IEC
Institute of Electronics and Computer

Journal of the Institute of Electronics and Computer, 2023, 5, 1-15
https://iecscience.org/journals/JIEC
ISSN Online: 2643-8240

IoT Based Fire Extinguisher Obstacle Detection for Industrial Safety

Sriram Baradwaj B, Vikram Raj N, Dr. G. Kannan*, Ms K Indra Gandhi

Department of Electronics and Communication Engineering, B.S Abdur Rahman Crescent Institute of Science and Technology, Vandalur, Tamil Nadu 600048, India
* Corresponding Author: Dr. G. Kannan, Email: kannan@crescent.education

How to cite this paper: Sriram Baradwaj B, Vikram Raj N, Dr. G. Kannan, Ms K Indra Gandhi (2023). IoT Based Fire Extinguisher Obstacle Detection for Industrial Safety. Journal of the Institute of Electronics and Computer, 5, 1-15. https://doi.org/10.33969/JIEC.2023.51001.

Received: February 3, 2023
Accepted: March 2, 2023
Published: March 15, 2023

Abstract

Fire extinguishers are an integral part of the safety measures of any building, mainly in industries and should be accessible at all times. They are an important part of the safety equipment used to prevent a fire accident. Fire accidents in industries may result in high valuable equipment being destroyed, chances of injuries to the employees. It may lead to temporary or permanent closing of the industries. Since there are thousands of fire extinguishers may be installed in the large scale industry, it is hard to keep track of them manually whether each fire extinguisher can be easily accessed during the fire accident. This paper proposes IoT based device to monitor the status of fire extinguisher with the help of array of sensors. The fire extinguisher is considered to be blocked if an obstacle is found less than 60 cm from the fire extinguisher. When an obstacle is found, it will give an alarm for localized alert. Every fire extinguisher in the industry is attached with this device, each device will have a unique Node ID. Through Internet of Things technology and with the help of unique Node IDs, globalized alert is also sent. To send data through IOT a localized server is created where all the unique Node IDs are connected to it. The power management strategy is also analyzed to enhance the device life time.

Keywords

Industrial IoT, Fire Extinguisher, Sensor, Safety, Embedded Systems.

1. Introduction

In today's world the use of fire extinguishers in case of a fire accident is going on increasing. Especially in industries like: Manufacturing Industry, Hazardous Industry, Bio-Medical Industry, Power Plants, Oil Industry, Hospitals Petrol Bunk, and Gas Industry.An industrial fire is a type of industrial disaster that occurs when a conflagration occurs in a production setting. Explosions are commonly, but not always, related with industrial fires. They're particularly likely to occur in areas with a lot of flammable stuff. Natural gas, petroleum, and petroleum compounds such as petrochemicals are examples of such materials.When combustible materials, such as hydrocarbons, are treated in units at high temperatures and/or pressures, the hazards are amplified. Combustible substances can be found in oil refineries, tank farms (oil depots), natural gas processing plants, and chemical industries, particularly petrochemical factories. Such structures usually have their own fire departments for firefighting. Dust and powder can be flammable, and when ignited, they can cause dust explosions. Severe industrial fires have resulted in multiple injuries, deaths, major financial loss, and/or harm to the surrounding population or environment.

1.1 Industrial Internet of Things

The term "Industrial Internet of Things" was coined to describe the Internet of Things (IOT) as it is used in a variety of industries, including manufacturing (Industry 4.0), logistics, oil and gas, transportation, energy/utilities, mining and metals, aviation, and other industrial sectors, as well as industry-specific use cases. The internet of things, or IOT, is an interconnected network of computing devices, mechanical and digital machinery, goods, animals, and people with unique identifiers (UIDs) and the ability to transfer data without requiring human-to-human or human-to-computer interaction.Industry 4.0 refers to the intelligent networking of equipment and processes for industry using information and communication technology. Industry 4.0 is defined as the generation, use, and exploitation of actionable data and information as a means of realising smart industries and ecosystems of industrial innovation and collaboration in a connected environment of big data, people, processes, services, systems, and IoT-enabled industrial assets. Long-range Wi-Fi networks were utilised to connect the devices to a server, ensuring that they were always online. As a result, this project complies with Industry 4.0 guidelines.

1.2 Problem Statement

In case of a fire accident, the fire extinguishers should be in a place that is easy to use and is easily accessible. For fire extinguishers to be accessible at all times there should be no obstacles blocking around it. As there are thousands of fire extinguishers in industries, monitoring each fire extinguisher manually is a tedious. The aim is to design a device capable of monitoring the surroundings of a fire extinguisher 24/7 at the same time the device should be compact, cost effective and power efficient. IIOT or Industrial Internet of Things can be used to monitor the fire extinguishers i.e., Automate the process.

2. Design and Development

The main objective of this project is to check the accessibility of fire extinguishers in Industries. It is a tedious task to keep every fire extinguisher in check. This project automates this process by constantly checking the area around the fire extinguishers and updating the status of the fire extinguishers in a server.

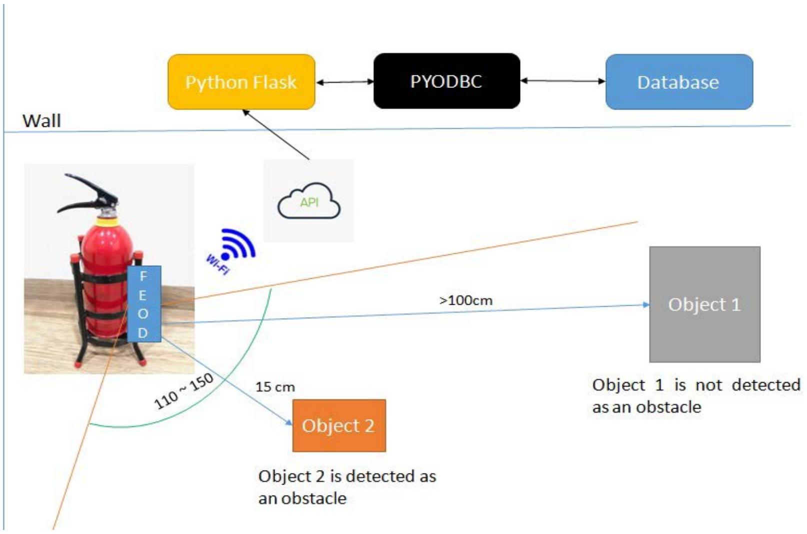

Figure 1 Proposed System Architecture

The device developed is a compact solution which acts as a standalone and an industrial 4.0 solution using Industrial Internet of Things (IIOT). The device is capable of monitoring the fire extinguisher 24/7. The device consumes less power and at the same time is cost efficient. The device detects any blockage / Obstacle in front of the fire extinguisher. A database has been to monitor the accessibility of the fire extinguisher. The device updates in the database when an obstacle is detected / when the fire extinguisher is not accessible.

Figure 1 represents a visualized solution of the device and its working. The Fire Extinguisher Obstacle Detection (FEOD) device majorly consists of PIR sensor, Ultrasonic sensor, Servo Motor, Node MCU, Buzzer and LED. PIR sensor and ultrasonic sensor are used to get input from the surroundings and servo motor helps ultrasonic sensor in this process. Then the data collected from these sensors are transmitted through Wi-Fi to the database. The given Threshold distance as 60cm. Here obstacle 1 is more than 100 cm away from the fire extinguisher. Hence it will be in a position such that it will be not blocking the view or accessibility of the fire extinguisher. But obstacle 2 is at a distance of 15 cm from the fire extinguisher which is placed at a distance that has high possibility of blocking the view or accessibility of the fire extinguisher. The status of the accessibility of the fire extinguisher will be updated in the database through Python flask and PYODBC frameworks for globalized alerts. LED and Buzzer are used for localized alerts.

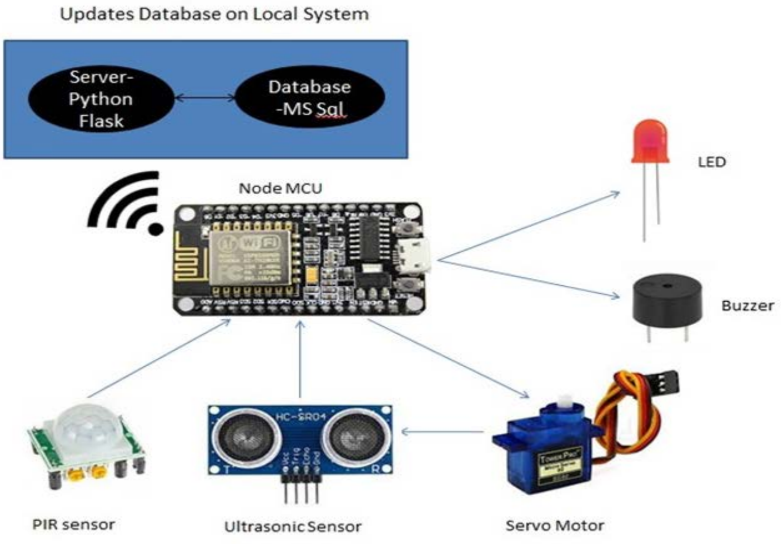

Figure 2 System Block diagram

It will be monitoring for any obstacle around the fire extinguisher in real time and also updating the accessibility of it, in a server. The PIR sensor detects for any obstacle brought into the field of view of the fire extinguisher and the servo motor aids the ultrasonic sensor to scan the area in front of the fire extinguisher for a specific angle and distance such that the obstacle clearly obstructs the field of view of the fire extinguisher. When it detects an obstacle, the alarm gets triggered and the LED is turned ON. The data sent through the http GET request with the help of a custom made API through Wi-Fi is retrieved using python flask. These data are set to a variable then a connection is requested to MS SQL with the help of PYODBC, when a connection is established the data in the variables are sent to the database and is committed to their respective tables.

Table 1 NODE MCU Connections

Component	Node MCU Port
Ultrasonic sensor trigger pin	GPIO 14 pin
Ultrasonic sensor echo pin	GPIO 12 pin
PIR sensor output pin	GPIO 2 pin
Servo motor PWM signal pin	GPIO 5 pin
LED	GPIO 4 pin
Buzzer	GPIO 4 pin
VCC	3.3v pin
Ground	Ground

2.1. FEOD

The device "Fire Extinguisher Obstacle detection" (FEOD) uses Ultrasonic sensor for detecting obstacles. The servo motor helps the ultrasonic sensor to scan the area in front of the fire extinguisher where obstacles can possibly obstruct the field of view of the fire extinguisher. The ultrasonic sensor and the servo motor create a scanning mechanism and scans for obstacles blocking the accessibility of the fire extinguisher. This scanning mechanism is established by mounting the ultrasonic sensor over the servo motor with the help of an ultrasonic sensor holder. This makes the ultrasonic sensor to move in the direction the servo motor moves. The servo motor is used to move the ultrasonic sensor for an angle of 0° to 110°. The ultrasonic sensor will notify obstacles which are in a range of 60 cm.

When the device (FEOD) detects an obstacle in a range of 60 cm for an angle of 0° to 110° in front of the fire extinguisher, it observes that obstacle for 10 seconds by stopping the motion of servo motor and being still in the direction where the obstacle was identified. If the obstacle is removed before 10 seconds then it will not notify and it will continue the scan.In case an obstacle is not removed before 10 seconds it will trigger the alarm. The alarm consists of a buzzer and a LED. The buzzer is set off and the LED will continue to glow till the obstacle is cleared. In order to reduce the frequency of unnecessary scans a triggering mechanism is required to initiate the scanning mechanism.

Therefore the scanning mechanism is triggered with the help of a PIR sensor. PIR sensor checks for any movement in front of the fire extinguisher. In case of a movement it will trigger the scanning mechanism which consists of ultrasonic sensor and servo motor. When a motion is detected, PIR sensor sends out a pulse which acts like a trigger to start the scanning mechanism.The reason for using PIR sensor as a trigger mechanism to initiate the scanning mechanism of ultrasonic sensor and servo motor is to reduce the power consumption. When the scanning mechanism alone is used, it needs to scan for obstacles all the time. The servo motor consists of mechanical motion which consumes more power and also requires ample amount of power in order to maintain the speed of the servo motor. So when this scanning mechanism is ON 24*7 AC power supply will be required to run the device. Fire extinguishers will be located in nooks and corners where there will be no source of AC power supply to connect this device. In order to make this device work remotely with a DC power supply we need to optimize the device so that it can function using DC power supply for a minimum of 30 days. Hence PIR sensor is used, which uses only 65 mA as a trigger mechanism. This helps in prolonging the battery life of the device.

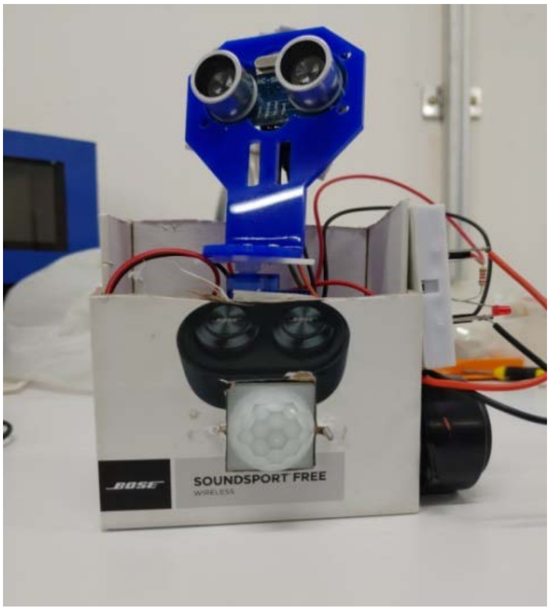

Figure 3 FEOD

To send data globally a server is setup with the help of python flask and Node MCU is connected to the server through Wi-Fi. Data is sent to the server by using HTTP requests which will contain the sensor values. After successful connection to Wi-Fi, data has to be sent through a custom made API whenever an obstacle is detected and to send the data when the obstacle is cleared after the alarm is triggered. Hence to do this a custom API is required. The Custom API used in this device is given below,

http://<IP V4 of the system> /<Location>/<Detection value>

Ipv4 of the system can be found by typing "ipconfig" in the command window. Location for the fire extinguisher is hard coded for each device. The detection value is updated in run time. The python flask is used to setup a web server using the Wi-Fi's IPv4 and custom port. This will start the local server on a URL = http://host:port. To receive the data from a custom API, Python Flask is used to decode the data from the API. The decoded data received is stored in a variable, and whenever a http GET request is requested the variable data's will get over written. The location and status of the fire extinguisher is stored in the server.

Whenever there is a difference between the current data and the data present in the database needs to be updated. This can be done using Pyodbc. To send data from Python flask to MS Sql. To do this first make a connection between Pyodbc and MS Sql. After the connection has been made the data can be sent using Sql queries to update the values in the database. To store the data sent from Pyodbc in the database tables needs to be created. Two tables have been created to store data. One table is to store the data of the individual device (FEOD) whenever an obstacle is blocking the fire extinguisher or when it is clear.

Table 2 SQL Device Status

Column Name	Type	Constraint
Node ID	VarChar	Foreign Key(references table 2 Node ID)
Accessibility	Number	-
Time	Date Time	-

The second table holds the data for all the devices (FEODs) with its real time accessibility. This table is used to check whether certain device is accessible or not.

Table 3 SQL Node ID Hub

Column Name	Type	Constraint
Node ID	VarChar	Primary Key
Accessibility	Number	Foreign Key(references table 1 accessibility)
Access	VarChar	-

3. Power Saving Mode

3.1. Node MCU Power Consumption

The ESP8266 uses between 15 µA and 400 mA of current. The current consumption of the Node MCU in idle mode with powered Wi-Fi is 70mA. In idle mode, a Node MCU with a 3.3V operational voltage requires the following power:

Watts required (W) = Operating voltage (U) * Current consumption (I)

231mW = 3.3V * 70mA

Node MCU energy consumption for a year is:

Energy = Watts required * days in a year (365) * hrs in a day(24) = 2.024 kWh

The device is powered by a power bank with 4000 mAh capacity. When this device is used without any power saving methods.

The total device run time:

4000mAh / 70mAh = 57.14 hrs = 2.38 days

3.2. Node MCU Sleep Mode

Node MCU comes with power saving architectures, there are mainly in 3 operating sleep modes: modem sleep mode, light sleep mode and deep sleep mode. By employing advanced power management techniques and logic to turn off non-essential functions and control the transition between sleep and active modes.ESP8266EX consumes about than 0.4mA in light sleep mode and less than 1.8mA (DTIM=1) or less than 0.55mA (DTIM=3) to stay connected to the access point. Only the calibrated real-time clock and watchdog remain operational when in sleep mode. The ESP8266EX can be set to wake up at any interval using the real-time clock.In this device light sleep mode (DTIM = 1) is used, by interfacing PIR sensor as a wake up trigger.

DTIM is a delivery traffic indication message. A DTIM period determines how often a beacon frame will contain a DTIM message, and the period value will be included in each beacon frame. When DTIM period = 1, it means that at every Beacon interval, the AP will send all temporarily buffered data packets.The average current per hour taken when using light sleep mode (DTIM = 1) = 1.8 mA.

The average current per hour taken when active = 70 mA

The device is active for at most 1 hour a day in total, Therefore the device is in sleep for 23 hrs.

Total Current Consumption in 24 hours = 70 + (1.8*23) = 111.4 mA

Current Consumption in 1 hour = 111.4 / 24 = 4.64 mAh

Total run time of device = 4000/4.64 = 862 hrs = 36 days.

Total Power Consumed in a year = 3.3v * 4.64mA * 365 * 24 = 1.341 kWh

3.3. Power Saving Mode

All the components are not active 24/7 which reduces the power intake. There are four stages in operating this device.

Stage 1: Motion detection, in this stage only one component is active. The PIR sensor is used to detect any movement and is set as a trigger when it detects movement and wakes up the Node MCU from light sleep Mode

Stage 2: Scanning, in this stage the scanning mechanism is active. After the PIR sensor detects movement, the scanning mechanism is started and the PIR sensor is put to sleep. The scanning mechanism consists of a servo motor with an ultrasonic sensor mounted on it. In case an obstacle is found below the threshold distance Stage 3 is activated. If there is no obstacle it reverts back to Stage 1 and goes to light sleep mode.

Table 4 Four stages

State	PIR Sensor	Ultrasonic Senor	Servo Motor	Buzzer	LED	Database Value
State 1: Initial State.	ON	OFF	OFF	OFF	OFF	0
State 2: Scanning Mechanism.	OFF	ON	ON	OFF	OFF	-
State 3: When Obstacle is detected.	OFF	ON	OFF	OFF	OFF	-
State 4: When Obstacle is still present after 10 seconds.	OFF	ON	OFF	ON	ON	1

Stage 3: Alarm, in this stage only the ultrasonic sensor and alarm system is active. When an obstacle is detected below threshold distance, the servo motor is stopped in that place and the ultrasonic sensor keeps detecting for a certain amount of time. If the obstacle is still in place the alarm system is activated and data is sent to the database.

Stage 4: Obstacle clearance, in this stage all components are turned off and reverted back to Stage 2. When an obstacle is cleared the alarm system is turned off, data is sent to the database and the system is reverted back to Stage 2.

Thus, with power saving mode it consumes way less power than the normal active mode.

4. Results and Discussion

The Node MCU is initially in sleep mode, it uses PIR sensor as a triggering mechanism to activate the Node MCU to start the scanning mechanism. In the Table 5, the 0 represents no movement in front of the FEOD. When there is motion in front of the FEOD the PIR sensor returns 1. This pulse activates the Node MCU from its sleep mode.

Table 5 Node MCU Activation

PIR Senor Values	Node MCU
0	Sleep
0	Sleep
0	Sleep
1	Awake
1	Awake

After the Node MCU gets activated, the scanning mechanism starts to monitor for any obstacles around the fire extinguisher. The maximum range of ultrasonic sensor is set to 700cm, as shown in 6. When the scanning mechanism starts and if it detects no obstacles it returns 700cm and the scanning continues.

Then an object is detected at 45cm. The scanning mechanism stops and ultrasonic sensor verifies if the obstacle is a moving obstacle or a stationary obstacle. A stationary obstacle is detected and the alarm is turned on. Each reading is taken in a time interval of 2 seconds. The above table 4.3 is a table created in MS SQL with three main columns namely Node ID, accessibility, and time. The Node ID is used to get the location of the device. Accessibility tells us the whether the device has any obstacle blocking the way in front of the fire extinguisher. The accessibility has two main values 0 and 1. 0 represents that the fire extinguisher is accessible and 1 represents that the fire extinguisher is not accessible. Time column displays date and time at the time of updation.

Table 6 Scanning Mechanism

Ultrasonic Sensor Values (cm)	Servo Motor	Alarm
700	On	0
700	On	0
700	On	0
45	Off	0
45	Off	0
45	Off	0
45	Off	0
45	Off	0

45	Off	1
45	Off	1

Table 7 Data from Node MCU

Results | Messages

	node_id	accessibility	time
1	test_001	1	2021-05-04 15:45:46.000
2	test_001	0	2021-05-04 15:45:35.000
3	test_001	1	2021-05-04 15:45:17.000
4	test_001	0	2021-05-04 15:44:31.000
5	test_001	1	2021-05-04 15:44:17.000
6	test_001	0	2021-05-04 15:44:07.000
7	test_001	1	2021-05-04 15:43:47.000
8	test_001	0	2021-05-04 15:43:32.000
9	test_001	1	2021-05-04 15:43:07.000
10	test_001	0	2021-05-04 15:35:42.000
11	test_001	1	2021-05-04 15:35:15.000
12	test_001	0	2021-05-04 15:35:05.000
13	test_001	1	2021-05-04 15:34:59.000
14	test_001	0	2021-05-04 15:34:44.000
15	test_001	1	2021-05-04 15:33:32.000
16	test_001	0	2021-05-04 14:54:16.000
17	test_001	1	2021-05-04 14:54:00.000
18	test_001	0	2021-05-04 14:51:17.000
19	test_001	1	2021-05-04 14:51:08.000
20	test_001	0	2021-05-04 14:50:38.000
21	test_001	1	2021-05-04 14:50:25.000
22	test_001	0	2021-05-04 14:50:08.000
23	test_001	1	2021-05-04 14:49:52.000
24	test_001	0	2021-05-04 13:49:00.000
25	test_001	1	2021-05-04 13:48:54.000
26	test_001	0	2021-04-21 14:08:25.000
27	test_001	1	2021-04-21 14:08:12.000
28	test_001	0	2021-04-19 15:38:51.000
29	test_001	1	2021-04-19 15:38:44.000

The Table 8 is created in MS SQL with three main columns namely Node ID, accessibility and access. Node ID is termed as a primary key meaning there can be only one device at a certain location. This table contains all the active devices and their location name is set in its Node ID. The Table 7 references this table 8 to get the location of the device and update its value. Accessibility column is updated with the help of table 7. It takes the latest value in the accessibility column of table 7 and sets the value in table 8 for each individual node separately. The third column access references the second column accessibility in order for a person to understand the status of the accessibility of the fire extinguisher in layman's terms. This table is created as a central viewing point for all the devices. These columns will be used by the end user to make a User Interface for better presentation of the data acquired.

Table 8 Accessibility of Fire Extinguisher

	node_id	accessibility	Access
1	test_001	1	Not Accessible
2	test_002	0	Accessible
3	test_003	0	Accessible
4	test_004	1	Not Accessible

5. Conclusion

The proposed system has been developed to remotely monitor the fire extinguishers 24/7 automatically in industry with Industry Internet of Things Technology (IIOT). The proposed system has been developed to consume less power, without the power saving methods the system would last 2.38 days with a 4000maH battery, the total power consumption in a year is 2.024 kWh. By using the power saving method the system would last 36 days with a 4000maH battery, the total power consumption in a year is 1.341kWh. This device (FEOD) is compact in size and light weight making sure that it does not occupy much space in front of the fire extinguisher. The FEOD is made using simple components which costs much less than a fire extinguisher, i.e. cost efficient. The proposed systems helps to maintain the fire extinguishers accessibility for any circumstance. Increases the wellbeing of the employee’s working environment.

Future Scope

Node MCU can be replaced with an Application Specific Integrated Circuit (ASIC) increasing the device efficiency. The server can be replaced with Raspberry Pi for better flexibility in instalment.

References

[1] Hatem, Haraa&Abdalla, Ali &Najeeb, Zainab&Abdulhameed, Zainab. (2018). Design and implementation of ultrasonic radar system for distance measurements using Arduino. International Journal of Engineering and Technology (UAE). 7. 3115-3118. 10.14419/ijet.v7i4.18103.

[2] Shkurti, Lamir&Bajrami, Xhevahir&Canhasi, Ercan&Limani, Besim&Krrabaj, Samedin&Hulaj, Astrit. (2017). Development of ambient environmental monitoring system through wireless sensor network (WSN) using NodeMCU and

“WSN monitoring”. 1-5. 10.1109/MECO.2017.7977235.
[3] K. C. Sahoo and U. C. Pati, "IoT based intrusion detection system using PIR sensor," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017, pp. 1641-1645, doi: 10.1109/RTEICT.2017.8256877.
[4] Li, Shengbo& Li, Guofa& Yu, Jiaying& Liu, Chang & Cheng, Bo & Wang, Jianqiang& Li, Keqiang. (2018). Kalman filter-based tracking of moving objects using linear ultrasonic sensor array for road vehicles. Mechanical Systems and Signal Processing. 98. 173-189. 10.1016/j.ymssp.2017.04.041.
[5] Azeta, Joseph &Bolu, Christian &Hinvi, Daniel &Abioye, Abiodun. (2019). Obstacle detection using ultrasonic sensor for a mobile robot. IOP Conference Series: Materials Science and Engineering. 707. 012012. 10.1088/1757-899X/707/1/012012.
[6] Chatterjee, Sumanta& Paul, Apurba | (2017). Autonomous, Surveillance Fire Extinguisher Robotic Vehicle with Obstacle Detection and Bypass using Arduino Microcontroller. Communications on Applied Electronics. 7. 12-17. 10.5120/cae2017652633.
[7] Chatterjee, Sumanta& Paul, Apurba& Paul, Ritaban& Banerjee, Souvik&Kanti, Pijus | (2016). Cognitive Mechatronic Fire Extinguishing Vehicle with Obstacle Detection and Bypass. Communications on Applied Electronics. 4. 54-58. 10.5120/cae2016652176.
[8] Yi Li, Jianjun Yi, Xiaoming Zhu, Zhuoran Wang and Fangwen Xu | Developing a Fire Monitoring and Control SystemBased on IoT | 2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE2016)
[9] Healey, O., Slater, D., Lin, T., Drda, B. Goedeke and A. D., "A system for real-time fire detection," in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1993, pp. 605-606.
[10] M. Hayashikoshi, Y. Sato, H. Ueki, H. Kawai and T. Shimizu, "Normally-off MCU architecture for low-power sensor node," 2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC), 2014, pp. 12-16, doi: 10.1109/ASPDAC.2014.6742852.
[11] R. S. Rosli, M. H. Habaebi and M. R. Islam, "Characteristic Analysis of Received Signal Strength Indicator from ESP8266 WiFi Transceiver Module," 2018 7th International Conference on Computer and Communication Engineering (ICCCE), 2018, pp. 504-507, doi: 10.1109/ICCCE.2018.8539338.
[12] Rajesh Singh; Anita Gehlot; Lovi Raj Gupta; Bhupendra Singh; PriyankaTyagi, "Getting Started for Internet of Things with Launch Pad and ESP8266," in Getting Started for Internet of Things with Launch Pad and ESP8266 , River Publishers, 2019, pp.i-xxiv.'
[13] A. R. Barai, M. RysulKibriaBadhon, F. Zhora and M. R. Rahman, "Comparison between Noninvasive Heart Rate Monitoring Systems using GSM Module and ESP8266 Wi-Fi Module," 2019 3rd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE), 2019, pp. 45-48, doi: 10.1109/ICECTE48615.2019.9303572.

[14] S. Saha and A. Majumdar, "Data centre temperature monitoring with ESP8266 based Wireless Sensor Network and cloud based dashboard with real time alert system," 2017 Devices for Integrated Circuit (DevIC), 2017, pp. 307-310, doi: 10.1109/DEVIC.2017.8073958.
[15] P. Srivastava, M. Bajaj and A. S. Rana, "IOT based controlling of hybrid energy system using ESP8266," 2018 IEEMA Engineer Infinite Conference (eTechNxT), 2018, pp. 1-5, doi: 10.1109/ETECHNXT.2018.8385294.
[16] R. K. Yadav and H. Vohra, "Design architecture and comparison of interactive smart button using HC-05 and ESP8266," 2017 International Conference on Computing, Communication and Automation (ICCCA), 2017, pp. 982-985, doi: 10.1109/CCAA.2017.8229937.
[17] D. Minchev and A. Dimitrov, "Home automation system based on ESP8266," 2018 20th International Symposium on Electrical Apparatus and Technologies (SIELA), 2018, pp. 1-4, doi: 10.1109/SIELA.2018.8447172.
[18] T. Thaker, "ESP8266 based implementation of wireless sensor network with Linux based web-server," 2016 Symposium on Colossal Data Analysis and Networking (CDAN), 2016, pp. 1-5, doi: 10.1109/CDAN.2016.7570919.
[19] I. Mosashvili and S. Oniani, "Distance Monitoring System with ESP8266 for Industrial Automation Machines," 2020 IEEE 5th International Symposium on Smart and Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS), 2020, pp. 1-5, doi: 10.1109/IDAACS-SWS50031.2020.9297094.
[20] 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," 2019 Fourth International Conference on Informatics and Computing (ICIC), 2019, pp. 1-5, doi: 10.1109/ICIC47613.2019.8985850.
[21] S. Lokesh, S. B. Patil and A. Gugawad, "Home Security And Automation Using NodeMCU-ESP8266," 2020 IEEE Bangalore Humanitarian Technology Conference (B-HTC), 2020, pp. 1-6, doi: 10.1109/B-HTC50970.2020.9297917.
[22] Rajesh Singh; Anita Gehlot; Lovi Raj Gupta; Bhupendra Singh; PriyankaTyagi, "Getting Started for Internet of Things with Launch Pad and ESP8266," in Getting Started for Internet of Things with Launch Pad and ESP8266 , River Publishers, 2019, pp.i-xxiv.
[23] S. S. Pakalapati, G. G. Chary, A. K. Yadaw, S. Kumar, H. K. Phulawariya and R. Kumar, "A prosthetic hand control interface using ESP8266 Wi-Fi module and Android application," 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), 2017, pp. 1-3, doi: 10.1109/ICIIECS.2017.8275978.
[24] R.Mohamed Thameez, Dr.G.Kannan,"Design and Implementation of Smart Sensor Interface for Herbal Monitoring in IoT Environment", International Journal of Engineering Research-Online, 2015, vol.3, no.2, pp.469-475.
[25] G. Suprianto and Wirawan, "Implementation of Distributed Consensus Algorithms for Wireless Sensor Network Using NodeMCU ESP8266," 2018 Electrical Power, Electronics, Communications, Controls and Informatics

Seminar (EECCIS), 2018, pp. 192-196, doi: 10.1109/EECCIS.2018.8692952.
[26] Ahmed, M.I., Kannan, G. Secure End to End Communications and Data Analytics in IoT Integrated Application Using IBM Watson IoT Platform. Wireless Pers Commun 120, 153–168 (2021). https://doi.org/10.1007/s11277-021-08439-7.