

國立聯合大學 電子工程學系
111學年度專題研究報告

水上智慧清道夫



指導教授：陳元炘
組 員：朱嫵臻(U0822204)
崔芸瑄(U0822304)
施俊億(U0822325)

中華民國 111年 12月

國立聯合大學 電子工程學系
111學年度專題研究報告審定

水上智慧清道夫

指導老師：_____ (簽名)

組 員：_____ (簽名)

組 員：_____ (簽名)

組 員：_____ (簽名)

中華民國 111年 12月

目錄

第一章 前言	5
1.1 研究背景與動機	5
1.2 目的	5
第二章 軟體及工具介紹	6
2.1 ARDUINO [1][2]	6
2.2 MIT APP INVENTOR[3]	7
2.3 L293D IC [4]	8
2.4 雙軸按鍵搖杆模組[5]	9
2.5 藍牙模組 HC-05[6]	9
第三章 專題架構	10
3.1 專題架構	10
3.2 電路圖[7]	11
3.3 ARDUINO 程式碼	12
3.4 MIT APP INVERTOR	24
第四章 成果展示	31
4.1 整體外觀	35
4.2 內部細節	35
4.3 實體操作	35
第五章 總結	35
5.1 結論	35
5.2 未來展望	35
參考文獻	36

圖目錄

圖1 ARDUINO UNO.....	6
圖2 MIT APP INVENTOR LOGO.....	7
圖3 MIT APP INVENTOR LOGO.....	8
圖4 裝軸按鍵搖杆模組	9
圖5 藍牙模組.....	9
圖6 機械手臂之電路圖	11
圖7 船體之電路圖	11
圖8 船體 APP 二維條碼	24
圖9機械手臂 APP 二維條碼	24
圖10 APP 船體介面.....	25
圖11 APP 機械手臂介面.....	25
圖12 APP 船體程式.....	26
圖13 APP 船體程式.....	26
圖14 APP 船體程式.....	27
圖15 APP 船體程式.....	27
圖16 APP 機械手臂程式.....	29
圖17 APP 機械手臂程式.....	29
圖18 APP 機械手臂程式.....	30
圖19 APP 機械手臂程式.....	30
圖20 APP 機械手臂程式.....	31
圖21整體外觀.....	33
圖22 線路正面.....	34
圖23線路背面.....	34
圖24船體.....	33
圖25 將夾取垃圾	34

圖26夾取垃圾.....	34
--------------	----

第一章 前言

1.1 研究背景與動機

台灣是個四面環海的國家，海洋對我們而言是無比重要，與我們生活息息相關，但也因此常看到人造垃圾在水上漂浮，然而海洋垃圾對海中生物造成威脅，現今在報章雜誌或新聞中看到海洋生物因垃圾所傷害，如：海龜鼻中插入吸管、魚類因食用塑膠袋死亡，更關係到人民食用的危害，更造成漁業損失。海洋垃圾隨著漂流，唯有針對垃圾減少改善，才能還給水生生物一個家。

1.2 目的

目前解決海洋垃圾的方法，常見的有機關團體或學校以淨灘方式，清理沙灘上的垃圾，或政府單位出動打撈垃圾，以人力方式打撈上岸，雖然能減少垃圾，卻耗費大量的時間，此計畫利用機械式手臂打撈大量垃圾，以便將來更有效率的在海上撈取垃圾，因此我們自行製作簡易型機械手臂及船體加上手機 app 更加方便操作。

第二章 軟體及工具介紹

2.1 Arduino [1][2]

Arduino 是一個開源嵌入式硬體平台，用來供使用者製作可互動式的嵌入式專案。可簡單地與感測器，各式各樣的電子元件連接，目的是為新手和專業人員提供一種低成本且簡單的方法，以建立使用感測器與環境相互作用的裝置執行器，在 Arduino 上執行的程式可以使用任何能夠被編譯成 Arduino 機器碼的程式語言編寫，Arduino 官方推薦使用整合了 Arduino 的 C++ 進行編程。

Arduino Uno 是一款基於 ATmega328P 的微控制器板。它有 14 個數位輸入輸出接腳（其中 6 個可用作 PWM 輸出），6 個類比輸入，16 MHz 石英晶體，USB 連接孔，電源插孔，ICSP 接頭和重置按鈕，如圖1。



圖1 Arduino UNO

2.2 MIT App Inventor[3]

網站提供雲端服務，它使用圖形化界面，非常類似於 Scratch 語言和 StarLogo TNG 用戶界面。用戶可以拖放圖形對象來創造一個運行在安卓系統上的應用，讓我們可以免費使用其提供的強大功能，來設計 Android 裝置上的 App 應用程式，使用者開發完成的程式，可以下載到任何使用 Android 作業系統的裝置，包括：智慧型手機或平板電腦。如圖2。



圖2 MIT App Inventor logo

2.3 L293D IC [4]

L293D IC 是16 針腳電機驅動器 IC，可以同時控制兩個直流馬達前進和後退。L293D 提供高達 600 mA 的雙向驅動電流，電壓範圍為 4.5 V 至 36 V。

接腳1和9是控制馬達轉速，接腳3、6、11和14連接馬達，接腳2、7、10和15連接 arduino，去控制馬達轉的方向，接腳8和16連接電池正端，接腳4、5、12和13連接電池負端如圖3。

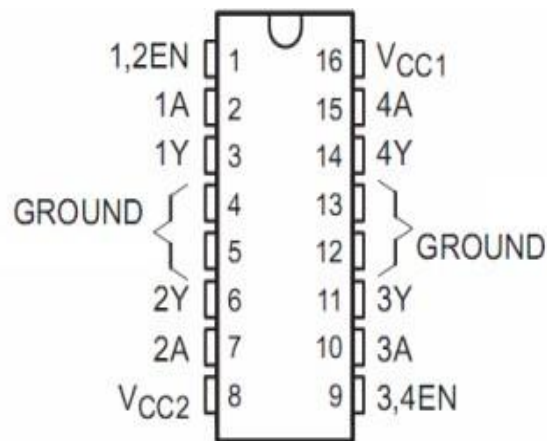


圖3 MIT App Inventor logo

2.4 雙軸按鍵搖杆模組[5]

雙軸按鍵搖桿模組，從左至右的接腳順序與標示分別是 GND、+5V、VRx、VRy、SW，搖桿可沿著 X 軸和 Y 軸兩個方向移動 (VRx 和 VRy)。因其內部十字搖桿為一個雙向的10K 電阻器，隨著搖桿方向不同，抽頭的阻值隨著變化，移動時，對外輸出 0到1023 的訊號。搖桿也提供按鍵開關 (SW)，對外輸出 0 或 1 的訊號，如圖 3。

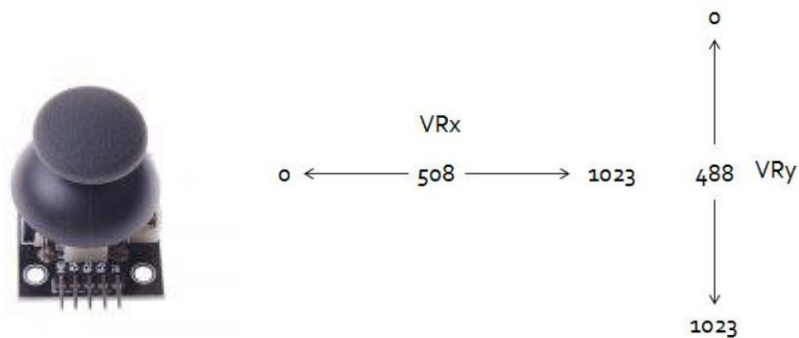


圖4 裝軸按鍵搖杆模組

2.5 藍牙模組 HC-05[6]

藍牙模組可以讓你原來使用串口的設備擺脫線纜的束縛在10米範圍內實現無線串口通信。介面包括 VCC,GND,TXD,RXD,KEY 引腳、藍牙連接狀態引出腳 (STATE)，led 指示藍牙連接狀態，快閃表示沒有藍牙連接，慢閃表示進入 AT 模式，雙閃表示藍牙已連接並打開了埠，如圖5。

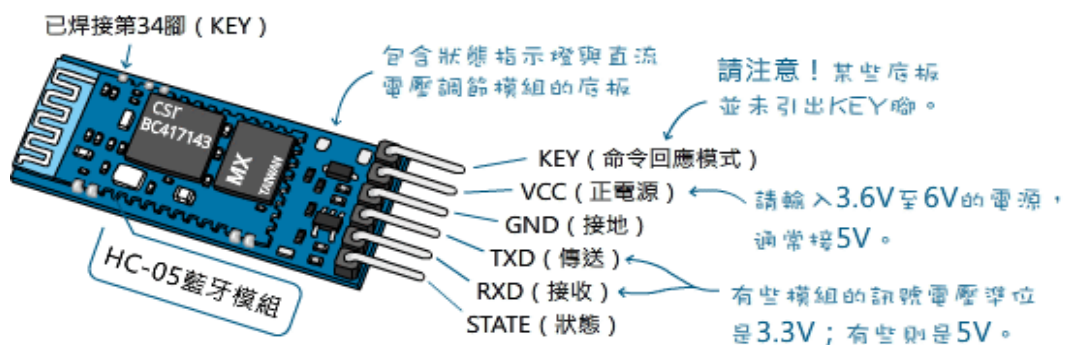
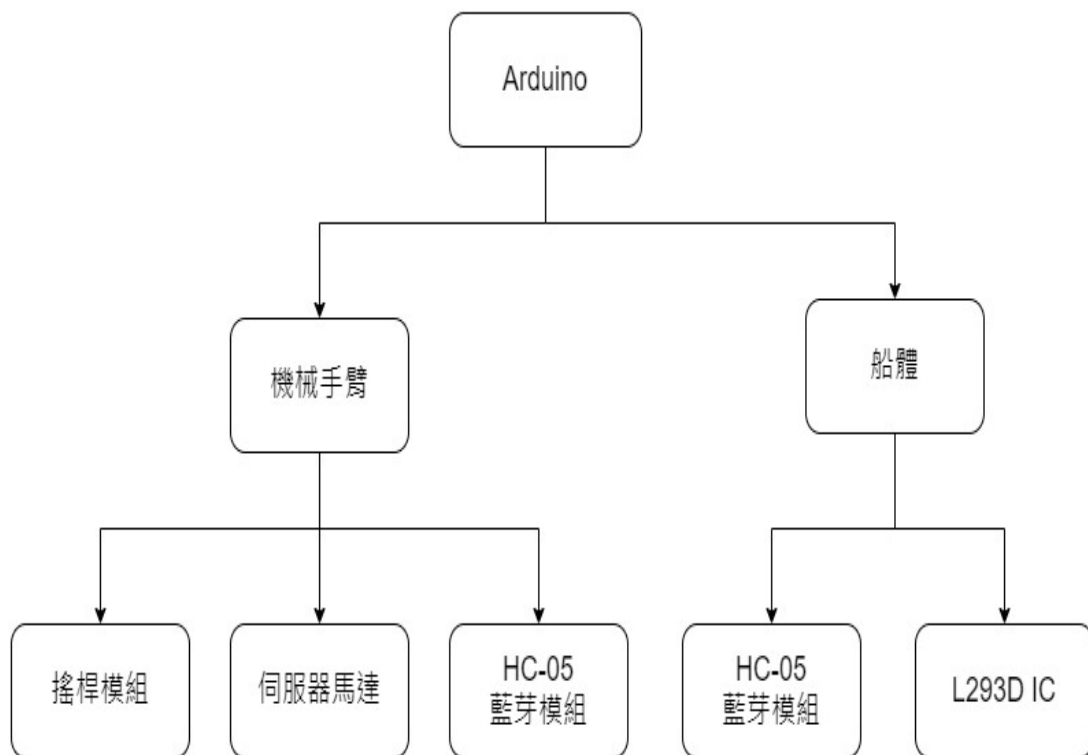


圖5 藍牙模組

第三章 專題架構

3.1 專題架構

機械手臂及船體都是利用 Arduino 撰寫，機械手臂利用搖桿模組、伺服器馬達、及藍牙模組所組成，搖桿結合伺服器馬達控制前後左右上下，以準確夾取垃圾，船體則是用 L293D IC 及藍牙所運作，L293D IC 為了控制船體速度快慢，為了操作方便，用 MIT APP INVERTOR 製作手機 app，與藍牙連接，只需在船體裡用手機控制船體方向、行駛速度以及機械手臂控制方向夾取，最終呈現結果，如表一。



表一 專題架構

3.2 電路圖[7]

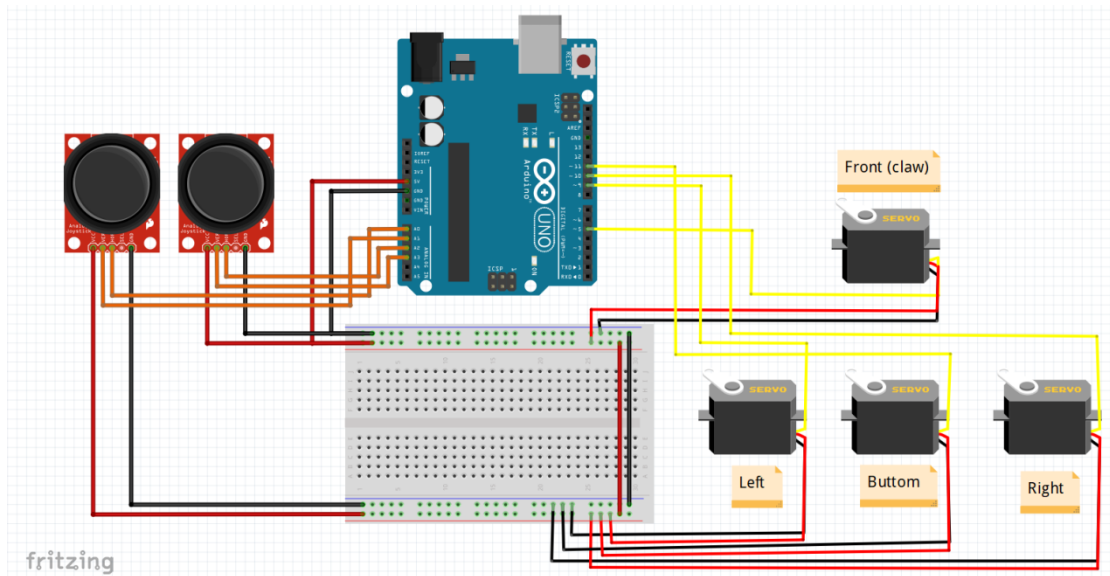


圖6 機械手臂之電路圖

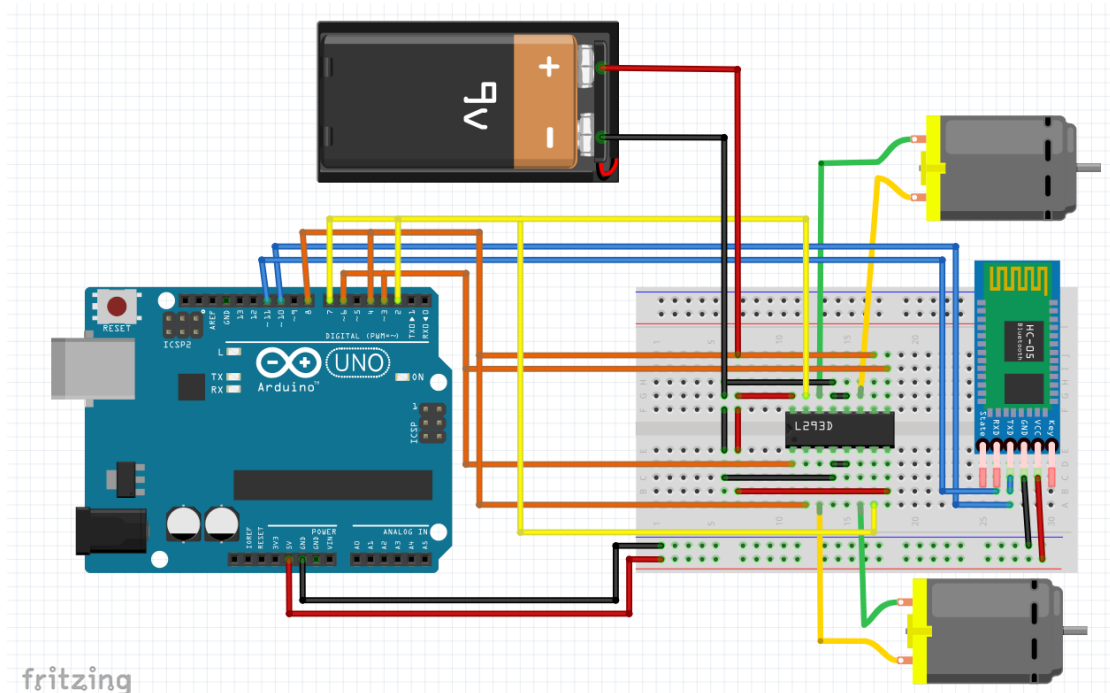


圖7 船體之電路圖

3.3 Arduino 程式碼

一、雙軸按鍵搖及伺服器馬達讓機械手臂抓取東西

```
#include <Servo.h>

const int SERVOS_TOTAL = 4;          // 總共有4顆伺服馬達

const int joystickOffsetMin = 100; // 搖桿至少扳動100才開始動作

const int joystickIdleMax = 100; // 搖桿未使用狀態計數器達100次後,馬達斷線

const int delayMilliseconds = 20; // 每隔20 MS 讀取一次資料


Servo myservo[SERVOS_TOTAL];        // 宣告 Servo 物件陣件

int idleCounter[SERVOS_TOTAL];      // 宣告記錄搖桿某方位多久沒動作的陣列


// 伺服馬達的設定, 一行設定一顆

// 參數依序為 : joystickPin, servoPin, min, max, initialAngle

// joystickPin : 搖桿輸入給 Arduino 哪個腳位(用 0~3 代表 A0~A3)

// servoPin : 伺服馬達(橙/黃線)接在 Arduino 哪個腳位

// min : 伺服馬達最小的角度是多少

// max : 伺服馬達最大的角度是多少

// initialAngle : 伺服馬達初始的角度是多少
```

```

int servoConfig[SERVOS_TOTAL][5] = {

    // joystickPin,servoPin,min,max,initialAngle

    { 0, 11, 0, 179, 90 }, //A0 = L.AX , Buttom

    { 1, 9, 179, 0, 150 }, //A1 = L.AY , Left

    { 2, 10, 179, 0, 90 }, //A2 = R.AY , Right

    { 3, 5, 179, 100, 110 }, //A3 = R.AX , Front (claw)

};

// 初始化的程序

void setup()

{

    int outputPin, angleMin, angleMax, angleInit;

    Serial.begin(9600);

    for(int i=0; i<SERVOS_TOTAL; i++) {

        outputPin = servoConfig[i][1]; //由設定取得馬達在 Arduino 的接腳參數

        angleInit = servoConfig[i][4]; // 由設定取得馬達初始的角度參數

        myservo[i].attach(outputPin);

        myservo[i].write(angleInit);    // 設定馬達的角度

        idleCounter[i] = 0;             // 將搖桿未使用記數器歸零

    }

}

// 重覆執行的程序

```

```

void loop()

{

    int inputValue, currentAngle, joyStickOffset, angleOffset;

    int inputPin, outputPin, angleMin, angleMax, temp;

    for(int i=0; i<SERVOS_TOTAL; i++) {

        inputPin  = servoConfig[i][0]; // 由設定取得搖桿輸入給 Arduino 的接腳參數

        outputPin = servoConfig[i][1]; // 由設定取得馬達在 Arduino 的接腳參數

        angleMin  = servoConfig[i][2]; // 由設定取得馬達最小角度的參數

        angleMax  = servoConfig[i][3]; // 由設定取得馬達最大角度的參數


        inputValue = analogRead(inputPin); // 讀取搖桿的數值(0~1023)

        currentAngle = myservo[i].read(); // 取得馬達目前的角度


        joyStickOffset = inputValue - 512;

        // 計算搖桿和未使用時(中間值 512)的差動多少, 用以判斷是否要改變馬達的角度


        // 如果搖桿扳動的幅度大於設定的最小值就改變馬達的角度

        if(abs(joyStickOffset) > joystickOffsetMin) {

            // 如果最小角度 > 最大角度 ==> 搖桿動作相反

            if(angleMin > angleMax) {

                joyStickOffset *= -1;          // 將搖桿動作設為相反 1 -> -1 , -1 -> 1

            }

            // 將最小角度和最大角度的值交換

```



```
temp = angleMin;

angleMin = angleMax;

angleMax = temp;

}
```

// 如果搖桿往比 512 大的方向扳動就讓馬達角度減少, 反之則增加

```
if(joyStickOffset > 0) {

    angleOffset = -1;

} else {

    angleOffset = 1;

}
```

```
currentAngle += angleOffset; // 計算新的馬達角度
```

// 如果新的馬達角度在最小角度參數和最大角度參數之間, 就進行馬達改變角度的程序

```
if(currentAngle > angleMin && currentAngle < angleMax) {

    idleCounter[i] = 0;
```

// 如果馬達未連線, 先行連線

```
if(!myservo[i].attached()) {

    myservo[i].attach(outputPin);

}
```

// 如果馬達已連線, 將角度設為新的角度

```

    if(myservo[i].attached()) {

        // sets the servo position according to the scaled value

        myservo[i].write(currentAngle);

    }

}

} else {

    // 搖桿未扳動或未達搖桿動作的標準, 將計數器加一

    idleCounter[i]++;

}

// 如果搖桿未動的計數超過設定的最大值後, 切斷馬達的連線

if(idleCounter[i] > joystickIdleMax) {

    idleCounter[i] = 0;    // 計數器歸零

    myservo[i].detach(); // 切斷馬達的連線

}

}

delay(delayMilliseconds);    // 暫停多少 MS 後繼續

}

```

二、手機接收藍牙來改變船體方向

```
#include <SoftwareSerial.h>

#include <Wire.h>

//Motor L

const int motorPin1 = 2; // Pin 15 of L293

const int motorPin2 = 4; // Pin 10 of L293

const int enablePin1 = 3; // Pin 1 of L293


//Motor R

const int motorPin3 = 7; // Pin 7 of L293

const int motorPin4 = 8; // Pin 2 of L293

const int enablePin2 = 6; // Pin 9 of L293


//定義 Arduino PIN10及 PIN11分別為 TX 及 RX 腳

SoftwareSerial BT(10,11);


//讀取手機傳輸數值的變數

char Car_status[2];


//定義轉速的變數

int runspeed;
```

```
void setup(){

    pinMode(motorPin1, OUTPUT);

    pinMode(motorPin2, OUTPUT);

    pinMode(motorPin3, OUTPUT);

    pinMode(motorPin4, OUTPUT);

    pinMode(enablePin1, OUTPUT);

    pinMode(enablePin2, OUTPUT);


    //設定初始值

    digitalWrite(motorPin1, LOW);

    digitalWrite(motorPin2, LOW);

    digitalWrite(motorPin3, LOW);

    digitalWrite(motorPin4, LOW);

    analogWrite(enablePin1, 0);

    analogWrite(enablePin2, 0);


    // 設定和 HC-05 通訊的速度 (預設 9600)

    BT.begin(9600);

    Serial.begin(9600);

}

void loop(){
```

//讀取藍牙模組訊息，並傳送到電腦

```
if (BT.available()){
```

```
    Serial.println("Car_status : ");
```

//透過 for 迴圈收集手機傳輸的瞭筆資料

```
for(int j = 0; j <= 1; j++){
```

```
    Car_status[j] = char(BT.read());
```

```
    Serial.println(Car_status[j]);
```

```
}
```

//判斷輸入字母去選擇馬達轉速

```
switch(int(Car_status[1])){
```

```
    case '0':        //0
```

```
        runspeed = 0;
```

```
        Serial.println("runspeed : ");
```

```
        Serial.println(runspeed);
```

```
    break;
```

```
    case 'A':        //A
```

```
        runspeed = 30;
```

```
        Serial.println("runspeed : ");
```

```
        Serial.println(runspeed);
```

```
    break;
```

```
        case 'B':          //B

        runspeed = 60;

        Serial.println("runspeed : ");

        Serial.println(runspeed);

        break;

        case 'C':          //C

        runspeed = 90;

        Serial.println("runspeed : ");

        Serial.println(runspeed);

        break;

        case 'D':          //D

        runspeed = 120;

        Serial.println("runspeed : ");

        Serial.println(runspeed);

        break;

        case 'E':          //E

        runspeed = 150;

        Serial.println("runspeed : ");

        Serial.println(runspeed);

        break;

        case 'F':          //F

        runspeed = 180;

        Serial.println("runspeed : ");
```

```

        Serial.println(runspeed);

    break;

    case 'G':        //G

        runspeed = 210;

        Serial.println("runspeed : ");

        Serial.println(runspeed);

    break;

    case 'H':        //H

        runspeed = 240;

        Serial.println("runspeed : ");

        Serial.println(runspeed);

    break;

    case 'I':        //I

        runspeed = 254;

        Serial.println("runspeed : ");

        Serial.println(runspeed);

    break;

}

}

//馬達轉速

analogWrite(enablePin1, runspeed);

analogWrite(enablePin2, runspeed);

```

```

//當讀取到資料第一筆，判斷該前後左右停

switch (Car_status[0]){

case '1':          //前進

    digitalWrite(motorPin1, LOW);

    digitalWrite(motorPin2, HIGH);

    digitalWrite(motorPin3, LOW);

    digitalWrite(motorPin4, HIGH);

    break;


case '2':          //左轉

    digitalWrite(motorPin1, LOW);

    digitalWrite(motorPin2, HIGH);

    digitalWrite(motorPin3, HIGH);

    digitalWrite(motorPin4, LOW);

    break;


case '3':          //右轉

    digitalWrite(motorPin1, HIGH);

    digitalWrite(motorPin2, LOW);

    digitalWrite(motorPin3, LOW);

    digitalWrite(motorPin4, HIGH);

    break;

```



```
case '4':          //停止

    digitalWrite(motorPin1, LOW);

    digitalWrite(motorPin2, LOW);

    digitalWrite(motorPin3, LOW);

    digitalWrite(motorPin4, LOW);

    break;

case '5':          //後退

    digitalWrite(motorPin1, HIGH);

    digitalWrite(motorPin2, LOW);

    digitalWrite(motorPin3, HIGH);

    digitalWrite(motorPin4, LOW);

    break;

}

}
```

3.4 MIT APP INVERTOR

利用 MIT APP INVERTOR 透過手機藍牙連線來控制機械手臂夾取及船體方向和行駛速度。

一、 app 行動條碼 如圖8和圖9

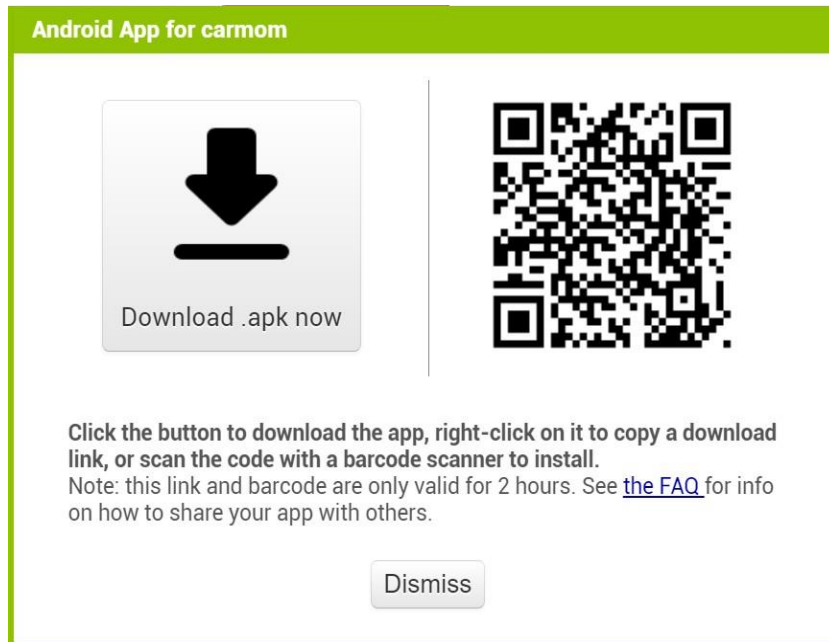


圖8 船體 app 二維條碼

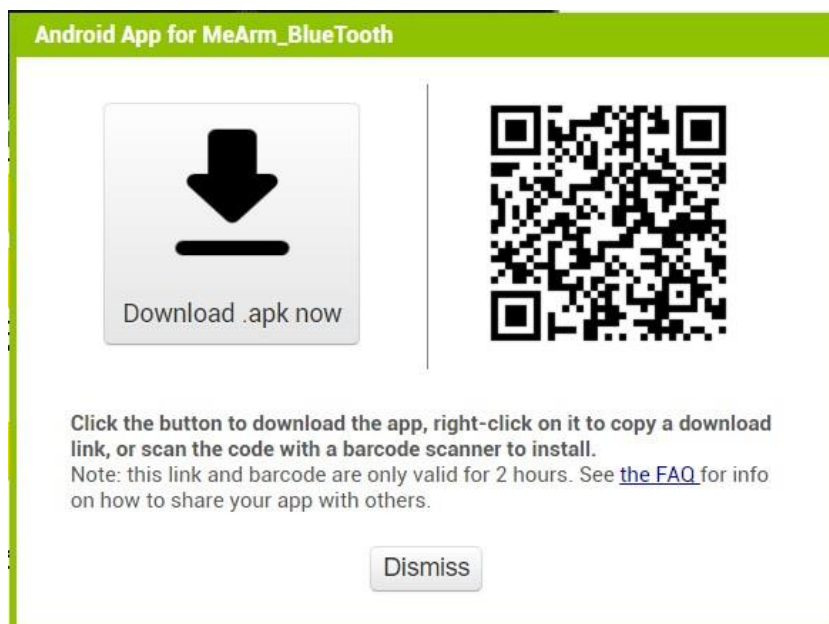


圖9機械手臂 app 二維條碼

二、 手機介面

掃描圖8和圖9的二維碼，能顯示出船體移動發向及速度的手機畫面如圖10及機械手臂夾取物品方向及夾取物件動作的手機畫面如圖11。



圖10 app 船體介面



圖11 app 機械手臂介面

三、 手機軟體程式

包括藍牙啟動連接、船體移動方向及速度選擇。

當螢幕啟動時，連接藍牙按鈕開啟，其他部分關閉以免觸碰如圖12點選中斷藍牙按鈕，呼叫藍牙中斷連接，開啟藍牙連接按鈕，其他功能關閉，顯示的數據會清空如圖13。

點選藍牙按鈕後，如果選擇連接藍牙，選好後會關閉藍牙按鈕，開啟其他部分功能如圖14。

點選 F 按鈕，會傳送給 ARDUINO“1”內容，在按鍵格子顯示 F 如圖15。

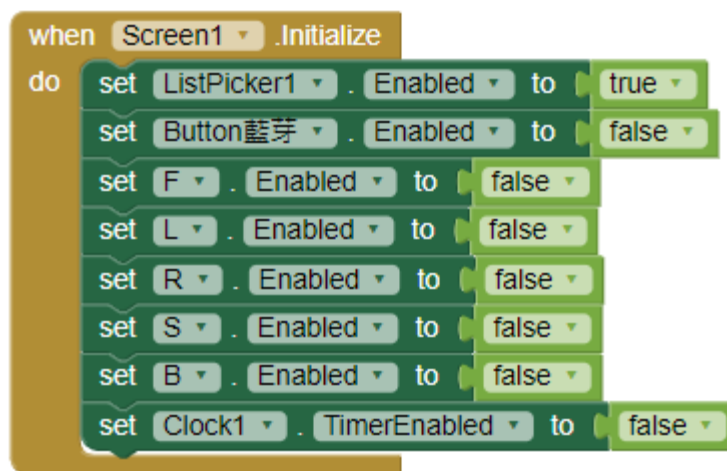


圖12 app 船體程式

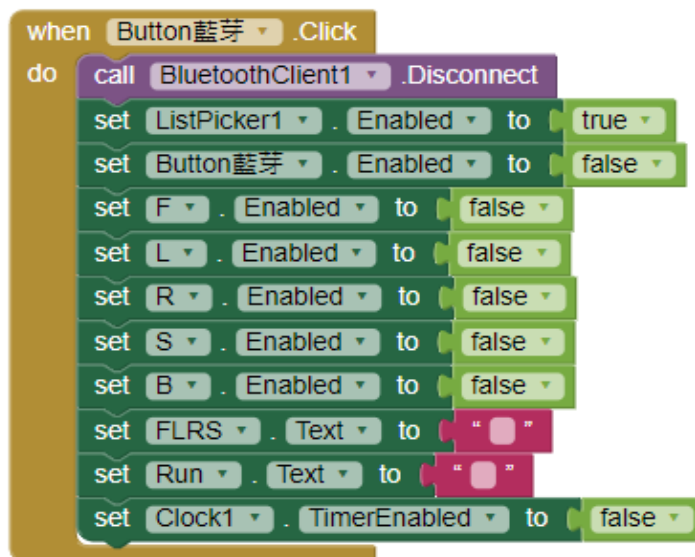


圖13 app 船體程式

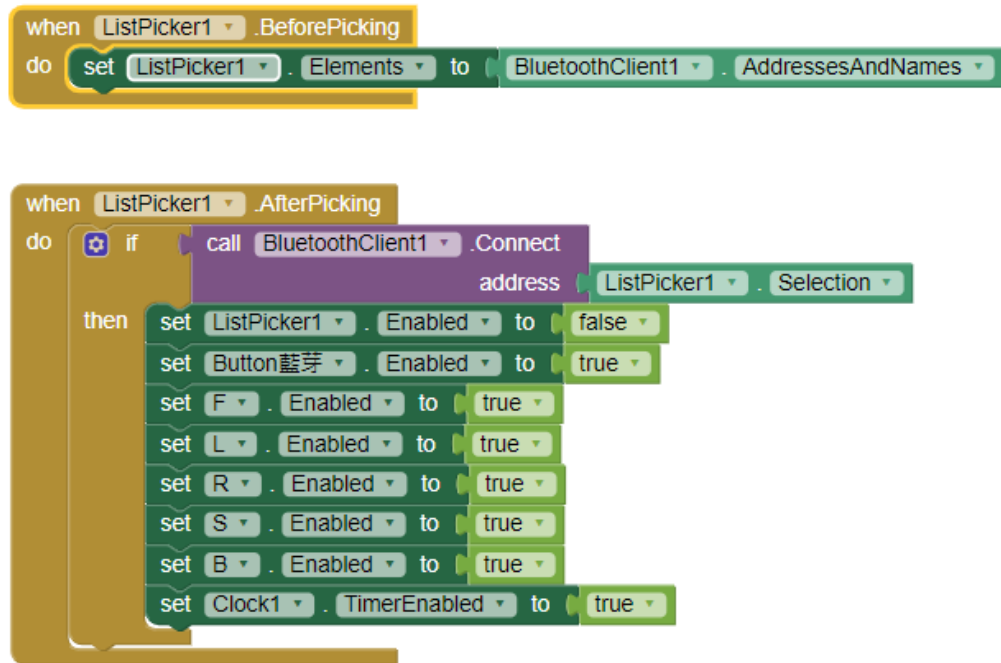


圖14 app 船體程式

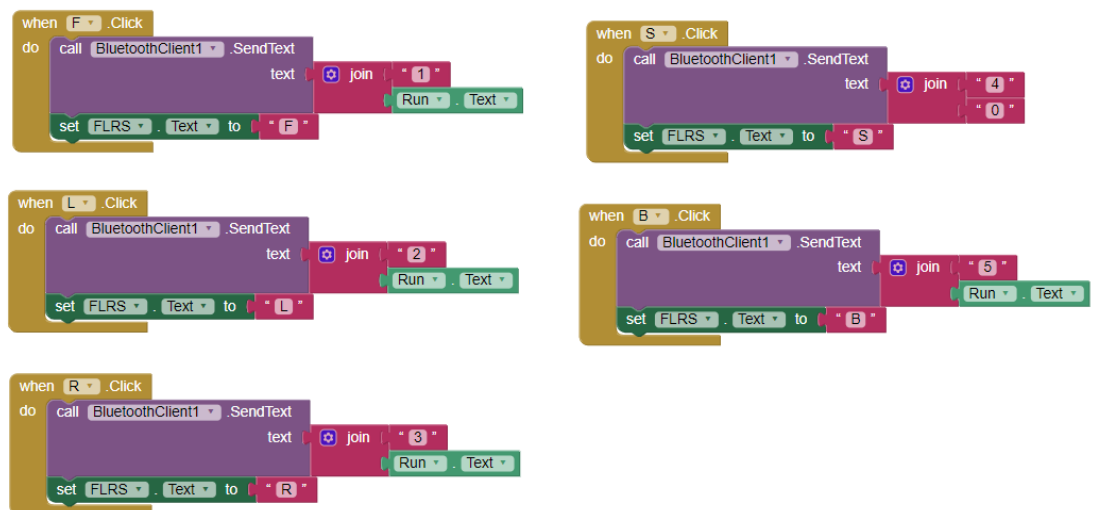


圖15 app 船體程式

藍牙啟動連接、機械手臂移動方向和控制。

當螢幕啟動時，連接藍牙按鈕開啟，其他部分關閉以免觸碰如圖20。

點選中斷藍牙按鈕，呼叫藍牙中斷連接，開啟藍牙連接按鈕，其他功能關閉，如圖20。

點選藍牙按鈕後，選擇連接的藍牙，選好後會關閉藍牙按鈕，開啟其他部分功能如圖19

上按鈕，會傳送給 ARDUINO“L”內容，控制機械手臂往上，
下是傳送給 ARDUINO“l”內容，控制機械手臂往下，
左是傳送給 ARDUINO“B”內容，控制機械手臂往左邊移動，
右是傳送給 ARDUINO“b”內容，控制機械手臂往右邊移動，
前是傳送給 ARDUINO“R”內容，控制機械手臂向前面伸展，
後是傳送給 ARDUINO“r”內容，控制機械手臂向後面收回去，
夾是傳送給 ARDUINO“C”內容，控制機械手臂夾子閉合，
開是傳送給 ARDUINO“c”內容，控制機械手臂夾子張開，如圖16。

為了讓按鈕有「連發」的功能，使用 Clock 元件做定時執行指令，八個按鈕的 TouchDown 會將要送出的指令設置給全域變數「sendcom2」，如圖16，而 TouchUp 會將「currentCmd」設置為空字串，如圖17。Clock1會檢查藍牙是否已連線及「currentCmd」是否有執行指令，如果有會每 10ms 會傳送指令給 arduino，如圖18。

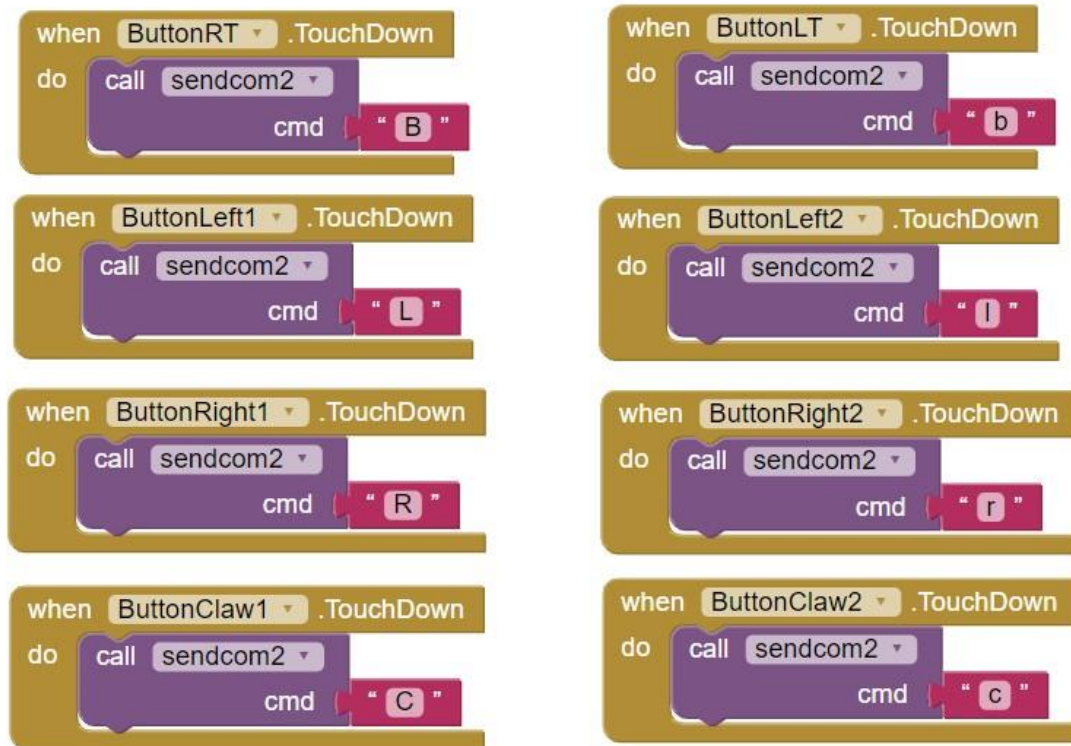


圖16 app 機械手臂程式

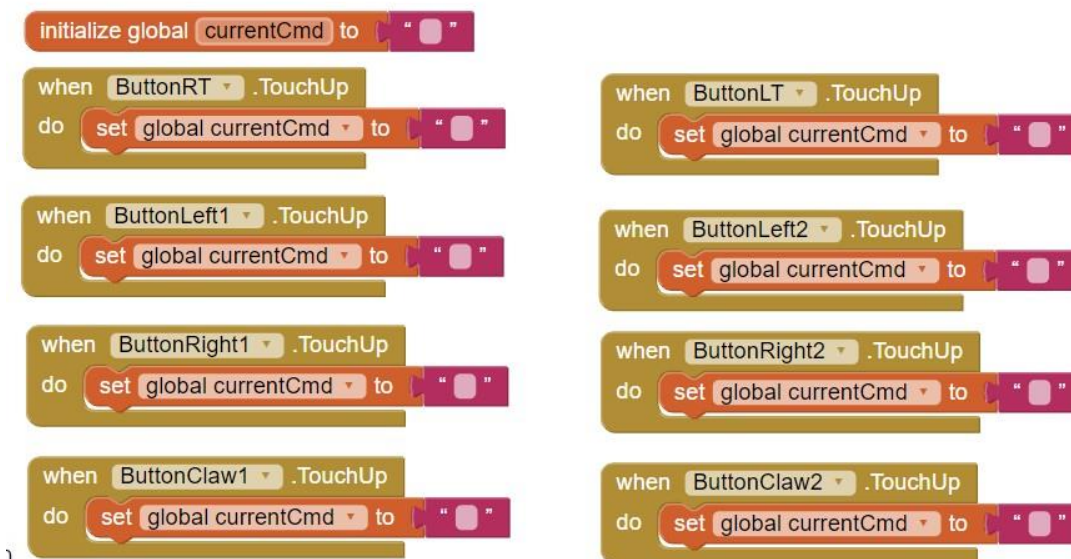


圖17 app 機械手臂程式

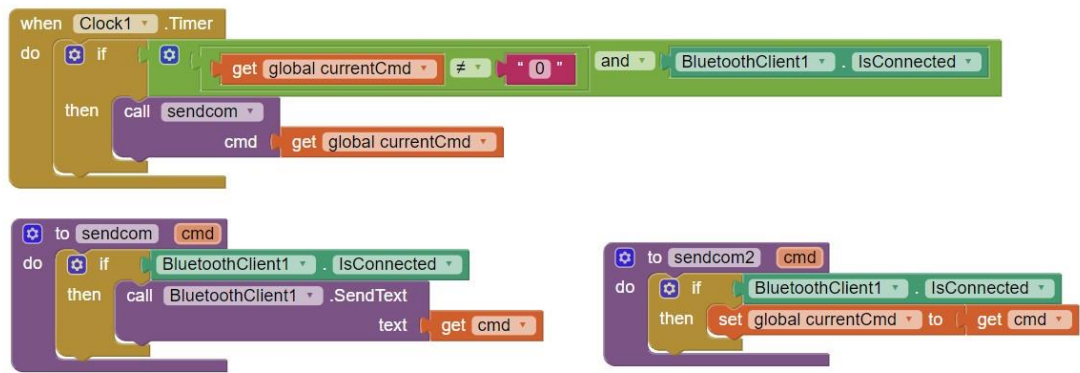


圖18 app 機械手臂程式

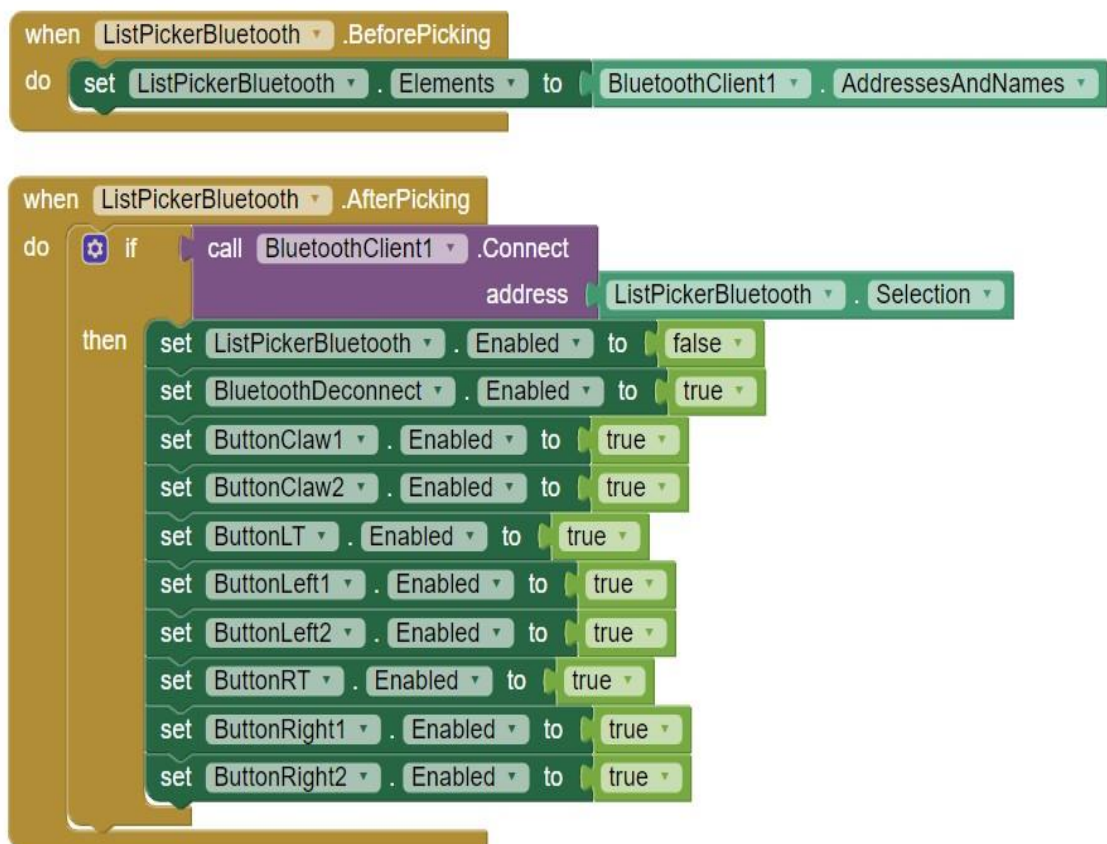


圖19 app 機械手臂程式


```

when BluetoothDeconnect .Click
do
  call BluetoothClient1 .Disconnect
  set ListPickerBluetooth . Enabled to true
  set BluetoothDeconnect . Enabled to false
  set ButtonClaw1 . Enabled to false
  set ButtonClaw2 . Enabled to false
  set ButtonLT . Enabled to false
  set ButtonLeft1 . Enabled to false
  set ButtonLeft2 . Enabled to false
  set ButtonRT . Enabled to false
  set ButtonRight1 . Enabled to false
  set ButtonRight2 . Enabled to false

```

```

when Screen1 .Initialize
do
  set ListPickerBluetooth . Enabled to true
  set BluetoothDeconnect . Enabled to false
  set ButtonClaw1 . Enabled to false
  set ButtonClaw2 . Enabled to false
  set ButtonLT . Enabled to false
  set ButtonLeft1 . Enabled to false
  set ButtonLeft2 . Enabled to false
  set ButtonRT . Enabled to false
  set ButtonRight1 . Enabled to false
  set ButtonRight2 . Enabled to false

```

圖20 app 機械手臂程式

第四章 成果展示

4.1 整體外觀如圖21。



圖21 整體外觀

4.2 內部細節

機械手臂及船體馬達線路利用洞洞板結合為圖22及圖23。

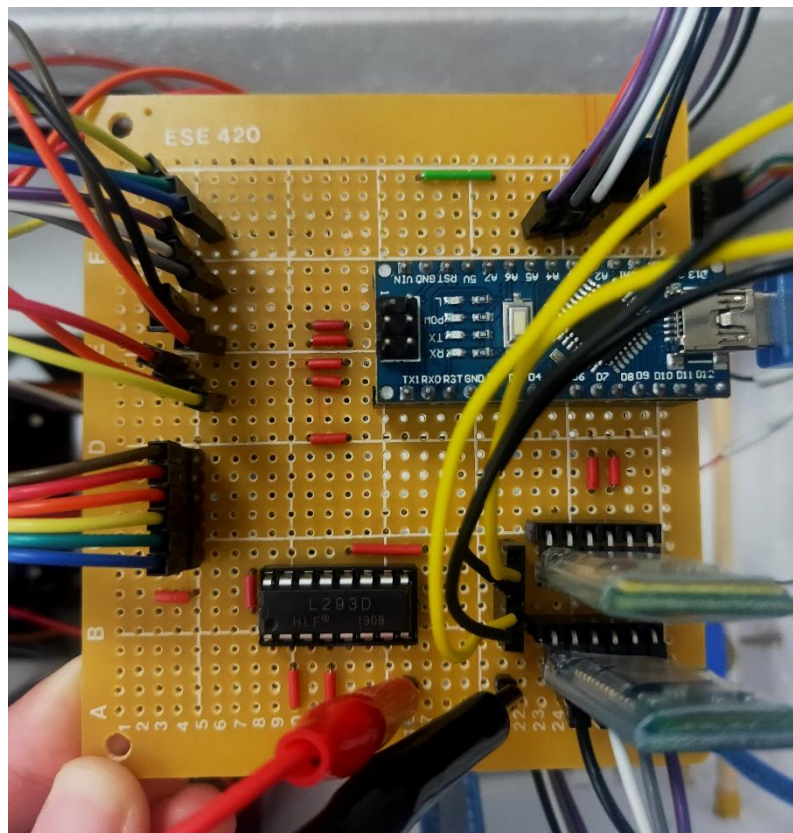


圖22 線路正面

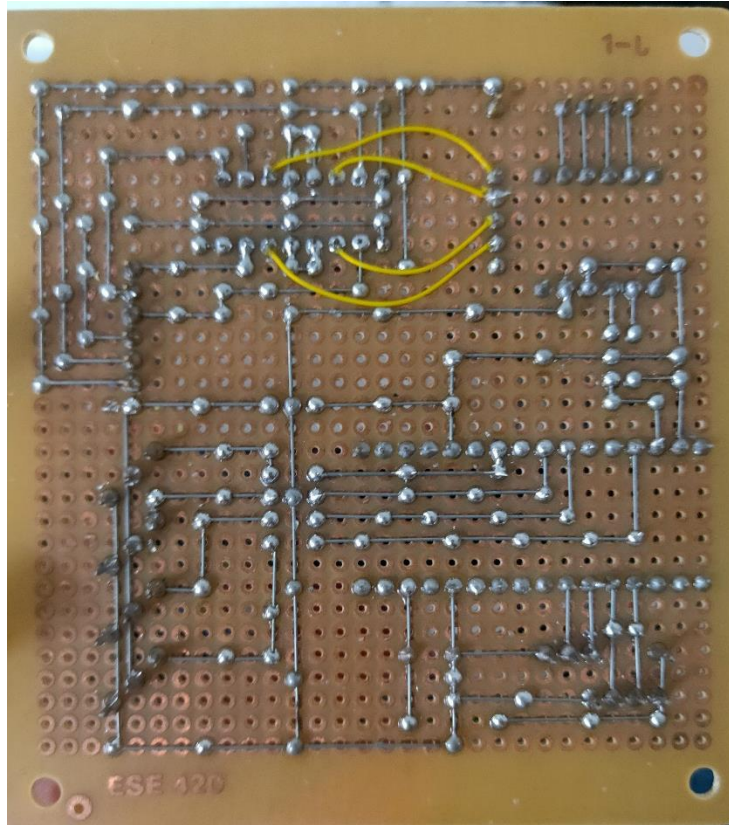


圖23 線路背面

4.3 實體操作

將零件及機械手臂放入船體中，以手機藍牙遠距離操作，如圖24。

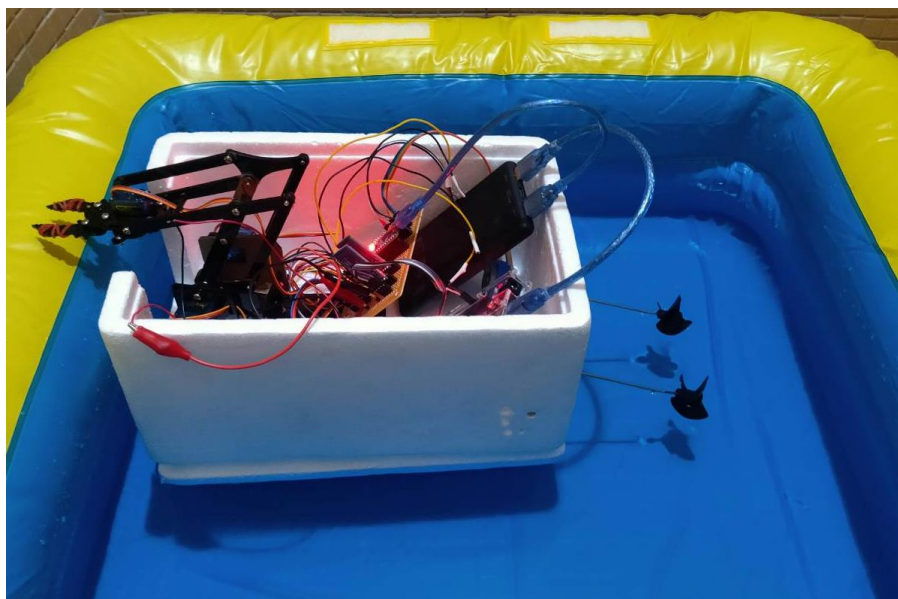


圖24 船體

利用馬達操控船體至垃圾前方，再用手機操控機械手臂抓取垃圾如圖25及圖26。

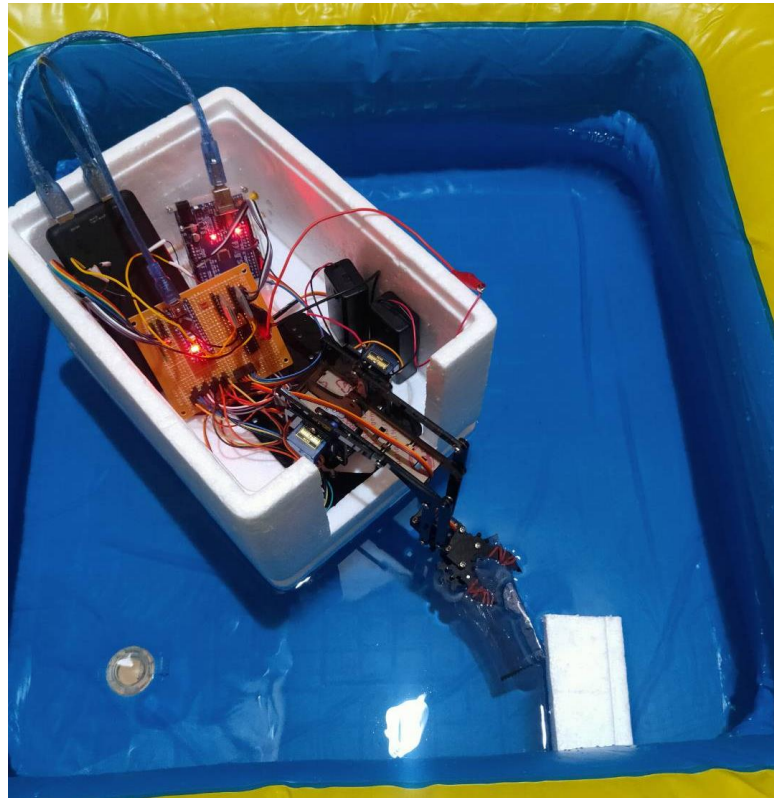


圖25 將夾取垃圾

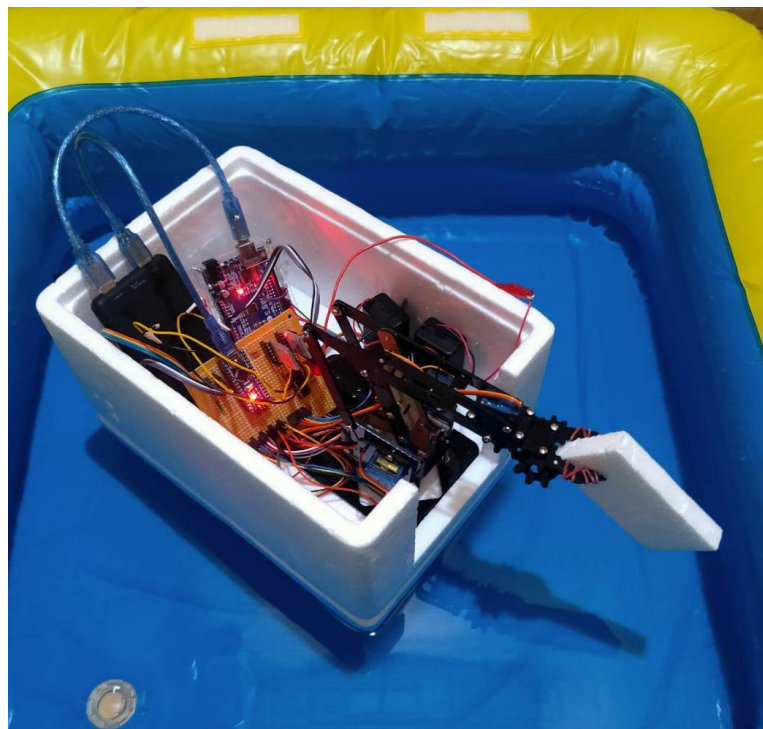


圖26 夾取垃圾

第五章 總結

5.1 結論

將上述所有程式及模型組裝過後，目前已成功架設出基本的模型，在水上也能控制機械手臂以及船體，可夾取簡易的垃圾。

本次專題中，更加了解 L293D IC 是來控制馬達轉速的也額外增加電壓讓轉速更加穩定，也認識到 MIT App Inventor 可以製作 app，方便在操作時的需求，容易下載至手機，中間有突發狀況，雙軸按鍵搖杆的數值亂跳，導致機械手臂的暴衝，最後一一排除了解是搖桿接電壓部分接觸不良，立即更換成洞洞板，解決問題，也讓所有接線的元件都在同一個板子上，減少不必要的空間，讓所有元件都裝進船體中。

5.2 未來展望

我們目前只作出簡易模型，只能在小型水池測試，無法評估遇到大浪時的未知狀況，因為範圍較小，夾取的東西過少，希望可以加強抓取的範圍，讓效率更加提升，以及藍牙模組更換為 WI-FI 功能，能在遠端操控，讓工作時的危險因素縮小，讓專題更加完善。

參考文獻

- [1] Arduino 維基百科 <https://zh.wikipedia.org/wiki/Arduino>
- [2] 什麼是 Arduino
http://cc.ee.ntu.edu.tw/~rbwu/rapid_content/course/IoT_Intro/Ch1_Arduino.pdf
- [3] MIT App Inventor 2 簡介
http://im.thu.edu.tw/upload/news_upload/Inventor%E4%BB%8B%E7%B4%B9%E5%92%8C%E4%BD%BF%E7%94%A8.pdf
- [4] How to Use the L293D
http://wiki.sunfounder.cc/index.php?title=L293D_Motor_Driver_Shield
- [5] 搖桿控制說明
<https://openhome.cc/Gossip/CodeData/mBlockArduino/mBlockArduino13.html>
- [6] HC-05 藍牙序列埠通訊模組 <https://swf.com.tw/?p=693>
- [7] MeArm 藍牙遙控 App
<https://gsyan888.blogspot.com/2014/11/android-me-arm-bluetooth-app.html>

光碟袋黏貼處

