A project report on

LORA COMMUNICATION USING ARDUINO

Under the esteemed guidance of

Sri B. CHANTI M.Tech

UNDER AICTE IDEALAB

S.R.K.R. ENGINEERING COLLEGE (A)

(Affiliated to JNTU KAKINADA)(Recognicognized by A.I.C.T.E, New Delhi)

(Accredited by N.B.A., NAAC with 'A+' grade, New Delhi)

CHINNA AMIRAM,BHIMAVARAM-534204

AICTE
IDEALab

Certificate

This is to certify that Mr. GIDUTURI HEMANTH have carried out the project work on "LORA COMMUNICATION USING ARDUINO" at S.R.K.R. Engineering College (A), Bhimavaram. This is a bonafide record of the work done by us during the academic year 2022-2023. The results of this project work have not been submitted to any other university or Institute for the award of any degree.

Guide	Coordinator	Chief Mentor
B. CHANTI	**Dr. N. GOPALAKRISHNA MURTHY**	**Dr. M. JAGAPATHI RAJU**
PROJECT ASSOCIATE	AICET IDEA LAB COORDINATOR	AICET IDEA LAB CHIEF MENTOR

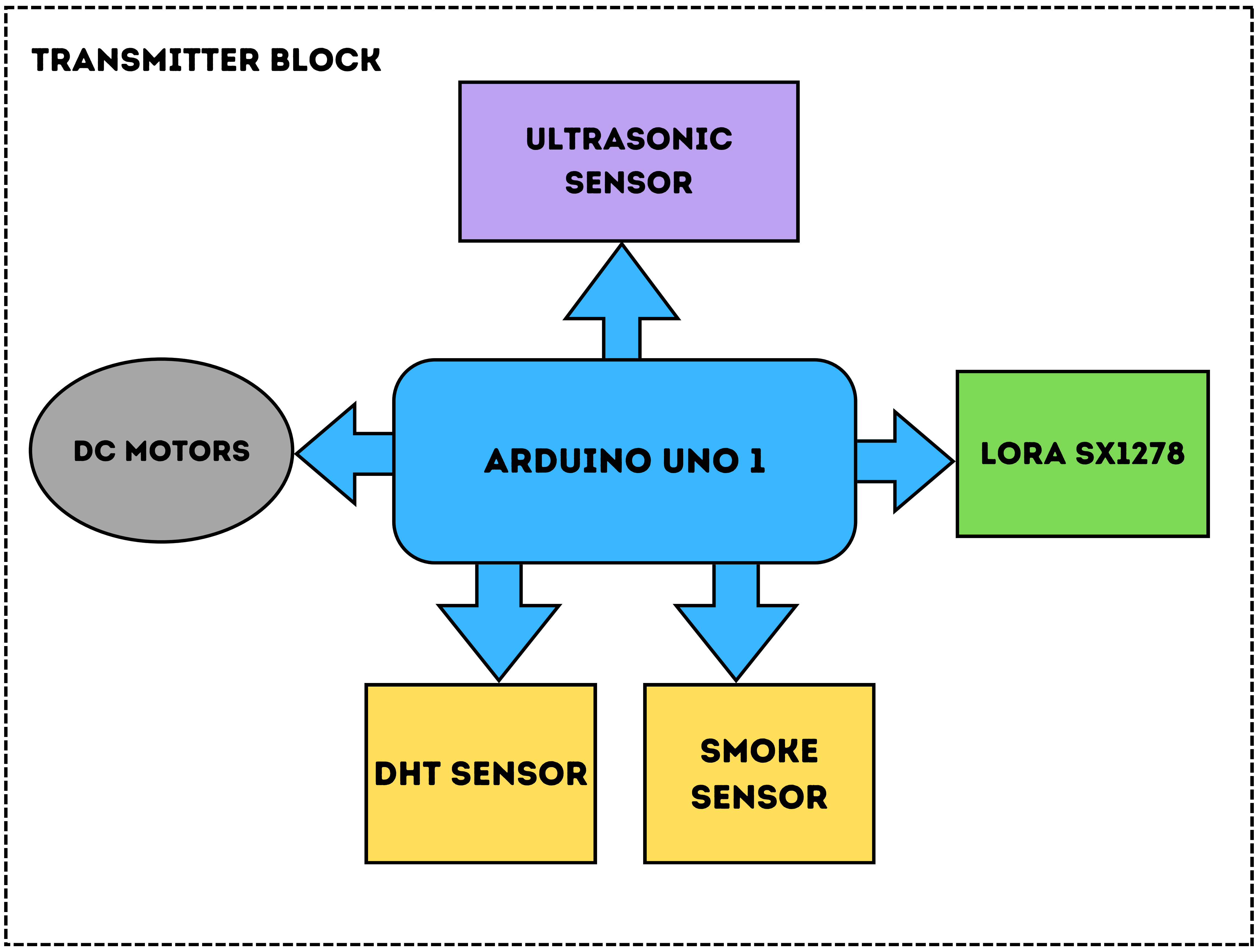
ABSTRACT:

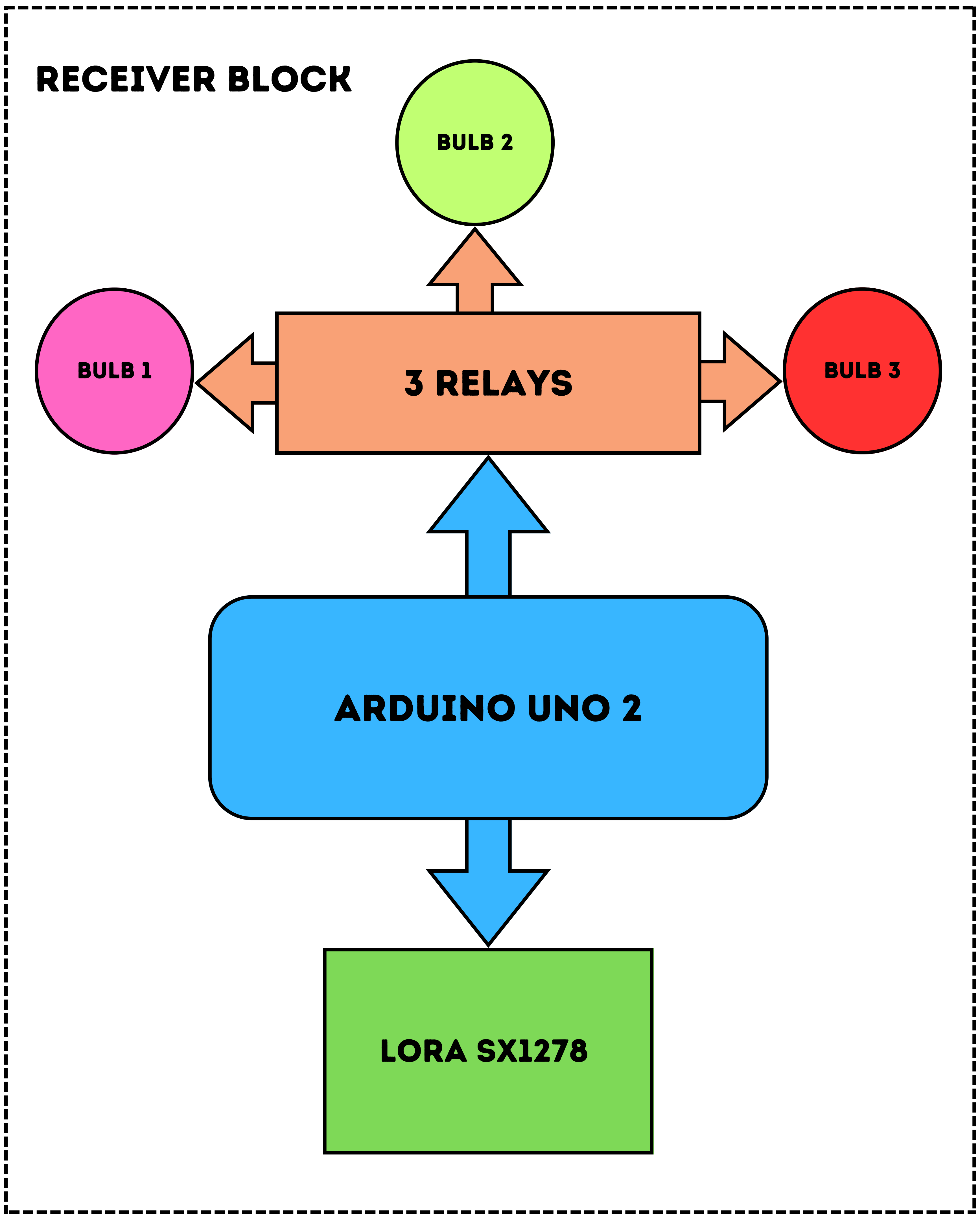
In this project, the utilization of two LoRa modules facilitated long-distance communication over distances of up to 10 kilometers. The primary objective was to enable an automated vehicle to transmit data from its onboard sensors to a remote location. One LoRa module was dedicated to gathering and transmitting the sensor data from the vehicle, while the complementary module was tasked with regulating light bulbs based on the received sensor information. The implementation relied heavily on Arduino programming, which was instrumental in managing the data transmission and controlling the light bulbs in response to the sensor readings. This project demonstrates the practicality and efficiency of using LoRa technology for remote monitoring and control applications in automated systems.

PROBLEM STATEMENT :

The objective of this project is to develop a reliable and efficient communication system using two LoRa (Long Range) modules to transmit sensor data over a distance of up to 10 kilometers. This system will facilitate the transmission of data collected from sensors installed on an automated vehicle to a remote location, where the data will be used to regulate light bulbs. The primary challenge is to ensure seamless, low-latency data transmission over the specified distance while maintaining data integrity and synchronization between the transmitting and receiving modules. Arduino programming will be employed to integrate the LoRa modules with the sensors and light bulbs, ensuring smooth operation and control. The project seeks to explore the potential of LoRa technology in long-range communication applications and demonstrate its viability for automation and remote control tasks.

FLOWCHART :


TRANSMITTER BLOCK
ULTRASONIC SENSOR
DC MOTORS
ARDUINO UNO 1
LORA SX1278
DHT SENSOR
SMOKE SENSOR


RECEIVER BLOCK
BULB 2
BULB 1
3 RELAYS
BULB 3
ARDUINO UNO 2
LORA SX1278

COMPONENTS :

- Two LoRa SX1278 Modules
- Two Arduino Uno Boards
- DHT Sensor
- Smoke Sensor
- Ultrasonic Sensors
- Two DC Motors
- Four Wheels
- Three Different Colored Bulbs

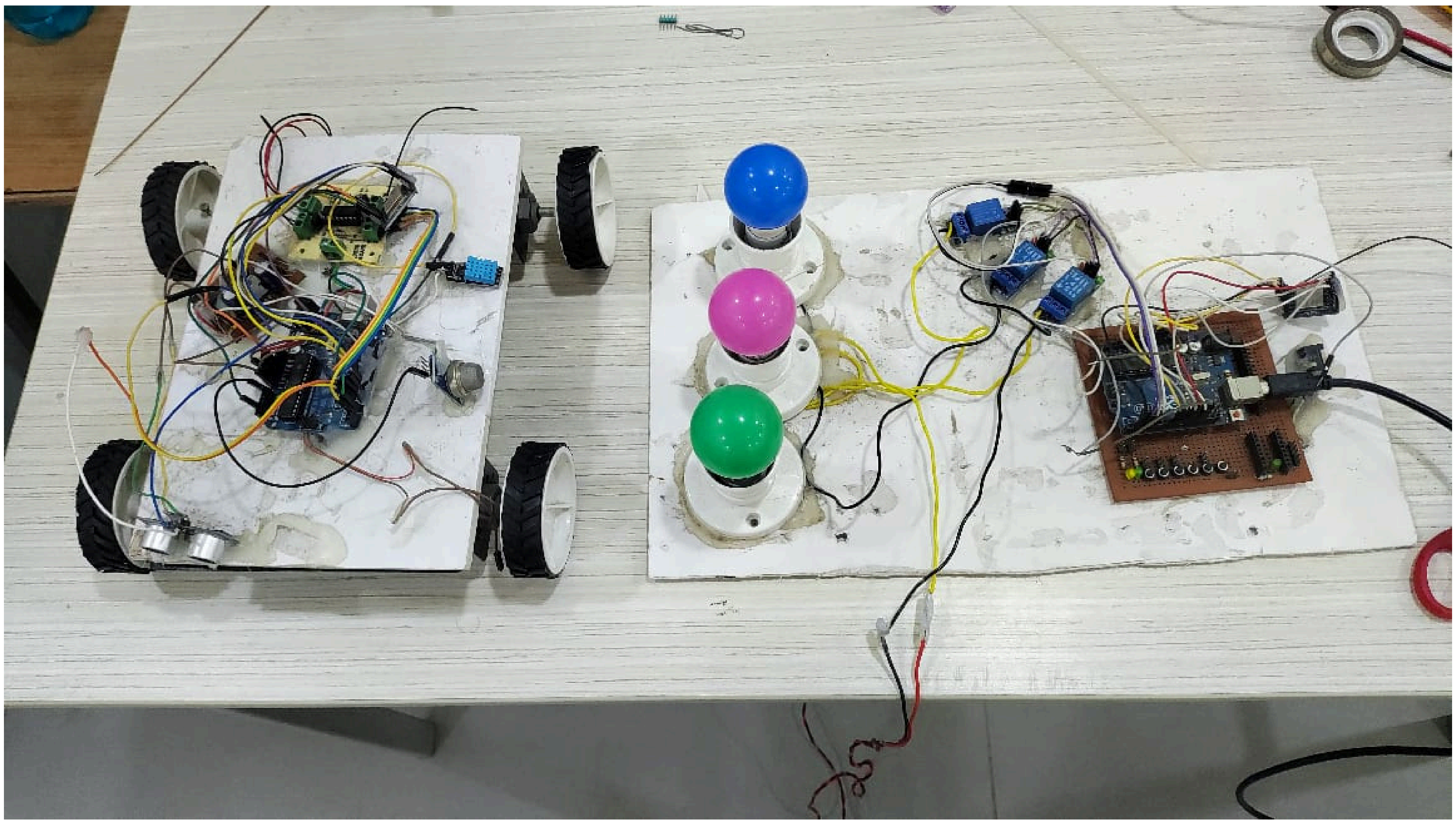
SOFTWARE :

- Arduino IDE
- DHT Libraries
- LoRa Libraries

WORKING :

1. Setting Up the Hardware :

- **Connecting the Motors and Wheels:** Fix the two DC motors to the rear wheels and attach the two dummy wheels to the front. Connect the motors to the motor driver module, which is then connected to the Arduino.
- **Mounting the Sensors:** Attach the ultrasonic sensor at the front of the car for effective obstacle detection. Connect the DHT sensor and smoke sensor to the Arduino for environmental monitoring.
- **Connecting the LoRa Modules:** Connect one LoRa SX1278 module to the transmitter Arduino and the other to the receiver Arduino. Ensure proper wiring for power and data transmission lines.



2. Writing the Code :

- **Arduino IDE Setup:** Install the necessary libraries for LoRa communication, ultrasonic sensor, DHT sensor, and smoke sensor.
- **Sensor Data Acquisition:** Write code to read data from the ultrasonic, DHT, and smoke sensors. For instance, use the 'dht.readTemperature()' and 'dht.readHumidity()' functions for the DHT sensor.
- **Data Transmission:** Program the transmitter Arduino to send sensor data via the LoRa module. Use functions like 'LoRa.beginPacket()' and 'LoRa.print()' for sending data.
- **Receiving Data:** Program the receiver Arduino to receive this data and process it. Use functions like 'LoRa.parsePacket()' and 'LoRa.read()' for reading incoming data.
- **Autonomous Navigation:** Write code for the car to move based on ultrasonic sensor readings. If an obstacle is detected within a certain distance, the car should stop or change direction.
- **Indicator Bulbs:** Code the receiver Arduino to light up the respective bulbs when high temperature, humidity, or gas readings are detected. Use 'digitalWrite()' to control the bulbs based on sensor thresholds.

CODES :

Transmitter code :

```
#include <dht.h>
#include <SPI.h>
#include <LoRa.h>

#define DHTPIN 6
#define smokeSensor A0
#define trigPin 5
#define echoPin 4
int m1=2;
int m2=3;
int m3=7;
int m4=8;
long duration;
int distance;
dht DHT;

void setup() {
  pinMode(m1,OUTPUT);
  pinMode(m2,OUTPUT);
  pinMode(m3,OUTPUT);
  pinMode(m4,OUTPUT);
  pinMode(smokeSensor,INPUT);
  pinMode(trigPin,OUTPUT);
  pinMode(echoPin,INPUT);
  Serial.begin(9600);
  while (!Serial);

  if (!LoRa.begin(433E6)) {
    Serial.println("LoRa initialization failed. Check your connections!");
    while (1);
  }
}

void loop() {
  digitalWrite(trigPin,LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin,LOW);
  duration=pulseIn(echoPin,HIGH);
  distance=duration*0.034/2;
  if(distance>30){
    digitalWrite(m1,HIGH);
    digitalWrite(m2,HIGH);
  }else
f {
 digitalWrite(m1,LOW);
 digitalWrite(m2,LOW);
}
 int s = analogRead(smokeSensor);
 int readData = DHT.read11(DHTPIN);
loat t = DHT.temperature; // Read temperature
 float h = DHT.humidity;

 // Convert temperature and humidity to byte arrays
 byte* temperatureBytes = reinterpret_cast<byte*>(&t);
 byte* humidityBytes = reinterpret_cast<byte*>(&h);
 Serial.print("Temperature: ");
 Serial.print(t);
 Serial.print(" °C, Humidity: ");
 Serial.print(h);
 Serial.println(" %");
 Serial.print("Smoke Sensor Value ");
 Serial.println(s);


 // Transmit temperature and humidity data
 LoRa.beginPacket();
 LoRa.write(temperatureBytes, sizeof(float));
 LoRa.write(humidityBytes, sizeof(float));
 LoRa.write(s);
 LoRa.endPacket();

 delay(5000);
}
```

Receiver code :

```
#include <SPI.h>
#include <LoRa.h>

int Relay1=3;
int Relay2=4;
int Relay3=5;
float s;
void setup() {
  Serial.begin(9600);
  pinMode(Relay1,OUTPUT);
  pinMode(Relay2,OUTPUT);
  pinMode(Relay3,OUTPUT);
  while (!Serial);

  if (!LoRa.begin(433E6)) {
    Serial.println("LoRa initialization failed. Check your connections!");
    while (1);
  }
}
```

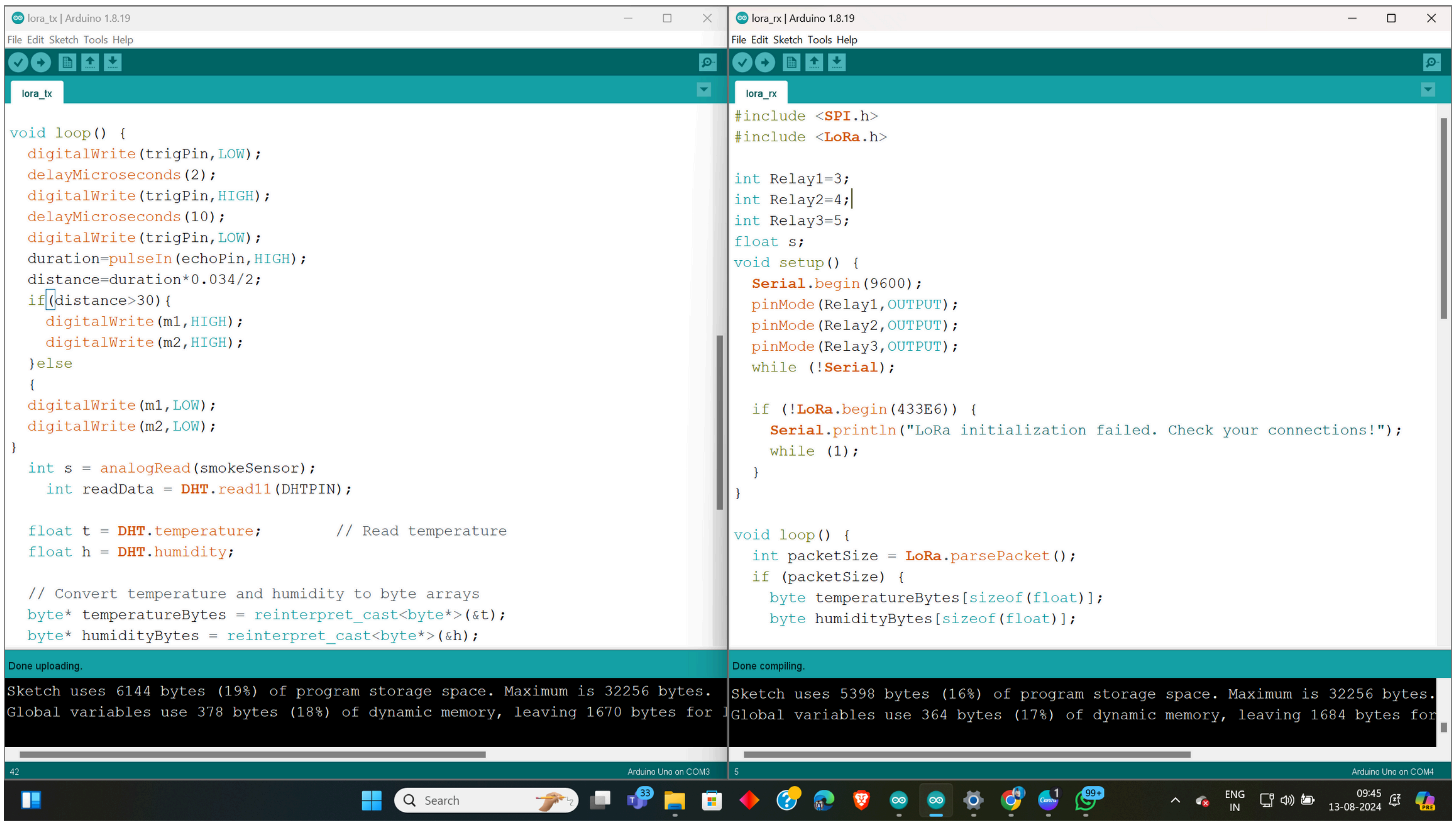
```
void loop() {
 int packetSize = LoRa.parsePacket();
 if (packetSize) {
 byte temperatureBytes[sizeof(float)];
 byte humidityBytes[sizeof(float)];

 // Read temperature and humidity data from packet
 for (int i = 0; i < sizeof(float); i++) {
 temperatureBytes[i] = LoRa.read();
 }
 for (int i = 0; i < sizeof(float); i++) {
 humidityBytes[i] = LoRa.read();
 }
 // Convert byte arrays to temperature and humidity values
 float t = *reinterpret_cast<float*>(temperatureBytes);
 float h = *reinterpret_cast<float*>(humidityBytes);
 int s = LoRa.read();
 // Print temperature and humidity values
 Serial.print("Temperature: ");
 Serial.print(t);
 Serial.print(" °C, Humidity: ");
 Serial.print(h);
 Serial.println(" %");
 Serial.print("smoke Sensor Value ");
 Serial.println(s);
 if(t<31)
 {
 digitalWrite(Relay1,HIGH);
 }
 else{
 digitalWrite(Relay1,LOW);
 }
 if(h<95){
 digitalWrite(Relay2,HIGH);
 }
 else{
 digitalWrite(Relay2,LOW);
 }
 if(s<13){
 digitalWrite(Relay3,HIGH);
 }
 else{
 digitalWrite(Relay3,LOW);
 }
 }
}
```

3. Uploading the Code :

- **Upload the Transmitter Code:** Connect the transmitter Arduino to your computer and upload the code using Arduino IDE.
- **Upload the Receiver Code:** Similarly, connect the receiver Arduino and upload the corresponding code.

4. Testing and Calibration :

- **Initial Testing:** Power up the system and ensure that the car moves and sensors are reading data correctly.
- **Range Testing:** Test the communication range of the LoRa modules to ensure it meets the 10-kilometer requirement.
- **Fine-Tuning:** Adjust the sensor sensitivity and code parameters based on initial test results for optimal performance.



```
void loop() {
  digitalWrite(trigPin,LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin,LOW);
  duration=pulseIn(echoPin,HIGH);
  distance=duration*0.034/2;
  if(distance>30){
    digitalWrite(m1,HIGH);
    digitalWrite(m2,HIGH);
  }else
  {
  digitalWrite(m1,LOW);
  digitalWrite(m2,LOW);
}
  int s = analogRead(smokeSensor);
    int readData = DHT.read11(DHTPIN);

  float t = DHT.temperature;         // Read temperature
  float h = DHT.humidity;

  // Convert temperature and humidity to byte arrays
  byte* temperatureBytes = reinterpret_cast<byte*>(&t);
  byte* humidityBytes = reinterpret_cast<byte*>(&h);
```

Done uploading.

```
Sketch uses 6144 bytes (19%) of program storage space. Maximum is 32256 bytes.
Global variables use 378 bytes (18%) of dynamic memory, leaving 1670 bytes for l
```

```
#include <SPI.h>
#include <LoRa.h>

int Relay1=3;
int Relay2=4;
int Relay3=5;
float s;
void setup() {
  Serial.begin(9600);
  pinMode(Relay1,OUTPUT);
  pinMode(Relay2,OUTPUT);
  pinMode(Relay3,OUTPUT);
  while (!Serial);

  if (!LoRa.begin(433E6)) {
    Serial.println("LoRa initialization failed. Check your connections!");
    while (1);
  }
}

void loop() {
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    byte temperatureBytes[sizeof(float)];
    byte humidityBytes[sizeof(float)];
```

Done compiling.

```
Sketch uses 5398 bytes (16%) of program storage space. Maximum is 32256 bytes.
Global variables use 364 bytes (17%) of dynamic memory, leaving 1684 bytes for
```

Conclusion :

This LoRa communication project showcases a hands-on example of applying IoT and wireless communication for autonomous navigation and environmental monitoring through illuminated three colored bulbs. By following the provided steps, you can efficiently establish and operate a system that enables long-distance data transmission and autonomous functions with Arduino and LoRa technology.