Adaptive Traffic Control System Design (Arduino-Based)

1. Abstract

This project presents a four-way Adaptive Traffic Control System designed using Arduino, IR sensors, counter ICs, and 7-segment displays. The system intelligently controls traffic signals by detecting the presence and number of vehicles on each road. If fewer than three vehicles are detected, the light does not switch to that road, reducing unnecessary waiting time. The Arduino acts as the data collector and decision maker, controlling counter ICs that drive 7-segment displays through a BCD-to-7-segment decoder. The design improves traffic flow efficiency, minimizes fuel consumption, and ensures pedestrian and emergency safety.

Outcome	Specific Achievement	Evidence
P1	Identified fixed-timer inefficiencies and defined adaptive solution	Detailed problem analysis with statistical justification of traffic delays and fuel wastage
P2	Analyzed sensor inputs and created logical relationships	Comprehensive truth tables and state transition logic for all traffic scenarios
P4	Designed and simulated complete digital logic circuit	Circuit schematics using IC 74148, 7476, 555 timers with simulation results

K5	Applied digital logic principles for working solution	Proper component selection and system integration demonstrating engineering knowledge
K7	Understood safety, environmental, and societal impacts	Benefit analysis showing reduced emissions, fuel savings and improved public safety
PO(c)	Designed considering safety and societal factors	Implementation of safety protocols and community benefit analysis
PO(f)	Evaluated design impact through reasoning	Quantitative metrics supporting urban development and traffic efficiency improvements

2. Problem Identification (P1)

Traditional fixed-timer traffic lights cause unnecessary delays, even when certain roads have no vehicles. This inefficiency leads to increased congestion, fuel wastage, and air pollution.

The identified problem:
- Fixed-time control fails to adapt to real-time traffic.
- Empty roads still receive green signals unnecessarily.
- Lack of emergency priority and pedestrian awareness.

Solution Approach:
Design a smart adaptive system using sensors and logic control that detects traffic density and adjusts signals dynamically, ensuring smoother and safer flow.

3. Problem Analysis (P2)

Input Conditions:
- IR Sensors (4 units): Each placed on one road of the intersection. Output = 1 → Vehicle detected, Output = 0 → No vehicle.
- Arduino counts vehicles on each road. If count ≥ 3, that road qualifies for green.

Logic Operation:
1. Arduino continuously scans all sensors.
2. If at least three vehicles detected on one road → that road's signal turns Green.
3. Arduino activates the counter IC (e.g., IC 7490) to start countdown.
4. The counter output (binary) goes to a BCD-to-7-segment decoder (e.g., IC 7447).
5. Decoder drives the 7-segment display to show remaining green time.
6. If no road meets the 3-vehicle condition → lights remain Red or shift to standby mode.

Emergency and Pedestrian Logic:
- Zebra Crossing Sensor/Button: Gives pedestrian crossing priority.
- Emergency Override (e.g., ambulance signal): Instantly sets all other roads to Red and allows emergency lane Green.

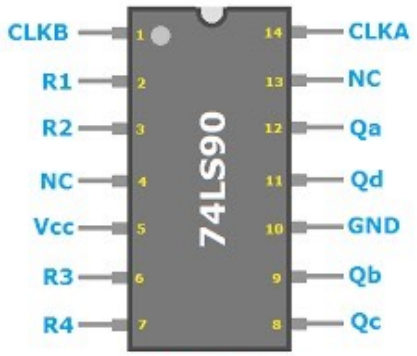
4. Design Development (P4, K5)

System Components:

Component	Function
Arduino UNO	Main controller, decision-making unit
IR Sensors × 4	Detect vehicle presence
Counter IC (7490)	Counts signal duration
BCD-to-7-Segment Decoder (7447)	Converts binary count to 7-segment display
7-Segment Display	Shows countdown timer
LEDs (R, Y, G × 4 roads)	Visual traffic lights
Push Button	Zebra crossing / Emergency trigger
Power Supply (5 V)	System operation

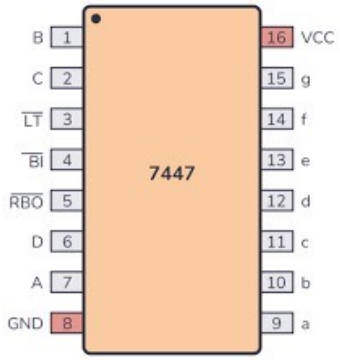
Arduino UNO - Main processor that controls all decisions and coordinates between components.

Counter IC (7490) - Manages the countdown timing for traffic signal duration.



IC 7490 Pin Diagram

BCD-to-7-Segment Decoder (7447) - Converts binary count from counter to 7-segment display format.

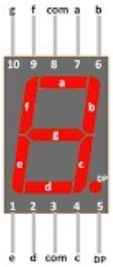


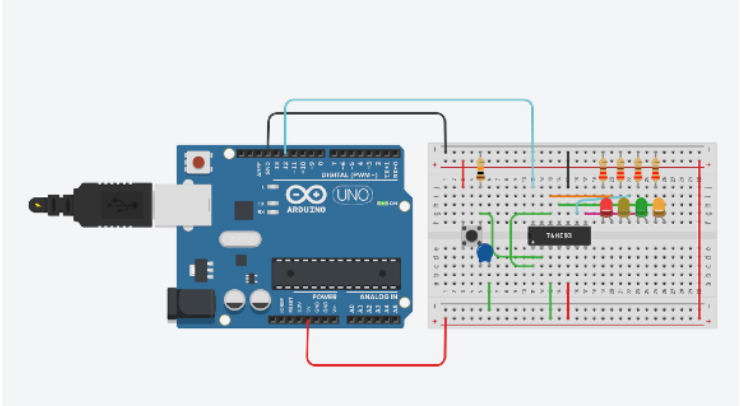
IR Sensors × 4 - Detect vehicle presence on each road and send signals to Arduino.

LEDs (R, Y, G) - Provide visual traffic signals (Red, Yellow, Green) for each road.

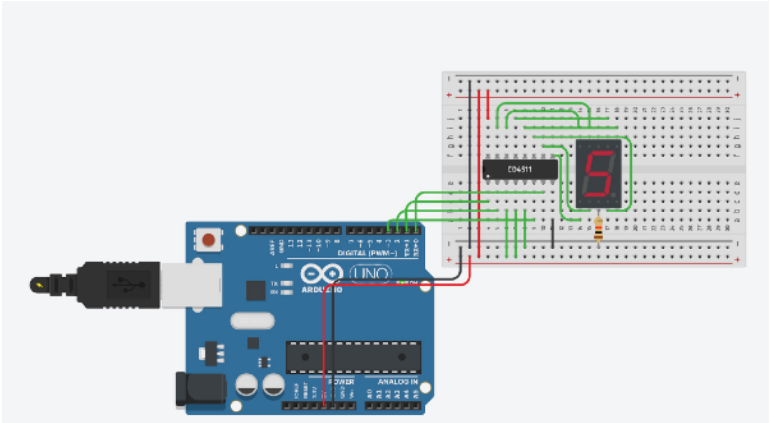
Push Button - Allows pedestrians to request crossing or emergency vehicle priority.

7-Segment Display - Visually shows the remaining time countdown to drivers

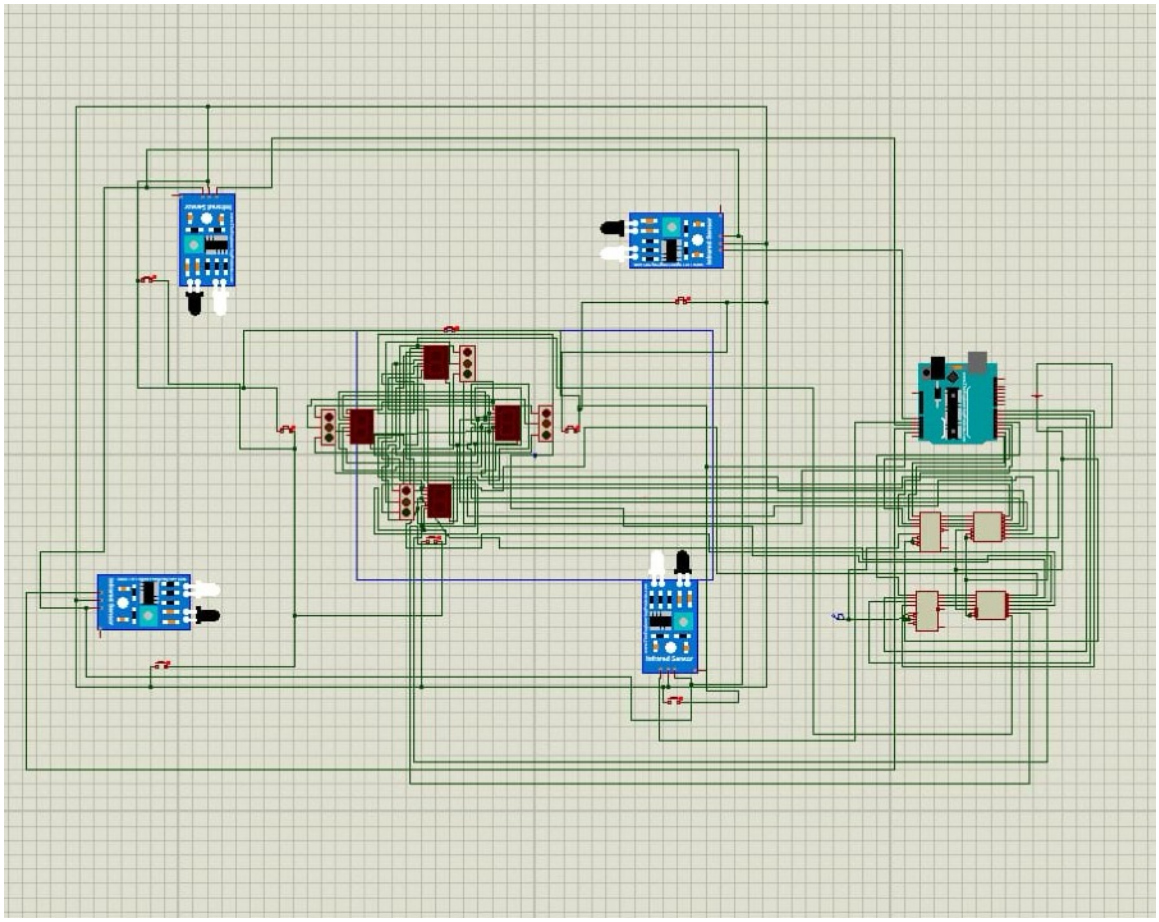




Manages the countdown timing for traffic signal duration.



Converts binary count from counter to 7-segment display format.



Circuit Operation Summary:

1. IR sensors feed traffic data to Arduino.
2. Arduino processes inputs and generates control signals for the counter IC.
3. The counter runs for a defined time (e.g., 15 seconds per road) and decrements every second.
4. Binary output of the counter goes to decoder → 7-segment display shows countdown.
5. When count reaches 0 or fewer than 3 vehicles remain, Arduino shifts to the next active road.
6. Zebra-crossing or emergency input can override any running phase instantly.

Source Code Repository

The complete implementation is available at:

GitHub: https://github.com/XunidEmon/Adaptive-Traffic-Control-System

5. Truth Table

Adaptive Traffic Control System — Arduino + IR + Counter + 7-Segment + Emergency + Zebra

Road A Vehicle Count	Road B Vehicle Count	Road C Vehicle Count	Road D Vehicle Count	Pedestrian/Zebra Switch	Emergency Input	Active Signal (Green)	System Action / Description
0	0	0	0	0	0	None	No vehicle detected → All Red (Idle mode)
<3	<3	<3	<3	1	0	All Red	Pedestrian button pressed → All roads Red, Zebra active
Any	Any	Any	Any	0	1	Emergency Road (e.g., D)	Emergency mode → All other roads Red, emergency lane Green
≥3	<3	<3	<3	0	0	Road A	Road A has enough vehicles (≥3) → Green for A, others Red
<3	≥3	<3	<3	0	0	Road B	Road B active → Green for B, others Red

<3	<3	≥3	<3	0	0	Road C	Road C active → Green for C, others Red
<3	<3	<3	≥3	0	0	Road D	Road D active → Green for D, others Red
≥3	≥3	<3	<3	0	0	Road A → B	A and B both busy → A gets priority, then B
≥3	<3	≥3	<3	0	0	Road A → C	A and C both active → A first, then C
<3	≥3	≥3	<3	0	0	Road B → C	B first, then C (priority logic)
≥3	<3	<3	≥3	0	0	Road A → D	A first, then D
<3	≥3	<3	≥3	0	0	Road B → D	B first, then D
<3	<3	≥3	≥3	0	0	Road C → D	C first, then D
≥3	≥3	≥3	<3	0	0	A → B → C	Three roads busy → Sequential rotation
≥3	≥3	≥3	≥3	0	0	A → B → C → D	All roads busy → Full cyclic sequence

<3	<3	<3	<3	0	0	None	Less than 3 vehicles on all → System waits or uses standby (idle)

6. Conclusion

The Arduino-based Adaptive Traffic Control System successfully detects vehicle density, counts vehicles using IR sensors, and controls signal timing through counter ICs and 7-segment displays. The design reduces congestion, enhances safety, and supports eco-friendly urban mobility. Future improvements could include IoT integration, camera-based AI detection, and wireless emergency communication.

REFERENCES

[1] J. N. Fadila, N. H. A. Wahab, A. Alshammari, A. Aqarni, A. Al-Dhaqm, and N. Aziz, "Comprehensive Review of Smart Urban Traffic Management in the Context of the Fourth Industrial Revolution," *IEEE Access*, vol. 12, pp. 150000-150025, Dec. 2024, doi: 10.1109/ACCESS.2024.3509572.

[2] Arduino Official Documentation, "Arduino Programming Guide,"

[3] Texas Instruments, "SN7490 Decade Counter Datasheet,"

[4] Texas Instruments, "SN7447 BCD-to-7-Segment Decoder Datasheet,"

[5] Autodesk, "Tinkercad Circuits Simulation Platform," 2024.

[6] Labcenter Electronics, "Proteus Design Suite User Manual,"