Babylonian Journal of Internet of Things
Vol.2024, **pp**. 16–26
DOI: https://doi.org/10.58496/BJIoT/2024/003; ISSN: 3006-1083
https://mesopotamian.press/journals/index.php/BJIoT

Research Article

Impact of IoT-based environmental monitoring on lab safety and sustainability

Nadia Mahmoud Hussein [1,*], Yasmin Makki Mohialden [1], Saba Abdulbaqi Salman [2]

[1] *Department of Computer Science, College of Science, Mustansiriyah University, Baghdad, Iraq.*
[2] *Department of Computer,College of Education, Al-Iraqia University, Iraq Baghdad.*

ARTICLE INFO

Article History
Received 25 Dec 2023
Revised 01 Feb 2024
Accepted 09 Feb 2024
Published 01 Mar 2024

Keywords
laboratory safety
IoT monitoring
gas detection
fire detection
pH measurement
remote monitoring
critical event reporting

ABSTRACT

Several internal and external dangers might threaten a laboratory, the institution, and the public if not addressed properly. These hazards are addressed by our extensive laboratory monitoring system. This device detects elevated temperatures, gas leaks, fires, and solution pH. Elements of this system include an Esp32 microcontroller, LCD, Arduino Uno, mq2 gas sensor, DS18B20 temperature sensor, flame sensor, relay, water pump, and pH sensor for acidity measurement. The Arduino IDE controls sensors. Once the temperature reaches 50 degrees Celsius, an alarm, red LED, and mobile phone notification activate. Gas leaks activate a green LED, an alarm, and a message. Flame sensors notify and start water pumps when fires occur. Gas leaks and fires activate a buzzer. The IoT offers essential incident reporting and remote laboratory monitoring. The code is written in Arduino C++.

1. INTRODUCTION

Chemical solutions and electrical equipment in chemistry labs can cause catastrophic catastrophes that cost lives and money [1]. Manual laboratory monitoring is difficult, requiring global remote monitoring. IoT allows real-time laboratory monitoring over the Internet [2-4]. Included in the system are an automatic water sprinkler system, fire sensors, and gas leak and high-temperature detectors. This system improves laboratory safety by monitoring and responding quickly to dangers and protecting people and assets. Lab chemicals can produce irritation, sensitization, carcinogenicity, flammability, corrosion, and explosibility. To ensure workplace chemical safety, OSHA's Hazard Communication Standard (HCS) requires employees to understand chemical identification and dangers. OSHA provides labelling, safety data sheets, hazard assessment, and chemical handling training [5]. From 0 to 14, the pH scale measures solution acidity or alkalinity. The neutral pH is 7, the acidic pH is below 7, and the basic pH is above 7. Water pH indicates H+ and OH- ions. Alkaline water has OH, whereas acidic water contains H+. Each logarithmic pH unit changes acidity or alkalinity tenfold. For instance, pH 5 water is ten times more acidic than pH 6 [6]. Real-time systems must react to external events like physical time within defined periods. Systems with considerable output production [7]. Three kinds of systems exist: Hard Real-Time Systems: These systems are extremely time-sensitive, and any breach of time constraints is unacceptable, even if the results are correct. They are used in mission-critical applications like aircraft control and robotics. Real-Time Systems with Latitude (Soft Real-Time Systems): These systems impose time constraints on activities but can tolerate minor delays to prevent system failure or significant losses. Examples include live video broadcasting and e-commerce. Firm Real-Time Systems: These systems also require adherence to deadlines, with missed deadlines potentially impacting product quality. The system continues to operate but discards late responses [8]. Embedded systems integrate hardware and software in a single environment to control devices. Examples include air conditioners, cell phones, and traffic lights [9]. Real-time embedded systems monitor, respond to, and interface with sensors, actuators, and other I/O interfaces connecting the computer system to the environment. These systems adhere to timing constraints imposed by the real-time behavior of the external world and may be part of larger embedded systems [10].

*Corresponding author. Email: nadia.cs89@uomustansiriyah.edu.iq

The Internet of Things (IoT) connects devices and powers local and cloud-based communication. Billion gadgets are online because to cheap CPUs and high-speed connections. Smart toothbrushes, vacuums, cars, and machines may use sensors to collect data and respond to humans. This system uses IoT[11-13].

Problem Statement Changes in chemical storage conditions can lead to disasters, emphasizing the need for intelligent laboratory monitoring.

The system aims to:

- Monitor and display storage area temperature on an LCD screen.
- Ensure laboratory safety.
- provide immediate alerts during emergencies.
- Minimize energy consumption.
- Communicate system information to administrators using IoT.
- Measure the solution pH.
- automatically suppress fires when detected.

Paper Outline Section 2: Related Work: Discuss prior research and technologies related to laboratory safety and IoT-based monitoring.Section 3: Proposed Method: Details the system's architecture, hardware components, and sensor usage for monitoring and safety. Section 4: Results and Discussion: Presents system performance results and discusses findings in the context of laboratory safety. Section 5 (Conclusion and Future Work) concludes the paper by summarizing key contributions and discussing potential future developments and enhancements.

2. RELATED WORK

Many programs were designed and built around Arduino. Some of these works: In 2020, they used ultrasonic sensors to send out pulses of radio waves or microwaves that bounced off any object in their path. Arduino is a single-board microcontroller that can support multiple accessories quickly and cheaply. Their paper's main goal was to find movable and immovable obstacles in a specific range using processing IDE software and a camera to get an image of the detected obstacles. This System was controlled by a Bluetooth-controlled car [14-18]. They will develop and implement an indoor temperature control system in 2021. In summer, particularly on bright days, drivers feel hot and uncomfortable in their automobiles. Especially if the car is parked in a sunny lot. To maintain comfort while using phase change material, a lot of cooling energy must be utilized to lower the temperature. This smart system controls vehicle temperature in sun-parked cars. Pouches inside the car's top absorb and release heat as phase-change materials melt and stiffen [19-22]. In 2022, they said that a system should be made to actively control the CO2 and O2 levels inside a container for storing fruit, even when the temperature stays the same or changes. A small air blower was used to move gases from the container to the outside and O2 levels inside a container for storing fruit, even when the temperature stays the same or changes. A small air blower moved gases from the container to the outside. A thin, long tube kept air from getting into the container, but when the blower was turned on, it made it easier for air to move around. The blower ON frequency (s h1) was modeled as a function of storage temperature, considering the type and amount of fruit, blower properties, tube size, and the setpoint of O2 concentration. The model was then used to program an Arduino microcontroller to control the blower based on real-time measurements of the storage temperature. The gas control system was then tested by putting sweet cherries in a container. The System could keep the CO2 level at the setpoint level (12.5%), even when the temperature changed from 17 °C to 9 °C or stayed at six °C or 17 °C. The number of times the blower turned on varied from 32 s/h at 6 °C to 350 s/h at 17 °C[22-24]. 2023,The authors monitor and control environmental parameters such as temperature and humidity to ensure that they remain within acceptable ranges for plant growth. If these parameters go beyond specified limits, the system takes action. Instead of relying solely on Wi-Fi for data transmission, it uses an Ethernet connection to ensure data is sent reliably. Additionally, the IFTTT integration enables the system to send alert emails to the administrator, providing a proactive warning mechanism for any unfavorable conditions that may affect the plants. Overall, the authors have designed and implemented a system that helps optimize plant growth by continuously monitoring environmental conditions and providing real-time alerts to address any issues that may arise[25].

3. PROPOSED METHOD

Figure 1 illustrates the block diagram that depicts the system's operational mechanism. The system comprises various interconnected devices, with some serving as input sources for data and others as outputs. The DS18B20 temperature sensor senses and displays temperature values on the LCD. When the temperature exceeds the allowable limit, an alarm message is sent to inform the administrator. The MQ2 gas sensor detects gas, and the gas status is displayed on the LCD. The pH sensor measures the solution's acidity and displays the value on the LCD. The flame sensor is used to detect fires, and in the case of fire detection, the water pump operates automatically. A buzzer has also been integrated into the system, which is triggered in the event of gas leakage or fire. Furthermore, IoT technology is applied by sending information about the lab via the ESP module to the Telegram bot when the user (admin) sends commands.

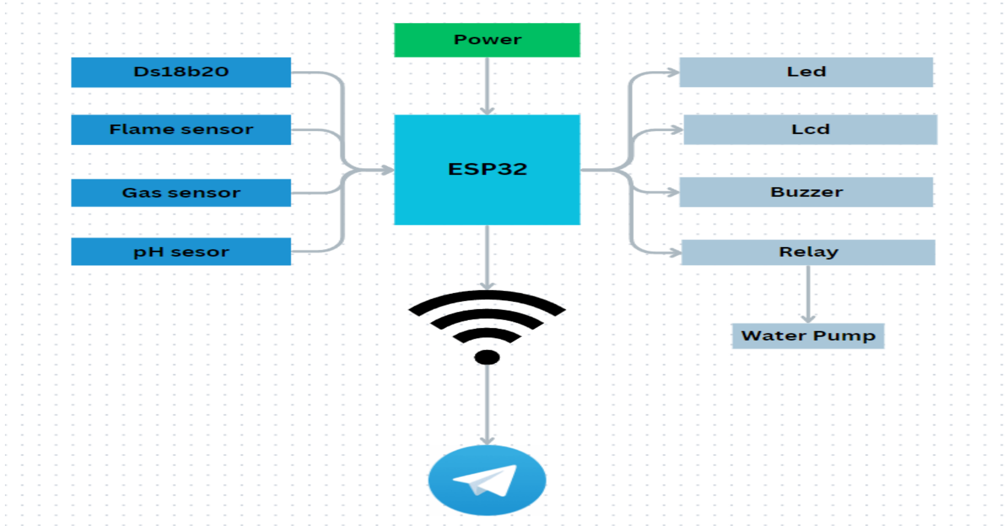


Fig.1. Block Diagram of the System

3.1 Use Case Modelling

A use case diagram is a dynamic or behavioral diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases represent a set of actions, services, and functions that the system needs to perform. Figure 2 illustrates the use-case modelling.

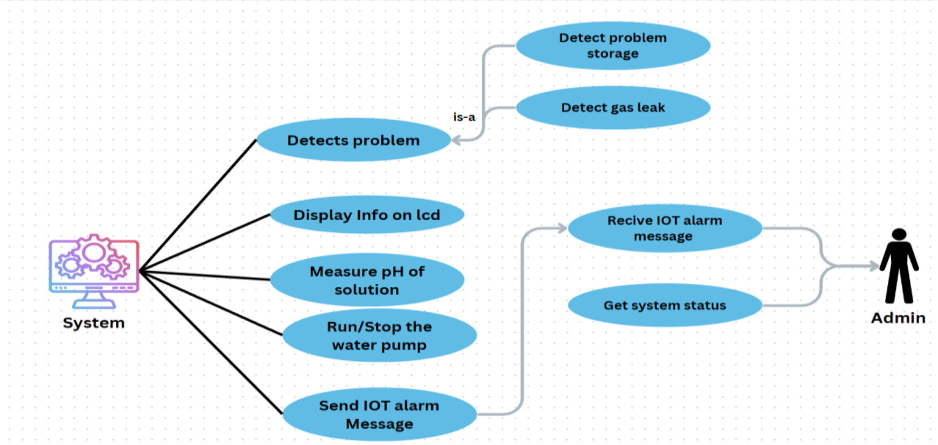


Fig.2. Use Case Modeling.

3.2 Hardware Requirements

The main pieces of hardware that comprise the proposed system are as follows:

1. Arduino microcontroller
   Arduino is an open-source electronics platform. A microcontroller and computer-based integrated development environment (IDE) software make up the system. Computer code is written and uploaded to the board using the IDE[26].
2. DS18B20 sensor

The DS18B20 digital thermometer provides temperature measurements ranging from 9-bit to 12-bit Celsius and includes an alarm function with nonvolatile, user-programmable upper and lower trigger points. Communication with the DS18B20 is achieved over a 1-wire bus, requiring only one data line (and ground) for communication with a central microprocessor. Furthermore, the DS18B20 can draw power directly from the data line, eliminating the need for an external power supply. Each DS18B20 possesses a unique 64-bit serial code, allowing multiple DS18B20s to function on the same 1-wire bus. This feature simplifies the control of multiple DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems within buildings, equipment, or machinery, and process monitoring and control systems [20].Key features of DS18B20:
Temperature range: -50~+125°C.
quick and accurate temperature sensing.
- Customizable.
impact-resistant, waterproof, and durable for a long service life[10].

3. Ph Sensor
   A pH meter is an instrument used to measure the acidity or alkalinity of a solution [11].
4. Flame Sensor
   A flame sensor is primarily used to detect and respond to the occurrence of a flame or fire. Its detection range can extend up to 100 cm [12].
5. Gas Sensor (MQ-2)
   The MQ-2 gas sensor exhibits high sensitivity to propane and smoke and can detect natural gas and other flammable gases [13].
6. LCD I2C (Liquid Crystal Display)
   We recommend the Grove 16 x 2 LCD for Arduino and Raspberry Pi I2C projects. Easy deployment and high contrast. "16x2" specifies two lines with 16 columns and 32 characters each. With two signals and two power connections, the Grove I2C connector simplifies cabling. VCC, GND, SDA, and SCL are I2C module outputs. Arduino GND grounds the I2C module, which receives 5V from VCC. Data and clock pins are SDA and SCL on the I2C module14].
7. ESP32 Microcontroller
   The powerful System-on-Chip (SoC) microcontroller ESP32 has integrated Wi-Fi (802.11 b/g/n), dual-mode Bluetooth 4.2, and peripherals. Two 240 MHz cores make it a sophisticated successor to the 8266 processor. The ESP32 has 4MB of flash memory, 36 GPIO pins, and 16 PWM channels[17].
8. Water Pump
   Irrigation pumps transfer water via canals to irrigated areas. They raise water pressure for field spraying via pipe systems[18].
9. Relay Module
   Relay modules manage current flow by turning on and off electrical switches. These modules may be managed with 3.3V (ESP32 and ESP8266) or 5V as needed [19].

3.3 system Circuit

The suggested System's system circuit will be introduced in this part. Figure (3) depicts the system design circuit with Portus Software.

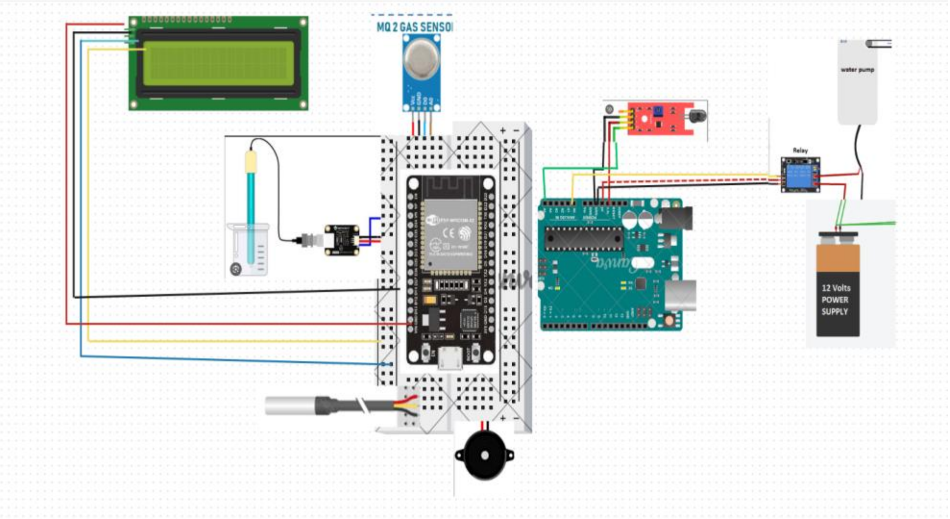


Fig.3. system design circuit.

3.4 Software Requirements

The Arduino IDE is the main software for programming Arduino boards. The components you mentioned are broken down:

1. Text Editor: The Arduino IDE has a built-in editor for coding. Its streamlined C++ programming language makes it easy for novices and experts.
2. The Message Area displays error messages, warnings, and feedback linked to your code. It's very useful when compiling (verifying) code to detect syntax mistakes or exporting to Arduino.
3. Console Text Output: Displays Arduino-generated text. It helps with debugging and transferring Arduino data to a computer.
4. Console Toolbar: The Arduino IDE's top toolbar offers buttons and choices for code and board interaction. Some typical buttons:
   a. Verify: This button checks your code for errors without uploading it to the board.
   b. Upload: Clicking this button compiles your code and uploads it to the connected Arduino board.
   c. New: Creates a new sketch (Arduino program).
   d. Open: Opens an existing Arduino sketch.
   e. Save: Saves the currently open sketch.
   f. Serial Monitor: This opens a window to monitor the serial communication between your Arduino board and the computer.

The bottom right-hand corner of the window typically displays information about the development board selected and the serial port in use. This information is important because it ensures that you're uploading code to the correct board and communicating with the right serial port[20].

3.5 Telegram bot

Telegram bots engage with people via Messenger. They transmit and receive messages, follow orders, and automate processes. Bots can provide information, reminders, and hardware control. Arduino Integration, the open-source electronics platform, makes hardware prototypes and projects easy and versatile. Connecting Telegram Bots to Arduino lets you control and communicate with your devices using Telegram Messenger. Applications, The integration offers several options. For instance: Remote Control: Send Arduino commands over Telegram. Turn on lights, modify the temperature, or operate a robot remotely. Monitor Arduino hardware to send Telegram bot data. Sensor data, status updates, and conditional alarms are examples. Arduino projects may notify, remind, or alert you. Your home security system can alert you when your plants need watering or when motion is detected. As seen in Figures 7 and 8, a user uses a Telegram bot on their smartphone to connect with an Arduino-based hardware prototype. It may show a Telegram user delivering commands to the Arduino.
These integrations make it easier for users to develop, operate, and monitor Arduino projects using Telegram, a popular chat app. Especially handy for remote or home automation applications.

3.6 Algorithm: System Implementation

This algorithm outlines the steps for implementing the described system, from setting up hardware components to testing, deployment, and ongoing maintenance. It also includes optional features like a user interface and considerations for security and future scalability.

1. Initialize hardware components:
   ESP32, Arduino Uno, LCD, MQ2 gas sensor, DS18B20 temperature sensor, flame sensor, relay, water pump, pH sensor
2. Initialize software components:
   a. Arduino IDE for programming.
   b. Telegram bot for alarm notifications
3. Configure IoT connectivity:
   a. Set up ESP32 for internet connectivity (e.g., Wi-Fi).
   b. Establish a connection to external services.
4. Implement Sensor Monitoring and Control Loop:
   While True:
   a. Read temperature data from the DS18B20 temperature sensor.
   b. Read gas data from the MQ2 gas sensor.
   c. Read flame data from the flame sensor.
   d. Read pH data from the pH sensor (if pH measurement is required).
5. Temperature Monitoring:
   If temperature > 50 degrees Celsius:
   a. Trigger an alarm.
   b. Turn on the red LED.
   c. Send a notification via Telegram to the admin's mobile phone.
6. Gas Leak Monitoring:
   If the gas level exceeds a threshold,
   a. Turn on the green LED.
   b. Trigger an alarm.
   c. Send a report or notification via Telegram to the admin.
7. Fire Detection:
   If a flame is detected by the flame sensor:
   a. Activate a notice/alert.
   b. Automatically turn on the water pump using the relay to extinguish the fire.
8. pH monitoring (if required):
   Implement logic for pH monitoring and actions based on pH levels.
9. Testing and Debugging:
   a. Continuously monitor system behavior.
   b. Check for errors, bugs, or hardware issues.
   c. Address any problems encountered during testing.
10. Deployment:
    Install the system in the target environment (home, industrial facility, etc.).
11. Continuous Monitoring and Maintenance:
    a. Continuously monitor system performance and sensor readings.
    b. Perform regular maintenance and updates to ensure reliability.
12. User Interface (optional):
    Develop a user interface for user interaction and data visualization.
13. Security Considerations:
    Implement security measures to protect the system, especially for IoT communication.
14. Scalability and Future Enhancements:
    Plan for future scalability and enhancements, such as adding more sensors or features.
15. Documentation:
    Create comprehensive documentation, documentation, including circuit diagrams, code explanations, and user instructions.
16. End of Algorithm

3.7 Final system components (external and internal view)

the final system components might look like from both an external and internal perspective

1. External View (Front Panel):
    a. LCD Display: The LCD is visible on the front panel, providing real-time information, system status, and alerts to users.
    b. Red LED Indicator: An LED indicator, likely red, is placed prominently to indicate an alarm triggered by high temperatures (above 50 degrees Celsius).
    c. Green LED Indicator: Another LED indicator, likely green, is present to indicate gas leak alerts.
    d. Water Pump Outlet: An outlet for the water pump might be visible, and connected to a hose or pipe for fire suppression.
    e. pH Sensor (if visible): If the pH sensor is part of the external view, it would be mounted on the front panel.
    f. Bot Telegram Integration: While not visible, the system connects to the Telegram platform for sending notifications to the admin's mobile phone.
2. Internal View (Inside the Enclosure):
    a. ESP32 and Arduino Uno: These microcontrollers are the brains of the system, hidden inside the enclosure.
    b. MQ2 Gas Sensor: The MQ2 gas sensor is connected to the microcontrollers and is responsible for detecting gas levels.
    c. DS18B20 Temperature Sensor: This temperature sensor is wired to the microcontrollers to monitor temperature conditions.
    d. Flame Sensor: The flame sensor is installed inside to detect fires.
    e. Relay: The relay is used to control the water pump for fire suppression.
    f. pH Sensor (if used): If applicable, the pH sensor is installed internally for acidity measurement.
    g. Wiring and Circuitry: Internal wiring connects all the components, allowing data exchange and control.
    h. IoT Connectivity Module: The ESP32 is equipped with IoT connectivity hardware, facilitating communication with external services.
    i. Power Supply: A power supply unit provides electricity to the system.
    j. Microcontroller Programming: The Arduino IDE is used for programming the microcontrollers, with the code responsible for sensor data processing, decision-making, and communication with external services.

The description shown in Figure 4 is from an external and internal perspective.

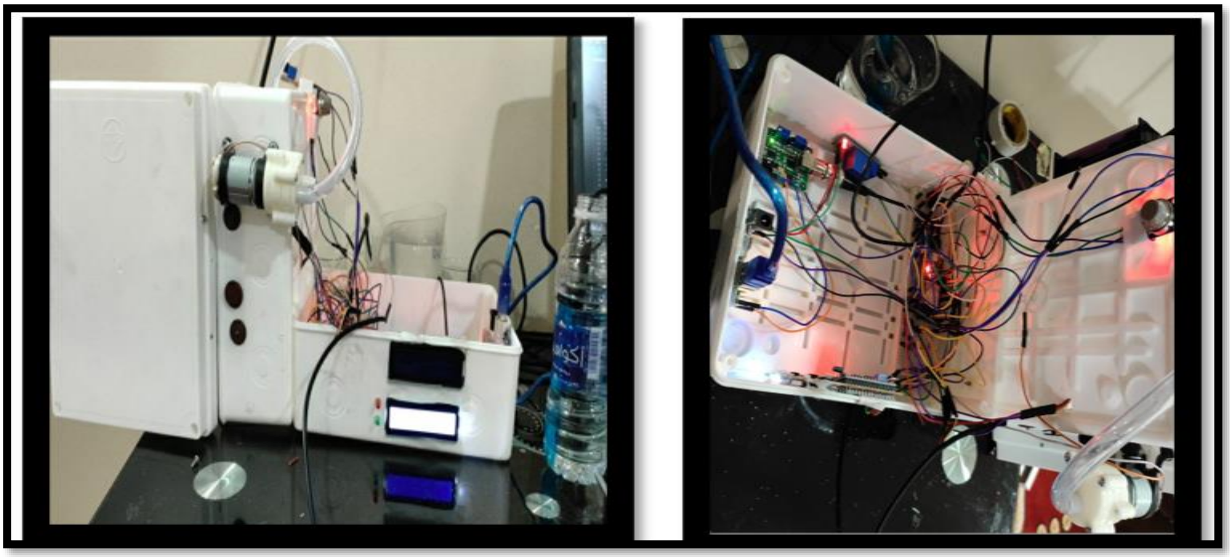


Fig.4. Final system components (external and internal view).

4 RESULTS AND DISCUSSION

the results from two perspectives: the general system result and the IoT result.

4.1 System Testing

To test the system, the software code is uploaded to the Arduino control board and ESP32. The green LED will come on and the LCD will warn of a gas leak. High temperature activates the red LED. Figure 4 shows all conceivable states. As seen in Figures 5 and 6, the water pump activates during a fire.

Fig. 5. Operates Water Pump When Detecting Fire.

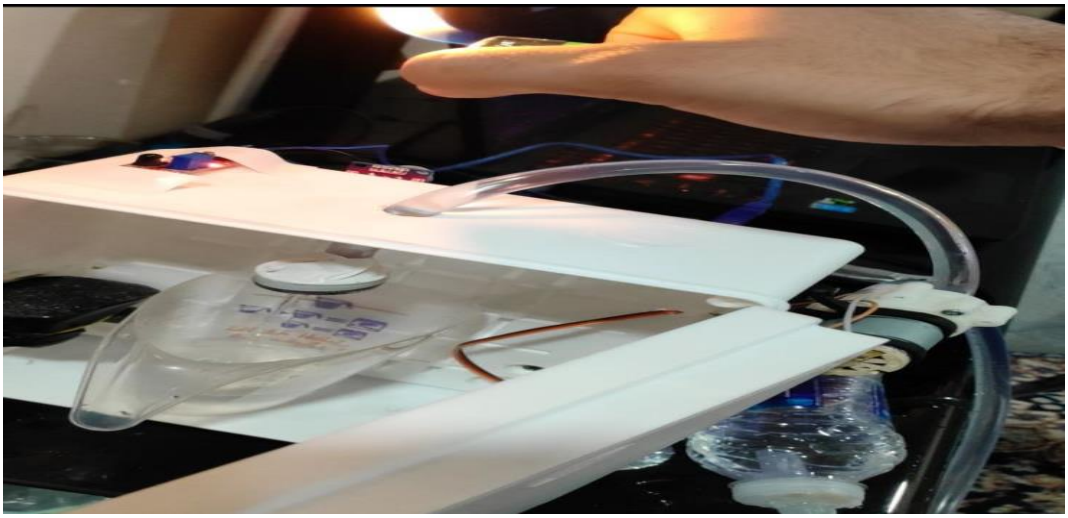


Fig. 6. Operates Water Pump When Detecting Fire.

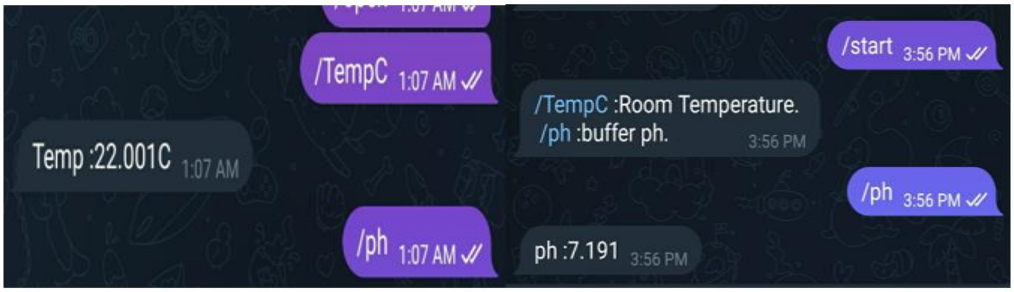
4.2 IoT Testing

An ESP module enables the system to connect to the internet via Wi-Fi. In this project, a Telegram bot is created. The administrator sends commands to the Telegram bot, which retrieves information from the system.

TABLE I. TELEGRAM BOT COMMANDS

System Response	Bot Commands
The System initializes to send information about the box and the vaccine it contains	/start
	/temp
Send current temperatures of the storage box or environment	
Send the current pH of the solution as measured by the pH sensor	/p

Additionally, the bot sends notifications in case of an emergency, as depicted in Figures 7 and 8. Table 1 outlines the system commands for the Telegram bot.



/TempC 1:07 AM

Temp :22.001C 1:07 AM

/ph 1:07 AM

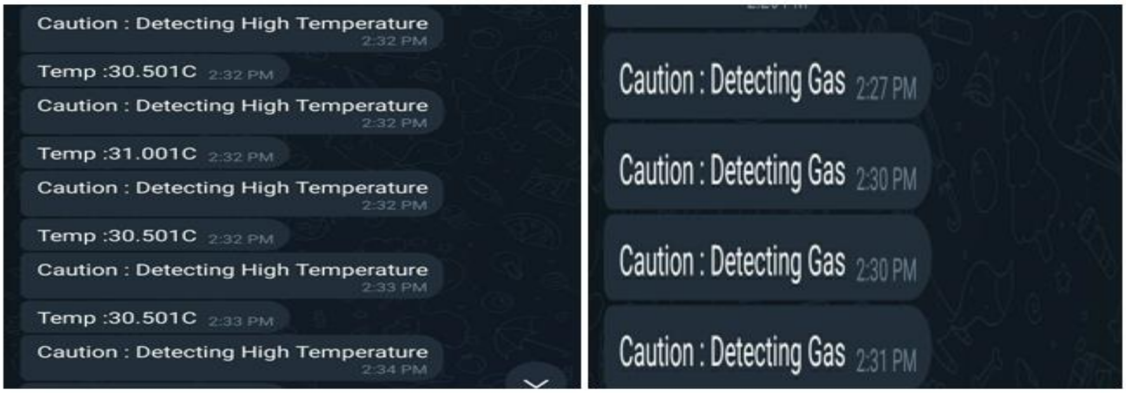
/start 3:56 PM

/TempC :Room Temperature.
/ph :buffer ph. 3:56 PM

/ph 3:56 PM

ph :7.191 3:56 PM

Fig. 7. Commands the Telegram Bot.



Caution : Detecting High Temperature 2:32 PM

Temp :30.501C 2:32 PM

Caution : Detecting High Temperature 2:32 PM

Temp :31.001C 2:32 PM

Caution : Detecting High Temperature 2:32 PM

Temp :30.501C 2:32 PM

Caution : Detecting High Temperature 2:33 PM

Temp :30.501C 2:33 PM

Caution : Detecting High Temperature 2:34 PM

Caution : Detecting Gas 2:27 PM

Caution : Detecting Gas 2:30 PM

Caution : Detecting Gas 2:30 PM

Caution : Detecting Gas 2:31 PM

Fig. 8. Detecting High Temperature and Gas Leak via Telegram Bot.

5. CONCLUSION

This paper presents an IoT-based environmental monitoring system designed to enhance laboratory safety and sustainability. The implementation of the system has resulted in significant benefits. Improved Safety, The IoT system continuously monitors lab conditions, including temperature, gas levels, and fire. Early detection of high temperatures, gas leaks, and fires enhances lab employee safety. Automatic Fire Suppression: The system's water pump plays a vital role in swiftly extinguishing fires, and minimizing damage. Real-time Lab Monitoring: IoT technology enables real-time monitoring of temperature, gas levels, and other critical parameters. and quick decision-making and problem-resolution rely on real-time data. Data Visualization: LCD screens provide a clear visualization of sensor data, including temperature, gas status, and pH. This visualization accelerates laboratory tests and facilitates monitoring of lab conditions. Safety Alarms: Buzzer alarms for gas leaks or fires enhance safety by providing audible alerts. Personnel can be alerted to emergencies even if they are not actively monitoring the system. Remote Monitoring, The ability to send IoT device instructions for remote lab data access is advantageous. and lab managers can remotely monitor lab conditions, improving situational awareness. Improved Sustainability: The system contributes to sustainability by preventing chemical spills, fires, and environmental damage. and sustainability goals include accident prevention and cost reduction. Cost-cutting: IoT technology can proactively resolve safety issues, reducing accident-related costs, equipment damage, and downtime.

The IoT-based environmental monitoring system described in this study significantly enhances laboratory safety and sustainability. Real-time monitoring, automatic fire suppression, and remote access provide administrators with security and peace of mind. A sensor data display and warning system improve decision-making and incident prevention. IoT-based laboratory monitoring has the potential to enhance lab safety and sustainability further. Future areas of improvement for the system include:Advanced Sensor Integration (e.g., air quality, humidity, chemical concentrations), Machine learning, and predictive analytics for safety forecasting energy efficiency through efficient power sources and low-energy communication, Scalability considerations for larger laboratories or multiple labs, and enhanced cybersecurity for remote monitoring. Integration with Laboratory Management Systems, compliance with safety and environmental regulations.user feedback and training for system optimization.Performance Metrics and Case Studies for Real-World Impact Assessment, Interoperability standards for lab equipment and systems Additional enhancements may include lab door security, gas sensors for detecting other gases, and GPS for emergency response coordination.

Conflicts Of Interest

No conflicts of interest.

Funding

the research was conducted without any funding from external sources.

Acknowledgment

The Authors would like to thank Mustansiriyah University(https://uomustansiriyah.edu.iq/) in Baghdad –Iraq, for its support in the present work.

References

[1] Department of Enterprise, Trade, and Employment. "Guidelines to the Safety, Health, and Welfare at Work (Chemical Agents) Regulations, 2001." Department of Enterprise, Trade, and Employment, 2001.
[2] Handbook of Research on the Internet of Things Applications in Robotics and Automation. [Online]. Available: https://www.igi-global.com/section/smart-plant-monitoring-system-using-iottechnology/237293. Accessed: May 14, 2022.
[3] Z. Wang et al., "Masked Face Recognition Dataset and Application," [Preprint]. Available: https://arxiv.org/pdf/2003.09093.pdf, 2020.
[4] G. A. Mabbott, "Teaching Electronics and Laboratory Automation Using Microcontroller Boards," J. Chem. Educ., vol. 91, no. 9, pp. 1458–1463, 2014.
[5] H. F. Launer, "Determination of the pH Value of Papers," Year.
[6] A. M. Zambrano et al., "Community early warning systems," in Wireless Public Safety Networks 3, Elsevier, 2017, pp. 39-66.
[7] H. Kopetz, "Real-Time Systems, Design Principles for Distributed Embedded Applications," Klower Academic Publishers, 1997, ch. 10.
[8] S. F. Barrett, "Arduino microcontroller processing for everyone!," Synthesis Lectures on Digital Circuits and Systems, vol. 8, no. 4, pp. 1-51, 2013.
[9] Analog Devices. "DS18B20." [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ds18b20.pdf.
[10] M. Yuqing et al., "New technology for the detection of pH," Laboratory of Bioanalysis and Biosensor, College of Chemistry and Life Science, Zhejiang Normal University, Year.
[11] A. Siddiqa and S. Johar, "Handling Fire Hazards by Fire Detection System Using IoT," Department of CSE, JNNCE, Shivamogga, Karnataka, India, Year.
[12] L. Ge et al., "Current Applications of Gas Sensor Based on 2-D Nanomaterial: A Mini Review," Year.
[13] Fujitsu. "LCD-Background." [Online]. Available: http://www.fujitsu.com/downloads/MICRO/fma/pdf/LCD_Backgrounder.pdf.
[14] M. Hemmings, "What is Jumper Wire?," [Online]. Available: https://blog.sparkfuneducation.com/what-is-jumper-wire, Jan. 30, 2018.
[15] M. Anandan, "LED Backlight: Enhancement of picture quality on LCD screen," in Proc of ASID, 2006, pp. 130-134.
[16] M. Babiuch, P. Foltýnek, and P. Smutný, "Using the ESP32 microcontroller for data processing," in 2019 20th International Carpathian Control Conference (ICCC), IEEE, pp. 1-6.
[17] K. Taneja and S. Bhatia, "Automatic irrigation system using Arduino UNO," in 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE.
[18] I. Z. Tsypkin and Y. Z. Tsypkin, "Relay control systems," CUP Archive, 1984.
[19] L. Louis, "Working principle of Arduino and using it," International Journal of Control, Automation, Communication, and Systems (IJCACS), vol. 1, no. 2, pp. 21-29, 2016.
[20] Amazon Web Services. "What is the Internet of Things (IoT)?," [Online]. Available: https://aws.amazon.com/ar/what-is/iot/, 2022.
[21] J. C. de Oliveira, D. H. Santos, and M. P. Neto, "Chatting with Arduino platform through Telegram Bot," in 2016 IEEE International Symposium on Consumer Electronics (ISCE), IEEE, pp. 131-132.

[22] S. Madakam et al., "Internet of Things (IoT): A literature review," Journal of Computer and Communications, vol. 3, no. 5, pp. 164-176, 2015.
[23] P. Ganguly, "Selecting the right IoT cloud platform," in 2016 International Conference on Internet of Things and Applications (IOTA), IEEE.
[24] J. C. de Oliveira, D. H. Santos, and M. P. Neto, "Chatting with the Arduino platform through a Telegram bot," in 2016 IEEE International Symposium on Consumer Electronics (ISCE), IEEE.
[25] Z. F. Lutfi et al., "Design and Implementation of an Internet of Things-based Environmental Monitoring System," International Journal of Intelligent Systems and Applications in Engineering, vol. 11, no. 5s, p. 261, 2023.