COLLEGE CODE: 3114

COLLEGE NAME: MEENAKSHI COLLEGE OF ENGINEERING

DEPARTMENT: ECE

STUDENT NM-ID: C888B5F646A4A6DACDA4FF48CFD9B048

ROLL NO: 311423106006

DATE: 08.05.2025

Completed the project named as

FLEET MANAGEMENT AND TELEMATICS - IOT

SUBMITTED BY,

NAME: D AMIRTHA

MOBILE NO: 9080457078

Phase 4: Performance of the Project

Title: IoT-Based Fleet Management and Telematics System Objective:

The focus of Phase 4 is to enhance the performance of the IoT-based fleet management system by refining its data analytics capabilities, optimizing system scalability, and ensuring the infrastructure can handle an increased number of connected vehicles. This phase also aims to improve real-time tracking accuracy, optimize communication with IoT devices (vehicles), and reinforce data security, while laying the foundation for multilingual user support in the management dashboard.

1. Telematics Analytics Enhancement

Overview:

The system's data analytics model for vehicle diagnostics, route optimization, and driver behavior analysis will be refined using feedback and historical data from earlier phases.

Performance Improvements:

- Data Accuracy: Models will be retrained with larger, more diverse datasets to handle various vehicle types, environmental conditions, and operational contexts.
- Model Optimization: Techniques like feature selection and parameter tuning will be applied to increase analysis speed and accuracy.

Outcome:

By the end of Phase 4, the system will deliver more precise insights on vehicle performance, driver behavior, and predictive maintenance, with reduced false alarms and improved route suggestions.

2. Dashboard and User Interface Optimization

Overview:

The fleet management dashboard will be optimized for faster response times, better user interaction, and more intuitive controls. This includes handling larger fleets and multilingual support preparation.

Key Enhancements:

- Performance Tuning: Dashboard responsiveness will be improved to support large datasets and realtime updates from hundreds of vehicles.
- User Experience: The interface will support regional language inputs and visual cues for global operations.

Outcome:

Fleet operators will experience a smoother UI, reduced data loading time, and enhanced usability, especially under peak traffic conditions.

---

3. IoT Device Integration Performance

Overview:

This phase focuses on optimizing the integration of telematics hardware (OBD devices, GPS trackers, engine sensors) across various vehicle models to ensure low-latency data collection.

Key Enhancements:

- Real-Time Data Flow: System latency in receiving GPS, speed, fuel consumption, and engine diagnostics will be minimized.
- API Optimization: Improved communication protocols and API calls to onboard units and third-party services (like traffic APIs) will ensure reliable data flow.

Outcome:

The platform will process real-time telematics data with minimal delay, enabling effective vehicle monitoring and faster alert generation for events such as speeding, harsh braking, or maintenance needs.

---

4. Data Security and Privacy Performance

Overview:

Security measures from earlier phases will be hardened to ensure protection of location and operational data under high load conditions.

Key Enhancements:

- Enhanced Encryption: Deployment of end-to-end encryption (e.g., TLS 1.3, AES-256) across all IoT communication channels.
- Security Audits: Penetration testing and stress testing to detect vulnerabilities in data handling and storage.

Outcome:

The system will maintain strong security standards, protecting sensitive fleet and route data even under increased usage, complying with regulatory standards (e.g., GDPR, ISO 27001).

---

5. Performance Testing and Metrics Collection

Overview:

Comprehensive testing will validate the system's readiness for large-scale deployments, with a focus on high availability and fault tolerance.

Implementation:

- Load Testing: Simulate large fleets with concurrent data streams to test system responsiveness and uptime.
- Metrics Monitoring: Key metrics such as data refresh rate, dashboard load times, and processing throughput will be tracked.
- User Feedback: Gather insights from fleet managers and drivers to refine usability and reliability.

Outcome:

The system will reliably scale to support fleets of various sizes with stable performance, and be capable of sustaining high-frequency data collection without performance degradation.

Key Challenges in Phase 4

1. Scalability Across Fleets o Challenge: Supporting large fleets with varying hardware and geographic spread.

- o Solution: Modular architecture, cloud-based scaling, and real-time streaming technologies (e.g., MQTT, Kafka).

2. Secure Data Transmission o Challenge: Protecting in-transit vehicle data from interception or tampering.
   - o Solution: Use secure transport protocols and regular penetration testing.
3. Hardware Variability o Challenge: Compatibility with multiple IoT devices and vehicle types.
   - o Solution: Implement adaptive drivers and standard communication formats (e.g., CAN bus, OBD-II).

Outcomes of Phase 4

1. Improved Analytics Accuracy:
   The system provides more reliable predictions for vehicle health and routing efficiency.
2. Enhanced User Interface:
   The dashboard is responsive and multilingual-ready, with faster load times.
3. Real-Time IoT Data Integration:
   Telematics data from onboard devices is processed with minimal latency for real-time decision-making.
4. Stronger Data Security:
   Fleet and location data are securely handled, even under high-volume scenarios.

Next Steps for Finalization

In the final phase, the platform will undergo full deployment with a pilot fleet, during which additional feedback will be collected to refine AI analytics, improve dashboard usability, and finalize multilingual support before public release.

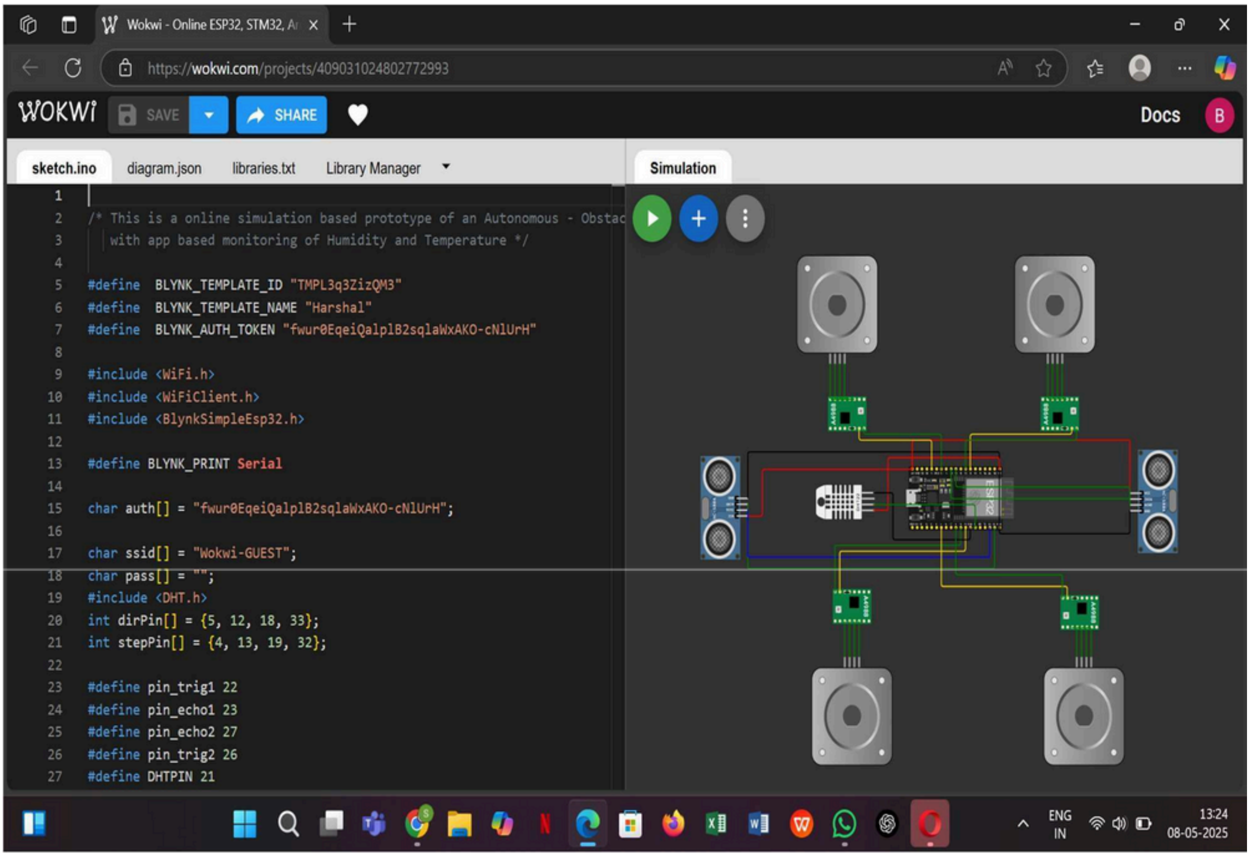
Sample Code for Phase 4 (Telematics Data Ingestion)

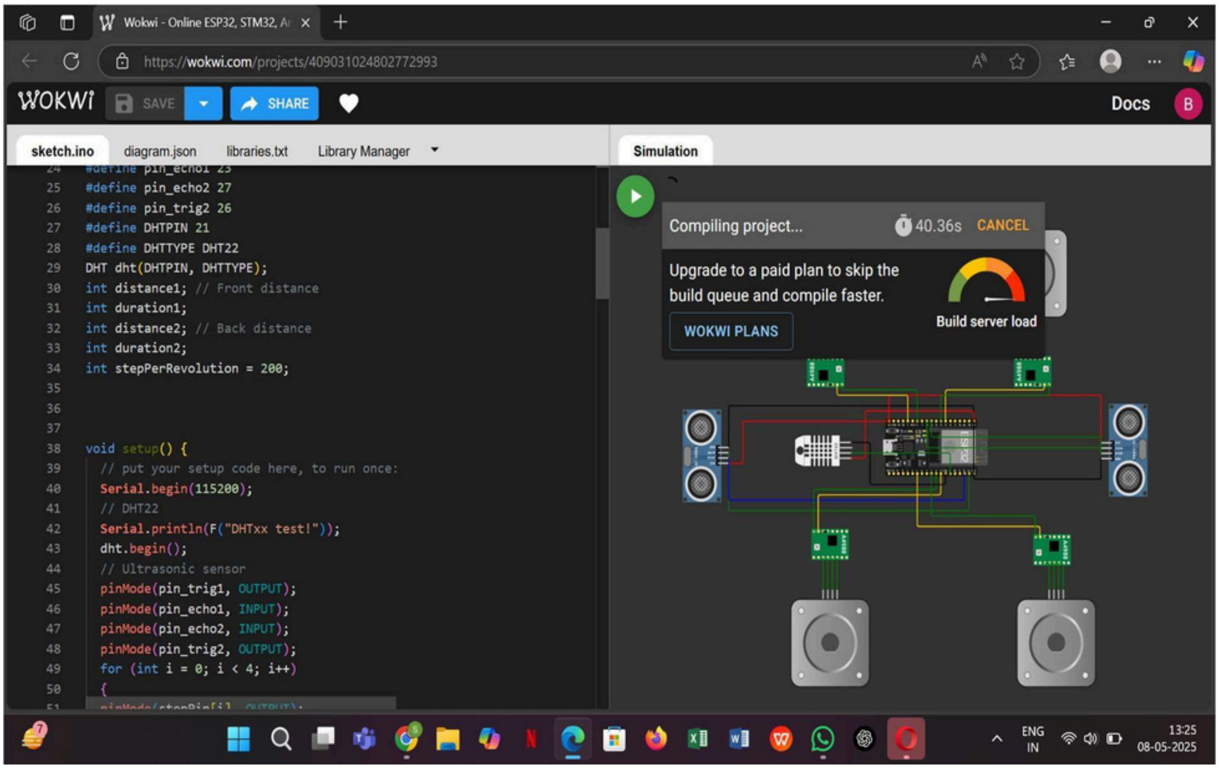
```
import paho.mqtt.client as mqtt
 def on_message(client, userdata, msg):     vehicle_id =
msg.topic.split("/")[-1]      data = json.loads(msg.payload)
    process_telematics_data(vehicle_id, data)


client = mqtt.Client() client.on_message =
on_message
client.connect("broker.example.com", 1883, 60) client.subscribe("fleet/+/telematics")
client.loop_forever()
```

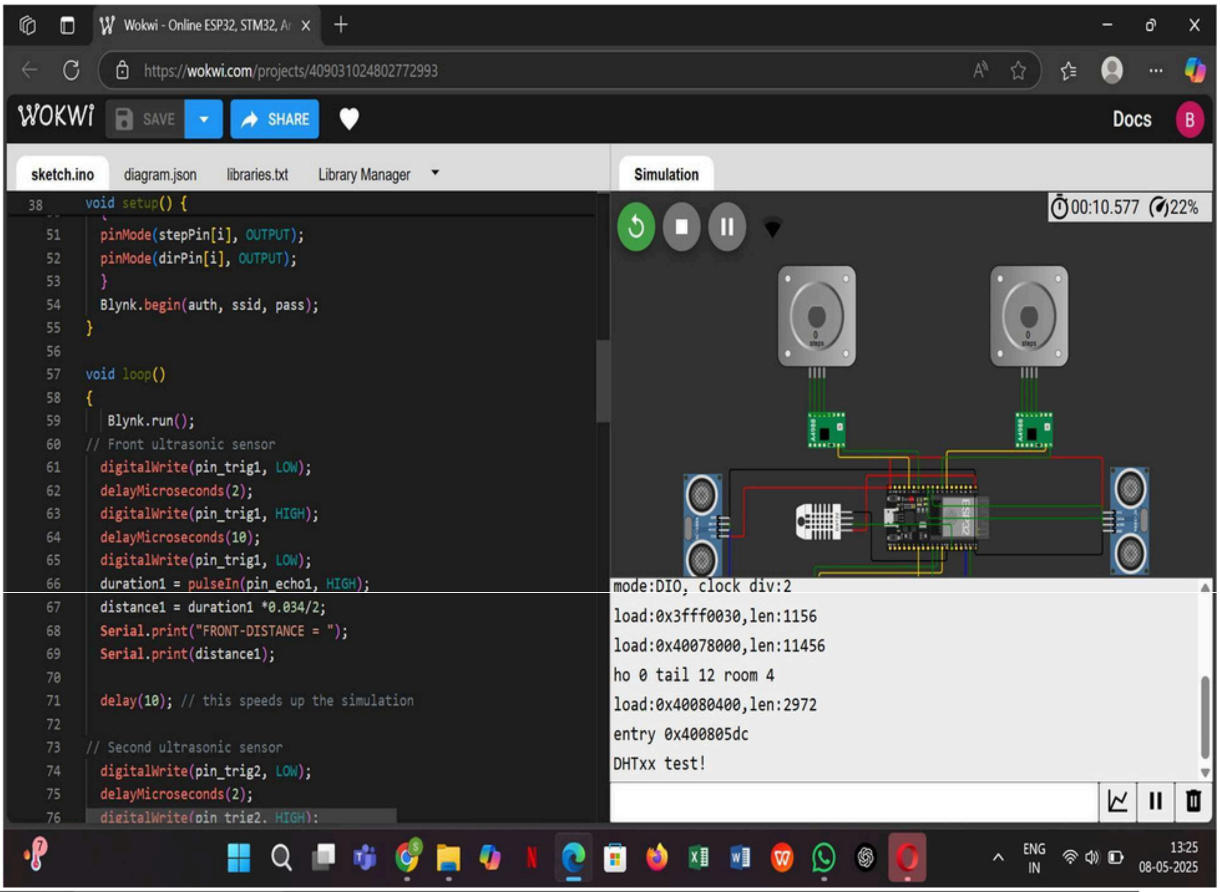
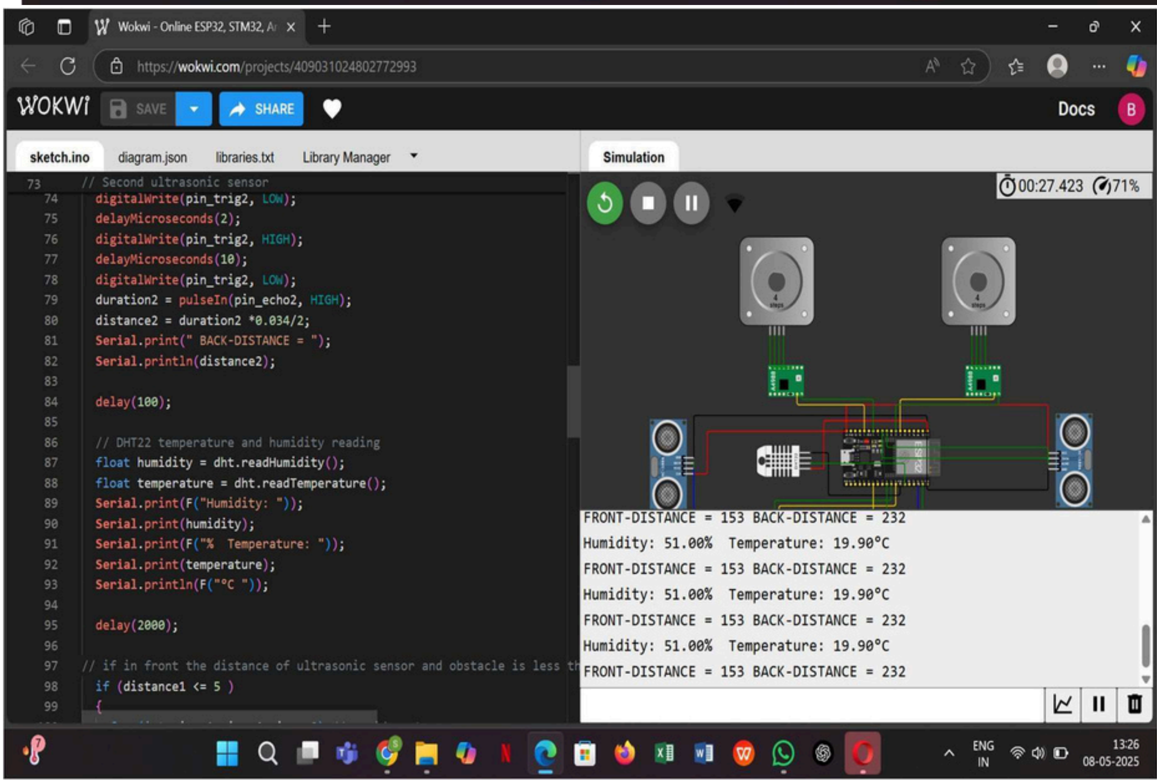
Performance Metrics Screenshot Placeholder

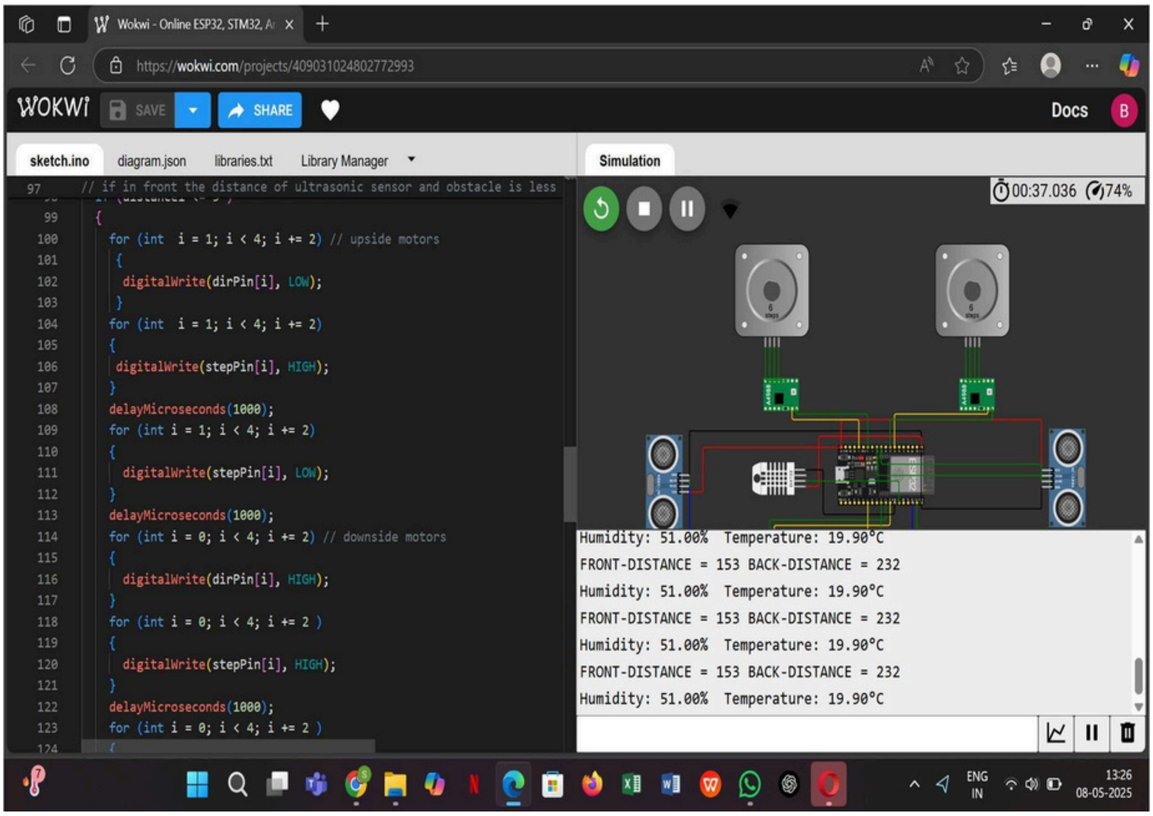
Include:

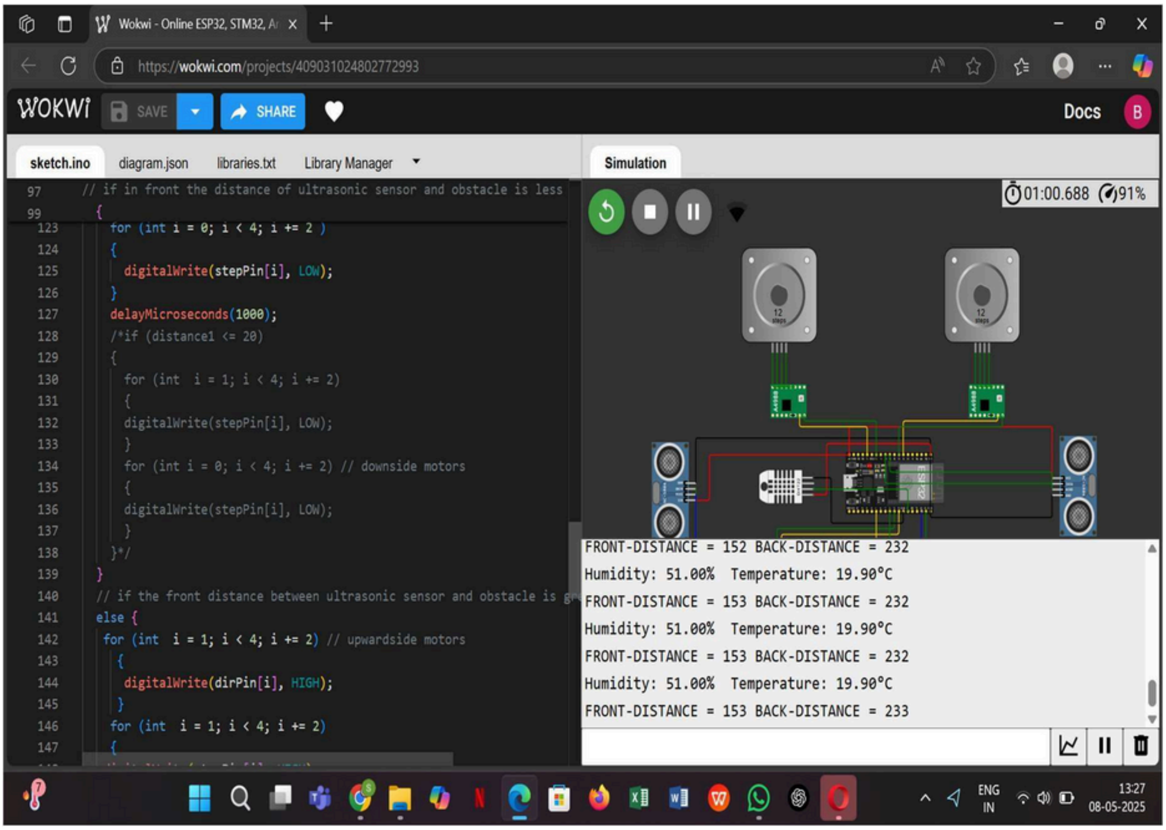
- Before & after latency graphs for data ingestion
- CPU/memory usage under load
- Dashboard load time comparison
- Map visualization showing real-time vehicle positions

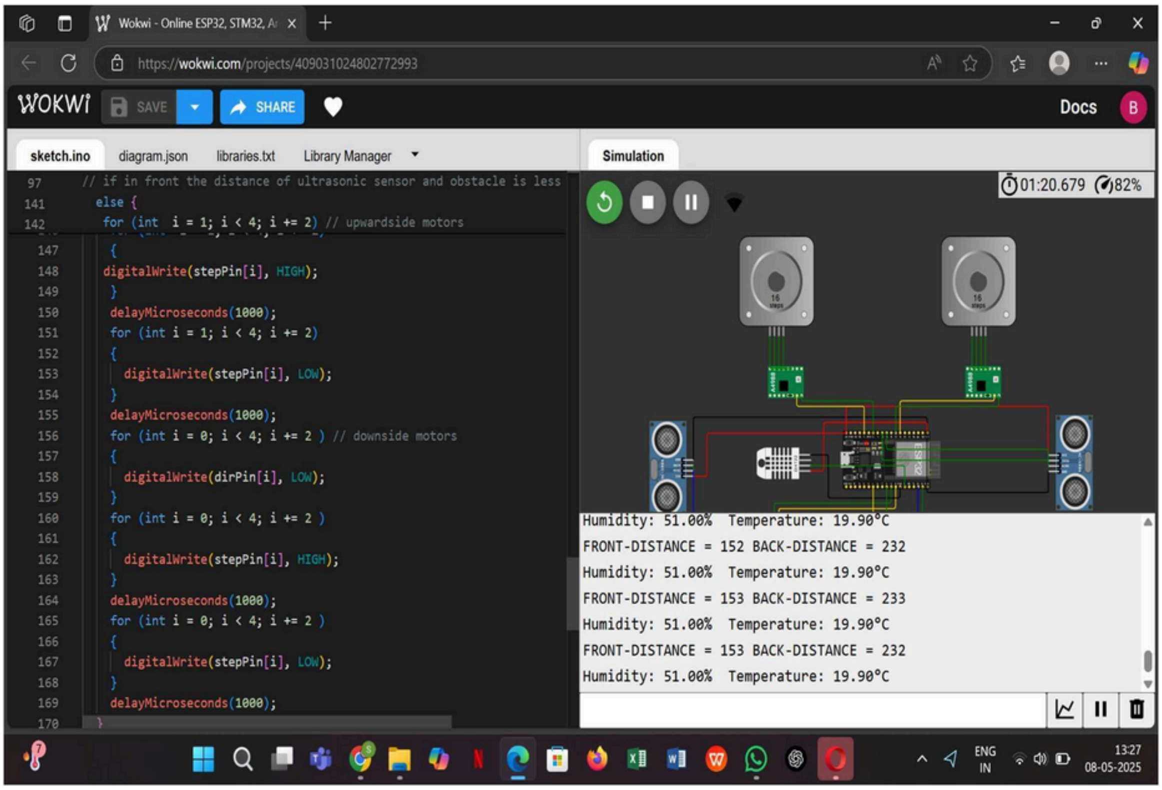


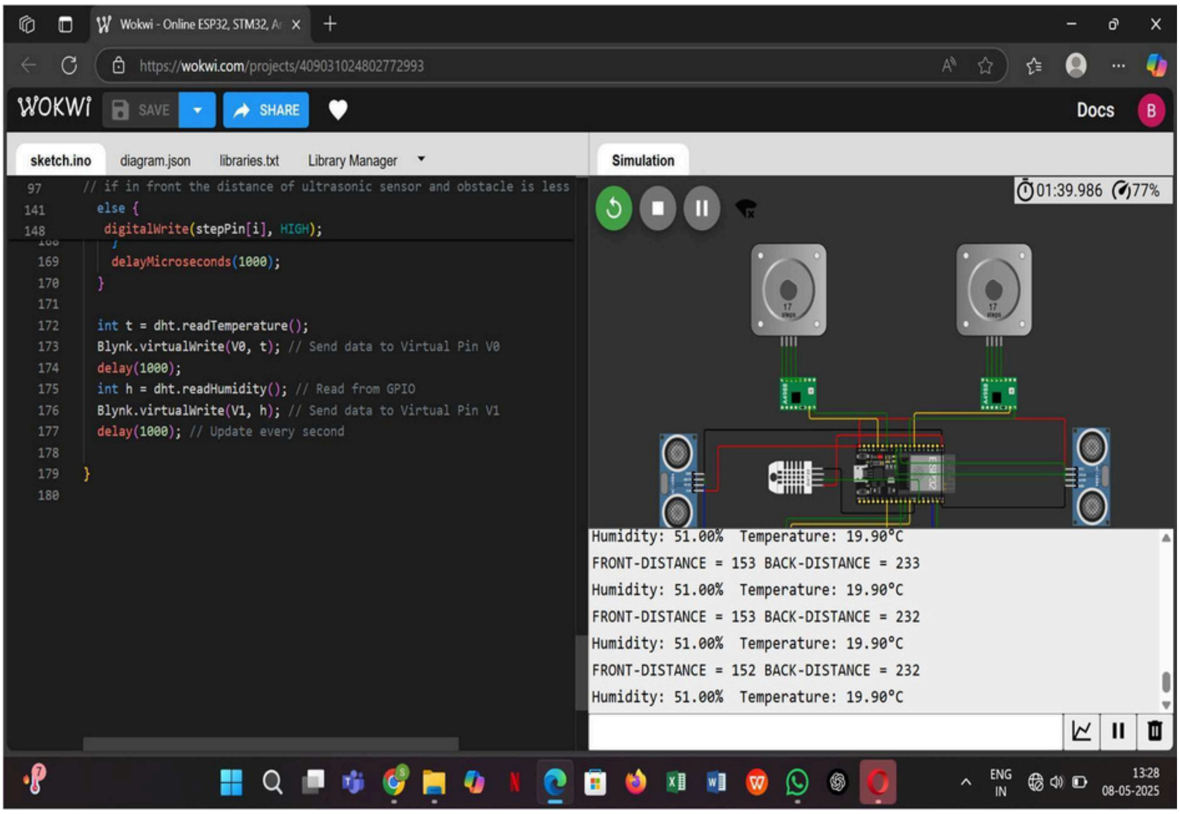

Wokwi - Online ESP32, STM32, Ar
https://wokwi.com/projects/409031024802772993
WOKWI
SAVE
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
#define pin_echo2 27
#define pin_trig2 26
#define DHTPIN 21
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
int distance1; // Front distance
int duration1;
int distance2; // Back distance
int duration2;
int stepPerRevolution = 200;
void setup() {
// put your setup code here, to run once:
Serial.begin(115200);
// DHT22
Serial.println(F("DHTxx test!"));
dht.begin();
// Ultrasonic sensor
pinMode(pin_trig1, OUTPUT);
pinMode(pin_echo1, INPUT);
pinMode(pin_echo2, INPUT);
pinMode(pin_trig2, OUTPUT);
for (int i = 0; i < 4; i++)
{
Compiling project...
40.36s
CANCEL
Upgrade to a paid plan to skip the
build queue and compile faster.
WOKWI PLANS
Build server load
ENG
IN
13:25
08-05-2025


Wokwi - Online ESP32, STM32, Ar
https://wokwi.com/projects/409031024802772993
WOKWi
SAVE
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
void setup() {
pinMode(stepPin[i], OUTPUT);
pinMode(dirPin[i], OUTPUT);
}
Blynk.begin(auth, ssid, pass);
}
void loop()
{
Blynk.run();
// Front ultrasonic sensor
digitalWrite(pin_trig1, LOW);
delayMicroseconds(2);
digitalWrite(pin_trig1, HIGH);
delayMicroseconds(10);
digitalWrite(pin_trig1, LOW);
duration1 = pulseIn(pin_echo1, HIGH);
distance1 = duration1 *0.034/2;
Serial.print("FRONT-DISTANCE = ");
Serial.print(distance1);
delay(10); // this speeds up the simulation
// Second ultrasonic sensor
digitalWrite(pin_trig2, LOW);
delayMicroseconds(2);
digitalWrite(pin_trig2, HIGH);
Simulation
00:10.577
22%
mode:DIO, clock div:2
load:0x3fff0030,len:1156
load:0x40078000,len:11456
ho 0 tail 12 room 4
load:0x40080400,len:2972
entry 0x400805dc
DHTxx test!
ENG
IN
13:25
08-05-2025

Wokwi - Online ESP32, STM32, Ar
https://wokwi.com/projects/409031024802772993
WOKWi
SAVE
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
// Second ultrasonic sensor
digitalWrite(pin_trig2, LOW);
delayMicroseconds(2);
digitalWrite(pin_trig2, HIGH);
delayMicroseconds(10);
digitalWrite(pin_trig2, LOW);
duration2 = pulseIn(pin_echo2, HIGH);
distance2 = duration2 *0.034/2;
Serial.print(" BACK-DISTANCE = ");
Serial.println(distance2);
delay(100);
// DHT22 temperature and humidity reading
float humidity = dht.readHumidity();
float temperature = dht.readTemperature();
Serial.print(F("Humidity: "));
Serial.print(humidity);
Serial.print(F("% Temperature: "));
Serial.print(temperature);
Serial.println(F("°C "));
delay(2000);
// if in front the distance of ultrasonic sensor and obstacle is less
if (distance1 <= 5)
{
Simulation
00:27.423
71%
FRONT-DISTANCE = 153 BACK-DISTANCE = 232
Humidity: 51.00% Temperature: 19.90°C
FRONT-DISTANCE = 153 BACK-DISTANCE = 232
Humidity: 51.00% Temperature: 19.90°C
FRONT-DISTANCE = 153 BACK-DISTANCE = 232
Humidity: 51.00% Temperature: 19.90°C
FRONT-DISTANCE = 153 BACK-DISTANCE = 232
ENG
IN
13:26
08-05-2025


Wokwi - Online ESP32, STM32, Ar
https://wokwi.com/projects/409031024802772993
WOKWI
SAVE
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:37.036
74%
// if in front the distance of ultrasonic sensor and obstacle is less
{
for (int i = 1; i < 4; i += 2) // upside motors
{
digitalWrite(dirPin[i], LOW);
}
for (int i = 1; i < 4; i += 2)
{
digitalWrite(stepPin[i], HIGH);
}
delayMicroseconds(1000);
for (int i = 1; i < 4; i += 2)
{
digitalWrite(stepPin[i], LOW);
}
delayMicroseconds(1000);
for (int i = 0; i < 4; i += 2) // downside motors
{
digitalWrite(dirPin[i], HIGH);
}
for (int i = 0; i < 4; i += 2)
{
digitalWrite(stepPin[i], HIGH);
}
delayMicroseconds(1000);
for (int i = 0; i < 4; i += 2)
Humidity: 51.00% Temperature: 19.90°C
FRONT-DISTANCE = 153 BACK-DISTANCE = 232
Humidity: 51.00% Temperature: 19.90°C
FRONT-DISTANCE = 153 BACK-DISTANCE = 232
Humidity: 51.00% Temperature: 19.90°C
FRONT-DISTANCE = 153 BACK-DISTANCE = 232
Humidity: 51.00% Temperature: 19.90°C
ENG
IN
13:26
08-05-2025


Wokwi - Online ESP32, STM32, Ar
https://wokwi.com/projects/409031024802772993
WOKWi
SAVE
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
01:00.688
91%
// if in front the distance of ultrasonic sensor and obstacle is less
{
for (int i = 0; i < 4; i += 2)
{
digitalWrite(stepPin[i], LOW);
}
delayMicroseconds(1000);
/*if (distance1 <= 20)
{
for (int i = 1; i < 4; i += 2)
{
digitalWrite(stepPin[i], LOW);
}
for (int i = 0; i < 4; i += 2) // downside motors
{
digitalWrite(stepPin[i], LOW);
}
}*/
}
// if the front distance between ultrasonic sensor and obstacle is g
else {
for (int i = 1; i < 4; i += 2) // upwardside motors
{
digitalWrite(dirPin[i], HIGH);
}
for (int i = 1; i < 4; i += 2)
{
FRONT-DISTANCE = 152 BACK-DISTANCE = 232
Humidity: 51.00% Temperature: 19.90°C
FRONT-DISTANCE = 153 BACK-DISTANCE = 232
Humidity: 51.00% Temperature: 19.90°C
FRONT-DISTANCE = 153 BACK-DISTANCE = 232
Humidity: 51.00% Temperature: 19.90°C
FRONT-DISTANCE = 153 BACK-DISTANCE = 233
ENG
IN
13:27
08-05-2025


Wokwi - Online ESP32, STM32, Ar
https://wokwi.com/projects/409031024802772993
WOKWi
SAVE
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
01:20.679
82%
// if in front the distance of ultrasonic sensor and obstacle is less
else {
for (int i = 1; i < 4; i += 2) // upwardside motors
{
digitalWrite(stepPin[i], HIGH);
}
delayMicroseconds(1000);
for (int i = 1; i < 4; i += 2)
{
digitalWrite(stepPin[i], LOW);
}
delayMicroseconds(1000);
for (int i = 0; i < 4; i += 2) // downside motors
{
digitalWrite(dirPin[i], LOW);
}
for (int i = 0; i < 4; i += 2)
{
digitalWrite(stepPin[i], HIGH);
}
delayMicroseconds(1000);
for (int i = 0; i < 4; i += 2)
{
digitalWrite(stepPin[i], LOW);
}
delayMicroseconds(1000);
Humidity: 51.00% Temperature: 19.90°C
FRONT-DISTANCE = 152 BACK-DISTANCE = 232
Humidity: 51.00% Temperature: 19.90°C
FRONT-DISTANCE = 153 BACK-DISTANCE = 233
Humidity: 51.00% Temperature: 19.90°C
FRONT-DISTANCE = 153 BACK-DISTANCE = 232
Humidity: 51.00% Temperature: 19.90°C
ENG
IN
13:27
08-05-2025


Wokwi - Online ESP32, STM32, A
https://wokwi.com/projects/409031024802772993
WOKWi
SAVE
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
// if in front the distance of ultrasonic sensor and obstacle is less
else {
digitalWrite(stepPin[i], HIGH);
delayMicroseconds(1000);
}
int t = dht.readTemperature();
Blynk.virtualWrite(V0, t); // Send data to Virtual Pin V0
delay(1000);
int h = dht.readHumidity(); // Read from GPIO
Blynk.virtualWrite(V1, h); // Send data to Virtual Pin V1
delay(1000); // Update every second
}
01:39.986
77%
Humidity: 51.00% Temperature: 19.90°C
FRONT-DISTANCE = 153 BACK-DISTANCE = 233
Humidity: 51.00% Temperature: 19.90°C
FRONT-DISTANCE = 153 BACK-DISTANCE = 232
Humidity: 51.00% Temperature: 19.90°C
FRONT-DISTANCE = 152 BACK-DISTANCE = 232
Humidity: 51.00% Temperature: 19.90°C
ENG
IN
13:28
08-05-2025