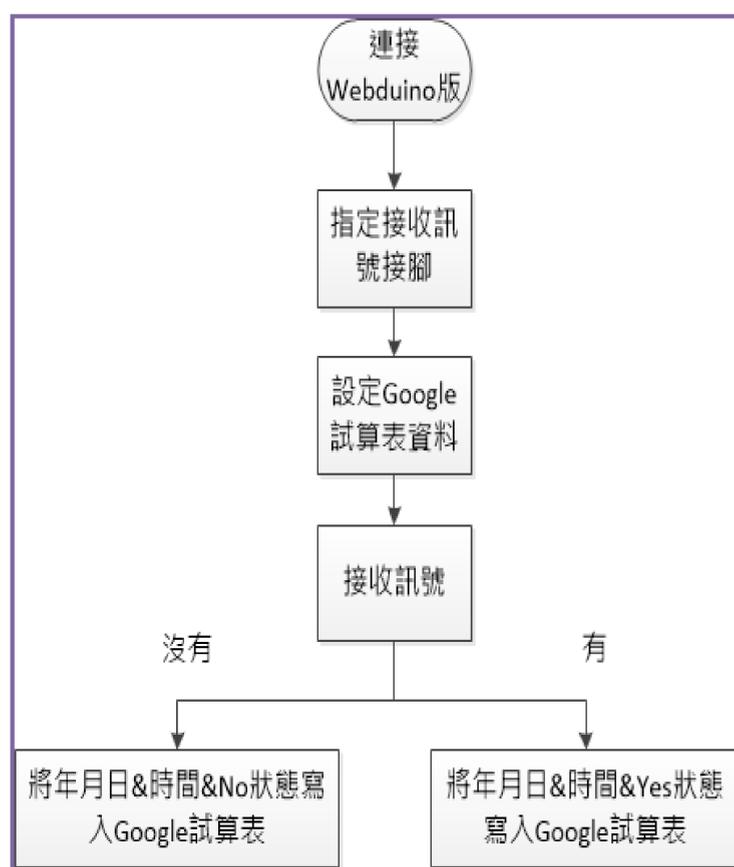
接收樹莓派訊息並在Goole試算表內寫入年月日&時間&現在狀態。

(四)、App inventor2程式流程

與Google試算表網址相連，點擊按鈕決定是否讀取資料。



(圖四) Raspberry Pi程式流程圖



(圖五) Webduino程式流程圖

研究結果

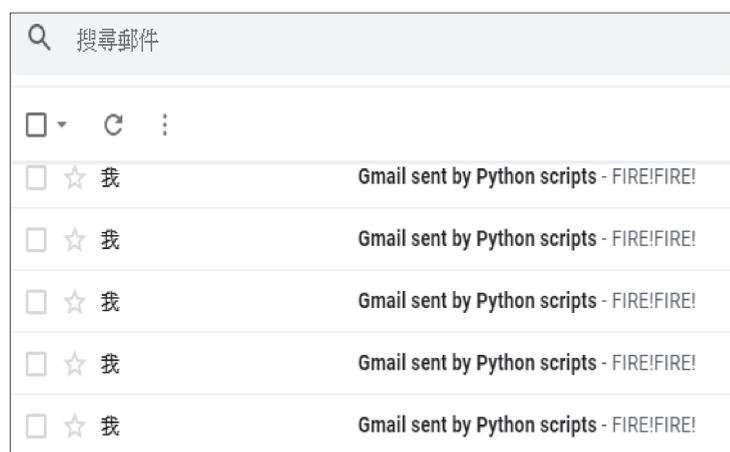
一、假定屋內一處發生火災，環境超過65°C系統啟動，如圖六

二、Raspberry Pi傳送Email通知，如圖七

```

File Edit Shell Debug Options Window Help
Python 3.5.3 (default, Jan 19 201
[GCC 6.3.0 20170124] on linux
Type "copyright", "credits" or "l
>>>
===== RESTART:
IN STAY IT'S OK TO START
Start : Thu Feb 27 05:11:32 2020
True
End : Thu Feb 27 05:11:35 2020
0 160 0 120
(120, 160, 3)
郵件傳送成功!
  
```

(圖六)系統啟動



(圖七)Email通知

三、攝影機偵測到目標，如圖八。

四、Webduino將現在日期、時間及偵測狀態傳到Google試算表，如圖九。

五、使用AI2上按鈕決定動作，有狀態時震動，如圖十。



fx	A	B	C
4904	2020/2/2	20:31:34	YES
4905	2020/2/2	20:31:35	YES
4906	2020/2/2	20:31:36	YES
4907	2020/2/2	20:31:37	YES
4908	2020/2/2	20:31:38	YES
4909	2020/2/2	20:31:39	YES
4910	2020/2/2	20:31:40	YES
4911	2020/2/2	20:31:41	YES
4912	2020/2/2	20:31:42	YES
4913	2020/2/2	20:31:43	YES



(圖八)攝影機找到目標

(圖九)Webduino上傳資料

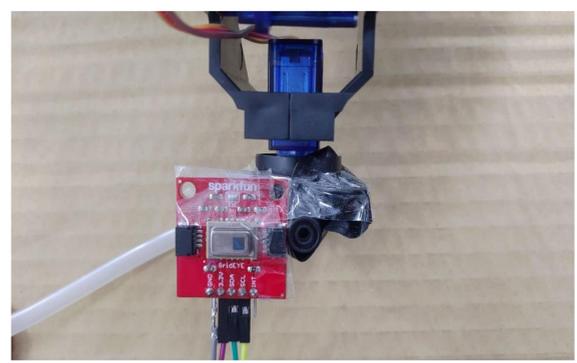
(圖十)AI2畫面

五、攝影機追蹤火源傳送角度給噴水端如圖十一，熱像儀修正路徑如圖十二

六、確定目標噴水如圖十三，根據水壓感測器如圖十四測得的水壓對應如圖十五、十六做角度調整，處理後如圖十七、十八



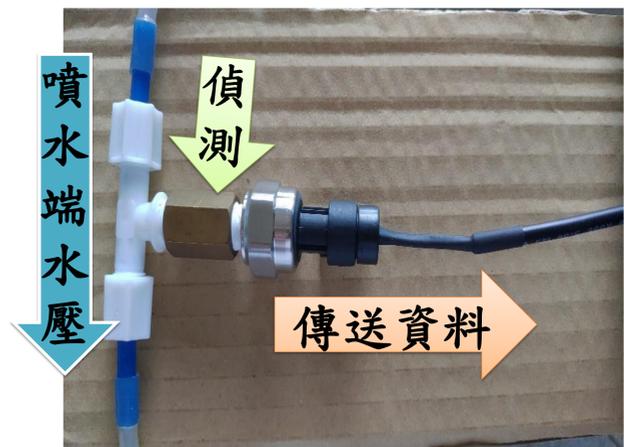
(圖十一)攝影機追蹤目標



(圖十二)熱像儀



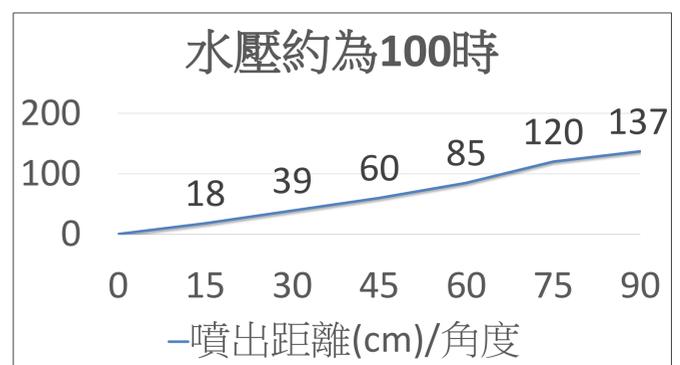
(圖十三)噴水的馬達向目標噴水



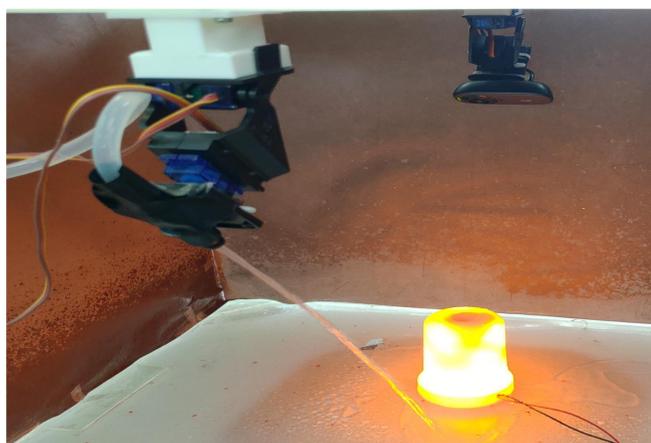
(圖十四)水壓感測器



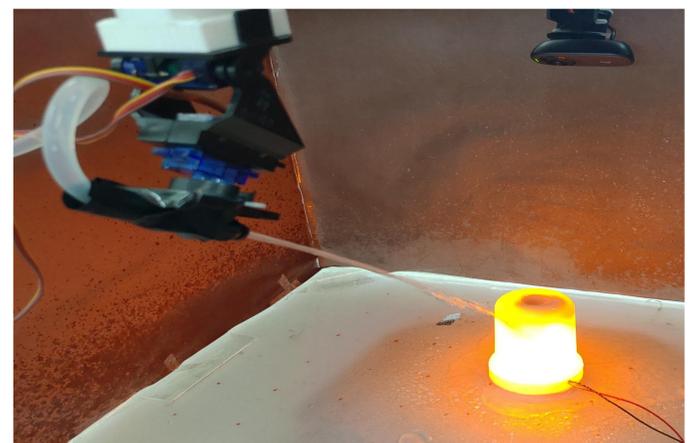
(圖十五)噴出距離與水壓關係



(圖十六)噴出距離與角度關係



(圖十七)角度修正前



(圖十八)角度修正後

結論

此作品有別於傳統灑水器的偵測及灑水方式，使用顏色辨識判斷火源，以定位噴水噴向火源，減少因大範圍灑水的損失，能夠自動掃描及追蹤火源也是其一大特點，此外，我們也建立起物聯網，使用戶能夠及時收到通知。