

Traffic Light Controller

ENGS110 Intro to Programming

8 May 2026

PRD (Product Requirements)

1: Project Overview

This project involves designing and implementing a realistic traffic light control system on a four-way crossing intersection using an Arduino microcontroller, programmed in the Arduino IDE software. The intersection will consist of four traffic lights controlling two main directions from North to South, East to West. Two pedestrian crossing indicators will be implemented as well. All of this seasoned with two independent pedestrian request buttons and an active buzzer for audio feedback.

2: Goals

- Ensure safe and realistic operation.
- Allow safe pedestrian crossings.
- Implement a safety gap between requests.
- Detect and queue pedestrian requests through dedicated buttons.
- Buzzer to provide real-life style audio cues during yellow flashing and pedestrian crossing phases.
- Preventing vehicle collisions

3: Stakeholders

- Drivers
 - Drivers must receive clear traffic signals, safely pass through intersections.
- Pedestrians
 - Pedestrians use the crossing buttons and pedestrian LEDs to safely cross the street.
- Traffic Authorities
 - Reliable signal timing and predictable traffic behavior with reduced accident risks.

PRD (Functional Specifications)

1: System Behavior

The system controls the traffic lights for a 4-way intersection. Only one traffic axis can have a green signal at a time. The system constantly switches between the NS and EW axis with “All Red” states in between.

2: Traffic Flow

Normal Flow

1. NS Green ON
2. NS Yellow flashing
3. All Red
4. EW Green On
5. EW Yellow Flashing
6. All Red
7. Repeat

Pedestrian Flow

Pedestrians press a button to request crossing
The request is stored
The crossing activates during the next safe window
Vehicle traffic remains stopped during pedestrian crossing
Pedestrian LEDs switch from red to green
A buzzer alarms during crossing

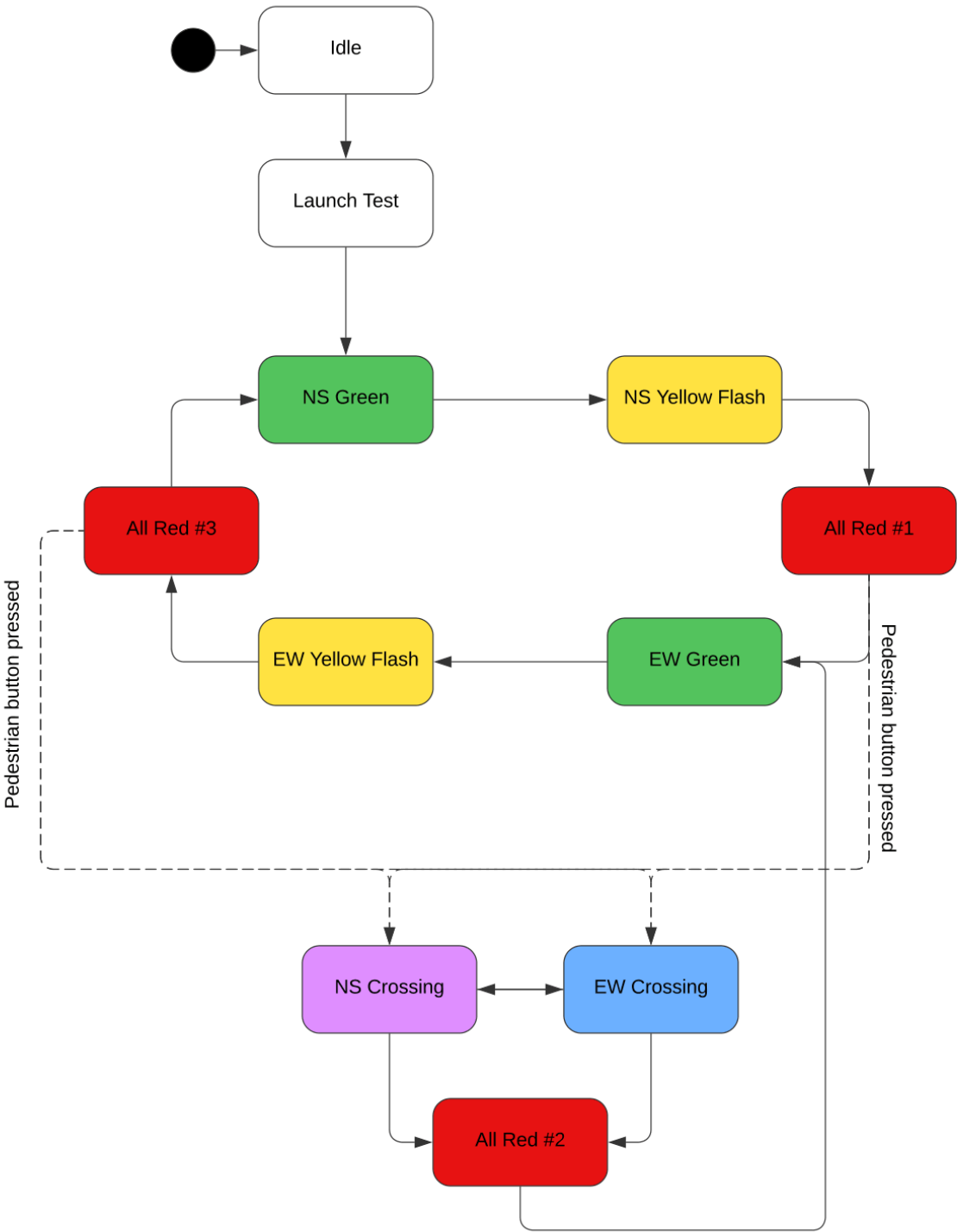
3: Timing

- NS Green – 5 seconds
- NS Yellow Flash – 3 seconds
- All Red – TBD
- Pedestrian Crossing - 5 seconds
- Pedestrian Warning Flash – last 1.5 seconds
- EW Green – 5 seconds
- EW Yellow Flash – 3 seconds
- All Red – TBD

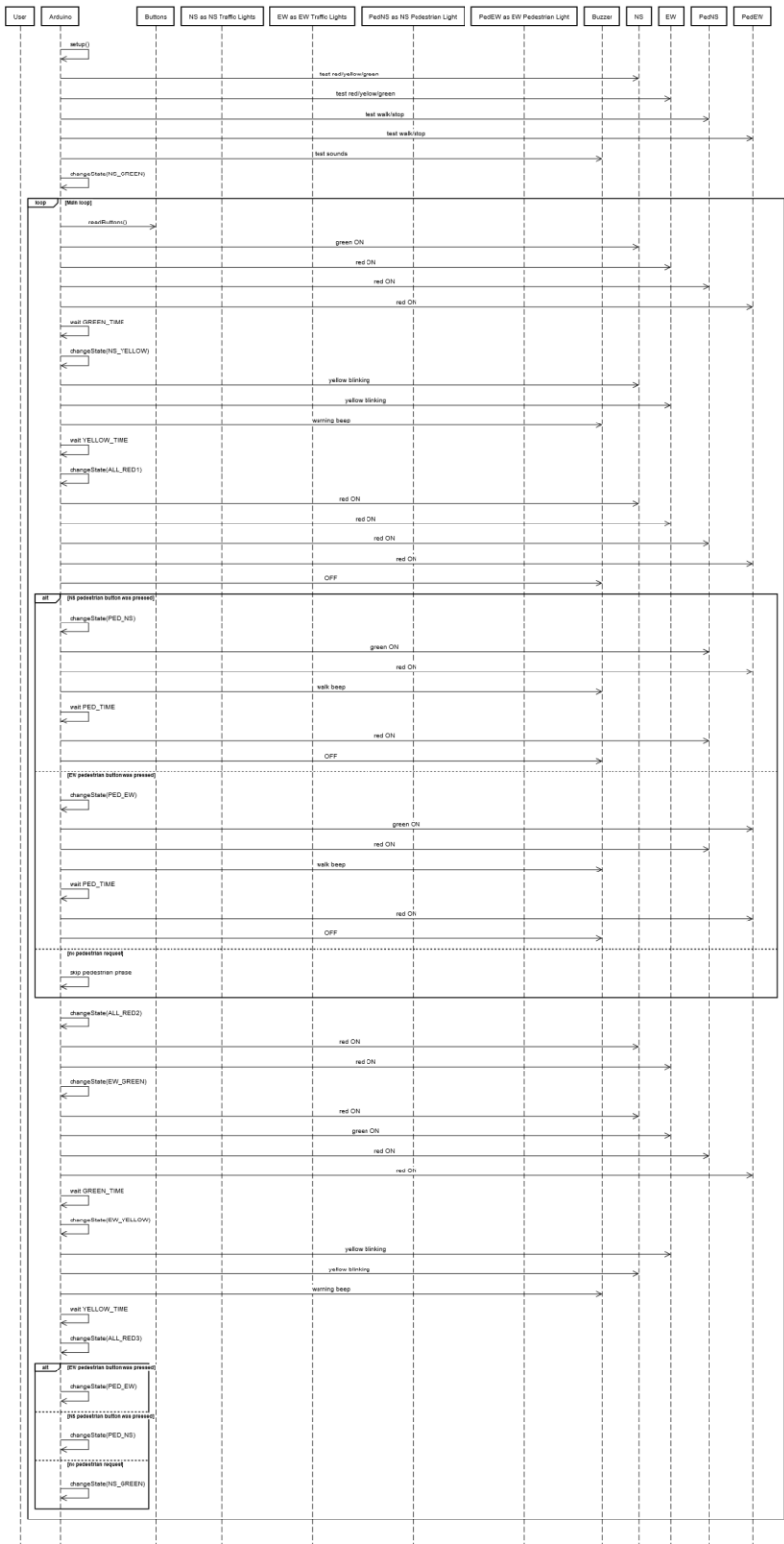
4: Timing Logic

The project uses `millis()` timing instead of `delay()`.

State Machine Diagram



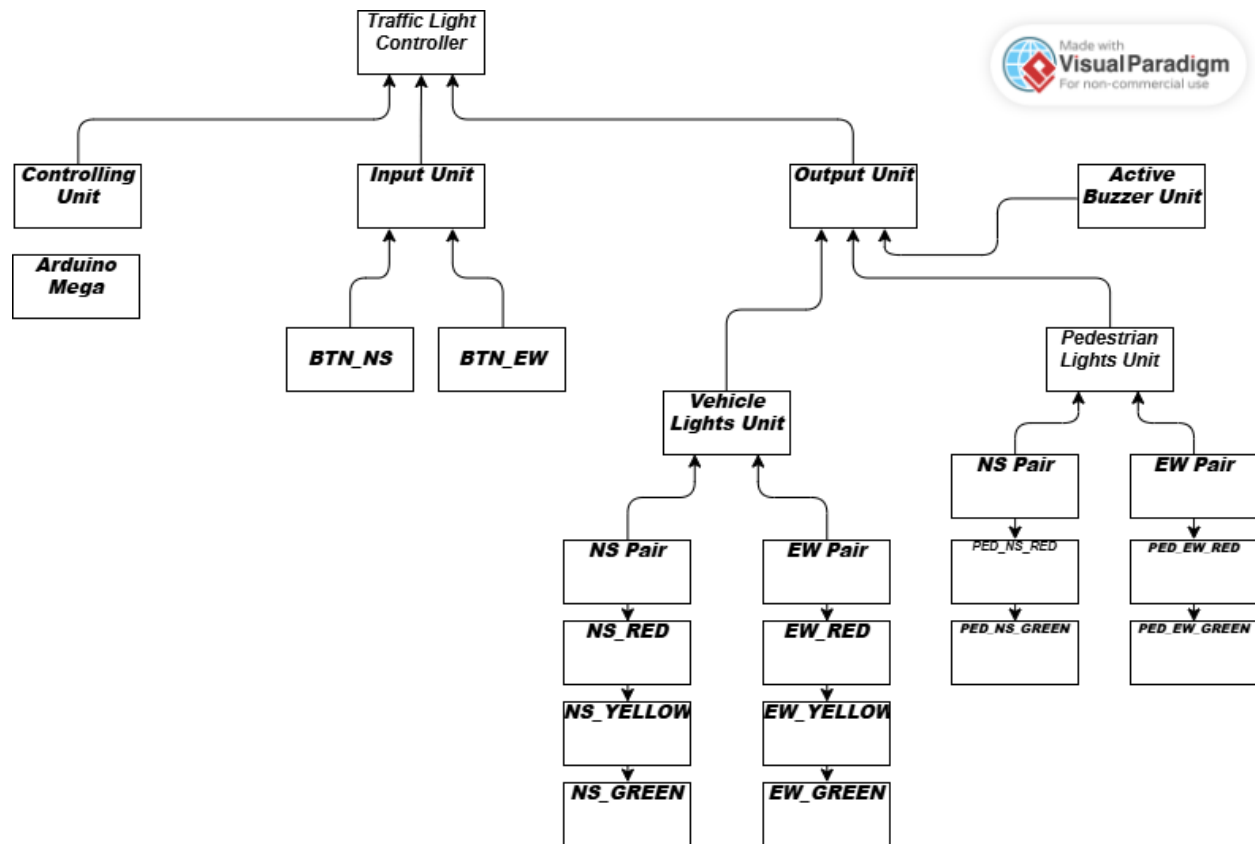
Sequence Diagram



DEVSPEC (Development Specification)

1: Components

- Arduino Mega 2560
- 12 LEDs (4x red, 2x yellow, 4x green)
- 8 smaller LEDs for pedestrian cues (4x red, 4x green)
- 2 push buttons with input pullups
- 1 active buzzer
- 1 NPN transistor
- 1 k Ω resistor for transistor base
- 220 Ω resistors for LEDs
- Breadboard and male-to-male jumper wires



2: Software Used

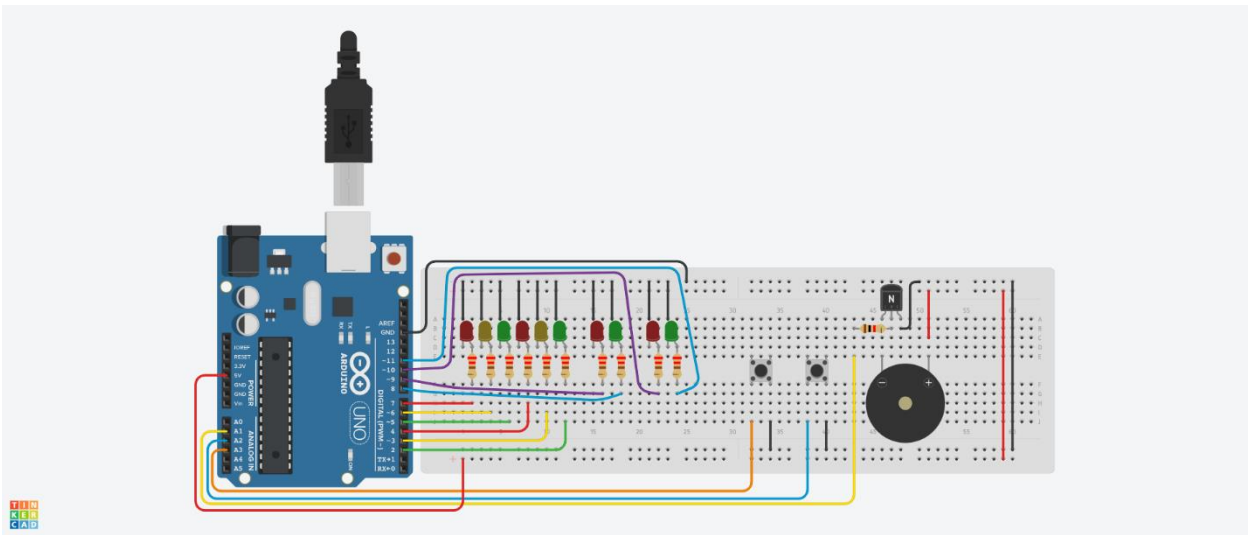
- Arduino IDE
- C/C++ language

3: Program Structure

• Pinout
• Timing Configurations
• Vehicle and Pedestrian light actuators/helpers
• changeState()
• readButtons()
• pedestrianPhase()
• launchtest()
• yellowFlash()
• setup()
• loop()

4: Pinout

Pin	Function
A2	Buzzer
A3	NS Button
A4	EW Button
D2-D4	NS Traffic LEDs
D5-D7	EW Traffic LEDs
D8-D9	NS Pedestrian LEDs
D10-D11	EW Pedestrian LEDs



5: Testing

Test Case	Description	Result	Status
Traffic Sequence Test	Observe full NS/EW traffic cycle	System followed correct sequence	Passed
Green Light Safety Test	Check for simultaneous green signals	No conflicting green states observed	Passed
Pedestrian Button Test	Press pedestrian buttons during traffic	Requests activated correctly	Passed
Pedestrian Crossing Test	Observe pedestrian crossing operation	Crossing sequence worked correctly	Passed
Yellow Warning Test	Observe yellow flashing phase	Warning signals synchronized correctly	Passed
Startup Initialization	Restart the controller multiple times	Startup state initialized correctly	Passed
Timing Accuracy Test	Measure approximate state durations	Timings remained consistent	Passed
Continuous Runtime Test	Run the system for extended time	No crashes or freezes observed	Passed

Circuit Diagram

