Journal of Green Engineering (JGE)

Volume-10, Issue-12, December 2020

IoT-Realtime Fire Detection and Monitoring

[1]Rostam Bin Salleh, [2]Sallehuddin Mohamed Haris, [3]Rizauddin Ramli

[1]Industrial Automation Section, University Kuala Lumpur Malaysia France Institute, Selangor, Malaysia.
E-mail: rostam@unikl.edu.my
[2.3]Department of Mechanical and Materials Engineering, Faculty of Engineering & Built Environment, Universiti Kebangsaan ,Malaysia.

Abstract

The purpose of this project is to decrease the chance of the large fire hazard from occurring due to unreliable existing fire alarm systems. Existing fire alarms have proven to be unreliable at times due to faulty or flawed systems. Our main purpose is to improve the existing fire alarms with new technology such as MICA2 device which contains several sensor nodes that are capable of reading several variables that can be processed to find the point which fire is present which will analyze the image to read the pattern of the fire to further improve the capability of detecting fire. With this device, a new and improved system can be produced. The new system will be an Internet-of-Things-based system (or IOT-based) that will allow firefighters or any person with authority to determine the presence of fire by looking at a warning display. The warning can be viewed on any device that is capable of connecting to the internet.

Keywords: Internet of thing, Sensor Network, MICA2 Sensor, PostgreSQL, Fire detection.

1 Introduction

This Fire detection and alarm systems are designed to provide warning of the outbreak of fire and allow appropriate firefighting action to be taken before the situation gets out of control. As all systems are designed primarily to protect life and property, this places a great responsibility on the designer because each building will present a different set of problems in relation to the risk of fire and fire spread. Each fire detection and alarm system therefore

Journal of Green Engineering, Vol. 10_12, 13076-13088.

must be specially designed to meet the requirements of each building. The most common device used for fire detection is the smoke detector that can be observed. Smoke detectors can be seen in most buildings.

The purpose of this project is to further improve the existing fire alarm system where it will increase the performance of smoke detector when combined with the MICA2 sensor. We will be using this device which detects several variables such as humidity, pressure, temperature and light density which are variables that are affected by the presence of fire. We will add imaging to the project as well in order to create a system with high level of reliability where we might just eliminate the use of sprinklers because sprinklers can also cause loss and damage to valuable assets and equipments that are easily destroyed when in contact with water.

A real-time data transfer will be developed where the data gathered by the devices can be shared among other parties such as the firefighting department where it will increase the response time of fire fighters in acting when a real fire hazard occurs.

2 Page Layout

A several studies and projects have been conducted regarding this field that can be made as references in order to gather and obtain knowledge to further improve or innovate the present system that exists for this project. A literature review is about discussing and comparing the information gathered and collected from resources such as articles, journal, research paper, and previous final year project which are related to this project.

2.1 Sensor

Sensor plays a vital role in this project because the result relies on the data that is being read and analyzed in order to get the accurate data which is important as a detecting and monitoring system. Therefore, sensor selection is very important.

Of course, in a fire detecting project, a lot of variables can be read to conclude that there is a fire present at the vicinity. Most common variables that are related to fire is temperature, light, and gas.

For the gas variable, the most common sensor is the MQ-02 which, in this case, was used and experimented by Vijayalakshmi and Muruganand [1] Tiwari and Bandopadhaya [2], and Sujatha, Bhavani and Ponmagal [3]. This sensor can detect methane, butane, and smoke as well as flammable gas that could lead to a disaster. The same type of sensor but a different generation which is MQ9 was used by Sujatha, Bhavani, Reddy and Kumar (n.d.). Just by viewing the paper, it proves that it is quite a reliable sensor node in detecting smoke or gas as accurate data can be obtained from it.

Secondly, for the light (lux) variable a variety of sensors are used. Vijayalakshmi and Muruganand [4] used TEMT6000 that is an ambient light sensor that mimics the behavior of a transistor. The higher the value of light being directed to the sensor, the higher the analog value it produces. Tiwari and Bandopadhaya [5], on the other hand, used a flame sensor which is an unidentified sensor that was used to detect the light intensity that is connected to the microcontroller to read the analog value that is being generated from reading the light. Like the MICA2 light sensor node, it has an infrared(IR) light emitting diode and IR photo diode that detects the IR light which is supposed to have the same wavelength as the one that is being transmitted wave by the IR LED (light emitting diode).

Temperature is the main variable that should be detected by a sensor because it is one hundred percent affected when there is fire. By detecting temperature change, we can easily determine the presence of fire before it can cause a larger fire that can cause disaster. Sujatha, Bhavani and Ponmagal [6] used LM35 series which is a device with an output voltage linearly proportional to the Centigrade temperature. It is an independent device which can read accurately without external calibration. Besides Sujatha, et. al. [7], Sujatha, Bhavani, Reddy and K. R. Kumar (n.d.) used the similar sensor because of its efficiency and calibrated feature which is just a good standalone device that does not require additional calibration.

2.2 Microcontroller

Vijayalakshmi and Muruganand [8] used two microcontrollers as a master and slave. The Raspberry Pi 3 (RPI) was the master device or main device and the Arduino Mega was the slave which sent data using the ESP-01 WLAN sensor module to the master. The Arduino converts the analog data gained from the sensors and during transmission to the RPI it converts it to digital therefore the data can be understood by the RPI. With the help of the Arduino it eases the task that is being done by the RPI which allows a smooth progression of data but at a costly procedure where two microcontrollers are used for the sake of smoothness of data transmission. The project proved to be successful despite the high cost involved.

Sujatha, Bhavani and Ponmagal [9], on the other hand, used an Intel Galileo Gen 2 board which is unique since it is the only research that could be found with such microcontrollers being used but unfortunately the project came out with having several hardware and software errors during development which might make an impression that it requires a lot of attention to the development phase in order to avoid any complication to develop during development.

Like Vijayalakshmi and Muruganand [10], RPI was used by Sujatha, Bhavani, Reddy and Kumar (n.d.). RPI is often being chosen for this type of project due to the versatility of the device which is independent yet functional.

2.3 Programming Language

2.3.1 C / C++

This language is quite often being used because of its simplicity compared to Python and Java which are complex. It is also being used because Arduino uses this language in programming the microcontroller which then could be programmed into the microcontroller through Arduino IDE (Integrated Development Environment). Of course, in [1] and [2] is used due to the involvement of Arduino in the system being built by using C++ language. Instead of using it as a programming language for the microcontroller, Sujatha, Bhavani, Reddy and Kumar (n.d.) used the language to create a virtual environment for simulation using Visual Studio 2017 which proves this language may be simple yet the result it is capable to yield is underrated.

2.3.2 Python

Python is one of those languages that is hard to master. However, it is worth going through the hassle of learning it because its capabilities are proven despite its complexity. RPI uses Python language which already proves the point where the RPI is a very rewarding microcontroller to be used because of its versatility. Vijayalakshmi and Muruganand [11], and Sujatha, Bhavani, R.Deepan , C.Jothi Kumar ,et. al. [12-19](n.d.) have proven that it is quite versatile in using it as a main microcontroller to control the output for both researches.

3 Development Processes

The sensor configuration was an important part of this project. In order to be able to read the right variables, some configuration was done for the mica2 device to function as it was intended. Mote.View was use in order to program the device to match the function of said device in Figure 1.

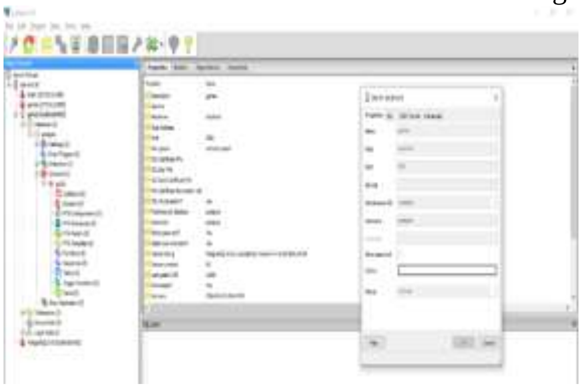


Figure 1 PgAdmin 4 Window Panel

3.1 Mica2 Device

Figure 2 Mica2 sensor is quite a unique device where several sensor nodes are already implemented in the device. It only requires a program to be implemented into the hardware for it to start. The device was given with a gateway that functions as a receiver to receive all the data read by the device and displayed into whatever device the gateway is connected, which, in this case, was the host computer.

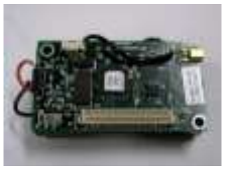


Figure 2 MICA2 Device

3.2 Mote.View

Mote.View is device that was created by Crossbow Technology which is the same company that created the mica2. It serves as a medium to program the mica2 and other crossbow devices. It helps further simplify the method as well as the reading of sensor where it serves as a GUI (Graphical User Interface) to the user where it shows the result. Some configuration was done using the software, so that the sensor reads all the data and acquire the data live and display it in Mote.View in Figure 3. The software was configured to read data every 5 seconds. The software also helps to store the data read into a PostgreSQL database to allow the user to go through past readings which that places all the data into a table with could serve as a history of data readings done by the device.

Figure 3 Mote.View GUI

3.3 PostgreSQL

PostgreSQL is the database used by Mote.View to store all data read from mica2 including the live as well as the past data. PgAdmin 3 and PgAdmin 4 were used as the user interface to further simplify the process of doing work with PostgreSQL. [20-22]The two databases used were the localhost's database which stores the data read by the sensor, and the other database is from a subscribed server.

One problem with the local host database is that the data being logged into the table were raw units or data that were not converted yet by the software. This caused a complication as the right value was not displayed as intended.

3.4 Python

Python programming language is an open-source coding language that was used in this project to extract the database from PostgreSQL and was also used to send it to subscribe server in order to be stored in a cloud storage so that it can be viewed by others that uses the Python coding with the right username and password. It is secured in the cloud with permission only to those who have access to it.

A few packages were added from the vanilla version of Python which allowed Python to implement the PostgreSQL language. One of these packages was psycopg2 which is used to connect with the host PC's database and extract the data to be implemented into the subscribed server for the app to reach and display as a GUI for users. Once the package was installed into the current Python version, a simple "import psycopg2" coding line is used to include the library into the Python's coding library.

Creating a connection to a PostgreSQL database is quite similar to creating a connection with a MySQL database but instead of connection = mysql.connect ([database credentials]) we implemented the PostgreSQL library. Therefore, the right way would be connection = psycopg2.connect([database credentials]). The integration was made was intended for PostgreSQL database connection shown in Figure 4.

Once connection was made with the host server that stored the data, some setup was required in order to prepare the Python for a PostgreSQL integration which is by typing cursor= connection.cursor () and cursor. Execute("SELECT version();") in order to allow the selected table to be executed. Once it is set up, with the same command in PostgreSQL in selecting a table just with a few additions by typing postgreSQL_select_Query = "select * from mts_results" in this case where the intended query to be selected is mts_results and to execute the PostgreSQL action is to type a set of coding which is cursor. Execute (PostgreSQL_select_Query). After executing the PostgreSQL command, command.fetchall() was used instead of a different fetch since all of the data in the data is needed so that it would simplify the task at hand rather than fetching a single data one at a time.[23-25] Once all the data are extracted, connection was closed by simply adding connection. Close ( ) to disconnect from the host server.

After the connection with the host server is no longer active, connection with the subscribed server was made in the same way as the previous connection was done but, instead, the credentials of the subscribed server were replaced such as server name, port, username, password and database name. This time, instead of selecting a query we will update a query. Therefore, the extracted data taken from the other database will be updated into the subscribed server to replace it in order to save space rather than adding a new data since the space for the subscribed server is limited in this case which is only 20MB.

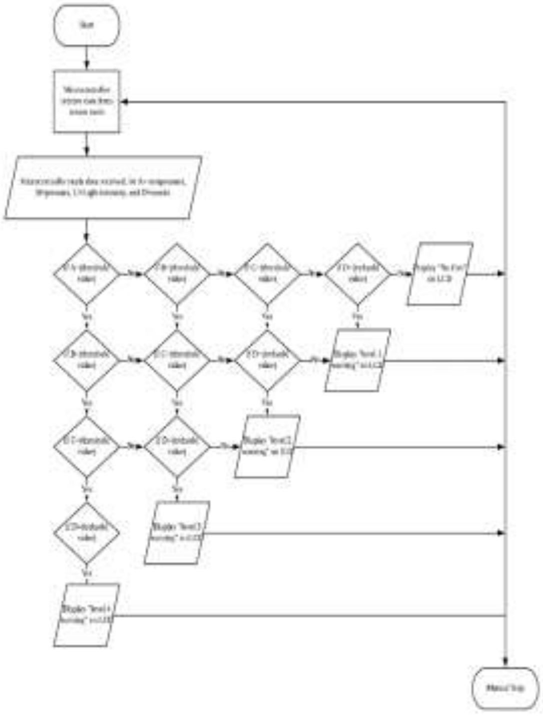


Figure 4 PostgreSQL Flow Connection

4 Result

Based on the results in Figure 5 & 6 shows that graphs of reading differences according to the fire size *L0,L1,L2* and *L3*. The variable of lux and temperatures graphs show relatively significant differences from the parameters measured.

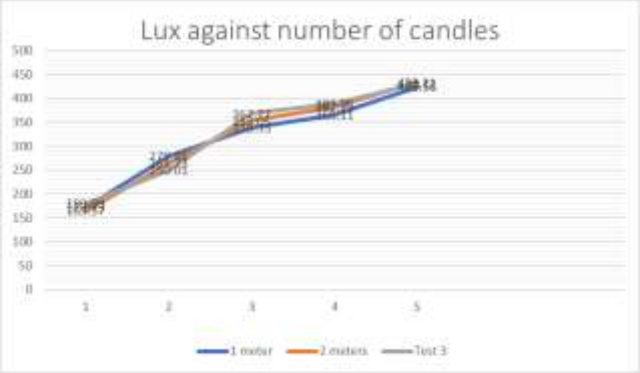


Figure 5 Lux Number of Candles

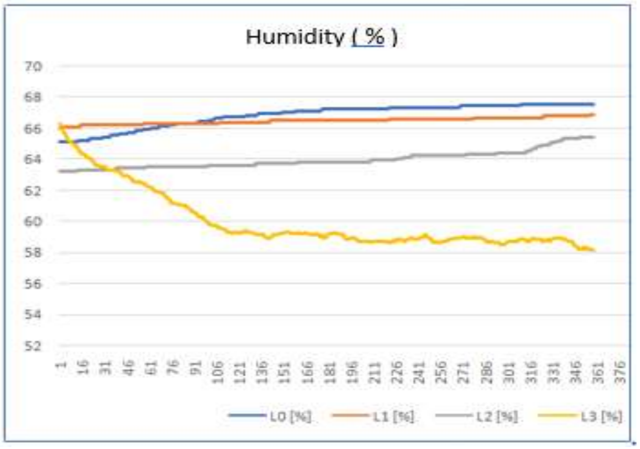


Figure 6 Humidity L0, L1, L2 & L3

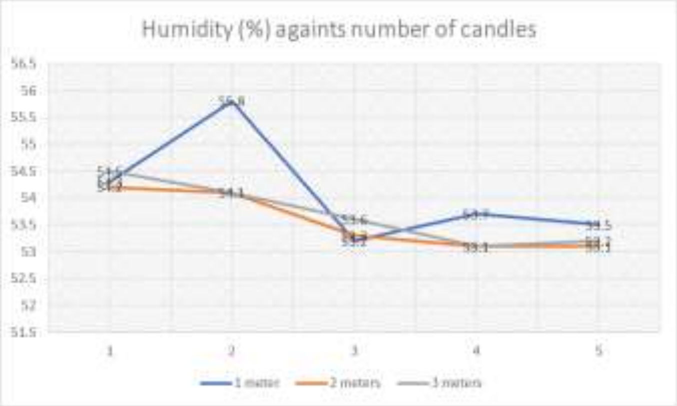
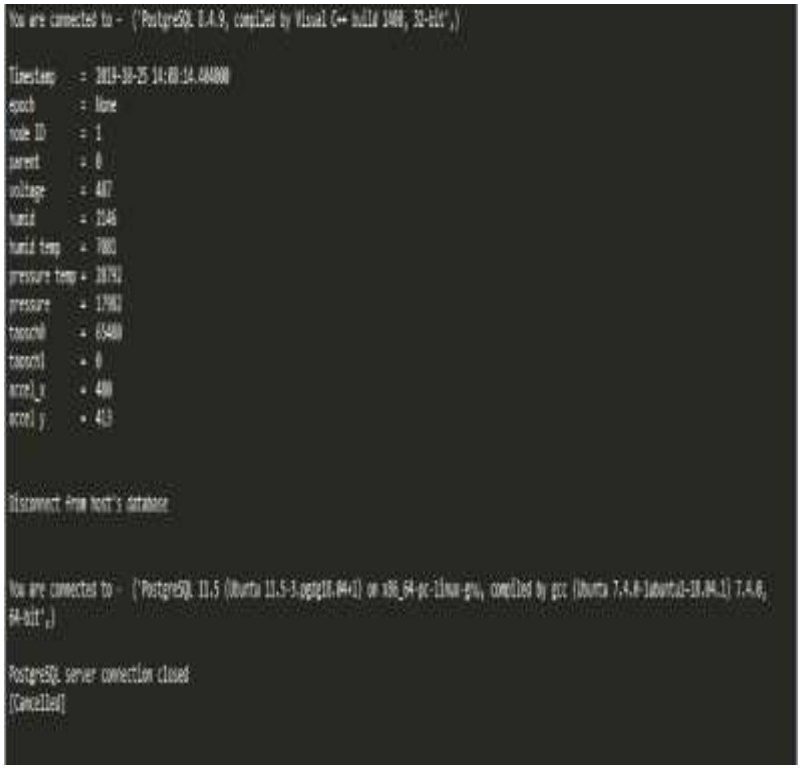


Figure 7 Humidity Number of Candles

Figure 7 shown the line graphs are the results of experiment done using the mica2 sensor which involved the light sensor, temperature sensor, as well as the humidity sensor. The results vary from each other which show that the lux is the most efficient in showing drastic differences of the experiment whereas the other two showed not much difference between the tests done.

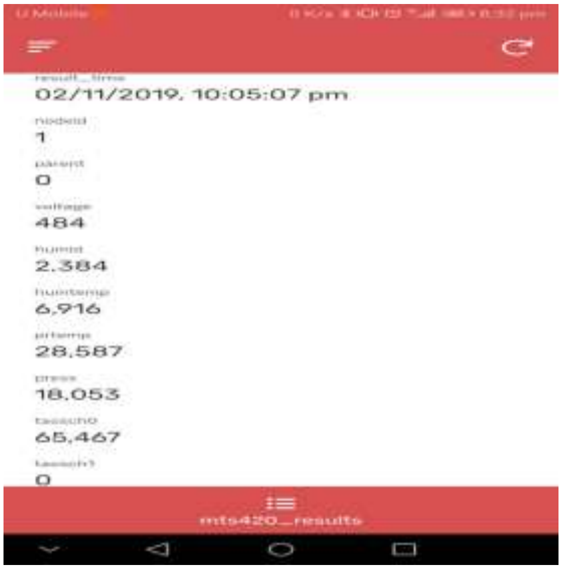
Our objective was to create a communication with the sensor where it can store the data gathered from the sensor into a cloud storage. Features in Mote. View should be able to allow remote acquisition of data but due to Xserve not being available anymore, Python was used. Python was used with specific coding which requires the PostgreSQL library implemented in the library in order to be able to use the set of lines to extract the data from the host PC which is connected with the gateway.



```
You are connected to -  ('PostgreSQL 8.4.9, compiled by Visual C++ build 1400, 32-bit',)

Timestamp     = 2019-10-25 14:08:14.484000
epoch         = None
node ID       = 1
parent        = 0
voltage       = 487
humid         = 2146
humid temp    = 7081
pressure temp = 28791
pressure      = 17982
taosch0       = 65480
taosch1       = 0
accel_x       = 400
accel_y       = 413


Disconnect from host's database


You are connected to -  ('PostgreSQL 11.5 (Ubuntu 11.5-3.pgdg18.04+1) on x86_64-pc-linux-gnu, compiled by gcc (Ubuntu 7.4.0-1ubuntu1~18.04.1) 7.4.0,
64-bit',)

PostgreSQL server connection closed
[Cancelled]
```

Figure 8 Output in Sublime Text 3

Figure 8 shows is the output printed on Python where it the connection made between the host PC's PostgreSQL database. The data is then extracted and displayed. Once the data is extracted, it connects with the subscribed server's PostgreSQL database and updates the table in the database. This allows the data stored in the database to replace old data with newly acquired data to avoid using so much space while updating the data. Figure 9 shows Mote.View GUI.



result_time
02/11/2019, 10:05:07 pm

nodeid
1

parent
0

voltage
484

humid
2,384

humtemp
6,916

prtemp
28,587

press
18,053

taosch0
65,467

taosch1
0

Figure 9 Mote.View GUI

Once the data is stored into the cloud server which is also a PostgreSQL database, another application which is Appsheet was used in order to create a simple phone app which could integrate the data stored inside the server's database and implement it into the mobile phone app which will display the data in real-time.

References

[1] S Vijayalakshmi and S Muruganand. "Internet of Things technology for fire monitoring system", International Research Journal of Engineering and Technology (IRJET), Vol: 04, no. 06, 2017.

[2] S Tiwari and S Bandopadhaya. "IoT Based Fire Alarm and Monitoring System", 3rd International Conference on Trends in Electronics and Informatics ,2019.

[3] S Sundar, P Vinothkumar, D Murthy and Kumar. Engineering Unit Conversion for Process Parameters", Indian journal of science and technology,Vol. 9, no.11, pp.1-4, 2016.

[4] K Sujatha, NP Bhavani, TK Reddy and KR Kumar. "Internet of Things for Flame Monitoring Power Station Boilers", IEEE Trends in Industrial Measurement and Automation (TIMA), 2017.

[5] K Sujatha, NP Bhavani and R Ponmagal. "Impact of NOx emissions on Climate and Monitoring using Smart Sensor Technology", International Conference on Communication and Signal Processing (ICCSP), 2017.

[6] B Stojkoska and D Davcev. "Web Interface for Habitat Monitoring using Wireless Sensor Network", Wireless and Mobile Communications, International Conference, 2009.

[7] BR Stojkoska. “Data Compression for Energy Efficient IoT Solutions”, Telfor, 2017.

[8] W Si, M Hashemi, L Xin, D Starobinski and A Trachtenberg. TeaCP: a Toolkit for Evaluation and Analysis of Collection Protocols in Wireless Sensor Networks”, IEEE International Conference on Microwaves, Communications, Antennas and Electronic Systems (COMCAS 2013), 2013.

[9] C Seiber, D Nowlin, B Laldowski and ME Tolentino. “Tracking Hazardous Aerial Plumes using IoT Enabled Drone Swarms”, IEEE 4th World Forum on Internet of Things (WF-IoT), 2018.

[10] F Saeed, A Paul, A Rehman, WH Hong and H Seo. “IoT-Based Intelligent Modeling of Smart Home Environment for Fire Prevention and Safety”, J. Sens. Actuator Netw., Vol. 7, no.1, 2018.

[11] NF Raun. Smart Environment using Internet of things(IOTS), IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016.

[12] K Padmanabh, L Malhotra, AM Reddy V, A Kumar, S Kumar V and S Paul. “MOJO: A Middleware that Converts Sensor Nodes into Java Objects”, Computer Communications and Networks (ICCCN), IEEE Xplore, 2010.

[13] WO Oduola, N Okafor, O Omotere and L Qian. “Experimental Study of Hierarchical Software Defined Radio Controlled Wireless Sensor Network”, IEEE Trends in Industrial Measurement and Automation (TIMA, 2017.

[14] K Muhammad, S Khan, M Elhoseny, SH Ahmed and SW Baik. “Efficient Fire Detection for Uncertain Surveillance Environment”, IEEE Transactions on Industrial Informatics, Vol. 15, no. 5, pp. 3113 - 3122 2019.

[15] C.Jothi Kumar R.Karthikeyan, Improved Reputation Systemfor Wireless Sensor Networks (WSNS),International Journal Of Innovations In Scientific And Engineering Research (IJISER),Vol.1,no.3,pp.191-195,2014.

16] J Lim and K Wong. “Wireless Sensor Networks: ThePotential Use of Received Signal Strength in Power Transmission Control for the MICA2”, Engineering e-Transaction, University of Malaya, Vol. 1, no.2, pp.9-13, 2006.

[17] WL Lee, A Datta and R Cardell-Oliver. “Network Management in Wireless Sensor Networks Conference: to appear in Handbook on Mobile Ad Hoc and Pervasive Communications”, American Scientific Publishers, 2006.

[18] H Larthani, A Zrelli and T Ezzedine. “On The Detection of Disasters: Optical Sensors and IoT Technologies”, Conference: International Conference on Internet of Things, Embedded Systems and Communications (IINTEC), 2018.

[19] R.Deepan S.Senthil Kumar,"A Energy-Efficient Probabilisitic Coverage Protocol in Wireless Sensor Networks",International Research Journal of Multidisciplinary Science & Technology (IRJMRS),Vol.1,no.7,pp.407-412,2016.

[20] RK Kodali and S Yerroju. “IoT Based Smart Emergency Response System for Fire Hazards”, IEEE 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), 2017.

[21] M Khadem, G Stamatescu and V Sgarciu. “Wireless Measurement Node for Dust Sensor Integration”, Sensorcomm, The Sixth International Conference on Sensor Technologies and Applications, 2012.

[22] Kaliski. “TinyOS Laboratory Development”, Thesis, California Polytechnic State University, Sensor networks, 2005.

[23] A Imteaj, T Rahman, MK Hossain, MS Alam and S Ahmad. “An IoT based Fire Alarming and Authentication System for Workhouse using Raspberry Pi 3”, IEEE International Conference on Electrical, Computer and Communication Engineering (ECCE), 2017.

[24] V Gokul and S Tadepalli. “Implementation of Smart Infrastructure and NonInvasive Wearable for Real Time Tracking and Early Identification of Diseases in Cattle Farming using IoT”, IEEE International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2017.

[25] S Duangsuwan, A Takarn and P Jamjareegulgarn. “A Development on Air Pollution Detection Sensors based on NB-IoT Network for Smart Cities”, IEEE 18th International Symposium on Communications and Information Technologies (ISCIT), 2018.

Biographies

Rostam Salleh His a Lecturer at University Kuala Lumpur- Malaysia France Institute, Bandar Baru Bangi, Selangor, Malaysia. His received Bachelor Eng Degree in Elect. Electronics Eng. University Malaya, Kuala Lumpur, Malaysia (2001), Master of Science Mechatronics Robotics, Signal and Systems IUUM (2019), Phd student in Mech. Eng. Control and Robotics in UKM. Malaysia (2020). He has published papers in International Journal and Conference. A member of Malaysian Automation and Control Engineering (MACE) and Malaysia Board of Technologist (MBOT). His research interests area an expertise in Sensor Network, Image Processing, ROS, Mobile Robotics and Robotics.

Sallehuddin Mohamed is an Associate Professor at the Department of Mechanical and Manufacturing Engineering, Universiti Kebangsaan Malaysia. He obtained the BEng degree in Manufacturing Systems Engineering from the University of Leeds, UK, in 1993, MSc degree in Mechatronics from the University of London (Kings College London) in 1996 and the PhD degree in Electronics and Electrical Engineering from the University of Southampton, UK, in 2006. His field of specialization is in control, mechatronic and cyber physical systems. His current research interests include autonomous robotic systems, magnetic gear systems, assistive technologies for people with disabilities and computer algebra applications in control systems.

Rizauddin Ramli received the Bachelor's Degree in Mechanical Engineering from Kyoto University, Japan, in 1997, Master's Degree in Mechanical System Engineering and the PhD Degree in Manufacturing systems from Gifu University, Japan in 2005 and 2008, respectively. He is currently an Associate Professor with the Department of Mechanical and Manufacturing, Faculty of Engineering and Built Environment, National University of Malaysia (UKM). His research interest's area of expertise in Robotics & Intelligent Manufacturing Systems.