ISSN 2224-087X. Електроніка та інформаційні технології. 2024. Випуск 28. С. 84–94
Electronics and information technologies. 2024. Issue 28. P. 84–94

UDC 681.518 DOI: https://doi.org/10.30970/eli.28.8

DESIGN AND IMPLEMENTATION OF AN INDOOR AIR QUALITY MONITORING SYSTEM BASED ON RASPBERRY PI AND ARDUINO PLATFORMS

I. Rudavskyi, L. Kovaliv, H. Klym

Lviv Polytechnic National University,
12, Bandera Str, Lviv, 79013, Ukraine.

ivan2001rud@gmail.com , *lesia.kovaliv.mkisk.2023@lpnu.ua* , *halyna.i.klym@lpnu.ua*

This paper presents the development of a comprehensive system of monitoring of air quality in closed premises, which is built on the basis of the microcomputer Raspberry Pi 4 and the Arduino hardware. The developed system allows not only to monitor the parameters of air, but also acts to influence its quality. This is achieved through the use of auxiliary devices that ensure efficient air purification, as well as ventilation of the room to maintain comfort. The system contains key hardware components: air cleaner, ventilation unit and sound alarm. Their work ensures the maintenance of air quality at the optimum level and promptly inform the user about any dangerous changes in the composition of the air. Control is carried out by collecting data from sensors that measure air quality parameters every five minutes with the possibility of adjusting the renewal frequency. The information collected is transmitted to the Raspberry Pi, where the data is analyzed and displayed through a specially designed interface, which allows the user to quickly monitor the current state of the room. The sound alarm reports critical changes in the quality of air, such as increasing CO_2 or the content of harmful substances. The system software is designed using Python and Tkinter, as well as Arduino IDE. This combination allows you to effectively organize the work of all components and ensure continuous monitoring of air quality. The wireless connection between the sensors and the central part of the system is implemented by Wi-Fi modules, which facilitates the installation and use of the system in any room. Experimental studies were conducted in two stages. In the first stage, the system worked in the conditions of ordinary office activity, where the basic air indicators during the workflow were recorded. In the second stage, which included physical activity, an increase in CO_2 levels was recorded, which indicates increased air pollution in the conditions of active activity. This emphasizes the importance of regular air monitoring to maintain comfortable conditions in the room, especially during exercise. The results of the experiments confirm the efficiency of the monitoring and quality control system. The system is able to respond promptly to changes in the surrounding sector, providing timely air purification and maintaining it at a safe level, which is important for the health and comfortable stay of people indoors.

Keywords: Monitoring system, Raspberry Pi, Arduino, Algorithm, Air quality

Introduction

Increasing levels of air pollution and its adverse effects on human health and the environment have become critical concerns in the modern world. As air quality continues to deteriorate, it has become a pressing global issue, making air quality monitoring and control essential to manage and mitigate pollution. A promising approach to advance air quality monitoring systems is the integration of miniature computing platforms such as Raspberry Pi

---

and Arduino microcontrollers [1-3]. These technologies enable the design of compact, efficient and cost-effective monitoring systems that incorporate advanced sensors [4,5] and can be deployed in different urban or regional locations. Given the rapid evolution of this technological field, the analysis of current research and methodologies is fundamental for the development of innovative air quality monitoring solutions and the improvement of existing systems.

Numerous studies and projects have explored the use of Raspberry Pi and Arduino platforms for the development of air quality monitoring systems. For example, in [6], the authors investigate the potential of Raspberry Pi for real-time data collection and transmission of air pollution levels. In [7,8], the focus is on using Arduino to create a mobile air quality monitoring device that can be mounted on a vehicle to measure various air parameters while in motion.

Despite significant technological advances and progress in air quality monitoring systems using Raspberry Pi and Arduino, several challenges remain unresolved. These include the need for more efficient methods of collecting and transmitting data from monitoring systems, which continue to be areas of active research [8].

This scientific paper focuses on the development of an air quality monitoring system based on Raspberry Pi and Arduino, using data fusion techniques for a multi-sensor configuration.

Hardware and software implementation

The system is implemented using a Raspberry Pi 4 B single-board computer and an Arduino Uno microcontroller, both of which are well suited for simple, repetitive tasks such as temperature measurement and sensor data acquisition. In this setup, the Arduino microcontroller collects data from connected sensors and transmits it to the Raspberry Pi 4 B for further processing.

The system uses several sensors to measure different air quality parameters. Carbon monoxide (CO) concentration is measured using the MQ-7 sensor, while an NDIR sensor (MH-Z14A) detects carbon dioxide (CO_2) levels in the range of 0 to 5000 ppm. In addition, an optical sensor (GP2Y1010AU0F) is used to monitor particulate matter and dust concentrations in the air.

The ESP8266 ESP-01 module is used for wireless communication between the Raspberry Pi 4 and the Arduino Uno. This module enables the transfer of indoor air quality data without the need for wired connections.

The system also includes an infrared module for load control, enabling wireless management of devices such as ventilation units and air purifiers. This approach overcomes the limitations associated with using a local network for wired control. The infrared receiver module integrates reception, demodulation and amplification, facilitating internal signal decoding. For optimum performance, it is important to use the infrared transmitter and receiver in tandem, otherwise signal sensitivity or link quality may be compromised.

This paper uses the PL2303 USB to TTL converter to convert a USB port into a UART interface. The PL2303 is connected to the ESP8266 module, which enables communication with the Raspberry Pi 4, allowing the Arduino Uno to use the ESP8266 to connect to a local Wi-Fi network. This setup acts as a bridge to transmit various air quality data. In this system, three main loads are used for air quality control: an air purifier, a ventilation unit and an acoustic sounder.

The operating system chosen is the Raspbian Linux distribution, which is optimized for single board computers such as the Raspberry Pi 4. The Arduino IDE (version 1.8.16) was used to program the Arduino Uno. Python was used to process the data from the sensors and to display the results to the user. The graphical user interface (GUI) in Python was implemented using the Tkinter framework.

The overall system design, shown in Figure 1, follows the Internet of Things (IoT) smart home concept and is divided into four key components: (1) Environmental monitoring, (2) Wireless sensor data transmission, (3) Data analysis, (4) Load control.

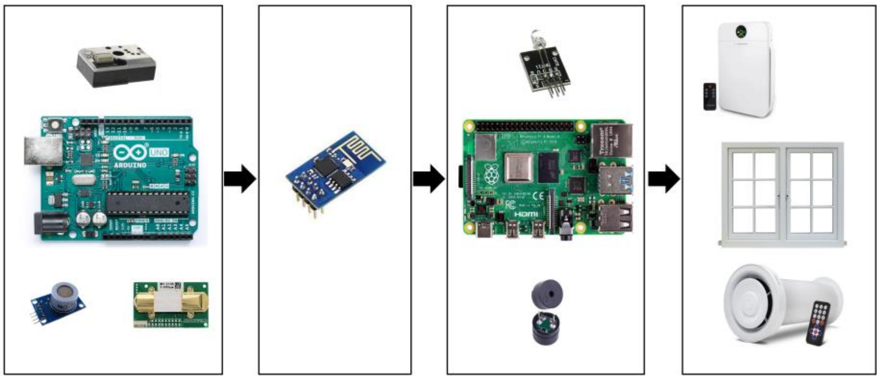

Fig. 1. Overall system design

In part 1, the internal nodes consist of three sensors: one for particulate matter, one for CO_2 and one for CO. These sensors collect data and send it to the Arduino Uno for further processing. An infrared emitter is used to control the ventilation system and air purification, and can be placed centrally in the monitored room to effectively manage air quality.

Each microcontroller is connected to three sensors that assess the air quality in the area where they are placed. For a more comprehensive analysis of indoor air quality, multiple microcontroller-sensor setups should be distributed around the perimeter of the room. In this study, three sets of microcontrollers and sensors are used. Wireless data transmission is facilitated by the ESP8266 Wi-Fi module, which acts as a communication bridge to send data from the Arduino Uno to a computer terminal or web interface.

A fuzzy logic control system is used for data analysis, integrated with Python software to implement an air quality index (AQI). The Python Tkinter framework provides the user interface, allowing users to view air quality status, analyses sensor data and calculate results. The interface also displays overall system performance in an informative manner.

Air quality correction is performed based on sensor data transmitted over the wireless network, followed by fuzzy data analysis. Depending on the air quality parameters, corrective actions are triggered: for example, activating the air purifier, switching on the ventilation system or generating an alarm to open a window. These adjustments are designed to optimize indoor air quality. The loads controlled for these adjustments include the air purifier, the window and the ventilation unit.

If the CO level rises slightly, an audible alarm is given first, alerting the user to open a window as soon as possible to reduce the CO concentration. If the concentration of CO_2 or particulate matter exceeds safe levels, a signal is sent to activate the ventilation unit and air purifier to reduce their levels.

The software architecture for the system consists of two main components: the internal node and the external node. The internal node includes the microcontroller and sensors, with the software responsible for reading the air quality data and transmitting it over a wireless network. Data is collected at regular intervals, with readings sent every 5 minutes using the ESP8266 Wi-Fi module for wireless transmission.

The external node is implemented on a Raspberry Pi 4 single-board computer, which receives data from the internal node. A fuzzy logic algorithm is used to evaluate the air quality based on the sensor data. Depending on the assessment, the external node software sends control signals via the infrared module or triggers the audible alarm. Based on the analysis, the system decides whether to activate the air purifier, ventilation unit or open the window to improve the air quality.

The overall system algorithm is shown in Figure 2, which illustrates the process flow from data collection to corrective action.

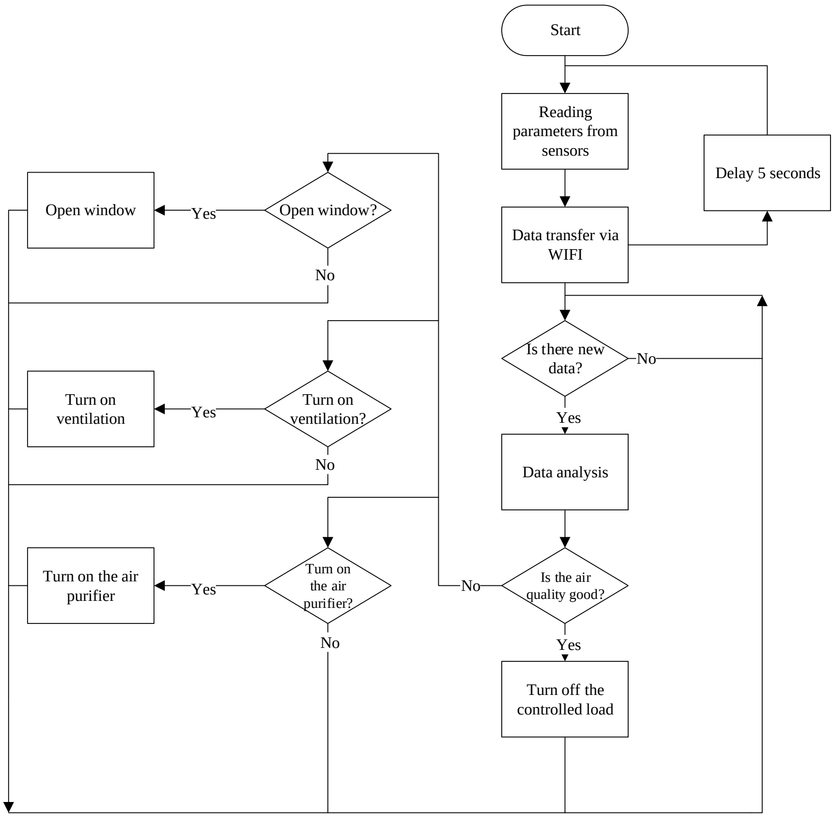

Fig. 2. Overall system algorithm.

When configuring the system, the Raspberry Pi 4 is set to operate in headless mode, which allows it to be controlled remotely over the network without the need for a monitor or keyboard. To achieve this, several steps are taken. First, SSH (Secure Shell) access is enabled and a configuration file called *wpa_supplicant.conf* is created with the necessary Wi-Fi connection parameters to allow the Raspberry Pi to automatically connect to the network.

Once the Wi-Fi router is configured, an infrared emitter is connected to control devices such as the air purifier and ventilation system, and an audible alarm is set up to alert users when the window should be opened for ventilation. In order to control any device equipped with an IR receiver, the infrared LED transmitter must send out a specific sequence of signals. This is achieved using the Linux Infrared Remote Control (LIRC) package, which can simulate the infrared signals of various remote controls. After rebooting the Raspberry Pi, these configurations become active. A profile is selected from the LIRC database [http://lirc.sourceforge.net/remotes/] to emulate a specific remote control.

The connection diagram for the microcontroller components is shown in Figure 3. Several sensors are connected to the Arduino Uno board to monitor air quality: the MQ-7 carbon monoxide (CO) sensor, the MH-Z14A carbon dioxide (CO_2) sensor, the GP2Y1010AU0F dust particle sensor, and the ESP8266-01 Wi-Fi module. The MQ-7 sensor requires four connectors: two for power (+5V and GND), which are connected to the Arduino's 5V and ground terminals, and two for the sensor's analogue and digital outputs. It is through these pins that power is supplied and the sensor functions.

In addition, the GP2Y1010AU0F dust sensor requires a pulse drive for its LED, which is provided by a 150Ω resistor (R1) and a 220μF capacitor (C1) in the circuit.

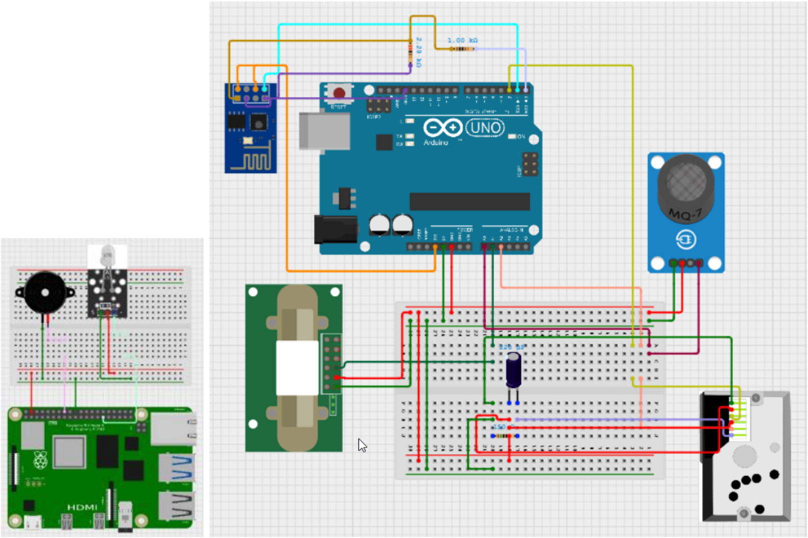

Fig. 3. Connection diagram for the microcontroller components.

Fusion of information for multisensory system

Given the multisensory design of the system, which incorporates numerous sensors, it is essential to consolidate the data collected from these devices. Information fusion refers to the integration of data from multiple sensors from different information sources. This system is equipped to automatically analyses and synthesize data, processing them according to specific criteria in order to improve the understanding of the phenomena being monitored.

The implementation of information fusion technologies in monitoring or control systems, which rely on a variety of sensor types at different operational levels and on numerous parameters, offers several advantages. For example, the fusion of data from multiple sensors provides a more holistic, accurate and interpretable representation of the system than information derived from a single sensor.

Data from a collection of similar or identical sensors may have redundancies, but their effective integration can reduce information uncertainty and increase the overall value of the data. Furthermore, the complementary nature of data collected from different types of sensors is significant. With appropriate processing, this complementarity can mitigate and reduce the uncertainties associated with individual sensors or limited measurement ranges. The use of multiple sensors increases the reliability of monitoring systems; for example, in the event of a sensor failure, the overall system can continue to operate effectively.

In the context of indoor air quality monitoring, the system utilizes an extensive array of sensors to ensure widespread data collection and sufficient information availability. In order to reduce the load on the transmission paths and to minimize the computational demands on the central processing unit, the system has been segmented into several smaller subsystems for localized analysis. The results of these analyses are then integrated to produce a comprehensive result that reflects the fusion of all the subsystems. The algorithm uses a decentralized two-stage merging approach tailored for local integration, with interfaces to the entire operational domain, as shown in Figure 5.

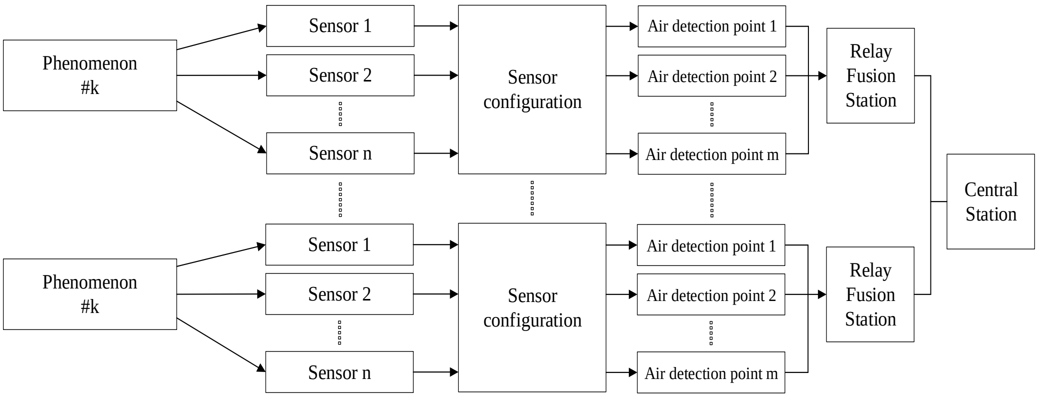

Fig. 5. The structure of the fusion system.

This data fusion approach assigns sensors to different air quality monitoring sites according to the specified design criteria. The algorithm retrieves data after each air quality monitoring station has completed its data collection. A fusion relay station performs a preliminary analysis, while the comprehensive detection station performs a global synthesis and generates additional solutions.

Since the number of air quality monitoring stations is typically limited, local fusion is performed using the classical Kalman vector filter algorithm. Let's denote the number of air quality monitoring points as qq. The signals from these monitoring points can be represented as a qq-dimensional vector $X(k) = [x_1(k)x_2(k)\cdots x_q(k)]^T$. In addition, we consider the presence of process noise, characterized by a sequence of independent "white" noise sources, represented as $\omega(k)=[\omega_1(k)\omega_2(k)\cdots\omega_q(k)]^T$. This allows us to formulate a mathematical model for a multidimensional random signal, as shown in equation (1):

$$X(k) = AX(k-1) + \omega(k-1), \tag{1}$$

where $A = \mathrm{diag}(a_1 a_2 \cdots a_q)$ is a matrix of coefficients.
To optimally filter a qq-dimensional random signal $X(k)X(k)$, the initial rr components of this vector $X(k)X(k)$ are measured simultaneously at kk instances, where rr is less than qq (i.e. $r<qr<q$). This results in an rr-dimensional vector of measurement data, denoted $Y(k)Y(k)$. The mathematical representation of this model is shown in equation (2):

$$Y(k) = CX(k) + V(k) \tag{2}$$

where $C = \mathrm{diag}(c_1 c_2 \cdots c_r)$ represents the observation matrix, and, $V(k) = [\upsilon_1(k)\upsilon_2(k)\cdots\upsilon_r(k)]$ denotes an additional sequence of measurement noise. The Kalman vector filter algorithm is defined by the following equations (3)-(5):

$$\hat{x}\,(k) = A\hat{x}(k-1) + K(k)[Y(k) - CA\hat{x}(k-1)], \tag{3}$$

$$K(k) = P_1(k)C^T[CP_1(k)C^T + R(k)]^{-1}, \tag{4}$$

$$P(k) = P_1(k) - K(k)CP_1(k), \tag{5}$$

where (3) is filter evaluation equation, (4) is the filter gain equation, where $P_1(k) = AP(k-1)A^T + Q(k-1)$, (5) is the covariance equation of the filter.

The Vector Kalman Filter algorithm operates as a prediction-correction mechanism characterized by its recursive filtering approach. This algorithm facilitates real-time signal processing using computational resources. Figure 5 shows the block diagrams representing the main program of the vector Kalman filter algorithms [9].

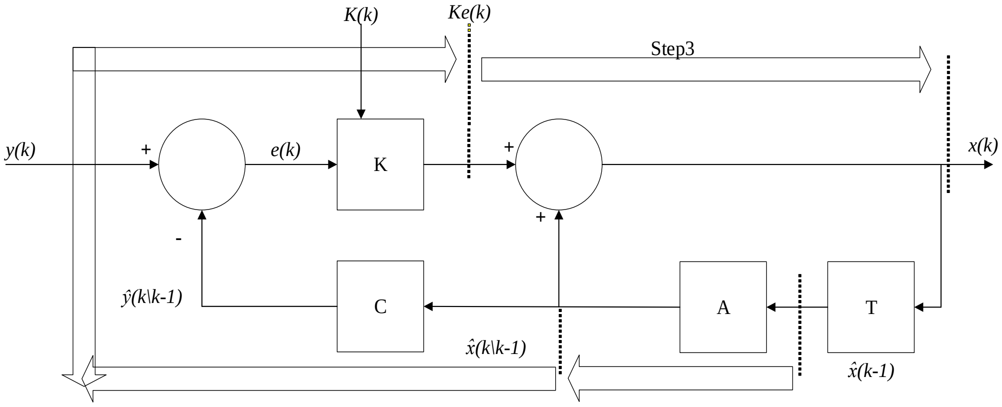

Fig. 6. Algorithm of the main Kalman program.

The air data filtered and processed by the fusion station provides a more accurate representation of air quality conditions. The processing of the incoming data can be seen as a transformation process that converts a set of input modes into corresponding outputs.

Therefore, it can be said that the global fusion algorithm [10] uses a simple single-layer neural network, as shown in Figure 7.

The output of each relay station forms the $X(k) = [x_1(k)x_2(k)\cdots x_n(k)]^T$ vector, which is the input group of the neural network, the output group accordingly forms the $Y(k) = [y_1(k)y_2(k)\cdots y_n(k)]^T$ vector.

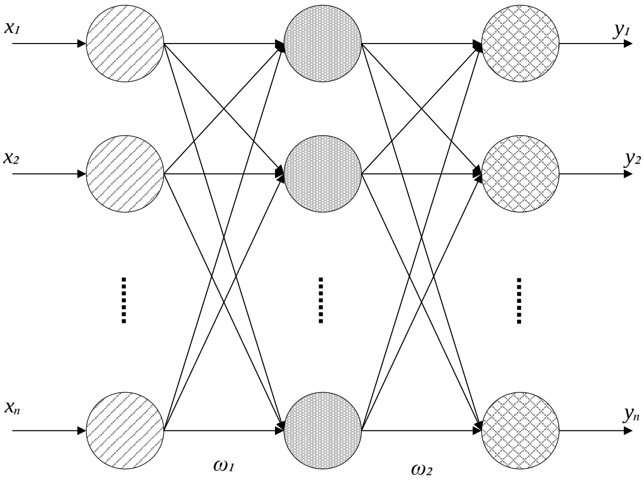

Fig. 7. A single-layer neural network framework for global fusion.

We use the sigmoid function (6) as the activation function for the network:

$$g(Z) = \frac{1}{1 + e^{-x}} \tag{6}$$

The implicit output of the network can be represented by equation (7):

$$x_{ni} = g\left(\sum_{j=1}^{N} \omega_{ij} Z_j + \theta_i\right) \tag{7}$$

For the excitation function of the output node we use a linear function (8), which reflects the total power of the network:

$$Y = \sum_{j=1}^{N} \omega_{IJ}^2 x_{ni} = f(x_1, x_2, \dots, x_n) \tag{8}$$

The (BP) algorithm is commonly used to train the neural network matrix. However, the traditional BP algorithm essentially functions as a least squares estimator, which often leads to reliability problems and is overly sensitive to outliers. To improve the robustness of the training process across multiple subsystems, we employ a more reliable method known as the Robust Backpropagation (RBP) algorithm [10]:

$$W_{IJ}(k+1) = W_{ij}(k) + \eta \delta_j O_I + a\left[W_{ij}(k) - W_{ij}(k-1)\right] \tag{9}$$

$$\theta_j(k+1) = \theta_j + \eta \delta_j + a[\theta_j(k) - \theta_j(k-1)]\,, \tag{10}$$

where η is the learning rate; α is the inertial constant, $\Psi(e) = \rho'(e)$, $\rho(e)$ is the Hampel function.

$$O_i = f_i(net_i) = 1 + exq\left(-\sum_j W_{ij} O_j - \theta_j\right)^{-1} \tag{11}$$

$$\delta_j = O_j(1 - O_j)\varphi(e) \text{ , } \delta_j = O_j\left(1 - O_j\right)\sum_k \delta_k W_{kj} \tag{12}$$

System performance testing

Algorithms for processing data collected from sensors are implemented to analyse the measurement results. To investigate the indoor air quality monitoring system, an initial setup is required. The air purifier and ventilation unit should be positioned within the line of sight of the infrared emitter. In addition, the microcontroller equipped with sensors is installed in the area of the room where people typically spend most of their time. Analytical methods were also used for testing purposes.

The experiment was carried out in several phases, measuring the CO_2 levels in the room in the presence of people and under conditions of inadequate ventilation. For experimental validation, the CO_2 concentration was measured using both the developed system and a reference analogue while the room was ventilated and unoccupied. CO_2 levels were then measured using the air quality monitoring system and compared with the analogue measurements. After the ventilation was stopped and people entered the room, CO_2 levels were measured again using both the developed device and the analogue after a period of 2 to 6 hours, showing an expected increase in CO_2 levels. To observe more pronounced changes, a gas burner was also introduced into the room.

An experiment was also carried out to assess the level of particulate matter. If the readings rise slightly, the system alerts the user with an audible signal and automatically activates the air purifier and ventilation unit. When the air quality improves, the system deactivates the equipment.

Conclusion

An indoor air quality monitoring system has been developed using the Raspberry Pi 4 microcomputer and the Arduino hardware programming platform. This system not only reports the air quality status in the room, but also actively influences it by using secondary loads that facilitate rapid air cleaning and ventilation. The control mechanism for these operations is primarily represented by three components dedicated to air quality management: an air purifier, a ventilation unit and a buzzer.

The system provides updates on air conditions every five minutes, with an adjustable reporting frequency. The software was developed using Python, Tkinter and the Arduino IDE. Users are informed about the air quality via a user interface hosted on the Raspberry Pi, as well as via an audible signal. Data transmission from the sensors to the control center is enabled by a Wi-Fi module. An information fusion algorithm is used to analyses the measured parameters.

The study of indoor air parameters is carried out in two phases. In the first phase, measurements were taken while a person was doing office work. In the second phase, measurements were taken after physical exertion, which indicated an increase in CO_2 levels in the room.

REFERENCES

[1] *Desnanjaya, I. G. M. N., & Arsana, I. N. A.* (2021). Home security monitoring system with IoT-based Raspberry Pi. Indones. J. Electr. Eng. Comput. Sci, 22(3), 1295. DOI: http://doi.org/10.11591/ijeecs.v22.i3.pp1295-1302

[2] *Taiwo, O., Ezugwu, A. E., Oyelade, O. N., & Almutairi, M. S.* (2022). Enhanced intelligent smart home control and security system based on deep learning model. Wireless communications and mobile computing, 2022, 1-22. DOI: http://doi.org/10.1155/2022/9307961

[3] *Diachok, R., Klym, H., & Vasylchyshyn, I.* (2021). Real-time mobile-based platform for determining level and location of radiation background. 22nd International Conference on Computational Problems of Electrical Engineering (CPEE), 1-4. DOI: http://doi.org/10.1109/CPEE54040.2021.9585271

[4] *Ranasinghe, R., Sounthararajah, A., & Kodikara, J.* (2023). An Intelligent Compaction Analyzer: A Versatile Platform for Real-Time Recording, Monitoring, and Analyzing of Road Material Compaction. Sensors, 23(17), 7507. DOI: http://doi.org/10.3390/s23177507

[5] *Zhang, J., Sheng, Y., Chen, W., Lin, H., Sun, G., & Guo, P*. (2021). Design and analysis of a water quality monitoring data service platform. Comput. Mater. Continua, 66(01), 389-405. DOI: http://doi.org/10.32604/cmc.2020.012384

[6] *Singh, A., Satapathy, S. C., Roy, A., & Gutub, A*. (2022). Ai-based mobile edge computing for iot: Applications, challenges, and future scope. Arabian Journal for Science and Engineering, 1-31. DOI: http://doi.org/10.1007/s13369-021-06348-2

[7] *Kumar, S., & Jasuja, A.* (2017). Air quality monitoring system based on IoT using Raspberry Pi. 2017 International conference on computing, communication and automation (ICCCA), 1341-1346. DOI: http://doi.org/10.1109/CCAA.2017.8230005

[8] *Ding, L., Wang, Z., Wang, X., & Wu, D.* (2020). Security information transmission algorithms for IoT based on cloud computing. Computer Communications, 155, 32-39. DOI: http://doi.org/10.1016/j.comcom.2020.03.010

[9] *Namdar, A., Samet, H., Allahbakhshi, M., Tajdinian, M., & Ghanbari, T*. (2022). A robust stator inter-turn fault detection in induction motor utilizing Kalman filter-based algorithm. Measurement, 187, 110181. DOI: http://doi.org/10.1016/j.measurement.2021.110181

[10] *Wang, Y., & Qing, D.* (2021). Model predictive control of nonlinear system based on GA-RBP neural network and improved gradient descent method. Complexity, 2021, 1-14. DOI: http://doi.org/10.1155/2021/6622149

РОЗРОБЛЕННЯ ТА ВПРОВАДЖЕННЯ СИСТЕМИ МОНІТОРИНГУ ЯКОСТІ ПОВІТРЯ В ПРИМІЩЕННІ НА ОСНОВІ ПЛАТФОРМ RASPBERRY PI ТА ARDUINO

І. Рудавський, Л. Ковалів, Г. Клим

Національний університет «Львівська політехніка»,
вул. С. Бандери, 12,79013 Львів, Україна
ivan2001rud@gmail.com , *lesia.kovaliv.mkisk.2023@lpnu.ua* , *halyna.i.klym@lpnu.ua*

У даній роботі представлено розробку комплексної системи моніторингу якості повітря в закритих приміщеннях, яка побудована на основі мікрокомп'ютера Raspberry Pi 4 та апаратної платформи Arduino. Розроблена система дозволяє не лише відстежувати параметри повітря, але й активно впливати на його якість. Це досягається завдяки використанню допоміжних пристроїв, які забезпечують ефективне очищення повітря, а також вентиляцію приміщення для підтримки комфортних умов перебування.

Система містить ключові апаратні компоненти: повітряний очищувач, вентиляційний блок та звуковий сигналізатор. Їхня робота забезпечує підтримання якості повітря на оптимальному рівні та оперативне інформування користувача про будь-які небезпечні зміни у складі повітря. Контроль здійснюється шляхом збору даних із сенсорів, які вимірюють параметри якості повітря кожні п'ять хвилин із можливістю регулювання частоти оновлення. Зібрана інформація передається на Raspberry Pi, де дані аналізуються та відображаються через спеціально розроблений інтерфейс, що дозволяє користувачеві оперативно відстежувати поточний стан повітря в приміщенні. Звуковий сигналізатор сповіщає про критичні зміни якості повітря, наприклад, підвищення рівня CO_2 або вмісту шкідливих речовин.

Програмне забезпечення системи розроблене з використанням мов програмування Python та інтерфейсу Tkinter, а також середовища Arduino IDE. Таке поєднання дозволяє ефективно організувати роботу всіх компонентів та забезпечити безперервний моніторинг якості повітря. Бездротовий зв'язок між сенсорами та центральною частиною системи реалізований завдяки Wi-Fi модулям, що полегшує установку та використання системи в будь-якому приміщенні.

Експериментальні дослідження проводилися у два етапи. На першому етапі система працювала в умовах звичайної офісної діяльності, де фіксувалися базові показники повітря під час робочого процесу. На другому етапі, який включав фізичну активність, було зафіксовано підвищення рівня CO_2, що свідчить про посилене забруднення повітря в умовах активної діяльності. Це підкреслює важливість регулярного моніторингу повітря для підтримання комфортних умов у приміщенні, особливо під час фізичного навантаження.

Отримані результати експериментів підтверджують ефективність роботи системи моніторингу та контролю якості повітря. Система здатна оперативно реагувати на зміни в навколишньому середовищі, забезпечуючи своєчасне очищення повітря та підтримання його на безпечному рівні, що є важливим для здоров'я та комфортного перебування людей у закритих приміщеннях.

Ключові слова: Система моніторингу, Raspberry Pi, Arduino, Алгоритм, Якість повітря

The article was received by the editorial office on 18.10.2024.
Accepted for publication on 01.11.2024.