

The Air and Sound Pollution Monitoring System using Internet of Things and Cloud Based Data Analysis

¹Deshinta Arrova Dewi, ¹Edward Lai Poh Yen and ¹Ponkoodalingam Kannan

¹Faculty of Information Technology and Science, INTI International University, Nilai, Negeri Sembilan, Malaysia.

Corresponding author: Deshinta.ad@newinti.edu.my, pon.kannan@newinti.edu.my

Abstract

The Air and Sound Pollution Monitoring System (ASPMS) is a new concept that can be used in many sectors like industry, home living, road and transportation, environmental surroundings, etc. Currently, the ASPMS has been developed as separate entity whereby the users need to activate two systems in order to use them. With Internet of Things (IoT) technology, both entities are combined to promote user friendliness. In this paper, not only both entities are combined, the proposed ASPMS is linked to cloud based service that enable users to see the graphical view of the sensor readings. The overall system receives data from relevant sensors and administered by a portable processor Raspberry Pi. In this study, the important sensors are employed, like MQ135 that is used as gas sensor, MQ7 for measuring the CO₂ level, and DHT11 for measuring temperature and humidity. Altogether, the values are analysed to define the pollution level. The connection is established using MQTT broker (Message Queuing Telemetry Transport). This design ensures a real-time monitoring system upon the quality of air and sound levels. A dedicated website is developed to contain variety of data which can be used for important action and precaution of a sudden pollution, for example, wearing masks or automatically switch-on a connected purifier device to protect the air quality in a building. The ASPMS is implemented using Java language, connected with Thingspeak, and openHab for system integration. The result shows a real-time pollution data are able to captured, monitored and analyzed accurately as per requirement.

Keywords

IoT, Pollution Monitoring, Air and Sound pollution

Introduction

Nowadays, surfing Internet is never been easy, from every individual to big commercial businesses there are internet everywhere. The broadband internet service become one of the most used platform to access the internet. The broadband internet service provides high speed access with lower costs compared to the old technologies such as DSL (Digital Subscriber Line) or satellite which is way too expensive.

In conjunction with the internet expansion around the world that is so fast and large, there is one concept called “Internet of Things (IoT)” (XMPP-IoT. 2017). The IoT technology enable data inputs from any hardware devices and brings a convenient to the users to experience new technology (PubNub. 2017). The term of IoT was invented by Kevin Ashton of Procter & Gamble in MIT in the year of 1999. Recently, the IoT technology is very popular and become one of major solutions to smart technologies (IoT Technology, 2017). The IoT connects almost everything and exchange data across devices for benefits of human life (Cisco. 2017). Hence, this paper attempts to address the most ongoing problem in many countries including Malaysia which is air and sound pollution.

The record shown that Malaysia experience periodic air pollution that comes from large scale forest fires from Sumatra Island in Indonesia. Besides, some activities like motor vehicle, open burning also industries pollution make situation worst. There is a need to identify the sudden changes of air pollution to assist human to emergency action (Schneider Electric, 2017). On top of this issue, environmental noise still remains a complex issue. Noise or sound pollution has been identified as number three in the world (kookye.com. 2018) as source of psychological disorder that gradually develop healthy implication like stress, sleeping disorder due to sound annoyance (Adafruit Learning System. 2018). By developing a system that integrate air and sound pollution, a healthy environment becomes an ultimate objective that can be used for smart city concept. The IoT technologies are selected as basis of system implementation due to its highly accessibility and low cost compared with others.

The idea of this paper is has been applied as a final year project for Bachelor in Computer Science (BCSI) in INTI International University. It consists of system analysis, design and implementation. The development of this system takes place in INTI International University Computer Lab in Nilai Campus, in the Faculty of Information Technology and Sciences (FITS). Through this system, users are able to monitor the air and sound level in real time manner and make action directly to protect own health and surroundings. The main objectives of this paper are:

1. To study the requirement of the air and sound pollution system.
2. To develop an air and sound pollution system using Internet of things and integrate it with cloud based service for a higher accessibility.
3. To perform testing and maintenance over the developed system.

Methodology

Requirement Gathering and Quick Design

Studying and understanding the requirements of air and sound pollution system were done using literature review comprises of newspapers (ReliaSENS 18-12, 2017), some documentaries (videos, photos, etc) that shows how Malaysia has overcome such pollutions over the years. Afterwards, some mini questionnaires are developed (UKEssays. 2017) to understand the need of a system that help users to identify such pollutions, increase the awareness of the important pollution monitoring and health consciousness. The raw accumulated data is processed to have a better view and understanding of what is required in the development.

The comparison with current application is also conducted to identify the good practice and the gap that can be improved by the ASPMS. Nonetheless, the user-friendliness is considered through the interface of current applications. Afterward, the system specification of ASPMS is defined, the functional and non-functional requirements are stated in order to achieve the research objectives. The overall flow process of the ASPMS is drawn for a better understanding for the overall system features as mentioned in the data collections.

Building and Refining the ASPMS

The ASPMS requires hardware and software development (Watchdog 1250, 2017). The hardware are sensors, mini controller Raspberry Pi (FrontPage - Raspbian. 2017) (Raspberry Pi. 2017) (Hackaday. 2017), IoT protocol (Electronic Design. 2017) (NetBeans IDE 2017), and microphone. The hardware selected for the ASPMS development are: Personal Computer, Raspberry Pi, MQ135 sensor (Olimax. 2017), MQ7 sensor (AOSONG. 2017), DHT11 sensor (Maxim Integrated. 2015), Microphone and 4-Channel Relay (Maxim Integrated. 2014). The software to process all data that comes from the hardware are consists of the Java language (Auth0 - Blog. 2017) (Oracle/Jose Cruz. 2014), Thingspeak Cloud-Based Service, MQTT broker (Roxanne Joncas. 2017) and OpenHAB platform. The overall design is depicted in the following picture.

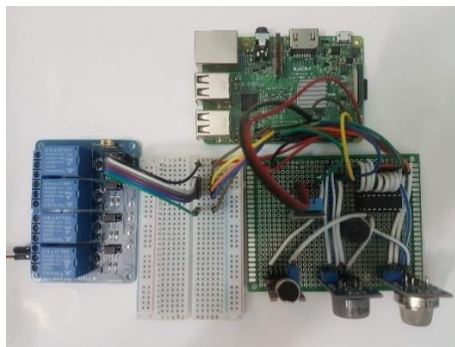


Figure 1. The hardware design of ASPMS

The ASPMS device is connected to a router that provide internet access, which allowing the ASPMS device to sends the data over the internet. This is important since the data that received from the sensors need to broadcast to Thingspeak, the cloud storage for data analysis. This stage onwards involves software development and configuration to integrate the overall system. The connection to Thingspeak as a cloud storage permits user to retrieve the sensor data via internet from any distant location, as long as internet connection is available. By having this feature, remote access and flexibility is delivered as what users stated in the requirement.

Thingspeak produce a friendly graphical diagrams that easy for users to see the live data from the sensors (Hackster. 2018). The examples of the graphical diagrams produced by Thingspeak are depicted in the following figures.

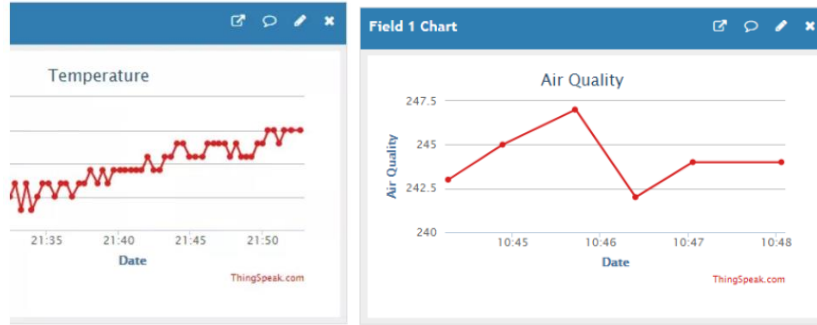


Figure 2. Thingspeak cloud based data analysis

Nonetheless, an openHAB platform need to configure as a service to integrate the overall ASPMS (openHABian, 2018). The Java Application Interface is required at this stage together with SSH (Secure Shell) connection to deliver a real-time data access and operation manipulation. The example of openHAB interface is depicted in the following figure.

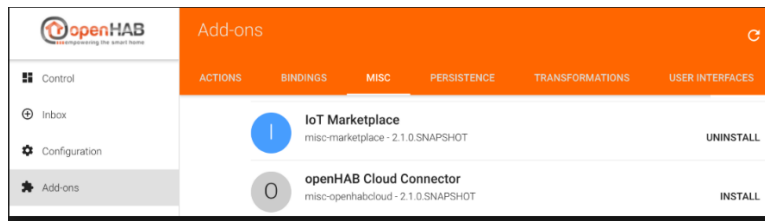


Figure 3. The openHAB configuration

Lastly, the MQTT broker is configured and installed towards the Raspberry Pi (Instructables.com. 2018). There are two methods available on how to install MQTT on Raspberry Pi. First, directly connect the keyboard, mouse and monitor to the Raspberry Pi then activate it. Second, is to access the Raspberry Pi through SSH (using putty (SSH)) then log in to the root to perform installation. The example of MQTT broker interface is shown in the following figure.

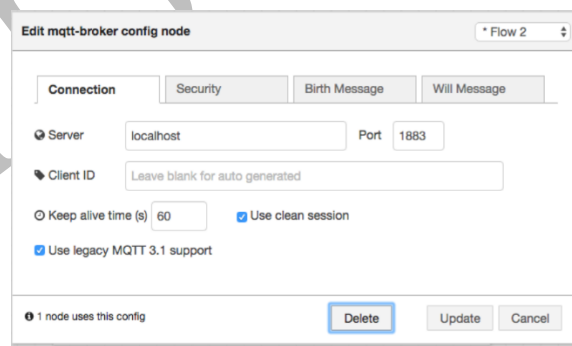


Figure 4. The MQTT broker configuration and installation

After everything is done, the user interface is developed to control the environment or devices after ASPMS detects an alarming state of pollution. A dedicated web site is created for the user interface and it is connected using wifi for smooth hardware and software connection. The overall ASPMS is depicted in this Figure 5.

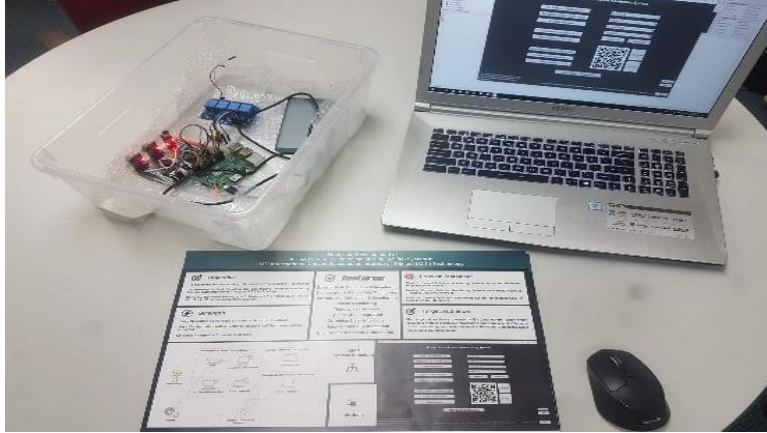


Figure 5. The proposed ASPMS

System Testing

A series of testing were conducted for both hardware and software that constitute the ASPMS. The testing stage is important to achieve the project objectives and maintain the quality and reliability of this project.

The testing performed for each hardware components (MQ135, MQ7, DHT11 and Microphone). The sensors are carefully tested before the actual implementation in order to ensure the smoothness of project development.

The software testing focuses on the functionality of the ASPMS that consist of modules, accuracy, real-time delivery and deployment to cloud platform to maximize the efficiency and effectiveness. The result of both testing confirm the project specification and project goals.

Results and Discussion

This section presents the testing stage and its results of the ASPMS development using IoT technology. There are three major testing that has been conducted that includes connectivity, functionality and system cost.

Connectivity: The results show that connectivity of the selected sensors and Raspberry Pi are well established. Meaning, the sensors used are relevant and overall designed is acceptable to achieve the main objectives in this paper. The IoT is proven flexible to be linked with cloud based services i.e. Thingspeak. This makes real time data is captured and analyzed easily.

Functionality: The testing of hardware and software functionality were successful. No major risk, problem and error during the entire testing stage. This shows the configuration and installation of hardware are well defined and the algorithm that was developed based on this configuration is accurate.

System Cost: The development of ASPMS using IoT has promoted a low cost expenses compared with other technologies. A total of 339.06 Malaysia Ringgit (MYR) are spent on the ASPMS hardware and its pre-stage prototyping materials. There is another 100MYR are spent on

the project such as welding material, carry-case for the project device and miscellaneous expenses (transport in material purchasing). So, the total expenses for the ASPMS development project is around 439.06MYR.

Conclusions

As conclusion, the ASPMS project is successfully carried out in about 6 month duration and able to provide its functionality that meet the user requirements. The ASPMS project is carefully designed, developed and implemented with the sufficient testing procedure on each components and functionality in-order to maximize the system quality. To develop this project and to meet its objective, there are few main parties involved and to be thank with as they have helped and assist in system development. They are the supervisors, the lecturers, the professors, FITS staffs, friends and family who have fully supported this project.

References

1. XMPP-IoT. 2017. Python Read and Write • XMPP-IoT. [ONLINE] Available at: <http://www.xmpp-iot.org/tutorials/python-tutorial/>. [Accessed 26 October 2017].
2. Electronic Design. 2017. Understanding The Protocols Behind The Internet Of Things | Electronic Design. [ONLINE] Available at: <http://www.electronicdesign.com/iot/understanding-protocols-behind-internet-things>. [Accessed 26 October 2017].
3. Roxanne Joncas. 2017. MQTT and CoAP, IoT Protocols. [ONLINE] Available at: https://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php. [Accessed 26 October 2017].
4. Auth0 - Blog. 2017. JavaScript for Microcontrollers and IoT: Part 1. [ONLINE] Available at: <https://auth0.com/blog/javascript-for-microcontrollers-and-iot-part-1/>. [Accessed 26 October 2017].
5. NetBeans IDE - Java ME and Embedded. 2017. NetBeans IDE - Java ME and Embedded. [ONLINE] Available at: <https://netbeans.org/features/java-on-client/java-me.html>. [Accessed 26 October 2017].
6. FrontPage - Raspbian. 2017. FrontPage - Raspbian. [ONLINE] Available at: <https://www.raspbian.org/>. [Accessed 26 October 2017].
7. Oracle/Jose Cruz. 2014. Java ME 8 + Raspberry Pi + Sensors = IoT World (Part 1). [ONLINE] Available at: <http://www.oracle.com/technetwork/articles/java/cruz-gpio-2295970.html>. [Accessed 27 October 2017].
8. Watchdog 1250 - Environmental Monitor with LCD Display & Buzzer | ITWatchDogs . 2017. Watchdog 1250 - Environmental Monitor with LCD Display & Buzzer | ITWatchDogs . [ONLINE] Available at: <http://www.itwatchdogs.com/IP-Climate-Monitor-Watchdog-1250-p11>. [Accessed 26 October 2017].
9. Eurotherm by Schneider Electric | Temperature Control, Process Control, Measurement and Data Recording Solutions. 2017. Environmental monitoring system. [ONLINE] Available at: <https://www.eurotherm.com/environmental-monitoring-system>. [Accessed 26 October 2017].
10. ReliaSENS 18-12 : Environmental Monitoring System Eurotech . 2017. ReliaSENS 18-12 : Environmental Monitoring System Eurotech . [ONLINE] Available at: <https://www.eurotech.com/en/products/ReliaSENS%2018-12>. [Accessed 26 October 2017].
11. PubNub. 2017. Internet of Things 101: Getting Started w/ Raspberry Pi | PubNub. [ONLINE] Available at: <https://www.pubnub.com/blog/2015-05-27-internet-of-things-101-getting-started-w-raspberry-pi/>. [Accessed 26 October 2017].

12. Raspberry Pi. 2017. Raspberry Pi 3 Model B - Raspberry Pi. [ONLINE] Available at: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. [Accessed 26 October 2017].
13. Hackaday. 2017. Introducing the Raspberry Pi 3 | Hackaday. [ONLINE] Available at: <https://hackaday.com/2016/02/28/introducing-the-raspberry-pi-3/>. [Accessed 26 October 2017].
14. IoT Technology | 2017 Overview Guide on Protocols, Software, Hardware and Network Trends . 2017. IoT Technology | 2017 Overview Guide on Protocols, Software, Hardware and Network Trends . [ONLINE] Available at: <http://postscapes2.webhook.org/internet-of-things-technologies>. [Accessed 26 October 2017].
15. Cisco. 2017. Internet of Things (IoT) - Cisco. [ONLINE] Available at: <https://www.cisco.com/c/en/us/solutions/internet-of-things/overview.html#~stickynav=1>. [Accessed 26 October 2017].
16. Olimax. 2017. MQ-135 GAS SENSOR. [ONLINE] Available at: <https://www.olimex.com/Products/Components/Sensors/SNS-MQ135/resources/SNS-MQ135.pdf>. [Accessed 27 October 2017].
17. AOSONG. 2017. Temperature and humidity module. [ONLINE] Available at: <https://akizukidenshi.com/download/ds/aosong/DHT11.pdf>. [Accessed 27 October 2017].
18. Maxim Integrated. 2015. Programmable Resolution 1-Wire Digital Thermometer. [ONLINE] Available at: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>. [Accessed 27 October 2017].
19. Maxim Integrated. 2014. Tiny, Low-Cost, Single/Dual-Input, Fixed-Gain Microphone Amplifiers with Integrated Bias. [ONLINE] Available at: <https://datasheets.maximintegrated.com/en/ds/MAX9812-MAX9813L.pdf>. [Accessed 27 October 2017].
20. UKEssays. 2017. Definition Of Fact Finding Techniques Information Technology Essay. [ONLINE] Available at: <https://www.ukessays.com/essays/information-technology/definition-of-fact-finding-techniques-information-technology-essay.php>. [Accessed 22 November 2017].
21. Design a CO gas detector through a raspberry pi board and MQ-7 CO sensor – kookye.com. 2018. Design a CO gas detector through a raspberry pi board and MQ-7 CO sensor – kookye.com. [ONLINE] Available at: <http://kookye.com/2017/06/01/co%E6%A3%80%E6%B5%8B%E5%99%A8/>. [Accessed 13 April 2018].
22. Measuring Sound Levels | Adafruit Microphone Amplifier Breakout | Adafruit Learning System. 2018. Measuring Sound Levels | Adafruit Microphone Amplifier Breakout | Adafruit Learning System. [ONLINE] Available at: <https://learn.adafruit.com/adafruit-microphone-amplifier-breakout/measuring-sound-levels>. [Accessed 13 April 2018].
23. Design a CO gas detector through a raspberry pi board and MQ-7 CO sensor – kookye.com. 2018. Design a CO gas detector through a raspberry pi board and MQ-7 CO sensor – kookye.com. [ONLINE] Available at: <http://kookye.com/2017/06/01/co%E6%A3%80%E6%B5%8B%E5%99%A8/>. [Accessed 13 April 2018].
24. Instructables.com. 2018. How to Use MQTT With the Raspberry Pi and ESP8266. [ONLINE] Available at: <http://www.instructables.com/id/How-to-Use-MQTT-With-the-Raspberry-Pi-and-ESP8266/>. [Accessed 13 April 2018].
25. Hackster. 2018. Raspberry Pi 2 IoT: Thingspeak & DHT22 Sensor - Hackster.io. [ONLINE] Available at: <https://www.hackster.io/adamgarbo/raspberry-pi-2-iot-thingspeak-dht22-sensor-b208f4>. [Accessed 13 April 2018].
26. openHABian - openHAB 2 - Empowering the Smart Home. 2018. openHABian - openHAB 2 - Empowering the Smart Home. [ONLINE] Available at: <https://docs.openhab.org/installation/openhabian.html>. [Accessed 13 April 2018].