

Project Documentation: Traffic Light Control System

Author: Adel Hani

Objective:

The project is a Traffic Light Control System designed to control the traffic lights for two streets, along with pedestrian signal management. The system operates in two modes:

1. **Normal Mode** (Traffic light control for cars)
2. **Pedestrian Mode** (Allowing pedestrians to cross by stopping cars)

The system is based on two main streets, each with separate lights for cars and pedestrians.

System Description:

Components Used:

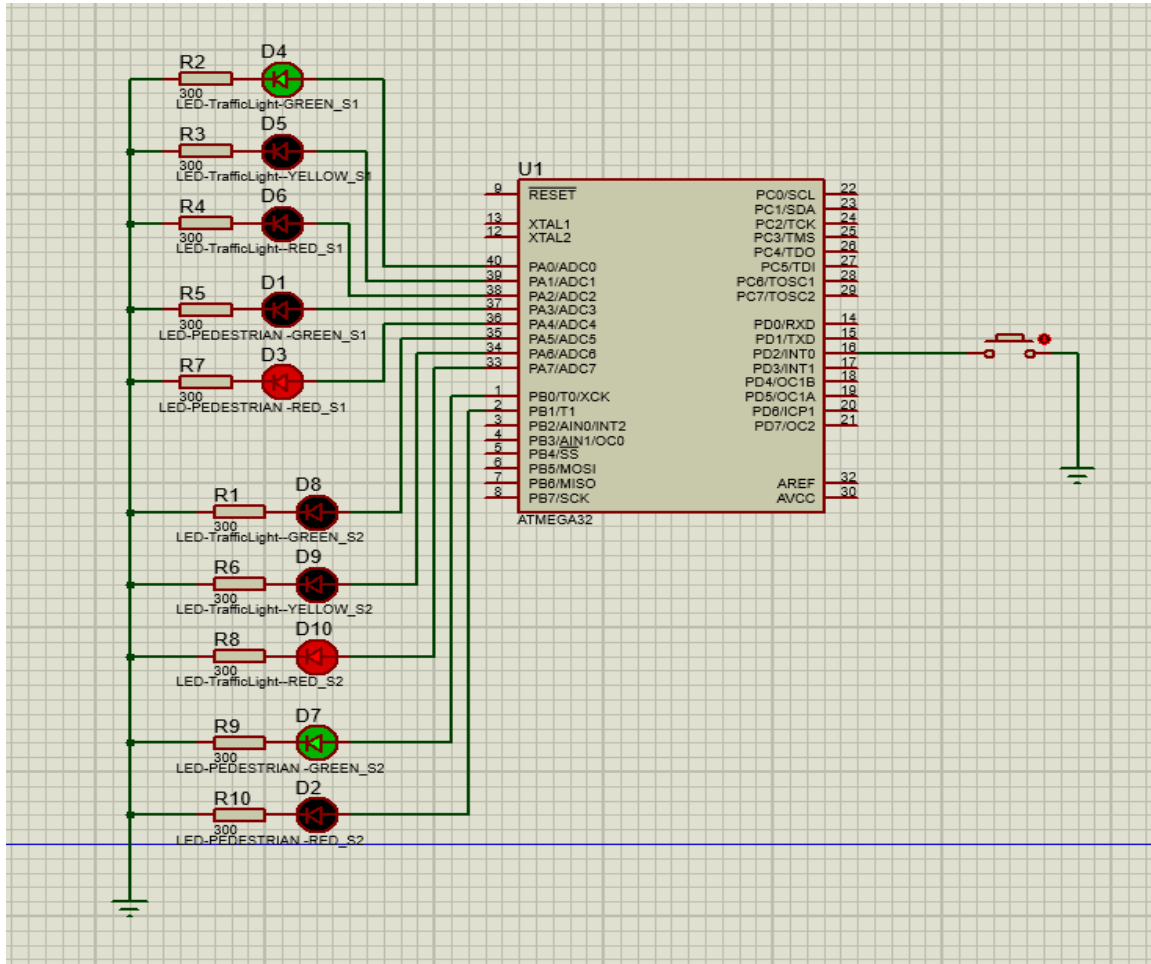
- **LEDs:** Represent traffic lights for both streets and pedestrian signals.
- **Timer:** Used to control delays between state transitions.
- **Interrupts:** Used to detect pedestrian button press to trigger pedestrian mode.

Main Features:

1. **Traffic Light Management:** Controls the car and pedestrian traffic lights based on two modes (Normal and Pedestrian).
2. **Mode Switching:** The system can switch between Normal Mode (for car traffic) and Pedestrian Mode (for pedestrians).
3. **Interrupt Handling:** The system handles external interruptions triggered by the pedestrian button.

Hardware Simulation:

The entire system was simulated using **Proteus** to visualize and simulate the behavior of the traffic lights. The components like LEDs, external switches, and microcontroller were all implemented to verify the behavior before moving to actual hardware.



State Diagram (Finite State Machine Design):

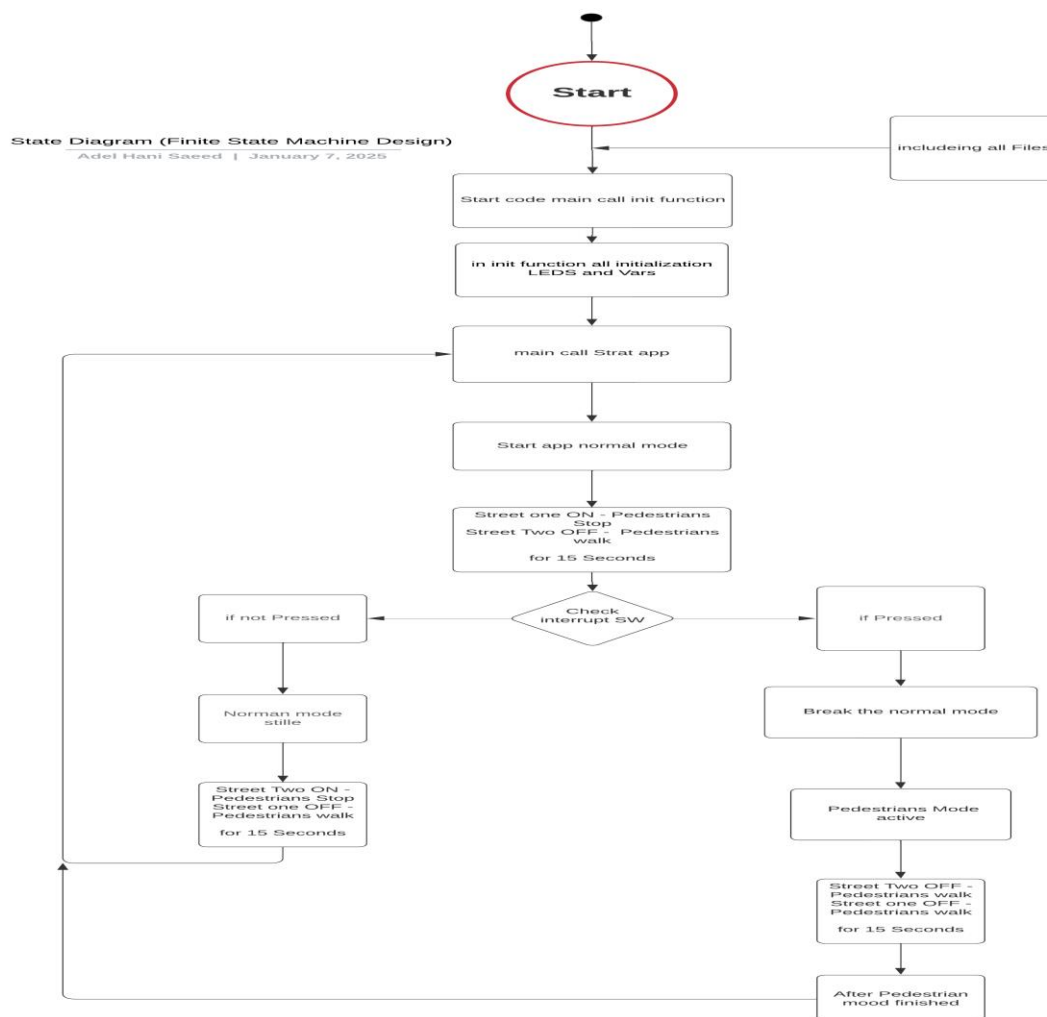
The Traffic Light Control System operates as a Finite State Machine (FSM) with the following states:

1. **Normal Mode** (Cars move, Pedestrians stop)
2. **Pedestrian Mode** (Cars stop, Pedestrians move)

State Transitions:

- **Normal Mode:**
 - In this state, the car lights on Street 1 turn green while pedestrian lights are red.
 - After a defined delay, the system switches the car lights to yellow, and pedestrian lights turn green.
 - After another delay, the system switches to the active street (between Street 1 and Street 2) and continues the cycle.
- **Pedestrian Mode:**
 - In this state, both car signals are turned off (red), and pedestrian signals for both streets turn green, allowing pedestrians to cross.
 - After a delay, the system switches back to Normal Mode, resuming the car traffic cycle.

State Diagram: A state diagram would illustrate the above transitions between **Normal Mode** and **Pedestrian Mode**, with the respective actions for car and pedestrian lights



Code Breakdown:

Application Layer (APP):

- **Initialization:** The system initializes LEDs, timers, interrupts, and the external pedestrian button.
- **State Management:** The APP_voidStartTrafficLight() function handles the state transitions, where it checks the current mode and switches between Normal and Pedestrian modes.

Interrupt Handling:

- The external interrupt (EXTI_voidINT0Init()) is configured to listen for the pedestrian button press. When pressed, it switches the system to Pedestrian Mode by invoking the callback function ButtonEXTI_voidTrafficLightState_Street().
-

Key Functions:

1. APP_voidInit(): Initializes the system components (LEDs, timers, and interrupts).
2. APP_voidStartTrafficLight(): Main loop that controls the traffic light system, switching between Normal and Pedestrian modes.
3. ButtonEXTI_voidTrafficLightState_Street(): Interrupt callback function that switches to Pedestrian Mode when the button is pressed.
4. Traffic light control functions for both car and pedestrian LEDs (e.g., Car_voidGREENLED_S1(), PEDESTRIAN_voidGREENLED_S1(), etc.).

Conclusion:

The system successfully controls traffic lights for two streets, switching between car and pedestrian modes based on predefined timings and external button presses. The design has been implemented and tested on **Proteus**, with the state transitions clearly defined.