Multidisciplinary Applied Research and Innovation Vol. 3 No. 1 (2022) 381-389
© Universiti Tun Hussein Onn Malaysia Publisher's Office

MARI

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/mari
e-ISSN :2773-4773

Air Pollution Monitoring System Using IoT

M.H. Mohd Sofri[1], M.D.H. Sulaiman[1], N.H. Jefree[1], N.F. Adan[1]*

[1]Electrical Engineering Department, Centre for Diploma Studies,
Universiti Tun Hussein Onn Malaysia, 84600 Muar, Johor, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/mari.2022.03.01.044
Received 30 September 2021; Accepted 30 November 2021; Available online 15 February 2022

Abstract: Air pollution levels are rising year after year in Malaysia. It is because Malaysia is a developing country, this is the case. Many developments result in numerous contaminants, particularly air pollution. For instance, there was a big case in Pasir Gudang, Johor, where pollution caused shortness of breath and vomiting [1], as well as the Covid-19 crisis, which was partly caused by poor air quality. Besides, fine particles are most harmful air pollutants to humans. They are easily transported from outdoor to indoor air [2]. As a result, we require systems that can detect and quantify air pollution. The major goal of using to create an Air Pollution Level Monitoring System using IoT is to identify air pollution levels in places with severe, moderate, and good air quality in order to combat the growing problem of air pollution. It must be accomplished in order to live a better and healthier life in the future. We propose a system that uses the Internet of Things to monitor and check air quality. This project is produced using Arduino software to collect data for the air pollution level monitoring system, which is then transferred directly to the Blynk application. The information gathered will be presented in the form of air quality and air danger ratings [3].

Keywords: Air Pollution, Arduino, Iot, Monitoring System, Blynk.

1. Introduction

According to a study on greenhouse gas emissions, long-term exposure to polluted air may be responsible for 15% of global COVID-19 deaths. According to previous studies, air pollution from vehicles and factories diminishes the life expectancy of every man, woman, and child on the planet by two years. Experts in Germany and Cyprus recently announced that they were able to forecast population mortality rates that might be linked to or aggravated by air pollution's impacts [4]. In order to prevent and improve the air pollution levels, an air pollution monitoring system could be very useful. Using IoT, Arduino software and gas sensor, an innovative air pollution monitoring system is proposed in this project. Wireless and long-distance use of this system allow to detect the air quality index (AQI) in a place within the radius and surrounding temperature. It will sound an alert if the air quality falls

**Corresponding author: nfaezah@uthm.edu.my*

below a specified threshold, indicating that there are significant amounts of dangerous substances in the atmosphere [5].

This system are used to collecting data and then transfer to smart devices such as smartphones and laptops/computers through internet via NodeMCU. The user could get transmitted data more simply with the Blynk application. The Internet of Things (IoT) is a concept in which data is collected and sent across a network by connecting equipment, machines, sensors, and gadgets to the internet [6]. It produces intelligence by allowing a network to detect, control, and programme itself and allows computer devices and equipment to connect directly, indirectly, or via the Internet using existing technologies. With the current state of COVID-19, IoT applications can be further broadened. The user could get transmitted data more simply with the Blynk application.

2. Methodology

2.1 Block Diagram of Overall System

Air pollution monitoring system using IoT is a system that makes it easier and safer for the user to know the ambient air quality and temperature in the chosen area. The system consists of two separate circuit, an Arduino controlled circuit and a NodeMCU controlled circuit as shown in Figure 1 and Figure 2 respectively. The Arduino circuit is used for local monitoring system while the NodeMCU is used for remote monitoring system. The NodeMCU circuit can monitor either temperature or air quality value. The IoT air pollution monitoring system will show the air quality in PPM on the LCD and as well as on Blynk so that it can be monitored very easily. Temperature is detected and monitored in the system. Blynk is a platform with iOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet. In addition, data obtained from the Blynk will be sent to email to facilitate a person or authority to take any action.

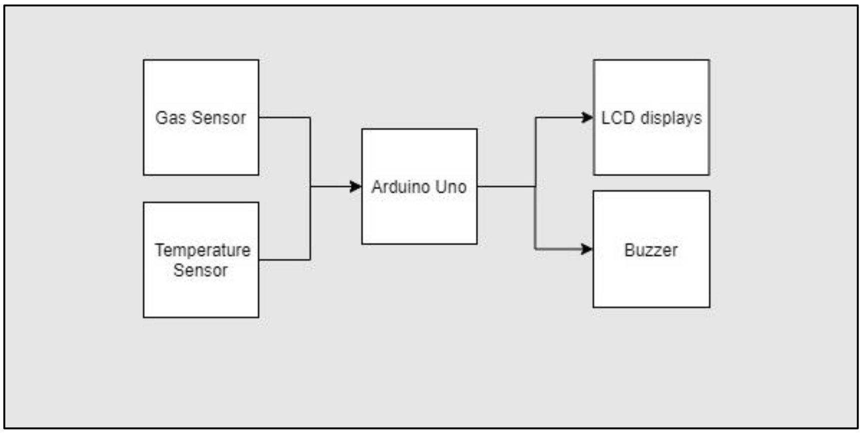

Figure 1: Block Diagram of Arduino Software

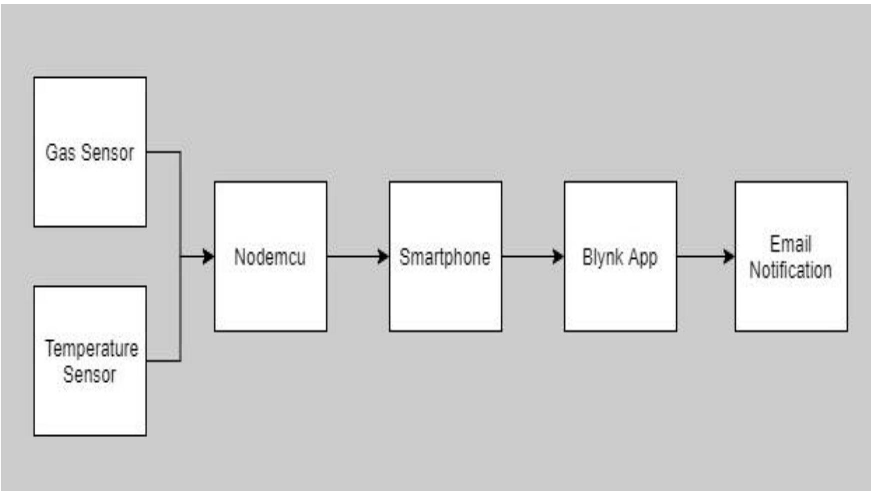

Figure 2: Block Diagram of NodeMCU

2.2 Flowchart

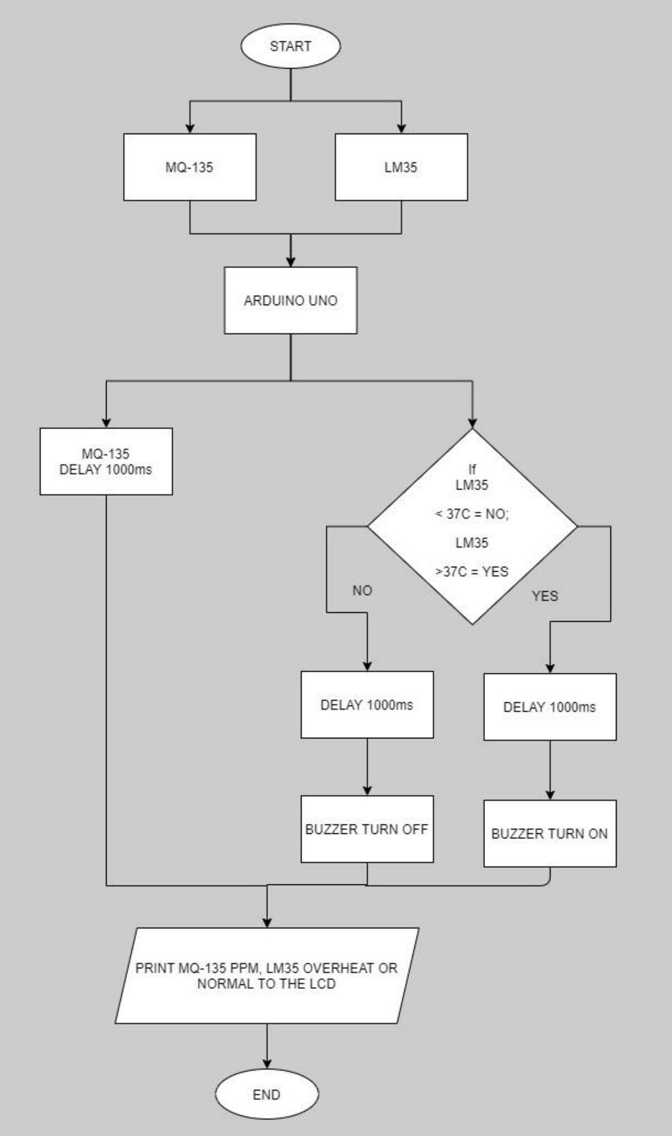

Figure 3: Flowchart of Arduino System

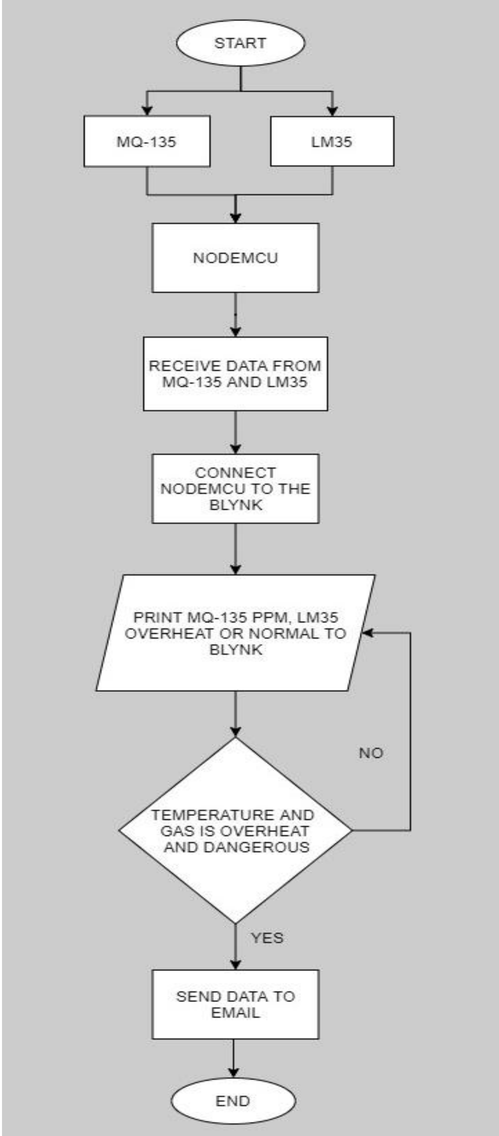

Figure 4: Flowchart of NodeMCU System

Flowchart in system coding is to represent how the system working and operate. Figure 3 and Figure 4 shows the flowchart of the system that explain the work of principle of the IoT air pollution monitoring system. The system will start with turn on the power supply and then the gas and temperature sensor will start detect the PPM and the temperature [7]. It is all connected to the Arduino and NodeMCU to transfer the data to the receiver which are the LCD and the buzzer. If the temperature is higher than 37℃ and the gas value is higher than 100 PPM, the LCD will display "OVERHEAT" and buzzer will on. Meanwhile, if the temperature is lower than 37℃ and the gas value is lower than 100 PPM, the LCD will display "NORMAL TEMPERATURE".

For remote monitoring system, NodeMCU will monitor either temperature or air quality value and send the data collected to be delivered in the Blynk app and the Blynk will send an email notification if the temperature reaches at a dangerous stage and it detect the air in a bad condition.

2.3 Electrical Design

Figure 5 shows the circuit diagram of IoT air pollution monitoring system. Proteus software is used to created this electrical design. A few modifications have been done like added and changed some components to make sure this project works properly and successfully without any problem occurs. A complete coding is uploaded to the Arduino to test the functionality of buzzer, LCD display and LED. The serial monitor is used to display the readings from the temperature and gas sensors. For Simulation, MQ-135 gas sensor has been replaced to MQ-6 gas sensor since there is no MQ-135 in Proteus. TMP36 is a temperature sensor. The NodeMCU circuit can't be simulated as the library for NodeMCU is not available in Proteus software. Hence, the actual testing for NodeMCU circuit is performed on the hardware.

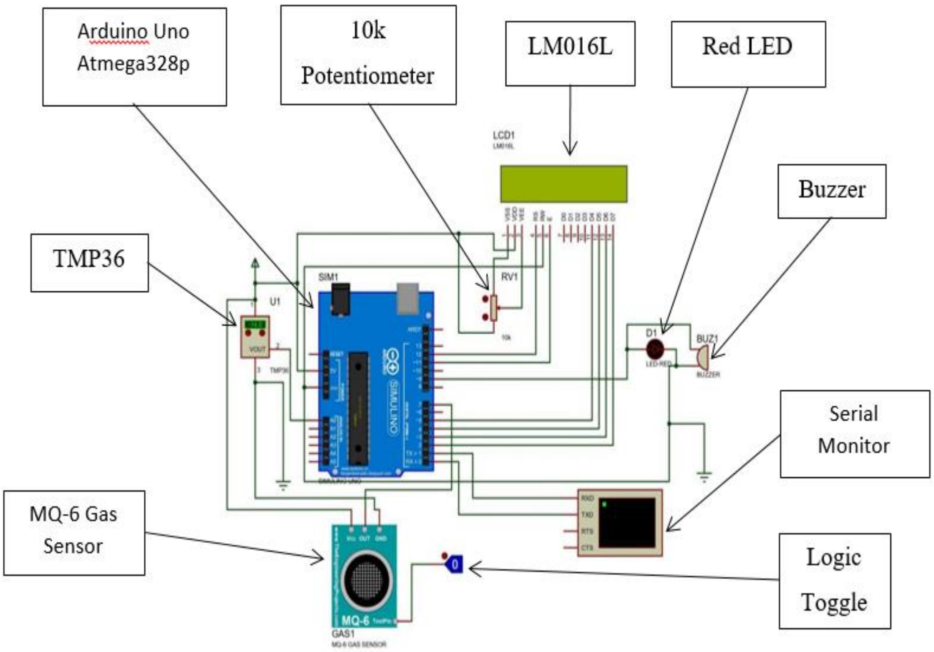

Figure 5: Electrical Design of IoT Air Pollution Monitoring System

3. Results and Discussion

3.1 Simulation Results

The output of TMP36 is analog and digital for the MQ-6. The Proteus library for MQ-6 component does not have an analog connection hence digital value is used to verify the circuit functionality. Table 1 summarizes the outcome of the simulation.

Table 1: Result of Simulation IoT Air Pollution Monitoring System

Temperature	Gas Sensor	Red LED	Buzzer
36.1°C	0	Turn OFF	Turn OFF
36.1°C	1	Turn ON	Turn ON
37.1°C	0	Turn ON	Turn ON
37.1°C	1	Turn ON	Turn ON

Figure 6 shows that LED and buzzer is OFF if TMP36 is below than 37.0℃ and MQ-6 is LOW. Figure 7 shows that LED and buzzer is ON if TMP36 is below than 37.0℃ but MQ-6 is HIGH. Figure 8 shows that LED and buzzer is ON if TMP36 is higher than 36.9℃ but MQ-6 is LOW. Figure 9 shows that LED and buzzer is ON if TMP36 is higher than 36.9℃ and MQ-6 is HIGH. New value of temperature and PPM will show after 1 second delay. This outcome show that all components and circuits in the simulation function successfully.

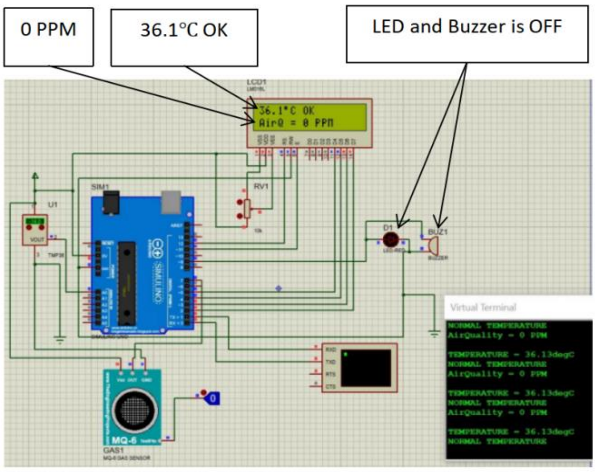

Figure 6: Normal Temperature and Good Air Quality

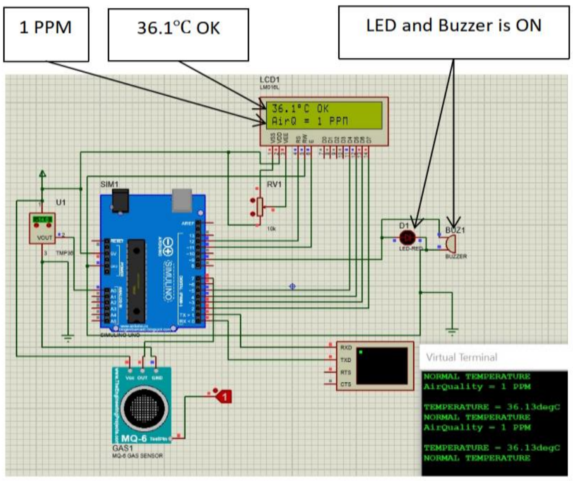

Figure 7: Normal Temperature and Bad Air Quality

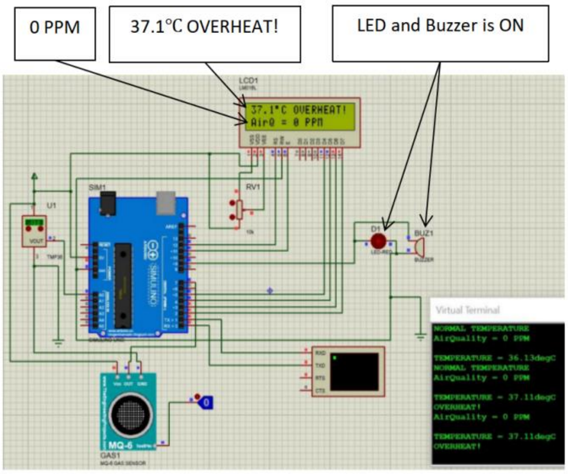

Figure 8: High Temperature and Good Air Quality

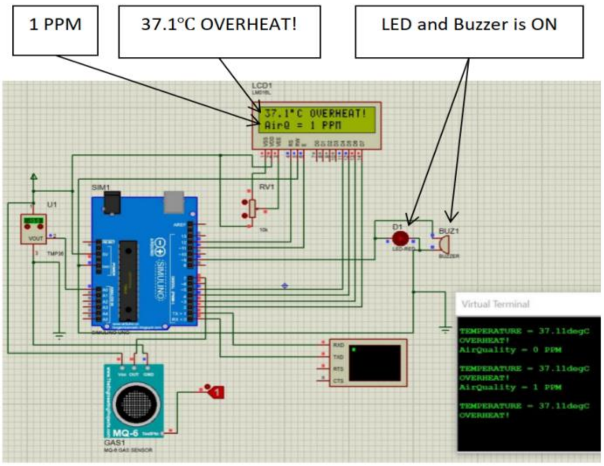

Figure 9: High Temperature and Bad Air Quality

3.2 Prototype Result for Arduino Circuit

The output of TMP36 and MQ-6 is analog value. Table 2 summarizes the outcome of the tested prototype.

Table 2: Result of Prototype IoT Air Pollution Monitoring System

Temperature	Gas Sensor	Buzzer
32.23℃	76 PPM	Turn OFF
32.23℃	144 PPM	Turn ON
37.71℃	60 PPM	Turn ON
38.35℃	126 PPM	Turn ON

Figure 10 shows on LCD that buzzer is OFF if LM35 is below than 37.0℃ and MQ-135 is below 100 PPM. Figure 11 shows on LCD that buzzer is ON if LM35 is below than 37.0℃ but MQ-135 is higher that 100 PPM. Figure 12 on LCD shows that buzzer is ON if LM35 is higher than 37.0℃ but MQ-135 is higher that 100 PPM. Figure 13 shows on LCD that buzzer is ON if LM35 is higher than 37.0℃ and MQ-135 is higher that 100 PPM. New value of temperature and PPM will show after 1 second delay. For this prototype, it is suitable for Local monitoring and does not need Blynk to use this prototype. This prototype will give two types of outcome temperature and air quality monitoring.

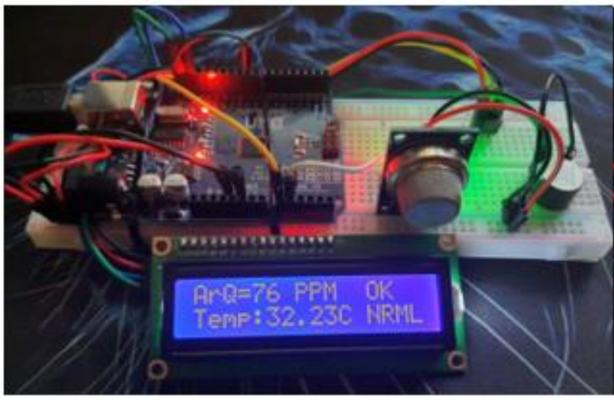

Figure 10: Normal Temperature and Good Air Quality

Figure 11: Normal Temperature and Bad Air Quality

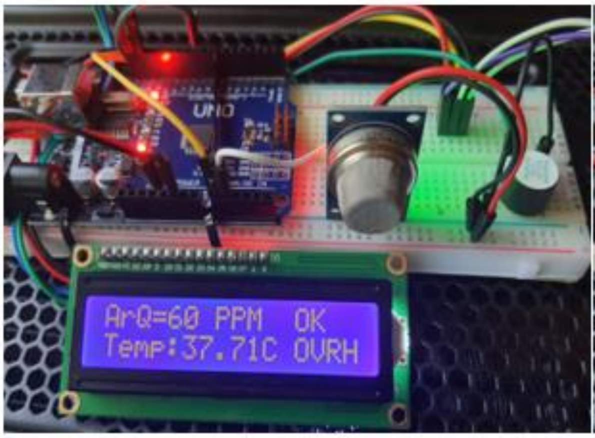

Figure 12: High Temperature and Good Air Quality

Figure 13: High Temperature and Bad Air Quality

3.3 Prototype Result for NodeMCU Circuit

For this Blynk simulation, it is suitable for remote monitoring use to monitor temperature and air quality result because it uses NodeMCU which has wireless capability to transmit data to Blynk apps on the smartphone. Figure 14 shows that the temperature that have been detected from LM35 with using Blynk as the output. From the Blynk shows the temperature as 37.06℃ and the buzzer will ON because when the temperature above 37℃ is OVERHEAT. The code that will be use to check the temperature is:

```
tempValue = analogRead(tempPin);
float mv = (tempValue/1024)*3300;
float celsius = mv/10;
```

As for Figure 15 shows that the air quality that have been detected from MQ135 with using Blynk as the output. From the Blynk shows the air quality as 101 PPM and the buzzer will ON because when the air quality above 101 PPM is DANGEROUS. The code that will be use to check the air quality is:

```
sensorValue = analogRead(0);
Serial.print("AirQua=");
Serial.print(sensorValue, DEC);
Serial.println(" PPM");
```

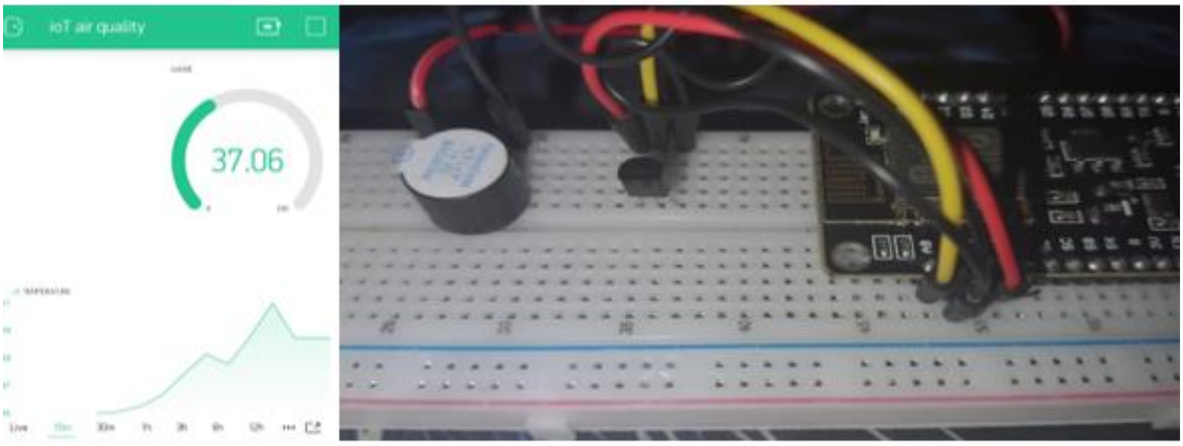

Figure 14: Sense Temperature Hardware and Blynk Result

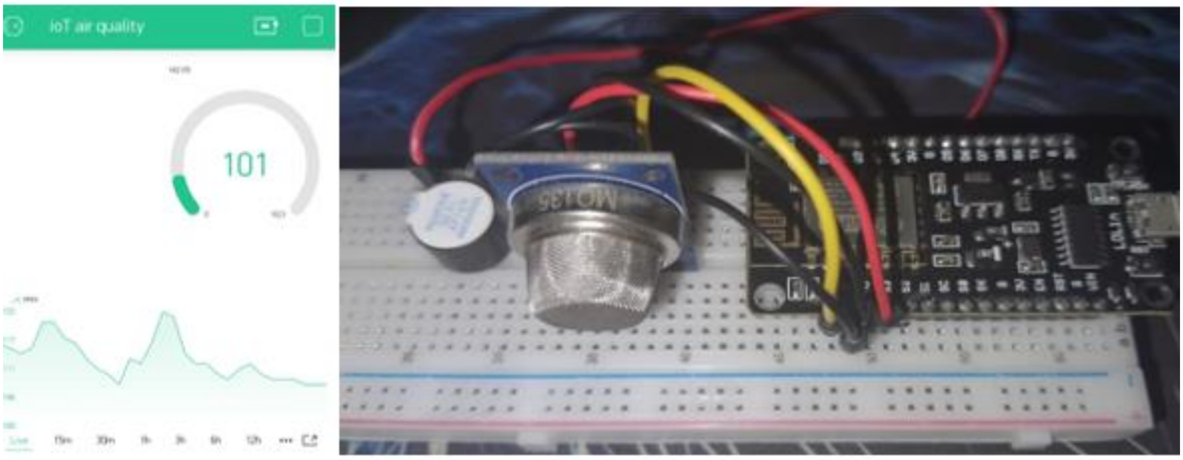

Figure 15: Sense Air Quality Hardware and Blynk Result

3.3 E-Mail Notification

For this E-mail notification, use Blynk for the main application and give a push notification to the e-mail that have been targeted if one of the temperature (LM35) and air quality (MQ135) exceed from the normal behaviour. Figure 16 and Figure 17 shows the notification sent to the email of intended user. The code that will be use to post the result of temperature from Blynk to email is:

```
Blynk.email("danialhafiz30012001@gmail.com", "ESP8266 Alert", "OVERHEAT! HIGHER THAN 37C");

Blynk.notify("ESP8266 Alert -> OVERHEAT! HIGHER THAN 37C");
```

The code that will be use to post the result of air quality from Blynk to email is:

```
Blynk.email("danialhafiz30012001@gmail.com", "MQ135 Alert", "DANGEROUS");

Blynk.notify("MQ135 Alert -> DANGEROUS");
```

4. Conclusion

This project has been successfully implemented using a simulation and hardware method. Projects conducted have demonstrated promising results, with Proteus simulations reacting to gas and temperature sensors and then applying them to hardware. The prototype then can communicate with the Blynk app using NodeMCU (wifi). When the system is turned on, the sensor will detect temperature and air quality values in the area being tested, and the information will be sent to the user via the Blynk app. Additionally, if the environment at the tested area has dangerous air quality, e-mail notifications from Blynk will be received. For Arduino circuit, the results shows on the LCD and buzzer. The temperature and the value of PPM is shown on the LCD and it triggered the buzzer when its exceeding the prescribed limit. This can be used to warn and notify consumers about the temperature and air quality in the location they intend to visit or any current residential area, so that they can take further action if any problems occur.

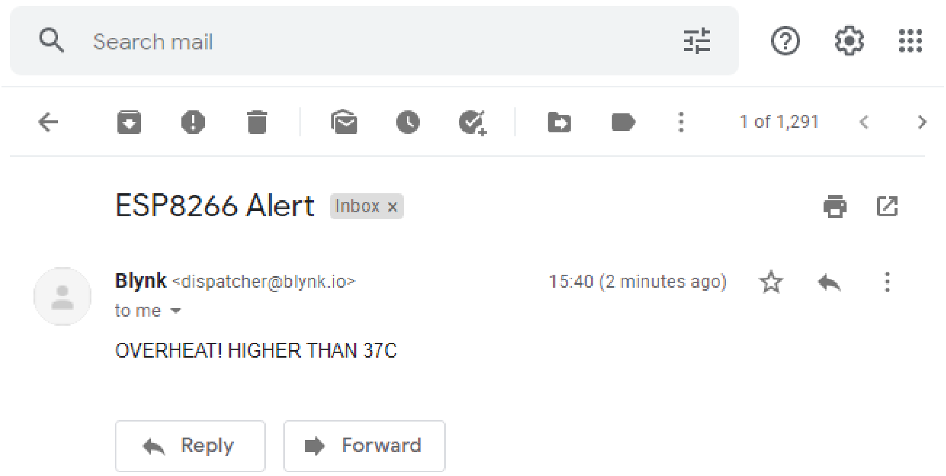

Figure 16: E-mail push notification will be sent when LM35 above 37℃

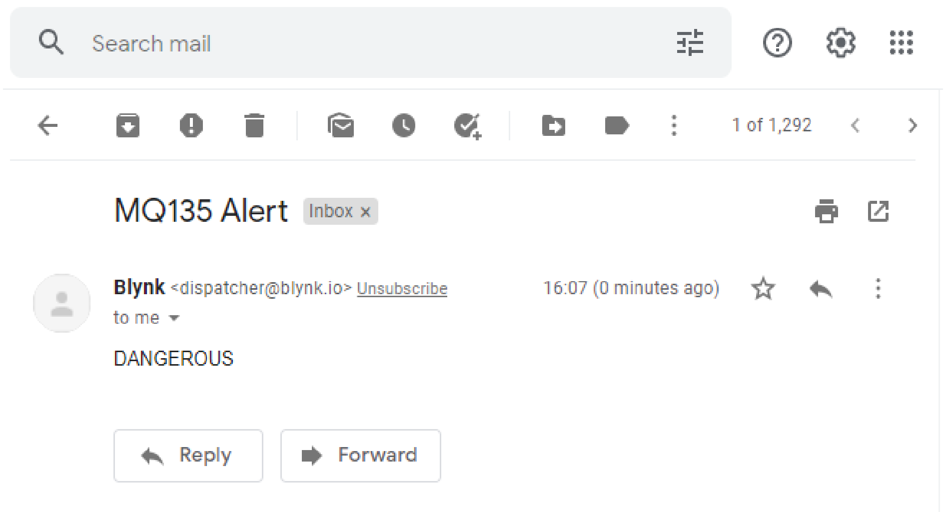

Figure 17: E-mail push notification will be sent when MQ-135 above 100PPM

Acknowledgement

The authors would like to thank the Centre of Diploma Studies, Universiti of Tun Hussein Onn Malaysia (UTHM) for its support.

References

[1] Bernama, "MalaysiaKini," Bernama, 28 June 2019. [Online]. Available: https://www. malaysiakini.com/news/481570.

[2] C. Jayapal et al., "IoT-Based Real Time Air Pollution Monitoring System," *International Journal of Grid and High Performance Computing*, 2019.

[3] online, "Air Pollution in Malaysia: Real-time Air Quality Index Visual Map," Aqicn. [Online].

[4] AFP, "Berita Harian Online," Berita Harian, 27 October 2020. [Online]. Available: https://www.bharian.com.my/dunia/eropah/2020/10/746666/pencemaran-udara-turut-menyumbang-kepada-15-peratus-kematian-covid-19.

[5] J. Waworundeng and Oktoverano Lengkong, "Indoor Air Quality Monitoring and Notification System," *Cogito Smart Journal*, vol. 4, no. 1, 2018.

[6] S. Dhingra, Rajasekhara Babu Madda, Amir H. Gandomi, Senior Member, IEEE, Rizwan Patan and Mahmoud Daneshmand, Life Member, IEEE, "Internet of Things Mobile–Air Pollution," *EEE INTERNET OF THINGS JOURNAL*, vol. 6, no. 3, 2019.

[7] S. N. Harsh et al., "IOT Based Air Pollution Monitoring System," *International Journal of Scientific & Engineering Research*, vol. 9, no. 2, 2018.