

Established – 1961

Subject: iot

SEVA SADAN'S
R. K. TALREJA COLLEGE
OF
ARTS, SCIENCE & COMMERCE
ULHASNAGAR – 421 003



CERTIFICATE

This is to certify that Mr./Ms. Yashwant gowda
of S.Y. Computer Science (SYCS) Roll No. 2524013
has satisfactorily completed
The Internet Of Thing Mini Project entitled
traffic density monitoring system

during the academic year 2025 – 2026, as a part of the practical
requirement. The project work is found to be satisfactory and is
approved for submission.

PROF. INCHARGE

HEAD OF DEPT

INDEX

Sr. No.	Chapters	Page No.
1	Introduction	
2	Requirement Specification	
3	System Design	
4	Implementation	
5	System Testing and Result	
6	Future Scope and Conclusion	
7	References	
8	Glossary	

INTRODUCTION

Introduction:

In modern cities, traffic congestion has become a major problem due to increasing numbers of vehicles.

Traditional traffic systems operate on fixed timers, which do not consider real-time traffic density. This leads to unnecessary waiting time, fuel wastage, air pollution, and traffic jams. Therefore, there is a need for an intelligent system that can monitor traffic density and control signals accordingly.

Purpose of the Project

The purpose of the Traffic Density Monitoring System is to monitor vehicle density on roads using sensors and automatically control traffic signals based on real-time traffic conditions. The system reduces congestion, saves time, and improves road efficiency using IoT and embedded systems technology.

Importance and Real-Life Applications

This system plays an important role in smart city development. It helps:

- Reduce traffic congestion
- Minimize fuel consumption
- Lower air pollution
- Improve emergency vehicle movement
- Enhance overall traffic management efficiency

Motivation

Traditional traffic lights operate on fixed time intervals regardless of traffic conditions. This causes unnecessary delays on empty roads while crowded roads remain

congested.

With advancements in automation, IoT, and sensor technology, traffic density can now be detected using IR sensors or ultrasonic sensors. By implementing an automated system, traffic signals can dynamically adjust timing based on vehicle density.

Problem Definition

Manual and fixed-time traffic signal systems fail to manage traffic efficiently during peak hours. They:

- Do not monitor real-time vehicle density
- Cause unnecessary waiting time
- Increase fuel consumption
- Increase pollution levels
- Delay emergency services

Therefore, there is a need for an automated Traffic Density Monitoring System that can detect vehicle density and adjust signal timing accordingly.

How It Works

The Traffic Density Monitoring System works using IR sensors placed on each lane.

1. Sensors detect the number of vehicles.
2. The microcontroller (Arduino/NodeMCU) reads sensor data.
3. The system calculates traffic density.
4. Traffic signal timing is adjusted:

- High density → Longer green signal
- Low density → Shorter green signal

5. LEDs simulate traffic lights (Red, Yellow, Green).

Key Features

- Automated Traffic Signal Control
 - Real-Time Traffic Monitoring
 - Reduced Waiting Time
 - Fuel Saving
 - Pollution Reduction
 - Emergency Mode (Future Enhancement)
 - Smart City Compatible
 - Cost Effective
-

Scope of the Project

The system can be implemented in:

- City intersections
- Highways
- Smart cities
- Parking management systems

Future enhancements:

- Cloud-based monitoring
 - AI-based traffic prediction
 - Emergency vehicle priority
 - CCTV integration
 - Mobile application interface
-

Objectives

- To design an automated traffic control system.
- To detect vehicle density using IR sensors.
- To control traffic signals dynamically.
- To reduce traffic congestion.
- To improve road safety.
- To design a scalable smart traffic solution.

REQUIREMENT SPECIFICATION

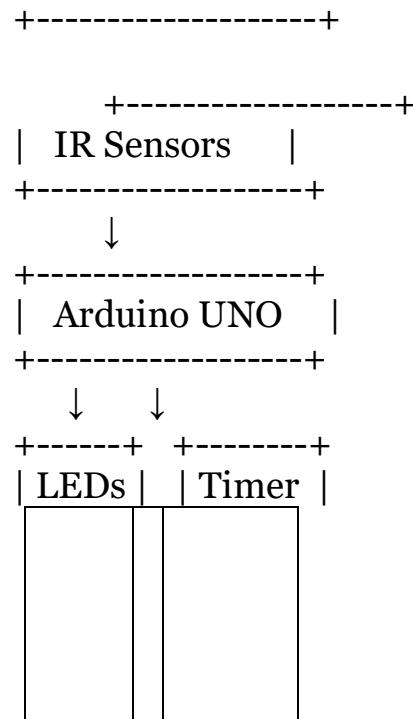
REQUIREMENT SPECIFICATION

Hardware Components

Component	Specification
Microcontroller	ESP32
IR Sensors	Vehicle Detection
LEDs	Traffic Signals
Resistors	Current Control
Breadboard	Circuit Connection
Jumper Wires	Wiring
Power Supply	5V

SYSTEM DESIGN

Block Diagram:



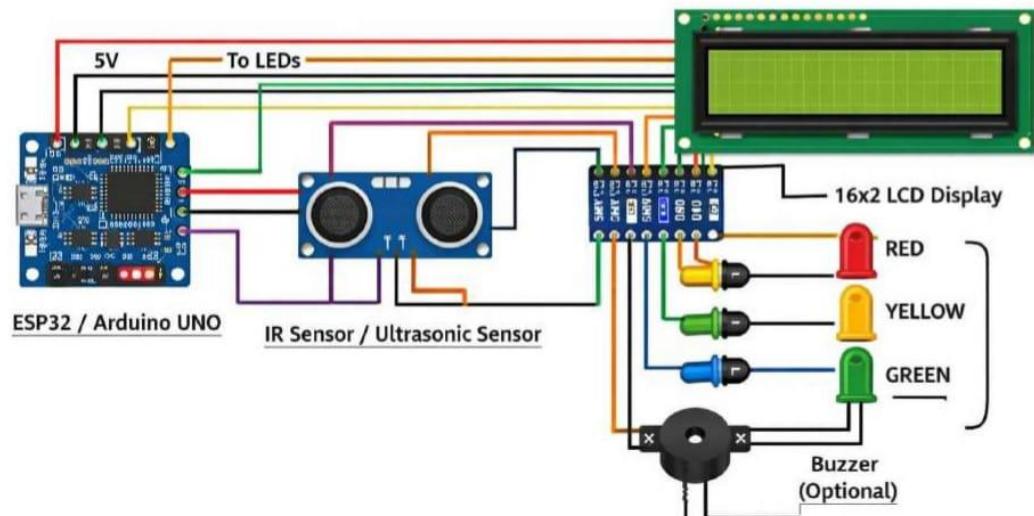
Working Process

1. System powers ON.
2. Sensors detect vehicles.
3. Arduino reads sensor values.
4. Density is calculated.
5. Signal timing adjusted automatically.
6. Loop continues.

Circuit Diagram:

Traffic Density Monitoring System

CIRCUIT DIAGRAM



CIRCUIT DIAGRAM

How It Works:

The Traffic Density Monitoring System works by using an Ultrasonic Sensor (or IR Sensor) to detect vehicle presence and measure traffic density. The sensor sends signals to the Arduino/ESP32 microcontroller. Based on the detected traffic level, the microcontroller controls the traffic signal LEDs (Red, Yellow, Green).

The LCD (16x2 Display) shows the traffic status such as:

- “High Traffic”
- “Low Traffic”
- “Signal Green”
- “Signal Red”

If traffic density is high, the green signal stays ON for a longer duration. If traffic density is low, the green signal stays ON for a shorter time. The buzzer (optional) can be used to indicate signal change or emergency alerts.

Key Steps of Circuit Diagram:

Step 1: Power Supply

- Arduino/ESP32 is powered using USB or 5V adapter.
- LCD and LEDs receive 5V supply.
- Sensor is connected to appropriate voltage (5V or 3.3V depending on module).
- All components share common GND.

Step 2: Sensor Connection

- Ultrasonic Sensor:
 - VCC → 5V
 - GND → GND
 - Trig → Digital Pin
 - Echo → Digital Pin
- OR
- IR Sensor:
 - VCC → 5V
 - GND → GND
 - OUT → Digital Pin

Step 3: LCD Connection (I2C)

- VCC → 5V
- GND → GND
- SDA → A4 (Arduino UNO)
- SCL → A5 (Arduino UNO)

Step 4: LED Connections

- Red LED → Digital Pin
- Yellow LED → Digital Pin
- Green LED → Digital Pin
- Resistors used for current limiting.

Step 5: Buzzer (Optional)

- Positive → Digital Pin
- Negative → GND

Step 6: Program Execution

- Arduino reads sensor data continuously.
- Traffic density is calculated.
- Signal timing is adjusted automatically.
- LCD displays traffic information.

Benefits of Circuit Diagram:

- ✓ Clear understanding of hardware connections
 - ✓ Prevents incorrect wiring
 - ✓ Makes troubleshooting easier
 - ✓ Helps in identifying power and signal flow
 - ✓ Useful for academic documentation
 - ✓ Easy to expand system (add WiFi, Cloud, Camera, etc.)
 - ✓ Improves project presentation quality
-

When to Use a Circuit Diagram?

Use a circuit diagram when:

- Building electronic hardware projects
- Submitting academic projects
- Troubleshooting wiring issues
- Testing before breadboard assembly
- Explaining system design in viva
- Creating technical documentation

Without a circuit diagram, there is a high chance of wiring mistakes.

Circuit Model Testing Methods

To ensure the system works properly, follow these testing methods:

1 Visual Inspection

- Check loose wires

- Verify correct pin numbers
 - Ensure proper voltage supply
 - Confirm common ground
-

2 Power Testing

- Measure voltage using multimeter
 - Sensor → 5V
 - LCD → 5V
 - Arduino → Proper power
-

3 Continuity Testing

- Check for short circuits
 - Confirm all grounds are connected
-

4 Component-wise Testing

Test individually before full integration:

- ✓ Upload LED Blink Program
 - ✓ Test each LED separately
 - ✓ Test Buzzer with simple ON/OFF code
 - ✓ Run Ultrasonic sensor distance test code
 - ✓ Run LCD test code (Hello World)
-

5 Software Debug Testing

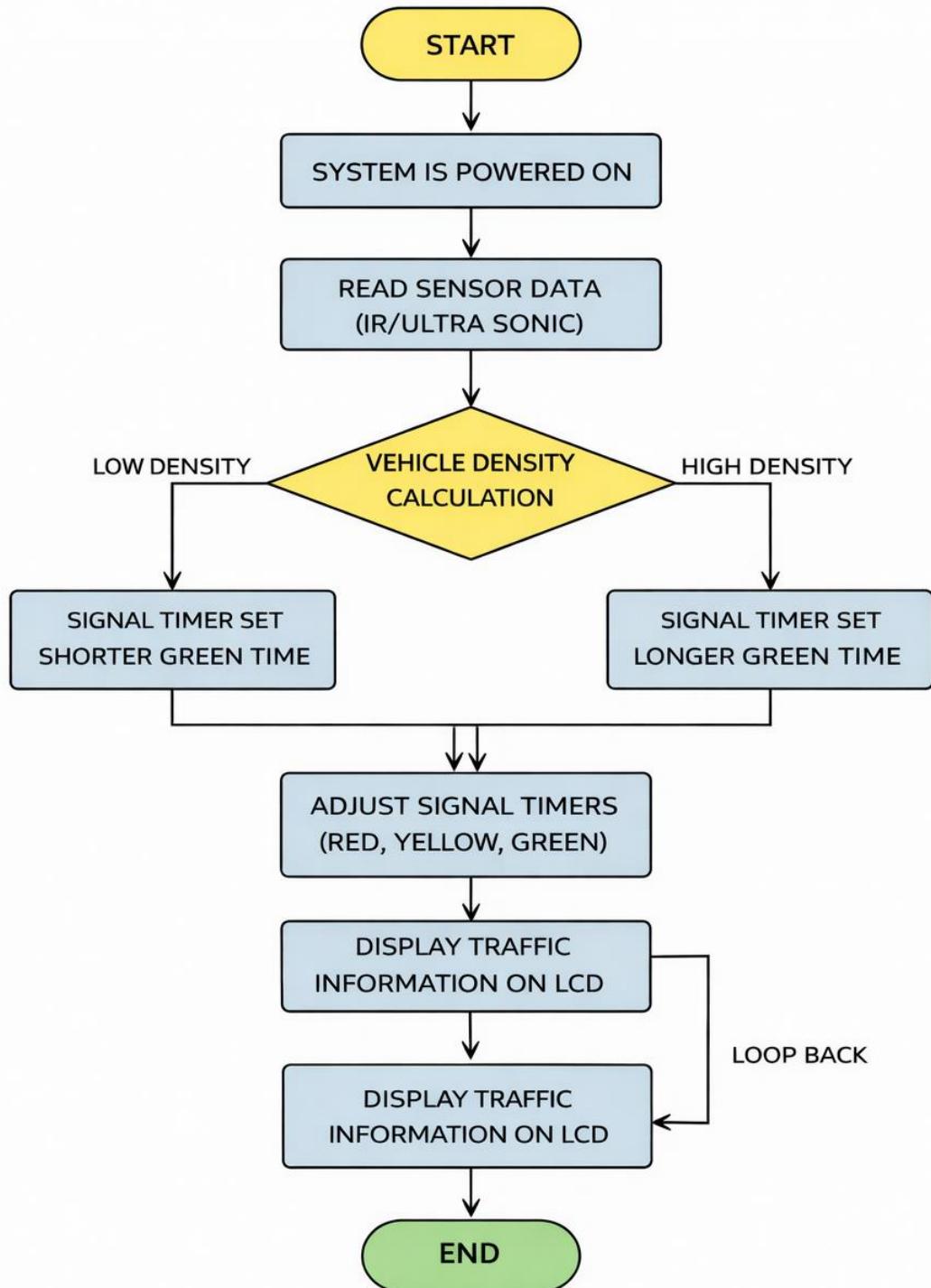
- Use Serial Monitor
 - Print sensor values
 - Verify signal timing logic
 - Check condition statements
-

6 Stress Testing

- Simulate high traffic repeatedly
- Check signal switching timing
- Observe LCD output consistency
- Verify system stability

Activity Diagram :

Traffic Density Monitoring System Activity Diagram



Activity Diagram Description

The Activity Diagram represents the complete working process of the Traffic Density Monitoring System. It visually explains how the system operates step-by-step from start to end. The process begins when the system is powered ON. After initialization, the microcontroller reads data from the IR or Ultrasonic sensor to detect the presence and number of vehicles. Based on the sensor input, the system performs vehicle density calculation. At this decision point, the system determines whether the traffic density is low or high.

- If traffic density is low, the system sets a shorter green signal duration.
- If traffic density is high, the system sets a longer green signal duration.

After setting the appropriate timer, the traffic signals (Red, Yellow, Green) are adjusted accordingly. The system then displays traffic information on the LCD screen. The process continues in a loop, constantly monitoring traffic conditions and adjusting signal timings in real time.

Finally, the process ends only when the system is turned OFF.

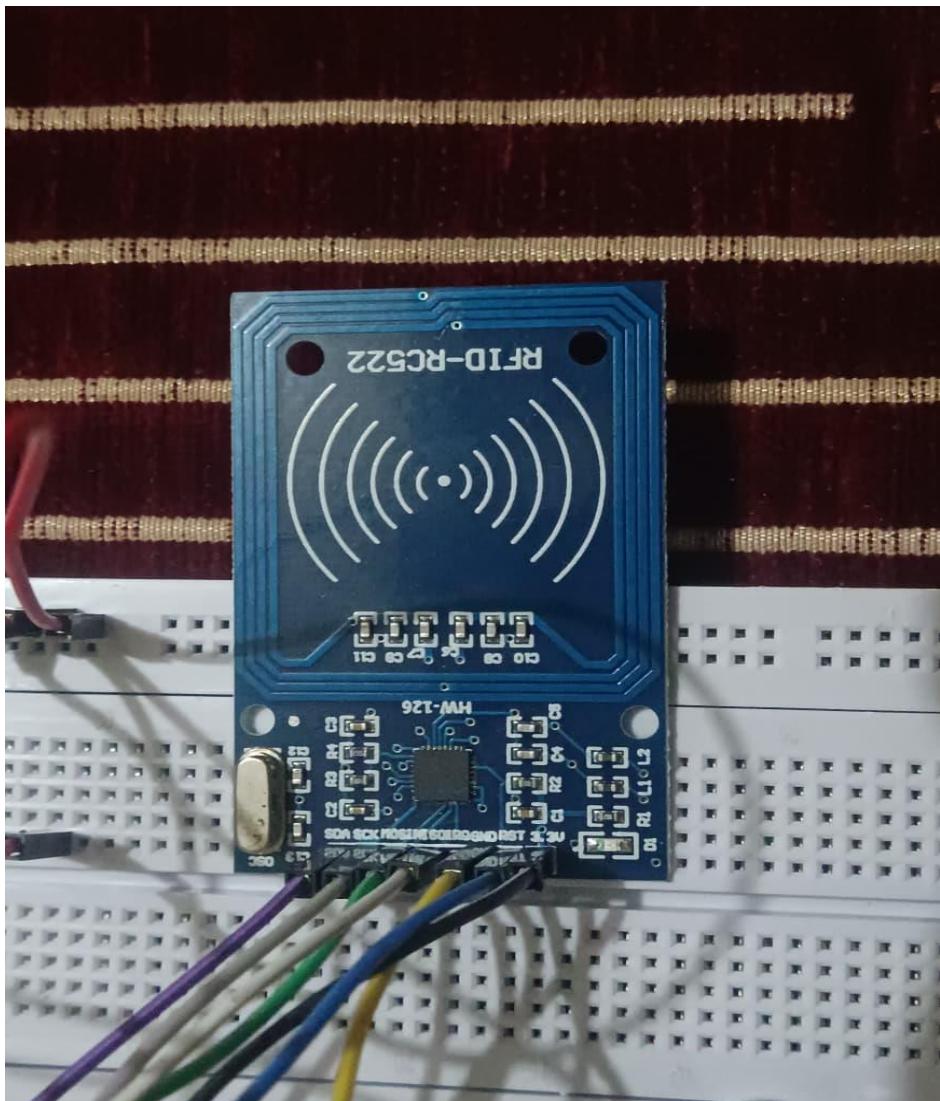
Purpose of the Activity Diagram

The Activity Diagram is created for the following purposes:

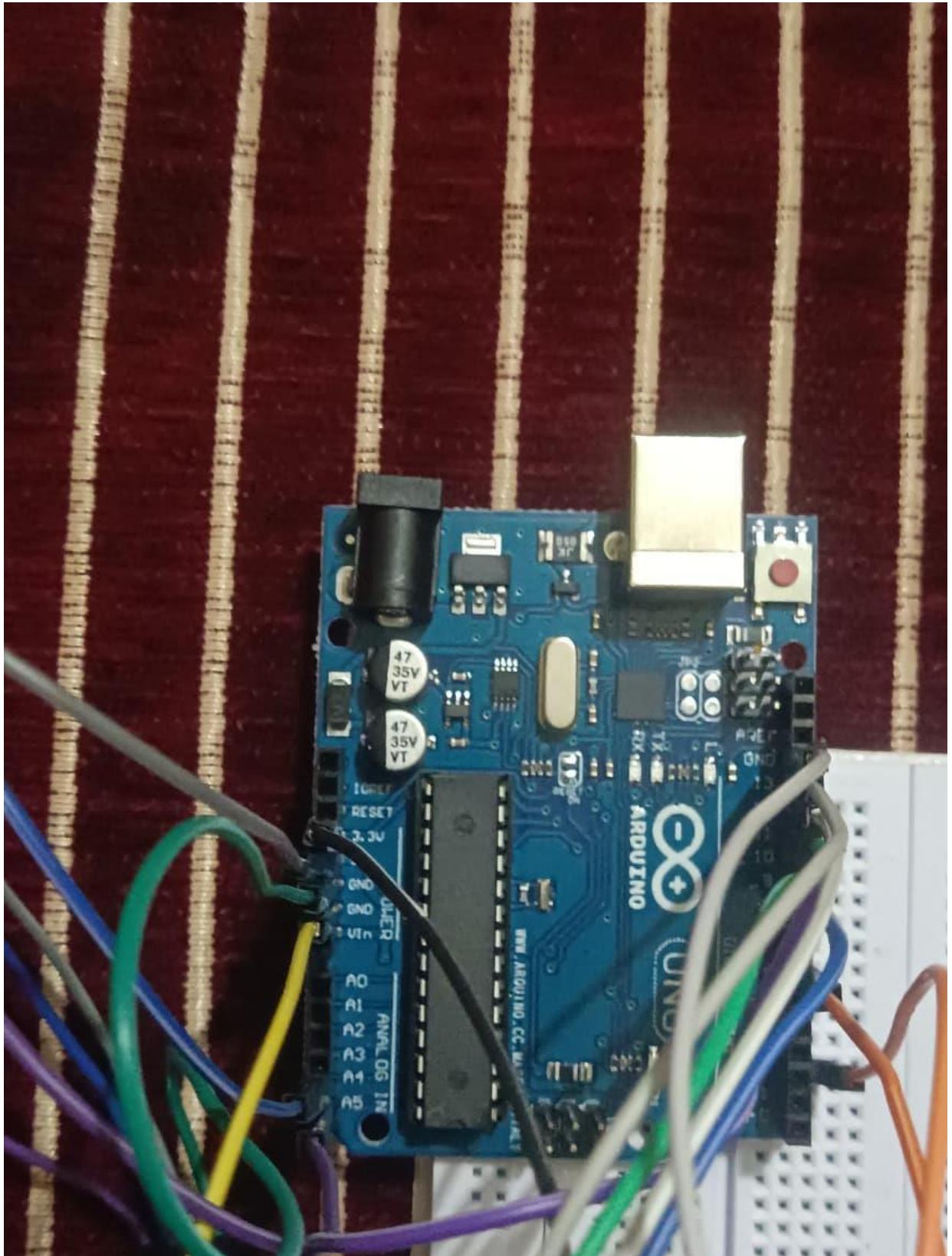
1. To visually represent the workflow of the system.
2. To show the sequence of operations clearly from start to end.
3. To explain decision-making points (Low Density / High Density).
4. To simplify understanding of system logic before coding.
5. To help in project documentation and presentation.
6. To identify possible logical errors in system design.
7. To improve communication between team members.
8. To make viva explanation easier.
9. To provide a structured model of system behavior.
10. To demonstrate real-time looping process in traffic monitoring.

SYSTEM IMPLEMENTATION

Step By Step assemble:

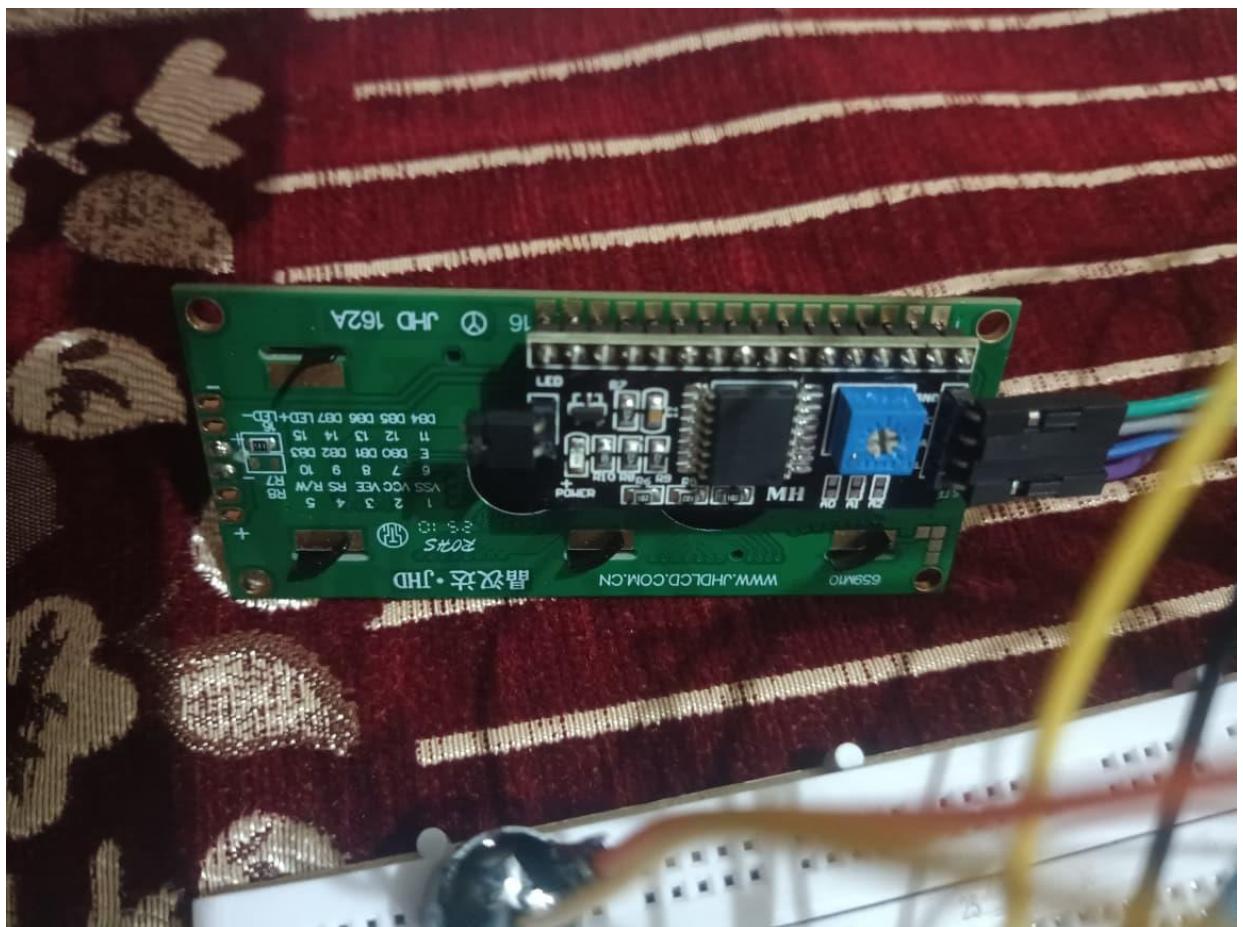


Step 1: Joining Male To Male Jumper wire in the RC522 Module

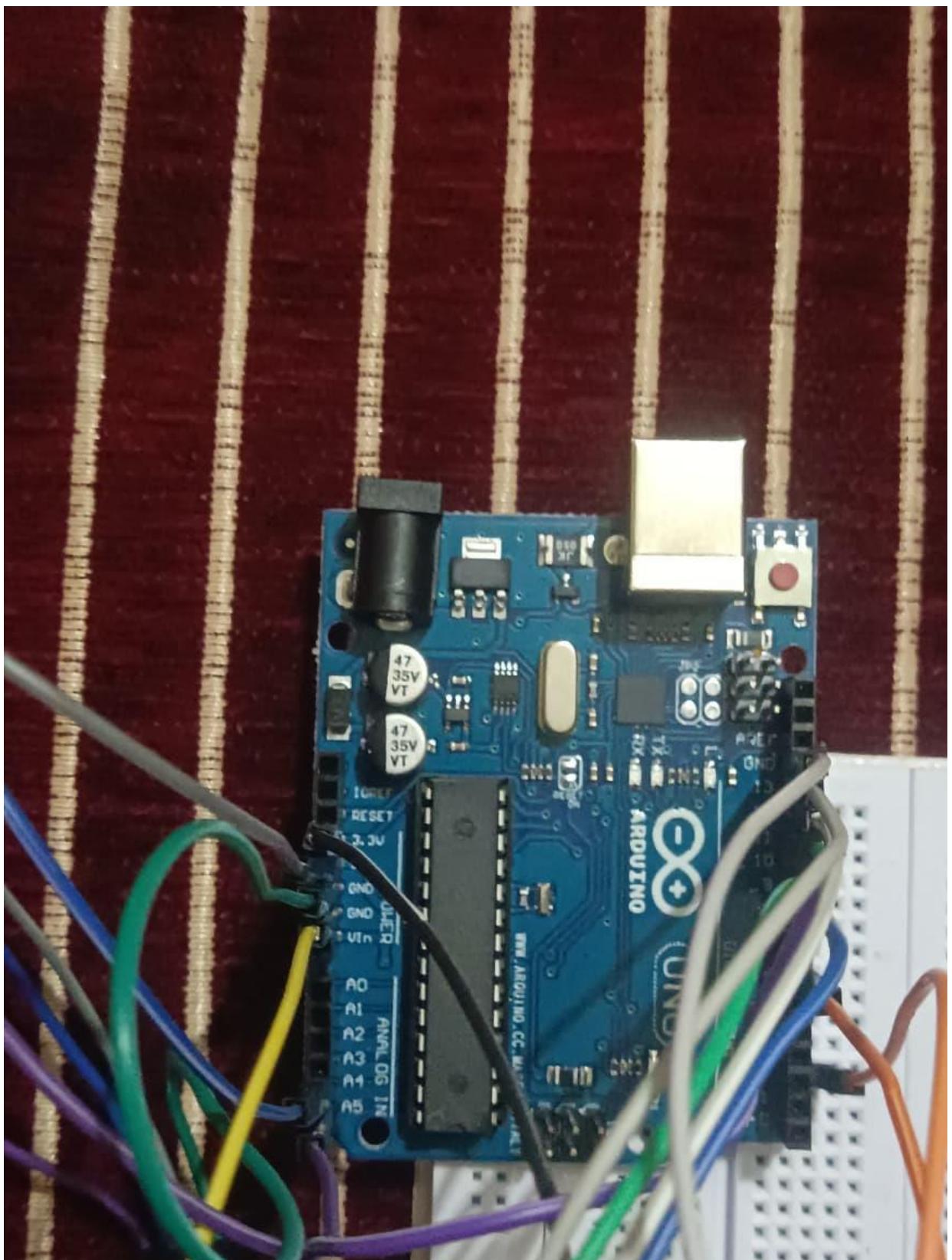


Step 2: Now joining the same wire which we had joined in the RC522 module the other end will be joined in Arduino UNO. As Written Below:

RFID Module	Arduino UNO
SDA	10 pin
SCK	13 pin
MOSI	11 pin
MISO	12 pin
IRQ	No Pin
GND	GND
RST	9 pin
3.3V	3.3 V

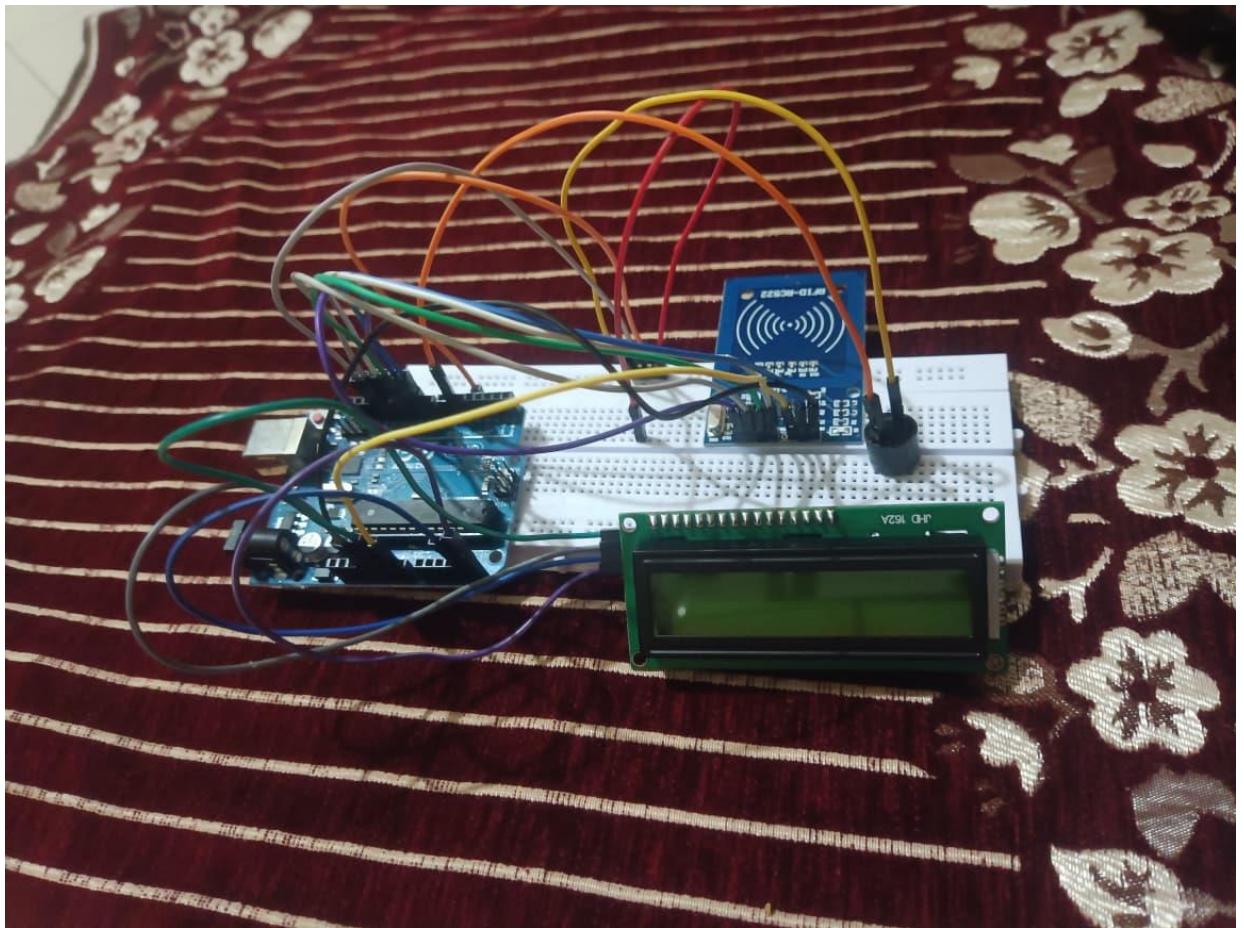


Step 3: Now We will join Male to Female Jumper Wires in I2C Module which is attached in LCD Display.



Step 4: Again will Join the I₂C module wire in Arduino UNO accordingly. As written Below:

I2C Module	Arduino UNO
GND	GND
VCC	5 v
SDA	A-4
SCL	A-5



Step 5: Completed!

Step 6: Now Lets Create A Programme so that is can perform.

Code:

```
#include <SPI.h>
#include <MFRC522.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define SS_PIN 10
#define RST_PIN 9
#define BUZZER 2

MFRC522 mfrc522(SS_PIN, RST_PIN);
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
String uidSunny = "97 13 B4 51";
String uidRahul = "11 22 33 44";
String uidAmit = "62 63 BA 5C";

// ===== Attendance Status =====
bool sunnyMarked = false;
bool rahulMarked = false;
bool amitMarked = false;

int totalPresent = 0;

void setup() {
    Serial.begin(9600);
    SPI.begin();
    mfrc522.PCD_Init();
    lcd.init();
    lcd.backlight();
    pinMode(BUZZER, OUTPUT);

    showMainScreen();
}

void loop() {

    if (!mfrc522.PICC_IsNewCardPresent()) return;
    if (!mfrc522.PICC_ReadCardSerial()) return;

    String content = "";

    for (byte i = 0; i < mfrc522.uid.size; i++) {
        content += String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
        content += String(mfrc522.uid.uidByte[i], HEX);
    }

    content.toUpperCase();
    content.trim();

    lcd.clear();

    // ===== SUNNY =====
    if (content == uidSunny) {
        if (!sunnyMarked) {
            sunnyMarked = true;
            totalPresent++;
            lcd.setCursor(0,0);
            lcd.print("Welcome Sunny");
            beep();
        }
    }
}
```

```
        } else {
            lcd.setCursor(0,0);
            lcd.print("Already Marked");
        }
    }

// ===== RAHUL =====
else if (content == uidRahul) {
    if (!rahulMarked) {
        rahulMarked = true;
        totalPresent++;
        lcd.setCursor(0,0);
        lcd.print("Welcome Rahul");
        beep();
    } else {
        lcd.setCursor(0,0);
        lcd.print("Already Marked");
    }
}

// ===== AMIT =====
else if (content == uidAmit) {
    if (!amitMarked) {
        amitMarked = true;
        totalPresent++;
        lcd.setCursor(0,0);
        lcd.print("Welcome Amit");
        beep();
    } else {
        lcd.setCursor(0,0);
        lcd.print("Already Marked");
    }
}

// ===== UNKNOWN CARD =====
else {
    lcd.setCursor(0,0);
    lcd.print("Access Denied");
}

// Always show total on second line
lcd.setCursor(0,1);
lcd.print("Present: ");
lcd.print(totalPresent);

delay(2000);
showMainScreen();
```

```
mfrc522.PICC_HaltA();  
}  
  
void showMainScreen() {  
    lcd.clear();  
    lcd.setCursor(0,0);  
    lcd.print("Scan Your Card");  
    lcd.setCursor(0,1);  
    lcd.print("Present: ");  
    lcd.print(totalPresent);  
}  
  
void beep() {  
    digitalWrite(BUZZER, HIGH);  
    delay(200);  
    digitalWrite(BUZZER, LOW);  
}
```

OUTCOME:

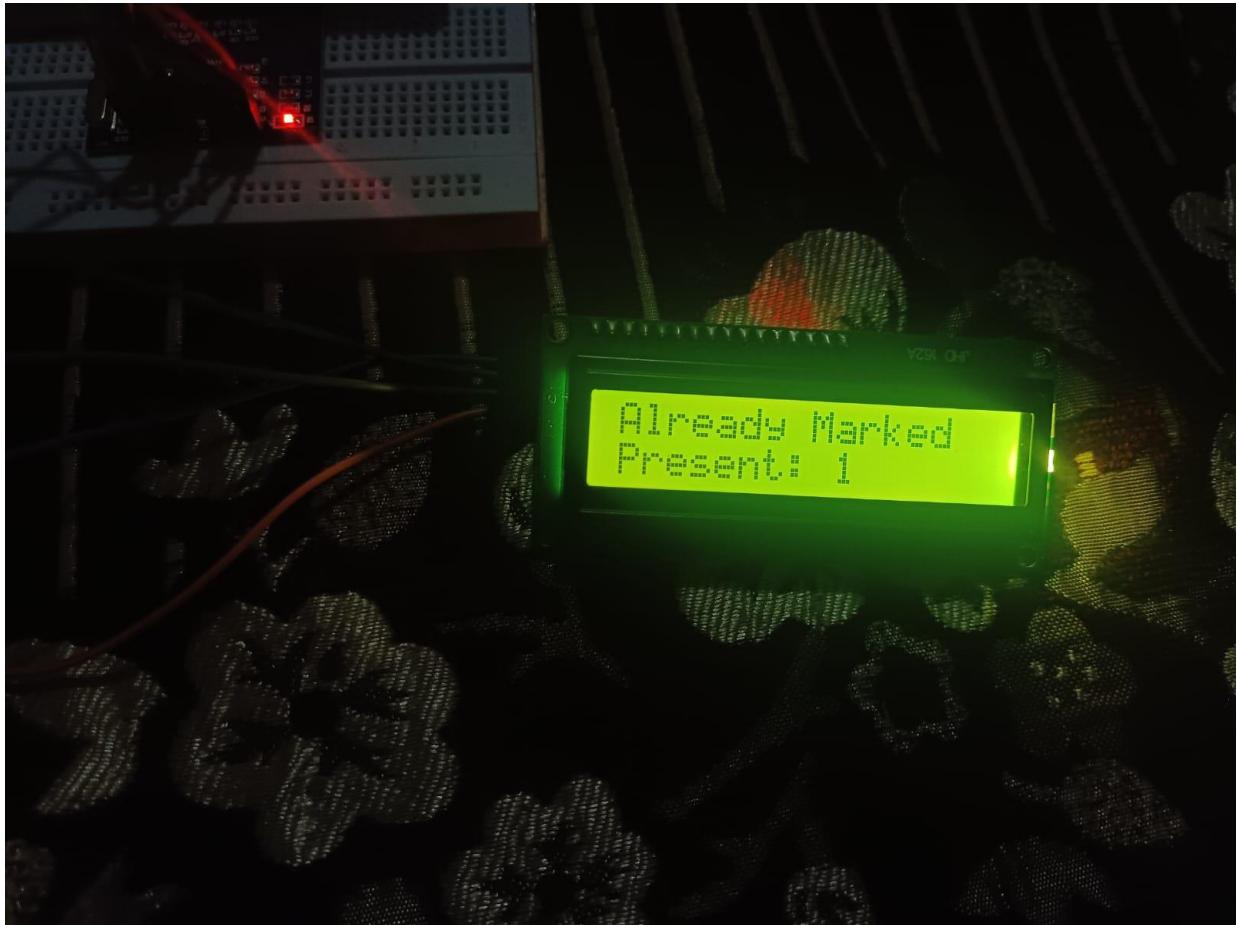
Case 1: When a Card whose UID is Not Registered is scanned:



Case 2: when a card with legit UID is Scanned:



Case 3: When a card which I already got scanned try to get scanned again:



FUTURE SCOPE AND CONCLUSION

Future Scope

- AI-based traffic prediction
 - Emergency vehicle detection
 - GPS-based tracking
 - Cloud data storage
 - Mobile app monitoring
 - CCTV integration
 - Smart city implementation
-

Conclusion

The Traffic Density Monitoring System provides an intelligent solution for traffic management. It dynamically controls traffic signals based on vehicle density, reducing congestion, saving fuel, and minimizing pollution. The system is scalable, cost-effective, and suitable for smart city applications.

REFERENCES

- Arduino Official Website –
<https://www.arduino.cc/>
- Arduino IDE –
<https://www.arduino.cc/en/software>
- IoT Traffic Control Research Papers
- ChatGPT

1. Prof. Sahil Shukla

GLOSSARY

Glossary:

- **Traffic Density:** Number of vehicles in a specific area.
- **IR Sensor:** Infrared sensor used to detect objects.
- **Microcontroller:** A programmable device that controls the system.
- **IoT:** Internet of Things.
- **Smart Signal:** Signal that adjusts timing automatically.
- .