

Real-Time CAN Communication System with FreeRTOS and IoT Integration

Project Members:

Asha Karwar

Jayshree Madankar

Alfiya Ibushe

Nihal Jalal

Content

1. Project Aim

2. Objectives

3. Introduction

4. Hardware
Components

5. Software
Technologies

6. Block Diagram

7. Future and
Application

8. Output



Project Aim

To design and implement a real-time embedded system using STM32 microcontroller with CAN protocol for reliable sensor data communication and remote IoT monitoring.

Objectives

- Implement multi-node communication using CAN protocol.
- Use FreeRTOS to manage tasks.
- Transmit sensor data via UART to ESP8266.
- Display data on LCD and Blynk dashboard.
- Ensure real-time monitoring.



Introduction

This project demonstrates a multi-node embedded communication system using STM32, FreeRTOS, and CAN protocol. Sensors such as MQ5, LDR, and DHT11 are used to gather data which is then transmitted for monitoring using IoT.

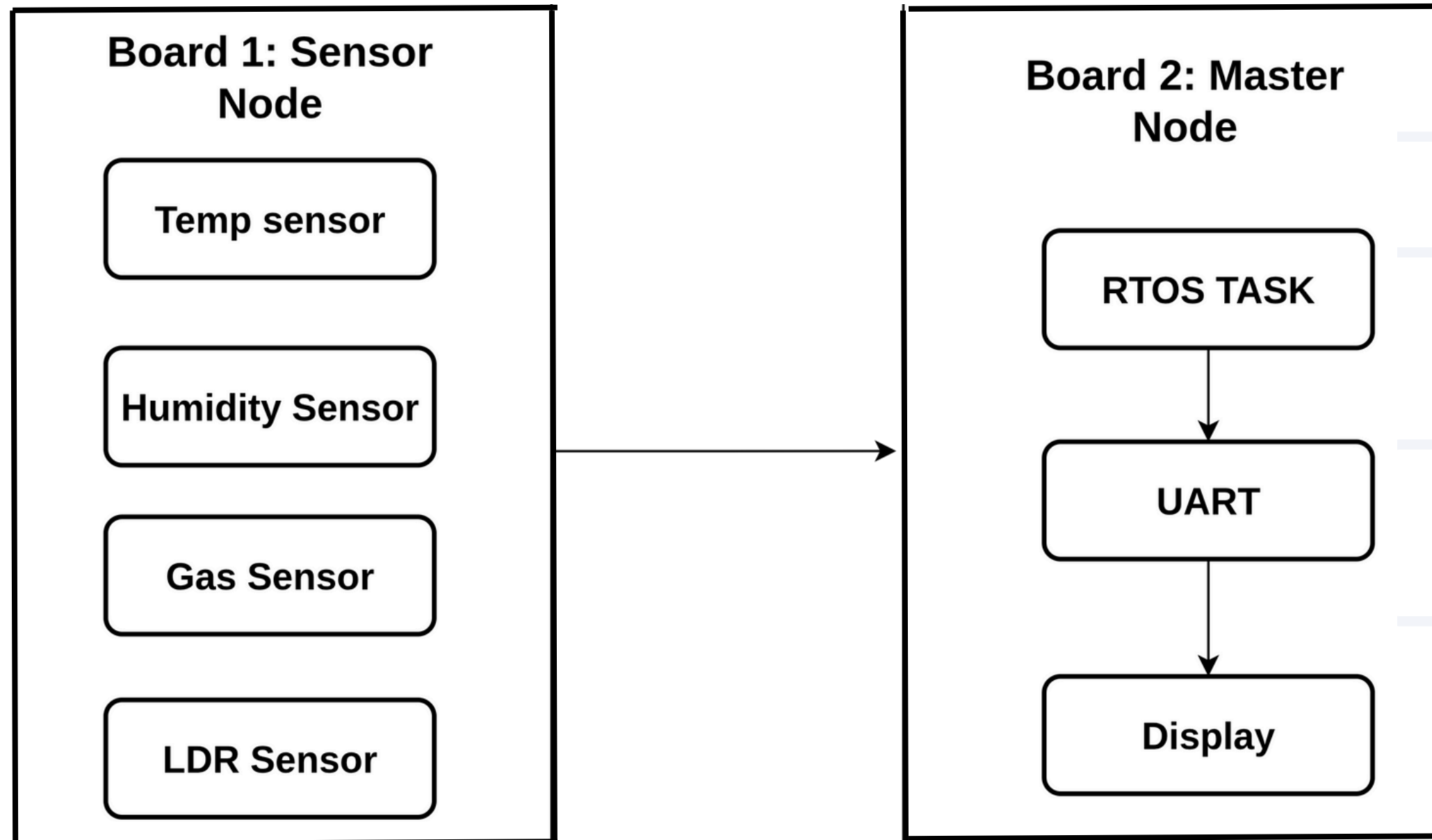
Hardware Components

- STM32F407 Microcontrollers (2 nodes) .
- MQ5 Gas Sensor, LDR Sensor, DHT11 Temp and Humidity Sensor .
- CAN Transceivers.
- ESP8266 (NodeMCU).
- 16x2 I2C LCD Display.

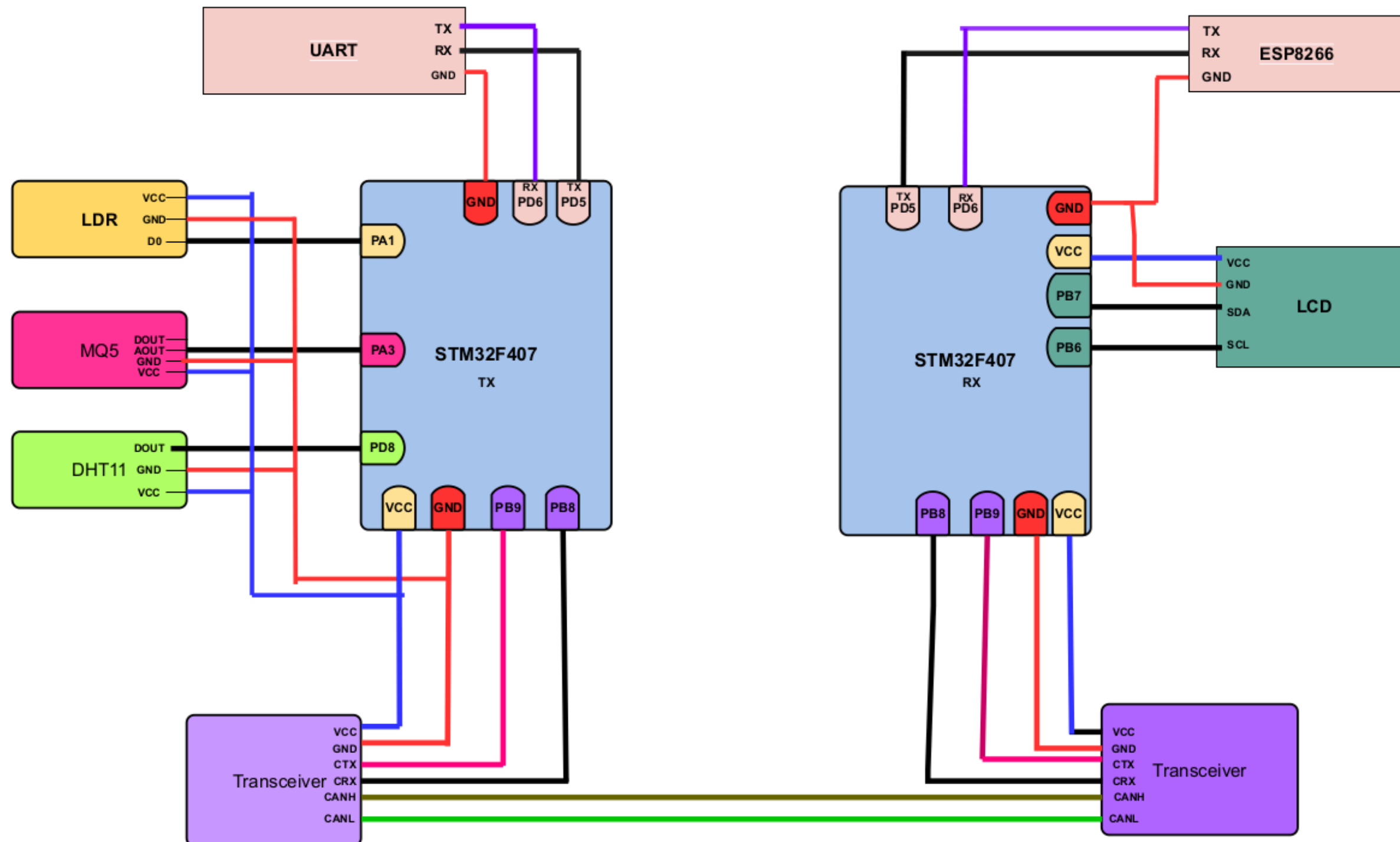
Software Technologies

- FreeRTOS for real-time task management
- HAL drivers for ADC, UART, I2C, CAN
- Flask + MySQL server for IoT backend
- Blynk for real-time mobile dashboard

Block Diagram



Pin Diagram



Project Architecture

- TX Node reads data from sensors → sends Using CAN.
- RX Node receives CAN data → displays on LCD.
- RX Node forwards data using UART to ESP8266.
- ESP8266 uploads data to Flask server and Blynk dashboard.

FreeRTOS Tasks

- Sensor Reading Task (MQ5, LDR, DHT11)
- CAN Communication Task (TX and RX)
- LCD Display Task
- UART Communication Task to ESP8266

The slide features a dark blue header bar with the title 'CAN Communication' in white. The background is white with faint, light blue rounded rectangular outlines. Dark blue geometric shapes, including triangles and parallelograms, are positioned in the top-left and bottom-right corners.

CAN Communication

- Used CAN protocol for STM32 to STM32 communication.
- Reliable, real-time, and noise-resistant data transfer.
- Each message contains sensor data packet.
- CAN filtering used to process specific IDs.

A dark blue arrow pointing left, containing the text "IoT Integration" in a bold, dark blue font.

IoT Integration

- ESP8266 sends data to cloud via HTTP.
- Flask server receives and stores in MySQL.
- Blynk app shows live sensor data on phone.
- Remote monitoring enabled over Wi-Fi.

Future Enhancements

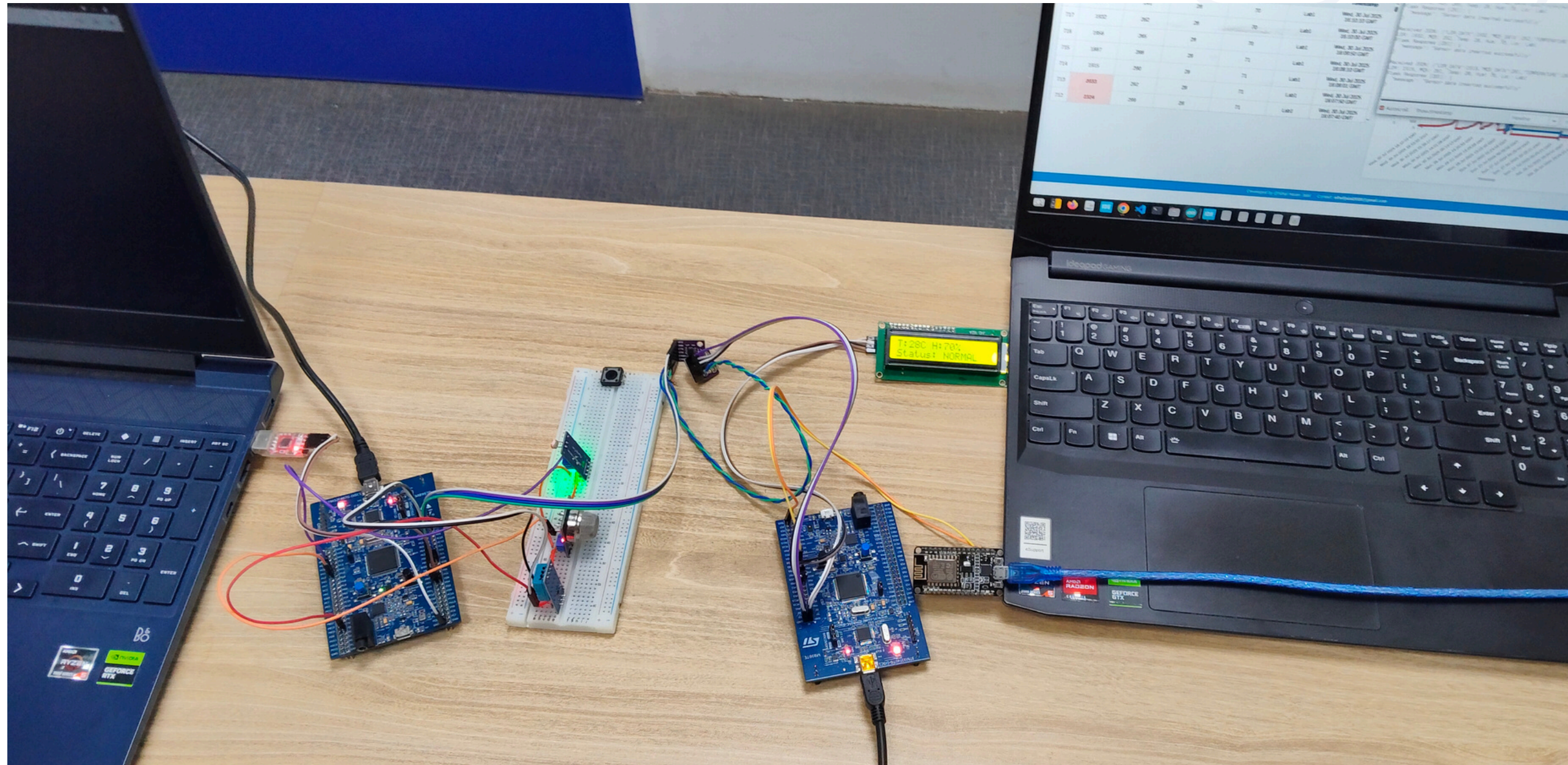
- Add more sensor nodes on CAN bus.
- Add encryption/security to data transmission.
- Improve calibration for gas sensor .
- Add web-based dashboard for data visualization.



Application

- Smart Environment Monitoring System
- Gas Leakage Alert System
- Industrial Safety Systems
- Remote Sensor Data Dashboard

OUTPUT



<https://drive.google.com/file/d/1YwYvyide56KVHidUxSF0YnXKjhQHA1qU/view?usp=drivesdk>

The image features a white background with a dark blue header and footer. The central text "Thank You" is in a bold, dark blue font. Scattered throughout the white area are numerous rounded squares of varying sizes, some with dark blue outlines and others with light blue outlines. The text is centered horizontally and has a slight drop shadow.

Thank You