

## A. 文件

### a. 專題動機與願景

腳踏車方向燈的出發點是因為，有鑑於成大校園內腳踏車車流量大，加上下課時趕課導致大家疏於注意車況，時常有腳踏車車禍發生，如果想要改善校園內的交通，就應該比照機車、汽車，藉由方向燈來輔助用路人，降低校園內車禍的發生率。

#### 預期成果

腳踏車前面：兩個按鈕(左右各一)加上語音控制前面的點矩陣亮方向燈

腳踏車後面：煞車手把上裝上壓力感測器，分段顯示不同亮度的煞車燈

用 dot matrix 顯示方向燈和煞車燈

黃色的是按鈕控制方向燈

灰色的是架手機(語音辨識)的地方



## b. 系統功能與原理說明

LED Dot Matrix Display 用 SPI 的 SSPBUF 傳送資料給 dot matrix，dot matrix 是一行一行去接收值，利用視覺暫留讓人覺得一次是全部顯示

### 1. 煞車燈：

利用壓力感測器(FSR400)，控制腳踏車後方 LED Dot Matrix Display 顯示煞車圖案。

壓力感測器是用 ADC 將類比訊號轉成數位訊號，將感測到的值分成兩段，顯示不同亮度的煞車燈。用 timer2 和 interrupt 做閃爍的燈，閃 1 秒停 0.5 秒。

### 2. 方向燈：

利用聲控(HC-05)或按鈕來控制方向燈，顯示方向燈時發出聲音提示，並用 LED Dot Matrix Display 顯示方向圖案。

聲音的提示使用蜂鳴器來呈現，以 PWM 實現。定義音階的頻率，藉由修改 PR2 的值，產生不同的聲音。

按鈕控制方向燈是利用 interrupt 來達成，將 interrupt 設成 high priority，和語音辨識的 interrupt(low priority)作區別。

聲音控制方向燈的部分，我們設計一個應用程式，使其能與微處理器溝通，所以用 HC-05 此項藍芽模組來接收手機方的訊息，進而控制燈的明滅與聲音播放停止。此部分在程式碼中利用 UART 來呈現。

## c. 創意特色描述

1. 方向燈的部分，用跑馬燈(LED Dot Matrix Display)呈現右轉或左轉。
2. 有不同的方式可以選擇如何開啟方向燈，用語音的方式或是按按鈕的方式，也可以一起混用。
3. 煞車燈設計驚嘆號的號誌，並閃爍顯示，提醒後方的用路人。
4. 聲控使用 google 語音辨識 API，比起使用語音辨識模組更節省成本，且準確度較高。
5. 設計獨一無二的手機 APP，提供簡易的藍芽設定畫面，方便接上方向燈系統。
6. 一鍵開啟語音辨識，使用者只要在騎車前把開關打開，就能夠持續背景偵測，不需要理會手機的狀態。
7. 提供 APP 下載的 QR code，掃描後快速下載，馬上就可以使用
8. 方向燈和煞車燈都可以充電

9. 方向燈亮起後會播「小星星」，用來提醒騎士現在方向燈正在亮，要記得關閉。

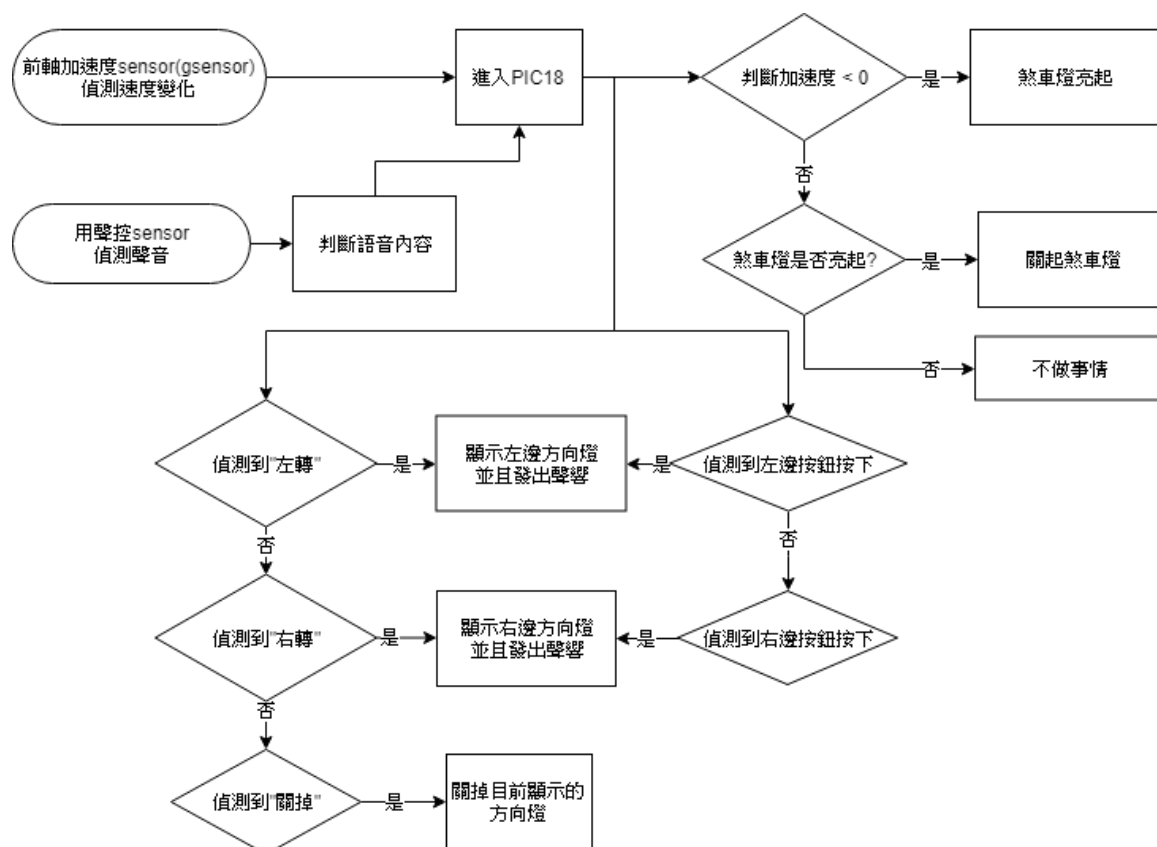
#### d. 系統使用環境及對象

- 適用環境
  - 戶外場域、裝在腳踏車上
- 適用對象
  - 腳踏車騎士
- APP 掃描 QR code



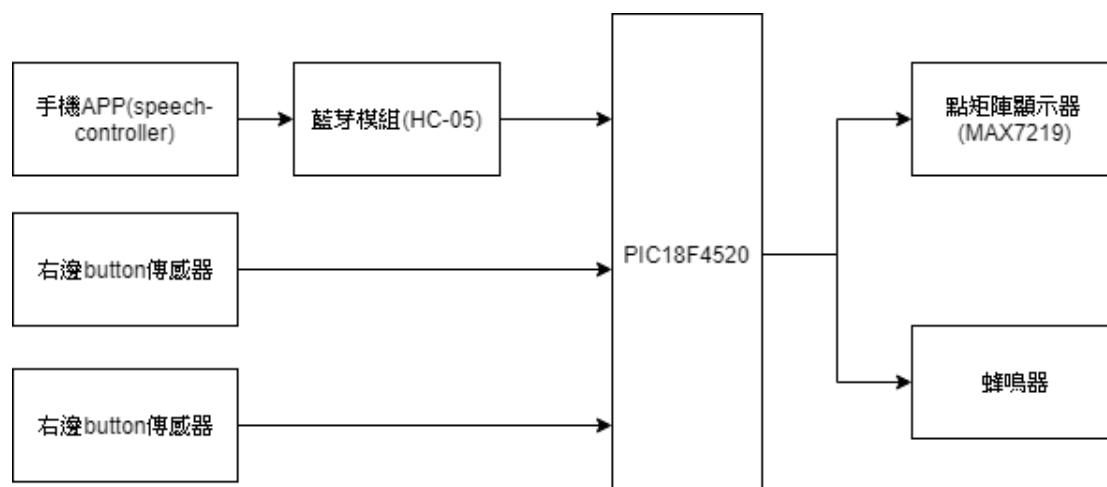
#### e. 系統完整架構圖、流程圖、電路圖、設計圖

系統 = 方向燈 + 煞車燈  
初步系統架構

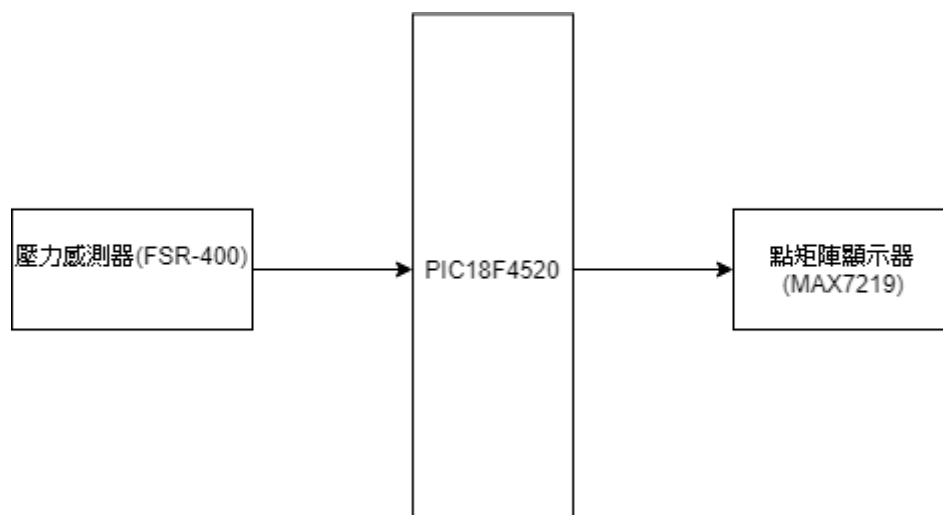


## 系統完整架構圖

- 方向燈



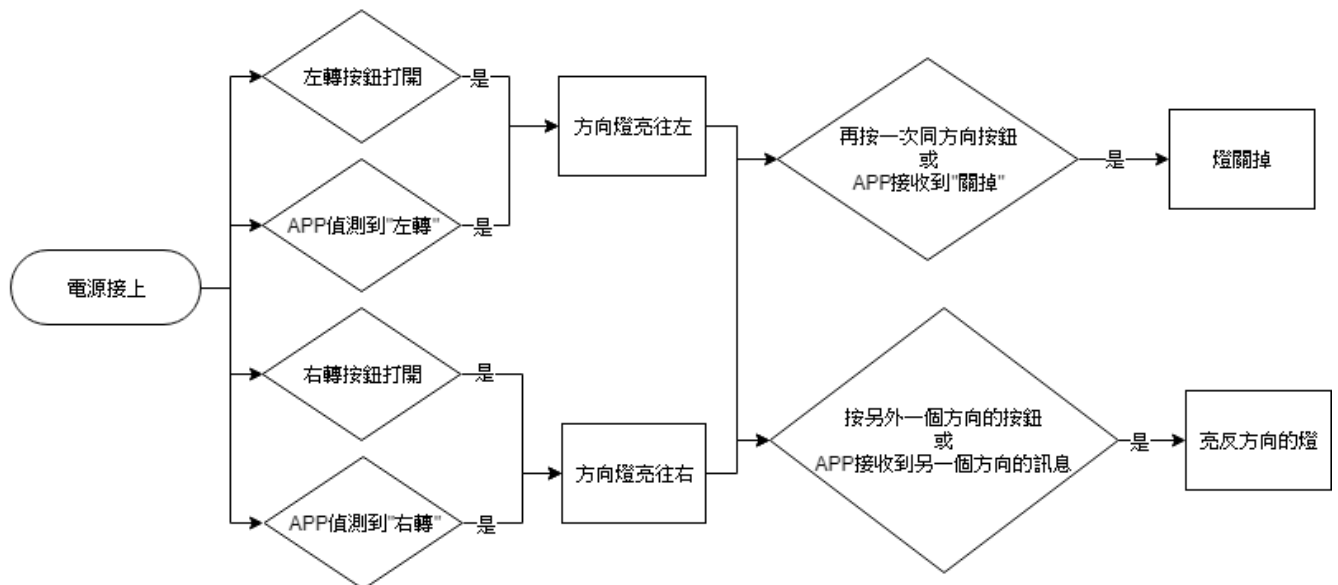
- 煞車燈



## 流程圖

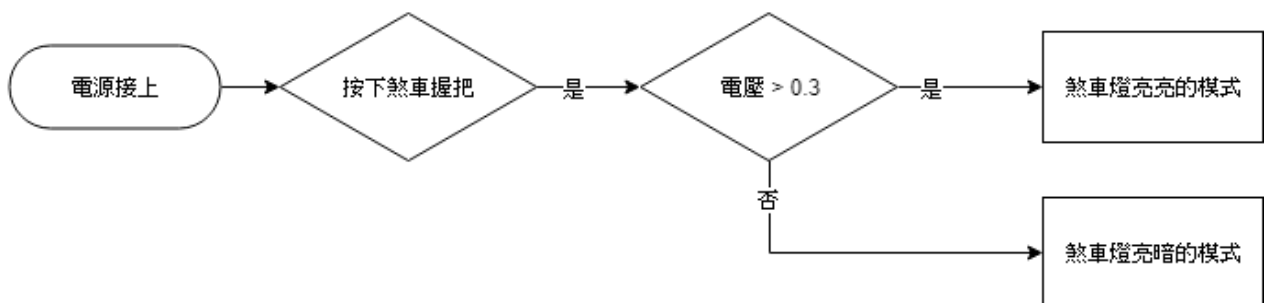
- 方向燈

### 前方點矩陣



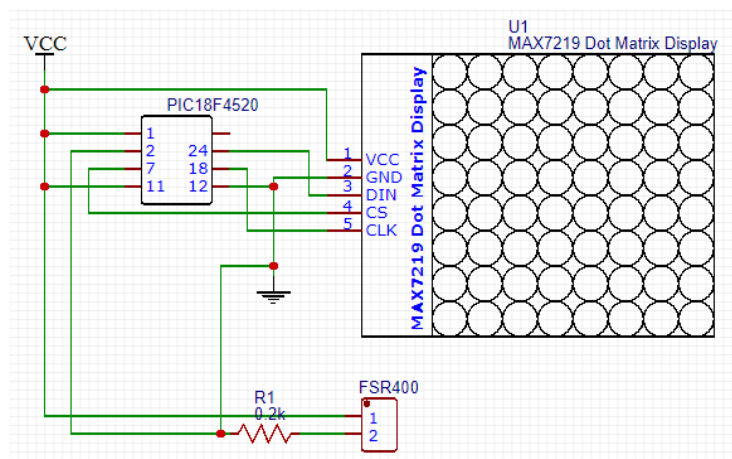
- 煞車燈

### 後方點矩陣

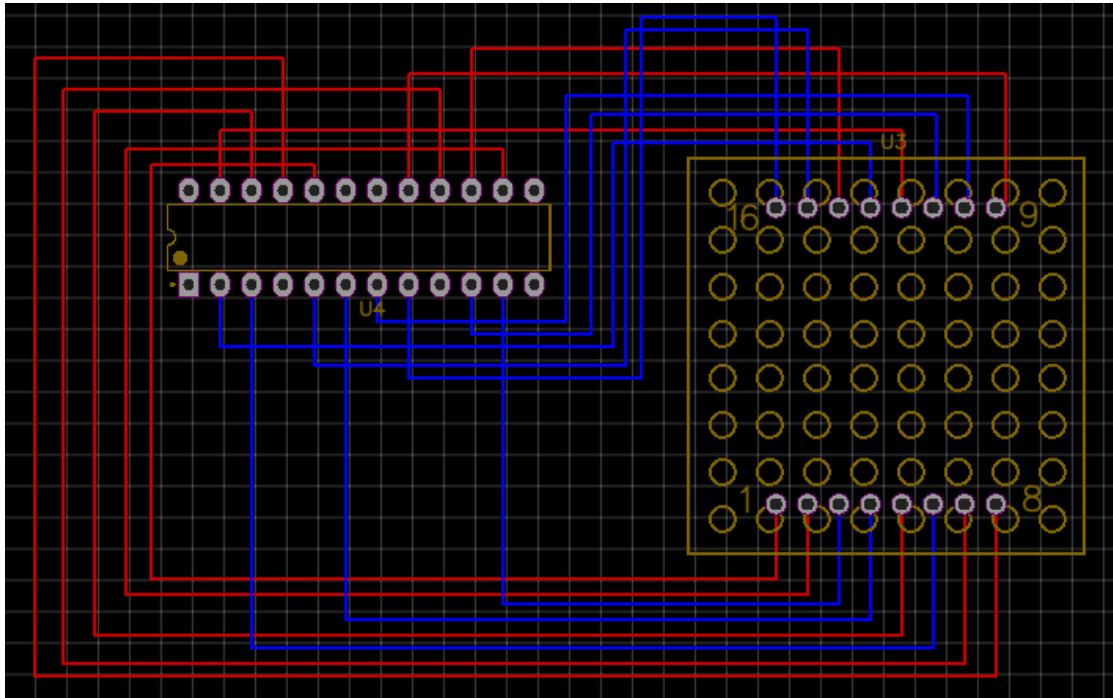


## 電路圖

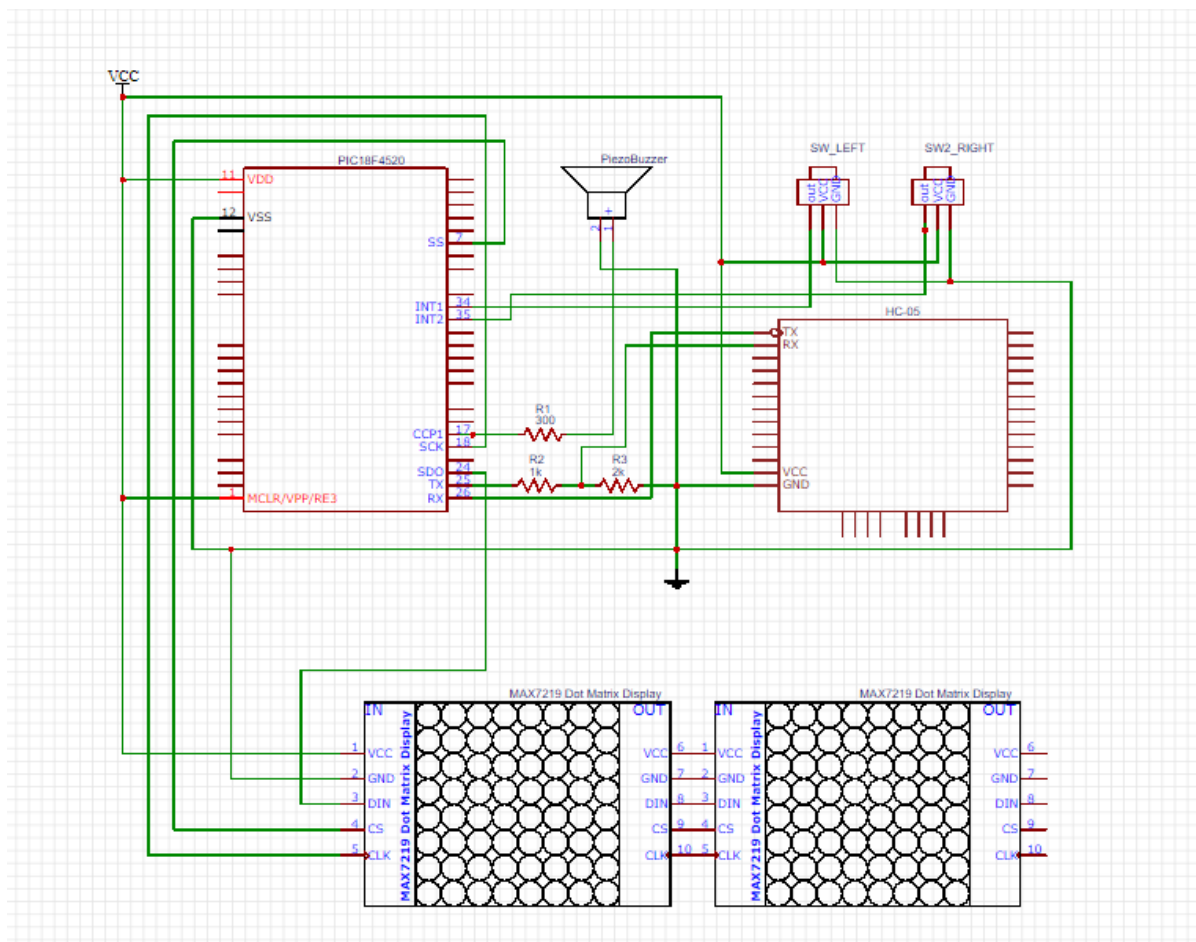
1. 壓力感測器接上 dot matrix



## MAX7219 & Dot matrix 接線圖



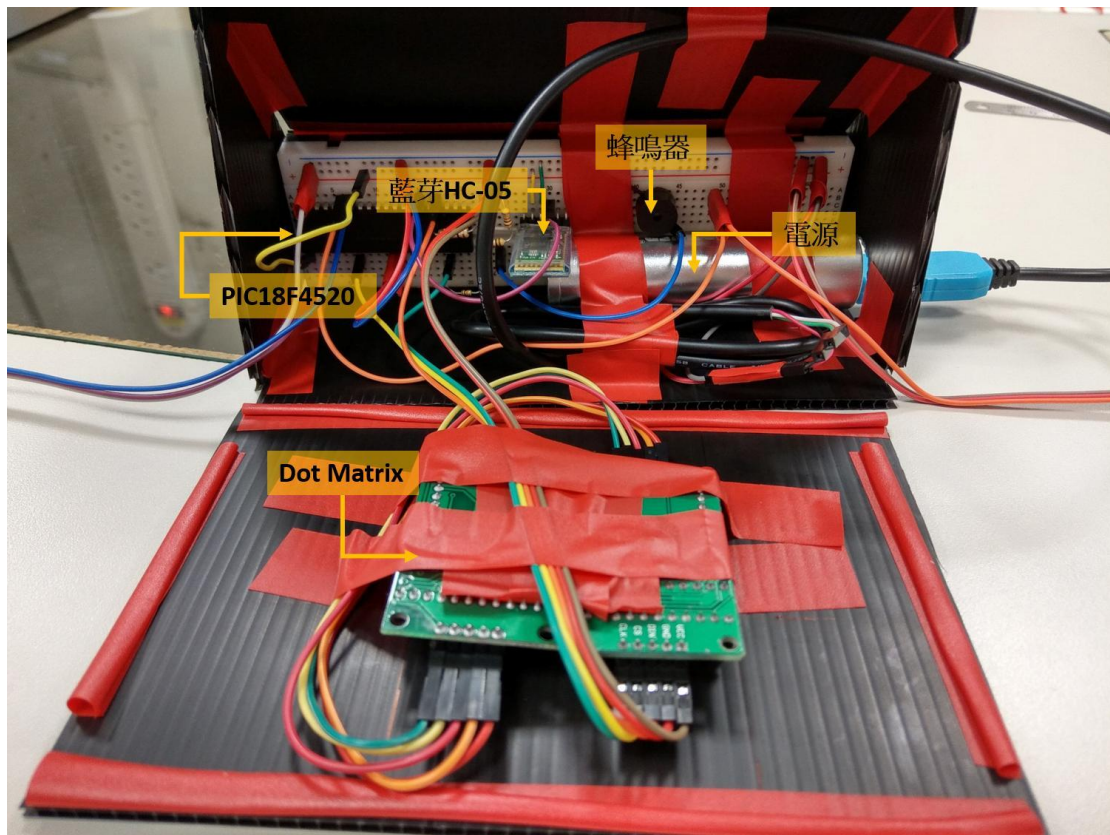
### 2. 按鈕和藍芽模組接上 dot matrix





## 設計圖

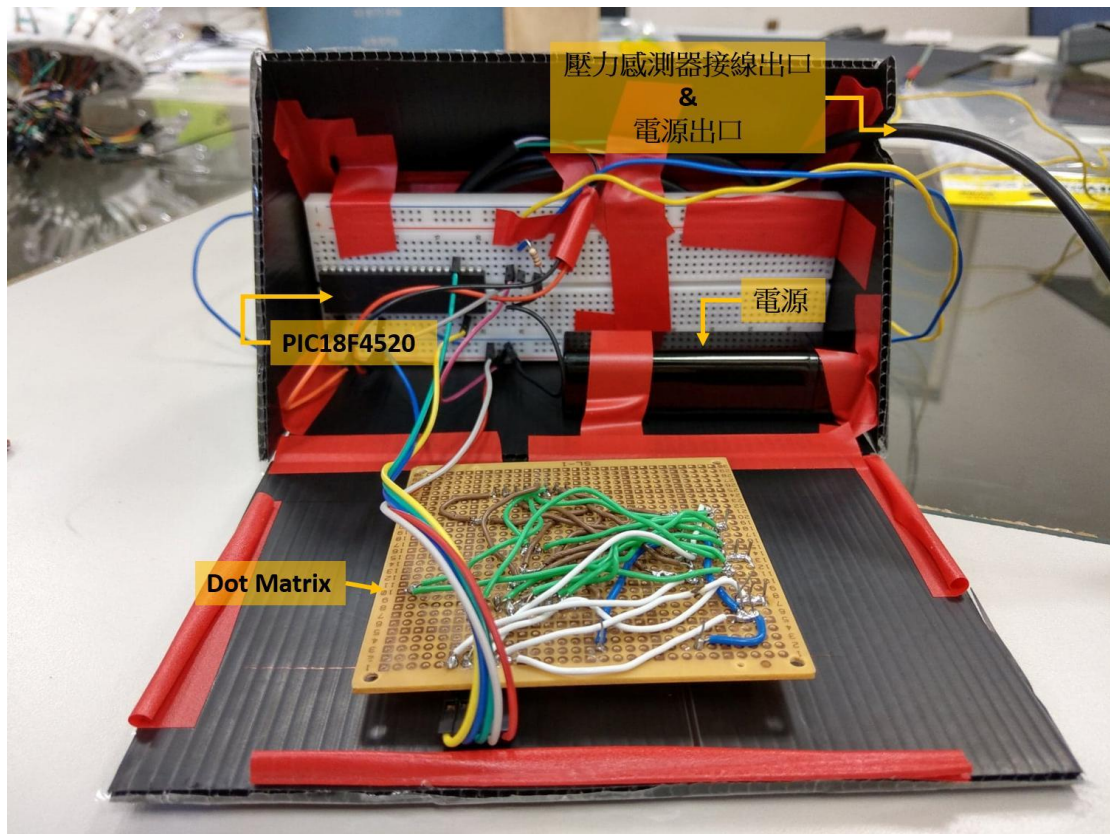
### 1. 方向燈



### 側邊

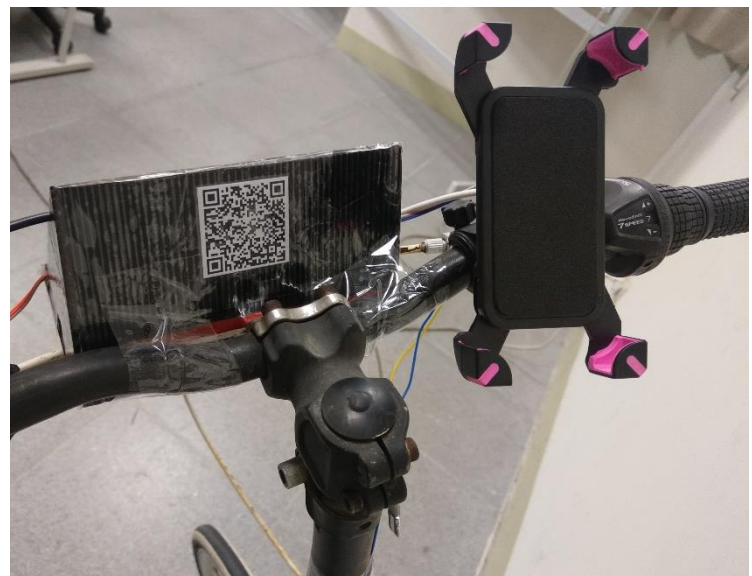


## 2. 煞車燈



### 設計圖

#### 1. 腳踏車前面

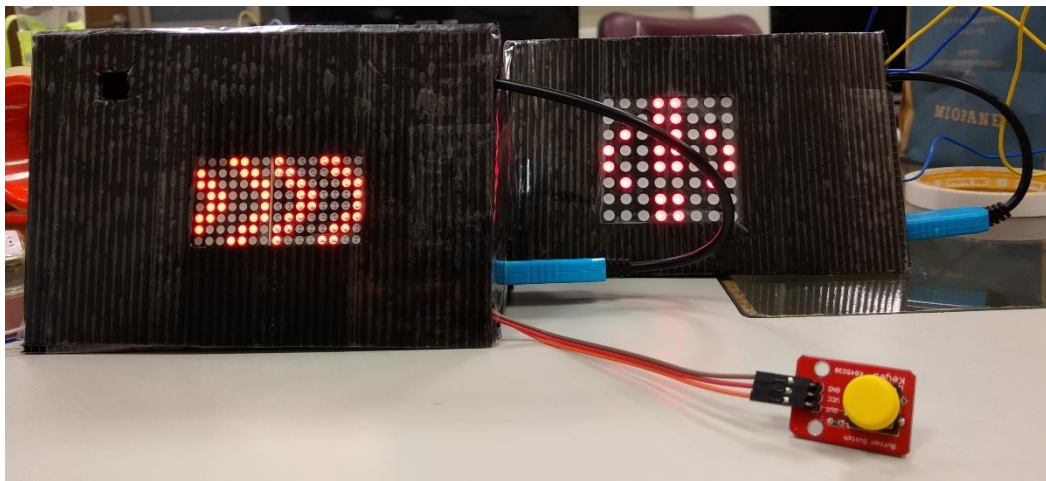




## 2. 腳踏車後面



## 3. 方向燈 & 煞車燈



## f. 系統開發工具、材料及技術

### 系統開發工具 **source code**

- 硬體部分
  - 開發環境：MPLAB IDE v5(XC8 compiler)
  - 程式語言：C 語言
- 軟體部分
  - 開發環境：Android Studio
  - 程式語言：Java

### 硬體元件

- 點矩陣顯示器模組\*2
- 自組的點矩陣顯示器模組\*1(點矩陣\*1、MAX2719\*1、電容\*2、電阻\*1)
- HC-05 藍芽模組\*1
- FSR-400 壓力感測器\*1
- 蜂鳴器\*1
- button 傳感器\*2

### 製作材料

1. 方向燈  
盒子(包住麵包板) 16.5cm\*4.5cm\*7.5cm  
前面方向燈開口 6cm\*3.2cm  
旁邊電源開口直徑 2.3cm  
旁邊按鈕開口(線從裡面接出來)  
黑色的紙包住按鈕
2. 煞車燈  
盒子(包住麵包板) 16.5cm\*5.5cm\*8cm  
接線到前面的煞車握把

### 技術

1. Interrupt  
按下按鈕或語音辨識接受到訊息時，皆會觸發 Interrupt(按鈕-high priority、語音 low priority)，開啟或關閉方向燈。

為了讓方向燈可以同時受按鈕或語音控制，將左右轉的 **flag** 設為全域變數，每進入一次 **interrupt** 就會更改 **flag** 的狀態，讓方向燈可以正確地閃爍。

煞車燈做閃爍的部分是用 **interrupt** 進去計算跑進去幾次，當跑進去的次數達到我們要的秒數的次數，就會把 **count** 歸零立一個 **flag** 起來，離開 **interrupt** 後，在迴圈裡判斷 **flag**，當 **flag** 立起來，就跑進去顯示煞車燈的 **function**。

## 2. Timer

煞車燈中的閃爍是透過 **timer2** 去計算時間，設定 **PR2**，當 **TMR2** 一直往上加到跟 **PR2** 一樣的時候，就會跑進去 **interrupt** 裡面。

## 3. PWM

使用 **CCP1**，定義音符的頻率，用來設定 **PR2** 的值(根據公式計算)；**duty cycle** 每次都設定為該頻率之週期的一半。當方向燈要亮起的 **flag** 立起後，以迴圈方式播放「小星星」的每個音符，不斷更改頻率，使蜂鳴器的播出的聲音有高低變化。關掉音樂的方式為調整 **CCP1M3** 和 **CCP1M2**，關掉 **PWM** 模式。

## 4. UART

將 **PIC** 的 **Rx** 設為 **input**，接收藍芽訊息，**baud rate** 設為 **9600**，使用 **low interrupt**，當 **RCIF** 為 **1** 時，讀取 **RCREG**，改變方向燈的 **flag** 和停止播放音樂。

## 5. ADC

設定 **ADCON1**，將 **AN0** 設成 **analog**，用壓力感測器改變電阻值，傳入 **AN0** 的電壓會因此改變，用電壓的變化去做判斷壓力的大小。

## 6. SPI

用單向傳輸。在 **MAX7219** 中，當 **CS** 設為 **low** 時，表示可以傳送資料進去，將位置及資料寫入 **SSPBUF**，透過 **SDO** 傳進 **MAX7219**，當資料傳輸完後，再將 **CS** 改回 **high**。

## g. 周邊接口或 Library 及 API 使用說明

- 共同

### <SPI>

#### SPI

```
signed char WriteSPI(unsigned char);
```

將 **SSPBUF** 的值傳入 **slave**。

## **max7219**

```
void init_MAX7219(void);
```

初始設定 MAX7219。

```
void MAX7219_1Unit(unsigned char , unsigned char);
```

將位址和資料傳入 WriteSPI(unsigned char)。

## **dot\_matrix**

```
void Draw(unsigned short *, int);
```

dot matrix 從第一行到第八行依序快速亮燈。

```
void TurnRight(void);
```

```
void TurnLeft(void);
```

方向燈每亮一次會播一個音，無限循環。

```
void Close(void);
```

call PWM1\_Stop()關掉音樂，將 MAX7219\_1Unit()傳送的資料設為 0，關掉方向燈。

```
void Stop(int)
```

設定 dot matrix 的亮度。

```
void TurnOff()
```

將 dot matrix 關掉。

- 煞車燈

## **<ADC>**

### **adc**

```
void ADC_Initialize(void);
```

初始設定 ADC 的輸入 pin、類比轉數位和清空 Register。

```
int ADC_Read(void);
```

讀取 ADC 輸入的值並回傳。

## **<Interrupt & Timer>**

### **timer**

```
void init_interrupt(void);
```

初始設定 interrupt，將 timer2 打開，並設定 PR2 為每 0.25 秒會觸發。

- 方向燈

## **<PWM>**



## sound

**void delay\_ms(unsigned int milliseconds);**

延遲多少毫秒，由傳入的參數決定。在此每次傳入 1000，延遲 1 秒鐘。

**void PWM1\_Init(long desired\_frequency);**

設定 PWM 頻率，將  $PR2 = Fosc / ((PWM \text{ 頻率}) * 4 * prescaler) - 1$ 。

**int PWM\_Max\_Duty();**

設定 PWM 最大的 duty frequency，即整個週期皆為 HIGH，回傳  $Fosc / (\text{目前音符的頻率}) * prescaler$ 。

**void PWM1\_Duty(unsigned int duty);**

設定 PWM 的 duty frequency，將 duty cycle 設為半週期。

**void PWM1\_Mode\_On();**

開啟 PWM 的模式，CCP1 模組選擇用 PWM。

**void PWM1\_Start();**

開始使用 PWM，把 Timer2 打開。

**void PWM1\_Stop();**

停止播放音樂，將 CCP1 模組關閉 PWM 模式。

**void playTone();**

播放音符，設定 PWM 頻率、duty，並播放目前的音一秒鐘。

## hc-05

**void HC\_05\_Init();**

初始化 HC-05，石英震盪器頻率設定為 8MHz，並初始化 UART，baud rate 為 9600。

**void receiveMsg();**

接收來自手機的訊息，設定左轉燈、右轉燈的 flag，並做出左轉、右轉、關掉等反應。

## <UART>

### uart

**void USART\_Init(long);**

初始化 UART，設定 Tx 為輸出、Rx 為輸入，baud rate 為傳入參數，在此設定為 9600，最後開啟 Tx、Rx 的 low priority interrupt。

**char USART\_ReceiveChar(void);**

回傳控制左轉、右轉、關掉的 flag。

**char USART\_ReceiveChar\_RCREG(void);**

回傳 RCREG，單純讀取接收到的藍芽資料。

## <Interrupt>

### dot\_matrix

```
void External_Interrupt_Init();
```

將 interrupt 設為 falling edge，設定 INT1 和 INT2。

## h. 實際組員之分工項目

### 預計

洪翊筑 F74051239 – 負責方向燈與按鈕控制部分

陳鈺潔 F74052201 – 負責方向燈與語音辨識控制部分

何品萱 F74056190 – 負責煞車燈及其控制部分

### 實際

洪翊筑 F74051239

負責方向燈(SPI)、煞車燈(SPI,interrupt,timer)、壓力感測器(ADC)、按鈕控制部分(interrupt)、蜂鳴器(結合方向燈)、組裝

陳鈺潔 F74052201

負責按鈕控制部分(interrupt)、語音辨識控制部分(UART、interrupt)、APP(JAVA,語音辨識,藍芽)、蜂鳴器(PWM,timer)

何品萱 F74056190

負責方向燈(SPI)、煞車燈(SPI)、壓力感測器(ADC)、按鈕控制部分(interrupt)、語音辨識控制部分(interrupt)、蜂鳴器(結合方向燈)、組裝

## i. 遇到的困難及如何解決

1. 語音辨識的模組使用 3.3V，但是 pic18f4520 適用電壓為 5V。
  - 解決方式：用 TL431 作為穩壓，接到 Vss；其他 pin 腳以串聯電阻方式降壓。
2. 語音辨識模組經過降壓後，依然無法在 pic18f4520 上使用。
  - 解決方式：改用軟體實現語音辨識，並用藍芽模組傳輸給 pic18。由手機接收使用者語音訊息後，將資料經由藍芽傳入 pic18 操控。
3. 點矩陣一直無法正常顯示。
  - 解決方式：確認 SPI 的初始設定與 buffer 的設定，補上點矩陣缺少電阻與電容。

4. 發現點矩陣的 pin 是亂的。
  - 解決方式：用焊接的方式將線焊死，避免接線混亂的問題，並且因為 pin 腳太雜亂，焊接上會需要大量時間，所以我們焊接了一塊，其他塊則是用已接好 MAX7219 的模組。
5. 三軸加速度模組使用 I2C，點矩陣顯示使用 SPI，導致 pin 腳重疊，無法實作。
  - 解決方法：將 I2C 改成用軟體實作，SPI 用硬體
6. I2C 軟體實作上設定複雜，一直無法成功。
  - 改成使用壓力感測器來判斷是否按下煞車，將 pin 腳留給點矩陣使用 SPI。
7. 把語音辨識與按鈕控制結合的時候，pic18 必須能夠接受語音的控制後又接受按鈕控制，或是先接受按鈕再接受語音。
  - 解決方法：透過設定 flag 的全域變數，每次進入 interrupt 去修改 flag 的設定，使方向燈能夠正確顯示。
8. 蜂鳴器與方向燈結合的時候，兩者的頻率不一致，且都須由 while 控制，不好合併。
  - 解決方法：將方向燈與蜂鳴器播放放在同一個迴圈控制，刪除音樂的 delay 用方向燈變換的時間當作音樂的 delay，使兩者能夠同時配合。

#### j. 預期效益與結語

透過普及腳踏車方向燈的安裝，讓在校園內的每個騎士都能及時反映現在自己行駛的狀況及方向，警示前後方的用路人，避免因突然的轉彎或煞車，所造成的擦撞，藉此改善交通亂象。

## B. 作品操作之 Demo 影片

實際騎乘腳踏車

[https://www.youtube.com/watch?time\\_continue=2&v=-bxiyGbQEzI](https://www.youtube.com/watch?time_continue=2&v=-bxiyGbQEzI)

煞車燈

[https://www.youtube.com/watch?time\\_continue=1&v=a9cO366AZsg](https://www.youtube.com/watch?time_continue=1&v=a9cO366AZsg)

## E. 會議記錄

Hackmd 連結：<https://hackmd.io/iZ9WvB8oTMSPQz1IGQ3vaQ>