Journal of Engineering Science and Technology
Special Issue on SIET2022, May (2022) 12 - 23
© School of Engineering, Taylor's University

MOBILE GAS MONITORING SYSTEM WITH IOT APPLICATION

MUHAMMAD ZULKHAIRI BIN CHE MI, ALVIN YAP CHEE WEI,
VINESH THIRUCHELVAM*, KRISHNA RAVINCHANDRA

School of Engineering, Asia Pacific University of
Technology and Innovation, Kuala Lumpur, Malaysia
*Corresponding Author: dr.vinesh@apu.edu.my

Abstract

The main aim of this paper is to develop a mobile gas monitoring system with IoT application to help in detecting hazardous gases that could cause health problems. In this proposed method, a GUI using Grafana integrated with Node-Red as MQTT broker and InfluxDB as the database was developed to demonstrate the proposed system. The system performance is evaluated by testing the data transmission rate, notification system performance, distance for gas detection and surrounding condition. It is observed that the integrated approach of these three software gives a better results and GUI compared to previous research for real-time monitoring. Based on the test done on data transmission rate, an average of 1.127 seconds is needed to transfer data from the input to the GUI. This process goes from the sensors – cloud – Node-RED – InfluxDB – Grafana. On the other hand, the time taken to transmit a notification once the threshold set exceeded only needs average of 5.03 seconds. Overall, this paper achieved the aim which to build a mobile gas monitoring system with IoT application and GUI interface. The low-cost budget is also commendable. However, in the future with upcoming technologies, future researchers can increase the accuracy of the calibration for each gas sensor, use different approach with different integration of software and as well as create a mobile phone application for better user-friendly interface

Keywords: Arduino, InfluxDB, Internet of things, Mobile gas monitoring system, Node-red.

1. Introduction

In this era of globalization where competition is everywhere, Malaysia is included in this race where currently Malaysia ranked 25th out of 140 others in the Global Competitiveness Report 2019-2020 released by the World Economic Forum. This growing in economy and population has their impact towards the surrounding environment such as deforestation to create more space, waste generation from increasing of population, emission of greenhouse gases and ultimately pollution. This paper will be focusing on the air pollution since in March 2019 there was a huge case that broke Malaysian news related to hazardous gas in Johor Bahru due to improper disposal.

Mobile gas monitoring system with IoT application is a project that is used to monitor the specified gases while providing early warning for authorized personnel to take action to contain any spread of hazardous gases. This system is essential when it comes to a place that uses or exposed to hazardous gases which concerns the people. This mainly refer to the industrial zone that uses chemicals in their manufacturing process. Gases are highly volatile which needed careful and trained personnel to handle them. With or without our knowledge, these gases may spread and could cause respiratory or skin disease that would be lethal to living things.

2. Literature Review

2.1. Microcontroller

Based on the previous research, various kind of microcontroller is used. However, the most common trend in the usage of microcontroller for this project is Arduino Uno. Based on reference [1, 3, 8, 12] all uses Arduino Uno in their hardware programming as Arduino Uno was the most compatible microcontroller for their system.

2.2. Sensors

Sensors that commonly used for previous papers also varies according to the budget costing and functionality of the proposed system. For example, [8] uses CO and NO_2 sensor to measure the gas concentration which has higher cost while [2] uses multiple gas sensors consists of PM2.5 and DHT22 sensors for the system. [7] uses BME680 as the gas sensors while this sensor also able to detect temperature, humidity, and pressure of the surrounding area.

2.3. Software

Overall, the software used by the previous researchers have one thing in common which is using a free open-source software. The software either used for programming only or with GUI features. Some of the commonly used software to display GUI from the previous research is using Blynk and Thinkspeak. Both of this GUI software have their own features which affects the selection of the software based on one's system implementation.

Even so, some of the researchers directly upload their data into their personal computers. This leads to limited usage for their system. Researchers [9] wishes that their system would have the online monitoring features in the future. Figure 1 shows the example of GUI from Thinkspeak based on researchers while Fig. 2 shows the example of GUI from Blynk.

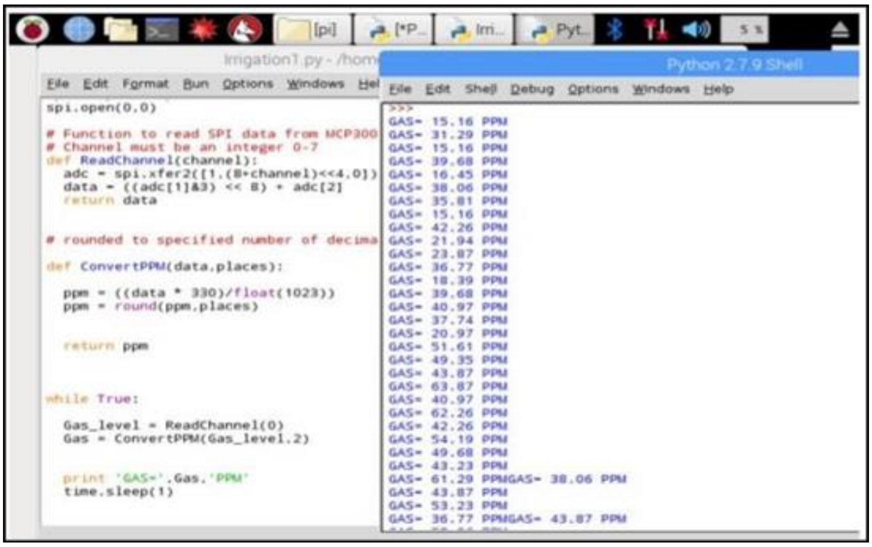

Fig. 1. Thinkspeak GUI example.

Fig. 2. Blynk application example.

3.Proposed Methodology

The proposed system for this Mobile Gas Monitoring System with IoT Application consists of hardware and software implementation system that integrates between hardware and software of the system. The inputs of the microcontroller come from the collection of data of the gas sensors. MQ-135 is used to detect Toluene, MQ-138 to detect Benzene, MQ-2 to detect Methane and MQ-136 to detect Hydrogen Sulfide. Calibration for each of the gas sensor will be done before deploying the system.

Figure 3 shows the overall block diagram for the proposed. After that, each sensor will record and send the data to the microcontroller which in this case is the

NodeMCU ESP32.After that process in the hardware, the data will be transmitted to Node-RED using MQTT broker. Depending on the flow nodes constructed using Node-RED, GUI will be displayed accordingly. Other than that, the features that had been set using Node-RED will start to compare the received value from the hardware with the threshold set in the programming. An automated notification system either via e- mail or Short Message System (SMS) could be send to the user to alert them regarding the current gas concentration.

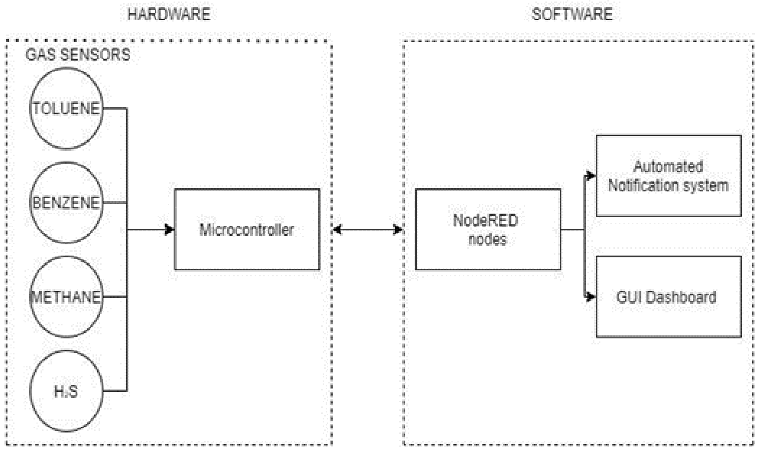

Fig. 3. Proposed block diagram of the system.

4. Program Design

The final design of this program is put into flowcharts and block diagram. Figure 4 shows the overall block diagram for the final design of the Mobile Gas Monitoring System with IoT Application. Power supply is introduced in this block diagram as the LiPo battery in use. The power supply is connected to the Step-Down Module as the supply voltage input into the microcontroller exceeds the operating voltage and this is worrisome. The Step-Down Module will regulate the voltage for the whole system. Vin of the microcontroller will be the power supply for all the gas sensors.

4 MQ-series gas sensors will act as the input for the whole system which is connected to the microcontroller directly following the respective pin as in the programming. MQ-2 is used to detect Methane, MQ-135 is to detect Toluene, MQ-138 to detect Benzene and MQ-136 to detect Hydrogen Sulphide. Calibration for each of the gas sensor will be done accordingly before deploying the system. After that, each sensor will record and send the data to the microcontroller.

After that process in the hardware, the data will be transmitted to Node-RED using MQTT broker. Depending on the flow nodes constructed using Node-RED, Node-RED will process the data and store them in InfluxDB. Next, Grafana which is used to display the GUI will use the database from InfluxDB and portrays the GUI in real-time data. Other than that, the features that had been set using Node-RED will start to compare the received value from the hardware with the threshold set in the programming. An automated notification system via e-mail will be send to the user to alert them regarding the current gas concentration. This combination of Node-RED, InfluxDB and Grafana is famously used because they are a free open-source software.

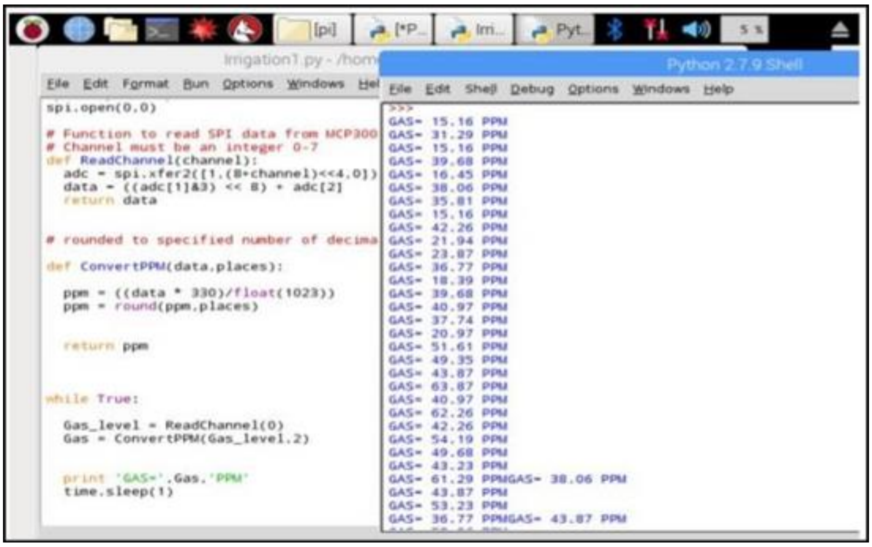

Fig. 4. Block diagram of the final design of the system.

Figure 5 shows the schematic design for the microcontroller used. This design is connected as such it will be interconnected with Figs. 6 and 7. ESP32 is the main brain to execute all the programming for this given research where it has Wi-Fi connection capabilities with operating voltage of 3.3 V. This Wi-Fi connection is used to update all the data to the cloud through Node-RED.

Figure 6 shows the design for Step-Down Module connection. This module is used to regulates the voltage from the power supply to the system. Input 1 and 2 of the modules is connected to a pin-header because this pin-header is to be used to connect the external power supply. Since this research is using a rechargeable LiPo battery, the battery can be disconnected once its low in power to charge. Output 4 of this module will be connected to the Vin of ESP32 as shown in Fig. 5.

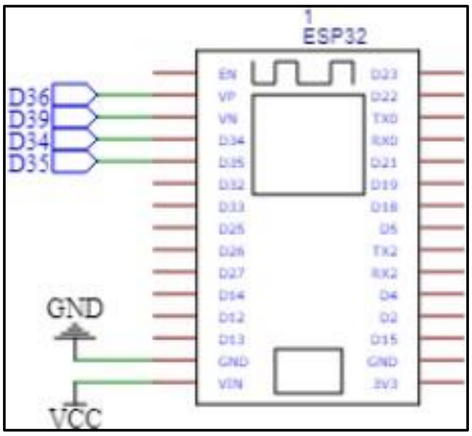

Fig. 5. Schematic diagram for NodeMCU ESP32.

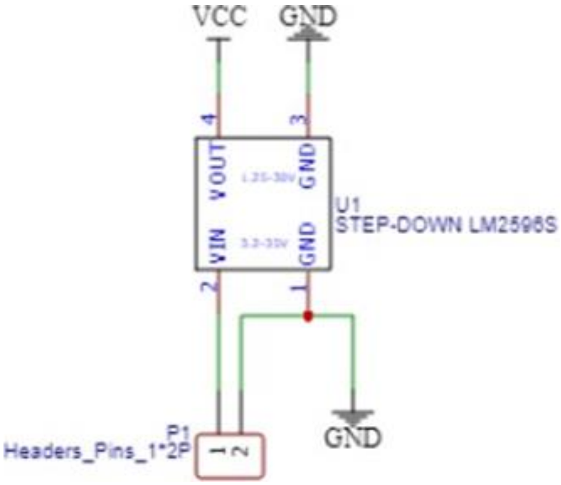

Fig. 6. Schematic diagram for step-down module.

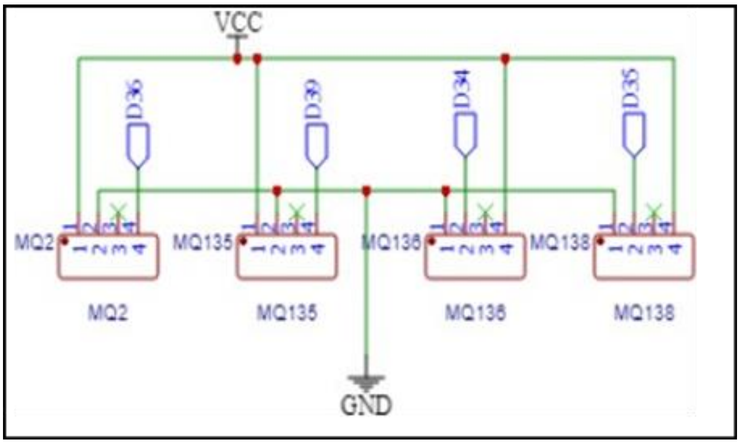

Fig. 7. Schematic diagram for gas sensors.

Figure 7 shows the wiring diagram for all the MQ-series gas sensors. Each of the gas sensors consists of 4 wiring pins. They are VCC, GND, A0 and D0. For this research, D0 pinout will not be in use as the data that is being capture are in analog form. D0 in Fig. 5 is represented under pin 3 and the connection for pin 3 is therefore ignored. For MQ2 and MQ135, pin 2 is connected to the ground while for MQ136 and MQ138 it is pin 1. The VCC pin for MQ2 and MQ135 is represented under pin 1 while for MQ136 and MQ138 is under pin 4. The analog data pin for MQ2 and MQ135 is at pin 4 while for MQ136 and MQ138 is at pin 2. The connection for these pins is very important when designing for the fabrication of PCB as correct techniques need to be use since fabrication of PCB takes some time to be done. Other than that, if the connection is wrong and the components is still mounted on the PCB it will damage the components. The analog pin is connected to the microcontroller in Fig. 7 accordingly.

Figure 8 shows the custom PCB design made for this paper with its arrangement. The power supply and gas sensors are connected using the pin headers as this is part of the innovation to make the components for plug and play purpose. Since there are many MQ-series gas sensors, user can simply disconnect the current gas sensors and connect a different gas sensor according to the needs.

Figure 9 shows the flowchart for the final design based on the execution of the whole system in a single scan cycle. This flowchart includes the processes from the hardware and software. Before starting the system, the power supply is ensured to be connected to the system, without power supply the system cannot run. As soon as the power supply is connected, the microcontroller will start to initialize all the inputs and outputs. For the gas sensors, a minimum 20 seconds of preheat timing needs to take place before the gas sensor can operate at its optimum point based on the data sheet provide by the supplier. In the meantime, the microcontroller will begin to connect with the system server. Once this process completed, the gas sensors will start to scan and deliver the analog value of gas concentration. Several parameters will be taken into account and calculated in the equation set in the programming.

After that, the system will begin to connect to the cloud for data transfer. If this process succeeds, the data will be stored in the database, but if the connection fail, the system will reattempt connection to the cloud until success. Not to fret, even if the system cannot connect to the cloud, the data will still be stored in the microcontroller until they can be sent to the cloud. Therefore, there will be no data loss during this connection. The only issue if the system is not connected to the cloud is that the GUI display will be delayed. Moving on, after the data is send to the database, this data will be read by Grafana, and the data will be displayed accordingly at the GUI. The last part of this whole system would be the notification system. A condition had been set in the programming at Node-RED where after certain value when the gas concentration exceeds the threshold, a notification will be sent to a particular email at a given intervals. In the email from Node-RED, user will be notified that certain gas is under dangerous level of concentration and appropriate action needs to be taken based on the gas concentration. Overall, this concludes the whole process for this system in each cycle.

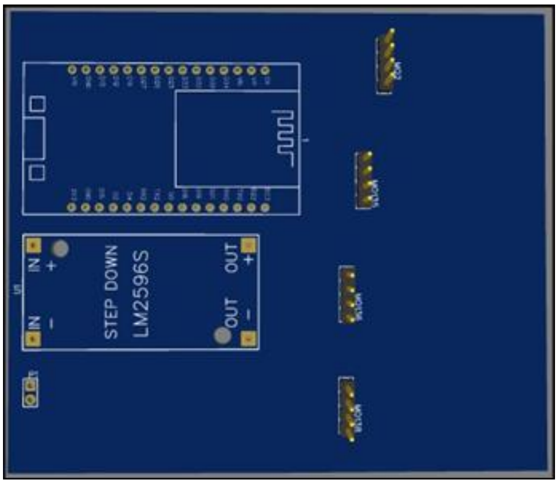

Fig. 8. Top view of PCB design.

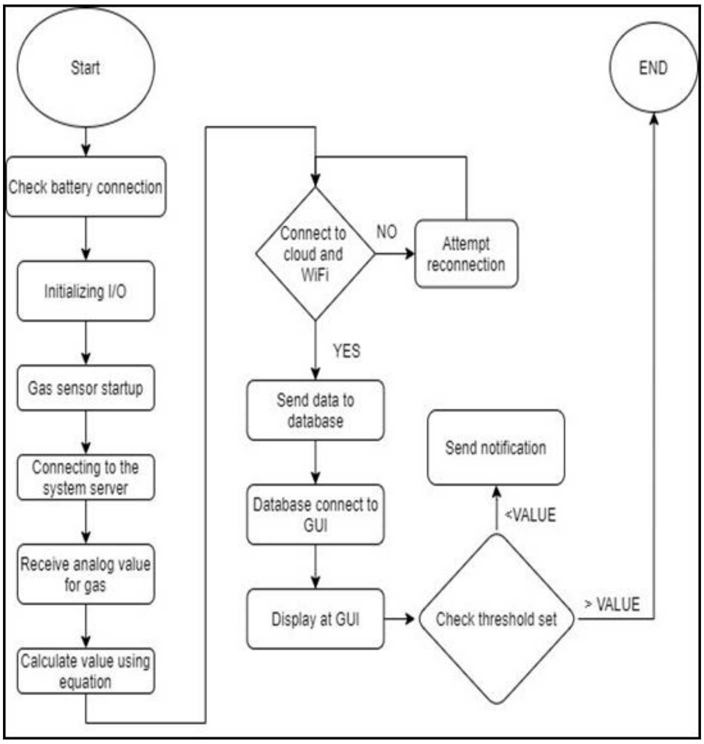

Fig. 9. Flowchart of the whole program.

5. System Implementation

The final design implementation for the system has a PCB board inserted inside a 3D printed casing as in Fig. 10. The board is integrated and ready to be used. Figure 11 shows the final GUI design for the research which will be accessible to everyone. This dashboard comprises all the individual gases graphs, real-time data, and overview of the graph difference for all the graphs. The good thing about using Grafana is that the refresh rate for the graphs can be adjust to the minimum of 5 seconds. Other than that, the user can specify the timeline where user wants to view the graphs and data at that period. In other words, this GUI can also be the database for the user to view the data from 5 years' time to 1 second period.

Fig. 10. Customize 3D printed casing and final hardware design.

Fig. 11. Final GUI for the proposed system.

6. Results and Analysis

A few tests had been carried out to check the efficiency and accuracy of this proposed system. The results for this testing are recorded and tabulated in the table below.

6.1. Data transmission rate

This test was carried out to determine the time taken for the data to be transmitted since the input is given. The result of this test shows how fast this system can keep up with the real- time condition. An average time will be taken over three sets of tests. The input of this test would be the time when the gas sensor is exposed to certain gas while the output of this test is the time when the GUI shows reaction to the input gas.

Table 1 shows the time taken for the data from three test to be transmitted along with the reading of gas concentration value. The gas concentration value is taken from two perspective to calculate the percentage error and data accuracy. The average duration for the data to be transmitted is about 1.127 second. The gas reading direct from the hardware and the gas reading at the GUI has a percentage error of 2.7% which makes the accuracy of this system is at 97.3% for the data transmission.

Table 1. Data transmission rate.

Tests	Input Time (HH:MM:SS:SSS)	GUI Display Time (HH:MM:SS:SSS)	Duration (SS:SSS)	Gas concentration from Arduino	Gas Concentration from Grafana (PPM)
1	14:53:46:580	14:53:47:660	1:80	3.127	3.018
2	14:57:23:500	14:57:24:600	1:100	3.776	3.638
3	14:58:31:300	14:58:32:500	1:200	3.527	3.492
Average			1.127	3.477	3.383

6.2. Notification system performance test

This test was carried out to determine the time taken for notification to be received by the user after the threshold had exceeded. The result from this test will give us an overview on how reliable this system is in ensuring the user is well updated on the dangerous level of gas concentration. The input of this test would be the time when the threshold had exceeded its limit while the output of this test is the time when the email notification is received by the user.

Based on Table 2, an average value for the duration whereby the email is delivered is taken. An average of 5.03 seconds is needed for the notification system to alert the user based on three tests. The duration for sending and receiving the email notification is analyze and conclusion made is that the internet connection may also give impact to the duration for the email sent.

Table 2. Time taken to send notification.

Tests	Threshold Exceed Time (HH:MM:SS:SSS)	Notification Receive Time (HH:MM:SS:SSS)	Duration (SS:SSS)
1	03:08:52:43	03:08:55:82	3.39
2	03:17:21:03	03:17:28:69	7.66
3	03:23:17:11	03:23:21:23	4.12
Average			5.03

6.3. Power consumption test

This last test required two or more type of battery for testing. However, with the resources available, two different type of battery is used to carry out this test. They are Lithium Polymer battery while the other one is alkaline battery. Both this battery has about the same voltage storage. The input for this test would be the start time for the system operation while the output will be the cut-off time for the system. The only reason why the system will start to cut-off is because the input voltage from the battery is insufficient to the operating voltage for the whole system. The system uses a rechargeable LiPo battery for the first test and non-rechargeable alkaline battery for the second test. The specification for the battery is listed in Table 3.

Table 3. Specification for the battery used.

Type of Battery	LiPo	Alkaline
Starting Voltage	8.4V	9V
Ending Voltage	4.38V	4.39V
Start Time	12:40 am	3:00 am
End Time	2:08 am	4:57 am
Duration	92 mins	117 mins
Charging Time	63 mins	N/A

Following the information listed in Table 3, the duration and voltage difference should be focused. As for the LiPo battery, every minute it consumed 0.0437 Volts until the voltage drop to 4.38 V and the system cut-off meanwhile for the alkaline battery, every minute it consumed 0.0394 Volts until the initial voltage drop to 4.39

V and the battery cut-off. The percentage difference in voltage consumption for these two batteries is at 9.15%. The voltage consumed by alkaline battery over time is lesser compared to the LiPo battery. Plus, the duration for the system to cut-off with alkaline battery is longer compared to the LiPo battery. However, the alkaline battery is not rechargeable while the price for an alkaline battery is about the same as the price of a rechargeable LiPo battery.

7. Conclusion

As a conclusion, it could be concluded that Mobile Gas Monitoring System with IoT Application has been successfully designed, constructed, implemented, developed, tested, and analysed with data transfer accuracy at 97.3% for the overall system has been achieved and the average time taken to transmit the data from sensor to the GUI is only 1.127 seconds, which is much at a commendable speed considering the amount of data and process that the system goes through. All the objectives were achieved during the progress of this project meeting the requirements needed. The limitation for this paper is that user needs to have access to technology that can view web browser in order for them to view the GUI. Other than that, the notification system could go under spam message if too many emails are sent under short period of time.

Acknowledgments

The authors would like to thank everyone who had directly and indirectly contributed ideas towards the process of completing this research. Not to forget, Asia Pacific University of Technology and Innovation for the accessibility that was provided throughout the whole research progress. Special thanks to the supervisor, Dr. Alvin Yap Chee Wei who had given valuable suggestion throughout this project.

References

1. Anandhakrishnan, S.; Deepesh, N.; Rakesh, K.; Sampath, K.; and Gayathri, S.N. (2017). IOT based smart gas monitoring system. *IOSR Journal of Electrical and Electronics Engineering*, 82-87.
2. Chanthakit, S.; and Rattanapoka, C. (2018). MQTT based air quality monitoring system using NodeMCU and Node-RED. *Proceedings of the Seventh ICT International Student Project Conference*. Nakhonpathom, Thailand, 1-5.
3. Divyasree, D.; and Bakar, G.A. (2019). Gas monitoring system using arduino. *International Journal for Research in Applied Science & Engineering Technology*, 7(IV), 1803-1810.
4. Elvitigala, C.S.; and Sudantha, B. (2017). Machine learning capable, IoT air pollution monitoring system with upgradable sensor array. *Proceedings of the International Symposium on Advanced Intelligent System*. Daegu, South Korea, 1-6.
5. Girish, S.V.; Prakash, R.; and Ganesh, A.B. (2016). Real-time remote monitoring of indoor air quality using Internet of Things (IoT) and GSM connectivity. *Advances in Intelligent Systems and Computing*, 394, 527-533.

6. Sai, K.B.K.; Mukherjee, S.; and Sultana, H.P. (2019). Low cost IoT based air quality monitoring setup using arduino and MQ series sensors with dataset analysis. *Procedia Computer Science*, 165, 322-327.
7. Penkov, S., Taneva, A. And Petrov, M. (2019). On-line monitoring system with LoRaWAN. *IFAC-PapersOnLine*, 52(25), 533-538.
8. Saikumar, C.V.; Reji, M.; and Kishoreraja, P.C. (2017). IOT based air quality monitoring system. *International Journal of Pure and Applied Mathematics*, 117(9), 53-57.
9. Yang, S.; Liu, Y.; Wu, N.; Zhang, Y.; Svoronos, S.; and Pullammanappallil, P. (2019). Low-cost, arduino-based, portable device for measurement of methane composition in biogas. *Renewable Energy*, 138, 224-229.
10. Oltean, S.-E. (2019). Mobile robot platform with arduino uno and raspberry Pi for autonomous navigation. *Procedia Manufacturing*, 32, 572-577.
11. Zakaria, N.A.; Abidin, Z.Z.; Harum, N.; Hau, L.C.; Ali, N.S.; and Jafar, F.A. (2018). Wireless internet of things-based air quality device for smart pollution monitoring. *International Journal of Advance Computer Science and Applications*, 9(11).
12. Sai, P.Y. (2017). An IoT based automated noise and air pollution monitoring system. *International Journal of Advanced Research in Computer and Communication Engineering*, 6(3), 419-423.