

# 朝陽科技大學

## 資訊工程系

### 【專題成果報告】

#### 無人取貨商店

指導教授：洪士程 博士  
專題組員：陳彥勳 (10727134)  
林彥謹 (10727428)  
廖子翔 (10727436)

中華民國 110 年 12 月

# 無人取貨商店

指導教授：洪士程 老師

陳彥勳、林彥瑾、廖子翔

朝陽科技大學資訊工程系

413 台中市霧峰區吉峰東路 168

Tel：(04)23323000 ext. 4682

E-mail：

## 摘要

我們本次的專題，主要是以 HTML+PHP 所做出的網頁去跟 KINOVA 機械手臂做連接，以實現快速找貨物且不需要經過人力來接觸去取得貨物。先假設所有貨物都有擺放在相對位置並且都將相關資料都儲存在裡面，藉由我們設計的取貨認證頁面，輸入自己的姓名以及末三碼，當資料確認無誤時，機械手臂會幫你夾出相對應的貨物。

### **Abstract**

Our topic ,this time is mainly based on the web page made by HTML+PHP to connect with the KINOVA robotic arm. Our goal is to find the merchandise quickly and accurately without human contact to get .For exmple, Let's assume that all goods are placed. In the relative position and all relevant information is stored in the robotic arm, through our designed pickup certification page,and enter your name and the last three codes of your phone, when the information is confirmed, the robotic arm will help you pick up the corresponding merchandise.

## 目 錄

目 錄.....	4
圖目錄.....	5
1. 專題簡介.....	6
2. 系統架構.....	7
2.1 技術採用與說明 .....	7
2.1.1 XAMPP .....	7
2.1.2 Visual Studio Code .....	7
2.1.3 HTML .....	8
2.1.4 PHP .....	9
2.1.5 C++ .....	9
2.2 硬體設備.....	10
2.2.1 kinova development center .....	10
2.2.2 機械手臂 kinova gen2 .....	11
2.2.3 範例程式 .....	12
2.3 流程圖 .....	14
3. 實際操作 .....	16
3.1 操作流程 .....	16
3.2 網頁範例程式 .....	20
4. 結果與討論 .....	21
5. 參考資料 .....	22

## 圖目錄

圖 2.1.1- 1 XAMPP 的使用介面 .....	7
圖 2.1.2- 1 VS CODE 使用介面 .....	8
圖 2.1.3- 1 HTML 範例 .....	8
圖 2.1.4- 1 PHP.....	9
圖 2.1.5- 1 C++.....	9
圖 2.2.1- 1 機械手臂定位.....	10
圖 2.2.1- 2development center.....	10
圖 2.2.3- 1 kinova gen2.....	11
圖 2.2.3- 2 位置一 .....	12
圖 2.2.3- 3 位置二 .....	12
圖 2.2.3- 4 位置三 .....	13
圖 2.2.3- 5 所有定位.....	13
圖 2.3- 1 網頁流程圖 .....	14
圖 3.1- 1 首頁 .....	16
圖 3.1- 2 首頁 .....	16
圖 3.1- 3 取貨介面.....	17
圖 3.1- 4 驗證身份.....	17
圖 3.1- 5 驗證成功.....	18
圖 3.1- 6 手臂抓取相對應貨物 .....	18
圖 3.1- 7 手臂抓去貨物到指定地點.....	19
圖 3.1- 8 完成.....	19

## 1. 專題簡介

由於近幾個月新冠肺炎(Covid-19)肆虐全世界，導致大家在網購上的使用率都有大大的提升，但還是需要到超商去取貨或有貨運公司到家送貨，但無論是哪一種方式都會有人與人的接觸，為了將互相接觸的機率降低，所以有了這個無人取貨商店的想法，所以我們本次專題利用手臂去取代人力，除了可以減少找貨的時間，還可以避免人與人的接觸，除此之外，需要到超商取貨的人也可以不必跟去購買商品的人排隊等待，也因為使用了機械手臂去取代人力，所以可以更精準更不費力地做完所有動作。

## 2. 系統架構

### 2.1 技術採用與說明

#### 2.1.1 XAMPP

XAMPP 是一個把 Apache 與 PHP 及 MariaDB 集合在一起的安裝包，允許使用者可以在自己的電腦上輕易的建立網頁伺服器。



圖 2.1.1- 1 XAMPP 的使用介面

#### 2.1.2 Visual Studio Code

VS Code 是由微軟開發且跨平台的免費原始碼編輯器。Visual Studio Code 預設支援非常多的程式語言，包括 JavaScript、TypeScript、CSS 和 HTML；也可以通過下載擴充支援 Python、C/C++、Java 和 Go 在內的其他語言。讓我們不管是在製作網頁或機械手臂上的定位都能在 VS CODE 上輕鬆完成。

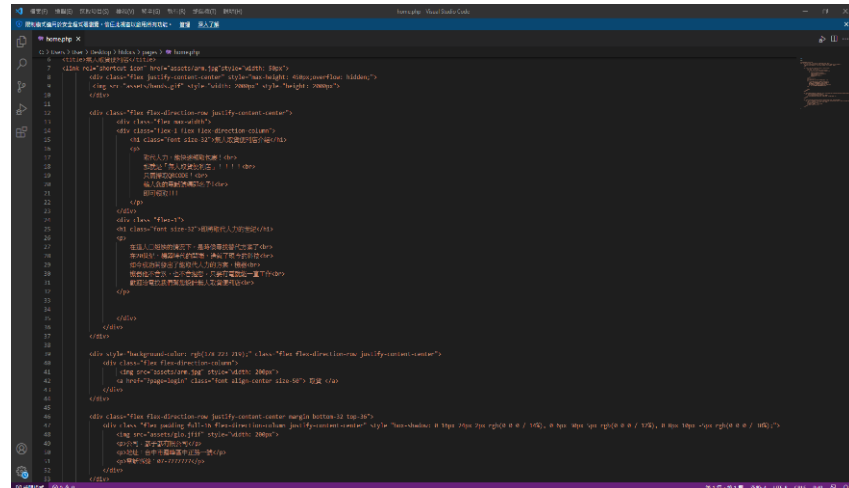


圖 2.1.2- 1 VS CODE 使用介面

### 2.1.3 HTML

超文本標記語言(HyperText Markup Language，簡稱：HTML)是一種建立網頁的標準標記語言，常常與CSS、JavaScript一起被眾多的網站用於設計網頁前端的架構或網頁應用程式的使用者介面，與PHP不同的地方是，HTML是一種靜態的語言，可以製作標題、段落等等，也可以說是整個網站的骨架。

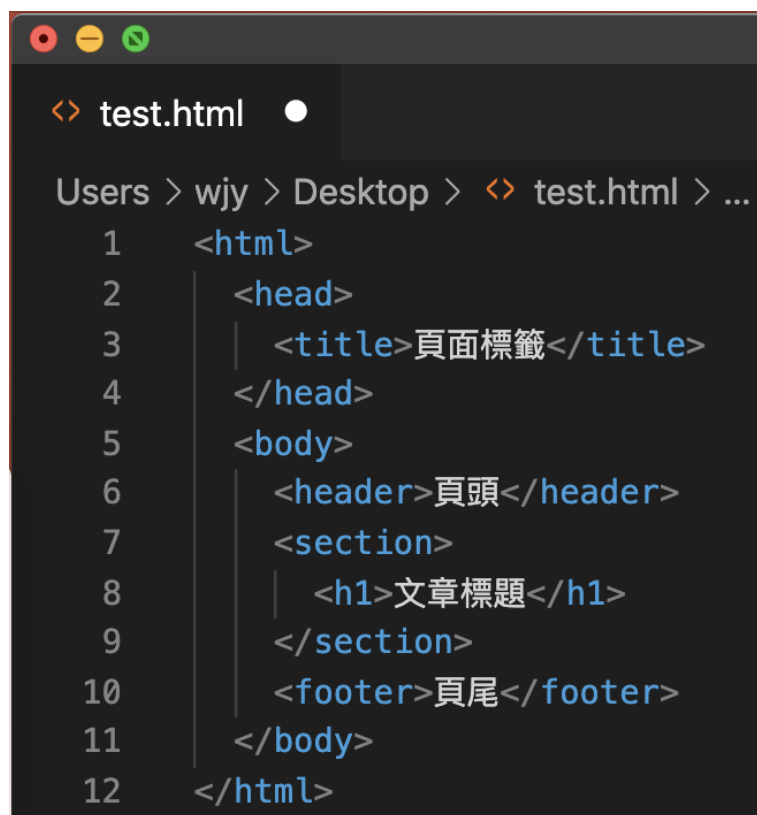


圖 2.1.3- 1 HTML 範例



#### 2.1.4 PHP

超文字預處理器，是一種開源的通用電腦手稿語言，尤其適用於網路開發並可嵌入 HTML 中使用。PHP 的語法借鑑吸收 C 語言、Java 和 Perl 等流行電腦語言的特點，易於一般程式設計師學習。



圖 2.1.4- 1 PHP

#### 2.1.5 C++

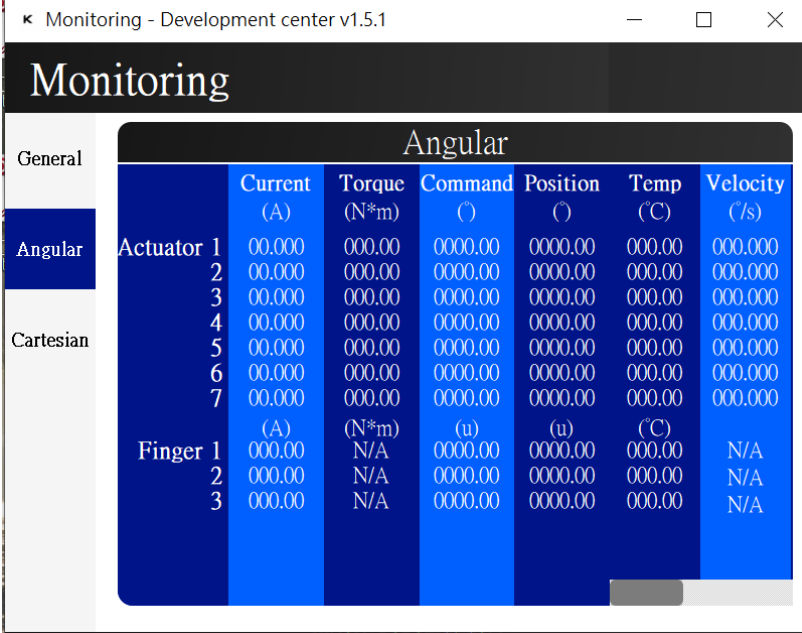
C++是一種被廣泛使用的電腦程式設計語言。它是一種通用程式設計語言，支援多重程式設計範式，例如過程化程式設計、資料抽象化、物件導向程式設計、泛型程式設計和設計模式等。



圖 2.1.5- 1 C++

## 2.2 硬體設備

### 2.2.1 kinova development center



Angular						
	Current (A)	Torque (N*m)	Command (°)	Position (°)	Temp (°C)	Velocity (°/s)
Actuator	1	000.00	000.00	0000.00	000.00	000.000
	2	000.00	000.00	0000.00	000.00	000.000
	3	000.00	000.00	0000.00	000.00	000.000
	4	000.00	000.00	0000.00	000.00	000.000
	5	000.00	000.00	0000.00	000.00	000.000
	6	000.00	000.00	0000.00	000.00	000.000
	7	000.00	000.00	0000.00	000.00	000.000
Finger	(A)	(N*m)	(u)	(u)	(°C)	
	1	N/A	0000.00	0000.00	000.00	N/A
	2	N/A	0000.00	0000.00	000.00	N/A
	3	N/A	0000.00	0000.00	000.00	N/A

圖 2.2.1- 1 機械手臂定位

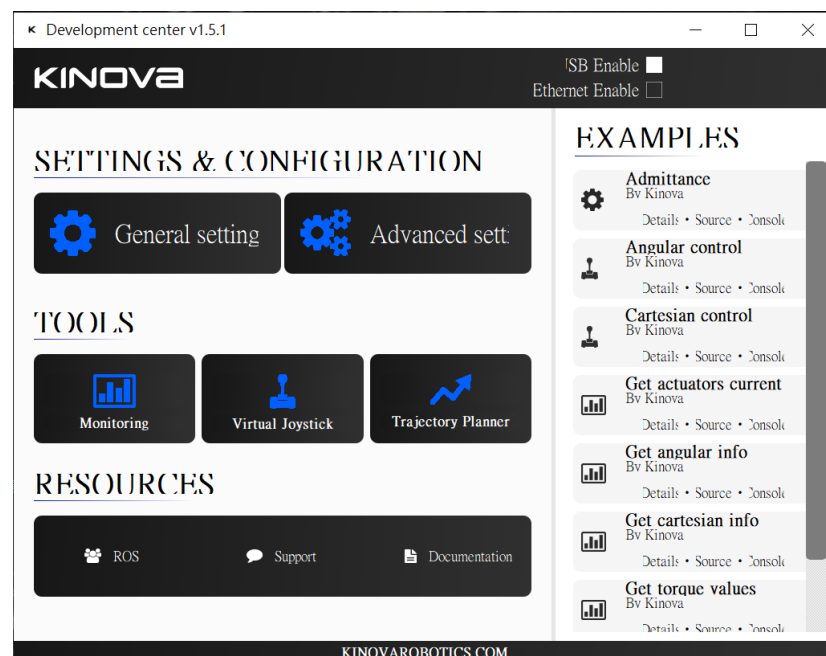


圖 2.2.1- 2development center

### 2.2.2 機械手臂 kinova gen2



圖 2.2.3- 1 kinova gen2

KINOVA®Gen 協作型機械手臂是由六個組成的輕型機器關節相互運作來進行活動。此機械手臂有幾項特色，高效能、靈活、可擴充，且他的手臂程式相容 ROS 和 SDK，每個執行器中有五個傳感器：位置、溫度、力矩、電流、加速度計度計。通過控制器或電腦，使用者可以在三維空間中移動機器手臂，並用夾爪抓取或放開物品。

## 2.2.3 範例程式

### 機械手臂定位

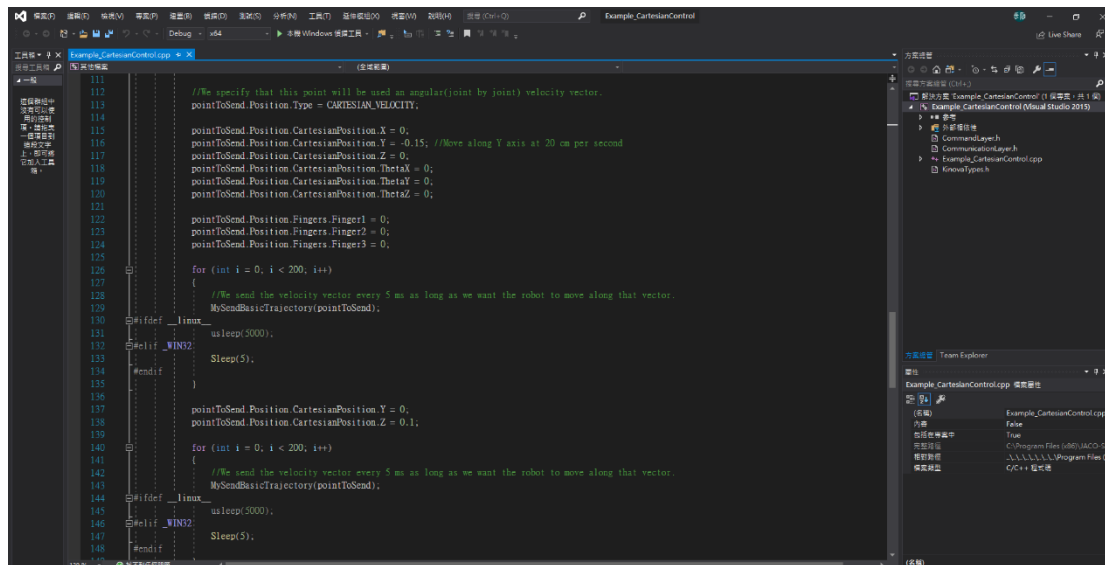


圖 2.2.3- 2 位置一

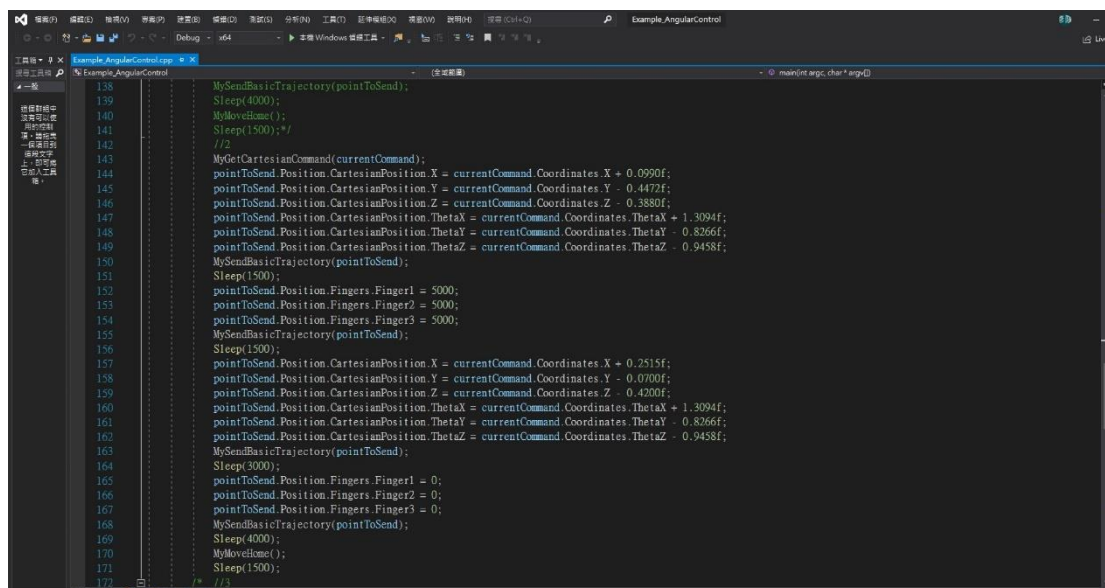


圖 2.2.3- 3 位置二

```

110 //We get the actual angular command of the robot.
111 //1
112 MyGetCartesianCommand(currentCommand);
113 pointToSend.Position.CartesianPosition.X = currentCommand.Coordinates.X - 0.0515f;
114 pointToSend.Position.CartesianPosition.Y = currentCommand.Coordinates.Y - 0.3647f;
115 pointToSend.Position.CartesianPosition.Z = currentCommand.Coordinates.Z - 0.4380f;
116 pointToSend.Position.CartesianPosition.ThetaX = currentCommand.Coordinates.ThetaX + 1.3094f;
117 pointToSend.Position.CartesianPosition.ThetaY = currentCommand.Coordinates.ThetaY - 0.8266f;
118 pointToSend.Position.CartesianPosition.ThetaZ = currentCommand.Coordinates.ThetaZ - 0.9458f;
119 MySendBasicTrajectory(pointToSend);
120 Sleep(1500);
121 pointToSend.Position.Fingers.Finger1 = 5000;
122 pointToSend.Position.Fingers.Finger2 = 5000;
123 pointToSend.Position.Fingers.Finger3 = 5000;
124 MySendBasicTrajectory(pointToSend);
125 Sleep(1500);
126 pointToSend.Position.CartesianPosition.X = currentCommand.Coordinates.X + 0.2515f;
127 pointToSend.Position.CartesianPosition.Y = currentCommand.Coordinates.Y - 0.0700f;
128 pointToSend.Position.CartesianPosition.Z = currentCommand.Coordinates.Z - 0.4200f;
129 pointToSend.Position.CartesianPosition.ThetaX = currentCommand.Coordinates.ThetaX + 1.3094f;
130 pointToSend.Position.CartesianPosition.ThetaY = currentCommand.Coordinates.ThetaY - 0.8266f;
131 pointToSend.Position.CartesianPosition.ThetaZ = currentCommand.Coordinates.ThetaZ - 0.9458f;
132 MySendBasicTrajectory(pointToSend);
133 Sleep(1500);
134 pointToSend.Position.Fingers.Finger1 = 0;
135 pointToSend.Position.Fingers.Finger2 = 0;
136 pointToSend.Position.Fingers.Finger3 = 0;
137 MySendBasicTrajectory(pointToSend);
138 Sleep(4000);
139 MyMoveHome();
140 Sleep(1500);
141 //2
142 MyGetCartesianCommand(currentCommand);
143 pointToSend.Position.CartesianPosition.X = currentCommand.Coordinates.X + 0.0990f;
144

```

圖 2.2.3- 4 位置三

```

170 MyMoveHome();
171 Sleep(1500);
172 //3
173 MyGetCartesianCommand(currentCommand);
174 pointToSend.Position.CartesianPosition.X = currentCommand.Coordinates.X - 0.0139f;
175 pointToSend.Position.CartesianPosition.Y = currentCommand.Coordinates.Y - 0.1550f;
176 pointToSend.Position.CartesianPosition.Z = currentCommand.Coordinates.Z - 0.4244f;
177 pointToSend.Position.CartesianPosition.ThetaX = currentCommand.Coordinates.ThetaX + 1.3094f;
178 pointToSend.Position.CartesianPosition.ThetaY = currentCommand.Coordinates.ThetaY - 0.8266f;
179 pointToSend.Position.CartesianPosition.ThetaZ = currentCommand.Coordinates.ThetaZ - 0.9458f;
180 MySendBasicTrajectory(pointToSend);
181 Sleep(1500);
182 pointToSend.Position.Fingers.Finger1 = 5000;
183 pointToSend.Position.Fingers.Finger2 = 5000;
184 pointToSend.Position.Fingers.Finger3 = 5000;
185 MySendBasicTrajectory(pointToSend);
186 Sleep(1500);
187 pointToSend.Position.CartesianPosition.X = currentCommand.Coordinates.X + 0.2515f;
188 pointToSend.Position.CartesianPosition.Y = currentCommand.Coordinates.Y - 0.0700f;
189 pointToSend.Position.CartesianPosition.Z = currentCommand.Coordinates.Z - 0.4200f;
190 pointToSend.Position.CartesianPosition.ThetaX = currentCommand.Coordinates.ThetaX + 1.3094f;
191 pointToSend.Position.CartesianPosition.ThetaY = currentCommand.Coordinates.ThetaY - 0.8266f;
192 pointToSend.Position.CartesianPosition.ThetaZ = currentCommand.Coordinates.ThetaZ - 0.9458f;
193 MySendBasicTrajectory(pointToSend);
194 Sleep(1500);
195 pointToSend.Position.Fingers.Finger1 = 0;
196 pointToSend.Position.Fingers.Finger2 = 0;
197 pointToSend.Position.Fingers.Finger3 = 0;
198 MySendBasicTrajectory(pointToSend);
199 Sleep(4000);
200 MyMoveHome();
201 Sleep(1500);
202
203

```

圖 2.2.3- 5 所有定位

## 2.3 流程圖

### 網頁登入流程圖

順利的話 1 打開我們做的首頁頁面 2 點選取貨 3 登入驗證身分 4 確認身分手臂開始動作。

錯誤的話 4 發現登入錯誤 5 再次輸入正確名稱及後三碼 6 確認身分手臂開始動作。

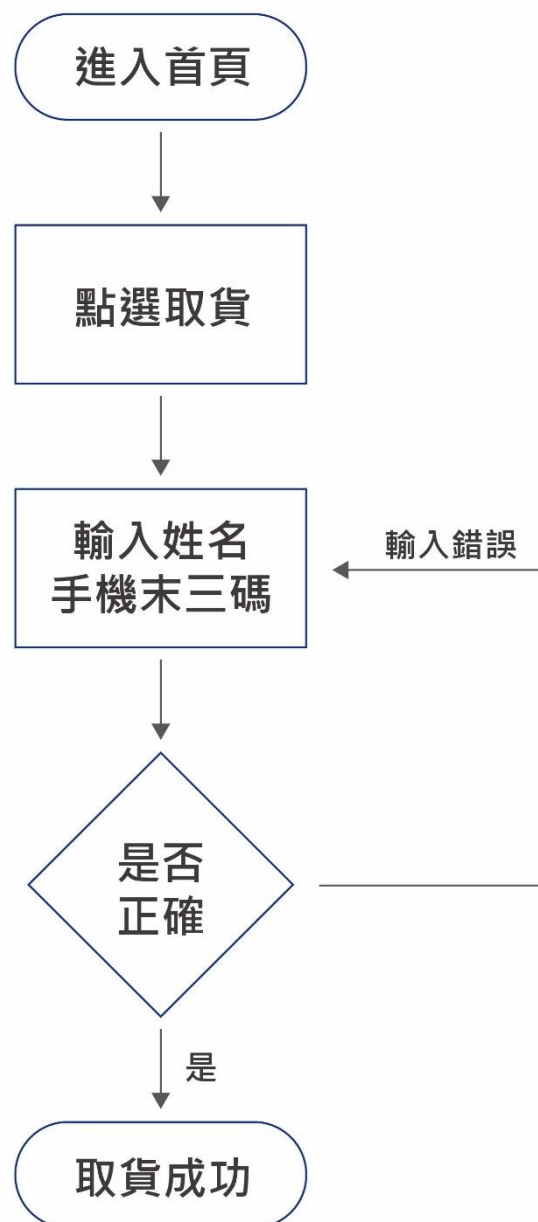


圖 2.3- 1 網頁流程圖

## 機械手臂流程圖

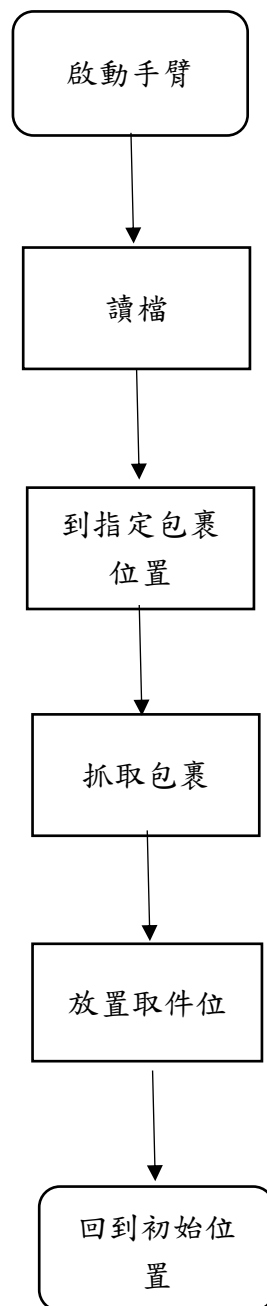


圖 2.3-2 手臂流程圖

### 3. 實際操作

#### 3.1 操作流程

##### 步驟一：點選首頁取貨

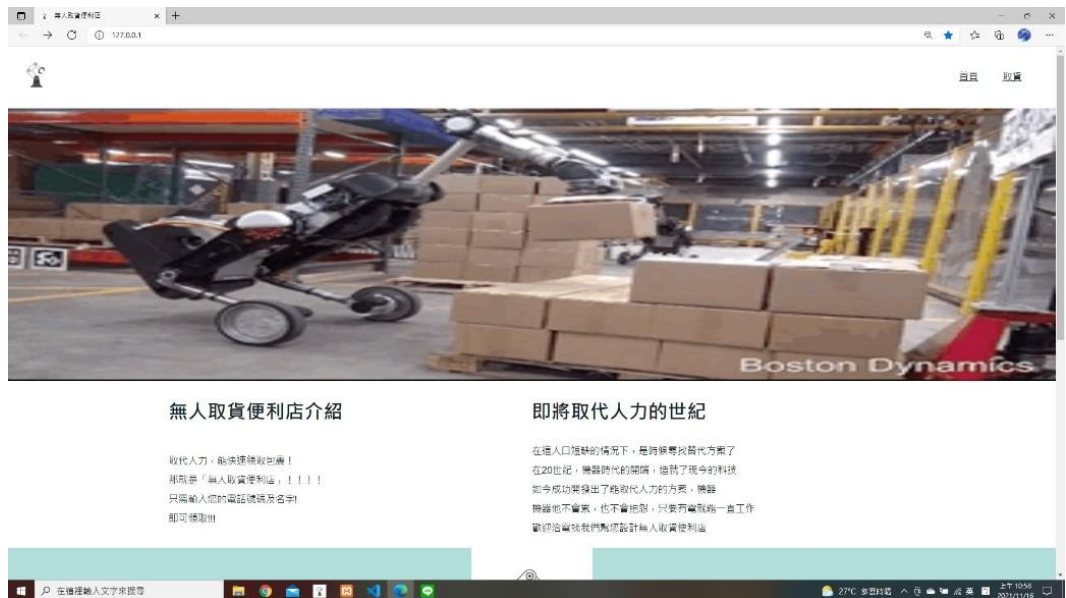


圖 3.1- 1 首頁



圖 3.1- 2 首頁



## 步驟二：進到取貨頁面驗證身分

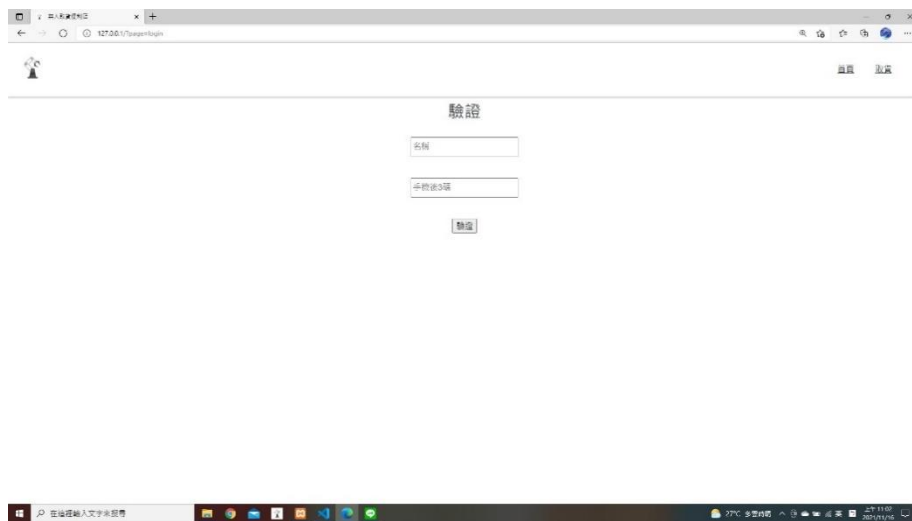


圖 3.1- 3 取貨介面

## 步驟三：身分驗證錯誤即可重新輸入

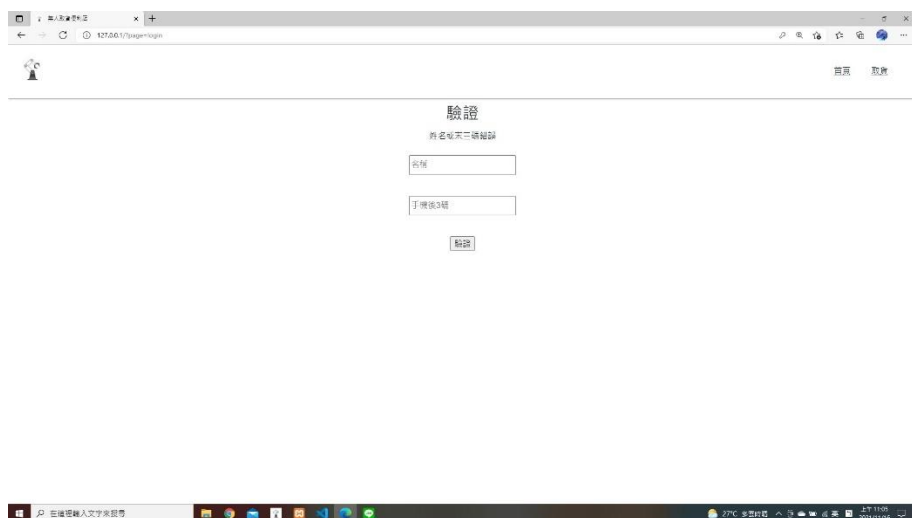


圖 3.1- 4 驗證身份

步驟四：成功登入後，手臂開始動作

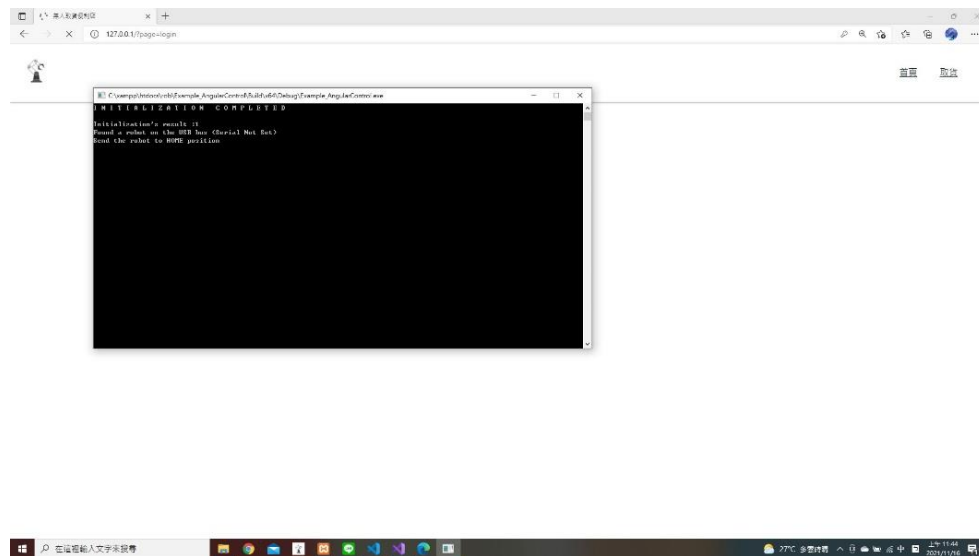


圖 3.1- 5 驗證成功

步驟五：手臂開始動作依照登入身分抓取包裹



圖 3.1- 6 手臂抓取相對應貨物

步驟六：手臂抓取指定包裹後放置取件人位置



圖 3.1- 7 手臂抓去貨物到指定地點

步驟七：取件完後手臂歸定位



圖 3.1- 8 完成

### 3.2 網頁範例程式

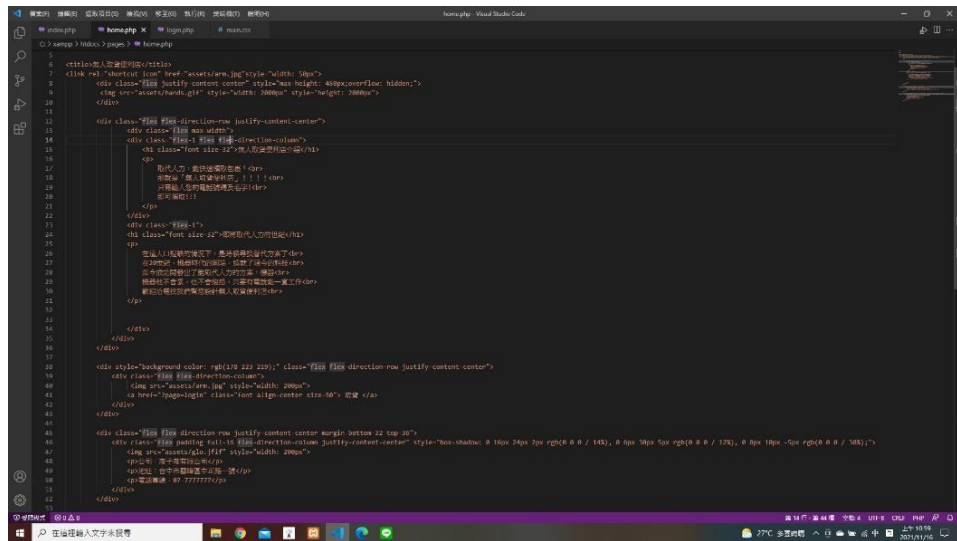


圖 3.2-1 首頁程式碼

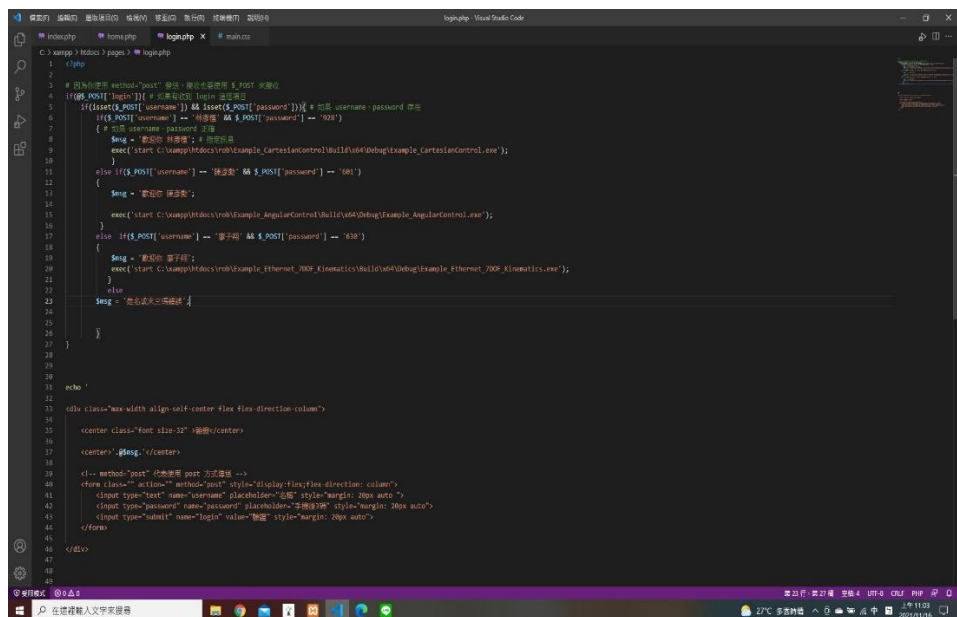


圖 3.2- 2 身分驗證程式碼

因為我們有先預設 3 組名稱和手機後三碼，所以圖 3.1-4 的部分是假設使用者輸入錯誤，可以再次輸入正確的名稱及後三碼來進行取貨的動作。

#### 4. 結果與討論

由於大家網路購物的需求提高，尤其在疫情期間，人人都不敢出門，幾乎所有想要的商品都透過網路訂購，這讓超商店員不但要處理超商的結帳等等的問題，還要應付取貨的手續，因此我們這組研發了無人取貨商店，可以直接在網路進行認證及拿貨的動作，可以減少人力成本，也可以減少人群接觸，任何時段都能取貨的優點。

我們這組沒有用到攝像頭的部分，導致夾取包裹部分只能把點寫死，變的制式，往後若有繼續探討部分，加入攝像頭可以做辨識，那麼此專案會更加完善。

## 5. 參考資料

<https://www.kinovarobotics.com/en/assistive-technologies>

<https://www.kinovarobotics.com/sites/default/files/UG->

[008\\_KINOVA\\_Software\\_development\\_kit-User\\_guide\\_EN\\_R02%20%281%29.pdf](008_KINOVA_Software_development_kit-User_guide_EN_R02%20%281%29.pdf)

<https://www.youtube.com/watch?v=CLUPkcLQm64&t=1s>

<https://www.youtube.com/watch?v=Ag5lH7XuLxc>

<https://www.youtube.com/watch?v=yu-0Vk7xz8s&t=2304s>

<https://ithelp.ithome.com.tw/questions/10204480>

<https://www.4gtv.tv/article/2021092912000016>

<https://zh.wikipedia.org/wiki/PHP>

<https://developer.mozilla.org/zh->

[TW/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](TW/docs/Learn/Getting_started_with_the_web/HTML_basics)

<https://zh.wikipedia.org/wiki/HTML>

[https://zh.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://zh.wikipedia.org/wiki/Visual_Studio_Code)

<https://zh.wikipedia.org/wiki/XAMPP>