An IoT Based Air Pollution Monitoring System for Smart Cities

Conference Paper · February 2019

DOI: 10.1109/ICSETS.2019.8744949

Arun Kumar
National University of Singapore

An IoT Based Air Pollution Monitoring System for Smart Cities

Harsh Gupta
Department of Electronics and Communication Engineering Jaypee Institute of Information Technology, Noida, India
harsh404me@gmail.com

Dhananjay Bhardwaj
Department of Electronics and Communication Engineering Jaypee Institute of Information Technology, Noida, India
dhananjayb090@gmail.com

Himanshu Agrawal
Department of Electronics and Communication Engineering Jaypee Institute of Information Technology, Noida, India
himanshu.agrawal@jiit.ac.in

Vinay Anand Tikkiwal
Department of Electronics and Communication Engineering Jaypee Institute of Information Technology, Noida, India line 4: City, Country
vinay.anand@jiit.ac.in

Arun Kumar
Department of Computer Science & Engineering
National Institute of Technology
Rourkela, India
kumararun@nitrkl.ac.in

***Abstract*—As the world's population is becoming increasingly urban, the cities are under pressure to remain livable. In recent years, the air quality of the cities has become one of the major cause of concern around the world. Thus, it is necessary to constantly monitor the air quality index of a city to make it smart and livable. In this paper, we propose and develop an IoT based Air Quality Monitoring System for Smart Cities. The real-time data of the air quality is accessed through the smart devices and analyzed to measure the impact on city dwellers. The smart devices are capable of measuring the Temperature, Humidity, Carbon Monoxide, LPG, Smoke and other hazardous particulate matters like PM2.5 and PM10 levels in the atmosphere. The gathered data is accessible globally through an Android Application.**

Keywords—Internet of Things; Air Quality; Smart Cities; Smart App

I. INTRODUCTION

In today's world, air pollution, climate change, and its consequences are of a great concern to the environmentalists and climate change scientists [1]. Emission of various poisonous gases from industries and vehicles are not only hazardous for the terrestrial organism, but the marine life is also getting adversely affected. As the world's population is becoming increasingly urban, the cities are under pressure to remain livable. Health problems arising due to poor air quality are in increase like stroke, heart diseases, lung cancer, respiratory diseases including asthma [2]. Poor air quality poses a significant risk to the vulnerable section of the society such as children, asthmatic, pregnant women, and the elderly persons. As per WHO statics, millions of premature death cases are reported due to air pollution every year worldwide [3]. Due to this, in recent years, the air quality of the cities has become one of the major cause of concern around the world [4]. Thus, it is necessary to constantly monitor the air quality index of a city to make it smart [5] and livable [6-8].

Around the world, governments are building the smart cities to keep a check on these problems and provide a healthy life for its inhabitants. The Indian government is in the process to build 100 smart cities by 2050. These cities will utilize advanced communication network, WSNs, and intelligent system to solve future challenges and create new services. Although the Indian cities have installed real-time air quality monitoring systems in Delhi and other cities, low-cost IoT enabled WSN technology is the future for the coming smart cities around the world [8].

Real-time monitoring of the air quality requires the live data transfer between the devices over the internet and it can be visualized using an Android Application. It reduces the mobilization of system hardware at different locations, which is cost efficient as only one-time installation cost is involved. In the Internet of Things (IoT) based applications [9], Raspberry pi 3B, having a wide range of specifications, is the mainstay of the system. It not only gathers data from various sensors via an inbuilt Wi-Fi module but is also responsible to send the recorded data to the ThingSpeak [10], an open source cloud platform on which data can be stored and retrieved via hypertext transfer protocol (HTTP) over the internet. ThingSpeak acts as a platform to store real-time sensor data and also used to plot graphs, charts, create plugins and apps for collaborating with web services, social network and other application program interface (API). Once signed in, a channel is created with a unique Channel ID. The primary feature of ThingSpeak is the term Channel in which there are eight fields for data storage, three fields to store latitude, longitude and elevation and one field to write a short message to describe the data. Once the channels are created in ThingSpeak, the data can be implemented and alternately one can process and visualize the information through various resources and platform, one of them being on an Android Application, designed in Android Studio. The increased demand for service over the internet has necessitated the data collection and exchange in an efficient manner. IOT has promised the ability to provide the efficient data storage and exchange by connecting the physical devices and vehicles via electronic sensors and internet.

Thus, in order to achieve efficient IOT accomplishment for an application; the proper sensing and monitoring system is essential. The devices can communicate with each other via Machine to Machine (M2M) communication and the physical devices can be controlled digitally [11]. An IOT based monitoring system can be one possible solution. For example, a device can be installed in factories that can constantly monitor the working of the installed machines and if any malfunctioning is detected, a message can be passed on immediately to the controlling unit so that further action can be taken and damage can be avoided.

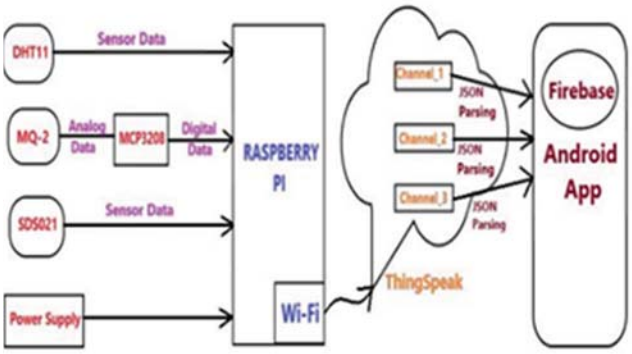

Fig. 1. An Architectural Diagram of the Monitoring System.

Tasks get automated as no more manual labour is required to constantly monitor the machines and thus saving the cost and time. The basic IOT system comprises three broad components namely the sensors, a processing unit or a gateway and a cloud platform. Sensors collect the information and pass it on to the processing unit which processes it as per the requirement and further sends the processed information to the cloud platform using a protocol, where the rest of the analysis can be performed.

II. System Setup

This section comprises of the hardware and software that were used to set up the experimental framework of the research work and to make it executable. Fig. 1 shows a simple architectural diagram of the system developed in the research work.

A. Raspberry Pi

The tiny sensors used in this research work are not capable enough to process the data by themselves and thus require a processing unit to handle the data efficiently. For the large data to be transferred over the internet, a processor with a high processing speed and an inbuilt Wi-Fi module is required. In this research work, Raspberry Pi is used for this purpose.

B. Sensors

Sensors are the peripheral part as they act as a link between the external world and the digital processors by sending the data collected via its vicinity to the processing unit for further processing. In this work, MQ-2 [12]sensor has been used to detect Carbon Monoxide (CO), Liquefied Petroleum Gas (LPG) and Smoke and DHT11 [13] sensor have been used to measure Temperature and Humidity levels of the surrounding [14]. To measure the PPM levels of PM 2.5 and PM 10 particulate matters, SDS021 [15] sensor has been interfaced with the Raspberry Pi.

- DHT11: DHT11 sensor has a moisture holding component sandwiched between two electrodes. A minute variation in humidity leads to change in the conductivity of the component that results in the resistance change between the electrodes which in turn is measured by an Integrated Circuit (IC). The temperature is measured by a thermistor which works as a temperature dependent variable resistor. DHT11 is preferred over DHT22 [13] as the sampling rate of DHT11 is much better than that of DHT22.
- MQ-2: The MQ-2 sensor works on the ionization principle. There is a sensing element inside the sensor which is heated in the presence of the current and then this current is passed through the connecting leads. The air around the heated sensing element ionizes, which further changes the resistance and henceforth the output current from the sensor varies.

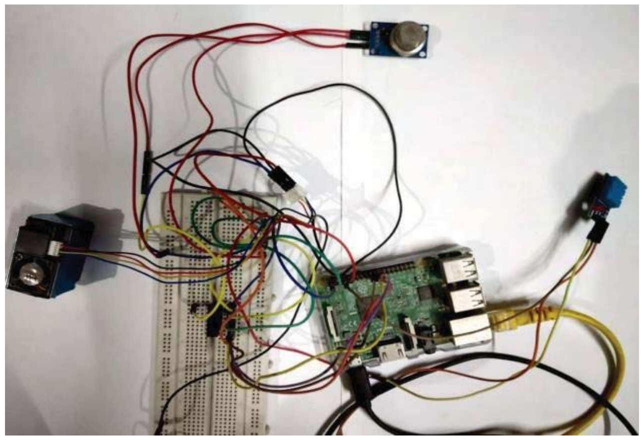

Fig. 2. A Complete Hardware System Setup.

- SDS021: SDS021 works on the principle of laser scattering. Light gets scattered when particles pass through the detecting area and the scattered light is converted into electrical signals which are further amplified and processed.

C. ADC

An Analog to Digital converter (ADC) is used to convert analog information to digital data. MCP3208 is a 16 pin Integrated circuit which operates over a voltage range of 2.7V - 5.5V and it is available in PDIP and SOIC packages. A Raspberry Pi only operates on digital inputs as the board lacks an ADC chip. The output of the MQ-2 sensor used in this research work is analog and therefore, to interface it, an eight-channel and 12-bit resolution ADC that can be interfaced with Raspberry Pi. The outputs of the remaining sensors, SDS021 and DHT11, are digital.

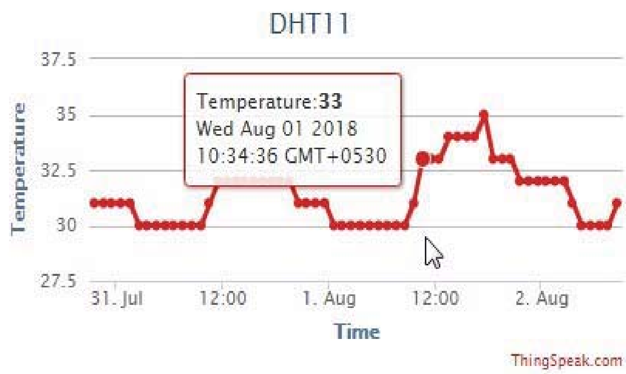

Fig. 3. Temperature variation over the Time (ThingSpeak).

D. Web service

One of the main aims of this research work is to make the real-time data available to authentic users anywhere in the world. This is possible using a cloud platform which can store the real-time data readings. These data points can be used for further processing and visualization. For this purpose, in this research work, ThingSpeak is chosen due to its ease of use and user-friendly interface.

E. Android Application

The app designed in the android studio is created in such a way that it fetches the data in real time from ThingSpeak with the help of an API. The readings of the various sensors along with the date and time are displayed in a tabular format in the app. Applications are handy as they are easy to access through our mobile phones. A user needs to simply sign in with an Email or with Google to further witness the content of the application.

III. Implementation Of The System

This section comprises the elements that were involved in the implementation of this system and the integration of these elements with the system.

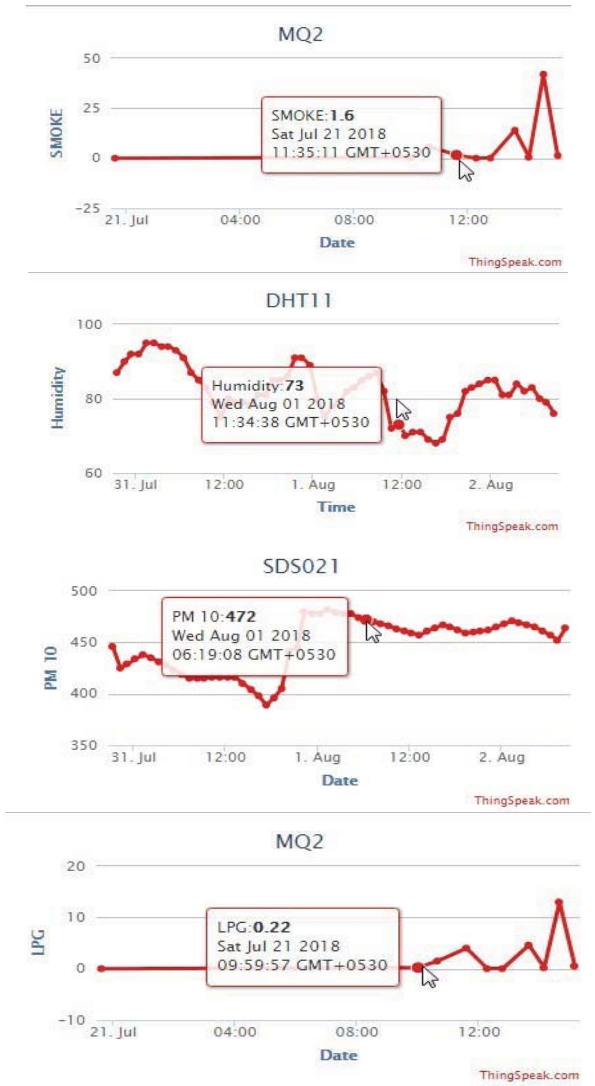

Fig. 4. Smoke, Humidity, PM10 and LPG content in atmosphere over the Time.

A. Sensor interface with Raspberry Pi

Raspberry Pi is a kind of minicomputer and the model, Raspberry Pi 3B, used in this research work has an inbuilt Wi-Fi, Broadcom BCM2837 64-bit quad core 1.2 GHz CPU and 40 general purpose input output pins (GPIO) and many more other features. The board can be powered with a power bank of up to 2.5 Amps. The Raspberry Pi acts as a gateway. In IOT, a gateway is a system that links the data collected by the sensors to the cloud. In order to interface an external MCP3208, the primary aim is to install and enable the Serial Peripheral Interface (SPI) interface. A python library is then created so that MCP3208 and Raspberry Pi can work together.

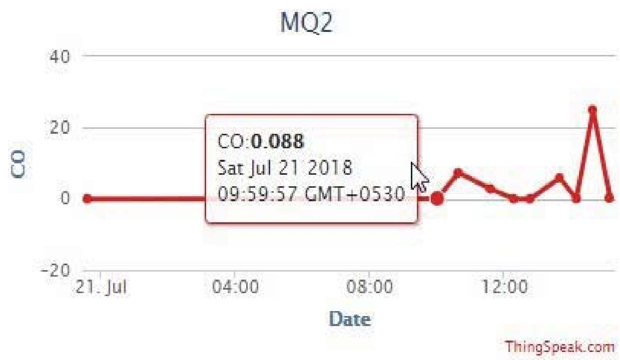

Fig. 5. The CO content in atmosphere over the Time.

The library designed is called in the python code of the MQ-2 sensor as its output is analog in nature. The MQ-2 sensor is first connected with the ADC using a breadboard and connecting wires and then the generated digital outputs are fed as an input to the processor. DHT11 and SDS021 are directly connected with the Raspberry Pi. The complete setup of a simple hardware system is shown in Fig. 2.

B. Sending Data to ThingSpeak

The recorded data by the sensors is further up-loaded to the ThingSpeak. It is stored and retrieved over HTTP which works on the basis of a request and response system. The client makes a request and the server has to respond accordingly. A library named Urllib2 is imported into the python code of sensors, as it fetches the URL. An object is created which specifies the URL we want to fetch and then urlopen function is called which results in returning of the response to the URL requested. Once the data is uploaded, then the platform is responsible for its security. In ThingSpeak, the data can be stored either in a private or a public channel but by default, it is stored in the private channel.

Fig. 6. The PM 2.5 content in atmosphere over the Time.

Data stored on the public channel can be shared with others but in order to connect with a private channel, a valid API key is required. ThingSpeak supports a web and application server and thus the API supports programming languages like Python, Ruby and Node.js.

DATE/TIME	HUMIDITY	TEMPERATURE	LPG	CO	SMOKE	PM2.5	PM10
Jul 10, 2018 09:03 p.m.	58.0	35.0	0.32	0.15	0.79	256	451
Jul 10, 2018 09:04 p.m.	59.0	35.0	0.32	0.15	0.79	256	451
Jul 10, 2018 09:05 p.m.	59.0	35.0	0.32	0.15	0.79	256	453
Jul 10, 2018 10:03 p.m.	59.0	35.0	0.07	0.018	0.16	256	453
Jul 10, 2018 10:04 p.m.	58.0	35.0	0.07	0.018	0.16	256	453

Fig. 7. The recorded data displayed on a Smart Phone.

The data sent by the sensors through an HTTP request is processed by the IOT service in ThingSpeak which communicates with a virtual server. The server and the IOT service communicate directly with the application. Three different ThingSpeak channels have been used for the respective three sensors. Creating three different channels leads to the generation of three different JSON files which is then fetched simultaneously using the Java code in the android studio. Fig. 3, Fig. 5 and Fig. 6 show the variation in Temperature, Carbon Monoxide and PM2.5 respectively with time. Fig. 4 shows the rest of the sensors plot i.e. LPG, PM10, Smoke and Humidity.

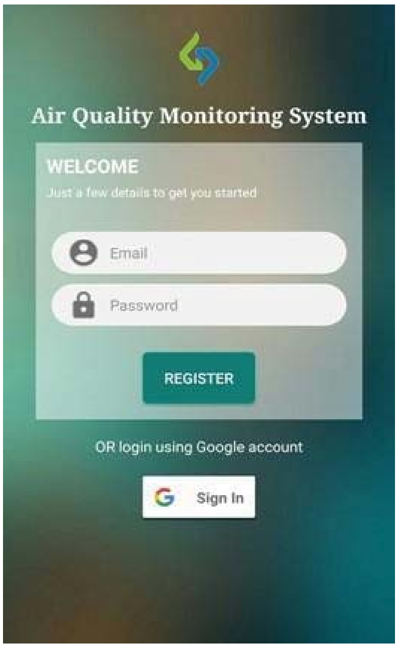

Fig. 8. Login Page Layout of the Android App

C. Android App

For visualization and authentication, in this research work, the team has built an Android App. The Android Application does not restrict itself to ThingSpeak but also has a Firebase login system. Firebase is a Google-provided API to create a database and fetch from it in real time. It also provides enhanced security for the developed App. Also, it is used for backend support and other functionalities like data storage, user authentication and hosting. Real-time data variations are recorded automatically and updates are sent to clients. The HTTP protocol works on a simple request and response system but firebase is different as its real-time database uses data synchronization.

It can be used for user authentication purposes as the user credentials are securely stored using bcrypt. Firebase authentication to the android studio is added using Firebase Android Studio tool. Whenever someone authenticates during the sign-in process either using an Email or Google, it handovers a firebase user object that represents that the authentication is successfully done by the user. This object contains the basic information about the user.

Once the firebase API is included in Android or iOS App, firebase features like Analytics, Authentication, Storage, Messaging, Hosting, Crash reporting, Real-time Database etc. can be used. Fig. 8 shows the login page layout of the developed Android App. The next section includes a basic introduction and then upon dragging from left, a menu section pops out. By clicking on Today in the menu, a list of cities is displayed of which the data is to be monitored. By clicking on any city, a table is displayed, showing the respective data of all the sensors interfaced along with a Date/Time column. Fig. 7 shows the recorded data of the respective parameters. The JSON data fetched by the Android App from ThingSpeak is displayed in a tabular form with the help of JavaScript Object Notation (JSON) parsing.

IV. Conclusion

The paper presented an IOT based Air Pollution Monitoring System for Smart Cities. A few sensors are deployed in the Jaypee Institute Campus to constantly monitor the Temperature, Humidity, Carbon Monoxide, Smoke, LPG, PM2.5 and PM10 levels in the atmosphere. In this work, a one-way communication between ThingSpeak, an open source cloud platform, and an Android Application has been developed. Raspberry Pi has been used as a gateway to interface the hardware system. The Graphs are plotted in ThingSpeak according to the sensors data received and the same can be visualized in an Android App in a tabular format. The Air Pollution Monitoring System is implemented and tested in real-life working condition and we are in the process of extending the nodes outside of the Jaypee Institute. In future, a large-scale node placement and data gathering have been planned for the purpose of forecasting of air pollution.

Acknowledgment

The authors would like to extend sincere thanks to Jaypee Institute of Information Technology, Noida for providing the space and logistics to install the systems in the different locations of the Institute. The author would also like to thank the department of ECE for their support during the system development and study. This research work has been carried out in the Department of ECE, Jaypee Institute of Technology, Noida, India.

References

[1] Frances Moore, "Climate Change and Air Pollution: Exploring the Synergies and Potential for Mitigation in Industrializing Countries", Sustainability, 2009. Vol. 1(1), pp. 43-54.

[2] Brook RD, F.B., Cascio W, Hong Y, Howard G, Lipsett M, Luepker R, Mittleman M, Samet J, Smith SC Jr, Tager I;, "Air pollution and cardiovascular disease: a statement for healthcare professionals from the Expert Panel on Population and Prevention Science of the American Heart Association", Circulation, Vol. 109 (21)June 1, 2004..

[3] H. Ali, J. K. Soe, and S.R. Weller, "A real-time ambient air quality monitoring wireless sensor network for schools in smart cities" In the

Proceedings of the IEEE First International Smart Cities Conference (ISC2'15). 25-28 Oct. 2015 Guadalajara, Mexico.

[4] Sarath K., Guttikundaab, R., "Health impacts of particulate pollution in a megacity—Delhi, India", Environmental Development, April 2013. Vol. 6, pp. 8-20.

[5] C. Kaiwen, et al. "An Intelligent Home Appliance Control-based on WSN for Smart Buildings", In the Proceedings of the IEEE International Conference on Sustainable Energy Technologies (ICSET), Hanoi, Vietnam, 14-16 November 2016, pp. 282-287.

[6] J. Dutta, et al., "AirSense: Opportunistic crowd-sensing based air quality monitoring system for smart city", In the Proceedings of the IEEE SENSORS, 30 Oct.-3 Nov. 2016 Orlando, FL, USA.

[7] Kwok Wai Tham, et al., "A Wireless Sensor-Actuator Network for Enhancing IEQ", In the Proceedings of the The 15th Conference of the International Society of Indoor Air Quality & Climate (ISIAQ) 2018. Philadelphia, PA, USA,.

[8] W. Wang, Y. Yuan, and Z. Ling, "The Research and Implement of Air Quality Monitoring System Based on ZigBee", In the Proceedings of the 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom). 23-25 Sept. 2011 Wuhan, China

[9] Arun Kumar, et al., "Implementation of Smart LED Lighting and Efficient Data Management System for Buildings", Energy Procedia, December 2017. Vol. 143, pp. 173-178.

[10] Marcello A. Gómez Maureira, D.O., Livia Teernstra, "ThingSpeak — an API and Web Service for the Internet of Things", LIACS, Leiden University.

[11] Ming Zhao, et al., "Machine-to-Machine Communication and Research Challenges: A Survey", Wireless Personal Communications, 2017. Vol. 97(3), pp. 3569-3585.

[12] https://www.pololu.com/file/0J309/MQ2.pdf.

[13] DHT11, DHT22 and AM2302 Sensors by lady ada, https://cdn-learn.adafruit.com/downloads/pdf/dht.pdf.

[14] Arun Kumar, H.Y.S., Kai Juan Wong, Peter H. J. Chong, "Location-Based Routing Protocols for Wireless Sensor Networks: A Survey", Wireless Sensor Network, 2017, Vol. 9(1), pp. 25-72.

[15] Laser PM2.5 Sensor specification, Product model: SDS018

http://www.innovazionelocale.it/wp-content/uploads/2018/01/DS018LaserPM2.5ProductSpecV1.5.pdf.