

智慧販賣機

Intelligent Vending Machine

指導教授：江緣貴老師

學生：吳邁龍、周子謙、方祥豪、張啟賢

國立聯合大學資訊工程學系

苗栗市南勢里聯大 2 號

ykchiang@nuu.edu.tw

摘要

隨著科技的日新月異，人類為了更有效的運用時間及控管成本而發明出了販賣機，目的在於更有效率的販賣商品，最大化自己的利潤。目前所研究的販賣機將結合網頁、RFID 和語音等方法進行控制，我們將透過網頁或手機 APP 輸入帳號或 RFID 等方式登入，並選擇使用網頁按鈕亦或是語音等方式控制機器選擇需要的商品。

關鍵字:物聯網、語音辨識、Raspberry Pi 3、Android、販賣機。

一、前言

隨著科技的快速發展，人們的生活品質不斷提升，市面上開始充斥著琳瑯滿目的商品，而在如此激烈的競爭環境下衍生出了各式各樣的自動販賣機的科技產品。我們希望藉由專題製作能夠打造出擁有多樣功能的販賣機。目前市面上現有的販賣機，大部分多是針對商品種類而衍生出的，而我們希望能夠打造出結合多樣消費方式的販賣機提供消費者更便利的購物體驗。在中國大陸，手機支付相當的普及，食衣住行只要用微信或支付寶感應一下，出門不用帶現金也可以，藉由虛擬貨幣消費的行為已經深植大陸人民的生活當中。藉由 RFID 技術，我們可以用卡片支付的方式來消費，擺除掉傳統用現鈔來消費的概念，

使得消費、找零都能用一張卡來完成，不僅方便又節省了成本。

二、專題內容與說明

2.1 開發環境

我們採用了 Raspberry Pi 並使用 Python 語言作為智慧販賣機的核心，因為它符合能做簡易的資料處理與連上網路的需求，加上 Arduino 來控制硬體電路，驅動馬達來控制販賣機。

語音辨識的部分將採用四麥克風陣列擴充版(Respeaker)接受使用者的指令。

取餐部分使用 WebCam 進行手機條碼辨識並使用 Arduino IDE 控制販賣機轉動與取餐櫃開關，資料庫使用 PHP 與 MySQL。

三、專題理論部份

3.1 系統架構介紹

1. 販賣機商品架:

利用 28BYJ-48 步進馬達配合螺旋的金屬線圈來製作一個商品架，出貨的原理是每當步進馬達轉一圈，金屬線圈也會跟著轉一圈，藉由金屬線圈商品會自動往前推進，直到從商品架上掉落，達到出貨的效果。

2. 光敏模組與 LED 燈條：

透過照度計模塊 GY-30(光敏模組)，量測光的強度(Lx)，Lux 勒克斯 Lx 代表光的亮度單位，透過讀取數值 Lx 以判斷附近環境的強弱，在光線較微弱的時候開啟 LED 燈條，來達到照明的效果，以用來模擬販賣機可以 24 小時營業的狀況。

3. 取餐櫃：

使用 arduino 和繼電器控制電池鐵，繼電器可以用較小的電流去控制較大電流，電磁鐵用較大的電流可以有較大的磁力，所以使用繼電器來控制點餐櫃的開關。

4. 語音辨識：

透過 Google ReSpeaker 抓取使用者的語音，使用 Google Assistant API 將語音轉換成文字，根據辨識出的文字來控制販賣機的選擇商品功能。

3.2 系統架構圖

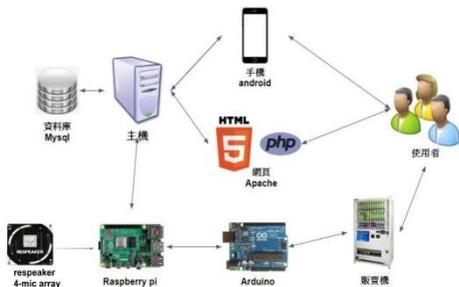


圖 1. 系統架構圖

3.3 功能架構圖

使用者操作的功能，需要有會員系統、點餐、購買商品及查詢紀錄等功能，因此我們設計的功能架構圖如下

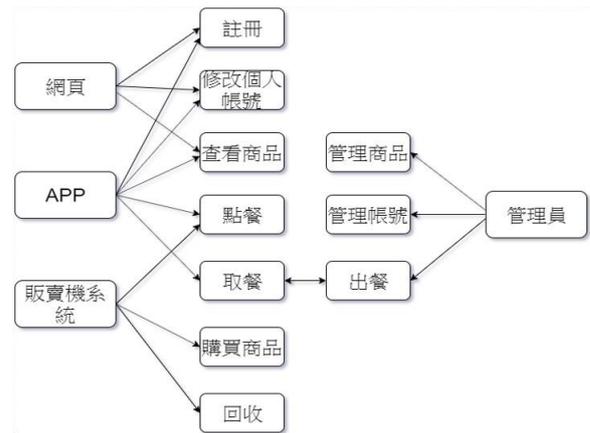


圖 2. 功能架構圖

3.4 ER 模型

販賣機系統裡有會員及購物等功能，需要用到資料庫來儲存商品、會員、訂單等資料，所以要繪製 ER 模型，來確保資料庫的正確及穩定，下圖為資料庫的 ER 模型

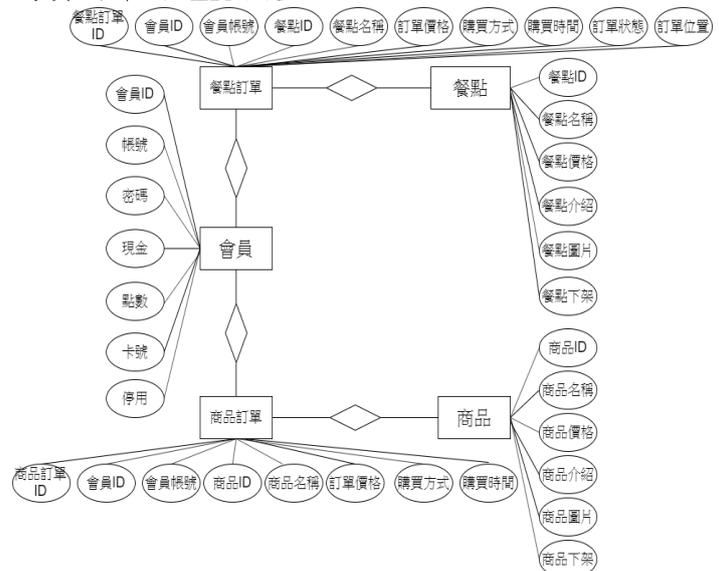


圖 3. ER 模型

四、專題實作部分

4.1 語音辨識

本次實作將使用 Google Assistant API，由 Actions Console 管理的 Google Cloud Platform 項目可讓使用者的項目訪問 Google Assistant API。該項目跟踪配額使用情況，並為您的硬件請求提供有價值的指標。首先打開 google 操作控制台，註冊完專案，之後啟用 api，接著進入 OAuth 同意畫面，儲存資料。

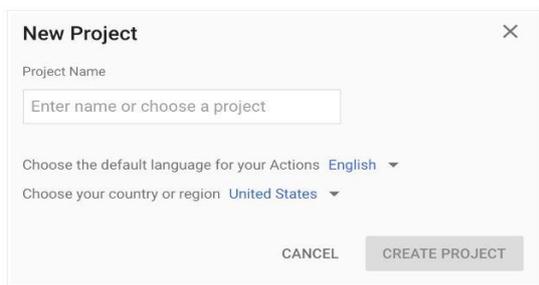


圖 4. API 註冊頁面

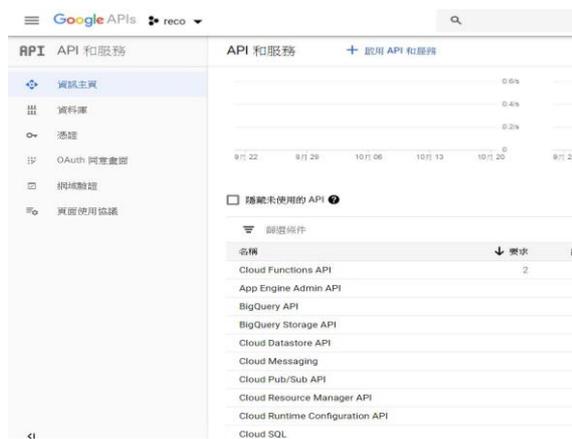


圖 5. API 啟用控制台



圖 6. OAuth 同意畫面

接著進入 google 的 Actions Console 中開啟專案後並註冊設備型號，透過註冊設備的方式，將產生一個特定設備的信息。使用者將提供此信息作為設備型號，其中包括諸如設備類和製造商之類的字段，之後將產生 json 檔作為驗證使用，如圖 7。

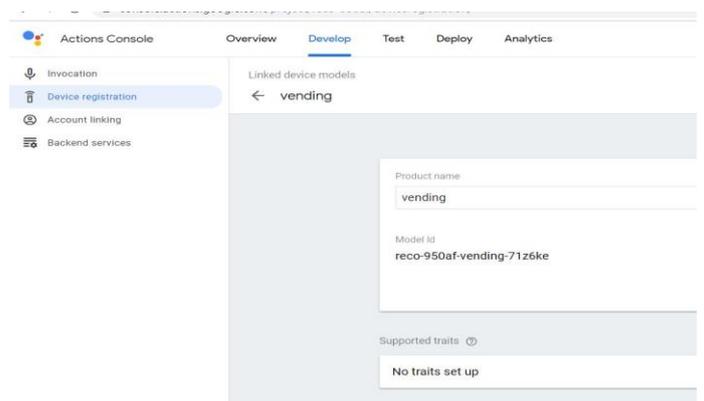


圖 7. 設備型號頁面

在 Raspberry Pi 3 中安裝 SDK 和示例代碼後，使用者將透過口述的方式將

聲音藉由 Raspberry Pi 3 將語音檔上傳進 Google Assistant API 中做後續處理，接著我們將透過程式碼抓取字串，傳入 Arduino 中做設備控制。

```
(env) pi@raspberrypi:~/assistant-sdk-python/google-assistant-sdk/google
assistant/grpc $ python3 pushtotalk.py
INFO:root:Connecting to embeddedassistant.googleapis.com
INFO:root:Using device model reco-950af-vending-71z6ke and device id 5e
d6-11e9-875a-b827ebbf6f34
Press Enter to send a new request...
INFO:root:Recording audio request.
INFO:root:Transcript of user request: "right".
INFO:root:Transcript of user request: "flights".
INFO:root:Transcript of user request: "flights".
INFO:root:End of audio request detected.
INFO:root:Stopping recording.
INFO:root:Transcript of user request: "flights".
INFO:root:Playing assistant response.
INFO:root:Finished playing assistant response.
Press Enter to send a new request...
```

圖 8. 語音辨識結果

```
self.conversation_stream.stop_recording()
if resp.speech_results:
    logging.info('Transcript of user request: "%s"',
                '\n'.join(r.transcript
                          for r in resp.speech_results))
    str = '\n'.join(r.transcript for r in resp.speech_results)
    if (str == 'right'):
        ser.write(b"R")
        sleep(0.5)
    elif str == 'left':
        ser.write(b"L")
        sleep(0.5)
if len(resp.audio_out.audio_data) > 0:
    if not self.conversation_stream.playing:
        self.conversation_stream.stop_recording()
        self.conversation_stream.start_playback()
        logging.info('Playing assistant response.')
        self.conversation_stream.write(resp.audio_out.audio_data)
```

圖 9. 擷取字串控制 Arduino

4.2 販賣機硬體與控制

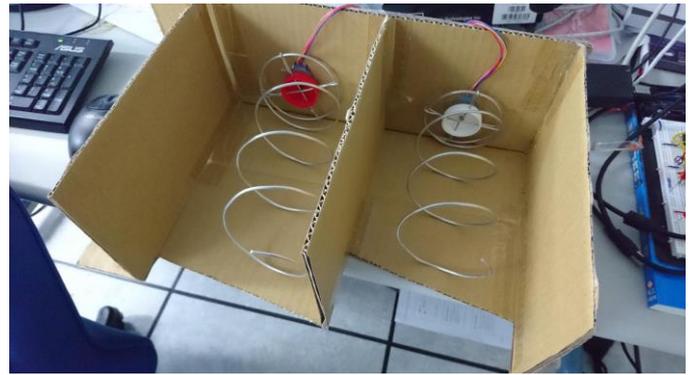


圖 10. 商品架

4.2.1 步進馬達

在使用步進馬達之前要先了解一個步進馬達的齒輪比，透過減速比與馬達步進角可以算出轉一圈總共需要幾步，以方便寫程式。

Rated voltage :	5VDC
Number of Phase	4
Speed Variation Ratio	1/64
Stride Angle	5.625°/64
Frequency	100Hz
DC resistance	50Ω±7%(25°C)
Idle In-traction Frequency	> 600Hz
Idle Out-traction Frequency	> 1000Hz
In-traction Torque	>34.3mN.m(120Hz)
Self-positioning Torque	>34.3mN.m
Friction torque	600-1200 gf.cm
Pull in torque	300 gf.cm
Insulated resistance	>10MΩ(500V)
Insulated electricity power	600VAC/1mA/1s
Insulation grade	A
Rise in Temperature	<40K(120Hz)
Noise	<35dB(120Hz, No load, 10cm)
Model	28BYJ-48 - 5V

圖 11. 步進馬達規格表

透過圖 11. 可以知道馬達的減速比是 1/64，步進角是 5.625 度/64，透過公式

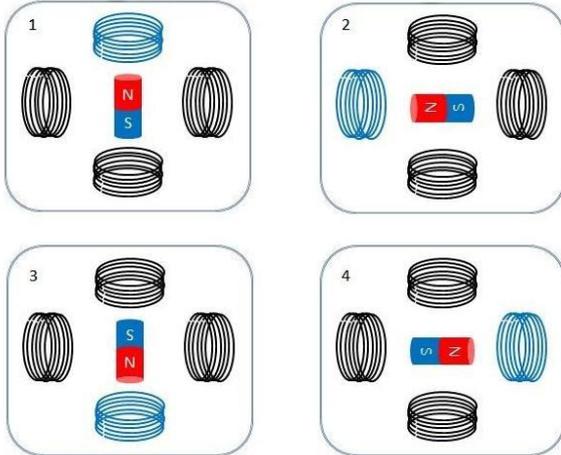
減速比 64:1 (輸入圈數：輸出圈數)

由上可知，若馬達步進角 5.625 度 再乘以減速比得到輸出的解析度：每步旋轉 $5.625 * 1/64 = 0.087890625$ 度

若要轉一圈，需要 $360 \text{ 度} / 0.087890625 \text{ 度} = 4096$ 步

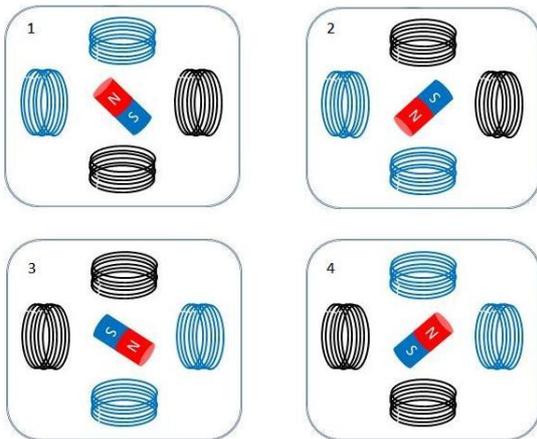
程式寫法:

(1)一相激磁(優點:省電 缺點:轉矩小、轉速慢)



一相激磁驅動法, 一圈需要2048步

(2)二相激磁 (優點: 轉矩大 缺點: 轉速慢、耗電)



二相激磁驅動法, 一圈需要2048步

圖 12. 一相激磁與二相激磁對照表

原本採用的是一相激磁的寫法, 在測試的時候有扭力不夠, 會有空轉的現象, 因此上網找了更多的寫法, 例如二相激磁, 一二相激磁的寫法, 最後選擇的是二相激磁的寫法, 如上圖, 二相激磁每次會輸出兩個訊號來控制馬達旋轉, 相對一相激磁的寫法, 轉速會比較慢, 不過轉矩大, 足夠帶動商品架使用的螺旋線圈, 因此使用二相激磁的寫法。

```
for(int i=0; i<512; i++){
digitalWrite(11,1);
digitalWrite(10,1);
digitalWrite(9,0);
digitalWrite(8,0);
delay(t);

digitalWrite(11,0);
digitalWrite(10,1);
digitalWrite(9,1);
digitalWrite(8,0);
delay(t);

digitalWrite(11,0);
digitalWrite(10,0);
digitalWrite(9,1);
digitalWrite(8,1);
delay(t);

digitalWrite(11,1);
digitalWrite(10,0);
digitalWrite(9,0);
digitalWrite(8,1);
delay(t);
}
```

圖 13. 二相激磁程式寫法

如圖 13, 使用二相激磁驅動法, 一圈需要 2048 步, 執行一次迴圈的時候會走 4 步, 所以迴圈走 512 次就能達到 2048 步, 也就是轉一圈的效果。

4.2.2 照度計模塊 GY-30

```
#include <Wire.h>
#include <BH1750.h>

BH1750 lightMeter;

void setup(){

Serial.begin(9600);
lightMeter.begin();
Serial.println(F("BH1750 Test"));
}

void loop() {

uint16_t lux = lightMeter.readLightLevel();
Serial.print("Light: ");
Serial.print(lux);
Serial.println(" lx");
delay(1000);
}
```

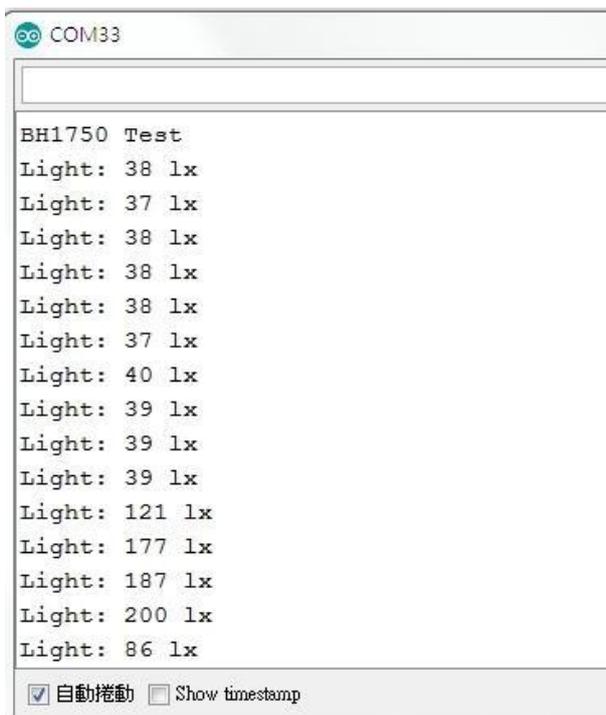


圖 14. GY-30 程式與亮度數據

在室內燈光下，光的亮度大概介於 30-40 多 lx 左右不等，視燈光亮度不等，在手電筒照射下可以到達 100-200 多，透過讀取光強度 lx 的數據，在強光的時候選擇不開啟 LED 燈，而在低光源的情況下，選擇自動打開 LED 燈以便照明，以達到販賣機在晚上也能營業的效果。

4.2.3 LED 燈條



圖 15. LED 燈條

5050LED 燈條主要有三個輸入分別是 R、G、B，這三個訊號源代表著光的三

原色 RGB，透過改變 RGB 訊號源可以，可以產生不同的顏色。如圖 16，RGB 的訊號在 0-255 之間，透過 RGB 訊號的增減可以改變輸出的顏色。



圖 16. 取餐櫃

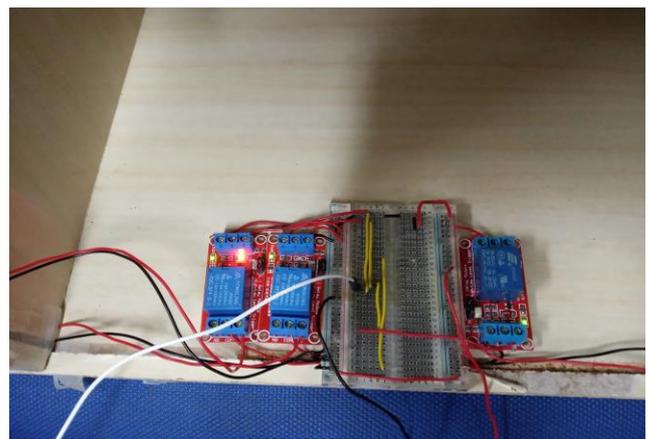


圖 17. 繼電器

使用 PHP 傳參數到 arduino 來控制要開啟哪個櫃子，被開啟的櫃子設為低電位，電池鐵便沒有磁力，即對應櫃子解鎖。

4.2.4 手機及網頁



透過網路大數據可以收集消費者的消費習慣，未來發展還是相當有競爭力的。

參考文獻

[1] 販賣機歷史

[https://zh.wikipedia.org/wiki/Wikipedia:
%E9%A6%96%E9%A1%B5](https://zh.wikipedia.org/wiki/Wikipedia:%E9%A6%96%E9%A1%B5)

[2] 步進馬達參考資料

[https://sites.google.com/site/csjhmaker/d-
dong-li-pian/28byj-48-bu-jin-ma-da](https://sites.google.com/site/csjhmaker/d-dong-li-pian/28byj-48-bu-jin-ma-da)

[3] LED 燈參考資料

[https://makezine.com/projects/android-
arduino-led-strip-lights/](https://makezine.com/projects/android-arduino-led-strip-lights/)

[4] 照度計模塊 GY-30

[http://ee543.blogspot.com/2016/12/arduino-
uno610gy-30_2.html](http://ee543.blogspot.com/2016/12/arduino-uno610gy-30_2.html)

[5] Google Assistant API

<https://developers.google.com/assistant>

五、結論與建議

當初在想這個題目的時候，我們曾經考慮販賣機是要買現成的抑或自己開發製作。在看了學長的智慧回收系統以後，發現學長有用到步進馬達，市面上的販賣機，也是透過類似的原理達到出貨的效果，因此我們決定自己動手做。馬達方面。自動販賣機一直都是歷久不衰的發明，在未來發展方面，透過物聯網可以將販賣機升級，能加載更多的功能，例如聲控或臉部辨識等功能，甚至