**BAHRIA UNIVERSITY, Karachi Campus**

*Department of Software Engineering*

**REPORT**

**Course Title:**  Embedded System Design Lab **Course Code**: CEL 439

**Course Instructor:**  Noman Ahmed Siddiqui **Class**: BSE- 5C

**Lab Instructor:** Noman Ahmed Siddiqui

PROJECT TITLE:

**Traffic Control System Using Arduino UNO**

GROUP MEMBERS LIST:

|  |  |  |  |
| --- | --- | --- | --- |
| S.NO | Enrollment | Name | Email |
|  | 02-131222-090 | Ayesha | ayeshamishree@gmail.com |
|  | 02-131222-097 | Arifa Naseem | arifa42986@gmail.com |
|  | 02-131222-088 | Hafsa Shahid | hafsashahid2704@gmail.com |

**Submission Date:**

**Signed: Remarks: Score:**

**Table Of Content:**

1. [Introduction: 1](#_Toc186367685)
2. [Objective: 1](#_Toc186367686)
3. [Working Principle: 1](#_Toc186367687)

[Hardware Implementation: 1](#_Toc186367688)

[Software Simulation: 1](#_Toc186367689)

1. [Code: 2](#_Toc186367690)
2. [Circuit Diagram(Hardware): 3](#_Toc186367691)
3. [Circuit Diagram(Software): 4](#_Toc186367692)
4. [Block Diagram: 4](#_Toc186367693)
5. [Components: 4](#_Toc186367694)

[Hardware Components 4](#_Toc186367695)

[Software Tools 5](#_Toc186367696)

1. [Conclusion: 5](#_Toc186367697)

# Introduction:

This project focuses on the development of a traffic light control system using an Arduino UNO microcontroller. The hardware implementation simulates a two-way traffic signal using LEDs, while the software simulation expands to a four-way traffic control system incorporating a seven-segment display for enhanced functionality. This approach demonstrates the versatility of embedded systems in traffic management solutions.

# Objective:

* To design and implement a two-way traffic control system using hardware components.
* To simulate a four-way traffic control system in software using seven-segment displays.
* To understand the integration of hardware and software in embedded systems.

# Working Principle:

## Hardware Implementation:

The hardware implementation focuses on simulating a two-way traffic signal using the Arduino UNO and LEDs. The following steps outline its working:

* **LED Control**: The Arduino UNO controls three LEDs (red, yellow, green) to represent traffic signals.
* **Signal States**: Each LED corresponds to a specific traffic state:
  + **Red**: Indicates stop.
  + **Yellow**: Serves as a warning or transition signal.
  + **Green**: Signals vehicles to go.
* **Timing Mechanism**: Delays are programmed for each signal state to mimic real-world traffic timings, ensuring smooth transitions between signals.
* **Wiring and Components**: The system uses jumper wires, resistors, and a breadboard to establish connections between the Arduino and LEDs.

## Software Simulation:

In the software simulation, the Arduino UNO manages a four-way traffic control system with a seven-segment display. The working principle includes:

* **Countdown Display**: The seven-segment display shows a countdown timer for each direction, providing a clear indication of the remaining signal time.
* **Signal Synchronization**: LEDs and the seven-segment display work in unison to simulate traffic flow in all four direction
* **Realistic Traffic Control**:
  + The countdown timer enhances the realism of the simulation.
  + Transition signals (yellow) and stopping signals (red) ensure logical traffic management.
* **Code Logic**: Arduino code defines the timing, sequences, and synchronization between LEDs and the display to replicate a functional traffic signal system.

# Code:

//Trafic Ligth With Arduino UNO

int red = A0;

int yellow = A1;

int green = A2;

int yellowTime = 1; //Means 20 sec, 0 means 10 sec

int TransfireTime = 5; //Means 60 sec

void setup()

{

for (int i = 0; i <= 13; i++) {

pinMode(i, OUTPUT);

//Set all pins from 0 to 13 as OUTPUT

}

pinMode(red, OUTPUT);

pinMode(yellow, OUTPUT);

pinMode(green, OUTPUT);

digitalWrite(red, LOW);

digitalWrite(yellow, LOW);

digitalWrite(green, LOW);

}

//The line below is the array containing all the binary numbers for the digits on a SSD from 0 to 9

const int number[11] = {0b0111111, 0b0000110, 0b1011011,

0b1001111, 0b1100110, 0b1101101,

0b1111101, 0b0000111, 0b1111111,

0b1101111};

void loop()

{

Red();

Green();

}

void display(const int from)

{

for (int tens = from; tens >= 0; tens--)

{

display\_tens(tens);

}

}

void display\_tens(const int tens)

{

int pin1, a, ones;

//pin1 is just used to deal with pins of the 1st SSD which desplays the tens digit

for (pin1 = 0, a = 0; pin1 < 7; pin1++, a++)

{

digitalWrite(pin1, bitRead(number[tens], a));

}

for (ones = 10; ones >= 0; ones--)

{

display\_ones(ones);

delay(1000);

//I have given a delay of 1 seconds. You can put your own Time!!

}

}

void display\_ones(const int x)

{

int pin2, b;

//pin2 is just used to deal with pins of the 2nd SSD which desplays the ones digit

for (pin2 = 7, b = 0; pin2 <= 13; pin2++, b++)

{

digitalWrite(pin2, bitRead(number[x], b));

}

}

void Red()

{

digitalWrite(red, HIGH);

digitalWrite(yellow, LOW);

digitalWrite(green, LOW);

display(TransfireTime);

digitalWrite(red, HIGH);

digitalWrite(yellow, HIGH);

digitalWrite(green, LOW);

display(yellowTime);

digitalWrite(red, LOW);

digitalWrite(yellow, LOW);

digitalWrite(green, LOW);

}

void Green()

{

digitalWrite(red, LOW);

digitalWrite(yellow, LOW);

digitalWrite(green, HIGH);

display(TransfireTime);

digitalWrite(red, LOW);

digitalWrite(yellow, HIGH);

digitalWrite(green, HIGH);

display(yellowTime);

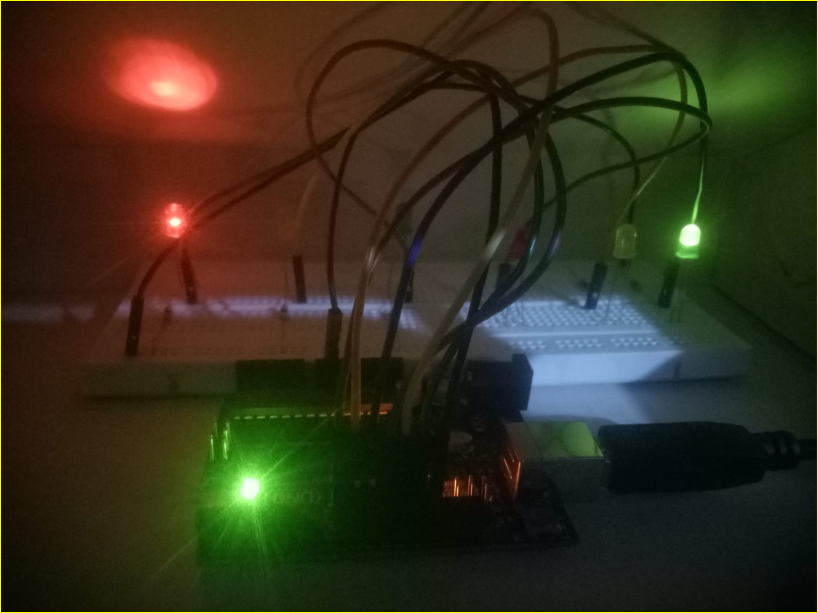
digitalWrite(red, LOW);

digitalWrite(yellow, LOW);

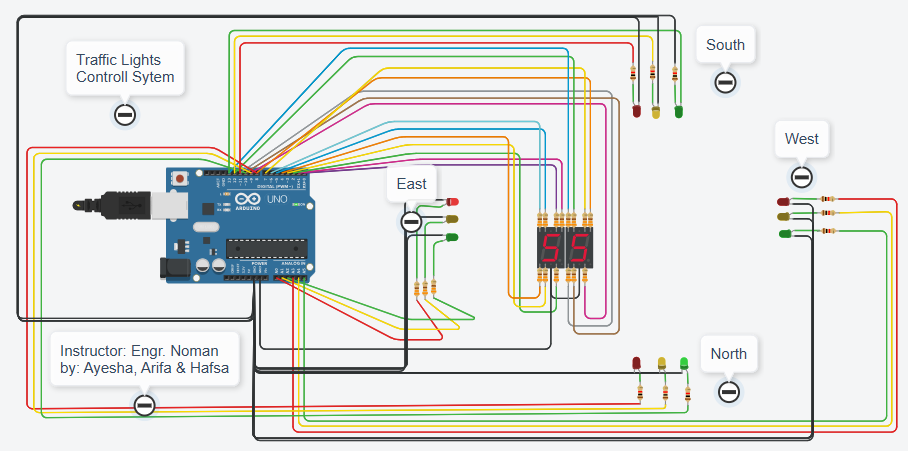
digitalWrite(green, LOW);

}

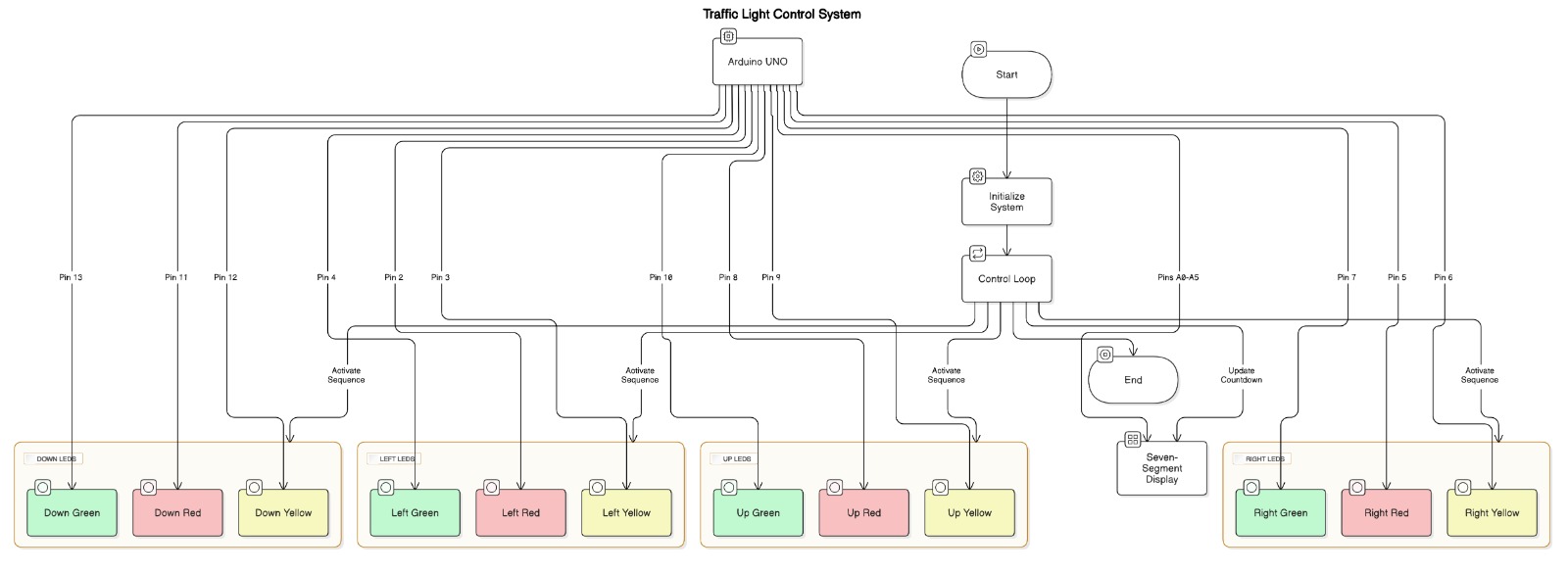
# Circuit Diagram(Hardware):



# Circuit Diagram(Software):



# Block Diagram:



# Components:

## Hardware Components

* Arduino UNO
* LEDs (Red, Yellow, Green)
* Resistors
* Jumper wires
* Breadboard

## Software Tools

* Arduino IDE

# Conclusion:

This project demonstrates the application of embedded systems in traffic management through both hardware and software implementations. The two-way traffic signal system provides a simple yet effective prototype, while the four-way software simulation showcases advanced functionalities. Together, they serve as a foundation for further exploration and development in traffic control systems.