Intelligent Traffic Light Controller with Day/Night Mode and Pedestrian Support

Abstract

This project addresses the inefficiency of fixed-timing traffic signals by designing an adaptive traffic light controller using the 8051 microcontroller. The system simulates a four-way intersection with dedicated phases for North, South, East, and West traffic, with right-turn and straight-green signals. To improve efficiency, the system integrates active-high vehicle detection buttons on P3.0–P3.3 that extend green light duration dynamically when vehicles are present, reducing waiting time. Pedestrian safety is ensured via request buttons on P3.4 and P3.5 that trigger the dedicated walk signals (on P3.6 and P3.7) with a walk phase. The system autonomously switches between Day Mode (6:00 AM–10:00 PM) and Night Mode (flashing yellow) using an internal software-based real-time clock synchronized via a 10ms Timer 0 interrupt. Implemented entirely in 8051 assembly language using keil and proteus, the project demonstrates core concepts such as interrupt handling, I/O control, state management, and timing. This results in a robust, energy-efficient prototype that significantly improves traffic flow and pedestrian safety over conventional fixed-timer systems.

Table of Contents

1. Introduction

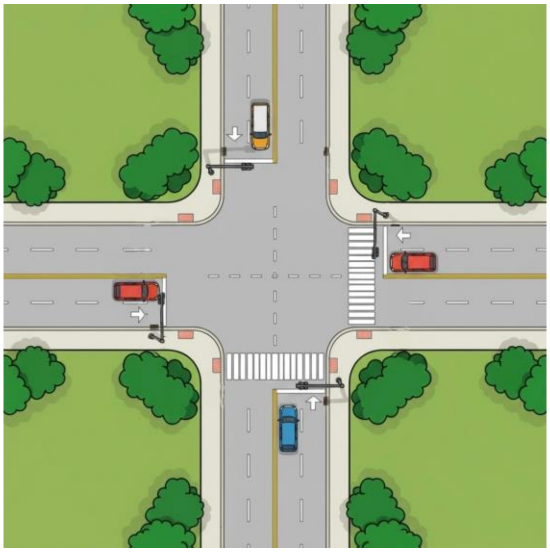
1.1 Objective of the Project

The primary objectives of this project are:

1. To design a 4-way traffic light controller with realistic green-yellow-red sequencing.
2. To implement automatic day/night mode based on a simulated 24-hour clock (day: 6 AM–10 PM, night: 10 PM–6 AM).
3. To integrate pedestrian crossing with request buttons and don't-walk signals that de-activate only during safe traffic phases.
4. To simulate a real-time clock using Timer 0 interrupts, enabling autonomous time-based operation without external hardware.
5. To implement IR sensors for real time traffic control.

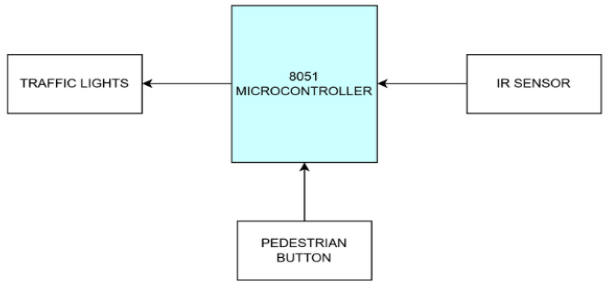
1.2 Project Motivation

Urban traffic congestion and inefficient signal systems are significant problems in modern cities. Fixed-timer traffic lights often cause unnecessary delays during low-traffic periods or cause unnecessary congestion during high-traffic periods. This project was motivated by the desire to demonstrate how embedded systems like the 8051 can be used to create a smarter, more responsive solution. Inspired by research utilizing technologies like Infrared (IR) sensors for vehicle detection, this project implements a simulated version of the same by using buttons as inputs. By implementing adaptive logic and time-based automation, we aimed to build a proof-of-concept that mimics real-world intelligent transportation systems, and reinforce our understanding of microcontroller programming, interrupt handling, and hardware interfacing.



2. Implementation

2.1 Block Diagram



- P1: Controls North (L1) and East (L2) traffic lights
- P2: Controls South (L3) and West (L4) traffic lights
- P3.0-P3.3: IR simulation buttons
- P3.4 / P3.5: Pedestrian request buttons
- P3.6 / P3.7: Pedestrian Light (N-S and E-W)

2.2 Explanation of Components

- 8051 Microcontroller (AT89C51): The central processing unit that executes the assembly code, manages I/O operations, handles timer interrupts, and controls the overall system state.
- LEDs (Vehicle & Pedestrian): Represent traffic lights (Red, Yellow, and Green) and pedestrian signals (Walk/Don't Walk). Connected to P1 and P2 for vehicle lights and P3.6/P3.7 for pedestrian lights.
- Push Buttons: Simulate vehicle presence (P3.0-P3.3) and pedestrian requests (P3.4-P3.5).
- Crystal Oscillator (12 MHz): Provides the master clock signal for the microcontroller's timing.

The 8051 interacts with these components by writing SETB/CLR instructions to set output pins as HIGH/LOW for LEDs, and reading the state of input pins using JB/JNB instructions for the buttons.

2.3 Implemented Code

The complete assembly code is implemented on the AT89C51. It features:

- Timer 0 ISR for 10ms interrupts to maintain a real-time clock.
- Day/Night mode logic based on the simulated hour.
- Dynamic green light extension based on vehicle button presses.
- Pedestrian request detection and safe crossing phases.
- Sequential control of traffic lights with yellow transitions.

(Note: Full code is attached in the appendix of the submitted PDF.)

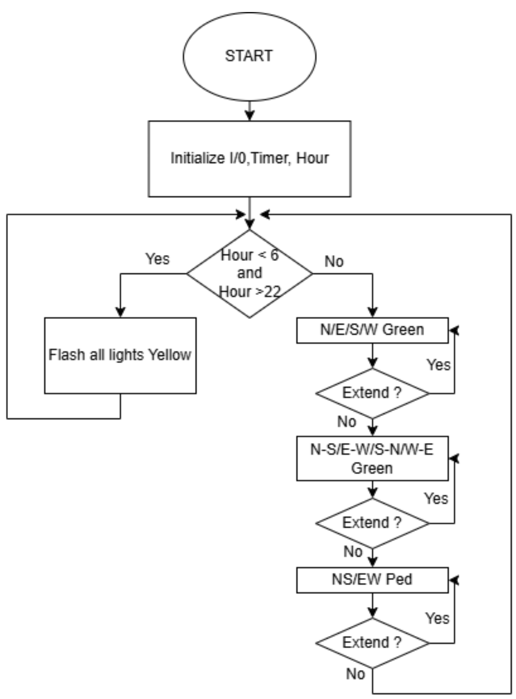
2.4 Hardware Specifications

- Microcontroller: AT89C51 (8051 core)
- Crystal Oscillator: 12 MHz
- LEDs: 16 total (4 per arm × 4 arms + 2 pedestrian)
- Resistors: 10kΩ pull-ups for buttons
- Push Buttons: 6 (SPST)

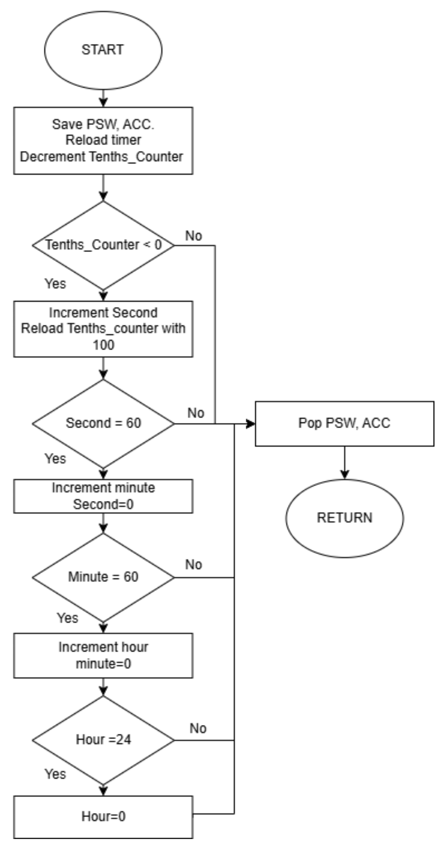
2.5 Software Specifications

- Programming Language: 8051 Assembly
- IDE: Keil µVision 5
- Assembler: Keil A51
- Simulation Tool: Proteus 8 Professional

2.6 Flowchart/Algorithm



ISR



3. Conclusion and Results

3.1 Test Cases

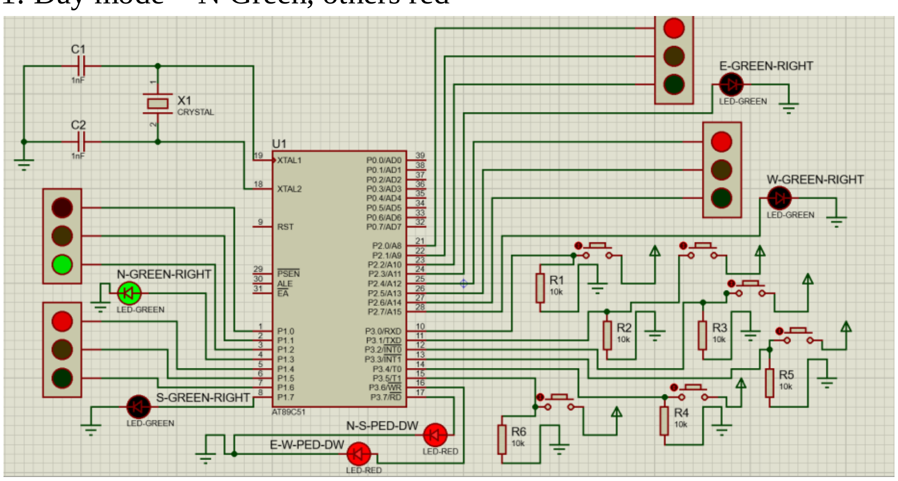
TEST CASE	INPUT CONDITION	EXPECTED OUTPUT
TC1	Hour = 05	Night mode: Flashing Yellow
TC2	Hour = 10	Day mode: 4-phase cycle
TC3	Press P3.0 during N phase	Time of N Green extend
TC4	Press P3.4 during any phase	All lights Red, P3.6 off

3.2 Results

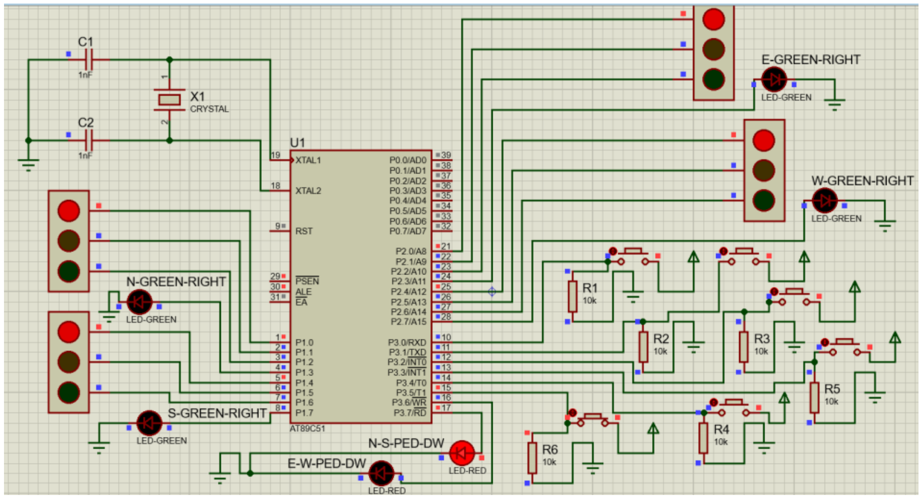
- The hardware setup and main program logic were successfully demonstrated.
- The simulated vehicle detection system (using buttons P3.0-P3.3 to mimic IR sensors) correctly extended the green light duration when the corresponding button was pressed.
- The pedestrian crossing logic worked as intended when tested independently.
- However, the Timer 0 Interrupt Service Routine (ISR) did not function as expected. The Hour variable failed to increment reliably, preventing the system from automatically transitioning between day and night modes based on the simulated time. This indicates a fundamental issue with the interrupt setup or execution, which requires further research into the 8051's interrupt vector handling, timer configuration, and register preservation (PUSH/POP) within the ISR.
- The system currently operates in Day Mode continuously unless manually reset.

3.3 Screenshots

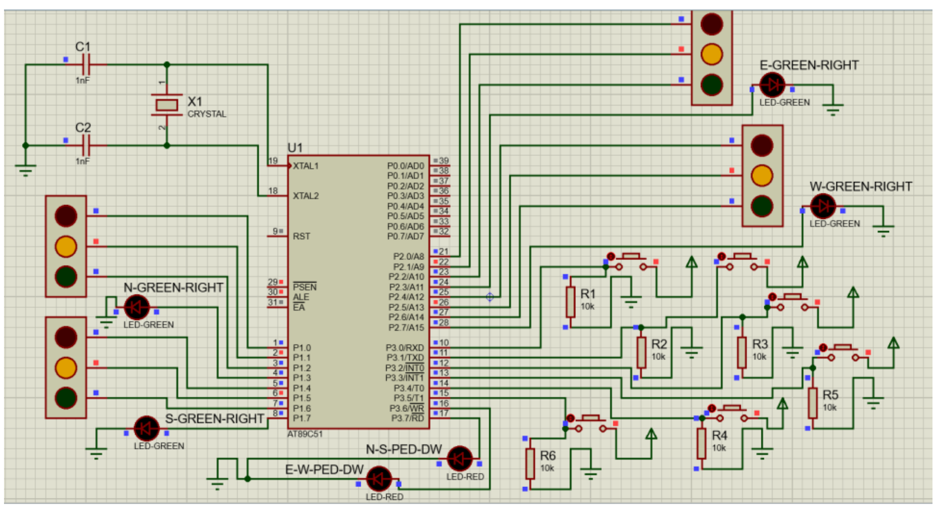
- Screenshot 1: Day mode – N Green, others red



- Screenshot 2: Pedestrian WALK signal active



- Screenshot 3: Night mode – all yellows flashing



4. Conclusion

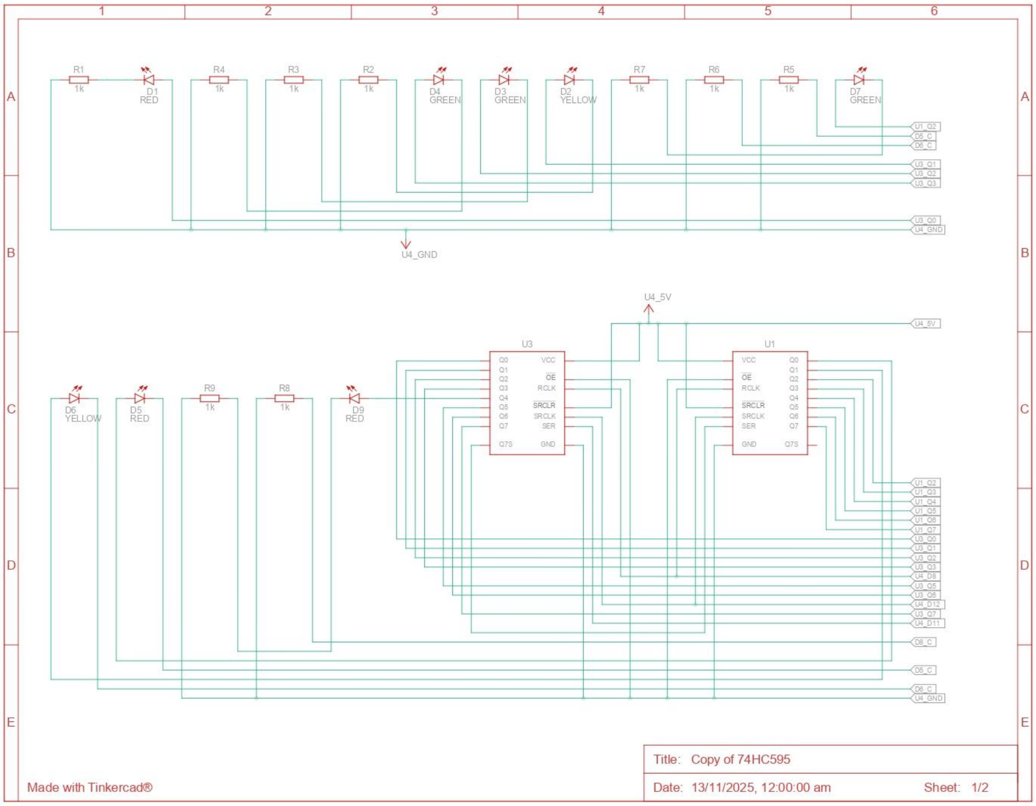
This project successfully developed an intelligent traffic light controller prototype using the 8051 microcontroller. The system effectively demonstrates adaptive traffic control through dynamic green light extension simulated via buttons (P3.0-P3.3) replacing IR sensors, and provides safe, demand-based pedestrian crossings. The core logic for managing multiple traffic phases and handling inputs is functional. However, the critical feature of autonomous day/night mode switching failed due to the Timer 0 ISR not operating correctly. Despite attempts to debug the interrupt initialization and code structure, the timekeeping mechanism remains non-functional. This highlights a key learning point about the complexity of interrupt-driven programming on legacy microcontrollers. Future work will focus on resolving this issue and potentially integrating actual IR sensors for vehicle detection, to replace the manual button simulation and achieve true environmental responsiveness.

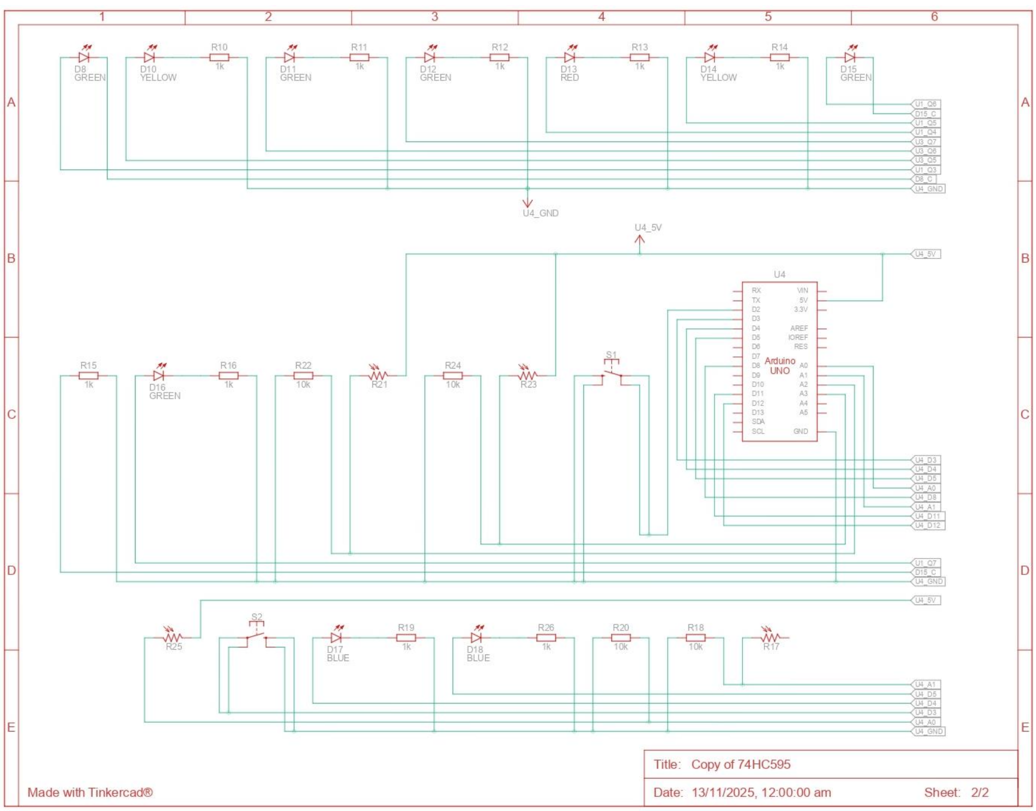
5. Enhancement: Transition to an Arduino-Based System

The limitations of the 8051-based system, particularly its fixed timing and reliance on manual simulation for sensor inputs, can be overcome by transitioning to a modern microcontroller like the Arduino. An Arduino-based system offers significant advantages:

- Simplified Time Management: The millis() function provides non-blocking delays, allowing the system to handle multiple tasks (e.g., checking sensors, updating displays, communicating over Wi-Fi) simultaneously without freezing.
- True Adaptive Control: Instead of simulating IR sensors with buttons, actual IR or ultrasonic sensors can be easily integrated to detect real-time vehicle queue length and dynamically adjust signal timing.
- Connectivity & IoT Integration: With an ESP32 variant, the system can connect to Wi-Fi, enabling features like remote monitoring, centralized control from a web dashboard, and coordination with other intersections ("Green Wave").
- Advanced Data Logging: Vehicle count data can be stored on an SD card or uploaded to a cloud database for city planning and optimization.
- Over-the-Air Updates: Firmware can be updated remotely without physical access to the device.
- Ease of Development: Programming in C++ with the Arduino IDE is more accessible than 8051 assembly, reducing development time and increasing code maintainability.

While the 8051 project laid the foundational groundwork, an Arduino-based implementation would transform it into a truly smart, connected, and responsive urban infrastructure component.


R1 1k
D1 RED
R4 1k
R3 1k
R2 1k
D4 GREEN
D3 GREEN
D2 YELLOW
R7 1k
R6 1k
R5 1k
D7 GREEN
U4_GND
U4_5V
U3
U1
D6 YELLOW
D5 RED
R9 1k
R8 1k
D9 RED
Title: Copy of 74HC595
Date: 13/11/2025, 12:00:00 am
Sheet: 1/2
Made with Tinkercad®


D8 GREEN
D10 YELLOW
R10 1k
D11 GREEN
R11 1k
D12 GREEN
R12 1k
D13 RED
R13 1k
D14 YELLOW
R14 1k
D15 GREEN
U4_GND
U4_5V
U4
Arduino UNO
R15 1k
D16 GREEN
R16 1k
R22 10k
R21
R24 10k
R23
S1
R25
S2
D17 BLUE
R19 1k
D18 BLUE
R26 1k
R20 10k
R18 10k
R17
Title: Copy of 74HC595
Date: 13/11/2025, 12:00:00 am
Sheet: 2/2
Made with Tinkercad®

6. Contributions

Avishka Thombare	1262241191	Report, Keil, Proteus Simulation
Niyati Priyani	1262241237	Presentation, Intro and Objectives
Daksh Agrawal	1262241235	Presentation, Result and Conclusion
Atharva Gadre	1262241081	Tinkercad Implementation

7. References

1. Intel Corporation. (1980). MCS-51 Microcontroller Family User’s Manual.
2. Keil Software. µVision IDE Documentation. https://www.keil.com
3. Proteus 8 Professional: https://www.geeksforgeeks.org/installation-guide/how-to-download-and-install-proteus-software-on-windows/
4. Talib, Z., et al. (2024). Smart Traffic System Using Infrared Sensors-based IoT.
5. Dzulkefli, N. N. S. N., et al. (2020). Density based traffic system via ir sensor. Journal of Physics: Conference Series, 1529(2), 022061.
6. Shanmugasundar, M., et al. (2020). IoT based smart traffic control using IR sensors. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 9(7), 2278-3075.
7. Federal Highway Administration (FHWA), "Non-Intrusive Detection," U.S. Department of Transportation. [Online]
8. N. H. Al-Qassim and K. H. Hamouda, "A Real-Time Traffic Density Estimation and Management System for Smart Cities," in *Proc. 2020 IEEE Int. Smart Cities Conf. (ISC2)*, Sept. 14-17, 2020, pp. 1-6. doi: 10.1109/ISC251055.2020.9242560.