IoT Enabled Software Platform for Air Quality Measurements

Nikola Mitrović

Department of microelectronics
Faculty of electronic engineering
Niš, Serbia
nikola.i.mitrovic@elfak.ni.ac.rs

Milan Đorđević

College of Applied Technical Sciences
Niš, Serbia
milan.nebojsa.djordjevic@gmail.com

Sandra Veljković

Department of microelectronics
Faculty of electronic engineering
Niš, Serbia
sandra.veljkovic@elfak.rs

Danijel Danković

Department of microelectronics
Faculty of electronic engineering
Niš, Serbia
danijel.dankovic@elfak.ni.ac.rs

Abstract—This paper presents an example of the IoT framework that is suitable for implementation into Smart City and environment system for air quality estimations. Framework consists of the web server application designed for IoT system in client-server mode, where hardware units present separate clients. Application is designed mainly using node.js tools and is accessible through web browsers. Framework is tested with appropriate ESP32 based measuring system being the client in the configuration. Measured data can be transferred, processed and shown to the users in real-time using graphs and images. Also, data is logged and stored for broader air quality assessments. Commercial capabilities and business approach is discussed.

Keywords - IoT, node.js, web application, air quality, ESP32

I. INTRODUCTION

Determining air quality remains a challenging task [1]. Air is perceived as a mix of different particles, and the structure of the air cannot be easily detected by humans. Presence of unwanted particles in the air is marked as air pollution. Likewise, air pollution often cannot be detected easily, until it is high enough to present a problem. Therefore, air pollution is often precisely measured with appropriate modules and devices. These modules are responsive to specific gases, where interaction between the module and the specific gases induces chemical reaction. Chemical reaction create changes in particular electrical parameter of the module (resistance or output voltage) that can be measured with microcontroller or similar controlling unit. Based on the magnitude of the change of the output voltage, concentration of unwanted gases in the air can be estimated. Upon these several eestimations, basic assessment of the air quality can be given [1, 2].

However, explained workflow has several flaws. Main flaw is that the measuring modules have limited reach, and area of air that modules can cover is fairly limited. Common misinterpretation of the results lies in the fact that single sensing module can give estimation for entire town or municipality. Modules can respond only if the targeted gases come in contact with the surface of the module. Natural air flow will, of course, bring more air to the module, but even then, the area where modules can detect presence of gases is limited [2, 3].

Therefore, in order to give more precise assesment of the air quality for a town, it is needed to increase the number of the measuring modules, and to position them in a various areas. In this manner, monitored area is far bigger, since more of the modules will contribute to the results with its measuring. With the increased number, it is needed to propose a technology that is able to monitor and to store the measured data as well as to present it to the multiple users in the same time. Internet of Things (IoT) technologies enable data aquistion from multiple points and simultaneous access to the measured data to multiple users [4, 5].

There have been many reports on IoT system that tackle air quality monitoring [4-10]. Khera proposed a system that can follow CO and CO_2 gases concentration [6]. This system also takes advantage of the LabVIEW, which is additional tool needed for the system. Kodali reported system where users can follow CO_2 concentration using MQTT protocol [7]. System is based on couple of common libraries and provides rather low-cost solution for some basic measurements. Munsadwala designed system that can also access the geographical location of the measuring unit [8]. System uses MQ gas sensor series that give elemental readings. This paper also tackles the ambiental conditions, mostly in securing optimal conditions for reliable measuring. Dhingra developed a system which is supplied with mobile application, where placing measuring units on vehicles is also discussed [4]. These approaches however suffer from the inconstant air pressure delivered to the measuring units, that can hinder the measurements. Đorđević presented a system where various data logging methods are discussed [9]. Data is sent using GSM communication, and can be easily stored to the external memory units.

Software framework presented in this paper presents sublimation of the reported solutions, developed in a different manner, with the goal to enable simple further implementation and addition into up-to-date technologies, mainly targeted at blockchain technologies. Framework is organized into client-server configuration where each measuring point, as well as each data observer is a separate client. Data recieved from separate clients can be encrypted and mutually dependent. Application for retrival and presentation of data is stored in a server. This approach present viable and confirmed solution for similar problems. In this paper, implementation of software framework for IoT system that is able to administer on multiple measuring points centered around measuring several gases concentration is presented. Framework does not rely on the existing commercial platforms, is independent of owned resources and is designed to be a groundwork for the implementation of the more advanced technologies that can

II. METHODOLOGY

IoT system generally consists of multiple measuring units on one side, and data processing and presentation control unit on the other side. Measuring units are using some of the wireless data transfer methods to communicate to the control unit, where data is processed and presented to the users of the system. In recent years significant development of different data transfer methods, as well as portable measuring devices that can measure different magnitudes is noted [4, 5, 10].

Measuring units can be connected to tens and tens of different measuring devices and can communicate with several wireless data transfer methods, while controlling web server applications should be able to ensure real time data presentation, high level of integration and simple data access. Measuring units and control unit can be integrated, beside others, into client-server configuration. General block concept of the proposed framework is given in the Fig. 1

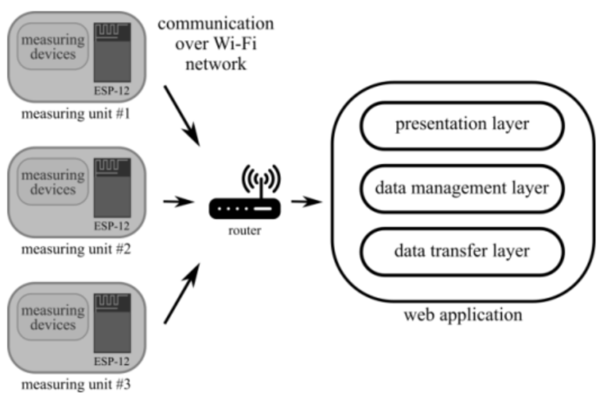

Fig. 1. Block schematioc of the designed web application.

Web server application that performs as a controlling unit for this IoT system demands three layers, data transfer layer, data management layer and presentation layer. It is based on node.js tool [11], designed to perform as a part of the client-server configuration in IoT system implementations, where clients are measuring units, as well as browsers following and accessing to the measured results. Application was developed locally at first, but also tested in real-time, with the usage of Heroku platform [12]. Parts of the developed framework are explained in more detail.

A. Data transfer layer

Data transfer layer deals with the transfer of the data to and from measuring units. Operation of this layer is most limited, since the same data transfer protocol should be used on both sending and receiving end. Designed web platform is set to communicate with the measuring units using WebSocket protocol. This protocol is chosen because of the widespread availability, low power demand and good usage support for various hardware architectures [13]. The WebSocket protocol enables full-duplex communication between a client running the code to a remote host that has opted-in to communications from that code, in a controlled environment, over single transfer control protocol connection. Developed application also supports data transfer through HTTP protocol, although WebSocket protocol is used as a primary transfer method.

B. Data management layer

Data management layer covers the processing of data. This layer of web application enables that needed calculations based on the data from the measuring units can be conducted in the controlling unit, that is often supplied with the stable voltage supply, rather than at the measuring unit, that is most often supplied through battery or similar limited supply [14].

In the designed software platform, data management layer is also used as the correction factor. If the received data is out of the anticipated range, platform can react with the appropriate notification to signal the user of the specific state. This layer of application is also very important for the future goal of the application, which is to implement blockchain technology to the framework [15]. This layer should cover process of addition of the processed data of the peer-to-peer organized blockchain nodes to the ledger.

C. Data presentation layer

Data presentation layer presents the measuring results to the user. Among many possibilities for data presentation, some of the appropriate for this type of usage are real-time graph that shows some of the recent measuring and a table that shows current measuring from the specific measuring unit. Interface of the designed software platform for air quality monitoring that follows the given concept is given in the Fig. 2.

Fig. 2. Interface of the designed web application.

Data presentation layer focuses around wide usage of the Chart.js module of the node.js based application that creates the appropriate graph in the center of the interface [16]. Data is expected to be presented on the graph as soon as received and processed, meaning, in real-time. Among other capabilities, data is shown can also be exported to the CSV file, where can be subsequently analyzed.

D. Measuring units

Important parts of the proposed framework are also measuring units. These units consist of measuring devices (sensors or similar sensing parts) and a wireless data transfer method enabled device, usually some type of a microcontroller or similar. Since the data transfer method between the controlling unit and the measuring unit is WebSocket method, measuring units incorporated microcontroller need to have the ability to use it.

Main measuring device that is the part of the measuring unit used for the development of the framework is MiCS5524 [17]. It is a micro-electromechanical system (MEMS) for carbon monoxide and natural gas leakage detection, that is able to measure concentration of CO in ppm. It comes with a simple interface for connection with other devices. Measuring of this MEMS are, according to the datasheet, as well as other reports [17, 18], heavily influenced by meteorological conditions. Therefore, output of this MEMS must be analyzed in the data management layer together with the value of the air temperature, air pressure and air humidity. These three magnitudes are measured using another measuring device, concretely BME280 [19], that is able to deliver it through I2C protocol.

Other part of the measuring unit is a Wi-Fi enabled microcontroller ESP12-E [20]. This module is the improved version of the commercially successful ESP8266 chip. It is characterized with even lower power consumption, especially in the sleep mode, and with increase in GPIO pins, meaning that is able to communicate with even more measuring devices. Implementation of WebSocket protocol on this microcontroller is done using Espruino [21, 22], programming and flashing JavaScript based tool.

Measuring unit is placed on a printed circuit board, so that it can be space effective and battery supplied. Designed PCB is presented in the Fig. 3.

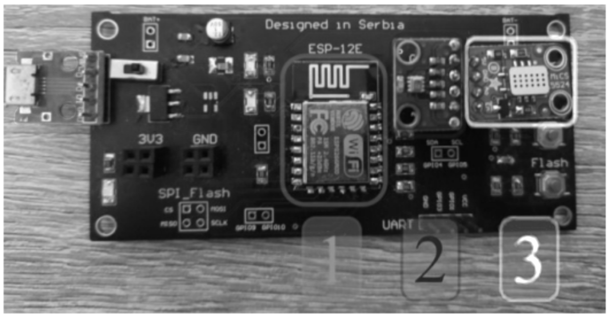

Fig. 3. Designed printed circuit board: 1) ESP12-E; 2) BME208; 3) MICS-5524.

III. FRAMEWORK TESTING

Testing of the framework, both application and measuring units is conducted with multiple ESP12-E based measuring units that are sending measured data using WebSocket protocol to the web server application. Tests were performed in the following manner.

Web application is activating one of the measuring units using WebSocket protocol. ESP12-E wakes from the sleep mode, and performs measurement from MiCS-5524 and BME280, and sends the measured data using WebSocket protocol back to the data transfer layer of the web server application. Upon receiving data, the application sends a message to the measuring unit that puts it into sleep mode until next wake-up signal. After sending the sleep signal, data transfer layer shares data with the data management layer. In this layer, measured values are processed according to the pre-calibrated values for the measuring devices. If the data is somehow corrupted, or there is an error in the reading, data transfer layer is reactivated with the goal to obtain the measurements again. If the data is in the range of expected, then, it is transferred to the data presentation layer and showed in the graph. During testing, measuring was performed in every hour, during one day, from three different measuring units, placed in different areas. Fig. 4 shows the results measured with professional equipment from the measuring devices itself. On the other hand, Fig. 5 and Fig. 6 present the data received through the software platform.

Measured results regarding CO concentration are in line with other investigations [1, 4, 6, 18]. As can be seen from the Fig. 4 and Fig. 5, developed framework shows pretty similar readings as the measurements made at the directly on the measuring units. However, another capability, presented in the Fig. 6 is noticed. For the readings from the measuring unit number 2, there is no line representing the ppm calculated values for the period after 10:00 AM. As can be seen from the Fig. 4, measurements from 07:00 AM to 10:00 AM show readings of 0 ppm of CO, where the real value is probably somewhere between 0 and 1 ppm. After receiving three successive zeros in this period of time web server application notifies the user of the measuring unit #2 using WebSocket method that the error is assumed and that operation of the unit should be checked. If the user checks that the device is operating reliably, reception of the data is continued, regardless of the values. If the user checks that device is not operating reliably, false measurement is prevented and the obtained results are, in overall, more reliable.

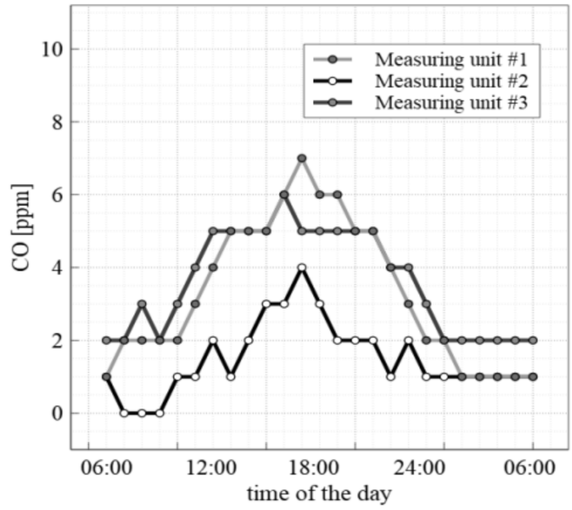

Fig. 4. Data measured directly from the measuring devices, using professional equipment.

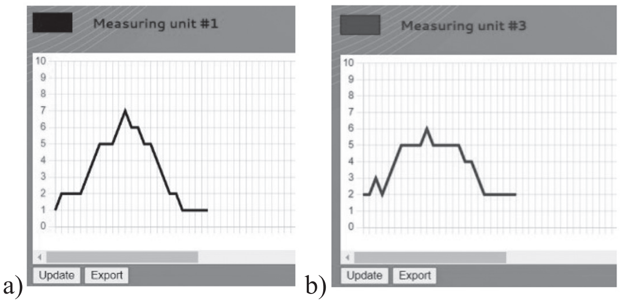

Fig. 5. Data from measuring units presented through the graph of the developed framework: a) measuring unit #1: b) measuring unit #3.

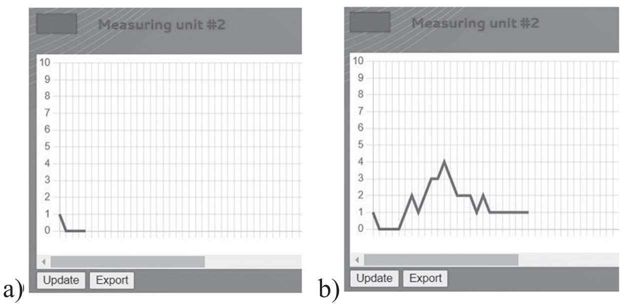

Fig. 6. Data from measuring unit #2 presented through the graph of the developed framework: a) receipt of succesive zeros; b) entire dataset.

IV. IMPLEMENTATION CAPABILITES

Presented framework offers a lot of commercial capabilites and potential for broad practical implementation. Software platform, together with its hardware counter part, provide good scalability and simple integration into installed systems. Number of measuring devices can easily be expanded with the goal to increase the number of measured magnitudes, or to increase the number of measurements related to the concentration of specific gases in the limited area. Hardware part of the system can be integrated with appropriate GPS modules, that will enable the users to have very precise knowing on the air quliaty in specific areas.

Beside broad capabilites that can be made use of from the hardware standpoint, amount of data collected from the devices allow basis for implementation of blockchain technologies in the management and processing of great amount of data. Blockchain's encryption technology and messaging protocol maintains data integrity, while enabling data sharing, as well as improve data processing efficiency. Developed framework can be also adapted to be compatible to Big data systems, if the number of hardware measuring units, as well as data obtained from them increases.

V. CONCLUSION

The paper present development and testing of the web application meant to be used as the software platform in IoT air quality system. Application is designed, realized and practically implemented. Presented framework offers a lot of commercial capabilities and potential for broad practical implementation. Software platform, together with its hardware counterpart, provide good scalability and simple integration into installed systems.

ACKNOWLEDGEMENT

This work was supported by the Ministry of Science of the Republic of Serbia under the Grant No. 451-03-9/2021-14/200102.

REFERENCES

[1] Environmental Protection Agency of the Ministry of Environmental Protection of Serbia, "The Annual Report on the Air Quality in the Republic of Serbia", 2019, ISSN 2334-8763.

[2] J. S. Munir et al., "Structuring an integrated air quality monitoring network in large urban areas – Discussing the purpose, criteria and deployment strategy", Atmospheric Environment, p. 100027, 2019.

[3] T. Popović, "Report on air quality protection at the local self-government level in the Republic of Serbia", National Association of Local Authorities in Serbia, 2020.

[4] S. Dhingra et al., "Internet of Things Mobile - Air Pollution Monitoring System (IoT-Mobair)", IEEE Internet of Things Journal, vol. 6, no. 3, pp. 5577-5584, 2019.

[5] N. Setiwan and I. Kustawian, "IoT based Air Quality Monitoring", IOP Conf. Series: Materials Science and Engineering, vol. 384, p. 012008, 2017.

[6] N. Khera, P. Sharma, D. Shukala and I. G. Dar, "Development of Web-Based Gas Monitoring System Using LabVIEW", in Proc. of Recent Developments in Control, Automation & Power Engi-

neering (RDCAPE), 2017, pp. 439-441.

[7] R. K. Kodali and B. S. Sarjerao, "MQTT based Air Quality Monitoring", in Proc. of IEEE Region 10 Humanitarian Technology Conference (R10-HTC), 2017, pp. 742-745.

[8] Y. Munsadwala, P. Joshi, P. Patel and K. Rana, "Identification and Visualization of Hazardous Gases Using IoT", in Proc. of 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), 2019, pp. 1-6.

[9] M. Đorđević, B. Jovičić, S. Marković, V. Paunović and D. Danković, "A smart data logger system based on sensor and Internet of Things technology as part of the smart faculty", Journal of Ambient Intelligence and Smart Environments, vol. 12, no. 1, pp. 1-15, 2020.

[10] D. Danković and M. Đorđević, "A Review of Real Time Smart Systems Developed at University of Niš", Facta Universitatis, Series: Electronics and Energetics, vol. 33, no. 4, pp. 669-686, 2020.

[11] Software tool: node.js. Available at: https://nodejs.org/en

[12] Heroku platform. Available at: https://www.heroku.com/

[13] I. Fette and A. Melnikov, The WebSocket Protocol, December 2011.

[14] H. Turkmanović, I. Popović, D. Drajić and Z. Čiča, "Green computing for IoT – software approach", Facta Universitatis, Series: Electronics and Energetics, vol. 35, no. 4, pp. 541-555, 2022.

[15] S. Benedict, P. Rumaize and J. Kaur, "IoT blockchain solution for air quality monitoring in smart cities", in Proc. of the IEEE Int. Conf. Adv. Netw. and Telecommunications Syst. (ANTS), 2019, pp. 1-6,

[16] Module Chart.js. Available at: https://www.chartjs.org/

[17] MiCS-5524 datasheet, Sensortech, 2018.

[18] J. A. Fatkhurrahman and I. R. J. Sari, "Affordable Metal Oxide Gas Sensor as Environmental Friendly Method for Ammonia Gas Measurement", IOP Conf. Ser.: Earth Environ. Sci. vol. 366, p. 012027, 2019.

[19] BME280 datasheet, Bosch, 2022.

[20] ESP12-E datasheet, Esspresif. 2015.

[21] Espruino tool. Available at: https://www.espruino.com/

[22] N. Mitrović et al. "Implementation of WebSockets in ESP32 based IoT systems", in Proc. of the 15th Int. Conf. Adv. Technol. Syst. Services in Telecommunications (TELSIKS), 2021, pp. 261-264.