AJEMA
GLOBAL RESEARCH NETWORK

AMERICAN Journal of Engineering, Mechanics and Architecture
Volume 01, Issue 09, 2023 ISSN (E): 2993-2637

Gas Leakage and Fire Detector Based on Internet of Thing (IoT) Network

Nebras Jalel Ibrahim
Computer Center, University of Diyala

Abstract: The Internet of Things (IoT) is a growing network of interconnected devices equipped with various sensors and internet connectivity. This widespread adoption of IoT devices, particularly in industrial automation, has paved the way for real-time remote monitoring and control of industrial equipment. Traditional gas leak detection systems, while precise, fall short in effectively alerting people about leaks. To address this, we've developed an IoT-based gas leak detector with smart alerting capabilities. Our device utilizes data analytics on sensor readings to predict hazardous situations and send timely alerts to both concerned authorities and individuals in the vicinity, enabling proactive measures to mitigate potential risks. Thus, this gives manufacturers the ability to automate, and therefore optimize their operating efficiency. The purpose of this project is to introduce an industrial monitoring system design based on IoT networks. The proposed system is divided into two parts: the first part is to detect and identify the gas leakage level and fire alarm detector by using MQ-2 sensor. The second part is to monitor the proposed system based on thing speak. The thing speak is a cloud platform and it is an open-source of IoT application which can receive the sensors data via ESP32 module and plots these data in a field graph within various parameters.

Keywords: Gas sensor; Internet of Things; Thing Speak; ESP32; Gas leakage; Fire Detector.

I. INTRODUCTION

Internet of Things (IoT) is the networking of physical objects that contain electronics embedded within their architecture in order to communicate and sense interactions amongst each other or with respect to the external environment[1].

In the upcoming years, IoT-based technology will offer advanced levels of services and practically change the way people lead their daily lives. Advancements in medicine, power, gene therapies, agriculture, smart cities, and smart homes are just a very few of the categorical examples where IoT is strongly established[2].

The IoT uses in many fields such agricultures, medicines, industries, militaries, etc. In industries, the IoT networks is use maintain the safety of workers from the danger of gases and combustion, as it was for monitoring, alerting about the occurrence of a disaster, avoiding the danger, and preserving everyone and the workflow.

In this paper, we design a system which enables early detection of factory fire and gas leaks. Our system safeguards against harmful gas incidents. It not only detects gas leaks and triggers audible alarms but also utilizes IoT technology to alert concerned authorities promptly, preventing potential mishaps.

Sensor data is transmitted via the ESP32 module and visualized on a field diagram, allowing for system monitoring anytime, anywhere. The system's logic operates based on defined fire/hazard rules and communication protocols between dedicated devices.

II. PROPOSED SYSTEM

This proposed system is mainly focused on processing monitoring and management by ESP32. It also communicates to other system and to web server and pc. Which is termed as

Master and slave communication. This system is so flexible and can be easy to monitor [3] .the system is shown in figure (1).The first part is to detect and determine the level of gas leakage detector by using MQ-2 sensor and fire alarm detector .Where a buzzer sounds continuously in the event of a gas presence and an intermittent alarm sounds in the event of a fire. The second part is observing the proposed system based on Thing speak. Thing Speak is a cloud platform which is an open source IoT application that can receive sensor data via the ESP32 module and plot this data in a field diagram within various parameters. It enables us to monitor system and data outside or inside the factory.

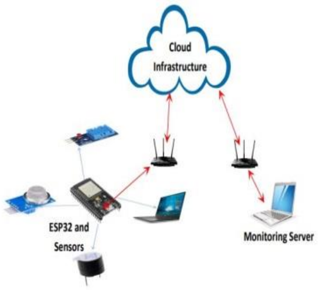


Figure1. The proposed system.

The proposed system is contained:

Gas sensor

The MQ-2 gas sensor relies on tin dioxide (SnO_2) as its sensitive material. SnO_2 exhibits lower conductivity in clean air environments. However, when exposed to flammable gases, the sensor's conductivity increases proportionally with the gas concentration. This change in conductivity can be translated into an output signal representing gas concentration using a simple circuit[4]. The MQ-2 gas sensor demonstrates a high sensitivity to propane and smoke, while also effectively detecting natural gas and other flammable vapors. Its low cost and versatility make it suitable for various applications involving the detection of different types of flammable gases[5].

Figure 2.MQ-2 sensor.

IoT kit (Esp32 model)

ESP32 stands out as the industry's most integrated solution for Wi-Fi and Bluetooth applications, requiring only a handful of external components (fewer than 10).

ESP32 seamlessly integrates essential components such as the antenna switch, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules, resulting in a remarkably compact Printed Circuit Board (PCB) footprint. Leveraging CMOS technology, ESP32 boasts a fully integrated single-chip radio and baseband, further reducing the need for external components. Additionally, it incorporates advanced calibration circuitry that dynamically adapts to external circuit imperfections and environmental changes, ensuring consistent performance. These advancements in integration and adaptability have eliminated the need for costly and specialized Wi-Fi test equipment during mass production, streamlining the manufacturing process. [6].

Figure 3. NodeMCU (Esp32) model

Temperature and Humidity Sensor (Type DHT11)

DHT11 is a cost-effective digital sensor that measures temperature and humidity. It is compatible with various microcontrollers, such as Arduino and Raspberry Pi, enabling real-time humidity and temperature monitoring[7].

DHT11 detects humidity by measuring the electrical resistance between two electrodes. As moisture is absorbed by the sensor's substrate, ionization occurs, increasing conductivity between the electrodes. The relative humidity is directly proportional to the change in resistance.

DHT11 comprises a capacitive humidity sensing element and a thermistor for temperature measurement. The humidity sensing capacitor features two electrodes with a moisture-holding substrate as the dielectric. Changes in humidity levels alter the capacitance value. An integrated circuit (IC) measures and processes the resistance changes, converting them into digital signals.

DHT11 offers several advantages, including[8]:

- Low cost
- Ease of use
- Compatibility with various microcontrollers
- Real-time humidity and temperature monitoring

Figure 4.DHT11 sensor.

Buzzer

This buzzer is self-contained, generating a consistent sound at a predetermined frequency upon receiving a steady direct current (DC) power supply. [9].

Figure5.Buzzer

III. EXPERIMENTAL RESULT

1-Python

Python is a popular programming language with a wide variety of applications including data science, web development, scientific computing, and software development[10].

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics[11]. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together[12]. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance[13]. Python supports modules and packages, which encourages program modularity and code reuse[14]. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed[15].

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library[16].

Features of Python

As a programming language, the features of Python brought to the table are many. Some of the most significant features of Python are[17]:

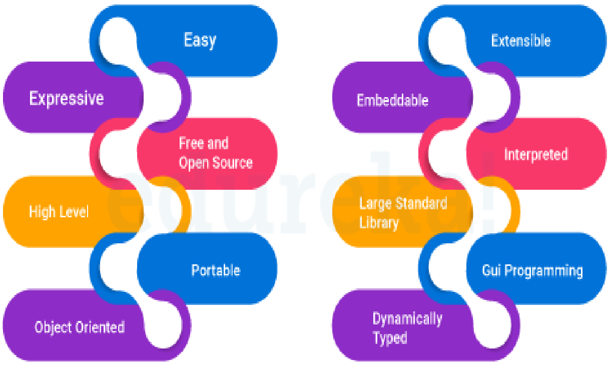


Figure 6. Features of Python

Python in Real-World Scenarios

Python is used in virtually every industry and scientific field that you can imagine, including[18]:

- Data Science.
- Machine Learning.
- Web Development.
- Computer Science Education.
- Computer Vision and Image Processing.
- Game Development.
- Medicine and Pharmacology.
- Biology and Bioinformatics.
- Neuroscience and Psychology.
- Astronomy.
- Other areas such as robotics, autonomous vehicles, business, meteorology, and graphical user interface (GUI) development.

2-Thonny

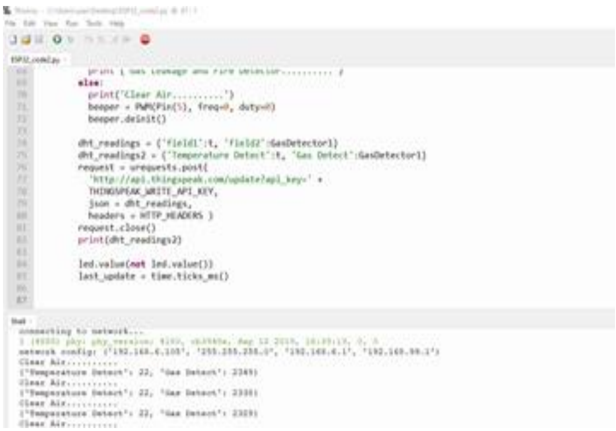
Thonny is a beginner-friendly Python programming environment that offers a variety of features to enhance the learning process.

It provides multiple code stepping methods, step-by-step expression evaluation, and a comprehensive visualization of the call stack to aid in understanding program execution.

Additionally, Thonny introduces the concepts of references and heap through a dedicated mode, making these abstract concepts more approachable for new programmers. [19].

Feature:

1. Intuitive Interface: Thonny's interface is designed to be beginner-friendly, with large icons, clear menus, and a clear separation of code, output, and variables.
2. Line Numbers and Highlighting: Thonny displays line numbers for easy navigation and highlights syntax errors and invalid code.
3. Code Completion and Auto-Formatting: Thonny provides code completion suggestions and automatically formats code to improve readability and reduce errors.
4. Step-by-Step Debugger: Thonny's debugger allows users to step through their code line by line, evaluating expressions, and examining variable values at each step.
5. Immediate Window: Thonny's immediate window enables interactive experimentation with Python expressions and variables.
6. Error Checking and Explanations: Thonny provides detailed error messages and explanations to help users understand and resolve errors they encounter.
7. Data Visualization: Thonny can generate various data visualizations directly within the IDE, making it easier to visualize and analyze data.
8. Integration with Third-party Libraries: Thonny integrates seamlessly with various Python libraries, allowing users to import and use them for their projects.
9. Educational Resources: Thonny provides built-in educational resources, including tutorials, exercises, and example projects.
10. Cross-Platform Compatibility: Thonny runs on Windows, macOS, and Linux, making it available to a wide range of users.
11. Code Assistant: Thonny's code assistant provides contextual information about Python concepts and syntax.
12. Variables View: Thonny's variables view displays the values of variables in real-time as the code is executed.



```
            print('Gas Leakage and Fire Detection..........')
        else:
            print('Clear Air..........')
            beeper = PWM(Pin(5), freq=0, duty=0)
            beeper.deinit()

        dht_readings = {'field1':t, 'field2':GasDetector1}
        dht_readings2 = {'Temperature Detect':t, 'Gas Detect':GasDetector1}
        request = urequests.post(
            'http://api.thingspeak.com/update?api_key=' +
            THINGSPEAK_WRITE_API_KEY,
            json = dht_readings,
            headers = HTTP_HEADERS )
        request.close()
        print(dht_readings2)

        led.value(not led.value())
        last_update = time.ticks_ms()
```

```
connecting to network...
I (4550) phy: phy_version: 4180, cb3948e, Sep 12 2018, 16:39:13, 0, 0
network config: ('192.168.6.105', '255.255.255.0', '192.168.6.1', '192.168.99.1')
Clear Air..........
{'Temperature Detect': 22, 'Gas Detect': 2349}
Clear Air..........
{'Temperature Detect': 22, 'Gas Detect': 2330}
Clear Air..........
{'Temperature Detect': 22, 'Gas Detect': 2329}
Clear Air..........
```

Figure 7. Thonny software.

3- Public cloud (Thing Speak)

Thing Speak is an open-source IoT analytics platform that allows users to collect, store, and analyze data from connected devices. It provides a web-based interface for visualizing data and triggering alerts. Thing Speak can be used to monitor a wide variety of devices and sensors, including those used in home automation, environmental monitoring, and industrial applications[21].

Thing Speak communicates with connected devices using an API over the internet. This API allows devices to send data to Thing Speak, and for Thing Speak to send alerts back to devices. Thing Speak also provides a number of channels for sharing data with other users and applications. Thing Speak is a powerful tool for IoT developers and users. It can be used to create a wide variety of IoT applications, from simple data monitoring to complex data analysis.

Thing Speak is an IoT platform that enables the creation of sensor-based logging applications, location tracking applications, and a virtual "social network" of objects with real-time status updates. It also allows for remote control of internet-connected home automation devices.

The core element of Thing Speak is the "channel," which contains data fields for sensor readings, location information, and status updates. Thing Speak is an IoT analytics platform that enables you to collect, visualize, and analyze live data streams from your connected devices. It provides ready-to-use data visualizations without requiring any coding.

Thing Speak integrates with MATLAB analytics, allowing you to write and execute MATLAB code for more sophisticated data preprocessing, visualizations, and analyses. This eliminates the need to set up servers or develop web software, streamlining the process of building IoT systems[22].

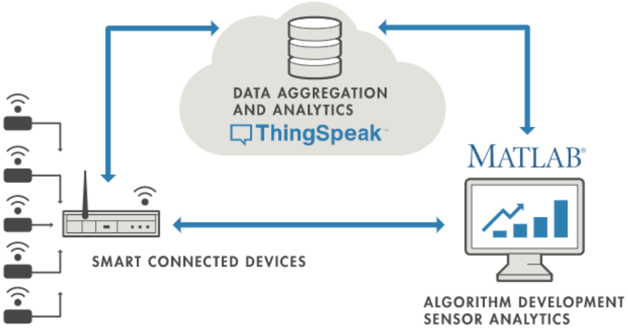


Figure 8. Thing speaks

Thing Speak Key Features

Thing Speak is an open-source IoT analytics platform designed to aggregate, visualize, and analyze live data streams from various devices and sensors[23]. It offers a suite of powerful features that cater to a wide range of IoT applications, enabling users to collect, process, and visualize data in real-time. Here are some of the key features of Thing Speak[24]:

- Data Collection: Thing Speak supports a variety of IoT protocols, including HTTP, MQTT, and TCP/IP, making it easy to connect and integrate various devices and sensors. This allows users to collect data from a wide range of sources, including temperature sensors, humidity sensors, air quality sensors, and more.
- Data Visualization: Thing Speak provides a variety of visualization tools to help users visualize their data in real-time. This includes charts, graphs, and maps, which can be customized to display the data in a way that is most meaningful to the user.
- Data Analysis: Thing Speak integrates with MATLAB, a powerful mathematical computing software, enabling users to perform advanced data analysis. This allows users to extract meaningful insights from their data, such as trends, patterns, and anomalies.

- Alerts and Notifications: Thing Speak can be configured to send alerts and notifications when certain conditions are met. This can be helpful for applications where it is critical to be aware of changes in the data, such as in the case of a gas leak or a sudden rise in temperature.
- Data Sharing: Thing Speak allows users to share their data with others publicly or privately. This can be useful for collaboration purposes or for sharing data with the public for research or educational purposes.
- Third-party Integrations: Thing Speak can be integrated with a variety of third-party services, such as social media platforms and weather data providers. This allows users to extend the functionality of Thing Speak and integrate it with their existing workflows.
- Open-source and Scalable: Thing Speak is an open-source platform, which means that it is freely available for anyone to use and modify. This makes it a cost-effective solution for both individuals and businesses. Additionally, Thing Speak is a scalable platform, which means that it can be used to collect and analyze data from a wide range of devices and sensors.

IV. RESULT AND DISCUSSION

This proposed system shows the how the gas leakage has been detected from the sensor. And show how fire detector from sensor. the data has been sent to the cloud (Speak Thing). Thing Speak can view the amount of leakage gas and temperature. In this system, we succeeded in preventing a disaster in the factory and preserving the safety of its workers.

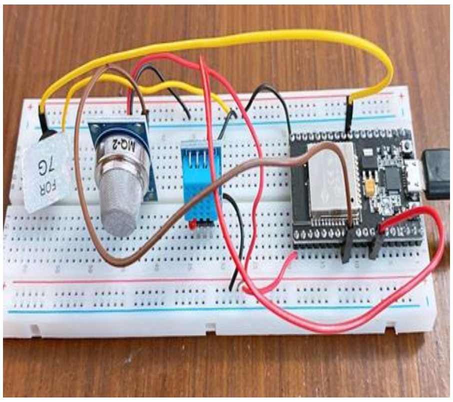


Figure 9. The experimental setup

Figures (10) and (111) show the graphic display of the data received from the two sensors. Each figure presents the data from a particular sensor, and each has two panels. Figure10 shows the screen shot from the monitoring server (laptop) after sent the data from the Esp32 and received by Thing Speak, which shows the reading of gas values each minute, and also the reading of the current gas value.

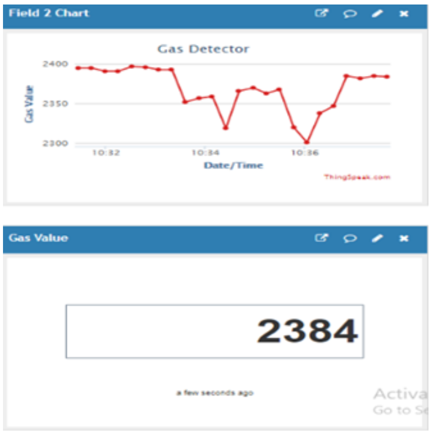


Figure 10. gas values

Figure11 shows the screen shot from the monitoring server (laptop) after sent the data from the Esp32 and received by Thing Speak, which shows the reading of temperature values each minute, and also the reading of the current temperature value.

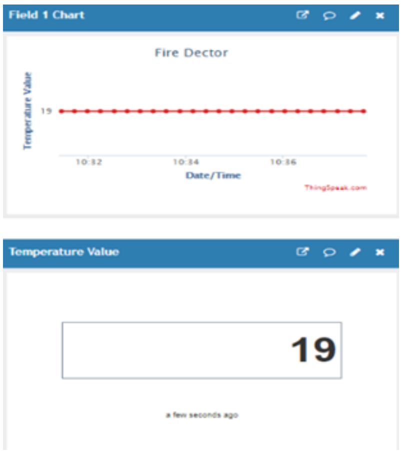


Figure 11. Temperature values

V. CONCLUSIONS

The proposed system makes the gas detection and its prevention easier of user, whether technically sound or not. It also alerts the user with the fire alarm system. The system won't solely offer safety to the users against harmful gases. However, this system provides alert information to the Gas or fire when gas Leak or High temperature occurs. It also gives an alert indication to buzzer.

This project is wirelessly transferred alert notification to the user and the user can easily connect the devices through the public cloud (Thing Speak) from any location. The proposed system is used the Esp32 microcontroller. When comparing to the other systems, it creates a less expensive and safety to the citizens also provide fast alerting compared to the other systems. Easy access and control make the system very useful. It can be monitored from anywhere, receive a warning, and avoid a disaster before it occurs.

References

1. Lombardi, M., Pascale, F., & Santaniello, D. (2021). Internet of things: A general overview between architectures, protocols and applications. Information, 12(2), 87.
2. Ghazal, T. M., Hasan, M. K., Alshurideh, M. T., Alzoubi, H. M., Ahmad, M., Akbar, S. S., ... & Akour, I. A. (2021). IoT for smart cities: Machine learning approaches in smart healthcare—A review. *Future Internet*, *13*(8), 218.
3. Stolojescu-Crisan, C., Crisan, C., & Butunoi, B. P. (2021). An IoT-based smart home automation system. *Sensors*, *21*(11), 3784.
4. GAURAV, B., SNEHAL, D., MOTE, K., & ZADBUKE, A. Intelligent Helmet Safety System for Mine Workers. *International Journal of Innovations in Engineering Research and Technology*, *4*(3), 1-5.
5. Mohammed, B. K., Mortatha, M. B., Abdalrada, A. S., & ALRikabi, H. T. S. (2021). A comprehensive system for detection of flammable and toxic gases using IoT. *Periodicals of Engineering and Natural Sciences*, *9*(2), 702-711.
6. Vales, V. B., Fernández, O. C., Domínguez-Bolaño, T., Escudero, C. J., & García-Naya, J. A. (2022). Fine time measurement for the Internet of things: a practical approach using ESP32. *IEEE Internet of Things Journal*, *9*(19), 18305-18318.
7. Ogunjirin, O. A., Adewunmi, A. A., Nwaobilor, P., Willoughby, F. A., & Ogini, F. U. (2021). Development of NCAM Temperature and Humidity Sensor. *International Journal of Power and Energy Engineering*, *3*(5).

8. Critcher, S., & Freeborn, T. J. (2021). Localized bioimpedance measurements with the max3000x integrated circuit: Characterization and demonstration. *Sensors*, *21*(9), 3013.

9. Ibrahim, M. N., & Francis, M. (2023). Design And Construction of an Automatic Shutdown Device Using Programmable PIC16F844A Micro-Controller. *Communication in Physical Sciences*, *9*(4).

10. Dhruv, A. J., Patel, R., & Doshi, N. (2021). Python: the most advanced programming language for computer science applications. *Science and Technology Publications, Lda*, 292-299.

11. Chromiński, K., Benko, Ľ., Hernández-Figueroa, Z. J., González-Domínguez, J. D., Rodríguez-del-Pino, J. C., & Přichystal, J. (2021). Python Fundamentals.

12. Deshmukh, R. R., Pawar, A., Katore, V., Dabhade, N., Kasture, V., & Bagul, S. (2021). Inventory Management System.

13. Munawar, K., & Naveed, M. S. (2022). The Impact of Language Syntax on the Complexity of Programs: A Case Study of Java and Python. *Int. J. Innov. Sci. Technol*, *4*, 683-695.

14. Lowther, D., Jacob, D., & Singer, J. (2023, October). Morello MicroPython: A Python Interpreter for CHERI. In *Proceedings of the 20th ACM SIGPLAN International Conference on Managed Programming Languages and Runtimes* (pp. 62-69).

15. Patkar, U., Singh, P., Panse, H., Bhavsar, S., & Pandey, C. (2022). Python for web development. *International Journal of Computer Science and Mobile Computing*, *11*(4), 36.

16. Clavié, B., & Soulié, G. (2023). Large Language Models as Batteries-Included Zero-Shot ESCO Skills Matchers. *arXiv preprint arXiv:2307.03539*.

17. Gad, A. F. (2021). Pygad: An intuitive genetic algorithm python library. *arXiv preprint arXiv:2106.06158*.

18. Lasser, J., Manik, D., Silbersdorff, A., Säfken, B., & Kneib, T. (2021). Introductory data science across disciplines, using Python, case studies, and industry consulting projects. *Teaching Statistics*, *43*, S190-S200.

19. Cabanero-Gomez, L., Hervas, R., Gonzalez, I., & Rodriguez-Benitez, L. (2021). eeglib: a Python module for EEG feature extraction. *SoftwareX*, *15*, 100745.

20. De Nardis, L., Mohammadpour, A., Caso, G., Ali, U., & Di Benedetto, M. G. (2022). Internet of things platforms for academic Research and Development: A critical review. *Applied Sciences*, *12*(4), 2172.

21. Oturak, M., & Dursun, E. (2021, August). A cost-effective IoT based smart home application. In *2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)* (pp. 1-6). IEEE.

22. Singh, R. R., Banerjee, S., Manikandan, R., Kotecha, K., Indragandhi, V., & Vairavasundaram, S. (2022). Intelligent IoT wind emulation system based on real-time data fetching approach. IEEE Access, 10, 78253-78267.

23. Pimprale, V., Arora, S., & Deshmukh, N. IoT Cloud Platforms: A Case Study in ThingSpeak IoT Platform. In Integration of Cloud Computing with Emerging Technologies (pp. 182-198). CRC Press.

24. Antonova, V. M., Malikova, E. E., Panov, A. E., Spichek, I. V., & Malikov, A. Y. (2021). Implementation of IoT technology for data monitoring via cloud services. T-Comm-Телекоммуникации и Транспорт, 15(2), 46-53.