

高雄市高英高級工商職業學校

Kao Ying Industrial Commercial Vocational High School

專題製作報告



機車傾倒求救裝置

指導老師： 林重仁 老師

科別班級： 汽車 科 三 年 3 班

座 號： 2、10、29、35、33

姓 名： 孫芳儀、夏振傑、賴文賢、林芳善、鐘弘智

中 華 民 國 109 年 12

機車傾倒求救裝置

摘 要

台灣有許多意外都是因為機車自摔或發生車禍但肇事逃逸，導致傷者沒有送醫而錯過了黃金救援時間。因此我們想製作這個機車傾倒求救裝置，讓機車一倒地就會發送信號至緊急聯絡人，讓機車騎士不會錯過黃金救援時間。

我們利用了三軸傳感儀+三軸加速度模組來量測機車是否有傾倒現象，如有傾倒再透過 LINE Notify 來對緊急聯絡人發送求救訊號，再藉由 NEO M8N 高精度 GPS 導航定位模組來讓緊急聯絡人能夠明確的知道受害者的地理位置，以達到能及時救援的效果。

我們需注意的不僅僅是機車的傾角而是在我們平常在騎車時就要特別的去注意自己車子的傾角最大值能達到多少，就會常常發生像這樣的交通意外發生。我們利用了 NodeMCU 32S(ESP-32S 開發版)的提醒，就可以減少許多機車騎士的交通意外。

今年新北 1 月至 4 月的交通事故死亡人數為 82 人，其中，機車事故死亡人數占總交通事故死亡人數的 6 成，顯示「肉包鐵」的機車車禍傷亡遠高於其他車種。機車交通事故死傷以自摔、自撞最多。遇到死亡車禍前往現場會勘發現，道路交通號誌完整，肇因多數為車速過快，或對於路況不熟悉。。103 年迄 9 月底止，本市機車死亡事故計 35 件，占整體死亡事故的 61%，其中機車發生自摔(撞)共 15 件，造成 15 人死亡，交通局表示，因機車機動性高，起步及轉向快，另外兩輪平衡較差，急煞車易失控自摔或自撞，加以機車肉包鐵的特性，其傷亡機率相對高，故提醒騎士除了遵守交通規則外，應具備防禦駕駛觀念，如勿超速、勿疲勞駕駛，行駛至路況不熟的地點應提高警覺。

目錄

摘要	i
目錄	ii
表目錄	iii
圖目錄	iv
壹、製作動機.....	1
貳、製作目的.....	2
參、文獻探討.....	2
一、NodeMCU 32S(ESP-32 開發版)	3
二、MPU-6050(GY-521) 三軸陀螺儀+三軸加速計感測模組.....	4
三、NEO M8N 高精度 GPS 導航定位模組	5
四、LINE Notify 權限	6
肆、製作方法.....	8
一、製作架構.....	8
二、製作材料.....	8
三、製作過程.....	9
伍、製作成果.....	12
一、成果示意圖.....	13
二、系統電路圖.....	13
三、成果說明.....	14
四、系統程式碼.....	20
五、成果討論.....	27
陸、結論.....	28
參考文獻	

表目錄

表 1 設備及器材.....	10
----------------	----

圖目錄

圖 1-1	機車意外事故統計表	1
圖 1-2	騎士自摔 送醫生命垂危	2
圖 3-1	32S(ESP-32S 開發版).....	3
圖 3-2		4
圖 3-3	MPU-6050(GY-521)三軸陀螺儀+三軸加速計感測模組	5
圖 3-4	NEO M8N 高精度 GPS 導航定位模組	6
圖 3-5		7
圖 3-6	LINE Notify	8
圖 3-7	通訊軟體的比較.....	8
圖 4-1	製作架構.....	8
圖 11		9
圖 12		9
圖 13		10
圖 14		10
圖 15		10
圖 16		10
圖 17		10
圖 18		11
圖 19		11
圖 20		11
圖 21		11
圖 22		11
圖 23		11
圖 24		11
圖 25		11
圖 26		12
圖 27		12
圖 28		12
圖 29		12
圖 30		13
圖 31		14
圖 32		15

機車傾倒求救裝置

壹、製作動機

台灣有許多交通事故，是因為自摔或者是被撞，但肇事駕駛者卻肇事逃逸，導致傷患沒有辦法及時送醫，因而錯過了最佳的救援時間，因此重傷甚至死亡。(圖 1-1)當看到類似的新聞時我們就在想，如果意外發生在我們身上或我們家人身上呢？(圖 1-2)我有一位朋友也是因為騎機車自撞上了鄉下路邊的樹，然而因為鄉下有很多路都很少人在經過，導致我那位朋友因為沒有及時送醫因此過世。所以我們才打算研究和製作「機車傾倒求救裝置」。利用我們在學校學習的專業技能及知識和後來專題指導老師教我們的 NodeMCU 32S(ESP-32S 開發版)的技術，將兩者結合在一起，製作出能夠在意外發生時發送訊號的這個裝置，使得意外發生後能盡快協助傷者接受治療。

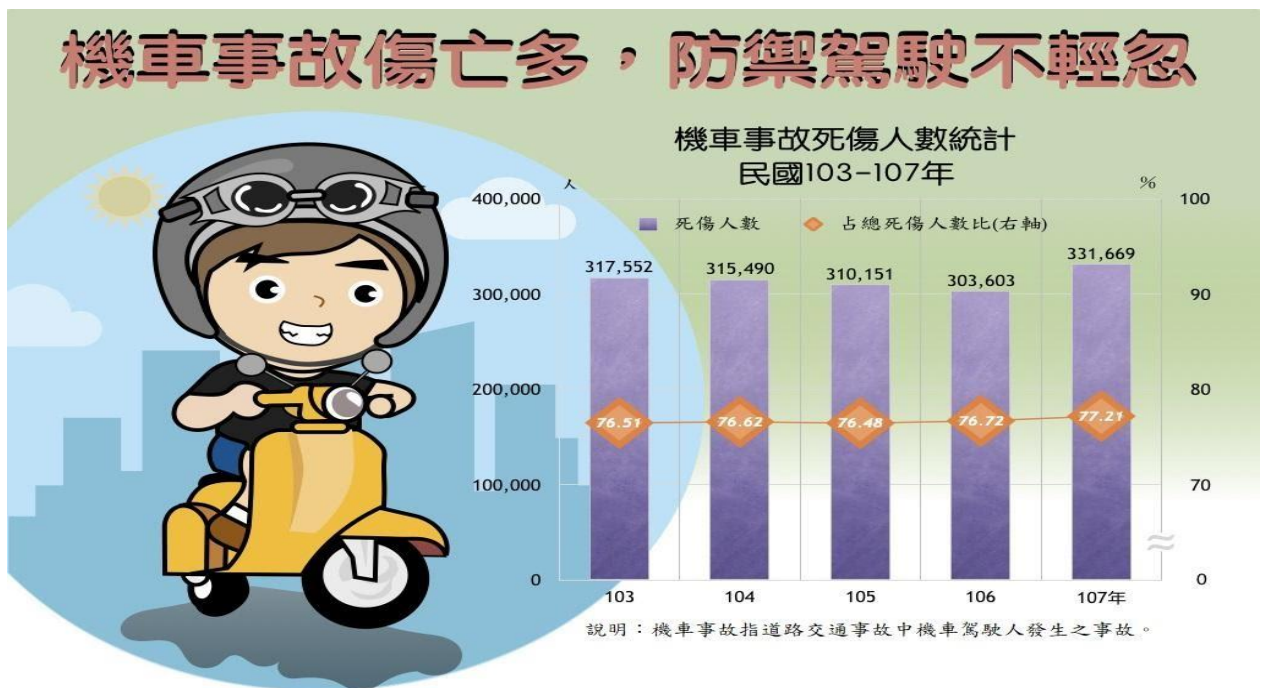


圖 1-1 機車意外事故統計表
內政部警政署。2020 年 12 月 20 日，取自
<https://reurl.cc/9XMI3Y>



圖 1-2：騎士自摔 送醫生命垂危

資料來源：2020 年 12 月 20 日，取自

https://www.youtube.com/watch?v=uQ_JAzLhUMA

貳、製作目的

- (一)學習撰寫及電路的設計。
- (二)學習三軸陀螺儀＋三軸加速計感測模組的使用方法
- (三)了解各項零件電路如何結合
- (四)了解 GPS 導航定位模組的原理
- (五)讓自摔的傷者不會錯過黃金救援時間

參、文獻探討

一、NodeMCU 32S(ESP-32S 開發版)

NodeMCU-32S 是基於 ESP32 芯片所設計的核心開發板。該開發板延續了 NodeMCU 1.0 經典設計，引出大部分 I/O 至兩側的排針，開發者可以根據自己的需求連接外設。使用面包板進行開發和調試時，兩側的標準排針可以使操作更加簡單方便。

為甚麼我們要選擇 NodeMCU 32S 呢？

因為 ESP32(圖 3-1)不僅內建有 WiFi 及低功率藍牙 BLE，可用腳位也擴增到約 26 根，大部分腳位同時支援類比及數位，內建霍爾、溫度、觸控感測器，主頻高達 260MHz 的雙核心 CPU，傳輸界面 SPI、I2C、UART 等強大的功能。

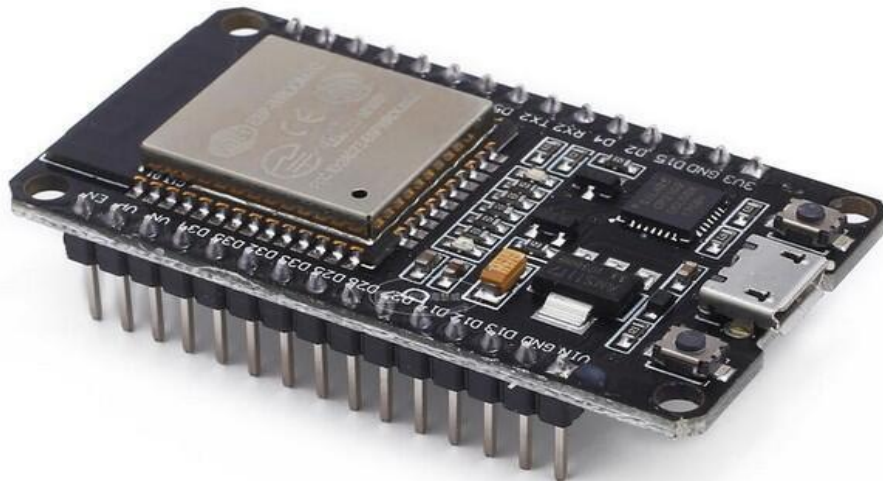


圖 3-1：32S(ESP-32 開發版)，

資料來源:2020 年 12 月 25 日取自

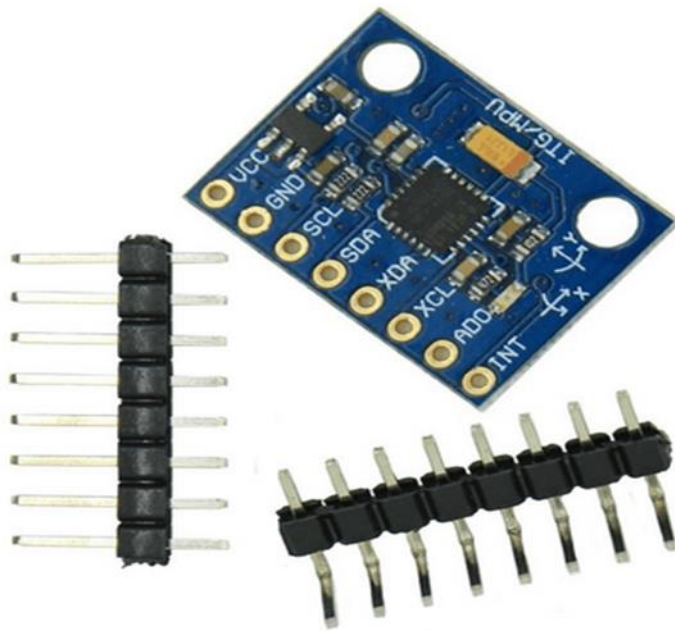
<https://reurl.cc/q8y7ng>

二、MPU-6050(GY-521)三軸陀螺儀＋三軸加速計感測模組

MPU-6050 為全球首例整合性 6 軸運動處理組件(圖 3-3)，相較於多組件方案，免除了組合陀螺儀與加速器時之軸間差的問題，減少了大量的包裝空間。

為什麼我們會選擇 MPU-6050 呢？

MPU-6050 整合了 3 軸陀螺儀、3 軸加速器，並含可藉由第二個 I2C 端口連接其他廠牌之加速器、磁力傳感器、或其他傳感器的數位運動處理(DMP: Digital Motion Processor)硬件加速引擎，由主要 I2C 端口以單一數據流的形式，向應用端輸出完整的 9 軸融合演算技術 InvenSense 的運動處理資料庫，可處理運動感測的複雜數據，降低了運動處理運算對操作系統的負荷，並為應用開發提供架構化的 API。



MPU-6050(GY-521)三軸陀螺儀＋三軸加速計感測模組

資料來源：2018 年 11 月 19 日，取自

<https://reurl.cc/2gXnLn>

三、NEO M8N 高精度 GPS 導航定位模組

Ublox NEO M8N 高精度 GPS 導航定位模組(圖 3-4)自帶信號放大支援 Arduino 與 樹莓派 Ublox NEO M8N 高精度 GPS 導航定位模組 採用 NEO M8 系列 多功能 u-blox M8 GNSS 模組, 多功能 GNSS 模塊採用不同型號, 易於製造。同時接收最多 3 個 GNSS (GPS, Galileo, GLONASS, BeiDou), 業界領先 -167 dBm 導航靈敏度。安全和完整性保護, 支持所有衛星增強系統。先進的干擾和欺騙檢測, 產品型號可滿足性能和成本要求。

為什麼我們要選擇 NEO M8N 高精度 GPS 導航定位模組呢?

因為自帶 SMA 天線接口, 可外接各種有源天線、自帶 EEPROM, 可保存配置信息, 只需要一條 USB 線連接電腦, 即可使用上位機軟件, 自備串口功能, 無需另購其他串口模塊、兼容 5V/3.3V 電壓, 適應各類單片機、支持樹莓派, 直插 USB 使用, 無需驅動、支持固件升級、自帶信號放大器

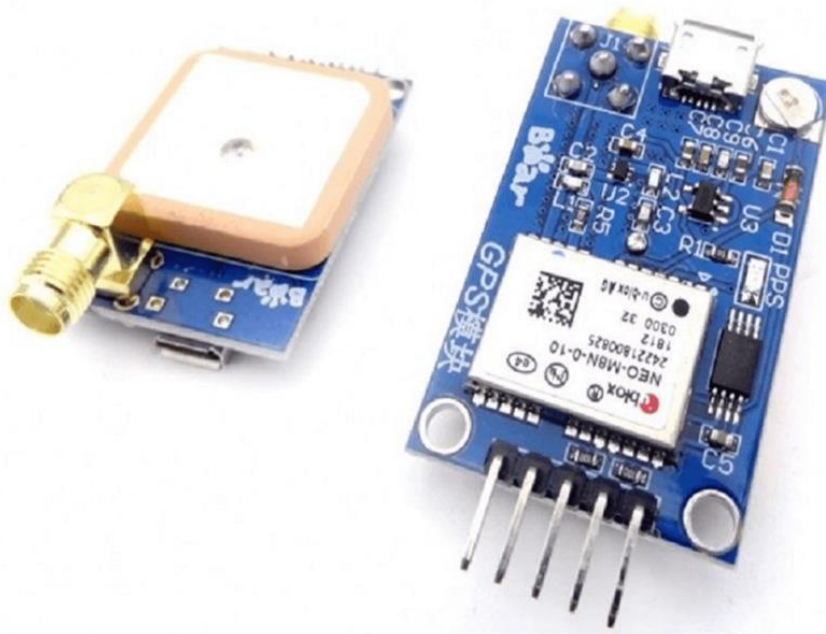


圖 3-3 NEO M8N 高精度 GPS 導航定位模組

資料來源:2020 年 12 月 20 日, 取自

<https://www.mirotek.com.tw/>

四、LINE Notify 權限

LINE Notify(圖 3-6)是一個「訊息推播」服務，但跟一般開發者所理解的推播不太一樣，他的運作方式是這樣的：

LINE 官方有一個特殊的 LINE Notify 帳號，這是一個 LINEBot 機器人，他是一個專門用來發送訊息的帳號。

你不需要事先加入這個帳號，只要你有訂閱任何一個 LINE Notify 通知服務，他就會主動發訊息給你。任何 LINE 用戶都可以成為服務提供者，可以發送推播給任何人，而且真的完全不用任何費用！任何 LINE 用戶都可以註冊一個以上的服務提供者，每個服務提供者通常代表著某種目的的通知服務。所有的服務提供者都必須由 LINE 用戶主動訂閱通知，該用戶才能收到推播通知，而且用戶隨時可以取消通知。一律由官方的 LINE Notify 帳號發出訊息，而不是由 LINE 官方帳號發出訊息。

為什麼我們會選用 LINE Notify 來作通訊軟體呢？

因為 LINE 這個 App 在智慧型手機上的普及率為 92% ，遠超過其他通訊軟體(圖 3-7)，並且我們這個裝置就是為了能夠及時求救，所以



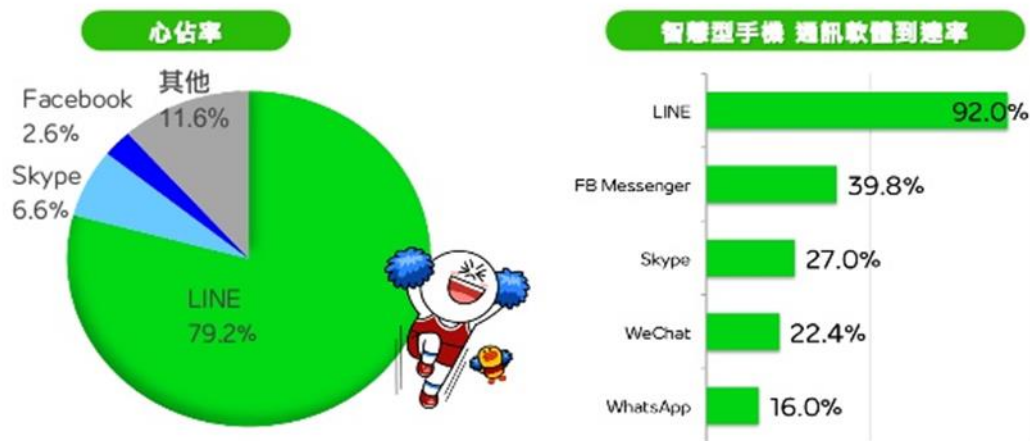
圖 3-4 LINE Notify

資料來源：2021 年 1 月 3 日，取自

https://notify-bot.line.me/zh_TW/

LINE 台灣強勁表現

有接近80%的使用者提到通訊軟體時第一個會想到LINE。
且LINE在智慧型手機上的普及率為92%，遠超過其他通訊軟體。



Source: Online Research by INSIGHTXPLOER, INC. (Feb 2014/May 2014) 5

通訊軟體的比較

資料來源：2021 年 1 月 5 日

肆、製作方法

一、製作架構(圖 4-1)

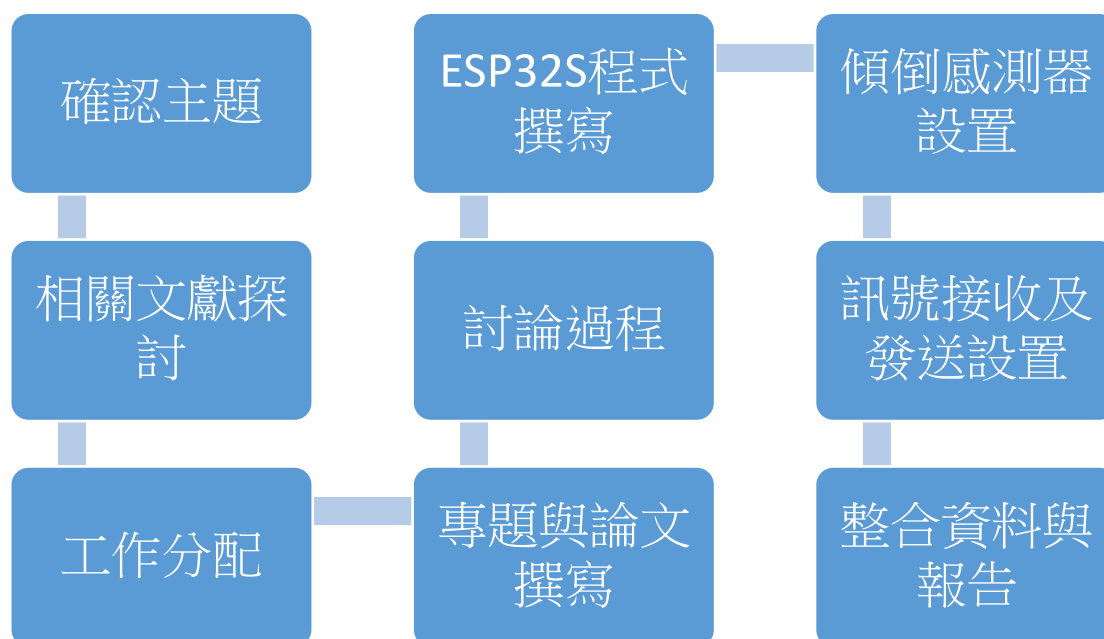


圖 4-1：製作架構
(資料來源：研究者繪製)

二、製作材料

表 4-1:設備及器材

名稱	規格	數量	名稱	規格	數量
ESP-32S 開發版	NodeMCU 32S	1 片	三用電表	通用型	1 台
電腦	Windows10	1 台	焊錫	通用型	3 條
杜邦線	通用型	50 條	吸錫器	通用型	1 支
3D 列印	CR-10	1 台	鐸槍	40W	1 支
麵包板	通用型	1 個	十字起子	通用型	1 支
機車模型	FiatYamaha 1：10	1 台	LED 發光二極體	通用型	1 顆
木板		1 個	傳輸線	android	2 條
行動電源	10050mAh	1 台	三軸陀螺儀+三軸加速計感測模組	MPU-6050(GY-521)	1 片
高精度 GPS 導航定位模組	NEO M8N	1 組	GPS 外部天線	GPS Antenna	一組

三、製作過程

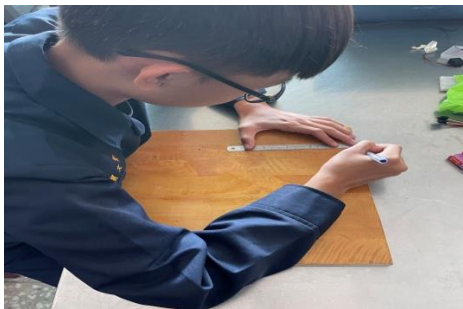
(一)3D 電腦繪圖:包含踏板、本體固定架設計

我們利用 F u s i o n 3 6 0 進行踏板總成的設計，為了增加踏板止滑性，特別在設計時於踏板上加上了稜格紋以增加摩擦力，另外在踏板與伺服馬達連接處採分離式設計，讓維修時拆裝更加快速與方便。

1.		2.
圖 4-2：		圖 4-3：

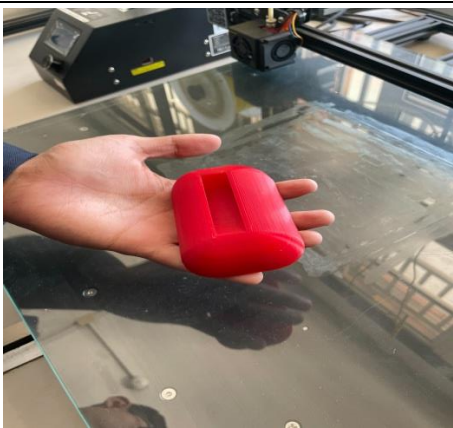
3.	
圖 4-4：	

(二)製作機車模型底座

1.		2.
		
圖 4-5：		圖 4-6：

3.伺服馬達安裝	4.迴旋踏板總成安裝
----------	------------

	
圖 4-7：伺服馬達安裝	圖 4-8：迴旋踏板總成安裝

1.線路配置	2.線路焊接
	
圖 4-9：線路配置	圖 4-10：線路焊接
3.線路整合	4. Arduino 控制總成安裝
	
圖 4-11：線路整合	圖 4-12：Arduino 控制總成安裝

1.壓力感知器安裝

2.超音波感知器安裝

		
圖 4-13：壓力感知器安裝		圖 4-14：超音感知器安裝

(四) 斷電延遲開關：在點火開關關閉後，系統是會立即斷電的，為了讓系統有時間可以偵測點火開關訊號及後座的乘坐情形並啟動繼電器，所以才裝設了斷電延遲開關。

1. 斷電延遲開關設定	2. 斷電延遲開關作動測試
圖 4-17：斷電延遲開關設定	圖 4-18：斷電延遲開關作動測試

(六)機車智能迴旋踏板裝置完成品

1. 裝置完成品 1	2. 裝置完成品 2
------------	------------

圖 4-19：裝置完成品 1		圖 4-20：裝置完成品 2

伍、製作成果

一、成果示意圖

圖 5-1：成果示意圖
(資料來源：研究者繪製)

二、系統電路圖

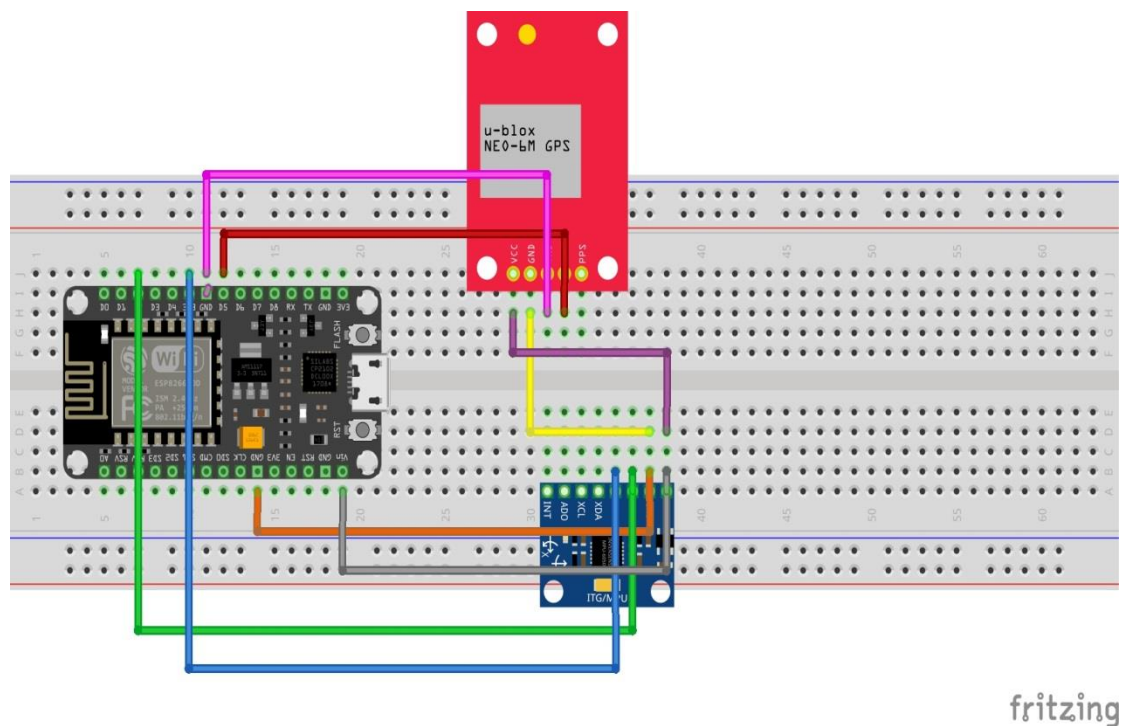


圖 5-2：系統電路圖
(資料來源：研究者繪製)

三、成果說明

本專題製作是用 ESP32S 開發版、控制核心，利用 MPU-6050(GY-521)三軸陀螺儀＋三軸加速計感測模組、

(一)點火開關之 ON/OFF 為系統判定供應電源的條件。

1.點火開關 OFF 無電源供應。

2.點火開關 ON，電源供應使讓系統作動。

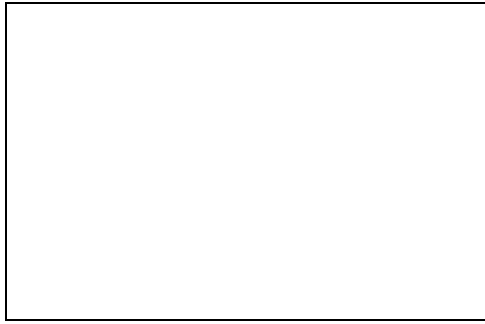


圖 5-3 無電源供應

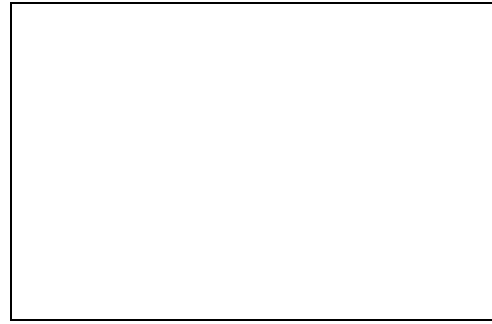


圖 5-4 有電源供應

(二)後座乘坐情形判定及踏板作動情形：

- 1.機車後座沒有放置任何東西，也沒有乘客，此時壓力及超音波感知器的偵測值都未達設定值，機車踏板不伸出。(如圖 5-5、圖 5-6)

(1)壓力感知器未達設定數值

(2)機車迴旋踏板不伸出

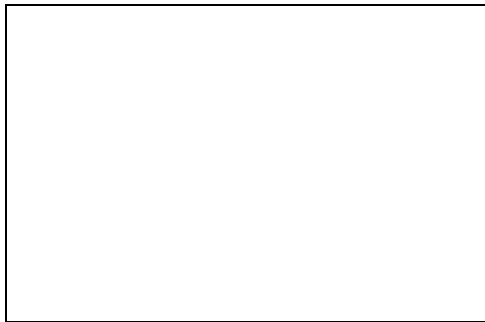


圖 5-5 壓力指示燈不作用

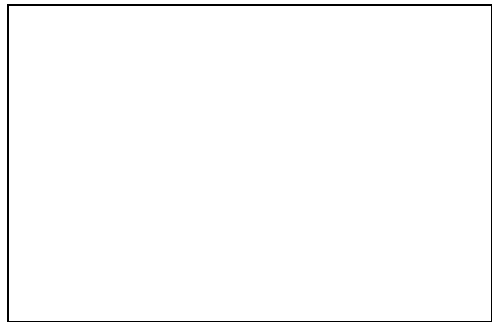


圖 5-6 機車踏板不伸出

2.當機車後座放置重物時，只有壓力感知器達到系統設定值，此時踏板不伸出。(如圖 5-75、圖 5-8)

(1)壓力感知器達設定數值，但超音波未偵測到訊號



圖 5-7 壓力指示燈亮起

(2)機車迴旋踏板不伸出



圖 5-8 機車踏板不伸出

3.當後座有乘客時，壓力及超音波感測器都達到系統設定值，此時踏板自動伸出。(如圖 5-9、圖 5-10)

(1)滿足所有條件



圖 5-9 滿足所有條件

(2) 機車踏板伸出



圖 5-10 機車踏板伸出

4.當駕駛者將乘客載至目的地，後座的乘客下車後，此時壓力與超音波感知器所感知到數值判斷為後座無人乘坐，踏板自動收回。(如圖 5-11、圖 5-12)

(1)點火開關開啟

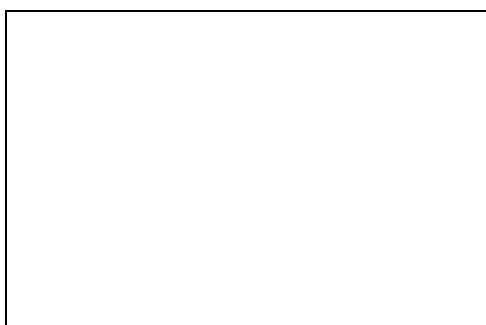


圖 5-11 點火開關開啟

(2)後座乘客下車，機車踏板自動收回

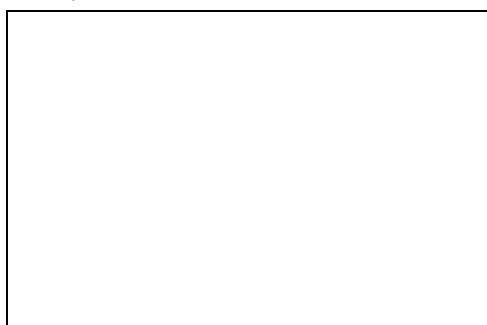


圖 5-12 機車踏板自動收回

5.當駕駛者和乘客一同下車，此時點火開關關閉，因斷電延遲開關的作動，系統會有五秒的時間偵測後座的乘坐情形，當壓力與超音波感知器所感知到數值判斷為後座無人乘坐後，機車踏板將自動收回，而在機車踏板收回後，系統會自動將電源切斷，以防止能源的浪費。(如圖 5-13、圖 5-14)

(1)點火開關關閉

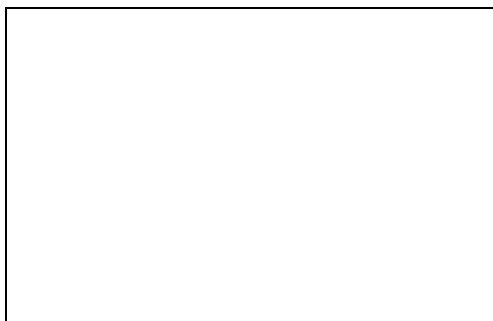


圖 5-13 點火開關關閉

(2)機車踏板自動收回

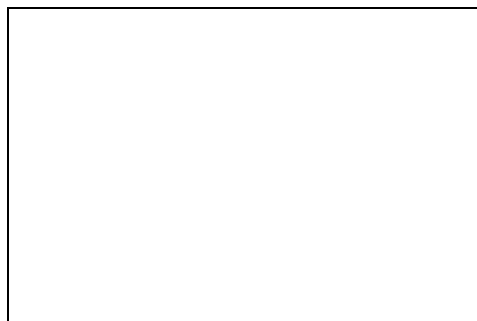


圖 5-14 機車踏板自動收回

(三)系統人性化設定：

我們除了想讓系統具有智慧化的特性外，同時也想讓系統更加的貼近於人性的需求，例如當前座的駕駛者暫時下車購物或繳費時，大多會將鑰匙抽離(如圖 5-15)，假如後座乘客仍在車上，而機車踏板收回，將會使乘客的雙腳處於無處可放的悶態，於是我們將系統設定為，若後座還有乘客乘坐，即使鑰匙抽離，機車踏板也不會收回(如圖 5-16)，直到後座乘客離開，踏板才會自動收回，同時系統也會自動將電源切斷，以防止能源的浪費。

1.機車鑰匙抽離

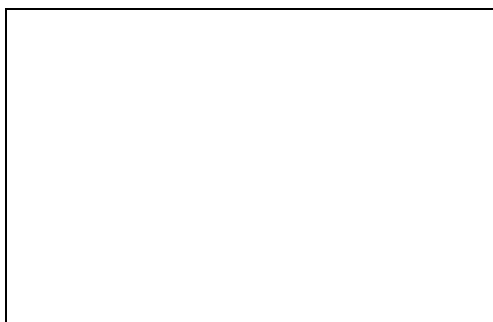


圖 5-15 機車鑰匙抽離

2.繼電器作動踏板不收回

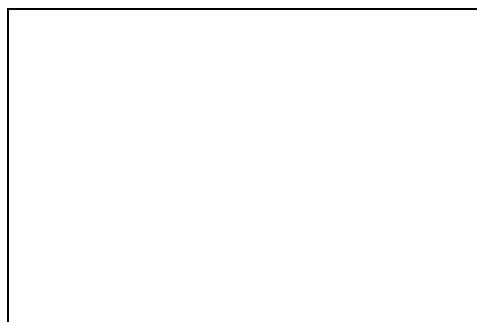


圖 5-16 繼電器作動踏板不收回

四、系統程式碼

```
#include <TridentTD_LineNotify.h>
#include "GY521.h"
#define SSID          "0952343885"
#define PASSWORD      "1234567890"
#define LINE_TOKEN    "MXHucwJr7Pru0LHWuOxWpemEgJEweO0aGbJmQU09d4N"
GY521 sensor(0x68);
uint32_t counter = 0;
int speedpin = 0;
TaskHandle_t Task1;
TaskHandle_t Task2;

float rpm_time = 0, rpm_time_1 = 0, rpm_time_value = 0;
float rev=0;
float rpm;
int rpm_TTL = 0, car_state = 0;

void setup()
{
    pinMode(16, INPUT); //轉速腳
    pinMode(0, OUTPUT); //LED 腳
    Serial.begin(115200);
    Wire.begin();

    while (sensor.wakeup() == false)
    {
        Serial.print(millis());
        Serial.println("\tCould not connect to GY521");
        delay(1000);
    }

    Serial.println(LINE.getVersion());
```

```

WiFi.begin(SSID, PASSWORD);
Serial.printf("WiFi connecting to %s\n",  SSID);
while(WiFi.status() != WL_CONNECTED) { Serial.print("."); delay(400); }
Serial.printf("\nWiFi connected\nIP : ");
Serial.println(WiFi.localIP());

LINE.setToken(LINE_TOKEN);

sensor.setAccelSensitivity(0);
sensor.setGyroSensitivity(0);
sensor.setThrottle();
Serial.println("start...");

sensor.axe = 0;
sensor.aye = 0;
sensor.aze = 0;
sensor.gxe = 0;
sensor.gye = 0;
sensor.gze = 0;

xTaskCreatePinnedToCore(
    Task1code,    /* Task function. */
    "Task1",      /* name of task. */
    10000,        /* Stack size of task */
    NULL,         /* parameter of the task */
    1,            /* priority of the task */
    &Task1,       /* Task handle to keep track of created task */
    0);           /* pin task to core 0 */

delay(500);

//create a task that will be executed in the Task2code() function, with priority 1 and executed on
core 1
xTaskCreatePinnedToCore(
    Task2code,    /* Task function. */
    "Task2",      /* name of task. */
    10000,        /* Stack size of task */
    NULL,         /* parameter of the task */
    1,            /* priority of the task */

```

```

        &Task2,      /* Task handle to keep track of created task */
        1);          /* pin task to core 1 */

    delay(500);

    Serial.println();
    Serial.println("          Accelarometer                      Gyrometer          ");
    Serial.println("-----");
    Serial.println(" Xraw          Yraw          Zraw          Xnorm          Ynorm          Znorm  ");
}

void loop()
{
}

void Task1code( void * pvParameters ){
    for(;;)
    {
        Serial.println();
        Serial.println("          Accelarometer                      Gyrometer          ");
        Serial.println("-----");
        Serial.println(" Xraw          Yraw          Zraw          Xnorm          Ynorm          Znorm  ");
        sensor.read();
        for(int i = 0; i <= 11; i++)
        {
            int ax = sensor.getAngleX();
            int ay = sensor.getAngleY();
            int az = sensor.getAngleZ();
            int gx = sensor.getGyroX();
            int gy = sensor.getGyroY();
            int gz = sensor.getGyroZ();
            int t = sensor.getTemperature();
            Serial.print(ax);

```

```

Serial.print("  ");
Serial.print(ay);
Serial.print("  ");
Serial.print(az);
Serial.print("  ");
Serial.print(gx);
Serial.print("  ");
Serial.print(gy);
Serial.print("  ");
Serial.println(gz);
delay(1000);
}

if(gx >= 100 && ax >= 50)
{
    car_state = 1;
    LINE.notify("右邊撞車");

}

```

//-----

```

    if(gx >= 100 && ax <= -50)
    {
        car_state = 1;
        LINE.notify("左邊撞車");

    }

```

//-----

```

if(gy >= 100)

```

```

{
//達成 gy >= 100 條件後的做動

}

//-----

    if(gz >= 100)
    {
//達成 gz >= 100 條件後的做動

    }

```

```

if(car_state == 1)//撞車
{
    digitalWrite(0,HIGH);
}

if(car_state == 0)//未撞車
{
    digitalWrite(0,LOW);
}

```

```

    }
}

void Task2code( void *pvParameters )
{
    for(;;)
    {
        if(digitalRead(16) == 1 && rpm_TTL == 1)
        {
            rpm_time = millis();
            rpm_time_value = rpm_time - rpm_time_1;
            rpm_time_1 = rpm_time;
            rpm = 1000 / rpm_time_value * 60;
            Serial.println(rpm);
            rpm_TTL = 0;
        }
        if(digitalRead(16) == 0 && rpm_TTL == 0)

```

```

    {
        rpm_TTL = 1;
    }
}
}

```

五、成果討論

(一)機械式踏板和電子式踏板的比較

(二)電子式踏板感測的缺點解決方法

陸、結論

(一)_

(二)

(三)

(四)

參考文獻

- (1) 陳佳新(2019)。LINE Notify API。彰化縣:奇步應用。
- (2) 內政部警政署(2016)。機車意外事故統計表。高雄市:內政部警政署。
- (3) 楊永盛(2019)。大學新生騎車下坡自撞分隔島命危。蘋果新聞網。
- (4) 渥瑪數位(2020)。ESP32 程式設計(基礎篇)。出版:渥瑪數位有限公司。
- (5) 鄭正元(2017)。3D 列印積層製造技術與應用。出版:全華書館。
- (6) 傅忠云(2015)。MPU6050 介紹。中國:南京航空航天大學自動化學院。
- (7) 安守中(2003)。GPS 全球衛星定位系統入門。出版社:全華圖書
- (8) [模組] 三軸陀螺儀與加速規 Gyro & Accelerometer。研發紀事
- (9) MPU6050 陀螺儀/加速器換算方式。Jesse.s biog