



# Pomas

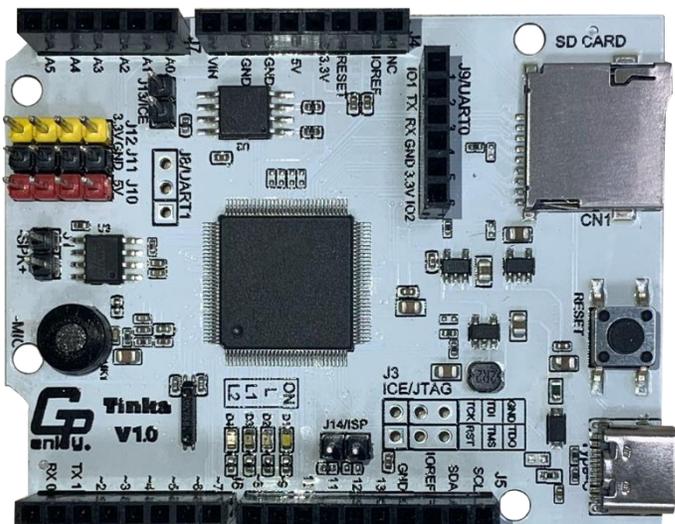
## MicroPython AIOT



名威智慧通訊

專案講師 林羿君 Lynn Lin

STRUCTURE  
架構



1

## 語音辨識

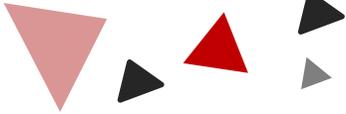
- 1 語音智慧管家
- 2 創建屬於自己的智慧管家
- 3 內建音效處理單元

2

## 簡易智慧家庭

- 1 智慧家庭Demo

# 0



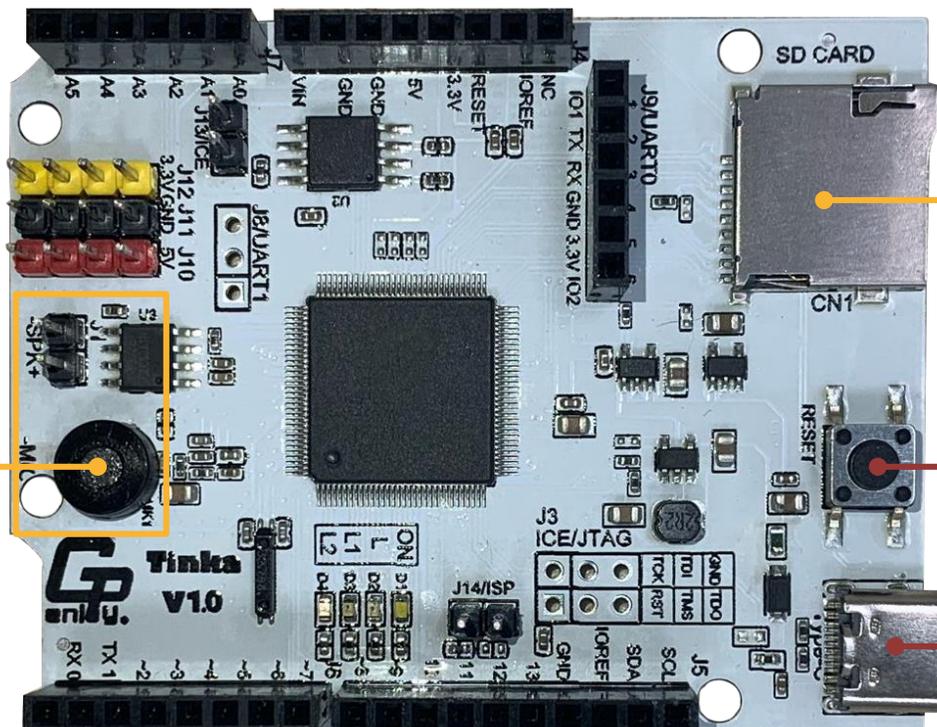
PART



# Genio Py.

硬體介紹

內建麥克風及揚聲器插口

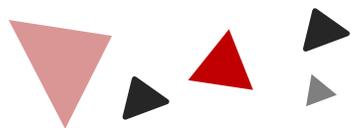


SD card

reset

USB-C 連接孔  
(power)

11



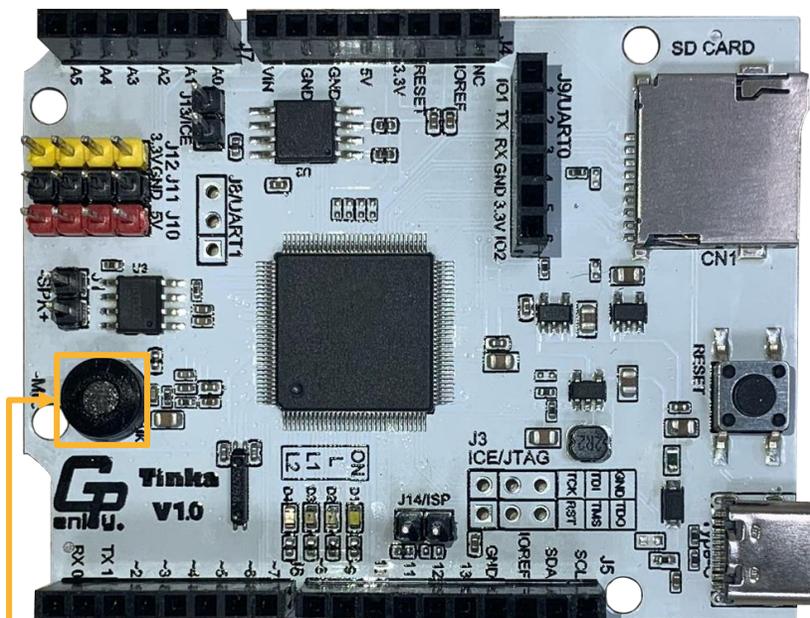
PART

語音辨識  
簡易智慧管家



# 原理

# 內建語音口令辨識



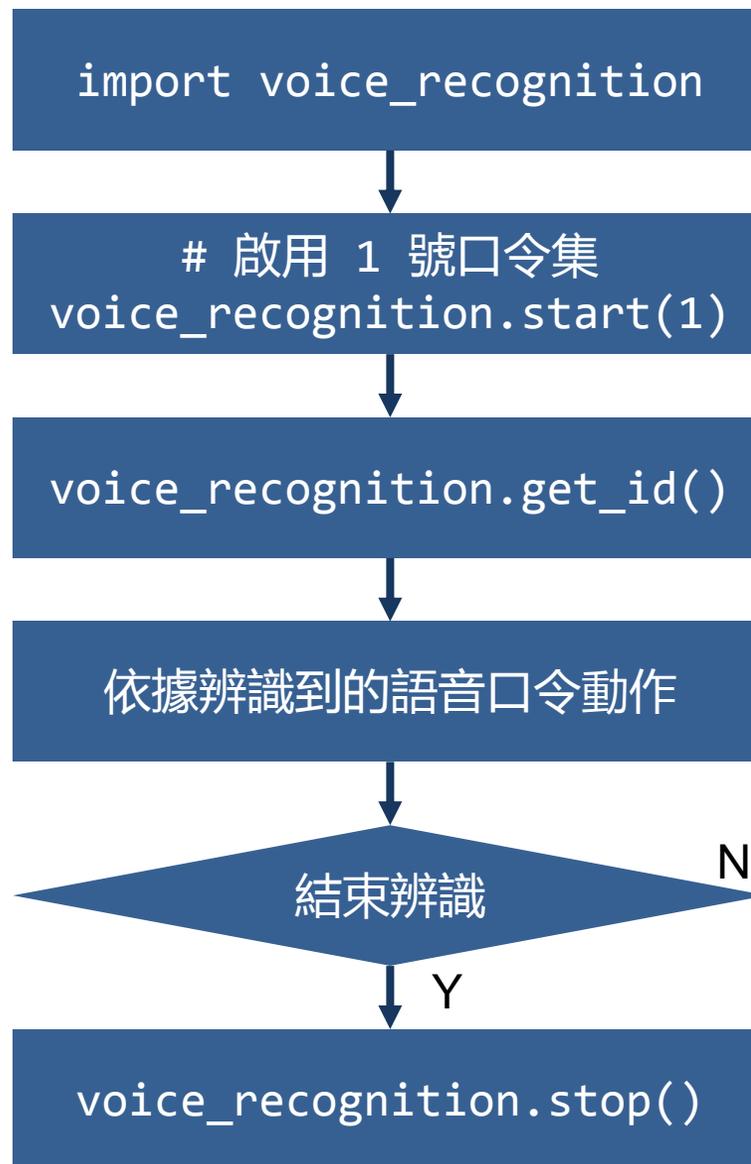
內建麥克風

編號id	口令集 1	編號id	口令集 1
1	小愛管家	11	停止音樂
2	打開電燈	12	現在溫濕度
3	關閉電燈	13	大門上鎖
4	打開風扇	14	大門解鎖
5	提高風速	15	凌威管家
6	降低風速	16	拍森管家
7	關閉風扇	17	小珍同學
8	播放音樂	18	志玲姐姐
9	增加音量	19	柔伊妹妹
10	降低音量	20	美玉阿姨

# 動手做

## 語音辨識

先念口令再執行



# 動手做

# 寫程式用語音控制LED

可參考sample code\_  
voice\_recognition\_led.py

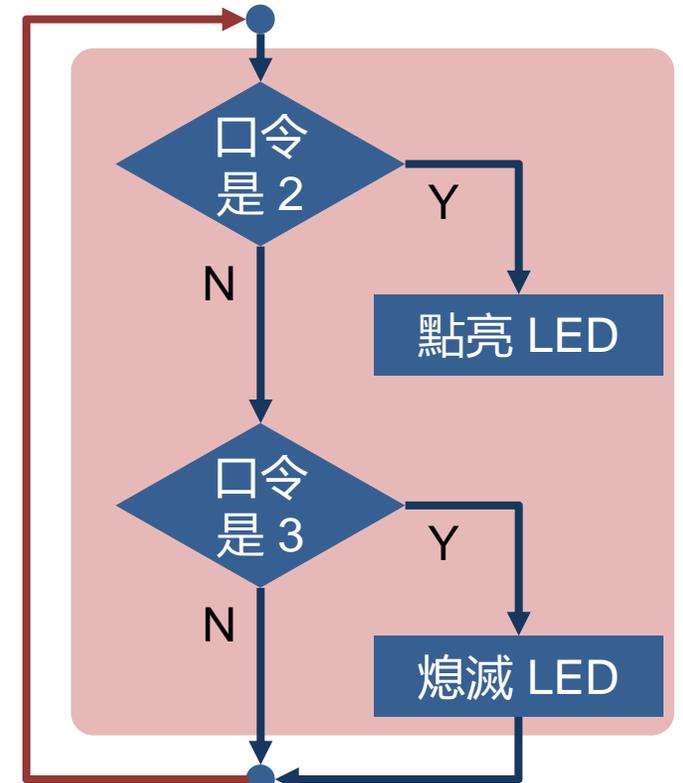
```
1 import voice_recognition
2 from gpb import LED,delay
3
4 led = LED(1)
5
6 voice_recognition.start(1)
7
8 while True:
9     cmd_id =voice_recognition.get_id()
10    if cmd_id == 1:
11        led.on()
12    elif cmd_id == 2:
13        led.off()
```

# 匯入語音辨識模組

# 啟用1號指令集

# 取得辨識結果  
# 指令:Hey Tinka

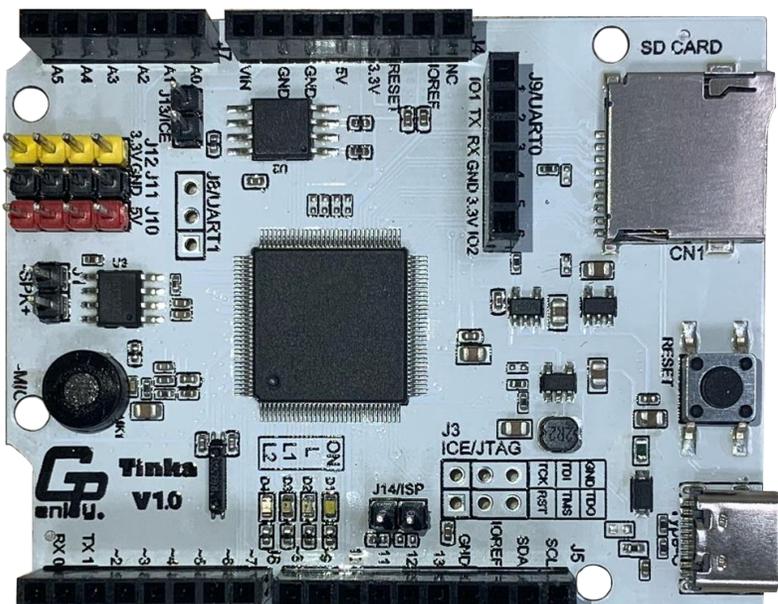
# 指令:OK Tinka



利用 elif 建立多個條件判斷的流程

# 原理

# 擴充語音口令辨識



編號id	指令集 1	編號id	指令集 1
1	Hey Tinka	9	播放音樂
2	OK Tinka	10	增加音量
3	打開電燈	11	降低音量
4	關閉電燈	12	停止音樂
5	打開風扇	13	開啟錄音
6	提高風速	14	播放錄音
7	降低風速	15	芝麻開門
8	關閉風扇	16	芝麻關門

語音建模說明書請參照PDF檔or教學資源下載區下載

# 動手做

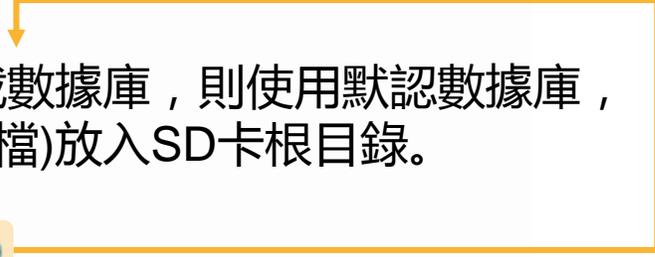


# 利用擴充語音控制LED

可參考sample code\_  
voice\_recognition\_led1.py

```
1 import voice_recognition # 匯入語音辨識模組
2 from gpb import LED, delay
3
4 led = LED(1)
5
6 voice_recognition.load_database('tinkaVR.bin')
7 delay(500)
8 voice_recognition.start(10) # 啟用10號指令集
9
10 while True:
11     cmd_id = voice_recognition.get_id() # 取得辨識結果
12     if cmd_id == 3: # 指令: 打開電燈
13         led.on()
14     elif cmd_id == 4: # 指令: 關閉電燈
15         led.off()
```

對於用戶數據庫，如果未加載數據庫，則使用默認數據庫，請把要擴充的語音指令集(bin檔)放入SD卡根目錄。



# 動手做

# 利用擴充語音控制Servo轉動

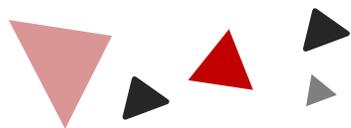
```
1 import voice_recognition # 匯入語音辨識模組
2 from gpb import Servo, delay # 匯入Servo類別、delay函式
3
4 # 建立伺服馬達物件
5 # 將馬達角度預設為0度
6
7 voice_recognition.load_database('tinkaVR.bin')
8 delay(500)
9 voice_recognition.start(10) # 啟用10號指令集
10
11 while True:
12     cmd_id = voice_recognition.get_id() # 取得辨識結果
13     if 聽到指令: 芝麻開門 # 指令: 芝麻開門
14         轉動伺服馬達 # 亮燈
15     elif 聽到指令: 芝麻關門 # 指令: 芝麻關門
16         轉動伺服馬達到原位 # 熄燈
```

# 解答

## 利用擴充語音控制Servo轉動

```
1 import voice_recognition # 匯入語音辨識模組
2 from gpb import Servo, delay # 匯入Servo類別、delay函式
3
4 door=Servo(7) # 使用PWM7腳位控制Servo1
5 door.angle(0)
6
7 voice_recognition.load_database('tinkaVR.bin')
8 delay(500)
9 voice_recognition.start(10) # 啟用10號指令集
10
11 while True:
12     cmd_id = voice_recognition.get_id() # 取得辨識結果
13     if cmd_id == 15: # 指令:芝麻開門
14         door.angle(-90) # 亮燈
15     elif cmd_id == 16: # 指令:芝麻關門
16         door.angle(0) # 熄燈
```

# 12



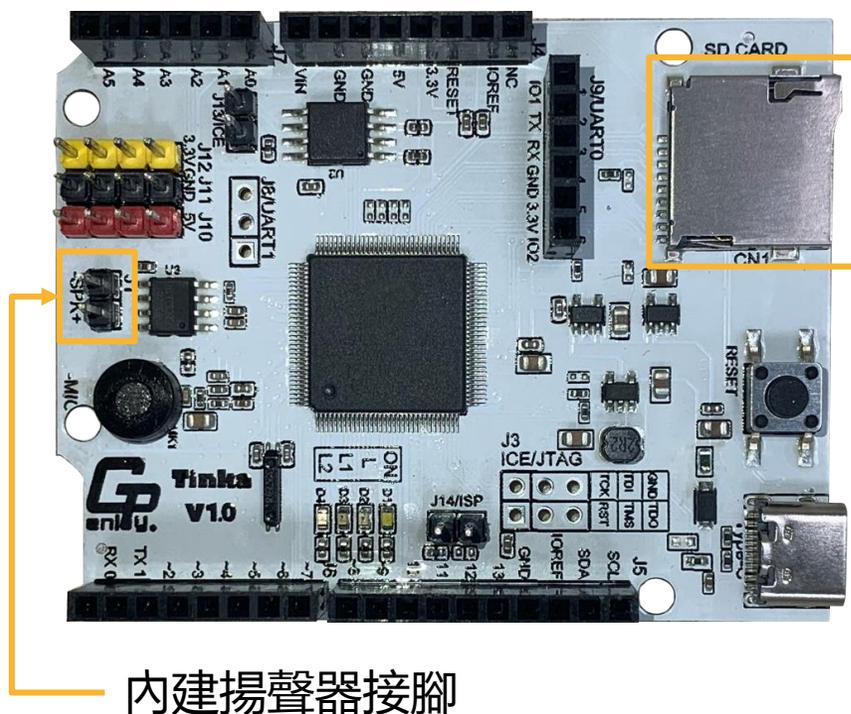
## PART

### 播放聲音

播放 SD 卡上的 mp3/wav 檔

# 原理

## 內建音效處理晶片



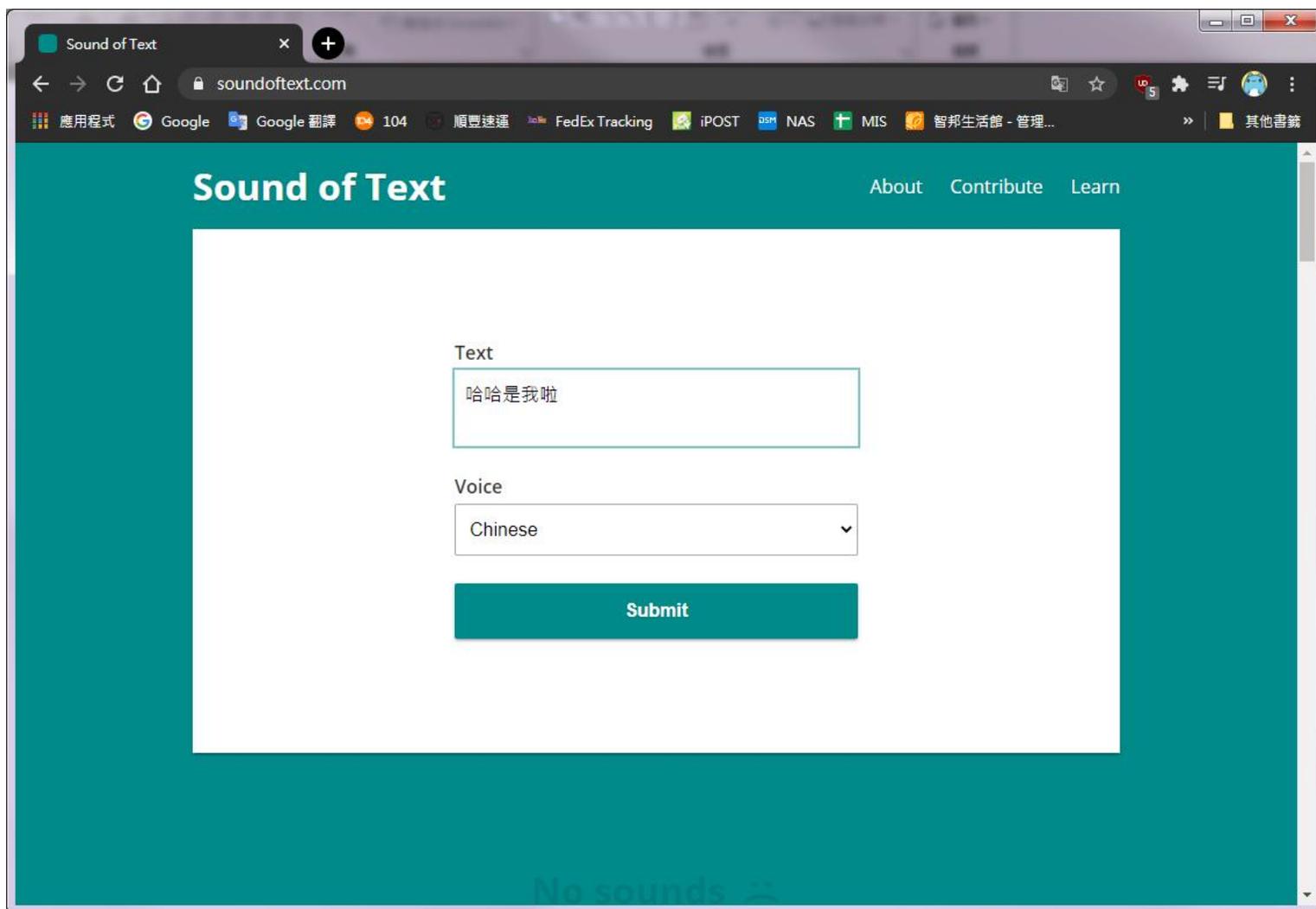
MicroSD 卡會變成隨身碟

可播放 SD 卡上的 mp3 / wav 檔

內建揚聲器接腳

# 動手做

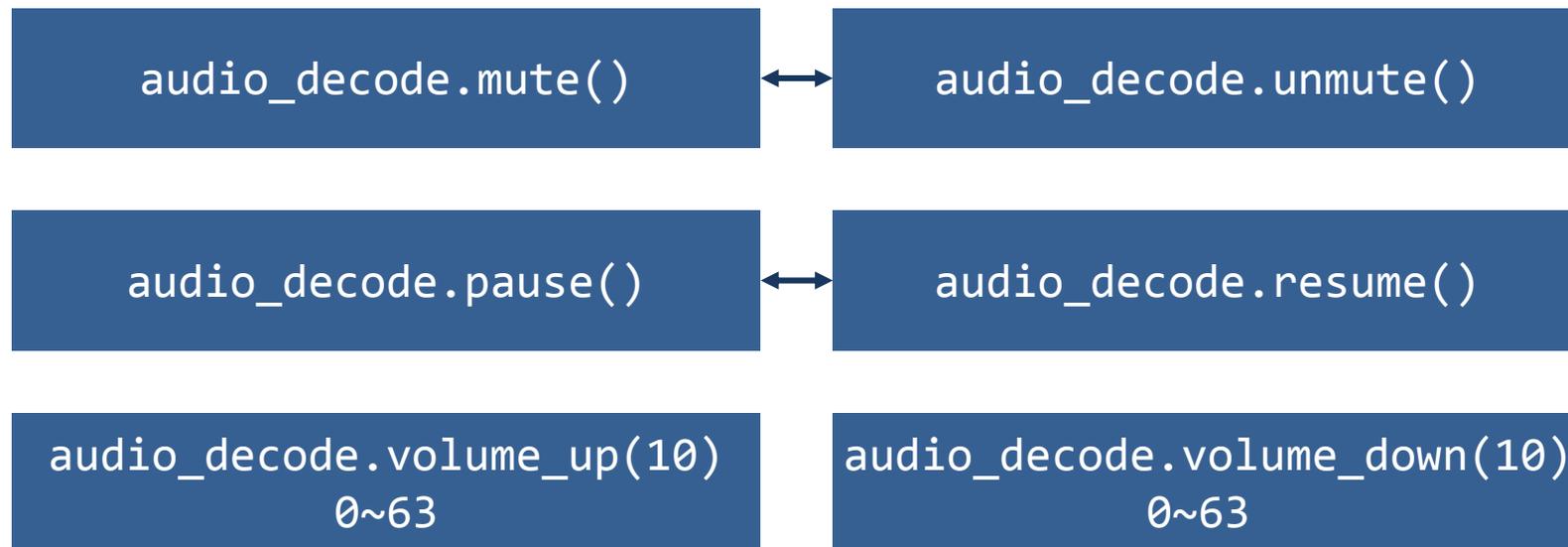
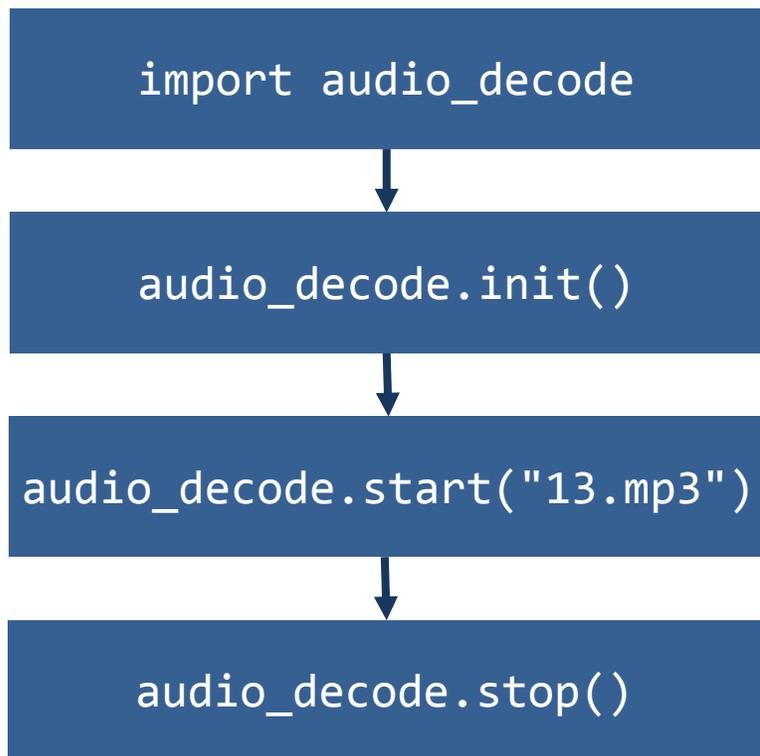
# 準備聲音



<https://soundoftext.com/>

試試到這裡打字下載  
Google 小姐唸出來的 mp3 檔

# 音效播放流程



```
>>> import audio_decode
>>> audio_decode.init()
>>> audio_decode.start("tank.mp3")
>>> audio_decode.volume_down(10)
>>> audio_decode.volume_up(10)
>>> audio_decode.pause()
>>> audio_decode.resume()
>>> audio_decode.stop()
```

# 動手做

## 用口令播放音樂

可參考sample code\_  
audio\_decode.py

```
1 import voice_recognition
2 import audio_decode
3 from gpb import delay
4
5 voice_recognition.load_database('tinkaVR.bin')# 載入擴充數據庫
6 delay(500)
7 voice_recognition.start(10) # 啟用10號口令集
8 audio_decode.init() # 啟用音效模組
9
10 while True:
11     cmd_id=voice_recognition.get_id() # 取得辨識ID
12     if cmd_id == 9: # 口令:播放音樂
13         print("start!")
14         audio_decode.start('IU.mp3')
15     elif cmd_id == 12: # 口令:停止音樂
16         audio_decode.stop() # 關閉音效模組
17         print("stop!")
```

播放音樂時需要執行其他口令，可能需要把喇叭遮住才能辨識



THANKS

Q&A 時間