**V2V COMMUNICATION & POTHOLE DETECTION**

|  |  |  |  |
| --- | --- | --- | --- |
| **Team Number** | **Full Name** | **USN** | **Section** |
| **62** | **ANUSHREE VINOD NAIK** | **1RVU23CSE069** | **H** |
| **CHAITANYA A S** | **1RVU23CSE126** | **A** |
| **DEEPAK K S** | **1RVU23CSE156** | **D** |
| **GOPIKA R** | **1RVU23CSE170** | **A** |
| **H S VEERESH** | **1RVU23CSE178** | **D** |
| **KRITI RAVI PANDIT** | **1RVU23CSE225** | **I** |

Submitted To : RV University

In partial fulfillments in the requirements for the course of

CS1822: ENGINEERING EXPLORATION

Project work carried out under the guidance of

**<Prof. Veena .S , Assistant Professor >**

Under the domain of

**INTELLIGENT TRANSPORT SYSYTEM**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Abstract  (2 Marks) | Introduction  (2 Marks) | Materials / Components Used  (4 Marks) | Circuit & Flowchart  (4 Marks) | Data Acquisition / Analysis / Visualization  (6 Marks) | Discussion /Conclusion  (4 Marks) | References  (2 Marks) | Formatting  (1 Mark) |
| Marks |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Marks Obtained (Max 25):** |  |
| **Signature of evaluator:** |  |

**TITLE:**

**VEHICLE TO VEHICLE COMMUNICATION & POTHOLE DETECTION - ENHANCING ROAD SAFETY**

**Acknowledgement**

Domain Leads: Prof VEENA S

We would like to express our sincere gratitude to our domain leads, Prof VEENA S mam, for their valuable guidance and expertise throughout the development of this project.

I acknowledge and appreciate the valuable insights and feedback provided by the professors, without whom this research would not have been possible.

**INDEX**

|  |  |  |
| --- | --- | --- |
| SL. NO. | TABLE OF CONTENTS | PAGE NO. |
| 1. | Title & Acknowledgement | PAGE 2 |
| 2. | Abstract | PAGE 4 |
| 3. | Introduction | PAGE 4 |
| 4. | Materials / Components Used | PAGE 5-6 |
| 5. | Circuits | PAGE 7-8 |
| 6. | Flowchart / Diagram | PAGE 9-12 |
| 7. | Data Visualisation | PAGE 13 |
| 8. | Discussion / Conclusion | PAGE 14 |
| 9. | References | PAGE 15 |
| 10. | Appendix | PAGE 16-22 |

**ABSTRACT**:

Vehicle-to-vehicle (V2V) communication's ability to wirelessly exchange information about the speed and position of surrounding vehicles shows great promise in helping to avoid crashes, ease traffic congestion, and improve the environment. Vehicle-to-Vehicle (V2V) communications is a system designed to transmit information between vehicles and vehicle on the road in real-time. This information provides warnings to drivers and other vehicles. Instead of cars working independently, vehicles will be able to transmit vital information to nearby vehicles to improve the overall efficiency and safety of the roadways. Vehicle-to-vehicle communication and Vehicle-to-vehicle communication for safety and security which is one of the main keys in the Intelligent Transportation System (ITS). This type of technology will allow the Department of Transportation to achieve its goal of zero automobile deaths within the next decades. Additionally in Pothole detection, the vehicle detects the pothole and sends the alert message to the driver.

**INTRODUCTION:**

In today's context , road safety is a critical concern, with accidents & damages caused by potholes being significant contributors to road hazards. To address these challenges, we present a prototype of a Vehicle-to-Vehicle (V2V) Communication system coupled with a pothole detection mechanism. The primary purpose of this project is to strengthen road safety by alerting the drivers , displaying about potential hazards on the road , enabling them to take preventive actions and avoid accidents .

MOTIVATION: The motivation behind this project originates from vital necessity to minimize road accidents and damages caused by potholes, which pose serious risks to drivers and vehicles.

WORK DONE: utilizing the technology , particularly Arduino Uno , Ultrasonic sensors , NRF24L01 wireless module with in-built antenna ,GPS module , and an LCD display, our aim is to create a cost-effective solution that can be easily incorporated in many other vehicles .

LITERATURE SURVEY: Vehicle Position and Context Detection Using V2V Communication” by Paul Watta, Ximu Zhang, and Yi Lu Murphey(IEEE Transactions on Intelligent Vehicles, Volume: 6, Issue: 4, December 2021)

Implementation of In-Vehicle and V2V Communication with Basic Safety Message Format Communication”.Vibin .V, Sivraj P, Dr.V. Vanitha Department of Electrical and Electronics Engineering Amrita School of Engineering, Coimbatore

**Materials / Components Used:**

1)**ARDUINO UNO**:

•Microcontroller: ATmega328P

•Operating Voltage: 5V

•Digital I/O Pins: 14.

•Analog Input Pins: 6

•Speed: 16 MHz

•USB Interface: ATmega16U2

**Working: The Arduino functions as the project's central processing unit, executing programmed tasks to manage communication between components, process sensor data, and control the LCD display.**

**2)GPS MODULE:**

NEO-6M

•Number of Channels: 50Receiver

Communication Protocol: NMEA

•Update Rate: Up to 5 Hz Position

• Accuracy: <2.5 meters.

Voltage: Typically 3.3V

**Working:Global Positioning System, uses a network of satellites orbiting the Earth to determine your location. Each satellite sends out signals that include the satellite's position and the precise time the signal was sent.**

3)**ULTRASONIC** **SENSORS**

* Detection Range: up to 100 cm
* Operating Voltage: 5V
* Frequency: 40kHz
* Detection Range: Up to 100 cm
* Operating Angle: 15 degrees
* Measurement Accuracy: Varies with distance and sensor quality, generally within a few millimeters to centimeters
* **Working Principle: Ultrasonic sensors emit high-frequency sound waves and calculate the time taken for the waves to reflect back from objects. By knowing the speed of sound, they determine the distance to the object.**

4)**NRF24L01** **MODULE**

* Communication Range: Up to 4 meters
* Data Rate: Up to 2 Mbps.
* Operating Voltage : 1.9V .
* Frequency: 2.4Ghz.
* Power Supply: 1.9V to 3.6V
* Interfaces: SPI interface for communication with microcontrollers like Arduino.

**Working: It facilitates wireless communication with both transmitter and receiver modules. It transmits distance data from one vehicle's ultrasonic sensor to another using RF protocol, enabling vehicle-to-vehicle communication**.



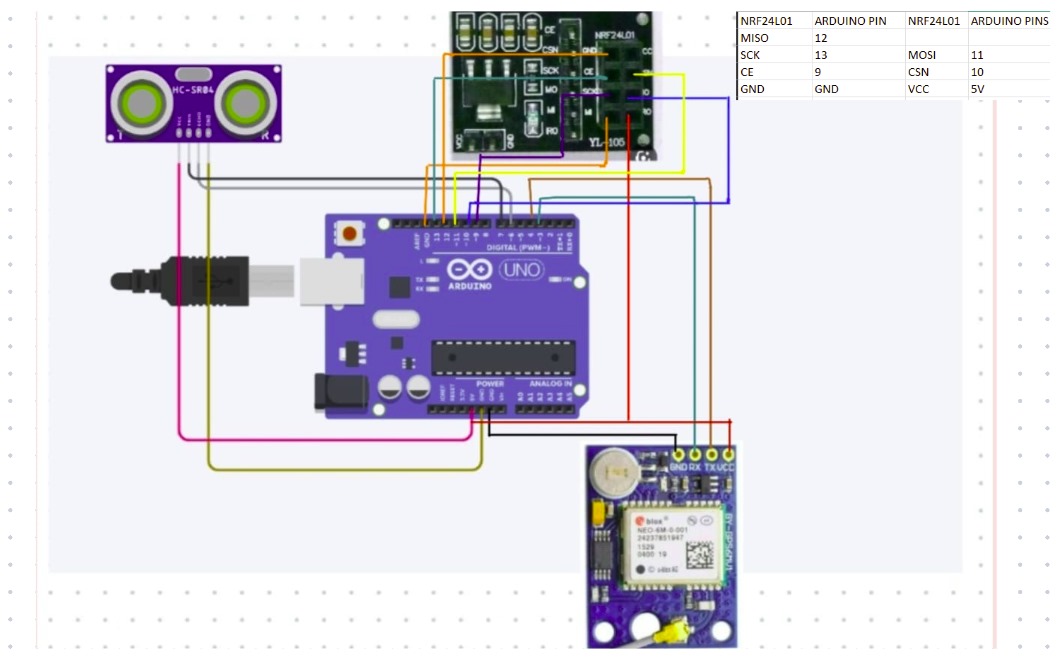
5)**LCD** **DISPLAY**

* Display Type: Liquid Crystal Display (LCD) .
* Display Size: 16\*2.
* Operating Voltage: Typically 5V.
* Contrast via potentiometer adjustment.

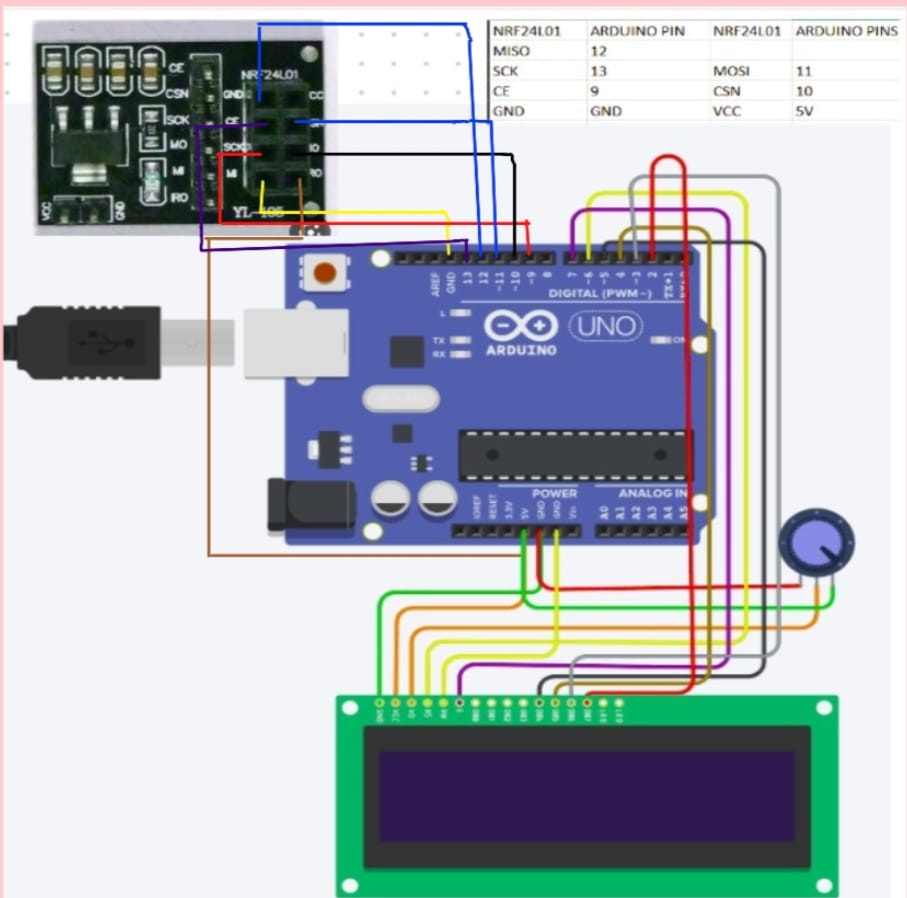
**Working : The LCD screen visually communicates various information such as distance between vehicles received from the NRF24L01 module.**

CIRCUITS:

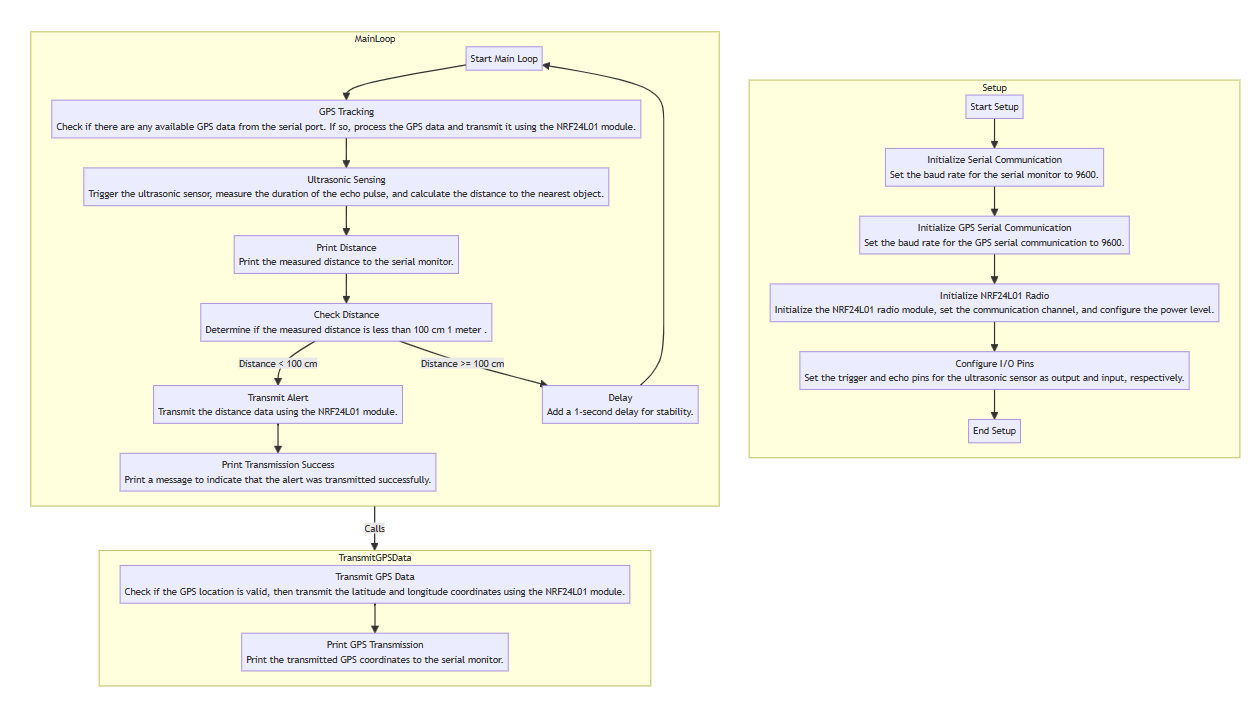
TRANSMITTER CAR:

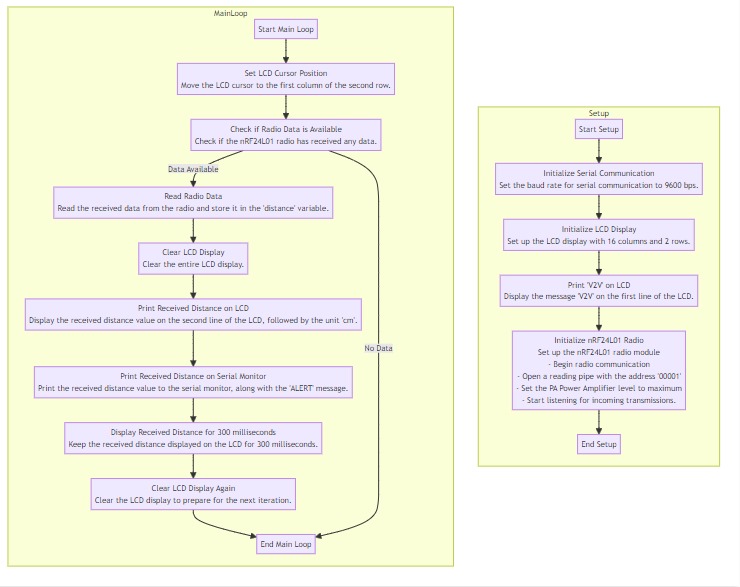


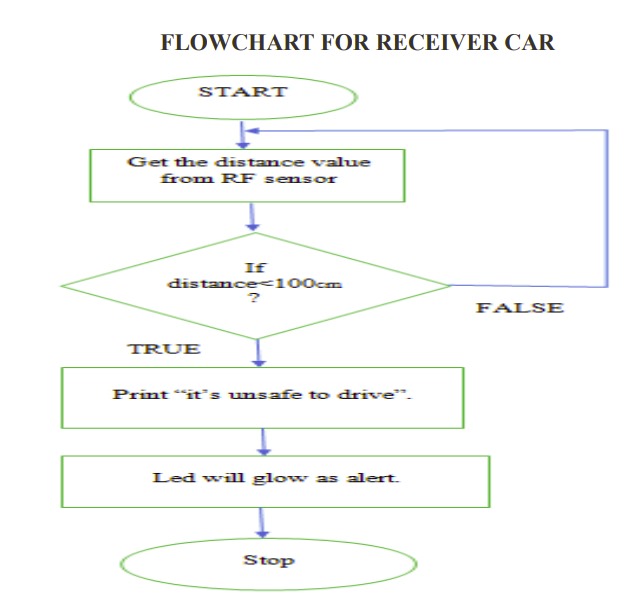
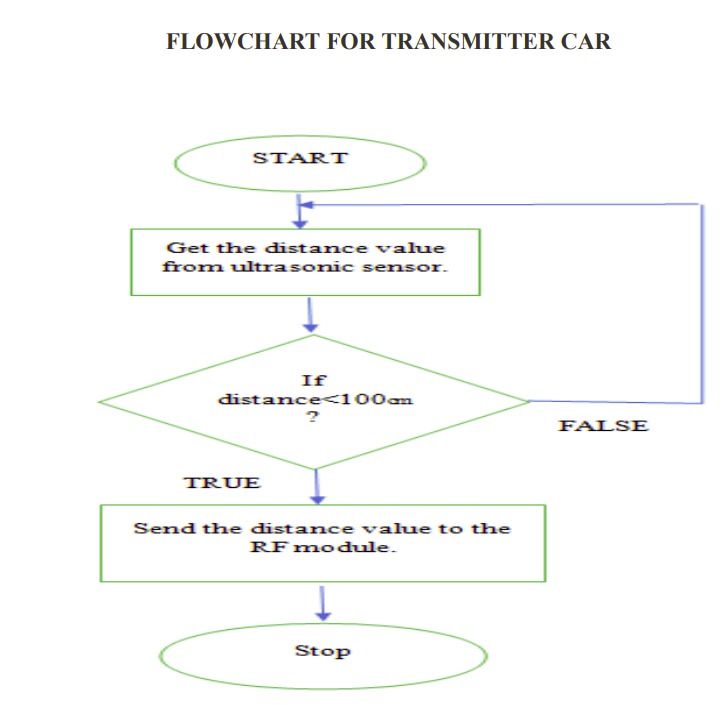
RECIEVER CAR:



FLOW CHART:







**Data:**

**Data Acquisition:** The key data being collected includes the distance between neighboring vehicles (measured by ultrasonic sensors) and the GPS-based location (latitude and longitude) of the transmitting vehicle.

**Data Storage:** The distance and location data is transmitted in real-time from the transmitter to the receiver vehicle via an NRF module.

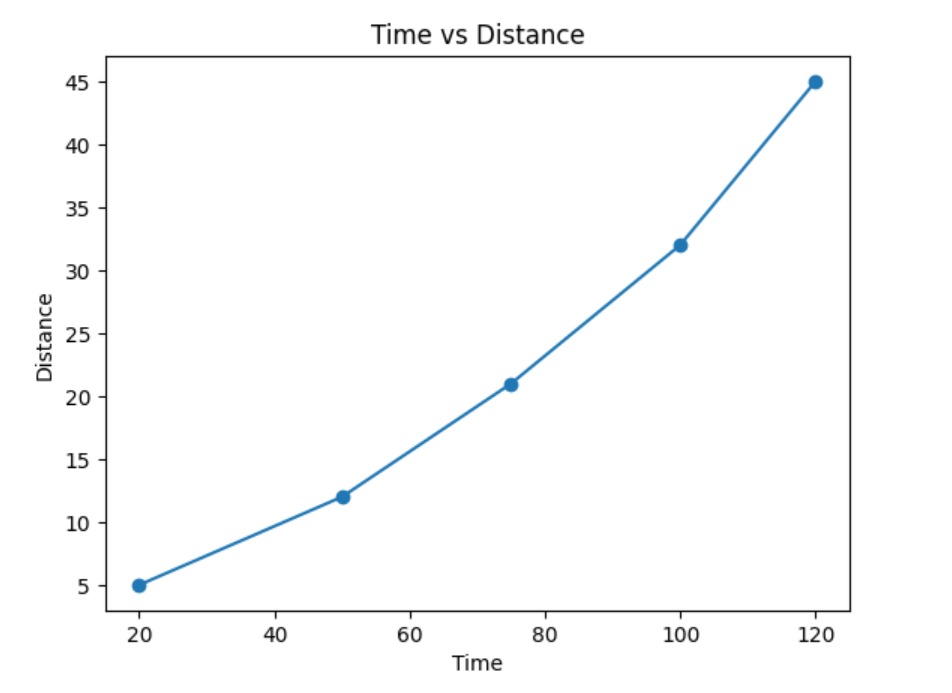
**Data Duration:** The data collection and transmission occurs continuously as long as the vehicles are within range of each other (less than 1 meter).

**Data Analysis and Inference**

**Proximity Detection:** The distance data allows the receiving vehicle to detect when a neighboring vehicle is within 1 meter, indicating a close proximity.

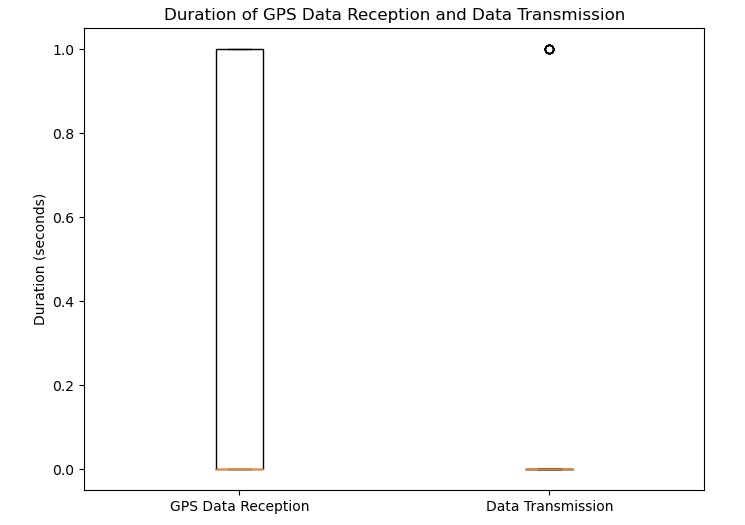
**Relative Positioning:** The GPS location data provides the relative position of thetransmitting vehicle, enabling the receiving vehicle to understand its surroundings.

**Collision Avoidance:** By combining the distance and location data, the system can infer potential collision risks and alert the driver or trigger autonomous safety features.



Distance(cm)

Time (s)



**Discussion / Conclusion:**

The model makes progress in road safety by detecting nearby vehicles and notifying the driver instantly. This new approach not only helps prevent accidents, but also improves overall traffic flow. As vehicle-to-vehicle (V2V) technologies continue to evolve, they are expected to revolutionize automotive safety. The proposed use of vacuum for research and advertising represents a new and promising solution. The use of this type of technology has the potential to increase the efficiency of the road while emphasizing the safety of all passengers.

The design has revolutionized road safety in the transportation and automotive industry by detecting nearby vehicles and alerting drivers, ultimately improving traffic flow and preventing collisions. Improve road layout and traffic management to create a safe and efficient city. To save a life. Additionally, its integration with smart city concepts has increased decision-making information and supported the development of smart city areas.

The use of this technology not only increases the safety and efficiency of traffic, but also contributes to the sustainability of the environment. The model will help reduce pollution from traffic by reducing traffic accidents and related accidents, resulting in cleaner air and a better environment for everyone.

**References:**

1.Watta, P., Zhang, X., & Murphey, Y. L. (2021). Vehicle Position and Context Detection Using V2V Communication. IEEE Transactions on Intelligent Vehicles, 6(4)

2.Vibin, V., Sivraj, P., & Vanitha, V. (Year). Implementation of In-Vehicle and V2V Communication with Basic Safety Message Format Communication. Department of Electrical and Electronics Engineering, Amrita School of Engineering, Coimbatore.

**GitHub link of all the implemented code**:

<https://github.com/ChaitanyaAS/V2V-COMMUNICATION-POTHOLE>

**Appendix**:

**TRANSMITTER** **CAR**:

#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

RF24 radio (9, 10); // CE, CSN

const byte address [6] = "00001";

const int trigPin = 4;

const int echoPin = 5;

long duration;

int distance;

void setup () {

Serial.begin(9600);

radio.begin();

radio.openWritingPipe(address);

radio.setPALevel(RF24\_PA\_MAX);

radio.stopListening();

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

}

void loop() {

// Ultrasonic sensor code

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = duration \* 0.034 / 2;

// Print the distance to the serial monitor

Serial.print("Distance: ");

Serial.print(distance);

Serial.println(" cm");

// Check if the distance is less than 100 cm (1 meter)

if (distance < 100) {

// Transmit distance data using NRF24L01 module

radio.write(&distance, sizeof(distance));

// Print message to indicate successful transmission

Serial.println("Alert transmitted successfully.");

}

// Add delay for stability

delay (1000);

}

**RECEIVER** **CAR**:

#include <TinyGPS++.h>

#include <SoftwareSerial.h>

#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

// The serial connection to the GPS module

SoftwareSerial gpsSerial(3, 4); // RX, TX

TinyGPSPlus gps;

RF24 radio(9, 10); // CE, CSN

const byte address[6] = "00001";

const int trigPin = 4;

const int echoPin = 5;

long duration;

int distance;

void setup() {

Serial.begin(9600);

gpsSerial.begin(9600);

radio.begin();

radio.openWritingPipe(address);

radio.setPALevel(RF24\_PA\_MAX);

radio.stopListening();

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

Serial.println(F("GPS Module test"));

}

void loop() {

// This sketch displays information every time a new sentence is correctly encoded.

while (gpsSerial.available() > 0) {

if (gps.encode(gpsSerial.read())) {

displayInfo();

}

}

// If 5000ms pass without any data, something may be wrong.

if (millis() > 5000 && gps.charsProcessed() < 10) {

Serial.println(F("No GPS detected: check wiring."));

while(true);

}

// Ultrasonic sensor code

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = duration \* 0.034 / 2;

// Print the distance to the serial monitor

Serial.print("Distance: ");

Serial.print(distance);

Serial.println(" cm");

// Check if the distance is less than 100 cm (1 meter)

if (distance < 100) {

// Transmit distance data and GPS location using NRF24L01 module

radio.write(&distance, sizeof(distance));

radio.write(&gps.location.lat(), sizeof(gps.location.lat()));

radio.write(&gps.location.lng(), sizeof(gps.location.lng()));

radio.write(&gps.altitude.meters(), sizeof(gps.altitude.meters()));

radio.write(&gps.date.month(), sizeof(gps.date.month()));

radio.write(&gps.date.day(), sizeof(gps.date.day()));

radio.write(&gps.date.year(), sizeof(gps.date.year()));

radio.write(&gps.time.hour(), sizeof(gps.time.hour()));

radio.write(&gps.time.minute(), sizeof(gps.time.minute()));

radio.write(&gps.time.second(), sizeof(gps.time.second()));

// Print message to indicate successful transmission

Serial.println("Alert transmitted successfully.");

}

// Add delay for stability

delay(1000);

}

void displayInfo() {

if (gps.location.isValid()) {

Serial.print(F("Location: "));

Serial.print(gps.location.lat(), 6);

Serial.print(F(","));

Serial.println(gps.location.lng(), 6);

Serial.print(F("Altitude: "));

Serial.print(gps.altitude.meters());

Serial.println(F("m"));

} else {

Serial.println(F("Location: NOT FOUND"));

}

if (gps.date.isValid()) {

Serial.print(F("Date: "));

Serial.print(gps.date.month());

Serial.print(F("/"));

Serial.print(gps.date.day());

Serial.print(F("/"));

Serial.println(gps.date.year());

}

if (gps.time.isValid()) {

Serial.print(F("Time: "));

Serial.print(gps.time.hour());

Serial.print(F(":"));

Serial.print(gps.time.minute());

Serial.print(F(":"));

Serial.println(gps.time.second());

}

Serial.println();

}

**POTHOLE** **DETECTION**:

import cv2

import numpy as np

def pothole\_detected(frame):

# Convert frame to grayscale

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

# Apply Canny edge detection

edges = cv2.Canny(gray, 50, 150)

# Use contour detection to check if there are any contours (potential potholes)

contours, \_ = cv2.findContours(edges, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)

potholes = []

for contour in contours:

# Filter contours based on area

area = cv2.contourArea(contour)

if area > 2000: # Adjust this threshold based on your requirements

# Get bounding box coordinates of the contour

x, y, w, h = cv2.boundingRect(contour)

potholes.append((x, y, x + w, y + h)) # Store bounding box coordinates

return potholes

# Open the video file

cap = cv2.VideoCapture(r"C:\Users\chand\Downloads\real-time-object-detection-potholes-2160-ytshorts.savetube.me.mp4")

while True:

# Read a frame from the video

ret, frame = cap.read()

if not ret:

break

# Detect potholes

detected\_potholes = pothole\_detected(frame)

# Highlight potholes by drawing rectangles around bounding boxes

for (x1, y1, x2, y2) in detected\_potholes:

cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)

# Display the frame (for visualization, can be removed in actual implementation)

cv2.imshow('Frame', frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

# Release the video capture object and close all windows

cap.release()

cv2.destroyAllWindows()

void setup() {

Serial.begin(9600);

}

void loop() {

if (Serial.available() > 0) {

char signal = Serial.read();

// Perform actions based on received signals

if (signal == '1') {

// Example action: print a message to the Serial monitor

Serial.println("Pothole detected!");

// You can add other actions here

}

}

}

**THANK YOU!!!**