高雄市立新莊高級中學



自主學習書面成果報告書

主題:紅外線分段計時器

1年7班24號 梁家熏

組員	製作分工	耗時
梁家熏	Arduino程式設計	六個月:假日製作
陳恩言	外殼設計	

目錄

壹	、前言及動機4
貳	、成品說明4
	製作目的
	基本運作邏輯
	實際使用7
參	、程式設計、架構設計、演算法和分項實作「重點摘要列表」7
肆	、實作詳細說明
	一、紅外線感應及光敏電阻8
	紅外線感應
	紅外線感應的困境9
	光敏電阻9
	線路圖10
	程式碼10
	程式說明11
	二、millis()函式12
	程式碼12
	程式說明12
	1.millis 基本概念12
	2.運算邏輯
	3. 數值換算
	三、Lcd 面板13
	線路圖
	程式碼
	程式說明
	貫際網不圖14 リーマに定定に使用時期的羽
	LCO
	四:監才週刊
	線路圖15
	程式碼16
	1. 正常傳輸16

2.AT Command 模式	16
程式說明	.17
1.藍芽基本概念	.17
2.AT Command 模式	.17
3.正常傳輸模式	.17
五、計畫架構概念、I2C 雙向通訊	. 18
計畫架構概念	18
I2C雙向通訊說明	. 18
1. 程式碼	.18
主端(master)	.18
從端(slave)	19
2.I2C 基本概念	19
3.接線	.19
4. 程式說明	20
六、系統整合	. 21
整合架構說明	21
程式碼 主端(master)	. 21
程式說明 主端(master)	24
程式碼 從端(slave)	. 26
程式說明 從端(slave)	27
美化包裝及電源、傳輸接線和零件配置	28
伍、心得感想與困境	29
一、 學習 Arduino 開發設計,覺得困難之處?	. 29
1. Arduino 開發的困境	. 29
2.查找資料的困境	. 30
二、 如何克服困難、跨越障礙?	. 30
三、目前尚無法克服之困難	. 30
四、自主學習 Arduino 開發設計的收穫?	. 30
五、學習之路的未來的期許	. 31
陸、參考資料連結	31
柒、完整程式碼連結	31
捌、自主學習計畫書及在校學習紀錄表	32

壹、前言及動機

我在國中畢業後的暑假想要自己做個分段計時器,因為可以得知自己 的練習狀態用以檢討訓練內容是否有效,但市面上的紅外線分段計時器 又要價10幾萬。所以開始在YouTube上開始自學程式,觀看彭彭的課程學 習python,學習到了基礎的語法和邏輯觀念。在無意間初次發現了 Arduino開發版,上網找資料自學,到了高中資訊課學習到了C++,發現 邏輯都是相通,而且Arduino的程式架構就是C語言,把通盤觀念再學得 更清楚,並且再加深學習,應用在實際層面。以下是我的紅外線分段計 時器實作。

貳、成品說明

▶ 製作目的

我本身是位田徑運動員,為了要提升運動表現,透過量化數據(跑步 秒數)了解自身訓練狀況以檢討課表的功效,但是手動計時又會有不小的 誤差,所以就提出了紅外線分段計時器的想法,用以精準計時。

▶ 基本運作邏輯

使用手機連接藍芽,傳送訊號給主端,進行發令(計時開始)。經過計時點時,雷射光被遮擋,接收器偵測到光線變暗,發出訊號給主機(主端 master)記錄下這時的時間點,並透過藍芽回傳資料給手機並顯示秒數。



接收器

紅雷射光



手機秒數顯示(單人模式)

手機秒數顯示(雙人模式)

主機(主端 master)



▶ 實際使用

1. 使用介紹

https://youtu.be/e-c3xRkvdRs

2. 單人模式(實測)

https://youtu.be/otYiGtN_JYA

3. 雙人模式(實測)

https://youtu.be/35Uwpzg9XyA

參、 程式設計、架構設計、演算法和分項實作「重點摘要列表」

分項重點	此項 的應用層面	問題研究發現或困境
1. 紅外線感應	感應人體經過計 時點,並且發送 訊號至主端。	起初構想用紅外線,上網查了很多 資料,但是一直無法試成功,所以 改用光敏電阻,利用可見光的光阻 斷原理來判斷是否有人經過。
2. 光敏電阻	利用光阻斷原理 來感應是否通過 計時點。	利用演算法運算亮度數值,直到光 源穩定,得到定值,人體經過光源 變暗,數值變低,記錄此時間點, 達到計時功能。
3.millis() 函式	用於程式內背景 計時,不受其他 程式碼影響。	Arduino開發版中使用millis()時, delay()函式和任何使用「中斷」功 能來執行的程式皆無法運行。

4.LCD面板	顯示計時秒數, 讓使用者得知。	發現顯示其實佔了運算效能很大一 部份,為了不影響計時,所以不能 持續顯示秒數(loop每跑一圈就顯示 一次),只在通過計時點時顯示(特 定條件下顯示)。
5. 藍芽通訊	方便使用者用手 機就能發令及得 知秒數	需先用AT 指令完成設定,安卓手機 相較於iphone比較能穩定連接,連 接後的回傳值為 -1。
 6.計畫架構概 念 	開發過程的順利 與否,及失敗了 的B計畫。	必須先規劃好從何做起及替待方 案,否則可能會重頭來過,要變更 的設計會更多。
 7. 系統整合、 演算法 	將所有分項結合 為一個系統,並 加入演算法。	統一變數及規劃觸發條件,將程式 中的bug除掉,封包程式以方便呼叫 及除錯。

肆、實作詳細說明

一、紅外線感應及光敏電阻

▶ 紅外線感應

我的感應設計為對射式的,也就是一邊發射訊號,另一邊接收訊號, 而兩邊之間只要有人經過就會阻擋住訊號,接收器收不到訊號即代表有 人經過(圖一)。



▶ 紅外線感應的困境

紅外線接收器使用時都正常,可以得知電視遙控器上的按鍵所對應的 十進位數值,而發射器使用 tone()函式來下指令,模組上的燈在發射時 也有發亮。持續讓發射器輸出紅外線訊號,接收器也正常運行,但是接 收器卻沒有一直顯示訊號有接收到,更奇怪的是在其兩點之間以步行速 度經過後,馬上顯示有訊號,以跑步速度經過卻又沒有顯示有訊號。在 這令我百思不得其解的狀況下,改變作法,換成使用光敏電阻。

▶ 光敏電阻

運作原理也跟紅外線感應一樣,使用訊號阻斷(光阻斷),只是改成可 見光而已,接收器為光敏電阻,發射器為紅色雷射光。光敏電阻持續接 受光源照射,測得當前亮度數值,有人經過時,光線被阻擋,亮度下 降、數值下降,以此判斷是否有人通過。



VCC ➡ 5V GND ➡ GND SIG ➡ 14 號腳

(圖二)

▶ 程式碼

```
int value=0,settinglight=600,count=0,a=0; //設定變數
int beforeState=1;
int light[3]={0},i=0,value1=0;
void setup() {
    pinMode(13,OUTPUT); //設定腳位輸出或輸入
    pinMode(4,OUTPUT);
    pinMode(14,INPUT);
    Serial.begin(9600); //開啟序列鋪,Baud rate 鮑率為9600,每秒鐘能傳輸9600個位元的資料
void loop() {
    value=analogRead(14); //類比讀取14號腳位數值,光敏電阻訊號
    digitalWrite(4,0); //數位4號腳,低電位
```

```
Serial.print(value); //序列監控視窗顯示當前亮度數值,為測試用
 Serial.print(" ");
                        //序列監控視窗顯示咸應次數,測試用
 Serial.println(count);
 if (value>settinglight&&beforeState==-1) {
   digitalWrite(4,0);
                          //數位4號腳,低電位
   digitalWrite(13,LOW);
                          //數位13號腳,低電位,燈暗
   beforeState=beforeState*-1; //將beforeState的值乘以-1
                          //序列監控視窗顯示跑到a迴圈,測試用
   Serial.println("a");
}
 else if (value <= settinglight&&beforeState == 1) {
                          //數位4號腳,高電位
   digitalWrite(4,1);
   digitalWrite(13,HIGH);
                         //數位13號腳,高電位,燈亮
   count+=1;
                          //計算咸應次數,測試用
   beforeState=beforeState*-1; //將beforeState的值乘以-1
   Serial.println("b"); //序列監控視窗顯示跑到b迴圈 測試用
   delay(10);
                           //延遲10毫秒,讓主端能偵測到此訊號
}
 else if(value<=settinglight) {</pre>
   digitalWrite(4,0); //數位4號腳,低電位
   digitalWrite(13, HIGH); //數位13號腳,高電位,燈亮
   Serial.println("c"); //序列監控視窗顯示跑到c迴圈 測試用
}
 else if(value>settinglight) {
   digitalWrite(4,0); //數位4號腳,低電位
   digitalWrite(13,LOW); //數位13號腳,低電位,燈暗
   Serial.println("d"); //序列監控視窗顯示跑到d迴圈 測試用
}
}
```

▶ 程式說明

以下的a、b、c、d 迴圈為上述程式碼註解所示

使用 value 儲存光敏電阻的亮度數值, settinglight 為判斷是否有 人經過的數值標準。當設備都準備完成,開始運作,會先跑進 d 迴圈, 直到有人經過,遮擋光線,跑進 b 迴圈,13 號腳的燈亮代表光線正在被 阻擋,4 號腳發出訊號給主端,延遲 10 毫秒是為了讓主端能夠有足夠時 間來接收訊號,否則主端可能就不會成功接收到訊號。此時為了避免人 體在遮擋過程中重複感應,所以藉由 beforeState=beforeState*-1 來改 變 beforeState 的值,也就是只會跑進 b 迴圈一次就會跳出來一直跑進 c 迴圈,直到光線沒被阻擋,跳進 a 迴圈,再次改變 beforeState 的值, 回到可再進行感應的狀態,如此重複。

二、millis()函式

▶ 程式碼

▶ 程式說明

1.millis 基本概念

millis 是使用 Arduino 的中斷功能來進行計時,而其他一樣使用中 斷功能來進行運算的程式就無法正常執行,例如: interrupt、delay 等。millis 函式會每秒回傳一千個數值,也就是每回傳1個數值就等於 1毫秒,millis 函式是會從開機時就不斷地遞增直到 4294967295 毫秒, 也就是 49.71027 天就會 overflow(溢出)、歸零。

2. 運算邏輯

millis 是一直持續跑的時間軸,所以必須紀錄下兩個時間點「開始」(startTime)和通過「計時點」(CheckTime),計算其時間差 (runningTime)。而當有多個「計時點」,就須計算每個「計時點」與 「開始」(startTime)的時間差,進而得知不同人從「開始」到經過「計 時點」的秒數。



3. 數值換算

millis 回傳的數值會非常大,所以變數需使用「unsigned long」來 儲存,millis 會每秒回傳 1000 個數值,也就是1分鐘 60000 毫秒,1秒 1000 毫秒,以此類推。而為了之後的方便及正常顯示,將每一位數(個 位、十位、小數點後一二位)精準的算出來,而不是只算出秒、分鐘、毫 秒。

三、LCD 面板

▶ 線路圖





▶ 程式碼

```
void lcd demonstration() {
                         //淨空面板
      lcd.clear();
      lcd.setCursor(1, 1); //在第二行,第二列,
      lcd.print(minTime); //顯示 minTime的數值
      lcd.setCursor(2, 1); //在第三行,第二列
      lcd.print(":"); //顯示「:」
      lcd.setCursor(3, 1); //以下以此類推
      lcd.print(tensecTime);
      lcd.setCursor(4, 1);
      lcd.print(secTime);
      lcd.setCursor(5, 1);
      lcd.print(".");
      lcd.setCursor(6, 1);
      lcd.print(msecTime);
      lcd.setCursor(7, 1);
      lcd.print(mmsecTime);
      lcd.setCursor(9, 1);
      lcd.print(analogNumber); //顯示傳入類比數值,測試用
```

}

▶ 程式說明

面板主要用來顯示秒數,也使用到二維陣列的概念。載入模組後,在 setup()做一些初始化的設定,再將顯示的格式固定,封包在 lcd_demonstration()函式中,方便呼叫。

▶ 實際顯示圖



▶ Lcd 面板實作的困境與學習

一開始並沒有將每一位數(個位、十位、小數點後一二位)精準的算出 來,而是只算出秒、分鐘、毫秒。秒有時只有一位數,有時又是兩位 數,而如果只固定秒、分鐘、毫秒的位置,就會造成秒在兩位數時正常 顯示(圖四),一位數時,原本應該顯示在個位數,卻顯示在十位秒數, 而個位數是空白的(圖五)。

1	:	1	2	•	9	9
分		秒	鐘		毫	秒
鐘						
1		2			9	9

鐘

秒

(圖四) 1分12秒99

(圖五) 1分2秒99

為了避免這種狀況,上述的 millis()函式說明中,將所有位數精準的算 出,並且固定其位置,就能解決此種問題了。

秒

毫

四、藍芽通訊

分

鐘

▶ 線路圖



(圖六)

▶ 程式碼

1. 正常傳輸

```
#include <SoftwareSerial.h>
SoftwareSerial BTSerial(10, 11); //RX:數位10號腳,TX:數位11號腳
void setup() {
 BTSerial.begin(9600);//開啟藍芽序列鋪,與連線裝置的鮑率為9600,每秒傳輸9600個位元的資料
 pinMode(9, OUTPUT); //數位9號腳為輸出
 digitalWrite(9,LOW);//9號腳低電位,進入訊號傳送模式
void bluetooth demonstration() {
     BTSerial.print(minTime); //顯示 minTime的數值
                              //顯示「:」
     BTSerial.print(":");
     BTSerial.print(tensecTime);//以下以此類推
     BTSerial.print(secTime);
     BTSerial.print(".");
     BTSerial.print(msecTime);
     BTSerial.print(mmsecTime);
     BTSerial.print(" ");
     BTSerial.println(analogNumber);//顯示傳入類比數值,並換行,測試用
}
```

2. AT Command 模式

```
#include <SoftwareSerial.h>
SoftwareSerial BTSerial(10, 11);//RX:數位10號腳,TX:數位11號腳
void setup()
{
 pinMode(9, OUTPUT); //數位9號腳為輸出
 digitalWrite(9,HIGH); //9號腳低電位,進入AT模式
 Serial.begin(9600);
                     //設定鮑率為9600
 Serial.println("Enter AT commands:");
 BTSerial.begin(38400);//設定藍芽序列鋪鮑率為38400
}
void loop()
{
 if (BTSerial.available()) {
   Serial.write(BTSerial.read());
 }//顯示藍芽回傳數值
 if (Serial.available()) {
   BTSerial.write(Serial.read());
 1//傳送序列鋪上的訊號至藍芽裝置
}
```

▶ 程式說明

1. 藍芽基本概念

為了不干擾開發板與電腦的通訊,所以設定藍芽專用的 RX、TX 腳 位,不使用 uno 板預設的通訊腳位。在使用 HC-05 之前須先進入 AT Command 模式進行設定,而 EN(KEY) 如果是高電位就進入 AT Command 模 式,低電位則正常傳輸(圖七)。

註: 副廠的 HC-05 需先按住按鈕再接上 KEY(EN),才能進入 AT 模式。



(圖七)

2. AT Command 模式

進入 AT 模式後就可進行設定,需要特別注意的是 HC-05 出場時鮑率 預設值為 38400,所以藍芽鮑率也要改為 38400,才能執行設定。AT 模式 中可以更改及詢問相當多種設定,如: 鮑率、藍芽位置、藍芽名稱、密 碼、重啟、是否連至手機、初始化 SPP 函式庫、詢問周遭藍芽裝置、更 改為主動、被動等。依照出場說明書(指令集)使用特定格式進行設定及 詢問。下圖為藍芽位置詢問結果。(圖八)

手機藍芽位置及名稱	+INQ:0CD7:46:ADF5A8, 7A020C, FFC8, My iphone
手機藍芽位置及名稱	+INQ:AC37:43:AFA5C9, 5A020C, FFC5, GOD
HC-05 藍芽位置	+ADDR:0019:10:097C27

(圖八)

3. 正常傳輸模式

設定完成後,手機即可搭配 APP 主動連線,即可雙向通訊。基本上只 有 android 系統的手機才能正常使用, ios 的就不行,詳細原因我也不太 清楚。APP 一開始我打算用 MIT 開發的 app inventor 自己寫一個,但也 只能在 android 系統上使用,而且有點太複雜了。所以我找了一個最好 用的 APP(圖九),但也只能使用在 android 系統的手機。



五、計畫架構概念、I2C 雙向通訊

▶ 計畫架構概念

一開始設計將所有分項先做完再組裝成一個完整的系統,這個概念是 好的,但沒考慮到的是失敗要換方法時,後續的計畫會被打亂,可能連 前面做好的開發都要換掉、重做,才能繼續執行計畫。如下述 I2C 雙向 通訊都開發完了,但卻不太符合我需要的,所以不能用會造成一些悲 劇。

程式碼的部分,越複雜的系統越要有好的規劃,不然可能寫到某一 行,發現這樣行不通,又要把前面寫好的刪掉,重來一遍。而未來如果 有更大型的開發,刪掉的就不會只有幾行而已了,所以初期的架構設計 就會是個蠻重要的課題。

▶ I2C 雙向通訊說明

```
1. 程式碼 主端(master)
#include <Wire.h>
char c="0";
int a=0;
void setup() {
                    //開始I2C通訊
 Wire.begin();
 Serial.begin(9600); //開啟序列鋪, 鮑率為9600
 pinMode(13,OUTPUT); //數位13為輸出
}
void loop() {
 Wire.requestFrom(8, 1); //向8號要求1 byte的資料
 while (Wire.available()) { / / 如果有收到資料
   char c = Wire.read(); //讀取並儲存為字串
   Serial.print(c); //印出char c
   a = c - 48;
                    //char 1=int 49
   Serial.print(a); //印出a
                          //如果a=1
      if(a==1){
       digitalWrite(13,1);//13號腳亮燈
      }
                          //其他
     else {
       digitalWrite(13,0);//13號腳燈暗
      1
  }
```

從端(slave)

```
#include <Wire.h>
int bottom=0;
void setup() {
                           //數位13為輸出
 pinMode(13,OUTPUT);
 pinMode(7,INPUT);
                           //數位7為輸入
 Wire.begin(8);
                           //設定I2C位置為8號
 Wire.onRequest(requestEvent);//設定為事件
}
void loop() {
}
void requestEvent() {
 bottom=digitalRead(7);//數位讀取7號腳
 if(bottom>0){ //如果bottom是高電位,按鈕按下
   Wire.write("1"); //傳送"1"
   digitalWrite(13,1); //13號腳亮燈
 }
 else if (bottom<=0) { //如果bottom是低電位,按鉛沒按
                     //傳送"2"
   Wire.write("2");
   digitalWrite(13,LOW);//13號腳燈暗
 }
}
```

2. I2C 基本概念

I2C 是用於開發板之間的溝通,分為「主端」和「從端」,大多數是一個主端和多個從端,也可以有多個主端,但是會變得很複雜。為了避免通訊錯誤,開發版地址需從7號開始。

3.接線

將每個 Arduino 的 SDA (analog pin 4)、SCL (analog pin 5)、 GND 和 5V 分別連接起來,另外在 SDA 和 SCL 上加個接到 5V 的 4.7K 歐姆的電阻(上拉電阻)以確保電壓為高電位(圖十)。



4. 程式說明

基本設定都完成後主端要發出要求,從端收到要求後才會回傳資料。 而我並不知道何時有人通過,要在何時發送要求,為符合需求,主端需 一直傳送請求至從端,導致會比較耗能。在有多個計時點(從端)且在多 人偵測時,會無法得知要向哪個從端發送要求,也會有一定的時間誤 差。所以目前 I2C 並無法使用在此系統中。

以 char 接收數據,但還要轉成 int 才比較好進行運算,不同的 char 對應到不同的十六進位數和十進位數(圖十),對照圖十以利進行換算。

Decimal	Hex	Char	Decimal	Hex	Char	JDecimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	0	96	60	•
1	1	[START OF HEADING]	33	21	1	65	41	Α	97	61	а
2	2	[START OF TEXT]	34	22		66	42	В	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	С	99	63	с
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	е
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	1.00	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	ĥ
9	9	[HORIZONTAL TAB]	41	29)	73	49	1.00	105	69	i i
10	Α	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	В	[VERTICAL TAB]	43	2B	+	75	4B	κ	107	6B	k
12	С	[FORM FEED]	44	2C	,	76	4C	L	108	6C	1
13	D	[CARRIAGE RETURN]	45	2D		77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	1.00	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	1	79	4F	0	111	6F	0
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	Р	112	70	р
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	S
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	т	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	w	119	77	w
24	18	[CANCEL]	56	38	8	88	58	Х	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	У
26	1A	[SUBSTITUTE]	58	3A	1.00	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	Λ	124	7C	1
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	1	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

ASCII TABLE

(<u>圖十一</u>)

六、系統整合

▶ 整合架構說明

發令的意思為使用手機傳輸任意數值至主端,計時開始

計時點偵測到有人經過時會發送訊號(5V的數位訊號),而每個計時 點會經過不同的電阻造成訊號電壓的改變,主端就會偵測到訊號並以類 比讀取來判斷此訊號從哪一計時點而來,以記錄時間點。

以手機連接主端進行發令(開始)(傳送1至主端),即開始計時,主端 運算完秒數後就會顯示於手機畫面,想要停止計時,再發令一次即可(傳 送1至主端)。

➢ 程式碼 主端(master)

以下為最終完整程式碼

```
const int buttonPin = 12; //開啟變數
const int ledPin = 13;
int buttonState = 0,group=0,group1=0,group2=0,p=0;
#include <SoftwareSerial.h> //載入函式
SoftwareSerial BTSerial(10, 11); //RX:數位10號腳,TX:數位11號腳
```

```
int beforeState=0,i=2,gunState=0,mood=0;//開啟變數
```

```
unsigned long startTime=0,a[10];
unsigned long checkTime=0, =0;
```

```
unsigned long runningTime=0;
```

```
unsigned long minTime=0;
```

```
unsigned long secTime=0,msecTime=0,tensecTime=0,device_number=2;
int gunNumber=0,analogNumber=0,sensorState=0,b=0;
```

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);//設定 lcd(I2C address,行數,列數)
```

```
void setup() {
  BTSerial.begin(9600);//開啟藍芽序列鋪,與連線裝置的鮑率為9600,每秒傳輸9600個位元的資料
  lcd.init(); //初始化lcd
  lcd.backlight(); //開啟背光
  pinMode(ledPin, OUTPUT); //設定腳位輸入、輸出
  pinMode(buttonPin, INPUT);
  pinMode(5, OUTPUT); //設位9號腳為輸出
  digitalWrite(9,LOW);//9號腳低電位,進入訊號傳送模式
  Serial.begin(9600);//開啟序列鋪,Baud rate 鮑率為9600,每秒鐘能傳輸9600個位元的資料
  }
```

```
void loop() {
 gunState=int(BTSerial.read());//讀取藍芽回傳值
 if (gunState!= -1) { //判斷回傳值是否不等於 -1
   b=1;
         //是,b=1
  }
 else{
   b=0; //否,b=0
  }
while (b==1) { //如果B=1
  gunNumber+=1;
  //lcd.clear(); //lcd版面清空
  //BTSerial.println(gunNumber);//測試用
  if(group1==0 && group2==0){//判斷是否執行
   startTime=millis();//紀錄此時時間
   tone (5,500,190); //5號腳、500hz、持續190毫秒
   BTSerial.println("Start");//藍芽顯示開始
   break; //強制跳出while迴圈
  }
 BTSerial.println("Finish");//藍芽顯示結束
 group1=0;
 group2=0;
 b=0;
}
if(gunNumber%2==1){ //判斷是否跳入request()
  request();
  //lcd.setCursor(0, 0); //測試用
  //lcd.print("*"); //測試用
}
//Serial.print(gunState); //數值測試用
//Serial.print(" ");
//Serial.print(gunNumber);
//Serial.print(" ");
//Serial.print("test
                      ");
//Serial.print(c);
//Serial.print(" analogNumber ");
```

}

```
void request() {
   analogNumber=analogRead(A1);//讀取回傳電阻值
   if(analogNumber<=200 && analogNumber>=101){ //藉由電阻值來判斷訊號從哪一裝置傳出//120
     checkTime = millis();//紀錄此時時間點
     BTSerial.print("device1 ");//藍芽顯示device1
     Time();//跳進Time函式
                                                    第一個裝置
     bluetooth demonstration();//藍芽顯示函式
                                                    (計時點一)
     //lcd demonstration();//跳進lcd顯示函式
     a[0]=checkTime;//儲存於陣列
     group1=group1+1; //用以幫助後續程式判斷是否跳進過此迴圈
   else if(analogNumber<=100 && analogNumber>=40){//藉由電阻值來判斷訊號從哪一裝置傳出//80
     checkTime = millis();//紀錄此時時間點
     BTSerial.print("device2 ");//藍芽顯示device2
     Time();//跳進Time函式
     bluetooth demonstration();//藍芽顯示函式
     //lcd demonstration();//跳進lcd顯示函式
     a[1]=checkTime;//儲存於陣列
                                                  A區程式碼
     group2=group2+1;//用以幫助後續程式判斷是否跳進過此迴圈
                  // ,和在一次週期内跳進過幾次
     if(group1==1 && group2==1){//如果各個裝置都只偵測到一個人
       gap();//跳進gap函式
     }
   else if (analogNumber>10) {//無法判斷訊號從哪一裝置傳出就進入此迴圈
     checkTime = millis();//紀錄此時時間點
     Time();//跳進Time函式
   }
}
void Time() {
                                                //計算出時間差
     runningTime=(checkTime-startTime);
     minTime=(runningTime/60000);
                                                // 換算分鐘
     tensecTime=(runningTime%60000)/10000;
                                                //換算第十位數的秒數
     secTime=((runningTime%60000)%10000)/1000;
                                                //換算個位數的秒數
     msecTime=((runningTime%60000)%1000)/100;
                                                // 換算小數點後第一位的秒數
     mmsecTime=(((runningTime%60000)%1000)%100)/10;//换算小數點後第二位的秒數
}
void gap() {
 BTSerial.println("gap"); //藍芽顯示gap
 for (p=0;p<device number-1;p++) { //根據裝置個數判斷要計算出幾個間隔秒差
     runningTime=(a[p+1]-a[p]);
                                //計算出時間差
                               //換算分鐘
     minTime=(runningTime/60000);
     tensecTime=(runningTime%60000)/10000;
                                         // 換算第十位數的秒數
     secTime=((runningTime%60000)%10000)/1000; //換算個位數的秒數
     msecTime=((runningTime%60000)%1000; //换算小數點後第一位的秒數
     mmsecTime=(((runningTime%60000)%1000)%100)/10;//換算小數點後第二位的秒數
     BTSerial.print("gap");//藍芽顯示gap
     BTSerial.print (p+1); //藍芽顯示第幾間隔
     BTSerial.print("
                          ");
     bluetooth demonstration(); //藍芽顯示函式
 }
}
```

```
void bluetooth demonstration() {
     BTSerial.print (minTime); //顯示 minTime的數值
     BTSerial.print(":");
                           //顯示「:」
     BTSerial.print(tensecTime);//以下以此類推
     BTSerial.print(secTime);
     BTSerial.print(".");
     BTSerial.print(msecTime);
     BTSerial.print (mmsecTime);
     BTSerial.print(" ");
     BTSerial.println(analogNumber);//顯示傳入類比數值,並換行,測試用
}
void lcd demonstration() {
                            //淨空面板
       lcd.clear();
       lcd.setCursor(1, 1); //在第二行, 第二列,
       lcd.print (minTime); //顯示 minTime的數值
       lcd.setCursor(2, 1); //在第三行, 第二列
       lcd.print(":");
                           //顯示「:」
       lcd.setCursor(3, 1); //以下以此類推
       lcd.print(tensecTime);
       lcd.setCursor(4, 1);
       lcd.print(secTime);
       lcd.setCursor(5, 1);
       lcd.print(".");
       lcd.setCursor(6, 1);
       lcd.print(msecTime);
       lcd.setCursor(7, 1);
       lcd.print(mmsecTime);
       lcd.setCursor(9, 1);
       lcd.print(analogNumber); //顯示傳入類比數值,測試用
```

}

▶ 程式說明 主端(master)

此段程式碼適用於發令、開始和結束,只要傳送任何東西,系統就 會偵測到並且開始計時或停止計時。

在藍芽實際測試時發現持續穩定連結且無傳輸資料時,會一直回傳 -1至主端。利用這點就可以判斷是否有傳送資訊,手機有傳輸b就為 1,進到下一個迴圈。while 迴圈裡 gunNumber 會加1,每發令一次 gunNumber 就會加1,一次會進入 request()進行計時,下一次則不會進 入 request(),周而復始的執行。 其中 group 值等於計時點的感應次數,一開始 group=0,進行發 令,會顯示 start 並發出嗶~~和跳出 while 迴圈。如果感應器有被觸 發,group 變得不等於零,再發令一次 group 會歸零並顯示 finish。



如果重頭到尾所有計時點都只被觸發一次,也就是只有一個人在使用(單 人模式),就會有間隔秒差(gap 函式),多人、雙人沒有,因為在每個計 時點的名次都不一定一樣,所以無法計算間隔秒差(gap 函式)。跳進 gap 函式後,會根據設定好的裝置個數來判斷要運算出幾個間隔秒差,然後 一一顯示。未來如果有增加感應裝置個數(計時點),上述A區程式碼需 要放置於最後一個裝置的迴圈裡,才能正確判斷。

▶ 程式碼 從端(slave) 以下為最終完整程式碼

```
int value=0, settinglight=600, count=0, a=0; //設定變數
int beforeState=1;
int light[3]={0}, i=0, value1=0;
void setup() {
 pinMode(13,OUTPUT); //設定腳位輸出或輸入
 pinMode(4,OUTPUT);
 pinMode(14, INPUT);
 Serial.begin(9600);//開啟序列鋪,Baud rate 鮑率為9600,每秒鐘能傳輸9600個位元的資料
 for(i=0;i<2;i++){ // 胞兩圈
   delay(200); //延遲0.2秒
   value=analogRead(14);//以類比訊號讀取14號腳
   light[i]=value;//以陣列儲存兩個數值
  }
   value1=light[0]-light[1]; //取兩數差值
   if((value1== 1)or(value1==0)or(value1==-1)){//判斷差值是否在容許值內
     settinglight=light[0]-10;//訂定光敏電阻亮度參考基準
   }
   else {//如果不符合,再次偵測數值,直到穩定
     set();//跳入set()函式
   }
}
void loop() {
 value=analogRead(14); //類比讀取14號腳位數值,光敏電阻訊號
                           //數位4號腳,低電位
 digitalWrite(4,0);
 /*Serial.print(settinglight);
 Serial.print(" ");
                      //序列監控視窗顯示當前亮度數值,為測試用
 Serial.print(value);
 Serial.print(" ");
 Serial.println(count); */ //序列監控視窗顯示感應次數,測試用
 if (value>settinglight&&beforeState==-1) {
                           //數位4號腳,低電位
   digitalWrite(4,0);
                           //數位13號腳,低電位,燈暗
   digitalWrite(13,LOW);
   beforeState=beforeState*-1; //將beforeState的值乘以-1
   Serial.println("a");
                           //序列監控視窗顯示跑到a迴圈,測試用
}
 else if(value<=settinglight&&beforeState==1) {</pre>
                           //數位4號腳,高電位
   digitalWrite(4,1);
   digitalWrite(13,HIGH);
                           //數位13號腳,高電位,燈亮
   count+=1;
                            //計算鳳應次數,測試用
   beforeState=beforeState*-1; //將beforeState的值乘以-1
   Serial.println("b"); //序列監控視窗顯示跑到b迴圈,測試用
   delay(10);
                            //延遲10毫秒,讓主端能偵測到此訊號
}
```

```
else if(value<=settinglight) {</pre>
   digitalWrite(4,0); //數位4號腳,低電位
   digitalWrite(13, HIGH); //數位13號腳,高電位,燈亮
   Serial.println("c"); //序列監控視窗顯示跑到c迴圈,測試用
}
 else if(value>settinglight) {
   digitalWrite(4,0); //數位4號腳,低電位
   digitalWrite(13,LOW); //數位13號腳,低電位,燈暗
   Serial.println("d"); //序列監控視窗顯示跑到d迴圈,測試用
}
}
void set() {
 for(i=0;i<2;i++){//跑兩圈
   delay(200); //延遲0.2秒
   value=analogRead(14);//以類比訊號讀取14號腳
   light[i]=value;//以陣列儲存兩個數值
  }
 value1=light[0]-light[1]; //取兩數差值
  if((value1== 1)or(value1==0)or(value1==-1)){//判斷差值是否在容許值內
     settinglight=light[0]-10;//訂定光敏電阻亮度參考基準
   }
   else {//如果不符合,再次偵測數值,直到穩定
     set();//跳入set()函式
   }
}
```

程式說明 從端(slave)

這裡用到的演算法概念,主要是在尋找出上述光敏電阻的比較、判斷 數值基準「settinglight」,而在不同地區的背景光源會不同,進而導 致感應靈敏度也會有所不同,所以必須偵測當時的背景光源,以達感應 穩定性。

先取兩個間隔 0.2 秒的亮度,相減後判斷是否在所設的容許值內,如 果是的話就找到了穩定的背景光源「light[0]」,否則就再取兩個背景 光源數值,如此重複,直到找到穩定數值為止。「light[0]」再去減 10 就是判斷數值基準「settinglight」,減的數字越小光敏電阻的感應就 越靈敏,反之如此。



▶ 美化包裝及電源、傳輸接線和零件配置

每條傳輸線使用一公一母的防呆接頭,並用端子將電線跟接頭合而為 一再塗上黏膠跟絕緣膠帶,這樣一來就不會接錯線也比較美觀方便使用





將電源線拉成正跟負,正接 uno 板的 5V 負接 Vin,並聯所有的裝置。

電源線

零件配置



主機跟感應裝置連接

感應裝置(正面)

伍、心得感想與困境

一、 學習 Arduino 開發設計,覺得困難之處?

1. Arduino開發的困境

我自己是無意間在網路上發現 Arduino,也是初次接觸到 Arduino 的 開發。Arduino 本身是 c 語言的架構,所以在程式上面比較沒有問題,最 困難的是要如何讓每個 Arduino 版都得到穩定的供電,不會衍生造成一 些錯誤,這裡卡了我足足兩個月。當程式越寫越大,到底是其中哪個環 節出現了問題,還是接線錯誤,都是要不斷測試,一個個的排除。 有時也會買到副廠的零件,廠商為了要節省成本會跟原廠的使用方式 會有些許的不同,而要在網路上找到副廠的使用資訊也相當的少。像是 副廠的 HC-05 就需先按住按鈕再接上 KEY(EN),才能進入 AT 模式。

2. 查找資料的困境

在網路上的中文資源其實並不算多,所以有時會需要參考到國外的資 料,而大部分都是英文,讀起來會有點吃力。網路上的資料有時也並不 完整,都需要再經過自己的統整、理解,才能學習到全面性的知識。

二、 如何克服困難、跨越障礙?

我認為克服困難最重要的就是找出原因再加以解決,而要找出原因就 需要不斷的思考、測試、查資料和觀察是環境造成感應器干擾還是程式 碼、電壓的問題,找出原因後就能加以解決了。而找資料時我會去一些 論壇上面看看有沒有人跟我遇到相同的困難,或者會留言問資料文章、 教學影片的作者,也會詢問學校老師,都有助於解決問題。

Arduino官網和一部分的資料都是使用英文,閱讀上雖有點難度,也 比較耗時,但其中的文法並不會很困難,透過自己理解文意和google翻 譯一些專有名詞,就能瞭解其所傳達的意涵。

三、目前尚無法克服之困難

目前我是以兩個計時點(從端)和一個主機(主端)總共三個Arduino開發版來做為系統的運作,這樣的配置,電壓還算穩定,但如果增加計時點(從端)就會導致電流不足而無法使用。

還有判斷訊號是從哪個計時點(從端)傳送,是藉由經過不同的電阻造 成電壓的不同來判斷的,所以需要先測量電壓大小再設定判斷值。而計 時點(從端)之間的電線長度就會導致電壓變小甚至訊號傳送不過來。這 些都是目前無法克服的困難。

目前的成品只能說堪用,但以產品製造、販售的角度來看是遠遠不及 格的,所以希望能夠就讀相關科系,製造出更好的產品。

未來可能可以往計時點(從端)的無線傳輸、獨立供電、繼電器的使用 或程式碼的改善解決上述問題,也創造更人性化的使用方式。

四、自主學習Arduino開發設計的收穫?

從Arduino開發中學到的不僅僅是程式和模組的應用,更重要的是學 到統整不同資料和理解使用方式及解決問題的能力。我的英文閱讀能力 也因為讀了很多英文文獻而進步。而解決問題的能力也是時常用到的, 這次的開發也更增進了我了解不同問題以進而解決。這些收穫都是在未 來升學和職涯路上會用到的。像是在科技產業資訊都是日新月異的,新 的技術、產品、文獻都在變革,唯有持續的自主學習才不會被淘汰。

五、學習之路的未來的期許

我希望以後能夠就讀電機、資工相關的校系,學習上述無法解決的難題,了解更多實際的應用。才不會像我因為初次接觸 Arduino,不了解開發分段計時器所需適合的模組,導致使用一些不太適合的方法、模組。

以後也想要結合我的田徑專長,投入運動科技這個領域,利用 AI 開發最適跑步運動模型,找到最適合的跑步動作、型態。(例如:臉書 AI 研究院所發表的 <u>Control Strategies for Physically Simulated</u> <u>Characters Performing Two-player Competitive Sports</u>)

或是利用多個攝影機拍攝建構 3D 畫面以利動作調整,還有使用各種 感測器和地面壓力感測,回饋數據,幫助提升運動表現,貢獻社會我的 所長。

陸、參考資料連結

熊的開發 DVPBear: 黄信惠的瘋狂教室 LANMOOTECH 佑來認真教 LazyTomato Lab 懶番茄工作室 Gensou Arduino 官網 【雙A計畫】藍牙模組(HC05、HC06)常見的指令使用教學 HC-05 蓝牙串口模块 AT 指令集 iCShop 傑森創工 Tinkercad 米羅科技 Arduino 液晶顯示屏控制 Arduino I2C LCD control 使用 Arduino 五分鐘 讓 LCD 顯示文字 HC-05 藍芽連線 JimmyHu Cooper Maa ASCII-Table-wide. svg 柒、完整程式碼連結

主端:<u>intergration master</u> 從端:<u>intergration slave</u>

高雄市立新莊高級中學 學生自主學習計畫申請書 申請人相關資料 共學同學 申請人 協助專家 法定代理人 導師 指導老師 (無則免填) (無訓安壇) (最多3人 (諸簽名) (請簽名) 姓名(請簽名) (諸答名) 开级 座號 陳思言 夏気を見 計量内容磁為本人振荡 東 24 F 1-7 N-D 教師顧乃 教師顧乃嘉 計畫概述 計畫內容說明 红外绵为段計時器 計畫主題 (條列式呈現、具體、扼要) (100字以内) 「工程 □數理化 標:實現計時功能,彌補人工 體資訊 1. 目 對應 □ 醫藥衛生 □生命科學 □生物資源 □建築與設計 十八學群 一地球與環境 □藝術 計時的誤差。 □大眾傳播 □社會與心理 □外語 機:公準確且有效率的方式帮助 (以圖呈現) 2. 動 □文史哲 □教育 □法政 (最多3項為限) □财经 ■进憩運動 □管理 理動人員。 □國文□英文□數學□自然□社會□藝能圖資訊 對應學科 3. 具體方式: (1)漆購红外網感應點以觸發計時 (Ardaino Uno 開發板)、一個非發射型、接後 (2) 宮電階程式以進行計時 □第一學期 申請學期 \ 學年度 **國**第二學期 (學期以■呈現) 6 週/共 [8 節 共 計畫期程 其他自學時間 (3)组合後週間並修改 (非在校時間) 利用LCD雨板霸武 所需設備場地 計畫成果 1. 红外雄鳳應器能夠觸發 2. 程式顺利 執行 預期成果 (保列式呈现) (能對應計重內容) 3. 昆页 式計 時物數 4 □書冊報告 □口頭報告 圖網路展示 □海報展示 □獎狀證書 成果發表形式 □作品展覽 ■動態表演 □其他: (可多選,以圖呈現) 審查結果 審查意見 評定結果 (委員可打勾,亦可書寫在空白處) □計畫周延詳實 □計畫空泛不具體 □細節解釋不清楚 □格式不符 日通過 □其他: 初審 「不通過 指導老師簽名:教師顧乃嘉 □格式不符 □計畫周延詳實 □計畫空泛不具體 □細節解釋不清楚 □通過 □初審缺失未修 □其他: 複審 □不通過 親自 雲道 名: 简前雅 審查委員會核章處 調爪凹額亚 đ,

捌、自主學習計畫書及在校學習紀錄表

研級	应號	书 請人	- 計畫主題	申請學期		
1-7	24	罕家重	红外線分段計時器			
節次	執行日日		學習內容、進度	場地/設備	指導老師簽名或點評	
1	3/5	编宫LCDT	户核程式	教室/康 腦	1148日台书	
2	34	導入 millis e	面式		此時非已台華	
3	3/5	光敏電阻	挐訯		副語台華	
4	3/2	整合程式7	A.		高端合举	
5	3/12	整合	¢		常語台華	
6	3	敷店			國際自己台華	
7	3 19	整合			10月1日本	
8	3/19	乾吉			警招台華	
9	3 19	自動化數:	橡		和招台華	
10	3 16	自動化教	據		****·吕台華	
11	3/26	自動化數	據		發展 書名 音举	
12	3/26	自動化數	據		教师派吕台華	
13	4.9	则量但別月	肉應幾 顏比訊號测量		★新来日 台章 ● 書日 台章	
14	f.g.	類比認號面	1.)對咸盛箔 编 號		教師兼呂台華	
15	4.g	测式,實際	谭 4分		和联合车	
16	9/16	美化包装外	觀		教师兼吕台華	
17	916	美化包装外	覾		教师兼吕台華	
18	9/16	美化包装料	R		教師兼呂台華	