VITAL TRACK

A REAL-TIME BODY MONITORING SYSTEM

DEVICE PROGRAMMING
FOR INTERNET OF THINGS

Submitted by:
Devansh Jain (21ucs060)
Diya Ghodasara (21ucs075)
Lokesh Dandwani (21ucs118)

REPORT DESCRIPTION

IoT has changed the way we interact with technology, making it easier to remotely monitor and handle numerous parts of our life. Healthcare is one area where IoT has had a huge impact. Remote body vital monitoring systems have developed as a viable option for patients who need continuous health monitoring. We will propose a project in this study that intends to create an IoT-based remote body vital monitoring system. The system is intended to monitor several body vitals in real time, such as heart rate, blood pressure, and temperature. The report will provide a thorough overview of the project's objectives, scope, methodology, and implementation. The paper will also evaluate the system's potential benefits and the influence on the healthcare business.

PROBLEM STATEMENT

Monitoring vital body signs, such as heart rate, blood pressure, and temperature, is essential for maintaining good health and preventing illness. However, traditional methods of monitoring these signs are often limited in terms of real-time monitoring, availability of healthcare professionals, and high costs, making it difficult for people to stay on top of their health. There is a need for an efficient and cost-effective remote body vital monitoring system that can provide real-time monitoring of vital signs for any individual, enabling them to receive timely medical attention and take proactive steps to maintain their health.

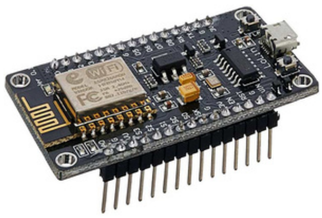
IDEA

This project represents the prototype for the problem statement stated.We have created prototype using very basic body vital such as body temperature. We are also measuring external Environmental temperature to monitor the change in body temperature with changes in external conditions.

OBJECTIVE

- Keep the track of changes in body vitals along with change in external condition.
- Send the data to cloud , so that third person could monitor regularly.
- Plot the data for easy analysis.
- Real time actuation on both sides(user as well as third person) for quick actions in times of emergency.

APPARATUS USED



NODEMCU (ESP8266)

DS18B20 (BODY TEMPERATURE SENSOR)

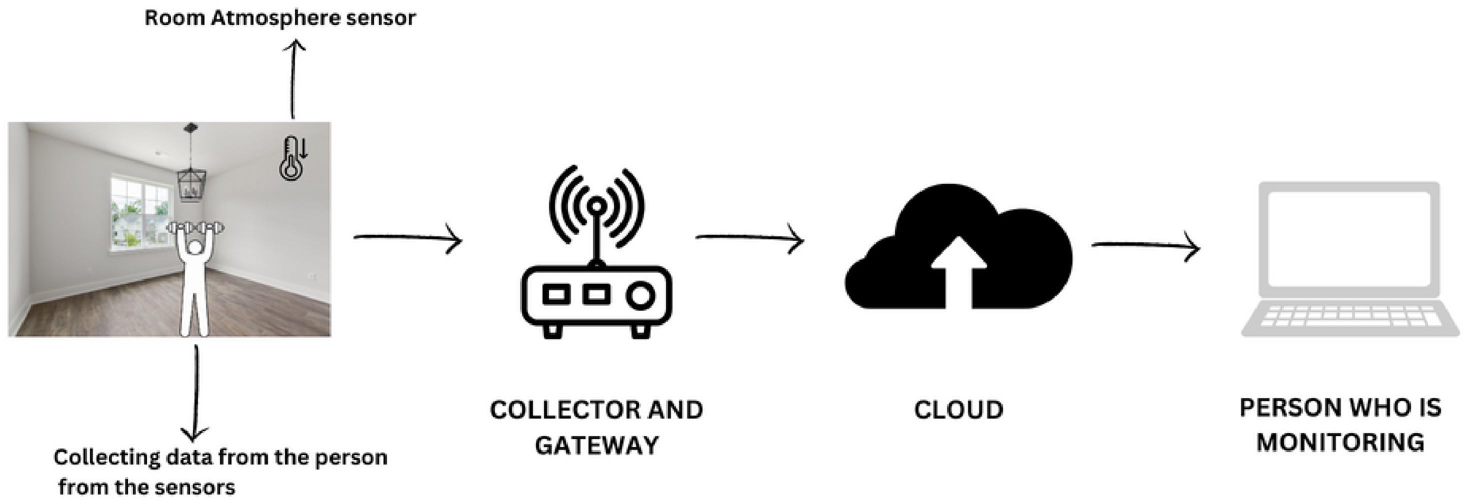
DHT22 (TEMPERATURE AND HUMIDITY SENSOR)

LEDS (ACTUATOR)

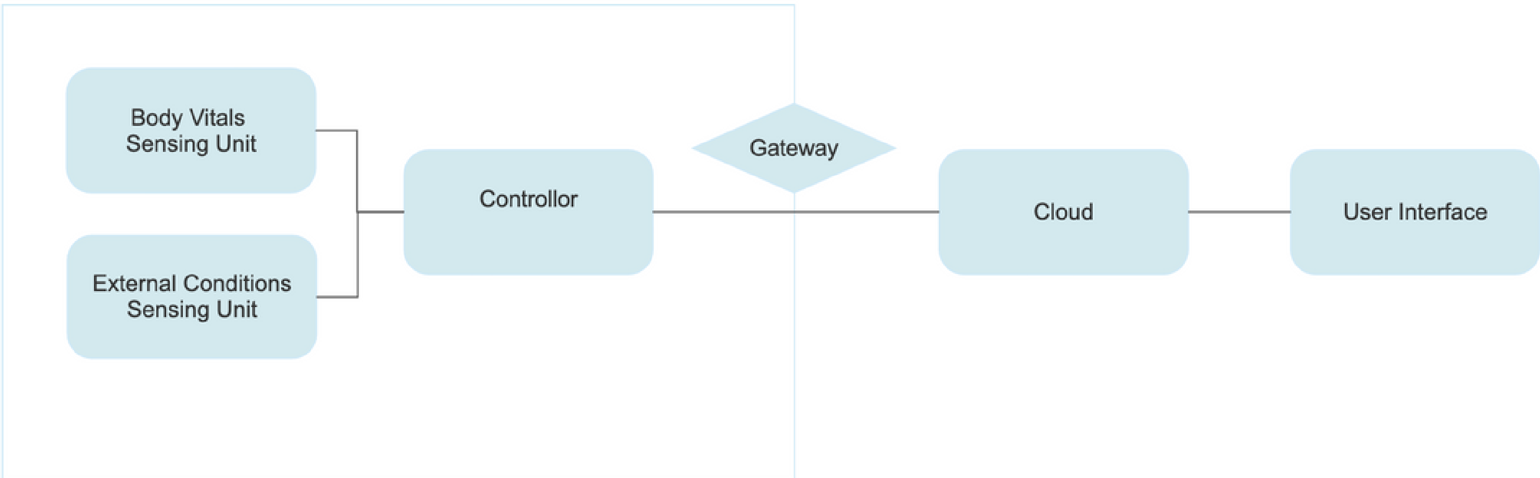
RESISTORS (100 OHMS)

Sensor	Range	Accuracy
DHT22	-40 to 125 C	+- 0.5 degrees
DS18B20	-55 to 125 C	+- 0.5 degrees

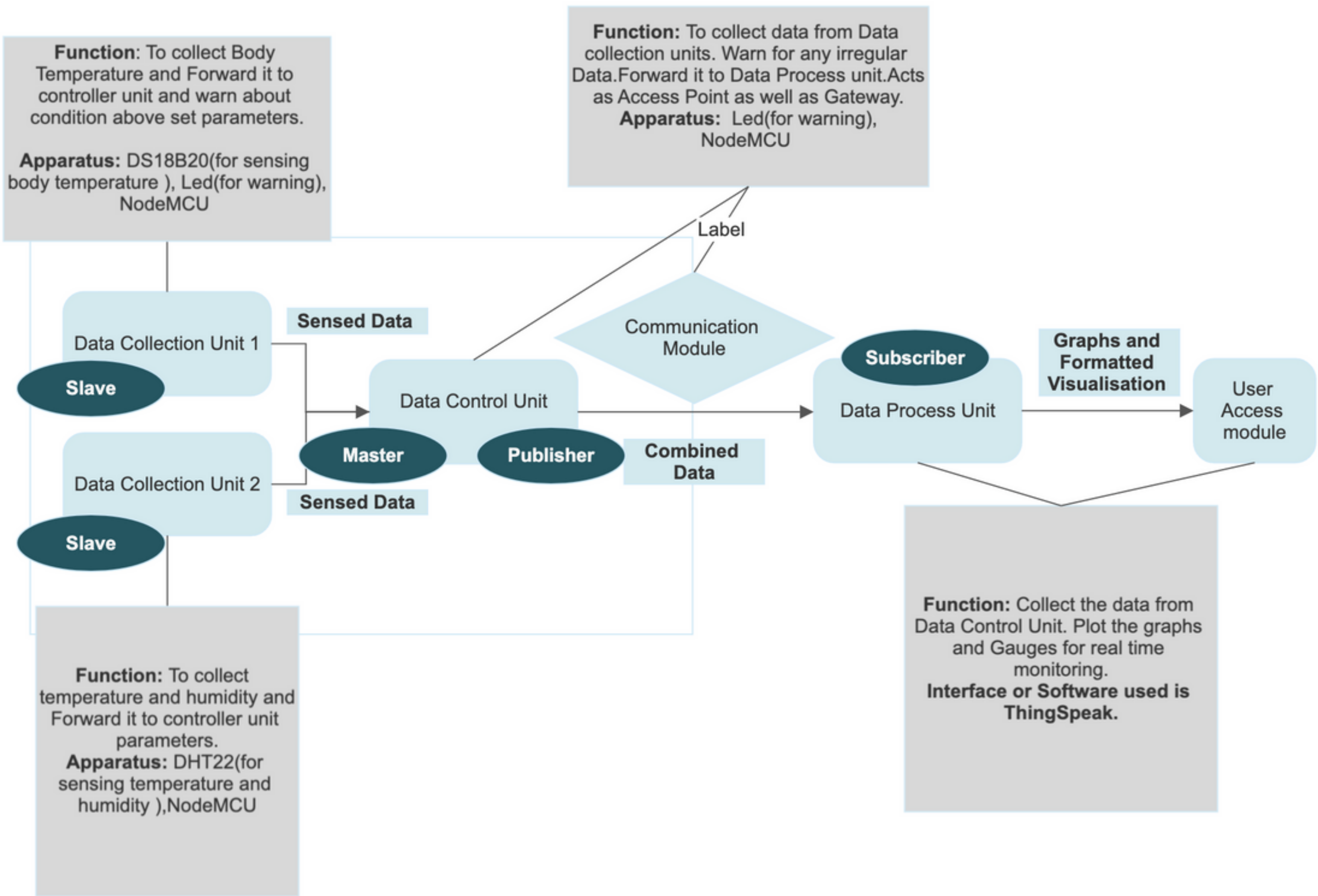
SCENARIO DIAGRAM



BLOCK DIAGRAM



DATA FLOW DIAGRAM



08 ALGORITHM AND PROTOCOLS

SLAVE -1 MODULE

- The code is an ESP8266 program that reads data from a DHT22 temperature and humidity sensor, sends the data using the ESP-NOW protocol to a receiver with a specified MAC address, and then waits for 5 seconds before repeating the process.
- The program first connects to a Wi-Fi network using the provided SSID and password. The struct_message data structure is defined to hold the temperature and humidity data.
- The OnDataSent callback function is registered to be executed when the data is successfully sent to the receiver.
- The ESP_NOW_ROLE_CONTROLLER is set, and the broadcast address of the receiver is added.
- Finally, the DHT22 sensor is initialized, and the loop() function reads the temperature and humidity data, sends the data to the receiver, and waits for 5 seconds before repeating the process.

SLAVE -2 MODULE

- The above code is an ESP8266 firmware program which collects temperature data from a DS18B20 sensor and sends it to a receiver via ESP-NOW protocol.
- The code connects to Wi-Fi, initializes ESP-NOW, registers a callback function, and adds the broadcast address of the receiver.
- The temperature data is read from the DS18B20 sensor, stored in a struct, and sent to the receiver using the esp_now_send() function.
- If the temperature exceeds 35°C, a LED connected to the ESP8266 is turned on.
- Overall, the code demonstrates how to use ESP8266 with a temperature sensor and ESP-NOW protocol to transmit data wirelessly to a receiver.

09

ALGORITHM AND PROTOCOLS

MASTER MODULE

This code is for an ESP8266 microcontroller board to receive data from two different sensors (DHT22 and DS18B20) using the ESP-NOW protocol and send it to ThingSpeak, an IoT platform for data analysis.

The code includes libraries for ESP8266WiFi, ESP8266WebServer, ESP-NOW, and HTTPClient. It defines the network credentials and the MAC addresses of the senders. Two structures are created to hold the received data from each sender. The received data is checked to identify the sender using the OnDataRecv1() function.

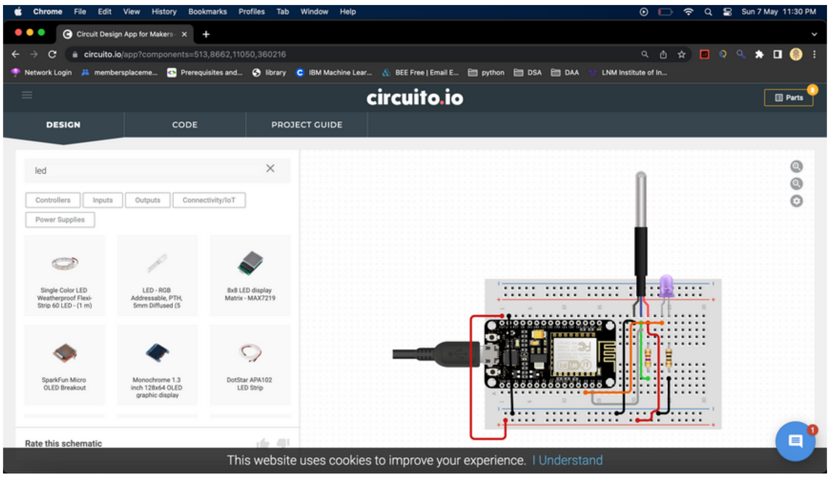
The setup function initializes the ESP-NOW, sets the role of the microcontroller board as a controller, adds the senders as slaves, and registers the callback function for receiving data. It also starts the web server for ThingSpeak communication.

The loop function waits for data to be received from both sensors and sends it to ThingSpeak. The data is first checked to identify the sender and then printed on the Serial Monitor and sent to ThingSpeak using HTTP GET requests.

CIRCUIT DIAGRAM

RECIEVER NODE

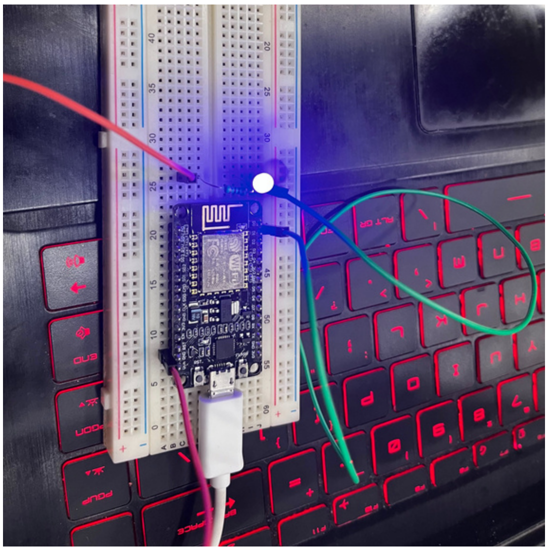
SENDER NODE

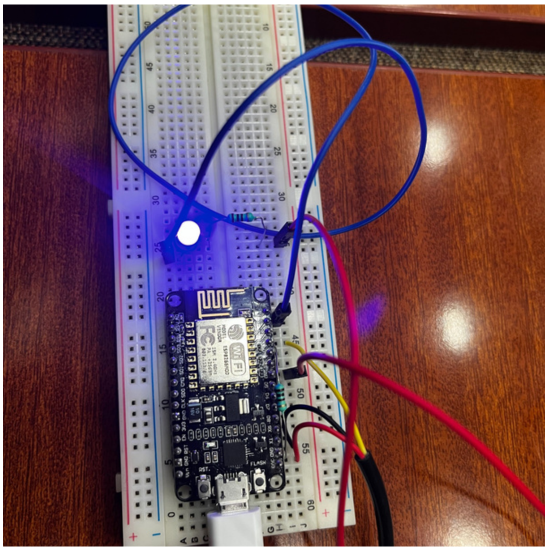


SENDER NODE

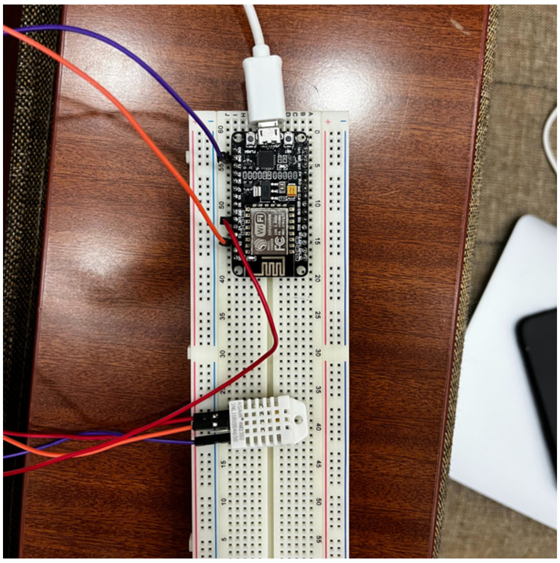
11 CIRCUITS

RECIVER NODE WHEN THRESHOLD HAS BEEN CROSSED

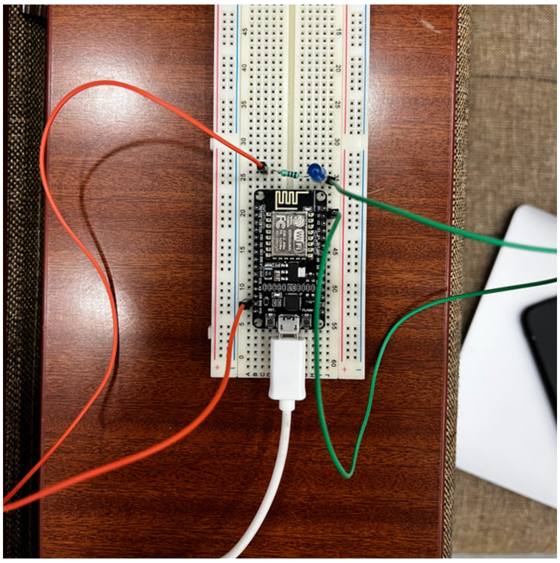




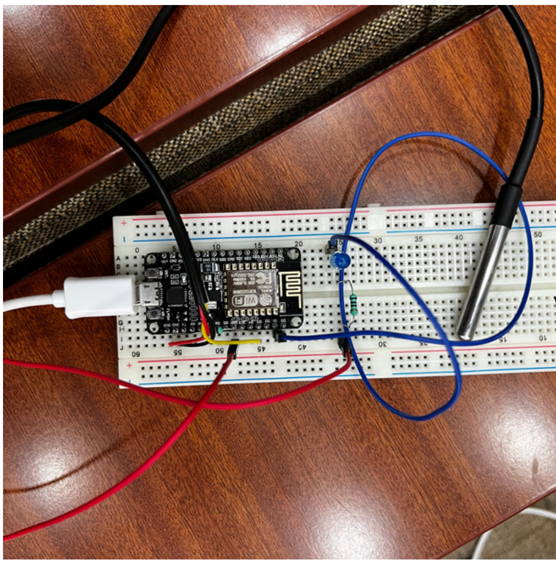
SENDER NODE WHEN THRESHOLD HAS BEEN CROSSED



SENDER NODE WITH DHT22

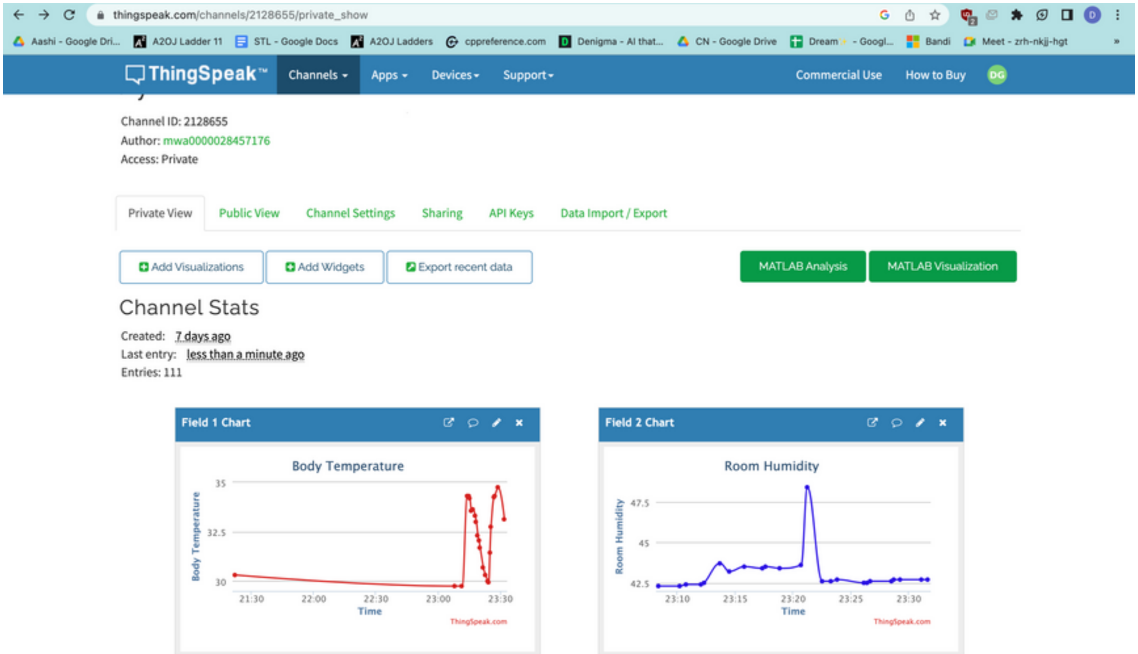


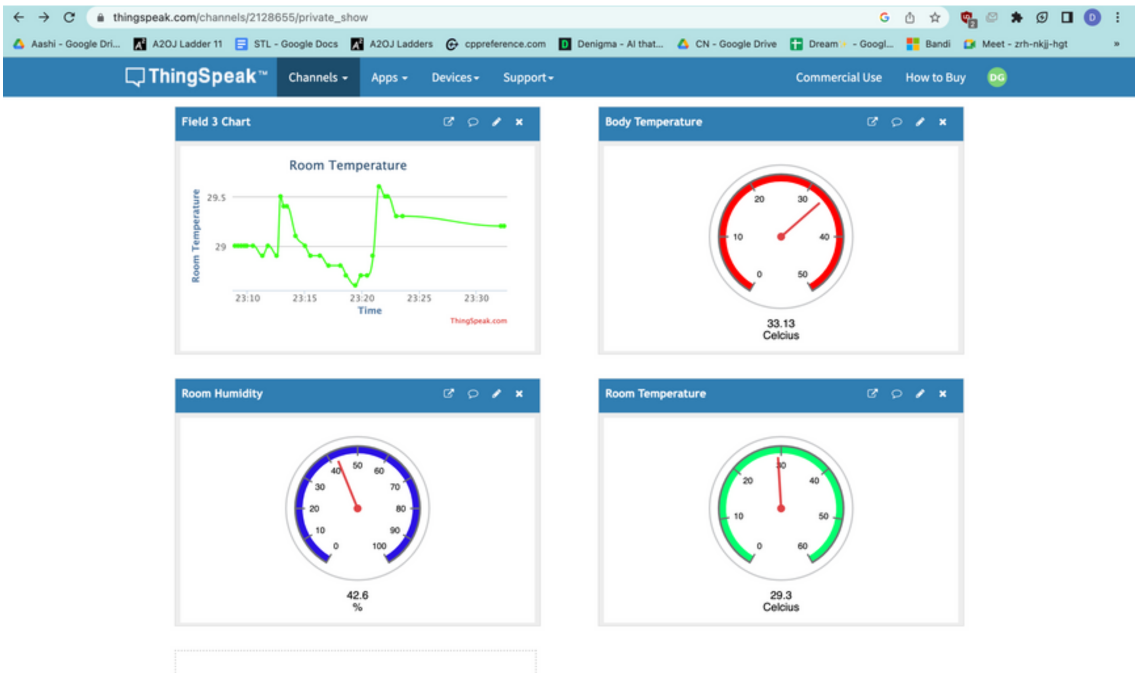
RECIVER NODE



SENDER NODE WITH DS18B20

RESULTS


thingspeak.com/channels/2128655/private_show
ThingSpeak™
Channels
Apps
Devices
Support
Commercial Use
How to Buy
Channel ID: 2128655
Author: mwa0000028457176
Access: Private
Private View
Public View
Channel Settings
Sharing
API Keys
Data Import / Export
Add Visualizations
Add Widgets
Export recent data
MATLAB Analysis
MATLAB Visualization
Channel Stats
Created: 7 days ago
Last entry: less than a minute ago
Entries: 111
Field 1 Chart
Body Temperature
Time
Field 2 Chart
Room Humidity
Time


thingspeak.com/channels/2128655/private_show
ThingSpeak™
Field 3 Chart
Room Temperature
Time
Body Temperature
33.13
Celcius
Room Humidity
42.6
%
Room Temperature
29.3
Celcius

CONCLUSIONS

Multiple models can be created by extending this prototype, allowing for the addition of various health monitoring sensors (such as heartbeat, spo2, blood glucose level, blood pressure, and others) in a body area network. Different combinations of these sensors can be utilized to create multiple models that cater to specific needs.

Few of them are:

- The model can be utilized in the room of individuals who require constant health monitoring, and the information gathered can provide valuable insights into the impact of specific medications on the individual.

- The application of this model can be broadened to individuals who work in risky environments such as mines, iron factories, etc., where external factors may occasionally be severe. As a result, this could diminish the number of deaths caused by insufficient oxygen or extremely high temperatures.

- This model has the potential to be applied in gymnasiums where a sudden increase in heart rate could potentially result in a fatal outcome.

BIBLIOGRAPHY

- https://randomnerdtutorials.com/esp-now-many-to-one-esp8266-nodemcu/
- https://www.arduino.cc/reference/en/libraries/
- https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html
- https://forum.arduino.cc/t/esp-now-data-is-sent-but-doesn-t-show-up-on-the-receiver/940799