

# **Multiple Audio Control**

**Version 3**

**User Manual**

蔡淳佐 Chun-Tso Tsai (2022.03.23)

# 硬體設定

## 裝置規格

- 此系統可以在任何通用作業系統(Windows/OSX/Linux)下運行，包含 Raspberry Pi OS
- 若要使用小型裝置，可以選擇任何配備有WIFI及符合音響接孔的Raspberry Pi。（測試環境是使用Raspberry Pi Model 3B+，但規格上3A+, 3B, 3B+, 4B都可以運行，音響則支援3.5mm接孔。）

# 硬體設定

## 網路設定

- 確認所有裝置在同一個WIFI訊號環境下（同IP分享器）。
- Version 2更新：程式會自動建立在既有本地網路(LAN)下，無需手動設定IP及port。

# Raspberry PI設定

- 初始設定作業系統可以參考本文(<https://ithelp.ithome.com.tw/articles/10235135>)
- 若要方便日後的調整設定的話也可以設定使用 SSH 遠端操作 RPI，參考(<https://ithelp.ithome.com.tw/articles/10235452>)
- 安裝 python, pip，以及 pygame 套件，led控制則需要 adafruit-circuitpython-ws2801，可遵照以下指令 (可參考 [https://github.com/adafruit/Adafruit\\_CircuitPython\\_WS2801](https://github.com/adafruit/Adafruit_CircuitPython_WS2801))  

```
sudo apt install python3  
sudo apt install python3-pip  
sudo pip3 install pygame  
sudo pip3 install adafruit-circuitpython-ws2801
```
- 有時RPI OS上會有nodejs衝突的問題，先測試node --version以及npm --version是否有回應。  
若回報錯誤再用以下指令安裝  

```
sudo apt install node  
sudo apt install npm
```

# RPI開機執行程式

- 將資料夾 "client" 放入裝置中的根目錄 ~/ 中 (可使用FTP及FileZilla)。
- 若要每次開機直接執行的話，輸入 `sudo nano /etc/rc.local` 來修改，在`exit 0`以前加入以下指令  
`npm --prefix  
<path_to_client_folder> run start  
<device_name> &`
- 其中 `<path_to_client_folder>` 為client資料夾的位址；`<device_name>` 為任何自訂義裝置名稱
- 下次 RPI 開機時即可自動執行用戶端程式。

```
GNU nano 3.2 /etc/rc.local

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi

npm --prefix /home/pi/client/ run start myPI &

exit 0
```

^G Get Help   ^O Write Out   ^W Where Is   ^K Cut Text   ^J Justify  
^X Exit   ^R Read File   ^\ Replace   ^U Uncut Text   ^T To Spell

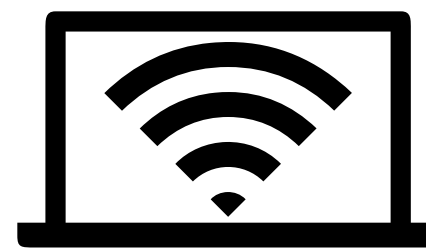
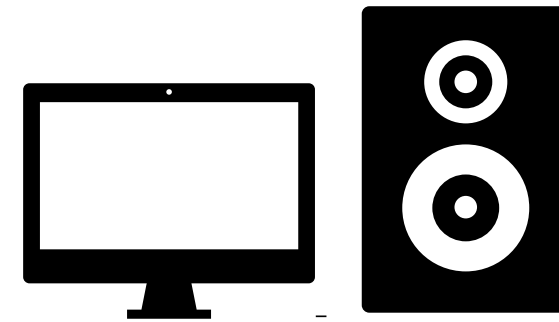
# 開始使用

- 在設定完網路及環境後，可以準備執行。
- 準備一台電腦當作中央控制台，匯入 `host` 資料夾
- 其餘裝置各自連接一台揚聲器，並且匯入 `client` 資料夾
- 作為控制台的電腦可以同時連接揚聲器，並且同時匯入 `client` 資料夾。  
(同一台電腦的 `host` 與 `client` 的 `port` 必須不同，若全部讓程式自動指定則不用在意)

# 裝置數量

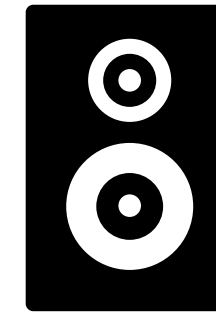
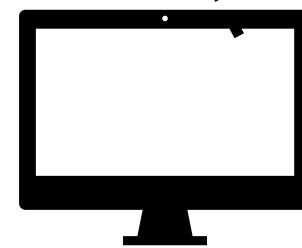
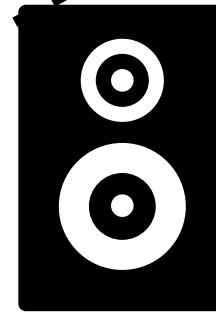
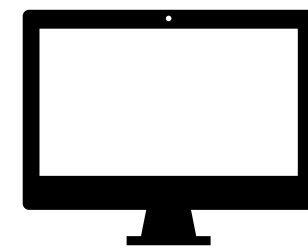
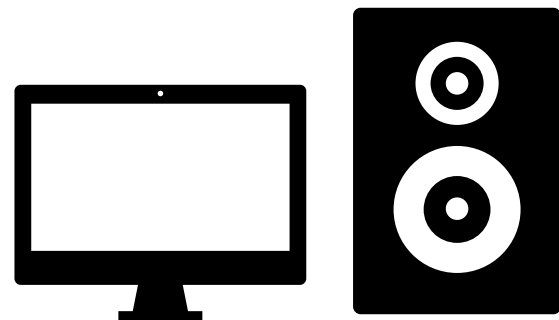
假設需要8台音響同時運作

中控台, 檔案={host + client}

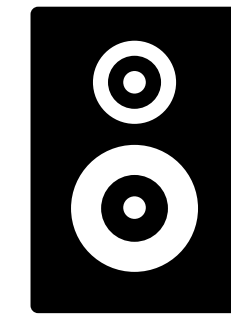


WIFI 分享器

控制器, 檔案={client}



.....



控制器, 檔案={client}

控制器, 檔案={client}

控制器, 檔案={client}

7台

# LED 連接

## GPIO連接口

- 首先在 /boot/config.txt 中加入以下這行來使用 spi  
dtoverlay=spi1-3cs
- 電源及接地分別連接到 5V Power (pin 2,4)
- 第一條光條的 CKI/CI 連到 GPIO11 (pin 23)  
SKI/DI 連到 GPIO10 (pin 19)
- 第二條光條的CKI/CI連到GPIO21 (pin 40)  
SKI/DI 連到 GPIO20 (pin 38)

3v3 Power	1			2	5v Power
GPIO 2 (I2C1 SDA)	3			4	5v Power
GPIO 3 (I2C1 SCL)	5			6	Ground
GPIO 4 (GPCLK0)	7			8	GPIO 14 (UART TX)
Ground	9			10	GPIO 15 (UART RX)
GPIO 17 (SPI1 CE1)	11			12	GPIO 18 (SPI1 CE0)
GPIO 27	13			14	Ground
GPIO 22	15			16	GPIO 23
3v3 Power	17			18	GPIO 24
GPIO 10 (SPI0 MOSI)	19			20	Ground
GPIO 9 (SPI0 MISO)	21			22	GPIO 25
GPIO 11 (SPI0 SCLK)	23			24	GPIO 8 (SPI0 CE0)
Ground	25			26	GPIO 7 (SPI0 CE1)
GPIO 0 (EEPROM SDA)	27			28	GPIO 1 (EEPROM SCL)
GPIO 5	29			30	Ground
GPIO 6	31			32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33			34	Ground
GPIO 19 (SPI1 MISO)	35			36	GPIO 16 (SPI1 CE2)
GPIO 26	37			38	GPIO 20 (SPI1 MOSI)
Ground	39			40	GPIO 21 (SPI1 SCLK)



# 執行步驟

- 同樣所有裝置必須安裝 npm, node, python (with pygame)。
- 使用 Terminal 進入指定資料夾 host 或是 client，在該目錄下輸入指令 `npm start <device_name>`，跳出提示訊息則代表成功啟動。

```
? ~/Documents/tony/projects/Multiple_audio_control/host ? ?main?origin/master ?? npm start myHost  
  
> mac_host@0.0.0 start  
> node ./bin/www.js "myHost"  
  
Service listening on 192.168.1.100:8080 as name "myHost"  
█
```

# 啟動指令

- `npm start <device_name>` 可以指定裝置名稱。
- 所有可定義參數為：`-i`，`-p`，`-n` 分別為IP address, port, name。例如：  
`npm start -i 192.168.1.100 -p 8084 -n`
- 三個參數順序可隨意調換，注意若手動指定參數則無法自動搜尋裝置。

# 網頁操作介面

## Audio control panel

- 開啟瀏覽器，網址輸入 host 啟動時顯示的 IP 及 port。
- 即可開啟控制面板

- Please upload the required audio files below.  
 未選擇任何檔案
- Use here to delete the audio files on the server.
- Click here to synchronize the remote devices, this contains the address of Host and the audio files.

Status of Remote Devices

<input type="text"/>	<input type="button" value="Update"/>
<input type="button" value="Check"/>	
<input type="button" value="Auto Scan"/>	
local@192.168.1.100:8085	
pi@192.168.1.140:8085	

## Execute the code

Upload the time stamp file  未選擇任何檔案

Choose the start time (if not specified, start from the beginning)

# 操作介面

## 連線狀態

- 顯示格式為“名稱@IP位址:port”
- 按下 Auto Scan 可以自動搜尋所有使用預設位址的本地裝置（點下後頁面會自動跳轉到搜尋結果的json格式，直接回到上一頁即可）
- 進行任何操作前，請確保裝置的連線狀態（全綠色再執行其他步驟）。隨時點擊“check”鈕可以更新狀態。
- 灰色：未知連線狀態  
紅色：無法連線到此裝置  
綠色：連接成功

Status of Remote Devices

Update

Check

Auto Scan

local@192.168.1.100:8085

pi0@192.168.1.140:8085

Status of Remote Devices

update

check

local@192.168.1.100:3000

pi@192.168.1.140:3333

# 操作介面

## 手動更新裝置列表

- 手動裝置更新方法：輸入一個新的裝置敘述

1. 若沒有已存在的名稱或IP

-> 按照輸入新增一個裝置

2. 若已存在名稱但IP或port不同

-> 更改該名稱裝置的 IP/port

3. 若存在同IP但名稱或port不同

-> 更改該IP裝置的名稱/port

4. 若存在完全相同的敘述

-> 刪除已存在的裝置

- 若更新完沒有及時刷新，可再按一次 check

### Status of Remote Devices

dev1@192.168.1.111:3043

update

check

local@192.168.1.100:3000

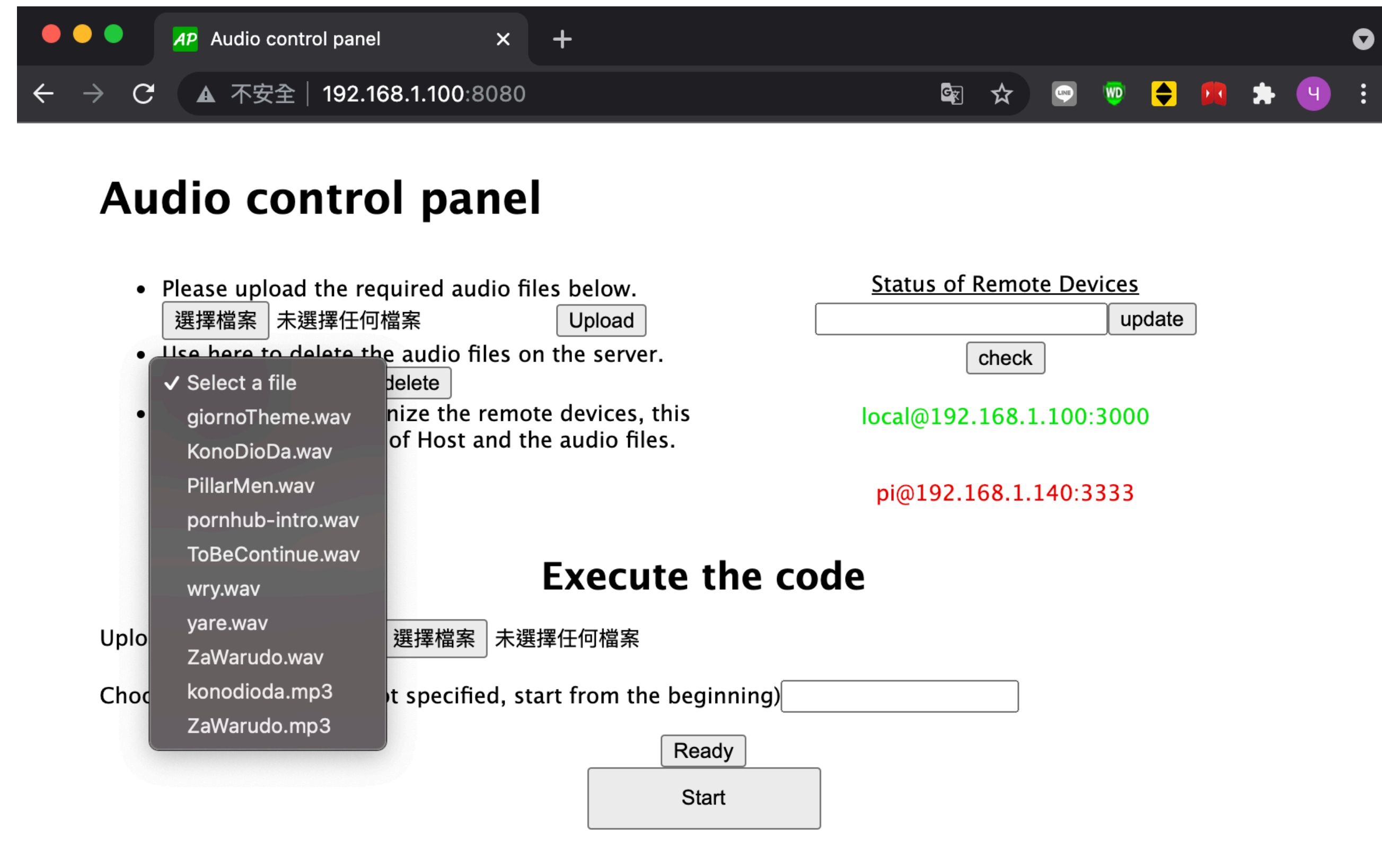
pi@192.168.1.140:3333



# 操作介面

## 更新檔案

- 第一欄位可以上傳檔案(支援mp3, wav)，選擇檔案並點擊 Upload 即可。
- 在刪除檔案的欄位可以看到現有已上傳的檔案，如果需要刪除，選擇並點擊 delete 即可。



# 操作介面

## 檔案同步

- 每次要播放前，請先點擊Synchronize
- 此功能會同步主機位址及所有包含的音檔，如果檔案較多或較大可能需要稍等一下。

### Audio control panel

- Please upload the required audio files below.  
 未選擇任何檔案
- Use here to delete the audio files on the server.
- Click here to synchronize the remote devices, this contains the address of Host and the audio files.

#### Status of Remote Devices

local@192.168.1.100:3000

pi@192.168.1.140:3000

### Execute the code

Upload the time stamp file  未選擇任何檔案

Choose the start time (if not specified, start from the beginning)

Synchronization complete.

# 操作介面

## 時間戳記檔

- 可以參考附上的 test.tms，可標記功能為

1. 新增播放，開頭為 n 或 new，格式為：  
n[ew] 時間 裝置名稱 音檔(含副檔名) 音量

音量可加可不加，不加就不調整

2. 調整音量，開頭為 v 或 volume，格式為：  
v[olume] 時間 裝置名稱 音量

音量為百分比記號

- 空行及縮排沒問題，#字開頭的行代表是註記。
- 注意請按照時間順序標記事件

```
25 new 0:00 local ZaWarudo.wav
26 new 0:02 local konodioda.mp3
27 new 0:03 pi PillarMen.wav
28 new 0:06 local wry.wav
29
30 v 0:04 pi 40
31
32 new 0:07 local pornhub-intro.wav
33
34 v 0:07.5 local 40
35 new 0:09 pi ToBeContinue.wav
36
37 new 0:10 local ToBeContinue.wav 100
38 v 0:12 local 50
39 v 0:13 local 10
```



# 操作介面

## 時間戳記檔

- 可以參考附上的 test.tms

3. LED 控制，開頭為 l 或 light，格式為

l[ight] 時間 裝置名稱 光條編號 LED編號 紅 綠 藍

範例：l 0:11 myDevice 0 1 255 255 255

在11秒的時候對驅動編號 0 的光條上的編號 1 的LED，顏色為  
(r,g,b)=(255,255,255) 白光。

- 一台RPI連接兩台光條
- 編號一率從0開始
- RGB 數值為 0-255 的整數

# 操作介面

## 上傳時間戳記

- 上傳時間戳記檔到第一欄位
- 第二欄位可以標記開始時間，可留空白，則從 0:00 開始。
- 所有時間記號請用 {分:秒} 格式，可以有小數點。
- 輸入完畢後按下 Ready

### Audio control panel

- Please upload the required audio files below.

選擇檔案

未選擇任何檔案

Upload

- Use here to delete the audio files on the server.

Select a file



delete

- Click here to synchronize the remote devices, this contains the address of Host and the audio files.

synchronize

#### Status of Remote Devices

update

check

local@192.168.1.100:3000

pi@192.168.1.140:3000

### Execute the code

Upload the time stamp file

選擇檔案

test.tms

Choose the start time (if not specified, start from the beginning)

0:02

Ready

Start

# 操作介面

## 準備播放

- 按下 Ready 後如果再點選 Check，可以看到是否準備完成。
- 準備完成的裝置會標識為 == Ready

### Audio control panel

- Please upload the required audio files below.  
 未選擇任何檔案
- Use here to delete the audio files on the server.
- Click here to synchronize the remote devices, this contains the address of Host and the audio files.

#### Status of Remote Devices

local@192.168.1.100:3000 == Ready

pi@192.168.1.140:3000 == Ready

### Execute the code

Upload the time stamp file  未選擇任何檔案

Choose the start time (if not specified, start from the beginning)

# 操作介面

## 開始播放

- 在標示 Ready 之後，按下 Start 即可開始播放。

# Matlab 介面

# Matlab介面

## 加入路徑

- 在 Matlab 中加入路徑本專案的 matlab 資料夾。  
Matlab 指令為 `addpath(<path_to_matlab_folder>)`

### Command Window

```
>> addpath('Multiple_audio_control/matlab/')  
fx >> |
```

# Matlab介面

## 建立連線

- 架設連線，使用 `serverSetup()` 函數。  
註：如果 `matlab/` 資料夾與 `host/` 資料夾沒有在同一個母資料夾下的話，則必須將 `host/` 的位址給入函數。
- 若成功則會回傳包含所有連線裝置資訊的 `table`。（僅支援自動IP位址）

```
>> devices = serverSetup()  
Server found on 192.168.1.100:8080. 2 connected devices found.  
devices =
```

2×3 [table](#)

DeviceName	Address	Port
"local"	"192.168.1.100"	8085
"pi0"	"192.168.1.140"	8085

```
>> devices = serverSetup('Multiple_audio_control/host')  
Server found on 192.168.1.100:8080. 2 connected devices found.  
devices =
```

2×3 [table](#)

DeviceName	Address	Port
"local"	"192.168.1.100"	8085
"pi0"	"192.168.1.140"	8085

# Matlab 介面

## 即時播放聲音

- 可以利用 `remoteSound(y, device [, light_info, volume, Fs])` 來即時播放聲音。
- 參數：
  - `y`: 可以是 1) double-valued signal 如同呼叫 `Sound(y, Fs)` ; 2) 音檔路徑。
  - `device`: 可以是 1) 裝置名稱如同表格中所示 ; 2) index (integer 或 logical array) 來指示在表格中位置。
  - `light_info`: 由 `setLightConfig()` 產生的模板 (細節後述)
  - `volume`: 播放音量, 為 0-100 的整數, 預設為100為最大音量(意思為100%)
  - `Fs`: Sampling Frequency, 預設為 8192 Hz 如同 `Sound(y,Fs)`。



# Matlab 介面

## 設定燈光

- 利用 `light_info = setLightConfig(strip_idx, led_idx, red, green, blue)` 來產生檔案。
- 參數：
  - `strip_idx`: 0或1來指定光條。
  - `led_idx`: 光條上的 LED 編號
  - `red`: 紅光量值 0-255
  - `green`: 綠光量值 0-255
  - `blue`: 藍光量值 0-255

# Matlab 介面

## 即時播放聲音

- 範例

```
>> strip_index = 0;  
>> led_index = 0;  
>> red = 255;  
>> green = 255;  
>> blue = 255;  
>> light_info = setLightConfig(strip_index, led_index, red, green, blue);  
>> remoteSound(y, 'myDevice', light_info, [], Fs);  
>> remoteSound('./Multiple_audio_control/host/resources/ZaWarudo.wav', 'local')  
>> remoteSound('./Multiple_audio_control/host/resources/ZaWarudo.wav', logical([1,0]))
```

# Matlab 介面

## 透過預設腳本播放聲音

- 利用函數 `timeStampStart(time_stamp_path, audio_files_involved)`
  - `time_stamp_path`: 腳本路徑
  - `audio_files_involved`: 相關的音檔路徑（本次事件要執行的音檔）

# Matlab 介面

## 透過預設腳本播放聲音

- 範例

### Command Window

```
>> audio_files
```

```
audio_files =
```

```
6×1 string array
```

```
"~/Desktop/Multiple_audio_control/host/resources/ZaWarudo.wav"
```

```
"~/Desktop/Multiple_audio_control/host/resources/konodioda.mp3"
```

```
"~/Desktop/Multiple_audio_control/host/resources/PillarMen.wav"
```

```
"~/Desktop/Multiple_audio_control/host/resources/wry.wav"
```

```
"~/Desktop/Multiple_audio_control/host/resources/pornhub-intro.wav"
```

```
"~/Desktop/Multiple_audio_control/host/resources/ToBeContinue.wav"
```

```
>> timeStampStart('./Multiple_audio_control/test.tms', audio_files)
```

```
Processing audio files...
```

```
Start playing assigned audio events.
```

# Matlab 介面

## 清除音檔

- 為了每次的執行效率，音檔會暫存於 server 及 client 的資料夾中，每次實驗結束建議清除音檔才不會佔用記憶體空間。在 `serverSetup()` 已完成的狀態下呼叫 `clearAudioFiles()` 即可。

# Matlab 介面

## 執行速度

- 如果實驗對於裝置之間的合作需要很精確的時間，建議使用 time stamp 方法。
- 這不是網路問題，同一個LAN底下的連線速度誤差合理都會在 10 ms 以下。
- 主因是 Matlab 的網路介面執行速度較慢，所以即時呼叫可能最差會到半秒左右的延遲，如果需要多裝置精確執行的話還是建議使用 time stamp。
- 使用 Matlab global variable “servC.default\_wait\_time” 可以調整系統等待時間，可以設定 0-5000 的值（單位為ms）此數值越高在同步上會比較穩定，但會延遲開始播放。