



DESIGN AND DEVELOPMENT OF AN IOT-BASED AIR QUALITY MONITORING SYSTEM FOR SMART CITIES

**Ramesh*¹, Satyam Kumar², Tapu Das³, Saqib Ahmed⁴, Mrs Saswati Behera⁵,
Dr Krishna Kumar P R⁶**

^{1,2,3,4}Students Dept of CSE,SEA College of Engineering & Technology,Bangalore-560049.

^{5,6}Faculty, Dept of CSE,SEA College of Engineering & Technology,Bangalore-560049.

Article Received: 07 December 2025

***Corresponding Author: Ramesh**

Article Revised: 27 December 2025

Students Dept of CSE,SEA College of Engineering & Technology,Bangalore-

Published on: 15 January 2026

560049.

DOI: <https://doi-doi.org/101555/ijrpa.3459>

ABSTRACT

Air pollution has emerged as a critical environmental and public health concern, particularly in rapidly urbanizing regions. Conventional air quality monitoring systems are often expensive, sparsely deployed, and lack real-time accessibility, limiting their effectiveness in proactive pollution management. This paper presents the **design and development of an IoT-based Air Quality Monitoring System** that enables continuous, real-time monitoring of ambient air quality parameters. The proposed system integrates low-cost gas and particulate sensors with a microcontroller-based embedded platform to measure key air quality indicators such as particulate matter (PM_{2.5}/PM₁₀), carbon monoxide (CO), nitrogen dioxide (NO₂), and volatile organic compounds (VOCs). Sensor data are transmitted wirelessly to a cloud server using IoT communication protocols, where the data are stored, processed, and visualized through a web and mobile-based dashboard. Threshold-based alerts are generated to notify users and authorities when pollutant levels exceed permissible limits. The system offers a scalable, cost-effective, and energy-efficient solution for urban air quality monitoring, enabling data-driven decision-making and public awareness. Experimental results demonstrate reliable real-time data acquisition, improved spatial coverage, and effective remote monitoring capabilities. The proposed system supports smart city initiatives and contributes to sustainable environmental monitoring and management.

KEYWORDS: Arduino Uno, Android, monitoring system, MQ135 Sensor, ThingSpeak.

1. INTRODUCTION

Human need air to breathe. In the breathing process, clean air is important to support the human body to stay healthy, while a polluted air can affect humans' body negatively. The human breathing system can't filter the air they want to inhale by free will. The air consists of Oxygen (O₂), but a human can also inhale dust particle, formaldehyde, pesticide, fungus and other dangerous things to human breathing system that can be found indoor.

Indoor air condition can be more harmful than outdoor air condition because there is more pollution source indoor and it's not as open as outdoor, so the air circulation is worse than the circulation outdoor. The indoor pollution can come from the condition of the building, the materials inside the building, or the activity of people inside. The humidity, heat, gas, air conditioner, chemical product, smoke, or even the floor material like woods can pollute the air [1]. Polluted air can affect humans' health [2]. The damage done by polluted air can be temporary, but some damage can be permanent [3]. It can cause headache, skin irritation, respiratory disorders and asthma. Some people won't be affected much by the polluted air, but there are also people whose health condition can be worsened by the polluted air.

The effects of polluted air need to be avoided and one of the alternatives is by using a system to monitor the air quality indoor. Based on the importance of indoor air quality, the researcher built a system which can monitor indoor air quality using microcontroller Arduino board and MQ135 sensors to detect the air quality with an interface of an Android application to get data and display the air quality, by means to raise the awareness to the user regarding any polluted air indoor.

The indoor air quality monitoring system is built by using Arduino and MQ135 sensors. Arduino is an electronic module which can be programmed. The input data read from MQ135 sensors, processed by Arduino (which connected to Ethernet Shield for internet connection) and then stored to IoT platform ThingSpeak. Sensor data stored to ThingSpeak channel which is already setup and then push the data to an Android application called AirQmon.

With the advances of information and communication technology triggered the modern equipment which can help human to get necessary information around its environment. Internet of Things (IoT) is evolved from time to time which integrates devices and supported human daily activities to be efficiently [4]. According to an article [5], every second there are 127 new IoT devices are connected to the web and during 2020, expert estimates 31 billion of

IoT devices. In this research, we build an IoT device to monitor the air quality which is integrated with an Android application called AirQmon.

2. LITERATURE REVIEW

2.1 R Air Quality

Air quality is a term which is related to how much pollution is present in the air. Good air quality means there is less air pollution, while poor air quality means there is more pollution [6]. Clean or unpolluted air is considered to be a basic requirement of human health. However, air pollution continues to pose a significant threat to the health of human worldwide [7]. The concentration of air pollutant should be monitored to increase human awareness regarding the quality of air.

Indoor air quality is essential to maintain a person's health because humans are spending most of their time indoors. The air quality indoor can be polluted by many factors, such as Respirable Particulate Contaminants, Asbestos, Carbon Monoxide, Mercury, Volatile Organic Compounds, Formaldehyde, Indoor Allergens, Indoor Pesticides, Radon, Indoor Molds and Environmental Tobacco Smoke [8]. A monitoring system can add human awareness of the air quality inside the room.

There are levels of pollution that humans need to avoid based on the Air Pollution Index.

Table 1, shows the level of danger which is contained in air pollution [9].

Table 1 Air Pollution Index on CO and NO2.

Category	Range	Carbon Monoxide	Nitrogen Dioxide
<i>Good</i>	0-50	No effect	A bit smelly
<i>Moderate</i>	51-100	Changes in blood chemistry, but undetected	Smelly
<i>Not Healthy</i>	101-199	Cardiovascular increase in heart disease smokers	Smelly and transparent colour. Increased vascular reactivity on asthma patients
<i>Very Unhealthy</i>	200-299	Increased cardiovascular on heart disease smokers. The weakening will be shown clearly	Increased sensitivity on non-asthma and bronchitis patients
<i>Hazardous</i>	300 or more	Hazardous level to the exposed population	

2.2 System

The system is an entity, which is coherent with a boundary that is perceived around it to distinguish internal and external elements and to identify input and output relating to and emerging from the entity [10]. There are two types of system, that is the subsystem and supersystem. The subsystem is a system inside another system; it means that there is more than one level of the system. A supersystem is a rare case. If a system is a part of a bigger system, the bigger system is called supersystem.

System characteristics [10] consists of basic components (identifiable and moving parts of the system), interaction and structure, goals, behaviour and life cycle. The characteristic of a system is determined by the system elements, their properties and relations. The system elements have input, processor, and output.

2.3 Android

Android is a software stack for mobile devices which includes an operating system, middleware and key applications [11]. Android is popular in the electronic market because of its open-source and free development tools [12]. Android system has been developed continuously. The development introduces a new feature or bug fixes.

2.4 Microcontroller

Microcontrollers are small computing system used for low power and low memory purposes. A microcontroller consists of a microchip on a circuit board with read-write capabilities, memory, inputs and outputs [13]. A microcontroller is a little computer on a chip. The microcontroller has more than the first home computer had. It has a processor, a kilobyte or two of random-access memory (RAM) for holding data, a few kilobytes of erasable programmable read-only memory (EPROM) or flash memory for holding users' program and it has input and output pins. These input/output (I/O) pins link the microcontroller to the rest of the electronics [14].

Arduino is a popular AVR (Advanced Virtual RISC)-based microcontroller [13]. Arduino is a small microcontroller board with a universal serial bus (USB) plug to connect to users computer and several connection sockets that can be wired to external electronics such as motors, relays, light sensors, laser diodes, loudspeakers, microphones and more [14].

An Arduino board is a microcontroller development board that happens to be an independent open-source hardware design. It means that the design files for the printed circuit board (PCB) and the schematic diagrams are all publicly available, and everyone is free to use the

designs to make and sell their own Arduino boards. There are various components on an Arduino board such as power supply, power connectors, analogue input, digital connection, microcontroller and other components.

2.5 Sensor

The sensor is a tool that detects and converts a physical measurement unit (light, heat, motion, moisture, pressure, or a great number of other environmental phenomena) into an electrical measurement unit (human-readable display for future processing) [15]. To be called a sensor, it needs some of these requirements:

1. high sensitivity (in accord with the measurement unit),
2. not sensitive to another measuring unit,
3. sensor won't impact the object natures.

MQ135 is a sensor that can detect air quality. MQ135 uses Stannic Oxide (SnO₂) as its sensitivity material. When the target combustible gas exists, the sensors' conductivity increased and the gas concentration rise. MQ135 has a high sensitivity to NH₃, NO_x, alcohol, benzene, smoke, CO₂ and other harmful gases [16]. Gas Sensor from the MQ series use a small heater with an electrochemical sensor and are usually used indoors at room temperature. The calibration of MQ135 is preferred to use a known concentration of the measured gas [17]. In this research, we use three MQ135 sensors.

2.6 Sound Buzzer

A buzzer or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm device, timers and confirmation of user input such as a mouse click or keystroke. The buzzer is widely used in computers, printers, copiers, alarms, electronic toys, automotive electronic equipment, telephones, timers and other electronic products for sound devices [18]. The sound buzzer will be used to sound an alarm to notify people nearby whenever the limit of dangerous gas or dust exceeds the normal limit.

2.7 Ethernet Shield

Ethernet shield is a shield that can connect an Arduino device to the internet. Ethernet shield is widely used for local area network. It is a link-layer protocol in the TCP/IP stack. By using the Ethernet library to write sketches, the Arduino can be connected to the internet [19]. The researcher needs the ethernet shield to connect the system to the internet. Ethernet shield will

be connected to a router by using an RJ45 ethernet cable.

2.8 Internet of Things (IoT) and ThingSpeak™

ThingSpeak is an Internet of Things platform which can be collect and store sensor data in the cloud and develop IoT applications. The ThingSpeak IoT platform provides apps that let the user analyze and visualize the data and then act on the data. Sensor data can be sent to ThingSpeak from Arduino, Raspberry Pi, BeagleBone Black, and other hardware [20]. In this research, the device needs to connect to ThingSpeak to upload sensors data values and also to retrieve the sensor data values from ThingSpeak to the AirQmon Android application.

2.9 Programming Language

The programming language is a notation that is used to write a computer program. This language is divided into three levels, which are machine language, low-level language and high-level language [21].

The machine language is microinstruction or hardwire. The programs are long and difficult to understand. Besides that, it is highly machine architecture-dependent. The advantage of machine languages is that the process is quick and users don't need an interpreter or low-level language interpreter, such as macroinstruction (assembly). Just as machine language, a low-level language is also machine architecture-dependent. The process is fast, but the programs are long and difficult.

High-level languages have a similar structure with the human language so that it's easier to understand. This language is not machine architecture-dependent, but it needs compiler or interpreter. There are two programming languages which are used to build the system, as explained as follows.

1) Java Language

An android uses Java as its programming language. Java is an object-oriented programming language developed by Sun Microsystems. Java is easier to learn than most other object-oriented programming languages because it's a collection of the best parts of existing ones. Java language is standardized enough so that executable applications can run on any computer that contains a Virtual Machine (run-time environment). Java isn't slower than any other object-oriented language. Java can be compiled to native code and run as fast as any other program [22].

2) C Language

Arduino Integrated Development Environment (IDE) is used to upload the code to the device.

Arduino programming language is based on C language. C is a general-purpose programming language with features economy of expression, modern flow control and data structures, and a rich set of operators. C is neither a very high-level language nor a big one and is not specialized to any particular area of application, but its absence of restrictions and its generality make it more convenient and effective for many tasks than supposedly more powerful languages [23].

2.10 Related Research

There are several related types of research which support the research of Indoor Air Quality Monitoring System based on Arduino.

Table 2 Related Research of Air Quality System.

Related Research	Sensor	Board	Monitor System
Arduair: Air Quality Monitoring [24]	MQ5	Arduino Uno	Computer via USB
Air Quality Monitoring System based on Arduino Microcontroller [25]	MQ135, MQ2, MQ7, DHT11	Arduino Uno	Message via GSM Module, Computer via Zigbee Module, Bluetooth Module
Ambient Environmental Quality Monitoring Using IoT Sensor Network [26]	Ambient Light Sensor, DHT11, Sound Sensor, MG811, MQ7, MQ6, MQ3	Arduino Mega 2560	Thingspeak.com via Wi-Fi Module
Air Quality Monitoring and Notification System with IoT Platform [27]	MQ135	Wemos Board	Thingspeak.com, Blynk Application
<i>AirQmon: Indoor Air Quality Monitoring System Based on Microcontroller, Android and IoT</i>	MQ135	Arduino Uno and Ethernet Shield	Thingspeak.com, AirQmon (Android Application)

Besides the list of research related to air quality shown in Table 2, there are some previous works develop by the researcher related to detection and monitoring system using microcontroller and sensors [28-29]. These researches are basic ground fundamental in the design of AirQmon indoor air quality monitoring system.

3. SYSTEM DESIGN AND IMPLEMENTATION

System design related to how to build the *AirQmon: Indoor Air Quality Monitoring System*.

3.1. Conceptual Research Framework

The conceptual research framework is based on Prototyping Model [30]. The first phase is

communication to gather information required to develop the system by literature review. The second phase is Quick Plan and Modeling Quick Design to design how the system will be represented with the user interface of an Android application, decide the programming language, the software and components needed. The researchers also analyze the application function. The third phase is the Construction of Prototype. The researcher develops the system, by coding the application and builds the prototype of an indoor air quality monitoring system using Android Studio for Android application and Arduino IDE to program the Arduino board and the whole system. The fourth phase is Testing to deploys a test to improve the system.

3.2 Hardware System and Application Design

Figure 1, explains the steps generated to run the system. First, MQ135 measured the level of air quality in a room. Then, the Arduino read the data provided by sensors. If the gas level exceeds certain amount value which is hazardous, the Arduino sound the alarm through the sound buzzer. Arduino will then upload the data to ThingSpeak through Ethernet Shield. Then, the user can view the data by using the AirQmon application (Indoor Air Quality Monitoring System based on Android). The AirQmon retrieve the data sent by ThingSpeak and display it to the user smartphone.

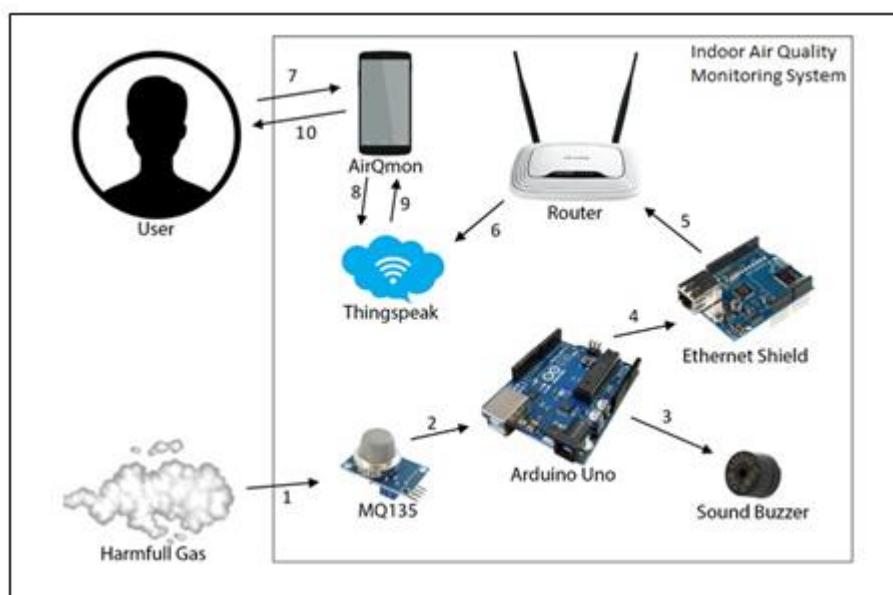


Figure 1 Hardware Conceptual System.

Figure 2 shows the steps done by the user to run the application. First, the user will need to open the AirQmon application. Then, AirQmon will retrieve the data from ThingSpeak. The

data retrieved then displayed to the user graphically. The user can then choose to reload the graph or just to display it.

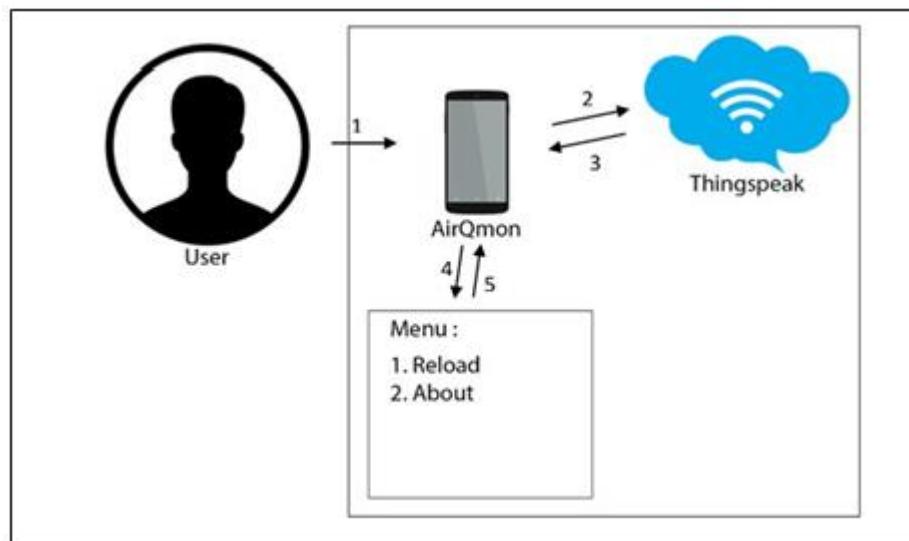


Figure 2 AirQmon Application Framework.

3.3 Hardware Implementation

Figure 3, shows the hardware schematic and components interconnection. Arduino board is associated with Ethernet Shield to provide an Internet connection. There are three MQ135 sensors which are used to detect the indoor air quality. The Arduino board is programmed to sense the air quality through the MQ135 sensors. When the air quality is detected in a polluted condition, the alarm buzzer turned on as a direct sound warning sign to the people inside a room or building.

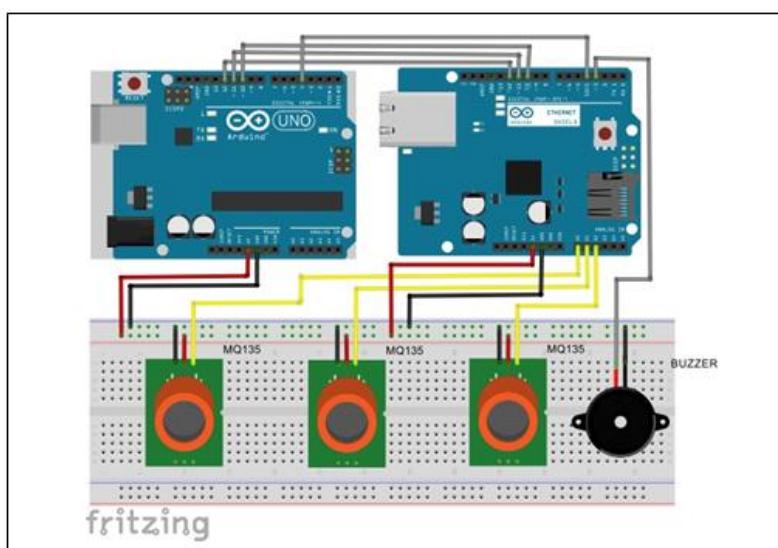


Figure 3 Hardware Schematic

3.4 AirQmon: Android Application Interface Implementation

Figure 4(a) shows the home interface of AirQmon application which with graphical implementation which retrieved from ThingSpeak that shows the sensors data on AirQmon application. There is also a refresh button to reload both the graphs data from the internet. The *about* button gave information about the application. Figure 4(b), shows the interface of AirQmon when the user gives input touch to point on the graph. The AirQmon application can show data from each MQ135 sensors.



Figure 4 AirQmon Application.

3.5 Testing Results

In this research, we use the Black Box method to test how the system works. The result of the testing is shown in Table 3.

Table 3 Final Testing Results of Hardware.

Components	Expected Results	Final Results
Gas Sensor 1	The MQ135 gas sensor could detect the air quality of the room	Success
Gas Sensor 2	The MQ135 gas sensor could detect the air quality of the room	Success
Gas Sensor 3	The MQ135 gas sensor could detect the air quality of the room	Success
Sound Buzzer	The Sound Buzzer could sound an alarm when the amount of gas reached a certain value	Success

Ethernet Shield	The Ethernet Shield could connect the Arduino Uno to the Internet through the router	Success
-----------------	--	---------

Table 4 Final Testing Results of AirQmon Application.

Features	Expected Results	Final Results
Home Screen	The user could see the air quality sensor graph, reload the graph and see about button	Success
About Dialog	The user could see a description of the application	Success
Sensor Graph	User could data sent by Arduino graphically	Success
Store Data	Data read by the sensor could be stored in ThingSpeak	Success
Display Data	Data read by the sensor could be displayed in ThingSpeak	Success

4. CONCLUSION

As the results of this research of Air Quality Monitoring System Based on Microcontroller, Android and IoT, it can be concluded as follows:

1. the device can upload sensor data to ThingSpeak IoT platform, fetch and display sensor data to AirQmon application interface, which is installed on user Android smartphone,
2. the system can monitor air quality and can ease the user on accessing the information regarding the indoor air quality by using AirQmon application.
3. the system can sound the alarm to alert the user when the air quality is detected in a hazardous condition,
4. though it's still in a prototype model, the system could potentially be used as a tool to raise human awareness of air quality in certain indoor places and environment.

REFERENCES

1. E. Flynn, P. Matz, A. Woolf and R. Wright, "Indoor Air Pollutants Affecting Child Health", A project of the American College of Medical Toxicology (ACMT), 2000
2. M. Kampa and E. Castanas, "Human health effects of air pollution," Environmental Pollution, vol. 151, no. 2, pp. 362-367, 2008.
3. US Environmental Protection Agency, Indoor Air Quality (IAQ), 2020. [Online].
4. Available: <https://www.epa.gov/indoor-air-quality-iaq/introduction-indoor-air-quality>.
5. A. S. Abdul-Qawy, P. J. Pramod, E. Magesh and T. Srinivasulu, "The Internet of Things (IoT): an overview", in International Journal of Engineering Research and Applications,

(IJERA). vol. 5, pp. 71-82, Dec. 2015.

6. G.D. Maayan, “The IoT Rundown For 2020: Stats, Risks, and Solutions”, 2020. [Online]. Available: <https://securitytoday.com/Articles/2020/01/13/The-IoT-Rundown-for-2020.aspx?Page=2>
7. R. Williams, V. Kilaru, E. Snyder, A. Kaufman, T. Dye, A. Rutter, A. Russell and H. Hafner, Air Sensor Guidebook, 2014.
8. World Health Organization, “Particulate Matter, Ozone, Nitrogen Dioxide and Sulfur Dioxide”, Air quality guidelines, 2006.
9. E. Flynn, P. Matz, A. Woolf and R. Wright, “Indoor Air Pollutants Affecting Child Health”, A project of the American College of Medical Toxicology (ACMT), 2000.
10. Badan Pengendalian Dampak Lingkungan (BAPEDAL), “Pedoman Teknis Perhitungan dan Pelaporan serta Informasi Indeks Standar Pencemar Udara”, 1998.
11. C. Mele, J. Pels and F. Polese, “A Brief Review of Systems Theories and Their Managerial Applications”, Service Science, vol. 2, no. 1-2, pp. 126-135, 2010
12. N. Gandhewar and R. Sheikh, “Google Android: An Emerging Software Platform for Mobile Devices”, International Journal on Computer Science and Engineering, pp. 12-17, February 2011.
13. L. Ma, L. Gu and J. Wang, “Research and Development of Mobile Application for Android Platform”, International Journal of Multimedia and Ubiquitous Engineering, vol. 9, no. 4, pp. 187-198, 2014.
14. D. Diakopoulos and A. Kapur, “Hiduino: A Firmware for Building Driverless USB-MIDI Devices using the Arduino Microcontroller”, in New Interfaces for Musical Expression, Oslo, 2011.
15. S. Monk, “Programming Arduino”, McGraw-Hill Companies, 2012.
16. F. Chandra and D. Arifianto, “Jago Elektronika”, 1st ed., A. Natik, Ed., Jakarta: PT Kawan Pustaka, 2010.
17. MQ135 Technical Data, “MQ135 Semiconductor Sensor for Air Quality Control”, 2017.
18. V. Kalra, C. Baweja, D. S. and S. Chopra, “Influence of Temperature and Humidity on the Output Resistance Ratio of the MQ-135 Sensor”, International Journal of Advanced Research in Computer Science and Software Engineering, vol. 6, no. 4, pp. 423-429, 2016.
19. “5V Buzzer Module”, [Online]. Available: <http://tinkbox.ph/learn/tutorial/how-use-5v-buzzer-module>. [Accessed 2016].
20. Arduino, 2018 “Getting Started with the Arduino Ethernet Shield and Ethernet Shield 2”.

[Online]. Available: <https://www.arduino.cc/en/Guide/ArduinoEthernetShield>.

21. The MathWorks, Inc., "ThingSpeak Documentation," 2018. [Online]. Available: <https://www.mathworks.com/help/thingspeak/>. Suprapto, K. T. Yuwono, T. Sukardiyono and A. Dewanto, "Bahasa Pemrograman", R. A. Avianti, Ed., Direktorat Pembinaan Sekolah Menengah Kejuruan, 2008.
22. K. Främling, "Introduction to Java Programming", CiteSeerX, 2003.
23. B. W. Kernighan and D. M. Ritchie, "The C Programming Language", 2nd ed., Bell Telephone Laboratories, 1978.
24. V. Chaudhry, "Arduair: Air Quality Monitoring", International Journal of Environmental Engineering and Management, vol. 4, no. 6, pp. 639-646, 2013.
25. N. Kaur, R. Mahajan and D. Bagai, "Air Quality Monitoring System based on Arduino Microcontroller", International Journal of Innovative Research in Science, Engineering and Technology, vol. 5, no. 6, pp. 9635-9646, 2016.
26. A. Djajadi and M. Wijanarko, "Ambient Environmental Quality Monitoring Using IoT Sensor Network", Internetworking Indonesia Journal, vol. 8, no. 1, pp. 41-47, 2016.
27. J. Waworundeng and O. Lengkong, "Indoor Air Quality Monitoring and Notification System with IoT Platform", Cogito Smart Journal, vol. 4, no. 1, pp. 94-103, 2018. [Online]. Available: <http://cogito.unklab.ac.id/index.php/cogito/article/view/105>
28. J. Waworundeng, "Design of Smoke and Flame Detection Systems Based on Sensors, Microcontrollers and IoT", Cogito Smart Journal, vol. 6, no. 1, pp. 117-127, 2020. [Online]. Available: <http://cogito.unklab.ac.id/index.php/cogito/article/view/239>
29. J. Waworundeng, M. Kalalo, D. Lokollo, "A Prototype of Indoor Hazard Detection System using Sensors and IoT," presented at 2nd ICORIS, virtual conference, Airmadidi, Indonesia, Oct. 27-28, 2020.
30. R. S. Pressman and B. R. Maxim, "Process Models", in Software Engineering A Practitioner's Approach, 8th ed. New York: McGraw-Hill Education, 2015, ch.4, pp. 45-47.