

**PROJECT PROPOSAL: SIMULATED TRAFFIC MANAGEMENT SYSTEM**

# PROJECT OVERVIEW:

We propose the development of a Simulated Traffic Management System that controls traffic lights efficiently based on pedestrian crosswalk requests. The system will use an STM32F103C8 microcontroller and various components, simulated within the Proteus environment, to manage traffic lights for both vehicles and pedestrians. Additionally, we will implement this system using FreeRTOS to ensure real-time operation and efficient multitasking.

# OBJECTIVES:

## The primary objectives of this project are to design, build, and demonstrate a Simulated Traffic Management System that can:

1. Control traffic lights for cars and pedestrians within the Proteus simulation environment.
2. Efficiently respond to pedestrian crosswalk requests.
3. Implement error handling and reporting.
4. Utilize FreeRTOS for real-time operation and multitasking.

## HARDWARE AND SOFTWARE REQUIREMENTS:

### Hardware Requirements (Simulated within Proteus):

* STM32F103C8 microcontroller simulation.
* Virtual push button for pedestrian requests.
* Virtual LEDs for traffic lights - Red, Yellow, and Green for cars and pedestrians.

### Software Requirements:

* In normal mode (simulation):
* Traffic lights for cars will cycle every five seconds: Green, Yellow, Red, Yellow, Green.
* The Yellow LED will blink for five seconds before transitioning to Green or Red.
* In pedestrian mode (simulation):
* Transition to pedestrian mode when the virtual pedestrian button is pressed.
* Pedestrians can cross the street when the pedestrian's Green LED is on.
* Pedestrians must wait when cars have the right of way.
* Specific sequences for pedestrian crosswalk requests.

# SYSTEM DESIGN

* Define system layers.
* Define system drivers.
* Place each driver into the appropriate layer following SOLID principles.
* Define APIs for each driver with documentation.
* Define new data types for drivers.
* Define error handling and reporting mechanisms.

# DEVELOPMENT ENVIRONMENT

* Configure Proteus simulation environment.
* Implement layered architecture in the Proteus project structure.
* Prepare all files for development with header file guards.
* Write typedefs and function prototypes.
* Include lower layer drivers in upper layer/calling drivers' header files.

# IMPLEMENTATION

* Write skeleton code for each function with comments.
* Convert comments into appropriate code.
* Ensure functions return error states.

# TESTING

* Develop test modules for each driver.
* Implement and validate driver output within the Proteus simulation.

# SYSTEM FLOW:

* Define the system flow based on requirements.
* Implement the system flow in the `APP\_start` function.

# USER STORIES AND TESTING

* Define user stories for pedestrian crosswalk requests.
* Implement and test user stories within the Proteus simulation.

# DELIVERABLES

1. PDF document containing:

* System description.
* System design.
* System flow chart or state machine.
* System constraints (if any).

1. Video presentation discussing the project.

# TIMELINE

* Day 1-2: System design and Proteus environment setup.
* Day 3-4: Software design and architecture.
* Day 5-6: Development within the Proteus simulation.
* Day 7: Testing and documentation.

# BUDGET

The estimated budget for this project includes any licensing costs associated with Proteus simulation software.

# CONCLUSION

This project aims to create an efficient Simulated Traffic Management System that handles both vehicle traffic and pedestrian crosswalk requests within the Proteus simulation environment. By implementing this system with FreeRTOS, we ensure real-time operation and responsiveness. The final product will contribute to safer and more organized traffic management, even in a virtual setting.