

# 專題研究計畫書

題目：

ESP32 自動感測式垃圾桶

ESP32 遙控車

指導教授：梁惟捷

學生：

應數三/俞名鴻/S07240001

應數三/溫宏岳/S07240018

應數三/胡家宏/S07240042

應數三/張維中/S07240046

應數三/陳昱廷/S07240047

中華民國一一零年六月二十五日

# 目錄

<u>一、摘要.....</u>	<u>3</u>
<u>二、研究動機與目的.....</u>	<u>4</u>
<u>三、研究過程與方法.....</u>	<u>6</u>
<u>3-1 自動感應式垃圾桶.....</u>	<u>6</u>
<u>3-2 遙控車.....</u>	<u>12</u>
<u>四、研究成果.....</u>	<u>16</u>
<u>五、結論心得.....</u>	<u>18</u>
<u>六、參考資料.....</u>	<u>19</u>
<u>七、附錄.....</u>	<u>20</u>

## 一、摘要

在各個地方，無論是要存取資訊、與人聯絡溝通或做生意，人們都會連線使用網際網路。但是，不只人類，物品也會使用網際網路。製造業和能源產業廣泛運用了機器對機器通訊，以便追蹤機械作業、回報錯誤，及發出服務警示。近年來，越來越多日常物品也會透過網際網路連線到雲端，形成所謂的「物聯網」。物聯網正在迅速地發展，到現在，從智慧型手環、玩具、相框，到醫療設備、地震感測器和飛機，無所不包。



解決方案  
IoT物聯網應用  
Application of Internet of things



(IoT)

## 二、研究動機與目的

### (一) 研究動機

為了增進人類便利的生活，我們這組決定做出能夠有實質幫助的成品，並且能結合課堂中所學，利用 ESP32、Arduino 來完成我們的作品，因為疫情的關係，為了避免病毒的傳播。

#### (1)自動感測式垃圾桶

我們決定做出一個不用手便能控制開關的裝置，即使在本身不支援感應的裝置上，也可以應用，例如說放在垃圾桶、電燈上，又或者是學校裡電腦教室的電腦電源按鈕，適合用在公眾場合，並且必須要透過接觸才能觸發的開關，來減少病毒傳播的機會。

由於這幾年許多病毒及細菌迫害，讓大家每天都生活得提心吊膽，任何一個地方都有可能藏有細菌或病毒，而垃圾桶也是其中一個容易滋生細菌的地方，堆滿垃圾且蓋子常常經過很多人的碰觸，所以希望能盡量減少接觸到垃圾桶蓋，於是乎就決定裝在垃圾桶上，這也是相較電源開關更容易有更多人接觸的物品，在一些本來並沒有配備腳踏式、自動感應器等垃圾桶上，利用我們本次設計的產品來達到一樣的效果。

#### (2)遙控車

同樣的，也是為了減少病毒傳播的機會，在嚴峻的疫情下，相信還是有不少人需要去公司上班，以本次 covid-19 來說，病毒的生命力之強是可以附著在鞋底上來造成傳播，在辦公室裡常常會有文件需要傳閱，就算是已經改以線上傳閱的方式來傳遞一些文件、信件，還是會有以前的資料，即使是較小型的公司也不例外，於是我們也做出了第二件作品為「遙控車」。

## **(二) 研究目的**

### **(1)自動感測式垃圾桶**

為了達到減少接觸垃圾桶的目的，我們希望能製作一個特殊的垃圾桶，只要感應到有人靠近或是要丟垃圾，就能自動打開蓋子讓使用者可以直接將垃圾丟進去，並監測裡面垃圾存量達到多少來提醒使用者何時要清垃圾。



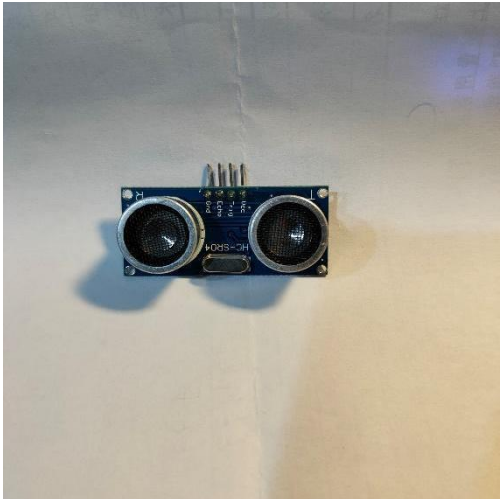
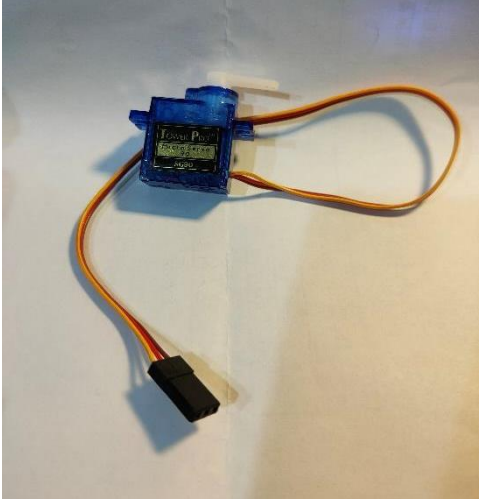
### **(2)遙控車**

了解小時候遙控汽車的原理，滿足自己的成就感，平時除了可以玩之外，再加上嚴峻的疫情，為了減少人與人之間的接觸，而誕生此作品，目的在於幫助大家運送一些文件、小東西…等，來有效減少帶著病菌走的機會。

### 三、研究過程與方法

#### 3-1 自動感測式垃圾桶

##### (一)準備材料

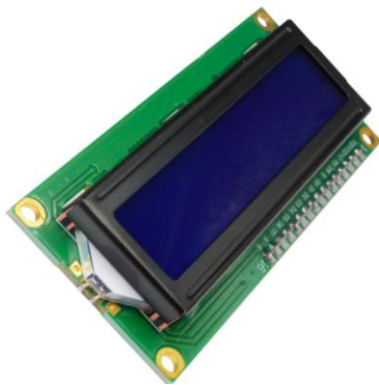
	
ESP32 開發板	USB 傳輸線
	
HC-SR04 超音波感測器 x3	伺服馬達 SG-90/小柵欄



細線



杜邦線

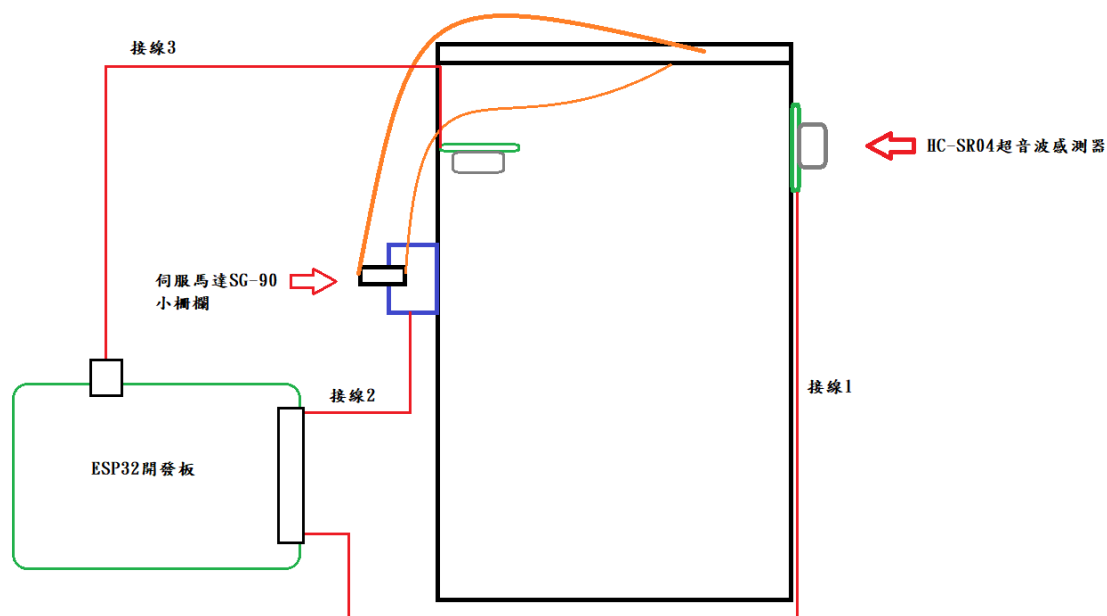


1602LCD 顯示器

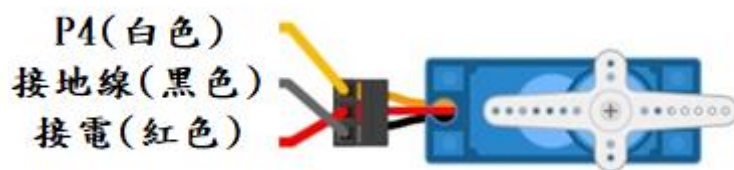


絕緣膠帶

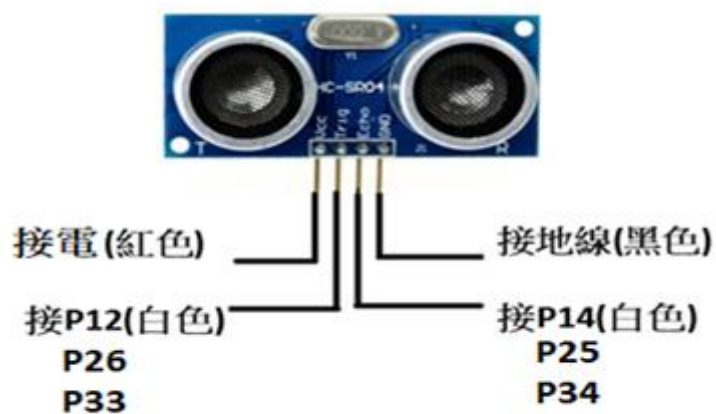
## (二) 示意圖



(自動感應式垃圾桶)



(伺服馬達 SG-90)



(超音波感測器)



### (三)製作過程

為了讓垃圾桶可以自動掀蓋，打了以下程式碼：

控制超音波感測器(外)開(HIGH)關(LOW)及延遲時間(delayMicroseconds):

```
digitalWrite(Trig,LOW);  
delayMicroseconds(5);  
digitalWrite(Trig,HIGH);  
delayMicroseconds(10);  
digitalWrite(Trig,LOW);
```

宣告超音波感測器(外)傳回時間(EchoTime)、換算距離(CMValue)的變數及延遲時間(delay):

```
float EchoTime = pulseIn(Echo, HIGH);  
float CMValue = EchoTime * 34/1000/2;  
Serial.println(CMValue);  
delay(50);
```

控制伺服馬達在超音波感測器(外)偵測物體靠近距離小於等於 15 公分時，旋轉 180 度，否則再往回旋轉至原始位置：

```
if (CMValue <= 15){  
    myServo.write(180);  
    delay(5000);  
}  
else{  
    myServo.write(0);  
}
```

為了讓超音波感測器(內)偵測垃圾量，打了以下程式碼：

控制超音波感測器(內部左側)開(HIGH)關(LOW)及延遲時間(delayMicroseconds):

```
digitalWrite(Trig2,LOW);  
delayMicroseconds(5);  
digitalWrite(Trig2,HIGH);  
delayMicroseconds(10);  
digitalWrite(Trig2,LOW);
```

宣告超音波感測器(內部左側)傳回時間(EchoTime)、換算與垃圾距離(CMValue)的變數及延遲時間(delay):

```
float EchoTime2 = pulseIn(Echo2, HIGH);  
float CMValue2 = EchoTime2 * 34/1000/2;
```

在 Arduino 監控視窗輸出 1 跟 CMValue2 代表左側的垃圾桶正在偵測：

```
Serial.println("1");  
Serial.println(CMValue2);  
delay(50);
```

控制超音波感測器(內部右側)開(HIGH)關(LOW)及延遲時間  
(delayMicroseconds):

```
digitalWrite(Trig3, LOW);  
delayMicroseconds(5);  
digitalWrite(Trig3, HIGH);  
delayMicroseconds(10);  
digitalWrite(Trig3, LOW);
```

宣告超音波感測器(內部右側)傳回時間(EchoTime)、換算與垃圾距離  
(CMValue)的變數及延遲時間(delay):

```
float EchoTime3 = pulseIn(Echo3, HIGH);  
float CMValue3 = EchoTime3 * 34/1000/2;
```

在 Arduino 監控視窗輸出 2 跟 CMValue3 代表右側的垃圾桶正在偵測：

```
Serial.println("2");  
Serial.println(CMValue3);  
delay(50);
```

在 1602LCD 顯示板上設定顯示的字元及顯示的位置，每次偵測時更新  
(clear 清除上筆數據)顯示板上的即時垃圾量：

```
lcd.clear();  
lcd.setCursor(2,0); //設定游標  
lcd.print("BLUE");  
lcd.setCursor(10,0); //設定游標  
lcd.print("RED");  
lcd.setCursor(5,1); //設定游標  
lcd.print("%");  
lcd.setCursor(13,1); //設定游標  
lcd.print("%");
```

宣告兩個變數 a、b，再設超音波感測器(內)往垃圾桶底部打的距離大約  
為 20 並設計一個公式把偵測到的距離大致換算成垃圾桶的容量：

```
int a=0;  
int b=0;  
int total_len = 20;  
  
if (CMValue2 <= total_len){  
    b=(1-(CMValue2/total_len))*100;  
}
```

```
lcd.setCursor(2,1); //設定游標
```

```
lcd.print(b);

if (CMValue3 <= total_len){
    a=(1-(CMValue3/total_len))*100;
}

lcd.setCursor(10,1); //設定游標
lcd.print(a);
}
```

#### (四) 遇到的困難




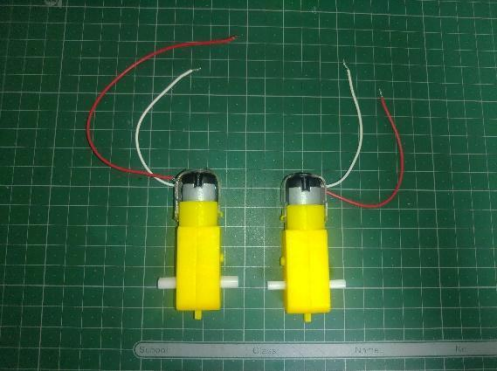


在程式碼的部分，其實就跟上課練習的差不多，沒遇到什麼問題。問題在於如何固定伺服馬達跟拉線的長度，讓垃圾桶蓋子能夠順暢地開啟與關閉，而且如果太緊繃伺服馬達固定不牢固很容易就會掉下來，接在上面打開關閉蓋子的線也要順應馬達旋轉的角度變鬆和變緊，這些都是需要多方嘗試就可以找到最佳且美觀的作法。

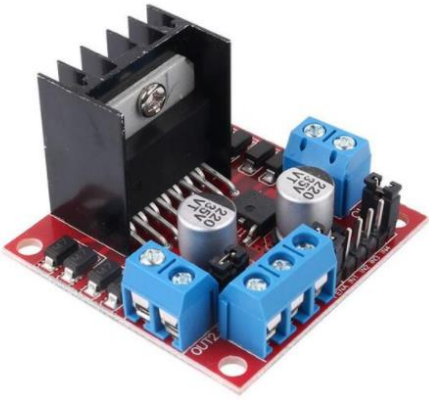

另外兩個超音波感測器接線要重新找，有些給的腳位插下去沒辦法執行，在顯示器的部分也有一點困難，要估計大概的距離，然後顯示垃圾桶的容量，這部分測試了很久，也失敗很多次。

而我們製作的垃圾桶因為都是小型的，空間不大，所以對於超音波感測器的發揮不是非常好，有可能會被邊緣影響，如果我們製作大型一點的垃圾桶效率應該可以更優化。

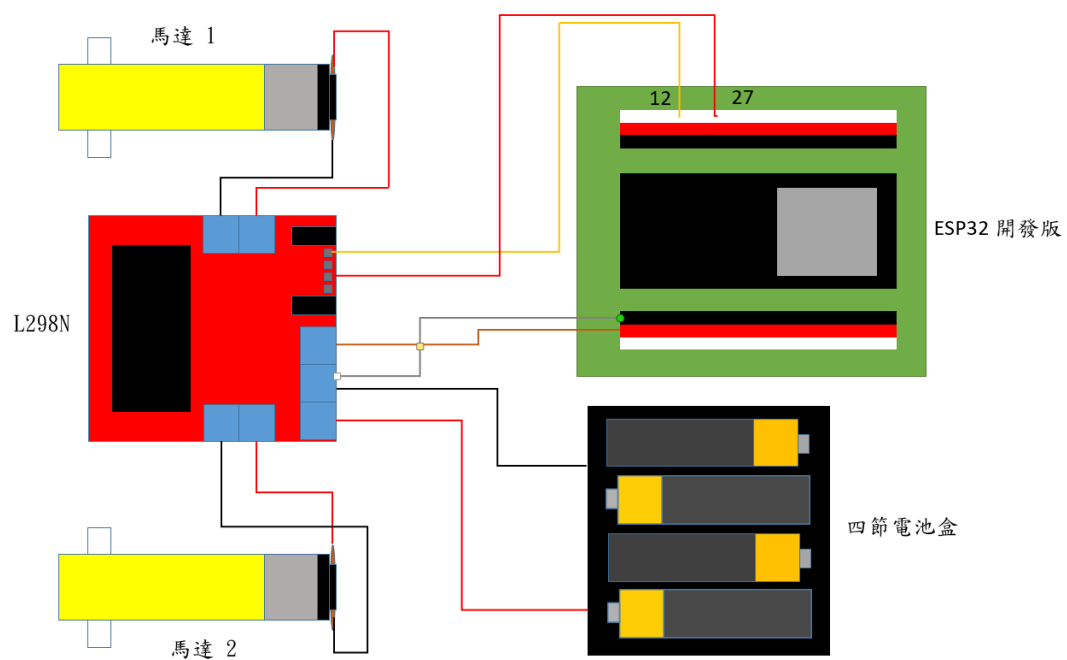
## 3-2 遙控車

### (一) 準備材料

	
ESP32 開發板	USB 傳輸線
	
車底盤及輪胎	馬達
	
四節電池盒	杜邦線

	
L298N	行動電源

## (二)示意圖



### (三)製作過程

為了使用藍芽功能，是用了以下程式：

```
#include <BluetoothSerial.h>
BluetoothSerial myBT;
myBT.begin("Camels BT");
```

給定右輪的輸出為腳位 27；左輪的輸出為腳位 12：

```
const byte r = 12;
const byte l = 27;
pinMode(r, OUTPUT);
pinMode(l, OUTPUT);
```

由 APP 經由藍芽送給開發版訊號：

```
char BTdata = myBT.read();
Serial.println(BTdata);
```

當輸入方向鍵「上」(APP 輸入數值「1」)時，左右兩輪同時輸出，達到前進的效果：

```
if (BTdata == '1'){
    digitalWrite(r, HIGH);
    digitalWrite(l, HIGH);}
```

當輸入方向鍵「右」(APP 輸入數值「3」)時，左輪輸出，右輪靜止，達到右轉的效果：

```
if (BTdata == '3'){
    digitalWrite(r, LOW);
    digitalWrite(l, HIGH);}
```

當輸入方向鍵「左」(APP 輸入數值「4」)時，右輪輸出，左輪靜止，達到左轉的效果：

```
if (BTdata == '4'){
    digitalWrite(r, HIGH);
    digitalWrite(l, LOW);}
```

當輸入方向鍵「下」(APP 輸入數值「2」)時，左右兩輪皆不輸出，達到煞車的效果：

```
if (BTdata == '2'){
    digitalWrite(r, LOW);
    digitalWrite(l, LOW);}
```

持續一秒後，若沒有持續輸入訊號，則煞車：

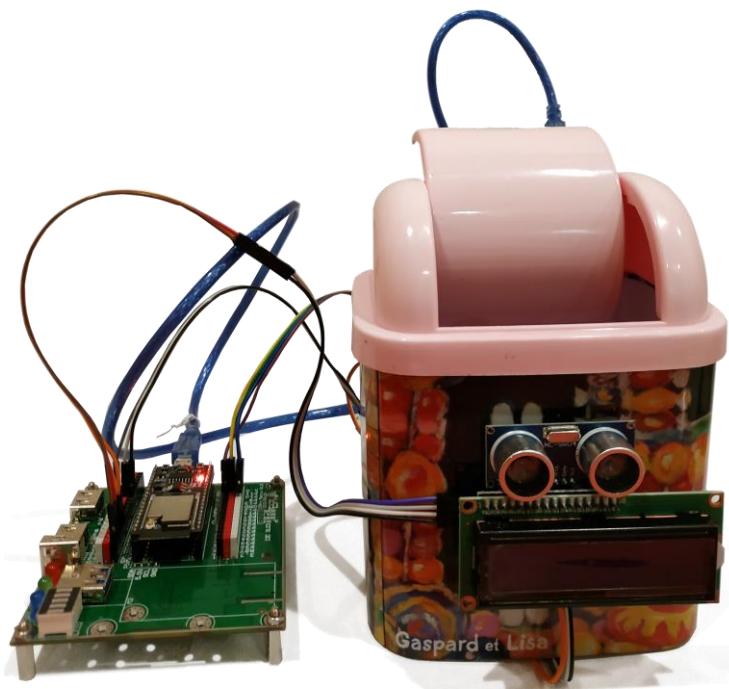
```
delay(1000);
digitalWrite(r, LOW);
digitalWrite(l, LOW);
```

#### (四)遇到的困難：

第一個遇到的困難是在我執行程式碼時顯示「開發板 NodeMCU-32S 編譯錯誤」，但程式碼大家都檢查過沒問題，花了半天一直找不到原因，到最後才發現是輸出 pin 腳位的名字出問題，把名字改了就執行成功了；再來就是在進行組裝時，發現缺了接馬達的線，跑了 3 間五金行找不到，最後才在書局找到；最後一個困難則是沒有電鍍的工具，找了許多辦法才把線固定在馬達上。

#### 四、研究成果

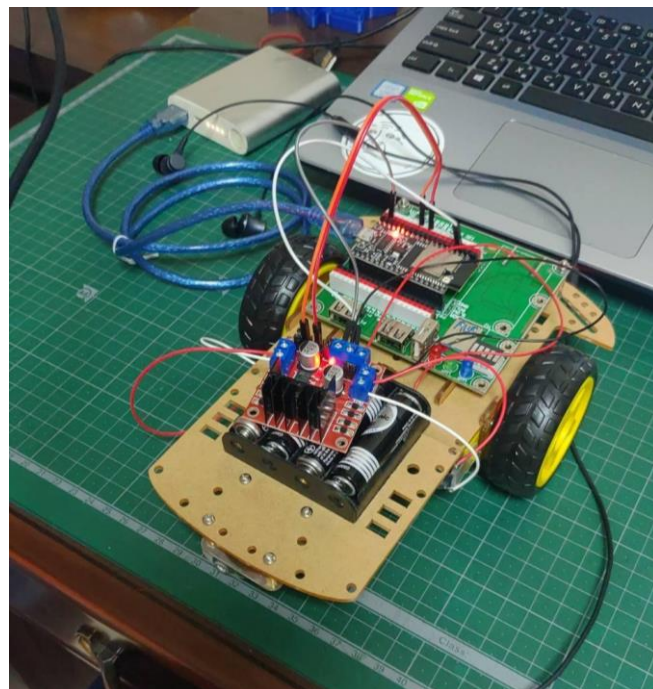
垃圾桶： <https://youtu.be/4fGe3sU6XCs>







遙控車: <https://youtu.be/Anp41bIXMSE>



## 五、結論心得

這次因為疫情的關係，我們製作過程多少有一些影響，因為這門課算是一堂偏實作類的課程，我們沒辦法面對面一起討論且組裝零件，所以我們就各自分工，上網購買一些需要的零件，這堂課讓我們了解許多可以用來應用的電子產品，還有 ESP32 開發板的應用，當然開發板不是只使用了這種，所以還有更多我們可以探討活動的東西，而這次製作出來的成品雖然跟原本想做的不同，但最後都有成功做出來，大家分散各地所以我們很難把所有東西集合再一起組裝，只能各自運用手邊零件，但我們做的這兩個作品其實也可以結合，把垃圾桶組裝在遙控車上，如此一來我們就可以透過手機操控垃圾桶的位置，就不用走來走去丟垃圾了，可以說是一個懶人垃圾桶，躺床也有垃圾桶過來。

## 六、參考資料

素材及內容：

<https://www.salesforce.com/tw/crm/internet-of-things/>

<http://www.fstop.com.tw/CustomizedContent.aspx?id=098055E441E97580>

<https://www.techapple.com/archives/3348>

<https://tutorials.webduino.io/zh-tw/docs/basic/component/servo.html>

文獻：

[http://www.kyicvs.khc.edu.tw/kyicvs/images/ckfinder/files/20200913\\_003320.pdf](http://www.kyicvs.khc.edu.tw/kyicvs/images/ckfinder/files/20200913_003320.pdf)

<https://www.shs.edu.tw/works/essay/2019/03/2019032622113841.pdf>

<http://ir.hust.edu.tw/bitstream/310993100/4293/1/機械系四丙專題自動化垃圾桶>

影片：

<https://www.instructables.com/Smart-Garbage-Can-With-Car/> (垃圾桶)

<https://www.youtube.com/watch?v=Wfpvj9KlEMs> (遙控車)

## 七、附錄

### 1. 自動感測式垃圾桶程式碼

```
int Trig  = 12;
int Echo  = 14;
int Trig2 = 26;
int Echo2 = 25;
int Trig3 = 33;
int Echo3 = 34;

#include <ESP32Servo.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2); //LCD 位址與大小
Servo myServo; //建立一個伺服馬達物件 yellow : P4

void setup() {
  //VCC: 紅 , GND: 黑 , ECHO: P14,25,34 , TRIG: P12,26,33
  Serial.begin(115200);
  pinMode(Echo, INPUT);
  pinMode(Trig, OUTPUT);
  pinMode(Echo2, INPUT);
  pinMode(Trig2, OUTPUT);
  pinMode(Echo3, INPUT);
  pinMode(Trig3, OUTPUT);
  myServo.attach(4, 500, 2400);
  lcd.init(); //初始化
  lcd.backlight(); //開啟 LCD 背光
}

void loop() {
  digitalWrite(Trig, LOW);
  delayMicroseconds(5);
  digitalWrite(Trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Trig, LOW);

  float EchoTime = pulseIn(Echo, HIGH);
  float CMValue = EchoTime * 34/1000/2;

  Serial.println(CMValue) ;
  delay(50);

  if (CMValue <= 15){
    myServo.write(180);
```

```

    delay(5000);
}
else{
    myServo.write(0);
}

digitalWrite(Trig2, LOW);
delayMicroseconds(5);
digitalWrite(Trig2, HIGH);
delayMicroseconds(10);
digitalWrite(Trig2, LOW);

float EchoTime2 = pulseIn(Echo2, HIGH);
float CMValue2 = EchoTime2 * 34/1000/2;

Serial.println("1");
Serial.println(CMValue2) ;
delay(50);

digitalWrite(Trig3, LOW);
delayMicroseconds(5);
digitalWrite(Trig3, HIGH);
delayMicroseconds(10);
digitalWrite(Trig3, LOW);

float EchoTime3 = pulseIn(Echo3, HIGH);
float CMValue3 = EchoTime3 * 34/1000/2;

Serial.println("2");
Serial.println(CMValue3) ;
delay(50);

lcd.clear();
lcd.setCursor(2,0); //設定游標
lcd.print("BLUE");
lcd.setCursor(10,0); //設定游標
lcd.print("RED");
lcd.setCursor(5,1); //設定游標
lcd.print("%");
lcd.setCursor(13,1); //設定游標
lcd.print("%");

int a=0;
int b=0;
int total_len = 20;

if (CMValue2 <= total_len){
    b=(1-(CMValue2/total_len))*100;
}

```

```
lcd.setCursor(2,1); //設定游標
lcd.print(b);

if (CMValue3 <= total_len){
  a=(1-(CMValue3/total_len))*100;
}

lcd.setCursor(10,1); //設定游標
lcd.print(a);
}
```

## 2. 遙控車程式碼

```
#include <BluetoothSerial.h>
BluetoothSerial myBT;
char incomeData;
const byte r = 12;
const byte l = 27;
void setup() {
  Serial.begin(115200);
  myBT.begin("Camels BT");
  pinMode(r, OUTPUT);
  pinMode(l, OUTPUT);
}
void loop() {
  while(myBT.available())
  {
    char BTdata = myBT.read();
    Serial.println(BTdata);
    if (BTdata == '5'){
      digitalWrite(r, HIGH);
      digitalWrite(l, HIGH);}
    if (BTdata == '3'){
      digitalWrite(r, LOW);
      digitalWrite(l, HIGH);}
    if (BTdata == '4'){
      digitalWrite(r, HIGH);
      digitalWrite(l, LOW);}
    if (BTdata == '2'){
      digitalWrite(r, LOW);
      digitalWrite(l, LOW);}
  }
  delay(1000);
  digitalWrite(r, LOW);
  digitalWrite(l, LOW);
}
```