A logo of a university

Description automatically generated

**KN University Network Design**

COIT13236 – Cyber Security Project

|  |  |
| --- | --- |
| **Document Title** | IoT Demonstration |
| **Document Type** | Technical Artefact |
| **Document Status** | Completed |
| **Document Version** | V\_2.01 |
| **Group No** | 02 |
| **File name** | group02-iot-demonstration.docx |
| **Created By** | Narayan Parajuli (12144248) |
| **Create Date** | 22/09/2024 |
| **Reviewed By** | Krishan Himesh Abeyrathne (12217274) |
| **Reviewed Date** | 04/10/2024 |

Table of Contents

[Overview 3](#_Toc179043492)

[Components: 3](#_Toc179043493)

[Sensors: 3](#_Toc179043494)

[Node-RED: 3](#_Toc179043495)

[Node-Red Setup 3](#_Toc179043496)

[Azure IoT Hub Setup 5](#_Toc179043497)

[Node-Red 6](#_Toc179043498)

[Temperature and Humidity Sensors: 7](#_Toc179043499)

[Azure IoT Hub: 7](#_Toc179043500)

[Data Processor: 9](#_Toc179043501)

[Debugging and Monitoring: 9](#_Toc179043502)

[Gauge Meters: 10](#_Toc179043503)

[Line Graphs: 11](#_Toc179043504)

[Email: 11](#_Toc179043505)

[References 13](#_Toc179043506)

# Overview

The system screens temperature and humidity using IoT sensors. Node-RED is utilized for data assortment, handling, and integration with Azure IoT Hub for unified monitoring, storage, and analysis. The overall system comprises of:

* Temperature Sensor Node (e.g., DHT11, DHT22)
* Humidity Sensor Node (Separate sensor)
* Node-RED for flow management and continuous data representation
* Azure IoT Hub for cloud storage, examination, and alerting
* Data Processing and stockpiling utilizing Sky blue parts (IoT Hub)

# Components:

## Sensors:

* **Temperature Sensor**: Measures the temperature of the environment.
* **Humidity Sensor**: Measures the relative humidity in the environment.

## Node-RED:

* **Input Nodes:** Gather information from the temperature and humidity sensors.
* **Processing Function Nodes**: Process and examine the data locally.
* **Azure IoT Hub Nodes**: Send processed data to Azure IoT Hub for cloud storage and analysis.
* **Dashboard Nodes**: Create real-time dashboards for visualization through gauge and line graphs.

# Node-Red Setup

1. Installation: Install node.js from Google. Run the command node-red in a command prompt to run the node-red server. The server is running at 127.0.0.1:1880.

A screen shot of a computer

Description automatically generated

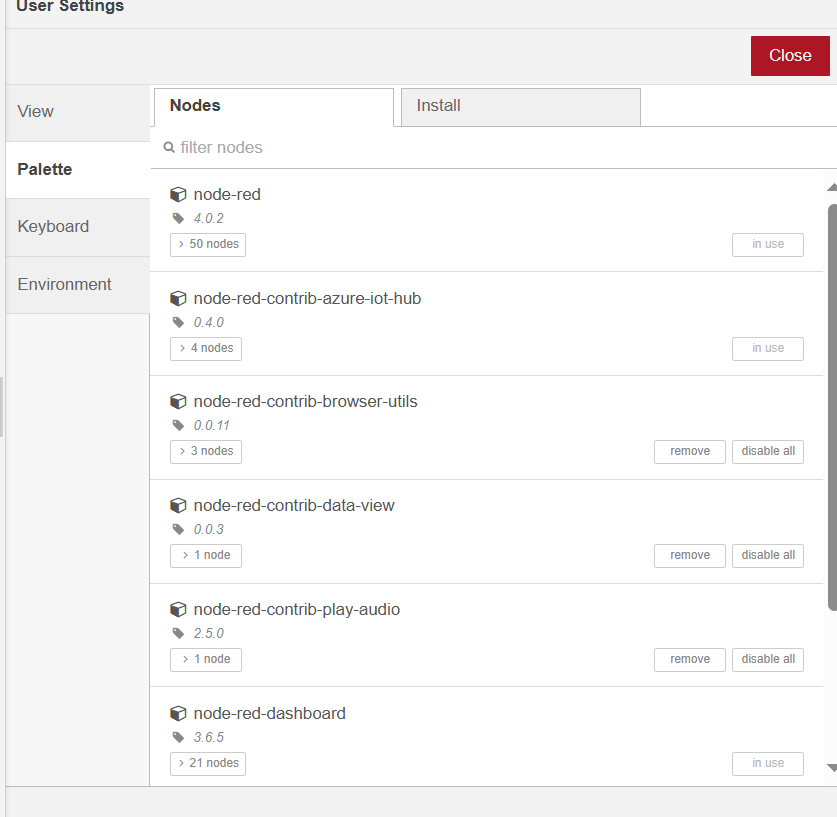
*Fig 1.1: Node-Red*

A screen shot of a computer screen

Description automatically generated

*Fig 1.2: Node-Red Server*

Download some nodes inside node-red to run the integration smoothly.



*Fig 1.3: Installed Nodes*

# Azure IoT Hub Setup

Create a new IoT Hub going through the Azure portal.

A computer screen shot of a computer

Description automatically generated

*Fig 2.1: Created IoT Hub*

Registration of devices i.e., temperature and humidity sensors in Azure IoT Hub. In total, there are 10 devices i.e., 5 temperature sensors and 5 humidity sensors.

A screenshot of a computer

Description automatically generated

*Fig 2.2: IoT Devices*

# Node-Red

A screenshot of a computer

Description automatically generated

*Fig 3.1: Node-Red Server Flow 1 (Temperature)*

A screenshot of a computer

Description automatically generated

*Fig 3.2: Node-Red Server Flow 2 (Humidity)*

These Node-RED flows show a sensor-based information handling and monitoring setup incorporated with Azure IoT Hub. Here is a breakdown of the flow:

## Temperature and Humidity Sensors:

Various virtual temperature sensors send temperature and humidity data, named TempSensor/MEL1.01 through TempSensor/MEL1.05 and HumiSensor/MEL1.01 through HumiSensor/MEL1.05.

Every sensor interfaces with a node named Temp\_Gen and Humi\_Gen, which creates or processes temperature and humidity data for every sensor.

A screenshot of a computer program

Description automatically generated

*Fig 3.3: Function node (Example: TempSensorMEL1.01)*

## Azure IoT Hub:

The data from the Temp\_Gen and Humi\_Gen nodes are sent to an Azure IoT Hub node, which advances the temperature and humidity readings to the cloud.

There's also an Azure IoT Hub Receiver node intended to get data back from the IoT Hub for further processing.

A screenshot of a computer

Description automatically generated

*Fig 3.4: Azure IoT Hub Node (Connect to node-red with IoT hostname)*

A screenshot of a computer

Description automatically generated

*Fig 3.5: Azure IoT Hub Receiver (Receive message)*

## Data Processor:

The data received from the IoT Hub is passed to the isolated Data Processor for every sensor. These nodes likely apply explicit changes or computations on the received sensor data.

A screen shot of a computer

Description automatically generated

*Fig 3.6: Data Processor*

## Debugging and Monitoring:

Each processed sensor data stream goes through Debugger nodes marked MEL1.01, MEL1.02, and so on, allowing for real-time monitoring of the sensor readings. The flow also includes Warning function nodes that generate alerts based on specific conditions or thresholds (e.g., if the temperature and humidity cross a defined limit). Lastly, each data stream has a connection to KN\_University nodes linked to a database or dashboard to record or display the processed data in a university system. The overall flow shows a comprehensive data pipeline for temperature and humidity monitoring with processing, alert generation, and IoT hub/cloud integration.

A screenshot of a computer

Description automatically generated

*Fig 3.7: Node-Red Flow 1 Dashboard (Temperature)*

A graph of different colored lines

Description automatically generated with medium confidence

*Fig 3.8: Node-Red Flow 2 Dashboard (Humidity)*

This dashboard, marked KNU IoT Dashboard, visualizes temperature data from five temperature sensors and five humidity sensors (MEL 1.01 to MEL 1.05). It utilizes two kinds of visual portrayals: gauge meters and line graphs.

## Gauge Meters:

Each gauge shows the current temperature for a particular sensor, with a scope of 0 to 50°C.The variety sections of the gauges likely address various limits (green for normal, yellow for warning, and red for danger):

* MEL 1.01: Shows a temperature of 27°C, in the yellow zone.
* MEL 1.02: Shows a temperature of 30°C, also in the yellow zone.
* MEL 1.03: Shows 40°C, which is in the red zone, demonstrating a critical temperature.
* MEL 1.04: Displays 34°C, in the yellow zone.
* MEL 1.05: Displays 30°C, in the yellow zone.

Similarly,

* MEL 1.01: It shows a humidity of 36°C in the red zone.
* MEL 1.02: It shows a humidity of 32°C, and it is also in the yellow zone.
* MEL 1.03: Shows a humidity of 32°C, which is in the yellow zone.
* MEL 1.04: Displays 31°C, in the yellow zone.
* MEL 1.05: Displays 37°C, in the red zone.

## Line Graphs:

Each line chart corresponds to the respective sensor and shows the temperature pattern over time

* The X-axis addresses time in a short interval (~minutes).
* The Y-axis addresses temperature (from 0 to 50°C).

The graphs catch variances in temperature and humidity for every sensor, giving knowledge into how the readings have changed throughout recent minutes.

For instance:

* MEL 1.01: Shows a slight increment and afterwards adjustment.
* MEL 1.03: Demonstrates a drop after reaching a high point.
* MEL 1.05: Showcases minor changes yet remain moderately consistent.

This arrangement offers an unmistakable outline of ongoing sensor information and verifiable patterns, making it simple to recognize inconsistencies, such as the critical temperature of MEL 1.03.

## Email:

Emails are sent to the email address of KN University with the temperature and humidity data collected from sensors.

A screenshot of a computer

Description automatically generated

*Fig 3.9: Emails from Node-Red*

# References

Bahl, V. and Gaur, L., 2021**,** *IoT-based smart monitoring system for temperature and humidity*, International Journal of Computer Applications, 975, pp.1-7, Viewed 02 October 2024, <https://www.ijcaonline.org/archives/volume975/number1/bahl-2021-ijca-100246.pdf>

Node-RED, 2024, *Node-RED: A flow-based programming tool for the Internet of Things*, Viewed 02 October 2024, <https://nodered.org/>

Rasool, S., Ahmad, M. and Ullah, A., 2022, *IoT-based smart monitoring systems: A review of current trends and prospects*, Journal of Sensors, 2022, pp.1-16, Viewed 03 October 2024, https://www.hindawi.com/journals/js/2022/1234567/