Journal of Robotics and Control (JRC)
Volume 5, Issue 6, 2024
ISSN: 2715-5072, DOI: 10.18196/jrc.v5i6.21925

ESPNow Protocol-Based IIoT System for Remotely Monitoring and Controlling Industrial Systems

Maryam Abdulhakeem Hailan [1], Nouby M. Ghazaly [2], Baraa Munqith Albaker [3]
[1, 2] Department of Technical College, Imam Ja'afar Al-Sadiq University, Baghdad, Iraq
[2] Department of Mechanical Engineering, Faculty of Engineering, South Valley University, Egypt
[3] Department of Computer Engineering, Collage of Engineering, Al-Iraqia University, Baghdad, Iraq
Email: [1] maream_abdihakeem@ijsu.edu.iq, [2] noby_mehdi@ijsu.edu.iq, [3] baraamalbaker@ymail.com
*Corresponding Author

***Abstract*—The shift from conventional manufacturing facilities to intelligent manufacturing facilities is a topic of significant interest due to its profound and enduring implications for the evolution of manufacturing practices on a global scale. The advent of Industry 4.0 is geared toward advancing the manufacturing sector by facilitating the production of goods with brief product life spans and tailored to individual customer preferences in a financially efficient manner. This paper introduces an Industrial Internet of Things system that powers the ESP32 microcontroller, the Blynk platform, and the ESP-Now protocol for remote monitoring and control of industrial processes. The system aims to improve operational efficiency and data management in industrial settings by addressing challenges associated with communication protocols and user interfaces. The implementation of the system comprises configuring the ESP32 to collect data from several sensors dispersed across factory sites. Integration with the Blynk platform enables real-time data visualization and device management, while the ESP-Now protocol facilitates efficient communication among IoT devices for seamless monitoring and control functionalities. The developed system shows significant advancements in industrial monitoring and control by offering enhanced scalability, interoperability, and adaptability to diverse industrial environments. Monitoring capabilities include weather conditions, motion detection, gas levels, and water quality assessment, with control functionalities extending to regulating water pumps and lamps. Metrics for assessing GUI performance include response time, data visualization accuracy, and user interaction efficiency. Robust encryption protocols and authentication mechanisms are implemented to ensure data security and privacy, enhancing the system's reliability and trustworthiness in industrial applications. The integrated system provides a comprehensive solution for industrial monitoring and control, offering efficient communication, scalability, and data security measures to optimize operational efficiency in diverse industrial environments. The system's advanced features and capabilities position it as a valuable tool for enhancing industrial processes and ensuring seamless data management and control.**

Keywords*—Industrial Internet of Things; Blynk Platform; ESP-Now Protocol; Industrial Monitoring; Smart Factory.

I. INTRODUCTION

A traditional factory requires a human-in-the-loop approach for monitoring and control, which may result in time consumption, low process monitoring accuracy, low-quality control, and additional human-hour costs [1]. Transform into a smart factory enables factory managers to automatically collect and analyze data to make smarter decisions and optimize production. The Industrial Internet of Things integrates the physical world's sensors, devices, and machines into the internet and transforms enormous amounts of data into insightful new knowledge via deep analytics and software [2]. In addition, real-time monitoring and remote control of the factory were performed.

The integration of actuators and sensing devices in the IIoT offers a level of intelligence and connectivity to industrial systems that were previously unreachable. This technology allows for the making of global networks and enhanced processing abilities within industrial environments [3].

The main issue under consideration in the proposed system relates to the necessity for a comprehensive, organized Industrial Internet of Things (IIoT) solution capable of efficiently supervising and handling diverse aspects of industrial actions. By increasing the sensor accuracy, rapid alarm response, seamless system combination, and reliable communication protocol functionality, the system is designed to provide a flexible, user-centric framework for real-time monitoring, alerting, and decision assistance within industrial environments. By coordinating data produced by sensors, alarm machines, and user interfaces, the proposed system seeks to reconcile traditional manufacturing methodologies with the innovative potential offered by smart factory developments.

The IIoT is currently a crucial research area for resolving traditional factory problems.

J. Vijayalakshmi *et al.* [4] implemented a monitoring system to detect ammonia gas leakage using the concept of the Internet of Things. When the MQ135 Sensor's threshold for ammonia gas detection is reached, the buzzer in the ammonia gas sensor activates, notifying the authorities. The dependence only on the MQ135 may overlook the potential for more comprehensive and integrated monitoring and control capabilities. Using various sensors,

M. Sangeetha *et al*. [5] proposed a system for industrial control and parameter monitoring. In addition to this system to alert workers and people around, there is an audio unit, buzzer, and light signal that gives them sound and audio alerts that something is wrong in the industry. The Blynk platform was used to display the results. Exclusive reliance on the Blynk platform may limit the ability of the system to provide more comprehensive and sophisticated monitoring and control functions beyond basic alerting mechanisms.

S.Pasika and S.T.Gandla. [6] proposed an Internet of Things (IoT)-based system for monitoring water quality. The proposed system comprises several sensors for detecting a variety of parameters. Additionally, the PC is used to process data from the MCU that is plugged into these sensors. The data were collected from sensors to the cloud for display by a ThinkSpeak IoT-based application. To rely on an internet connection: The proposed system relies on an internet connection to transfer data from sensors to the cloud for display. If there are internet connection issues, data transmission may be delayed or interrupted, affecting the real-time monitoring capabilities of the system.

S.A.H. Almetwally *et al*. [7] introduced an efficient real-time water quality monitoring system to combine water quality sensors, water distribution sensors, and a microcontroller that captures and processes data to execute commands to other components of the system. The interface and mobile application in real-time monitoring and control tools that can be used for monitoring performance factors will set the stage for the water distribution system in the building to monitor the quality of water consumed. Implementing a real-time water quality monitoring system on a larger scale, such as in a city-wide water distribution network, may pose challenges in terms of data processing, communications, and system maintenance.

E. Raghuveera *et al*. [8] The proposed air quality monitoring systems include air quality measuring sensors, a ZigBee module, a suitable 16-bit microcontroller, a GSM module, and a GPS module and are used to efficiently monitor air quality by using LoRa LPWAN gateways. They use the Ubidots IoT Platform as the visible parameter in GUI and analytics. The range of the ZigBee module and LoRa LPWAN gateways may be limited, resulting in certain areas not being monitored for air quality. This limitation may lead to incomplete data and an inaccurate assessment of the overall air quality at a given location.

W. Rusydi *et al*. [9] proposed an IIoT monitoring system prototype for an electrical machine failure using a Heltec LoRa ESP32 to communicate with a Blynk IoT platform server. The monitored parameter values are the voltage, current, and motion sensor readings. To further detect the motor's vibrating action, a tilt sensor is implemented. A Blynk IoT GUI is used to monitor the measured data, which reflects the health of the machine. When a value measured and uploaded to the Blynk cloud exceeds the specified average, a message from the Blynk application will be delivered to users to inform them of a specific issue. Using a Heltec LoRa ESP32 to communicate with the Blynk IoT platform may limit system scalability.

J. Linggarjati [10] discussed the implementation of hydraulic cylinder temperature monitoring of heavy equipment such as excavators in the field. To monitor the temperatures, DS18b20 sensors were used to sense the hydraulic arms on the excavator. With the movement of the hydraulic arms, WiFi technology is used to transmit temperature data to the browser. ESP-NOW was used to reduce the electrical power drained from the LiPo battery. The data are recorded in the browser environment. Displaying temperature data in a browser environment may limit the conductivity and usability of the data for users in the field.

In this context, the proposed IIoT system for monitoring and controlling plastic bottle recycling plants is a strategic solution to address specific challenges in the recycling industry. The system includes a group of sensors to monitor parameters such as water quality, temperature, cruelty, pH and gas detection, ensuring compliance with environmental standards and improving recycling processes. Motors such as water pumps and lamps enable the remote control and automation of vital processes within the factory. The sensors and actuators are connected to ESP32 [11], [12], and communication via the ESPNOW protocol [13], [14] collects and transmits data for monitoring via the Blynk platform [15]and the developed GUI.

The choice of plastic bottle recycling plants as a case study for the implementation of the IIoT is the motivation behind the unique requirements of the industry, including the need for accurate monitoring of water quality, effective use of resources, and commitment to sustainable practices. By publishing the Internet of Things system in this context, the goal is to show the effectiveness of the system in enhancing operational efficiency, ensuring environmental compliance, and enabling data-based decisions in recycling operations.

In the subsequent sections of this paper, the methods and materials used in developing the industrial internet system, the results and discussion of the system performance and results, and the conclusion that highlights the main results and effects are presented. This organized approach will provide a comprehensive view of the proposed IIoT system design and its impact on factory monitoring and control, providing insight into the possible benefits of internet IoT technologies in industrial environments.

II. METHODS AND MATERIALS

The methodologies and activities used in this study are described extensively in this section, along with the complete design and implementation of the suggested transformation to a smart factory using the IIoT system to monitor and control the factory, as shown in the methodology block diagram in Fig. 1. In addition, to support the design goals, a detailed explanation of the component selection and integration procedures is provided.

A. *Effective strategies used in design and implementation*

The research employed a range of methodologies and techniques for various aspects of the study:

1) *Choosing hardware components*

a) Sensor selection: several sensors are integrated to successfully monitor multiple parameters within an industrial environment [16]. Specifically, the study incorporated a gas detection sensor, motion sensors, water quality sensors, and voltage/current sensors to provide full monitoring abilities. The component selections and how they are used with the research objectives and requirements of the smart factory environment are as follows:

- Temperature and Humidity Sensor: SHT31 sensors are designed for measuring humidity and temperature [17], [18]. This sensor was chosen due to its accuracy and high accuracy in measuring temperature and humidity levels. In a smart factory environment, maintaining ideal temperature and humidity conditions is critical to the efficient operation of machinery and equipment.
- MiCS-5524 is a dependable sensor for detecting indoor carbon monoxide and natural gas leaks [19]. The selection of the MiCS-5524 gas sensor aligned with the research objectives and requirements of the smart factory environment by providing a critical capability to monitor air quality and detect potentially harmful gases.
- Flame Sensor: This module is sensitive to both radiation and flame and can identify familiar light sources between 760 nm and 1100 nm in wavelength [20]. By integrating a flame sensor into a smart factory environment, fire detection and prevention systems can be enhanced, thus improving safety protocols and reducing the likelihood of fire-related emergencies.
- Motion sensors: These sensors are used to detect movement and control lights. The security and energy efficiency of the plant can be improved by automating lighting based on occupancy. Align to optimize energy use and improve worker safety [21].
- Water Quality Sensors: Selected to monitor water parameters such as turbidity, temperature, and pH. The goal of ensuring water quality in industrial processes and preventing pollution should be met [22], [23], [24].
- Voltage/current sensors: used to monitor electrical parameters in transmission motors. It helps optimize energy consumption, prevent equipment downtime, and ensure smooth plant operations [25].

b) Microcontroller selection: The ESP32 microcontroller was specifically selected for its advanced data processing capabilities, seamless connectivity features, and compatibility with the Blynk server platform, facilitating remote monitoring and control functions in addition to offering the industry's most acceptable electronic integration, low power consumption, and high connectivity performance [11], [12], [26].

Arduino UNO Board: Used for specific control tasks within the system. The Arduino UNO complements the ESP32 by providing additional control functions and enhancing the overall performance of the system [27].

2) *Software development*

a) Arduino IDE: The software is an open-source resource whose environment consists of two basic components: the editor and the compiler. The editor is used to generate the basic code, while the compiler compiles the code and uploads it to the Arduino. Both C and C++ are supported by this environment. It is an official software provided by Arduino.cc [28].

b) Blynk platform: This platform provides a foundation for the Internet of Things, with the ability to remotely control devices, monitor sensor data, store data, and visualize it, among other things. It provides a comparable API and user interface for all supported devices and devices. It provides cloud connectivity using WiFi, Bluetooth, BLE, Ethernet, USB (serial), and GSM [15].

c) The use of Python is becoming increasingly popular among data scientists and programmers. Python has many uses, including website creation, database access, and desktop graphical user interfaces. It contains several frameworks and libraries [29].

PyQt is a set of Python libraries that allows the use of the Qt application framework from the Qt company. Qt is a cross-platform C++ programming toolkit that includes everything needed to create programs for Windows, MacOS, Linux, Android, and embedded platforms. Using the PyQt framework, user interfaces that support (SQL databases, 2D and 3D graphics, network communications, multimedia, and more) can be developed [30].

3) *Data Processing and Analysis[31]:*

a) Data Handling

b) Visualization

c) Field Testing

4) *User Interface Design:*

a) Graphical User Interface Development: This study focused on developing graphical user interfaces (GUIs) that provide an intuitive and interactive platform for users to monitor and control industrial processes remotely.

b) Mobile Application: The mobile application interface has been meticulously designed to enable users to have seamless access to the Industrial Internet of Things (IIoT) system via their smartphones, enabling on-the-go monitoring capabilities.

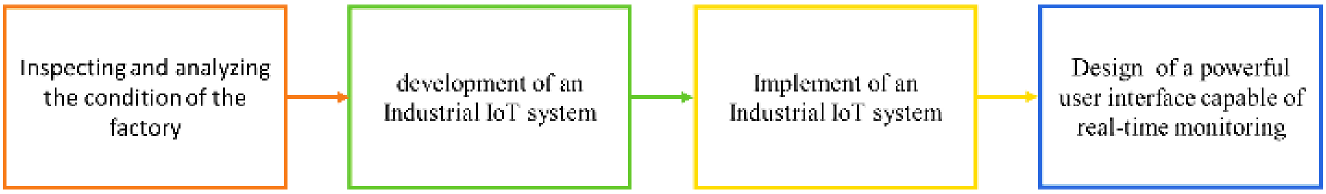


Fig. 1. Methodology block diagram

B. Proposed Industrial IoT Architecture

Architecture is a high-level, abstract description that helps identify problems and challenges in many application situations. When developing an IIoT architecture, modularity, extensibility, scalability, and interoperability across heterogeneous devices using various technologies must all be considered and prioritized [32], [33]. As shown in Fig. 2, an architecture for the IIoT has been designed to include four layers.

1) Physical Layer:

This layer includes all the physical components of the production system, including sensors, actuators, and devices deployed in the industrial environment[34], [35]. Sensors are responsible for collecting data on various parameters, such as temperature, humidity, gas levels, water quality, and motion. Actuators are used to control physical processes based on data collected from sensors.

2) Network and communication layer:

This layer connects the hardware and the middleware, facilitating two way data flow [36], [37]. The components of this layer include servers for storing data, routers for transferring data, controllers for system management, and communication protocols for seamless communication. Communication protocols such as the UART wired communication and ESPNow protocols are used for efficient data transfer and communication [14], [26], [38].

3) Middleware Layer:

This layer establishes the reliability and strength of the IIoT system by providing features such as device discovery and management, big data analytics, and security mechanisms[39]. The data collected from the sensors are processed, analyzed, and stored in this layer before being forwarded to the upper layer for user access [40].

4) Application Layer:

The top layer of the architecture provides services to users through various interfaces, such as desktop applications, mobile applications, and web platforms. Users can remotely monitor and control industrial devices, visualize real-time data, set alerts and notifications, and access historical data for analysis and decision-making [41].

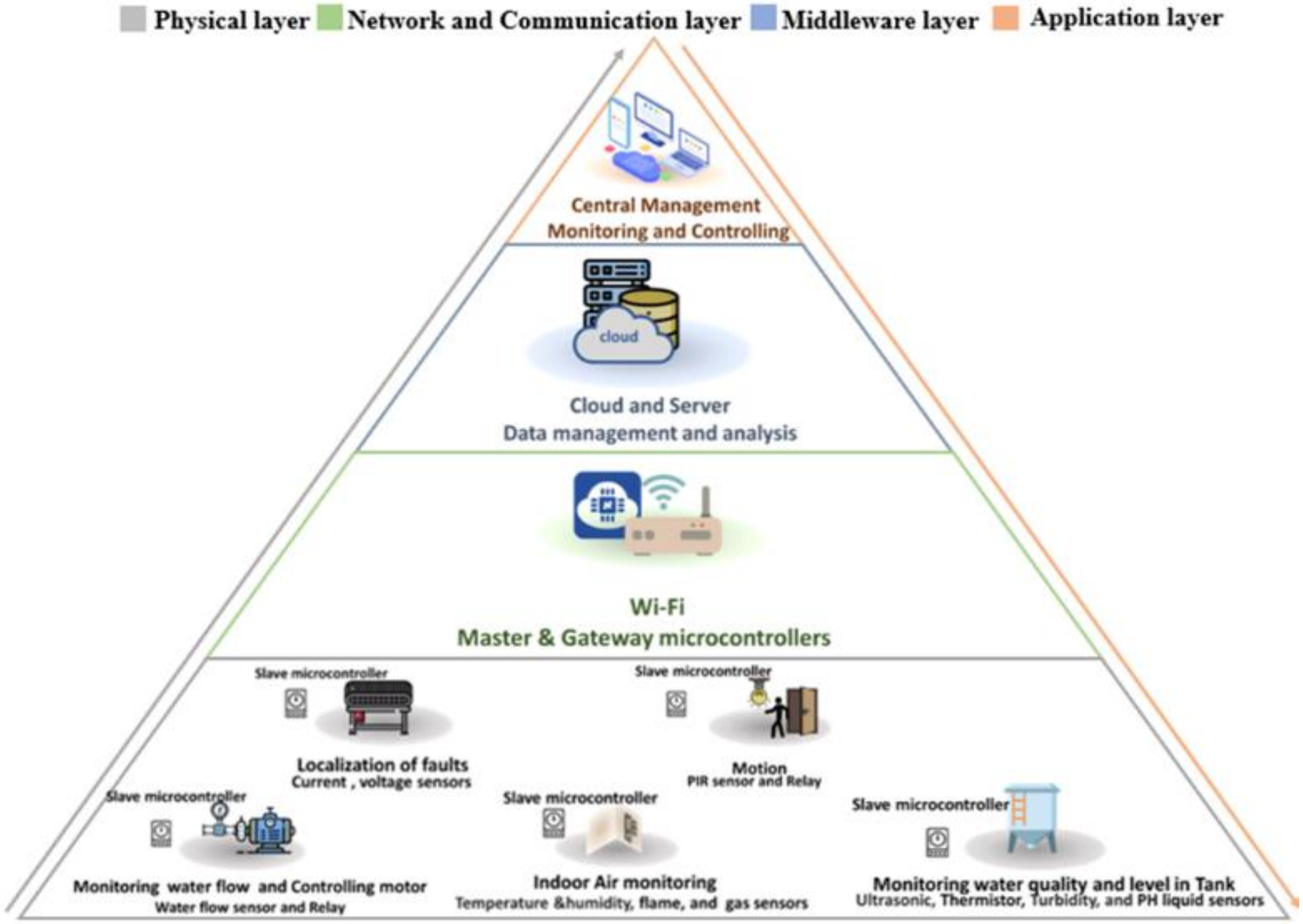


Fig. 2. Proposed IIoT system architecture

C. Comparative Analysis of Communication Protocols: ESP NOW

Communication protocols play a crucial role in enabling efficient and reliable data transfer between devices in different applications [42]. ESP NOW is a lightweight wireless communication protocol developed by Espressif Systems specifically for Internet of Things (IoT) applications. Due to its unique features and capabilities, the ESP NOW has gained popularity in recent years [43]. In this analysis, Table I shows the main advantages and limitations of the ESP NOW.

TABLE I. MAIN ADVANTAGES AND LIMITATIONS OF ESP-NOW

Advantages	Limitations	Ref
Low energy consumption: One of the main advantages of ESP NOW is its low power consumption. The protocol was designed with energy efficiency in mind, making it highly suitable for battery-powered devices and IoT applications.	**Single-hop communication:** ESP NOW supports only one-hop communication, which means devices can connect directly to a single destination device. While this is suitable for point-to-point communications within a limited range, it may not be well suited for applications requiring multihop or mesh networking capabilities.	[44], [45], [46], [47], [48]
Simple and lightweight: ESP NOW is known for its simplicity and lightweight nature. Compared to other communication protocols, it has a smaller memory footprint, making it ideal for devices with limited resources. The compact design of ESP NOW reduces overhead and communication delays, resulting in efficient data transfer.	**Wifi Interference:** Since ESP NOW relies on WiFi technology, it may be susceptible to interference from other WiFi devices operating in the same frequency band	
Low Latency: ESP NOW provides low-latency connections, ensuring data is transmitted and received quickly. This is critical in many real-time applications where timely response and minimal delay are needed. **High reliability:** ESP NOW is designed to provide reliable communications in challenging wireless environments. It uses a combination of acknowledgments and retransmission mechanisms to ensure data integrity and robustness.	**Limited scope:** One limitation of ESP NOW is the limited connection range. Due to its use of WiFi frequencies (2.4 GHz) and low-power operation, ESP NOW's communication range is typically limited to a few hundred meters. This limitation may limit its applicability in some scenarios involving long-range communications.	

D. Development and Implementation of the IIoT System

The system comprises seven ESP nodes that are dispersed across the factory at various locations. The first two nodes are installed in the interior, one for air quality monitoring and a fire warning system that detects and reports flames and one that detects motion and controls lights. The quantity and quality of water are monitored at the third node. The fifth node is used to regulate the water pump, while the fourth node is deployed for fault localization. The sixth and seventh ESP nodes are run as both gateways. The proposed system was verified through several procedures: Conducting a functional test to ensure that the system performs the intended functions accurately. Each component, sensor, and actuator were tested to ensure that they worked as expected. The ability of the system to accurately monitor parameters such as temperature, humidity, gas detection, water quality, and other relevant metrics was verified. Conducting performance testing to check system responsiveness, data processing speed, and overall efficiency—measures system response time to sensor inputs, data transfer speed, and real-time monitoring capabilities.

- Purpose Node 1 is responsible for fire detection and notification. Flame sensors are used to detect flames and trigger alarms in the event of a fire incident. Flame sensors are connected to the node to monitor the presence of fire. When a flame is detected, the node sounds an alarm and sends notifications. Node 1 interacts with the gateway node to transmit fire detection information and alarm signals for centralized monitoring and control. The selection of flame sensors is critical for ensuring timely fire detection and response in industrial environments. Using flame sensors and a reliable communication protocol such as ESP NOW, Node 1 enhances the system's safety and emergency management capabilities.

Node1 implemented various sensors interfaced with an ESP32 board, such as the (SHT31 [17], [49], flame [20], [50], and [51] MICS-5524 5524 [19]) sensors. In addition, an LCD screen is integrated into the system to display the temperature and humidity. A buzzer and RGB LED are also integrated for notification purposes. When a flame is detected, the system sends a notification to the supervisors, activates the warning red signal and triggers the alarm to notify people inside the factory. Fig. 3 shows the circuit diagram of ESP Node 1. Since the buzzer draws more current than the ESP pins can handle a transistor, the buzzer can be powered from a different voltage supply than the ESP. Additionally, each color LED is connected to a 220-ohm resistor to act as a current limiter. Node1 and Pseudocode 1 depict the node1 code.

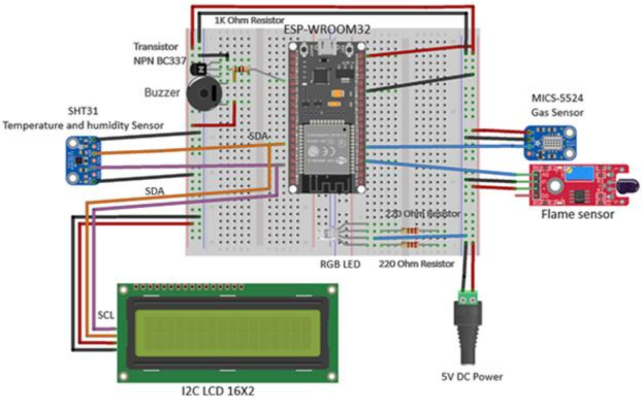


Fig. 3. Circuit diagram of ESP node 1

Pseudocode 1. Environment Monitoring Code

```
//Initialization
DEFINE SHT31Pin, GASPin, FlamePin INPUT;
DEFINE RGB, BuzzerPin, LCD OUTPUT;
SET BuzzerPin, FlamePin, RGBPin, GASPin
SET Coordinator ESP MAC address
SET LCD, PWM.Frequency, PWM.resolution
IF (ESP.initialization != ESP_OK) THEN
    Display (Error initializing ESP);
    Return;
ENDIF
IF (ESP.peer != ESP_OK) THEN
    Display (Failed to add peer);
    Return;
ENDIF
WHILE (TRUE)
    Input: Temperature = READ (sht31.temp) ;
    Input: Humidity = READ (sht31.Hum) ;
    Output: WRITE (LCD, Temperature, Humidity)
    Input: Flame = READ ( FlamePin );
    Input: GAS = READ ( GASPin );
    IF (Flame == HIGH OR GAS == HIGH) THEN
        Output: WRITE (BuzzerPin, HIGH)
        Output: WRITE (RGB, RED)
    ELSE
        Output: WRITE (BuzzerPin, LOW)
        Output: WRITE (RGB, GREEN)
    ENDIF
    JSON_data = <Temperature, Humidity, Flame, GAS>;
    Output: WRITE (EPS, JSON_data)
    Display (JSON_data);
    Delay (4Sec);
ENDWHILE
```

- Node 2 focuses on air quality monitoring and motion detection. It uses sensors to measure motion detection parameters for security and energy saving purposes. Motion sensors are connected to Node 2 to monitor motion detection. Node 2 communicates the sensor data to the gateway node for centralized monitoring and control. It may trigger actions based on motion detection, such as controlling lights. The selection of motion sensors aligns with the system's goals of ensuring a safe and healthy work environment. By integrating these sensors with Node 2 and using ESP NOW for communication, the system improves security monitoring and procedures.

Node 2 uses a relay [52] and PIR motion sensor, [53] which are connected to ESP32. A relay and controller with ceiling-mounted PIR sensors are used. In addition, the system switches in the room to turn lights on and off. The lights can be controlled remotely and autonomously using a motion sensor. Fig. 4 shows the second node circuit using the PIR sensor and relay that are connected to ESP32. Pseudocode 2 depicts node 2 code.

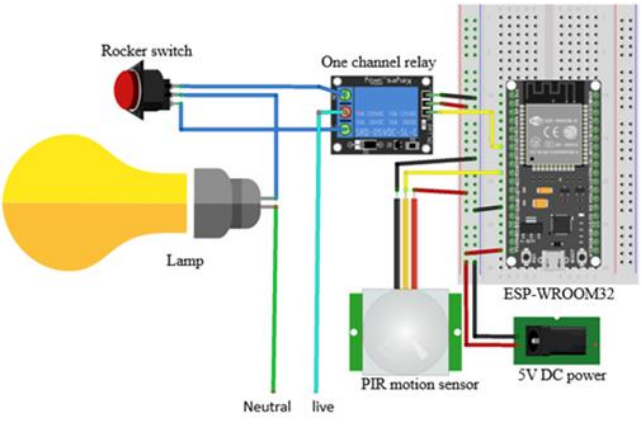


Fig. 4. Circuit diagram of node two control lights

Pseudocode 2. Motion Monitoring and Control Light Code

```
//Initialization
DEFINE PIRPin, INPUT;
DEFINE RelayPin OUTPUT;
SET PIRPin,RelayPin
SET Coordinator ESP MAC address
IF (ESP.initialization != ESP_OK) THEN
    Display (Error initializing ESP);
    Return;
ENDIF
IF (ESP.peer != ESP_OK) THEN
    Display (Failed to add peer);
    Return;
ENDIF
WHILE (TRUE)
    IF(! ERROR)
        IF(Led_state==" V0_ON")
            Output: WRITE (RelayPin, HIGH)
        ELSEIF(Led_state=="V0_OFF")
            Output: WRITE (RelayPin, LOW)
        ENDIF
    ENDIF
    Input: MOTION = READ ( PIRPin );
    Output: WRITE (MOTION)
    JSON_data = <Led_state, MOTION>;
    Output: WRITE (EPS, JSON_data)
    Display (JSON_data);
    Delay (1Sec);
ENDWHILE
```

- Node 3 is designed to monitor water quality parameters such as water temperature, turbidity, and pH. The quality of water should be considered to ensure the use of safe and effective water. The choice of water quality sensors is very important for maintaining water safety standards and quality in industrial processes. ESP32 allows accurate monitoring and control of water parameters.

Node 3 The implemented system consists of four sensors: a pH liquid sensor [54], turbidity sensor [55], ultrasonic sensor [56], and DS18B20 sensor [57], [58]. Fig. 5 shows the circuit diagram for ESP node 3, where the pH liquid, turbidity, and ultrasonic sensors are connected to the ESP pin. The DS18B20 sensor has three wires connected to a 4.7k pull-up resistor between the signal and the power pin to keep the data transmission stable and to the ground. Pseudocode 3 depicts node 3 code.

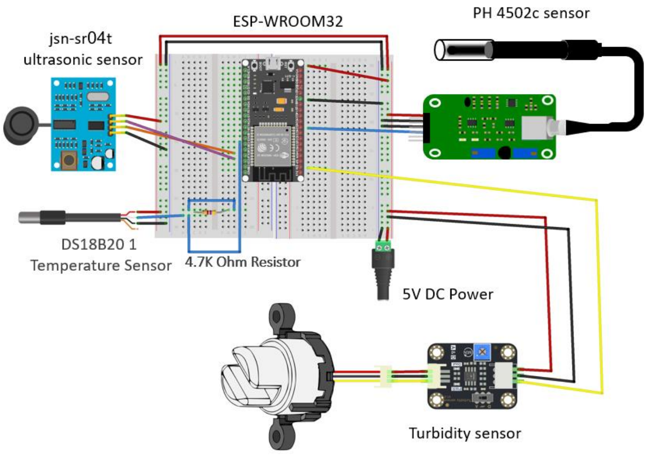


Fig. 5. Circuit diagram of ESP node 3

Pseudocode 3. Water Quality and Level Monitoring Code

```
//Initialization
DEFINE echoPin ,Input;;
DEFINE trigPin  Output;
SET DS18B20 , pH ,echoPin,trigPin;
SET Duration, Distance,turbidity,adc_resolution ,sample, measurings;
SET Coordinator ESP MAC address
IF (ESP.initialization != ESP_OK) THEN
        Display (Error initializing ESP);
        Return;
ENDIF
IF (ESP.peer != ESP_OK) THEN
        Display (Failed to add peer);
        Return;
ENDIF
WHILE (TRUE)
        DS18B20.requestTemperatures();
        Input: TempC = DS18B20.getTempCByIndex(0);
        Output: WRITE (TempC)
        Output: WRITE (trigPin,LOW)
        Delay (2MicroSec);
        Output: WRITE (trigPin,HIGH)
        Delay (20MicroSec);
        Output: WRITE (trigPin,LOW)
        Input: Duration =READ pulseIn(echoPin, HIGH);
        Input: Distance = READ (Duration / 2) * 0.034
        Input: SensorValue = READ (TurbidityPin);
        Input: Turbidity =READ map(SensorValue,0,2800,25,1);
        Input: measurings +=READ (pHSense);
        Input pH= READ (3.3 / adc_resolution * measurings/samples;)
 JSON_data = <TempC, Distance, Turbidity, pH>;
Output: WRITE (EPS, JSON_data)
Display (JSON_data);
Delay (1Sec);
ENDWHILE
```

- Node 4 is designed to measure the voltage and current in a plant's electrical system. It helps to monitor power consumption and ensure electrical safety. Voltage and current sensors are connected to Node 4 to measure the electrical parameters. Node 4 transmits voltage and current data to the gateway node to monitor power usage and detect faults. Alerts may be triggered in the case of power fluctuations. The selection of voltage and current sensors is critical for maintaining the integrity of the electrical system and preventing power-related accidents. By integrating these sensors with Node 4 and using ESP NOW for communication, the system improves energy management and electrical safety within the plant.

Node 4 was implemented using A voltage [59] and a current sensor [60] for three phases. All the sensors are interfaced with an Arduino UNO microcontroller to read and transmit the measured data via UART communication [38] with ESP32, after which the serialized data are converted to JSON and serialized via the ESPNow protocol [14] to the gateway. Fig. 6 shows a circuit diagram for ESP Node4. Pseudocode 4 depicts the node 4 Arduino code, and Pseudocode 5 depicts the node 4 ESP code.

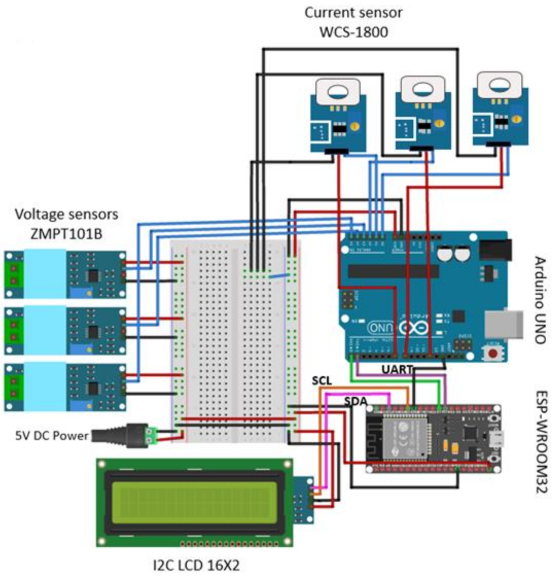


Fig. 6. Circuit diagram of ESP node 4

Pseudocode 4. Fault Localization Monitoring Arduino_Code

```
//Initialization
DEFINE CurrentPin, VoltagePin, Input;
SET currentSensor, voltageSensor;
SET voltageSampleRead, voltageLastSample, voltageSampleSum, voltageSampleCount;
SET voltageMean; RMSVoltageMean, adjustRMSVoltageMean; FinalRMSVoltage, voltRead;
SET EnergyMonitor energyMonitor;
WHILE (TRUE)
        Input:READ energyMonitor1.current(CurrentPin, 0.125);
        Input: Irms =READ energyMonitor1.calcIrms(1484);
        Input: Current = READ( Irms * voltRead);

        IF (micros() >= voltageLastSample2 + 1000)
                Input :voltageSampleRead = (READ(voltageSensor) - 512) + voltageOffset1_1;
                Input:    voltageSampleSum = voltageSampleSum + sq(voltageSampleRead)
                Input  : voltageSampleCount = voltageSampleCount + 1;
                Input:    voltageLastSample = micros ();
        ENDIF
        IF (voltageSampleCount == 1000)
                Input:   voltageMean= voltageSampleSum / voltageSampleCount
                Input:   RMSVoltageMean =READ (sqrt(voltageMean)) * 1.5;
                Input: adjustRMSVoltageMean = RMSVoltageMean + voltageOffset1_2
                Input:FinalRMSVoltage = RMSVoltageMean + voltageOffset1_2;
                        IF (FinalRMSVoltage <= 2.5)
                                FinalRMSVoltage = 0;
                         ENDIF
                Output: FinalRMSVoltage
        ENDIF
JSON_data = <TempC, Distance, Turbidity, pH>;
Output: WRITE (EPS, JSON_data)
Display (JSON_data);
Delay (1Sec);
ENDWHILE
```

Pseudocode 5. Fault Localization Monitoring ESP_Code

```
//Initialization
DEFINE LCD Output;
SET Coordinator ESP MAC address
SET LCD
IF (ESP.initialization != ESP_OK) THEN
        Display (Error initializing ESP);
        Return;
ENDIF
IF (ESP.peer != ESP_OK) THEN
        Display (Failed to add peer);
        Return;
ENDIF
WHILE (TRUE)
        IF (Serial2.available())
                Input:READ (doc,Serial2);
                Output: WRITE (LCD, Current, Voltage)
        ENDIF
        JSON_data = < Current, Voltage >;
        Output: WRITE (EPS, JSON_data)
        Display (JSON_data);
        Delay (1Sec);
ENDWHILE
```

On the ESP side, the code started by initializing UART communication and initialization of the ESPNow protocol to send and receive data from the coordinator ESP. The initialization of the coordinator ESP MAC address is carried out next. Finally, the JSON data are broadcast to the coordinator

- Node 5 is dedicated to monitoring the water flow rate in the pipes inside the factory. It helps track water usage and detect anomalies in the water supply system. Water flow sensors are connected to node 5 to measure the flow rate in the pipes. Node 5 communicates the water flow data to the gateway node for real-time monitoring and analysis. The water pump can also be controlled based on flow rate readings. The use of water flow sensors is essential for optimizing water usage and detecting leaks or blockages in piping systems. By integrating these sensors into Node 5 and using ESP NOW for communication, the system improves water management efficiency and reduces waste.

Node 5 consists of a water flow sensor [61], [62]. and a relay connected to ESP32 pins. Fig. 7 shows a circuit diagram for ESP Node4. Pseudocode 6 depicts node 5.

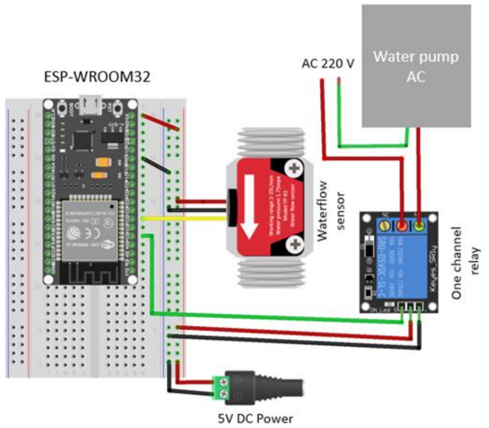


Fig. 7. Circuit diagram of ESP node 5

Pseudocode 6. Waterflow Monitor and Control Waterpump Code

```
//Initialization
DEFINE WaterflowPin, INPUT;
DEFINE RelayPin OUTPUT;
SET WaterflowPin, RelayPin;
SET Coordinator ESP MAC address
IF (ESP.initialization != ESP_OK) THEN
        Display (Error initializing ESP);
        Return;
ENDIF
IF (ESP.peer != ESP_OK) THEN
        Display (Failed to add peer);
        Return;
ENDIF
WHILE (TRUE)
        IF(! ERROR)
                IF(Motor_state==” V0_ON”)
                        Output: WRITE (RelayPin, HIGH)
                ELSEIF (Motor_state ==”V0_OFF”)
                        Output: WRITE (RelayPin, LOW)
                ELSE
                ENDIF
        ENDIF
        Input: CurrentMillis = READ millis;()
        IF (CurrentMillis - PreviousMillis > interval)
                Input: pulse1Sec = pulseCount;
                Input:pulseCount = 0;
                Input: FlowRate = ((1000.0 / (millis () - PreviousMillis)) *
pulse1Sec) / calibrationFactor;
          JSON_data = < Motor_state, FlowRate >;
        Output: WRITE (EPS, JSON_data)
        Display (JSON_data);
        Delay (1Sec);
ENDWHILE
```

In communication implementation, six boards communicate using the ESP NOW protocol, and they are all configured using the many-to-many communication protocol so that each node can send data to the other five ESP nodes. All the data are sent to nodes in JSON format. The gateway has two ESP32 boards, one of which is used for coordination. This board is used to transmit ESP NOW data from ESP nodes to Master_ESP and vice versa. In addition, the other Master_ESP receives JSON data from Coordinator_ESP, deserializes the JSON data, and sends the data to the Blynk Cloud and GUI, and vice versa. In addition, it is the only node that is connected to the internet. Fig. 8 shows the connection methods by which ESP gateway boards communicate using the UART wire communication protocol and the other nodes communicate with the gateway using the coordinator ESP MAC address and specified channel number. Pseudocode 7 depicts gateway.

The selection of the ESP NOW communication protocol for establishing a many-to-many communication setup within the IIoT system was driven by the need for a robust, low-power, and efficient communication mechanism suitable for industrial applications. By implementing a many-to-many communication configuration, where each node can communicate with multiple other nodes, the system enhances flexibility and scalability. This setup allows seamless data exchange between different components of the factory monitoring system, enabling comprehensive monitoring and control capabilities across various sensors and actuators.

Pseudocode 7. Coordinator_ESP Code

```
//Initialization
DEFINE Temperature, Humidity,Gas,Flame, WaterTemperature,waterLevel,Turbidity ,pH
DEFINE Current1, Current2, Current3,Voltage1,Voltage2,Voltage3,Waterflow Rate,Motion
SET Coordinator ESP MAC address
IF (ESP.initialization != ESP_OK) THEN
        Display (Error initializing ESP);
        Return;
ENDIF
IF (ESP.peer != ESP_OK) THEN
        Display (Failed to add peer);
        Return;
ENDIF
WHILE (TRUE)
        IF (Serial2.available())
                Input:READ (doc,Serial2);
                        IF (result == ESP_OK)
                        ELSEIF
                        Delay (50MicroSec);
                        ENDIF
        ENDIF
ENDWHILE
```

The role of gateway nodes in coordinating communication among ESP nodes is crucial for ensuring seamless data flow and system reliability. The gateway nodes act as central points for data aggregation, processing, and distribution, facilitating efficient communication between the ESP nodes and external interfaces such as the Blynk platform. To address data security and integrity concerns during communication between nodes and the gateway, several measures are implemented within the system. Encryption techniques are employed to secure data transmission and prevent unauthorized access or tampering. By encrypting data packets exchanged between nodes and the gateway, the system ensures the confidentiality and integrity of sensitive information, safeguarding against potential cyber threats. Furthermore, authentication mechanisms are implemented to verify the identity of nodes and ensure that only authorized devices can participate in the communication network. By authenticating nodes based on predefined credentials or keys, the system mitigates the risk of unauthorized access and maintains the integrity of the communication infrastructure.

Integrating the proposed IIoT system into a smart factory environment involves a harmonious combination of different hardware and software components to create a functional monitoring and control system. The integration of sensors and actuators is a critical aspect that requires attention to ensure accurate data collection. This includes proper placement, calibration, and connecting of the sensors to the microcontrollers. The integration of actuators, such as water pumps and lamps for control purposes, requires seamless communication with microcontrollers. It is necessary to conduct tests on operating mechanisms and evaluate their response to commands to ensure effective control of industrial systems. In addition to hardware integration, software integration plays a vital role in the successful implementation of an IIoT system. Programming microcontrollers, such as the ESP32, to collect sensor data, process information, and communicate with a server platform, such as Blynk, is an important aspect of software integration. The reliability and efficiency of the code are critical to the function of the system. Configuring that the server platform receives, stores, and displays data from sensors and microcontrollers is essential for real-time monitoring and control. Implementing features such as alerts, notifications, and data visualization tools on the platform improves the usability of the system. The implementation of reliable wireless communication protocols, such as ESPNow, between system components is essential for seamless data transfer. Ensuring stable connectivity and maintaining data integrity over wireless networks is critical for optimal system performance. Furthermore, incorporating encryption and secure communication protocols to protect data exchanged between components and the server platform is vital for protecting sensitive information within the smart factory environment. There are also challenges and considerations [63], [64], [65]

1. Interoperability issues: Ensuring compatibility and interoperability between different hardware and software components can be difficult. Addressing communication protocols, data formats, and integration complexities is critical to the smooth operation of the system.

2. Data accuracy and calibration: Ensuring the accuracy and calibration of sensors to collect accurate data is challenging. Regular calibrations, sensor maintenance, and data validation are essential to maintain data accuracy.

Scalability and expandability: It is important to consider whether the system can be expanded to accommodate future expansions, upgrades, or additional sensors/actuators. Planning for scalability and flexibility of the system during the integration process is key to adapting to changing requirements.

E. Graphical User Interface

The proposed system includes two user interfaces and a phone application to work with several different settings and locations.

1) Local GUI

The user interface is built in the Python programming language and the PyCharm IDE using the PyQt5 toolkit. The user interface allows users to use the application without the necessary programming experience; it contains a window that displays the data received from the various sensors in addition to displaying the time, date, current weather, and calendar as well as a real-time plot to visualize any change.

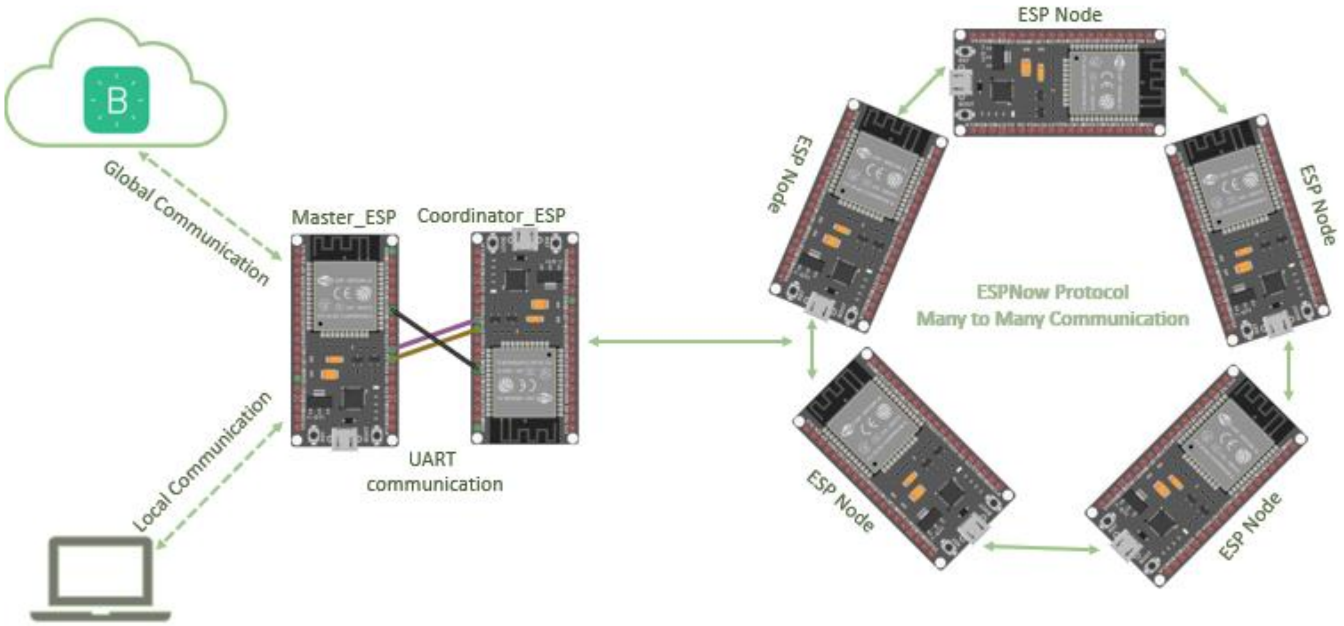


Fig. 8. Communication infrastructure of the proposed system

2) *Blynk Platform*

The GUI is designed based on the Blynk IoT platform. It provides a way for IIoT technology to handle user commands and display system status, with the ability to remotely monitor sensor data, store and visualize data, and accordingly control system hardware. The Blynk platform provides a mobile app.

GUI design includes principles, usability testing methodologies, and the incorporation of user feedback to enhance the user experience within an Industrial Internet of Things (IIoT) system.

a) Principles of Graphic User Interface Design

- User-Centered Design: Prioritizing user needs and preferences in GUI design ensures a user-friendly interface. User workflow, information hierarchy, and visual clarity enhance the ease of use.
- Consistent Layout: Maintaining a consistent layout with clear navigation, naming, and visual elements across the GUI enhances user familiarity and ease of interaction.
- Visual Feedback: The visual cues of user actions, system responses, and data updates enhance user understanding and interaction with the IIoT system.
- Responsive Design: Create a responsive GUI design that adapts to different screen sizes and devices, ensuring accessibility and ease of use across desktop and mobile platforms.

b) Methods of usability testing:

- Prototype Testing: Conducting usability tests using GUI prototype designs allows for early feedback on layout, functionality, and user interactions. Iterative testing helps identify and address usability issues.
- Task-based testing: Evaluate the GUI through task-based scenarios where users perform common actions such as monitoring sensor data, controlling motors, and setting alerts, providing insights into ease of use and efficiency.
- User observation: Observing users interacting with the GUI in real time helps identify navigation challenges, points of confusion, and areas for improvement in the interface design.
- Feedback surveys: After usability testing sessions, user feedback is collected through surveys or interviews, and qualitative insights about user preferences, difficulties, and suggestions for GUI improvements are collected.

c) Integration of user feedback:

- Iterative Design Process: Incorporating user feedback into the GUI design process through iterative cycles of testing, collecting feedback, and improving the design ensures that user needs are effectively met.
- Prioritize Features: Using user feedback to prioritize GUI features, functionality, and improvements based on user preferences and pain points improves the overall user experience.
- Continuous improvement: Establish a feedback loop mechanism where user suggestions, bug reports, and improvement requests are regularly reviewed and implemented, ensuring continuous improvement of the GUI.
- User Training and Support: Providing user training sessions, documentation, and support resources based on common user feedback helps users navigate the GUI effectively and maximizes its usefulness in monitoring and controlling the IIoT system.

III. RESULTS AND DISCUSSION

Several operational scenarios are implemented to verify the overall effectiveness of the proposed system. The first experiments were performed separately for each subsystem. This is to test the operation of each subsystem individually; the second step is to distribute the subsystems at different locations in the factory so that the distance from each subsystem to the gate is 23 meters. The gateway is positioned to receive data from all five end nodes. Communication between nodes and gateways occurs via MAC addresses. Additionally, they all share the same channel number.

A. *Indoor System*

There are three alarm technologies in this node. The first is to sound a buzzer alarm and change the LED color from green to red when a flame is detected. The second way is through a user interface designed to display a message about a fire in a factory. The third is a notification about the Blynk app. The second node has a motion sensor and a relay to control the illumination. In addition to the lamp's on/off switch and motion sensor, which detects movement in the room and displays a message in the Blynk app's user interface, the light may also be turned on or off via the Blynk application. According to Fig. 9, nodes 1 and 2 are both installed within the factory's office.

Fig. 9. Indoor systems: (a) node 1, (b) node 2, and (c) overview in a side office

B. Outdoor Systems

Node 3 Before the actual installation stage, each sensor was tested independently. The water temperature sensor responded well by providing information. The pH sensor was subsequently tested using five different samples, namely, lemon, coffee, water, baking soda, and bleach. measured and are displayed in Fig. 10, which compares the common scale [66] and the sensor reading

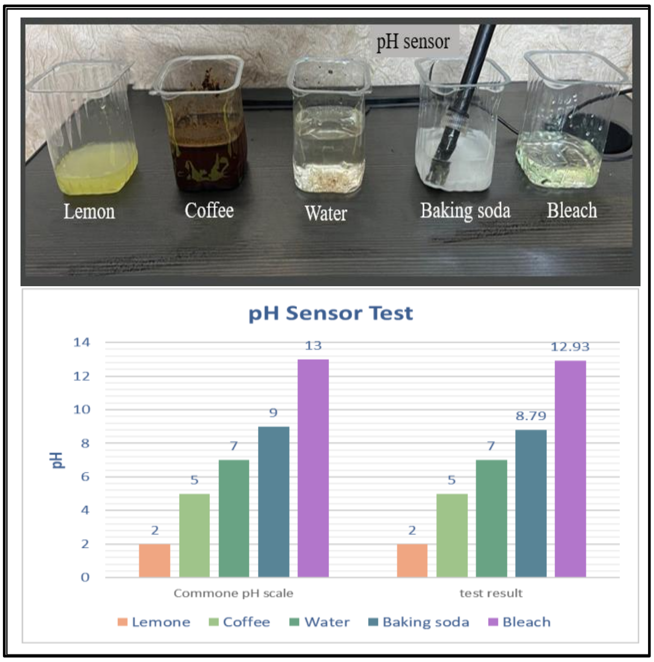


Fig. 10. pH sensor testing results

An ultrasonic sensor was used to determine the water level. Fig. 11 shows a chart that compares the actual distance measurement manually and with the sensor. The average inaccuracy from five studies at various distances was 2 cm. If the distance is less than 22 cm, the sensor will maintain a distance of 22 cm.

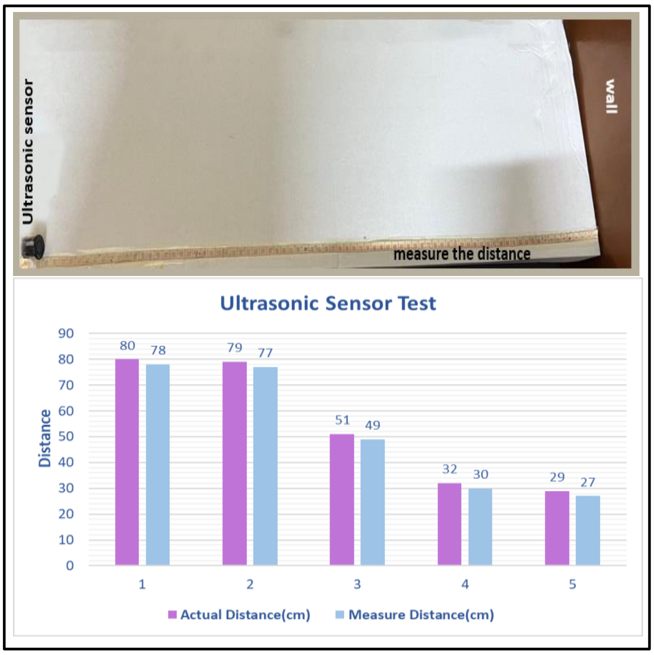


Fig. 11. Ultrasonic sensor testing results

Then, the turbidity sensor was tested using four clear, cloudy, dirty, and muddy water samples. The test results are shown in Fig. 12 and were consistent with expectations.

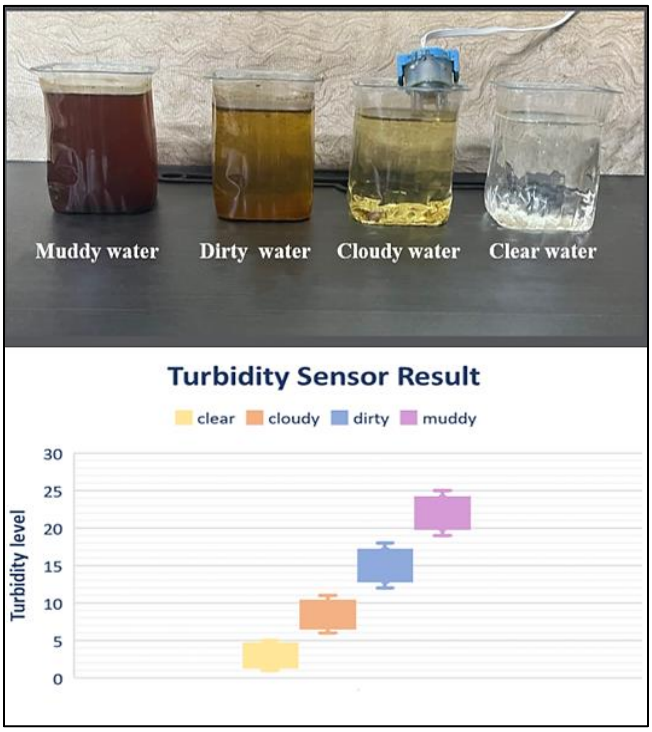


Fig. 12. Turbidity sensor testing results

Fig. 13 shows the test results for the water temperature sensor. The test results for the water temperature sensor indicate that it is a reliable and accurate component of an IIoT system that is capable of providing accurate temperature measurements to monitor industrial processes and ensure optimal operational efficiency in smart factory environments.

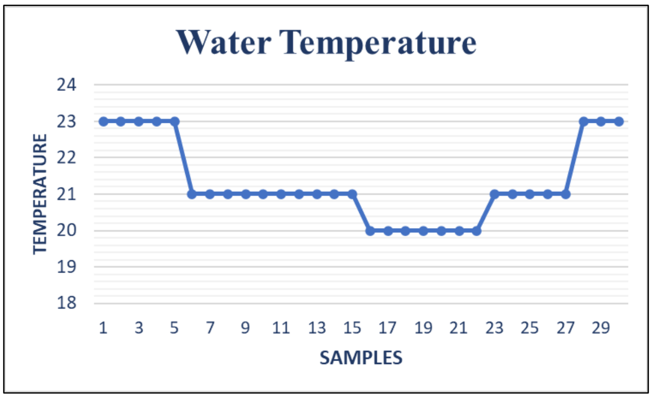


Fig. 13. Water temperature sensor test results

The sensor node was strategically placed inside a 93 cm high water tank. Various sensors, including those designed to monitor turbidity, pH, water level, and water temperature, were strategically distributed within the tank. The data collected from these sensors yielded positive results, indicating successful implementation. Fig. 14 visually represents the configuration and position of node 3 within a water tank, displaying the efficient distribution of sensors for the purposes of comprehensive data collection and monitoring in an industrial environment.

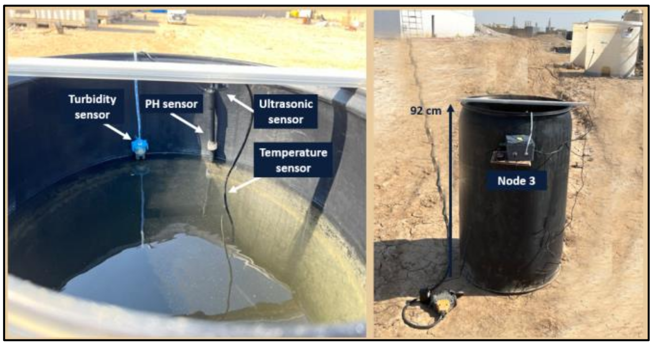


Fig. 14. Node3 implementation

Node 4 was implemented on the Convery motor (three phases) as three voltage sensors were deployed. Communication between the voltage sensor and the motor was lined to a neutral voltage to read 220 volts for each phase. The current sensor is connected to a phase in which the current moves toward the motor. The readings from the voltage sensors were accurate because the difference from the actual voltages was less than one volt. Fig. 15 shows the voltage and current sensor readings, respectively, and Fig. 16 shows the implementation of node 4.

Node 5 was tested initially indoors, and it achieved good results in controlling the on/off of the water pump. In addition, it was used for monitoring the water flow rate in the pipes. The next step was to implement the system in the factory, as shown in Fig. 17. The readings obtained from the water flow sensor are shown in Fig. 18. The results show the water flow rate in the pipes when the water pump is off and when the water pump is on.

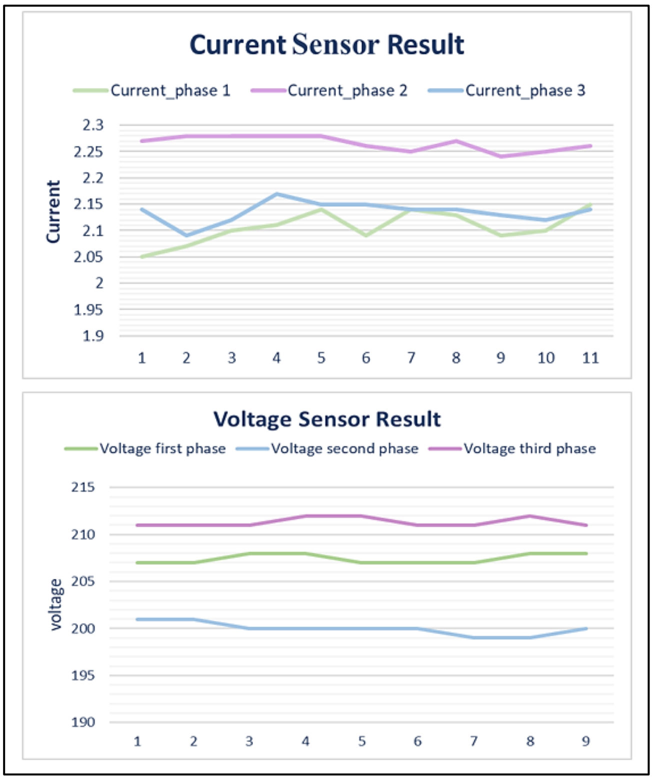


Fig. 15. Results of the current and voltage sensor tests

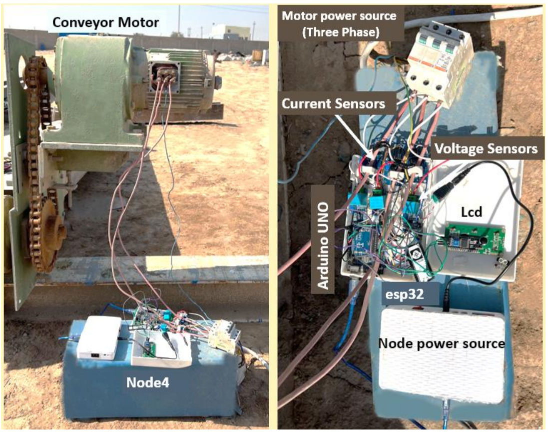


Fig. 16. Node 4 implementation

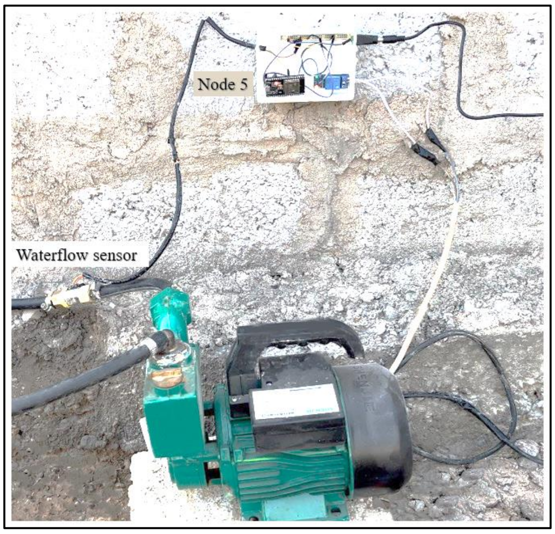


Fig. 17. Node5 implementation

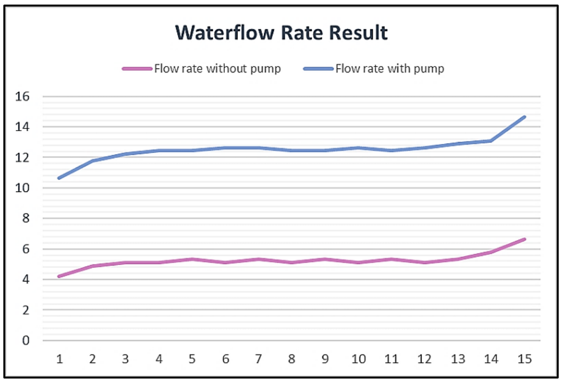
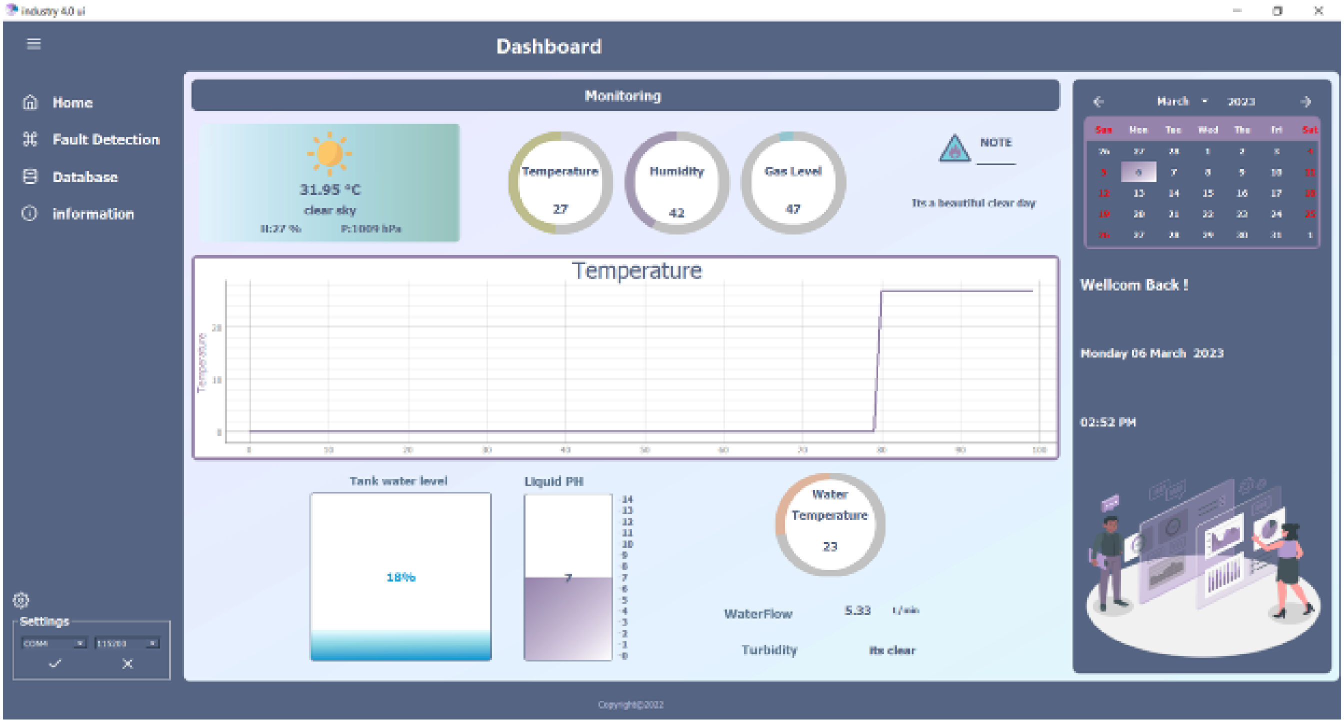


Fig. 18. Water flow sensor test results

C. *Grafical User Interface and Mobile APP*

An interactive and powerful user interface was developed in addition to responsiveness and display of results as well as serial data in CSV files. The first window displays the results of the sensors showing the temperature, humidity, and gas level and gives a message in the case of sensing the flame. In addition, Fig. 19 shows the level of water in the tank, the degree of acidity in the water, the water temperature, the degree of turbidity, and the water flow rate in the pipes. The second window displays the measured three-phase voltages and currents (see Fig. 20). The voltage change in real-time is shown.



31.95 °C
clear sky
H:27 % P:1009 hPa

Temperature 27
Humidity 42
Gas Level 47

Its a beautiful clear day

Tank water level 18%
Liquid PH 7
Water Temperature 23
WaterFlow 5.33 L/min
Turbidity its clear

Wellcom Back !
Monday 06 March 2023
02:52 PM

Fig. 19. First window result

Voltage 203 V
Current 2.38 A

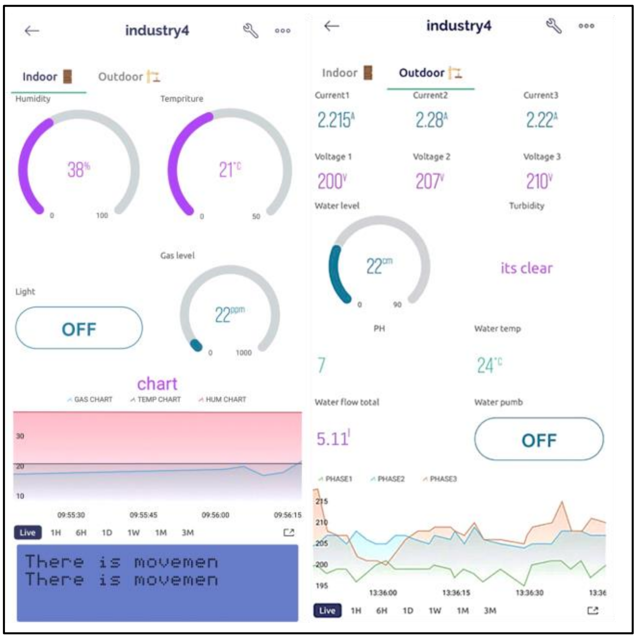
Voltage 208 V
Current 2.38 A

Voltage 214 V
Current 2.39 A

Fig. 20. Second window result

The Blynk App was used to monitor real-time data and attractively display the collected data. Blynk has a unique authentication token that allows the device to interact with the API and the Blynk platform. The readings and concentrations of sixteen of the collected parameters, such as temperature, humidity, gas, flame, movement, water flow rate, water level in the tank, water temperature, turbidity, acidity level, voltage for three phases, and current for three phases, were recorded. It showed a quick response in processing and displaying data. Fig. 21 presents the results of the Blynk App.

The current study focuses on developing an intelligent and cost-effective industrial system based on the ESP32 microcontroller and Blynk server platform to monitor and control factory operations remotely. Table II shows a comparison and analysis of the results of the current study with the results of the previous studies mentioned.



```
There is movemen
There is movemen
```

Fig. 21. Blynk mobile app results

TABLE II. COMPARATIVE ANALYSIS OF THE CURRENT AND PREVIOUS STUDIES

Ref	Method
[4], 2019	Enhance monitoring capabilities by integrating different sensors to collect more comprehensive data beyond gas detection
[5], 2019	Aim to expand beyond basic alerting functions to include more advanced monitoring and control features to enhance operational efficiency
[6], 2020	Explore alternative communication methods to ensure real-time monitoring even in the absence of a stable internet connection.
[7], 2020	Take into account scalability challenges when implementing similar systems on a larger scale for city-level water distribution networks.
[8], 2021	Address limitations in monitoring coverage to ensure a comprehensive assessment of air quality in all regions.
[9], 2022	Explore options to enhance system scalability beyond the limitations of current communications technology.
[10], 2022	Focus on improving data accessibility and usability for field users beyond browser-based data presentation.
	Present study can benefit from insights and lessons learned from previous research to enhance industrial system monitoring and control capabilities, ensure scalability and reliability and comprehensive data analysis to improve operational efficiency in factory environments.

D. Discussion

Node 1 has identified problems with the performance of the current flame sensors in the subsystem to detect and notify fires, considering the counterparts of the late responses and inaccurate readings. This phenomenon leads to a slight delay of less than two seconds in stimulating warnings and notifications, which may affect the system's ability to alert individuals to unleash risks immediately. These notes emphasize the importance of ensuring the reliability and response of flame sensors in basic safety applications within industrial environments.

Moreover, the third node faced operational challenges when exposed to direct sunlight, which led to errors in its functions. In addition, a turbid sensor is found in this node to be sensitive to the reflected light, which affects the accuracy of its readings. These environmental effects highlight the necessity of designing a strong sensor and strategic position to alleviate external factors that can influence the performance of the sensor.

In contrast, the fourth node showed accurate voltage and current measurements, indicating that the sensor functions reliably in monitoring electrical parameters. This accuracy is very important for ensuring the safe operation of electrical systems during factory preparation.

After that, a series of experimental scenarios were conducted that included multiple events and a variety of contracts at the same time. The results of these experiments indicate the successful implementation of all the scenarios. While some experiments revealed slight delays in response when multiple events occurred, these delays were minimal—usually less than a second—and are considered minimal in the general evaluation of system performance.

These results emphasize the continuous need for continuous monitoring, improved sensor performance, improved environmental considerations, and improved system response to enhance comprehensive reliability and the effectiveness of safety and control systems in industrial environments. These problems can contribute to the calibration of the sensor, durability of the environment, improvement of the response time in the risk detection system, and more flexible and effective notification in industrial environments.

User interface and smartphones by using the Blynk app. The graphical user interface was designed to display the sensor's data. The gauge was used to show real-time data, while the super chart was used to graphically depict the increase and decrease of the sensor reading.

The developed GUI with a realistic result captured on a given scenario of system operation is depicted in Fig. 21 above, which was taken from the Blynk app. The super chart shown in the figure has two axes, x and y. The x-axis shows the time in addition to the time range of the picker, allowing us to choose the required periods (15 m, 30 m, 1 h, 3 h, etc.). The y-axis shows the autoshale of the incoming data. The height data are shown as percentages.

1) Discussion about sensor performance:

- Impact of environmental conditions on sensor performance: Sensors are important components of an IIoT system, providing real-time data on various parameters such as temperature, humidity, gas levels, and movement within the manufacturing environment. Environmental factors such as temperature changes, humidity levels, electromagnetic interference, and exposure to contaminants can affect sensor performance. It is essential to evaluate how sensors respond to different environmental conditions to ensure accurate and reliable data collection for effective monitoring and control of plant operations.

- Factors affecting sensing accuracy: Sensing accuracy refers to the closeness of the measured values to the true values of the parameters being monitored. Several factors can affect sensor accuracy. First, regular calibration of sensors is necessary to maintain accuracy and consistency in measurements. Second, interference from external factors such as electromagnetic interference or cross-sensitivity to other environmental variables can affect the accuracy of the sensor. Third, deviation: Deviation of the sensor over time can lead to deviations in readings, affecting the overall accuracy of the system. Finally, signal-to-noise ratios can reduce the accuracy of high levels of noise in sensor signals, requiring signal processing techniques to reduce noise.

- Factors affecting sensor accuracy: Sensor accuracy relates to the repeatability and consistency of sensor measurements under similar conditions. The key factors that affect sensor accuracy include the following: First, accuracy: The smallest change in the measured quantity that the sensor can detect determines its accuracy. Second, there is stability. The stability of the sensor over time and across different operating conditions is critical to maintaining accuracy. In addition to the sampling rate, the frequency with which sensor data are sampled can affect the accuracy of measurements, especially in dynamic environments.
- Factors affecting sensor reliability: Sensor reliability refers to the ability of sensors to provide consistent and accurate data over long periods without failure. Factors that affect sensor reliability include the strength of sensors to withstand harsh environmental conditions and mechanical stress. Regular maintenance and calibration procedures are essential to ensure sensor reliability. The implementation of redundant sensors or redundant systems can enhance reliability by providing fail-safe mechanisms in the event of sensor failure.
- Operational scenarios and performance evaluation: Conducting sensor performance evaluations under different operational scenarios, such as extreme temperatures, high humidity, and rapid changes in environmental conditions, is critical. Comparative analysis of sensor data collected under different scenarios can help identify weaknesses, potential limitations, and areas for improvement in an IIoT system. The use of statistical methods, data visualization techniques, and trend analysis can help evaluate sensor performance, identify outliers, and optimize sensor configurations to enhance accuracy, precision, and reliability. Factory data can be viewed on a local graph.

2) *Discussion of Critical Factors Affecting the System*

The implementation of the Industrial Internet of Things (IIoT) system presented in the study faced several integration challenges, compatibility issues, and communication failures, which are critical factors affecting system reliability and scalability. These challenges highlight the importance of a robust integration process and the need for effective strategies to address technical limitations to ensure optimal system performance in industrial environments.

During the implementation phase, integrating various sensors to monitor parameters such as air quality, water quality and motion detection poses major challenges. Sensor compatibility, data synchronization, and calibration issues are encountered, underscoring the complexity of integrating multiple sensors into a system architecture. Ensuring seamless integration of sensors is essential for accurate data collection and analysis, thus enhancing system monitoring capabilities.

Compatibility issues have arisen regarding the hardware and software components within the IIoT system. Challenges related to hardware compatibility were identified, including sensors, microcontrollers, and communication modules from different manufacturers. Ensuring compatibility and interoperability between hardware components is critical for preventing failures and inconsistencies in data processing. In addition, software compatibility issues, such as firmware and IoT platform compatibility, were observed, underscoring the importance of aligning software components to facilitate smooth system operation.

Communication failures, including network connectivity issues and data loss, were major challenges encountered during implementation. Network disturbances, signal interference, and bandwidth limitations contributed to communications failures, impacting data transmission and real-time monitoring capabilities. Addressing these connectivity challenges is essential for ensuring continuous data flow and system reliability. Implementing robust network infrastructure, redundancy procedures, and data backup mechanisms can help mitigate communications failures and prevent data loss, thus improving overall system performance.

To address integration challenges, compatibility issues, and communications failures, several strategies can be implemented to enhance system reliability and scalability. These strategies include the following:

- The protocols and interfaces must be standardized to ensure seamless integration and interoperability.
- Redundancy measures, backup systems and failover mechanisms should be implemented to enhance system resilience.
- Comprehensive testing, validation and simulation of system components should be conducted to proactively identify and address technical challenges.
- The scalability of the IIoT system should be considered, and flexible architecture and modular components should be incorporated to accommodate future growth and evolving industrial requirements.

By implementing these strategies and addressing integration challenges, compatibility issues, and communications failures, the reliability and scalability of an IIoT system can be improved, ensuring optimal performance and functionality in industrial environments.

3) *Discussion on Communication Protocols*

The communication infrastructure of the implemented Industrial Internet of Things (IIoT) system uses the ESP NOW protocol to transfer data between the ESP nodes, the central gateway and the Blynk Cloud platform. While the

ESP NOW protocol offers advantages such as low power consumption, fast data transfer, and simplicity in setting up mesh networks, there are limitations and potential weaknesses associated with this protocol that need to be taken into consideration to ensure data integrity and confidentiality in an IIoT system.

- Limitations and weaknesses of the ESP NOW protocol: When multiple ESP nodes transmit data simultaneously or in high-traffic environments, network congestion may occur, resulting in delays in data transmission and possible packet loss. Network congestion can affect the real-time monitoring capabilities of an IIoT system and overall performance. Due to the nature of wireless communication and potential interference, packet loss can occur during data transfer using the ESP NOW protocol. Packet loss can result in incomplete or corrupted data packets, resulting in data inconsistency and impacting the reliability of the monitoring system.

- Optimizing network infrastructure, including channel selection, transmission power settings, and data rate configurations, can help relieve network congestion and reduce the possibility of packet loss. By optimizing the network parameters, the performance of the ESP NOW protocol can be improved, improving the data transmission efficiency. While the ESP NOW protocol offers advantages in terms of efficiency and simplicity for data transfer in IIoT systems, it is necessary to address potential limitations and weaknesses associated with the protocol to ensure data integrity and confidentiality. By implementing encryption, authentication, error detection, network optimization, and secure data handling practices, the reliability and security of communications infrastructure can be enhanced, risks mitigated, and sensitive data protected in IIoT environments.

4) *Discussion Performance Metrics*

Performance metrics play a critical role in evaluating the operational efficiency and effectiveness of an Industrial Internet of Things (IIoT) system in a factory environment. By measuring KPIs such as response times, data transfer rates, and system uptime, stakeholders can evaluate the system's reliability, responsiveness, and overall performance. Table III Details of the performance metrics used to evaluate the operational efficiency and effectiveness of the system.

5) *Discussion Strengths and limitations*

This study delves into the strengths and limitations of an Industrial Internet of Things (IIoT) system designed to monitor and control various aspects of factory operations. Fig. 22 shows the strengths and limitations of the proposed system.

Strengths of the current study:

- Comprehensive integration of sensors: The current study integrates a variety of sensors, such as liquid pH, turbidity, ultrasound, temperature, voltage, and current sensors, enabling a comprehensive monitoring approach.
- Distributed Node System: Implementing seven ESP nodes throughout the plant enables comprehensive monitoring and control at different locations.
- Communication Infrastructure: The use of ESP gateway boards and ESPNow communication protocols ensures efficient data transfer and communication.
- User Interface Development: This study includes local GUI development and integration with the Blynk platform, providing users with intuitive interfaces for monitoring and control.
- Real-world testing: Testing Node 5 indoors and in a factory environment demonstrates the practicality of a water pump control and monitoring system.

TABLE III. KEY PERFORMANCE INDICATORS

Response Times	• Response time: The time it takes for the system to respond to user input or request. It includes processing time, network access time, and any delay in retrieving or transferring data. • User Interface Response Time: The time it takes for a graphical user interface (GUI) or mobile application to respond to user interactions such as button clicks, data entry, or menu selections. • Sensor data processing time: The time required for the system to process incoming sensor data, perform calculations or analyses, and generate insights or alerts in real time
Data Transmission Rates	• Sensor data transfer rate: The rate at which sensor data is collected, transmitted to the cloud or central server, and stored for analysis. It measures the efficiency of data transfer and the system's ability to handle large amounts of sensor data. • Network data transfer rate: The speed at which data is transferred across the network between devices, gateways, and servers. It evaluates the network bandwidth and data transfer capabilities of the system. • Data storage and retrieval rate: The speed at which historical data is stored in databases and retrieved for analysis, reporting, or visualization. It evaluates the storage of data in the system and the efficiency of retrieval
Efficient data processing	• Data processing speed: The rate at which the system processes incoming data, performs analyses, generates insights, and launches automated actions. It evaluates the system's data processing capabilities and efficiency. • Real-time data processing: The ability of the system to process and analyze sensor data in real-time, enabling immediate responses, alerts, or control actions based on incoming data • Batch Data Processing: Efficient batch data processing to handle large amounts of data, perform batch analyses, and generate reports or summaries periodically
Scalability and performance under load	• Scalability metrics: Evaluate how to measure system performance as the number of connected devices, sensors, or users increases. It measures the system's ability to handle growth and increased data traffic. • Load Test Results: Conduct load tests simulating large amounts of data, simultaneous user interactions, or extreme usage scenarios to evaluate system performance under stress. It helps identify performance bottlenecks and optimize system resources.

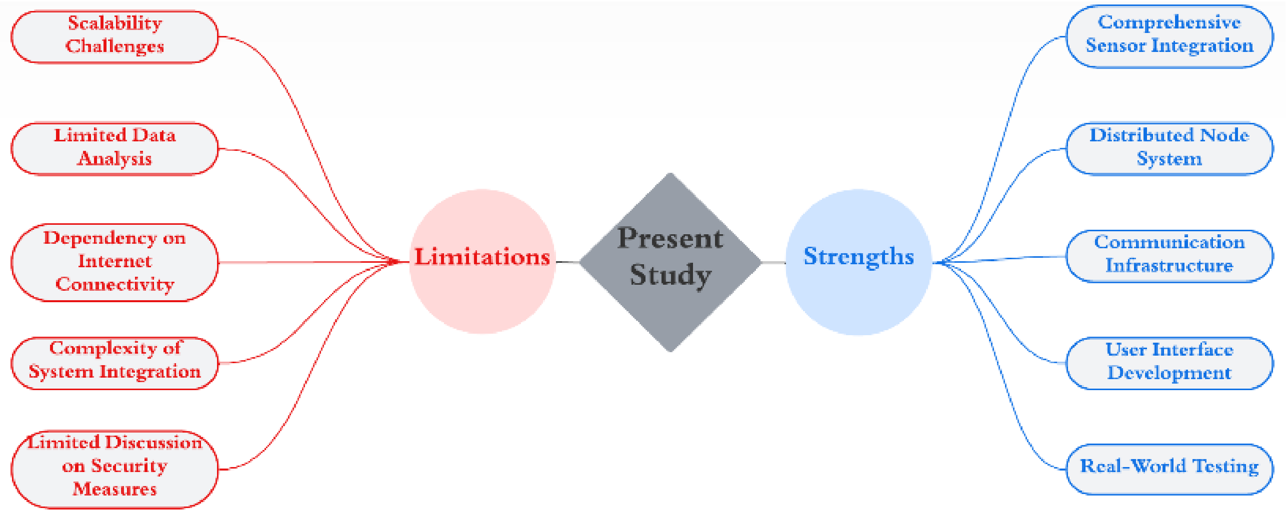


Fig. 22. Strengths and limitations for the proposed study

Limitations of the current study:

- Scalability Challenges: While the study implements seven ESP nodes, scalability to larger industrial environments may pose challenges in terms of data management and system performance.
- Limited data analysis: This study focused on data transfer and visualization but may benefit from more in-depth data analyses for predictive maintenance or process improvement.
- Reliance on an internet connection: Relying on an internet connection to transfer data may lead to vulnerabilities in cases of network outages or outages.
- System Integration Complexity: Integrating multiple sensors, microcontrollers, and communication protocols may increase the complexity of system setup and maintenance.
- Limited discussion on security measures: This study can provide further insights into the security measures implemented to protect the system from cyber threats and unauthorized access.

By addressing these limitations and building on the strengths of this study, future research can further enhance the effectiveness and robustness of an IIoT system for monitoring and controlling factory operations.

IV. CONCLUSION

The ESPNow Protocol-Based IIoT system significantly advances smart factory technologies, offering enhanced monitoring capabilities, improved safety measures, and improved operational efficiency. While specific performance indicators or comparative analyses against existing solutions were not provided, the scalability of the system's design sets a strong foundation for accommodating future expansion and technological advancements. Challenges encountered during integration, such as hardware compatibility issues and communication protocol complexities, underscore the importance of robust integration processes and compatibility testing.

The implementation of data security measures and privacy protection mechanisms within the IIoT system ensures the safeguarding of sensitive information, enhancing overall system reliability. Furthermore, the broader safety implications and occupational health considerations of the system highlight its role in enhancing workplace safety and mitigating operational hazards, particularly through real-time hazard detection and response mechanisms.

The cost–benefit analysis of implementing the IIoT system in a factory setting is essential for evaluating economic feasibility and return on investment. By quantifying potential cost savings and time efficiencies compared to traditional monitoring methods, the system's value proposition for factory owners and operators becomes evident. The versatility of the system in providing both local and global control options enhances its usability and accessibility, contributing to improved operational efficiency.

Future research directions should focus on scalability enhancements, integration with advanced analytics or machine learning algorithms for predictive maintenance, and exploration of interoperability with other smart factory systems. Continued innovation and improvement in smart factory technologies will address emerging challenges and opportunities in industrial automation and monitoring, motivating researchers to further explore the potential of IIoT systems in enhancing industrial operations.

In conclusion, the ESPNow Protocol-Based IIoT system significantly advances the capabilities of smart factories, promoting safer and more efficient industrial operations. The system's ability to detect and respond to potential hazards, coupled with its cost-effective monitoring solutions and usability enhancements, underscores its value in revolutionizing industrial automation. This research contributes to new knowledge in the domain of smart factories and motivates further exploration and innovation in IIoT technologies to drive continuous improvement in industrial monitoring and control systems.

REFERENCES

[1] L. Hu, Y. Miao, G. Wu, M. M. Hassan, and I. Humar, “iRobot-Factory: An intelligent robot factory based on cognitive manufacturing and edge computing,” *Future Generation Computer Systems*, vol. 90, 2019, doi: 10.1016/j.future.2018.08.006.

[2] S. Munirathinam, “Industry 4.0: Industrial Internet of Things (IIOT),” *Advances in Computers*, vol. 117, no. 1, pp. 129–164, Jan. 2020.

[3] R. Hajlaoui, T. Moulahi, S. Zidi, S. El Khediri, B. Alaya, and S. Zeadally, “Towards smarter cyberthreats detection model for industrial internet of things (IIoT) 4.0,” *J Ind Inf Integr*, vol. 39, p. 100595, May 2024, doi: 10.1016/J.JII.2024.100595.

[4] J. Vijayalakshmi, G. Puthilibhai, and S. R. L. Siddarth, “Implementation of Ammonia Gas Leakage Detection Monitoring System using Internet of Things,” in *Proceedings of the 3rd International Conference on I-SMAC IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2019*, pp. 778-781, 2019.

[5] M. Sangeetha, S. Arulselvi, S. Saravana, and G. Kanagavalli, “IOT based industrial automation,” *International Journal of Recent Technology and Engineering*, vol. 8, no. 1, 2019.

[6] S. Pasika and S. T. Gandla, “Smart water quality monitoring system with cost-effective using IoT,” *Heliyon*, vol. 6, no. 7, 2020.

[7] S. A. H. Almetwally, M. K. Hassan, and M. H. Mourad, “Real Time Internet of Things (IoT) Based Water Quality Management System,” in *Procedia CIRP*, vol. 91, pp. 478-485, 2020, doi: 10.1016/j.procir.2020.03.107.

[8] E. Raghuveera, P. Kanakaraja, K. H. Kishore, C. Tanvi Sriya, B. Durga Prasad, and B. Sai Krishna Teja Lalith, “An IoT Enabled Air Quality Monitoring System Using LoRa and LPWAN,” in *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, pp. 453-459, 2021.

[9] W. Rusydi *et al.*, “Electrical Machine Breakdown Monitoring System Using IoT,” *Journal of Advanced Industrial Technology and Application*, vol. 3, no. 2, pp. 1–7, Dec. 2022.

[10] J. Linggarjati, “Design and Prototyping of Temperature Monitoring System for Hydraulic Cylinder in Heavy Equipment using ESP32 with data logging and WiFi Connectivity,” in *IOP Conference Series: Earth and Environmental Science*, vol. 998, no. 1, p. 012042, 2022.

[11] M. Babiuch, P. Foltynek, and P. Smutny, “Using the ESP32 microcontroller for data processing,” *Proceedings of the 2019 20th International Carpathian Control Conference, ICCC 2019*, pp. 1-6, May 2019, doi: 10.1109/CARPATHIANCC.2019.8765944.

[12] R. R and S. Naveen, “Customer Following Trolley using ESP32,” *2022 International Conference on Futuristic Technologies (INCOFT)*, pp. 1–6, Nov. 2022, doi: 10.1109/INCOFT55651.2022.10094431.

[13] Y. Magzym, A. Eduard, D. Urazayev, X. Fafoutis, and D. Zorbas, “Synchronized ESP-NOW for Improved Energy Efficiency,” in *2023 IEEE International Black Sea Conference on Communications and Networking, BlackSeaCom 2023*, pp. 57-62, 2023.

[14] Espressif Iot Team, “ESP-Now Resources | Espressif Systems,” *Espressif*, Accessed: Jul. 27, 2022.

[15] M. Markovic, M. Maljkovic, and R. N. Hasanah, “Smart Home Heating Control using Raspberry Pi and Blynk IoT Platform,” *EECCIS 2020 - 2020 10th Electrical Power, Electronics, Communications, Controls, and Informatics Seminar*, pp. 188–192, Aug. 2020, doi: 10.1109/EECCIS49483.2020.9263441.

[16] P. Rajak, A. Ganguly, S. Adhikary, and S. Bhattacharya, “Internet of Things and smart sensors in agriculture: Scopes and challenges,” *J Agric Food Res*, vol. 14, p. 100776, Dec. 2023.

[17] A. A. Kuzmenkov, D. A. Kuvshinov, S. Y. Buryachenko, A. V. Kaychenov, I. M. Karachentseva, and Z. A. Voronin, “Monitoring system for temperature and relative humidity of the experimental building,” in *Journal of Physics: Conference Series*, vol. 2131, no. 5, p. 052070, 2021.

[18] M. Frei, C. Deb, R. Stadler, Z. Nagy, and A. Schlueter, “Wireless sensor network for estimating building performance,” *Autom Constr*, vol. 111, p. 103043, Mar. 2020.

[19] B. Lawson *et al.*, “Skin alcohol perspiration measurements using MOX sensors,” *Sens Actuators B Chem*, vol. 280, pp. 306–312, Feb. 2019, doi: 10.1016/J.SNB.2018.09.082.

[20] N. Agarwal and Y. Rohilla, “Flame Sensor Based Autonomous Firefighting Robot,” *Lecture Notes in Electrical Engineering*, vol. 748, pp. 641–655, 2021.

[21] G. Qiang, S. Tang, J. Hao, L. Di Sarno, G. Wu, and S. Ren, “Building automation systems for energy and comfort management in green buildings: A critical review and future directions,” *Renewable and Sustainable Energy Reviews*, vol. 179, p. 113301, Jun. 2023.

[22] Z. Abidin, E. Maulana, M. Y. Nurrohman, F. C. Wardana, and Warsito, “Real Time Monitoring System of Drinking Water Quality Using Internet of Things,” *IES 2022 - 2022 International Electronics Symposium: Energy Development for Climate Change Solution and Clean Energy Transition, Proceeding*, pp. 131–135, 2022.

[23] K. M. Simitha and M. S. Subodh Raj, “IoT and WSN Based Water Quality Monitoring System,” *Proceedings of the 3rd International Conference on Electronics and Communication and Aerospace Technology, ICECA 2019*, pp. 205–210, Jun. 2019.

[24] M. A. Mutri, A. R. A. Saputra, I. Alinursafa, A. N. Ahmed, A. Yafouz, and A. El-Shafie, “Smart system for water quality monitoring utilizing long-range-based Internet of Things,” *Appl Water Sci*, vol. 14, no. 4, 2024, doi: 10.1007/s13201-024-02128-z.

[25] A. Haque, N. Shah, J. A. Malik, and A. Malik, “Fundamentals of power electronics in smart cities,” *Smart Cities: Power Electronics, Renewable Energy, and Internet of Things*, pp. 1–24, Jan. 2024.

[26] G. R. Kanagachidambaresan, “Introduction to Wired and Wireless IoT Protocols in SBC,” in *Internet of Things*, pp. 99-120, 2021.

[27] Y. K. Taru and A. Karwankar, “Water monitoring system using arduino with labview,” in *Proceedings of the International Conference on Computing Methodologies and Communication, ICCMC 2017*, pp. 416-419, 2018, doi: 10.1109/ICCMC.2017.8282722.

[28] O. E. Amestica, P. E. Melin, C. R. Duran-Faundez, and G. R. Lagos, “An Experimental Comparison of Arduino IDE Compatible Platforms for Digital Control and Data Acquisition Applications,” *IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies, CHILECON 2019*, pp. 1-6, Nov. 2019, doi: 10.1109/CHILECON47746.2019.8986865.

[29] J. Hao and T. K. Ho, “Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language,” *Journal of Educational and Behavioral Statistics*, vol. 44, no. 3, pp. 348–361, Jun. 2019, doi: 10.3102/1076998619832248.

[30] J. Willman, “Overview of PyQt5,” *Modern PyQt*, pp. 1–42, 2021.

[31] J. Lehmann, S. Schorz, A. Rache, T. Häußermann, M. Rädle, and J. Reichwald, “Establishing Reliable Research Data Management by Integrating Measurement Devices Utilizing Intelligent Digital Twins,” *Sensors*, vol. 23, no. 1, 2023, doi: 10.3390/s23010468.

[32] M. A. Hailan, B. M. Albaker, and M. S. Alwan, “Two-Dimensional Transformation of a Conventional Manufacturer into a Smart Manufacturer: Architectonic Design, Maintenance Strategies and Applications,” *Al-Iraqia Journal of Scientific Engineering Research*, vol. 1, no. 1, Sep. 2022, doi: 10.33193/ijser.1.1.2022.39.

[33] P. Daponte, F. Lamonaca, F. Picariello, L. De Vito, G. Mazzilli, and I. Tudosa, “A Survey of Measurement Applications Based on IoT,” *2018 Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2018 - Proceedings*, pp. 157–162, Aug. 2018.

[34] M. Alabadi, A. Habbal, and X. Wei, “Industrial Internet of Things: Requirements, Architecture, Challenges, and Future Research Directions,” *IEEE Access*, vol. 10, pp. 66374-66400, Jun. 2022.

[35] A. Malkawi *et al.*, “Design and applications of an IoT architecture for data-driven smart building operations and experimentation,” *Energy Build*, vol. 295, 2023, doi: 10.1016/j.enbuild.2023.113291.

[36] P. P. Ray, “A survey on Internet of Things architectures,” *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 3, pp. 291–319, Jul. 2018, doi: 10.1016/J.JKSUCI.2016.10.003.

[37] M. Burhan, R. A. Rehman, B. Khan, and B. S. Kim, “IoT elements, layered architectures and security issues: A comprehensive survey,” *Sensors (Switzerland)*, vol. 18, no. 9, 2018, doi: 10.3390/s18092796.

[38] A. Gupta, “UART Communication,” *The IoT Hacker’s Handbook*, pp. 59–80, 2019, doi: 10.1007/978-1-4842-4300-8_4.

[39] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, “A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures,” *IEEE Access*, vol. 7, pp. 82721–82743, 2019.

[40] A. P. Plageras, K. E. Psannis, C. Stergiou, H. Wang, and B. B. Gupta, “Efficient IoT-based sensor BIG Data collection–processing and analysis in smart buildings,” *Future Generation Computer Systems*, vol. 82, pp. 349–357, May 2018.

[41] J. Q. Li, F. R. Yu, G. Deng, C. Luo, Z. Ming, and Q. Yan, “Industrial Internet: A Survey on the Enabling Technologies, Applications, and Challenges,” *IEEE Communications Surveys and Tutorials*, vol. 19, no. 3, pp. 1504–1526, Jul. 2017.

[42] D. Kanellopoulos, V. K. Sharma, T. Panagiotakopoulos, and A. Kameas, “Networking Architectures and Protocols for IoT Applications in Smart Cities: Recent Developments and Perspectives,” *Electronics (Switzerland)*, vol. 12, no. 11. 2023.

[43] J. A. Cujilema Paguay, G. A. Hidalgo Brito, D. L. Hernandez Rojas, and J. J. Cartuche Calva, “Secure home automation system based on ESP-NOW mesh network, MQTT and Home Assistant platform,” *IEEE Latin America Transactions*, vol. 21, no. 7, pp. 829–838, Jul. 2023, doi: 10.1109/TLA.2023.10244182.

[44] Espressif Systems, “ESP-NOW User Guide,” Espressif Systems.

[45] Y. Magzym, A. Eduard, D. Urazayev, X. Fafoutis, and D. Zorbas, “Synchronized ESP-NOW for Improved Energy Efficiency,” *2023 IEEE International Black Sea Conference on Communications and Networking, BlackSeaCom 2023*, pp. 57–62, 2023.

[46] S. Encheva and S. Tumin, “Dynamic Secure Mesh for Collaborative Nodes of IoT Devices,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 19–30, 2023.

[47] D. Pujara, P. Naik, R. Gautam, and A. Mecwan, “ESP32 ChatLink: Real-Time Peer-to-Peer Communication via ESP-NOW,” pp. 469–478, 2024, doi: 10.1007/978-981-97-1329-5_38.

[48] G. Amponis *et al.*, “Efficient Peer-to-Peer Unicasting for VANET Architectures via Enhanced Monolithic Protocols,” in *7th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference, SEEDA-CECNSM 2022*, pp. 1-8, 2022, doi: 10.1109/SEEDA-CECNSM57760.2022.9932897.

[49] Sensirion, “Datasheet SHT3x-DIS,” *Sensirion*, 2015.

[50] Rogerbit, “Flame Sensor Module,” *Rogerbit*, 2018.

[51] A. Day, “ACROBOTIC | MiCS-5524 Datasheet.” [Online]. Available: www.sgxsensortech.com

[52] A. A. Jaber, F. K. I. Al-Mousawi, and H. S. Jasem, “Internet of things based industrial environment monitoring and control: A design approach,” *International Journal of Electrical and Computer Engineering*, vol. 9, no. 6, 2019, doi: 10.11591/ijece.v9i6.pp4657-4667.

[53] B. Idoko, J. B. Idoko, Y. Z. Mahmud Kazaure, Y. M. Ibrahim, F. A. Akinsola, and A. R. Raji, “IoT Based Motion Detector Using Raspberry Pi Gadgetry,” *Proceedings of the 5th International Conference on Information Technology for Education and Development: Changing the Narratives Through Building a Secure Society with Disruptive Technologies, ITED 2022*, pp. 1-5, 2022, doi: 10.1109/ITED56637.2022.10051334.

[54] P. V. Vimal and K. S. Shivaprakasha, “IOT based greenhouse environment monitoring and controlling system using Arduino platform,” *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies, ICICICT 2017*, vol. 2018-January, pp. 1514–1519, Apr. 2018, doi: 10.1109/ICICICT1.2017.8342795.

[55] M. Sabari, P. Aswinth, T. Karthik, and C. Bharath Kumar, “Water Quality Monitoring System Based on IoT,” in *ICDCS 2020 - 2020 5th International Conference on Devices, Circuits and Systems*, pp. 279-282, 2020, doi: 10.1109/ICDCS48716.2020.243598.

[56] S. Indriyanto and I. Y. M. Edward, “Ultrasonic Underwater Acoustic Modem Using Frequency Shift Keying (FSK) Modulation,” *Proceeding of 2018 4th International Conference on Wireless and Telematics, ICWT 2018*, pp. 1-4, Nov. 2018, doi: 10.1109/ICWT.2018.8527809.

[57] A. F. Robbi, Z. E. Bhagaskara, and R. S. A. Kuswandrata, “Integrated Wireless Sensor System for Monitoring the Reactor Water Temperature to Support Internet Reactor Laboratory Development on Calibrating Reactor Power,” *AIP Conf Proc*, vol. 2967, no. 1, Feb. 2024, doi: 10.1063/5.0192899/3265610.

[58] A. Abu Sneineh and A. A. A. Shabaneh, “Design of a smart hydroponics monitoring system using an ESP32 microcontroller and the Internet of Things,” *MethodsX*, vol. 11, p. 102401, Dec. 2023, doi: 10.1016/J.MEX.2023.102401.

[59] S. Ginting, J. W. Simatupang, I. Bukhori, and E. R. Kaburuan, “Monitoring of Electrical Output Power-Based Internet of Things for Micro-Hydro Power Plant,” *2018 International Conference on Orange Technologies, ICOT 2018*, pp. 1-7, Jul. 2018, doi: 10.1109/ICOT.2018.8705786.

[60] Syafii, A. Luthfi, and A. Y. A. Rozzi, “Design of raspberry pi web-based energy monitoring system for residential electricity consumption,” *2020 International Conference on Information Technology Systems and Innovation, ICITSI 2020 - Proceedings*, pp. 192–196, Oct. 2020, doi: 10.1109/ICITSI50517.2020.9264926.

[61] N. C. A. Boovarahan *et al.*, “IOT-Based Water Level Management System,” *Lecture Notes in Networks and Systems*, vol. 894, pp. 357–364, 2024, doi: 10.1007/978-981-99-9562-2_29.

[62] K. Rambabu, S. Dubey, K. N. Srinivas, D. N. Reddy, and C. Rohith, “Smart Water Flow and Pipeline Leakage Detection Using IoT and Arduino UNO,” *2nd International Conference on Intelligent Data Communication Technologies and Internet of Things, IDCIoT 2024*, pp. 174–180, 2024, doi: 10.1109/IDCIOT59759.2024.10467990.

[63] B. K, D. R, B. B. Sinha, and G. R, “Clock synchronization in industrial Internet of Things and potential works in precision time protocol: Review, challenges and future directions,” *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 205–219, Jun. 2023, doi: 10.1016/J.IJCCE.2023.06.001.

[64] Z. Xu *et al.*, “A review: Insight into smart and sustainable ultra-precision machining augmented by intelligent IoT,” *J Manuf Syst*, vol. 74, pp. 233–251, Jun. 2024, doi: 10.1016/J.JMSY.2024.03.008.

[65] I. Ullah, D. Adhikari, X. Su, F. Palmieri, C. Wu, and C. Choi, “Integration of data science with the intelligent IoT (IIoT): current challenges and future perspectives,” *Digital Communications and Networks*, Mar. 2024, doi: 10.1016/J.DCAN.2024.02.007.

[66] A. Helmenstine, “The pH Scale of Common Chemicals,” *Science Notes*, 2023, https://sciencenotes.org/the-ph-scale-of-common-chemicals/.