

臺中市立大甲工業高級中等學校

專 題 製 作

可應用於小型工廠之自動化倉儲分揀系統  
An Automated Warehousing and Sorting System  
for Application in Small-Scale Manufacturing  
Facilities

科 別：資訊科

組 員：何欣諭、施建宇、陳俊錡、陳育威

指導老師：陳俊男、林世原

中 華 民 國 1 1 4 年 5 月 6 日

## 目 錄

壹、 前言.....	- 3 -
一、 研究背景.....	- 3 -
二、 研究動機.....	- 3 -
貳、 研究目的.....	- 4 -
一、 提升倉儲操作效率.....	- 4 -
二、 探索智慧倉儲系統的實現方式.....	- 4 -
三、 提高貨物定位精確度.....	- 4 -
四、 簡化倉儲管理流程.....	- 4 -
五、 為未來研究奠定基礎.....	- 4 -
參、 文獻探討.....	- 5 -
一、 myRIO 相關知識.....	- 5 -
二、 市面上倉儲系統.....	- 8 -
三、 TCS3200 顏色感測相關知識.....	- 9 -
肆、 研究方法.....	- 10 -
伍、 研究分析及結果.....	- 17 -
陸、 研究結論與建議.....	- 21 -
柒、 資料來源.....	- 22 -

## 壹、前言

### 一、研究背景

在全球COVID-19疫情期間，世界各國普遍面臨人力短缺的困境，在第四次工業革命後，生產現場已大幅減少人力投入。然而，產品製造完成後，其撿貨及發配貨物的過程尚未完全自動化，這意味著智慧倉儲將會是企業必須投入的重要項目之一。隨著IoT技術的迅速發展，自動化物流的研究領域正在不斷擴大及深化，吸引了來自物流管理、資訊科技、工業工程等多方面研究人員關注。

在此背景下，本研究旨在探索並實現一個高效的智慧倉儲系統。我們將運用馬達編碼器、紅外線測距技術和顏色感測器，已成功構建並實現了一個小型智慧倉儲系統。

### 二、研究動機

選擇進行智慧倉儲系統作為專題，是希望能透過實作深入學習自動化與感測技術，並將所學應用於實際的系統設計中。在本專題中，我們設計了一套完整的倉儲流程，包括利用 TCS3200 色彩感測模組進行物品辨識、輸送皮帶傳遞貨物、簡易機械手臂進行移載，以及以 AGV（自動導引車）完成最終的分類配送。這不僅讓我有機會接觸並操作多種硬體元件，更加強了我在系統整合與流程規劃方面的能力。透過這樣的專題實作，我能更具體了解智慧倉儲的實務應用與未來發展，也為我日後進入自動化、機電整合或智慧製造等相關領域打下良好基礎。因此，我期望藉由這次專題，累積跨領域的技術能力與實作經驗，提升自己在未來職場的競爭力。

## 貳、研究目的

### 一、提升倉儲操作效率

通過實現高度自動化，將大部分操作過程機械化，僅需在貨物放置階段依賴人力，從而顯著提升效率並降低成本。

### 二、探索智慧倉儲系統的實現方式

整合感測技術與控制系統，實現基本的分揀與配送功能，探索智慧倉儲系統的技術可行性與實際應用。

### 三、提高貨物定位精確度

利用高精度感測器與演算法提升貨物定位精度，確保操作準確性並提升整體作業效率。

### 四、簡化倉儲管理流程

藉由智能化系統優化倉儲流程，減少人工干預，提升管理效率與作業流暢度。

### 五、為未來研究奠定基礎

本研究為智慧倉儲系統的實作提供寶貴經驗，為後續自動化與智能物流技術的研究鋪路。

## 參、文獻探討

### 一、myRIO 相關知識

NI myRIO是一款功能強大的嵌入式開發平台，專為工程教育和創新應用而設計。本報告將詳細介紹myRIO的硬體規格、功能特色，並探討其在機器人控制領域中的廣泛應用。myRIO結合了即時處理器與可編程FPGA的優勢，為機器人設計提供了靈活且高效的控制解決方案，能夠處理從基本運動控制到複雜的自主導航等多種任務。

#### 1. NI myRIO 概述

NI myRIO是由美國國家儀器公司(National Instruments)開發的嵌入式重配置I/O (RIO)設備，主要用於教育和研究領域。這款設備專為幫助學生設計控制、機器人和機電一體化系統而設計，提供了一個完整的開發環境。

#### 2. 硬體架構

NI myRIO有兩種主要型號：myRIO-1900和myRIO-1950。這兩款產品核心功能相似，但外觀和某些接口細節有所不同。myRIO採用了Xilinx Zynq-7010系統單晶片，整合了雙核ARM Cortex-A9處理器和可重配置的FPGA。[2][3]

#### 3. LabVIEW 編程環境

NI myRIO主要使用LabVIEW進行編程，這是一種圖形化編程語言，具有以下特點：

- (1) 圖形化編程：LabVIEW 採用圖形化的編程環境，使用戶能夠通過連接不同的功能模塊來創建應用程序，減少了編程的複雜性。[4]
- (2) 模塊化設計：LabVIEW 支持模塊化的軟體設計流程，非常符合工程思維，能夠大大縮短開發和調試週期。[4]

(3) 三層編程架構：

I.即時系統層（LabVIEW RT）：用於實現確定性任務。

II.FPGA 層（LabVIEW FPGA）：用於實現高速且並行的 I/O 操作。

III.主機層：用於建立使用者介面和高級功能。[5][6]

4. 擴展性與兼容性

myRIO的FPGA隨附預先定義的功能，包括將數位I/O通道用作PWM、UART、編碼器輸入、I2C和SPI的功能。用戶可以通過LabVIEW FPGA Module修改預設屬性或創建全新的功能。[6]

此外，myRIO 還支持 C/C++編程，提供了更多的靈活性。[7]

5. 基礎運動控制

NI myRIO在機器人運動控制方面表現出色，特別適用於以下場景：

(1)馬達控制：

I.支持PWM輸出，最高頻率100 kHz，可用於控制直流馬達和伺服馬達。

II.數位I/O可用於控制H橋馬達驅動模組。

III.支持編碼器輸入（最高100 kHz），可實現閉環位置控制。[1][2]

(2)全向移動機器人：

I.可控制全向輪等特殊輪子，實現全向移動。

II.通過編碼器馬達提供精確的位置和速度控制。

III.搭配九軸陀螺儀感測器可修正機器自轉誤差。[1]

## 6. 教育與研究價值

NI myRIO在教育和研究領域有著特殊的價值：

### (1)教學工具；

I.單一裝置可滿足多種教學需求，包括控制、機器人、機電整合、嵌入式系統等。

II.圖形化的編程介面降低了學生的學習門檻。

III.豐富的教學資源和範例，有助於快速上手。[7]

### (2)專案開發平台：

I.自給自足的多功能工具，適合開發實用的專案

II.使用業界標準的軟硬體技術，幫助學生建立實際工程能力

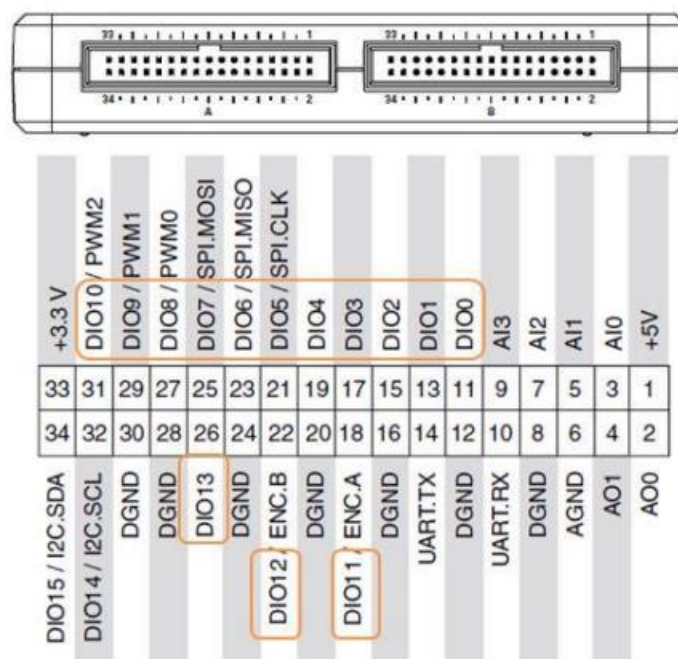
III.可用於畢業設計作品和創新競賽[7]

### (3)研究應用：

I.擴展性強，可連接現有的感測器和執行器

II.支持高級算法實現，適合研究複雜的控制系統

III.實時處理和FPGA的組合提供了高效能的計算平台[8]



(圖一)(星協系統科技有限公司)

## 二、 市面上倉儲系統

智慧倉儲機器人已成為現代物流、電商與製造業提升倉儲效率、降低人力成本的重要工具。隨著AI、3D視覺、自動導航等技術的成熟，市面上已有各種不同功能的智慧倉儲機器人，適用於揀貨、搬運、分揀、庫存管理等不同環節。

### 1. 應用場景

(1)電商與物流中心：自動揀貨、分揀、包裝與搬運，滿足大量訂單需求。[9][10][14]

(2)製造業：自動化物料搬運、庫存管理與夾取作業。[9][15]

(3)第三方物流與零售：提升倉儲空間利用率與作業彈性。[10]

### 2. 市場趨勢

智慧倉儲機器人市場持續成長，技術不斷創新，未來將朝向更智慧化、柔性化、客製化與全球化發展[9][10][14]。各大廠商持續推出新產品，協助企業實現倉儲全面智動化，提升競爭力。



### 三、 TCS3200 顏色感測相關知識

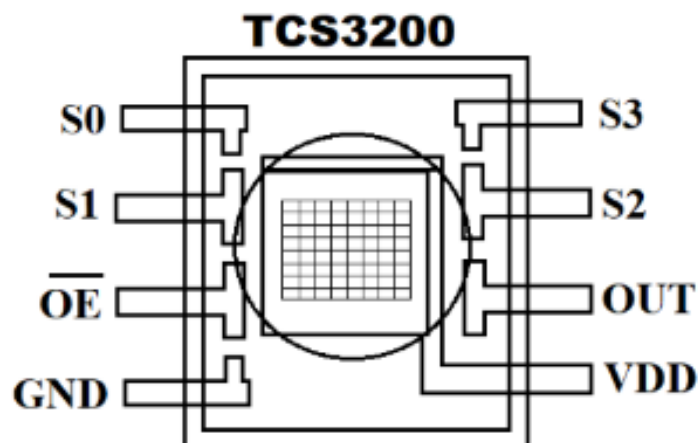
TCS3200是一種柔性的顏色光到頻轉換器，具有矽光電二極管和單個CMOS芯片中的電流到頻轉換器。它的核心功能是將檢測到的光強度轉換為方波，其頻率與光強度成正比。8X8光電二極管陣列具有16個用於紅色，綠色，藍色和未過濾光的光電二極管，從戰略上分佈，以確保一致的光檢測。相似色的光二極管並行連接，通過S2和S3鎖選擇了活性集。每種光電二極管的尺寸為110μm，散佈為134μm。(IC Components)

腳位	S0	S1	S2	S3
功能	頻率縮放 選擇	頻率縮放 選擇	光電二極體 選擇	光電二極體 選擇

(表一)

腳位	OE	OUT	VCC	GND
功能	輸出啟用	輸出頻率	供電	接地

(表二)

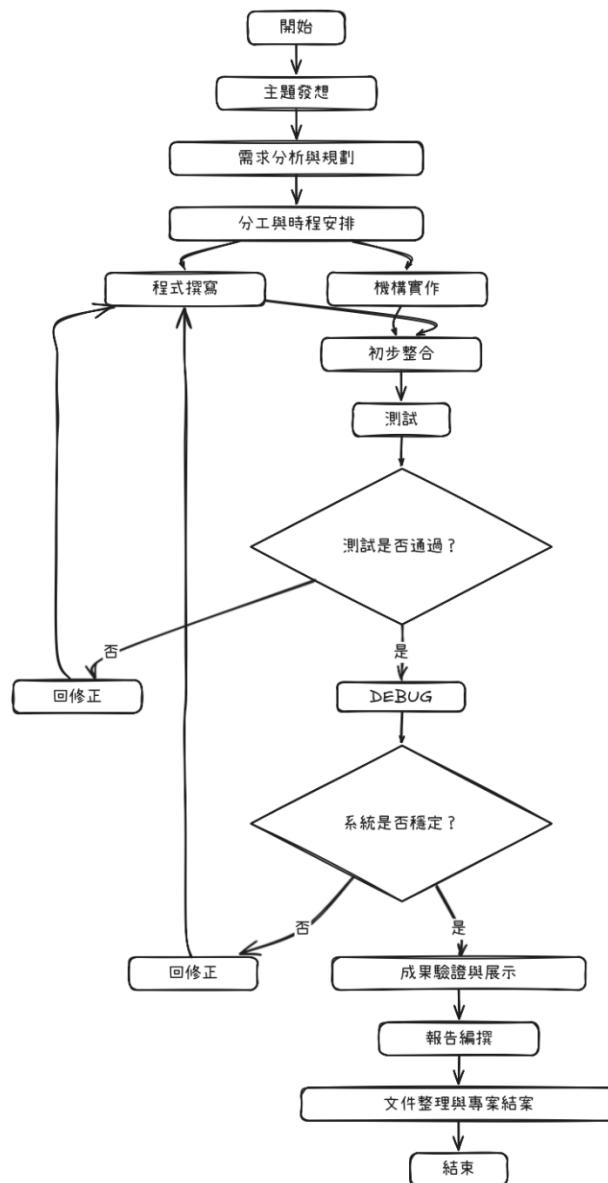


(圖二)( arduino portugal)

## 肆、研究方法

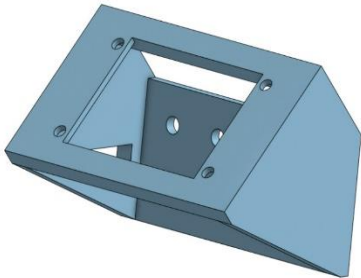

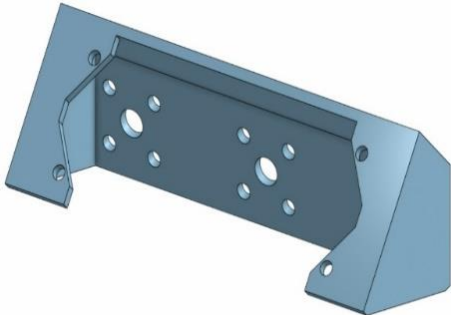

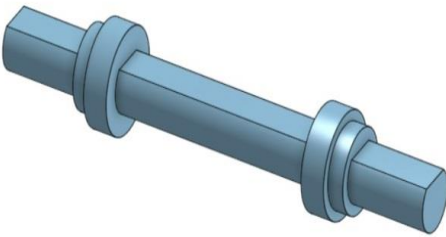
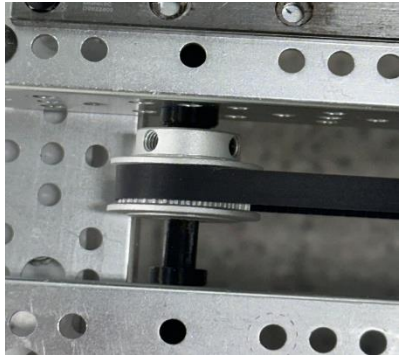
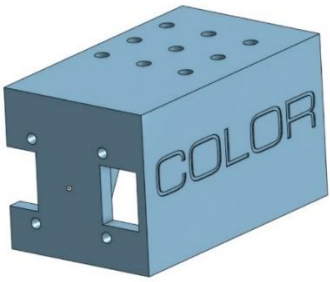
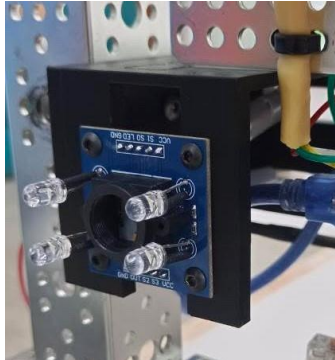
### 一、研究流程圖

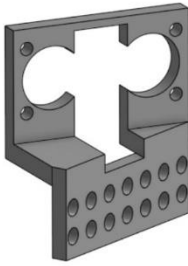
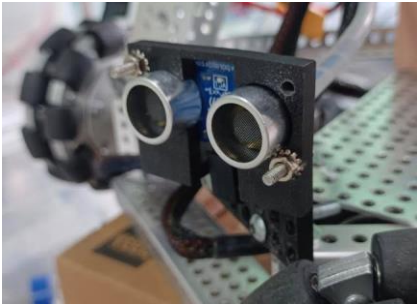
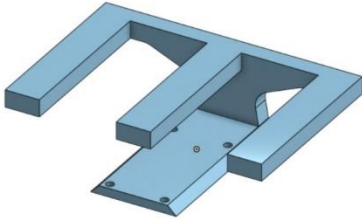
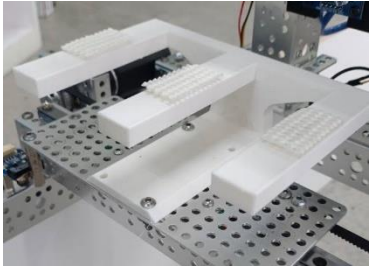
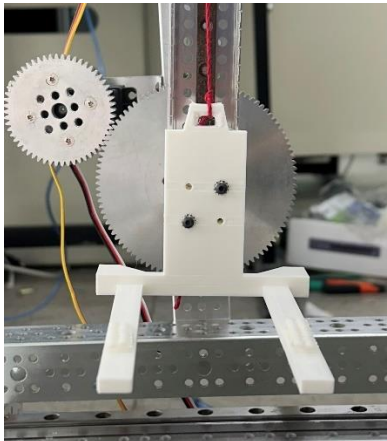
當初本組四人共同討論並確立專題主題後，隨即依據成員專長進行分工，分別負責機構設計與程式開發等工作。專題實作期間，團隊持續進行修正與優化，透過多次討論完善系統內容與架構。專題完成後，再將整體流程與成果系統化整理，撰寫成正式報告，以呈現本次專題實踐之完整歷程與成果。

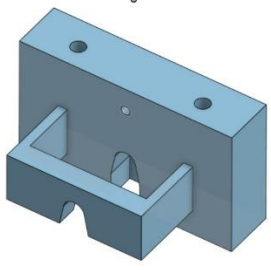
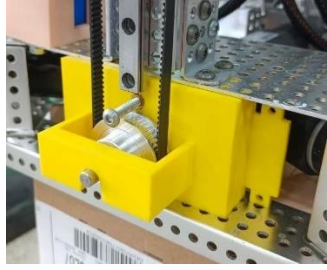
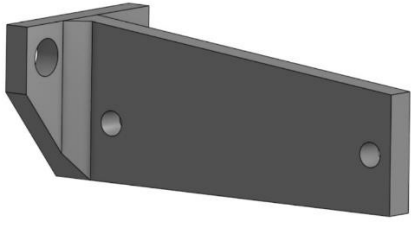
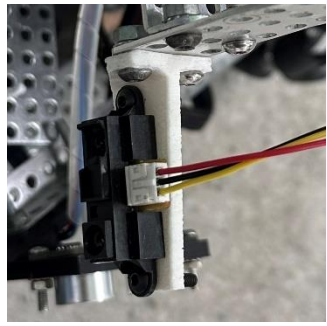


資料來源：研究者繪製

## 二、 設計 3D 列印件

使用位置	3D圖	實體位置
用於穩固 ST7735 顯示模組 之自製 3D列印 構件		
自行設計 之 3D 列印支架 用於穩固 MYRIO 開關組		
針對皮帶 安裝需求 設計的 3D 列印 功能件		
應用於 TCS3200 安裝之自 製 3D 列印支架		

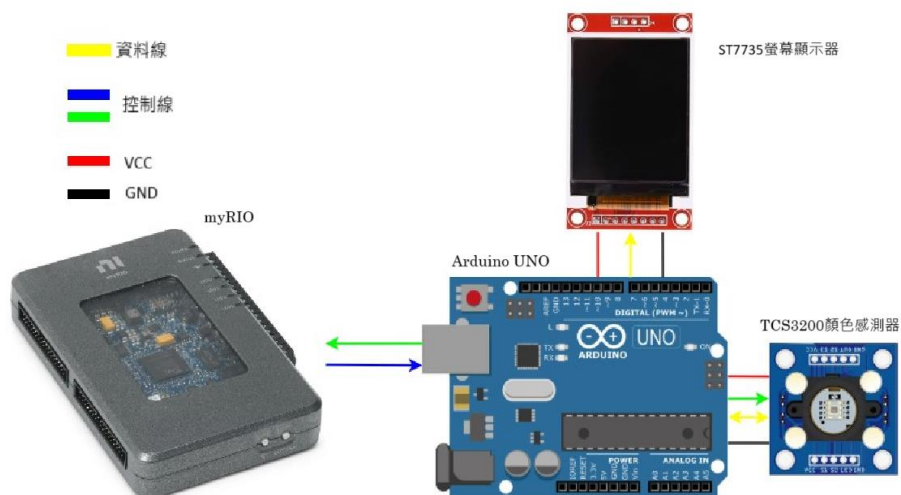
使用位置	3D圖	實體位置
超音波感測器對應之自製3D 列印安裝件		
作為貨物承載平台之 3D 列印件		
為 AGV 夾爪升降機構中皮帶與驅動馬達固定所設計之 3D 列印部件		
用於倉儲系統機械手臂叉取貨物之 3D 列印構件		

使用位置	3D圖	實體位置
用於 AGV 夾爪皮帶 固定的3D 列印組件		
專為固定 紅外線感 測器設計 的3D列印 部件		

(以上皆為研究者自製)

### 三、 電路示意圖

本系統以 Arduino UNO 為核心，結合 ST7735 彩色螢幕與 TCS3200 色彩感測器，實現倉儲物品顏色辨識與即時顯示功能。透過 TCS3200 擷取物品顏色後，由 UNO 處理並在 ST7735 顯示結果，同時透過序列通訊 (UART) 與 NI myRIO 傳輸感測資料，讓 myRIO 進一步進行分類控制與倉儲管理。此系統具備即時性與擴充性，適用於智慧倉儲應用。



(資料來源：研究者繪製)

## 四、Arduino 程式撰寫

### 1. 腳位定義及變數設值

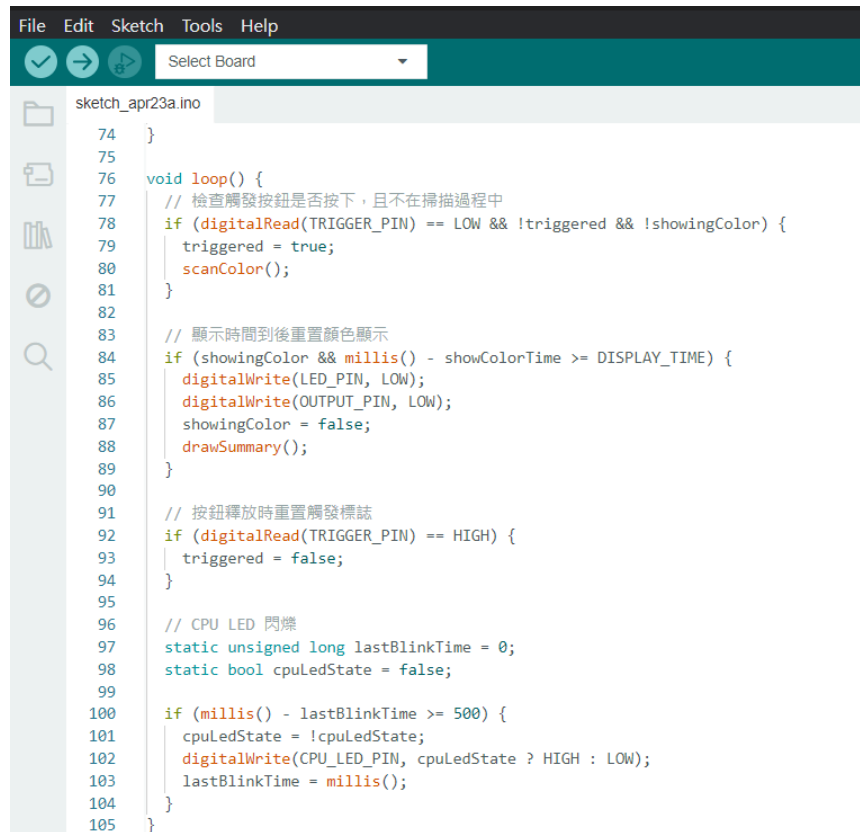
```
sketch_apr23a.ino
1 #include <Adafruit_GFX.h>
2 #include <Adafruit_ST7735.h>
3
4 // 腳位定義
5 #define TFT_CS 10
6 #define TFT_RST 9
7 #define TFT_DC 8
8 #define S0 3
9 #define S1 4
10 #define S2 5
11 #define S3 6
12 #define SENSOR_OUT 7
13 #define LED_PIN A0 // 顏色感測LED控制
14 #define CPU_LED_PIN A1
15 #define TRIGGER_PIN 12
16 #define OUTPUT_PIN 2
17
18 // 顏色常數
19 #define RED 0
20 #define GREEN 1
21 #define BLUE 2
22 #define COLORS_COUNT 3
23
24 // 時間常數
25 #define DISPLAY_TIME 2000
26 #define SAMPLING_DELAY 50
27 #define SAMPLING_COUNT 5
28 #define STABILIZATION_DELAY 200
29
30 Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);
31
32 // 顏色值和計數
33 const uint16_t colors[COLORS_COUNT] = {ST77XX_RED, ST77XX_GREEN, ST77XX_BLUE};
34 uint8_t quantities[COLORS_COUNT] = {0, 0, 0};
35 const char* colorNames[COLORS_COUNT] = {"R", "G", "B"};
36
37 // 狀態變數
38 bool triggered = false;
39 unsigned long showColorTime = 0;
40 bool showingColor = false;
41 int8_t detectedColorIndex = -1;
```

### 2. 設定感測器初始值及狀態

```
File Edit Sketch Tools Help
sketch_apr23a.ino
41 int8_t detectedColorIndex = -1;
42
43 void setup() {
44   // 初始化 TFT 顯示器
45   tft.initR(INITR_BLACKTAB);
46   tft.setRotation(3);
47   tft.fillScreen(ST77XX_BLACK);
48   tft.setTextColor(ST77XX_WHITE);
49   tft.setTextSize(2);
50
51   pinMode(S0, OUTPUT);
52   pinMode(S1, OUTPUT);
53   pinMode(S2, OUTPUT);
54   pinMode(S3, OUTPUT);
55   pinMode(SENSOR_OUT, INPUT);
56   pinMode(LED_PIN, OUTPUT);
57   pinMode(CPU_LED_PIN, OUTPUT);
58   pinMode(TRIGGER_PIN, INPUT_PULLUP);
59   pinMode(OUTPUT_PIN, OUTPUT);
60
61   digitalWrite(S0, HIGH);
62   digitalWrite(S1, LOW);
63
64   // 啟動時閃爍 LED
65   digitalWrite(LED_PIN, HIGH);
66   delay(500);
67   digitalWrite(LED_PIN, LOW);
68
69   // 初始化 LED 和輸出腳位
70   digitalWrite(OUTPUT_PIN, LOW);
71   digitalWrite(CPU_LED_PIN, LOW);
72
73   drawSummary();
74 }
```



### 3. 檢查觸發訊號狀態



```
File Edit Sketch Tools Help
Select Board

sketch_apr23a.ino
74 }
75
76 void loop() {
77   // 檢查觸發按鈕是否按下，且不在掃描過程中
78   if (digitalRead(TRIGGER_PIN) == LOW && !triggered && !showingColor) {
79     triggered = true;
80     scanColor();
81   }
82
83   // 顯示時間到後重置顏色顯示
84   if (showingColor && millis() - showColorTime >= DISPLAY_TIME) {
85     digitalWrite(LED_PIN, LOW);
86     digitalWrite(OUTPUT_PIN, LOW);
87     showingColor = false;
88     drawSummary();
89   }
90
91   // 按鈕釋放時重置觸發標誌
92   if (digitalRead(TRIGGER_PIN) == HIGH) {
93     triggered = false;
94   }
95
96   // CPU LED 閃爍
97   static unsigned long lastBlinkTime = 0;
98   static bool cpuLedState = false;
99
100  if (millis() - lastBlinkTime >= 500) {
101    cpuLedState = !cpuLedState;
102    digitalWrite(CPU_LED_PIN, cpuLedState ? HIGH : LOW);
103    lastBlinkTime = millis();
104  }
105 }
```

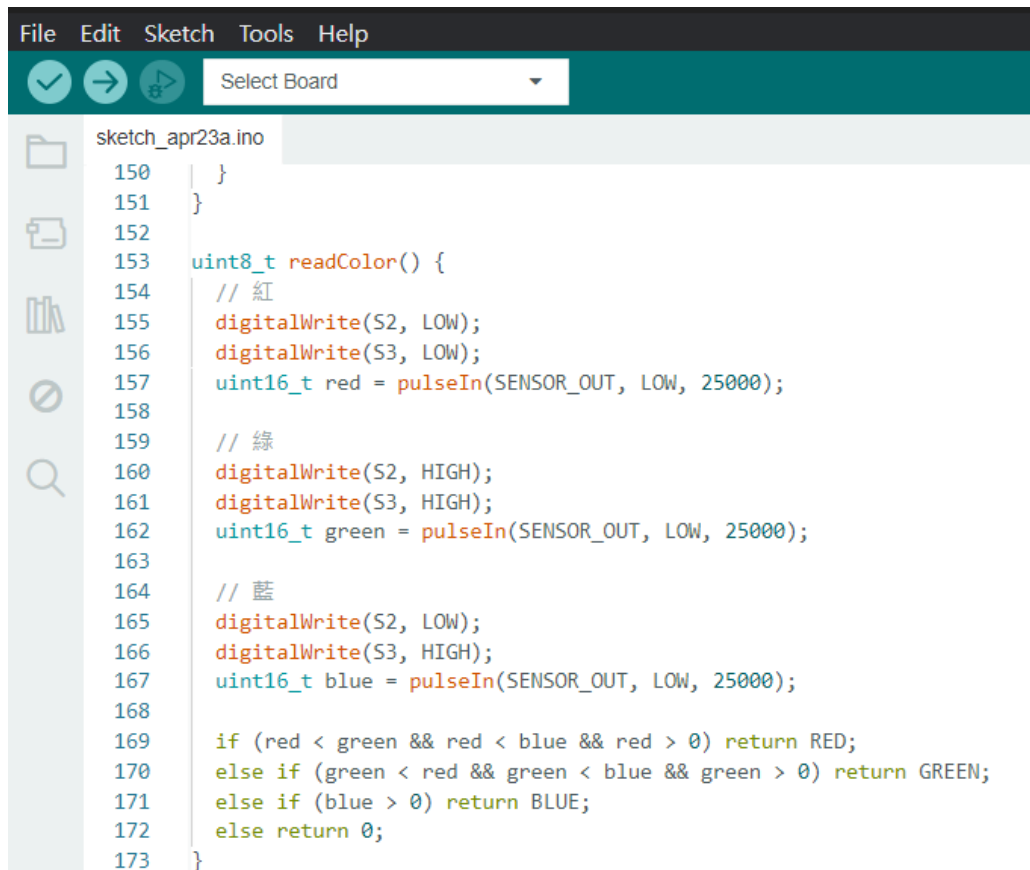
### 4. 顏色辨識及防誤判邏輯



```
File Edit Sketch Tools Help
Select Board

sketch_apr23a.ino
107 void scanColor() {
108   // 開啟 LED 以提高顏色檢測準確度
109   digitalWrite(LED_PIN, HIGH);
110
111   // 等待感測器在 LED 開啟後穩定
112   delay(STABILIZATION_DELAY);
113
114   // 進行多次採樣以提高準確度
115   uint8_t colorVotes[COLORS_COUNT] = {0};
116
117   for (uint8_t i = 0; i < SAMPLING_COUNT; i++) {
118     uint8_t detectedColor = readColor();
119     if (detectedColor < COLORS_COUNT) {
120       colorVotes[detectedColor]++;
121     }
122     delay(SAMPLING_DELAY);
123   }
124
125   // 找出投票最多的顏色
126   uint8_t maxVotes = 0;
127   detectedColorIndex = -1;
128
129   for (uint8_t i = 0; i < COLORS_COUNT; i++) {
130     if (colorVotes[i] > maxVotes) {
131       maxVotes = colorVotes[i];
132       detectedColorIndex = i;
133     }
134   }
135
136   if (detectedColorIndex >= 0 && maxVotes > (SAMPLING_COUNT / 3)) {
137     quantities[detectedColorIndex]++;
138     tft.fillScreen(colors[detectedColorIndex]);
139
140     tft.setCursor(tft.width()/2 - 20, tft.height()/2 - 10);
141     tft.setTextColor(ST77XX_WHITE);
142     tft.setTextSize(3);
143     tft.print(colorNames[detectedColorIndex]);
144
145     digitalWrite(OUTPUT_PIN, HIGH);
146     showingColor = true;
147     showColorTime = millis();
148   } else {
149     digitalWrite(LED_PIN, LOW);
150   }
```

## 5. 顏色辨識結果



```

150     }
151 }
152
153 uint8_t readColor() {
154     // 紅
155     digitalWrite(S2, LOW);
156     digitalWrite(S3, LOW);
157     uint16_t red = pulseIn(SENSOR_OUT, LOW, 25000);
158
159     // 綠
160     digitalWrite(S2, HIGH);
161     digitalWrite(S3, HIGH);
162     uint16_t green = pulseIn(SENSOR_OUT, LOW, 25000);
163
164     // 藍
165     digitalWrite(S2, LOW);
166     digitalWrite(S3, HIGH);
167     uint16_t blue = pulseIn(SENSOR_OUT, LOW, 25000);
168
169     if (red < green && red < blue && red > 0) return RED;
170     else if (green < red && green < blue && green > 0) return GREEN;
171     else if (blue > 0) return BLUE;
172     else return 0;
173 }
    
```

## 6. 將色辨識結果顯示在螢幕上



```

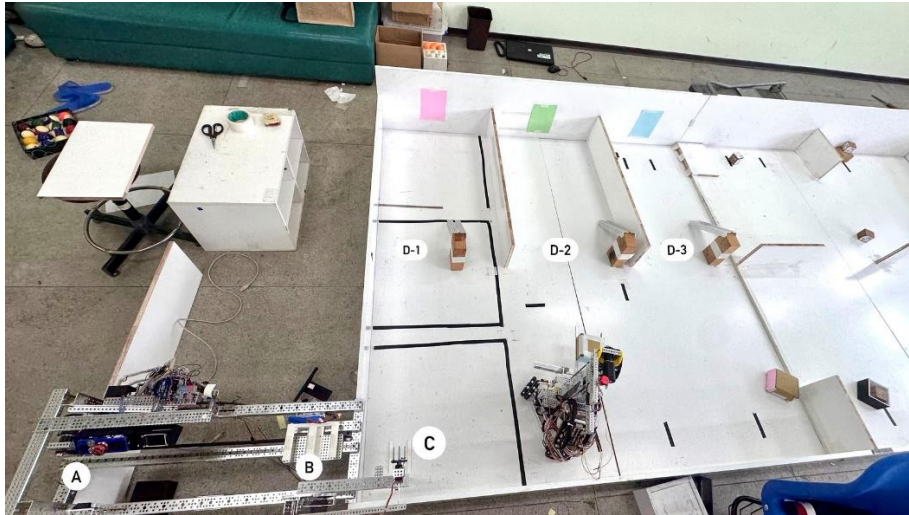
175 void drawSummary() {
176     tft.fillScreen(ST77XX_BLACK);
177
178     uint8_t sectionWidth = tft.width() / COLORS_COUNT;
179     uint8_t sectionHeight = tft.height() / 2;
180
181     for (uint8_t i = 0; i < COLORS_COUNT; i++) {
182         uint8_t x = i * sectionWidth;
183
184         tft.fillRect(x, 0, sectionWidth, sectionHeight, colors[i]);
185
186         tft.fillRect(x, sectionHeight + 2, sectionWidth, tft.height() - sectionHeight - 2, ST77XX_BLACK);
187
188         tft.setCursor(x + sectionWidth/2 - 10, sectionHeight + 10);
189         tft.setTextColor(ST77XX_WHITE);
190         tft.setTextSize(2);
191         tft.print(colorNames[i]);
192
193         tft.setCursor(x + 5, sectionHeight + 40);
194         tft.setTextSize(3);
195         tft.print(quantities[i]);
196     }
197 }
198
    
```



## 伍、研究分析及結果

### 一、 自動化搬運流程說明

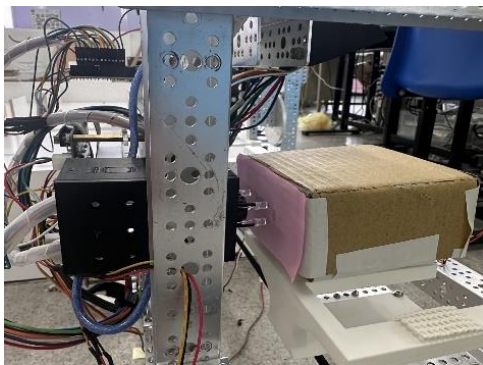
貨物首先置於 A 區，由 TCS3200 色彩感測模組進行顏色辨識與數據擷取。完成感測後，貨物透過輸送皮帶自動移動至 B 區，再由簡易機械手臂執行定位與搬運作業，將其移送至 C 區。最終，AGV（自動導引車）根據辨識結果，自動將貨物運送至對應的分類區域，完成整體智慧分揀流程。



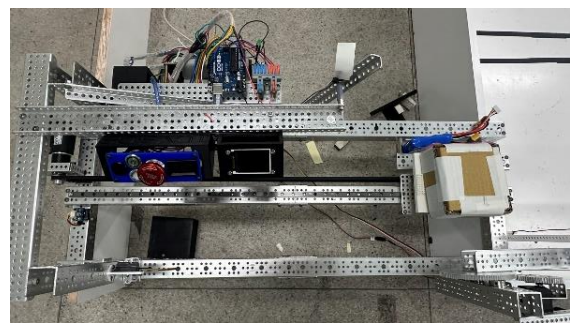
(上圖為場地圖)

### 二、運作過程步驟分解


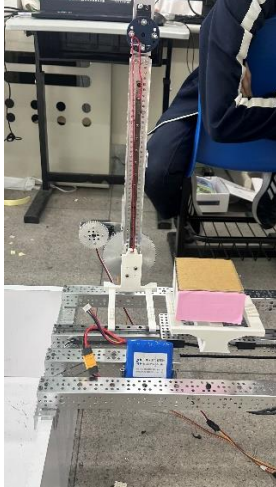
1.使用 TCS3200 模組偵測貨物顏色，轉換為數位訊號供後續分類判斷。



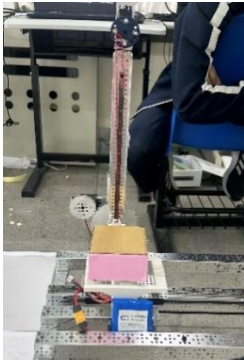
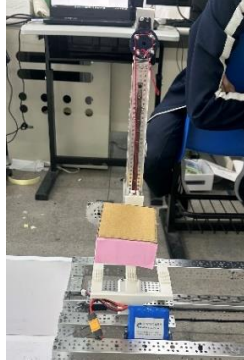
2.於起點加裝極限開關，作為判斷傳輸起始位置的感測機制，以控制輸送行程。



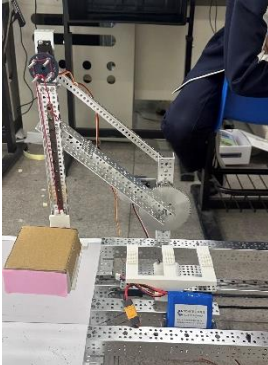
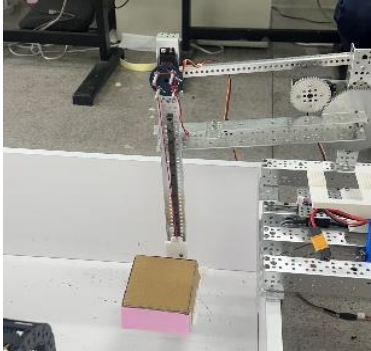
(本圖以紅色貨物進行展示)

3.機械手臂待命定位	4.貨物即將進入機械手臂作業區
	

(本圖以紅色貨物進行展示)

5. 貨物通過皮帶輸送，準確到達機械手臂的操作範圍，準備進行搬運。	6.機械手臂啟動上升動作，準備將貨物移至指定位置。
	

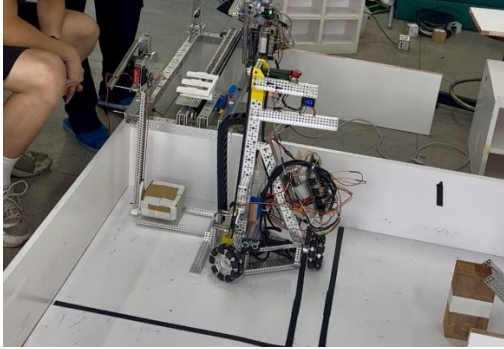
(本圖以紅色貨物進行展示)

7. 機械手臂啟動左移動作，將貨物移至目標位置。	8.機械手臂啟動下降動作，將貨物精確放置於指定位置。
	

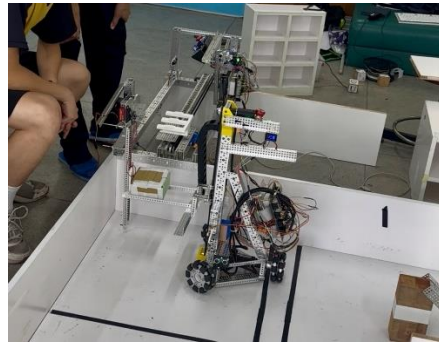
(本圖以紅色貨物進行展示)



9. 機械手臂將貨物準確移送至 AGV，準備進行後續運輸。

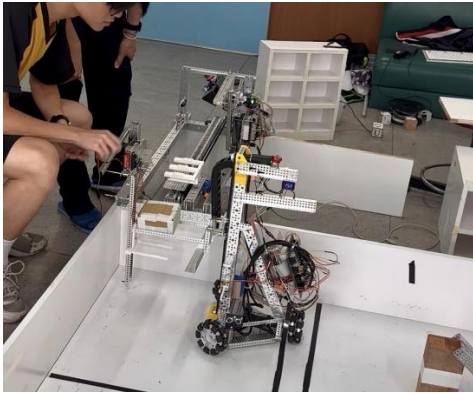


10. AGV 夾爪啟動上升動作，準備抓取並固定貨物。

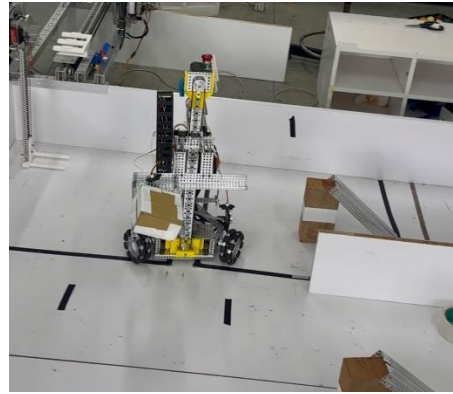


(本圖以綠色貨物進行展示)

11. AGV 夾爪完成貨物抓取後，與機械手臂安全分離，準備進行運輸。

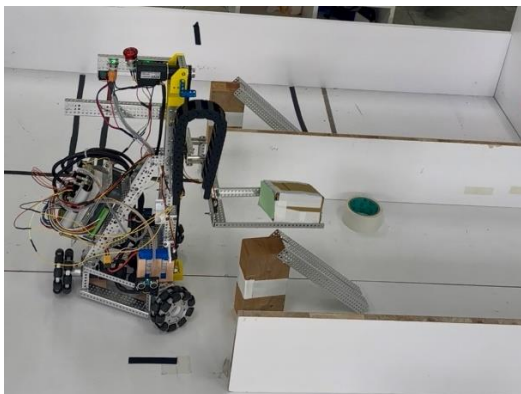


12. AGV 根據預設路徑自主運行，將貨物送至指定分類區域。

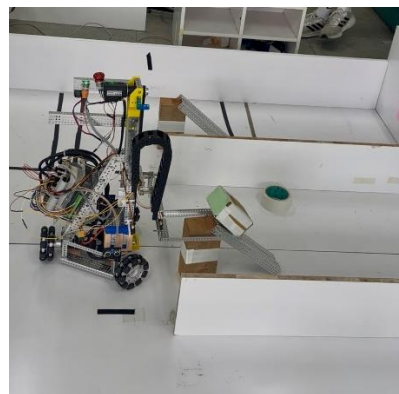


(本圖以綠色貨物進行展示)

13. AGV 根據路徑規劃，自主移動至預定的集貨區域，準備進行卸貨作業。

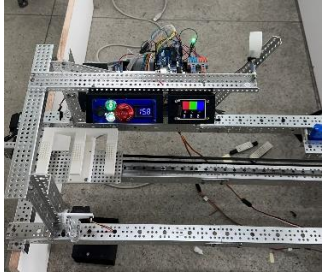
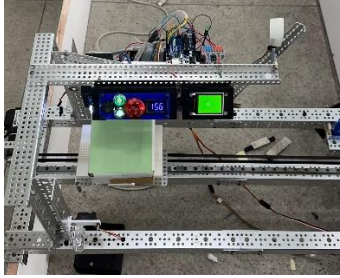
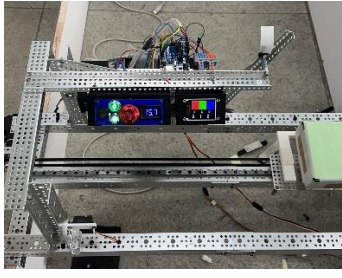


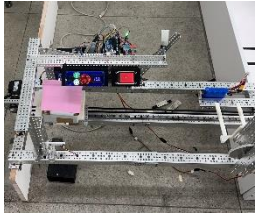
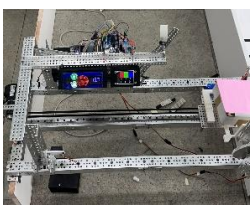
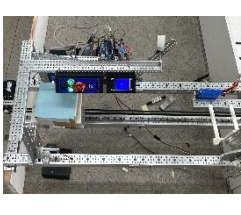
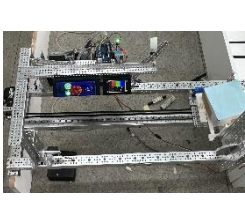
14. AGV 夾爪下降，將貨物精確放置於指定位置，完成卸貨操作。



(本圖以綠色貨物進行展示)

### 三、 色彩辨識螢幕顯示

1. 尚未放置任何物品	2. 感測綠色貨物	3. 綠色顯示數值遞增 1
		

4. 感測紅色貨物	5. 紅色顯示數值 遞增 1	6. 感測藍色貨物	7. 藍色顯示數值 遞增 1
			

### 四、 技術瓶頸與解決辦法

**問題(一)：**機械手臂夾爪的上下移動。主機構手的想法一直卡在想用皮帶帶動，但因為會卡到，所以一直無法實現，一度想捨棄整個機械手臂。

**解決方法：**

副機構手提出了用紗線固定夾爪，再使用馬達帶動，達成上下移動的效果。

**問題(二)：**顏色感測結果存在一定誤差，無法完全準確辨識目標顏色。

**解決方法：**

1. 對 TCS3200 模組周圍的四組光源進行角度微調，以降低光線過度聚焦所造成的干擾，提升感測均勻性。
2. 原先僅進行一次顏色掃描，現優化為進行十次感測並取出出現次數最多的結果，以提高資料穩定性與準確性。

## 陸、研究結論與建議

本研究所開發之智慧型分揀裝置，歷經一個多月的設計、整合與測試，已成功完成初步目標，具備辨識、搬運與分類小型貨物的功能。系統結合色彩感測模組（TCS3200）、彩色顯示器（ST7735）、簡易機械手臂與自動導引車（AGV），建立一套從感測辨識、螢幕顯示、機械搬運到分類運輸的自動化流程，展現良好的實作成果與應用潛力。

整體運作流程清晰：貨物進入感測區後，由 TCS3200 進行顏色辨識，結果即時顯示於螢幕上，並依辨識結果分類記錄；接著透過輸送皮帶將貨物傳送至機械手臂作業區，由手臂搬運至 AGV；AGV 再根據辨識結果，自主行駛至對應的集貨區完成分類。此流程模擬智慧倉儲中的基本分揀作業，具備教學價值與技術延伸性。

在實作過程中，我們也發現部分設計仍有優化空間。以機械手臂與 AGV 的交接機構為例，目前採用 AGV 夾爪上升托住貨物，使機械手臂自動釋放的方式雖簡便有效，惟傳動結構以紗線模擬皮帶，於搬運較重或體積較大貨物時，容易出現張力不穩、定位誤差或傳動失效等情形，影響交接穩定性與作業效率。

為提升系統可靠性與擴展性，建議未來在硬體方面導入更穩定的傳動機構，如高強度同步帶、線性滑軌或電動驅動模組，以提升承載能力與運作精度；在控制系統方面，可整合感測回饋與閉迴路控制技術，以強化整體精準度與容錯能力，並提升模組協作的效率與靈活性。此外，通訊協定與模組同步控制邏輯亦可優化，朝向多模組協作與即時反應的方向發展。

整體而言，本系統已建構出具備基本辨識與分類功能的智慧分揀流程，對未來智慧倉儲自動化具有良好參考價值。透過持續優化硬體架構與控制演算法，未來本裝置可望應用於更大規模、需求更複雜的智慧物流場域，成為具實用性與發展潛力的自動化倉儲解決方案。

## 柒、資料來源

- [1] [PDF] 語音控制、視覺辨識、機器人 <https://twsf.ntsec.gov.tw/activity/race-1/57/pdf/052306.pdf>
- [2] [PDF] NI myRIO-1900 User Guide and Specifications - National Instruments  
<https://docs-be.ni.com/bundle/myrio-1900-getting-started/raw/resource/enus/376047d.pdf>
- [3] 19 [PDF] NI myRIO-1950 User Guide and Specifications - WPC Systems Ltd.  
<https://www.wpc.com.tw/uploads/2/5/6/6/25663802/myrio-1950.pdf>
- [4] 基于NI myRIO 的智能家居设计\_labview作品 - CSDN博客  
[https://blog.csdn.net/qq\\_42078934/article/details/104671041](https://blog.csdn.net/qq_42078934/article/details/104671041)
- [5] Implementation overview of a novel approach to Smart Microgrid Real Time Simulation  
<https://www.semanticscholar.org/paper/4b1fbc2b2922dd543cae8b192f1b3447b18df41b>
- [6] NI myRIO 常見問答集 <https://www.ni.com/zh-tw/support/documentation/supplemental/13/ni-myrio-frequently-asked-questions.html>
- [7] [PDF] NI myRIO 學生快速實作工程的最佳利器  
[https://www.ni.com/content/dam/web/pdfs/Taiwan\\_Traditional\\_Chinese\\_Manual\\_myRIO.pdf](https://www.ni.com/content/dam/web/pdfs/Taiwan_Traditional_Chinese_Manual_myRIO.pdf)
- [8] Experience Report on Teaching Distributed Testing Bench for Cloud-Based Applications  
<https://www.semanticscholar.org/paper/a82a3d2e33ad4b9e0ff4fa4ca7dcbc4b3ff8147e>
- [9] JustPick 智能分揀系統 – 智慧物流解決方案 - SOLOMON 3D  
<https://www.solomon-3d.com/tw/products/justpick/>

[10] 海柔创新：市占率超90%，箱式仓储机器人领航者的五年奋斗史

<https://www.hairobotics.com/tw/news/case-handling-warehouse-robot-pioneer>

[11][專欄] 物流黑科技AMR 機器人的發展

<https://www.logisticnet.com.tw/publicationArticle.asp?id=1352>

[12] TAIROA 台灣智慧自動化與機器人協會Taiwan Automation ...

[https://www.tairoa.org.tw/column/column.aspx?column\\_id=2622&column\\_type\\_id=1](https://www.tairoa.org.tw/column/column.aspx?column_id=2622&column_type_id=1)

[13]運輸機器人跨足新領域:運輸機器人,AGV,3D視覺,洛克威爾,UL - 智動化

<https://smartauto.ctimes.com.tw/DispArt-tw.asp?O=HK5BT6GBXIGARASTDE>

[14]自動化倉儲| AMR智能物流機器人 - Unitech

<https://www.ute.com.tw/solution/category/Warehousing/303>

[15]運輸機器人跨足新領域:運輸機器人,AGV,3D視覺,洛克威爾,UL - 智動化

<https://smartauto.ctimes.com.tw/DispArt-tw.asp?O=HK5BT6GBXIGARASTDE>