

新竹市第四十一屆中小學科學展覽會

作品說明書

科 別：生活與應用科學科(一)(機電與資訊)

組 別：國小組

作品名稱：來去難逃我算計！

關 鍵 詞：Arduino、雷射切割、人流計數

編 號：

目 錄

壹、前言.....	1
一、研究動機.....	1
二、研究目的.....	1
三、文獻回顧.....	2
貳、研究設備及器材.....	4
一、Arduino UNO 主機板.....	4
二、Arduino UNO 九合一多功能擴充版.....	5
三、紅外線感測模組 (LM393 IR LED)	5
四、紅外線光電感測模組(E18-D80NK).....	5
五、Arduino 藍芽模組 (DX-BT19-A)	5
六、液晶顯示器(LCD 1602A)	6
七、Infinity X1V8 中型 3D 列印機.....	6
八、Beambox Pro 雷射切割機.....	6
九、其他使用材料.....	6
參、研究過程或方法.....	7
一、研究問題與假設.....	7
二、研究流程.....	7
肆、研究結果.....	8
一、採用 Arduino 程式開發板發展平均速率測量裝置	8
二、採用 Arduino 程式開發板發展人流檢測器	13
三、人流檢測器應用於實際情境進行改良.....	20
伍、討論.....	26
一、各類型應用與發展.....	26
二、藍芽無線設備與行動載具連動.....	28
陸、結論.....	29
柒、參考文獻資料.....	30

摘要

本研究以 Arduino 開發板為主架構，使用紅外線感測器、雷射製作底板來自製一組平均速率測量裝置，再延伸發展為人流檢測器，可作為特定場所容留人數的計算。並利用 LCD 液晶螢幕來顯示容留人數，及設有綠、黃、紅燈號警示。人流檢測器發展為四個階段：

【階段一】：單支感測器，用於單一方向進出的人流計算。

【階段二】：設有進門、離開感測器各一支，用於同門進出的人流計算。

【階段三】：於階段二程式碼加入時間差參數，防止人流誤觸感測器。即若有人經過「進門感測器」，必須在時間差內接觸到「離開感測器」，表示確實進入空間內，而非誤觸。

【實境改良】：門口二側有進門、離開感測器各一組，解決多人接續進出、同進同出的人流精算問題，並加入燈號警示。

壹、前言

一、研究動機

當我們開始學習程式設計時，討論到紅外線感測器有沒有可以運用在校園生活中的地方？想起之前跑 100 公尺的經驗，因為老師在進行手動按壓碼表計時，或多或少會有一些人為誤差存在，也有可能不小心漏按馬錶，導致計時結果不精準甚至需要重測，因在六年級的課程中，已學到平均速率以及力與運動的概念，激發了我們想要利用 Arduino 以及紅外線感測器，製作一臺可以測量時間差的機器，並進而計算出平均速率。

從 2019 年 12 月底開始，新冠肺炎肆虐全球，為了避免太多人在密閉式空間而造成感染，有些店家開始進行人流控管，但大部分店家都採用人工計數，而我們想用機器來取代人力，雖然一些大型的連鎖店家也會使用機器計數，但我們發現其機器造價昂貴，難以普遍使用。

再加上韓國的梨泰院踩踏事件，釀成一場沒有做好人流控管而產生群眾雪崩的悲劇，當時人潮密集到大家幾乎無法動彈時，有人跌倒後，群眾如同骨牌效應般倒下，導致死傷慘重，更讓我們了解到人流控管的重要性，避免悲劇再一次的重演，於是萌發我們將紅外線感測器發展為人流檢測器的動機。

二、研究目的

- (一) 採用 Arduino UNO 及紅外線感測器製作平均速率測量裝置。
- (二) 運用平均速率測量的概念延伸發展人流檢測器。
- (三) 探討人流檢測器應用於實際情境可能會遇到的問題並加以改良。

三、文獻回顧

(一) 室內空間人流相關規範

「容留人數」指的是特定場所顧客人數、從業員工人數及其他在場人數之合計。根據「臺北市特定場所容留人數管制規則」將室內空間場所區分為三種類型(因新竹市消防局缺乏相關公開資訊，故引用臺北市之規範)。

表 1-3-1 臺北市特定場所容留人數管制量計算表

類別	場所名稱		容留人數計算方式				容留人數管制量
第一類	舞廳、舞場、酒家、酒吧、飲酒店及其他類似場所	設固定席位	從業員工數 A	從業員工數以外之容留人數 B (C+D+E)			A+B
		未設固定席位		從業員工數 A	其他部分 B [其他部分面積(m ²)/1 (m ² /人)]		
第二類	視聽歌唱及其他類似場所		從業員工數 A	從業員工數以外之容留人數 B (C+D+E)			A+B
				固定席位部分 C	固定席位(連續式)部分 D[座椅正面寬度合計(m)/0.5 (m/人)]	其他部分 E [其他部分面積 (m ²)/3 (m ² /人)]	A+B
第三類	百貨商場、超級市場及其他類似場所、經指定之臨時大量聚集人潮之室內場所		從業員工數 A	其他部分 B [其他部分面積(m ²)/1.5 (m ² /人)]			A+B

備註：第一類場所（設固定席位）：指特定場所實際使用樓地板面積合計扣除固定席位區域及非消費性空間面積合計所得之數。

第一類場所（未設固定席位）：指特定場所實際使用樓地板面積扣除非消費性空間面積合計所得之數。

第二類場所：指特定場所實際使用樓地板面積合計扣除固定席位區域及非消費性空間面積合計所得之數。

第三類場所：指特定場所實際使用樓地板面積合計扣除非消費性空間面積合計所得之數。

(二) 市售人流檢測器分析與探討。

總整目前市售各項人流檢測器進行分析，我們發現市售多數檢測器所採用的原理與本研究相似，皆是採用紅外線感測器製作，但其是利用 3D 監測系統進行全域空間的掃描，達到更精確的人流計數，價位相較於我們自製的 Arduino 人流檢測器來說皆昂貴許多，若要在學校等非商業用途使用易因經費不足而使用受限，因此透過 Arduino 開發版自製人流檢測器可大幅度降低應用成本(約為 500 元以下)，能提供更多空間計數使用。

1. 人流量計數器(130-SCT15+)：

於 112 年 3 月份中旬 Pchome 所查詢的價位為 5,899 元，其運作原理採用紅外線反射進行人流檢測，自動辨識進出方向並統整空間內部人數，感應距離約 0~3m(不建議超過 2.5m，建議設置高度為 120cm)，計數範圍為 1~99999 次，可透過切換按鈕單獨顯示空間內部人數、進入人數以及離開人數，並設有歸零按鈕，使用雙面膠或螺絲釘固定在門左右並安裝電源即可立即使用。



2. 人流計數器(MET-CC999)：

為錫特工業科技有限公司所開發的場域人數顯示面板，於 112 年 3 月份中旬公司官網所查詢的價位為 12,950 元，其運作原理採用紅外線反射進行人流檢測，設置在出入門口，一旦有人通過紅外線感應區域後，計數器自動人數+1、出-1，可計數範圍為 1~999 次。場域人數顯示面板具有人數管制的功能，當人數超過管制數量時則會響起超額警報，本面板具有遠端遙控設置功能，能夠透過遙控器將時間點或是人數自動歸零，提升操作的便利性。



3. FootfallCam 3D Pro2

FootfallCam 成立於 2002 年，是一家專注開發人流量統計的國際化公司，FootfallCam 3D Pro2 是一透過 AI 驅動的 3D 立體視覺人流量統計裝置，並具有弱光與夜間模式，運用 3D 立體圖像處理和 AI 分析方式進行大數據演算，其準確度高達 99.5%，除此之外運用雙鏡頭捕捉與演算法可針對對象進行分類，包含輪椅、手推車、兒童等.....，其價位未顯示於官網中，需透過報價單詢價，依功能性及複雜度推估價位約在數萬元甚至數十萬元以上。



貳、研究設備及器材

一、Arduino UNO 主機板

Arduino 在國小的程式學習階段較不廣泛，常使用 micro:bit 或是 brain go 等較簡易的程式開發板，但若以廣泛性及程式設計的彈性而言做考量，我們最終仍採用 Arduino 做為後續紅外線感測器的程式設計使用，維基百科是這樣介紹 Arduino 的：「Arduino 是一家製作開源硬體和開源軟體的公司，同時兼有專案和用戶社群，該公司負責設計和製造單板微控制器和微控制器套件，用於構建數位裝置和互動式物件，以便在物理和數位世界中感知和控制物件。」

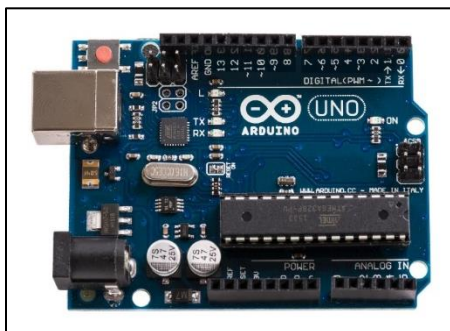


圖 2-1-1 Arduino 開發板(正面)

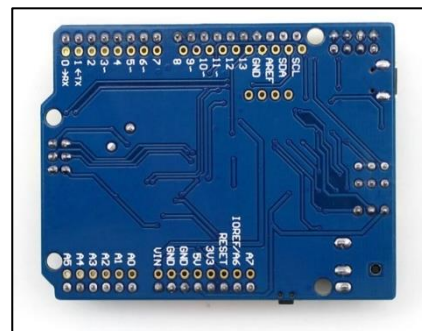


圖 2-1-2 Arduino 開發板(背面)

二、Arduino UNO 九合一多功能擴充版

多功能擴充版包含了按鈕、LED 燈、DHT11 溫濕度感測器、蜂鳴器、可變電阻、光敏感測器、紅外線接收器等各項不同功能，並提供 LED 顯示器腳位，藍芽或超音波腳位，兩個數位腳位及一個類比腳位可供使用，一塊擴充版上集成多種模塊功能，不用焊接，不用連線，直接下載程序就可以完成實驗，對於初次接觸 Arduino 的小學生而言相對容易上手。(圖 2-2-1)

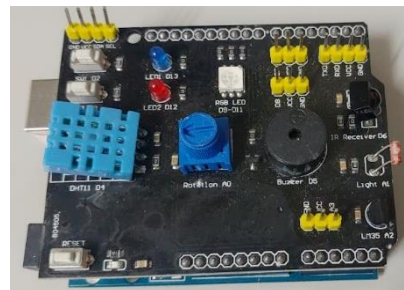


圖 2-2-1 Arduino UNO 九合一多功能擴充版

三、紅外線感測模組 (LM393 IR LED)

LM393 紅外線感測模組引出腳位包含 VCC、GND、OUT，透過紅外線的輸出及接收，可以檢測約 2 ~ 25cm 距離內的物體，可以做為物體經過的感測裝置使用，價格便宜但感測距離較短，其工作穩定電壓約為 5V 左右。(圖 2-3-1)

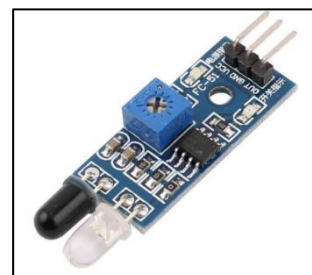


圖 2-3-1 LM393 紅外線感測模組

四、紅外線光電感測模組(E18-D80NK)

E18-D80NK 紅外線光電感測模組引出腳位包含 VCC、GND、OUT，透過紅外線的輸出及接收，可以檢測約 3 ~ 80cm 距離內的物體，能改善 LM393 距離較短的問題，可以做為物體經過的感測裝置使用，其工作穩定電壓約為 5V 左右。(圖 2-4-1)



圖 2-4-1 E18-D80NK 紅外線光電感測模組

五、Arduino 藍芽模組 (DX-BT19-A)

HC-06 引出腳位包含 VCC、GND、TXD、RXD，其輸入電壓為 3.6 ~ 6V，此藍芽模組中的 LED 指示燈可以表示藍芽使用情況，當指示燈閃爍時表示藍芽未連接，此時可以用電子設備連接藍芽模組，當 LED 燈長亮時表示藍芽已連接。(圖 2-5-1、圖 2-5-2)

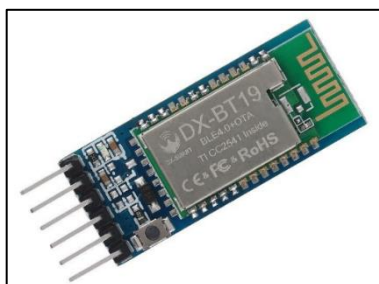


圖 2-5-1 Arduino DX-BT19-A 藍芽模組正面

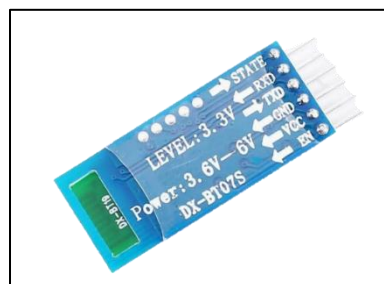


圖 2-5-2 Arduino DX-BT19-A 藍芽模組背面

六、液晶顯示器(LCD 1602A)

本液晶顯示器可以顯示兩行，每行 16 個字元，藍底白字，僅能顯示英文字母、希臘字母、標點符號以及數學符號，具有 LED 背光的功能，外框：80 x 36 x 11mm，玻璃尺寸：69.5 x 21.8 x 17.8mm，其工作穩定電壓約為 5V 左右。(圖 2-6-1)

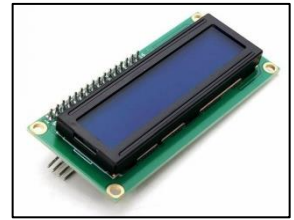


圖 2-6-1 LCD 1602A 液晶顯示器

七、Infinity X1V8 中型 3D 列印機

此 3D 列印機可兼容市售大多數常見之 3D 列印耗材，其噴頭溫度介於 100 ~ 250 °C，列印的容積為 210 x 210 x 240 mm，其應用性相當廣泛，因此作為本次科展的零組件設計使用，藉由科展的研究也讓我們更為了解目前科技的日新月異，以及能夠自行透過 3D 繪圖軟體繪製本研究所需要的零組件，作為感測器裝設支架使用。(圖 2-7-1)

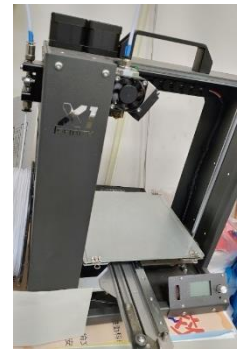


圖 2-7-1 Infinity X1V8 中型 3D 列印機

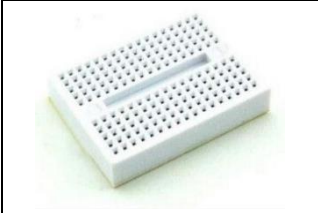






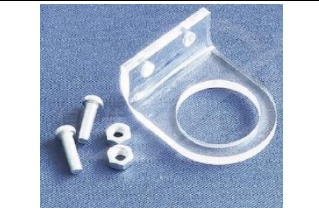
八、Beambox Pro 雷射切割機

本雷射切割機工作範圍 60 x 37.5 x 8 cm，可支援多項繪圖檔案，軟體操作相當簡易，小學生也可以透過電腦自己設計想要的雷射切割作品，且可以有效解決 3D 列印印製大範圍零件需要耗費過多時間的問題，與 3D 列印機搭配使用，可以有效提升科展製作的品質，此切割機不只是木板，皮革、壓克力甚至是金屬等材料都可以使用，作為創作的工具相當具有彈性。(圖 2-8-1)



圖 2-8-1 Beambox Pro 雷射切割機

九、其他使用材料

			
麵包板	杜邦線	18650 電池	18650 電池盒
			
PP 板	雷射切割木板	3D 列印材料(PLA)	E18-D80NK 支架

參、研究過程或方法

一、研究問題與假設

依據上述研究動機以及研究目的，本研究提出下列研究問題：

- (一) 是否能透過 Arduino UNO 程式開發板製作平均速率測量裝置？
- (二) 是否能透過 Arduino UNO 程式開發板製作人流檢測器？
- (三) 是否能貼近日常生活情境，將人流檢測器運用於生活當中？

二、研究流程

本研究之流程共區分為四個階段進行，依序為「研究準備」、「程式學習」、「紅外線工具應用開發」、「結果探討」，本研究流程如圖 3-2-1 所示。

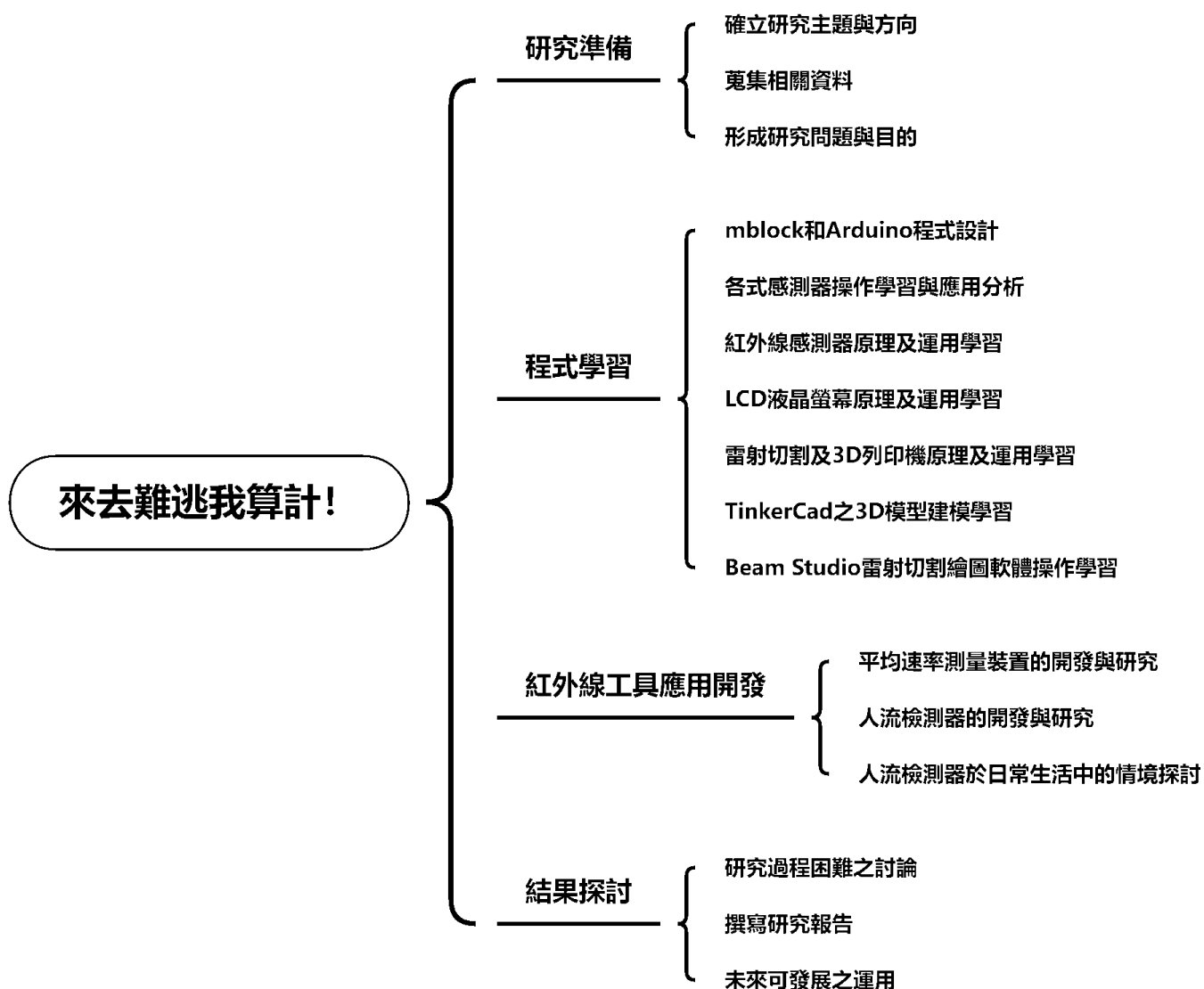


圖 3-2-1 研究流程圖

肆、研究結果

一、採用 Arduino 程式開發板發展平均速率測量裝置

採用 Arduino 連接及雷雕模組裝設，將紅外線感測器裝設在麵包板上，並連接 LCD 液晶螢幕，運用兩顆 18650 鋰電池供電，以方便觀測，透過程式設計及線路連接，以達到測量兩點間時間差與平均速率的目標，本實驗裝置可運用於數學和自然課程中，在六下數學已學習到距離除以時間差會等於平均速率，可利用我們所製作的平均速率測量裝置，在相同距離內，去測量移動時間，即可運算出平均速率，國道或快速道路上常見的區間測速系統，原理也與我們的研究裝置相似；而自然的力與運動單元則會讓我們知道在相同的距離內，使用較大的力量推動物體，物體從起點移動至終點所需的時間較短，此現象也可以利用平均速率測量裝置測量，可以直接感受到用大小不同的力量從起點推動軌道上的物體時，到達終點所需要的時間會不同。

(一)平均速率測量裝置裝設與接線

採用 PP 塑膠板作為裝置測量基座，並利用長條塑膠板製作軌道，以作為運動範例(乒乓球)滾動使用，並利用方形塑膠板作為擋板，將麵包版、Arduino 開發版(含擴充板)、電池盒與 LCD 液晶螢幕固定於長方形 PP 板上，將 LED 裝設於麵包板上，兩顆 LED 距離設定為 30 公分，LED 上訊號線連接至擴充板上的 D7(起點)與 D8(終點)，VCC 與 GND 接連接至擴充板接電與接地，LCD 液晶螢幕上的 SCL、SDA、VCC 與 GND 分別連接至擴充板上所相對應的腳位，並將 LED POWER 腳位額外連接至依 VCC 腳位接電，已啟動 LCD 螢幕背光，實體裝置圖請參考圖 4-1-1，完整裝置接線圖如圖 4-1-2 所示。

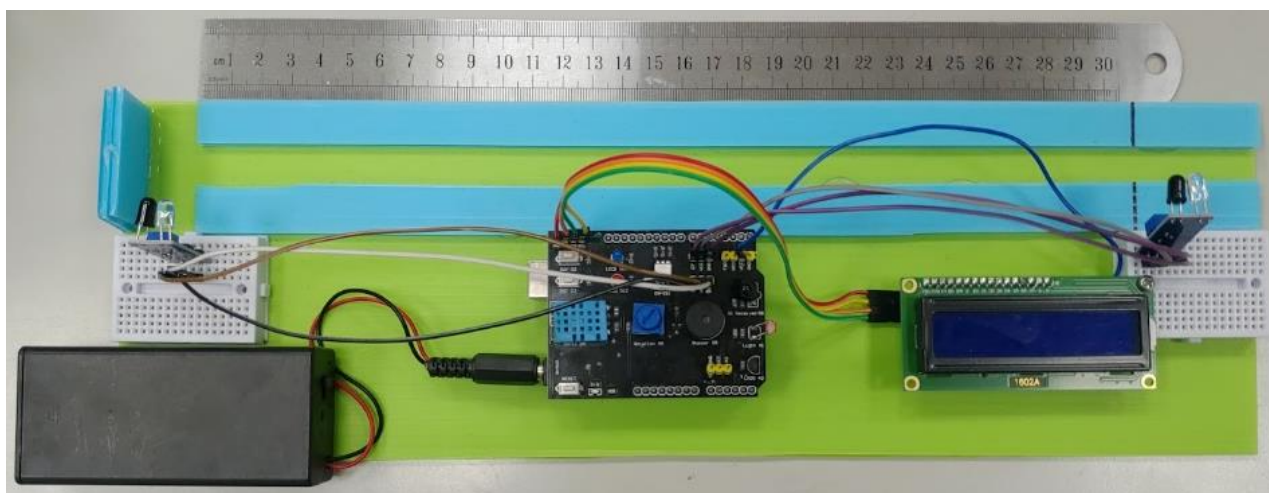


圖 4-1-1 Arduino 平均速率測量裝置圖

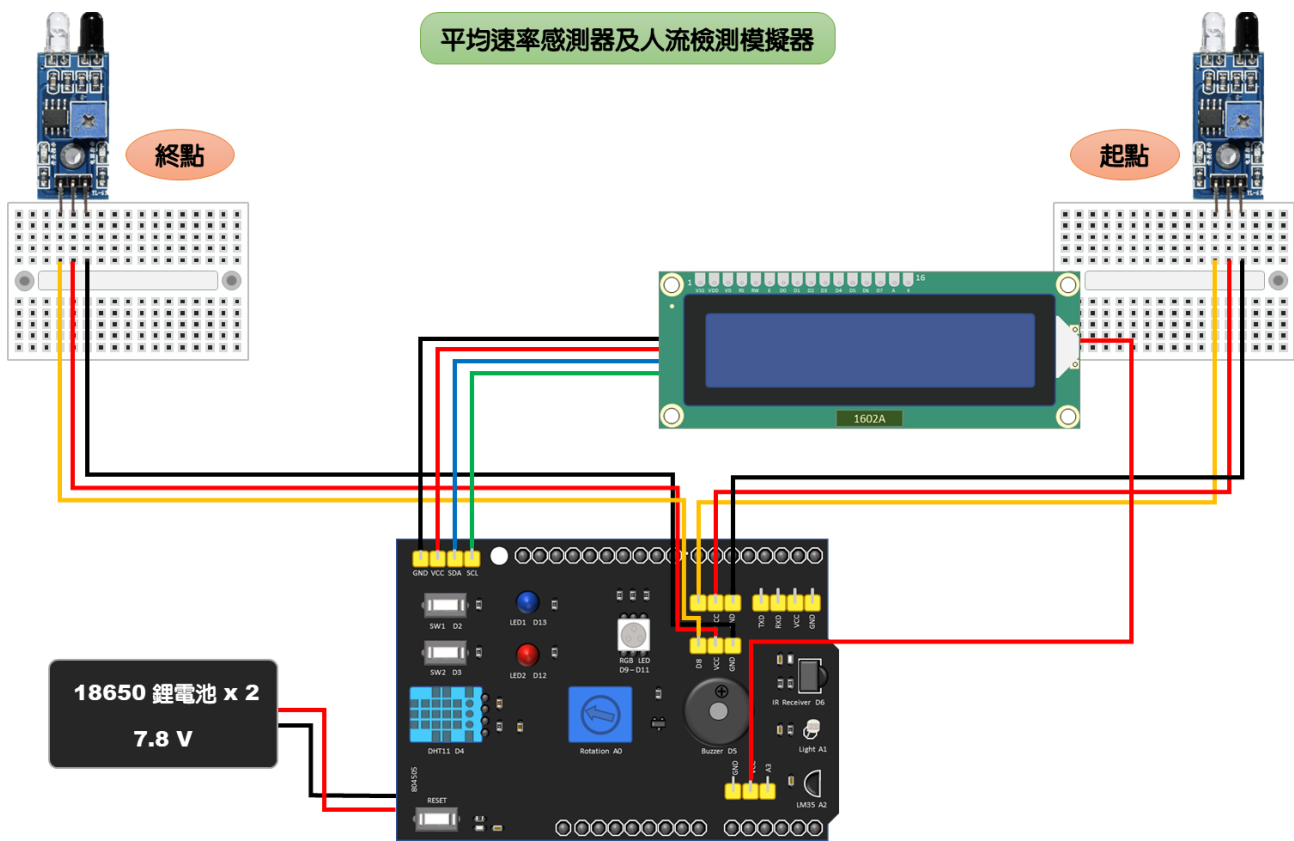


圖 4-1-2 Arduino 平均速率測量裝置接線圖

(二)平均速率測量裝置程式碼說明

由於我們會在裝置中加裝 LCD 液晶顯示器與外接電源，方便未能連接在電腦也可操作使用，因此在程式碼中會分成 LCD 液晶顯示器相關與紅外線速率感測相關，以下程式說明將分成三個階段進行：「參數設置」、「void setup」與「void loop」。

1. 參數設置：

下述兩行程式碼主要是針對 LCD 液晶螢幕的使用進行撰寫，首先載入預先載下的程式資料庫<LiquidCrystal_I2C.h>，讓 Arduino 知道函數資料庫中有這個資料可以使用，並告知 Arduino 說 LCD 的液晶面板大小為 2 行 16 列。

```
#include <LiquidCrystal_I2C.h> //LCD函數資料庫
LiquidCrystal_I2C lcd(0x27, 16, 2); //LCD面板大小
```

接下來定義各個參數名稱以及訊號腳位，讓 Arduino 知道第一個(起點)紅外線感測器連接 Arduino 的第 7 腳位，第二個(終點)紅外線感測器連接 Arduino 的第 8 腳位，並建立 double 參數 begintime(起始時間)、endtime(結束時間)、timelag(兩點間的時間差)、d = 0.3 兩點間的距離 30 公分、v(平均速率)，並建立 start = 0 的整數變數標籤(作為是否通過起點後才通過終點的檢視使用，以避免 BUG 的出現)。

```
const int IR_Sensor1=7; //起點紅外線腳位(整數)
const int IR_Sensor2=8; //終點紅外線腳位(整數)
double begintime; //建立起始時間的參數
double endtime; //建立結束時間
double timelag; //建立兩點間的時間差
double d = 0.3; //建立兩點間的距離 (30公分)
double v; //建立平均速率的參數
int start=0; //宣告start為數字0 (整數變數)
```

2. void setup(開機時會執行一次)：

在 Arduino 開機後告知 Arduino 利用 IR_Sensor1 作為訊號輸入端(第 7 腳位)，利用 IR_Sensor2 作為訊號輸入端(第 8 腳位)。

```
pinMode (IR_Sensor1, INPUT); //使用IR_Sensor1作為訊號輸入端
pinMode (IR_Sensor2, INPUT); //使用IR_Sensor2作為訊號輸入端
```

LCD 液晶顯示器於開機時進行初始化並自動開啟背光照明，並於顯示器最左上角(0, 0)的位置顯示「Time：」，最左下角(0, 1)的位置顯示「Velocity：」。

```
lcd.init(); //初始化LCD
lcd.backlight(); //是否開啟背光
lcd.setCursor(0,0); //從(0,0)開始顯示，最左上角
lcd.print("Time:"); //顯示「Time:」
lcd.setCursor(0,1); //從(0,1)開始顯示，最左第二行
lcd.print("Velocity:"); //顯示「Velocity:」
```


3. void loop(開機後會持續執行)：

假如(if)「起點感測器」感測到物體經過時，Arduino 會設 `begintime` 為當下的時間點(當 Arduino 開機後到感測物體瞬間所花的時間)，並將起點偵測器標示為已啟用(從 0 改為 1)。

```
if(digitalRead(IR_Sensor1)==LOW) //「起點」的紅外線感測到物體
{begintime = micros(); //Arduino開機到起點紅外線感測到物體瞬間所花的時間
start=1;} //宣告起點偵測器已啟用
```

假如(if)「終點感測器」感測到物體經過時，且起點偵測器標示為已啟用，Arduino 會設 `endtime` 為當下的時間點(當 Arduino 開機後到感測物體瞬間所花的時間)；計算「時間差(`timelag`)」為「終點時間(`endtime`)」減掉「起點時間(`begintime`)」，再將其數值由「微秒」換算成「秒」，並將起點偵測器標示為未啟用(從 1 改為 0)，最終運用「距離(`d`)」除以「時間差(`timelag`)」來計算物體移動的「平均速率(`v`)」。

```
if(digitalRead(IR_Sensor2)==LOW&&start==1) {
//「終點」的紅外線感測到物體且起點偵測器已啟用
endtime = micros(); //Arduino開機到終點紅外線感測到物體瞬間所花的時間
timelag = (endtime-begintime)/1000000; //起點到終點所需時間(微秒換算單位為秒)
start=0; //宣告起點偵測器未啟用
v = d / timelag; // 計算平均速率(距離除以時間差)
```

將「時間差(`timelag`)」、「時間單位(s)」、「平均速率(`v`)」與「平均速率單位(m/s)」顯示於 LCD 液晶顯示器上。

```
lcd.setCursor(5,0); //在LED第一行第五格開始顯示
lcd.print(timelag); //顯示兩點間的時間差距
lcd.setCursor(9,0); //在LED第一行第九格開始顯示
lcd.print("s"); //顯示「s」
lcd.setCursor(9,1); //在LED第二行第九格開始顯示
lcd.print(v); //顯示計算出的平均速率
lcd.setCursor(13,1); //在LED第二行第十三格開始顯示
lcd.print("m/s"); //顯示「m/s」(平均速率的單位)
```



圖 4-1-3 LCD 螢幕顯示器顯示畫面圖

(三)雷切底板繪製

為改善使用 PP 塑膠板製作裝置底板與軌道可能產生的幾個問題：1. 人為測量定位與切割時可能產生誤差，2. 需要手工切割與製作不易作為大量推廣使用，因此採用 Beambox Pro 雷射切割機與密集板進行板材設計，其設計目標是能將此感測器底板標準化，確保裝置設置的準確性，以提升最終測量的數據精準度，底板上會清楚表示各項電子零件所需裝設的位置，以避免未來製作底板支架時能夠盡量減少裝設誤差。(圖 4-1-4、圖 4-1-5、圖 4-1-6)



圖 4-1-4 平均速率測量裝置底板繪製過程

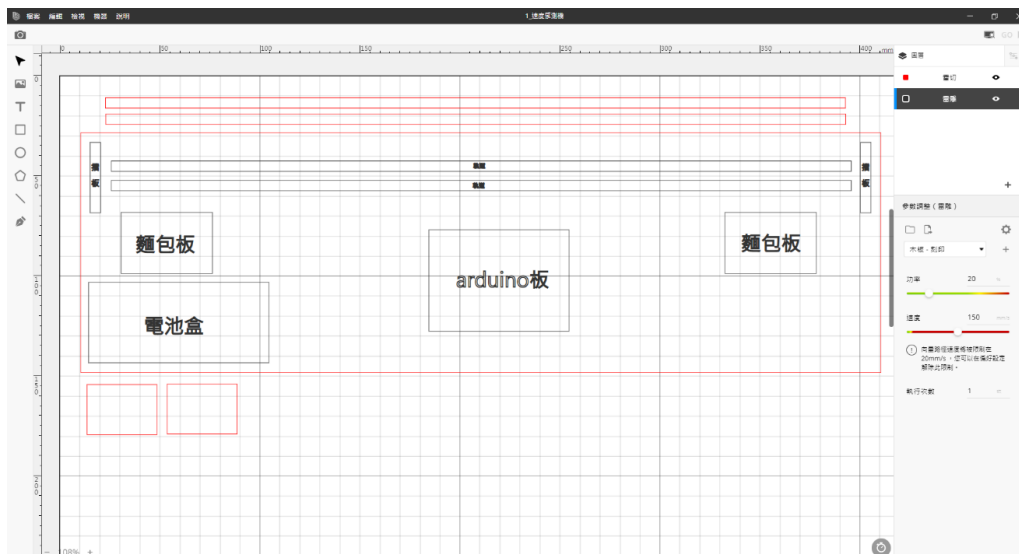


圖 4-1-5 平均速率測量裝置底板雷切設計圖稿



圖 4-1-6 平均速率測量裝置雷切底板成品

二、採用 Arduino 程式開發板發展人流檢測器

透過 Arduino 連接及雷切模組裝設，將紅外線感測器裝設在麵包板上，並連接 LCD 液晶螢幕，運用兩顆 18650 鋰電池供電，以方便觀測，透過程式設計及線路連接，以達到測量兩點間時間差與平均速率的目標。

(一)人流檢測器裝設與接線

人流檢測器模擬裝置是採用平均速率測量裝置，僅修改程式碼內容，裝設於出入口之人流檢測器裝置將於最終階段再做更進一步的詳細敘述。

(二)人流檢測器裝置程式碼說明【階段一】

人流檢測器的開發歷程共為五個階段，階段一為基礎人流檢測器裝置，僅採用單一紅外線感測器作為單一出入口的裝置使用，以下程式設計說明皆以單一出入口進行，若有多個出入口，僅需增加紅外線感測器及程式碼即可達到目標。

1. 參數設置：

設置完成 LCD 程式碼後，讓 Arduino 知道紅外線感測器連接 Arduino 的第 7 腳位，並建立 `People = 0` 的整數變數，此變數為起始空間人數。

```
#include <LiquidCrystal_I2C.h> //LCD函數資料庫
LiquidCrystal_I2C lcd(0x27, 16, 2); //LCD面板大小
const int IR_Sensor1=7; //紅外線腳位(整數)
int People = 0; //設定起始空間人數
```

2. void setup(開機時會執行一次)：

於 LCD 液晶顯示器最左上角(0, 0)的位置顯示「People：」

```
pinMode(IR_Sensor1, INPUT); //使用IR_Sensor1作為訊號輸入端
lcd.init(); //初始化LCD
lcd.backlight(); //是否開啟背光
lcd.setCursor(0, 0); //從(0, 0)開始顯示，最左上角
lcd.print("People:"); //顯示「People:」
```

3. void loop(開機後會持續執行)：

簡化數位輸出腳位 `digitalRead (IR_Sensor1)` 為 `dig1`，以便後續程式撰寫使用。假如(if)紅外線感測器感測到物體時，則空間人數會增加 1 人，並暫停一秒鐘，以避免連續計算的 BUG 產生(此程式運作無法區分進入或離開的人，只要有人經過紅外線感測器時，空間人數就會增加 1 人，因此必須限制單一方向，若有同門進出動線需求，將於【階段二】進行改良)。

```
int dig1 = digitalRead(IR_Sensor1); //簡化成dig1
if(dig1==LOW) { //紅外線感測到物體
  People = People+1; //空間人數+1
  delay(1000); //暫停1秒鐘
```

將「空間人數(People)」顯示於 LCD 液晶顯示器上。

```
lcd.setCursor(8,0); //從(0,0)開始顯示，最左上角
lcd.print(People); //印出空間人數
```

(三)人流檢測器裝置程式碼說明【階段二】

【階段一】因只有一支感測器，必須限制人流僅能從一個門口進入、另一個門口離開，且可能容易有誤觸的問題存在，因此於【階段二】進一步改善此問題，於單一門口同時運用兩個紅外線感測器，將感測器區分為「進門感測器」與「離開感測器」二種，此方法可改善【階段一】無法同門進出的問題，當有人從門外走進門內時，空間人數會增加一人，當有人從門內離開時，空間人數則會減少一人。

待處理問題：雖然【階段二】可以有效解決【階段一】無法同門進出的限制，但【階段二】仍有紅外線感測器可能誤觸的問題，若有人準備進入空間已接觸「進門感測器」，但最終未確實走進空間內，此時空間人數仍會增加一人；或是有人準備離開空間且已接觸「離開感測器」，但最終未確實離開空間，此時空間人數仍會減少一人，此紅外線誤觸問題的 BUG 將於【階段三】進行改良。

1. 參數設置：

設置完成 LCD 程式碼後，讓 Arduino 知道進門紅外線感測器連接 Arduino 的第 7 腳位，出門紅外線感測器連接 Arduino 的第 8 腳位，並建立 fo=0 與 fi=0 的整數變數，分別代表從門外進來的人(from outdoor)，以及從門內離開的人(from indoor)，再建立 People=0 的整數變數為起始空間人數。

```
#include <LiquidCrystal_I2C.h> //LCD函數資料庫
LiquidCrystal_I2C lcd(0x27, 16, 2); //LCD面板大小
const int IR_Sensor1=7; //進門紅外線腳位(整數)
const int IR_Sensor2=8; //出門紅外線腳位(整數)
int fo = 0; //設定起始人數(從門外來的人from outdoor)
int fi = 0; //設定起始人數(從門內離開的人from indoor)
int People = 0; //設定起始空間人數
```

2. void setup(開機時會執行一次)：

採用 Sensor1(進門感測器)與 Sensor2(離開感測器)作為輸入源，並於 LCD 液晶顯示器最左上角(0,0)的位置顯示「People：」。

```
pinMode(IR_Sensor1, INPUT); //使用IR_Sensor1作為訊號輸入端
pinMode(IR_Sensor2, INPUT); //使用IR_Sensor2作為訊號輸入端
lcd.init(); //初始化LCD
lcd.backlight(); //是否開啟背光
lcd.setCursor(0,0); //從(0,0)開始顯示，最左上角
lcd.print("People:"); //顯示「People:」
```

3. void loop(開機後會持續執行)：

將「空間人數」設定為「進入的人數」減去「離開的人數」，並簡化數位輸出腳位 digitalRead (IR_Sensor1)為 dig1，digitalRead (IR_Sensor2)為 dig2，以便後續程式撰寫使用。

```
int People = fo - fi; //設定空間人數為進入-離開
int dig1 = digitalRead(IR_Sensor1); //簡化成dig1
int dig2 = digitalRead(IR_Sensor2); //簡化成dig2
```

假如(if)「進門感測器(dig1)」偵測到物體，且「離開感測器(dig2)」未偵測到物體時(為避免同時接觸二支感測器的狀況產生，提升檢測準確度)，進入人數會增加一人，並將 Arduino 暫停兩秒鐘，以避免接觸到「離開感測器(dig2)」時會觸發「減少人數的程式碼」。

```
if(dig1==LOW && dig2==HIGH){ //「門外」的紅外線感測到物體
  fo = fo+1;//進入人數+1
  delay(2000);//暫停2秒鐘
}
```

假如(if)「離開感測器(dig2)」偵測到物體，且「進門感測器(dig1)」未偵測到物體時(為避免同時接觸二支感測器的狀況產生，提升檢測準確度)，離開人數會增加一人，並將 Arduino 暫停兩秒鐘，以避免接觸到「進門感測器(dig1)」時會觸發「增加人數的程式碼」。

```
if(dig1==HIGH && dig2==LOW){ //「門內」的紅外線感測到物體
  fi = fi+1;//離開人數+1
  delay(2000);//暫停2秒鐘
}
```

將「空間人數(People)」顯示於 LCD 液晶顯示器上。

```
lcd.setCursor(8,0);//從(0,0)開始顯示，最左上角
lcd.print(People);//印出空間人數
```

(四)人流檢測器裝置程式碼說明【階段三】

【階段三】的目標是透過結合前述所製作平均速率測量裝置的程式思維進程式碼調整，運用 Arduino 內部的時間計時器作為時間差工具，並運用整數變數進行狀態標籤，最終可以有效解決【階段二】所遇到的誤觸問題。當人體經過「進門感測器」時，若沒有在一定時間內接觸到「離開感測器」，表示此人未進入本空間內，因此進入人數不會改變；當人體經過「離開感測器」時，若沒有在一定時間內接觸到「進門感測器」，表示此人未離開本空間，因此離開人數不會改變。

1. 參數設置：

運用【階段二】的程式碼，額外建立 `double` 參數 `startTime`(偵測到物體的時間)、`timelag`(時間差)，並建立 `door1 = 0` 與 `door2 = 0` 的整數變數標籤(作為門口標記使用，`door1` 為確保二支感測器不會相互干擾，`door2` 為確保室內空間人數大於等於 0 人，以避免 BUG 的出現)。

```
double startTime;//設定偵測到物體的時間
double Timelag;//設定時間差
int door1=0;//設定門口標記1，確保入門感測器正確性
int door2=0;//設定門口標記2，確保室內有人
```

2. void setup(開機時會執行一次)：

運用【階段二】的程式碼外，額外增加一行程式碼，讓 Arduino 知道需要運用 `millis()` (Arduino 開機到記錄當下的歷程時間)作為偵測到物體時的時間，記為 `startTime`。

```
startTime=millis();//偵測到物體時的瞬間
```

3. void loop(開機後會持續執行)：

「空間人數」設定為「進入的人數」減去「離開的人數」，並簡化數位輸出腳位 `digitalRead (IR_Sensor1)` 為 `dig1`，`digitalRead (IR_Sensor2)` 為 `dig2`，以便後續程式撰寫使用，並額外增加一行程式碼，讓 Arduino 知道 `Timelag`(時間差)會等於 `millis()` 與 `startTime` 兩者間的差，若紅外線感測器偵測到人體時，會將 `startTime` 設定為 `millis()`，因此瞬間的時間差會是 `Timelag = millis() - millis()`，因此時間差會趨近於零，達到歸零的效果。

```
int People = fo - fi;//設定空間人數為進入-離開
int dig1 = digitalRead(IR_Sensor1);//簡化成dig1
int dig2 = digitalRead(IR_Sensor2);//簡化成dig2
Timelag=millis()-startTime;
//時間差=「Arduino當下時間」-「偵測到物體的時間」，會趨近於零可視同歸零
```

(1)進門相關的程式碼設計

假如(`if`)「進門感測器」偵測到物體，且時間差(`Timelag`)大於 2 秒時，此時將 `startTime` 設定為當下瞬間的 `millis()`，因此時間差會自動歸零。

```
if (dig1==LOW&&Timelag>2000) { //「門外」的紅外線感測到物體，且時間差大於2秒
  startTime=millis();//時間差歸零
```

假如(if)「離開感測器」偵測到物體，且時間差(Timelag)小於等於 2 秒時，且 door1 標籤為 0 時(設置 door1 標籤是為了避免誤觸「減少人數的程式碼」而導致人數計算錯誤)，會使進入人數增加一人，因此空間人數會增加一人，並將 Arduino 暫停 1 秒鐘，以避免重複計算的 BUG 出現。

```
if (dig2==LOW&&Timelag<=2000&&door1==0) {  
    //「門內」的紅外線感測到物體，且時間差小於等於2秒，且door1=0  
    fo = fo+1;//進入人數+1  
    delay(1000);//暫停1秒鐘
```

時間差 2 秒鐘為避免感測器誤觸的設計，要求人流必須在一定時間內先後經過兩支感測器，此設計可以有效解決【階段二】所遇到的感測器誤觸問題；暫停 Arduino 時間 1 秒鐘為避免重複計算的 BUG 出現，以上所提及「時間差」與「暫停時間」的數據選用為撰寫程式碼時的推估時間，實際空間使用所需的確切時間將於下階段進行實驗測量與詳細說明。

(2)離開相關的程式碼設計

假如(if)「離開感測器」偵測到物體，且時間差(Timelag)大於 2 秒時，且 door2 標籤為 1 時(表示空間人數大於 0 時)，此時將 startTime 設定為當下瞬間的 millis()，因此時間差會自動歸零，並將 door1 標籤更改為 1。

```
if (dig2==LOW&&Timelag>2000&&door2==1) {  
    //「門內」的紅外線感測到物體，且時間差大於2秒，且door2=1(裡面有人)  
    startTime=millis();//時間差歸零  
    door1=1;//door1改成1
```

假如(if)「進門感測器」偵測到物體，且時間差(Timelag)小於等於 2 秒時，且 door1 標籤為 1 時(設置 door1 標籤是為了避免誤觸「增加人數的程式碼」而導致人數計算錯誤)，會使離開人數增加一人，因此空間人數會減少一人，並將 Arduino 暫停 1 秒鐘，以避免重複計算的 BUG 出現，最終將 door1 標籤更改回 0。

```
if (dig1==LOW&&Timelag<=1000&&door1==1) {  
    //「門外」的紅外線感測到物體，且時間差小於等於2秒，且door1=1(避免BUG)  
    fi = fi+1;//離開人數+1  
    delay(1000);//暫停1秒鐘  
    door1=0;//door1改成0
```


(3)空間人數相關程式碼

假如(if)空間人數大於一人時，將 door2 標籤更改為 1，表示空間內部有人，「離開感測器」與程式碼才會執行，假若空間人數為大於一人時，則 door2 標籤則為 0，表示空間內部無人，此時即使有人從內部離開時 Arduino 也不會產生作用，最終再將「空間人數 (People)」顯示於 LCD 液晶顯示器上。

```
if (People>0) { //室內有人時door2改成1
  door2=1;
}
else { //室內沒人時door2改成0
  door2=0;
}
lcd.setCursor (8,0); //從(8,0)開始顯示
lcd.print (People); //印出空間內人數
```

待處理問題：雖然【階段三】的人流檢測器功能性已達到我們所設定的基本需求，但仍需透過實驗來針對「時間差」與「暫停時間」的數據選用進行更詳細的探討；再者，【階段三】仍有多人接續進出，以及兩人同進同出或一進一出的問題待解決；最後，目前的人流檢測器皆是採用平均速率測量裝置進行程式編寫與模擬操作，必須將感測器裝設於真實環境中，且平均速率所採用的 LM393 紅外線感測模組僅能感測 2 ~ 25cm 範圍內的物體，若要於現實生活中進行應用將會大大限縮其功能性，因此須將其替換，這些問題將於後續進行更進一步的改良。



圖 4-2-1 人流感測器程式碼撰寫與討論歷程

三、人流檢測器應用於實際情境進行改良

針對【階段一、二、三】平均速率所採用的 LM393 紅外線感測模組感測範圍距離不足的問題，我們將其替換成 E18-D80NK 紅外線光電感測模組，其感測範圍介於 3 ~ 80cm，已足夠適用於一般室內空間的出入口(測量學校自然教室前後門的寬度均為 92cm)，因此接下來的將採用 E18-D80NK 作為本研究的紅外線感測器，進行人流情境並針對其設計裝置支架。

(一)各式人流情境探討與改良

1. 改善多人接續進出的問題：

測量科展組員(5 位小學六年級學生：2 位男生，3 位女生)的步行平均速率，每生測量十次，進一步推估小學六年級學生步行平均速率約為 1.92m/s，運用此步行平均速率以及最小步行平均速率作為二支感測器間的感測時間延遲考量，透過 Arduino 計算，我們發現科展夥伴們在 80cm 的距離間所需的步行平均時間為 0.43 秒，而最高需花費 0.54 秒的時間，因此我們將原先的感測間距時間縮短至 1 秒鐘(兩倍最高花費時間並捨去取整數)，且暫停時間設置為 0.57 秒(考量步行平均時間，確保下一人次接觸「進門感測器」前時間已超過 1 秒鐘，兩者相加為 1 秒鐘)，經測量結果發現單人進出時可順暢計數人流，不易有重複計算或是未計算的情形發生，且設置此時間(0.57 秒)時，經由測試發現多人接續進出不會有問題產生，仍可持續計算人數(假定隊伍排列為一長列，倆倆間距皆一個手臂長以上，若兩人太過靠近則計數可能有誤差，但一般走路的狀況倆倆不會過於緊貼，此狀況可視為特例不考慮)。

後來我們發現二支感測器的距離若盡可能縮短，除了可方便裝設裝置並且節省空間外，其計數的準確性也會因而提升，比較不會有太多誤差產生(距離過長可能會出現上一個人還未接觸「離開感測器」前，下一個人就已經接觸到「進門感測器」，因此會導致 Arduino 少計算到一個人次)，但若二支感測器距離過短，可能會因為人的身體厚度而導致二支感測器同時感測到，為避免此情況的發生，我們測量各組員的身體厚度，約介於 20 ~ 30cm，因此我們最終決定將二支感測器間距縮減至 40cm(我們後續研究發現距離縮短到 10cm 仍可使用，但若人體在門口停留過久可能還是會有部分誤差值存在)，並將間距時間縮短至 0.5 秒鐘，且暫停時間設置為 0.285 秒，經由測試可以發現連續人流進出並不會有問題產生，仍可以持續計算人數，運用時間調整與搭配即可有效處理多人接續進出的問題。

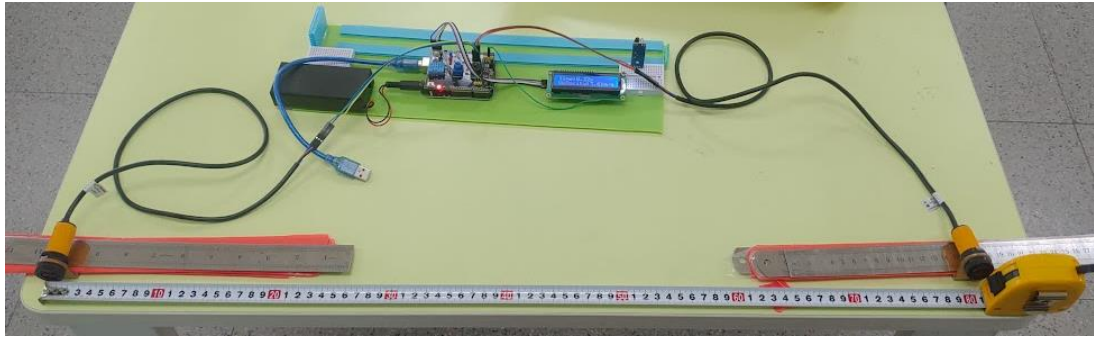


圖 4-3-1 人體步行平均速率測量裝置圖



圖 4-3-2 平均步行速率與時間差測量過程圖

2. 改善同進同出的問題：

(1) 不須考慮同進同出的情況：門口寬度過窄，不允許同時兩人進出。

透過測量 20 位小學六年級學生(10 位男生、10 位女生)的平均肩膀寬度(雙臂自然垂下測量)，最終計算其平均值為 40.275cm，因此本研究若門寬 $< 80.550\text{cm}$ (兩倍平均肩寬)，兩人不易同時進出門口，因此可忽略同進同出的問題。



圖 4-3-3 小學六年級學生平均肩膀寬度測量歷程圖

(2) 須考慮同進同出的情況：

本研究設定若門寬 $> 86\text{cm}$ ，則須考慮兩人同進同出的問題，以本校自然科教室門寬 92cm 為例進行討論，利用 E18-D80NK 紅外線光電感測模組測量距離 3 ~ 80cm 的特性進行調整，透過測量發現小學六年級學生的肩寬大多介於 40 ~ 50cm 之間，身體厚度大多介於 20 ~ 30cm

之間，若門寬足以可提供兩人同進同出的寬度時可能會產生四種情況：「單一個人中心點從門口正中間(46cm 處)進出」、「單一個人中心點從門口偏左或偏右進出」、「兩人肩並肩進出」、「兩人側身一前一後進出」，只要我們在門口左右二側各裝置一組紅外線感測器(二支進門感測器、二支離開感測器)，並將感測器的測量距離調整至符合前述的四種情況，即可計數同進同出的人流(但我們發現紅外線感測器若兩兩相對時，可能會產生誤觸紅外線感測器而導致計算誤差，因此左右兩側感測器在設置時不可直接相對，必須有約 5~10cm 的前後距離差，則可避免此現象發生)。最終透過多次測試的結果，我們發現將其中一側的感測器感測距離設置為 10cm，另一側的感測器距離設置為 40cm，則可以達到有效率且容錯機率最的狀況(若門口寬度過寬的情況，可在門口中心的上方加裝一組感測器計數從正中間經過的人流)，只要「紅外線感測的範圍」約略等於「門口寬度」扣除「六年級學生平均肩寬」，且「紅外線未感測的範圍」未處於門口中心點，此狀況下則可大幅減低計算誤差，可能的誤差來自於：a. 低於平均肩寬的學生，恰好由「紅外線未感測的範圍」進出、b. 兩人側身進出時，其中一側的學生恰好由「紅外線未感測的範圍」進出，但經由多次的測試，我們發現這二種狀況的發生率極低，因此可暫時忽略不計，留待未來研究再進行更進一步完善的改良。

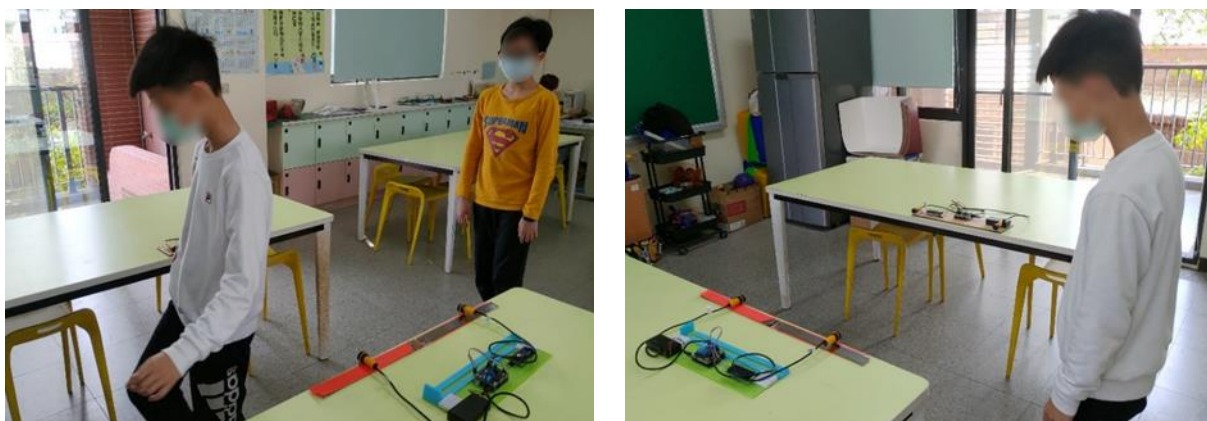


圖 4-3-4 門口同進同出情況測試過程圖

(二)室內外空間人數警示功能

確保人流檢測器於實際情境可正常運作後，我們運用擴展板內建的 RGB 模組做為研發的警示模擬操作使用，採用綠色(表示空間內部人數偏少舒適)、黃色(表示空間內部人數偏多)以及紅色(表示空間內部人數過高危險)，警示功能可以運用其他額外的電子零件使警示更為清楚明顯，其應用可依循使用的場所與環境進行調整。

參考臺北市消防局針對特定場所容留人數管制量，試著以班級教室 $9\text{m} \times 6.5\text{m} = 58.5\text{m}^2$ 、

自然教室 $8.5\text{m} \times 6.5\text{m} = 55.25 \text{ m}^2$ 的空間，模擬成不同類別的場域，試算出班級教室(表 4-3-1)和自然教室(表 4-3-2)容留人數管制量的最大值及最小值，假設學生人數 25 人、教師人數 1 人，以班級教室為例進行說明，其他部分面積計算： 58.5 m^2 - 學生桌面積($0.6\text{m} \times 0.4\text{m} \times 26$ 張，含備用桌) 6.24 m^2 - 老師辦公桌面積($1.65\text{m} \times 1\text{m}$) 1.65 m^2 - 講桌面積($1.2\text{m} \times 0.7\text{m}$) $0.84 \text{ m}^2 = 49.77 \text{ m}^2$ ，因此第一、二類的容留人數為 26 人 + 16 人($49.77 \text{ m}^2 / 3 = 16.5$ ，並無條件捨去取整數) = 42 人，第三類的容留人數為 33 人($49.77 \text{ m}^2 / 1.5 = 33.18$)。

表 4-3-1 班級教室容留人數管制量表

班級教室(學生桌 $0.6 \times 0.4 \times 26$ 張= 6.24 m^2 、辦公桌 $1.65 \times 1 = 1.65 \text{ m}^2$ 、講桌 $1.2 \times 0.7 = 0.84 \text{ m}^2$)			
第一、二類 (有設立固定席位)	A. 固定席位 師生人數	B. 其他部分面積 $\text{m}^2 / 3 (\text{m}^2 / \text{人})$	容留人數管制量 A+B
	26	16	42
第三類 (未設立固定席位)	其他部分面積 $\text{m}^2 / 1.5 (\text{m}^2 / \text{人})$		容留人數管制量
	33		33
班級教室容留人數(人)	≥ 42	34~41	≤ 33
RGB 模組設定燈號	紅燈	黃燈	綠燈

試算結果：班級教室容留人數管制量最大值為 42 人，最小值為 33 人

表 4-3-1 自然教室容留人數管制量表

自然教室 (講桌 $1.2 \times 0.7 = 0.84 \text{ m}^2$ 、實驗桌 $1.8 \times 0.9 \times 6$ 張= 9.72 m^2 、電腦桌 $1.2 \times 0.7 = 0.84 \text{ m}^2$)			
第一、二類 (有設立固定席位)	A. 固定席位 師生人數	B. 其他部分面積 $\text{m}^2 / 3 (\text{m}^2 / \text{人})$	容留人數管制量 A+B
	26	14	40
第三類 (未設立固定席位)	其他部分面積 $\text{m}^2 / 1.5 (\text{m}^2 / \text{人})$		容留人數管制量
	29		29
自然教室容留人數(人)	≥ 40	30~39	≤ 29
RGB 模組設定燈號	紅燈	黃燈	綠燈

試算結果：自然教室容留人數管制量最大值為 40 人，最小值為 29 人

1. 參數設置：

接續【階段三】所運用的程式碼，並額外建立 const int 常數變數 Red = 9、Green = 11，表示紅色 LED 燈訊號來源來自於 Arduino 第 9 號腳位，綠色 LED 燈訊號來源來自於 Arduino 第 11 號腳位。

```
const int Red = 9;
const int Green = 11;
```


2. void setup(開機時會執行一次)：

運用【階段三】的程式碼外，額外讓 Arduino 知道需運用 Red 以及 Green 兩變數作為訊號輸出使用。

```
pinMode (Red, OUTPUT);  
pinMode (Green, OUTPUT);
```

3. void loop(開機後會持續執行)：

運用【階段三】的程式碼外，以自然教室容留人數管制為例，假如(if)空間容留人數「大於或等於 0 時」，RGB 模組則顯示綠燈(0~255 表示其明亮程度，數字越大代表越亮，可依循使用環境與需求再進行微調)，表示此時室內空間人數偏少；假如(if)空間容留人數「大於或等於 30 時」，RGB 模組則顯示黃燈(由紅燈與綠燈混色後的結果)，表示此時室內空間人數偏多；假如(if)空間內部人數「大於或等於 40 時」，RGB 模組則顯示紅燈，表示此時室內空間人數過高，可能會存在著踩踏與呼吸困難等風險。

```
if (People >= 0) { //空間容留人數大於等於0，顯示「綠燈」  
  analogWrite (Red, 0);  
  analogWrite (Green, 255);  
}  
if (People >= 30) { //空間容留人數大於等於30，顯示「黃燈 (紅光加綠光)」  
  analogWrite (Red, 255);  
  analogWrite (Green, 255);  
}  
if (People >= 40) { //空間容留人數大於等於40，顯示「紅燈」  
  analogWrite (Red, 255);  
  analogWrite (Green, 0);  
}
```

(三)人流檢測器裝設支架製作

運用 Beambox Pro 雷射切割機與密集板進行板材設計，並利用 E18-D80NK 紅外線光電感測模組支架將其鎖於木板上固定，於板材上雕製 40cm 的實體量尺，並於每 2.5cm 設置提供支架裝設的孔洞，以快速更改感測器間的距離使用，因量尺的 16cm 至 24cm 間若裝設紅外線光電感測模組，會與下方的 Arduino 開發版相互牴觸，在節省板材使用及達到目標的取捨下，將此距離內的裝設孔洞移除以便 Arduino 開發版裝設。

因為【階段三】所製作的人流檢測器支架僅能放置於平面使用，因此我們再利用 3D 列印技術，繪製一個可以轉向的紅外線支架，如此一來則可以將人流檢測器吊掛於門口旁(可利用無痕掛勾或是螺絲鑽孔鎖在牆上等方式)，且不會佔據過多門寬空間，以利實際情境使用。

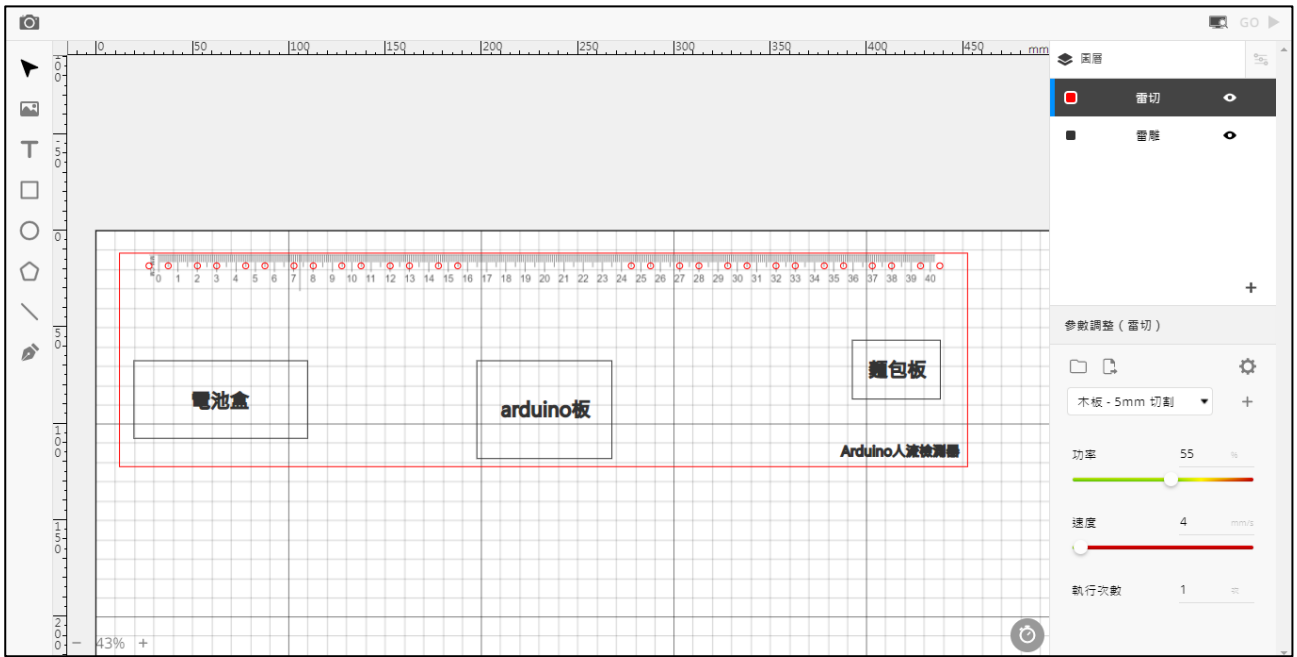


圖 4-3-5 人流檢測器雷切底板設計圖



圖 4-3-6 人流檢測器雷切底板成品

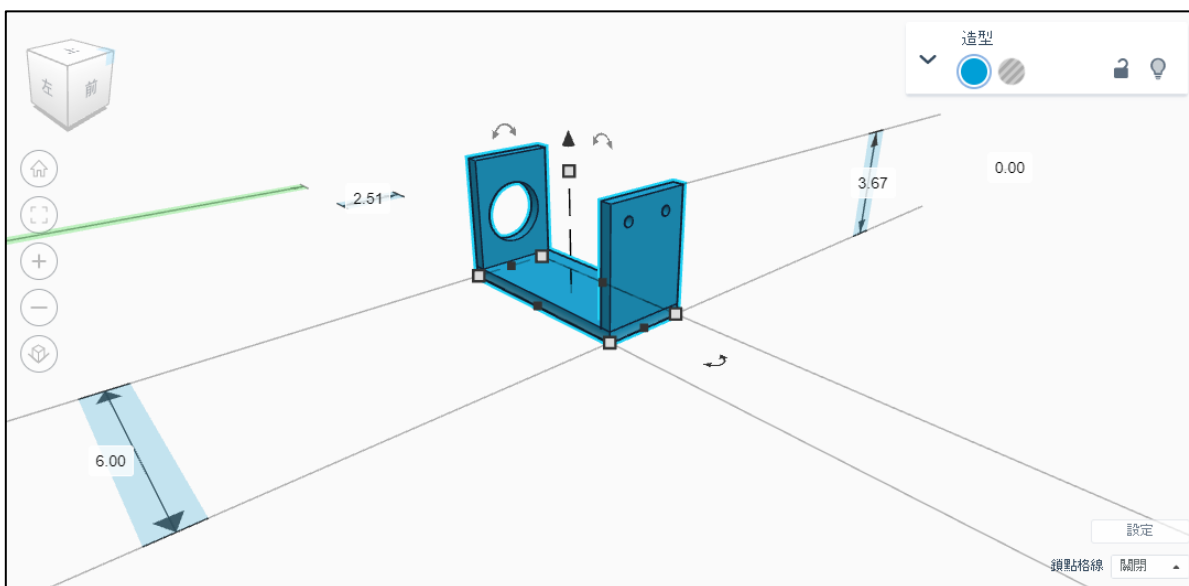


圖 4-3-7 紅外線感測器轉接支架 3D 列印尺寸圖

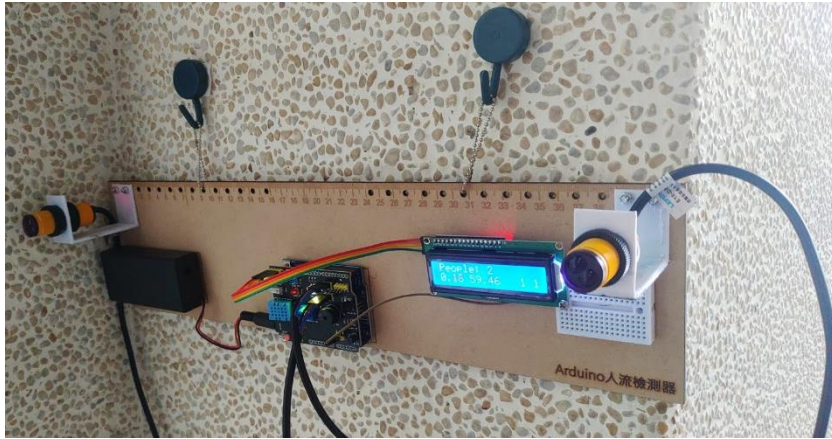
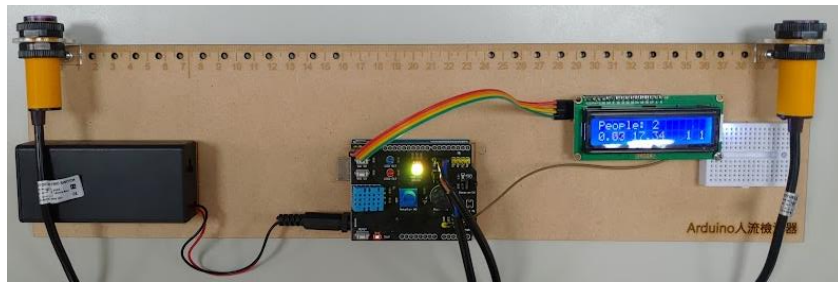


圖 4-3-8 人流檢測器門邊裝設成品圖



圖 4-3-9 紅外線感測器轉接支架 3D 列印繪製歷程圖

伍、討論

一、各類型應用與發展

一開始我們認為簡單的人流計算應該不是件很困難的事情，但實際開始進行研究後才發現需要考量的情況多面非常複雜，因此在過程中相當具有挑戰性及趣味性，透過組員及老師共同集思廣益，思考進出門口時可能會發生的各種情況，以及可以透過哪些方法設法去解決，在這一連串的思考過程中，加深我們對於程式設計的了解，也更能夠透過程式的編寫去思考 Arduino 的運作模式，並且激發出感測器在日常生活當中能應用的地方，以下為本研究針對日常生活應用方面所製作的統整：

1. 物體偵測與計數控管：

- (1) 可用於車流的控管，有時上下班尖峰時段車流量大、連假湧現旅遊車潮，造成橋上、隧道、匝道車輛壅塞排隊回堵狀況；如果能在出入口設置檢測器，計算車流量，便可提醒駕駛人改道，解決塞車問題。
- (2) 吊橋有安全結構方面的承載考量，要落實人數管制，以往多使用人工在入口處計算人次，若設置檢測器即時監控，能減少人力成本並避免承載過重的風險。
- (3) 停車場車流的計算，因有些是使用一塊大空地來設立臨時停車場，未劃停車格，車輛湧入過多造成停車場水洩不通，檢測器設立能控管車輛進出的數量。
- (4) 計算遊戲器材、餐廳排隊人數，可調節時段與疏散人潮，避免排隊過久。
- (5) 於學校的體育課或者社團訓練中，可以做為重複性動作執行時的計數工具，藉由動作的移動接觸感測器，能有效確保此動作的執行確實性，例如：伏地挺身、仰臥起坐、折返跑次數、揮棒次數等……。

2. 平均速率的測量：

- (1) 於學校的體育課或者社團訓練中，可以做為短距離衝刺平均速率測量(根據訓練強度確認是否有在規範平均速率之上)，或是透過 Wifi 系統將裝置無線化後，作為一百公尺的平均速率測量使用。
- (2) 本校有設立棒球隊，雖然紅外線感測器無法針對投球球速或擊球初速進行測量，但可以針對揮棒速率進行測量，作為打擊訓練的參考數據，若將紅外線感測器更改為超音波感測器，甚至可以用以測量揮棒角度並推測其可能軌跡。
- (3) 可以作為走廊奔跑的警示系統，設置其步行平均速率不得大於快走速率，若步行速率過高會使警報器產生作用，提醒學生勿在走廊奔跑以避免發生危險。

3. 時間的計算：

- (1) 球場、遊戲室、KTV、游泳池盥洗室等……使用時間的計算，在出入口設置感測器就能自動計算使用的時間，避免佔用過久的問題。
- (2) 在上學期間，於學校廁所出入口或是一些偏僻危險的空間，設置自動定時啟動系統，若學生在這些地方停留過久，外部警報器會發出聲響或是可透過 Wifi 系統將資訊傳給學務處或警衛等相關人員，以避免學生發生意外而延誤急救時間。

二、藍芽無線設備與行動載具連動

DX-BT19-A 藍芽模組的引腳 VCC 連接 5V 電壓，GND 接地，RX 為接收腳位連接 Arduino 擴展板的 RX 腳位，TX 為傳送腳位連接 Arduino 擴展板的 TX 腳位，當藍芽接通電後，LED 燈會閃爍表示進入配對狀態，當手機或其他電子設備連接上藍芽模組後，LED 燈會保持恆亮

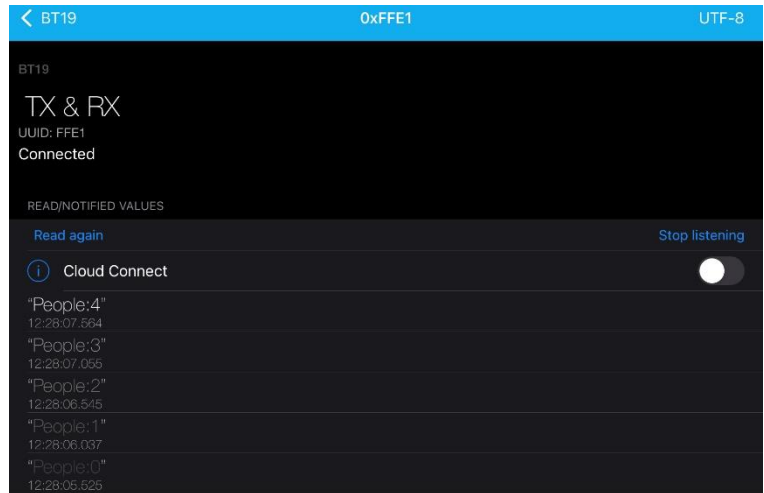


圖 5-2-1 LightBlue 軟體顯示畫面

狀態，表示此時可進行藍芽傳輸，運用 IOS 系統中的 LightBlue 軟體作為接收工具，可以將 Arduino 開發版中所得到的 人流訊息傳遞至 iPad 上，運用行動載具與雲端網路的連結，可以更進一步提升檢測器的功能。

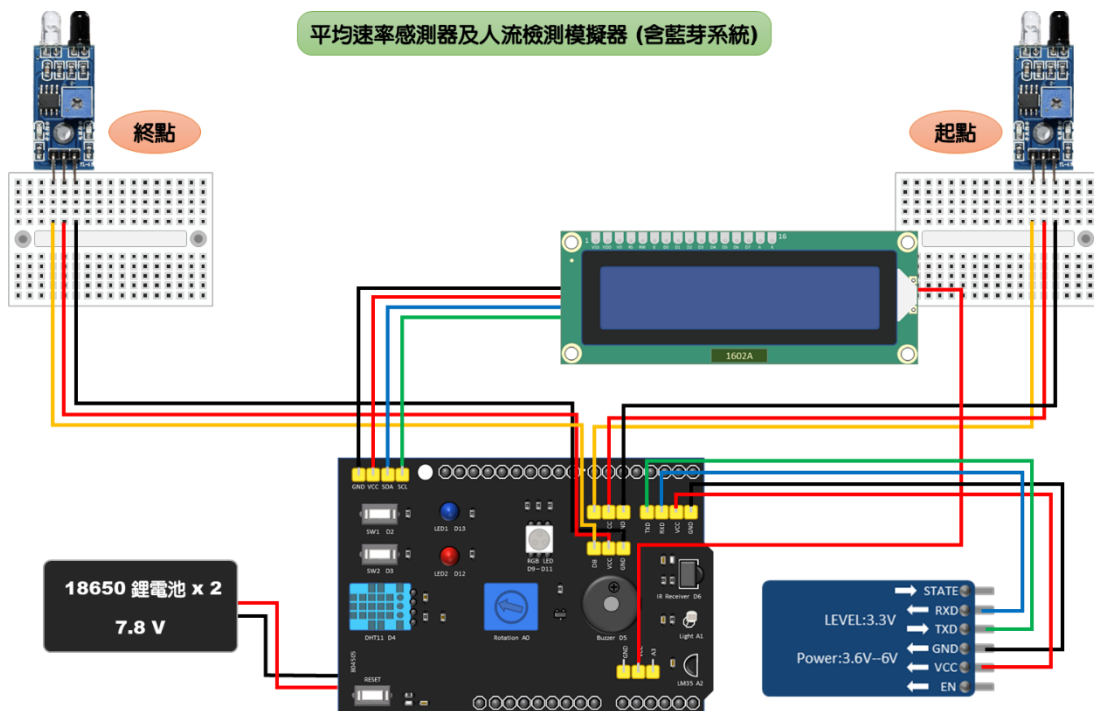


圖 5-2-2 Arduino 人流檢測器(含藍芽系統)裝置接線圖

陸、結論

透過 Arduino 本身所具備良好的發展性及參數修改彈性，我們可以確定利用 Arduino 開發板來做為平均速率測量裝置與人流檢測器的設計是可行的，以下為本研究之結論及未來發展的建議：

- 一、 透過紅外線感測器與 Arduino 程式的結合，可以達到測量平均速率之目標。
- 二、 運用雷射切割技術將平均速率測量裝置底板模組化，只要依循 Arduino 等電子材料的簡易裝設並安裝程式碼，即可完成 Arduino 平均速率測量裝置。
- 三、 可利用 Arduino 平均速率測量裝置作為基礎，轉換程式碼製作簡易人流檢測器。
- 四、 運用時間點運算的程式碼，並進行各項測試實驗，可使 Arduino 人流檢測器在日常生活中的複雜情境中使用(包含警示系統)。
- 五、 運用雷射切割以及 3D 列印技術將人流檢測器底板模組化，只要依循 Arduino 等電子材料的簡易裝設並安裝程式碼，即可完成 Arduino 人流檢測器。
- 六、 可透過藍芽系統的引入，將 Arduino 所偵測到的數值即時傳遞至 iPad 或其他電子設備，並透過 LINE 等網際網絡系統上傳雲端，則可達到遠距離觀看控管容留人數的目標。
- 七、 平均速率測量裝置可以進一步發展成斜面摩擦力測量裝置，透過改變斜面的角度、物體的重量，再運用數學公式計算出其斜面靜止或移動物體所受到的摩擦力大小。
- 八、 市售高價位的人流檢測器大多有引入 AI 掃描監測功能(人臉辨識或人體辨識系統)，未來研究能朝此方向作為研究主軸，結合 AI 自動掃描監測空間以及大數據分析等功能，藉由各項感測器與 AI 演算法自動分析空間內部的人流狀況，則可達到更準確計數人流的目標。
- 九、 未來可針對無線化系統進行更進一步改良，加入無線 Wifi 系統(例如：ESP32 WiFi)，則可使裝置的運用更為彈性，例如：可實際運用於 100 公尺的競賽測量，於起點架設一支起點感測器，於終點處架設一支終點感測器，可達到名次統計、秒數以及平均跑速測量。
- 十、 未來可針對複雜空間之人流檢測器進行發展，以梨泰院事件為例，此踩踏空間為一個狹小巷弄，且坡道傾斜，若在此空間的地板裝設重量壓力感測器，並在周圍裝設二氧化碳或溫度感測器，以評估該空間是否超過容留人數，避免發生人群推擠壓力的潛在危害。

柒、參考文獻資料

- 一、阿玉 Arduino 研究室：藍芽傳送字串至 Arduino。
取自：<https://sites.google.com/site/studyarduino/02lan-ya-yun-yong/03shuanga-chuan-xun>
- 二、使用手機透過藍牙控制 Arduino。
取自：<http://drho.club/2018/06/arduino-bluetooth-mobile-phone01/>
- 三、雙 A 學習：藍芽模塊 HC-06。
取自：<http://ee543.blogspot.com/2018/04/a180414hc06.html>
- 四、Arduino: Timer(計時器) 使用教學
取自：<https://crazymaker.com.tw/arduino-how-to-use-timer/>
- 五、人體紅外線感測器 (PIR Motion Sensor)HC-SR501 簡單實作
取自：https://blog.jmaker.com.tw/hc_hr501/
- 六、簡易版 Arduino 人流計數器
取自：http://miscp3.cnu.edu.tw/myblog/blogMessage.aspx?blog_id=293
- 七、FootfallCam 3D Pro2
取自：<https://www.footfallcam.com/zh-TW/Product/FootfallCam-3D-Pro2>
- 八、130-SCT15+人流量計數器
取自：<https://24h.pchome.com.tw/prod/DEACCI-A900FE54P3>
- 九、MET-CC999_場域人數顯示面板
取自：<https://www.thingiverse.com/>
- 十、TinkerCad
取自：<https://www.tinkercad.com/dashboard>
- 十一、〈韓國梨泰院踩踏 日本也曾發生類似「群眾雪崩」〉。《中央通訊社》。
取自：<https://www.cna.com.tw/news/aopl/202210300112.aspx>
- 十二、臺北市特定場所容留人數管制規則
取自：[臺北市政府消防局官網](#)