Journal of Chemical Engineering

Vol. ChE 31, No. 1, January 2023

Article

Design And Development of An Open-Source and Affordable IoT System for a Wide Array of Industrial and Domestic Applications

Anindya Karmaker, Mahmudul Hasan, and Ahaduzzaman Nahid

To cite the article:
A. Karmaker, M. Hasan, A. Nahid (2023), “Design And Development of An Open-Source and Affordable IoT System for a Wide Array of Industrial and Domestic Applications”, Journal of Chemical Engineering, IEB, ChE 31 (1), pp. 62-68.

The Institute of Engineers, Bangladesh (IEB)
Chemical Engineering Division

Journal of Chemical Engineering, IEB
Vol. ChE 31, No. 1, January 2023

Design And Development of An Open-Source and Affordable IoT System for a Wide Array of Industrial and Domestic Applications

Anindya Karmaker, Mahmudul Hasan, and Ahaduzzaman Nahid*

Department of Chemical Engineering, Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh

Received: 5 September 2022; Accepted: 19 October 2022; Published: 30 April 2023

Abstract: The Internet of Things (IoT) has improved the quality of our lives through various interconnected gadgets and devices. Despite having great potential, IoT systems still have not made it into the mainstream industries or residential complexes due to the high cost associated with commercial IoT solutions particularly in developing or least developed countries. In-expensive and open-source IoT-based systems could be employed in these fields to improve overall safety and drastically reduce accidents. In this paper, a model IoT system has been presented that could be used in any commercial apartment or building complex or retrofitted to existing industry and can augment safety and reliability along with improving the quality of life. For this purpose, Atmega1284P and ESP32 microcontroller-based IoT device was developed and implemented. For IoT integration and cloud operation, the Google Firebase IoT platform was chosen. To monitor the system using both Bluetooth (low energy, BLE) and Wi-Fi, an Android application was created using MIT App Inventor 2. This device is capable of receiving and handling multiple analog or digital sensors and will send an automated email whenever a threshold value for the sensor is exceeded which could be easily changed via the application. Furthermore, the delay and system response of this device were experimentally determined. This type of system could be easily scaled for various applications which are also discussed in this paper. It also offers better flexibility and cost-effective solution to commercially available alternatives.

Keywords: IoT, ESP32, Atmega1284P, Firebase, Industrial Safety, Automated email response

1. Introduction

The Internet of Things (IoT) is a network of physical objects embedded with sensors, software, and other technologies to connect and exchange data with other devices or systems over the internet[1]. IoT has taken the modern technological world by storm[2] and is omnipresent in our day-to-day lives. From smart sensors to smart cars, these devices are improving people's lives and making them easier and safer[3], [4]. However, these systems are still not being used to their full extent since they are very expensive. Almost every day in many industries and residential complexes, various accidents are occurring. These accidents could easily be prevented if there were automatic response systems instead of age-old systems that rely on human intervention. For instance, fire hazard is one of the major causes of accidents in developing and least-developed countries[5]. Electrical short circuits and gas leakage are two of the most common causes of fire hazards. However, despite the presence of cheap gas leakage detectors or fire alarms, many accidents could not be prevented since they require some form of human interaction, which is time-consuming and consequently increases the damage. Some industries and apartments do not even have gas detectors, smoke, or fire alarms to alert employees in case of an accident. Those that do have such systems are mostly manually operated, and hence fail to serve their purpose if no one is present on-site to monitor them. Moreover, various small industries in developing countries do not usually require sophisticated control systems, and sometimes they do not even have the capital to invest in modern control and monitoring systems like DCS (distributed control system) or SCADA (supervisory control and data acquisition)[6]. Though there are some inexpensive but sophisticated IoT monitoring systems available nowadays, they are quite unstable for 24x7 operation and have cheap components with closed source software. Some commercial IoT systems employ a special watchdog service for added reliability but charge a premium for added features[7]. There has been a lot of research into developing inexpensive IoT solutions for developing countries, however, they use new cloud platforms or LoRa based systems[8], [9]. In this paper, a model IoT system has been presented which has several analog and digital sensors and costs less than 100 USD. This system uses traditional communication systems like Wi-Fi and Bluetooth but could also be adapted for LoRa operation. Since this system is built from scratch, it is possible to choose the best components required and adjust the costing appropriately. A detailed framework and flowchart of the design and working process are explained in this paper for a better understanding of this system so that anyone can build it using this system as a reference. The major advantage of this system is that it can provide real-time monitoring of important process variables with an automated response system to alert the user or emergency response services in case of an accident as early as possible to minimize the loss of lives and damage of property. An authorized user can monitor this system from anywhere in the world using an internet connection. This system could be scaled as required and due to the open-source nature of this system, it could be improved or modified without needing expensive licensing permission. This system can also control other devices (via infrared), like turning on the exhaust fan or an appliance as required which is a useful feature not present in even some of the commercial solutions. To summarize, industrial and residential development in developing and least-developed countries are occurring at an unprecedented rate. However, the number of accidents is also increasing proportionally. This project aims to reduce avoidable accidents by warning and alerting the people beforehand, and at least keep the damage to a minimum with the least investment cost.

* Corresponding Author: Ahaduzzaman Nahid,
E-mail: ahaduzzaman@che.buet.ac.bd

2. Instrumentation and Design

Design: The IoT system consisted of two independent microcontroller circuits. One of the systems was powered by Atmega1284P, which houses various ADCs (analog to digital converter) and is called the sensor hub for referencing purposes. The microcontroller received input from an RTC module DS3231 to maintain accurate timing, a thermocouple amplifier MAX31855, an RTD-to-digital converter MAX31865, a 40 kHz infrared receiver TSOP31240, and a 16-bit ADC ADS1115. Analog sensors were connected to the ADC, which was operated in single-ended input channel mode. The analog sensors consisted of temperature sensor LM35, gas sensor MQ2, and an analog pH sensor. RCWL-0516 Microwave proximity sensor was embedded into the circuit for monitoring or detecting people. A GPS module, NEO-M8N was also connected to the microcontroller along with four infrared LEDs and a buzzer. This microcontroller board was also connected to a Bluetooth low-energy (BLE) wireless serial module, HM-10.

The other microcontroller system was powered by ESP32. This board is referred to as the communication hub of the system. ESP32 was programmed to receive serial data from the sensor hub circuit and send the data to the Google cloud service, Firebase. The communication hub took less than 30 seconds to connect to the Wi-Fi network and the internet. In case the Wi-Fi network was unavailable, the hub was designed to create its Wi-Fi hotspot in less than 60 seconds. The user could connect to the hotspot and enter the Wi-Fi credentials manually or search nearby Wi-Fi networks and set it up for online operation. ESP32 was chosen due to its availability; however, it could be replaced by a lower-cost ESP8266 board to reduce the cost.

The sensor hub received sensor data from various amplifiers and ADC and transmitted the data via serial communication (UART) to both the BLE module and the communication hub and received serial data (UART) from both. The sensor hub circuit was coded to receive, save and transmit IR codes of devices like Fan, AC, or other IR-based relay modules to control those devices. The system also had a battery management system (BMS) using the conventional TP4056 module, which was modified so that the ADC could measure the battery voltage and give a warning if the BMS fails.

The system could be shifted to a single microcontroller board; however, multiple microcontrollers were used to ensure stable operation, easier serviceability, and maintenance. Both controllers operated in such a way that each could receive and transmit data independently and could be easily diagnosed if one or the other stopped working. Various modules like the GPS or ESP32 were powered using MOSFETS so that they could be turned off for debugging purposes or to reduce power consumption. The system could also be connected to a GSM shield directly so that it could operate without Wi-Fi or send critical information to the user via SMS.

Programming: Both microcontrollers were programmed using the Arduino Integrated Development Environment (IDE). Various libraries were used in the Arduino code, particularly IOXhop_FirebaseESP32, which was used to communicate ESP32 with Firebase. For thermocouple reading, RTD, and ADS1115, libraries from Adafruit were used. Moreover, SPI, Wire, Liquid Crystal, Software Serial, EEPROMAnything, TinyGPS++ by Mikal Hart, Wi-Fi Manager by Zhouhan, and IRremote library by Ken Shirriff were used. The mobile app interface was developed using the MIT App Inventor. The complete block diagram for the mobile application is presented in supplementary figures. Simplified block diagrams for the IoT device (both sensor hub and communication hub) are shown in Figures 1 and 2.

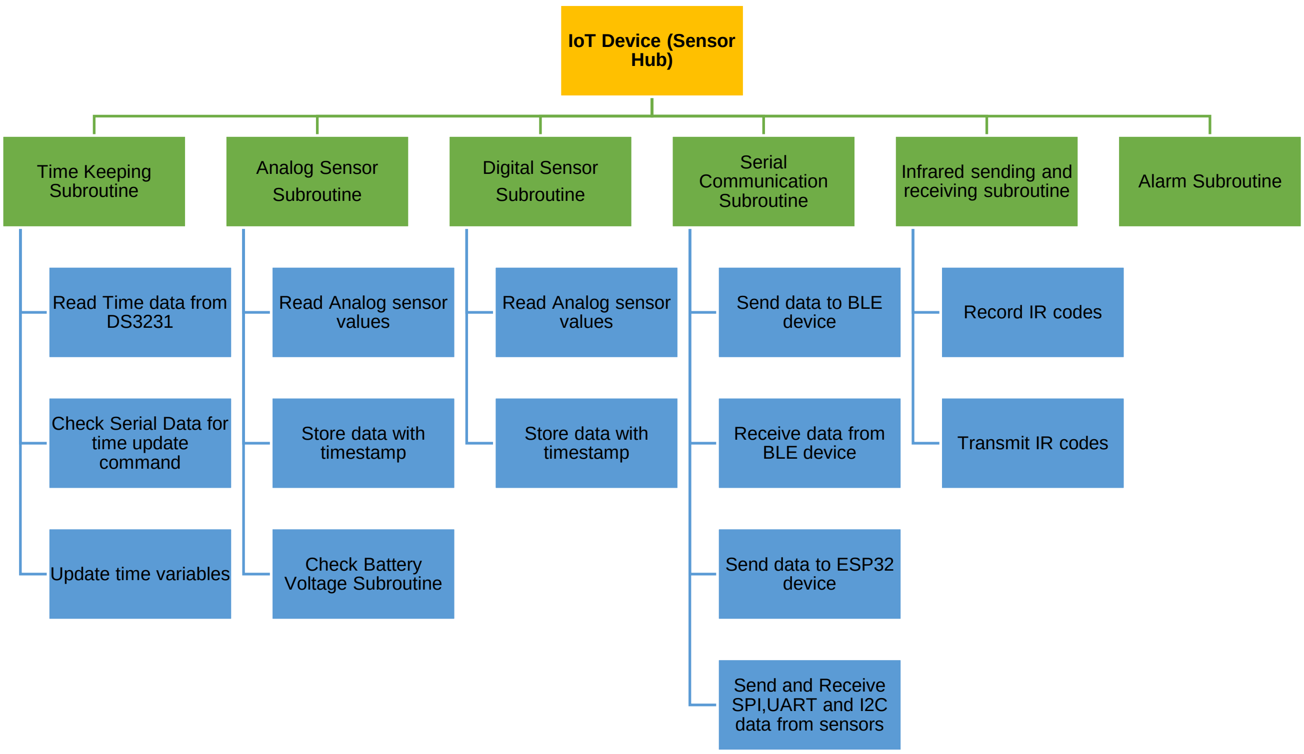

Figure 1: Simplified Block Diagram for Sensor Hub Circuit.

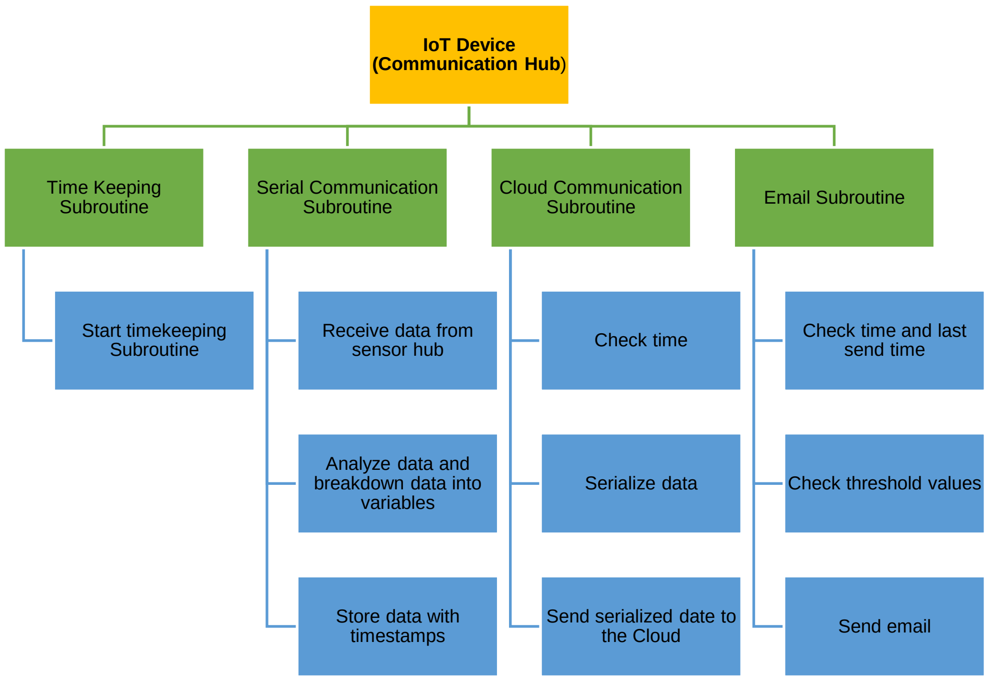

Figure 2: Simplified Block Diagram for Communication Hub Circuit.

From Figure 1, it can be observed that there were five important subroutines for the sensor hub circuit. The first one is the timekeeping subroutine. To accurately measure variables and average the readings, it was necessary to measure time accurately. DS3231 chip was used to measure time. It was connected via the I_2C protocol. To update time, this subroutine regularly checked for the time update command received from the BLE device, like a mobile phone, using the developed application. The accurate time data was stored in variables for other purposes. Analog sensor subroutine checked for sensor value changes. It stored the data with timestamps. It also had a battery voltage measuring subroutine with an alarm function. It had several other subroutines to precisely measure temperature, pH, etc. Similarly, the digital sensor subroutine performed the same task. Serial communication subroutine sent and received data from various I_2C, SPI, or UART sensors. It also stored values with timestamps. The infrared sending and receiving subroutine currently stores sensor values when the user enters the option using the device manually. It has several subroutines to decode and store the IR codes. It can also send the IR codes. However, the process of automatic sending of IR codes via Wi-Fi or based on the threshold values has not yet been implemented. The IR codes can be transmitted via the BLE commands using the mobile application. The alarm subroutine sends a signal to a piezoelectric buzzer to create a loud sound to warn the user if any subroutine sends the alarm command.
From Figure 2, it is observed that there were four important subroutines for the communication hub circuit. The timekeeping subroutine was similar to the sensor hub, however, the major difference was that it used the internal clock of ESP32. The serial communication subroutine also worked in a similar manner. It stored data with timestamps and broke down the data into variables for storage. Cloud communication subroutine checked time, serialized the data into proper formats, and finally sent the data to the Firebase Cloud server. Email subroutine was an important part of the communication hub circuit. It checked the time and stored them. It also compared the sensor values to the threshold values and sent them via email. To avoid spamming, it measured the time difference values and sent them after a regular interval.

Setup: The IoT system had a simple setup. The setup and operation process are illustrated in Figure 3. First, the device was powered on and connected to a 5V power source, the required sensors were connected and their values were monitored through the LCD on the device. The device created a hotspot Wi-Fi network. The user then connected to this hotspot and set up a Wi-Fi connection; ran the mobile application . A green Wi-Fi connection option indicated successfully connection of the device to the internet. To prevent unauthorized access and ensure security, device properties like threshold values and other feature selection options like GPS were activated when the device was connected via Bluetooth that was pin protected. The device then works autonomously and alerts the user or emergency services personnel based on the set threshold values.

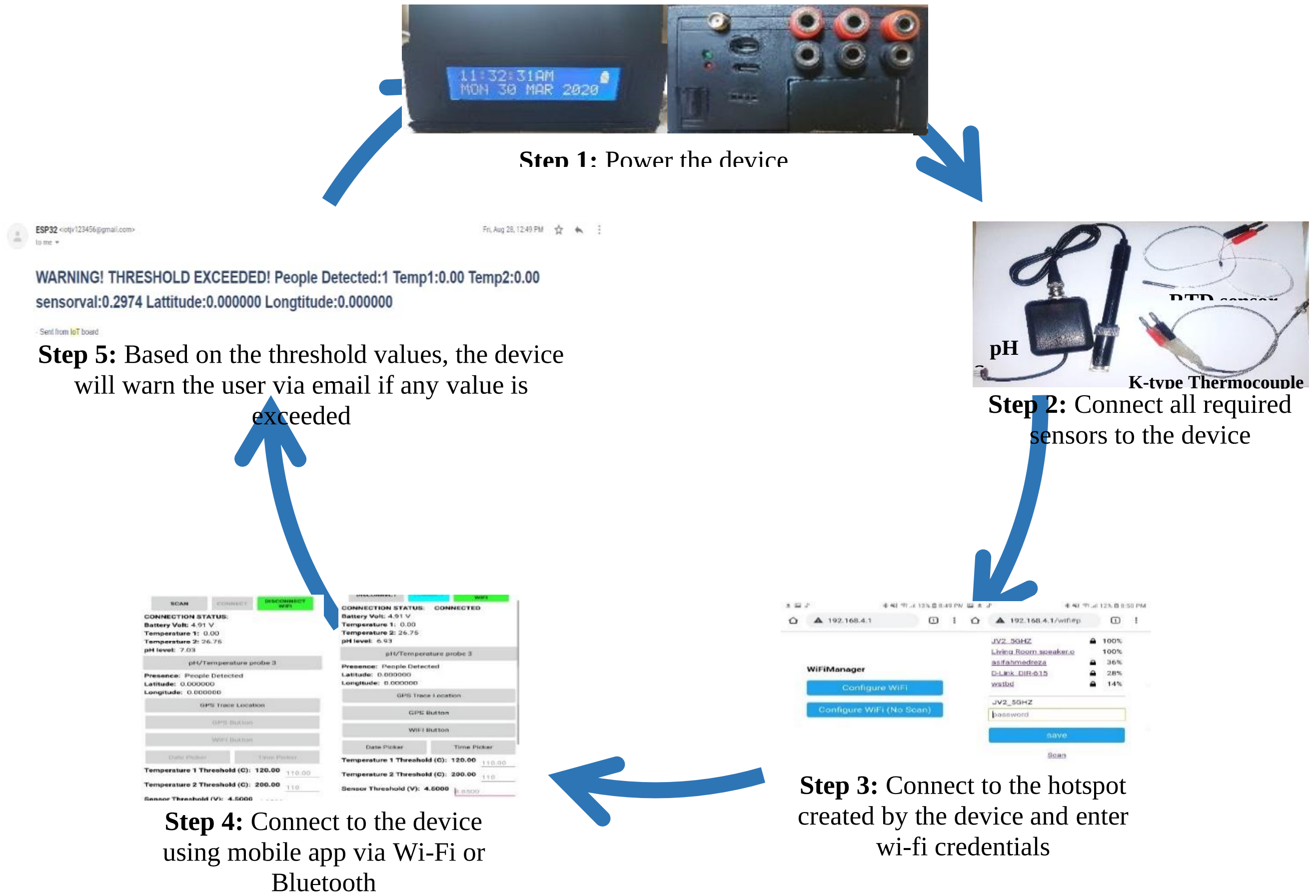

Figure 3: IoT device setup and working procedure.

3. Simulation and Results

Since the sensor hub circuit is complicated and has lots of components and libraries, the circuit was simulated in Proteus 8 Professional, to test how the various libraries would work together and be debugged as required. Using Proteus, the PCB was also designed and fabricated in-house. Atmega1284P was operated at 16MHz using an external crystal. The BLE and ESP32 modules were simulated using two terminals (UART communication). The thermocouple module was simulated using the MAX31855 module. The circuit diagram of the system is shown in Figure 4.

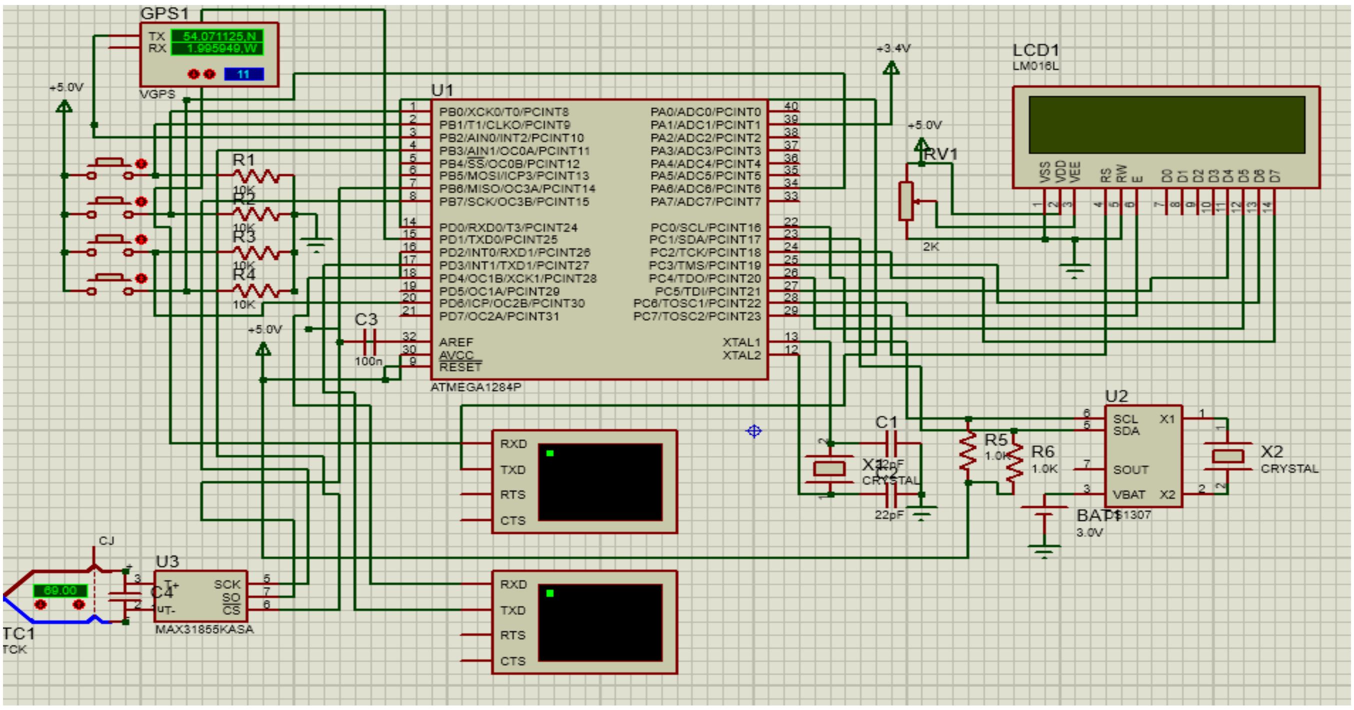

Figure 4: Simulation of sensor hub circuit.

The IoT system was also tested in lab condition to determine the delay and system response. During the testing, system lag, response lag, and accuracy were tested. Thermocouples, pH sensors, GPS sensors, and IR LEDs were tested during this process. Figure 5 and 6 compares the output data from the developed system taken using Wi-Fi and Bluetooth connection simultaneously and input data measured using locally connected sensor probes. This data was used to calculate the delay and evaluate system performance.

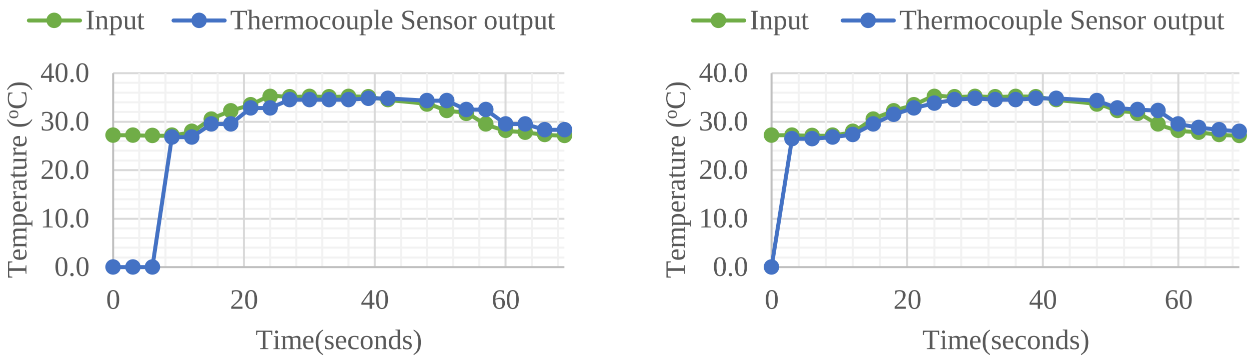

Figure 5: Temperature Input vs. Thermocouple Output on Wi-Fi connection(left) and Bluetooth (right).

A sample water bath with a temperature of ~27°C and pH of ~7.05 was used to determine the temperature and pH sensor accuracy and response lag. An RTD thermometer of 0.1°C accuracy was used for comparing the output from the IoT device and a benchtop pH meter was used to measure the pH. A water bath was heated and then cooled by adding hot and cold water and stirred continuously. The temperature was noted using both thermometer and IoT device. The readings were taken using both Bluetooth and Wi-Fi connection simultaneously at an interval of 3 seconds. There was an initial delay of 3 seconds for Bluetooth and ~6 seconds for Wi-Fi. The thermocouple sensor values varied within a margin of 0.5 °C from the RTD thermometer. The response lag was roughly 1 second for the Bluetooth and about 6 seconds for the Wi-Fi though the initial response lag was 3 seconds and 6 seconds respectively for the Bluetooth and Wi-Fi. The same response lag was also observed from the pH sensor reading values as well.
The whole system was connected to the network and fully operational within 1 minute and 30 seconds after powering on. The sensors, particularly the GPS sensor took an average of 3 minutes to connect to the satellite and send accurate GPS coordinates within 10-meter accuracy without an external antenna. When the Wi-Fi was disconnected, the system warned the user in less than ~15 seconds. The battery backup system with 6000 mAh battery ran the system for an average of ~6 hours 32 minutes. All tests proved the reliability and effectiveness of this device in various industrial and residential apartment applications.

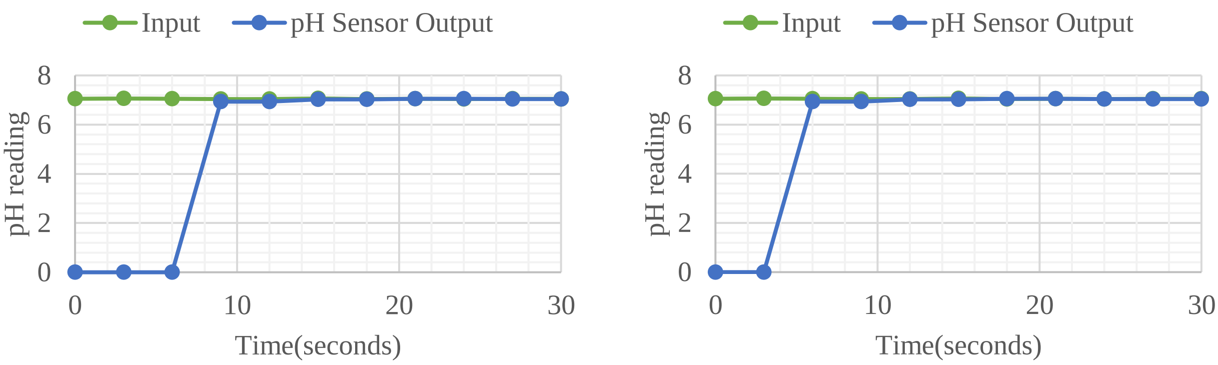

Figure 4: pH Input vs. pH Output on Wi-Fi connection(left) and Bluetooth (right).

4. Applications

This system has a lot of potential applications and therefore could be used in a variety of fields ranging from industry to healthcare. The potential applications of this system are listed below:

1) **Industrial Applications:** This type of device can be used to monitor non-process variables like electricity flow in the wiring, smoke detection, gas leakage detection, people detection and monitoring, etc. Since this device is inexpensive, both large and small-scale industries can adopt it and use it to improve reliability and safety. Moreover, improved security measures could be implemented in this system for better safety and reliability[10], [11].

2) **Smart Home Applications:** This IoT device is perfectly suitable for both small homes to large residential complexes[12]. If this device is present in every apartment then common accidents due to gas leakage or shortcircuit can be easily prevented[13]. It can also monitor for intruders to prevent burglary. Moreover, since only a few sensors are needed, the cost is much lower compared to other market solutions.

3) **Medical Applications:** By connecting smart sensors such as SpO_2, heart rate, etc. to this IoT device the health condition of the patients can be monitored at home or hospitals and doctors can monitor a large number of patients quickly and easily[14], [15]. Moreover, an automated response can possibly prevent unnecessary deaths that usually occur due to negligence or systematic issues[16].

4) **Agricultural Applications:** This device can also be used for agricultural applications[17]. By monitoring soil water content, humidity, temperature, pH, etc. farmers can grow healthier crops easily[18]. Moreover, since all crucial parameters being measured can be sent to the agricultural specialists automatically, they can utilise the data to improvise crop growth and increase yield.

5) **Water Management Applications:** This device can be used to measure the water levels of ponds, lakes, wells, etc. by using various sensors[19], [20]. In addition, long-term water monitoring can signify drought early and appropriate measures can be taken quickly before the situation escalates.

In short, this type of open-source and inexpensive IoT design is useful and can be applied to various fields. From monitoring to preemptive decision making, this system offers a lot of advantages over commercial systems that are both expensive and closed-source in nature[21]. IoT has a lot of potential applications and it is time to utilize this technology in improving the safety and ease of monitoring in industries, hospitals, and houses.

5. Conclusions

Every year, thousands of people die, and millions of dollars are lost due to accidents in industries and households. IoT is not something new and is present in various gadgets and smart-home appliances that we use regularly. Using this developed IoT system, major accidents can be minimized and loss of lives can be prevented. This type of device with its low cost is perfectly suitable for developing countries where industrial and residential development is occurring rapidly, and an affordable solution is needed to prevent accidents and improve quality of life. The proposed IoT-based developed model will provide the groundwork for advanced IoT applications. Using the parameters obtained and the open-source code and designs, it is possible to optimize this device for maximum performance and utilize it for both personal and commercial applications.

Reference

1. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.
2. P. Suresh, J. V. Daniel, V. Parthasarathy, and R. H. Aswathy, “A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment,” in *2014 International Conference on Science Engineering and Management Research, ICSEMR 2014*, 2014.
3. J. F. Ensworth and M. S. Reynolds, “BLE-Backscatter: Ultralow-Power IoT Nodes Compatible with Bluetooth 4.0 Low Energy (BLE) Smartphones and Tablets,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 65, no. 9, pp. 3360–3368, Sep. 2017.
4. M. Gupta, J. Benson, F. Patwa, and R. Sandhu, “Secure Cloud Assisted Smart Cars Using Dynamic Groups and Attribute Based Access Control,” Aug. 2019.
5. T. A. Azad, I. Hasan, M. K. Saha, R. Ahmmed, S. J. Moni, and H. Kabir, “Risk of Fire Disaster: Consequences on Industry Sectors in Bangladesh,” *International Journal of Energy and Sustainable Development*, vol. 3, no. 3, pp. 52–63, 2018.
6. S. Karnouskos and A. W. Colombo, “Architecting the next generation of service-based SCADA/DCS system of systems,” in *IECON Proceedings (Industrial Electronics Conference)*, 2011, pp. 359–364.
7. A. Celesti, L. Carnevale, A. Galletta, M. Fazio, and M. Villari, “A watchdog service making container-based micro-services reliable in IoT clouds,” in *Proceedings - 2017 IEEE 5th International Conference on Future Internet of Things and Cloud, FiCloud 2017*, 2017, vol. 2017-January, pp. 372–378.
8. C. Pham, A. Rahim, and P. Cousin, “WAZIUP: A Low-Cost Infrastructure for Deploying IoT in Developing Countries,” in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, 2018, vol. 208, pp. 135–144.
9. C. Pham, A. Rahim, and P. Cousin, “Low-cost, long-range Open IoT for smarter rural African Villages,” in *IEEE 2nd International Smart Cities Conference: Improving the Citizens Quality of Life, ISC2 2016 - Proceedings*, 2016.
10. D. Kwon, M. R. Hodkiewicz, J. Fan, T. Shibutani, and M. G. Pecht, “IoT-based prognostics and systems health management for industrial applications,” *IEEE Access*, vol. 4, pp. 3659–3670, 2016.
11. I. Mistry, S. Tanwar, S. Tyagi, and N. Kumar, “Blockchain for 5G-enabled IoT for industrial automation: A systematic review, solutions, and challenges,” *Mechanical Systems and Signal Processing*, vol. 135, p. 106382, 2020.
12. B. L. R. Stojkoska and K. V Trivodaliev, “A review of Internet of Things for smart home: Challenges and solutions,” *Journal of Cleaner Production*, vol. 140, pp. 1454–1464, 2017.
13. M. Alaa, A. A. Zaidan, B. B. Zaidan, M. Talal, and M. L. M. Kiah, “A review of smart home applications based on Internet of Things,” *Journal of Network and Computer Applications*, vol. 97, pp. 48–65, 2017.
14. S. Tyagi, A. Agarwal, and P. Maheshwari, “A conceptual framework for IoT-based healthcare system using cloud computing,” in *Proceedings of the 2016 6th International Conference - Cloud System and Big Data Engineering,*

Confluence 2016, 2016, pp. 503–507.
15. M. Hassanalieragh *et al.*, "Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges," in *2015 IEEE International Conference on Services Computing*, 2015, pp. 285–292.
16. F. Feria, O. J. S. Parra, and B. S. R. Daza, "Design of an architecture for medical applications in IoT," in *International Conference on Cooperative Design, Visualization and Engineering*, 2016, pp. 263–270.
17. D. Garg, S. Khan, and M. Alam, "Integrative use of IoT and deep learning for agricultural applications," in *Proceedings of ICETIT 2019*, Springer, 2020, pp. 521–531.
18. H. N. Saha, R. Roy, M. Chakraborty, and C. Sarkar, "IoT-Enabled Agricultural System Application, Challenges and Security Issues," *Agricultural Informatics: Automation Using the IoT and Machine Learning*, pp. 223–247, 2021.
19. C. Rajurkar, S. R. S. Prabaharan, and S. Muthulakshmi, "IoT based water management," in *2017 International Conference on Nextgen Electronic Technologies: Silicon to Software (ICNETS2)*, 2017, pp. 255–259.
20. P. Verma *et al.*, "Towards an IoT based water management system for a campus," in *2015 IEEE First International Smart Cities Conference (ISC2)*, 2015, pp. 1–6.
21. A. Al-Sakran, M. H. Qutqut, F. Almasalha, H. S. Hassanein, and M. Hijjawi, "An Overview of the Internet of Things Closed Source Operating Systems," in *2018 14th International Wireless Communications and Mobile Computing Conference, IWCMC 2018*, 2018, pp. 291–297.