

中華民國第 59 屆中小學科學展覽會 作品說明書

高級中等學校組 工程學(一)科

(鄉土)教材獎

052305

機械導覽車

學校名稱：臺北市立內湖高級工業職業學校

作者： 職二 吳柏均 職二 尤奕昕 職二 林庠葳	指導老師： 侯淇健 陳昭安
---	-----------------------------

關鍵詞：機械板車、自動化導覽、空間定位

摘要

本研究目的在製作一個可以輔助移動的機械板車，由發出與接受語音指令引導使用者到達指定之位置，接收到指令時導覽車將會自動將預先建立好的資料庫中找尋這件作品的資料以及視覺定位來到達位置。作品採用 Google 雲端服務進行語音辨識透過辨識的結果，對導覽車進行操控，甚至導航到相關位置。用 Speech-to-Text 功能播放聊天機器人回應內容。此外作品亦考慮到安全，避免行進過程中對服務對象或周圍民眾造成傷害。透過這樣的一個導覽機器車，可以讓行動不便者，自由的在展場中悠遊。

壹、研究動機

在日常生活中，我們總是享有理所當然的方便與舒適，但社會中仍有一些行動不便的族群。在服務弱勢觀眾參觀博物館需求探究的論文中提到「參觀輔助資源、多媒體數位科技輔具與導覽服務皆為參觀過程的延伸服務，可協助弱勢者獲取更多展覽資訊。」(林詠能、趙欣怡，2016)。若能藉由機械板車的功能，輔助行動不便的族群，讓他們可以在展場輕鬆欣賞藝術作品。

貳、研究目的

- 一、探討運用麥克納姆輪及直流減速編碼馬達進行機械板車的方向移動控制。
- 二、研究以 Google Speech-to-Text 的 API 進行語言辨識。
- 三、研究使用超音波建構安全及避障的機制。
- 四、探討三角定位與視覺定位對比。
- 五、研究以 Node-Red 監控使用端的使用狀況並蒐集使用者的資訊。

參、研究設備及器材

表 1. 材料表

編號	材料名稱	材料型號	數量
01	微控制器	MakeBlock ORION	1
02	語音辨識器	Grove Speech Recognizer	1
03	FULLYMAX 富力電池	11.1V 2700mAh 30C~60C	1
04	myRIO-1900	FPGA development board NI MyRIO Xilinx Zynq - 7000	1
05	樹莓派	Raspberry Pi 3 Model B	2
06	馬達晶片	MakeBlock Power Encoder Motor Driver	2
07	超音波感測器	HC-SR04	4
08	直流減速馬達	36mm DC Motor	4
09	麥克納姆輪	Mecanum Wheel 100mm	4
10	底盤和骨架	MakeBlock Beam	9

肆、研究過程或方法

一、系統規劃

(一) 導覽車系統架構：本「機械導覽車」藉由 Node-Red、Raspberry Pi、myRIO 來做為後台以及車體的系統。使用者可使用 RFID 卡認證登入，取得機械板車的服務。接下來可以運用語音指令，命令板車引導使用者到達指定位置。移動過程中，導覽車也會隨時將所在位置、服務需求等使用者資訊，透過 Node-Red 回傳給資料庫，作為統計分析使用或即時服務使用。

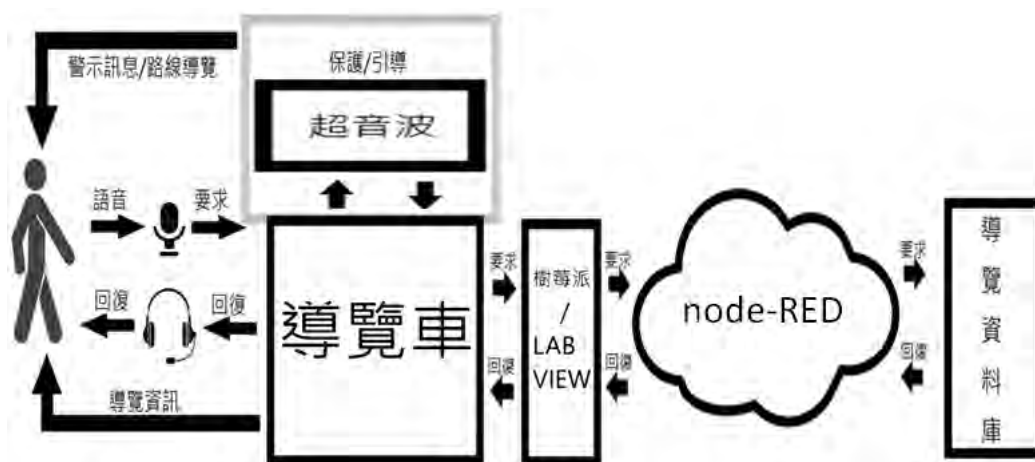


圖 1. 系統架構圖

(二) 車體移動規劃：Raspberry Pi 在接收到來自使用者的語音指示之後，myRIO 會綜合超音波感測器以及影像辨識鏡頭所得到的資訊，來計算現在位置和到達目的地的最佳路徑，在移動的路程中也會以超音波感測器來偵測任何障礙物。

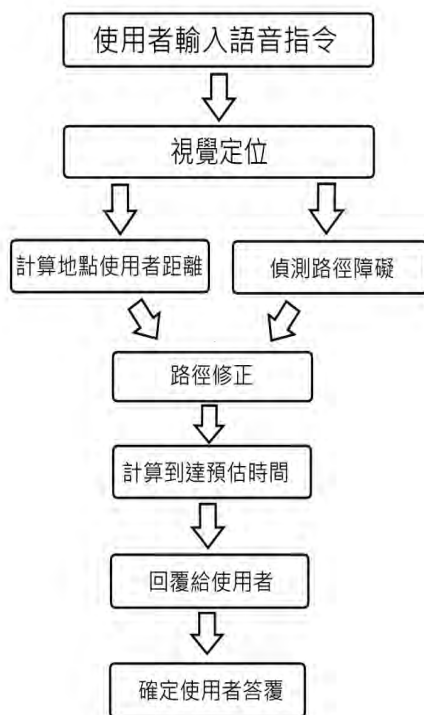


圖 2. 導覽車視覺定位及回覆流程圖

二、機械導覽車車身結構與動力原理

(一) 車身結構：車身的部分是使用了 MakeBlock 的配套式骨架，配合 Raspberry Pi 的 myRIO 主機板、馬達控制板及四組馬達與麥克納姆輪控制車子的移動。運用樹莓派的無線功能收集及傳送資訊，與遠端伺服器構成一個遠距的監控服務平台及資料庫。電池採用 FULLYMAX 鋰電池提供高性能的動力輸出，以提高載重及續航能力。



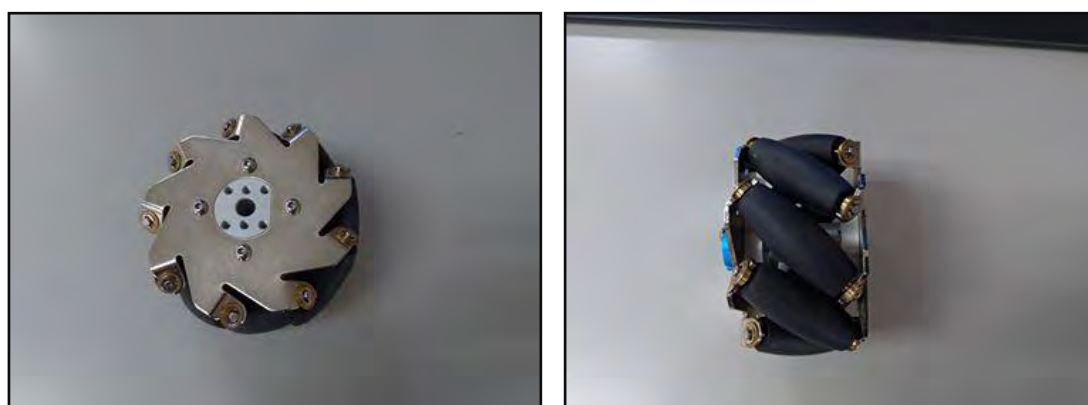
圖 3. 機器導覽車車身

(二) 直流減速編碼馬達：使用 H-Bridge 配合 PWM 信號控制馬達轉速及方向，PWM 透過改變脈波寬度使轉速改變，脈波的工作周期（Duty Cycle）越大轉速越高，反之轉速越低。馬達內置一個 1:25 的行星式減速機，可降低轉速提高扭力。另外馬達尾端安裝一組旋轉編碼器（Rotary Encoder），當馬達旋轉的時候，便能透過 A、B 通道產生的脈波數量及相位，計算馬達轉動角度、圈數及方向，進而得知車子所在的位置。



圖 4. 直流減速馬達

(三) 麥克納姆輪：目前普遍用於競賽或工業用的機器人所使用的輪子，此輪子的設計，因為輪子外圈的 45 度的小輪子，四顆輪子的組合可以因為 45 度的轉向，而搭配出任意方位的直接移動，無須轉動車體。



(a) 側視圖

(b) 正視圖

圖 5. 麥克納姆輪

(四) 剛體移動方向與麥克納姆輪轉速關係：輪子往 90° 方向轉動時每顆輪子所帶的動力方向不同，由圖推演，將剛體移動向量正投影到四個輪子所產生的向量即為四個輪子的轉向與速度。

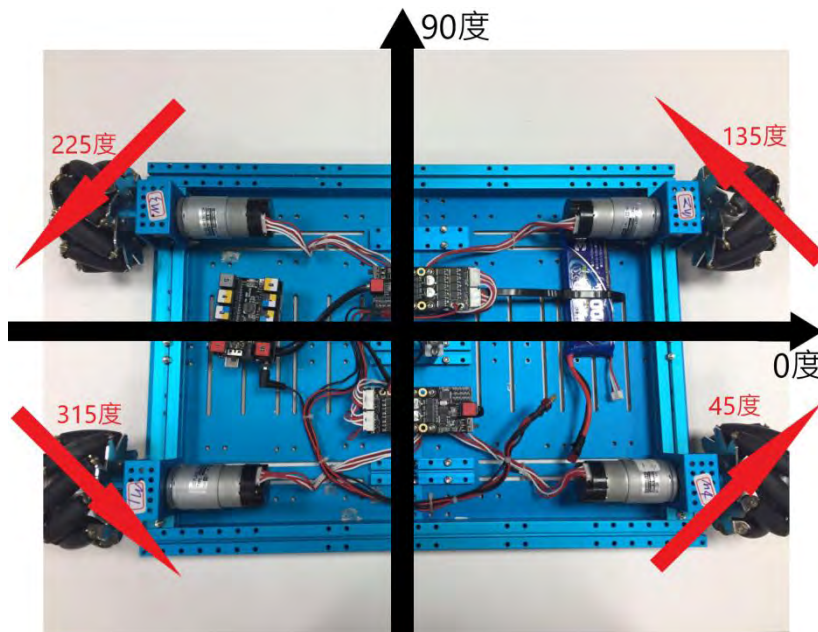


圖 6. 剛體定義方向與四顆麥克納姆輪的方向關係（以 90° 為移動方向）

(五) 輪子轉速公式推導： \overrightarrow{OA} 為車子的主移動方向， $\pm\overrightarrow{OC}$ 為輪子移動方向速度，藉由下列公式推算出各個輪子方向的速度。

$$|\overrightarrow{OA}| \times \cos(\theta_{\text{輪子角度}} - \theta_{\text{車體移動方向}}) = \text{輪子方向速度 cm/s}$$

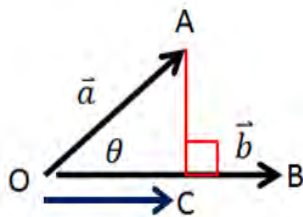


圖 7. 移動向量 \overrightarrow{OA} 正投影在輪子的投影量

表 2. 理論推導（以向前 90°，10cm/s 為例）

麥克納姆輪	輪子速度
A 輪	$10 \times \cos(45^\circ - 90^\circ) = 5\sqrt{2}$
B 輪	$10 \times \cos(225^\circ - 90^\circ) = -5\sqrt{2}$
C 輪	$10 \times \cos(135^\circ - 90^\circ) = 5\sqrt{2}$
D 輪	$10 \times \cos(-45^\circ - 90^\circ) = -5\sqrt{2}$

三、Google Speech-to-Text API 語言辨識

(一) 利用 Google Speech-to-Text API 辨識語音：將 PyAudio 錄出來的語音利用 Python 丟上 Google Speech-to-Text API 進行辨識，Google 會將辨識完的文字與自信值丟回 Python，如此一來就可以再利用語音聊天機器人將使用者講的話來進行回答。

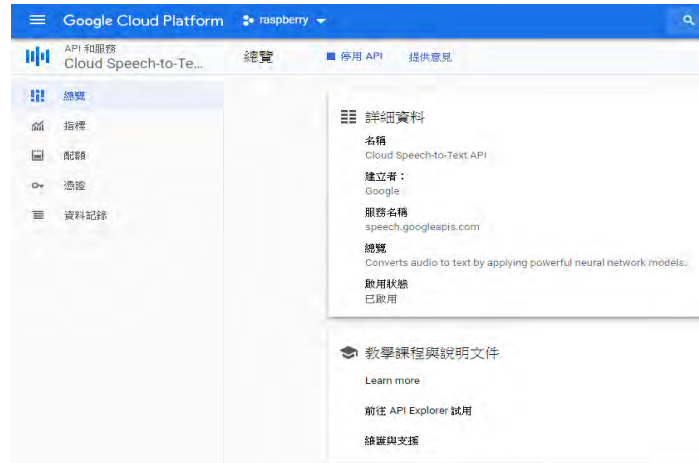


圖 8. 利用 Google Cloud Platform 的 Speech-to-Text API 辨識語音

```
import io
import os

# Imports the Google Cloud client library
from google.cloud import speech
from google.cloud.speech import enums
from google.cloud.speech import types

# Instantiates a client
client = speech.SpeechClient()

# The name of the audio file to transcribe
file_name = os.path.join(
    os.path.dirname(__file__),
    'voice.wav')

# Loads the audio into memory
with io.open(file_name, 'rb') as audio_file:
    content = audio_file.read()
    audio = types.RecognitionAudio(content=content)

config = types.RecognitionConfig(
    encoding=enums.RecognitionConfig.AudioEncoding.LINEAR16,
    sample_rate_hertz=16000,
    language_code='cmn-Hans-CN')

# Detects speech in the audio file
response = client.recognize(config, audio)

for result in response.results:
    print('Transcript: {}'.format(result.alternatives[0].transcript))
    print('Confidence: {}'.format(result.alternatives[0].confidence))
```

圖 9. Speech-to-Text API 範例程式 (Python)

(二) 聊天機器人：利用 Python 內的外掛模組 AIML 來製作聊天機器人，當使用者說出的話經過辨識完，丟入 AIML 利用自己創造的中樞回答大腦進行辨識，使用者可以提出反應，如果有應該要辨識卻沒辦法辨識的輸入值，可以進行重製語音模型輸入，增加語音辨識的靈活性。

```
<aiml version="1.0.1" encoding="UTF-8">
  <!-- std-startup.xml -->

  <!-- Category is an atomic AIML unit -->
  <category>

    <!-- Pattern to match in user input -->
    <!-- If user enters "LOAD AIML B" -->
    <pattern>LOAD AIML B</pattern>

    <!-- Template is the response to the pattern -->
    <!-- This learn an aiml file -->
    <template>
      <learn>basic_chat.aiml</learn>
      <!-- You can add more aiml files here -->
      <!--<learn>more_aiml.aiml</learn>-->
    </template>
  </category>
</aiml>
```

圖 10. 中樞大腦寫入

```
<aiml version="1.0.1" encoding="UTF-8">
  <!-- basic_chat.aiml -->

  <category>
    <pattern>HELLO</pattern>
    <template>
      Well, hello!
    </template>
  </category>

  <category>
    <pattern>WHAT ARE YOU</pattern>
    <template>
      I'm a bot, silly!
    </template>
  </category>
</aiml>
```

圖 11. 大腦回答資料庫範例

(三) 語音合成：利用 Python 外掛模組 pyttsx 來進行語音合成，將 AIML 的回答結果丟入語音機器人裡面讀出。

```
# coding:utf-8
import sys

sys.setdefaultencoding('utf8')

import pyttsx
engine = pyttsx.init()
engine.say('hello world')
engine.say('你好嗎')
engine.runAndWait()
```

圖 12. 語音合成範例

四、超音波安全避障

在展場中參觀的民眾及各種可能臨時或長期擺設的物品，可能會出現在預先規劃的路線中，車子行進時遇到這些狀況必須即時的停下，以避免撞到民眾或展品的危險。而另外一方面車子上的參觀人，也不可以因為煞車的重心偏移而造成跌倒的情況。本研究使用超音波配合 PID 控制動力來達成，設計說明如下：

(一) 超音波測距理論：公式中 V 為超音波在空氣中傳播速度； T 為環境溫度； S 為被測距離； t 為發射超聲脈沖與接收其回波的時間差； t_1 為超聲回波接收時刻； t_0 為超聲脈沖發射時刻。利用 MCU 的捕獲功能可以很方便地測量 t_0 時刻和 t_1 時刻，根據以上公式，用軟體編程即可得到被測距離 S 。

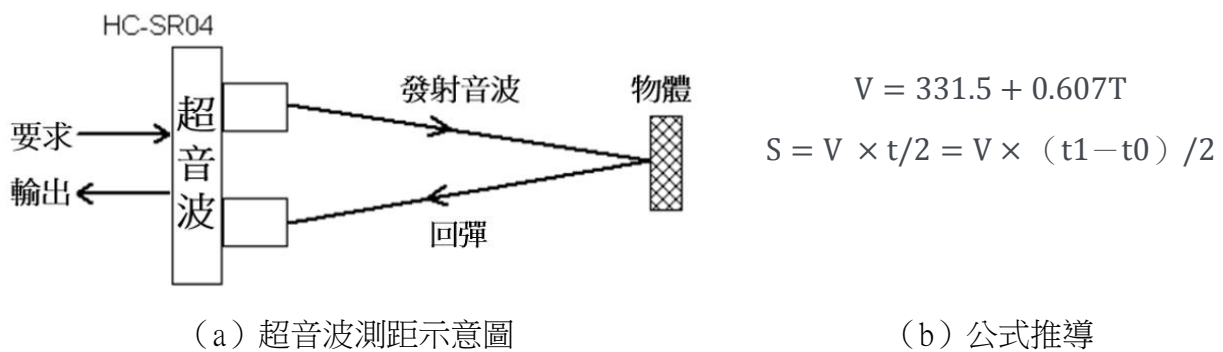


圖 13. 超音波測量距離理論 (引用網路文章)

(二) PID 減速設計：當車子遇到障礙物時會進行減速並迴避，但減速太快會造成使用者發生危險，故需要設計一個減速系統來保障使用者的安全。我們參考 PID 控制理論完成這個減速系統。

P：比例控制器，利用超音波的數值控制放大倍數使系統反應加速。

$$\left\{ \begin{array}{l} \text{距離} > 100\text{cm}, V = V_{\text{max}} \\ \text{距離} < 100\text{cm} \text{ 且 } \text{距離} \geq 50\text{cm}, V_{\text{max}} \times \frac{\text{距離} - 50\text{cm}}{50\text{cm}} \\ \text{距離} < 50\text{cm}, V = 0 \end{array} \right.$$

I：積分控制器，消除穩態誤差使系統更精準，本系統需求度不高故不使用。

D：微分控制器，改善系統穩定度使系統穩定工作消除震盪或是發散現象。藉由每個單位時間對超音波距離的變化量，回授給方向向量。

圖 14 由比例單元 (P)、積分單元 (I) 和微分單元 (D) 組成。可以透過調整這三個單元的增益 K_p , K_i 和 K_d 來調定其特性。PID 控制器主要適用於基本上線性，且動態特性不隨時間變化的系統。利用這套系統可以做到穩定的等加速度運動和等減速度移動，讓使用者搭乘導覽車時不因為急停或突然開動造成危險發生。

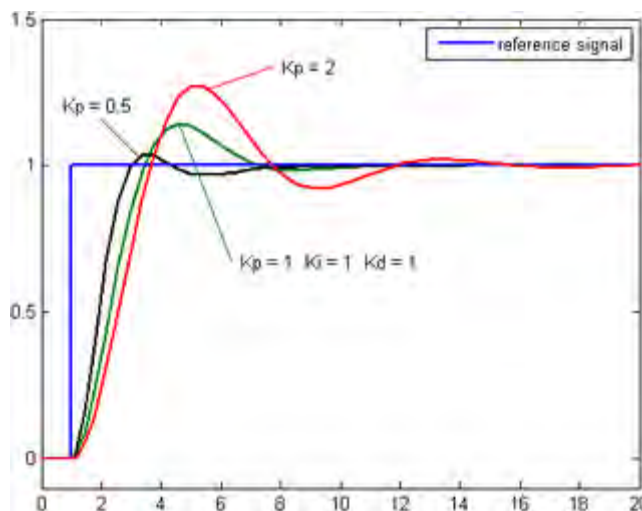
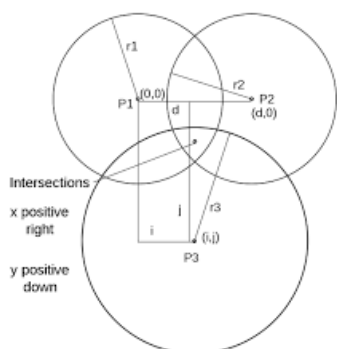


圖 14. PID 系統運統曲線圖 (引用網路文章)

五、空間定位

了解車子在空間中的座標，對於一台可以自主移動的車子是很重要的。因此必須尋求有效率而且可靠的定位法。本研究將進行樹莓派本身的藍芽裝置進行三角定位及運用本裝置的攝影機，進行的視覺定位進行研究及比較，說明如下：

- (一) 藍芽三角定位：圖中假設 P1, P2, P3 點為樹莓派的藍芽接收站，接收導覽車持續發射的藍芽訊號，根據接收站收到的 RSSI (Received Signal Strength Indicator) 強度來推算接收站和訊號原的距離，得到導覽車與 P1, P2, P3 之間的距離 r_1, r_2, r_3 ，利用這三個距離為半徑畫圓，三圓交點處即為導覽車的位置。但由於數值會有誤差，三圓可能不相交，所以要去計算三圓面積交疊處的中點，才是導覽車的正確位置。



(a) 三角定位示意圖

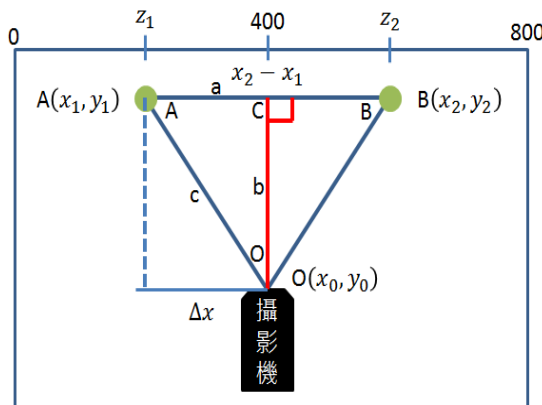
$$\begin{aligned}
 r_1^2 &= x^2 + y^2 + z^2 \\
 r_2^2 &= (x-d)^2 + y^2 + z^2 \\
 r_3^2 &= (x-i)^2 + (y-j)^2 + z^2
 \end{aligned}$$

$$\begin{aligned}
 x &= (r_1^2 - r_2^2 + d^2) / 2d \\
 y &= (r_1^2 - r_3^2 - x^2 + (x-i)^2 + j^2) / 2j
 \end{aligned}$$

(b) 公式推導

圖 15. 三角定位理論 (引用網路文章)

(二) 視覺定位：採用尺度不變特徵轉換 (Scale-invariant feature transform, SIFT) 技術進行視覺定位，這個方法主要是偵測影像中的局部特徵點，並藉由特徵點與攝影機之間的三角關係，可得到位置、尺度、旋轉的數據。下圖的兩綠點為物體特徵點，設置的特徵點為已知的座標，當攝影機拍攝到這兩個座標點後，依照相片中兩點的距離及座標，推算攝影機本身的座標，若產生障礙物，亦可推算離障礙物的距離。



$$\frac{400 - x_1}{400} \times 55^\circ = \theta_{\text{物體與攝影機拍攝夾角}}$$

$$\overline{AC} = a = 400 - z_1$$

$$\overline{BC} = z_2 - 400$$

$$c = \frac{\overline{AC}}{\sin \theta_{\text{物體與攝影機拍攝夾角}}}$$

$$b^2 = \sqrt{c^2 - a^2}$$

$$\Delta x = (x_2 - x_1) \times \frac{\overline{AC}}{\overline{AC} + \overline{BC}}$$

$$x_0 = \Delta x + x_1$$

$$y_0 = y_1 - \overline{OC}$$

(a) 用特定的特徵點 (綠圓圈代表) 來定位。(b) 透過上述的公式計算車體座標。

圖 16. 採用 SIFT 的視覺定位

(三) 藍芽三角定位與視覺定位比較：以空間限制和定位精準度比較結果，則使用 SIFT 的影像辨識作為三角定位的使用方法較為適合。

表 3. 定位比較方式

比較項目 \ 定位方式	藍芽三角定位	SIFT
定位精準度	公分級 (5-10cm)	公分級 (1-2cm)
設備	至少 3 個藍芽基地站	1 個攝影機
定位架設與準備時間	開機即可使用	需花時間計算特徵點
展場大小	受藍芽傳輸功率影響, 約 300 平方公尺 (以 class 2 為標準)	無限制
定位方式限制	基地台與定位物件間有障礙物時會使定位精度下降。	若人潮過多, 會造成特徵點過少
成本	大約 1800 元	大約 3500 元

六、Node-Red

Node-Red 是由 IBM Emerging Technology 所開發的系統，採用 Node.js 開發的開發環境及節點套件，可以針對節點進行拉線的方式進行程式編寫的視覺化套件，除了本身的服務外，也可以透過擴充增加功能，或其他擴充 IBM Bluemix 的雲端服務 API。

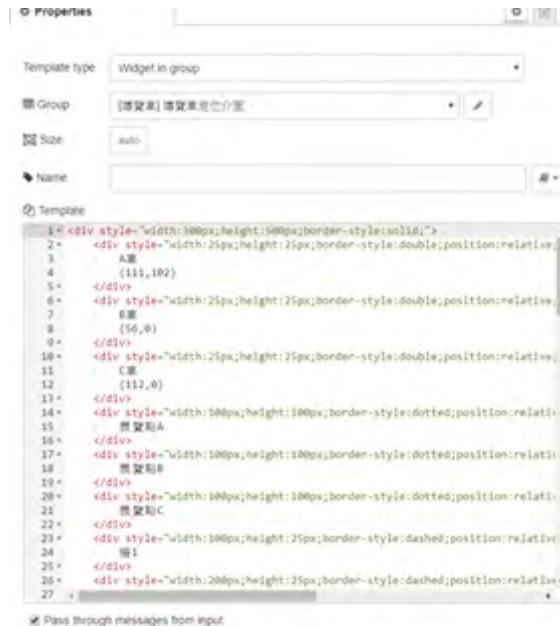


圖 17. Node-Red 網頁式的開發介面

- (一) 以 Node-Red 監控使用端的使用狀況：Node-RED 伺服器會利用 MQTT 接收機械導覽車和使用者的資訊，顯示給後端管理員看，而使用者會看到使用者介面，當使用者發生什麼事情或想要知道更多資訊都可以透過 Node-RED 來傳送需求給導覽車或後端管理員。

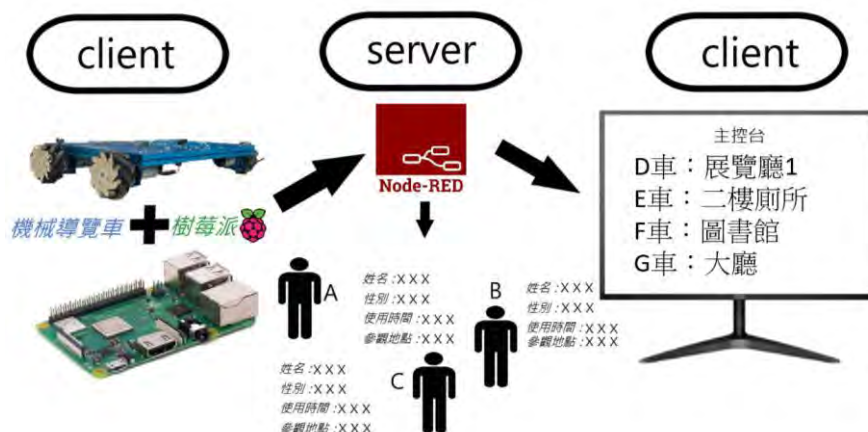
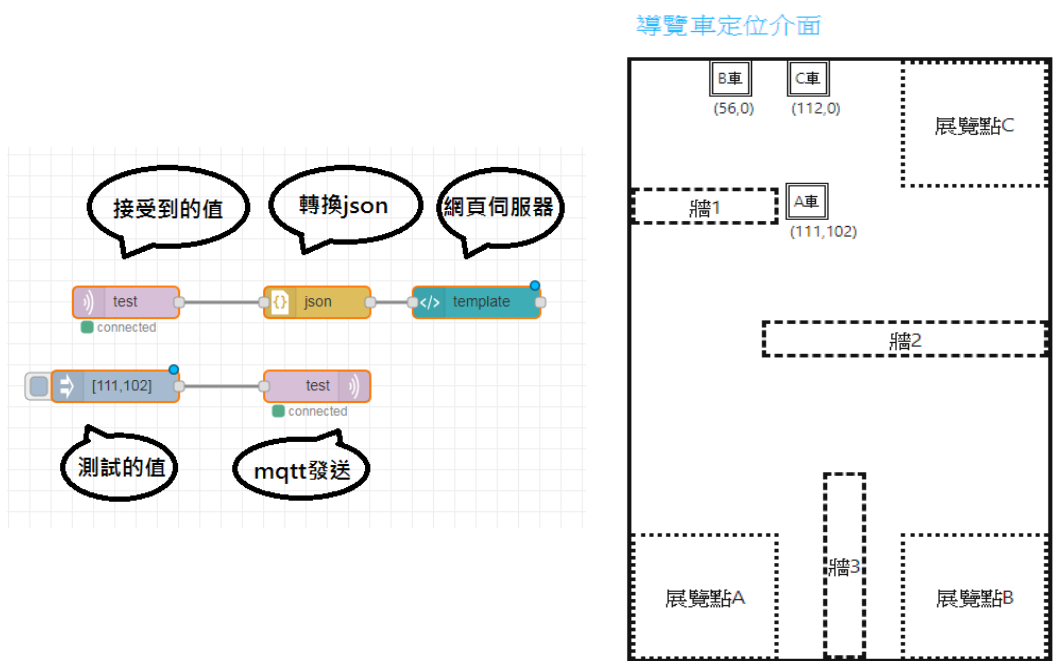


圖 18. MQTT 的資訊傳輸概念圖

(二) Node-Red 伺服器：當車子移動時，透過視覺定位取得車體的 (x, y) 座標，在利用 MQTT 傳送到伺服器端，伺服器接收到 x 和 y 值時候就會把車子的位子顯示在網頁介面上，以圖形化的方式顯示車子目前的位置。下圖粉紅色的節點為 MQTT 的發射端（下）及接受端（上），依下列的範例 MQTT 的 Topic 為 test。灰色的節點的功能主要模擬數值的輸出，假設車子傳出的座標為 (111,102)，當按下按鈕後這個數值會送出，透過 json 的節點（橘色）轉換成 json 格式，接下來在用 template 的節點（湛藍色）顯示座標位置。



(a) 使用 MQTT 傳送座標值的測試程式 (b) 使用圖形化介面顯示座標位置

圖 19. 採用 MQTT 發送座標值的範例程式

伍、研究結果

一、機械導覽車運作結果

(一) 機械板車移動數值：在相同的速率下，以不同移動方向的角度進行測出各項目的秒數和距離誤差值。

表 4. 測量數值（移動方向角度改變，其他參數不變）

項目	移動方向角度	速率	移動秒數	理想距離	實際距離	誤差值
1	0°	10 cm/s	5 秒	50cm	49cm	-2%
2	90°	10 cm/s	5 秒	50cm	52cm	4%
3	135°	10 cm/s	5 秒	50cm	39cm	-22%
4	180°	10 cm/s	5 秒	50cm	51cm	2%

表 5. 測量數值（速率改變，其他參數不變）

項目	移動方向角度	速率	移動秒數	理想距離	實際距離	誤差值
1	90°	10 cm/s	5 秒	50cm	48cm	-4%
2	90°	20 cm/s	5 秒	100cm	96cm	-4%
3	90°	40 cm/s	5 秒	200cm	190cm	-5%
4	90°	70 cm/s	5 秒	350cm	328cm	-6.2%

表 6. 測量數值（移動秒數改變，其他參數不變）

項目	移動方向角度	速率	移動秒數	理想距離	實際距離	誤差值
1	90°	10 cm/s	2.5 秒	25cm	26cm	4%
2	90°	10 cm/s	5 秒	50cm	47cm	-6%
3	90°	10 cm/s	7.5 秒	75cm	77cm	2.6%
4	90°	10 cm/s	12 秒	120cm	119cm	-0.8%

實驗結論：表 4 中得知在進行前後左右等直線移動時移動偏差較小，但在進行斜線移動時，由於加速度過大而導致輪子打滑最終移動偏差幅度較大。表 5 證實加速度越大輪子越容易打滑而造成移動偏差。表 6 則是在將移動方向與速率都維持在最佳狀況時改變加速時間所造成的短小誤差。

二、語音辨識結果

(一) 語音辨識資料自信值：當語音辨識完時會有一個自信值被丟回，這個自信值代表語音辨識 API 對自己辨識出來的語音資料的準確度自認有多少，透過這個數值進行了 20 筆的辨識，男、女、老、少……等來進行數據收集，得出結果不管是什麼樣的人，說話只要發音正確辨識的自信值一定有 90% 以上，統整數據後辨識率 > 0.9 的有 80%。

```
done
take me to the bathroom
0.978923738003
```

圖 20. 利用 Google 翻譯念出來的基準值

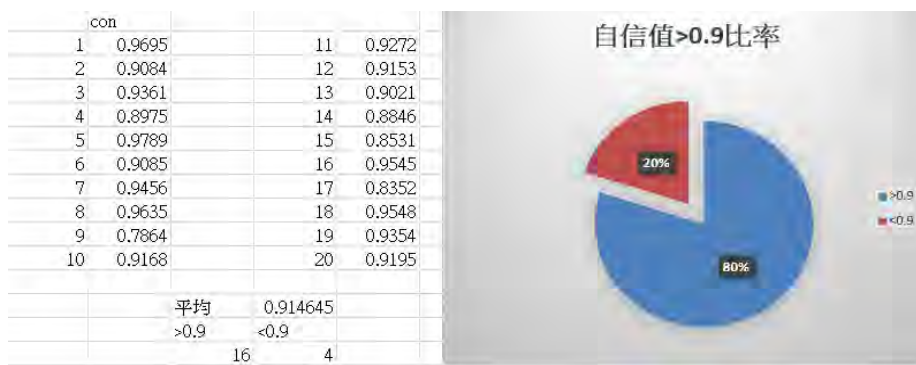


圖 21. 自信值推算圖表

三、定位實驗結果

定位實驗採用以藍芽的 RSSI 進行三角定位，及以 SIFT 技術為主的視覺定位進行實驗，評估何為適當的定位方法。實驗結果如下：

(一) 三角定位測試：利用空曠場地進行三角定位之實驗，將接收站與信號源準確地擺在理想值之距離（中間無障礙物），藉由後台算出以下數據。

表 7. 三角定位實際與測量值

理想值	測量圖式	測量值 (RSSI)	測量值 (CM)	誤差值
50cm	<pre> Discovered device b8:27:eb:af:bc:fc Device b8:27:eb:af:bc:fc (public), RSSI=-53 dB 0.1 -0.603021516837 0.54715589518 -53 Flags = 1a Manufacturer = 4c000215e20a39f473f54bc4a12f17d1ad07a9610000000c8 </pre>	-53db	54.7cm	9.4%
75cm	<pre> Discovered device b8:27:eb:af:bc:fc Device b8:27:eb:af:bc:fc (public), RSSI=-61 dB 0.1 -0.263458475048 0.768389529115 -61 Flags = 1a Manufacturer = 4c000215e20a39f473f54bc4a12f17d1ad07a9610000000c8 </pre>	-61db	76.8cm	2.4%
125cm	<pre> Discovered device b8:27:eb:af:bc:fc Device b8:27:eb:af:bc:fc (public), RSSI=-73 dB 0.1 0.245886087635 1.27875389936 -73 Flags = 1a </pre>	-73db	127.8cm	2.24%

實驗結論：用三角定位測量時，因藍芽訊號的穩定值不夠，單一個藍芽和定位點就有誤差，雖然使用更多藍芽來定位時誤差將會縮小，但展場還有障礙物干擾藍芽訊號強度，因此測量值與實際值的誤差值很大，故不適用此系統。

(二) 視覺定位測試：透過 SIFT 處理，可以在影像中找到特徵點，藉由前後照片的比對，可以得知自己本身的位置及方位。



(a) 找出畫面的特徵點


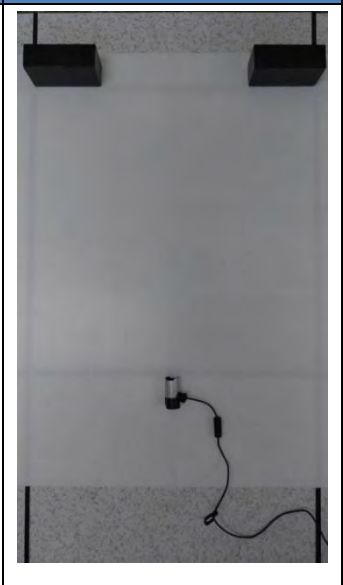


(b) 找出前後兩張照片相關的特徵點

圖 22. 實測 SIFT 的功能

(三) 視覺定位實驗結果：利用兩物體的特徵點求出外框，得出物體中心點 (x,y) 座標，再利用和攝影機照出的照片相素的值，帶入圖 16(b) 公式，求出攝影機時機位置。

表 8. 視覺定位實際計算

物體 x、y 座標	測量俯視圖
	

以兩物體之座標數值及照片相素帶入公式算出攝影機位置。

$$\frac{400 - 91}{400} \times 55^\circ = 42.5^\circ$$

$$\overline{AC} = a = 400 - 87 = 313$$

$$\overline{BC} = 714 - 400 = 314$$

$$c = \frac{313}{42.5} = 463$$

$$b = \sqrt{463^2 - 313^2} = 341$$

$$\Delta x = (695 - 91) \times \frac{313}{313 + 314} = 302$$

$$x_0 = 302 + 91 = 393$$

$$y_0 = 273 - 341 = -65$$

∴ 攝影機位置為(393, -65)

圖 23. 數值帶入公式計算

四、系統整合測試

測試導覽車是否能正確的讀取影像定位完的值，移動並且傳回 Node-Red 伺服端，讓後台跟使用者都能從介面中看到車子的移動狀況，並收集使用者的資訊。

(一) 導覽車移動實驗：在移動式驗中導覽車可以正確讀到影像定位值，並且知道自己在哪如何移動到使用者想到達之地點，下圖的例子中使用者希望移動到 A 點。使用者透過語音，使用者只要喊 A，則會啟動導引到 A 點的程序。

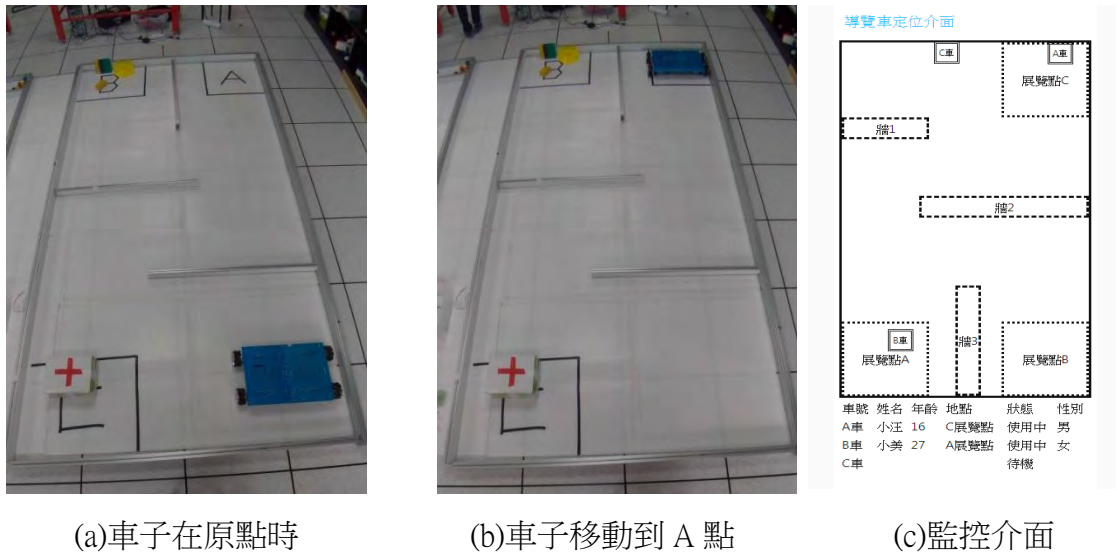


圖 24. 導覽車移動環境及監控

後台可以藉由導覽車定位介面來了解每個車子的使用狀況。透過這個介面，工作人員可以得知所有導覽車所在，也可以監控是否會發生碰撞的狀況，並提出警示，或者進行人為的服務程序。

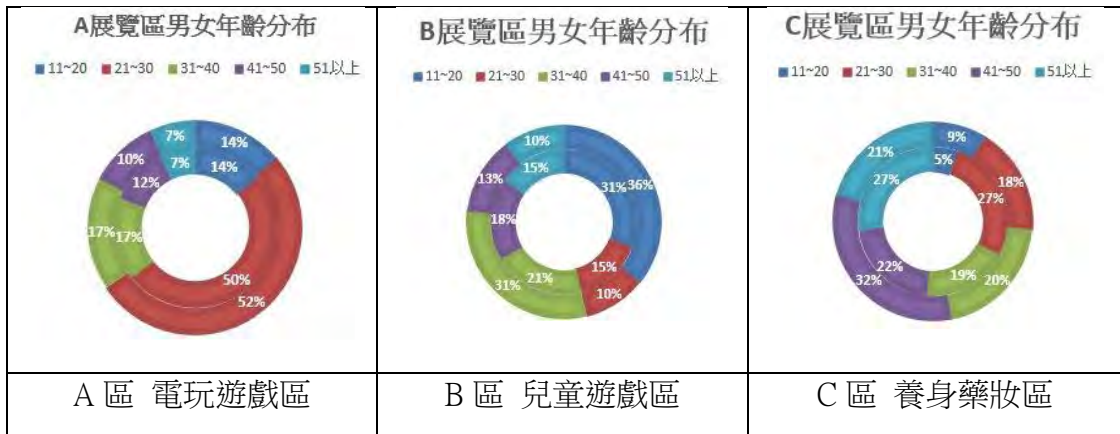
(二) 導覽車使用者控制面板：使用者可以用控制面板，跟機械車對話，達到移動和導覽目的，並且數值會傳回 node-red 的後端管理員看，讓管理員可以即使應對問題。



圖 25.使用者介面

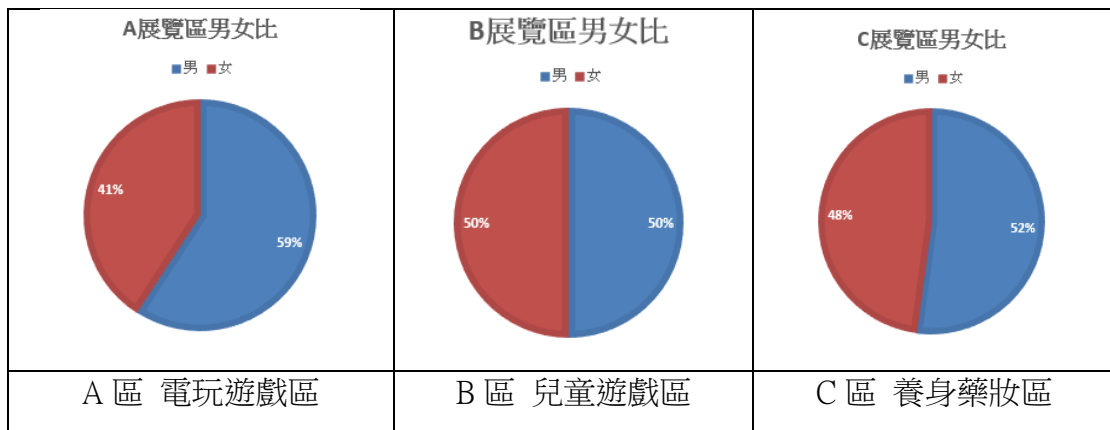
(三) 服務資訊統計：為了提供後續服務策略的改進及修正參考，本系統可以利用使用者相關的數據進行統計，以機械導覽車的在各展覽區使用率狀況，依據性別、年齡統計為例。

表 9. 各展場使用狀況統計（內圈為男生，外圈為女生）



上表是在一天內參展的男性、女性、年齡及人數的比例圓餅圖

表 10. 各展場使用性別比狀況統計



陸、討論

Q1：除了單一語音的操作模式之外，有沒有考慮其他控制方式？

A：考慮到使用者希望可以涵蓋各種需求，比如說盲人，因此決定採用語音控制方式為主，包括環境狀況、按鍵操作方面困難等問題。另外有一個操作介面可以供使用者在平板上按一鍵操作。

Q2：該機械板車是否有跟生活中的東西相同？

A：有，可採用 oBike 共享單車的方式，讓使用者可以更方便取得服務。

Q3：本機器應該放在哪裡才能造福行動不便者？

A：學校、捷運站或大型展場。

Q4：如何將文字轉化成語音讓使用者了解？

A：本系統使用「Google Speech API」的語音功能，將字串輸入「Google Speech API」的語音播放函式進行播放。

Q5：超音波感測器是否可以有效辨別障礙物

A：在經過多次可以感應 5~30 cm 且可以利用 PID 有效地進行等減速運動。

Q6：影像辨識的準確度與速度是可以應付展場的複雜環境嗎？

A：經過測試便是的準確度是沒有問題的，但是受限於樹莓派的硬體效能，辨識的速度可以再加強。

Q7：伺服後端的管理者可以有效掌握導覽車的動向和狀況嗎？

A：利用 Node-Red 的物聯網系統我們可以有效掌握車輛動向、使用者的資訊、車體狀況。

Q8：當車體出問題時，可以迅速知道且處理嗎？

A：車體出問題時伺服後端會收到資料，且派人趕到現場排解問題

柒、結論

本團隊所製作的「機械導覽車」，經過模擬使用者實境的實地測試後，達成以下功能：

- 一、確實可有效的幫助使用者使用語音操作前後左右的移動。
- 二、藉由語音辨識理解使用者的需求，透過聊天機器人進行適當的回應，提供適切的服務，服務有：
 - (一) 依照語音要求載送使用者到指定展區位置或其他設施，如廁所、餐廳…等。
 - (二) 並透過語音撥放需要的導覽資訊。
- 三、安全上，前後左右採用 4 個超音波感測，避免碰撞等危險發生。
- 四、利用視覺定位算出物體實際的準確位置。
- 五、完成簡單明瞭的使用者介面及情報詳細的導覽車定位介面。

藉此改善視障者在公共場所的生活品質，以落實無障礙空間的精神。

未來展望：

- 一、機器車可以任意移動，但安全性上仍有顧慮，應進行更嚴謹的測試。未來也考慮加入倒單擺自動平衡機制，讓使用者可以平穩的在車上站立。
- 二、運用空間定位及辨識，讓車子移動的位置可以在中控室嚴密監控，在必要時介入人力的服務。
- 三、在機構加強舒適性和美觀，如加入握把讓使用者可以更穩定得站在車上、加入軟質布料讓使用者站立或甚至讓使用坐在椅子上可以更加舒適。

捌、參考資料及其他

一、論文或書籍

施士文（2014）。Arduino 微電腦應用實習。新北市：台科大圖書

鄒應嶼（1996）。直流電動機的工作原理與特性。交通大學控制工程系所

林詠能、趙欣怡（2016）。視障觀眾的博物館服務需求探究（未出版）。臺北市：國立臺北教育大學文化創意產業經營學系

黃仲宇、梁正（2014）。基本電學 I。臺北市：台科大圖書

林后鐘、張俊雄（2017）。程式設計實習使用 C 語言。臺北市：台科大圖書

柯博文（2016）。Raspberry Pi 超炫專案與完全實戰。臺北市：碁峰

二、網路資源

Makeblock Library v3.26。引用自 <https://github.com/Makeblock-official/Makeblock-Libraries>

強大的語音辨識功能。引用自 <https://cloud.google.com/speech-to-text/>

MakeBlock Arduino 微控制器程式。引用自

<https://www.makeblock.com/project/learning-arduino-programming>

<https://github.com/Makeblock-official/Makeblock-Libraries>

超音波避障系統。引用自

<http://epaper.gotop.com.tw/pdf/AEH003200.pdf>

Cloud Speech API 應用實例。引用自

https://blog.gcp.expert/machine-learning-cloud-speech-api/fbclid=IwAR0cqCo5LtB89ZcaehgJ2L_2YF3rVVG4m2_WXsoKOpV6o5i845f5zYhNd-k

使用 Python 把樹莓派改造成語音助手。引用自

<https://segmentfault.com/a/1190000014000349>

利用 Python AIML 製作聊天機器人。引用

<https://www.devdungeon.com/content/ai-chat-bot-Python-aiml>

openCV 影像處理。引用自

<https://makerpro.cc/2018/11/opencv-background-subtractor/>

<https://www.pyimagesearch.com/2017/09/04/raspbian-stretch-install-opencv-3-python-on-your-raspberry-pi/>

空間中向量的正射影。引用自

https://www.youtube.com/watch?v=uEtHCU_EUc

PID 控制器。引用自

<https://zh.wikipedia.org/wiki/PID%E6%8E%A7%E5%88%B6%E5%99%A8>

Node-Red 與 MQTT。引用自

<https://swf.com.tw/?p=1002>

三角定位。引用自

<https://blog.csdn.net/anitinaj/article/details/84396532>

【評語】 052305

1. 此機械導覽車之規劃與設計都相當完善，具備語音操控、聊天、導航、和避障功能等，可視為室內自動駕駛車的原型版。本作品具高強度軟硬體整合，開發出智慧型導覽車，系統複雜，開發具不小難度與挑戰，團隊對製作過程中所面對的問題也都能謹慎探索並一一解決，值得鼓勵。
2. 所使用車體驅動、定位、和避障等技術存在已久，建議團隊可進一步具發想工程和應用之創新性。
3. 建議加強說明所採用這些特定軟硬體的緣由、特色、與不可取代性等，並與系統開發的目的與應用有效連結，提升關鍵效益。
4. 建議針對應用主題，以建構完成的系統，進行量化與質化的驗證。
5. 建議針對應用主題，以建構完成的系統，與現有採行解決方案進行規格與特性比較，已確認所開發系統的競爭力。

摘要

本研究目的在製作一個可以輔助移動的機械板車，作品採用Google雲端服務進行語音辨識透過辨識的結果，對導覽車進行操控，甚至導航到相關位置。用Speech-to-Text功能播放聊天機器人回應內容。此外作品亦考慮到安全，避免行進過程中對服務對象或周圍民眾造成傷害。讓行動不便者，自由的在展場中悠遊。

研究動機

在日常生活中，我們總是享有理所當然的方便與舒適，但社會中仍有一些行動不便的族群。在服務弱勢觀眾參觀博物館需求探究的論文中提到「參觀輔助資源、多媒體數位科技輔具與導覽服務皆為參觀過程的延伸服務，可協助弱勢者獲取更多展覽資訊。」(林詠能、趙欣怡，2016)。若能藉由機械板車的功能，輔助行動不便的弱勢族群跟我們一樣可在展場享受移動便利。

研究目的

- 一、探討運用麥克納姆輪及直流減速編碼馬達進行機械板車的方向移動控制。
- 二、研究以Google Speech-to-Text的API進行語言辨識。
- 三、研究使用超音波建構安全及避障的機制。
- 四、探討三角定位與視覺定位對比。
- 五、研究以Node-Red監控使用端的使用狀況並蒐集使用者的資訊。

研究方法及設計

系統架構

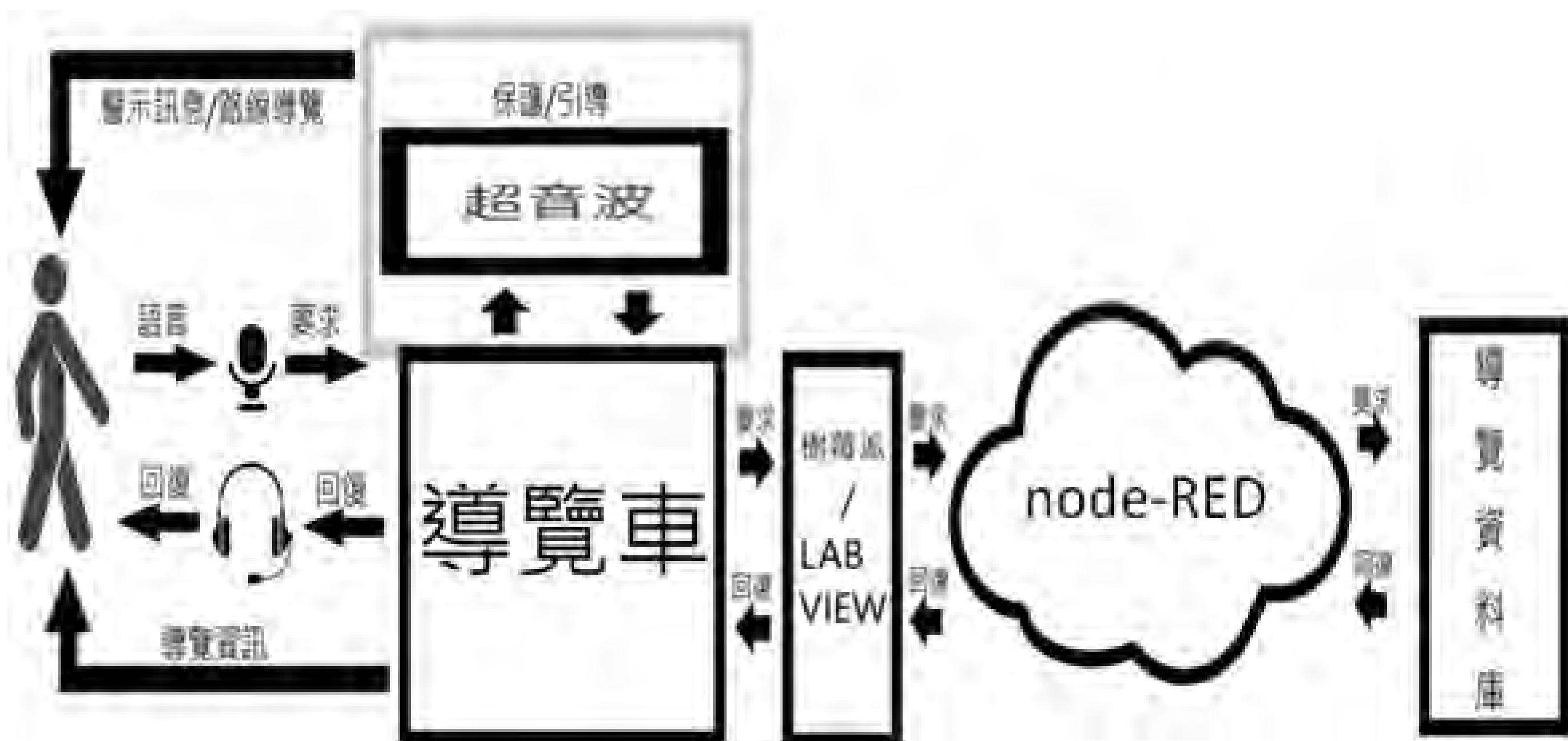


圖1 導覽車運行流程圖

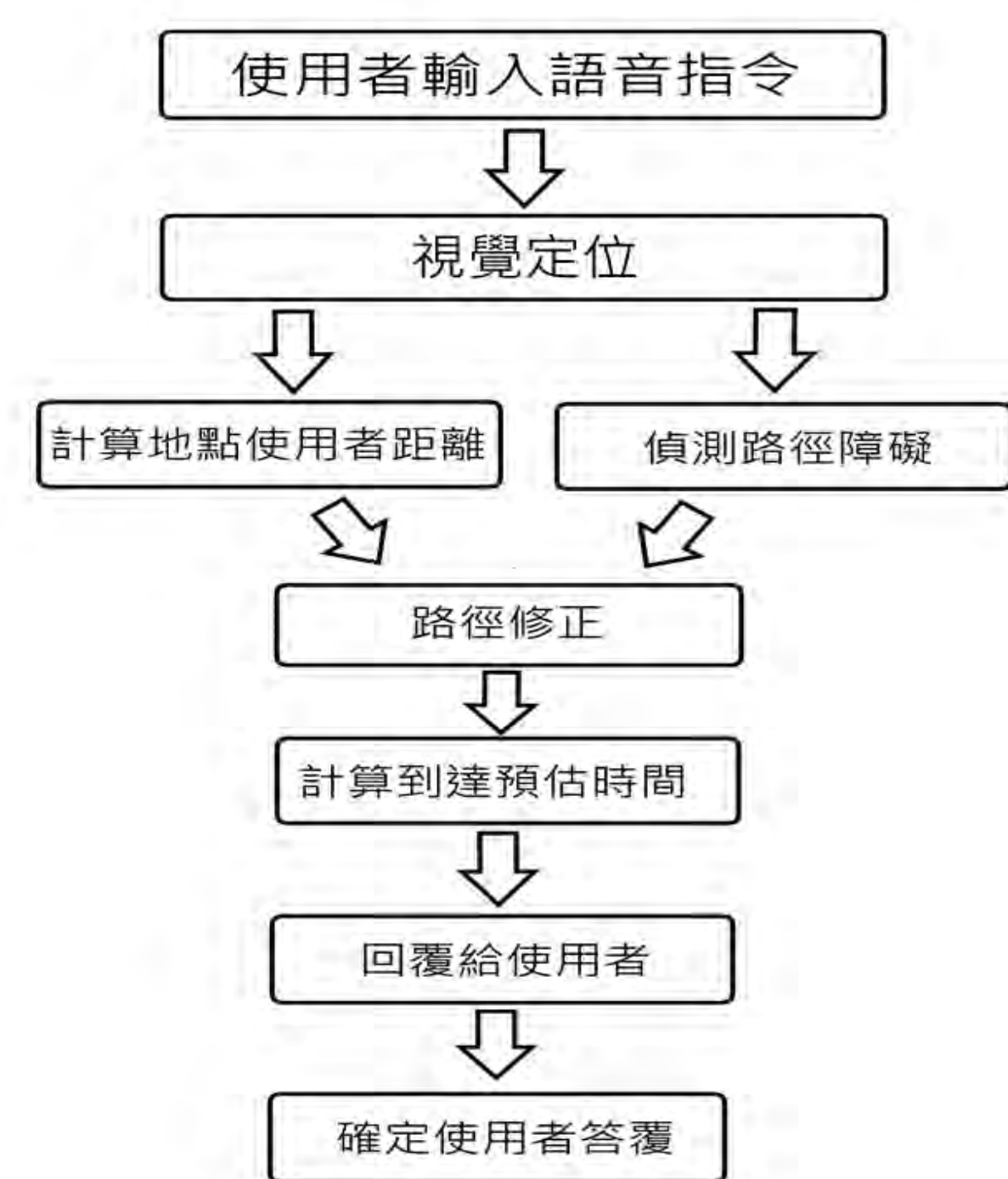


圖2 導覽車程式流程圖

本「機械導覽車」藉由Node-Red、Raspberry Pi、myRIO做為後台以及車體的系統。使用者可使用RFID卡認證登入，取得機械板車的服務。接下來可以運用語音指令，命令板車引導使用者到達指定位置。移動過程中，導覽車也會隨時將所在位置、服務需求等使用者資訊，透過Node-Red回傳給資料庫，作為統計分析使用或即時服務使用。Raspberry Pi在接收到來自使用者的語音指示之後，myRIO會綜合超音波感測器及影像辨識鏡頭所得到的資訊，計算現在位置和到達目的地的最佳路徑，在移動的路程中也會以超音波感測器來偵測任何障礙物。

車身結構介紹

車身部分是使用了MakeBlock配套式骨架，配合Raspberry Pi的myRIO主機板、馬達控制板及四組馬達與麥克納姆輪控制車子的移動。運用樹莓派的無線功能收集及傳送資訊，與遠端伺服器構成一個遠距的監控服務平台及資料庫。電池採用FULLYMAX鋰電池提供高性能的動力輸出，以提高載重及續航能力。



圖3 麥輪機械板車

剛體移動方向與輪子轉速關係

輪子往90°方向轉動時每顆輪子所帶的動力方向不同，由圖推演，將剛體移動向量正投影到四個輪子所產生的向量即為四個輪子的轉向與速度。

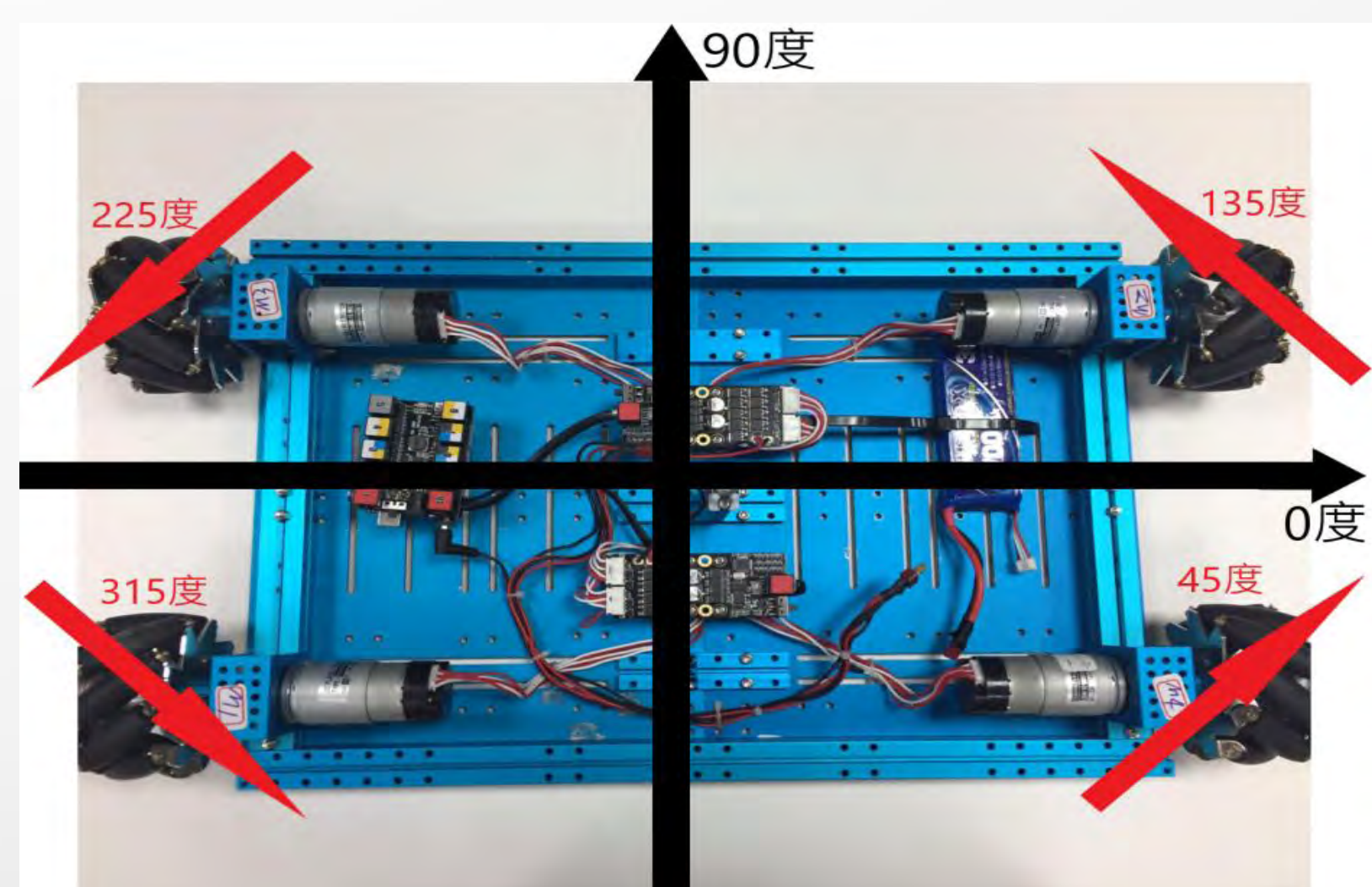


圖4

剛體定義方向與四顆麥克納姆輪的方向關係 (以90°為移動方向)

輪子轉速公式推導

以 \vec{OA} 為車子的主移動方向， $\pm\vec{OC}$ 為輪子移動方向速度，藉由下列公式推算出各個輪子方向的速度。

$$|\vec{OA}| \times \cos(\theta_{\text{輪子角度}} - \theta_{\text{車體移動方向}}) = \text{輪子方向速度 } cm/s$$

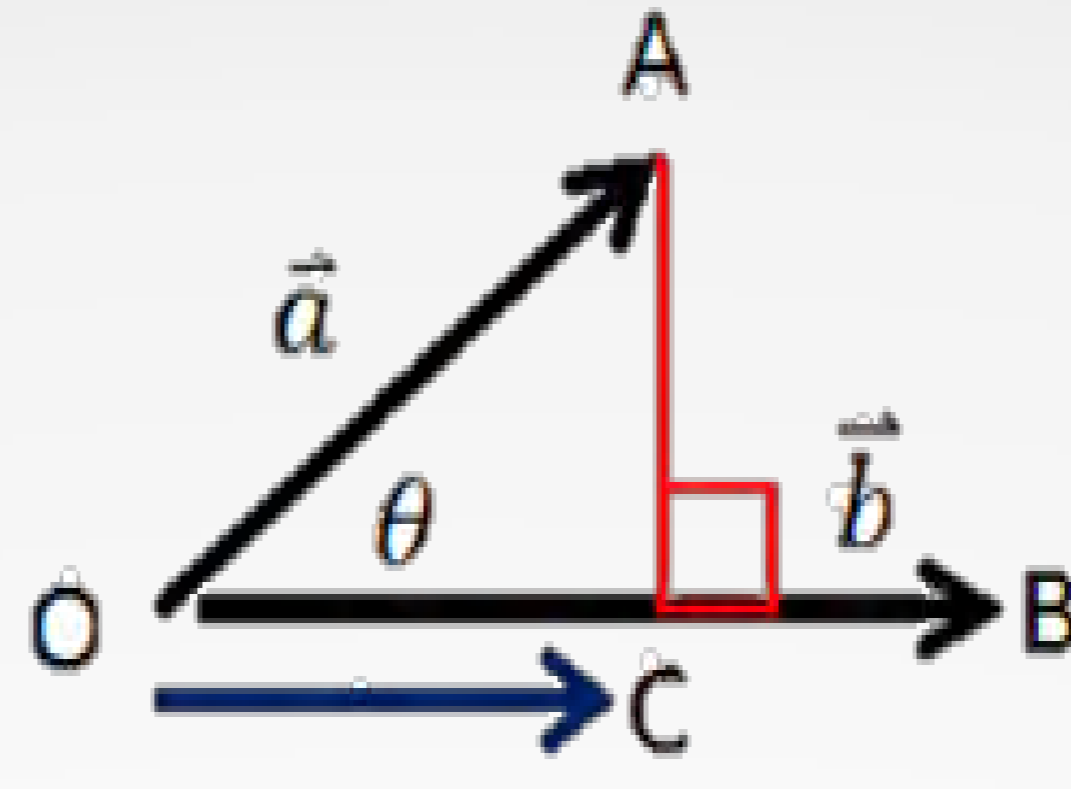


圖5 移動向量 \vec{OA} 正投影在輪子的投影量

麥克納姆輪	輪子速度
A輪	$10 \times \cos(45^\circ - 90^\circ) = 5\sqrt{2}$
B輪	$10 \times \cos(225^\circ - 90^\circ) = 5\sqrt{2}$
C輪	$10 \times \cos(135^\circ - 90^\circ) = 5\sqrt{2}$
D輪	$10 \times \cos(-45^\circ - 90^\circ) = -5\sqrt{2}$

圖6 理論推導 (以向前 $90^\circ \cdot 10cm/s$ 為例)

Google Speech-to-Text API 語言辨識



將 PyAudio 錄出來的語音利用 Python 丟上 Google Speech-to-Text API 進行辨識，Google 會將辨識完的文字與自信值丟回，再利用語音聊天機器人將使用者講的話來進行回答。

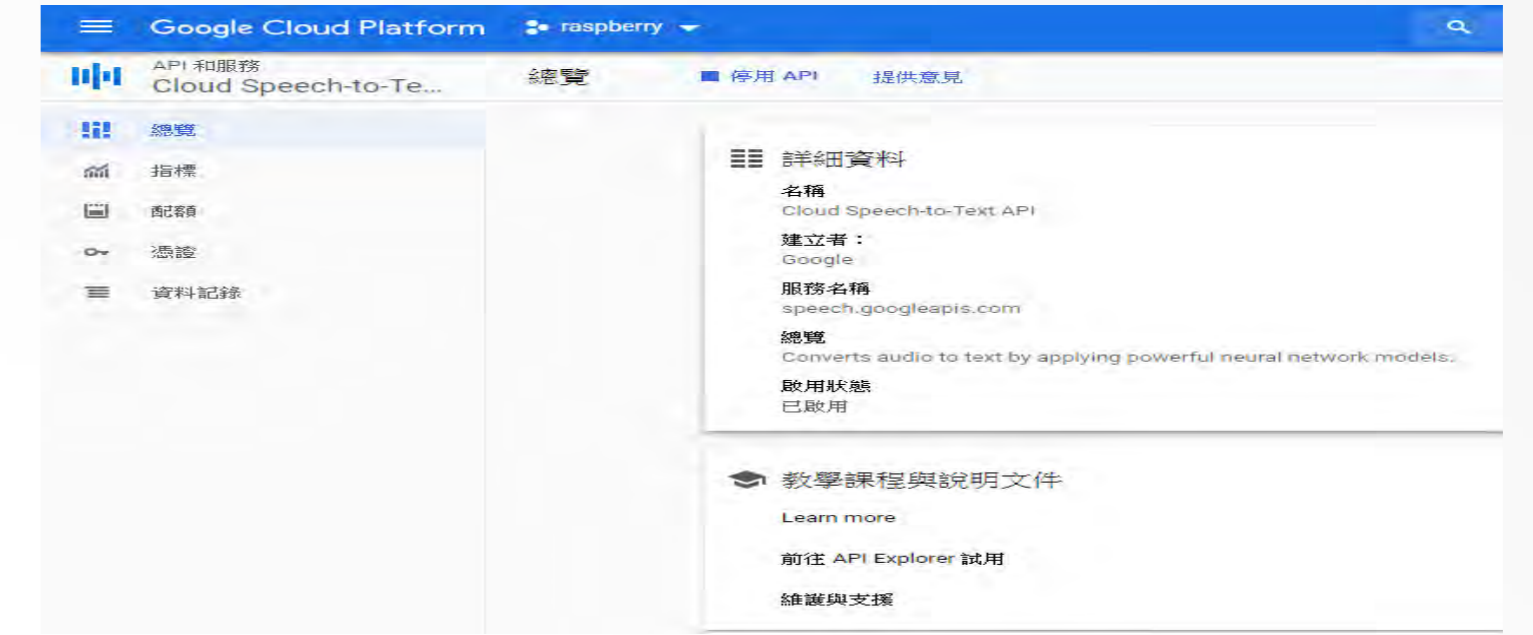


圖7 利用 Cloud Platform 的 Speech-to-Text API 辨識語音

安全避障

在展場中參觀的民眾及各種可能臨時或長期擺設的物品，可能會出現在預先規劃的路線中，車子行進時遇到這些狀況必須即時的停下，以避免撞到民眾或展品的危險。而另外一方面車子上的參觀人，也不可以因為煞車的重心偏移而造成跌倒的情況。本研究使用超音波配合 PD 控制動力來達成，設計說明如下：

超音波測距理論

公式中 V 為超音波在空氣中傳播速度； T 為環境溫度； S 為被測距離； t 為發射超聲脈沖與接收其回波的時間差； t_1 為超聲回波接收時刻； t_0 為超聲脈沖發射時刻。利用 MCU 的捕獲功能可以很方便地測量 t_0 時刻和 t_1 時刻，根據以上公式，用軟體編程即可得到被測距離 S 。

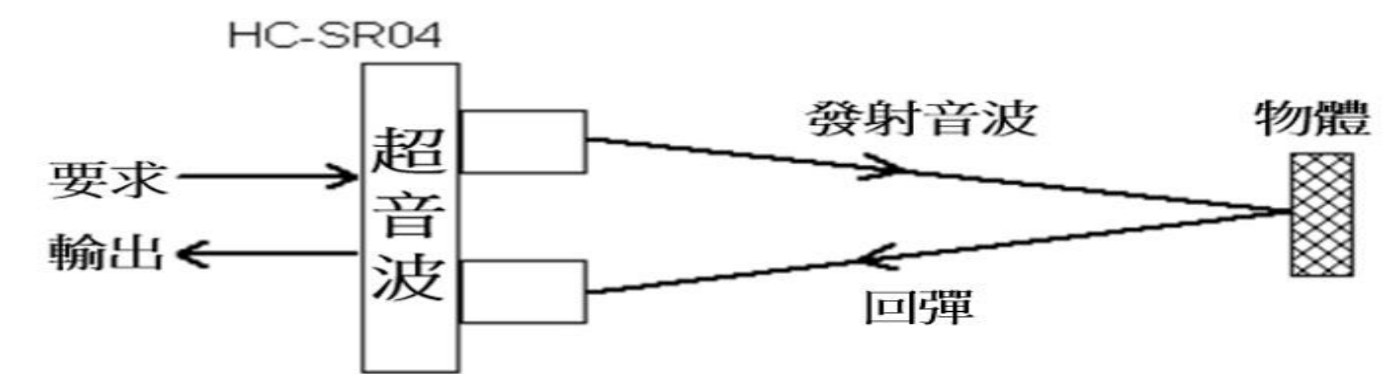


圖7 超音波測距示意圖

$$V = 331.5 + 0.607T$$

$$S = V \times t/2 = V \times (t_1 - t_0) / 2$$

公式推導

PD 減速設計

車子遇到障礙物進行減速並迴避，減速太快會造成使用者發生危險，需要設計減速系統保障使用者的安全。減速系統參考 PID 控制理論。

P：比例控制器，利用超音波的數值控制放大倍數使系統反應加速。

I：積分控制器，消除穩態誤差使系統更精準，本系統需求度不高故不使用。

D：微分控制器，改善系統穩定度使系統穩定工作消除震盪或是發散現象。藉由每個單位時間對超音波距離的變化量，回授給方向向量。

本裝置由比例單元 (P) 和微分單元 (D) 組成，可以透過增益 K_p 和 K_d 來調定其特性，做到穩定的減加速度移動，讓使用者搭乘導覽車時不因為急停或急駛發生危險。

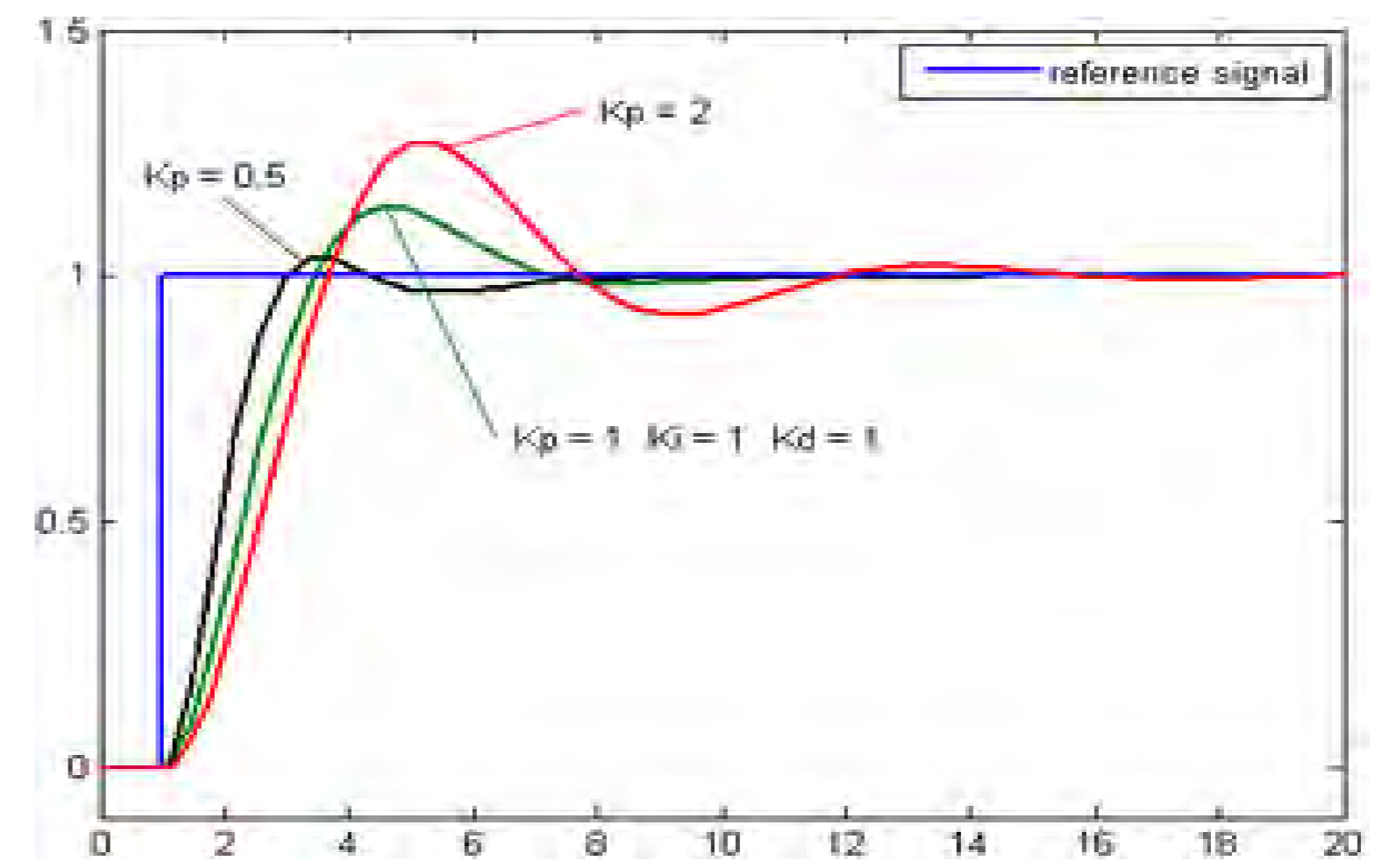


圖8 PID 系統運統曲線圖

$$\begin{cases} \text{距離} > 100cm, V = V_{max} \\ \text{距離} < 100cm \text{ 且 } \text{距離} \geq 50cm, V_{max} \times \frac{\text{距離} - 50cm}{50cm} \\ \text{距離} < 50cm, V = 0 \end{cases}$$

視覺定位

採用尺度不變特徵轉換 (Scale-invariant feature transform, SIFT) 技術進行視覺定位，主要是偵測影像中的局部特徵點，並藉由特徵點與攝影機之間的三角關係，可得到位置、尺度、旋轉的數據。圖的兩綠點為物體特徵點，設置的特徵點為已知的座標，當攝影機拍攝到這兩個座標點後，依照相片中兩點的距離及座標，推算攝影機本身的座標，若產生障礙物，亦可推算離障礙物的距離。

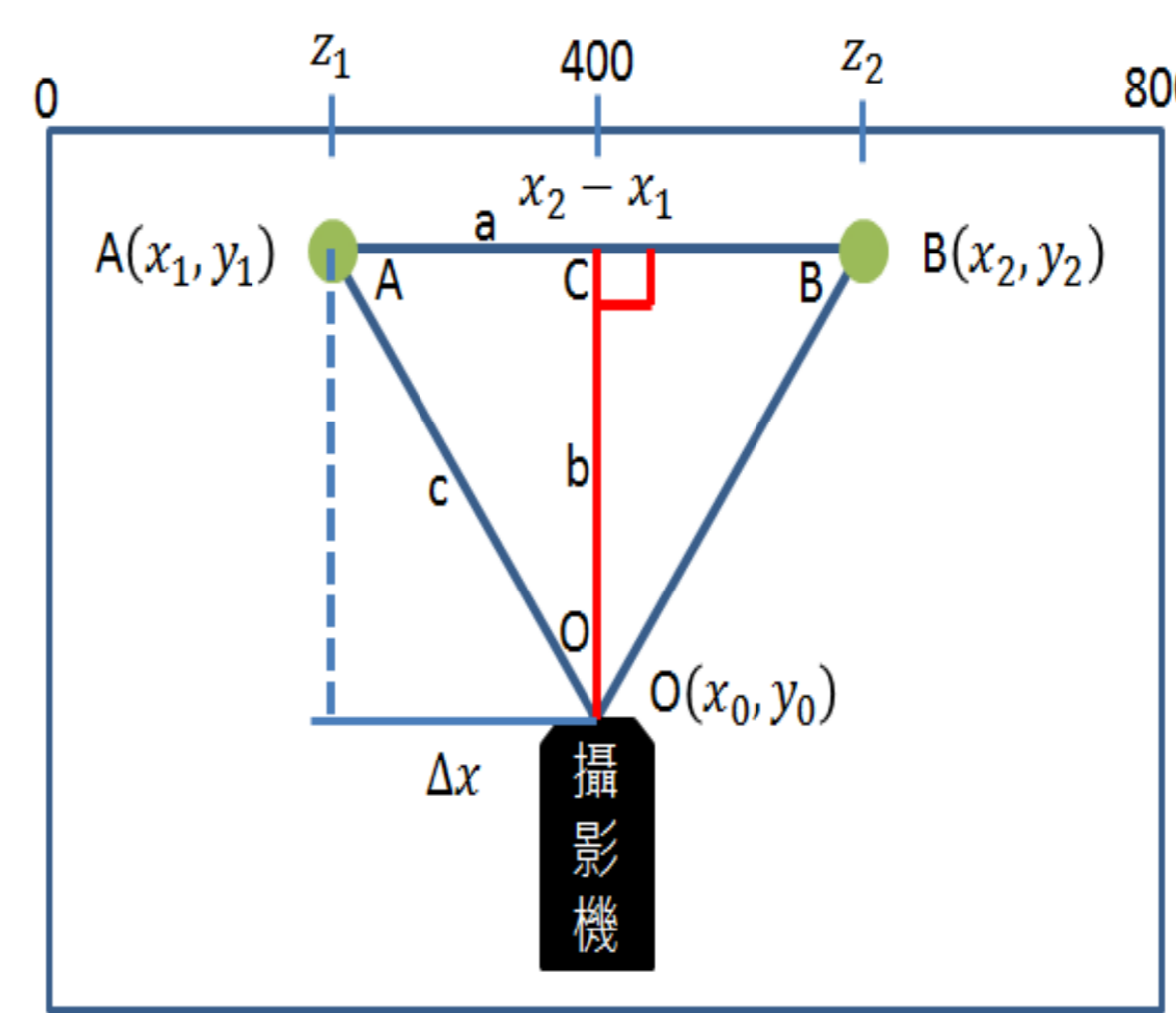


圖9 用特定的特徵點 (綠圓圈代表) 來定位

$$\frac{400 - x_1}{400} \times 55^\circ = \theta_{\text{物體與攝影機拍攝夾角}}$$

$$\overline{AC} = a = 400 - z_1$$

$$\overline{BC} = z_2 - 400$$

$$c = \frac{\overline{AC}}{\sin \theta_{\text{物體與攝影機拍攝夾角}}}$$

$$b^2 = \sqrt{c^2 - a^2}$$

$$\Delta x = (x_2 - x_1) \times \frac{\overline{AC}}{\overline{AC} + \overline{BC}}$$

$$x_0 = \Delta x + x_1$$

$$y_0 = y_1 - \overline{OC}$$

圖10 車體座標計算公式

Node-Red

Node-Red 是由 IBM Emerging Technology 所開發的系統，採用 Node.js 開發的開發環境及節點套件，可以針對節點進行拉線的方式進行程式編寫的視覺化套件，除了本身的服務外，也可以透過擴充增加功能，或其他擴充 IBM Bluemix 的雲端服務 API。

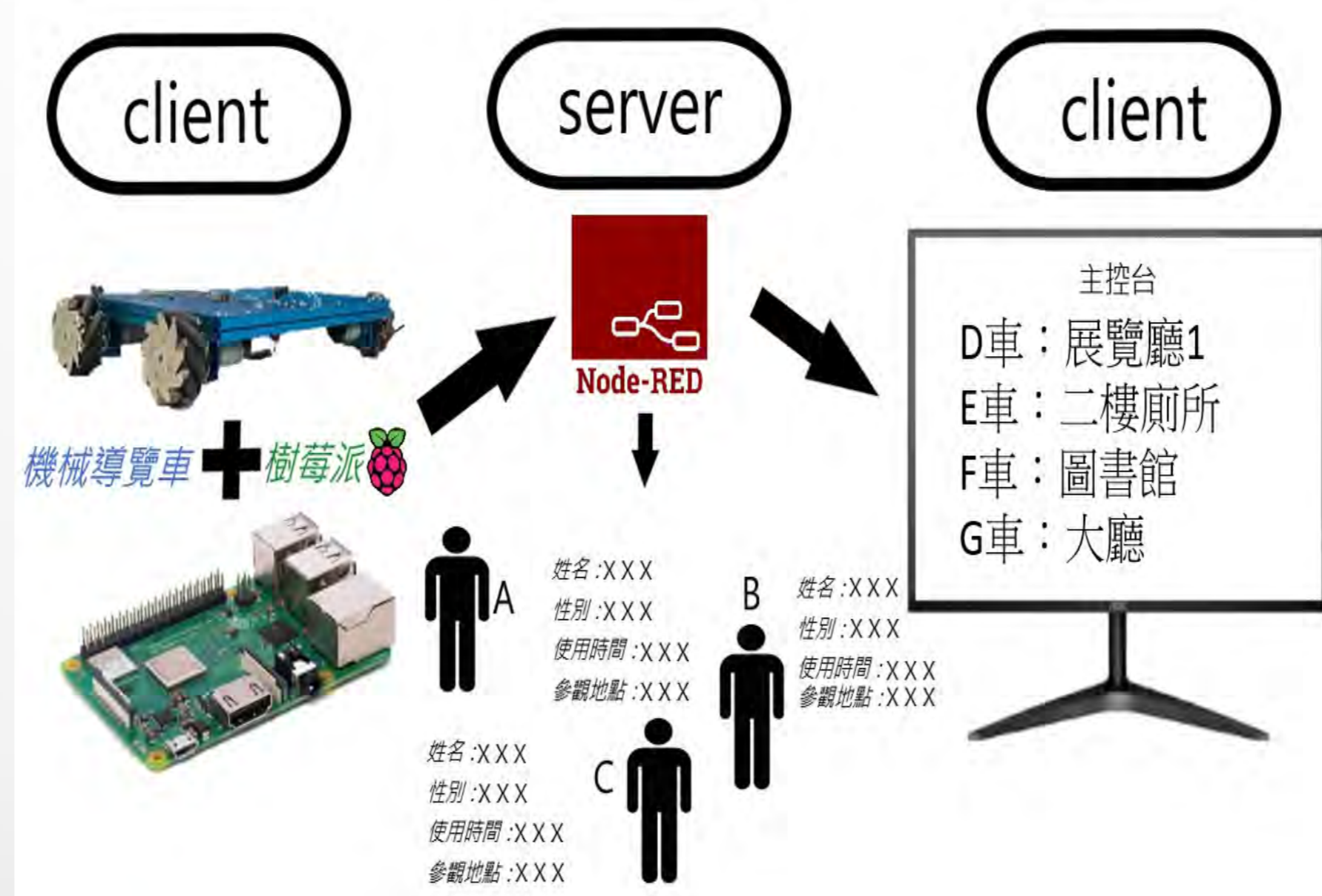


圖11 MQTT 的資訊傳輸概念圖

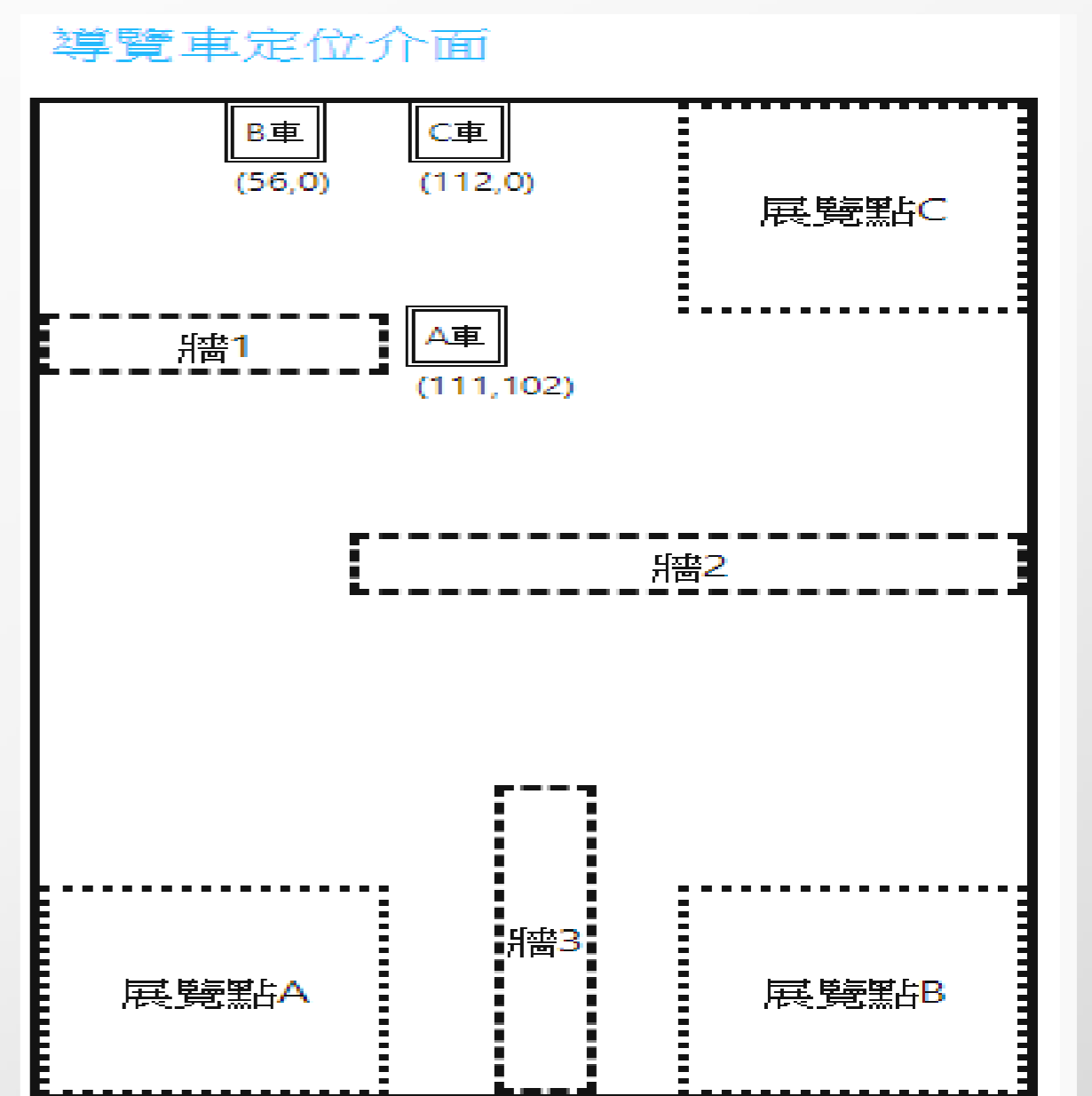


圖12 使用圖形化介面顯示座標位置

研究結果

機械導覽車運作結果

機械板車移動數值：在相同的速率下，以不同移動方向的角度進行測出各項目的秒數和距離誤差值。

表1	移動角度	速率 (cm/s)	移動秒數 (s)	理想距離 (cm)	實際距離 (cm)	誤差值 (%)
1	0°	10	5	50	49	-2
2	90°	10	5	50	52	4
3	135°	10	5	50	39	-22
4	180°	10	5	50	51	2

表1

表2	移動角度	速率 (cm/s)	移動秒數 (s)	理想距離 (cm)	實際距離 (cm)	誤差值 (%)
1	90°	10	5	50	48	-4
2	90°	20	5	100	96	-4
3	90°	40	5	200	190	-5
4	90°	70	5	350	328	-6.2

表2

表3	移動角度	速率 (cm/s)	移動秒數 (s)	理想距離 (cm)	實際距離 (cm)	誤差值 (%)
1	90°	10	2.5	25	26	4
2	90°	10	5	50	47	-6
3	90°	10	7.5	75	77	2.6
4	90°	10	12	120	119	-0.8

表3

測量數值（移動角度改變，其他參數不變） 測量數值（速率改變，其他參數不變） 測量數值（移動秒數改變，其他參數不變）
表1 中得知在進行前後左右等直線移動時移動偏差較小，但在進行斜線移動時，由於加速度過大而導致輪子打滑最終移動偏差幅度較大。表2證實加速度越大輪子越容易打滑而造成移動偏差。表3則是在將移動方向與速率都維持在最佳狀況時改變加速時間所造成的短小誤差。

語音辨識結果

當語音辨識完時會有一個自信值被丟回，這個自信值代表語音辨識API對自己辨識出來的語音資料的準確度自認有多少，透過這個數值進行了20筆的辨識，男、女、老、少.....等來進行數據收集，得出結果不管是什麼樣的人，說話只要發音正確辨識的自信值一定有90%以上，統整數據後辨識率>0.9的有80%。

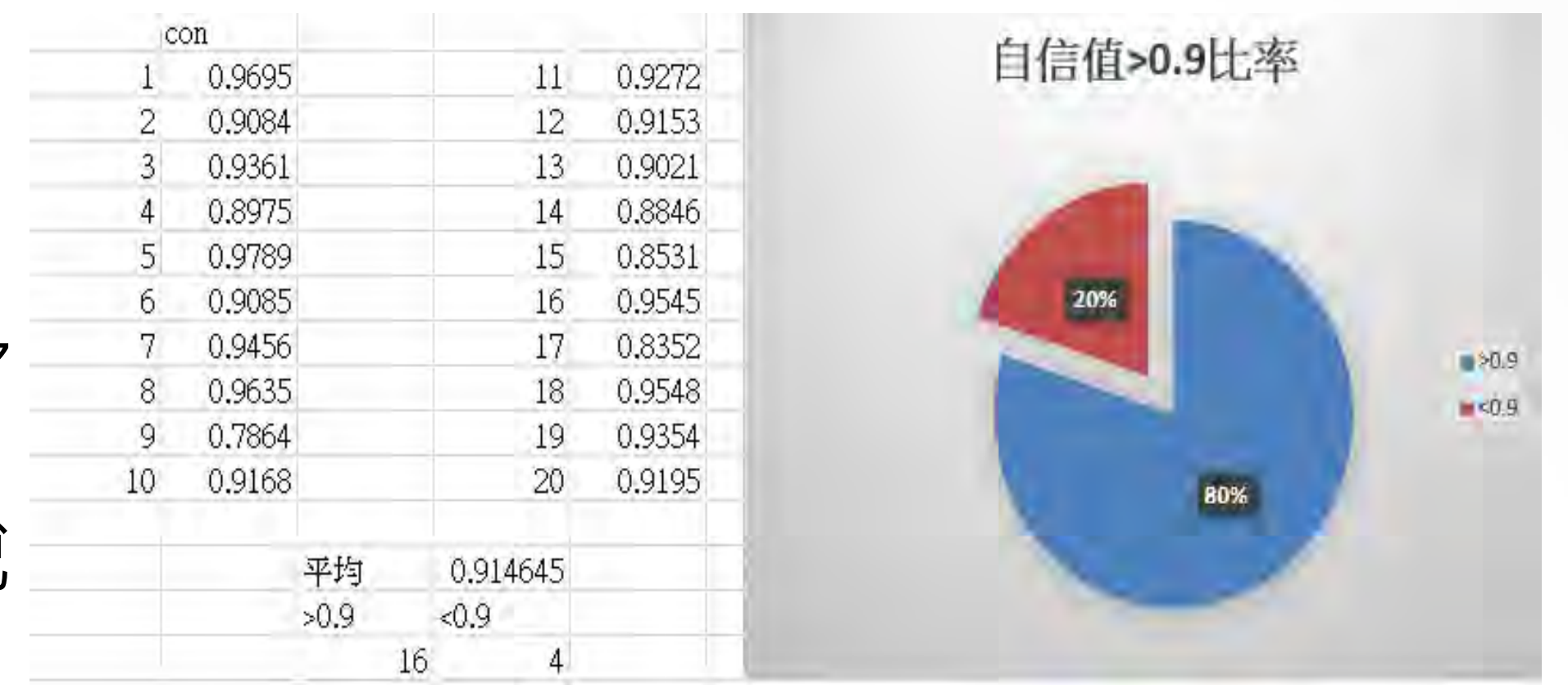


圖13 自信值推算圖表

視覺定位測試

透過 SIFT 處理，可以在影像中找到特徵點，藉由前後照片的比對，可以得知自己本身的位置及方位。



圖14 找出畫面的特徵點

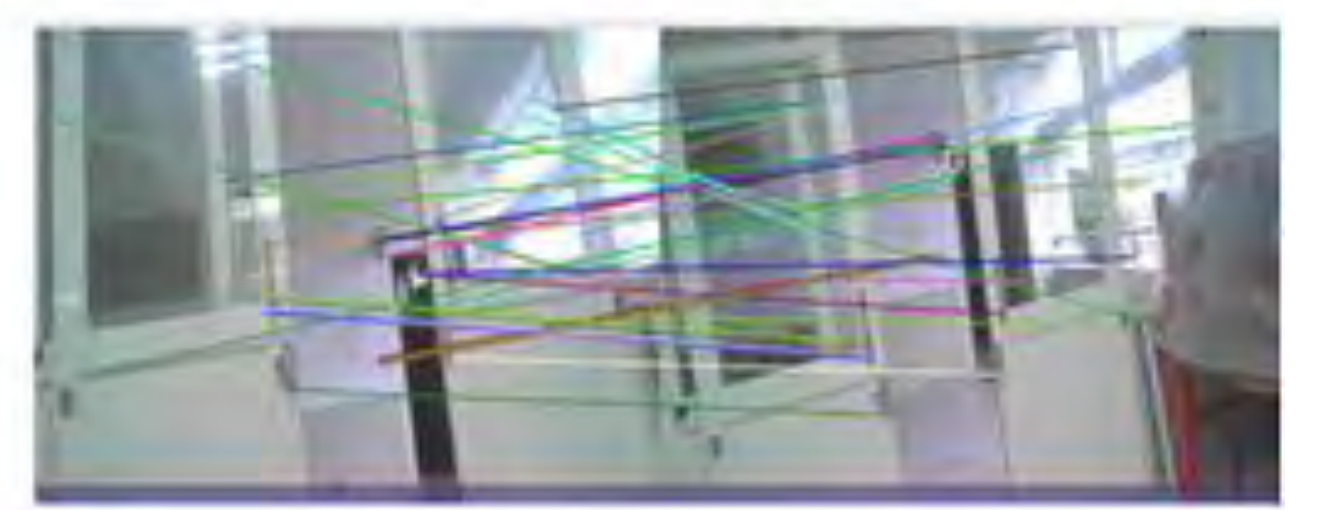


圖15 找出前後兩張照片相關的特徵點

視覺定位實驗結果

利用兩物體的特徵點求出外框，得出物體中心點 (x,y) 座標，再利用和攝影機照出的照片相素的值，帶入圖16 (b) 公式，求出攝影機時機位置。

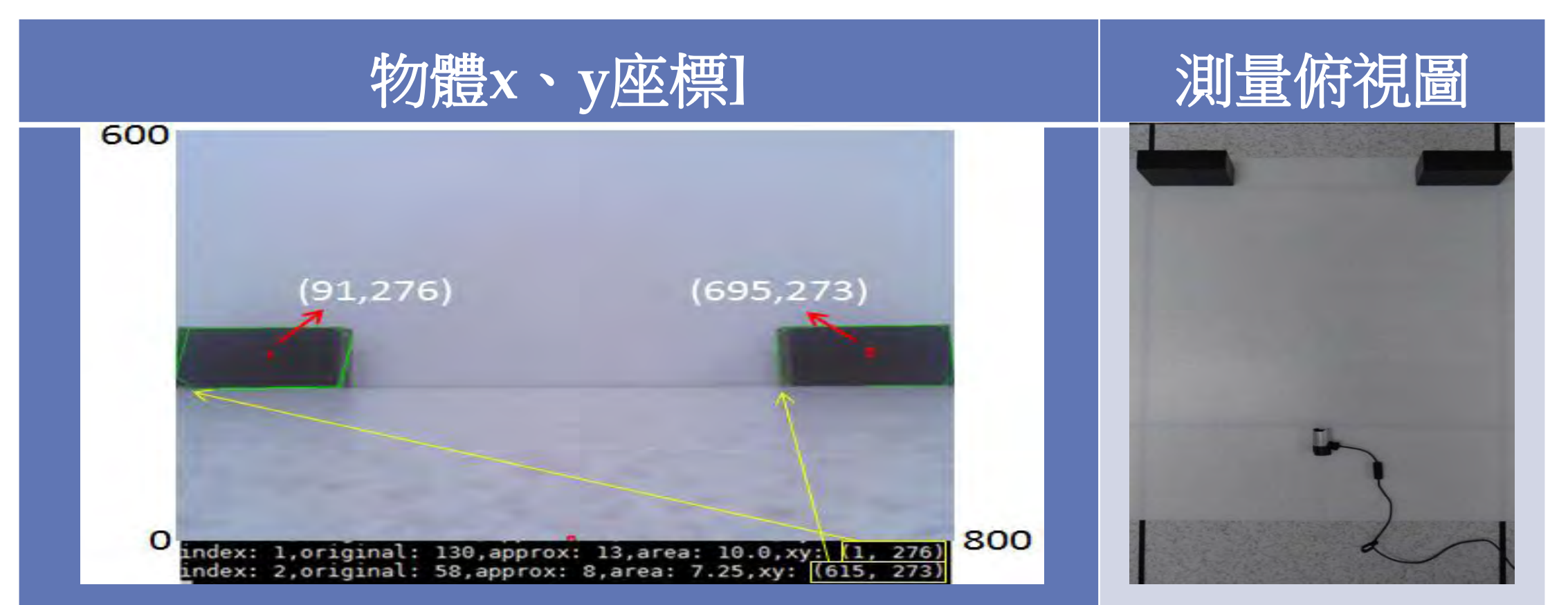
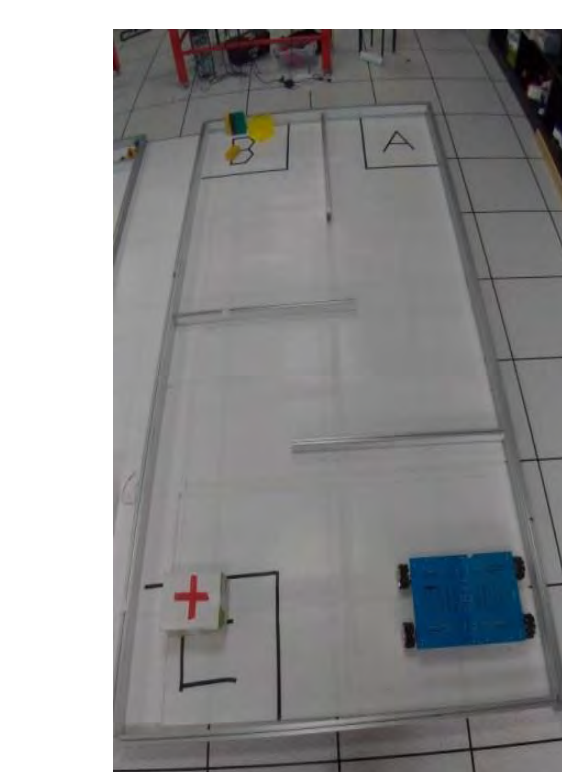


圖16 視覺定位模擬圖

系統整合測試

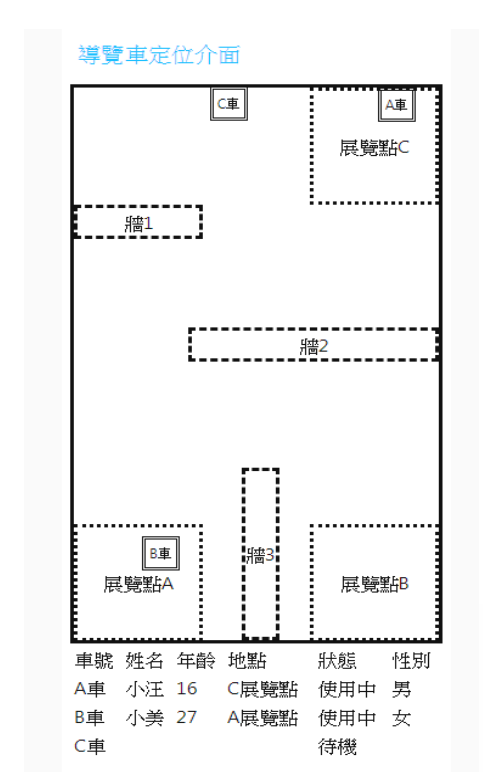
測試導覽車是否能正確的讀取影像定位完的值，移動並且傳回Node-Red伺服端，讓後台跟使用者都能從介面中看到車子的移動狀況，並收集使用者的資訊。後台可以藉由導覽車定位介面來了解每個車子的使用狀況。透過這個介面，工作人員可以得知所有導覽車所在，也可以監控是否會發生碰撞的狀況，並提出警示，或者進行人為的服務程序。



(a)車子在原點時



(b)車子移動到A點



(c)監控介面

圖17 系統整合模擬圖

結論

本團隊所製作的「機械導覽車」，經過模擬使用者實境的實地測試後，達成以下功能：

- 一、確實可有效的幫助使用者使用語音操作前後左右的移動。
 - 二、藉由語音辨識理解使用者的需求，透過聊天機器人進行適當的回應，提供適切的服務，服務有：
 - (一) 依照語音要求載送使用者到指定展區位置或其他設施，如廁所、餐廳...等。
 - (二) 並透過語音撥放需要的導覽資訊。
 - 三、安全上，前後左右採用4個超音波感測，避免碰撞等危險發生。
 - 四、利用視覺定位算出物體實際的準確位置。
 - 五、完成簡單明瞭的使用者介面及情報詳細的導覽車定位介面。
- 藉此改善視障者在公共場所的生活品質，以落實無障礙空間的精神。

未來展望

- 一、機器車可以任意移動，但安全性上仍有顧慮，應進行更嚴謹的測試。未來也考慮加入倒單擺自動平衡機制，讓使用者可以平穩的在車上站立。
- 二、運用空間定位及辨識，讓車子移動的位置可以在中控室嚴密監控，在必要時介入人力的服務。
- 三、在機構加強舒適性和美觀，如加入握把讓使用者可以更穩定得站在車上、加入軟質布料讓使用者站立或甚至讓使用坐在椅子上可以更加舒適。