

# 中華民國第 65 屆中小學科學展覽會

## 作品說明書

---

高級中等學校組 工程學科(一)

052316

智慧定點式道路警示牌

學校名稱： 新北市立樟樹國際實創高級中等學校

作者：  職三 周恭煥  職三 吳致穎  職三 張芯瑜	指導老師：  許國財
---	------------------

關鍵詞： Arduino、交通安全、物聯網

# 智慧定點式道路警示牌

## 摘要

在高速公路上，駕駛者若遇到車輛故障或事故，必須依照交通規定，在事故發生點後方100公尺處擺放三角警示牌，以提醒後方來車。然而，在高速公路環境中，車輛行駛速度較快，駕駛者下車擺放警示牌的過程中，極易發生二次事故。因此我們設計「智慧定點式道路警示牌」，本裝置安裝於公路路肩，透過網路遠程開啟。這樣一來就可以避免下車擺放警示牌的過程中發生意外。本次設計運用到 Wi-Fi 做為與雲端通訊之媒介，也達成了物聯網與雲端化。使用者可用網頁進行遠端開啟警示牌或在車內加入車禍自動回報系統，不僅如此，我們還可以用來檢視警示牌狀態。

## 壹、前言

在高速公路上，駕駛者若遇到車輛故障或事故，必須依照交通規定，在事故發生點後方100公尺處擺放三角警示牌，以提醒後方來車。然而，在高速公路環境中，車輛行駛速度較快，且駕駛者下車擺放警示牌的過程中，極易發生二次事故。根據統計，高速公路上因駕駛者下車擺放警示牌而發生的二次事故比例高達7.51%，其中多數案例涉及夜間能見度低、車流量大或駕駛者因緊張而行動遲緩等因素，導致後方來車反應不及，進而發生碰撞（曹岡陽，2024）。這顯示目前的警示方式不僅存在執行困難，也對駕駛者安全造成嚴重威脅。因此，我們希望開發一款具備低成本、易部署、遠端控制功能與智能的智慧警示裝置，以提升高速公路事故處理時的安全性，降低二次事故發生的機率（游書昀，2020）。

目前的道路警示方式存在諸多不足，例如：需手動擺放警示牌，這可能耗時耗力，並增加駕駛者的風險、當路況突變時，無法即時提醒後方來車，容易導致二次事故。因此，本專題的目標是設計一款對駕駛更安全的警示系統，期盼以科技方式提升道路警示效果。本產品的主要特點如下分點敘述。

一、從手動放置改為遠端操控：

- （一）透過行動裝置，讓駕駛者無需下車即可啟動警示牌。
- （二）降低駕駛者在高速公路上行動的風險，提高安全性。

二、可遠端檢視警示牌狀態：

(一) 使用行動裝置連接網頁，即時得知警示牌狀態。

(二) 方便管理者管理每個警示牌，確保每個警示牌都能隨時開啟。

透過本專題的研究，我們期望能夠解決現有警示設備的問題，提供更有效率的解決方案，進而降低高速公路上的事故發生率，保障駕駛者與乘客的生命安全。

本作品在參加科展，經由與評審討論後。發現前次設計之警示功能不足，以及探討結合車聯網之可能性。

## 貳、研究設備及器材

表1為製作警示牌本體所用到的設備器材、規格、數量與使用說明。

表1：研究設備及器材統計表

名稱	規格	數量	使用說明
太陽能板	單矽晶16cm*11cm	1個	裝設在機身上方，用來提供電能
太陽能控制器	DFR0559	1個	控制整個電能的分配
電池	18650	1顆	存取多餘的電能
燈條	寬:5mm 長:1m	1條	強化夜間警示效果
直流電流電壓感測器	INA226	1個	讀取電流和電壓，計算電池的電量
開發板	Arduino UNO R4 WiFi	1個	負責處理和傳輸資料、控制伺服馬達、LED 燈和檢測太陽能板是否有再提供電能
伺服馬達	TD-8120MG	1顆	負責控制警示牌的開與關
密集板	20cm*15cm	6片	利用雷射切割做出警示牌的外殼
風扣板	20cm*20cm	1片	做為警示牌的牌子
反光貼	20mm	1捲	貼出警示圖標用，可以在晚上時起到明顯的作用
電路板		1片	整合電路用
麵包版		1塊	當有其他擴充功能時的擴充區
3D 列印件		1塊	做為警示牌牌子與馬達軸承的連接器

## 參、研究過程或方法

### 一、實現目標

若本裝置要達成所設計的運作模式，需要具備一下幾點

- 1.遠端開、關裝置及檢視裝置狀態。
- 2.警示牌開啟結構。
- 3.裝置電力來源及儲能。
- 4.監測電量及充電狀態。
- 5.檢視網路連接狀態。
- 6.裝置與伺服器傳輸協議。
- 7.車禍自動回報系統。

### 二、硬體研究過程

#### （一）開發版

選用 Arduino UNO R4 WiFi 作為主控板。主要原因是其擁有內建 Wi-Fi 功能，可直接與伺服器通訊，減少外接模組的複雜度。此外，Arduino 在網路上有大量的教學資源與應用範例，對於開發初期問題的排除與功能延伸都非常有幫助，能大幅提升開發效率。

#### （二）警示牌門板

警示牌門板的主要結構選用風扣板。風扣板相較於其他塑膠板材，不僅重量輕、便於加工與切割，更重要的是在實際路面使用時，即使不慎被車輛撞擊也不容易對車輛造成刮傷或損壞，具有良好的安全性與便利性。

此外，為了提升警示效果，我們採用反光條與 LED 燈條雙重警示機制。燈條具備高亮度、耗電量低且發光範圍廣的特性，能在夜間或惡劣天氣下維持良好可視性，達到警示效果。

不過，警示牌電力系統較不完善，無法直接加入 LED 燈條，因其耗電量會影響整體續航，只能使用反光條。為此，我們調整了電池容量，並選用低功耗燈條，使裝置既能持久運作，也保有良好的警示效果。

### （三）電力系統

本裝置採用太陽能板與18650鋰電池作為主要電力來源，搭配太陽能控制器進行充放電管理，以確保在缺乏外部電源的環境下也能穩定運作。為強化電力監測功能，我們加入 INA226 電源監測晶片，用來精準量測電池端的電壓與電流，並計算剩餘電量。由於電池負責在夜間與陰天提供穩定電力，因此需要精確監測；而太陽能板部分，我們僅需確認是否有輸出電壓即可，無需電流數據。

### （四）LED 狀態顯示

為方便檢測裝置運作情況，我們設計 LED 指示燈。包括網路連線狀態指示燈與電源指示燈，能清楚反映裝置當前是否有成功連上網路或正在回傳還是接收資料、是否有供電，方便在測試與實際使用時快速檢查系統是否正常。

### （五）機殼材料

為保護內部電子元件並支撐整體結構，我們使用密集板與雷射切割製作裝置的外殼。密集板材質堅固且加工彈性高，適合雷射雕刻與開孔，更重要的是它不具導電性，在裝置內含電子電路與供電模組的情況下，可有效避免短路風險，提高整體安全性。其結構穩定、硬度適中，有助於保護內部元件並確保裝置整體的耐用性與維修便利性。

### 三、軟體研究過程

本系統採用 Django 網頁框架開發並部屬至 Vercel 雲端平台，透過 API 傳輸資料給裝置，讓使用者可透過網頁介面遠端啟動警示牌。

- 1.用戶端：行動裝置介面提供操作與即時資訊顯示。
- 2.伺服器端：處理來自用戶的請求，並傳送指令至警示牌。
- 3.資料庫：記錄設備狀態與使用歷史，檢視路段有無安全隱患，且防治警示牌遭到濫用 (Mike Ku, 2020)。
- 4.物聯網整合：透過 HTTP 協議，並使用 JSON 資料格式進行裝置間通訊。
- 5.車聯網應用：利用車速判斷車輛位於高速公路時是否存在異常，若存在異常，使用車載車聯網設備回傳至雲端，並廣播給其他車輛，以達成警示之目的。

### 四、操作說明

- 1.駕駛發生車輛故障或事故時，透過車輛事故回報模組或透過行動裝置傳送開啟訊號。
- 2.透過雲端伺服器傳送至警示牌。
- 3.警示牌開啟，提醒後方來車減速或換道。
- 4.狀態監控，系統會記錄啟動時間與狀態，確保設備運行正常。
- 5.資訊回報，警示牌接收到任何指令，也會同步發送通知至 Telegram，使管理者了解當前狀況。

## 五、架構圖

如圖1所示，展示警示牌系統的技術架構，分為警示牌本體與雲端服務兩大部分。

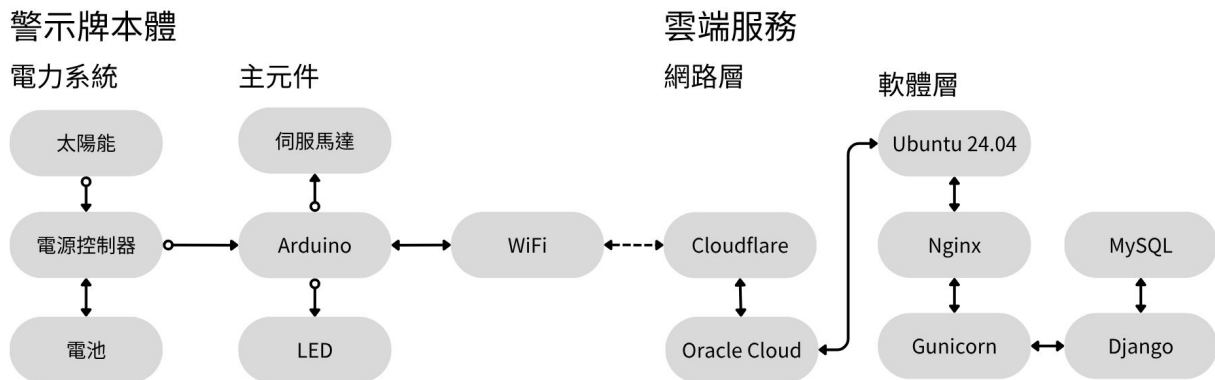


圖1：警示牌架構圖

### （一）警示牌本體

警示牌本體主要由電力系統與主元件組成，以確保設備正常運行。

#### 1.電力系統：

- （1）太陽能板為系統提供電力，確保警示牌可在戶外長時間運行。
- （2）電源控制器負責管理電力輸出，確保Arduino及其他元件獲得穩定的電源供應。
- （3）電池儲存能量，確保在無日照時仍能運作。

#### 2.主元件：

- （1）Arduino 作為核心控制器，負責接收遠端指令並控制 LED 燈的狀態。
- （2）伺服馬達 可能用於調整警示牌的角度或其他可動部件。
- （3）LED 做為電源指示燈及 Wi-Fi 連接狀態。
- （4）WiFi 用於與雲端伺服器進行通訊，接收遠端指令並回報系統狀態。

### （二）雲端服務

提供管理與數據存取功能。

#### 1.網路層：

- （1）Cloudflare 用於提供網路安全與 DDoS 防護，確保系統不受惡意攻擊影響。
- （2）Vercel 作為伺服器的運行平台，提供運算與存儲資源。

#### 2.軟體層：

- （1）Wsgi 用於串接 Django 與 Vercel。
- （2）Django 作為 Web 框架，負責處理來自 Arduino 和用戶的請求，提供管理界面與 API。
- （3）PostgreSQL 負責儲存警示牌的狀態、使用紀錄與用戶數據。

## 肆、研究結果

本專題研究結果如下說明。

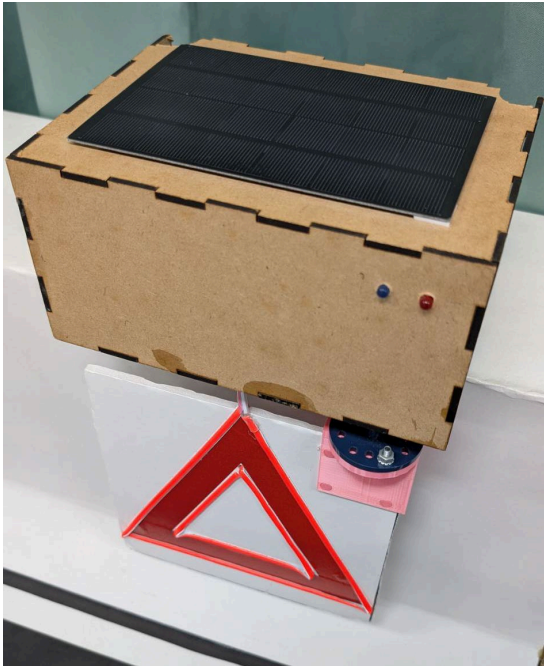


圖2：智慧定點道路警示牌待機示意圖

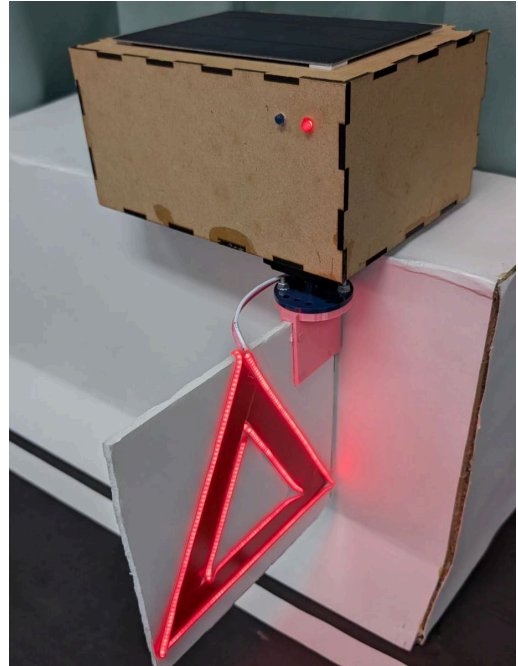


圖3：智慧定點道路警示牌開啟示意圖

下方的圖4是我們警示牌系統的遠端介面，讓使用者可以監控警示牌的運作狀態，並進行遠端控制。畫面中央顯示使用者名稱「kenchou2006」，代表使用者已成功登入系統，接著可以進行各種操作。

本專題介面上方排列四個功能按鈕，包含「系統管理」按鈕可能用來設定設備或管理權限，而「會員登出」則讓使用者可以安全登出帳戶。此外，使用者還能透過「啟用立牌」與「收回立牌」遠端控制警示牌的開啟及收回，或點擊「檢視 Log」來查看警示牌得開啟紀錄(如圖5所示)。



頁面下方則詳細顯示當前系統的狀態資訊。從「伺服器狀態」可以看到，目前系統沒有任何請求處理中，而「Arduino 狀態」顯示設備處於在線狀態，證明連線正常。電量資訊則顯示設備目前有95%的電力，並且正在充電中。警示牌的當前狀態為開啟，表示設備目前正處於警示狀態，而省電模式則顯示為關閉，說明設備尚未進入省電運行模式。

這個系統提供一個直觀且高效的方式，讓使用者能夠隨時掌握警示牌的運作狀態，並進行遠端操控，大幅提升道路安全與設備管理的便利性。

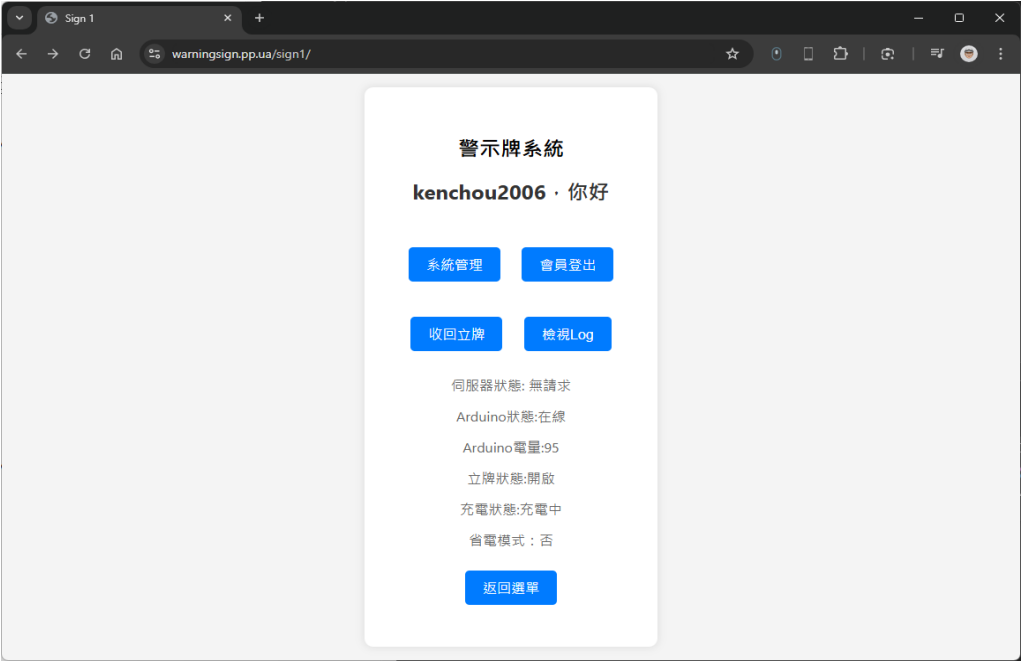


圖4：Django 網頁示意圖

如圖5顯示的是警示牌系統的存取記錄列表（Access Record List），其主要功能是防止警示牌遭到濫用，並確保系統的使用行為可被追蹤與管理。畫面中的表格記錄每次存取系統的詳細資訊，包括存取時間、IP 位址、訪問的 URL、使用的作業系統、瀏覽器名稱及版本。

The screenshot shows a web browser window with the URL 'warningsign.pp.ua/access-records/'. The page title is 'Access Record List'. Below the title, there is a table with 7 columns: 'Access Time', 'IP Address', 'URLs', 'OS Name', 'OS Version', 'Browser Name', and 'Browser Version'. The table contains 20 rows of access records.

Access Time	IP Address	URLs	OS Name	OS Version	Browser Name	Browser Version
2024年10月1日 14:46	162.158.243.56	input	Windows	10	Chrome	129.0.0
2024年10月1日 14:46	172.68.87.132	output	Other		Other	
2024年10月29日 13:23	::ffff:172.68.87.132	input	Windows	10	Chrome	130.0.0
2024年10月29日 13:26	::ffff:172.69.221.168	output	Other		Other	
2024年10月29日 13:33	::ffff:172.69.221.167	input	Windows	10	Chrome	130.0.0
2024年10月29日 13:33	::ffff:172.69.221.167	output	Other		Other	
2024年10月29日 13:34	::ffff:172.69.221.168	input	Windows	10	Chrome	130.0.0
2024年10月29日 13:34	::ffff:172.69.221.167	output	Windows	10	Chrome	130.0.0
2024年10月29日 13:37	::ffff:172.69.221.167	input	Windows	10	Chrome	130.0.0
2024年10月29日 13:45	::ffff:172.68.87.132	output	Other		Other	
2024年10月29日 13:47	::ffff:172.69.221.167	input	Windows	10	Chrome	130.0.0
2024年10月29日 13:48	::ffff:172.68.87.133	output	Other		Other	
2024年11月5日 08:00	::ffff:172.69.221.84	input	Windows	10	Chrome	130.0.0
2024年11月5日 08:01	::ffff:172.69.221.84	output	Windows	10	Chrome	130.0.0
2024年11月5日 08:02	::ffff:172.69.221.84	input	Windows	10	Chrome	130.0.0
2024年11月5日 13:34	::ffff:172.69.221.168	input	Windows	10	Chrome	130.0.0
2024年11月5日 13:40	::ffff:172.68.87.132	output	Windows	10	Chrome	130.0.0

圖5：伺服器訪問紀錄示意圖

```
WarningSign > main > Current File > views.py
92 csrf_exempt 1 usage 周慕煥
93 def arduino_info(request):
94     global arduino_battery_num, warningsign_status, arduino_status_UltraSound, line_notify_arduino_status_Ultr
95     if request.method == 'POST':
96         try:
97             data = json.loads(request.body)
98             def parse_int(value, default=0): 周慕煥
99                 try:
100                     return int(value)
101                 except (ValueError, TypeError):
102                     return default
103             arduino_battery_num = parse_int(data.get('battery', 0))
104             warningsign_status_str = parse_int(data.get('warningsign_status', 2))
105             arduino_charing_str = parse_int(data.get('charging', 2))
106             arduino_eco_str = parse_int(data.get('eco', 2))
107
108             if warningsign_status_str == 1:
109                 warningsign_status = 1
110                 warningsign_status_input = True
111             elif warningsign_status_str == 2:
112                 warningsign_status = 2
113                 warningsign_status_input = True
114             else: warningsign_status_input = False
115             if arduino_charing_str == 1: arduino_charing = True
116             else: arduino_charing = False
117             if arduino_eco_str == 1: arduino_eco = True
118             else: arduino_eco = False
119             if arduino_battery_num <= 20 and notify_battery:
120                 notify_battery = False
121                 asyncio.run(send_notify("電量低於 20，請檢查狀態"))
122             elif not arduino_battery_num <= 20: notify_battery = True
123             if warningsign_status_input: return JsonResponse({"status": "success"})
124             else: return JsonResponse({"status": "error", "message": "Invalid data"})
125             except json.JSONDecodeError: return JsonResponse({"status": "error", "message": "Invalid JSON data"})
126             else: return JsonResponse({"status": "error", "message": "Invalid method"})
```

圖6：接收 Arduino 資料程式碼示意圖

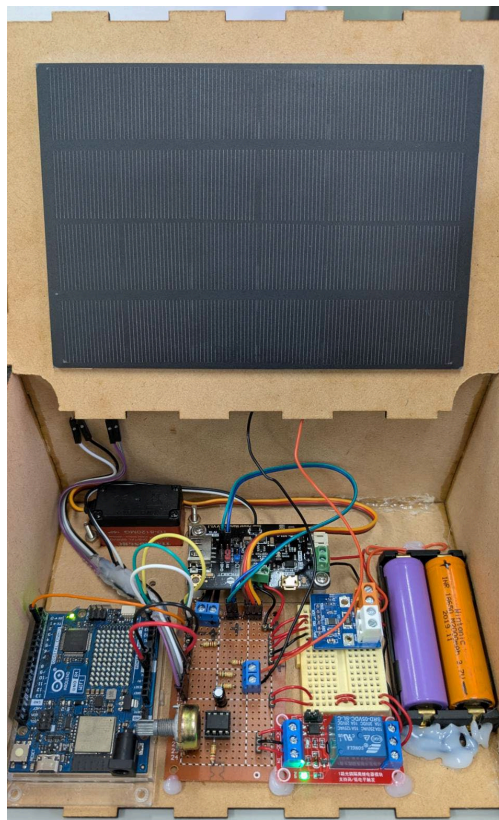


圖7：作品內部構造示意圖

本專題內部結構，根據本專題研究動機與目的，藉由 Arduino 控制硬體感測設備，以及透過 API 與雲端服務技術，讓使用者可遠端啟動警示牌，以降低發生二次失故。如圖7所示。



圖8：速度異常展示



圖9：接收廣播訊息

利用在高速公路的最低時速限制規範，本次也將時速作為判斷車輛是否故障之依據。此功能使用 Flutter 開發手機 App(Baseflow，2024)以模擬車聯網設備，在車輛時速過低時即傳送警報。其他車輛亦可透過相同的介面接收廣播資訊。本系統使用 Django 接收回報資訊，及傳送接收到的資訊給其他民眾。如圖8、圖9所示。

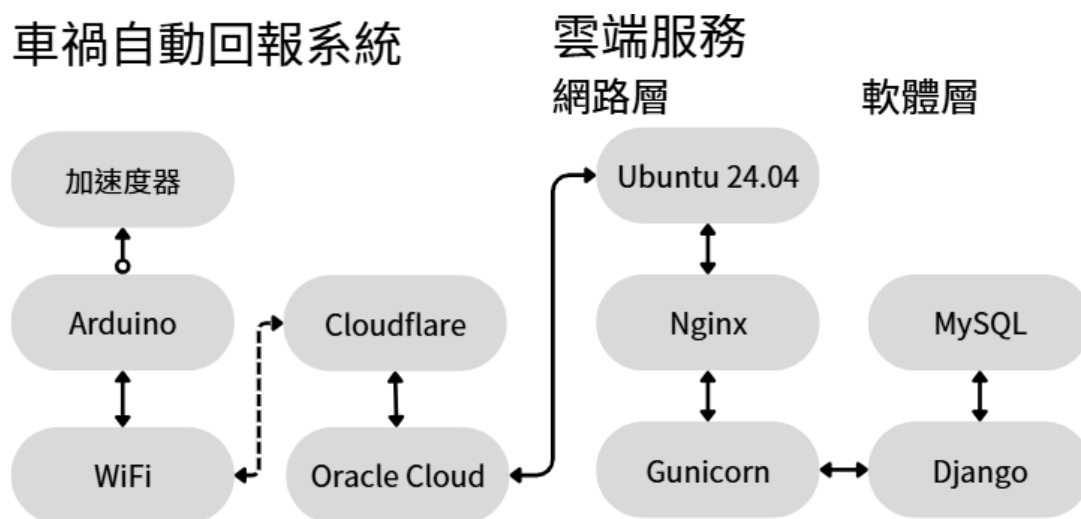


圖10：車禍自動回報系統方塊圖

透過在展示模型車輛上裝設開發版加速度計、讓車輛在撞擊後可以主動回報狀況至 Django 伺服器，達成自動開啟警示牌。實現本作品與車聯網應用的結合。

## 伍、討論

本裝置的開發過程中，針對現有警示系統的不足進行深入研究與改進。透過遠端控制與高亮度 LED 顯示技術，我們有效提升裝置的可視性與安全性。然而，在實施過程中仍面臨一些挑戰，例如通訊距離限制與使用範圍受限。主要挑戰與解決方案如下分點敘述。

### 一、通訊問題

目前本裝置採用 WiFi 進行短距離通訊，最大傳輸距離僅100公尺。解決方案是引入 LoRa 技術(維基百科，2024)，以擴展通訊距離。然而，LoRa 的頻寬較低，在數據傳輸方面可能受到限制。

### 二、使用範圍限制

原先我們想將其裝設於路面，可使任何道路皆可使用警示牌。但考量到對路面結構可能造成的影響及成本。所以改為用於路肩使用，但未來我們希望可以結合高速公路上的電子顯示器來顯示中間的情況。

## 陸、結論

本裝置透過遠端控制，成功提升使用警示牌的安全性與便利性，減少駕駛因故障停車後的風險。同時，透過通訊技術的改進，提高遠端操控的穩定性。然而，仍需進一步解決通訊距離與適用範圍的問題，以提高本裝置的普及性。未來發展方向如下分點敘述。

- 一、**提升通訊技術**：未來希望結合 LoRa，實現高效、穩定的遠端操控，確保訊號不易中斷 (Semtech, 2003)。
  - 二、**提高設備部署範圍**：研究不同道路環境（如城市道路、高速公路等）的適用性，確保裝置能夠普及應用。
  - 三、**增加智能偵測功能**：未來希望運用 AI 技術，讓裝置能夠自動感測車輛故障並立即啟動警示，提高反應速度。
  - 四、**駕駛者接受度提升**：透過推廣與宣導，提高駕駛者對該系統的接受度，促使其廣泛應用。
- 透過上述改進，本裝置能夠更有效地降低道路上的二次事故風險，提升駕駛與乘客的安全性。

## 柒、參考資料

曹岡陽(2024年12月11日)。下車放三角錐！女遭砂石車猛撞當場輾斃 台88線4車追撞。太報。

2025年1月2日。

游書昀(2020)。高速公路二次事故之研究－以國道一號楊梅至頭份路段為例。中央警察大學  
交通管理研究所碩士論文。

Jacob Hylén(2024)。UNO R4 WiFi 網路範例。Arduino DOCS，至2024年5月7日。

Gareth D(2024)。Creating a JSON response using Django and Python。SENTRY，至2024年7月23日。

維基百科(2024)。LoRa。維基百科，至2025年1月20日。

Mike Ku(2020)。Django連接PostgreSQL資料庫手把手教學。你的Py教練Mike，至2024年8月16日。

Baseflow(2024)。Flutter GPS 插件，可輕鬆存取平台特定的位置服務。pub.dev，至2024年5月30日。

以上所有圖片皆為作者親自製作。

## 【評語】 052316

1. 本作品探討利用定點道路警示牌，改善駕駛人擺放三角警示牌易發生二次事故問題，值得鼓勵。
2. 目前作品具遠端控制功能，實際運用於高速公路場域可能產生很多問題，應仔細評估思考如何解決，才可能使作品實際落實於高速公路場域。
3. 遠端操作介面具基本功能，建議進一步優化導入圖像化狀態回饋，例警示牌開啟、電量警示圖示，讓用路人在緊急狀況下更容易使用與理解；系統具操作紀錄功能，可協助管理單位分析事故頻率較高之路段環境。

作品海報



智慧定點式道路敬示牌

# 壹、研究動機

目前於高速公路遇到車輛拋錨時，需要車主自行下車至車輛後方100公尺處放置三角牌。由於高速公路的車速非常快，車流量大，手動放置警示牌可能有遭到來車追撞的風險，本產品設計亦在解決此問題。



Source : 一朵花汽車百貨

# 貳、研究目的

針對在高速公路放置警示牌可能遇到的安全問題，在111年，動力機械群的學長製作了”移動式三角警示牌”，但仍需車主下車放置地面再以手機遙控到100公尺外，並需要所有車主自行購買該裝置，我們認為此裝置依然有安全性問題，且未必所有車主都會為了突發狀況而購買該裝置。因此，我們設計了”智慧定點式道路警示牌”，除了解決車主安全問題也不用車主購買，結合綠能科技，安全又環保且達到聯合國永續發展目標SDGs的SDG 7 可負擔的潔淨能源、SDG 11 永續城鄉與SDG 12 責任消費及生產。

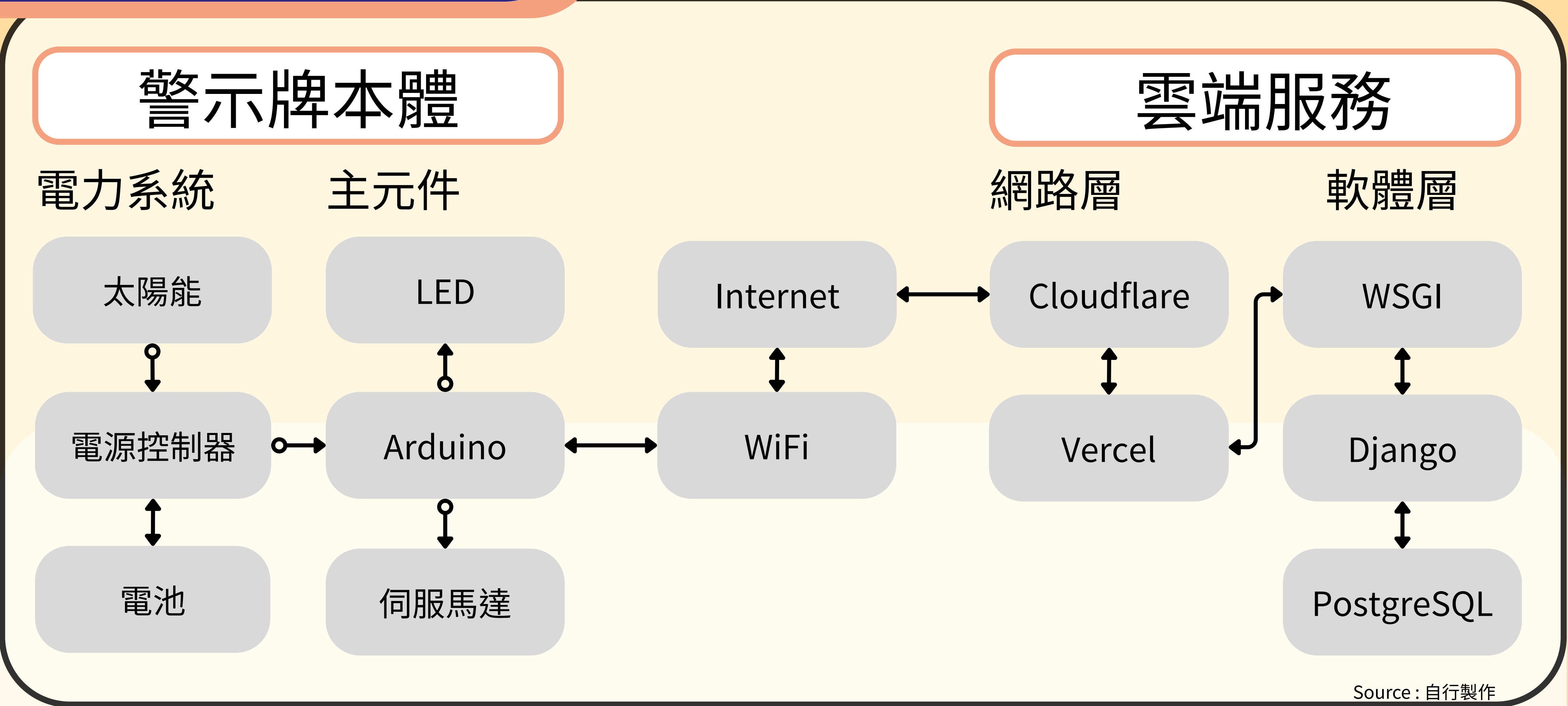


Source : United Nations

表一:各種道路警示牌比較					
項次	安全性	便利性	價格	智慧化	環保永續
傳統警示牌	X	X	低	X	X
移動式警示牌	X	△	高	X	X
智慧定點式道路警示牌	O	O	無	O	O

Source : 自行製作

# 參、系統方塊圖



Source : 自行製作

圖1：系統方塊圖

## 功能說明

- 一、本裝置結合物聯網，透過WiFi連結至雲端伺服器，達成遠距控制及管理智慧警示牌之目標，並將雲端數據提供給交通單位參考。
- 二、使用綠能供電，結合綠能與科技，安全又環保，且達到聯合國永續發展SDGs目標。



# 肆、研究過程

## 使用零件



Arduino UNO R4 WiFi



太陽能板



伺服馬達



電源控制器



LED燈條



加速度感測器

Source : Arduino Store , Taiwaniot , RCdrone , element14 , jmaker , QuartzComponents

## 雲端服務



Cloudflare



Vercel

Source : iconduck

## 使用軟體



Django

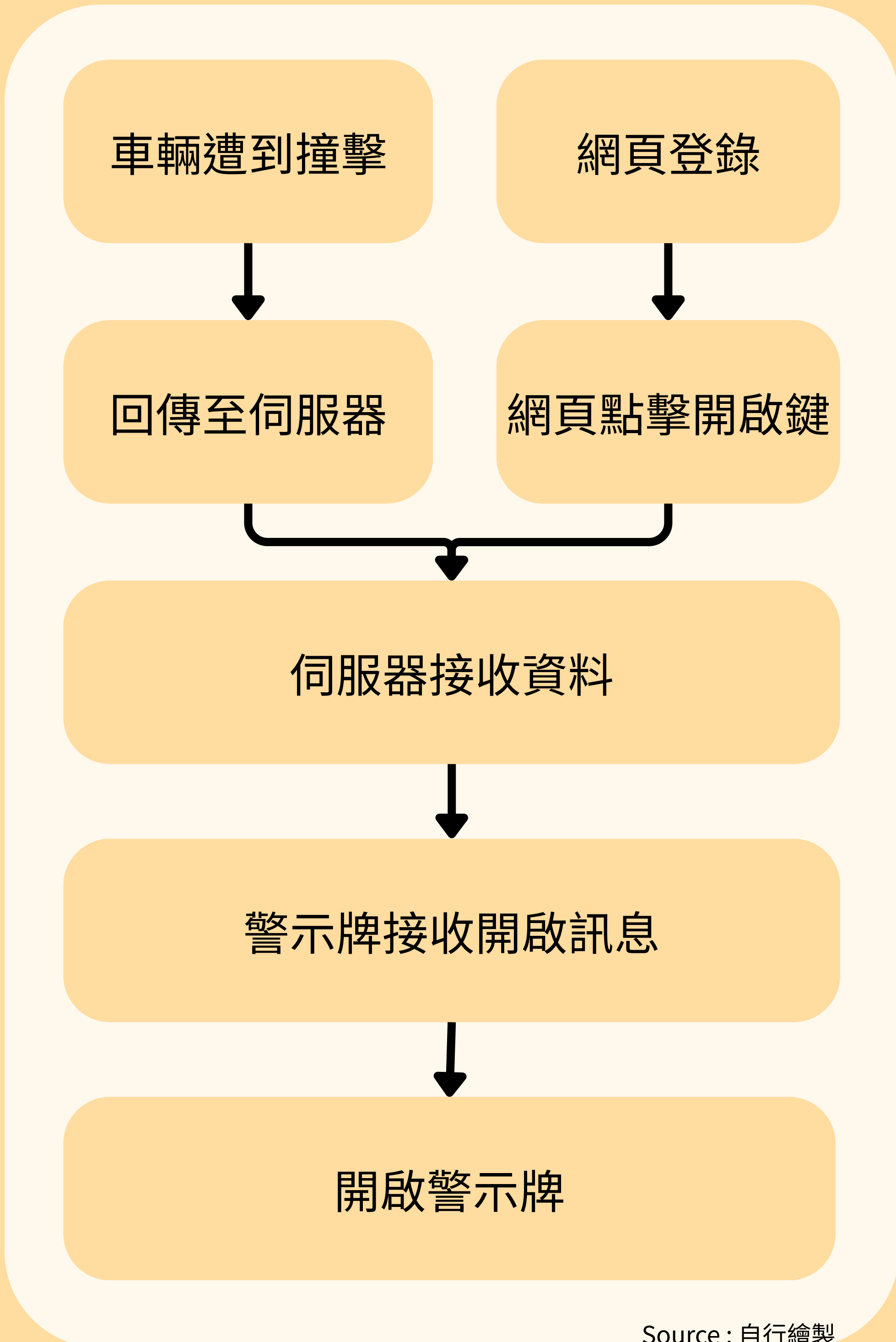


PostgreSQL



Flutter

Source : iconduck



**硬體設計** | 智慧定點式道路警示牌直接安裝在**高速公路的路肩**，民眾需要使用時，可透過指定的App開啟警示牌。

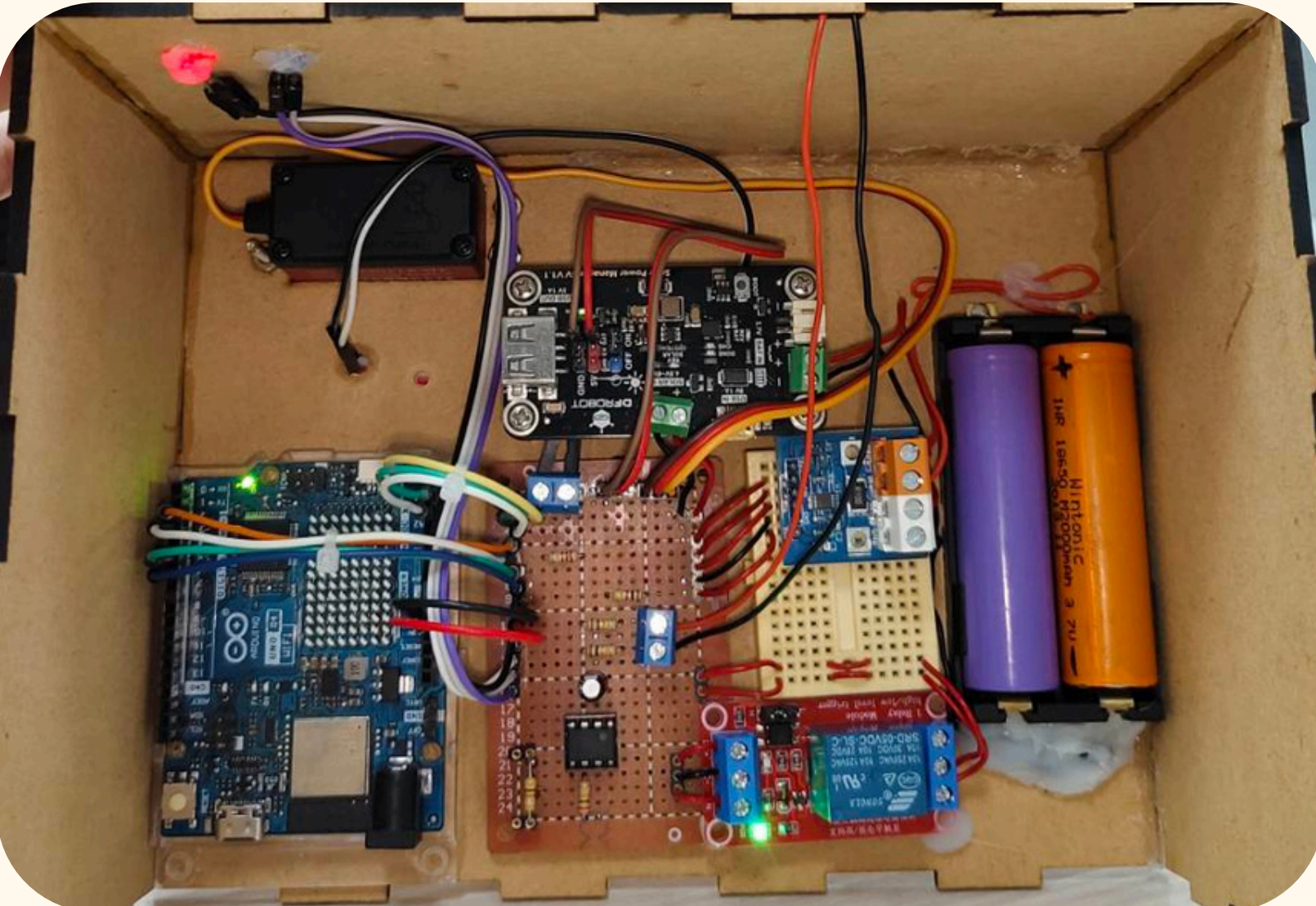


圖2：內部構造與配線



圖3：跑馬燈測試

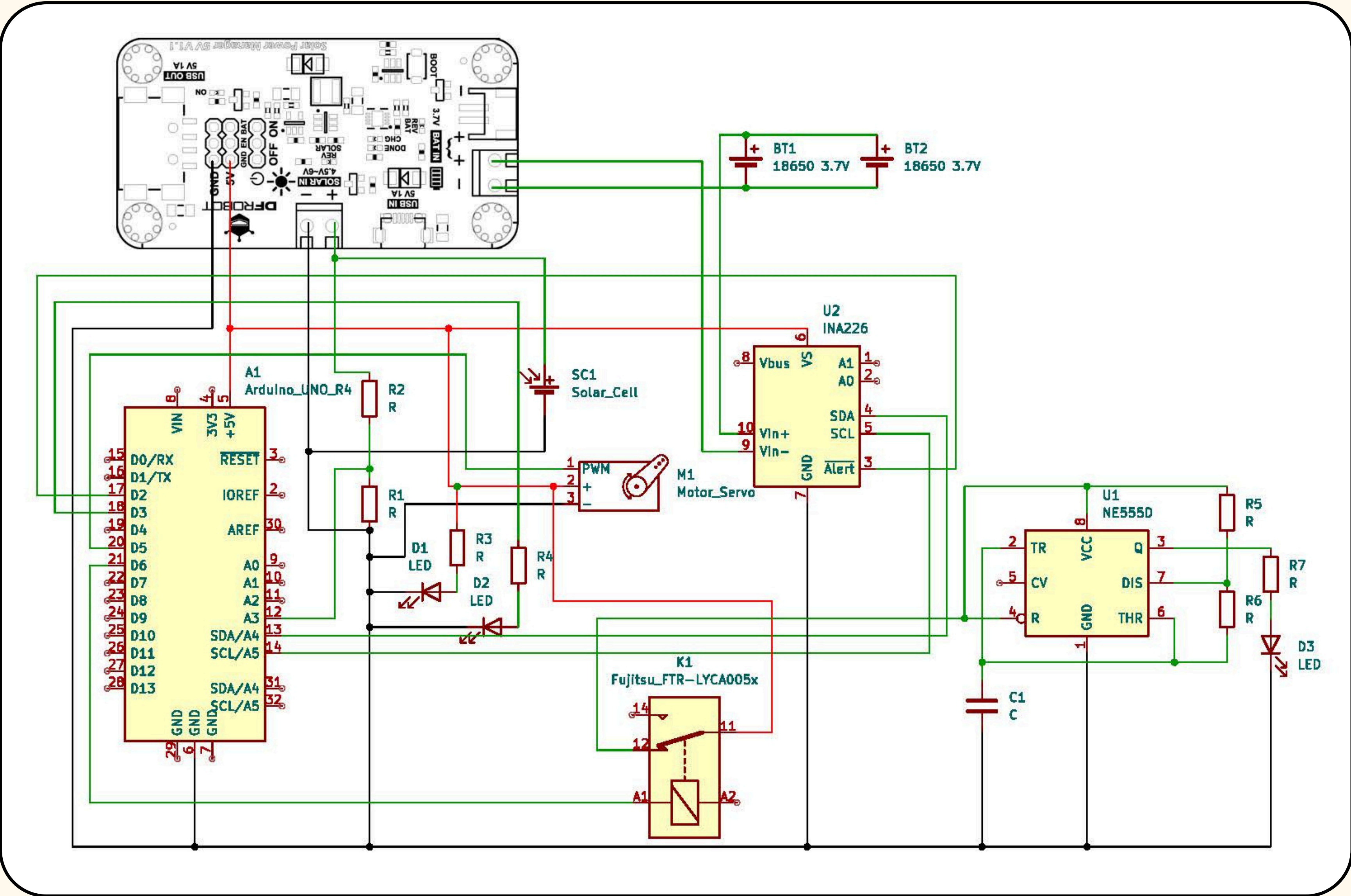


圖4：內部電路圖

Source : 自行拍攝

**軟體設計** | 本專題使用**Django**開發網頁伺服器，並且部署至**Vercel**雲端平台，隨處皆可存取及使用伺服器功能。

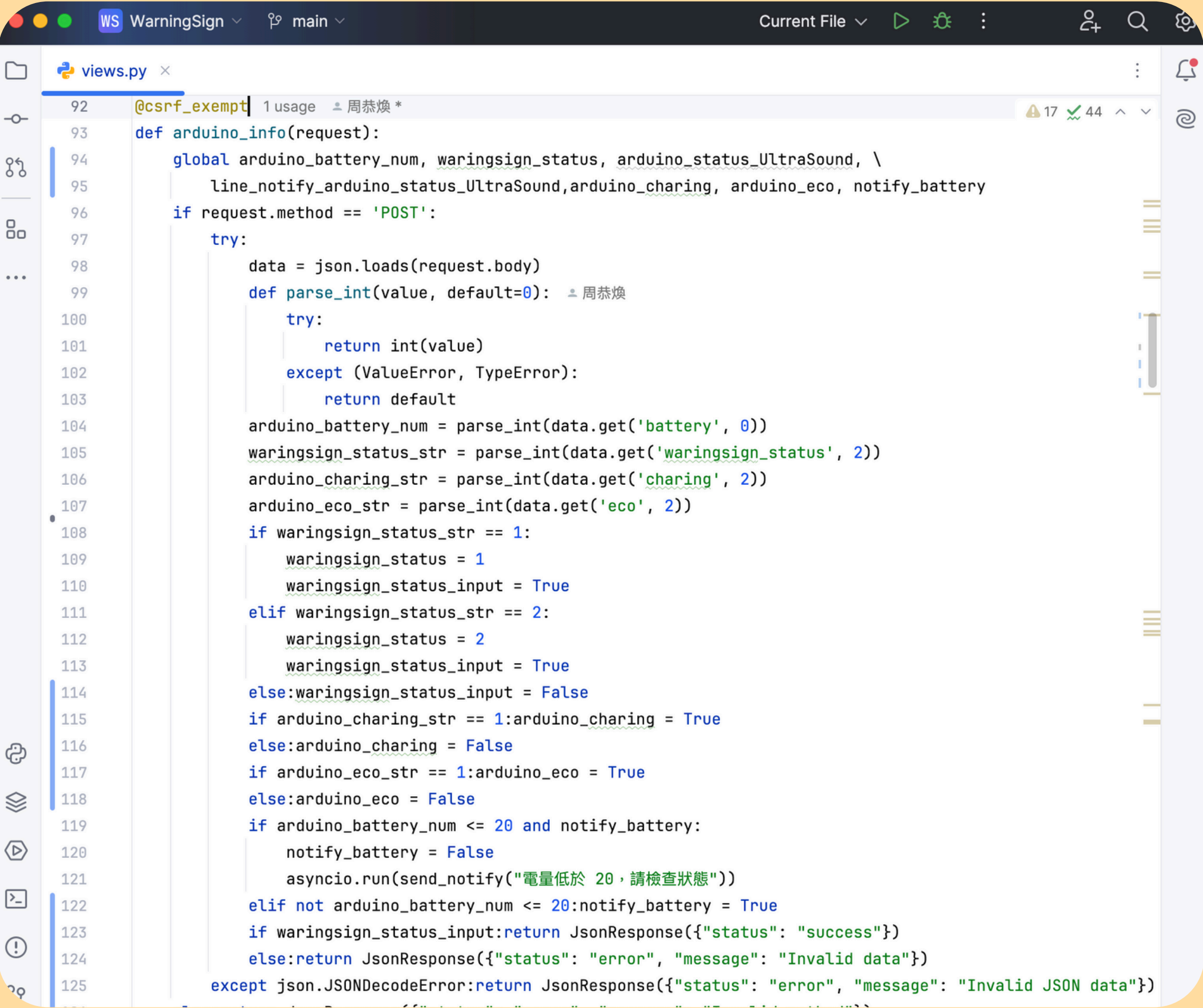


圖5：接收資料程式

Source : 自行製作

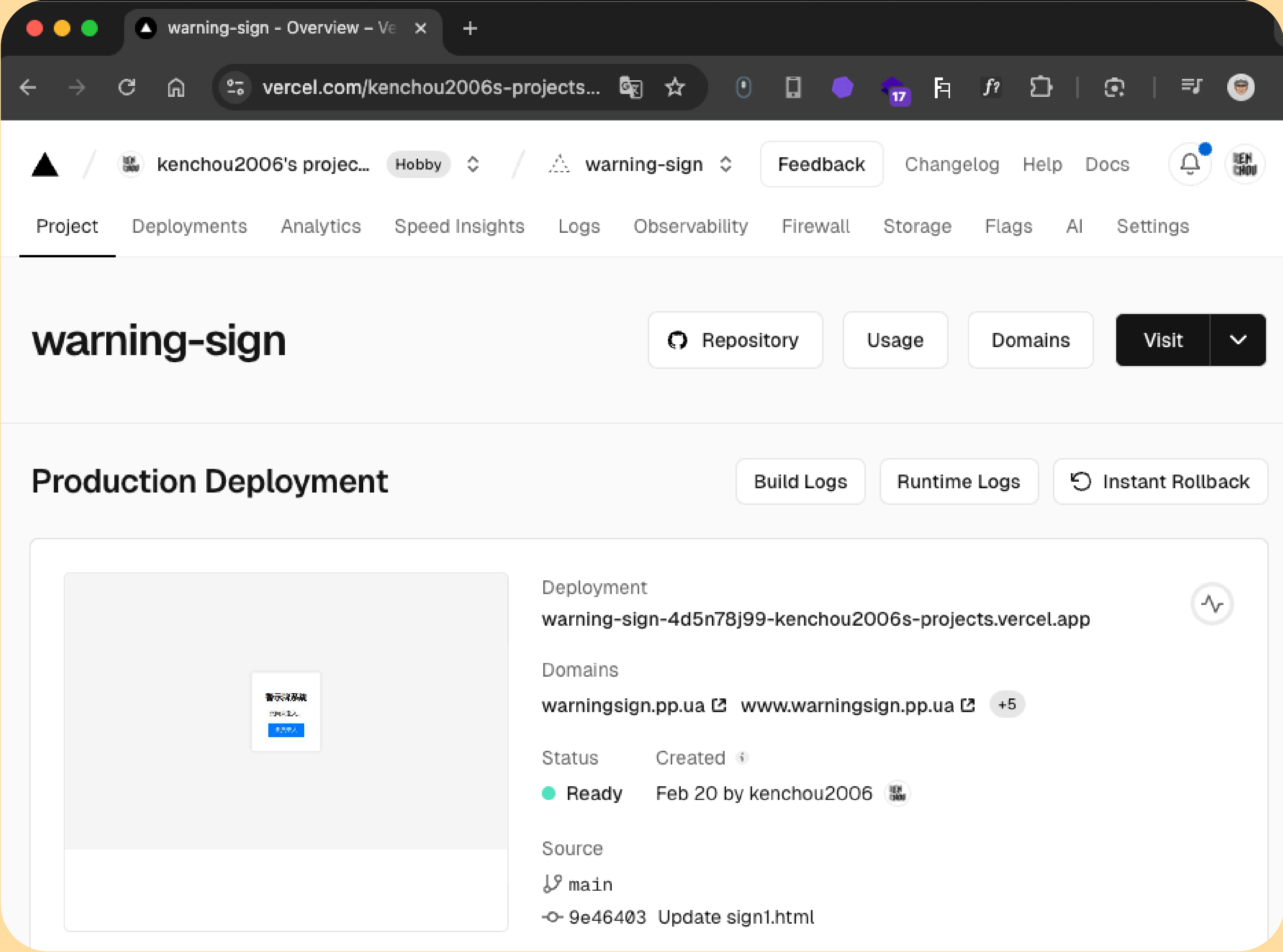


圖6：雲端平台部署

Source : 自行製作



# 伍、研究結果

Source : 自行拍攝

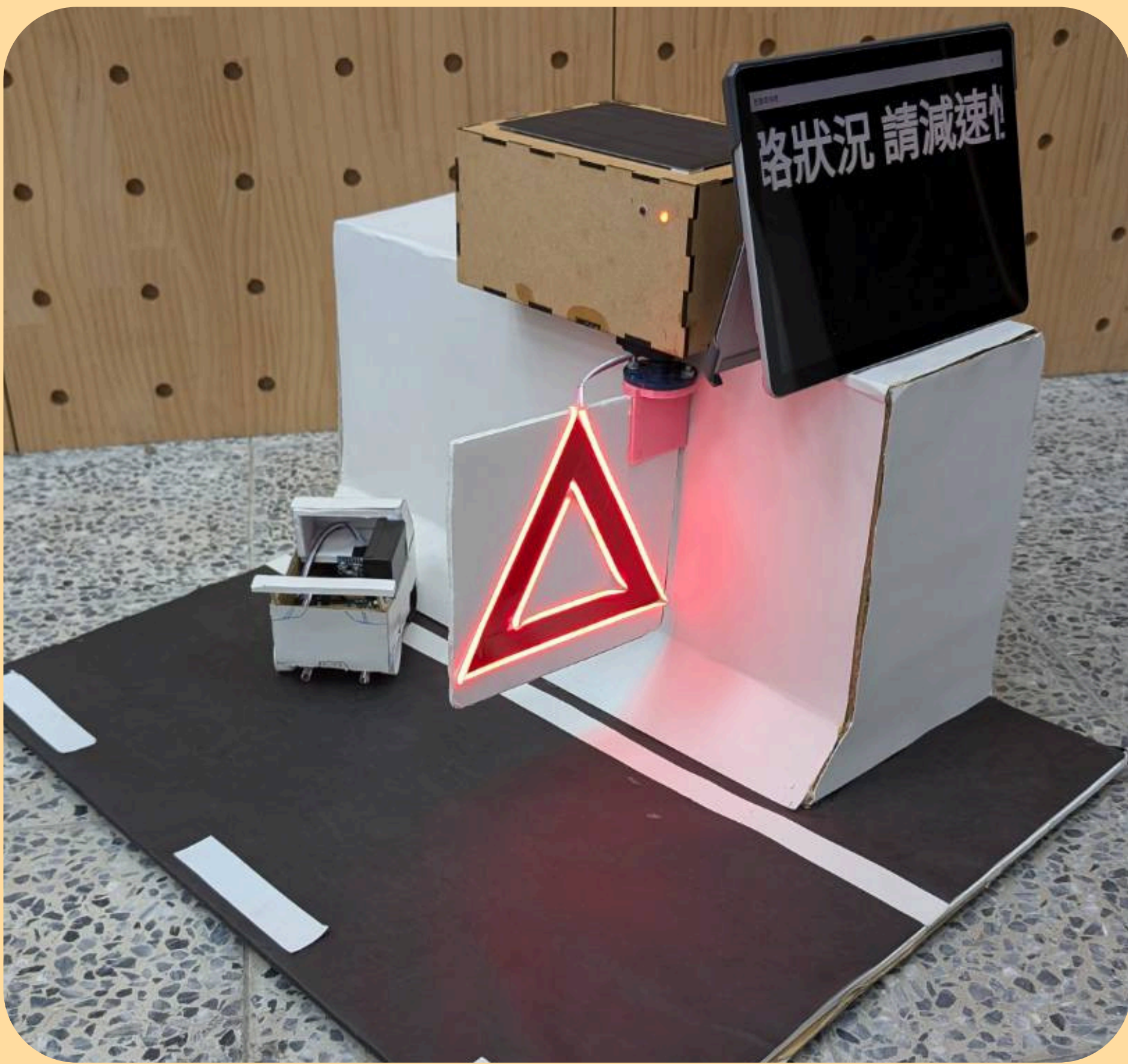


圖7：智慧定點式道路警示牌成果

Source : 自行製作



圖8：裝置放置位置

將裝置狀設於路肩位置，預計將每300~1000公尺放置一個。

Source : 自行製作



圖9：操作介面

用戶端操作介面，具備開啟、關閉警示牌及查看設備狀態等功能

Source : 自行製作



圖10：時速異常檢測

以Flutter應用程式模擬車聯網裝置。透過偵測時速，監測車輛是否為異常。偵測到異常後會發送警報給其他車輛。

Source : 自行製作

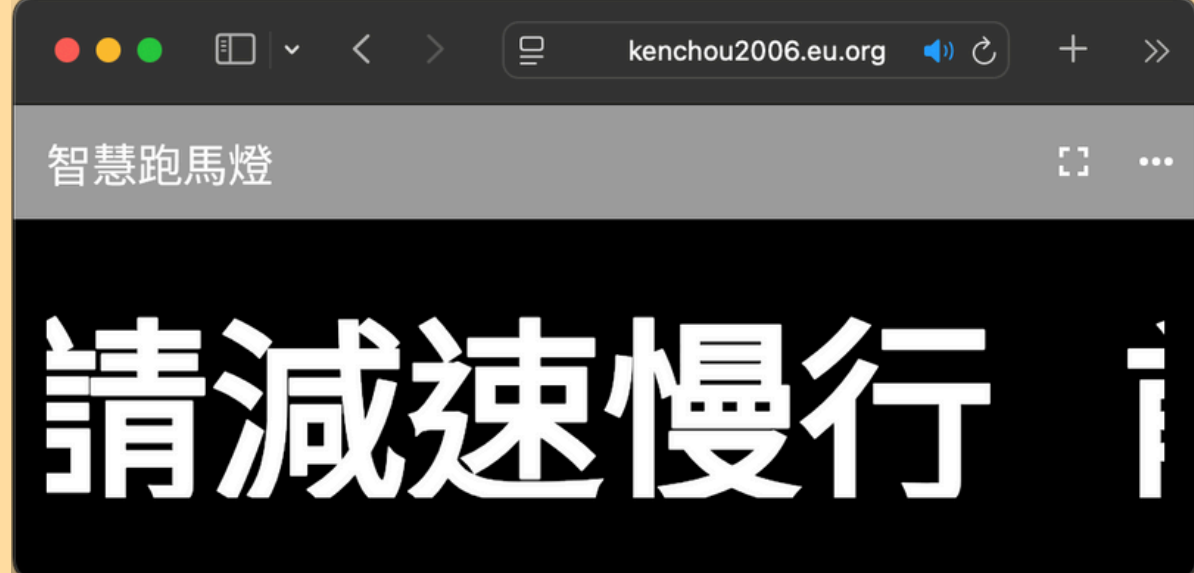


圖12：跑馬燈

應用程式會接收並顯示最新訊息。

Source : 自行製作

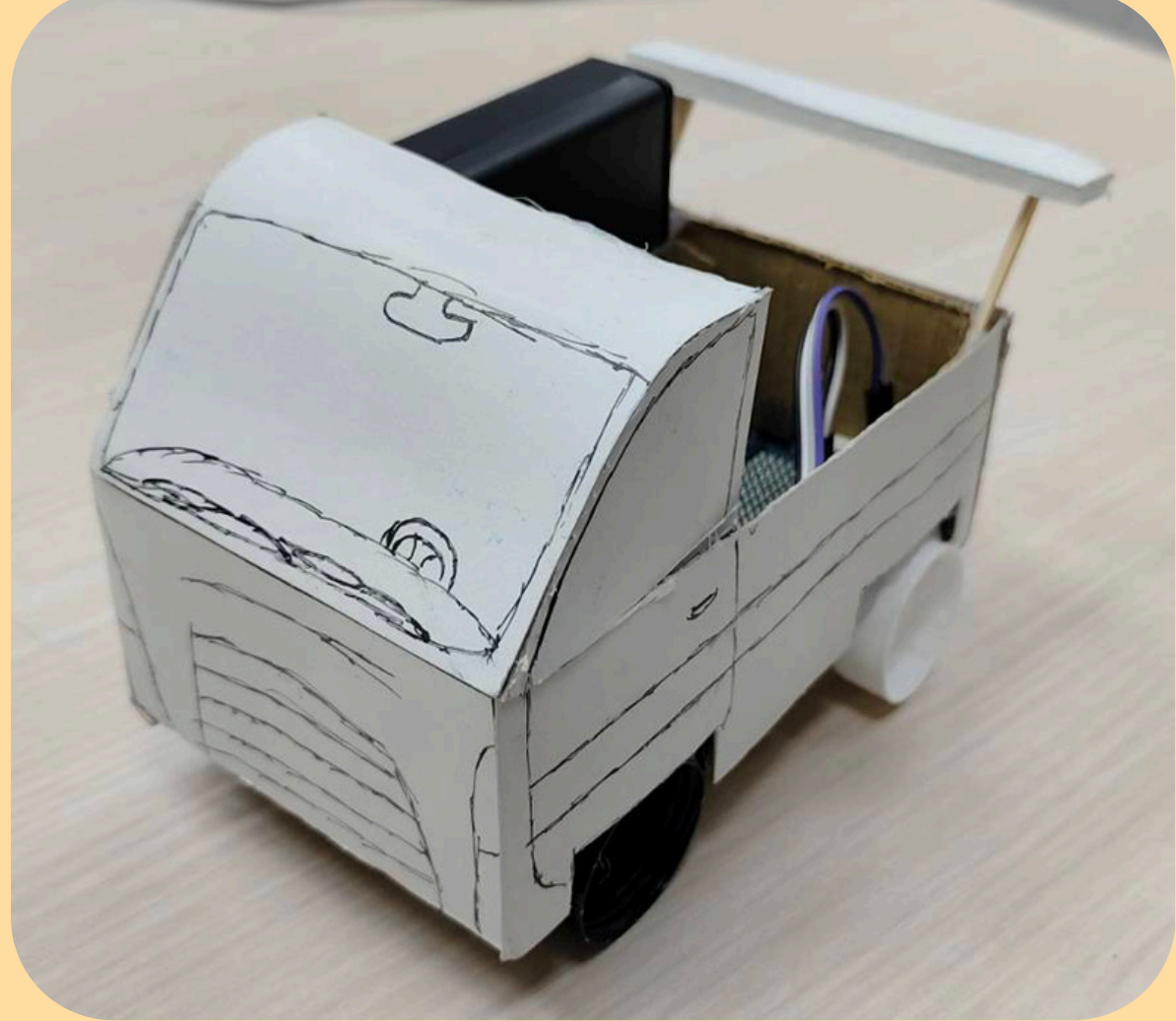


圖13：模擬車

利用加速度感測器檢測車輛撞擊，檢測到撞擊時開啟警示牌

# 陸、結論

## 一、本作品優點

1. 減少”車禍後”可能發生二次事故的機率
2. 藉由雲端化設計及物聯網應用，讓使用更為便捷
3. 透過使用紀錄統計，使交通部了解道路狀況
4. 作為公共設施，無需車主購買，可減少資源浪費
5. 自動檢測事故的發生，並自動開啟警示牌
6. 即時路況顯示，利用文字播報路況

## 二、未來展望

目前的操作介面為網頁，若在實際需要提供民眾使用的情況下，我們希望可以結合各縣市推出的App(台北通)或是車廠App(My Toyota)，讓民眾的使用更便捷。藉由以上App已經填寫實名資料的特性，同時能夠紀錄警示牌使用者，進而防止警示牌遭到濫用。

Source : TaipeiPASS,Toyota

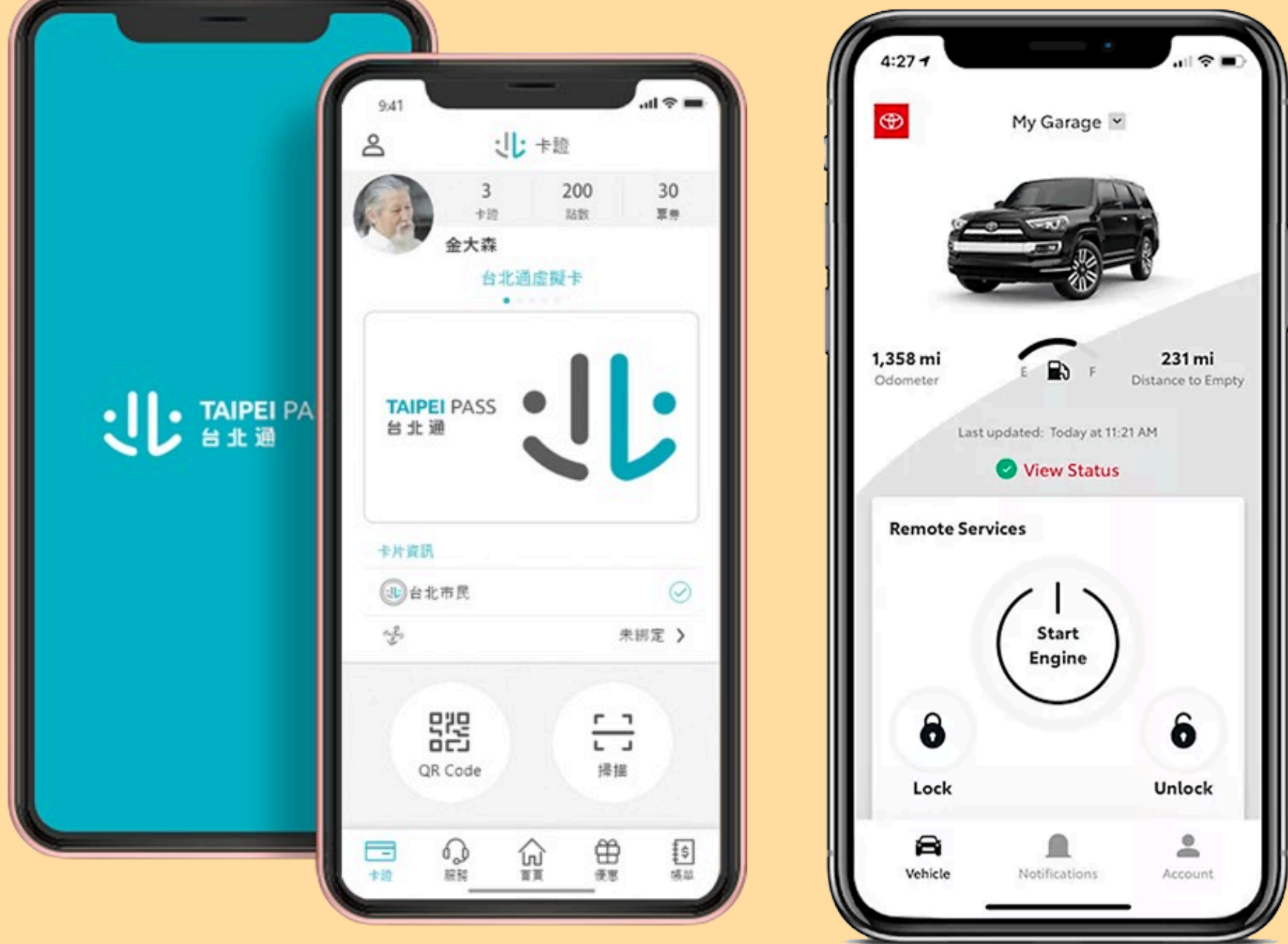


圖14：現有App



研究日誌及專案連結