Title:

4115-Environmental Monitoring:

Proj_227836_Team_2:

S. Abinash(au411521104003)

A. Abdul majith(au411521104001)

R. Gokul(au4115211040037)

D. Gokul(au411521104036)

1.Description

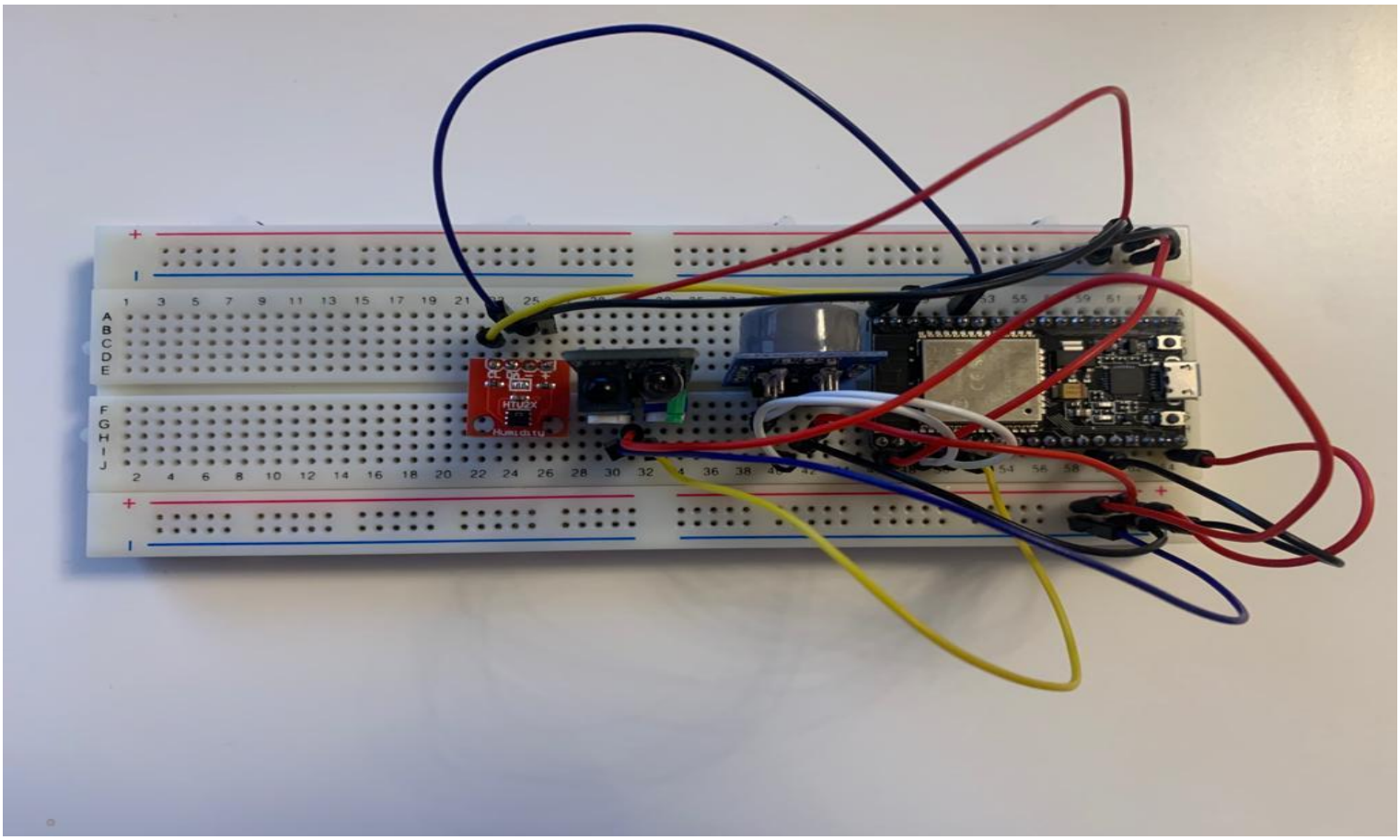
The purpose of the project is to develop an Environment Monitor using three different sensors In this project I used a temperature and humidity sensor, an gas sensor, a IR sensor and ESP32-WROOM Dev Board which is storing the data from the sensor and holds the webserver. The temperature and humidity are displayed on a web server held by the ESP32, which establishes the connection with the router inside the room, and when a gas leak is detected or a movement in front of the IR sensor an email is generated.

2. Hardware Description

The components used in this project are:

- ESP-WROOM-32
- HTU21D Temperature&Humidity Sensor
- IR Sensor

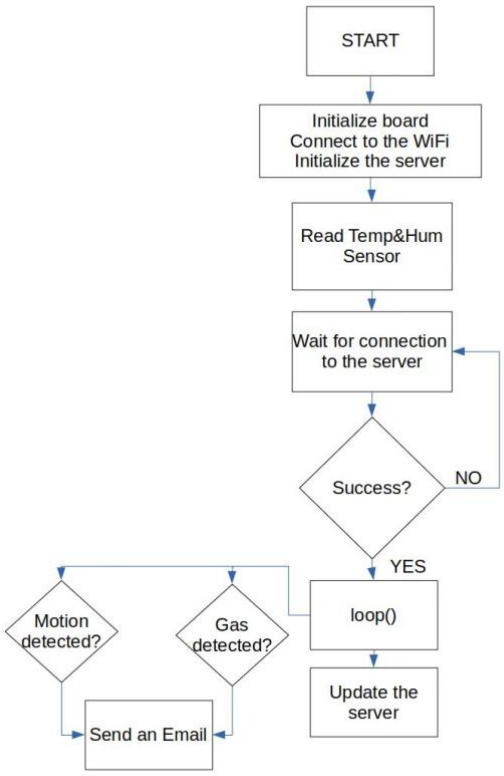
- MQ-9 Gas Sensor
- Micro-USB cable
- Jumper wire



3. Software Implementation

The code begin with initialization of the the modules used in the project. First, the HTU21D sensor is initialized and the pins used for the rest of the sensor. Following that the function which handle the WiFi connection is called. Once the connection is establish, the files used for the server is upload from the flash memory of the ESP32 (The files was uploaded using SPIFFS library). Next step is the infinite loop where the data from the sensor are being read. The update of the server is made once at every 10 seconds. If the IR sensor detect a movement

or the gas sensor detect a high value of the gas in the room, the module will send a request to the IFTTT, which will trigger an email.



4. Connectivity:

1.BLE (Bluetooth Low Energy):

Energy Efficiency:

BLE is designed for low power consumption, making it ideal for battery-powered devices like fitness trackers and smartwatches.

Short Range:

It's typically used for communication within a range of a few meters.

Data Rate:

Suitable for low to moderate data transfer rates.

Applications:

Commonly used in wearable technology, healthcare devices, and smartphone accessories.

2. WiFi:

High Data Rates:

WiFi provides high-speed data transfer, making it suitable for internet access and multimedia streaming.

Consumption:

It consumes more power compared to BLE but less compared to some other wireless technologies.

Medium Range:

Offers a range that can cover a home or office.

Applications:

Used for internet access, smart home devices, and most wireless networking applications.

3. Zigbee:

Low Power and Low Data Rate:

Zigbee is designed for low data rate applications with a focus on low power consumption.

Mesh Networking:

Zigbee supports mesh networking, allowing devices to relay data over longer distances.

Medium Range:

It offers a range that is suitable for home automation and industrial applications.

Applications:

Commonly used in smart homes for lighting control, HVAC systems, and industrial automation.

5. Beepceptor :

Mock API Endpoints:

Create fake or simulated API endpoints that mimic the behavior of real APIs. This is useful for testing applications when the actual API is not available or fully implemented yet.

Record and Inspect API Requests:

Beeceptor can capture incoming API requests, allowing developers to inspect the data being sent to the mock API. This helps in understanding how the application interacts with external APIs.

Simulate Responses:

Developers can configure Beeceptor to generate specific responses to API requests. This includes defining response data, status codes, headers, and more, which aids in testing various scenarios and error handling.

Webhooks Testing:

Beeceptor can be used to test webhooks by providing a URL for incoming webhook requests. Developers can then examine the webhook payloads and responses.

Logging and Monitoring:

Beeceptor offers logging and monitoring capabilities, allowing developers to track and analyze API interactions and identify any issues or bottlenecks.

6. Protocol:

1.MQTT (Message Queuing Telemetry Transport):

Publish-Subscribe Model: MQTT follows a publish-subscribe communication model. Clients publish messages to topics, and other clients subscribe to specific topics to receive those messages.

Lightweight and Low Overhead: It is designed to be lightweight and efficient, making it suitable for IoT and low-bandwidth, high-latency, or unreliable networks.

Quality of Service (QoS): MQTT supports different QoS levels for message delivery, from at most once (0) to at least once (1) and exactly once (2).

Broker-Based: MQTT typically uses a message broker to route messages between publishers and subscribers.

IoT and M2M Applications: MQTT is widely used in the Internet of Things (IoT) and Machine-to-Machine (M2M) applications.

2. HTTP (Hypertext Transfer Protocol):

Request-Response Model: HTTP is a request-response protocol where a client sends a request to a server, and the server responds with data.

Human-Centric: It is designed for human-readable content and is commonly used for web applications, RESTful APIs, and websites.

Stateless: HTTP is stateless, meaning each request from a client to a server must contain all the information needed to understand and fulfill the request.

Widely Used: HTTP is the foundation of the World Wide Web and is used for web browsing, API communication, and data transfer on the internet.

3. AMQP (Advanced Message Queuing Protocol):

Message Queuing Protocol: AMQP is a messaging protocol that focuses on reliable and efficient message queuing between applications or services.

Message Brokers: It's often used with message brokers like RabbitMQ and Apache ActiveMQ.

Queues and Topics: AMQP supports message queues and topics, allowing for flexible message routing and delivery.

Enterprise Applications: AMQP is commonly used in enterprise messaging systems, financial services, and scenarios requiring guaranteed message delivery.

7. Public platform:

Features:

.Data Sharing and Visualization: Provide the public with access to real-time or historical environmental data through interactive maps, charts, and dashboards. Users can explore air quality, water quality, temperature, and other data relevant to their region.

.Crowdsourced Data Collection: Allow users to contribute their own environmental observations or measurements, such as reporting pollution, wildlife sightings, or weather conditions. This crowdsourced data can supplement official monitoring efforts.

.Educational Resources: Offer educational content, articles, videos, or courses on environmental topics to help users understand the issues and solutions better. This can include resources for students and educators.

.Community Forums and Discussions: Create online forums or discussion boards where users can share ideas, ask questions, and collaborate on local environmental initiatives. These forums can be organized by location or topic.

.Alerts and Notifications: Implement alert systems to notify users about significant environmental events, such as air quality alerts, weather warnings, or ecological changes in their area.

.Mobile Apps: Develop mobile apps that allow users to easily access environmental data, report incidents, and receive alerts while on the go. These apps can encourage regular engagement.

.APIs for Developers: Provide APIs to encourage developers to build applications or tools that leverage the platform's environmental data. This can foster innovation and expand the platform's capabilities.

.Open Data Access: Make environmental data accessible in open formats and under open licenses, allowing researchers, NGOs, and government agencies to use the data for research and policy-making.

.Community Projects and Challenges: Organize environmental initiatives, challenges, or projects that encourage users to take concrete actions to improve their environment, such as tree planting or waste reduction campaigns.

.Social Media Integration: Allow users to share their environmental activities and findings on social media platforms to raise awareness and encourage others to get involved.

.Reporting and Feedback: Enable users to report environmental issues directly to relevant authorities or organizations through the platform, facilitating quicker response to problems.

.Partnerships and Collaborations: Partner with environmental organizations, research institutions, and government agencies to enhance the credibility and effectiveness of the platform.

8. Conclusions:

The purpose of this project was to implement a system which is capable of collecting data from multiple sensors from an indoor room, to display the data collected on a web server and to warning the user in case of a motion or a gas detection. The project can be improved by adding more sensors, a microphone, the capability to send SMS. The project help me to understand how to run a web server on the ESP32, how to trigger an email using IFTTT