A

Project Report

On

"REMOTE MONITORING UNIT FOR INDUSTRY USING RTOS AND IoT"

Submitted in the partial fulfillment of the
requirements for the PG Diploma in

EMBEDDED SYSTEMS & DESIGN

Abhishek Shukla	250844230003
Sagar Kharade	250844230041
Aashik Mekala	250844230001

Sunbeam Institute of Information Technology, Pune & Karad.

Year: 2025-2026

Sunbeam Institute of Information Technology, Pune & Karad.

CERTIFICATE

This is to certify,

Mr. Mekala Aashik Babu	250844230001
Mr. Abhishek Shukla	250844230003
Mr. Kharade Sagar	250844230041

have satisfactorily completed the project work and successfully presented the project report titled “Remote Monitoring Unit for Industries Using IoT and RTOS” at Sunbeam Institute of Information Technology, in partial fulfilment of the requirements for the award of the Post Graduate Diploma in Embedded Systems & Design (PG-DESD) for the Academic Year 2025–2026

Date: 3/02/2026

ACKNOWLEDGEMENT

This project titled **“Remote Monitoring Unit for Industries using IoT and RTOS”** has been a valuable learning experience for us, and we are pleased to submit this work to **Sunbeam Institute of Information Technology** as part of our academic curriculum.

We express our sincere gratitude to **Vrushabh Patil** for his valuable guidance, continuous support, and technical mentorship throughout the project. His expertise, encouragement, and timely suggestions helped us overcome various challenges and gain a deeper understanding of the concepts involved in the successful development of this system.

We would also like to extend our heartfelt thanks to **Devendra Dhande**, Course Coordinator (PG-DESD), for his constant encouragement, effective coordination, and for providing all the necessary facilities such as hardware resources, internet access, and extended laboratory hours. His support throughout the course played a vital role in the timely completion of this project. Finally, we thank all the faculty members and staff for their cooperation and assistance during the entire course duration.

INDEX

ABSTRACT

This project presents the design and implementation of an Industrial Monitoring System using Real-Time Operating System (RTOS) and Internet of Things (IoT) technologies to enable reliable, real-time supervision of critical industrial parameters. The system continuously monitors operational variables such as temperature, humidity, and pressure, ensuring proper working conditions, improved system stability, and enhanced operational safety within industrial facilities.

The proposed solution utilizes an STM32F407 microcontroller running FreeRTOS to execute multiple tasks such as sensor interfacing, data acquisition, processing, and communication in a deterministic and time-synchronized manner. The RTOS-based architecture ensures predictable behavior, efficient resource management, and reliable performance required in time-critical industrial applications.

Sensor data collected by the controller is transmitted through UART communication to an ESP32 Wi-Fi module, which functions as an IoT gateway and uploads the readings to the Firebase cloud platform. A web-based dashboard provides remote monitoring and real-time visualization, allowing operators to access system data from anywhere through the internet.

The developed system offers a low-cost, scalable, and robust industrial monitoring solution, reducing manual inspection while improving reliability, automation, and overall operational efficiency.

1. INTRODUCTION

About Project:

Industrial systems require continuous monitoring of operational parameters to ensure stable performance and efficient functioning. Conventional monitoring methods often depend on manual inspection and periodic checks, which may lead to delays, inefficiencies, and lack of real-time visibility. To overcome these limitations, an automated monitoring solution is necessary to provide continuous data acquisition and remote accessibility.

The proposed Industrial Monitoring System using RTOS and IoT is designed to acquire and display important industrial parameters such as temperature, humidity, and pressure in real time. The system collects sensor data through an STM32F407 microcontroller, processes it using FreeRTOS-based task scheduling, and transmits the information to the cloud through an ESP32 Wi-Fi module. The collected data is stored on the Firebase platform and visualized on a web-based dashboard for remote observation.

The integration of IoT connectivity enables users to monitor system data from any location through an internet-enabled device. The use of RTOS ensures organized task management, timely sensor reading, and reliable communication between modules. This deterministic behavior improves system stability and makes the solution suitable for industrial monitoring applications where consistency and reliability are important.

Overall, the system provides a simple, low-cost, and scalable monitoring framework, reducing manual effort while enabling continuous and centralized supervision of industrial parameters.

1.1 Project Scope

The scope of this project is to design and develop an Industrial Monitoring System using RTOS and IoT technologies for continuous acquisition, processing, and remote visualization of important industrial parameters. The system focuses on implementing a reliable and structured embedded solution capable of collecting real-time data and transmitting it to the cloud for centralized monitoring.

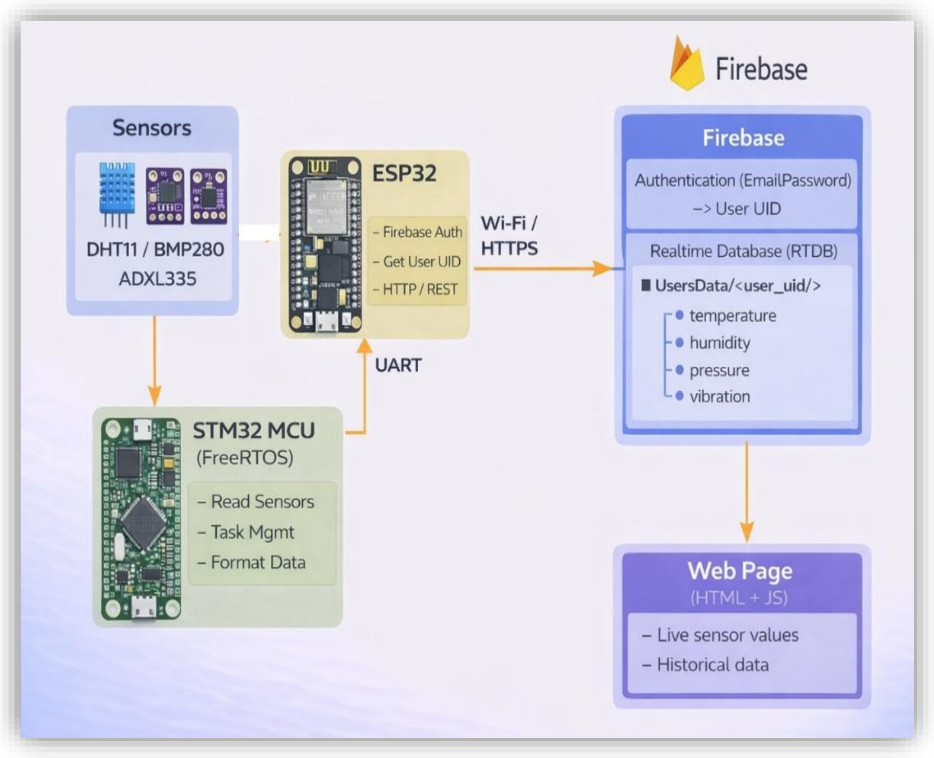
The project involves interfacing sensors for measuring parameters such as temperature, humidity, and pressure with an STM32F407 microcontroller operating under FreeRTOS. The RTOS-based architecture ensures organized task scheduling, efficient resource utilization, and deterministic execution of time-critical operations such as sensor reading and communication. The processed data is transferred to an ESP32 Wi-Fi module, which uploads the information to the Firebase cloud platform.

A web-based dashboard is developed to display the collected data in real time, enabling

users to monitor industrial conditions remotely through any internet-enabled device. The system demonstrates the practical integration of embedded systems, wireless communication, cloud services, and web technologies to create a scalable and low-cost industrial monitoring solution.

The current implementation focuses on data acquisition, cloud transmission, and real-time visualization. Further enhancements such as additional sensors, alert mechanisms, advanced analytics, and mobile applications can be incorporated in the future to extend the system's capabilities.

BLOCK DIAGRAM



1.2 RTOS Overview

A Real-Time Operating System (RTOS) is a specialized operating system designed to execute tasks within strict and predictable timing constraints. Unlike general-purpose operating systems, which focus on maximizing throughput or user interaction, an RTOS ensures deterministic behavior, meaning tasks are completed within defined time limits. This characteristic makes RTOS highly suitable for embedded and industrial systems where timing accuracy, reliability, and fast response are critical.

An RTOS manages multiple tasks simultaneously using features such as task scheduling, prioritization, inter-task communication, and resource management. It allows high-priority tasks to execute immediately while lower-priority tasks wait, ensuring that time-critical operations are never delayed. This structured execution improves system stability, reduces latency, and enables efficient utilization of microcontroller resources.

Key Features of RTOS:

In this project, these RTOS features enable separate tasks for sensor acquisition, data handling, and communication with the ESP32 module. This ensures organized execution, accurate timing, and reliable data transmission to the cloud, thereby improving the overall performance of the industrial monitoring system.

FreeRTOS in This Project:

In this project, **FreeRTOS** is implemented on the STM32F407 microcontroller to manage multiple operations efficiently. The system performs tasks such as sensor data acquisition, processing, and communication simultaneously. FreeRTOS enables multitasking by dividing these functions into separate tasks that run concurrently. Priority-based scheduling ensures that time-critical tasks like sensor reading are executed without delay. The RTOS also improves CPU utilization and reduces blocking during communication with the ESP32 module. This results in predictable and reliable system behavior. Hence, FreeRTOS enhances the stability and real-time performance of the industrial monitoring system.

1.3 IoT Overview

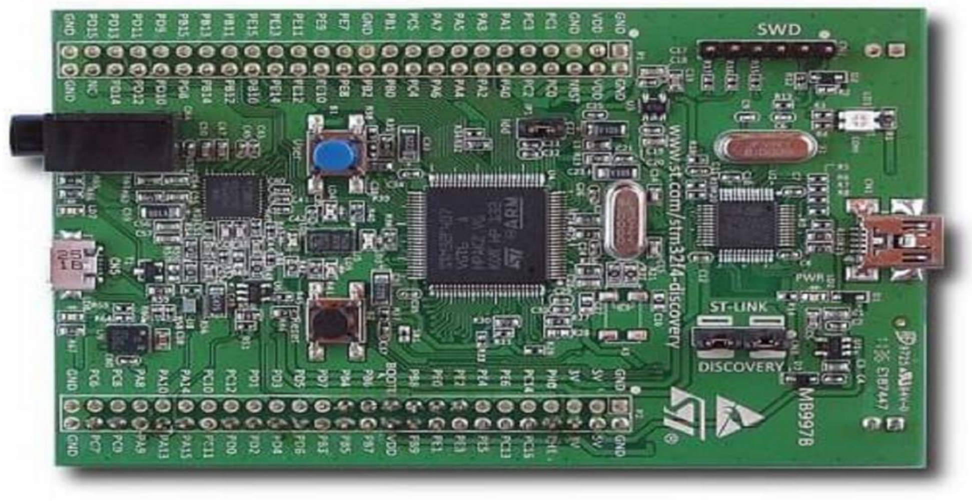
In this project, Internet of Things (IoT) technology is used to enable remote monitoring of industrial parameters through cloud connectivity. The STM32 microcontroller collects sensor data and sends it to the ESP32 Wi-Fi module via UART communication. The ESP32 acts as an IoT gateway and uploads the data to the Firebase cloud platform using internet protocols. This allows real-time storage and access of data from anywhere. A web-based dashboard retrieves the cloud data and displays it for continuous monitoring. IoT integration eliminates the need for manual inspection and provides centralized supervision. Thus, IoT enhances accessibility, flexibility, and overall efficiency of the industrial monitoring system.

2. HARDWARE REQUIREMENTS

1. STM32F407 Discovery Board

The STM32F407G-DISC1 Discovery Kit is designed to help both beginners and experienced users explore the features of the STM32F407/417 line and develop applications. It's based on the STM32F407VGT6 microcontroller and includes an embedded ST-LINK/V2 debug tool, MEMS sensors, an audio DAC, LEDs, push buttons, and a USB

OTG micro-AB connector.



Key Features and Specifications:

Microcontroller: STM32F407VGT6 featuring a 32-bit ARM Cortex-M4F core, 1 MB Flash, and 192 KB RAM in an LQFP100 package. The core runs at up to 168 MHz and includes a floating-point unit (FPU) and Digital Signal Processing (DSP) instructions for high-performance embedded applications.

On-board ST-LINK/V2: With a selection mode switch, it can be used as a standalone ST-LINK/V2 with an SWD connector for programming and debugging. Newer boards use ST-LINK/V2-A.

Power Supply: Can be powered through a USB bus or an external 5 V supply voltage. It also provides an external application power supply at 3 V and 5 V.

MEMS: Includes a LIS302DL or LIS3DSH ST MEMS 3-axis accelerometer and an MP45DT02 ST MEMS audio sensor, which is an omnidirectional digital microphone.

LEDs: Eight LEDs are present: LD1 (red/green) for USB communication, LD2 (red) for 3.3 V power on, four user LEDs (orange, green, red, and blue).

Push Buttons: It has two push buttons for user input and reset.

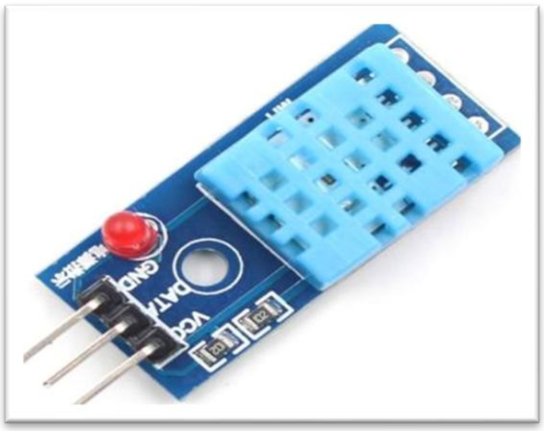
USB OTG FS: Comes with a micro-AB connector.

Debug and Programming: The ST-LINK circuitry allows you to flash code onto the microcontroller and debug it using host software like Keil or STM Cube IDE.

The STM32F4 Discovery board allows users to develop and design applications, offering modules for communication and interface design without relying on third-party devices. It incorporates modern system modules and peripherals like DAC, ADC, and UART. It supports a wide choice of Integrated Development Environments (IDEs) including IAR Embedded Workbench®, MDK-ARM, and STM32CubeIDE.

2. DHT11 Temperature and Humidity Sensor

The DHT11 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and outputs a digital signal on the data pin. The sensor is calibrated and provides reliable readings with good long-term stability.

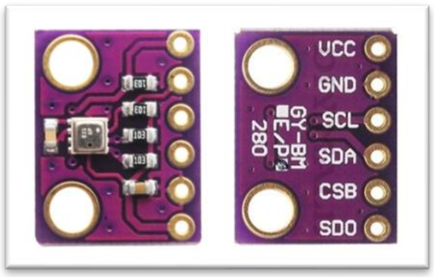


Specifications:

- Sensor Type: Digital temperature and humidity sensor
- Operating Voltage: 3.3V to 5V
- Temperature Range: 0°C to 50°C
- Humidity Range: 20% to 90% RH
- Accuracy: ±2°C (temperature), ±5% RH (humidity)
- Interface: Single-wire digital communication

3. BMP280 Pressure Sensor

BMP280 is a digital barometric pressure and temperature sensor used for precise atmospheric measurements. It operates in the range of 300–1100 hPa pressure and −40°C to +85°C temperature with high accuracy and low power consumption. The sensor supports I2C and SPI communication, making it easy to interface with microcontrollers like STM32..

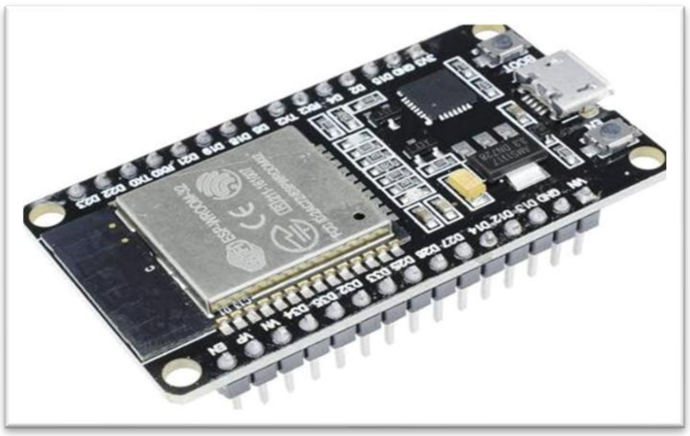


Specifications:

- Pressure Range: 300-1100 hPa (equivalent to +9000m to -500m altitude)
- Pressure Accuracy: ±1 hPa
- Operating Voltage: 1.71V to 3.6V
- Interface: I2C and SPI
- Power Consumption: Ultra-low power, 2.7 µA at 1 Hz sampling rate

4. ESP32 Wi-Fi Module

ESP32 is a low-cost, high-performance microcontroller with built-in Wi-Fi and Bluetooth connectivity, widely used in IoT applications. In this project, it acts as a communication gateway between the STM32 controller and the cloud platform. It receives sensor data via UART and uploads it to Firebase for real-time remote monitoring..



Specifications:

- Processor: Dual-core Tensilica LX6 32-bit microprocessor up to 240 MHz
- Memory: 520 KB SRAM, external 4 MB Flash
- Wi-Fi: 802.11 b/g/n with WPA/WPA2 encryption
- Bluetooth: Bluetooth v4.2 (Classic + BLE)
- Operating Voltage: 3.0V to 3.6V
- GPIO: Up to 34 programmable GPIO pins
- Interfaces: UART, SPI, I2C, ADC, DAC, PWM

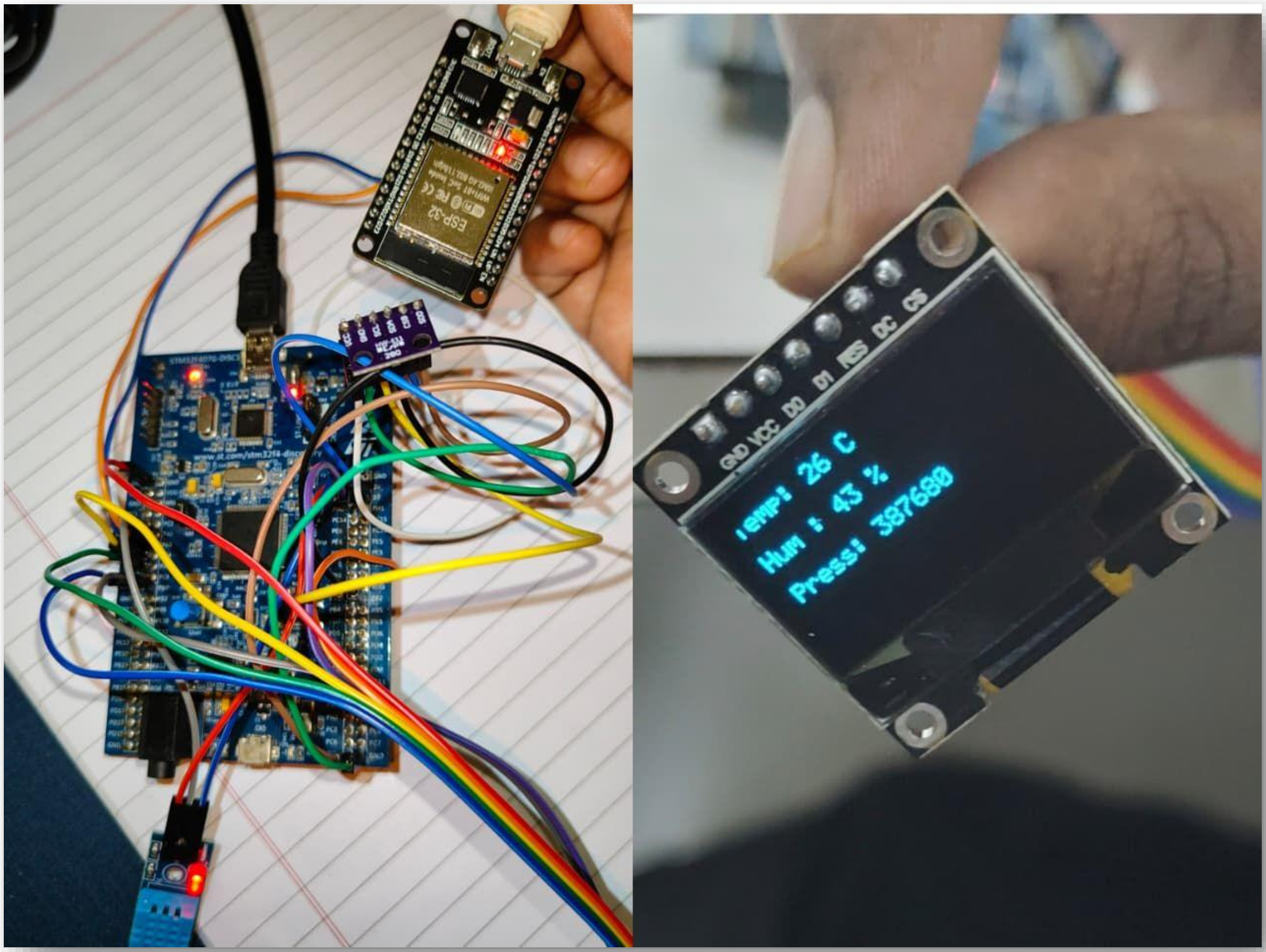
5. Additional Components

OLED Display (16x2): For local display of sensor readings and system status

LEDs: Status indicators for system operation and alerts

Power Supply: 5V regulated power supply for the entire system

Connecting Wires : For circuit assembly and connections.


ESP-32
GND VCC D0 D1 RES DC CS
emp: 26 C
Hum : 43 %
Press: 387680

3. SOFTWARE REQUIREMENTS

STM32CubeIDE is a free, all-in-one integrated development environment (IDE) from STMicroelectronics, designed for developing applications on STM32 microcontrollers and microprocessors. It is part of the STM32Cube software ecosystem and is based on the Eclipse/CDT framework. STM32CubeIDE integrates features from STM32CubeMX for configuration and project management, along with a GNU C/C++ compiler toolchain and GDB debugger.

Key features:

Integration of STM32CubeMX: Allows for the selection of STM32 microcontrollers, pin assignments, clock and peripheral configuration, and initialization code generation.

Eclipse-based: Supports Eclipse plug-ins and uses the GNU C/C++ toolchain for ARM and GDB debugger.

Debugging Tools: Offers advanced debugging features, including CPU core, peripheral register, and memory views, live variable watch, system analysis, real-time tracing, and CPU fault analysis.

Build and Stack Analyzers: Includes build and stack analyzers that provide information on project status and memory requirements.

Project Import: Supports importing projects from Atollic TrueSTUDIO and AC6 System Workbench for STM32 (SW4STM32).

Multi-OS Support: Compatible with Windows, Linux, and MacOS.

STM32CubeIDE helps developers with chip selection, project configuration, code generation, editing, compiling, debugging, and burning. It uses a workspace to manage projects, where each workspace is a folder containing project folders and a ".metadata" folder with project information.

2. FreeRTOS

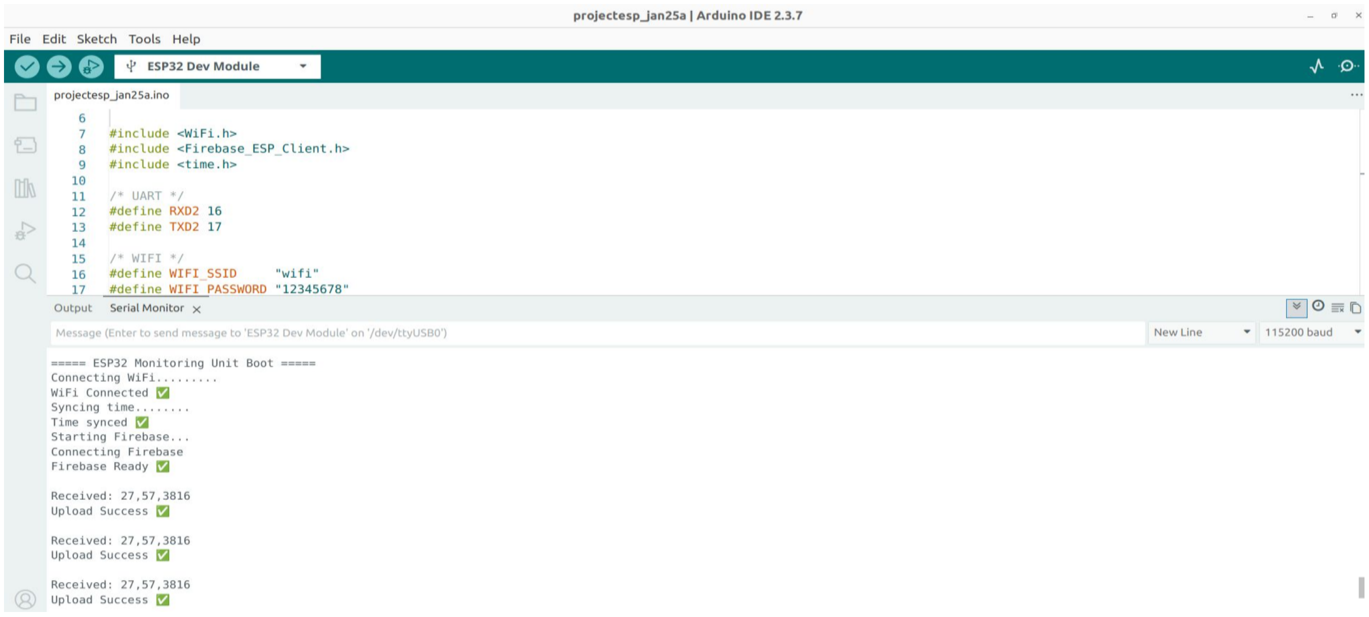
FreeRTOS is a real-time operating system kernel for embedded devices that has been ported to numerous microcontroller platforms. It is designed to be small, simple, and easy to use, while providing sophisticated features for task scheduling and inter-task communication.

Key Features:

- Preemptive and cooperative scheduling options
- Queue management, semaphores, and mutexes for task synchronization
- Software timers for delayed execution
- Low power tickless mode for energy efficiency Stack overflow detection

3. Arduino IDE

Used for programming the ESP32 Wi-Fi module. The Arduino IDE provides a simple interface for writing, compiling, and uploading code to the ESP32. It includes libraries for Wi-Fi connectivity, HTTP communication, and MQTT protocol implementation.


projectesp_jan25a | Arduino IDE 2.3.7
File Edit Sketch Tools Help
ESP32 Dev Module
projectesp_jan25a.ino
Output
Serial Monitor

```

#include <WiFi.h>
#include <Firebase_ESP_Client.h>
#include <time.h>

/* UART */
#define RXD2 16
#define TXD2 17

/* WIFI */
#define WIFI_SSID     "wifi"
#define WIFI_PASSWORD "12345678"
```

Message (Enter to send message to 'ESP32 Dev Module' on '/dev/ttyUSB0')

New Line — 115200 baud

```
===== ESP32 Monitoring Unit Boot =====
Connecting WiFi.........
WiFi Connected ✅
Syncing time........
Time synced ✅
Starting Firebase...
Connecting Firebase
Firebase Ready ✅

Received: 27,57,3816
Upload Success ✅

Received: 27,57,3816
Upload Success ✅

Received: 27,57,3816
Upload Success ✅
```

4.Cloud Platform:

The system uses a cloud platform to store and manage sensor data for remote monitoring and accessibility. In this project, Firebase Realtime Database is used to upload and store data received from the ESP32 through Wi-Fi. The cloud enables real-time data synchronization between the hardware and the web dashboard. This allows users to view live readings from anywhere using an internet-enabled device. Cloud integration ensures centralized monitoring, easy scalability, and reliable data access.

4. DESIGN AND IMPLEMENTATION

System Architecture:

The proposed system follows a layered architecture consisting of sensing, processing, communication, cloud, and visualization stages. Sensors such as DHT11 and BMP280 collect real-time industrial parameters and send the data to the STM32F407 microcontroller for processing. The STM32, running FreeRTOS, manages sensor acquisition and formats the data efficiently. The processed information is transmitted to the ESP32 module via UART, which acts as an IoT gateway. ESP32 uploads the data to the Firebase cloud platform through Wi-Fi. Finally, a web-based dashboard retrieves and displays the data for remote monitoring and supervision.

Task Structure:

The system is organized using multiple FreeRTOS tasks to ensure efficient and concurrent execution of operations. A dedicated **Sensor Task** periodically reads temperature, humidity, and pressure data from the connected sensors. A **Processing Task** formats and prepares the collected data for transmission. A **Communication Task** handles UART communication between the STM32 and ESP32 modules to send sensor readings to the cloud. These tasks run concurrently with assigned priorities to ensure timely execution of critical operations. This structured task division improves system reliability, responsiveness, and real-time performance.

Communication Protocol:

The proposed system uses multiple communication protocols to ensure reliable data transfer between different modules. I2C protocol is used for interfacing sensors such as BMP280 with the STM32 microcontroller. UART communication is employed for serial data exchange between the STM32 and ESP32 modules. The ESP32 utilizes Wi-Fi to connect the system to the internet for cloud communication. Sensor data is transmitted to the Firebase platform using HTTP/HTTPS protocols, enabling secure and real-time cloud data transfer. These protocols together ensure efficient, stable, and seamless communication throughout the system.

Data Flow

1. Sensors measure temperature, humidity, and pressure parameters continuously.
2. STM32F407 reads sensor data using interfaces such as I2C and GPIO.
3. FreeRTOS tasks process and organize the acquired readings.
4. The processed data is transmitted to the ESP32 module via UART communication.
5. ESP32 connects to Wi-Fi and uploads the data to the Firebase cloud platform using HTTP/HTTPS.
6. The cloud database stores the received data in real time.
7. The web dashboard retrieves the cloud data and displays live readings for remote monitoring.

Implementation Highlights:

- Interfaced DHT11 and BMP280 sensors with STM32F407 for real-time parameter acquisition.
- Implemented **FreeRTOS** to create separate tasks for sensor reading, data processing, and communication.
- Established **UART communication** between STM32 and ESP32 for reliable serial data transfer.
- Configured ESP32 Wi-Fi module to upload sensor data to the **Firebase Realtime Database**.
- Developed a **web-based dashboard** using HTML, CSS, and JavaScript for live data visualization.
- Achieved real-time cloud monitoring with stable and continuous data updates.
- Designed the system to be modular and scalable for adding additional sensors in the future.

5. ADVANTAGES

- Provides real-time monitoring of important industrial parameters.
- Enables remote access to data through cloud and web dashboard.
- Reduces manual inspection and human effort.
- Improves system reliability using RTOS-based task management.
- Low-cost and easy-to-implement IoT solution.
- Scalable design allows addition of more sensors and features.
- Wireless communication eliminates complex wiring.

Limitations

- Requires continuous Wi-Fi connectivity for cloud data transmission.
- Limited to basic monitoring and does not provide automatic control or alert mechanisms.
- Supports only a small number of sensors in the current implementation.
- Depends on cloud services for data storage and visualization.
- Not suitable for extremely harsh industrial environments without additional protection.
- Basic dashboard provides visualization only without advanced analytics.

7. FUTURE SCOPE

- Integration of additional sensors to monitor more industrial parameters.
- Implementation of alert and notification systems for abnormal readings.
- Development of a mobile application for easier remote access.
- Addition of data logging and graphical analysis for historical trends.
- Support for more communication protocols such as MQTT for large-scale deployments.
- Integration with industrial automation or control systems for automatic actions.
- Enhancement of security features for safe cloud communication.

8. CONCLUSION

The Industrial Monitoring System using RTOS and IoT successfully demonstrates the practical implementation of embedded systems, real-time task management, and cloud-based connectivity for industrial applications. The developed system efficiently acquires critical parameters such as temperature, humidity, and pressure using multiple sensors interfaced with the STM32F407 microcontroller. The integration of FreeRTOS enables structured multitasking, deterministic execution, and reliable handling of concurrent operations such as sensor reading, data processing, and communication.

The ESP32 Wi-Fi module acts as an IoT gateway, transmitting processed data to the Firebase cloud platform, where it is stored and visualized through a web-based dashboard. This architecture enables real-time remote monitoring, reduces the need for manual supervision, and improves overall system accessibility and reliability. The combination of RTOS scheduling and IoT connectivity ensures stable and continuous performance suitable for industrial environments.

Overall, the project provides a low-cost, scalable, and efficient monitoring solution while offering valuable hands-on experience in embedded programming, communication protocols, cloud integration, and system design. The developed system lays a strong foundation for future enhancements and can be extended further to support advanced industrial automation requirements.

REFERENCES

1. R. Barry, Mastering the FreeRTOS Real Time Kernel – A Hands-On Tutorial Guide, Real Time Engineers Ltd., 2021.
2. STMicroelectronics, STM32F407 Reference Manual (RM0090), Rev. 19, 2024.
3. R. Patel and S. Kumar, “IoT Based Industrial Monitoring and Control System,” International Journal of Engineering Research and Technology, vol. 12, no. 8, Aug. 2023.
4. E. A. Lee, “Cyber Physical Systems: Design Challenges,” IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, pp. 363–369, 2008.
5. C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, “DDoS in the IoT: Mirai and Other Botnets,” Computer, vol. 50, no. 7, pp. 80–84, 2017.
6. FreeRTOS Documentation, Real Time Engineers Ltd., 2024. Available: www.freertos.org
7. Espressif Systems, ESP32 Technical Reference Manual, 2024.
8. Bosch Sensortec, BMP280 Digital Pressure Sensor Datasheet, Rev. 1.19, 2024.
9. A. Gupta and R. K. Jha, “Industrial IoT: Applications, Security Threats, and Countermeasures,” IEEE Communications Surveys & Tutorials, vol. 25, no. 2, pp. 1234–1268, 2023.