ENERGY ENGINEERING AND CONTROL SYSTEMS

Volume 11, Number 2, 2025 199

Remote Monitoring System for Microclimate Parameters of Smart Home and Industrial Premises Based on ESP8266 Microcontroller

Pavlo Skibel, Fedir Matiko*, Ihor Demkiv

Lviv Polytechnic National University, 12 S. Bandery St., Lviv, 79013, Ukraine

Received: October 15, 2025. Revised: November 25, 2025. Accepted: December 18, 2025.

Abstract

The paper analyzes developments in the field of environmental parameter monitoring using microcontrollers and Internet of Things technologies, with results that justify the use of ESP8266 microcontroller. A monitoring system for microclimate parameters (temperature, humidity and atmospheric air pressure) in smart homes and industrial premises based on ESP8266 microcontroller has been developed, which demonstrated its effectiveness through a combination of hardware and software tools. The integration of BMP280 and AHT10 sensors ensured the accurate measurement of temperature, humidity and atmospheric air pressure. The asynchronous web server integrated into the ESP8266 allows for displaying the results as interactive, real-time graphs. In the developed system, users access data directly from the local network, ensuring functionality even without Internet access. Furthermore, the cost of the hardware (ESP8266 controller, BMP280 and AHT10 sensors) remains minimal, making the system accessible for widespread use. The implementation of a user notification system via WhatsApp Messenger using CallMeBot service became an important addition, allowing for the prompt notification of users about exceeding parameter limit values, even without logging into the web application. This approach increases the level of automation, convenience and reliability of microclimate control, making the system suitable for a wide range of tasks in domestic and industrial applications.

Keywords: microclimate parameters; smart home; monitoring system; microcontroller; Internet of Things; web application.

1. Definition of the problem to be solved

In the modern world, there is a rapidly growing interest in process automation across all spheres of human activity. In particular, the task of automating parameter monitoring in industrial and residential premises often arises. This task is relevant for production facilities in the food and processing industries, pharmaceutical manufacturing, warehouses, greenhouses, server rooms, and smart homes. In such environments, even a slight error in temperature, humidity or atmospheric pressure can lead to reduced product quality, spoilage of goods, disruption of technological processes or discomfort for people.

Internet of Things (IoT) technologies are widely used for remote monitoring of parameters in industrial and residential premises, enabling the creation of intelligent systems for collecting, transmitting, and processing data in real time. One of the most popular solutions in this field is ESP8266 microcontroller, which combines low cost, compact size, and a built-in Wi-Fi module. Thanks to these advantages, it is widely used for building wireless sensor networks, home automation, environmental monitoring, and many other tasks. The platform is supported in the

* Corresponding author. Email address: fedir.d.matiko@lpnu.ua

This paper should be cited as: P. Skibel, F. Matiko, I. Demkiv. (2025). Remote monitoring system for microclimate parameters of smart home and industrial premises based on ESP8266 microcontroller. *Energy Engineering and Control Systems*, Vol. 11, No. 2, pp. 199 – 209. https://doi.org/10.23939/jeecs2025.02.199

Arduino IDE development environment, which significantly simplifies firmware writing, and the ability to use the LittleFS file system allows for the implementation of an autonomous web server without the need for external hosting. Combining ESP8266 with BMP280 barometric-temperature sensor and AHT10 humidity sensor provides the capability to measure microclimate parameters – temperature, barometric pressure, humidity – and transmit them in real time to a web interface. The use of standard web technologies (HTML, CSS, JavaScript) along with the Highcharts library ensures convenient and interactive visualization of the measured data, allowing the user to dynamically select graphs, adjust the update period and monitor instantaneous values. This approach not only demonstrates the practical capabilities of ESP8266 in the IoT sphere but also creates a flexible, adaptive, and easily scalable system for further development and application in automating processes in daily life and industry.

2. Analysis of recent publications and research

In recent years, many developments and studies have been carried out in the field of environmental parameter monitoring using ESP8266 microcontrollers and IoT technologies, which indicates a steady trend towards the use of inexpensive and compact solutions for real-time data collection and processing.

Publication [1] presents an algorithm for the functioning of a monitoring system for technological process parameters based on mobile communication networks. Publication [2] develops the concept of a "smart system" for monitoring and protecting power equipment, which allows for timely response to dangerous parameter deviations. These works present monitoring systems developed based on programmable logic controllers, communication modules, and HMI tools from leading automation manufacturers. The disadvantage of the presented systems is the high cost of equipment, software, and, consequently, these monitoring systems themselves.

Publication [3] proposes a portable IoT station for measuring temperature, humidity, pressure, and air quality indicators based on ESP8266 microcontroller with AHT10 and BMP280 sensors. A similar approach is used in publication [4], which developed a system for monitoring air pollution in urban conditions using ESP8266, ensuring real-time data transmission to a server. The disadvantage of both approaches is the dependence on external servers or cloud services, which reduces the system's autonomy and complicates its use in conditions of unstable Internet connection.

Publication [6] presents a portable sensor node for robotic systems, which can integrate ESP8266 or ESP32 with many environmental sensors. The authors in publications [7] and [8] emphasize the role of IoT solutions in the "smart city" concept, where monitoring atmospheric parameters is a basic element of infrastructure management.

Publications [9]–[11] propose architectures for low-cost systems for environmental monitoring using ESP8266, Arduino, and cloud platforms. The authors in publications [13], [15], and [17] demonstrate practical skills in using IoT devices for data collection and analysis, which confirms the high level of interest in creating accessible solutions in this area.

Publications [16], [18] and [19] examine advanced approaches to forecasting and the educational use of IoT monitoring systems, indicating their potential for practical implementation. Publication [21] compares ESP8266 with other popular platforms (Arduino, Raspberry Pi), confirming its advantages in building compact and wireless systems.

In general, the review shows that ESP8266 remains one of the most promising solutions for implementing IoT environmental monitoring systems due to its low cost, support for wireless data transmission and ease of programming in the Arduino IDE environment. The task of developing IoT monitoring systems for each specific application remains relevant, as it provides the opportunity to develop simple and inexpensive monitoring systems with a set of functions adapted specifically for that application.

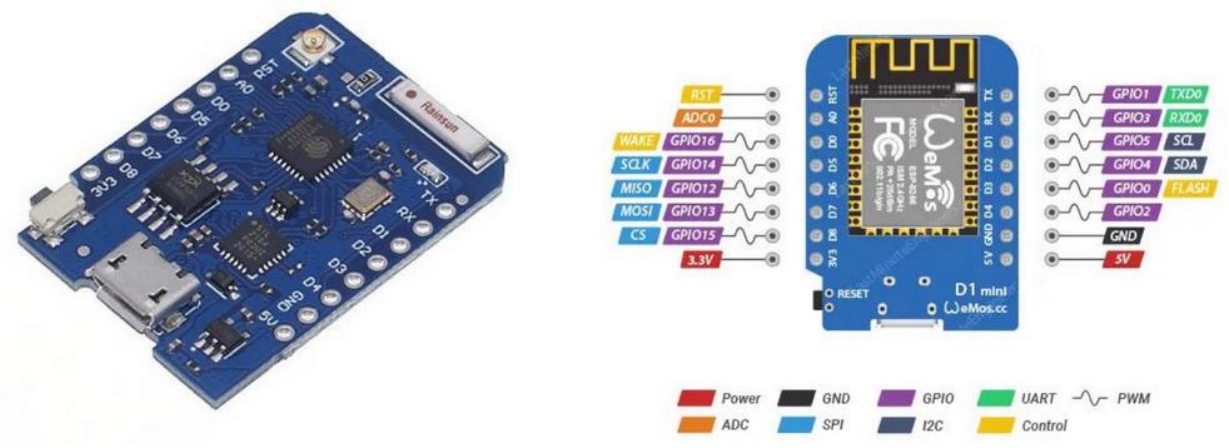
3. Purpose of the work

The purpose of this article is to create a prototype of an IoT monitoring system for microclimate parameters based on ESP8266 microcontroller, which ensures sensor integration, autonomous data collection and processing, organization of access to results via a built-in web server, and the possibility of their further visualization and analysis. The work involves creating the hardware part, developing software in the Arduino IDE environment and implementing a web interface for the operational display of environmental parameter dynamics.

4. Development and testing of the monitoring system

ESP8266 (WEMOS D1 mini module) microcontroller serves as the central component of the developed hardware–software environmental-parameter monitoring system. In addition to the built-in microcontroller, this module contains a flash-memory chip for program storage of 1, 2 or 4 MB and a CP2104 interface chip for converting the USB control signal to UART. The ESP8266 microcontroller has no built in on-chip non-volatile user memory; program execution is performed from external ROM through dynamic loading of required program fragments into the instruction cache. The ESP8266 consists of a 32-bit Tensilica Xtensa L106 processor with a clock frequency of up to 80 MHz, a Wi-Fi 802.11 b/g/n module, 14 input/output pins, SPI, I²C, I²S and UART interfaces, and a 10-bit ADC. General view and pin assignment of the ESP8266 microcontroller (WEMOS D1 mini module) are presented in Fig. 1.

Digital sensors BMP280 and AHT10 are connected to the ESP8266 (WEMOS D1 mini) to measure ambient temperature, barometric pressure and humidity. Both sensors are placed on the same board. The connection diagram of the board with BMP280 and AHT10 sensors to the microcontroller is presented in Fig. 2.



a) *b*)

Fig. 1. ESP8266 microcontroller (WEMOS D1 mini module) with CP2104 converter:
(*a*) general view; (*b*) pin assignment of the microcontroller.

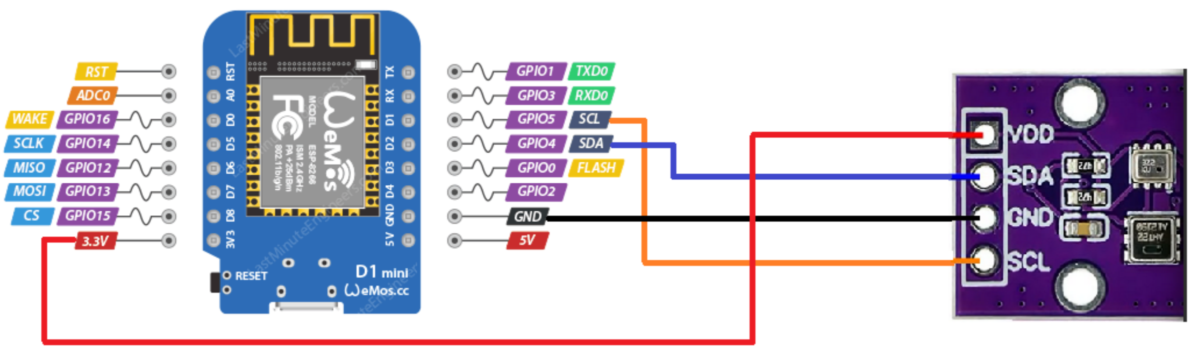


Fig. 2. Connection diagram of BMP280 and AHT10 sensors to WEMOS D1 mini module via the two-wire I²C interface.

These sensors offer significant advantages at low cost. The measurement error for BMP280 is about ±1 hPa for pressure and ±1 °C for temperature; for AHT10 the relative humidity error is only ±2 % RH. Both sensors feature low power consumption, fast response, and compact size, and are connected to the board through the I²C bus. Such a hardware combination ensures compactness and low cost of the solution. Thanks to the Wi-Fi module integrated on the ESP8266 board, wireless communication is provided, allowing implementation of an autonomous local web server for remote access to the measured values within a single local network. The LittleFS file system is used to store web-interface files (HTML/CSS/JS), enabling frontend hosting directly on the controller without any external servers or hosting services.

A functional algorithm of the monitoring system based on WEMOS D1 mini module has been developed, including the following steps (see Fig. 3). At the initial stage, network parameters (SSID and Wi-Fi password) are initialized, followed by mounting of the LittleFS file system that stores web pages and configuration files. Next, the I²C interface is initialized and the functionality of BMP280 and AHT10 sensors is checked to ensure correct measurement of temperature, humidity and barometric pressure. ESP8266 connects to the Wi-Fi network, obtains a local IP address, and launches an asynchronous web server based on the AsyncWebServer and LittleFS libraries, which enables user interaction through a web browser. After the web interface is started, initial limit values of the parameters are set; they can be adjusted in a dedicated settings section. The microcontroller periodically reads data from BMP280 and AHT10 sensors, compares them with the preset limit values, and updates the values on the web interface in real time. If any parameter exceeds the permissible range, the system automatically sends a notification to WhatsApp via the CallMeBot API, which improves responsiveness and user awareness of critical environmental changes. This algorithm ensures autonomous operation of the device, continuous data acquisition, and reliable information delivery through a user-friendly web interface and mobile application.

The software implementation of the functional algorithm was performed in the Arduino IDE environment using the ESPAsyncWebServer, ESPAsyncTCP and LittleFS libraries, as well as sensor drivers such as Adafruit_BMP280 and Thinary_AHT10. During initialization, the serial port is launched for diagnostics, the LittleFS file system is mounted, the I²C interface is initialized and sensor availability is checked, followed by connection to Wi-Fi and startup of an asynchronous HTTP server. After a successful connection, the controller outputs to the serial monitor the local IP address at which the web interface can be accessed (see Fig. 4).

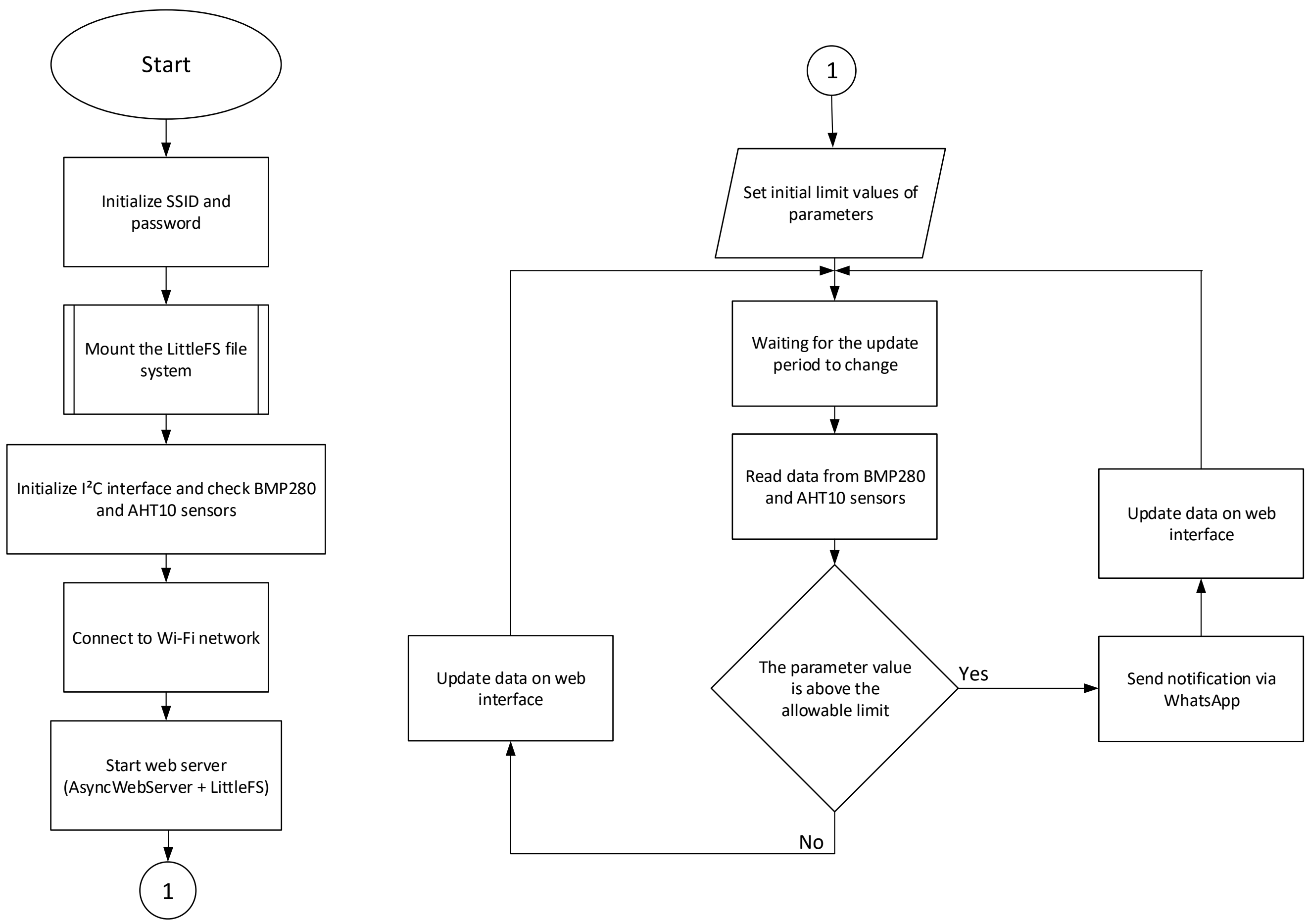


Fig. 3. Block diagram of ESP8266 microcontroller functional algorithm.

```
Connecting to WiFi..
Connecting to WiFi..
Connecting to WiFi..
192.168.175.187
```

Fig. 4. Obtaining a local IP address to access the web page.

The process of establishing Wi-Fi connection and reading the local IP address is implemented using WiFi.begin(ssid, password) and while (WiFi.status() != WL_CONNECTED) instructions in the controller program.

The asynchronous web server defines routes for the main page (/), for individual trend pages (/temp, /hum, /pres, /all) and for API endpoints returning current parameter values in plain-text format (/temperature, /humidity, /pressure). This enables the web server to perform periodic AJAX requests at specified intervals and retrieve updated data for trend plotting. An example of an API endpoint that returns the temperature value as unformatted data is:

```
server.on("/temperature", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", readBMP280Temperature().c_str());
  });
```

Sensor-reading functions include validation of measurements: if a NaN value is received or no response is detected, an error message is displayed in the Serial Monitor and an empty string or error message is returned to the HTTP request. This minimizes incorrect readings and helps to diagnose hardware issues during debugging. If a sensor (such as BMP280) is not initialized, the controller's operation is blocked to prevent further errors.

The user web application interface is implemented entirely with HTML/CSS/JavaScript and stored as files on the controller's flash memory formatted with the LittleFS file system, ensuring autonomous operation. On the main page, the user can select a window to monitor a specific parameter, view all parameters simultaneously, or open a settings page to adjust limit values (Fig. 5).

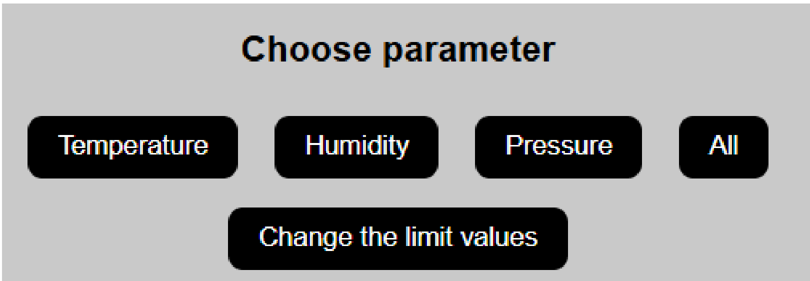
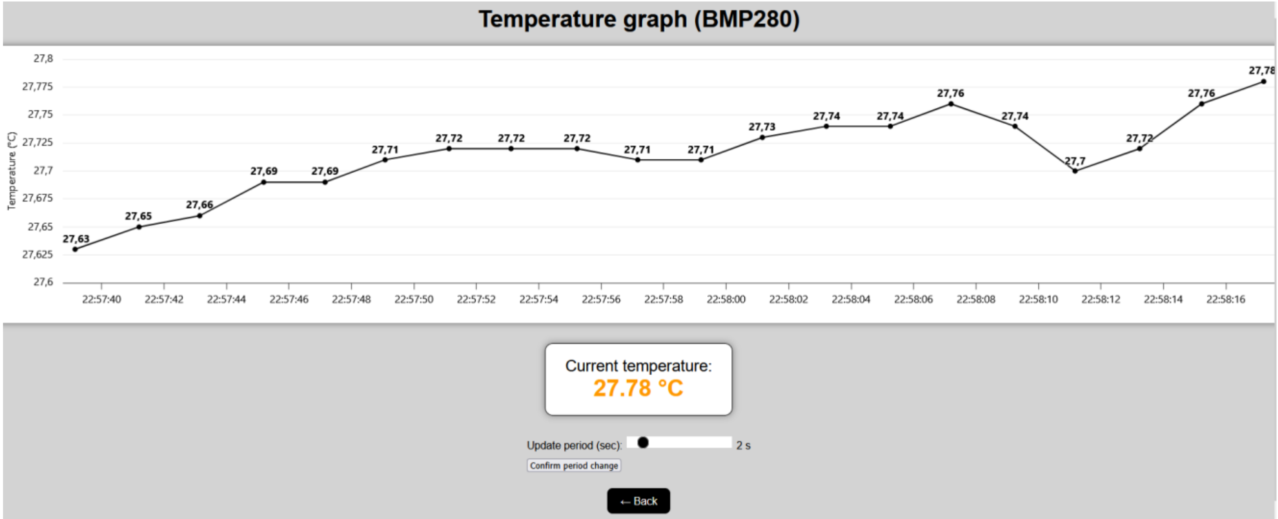
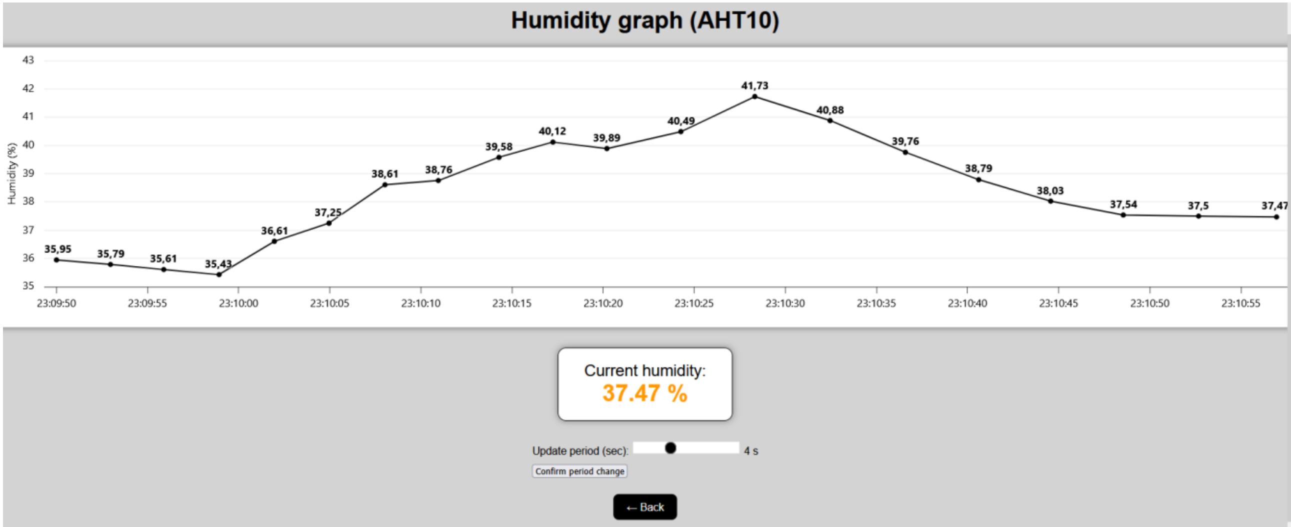


Fig. 5. Main page of the user web application.

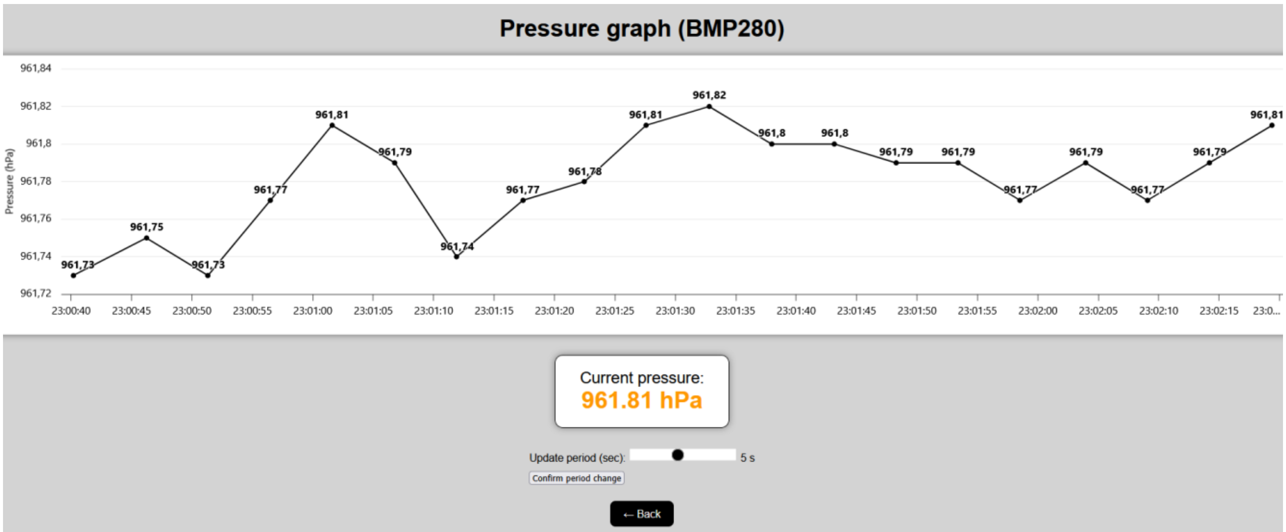
Each trend page contains an interactive chart built with the Highcharts library, a field showing the current parameter value, a slider to set the update period (1 to 10 s), and a confirmation button to apply changes. When the page loads, the program automatically sets a timer that executes an AJAX call to the corresponding API endpoint every *N* seconds. The update period is user-defined. Each received value is displayed both on the chart and in the instant value field. Fig. 6 shows the examples of web-application windows displaying temperature, humidity, and air-pressure changes, each with its own instant value field, update-interval slider and "Confirm interval change" button.



a)



b)



c)

Fig. 6. Web-application windows for monitoring: (*a*) temperature; (*b*) humidity; (*c*) air pressure.

To save the graph data after page reloads, the browser's built-in localStorage mechanism is used, allowing storage of trend point arrays in JSON format and restoration upon reload. Thus, the user can view measurement history without continuous connection to a server or database.

For the page displaying all the graphs, createChart(...) set of functions has been implemented to standardize appearance and behavior. Data updates are handled by a timer responsible for temperature, humidity and pressure, while the update period for all three parameters is controlled by a single slider element (Fig. 7).

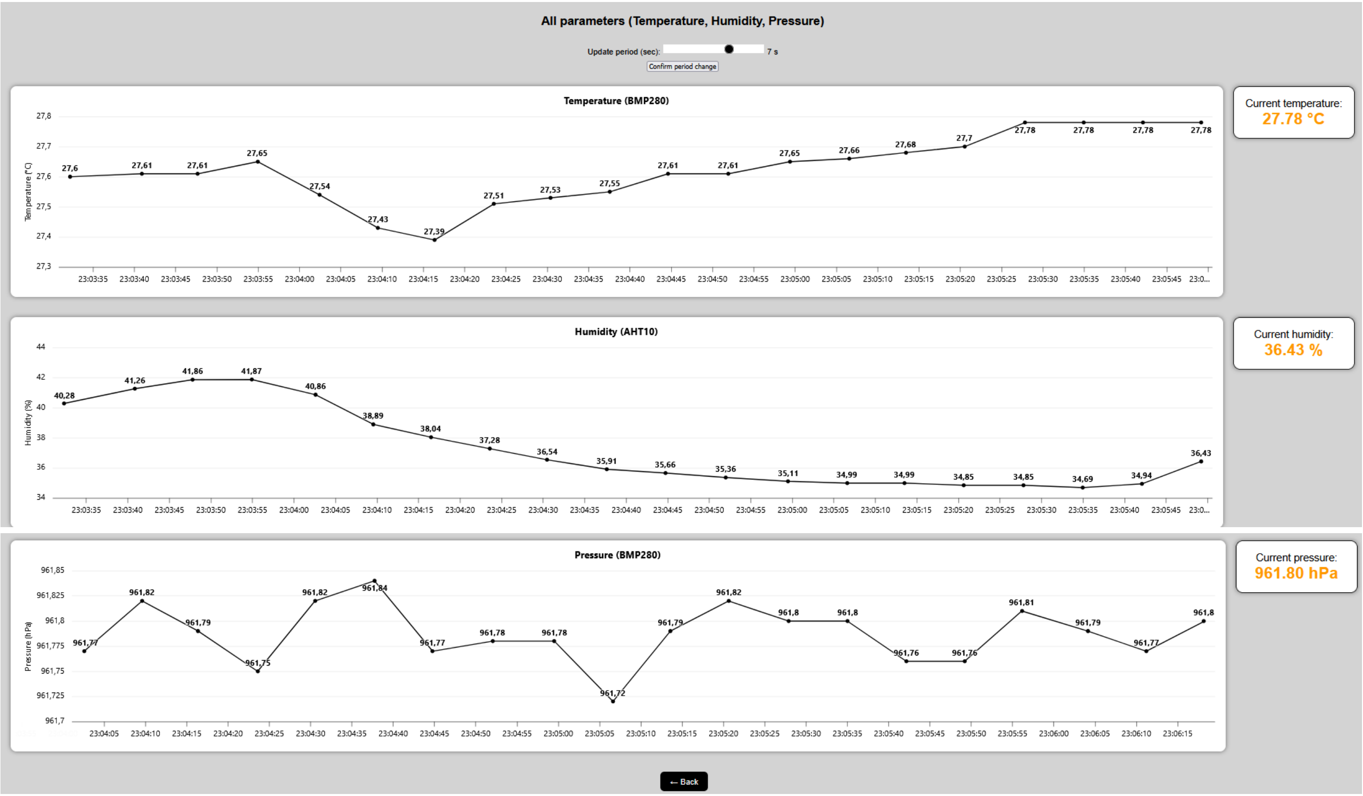


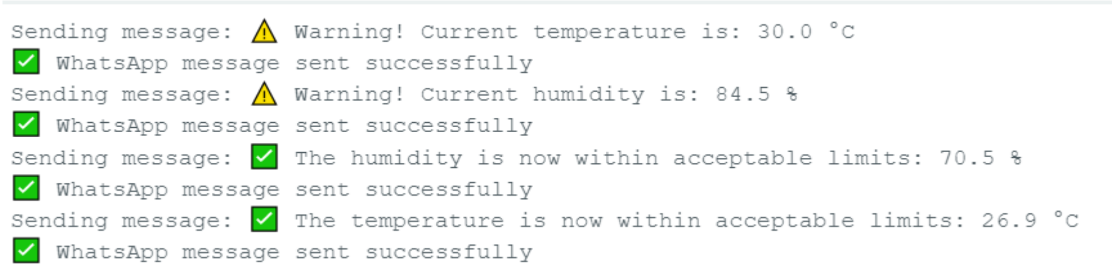
Fig. 7. Monitoring window of all parameters at initial update period of 5 s.

To enhance responsiveness to critical environmental changes, the developed system includes an alert algorithm via WhatsApp messenger. This feature complements the local web interface and enables the user to receive notifications of exceeded limit values of temperature, humidity and pressure directly on a smartphone, even without local-network access. The functionality is implemented through the CallMeBot API, which sends HTTP requests from ESP8266 microcontroller. sendWhatsAppMessage() function forms the request based on the current Wi-Fi connection status and message content.

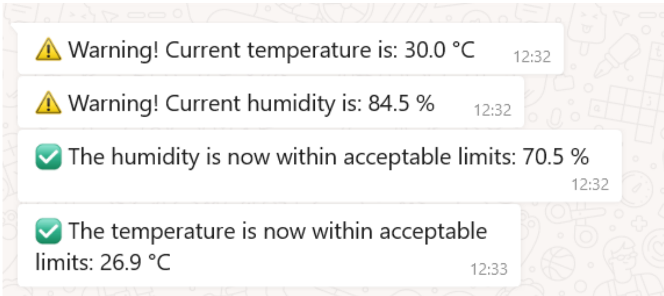
Within the controller program, the loop() cycle periodically reads data from BMP280 and AHT10, analyzes them and automatically generates WhatsApp alerts when set limit values are exceeded (see Fig. 8). This allows the user to respond promptly to parameter errors without opening the web application.

To set limit values, the user can open the "Change the limit values" page in the browser (see Fig. 9) and adjust them as needed.

The update periods for the bot and AJAX frontend updates should be selected conservatively to avoid connection interruptions. It is also important to keep the token and chat_id confidential and test the system with an active Internet connection, since WhatsApp notifications function only online.



a)



b)

Fig. 8. Example of messages in Arduino IDE Serial Monitor (*a*) and in WhatsApp messenger (*b*) when limit values are checked.

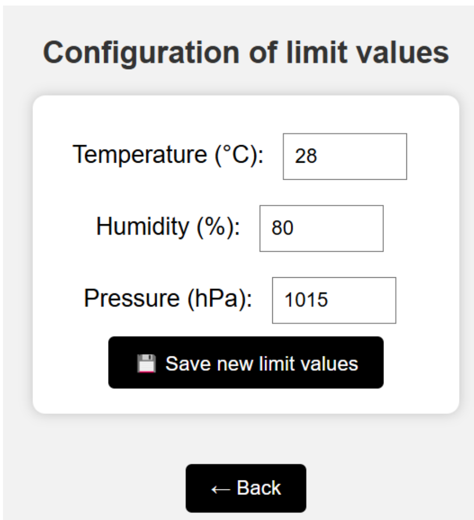


Fig. 9. Configuration of parameters limit values in the browser window.

During the work, the sensor initialization and data validity were verified, Wi-Fi connection and web interface availability were checked, the update period behavior was evaluated at different intervals (1–10 s) and system stability was assessed under moderate load with multiple LAN clients. It was found that at a 5-second update interval the system operates stably: charts update correctly, and current values are displayed with two decimal places accuracy. When the update period is changed, timers are restarted and frontend behavior is adjusted accordingly. At the minimum polling interval (1 s) and with many simultaneous clients, occasional chart-point skips or increased server response latency may occur.

Both the controller firmware and frontend include basic data-validation mechanisms: check of NaN, control of trend-series length, and limits of maximum message-buffer size. Use of the asynchronous web server (AsyncWebServer) prevents blocking of the main loop and enables parallel request handling, improving multitasking and allowing simultaneous operation of multiple browser tabs within one local network.

Integration of the WhatsApp bot significantly expanded the functionality of the developed monitoring system: it provides real-time remote alerts on out-of-range temperature, humidity and pressure, allows querying of current values, and enables dynamic limit values adjustment without re-flashing the controller. This communication method improves operational convenience and reaction speed while reducing the need for constant web-interface monitoring.

A comparison was made between the developed monitoring system and those presented in other publications. For example, publication [1] proposed an algorithm for generating and sending SMS messages about technological-process status implemented in a PLC program – a solution aimed at integration with existing automation systems using mobile networks for operator notification. Publication [2] describes a smart system for monitoring and protecting energy equipment based on a distributed PLC SIMATIC S7-1200 architecture with message-queue algorithms and SCADA/HMI integration. Both systems offer good industrial scalability but require complex and costly hardware and qualified personnel.

Publication [3] presents a portable station for measuring temperature, humidity, pressure and air quality based on NodeMCU ESP8266. However, its implementation is limited to simple data reading and server transmission, unlike the system proposed here, which provides an interactive web interface with flexible update period settings and real-time visualization, significantly improving functionality.

A similar approach is in publication [4], where an air-pollution monitoring system for urban environments is developed, but its architecture depends on an external data-processing server. In contrast, the implementation presented here, based on ESP8266 with the LittleFS file system, ensures complete autonomy: the web interface is hosted directly on the microcontroller, eliminating the need for external hosting.

Table 1. Comparison results of microclimate parameter monitoring systems.

Source	Advantages of the proposed system	Disadvantages	Brief description
[1] Shaleva et al., 2021	1) Integration with PLC and SCADA; 2) Use of mobile networks for notifications; 3) Focus on industrial applications.	1) Implementation complexity; 2) High system cost.	Monitoring system for technological process parameters using mobile communication networks.
[2] Shaleva et al., 2022	1) Scalability; 2) Application in industrial energy systems; 3) Integration capability with SCADA/HMI.	1) Implementation complexity; 2) High system cost.	Smart system for monitoring technological parameters and protecting energy equipment.
[3] Aashiq et al., 2023	1) Compactness; 2) Portability.	1) Limited web interface functionality, including lack of trend charts and update-frequency settings.	Portable environmental parameter monitoring station based on ESP8266.
[4] Gueye et al., 2024	1) Real-time data transmission to the server; 2) Suitable for urban environments.	1) Dependence on external server and Internet access.	Air-pollution monitoring system.
[6] Al-Okby et al., 2024	1) Integration with robotic platforms; 2) Support for ESP8266/ESP32 and various sensors.	1) Not intended for local visualization; 2) Requires integration with other systems.	Portable sensor node for robot-based monitoring systems.
[7] Anjali & Sukhada, 2022	1) Focus on scalability and urban applications; 2) Integration into the smart city environment.	1) High architectural complexity; 2) Dependence on external services.	IoT-based monitoring system for smart city applications.
[9]-[11] Gupta et al., 2019; Patil & Deshmukh, 2018; Shah & Memon, 2020	1) Low hardware cost; 2) Ease of implementation.	1) Dependence on Internet connectivity; 2) Reduced reliability in case of network failure.	IoT-based monitoring systems using ESP8266 and Arduino with cloud data transmission.

Publications [9]–[11] proposed low-cost models for environmental monitoring using ESP8266, Arduino and cloud platforms. Their main drawback is dependence on stable Internet connections and third-party services. In the proposed system, the user accesses data directly from the local network, ensuring operation even without Internet access. At the same time, the hardware cost (ESP8266 controller and BMP280/AHT10 sensors) remains minimal, making the system affordable for a wide range of applications – from educational projects to domestic and industrial use. This implementation demonstrates an efficient practical approach for building an autonomous ESP8266-based monitoring system combining simple hardware, flexible software architecture, and an intuitive web interface with interactive visualization. The developed system is suitable for educational laboratories, household microclimate control, IoT prototyping, and small local applications, and can serve as a foundation for further scaling to industrial or scientific tasks with additional data-collection and storage integration. The results of the comparison of the analyzed monitoring system characteristics are presented in Table 1.

5. Conclusion

The developed environmental parameter monitoring system based on ESP8266 microcontroller has demonstrated its effectiveness through the combination of hardware and software tools to create an affordable and reliable solution. The integration of BMP280 and AHT10 sensors ensured accurate measurement of temperature, humidity, and atmospheric pressure, while the built-in asynchronous web server enables real-time visualization of results as interactive graphs. An important enhancement of the system is the implementation of a user notification mechanism via the WhatsApp messenger using the CallMeBot service, which allows prompt alerts when limit values are exceeded – even without accessing the web application. This approach increases the level of automation, convenience, and safety of microclimate control, making the system suitable for a wide range of applications.

In the proposed system, the user can access the data directly from the local network, ensuring operability even in the absence of Internet connection. At the same time, the hardware cost (ESP8266 controller and BMP280 and AHT10 sensors) remains minimal, making the system accessible for broad use – from educational projects to household and industrial applications. The developed system is suitable for educational laboratories, household microclimate monitoring, prototyping of small-scale IoT applications and can also serve as a foundation for further scaling to industrial or research tasks through additional integration with data collection and storage services.

References

[1] Shaleva Volodymyr, Matiko Fedir, Krykh Hanna. (2021). Algorithm for operating the system of monitoring the technological process parameters using mobile communication networks. *Selected issues of electrical engineering and electronics : 16th International conference*, Rzeszów 13-15 September, 2021. – p. 1–4.

[2] Shaleva V., Matiko F., Krykh H., Roman V. (2022). Smart system for monitoring technological process parameters and energy equipment protection. *2022 IEEE 8th International Conference on Energy Smart Systems (ESS)* : proceedings (Kyiv, 12–14 October 2022). – p. 216–220.

[3] Aashiq, M.N.M.; Kurera, W.T.C.C.; Thilekaratne, M.G.S.P.; Saja, A.M.A.; Rouzin, M.R.M.; Neranjan, N.; Yassin, H. (2023). An IoT-Based Handheld Environmental and Air Quality Monitoring Station. *Acta IMEKO 2023*, 12, 1–9.

[4] Gueye A, Drame M.S., Niang S.A.A. (2024). A low-cost IoT-based real-time pollution monitoring system using ESP8266 NodeMCU. *Measurement and Control*. 58(10):1337-1345. doi:10.1177/00202940241306690

[5] Huang, Y., Zhao, Q., Zhou, Q., Wanchang, J. (2018). Air Quality Forecast Monitoring and Its Impact on Brain Health based on Big Data and Internet of Things. *IEEE Access*. PP. 1-1. 10.1109/ACCESS.2018.2885142.

[6] Al-Okby, M. F. R., Junginger, S., Roddelkopf, T., Huang, J., Thurow, K. (2024). Ambient Monitoring Portable Sensor Node for Robot-Based Applications. *Sensors*, 24(4), 1295. https://doi.org/10.3390/s24041295

[7] Anjali, D. G., Sukhada, S. (2022). IoT Based Environmental Monitoring System for Smart City. *Proceedings of the 2022 International Conference on Smart Technologies in Computing, Electrical, and Electronics*. IEEE. pp. 175-180.

[8] Dhananjay, K. B., Sandeep, K. S. (2020). IoT-Based Smart Environmental Monitoring System: Design and Implementation. *Proceedings of the 2020 International Conference on Communication and Electronics Systems (ICCES)*. IEEE. pp. 23-27.

[9] Gupta, H., Kumar, S. (2019). Environmental Monitoring System Using IoT. *International Journal of Computer Applications*, 975, 8887. https://doi.org/10.5120/ijca2019918820

[10] Patil, M. S., Deshmukh, S. P. (2018). Smart Environmental Monitoring System Using IoT. *International Journal of Science and Research*, 7(2), 113-117. https://www.ijsr.net/archive/v7i2/ART20192563.pdf

[11] Shah, A. M., Memon, S. (2020). Design of a Smart Environmental Monitoring System Using IoT. *International Journal of Computer Applications*, 975, 8887. https://doi.org/10.5120/ijca2020910665

[12] BMP280 - Ultra-Low Power Sensor for Barometric Pressure and Temperature. https://www.bosch-sensortec.com/media/boschsensortec/downloads/product_flyer/bst-bmp280-fl000.pdf

[13] S. Zafar, G. Miraj, R. Baloch, D. Murtaza, K. Arshad. (2018). An IoT Based Real-Time Environmental Monitoring System Using Arduino and Cloud Service. *Eng. Technol. Appl. Sci. Res.*, vol. 8, no. 4, Aug. 2018, pp. 3238–3242.

[14] Manisalidis, I., Stavropoulou, E., Stavropoulos, A., Bezirtzoglou, E. (2020). Environmental and Health Impacts of Air Pollution: A Review. *Frontiers in public health*, *8*, 14. https://doi.org/10.3389/fpubh.2020.00014

[15] J. Jo, B. Jo, J. Kim, S. Kim, W. Han. (2020). Development of an IoT-Based Indoor Air Quality Monitoring Platform. *J. Sens.*, vol. 2020, 1-14. 10.1155/2020/8749764.

[16] A. Kulkarni, D. Mukhopadhyay. (2018). Internet of Things Based Weather Forecast Monitoring System. *Indones. J. Electr. Eng. Comput. Sci.*, vol. 9, no. 3, pp. 555–557.

[17] N. A. Zakaria, Z. Zainal, N. Harum, L. Chen, N. Saleh, F. Azni. (2018). Wireless Internet of Things-Based Air Quality Device for Smart Pollution Monitoring. *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 11, pp. 65–69

[18] Gryech, I., Ben-Aboud, Y., Guermah, B., Sbihi, N., Ghogho, M., Kobbane, A. (2020). MoreAir: A Low-Cost Urban Air Pollution Monitoring System. *Sensors*, 20(4), 998. https://doi.org/10.3390/s20040998

[19] Carlos-Mancilla, M. A., Luque-Vega, L. F., Guerrero-Osuna, H. A., Ornelas-Vargas, G., Aguilar-Molina, Y., González-Jiménez, L. E. (2021). Educational Mechatronics and Internet of Things: A Case Study on Dynamic Systems Using MEIoT Weather Station. *Sensors*, 21(1), 181. https://doi.org/10.3390/s21010181

[20] Santos R. (2024) Getting Started with ESP8266 NodeMCU Development Board. *Random Nerd Tutorials*. https://randomnerdtutorials.com/projects-esp8266/.

[21] Patnaikuni D. (2017). A comparative study of Arduino, Raspberry Pi and ESP8266 as IoT development board. *Int. J. Adv. Res. Comput. Sci.*; 8(9): 2350–2352. https://doi.org/10.26483/ijarcs.v8i5.3959

Система віддаленого моніторингу параметрів мікроклімату для розумного будинку та виробничих приміщень на основі мікроконтролера ESP8266

Павло Скібель, Федір Матіко, Ігор Демків

Національний університет «Львівська політехніка», вул. С. Бандери, 12, Львів, 79013, Україна

Анотація

У статті виконано аналіз розробок у сфері моніторингу параметрів навколишнього середовища із застосуванням мікроконтролерів та технологій Інтернету речей, за результатами якого обґрунтовано застосування мікроконтролера ESP8266**.** Розроблено систему моніторингу параметрів мікроклімату (температури, вологості та атмосферного тиску повітря) у розумному будинку та виробничих приміщеннях на базі мікроконтролера ESP8266, яка продемонструвала ефективність завдяки поєднанню апаратних і програмних засобів. Інтеграція сенсорів BMP280 та AHT10 забезпечила точне вимірювання температури, вологості та атмосферного тиску, а інтегрований в ESP8266 асинхронний web-сервер дозволяє відображати результати у вигляді інтерактивних графіків у реальному часі. У розробленій системі користувач отримує доступ до даних напряму з локальної мережі, що гарантує працездатність навіть за відсутності доступу до Інтернету. При цьому вартість апаратної частини (контролер ESP8266 і сенсори BMP280 та AHT10) залишається мінімальною, що робить систему доступною для широкого використання. Важливим доповненням стало впровадження системи сповіщень користувача через месенджер WhatsApp із використанням сервісу CallMeBot, що дало змогу оперативно інформувати користувача про перевищення граничних значень параметрів навіть без входу у web-додаток. Такий підхід підвищує рівень автоматизації, зручності та надійності контролю мікроклімату, роблячи систему придатною для широкого кола задач у побуті та промисловості.

Ключові слова: параметри мікроклімату; розумний будинок; система моніторингу; мікроконтролер; Інтернет речей; web-додаток.