

目錄

動機.....	2
文獻探討.....	4
程式語言的三個主要層次.....	4
直譯語言與編譯語言.....	5
C 語言.....	6
Python.....	8
比較 Python 跟 C 語言.....	9
學習跟實作.....	10
Arduino.....	11
LED.....	11
蜂鳴器.....	14
小遊戲.....	20
ESP 32.....	24
開關 LED 燈.....	24
esp 32 遙控車.....	27
組裝 3D 列印機.....	29
圖像辨識.....	32
Opencv 的基本功能.....	33
手勢辨識.....	38
圖像辨識結合遙控車.....	41
計畫一 ESP 32 cam.....	41
計畫二 樹莓派.....	43
結論與感謝.....	47
附錄.....	49
參考資料.....	50

動機

我想做程式相關的題目是因為我對程式設計很有興趣，我覺得可以用自己做（寫）出來的東西來完成某件事或解決某個問題是一件特別有趣的事情，程式跟數學一樣可以用不同的寫法來解決同一件事，每次成功解決問題後都會特別地有成就感，解不出來的時候就會覺得：「不就是數字加 ABC 再加一點標點符號嗎？我就不信我會想不出來，我跟你拚了！」。

我是一個很懶惰的人，我覺得要自己去拿東西跟整理環境是超級麻煩的事，所以我喜歡把所有要做的事堆在一起做，雖然覺得這樣沒什麼不好，已經比隨手做省下不少時間跟精力了，但是我還是覺得好累，我想要所有麻煩的事都交給別人來做，但是，這是不可能的，所以我暫時放棄這個想法。直到我看鋼鐵人的時候，看到他有一個可以幫他拿東西跟整理房間的機器手臂，我也想要有一個可以幫我整理房間的機器手臂，這也變成我的夢想。

我這次專題跟國中專題的目標都是縮短我跟夢想的距離，國中的專題是藉由自己做出一台遙控車來熟悉電路，但是現成的模組（裝好就可以用）的限制太多不符合我的需求，所以高中的重點放在學習程式然後做出自己需要的功能，離我的夢想——我的夢想是做出一個可以幫我整理房間的機器手臂更進一步，所以這次我幫我自己訂的目標是做出一台可以用手勢控制的遙控車。

文獻探討

程式語言的三個主要層次

1、機器語言：

機器語言就是機器（CPU）看得懂的語言，由 0 跟 1 組成，直接控制通電或不通電，對人類來說是最不容易看懂的語言，但也是執行最快的語言。

2、組合語言（低階語言）：

因為機器語言的實用性太差，所以發明了組合語言，組合語言開始有英文（縮寫）跟符號。

3、高階語言：

機器語言還是太難了，所以發明了高階語言，高階語言就是我們現在常用的程式語言，像是 Python、C++、C、Java，基本是現在看得到的程式語言都是高階語言。

高階語言執行前都需要編譯成機器語言，電腦只看得懂機器語言。

直譯語言與編譯語言

編譯語言在程式執行前會先透過編譯器把程式碼編譯成電腦看得懂的機器碼，再執行，一般會有一個副檔名為.exe (Windows) 的執行檔，例如 C++跟 C 語言。

直譯語言不同於編譯語言，執行語言在執行的時候會一行一行的編譯再執行，所以一般不會有執行檔，例如 Python。

編譯語言與直譯語言對我來說最大的不同是：編譯語言只要有一個語法錯誤就沒辦法執行跟編譯，但是直譯語言不同，直譯語言會執行到語法錯誤的地方然後停下來，所以在除錯的時候會體驗到不同的痛苦。

C 語言

在 1969 到 1973 年，肯·湯普森與丹尼斯·里奇為了讓 Unix 系統更高效，在貝爾實驗室以 B 語言為基礎開發了 C 語言，之後 C 語言越來越多人使用。美國國家標準局為了避免大家使用 C 語言的語法差異，為 C 語言訂了一套國際標準語法，稱為 ANSI，現在使用的版本主要是 C18，每個版本都有些差異，不同的版本可能會互不相容。

我覺得 C 語言最大的特色，是可以直接控制機記憶體（指標），所以很多人說會用指標才算學會 C 語言。C 語言跟現在常用的程式語言一樣可以使用函數、循環、條件，但是沒有物件導向的功能。

C 語言在執行前需要先用編譯器編譯成機器碼再執行，會產生一個副檔名是 .exe 的執行檔，所以不需每次執行前都需要編譯，只有在修改程式時才需重新編譯，執行的速度也比 Python 快很多。

```

1  #include<stdio.h>
2  #define size 8
3
4  void bubbleSort(int *);
5  void printArray(int *);
6  int main(){
7      int date[size]={16,25,39,27,12,8,45,63};
8      bubbleSort(date);
9      printArray(date);
10 }
11
12 void bubbleSort(int *array){
13     void swap (int *,int *);
14     int i,j;
15     for(i=0;i<size-1;i++){
16         for(j=0;j<size-1-i;j++){
17             if(array[j]>array[j+1]){
18                 swap(&array[j],&array[j+1]);
19             }
20         }
21     }
22 }
23
24 void swap(int*a,int*b){
25     int temp;
26     temp=*a;
27     *a=*b;
28     *b=temp;
29 }
30
31 void printArray(int *array){
32     int i;
33     for(i=0;i<size;i++){
34         printf("%d ",array[i]);
35     }
36 }
37 }

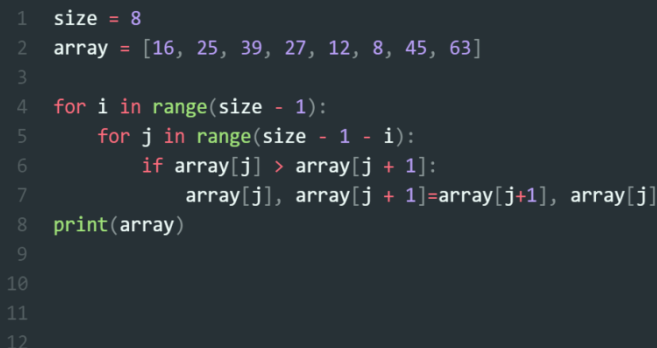
```

上面是一段 C 語言的程式，這段程式可以把串數字按照大小排好，這段程式有用到自訂函式、指標、迴圈、判斷式等功能。程式的一到十行主要在定義變數，十二到二十三行在判斷數字的順序是否需要交換，二十五行到三十行用來交換數字的順序，三十二行到三十七行負責輸出排好順序的數字。

Python

Python 是一種高級程式設計語言，由 Guido van Rossum 在 1989 年創建，並於 1991 年首次發佈。特色是簡潔、可讀性強。相對 C 語言 Python 好維護又好學，最近幾年幾乎都是最熱門的程式語言的前三名。

我覺得 Python 最大的特色，是擁有非常多的模組，所以 Python 可以使用的領域非常的廣，例如：網頁開發、資料科學、機器學習、自動化腳本、遊戲開發域、影像辨識。Python 除了執行速度慢以外沒有太大的缺點。

A screenshot of a code editor with a dark background and light-colored text. The code is a Python implementation of a bubble sort algorithm. It starts with a list 'array' containing the numbers [16, 25, 39, 27, 12, 8, 45, 63]. A 'for' loop iterates over the list, and an inner 'for' loop compares adjacent elements. If an element is greater than the next one, they are swapped. The list is printed at the end. The code is numbered from 1 to 12 on the left side of the editor.

```
1 size = 8
2 array = [16, 25, 39, 27, 12, 8, 45, 63]
3
4 for i in range(size - 1):
5     for j in range(size - 1 - i):
6         if array[j] > array[j + 1]:
7             array[j], array[j + 1] = array[j + 1], array[j]
8 print(array)
9
10
11
12
```

這段 Python 程式的功能跟上面 C 語言的功能完全一樣。

比較 Python 跟 C 語言

Python 跟 C 語言對我來說有兩個最大的差別，執行速度跟函式庫這兩個部分，相同的程式在大部分的情況下 C 語言執行的速度會比 Python 快很多，Python 有很多的函式庫可以使用，可以更容易地解決問題。

執行速度測試：

分別讓 C 語言跟 Python 執行 $a=a+1$ 這段程式 10^4 次。

測試環境：JDoodle （線上編譯器）

結果：

C 語言:2.60 秒

Python:9.01 秒

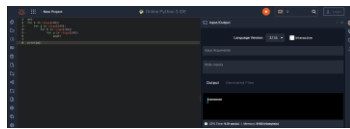
相同的程式 C 語言比 Python 快了將近 3.5 倍。

下圖為測試截圖

C 語言



Python



學習跟實作

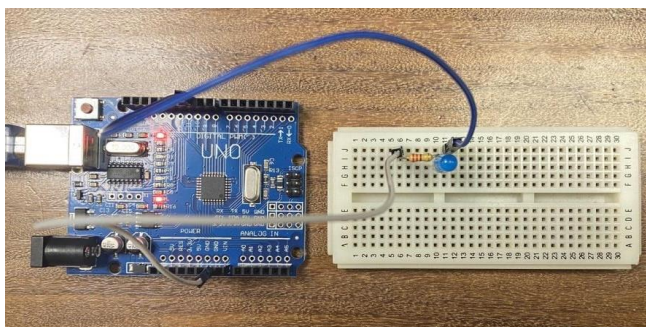
我一開始接觸的程式是 Arduino，一邊學習 C 語言一邊學習簡單的電路，接下來為了寫手部辨識學了 Python，為了考 APCS 又認真的再學一次 C 語言。我覺得我讓我的專題有一點極限，我的專題是一個實做型的專題，但是我再在這次專題裡用到的東西幾乎都是自學的，所以前面花在學程式的時間有一點長，導致我實作的時間有一點短，我覺得我已經快要變成學習型專題了。

我的專題是實作型專題，但是我花了很多時間在學習寫程式，所以我想要簡單的說一下我學程式的過程，我最一開始跟爸爸的朋友張凌倩老師學，我們先從 Arduion 開始學了一段時間後，換成 Python 基礎的部分學完以後我就改成自學了，我先學了 C 語言，然後去考了 APCS（觀念 2 級分實作 3 級分），但是成績很不理想，同時我也在研究 Python 的圖像辨識，最後為了使用樹莓派又學了一點 Linux 的指令。

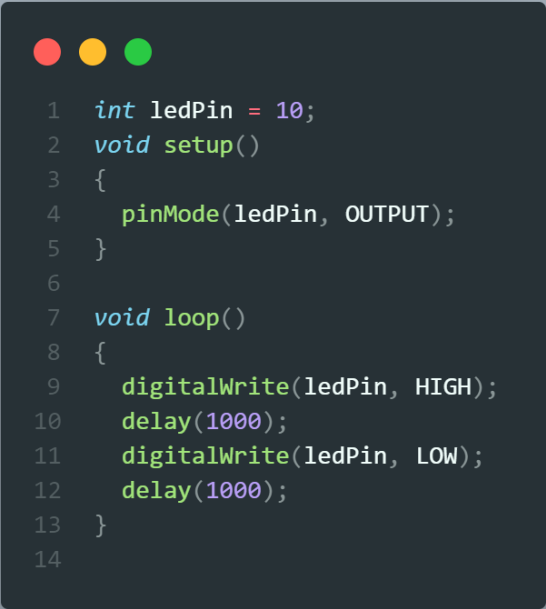
Arduino

LED

控制 LED 燈通常是學習 Arduino 第一個做的小專案。



程式最開始定義了一個整數變數 ledPin 用來處存腳位的位子，setup 函數只會在程式開始時執行一次，一般用來設定腳位的功能，在 setup 函數中用 pinMode 定義 ledPin (pin10) 是輸出功能。loop 函數會重複執行，會用來放要被執行的動作，在 loop 函數中 digitalWrite 控制 ledPin 輸出高電位或是底電位 (HIGH、LOW)，delay 是用控制持續的時間，所以這段程式執行的結果是 LED 燈亮一秒暗一秒無限重複。



```
1  int ledPin = 10;
2  void setup()
3  {
4      pinMode(ledPin, OUTPUT);
5  }
6
7  void loop()
8  {
9      digitalWrite(ledPin, HIGH);
10     delay(1000);
11     digitalWrite(ledPin, LOW);
12     delay(1000);
13 }
14
```

這個小專案只會燈亮一秒暗一秒太無聊了，
所以可以加上一個開關來控制 LED 的亮與暗。

先定義變數 `buttonState` 是處存按鈕的狀態（HIGH 或 LOW，HIGH 是通電 LOW 是不通），`buttonPin` 是按鈕的腳位，`ledPin` 是 LED 燈的腳位。用 `digitalRead` 讀取按鈕的狀態存到 `buttonState`，然後判斷按鈕的狀態來控制 LED 燈的開關。



```
1  int buttonState;
2  int buttonPin = 2;
3  int ledPin = 9;
4
5  void setup() {
6      pinMode(buttonPin, INPUT);
7      pinMode(ledPin, OUTPUT);
8  }
9
10 void loop() {
11     buttonState = digitalRead(buttonPin);
12
13     if(buttonState == HIGH){
14         digitalWrite(ledPin,HIGH);
15     }
16     else{
17         digitalWrite(ledPin, LOW);
18     }
19 }
```

蜂鳴器

蜂鳴器可以分成有源蜂鳴器跟無源蜂鳴器，有源蜂鳴器只能輸出固定的聲音，只要通電就可以已發出聲音，無源蜂鳴器可以根據輸入的不同發出不通頻率的聲音。

有源蜂鳴器



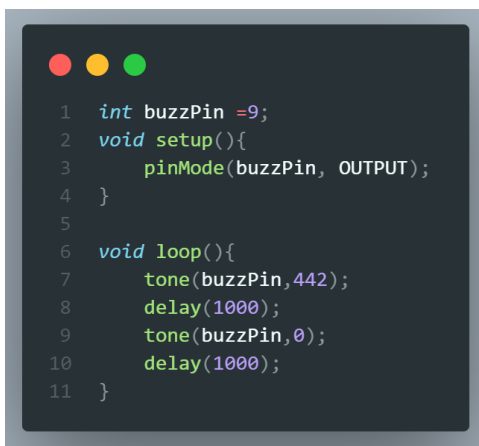
```
1  int buzzPin =9;
2  void setup(){
3      pinMode(buzzPin, OUTPUT);
4  }
5
6  void loop(){
7      digitalWrite(buzzPin,HIGH);
8      delay(1000);
9      digitalWrite(buzzPin, LOW);
10     delay(1000);
11 }
```

有源蜂鳴器的程式比較簡單，但是變化相對比較少，用法跟 LED 燈一樣，只要控制通電的時間就可以了。左邊的程式是控制源蜂鳴器發出聲音一秒停止一秒，無限重複。

無源蜂鳴器

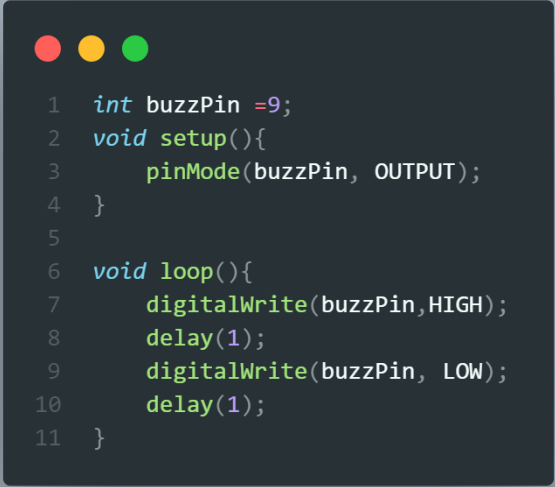
無源蜂鳴器的變化比較多，但是用起來會稍微麻煩一點，需要自己設好需要的頻率，可以用來放音樂。

無源蜂鳴器的程式有兩種寫法，第一種是基本的寫法，麻煩又不好用。蜂鳴器裡面有一個膜跟電磁鐵，透過電磁鐵的吸放讓膜產生震動發出聲音，所以可以利用通電或不通電來控制膜震動的頻率（圖一）。方法二也是比較常用的方法，Arduino 有一個專門用來控制無源蜂鳴器的函數，tone 函數可以直接讓蜂鳴器輸出指定的頻率（圖二）。



```
1  int buzzPin =9;
2  void setup(){
3      pinMode(buzzPin, OUTPUT);
4  }
5
6  void loop(){
7      tone(buzzPin,442);
8      delay(1000);
9      tone(buzzPin,0);
10     delay(1000);
11 }
```

（圖一）



```
1  int buzzPin =9;
2  void setup(){
3      pinMode(buzzPin, OUTPUT);
4  }
5
6  void loop(){
7      digitalWrite(buzzPin,HIGH);
8      delay(1);
9      digitalWrite(buzzPin, LOW);
10     delay(1);
11 }
```

(圖二)

知道蜂鳴器的基本用法後就可以做一些有趣的小專案，上音樂課的時候有一個作業是每組要表演一個音樂小節目，我的小組要用蜂鳴器來演奏，我負責的是程式的部分。



```
1  int a[63]={4,2,2,4,2,2,2,2,2,
    2,8,4,2,2,4,2,2,2,2,2,2,8,4,
    6,2,2,2,2,2,6,2,2,2,2,2,6,2,
    2,2,2,2,6,2,2,2,2,2,1,1,1,1,
    2,2,1,1,1,1,2,2,1,1,1,1,8};
2  int b[63]={525,0,393,525,0,39
    3,525,393,525,661,786,701,0,5
    90,701,0,590,701,590,497,590,
    393,525,525,662,590,525,525,4
    97,497,590,701,497,590,525,52
    5,625,590,525, 525,497,497,59
    0,701,497,525,525,525,497,44
    2,497,525,525,662,590,525,59
    0,662,662,786,701,662,701,78
    6};
3
4  int buzzPin=2;
5
6  void setup() {
7      digitalWrite(buzzPin,OUTPUT
    T);
8
9  }
10 void loop() {
11     for(int i=0;i<63;i++){
12         tone(buzzPin,b[i]);
13         delay(a[i]*250);
14     }
15 }
```

在程式的開頭有兩個陣列分別用來拍子跟頻率，用迴圈來讀取它控制蜂鳴器。

在這個小專案中有一個需要注意的地方，蜂鳴器的音可能會不準，讓它輸出 442 赫茲，它不一定會輸出 442 赫茲，可能會是 430、450……之類的，會有誤差，所以最好要先調音。

我當時有另外寫一個用來調音的程式，但是時間不夠沒有用到，所以我不太確定這個程式可不可以用。

這個方法需要配合調音器使用，慢慢的把會用到的音找出來。

```

1  int bottonPin1=2; // 頻率加2
2  int bottonPin2=3; // 頻率減2
3  int sum=442; // 頻率
4  int berPin=4;
5  int bottonA=0;
6  int bottonB=0;
7
8
9
10
11 void setup() {
12     pinMode(bottonPin1, INPUT);
13     pinMode(bottonPin2, INPUT);
14     pinMode(berPin, OUTPUT);
15     Serial.begin(115200);
16 }
17
18
19 void loop() {
20
21
22
23
24
25     Serial.println(sum);
26     tone(berPin, sum);
27
28     bottonA=digitalRead(bottonPin1);
29     bottonB=digitalRead(bottonPin1);
30     if(bottonA==HIGH){
31         sum=sum+2;
32     }
33     if(bottonB==HIGH){
34         sum=sum-2;
35     }
36     delay(100);
37
38
39 }
40

```

這段程式可以用來調音，需要配合兩個按鈕跟一個蜂鳴器使用，如果覺得音太高就按下增加頻率的按鈕，相反，如果因太高就按下降低頻率的按鈕，直到音轉準為止，同時電腦也會顯示目前的頻率。

小遊戲

結合前面用過的 LED、按鈕、蜂鳴器，可以做出一個簡單的記憶遊戲。LED 燈會按順序閃爍，玩家需要記下 LED 燈閃爍的順序，按照順序按下對應的按鈕，順序正確 LED 燈會從左到右閃三次，然後進入下一關，如果順序不正確 LED 燈會一起閃三下，然後遊戲結束。遊戲一共有十個關卡，每過一關需要記憶的順序都會增加一個，每次的順序都是隨機產生。

這個小專案是我學習 Arduino 的一個成果檢驗，我的目標是在完全不查資料和問 AI 的情況下自己寫出來。

我覺得這樣的練習對學習寫程式很有幫助，因為我在遇到不會的事情的時候會習慣直接去查資料，但是在寫程式的時候同樣的問題可以用不同的方式去解決，規定自己不能查資料可以練習到跟網路上不同的寫法也可以知道自己對哪些部分還不熟。

這個程式可以分成八個部分，定義變數、seed 函數、outPut 函數、input 函數、pass 函數、noPass 函數、setup 函數、loop 函數。

seed 函數用來產生關卡，我用了 rand 函數生成隨機數，把這個數字除以 10 取於數可以得到 0 到 10 的隨機數，如果這個數字等於 5、6、7、8 其中一個，這個隨機的數字就會被放到處存關卡的變數了，重複以上的步驟直到指定的次數，最開始是四次，每通關一次關卡的長度就會加一。

outPut 函數是用來點亮指定的 LED 燈跟發出對應的聲音，每一個 LED 燈都有對應一個固定的聲音，隨著關卡的增加 LED 燈亮起的時間會逐漸減少，可以記憶時間也會變短。

input 函數是用來紀錄使用者輸入的順序，然後確定順序是否正確，正確會回傳 0，不正確會回傳 0 以外的整數。

pass 函數是通關的指示燈，通關的時候 LED 燈會從左到右從右到左再從左到右依序點亮。

nopass 函數是順序錯誤的指示燈，LED 燈會一起閃三下。

```

1  #include <stdlib.h>
2
3  int buttonPin[] = {1, 2, 3, 4};
4  int ledPin[] = {5, 6, 7, 8};
5  int haha[10] = {0};
6  int buzzPin = 12;
7  int level = 4;
8  int ans[10] = {0};
9
10 void seed(int c) {
11     int a = 0;
12     int b;
13
14     while (c > a) {
15         b = rand() % 10;
16
17         if (b != buttonPin[0] && b != buttonPin[1] && b != buttonPin[2] && b != buttonPin[3]) {
18             haha[a] = b;
19             a++;
20         }
21     }
22 }
23
24 void outPut(int c) {
25     for (int i = 0; i < c; i++) {
26         digitalWrite(haha[i], HIGH);
27         tone(buzzPin, haha[i] * 50);
28         delay(1000 - (c * 10));
29         digitalWrite(haha[i], LOW);
30     }
31 }
32
33 int input(int c) {
34     int a = 0;
35     for (int i = 0; i < c; i++) {
36         for (int j = 0; j < 4; j++) {
37             if (digitalRead(buttonPin[j]) != LOW) {
38                 ans[i] = buttonPin[j];
39             }
40         }
41     }
42
43     for (int i = 0; i < c; i++) {
44         if (haha[i] != ans[i]) {
45             a++;
46         }
47     }
48     return a;
49 }

```

(圖上)

```

51 void pass() {
52     int a = 0;
53     int b = 1;
54     for (int i = 0; i < 3; i++) {
55         while (a > -1 && a < 4) {
56             digitalWrite(ledPin[a], HIGH);
57             delay(40);
58             a += b;
59         }
60         b = -b;
61     }
62 }
63
64 void noPass() {
65     for (int i = 0; i < 3; i++) {
66         for (int j = 0; j < 4; j++) {
67             digitalWrite(ledPin[i], HIGH);
68         }
69         delay(1000);
70     }
71 }
72
73 void setup() {
74     for (int i = 0; i < 4; i++) {
75         pinMode(buttonPin[i], INPUT);
76         pinMode(ledPin[i], OUTPUT);
77     }
78     pinMode(buzzPin, OUTPUT);
79 }
80
81 void loop() {
82     while (level < 14) {
83         seed(level);
84         outPut(level);
85         if (input(level) > 0) {
86             noPass();
87             return;
88         } else {
89             pass();
90             level++;
91         }
92     }
93 }
94

```

(圖下)

ESP 32

開關 LED 燈

有一天我看到了一個用網頁控制 ESP32 內建 LED 燈的教學影片，我覺得很有趣就跟著做做看。



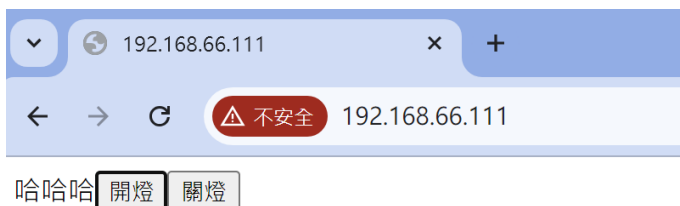
可以控制圖中藍色 LED 開關。


```

1 #include <WiFi.h>
2 #include <WebServer.h>
3
4 const char* ssid= "Ci-Xin";
5 const char* password="I do not know";
6 const int ledPin=2;
7
8 WebServer server(80);
9
10 void handelRoot(){
11 String HTML= "<!DOCTYPE html>\
12 <html>\
13 <head><meta charset='utf-8'></head>\
14 <body>\
15 哈哈\
16 <script>var xhttp= new XMLHttpRequest();\
17     function sw(arg){\
18         xhttp.open('GET', '/sw?led=' + arg, true);\
19         xhttp.send();\
20     }\
21 <button onmousedown=sw('on')>開燈</button>\
22 <button onmousedown=sw('off')>關燈</button>\
23 </body>\
24 </html>";
25 server.send(200,"text/html",HTML);
26 }
27
28 void ledSwitch(){
29 String state=server.arg("led");
30 if(state=="on"){
31     digitalWrite(ledPin,HIGH);
32 }
33 else if (state=="off"){
34     digitalWrite(ledPin,LOW);
35 }
36 server.send(200,"text/html","led is <b>"+state+"</b>.");
37 }
38
39 void setup() {
40     pinMode(ledPin,OUTPUT);
41     Serial.begin(115200);
42
43     WiFi.mode(WIFI_STA);
44     WiFi.begin(ssid, password);
45
46     while(WiFi.status()!=WL_CONNECTED){
47         delay(500);
48         Serial.print(".");
49     }
50     Serial.print("\nIP:");
51     Serial.println(WiFi.localIP());
52
53     server.on("/",handelRoot);
54     server.on("/sw",ledSwitch);
55     server.begin();
56 }
57
58
59 void loop() {
60     server.handleClient();
61 }
62

```

首先要引入 WiFi.h 跟 WebServer 這兩個函式庫，因為這個專案會需要連接 WIFI 跟建立一個網路伺服器。接著設定 WIFI 的地址跟密碼，定義一個整數變數 ledPin 等於 2，2 在 ESP32 裡代表內建的 LED 燈。handleRoot 函式中建立了一個簡單的 HTML 網頁，網頁中有兩個按鈕「開」跟「關」，用 JavaScript 中的 button 功能來回傳按鈕的狀態。ledSwitch 函式會根據按鈕的回傳值來控制 LED 燈的開跟關。setup 函式主要用來連接 WIFI 跟開啟伺服器。loop 負責開啟網頁。

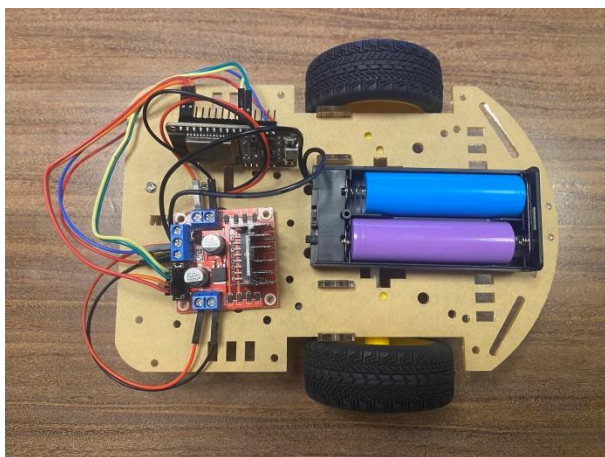


esp 32 遙控車

做完上面的 LED 小專案後我覺得可以把學到的東西重新組合一下變成一台遙控車。我的想法是我只要把原本的兩開關增加成五個開關，這五個開關分別代表前、後、左、右、停止這五個功能。



這次的小專案應該是我專題中最順利的地方，沒有遇到特別奇怪的問題，可以算是一次成功，花最多時間的地方是網頁的排版，因為寫網頁的時候會用到 HTML 這個語言，HTML 我比較不熟悉，花了一些時間來研究。



家裡剛好有車子的零件可以直接拿來用，省了我多的時間。

遙控車的程式幾乎跟開關 LED 燈的程式一樣，所以我這邊只放連結，有興趣的人可以上去看一下。

程式在下方 QRcode 裡



組裝 3D 列印機

我在做這個專題之前就有接觸 3D 繪圖跟 3D 列印，在專題期間我覺得我原本的 3D 列印機的列印尺寸跟列印速度太慢了，用起來不太方便，所以想換一台列印尺寸大列印速度快的 3D 列印機，當時符合這兩個條件的主要有兩家「拓竹」跟「Voron」（最近好像又多了好幾家），拓竹的機器特色是簡單好用但價格高，Voron 是開源的 3D 列印機價格相對便宜很多，因為可以買零件會來自己組。

我最後選擇了 Voron 的 3D 列印機，因為便宜而且在組裝的過程中可以學到很多，像是電線要怎麼走比較順、看錯誤碼，跟很多有趣又痛苦的事。

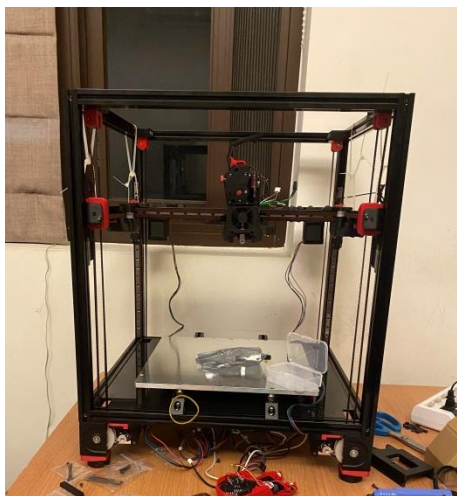
Voron 有幾個不同的型號，我選擇的是 Voron2.4。在 Voron 的官網上可以下載組裝說明書跟零件清單，因為零件一個一個自己買會花太多時間所以我選擇別人配好的零件包。



先把框架組起來，組裝框架比較簡單，只要注意對齊就可以了。



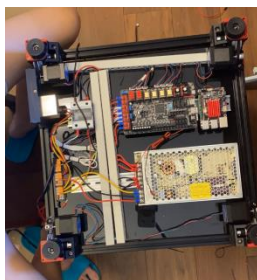
框架組完以後裝上馬達、皮帶跟其他會動的零件。



最後把線接好



整理前



整理後

線接好以後快樂的組裝時間就結束了，因為要通電測試系統跟零件了，這一步也是最難最痛苦的一步。

圖像辨識


圖像辨識我是使用 Python 加上 Opencv 跟 Mediapipe 兩個函式庫，我原本想用的語言是 C 或是 C++，但是用兩個語言來做圖像辨識太難了，而且網路上的資料也很少。Python 相對 C 系列的語言簡單很多，資料也相對比較好找，所以我決定手部辨識要用 Python。

Opencv 是 Open Source computer Vision Library 的縮寫，是一個跨平台的電腦視覺庫，主要用來處理與分析影像，是由英特爾公司發起並參與開發，以 BSD 授權條款授權發行，可以免費使用。

MediaPipe 是 Google Research 所開發的多媒體機器學習模型應用框架，支援 JavaScript、Python、C++ 等程式語言，可以簡單地實現手部追蹤、人臉檢測或物體檢測等功能。

Opencv 的基本功能

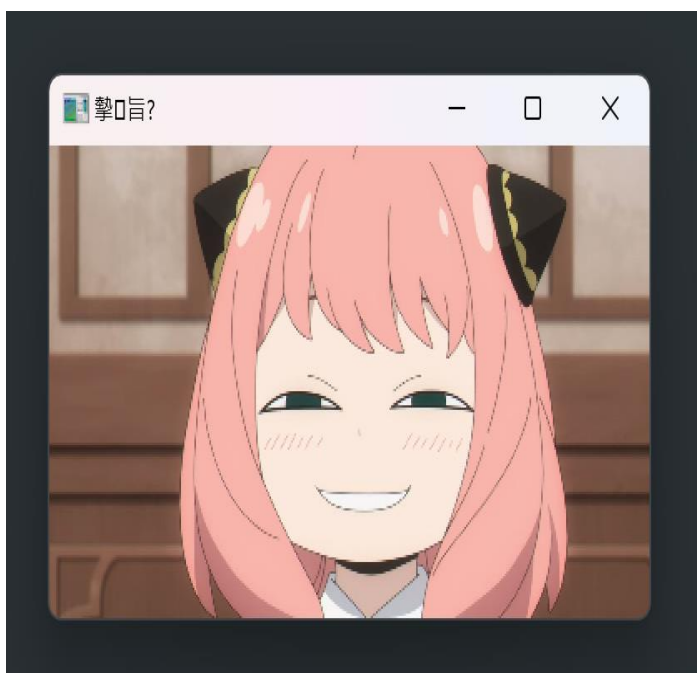
讀取照片



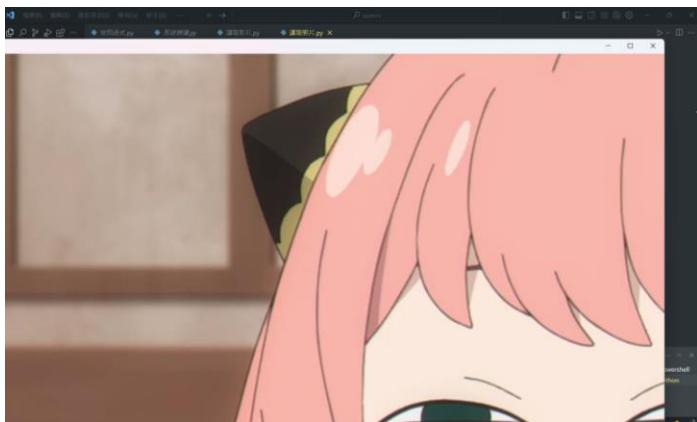
```
1 import cv2
2
3 abc = cv2.imread('bbbbb.jpg')
4 img=cv2.resize(abc, (0, 0),fx=0.5,fy=0.5)
5 cv2.imshow('安妮雅',img)
6 cv2.waitKey(0)
7
8
```

先用 `cv2.imread` 把照放到變數 `abc` 裡，用 `cv2.resize` 改變照片的大小，把改好尺寸的照片放到變數 `img` 裡，用 `cv2.imshow` 把照片顯示出來，`cv2.waitKey(0)` 是播放照片直到空白鍵被按下。

要注意的地方是第五行 `imshow` 後面的文字最後不要像我一樣放中文，因為編碼的問題，可能會出現亂碼，命名用英文跟數字會較好，可以減少很多不必要的的問題。



如果不調整照片的話，大小會是原始的大小，常常會超過螢幕或是太小。



把照片縮小一半以後就可以完整的出現在畫面裡了。



讀取影片

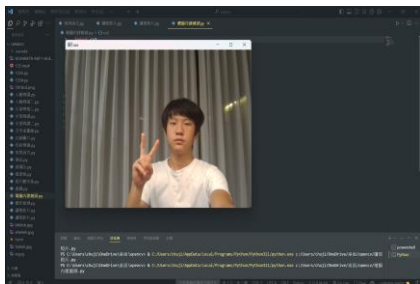
```
1  import cv2
2  abc=cv2.VideoCapture('123.mp4')
3
4  while True:
5
6      ret, fame =abc.read()
7      if ret:
8
9          cv2.imshow('aaa',fame)
10     else:
11         break
12     if cv2.waitKey(100)== ord('q'):
13         break
14
```

讀取影片的方式跟照片差不多，要把影片一幀一幀的放出來。先用 `cv2.VideoCapture` 讀取影片，然後判斷還沒有圖片，如果又就顯示出來，沒有就結束程式，`cv2.waitKey` 可以用來調整幀數。影片可能也要調整大小，調整的方式跟照片一樣。

讀取鏡頭

```
1 import cv2
2 abc=cv2.VideoCapture(0)
3
4 while True:
5
6     ret, fame =abc.read()
7     if ret:
8
9         cv2.imshow('aaa',fame)
10    else:
11        break
12    if cv2.waitKey(1 )== ord(' '):
13        break
14 print(abc)
15
```

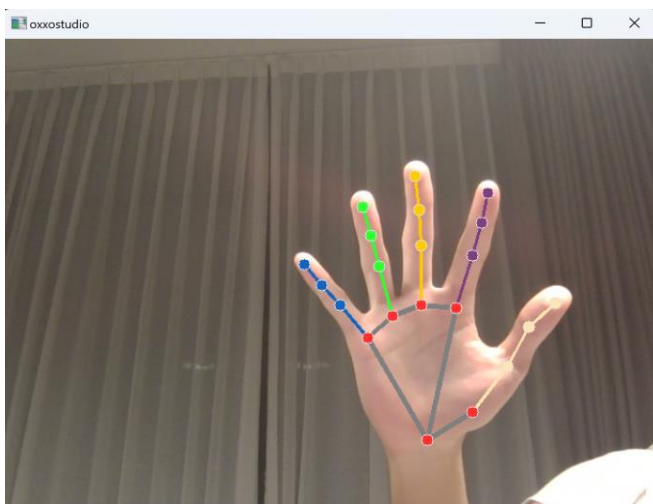
讀取鏡頭的方式跟讀取影片的放式幾乎完全一樣，只要把影片的檔案改成 0 就可以了，0 代表的是電腦內建的相機，如果是外接相機可能是 1 或其他數字。



手勢辨識

這個程式會把手上的關節都標示出來，把這些點連線，測量這些關節的角度來判斷是什麼手勢。

首先要用 MediaPipe 內建的功能把手上的關節標示起來。



每一個點關節都會有自己的編號，計算出指定關節間的角度後，再找到角度對應的手勢，例如 1 就是食指伸直其他手指彎曲。

```

1 import cv2
2 import mediapipe as mp
3 import math
4
5 mp_drawing = mp.solutions.drawing_utils
6 mp_drawing_styles = mp.solutions.drawing_styles
7 mp_hands = mp.solutions.hands
8
9 def vector_2d_angle(v1, v2):
10     v1_x = v1[0]
11     v1_y = v1[1]
12     v2_x = v2[0]
13     v2_y = v2[1]
14     try:
15         angle_ = math.degrees(math.acos((v1_x*v2_x+v1_y*v2_y)/(((v1_x**2+v1_y**2)**0.5)*((v2_x**2+v2_y**2)**0.5))))
16     except:
17         angle_ = 180
18     return angle_
19
20 def hand_angle(hand):
21     angle_list = []
22     angle_ = vector_2d_angle(
23         ((int(hand[0][0]) - int(hand[2][0])), (int(hand[0][1]) - int(hand[2][1]))),
24         ((int(hand[3][0]) - int(hand[4][0])), (int(hand[3][1]) - int(hand[4][1]))))
25     )
26     angle_list.append(angle_)
27
28     angle_ = vector_2d_angle(
29         ((int(hand[0][0]) - int(hand[6][0])), (int(hand[0][1]) - int(hand[6][1]))),
30         ((int(hand[7][0]) - int(hand[8][0])), (int(hand[7][1]) - int(hand[8][1]))))
31     )
32     angle_list.append(angle_)
33
34     angle_ = vector_2d_angle(
35         ((int(hand[0][0]) - int(hand[10][0])), (int(hand[0][1]) - int(hand[10][1]))),
36         ((int(hand[11][0]) - int(hand[12][0])), (int(hand[11][1]) - int(hand[12][1]))))
37     )
38     angle_list.append(angle_)
39
40     angle_ = vector_2d_angle(
41         ((int(hand[0][0]) - int(hand[14][0])), (int(hand[0][1]) - int(hand[14][1]))),
42         ((int(hand[15][0]) - int(hand[16][0])), (int(hand[15][1]) - int(hand[16][1]))))
43     )
44     angle_list.append(angle_)
45
46     angle_ = vector_2d_angle(
47         ((int(hand[0][0]) - int(hand[18][0])), (int(hand[0][1]) - int(hand[18][1]))),
48         ((int(hand[19][0]) - int(hand[20][0])), (int(hand[19][1]) - int(hand[20][1]))))
49     )
50     angle_list.append(angle_)
51     return angle_list
52

```

```

93
94 def hand_pos(finger_angle):
95     f1 = finger_angle[0]
96     f2 = finger_angle[1]
97     f3 = finger_angle[2]
98     f4 = finger_angle[3]
99     f5 = finger_angle[4]
100
101
102     if f1>50 and f2>50 and f3>50 and f4>50 and f5>50:
103         return 'good'
104
105     elif f1>50 and f2>50 and f3>50 and f4>50 and f5>50:
106         return 'no!!!'
107     elif f1<50 and f2<50 and f3>50 and f4>50 and f5<50:
108         return 'ROCK!'
109     elif f1>50 and f2>50 and f3>50 and f4>50 and f5>50:
110         return '0'
111     elif f1>50 and f2>50 and f3>50 and f4>50 and f5<50:
112         return 'pink'
113     elif f1>50 and f2<50 and f3>50 and f4>50 and f5>50:
114         return '1'
115     elif f1>50 and f2<50 and f3<50 and f4>50 and f5>50:
116         return '2'
117     elif f1>50 and f2>50 and f3<50 and f4<50 and f5>50:
118         return 'ok'
119     elif f1<50 and f2>50 and f3<50 and f4<50 and f5<50:
120         return 'ok'
121     elif f1>50 and f2<50 and f3<50 and f4<50 and f5>50:
122         return '3'
123     elif f1>50 and f2<50 and f3<50 and f4<50 and f5<50:
124         return '4'
125     elif f1<50 and f2<50 and f3<50 and f4<50 and f5<50:
126         return '5'
127     elif f1<50 and f2>50 and f3>50 and f4>50 and f5<50:
128         return '6'
129     elif f1<50 and f2<50 and f3>50 and f4>50 and f5>50:
130         return '7'
131     elif f1<50 and f2<50 and f3>50 and f4>50 and f5>50:
132         return '8'
133     elif f1<50 and f2<50 and f3<50 and f4<50 and f5>50:
134         return '9'
135     else:
136         return ''
137
138 cap = cv2.VideoCapture(0)
139
140
141
142 with mp_hands.Hands(
143     model_complexity=0,
144     min_detection_confidence=0.5,
145     min_tracking_confidence=0.5) as hands:
146
147     if not cap.isOpened():
148         print("Cannot open camera")
149         exit()
150     w, h = 540, 310
151     while True:
152         ret, img = cap.read()
153         img = cv2.resize(img, (w,h))
154         if not ret:
155             print("Cannot receive frame")
156             break
157         img2 = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
158         results = hands.process(img2)
159         if results.multi_hand_landmarks:
160             for hand_landmarks in results.multi_hand_landmarks:
161                 finger_points = []
162                 for i in hand_landmarks.landmark:
163                     x = i.x*w
164                     y = i.y*h
165                     finger_points.append((x,y))
166                 if finger_points:
167                     finger_angle = hand_angle(finger_points)
168
169                 print(hand_pos(finger_angle))
170             cv2.imshow('Oxoxostudio', img)
171             if cv2.waitKey(5) == ord('q'):
172                 break
173     cap.release()
174     cv2.destroyAllWindows()

```


圖像辨識結合遙控車

我想要把手勢辨識的程式放到開發板裡，然後接上馬達跟相機，就可以做出可以用手勢控制的遙控車了。

計畫一 ESP 32 cam

開發板的選擇

因為之前的寫好的手勢辨識程式是 Python 的，我只能選擇支援 Python 的開發板，所以我先從 Arduino 系列中開始找，然後我找到了 ESP 32 cam，ESP 32 cam 是有相機的 ESP 32，它同時支援 Python 又有相機，當時我覺得 ESP 32 cam 超完美。

問題的開始

要把 Python 的程式放到 ESP 32 cam 裡不能使用 Arduino 的 ide，因為 Arduino 的 IDE 支援 C 語言，所以要其他的 IDE，我選擇的是 Thonny，因為除了 Thonny 以外幾乎找不到其他可以用的。

要使用 Python 控制 ESP 32 就必須安裝 MicroPython，影像辨識需要遇到 Opencv，Opencv 沒有辦法單獨安裝，在安裝前需要先安裝其他的模組，其中一個叫做 Pycrypto 的模組無法跟 MicroPython 同時存在，所以我沒有辦法把 Opencv 安裝到 ESP32 cam。

要解決這個問題有幾種方法：

- 1、嘗試其他的開發版：之後有試過樹莓派板。
- 2、改用 C++：C++的手部辨識對我來說太複雜了，而且中文的資料太少了。
- 3、把影像傳到電腦，電腦辨識完，再把指令傳到開發板裡：我有找到有寫一個網頁，把影像傳到網頁上去辨識，後來發現這個方案只能把手的關節標示出來，沒有辦法把結果傳回開發板。
- 4、把 Opencv 的裡的函數自己寫出來：我有聽說有人做出來了，但是我找不到資料。

計畫二 樹莓派

有一天我在網路上看到有人把 Opencv 裝進樹莓派來做手部辨識，我就上網了解一下資料發現用樹莓派來做手勢控制的遙控車應該沒問題非常的適合。為了防止網路上的資料不完整我特地先買了一本樹莓派跟 Python 的書，為了避免書太舊跟現況不符合我特別找了 2023 年 6 月印刷的書，仔細的看過書以後我覺得完全沒問題一定會成功，我就根據書上的建議買了樹莓派 4B 的 4GB 版本，接下來就是問題的開始。

遇到的問題

1. 軟體版本跟書上的版本不同，其實我覺得我很可憐，買的書才出廠兩個月就過期了。這個軟體的更新讓我產生了很多麻煩的問題，這次的更新應改算是一個小更新，但是不少的預設跟指令都改掉改掉了。

2. 相機無法使用

我在確定相機可不可以使用的時候遇到了一連串問題，第一個問題是拍照的指令改掉了，從原本的 `raspistill -o test.jpg` 改成了 `libcamera - jpeg - test.jpg`。跟舊版的相機是預設關閉需要手動開啟，新

版相機是預設開啟，所以把開關相機的指令刪除了。這兩個問題讓我煩惱了很久，因為書上說要先開啟鏡頭，我找不到開啟鏡頭的地方，我覺得可能是這版不用自己開啟，所以我直接使用了拍照的指令，但是樹莓派找不到這個指令，我就想會不會是我打錯了，我又按照書上過期的指令再試了好幾次，結果還是不行，我想不按照書上的方法打開鏡頭果然是不行的，我又再嘗試了好幾次，軟體也重裝了好幾次，結果還是不行。所以我上網查了資料，網路上部份的資料也是用舊版的軟體，查到的步驟跟書上完全一樣，我覺得是我操作的問題，又嘗試了好幾次，直到我發現可能問題是出在指令上，我查了這個指令發現這個指令已經被刪除了，換成新的指令後一次就成功了。

我從這個烏龍中學到一件事情，當失敗很多次後可以想想問題有沒有可能是出在自己身上。

3. Opencv 無法安裝

在安裝 Opencv 之前需要先安裝一些相關的套件，我按照書上安裝了很多套件，其中一個叫做 libatlas4 的套件沒有辦法安裝，可能是軟體改版後不支援了。網路上的方法都跟書上的大同小異都需要用到無法安裝的套件，在嘗試了無數個方法後，找到了一個一鍵安裝的方法，這個方式是用連結到 Github 下載檔案，很神奇的只有這個方法不會用到 libatlas4。

4. Opencv 無使用鏡頭

安裝完 Opencv 以後我發現沒有辦法讀取鏡頭，從錯誤碼可以看出來問題可能是出在畫質或是記憶體不足，我先從比較容易檢查的記憶體開始解決，我先查了還有多少記憶體沒有被使用，一共使了 0.2GB 還剩下 3.8GB 沒被使用，一般來說 3.8GB 是完全夠用了，以防萬一從 SD 卡裡拿了 2GB 來當記憶體（swap），所以可用的記憶體就有 6.8GB 了，應該可以排除是記憶體不足的問題了。處理完記憶體後基本上就可以確定問題是出在畫質上，從錯誤碼上看不出來是無法讀取鏡頭還是無法顯示讀取到的影像，我先從無法顯示讀取到的影像開始解決，我的解決方

式是在程式直接調整再輸出，我試了各種常見的畫質後問題還是存在。所以現在可以確定問題是因為畫質不支援導致 OpenCV 無法讀取鏡頭，我查了一下發現也有人遇到相同的問題，他的解決方式是自己寫一個程式來讀取影像然後改變畫質，再把這個程式當作一個函數來使用，我把程式寫完之後發現了一個神奇的問題，錯誤碼說 PiCamera 未被定義（PiCamera 是一個函數），但是在終端機可以找到這個函數，這代表這個函數是存在的，我也重新安裝了這個模組，問題還在。

結論與感謝

雖然我的專題到目前為止尚未完成最初預設的目標「圖像辨識結合遙控車」，但我覺得我很棒了！因為我遇到任何問題我都有盡全力地去解決，我努力地去查找資料跟尋求協助，把我知道的解決方法都試了一次，而不是遇到問題就放棄。

人因夢想而偉大，有了最初的夢想，從國中專題「製作一個遙控汽車」到高中專題「影像辨識的遙控車（圖像辨識結合遙控車）」，這中間激發起我學習的動力，我明白築夢需要踏實才能一步步地朝我的夢想前進，這當中我學習了 3D 繪圖，獨立組裝一台 3D 印表機，程式語言從完全不了解到可以使用 C 語言跟 Python 去做一些小的專案，在專題的最尾聲也接觸到了 Linux 系統。

另外，我發現了一個大道理，只要你可以想到或是中文可以查到的都不能算是問題，真正的問題會藏在想不到的地方，根據我的一個小經驗，解決問題的小偏方是放綠色乖乖，雖然很玄，但真的有用。

雖然高中專題結束了，但我依舊會朝著我的夢想前進，繼續加強及學習我所需要的知識及技能，完成我的夢想「做出可以幫我拿東西的機器手臂」。

最後，我要感謝我的專題指導老師 賴奕綾老師、技術指導 張凌倩老師、特教老師 吳靜宜老師，謝謝她們的指導及鼓勵，也要感謝我的父母，謝謝他們支持我，更要謝謝我自己，堅持完成我的夢想。

附錄

我把這次專題裡做的練習跟專案搭部分都整理在下面的 QR code 裡有興趣的人可以去看一下，我整理了很久，裡面也幾個小遊戲可以下載來玩。



<https://github.com/jin-kuan/high-school-project>

參考資料

網路資料：

STEAM 教育學習 <https://steam.oxxostudio.tw>

CodeShiba 程式柴

<https://www.youtube.com/@CodeShiba>

GrandmaCan -我阿嬤都會

<https://www.youtube.com/@GrandmaCan>

桩桩計算

<https://space.bilibili.com/430489460>

Voron

<https://vorondesign.com>

Medium

<https://medium.com>

維基百科

<https://zh.wikipedia.org>

ChatGPT

<https://chatgpt.com>

CSDN 博客

<https://www.csdn.net>

樹莓派

<https://www.raspberrypi.com>

高中生程式解題系統

<https://zerojudge.tw>

書本資料：

大槻兼資（2024）。鍛鍊問題解決力！演算法與資料結構應用全圖解。臉譜出版

趙英傑（2022）。超圖解 C 語言 - 用資料結構x演算法突破 APCS。旗標科技股份有限公司

陳會安（2024）。RaspberryPi 數梅派—PythonxAI
超應用聖經。旗標科技股份有限公司

洪錦魁（2023）。史上最強 Python 入門邁向頂尖高
手之路王者歸來。深智數位股份有限公司

洪錦魁（2020）。Python 邁向領航者之路：超零
基礎。深智數位股份有限公司

趙英傑（2016）。超圖解物聯網 IoT 實作入門：
使用 JavaScript／Node.JS／Arduino／
Raspberry Pi／ESP8266／Espruino。旗標出版
股份有限公司