

智慧駕駛安全監測手環

An Intelligent Wearable System for Driving Safety Monitoring

Source Code: https://github.com/DennisLee03/mcu_final

Demo Video: <https://youtu.be/p8hfKhod48c>

Group: 24

Authors: 詹博堯、劉奕辰、王煒翔、李和穆、張品堯

| | |
|---|----------|
| 智慧駕駛安全監測手環..... | 1 |
| 1. 系統功能與原理說明..... | 3 |
| 2. 系統使用環境及對象..... | 4 |
| 3. 系統完整架構圖、流程圖、電路圖、設計..... | 5 |
| 4. 系統開發工具、材料及技術..... | 6 |
| ● 硬體材料..... | 6 |
| ● 開發工具..... | 6 |
| ● 技術..... | 6 |
| 5. 周邊接口或 Library 及 API 使用說明..... | 8 |
| 周邊接口..... | 8 |
| Library..... | 8 |
| API..... | 8 |
| 6. 實際組員之分工項目..... | 8 |
| 7. 遇到的困難及如何解決..... | 9 |
| PIC18F4520 I2C 韌體實作複雜與初始化挑戰..... | 9 |
| 通訊速率不足導致 MAX30102 FIFO 數據溢出..... | 9 |
| ESP32 WiFi module 無線傳輸產生之延遲..... | 9 |

1.系統功能與原理說明

本系統整合了物聯網 (IoT) 與穿戴式生理監測技術，主要提供以下三大核心功能：

1. 即時疲勞與壓力監測(常駐功能)：

- 系統預設為背景運作模式，透過光學心率感測器持續蒐集駕駛的生理訊號。
- 系統透過光學感測器 (PPG) 取得 R-R 間隔，利用 rMSSD 演算法分析心率變異度 (HRV)。HRV 數值反映了自律神經系統的平衡狀態：當駕駛疲勞時，副交感神經活性降低，導致 HRV 數值顯著下降。系統以此生理指標量化精神負荷。
- 一旦偵測到 HRV 數值異常下降(顯示疲勞或壓力過大)，系統將發出警示 (電腦將提示色從綠色改成紅色)。

2. 主動式酒精濃度檢測(按需功能)：

- 考量感測器特性與節能，系統設有實體啟動按鈕。駕駛在開車前或懷疑有酒精殘留時，可按下按鈕啟動酒測模式。對著感測器呼氣後，系統即時判讀呼氣酒精濃度，並判斷是否達到危險標準。

3. 雲端數據同步與遠端管理：

- 手環蒐集到的生理數據與酒測結果，會透過 Wi-Fi 模組即時上傳至後端伺服器。車隊管理者可透過監控儀表板 (Dashboard) 查看所有駕駛的即時狀態(如：心率、疲勞指數、酒精反應)，實現數據化的車隊安全管理。

2. 系統使用環境及對象

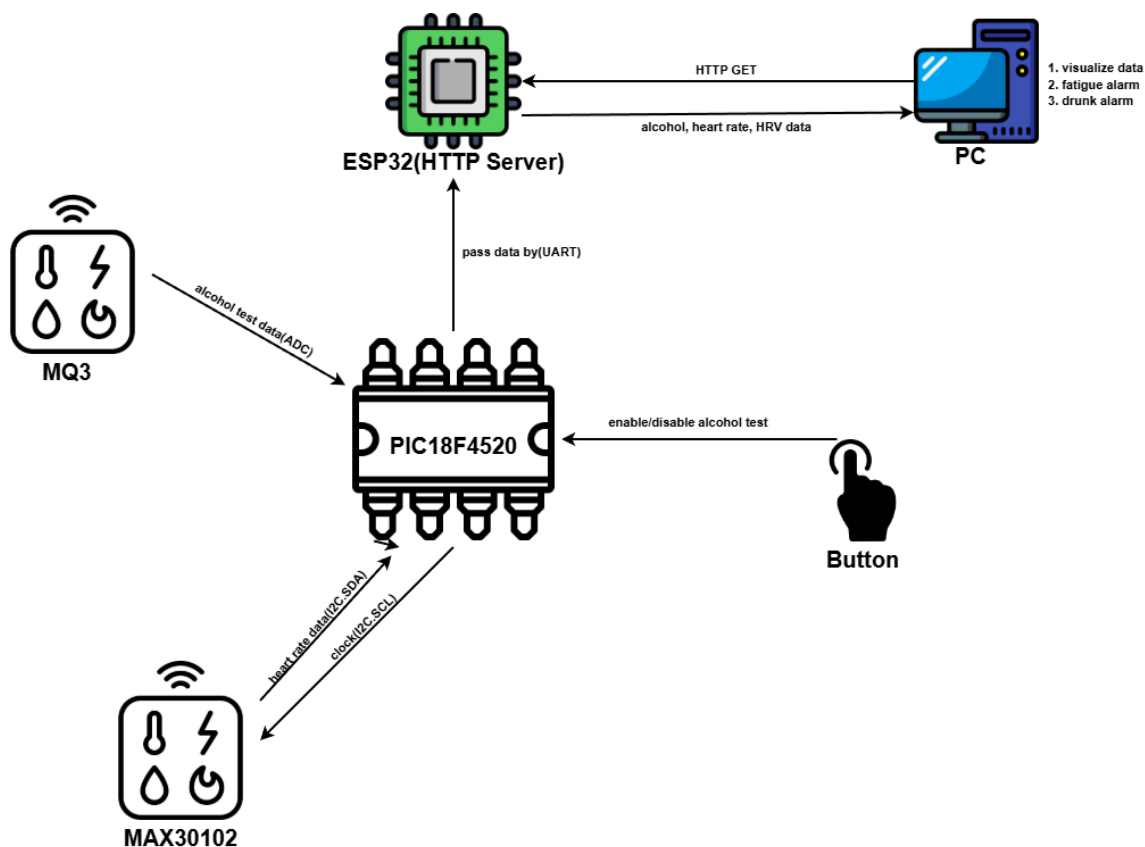
1. 適用對象 (Target Audience): 主要針對需要長時間專注駕駛、且肩負行車安全責任的族群，以及需要管理駕駛狀態的單位。

- **職業駕駛員：**包括長途貨運司機、客運/公車司機、物流配送員或計程車/網約車駕駛。這類族群長時間處於單調的駕駛環境，最容易產生疲勞，且其精神狀態直接影響公共安全。
- **車隊管理中心/管理者：**物流公司、客運業者或租車公司的管理人員。他們需要透過後端儀表板 (Dashboard) 遠端監控多位駕駛的即時生理數據 (心率、疲勞指數) 與酒精檢測結果，以進行調度或安全介入。
- **一般長途駕駛者：**關注自身行車安全，希望透過穿戴裝置即時了解自身精神負荷與酒精殘留狀況的一般車主。

2. 系統使用環境 (Usage Environment): 描述系統運作的物理場景與網路需求。

- **車輛駕駛座艙內：**使用者在駕駛過程中配戴手環。環境可能伴隨車輛震動，系統需在背景持續運作，透過光學感測器蒐集生理訊號。
- **行車前準備區域：**在發動引擎上路前，駕駛可於靜止狀態下按下按鈕進行主動式酒精呼氣測試，確保無酒精殘留。
- **具備無線網路覆蓋的環境：**由於系統使用 ESP32 模組透過 Wi-Fi 進行資料上傳，因此使用環境需具備車載 Wi-Fi、手機熱點或定點 Wi-Fi，以利電腦 (PC) 向 ESP32 模組請求資料並顯示。
- **遠端監控端：**管理者可以使用電腦 (PC) 或行動裝置，透過網頁瀏覽器或應用程式觀看後台顯示介面，接收視覺化的駕駛狀態數據與警示。

3.系統完整架構圖、流程圖、電路圖、設計



4. 系統開發工具、材料及技術

● 硬體材料

| 項目 | 型號/規格 | 用途 |
|--------|----------------------|-----------------|
| 微控制器 | Microchip PIC18F4520 | 核心運算、感測器控制 |
| 心率感測器 | MAX30102 | 擷取 PPG 訊號、傳給MCU |
| 酒精感測器 | MQ-3 | 偵測呼氣酒精濃度 |
| 無線通訊模組 | ESP32 | Wi-Fi 連網、數據上傳 |
| 其他 | 按鈕、電阻 | 觸發酒測、電路整合 |

● 開發工具

- **整合開發環境 (IDEs):** MPLAB X IDE v5.20, PlatformIO, vs code
- **編譯器:** XC8 Compiler
- **直譯器:** Python Interpreter

● 技術

- **按鈕Interrupt:** 當按鈕按下時，系統會立即開始或中斷酒測程序，採集 MQ-3 數據。
- **ADC 類比數位轉換:** 將 MQ-3 的類比電壓訊號轉換為 10-bit 數位數值。
- **UART 串列通訊:** 實現 PIC 與 ESP32 之間的資料傳輸及 AT 指令控制。
- **I2C 通訊協定:** 用於 PIC 與 MAX30102 之間的高速數據交換。
- **PBA (Peripheral Beat Amplitude) 算法:** 透過直流電平估計DC Estimator去除基線漂移，再以Low-Pass Filter抑制高頻雜訊，強化脈搏波形，進而準確偵測心跳波峰。由此可穩定取得 R-R 間隔 (IBI)，作為HRV 分析的可靠基礎。
- **HRV 演算法 (rMSSD):** 在 MCU 上實作均方根差公式，維護計算相鄰心跳間距

變異，以量化副交感神經活性。

- **WIFI 通訊：**利用 ESP32 模組，通過 TCP/IP 協定建立網路連線。數據以 JSON 格式封裝後，透過 HTTP GET 請求發送至遠端後台伺服器，實現車隊駕駛狀態的即時監控。

5. 周邊接口或 Library 及 API 使用說明

周邊接口

- **MSSP (I2C Mode):** 設定為 Master Mode, 時脈 400kHz, 用於讀取 MAX30102 的 FIFO 暫存器 (0x07) 以獲取紅外光數據。
- **ADC Module:** 設定通道 AN0, 參考電壓 V_{ref+} 為 5V, 用於讀取酒精感測器數值。
- **UART:** 設定 Baud Rate 115200, 用於與 ESP32 通訊。

Library

- [SparkFun_MAX3010x_Sensor_Library](#): 提供心跳判斷函式

API

- ESP32 透過 HTTP GET 請求將資料傳送至後端

6. 實際組員之分工項目

| 名字 | 分工 |
|-----|------------------------|
| 詹博堯 | PIC18 主控與周邊整合/HR/HRV計算 |
| 劉奕辰 | MQ-3 酒精感測整合 |
| 王煒翔 | MAX30102 疲勞/HRV 感測整合 |
| 李和穆 | 無線傳輸/HR/HRV計算 |
| 張品堯 | 後台顯示界面 / 資料視覺化 |

7.遇到的困難及如何解決

PIC18F4520 I2C 韌體實作複雜與初始化挑戰

- 缺乏現成的函式庫，所有 I2C 通訊和 MAX30102 的設定細節均需自行在韌體層面實現。
- 初期難以判斷是 I2C 時序錯誤還是 MAX30102 暫存器配置錯誤。

解決方法

- 參考 MAX30102 的官方Datasheet 與網路上印度老哥影片，手動編寫 PIC 的 I2C Master 驅動程式，並建立標準化的初始化函式(max30102_write_reg, max30102_read_reg 等)，確保所有關鍵暫存器(如 FIFO Config, SpO2 Config 等)均能正確寫入。
- 透過 UART 輸出除錯訊息，逐步驗證各設定流程與通訊結果。

通訊速率與運算效能不足導致 MAX30102 FIFO 數據溢出

- HRV 分析需要高採樣率(400Hz)，意即每 2.5ms 需完成一次數據採集與處理。然而，PIC18F4520 最高運作於 8MHz 時脈下。若同時執行軟體 I2C 讀取、ADC 轉換、複雜的 rMSSD 浮點運算以及 UART 傳輸，CPU 資源容易被耗盡，導致 I2C 時序延遲，資料來不及讀出，塞滿 MAX30102 的內部緩衝區(FIFO)造成溢出與數據遺失。

解決方法

- **提升時脈**：將 I2C 頻率由預設值(100kHz)提升至 **400 kHz (Fast Mode)**。
- **優化讀取**：改寫讀取邏輯，由單次讀取改為**批量讀取**，大幅減少通訊延遲。
- **調升Baud Rate**
- 確認 PIC 算力極限後，**改變系統架構**。將 PIC 定位為「即時 I/O 控制器」，僅負責採集 Raw Data；將複雜運算移交給ESP32 (80MHz)。

分散式運算架構與無線傳輸優化

- 會將運算交給esp還有一個原因: 若將未處理的原始波形數據直接透過 ESP32 WiFi 傳送到 PC 端計算, 會因無線網路的不確定性產生傳輸延遲 (Latency), 導致 PC 端顯示的波形與數值不同步, 且容易因封包堆積造成即時監控失效。

解決方法

- 將高時間解析度之訊號處理與心率(BPM)計算使用 ESP32 端執行, 僅將計算後的結果與趨勢資料傳輸至 PC 端進行顯示。此方式可避免無線傳輸延遲對即時訊號處理的影響, 同時提升系統整體穩定性與資料可靠度。