

南 臺 科 技 大 學

資 訊 管 理 系

108 學 年 度 畢 業 專 題

樹 莓 派 之 物 聯 網 應 用

指 導 老 師：王 派 洲 教 授

組 員		
班 級	學 號	姓 名
資管四甲	4A590003	黃智暘
資管四甲	4A590045	楊豐嘉
資管四甲	4A590127	高胤恒
資管四甲	4A590148	蔡宗傑

中 華 民 國 一〇九 年 五 月

南臺科技大學
資訊管理系

108學年度畢業專題

樹莓派之物聯網應用

中華民國一〇九年五月出版

摘要

近期資訊科技發展迅速，許多新技術、新科技問世，其中物聯網技術在工業 4.0 的熱潮下被發掘出其價值。本專題為介紹與認識物聯網之功能與應用，以可在空中彈奏的鋼琴為展示實物，也可用於手機解鎖之實務應用，透過此裝置可讓國中小學生簡單了解物聯網技術原理及未來的應用，使其對物聯網科技產生興趣。本專題將使用超音波感應器作為偵測裝置，偵測手勢高低與否判斷音階，再透過樹莓派運算後以藍芽通訊裝置傳至揚聲器。

關鍵字：物聯網、工業 4.0、超音波感應器、樹莓派

目錄

壹、緒論	1
一、研究動機	1
二、研究目的	2
三、原創性	2
四、設計原理	3
五、軟體需求分析	3
貳、文獻探討	5
一、起始	5
二、HC-SR04 超音波距離感測器	6
三、Raspberry Pi 樹莓派	7
1. 什麼是樹莓派？	7
2. 軟硬體設備	7
參、研究方法	8
一、系統架構圖	8
二、開發技術平台	9
1. Python	9
2. 樹莓派(Linux)	9
3. 超音波感測元件	9
4. 藍芽喇叭	10
5. Sonic pi	10
三、本系統流程圖	11
四、系統功能架構圖	12
五、內部程式碼	15
1. 設定 GPIO 相關資訊	15
2. 主程式	16
3. 超音波感測器_1	17
4. 超音波感測器_2	18
肆、測試規劃	19
一、測試規劃	19
二、測試結果	24

伍、研究成果.....	30
一、成品展示	30
陸、系統測試結論與未來研究方向.....	32
一、技術方面	32
二、團隊合作方面	33
柒、工作分解.....	34
一、工作分解	34
二、任務分配表	39
1.組長.....	39
2.系統分析人員	39
3.程式設計人員	39
4.測試人員	39
5.技術支援人員	39
6.文書管理人員	39
三、專案管理方式	41
四、資源平滑	43
捌、參考文獻.....	44
一、Raspberrypi.....	44
二、VR 空氣鋼琴.....	44
三、Sonic pi.....	44
四、w3schools-Python	44
五、天下雜誌:全球物聯網科技先驅 Tibbo 與 UPS 攜手，搶佔 AIoT 商機 .44	
六、光感應鋼琴結構，黃品瑞，公告號:M477021(2014/01/15).....44	
七、新通訊:5G 風潮推波 IoT 戰場決勝在應用	44
八、維基百科:工業 4.0	44
九、鋼琴彈奏手部教學裝置，謝哲人、高嘉駿、廖瑾辰，公告號:I623922 (2017/05/05).....	44
十、科技網:用科技對抗新冠肺炎 台灣微軟聚焦 4 大 AI 應用場景	44
玖、自傳與簡歷	45

圖目錄

圖片 1.1 使用者案例圖	4
圖片 3.1 系統架構圖	8
圖片 3.2 PYTHON.....	9
圖片 3.3 樹莓派.....	9
圖片 3.4 超音波感測元件.....	9
圖片 3.5 BLUETOOTH	10
圖片 3.6 SONIC PI	10
圖片 3.7 流程圖	11
圖片 3.8 物聯網距離感測超音波子系統	12
圖片 3.9 藍芽音響控制子系統.....	13
圖片 3.10 鋼琴彈奏輔助子系統.....	14
圖片 5.1 作品圖(1)	30
圖片 5.2 作品圖(2)	30
圖片 5.3 作品圖(3)	31
圖片 5.4 作品圖(4)	31
圖片 7.1 工作分解圖	34
圖片 7.2 時程規劃(1).....	41
圖片 7.3 時程規劃(2).....	42
圖片 7.4 資源平滑圖(1).....	43
圖片 7.5 資源平滑圖(2).....	43

表目錄

表格 4.1	測試規劃 RASPBERRY PI 作業系統安裝與 PYTHON 版本確認.....	19
表格 4.2	測試規劃 PYTHON 3.6 環境建置	20
表格 4.3	測試規劃 測試 GPIO 套件_1(單個超音波感測器作用).....	20
表格 4.4	測試規劃 測試 GPIO 套件_2(單個超音波感測器作用).....	21
表格 4.5	測試規劃 測試 GPIO 套件_1+GPIO 套件_2(兩個超音波感測器作用)	21
表格 4.6	測試規劃 測試 GPIO 套件_1+GPIO 套件_2(兩個超音波感測器作用)	22
表格 4.7	測試規劃 SONIC-PI 測試	22
表格 4.8	測試規劃 GPIO-4 + SONICPI-1	23
表格 4.9	測試結果 RASPBERRY PI 作業系統安裝與 PYTHON 版本確認	24
表格 4.10	測試結果 PYTHON 3.6 環境建置.....	25
表格 4.11	測試結果	25
表格 4.12	測試結果 測試 GPIO 套件_2(單個超音波感測器作用).....	26
表格 4.13	測試結果 測試 GPIO 套件_1+GPIO 套件_2(錯誤).....	27
表格 4.14	測試結果 測試 GPIO 套件_1+GPIO 套件_2(成功).....	28
表格 4.15	測試結果 SONIC-PI 測試	29
表格 4.16	測試結果 GPIO-4 + SONICPI-1	29
表格 7.1	工作分解 專題聯絡.....	34
表格 7.2	工作分解 系統安裝	35
表格 7.3	工作分解 連接藍芽	35
表格 7.4	工作分解 超音波系統	36
表格 7.5	工作分解 距離計算	36
表格 7.6	工作分解 彈奏混音	37
表格 7.7	工作分解 書面撰寫	37
表格 7.8	工作分解 系統測試	38
表格 7.9	工作分解 口頭報告	38
表格 7.10	任務分派表(1).....	40
表格 7.11	任務分派表(2).....	40

壹、 緒論

一、 研究動機

近期資訊科技發展迅速，許多新技術、新科技問世，其中物聯網技術在工業 4.0 的熱潮下被發掘出其價值，而近年而隨著通訊技術 5G 的研究發展，更使得 IoT 距離實現又更近了一步。本專題為介紹與認識物聯網之功能與應用，物聯網是將各自獨立運行的電子產品、物體透過網際網路實現互聯互通的技術，此概念起源於比爾·蓋茲(Bill Gates)於 1995 年出版的《未來之路》書中提出物品互聯的概念，在 1998 年麻省理工學院提出了當時被稱作 EPC 系統的 IoT 構想，於 2000 年以後，網路逐漸普及且在近幾十年間急遽成長，使得此一構想有了實現的可能，在現今構想的物聯網中使用了許多新的技術如 IPv6、Web3.0 等；在各項領域上也促進了許多新營的想法產生如工業 4.0、商業 4.0、農業 4.0 等等。本專題以可在空中彈奏的鋼琴為展示實物，也可用於手機解鎖之實務應用，透過此裝置可讓年齡尚幼的兒童培養音感，作為玩具能夠寓教於樂。本專題將使用超音波感應器作為偵測裝置，偵測手勢高低與否判斷音階，再透過樹莓派運算後以藍芽通訊裝置傳至揚聲器。

二、 研究目的

本專題用 Python 做為主要編寫的程式工具，再利用樹莓派其多樣化的連接槽來實現不同方面及功能的應用。實體裝置方面是以樹莓派連接超音波感測裝置作用於手勢感測，並回傳樹莓派運算其音調為何並利用藍芽傳輸系統連接音響發出聲調。超音波感測裝置為主要感測裝置，前述之感測手勢原理係利用手掌與感測器間所距離之高度，判斷其位於高度應為何種音階，音階之判斷依據為五線樂譜上各音階所在位置作為參考，在樹莓派中運算其音調，如果遇到雙手時輸出為混音，待樹莓派完成其運算後以藍芽傳輸系統至藍芽音響發聲。除用於鋼琴彈奏之外，本專題未來也可連接如電視等電子產品，透過超音波裝置感測來達到手勢、體感控制等效果。預期本專題之目標對象能夠瞭解到：新的資訊科技結合生活的事物也能夠很有趣，預期能夠有不錯的效果。

三、 原創性

市面上的超音波應用繁多，但是針對音樂這一方面較少，所以我們針對這一類的需求設計出了這個軟硬體。其中我們使用了 Raspberry Pi、Python、Sonic pi，其餘的程式碼皆由我們自行撰寫。我們參考了網路上的一寫相關資訊，設計出了較好的軟硬體，讓使用者可以更方便的去使用我們的產品。在這個資訊爆炸的時代裡，希望能夠讓我們的產品帶給使用者諸多的方便以及趣味性。

四、 設計原理

利用超音波碰到物體會回彈的特性製作超音波鋼琴，藉由左右兩邊發送超音波使超音波互不干擾，以達到彈出不同音階及音調的目的。裝置前方為喇叭，聲音輸出供操作者清楚聽到聲音。

五、 軟體需求分析

1. 輸入功能

利用超音波感測距離。

2. 運算功能

利用得到的距離轉換成音階。

彈奏出的音階要能夠混音或曲調和弦。

3. 輸出功能

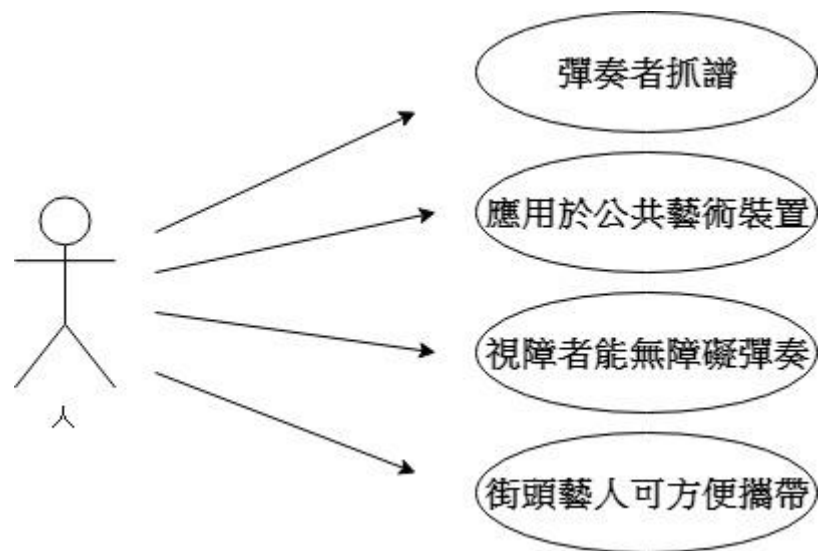
利用藍芽傳輸運算的資料到藍芽喇叭。

4. 使用案例

本案可應用於其他樂器(如鋼琴、豎琴、吉他等)之彈奏。

5. 使用者案例圖

圖一為本系統之系統使用者案例圖，右方每個橢圓分別代表每個使用案例。



圖片 1.1 使用者案例圖

貳、 文獻探討

一、 起始

本專題將物聯網技術與樂器結合，針對物聯網在即將來臨的 5G 時代底下所衍生出的工業 4.0 的概念，以及此技術如何走入我們的生活的各個角落等技術應用及相較以往有哪些不同的差異性。

2016 年 10 月廖專崇在 <5G 風潮推波 IoT 戰場決勝在應用> 一文中提到聯網裝置的出貨量正快速的上升當中，根據 BI Intelligence 估計，到 2017 年聯網裝置出貨量將超越智慧型手機，顯示出隨著行動裝置逐漸普及，為能夠讓行動裝置有能夠更大的應用範圍，將各式機具、家電等電子產品連接網路，這是人類將生活數位化的路程中所必要的技術，蔡騰輝(2020)也提到尤其在 2020 年肺炎疫情大流行的情況下更進一步加速了物聯網在醫療方面的發展，也是人們開始走入數位化生活的重要契機。

二、 HC-SR04 超音波距離感測器

當持續時間至少為 $10\ \mu\text{S}$ (10 微秒) 的脈衝施加到觸發引腳時，一切就開始了。響應於此，傳感器以 40 KHz 發射八個脈衝的聲音脈衝。這種 8 脈衝模式使設備的“超聲特徵”變得獨一無二，從而使接收器能夠將發射模式與環境超聲噪聲區分開。八個超聲波脈衝通過空氣傳播，遠離變送器。同時，迴聲引腳變為高電平，開始形成迴聲信號的開始。

如果這些脈衝沒有被反射回來，則回波信號將在 38 毫秒後超時並返回低電平。因此，38 ms 的脈衝表示在傳感器範圍內沒有阻塞。

如果這些脈衝被反射回去，則在收到信號後，Echo 引腳就會變低。這會產生一個脈衝，其寬度在 $150\ \mu\text{S}$ 至 25 mS 之間變化，具體取決於接收信號所花費的時間。

三、 Raspberry Pi 樹莓派

1. 什麼是樹莓派？

Raspberry Pi, 中文:樹莓派, 是一款基於 Linux 系統的單板機電腦, 體積只有一張信用卡的大小, 價格便宜, 只有 25~35 美金。

它由英國的樹莓派基金會所開發, 目的是以低價硬體及自由軟體刺激在學校的基本的電腦科學教育。

由於樹莓派設計有對外的介面(GPIO), 可讓程式師控制外部的自組電路, 因此樹莓派慢慢也被考慮用來進行機電控制方面的實習工具。

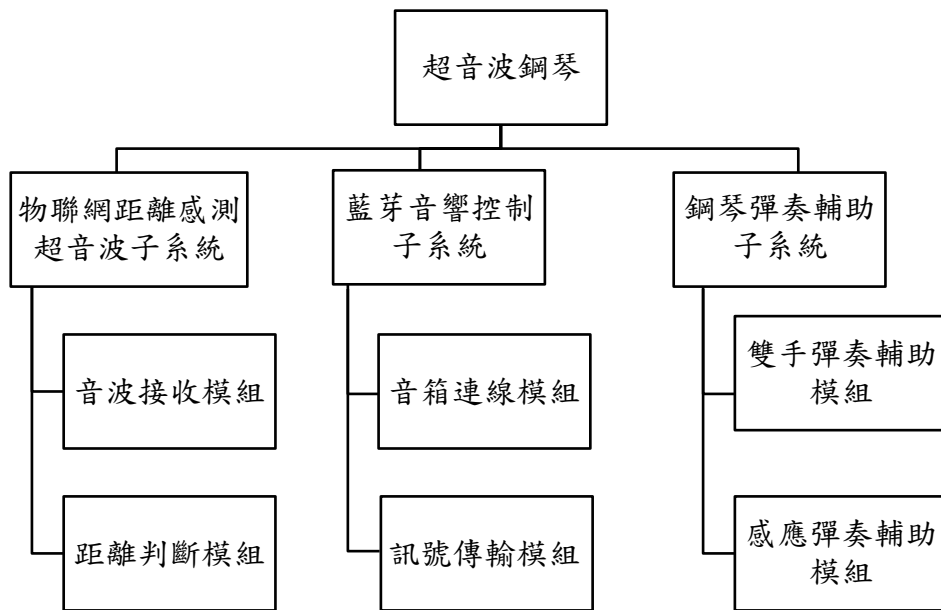
2. 軟硬體設備

樹莓派配備一枚 700MHz 博通出產的 ARM 架構 BCM2835 處理器, 256MB 內存 (B 型為 512MB), 使用 SD 卡當作儲存媒體, 擁有一個 Ethernet、兩個 USB 介面、以及 HDMI 和 RCA 端子輸出支援。

操作系統採用開源的 Linux 系統, 自帶的軟體能夠滿足基本的網路瀏覽、文字處理以及電腦學習的需要。

參、 研究方法

一、 系統架構圖



圖片 3.1 系統架構圖

本專題分為三個子系統，分別為物聯網距離感測超音波子系統、藍芽音響控制子系統、鋼琴彈奏輔助子系統。物聯網距離感測超音波子系統可以偵測手與感測器間的距離，用來判斷位置決定其音階；藍芽音響控制子系統將訊號傳輸至藍芽音響，發出聲調；琴彈奏輔助子系統作用於使用者雙手間彈奏的混音及感測。

二、 開發技術平台



圖片 3.2 Python

1. Python

Python，是一種廣泛使用的高階程式語言，Python 的設計哲學強調程式碼的可讀性和簡潔的語法。相比於 C++ 或 Java，Python 讓開發者能夠用更少的代碼表達想法。



圖片 3.3 樹莓派

2. 樹莓派(Linux)

樹莓派(Raspberry Pi)面積只有一張信用卡大小，體積大概是一個火柴盒大小，操作系統採用開源的 Linux 系統。



圖片 3.4 超音波感測元件

3. 超音波感測元件

感測器 (Sensor) 是用於偵測環境中所生事件或變化，並將此訊息傳送出至其他電子裝置 (如 CPU) 的裝置，通常由敏感元件和轉換元件組。



圖片 3.5 Bluetooth

4. 藍芽喇叭

藍牙 (Bluetooth)，一種無線通訊技術標準，用來讓固定與行動裝置，在短距離間交換資料，以形成個人區域網路 (PAN)。

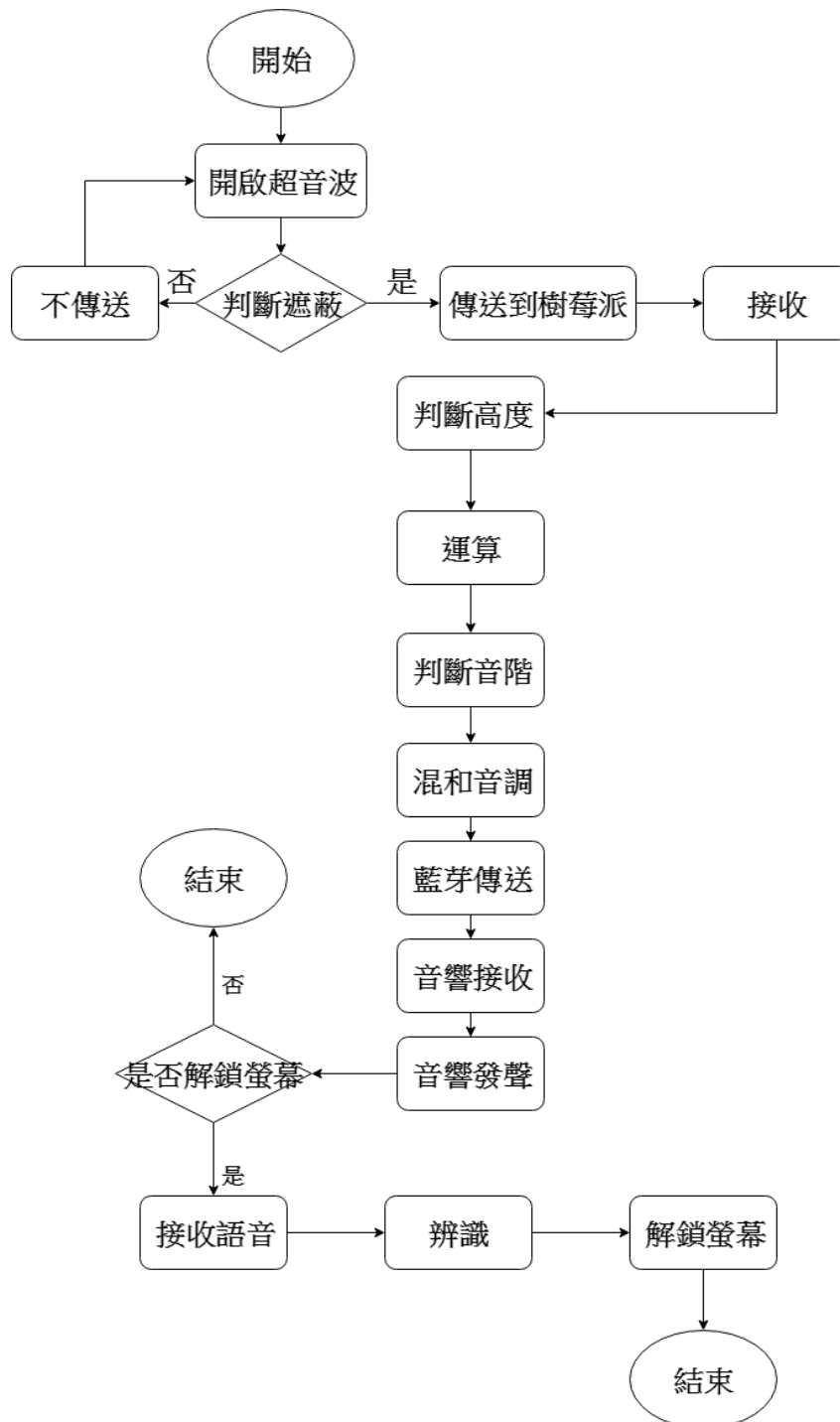


圖片 3.6 Sonic pi

5. Sonic pi

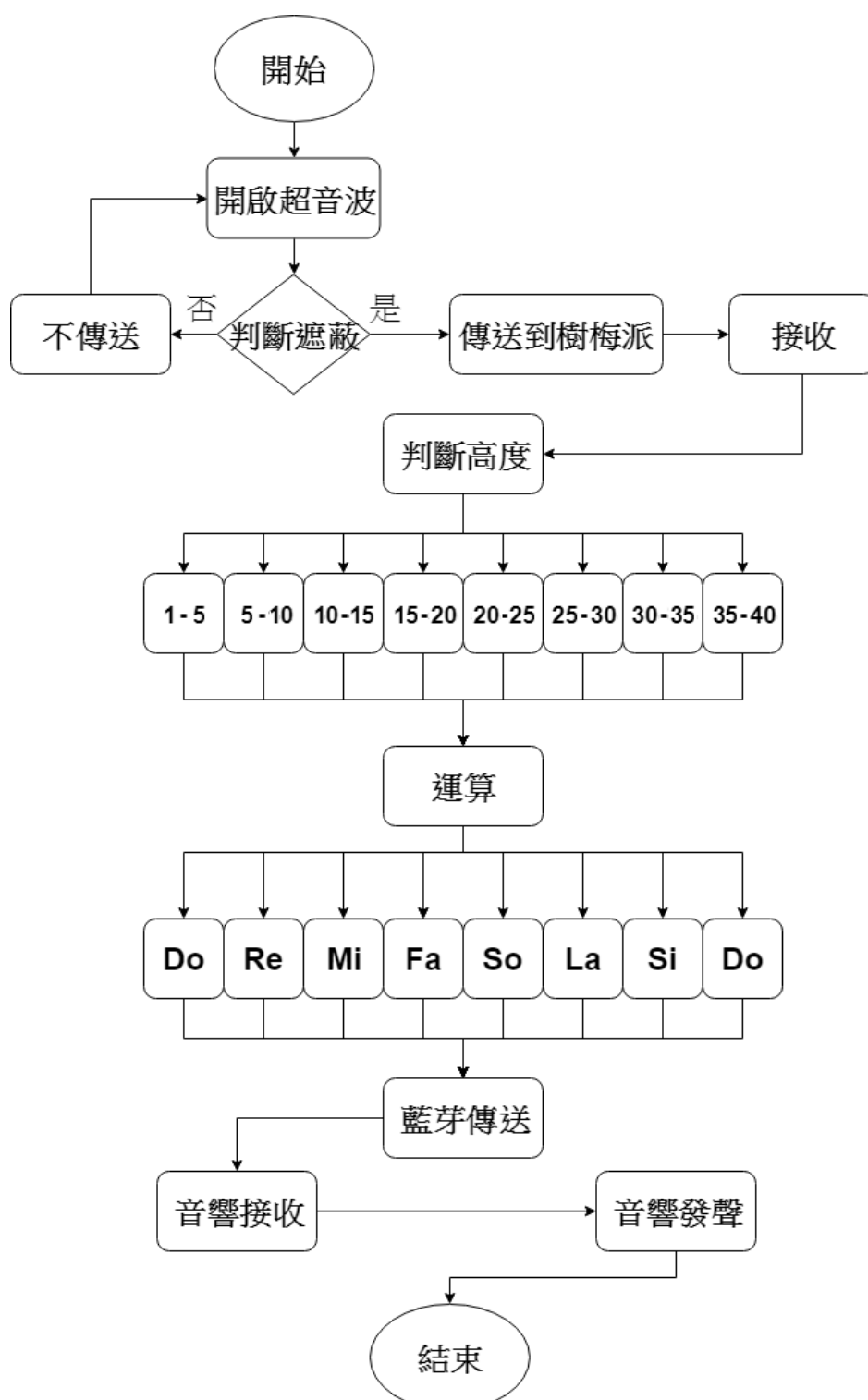
Sonic Pi 是基於 Ruby 的實時編碼環境，最初是為支持學校中的計算和音樂課程而設計的，由劍橋大學計算機實驗室的 Sam Aaron 與 Raspberry Pi Foundation 合作開發。由於使用了 SuperCollider 綜合引擎和準確的時序模型，還用於現場編碼和其他形式的算法音樂表演和製作，包括 algoraves。它的研究和發展已受到支持內斯塔(通過 Sonic PI：實施編碼項目)。

三、 本系統流程圖

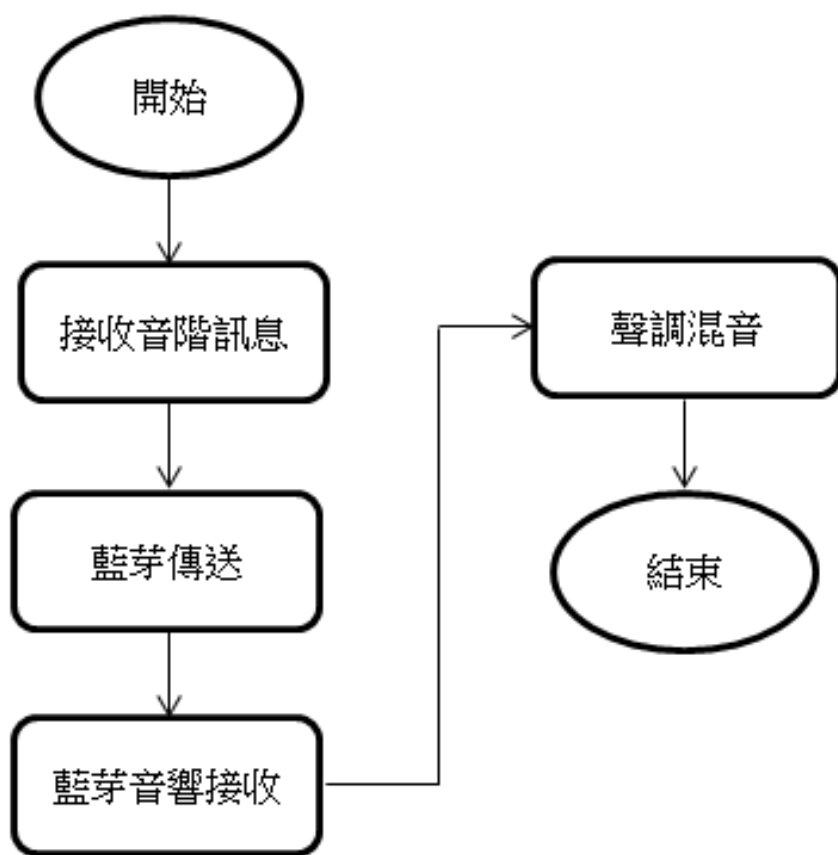


圖片 3.7 流程圖

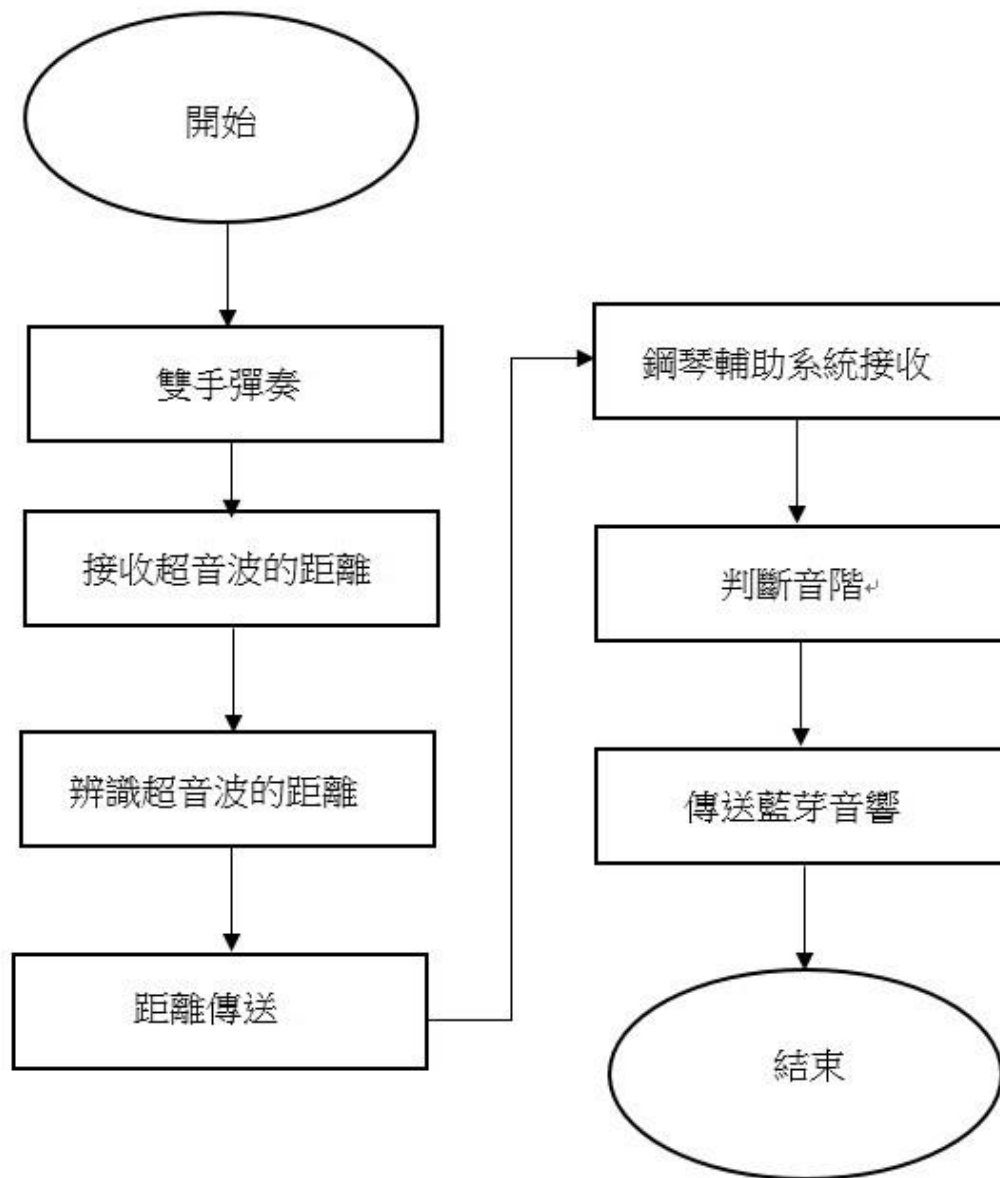
四、系統功能架構圖



圖片 3.8 物聯網距離感測超音波子系統



圖片 3.9 藍芽音響控制子系統



圖片 3.10 鋼琴彈奏輔助子系統

五、 內部程式碼

1. 設定 GPIO 相關資訊

```
1. import RPi.GPIO as GPIO
2. import time
3. from psonic import *
4. from pathlib import Path
5. import numpy as np
6. from threading import Thread, Condition
7. #設定兩個超音波
8. TRIGGER_PIN_1 = 25
9. ECHO_PIN_1 = 8
10. TRIGGER_PIN_2 = 24
11. ECHO_PIN_2 = 10
12. GPIO.setmode(GPIO.BCM)
13. #pin 編碼方式
14. GPIO.setup(TRIGGER_PIN_1, GPIO.OUT)
15. GPIO.setup(ECHO_PIN_1, GPIO.IN)
16. GPIO.setup(TRIGGER_PIN_2, GPIO.OUT)
17. GPIO.setup(ECHO_PIN_2, GPIO.IN)
18. GPIO.output(TRIGGER_PIN_2, GPIO.LOW)
19. #設 trig 低電位
20. GPIO.output(TRIGGER_PIN_1, GPIO.LOW)
21. use_synth(PIANO)
22. #設 trig 低電位
23. time.sleep(1)
24. #time.sleep(1)
25. condition = Condition()
26. key2=-1  key1=-1
27. # 4hz thread is the same as running 'countTo(4, 0.25)'
28. sonicsensor1= Thread(target=sonic1, args=(condition,))
29. # 10hz thread is the same a
30. sonicsensor2= Thread(target=sonic2, args=(condition,))
31. playthread=Thread(target=playnote, args=(condition,))
32. sonicsensor1.start()
33. sonicsensor2.start()
34. playthread.start()
```

2. 主程式

```
1. #function for tone sensor
2.
3. def playnote(condition):
4.     global key2
5.     global key1
6.
7.     ironman_empty = np.array([
8.         [48, 50, 52, 53, 55, 57, 59],
9.         [60, 62, 64, 65, 67, 79, 71],
10.        [72, 74, 76, 77, 79, 81, 83]])
11.     try:
12.         while True:
13.             with condition:
14.                 condition.wait()
15.                 if (key1!=-1) and (key2!=-1) and (key2!=3):
16.
17.                     play(ironman_empty[key2, key1])
18.                     sleep(1)
19.     except KeyboardInterrupt:
20.         print('關閉程式')
```

3. 超音波感測器_1

```
1. # 判斷音階
2. def sonic1(condition):
3.     global key1
4.     tkey=-1
5.     try:
6.         while True:
7.             GPIO.output(25, GPIO.HIGH)
8.             time.sleep(0.00001)
9.             GPIO.output(25, GPIO.LOW)
10.            start_time=time.time()
11.            end_time=time.time()
12.            while GPIO.input(8) == 0:#
13.                start_time = time.time()
14.            while GPIO.input(8) == 1:
15.                end_time = time.time()
16.            etime = end_time - start_time
17.            distance1 = 17150 * etime
18.            print('sensor 1 距離為 {:.1f} 公分'.format(distance1))
19.            if(distance1<10.0 and distance1>00.0):tkey=0
20.            if(distance1<20.0 and distance1>10.1):tkey=1
21.            if(distance1<30.0 and distance1>20.1):tkey=2
22.            if(distance1<40.0 and distance1>30.1):tkey=3
23.            if(distance1<50.0 and distance1>40.1):tkey=4
24.            if(distance1<60.0 and distance1>50.1):tkey=5
25.            if(distance1<70.0 and distance1>60.1):tkey=6
26.            if (key1== -1) :
27.                key1=tkey
28.                with condition:
29.                    condition.notifyAll()
30.            else:
31.                if (key1!=tkey):
32.                    key1=tkey
33.                    with condition:
34.                        condition.notifyAll()
35.                    time.sleep(0.5)
36.        except KeyboardInterrupt: print('關閉程式')
```


4. 超音波感測器_2

```
1. def sonic2(condition):
2.     global key2
3.     tkey=-1
4.     try:
5.         while True:
6.             GPIO.output(24, GPIO.HIGH)
7.             time.sleep(0.00001)
8.             GPIO.output(24, GPIO.LOW)
9.             start_time=time.time()
10.            end_time=time.time()
11.            while GPIO.input(10) == 0:
12.                start_time = time.time()
13.            while GPIO.input(10) == 1:
14.                end_time = time.time()
15.            etime = end_time - start_time
16.            distance1 = 17150 * etime
17.            print('sensor 2 距離為 {:.1f} 公分'.format(distance1))
18.            if(distance1<10.0 and distance1>00.0):tkey=0
19.            if(distance1<20.0 and distance1>10.1):tkey=1
20.            if(distance1<30.0 and distance1>20.1):tkey=2
21.            if(distance1>30.1):tkey=3
22.            if (key2== -1):
23.                if (tkey!=3):
24.                    key2=tkey
25.                    with condition:
26.                        condition.notifyAll()
27.            else:
28.                if (key2!=tkey and tkey!=3):
29.                    key2=tkey
30.                    with condition:
31.                        condition.notifyAll()
32.            else:
33.                key2=-1
34.            time.sleep(0.5)
35.        except KeyboardInterrupt:
36.            print('關閉程式')
```

肆、 測試規劃

一、 測試規劃

表格 4.1 測試規劃 Raspberry Pi 作業系統安裝與 Python 版本確認

需求 ID	Raspberry		
需求名稱	Raspberry Pi 作業系統安裝與 Python 版本確認		
測試案例 ID	Raspberryl-1		
測試情節名稱	作業系統安裝		
情節內容	步驟	操作說明	預期結果
	1	用 SDFormatter 把 OS 丟到 SD 卡	成功在 SD 資料夾發現檔案
	2	SD 卡插入 Raspberry Pi 並進行 OS 安裝	Raspberry Pi 可讀取 SD 卡且執行
	3	基本設定完成	可以正常使用 Raspberry Pi OS
	4	檢查 Python 版本是否可用	希望為 Python3.6
測試指引	可正常進入系統畫面		
備註	Python 版本需重新建置		

表格 4.2 測試規劃 Python 3.6 環境建置

需求 ID	Raspberry		
需求名稱	Python 3.6 環境建置		
測試案例 ID	Raspberryl-2		
測試情節名稱	環境安裝		
情節內容	步驟	操作說明	預期結果
	1	下載 miniconda 安裝程式(用來安裝 Python3.6)	安裝成功
	2	執行安裝程式設定 Path	設定成功
	3	conda install python=3.6	沒有錯誤反應
	4	確認 Python 版本	Python3.6
測試指引	終端機中輸入 Python 指令彈出版本為 3.6		
備註			

表格 4.3 測試規劃 測試 GPIO 套件_1(單個超音波感測器作用)

需求 ID	GPIO		
需求名稱	測試 GPIO 套件_1(單個超音波感測器作用)		
測試案例 ID	GPIO-1		
測試情節名稱	GPIO 套件_1		
情節內容	步驟	操作說明	預期結果
	1	Raspberry Pi 連接感測器 TRIGGER_PIN = 25 ECHO_PIN = 8	線與孔位置正確
	2	程式碼測試	套件都可使用
	3	程式執行	正常執行
	4	終端機收到傳回距離(備註)	sensor 1 距離為 10.5 公分
測試指引	感測器可正常運作並傳回距離		
備註	sensor 1 距離為 10.5 公分 sensor 1 距離為 31.5 公分' sensor 1 距離為 41.5 公分' sensor 1 距離為 51.5 公分'		

表格 4.4 測試規劃 測試 GPIO 套件_2(單個超音波感測器作用)

需求 ID	GPIO		
需求名稱	測試 GPIO 套件_2(單個超音波感測器作用)		
測試案例 ID	GPIO-2		
測試情節名稱	GPIO 套件_2		
測試情節	步驟	操作說明	預期結果
	1	Raspberry Pi 連接感測器 TRIGGER_PIN = 24 ECHO_PIN = 10	線與孔位置正確
	2	程式碼測試	套件都可使用
	3	程式執行	正常執行
	4	終端機收到傳回距離(備註)	sensor 2 距離為 10.5 公分
測試指引	感測器可正常運作並傳回距離		
備註	sensor 2 距離為 10.5 公分 sensor 2 距離為 31.5 公分' sensor 2 距離為 41.5 公分' sensor 2 距離為 51.5 公分'		

表格 4.5 測試規劃 測試 GPIO 套件_1+GPIO 套件_2(兩個超音波感測器作用)

需求 ID	GPIO		
需求名稱	測試 GPIO 套件_1+GPIO 套件_2(兩個超音波感測器作用)		
測試案例 ID	GPIO-3		
測試情節名稱	GPIO 套件_1+2		
測試情節	步驟	操作說明	預期結果
	1	Raspberry Pi 連接感測器 TRIGGER_PIN = 25 ECHO_PIN = 8 TRIGGER_PIN = 24 ECHO_PIN = 10	線與孔位置正確
	2	程式碼測試	套件都可使用
	3	程式執行	失敗。會卡在第一個 while 迴圈
	4	終端機收到傳回距離(備註)	失敗
測試指引	感測器可正常運作並傳回距離		
備註	sensor 1 距離為 10.5 公分 sensor 1 距離為 31.5 公分' sensor 1 距離為 41.5 公分'		

表格 4.6 測試規劃 測試 GPIO 套件_1+GPIO 套件_2(兩個超音波感測器作用)

需求 ID	GPIO		
需求名稱	測試 GPIO 套件_1+GPIO 套件_2(兩個超音波感測器作用)		
測試案例 ID	GPIO-4		
測試情節名稱	GPIO 套件_1+2		
測試情節	步驟	操作說明	預期結果
	1	Raspberry Pi 連接感測器 TRIGGER_PIN = 25 ECHO_PIN = 8 TRIGGER_PIN = 24 ECHO_PIN = 10	線與孔位置正確
	2	程式碼(備註)測試	套件都可使用
	3	程式執行	成功
	4	終端機收到傳回距離	成功
測試指引	感測器可正常運作並傳回距離		
備註	sensor 1 距離為 10.5 公分 sensor 2 距離為 31.5 公分' sensor 1 距離為 41.5 公分' sensor 2 距離為 51.5 公分' sensor 1 距離為 10.5 公分 sensor 2 距離為 31.5 公分' sensor 1 距離為 41.5 公分' sensor 2 距離為 51.5 公分'		

表格 4.7 測試規劃 Sonic-Pi 測試

需求 ID	SonicPi		
需求名稱	Sonic-Pi 測試		
測試案例 ID	SonicPi-1		
測試情節名稱	Sonic-Pi 測試		
測試情節	步驟	操作說明	預期結果
	1	下載檔案	下載成功
	2	安裝檔案	安裝完成
	3	執行 SonicPi	成功啟動
	4	程式碼輸入 play(60)	聽到 D0
測試指引	聽到 D0 的音		
備註	Python 使用到 SonicPi 時 SonicPi 須開啟		

表格 4.8 測試規劃 GPIO-4 + SonicPi-1

需求 ID	End		
需求名稱	GPIO-4 + SonicPi-1		
測試案例 ID	End-1		
測試情節名稱	最終測試		
測試情節	步驟	操作說明	預期結果
	1	執行程式碼	可看到距離加上聽到聲音
測試指引			
備註	sensor 1 距離為 10.5 公分 sensor 2 距離為 31.5 公分' sensor 1 距離為 41.5 公分' sensor 2 距離為 51.5 公分' sensor 1 距離為 10.5 公分 sensor 2 距離為 31.5 公分' sensor 1 距離為 41.5 公分' sensor 2 距離為 51.5 公分'		

二、 測試結果

表格 4.9 測試結果 Raspberry Pi 作業系統安裝與 Python 版本確認

需求 ID	Raspberry				
需求名稱	Raspberry Pi 作業系統安裝與 Python 版本確認				
測試案例 ID	Raspberry1-1				
測試情節名稱	作業系統安裝				
情節內容	步驟	操作說明	預期結果	是否通過	說明
	1	用 SDFormatter 把 OS 丟到 SD 卡	成功在 SD 資料夾發現檔案	是	正常
	2	SD 卡插入 Raspberry Pi 並進行 OS 安裝	Raspberry Pi 可讀取 SD 卡且執行	是	正常
	3	基本設定完成	可以正常使用 Raspberry Pi OS	是	正常
	4	檢查 Python 版本是否可用	希望為 Python3.6	是	正常
測試結果	結果：通過				
錯誤補救					
備註					

表格 4.10 測試結果 Python 3.6 環境建置

需求 ID	Raspberry				
需求名稱	Python 3.6 環境建置				
測試案例 ID	Raspberry1-2				
測試情節名稱	環境安裝				
情節內容	步驟	操作說明	預期結果	是否通過	說明
	1	下載 miniconda 安裝程式(用來安裝 Python3.6)	安裝成功	是	正常
	2	執行安裝程式設定 Path	設定成功	是	正常
	3	conda install python=3.6	沒有錯誤反應	是	正常
	4	確認 Python 版本	Python3.6	是	正常
測試結果	結果：通過				
錯誤補救					
備註					

表格 4.11 測試結果

需求 ID	GPIO				
需求名稱	測試 GPIO 套件_1(單個超音波感測器作用)				
測試案例 ID	GPIO-1				
測試情節名稱	GPIO 套件_1				
情節內容	步驟	操作說明	預期結果	是否通過	說明
	1	Raspberry Pi 連接感測器 TRIGGER_PIN = 25 ECHO_PIN = 8	線與孔位置正確	是	正常
	2	程式碼測試	套件都可使用	是	正常
	3	程式執行	正常執行	是	正常
	4	終端機收到傳回距離(備註)	sensor 1 距離為 10.5 公分	是	正常
結果	結果：通過				

表格 4.12 測試結果 測試 GPIO 套件_2(單個超音波感測器作用)

需求 ID	GPIO				
需求名稱	測試 GPIO 套件_2(單個超音波感測器作用)				
測試案例 ID	GPIO-2				
測試情節名稱	GPIO 套件_2				
測試情節	步驟	操作說明	預期結果	是否通過	說明
	1	Raspberry Pi 連接感測器 TRIGGER_PIN = 24 ECHO_PIN = 10	線與孔位置正確	是	正常
	2	程式碼測試	套件都可使用	是	正常
	3	程式執行	正常執行	是	正常
	4	終端機收到傳回距離(備註)	sensor 2 距離為 10.5 公分	是	正常
測試結果	結果：通過				
錯誤補救					
備註					

表格 4.13 測試結果 測試 GPIO 套件_1+GPIO 套件_2(錯誤)

需求 ID	GPIO				
需求名稱	測試 GPIO 套件_1+GPIO 套件_2(錯誤)				
測試案例 ID	GPIO-3				
測試情節名稱	GPIO 套件_1+2				
測試情節	步驟	操作說明	預期結果	是否通過	說明
	1	Raspberry Pi 連接感測器 TRIGGER_PIN = 25 ECHO_PIN = 8 TRIGGER_PIN = 24 ECHO_PIN = 10	線與孔位置正確	是	正常
	2	程式碼測試	套件都可使用	是	正常
	3	程式執行	正常執行	是	正常
	4	終端機收到傳回距離(備註)	sensor 1 距離為 10.5 公分 sensor 2 距離為 10.5 公分	否	只有一個 sensor 有距離
測試結果	結果：不通過				
錯誤補救	發現會在其中一個迴圈裏面重複進行，打算使用 threading 多執行緒解決				
備註					

表格 4.14 測試結果 測試GPIO 套件_1+GPIO 套件_2(成功)

需求 ID	GPIO				
需求名稱	測試 GPIO 套件_1+GPIO 套件_2(成功)				
測試案例 ID	GPIO-4				
測試情節名稱	GPIO 套件_1+2				
測試情節	步驟	操作說明	預期結果	是否通過	說明
	1	Raspberry Pi 連接感測器 TRIGGER_PIN = 25 ECHO_PIN = 8 TRIGGER_PIN = 24 ECHO_PIN = 10	線與孔位置正確	是	正常
	2	程式碼測試	套件都可使用	是	正常
	3	程式執行	正常執行	是	正常
	4	終端機收到傳回距離(備註)	sensor 1 距離為 10.5 公分 sensor 2 距離為 10.5 公分	是	正常
測試結果	結果：通過				
錯誤補救					
備註					

表格 4.15 測試結果 Sonic-Pi 測試

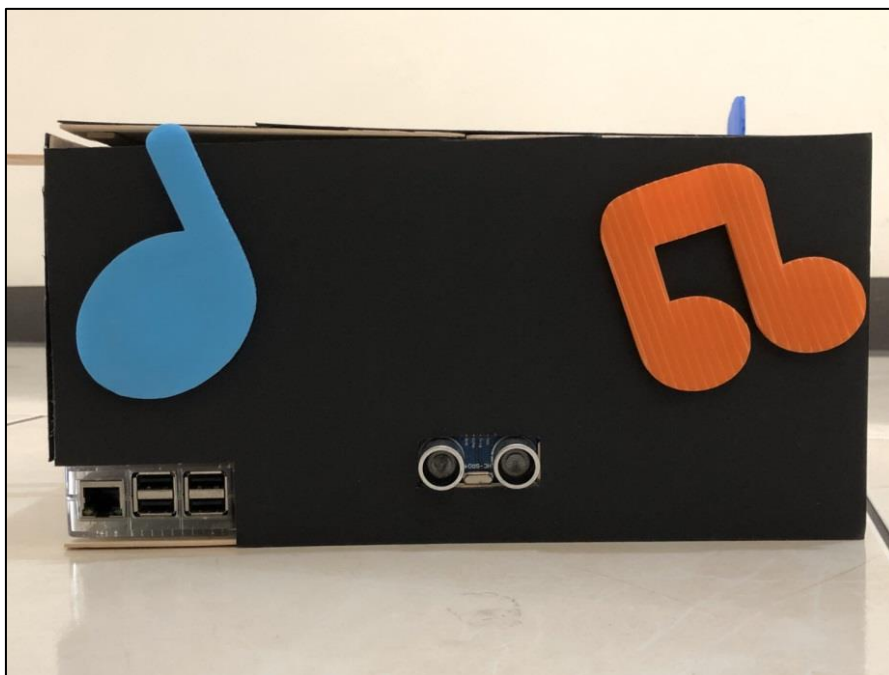
需求 ID	SonicPi				
需求名稱	Sonic-Pi 測試				
測試案例 ID	SonicPi-1				
測試情節名稱	Sonic-Pi 測試				
測試情節	步驟	操作說明	預期結果	是否通過	說明
	1	下載檔案	下載成功	是	正常
	2	安裝檔案	安裝完成	是	正常
	3	執行 SonicPi	成功啟動	是	正常
	4	程式碼輸入 play(60)	聽到 D0	是	正常
測試結果	結果：通過				
錯誤補救					
備註					

表格 4.16 測試結果 GPIO-4 + SonicPi-1

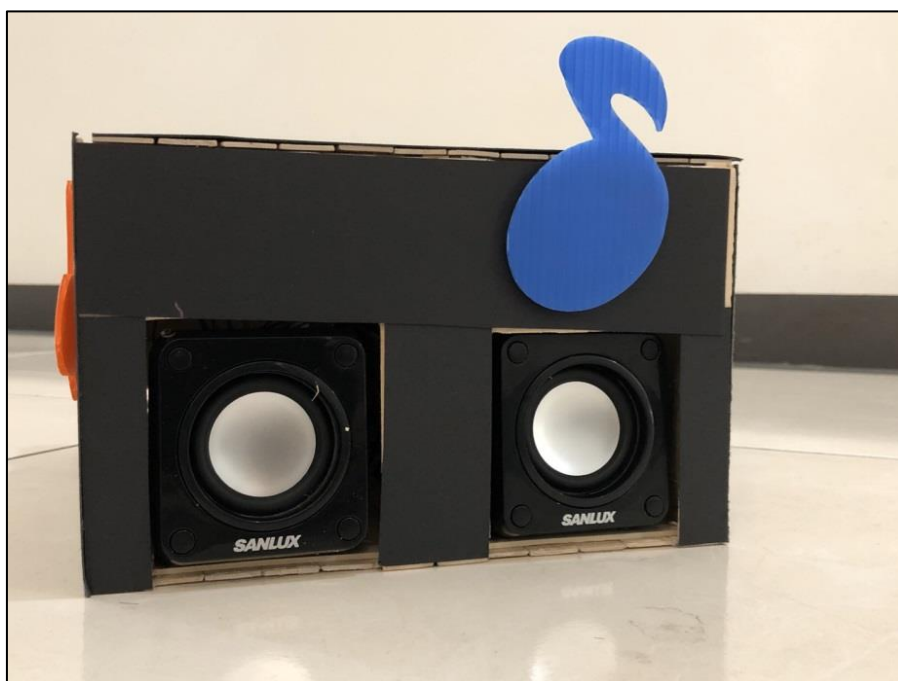
需求 ID	End				
需求名稱	GPIO-4 + SonicPi-1				
測試案例 ID	End-1				
測試情節名稱	最終測試				
測試情節	步驟	操作說明	預期結果	是否通過	說明
	1	執行程式碼	可看到距離加上聽到聲音	是	正常
測試結果	結果：通過				
錯誤補救					
備註					

伍、 研究成果

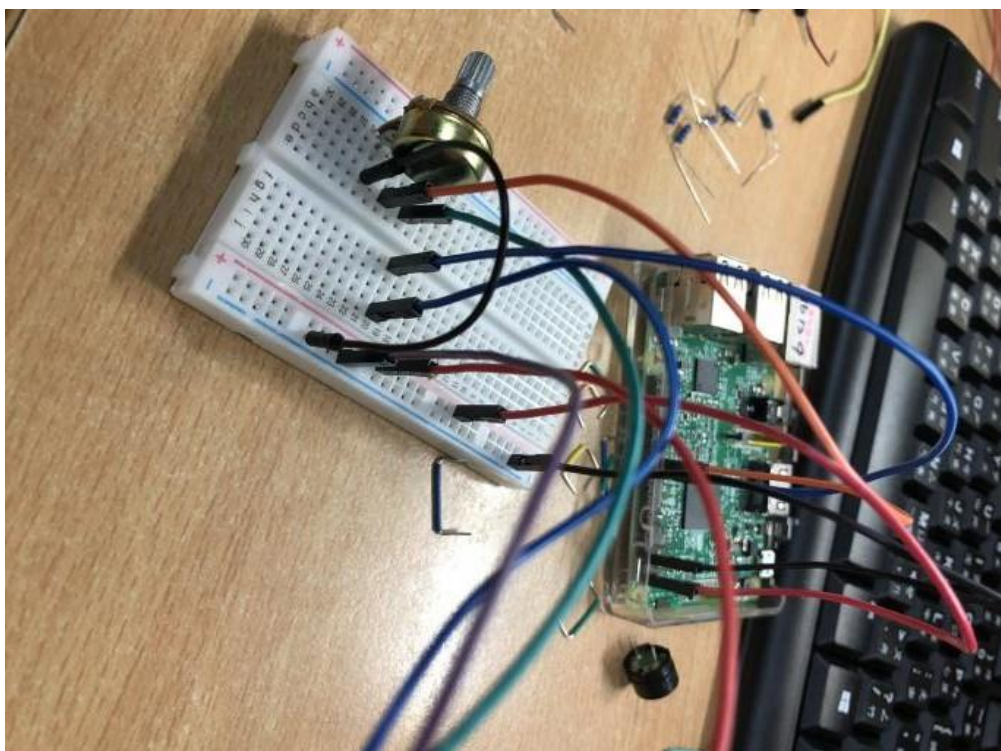
一、 成品展示



圖片 5.1 作品圖(1)



圖片 5.2 作品圖(2)



圖片 5.3 作品圖(3)



圖片 5.4 作品圖(4)

陸、 系統測試結論與未來研究方向

一、 技術方面

這次開發物聯相關技術，由於沒有學過電路學、Python，所以花了很多時間在學習研究麵包板的接法。在程式開發方面，先前有學習過 C# 以及 Java 所以遇到的困難經由小組討論以及與指導教授討論，都得以解決。系統上我們使用 raspberry pi 進行開發，由 raspberry pi 系統到麵包板上的電路還有 python 中的各種程式碼。雖然我們剛開始有畫出系統的配置圖，但當開始操作後，發現需要的東西不像剛開始想的那麼簡單，例如在麵包板上就是一個大難題，對沒有相關電子知識的我們，只能到處找人學習，安裝套件也不停的失敗，同時擔心在通電後他會不會直接冒煙，每次的啟動可以說是心驚膽顫，但還好最後成功的讓所有感測器連接成功，也能正常運作；在系統開發中也是，以為只要把程式丟到 python 中就可以動，但事情沒有那麼簡單，需要裝許多的套件，才能讓想動的東西動起來，在裝套件的路上，也遇到了許許多多的失敗。

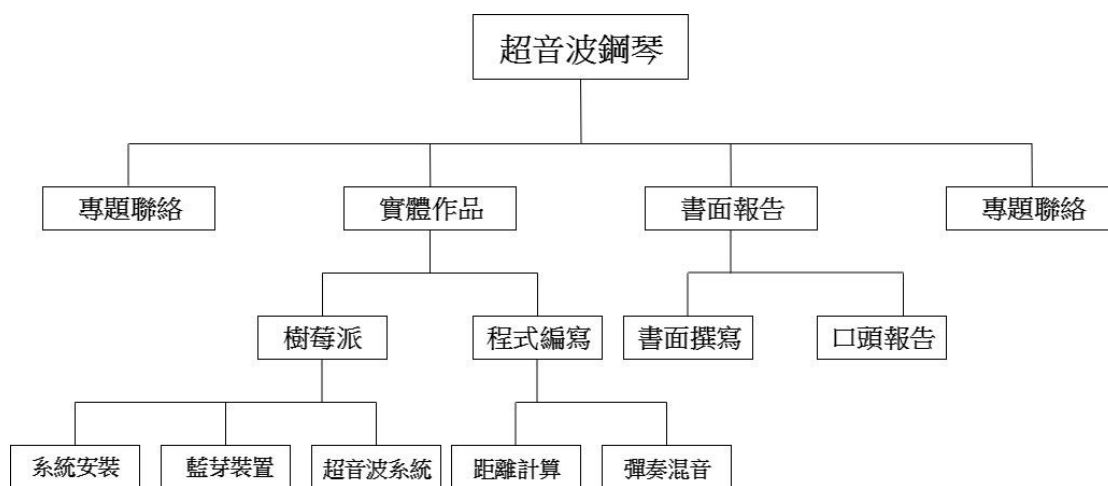
在軟體套件方面，我們所使用的是 Sonic-pi 音樂套件，只需要少少的程式碼，就可以發出聲音，不需要額外準備音檔。

二、 團隊合作方面

本專題使我們整個團隊 4 人一起創作的專題，整合大家的意見花費的時間不多，我們 4 人的想法相近，默契十足，在實作過程中，雖然碰上了難關，但大家還是協力去克服，大家各司其職，4 人當中只要有進度落後的地方，大家也會去幫助並完成工作。每週我們都會一起討論遇到的問題，而且讓每個團隊成員都能發表自己的問題與意見，並與指導教授討論方向、應用、進度以及程式方面問題，這樣子維持穩健的專案進度並持續至專題完成！

柒、 工作分解

一、 工作分解



圖片 7.1 工作分解圖

表格 7.1 工作分解 專題聯絡

工作項目名稱	專題聯絡
編號(ID)	RE1-1
工作內容描述	與指導教授討論，決定專題方向與題目並蒐集相關資料
完成項目	1. 與指導教授討論 2. 確定題目 3. 蒐集相關資料
交付項目 (程式、文件或資料)	無
投入人數	4 人
工期	1 週
工作量	4 人週

表格 7.2 工作分解 系統安裝

工作項目名稱	系統安裝
編號(ID)	SA3-1
工作內容描述	安裝樹莓派作業系統與程式軟體
完成項目	1. 安裝作業系統 2. 安裝程式軟體
交付項目 (程式、文件或資料)	1. 順利執行作業系統
投入人數	1 人
工期	1 週
工作量	1 人天

表格 7.3 工作分解 連接藍芽

工作項目名稱	連接藍芽
編號(ID)	SA3-2
工作內容描述	安裝藍芽裝置及準備相關周邊設備。
完成項目	1. 樹莓派設定並連結藍芽 2. 準備藍芽音響
交付項目 (程式、文件或資料)	1. 試行已安裝之藍芽傳輸裝置正常運作
投入人數	2 人
工期	1 週
工作量	2 人週

表格 7.4 工作分解 超音波系統

工作項目名稱	超音波系統
編號(ID)	SA3-3
工作內容描述	安裝兩台超音波裝置，並測試設備是否有相互干擾或訊號不正常之情況。
完成項目	1. 安裝超音波裝置 2. 無干擾狀況發生 3. 訊號傳輸正常
交付項目 (程式、文件或資料)	1. 超音波裝置經測試無干擾狀況發生 2. 訊號傳輸正常
投入人數	2 人
工期	2 週
工作量	4 人週

表格 7.5 工作分解 距離計算

工作項目名稱	距離計算
編號(ID)	SA3-4
工作內容描述	設定兩台超音波的距離感測設定，不同距離發出不同聲調，且確定感應範圍、高度能夠輕鬆方便使用。
完成項目	1. 設定距離 2. 設定各個距離發出不同的音調 3. 感應範圍調整至易於操作不會太過狹小
交付項目 (程式、文件或資料)	1. 超音波距離運算程式
投入人數	3 人
工期	3 週
工作量	9 人週

表格 7.6 工作分解 彈奏混音

工作項目名稱	彈奏混音
編號(ID)	SA3-5
工作內容描述	兩台超音波裝置感測後運算出其音調，兩種聲調要 可以混音、合奏
完成項目	1. 兩個聲音可同時發出 2. 兩種聲調能夠混音
交付項目 (程式、文件或資料)	1. 混合音調程式
投入人數	3 人
工期	4 週
工作量	12 人週

表格 7.7 工作分解 書面撰寫

工作項目名稱	書面撰寫
編號(ID)	BU2-1
工作內容描述	撰寫專題需繳交之書面報告
完成項目	1. 完成整份書面資料
交付項目 (程式、文件或資料)	1. 書面文件
投入人數	2 人
工期	4 週
工作量	8 人週

表格 7.8 工作分解 系統測試

工作項目名稱	系統測試
編號(ID)	TE1-1
工作內容描述	測試專題作品是否能夠順利展出
完成項目	1. 測試程式
交付項目 (程式、文件或資料)	1. 專題作品
投入人數	2 人
工期	2 週
工作量	4 人週

表格 7.9 工作分解 口頭報告

工作項目名稱	口頭報告
編號(ID)	BU2-2
工作內容描述	準備上台介紹專題之口頭稿、說話流暢度、清晰度等
完成項目	1. 經演練上台口說後能清晰流暢地表現
交付項目	
(程式、文件或資料)	1. 演說稿
投入人數	1 人
工期	1 週

二、 任務分配表

1. 組長

負責時程規劃、工作分配、工作協調、掌握專題進度。

2. 系統分析人員

負責需求分析、系統分析與設計、測試規劃、相關文件等。

3. 程式設計人員

負責編寫程式、建立資料庫、整合測試等。

4. 測試人員

負責系統測試、系統展示與操作、系統操作說明書等。

5. 技術支援人員

負責開發環技術的調查；開發環境建置、資料庫建立

6. 文書管理人員

負責資料收集；資料輸入；程式、文件、資料的版本管理；各階段
專題報告等。

表格 7.10 任務分派表(1)

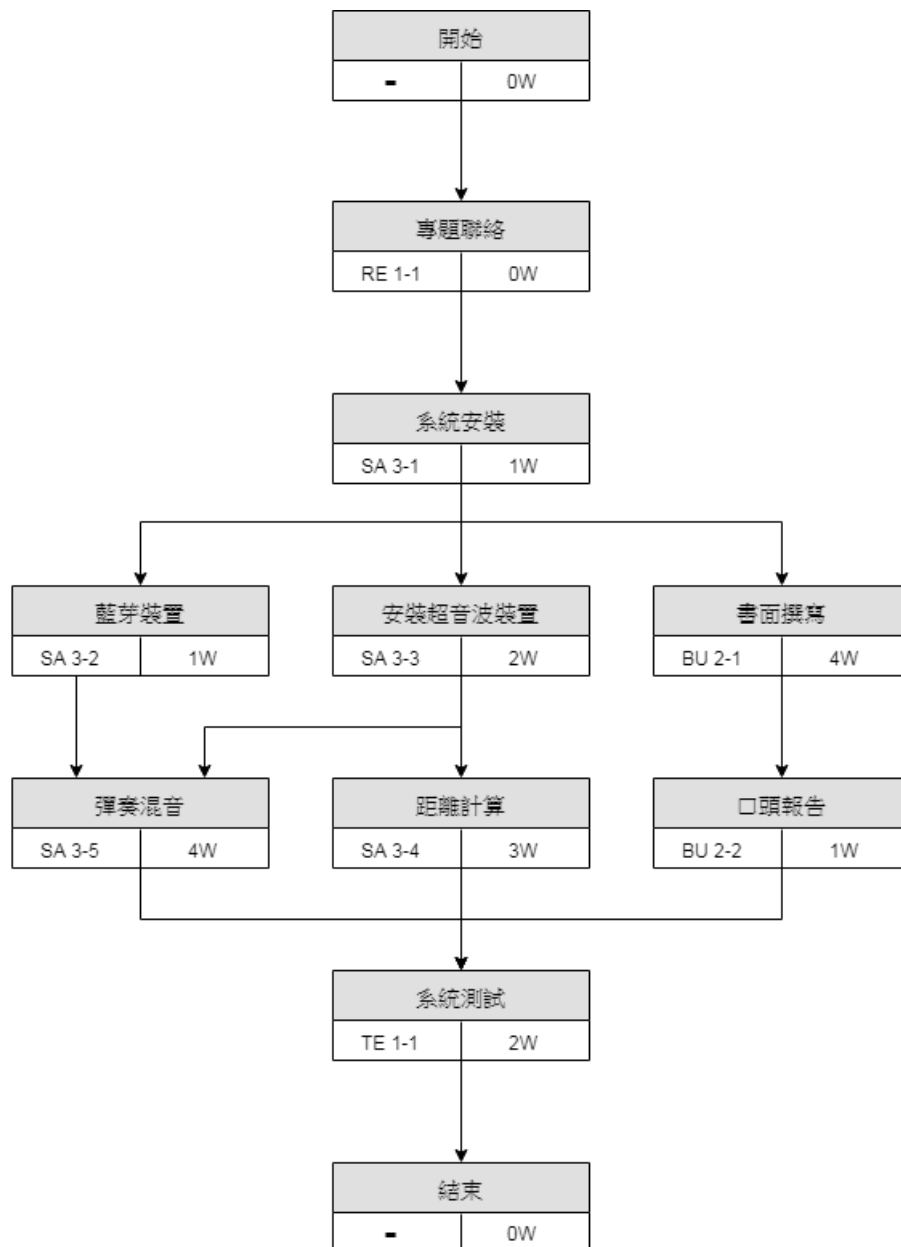
角色 \ 組員	黃智暘	楊豐嘉	高胤恒	蔡宗傑
組長				✓
系統分析人員	✓	✓		
程式設計人員	✓	✓	✓	
系統測試人員		✓	✓	✓
技術支援人員	✓		✓	
文書管理人員				✓

表格 7.11 任務分派表(2)

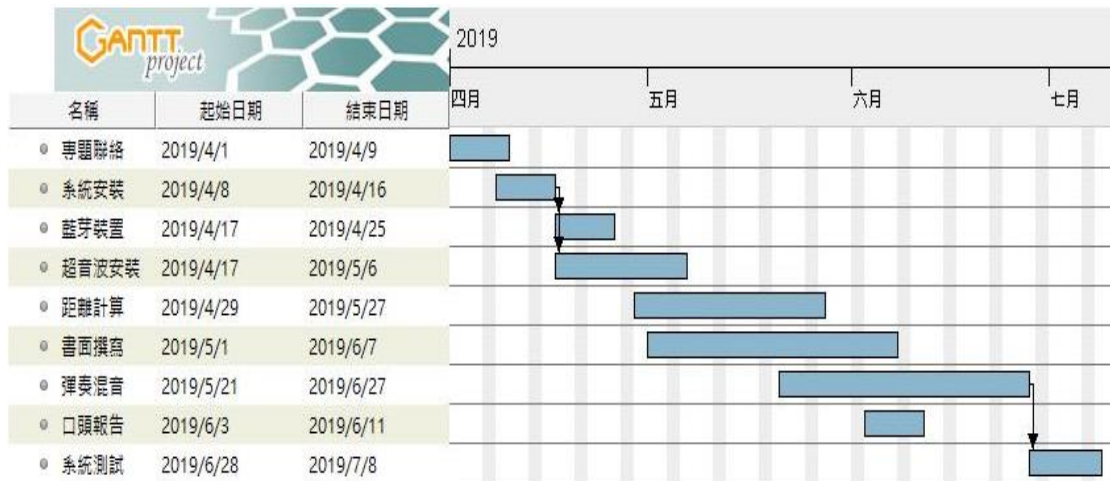
工作\組員	黃智暘	楊豐嘉	高胤恒	蔡宗傑
工作協調	✓	✓	✓	✓
需求分析	✓	✓		
功能規劃			✓	✓
流程設計	✓			✓
資料收集		✓	✓	
程式建構	✓	✓		
驗收測試			✓	✓
持續改善	✓	✓		
文件管理			✓	✓
賽事報名	✓			✓

三、 專案管理方式

本專案使用 e-mil 方式管理及控制專案進程，與小組內部進行意見交換並與指導教授討論方案可行性。

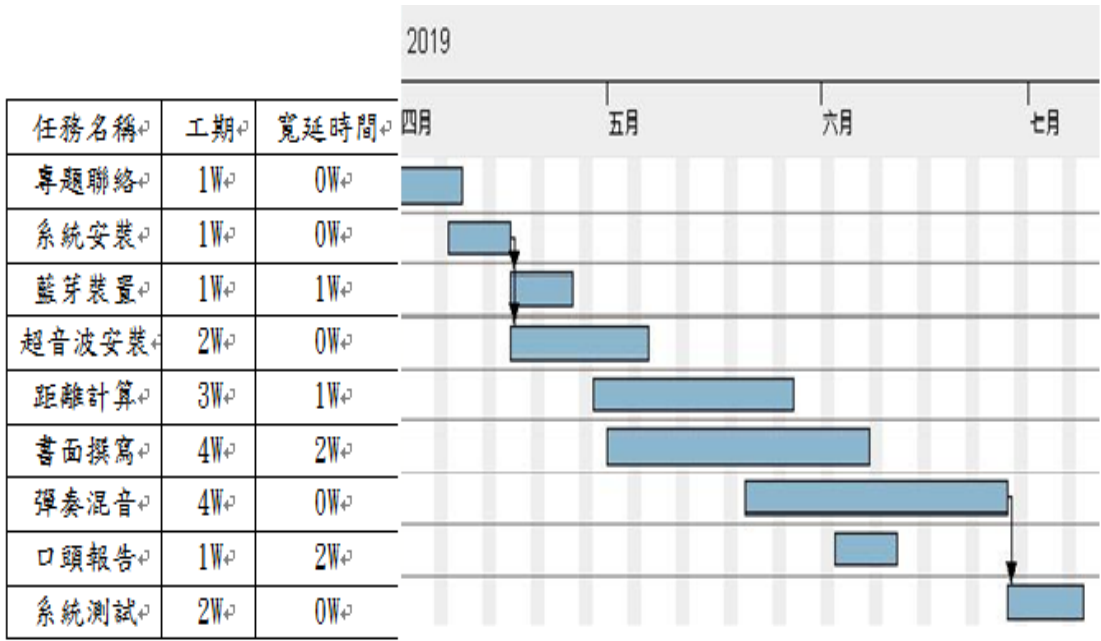


圖片 7.2 時程規劃(1)

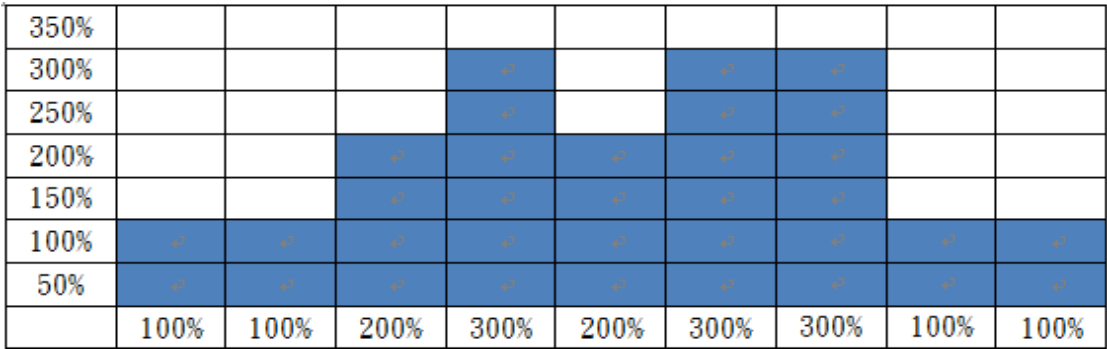


圖片 7.3 時程規劃(2)

四、 資源平滑



圖片 7.4 資源平滑圖(1)



圖片 7.5 資源平滑圖(2)

捌、 參考文獻

一、 **Raspberrypi**

(<https://www.raspberrypi.org/downloads/>)

二、 **VR 空氣鋼琴**

(https://www.bnext.com.tw/ext_rss/view/id/1941083)

三、 **Sonic pi**

(<https://sonic-pi.net/tutorial.html>)

四、 **w3schools-Python**

(<https://www.w3schools.com/python/>)

五、 **天下雜誌:全球物聯網科技先驅 Tibbo 與 UPS 攜手,搶佔 AIoT 商機**

(<https://www.cw.com.tw/article/article.action?id=5092726>)

六、 **光感應鋼琴結構,黃品瑞,公告號:M477021(2014/01/15)**

(<https://twpatsimple.tipo.gov.tw/tipotwoc/tipotwkm?@@1454871784>)

七、 **新通訊:5G 風潮推波 IoT 戰場決勝在應用**

(<https://www.2cm.com.tw/2cm/zh-tw/magazine/-CoverStory/04EADFE3A8A04144A7E031DBB8075439>)

八、 **維基百科:工業 4.0**

(<https://zh.wikipedia.org/wiki/%E5%B7%A5%E6%A5%AD4.0>)


九、 **鋼琴彈奏手部教學裝置,謝哲人、高嘉駿、廖瑾辰,公告號:I623922 (2017/05/05)**

(<https://twpat-simple.tipo.gov.tw/tipotwoc/tipotwkm?@@1323722400>)

十、 **科技網:用科技對抗新冠肺炎 台灣微軟聚焦 4 大 AI 應用場景**

(https://www.digitimes.com.tw/iot/article.asp?cat=158&cat1=20&cat2=70&id=000579531_PO54FO4U4GFKPM04AF1D7)

玖、 自傳與簡歷

基本資料		
姓名：	蔡宗傑	
性別：	男	
出生日期：	1998/05/23	
電子郵件：	4a590148@stust.edu.tw	
手機號碼：	0978928023	
自傳		
<p>我的名字是蔡宗傑，平時外向、個性沉穩邏輯清晰、做事情的方式有條理，喜歡將事情提前完成，面對問題或議題有想法和點子，這讓我在課業上的報告都能夠拿到不錯的成績，培養了我對於問題和處理事情的態度。</p> <p>在大一上擔任班上的副班代會大家服務，之後則擔任班級代表，始終秉持會大家服務的精神，讓我處事更圓滑，也能妥善處理班上同學們的問題。大一曾參加學校桌遊社，參與並了解辦理活動的流程。在校期間學業始終保持在前列，除了學業以外，自大學以來到至今，沒有翹過任何一堂課程，這是一件簡單且應當達成的事，但鮮少人有做到，全勤對我來說是對自己的事秉持認真負責的原則，而這份堅持常常反映在我做事的態度上，對每件事都要求自己盡可能地做到更好。</p> <p>升上四年級的暑假我參加了經濟部工業局的DIGI+ Talent計畫的海選面試，雖然最終沒有得到實習的機會，但也因為這次的面試機會，我接觸了很多業界裡的職員或主管，了解到許多業界發展的狀況以及需求，同時也認識到除了資訊業以外的產業資訊，對於這次的經驗我獲益良多，面對未來的挑戰也期望能夠將自己的能力更加精進！</p>		
求學歷程		
台南市立復興國中 國立新豐高級中學 南臺科技大學		
工作經歷		
南臺科技大學 資料結構 教學助理		
獲獎紀錄		
第 50 屆高級中等學校分區科學展覽 佳作		

證照技能

◇技術證照

證照名稱：中華民國技術士 - 電腦軟體應用 分數或級數：丙級

證照名稱：中華民國技術士 - 會計事務 - 人工記帳 分數或級數：丙級

證照名稱：中華民國技術士 - 網頁設計 分數或級數：丙級


證照名稱：EEC-EEAP 企業電子化助理規劃師

證照名稱：TQC 電子商務概論(專業級)


證照名稱：TQC 軟體專案管理(專業級)

證照名稱：ICDL Data Analytic 國際證照


證照名稱：日本語能力試驗 JLPT 分數或級數：N3

基本資料		
姓名：	黃智暘	
性別：	男	
出生日期：	1997/10/08	
電子郵件：	4a590003@stust.edu.tw	
手機號碼：	0952310272	
自傳		
<p>高一上學期對自己的未來毫無頭緒，成績也一直下滑，直到各學程主任到班級做學程宣導，我發現我對資訊應用學程的課程有興趣，於是高一下學期我決定進入資訊應用學程，在接觸專業科目後，我也慢慢的在專業科目中找出樂趣，成績也逐漸地進步。在這期間也慢慢的去了解哪些科系是跟我們學程相對應的，未來可以透過哪些管道進入科大就讀。高二也一直努力的考取丙級證照，也都很順利地通過每一次的證照考試，很感謝老師的辛苦指導！，高二參加了校內技藝競賽選手選拔，榮獲代表學校出去參加比賽的選手資格。也很榮幸獲得了全國優勝，利用了技優甄審順利的進入南臺科技大學就讀。</p> <p>大一上學期喜歡熱心助人的我自願自發擔任班長，為班上同學服務。大二、三也陸續接了副班長以及輔導股長，我認為世界上最美的風景就是人了，希望可以從我為中心把熱心助人的精神發揚光大。大三上學期擔任網頁程式設計「ASP.NET」的教學助理，使我對於網頁程式設計有了更深的了解。而升上四年級的那個暑假我參加了經濟部工業局的DIGI+ Talent計畫的海選面試，也很榮幸的能夠到金屬工業研究發展中心研習，過程裡學習到很多不管是專業上或者是職場的一些相關知識，都讓我獲益良多。</p>		
求學歷程		
台南市立永康國中 國立新化高級工業職業學校 南臺科技大學		
工作經歷		
南臺科技大學 交通隊 南臺科技大學 ASP.NET 網頁設計 教學助理 經濟部工業局 DIGI+Talent 研習生 單位：高雄金屬研究發展中心		
獲獎紀錄		
教育部全國高級中等學校技藝競賽 文書處理 - 優勝		
證照技能		
◇技術證照 證照名稱：中華民國技術士 - 電腦軟體應用 分數或級數：丙級		

證照名稱：中華民國技術士 - 會計事務 - 人工記帳 分數或級數：丙級
證照名稱：商教會英文檢定第四級
證照名稱：EEC-EEAP 企業電子化助理規劃師
證照名稱：TQC 電子商務概論(專業級)
證照名稱：TQC 軟體專案管理(專業級)
證照名稱：ICDL Data Analytic 國際證照

基本資料		
姓名：	楊豐嘉	
性別：	男	
出生日期：	1997/10/10	
電子郵件：	4a590045@stust.edu.tw	
手機號碼：	0905232506	
自傳		
<p>進入南臺的第一年，我發現南臺的資源相當的豐富，每一個地方都值得我們好好的運用。第一年我還不太熟悉環境所以還在摸索，對於課程上的安排我也覺得跟之前有很大的差別，但是還可以適應。</p> <p>第二年的課程比較多，也比較深入，所以讀起來有一點吃力，不過老師們都很用心地替學生解決問題，所以在二年級的時候雖然讀得很辛苦但是最後的結果是好的。第三年的課程選修居多，我的興趣偏向於程式類的所以我選的蠻多有關程式的課程，雖然讀起來特別辛苦，但是我覺得收穫滿滿，有一些比較新的程式軟體，我們也可以去嘗試，並且在上課的時候將不懂的地方去詢問老師。</p> <p>在三年級的時候我也開始去弄專題了，我們的題目是超音波鋼琴，利用樹莓派接 PIN 腳連接到超音波，這是一個新的體驗，我覺得相當的有趣，雖然過程中，遇到了很多的困難，但是在老師的幫助之下，我們的問題都順利的解決了。第四年，這一年即將畢業，我在三年級也報考了本校的研究所，在學校裡面有預研的課程，我也先上了，這對在未來我去找研究所很有幫助，我可以先學習到研究所的課程，也可以早一點體驗到研究所的生活，研究所的課程明顯的比大學的課再加深許多，讓我知道在某一些方面我比較不懂，也可以在這一年趁早補強，在未來上研究所的課程也許就不會那麼困難。</p>		
求學歷程		
嘉義市北興國中 嘉義私立高級興華中學 南臺科技大學		
社團經歷、幹部經驗		
網球社、網球校隊 體育股長、網球社財務組長		
證照技能		
◇技術證照 證照名稱：中華民國技術士 - 電腦軟體應用 分數或級數：丙級 證照名稱：中華民國技術士 - 電腦軟體應用 分數或級數：乙級		

證照名稱：中華民國技術士 - 網頁設計	分數或級數：乙級
證照名稱：中華民國技術士 - 會計事務-資訊	分數或級數：乙級
證照名稱：商教會英文檢定第四級	
證照名稱：EEC-EEAP 企業電子化助理規劃師	
證照名稱：TQC 電子商務概論(專業級)	
證照名稱：TQC 軟體專案管理(專業級)	

基本資料		
姓名：	高胤恒	
性別：	男	
出生日期：	1998/02/07	
電子郵件：	4a590127@stust.edu.tw	
手機號碼：	0976758656	
自傳		
<p>【家庭背景】</p> <p>我是高胤恒，就讀於南臺科技大學資訊管理系。當我在學習上遇到問題時，是會想辦法解決問題的人。</p> <p>出生在一個幸福小家庭，媽媽服務於工業園區，爸爸退休回家種田，弟弟還在學校求學。從小，父母以身教的方式深深影響我，待人處世要誠懇勤勞及珍惜所有。</p> <p>【學習歷程】</p> <p>求學過程中，影響我最多的是高中時期，由於學校離家較遠，選擇在外租房，讓我確實體會到一個人的生活不是那麼簡單，讓我學會了獨立。高中擔任過康樂股長和大學的總務股長，讓我了解幹部需要對自己的任務負責。像是康樂負責的暖身或總務負責的收款，讓我了解到責任的重要。</p> <p>面對同學或自己在課業上的問題，我們時常都會互相討論，或是上網找方法，如此，不但解決了課業的問題，也讓觀念也更加清楚。此外，在大學時期曾經修過 C#、CSS、HTML、JavaScript、PHP、MySQL、Android Studio、Swift 等程式語言，可以運用在網頁前端、後端或是手機應用程式。</p> <p>【專題經驗】</p> <p>大學時的畢業專題，也令我印象深刻，我們這組所做的是超音波鋼琴，是由 Raspberry Pi、Python、sense 組成的。用到的 Raspberry Pi 系統、Python 語言還有麵包版的結構，都是自學而成的，雖然比賽都沒有得名，但這是我們小組一起從無到有，從不會到會努力的成果。也很感謝一路都有老師的幫助，讓我們在無解的時候給我們一點方向，雖然自學的過程很辛苦，但做出東西時，真的很開心。</p> <p>【未來展望】</p> <p>在專題小組中，讓我體會到團隊是一種怎樣的存在，互相幫忙、一起學習、成長，也可以更有效率地完成。過程也讓我發現還有很多的不足，值得讓我學習。</p>		
求學歷程		
台南市立安南國中 國立新營高級中學 南臺科技大學		

證照技能

◇技術證照

證照名稱：EEC-EEAP 企業電子化助理規劃師

證照名稱：TQC 電子商務概論(專業級)

證照名稱：TQC 軟體專案管理(專業級)

證照名稱：ICDL Data Analytic 國際證照