Android Based Real-Time Industrial Emission Monitoring System Using IoT Technology

Dennis A. Martillano, Joshua Miguel R. Dita, Christian G. Cruz, and Kunal S. Sadhra
Malayan Colleges Laguna, Pulo Diezmo Road, Cabuyao City, Laguna 4025, Philippines
Email: damartillano@mcl.edu.ph, dita_joshua@yahoo.com; cruz_christian@gmail.com, sadhrakunal@gmail.com

Abstract —The need to industrialize to compete with global standards is a complete requisite to realize a booming economy. However, there is no question that it has wreaked havoc on the environment caused industrial emissions of dangerous chemicals. This study aimed to create a system that will allow Industrial plants and factories to monitor the emission of the smoke stacks held in a manufacturing company anytime, anywhere using IoT or Internet of Things Technology. "Ref. [1]" explains IoT as a system of physical things embedded with different sensors, software, electronics and connectivity to allow it to perform better by exchanging information with other connected devices. This will help companies in maintaining the machine and provide them emission data of gaseous elements such as carbon monoxide, particulate matter, sulfur and nitrogen dioxide that will help them in complying with the environmental standards of industrial emission. Enabling manufacturing companies to gather plot and interpret data using the system which could be used to further improve emission output and make necessary decisions and corrective actions while imposing cleaner air will benefit the company, the people and the environment.

Index Terms—Internet of Things, Industrial Emission, Air Quality, Mobile Technology, Sensors, Microcontrollers, Arduino

I. Introduction

Industrialization is inevitable in a progressive country. As the Philippines makes its way to finally realize a booming economy, the need to industrialize to compete with global standards is a complete requisite. This requisite entails innovation of new technology, commencing manufacturing corporations and building and housing a den of factories and industrial plants. However, complex changes in the process to elevate and revolutionize things around could also mean complex changes with respect to materials utilization, natural resources and the environment. Industrial plants and factories emit dangerous smoke and chemicals into the atmosphere causing harm to the environment.

With all of these complex changes, there should be a complete balance between industrialization and environmental protection. Several policies and environmental protection procedures that may vary from country to country have been implemented around the world. These are enacted as laws and some are set as a global standard [2]. Clean Air Act is one of these laws which is mandated to be complied as part of Industrial Regulations [3].

Manufacturing companies and factories employ different technologies to help them comply with the acts pertaining to environmental compliance. Pocket-sized environmental sensors can now be conveyed, observing the airborne quality, radiation, water quality, hazardous airborne chemicals and numerous other environment markers [4]. This is where Internet of things (IoT) can create huge opportunities in creating solution to improve industrial processes that will have great impact to the environment [5]. Internet of Things or, more accurately the internet of devices can help in commencing solution that will let devices and sensors connect to the internet and letting them communicate to different entities, applications and each other [6]. In order to identify the requirements, elements and the status of IoT that can help in achieving the primary goal, the study considered collecting and analyzing an overview of IoT, architectures, and vital technologies, its impact, elements, framework, use, and applications.

A systematic review of literatures in [7] identifies real time needs, efficient power consumptions of applications and access to an open and inter operable cloud system as requirements in IoT. This solidifies the need that this study is aimed at.

The objective of the study is to develop a system that will monitor smoke stack emissions by incorporating low powered devices in a microcontroller including nitrogen dioxide, carbon monoxide, particulate matter, and sulfur dioxide sensors. An android based mobile application, incorporated with IoT Technology, connects to a cloud server that collects data and shows accurate real-time emission data that will monitor smoke stack emission anywhere and anytime. The advantages of real-time emissions measurement extend beyond ensuring regulatory compliance and are also central to improving the operational performance of any industrial plants or factories [8]. Having information in real time allows industrial plants to avoid emission breaches and prevent dangerous scenarios for the citizens and for the environment.

II. Methodology

A reformed prototyping intended for the development of projects under Internet of Things was utilized. The

Manuscript received June 30, 2017; revised November 16, 2017.
This work was supported by Malayan Colleges Laguna
Corresponding author email: damartillano@mcl.edu.ph.
doi:10.12720/jcm.12.11.623-629

Modified Prototyping depended on the Nurun Procedure that takes into account the utilization of microcontrollers like Arduino in mixing it with IoT [9]. Fig. 1 demonstrates the 3-way system utilized as a part of building up the task.

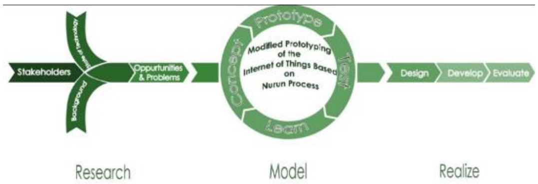

Fig. 1. Nurun-Based modified prototyping for IoT

Research. During this stage, a study was led through actual immersion to a manufacturing company in order to observe and identify the need for the project. Interviews, observations, literature and related studies were gathered and examined with a specific end goal to figure out if the system is practical.

Model. During this stage, the functionalities and the configuration of the system was figured and a working model was then created, peer tried, and analyzed for further improvements. This procedure was repeated a few times until the required system was at long last developed.

Realize. In this stage, the chose respondents were requested to test the system. The tests are explained in the succeeding part of the documentation. The information was then gathered and broken down with a specific end goal to figure out whether the study's targets have been met and if the framework was working legitimately.

III. Results and Discussion

The results of the study can be viewed based on the conceptualized diagram of the actual system set-up and Reliability Tests for the sensors. In this study, the researchers used MQ series gas sensors namely MQ-7(for SO_2 or Sulfur Dioxide), MQ-2(for PM or Particulate Matter), MQ-135(for NO_2 or Nitrogen Dioxide) and MQ9(for CO or Carbon Monoxide). For sensors, a self-calibration (the self-testing of the sensors) and third party tests were performed guided by a pollution control expert prior to integration to the system. Tests were directed at emission testing center situated in manufacturing company.

To increase the accuracy of data, the device sensors must undergo calibration. Baseline data is the initial measurement of information collected prior to the start of the intervention of the program. The value of having a baseline is it really serves as a point of reference. "Ref. [10]" suggests that evaluation is made stronger by having a point of reference; otherwise it will just have the one point in time, and that will not tell if improvements were made in the program. By having a baseline data, progress has been made can be determined.

A. *Self-Calibration of Sensors*

Self-testing was directed utilizing the data for every sensor to recognize the normal yield. For self-calibration, each sensors that were used were exposed to different substances that contains Carbon Monoxide, Nitrogen Dioxide, Sulfur Dioxide, in expected clean air. Fig. 2 indicates the sample readily available baseline data from the sensor datasheet of MQ2(PM or Smoke) sensor [11].

Percent error can be characterized as the room for the margin of error and distinction of the normal and registered yield against the actual readings of the sensors after exposure to various gasses. Values were then thought about and arranged for reference. Table I shows the results of self-calibration.

As shown in Fig. 2, there are 2 axes on the graph and these are Rs/Ro ratio and ppm. The Rs/Ro ratio is the ratio of sensor resistance at various concentration gases and the sensor resistance at 1000 ppm of PM in clean air. The graph shows that if the parts per million (ppm) reaches 200, the Rs/Ro ratio value should be at 3.4. In order to test the reliability of the data sheet, the gas sensor was exposed to a specific gas until the sensor returned an Rs/Ro ratio equivalent in the graph above which is 3.4. The ppm was then computed and compared with the expected ppm based on the graph which is 200ppm.

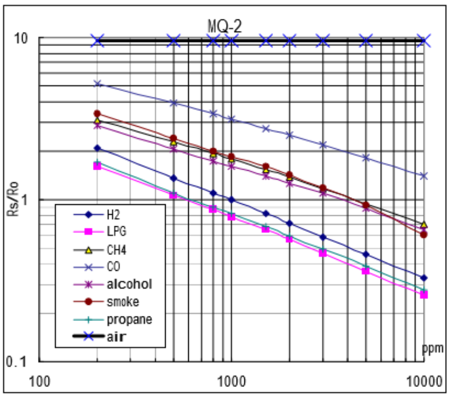

Fig. 2. Baseline data for MQ2(PM or Smoke).

Mathematically, a specific ppm value can be pointed given the graph by identifying the Rs/Ro ratio. The graph in the datasheet shows linear exponentiation of the values of ppm given the Rs/Ro ratio. Therefore, computation of slope can be performed in order to identify value of specific points that can be used in providing self-calibrating algorithm. Given 2 points in the graph, the slope designated by m can be identified through the log scale of the two given points of x and y. Points for different gaseous elements can be seen in the data sheet of MQ2. For the example computation in this paper, the points for smoke were used.

Equation (1) displays the formula for solving of slope.

$$(y2 - y1) / (x2 - x1) \quad (1)$$

The first step in solving the graph is by getting 3 values which are the (1) log value of x1 point in the graph, (2) log value of y1 point in the graph and the (3) the value of slope. To solve for the slope, 2 points were identified using the datasheet of MQ2 (PM or Smoke) sensor. These points are (200, 3.5) for the first x,y point in the graph, and (10000, 0.6) for the second x,y point in the graph. For the MQ2 sensor specifically, the first point is the log value of the first x,y points indentified (2.30103, 0.531479) and the second point is the log value of the last x,y points which is (4, -0.22185). Using the log scale value of x and y points, the result of the slope can be viewed as (-0.22185-0.531479) / (4 - 2.30103) which yielded to -0.44340262041119. The 3 values computed namely: value1 or the x intercept (2.30103), value 2 or the y intercept (0.531479) and the actual slope value (-0.44340262041119) can now be used in slope formula to yield an algorithm that computes for actual ppm value.

To compute for ppm, the value is computed based on the formula provided by one of the manufacturers of the sensors and the slope concept as shown in Equation (2).

(10, (((log(rs_ro_ratio)-value2)/value3) +value 1)) (2)

The result from this computation is now the theoretical value. The percent error can now be computed through this theoretical value. This computation was repeated for all sensors used in the study. To reduce the percent error, the values assigned to the 3 curve points should be adjusted manually until it produces a result with minimal percent error. The results of the self-calibration test should be nearly identical with the expected values. Unstable currents of electricity however can affect the output of the sensor reading, resulting into some minimal but acceptable percent error.

An algorithm within the Arduino microcontroller as shown in Fig. 3 was made to calibrate sensors programmatically in an embedded system based on the results of the computations above.

TABLE I: SELF-CALIBRATION RESULTS OF THE SENSORS USED BY COMPARING THE COMPUTED PPM BASED ON THE DATA SHEET FROM THE READING OF THE SENSORS

Sensor Name	Expected and Computed PPM	Actual Reading (in ppm)	Percent Error
MQ2(PM)	200	198.6	0.7*
MQ135(NO2)	10	9.9	1*
MQ9(CO)	200	199.2	0.4*
MQ7(SO2)	50	49.4	1.2*

```
float MQGetPPM (float rs_ro_ratio, float *pcurve)
{
return (pow (10, (((log (rs_ro_ratio)-pcruve [1]) +pcurve [0]))) ;
}
```

Fig. 3. Algorithm used to convert Rs/Ro ratio into PPM

B. *Third Party Testing*

Third Party testing was performed in order to check the accuracy of the self-calibration testing done by comparing actual reading from hand held device used in an Industrial plant and actual reading from the calibrated sensors. The test was performed under the supervision of a Pollution Control Expert.

Tables II-IV show the result of the testing performed.

TABLE II: COMPARISON OF EMISSION DATA AGAINST THIRD PARTY RESULTS (FIRST TEST)

Element	Actual Reading from Hand held device (in ppm)	Actual Reading from sensor (in ppm)	Percent Error	Remarks
PM	0.95	0.92	3.1579*	Acceptable
NO2	0.62	0.64	3.2258*	Acceptable
CO	13.54	13.39	1.1078*	Acceptable
SO2	0.82	0.84	2.439*	Acceptable

TABLE III: COMPARISON OF EMISSION DATA AGAINST THIRD PARTY RESULTS (SECOND TEST)

Element	Actual Reading from Hand held device (in ppm)	Actual Reading from sensor (in ppm)	Percent Error	Remarks
PM	0.93	0.91	2.1505*	Acceptable
NO2	0.66	0.65	1.5152*	Acceptable
CO	14.7	14.63	0.4762*	Acceptable
SO2	0.89	0.9	1.1236*	Acceptable

TABLE IV: COMPARISON OF EMISSION DATA AGAINST THIRD PARTY RESULTS (THIRD TEST)

Element	Actual Reading from Hand held device (in ppm)	Actual Reading from sensor (in ppm)	Percent Error	Remarks
PM	0.97	0.95	2.0619*	Acceptable
NO2	0.7	0.69	1.4286*	Acceptable
CO	14.11	14.17	0.4252*	Acceptable
SO2	0.91	0.92	1.0989*	Acceptable

*Acceptable percent error is based on the +/- tolerance level of the sensors from the data sheet

According to the data gathered during the tests, the data obtained from the sensors were near the results using the hand-held device given the percentage error. Each table had a 1 (one) hour difference between each test. As the sensors are exposed to the gas, the expected percent error decreases. This means that the longer the sensors are exposed to the gas, the more accurate it becomes [12]. This suggests that sensors are recommended to be exposed to gases for a long period of time to get an accurate result.

C. *System Set-up*

IoT is driven by device to device (D2D) communication. In the system set-up, D2D communication relies on device to server (D2S) communication and the server must share the data it receives with other servers [13]. The set-up includes in its core a mobile technology using a smart phone to elevate the process by which IoT hardware communicates with the Arduino microcontroller over a network that exchange data [14], [15]. "Ref. [16]" and "Ref. [17]" also suggest that smart phones can elevate and power Internet of Things to commence wireless communication among devices and sensors.

Fig. 4 shows the system setup that was implemented in an Industrial plant. First, the gas sensors were connected to analog inputs and set to send emission readings to the Arduino Mega 2560. These analog readings are converted into an actual ppm emission data through the mathematical algorithm within the microcontroller.

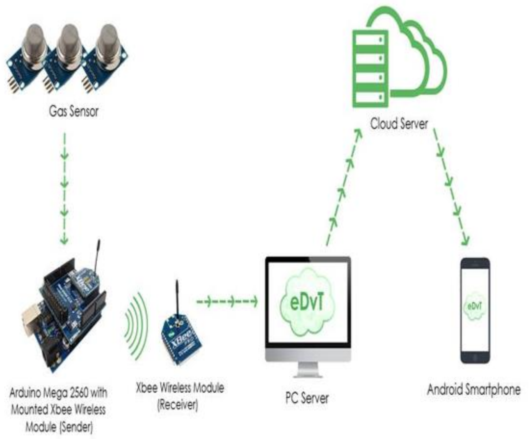

Fig. 4. Block diagram of the system setup

For wireless transmission within the industrial area, two (2) XBEE Modules (an IoT device using IEEE standard 802.15.4) were utilized by the researchers.

The mounted XBEE module configured and connected to the Arduino Mega 2560 served as a sender of the emission data. The emission data is received wirelessly through another XBEE module connected and configured to the PC server through a USB uart cable, which pushes data to database using HTTP protocol. This pushing of data concept can be viewed in the application framework for smart homes [18]. (Fig. 9 shows the circuit diagram of the gas sensors connection to Arduino Mega and Fig. 10 shows the XBEE configuration). The PC server in Fig. 4 then sends the emission data to the cloud server using a web service API. These emission data can be accessed anywhere and anytime through a mobile application that fetches data through the web service API connected to the cloud.

Fig. 5, Fig. 6, Fig. 7A, 7B and Fig. 8, show the actual sensors, the installation/deployment set-up within a manufacturing company, and screenshots of the developed android application respectively.

Fig. 5. Actual Sensors placed in smoke stack of a factory

In actual deployment test, the researchers, with the consent of the Industrial Plant Officers, designed the process by incorporating gate valve that will allow emission to be controlled. (Fig. 8 shows actual deployment of the device). In the block diagram of the deployment test shown in Fig. 6, a long copper tubing was connected to the valve to allow gas emissions to travel along, controlled enough to reach to gas sensors without burning or melting the sensors and other components. This is part of the Industrial standard locally provided in the company. The set-up provides a better way of collecting emission that assist the pollution officers in gathering emission sample without having the need to go to the potentially dangerous smoke stack, which is manually done in their current process to provide environmental compliance.

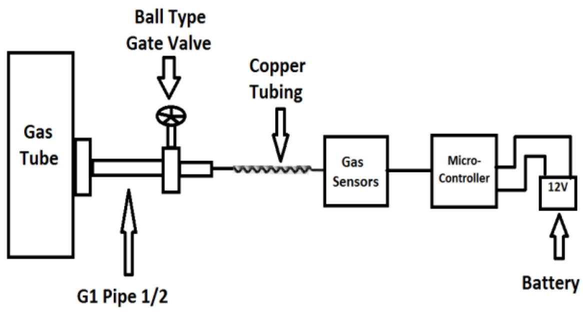

Fig. 6. Block diagram of the deployment Test of the System in the smoke stack of a manufacturing company

Fig. 7A. A screenshot of graph emission data

The mobile application developed can be depicted by the figures shown in Fig. 7A, 7B. The application can provide graphical emission data (in ppm) of the smoke stack. The user can select averaged data of the ppm values of the real-time emission data (CO, NO2, SO2 and Particulate Matter). A user-defined search engine can also be used to extract previous records given input dates.

Fig. 7B. A screenshot emission data history (Yellow for Alarming, Orange for Near Critical, Red for Critical)

A standardized color-coded emission data reading can also be viewed using the mobile application. Emission data are displayed as numerical ppm value. A corresponding color code was set accordingly base on the Air Quality Guidelines provided by the Philippine Clean Air Act that can help identify alarming and critical emission values that can help the factory in maintaining the device and the complying with the emission standards. The Mobile application can identify the actual time and date of the occurrences of alarming and critical readings. These data can be exported into a report in PDF format that can be used in maintenance planning and compliance to mandated emission standard.

Fig. 8. Actual deployment of the device in a smoke stack

Fig. 8 shows how sensors are placed within the smoke stack in a manufacturing company. The study followed deployment process shown in Fig. 6

D. *Circuit Diagrams and Network Configuration using XBEE*

In this study, XBEE S2C Zigbee Modules were used. ZigBee is an open, global wireless standard developed to address the unique needs of low-cost, low-power wireless Machine to Machine networks. This module is also utilizing an IEEE 802.15.4. The 802.15.4 is a standard for wireless communication by the IEEE. The 802.15.4 standard specifies that communication should occur in 5 MHz channels ranging from 2.405 to 2.480 GHz [19]. In an Urban/Indoor, XBEE 802.15.4 and S2C range can be from 100 feet (30 meters) to 200 feet (60 meters) respectively. Outdooor(RF line of sight) range can be up to 4000 ft or 1200 meters. [20]

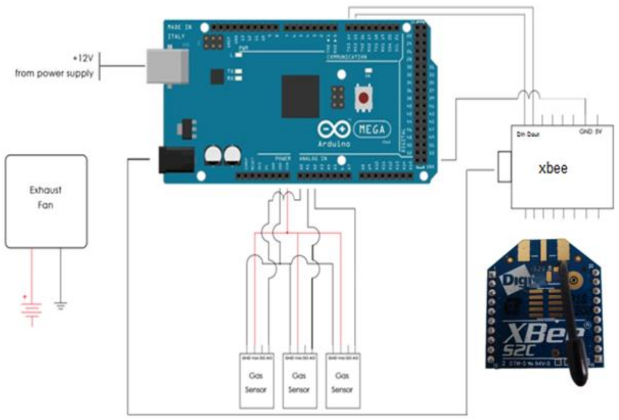

Fig. 9. Circuit diagram of the connection of Sensors and XBEE Module to arduino mega and the actual XBEE S2C

Fig. 9 shows how sensors and XBEE Module are connected to the Arduino Mega. Analog output pins of the sensors are connected to the Analog input pins of the Arduino Mega. Voltage input pins are connected to common 5v source in the microcontroller. Gnd pins of sensors were also commonly grounded to the microcontroller. The Analog signal readings are converted into actual ppm values provided by the algorithm set within the microcontroller

For the XBEE Module, Din and Dout pins are set to Tx (transmit) and Rx (receive) pins of the Arduino Mega respectively. Prior to this connection, the XBEE Module was configured using XCTU Software. The configuration process can be seen in the succeeding part of this paper.

TABLE V: TECHNICAL CONFIGURATION SET-UP IN THE XCTU SOFTWARE FOR TWO XBEE MODULES IN A POINT TO POINT COMMUNICATION

Setting	Acronym	XBEE1 Config	XBEE2 Config
CHANNEL	CH	C	C
PAN ID	ID	1234	1234
DESTINATION ADDRESS High	DH	0	0
DESTINATION ADDRESS Low	DL	1	0
Source Address	MY	0	1
Coordinator Enable	CE	1	0

XBEE S2C Modules were used to allow wireless point to point communication and sending of emission data between the sensors in Arduino and the computer server. Table V shows the configuration performed in the two XBEE Modules in the study. Critical settings were identified with correct configuration. This setting was performed using the XCTU software intended for XBEE Zigbee devices. A Screenshot of the software used to configure the XBEE Modules can also be seen in Fig. 10.

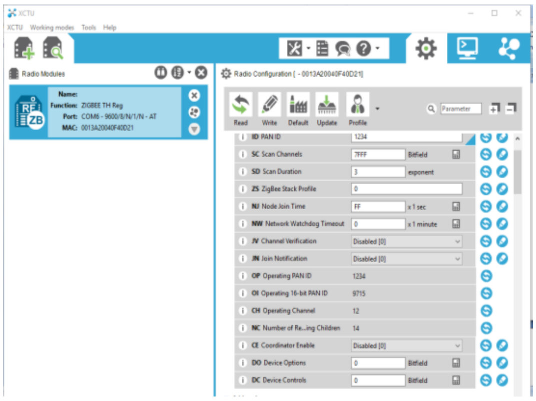

Fig. 10. XCTU screenshot for XBEE configuration

In order to provide a network environment using XBEEs, several configurations should be set in hardware using a free configuration software called XCTU. First set-up is the channel. This controls the frequency band that your XBEE communicates over. Most XBEEs are operating on the 2.4GHz 802.15.4 band, and the channel further calibrates the operating frequency within that band. We can usually leave the channel setting, or at least make sure every XBEE Module are on the same network and operating on the same channel.

The personal area network ID (PAN ID) should also be configured. The network ID is some hexadecimal value between 0 and 0xFFFF. XBEEs can only communicate with each other if they have the same network ID. Therefore, all XBEEs should have the same PAN ID. In this study, both XBEE set in Arduino Mega and XBEE in the computer server use the same PAN ID.

Finally each XBEE in a network should be assigned a 16-bit address (again between 0 and 0xFFFF), which is referred to as MY address, or the "source" address. Another setting, the destination address, determines which source address an XBee can send data to. For one XBee to be able to send data to another, it must have the same destination address as the other XBee's source. An option for this is to leave DH set to 0, and set DL to the MY address of the receiving XBee. Generally, we could set DH to the Serial Number High (SH) and DL to the Serial Number Low (SL) of your destination XBEE. Also, the Coordinator setting should also be enabled to the sending module and disabled to another XBEE in the network to commence communication.

IV. Conclusions

In this study, IoT Technology and the required sensory elements that will provide real-time Industrial emission monitoring were integrated. A device was developed by incorporating calibrated nitrogen dioxide, carbon monoxide, particulate matter, and sulfur dioxide sensors to Arduino Mega that produce accurate readings base on different calibration techniques used. Self Calibration was made and baseline data was used to provide accurate output. Algorithm base on a mathematical formula using the data sheet of MQ sensors was developed to convert analog readings into actual ppm values. Third party testing was performed under the supervision of a Pollution Control Officer to compare results and determine reliability of the device. Configured XBEE modules that utilize IEEE 802.15.4 standard were set-up to commence communication in a point to point wireless network among the entities in the system including the sensors, the computer server and the android application. An android application was developed that connects to a cloud server through a web API, allows the Industrial plant officers to view, collect, and monitor industrial emission data in real-time, thereby reducing the need of manual emission data gathering and checking.

The system could be used in any industrial factory that emits dangerous smoke, and specific chemicals and gaseous elements. This could enable an Industrial Plant to provide self-monitoring of emissions that can help in identifying issues in machines and processes while complying with the local and global environmental standards with respect to industrial emissions.

V. Recommendations

Testing and other ways of gathering information and feedbacks from the user can make a huge impact in improving a system. One of the features that should be included is prediction of emission results for the users to know the pattern of the emission data that would allow the engineers to predict the next levels of emission. With this, the company would be prepared for possible clean-ups in-between operations.

To further improve the study, it is also recommended to use additional sensors with Arduino to monitor different kinds of gaseous elements other than sensory devices used in this study. This could provide a better grasp of the level of gaseous elements produced in a factory. Also, in order to produce a more accurate output, the study recommends exposing the sensors longer than 1 hour to a gaseous element prior to implementation. The study would also like to direct the future researchers to consider mathematical proportionality computation of the effect of external factors including distance of emission source, temperature index, and even humidity index that could have direct impact to accuracy of the output.

Acknowledgment

This study was conducted at Pigmentex Incorporation, a manufacturing Company in Carmona, Cavite, Philippines through the supervision of Mr. Romeo Imperial. Presentation of this paper is supported by Malayan Colleges Laguna.

REFERENCES

[1] K. Ashton, “That 'Internet of Things' Thing,” *The RFID Journal*, June 2009.

[2] Process Online Staff. (July 2010). Continuous emission monitoring. [Online]. Available: http://www.processonline.com.au/content/instrumentation/article/continuous-emission-monitoring-300701259

[3] Republic Act No. 8749, Philippines Clean Air Act, 1999

[4] S. Brienza, A. Galli, G. Anastasi, and P. Bruschi, “A low-cost sensing system for cooperative air quality monitoring in urban areas,” in *Proc. International Conference on Smart Computing Workshops*, 2015.

[5] (February 2016). The Industrial Revolution and Its Impact on Our Environment. [Online]. Available: http://ecoissues.com/TheIndustrialRevolutionandItsImpactonOurEnvironment.html

[6] S. G. Hong, T. Heo, N. Kim, and K. B. Lee, “ThumbUp: Dynamic updates of loT device applications for smart personalized services,” in *Proc. International Conference on Information and Communication Technology Convergence*, 2016, pp. 570-575.

[7] S. Madakam, R. Ramaswamy, and S. Tripathi, “Internet of Things (IoT): A literature review,” *Journal of Computer and Communications*, May 2015.

[8] C. Burlew and R. Oommen, Eastern Research Group (ERG). Memorandum to Jim Eddinger, U.S. Environmental Protection Agency, OAQPS. Development of Average Emission Factors and Baseline Emission Estimates for the Industrial, Commercial, and Institutional Boilers and Process Heaters National Emission Standard for Hazardous Air Pollutants, October 2002

[9] Turin. (June 2014). Prototyping the Internet of Things With Nurun and Arduino. Nurun.

[10] *Continuous Emissions Monitors (CEMs).* Emerson Process Management, 2012.

[11] MQ2-Datasheet. HANWEI ELETRONICS CO., LTD, 2017.

[12] A. V. Cao, Air Pollution Tracking Device for Smokestacks of Factories and Industrial Plants, May 2016.

[13] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, “Smart objects as building blocks for the internet of things,” *IEEE Internet Computing*, vol. 14, pp. 44-51, 2010.

[14] Yegulalp, S. (February 2014). Android will power the Internet of things. [Online]. Available: http://www.infoworld.com/article/2610361/big-data/android-will-power-the-internet-of-things.html

[15] C. Doukas, Building Internet of Things with the Arduino vol. 1, 2012.

[16] U. Sharma and S. R. N. Reddy, “Design of Home/Office automation using wireless sensor network,” *International Journal of Computer Applications*, vol. 43, pp. 53-60, 2012.

[17] D. Blouin. (September 2014). Mobile Devices are the Gateway to the Internet of Things. [Online]. Available: http://www.cmswire.com/cms/internet-of-things/mobile-devices-are-the-gateway-to-the-internet-of-things-026531.php

[18] A. Kamilaris, V. Trifa, and A. Pitsillides, “HomeWeb: An application framework for Web-based smart homes,” in *Proc. 18th International Conference on Telecommunications*, 2011, pp. 134-139.

[19] J. Lee and Y. Wang, “Experimental evaluation of ZigBee-based wireless networks in indoor environments,” *Journal of Engineering*, vol. 2013, 2013.

[20] Digi International Inc. XBEE Protocol comparison. [Online]. Available: https://www.digi.com/pdf/xbee-802-15-4-protocol-comparison.pdf

Dennis A. Martillano is a graduate of Bachelor of Science and Computer Engineering from STI College Southwoods. He was born in Carmona, Cavite Philippines on May 6, 1985. He has units in Mathematics from the University of the Philippines and a Master’s degree holder in IT and a Graduate Degree holder in Distance Education also from the University of the Philippines. He is currently taking up his Doctor in Information Technology in Technological Institute of the Philippines and is currently teaching in the College of Computer and Information Science at Malayan Colleges Laguna in Cabuyao Laguna, Philippines.

Joshua Miguel R. Dita is a graduate of Bachelor of Science in Information Technology at Malayan Colleges Laguna. He was born on July 9, 1997 in Biñan, Laguna. He took up Web and Mobile Technology as his Specialization.

Christian Kevin G. Cruz is a fourth-year college student taking up Bachelor of Science in Information Technology at Malayan Colleges Laguna. He was born on December 25, 1992 in Sampaloc, Manila. He took up CISCO Network Administration as his specialization.

Kunal S. Sadhra is a fourth-year student taking up Bachelor of Science in Information Technology. Mobile and Web Development is the specialization he chose. He was born in BF Homes, Parañaque on November 20, 1996.He likes to work in teams and groups and likes to create a positive hard-working environment to finish specific tasks.