

臺北市立大安高級工業職業學校

電子科

專題報告

Line 訊息控制器

Line message controller

學生 組長：姓名 謝宗祐

組員：姓名 林瑋翔

組員：姓名 邱康文

組員：姓名 曾宏義

指導老師：楊仁元

中華民國 108 年 ○ 月

臺北市立大安高工 電子科

專題報告：

Line 訊息控制器

學生：0604111 林瑋翔 _____ (簽名)

0604113 邱康文 _____ (簽名)

0604123 曾宏義 _____ (簽名)

0604138 謝宗祐 _____ (簽名)

期末專題報告合格，特予證明

指導老師：楊仁元 _____

科主任：薛元陽 _____

中華民國 108 年 ○ 月

中文摘要

有感於導師和學生的交流有所限制，我們希望能嘗試一些新的交流方式來連接學生與導師，幫助老師更方便管理班級。而近年來的無線網路普及率已經相當高，因此本次專題的研究將會以 Line 和訊息控制的結合為主。

專題的 Line 訊息控制器會使用樹莓派，程式語言 python，接收以及發送訊息的 linebot 等等，經過討論以及資料的查找之後，我們決定使用手機 app 來操控它，使大家都可以輕易地使用手機操作。

關鍵字：樹莓派，python，linebot，訊息控制

英文摘要

Because of that connections between students and teacher is limited, so we hope to attempt some new kinds of method to connect them and help teacher to manage the class more convenient. For the past few years , wifi is appropriate, so our project studies will mainly focus on how to connect line and the message control.

Our project studies will include Raspberrypi ,the programming language python , and linebot to receive and transmit the message, after discussing and imformation inquiry, We decided to use App to control it , though using mobile phone is simple to everyone.

Keywords: Raspberry Pi python linebot

目錄

中文摘要.....	II
英文摘要.....	III
目錄.....	IV
表目錄.....	VI
圖目錄.....	VII
壹、 前言（概論／緒論）.....	1
一、 專題製作背景及目的.....	1
二、 預期成果.....	1
貳、 理論探討.....	2
一、 硬體.....	2
（一） Raspberry Pi.....	2
（二） 音訊放大器 Lm386.....	3
二、 軟體.....	4
（一） Raspberry Pi 作業系統.....	4
（二） Python 整合開發環境.....	4
（三） Ngrok 原理介紹及使用.....	5
參、 專題準備.....	6
一、 專題架構圖.....	6
二、 專題流程圖.....	7
三、 專題甘特圖.....	8
肆、 專題成果.....	9
一、 歷程記錄.....	9
二、 成果.....	9
（一） Ngrok.....	9
（二） Line.....	10
（三） 專題實體.....	11
伍、 結論與建議.....	12

一、	結論.....	12
	(一) 可以完成一對一的訊息控制.....	12
	(二) 可以完成定址訊息控制.....	12
	(三) 可以預先設定時間，在正確的時間將傳送訊息.....	12
	(四) 可以在一段時間之內重複訊息.....	12
二、	建議.....	12
	(一) 因為我們是使用 LINEBOT 來完成回傳的功能，因此不能超出 LINE 自己本身的框架。.....	12
	(二) 因為是使用樹莓派進行開發，每次使用都必須加裝滑鼠,螢幕, 鍵盤 之類的配件對於經常使用其實相當不便。.....	13
	(三) 樹梅派雖說基本等於一台個人電腦,但是處理速度無法與一般電腦相比，執行速度偏慢。.....	13
	(四) 因為同一群組只能允許存在一隻機器人，因此 LINE 無法進行對多的傳輸，可以考慮其他媒介來完成此目標。.....	13
	(五) Ngrok 因為設定，每次重新開機都必須新生成網址。.....	13
	(六) 樹梅派使用配件較多，建議之後開發可以使用遠端操控的方式。 13	
	(七) 將 Ngrok 改成其他網內穿透工具，例如 natapp.....	13
	參考文獻.....	14
	附錄.....	15

表目錄

錯誤! 找不到圖表目錄。

圖目錄

圖 1 樹莓派.....	2
圖 2 LM386.....	4
圖 3 NGROK 運行圖	5
圖 4 架構圖.....	6
圖 5 流程圖.....	7
圖 6 專題甘特圖	8
圖 7 NGROK 執行圖.....	9
圖 8 執行成功.....	10
圖 9 LINE SERVER.....	10
圖 10 LINE 執行圖	10
圖 11 外盒設計圖	11
圖 12 專題完成圖	11
圖 13 內視圖	12

壹、 前言（概論／緒論）

一、 專題製作背景及目的

在現代中，台灣的科技正以極快的速度成長，為了方便導師管理自己的班級，我們想到了一些實用的功能，利用一般人就常用的手機 app 上傳送需要的訊息給訊息盒，讓訊息盒能夠在指定的時間內，傳送訊息內容，但下課時間同學們常常會因為聊天的聲音太大聲以致於沒有聽到廣播內容，這時我們就能設定廣播的次數，以改善這類的問題，能大大的提升廣播系統的功用。剛好最近學校有 Linux 系統跟樹莓派的教學，希望能藉由專題和課程的合併來達到更廣域的學習。

二、 預期成果

本次專題預計完成一個無線網路定時廣播盒，預先將訊息盒放置在導師的教室並連上網路。希望廣播時就使用手機 app 將廣播內容傳送到訊息盒，訊息盒便會向周圍廣播出該訊息。具體如下：

1. 可以完成一對一的訊息控制
2. 可以完成定址訊息控制
3. 可以預先設定時間，在正確的時間將傳送訊息
4. 可以在一段時間之內重複訊息

製作過程中，除了學習程式語言外，更可以學到在失敗中尋找方法，同學之間的配合和鼓勵都使我們獲益良多。

CPU :	1.2 GHz 64-bit quad-core ARM Cortex-A53
GPU :	Dual Core VideoCore IV® Multimedia Co-Processor; Open GL ES 2.0; hardware-accelerated OpenVG; 1080p60 H.264 high-profile decode
記憶體 :	1GB LPDDR2 (和 GPU 共享)
視訊輸出 :	Composite RCA; HDMI
音訊輸出 :	3.5 mm jack; HDMI(1.3 & 1.4)
儲存 :	microSD
USB :	USB 2.0 x 4
Ethernet :	10/100 RJ45
Wireless :	802.11n
Bluetooth :	Bluetooth 4.1; Bluetooth Low Energy(BLE)
GPIO :	40-pin 2.54 mm (100 mil) expansion header: 2×20 strip
工作電流 :	800 mA
尺寸 :	85mm x 56mm x 17mm
重量 :	42g

(二) 音訊放大器 Lm386

LM386 的增益調整是利用第 1 腳與第 8 腳來控制的，當腳 1 與腳 8 開路時，內部電路中的 $1.35k\Omega$ 設置電壓增益為 20(26dB)；若在腳 1 與腳 8 間接上一個 $10\mu f$ 電容器，將 $1.35k\Omega$ 旁路，則電壓增益會提升為 200(46dB)；如想讓它的增益為 20 至 200 中的任一值；只需在腳 1 與腳 8 間增加一個電阻與電容的串聯電路

就可以了。

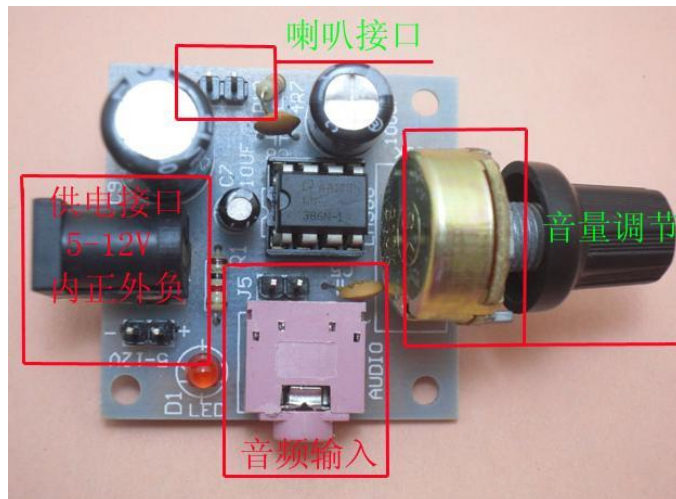


圖 2 1m386

二、軟體

(一) Raspberry Pi 作業系統

是一個開放源程式碼的硬體專案平台,該平台包括一塊具備簡單 I/O 功能的電路板以及一大堆的 Linux 軟體。

Raspberry Pi 可以用來開發交互產品,比如它可以讀取大量的開關和感測器信號,並且可以控制電燈、電機和其他各式各樣的物理設備。

Raspberry Pi 也可以開發出與 PC 一樣的周邊裝置,也可以運行在 Linux PC 上的軟體進行通信。

Raspberry Pi 的硬體電路板可以自行焊接組裝,也可以購買已經組裝好的模組,而程式開發環境的軟體則可以從網上免費下載與使用。重點是它的價格非常的便宜,已今天的市價約要 1200 元台幣就可以買到。

Raspberry Pi 作業系統可以在 Resberry pi 官網安裝。

(二) Python 整合開發環境

Python 是物件導向程式級高階程式語言,也是直譯式程式語言。Python 以強調對程式語言的語句易讀、易懂、易學(簡潔和清晰的語法特點)及加快程式開發的時效,方便使用,可以完成各種難度的應用,並可在大多數的系統中運行,以減少開發及維護成本的觀念進行發展。

Python 同時支援 modules 和 packages，另外 Python 為跨平台程式語言也支援 unicode 字元。功能強大而完善的通用型語言，可以用於很多種軟體開發動態程式，使得 Python 非常有吸引力，發展至今已有十多年的歷史，成熟且穩定。

通常，有許多程式設計者接觸 Python 後都會被她的優點所吸引，並提供了許多自行開發的 library(函式庫)以提供其他 Python 程式設計者下載使用，所以具有豐富和龐大的類別函式庫，可以負荷支持平常大部份的應用，使得現今 Python 發展迅速及使用廣泛，相關資源也相當豐富。

(三) Ngrok 原理介紹及使用

Ngrok 是一個連接樹莓派與 Line 的媒介，有了他就可以讓樹莓派和 Line 進行連接。

首先註冊並下載 ngrok 到自己的電腦上，Ngrok 生成一個網址並將此網址設定到 line 的 webhook，當使用者跟 Linebot 進行溝通時，messengerApi 會觸發設定好的 webhook，而此 webhook 就是 Ngrok 生成的網址，會再把他轉到自己的電腦上。

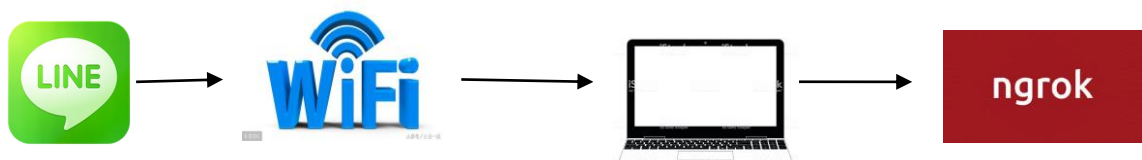


圖 3 Ngrok 運行圖

參、 專題準備

一、 專題架構圖

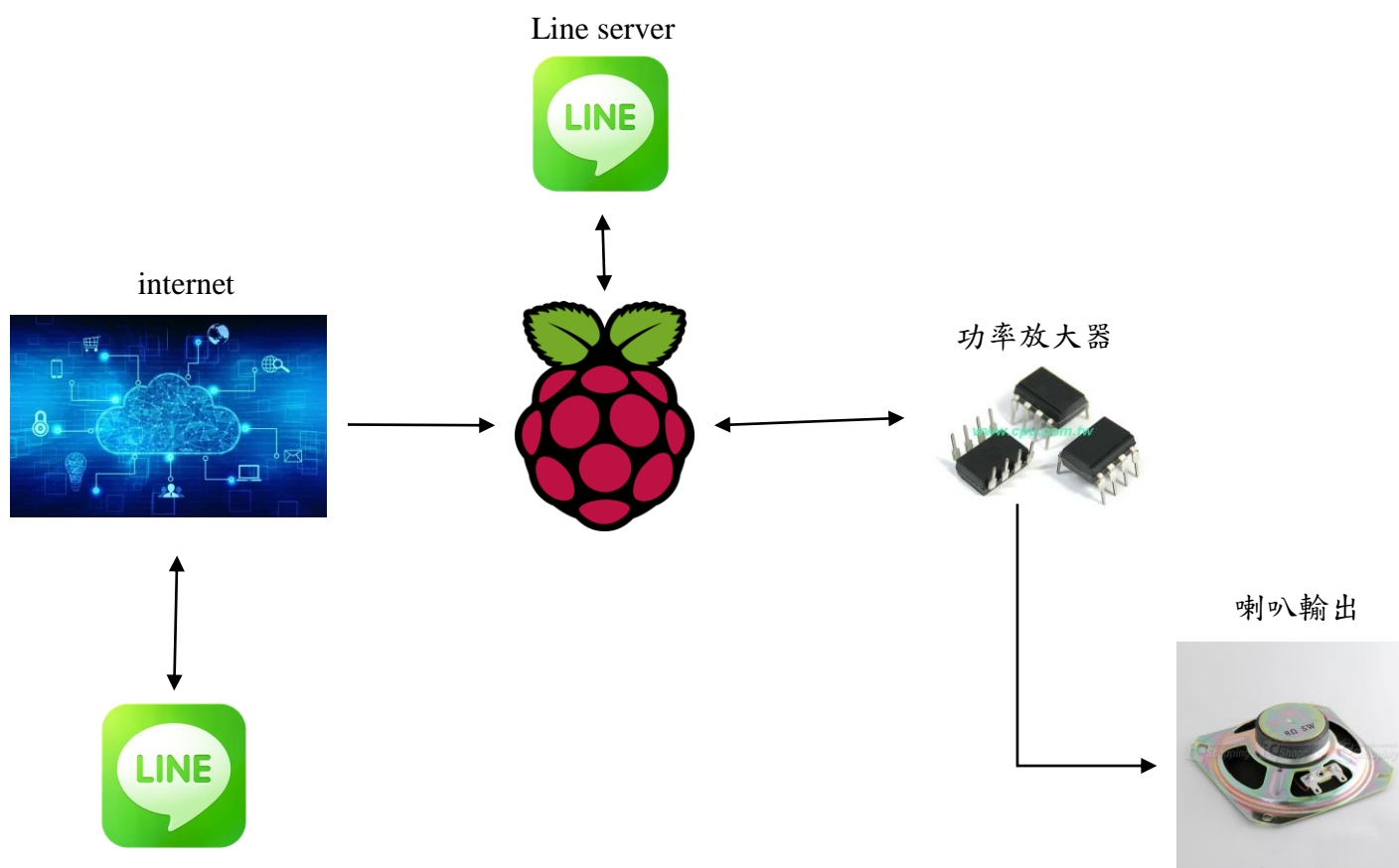


圖 4 架構圖

二、 專題流程圖

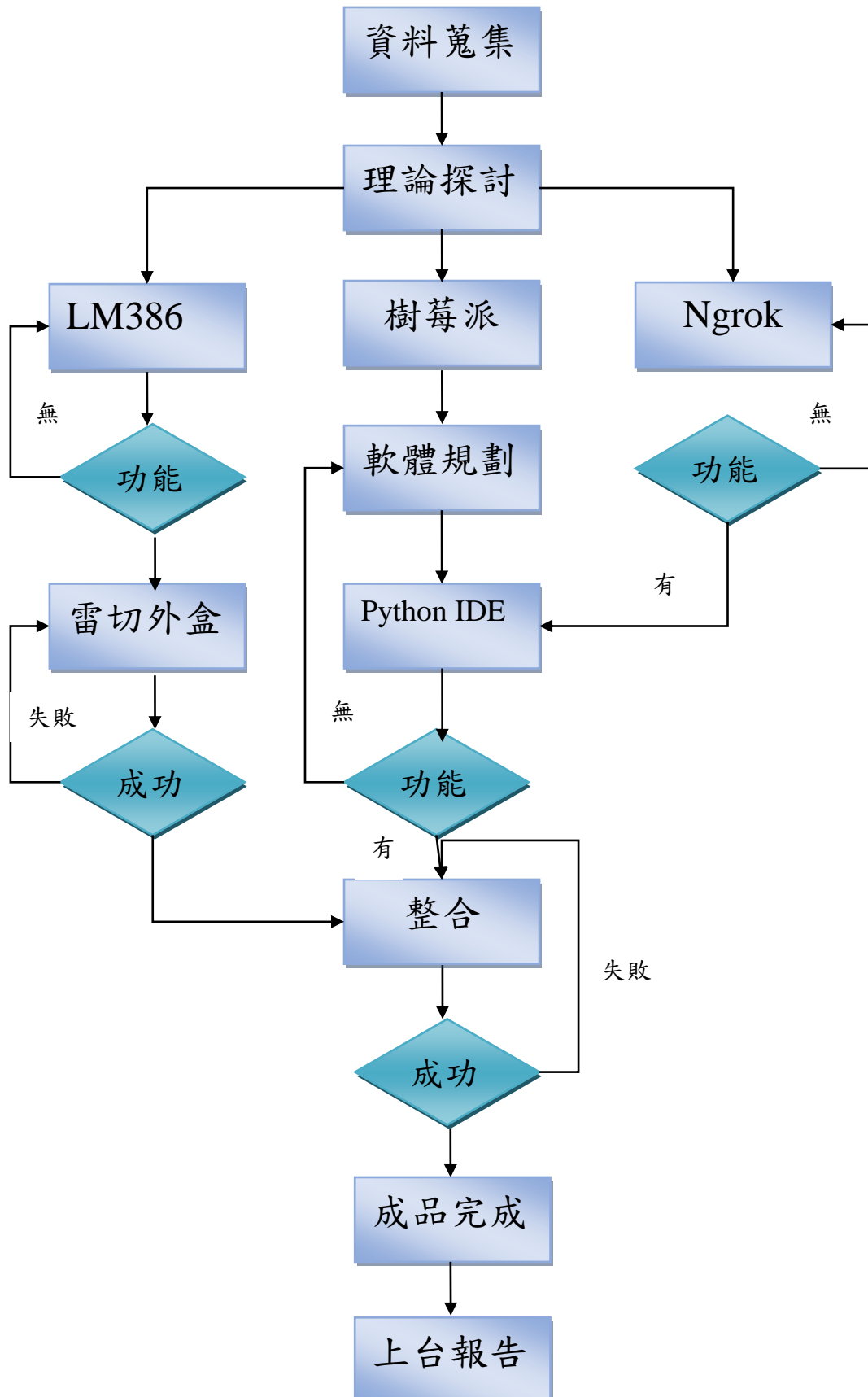


圖 5 流程圖

三、 專題甘特圖

工作項目	週次																				負責成員	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
專題準備	■																					全體組員
資料蒐集	■	■	■																			全體組員
理論探討	■	■	■																			全體組員
RaspberryPi OS 安裝				■																		邱
Python 程式撰寫					■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	邱
軟體控制					■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	邱
LM386 電路製作																	■					謝
雷射切割建模																			■	■	■	謝
外殼製作																				■	■	謝
軟硬體整合																				■	■	謝、邱
撰寫報告					■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	林、曾
成品完成																				■	■	全體組員
上台報告					■				■			■				■		■		■	■	全體組員
預定進度	5%	10%	15%	18%	20%	23%	25%	28%	30%	35%	40%	45%	50%	55%	60%	65%	70%	80%	90%	100%	累積百分比 %	

圖 6 專題甘特圖

肆、 專題成果

一、 歷程記錄

- (1) 為甚麼使用樹莓派來完成？

因為原本使用的 8266 晶片無法做到預期的效果。

- (2) 專題可以跟課堂上的內容結合應該很簡單吧？

因為課堂學習的部分不屬於我們需要的，因此模組和程式都必須重新修改。

- (3) 為何無法撥放聲音？

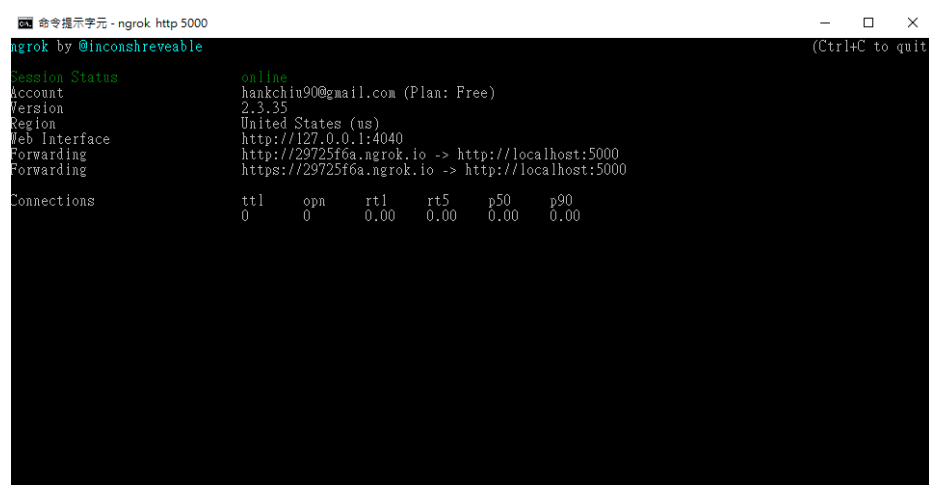
因為 Python 的聲音模組不相通，花了許多時間測試在 linux 上可以執行的聲音模組

- (4) 喇叭的聲音模糊不清？

喇叭雜訊過多，聲音太小，縮短電路之間的距離以減少雜訊，以及更換一顆歐姆數更大的喇叭

二、 成果

(一) Ngrok



```
ngrok by @Inconshreveable
Session Status      online
Account             hankchin90@gmail.com (Plan: Free)
Version             2.3.35
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding           http://29725f6a.ngrok.io -> http://localhost:5000
                   https://29725f6a.ngrok.io -> http://localhost:5000
Connections
  ttl    opn    rt1    rt5    p50    p90
   0      0     0.00  0.00  0.00  0.00
```

圖 7 Ngrok 執行圖

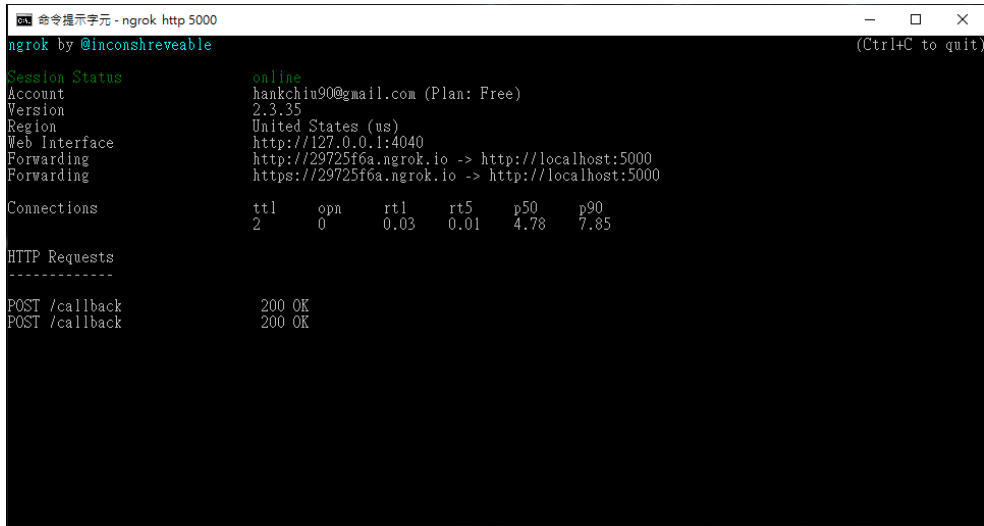


圖 8 執行成功

(二) Line

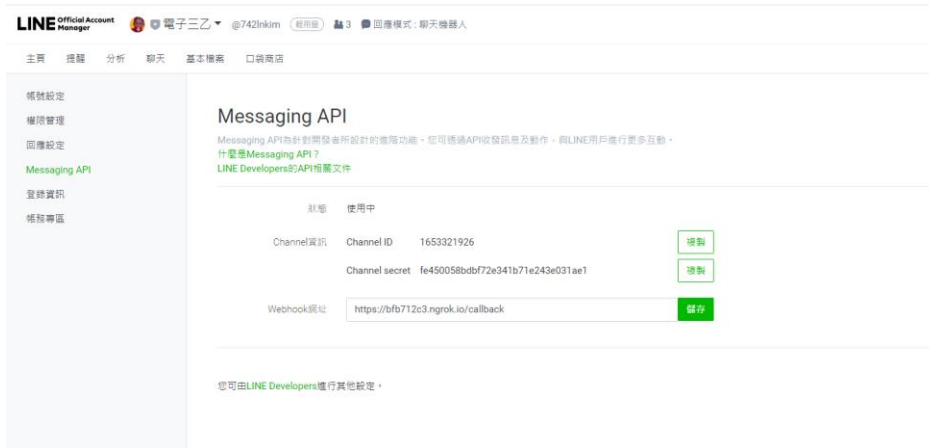


圖 9 Line server



1. 回傳
2. 回傳設定次數
3. 回傳定時
4. 回傳定時設定次數
5. 格式 (主訊息#時#分#次數)

圖 10 Line 執行圖

(三) 雷射切割設計圖

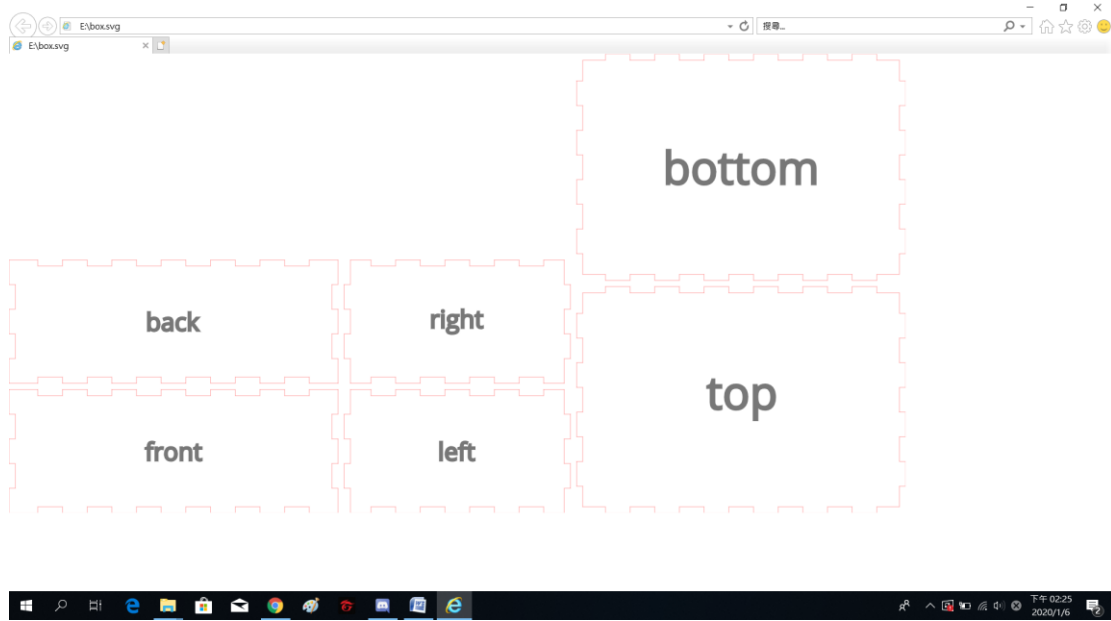


圖 11 外盒設計圖

(四) 專題實體

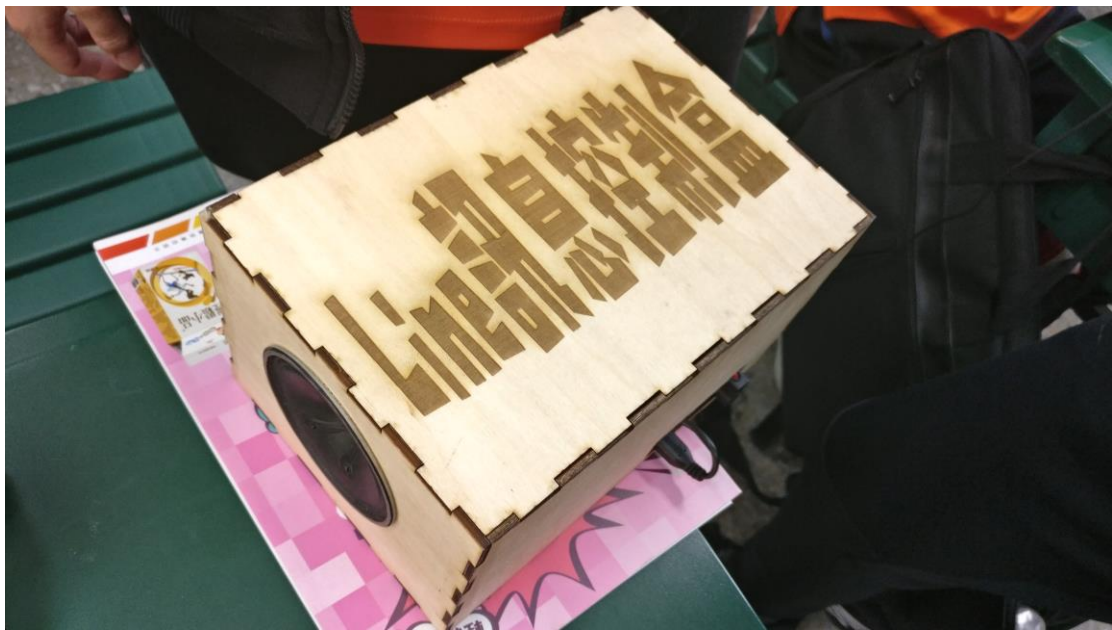


圖 12 專題完成圖

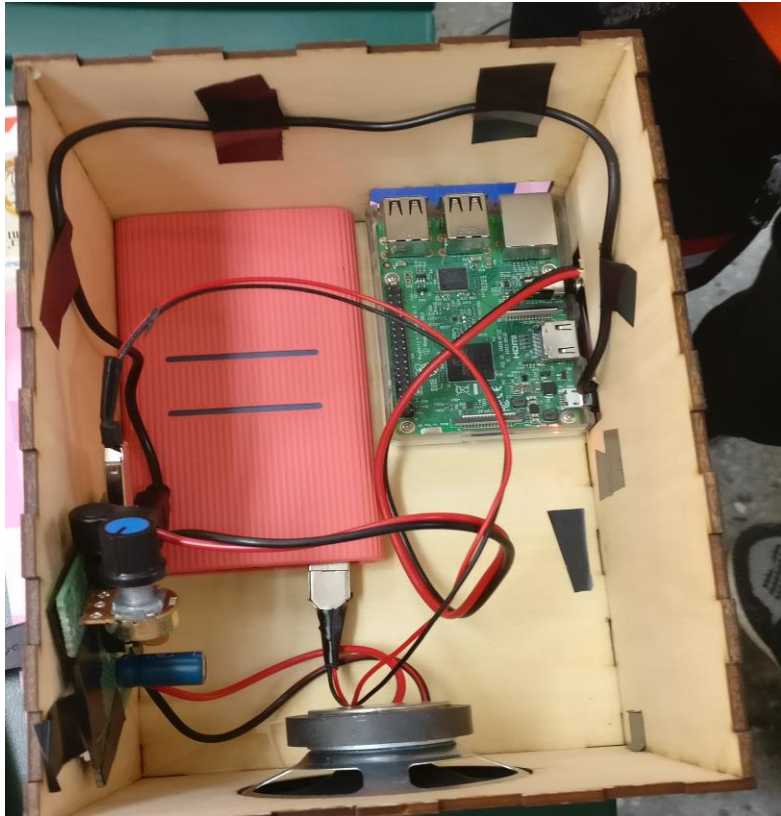


圖 13 內視圖

伍、 結論與建議

一、 結論

- (一) 可以完成一對一的訊息控制
- (二) 可以完成定址訊息控制
- (三) 可以預先設定時間，在正確的時間將傳送訊息
- (四) 可以在一段時間之內重複訊息

二、 建議

- (一) 因為我們是使用 LINEBOT 來完成回傳的功能，因此不能超出 LINE 自己本身的框架。

- (二) 因為是使用樹莓派進行開發，每次使用都必須加裝滑鼠,螢幕,鍵盤 之類的配件對於經常使用其實相當不便。
- (三) 樹梅派雖說基本等於一台個人電腦，但是處理速度無法與一般電腦相比，執行速度偏慢。
- (四) 因為同一群組只能允許存在一隻機器人，因此 LINE 無法進行 1 對多的傳輸，可以考慮其他媒介來完成此目標。
- (五) Ngrok 因為設定，每次重新開機都必須新生成網址。
- (六) 樹梅派使用配件較多，建議之後開發可以使用遠端操控的方式。
- (七) 將 Ngrok 改成其他網內穿透工具，例如 natapp

參考文獻

- 文淵閣工作室. (2019). Python 與 linebot 機器人. 基峰出版社
- 張義和 程兆龍. (2018). 微電腦應用實習. 新文京出版社
- 楊仁元 張顯盛 林家德. (2010). 專題製作理論與呈現技巧. 台科大出版社
- 董大偉. (2019). Linebot 與人工智慧辨識. 基峰出版社
- 電子電路 DIY 之家. (2017 年 10 月 7 日).
- 宅學習. (2009 年 1 月 2 日). Python. 擷取自 <https://sls.weco.net/CollectiveNote20/Python>
- lm386 講解 (參考) (包含電路圖). 擷取自 <https://kknews.cc/digital/kal646b.html>
- 揚聲器原理. 擷取自 <https://zh.wikipedia.org/wiki/%E6%8F%9A%E8%81%B2%E5%99%A8>
- 樹莓派介紹. 擷取自 <https://zh.wikipedia.org/wiki/%E6%A0%91%E8%8E%93%E6%B4%BE>
- Ngrok 介紹.IT 技術. 擷取自 <https://www.itread01.com/articles/1476116114.html>

附錄

附錄一 專題程式

```
from gtts import gTTS
from pygame import mixer
mixer.init()
import tempfile
import time
#import winsound
from flask import Flask, request, abort

from linebot import (LineBotApi, WebhookHandler)
from linebot.exceptions import (InvalidSignatureError)
from linebot.models import (MessageEvent, TextMessage, TextSendMessage,)

app = Flask(__name__)

line_bot_api =
LineBotApi('EwbKJLBgSXMMy3lSakp2FwnhhyySwiP8Y7nIM8bXl3kv0PQgvEdyiD1iVzntFOJSeinDSPOgJrrjX+7NTx6JfK7D
sOIOUuuUPe3M8m+Ttl+iah1YT7R9pQUuD5Nvg8cZUJO6yWuTJ+1QN7nek4fDOJAdB04t89/1O/w1cDnyilFU=')
handler = WebhookHandler('fe450058bdbf72e341b71e243e031ae1')

@app.route("/callback", methods=['POST'])
def callback():
    # get X-Line-Signature header value
    signature = request.headers['X-Line-Signature']

    # get request body as text
    body = request.get_data(as_text=True)
    app.logger.info("Request body: " + body)

    # handle webhook body
    try:
        handler.handle(body, signature)
    except InvalidSignatureError:
        abort(400)

    return 'OK'

@handler.add(MessageEvent, message=TextMessage)
def handle_message(event):
    line_bot_api.reply_message(event.reply_token, TextSendMessage(text=event.message.text))
    grab = event.message.text.split('#', 3)
    try:
        my_hour = str(grab[1])
        my_minute = str(grab[2])
        #repeat = str(grab[3])
```

```

#n = repeat
while True:
    current_time = time.strftime('%H{h}:%M{m}', time.localtime()).format(h='時',m='分')
    now = current_time.split(':')

    if my_hour == now[0] and my_minute == now[1]:
        try:
            repeat = str(grab[3])
            if repeat == str(grab[3]):
                for i in range(int(grab[3])):
                    time.sleep(1)
                    speak(str(grab[0]))
                break
        except:
            repeat = ""
            speak(str(grab[0]))
            print(str(grab[1]))
            print(str(grab[2]))
            break
    except:
        try:
            my_hour = ""
            my_minute = ""
            repeat = str(grab[1])
            if repeat == str(grab[1]):
                for i in range(int(grab[1])):
                    time.sleep(1)
                    print(str(grab[1]))
                    speak(str(grab[0]))
        except:
            my_hour = ""
            my_minute = ""
            repeat = ""
            speak(str(event.message.text))

def speak(text):
    with tempfile.NamedTemporaryFile(delete=True) as fp:
        tts = gTTS(text = text, lang='zh-TW')
        tts.save("{} .mp3".format(fp.name))
        mixer.music.load("{} .mp3".format(fp.name))
        mixer.music.play()

#speak(str(event.message.text))


if __name__ == "__main__":
    app.run()

```


--	--	--	--	--	--

附錄四 研究成員簡歷

姓名	邱康文	班級	電子三甲	 <p style="text-align: center;">照片</p>
曾修習專業科目	<ol style="list-style-type: none"> 1 基本電學與實習 2 電子學與實習 3 數位邏輯與實習 4 微處理機與實習 5 電子電路與實習 6 CPLD 邏輯實習 7 Visual Basic 程式設計實習 8 Altium Designer 電路圖與印刷電路板設計 			
參與專題工作項目	<ol style="list-style-type: none"> 1. 專題準備 2. 資料蒐集 3. 理論探討 4. 軟體製作 5. 程式設計 6. 口頭報告 			
經歷簡介	<ol style="list-style-type: none"> 1. 工業電子丙級技術士合格 2. 擔任 106 學年度輔導股長 			

姓名	林瑋翔	班級	電子三甲	
曾修專業科目	<ol style="list-style-type: none"> 1. 基本電學與實習 2. 電子學與實習 3. 數位邏輯與實習 4. 微處理機與實習 5. 電子電路與實習 6. CPLD 邏輯實習 7. Visual Basic 程式設計實習 8. Altium Designer 電路圖與印刷電路板設計 			
參與專題工作項目	<ol style="list-style-type: none"> 1. 專題準備 2. 資料蒐集 3. 理論探討 4. 計畫書製作 5. 專題網站製作 6. 簡報製作 7. 口頭報告 			
經歷簡介	<ol style="list-style-type: none"> 1. 工業電子丙級技術士合格 2. 擔任 106 學年度副班長 3. 擔任 106 學年度工場工具管理員 4. 擔任 106 學年度輔導小義工 5. 擔任 107 學年度輔導小義工 			

姓名	曾宏義	班級	電子三甲	 <p data-bbox="1134 882 1193 913">照片</p>
曾修習專業科目	<ol style="list-style-type: none"> 1. 基本電學與實習 2. 電子學與實習 3. 數位邏輯與實習 4. 微處理機與實習 5. 電子電路與實習 6. CPLD 邏輯實習 7. Visual Basic 程式設計實習 8. Altium Designer 電路圖與印刷電路板設計 			
參與專題工作項目	<ol style="list-style-type: none"> 1. 專題準備 2. 資料蒐集 3. 理論探討 4. 計劃書製作 5. 專題網站製作 6. 簡報製作 7. 口頭報告 			
經歷簡介	<ol style="list-style-type: none"> 1. 工業電子丙級技術士合格 2. 擔任 106 學年度工場安全衛生股長 3. 擔任 106 學年度學藝股長 4. 擔任 107 學年度工場材料管理員、領班 5. 擔任 108 學年度工場領班 			

姓名	謝宗祐	班級	電子三甲	 <p style="text-align: center;">照片</p>
曾修習專業科目	<ol style="list-style-type: none"> 1. 基本電學與實習 2. 電子學與實習 3. 數位邏輯與實習 4. 微處理機與實習 5. 電子電路與實習 6. CPLD 邏輯實習 7. Visual Basic 程式設計實習 8. Altium Designer 電路圖與印刷電路板設計 			
參與專題工作項目	<ol style="list-style-type: none"> 1. 專題準備 2. 資料蒐集 3. 理論探討 4. 硬體製作 5. 外盒設計 6. 雷射切割 7. 口頭報告 			
經歷簡介	<ol style="list-style-type: none"> 1. 工業電子丙級技術士合格 2. 擔任 107 學年度風紀股長 3. 擔任 107 學年度學藝股長 4. 擔任 108 學年度學藝股長 			