ICIC Express Letters
Volume **15**, Number **10**, October **2021**

ICIC International ©2021 ISSN 1881-803X
pp. 1101–1108

FIRE EARLY DETECTION IOT-BASED USING NODEMCU ESP8266 AND THINGER.IO WEB SERVER

Ardi Rahmawan, Suyoto* and Djoko Budiyanto Setyohadi

Informatics Department
Universitas Atma Jaya Yogyakarta
Jl. Babarsari 44, Yogyakarta 55281, Indonesia
ardi.rahmawan2019@gmail.com; djoko.budiyanto@uajy.ac.id
*Corresponding author: suyoto@staff.uajy.ac.id

Received December 2020; accepted March 2021

Abstract. *Fires are very detrimental, ranging from loss of property to even casualties. Inflorescence can be seen when the fire has grown, or smoke has risen from the building. Therefore, we need a tool that can detect an early fire source's presence not to spread. This study aims to design and build an efficient and affordable device, which can be applied to housing or warehouse. This tool is intended to use NodeMCU ESP8266 as the central controller. A flame sensor can respond to infrared light emission in the modulation spectrum from 5 to 30 cycles per second, Thinger.Io Web as a server, and data storage on the Internet of Things. The experimental results show that when the fire sensor reads the fire wave, the system will send a message and activate an alarm at the scene. This research produces a tool that can detect the early presence that can notify the user's mobile device to minimize the spread of a more massive fire.*
Keywords: Microcontroller, NodeMCU ESP8266, Thinger.Io, Internet of Things

1. **Introduction.** Fire events can occur in public or residential areas, which can harm humans [1]. Several causes can cause the incidence of fire, including a short circuit (short circuit) [2], leakage of LPG (Liquid Petroleum Gas) pipes, stove fire, garbage burning, and cigarette fire. Fires can also arise due to natural and weather factors such as lightning, earthquakes, volcanic eruptions, and drought [3]. A fire can be detected when the fire has grown, and black smoke comes out of the building.

In general, a fire can be concluded as an uncontrolled fire incidence or event that can endanger life and property [4]. Therefore, it is necessary to have the technology to minimize fires by implementing a device that can detect flames on the NodeMCU ESP8266 device. NodeMCU is a development device capable of connecting to WiFi [5]. The device can transmit data wirelessly (warless), connected to the Thinger.Io Web server to send information or notifications of fire waves quickly to the user's smartphone.

Internet of Things (IoT) technology is a network device that can communicate with each other using IP connectivity without human intervention [6]. Internet of Things (IoT) is widely used as one of the potential technologies that can be used with everyday devices in networks equipped with colossal levels of intelligence [7,8]. In recent times, the IoT has been increasing. Mostly, billions of IoT devices (ranging from tiny sensor units to complex interactive systems) are connected to the transmission network through widely deployed wireless access technologies [9]. Almost all electronic devices can be controlled by mobile applications associated with the Internet using IoT [10]. IoT is gradually gaining an essential war in several research fields [11], which can be used to simplify and increase the time for each job and business in various aspects of life, such as smart homes, smart cars,

DOI: 10.24507/icicel.15.10.1101

agriculture, transportation, and health care [9,12]. Thus, as these technologies advance, consumers' behavior in utilizing and seeking information about them is also changing [13].

This research aims to develop a tool that can detect the early presence of fire. The system prevents the spread of a larger fire by using NodeMCU ESP8266 as the central controller, flame sensor to catch fire, buzzer as a warning alarm if the sensor detects a fire wave, the LED functions. The WiFi network is connected to the microcontroller and Microsoft Outlook as a notification sender to its mobile device.

The remainder of this paper is organized as follows. Section 2 discusses the literature review. In Section 3, we present the proposed method. Section 4 offers the testing with results and discussion. Finally, we conclude the paper in Section 5.

2. **Literature Review.** IoT is a series of machines connected between mechanical machines and digital machines, which can transfer data via computer network devices. In the agricultural sector, IoT can perform data analysis, predict weather conditions, and increase productivity, expenditure, and resources. Whereas in smart transportation, IoT provides innovative services that allow various users to get better, safer, more codified, and more intelligent information [14].

The Thinger.Io Web server is the IoT cloud platform chosen for data transmission in making this API early detection tool because it supports all types of boards such as Arduino, ESP8266, Raspberry Pi, and Intel Edison [13,15]. Thinger.Io has also provided every tool needed to prototype, scale, and manage products. Thinger.Io's goal is to demonstrate IoT users so that users worldwide can access them; besides that Thinger.Io provides a freemium account, Thinger.Io has been designed to allow connectivity to many devices open-source nature [16].

Several studies have developed Internet of Things (IoT) technology that applies fire sensors, such as research from Muheden et al. [14], Sarkar et al. [17], Ralevski and Stojkoska [10], Mahgoub et al. [18] and Salam et al. [19].

The next research uses Arduino technology for fire detection (fire detector) systems by Halim et al., Hyeong-Su et al. and Salam et al. The system is designed to provide early warning in case of fire. This system works using a fire sensor system and a temperature sensor connected to Arduino Uno. and an Ethernet shield connected to Internet network [19-21].

Prayogo et al. discussed designing a sensor-based home security device and a panic button as an emergency alarm using NodeMCU ESP8266 [22]. Megantoro and Winarno used the NodeMCU ESP8266 board as a client and Raspberry Pi 3 as a server. The proposed emergency call auto-register is an emergency warning system based on IoT for intensive care patients at the hospital [23].

In another study, Kharade et al., discussed the design of an IoT, fire safety, NodeMCU ESP8266, and flame sensor [24]. For fire safety testing, a burning flame, when brought near up to a maximum range of 100 cm, detected the fire and notified over the Internet and the fire alarm responded. The device as a whole performed well, also in case of fire, the flame sensor would detect and ring the alarm interfaced on the relay channel.

Referring to the case described by previous researchers, no system has ever been designed that has implemented Thinger.Io Web as a monitoring, data storage, and notification system. Researchers intend to develop innovation, namely the application of Thinger.Io Web on the NodeMCU ESP8266 fire early detection sensor based on the Internet of Things (IoT), which can send fire hazard messages to mobile users via the Microsoft Outlook application.

3. **Proposed Method.** The system aims to detect the fire's early presence to avoid a giant fire. We proposed using the Thinger.Io Web, a buzzer for the flame sensor, and an LED. The LED will turn on when connected to the Internet and automatically turn off

when the WiFi network is cut off. If there is a fire, the system will send a notification to the user's smartphone.

This system requires a WiFi connection to carry out its functions and always be connected to a power check or power supply, and the user's smartphone must have Microsoft Outlook installed.

Figure 1 shows the research flowchart. There are five steps: data collection, system design, coding, testing, and implementation.

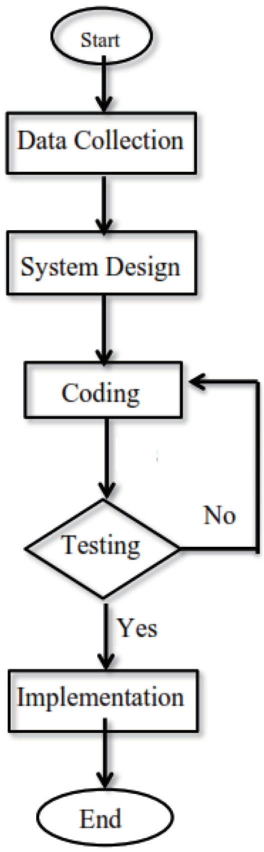

Figure 1. Research flowchart

Data collection is to collect data for the system. Furthermore, for system design, we design for microcontroller and Thinger.Io Web Library, also system workflow.

The microcontroller system is designed to detect fires early using the Fritzing software. This system is used to avoid breaking the equipment circuit directly when experimenting. This tool connects several components using a jumper cable in the manufacturing series, including NodeMCU to the beard board, VCC NodeMCU connected to the negative board, GND NodeMCU to the positive board, and 3V NodeMCU to the positive board. Next, connect NodeMCU to the flame sensor, D1 NodeMCU is connected to the DO flame sensor. Next, connect NodeMCU with buzzer, D4 NodeMCU to the positive buzzer. The next step is connecting NodeMCU to the LED. D2 NodeMCU is connected to the positive LED. Then the LED to the beard load, the LED's negative pole is connected to the positive board. Next, connect NodeMCU to Buzzer, and NodeMCU D4 to positive Buzzer. After that, make connecting the NodeMCU to the LED, NodeMCU D2 to the positive pole of the LED, and the negative pole of the NodeMCU to the positive LED. Finally, connect the NodeMCU to RELAY, GND to IN, VIN to GND, and D7 to VCC Relay.

Next, Thinger.Io Web Library Design contains Thinger.Io Web registration and dashboard planning.

3.1. **Thinger.Io Web registration.** Following are the steps for signing up for Thinger.Io Web. In the first step, the user registers on the website https://thinger.io, using the registered email.

When the user has successfully logged in, add the device and dashboard on the left side menu. These "device" and "dashboard" are useful for NodeMCU to communicate with Thinger.Io. After the device and dashboard are made, a display will appear so that users fill in detailed data. In the third step, the user fills in the data and adjusts it to the program code that has been made previously. This adjustment starts from the device id, device description, and device credential. These three adjustments are useful for NodeMCU to communicate with Thinger.Io.

3.2. **Dashboard planning.** The dashboard serves to display real-time information from NodeMCU devices and functions to get historical information stored in data baskets sent periodically. The data source for each dashboard widget can be configured independently for devices connected to the Thinger.Io platform.

Finally, the system workflow is the flame sensor that detects the presence of fire. Furthermore, the microcontroller carries out the data collection process before sending it to Thinger.Io Web. The data is 1 and 0. If the sensor data is 1, it means that there is no fire or safe. Conversely, if the fire sensor value is 0, there is a fire or danger status. If the sensor detects a fire hazard, the microcontroller sends a danger status to Thinger.Io Web, and the buzzer alarm will turn on. Thinger.Io Web sends a notification of a detected fire to the user's mobile device via the Microsoft Outlook application.

4. **Results and Discussion.** The testing process is carried out to check each hardware and software component to determine which tools are designed to run correctly. The examination is carried out by conducting experiments to see possible errors that occur in each process.

a) Flame sensor testing

Initial testing is carried out on the flame sensor to determine the sensor's performance in capturing fire waves. Testing can be seen in Figure 2.

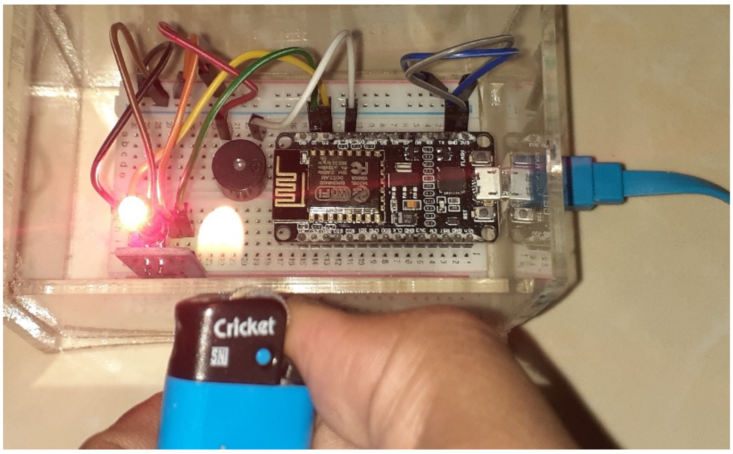

FIGURE 2. Flame sensor testing

This test aims to test the fire detector's flame sensor device displayed on the Arduino IDE monitor serial. The reading system of the fire sensor value only has a digital value of 0 and 1. If the sensor is 0, it is indicated that the sensor reads a fire, and if it is 1, the sensor does not read the presence of fire. The sensor value reading system stops every 3 seconds.

b) Thinger.Io Web testing

In Figure 3, you can see the display of Thinger.Io connected to NodeMCU via a different WiFi network. Next, Thinger.Io waits for the data sent by the NodeMCU flame sensor. Then send a notification to the user's smartphone device.

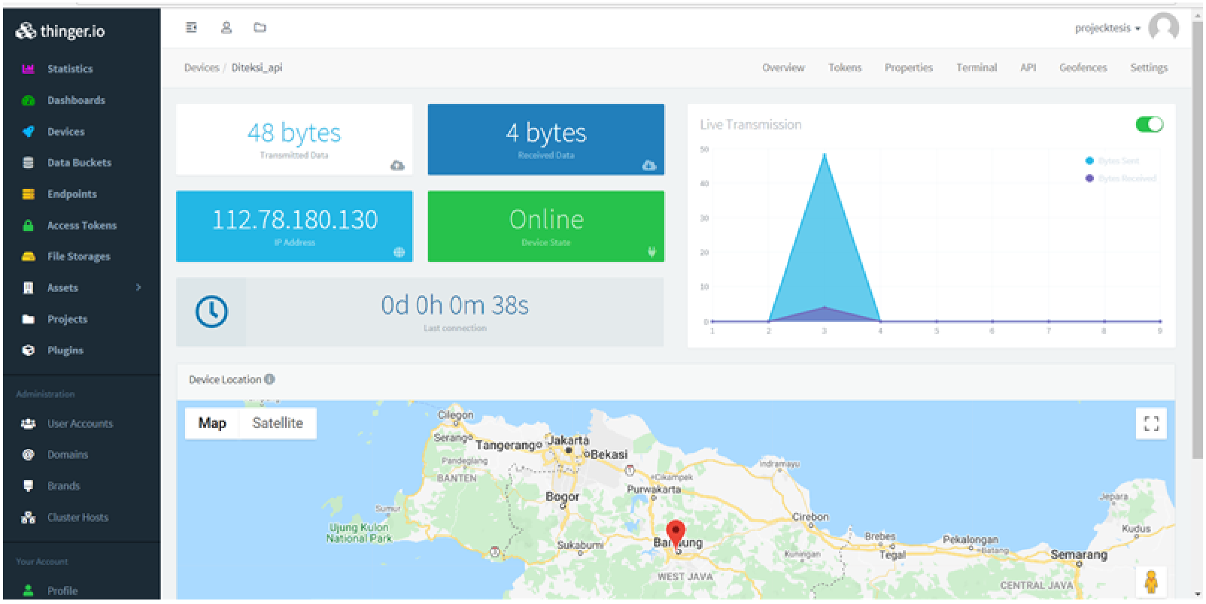

FIGURE 3. Thinger.Io Web interface connected with NodeMCU

c) Smartphone testing

In Figure 4, you can see the results of a fire detection warning notification sent by Thinger.Io Web to a mobile device. The information sent can be accessed anywhere as long as the user is always connected to the Internet network.

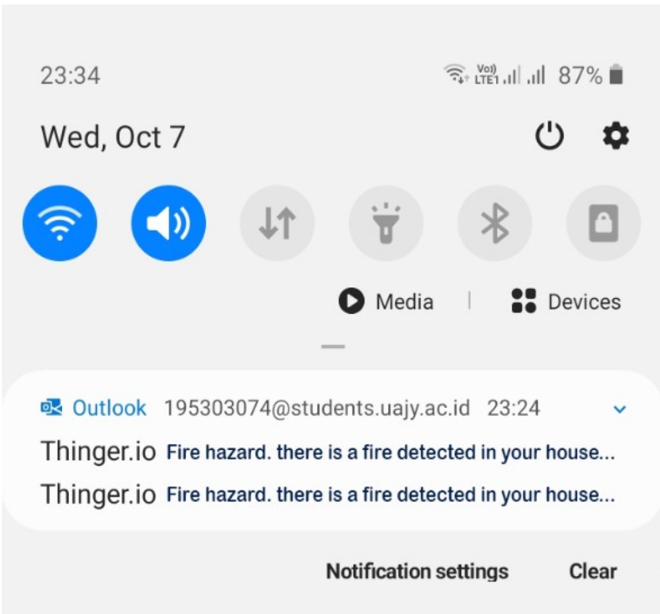

FIGURE 4. Smartphone notification testing

d) Testing tools as a whole

This test is carried out in seven stages to determine the accuracy of the fire detection distance on the microcontroller to determine how fast the sensor sends the data to the Thinger.Io Web. In the test, the area has been designed in the form of a prototype. The

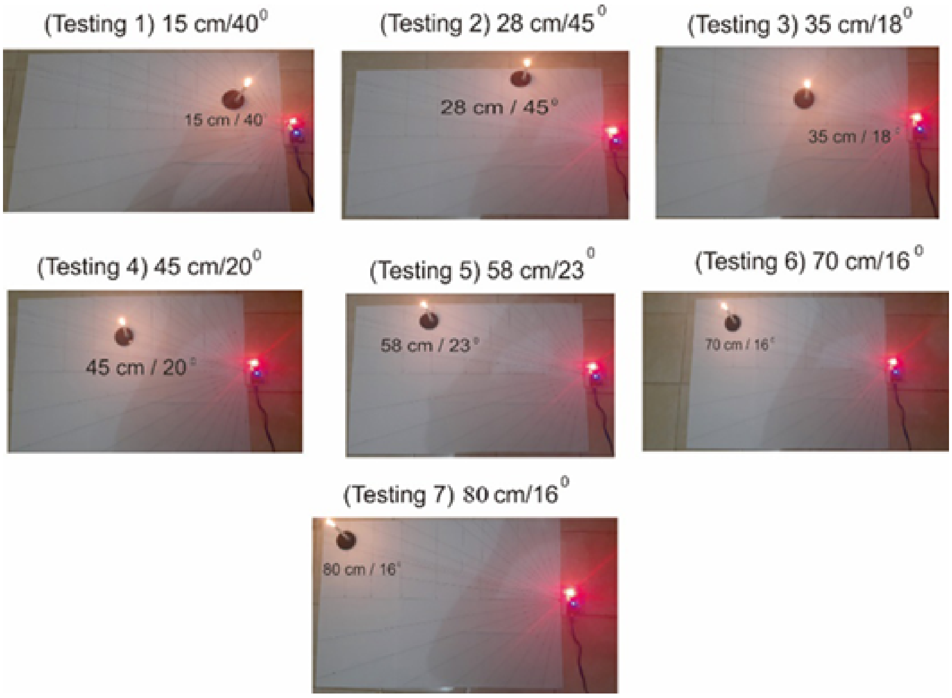

Figure 5. The testing tool as a whole

Table 1. Flame sensor testing to detect fire

Fire sensor testing (Flame sensor)					
No	Distance	Corner	Access time (seconds)	Send to Thinger.Io	Information
1	15 cm	40°	±2	Detected	Right
2	28 cm	45°	±2	Detected	Right
3	35 cm	18°	±2	Detected	Right
4	45 cm	20°	±2	Detected	Right
5	58 cm	23°	±2	Detected	Right
6	70 cm	16°	±2-3	Detected	Delay
7	80 cm	16°	±2-3	Detected	Delay

prototype dimension is 80 cm long and 75 cm wide with an angle of 800. In this study, the authors used a candle flame as an experimental medium. The time to transfer data to the server takes the range of ±2 to 3 seconds. Testing tools can be seen in Figure 5.

The results of testing the time of sending data by Table 1 shows flame sensor testing to detect fire, and Figure 6 shows the fire sensor testing graph. It is proven that the system can detect fires in open spaces with a distance of 15 to 80 cm and an angle of 16° to 40°.

The first test is carried out at a distance of 15 cm with an angle of 40° from the fire source to the flame sensor, and data transmission by the microcontroller to the Thinger.Io Web can travel well with a duration of 2 seconds of delivery time.

The second test is carried out with a distance of 28 cm with an angle of 45° from the fire source to the flame sensor. The process of sending data from the flame sensor to the Thinger.Io Web is still the same as the first test. It can run well with a duration of 2 seconds of delivery.

The third test is tested with a distance of 35 cm with an angle of 18° from the fire source to the flame sensor which is still the same as the first and second tests. The process of sending data to the Thinger.Io Web can run well with 2 seconds of delivery time.

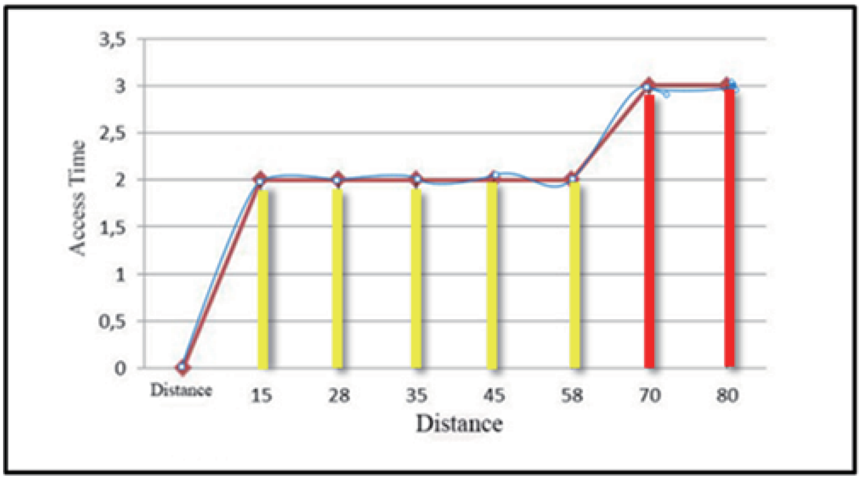

Figure 6. Fire sensor testing graph

The fourth test with a distance of 45 cm and an angle of 20° of the sending process data to Thinger.Io can run well with a duration of 2 seconds of delivery.

In the fifth test, with a distance of 58 cm and an angle of 23° from the fire source to the flame sensor, the process of sending data to the Thinger.Io Web is still running well with a duration of 2 seconds of delivery time.

In the sixth test, with a distance of 70 cm and an angle of 16° from the fire source to the flame sensor, experience a delay of 1 second in sending data to Thinger.Io Web with a total duration of 3 seconds of sending.

The last test is performed at a distance of 80 cm and an angle of 16°. The result shows that sending data to the Thinger.Io Web is similar to the previous sixth test. The data transmission is delayed by 1 second, and the total transmission time is 3 seconds. Thus, the situation occurs since the distance between the fire source and the flame sensor is far, so the data transmission process cannot run properly.

5. **Conclusion.** Based on the results of the design, implementation, and testing that has been done, conclusions can be drawn. The NodeMCU ESP8266 fire early detection system and the application of Thinger.Io Web as a notification media based on the Internet of Things (IoT) have been successfully created, which can detect the presence of fire early to minimize the risk of a more significant fire. Each status will be recorded and sent by the microcontroller, which is integrated into Thinger.Io Web. Another use of Thinger.Io is that it will send a notification as fire hazard text to the smartphone. The house or warehouse condition can be monitored even though the owner is not at the building.

Acknowledgments. The authors acknowledge the support of Universitas Atma Jaya Yogyakarta in publishing this article.

REFERENCES

[1] O. Al-Mahmud, K. Khan, R. Roy and F. M. Alamgir, Internet of Things (IoT) based smart health care medical box for elderly people, *Int. Conf. Emerg. Technol. (INCET2020)*, pp.1-6, DOI: 10.1109/INCET49848.2020.9153994, 2020.

[2] H. Garg and M. Dave, Securing IoT devices and SecurelyConnecting the dots using REST API and middleware, *The 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU2019)*, pp.1-6, DOI: 10.1109/IoT-SIU.2019.8777334, 2019.

[3] S. Sachdev, J. MacWan, C. Patel and N. Doshi, Voice-controlled autonomous vehicle using IoT, *Procedia Comput. Sci.*, vol.160, pp.712-717, DOI: 10.1016/j.procs.2019.11.022, 2019.

[4] A. Kaur, S. S. Saini, L. Singh, A. Sharma and E. Sidhu, Efficient Arduino UNO driven smart highway/bridge/tunnel lighting system employing rochelle piezoelectric sensor, *The 2nd IEEE Int. Conf. Control Comput. Commun. Mater.*, pp.9-12, DOI: 10.1109/ICCCCM.2016.7918247, 2016.

[5] L. K. P. Saputra and Y. Lukito, Implementation of air conditioning control system using REST protocol based on NodeMCU ESP8266, *Int. Conf. Smart Cities, Autom. Intell. Comput. Syst. (ICON-SONICS2017)*, pp.126-130, DOI: 10.1109/ICON-SONICS.2017.8267834, 2017.
[6] J. Mocnej et al., Quality-enabled decentralized IoT architecture with efficient resources utilization, *Robot. Comput. Integr. Manuf.*, vol.67, DOI: 10.1016/j.rcim.2020.102001, 2020.
[7] S. S. Prayogo, Y. Mukhlis and B. K. Yakti, The use and performance of MQTT and CoAP as Internet of Things application protocol using NodeMCU ESP8266, DOI: 10.1109/ICIC47613.2019.8985850, 2019.
[8] B. Miles, E. B. Bourennane, S. Boucherkha and S. Chikhi, A study of LoRaWAN protocol performance for IoT applications in smart agriculture, *Comput. Commun.*, vol.164, pp.148-157, DOI: 10.1016/j.comcom.2020.10.009, 2020.
[9] Y. Duan, H. Ni, X. Zhu and X. Wang, Reliable broadcast for WiFi-based Internet of Things through multi-access edge-dynamic FEC adaptation, *International Journal of Innovative Computing, Information and Control*, vol.16, no.5, pp.1739-1755, 2020.
[10] M. Ralevski and B. R. Stojkoska, IoT based system for detection of gas leakage and house fire in smart kitchen environments, *The 27th Telecommun. Forum (TELFOR2019)*, pp.16-19, DOI: 10.1109/TELFOR48224.2019.8971021, 2019.
[11] R. K. Kodali and V. S. K. Gorantla, RESTful motion detection and notification using IoT, *Int. Conf. Comput. Commun. Informatics (ICCCI2018)*, pp.1-5, DOI: 10.1109/ICCCI.2018.8441423, 2018.
[12] Z. Jian, L. Yang, X. Xu, X. Huai, Y. Di and Y. Zhao, Analysis of temporal-spatial characteristics of wildfire in Hunan province during Qingming Festival, *The 3rd IEEE Conf. Energy Internet Energy Syst. Integr. Ubiquitous Energy Netw. Connect. Everything (EI2 2019)*, pp.842-845, DOI: 10.1109/EI247390.2019.9061840, 2019.
[13] S. N. N. Htun, T. T. Zin and P. Tin, Consumer behavior analyzer in Internet of Things (IoT) environments, *International Journal of Innovative Computing, Information and Control*, vol.17, no.1, pp.345-353, 2021.
[14] K. Muheden, E. Erdem and S. Vançin, Design and implementation of the mobile fire alarm system using wireless sensor networks, *The 17th International Symposium on Computational Intelligence and Informatics (CINTI)*, pp.243-246, DOI: 10.1109/CINTI.2016.7846411, 2016.
[15] Z. Zhang and C. Williamson, A campus-level view of outlook email traffic, *ACM Int. Conf. Proceeding Ser.*, pp.299-306, DOI: 10.1145/3301326.3301371, 2018.
[16] M. Fezari and A. Al Dahoud, *Integrated Development Environment 'IDE' for Arduino*, Amman, Jordan, https://www.researchgate.net/publication/328615543, 2018.
[17] S. Sarkar, S. Gayen and S. Bilgaiyan, Android based home security systems using Internet of Things (IoT) and firebase, *International Conference on Inventive Research in Computing Applications (ICI RCA2018)*, pp.102-105, DOI: 10.1109/ICIRCA.2018.8597197, 2018.
[18] A. Mahgoub, N. Tarrad, R. Elsherif, A. Al-Ali and L. Ismail, IoT-based fire alarm system, *The 3rd World Conference on Smart Trends in Systems Security and Sustainablity (WorldS4)*, pp.162-166, DOI: 10.1109/WorldS4.2019.8904001, 2019.
[19] A. E. U. Salam, R. S. Sadjad, F. A. Samman, I. R. Sahali and Zulharman, IoT-based fire and theft detection system in housing, *ICIC Express Letters*, vol.14, no.9, pp.917-925, 2020.
[20] D. K. Halim, T. C. Ming, N. M. Song and D. Hartono, Arduino-based IDE for embedded multiprocessor, *The 5th International Conference on New Media Studies (CONMEDIA)*, pp.135-138, DOI: 10.1109/CONMEDIA46929.2019.8981862, 2019.
[21] K. Hyeong-Su, K. Jin-Woo, S. Yun and W. T. Kim, A novel wildfire digital-twin framework using interactive wildfire spread simulator, *Int. Conf. Ubiquitous Futur. Networks (ICUFN)*, pp.636-638, DOI: 10.1109/ICUFN.2019.8806107, 2019.
[22] S. S. Prayogo, F. Al Rafi and Y. Mukhlis, Design and built IoT home panic button for smart city, *IOP Conf. Series: Journal of Physics: Conf.*, DOI: 10.1088/1742-6596/1175/1/012097, 2019.
[23] P. Megantoro and H. A. Winarno, EKA v1: Emergency call auto-register, an emergency warning system based on Internet of Things for intensive care patient at hospital, *IOP Conference Series: Materials Science and Engineering*, vol.835, no.1, DOI: 10.1088/1757-899X/835/1/012033, 2020.
[24] M. Kharade, S. Katangle, G. M. Kale, S. B. Deosarkar and S. L. Nalbalwar, A NodeMCU based fire safety and air quality monitoring device, *Int. Conf. Emerg. Technol. (INCET2020)*, pp.3-6, DOI: 10.1109/INCET49848.2020.9153983, 2020.