

Traffic Control by Using Arduino UNO

Project Report



Supervisor

Khandaker Iftakher Ahmed

Lecturer C.C. of CSE
European University of Bangladesh

Submitted By

[230122003, 230122004, 230122015,
230122031, 230122077]

Students of CSE
European University of Bangladesh

[January 2025]

Department of Computer Science & Engineering

European University of Bangladesh.

Traffic Control by Using Arduino UNO

Faculty of Computer Science & Engineering, European University of Bangladesh.

Abstract— Our main goal is to develop an ultramodern and reliable traffic control by using arduino uno. Which is faced in almost every field in our daily life. For example road crossing, human, car, bus, motor-cycle, three-wheelers, etc. To develop this system, we used an Arduino UNO as the main processor, LEDs, servo motors, buzzers, resistors, jumper wires, a 9V battery, and other components.

Introduction

The main aim of this project is to reduce the wastage of people's precious time, and to ensure the safety of all while crossing the road, as well as to develop a smooth management. Working to reduce the wastage of valuable time in daily life and avoid various unwanted road accidents. So in this project we can create an orderly traffic control system by using Arduino UNO. Which will play an important role in the welfare of our country and nation.

Problem Description

Arduino UNO must be used properly in making this project. And how to complete the project using Arduino UNO must be noted. In traffic control management, state-of-the-art creativity, intelligence and future possibilities must be exploited. Also, to solve the traffic crisis in the near future, modern technology must be used. And new techniques and technologies have to be developed to avoid unwanted road accidents. At the same time, modern technology should be used in such a way that everyone obeys the traffic laws.

Methodology

Equipments:

- Arduino UNO
- BreadBoard
- Buzzer
- Servo Motor
- IC7400 (NAND)
- LDR (Light Dependent Sensor)
- LEDs (Red, Yellow, Green)
- Resistors
- Jumper Wires
- 9V Battery

*To enhance the project's quality, we can modify the materials as needed. The selection of materials will depend on the tasks that can be completed within the project's limited timeframe.

Use Case Diagram: A use case diagram, a type of Unified Modeling Language (UML) diagram, visually represents the interactions between users (actors) and a system, outlining the different ways a user can interact with and achieve goals within the system.

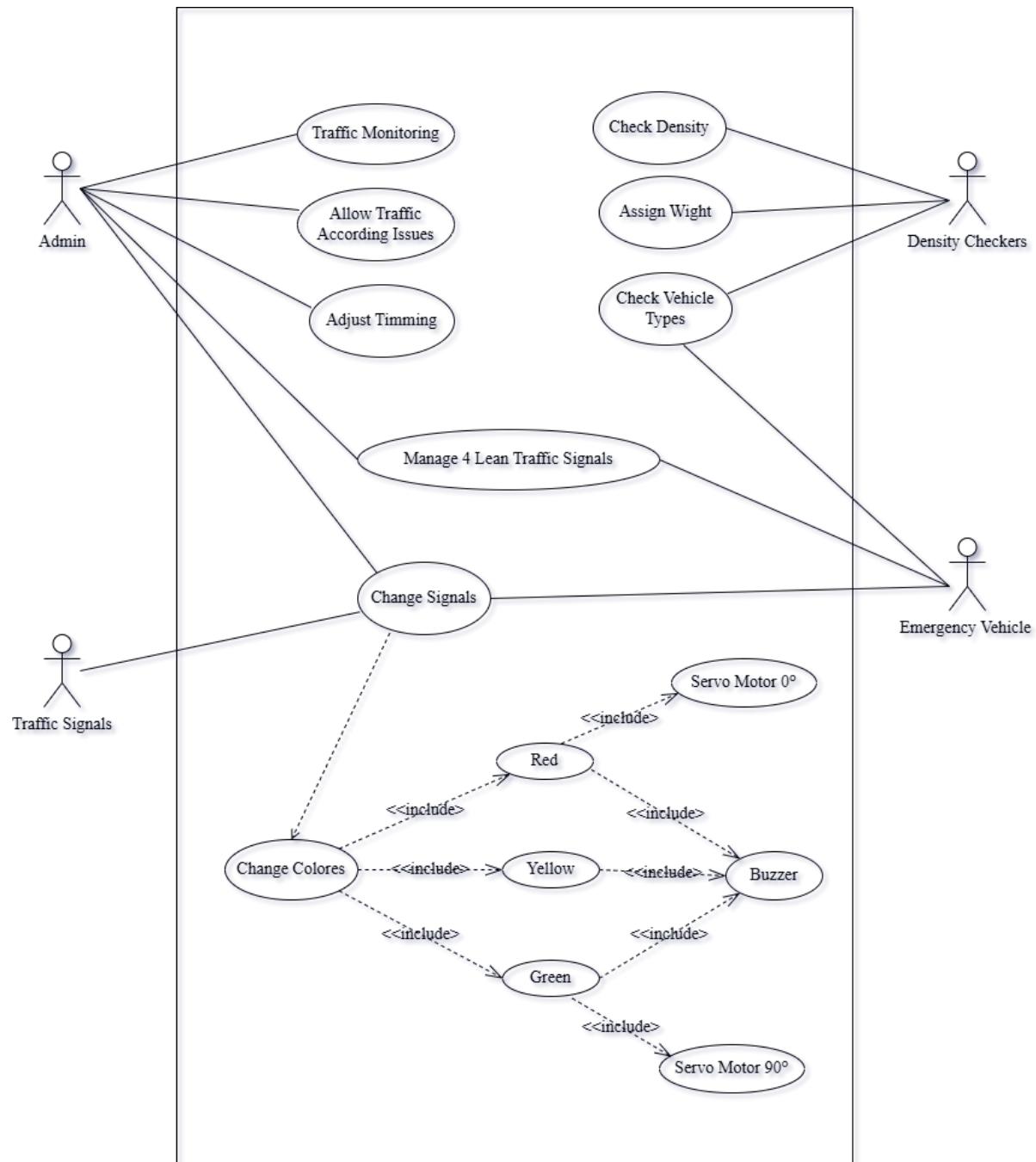


Fig. Traffic Control System

Circuit Diagram: To develop this system, we used an Arduino UNO as the main processor, LEDs, servo motors, buzzers, resistors, jumper wires, a 9V battery, and other components.

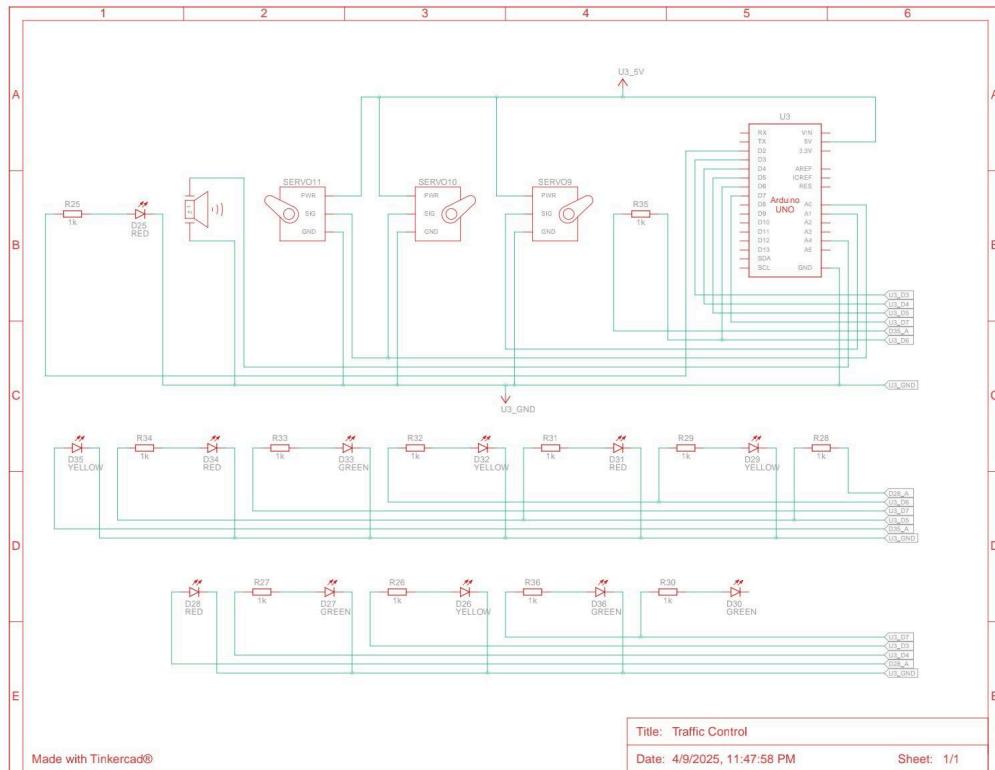


Fig. Circuit Diagram

Connection Diagram: A connection diagram, also known as a wiring diagram, is a visual representation that shows how electrical components are connected, including their relative positions and the connections between them, often using standardized symbols and lines.

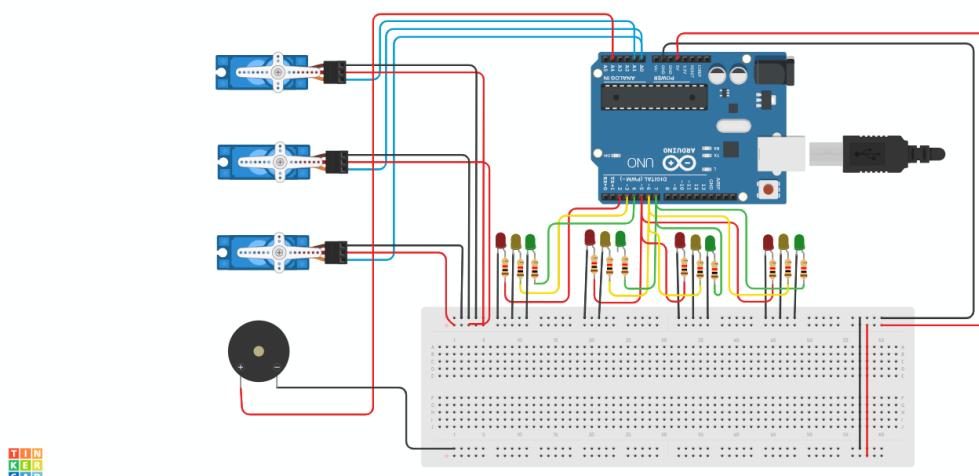


Fig. Connection Diagram

Arduino UNO Code:

```
#include <Servo.h>

Servo servo1, servo2;

int servoPins[2] = {A0, A1}; // Lane 1 and Lane 2 Servo pins
int ledPins[2][3] = {
  {2, 3, 4}, // Lane 1: Red, Yellow, Green
  {5, 6, 7} // Lane 2: Red, Yellow, Green
};

const int buzzer = A4;

void setup() {
  for (int i = 0; i < 2; i++) {
    for (int j = 0; j < 3; j++) {
      pinMode(ledPins[i][j], OUTPUT);
    }
  }

  servo1.attach(servoPins[0]);
  servo2.attach(servoPins[1]);
  servo1.write(0); // Closed
  servo2.write(0); // Closed

  pinMode(buzzer, OUTPUT);
}

void loop() {
  laneGo(0, servo1); // Lane 1 Go
  laneGo(1, servo2); // Lane 2 Go
}

void laneGo(int lane, Servo &servo) {
  int otherLane = lane == 0 ? 1 : 0;

  // Red for other lane
  setLights(otherLane, 0);
  moveServo(otherLane, 0);

  // Green for current lane
  setLights(lane, 2);
}
```

```

moveServo(lane, 90);
buzzerSignal(2000, 3);
delay(5000);

// Yellow warning
setLights(lane, 1);
buzzerSignal(1500, 1);
delay(2000);

// Red and close barrier
setLights(lane, 0);
moveServo(lane, 0);
buzzerSignal(1000, 1);
delay(1000);
}

void setLights(int lane, int signal) {
    for (int i = 0; i < 3; i++) {
        digitalWrite(ledPins[lane][i], LOW);
    }
    digitalWrite(ledPins[lane][signal], HIGH);
}

void moveServo(int lane, int angle) {
    if (lane == 0) servo1.write(angle);
    else servo2.write(angle);
    delay(500);
}

void buzzerSignal(int frequency, int count) {
    for (int i = 0; i < count; i++) {
        tone(buzzer, frequency);
        delay(300);
        noTone(buzzer);
        delay(300);
    }
}

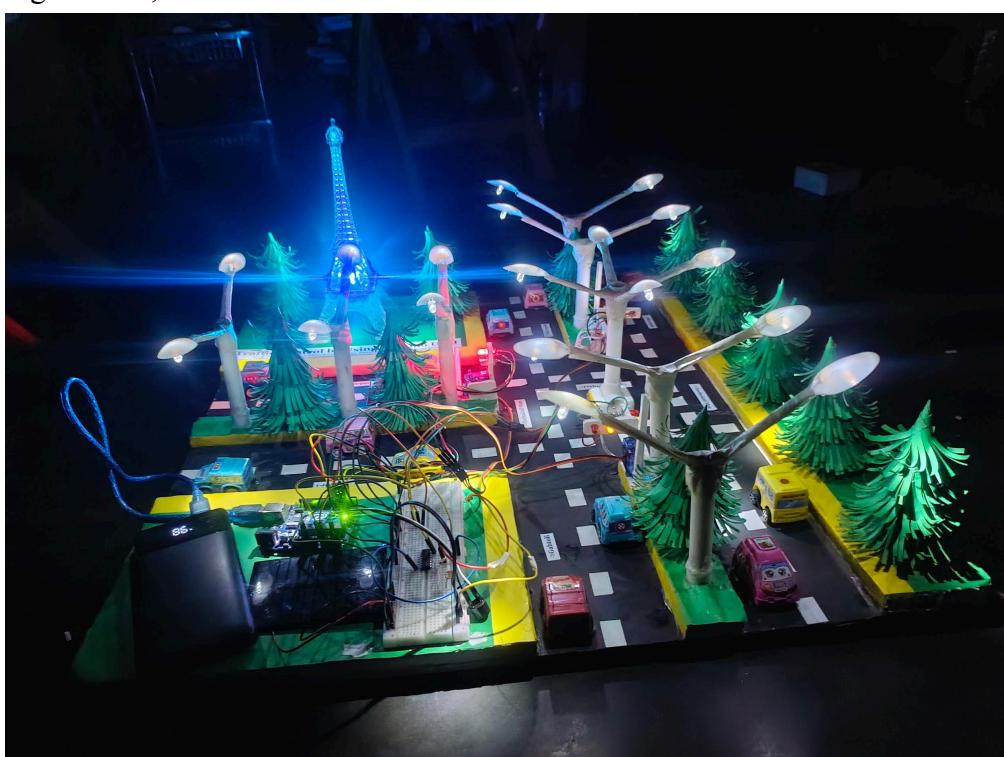
```

Project Controlling: Project controlling, a crucial element of project management, involves monitoring and managing a project's progress to ensure it stays on track, achieves goals, and stays within budget and timeline, by comparing actual performance against planned objectives and taking corrective actions when necessary.

*When light mode,



*When night mode,



Project Scope

1. **Advantages**
 - a. **Cost-Effective:** Affordable microcontroller.
 - b. **Educational Value:** Hands-on learning experience.
 - c. **Customization:** Easily modifiable and expandable.
 - d. **Real-World Application:** Practical solution for traffic management.
2. **Limitations**
 - a. **Limited Processing Power:** May restrict complex algorithms.
 - b. **Reliability:** Affected by power supply, environment, and component quality.
 - c. **Sensor Integration:** Requires additional components and complex programming.
 - d. **Limited Connectivity:** Challenging to integrate with other systems.
3. **Enhancements**
 - a. **Energy Efficiency:** Use energy-efficient components and solar power.
 - b. **Vehicle Detection:** Use sensors to detect vehicles and adjust light timings.
 - c. **Remote Monitoring:** Integrate with IoT for remote monitoring and control.
 - d. **Adaptive Traffic Control:** Implement algorithms to adapt to real-time traffic conditions.

Feasibility Study

1. **Technical Feasibility**
 - a. **Hardware:** Limited processing power and memory.
 - b. **Real-Time:** Challenges with multiple inputs/outputs.
 - c. **Sensors:** Requires extra components and complex programming.
 - d. **Scalability:** Managing intersections needs advanced hardware.
2. **Economic Feasibility**
 - a. **Low Cost:** Affordable components and development.
 - b. **Energy Efficiency:** Low power consumption reduces operational costs.
 - c. **Open-Source:** Access to free resources and community support.
 - d. **Scalability:** Cost-effective for small-scale implementations.
3. **Operational Feasibility**
 - a. **Ease of Implementation:** Simple setup and programming.
 - b. **Maintenance:** Low maintenance requirements.
 - c. **Reliability:** Dependent on component quality and environmental conditions.
 - d. **Scalability:** Suitable for small-scale implementations.

Solution Application Areas

1. Urban Intersections

- a. **Traffic Flow Management:** Efficiently manage traffic at busy intersections.
- b. **Pedestrian Safety:** Ensure safe crossing for pedestrians.
- c. **Emergency Vehicle Priority:** Allow emergency vehicles to pass quickly.
- d. **Adaptive Control:** Adjust light timings based on real-time traffic conditions.

2. Pedestrian Crossings

- a. **Safety:** Ensure safe crossing for pedestrians.
- b. **Button Integration:** Use a button to trigger pedestrian crossing signals.
- c. **Timed Signals:** Implement timed signals for pedestrian crossing.
- d. **Visual and Audible Alerts:** Provide visual and audible alerts for pedestrians.

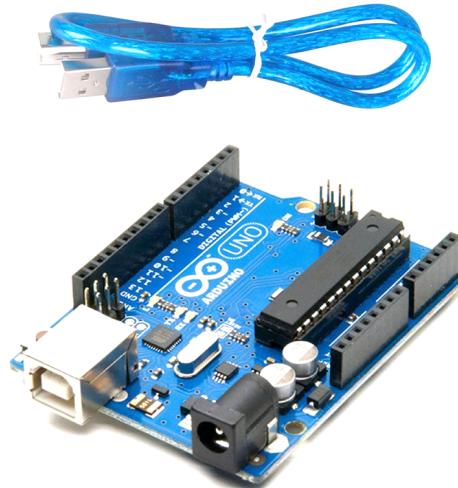
3. Emergency Routes

- a. **Priority Control:** Allow emergency vehicles to pass quickly.
- b. **Real-Time Detection:** Use sensors to detect emergency vehicles.
- c. **Override Function:** Implement an override function to change traffic signals.
- d. **Communication:** Integrate with emergency services for real-time updates.

Tools/Technology

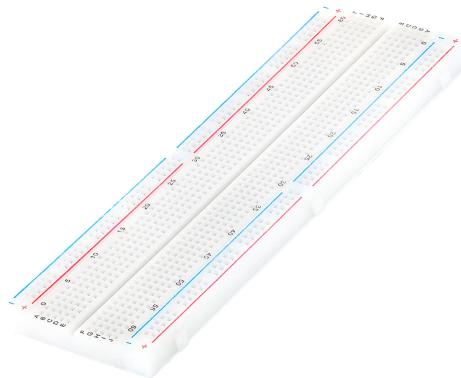
Hardware:

- **Arduino UNO**



The Arduino Uno is a popular, open-source microcontroller board based on the ATmega328P, designed for creating interactive and electronic projects, and is known for its ease of use and programming, making it a great tool for both beginners and experienced users.

- **BreadBoard**



The breadboard is the basic component of any circuit building process. All components, be it input sensors or output display devices are connected to the power supply, microcontroller using wired connections through a breadboard.

- **Buzzer**



A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short).

- **Servo Motor**



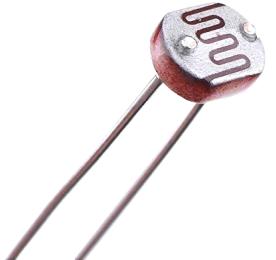
A servomotor is a closed-loop servomechanism that uses position feedback (either linear or rotational position) to control its motion and final position.

- **IC7400 (NAND)**



The IC 7400 is a 14-pin integrated circuit (IC) containing four independent two-input NAND gates, a fundamental building block in digital logic circuits.

- **LDR (Light Dependent Sensor)**



An LDR, or Light Dependent Resistor, is a passive electronic component that changes its resistance based on the amount of light it is exposed to, acting as a light sensor.

- **LEDs (Red, Yellow, Green)**



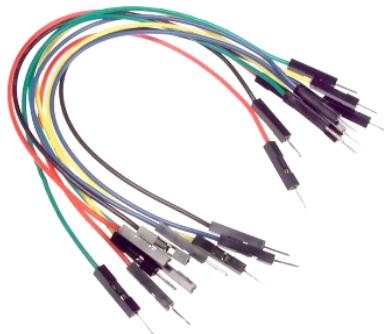
Red, yellow, and green LEDs are a type of light-emitting diode (LED) that emits light of those specific colors when current flows through them, commonly used in signals and indicators.

- **Resistors**



Resistors are passive devices that restrict the flow of current or divide the voltage through the circuit. The input power passes through these resistors and then to the sensors to avoid damage.

- **Jumper Wires**



These are the main components that are used to establish the connections between different devices of the circuit.

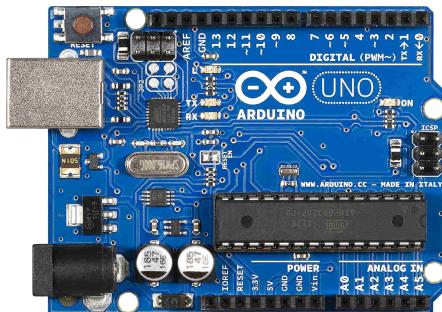
- **9V Battery**



A 9V battery is a compact, rectangular-shaped battery that provides a nominal voltage of 9 volts, commonly used in devices like smoke detectors, clocks, and remote controls, etc.

Software:

- Not applicable (if built as a hardware-based project).



The Arduino Uno is a series of open-source microcontroller boards based on a diverse range of microcontrollers. It was initially developed and released by Arduino company in 2010.

- For microcontroller-based systems, software like Arduino UNO IDE may be used.



The Arduino IDE (Integrated Development Environment) is a software application used to write, compile, and upload code to Arduino boards, including the Arduino Uno, allowing users to program and control the microcontroller on the board.

- TinkerCad circuit simulation software.



Tinkercad is a free, web-based 3D design, electronics, and coding platform developed by Autodesk, known for its ease of use and accessibility, making it ideal for beginners and educators to explore 3D modeling, circuits, and coding concepts.

Expertise of the Team Members

We present the significance of our project with the overall cooperation of the team members:

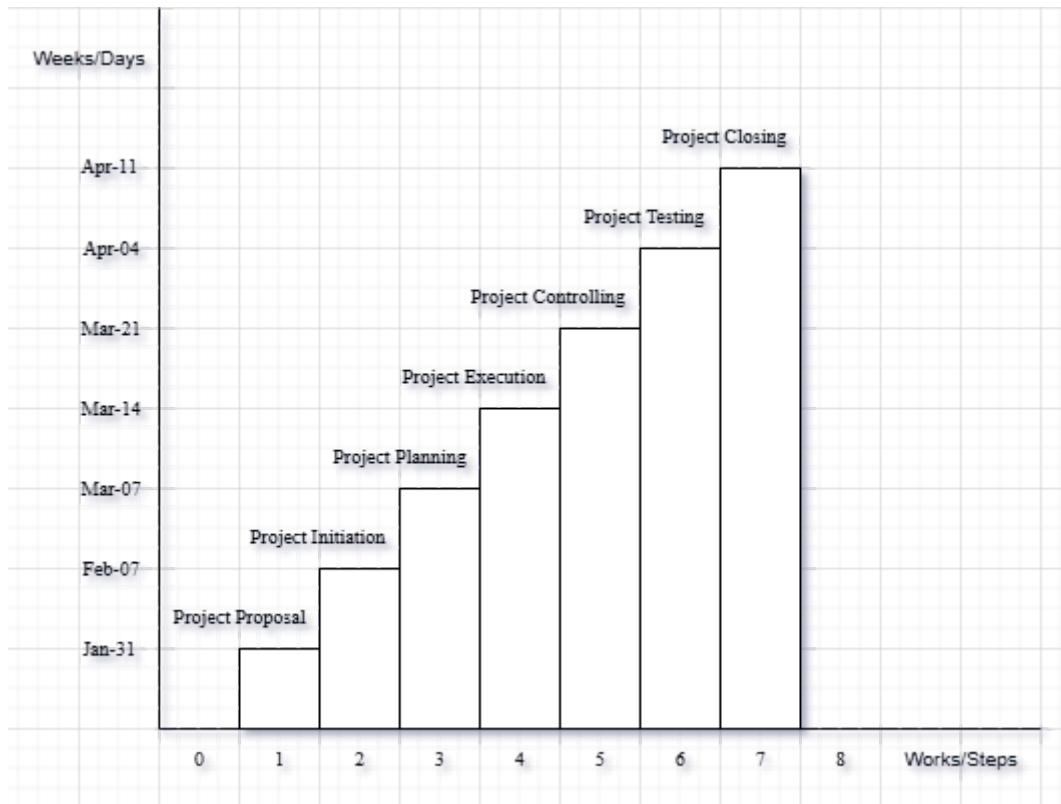
1. **Cost-Effective:** Arduino UNO and its components are affordable, making it a budget-friendly option for developing traffic control systems.
2. **Educational Value:** Working with Arduino UNO provides a hands-on learning experience, making it an excellent educational tool for students and hobbyists.
3. **Customization:** Arduino UNO is easily modifiable and expandable, allowing you to tailor the system to specific traffic control needs.
4. **Scalability:** The system can be scaled up to manage larger and more complex traffic networks.
5. **Open-Source Community:** The extensive open-source community offers a wealth of shared resources, support, and project ideas.
6. **Real-World Application:** Arduino UNO provides a practical solution for traffic management, enabling real-time monitoring and control of traffic signals.

Milestones

→ Cost:

SN	Hardware	Cost
1	Arduino UNO	580
2	BreadBoard	70
3	Buzzer	20
4	Servo Motor	120
5	IC7400 (NAND)	20
6	LDR (Light Dependent Sensor)	20
7	LEDs (Red, Yellow, Green)	20
8	Resistors	20
9	Jumper Wires	70
10	9V Battery	80

→ **Flowchart:**



Conclusions

The "Traffic Control by Using Arduino UNO" project effectively demonstrates an automated traffic management system using Arduino UNO as the main processor. By integrating LEDs, servo motors, buzzers, resistors, jumper wires, and a 9V battery, the system efficiently controls traffic flow, reducing congestion and enhancing road safety. Its scalability allows for future upgrades, such as real-time data processing and AI-based optimization, making it a viable solution for smart city infrastructure.

References

We mentioned all the literature or web references, etc here-

- Internet web page & resources: [ResearchGate](#), Gemini, etc.

End!