Tech-Sphere Journal of Pure and Applied Sciences (TSJPAS)
A Subsidiary of Tech-Sphere Multidisciplinary International Journal (TSMIJ)
Vol 1, Issue 1, 2024 Publication Edition
ISSN: 3672-4648

Node-RED and IoT Analytics: A Real-Time Data Processing and Visualization Platform

Onwuegbuzie Innocent Uzougbo[1*], Olowojebutu Akinyemi Olanrewaju[2], Akomolede Kehinde Kayode[3]
1, 2, 3 Computer Science Department, The Federal Polytechnic Ado-Ekiti, Ekiti State, Nigeria
*Corresponding author's email: onwuegbuzie_in@fedpolyado.edu.ng
DOI: https://doi.org/10.5281/zenodo.13856860
Publication URL: https://stem.techspherejournal.com/index.php/tsjpas/article/view/13

ABSTRACT

With the advent of IoT, vast amounts of data are generated by sensors and devices in real-time, posing significant challenges in terms of processing and analyzing the data due to its complexity and scale. This paper investigates the integration of Node-RED and IoT analytics for real-time data processing and visualization in the context of the Internet of Things (IoT). Node-RED is an open-source visual programming tool that simplifies the development of IoT applications by providing an intuitive interface for designing data flows between devices and services. On the other hand, IoT analytics offers a range of techniques and tools for processing and analyzing IoT data in real-time. A comprehensive overview of Node-RED and IoT analytics and their potential synergy for developing real-time IoT applications is discussed. Furthermore, it presents a case study of an industrial IoT system that leverages Node-RED and IoT analytics to process and visualize real-time data from multiple sensors, demonstrating the effectiveness of the integrated approach in reducing development time and complexity while providing a flexible and scalable solution. The study underscores the significance of integrating Node-RED and IoT analytics for developing real-time IoT applications. It offers a user-friendly interface for designing data flows and enables real-time processing and visualization of IoT data, promising a powerful approach to overcome the challenges of real-time data processing and visualization in the IoT domain, that supports productive business decisions. In the end, it presents potential future research directions to enhance their performance and reliability.

Keywords: Node-Red, IoT Analytics, Data Visualization, Raspberry Pi.

1 Introduction

The Internet of Things (IoT) has brought about significant changes in the way we interact with technology, and it has impacted many areas of our lives, including healthcare, transportation, manufacturing, and agriculture. The IoT enables devices to connect and communicate with each other, generating vast amounts of data that can be used for various applications. However, the challenge of processing and analyzing such large amounts of data presents significant obstacles for organizations that are looking to harness the power of the IoT.

Node-RED is an open-source visual programming tool that simplifies the development of IoT applications. It provides an intuitive interface for designing data flows between devices and services, which can help to reduce the complexity and time required to develop IoT applications. Additionally, Node-RED enables the integration of a variety of data sources and services, such as databases, APIs, and cloud services, which can help organizations to leverage the full potential of their IoT infrastructure.

IoT analytics provides a set of techniques and tools for processing and analyzing data generated by IoT devices in real-time. It enables organizations to extract meaningful insights from IoT data and make informed decisions that can improve their operations and business outcomes. IoT analytics encompasses various data processing

ISSN: 3672-4648

techniques, including data cleansing, data transformation, and data enrichment, as well as analytical methods such as machine learning, predictive analytics, and anomaly detection.

This paper explores the integration of Node-RED and IoT analytics for real-time data processing and visualization. The combination of these two technologies can provide organizations with a powerful platform for developing real-time IoT applications that can process, analyze, and visualize IoT data in real-time. By combining the simplicity of Node-RED with the analytical capabilities of IoT analytics, organizations can develop flexible, scalable, and user-friendly solutions for their IoT infrastructure.

The paper aims to demonstrate the potential of integrating Node-RED and IoT analytics for real-time data processing and visualization. The combination of these two technologies can provide organizations with a powerful platform for developing real-time IoT applications that can process, analyze, and visualize IoT data in real-time. The paper provides a comprehensive overview of Node-RED, IoT analytics, and their integration, as well as a case study that demonstrates the effectiveness of this approach in an industrial IoT system.

2 Overview of Node-Red

Node-RED is an open-source visual programming tool that enables the creation of IoT applications by connecting various devices, APIs, and services. It uses a flow-based programming model, where the application is built by connecting nodes that represent different functions. Each node performs a specific task, such as reading data from a sensor, transforming data, or sending data to a database. Nodes can be easily customized, and new nodes can be created using JavaScript. One of the key advantages of Node-RED is its simplicity and ease of use. The visual interface allows developers to quickly build and modify data flows, making it an ideal tool for prototyping and testing. Node-RED supports a wide range of protocols and data formats, including the Message Queuing Telemetry Transport (MQTT) protocol, Hyper Text Transfer Protocol (HTTP), and JavaScript Object Notation (JSON), which enables it to connect to a variety of devices and services. Node-RED also provides a rich library of pre-built nodes that can be used to create IoT applications quickly. These nodes include nodes for working with databases, web services, and IoT devices such as sensors and actuators.

3 IoT Analytics

IoT analytics is a set of techniques and tools for processing and analyzing data generated by IoT devices in real-time. It encompasses various data processing techniques, including data cleansing, data transformation, and data enrichment, as well as analytical methods such as machine learning, predictive analytics, and anomaly detection. One of the challenges associated with processing IoT data is the sheer volume of data generated by IoT devices. This data can be structured or unstructured, and it may be in various formats. IoT analytics tools can help to transform this data into a structured format that can be easily analyzed. Additionally, it can detect patterns and anomalies in the data, which can provide insights into the performance of IoT devices and enable organizations to identify and address potential issues before they become critical. IoT analytics can also help organizations to optimize their IoT infrastructure by providing insights into the performance of devices and the efficiency of the overall system.

4 IoT Analytics

[1] presents an Internet of Things (IoT) architecture that uses the MQTT protocol and Node-RED for monitoring radon gas. The authors propose a system that includes sensors for measuring radon gas levels and a Node-RED-based dashboard for visualization and monitoring. The MQTT protocol is used for communication between the sensors and the dashboard, enabling real-time data transmission and analysis. The paper presents a case study of the system in action, demonstrating its effectiveness in monitoring radon gas levels in a building. The results show that the system can provide accurate and timely information about radon gas levels, which can be used for decision-making and risk assessment purposes.

The authors in [2] proposed an efficient approach to orchestrating Internet of Things (IoT) workflows using a Vector Symbolic Architecture (VSA) in Node-RED. The authors describe how IoT workflows are often complex and require significant computational resources to manage, particularly when dealing with large amounts of data. They argue that the VSA approach can help to reduce the computational overhead of these workflows by providing a more efficient way of managing data. The paper describes the design and implementation of a VSA-based system for orchestrating IoT workflows in Node-RED and evaluates its performance using a series of experiments. The results of the experiments show that the VSA-based approach can reduce the computational overhead of IoT workflows while maintaining their functionality.

[3] proposed a Supervisory Control and Data Acquisition (SCADA) approach in Node-RED for monitoring and controlling industrial processes. The authors argue that traditional SCADA systems have limitations, such as high costs and inflexibility, and that Node-RED provides a more flexible and cost-effective platform for implementing SCADA systems. They describe the architecture and implementation of their SCADA system in Node-RED and demonstrate its effectiveness using a case study involving the monitoring and control of a water treatment plant. The results show that the system can provide real-time monitoring of the water treatment process and allows operators to make informed decisions about how to optimize the process.

The authors in [4] present a low-cost SCADA system based on ESP32, Raspberry Pi, Node-Red, and MQTT protocol for monitoring and controlling industrial processes. The authors argue that traditional SCADA systems are expensive and complex and that their proposed system provides an affordable and user-friendly alternative for small-scale industrial applications. The authors describe the hardware and software components of their system and provide a case study of its implementation in a water treatment plant. The results show that the system can provide real-time monitoring of various parameters, such as pH, temperature, and turbidity, and allows for remote control of the plant's pumps and valves.

[5] presents an IoT-based SCADA system for remote oil facilities, utilizing open-source technologies such as Node-RED and Arduino microcontrollers. The proposed system aims to monitor and control various aspects of the oil production process, including flow rates, temperatures, and pressure. The system utilizes wireless sensor networks for data acquisition and transmission, while the Node-RED platform provides a flexible and easy-to-use interface for data processing and visualization. The Arduino microcontrollers are used to interface with the sensors and actuators, allowing for real-time control of the oil production process. The authors also discuss the security implications of the proposed system and present some best practices for ensuring the confidentiality, integrity, and availability of the data.

The authors in [6] presented an empirical study that explored the real-time feedback feature of Node-RED for IoT development. The authors argued that real-time feedback could be useful to assess the behavior of IoT applications and, thus, improve their design and implementation. To support their argument, the authors developed a proof-of-concept application that consisted of a temperature sensor, an actuator, and a Node-RED flow that controlled the actuator based on the sensor's readings. The authors then used Node-RED's dashboard feature to visualize the behavior of the flow in real-time and conducted several experiments to evaluate the impact of the real-time feedback on the developers' ability to debug and improve the flow. The results showed that the real-time feedback was effective in identifying issues and improving the flow's behavior, especially for developers with less experience in IoT development.

[7] discusses the development of an IoT-based open-source SCADA system for monitoring photovoltaic (PV) systems. The system is implemented using Node-RED, an open-source visual programming tool for IoT applications, and MQTT, a lightweight messaging protocol for IoT devices. The system allows real-time monitoring of the PV system's performance and generates alerts in case of any anomalies. The authors present the architecture of the system, including the hardware and software components, and discuss its implementation and testing. The results show that the system is effective in monitoring and analyzing the performance of PV systems in real-time.

Tech-Sphere Journal of Pure and Applied Sciences (TSJPAS)
A Subsidiary of Tech-Sphere Multidisciplinary International Journal (TSMIJ)
Vol 1, Issue 1, 2024 Publication Edition
ISSN: 3672-4648

5 Objectives and Contribution

The main objective of this study is to show the capability of node-red in designing and implementing a low-cost yet realistic real-time data processing and visualization IoT platform for near or remote monitoring of sensor devices. The Publish/Subscribe approach of the MQTT protocol is used to send and receive sensor data amongst constrained IoT devices. The processed data is visually displayed on a computer or handheld device such as a smartphone. In the long run, these visualized outcomes can be used to take a well-informed business decision and help upscale production.

6 Proposed System Architecture

Fig 1 presents the proposed architecture of the system. The prototype is built on Raspberry Pi 4, as the host hardware platform. The following hardware devices were also used; ESP32 module, BME280 module, Breadboard, and Jumper wires. The following section details how these components come together to form the proposed prototype system.

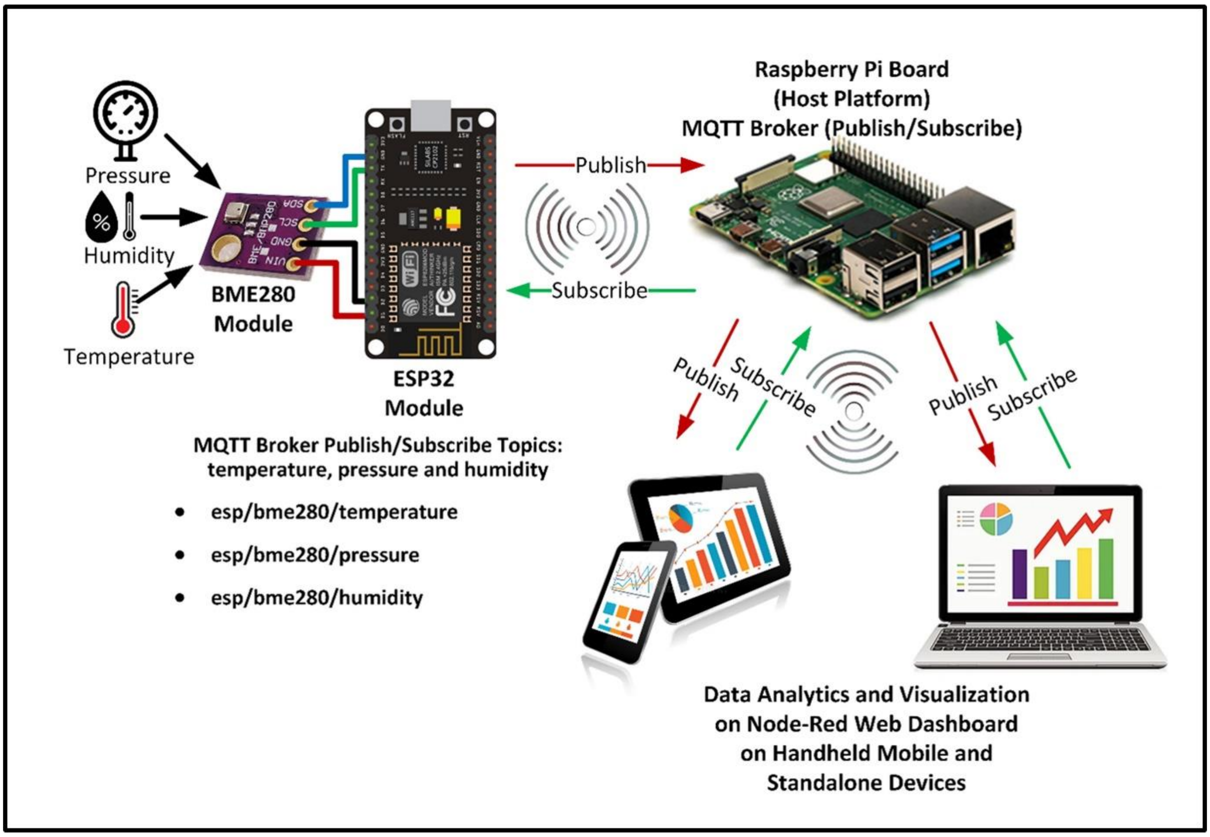

Fig 1: Proposed System Architecture

6.1 System Components and Specifications

Detailed information on the various components used to build the proposed prototype is detailed in this section. The proposed prototype is built as a composite of hardware and software platforms. These platforms are discussed below.

Tech-Sphere Journal of Pure and Applied Sciences (TSJPAS)
A Subsidiary of Tech-Sphere Multidisciplinary International Journal (TSMIJ)
Vol 1, Issue 1, 2024 Publication Edition
ISSN: 3672-4648

6.1.1 Hardware Platform

i. Raspberry Pi 4

Fig 2 shows the photo of the Raspberry Pi board. The Raspberry Pi is a cheap computer that runs on an optimized version of the Linux Operating System called Raspberry OS (formerly called Raspbian). It provides a set of GPIO (general purpose input/output) pins, that make it possible to control electronic components for physical computing and exploration of the Internet of Things (IoT). The Raspberry Pi boards are about the size of a credit card and come with all the basic components of a typical computer, including a processor, memory, and input/output ports, and can be programmed using a variety of programming languages, including Python.

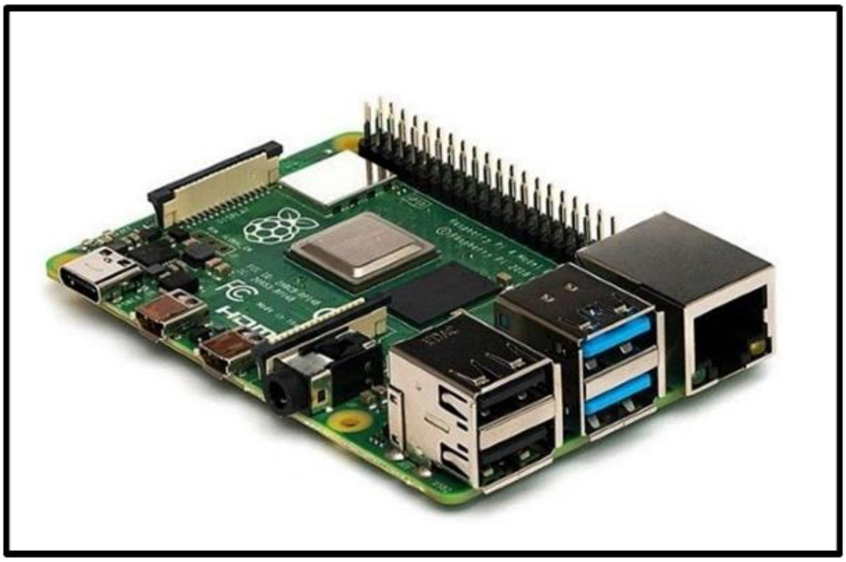

Fig 2: Raspberry Pi 4 Board

Raspberry Pi 4 is the latest version of the Raspberry Pi series of single-board computers, released by the Raspberry Pi Foundation in 2019. It features significant upgrades over its predecessors, including a faster 64-bit quad-core ARM Cortex-A72 CPU, clocked at 1.5GHz, a choice of 1GB, 2GB, 4GB, or 8GB of LPDDR4-3200 SDRAM, Dual-band 802.11ac wireless networking and Bluetooth 5.0, Gigabit Ethernet, two USB 3.0 ports and two USB 2.0 ports, two micro-HDMI ports that support resolutions up to 4Kp60, new USB-C power supply port, and support for dual-display output. These improvements make the Raspberry Pi 4 a much more capable device than its predecessors, with better performance and more connectivity options.

ii. BME280 Sensor Board

The BME280 board is a sensor module that integrates a temperature sensor, a humidity sensor, and a barometric pressure sensor into a single small package. It is designed for use with microcontrollers such as Arduino or Raspberry Pi and can be used for a variety of applications, including weather monitoring, indoor climate control, and environmental sensing. Fig 3 shows the photo of the BME280 sensor module.

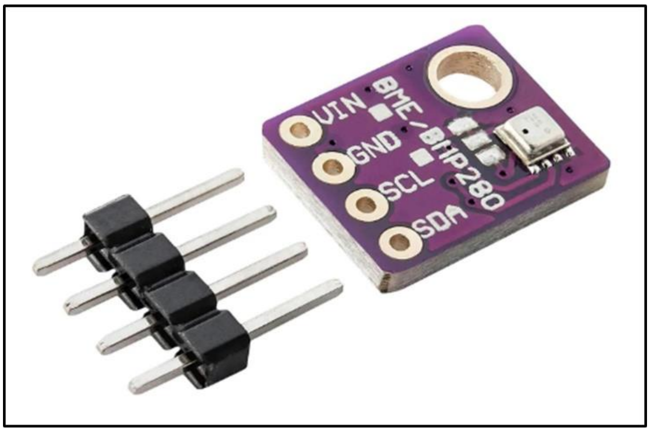

Fig 3: BME280 Sensor Board

iii. ESP32 Module

The ESP32 board is a low-cost, low-power, and highly versatile microcontroller board designed for use in a wide range of projects, particularly those involving the Internet of Things (IoT). It is based on the ESP32 microcontroller chip developed by Espressif Systems, and features built-in Wi-Fi and Bluetooth connectivity, making it ideal for applications that require wireless communication. Fig 4 shows the photo of ESP280 Module.

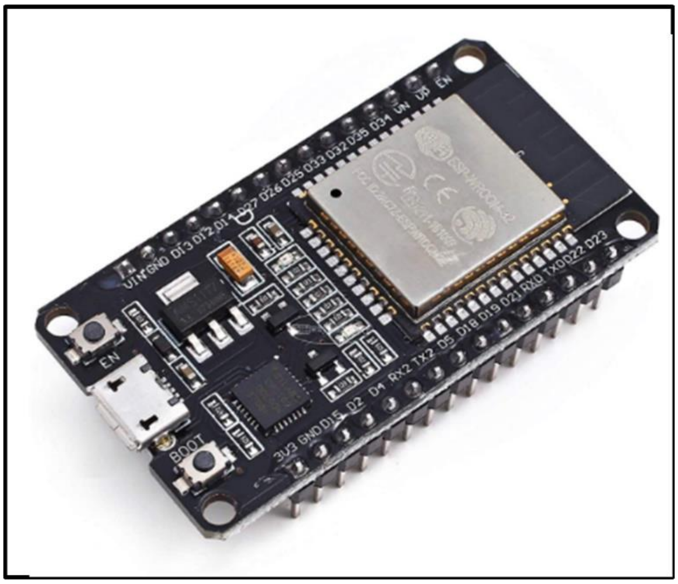

Fig 4: ESP32 Module

The ESP32 board is capable of running at up to 240 MHz and has a range of connectivity options, including Wi-Fi, Bluetooth, and BLE (Bluetooth Low Energy). It also features a wide range of peripheral devices, including ADCs, DACs, SPI, I2C, and UART interfaces, making it compatible with a wide range of sensors and other devices.

iv. Other Passive Hardware

Passive hardware appliances are Breadboards, Jumpers Wires, and LEDs

i. How it Works

A. Detailed explanation of how the prototype IoT platform works is step-wisely described below

- The ESP32 board is interfaced with the BME280 sensor which is used to measure ambient temperature, barometric pressure, and relative humidity. The board connects wirelessly to the Raspberry Pi which acts as a gateway as well as the MQTT message broker/server between sensor devices, most specifically the BME280 module.
- After acquiring the sensor data from BME280, the ESP32 board wirelessly publishes its readings on individual MQTT topics as esp/bme280/temperature is for publishing temperature, esp/bme280/humidity is for publishing humidity, and esp/bme280/pressure is for publishing pressure.
- Via Node-Red, devices such as smartphone and computers wirelessly subscribes to the MQTT broker (Raspberry Pi) to retrieve and display the reading of the BME280 sensor on the Node-Red dashboard in the form of gauges, line charts, and so on.

B. Software Platform

The software platforms used to build the prototype are discussed below:

i. Raspberry OS

Raspberry OS is an operating system (OS) based on the Debian Linux distribution and designed for use with the Raspberry Pi single-board computer (SBC). It is an open-source operating system that provides a user-friendly interface, tools, and applications for the Raspberry Pi board. The Raspberry Pi board is a low-cost computer that was designed to promote the teaching of basic computer science in schools and developing countries. The Raspberry OS is designed to run on ARM-based processors, such as those found on the Raspberry Pi board. It is a lightweight operating system that is optimized for the Raspberry Pi board and can run smoothly on its limited hardware resources. Raspberry OS comes with several pre-installed applications, including a web browser, a programming environment, and multimedia applications.

ii. MQTT Broker

Message Queuing Telemetry Transport (MQTT) is a lightweight messaging protocol designed for use in low-bandwidth, high-latency, or unreliable networks. An MQTT broker is a server that acts as a central hub for exchanging messages between MQTT clients. MQTT clients can be devices, sensors, or applications that communicate with each other using MQTT. An MQTT broker receives messages from MQTT clients and forwards them to other clients based on the topics to which they are subscribed. The broker ensures that messages are delivered to the correct clients and that the message delivery is reliable and efficient. MQTT brokers can be used in various IoT and M2M (Machine-to-Machine) applications, where there are large numbers of devices and sensors that need to communicate with each other. The MQTT protocol's lightweight nature makes it an ideal choice for these applications because it can run on small devices with limited processing power and memory.

iii. Node-Red

Node-RED is an open-source programming tool that allows users to visually create logic flows for Internet of Things (IoT) applications. It is a browser-based flow editor that makes it easy to wire together various devices, APIs, and online services to create IoT applications. Node-RED provides a visual interface to create and manage complex IoT workflows by connecting nodes or blocks that perform specific functions. The nodes can be added, deleted, or modified to create the desired IoT workflow. Node-RED comes with a vast collection of pre-built nodes for popular IoT devices and services, making it easy for developers to create powerful IoT applications without having to write any code.

7 Sensors Setup, Commination, and Node-Red Flows

The sensor module that senses the target environment is the BME280 module. This module senses temperature, pressure, and humidity, while the ESP32 module wirelessly transmits the sensed data across the network through the MQTT Broker for analytical display and visualization through the node-red dashboard on either a smartphone or computer at a near or remote location. The connection between the ESP32 and BME280 is made possible through the ESP32 I2C pins as shown in Table I, while Fig 5 shows the ESP32 and BME280 setup.

TABLE I. **BME280 and ESP32 Pins connection**

BME280	ESP32
Vin	3.3V
GND	GND
SCL	GPIO 22
SDA	GPIO 21

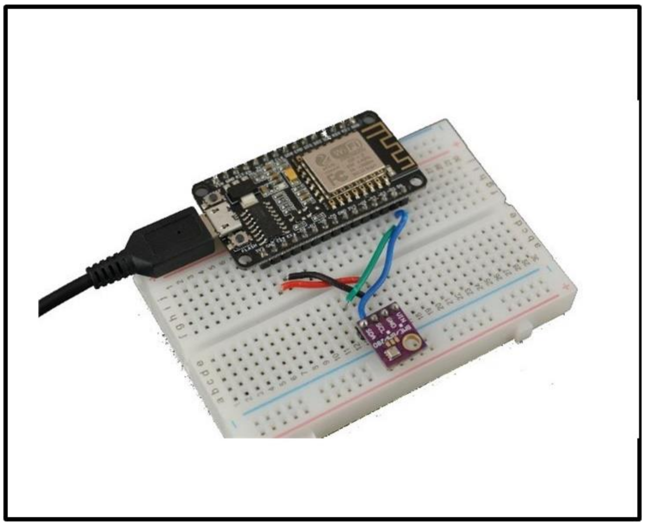

Fig 5: ESP32, BME280, and Breadboard and Jumper Wires Setup Up

The Node-Red flow is shown in Fig G. The MQTT node publishes the sensed temperature, humidity, and pressure using the topics **esp/bme280/temperature** for temperature, **esp/bme280/humidity** for humidity, and **esp/bme280/pressure** for pressure respectively. To read these parameters, the clients such as smartphones and computers will subscribe to the MQTT Broker/server, which is installed on the Raspberry Pi board. Fig 6 shows the photo of Node-Red flows interface for this project.

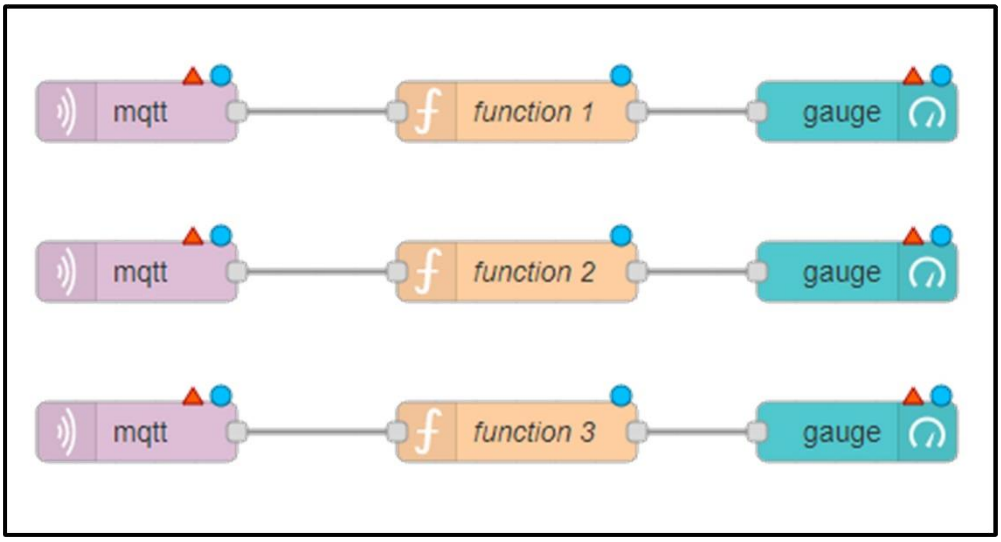

Fig 6: Node-Red Flows Interface

8 Handling MQQT Events

The sensor module that senses the target environment is the BME280 module. This module senses temperature, pressure, and humidity, while the ESP32 module wirelessly transmits the sensed data across the network. Node-Red acts as the MQTT Broker. Below is the description of the steps or stages taken to handle the MQTT events and the JavaScript code that generates and publishes the sensed data.

i. **Install MQTT Broker node in Node-Red**
First, we installed the needed MQTT node package in Node-RED by clicking on the hamburger menu, then selecting "Manage palette." From there, searched for the "mqtt" package and install it.

ii. **Setup and configure MQTT Broker Node**
Once the package is installed, we created an MQTT input node by dragging it onto your Node-RED workspace. The MQTT input node is used to subscribe to the MQTT topic where the sensor data is published. In Node-RED, drag and drop an MQTT input node onto the workspace. Double-click the node to open the configuration dialog box. Double-click on the node to configure it, and enter the MQTT topic that we want to subscribe to. In our case, the topics are; temperature, humidity, and pressure.

iii. **Create a JavaScript Function Node**
The function node is used to process the incoming data from the MQTT input node. Next, we created a function node by dragging it onto the node-red workspace and connecting it to the MQTT input node. Double-clicked on the function node to open the editor, and entered the following code:

```
// Parse the incoming JSON message
var data = JSON.parse(msg.payload);

// Create a new payload with temperature, humidity, and pressure data
msg.payload = {
  temperature: data.temperature,
  humidity: data.humidity,
  pressure: data.pressure
```

```
};

// Return the new payload
return msg;
```

This code extracts the temperature, humidity, and pressure data from the incoming MQTT message payload and creates a new message object with these values. This message object is then passed on to the next node in the flow.

iv. **Setup the Dashboard Node**
The dashboard nodes are used to display the temperature, humidity, and pressure readings on the Node-RED dashboard. We dragged and dropped a dashboard gauge node onto the workspace, and double-click the node to open the configuration dialog box. In the configuration dialog box, for the temperature readings, we entered the following settings:

- Name: Temperature
- Label: Temperature (°C)
- Topic: Temperature
- Units: °C
- Minimum: 0
- Maximum: 40

The above steps are repeated to add the gauge nodes for humidity and pressure while using "humidity" and "pressure" for the topic fields respectively.

v. **Connect the Nodes**
Finally, to view the Temperature, Humidity, and Pressure readings, all three gauges are connected to the function nodes, and the flow is deployed by clicking the Deploy button. The Node-red web interface Dashboard is opened through the Raspberry Pi IP address.

9 Node-Red Web Dashboard

The sensor module that senses the target environment is the BME280 module. The output, which is the sensed data (temperature, pressure, and humidity) are analytically and visually displayed via the node-red web interface dashboard on mobile phones and computers which may be connected to the MQTT broker wirelessly or by wire. In this project, they are connected wirelessly.
Figs 7 and 8 show the Node-Red web interface dashboard display on a desktop, laptop display, or mobile phone interface respectively.

Tech-Sphere Journal of Pure and Applied Sciences (TSJPAS)
A Subsidiary of Tech-Sphere Multidisciplinary International Journal (TSMIJ)
Vol 1, Issue 1, 2024 Publication Edition
ISSN: 3672-4648

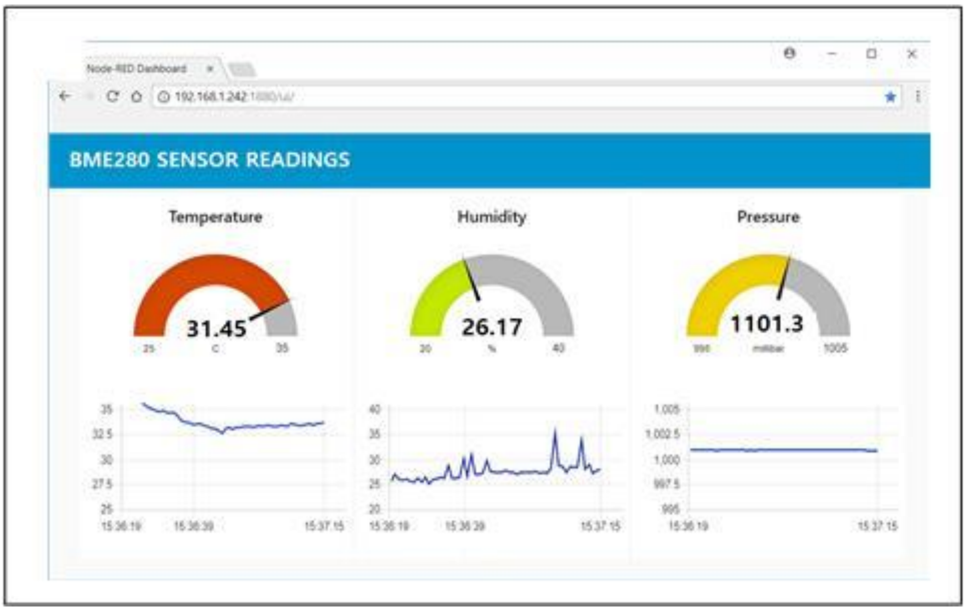

Fig 7: Node-Red Dashboard for Computer Desktop Display

Fig 8: Node-Red Dashboard for Smartphone Desktop Display

10 Discussion

This paper presents an interesting approach to integrating Node-RED and IoT analytics for real-time data processing and visualization. Raspberry Pi 4, esp32, and bme280 sensor modules are used to generate temperature, humidity, and pressure readings, which are then displayed on a smartphone and computer through the Node-RED web interface dashboard. The use of Node-RED as a data processing and visualization tool has become increasingly popular due to its ease of use and flexibility. This paper also demonstrates how Node-RED and IoT analytics can enable real-time data processing and visualization, which can be useful in various applications such as smart home automation, industrial control systems, and environmental monitoring. The use of Raspberry Pi 4 as the main computing platform provides an affordable and powerful option for data processing and visualization. esp32 and bme280 sensor modules were easily integrated with the Raspberry Pi 4 to generate accurate and reliable real-time sensor data. The use of these devices also provides a scalable solution for expanding the sensor network. One of the key strengths of this paper is the demonstration of the Node-RED web interface dashboard, which allows users to visualize sensor data in real-time on a smartphone or computer. It also shows how the dashboard can be easily customized to display various types of sensor data and how it can be used to trigger alerts and notifications based on certain threshold values.

11 Conclusion and Future Work

In this paper, we demonstrated the integration of Node-RED and IoT analytics for real-time data processing and visualization. Raspberry Pi 4, esp32, and bme280 sensor module were used to generate temperature, humidity, and pressure readings, and displayed the data on a smartphone and computer through the Node-RED web interface dashboard. The system provided a user-friendly interface for real-time data visualization and analysis, enabling users to monitor the environmental conditions and make informed decisions based on the data. The integration of Node-RED and IoT analytics provides a powerful platform for real-time data processing and visualization. The study has shown that with limited resources and time a functional IoT appliance can be built and ready to be deployed in any production environment. As a future work, the system could be enhanced by integrating machine learning algorithms for predictive analytics, enabling users to anticipate changes in environmental conditions and take proactive measures. Finally, the scalability of the system could be improved by implementing a cloud-based architecture, enabling the system to handle large volumes of data and support a larger user base.

Competing Interest

The authors declare no competing interest.

Funding

No external funding was provided for this research work.

References

[1] A. Medina-Pérez, D. Sánchez-Rodríguez, and I. Alonso-González, “An Internet of Thing Architecture Based on Message Queuing Telemetry Transport Protocol and Node-RED: A Case Study for Monitoring Radon Gas,” Smart Cities, vol. 4, no. 2, pp. 803–818, May 2021, doi: 10.3390/smartcities4020041.

[2] C. Simpkin, I. Taylor, D. Harborne, G. Bent, A. Preece, and R. K. Ganti, “Efficient orchestration of Node-RED IoT workflows using a Vector Symbolic Architecture,” Futur. Gener. Comput. Syst., vol. 111, pp. 117–131, Oct. 2020, doi: 10.1016/j.future.2020.04.005.

[3] I.-V. Nițulescu and A. Korodi, “Supervisory Control and Data Acquisition Approach in Node-RED: Application and Discussions,” IoT, vol. 1, no. 1, pp. 76–91, Aug. 2020, doi: 10.3390/iot1010005.

[4] A. Zare and M. T. Iqbal, “Low-Cost ESP32, Raspberry Pi, Node-Red, and MQTT Protocol Based SCADA System,” in 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Sep. 2020, no. November 2021, pp. 1–5. doi: 10.1109/IEMTRONICS51293.2020.9216412.

[5] C. A. Osaretin, M. Zamanlou, M. T. Iqbal, and S. Butt, “Open Source IoT-Based SCADA System for Remote Oil Facilities Using Node-RED and Arduino Microcontrollers,” in 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Nov. 2020, pp. 0571–0575. doi: 10.1109/IEMCON51383.2020.9284826.

[6] D. Torres, J. P. Dias, A. Restivo, and H. S. Ferreira, “Real-time Feedback in Node-RED for IoT Development: An Empirical Study,” in 2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT), Sep. 2020, pp. 1–8. doi: 10.1109/DS-RT50469.2020.9213544.

[7] L. O. Aghenta and M. T. Iqbal, “Development of an IoT Based Open Source SCADA System for PV System Monitoring,” in 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), May 2019, no. November 2021, pp. 1–4. doi: 10.1109/CCECE.2019.8861827.