**全國高級中等學校專業群科111年**

**專題及創意製作競賽**



**群別:電機與電子群**

**作品名稱:**

**關鍵詞:IoT、遠端控制**

**目錄**

**壹、摘要**

**貳、研究動機**

**參、主題與課程之相關性或教學單元之說明**

**肆、研究過程**

**伍、研究結果**

**陸、討論**

**柒、結論**

**捌、參考資料及其他**

**玖、附錄**

**表目錄**

**表1、主題與課程相關性**

**圖目錄**

**圖1、ESP32**

**圖2、Raspberry Pi**

**圖3、步進馬達及驅動板**

**圖4、紅外線接收器**

**圖5、紅外線發射器**

**圖6、Arduino**

**圖7、App Inventor2**

**圖8、Python**

**圖9、Google Cloud Speech-to-Text**

**圖10、Fritzing**

**圖11、APP UI介面(一)**

**圖12、APP UI介面(二)**

壹、摘要:

    近年來，由於全球網路技術的進步加上晶片運算能力的提升以及製造成本的下降，促使物聯網慢慢民用化，時至今日，這個領域已是熱門的話題及技術研究的重點。 從1980發展至現今，IOT的概念和技術漸漸成熟，IOT這項技術讓人們的生活便利了許多，我們打算運用這次機會來深入了解這項領域，因此這次的專題，我們將展示的作品會運用到IOT這項技術，我們會運用到WIFI、藍芽和紅外線這三種傳輸方式，研究各種的傳輸優點和缺點，當然我們的目標是將作品與生活中做結合，讓這次的作品有延續利用的價值。

貳、研究動機:

   有一天我去拜訪朋友的家，剛進門時就被眼前的事物所吸引，門一開電燈就自動亮起來，以及智慧管家的問好，炎炎夏日外面氣溫高達34度，它貼心詢問是否要開冷氣，我被它的舉動給嚇到，但好友一臉淡定的說不用，因為我們只是短暫待一下，但裡面很悶，好友只說開窗和開電扇，本來以為需要找尋開關來觸發，但出乎意料的是管家自動為我們開好了，這次的拜訪我下定決心要去了解物聯網。

我們希望透過這次的機會，來實現當初看到的震撼感，我們需要對IOT有基本的概念以及認知，並學習IOT的相關專業知識，進一步做出一個完整功能的智慧管家，與日常生活息息相關的智慧管家，能讓使用者在沙發上坐著或床上躺著就能輕鬆控制家中電器，免去不必要的活動，或是在還沒回到家時先開啟家中電器，使人回到避風港時能夠馬上休息，省去等待的時間。

參、主題與課程之相關性或教學單元之說明:

表1:主題與課程相關性

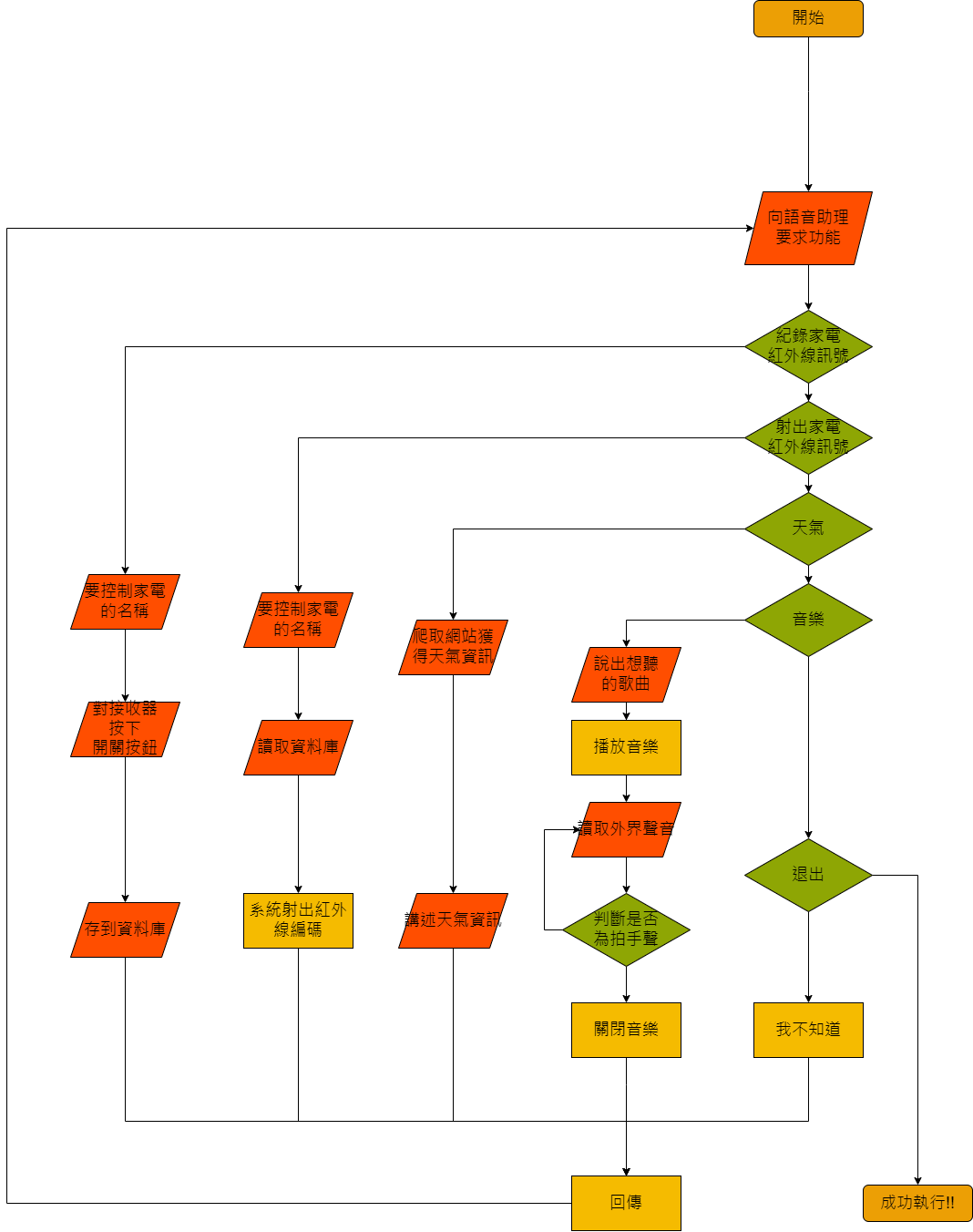
| 課程名稱 | 專題應用 |
| --- | --- |
| 1.微處理機實習 | ESP32應用 |
| 2.基本電學實習 | 紅外線發射與接收電路 |
| 3.微電腦應用實習 | 樹梅派應用 |
| 4.電子學實習 | 紅外線發射電路 |
| 5.程式設計實習 | 程式編寫 |

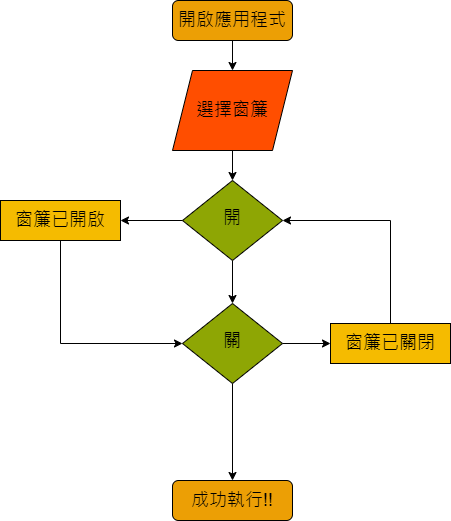
肆、研究過程及制作方法:

一、系統架構圖

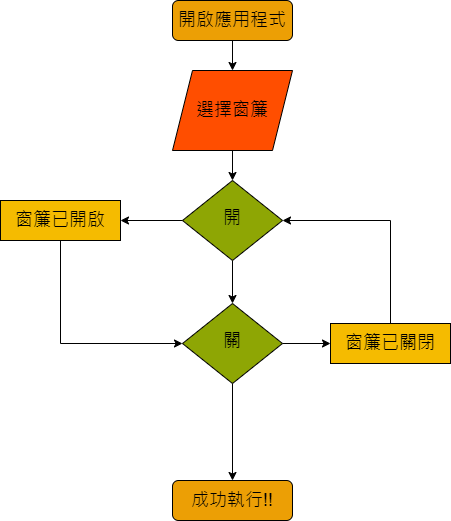
本專題使用ESP32的WiFi和藍芽兩種無線連接方式，再加上紅外線的控制，分別去控制電燈、窗簾以及電風扇，讓使用者能夠在不移動位置的情況下，對家中的電器做出控制，比較特別的是，我們在紅外線的部分多添加了以Raspberry Pi做主機的語音助理，讓原本單調的紅外線控制增添了一種味道。

1.語音控制流程圖



2.電燈開關流程圖

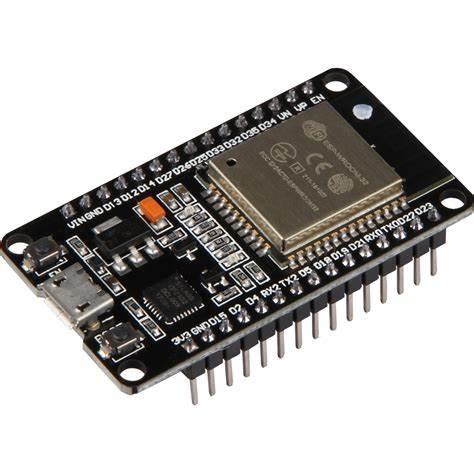
3.窗簾開關流程圖



二、硬體介紹

(一)ESP32

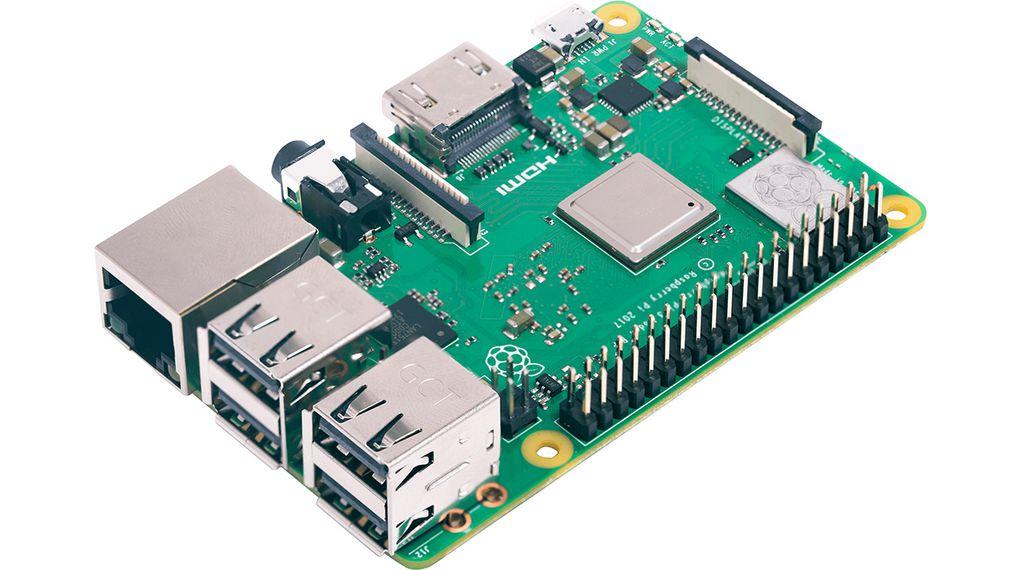
低功耗的單晶片微控制器,整合了 Wi-Fi 和藍牙功能。ESP32 系列採用 TensilicaXtensa LX6 微處理器,包括雙及單核變體,內建天線開關,RF變換器,功率放大器,低雜訊接收放大器等模組。同時，它也是本專題使用最多的元件





(二)Raspberry Pi3

英國樹梅派基金會所開發的微型單板電腦，主要以TF卡作為系統儲存媒體配備USB介面和HDMI的視訊輸出，內建有Ethernet、WLAN、Bluetooth網路連接的方式樹莓派系列電腦每一代均使用博通（Broadcom）出產的ARM處理器，如今生產的機型記憶體在2GB和8GB之間





(三)步進馬達(28BYJ-48)及控制模組(ULN2003A驅動板)

1.步進馬達

無刷直流電動機，它將一個完整的旋轉分成多個相同的步長可以命令電機的位置移動並保持在這些步驟之一，而無需任何船趕氣進行反饋，只要電機在扭距和速度方面的尺寸適合就好。

2.ULN2003A驅動板

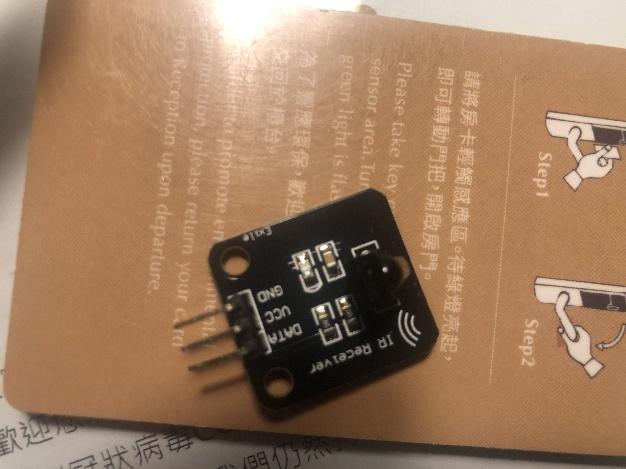
高壓大電流的共射極達靈頓電晶體，內存7組達林頓對管每組對管的電流容量是500mA，輸出的電壓最高50V。它還整合了7隻共陰極形式連接的續流二極體，用於電感性負載的開關動作的電流續流。





(四)紅外線發收(LED-940R-5T)及接收器

利用紅外線來傳遞數據，是無線通訊技術的一種。不需要實體連線，簡單易用且實現成本較低，因而廣泛應用於小型移動設備互換數據和電器設備的控制中，例如筆記本電腦、個人數碼助理之間或與電腦之間進行數據交換，電視機、空調的遙控器等。由於紅外線的直射特性，紅外通訊技術不適合在障礙較多的地方使用，大多數情況下傳輸距離短、傳輸速率不高







(五)麥克風

DC4.5V，頻率10HZ~10KHZ的全指向型麥克風，靈敏度-58+-20db

(六)藍芽喇叭

我們在語音助理的輸出設備用的是金冠K88藍芽音箱，除了藍芽之外，K88自帶MP3解碼器，支持TF卡音樂播放、免持電話、睡眠助理以及AUX接口接駁，內置3.7V/3600Ah鋰電池，是個十分方便的小型藍芽音箱

三、軟體介紹

(一)Arduino

Arduino是一個使用JAVA、C、C++編寫程式的跨平台應用程式提供簡單的一鍵式編譯程序並將程序上傳到 Arduino 板的機制。它還包含一個消息區、一個文本控制台、一個帶有常用功能按鈕的工具欄和一個層次結構的操作菜單。也是我們在本專題使用最多次的軟體





(二)App Inventor2

App Inventor 2是使用簡單的拼圖來組合程式，進而發展Android裝置的應用程式，網路上有需多教學資源，幫助使用者開發，開發完成後，就可以自行下載使用或者上傳 Google Play 商店提供大眾下載。





(三)Python

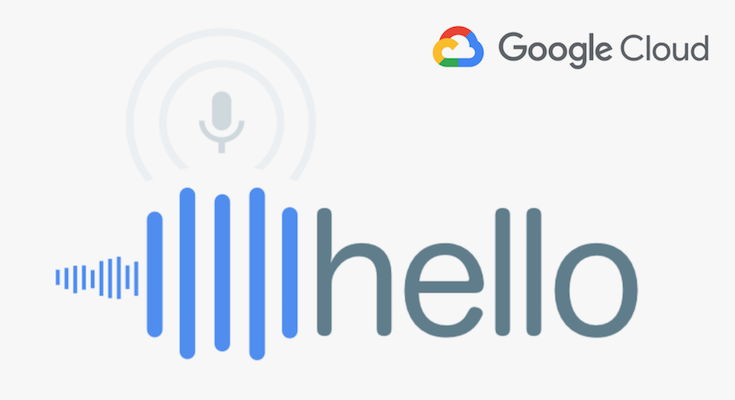
一種廣泛使用直釋式、進階和通用的程式語言，Python支援多種程式設計範式它擁有動態型別系統和垃圾回收功能，也能夠自動管理記憶體使用，並且其本身擁有一個巨大的標準庫。它的語言結構以及物件導向的方法，為了幫助程式設計師為小型的和大型的專案編寫邏輯清晰的程式碼。





(四) Google Cloud Speech-to-Text

使用Google最先進的深度學習類神經網絡演算法，提供自動語言語音識別功能，且能透過Speech-to-Text的使用者介面及管理來進行實踐，甚至可以在任何需要位置設置自動語言語音識別功能技術：雲端環境使用API​​，終端部系統則是使用文字轉語音部於本地。

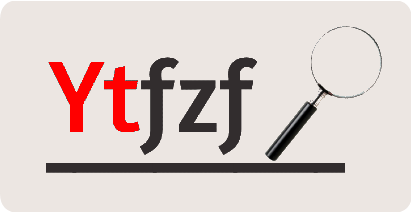




\

(五)YTFZF

YTFZF只是一個POXIC腳本，可以在不離開終端機的情況下搜尋、播放及下載Youtube影片，除此之外YTFZF還支援播放歷史、允許選擇媒體格式，並且可以對多個曲目進行排列。目前它可用於Unix的操作系統，包括Linux和macOS等作業系統



(六)MySQL

MySQL是一種免費的資料庫管理系統，大多搭配php、ASP或、ASP.NET等網頁程式储存大量數據，大多運用在中小型的網站當中。

(七)phpmyadmin

phpmyadmin最大的特點是能夠讓使用者在網頁中以圖形化介面的方式來直接管理資料庫，現今的MySQL資料庫常常搭配phpmyadmin來進行管理，兩者缺一不可。

(八)apache

apache是一種免費的網頁服務器，這項工具的目標在於提供一個安全、高效、可擴展的HTTP服務器。雖然經過20多年，他在全球仍然擁有數百萬用戶，可見這項工具相當可靠。



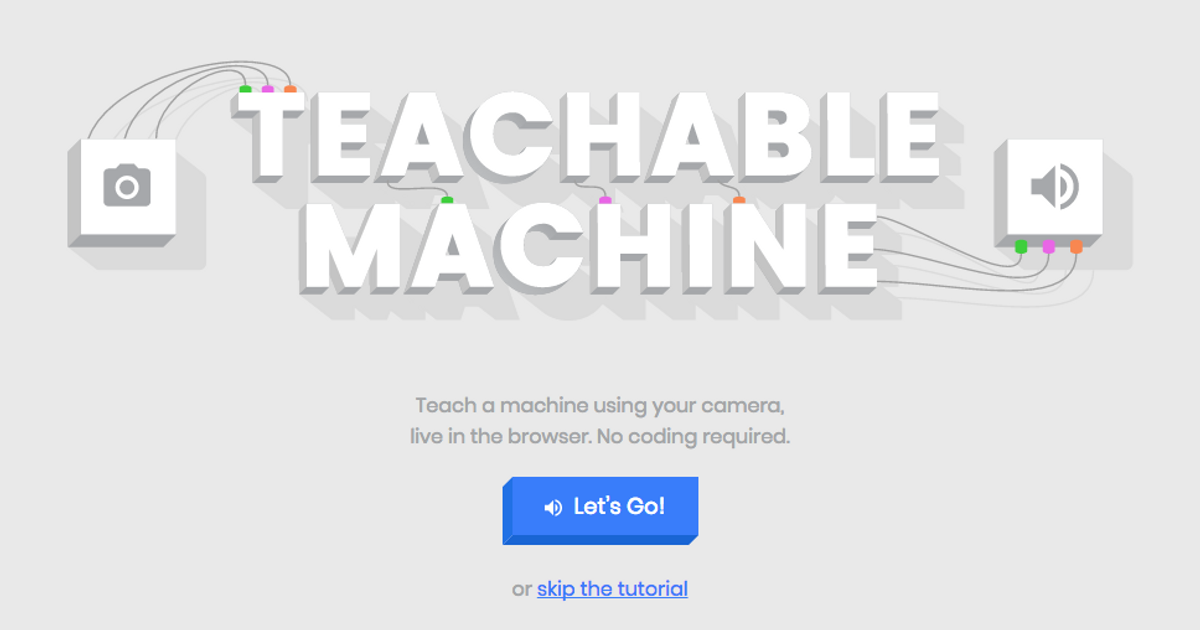
(九)php

php是一種免費的動態網頁開發語言，其特點為它相當優良的跨平台能力，無論是在Linux(最適合)、Unix、 Windows都可以執行運作，並且可以搭配多種資料庫伺服器，包括前面提到的MySQL。



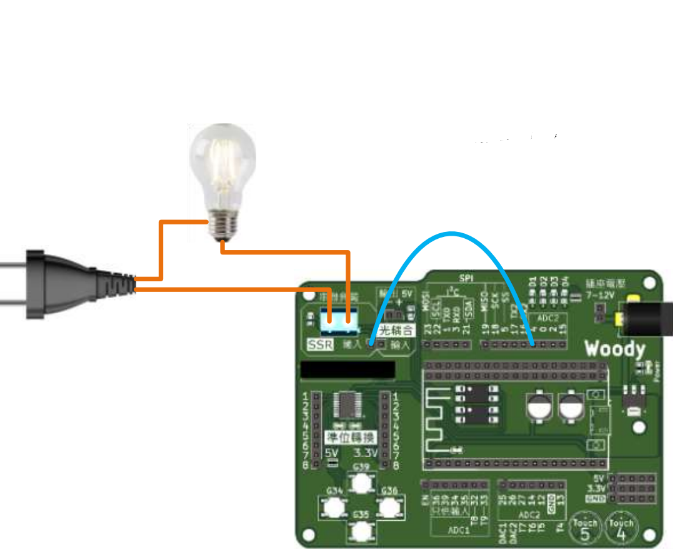
(十)Google Teachable Machine

Google Teachable Machine是由Google推出的網路線上機器學習平台，可以讓我們在硬體設備不佳的情況下，把資料交由Google雲端運算處理，而我們只需經過蒐集資料和訓練的步驟就可以輕鬆地建立自己的模型。



四、程式碼及電路圖

(一)WiFi電燈控制

1.電路圖

2.製作過程

我們先利用剪刀把購買來的電燈的其中一條線剪一半並搭配ESP32輔助開發板:Woody上的固態繼電器SSR來串連交流負載。

(實際電路)

3.程式碼

#include <WiFi.h> //引用Wi-Fi函式庫

#include <WebServer.h> //引用WebServer函式庫

const char \*ssid = "c8"; //無線網路基地台的SSID

const char \*password = "a2004931123"; //無線網路基地台的密碼

const byte LED = 2; //LED接腳為 GPIO 2

WebServer server(80); //建立伺服器物件，設定監聽 80 號埠

void handleRoot() { //處理程序：回覆根目錄請求

String HTML = "<a style='font-size:72px' href=\"/on\">開啟電燈</a>"; //開始組合HTML

HTML += "<BR>";

HTML += "<a style='font-size:72px' href=\"/off\">關閉電燈</a>";

server.send(200, "text/html; charset=UTF-8", HTML ); //送出訊息到客戶端

}

void handleNotFound() { //處理程序：未找到檔案時

server.send(404, "text/html; charset=UTF-8", "找不到檔案");

}

void handleLedOn() { //處理程序：點亮電燈

digitalWrite(LED, HIGH);

server.send(200, "text/html; charset=UTF-8", "<p style='font-size:72px'>電燈已開啟</p><BR>\

<a style='font-size:72px' href=/off>關閉電燈</a>");

}

void handleLedOff() { //處理程序：熄滅電燈

digitalWrite(LED, LOW);

server.send(200, " text/html; charset=UTF-8", "<p style='font-size:72px'>電燈已開啟</p><BR>\

<a style='font-size:72px' href=/on>開啟電燈</a>");

}

void setup() {

Serial.begin(9600);

pinMode(LED, OUTPUT);

Serial.println("1234");

WiFi.begin(ssid, password); //連線到無線基地台

while (WiFi.status() != WL\_CONNECTED) { //持續嘗試連線至成功為止

delay(500);

Serial.print(".");

}

Serial.print("已連線到基地台：");

Serial.println(ssid);

Serial.print("基地台配發給我的IP位址：");

Serial.println(WiFi.localIP()); //列出基地台配發給我的位址

server.on("/", handleRoot); //登錄處理程序：根目錄

server.onNotFound(handleNotFound); //登錄處理程序：未找到檔案

server.on("/on", handleLedOn); //登錄處理程序：點亮電燈

server.on("/off", handleLedOff); //登錄處理程序：熄滅電燈

server.begin(); //啟動伺服器物件

Serial.println("HTTP 伺服器已啟動");

}

void loop() {

server.handleClient(); //啟動客戶端請求處理程序

}

(二)藍芽窗簾控制

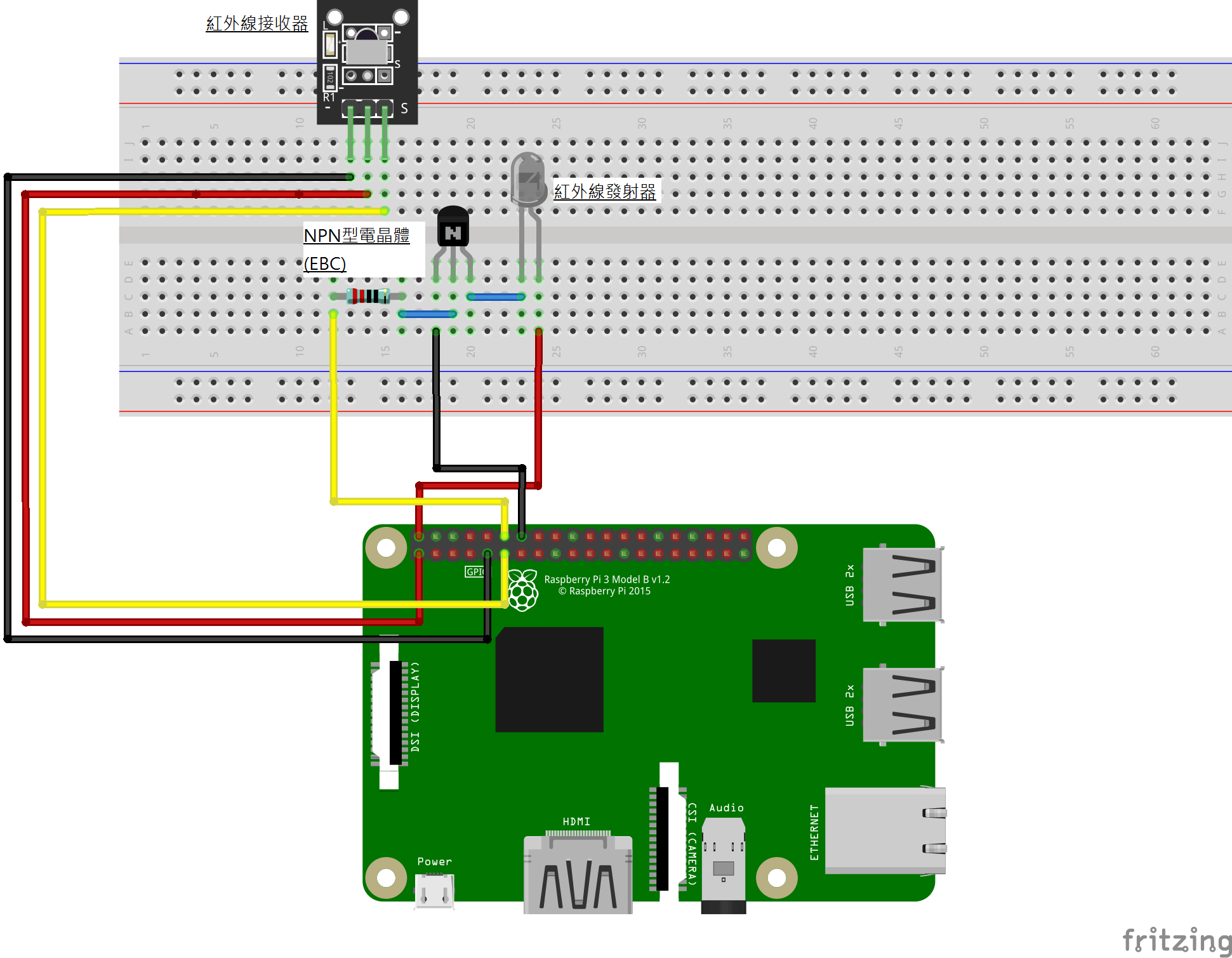
1.製作過程

由於時間問題，本想在新增一個資料庫能夠上傳使用紀錄，來不及每個系統都製作，指來得及在窗簾的app中製作這項功能，在app中輸入網址，上傳至資料庫，，再藉由phpadmin介面展示出來

2.程式碼

(三)紅外線電風扇控制

1.電路圖



2.製作過程

現今的Linux核心已經可以直接支援紅外線的接收與發射，在一開始的時候我們先輸入sudo nano /boot/config.txt這項指令，找到#dtoverlay=gpio-ir ,gpio\_pin=17(接收)和#dtoverlay=gpio-ir-tx,gpio\_pin=18(發射)這兩個之後開啟核心的支援(把前面的#號刪掉)完成之後重新開機。



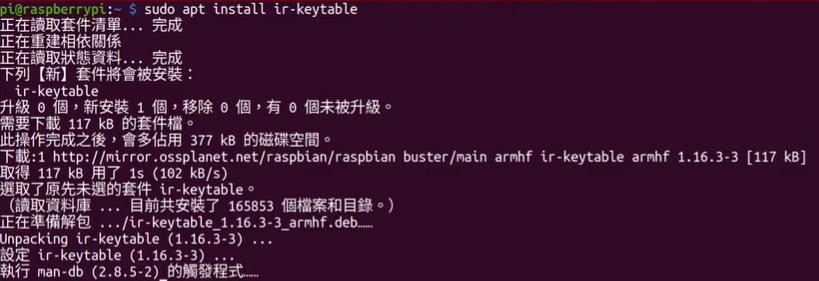
開啟前 (開啟後)

開機後再利用dmesg | grep -i rc確認接收與發射的代號

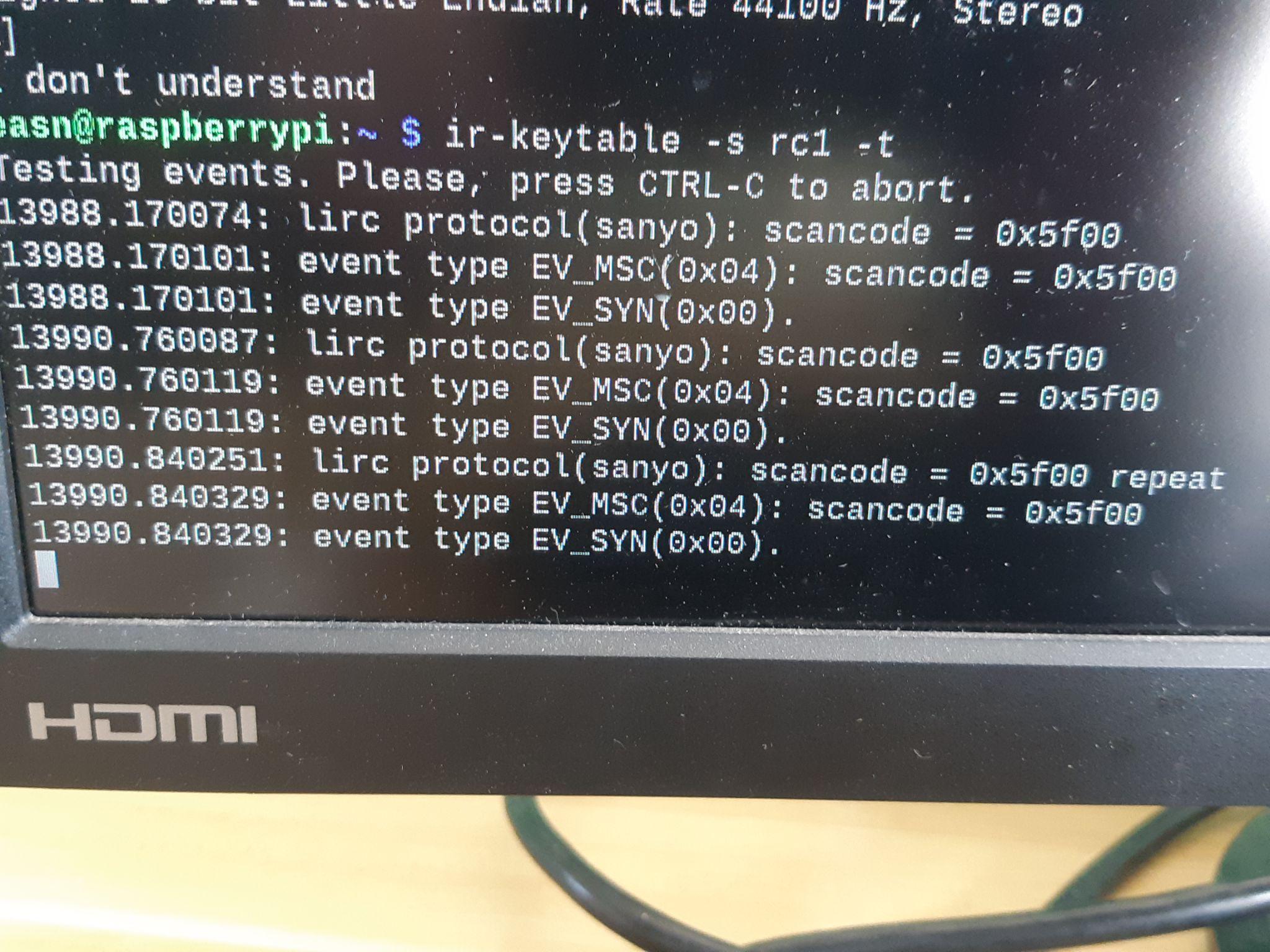
(rc1為接收) (rc0為發射)

之後再安裝ir-keytable這項套件，這項套件就是我們之後在做紅外線接收與發射時的主要工具，而在安裝好之後可以直接輸入ir-keytable來插看許多資訊，像是樹莓派上接的元件的代號(rc0、rc1)以及驅動程式還有它所支援的傳輸協定等等有用的資訊。

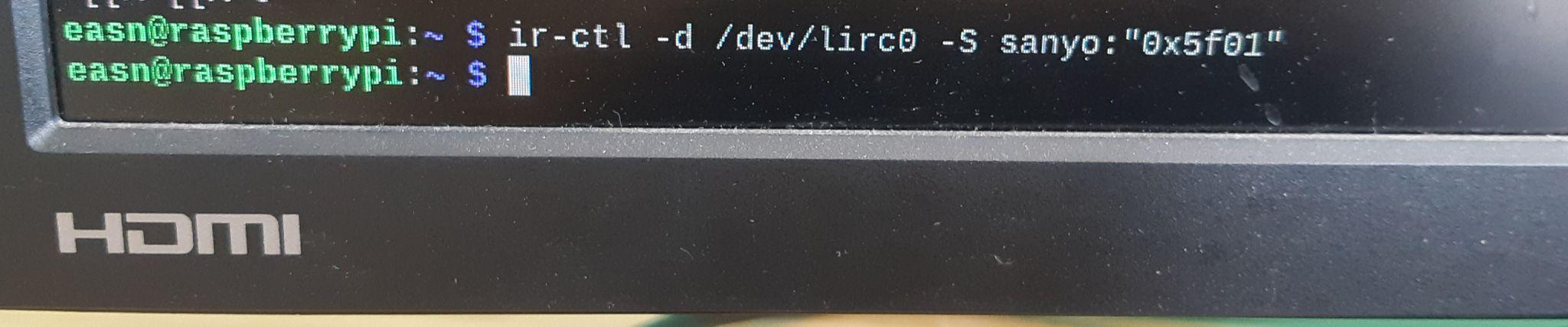


(安裝ir-keytable)

再來就是利用ir-keytable -s rc1 -t這項指令來抓取輸入的紅外線協議與編碼，並再用ir-ctl -d /dev/lirc0 -S sanyo(協議名稱):”0x5f00(編碼)”透過紅外線發射器發射紅外線訊號。



(讀取，協議為sanyo編碼為0x5f00)



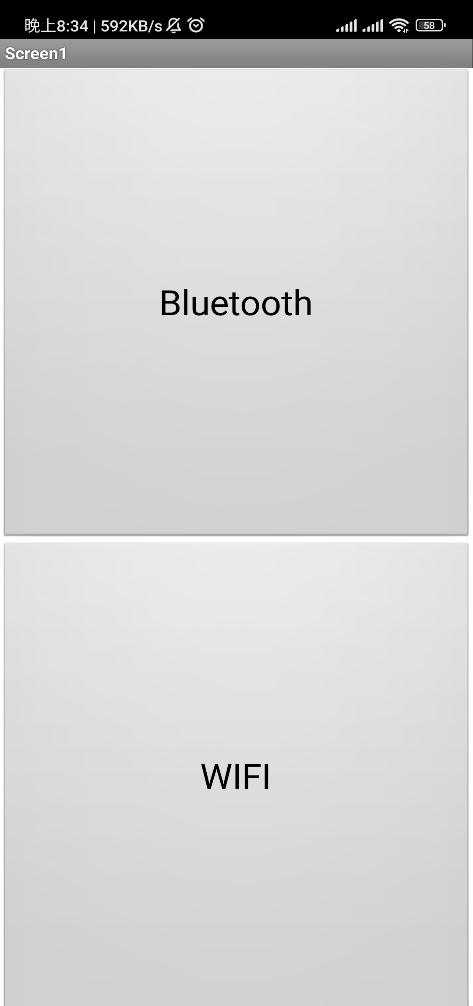
(發射)

(四)語音及紅外線整合

1.製作過程

2.程式碼

五、App UI







伍、研究結果

陸、討論

一、問題與解決方法

紅外線

一開始我們在做紅外線的接收與發射時是使用Arduino uno以及ESP32這兩塊板子來做，畢竟有關Arduino紅外線接收與發射的資料網路上也比較豐富，但後來要整合時才發現Arduino uno不像ESP32有內建WiFi的功能，而那時也因為時間與錢的關西沒有再去買而外的WiFi模組，所以我們後來直接開始找有關樹莓派紅外線發射與接收的資料才克服這個問題。

由於時間問題，本想在新增一個資料庫能夠上傳使用紀錄，來不及每個系統都製作，指來得及在窗簾的app中製作這項功能，真的十分可惜

二、未來期望

柒、結論

捌、參考資料及其他:

1.[物聯網](https://zh.wikipedia.org/zh-hans/%E7%89%A9%E8%81%94%E7%BD%91)

<https://aws.amazon.com/tw/what-is/iot/>

2.[IEEE 802.11](https://zh.wikipedia.org/wiki/IEEE_802.11)

<https://zh.wikipedia.org/wiki/IEEE_802.11>

3.[無線區域網路](https://zh.wikipedia.org/wiki/%E7%84%A1%E7%B7%9A%E5%8D%80%E5%9F%9F%E7%B6%B2%E8%B7%AF)

<https://zh.wikipedia.org/wiki/%E6%97%A0%E7%BA%BF%E5%B1%80%E5%9F%9F%E7%BD%91>

4.藍芽

https://zh.m.wikipedia.org/zh-hant/%E8%97%8D%E7%89%99

5. Google Cloud Speech-to-Text

https://cloud.google.com/speech-to-text/

6.App Inventor2

https://appinventor.mit.edu/

7.Fritzing

<https://fritzing.org/>

8.ESP32藍芽連線教學

<https://shop.mirotek.com.tw/iot/esp32-start-19/>

9.Google Teachable Machine

https://blog.cavedu.com/2022/12/03/teachable-machine-audio\_classify/

玖、附錄

| 年 | 月 | 日 | 進度 | 紀錄 | 工作分配 |
| --- | --- | --- | --- | --- | --- |
| 110 | 9 | 5 | 查閱歷屆得獎作品 | ：資訊科實習工廠  器材：電腦 時數：3 小時 | 14、18、32:  查資料、討論 |
| 110 | 9 | 11 | 討論及制定主題  製作報告 | 線上討論及製作 | 14、18:資料查詢、討論  32:報告製作、討論 |
| 110 | 9 | 12 | 第一次分組報告 | 科實習地點:資訊科實習工廠  時數：3 小時 | 14、18、32:  上台報告 |
| 110 | 9 | 19 | 分配工作主題  研究專題架構  軟體規劃  材料清單 | 地點:資訊科實習工廠  時數：3 小時 | 18:分配工作、專題架構  14、32:軟體規劃、專題架構 |
| 110 | 9 | 23 | 材料購買 | 地點:台中電子街 | 14、18、32:購買材料 |
| 110 | 9 | 26 | 製作專題(WiFi、紅外線) | 地點:資訊科實習工廠  時數：3 小時 | 18、32:編寫程式碼(WiFi)  14:編寫arduino程式碼  (紅外線接收) |
| 110 | 10 | 3 | 電路製作及程式編寫(WiFi、紅外線)  WiFi功能測試 | 地點:資訊科實習工廠  時數：3 小時 | 18、32:編寫程式碼(WiFi)、電路製作  14:編寫arduino程式碼、電路製作  (紅外線發射) |
| 110 | 10 | 10 | 製作專題本體(藍芽)  紅外線架構重修 | 地點:資訊科實習工廠  時數：3 小時 | 18、32:編寫程式碼(藍芽)、電路製作  14、18:討論紅外線問題 |
| 110 | 10 | 17 | 程式編寫(藍芽) |  | 18、32:編寫android app(藍芽)  14:確診在家休息 |
| 110 | 10 | 24 | 電路製作及程式編寫(藍芽、紅外線)  藍芽功能測試  測試紅外線發射功能 |  | 32:編寫android app(藍芽)、測試功能  14、18:程式碼編寫(紅外線接收及發射)、電路製作 |
| 110 | 10 | 31 | 電路製作及程式編寫(紅外線) |  | 14、18:程式碼編寫(紅外線接收)  32:報告製作 |
| 110 | 11 | 7 | 分組報告製作 |  | 14、18、32:報告製作 |
| 110 | 11 | 14 | 第二次分組報告 |  | 32、14、18:  上台報告 |
| 110 | 11 | 21 | 電路製作及程式編寫(紅外線)  紅外線接收功能測試 |  | 32:實體窗簾製作  14、18:程式碼編寫(紅外線接收)、電路製作 |
| 110 | 11 | 28 | 電路製作及程式編寫(紅外線、實體窗簾)  紅外線接收功能測試 |  | 32:實體窗簾製作  14、18:程式碼編寫(紅外線)、電路製作 |
| 110 | 12 | 5 | 程式編寫(語音整合)  專題簡報製作  模型外殼設計 |  | 14:查詢資料、程式碼編寫  18:模型外殼設計  32:專題簡報 |
| 110 | 12 | 9 | 模型材料購買 |  | 18:單獨採買 |
| 110 | 12 | 12 | 程式編寫(語音整合)  專題簡報製作  模型外殼製作 |  | 14:訓練模型、程式碼編寫  18:模型外殼製作  32:專題簡報 |
| 110 | 12 | 19 | 分組報告製作  語音整合測試 |  | 14:語音整合測試  18、32:分組報告製作 |
| 110 | 12 | 26 | 第三次分組報告 |  | 14、18、32:上台報告 |
| 110 | 12 | 31 | 優化程式碼  跨年 |  | 14、18、32:程式碼註解 |
| 110 | 1 | 2 | 專題簡報製作 |  | 14、18、32:專題簡報製作 |
| 110 | 1 | 8 | 專題簡報製作 |  | 14、18、32:專題簡報製作 |
| 110 | 1 | 9 | 專題簡報製作 |  | 14、18、32:專題簡報製作(最終確認) |