**Internet of Things and Microprocessor System Design**

**Final Project Report**

**搭載在汽車上的智慧語音助理**

**壹、成員名單及分工 :**

1083307 邵悅慈 寫程式、做報告

1083315 吳承軒 寫程式、做報告

1083321 葉品和 寫程式、做報告

**貳、專題功能說明 :**

* 車輛移動功能總共有 4 個關鍵字 :

1. **前進** : 語音助理說"前進中"，螢幕印出"前進中"，車車向前移動。
2. **左轉** : 語音助理說"車輛左轉"，螢幕印出"車輛左轉"，車車向左轉。
3. **右轉** : 語音助理說"車輛右轉"，螢幕印出"車輛右轉"，車車向右轉。
4. **後退** : 語音助理說"倒車中"，螢幕印出"倒車中，後方車輛請注意"，車車向後移動。

* 提醒功能能讓使用者設置一個定時的提醒，當時間到時會提醒使用者對應的事項。

關鍵字為 : **\_\_分\_\_秒後提醒我\_\_\_**，如果關鍵字沒有**分**，則預設為 0。

當語音助理收到訊息後，螢幕上會印出"已設定\_\_秒後的提醒" (**分**會被轉成**秒**)，並開始倒數計時，倒數計時的期間，語音助理能繼續聽指令並執行其他功能，當倒數計時結束後，語音助理會說"記得\_\_\_"。

* 音樂功能主要分為 3 個部分 :

1. **隨機放音樂** 或 **隨機播音樂** 或 **隨機播放** : 系統會從音樂庫裡面選擇一首歌曲，語音助理說"現在播放\_\_\_"，螢幕印出"現在播放\_\_\_"，並開始播放音樂。
2. **播放\_\_\_** 或 我要**聽\_\_\_** : 能播放指定的歌曲，螢幕上印出"播放\_\_\_"，如果系統找尋不到歌曲，語音助理會說"查無歌曲"，螢幕印出"查無歌曲"。
3. **停止播放** : 將播放中的歌曲停止。

ex. 可以用鍵盤上的 + 與 - 控制音量大小，以空白鍵暫停/繼續歌曲。

* 清單功能共有 2 種清單 :

1. **音樂清單** : 會將音樂庫的所有歌曲列出，語音助理說"第\_\_首是\_\_\_"，螢幕印出歌曲名稱。
2. **指令清單** : 語音助理會將所有指令說出，螢幕上也會列出所有指令。

* 時間功能能讓使用者得知現在的時間。

關鍵字為 : **現在幾點** 或 **現在時間** 或 **現在的時間。**

系統會以年、月、日、小時、分鐘、秒數的格式將現在的時間讓語音助理說出。

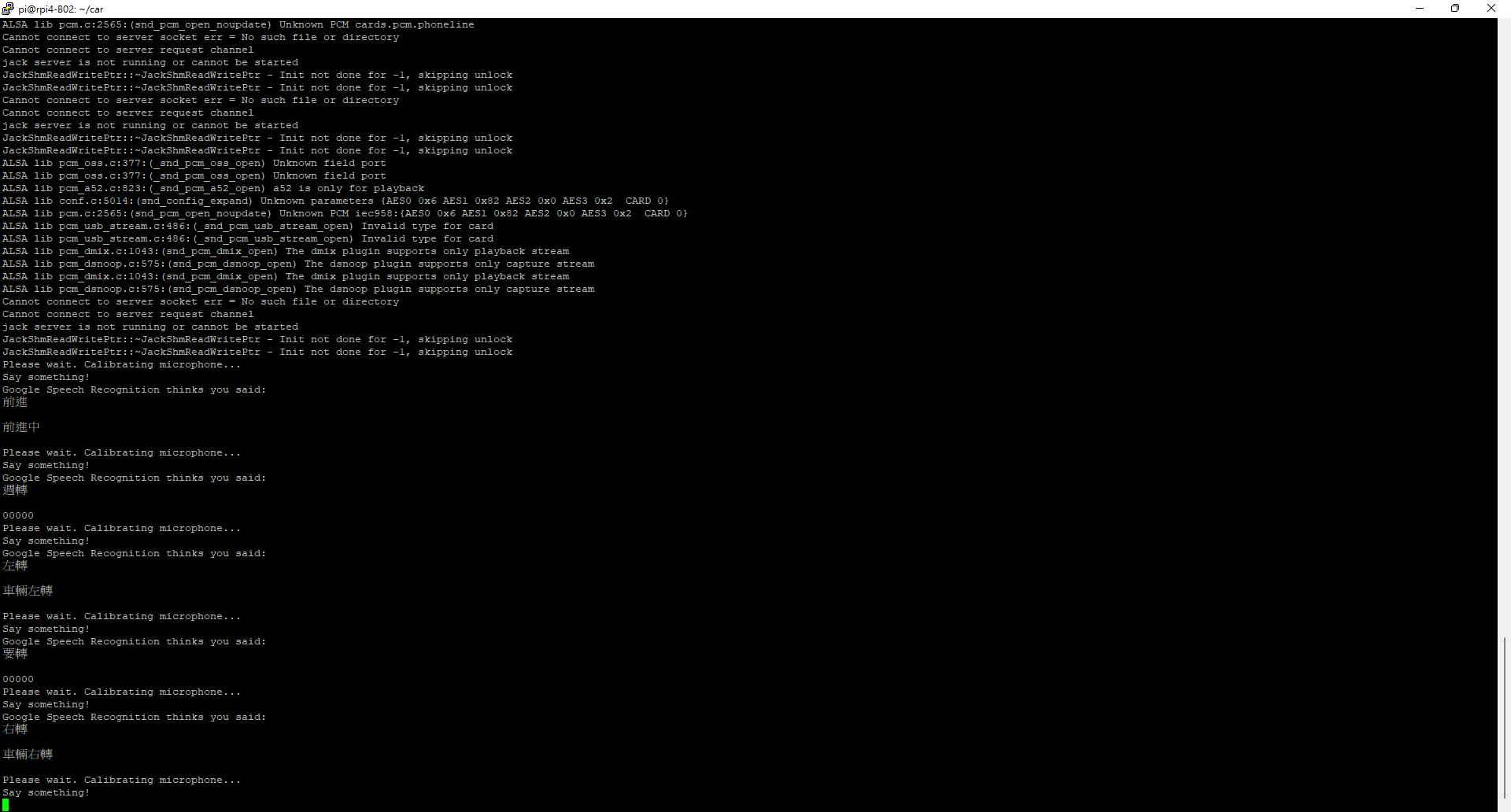
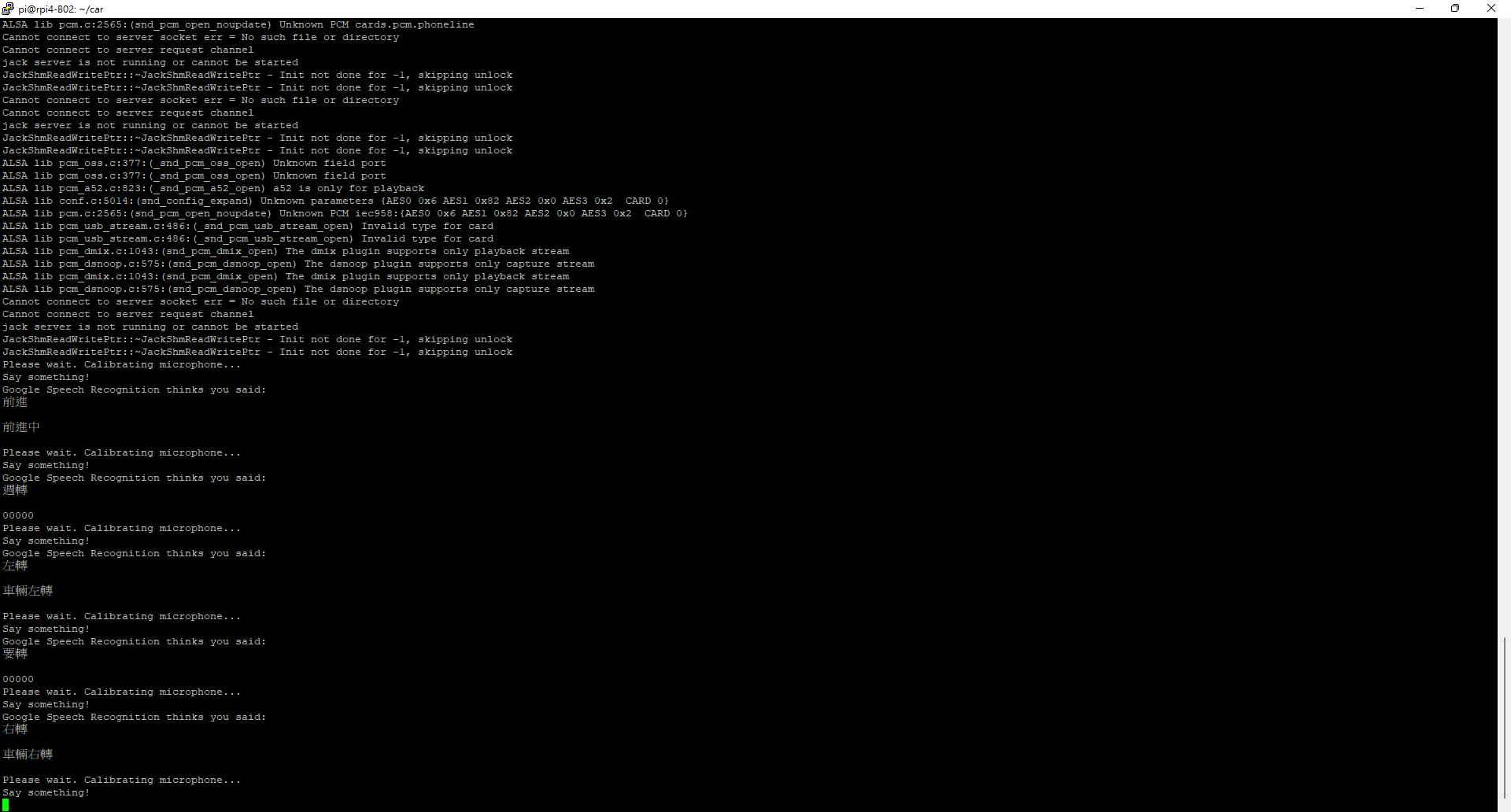
* 資料庫功能會將使用者使用過的指令進行紀錄，並上傳至 Ubidots 的資料庫。

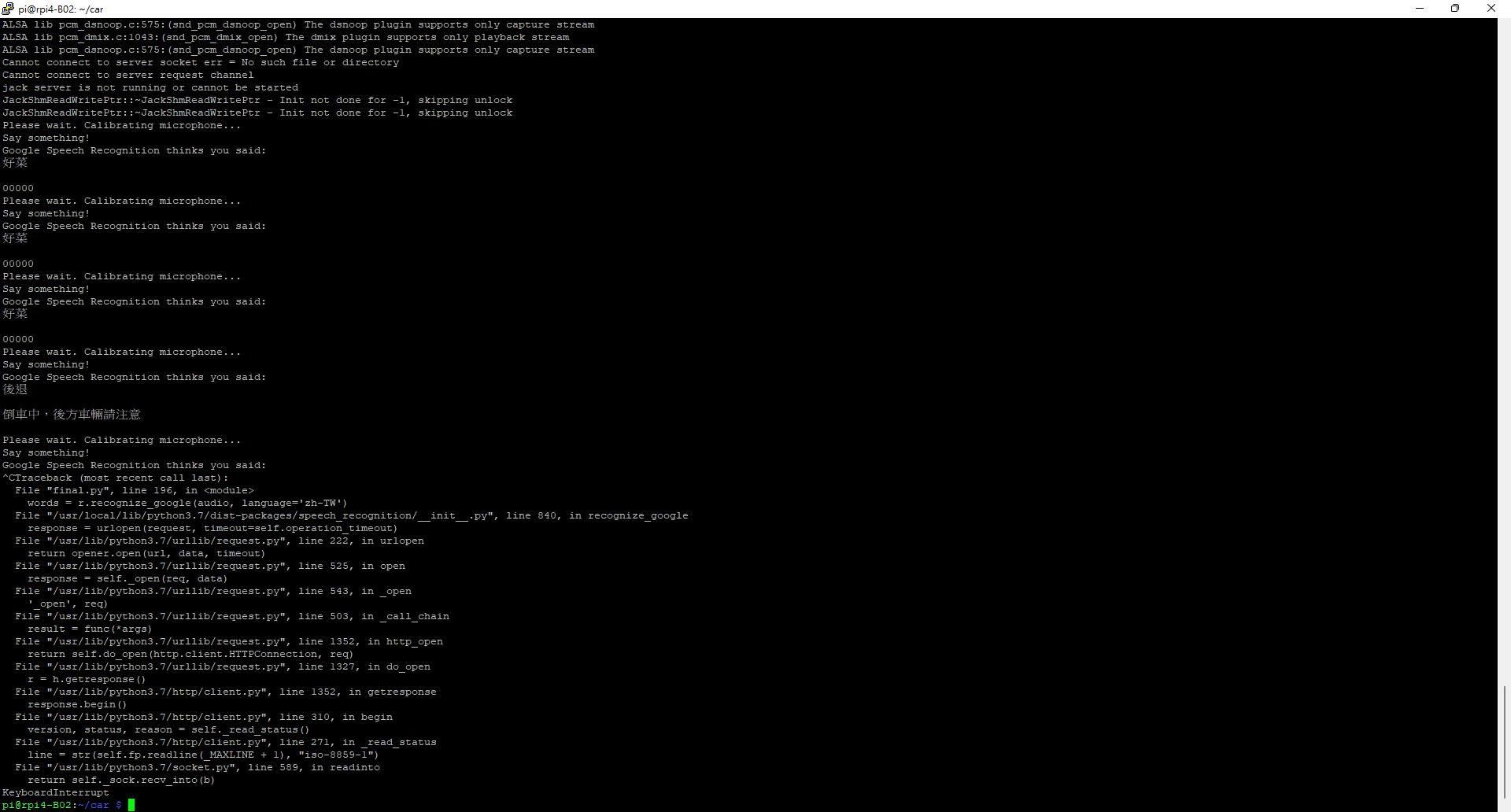
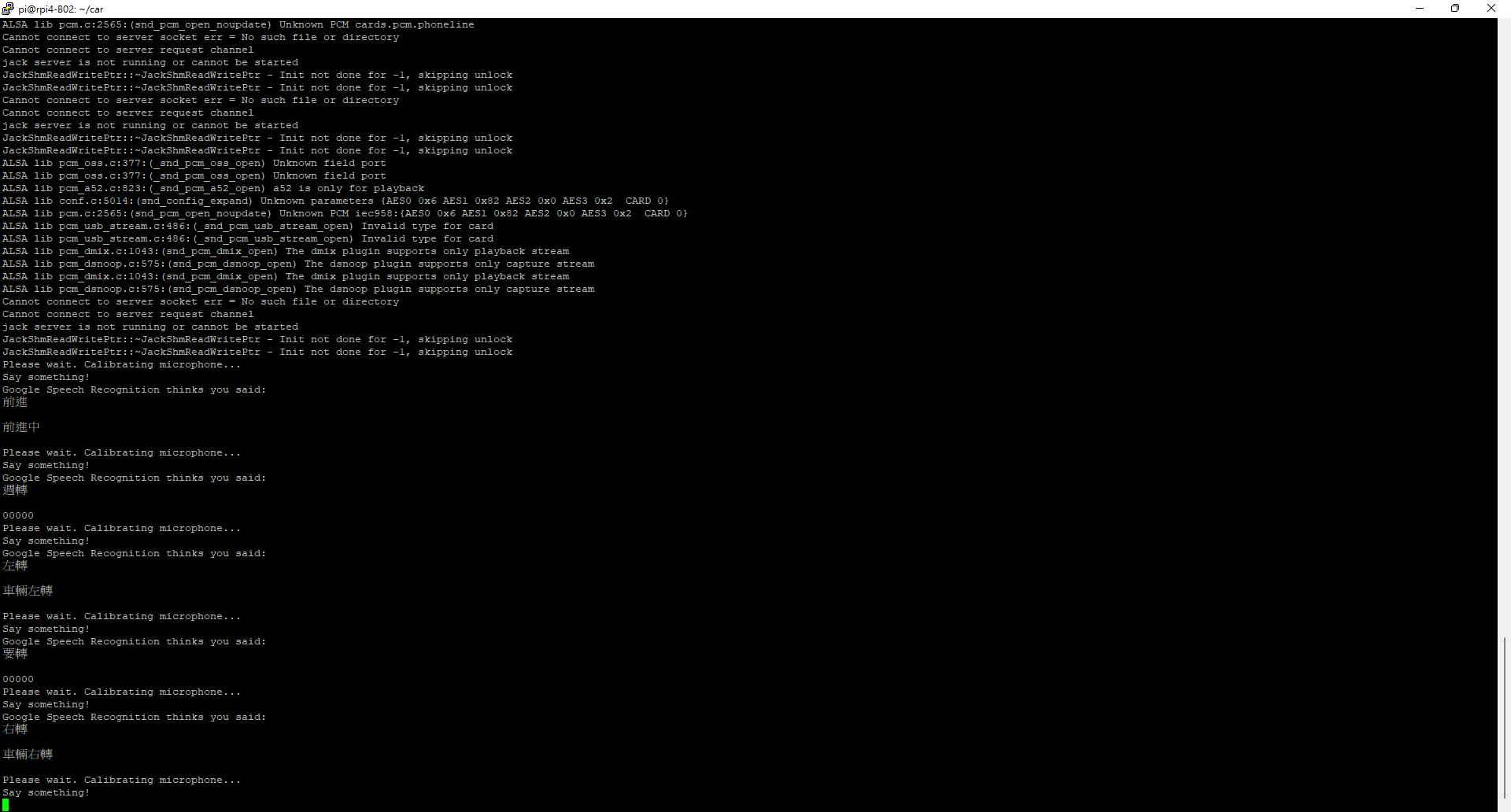
當程式執行時，系統會先從資料庫將資料抓下來，進行儲存。與資料庫相關的指令共有 2 種 :

1. **清空資料庫** : 將資料庫所有指令執行的次數都重置為 0。
2. **結束** : 將程式結束並將目前的資料上傳至資料庫。

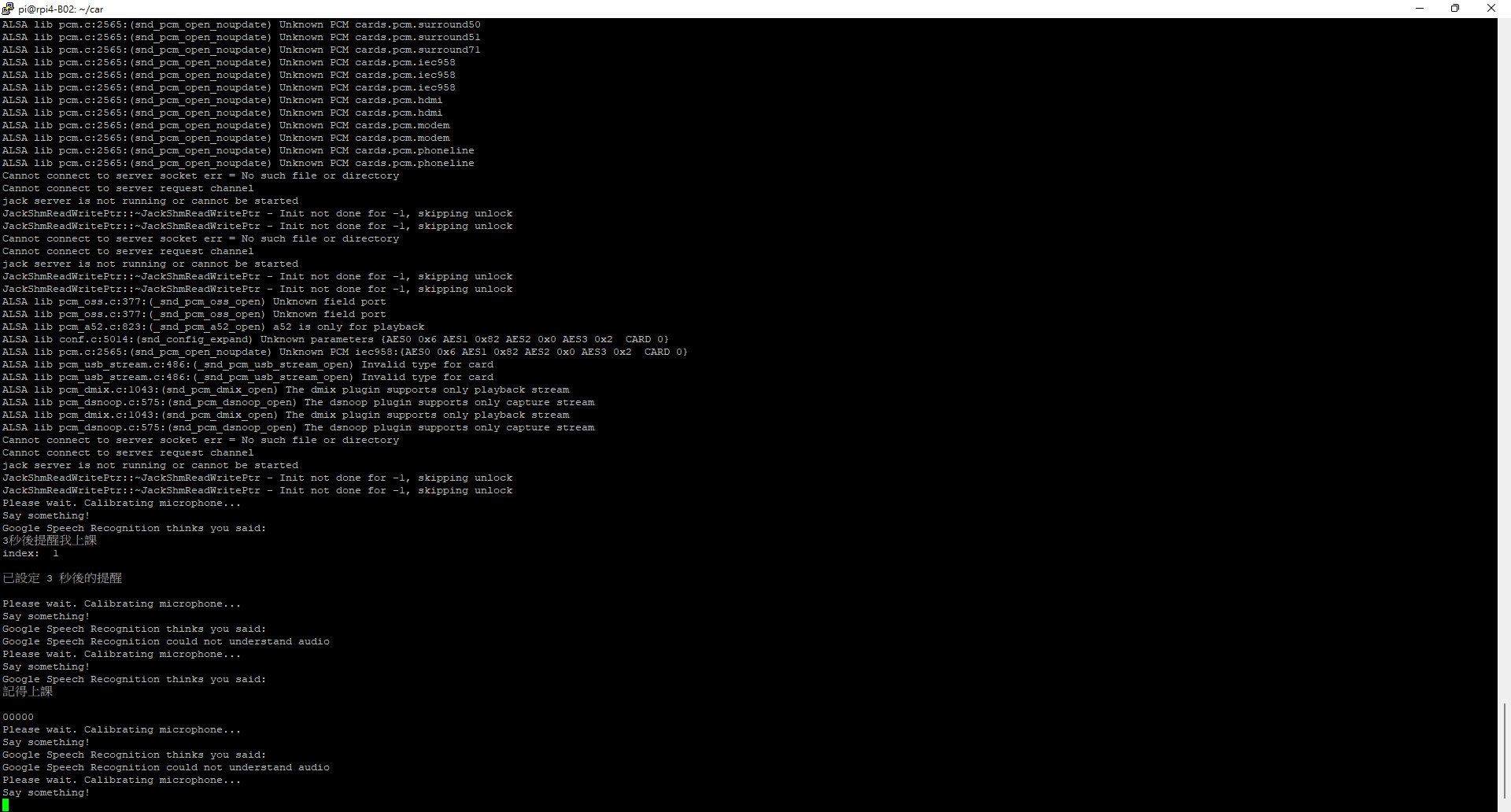
* 網頁顯示功能，系統會 post 一個 host 為 raspberry pi 的 ip、port 為 8000 的網頁，每次重新整理網頁時，系統會先從 Ubidots 資料庫將目前的指令執行次數抓下來，並傳到網頁的 html 檔，網頁會將這些數據繪製成圓餅圖與長條圖，提供視覺化的統計次數。

**叁、專題操作與執行流程 :**

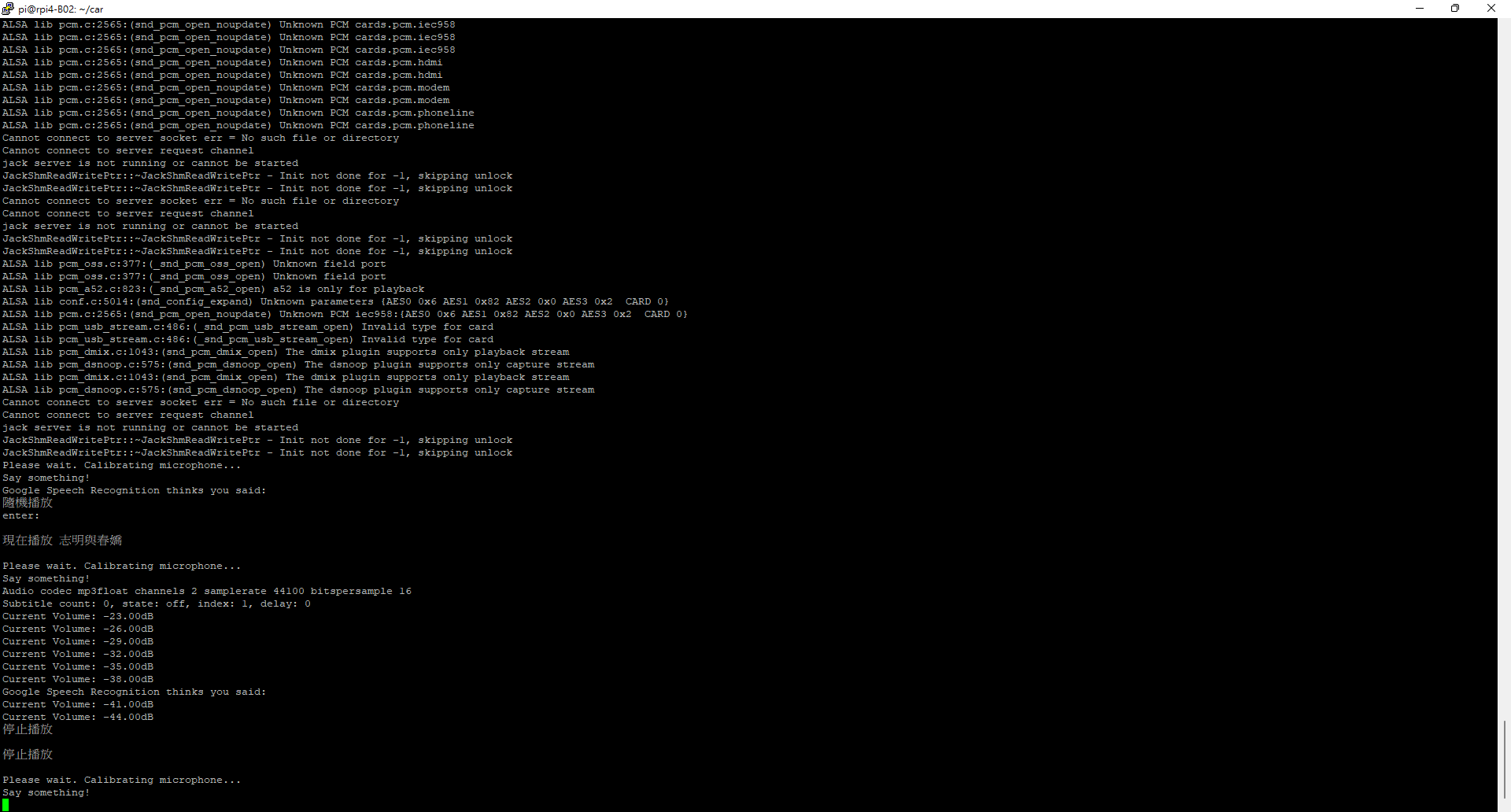
****

****

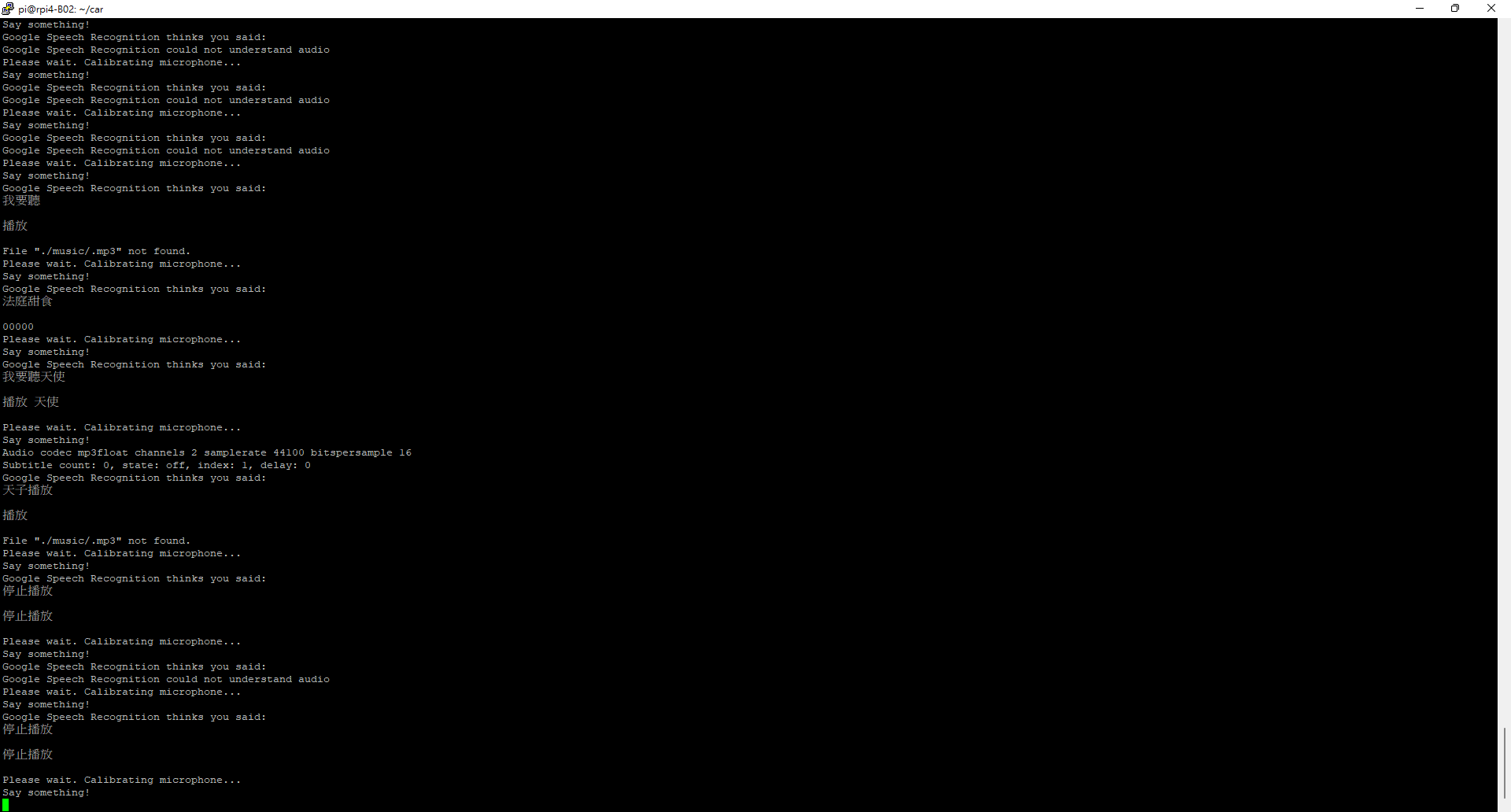
**車輛行動**

****

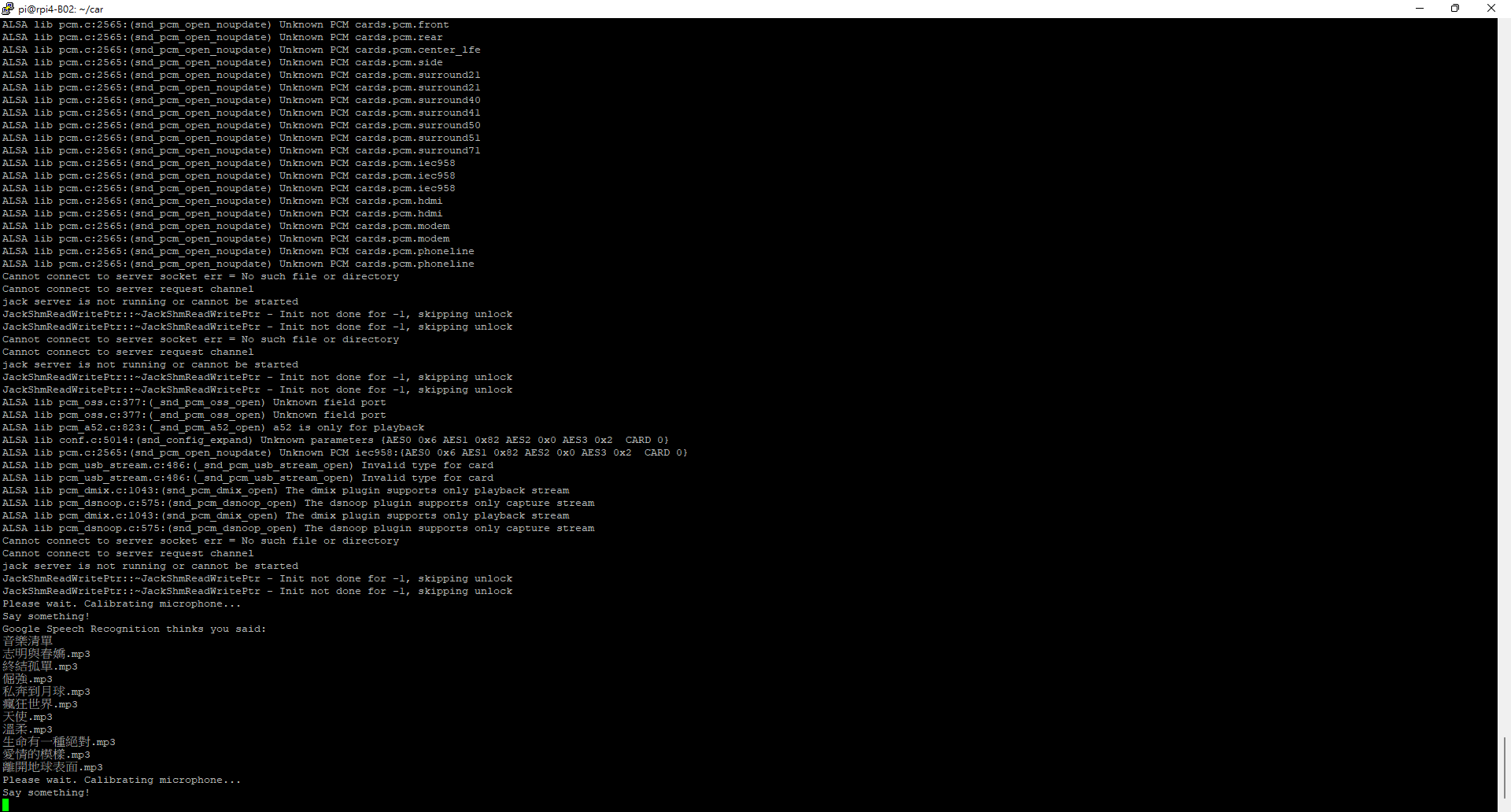
**設定提醒**

****

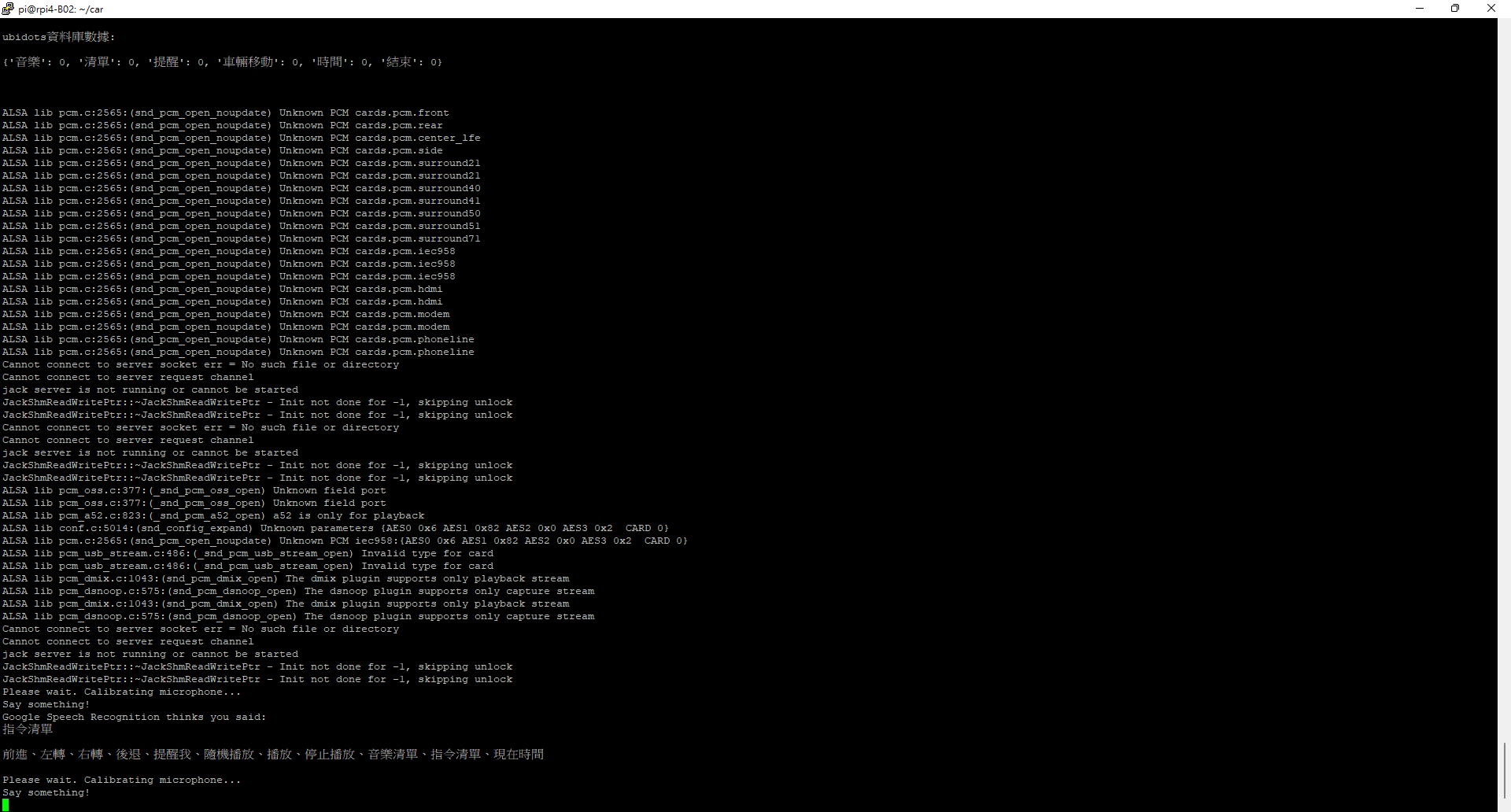
**隨機播放、停止播放**

****

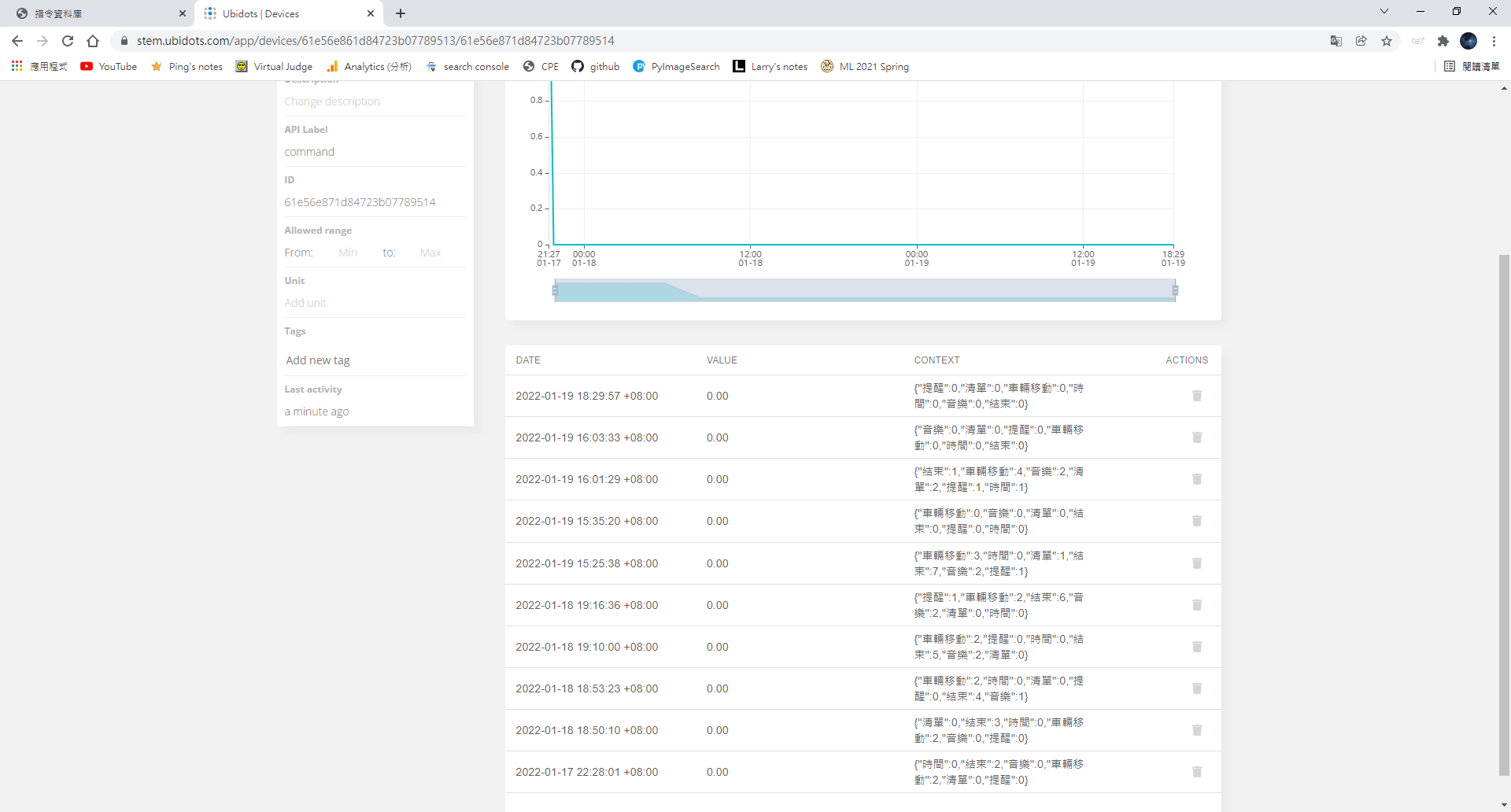
**指定歌曲播放、停止播放**

****

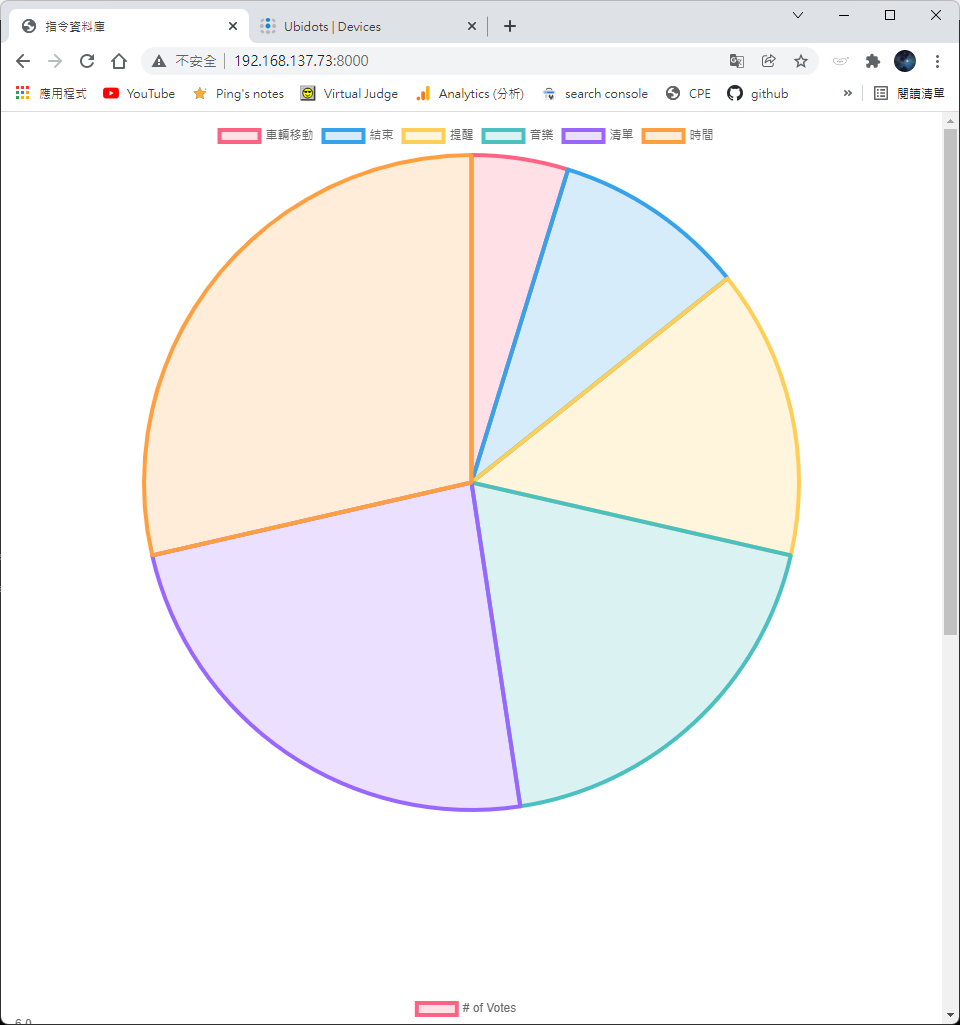
**音樂清單**

****

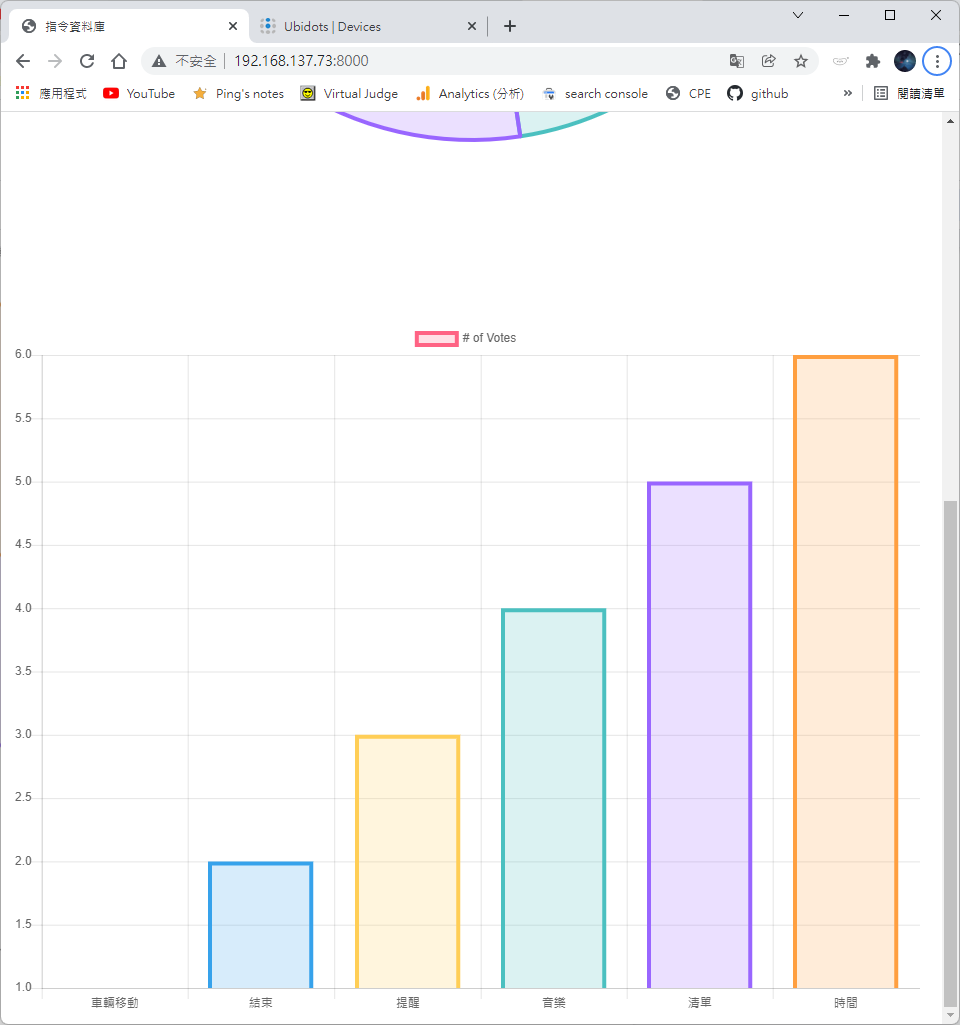
**指令清單**

****

**資料庫紀錄指令**

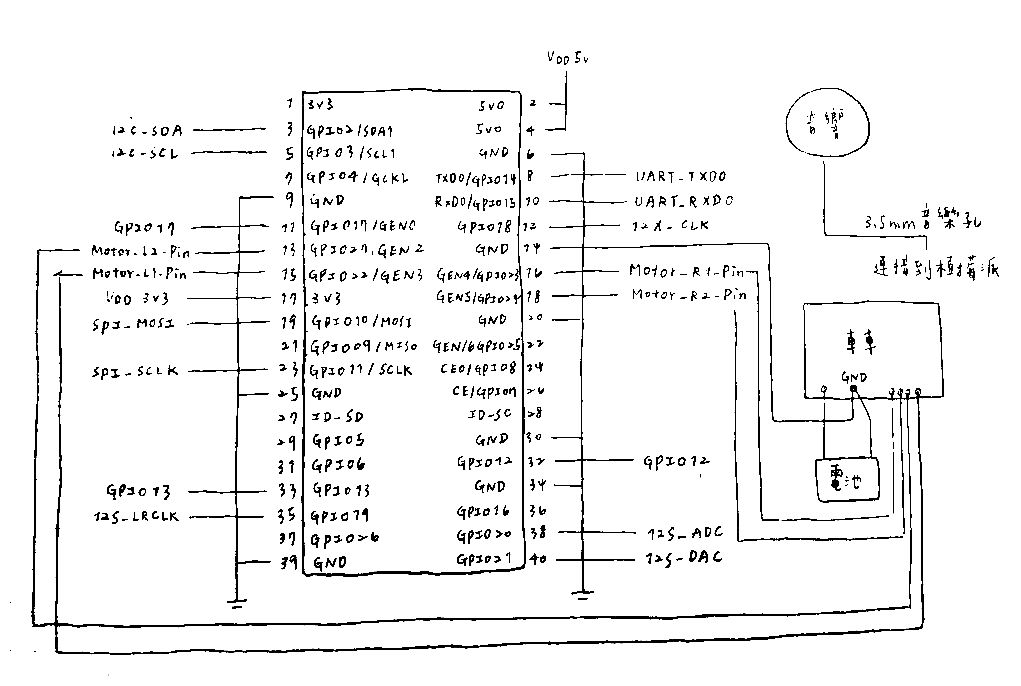
****

**繪製圓餅圖**

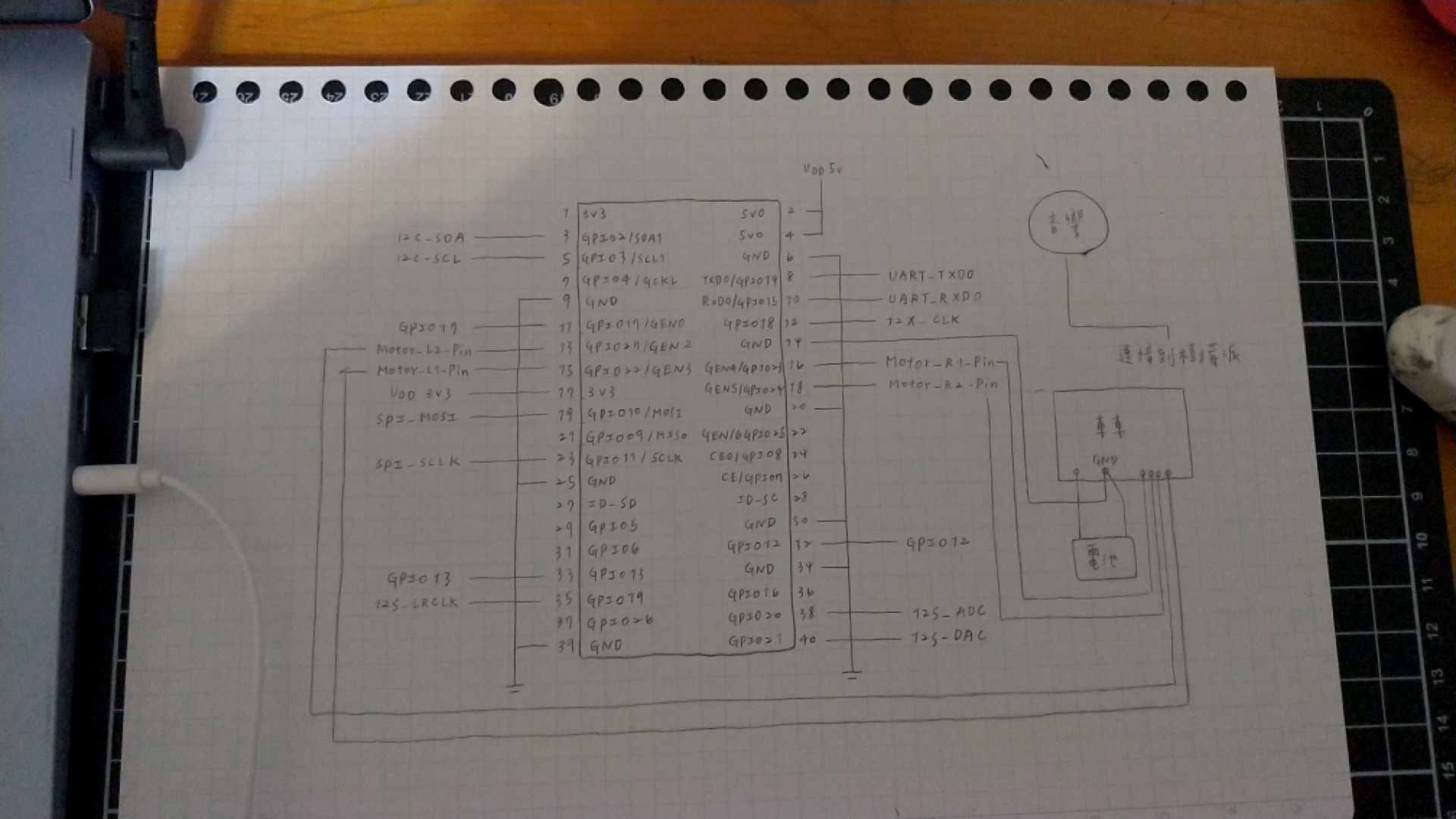
****

**繪製長條圖**

**肆、硬體電路示意圖 :**

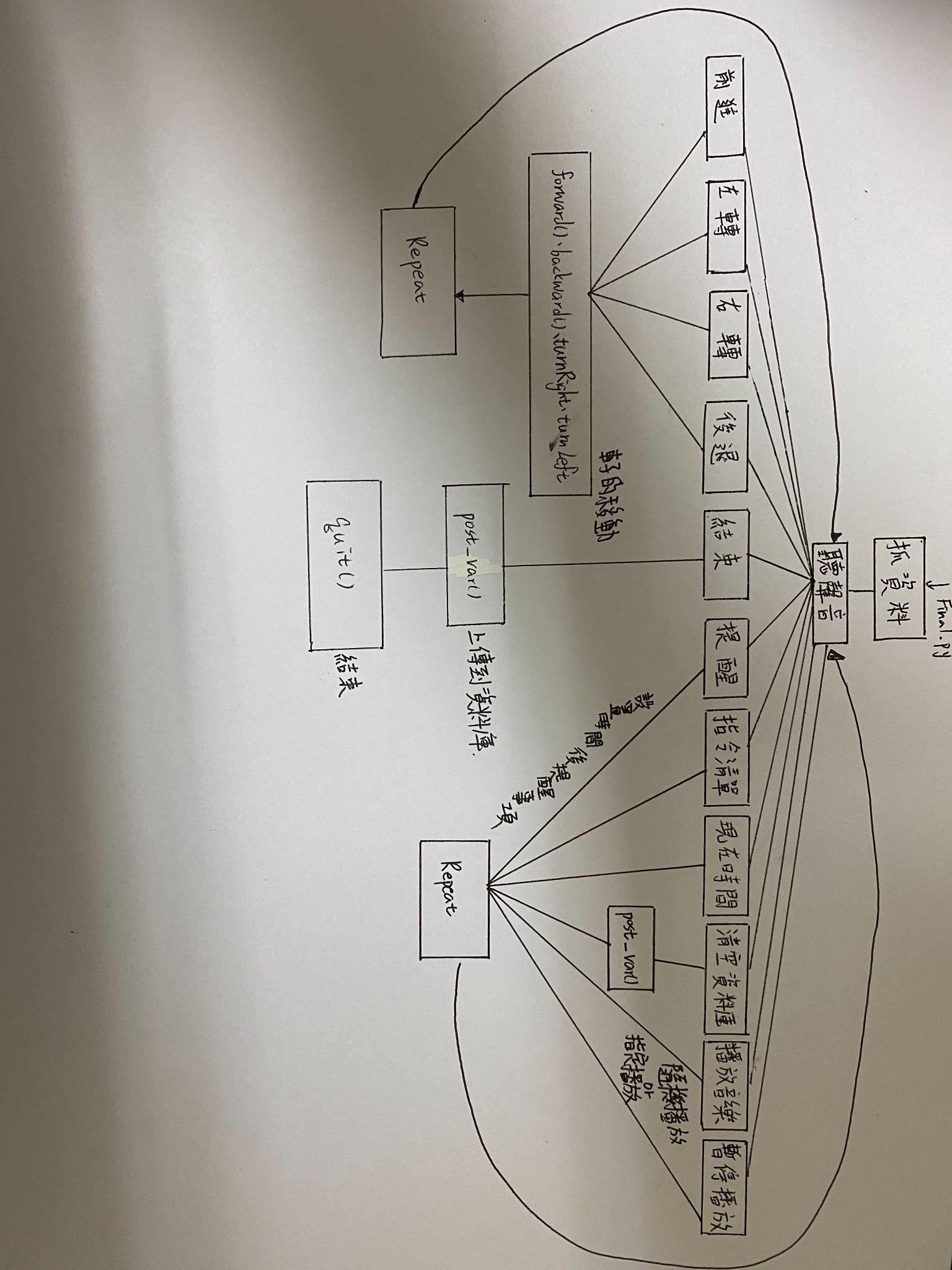
****

**硬體電路示意圖電子檔**

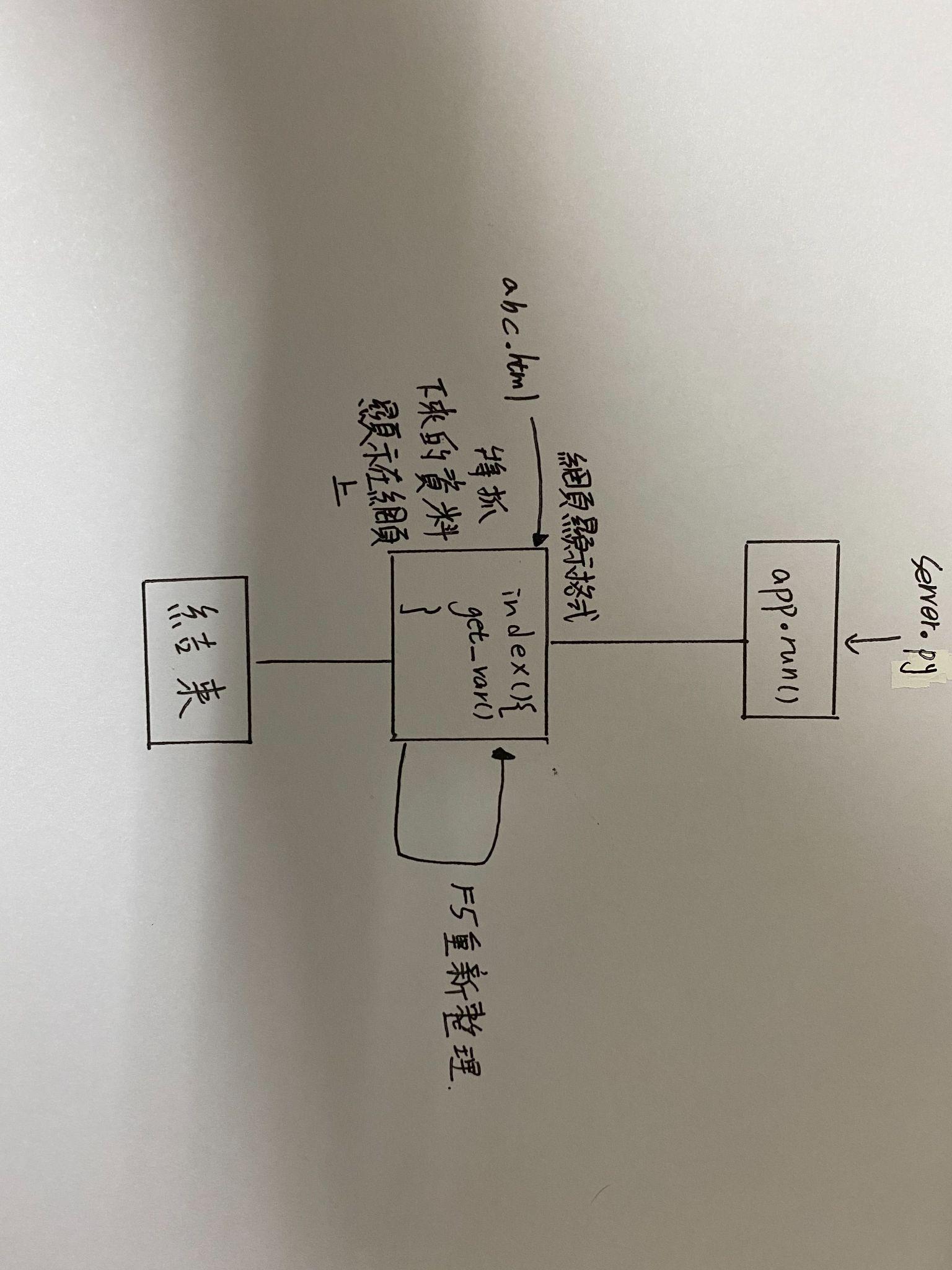
****

**硬體電路示意圖原稿**

**伍、軟體程式執行流程圖 :**

****

**軟體程式執行流程圖 - 主程式**

****

**軟體程式執行流程圖 - server端程式**

**陸、參考資料 :**

* 上課講義 lec.2 6 7 9 10
* <https://noob.tw/web-visualization-chartjs/>
* <https://www.liutingchun.com/post/raspberry-pi-video-switching>
* <https://andy6804tw.github.io/2021/01/21/time-interval/>
* <https://hackmd.io/@shaoeChen/SJpXnBTez?type=view>

**柒、開發最耗時的部分與原因 :**

最耗時的部分主要是在處理要怎麼樣讓語音助理能夠「在播放音樂的同時可以辨別說話 (下指令) 的聲音」。

一開始我們的想法是利用 create 一個 thread 去執行播放音樂的工作，但是我們後來發現這樣會使播放音樂的功能失效，我們在這個部分研究了一整天的時間，最後才發現如果利用 thread 放在 mainprocess 執行的話，就沒辦法執行其他指令。

後來在偶然的情況下，從網路上找到有人有做相關的操作去操控影片的播放，後來才知道原來不能使用 thread，而是要 create subprocess ，讓它去播放音樂，才能對它做操控。

**捌、展示影片 :**

<https://www.youtube.com/watch?v=IJhPtVjmK2A>