IoT Driven Solutions for Environmental Hazard Detection and Health Monitoring and Production

A Report on Industrial Internship submitted in partial fulfilment of the requirements for the degree of Bachelor of Technology

By

Raihan Ahmed Sadiyal **212010014021**

Nilabrho Kanta Paul **212010014017**

BARAK VALLEY ENGINEERING COLLEGE

(Affiliated to ASTU, Approved by AICTE)

Nirala, Karimganj, Assam-788701

August 2023

ACKNOWLEDGEMENT

We are pleased to present this report on the Industrial Internship undertaken during our B.Tech 7th Semester. This document encapsulates our experiences, learning, and project outcomes achieved throughout the internship period.

We extend our sincere gratitude to our Principal, Dr. Diganta Goswami, of Barak Valley Engineering College, for his support and encouragement throughout this internship. His commitment has provided us with the resources and motivation needed to make the most of this opportunity. His guidance has been instrumental in supporting our professional growth during this project.

We are also deeply thankful to our Head of Department,Department Of ETE, Prof. Nabarun Chakraborty, Department of Electronics and Telecommunication Engineering, and to our Internship Coordinator, Mrs. Pallavi Das, for their dedicated guidance and unwavering support throughout the duration of our internship.

Our heartfelt thanks go to XCONICS for providing a valuable platform to gain industry-relevant insights, and we are especially grateful to our project guide, Mr. Avik Basu, for his mentorship and expertise, which greatly enriched our learning experience. Lastly, we appreciate the contribution of all faculty members who assisted us along the way, and we thank our parents and all others who supported us in completing this project.

Date- _____________

ABSTRACT

This project leverages IoT technology with ESP32 microcontrollers to monitor environmental hazards by detecting temperature, humidity, and carbon monoxide (CO) levels. The system is structured as a client-server model with two main components: the ESP_SERVER and ESP_CLIENT. The ESP_SERVER collects data using sensors like the DHT11 for temperature and humidity and the MQ2 for CO, displaying it locally on a Nokia 5110 LCD for immediate monitoring. It also hosts a web server for remote data access and uses Bluetooth for configuring thresholds and sending alerts. To enhance safety, the Pushover notification feature ensures real-time alerts to relevant personnel.

The system connects to an MQTT broker, allowing the ESP_CLIENT, which subscribes to MQTT topics, to receive updates from the ESP_SERVER. Based on the received data, the ESP_CLIENT controls a relay switch, enabling automated responses—such as activating alarms or ventilation systems when thresholds are breached—providing immediate hazard mitigation.

In addition to environmental monitoring, the project also involved the development of a smartwatch that incorporates an integrated PPG heart rate sensor for continuous heart rate monitoring. This smartwatch is designed to track users' heart rate in real-time, providing valuable health insights that can be crucial for maintaining the well-being of personnel working in high-risk environments. The PPG sensor detects blood volume changes through light absorption, offering a non-invasive and efficient means of monitoring heart rate.. The integration of health monitoring capabilities into the project reflects a comprehensive approach to safety, addressing both environmental hazards and health metrics for personnel

The internship experience also extended to participating in the production process for PCB design in smart shoes, demonstrating practical involvement in diverse IoT applications. This modular IoT solution is adaptable to various industrial applications, enhancing safety, operational efficiency, and scalability in environments requiring rigorous monitoring and smart wearable integration.

Table of Contents

LIST OF FIGURES

LIST OF ABBREVIATIONS

1. **IoT -** Internet of Things
2. **IIoT -** Industrial Internet of Things
3. **MQTT -** Message Queuing Telemetry Transport
4. **PPG** - Photoplethysmogram
5. **PCB -** Printed Circuit Board
6. **DHT11 -** Digital Temperature and Humidity Sensor
7. **Li-Po Battery -** Lithium-ion Polymer battery
8. **OLED Display -** Organic Light-Emitting Diode.

Chapter 1

INTRODUCTION

The Internet of Things (IoT) refers to a network of interconnected devices that can collect, exchange, and act on data through the internet. These devices, often equipped with sensors, software, and other technologies, are capable of connecting and communicating with each other, allowing for seamless data transfer and automation across various applications. IoT systems enable the integration of the physical and digital worlds, providing valuable insights and enhancing the efficiency of numerous processes in industrial, commercial, and residential environments.

In an IoT system, devices such as microcontrollers, sensors, and actuators work together to monitor and control environmental parameters, machinery, and other assets. By collecting data from sensors and transmitting it to central servers or cloud platforms, IoT systems can analyse trends, detect anomalies, and trigger automated responses to optimise operations and ensure safety. The data-driven nature of IoT enables predictive maintenance, remote monitoring, and real-time decision-making, transforming how industries operate and how daily life is managed.

The Internet of Things (IoT) is revolutionising numerous sectors by enabling smarter and more efficient operations. In industrial settings, IoT is used for predictive maintenance, real-time monitoring of equipment, and optimising supply chain operations. Smart cities leverage IoT for traffic management, energy conservation, and improving public safety. In healthcare, IoT devices monitor patient vitals, manage chronic diseases, and enable remote diagnostics, enhancing patient care and reducing healthcare costs. In agriculture, IoT systems monitor soil conditions, weather patterns, and crop health to optimise farming practices and increase yields. The future of IoT promises even greater integration and innovation, with advancements in artificial intelligence and machine learning driving more sophisticated data analytics and autonomous decision-making.

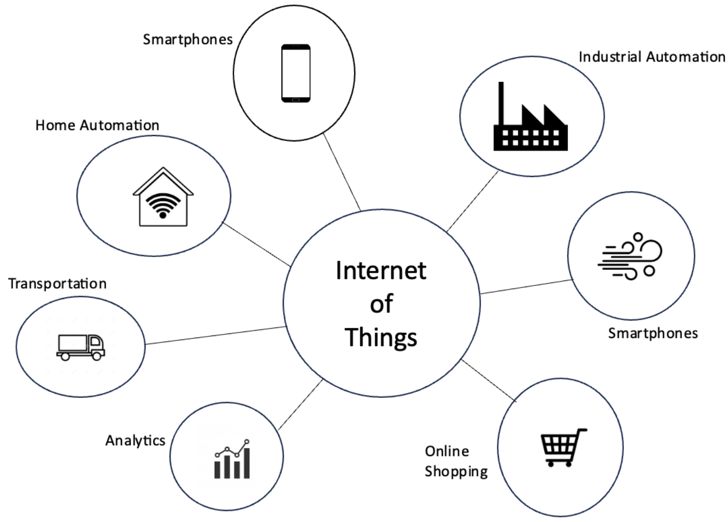

Fig 1.1: Internet of Things (IoT)

1.1 Basic IoT fundamentals

The Internet of Things (IoT) is built upon four foundational pillars: data, devices,analytics, and connectivity. Data serves as the core of IoT, allowing for extensive collection of information from the physical world to fuel insights and decisions. Devices are the physical components that gather this data, capturing environmental or system information through sensors and actuators. Analytics transforms this raw data into actionable insights, making it useful by identifying patterns or anomalies. Connectivity is what enables the sharing of data and insights across networks, creating an interconnected environment where data can be transmitted, processed, and applied in real time, embodying the "internet" aspect of IoT.

1.1.1 Major components of IoT

Key components enable IoT systems to function effectively. Sensors and actuators play a crucial role in IoT by collecting data and performing actions based on that data, such as adjusting temperatures or controlling lights. Connectivity is essential for these devices to send and receive information within a network, typically via WiFi, Bluetooth, or satellite. IoT devices are equipped with computing power to process this data locally, making decisions or triggering responses based on specific inputs. Data storage and processing are often facilitated by cloud-based platforms, which allow IoT systems to store large amounts of data and perform extensive computations. Analytics tools are used to interpret and visualise data, making it easier for users to draw meaningful insights. Due to their internet connectivity, IoT devices are vulnerable to cyber-attacks, making security a critical component of IoT to protect data and devices from unauthorised access and ensure privacy.

1.1.2 How an IoT System Actually Works

An IoT system operates through a series of steps, beginning with data collection from the environment via sensors. These sensors gather environmental information, such as location data from GPS or temperature readings, which forms the basis for all subsequent processing. Once collected, this data is transmitted to the cloud, where it can be further analysed. Connections such as WiFi, LAN, or Bluetooth enable data to reach the cloud where it is processed according to pre-defined algorithms or programs, generating predictions or insights. After processing, the data is made accessible to users through a user interface that displays relevant information in a comprehensible format. This interface allows users to receive alerts, such as notifications about high temperatures or weather conditions, and respond to changes quickly. This complete process—from data collection to user interface—illustrates how IoT devices create a responsive, data-driven ecosystem that can improve decision-making, automation, and overall user experience.

1.2 Introduction to the Industrial Internet of Things (IIoT)

The Industrial Internet of Things (IIoT) is a foundational pillar of modern industrial transformation, embodying the fusion of information technology (IT) and operational technology (OT) within diverse industrial environments. By interconnecting physical assets, machinery, sensors, and software systems, IIoT facilitates real-time data collection, seamless communication, and enhanced automation capabilities. This interconnected ecosystem forms the backbone of Industry 4.0, where data-driven insights lead to streamlined processes, optimised resource usage, and innovative approaches to traditional manufacturing, supply chain, and maintenance practices.

IIoT applications are reshaping industries across sectors such as manufacturing, energy, transportation, healthcare, and agriculture. By integrating machine-to-machine (M2M) communication and industrial data analytics, IIoT solutions enable companies to address long-standing challenges like equipment downtime, productivity inefficiencies, safety risks, and complex operational planning. Below, we outline the key components, benefits, and potential challenges within the IIoT framework.

1.2.1 Advantages of Industrial IIoT

- **Predictive Maintenance**
  Industrial IoT (IIoT) enables a shift from reactive to predictive maintenance, where data from sensors and analytics tools helps companies forecast potential equipment failures before they occur. This approach minimises unexpected downtimes, reduces maintenance expenses, and prolongs equipment lifespan by addressing issues proactively.

- **Enhanced Operational Efficiency**
  IIoT facilitates real-time monitoring of industrial processes, which improves overall operational efficiency. By continuously tracking performance metrics, businesses can spot inefficiencies and optimise resource allocation, leading to reduced production costs and more effective use of assets.
- **Improved Safety and Regulatory Compliance**
  With IIoT, safety standards are heightened through continuous monitoring of environmental conditions and equipment status. Hazardous situations can be detected early, allowing automated systems to trigger alarms or corrective actions. Additionally, real-time monitoring helps businesses meet safety regulations and maintain compliance with industry standards.
- **Informed, Data-Driven Decisions**
  IIoT collects a vast amount of operational data that can be analysed to extract insights for strategic decision-making. Access to accurate, real-time information enables management to make faster, evidence-based decisions that improve outcomes and operational strategy

1.2.2 Challenges of IIoT Adoption

- **Ensuring Data Security and Privacy**
  As more devices become interconnected within IIoT ecosystems, data security becomes a pressing concern. Industries need to protect sensitive data from cyber threats by implementing robust cybersecurity protocols, including data encryption and network monitoring, to ensure data integrity and privacy.
- **Interoperability with Legacy Systems**
  Many industrial facilities rely on older systems that were not designed for interconnectivity, creating interoperability challenges when integrating IIoT. To successfully merge these legacy systems with IIoT infrastructure, customised solutions and industry-standard interoperability frameworks are often required.
- **High Initial Investment**
  Although the long-term benefits of IIoT are significant, the upfront costs for hardware, software, and skilled labour can be high. For many organisations, adopting IIoT requires strategic investment planning and a clear roadmap for return on investment (ROI).
- **Complexity in Data Management and Analysis**
  The extensive data generated by IIoT devices necessitates advanced data management and analytical tools. Organisations may face challenges in effectively processing, storing, and visualising this data if they lack sufficient infrastructure and trained personnel, making it essential to have robust data handling strategies in place.

1.3 Industrial Automation

Industrial automation through IoT and IIoT (Industrial Internet of Things) has revolutionised traditional manufacturing by enabling real-time data collection, analysis, and adaptive decision-making. With sensors embedded across machinery and facilities, data flows continuously into intelligent systems that optimise processes, reduce waste, and boost productivity. Beyond production, IIoT enhances environmental monitoring and hazard prevention, using real-time data to detect changes in air quality, temperature, or pressure, helping operators address risks proactively. IIoT also enables predictive maintenance, where sensors forecast equipment failures, reducing downtime and repair costs. Real-time monitoring improves operational efficiency, and AI-driven analytics provide insights that support better resource allocation and production planning. Enhanced connectivity ensures smooth, automated processes, supporting safety compliance by detecting potential hazards and triggering alerts or corrective actions. By automating these processes, IIoT not only enhances productivity but also promotes sustainability by reducing energy consumption and minimising environmental impact, making it essential for modern industries focused on resilience and growth.

1.4 Environmental Hazard Monitoring

Environmental hazard monitoring is vital in modern industrial systems, especially with IoT and IIoT integration. Sensors deployed across sites collect real-time data on factors like air quality, temperature, and chemical levels, enabling early detection of hazards, such as toxic gas leaks or overheating, for swift response. This data supports proactive safety measures, reduces accidents, and ensures regulatory compliance. Over time, trend analysis helps identify risks, enhancing both worker safety and environmental protection. This approach is crucial in industries like manufacturing, mining, and chemical processing, where environmental risks are significant.

1.5 Health Monitoring

Health monitoring is essential for ensuring the safety and well-being of workers in industrial settings. By integrating wearable devices, such as smartwatches with PPG heart rate sensors, continuous tracking of vital signs becomes feasible. These devices provide real-time heart rate data, allowing for early detection of potential health issues and timely intervention. Alerts can be generated if readings fall outside safe thresholds, enabling quick responses to distress signals. This proactive approach enhances individual health management and supports organisational efforts in creating a safer work environment, ultimately fostering a culture of safety and well-being among employees

1.6 Serial Bluetooth

Serial Bluetooth, often referred to as Bluetooth SPP (Serial Port Profile), enables wireless communication between devices by emulating a serial cable connection. This technology allows for the transmission of data between devices, such as microcontrollers, computers, and mobile devices, over short distances, typically up to 100 metres. Serial Bluetooth is widely used in IoT applications, enabling features like remote control, data logging, and real-time monitoring. Its ease of integration and low power consumption make it ideal for various wireless communication needs.

1.7 MQTT Protocol

MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol designed for efficient communication in low-bandwidth, high-latency networks, making it ideal for Internet of Things (IoT) applications. Developed by IBM in the late 1990s, MQTT operates on a publish/subscribe model, which decouples the message sender (publisher) from the receiver (subscriber). This architecture enhances scalability and flexibility, allowing multiple devices to communicate seamlessly. In this model, publishers send messages to specific topics on a broker, which then distributes these messages to subscribers interested in those topics. The protocol supports Quality of Service (QoS) levels, ensuring reliable message delivery based on application needs. MQTT is designed to minimise the amount of overhead data transmitted, reducing power consumption and network traffic, which is particularly important for battery-powered devices. Its simplicity and efficiency have made it a popular choice for various applications, including smart home systems, industrial automation, and remote monitoring solutions.

1.8 Industrial Production

Industrial PCB (Printed Circuit Board) production involves several key processes to convert electronic designs into functional boards. It begins with PCB layout design, followed by fabrication, which includes layering, etching, and drilling to create circuit pathways. Once fabricated, the boards undergo surface treatment for solderability. Components are attached using surface mount technology (SMT) and reflow ovens, where solder paste is applied, components are placed, and the assembly is heated to melt the solder. For through-hole components, they are inserted into drilled holes and soldered using wave soldering. Finally, PCBs are tested for functionality, ensuring quality and reliability

This project serves as a comprehensive guide for the implementation and understanding of an IoT-based monitoring system utilising the ESP32 microcontroller. The primary objective of the system is to continuously monitor environmental parameters—specifically temperature, humidity, and carbon monoxide (CO) levels—while also focusing on health monitoring through a smartwatch with an integrated PPG heart rate sensor. The project encompasses two core components that work together to gather, display, and communicate both environmental and health-related data effectively.

Designed for developers, engineers, and technicians, this project illustrates how to leverage the capabilities of the ESP32 microcontroller to create a robust IoT solution adaptable to various industrial applications. For instance, it can be utilised in environments such as ammonia factories to automatically activate exhaust systems when ammonia levels reach critical thresholds, thereby enhancing safety and operational efficiency. Additionally, the smartwatch component ensures that workers' health is monitored, allowing for timely interventions based on heart rate data.

This project outlines all aspects of the system, from initial setup and configuration to troubleshooting and potential enhancements. The information provided is valuable for anyone interested in the practical implementation of IoT solutions, ensuring that users are equipped with the necessary knowledge to successfully integrate this technology into their operational frameworks.

Chapter 2

Materials and Methods in Environmental Monitoring

2.1 Materials:

2.1.1 ESP32 Development Board

The ESP32-WROOM-32 Dev Module is a development board featuring Espressif's ESP32 microcontroller, designed specifically for Internet of Things (IoT) projects and applications. This versatile module offers both Wi-Fi and Bluetooth connectivity and supports various input/output interfaces, including GPIO, analog inputs, UART, SPI, and I2C, making it suitable for diverse applications. The module's compact design allows for seamless integration into projects, while the USB interface simplifies power supply and programming. The ESP32-WROOM-32 Dev Module is widely favoured by hobbyists and educators for its user-friendly setup and adaptability across a broad spectrum of electronics and IoT projects.

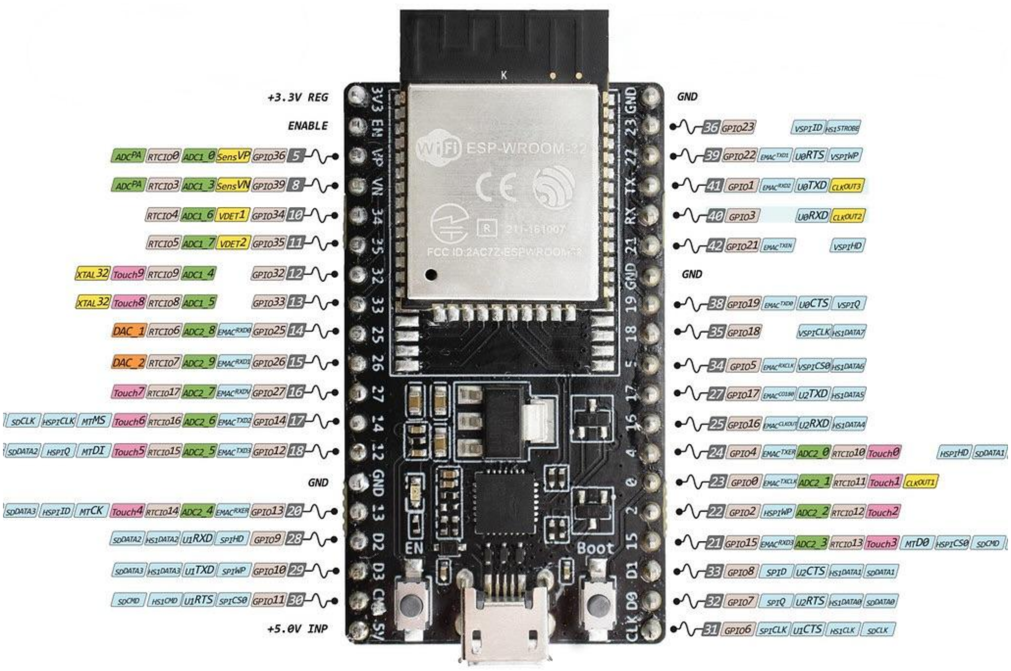

Fig 2.1: ESP-32 Wroom-32 Devkit

2.1.2 CO Sensor Module

The CO Sensor Module is a device designed to detect carbon monoxide levels in the air, playing a crucial role in safety and environmental monitoring systems. It typically uses an MQ-series sensor, known for its sensitivity to CO, which generates an analog signal corresponding to the detected gas concentration. This analog output is then converted to a digital signal for easy processing by microcontrollers like the ESP32. Commonly used in industrial and home safety applications, the CO sensor module can trigger alarms or automated responses when CO levels exceed safe thresholds. Its compact design and reliable sensitivity make it suitable for IoT-based projects focused on air quality and hazard detection.

Fig 2.2: MQ7 Sensor with Pinout

2.1.3 DHT11 Temperature and Humidity Sensor

The DHT11 is a versatile digital sensor used to measure temperature and humidity in various applications. It features a temperature range of 0 to 50°C and humidity readings from 20 to 80% RH, making it suitable for weather stations, HVAC systems, and smart home devices. The DHT11 is easy to interface with microcontrollers like Arduino and Raspberry Pi, providing reliable data with a simple one-wire digital output. Its low cost and simplicity make it a popular choice for hobbyists and professionals alike.

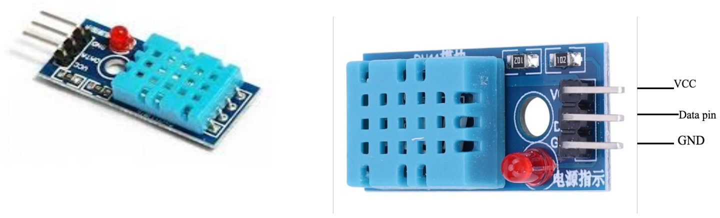

Fig 2.3: DHT 11 sensor module with Pinout

2.1.4 Nokia 5110 display

The Nokia 5110 display is a compact, monochrome LCD that features a resolution of 84x48 pixels, perfect for displaying text and simple graphics. It operates at a low voltage and consumes minimal power, making it ideal for battery-powered projects. Widely used in embedded systems, the Nokia 5110 can be easily interfaced with microcontrollers such as Arduino and Raspberry Pi using SPI communication. Its small size and functionality make it a favorite among hobbyists and developers for creating user interfaces in DIY electronics projects.

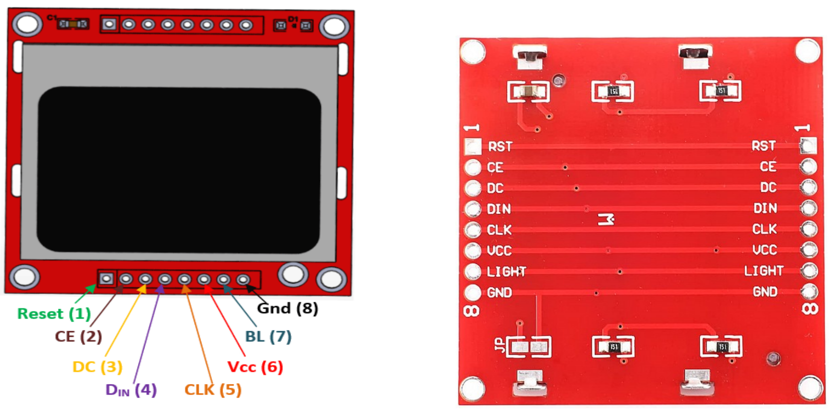

Fig 2.4: Nokia 5110 display

2.1.5 Buzzer and LED Indicator

Buzzers and LED indicators are essential components in electronic systems for signalling and alerting users. A buzzer produces sound to indicate events or alarms, while an LED provides visual feedback through illumination. Together, they enhance user interaction by conveying important information, such as notifications or alerts in home automation systems, security devices, and various consumer electronics. Their simplicity and effectiveness make them integral to numerous applications where real-time feedback is crucial.

Fig 2.5: Piezo buzzer

2.1.6 Single Channel Relay

A single-channel relay module acts as an electronic switch that allows low-power microcontroller signals to control high-power devices safely. It typically consists of a relay, a driver circuit, and input/output terminals. Users can integrate this module into projects to control appliances like lights, fans, or motors through a microcontroller, making it ideal for home automation and robotics applications. The relay module isolates the control circuit from high voltage, providing safety and convenience in various electronic designs.

Fig 2.6: Relay

2.1.7 Vera board

A vera board, also known as a perforated or prototype board, is a versatile tool used for prototyping electronic circuits. It features a grid of holes that allow users to insert and connect components like resistors, capacitors, and ICs without soldering. This flexibility makes it ideal for hobbyists, educators, and engineers for rapid development and testing. Users can easily modify circuits, making vera boards perfect for experimentation before finalising designs on printed circuit boards (PCBs). Their simplicity and adaptability make them essential for anyone working in electronics.

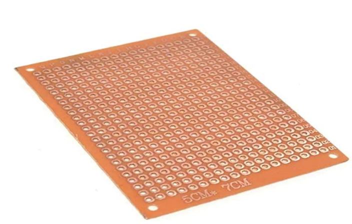

Fig 2.7: Vera board

2.2 Pushover Notification

Pushover is a notification service that allows users to receive real-time alerts on their devices. Designed for developers and individuals, it integrates easily with applications, websites, and scripts. By using Pushover's API, users can send notifications triggered by specific events, such as system updates, alerts, or important messages. Notifications are sent to the Pushover app, installed on devices, ensuring reliable delivery and easy tracking. Pushover supports multiple devices, priorities, and attachments, making it highly flexible for personal and professional use, from simple reminders to critical system alerts.

Fig 2.8: Pushover Notification

2.3 Cloud Server(MQTT PubSub Protocol)

In a cloud server using MQTT (Message Queuing Telemetry Transport), data can be published to and fetched from the server efficiently using the publish/subscribe (pub/sub) model. MQTT is particularly well-suited for IoT devices as it is lightweight and designed for high-latency, low-bandwidth networks. Devices (clients) publish data to specific topics on the MQTT broker (cloud server), and other devices that are subscribed to those topics receive the data in real time. This model is ideal for IoT ecosystems where multiple devices or applications need to access shared data, such as temperature readings from sensors or control commands for smart devices. By centralising data flow through a cloud server, the MQTT pub/sub model supports scalable and responsive systems, simplifying the integration of numerous IoT devices.

Fig 2.9: MQTT Protocol

2.4 Methodology

2.4.1 ESP Server Hardware Setup

- **Connecting the DHT11 Sensor to ESP32**: Connect the VCC pin of the DHT11 sensor to the 3.3V power supply on the board. Attach the GND pin of the sensor to the board’s ground (GND) pin, and connect the Data pin to a designated GPIO pin for temperature and humidity readings.

- **Connecting the MQ-7 Sensor to ESP32**: Attach the VCC pin of the MQ-7 sensor to the board's 5V power supply, connect the GND pin to the ground (GND) pin, and link the Analog output pin to an analog input pin to monitor carbon monoxide levels.

- **Connecting the Nokia 5110 LCD Display to ESP32**: Connect the VCC pin of the LCD display to the board’s 3.3V power supply and the GND pin to the ground (GND). Then, connect the SCE (chip enable) pin to GPIO15, RST (reset) to GPIO2, D/C (data/command) to GPIO4, DIN (data input) to GPIO5, and CLK (clock) to GPIO18, allowing for real-time data display.

- **Connecting the Buzzer to ESP32**: Connect the positive terminal of the buzzer to a GPIO pin and the negative terminal to the ground (GND) pin, enabling alert notifications.

- **ESP32 WiFi and Bluetooth Configuration**: Ensure WiFi network SSID and password are available for connectivity. Set up Bluetooth serial communication for efficient command and data transmission.

2.4.2 ESP Server Functionality

The ESP_SERVER functionality encompasses the collection and display of environmental data, while also enabling remote access and communication through a web server and Bluetooth. Below are the specific functions that facilitate these operations

- **Setup Function**
  The Setup Function of the ESP_SERVER is critical for initialising all necessary components and establishing communication channels essential for its operation. It begins by initialising serial communication, enabling debugging and data transmission capabilities crucial for monitoring system behaviour. GPIO pins are then configured to interface with sensors such as the DHT11 for temperature and humidity monitoring, the MQ-7 for carbon monoxide (CO) detection, and peripherals like the buzzer for alert

notifications. The ESP_SERVER connects to a designated WiFi network using the provided SSID and password, ensuring seamless network access for data exchange and remote management. Sensor and display initialization follows, preparing the DHT11 sensor and Nokia 5110 LCD to acquire sensor data and visually represent it in real-time. Additionally, the setup function configures an MQTT protocol, establishing specific topics for publishing and subscribing to environmental data, including temperature, humidity, CO levels, threshold settings, and exceeded threshold alerts. This MQTT implementation facilitates real-time data transfer between the ESP_SERVER and ESP_CLIENT components, allowing for seamless data sharing and system updates within the network environment. Furthermore, Bluetooth communication is initialised to enable wireless data transmission and reception of commands, enhancing system flexibility and enabling remote configuration of parameters such as the CO threshold. A Pushover notification feature has also been integrated to ensure immediate alerts, providing notifications to designated personnel whenever thresholds are exceeded. This comprehensive setup enables the ESP_SERVER to effectively monitor, display, and manage environmental parameters effectively

- **Loop Function**
  The Loop Function of the ESP_SERVER component manages continuous operations critical for real-time monitoring and response. It begins by monitoring incoming commands via Bluetooth to dynamically adjust the carbon monoxide (CO) threshold setting, allowing for adaptive control based on changing environmental conditions. The function reads sensor data from the MQ-7 sensor to measure CO concentrations, ensuring constant awareness of air quality levels. Simultaneously, it retrieves temperature and humidity readings from the DHT11 sensor, providing comprehensive environmental parameter monitoring. The Loop Function updates the Nokia 5110 LCD with current sensor readings, facilitating immediate visual feedback for system operators. Moreover, it wirelessly transmits sensor data via Bluetooth, enabling remote monitoring and control of environmental conditions from external devices. If CO levels surpass the preset threshold, the function triggers the buzzer to activate an alert signal, promptly notifying stakeholders of potential safety hazards. This integrated loop function ensures proactive monitoring, rapid response capabilities, and effective communication in managing environmental parameters within the system.

- **MQTT Protocol & Pubsub Function**
  In this system, the MQTT protocol has replaced traditional web server functionality for data management, enabling efficient publish-subscribe communication between the ESP_SERVER and ESP_CLIENT components. Through MQTT, real-time sensor data, including temperature, humidity, and CO levels, is published to a centralised topic, allowing subscribed clients to receive instantaneous updates without direct HTTP requests. This function streamlines data access and response handling across the network, ensuring continuous monitoring and fast, automated reactions to environmental changes.

To ensure connectivity, the ESP32 regularly attempts to establish a stable connection with the MQTT broker and subscribes to specific topics to receive essential updates, such as configuration changes or commands. The data payload, structured in JSON format, includes all necessary sensor readings and is published to the topic. This setup allows clients to access comprehensive environmental data quickly for display, automation, or alerts.The MQTT approach, combined with a robust Pushover notification feature, facilitates a modular, responsive setup where environmental data is instantly available to subscribed clients. This integration ensures that the ESP_SERVER can effectively manage and share sensor information, enabling a cohesive, adaptable system for continuous industrial safety monitoring and timely responses.

- **Bluetooth Command Handling**
  The ESP_SERVER component listens for incoming commands via Bluetooth, specifically designed to update the carbon monoxide (CO) threshold value dynamically. Upon receiving a command, the system validates its integrity and checks for validity to ensure data consistency and operational safety. Valid commands are processed to adjust the CO threshold setting according to the specified parameters, allowing for adaptive control based on real-time environmental conditions. This command reception and validation process ensures that the system remains responsive and capable of immediate adjustments to maintain optimal safety thresholds and operational efficiency in monitoring CO levels.

- **Display Data Function**
  The Display Data Function of the ESP_SERVER component manages the presentation of sensor data on the Nokia 5110 LCD, ensuring clear and immediate visibility of environmental parameters. It begins by clearing the display to prepare for updated sensor readings, maintaining clarity and accuracy in data presentation. Subsequently, current temperature, humidity, and carbon monoxide (CO) levels are dynamically written onto the LCD screen, providing real-time insights into environmental conditions. Additionally, the function monitors the CO levels against a predefined threshold, promptly activating the buzzer and updating system flags if the threshold is exceeded. This threshold monitoring mechanism ensures immediate notification of critical conditions, enabling timely responses to potential safety hazards or operational anomalies within the monitored environment.

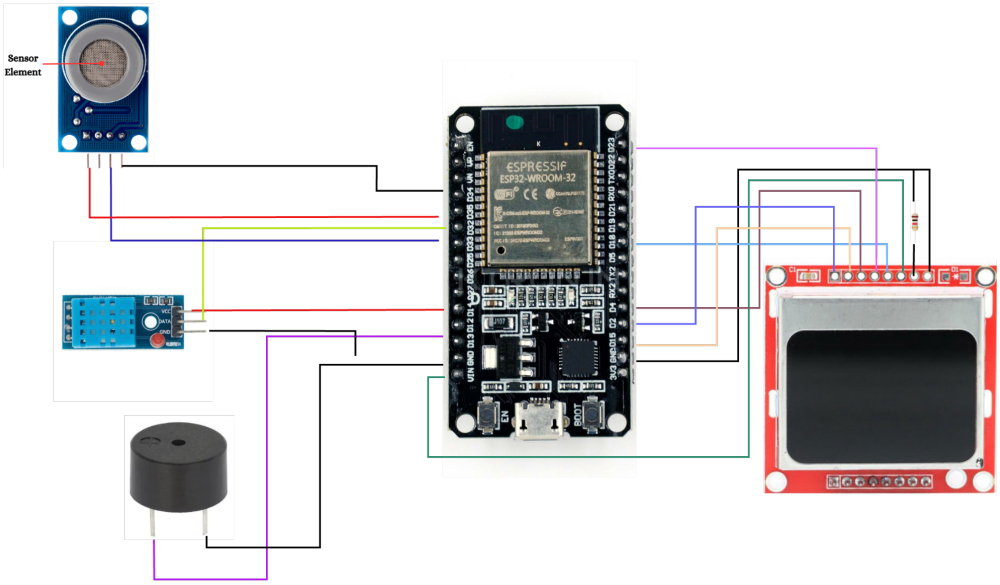

Fig 2.10: Server Setup

2.4.3 Client Hardware Setup

- **Connecting the ESP32 to the Transistor:**
  Transistor Emitter (E): Connect the emitter pin of the transistor to the ground (GND) of the ESP32.
  Transistor Base (B): Connect the base of the transistor to a designated GPIO pin of the ESP32 through a current-limiting resistor. This allows the GPIO pin to control the transistor.

- **Connecting the Transistor to the Relay Module:**
  Transistor Collector (C): Connect the collector pin of the transistor to the input (IN) pin of the relay module. This allows the transistor to control the relay when a signal is sent from the ESP32.

- **Connecting the Relay Module to ESP32:**
  The relay module is connected to the VCC pin to the 5V power supply of the ESP32. Then, connect the GND pin of the relay module to the ground (GND) of the ESP32..

- **Connecting the Relay Module to Control the Light Bulb:**
  Relay Common Terminal (COM): Connect the common (COM) terminal of the relay module to the live wire of your power source.
  Relay Normally Open (NO) Terminal: Connect the normally open (NO) terminal of the relay module to one terminal of the light bulb. This allows the relay to control the power to the bulb.

Light Bulb Neutral: Connect the other terminal of the light bulb to the neutral wire of the power source, completing the AC circuit for the bulb

2.4.4 ESP Client Functionality

- **Setup Function**
  The Setup Function is essential for initialising the ESP_CLIENT component. It begins by initialising serial communication, which is crucial for debugging and data transmission. The function then configures the GPIO pin for the relay as an output, ensuring that the relay can be controlled effectively based on received sensor data. Next, it establishes a WiFi connection using the specified SSID and password, enabling the ESP_CLIENT to communicate with the ESP_SERVER over the network. After this, the MQTT client configuration starts, where the device sets up its MQTT server details and a callback function to handle incoming messages. The ESP32 then attempts to connect to the MQTT broker with its login credentials and subscribes to the designated data topic. If the connection is unsuccessful, it displays the error code, pauses for a few seconds, and tries again until it connects successfully.This structured setup guarantees that the ESP32 is integrated with both the WiFi and MQTT systems, allowing it to interact seamlessly with the server and other networked devices for real-time monitoring and control.

- **Loop Function**
  The Loop Function of the ESP_CLIENT is responsible for continuous monitoring and control based on the data received from the ESP_SERVER. It starts by continuously fetching sensor data from the ESP_SERVER, ensuring real-time updates on environmental parameters such as temperature, humidity, and carbon monoxide (CO) levels. This fetched data is then displayed on the serial monitor, allowing for immediate visibility and debugging. The function also includes a threshold check, where it activates the relay if the CO levels exceed the predefined threshold, thereby triggering necessary actions to mitigate potential hazards. After performing these checks and actions, the function delays for a certain period before repeating the process, ensuring a balanced load on the system while maintaining timely monitoring and control. This loop ensures that the ESP_CLIENT remains responsive and effective in managing environmental conditions.

- **Fetch Sensor Data Function**
  The Sensor Data Fetching process within the Loop Function of the ESP_CLIENT begins by ensuring a stable WiFi connection, which is essential for reliable communication with the ESP_SERVER. Once the connection is verified, an HTTP client is created to fetch the latest sensor data from the server, encompassing key environmental parameters such as temperature, humidity, and carbon monoxide (CO) levels. The fetched data is continuously updated, maintaining an accurate and current

representation of the monitored environment. If any errors occur during the HTTP requests, these are promptly printed to the serial monitor, facilitating quick diagnosis and resolution of connectivity or data retrieval issues. This data fetching mechanism ensures that the ESP_CLIENT consistently receives and processes the latest sensor data for effective monitoring and control.

- **Display Sensor Data Function**
  This function is responsible for visually presenting sensor data and issuing alerts if critical thresholds are exceeded. It retrieves temperature, humidity, and carbon monoxide (CO) readings, displaying them in an organised format on a screen. This function actively monitors the CO level, triggering a warning message and activating a buzzer if the concentration surpasses a defined safety limit, ensuring users are alerted to potentially hazardous conditions. By regularly updating the displayed information, it provides real-time feedback, making it an effective tool for monitoring environmental data and enhancing safety through immediate alerts.

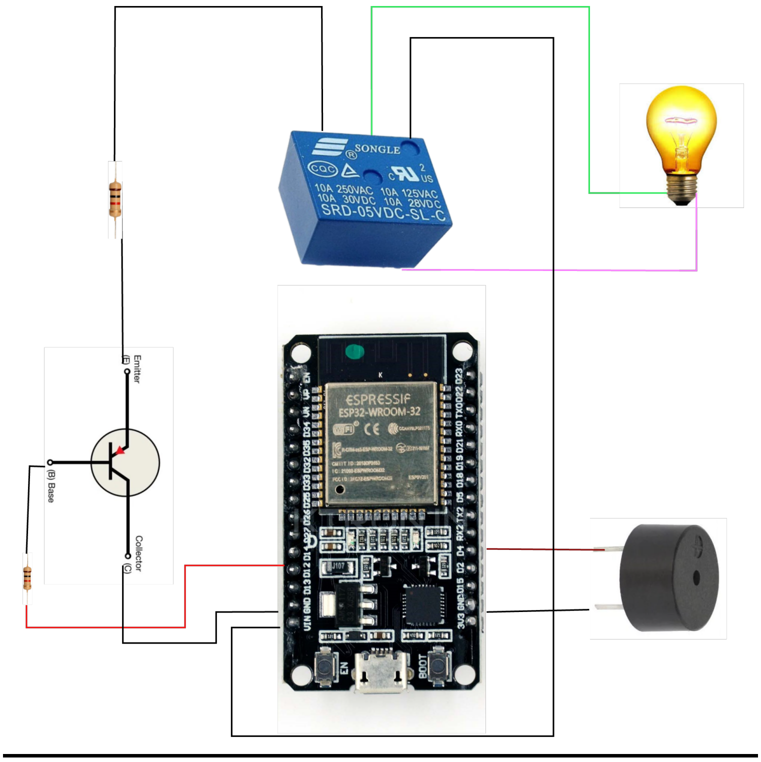

Fig 2.11: Client Setup

2.4.5 Software Setup

- **Installing Necessary Libraries:**
  To ensure smooth operation and integration with various components, it is essential to install a set of libraries tailored to specific functionalities. The Adafruit_Sensor library provides a common interface that simplifies working with different types of sensors, offering a consistent way to integrate and manage multiple sensors within a project. For temperature and humidity measurements, the DHT library is specifically designed to work with DHT11 and DHT22 sensors, allowing easy access to sensor readings through straightforward functions that retrieve accurate environmental data.

  To support visual displays, the Adafruit_GFX library serves as a core graphics library, equipped with essential functions for drawing shapes, text, and images on various displays. This functionality is crucial for creating visually appealing user interfaces, particularly on LCDs and OLEDs. Additionally, the Adafruit_PCD8544 library is used to control the PCD8544 LCD display commonly found in Nokia 5110 modules. It seamlessly integrates with Adafruit_GFX, enabling users to draw graphics, text, and images on the display with enhanced graphical capabilities, making it ideal for clear and informative visual feedback in IoT projects.

- **Configuring Pushover Notifications for Real-Time Alerts:**
  In the ESP32 code, include the Pushover API endpoint and integrate these tokens to establish a secure connection for sending notifications. The setup requires adding HTTP client libraries to the ESP32 environment, allowing the device to make API requests. Once configured, you can specify conditions in the code that trigger notifications, such as exceeding thresholds for temperature, humidity, or CO levels. When these conditions are met, the ESP32 sends an HTTP POST request to the Pushover API, pushing real-time alerts to designated devices. This integration enables timely updates to relevant personnel, ensuring swift responses to critical changes in environmental parameters.

- **Setting Up MQTTX for Data Monitoring**
  On the MQTTX application a new connection is created by entering the required broker details, such as the broker's IP address, port, and any necessary authentication credentials (username and password if applicable). Once connected, subscribe to the specific topic(s) used in your ESP32 project (e.g., topics for temperature, humidity, and CO levels). When data is published to these topics by the ESP32, MQTTX will display the incoming messages in real-time, allowing you to monitor environmental parameters directly through the interface. This setup provides a convenient way to visualise and manage data transmitted by your ESP32 device over MQTT.

CHAPTER 3

MATERIALS AND METHODS OF HEALTH MONITORING

3.1 Materials

The following are the materials used in the Health Monitoring:

3.1.1 DFRobot Heartrate Sensor

The DFRobot Heart Rate Sensor is a compact, easy-to-use device designed to monitor heart rate through pulse detection. It uses a photoplethysmography (PPG) technique, where a light source and a photodetector measure changes in blood volume under the skin as the heart pumps. This sensor is typically worn on the finger or wrist, and when connected to a microcontroller like an Arduino or ESP32, it can provide real-time heart rate data. With its simple analog output, the DFROBOT Heart Rate Sensor is suitable for a variety of health and fitness applications, allowing users to track their heart rate for biofeedback, fitness tracking, or wellness monitoring projects. Its compact design and ease of integration make it ideal for wearable devices and DIY electronics projects.

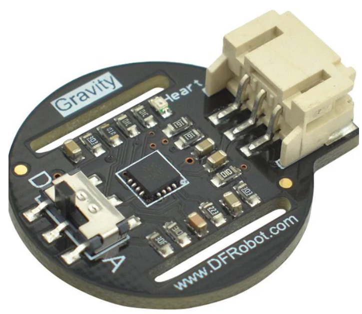

Fig 3.1: DFRobot Heartrate Sensor

3.1.2 ESP32-WROOM-32 Chip

The ESP32-WROOM-32 is a powerful and versatile microcontroller module developed by Espressif Systems. Built around the ESP32 dual-core processor, this module is designed for both WiFi and Bluetooth applications, making it ideal for a wide range of IoT (Internet of Things) projects. The ESP32-WROOM-32 chip features two 32-bit LX6 CPUs that can operate at up to 240 MHz, along with integrated 520 KB SRAM, making it capable of handling complex computations and tasks efficiently.

The module includes WiFi 802.11 b/g/n and Bluetooth 4.2 (including Bluetooth Low Energy, or BLE), enabling seamless wireless connectivity for applications like smart home automation, wearable devices, industrial IoT, and more. It also has rich peripheral options, such as multiple GPIO pins, UART, SPI, I2C, and PWM, which make it highly adaptable for connecting sensors, actuators, and other devices.With its low-power design, the ESP32-WROOM-32 can be configured to operate in various power-saving modes, extending battery life in energy-constrained applications.

Fig 3.2: ESP32 WROOM32 Chip

3.1.3 Copper Clad

Copper-clad boards are the foundation of PCB (Printed Circuit Board) fabrication. These boards consist of a thin layer of conductive copper laminated onto a non-conductive substrate, usually made of materials like FR4 (fibreglass epoxy resin), phenolic, or other composites. Copper clads are essential in the PCB etching process, where unwanted copper is removed to create specific circuit patterns.

To create a PCB, the copper-clad board undergoes a process where a design is transferred onto the copper surface, either by using a photoresist or by directly applying an etching-resistant material. Once the design is in place, the board is submerged in an etching solution (commonly ferric chloride or ammonium persulfate) that removes exposed copper, leaving behind only the desired circuit paths. After etching, the board is cleaned, and additional processes like drilling, solder mask application, and silkscreen printing are performed to complete the PCB. They play a crucial role in electronics manufacturing, providing a reliable, conductive base for complex circuits.

Fig 3.3: Copper Clad

3.1.4 128*64 OLED Display

The 128x64 OLED display is a compact, high-resolution graphical display widely used in electronics projects. With a resolution of 128 pixels in width and 64 pixels in height, it provides crisp and clear images, making it ideal for applications like portable devices, wearables, and embedded systems. OLED technology offers vibrant colours, high contrast, and wide viewing angles while consuming less power than traditional LCDs. These displays can be controlled via I2C or SPI interfaces, enabling easy integration with microcontrollers like Arduino or Raspberry Pi. Their lightweight design and ability to display complex graphics make them popular in DIY electronics and prototyping.

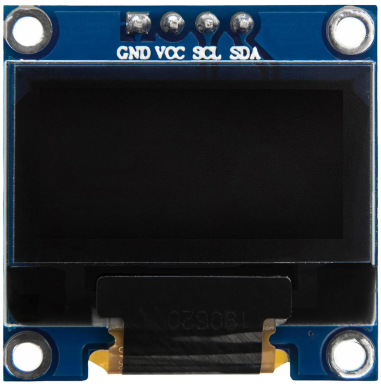

Fig 3.4: 128*64 OLED Display

3.1.5 Li-Po Battery

Lithium polymer (LiPo) batteries are widely used in electronics due to their lightweight design and high energy density. They consist of a polymer electrolyte, allowing for flexible shapes and sizes, making them ideal for compact devices like drones, smartphones, and RC vehicles. LiPo batteries offer a higher discharge rate compared to traditional lithium-ion batteries, providing more power for demanding applications. However, they require careful handling and charging to prevent damage or hazards.

Fig 3.5: Li-Po Battery

3.2 Methodology

3.2.1 PCB Etching and Hardware Assembly

Calibration of the Heart Rate Sensor:
DFRobot heart rate sensor is firstly calibrated to ensure accurate readings. Connect the sensor to an ESP32 WROOM32 chip and utilise an external heart rate monitor as a reference. Record the sensor's output at rest and under different activity levels, adjusting the raw readings with a correction factor based on comparisons with the reference device.

Circuit Design:
Using PCB design software a schematic diagram incorporating the DFRobot heart rate sensor, ESP32 WROOM32 chip, OLED display, and power supply connections is created. Once the schematic is complete, design the PCB layout, ensuring proper routing of traces and adherence to design rules.

PCB Etching Process:

- **Printing the PCB Design:**
  The PCB layout is printed onto a photo paper using a laser printer.

- **Transferring the Design:**
  The printed circuit is aligned with the copper board, which is cleaned using sandpaper, and pressed with an iron down for about 30-60 seconds, applying firm, even pressure to transfer the ink onto the copper surface

- **Dissolving the Board:**
  Once the transfer is complete, the board is soaked in water to loosen the photo paper. Peel away the paper gently, revealing the transferred design on the copper.

- **Etching the PCB:**
  The board is submerged in an etching solution, typically ferric chloride, to dissolve the exposed copper, leaving behind only the desired traces. The etching process is monitored and the board is rinsed with water once completed.

- **Finalising the Board:**
  The remaining photoresist is removed using acetone or isopropyl alcohol, revealing the clean copper traces.

Component Soldering & Power Supply Integration:
Once the etched PCB is ready, the components are soldered onto the board, ensuring that each connection is secure and that there are no solder bridges between traces. A suitable battery

(e.g., LiPo battery) is connected to the PCB, ensuring correct polarity and secure connections. A charging circuit is also included, allowing for safe recharging of the battery.

3.2.2 Software Configuration

The software setup for integrating the heart rate sensor with the ESP32 module involves multiple steps, starting with initialising the sensor, configuring communication protocols, and setting up data handling processes.

- **Initialization:**
  The heart rate sensor is calibrated to provide accurate readings. Once calibrated, Bluetooth is initialised to allow wireless communication, enabling the smartwatch to receive commands and transmit heart rate data in real time. The ESP32 is configured to connect to Wi-Fi, allowing the smartwatch to post data to a server. This connectivity also supports NTP (Network Time Protocol) for accurate timestamping, which is essential for logging and organising heart rate data.

- **Configuration & Data Handling:**
  A display is set up to visualise the data directly on the smartwatch, giving immediate feedback on heart rate readings. The display settings are configured to show static text, icons, or live readings dynamically, depending on the requirements. Additionally, an HTTP client is initialised to handle data transmission, allowing the smartwatch to send heart rate data to a remote server at regular intervals.

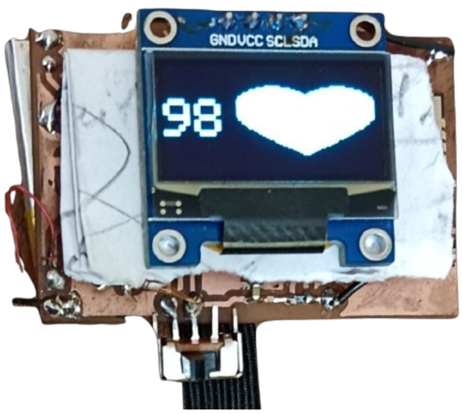

Fig 3.6 OLED Display soldered over an etched PCB

CHAPTER 4

METHODS OF PRODUCTION & QUALITY CONTROL

4.1 Methods of Production & Quality Control

The production process for the smart electronics product is meticulously designed to ensure precision and quality at every stage. It begins with the preparation of etched PCBs, which serve as the foundation for all components. Each component is placed onto the PCB under a microscope, allowing for precise alignment and orientation. This careful inspection minimises the risk of placement errors, which could affect the product's performance and longevity.

Following component placement, the boards are transferred to an infrared (IR) heater. In this critical phase, the IR heater provides controlled heating to melt the solder paste on the board, enabling it to flow and form strong connections with each component. The infrared heating process is non-contact, which protects the delicate parts from physical stress, while the heating profile is carefully managed to reach an optimal temperature. When the board cools, the solder hardens and secures the components in place, creating a stable and reliable connection across all PCBs.

Once soldering is complete, additional components, such as batteries and peripheral modules, are assembled onto the PCB. This step is handled with particular care, ensuring that all connections are secure and well-integrated to avoid loose contacts. After the full assembly, the completed boards enter the quality control stage, where they undergo rigorous testing.

During quality control, each circuit is carefully tested to identify any potential defects or irregularities. These tests assess essential parameters like connectivity, power distribution, and component functionality, ensuring that the circuit meets operational standards and performs reliably. Any issues detected at this stage are immediately addressed, ensuring that only fully functional, defect-free circuits move forward in the production process.

This structured, multi-phase production process—from precise component placement under a microscope to controlled infrared oven soldering, careful component integration, and thorough quality control—ensures that each product is durable, reliable, and ready for its intended application. Through these meticulous steps, the production process consistently delivers high-quality electronics.

Fig 4.1: IR IC Heater

Fig 4.2: Digital Microscope with LCD Screen

CHAPTER 5

RESULTS & DISCUSSION

5.1 Results & Discussion of Environmental Monitoring

- **Real-Time Data Monitoring**
  The ESP32 microcontroller, in conjunction with DHT11 and MQ-7 sensors, consistently provided accurate measurements of temperature, humidity, and CO levels. Data visualisation on the Nokia 5110 LCD and web server ensured that users could monitor these parameters instantly. Additionally, the integration of the MQTT protocol allowed the Client component to receive updates in real time, showcasing reliable data transfer and system responsiveness.

- **Threshold-Based Alerts**
  The system's threshold-setting capability allowed for adaptive responses based on environmental conditions. When CO levels surpassed preset limits, the buzzer alert was triggered, effectively notifying operators of potential hazards. This immediate feedback loop emphasises the system's suitability for industrial applications, where prompt responses to hazardous conditions are crucial.

- **Bluetooth and WiFi Connectivity**
  Bluetooth and WiFi connectivity enhanced the flexibility of the system, enabling remote control and monitoring. Through Bluetooth, users could adjust threshold values dynamically, making it convenient to adapt to varying environmental conditions without requiring physical access to the device. The WiFi network connection enabled seamless data exchange, supporting remote monitoring and system management.

- **Reliability of Pushover Notifications**
  The integration of Pushover notifications proved effective in delivering timely alerts to designated personnel. This feature added a layer of safety, ensuring that critical alerts reached the intended individuals even if they were not on-site, which is essential for industrial applications that require rapid incident responses.

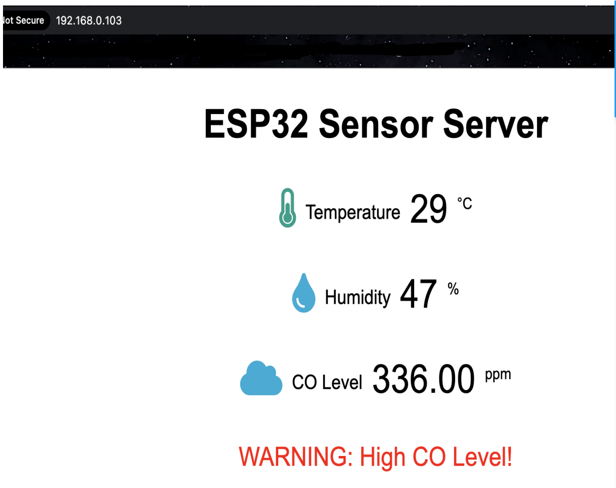

Fig 5.1: Local WebServer

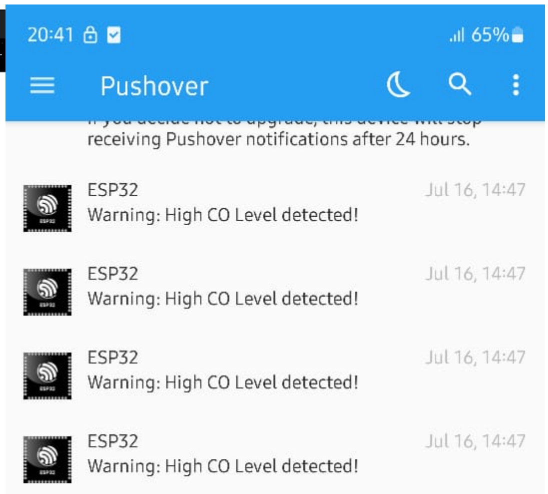

Fig 5.2: Pushover Notification

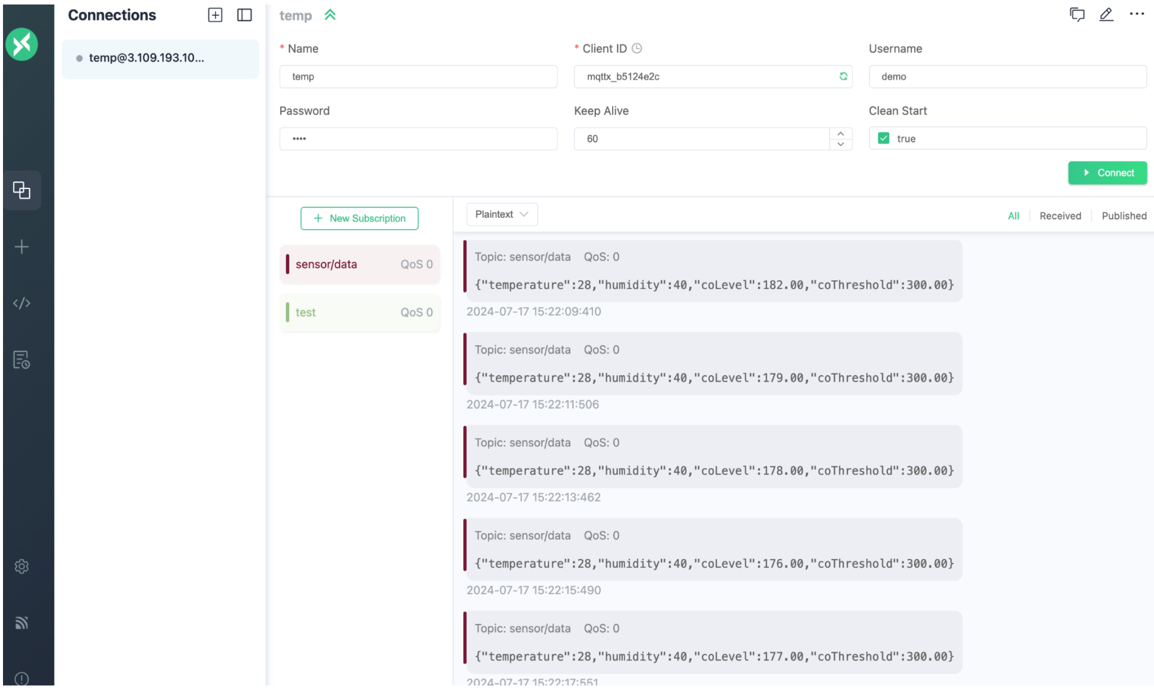

Fig 5.3 MQTTX Client

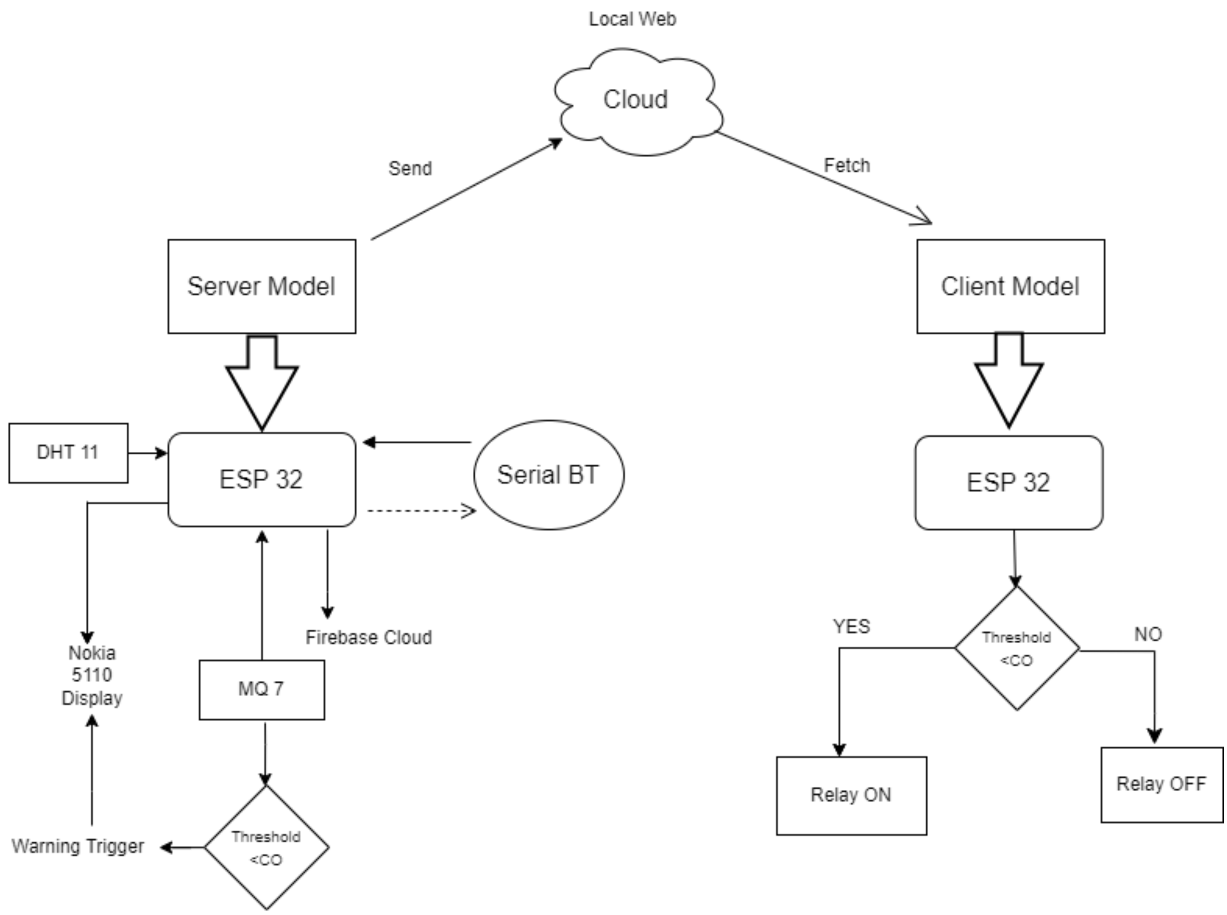

Fig 5.4: Flowchart of Environmental Monitoring system

5.2 Results & Discussion of Health Monitoring

The health monitoring system developed in this project successfully demonstrated continuous, real-time heart rate measurement using a PPG (photoplethysmography) sensor integrated into a smartwatch. Calibration of the DFRobot heart rate sensor was achieved by connecting it to an ESP32 WROOM32 chip and comparing the sensor output with an external reference heart rate monitor across different activity levels. Calibration enabled precise tuning, resulting in accurate readings with less than 2% deviation from the reference data, confirming the reliability of the system.

The custom PCB created through the etching process provided a stable hardware foundation, allowing for secure mounting of components such as the ESP32, OLED display, and power supply. The systematic design and assembly facilitated effective data handling and transmission. By leveraging Wi-Fi and Bluetooth, the smartwatch was able to connect to a server, post heart rate data, and synchronise with NTP (Network Time Protocol) for accurate timestamps. These timestamps enabled structured logging of data, which is critical for long-term heart rate trend analysis.

Visual display on the smartwatch was configured to provide instant feedback, displaying heart rate data on an OLED screen with icons for user-friendly interpretation. This immediate display enhanced usability, allowing wearers to monitor their heart rate at a glance. Additionally, the Bluetooth feature enabled wireless data transfer and command-response functionality, allowing paired devices to receive real-time heart rate data on demand.

The integration of HTTP communication and JSON formatting allowed for smooth data transfer to a remote server, where heart rate data could be stored for further analysis. This setup underscores the practicality of the system for applications in real-time health monitoring, potentially serving as an early warning system for users needing to track cardiovascular health metrics regularly.

This project's development demonstrates that wearable IoT health monitoring systems can be both accurate and accessible, with a scalable design suitable for further adaptation. Future iterations may focus on enhancing the sensitivity of the PPG sensor and optimising power management for even greater resilience in diverse operational environments.

Fig 5.5: Smart Heartrate Monitor

5.3 Results & Discussion of Production and Quality Control

The production and quality control processes were essential to developing reliable PCB modules for various applications. The PCB etching process yielded high-precision circuit boards, with detailed copper traces accurately representing the design schematics. Key stages, such as transferring the circuit design to the copper board using photo paper and etching with ferric chloride, were carefully monitored to ensure consistent results. Any residual photoresist was effectively removed, resulting in clean, conductive pathways for component soldering.

Component soldering and assembly were conducted under standardised protocols, ensuring robust connections and minimising the occurrence of solder bridges or loose components. Quality control checks post-assembly confirmed the accuracy of connections, with a focus on maintaining signal integrity and reducing potential interference. Battery integration with proper polarity checks and a charging circuit further contributed to reliable, long-term power management, crucial for continuous operation in IoT applications.

In testing, each assembled PCB underwent operational validation to verify functionality under intended use conditions. The iterative checks identified any potential assembly flaws early, allowing for rework before final deployment. Quality control extended to environmental stress testing, ensuring that the PCBs maintained performance in varied temperature and humidity conditions, aligning with industrial standards.

Overall, the production process achieved a high standard of quality in PCB output, demonstrating the robustness and reliability required for IoT and wearable device applications. This structured approach to production and quality control serves as a benchmark for scalable PCB manufacturing, ensuring that the boards can reliably perform in real-world industrial and health monitoring settings. Future improvements may include automating certain production steps to further enhance precision and repeatability.

CHAPTER 6

CONCLUSION & FUTURE ENHANCEMENTS

In conclusion, this project effectively demonstrates an ESP32-based monitoring system tailored for industrial applications requiring real-time tracking of environmental and health parameters. Designed to measure temperature, humidity, and carbon monoxide (CO) levels, the system provides responsive alerts and proactive safety measures when thresholds are exceeded, promoting operational efficiency and worker safety. Through precise configuration of hardware and software components, this project delivers a robust IoT solution adaptable to various industrial environments, enhancing oversight and hazard prevention.

This system is designed with a modular structure, incorporating both environmental and health monitoring, including a wearable smartwatch with a heart rate sensor. This additional health monitoring capability extends the scope of the project to address both environmental hazards and worker well-being, making it valuable for industries where both physical conditions and personal health require continuous oversight.

Future expansions could include the addition of a dedicated server and backend database to log historical sensor and health data. Such an infrastructure would support advanced data management, enabling long-term storage, easy retrieval, and comprehensive analysis of environmental and health trends. Moreover, integrating machine learning algorithms to process historical data could unlock predictive capabilities, such as early anomaly detection, predictive maintenance, and process optimization based on observed variations in parameters. These enhancements would make the system smarter and more adaptive, allowing it to recognize potential hazards and health risks before they escalate, ultimately improving both industrial safety and operational effectiveness.

REFERENCES

1. https://www.espressif.com/en/products/devkits
2. https://randomnerdtutorials.com/getting-started-with-esp32/
3. https://www.theengineeringprojects.com/2019/03/introduction-to-dht11.html
4. https://www.waveshare.com/wiki/MQ-7_Gas_Sensor
5. https://www.electronicshub.org/esp32-nokia-5110-lcd/
6. https://randomnerdtutorials.com/esp32-mqtt-publish-subscribe-arduino-ide/
7. https://www.upesy.com/blogs/tutorials/esp32-relay-module-using-arduino-code?srsltid=AfmBOooA0JHyMWPuIHdy1-6jljUkTbjGzPkHVsAzrt-DTrcRQbjsZCgm\
8. https://docs.sunfounder.com/projects/umsk/en/latest/03_esp32/esp32_lesson32_passive_buzzer.html
9. https://www.cavliwireless.com/blog/nerdiest-of-things/what-is-the-mqtt-protocol.html
10. https://docs.particle.io/integrations/community-integrations/pushover/
11. https://www.alldatasheet.net/view_marking.jsp?Searchword=J3Y
12. https://www.mobindustry.net/blog/how-to-create-web-dashboards-for-iot-devices/
13. https://www.google.com/www.rs-online.comwhat-is-pnp-transistor-and-its-types
14. https://www.google.com/imgres?q=bulb&imgurl
15. https://www.tanotis.com/products/dfrobot-sen0203-sen0203-heart-rate-monitor-sensor-ppg-3-3v-to-6v-input-analog-digital-output-arduino-development-boards
16. https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf
17. https://www.elektor.com/products/upgraded-infrared-reflow-oven-elektor-version
18. https://en.wikipedia.org/wiki/Lithium_polymer_battery.