Design and Development of an Open-Source and Affordable LoT System for Developing and Least Developed Countries

Anindya Karmaker  
Bangladesh University of Engineering and Technology

Mahmudul Hasan  
Bangladesh University of Engineering and Technology

Ahaduzzaman Nahid  
Bangladesh University of Engineering and Technology

Research Article

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

DOI: https://doi.org/10.21203/rs.3.rs-778044/v1

License: ☕️ Ⓟ This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License
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. Inexpensive 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.

I. 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 it 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, the fire hazard is one of the major causes of accidents in developing and least developed countries[5]. Electrical short circuit and gas leakage are two of the common causes of fire hazard. However, despite the presence of cheap gas leakage detectors or fire alarms, many accidents could not be prevented since they require some sort of human interaction which wastes time and increases the damage. Some industries and apartments do not even have gas detectors, smoke, or fire alarms to alert the 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 WiFi 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 could build it using this system as a reference. The advantages 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 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 are also increasing proportionally. This project aims to reduce avoidable accidents by warning and alerting the people beforehand and at least keep the damage to the minimum with the least investment cost.

ii. Instrumentation And Design

Design

This IoT system consists of two independent microcontroller circuits. One of them is powered by Atmega1284P which houses various ADCs (analog to digital converter) and is called the sensor hub for referencing purposes. This microcontroller receives 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 are connected to the ADC which is operated in single-ended input channel mode. The analog sensors consist of temperature sensor LM35, gas sensor MQ2, and an analog pH sensor. RCWL-0516 Microwave proximity sensor is embedded into the circuit for monitoring or detecting people. A GPS module, NEO-M8N is also connected to the microcontroller along with four infrared LEDs and a buzzer. This microcontroller board is also connected to a Bluetooth low-energy (BLE) wireless serial module, HM-10.

The other microcontroller system is powered by ESP32. This board is referred to as the communication hub of the system. ESP32 is programmed to receive serial data from the sensor hub circuit and send the data to the Google cloud service, Firebase. The communication hub takes less than 30 seconds to connect to the WIFI network and to the internet. If the WIFI network is not available, then it creates its WIFI hotspot in less than 60 seconds. The user can connect to the hotspot and enter the WIFI credentials
manually or search nearby WIFI 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 receives sensor data from various amplifiers and ADC and transmits the data via serial communication (UART) to both the BLE module and the communication hub and also receives serial data (UART) from both. The sensor hub circuit is 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 has a battery management system (BMS) using the conventional TP4056 module which is modified so that the ADC can measure the battery voltage and give a warning if the BMS fails.

The system could be shifted to a single microcontroller board, however, the reason for employing multiple microcontrollers was to ensure stable operation, easier serviceability, and maintenance. Both controllers operate in such a way that each can receive and transmit data independently and can be easily diagnosed if one or the other stops working. Various modules like the GPS or ESP32 are 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 can operate without WIFI or send critical information to the user via SMS.

**Programming:** Both of the 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 reading thermocouple, RTD, and ADS1115, libraries from Adafruit were used. Moreover, SPI, Wire, Liquid Crystal, Software Serial, EEPROMAnything, TinyGPS++ by Mikal Hart, Wifi Manager by Zhouhan, and IRremote library by Ken Shirriff were used. The mobile app interface was developed using MIT App Inventor. The complete block diagram for the mobile application is available in supplementary figures. A simplified block diagram for the IoT device (both sensor hub and communication hub) is given in figures 1 and 2.

From figure 1, it can be observed that there are five important subroutines for the sensor hub circuit. The first one is the timekeeping subroutine. To accurately measure variables and average the readings, it is necessary to measure time accurately. DS3231 chip was used to measure time. It was connected via the I\textsubscript{2}C protocol. To update time, this subroutine regularly checks for the time update command received from the BLE device like a mobile phone using the developed application. This accurate time data is stored in variables for other purposes. Analog sensor subroutine checks for sensor value changes. It stores the data with timestamps. It also has a battery voltage measuring subroutine with an alarm function. It has several other subroutines to precisely measure temperature, pH, etc. Similarly, the digital sensor subroutine performs the same task. Serial communication subroutine sends and receives data from various I\textsubscript{2}C, SPI, or UART sensors. It also stores 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 WIFI or based on the threshold values is not yet 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 Fig. 2, it can be seen that there are four important subroutines for the communication hub circuit. The timekeeping subroutine is similar to the sensor hub one, however, one major difference is that it uses the internal clock of ESP32. The serial communication subroutine also works similarly. It stores data with timestamps and breaks down the data into variables for storage. Cloud communication subroutine checks time, serializes the data into proper formats, and finally sends the data to the Firebase Cloud server. Email subroutine is an important part of the communication hub circuit. It checks the time and stores them. It also compares the sensor values to the threshold values and sends them via email. To avoid spamming, it measures the time difference values and sends them after a regular interval.

Setup: It is very easy to set up the IoT system. The setup and operation process is illustrated in figure 3. Firstly, the device is powered on and connected to a 5V power source, the required sensors are connected and their values are monitored through the LCD on the device. In the meantime, the device creates a hotspot wifi network. The user needs to connect to this hotspot and set up a wifi connection. Then the mobile application needs to be run and if the wifi connection option turns green then the device is successfully connected to the internet. For security reasons, the device properties like threshold values and other feature selection options like GPS can be activated when the device is connected via Bluetooth only which is pin protected to prevent unauthorized access. The device now works autonomously and alerts the user or emergency services personnel based on the set threshold values.

iii. Simulation And Results

Since the sensor hub circuit is complicated and has lots of components and libraries, so the circuit was simulated in Proteus 8 Professional, to test how the various libraries would work together and 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.

The IoT system was also tested in a lab condition to determine the delay and system response. During the testing, the system lag, response lag, and accuracy were tested. Thermocouple, pH sensor, GPS sensor, and IR LEDs were tested during this process. A sample water bath with a temperature of ~27°C and a 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 bench pH meter was used to measure the pH. A water bath was heated and then cooled by adding hot and cold water and continuously stirred. The temperature was noted using both the thermometer and the IoT device. The readings were taken from both Bluetooth and WiFi simultaneously at an interval of 3 seconds. There was an initial delay of 3 seconds for Bluetooth and ~6 seconds for WiFi. 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 WiFi though the initial response lag was 3 seconds and 6 seconds respectively for the Bluetooth and WiFi. 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 sent accurate GPS coordinates within 10-meter accuracy without an external antenna. When the WIFI 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 of the tests proved the reliability and effectiveness of this device in various industrial and residential apartment applications.

Iv. Applications

This system has a lot of potential and 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 could 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 very inexpensive, so both large and small scale industries can adopt it and use it to improve the 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 could be easily prevented[13]. It could also monitor for intruders to prevent burglary. Moreover, since only a few sensors would be needed so the cost will be even lower.

3) **Medical Applications**: By connecting smart sensors such as SpO$_2$, heart-rate, etc. to this IoT device the health condition of the patients could be monitored at home or hospitals and doctors could monitor a huge number of patients quickly and easily[14], [15]. Moreover, automated response could prevent unnecessary deaths that usually occur due to negligence or systematic issues[16].

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

5) **Water Management Applications**: This device could be used to measure the water levels of ponds, lakes, wells, etc. by using various sensors[19], [20]. Moreover, long term water monitoring could signify drought early and appropriate measures could 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 which 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.

V. Conclusion

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, lots of major accidents could be minimized and quick action could be taken before the situation escalates without human interaction. This type of device with its low budget is perfectly suitable for developing and least developed 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 obtained parameters and open-source code and designs, it is possible to optimize this device for maximum performance and employ it for both personal and commercial applications.

Declarations

DATA AVAILABILITY STATEMENT

The authors declare that all data supporting the findings of this study are available within the article and its supplementary information files.

References

[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. Ahmed, 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. Hassanaleragh *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.

**Figures**

![Diagram](image)

**Figure 1**
Simplified Block Diagram for Sensor Hub Circuit

Figure 2

Simplified Block Diagram for Communication Hub Circuit
Figure 3

IoT device setup and working procedure

**Step 1:** Power the device

**Step 2:** Connect all required sensors to the device

**Step 3:** Connect to the hotspot created by the device and enter

**Step 4:** Connect to the device using mobile app via WIFI or Bluetooth

**Step 5:** Based on the threshold values, the device will warn the user via email if any value is
Figure 4

Simulation of sensor hub circuit

Figure 5

Temperature Input vs. Thermocouple Output on WIFI connection (left) and Bluetooth (right)
**Figure 6**

pH Input vs. pH Output on WIFI connection (left) and Bluetooth (right)

**Supplementary Files**

This is a list of supplementary files associated with this preprint. Click to download.

- MITappinventorCODEBLOCKS.png
- SupplementaryData.xlsx