PROJECT REPORT ON

Smart Industrial Machine Monitoring using IoT and CAN

Submitted To Sunbeam Pune in Partial Fulfilment of the requirement for the Award of

Post Graduation Diploma in Embedded System Design

(PG-DESD)

Submitted by :

Himanshu Dahake

Piyush Sharma

Rutuja Mahajan

Prasanna Gulvani

CERTIFICATE

This is to certify that this is a Bonafide record of the project work done by Mr. Piyush Sharma, Miss. Rutuja Mahajan, Mr. Prasanna Gulvani, and Mr. Himanshu Dahake under the guidance of Ms. Utkarsha Nikam , in partial fulfillment of the requirements for the award of the **Diploma in Embedded System Design** during the academic session of August 2025 .The project entitled "SMART INDUSTRIAL MACHINE MONITORING USING IoT AND CAN" has been carried out satisfactorily as per the curriculum prescribed by the institute.

Project Guide:-

Name: Ms. Utkarsha Nikam

ACKNOWLEDGEMENT

We have taken sincere efforts in completing this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to Sunbeam for their guidance and constant supervision, as well as for providing necessary information regarding the project. Their support and encouragement were instrumental in the successful completion of this project.

We would like to express our heartfelt gratitude towards our parents and the members of the DESD department for their kind cooperation and encouragement, which greatly helped us in completing this project successfully.

We would also like to express our special gratitude and thanks to industry professionals for giving us their valuable time, attention, and guidance.

Our sincere thanks and appreciations also go to our colleagues and friends for their cooperation in developing the project and to all those who willingly helped us with their abilities and support.

ABSTRACT

This project presents the design and implementation of a Smart Industry Monitoring and Controlling System based on Internet of Things (IoT) and Controller Area Network (CAN) communication, aimed at replicating a real-world industrial monitoring architecture. The system continuously monitors critical industrial parameters such as temperature, humidity, gas concentration, and vibration using distributed sensor nodes deployed across the industrial environment.

A sensor node built around the STM32 Nucleo-F446 microcontroller collects data from multiple sensors and transmits it reliably to a central gateway using CAN bus communication, ensuring noise-resistant and deterministic data transfer suitable for industrial conditions. The gateway controller (STM32F407) processes the received CAN messages and forwards the sensor data to an ESP32-C3 module via UART communication. The ESP32-C3 enables wireless connectivity and uploads the processed data to the cloud for remote access.

An IoT dashboard is developed to visualize real-time sensor readings, providing continuous monitoring and generating alerts whenever any parameter exceeds predefined safety thresholds. This allows timely detection of abnormal conditions and enhances industrial safety and operational efficiency. The proposed system demonstrates a scalable, reliable, and industry-oriented embedded solution, integrating multiple controllers, sensors, and communication protocols.

INDEX

1. INTRODUCTION

With the rapid growth of embedded systems and Internet of Things (IoT) technologies, monitoring and control applications have become increasingly complex and distributed in nature. Modern monitoring systems often involve multiple sensors, microcontrollers, and communication protocols working together to ensure reliable data acquisition and transmission. While theoretical knowledge of microcontrollers, communication interfaces, and IoT platforms is essential, it is not sufficient on its own. Practical implementation plays a crucial role in understanding real-world challenges such as system integration, data reliability, synchronization, and scalability.

During the CDAC course, core concepts related to embedded systems were studied, including microcontroller architecture, peripheral interfacing, communication protocols such as UART, SPI, I²C, and CAN, as well as IoT fundamentals and cloud connectivity. This project was undertaken to bridge the gap between theoretical understanding and practical application by designing and implementing a complete, working multi-node monitoring system.

This project focuses on the development of an IoT-based multi-sensor monitoring system using STM32 microcontrollers, Controller Area Network (CAN) communication, and ESP32-based cloud connectivity. The system is designed to monitor multiple environmental and motion-related parameters such as temperature, humidity, gas or smoke concentration, and motion or vibration. Various sensors are interfaced with STM32-based sensor nodes, which continuously acquire real-time data from the environment.

To ensure reliable and noise-resistant communication between multiple sensor nodes and the central gateway, CAN communication is employed. CAN is widely used in industrial and automotive applications due to its robustness, error detection mechanisms, and ability to support multi-node communication over a shared bus. The collected sensor data is transmitted over the CAN network to a gateway controller, where it is further processed and forwarded to an ESP32 module using serial communication. The ESP32 connects to an IoT cloud platform, enabling real-time data visualization and remote monitoring through a cloud dashboard.

The primary objective of this project is to understand system integration and communication flow in a real-time embedded system rather than to design a new product or develop a novel algorithm. Emphasis is placed on understanding how different hardware components, communication protocols, and software modules interact to form a complete end-to-end system. The project also highlights practical considerations such as data synchronization, message prioritization in CAN communication, and reliable data transmission to the cloud

2. Objective and Purpose

2.1 Objective

The objective of this project is to design and implement an IoT-based multi-sensor monitoring system using STM32 microcontrollers and CAN communication. The system collects data from MPU6050, MQ2 gas sensor, and DHT11 temperature-humidity sensor, transmits it from a sensor node to a gateway node using CAN, and displays the data on a cloud platform using ESP32.

2.2 Purpose

The purpose of this project is to apply the concepts learned during the CDAC program in a real-world embedded system setup. Through this project, I aimed to:

Understand multi-sensor interfacing with microcontrollers

Implement CAN communication between two STM32 boards

Use ESP32 for cloud connectivity and real-time monitoring

Build a complete system from sensor data acquisition to cloud visualization

Overall, this project successfully bridged the gap between theoretical learning and practical implementation by providing hands-on experience in embedded system design, communication protocols, and IoT integration. The knowledge gained through this project strengthens the understanding of real-world embedded architectures and prepares a solid foundation for advanced applications in industrial monitoring and automation.

3. Problem Statement

Monitoring systems that involve multiple parameters such as environmental and motion-related data require not only accurate sensing but also reliable communication and a well-structured system architecture. In real-world applications, especially in industrial and distributed environments, simple point-to-point communication or direct Wi-Fi–based monitoring systems are often insufficient. Such approaches may suffer from limitations related to communication reliability, network congestion, scalability, and fault tolerance when multiple nodes are involved.

To address these challenges, modern embedded monitoring systems adopt layered and modular architectures, where sensing, data processing, communication, and cloud interaction are handled by dedicated components. This separation of functionality improves system reliability, simplifies expansion, and enables easier debugging and maintenance. Understanding this architectural approach is essential for designing scalable and industrial-grade embedded systems.

The primary goal of this project is not to propose or solve a new problem, but rather to implement and study a structured embedded system architecture that closely resembles real industrial monitoring solutions. The system is designed around four key building blocks:

1. Dedicated Sensor Node – Responsible for interfacing with multiple sensors and acquiring real-time data such as temperature, humidity, gas or smoke levels, and motion or vibration.
2. Gateway Controller – Acts as an intermediate processing and aggregation unit, receiving data from multiple sensor nodes and preparing it for external communication.
3. Robust Communication Protocol (CAN) – Used to ensure reliable, noise-resistant, and prioritized data transfer between distributed nodes, making the system suitable for harsh and industrial environments.
4. Cloud-Based Monitoring using IoT – Enables real-time visualization, remote monitoring, and data logging through an IoT cloud platform.

This project focuses on understanding how data flows across multiple hardware and software layers in a practical monitoring system.

4.Scope of Project

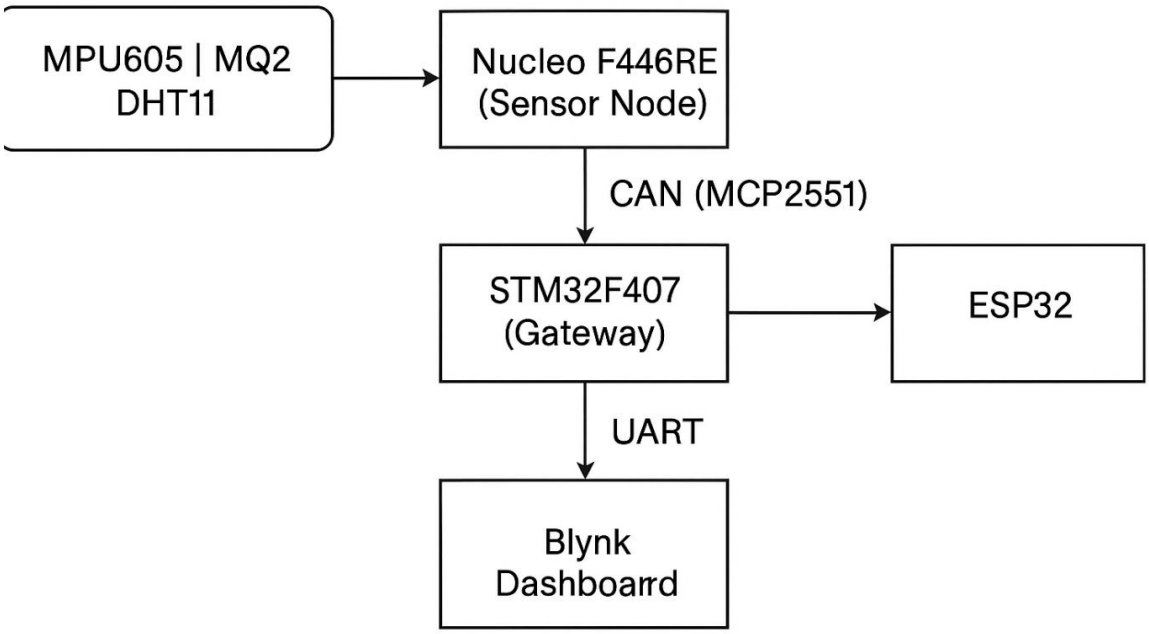
The scope of this project includes:

1. Real-time monitoring of temperature, humidity, gas/smoke, and motion data
2. CAN-based communication between sensor node and gateway
3. UART-based communication between gateway and ESP32
4. Cloud-based visualization using the Blynk platform
5. Alert generation when sensor readings exceed predefined thresholds

The project emphasizes functional correctness, communication reliability, and system integration, rather than sensor accuracy or industrial deployment.

5. System Architecture

The proposed system is organized into four main functional blocks: Sensor Node, CAN Communication Bus, Gateway Node, and Cloud Interface. Each block plays a critical role in ensuring reliable data acquisition, transmission, and visualization.



System Architecture

1. Sensor Node

- Components Used:
    1. MPU6050 – 6-axis accelerometer and gyroscope for motion and orientation sensing.
    2. MQ2 – Gas sensor for detecting combustible gases such as LPG, methane, and smoke.
    3. DHT11 – Temperature and humidity sensor for environmental monitoring.
- Controller:
    1. Nucleo F446RE (STM32 series) acts as the sensor node controller. It collects raw data from the connected sensors, performs initial preprocessing (e.g., filtering, scaling), and prepares the data for transmission.

- Purpose:

1. This block is responsible for data acquisition. It ensures that physical parameters (motion, gas concentration, temperature, humidity) are continuously monitored and digitized for further processing.

2. CAN Communication Bus

- Interface:
    1. MCP2551 CAN Transceiver is used to establish communication between the sensor node and the gateway node.
- Purpose:
    1. The Controller Area Network (CAN) bus provides a robust, fault-tolerant communication protocol suitable for real-time applications.
    2. It ensures reliable data transfer even in noisy environments, making it ideal for industrial and automotive-grade systems.
- Role in System:

1. Acts as the communication backbone between distributed sensor nodes and the central gateway.

3. Gateway Node

- Controller:
    1. STM32F407 microcontroller is used as the gateway node.
- Functions:
    1. Receives sensor data from the CAN bus.
    2. Performs additional processing, aggregation, or formatting of data.
    3. Converts CAN data into UART-compatible format for transmission to the cloud interface.
- Purpose:
- The gateway node serves as a protocol translator and data manager, bridging the CAN network with the cloud interface.

4. Cloud Interface

- Hardware:
    1. ESP32 microcontroller with built-in Wi-Fi capability.
- Functions:

1. Receives data from the gateway node via UART.
2. Establishes a wireless connection to the internet.
3. Uploads sensor data to the Blynk Dashboard (IoT platform).

- Purpose:
- Provides remote monitoring and visualization of sensor data.
- Enables users to view real-time readings (motion, gas levels, temperature, humidity) through a mobile or web dashboard.
- Supports alerts, notifications, and data logging for analysis.

5. Data Flow Summary

1. Sensors (MPU6050, MQ2, DHT11) → Data captured.
2. Nucleo F446RE (Sensor Node) → Preprocesses and sends data via CAN.
3. CAN Bus (MCP2551) → Ensures reliable communication.
4. STM32F407 (Gateway Node) → Aggregates and forwards data via UART.
5. ESP32 (Cloud Interface) → Uploads data to Blynk Dashboard for visualization.

6. Hardware Components Description

STM32F407 (Gateway Node)

- Function: Acts as the central gateway between the sensor node and the cloud interface.
- Role in System:
    1. Receives sensor data via the CAN bus.
    2. Converts and forwards the data to the ESP32 using UART communication.
- Features:

1. High-performance ARM Cortex-M4 processor.
2. Supports multiple communication protocols including CAN and UART.

Nucleo F446RE (Sensor Node)

- Function: Serves as the primary data acquisition unit.
- Role in System:
    1. Interfaces with three sensors: MPU6050, MQ2, and DHT11.
    2. Collects and preprocesses sensor data.
    3. Transmits data over the CAN bus using MCP2551 transceiver.
- Features:
- STM32F446RE microcontroller with flexible I/O and communication support.
- Compatible with Arduino and ST morpho connectors.

MPU6050 (Motion Sensor)

- Function: Detects motion and vibration.
- Role in System:
    1. Provides 3-axis accelerometer and 3-axis gyroscope data.
    2. Communicates with the Nucleo board via I2C protocol.
- Applications:

1. Useful for detecting movement, tilt, and vibration patterns.

MQ2 Gas Sensor

- Function: Detects the presence of gases and smoke.
- Role in System:
    1. Outputs analog signals proportional to gas concentration.
    2. Connected to the Nucleo board for monitoring air quality.
- Applications:
- Suitable for detecting LPG, methane, smoke, and other combustible gases.

DHT11 (Temperature & Humidity Sensor)

- Function: Measures ambient temperature and humidity.
- Role in System:
    1. Communicates with the Nucleo board using a single-wire digital protocol.
    2. Provides periodic environmental readings.
- Applications:

1. Ideal for basic climate monitoring in indoor environments.

ESP32 (Cloud Interface)

- Function: Enables wireless data transmission to the cloud.
- Role in System:
    1. Receives data from STM32F407 via UART.
    2. Connects to Wi-Fi and uploads data to the Blynk Dashboard.
- Features:

1. Dual-core processor with integrated Wi-Fi and Bluetooth.
2. Supports IoT platforms and cloud APIs.

MCP2551 (CAN Transceiver)

- Function: Facilitates CAN communication between microcontrollers.

- Role in System**:**
    1. Converts digital CAN signals from the Nucleo board into differential signals for transmission.
    2. Ensures reliable and noise-resistant communication over the CAN bus.
- Features:

1. High-speed CAN transceiver compliant with ISO 11898 standard.

.

7. Software Components Description

STM32CubeIDE

- Purpose:
    1. Integrated Development Environment (IDE) provided by STMicroelectronics for STM32 microcontrollers.
- Role in System:
    1. Used to program and debug the Nucleo F446RE (Sensor Node) and STM32F407 (Gateway Node).
    2. Provides project management, code editing, compilation, and debugging features.
    3. Supports integration of HAL (Hardware Abstraction Layer) libraries for peripheral handling.
- Advantages:

1. Simplifies embedded development with graphical configuration tools.
2. Offers direct support for CAN communication and other STM32 peripherals.

Arduino IDE

- Purpose:
    1. Open-source IDE widely used for programming microcontrollers, especially ESP32.
- Role in System:
    1. Used to program the ESP32 (Cloud Interface).
    2. Provides libraries and examples for Wi-Fi connectivity and IoT integration.
    3. Enables UART communication with STM32F407 and cloud data transfer.
- Advantages:

1. User-friendly interface with extensive community support.
2. Simplifies integration with IoT platforms like Blynk.

Blynk Platform

- Purpose:
    1. IoT platform for cloud-based visualization, monitoring, and control.
- Role in System:
    1. Receives sensor data from ESP32 via Wi-Fi.
    2. Provides a dashboard for real-time visualization of motion, gas concentration, temperature, and humidity.
    3. Supports alerts, notifications, and data logging.
- Advantages:

1. Mobile and web-based dashboards for easy access.
2. Enables remote monitoring and control without complex server setup.

CAN Drivers & HAL Libraries

- Purpose:
    1. Software libraries that handle low-level communication between microcontrollers and peripherals.
- Role in System:
    1. CAN Drivers: Manage data transmission and reception over the CAN bus using MCP2551 transceiver.
    2. HAL Libraries: Provide abstraction for STM32 peripherals (I2C for MPU6050, ADC for MQ2, single-wire protocol for DHT11, UART for ESP32).
- Advantages:
- Simplifies hardware interaction by providing high-level APIs.
- Ensures portability and scalability across STM32 boards.

8. Implementation Flow

8.1 Interfacing MPU6050, MQ2, and DHT11 with STM32F407

The first stage of implementation involved interfacing all sensors with the STM32F407 Discovery board, which acts as the sensor node in the system. Each sensor uses a different communication method, requiring individual configuration and testing.

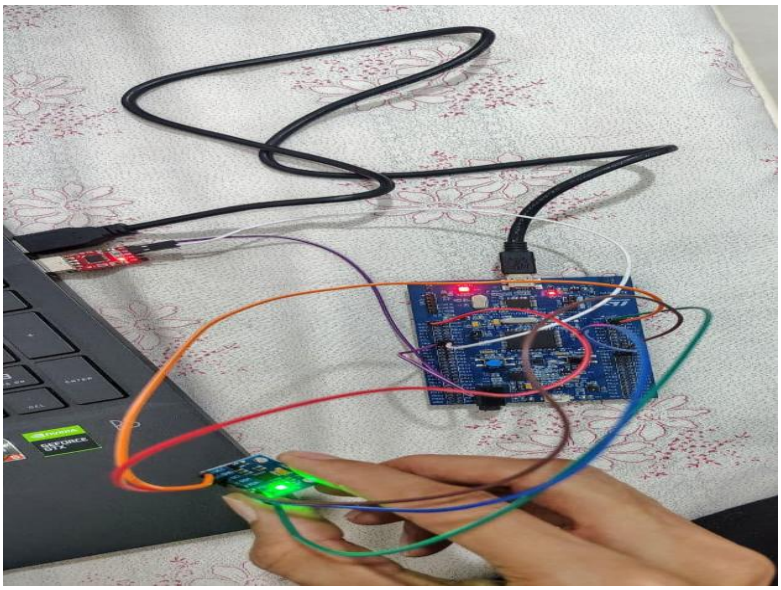


Figure 8.1: Interfacing of MPU6050, MQ2, and DHT11 Sensors with STM32F407

The MPU6050 sensor was interfaced using the I2C protocol, allowing the STM32F407 to read motion and vibration data such as acceleration and angular velocity. Proper I2C initialization, device addressing, and clock configuration were ensured for stable communication.

The MQ2 gas sensor was connected to the ADC input of the STM32F407. The sensor provides an analog voltage corresponding to gas or smoke concentration. The ADC was configured in polling mode to sample sensor values periodically.

The DHT11 sensor was interfaced using its single-wire digital communication protocol. Since DHT11 requires strict timing for data transmission, delay routines were carefully implemented to correctly decode temperature and humidity data.

Each sensor was tested individually to confirm correct interfacing before combining all sensors into a single application.

8.2 Testing Individual Sensor Readings

After successful interfacing, each sensor was tested independently to verify correct data acquisition. This step was essential to ensure that sensor values were accurate and stable before enabling communication with other system components.

The **MPU6050** was tested by observing changes in acceleration and motion data when the board was moved.

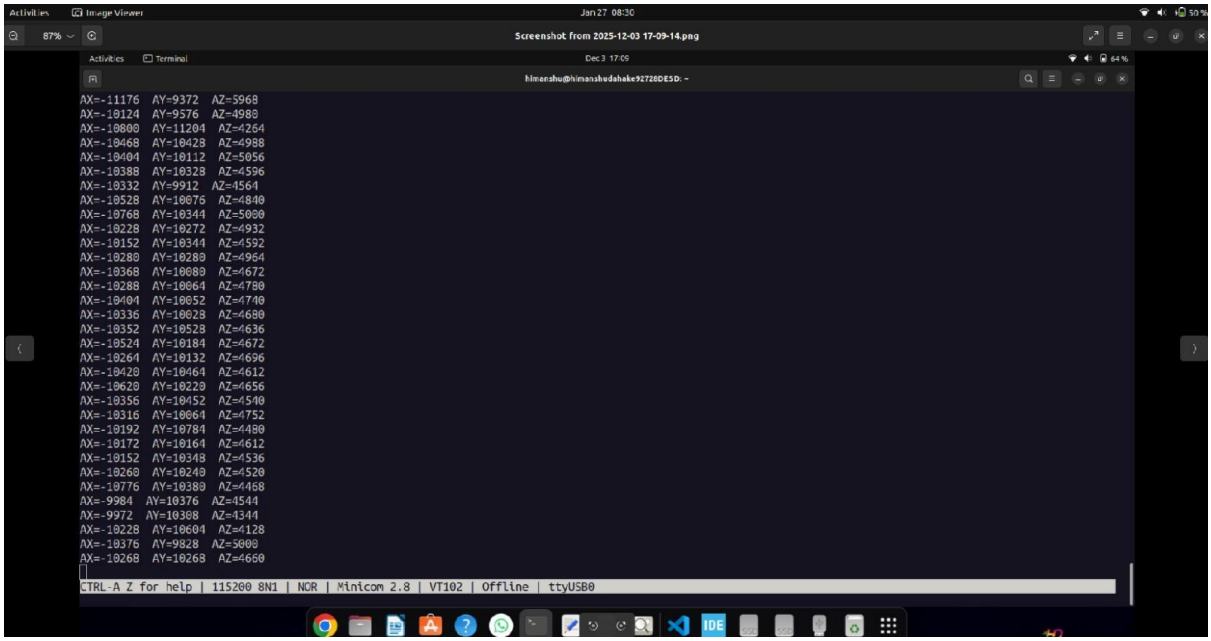


Figure 8.2: Serial Output Showing MPU6050 Sensor Readings

The **MQ2 sensor** readings were monitored under normal conditions and in the presence of smoke or gas to verify response variation.
The **DHT11 sensor** readings for temperature and humidity were checked against ambient conditions.

During this stage, sensor outputs were observed using **serial debugging** to ensure that the STM32F407 was receiving valid data.

8.3 Configuring CAN Communication and Message IDs

Once sensor data acquisition was verified, CAN communication was configured between the STM32F407 (sensor node) and the Nucleo F446RE (gateway).

The CAN peripheral on the STM32F407 was initialized with appropriate bit timing parameters to match the gateway configuration. Unique CAN message identifiers (IDs) were assigned to distinguish sensor data packets.

Each CAN frame contained formatted sensor data, ensuring consistency and ease of decoding at the gateway. Error handling and message filtering were configured to ensure reliable communication.

8.4 Transmitting Sensor Data over CAN

After CAN configuration, sensor data collected by the STM32F407 was packaged into CAN frames and transmitted periodically over the CAN bus. Each frame contained sensor readings mapped to specific data bytes.

Transmission was scheduled at fixed intervals to maintain consistency in data flow. Successful transmission was confirmed by monitoring CAN status flags and gateway reception.

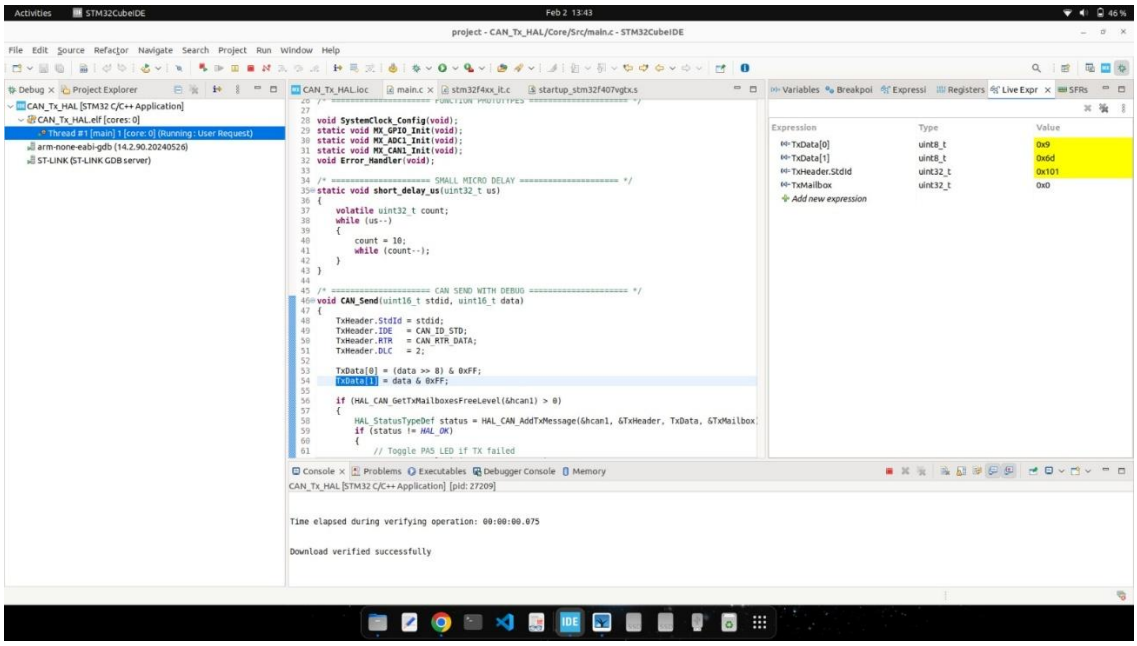


Figure 8.4: Transmission of Sensor Data over CAN Bus

8.5 Receiving CAN Data on Nucleo F446RE (Gateway)

The **Nucleo F446RE** was configured as the **gateway node**, responsible for receiving CAN messages sent by the STM32F407. CAN filters were configured to accept relevant message IDs.

Upon receiving a CAN frame, the gateway extracted sensor data and validated it before forwarding. This ensured that only correct and complete data was sent to the next stage.

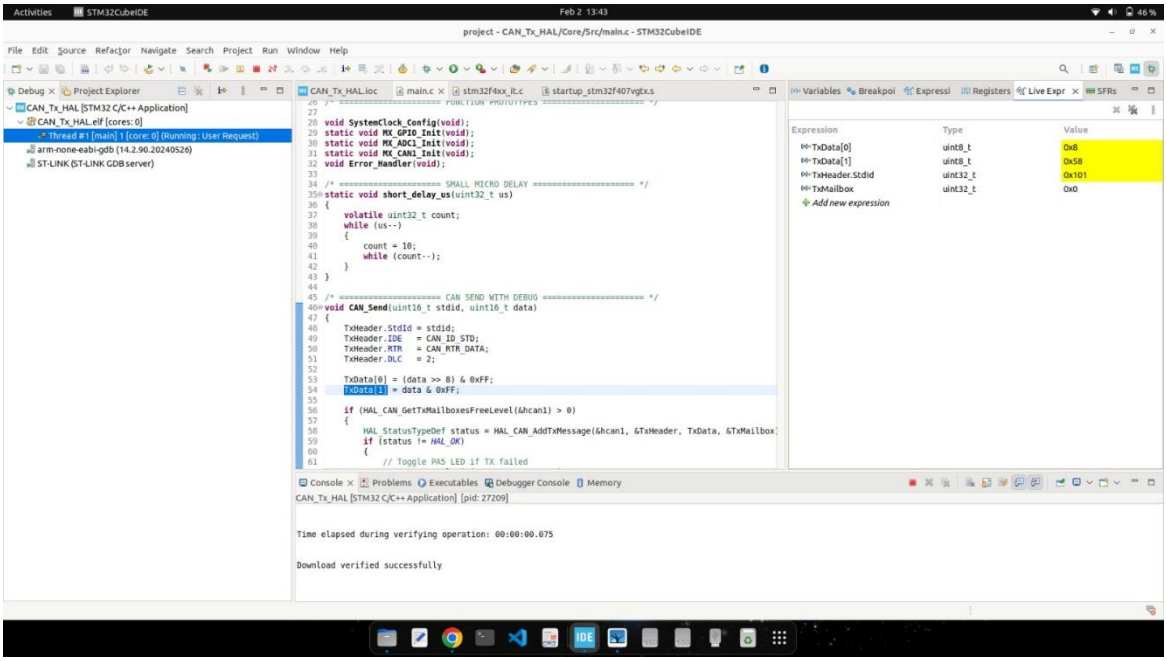


Figure 8.5: Received Sensor Data over CAN Bus

8.6 Forwarding Data to ESP32 via UART

After processing CAN data, the gateway forwarded the sensor readings to the ESP32 module using UART communication. A structured data format was used to ensure correct parsing at the ESP32 side.

UART baud rate, data bits, and stop bits were configured consistently on both devices. Transmission reliability was verified by monitoring UART debug outputs.

8.7 Sending Data to Blynk Cloud

In the final stage, the ESP32 transmitted sensor data to the Blynk cloud platform over Wi-Fi. The data was mapped to virtual pins and displayed in real time on the Blynk dashboard.

Threshold values were configured in Blynk to generate alerts when sensor readings exceeded predefined limits. This completed the end-to-end data flow from sensor node to cloud visualisation.

9. Results and Observations

9.1 Sensor Data Acquisition Results

After successful interfacing and testing, all sensors connected to the **STM32F407 sensor node** were able to generate valid and stable readings. The MPU6050 provided motion and acceleration data, the MQ2 sensor responded to changes in gas or smoke concentration, and the DHT11 sensor produced temperature and humidity values corresponding to ambient conditions.

The sensor readings were observed to change appropriately with environmental variations and physical movement of the board, confirming correct sensor operation and data acquisition.

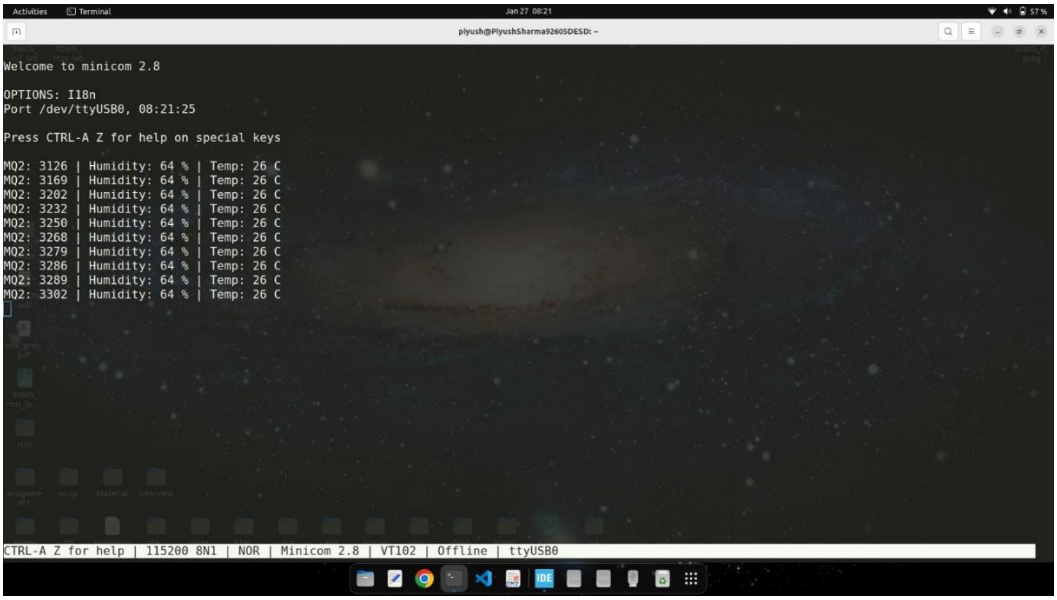


Figure 9.1: Serial Output Showing MQ2 and DHT-11 Sensor Readings

9.2 CAN Communication Results

CAN communication between the **STM32F407 sensor node** and the **Nucleo F446RE gateway** was tested using real sensor data. The sensor node successfully transmitted CAN frames at regular intervals, and the gateway was able to receive and decode the messages correctly.

No data loss or unexpected behavior was observed during normal operation. The use of CAN ensured reliable data transfer between the two controllers.

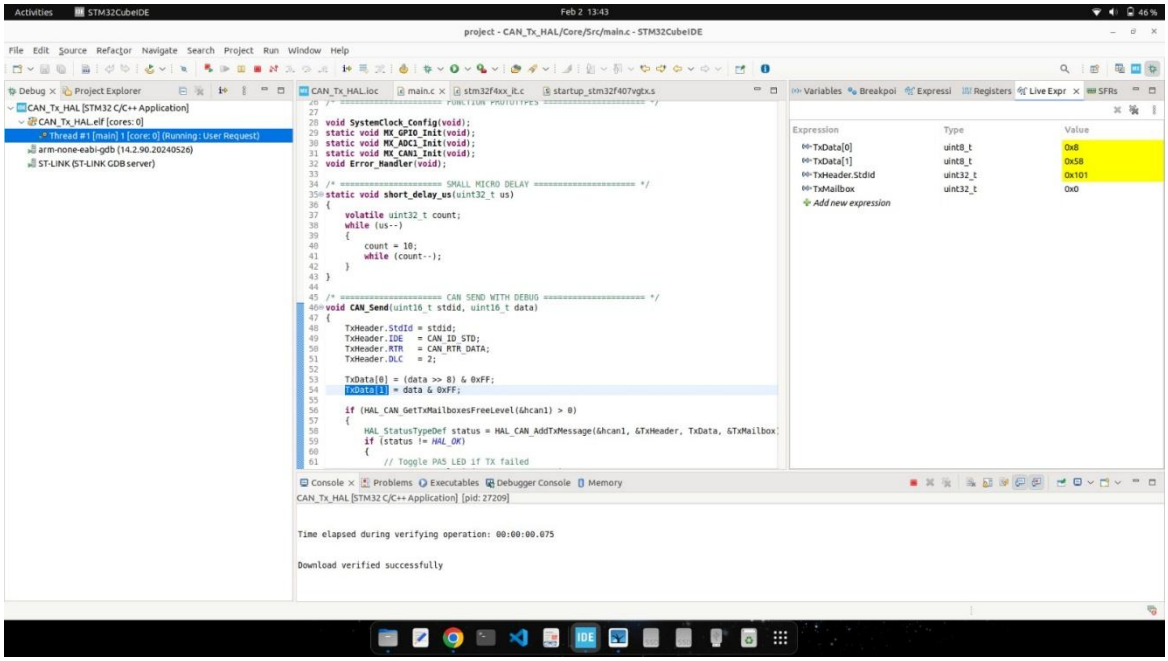


Figure 9.2: Received Sensor Data over CAN Bus

9.3 Cloud Data Visualization Results

The ESP32 successfully transmitted sensor data to the **Blynk cloud platform** over Wi-Fi. The Blynk dashboard displayed live values of temperature, humidity, gas concentration, and motion parameters.

The dashboard updated in real time, providing a clear visual representation of sensor data. This confirmed the successful integration of the embedded system with a cloud-based IoT platform.

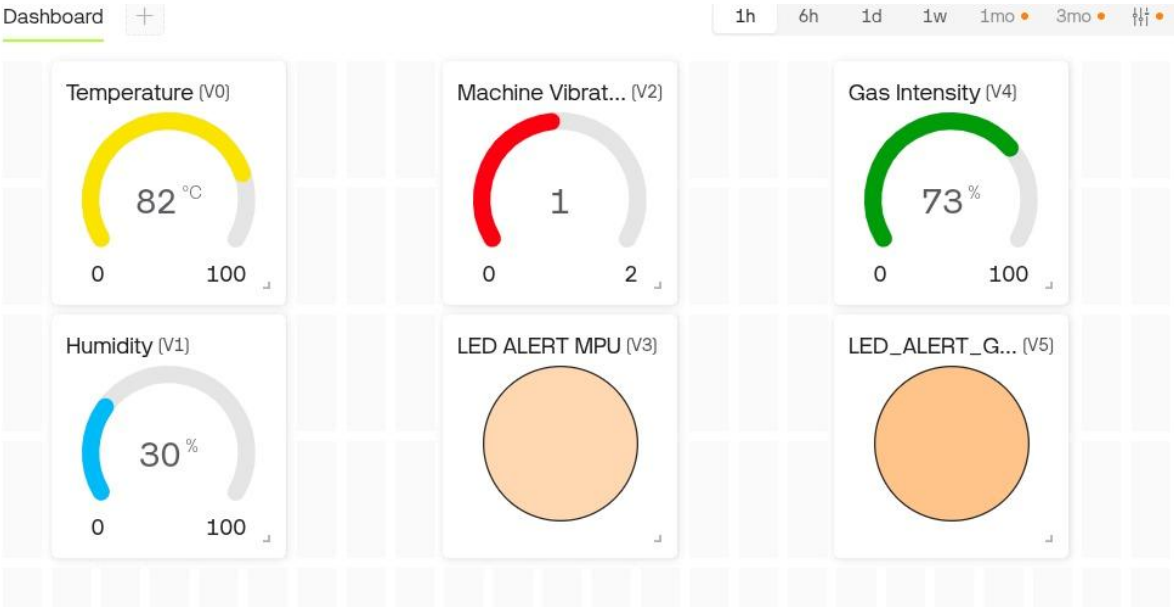


Figure 9.3: Real-Time Sensor Data Visualization on Blynk Dashboard

9.4 Overall System Performance

The complete system operated as intended, with reliable data flow from the sensor node to the cloud platform. Each stage of communication—sensor acquisition, CAN transmission, UART forwarding, and cloud upload—functioned correctly.

The results confirm that the project successfully demonstrates a **working IoT-based multi-sensor monitoring system** using STM32 microcontrollers, CAN communication, and ESP32-based cloud connectivity.

10. Challenges Faced and Solutions

During the development of this project, several challenges were encountered at different stages of implementation. These challenges were mainly related to sensor interfacing, communication configuration, and system integration. Each issue was resolved through systematic debugging and testing.

10.1 Sensor Interfacing Challenges

Interfacing multiple sensors with the STM32F407 required careful configuration, as each sensor used a different communication method. Timing issues were observed while interfacing the DHT11 sensor due to its strict communication protocol. Initial readings were inconsistent until proper delay routines were implemented.

The MQ2 sensor required stabilization time before producing meaningful readings. This was addressed by allowing a warm-up period before using the sensor data. MPU6050 communication issues were resolved by verifying I2C connections and addressing configuration.

10.2 CAN Communication Configuration Issues

Configuring CAN communication between STM32F407 and Nucleo F446RE was one of the major challenges. Initial communication failures were caused by mismatched bit timing parameters and incorrect CAN filter settings.

These issues were resolved by carefully matching CAN baud rate configurations on both boards and correctly setting message identifiers and acceptance filters. Continuous testing with sample frames helped verify reliable communication.

10.3 Data Formatting and Synchronization

Another challenge was maintaining consistency in data formatting across different stages of the system. Sensor data had to be properly packaged into CAN frames and later decoded at the gateway before forwarding to ESP32.

This issue was resolved by defining a structured data format and maintaining the same order of parameters throughout CAN and UART communication.

10.4 System Integration and Debugging

Integrating all components into a single system required extensive testing. Debugging was performed incrementally at each stage to isolate and resolve issues effectively.

This approach ensured that errors were identified early and did not propagate across the system.

11. Conclusion and Future Scope

11.1 Conclusion

This project successfully demonstrates the design and implementation of an **IoT-based multi-sensor monitoring system** using STM32 microcontrollers, CAN communication, and ESP32-based cloud connectivity. The system was able to acquire data from multiple sensors, transmit it reliably between controllers, and visualize it in real time on a cloud dashboard.

Through this project, I gained practical experience in sensor interfacing, embedded communication protocols, and IoT integration. The project fulfilled its objective of applying CDAC course concepts in a real-world embedded system setup.

11.2 Future Scope

The system developed in this project can be further enhanced in several ways:

- Addition of more sensor nodes using CAN
- Integration of data logging and analytics features
- Implementation of mobile notifications and email alerts
- Migration to more advanced cloud platforms
- Improved data processing and filtering techniques

These enhancements can extend the system's functionality and applicability in larger monitoring applications.

References

1. STMicroelectronics, “STM32F407 Reference Manual.”
2. STMicroelectronics, “STM32F446 Reference Manual.”
3. STMicroelectronics, “STM32CubeIDE User Guide.”
4. Bosch Sensortec, “MPU6050 Datasheet.”
5. DHT11 Temperature and Humidity Sensor Datasheet.
6. MQ-2 Gas Sensor Datasheet.
7. Microchip Technology, “MCP2551 CAN Transceiver Datasheet.”
8. Espressif Systems, “ESP32 Technical Reference Manual.”
9. Blynk Inc., “Blynk Platform Documentation.”