

## 車輛防盜警報與即時影像

組員：蔡宜庭 張家維 薛羽倫 張廷禧

執行期間：109 年 7 月至 110 年 6 月

### 一、介紹：

#### 1. 緣起

起初我們在剛進入到大學的時候，都曾夢想擁有一台機車，看著電影或學長姐回高中時跟我們訴說那些青春的故事，其中一定會有騎著機車環島或出去玩，讓我們對著人生中第一台機車有著無限美好幻想，經過多重努力之下，入手了第一台寶貝機車，在現在這個一台機車起碼都八九萬塊起跳的社會，對大學生來說就是寶物，而台灣的環境，又對機車不友善，不是一格機車格硬要擠進兩台車，就是普遍大家對別人或自己機車都不在乎，造成一台新車常常過沒幾天就傷痕累累。

所以我們注意到了初概念發源於在車壇且掀起熱烈討論的、搭載於特斯拉電動汽車上的「哨兵模式」，其用途為當車主上鎖並離開車輛時，車載電腦將持續監控車輛周遭是否有異常行為進而啟動錄影或是向車主發送警告。

我們希望透過此專題計畫，讓類似功能能夠套用於其他不具備此功能的車輛，並具備低成本、高泛用性等特色。

#### 2. 初步構想

在經過數次討論後，設計方向確定為車載裝置、中央伺服器以及手機用戶端三部分，其中車載裝置扮演著收集車輛周遭狀況（透過鏡頭及加速度計等感測裝置）並回傳伺服器的角色，而伺服器則為串連使用者的手機 app 及車載裝置的橋樑。

### 3. 系統架構

#### A. 車載裝置端

**功能用途：**透過多軸感測器及鏡頭收集車輛周遭動態，並回傳伺服器。亦包含低電壓保護機制。

#### B. 實作細節：

1. **供電：**由於汽機車電系以 12V DC 為主流，故以一 Buck Converter 模組降為 5V DC 以供應整個電路。又以一耗電極低的 Arduino Nano 結合 MOSFET 開關在

電瓶低電壓時關閉主電路供電。

2. **感測器：**以 Raspberry Pi 原廠推出的 Raspberry Pi Camera V1.3 五百萬畫素相機模組提供車輛周遭影像、以及 Seeed Studio 推出的 GrovePi+ 模組連接六軸加速度計提供車輛動態資訊。

3. **主控制器：**使用 Raspberry Pi 3 B+ 單板電腦及 Raspbian 作業系統（原廠基於 Debian 開發的作業系統，由於其對 Pi 的相容性最好故選擇其為開發平台）作為整個車載端的大腦，統整各控制器內容並回傳伺服器。其中影像回傳使用了在 Pi 上支援硬體編碼加速的 GStreamer 函式庫，將影像編碼為

H.264 後透過 RTMP 協定與伺服器通訊。六軸加速度計部分則為持續監測數值，一旦各軸測值加總超過預先設定之閾值，即會將異常資料上傳伺服器

#### C-1. 伺服器端

**功能用途：**儲存 rpi 上傳的資料及作為影像串流中繼站

**實作細節：**訊息回報部分為將 rpi 傳來之六軸加速度計資料存在 MySQL 資料庫

中，並且撰寫一 php 程式將異常發送至用戶端。

影像串流部分則使用 Docker 作為平台，並執行第三方現成的 nginx-rtmp 映像以便於部署，其工作原理為使用 nginx 及其 RTMP plugin 搭建串流伺服器，使車載裝置端的 rpi 能利用 GStreamer 將影像串流至伺服器，並提供用戶端觀看。

## C-2. 用戶端

**功能用途：**在手機上有 app 可以操作

**實作細節：**我們設置了一個登錄帳號密碼的首頁(圖 3)，這次我們事先預設好帳號密碼(圖 1)，若輸入錯誤密碼，就無法登陸(圖 4)，且我們一個帳戶綁定一台 rpi，這樣的好處是若今天同時擁有多台車，只需換帳號即可在同一台手機上得到兩台車的即時保護。

```
public class MYTLogin extends Activity
{
    Button mylogin;
    Button myregister;
    Button abouts;
    EditText account;
    EditText mypwd;
    TextView loginErrorMsg;

    String accounts[]={"tsai890409","chang890117","tsai881218","hsues890329","dan0111002"};
    String passwords[]={"00000000","00000001","00000002","00000003","00000004"};

    public static String IPAddress;

    public static MYTLogin rent;

    String saccount, spwd, small, sreg, sphone, sspwd;

    static String accountId;
    static String groupId;

    String myaccountId;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.mylts);

        rent = this;

        Resources res = getResources();
        IPAddress = (String) res.getString(R.string.IPAddress);
        if (android.os.Build.VERSION.SDK_INT > 5) {
            StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);
        }

        account = (EditText) findViewById(R.id.account);
        mypwd = (EditText) findViewById(R.id.mypwd);

        mylogin = (Button) findViewById(R.id.mylogin);

        //check login
        mylogin.setOnClickListener(new View.OnClickListener() {
            for (int i=0; i<accounts.length; i++)
            {
                if (account.getText().toString().equals(accounts[i]) && mypwd.getText().toString().equals(passwords[i]))
                {
                    Intent app = new Intent("packageContent:MYTLogin.this, LocationVAP.class");
                    startActivity(app);
                    return;
                }
            }
            Toast.makeText(view.getContext(), "使用者帳號或密碼", Toast.LENGTH_LONG).show();
        });
    }
}
```

圖 1

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.mylts);

    rent = this;

    Resources res = getResources();
    IPAddress = (String) res.getString(R.string.IPAddress);
    if (android.os.Build.VERSION.SDK_INT > 5) {
        StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);
    }

    account = (EditText) findViewById(R.id.account);
    mypwd = (EditText) findViewById(R.id.mypwd);

    mylogin = (Button) findViewById(R.id.mylogin);

    //check login
    mylogin.setOnClickListener(new View.OnClickListener() {
        for (int i=0; i<accounts.length; i++)
        {
            if (account.getText().toString().equals(accounts[i]) && mypwd.getText().toString().equals(passwords[i]))
            {
                Intent app = new Intent("packageContent:MYTLogin.this, LocationVAP.class");
                startActivity(app);
                return;
            }
        }
        Toast.makeText(view.getContext(), "使用者帳號或密碼", Toast.LENGTH_LONG).show();
    });
}
```

圖 2

圖 1 和圖 2 為部分程式碼

## D. app 介面與功能



圖 3



圖 4

登陸之後，即可持看使用者位置 (HERE)，圖片中 HERE 會在海上是因為此台電腦無 gps 定位，故在開啟模擬器時無法取得使用者位置，在手機上則可以正常使用。

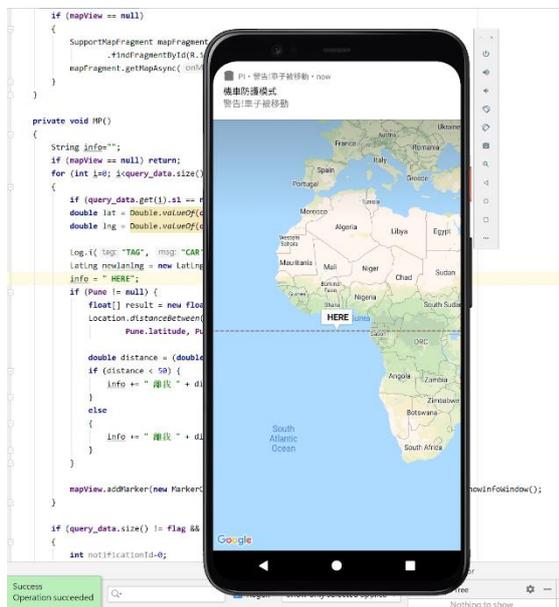


圖 5

此 app 有以下主要功能：

(一)即時查看車輛位置(圖 6):通過 gps 模組上傳 rpi 位置資料，讓 app 取用，其中紅色定位符號就是目前車輛所在位置(圖 6)。

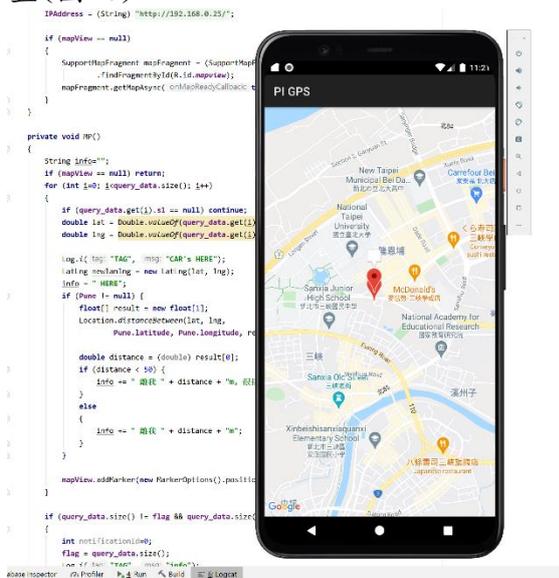


圖 6

(二)即時串流車輛周圍影像:透過 RTMP 協定與伺服器通訊，其中圖 7 右上角的

RTMP\_DEMO 就是我們的串流 app，用來監視車輛周圍的。

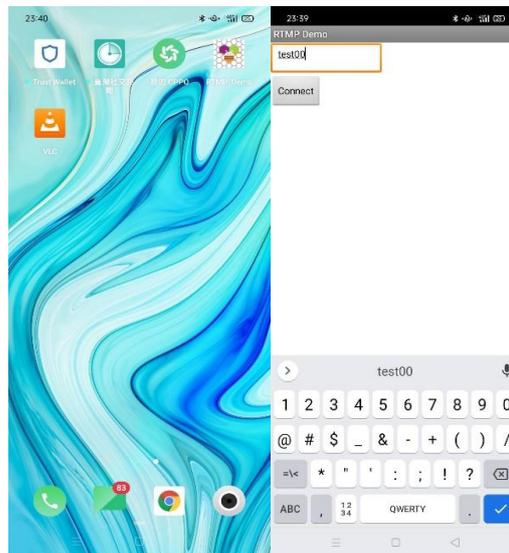


圖 7

圖 8

(三)追蹤車輛移動軌跡:通過每個 5 秒上傳一次 gps 資料，讓其成為一條軌跡，在車輛失蹤被偷時很有幫助。

使用工具及技術：、XAMPP、MySQL、Python、Android Studio、GStreamer、Arduino、Raspberry Pi 3B+。

## 二、專題進行方式：

### 1.1 資料搜尋：

先是從如何建置資料庫開始，我們率先考慮該用哪種方式進行撰寫，最後選擇用 MySQL，因為其體積小、速度快、總體擁有成本低，開源、而且支持多種作業系統，又是開源資料庫，提供的接口支持多種語言連接操作。

接著下一步就是決定該寫 ios 還是 android 的 app，我們放棄 ios 而轉向安卓是因為 ios 的認證程序繁瑣且複雜，而且在開發上的困難比較多，畢竟不是每個人的電腦都能安裝 xcode，所以我們才決定做 android，但是安卓也有缺點，bug 特別多，而且平均手機效能流暢度不是很高，當然也有很高端的

安卓手機，不過我們沒有人有。

## 1.2 蒐集並統整資料:

最困難的部分不是找資料，而是找到的資料是否可用，和整合起來是否可運作，再找 rpi gps 模組的資料時我們就碰壁過一次，不小心選擇到錯誤的模組，造成無法抓到 gps 資料。

鏡頭模組也是在 rpi 原生鏡頭跟網路視訊鏡頭間做挑選，最終選擇了原生鏡頭。

連線部分是在 NB-iot 跟 wifi 之間做選擇，因為 NB-iot 受限於他的頻寬跟傳輸速度太慢，無法及時串流影像，所以最終我們選擇了 wifi。

## 1.3 裝置安裝與運行模式

最初我們就想將使用者的操作程序降到最簡單，因此來實現攜帶以及安裝到車輛之便利性，首先將 rpi 裝到車子上，利用其上的 Buck Converter 將車上的 12v 電壓降壓成 5v，若車輛本身具有 5V DC 電源亦可直接使用，接著讓 rpi 接上電源，固定六軸加速度計並且將鏡頭至適當位置，等待他開機後運行事先設定好開機自動運行的程式，開啟手機 app 登陸帳號密碼，這樣即可啟動背景偵測模式，等待車子被搖動，app 即會推送通知，由於我們的設定是讓機車在停車的狀況下才讓 rpi 啟動，平時是不會上傳資料的，而且這樣做的好處是當我們需要時才打開防護模式，不需要時就部會誤觸造成 app 一直跳通知。

## 三、評估與結果:

我們在最終做出 app 時，其實是有點小失望，在 app 內做疊圖功能時，google map 會持續顯示不出來，因為其圖層顯示的問題，所以最後我們將視訊串流的 app 跟顯示位置的 app 分開來寫，這樣雖然要載兩次 app 可是卻可以完美解決圖層無法重疊之問題。

第二，因為現有設備效能問題，無法同時執行及時串流影像和上傳資料，又設備是在車廂內部，溫度高，造成 rpi 的 cpu 時常熱當而效能低落，故必須分成

兩個 app 分開進行。

另外就是在 app 外觀設計部分，我們無特別設計，這也是比較可惜的地方。

## 四、結語與展望

此次的專題我們克服了如何將各個原本不是使用同一系統的裝置連結在一起，還有這次是我初次寫 app，雖然這次的 app 是非常簡單而且功能性不豐富也不美觀的 app，不過還是讓我寫了很久，我希望下次能夠寫 ios 的 app，因為我自己使用的手機跟平板都是屬於蘋果生態圈的，而且說實話蘋果的效能真的不是安卓能比的。

不過這款 app 能夠實現及時通知車主自己的愛車被移動，我想光是這一項功能就有一定的價值在了吧，我們希望以後能夠學習特斯拉那樣，建置出一個完善的車聯網系統，讓台灣的用機車環境更為友善，另外如果可能希望與各縣市政府合作，取消人工收費，在所有機車上加裝類似 etag 之類的東西，然後在路邊公有停車格設置一個收費系統，向高速公路的那種，這樣只要停路邊的公有停車格裡，自動收費且不用擔心被移車，因為若是大家都有這套系統，是誰移車都瞭若指掌，有了我們的機車防盜 app 和類似 etag 的辨識晶片，雙重比對之下能夠輕輕鬆鬆找出來，想必這樣應該沒有人會輕易地去移別人的車子，畢竟所有證據都有，若提告損毀罪必定能夠成立，在這樣的系統下，移車問題應該能輕易解決，畢竟想惹麻煩上身的人應該沒幾個吧。

## 五、銘謝

感謝指導教授-張玉山教授，對我們沒有放棄，有好多次我們讓您失望了，但是您還是願意相信我們，相信我們能做得出來，而常常給我們有用的意見，真的是非常感謝。

## 六、參考文獻

- [1] Raspberry Pi 物聯網應用  
(Python) / 王玉樹著  
ISBN: 9865035243
- [2] Practical Linux with Raspberry  
Pi OS: Quick Start / Pajankar,  
Ashwin 著  
ISBN: 9781484265093
- [3] Beginning Sensor Networks with  
Xbee, Raspberry Pi, and Arduino:  
Sensing the World with Python and  
Micropython / Bell, Charles  
ISBN: 9781484257951
- [4] 樹莓派用戶指南 / 埃本·阿普頓  
(Eben Upton) 加雷思·哈菲克  
(Gareth Halfacree) 著  
ISBN: 9787115524072
- [5] 必須精通 MySQL 最新版：最成功應  
用範例全書 / 王英英著  
ISBN: 9789865501679
- [6] Blogger 舊站  
<https://blog.gtwang.org/iot/raspberry-pi-nginx-rtmp-server-live-streaming/>
- [7] 添加 GPS 时间和位置至 Raspberry  
Pi 项目  
<https://www.rs-online.com/designspark/add-gps-time-and-location-to-a-raspberry-pi-project-cn>  
ISBN: 9789861302782
- [8] Blogger 舊站  
<https://blog.gtwang.org/iot/raspberry-pi-mpu6050-six-axis-gyro-accelerometer-1/>
- [9] XAMPP 網頁伺服器架站工具設定與  
使用教學 | KJie Notes  
<https://www.kjnotes.com/devtools/54>
- [10] IT 邦幫忙  
<https://ithelp.ithome.com.tw/articles/10197921>
- [11] 零基礎快速自學 SQL，1 天足矣—  
附最全 SQL 學習資源和練習題！  
<https://www.finereport.com/tw/data>