

手機遙控取物自走車之研製

中華科技大學 電機工程系 林璟耀、邱駿騰、簡駿任

謝宗煌、周書碩

臺北市立南港高級工業職業學校 圖書館主任 丁一能 推薦

摘 要

物聯網與工業 4.0 結合，是當前與未來自動化發展的重要趨勢，而機器人在其中更扮演重要功能。本研究旨在研製手機遙控取物自走車，以手機藍芽裝置搭配 Arduino 作為控制核心進行遙控取物。

首先設計好各部分組件功能，確定功能後進行整合，先完成自走車功能測試，再進行機器手臂組裝並進行功能測試，最後將機器手臂與自走車組合測試，其中包括藍芽軟體、內建程式測試、自己設計的 C 語言及 Arduino 程式，最後進行整體的整合執行狀態測試。依據手機傳送命令訊息給機器執行，當執行成功時會進入等待狀態，等待手機端給予指令，在執行完每一個動作時都會回到等待區，達到手機遙控自走車到目標及取物之功能，整體實測結果令人滿意。

關鍵字：物聯網，機器手臂，App Inventor，Arduino

壹、簡介

一、動機與目的

最近熱門的物聯網與工業 4.0 結合是當前與未來的自動化發展趨勢，因此機器人在其中更扮演重要功能。本文的發想乃根據手機的便利性加上自走車的移動性與機器手臂的操作性進行專題研究，以小型的機體為開發選項，使用手機 App Inventor 研發操作界面，並且以 HC-06 藍芽裝置為通訊裝置，利用 Arduino 的擴展性當作核心元件，挑選伺服馬達來驅動機器手臂可達成精確動作，選擇 L298n 驅動晶片來提昇直流馬達的驅動扭力，完成手機遙控自走車到達目標取物之功能。

本研究之目的可以做為物件搬運或特殊構造之物流倉儲，或是特殊環境在人體不易進入的小空間進行物件之夾取以期待擴大機器控制與應用，本文預期完成透過手機遙控自走車到達目標取物之功能。

二、系統架構方塊圖說明

系統架構方塊如圖 1 所示，手機 App Inventor 傳送訊號給藍芽連線，確認連結成功後，傳送訊號進入 Arduino 端，此控制核心可以傳送指令給伺服馬達驅動機器手臂或送指令給 L298n 驅動晶片操作自走車。

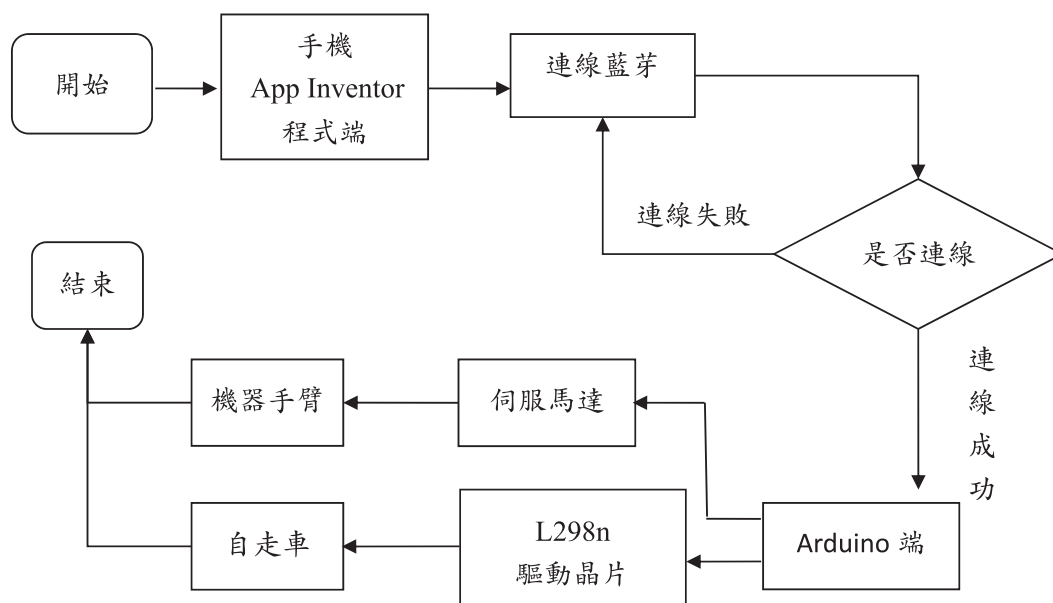


圖 1 系統架構方塊圖

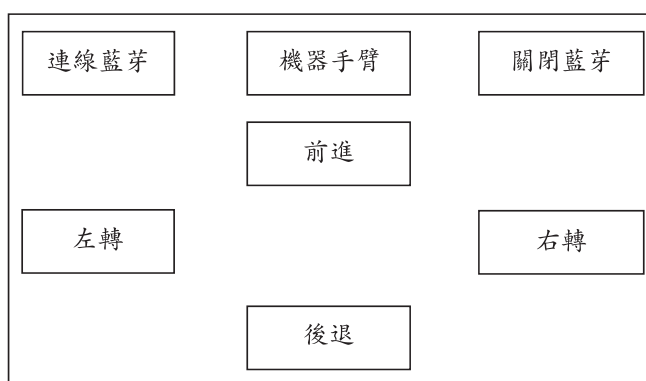
貳、手機 App Inventor、藍芽流程圖及程式設計

一、App Inventor 應用介紹

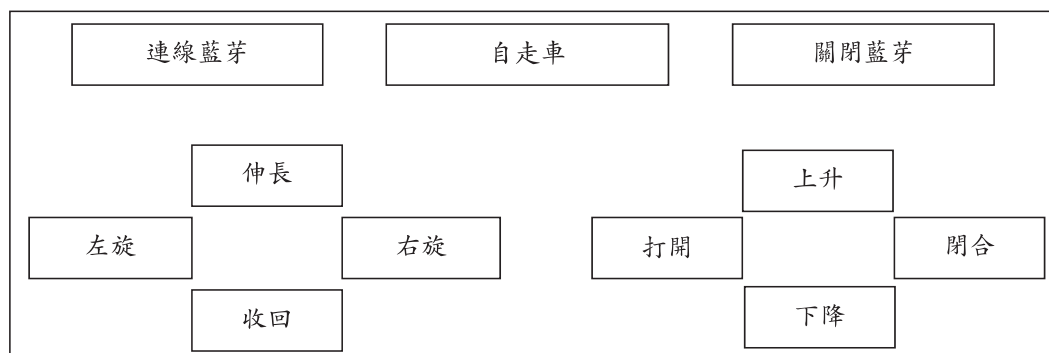
App Inventor 是起先由 Google 提供的應用軟體，現在由麻省理工學院維護及營運。它可以讓任何熟悉或不熟悉程序設計的人來創造基於 Android 作業系統的應用軟體。使用圖形化界面來創造一個運行在 Android 系統上的應用，它就可以在許多手機設備上運作。

二、手機設計界面圖示與說明

依圖 2 所示，在數字界面可以依照圖 2(a)(b) 看出動作所對應的指令，在手機端可選擇需要連結的自走車界面或是機器手臂界面，也可以關閉藍芽和連線藍芽。



(a)



(b)

圖 2 手機設計界面圖

實作 App Inventor 手機自走車界面端之畫面

依圖 3 指示，在界面可以依照圖 3 看出動作所對應圖 4 的指令，在自走車界面可以操作自走車的前進、後退、左轉和右轉，手機自走車界面端畫面。



圖 3 手機自走車界面端畫面

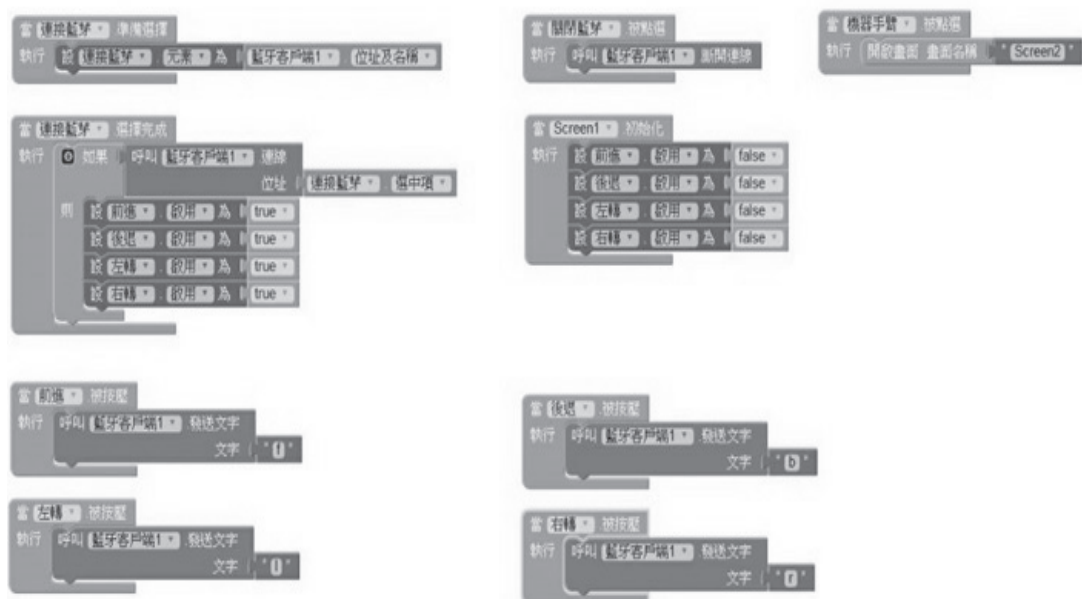


圖 4 App Inventor 手機自走車程式碼畫面

實作 App Inventor 手機機器手臂界面端之畫面

依圖 5 指示，在界面可以依照圖 5 看出動作所對應圖 6 的指令，在機器手臂界面可以操作機器手臂的伸長、收回、左旋、右旋、上升、下降、打開和閉合指令。



圖 5 手機機器手臂界面端畫面

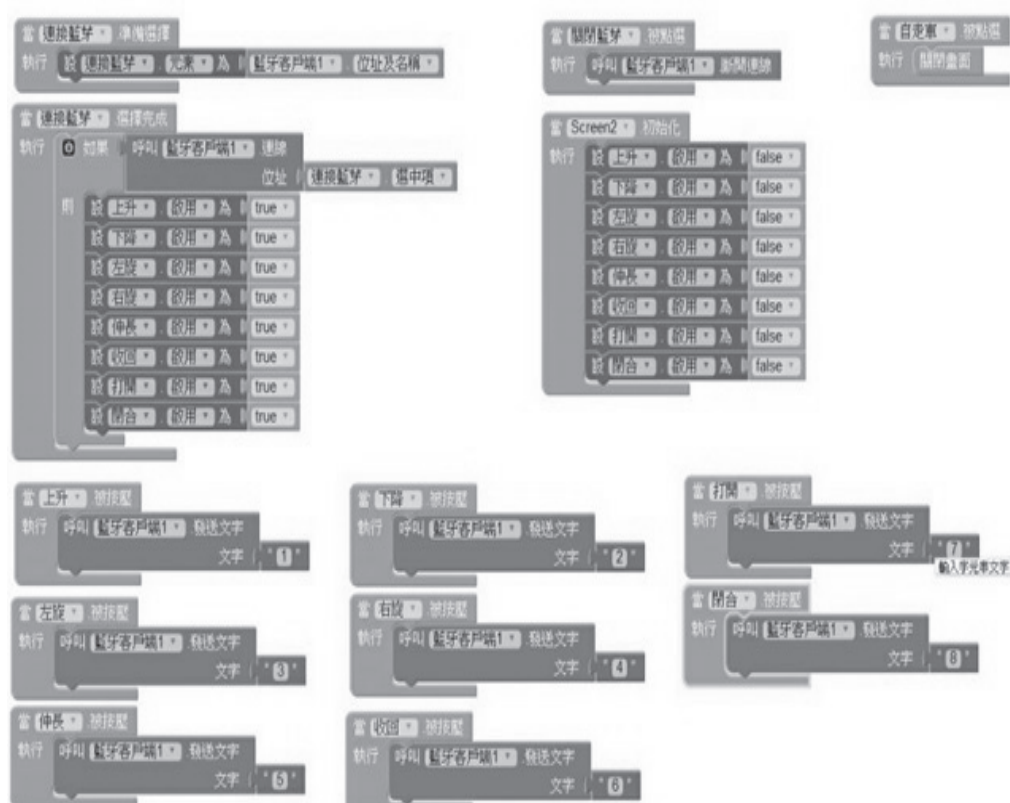


圖 6 App Inventor 手機機器手臂程式碼畫面

三、藍芽裝置

如圖 7 所示的藍芽裝置，一般的 Arduino 藍牙遙控和通訊實驗，是用「從端」模式，所以 HC-06 模組足以應付大多數需求，接線只需要 4 條線：電源、接地、傳送 (TXD) 和接收 (RXD)，主控端或從端模式，出廠前已設定好，不能更改。。

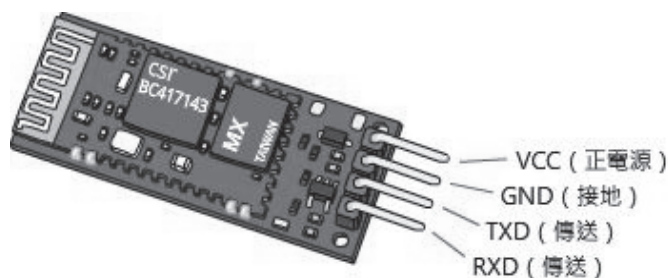


圖 7 HC-06 接腳圖 [4]

四、藍芽連線流程圖及說明

依圖 8，手機藍芽發出訊號到 HC-06 藍芽裝置連線，連線成功 HC-06 上的連線狀態指示燈會恆亮，但當連結失敗時，HC-06 上的連線狀態指示燈會閃爍。

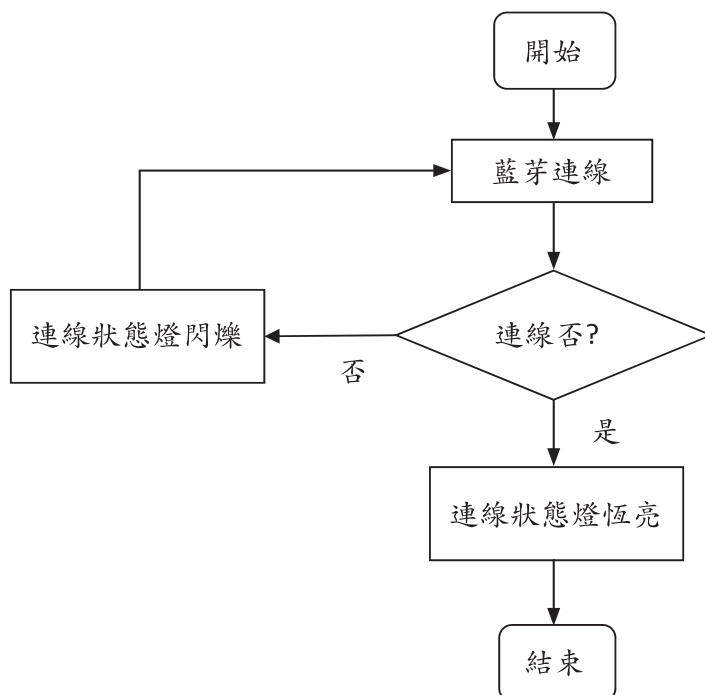


圖 8 藍芽連線流程圖

參、Arduino 應用介紹

Arduino 微控器的許多接腳身兼兩種用途，像類比 A4 和 A5 腳，也是 I2C 介面的 SDA 和 SCL 腳位。新增的 SCL 和 SDA 插孔，分別和類比腳位 A5 與 A4 相連，新增這兩個插孔的好處是，做實驗時從板子的標示（註：標示在板子背後），能立即分辨 SCL 和 SDA。IOREF（輸出入埠電壓參考）插孔和 5V（電源輸出）插孔相連，它可以讓擴充板（Shield）得知此 Arduino 板的運作電壓。IOREF 左邊插孔是空接，所以沒有任何作用，未來的控制板也許會用到此插孔。

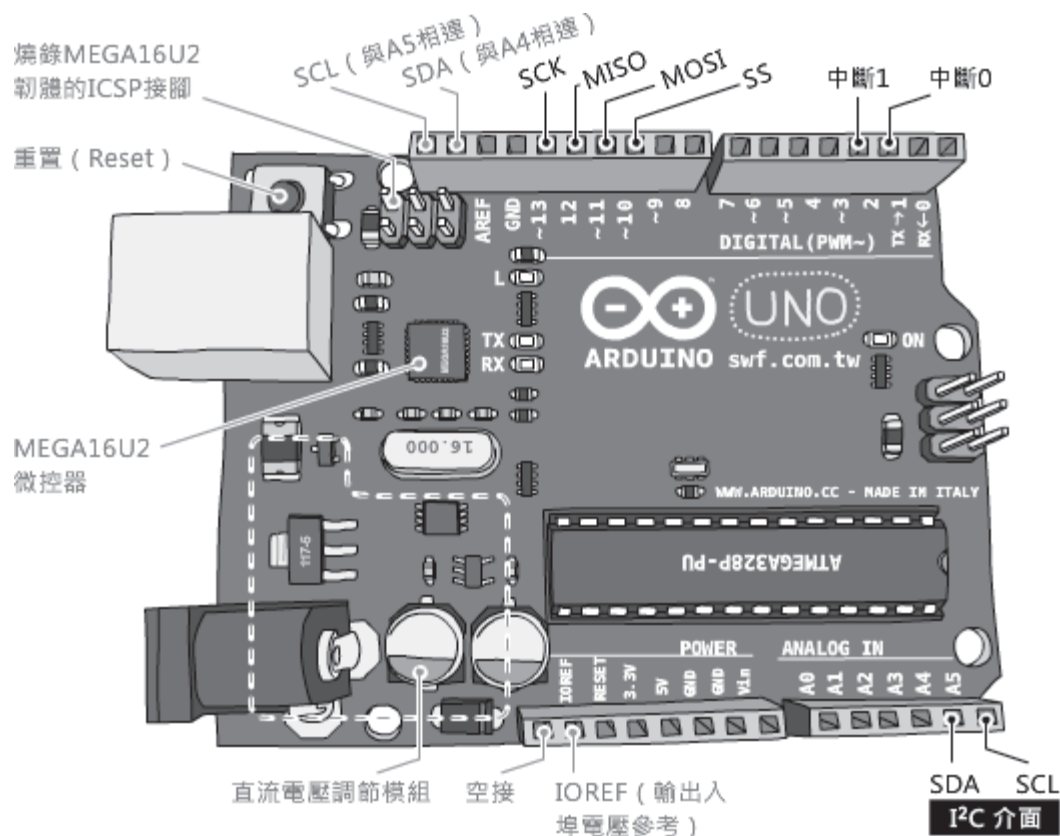


圖 9 Arduino 接腳圖 [9]

肆、硬體與流程圖說明

使用手機 App inventor 通過 HC-06 藍芽裝置下達指令 Arduino，再由 Arduino 分類指令種類去挑選伺服馬達來驅動機器手臂或是選擇 L298n 驅動晶片來驅動直流馬達操作自走車。定位正確與否則是依靠人工去觀測而不是由機器去判斷，所以使用寸動的方式才能達成目的，而寸動的標準大約是把可動範圍切割成 2 份至 6 份，這樣的好處是可以明確了解目前移動的範圍，缺點則是精度沒有完全發揮出來，正常的伺服馬達可以做更精細，這方面的精密調整也是需要再研發。

一、自走車流程圖及說明

依圖 10 所示，在手機端給予等待指令的自走車動作指令，並且自走車接收到指令動作完後，又回到等待指令區，等待下一個指令。

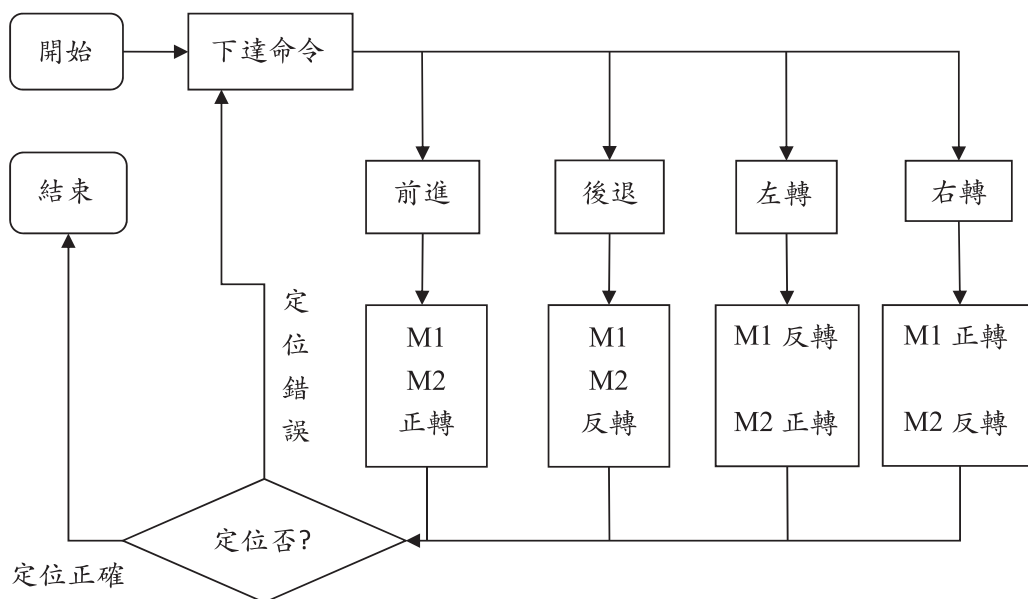


圖 10 自走車流程圖

二、機器手臂介紹

這次選擇的機器手臂是 mearm v1.0 版的機械手臂，內含 4 顆伺服馬達與組裝模型，伺服馬達採用的是 sg-90。

三、機器手臂操作示意流程圖說明

依圖 11 所示，在 Arduino 端先從開始時傳送機器手臂歸零指令，接收到時，傳送到藍芽端，在藍芽端接收到指令時，等待手機端傳送指令，沒接收到時，再重新發送藍芽訊息，當下達指令完再回到等待指令中。

◎機器手臂校正：

當啟動 Arduino 時，在機械手臂端會接收到由 Arduino 發送訊息給馬達以及晶片做微調的指令，當眼睛所看見機器手臂停止動作並且到等待狀態時確認已經微調結束。機器手臂有其他八種操作功能。

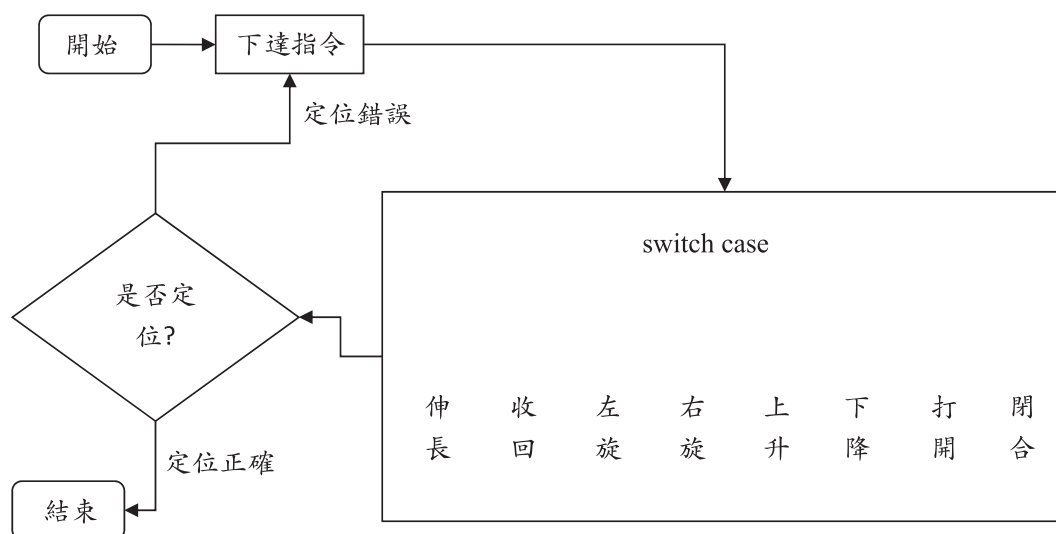


圖 11 機器手臂操作示意流程圖

四、機器手臂操作流程圖說明

依圖 12 所示，從機器手臂端開始，先做手臂的校正，再遙控端給予指令，在手臂端接收到指令之後動作到指令位置，到達目標動作後，啟動馬達手臂歸零校正，再以寸動方式收回。

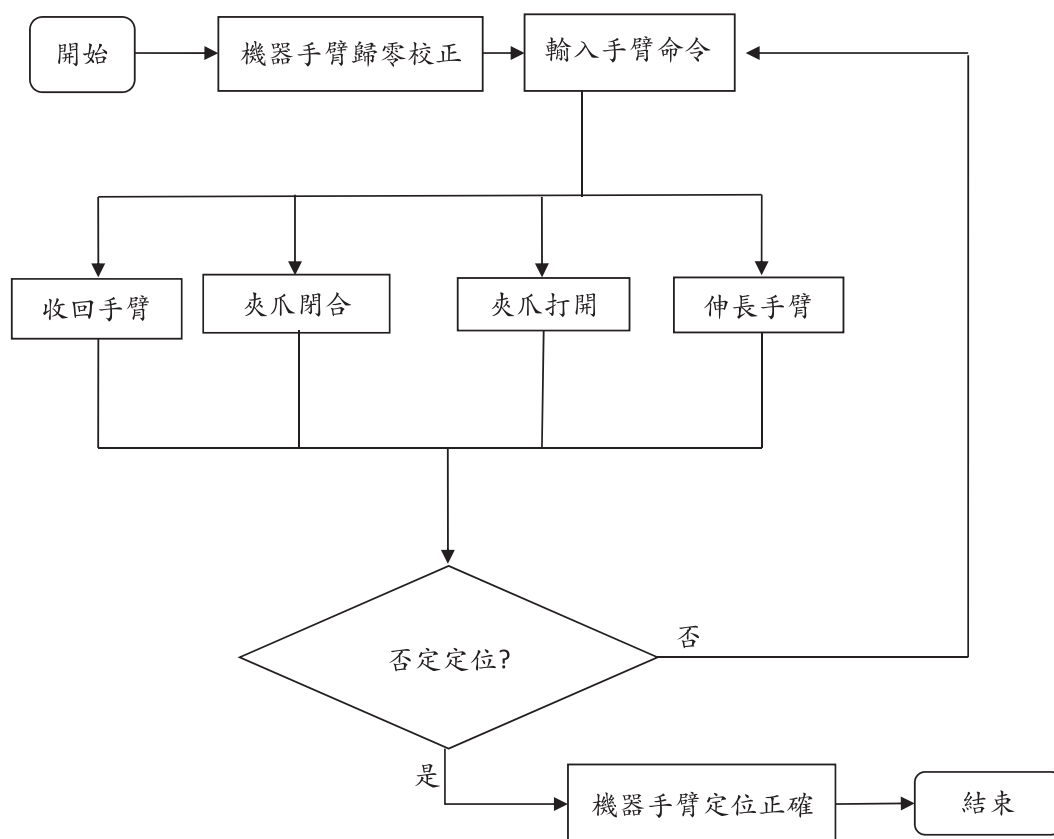


圖 12 機器手臂操作流程圖

伍、成果說明

一、硬體成果及說明

1、自走車介紹

遙控車只有三個輪子，其中兩個後輪分別由兩個馬達來控制，至於第三個輪子則只是用來平衡用，並沒有動力。圖 13 為自走車成品。

圖 14 夾取的是一個小型鐵夾，夾取的目的是確認機夾爪是否會讓鐵夾掉落。

圖 15 夾取的是一個小型金屬開罐器，夾取的目的是確認夾爪是否會讓小型金屬開罐器掉落。

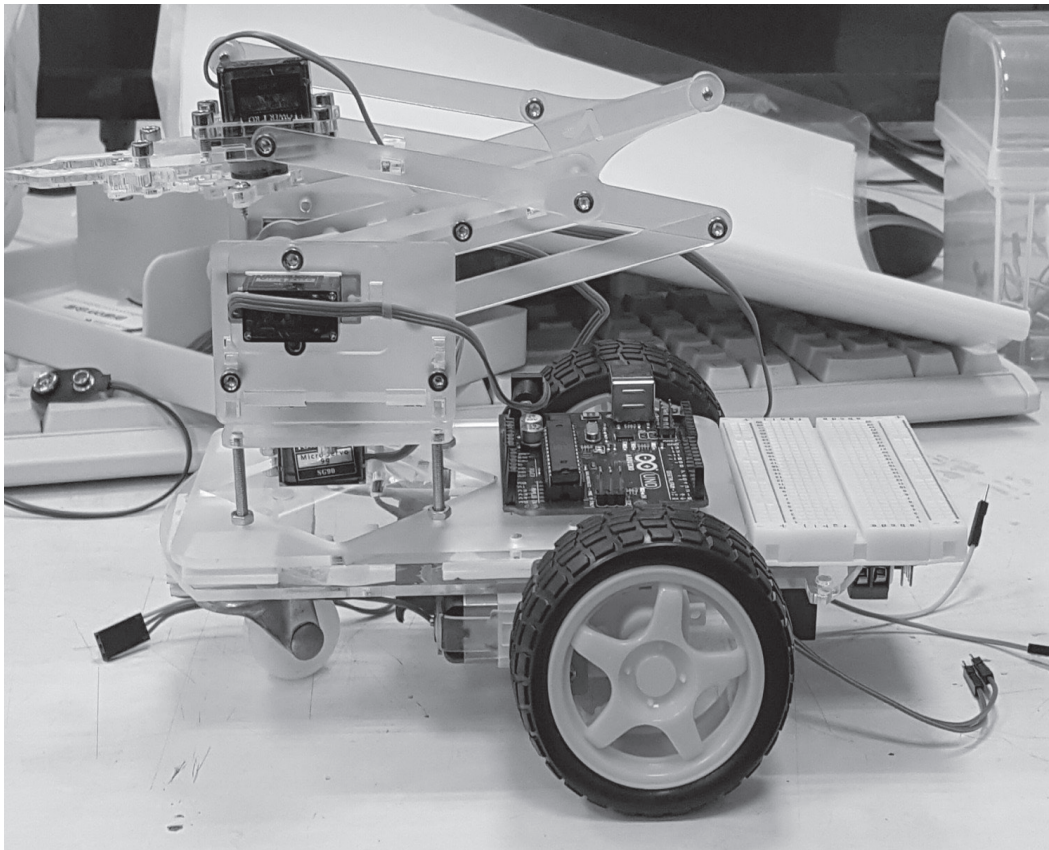


圖 13 自走車成品

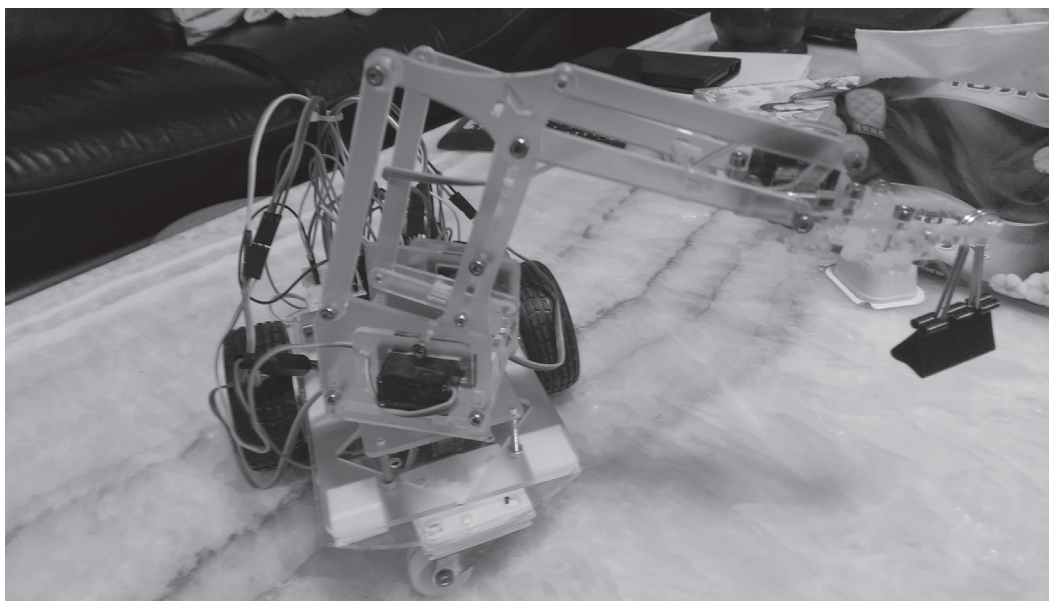


圖 14 夾取小型鐵夾力道示範圖

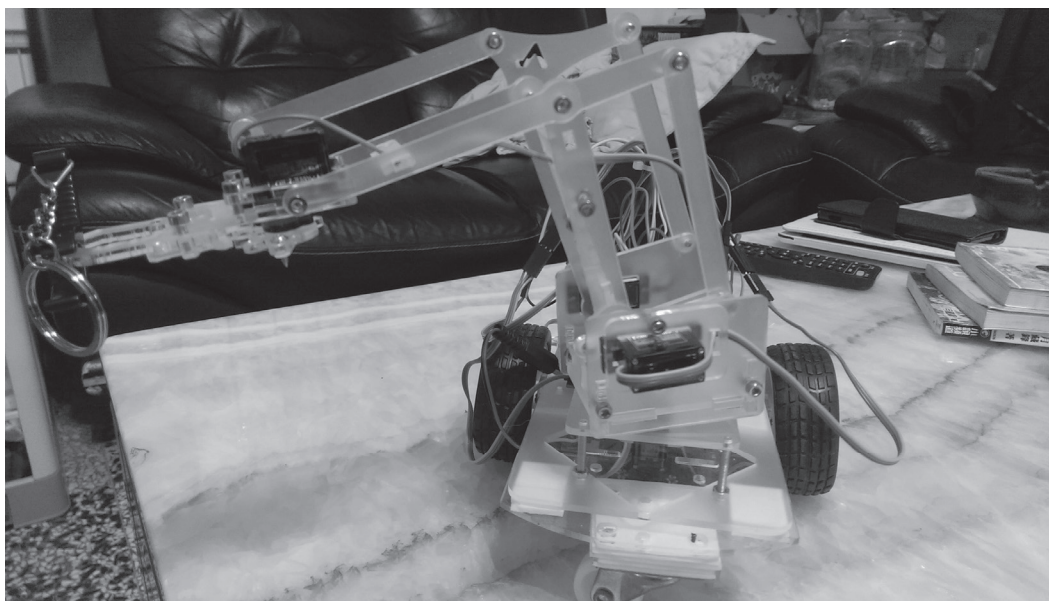


圖 15 夾取小型金屬開罐器力道示範圖

本研究有提供夾取物品成果圖以及夾取的操作影片，可以為手機遙控取物自走車提供最直接的說明與介紹。手機遙控取物自走車操作影片：

實際夾取電池盒的影片網址：<https://www.youtube.com/watch?v=hHN7vWqhs1M>

操作影片介紹利用自走車與機器手臂的混和應用達成將電池盒從定點運送給指定人員，可以看到車輛在寸動得反應速度是很靈敏的，所以在操作時需要小心操作才能達成指定任務。

二、軟體成果

所有的程式碼分成 Arduino 啟動碼、藍芽連線程式碼、機械手臂校正程式碼、自走車程式碼和機械手臂程式碼這幾個區域，這種分類有助於查錯與區分功能種類以及提升辨識性。

1、arduino 啟動（一部分）程式碼

```
#include<SoftwareSerial.h>    // 使用序列埠系統庫
#include<Servo.h>              // 使用伺服馬達系統庫
Servo servoa; // 設定伺服馬達編號 a
Servo servob; // 設定伺服馬達編號 b
Servo servoc; // 設定伺服馬達編號 c
Servo servod; // 設定伺服馬達編號 d
SoftwareSerial mySeria(0, 1); // 設定 tx,rx 使用腳位
int pos1 = 90 ;// 設定 pos1 初始值 90
int pos2 = 90 ; // 設定 pos2 初始值 90
int pos3 = 90 ; // 設定 pos3 初始值 90
int pos4 = 25 ; // 設定 pos4 初始值 25
void setup() { // 初始設定區域
```

2、藍芽連線部分程式碼

```
Serial.begin(9600); // 設定序列埠連線速率為 9600
mySeria.begin(9600); // 設定藍芽連線速率為 9600
Serial.println("Please any Key To LCD ...");// 在序列埠監控視窗發送 Please
any Key To LCD ...
```

3、機械手臂校正（一部分）程式碼

```
servoa.attach(8); // 設定腳位 8 伺服馬達輸出  
servob.attach(9); // 設定腳位 9 伺服馬達輸出  
servoc.attach(10); // 設定腳位 10 伺服馬達輸出  
servod.attach(11); // 設定腳位 11 伺服馬達輸出
```

4、自走車（一部分）程式碼說明

```
case 'f':// 方案 f  
{  
  digitalWrite(3, HIGH); // 給第 3 腳位高電壓  
  digitalWrite(4, LOW); // 給第 4 腳位低電壓  
  digitalWrite(5, HIGH); // 給第 5 腳位高電壓  
  digitalWrite(6, LOW); // 給第 6 腳位低電壓  
  Serial.println("motor is mode 1."); // 在序列埠監控視窗發送 motor is mode 1  
  Serial.println("follow"); // 在序列埠監控視窗發送 follow  
  delay(1000); // 延遲 1 秒  
  digitalWrite(3, LOW); // 給第 3 腳位低電壓  
  digitalWrite(4, LOW); // 給第 4 腳位低電壓  
  digitalWrite(5, LOW); // 給第 5 腳位低電壓  
  digitalWrite(6, LOW); // 給第 6 腳位低電壓  
  Serial.println("motor is mode 5."); // 在序列埠監控視窗發送 motor is mode 5  
  Serial.println("STOP"); // 在序列埠監控視窗發送 STOP  
  break; // 方案跳離  
}  
  
case 'b':// 方案 b  
{  
  digitalWrite(3, LOW); // 給第 3 腳位低電壓  
  digitalWrite(4, HIGH); // 給第 4 腳位高電壓  
  digitalWrite(5, LOW); // 給第 5 腳位低電壓  
  digitalWrite(6, HIGH); // 給第 6 腳位高電壓  
  Serial.println("motor is mode 2."); // 在序列埠監控視窗發送 motor is mode 2.  
  Serial.println("back"); // 在序列埠監控視窗發送 back
```



```
delay(1000); // 延遲 1 秒
digitalWrite(3,LOW); // 給第 3 腳位低電壓
digitalWrite(4, LOW); // 給第 4 腳位低電壓
digitalWrite(5, LOW); // 給第 5 腳位低電壓
digitalWrite(6, LOW); // 給第 6 腳位低電壓
Serial.println("motor is mode 5.");// 在序列埠監控視窗發送 motor is mode 5
Serial.println("STOP");// 在序列埠監控視窗發送 STOP
break; // 方案跳離
}
case 'l':// 方案 l
{
digitalWrite(3, LOW); // 給第 3 腳位低電壓
digitalWrite(4, HIGH); // 給第 4 腳位高電壓
digitalWrite(5, HIGH); // 給第 5 腳位高電壓
digitalWrite(6, LOW); // 給第 6 腳位低電壓
Serial.println("motor is Mode 3.");// 在序列埠監控視窗發送 motor is Mode 3.
Serial.println("left");// 在序列埠監控視窗發送 left
delay(1000);// 延遲 1 秒
digitalWrite(3,LOW); // 給第 3 腳位低電壓
digitalWrite(4, LOW); // 給第 4 腳位低電壓
digitalWrite(5, LOW); // 給第 5 腳位低電壓
digitalWrite(6, LOW); // 給第 6 腳位低電壓
Serial.println("motor is mode 5.");// 在序列埠監控視窗發送 motor is mode 5
Serial.println("STOP");// 在序列埠監控視窗發送 STOP
break; // 方案跳離
}
case 'r':// 方案 r
{
digitalWrite(3, HIGH); // 給第 3 腳位高電壓
digitalWrite(4, LOW); // 給第 4 腳位低電壓
digitalWrite(5, LOW); // 給第 5 腳位低電壓
digitalWrite(6, HIGH); // 給第 6 腳位高電壓
```

```
Serial.println("motor is Mode 4.");// 在序列埠監控視窗發送 motor is mode 4.  
Serial.println("right");// 在序列埠監控視窗發送 right  
delay(1000); // 延遲 1 秒  
digitalWrite(3,LOW); // 給第 3 腳位低電壓  
digitalWrite(4, LOW); // 給第 4 腳位低電壓  
digitalWrite(5, LOW); // 給第 5 腳位低電壓  
digitalWrite(6, LOW); // 給第 6 腳位低電壓  
Serial.println("motor is mode 5.");// 在序列埠監控視窗發送 motor is mode 5  
Serial.println("STOP");// 在序列埠監控視窗發送 STOP  
break; // 方案跳離  
}
```

5、機器手臂部分程式碼說明

```
case '1':// 方案 1  
if (pos1 >= 135) { // 如果 pos1 值大於或等於 135  
pos1 = 135; // pos1 值等於 135  
servoa.write(pos1); // 伺服馬達 a 角度寫入 pos1 值  
Serial.println("servoa is mode1 "); // 在序列埠監控視窗發送 servoa is mode1  
Serial.println("135"); // 在序列埠監控視窗發送 135  
break; // 方案跳離  
}  
else { // 否則  
pos1 = pos1 + 9; // pos1 值加 9  
servoa.write(pos1); // 伺服馬達 a 角度寫入 pos1 值  
Serial.println("servoa is mode1 "); // 在序列埠監控視窗發送 servoa is mode1  
Serial.println("+9"); // 在序列埠監控視窗發送 +9  
break; // 方案跳離  
}  
case '2':// 方案 2  
if (pos1 <= 45) { // 如果 pos1 值小於或等於 45  
pos1 = 45; // pos1 值等於 45  
servoa.write(pos1); // 伺服馬達 a 角度寫入 pos1 值
```



```
Serial.println("servoa is mode2");// 在序列埠監控視窗發送
Serial.println("45");// 在序列埠監控視窗發送
break; // 方案跳離
}
else { // 否則
pos1 = pos1 - 9; // pos1 值減 9
servoa.write(pos1); // 伺服馬達 a 角度寫入 pos1 值
Serial.println("servoa is mode2");// 在序列埠監控視窗發送 servoa is mode2
Serial.println("-9");// 在序列埠監控視窗發送 -9
break; // 方案跳離
}
```

陸、結論與未來研究方向

本文所研製的手機遙控取物自走車，主要以手機藍芽裝置搭配 Arduino 為控制核心進行遙控，功能上使用寸動方式有效去操作夾取以及移動，可以更穩的微調操控。

本研究自行設計的 App Inventor 及 Arduino 程式，整體的整合執行測試，能依手機命令訊息讓機器人執行動作，能達到手機遙控自走車到目標及取物之功能，整體測試結果令人滿意。

本研究自行撰寫的 App 程式，由於選擇小型機體做為開發標準，所以夾取力道以及載重性皆需要再調適，整體上仍有很大的開發空間與擴展元件。

未來升級方向，可朝功能導向加裝紅外線變成尋跡車或加裝攝影鏡頭提升操作性，另外一種升級是選擇開發中大型機體自走車與機器手臂的性能，同時也可以增加擴展元件的數量與質量，期望達到更實務的應用目標。

參考文獻

- 1、楊明豐，ARDUINO 最佳入門與應用，基峯出版社，2016 年。
- 2、趙英傑，超圖解 Arduino 互動設計入門，旗標出版社，2016 年。
- 3、App Inventor wiki
<https://zh.wikipedia.org/wiki/MIT%E5%BA%94%E7%94%A8%E5%BC%80%E5%8F%91%E8%80%85>
- 4、HC-05 與 HC-06 藍牙模組補充說明（一）。超圖解 Arduino 互動設計入門。
取自 <http://swf.com.tw/?p=693>
- 5、HC-05 與 HC-06 藍牙模組補充說明（二）：連線電路。超圖解 Arduino 互動設計入門。
取自 <http://swf.com.tw/?p=705>
- 6、HC-05 與 HC-06 藍牙模組補充說明（三）：使用 Arduino 設定 AT 命令。
超圖解 Arduino 互動設計入門。取自 <http://swf.com.tw/?p=712>
- 7、利用 Arduino 控制 2 顆伺服馬達
取自 <http://diveduino.blogspot.tw/2012/05/page-3-arduino-2-control-2-servo-motors.html>
- 8、小型伺服馬達 SG90 控制，取自 <http://frank1025.pixnet.net/blog/post/130577051>
- 9、Arduino UNO 控制板簡介，取自 <http://swf.com.tw/?p=569>