

中華民國第 62 屆中小學科學展覽會

作品說明書

高級中等學校組 電腦與資訊學科

佳作

052501

IoT 智慧取貨機

學校名稱：基隆市立八斗高級中學

作者： 高二 林苙紋 高二 賴禹孜	指導老師： 葉育睿
-------------------------	--------------

關鍵詞：Arduino、IoT、App Inventor 2

摘要

現在的超商取貨機制存在著很大的時間成本，一次取貨要是花費數分鐘在找貨，那長久下來會造成大量時間的消耗，因此為了提高領或效率，我們設計了一台裝置來減少取貨的過程之中，因人為造成的時間耗損，並降低領錯貨物的可能性。

而我們也可以依據存放貨物種類及應用方式而改造，製作出適合不同物品的存取裝置，像是用於書籍的存放，讓使用者可以自行借閱或歸還，藉由改造裝置，普及於各種類型貨物的存放，為我們的生活增加便利性。

除了擴大使用方式，增添提領貨物的提示，為貨物增設設定，關於提升整體實用性的部分也有待我們進行更進一步的探討。

壹、前言

一、研究動機

平常在便利超商取貨的時候，通常都是報後三碼，確認姓氏，接著店員就會去倉庫找我們的貨物，有的時候很快就能找到，但通常會需要花上一小段時間。如果我們能設計出一項方便取件機器，就可以增加店員的工作效率及顧客的取貨便利性。

二、研究目的

設計一台方便的裝置，幫助我們將取貨所花費的時間節省下來，顧客不須因店員花費時間翻找貨物而等待，而店員也不須因為要查找貨物而無法照顧到後面的客人，而導致店內商品被竊取。

三、文獻回顧

整體而言，我們結合了 IoT 與倉儲管理系統，讓貨物資料透過 IoT 傳入。在 Google sheets 先進行一個初步的整理，再讓 Arduino Uno Wi-Fi 藉由 Google Apps Script 取得新增的資料，在程式中進行資料的記載，並將須進行的資料以 Inter-Integrated Circuit (後稱 I²C) 傳遞給其他的 Arduino 進行分工。

主要採用了倉儲管理系統中放置位置的記憶方式，以目前的規模而言，已經足夠將對應的貨物分別處理。

貳、 研究設備及器材

名稱	功能
Arduino Uno	微電腦操作
Arduino Mega 2560	微電腦操作
Arduino Uno Wi-Fi	微電腦操作
傳輸線	傳輸程式至 Arduino
杜邦線	試做線路連接
麵包板	試做線路連接
電烙鐵	焊接器具
錫	焊接材料
單芯線	線路連接
多芯線	線路連接
電路板	焊接材料
LCD 顯示模組	顯示文字
Google Apps Script	App Inventor 2 資料分類，並將其傳送至 Arduino Uno Wi-Fi
CorelDRAW(軟體)	設計雷雕圖檔
RDworks	雷雕機應用軟體
雷雕機	切割工具
3mm 密集板	料材
壓克力板	料材
木工膠	組裝用材料
熱熔膠	組裝用材料
螺絲	組裝用材料
螺母	組裝用材料
履帶	動力傳導
輪子	移動定位
減速馬達	提供動力
手機	使用 App Inventor 2 所製作的 APP
L9110S	減速馬達控制模組

參、研究過程或方法

一、裝置構思

確立主題後，先進行小組討論，起初只打算製作類似於自動販賣機的裝置，再對其進行修改，但發現由 Arduino 所設計的自動販賣機早已被研發出來，因此我們打算加入補貨端可以將貨自動放入的功能，並添加手機取進貨資料輸入的設計。

藉由 App Inventor 2 讀取資料，傳送至 Google 表單，並藉由 Google Apps Script 編寫程式碼，來對資料進行分析，再將取得的資料傳給 Arduino Uno Wi-Fi，以 I²C 的方式將指令傳達給其他 Arduino 分工處理。

二、文獻探究

對於我們所發想的這種概念，其中的系統，在世界各地，有許多都發展了一段時間，以下會列出本設備最具代表性的兩項，物聯網(IoT)及倉儲管理系統(WMS)。

(一) 物聯網

物聯網(英文：Internet of Things，IoT)，起源於1999年，是 Ashton 教授在研究 RFID 時，提出 RFID 結合網路架構，定義為：**將感測設備與網際網路連接起來，實現智慧化的識別和管理**。2005年，國際電信聯盟，將其命名為 Internet of Things，正式提出其架構，物聯網則成為現代趨勢。

架構分為三層，感知、網路及應用。感知如同人類眼睛，是用來接收資料；網路如同人體神經，是用來傳送資訊；應用如同人類大腦，將資訊統整應用。目前對於嵌入式系統的發展更為顯著。

對我們來說，物聯網會使我們的效率提升，自動收集數據，既節省金錢，也節省空間。但相對來說，不易發現其是否損壞，安全強化不易，因此易遭受駭客攻擊，產生資安漏洞。

(二) 倉儲管理系統

倉儲管理系統(英文：Warehouse Management System，WMS)，可分為全自動、半自動、全人工，以上都必須含有一個倉儲管理系統軟體，並具有資料收集、批次管理及存取功能。

目前常常提到的亞馬遜公司 (Amazon)，是最早開啟電子商務的公司，只要在網上下訂，自動化倉儲便會開始準備商品並送到你家，其倉儲有許多 Kiva (橘紅色機器人)，外型類似於掃地機器人，它們到對應的貨架，將貨架抬起，

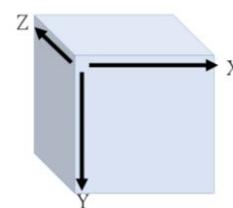
並送至職員旁，讓職員揀選、掃描條碼並包裝。四層樓高工廠中還有大型的機械手臂，將大型存貨抬至其他樓層。

倉儲管理其中的商品要包含入倉時間、時效、重量、放置位置、擺放方式，並對商品進行劃區及標示。相同或相似放在一起方便找尋，如果是販售相同商品，因為有變質及有效期限的問題，因此要以先進先出的概念來出售商品。

三、設計結構

(一) X-Y-Z 平台

皆採用減速馬達控制，並依照附圖 1 所配置的方式設定 X-Y-Z 的方向。我們將 X 軸設計成車子的形狀，並以軌道及齒輪對尺條加以穩固 X 軸，用以確保 X 軸會依著我們所想的方向移動。Y 軸的部分我們採用齒輪對履帶去帶動上下移動，將馬達置於下方，避免 Y 軸會有傾倒的可能，在 Y 軸上又設計軌道，以穩固裝在 Y 軸履帶上的 Z 軸。而 Z 軸我們將它設計為類似於手臂的樣子，讓它可以伸長及收回，伸出的部分則設計成 \sqcap 字型，用來將貨櫃上的貨物取出。



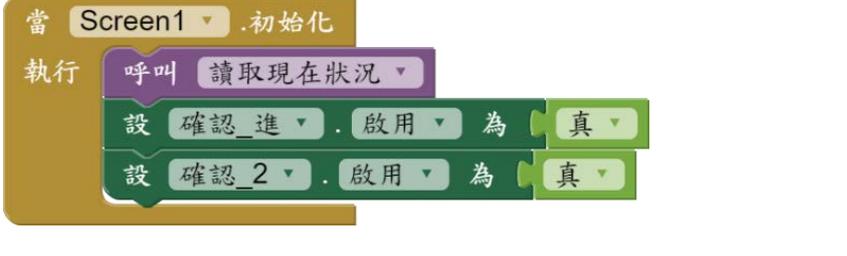
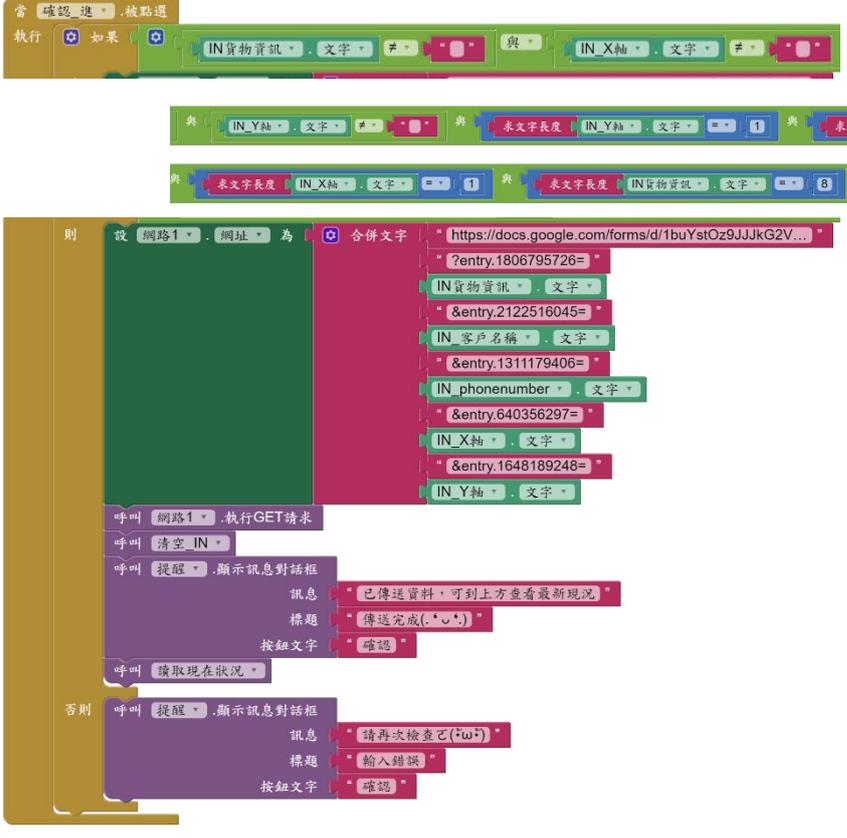
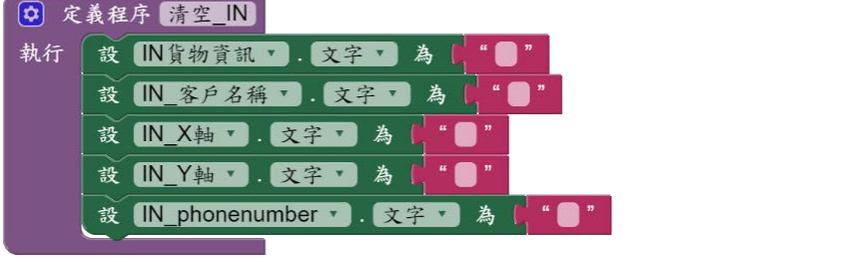
附圖1. XYZ 的配置

(二) 資料傳輸及處理

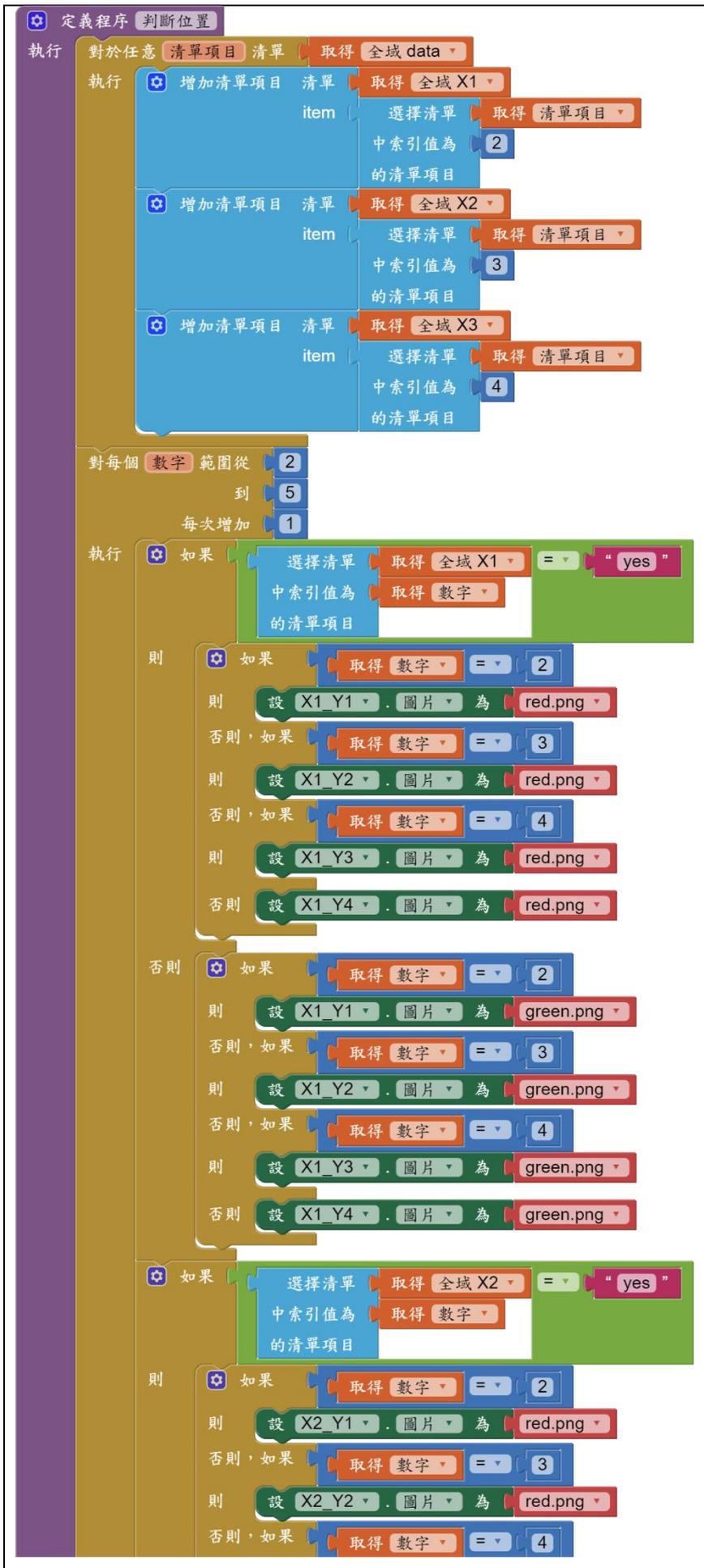
我們利用 App Inventor 2 來進行手機程式的設計，將其進行排版及美化，並在程式的部分進行基礎的資料判定，確保資料長度並確定指定的存放位置是空的，不符合的資料則會清空並發出提醒，告知使用者資料有誤，而符合的資料則會利用回復 Google Forms 的方式，並利用 Google Apps Script 將資料整理並放置於 Google Sheets 之中，當資料放置好時，Arduino Uno Wi-Fi 再從 Google Sheets 取得須進行的移動，而 Arduino Uno Wi-Fi 再將這些資料以 I²C 的方式，傳遞給其他 Arduino 負責。

四、程式設計

(一) App Inventor 2

圖示	功能
	<p>螢幕刷新時，執行「讀取現在狀況」，並讓兩個確認鍵呈現可被點選的狀態。</p>
	<p>點選刷新現況，執行「讀取現在狀況」。</p>
	<p>針對進貨的輸入進行處理，資料不符則跳出提醒告知使用者。資料符合則將資料送至 Google Forms，並清空儲存格，呼叫提醒告知已傳送。</p>
	<p>清空進貨輸入框的所有內容。</p>

	<p>清空出貨輸入框的所有內容。</p>
	<p>針對出貨的輸入進行處理，資料不符則跳出提醒告知使用者。資料符合則將資料送至 Google Forms，並清空儲存格，呼叫提醒告知已傳送。</p>
	<p>取得 Google Sheets 位置配置的 csv 檔。</p>
	<p>設定變數儲存從 Google Sheets 得到的資料。</p>
	<p>將取得的 csv 檔轉為陣列，並執行「判斷位置」，處理結束後，跳出對話框，提示已更新</p>



「判斷位置」

先將資料進行分類，先將同一行放置於同一清單中，並針對每個位置的資料進行判斷，將資料為「yes」

(在 Google Sheets 中，具備貨物的會顯示「yes」) 的位置，改變圖片為「red」(有貨物存在)，其他則設為「green」(無貨物存在)。

(接續上一頁)

The image shows a Scratch script with the following logic:

- Else, if** (取得 數字) = 4:
 - Then** 設 X2_Y3 . 圖片 為 red.png
 - Else** 設 X2_Y4 . 圖片 為 red.png
- Else** **if** (取得 數字) = 2:
 - Then** 設 X2_Y1 . 圖片 為 green.png
 - Else, if** (取得 數字) = 3:
 - Then** 設 X2_Y2 . 圖片 為 green.png
 - Else, if** (取得 數字) = 4:
 - Then** 設 X2_Y3 . 圖片 為 green.png
 - Else** 設 X2_Y4 . 圖片 為 green.png
- if** 選擇清單 中索引值為 的清單項目 (取得 全域 X3) = "yes" (取得 數字):
 - Then** **if** (取得 數字) = 2:
 - Then** 設 X3_Y1 . 圖片 為 red.png
 - Else, if** (取得 數字) = 3:
 - Then** 設 X3_Y2 . 圖片 為 red.png
 - Else, if** (取得 數字) = 4:
 - Then** 設 X3_Y3 . 圖片 為 red.png
 - Else** 設 X3_Y4 . 圖片 為 red.png
 - Else** **if** (取得 數字) = 2:
 - Then** 設 X3_Y1 . 圖片 為 green.png
 - Else, if** (取得 數字) = 3:
 - Then** 設 X3_Y2 . 圖片 為 green.png
 - Else, if** (取得 數字) = 4:
 - Then** 設 X3_Y3 . 圖片 為 green.png
 - Else** 設 X3_Y4 . 圖片 為 green.png

(二) Arduino Uno Wi-Fi

內容	說明
<pre>char sth[11]='\0'; char First ='\0'; char LocateX ='\0'; char LocateY ='\0'; char contant[8]='\0';</pre>	變數設定(全域)
<pre>First=sth[0]; LocateX=sth[1]; LocateY=sth[2]; for(int i=0;i<8;i++){ contant[i]=sth[i+3]; }</pre>	變數 sth 為由 Google Apps Script 取得的資料，將其內容進行分類。 First 儲存進出貨的判斷 LocateX 儲存 X 座標 LocateY 儲存 Y 座標 contant 儲存貨物資料
<pre>if(strlen(contant)!=8&&LocateX!='\0'&&LocateY!='\0'){ if(First=='i'){ if(in(contant ,change(LocateX) ,change(LocateY))==1) moving(First,LocateX ,LocateY); } if(First=='o'){ if(out(contant ,change(LocateX) ,change(LocateY))==1) moving(First,LocateX ,LocateY); } LocateX ='\0'; LocateY ='\0'; memset(contant, 0, 8);</pre>	當資料具備，進行處理，先判別進出貨，再將給予副程式判別資料正確與否，正確的會儲存並回傳 1。當副程式回傳 1，進入副程式 moving。執行結束後，清空暫存資料。 (在 void loop()中)
<pre>void moving(char io,char x,char y){ Wire.beginTransmission(2); //開始傳送，指定給從端2號 (LCD+出口) Wire.write("U"); //給定指令_LCD 顯示 Using Wire.endTransmission(); //結束通訊</pre>	

```
Wire.beginTransaction(1); //開始傳送，指定給從端1號 (xyz)
Wire.write(io+x+y); //給定指令_出或進
Wire.requestFrom(1,4); //讓從端1號 (xyz)回傳
while(Wire.available()){
    Serial.print(Wire.readString());
} //接收回傳"done"並輸出
Wire.endTransmission(); //結束通訊
Wire.beginTransaction(2); //開始傳送，指定給從端2號 (LCD+出口)
Wire.write("S"); //給定指令_LCD 顯示 Standby
Wire.endTransmission(); //結束通訊
}
```

(三) Arduino Uno

內容
<pre>#include <Wire.h> //I²C 函式庫 #include <SoftwareSerial.h> //序列埠函式庫 #include <motoLiquidCrystal_I2C.h> //motoduino 的 LCD 模組函式庫 LiquidCrystal_I2C mylcd(0x27,16,2); //初始設定 void setup(){ Wire.begin(2); //I²C 位址與連線設定，這裡設定為2號 Serial.begin(9600); //Serial 速率為9600 //LCD mylcd.init(); mylcd.backlight(); mylcd.clear(); mylcd.setCursor(0,0); mylcd.print("Standby"); Wire.onReceive(receiveEvent); //當收到 I²C 訊號時，啟動的動作 } void loop(){ delay(100); //延遲0.1秒 } //自定義的接收後的啟動副程式 void receiveEvent(int numBytes){ while(Wire.available()){ //判斷 Wire.available()有沒有訊號 char c = Wire.read(); //將傳入的訊號 Wire.read()指定給字串 C Serial.print(c); //透過 Serial 印出字串 C 的內容 if(c=='S'){ //收到指令為 S，顯示 Standby</pre>

```
mylcd.clear();  
mylcd.setCursor(0,0);  
mylcd.print("Standby");  
}  
if(c=='U'){ //收到指令為 U，顯示 Using  
    mylcd.clear();  
    mylcd.setCursor(0,0);  
    mylcd.print("Using");  
}  
}  
}
```

(四) Arduino Mega

內容
<pre>#include <Wire.h> //I²C 函式庫 #include <SoftwareSerial.h> //序列埠函式庫 #include <stdio.h> #include <stdlib.h> #include <string.h> //C 函式庫 //變數設定 char sth[3] = {'\0'}; //I²C in int LocateX = -1; //X 軸 int LocateY = -1; //Y 軸 //字串轉數字 int change(char what){ int ans = atoi(&what); return ans; } //setup void setup(){ Serial.begin(9600); //Serial 速率為9600 Wire.begin(1); //I²C 位址與連線設定，從端 1 號 Wire.onReceive(receive); //當收到 I²C 訊號時，啟動的動作 } //loop void loop(){ delay(100); } void receive(){ if(Wire.available()){ //判斷主端有沒有傳送可接收訊號 strcpy(sth, Wire.readString().c_str()); //將主端傳入的訊號指定給 sth } }</pre>

```
LocateX = change(sth[1]);
LocateY = change(sth[2]);
if(LocateX!=-1 && LocateY!=-1){
    if(sth[0]=='o'){ //進出貨判斷
        move_in(LocateX ,LocateY); //進貨副程式
    }
    else if(sth[0]=='i'){
        move_out(LocateX ,LocateY); //取貨副程式
    }
}
LocateX = -1;
LocateY = -1;
Wire.write("done",4);
}
}
```

五、測試功能

我們將整體分為四大項，分別是 App Inventor 2 對 Google Sheets 的資料傳遞與整理、Google Sheets 對 Arduino Uno Wi-Fi 的資料傳遞、Arduino Uno Wi-Fi 對其他 Arduino 的資料傳遞以及 X-Y-Z 的移動，而後再分細項去分開測試。

(一) App Inventor 2 對 Google Sheets 的資料傳遞與整理

1. App Inventor 2 的資料判定

先確定所製作的程式能夠辦定出不符合的資料，並發送提醒，告知使用者資料錯誤。

2. App Inventor 2 將資料傳送至 Google Forms

我們利用網址去回覆 Google Forms，一開始是採用學校帳號製作，但是第一版無法順利回覆 Google Forms，後來換成個人帳號測試，相同的方式卻可以取得回覆，因此我們推測初次的失敗是因為學校帳號的設定問題，導致這項功能無法進行。

3. 利用 Google Apps Script 將回覆的資料整理至 Google Sheets

先初步學習了 Google Apps Script 編寫用的語法，參考了 Google 所提供的程式使用說明，先以直觀的方式撰寫了初步的程式，此時總行數為 166 行，並且確定此程式可運作，而後找尋程式的邏輯，並將程式總行數從 166 行降低至 70 行，不但縮短了程式，更將每段程式的作用，註記於後方，以便往後增修功能時，能夠快速掌握先前的內容。

(二) Google Sheets 對 Arduino Uno Wi-Fi 的資料傳遞

利用 Arduino 官方的範例程式碼，加上於網路上查詢的資料，撰寫 Google Sheets 和 Arduino Uno Wi-Fi 的程式碼，在 Google Sheets 上撰寫讀取資料的程式碼，在 Arduino Uno Wi-Fi 先做讀取的測試，確定可以讀到對應資料。

(三) Arduino Uno Wi-Fi 對其他 Arduino 的資料傳遞

採用 I²C 的方式，讓 Arduino Uno Wi-Fi 連接其他 Arduino，並分別指派任務，會按照順序顯示資料。

(四) X-Y-Z 的移動

1. 齒輪與齒條的咬合度及馬達扭力

原先採用了步進馬達，但因為其自身扭力無法突破最大靜摩擦力，而後又嘗試 360 度伺服馬達，但仍舊有些會有無法突破最大靜摩擦力的問題，因此最後更換為減速馬達。

將齒輪固定於減速馬達上，並固定尺條，將減速馬達兩側焊接出兩條線，分別接於 Arduino 上的 GND 及 5V，使減速馬達轉動，讓在減速馬達上的齒輪對齒條施力，並確定馬達具備足夠扭力，能使齒輪的運動突破最大靜摩擦力。

2. 齒輪與履帶的咬合度及馬達扭力

與前項測試類似，將齒輪固定於減速馬達上，並將履帶卡在齒輪上，在另一側放置一個可轉動無馬達的齒輪，將減速馬達兩側焊接出兩條線，分別接於 Arduino 上的 GND 及 5V，使減速馬達轉動，讓在減速馬達上的齒輪可以帶動履帶，確定馬達具備足夠扭力，能使齒輪的運動突破最大靜摩擦力，並可正轉逆轉使履帶上下運動。

3. 手臂的伸縮

將手臂固定於減速馬達上，把減速馬達兩側焊接出兩條線，分別接於 Arduino 上的 GND 及 5V，使減速馬達轉動，讓在減速馬達上的手臂能夠伸縮，並確定最長及最短的伸長角度。

六、修訂並焊接

在確定前面的各項功能都可使用時，我們將其稍作修改，使須連接的部分可以接在一起，調配重心，避免再運作過程中傾倒，並整理線，確保不會被線卡到，而無法運作，而後就將以確定的部分焊接在一起，以免在運行時，線沒插好而跑掉無法使用。

肆、 研究結果

一、 App Inventor 2

(一) 客戶端

1. 配色

採用柔和的色調，讓使用者看了能較為舒適，提示字採用醒目的紅色，使其於整體畫面中較為突出，讓使用者能夠注意到文字內容的提醒。(如附圖 4)

2. 輸入內容的格式提醒

在需輸入內容的格子內加入格式提醒，避免使用者不清楚需輸入的內容。(如附圖 4)

3. 傳輸提示

在輸入內容並按確認送出時，會先進行辨別文字格式，不符合者會跳出提醒，符合者會告知傳送成功。(如附圖 2 及附圖 3)



附圖2. 成功傳輸的提示



附圖3. 資料錯誤的提示



附圖4. 客戶端的畫面配置

(二) 服務端

1. 配色

採用柔和的色調，並讓使用者可以利用背景顏色去區分進貨及取貨，提示字採用紅色，使其於整體畫面中較為突出，讓使用者能夠注意到文字內容的提醒。(如附圖 8)

2. 輸入內容的格式提醒

在需輸入內容的格子內加入格式提醒，避免使用者不清楚需輸入的內容。(如附圖 8)

3. 傳輸提示

在輸入內容並按確認送出時，會先進行辨別文字格式，不符合者會跳出提醒，符合者會告知傳送成功。(如附圖 5 及附圖 6)

4. 現況配置圖

以綠色底的圈圈(如附圖 7 上方 X2_Y1 的圖示)表示該儲存格是可放入貨物的，用紅色底的叉叉(如附圖 7 上方 X1_Y1 的圖示)表示該儲存格已有存放的貨物，不可再放入貨物，點選下方的「刷新現況」，可以取得最新現況，在刷新成功時，會跳出提示，告知使用者已更新。



附圖5. 成功傳輸的提示



附圖6. 資料錯誤的提示



附圖7. 資料更新的提示

附圖8. 服務端的畫面配置

二、裝置

(一) 外觀

整體機器外觀的正面如附圖 9 所示。

右側會有一個 LCD 顯示模組，在無人使用時，會顯示「Standby」，運作時會顯示「Using」。

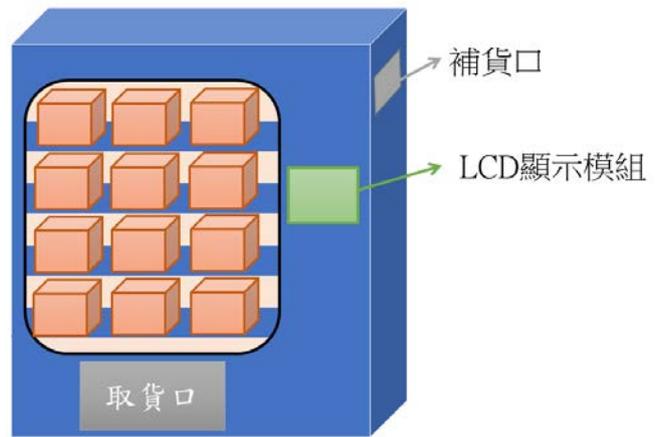
(二) 內部配置

附圖 9 中，右上角的補貨口，將貨物放入後會位於附圖 10 中的左上角，而我們的物品移動及放置，採用了 X-Y-Z 的移動方式。

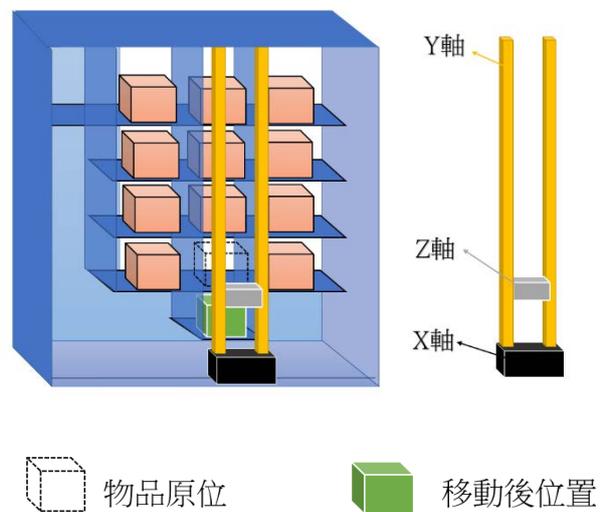
X 軸為移動小車加上齒輪對尺條的穩定，採用了單邊的減速馬達，車子底部為了使減速馬達的平面能剛好卡在底板上，因此將減速馬達上突出處所對應的位置開洞。

將 Y 軸放大檢視，會如同附圖 11 所顯示。上下各放置兩個齒輪，採用了雙邊的減速馬達，並將減速馬達放置於下方，使其連接兩個小齒輪，讓兩個小齒輪帶動下方的大齒輪，增加整體扭力。

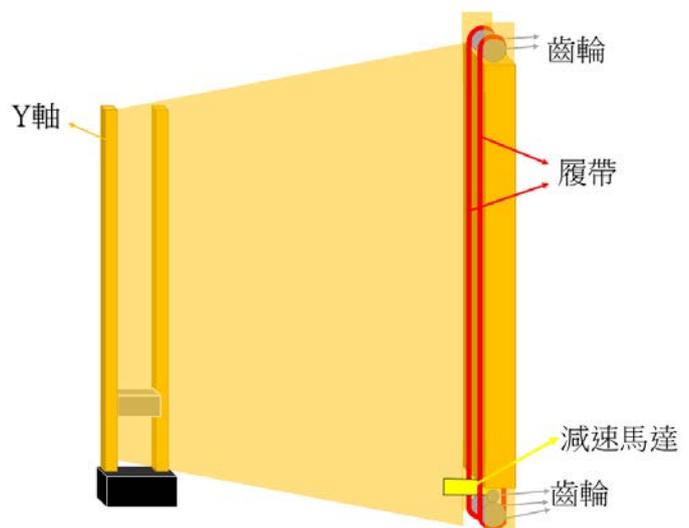
再利用履帶連接上下的齒輪，操控減速馬達的轉向讓履帶正轉逆轉，在履帶的左右兩側各放置了兩條軌道以穩定 Z 軸，並將兩段中的其中一段履帶固定於 Z 軸上，讓 Z 軸可以跟隨履帶移動。



附圖9. 機器外觀



附圖10. 機器內部

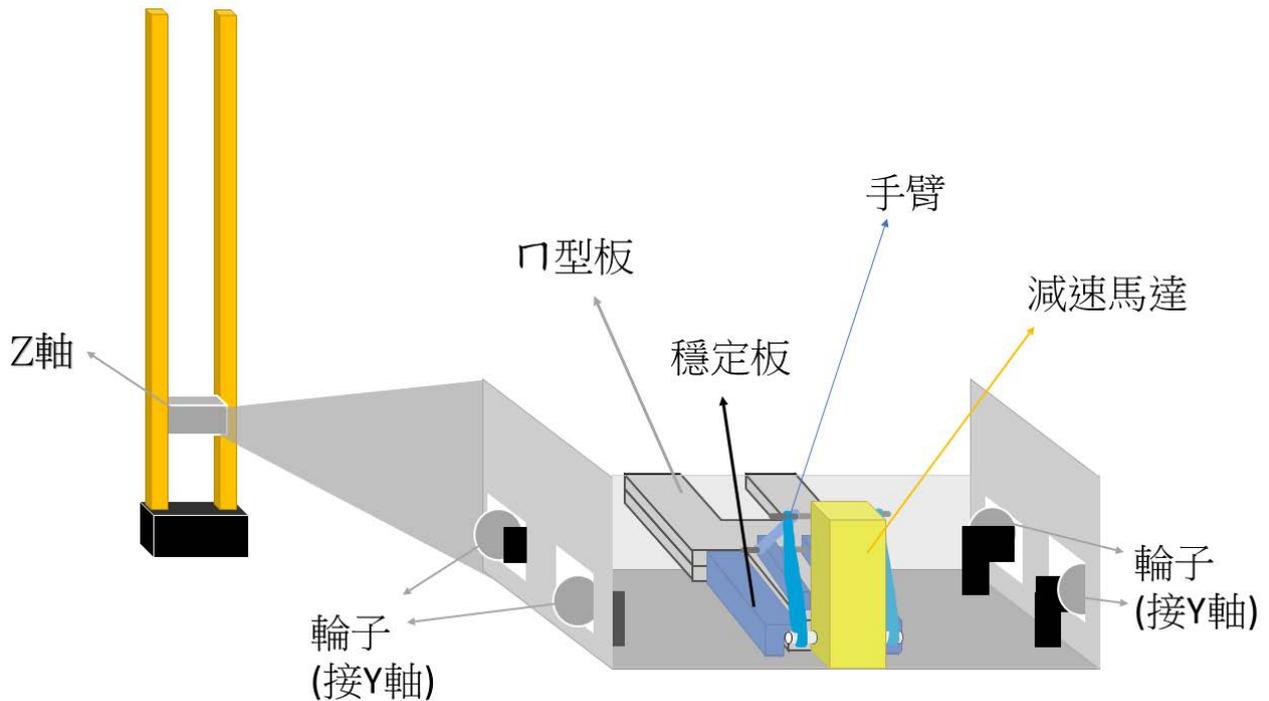


附圖11.Y 軸設計

Z軸的設計如同附圖 12 所示。我們利用了穩定板，來使承載貨物的 Γ 型板不會因為乘載了貨物的重量，而有傾倒的可能。

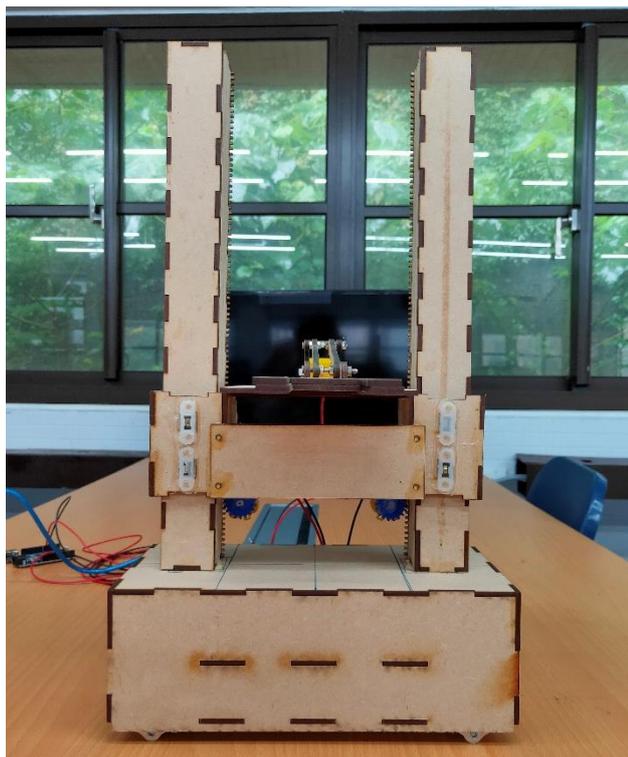
手臂的固定採用了長螺絲，使兩側的手臂利用相同的螺絲接在一起，確保左右兩側手臂的伸出及收回能夠同步運行。

在左右兩側都會將履帶固定在板子上，以利用履帶去進行上下運動，而左右兩側的輪子是為了使接在 Y 軸上能夠較為穩定，如果只使用單邊會較為不穩固，因此選擇製作了四個輪子去穩定與 Y 軸的連結。

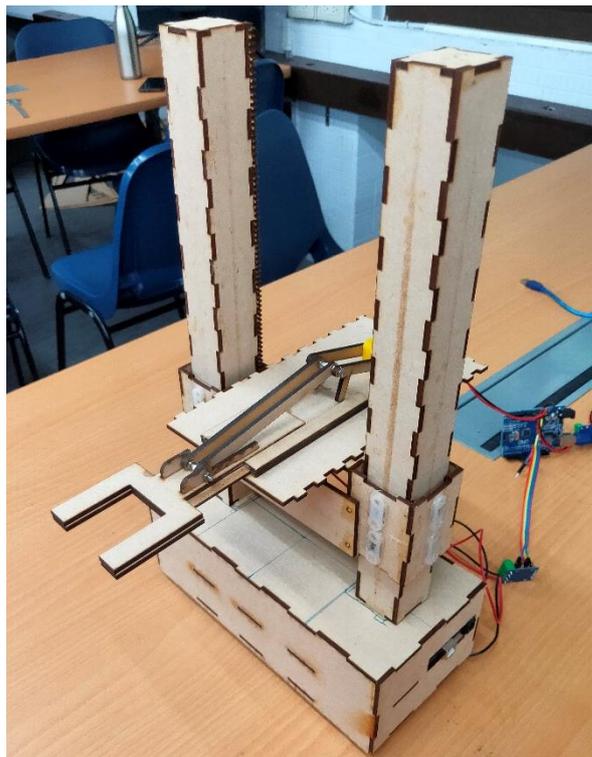


附圖12. Z 軸設計

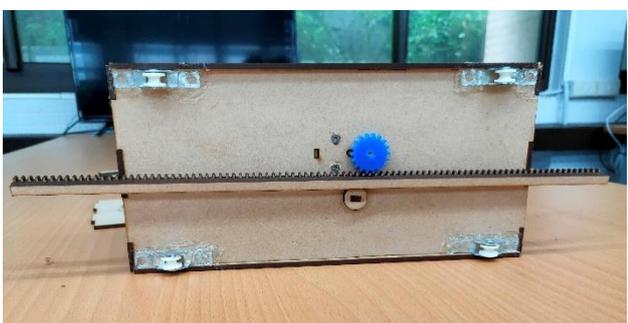
(三) 實際照片



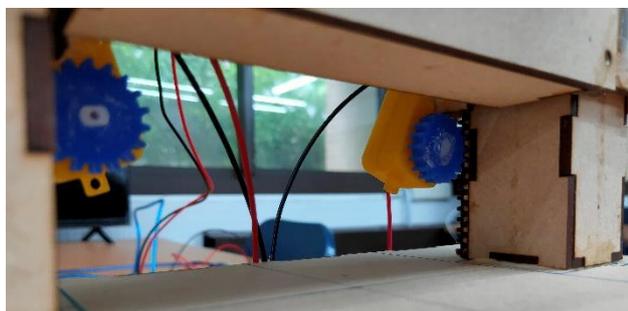
附圖13. XYZ 機構正視圖



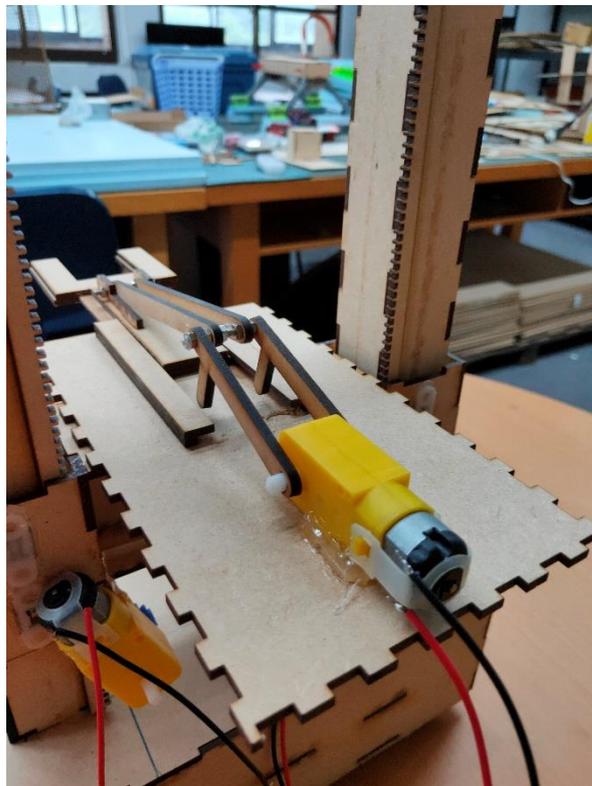
附圖14. XYZ 機構側視圖



附圖15. X 軸底部傳動機構



附圖16. Y 軸上升下降機構



附圖17. Z 軸伸縮取物機構

伍、 討論

一、有無其他方法增加實用性？

(一) 對於客戶

1.製作 LINE 的機器人

因考量到有些人的手機存取空間有限，無法足夠空間裝載新的手機程式，因此可利用大多手機必備的通訊程式 LINE 製作機器人，讓客戶能夠在不額外加裝一個新的程式之下，也能以回覆機器人的方式取得貨物。同時，使用 LINE 機器人也可以提醒客戶該去取貨了。

2.加入 QR code 的功能

因考量到方便性，有些顧客會請親友協助代領貨物，在這種情況下無論是利用 LINE 或手機程式，都會有些不方便，但若可採用 QR code 去提領貨物，只需在機器掃描即可取貨，相較之下會方便一些。

(二) 對於補貨者

1.對貨物設定日期

考量在貨物超過期限時都需要將貨品退回，因此在紀錄資料時可以再加入時間，並為補貨者增加一鍵取出所有逾期貨物的功能，能夠不用花費時間查找資料的對應貨物。

2.加入 QR code 的功能

利用 QR code 可以減少輸入資料的時間，並且因為貼在貨物上的 QR code 會是對應貨物的，因此也減少了資料輸出錯誤的可能性，這部分可以利用 App Inventor 2 本身的判讀 QR code 的功能去製作。

3.加入自動進貨的功能

於頂端配置履帶，使物品可以先全部置於機器內，再等待機器運作並置於其對應位置，依照順序先進先放入對應儲存格，而資料傳輸部分，可搭配 QR code 分別讀取，並依順序傳輸。

(三) 並用 IoT 取貨和現場取貨

因為目前的功能只有 IoT 的應用，但考量到有些人較習慣傳統取貨，因此需要現場與雲端的功能可以同步，所以要加入互相傳資料的功能時，需要將已被取走的貨物資料利用 Google Apps Script 將位於 Google Sheets 的資料刪除，並且可

以利用顏色或距離來判定真實存在的貨物，與 Arduino Uno Wi-Fi 中的資料比較，並將資料給予 Google Apps Script 針對不同資料進行處理。

二、與現在機制的比較

（一）找尋貨物的速度

現在的機制主要是告知店員再由店員去找尋貨物，而依此方式找尋貨物會有兩種情況，一種是店員清楚的知道你的貨品在哪裡，另一種則是他還需要花上一點時間去翻找你的貨物，此時找貨的時間就會增加。但本作品是本身就會記憶物品的位置，因此只須告知需取出的貨物資訊，即可將貨物取出。

（二）取得正確貨物的機率

本作品因可記錄貨物資料，因此除非是貨物在放入的時候就放置錯誤，否則都可以取得正確的貨物。而現在的機制確認貨物是否正確都是靠顧客確認資訊，萬一顧客沒有仔細看清楚貨物的資訊，就會有取錯物品的可能性，此時取錯的貨物還會造成店員的困擾，他們須要先墊付金額等該貨物的訂購者到才可以取得墊付的費用，一旦該訂購者不去取貨，此時店員就只好自行吸收這個物品。

陸、 結論

此項作品為我們的生活增添不少便利性，不僅縮短了取貨時間，還降低在取貨過程中人為疏失的可能性。

此裝置可依據存放的物品進行設定，不僅可以用於超商取貨，還可另外改造成適合書籍存放取用的方式，使圖書館的借閱能夠更加便利，可以讓民眾於機器投入書籍刷借閱卡進行還書的動作，也可在現場刷借閱卡借閱機器內的書籍。關鍵原因在於 X-Y-Z 的裝置，由於採用的是 Arduino 所以整體的價格會較市售的便宜不少。而改裝只需要配合物品改變貨櫃及 Z 軸的手臂樣式即可，就整體而言，這項裝置的可塑性較高。

雖然目前的裝置尚未有如同討論中更進一步的設計，但期許我們之後將這些增加便利性的設計加入作品後，能夠成為一個新穎的裝置來幫助大眾，為我們的生活取得較大的利益。

柒、參考資料及其他

一、Jiun-Yan Shiau Ph.D。Warehouse Management System (WMS)。取自：

<http://www2.nkust.edu.tw/~sho/wms.htm>

二、MIC 資策會產業情報研究所 (2020)。【網購大調查系列一】行動下單急追 PC 呈五五波行動商務正式成為主流。取自：<https://mic.iii.org.tw/news.aspx?id=555>

三、楊惟雯 (2012)。物流寄情13：物聯網的發展趨勢。取自：

<https://nmart.pixnet.net/blog/post/47268694%E7%89%A9%E8%81%AF%E7%B6%B2%E7%9A%84%E7%99%BC%E5%B1%95%E8%B6%A8%E5%8B%A2>

四、Eugene Michaels(2020)。物聯網：定義、應用和風險。取自：<https://nordvpn.com/zh-tw/blog/wulianwang/>

五、黃柏堯 (2020)。連上網就叫 IoT？解析物聯網的定義跟發展趨勢。數位時代，智慧城市。取自：<https://www.bnext.com.tw/article/60182/iot-trend-omo>

六、黃柏堯 (2020)。引爆物聯網商機的最後一哩路。數位時代，物聯網。取自：

<https://www.bnext.com.tw/article/56263/iot-business-opportunity>

七、半工人 YZ (2018)。NodeMCU 教學 - 11：使用 NodeMCU 以及 Google script 寫入&讀取 Google sheets 資料。取自：<http://wyj-learning.blogspot.com/2018/02/nodemcu-google-script-google-sheets.html>

八、黑修斯 (2018)。Arduino I2C 練習(1)：使用兩個 Arduino 做溝通。取自：

<https://hugheschung.blogspot.com/2018/05/arduino-i2c1arduino.html>

九、Cooper Maa (2011)。多台 Arduino 間的通訊 - 透過 I2C #1。取自：

<http://coopermaa2nd.blogspot.com/2011/12/arduino-i2c.html>

十、Super169 (2015)。Arduino 之間的 I2C 通訊 (4) 由 master 向 slave 要求資料回傳。取自：

<http://arduino169.blogspot.com/2015/08/arduino-i2c-4-master-slave.html>

【評語】 052501

本作品以 IoT 實作模擬店家（超商）對於小型物品的進、取貨的流程，目的為加速服務取件客戶的時效。作品結合手機上的 APP 程式開發、IoT、sensor/actuator、減速馬達、和夾取手臂等多種技術，在開發和整合這些技術上具有相當深度。此系統只曾運作過一次，因為沒有進行多次操作實驗來量測系統的運作成功率，因此此系統的運作成功率和穩定性沒有探討，建議未來要進行這些實驗來探討這些議題。另外，此系統可考慮採用條碼或 QR code 自動辨識方式，如此可免除重複出入物品取貨所需資訊且增進系統自動化。

作品簡報

IoT智慧取貨機

電腦與資訊學科_高級中等學校組

前言

一、研究問題

觀察到超商取貨，耗時、費力、工作效率差、便利性較低。

二、研究目的

設計一台方便的裝置，幫助我們節省取貨時間，顧客不須因店員花費時間翻找貨物而等待，而店員也不須因為要查找貨物而無法照顧到後面的客人，或導致店內商品被竊取。



附圖1. 現今超商取貨流程

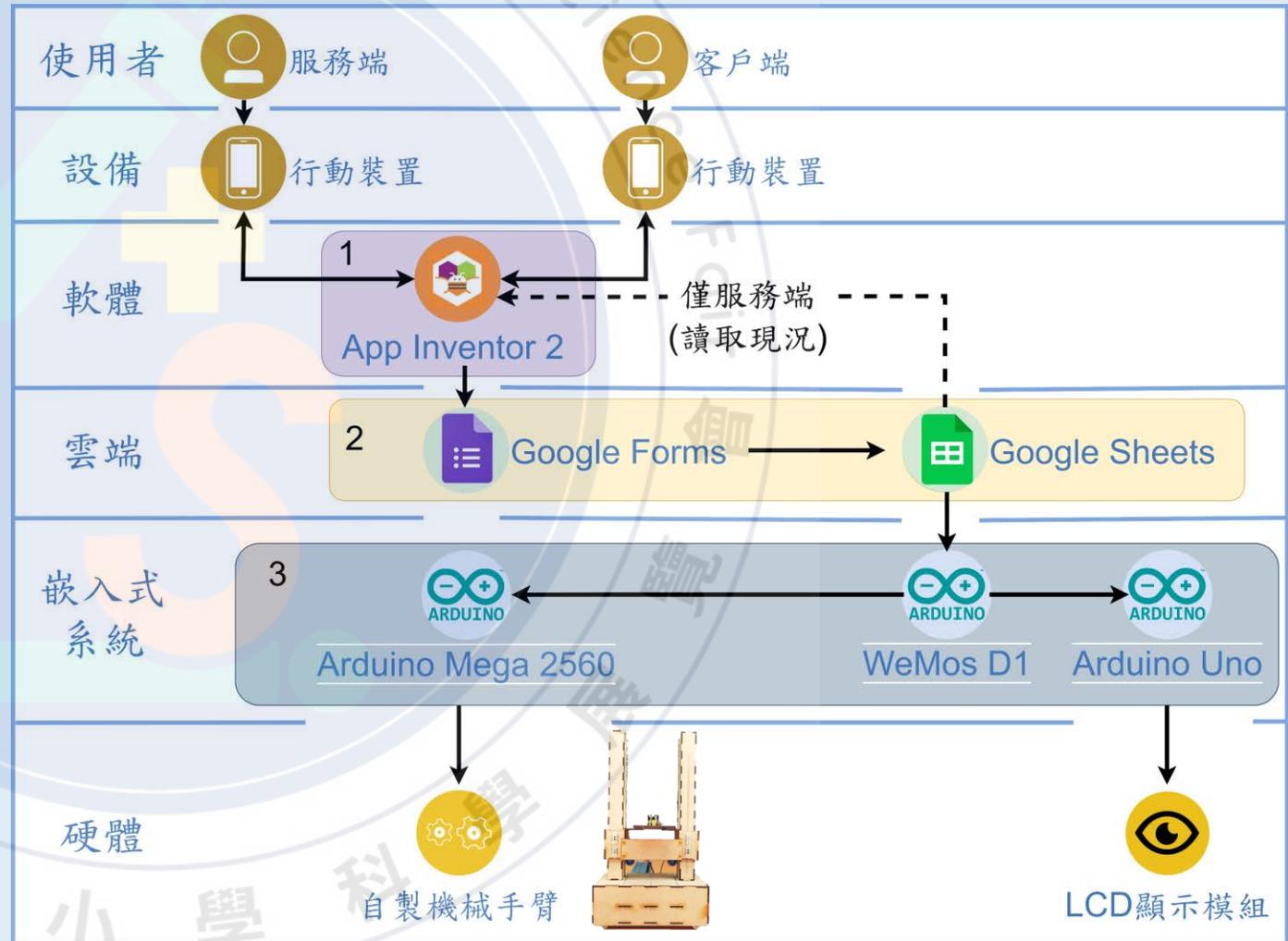
研究過程

一、裝置構思

藉由 App Inventor 2 (後稱AI2) 讀取資料，回覆 Google 表單，並藉由編寫 Google Apps Script (後稱GAS) 程式碼，來對資料進行分析，並將取得的資料傳給 Arduino Uno Wi-Fi，以 I²C的方式將指令傳達給其他 Arduino 分工處理。

二、設計結構

自製手臂採用減速馬達控制，X軸及Y軸採用齒輪對齒條，轉動齒輪控制齒條以移動，Z軸則是手臂，可伸縮取放貨物。



附圖2. 運行架構

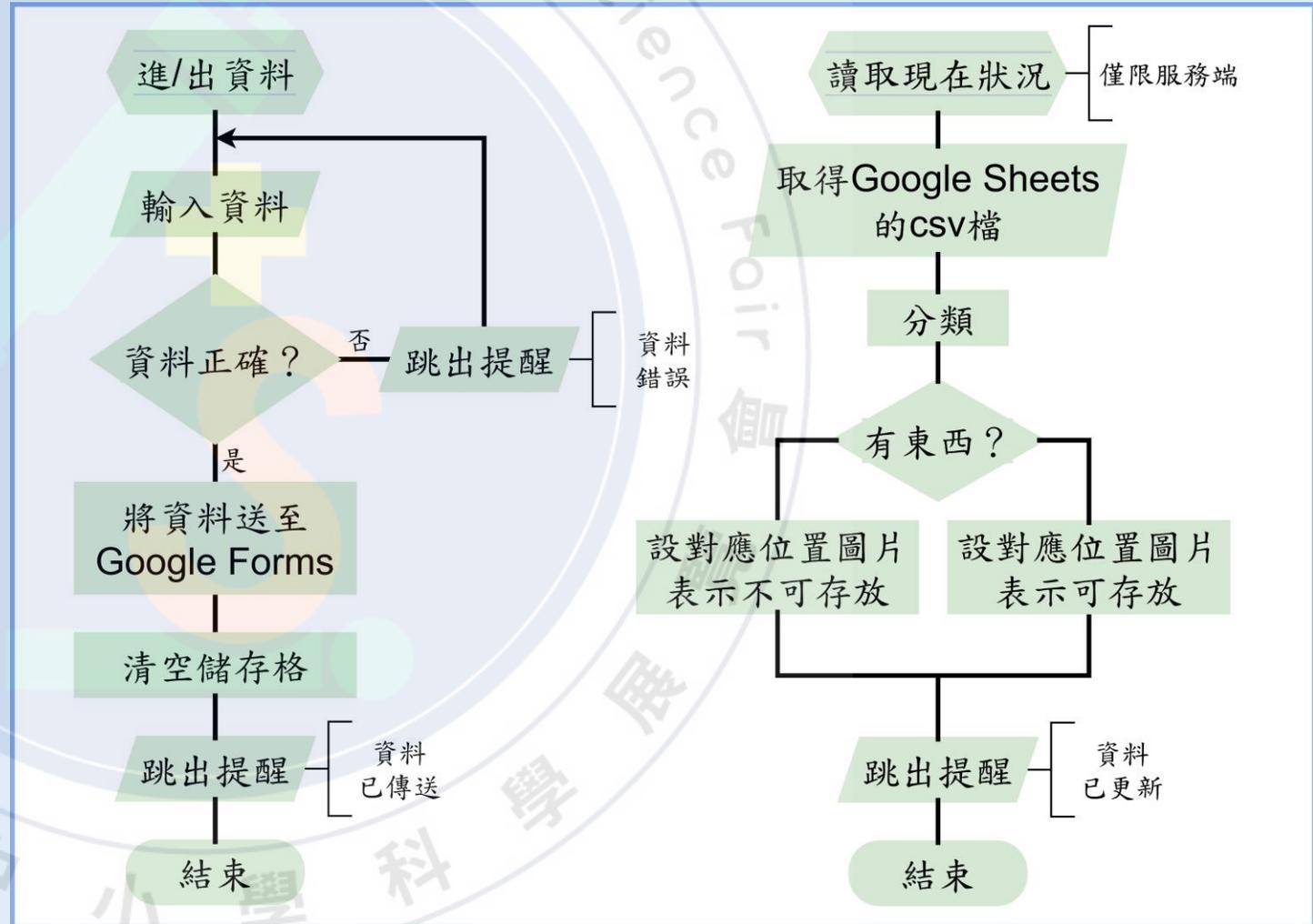
研究過程

(一) AI2

客戶端僅具備取貨的功能，其餘功能只存在於服務端。

輸入資料後，判斷格式，正確則送出，並清空輸入框，給使用者填入下一則資料。

讀取現在狀況會在點選更新現況或傳送後更新。我們利用 Google Sheets 提供的 csv 檔，去判讀更新資料。



附圖3. 運行架構_ App Inventor 2

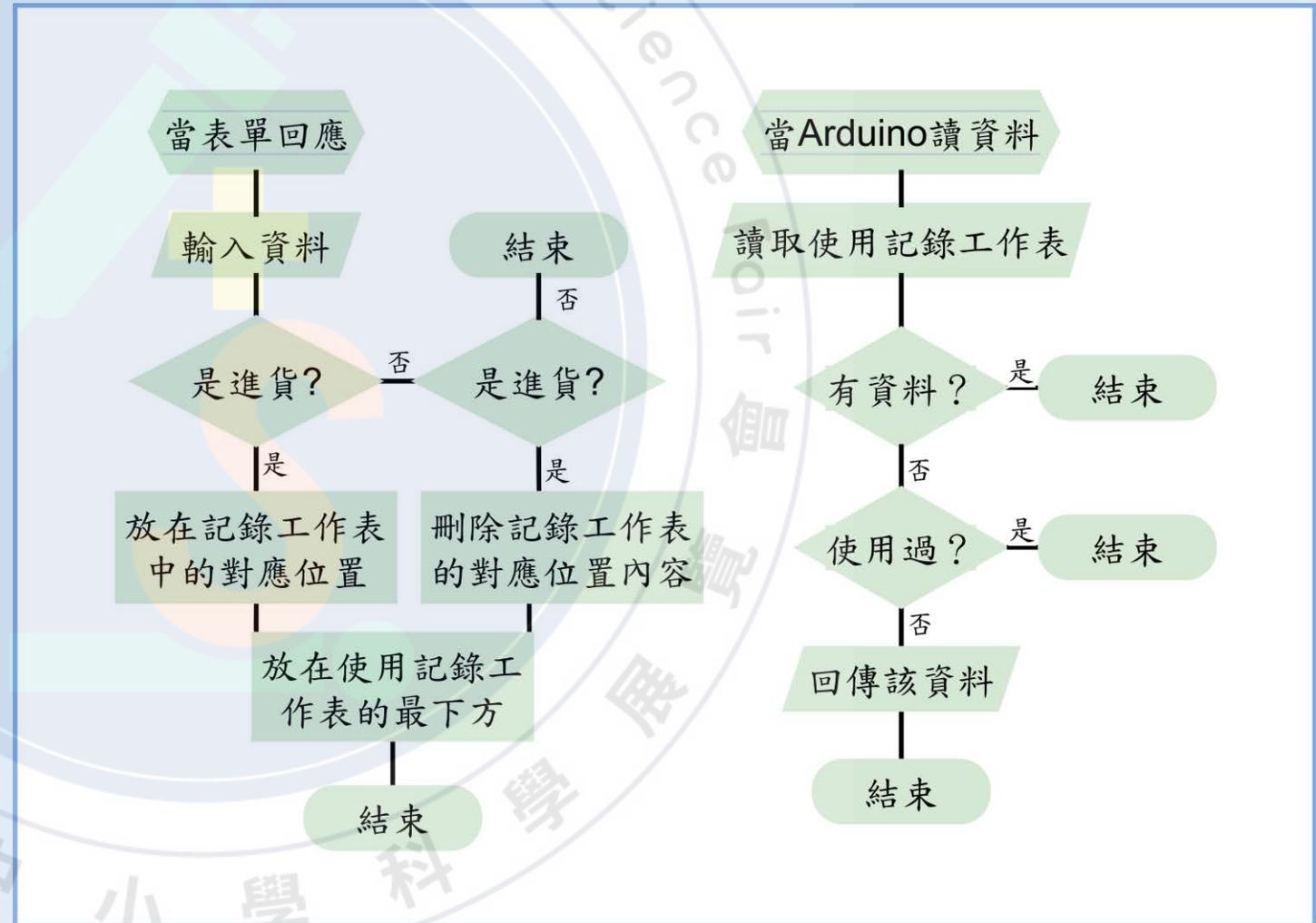
研究過程

(二) GAS

在 Google Sheets 中，總共有 4 個，進貨資料輸入、出貨資料輸入、現況記錄以及使用記錄。前 2 為 AI2 所傳遞的資料。

現況紀錄，由 GAS 整理資料，並發布至網路，給予 AI2 讀取。

使用紀錄，為了提供 Arduino 讀取，由 GAS 整理資料，依據時間先後順序使用。



附圖4. 運行架構_ App Inventor 2

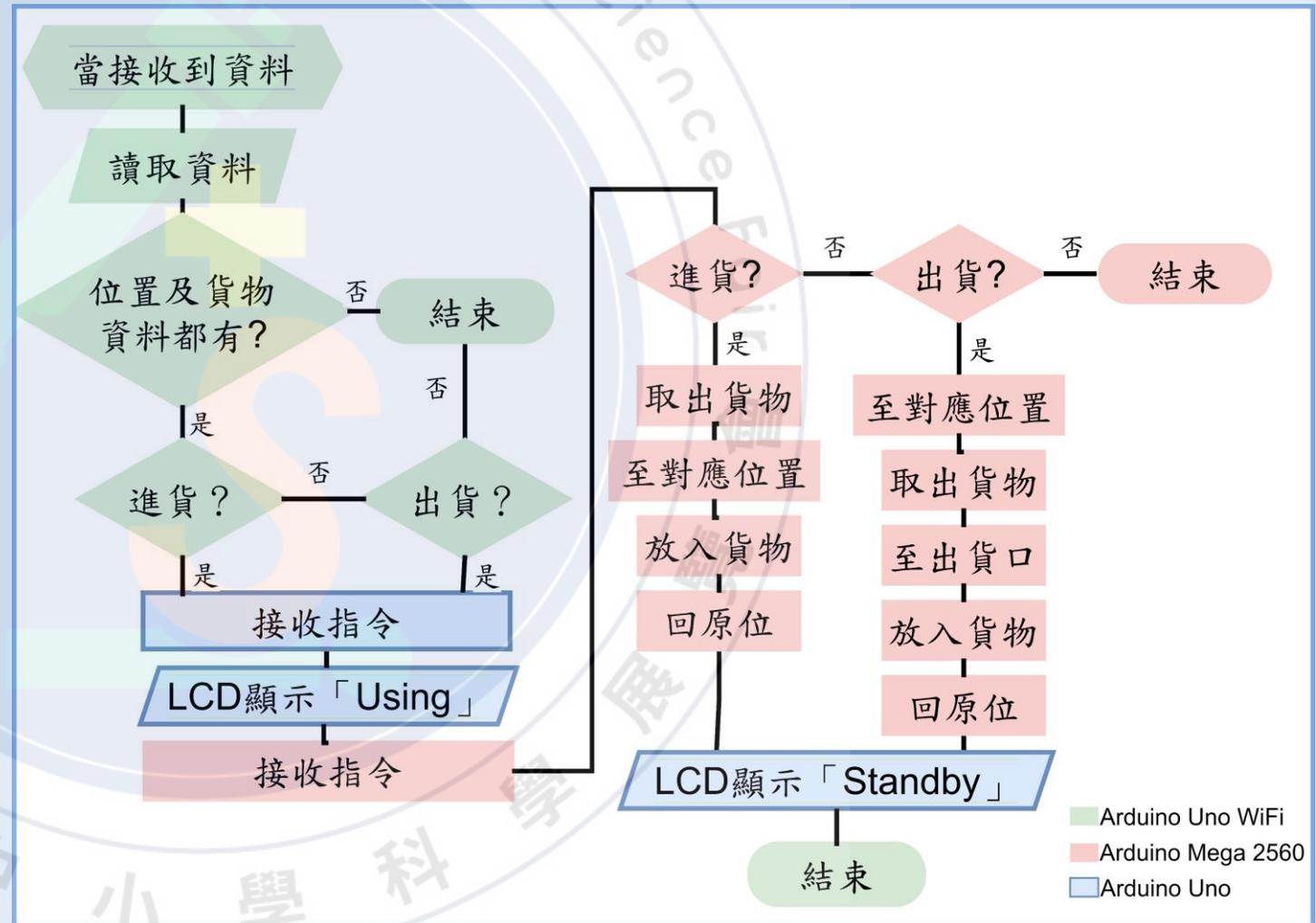
研究過程

(三) Arduino

綠色是 Arduino Uno WiFi，負責讀取資料、記錄現況、判斷資料，以及分工給其他Arduino。

藍色是 Arduino Uno，操控LCD顯示模組。

紅色是Arduino Mega 2560，負責操控整體的機械運作。



附圖5. 運行架構_ App Inventor 2

研究結果

一、App Inventor 2

(一) 客戶端

1. 配色

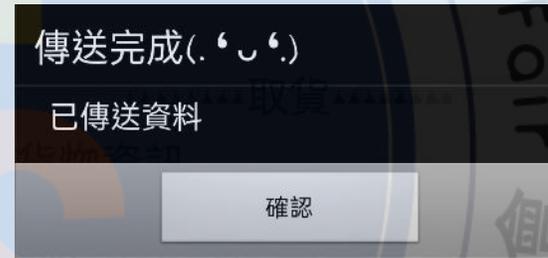
採用柔和的色調，讓使用者看了能較為舒適，提示字採用醒目的紅色，使其於整體畫面中較為突出，讓使用者能夠注意到文字內容的提醒。

2. 輸入內容的格式提醒

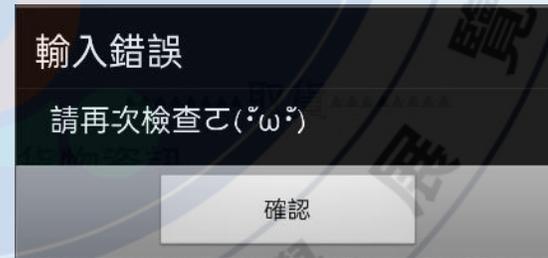
在需輸入內容的格子內加入格式提醒，避免使用者不清楚需輸入的內容。

3. 傳輸提示

在輸入內容並按確認送出時，會先進行辨別文字格式，不符合者會跳出提醒，符合者會告知傳送成功。



附圖6. 成功傳輸提示



附圖7. 資料錯誤提示



附圖8. 客戶端的畫面配置

研究結果

一、App Inventor 2

(二) 服務端

1. 以客戶端為基礎

配色、輸入內容的格式提醒、傳輸提示，與之無異，額外的功能也有不同的配色。

2. 現況配置圖

以綠色底的圈圈表示該儲存格是可放入貨物的，用紅色底的叉叉表示該儲存格已有存放的貨物，不可再放入貨物，點選下方的「刷新現況」，可以取得最新現況，在刷新成功時，會跳出提示，告知使用者已更新。

The screenshot displays the service end interface. At the top, it says "▶▶服務端◀◀". Below that is a yellow box with a red border containing a warning: "小提示 !!請檢查後再送出!!". The main area is a grid of storage slots. The columns are labeled "x1", "x2", and "x3". The rows are labeled "y1", "y2", "y3", and "y4". In the y1 row, the x1 slot is red with a white 'X', while the x2 and x3 slots are green with white circles. In the y2, y3, and y4 rows, all slots are green with white circles. Below the grid is a red button labeled "刷新現況". At the bottom, there is a purple box with the text "進貨" and a "確認" button.

位置	x1	x2	x3
y1	×	○	○
y2	○	○	○
y3	○	○	○
y4	○	○	○

進貨

貨物資訊
輸入貨物代碼8碼

存放位置
x軸 輸入個位數 y軸 輸入個位數

客戶名稱
請輸入客戶名稱(全英小寫)

電話末三碼
請輸入3位數字

確認

附圖9.
服務端的畫面配置

The screenshot shows a black dialog box with white text. It says "已更新" and "內容現狀已更新". At the bottom, there is a white button labeled "確定".

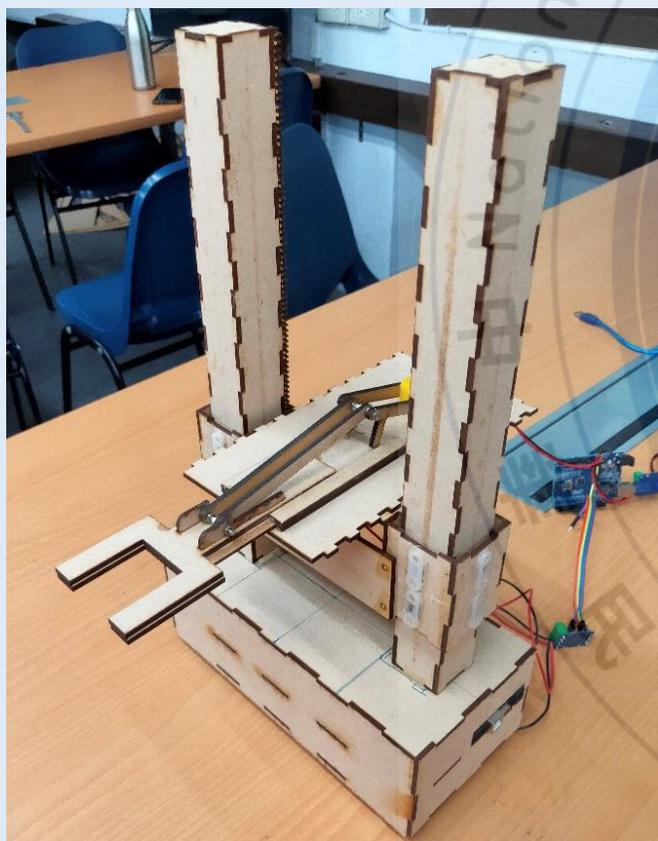
已更新
內容現狀已更新

確定

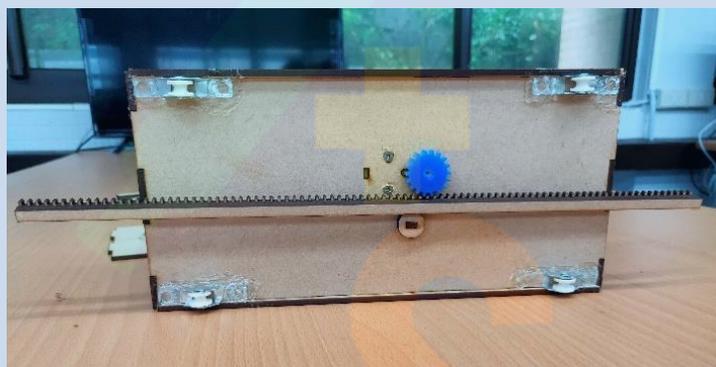
附圖10. 資料更新提示

研究結果

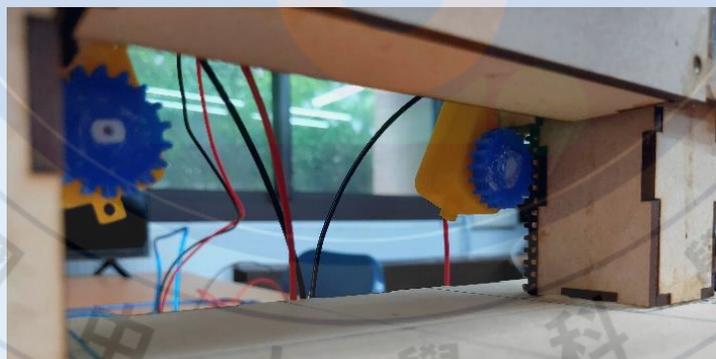
二、自製機械取貨裝置



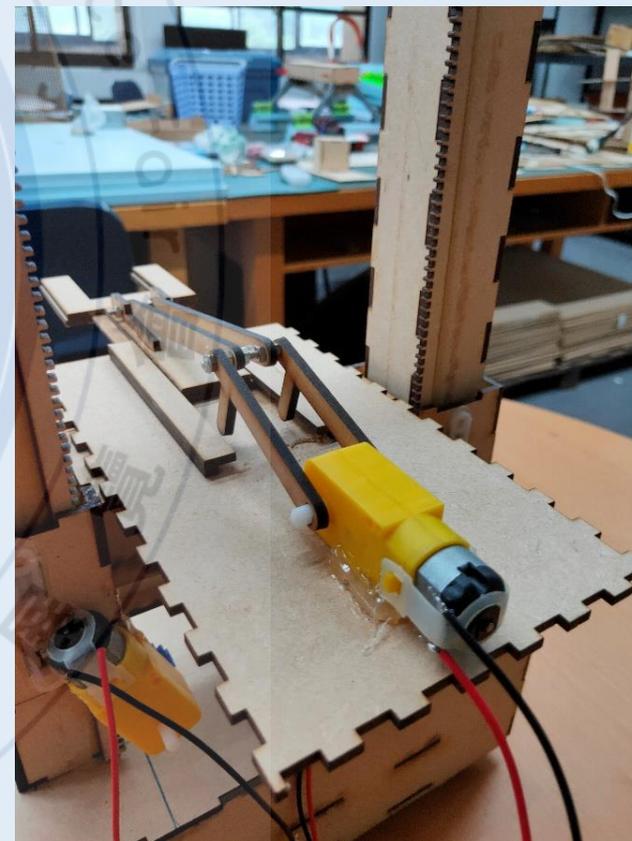
附圖11. 自製三維手臂側視圖



附圖12. X軸底部傳動機構



附圖13. Y軸上升下降機構



附圖14. Z軸伸縮取物機構

討論

一、有無其他方法增加實用性？

(一) 對於客戶

1. 製作LINE的機器人
2. 加入QR code的功能

(二) 對於補貨者

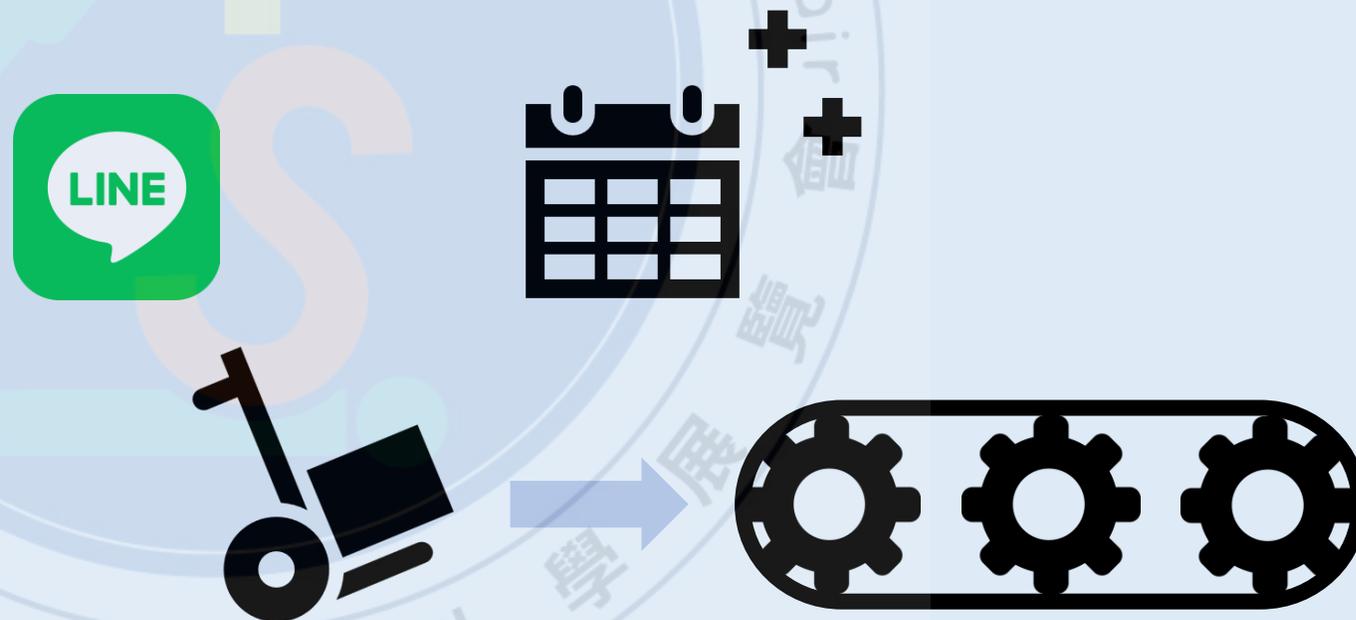
1. 對貨物設定日期
2. 加入QR code的功能
3. 加入自動進貨的功能

(三) 並用IoT取貨和現場取貨

二、與現在機制的比較

(一) 找尋貨物的速度

(二) 取得正確貨物的機率



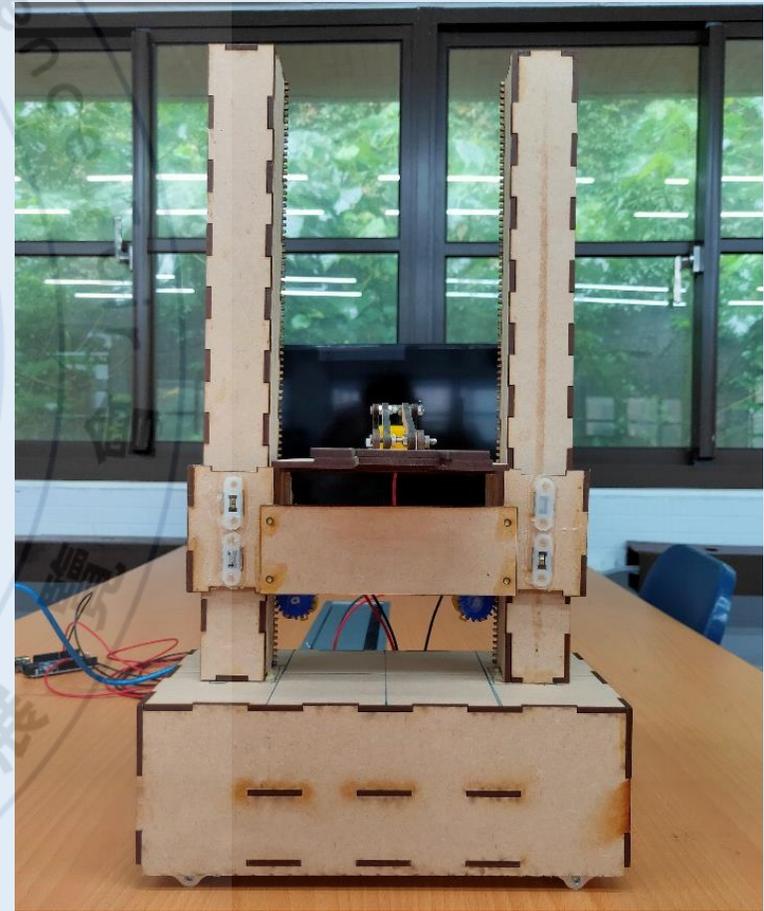
附圖15. 增修內容

結論

此項作品為我們的生活增添不少便利性，不僅縮短了取貨時間，還降低在取貨過程中人為疏失的可能性。

此裝置可依據存放的物品進行設定，不僅可以用於超商取貨，還可另外改造成適合書籍存放取用的方式，使圖書館的借閱能夠更加便利，可以讓民眾於機器投入書籍刷借閱卡進行還書的動作，也可在現場刷借閱卡借閱機器內的書籍。關鍵原因在於自製三維手臂，由於採用的是Arduino所以整體的價格會較市售的便宜不少。而改裝只需要配合物品改變貨櫃及Z軸的手臂樣式即可，就整體而言，這項裝置的可塑性較高。

雖然目前的裝置尚未有如同討論中更進一步的設計，但期許我們之後將這些增加便利性的設計加入作品後，能夠成為一個新穎的裝置來幫助大眾，為我們的生活取得較大的利益。



附圖16. 自製三維手臂正視圖

參考資料

- 一、Jiun-Yan Shiau Ph.D。Warehouse Management System (WMS)。取自：
<http://www2.nkust.edu.tw/~sho/wms.html>
- 二、MIC資策會產業情報研究所 (2020)。【網購大調查系列一】行動下單急追PC呈五五波 行動商務正式成為主流。取自：<https://mic.iii.org.tw/news.aspx?id=555>
- 三、楊惟雯 (2012)。物流寄情13：物聯網的發展趨勢。取自：
<https://nmart.pixnet.net/blog/post/47268694%E7%89%A9%E8%81%AF%E7%B6%B2%E7%9A%84%E7%99%BC%E5%B1%95%E8%B6%A8%E5%8B%A2>
- 四、Eugene Michaels(2020)。物聯網：定義、應用和風險。取自：<https://nordvpn.com/zh-tw/blog/wulianwang/>
- 五、黃柏堯 (2020)。連上網就叫IoT？解析物聯網的定義跟發展趨勢。數位時代，智慧城市。取自：
<https://www.bnext.com.tw/article/60182/iot-trend-omo>
- 六、黃柏堯 (2020)。引爆物聯網商機的最後一哩路。數位時代，物聯網。取自：
<https://www.bnext.com.tw/article/56263/iot-business-opportunity>
- 七、半工人YZ (2018)。NodeMCU 教學 - 11：使用 NodeMCU 以及 Google script 寫入&讀取 Google sheets 資料。取自：<http://wyj-learning.blogspot.com/2018/02/nodemcu-google-script-google-sheets.html>
- 八、黑修斯 (2018)。Arduino | I2C練習(1)：使用兩個Arduino做溝通。取自：
<https://hugheschung.blogspot.com/2018/05/arduino-i2c1arduino.html>
- 九、Super169 (2015)。Arduino 之間的 I2C 通訊 (4) 由 master 向 slave 要求資料回傳。取自：
<http://arduino169.blogspot.com/2015/08/arduino-i2c-4-master-slave.html>