On the Use of Low Cost Sensors for the Implementation of a Real-Time Air Pollution Monitoring System Using Wireless Sensor Networks

Diego Ortiz*, Diego S. Benítez*, Walter Fuertes†, and Jenny Torres‡

*Colegio de Ciencias e Ingenierías "El Politécnico",
Universidad San Francisco de Quito USFQ, Campus Cumbayá, Casilla Postal 17-1200-841, Quito, Ecuador
†Departamento de Ciencias de la Computación,
Universidad de las Fuerzas Armadas ESPE, 171-5-231B, Sangolquí, Ecuador.
‡ Departamento de Informática y Ciencias de la Computación,
Escuela Politécnica Nacional, P.O. Box 17-01-2759, Quito-Ecuador
diego.ortiz@estud.usfq.edu.ec, dbenitez@usfq.edu.ec, wmfuertes@espe.edu.ec, jenny.torres@epn.edu.ec

***Abstract*—In this paper, we investigate the use of low cost wireless sensor networks for the development of an environmental monitoring system that can be used to visualize air pollutants in real time by means of a web application. Measuring nodes were implemented, using low cost analog and digital sensors, for collecting data about air pollutants such as CO_2 and dust particles concentrations present in the environment. The system works in such a way that the values measured by the sensor nodes are encoded and sent via Wireless to the Gateway (Raspberry pi) by means of Xbee pro S1 modules. The data are decoded and stored in a local database. Afterwards, a web application makes a PHP connection with the database, and data are plotted in real-time, using the HighChart JavaScript library with AJAX methods and JSON notation. To check the functioning of the system and to compare the behavior of the sensors, measurements of the concentrations of CO_2 and dust particles were taken outdoors and inside an underground parking lot.**

I. Introduction

In present time, environmental pollution is one of the major global problems affecting human health. Approximately, 7 million people die each year due to diseases related to air pollution [1]. With the growth of population and their associated pollutants emission sources and activities, controlling the air quality is therefore a growing challenge [2]. In such sense, the information that could be provided in real-time by an air quality monitoring system may be very useful for: determining places where there is high exposure to pollutants, analyzing the danger to which people could be exposed due to constant presence of environmental contaminants, avoiding serious consequences to the ecosystems and living beings of the planet, and promoting environmental awareness.

Wireless sensor networks (WSN) have been very useful for creating low cost environmental monitoring systems [3]. Some advantages of using these sensor networks are: a greater number of measurement nodes can be implemented to increase the monitoring area, the installation is simple, they require low maintenance, and they are appropriate for collecting information. However, one of the biggest disadvantages is that they depend on a limited power source, since nodes can be located in remote places with difficult access to traditional power sources.

The human activity that produces the biggest amount of polluting emissions such as: oxides, carbon monoxide (CO) and carbon dioxide (CO_2) is transport. This activity directly affects the air quality of a city. The emissions of polluting gases produced by the daily transport of people and goods have a great impact on the lower layers of the atmosphere, which directly affects people's health [7]. In recent years, the number of vehicles has increased significantly in urban and rural areas, so it is necessary to measure the amount of polluting gases they produce.

In such sense, several environmental monitoring systems have been proposed in the literature, for example in [4], CO_2 and NO_2 sensors with analog outputs were used to measure the concentrations of gases emitted near a motorcycle and also in a closed space full of people. Real-time monitoring systems have also been proposed coupled on a mobile unit [5], which can measure the gas levels in the environment, using a wireless sensors associated with a smart-phone to act as an interface.

There are several environmental and air quality monitoring centers, where very sophisticated and high cost atmospheric gas measurement equipment is used. However, in this paper an alternative is proposed for having a greater number of measurement nodes based on Wireless Sensor Networks (WSN), with a much lower budget and easy implementation, but integrated into the concept of Internet of Things (IoT), increasing therefore the capacity of the network nodes, as described in [6]. Nonetheless, when using low-cost sensors, the concentrations of the gases to be measured must be within the sensitivity range of the sensors. Although, environmental measurements will be made in places where these sensitivity ranges are met to guarantee the reliability of the data and the correct functioning of the sensors, the aim of this study is to understand the behavior of low cost sensors and investigate

978-1-5386-5935-9/18/$31.00 ©2018 IEEE

their limitations for this application.

The environmental gas monitoring system implemented here was designed for the collection, analysis and storage of measurements from harmful gases to human health, such as fine particulate matter lower than 10 and 2.5 μm in diameter (PM10 and PM2.5, respectively), and carbon dioxide (CO_2). Data measured by the sensors will be sent via Wireless using Xbee communication modules from the data acquisition boards (based on Arduino microcontroller boards) to the output port (based on a Raspberry Pi board), data will be decoded and stored in a local database. The system has a web server that makes a PHP connection to make queries of records to the database. Data were transformed into JSON format and plotted in real time using AJAX methods and the Highchart JavaScript library.

II. Methodology

Fig. 1 shows the topology of the proposed environmental pollution monitoring system implemented in this work as an IoT application to visualize the levels of air pollution in Web services. The system is composed of measurement nodes that contain environmental gas sensors (analog or digital) connected to a microcontroller board (either an Arduino nano or an Arduino Mega), a Raspberry Pi (Gateway), and a PC that serves as a web server for the online application. An advantage of using this topology for environmental measurements is that the sensor nodes can be in remote locations where there is no connection to the Internet and with limited power supplies, the data collected by the electronic boards can be transmitted via Wireless by the Xbee modules up to 500 m range towards the Gateway in open places. Another advantage of this topology is that if the connection to the Internet or the connection to the local network is interrupted, the received data are not lost since data measurements are written locally in a database independently of the network connection.

Data measured by the sensor nodes are encoded in a frame and sent to the Gateway (Raspberry Pi) via Wireless, where they are decoded and stored in the database. Subsequently, a PC containing the Apache application server, connected to the same LAN network of the Raspberry Pi, by means of a PHP connection reads the records available in the database. These data are plotted in real time, with the use of the HighChart JavaScript library and with AJAX methods and JSON notation. Finally, with the help of the toolkit of NO-IP.com DUC 4.1.1 [11] the desired configuration is achieved for the publication of a web page on the Internet showing data in real time.

Since the aim of this work was to used low cost sensor for the sensor nodes, we first experimented with the low-cost gas and pollution detector modules (MQ sensors) easily available for Arduino boards, and then we also experimented with digital gas sensors a bit more expensive but within the low cost range. Details of such implementations are described next.

A. Sensor Nodes

1) Connection and configuration of analog sensors: For the acquisition of data measured by analog sensors, an Arduino Nano board was used, which contains the microprocessor ATmega328P [13]. This model has 6 analog inputs of 5V, and has a digital to analog converter with 10 bits of resolution. The analog outputs of sensor modules were connected to the analog inputs of the Arduino for reading. Additionally, sensors were polarized with 5 volts supplied by the Arduino Nano.

The sensors used for CO_2 measurements was the SAINSMART MQ135 and for the PM2.5 dust particles measurements the Sharp GP2Y10. Additionally, since these low cost sensor also have limitations related their operating temperature range, a temperature sensor (LM35) was also included in the node. Table I shows the measurement ranges of these sensors.

TABLE I
ANALOG SENSORS USED BY THE MONITORING SYSTEM

Sensor type	Model	Range	Units
Dust particles PM2.5	Sharp GP2Y10	0 a 0.8	mg/m^3
Carbon dioxide CO_2	MQ135	10 a 10000	ppm
Temperature	LM35	-55 a 150	°C

The CO_2 sensor has to be preheated for a period of 12 hours for its correct operation. The Sharp GP2Y10 optical dust sensor has an infrared diode and an aligned phototransistor to take measurements. To achieve good measurements with this sensor, it is necessary to control the integrated infrared diode in such a way that it is ON for 320 μs, and to measure the analog value in the time interval from 280 μs to 320 μs, then the diode should remain OFF for 9680 μs.

Because these sensors are of low cost, there are limitations when obtaining very precise data, however, in places of high concentrations these sensors manage to have an acceptable performance, giving a good reference of the measured concentrations.

The voltage output values of the CO_2, temperature, and dust particles PM2.5 sensors, are internally transformed within the measurement node to their equivalent values of parts per million (ppm), degrees Celsius (°C) and ug/m^3 respectively. Subsequently, these values must be transmitted to the Gateway, for this it is necessary to encode the data in a serial frame. The coding consists of control characters between the values measured by the sensors. This frame is then sent by the serial port for wireless transmission through the Xbee module. Fig. 2 shows a photography of the physical implementation of the analog measurement node prototype.

2) Connection and configuration of digital sensors: In this case, the measurement node is composed by an Arduino Mega board, a CO_2 sensor, dust particle concentration sensors (PM2.5 and PM10), Hydrogen Sulfide (H_2S) sensor and a GPS module. The Arduino Mega board has the ATmega 2560 microprocessor that has 4 serial communication ports and supports I2C communication. The Telaire T6713 CO_2 sensor communicates with the Arduino via the I2C protocol. The Honeywell HPM sensor for PM2.5 and PM10 and the H_2S sensor (DGS 968-036) communicate with the Arduino via the serial ports. The data frames sent by the sensors were decoded to obtain the value measured in concentration units. The sen-

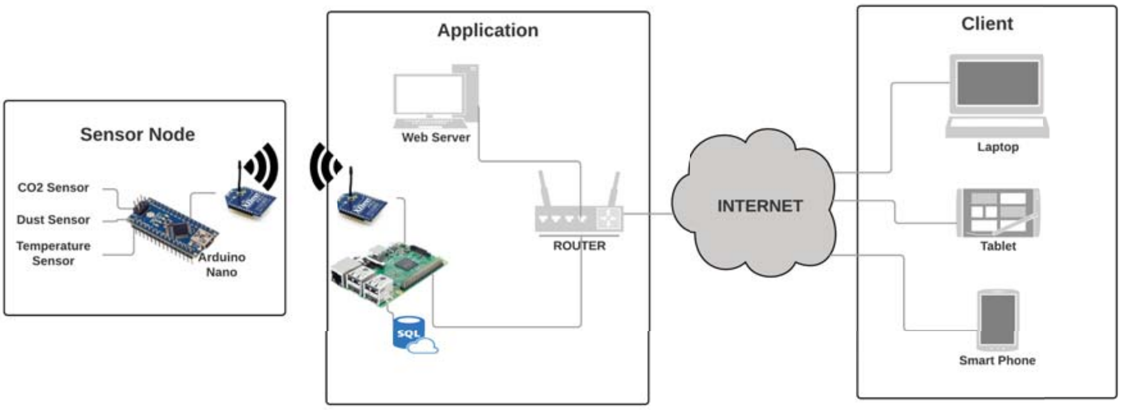

Fig. 1. Topology of the environmental monitoring system proposed.

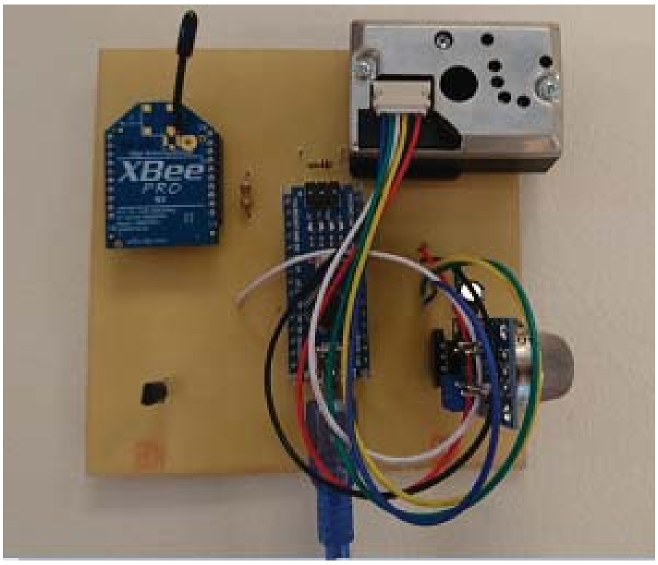

Fig. 2. Photograph of the implemented measuring node prototype, electronic board with the analog sensors.

sors used do not need calibration. The measurement ranges for these sensors are presented in the Table II. Additionally, a GPS module (GY-GPS6MV2) was also included in the note. The GPS communicates via serial with the Arduino and provides the latitude and longitude of the place where measurement is obtained. Fig.3 shows a photo of the measurement node implemented.

TABLE II
DIGITAL SENSORS USED BY THE MONITORING SYSTEM

Sensor type	Model	Range	Units
Dust particles (PM2.5 and PM10)	Honeywell HPM	0 to 1,000	μg/m^3
Carbon dioxide CO_2	Telaire T6713	0 to 5000	ppm
Hydrogen Sulfide (H_2S)	DGS 968-036	0 to 10	ppm

B. Getaway implementation

Data from the measurement node are transmitted to the gateway, which was implemented using a Raspberry Pi board,

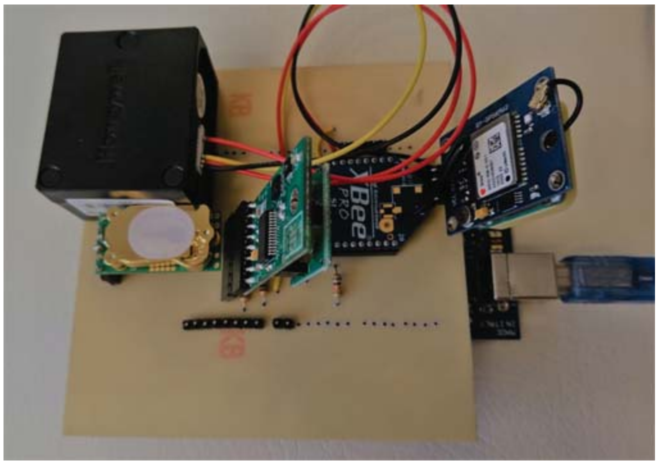

Fig. 3. Photograph of the measurement node prototype implemented with digital sensors.

via Wireless using the Xbee module, to be interpreted and stored in the database. The decoding of the data frames was implemented in the Raspberry Pi using a Python 2 programmed script. For this, it is necessary to configure the serial port to be used by the external digital ports of the Raspberry, since that is where the Xbee module connects. The implementation is shown in Fig. 4.

The decoding is used to separate the values measured by each sensor from the received data frame from the sensor node, first the control characters are located and then information from the sensors is separated and extracted from the frame and written in the database. T

C. Wireless communication configuration

For wireless communication between the data acquisition nodes and the Raspberry pi (Gateway), Xbee Pro S1 802.15.4 [12] modules were used, the modules were configured to have a point-to-point communication. For this, it is necessary to have one Xbee module configured as a coordinator and another as an exit point (End Point). The communication parameters between the two devices were: Personal Area Network ID (PAN ID), Destination High (DH) and Low Address (DL).

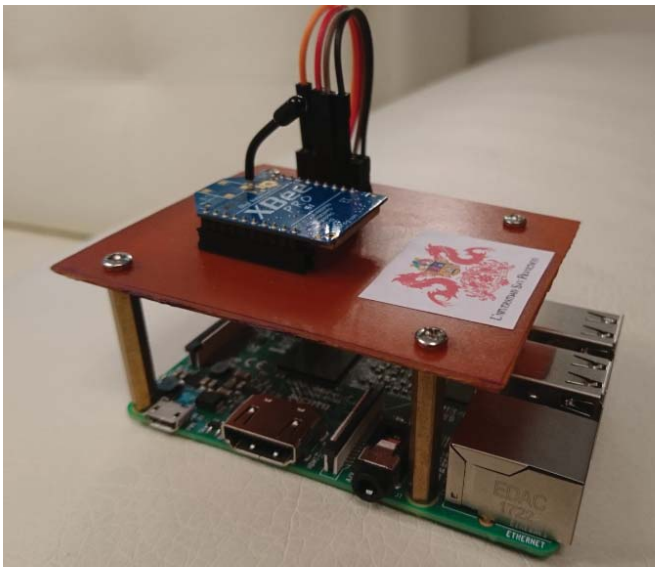

Fig. 4. Photograph of the gateway implemented with the Raspberry Pi board and the Xbee communication module.

The PAN ID should be the same in both modules so that they have communication within the same wireless network. Since a point-to-point link was used, it was necessary to define the destination address in each module. In this case, in each device the address (DL) of the module with which it is to be communicated is entered and vice versa. The high destination address (DH) is the same for devices of the same family, in this case the Xbee Pro S1 model. The parameters used for serial communication in the Xbee, Arduino boards and Raspberry Pi modules were: 9600 Baud Rate, without parity, one stop bit and eight bits data.

D. Database

Within the Gateway (Raspberry Pi), a MySQL database (MariaDB 10.1 [14]) was installed. Data measurements from the sensors are written to a table within the database. The created table has fields for: day, hour, measurement node identification, CO2 concentration value, dust concentration value (PM2.5 and/or PM10), Temperature value, and in the case of the digital sensor node (H_2S) concentration. In the implemented script, a local connection was made to the database, the previously decoded data measurement are taken and inserted into the database.

E. Web server implementation

A web application server was used to perform the real-time visualization through a web page were real-time and historical sensor data can be displayed, as illustrated in Fig. 5. The server used in this project was Apache 2.4.29 [15]. The connection was made to the database located in the Raspberry Pi using PHP. For which it is necessary that the Raspberry Pi and the web server are in the same local network with the necessary permissions to be able to access the database.

The parameters necessary to establish a connection to the database remotely are: Host IP, in this case the Raspberry Pi, the user name, password and the name of the database to be accessed. Subsequently, a query instruction is performed, which returns the requested data. For filling real-time charts, the database is consulted for the value and date of the last stored data. In the case of the historical data chart from previous days, all the values stored on the date selected by the user are consulted. By using PHP instructions, it is possible to capture the data found in the database. Data are transformed through the PHP command JSON encode (JavaScript Object Notation), which converts a data object into a string of characters that JavaScript can interpret.

Fig. 5. Screenshot of the real-time web page implemented.

III. Experimentation and Results

To verify the correct operation of the implemented prototypes, several measurements were made at the facilities of the Universidad San Francisco de Quito (USFQ), in Cumbayá, Ecuador. Measurements were taken outdoors in the campus areas, and inside a parking lot located at the underground level. Measurements were taken every 2 seconds for 5 consecutive hours.

A. Results using analog sensors node

The results obtained using the analog sensors node are shown in Fig. 6 where the CO_2 concentrations are shown in both places. As expected, the concentration inside the underground parking lot exceeds the concentration in open air. The average concentration within the underground parking lot is 1097 ppm with a standard deviation of 207.70, while the average outdoor concentration is 316.77 ppm with a standard deviation of 50.15. According to Kane International [8] the actual concentration of CO_2 in the open air is between 250 and 350 ppm while at the atmospheric level it is around 400 ppm [9]. The values obtained for exteriors CO_2 concentrations are within this reference range, while the concentration in the underground parking is within the range of bad air quality [8].

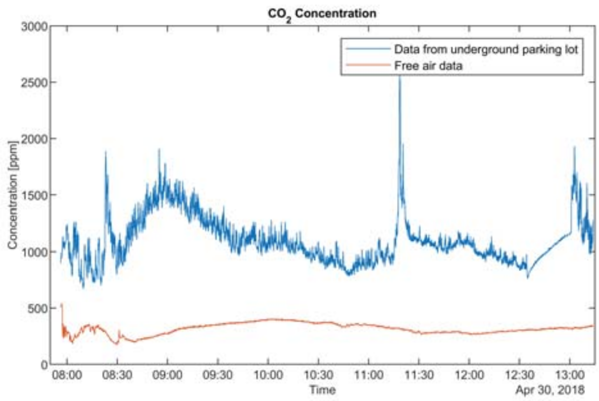

Fig. 6. CO_2 measurements obtained using analog sensor node inside an underground parking lot and outdoors.

Similarly, the PM2.5 sensor data obtained using analog sensors node are shown in Fig. 7. As it can be seen again, the concentrations of dust particles are higher in the interiors of the parking lot. The average concentration value within the underground parking lot is 13.59 μg/m^3 with a standard deviation of 3.80, while the concentration in free air is 6.85 μg/m^3 with a standard deviation of 3.04. According to the Environmental Agency (Secretaria del Ambiente) in Quito [9], in the month of April 2018 the monthly average in Tumbaco was 12.09 μg/m^3. Finally the temperature values were obtained in both places. The average value for the parking lot was 26.42 °C while for the exteriors of 22.40 °C.

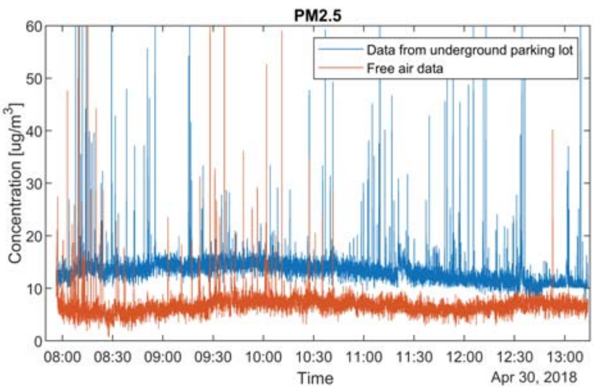

Fig. 7. Dust particles (PM2.5) measurements obtained using analog sensor node inside an underground parking lot and outdoors.

B. Results using digital sensors node

Similar experimentation was carried about with the digital sensors node, the coordinates obtained by the GPS module were -0.19949, -78.4361 for the underground parking and -0.19636, -78.4348 for the exterior areas.

The CO_2 concentrations in both places are shown in Fig. 8. The average concentration obtained with the digital sensors inside the underground parking lot was 445.26 ppm with a standard deviation of 198.79, while for the outdoors areas the average concentration was 260.77 ppm with a standard deviation of 64.33.

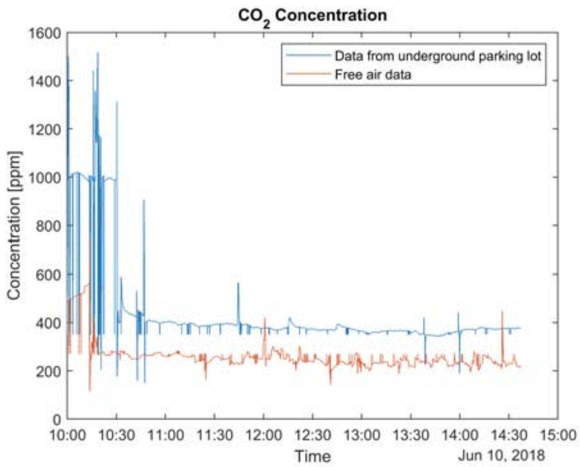

Fig. 8. CO_2 concentration measurements obtained using the digital sensor node inside an underground parking lot and outdoors.

The results obtained for PM2.5 dust particles are shown in Fig. 9. In this case, the average concentration in the parking lot was 15.46 μg/m^3 while for the outdoors it was 4.34 μg/m^3.

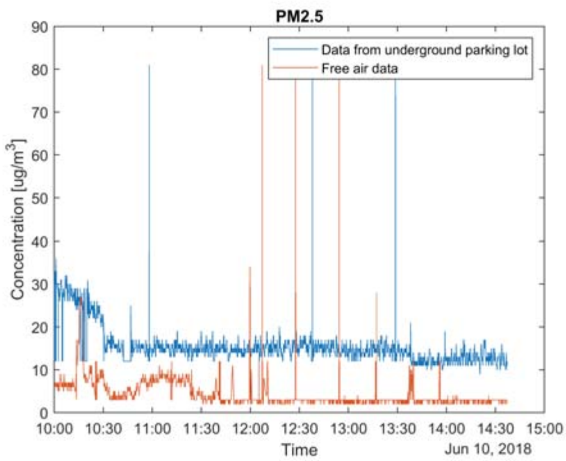

Fig. 9. Dust particles (PM2.5) measurements measurements obtained using the digital sensor node inside an underground parking lot and outdoors.

The results obtained for PM10 dust particles are shown in Fig. 10. In this case, the average concentration inside the parking lot was 15.93 μg/m^3 while for the outdoors it was 4.59 μg/m^3.

In Fig. 11 the measurements of H_2S concentration are shown. In this case, the average concentration in the parking lot was 83.52 parts per billion (ppb) while in the exterior it was 50.44 ppb.

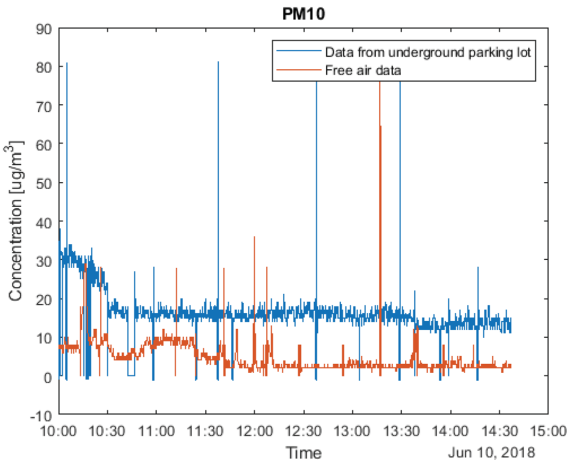

Fig. 10. Dust particles (PM10) measurements obtained using the digital sensor node inside an underground parking lot and outdoors.

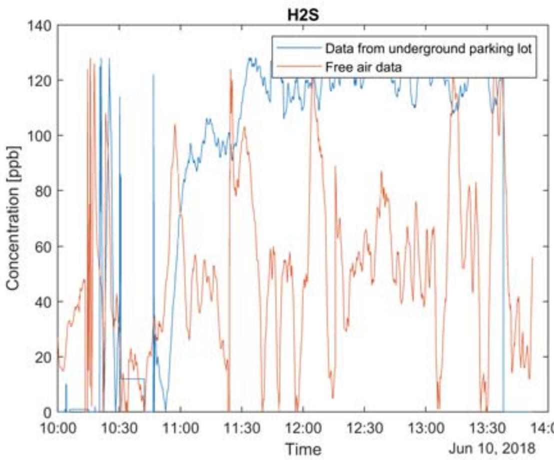

Fig. 11. H_2S concentration measurements obtained using the digital sensor node inside an underground parking lot and outdoors.

IV. Conclusion

In this paper, a low-cost environmental monitoring system based on wireless sensor networks (WSN) was designed and implemented under the concept of IoT. A web application was created to be able to visualize the data measured by the sensors in real-time as well as historical measurements. The proposed topology prevents data from being lost when there is no Internet or network connection, since the data has been stored locally within the data processing nodes. The system proposed is also expandable since the number of sensors for different gases can be increased for monitoring a bigger area.

Both low cost analog and digital sensors were used with Arduino-based hardware sensor nodes to make measurements of harmful gases and particulate dust material. Initial tests showed that the system works properly, and all sensors used were capable of producing measurements within the expected span range. Environmental measurements were obtained around an university campus in the Cumbayá, inside and outside an underground parking lot where the concentrations of the gases were within the range of measurement of the low cost sensors used. Although low-cost sensors have a limited working range, they could still detect anomalous situations when the gas concentration is above a tolerable limit.

For future work we plan to increase the number of nodes, use more precise sensors, and experiment with an Odroid electronic boards as gateway in order to integrate the proposed solution with an analytical data system based on big data tools. Additionally, comparison with reference instruments is also needed.

V. Acknowledgment

The authors would like to thank the financial support of the Ecuadorian Corporation for the Development of Research and the Academy (Corporación Ecuatoriana para el Desarrollo de la Investigación y la Academia, RED CEDIA) during the development of this work, under Project Grant CEPRA-XI-2017-13.

References

[1] World Health Organization, “7 million deaths linked to air pollution,” Department of public Health, Environmental and Social Determinants of Health (PHE), Issue 63, March 2014.

[2] S. Kumar, D. Katoria. “Air Pollution and its Control Measures,” International Journal of Environmental Engineering and Management, Vol. 4 (5), pp. 445-450, 2013.

[3] S. Choi, N. Kim, H. Cha, R. Ha, “Micro Sensor Node for Air Pollutant Monitoring: Hardware and Software Issues,” Sensors, Vol. 9(10), pp. 7970-7987, 2009.

[4] R.V. Prasad, M.S. Baig, R. K. Mishra, P. Rajalakshmi, U.B. Desai, S.N. Merchant, “Real Time Wireless Air Pollution Monitoring System,” ICTACT Journal on Communication Technology: Special Issue on Next Generation Wireless Networks and Applications, Vol. 2 (2), pp. 370-375, 2011.

[5] S. Devarakonda, P. Sevusu, H. Liu, R. Liu, L. Iftode, B. Nath, “Real-time air quality monitoring through mobile sensing in metropolitan areas,” Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing. ACM, pp. 15:1-15:8, 2013.

[6] B. Guanochanga, R. Cachipuendo, W. Fuertes, D. S. Benítez, T. Toulkeridis, J. Torres, C. Villacís, F. Tapia, and F. Meneses, “Towards a Real-Time Air Pollution Monitoring Systems Implemented using Wireless Sensor Networks: Preliminary Results,” 2018 IEEE Colombian Conference on Communications and Computing (COLCOM), Medellin, Colombia, pp. 1-5, 2018.

[7] P. Baltrenas, D. Kaziukoniene and M. Kvasauskas, “Air pollution at parkinglots of Vilnius,” Journal of Environmental Engineering and Landscape Management, Vol.12 (1), pp. 38-43, 2004.

[8] Kane International, “What are safe levels of CO and CO_2 in rooms?,” [Online] Available: https://www.kane.co.uk/knowledge-centre/what-are-safe-levels-of-co-and-co2-in-rooms.

[9] NASA, Global Climate Change, 2018. Available online https://climate.nasa.gov/

[10] Secretaria de Ambiente Quito, CO_2 April Monthly report, 2018 - Reporte mensual de Abril de 2018 de CO_2, 2018.

[11] NO-IP.com. Dynamic DNS Update Client (DUC) for Windows. [Online]. Available: https://www.noip.com/download

[12] Digi International, XBee/XBee-PRO S1 802.15.4 (Legacy). [Online]: Available at: www.digi.com/resources/documentation/digidocs/pdfs/90000982.pdf

[13] Atmel.8-bit AVR Microcontrollers Atmega328P. [Online]: Available: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf

[14] MariaDB. What is it? [Online] Available: https://mariadb.com/sites/default/files/MariaDB.pdf

[15] Apache Sotware Foundation. Apache HTTP Server Documentation Version 2.4. 2015. [Online] Available: https://archive.apache.org/dist/httpd/docs/httpd-docs-2.4.16.en.pdf