

Development of an IoT based Air Quality Monitoring System

Sreevas R, Shanmughasundaram R, VRL Swami Vadali

Abstract: *The proposed work aims for a large-scale air pollutant monitoring for ambient and indoor environments. This system is developed to measure various environmental parameters. Source of pollutants can be identified by analyzing the data collected from the various sensor nodes, so that air quality can be monitored by applying engineering science and data. This is achieved by installing multiple sensor stations in various locations such as hospitals, factories, Offices, streets and weather stations. These sensor stations measure the environmental parameters such as PM2.5, PM10, Sulphate (SO_x), Nitrate (NO_x), Ozone (O₃), Volatile Organic Compounds (VOC), Temperature and Humidity. The sensor stations communicate with cloud over HTTP protocol. Each station has ESP 8266 smart controller which captures the sensor data and creates forms the JavaScript Object Notation (JSON) data packets that mainly consists of sensor data along with node address. These packets will be sent to the cloud over HTTP protocol. The user can access the air quality data from the web application.*

Index Terms: Air quality; HTTP; IoT; Cloud; ESP 8266.

I. INTRODUCTION

Recent media reports suggest that millions of people die in world every year due to air pollution. Most of the people are not aware of air quality in their surroundings. The quality of air cannot be judged so easily and most of the cities have alarming levels of PM2.5, PM10, SO₂ and NO_x and this will put the people at the cities at risks of various respiratory diseases [4][6]. A specific particle in the air can cause a respiratory disease so it will be a great idea to monitor that particle in the air. The major cause of air pollution needs to be identified and better living environment need to be provided. This project aims to monitor various environmental pollutants by using different sensors and make this information available to various stakeholders. All the information can be accessed from a web application that can be accessed by the user from any browser. User can easily monitor the real time data, historical trends and report generation from this web application. The objective of the project is to build an IoT platform to monitor the air quality

Revised Manuscript Received on July 08, 2019.

Sreevas R, Dept., of Electrical and Electronics Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India.

Shanmughasundaram R, Dept., of Electrical and Electronics Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India.

VRL Swami Vadali, Chief Executive Officer, Ayati Labs Pvt. Ltd. Hyderabad, India.

across the nation by installing affordable sensor stations. The data obtained from the stations are integrated with other services so that the people can access the information.

Various ultra-high precision sensors are available which are used for monitoring various particles in the air. These sensors measure the concentration of different particles such as PM2.5, PM10, SO_x, NO_x and VOC [1]. Apart from these particles, temperature and humidity also affects the air pollution. So, temperature and humidity need to be measured with ultra-high precision. Selection and placement of these sensors are important [8]. These sensors should be exposed to ambient environment by considering safe operating conditions. Most of these sensors are pre-calibrated and ready to use but calibration need to be checked before using it. These sensors make use of different communication standard for data transfer. The challenge of the project lies not only in the accurate measurement of air quality but also in presenting this data to different stakeholders in a standard manner. The data can be accessed from any part of the world and this data can be used for various studies and analysis.

The controller collects the data from the various sensors, converts the raw data to useful information. This information sends to the cloud. Various alarms, warnings and alerts are created so that the user can easily act on it and take necessary precautions. Each local node can be configured by the user remotely. Real time monitoring of various environmental parameters is possible and different historical trends are available.

To summarize, this work aims at building an IoT platform to monitor the air quality over a large area by installing different sensor stations at different lo-cations. All the information can be accessed easily by the public. This system can be used for indoor and outdoor services for hospitals, offices and many other industries.

II. OVERVIEW AND COMPONENTS OF AIROK

Fig.1 shows the overview of the proposed air quality monitoring system. This system can be used in the ambient environmental conditions and climate-controlled rooms. This system has three major parts that are sensing, controller and server. Sensing part consist of one or more sensors to measure the various particles in the atmosphere. These sensors are placed such that optimum air flow is present and within the safe operating region. Controller collects the data from the sensor using I²C protocol. Advantage of this protocol is multiple devices can

be connected over the same data line. All the data collected will be sent to the server. Ethernet or Wi-Fi is used for communication between the controller and server [10]. All the data will be stored locally on the controller if there is any failure in communication between controller and server.

Various alerts are generated based on any communication failure and various parameters measured by the sensors. Alerts can be configured through a web application. The web application is developed to display the live data in a graphical format. The application enables the user to obtain the historical trends and reports.

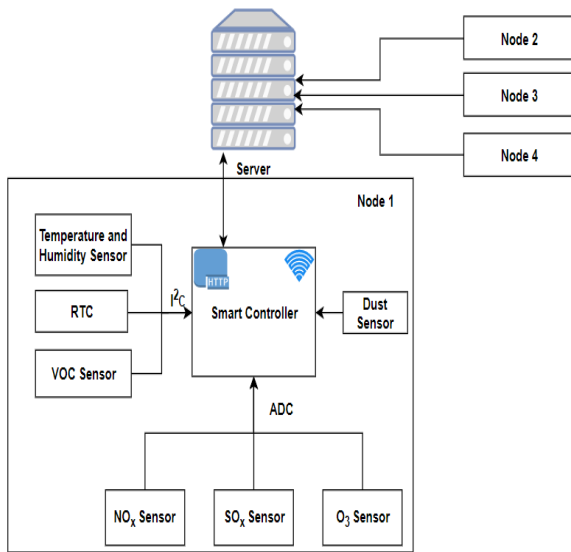


Fig .1: SystemOverview

A. Temperature and humidity sensor

AMS ENS210 sensor is used for the measurement of temperature and humidity. This sensor make use of I²C protocol for sending data to ESP 8266 smart controller. This sensor is mounted on a sensor daughter board. It has ±0.2 precision in the case of temperature and ±3.5% in case of humidity [5]. Based on temperature and humidity, alerts are generated. Fig 2 shows the ENS 210 sensor and Fig 3 shows the sensor daughter board.

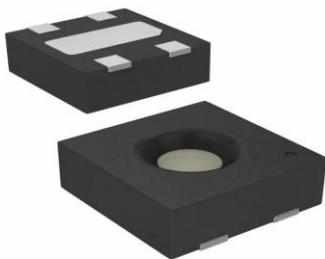


Fig .2: AMSENS210sensor

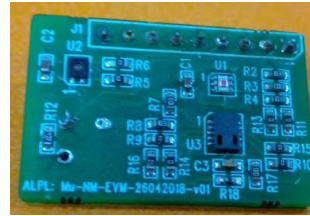


Fig .3: Sensor daughter board

B. VOC Sensor

The AMS CCS811 sensor is used for the measurement of volatile organic compounds. It uses I²C protocol for sending data to the ESP 8266 smart controller. This sensor mounted on sensor daughter board is shown in the Fig. 3. Alerts are generated based on VOC level in the environment. The equivalent CO₂ output range is from 400ppm to 8192ppm. The total VOC output range is from 0ppb to 1187ppb. Fig. 4. Shows the AMS CCS811 sensor.

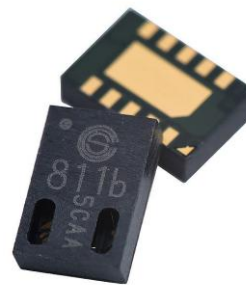


Fig .4: AMSCCS811sensor

C. Dust Sensor

The Amphenol SM-PWM-01C smart dust sensor is used for measuring PM_{2.5} and PM₁₀. This sensor generates the output signal as PWM pulses. There are two output pins in the sensor; one for PM_{2.5} and the other for PM₁₀. This sensor uses IR LED and a photo detector for detecting the dust. Fig. 5 shows the SM-PWM-01C smart dust sensor. This sensor is connected to digital input/output pins of ESP 8266 smart controller.



Fig .5: SM-PWM-01Csmartdustsensor

D. SO_x Sensor

The SPEC 3SP_SO₂_20 C is an analog sensor and is used for measuring SO₂ level in the environment. It can measure from 0ppm to 20ppm. This sensor is connected to ADC pin of ESP8266 smart controller. Alerts are generated based on the concentration of SO₂ in the atmosphere.



Fig .6: SPECSO2Sensor

E. NO_x Sensor

The SPEC NO₂ sensor is used to measure the concentration of NO₂ from 0 ppm to 5 ppm in the atmosphere. This sensor is connected to the ADC channel of ESP8266 smart controller.

O₃ Sensor

The SPEC O₃ sensor is used to measure the ozone concentration in the atmosphere. This is an analog sensor and is connected to the ADC channel of the ESP8266 smart controller. Fig .7 shows the SPEC O₃ sensor.



Fig .7: SPECO3Sensor

F. ESP8266 Smart Controller

The ESP8266 smart controller is a custom microcontroller using ESP8266 chip which has built in Wi-Fi [9]. It collects all the data from the sensors. This data is sent to the server in JSON format. HTTP protocol is used for sending the data to the server. Smart controller receives the data from the server.



Fig .8: ESP8266SmartController

G. Real Time Clock

The DS3231 RTC is used in this system. Ds3231 communicate with ESP8266 smart controller over I²C protocol. It has a very crucial role in this system. Data packet consist of time stamp along with sensor data and MACID.

III. SYSTEM OVERVIEW AND METHODOLOGY

The system consists of multiple sensor nodes and all the nodes communicate with server [7]. HTTP protocol is used for communication between smart controller and the server. Each sensor node consists of a smart controller and multiple sensors interfaced with this smart controller. Fig. 1 shows the system overview. Both VOC sensor, temperature-humidity

sensor and RTC communicate with smart controller over I²C protocol. Both the sensors mounted on a sensor daughter board.

All the analog sensors are multiplexed and connected to ADC channel of smart controller. Smart controller collects the data from the sensors. This data is converted to JSON format and post to cloud over HTTP protocol. In HTTP protocol the http client sends the requests to web server and the server sends the response to the client. In this system smart controller acts as an http client and it send requests for logging live data. Smart controller uses this HTTP protocol for downloading the node configuration too. Based upon the response for each http request smart controller decide whether the communication with server is successful or not. Live data logging happens in every minute and this live data consists of sensor values, MAC address, average values, node configuration details. Fig .9 shows the sequence of events that happens during the live data logging to the server.

Every minute smart controller samples the sensor data and convert this data to JSON format and post this data to cloud. During live data logging, any failure in communication between smart controller and server can happen. This failure is identified by the smart controller based on response received by it for each request. So, the smart controller stores all the data locally in case of any communication failure with the server. The locally stored data will be sent again when there is a successful communication between smart controller and server. After successful posting to cloud, the smart controller discards the stored data. The user can analyze and generate reports based on the old data from the web application. This can be achieved by storing the data locally when the communication with server fails. The locally stored data will be available at web application when there is a proper communication with server. So, user will not miss any data during the communication failure.

A Sample data can be shown as {"lup": 260, "ts": 1553777629861, "alert": true, "averages": {"rh": "47.52", "tm": "29.41"}, "mac": "b8:27:eb:4d:ad:c6", "rh": "47.52", "tm": "29.41"}. This is a sample data packet sent by the smart controller to the cloud. It consists of live sensor values (tm and rh), MAC ID of the smart controller(mac), current node configuration or last updated configuration(lup), average value of different parameters based on the last five samples (averages) and time stamp in milli seconds (ts) with the help of RTC DS3231. Response from the server for the above data can be shown as {'status': True, 'code':100, 'getconfig': False}. The 'status' specifies if the data logging is successful or not. The 'getconfig' indicates that node configuration at server and node side is matching or not. The last updated configuration (lup) value decides whether the node

requires to download the node configuration from the server. If the 'lup' value sent by the node is not matching with 'lup' value at the server then the smart controller gets a response message as {'status': True, 'code':100, 'getconfig': True}. Then node will try to download the node configuration from the server by sending a request which contains it's MAC ID. Once the server receives the request for node configuration it will send response message as {'status': True, 'code': 100, 'data': {'node_rate': 0, 'node_lup': 260, 'node_opr': [{'type': 0, 'max': 40, 'min': 17}], {'type': 1, 'max': 60, 'min': 10}}, 'node_id': 66, 'account_id': 10, 'node_mac': 'b8:27:eb:4d:ad:c6'}}. This message consists of latest node configuration and MAC ID of the node.

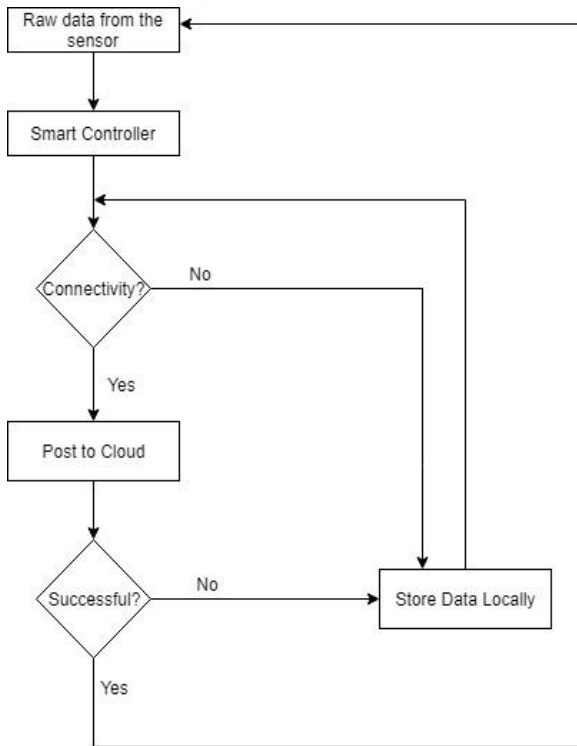


Fig .9: Flowchart of data logging

The smart controller generates the alerts based on the average value of last five samples collected from the sensors. Smart controller checks whether the average value lies within the safe region. An alert is generated if the average value lies outside the safe region. Safe region is defined from the server. Fig.10 and Fig. 11 shows the trends available in the web application.

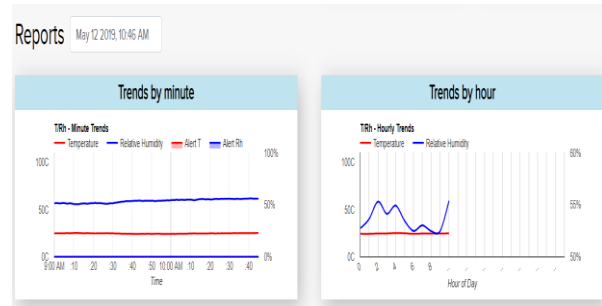


Fig .10: Trends of humidity and temperature without any alert

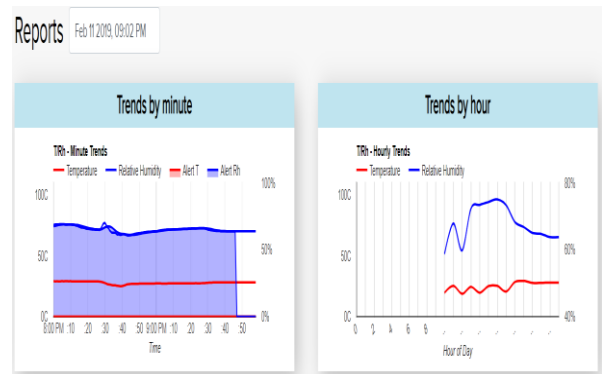


Fig .11: Trends with alert in humidity and no alert in temperature

The web application shows the real time trends and historical trends of all the parameters such as temperature, humidity, volatile organic compounds (VOC), sulphate (SO_x), nitrate (NO_x), PM2.5, PM10 and ozone (O₃) [3]. Fig.10 shows the trend of temperature and humidity during a selected time. No shaded region is present in this trend. Shaded region indicates the presence of alert. Blue line indicates the humidity value and red line indicates the temperature value. When an alert is present for a parameter, then a shade of that parameter line colour will appear in the trend. Alert scenario is very clear in the Fig 11, that is a blue shade is present in the trend for some time period of 1 hour 45 minutes (from 8pm to 9:45pm) out of 2-hour data (from 8pm to 10pm). From this user can understand that the humidity is outside the specified region and causing alert.

All the users can identify the presence alert by three different ways. First way is by monitoring the web application. From the trends shown in the web application, user can identify the alert as shown in Fig.11. Second method is by seeing the LED status on the node. In third and final method each user receives E-mail and SMS when the alert is present.

Implementation

The whole system consists of multiple sensor stations and all these sensor stations are connected to the server. Each sensor station has ESP8266 smart controller and various sensors sends the data to smart controller. Interfacing of digital sensors with smart controller is achieved by I²C protocol. Analog sensors are connected to ADC channel of smart controller. The smart controller converts the raw data from the sensors to useful data. JSON data packets are formed by the smart controller and send this

data packets to server over HTTP protocol. Fig.12 shows the four sensor stations kept inside a temperature calibration chamber. Chamber testings are done at a NABL accredited laboratory. This sensor stations have slits on the outer surface of the enclosure for natural ventilation.



Fig .12:Sensor stations inside a temperature calibration bath

In this system smart controller and sensors kept in an enclosure which has two chambers such that all the sensors are kept in a separate chamber with proper ventilation so that the sensors are exposed to ambient conditions. The smart controller kept in another chamber with necessary heat sinks. Due to this type of arrangement, any effect of heat generated by the controller on the sensors will be minimized.

Table I. Calibration data of temperature and humidity

Temperature			Humidity		
Set Point (°C)	Ref (°C)	Node (°C)	Set point (%rh)	Ref (%rh)	Node (%rh)
10	10.17	12.51	30.9	29.3	31.52
15	15.67	15.52	45	45.1	46.88
21	21.92	21.55	50	50.0	50.26
22	23.02	21.48	52	52.0	52.36
23	24.22	25.8	54	54.1	54.28
24	25.21	24.8	56	56.1	56.42
25	26.4	25.87	58	58.2	58.91
30	31.78	30.98	60	59.7	60.5
35	37	36	65	65.1	64.

	.07	.05		4	74
40	42.28	41.16	75	76.2	74.85

Calibration of temperature and humidity is achieved by keeping the enclosed sensor node into a calibration bath which has a reference thermometer and hygrometer. The nodes are calibrated at nine different points. Table.1 shows the calibration data of temperature and humidity of one node. During calibration, each point is set for one hour. Comparison of temperature and humidity of sensor station (Node) with the reference device (Ref) is carried out.

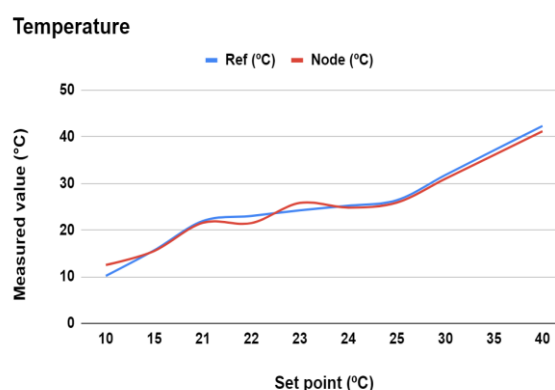


Fig .13: Temperature calibration chart

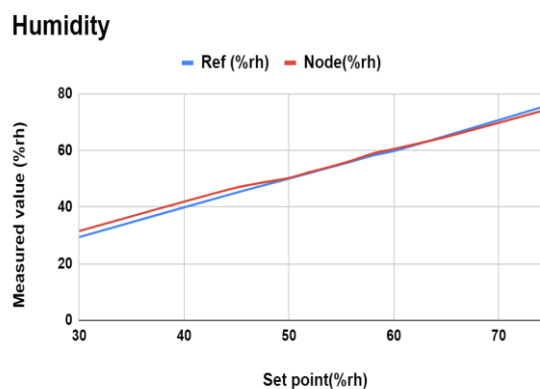


Fig .14: Humidity calibration chart

IV. CONCLUSION

Using this air quality monitoring system, continuous monitoring of different environmental parameters such as PM2.5, PM10, sulphate (SOx), nitrate (NOx), ozone (O3), temperature and humidity is achieved, and user can monitor these parameters through a web application. This is achieved by using different sensor stations and each sensor station has an ESP8266 smart controller. The smart controller collects the sensor data over I2C protocol or its ADC channel. HTTP protocol is used for sending the data from controller to the cloud.

Real time trends, historical trends, alerts and reports are



accessed by a user by using the web application. In future number of nodes can be increased, so user can monitor the air quality over a large area [2] and this system can be integrated with google weather or similar services so that the public can easily access the air quality data.

REFERENCES

1. S. Kumar and A. Jasuja, "Air quality monitoring system based on IoT using Raspberry Pi," 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, 2017, pp. 1341-1346.
2. SM, Shiva Nagendra, et al. "Mobile monitoring of air pollution using low cost sensors to visualize spatio-temporal variation of pollutants at urban hotspots." *Sustainable Cities and Society* 44 (2019): 520-535.
3. C. Xiaojun, L. Xianpeng and X. Peng, "IOT-based air pollution monitoring and forecasting system," 2015 International Conference on Computer and Computational Sciences (ICCCS), Noida, 2015, pp. 257-260.
4. <https://thewire.in/environment/air-pollution-monitoring-diwali-pm2-5-pm10-cpcb-namp>.
5. https://www.mouser.com/ds/2/588/ams_11032016_ENS210-1214724.pdf
6. Cheng, Yun, et al. "AirCloud: a cloud-based air-quality monitoring system for everyone." *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*. ACM, 2014, pp.251-265.
7. Jen-Hao Liu et al., "Developed urban air quality monitoring system based on wireless sensor networks," 2011 Fifth International Conference on Sensing Technology, Palmerston North, 2011, pp. 549-554..
8. Abraham, Sherin, and Xinrong Li. "A cost-effective wireless sensor network system for indoor air quality monitoring applications." *Procedia Computer Science* 34 (2014): 165-171.
9. G. Spandana and R. Shanmughasundram, "Design and Development of Air Pollution Monitoring System for Smart Cities," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 1640-1643.
10. M. V. Ramesh et al., "Water quality monitoring and waste management using IoT," 2017 IEEE Global Humanitarian Technology Conference (GHTC), San Jose, CA, 2017, pp. 1-7.

AUTHORS PROFILE

Sreevas R., Dept., of Electrical and Electronics Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India.

Shanmughasundaram R., Dept., of Electrical and Electronics Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India.

VRL Swami Vadali, Chief Executive Officer, Ayati Labs Pvt. Ltd. Hyderabad, India.