Indicators Control Over Mobile Phones (using BLE Protocol)

Name: Uppula Vivekananda Raman

Institute: Sreenidhi Institute of Science and Technology, Hyderabad.

Submission Date: 21-07-2025.

# 1. High-Level Architecture and Software Requirement Specification

This project implements an **ESP32-based Vehicle Indicator Control System**, which allows controlling **left**, **right**, and **hazard** indicator lights through **push-button inputs**. The system is developed in **C** using a modular structure split between **driver and application layers**, and follows the assignment title:  
**“Indicators Control Over Mobile Phones (using BLE Protocol)”** *(BLE integration to be added in future)*.

Key Features Implemented:

* 1-second button hold detection using **hardware timers** (no FreeRTOS timers used).
* **Left / Right Indicator Toggle**: Hold the respective button for 1 second to turn ON/OFF.
* Pressing opposite button switches indicator.
* **Hazard Mode:** Hold both buttons for 1 second to toggle hazard blinking mode.
* **Hazard Exit**: Hold either button for 1 second while in hazard mode to turn off all indicators.
* 300ms LED Blinking logic (300ms ON, 300ms OFF) **using software timer logic**.
* 100ms **Scheduler** implemented via **FreeRTOS** tasks for base and application software updates.
* **UART Logging of:**

1. Button press and release events
2. Indicator mode transitions
3. LED ON/OFF status with real-time updates.

* **Hardware Abstraction Drivers:** 
  + gpio\_driver: For reading push-button inputs and initializing **GPIOs**.
  + pwm\_driver: Controls LED brightness and toggling using **ESP32 LEDC PWM.**
  + hw\_timer\_driver: hardware timer for detecting 1-second button holds.
* **TTL-Based UART Logging**: Implemented using **HW-417-V1.2 USB to TTL Converter** and **Tera Term.**
  + Full serial logs are captured externally for debugging and documentation.

(UART logs saved via Tera Term are included in project submission.)

# 2. GitHub Repository

<https://github.com/vivekanandaramanu/esp32-indicator-system>

# 3. Google Drive Video Recording

<https://drive.google.com/file/d/1q4hKbzImOk4m6uuVi8Ycv6JALjvGyG4a/view?usp=sharing>

# 4. UART Log File

Log captured using HW-417-V1.2 USB to TTL module and Tera Term.

<https://github.com/vivekanandaramanu/esp32-indicator-system/blob/main/indicator_log.txt>

OR

<https://github.com/vivekanandaramanu/esp32-indicator-system>

See: indicator\_log.txt

# Note on MATLAB Simulink Integration

Although the assignment specifies Simulink-based application logic generation and integration, this part has not yet been completed. The logic has been modularly implemented in the **“indicator\_app.c”** file for now. The project is structured such that future integration of Simulink-generated code into the application layer will be straightforward. This will be updated in the documentation once completed.

# Known Limitations

- Button-based indicator switching mechanism can be further improved using **button latching** to handle overlaps better.  
- Hazard mode toggling while buttons are pressed simultaneously may need timing fine-tuning.  
- The current solution meets functional requirements but will be improved further post-deadline.