QoS Enabled IoT Based Low Cost Air Quality Monitoring System with Power Consumption Optimization

Virendra Barot[1]*, Viral Kapadia*[2]*, Sharnil Pandya*[3]

[1]*I. T Department, Government Engineering College, Bhavnagar, Gujarat, India*
[2]*Department of CSE, Faculty of Technology and Engineering, The M S University of Baroda, Vadodara, India*
[3]*Symbiosis Centre for Applied Artificial Intelligence, Symbiosis International (Deemed) University, Pune, Maharashtra, India*
E-mails: *viren.rao82@gmail.com*   *viral.kapadia-cse@msubaroda.ac.in*
sharnil.pandya@sitpune.edu.in

Abstract: *Air pollution has emerged as a major concern of the current century. In recent times, fellow researchers have conducted numerous researches in the area of air quality monitoring. Still, air quality monitoring remains an open research area due to various challenges such as sophisticated topology design, privacy and security, power backup, large memory requirements and deployment of such systems at resource-constrained sites. The proposed research work is an attempt to address the issues of communication topology design, assessment of the Quality of Service (QoS) levels against accuracy, sensing throughput and power consumption optimization. In the undertaken work, the proposed IoT based Air Quality Monitoring system has been deployed at indoor and outdoor sites to measure air quality parameters such as PM10, PM2.5, carbon monoxide, temperature and humidity. The proposed system is also tested at variety of quality of service levels at the indoor and outdoor sites. The conducted experiments have also recorded accuracy in terms of reliable delivery of the messages under employed protocol.*

Keywords: *Air quality monitoring, Internet of Things, Message Queue Telemetry Transport protocol, smart city, ESP8266.*

1. Introduction

With the fast-developing economy, an increase in industrial park construction, and production processes, air quality monitoring has become an alarming issue for the society. In general, air pollution can be described as the contamination of surrounding by various gaseous, liquid and solid wastages, which can cause enormous effects on human health, the welfare of plants and animals [1]. These pollutants create health risks such as sinusitis, asthma, allergic, nasal irritation, central nervous system symptoms, deices and many more. According to the reports of [2], household and

outdoor air pollution had resulted in 4.3 and 3.7 million deaths. The Air pollution Index (AQI) monitoring has emerged as a critical issue in developing countries such as India. Recently, in Delhi, AQI has crossed the bar of 1000. According to [3], significant concern for air pollution is the presence of high concentration of Particulate Matter (PM) in atmosphere. PM2.5 can cause harmful biological effects to various human organs such as lungs, and nervous system [4, 6].

2. Related works

Many researchers are actively involved in the air monitoring field due to their concern to the critical problem. L i, C h o u and P a n [7] have investigated a method of analyzing hourly data produced by 71 fixed monitoring stations distributed over Taiwan using neural networks. The conducted study lacked an air quality framework for air pollution data collection and monitoring. R e i s i n g e r et al. [8] have described a Differential Optical Absorption Spectroscopy based (DOAS-based) instrument for measuring air pollutants. R i c h a r d s et al. [9] have designed a high-throughput sensor network to address the issue of real-time urban air pollution monitoring using Generic Ultraviolet Sensors Technologies and Observations (GUSTO) sensor technology. Their system had measured a variety of pollutants accurately in the form of ppb (parts per billion) at short intervals of 2 s. DOAS and GUSTO based projects have used a wired network setup of optical sensors due to the unavailability of technologies such as wireless sensor networks and the Internet of Things (IoT).

   With the advancements in wireless sensor networks, sensing technology and the internet of things, many researches attempted to implement air quality monitoring systems to measure various air quality parameters. A l-H a i j a, A l-Q a d e e b and A l-L w a i m i [10] and K u l a r a t n a and S u d a n t h a [11] have presented microcontroller-based systems using general-purpose gas sensors. However, the system did not facilitate to report the monitoring results in real-time. A l-a l i, Z u a l k e r n a n and A l o u l [12] and D e v a r a k o n d a et al. [13] have implemented the distributed monitoring system using GPRS network. The pollution data from sensor arrays have been transmitted to a local server. Social Cops is a company in India which has developed a project to measure air pollution in Delhi. This project comprises of sensors, GPS module and GPRS module. In this approach, data is transmitted with GPRS network [14]. GPRS is now considered outdated technology due to the latest advancements. A b r a h a m and X i n r o n g [15], 2014 developed system that measures indoor levels of Carbon Monoxide (CO), Ozone and Carbon dioxide (CO_2) using the Zigbee. However, they did not conduct experiments in outdoor environments. K u m a r and H a n c k e [16], F e r d o u s h and L i [17], B a c c o l et al. [18] and M o r a w s k a et al. [19] reported system of air pollution parameters. In all the approaches, Zigbee is used for communication. Zigbee has limitation of distance in relaying packets. Implementation requires higher nodes to relay data over a longer range. CNT-based gas sensor has been developed and the MSP430 controller was used to detect ammonia by S h a r m a et al. [20], they did not deploy any data transmission methods. T i w a r i, S a d i s t a p and M a h a j a n [21]

at Bits Pilani research lab in India have developed a system for monitoring methane using Raspberry Pi, which communicates the received data to the local web server. M a r q u e s et al. [22] have developed system to monitor LPG propane gas using MQ6 sensor connected to the laptop for data transmission. H o n g-d i et al. [23] have presented a methodology to predict PM2.5 with the available data from the surrounding monitoring stations. The approach did not consider direct measurements of air quality parameters. A wireless mobile air pollution monitoring application has been designed by D h i n g r a et al. [24], S u n et al. [25] and H u a n g et al. [26]. They used cloud-based services in a cost-effective manner using low-cost sensors. However, the system has not been tested in indoor and outdoor environments to check its reliability and accuracy. In this approach, generic air quality sensors were used instead of specific sensor for certain parameters.

In developing countries such as India, government is running the National Air Quality Monitoring Program [27]. Program is handling a network of 779 fixed stations. The government of India is funding projects related to air quality pollution looking at the alarming situations in metro cities. However, the program has failed to reach to urban and suburban areas. So there is an immediate need of establishing a complete air quality framework to monitor air pollution.

3. Necessity of a new IoT based air quality monitoring system

Air pollution has been a critical concern of the 21st century. Many efforts have been made to monitor air quality using IoT based technologies, still, the air quality monitoring system using IoT is an open research area because of the challenges of IoT discipline such as complex architecture, no standardization, less memory, power consumption, interfacing of sensors, reliable delivery and security. In the proposed research work, we have presented cost-effective indoor and outdoor pollution monitoring systems based on the real-time experiments conducted at two different sites in the polluted industrial cluster of India, which is Vadodara [28]. Firstly, system makes the use of lightweight protocol and compatible devices to transmit air quality parameters to the remote server without building complex network paradigm. Secondly, the presented work represents the implementation of power consumption optimization by switching a smart sensing node in a variety of five different modes by utilizing event-based transmission aspects. Also, system is implemented and tested with a variety of Quality of Service levels. The system also avails data logging in the data base for further analytics. A customized web interface and mobile application are designed to represent daily updates of pollutants at different indoor and outdoor sites.

Contributions of the presented work to the science community: 1) The air pollutant data of such air pollution monitoring system is facilitated with user-friendly web and mobile interface to assist citizens in avoiding emergency health conditions. 2) The logged monitoring data received from different deployment sites can assist an individual in determining the status and trends of air quality at various (intra and intercity) regions. The result section discusses the comparison of various air quality parameters for two deployment sites and represents identified critical hour durations

based on the hourly aggregated data. 3) The historically collected (long-term) data can assist future researchers in identifying characteristics, seasonal patterns, sources of pollution, agricultural and ecosystem effects and climate effects of air pollutants. 4) Distribution of pollutants at various indoor and outdoor sites can be predicted from the historical data logged by such system.
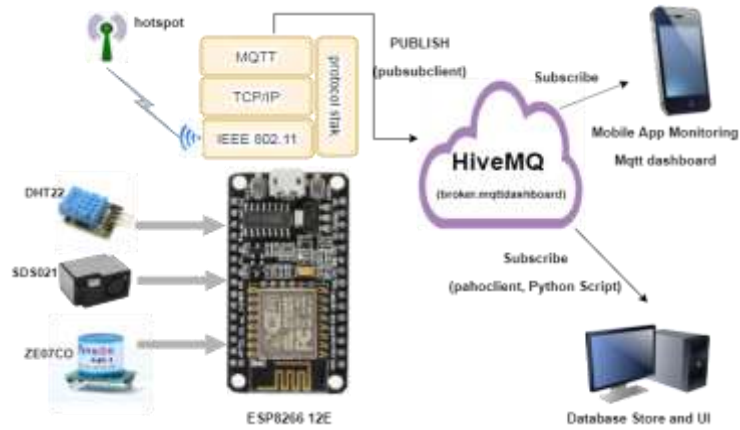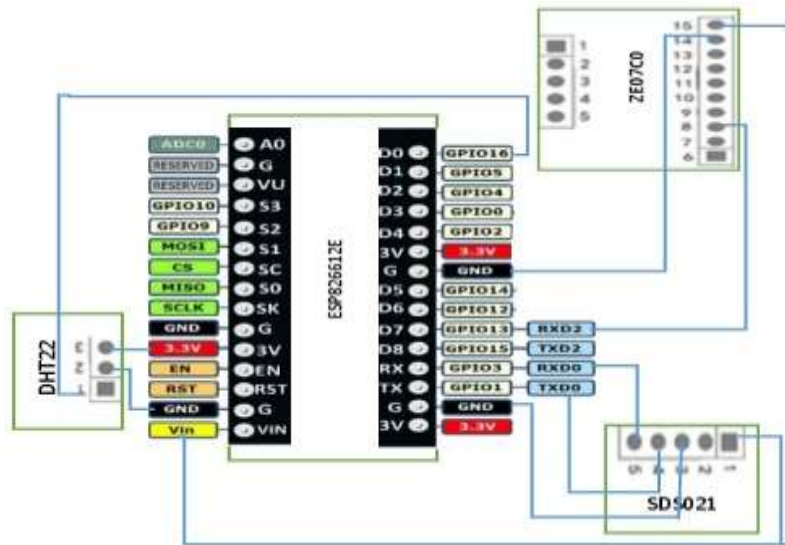

Fig. 1. Topology design

Fig. 2. Detailed circuit design of sensor interfacing

The proposed article is organized as follows: Section 4 presents the details of a controller and sensors. Section 5 presents the design and experimental setup used in conducting various experiments at indoor and outdoor sites and also discusses the detailed workflow of the proposed system and Section 6 discusses results of air quality monitoring system. The concluding remarks, future enhancements discussions have been carried out in Section 7.

4. Details of a controller and sensing units

4.1. ESP8266 12E/NodeMCU controller

ESP8266 12E/NodeMCU is an open-source cost-effective IoT development platform. This module is equipped with a 32-bit microcontroller with 4 MB memory for storing sensing programs. The "NodeMCU" requires 3.3 V power supply. NodeMCU board is consisting of GPIOs, I2C, UART, ADC, and PWM pins, enable it to connect with a variety of sensors. There are a plenty of IoT development boards/processing units available, selection of processing unit affected by inherent constraint of application domain and budget. Cost is very important aspect considering vast deployment of sensor monitors needed in air quality monitoring system. Arduino and ESP8266 12E/NodeMCU are most economic options as can be seen from Table 1. However, both processing units contain microcontroller and comparatively less processing memory than other alternatives. In our application, sensing nodes do not involve any edge computing (analytics at data origin) task that demand for more processing power. The task of sensor node is to read sensor data at employed frequency and transmit over IEEE 802.11. The NodeMCU development board is available with Wi-Fi capability (IEEE 802.11 /b/g/n) and full TCP/IP stack needed to implement application layer protocol.

Table 1. Existing major processing units for building sensing layer

Name	CPU (memory, Weight)	Price (RS)	Analog I/O pins	Digital I/O pins
Raspberry Pi	Quad Core1.2 GHz Broadcom BCM2837 64 bit Processor(1 GB, 42 g)	3200	0	14
Arduino Uno	ATMEGA 328 microcontroller (2 Kb, 25 g)	700	6	14
Arduino Due	Atmel SAM3X8e 32-bit ARM cortex controller (96 Kb, 30 g)	1500	12	54
Node MCU	Tensilica XtensaLX106 32-bitcontroller WI-Fi SOC integrated (4 MB, 8 g)	400	1	17
BeagleBone Black	AM335x, 1 GHz ARM Cortex A8 Processor (512 MB, 40 g)	6000	6	14
Udoo (Quad)	Freescale MX6 Quad, 4×Cortex™ A9 core @ 1GHz processor With Atmel 32 bit cortex (1 GB, 150 g)	9000	14	62+14
Intel Galileo Gen 2	Quark ™ X1000 32-bit 400 MHz (256 MB, 370 g)	7000	6	14

4.2. SDS021 particulate matter sensor

The SDS021 sensor works on the principle of laser scattering, which starts functioning whenever a particle passes through the detection area to measure the 2.5 µm and 10 µm particles. The SDSO21 sensor module contains a fan for keeping consistent wind stream over the detection chamber. The module gives output using the UART protocol with a bit rate of 9600 bits per 1 s. Detection range is 0.0-999.9 µg/m^3. The sensor works at 5 V operating voltage, [Vin] of Esp8266 12E is sufficient for drawing the required current. Table 2 shows the byte significance of the data frame received on the UART port of a controller from the SDS021 sensor.

126

4.3. Carbon monoxide (ZE07-CO-sensor)

ZE07-CO module depends on an electrochemical principle to measure the Carbon Monoxide. The sensor works at 5~12 V with the 9600-bit rate for UART output which measures in the range 0-500 ppm. Table 3 shows the byte significance of the data frame received on the UART port of ESP8266 12E from a sensor module.

4.4. DHT22 temperature and humidity sensor

DHT22 is a low-cost digital temperature and humidity sensor. The sensor gives measured value as an output (in the form of a calibrated digital signal on the data pin). The power requirement of the sensor is 3-5 V.

Table 2. UART data frame for SDS021sensor

Byte number	Significance	Byte number	Significance
0	Message Header (AA)	5	DATA4-PM10 high byte
1	Command ID (CO)	6	DATA5-IDbyte1
2	DATA1-PM2.5 low byte	7	DATA6-IDbyte2
3	DATA2-PM2.5 high byte	8	Checksum
4	DATA3-PM10 low byte	9	Message Tail (AB)

Table 3. UART data frame for ZE07-CO sensor

Byte number	Significance	Byte number	Significance
0	Start byte (0×FF)	5	Concentration CO low byte
1	Gas type (0×04)	6	Full range high byte
2	Unit – ppm (0×03)	7	Full range low byte
3	No of decimal	8	Checksum
4	Concentration CO high byte	–	–

5. System design and experimental setup

Internet of Things (IoT) is a network of "things" (objects such as software agents, hardware etc.) where things are empowered with network for communication without active human interaction. IoT has the potential to monitor air pollution by providing real-time updates related to sudden changes in the air quality [5]. There are various kinds of "Things" used in the proposed system that includes NodeMCU (controller), HiveMQ (cloud broker), Sensors, Python Script (Paho MQTT subscriber) and Android. These things coordinate with each other for the purpose of air quality parameter collection, transmission, storage and retrieval individually towards the implementation of the complete monitoring system. Also, the system uses Message Queue Telemetry Transport broker (MQTT-broker) architecture for the proposed system, with limited bandwidth requirement. The underlying architecture with HiveMQ broker provides an automatic authentication of publishing device as explained in Subsection 5.2. On availability of sensing data from a publisher (things) at broker site, HiveMQ broker will notify and deliver data to all the subscribers (things) registered for those data. The distribution of the site-specific data from the smart node to relevant subscribers is handled by the broker under an IoT based architecture. The underlying IoT based architecture is required to implement a

complete framework starting from data collection (at deployment sites) to logging and representation of the air quality parameters on different platforms such as web and mobile interface at the same time by providing semantic interoperability between various objects.

5.1. Layered architecture of IoT based monitoring system

The layered architecture of system has been classified into five layers. Fig. 1 shows the topology design of the proposed system, and Fig. 2 represents the detailed circuit design of the interfacing of sensors with a controller ESP8266 12E.

- **Physical sensing layer.** The physical sensing layer consists of a variety of modules such as SDS021 Particulate Matter Sensor [31], ZE07-CO carbon monoxide sensor [30] and DTH22. These sensor modules are capable of measuring air quality parameters such as PM10, PM2.5 CO, temperature, and humidity. These sensor modules are connected to a microcontroller ESP8266 12E. The microcontroller is equipped with IEEE 802.11 and TCP/IP stack.
- **Communication and Networking layer.** Communication and networking layer is responsible for establishing interfaces between a physical sensing layer, a cloud MQTT broker, a storage server and web and mobile interfaces. The Wi-Fi access point is essential for the transmission of sensor data using Message Queue Telemetry Transport (application layer protocol) over internet (with underlying network protocol-TCP).
- **Cloud service layer.** It stores the data received from sensors nodes or motes. Controller, along with the equipped sensors, forms a smart node that can be identified as mqtt publisher. Publisher (smart node) publishes the air quality data from an individual sensing site to MQTT broker with topics specific to air pollutant parameters. This layer is also responsible for device authentication based on a unique client ID provided by node during initialization process utilizing the ClientID field of MQTT message field as shown in Fig. 4.
- **Data processing layer.** This layer is responsible for fetching data from cloud broker, process raw data of messages and store data in format compatible with application layer requirement and suited for further analysis. MQTT subscriber is deployed on a server that is responsible for fetching observed data of a remote site available at broker. Data received from broker is handed over for rendering to graphical user interface and stored in a database for analytics.
- **Application layer.** The application layer provides real-time air quality monitoring updates via web and mobile GUI interfaces.

5.2. Detailed working flow of IoT based air quality monitoring system

Fig. 3 shows detailed working flow of the system. Publish-subscribe pattern requires a message broker in between a publisher and a subscriber in MQTT protocol. HIVEMQ cloud broker is used as a public cloud broker. MQTT publisher is implemented on NodeMCU, which sends sensor data in the form of MQTT messages to an HIVEMQ cloud broker. Data acquisition at a server (MQTT subscriber) has been implemented using Python language packages "paho-mqtt" and

128

"paho.mqtt.client". Moreover, the sensing layer is programmed using the Arduino IDE, which is responsible for reading data from the sensors using API calls designed for air quality monitoring system.
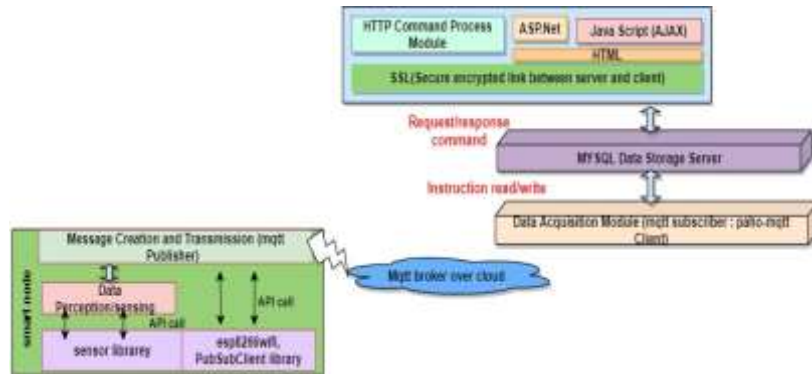

Fig. 3. Detailed workflow of IoT based air quality monitoring system

Fig. 4. MQTT message format

- **Message Queue Telemetry Transport (MQTT) protocol.** MQTT is an ISO standard open-source publish-subscribe based lightweight messaging protocol. It works on a TCP/IP network layer. It has been developed for providing connections with remote locations where available network bandwidth is limited for resource-constrained devices with high latency output. Fig. 4 represents the MQTT connect message format, ClientID is utilized for device identification and QoS field for setting QoS level in the proposed framework.
- **Power consumption optimization.** Power usage has been optimized by switching to a sleep mode at appropriate intervals instead of hardware optimization. In sleep mode, device draws very less power compared to an active mode. ESP8266 12E controller facilitates three types of sleep modes: modem, light and deep-sleep [29]. In a light-sleep mode, the controller draws around 0.9 mA compared to 120 mA in an active mode. So, to achieve power consumption optimization, controller has been switched to a light sleep mode as and when it is not acquiring data from the sensors. While the controller is reading data from the sensors, the RF transceiver is kept on an idle mode to reduce the power consumption, which can be switched on whenever transmission restarts. As shown in Fig. 2, particulate matter sensor module SDS021 and CO sensor module ZE07C0 will be connected to [Vin] of a controller. [Vin] is the direct output from the supply source, which provides

129

power back up to an Esp8266 12E controller. Whenever a controller is in a light sleep mode, [Vin] supply will not remain idle and it will supply the necessary power to connected sensors. In general, it is essential to keep the CO sensor in the active mode most of the time because it requires a 3-minute warm-up time whenever it will be switched on and off [30]. Therefore, it is not advisable to switch-off the CO sensor during the communication phase. On the contrary, it is possible to power down particulate matter sensor module SDS021 by switching it to a sleep mode.

The sensing operation of the system can be divided in five phases as shown in Table 4. Initially, system will remain in the P1 phase where controller will switch into a light sleep mode for the duration of 60 s. At this stage, SDS021 sensor module will remain in a hibernation mode and DHT22 sensor module will remain in a sleep mode. After the expiration of a timer, a controller will switch into an active mode from a light sleep mode. Whenever a proposed system will enter into the P2 phase, DHT22 sensor module will wait for 10 s for stabilization and recording of temperature and humidity data. In the case of P3 phase, a controller will activate the SDS021 sensor module. It will record particulate matter data for the duration of 10 seconds and switch into a sleep mode. After recording CO data, the proposed system will enter into the P5 phase from the P4 phase. Eventually, a publisher turns on transceiver and transmits the gathered data using MQTT messages. The average power usage of node can be calculated as:

$$(1) \qquad P = \left[\frac{I_{lsm} \times T_{sleep} + I_{active} \times T_{active}}{T_{si}} + I_{CO}\right] \times V_{in},$$

where I_{lsm} is current drawn in light sleep mode by the smart node, T_{sleep} is time duration in sleep mode, I_{active} is current drawn in an active mode, T_{active} is time that includes data collection and stabilization time of sensors, T_{si} is the total time/sampling interval that includes node operation and sleep period, I_{CO} is the current drawn by Carbon Monoxide sensor which is never at rest and V_{in} is the input voltage. A smart node will remain in a sleep mode for 60 s out of around 90 s sampling interval. Smart node draws 0.9 mA in a light sleep mode and 120 mA in an active mode. The average power consumption of a smart node for the period of 90 s is 203 mW plus consumption of CO sensor (CO sensor is never at rest) by applying power optimization scheme which is around 600 mW plus consumption of Carbon Monoxide sensor without power optimization.

Algorithm 1 represents the step-wise functioning of a MQTT publisher. During the process of device initialization, a smart node is registered with broker with a unique CliendID, which will be stored at a broker and used for authorization of an individual node for future communications. Algorithms 2 and 3 describe procedures to measure the values of PM2.5, PM10 and CO parameters as per the using Equations (2)-(6). The proposed algorithms also perform integrity checking of received data frames to avoid any sampling error at the physical level.

Table 4. Sensing cycle phases for various components of sensing unit during parameter reading

Phases	Controller	DHT22	SDS021	ZE07CO	Wi-Fi
P1	light sleep	Power down	sleep	active	Off
P2	active	active	sleep	active	Off
P3	active	active	active	active	Off
P4	active	active	sleep	active	Off
P5	active	active	sleep	active	On

130

Algorithm 1. MQTT Publisher Process Flow

Step 1. Registration of a Smart node with an MQTT broker using unique ClientID.

Step 2. The controller (publisher) switches on the PM sensor, reads the value from three sensors, and again switches the PM sensor in a hibernation mode.

Step 3. Creating an MQTT message by assigning value read in Step 2 to the relevant sub-topic.

Step 4. Publisher connects to the MQTT broker authenticated with unique ClientID.

Step 5. The publisher publishes the mqtt message created in Step 3 with the topic set for an individual site.

Step 6. Controller switches to a sleep mode.

Step 7. Auto awaking of a controller after the expiration of a timer then go to Step 2.

Algorithm 2. PM2.5 and PM10 Parameter Measurement Process (with Integrity Check)

Input: Pin numbers

Output: PM2.5 an PM10 Level

Step 1. Reading serial port of a sensor, on the accumulation of continuous ten bytes clearing of a local buffer and searching for a message header $0\times AA$

Step 2. Once the required header of frame is found in Step 1, checking for the second byte of the data frame which must be $0\times CO$ to get the reading in $\mu g/m^3$ unit.

Step 3. Read next second byte in μ, third byte in α, fourth byte in β, fifth byte in γ.

Step 4. Read the eighth byte and perform integrity check as per Equation (4). If integrity is not maintained, go to Step 6, otherwise, go to Step 5.

Step 5. Calculate PM2.5 (mg/m³) and PM10 (mg/m³) values using Equations (2) and (3).

Step 6. If integrity is lost, then go to Step 1.

$$(2) \qquad \text{PM2.5} = \big((\alpha \times 256 + \mu)\big)/10,$$
$$(3) \qquad \text{PM10} = \big((\gamma \times 256 + \beta)\big)/10,$$
$$(4) \qquad \text{Checksum} = \sum_{i=1}^{6} \text{DATA\_BYTE}_i,$$
$$(5) \qquad \text{Checksum} = \big(\sum_{i=1}^{7} \text{DATA\_BYTE}_i\big)' + 1,$$
$$(6) \qquad \text{CO(PPM)} = \big((\psi \times 256 + \omega)\big)/10.$$

Algorithm 3. CO Parameter Measurement Process (with Integrity Check)

Input: Pin numbers

Output: CO Level

Step 1. Read the serial port of a CO sensor and on availability of start byte $0\times FF$ go to Step 2.

Step 2. Read the second byte and the third byte if bytes are matched with 0×04 (gas type CO) and 0×03 (unit is in PPM), then go to Step 3 otherwise go to Step 1.

Step 3. Read next coming the fourth byte in ψ and fifth byte in ω.

Step 4. Read the eighth byte and check for integrity using Equation (5). If the match is successful, then go to Step 5 otherwise go to Step 6.

Step 5. Calculate CO values using Equation (6).

Step 6. If integrity is lost, then go to Step 1 again.

5.3. Prototype design and deployment

Fig. 5a depicts the prototype design of the proposed system. The system was deployed at indoor and outdoor sites: rooftop (outdoor) and home (indoor). Figs 5b and 5c show the proposed system deployment at home (indoor) and rooftop (outdoor) respectively.

| (a) | (b) | (c) |

Fig. 5. Prototype (a); deployment at home(indoor) (b); deployment rooftop (outdoor) (c)

6. Results and discussion

6.1. Visualization and data analytics

The installation and testing of the proposed system has been done at indoor and outdoor sites. The indoor site experiments were conducted at home (Site 2), and outdoor experiments were done at Vadodara, India (Site 1). In the conducted experiments, HiveMQ MQTT broker was used. MQTT messages were published with QoS levels 0 and 1. Fig. 6 depicts the real-time graphical user interface of the observed data received by MQTT subscriber from smart node deployed at outdoor site. Real time graphs of sensor data implemented using "Matplotlib" library inside MQTT subscriber script using Python. As shown in Fig. 6, obtained results display observed data of CO, PM2.5 and PM10. Observation data has been received at an interval of 60 s and displayed by updating every one-hour sliding window. Along with real time graph rendering of observation parameters MQTT subscriber fetches readings from the sensing units (available with broker) and also stores these data in MYSQL for analysis, however it can be stored in any database of choice.
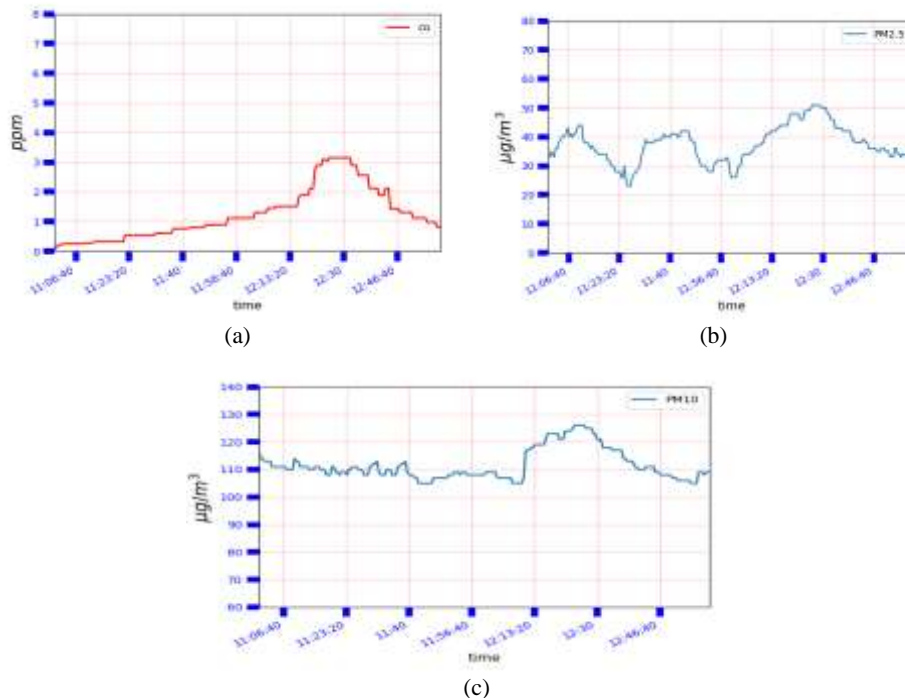
Fig. 6. Some snaps of GUI cum graphs generated for live monitoring of air quality parameters at home rooftop (Site 1 – outdoor) per one-hour sliding window: CO (a); PM2.5 (b); PM10 (c)

Fig. 7a shows distribution and comparison of observed pollutants based on the data collection of day time for indoor and outdoor deployments. PM2.5 values vary in the range of 23 to 51 μg/m^3 with the median of 35, PM10 varies in range 98 to a maximum value of 126 μg/m^3 with observed median of 110. The maximum observed value of CO at Site 1 (outdoor) is 3.15 ppm. The obtained readings are recorded and analyzed for 6 hours for Site 1 (outdoor) and Site 2 (indoor). From the obtained results of Site 1, it is observed that the temperature is varying in the range of 32.5 °C to 34.8 °C, and humidity is varying in the range of 24.1 % to 31.1 %. In the case of Site 2 (indoor), median values for CO, PM10, and PM 2.5 are recorded around 0.38 ppm, 80 μg/m^3 and 24 μg/m^3 respectively. During day-time, the maximum value of PM10 that has been recorded is 102 μg/m^3 for Site 2 (indoor), which is 23% lesser compared to maximum value recorded at Site1. Figs 7b and 7c show the observed data of Particulate matter 10, 2.5 and CO during day time for outdoor and indoor deployments. They also depict the differences in the air quality level between both sites. It can be observed that the indoor environment site (Site 2) is less polluted compared to the outdoor site (Site 1). Moreover, the observation of Site 1 (outdoor) also indicates that it has recorded a higher concentration of PM2.5, PM10 and CO air pollutants during mid-day. The observations of Site 1 also pinpoints that the recorded value of parameters PM10 and PM2.5 have fluctuated around 110 and 35, respectively.

(a)   (b)
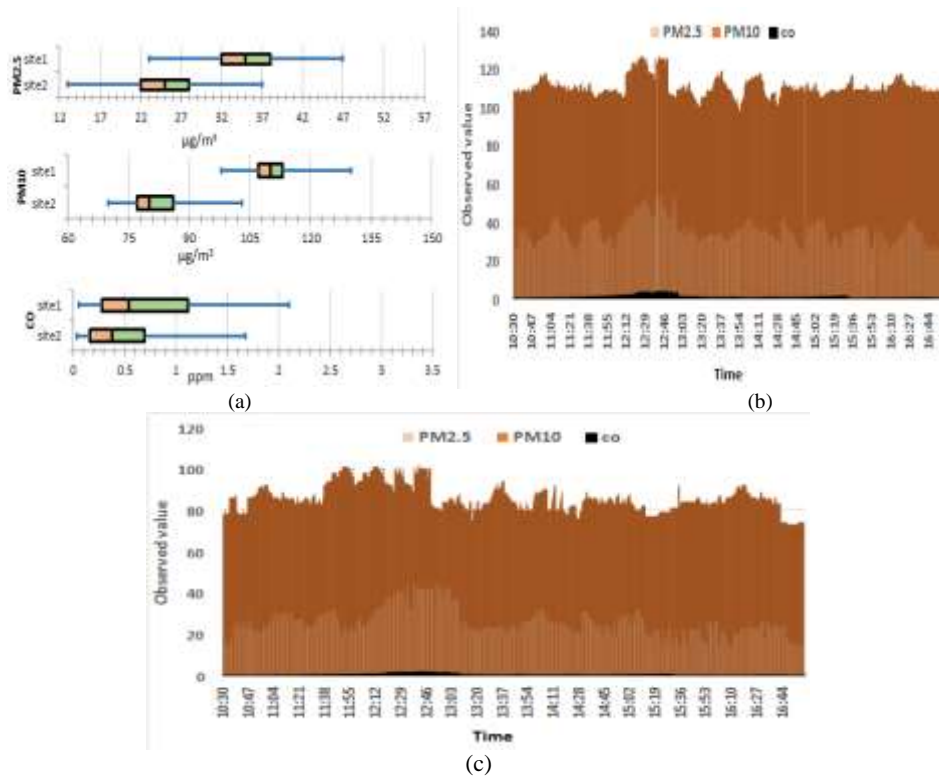

(c)

Fig. 7. Distribution comparison of observed Parameters (a); observation of PM2.5, PM10 and CO at rooftop during day time (Site 1 – outdoor) (b); observation of PM2.5, PM10 and CO at home during day time (Site 2 – indoor) (c)

Figs 8a and 8b represent aggregated values of PM10 and PM 2.5 for three days. The PM10 and PM2.5 values were aggregated every three hours. Figs 8a and 8b depict that the outdoor site (Site 1) has recorded very high pollution index during the slot of 12-15 hours for all the three days. Fig. 9 represents a few mobile interface snapshots of PM2.5 and PM10 parameters for Site 1 (outdoor site).
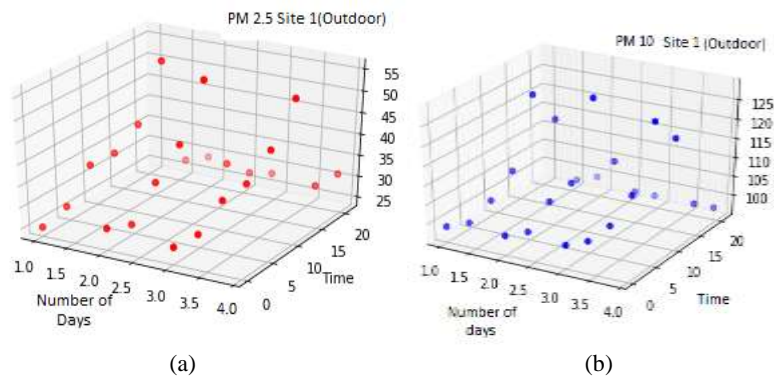

(a)   (b)

Fig. 8. Scatter plot of PM2.5 at rooftop (Site 1 – outdoor) over 3 days (a); scatter plot of PM10 at rooftop (Site 1 – outdoor) (b)

134

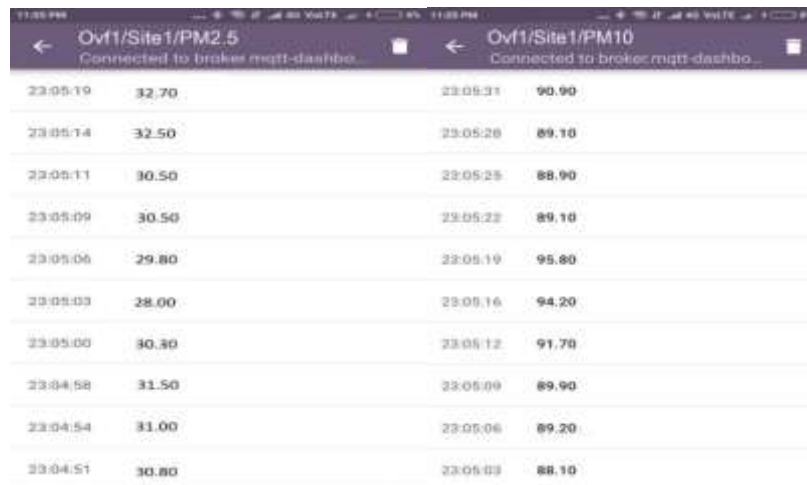| 23:05:19	32.70		23:05:31	90.90
23:05:14	32.50		23:05:28	89.10
23:05:11	30.50		23:05:25	88.90
23:05:09	30.50		23:05:22	89.10
23:05:06	29.80		23:05:19	95.80
23:05:03	28.00		23:05:16	94.20
23:05:00	30.30		23:05:12	91.70
23:04:58	31.50		23:05:09	89.90
23:04:54	31.00		23:05:06	89.20
23:04:51	30.80		23:05:03	88.10

Fig. 9. Screenshots of Environment Monitoring System

6.2. Quality of Service (QoS) and system performance under periodic transmission

IoT based ecosystem such as air quality monitoring systems are transporting real-time parameters and updates to the remote server. Such system can also provide threshold based notifications based on received data. In such real time system reliable delivery of the messages (in terms of messages transmitted Vs message received) to the subscribers is very important parameter. Reliable delivery or accuracy is one of the metrics representing Quality of Service provided by system. Thus implementation of QoS adds value to such diverse system by providing performance, visibility and usability of the services offered. Vey few efforts have been attempted to implement and assess performance of implemented system under complex architecture design. Experiments discussed below show that the QoS implementation costs more delay due to retransmission overhead. So tradeoffs between end-to-end delay Vs reliable delivery (accuracy) are clear.

There are three levels of QoS in MQTT. Level 0 is least reliable communication out of the three levels. It is a fire and forgets the kind of communication where the publisher sends the message and never checks for the delivery of the message at the destination. In QoS Level 1 publisher keeps the message until the acknowledgment using the PUBACK message from the broker. The publisher publishes the message again if PUBACK is not received from a subscriber. QoS Level 2 ensure exactly one-time message delivery at the subscriber. At this point, a sender and a receiver will use various message identifiers for the synchronization of delivery. A publisher sends the message again with a duplicate flag if PUBREC is not acknowledged.

Fig. 10 shows the performance of the system at QoS Level 0 and Level 1 for the duration of 12 hours. In the case of QoS Level 0, packet dropout ratio has been recorded around 1.97% and for QoS, Level 1 the packet dropout ratio has been recorded around 0.94 % at Site 1 (indoor). To analyze effect of QoS level on end-to-end delay, we simulate the publisher environment using MQTT-JMeter (a tool from Apache). MQTT plugin configured with JMeter, can serve to perform testing in which simulated clients register to the broker. Fig. 11 shows the end-to-end delay

corelation with transmission rate under various QoS levels during the simulation. It is observed that the end-to-end delay will also increase whenever a packet is transmitted from a lower QoS level to a higher QoS level due to retransmission and acknowledgment overhead. So the selection of the QoS level becomes an essential criterion for mitigating end-to-end delay and packet loss ratio. As shown in Fig. 10, the accuracy values recorded at Site 1 (outdoor) and Site 2 (indoor) are around 98% and 96% respectively. The accuracy values have been calculated considering the number of transmitted and received packets at indoor and outdoor sites. Accuracy and dropout parameters are representing the system performance in terms of reliable delivery of MQTT messages including sensing parameters. However, it cannot be interpreted as performance measurement or accuracy measurement of air quality pollution sensors. As shown in Fig. 12a and 12b, it is observed that the recorded average throughput of the sensing node is around 4.28 for Site 1 and 4.6 for Site 2 for the duration of 6 hours.

Fig. 10. System performance over an MQTT protocol: rooftop (Site 1 – outdoor) and home (Site 2 – indoor)
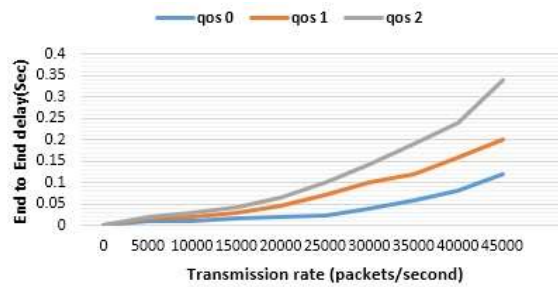

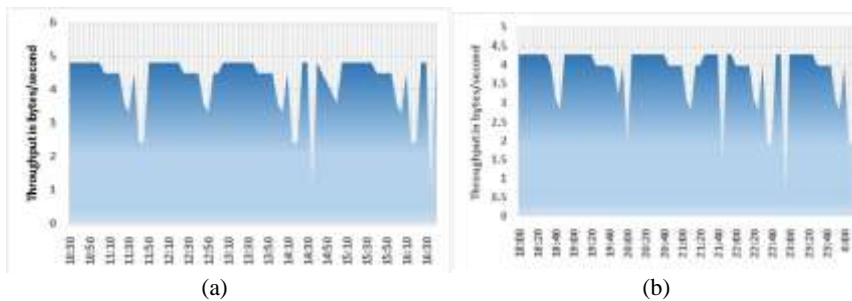Fig. 11. End-to-end delay against QoS level in simulation

| (a) | (b) |

Fig. 12. Throughput of the Sensing Unit: home (Site 2 – indoor) (a); rooftop (Site 1 – outdoor) (b)

136

6.3. Event based transmission experimentation towards power usage reduction

Average power usage of smart node relies on number of measurements taken periodically and the number of transmission of measured data. As shown in Table 4 the transceiver is kept off and only activated when transmission occurs. Power consumption can still be reduced if number of transmissions are reduced, in addition to the power optimization scheme shown (Table 4) and explained in Section 5. The aperiodic transmission scheme, updates receiver (transmits data) only if significant change in current (recent) measurement is observed compared to the last or previous measurement. The well-known techniques include Send on Delta (SoD) [32], Send on Area (SoA) [33] and Send on Prediction (SoP) [34]. SoD technique updates the measurement when difference between current measurement and previous one reaches a set threshold. In SoA the condition is to transmit when integral of the difference between recent and previous measurement over the interval reaches the set threshold. SoP calculates prediction value based on few previous terms (measured values) and difference between prediction and current value is utilized for transmission decision.

(7)
$$| \overline{X_t} - \overline{X_{t-1}} | > \delta \, \overline{X_{t-1}} ,$$

where

$$\overline{X_t} = \frac{\sum_{k=t-(N-1)}^{k=t} X_k}{N} \quad \text{and} \quad \overline{X_{t-1}} = \frac{\sum_{k=t-(N-1)}^{k=t-1} X_k}{N-1}.$$

Proposed system has been also experimented with event-based transmission to reduce overall transmission and ultimately reduction in power consumption. Event based transmission scheme uses condition given in Equation (7) for transmission decision. If the condition is satisfied transmission takes place otherwise not. Here $\overline{X_t}$ is the average of last N measurement including the latest measurement at time t and $\overline{X_{t-1}}$ is the average of previous $N-1$ measurement until time Step $t-1$ (excluding the recent measurement). If the change in the average due to recent measurement is more than δ percent then change is considered significant and transmission takes place. If any of the CO, PM2.5 and PM10 follows the given condition then the MQTT message is transmitted otherwise message transmission is skipped. It is possible that gradual and steady increase in measurement, never (or for longer period of time) makes the condition in Equation (7) to be true. In that case the measurement is not reported to receiver for very longer period of time. To avoid such scenario, we added one more condition, that is continuous nine transmission skipping (no transmission of measured value) will trigger one transmission on tenth measurement.

Table 5. Effect of event based transmission

Number of transmissions	Periodic	Event based transmission (with $N=3$, $\delta=0.10$)
Day one	240	62
Day two	240	57
Day three	240	73

Table 5 shows the number of transmission under the employed scheme experimented for 6 hours a day. Number of transmission is counted considering number of subscribed MQTT messages available at the subscriber. Table 5 indicates very less number of transmitted messages in event based transmission scheme.

137

Reduced number of transmission results in power consumption reduction. Employability of such scheme depends on tradeoff between the requirements of tolerance in micro level changes versus power consumption.

7. Conclusion and future enhancements

Recent Advancements in the field of Internet of Things, low-cost smart sensor nodes, and embedded systems have motivated fellow researchers to carry out in-depth researches in the area of Air Quality Monitoring. In the present research work, we have proposed a cost efficient economic IoT based Air Quality Monitoring system to measure a variety of air pollutants for indoor and outdoor environments. The proposed system consists of an air quality smart sensing node equipped with a controller and sensors, which are connected to an MQTT public cloud broker. The smart sensing node is equipped with a WiFi enabled ESP8266 12E controller which is suitable for the implementation of an MQTT protocol broker architecture. In the proposed research work, we have utilized DHT22 digital sensor for measuring temperature and humidity readings, electrochemical sensor module ZE07-CO for carbon monoxide, and SDS021 sensor module for PM10 and PM2.5 readings. Furthermore, customized libraries have been designed and developed for obtaining readings from the sensors. The sampled data for each air quality parameter has been sent periodically to a remote HiveMQ public broker. The data from a HiveMQ broker have been subscribed to a server for generating graphical results. The obtained results are stored in an MYSQL database.

In the proposed research work, we have made an attempt to address the issues of communication topology design, quality of service, accuracy and power consumption optimization. Furthermore, in the undertaken study, the proposed IoT based Air Quality Monitoring system has been deployed at two sites to measure air quality parameters such as PM10, PM2.5, carbon monoxide, temperature and humidity. The system also reduces power consumption by switching a smart sensing node into five different modes. The proposed system is also tested at a variety of quality of service levels at two sites. The goal of QoS level demonstration and testing is to justify the data delivery accuracy of the system, which is very critical for real time air quality monitoring. The obtained results have recorded accuracy of 98% at Site 1 (outdoor) and 96% at Site 2 (indoor), accuracy in terms of reliable delivery. Moreover, a novel event based transmission scheme is implemented to reduce number of transmission. However exact power saving in terms of absolute power unit is not represented, rather reduction in transmission of MQTT messages under the employed scheme is represented. A web and mobile interface has been designed to display the measured parameters in near real-time. In the future, the proposed system can be deployed at multiple sites to carry out a detailed analysis of gathered data using big data and machine learning algorithms. Furthermore, we have represented QoS analysis which assists the proposed IoT based air quality monitoring system in the implementation of an adaptive algorithm, which facilitates auto switching between varieties of QoS levels as and when required. The auto-switching process highly depends on a parameter such as critical pollution duration.

Summary points:

- In the undertaken study, a complete economic air quality monitoring system has been proposed for indoor and outdoor environments. The proposed system facilitates sensor interfacing with physical layer integrity checking, complex topology design for indoor and outdoor environments with real-time monitoring of air quality parameters using the web and mobile interfaces.

- In this study looking at the real time reporting requirement, the data delivery accuracy (performance) assessment of the proposed system has been tested at variety of QoS levels offered.

- In this proposed study, power consumption optimization aspects were covered, and an attempt has been made to provide a software-based solution.

R e f e r e n c e s

1. Z a n e l l a, A., N. B u i, A. C a s t e l l a n i, L. V a n g e l i s t a. Internet of Things for Smart Cities. – IEEE Internet of Things Journal, Vol. **1**, February 2014, No 1, pp. 22-32.
2. World Health Assembly, 69, Health and the Environment: A Draft Road Map for an Enhanced Global Response to the Adverse Health Effects of Air Pollution: Report by the Secretariat. 2015. Available: World Health Organization.
   https://apps.who.int/iris/handle/10665/252673.
3. WHO, Air Quality Guidelines for Particulate Matter, Ozone, Nitrogen Dioxide, and Sulfur Dioxide. Global Update, 2005.
   http://whqlibdoc.who.int/hq/2006/WHO_SDE_PHE_OEH_06.02_eng.pdf
4. H u, R., X. X y, X u S k, Y. W a n g, M. J i a n g, L. R. W e n, W. L a i. PM2.5 Exposure Elicits Oxidative Stress Responses and Mitochondrial Apoptosis Pathway Activation in HaCaT Keratinocytes. – Chin Med J., Vol. **130**, 2017, pp. 2205-2214.
5. A t z o r i, L., A. I e r a, G. M o r a b i t o. The Internet of Things: A Survey. – Computer Networks, Vol. **54**, 2010, Issue 15, pp. 2787-2805.
6. P o p e I I I, C. A., D. W. D o c k e r y. Health Effects of Fine Particulate Air Pollution: Lines That Connect. – Journal of the Air & Waste Management, Vol. **56**, 2006, Issue 6, pp. 709-742.
7. L i, S.-T., S.-W. C h o u, J. P a n. Multi-Resolution Spatio-Temporal Data Mining for the Study of Air Pollutant Regionalization. – In: Proc of 33rd Annual Hawaii International Conference on System Sciences, Maui, HI, USA, 2000, p. 7.
8. R e i s i n g e r, G. F r a s e r, P. J o h n s t o n, R. M c K e n z i e, W. A. M a t t h e w s. Slow-Scanning DOAS System for Urban Air Pollution Monitoring. – In: Proc. of 18th Quadrennial Ozone Symposium, 1996, pp. 12-21.
9. R i c h a r d s, M., M. G h a n e m, M. O s m o n d, Y. G u o. Grid-Based Analysis of Air Pollution Data. – Ecological Mod., Vol. **194**, 2006, No 1-3, pp. 274-286.
10. A l-H a i j a, Q. A b u, H. A l-Q a d e e b, A. A l-L w a i m i. Case Study: Monitoring of AIR Quality in King Faisal University Using a Microcontroller and WSN. – Procedia Computer Science, Vol. **21**, 2013, pp. 517-521.
11. K u l a r a t n a, N., B. H. S u d a n t h a. An Environmental Air Pollution Monitoring System Based on the IEEE 1451 Standard for Low Cost Requirements. – IEEE Sensors Journal, Vol. **8**, April 2008, No 4, pp. 415-422.
12. A l-A l i, A. R., I. Z u a l k e r n a n, F. A l o u l. A Mobile GPRS-Sensors Array for Air Pollution Monitoring. – IEEE Sensors Journal, Vol. **10**, October 2010, No 10, pp. 1666-1671.
13. D e v a r a k o n d a, S., P a r v e e n S e v u s u, H. L i u, R. L i u, L. B a d r i N a t h. Real-Time Air Quality Monitoring Through Mobile Sensing in Metropolitan Areas. – In: Proc. of 2nd ACM SIGKDD International Workshop on Urban Computing, 2013.
14. Tracking Air Pollution in Delhi. 2014. Accessed on 12 December 2019.
   https://blog.socialcops.com/engineering/tracking-air-pollution-in-delhi,

15. S h e r i n, A., L. X i n r o n g. A Cost-Effective Wireless Sensor Network System for Indoor Air Quality Monitoring Applications. – Procedia Computer Science, Vol. **34**, 2014, pp. 165-171.
16. K u m a r, A., G. P. H a n c k e. Energy Efficient Environment Monitoring System Based on the IEEE 802.15.4 Standard for Low Cost Requirements. – IEEE Sensors J., Vol. **14**, 2014, No 8, pp. 2557-2566.
17. F e r d o u s h, S., X. L i. Wireless Sensor Network System Design Using Raspberry Pi and Arduino for Environmental Monitoring Applications. – Procedia Computer Science, Vol. **34**, 2014, pp. 103-110.
18. B a c c o, M., F. D e l m a s t r o, E. F e r r o, A. G o t t a. Environmental Monitoring for Smart Cities. – IEEE Sensors Journal, Vol. **17**, 2017, No 23, pp. 7767-7774.
19. M o r a w s k a, L., P. T h a i, X. L i u, A. A s u m a d u-S a k y i a, G. A y o k o, A. B a r t o n o v a, A. B e d i n i, F. C h a i, B. C h r i s t e n s e n. Applications of Low-Cost Sensing Technologies for Air Quality Monitoring and Exposure Assessment: How Far Have They Gone? – Environ. Int., Vol. **116**, 2018, pp. 286-299.
20. S h a r m a, D. K., M. R a j p u t, S. A. A k b a r, A. K u m a r. Development of Embedded System for Carbon Nano Tube (CNT) Based Ammonia (NH3) Gas Sensor. – In: Annual IEEE India Conference (INDICON), 2015, pp. 1-4.
21. T i w a r i, A., S. S a d i s t a p, S. K. M a h a j a n. Development of Environment Monitoring System Using Internet of Things. – Ambient Communications and Computer Systems. AISC, Vol. **696**, 2018, pp. 403-412.
22. M a r q u e s, G., I. M. P i r e s, N. M i r a n d a, R. P i t a r m a. Air Quality Monitoring Using Assistive Robots for Ambient Assisted Living and Enhanced Living Environments through Internet of Things. – Electronics, Vol. **8**, 2019, Issue 12.
23. H e, H o n g-d i, M i n L i, W e i-l i W a n g, Z h a n-y o n g W a n g, Y u X u e. Prediction of PM2.5 Concentration Based on the Similarity in Air Quality Monitoring Network. – Building and Environment, Vol. **137**, 2018, pp. 11-17.
24. D h i n g r a, S., R. B. M a d d a, A. H. G a n d o m i, R. P a t a n, M. D a n e s h m a n d. Internet of Things Mobile – Air Pollution Monitoring System (IoT-Mobair) – Internet of Things Journal, Vol. **6**, 2019, Issue 3, pp. 5577-5584.
25. H u a n g, J., et al. A Crowdsource-Based Sensing System for Monitoring Fine-Grained Air Quality in Urban Environments. – IEEE Internet of Things Journal, Vol. **6**, 2019, No 2, pp. 3240-3247.
26. S u n, S., X. Z h e n g, J. V i l l a l b a-D í e z, J. O r d i e r e s-M e r e. Indoor Air-Quality Data-Monitoring System: Long-Term Monitoring Benefits. – Sensors, Vol. **19**, 2019, Issue 19.
27. CPCB. 2019. National Air Quality Monitoring Programme. Accessed: 12 December, 2019.
   https://cpcb.nic.in/about-namp/
28. **https://timesofindia.indiatimes.com/city/ahmedabad/vadodara-most-polluted-industrial-cluster-ngt/articleshow/70312184.cms**
29. Espressif, Inc. User Manual. Espressif, Inc., Shanghai, China, 2016. Accessed on 12 December 2019.
   https:// www.espressif.com/sites/default/files/9b-esp8266-low_power_solutions_ en_0.pdf
30. Winsen Electronics, Co. Electro Chemical CO Module User Manual, Winsen Electronics, Co., China, 2015.
   https://www.winsen-sensor.com /sensors /co-sensor /ze07-co.html
31. Nova Fitness, Co. Laser PM Sensor Specification, Nova Fitness, Co., China, 2015.
   http://inovafitness.com/en/a/chanpinzhongxin/ 95.html
32. M i s k o w i c z, M. Send-On-Delta Concept: An Event-Based Data Reporting Strategy. – Sensors, Vol. **6**, 2006, Issue 1, pp. 49-63.
33. N g u y e n, V., Y. S u h. Networked Estimation with an Area-Triggered Transmission Method. – Sensors, Vol. **8**, 2008, No 2, pp. 897-909.
34. S u h, Y. Send-On-Delta Sensor Data Transmission with a Linear Predictor. – Sensors, Vol. **7**, 2007, No 4, pp. 537-547.