THE XIth INTERNATIONAL SYMPOSIUM ON ADVANCED TOPICS IN ELECTRICAL ENGINEERING
March 28-30, 2019
Bucharest, Romania

Publish-Subscribe Deployment Alternatives for Scenarios Related to University Laboratory Safety

Radu Nicolae Pietraru, Luminița Gabriela Zegrea, Anca Daniela Ionita
University POLITEHNICA of Bucharest, Automatic Control and Industrial Informatics Department
radu.pietraru@aii.pub.ro, luminita.zegrea@aii.pub.ro, anca.ionita@aii.pub.ro

***Abstract-* University laboratories are subject to various risks, in respect with their specific activity domains, and a constant monitoring based on sensors and automate notifications can contribute to ensure safety conditions, in addition to the typical guidelines. This paper presents an experiment performed with a publish-subscribe platform typical for home automation, applied for monitoring various parameters within laboratories, and triggering sound and light alarms. Four scenarios, correspondent to different deployment alternatives, were tested, measuring alarm triggering times, to investigate the impact of the publish-subscribe broker deployment decision.**

***Keywords*: timeliness assessment, laboratory safety, smart building, home automation, publish-subscribe architecture**

I. INTRODUCTION

Academic and research laboratories are exposed to a series of hazards that have to be carefully identified and systematically monitored [1]. They are also very diverse, as the risks in a room with personal computers are different from the heating probabilities in a data center, or from the high hazard substances in a chemistry laboratory, where one should be aware of toxins, carcinogens, explosives, air or water reactivity and many others [2]. Given a technical university, with a large variety of engineering domains, the requirements for hazard management may be very different from one laboratory to another, therefore a centralized campus monitoring system should take into account this non-homogeneity.

General safety guidelines are established in each institution, in regard with safety equipment, responsibilities, identification of hazards, procedures for normal operation and for emergency reactions. This is valid not only for universities [3], but also for science programs in secondary schools [4]. Several automated solutions are in place and more are investigated for various laboratory categories. Internet of Things (IoT) is typically used in the new monitoring approaches, like the one proposed in [5] for a chemical laboratory. The environment can be monitored remotely with temperature, light and gas sensors, as well as video cameras, and controlled with embedded algorithms whose results may be transmitted through an Ethernet connection [6]. The physical presence of the equipment in the appropriate laboratory was also tracked with Radio Frequency Identification (RFID) [7].

Our goal is to investigate the possibility to monitor specific hazards in a non-homogeneous assembly of laboratories, which is the current situation in technical universities. On the one hand, we are interested of creating personalized sensing units for each laboratory, to measure the physical quantities that are related to the potential risks that are characteristic to the kind of practical work performed there by students or researchers, and to their correspondent domain of study. On the other hand, we are interested of creating a dependable architecture [8], where potential failures of its parts do not lead to undesirable effects for various scenarios and emergency situations. Among the attributes that characterize dependability, we are particularly interested of safety, to mitigate the risks students would be exposed to.

This work proposes a solution for the identification and management of hazard situations using the openHAB platform. The openHAB (open Home Automation Bus) platform is an open-source, technology-independent platform for home automation [9]. The openHAB platform is flexible enough to meet the intended purpose of the work even though it is not part of the core functionality. There are other scientific works such as [10], which extend the functional area of openHAB beyond the basic functionality. The platform collects data from multiple sensors and allows detection of hazard situations and alarming.

The communication between sensors and openHAB is achieved through MQTT (Message Queuing Telemetry Transport) – a client-server publish / subscribe message transmission protocol, which is light, simple and designed to be easy to implement. These features make it ideal for use in many situations, including in constrained environments such as Machine to Machine (M2M) and Internet Objects (IoT), where a small code and bandwidth of the network are required [11].

The paper continues with related work, then it presents the three sensor nodes developed for our experiment (Section III) and the deployment alternatives proposed for the MQTT broker (Section IV). Sections V and VI describe the tests performed, based on four scenarios, and the results obtained for various alarm triggering times. Section VII discusses the data and concludes the paper.

II. RELATED WORK

The use of openHAB at the level of a large building was approached in [12]. They use a central master and multiple slave controllers, distributed among the rooms, organized according to a publish-subscribe architectural style. The slave nodes that monitor various areas of the building are organized into groups with specific access permissions, managed through an authentication service.

978-1-7281-0101-9/19/$31.00 ©2019 IEEE

Frei et al. designed an extensible wireless sensor network with eight possible configurations of sensor boards, to determine the thermal performance of a building and apply it to a single-family home in Switzerland [13]. In addition to the typical sensor nodes, Alnabhan et al. introduce the notion of decision nodes, to give advice on the safest evacuation paths in case of emergency and to avoid people getting trapped in areas without any available exit in respect with the circumstances [14]. The algorithms, driven by events, are implemented in MATLAB and are simulated for three scenarios that vary the scale of the building and the intensity of the hazard.

The Safety of Buildings Structure (SBS) is also ensured using IoT-based monitoring systems, to be able to detect hidden structural dangers and to provide services to a large range of stakeholders, from residents to government representatives. The solution proposed in [15] applies the Wisdom Web of Things (W2T), considering the relation between the physical, cyber and social worlds, and using semantic information about sensors, buildings, weather, and geospatial localization. Other systems are developed for specific hazards, like earthquakes, as the example concerning a high-rise university building, where motion monitoring supports early warning and emergency response [16].

Such applications generally belong to the domain of Building Information Modeling (BIM) [17], which covers the entire lifecycle, including the construction phase. This is also applied, for instance, to monitor the gas concentrations in underground construction sites, like tunnels. Cheung et al. introduce a Zigbee wireless modules for sensing methane gas, temperature and humidity, and use control nodes that can activate alarms and ventilators in case of emergency [18].

III. Laboratory-Specific Sensor Nodes

This section presents three nodes developed for the experiment performed to test the deployment alternatives of the MQTT broker and their influence on the alarm triggering times. Two of them are conceived for monitoring the laboratory environment and the fire risks respectively. The third introduces an alarm that may be automatically commanded in case a risk event occurs. However, other types of monitoring nodes can also be used within the architecture; one can add other hardware and sensor architecture configurations, as long as the connection and the transmission of data to the MQTT broker can be achieved.

A. Environment Monitoring Node

Monitoring the total concentration of volatile compounds is important for the quality of the working environment, as shown in [19].

The environment monitoring nodes measure the temperature, the humidity and the total concentration of volatile organic compounds. These parameters characterize the air quality of an office or a classroom, and can also indicate a harmful working environment or, in an extreme case, even an immediate danger situation.

The CS811 sensor [20] requires compensation for relative temperature and relative humidity, therefore it is also necessary to measure these parameters. Another feature of this sensor is the possibility of calculating the eCO2 (equivalent carbon dioxide) required to correctly determine the ventilation requirement in a room.

The node built for our experiment is based on an ESP8266 WiFi microprocessor [21] that allows both digital parameter acquisition and WiFi communication, to connect to the MQTT broker (Fig. 1). The node is characterized by a low consumption mode and can operate for a long time (6-12 months), using its own power sources. This type of node was used within all the tests subsequently described in this paper.

B. Fire Risk Detection Node

For fire risk detection, we built a trigger type node (Fig. 2). The selected sensor (MQ2) is sensitive to both flammable gases and smoke, allowing the monitoring of pre-fire and actual fire conditions. The MQ-2 sensor [22] can also be used for the detection of health-damaging factors, such as cigarette smoke [23].

In order to bring an additional element to a typical solution, the node architecture is not based on a programmable acquisition and a communication electronic circuit, but on a mixed microprocessor and microcontroller platform, known as Arduino Yun [24]. This platform ensures both real-time operation and support for the complexity required for secure remote communication.

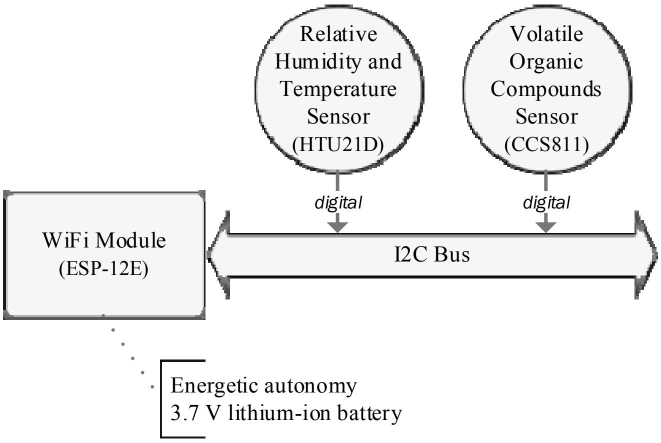

Relative Humidity and Temperature Sensor (HTU21D)
Volatile Organic Compounds Sensor (CCS811)
digital
digital
WiFi Module (ESP-12E)
I2C Bus
Energetic autonomy
3.7 V lithium-ion battery

Fig. 1. Environment Monitoring Node

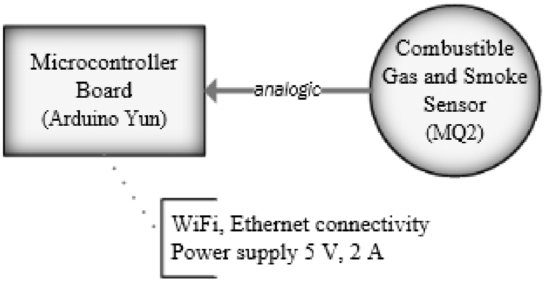

Microcontroller Board (Arduino Yun)
analogic
Combustible Gas and Smoke Sensor (MQ2)
WiFi, Ethernet connectivity
Power supply 5 V, 2 A

Fig. 2. Fire Risk Detection Node

In addition, the presence of the OpenWRT Linux operating system on the platform's Atheros AR9331 microprocessor allows the node to be transformed into a WiFi access point for other nodes - useful for deploying it in an area lacking the required WiFi infrastructure.

C. *Alarm Node*

The alarm node has the role of alarming people situated in the monitored perimeter about the detected danger. Our solution is based on an ESP8266 WiFi microprocessor [21], which retrieves information from an MQTT broker (Fig. 3).

The processing power provided by the Tensilica Xtensa architecture allows the implementation of local trigger conditions, as explained in the next sections.

IV. Deployment Alternatives

A. *Internet Deployment of the MQTT Broker*

The first implementation option stands in connecting the nodes to the Internet via a WiFi infrastructure. The Internet connection will allow the nodes to communicate with an MQTT broker located on the Internet. The communication will be achieved through the publish / subscribe architectural style, specific to the MQTT protocol. The public MQTT HiveMQ server [25] was used for testing (see Fig. 4).

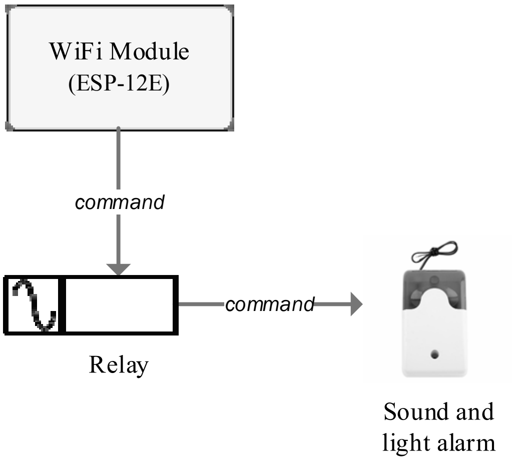


Fig. 3. Alarm Node

The transmitted data are collected by an openHAB platform that will allow one to have access to the centrally acquired values and to implement alarm mechanisms. For testing, a server running Linux CentOS 7 and OpenHAB 2.4 was used.

The users can visualize the measured values in the openHAB platform web interface (Fig. 5), as well as in the mobile app (Fig. 6). The mobile app can also offer instant alarm functionality.

For testing, version 2.5.6 of the mobile application under the Android operating system was used, together with the OpenHAB Cloud Connector 2.4.0 add-on.

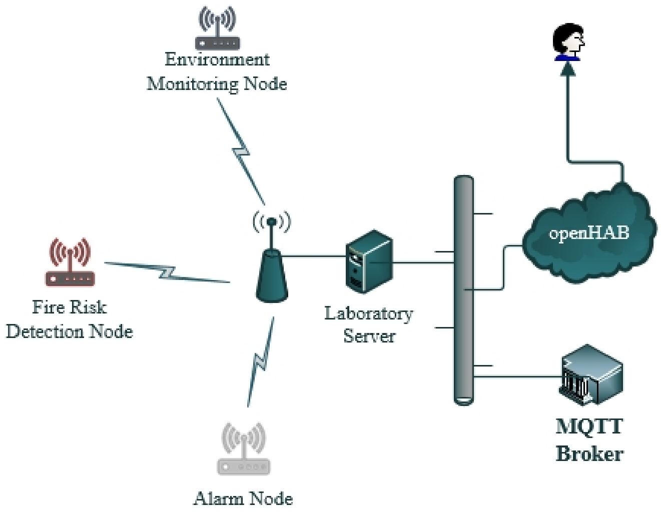
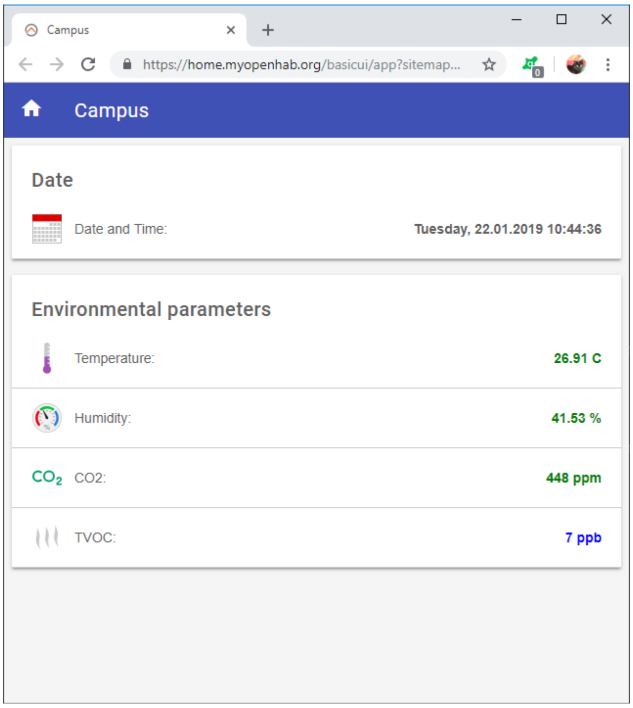


Fig. 4. Deployment alternative with MQTT Broker deployed on the Internet



Campus

Date

Date and Time: Tuesday, 22.01.2019 10:44:36

Environmental parameters

Temperature: 26.91 C

Humidity: 41.53 %

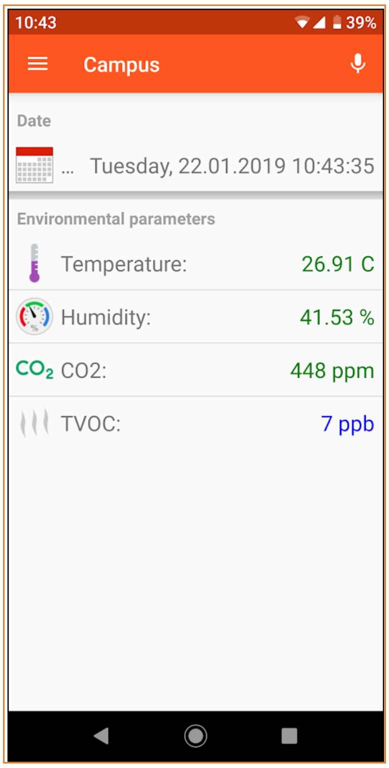
CO_2 CO2: 448 ppm

TVOC: 7 ppb

Fig. 5. OpenHAB web interface for an environmental node

B. *Local Deployment of the MQTT Broker*

In this deployment alternative, the MQTT broker is on the same local network as the set of nodes (Fig. 7), and it does not serve to other purposes also. Grouping of nodes can be done based on a top-level campus distribution, a group of laboratories, or a building. The openHAB platform will thus communicate with multiple MQTT brokers. For the users, there will be no difference in the use of openHAB.



10:43 39%

Campus

Date

... Tuesday, 22.01.2019 10:43:35

Environmental parameters

Temperature:	26.91 C
Humidity:	41.53 %
CO2:	448 ppm
TVOC:	7 ppb

Fig. 6. OpenHAB mobile interface for an environmental node

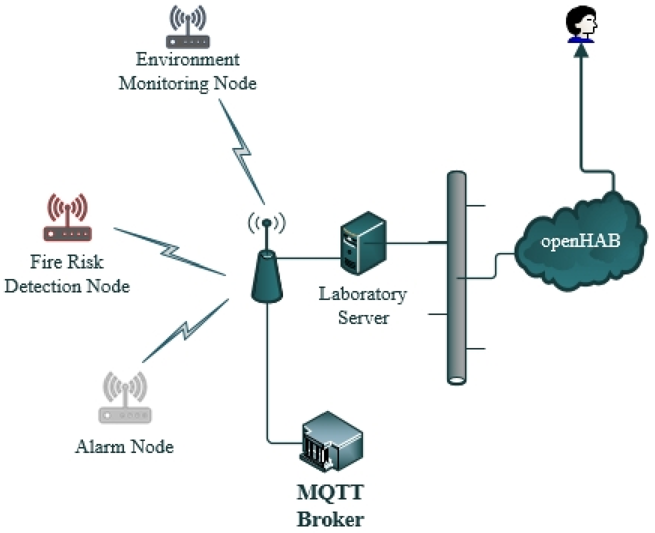


Fig. 7. Deployment alternative with MQTT Broker deployed locally

The tests were performed using Mosquitto version 1.5.5 [26] as MQTT broker, installed on an embedded Orange Pi One platform [27], running Linux Raspbian 8.

V. Test Scenarios

Four warning scenarios were tested by comparing the time elapsed between the occurrence of the alarm event and the actual alarm of the users. Two scenarios are based on the MQTT Broker Internet-based deployment, and two on the local deployment of the MQTT Broker. The scenarios in the two categories differ by the level at which the alarm mechanism logic is implemented. The alarm logic can be implemented at the openHAB platform level, or it can be deployed at the alarm node level.

A. Scenario 1

In the first scenario, the acquisition nodes send the measured values by publishing events to the MQTT Internet broker. From there, the values are taken over by the OpenHAB platform and, if the values meet the alarm criteria, MQTT messages are posted, through the same broker, to the alarm nodes.

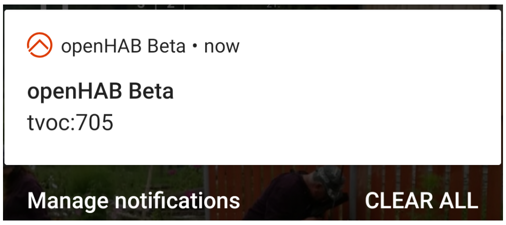
B. Scenario 2

In the second scenario, the alarm nodes directly take the messages from the acquisition nodes through the MQTT Internet broker and they decide on their own whether the values meet the alarm criteria.

The difference of *Scenario 2* in respect with *Scenario 1* consists in the local trigger logic, through the alarm nodes installed in the monitored area, alarming the users in that area. The remote monitoring and alarming (emails, mobile notifications – Fig. 8, Fig. 9) through the openHAB platform has the same logic of deploying in both scenarios: the platform takes messages via the MQTT Internet broker and if the values meet the alarm criteria, specific alarms will trigger.

```
rule tvoc
 when Item tvoc changed
 then if (tvoc.state>660) {
  sendNotification("mobileacc","tvoc:"+
                            tvoc.state.toString)
  sendMail("email", "eCO2",tvoc.state.toString) }
end
```

Fig. 8. OpenHAB rules for mobile and email alarm



openHAB Beta • now

openHAB Beta
tvoc:705

Manage notifications CLEAR ALL

Fig. 9. Mobile notification

Scenario 2 attempts to test a possible faster local alarm, for the people in the monitored area. Even if the implementation is difficult at first sight, due to the management of node alarm criteria, this impediment can be countered by implementing a remote firmware change system.

C. Scenario 3

The third scenario is based on a local MQTT broker that can serves an area or a collection of nodes (floor level, group of laboratories, building). This deployment method attempts to introduce a scalability factor in the proposed alarm architecture.

Each node in a particular group reports to the MQTT Broker to which it belongs. The openHAB platform gets connected to each MQTT broker separately. The alarm trigger logic is implemented at the OpenHAB platform level.

D. Scenario 4

In the fourth scenario, the alarm trigger logic is implemented at the alarm node level (in the same way as in *Scenario 2*).

Unlike the single MQTT-based implementation, multi-MQTT broker-based architecture requires a larger configuration effort, but this is natural for a large-scale deployment. Moreover, a major functional advantage of this deployment solution is the possibility of triggering a local alarm, even if the Internet connection is inoperable. For alarm nodes, the trigger loop closes at the local network level, i.e. at the MQTT broker level in the local network.

VI. Results

The tests were performed in accordance with the scenarios described above. They measured the time elapsed between the occurrence of the alarm event (changing environmental parameters or measurements specific to health-damaging cases) and triggering local / remote alarms (emails and mobile alerts). The term mobile alerts is used for mobile push alerts generated by the openHAB mobile application (pop-up messages generated on the mobile phone). It has been assumed that all the systems are connected to the Internet and there is a common reliable synchronization source through the NTP service. The time difference calculation was performed between the times reported by each system.

The results are presented in Tables I÷IV. Three series of tests were performed, based on:

- the occurrence of a single event,
- triggering a series of 100 sequential events generated by a single acquisition node (in 10 minutes window) and
- triggering of a series of 1000 events, generated by 10 nodes in parallel (in 10 minutes window).

All the tests were performed using a single local alarm node, one email address, and one mobile phone (connected to the Internet via 4G services).

TABLE I
Alarm Triggering Times for *Scenario 1*

Metrics / Test Series	Local Alarm Node Triggering	Email Receiving Time	Mobile Notification
1 single event	10 ms	10 s	3 s
100 sequential events (average)	200 ms	10 min	4 min
1000 parallel events (average)	1400 ms	12 min	6 min

TABLE II
Alarm Triggering Times for *Scenario 2*

Metrics / Test Series	Alarm Node (local alarm)	Email Receiving Time	Mobile Notification
1 single event	4 ms	12 s	2 s
100 sequential events (average)	180 ms	10 min	5 min
1000 parallel events (average)	1200 ms	11 min	10 min

TABLE III
Alarm Triggering Times for *Scenario 3*

Metrics / Test Series	Alarm Node (local alarm)	Email Receiving Time	Mobile Notification
1 single event	12 ms	10 s	3 s
100 sequential events (average)	200 ms	12 min	4 min
1000 parallel events (average)	1000 ms	12 min	10 min

TABLE IV
Alarm Triggering Times for *Scenario 4*

Metrics / Test Series	Alarm Node (local alarm)	Email Receiving Time	Mobile Notification
1 single event	2 ms	9 s	4 s
100 sequential events (average)	100 ms	10 min	5 min
1000 parallel events (average)	300 ms	13 min	10 min

The testing was done using a local network and an Internet connection in the normal operation parameters. Given the nondeterministic character of TCP/IP protocols, the test results can be greatly affected by overcrowded or defective communication networks. High uploading of email or mobile phone networks can also lead to increased alarm times.

VII. Conclusion

The paper demonstrated the possibility of using the openHAB platform as a monitoring and alarm tool for several scenarios regarding the safety of university laboratories. Using IoT devices, the openHAB platform, and the MQTT protocol, an integrated scalable monitoring and alarm solution was experimented.

The carried-out tests revealed the conclusions below:

a) The proposed solution ensures enough time to alarm the persons in the monitored area, e.g. in case an emergency evacuation is needed.

b) For remote alarming by email or mobile notifications, time is too long to be effective to people directly affected by the emergency, but they can be considered for the university management personnel.

c) The implementation of local MQTT brokers is a solution that offers better local alarm times, allowing for a scalability of the proposed solution.

d) In the event of a hazard occurring on an extending area, a large number of monitoring nodes can lead to overloading the warning system through many messages.

e) Even if the implementation of nodes alarm trigger logic raises management problems, it can lead to better response time, and independence from Internet connection.

REFERENCES

[1] A. Olteanu, F. Lacatusu, I. Craciun, M. Lacatusu, A. D. Ionita, Mobile Application for Crisis Situations in a University Campus, International Scientific Conference eLearning and Software for Education Bucharest, April 19-20, 2018, 10.12753/2066-026X-18-109, pp. 280-287

[2] ACS, Identifying and Evaluating Hazards in Research Laboratories. Guidelines developed by the Hazard Identification and Evaluation Task Force of the American Chemical Society’s Committee on Chemical Safety, American Chemical Society, 2015, Available at: https://www.acs.org/content/dam/acsorg/about/governance/committees/chemicalsafety/publications/identifying-and-evaluating-hazards-in-research-laboratories.pdf

[3] University of Toronto, Laboratory Safety Program, October 2012, Available at: https://ehs.utoronto.ca/wp-content/uploads/2015/10/Laboratory-Safety-Program.pdf

[4] Safety in Science Laboratories, Education Bureau Kowloon Tong Education Services Centre, 2013, Available at: https://cd1.edb.hkedcity.net/cd/science/laboratory/safety/SafetyHandbook2013_English.pdf

[5] H. Kim, E. Lee, D. Kwon and H. Ju, "Chemical laboratory safety management service using IoT sensors and open APIs," 2017 International Conference on Information and Communications (ICIC), Hanoi, 2017, pp. 262-263.

[6] M. Manoj Kumar, G. Srinivasa Raju, Design and Implementation of the Lab Remote Monitoring and Controlling System Based on Embedded Web Technology, International Journal of Scientific and Research Publications, Volume 3, Issue 3, March 2013

[7] M. H. A. Wahab et al., "Web-based laboratory equipment monitoring system using RFID," 2010 International Conference on Intelligent and Advanced Systems, Manila, 2010, pp. 1-5.

[8] M.L. Fairbairn, Dependability of Wireless Sensor Networks, PhD Thesis, University of York, Computer Science, September 2014, Available at: https://core.ac.uk/download/pdf/30267741.pdf

[9] OpenHAB Official Documentation https://www.openhab.org/docs/ Accessed February 28, 2019

[10] M. F. Roslan, A. Ahmad, A. Amira, Real-Time High Jump Wearable Device with ESP8266 for High-Performance and Low-Injury, International Journal of Integrated Engineering – Special Issue 2018: Seminar on Postgraduate Study, Vol.10 No. 3 (2018) p. 14-19

[11] MQTT Version 3.1.1, OASIS Standard, 29 October 2014 http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html Accessed February 28, 2019

[12] F. Heimgaertner, S. Hettich, O. Kohlbacher, M. Menth, Scaling Home Automation to Public Buildings: A Distributed Multiuser Setup for OpenHAB 2, Global Internet of Things Summit (GIoTS 2017), June 2017, Geneva, Switzerland

[13] M. Frei, J. Hofer, A. Schlüter, Z. Nagy, An easily-deployable wireless sensor network for building energy performance assessment, Energy Procedia, Elsevier, Volume 122, 2017, Pages 523-528.

[14] N. Alnabhan, N. Al-Aboody, H. Al-Rawishidy, Adaptive Wireless Sensor Network and Cloud-based Approaches for Emergency Navigation, The 42nd IEEE Conference on Local Computer Networks (LCN), October 9-12, 2017, Singapore, Demonstration, Available at: https://www.ieeelcn.org/prior/LCN42/lcn42demos/1570388472.pdf, Accessed November 16, 2018

[15] H. Wang, Z. Huang, N. Zhong, J. Huang, Y. Han, F. Zhang, An Intelligent Monitoring System for the Safety of Building Structure under the W2T Framework, International Journal of Distributed Sensor Networks, Volume 2015, Article ID 378694, 16 pages

[16] T. Kubo, Y. Hisada, M. Murakami, F. Kosuge, and K. Hamano, “Application of an earthquake early warning system and a real-time strong motion monitoring system in emergency response in a high-rise building,” Soil Dynamics and Earthquake Engineering, vol. 31, no. 2, pp. 231–239, 2011.

[17] M.D. Martínez-Aires, M. López-Alonso, M. Martínez-Rojas, Building information modeling and safety management: A systematic review. Safety Science, Volume 101, January 2018, Pages 11-18.

[18] W.-F. Cheung, T.-H. Lin, Y.-C. Lin, A Real-Time Construction Safety Monitoring System for Hazardous Gas Integrating Wireless Sensor Network and Building Information Modeling Technologies, Sensors 2018, 18, 436

[19] O. J. Adebayo, O. O. Abosede, F. B. Sunday, A. Ayooluwa, A. J. Adetayo, S. J. Ademola, A. F. Alaba, Indoor Air Quality Level of Total Volatile Organic Compounds (Tvocs) in a University Offices, International Journal of Civil Engineering and Technology, 9(11), 2018, pp. 2872–2882

[20] CCS811 Ultra-Low Power Digital Gas Sensor for Monitoring Indoor Air Quality, https://ams.com/documents/20143/36005/CCS811_DS000459_6-00.pdf/c7091525-c7e5-37ac-eedb-b6c6828b0dcf Accessed February 28, 2019

[21] ESP8266EX Datasheet V6.0 2018 https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf Accessed February 28, 2019

[22] MQ-2 Semiconductor Sensor for Combustible Gas https://www.pololu.com/file/0J309/MQ2.pdf Accessed February 28, 2019

[23] S. Panpaeng, P. Phanpeang, E. Metharak, Cigarette Smoke Detectors for Non-Smoking Areas in the Building, 22nd International Computer Science and Engineering Conference (ICSEC 2018), p.208-211

[24] Arduino Yún https://www.jameco.com/jameco/products/prodds/2193441.pdf Accessed February 28, 2019

[25] HiveMQ | Public Broker | MQTT Dashboard, http://www.mqtt-dashboard.com/

[26] Eclipse Mosquitt An open source MQTT broker, https://mosquitto.org/

[27] Orange Pi One, http://www.orangepi.org/orangepione/