

Designing a Remote Room Monitoring System with Arduino and BME280 Sensor[†]

Gergana Spasova *  and Iliyan Boychev 

Department of Computer Systems and Technologies, Technical University of Varna, 9010 Varna, Bulgaria;
i.boychev@tu-varna.bg

* Correspondence: g.spasova@tu-varna.bg

[†] Presented at the International Conference on Electronics, Engineering Physics and Earth Science (EEPES 2025), Alexandroupolis, Greece, 18–20 June 2025.

Abstract

This article describes the hardware and software implementation of an application for monitoring parameters in a room. The reported values are visualized in a web application, which allows for the data to be viewed from anywhere in the world. The development made and the sensor used allow for measuring temperature, humidity, and pressure in a room. This is of particular importance for people who work with food products, materials that are dependent on temperature conditions, for families with children who want to maintain a certain home temperature and humidity, and many others. Each parameter is of particular importance for a person's health. All measurement data is stored in a database and can be used for statistics and analysis of changes in the room. The components used in the hardware implementation of the project are a Wi-Fi board ESP8266, Arduino UNO, and a sensor for simultaneous measurement of temperature, humidity, and pressure—BME280. The technologies used in the design of the software implementation of the project are JAVA, PHP, MySQL, Arduino IDE, hosting server, and domain name.

Keywords: monitoring system; Arduino; BME280 sensor; ESP8266; remote monitoring; temperature; humidity; pressure



Academic Editor: Grigor Mihaylov

Published: 27 August 2025

Citation: Spasova, G.; Boychev, I. Designing a Remote Room Monitoring System with Arduino and BME280 Sensor. *Eng. Proc.* **2025**, *104*, 52. <https://doi.org/10.3390/engproc2025104052>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The desire to monitor ambient thermal conditions has been an integral aspect of human existence throughout history. At higher temperatures, people of different ages react differently. It is very common for a person to wake up more tired than before going to bed when they go to sleep at high temperatures. At lower or higher humidity, older people often experience joint pain. Prolonged contact with insufficient moisture content in the atmosphere may result in dehydrated skin, nasal passages, and ocular surfaces, whereas excessive atmospheric moisture can trigger mucous secretion within nasal cavities and throat areas, resulting in blockage, frequent sneezing, and nasal discharge [1]. This creates the need to maintain relatively uniform humidity values in the room. Atmospheric pressure often leads to blood pressure problems, and people who are more vulnerable need to know whether a given environment is favorable for them. There are various solutions to the problem.

An automatic room temperature control system [2] based on the Arduino platform and a DHT11 sensor has been implemented. The system allows the user to set minimum and maximum temperature thresholds via a keyboard, and the measured temperature is visualized on an LCD display. Arduino processes the data from the sensor and makes

decisions to manage the temperature comfort in the room. At a temperature below the minimum threshold, a heater is activated, and when the maximum value is exceeded, a fan is turned on, the speed of which is controlled by pulse width modulation (PWM), depending on the temperature difference [3–5]. At a temperature within the set range, all devices are turned off, ensuring a stable and comfortable atmosphere in the room [6].

Article [7] presents a developed intelligent and economical human body temperature detector that uses a portable system based on Arduino UNO and an infrared sensor MLX 90614. The measured temperature is transmitted to the control unit via a Bluetooth module HM-10 and visualized on an LCD display (16×2). The proposed system demonstrates excellent performance, contactlessness, fast response, and low energy consumption.

Paper [8] describes the integration of a two-dimensional control system based on information retrieval methods to create a database and a data storage device. The system is based on sensors, which serve as the main components in the real-time monitoring of climatic conditions. The output data is visualized on an LCD screen, while the recorded information can be processed and presented as a virtual output.

Article [9] discusses the development of a monitoring system based on the Internet of Things (IoT) to ensure optimal conditions in warehouses to preserve the health of products. The system consists of two main layers: a sensor device that collects data and connects to the cloud, and a cloud service that hosts the environmental information. Various access methods, such as Wi-Fi, GPRS, and Ethernet, are used to store the collected data. Based on ATmega 328 as a microcontroller and PC/OS-III as an operating system, the system provides reliable data input. Article [10] describes the development of an agreement for integrating wireless sensor networks with the Arduino cloud platform, which allows for the transmission of temperature and humidity data.

In document [11], an automatic control solution is proposed for controlling the speed of a fan. An Arduino microcontroller is used, a LM35DZ temperature sensor, which converts the temperature into an electrical signal, the ATmega8/168/328 microcontroller (MCU) allows for dynamic and faster control, and the LCD display makes the system easy to use.

This paper presents the creation of an environmental monitoring system designed to track atmospheric conditions, including temperature, moisture levels, and barometric pressure within residential spaces. The implementation utilizes wireless connectivity modules, environmental sensing devices, an Arduino programming platform, and internet-based technologies. Information collection occurs through four monitoring units positioned strategically throughout different areas of a home: the main living area, cooking space, master sleeping quarters, and children's sleeping area. The primary capability of this solution involves real-time display of current atmospheric measurements for temperature, humidity, and air pressure at each sensor location. Additionally, the system generates historical trend visualizations showing temperature patterns from the preceding seven-day period based on collected and processed information. The gathered data is presented through both graphical charts and structured table formats for comprehensive analysis.

2. Application Topology

This implementation utilizes an ESP8266 wireless microcontroller operating in client mode to transmit HTTP POST communications to a server-side PHP handler, facilitating the storage of environmental measurements into a MySQL data repository. The system presents collected sensor information, temporal records, and additional metadata retrieved from the database storage. Remote access to this information is available globally, provided the hosting server remains reachable through network connectivity. The current implementation incorporates a BME280 environmental monitoring device interfaced with the ESP8266 microcontroller platform.

The following technologies are used to build this project:

- ESP8266 programmed with Arduino IDE;
- Hosting server and domain name;
- PHP script to insert data into MySQL and display on a web page;
- MySQL database to store readings.

Figure 1 shows the topology of the designed system.

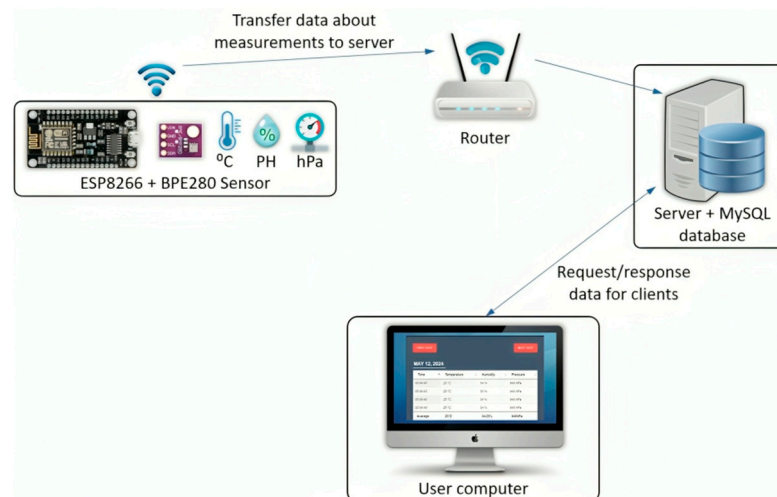


Figure 1. Block scheme of system.

The system consists of three components:

- Microprocessor system consisting of ESP 2866 microcontroller and BPE280 sensor measuring various parameters (temperature, humidity, and pressure). BPE280 sensor measures parameters in raw form and transmits data to ESP 2866. In ESP 2866, measurements are converted into an understandable data type. Then, ESP 2866 sends the converted data to a server via wireless connection via Wi-Fi;
- Server and database—the server has two functionalities. One is to receive measured data from ESP 2866 and store them in a database. The other functionality is to receive requests from the client application and send the data;
- Client computer—a specially developed WEB-based application is installed on it, which visualizes the measured values. This application sends requests to the server, receives the measurement data in response, and visualizes them.

3. System Design

3.1. Hardware Staging

The components used in the implementation of the project are as follows (Figure 2):

- Wi-Fi Board (ESP8266 or ESP32);
- BME280—sensor for measuring three types of quantities: Temperature, Humidity, and Pressure;
- Breadboard basic construction for electronic prototypes;
- Power supply battery—9800 mAh 3.7 V;
- Power supply battery slot;
- Connecting the two components.

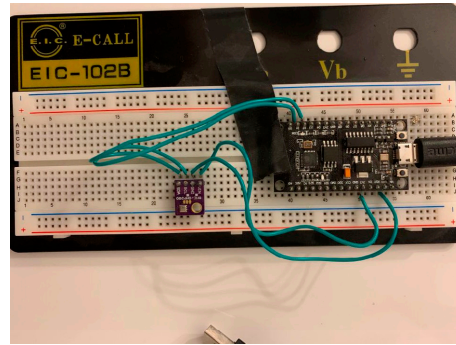


Figure 2. Arduino connectivity with other system components.

To obtain the electrical energy it requires, the microcontroller platform is linked to a portable power source through a Universal Serial Bus connector. This configuration provides operational autonomy and enables positioning the unit at various locations throughout a designated space. The sole limitation involves finite energy capacity, necessitating periodic monitoring to determine when the power cell requires replacement. The energy storage unit employed in this setup possesses specifications of 9800 milliampere-hours capacity at 3.7 volts output. The wiring configuration is illustrated in Figure 3.

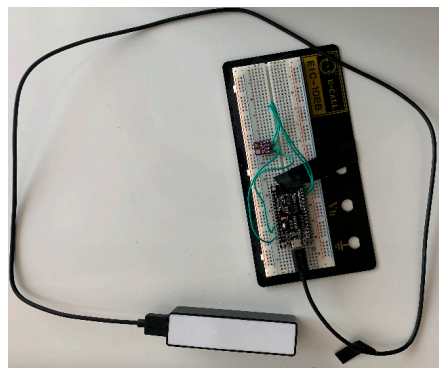


Figure 3. Powering the Arduino board with an external battery.

3.2. Software Implementation

Figures 4 and 5 present the block diagrams of the microprocessor system for measuring parameters and the client–server part for visualization.

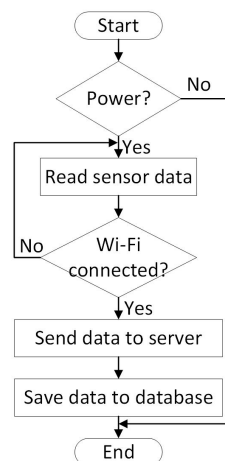


Figure 4. Flowchart data collection.

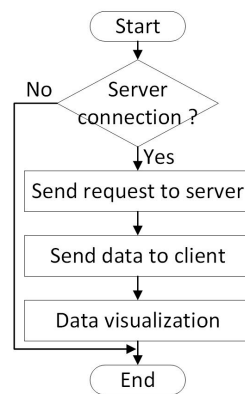


Figure 5. Flowchart data saving and collection.

The BPE280 sensor measures the physical parameters of temperature, pressure, and humidity of the environment and sends them to the ESP 2866 Microcontroller. Reading is in real time. The ESP 2866 is a client device that connects to a server via a wireless connection. If a connection to the server is established, the data is sent automatically. This allows the two devices to communicate without the need for human intervention. The server, upon receipt of new data, records it in a database. It is the connection to a client application that reads and visualizes them.

The client application connects to the server and reads the data stored in the database in real time, sending a request to receive it. In case of a successful response, the data is visualized in a GUI application using a format suitable for the user.

The user application requires registration. The roles that each user can obtain are the following:

- Administrator—Anyone with this role can see all rooms and measured parameter values and create/change/delete new and current users;
- User—depending on what rights are granted to him by the Administrator, the user can see data for certain rooms (some of them, all rooms, or only one room);

The database consists of five tables shown in Figure 6.

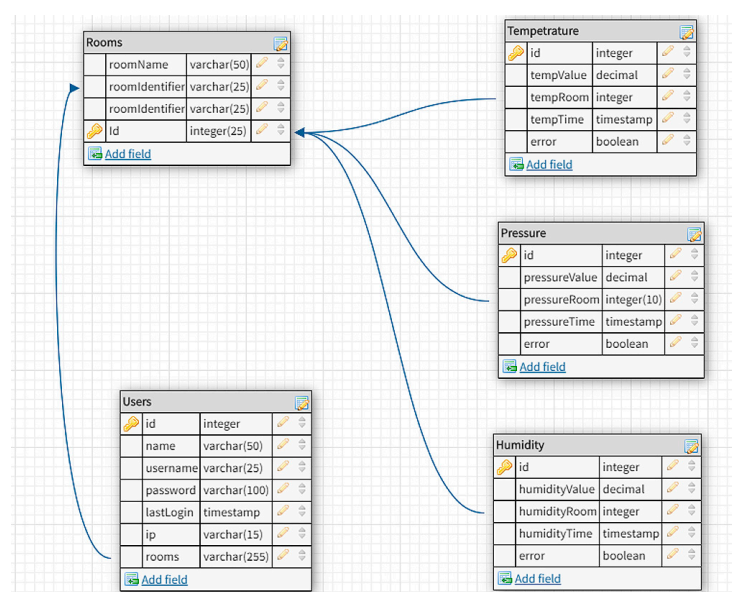


Figure 6. Database structure.

- Users table—the table contains all users registered in the system with their respective rights in it;
- The room table contains information about the premises/rooms;
- The temperature table contains information about the temperature in a specific room;
- The humidity table contains information about the humidity in a specific room;
- The pressure table contains information about the pressure in a specific room.

The user has the opportunity to view data about the temperature, pressure, and humidity in four rooms. For greater convenience and orientation, a photo depicting the given room has been added to each room.

Data is the main information provided by the application. For each room, three types of data are reported, which are visualized in tabular and graphical forms. Temperature is measured in Celsius ($^{\circ}\text{C}$), humidity in percent (%), and pressure in hectopascals (hPa).

The sensor makes measurements for the three quantities four times a day. Their average value is visualized in the last row of the table (Figure 7).

Time	Temperature	Humidity	Pressure
10:38:06	24 $^{\circ}\text{C}$	48 %	948 hPa
10:38:09	24 $^{\circ}\text{C}$	48 %	948 hPa
10:38:12	24 $^{\circ}\text{C}$	48 %	948 hPa
10:38:15	24 $^{\circ}\text{C}$	48 %	948 hPa
Avarage	24 $^{\circ}\text{C}$	48%	948hPa

Figure 7. Visualization the average data from measurements taken.

The graph shows the average temperature values for the current day plus a week ago. The graph period is a total of 8 days (Figure 8).

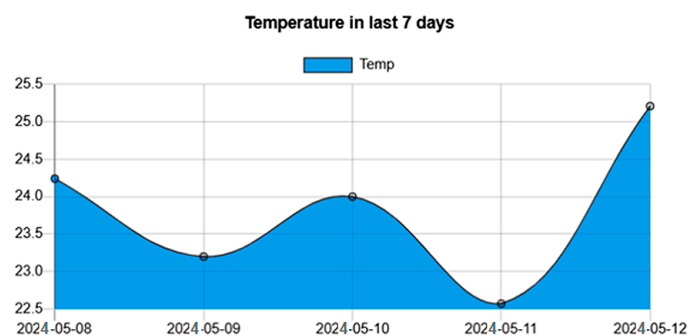


Figure 8. Graphical display of measured values.

The graph does not show data older than 8 days. If the user wants to see such information, he will not be able to. This information will only be stored in the database, but not in the application:

- On the Y axis, data is displayed showing the day when the temperature was measured. The data is displayed in the following format: YYYY-MM-DD;
- On the X axis, the measured temperature values are displayed with an accuracy of 1 tenth. The implementation assumes that a user can
- view data for specific rooms;
- view data for one or more rooms (Figure 9).

Figure 9. Data visualization capability.

One way to visualize the received data for a given room is in tabular form (Figure 10).

Time	Temperature	Humidity	Pressure
05:04:40	25 °C	34 %	949 hPa
05:04:43	25 °C	35 %	949 hPa
05:04:46	25 °C	34 %	949 hPa
05:04:49	25 °C	34 %	949 hPa
Avarage	25 °C	34.25%	949hPa

Figure 10. Displaying values in tabular form.

4. Experimental Results

4.1. Checking the Sensor's Battery Life

4.1.1. Test Scenario Steps

- Connect the device to the portable battery connector;
- Place the device in a location distant from any electrical appliances;
- Set it to read temperature, pressure, and humidity every four hours;
- Turn on the device.

4.1.2. Expected Result

Because data is read every four hours, the battery should last more than two weeks.

4.1.3. Result Obtained

The battery lasts almost a month.

4.2. Checking Reported Data for a Different Time Period

4.2.1. Test Scenario Steps

- Connect the device to the battery slot;
- Position the apparatus in a location distant from any electronic equipment;
- Set it to read temperature, pressure, and humidity every four hours;
- Turn on the device;
- Wait seven days.

4.2.2. Expected Result

- The data in the application table for a given room should be displayed for the current day plus six days ago;
- The application graph for a given room should visualize the average temperature value for each day.

4.2.3. Result Obtained

The data has been read against the above-mentioned requirements and meets the expected result.

4.3. Testing the Sensor to See if It Can Report Data at Certain Times

4.3.1. Test Scenario Steps

- Connect the device to the battery slot;
- Position the apparatus in a location distant from any electronic equipment;
- Set it to read temperature, pressure, and humidity for different hours;
- Turn on the device

4.3.2. Expected Result

The data is checked in the database and must correspond to the set time intervals.

4.3.3. Result Obtained

The results, according to the data in the database, are correctly and timely read.

4.4. Testing the Application with No Data Received

4.4.1. Test Scenario Steps

- Connect the device to the battery slot;
- Position the apparatus in a location distant from any electronic equipment;
- Set it to read temperature, pressure, and humidity for different hours;
- Set the device to read data for a specific room.

4.4.2. Expected Result

- The data results should not be visualized for rooms that are not mentioned;
- The data should be visualized only for the given target room;
- The tabular view for the remaining three rooms in the application should only say that there is no data.

4.4.3. Result Obtained

The test result in Figure 11 shows that the application works as required.

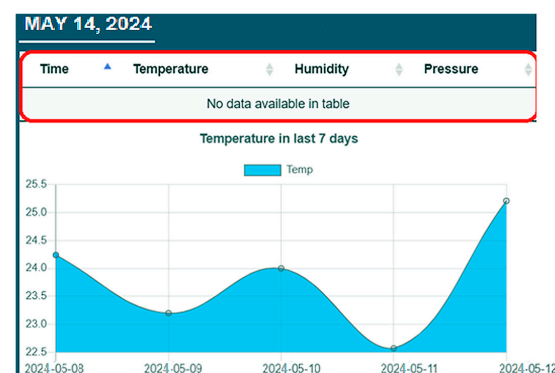


Figure 11. Result when disconnecting.

4.5. Checking the Application When More than Four Measurements Are Received per Day

4.5.1. Test Scenario Steps

- Connecting the device to the battery slot;
- Placing the device in a place away from any electrical appliances;
- Setting it to read temperature, pressure, and humidity every three hours;
- Setting the device to read data for a specific room.

4.5.2. Expected Result

- The data in the application table will visualize the first four measurements for the day;
- The mean calculation will represent the initial quartet of recorded data points.

4.5.3. Result Obtained

- The application data visualizes the first four measurements;
- The average value is from all measurements for the day.

4.6. Testing the Device for Operation at Sub-Zero Temperatures

4.6.1. Test Scenario Steps

- Connecting the device to the battery slot;
- Placing the device in the refrigerator compartment with the door slightly ajar;
- Device operation time—1 h;
- Setting it to read temperature, pressure, and humidity every five minutes for one hour;
- Setting the device to read data for a specific room.

4.6.2. Expected Result

The device sensor will read all data for the entire hour.

4.6.3. Result Obtained

The device works as required in Figure 12.

Time	Temperature
11:54:43	-3 °C
11:55:04	-2 °C
11:55:26	-3 °C
11:55:35	-1 °C
Avarage	-2.25°C

Figure 12. Results when reading negative temperature values.

4.7. Testing Two or More Devices to Send Data Simultaneously for the Same Room

4.7.1. Test Scenario Steps

- Connect the devices to the battery slot;
- Place the devices in a place away from any electrical appliances;
- Set them to read temperature, pressure, and humidity every four hours;
- Set the devices to read data for a specific room.

4.7.2. Expected Result

- The data in the application table will display the first four measurements of the day;
- The average value will be that of the first four measurements.

4.7.3. Result Obtained

- The application data displays the first four measurements (Figure 13);
- The average value is from all measurements for the day.

JUNE 15, 2024			
Time	Temperature	Humidity	Pressure
10:17:47	24 °C	48 %	950 hPa
11:17:38	24 °C	%	hPa
11:17:41	24 °C	48 %	950 hPa
11:17:44	24 °C	48 %	950 hPa
Avarage	24°C	36%	712.5hPa

Figure 13. Results of data read from two devices simultaneously.

4.8. Testing When the Power Supply to the Device Is Suddenly Stopped

4.8.1. Test Scenario Steps

- Connecting the device to the battery slot;
- Placing the device in a place away from any electrical appliances;
- Setting it to read temperature, pressure, and humidity for different hours;
- Setting the device to read data for a specific room;
- After the first result, the device is stopped.

4.8.2. Expected Result

The output within the online platform's data grid ought to display a single measurement, with the mean thermal value remaining identical.

4.8.3. Result Obtained

The test result in Figure 14 shows that the application works according to the requirements.

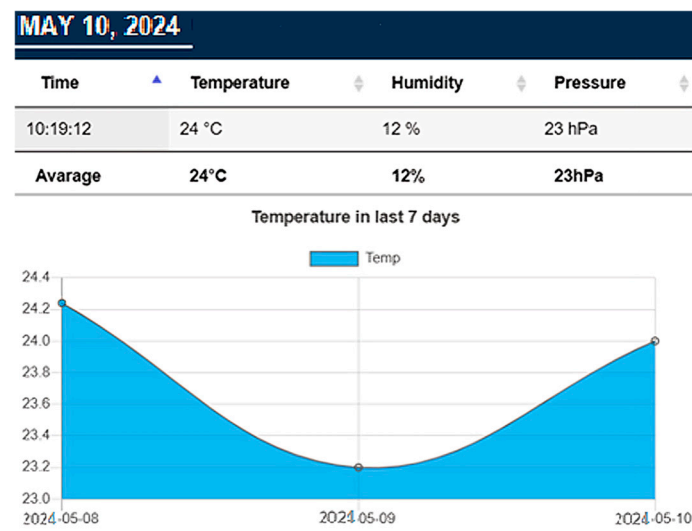


Figure 14. Results when power is cut off to the device.

All tests confirm that the sensor and the connected application function correctly and meet the specified requirements. The battery has better endurance than expected; data is validated on time and correctly, and the system copes with various scenarios such as lack of data, multiple devices, and power outages.

5. Conclusions

The developed application—Measuring Device has practical importance for the home life of every person. The main advantages of the developed system are: low price—it costs no more than USD 50; cross-platform—the development environment functions without problems both under Windows and under Macintosh OSX and Linux; simplified, clean programming environment; extensible open source software; extensible open source hardware.

The web application, which visualizes the data from the device, implements various functionalities used in already implemented developments. The design of the system predisposes to easy orientation and use. The web application is optimized to look the same on different web browsers on the Internet.

Upcoming development efforts will focus on optimizing power consumption longevity. Presently, the system continuously transmits connectivity verification requests to the wireless access point, which additionally diminishes the energy storage duration.

Author Contributions: Conceptualization, G.S. and I.B.; methodology, G.S.; software, G.S.; validation, G.S. and I.B.; formal analysis, I.B.; investigation, G.S.; resources, G.S. and I.B.; data curation, I.B.; writing—original draft preparation, G.S.; writing—review and editing, G.S. and I.B.; visualization, G.S.; supervision, G.S.; project administration, G.S.; funding acquisition, G.S. All authors have read and agreed to the published version of the manuscript.

Funding: The scientific research, the results of which are presented in this paper, was carried out under project NP3 within the framework of the scientific research activity inherent to TU-Varna, funded by the state budget. This research was funded by TU-Varna grant number NP3.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Arcenal, K.V.R.; Carmen, M.S.; Garcia, R.G. Effects of Low Humidity and High Humidity on the Nasal Area of the People. In Proceedings of the 2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET), London, UK, 19–21 May 2023; pp. 1–6. [\[CrossRef\]](#)
2. Debele, G.M.; Qian, X. Automatic Room Temperature Control System Using Arduino UNO R3 and DHT11 Sensor. In Proceedings of the 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 18–20 December 2020; pp. 428–432. [\[CrossRef\]](#)
3. Kumaran, M.; Vikram, I.; Kishore Kumar, S.; Rajesh Kumar, R.; Lokesh, S. Design of An Automatic Fan Speed Controlling System Using Arduino UNO. *Int. J. Intellect. Adv. Res. Eng. Comput.* **2018**, *6*, 2039–2042.
4. Kaushik, S.; Chouhan, Y.S.; Sharma, N.; Singh, S. Automatic Fan Speed Control using Temperature and Humidity Sensor and Arduino. *Int. J. Adv. Res. Ideas Innov. Technol.* **2018**, *2*, 8.
5. Siddika, A.; Nasrin, S.F. Design and Arduino-based Automatic Fan Control System Using PIR Sensor and LM35 Sensor. *GSJ* **2018**, *6*, 8.
6. Zungeru, A.M.; Mangwala, M.; Chuma, J.; Gaebolae, B.; Basutli, B. Design and simulation of an automatic room heater control system. *Heliyon* **2018**, *4*, e00655. [\[CrossRef\]](#) [\[PubMed\]](#)
7. N, Y.k.; A, O.K.; Abdulhameed, Z.N. Design and Implement a Wireless Temperature Monitoring System using Noncontact IR Sensor Based on Arduino. In Proceedings of the 2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG), Indore, India, 8–9 December 2023; pp. 1–5. [\[CrossRef\]](#)
8. Nanthini, N.; A, S.; K, S.S.; C, S.; S, S. Arduino Powered Smart Weather Monitoring System. In Proceedings of the 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 4–6 August 2021; pp. 229–235. [\[CrossRef\]](#)
9. Roy, A.; Das, P.; Das, R. Temperature and humidity monitoring system for storage rooms of industries. In Proceedings of the 2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN), Gurgaon, India, 12–14 October 2017; pp. 99–103. [\[CrossRef\]](#)

10. Sharmin, F.; Moon, N.N.; Saifuzzaman, M.; Hasan, A.; Hossain, M.A.; Nur, F.N. Humidity Based Automated Room Temperature Controller Using IoT. In Proceedings of the 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 12–14 December 2019; pp. 226–231. [[CrossRef](#)]
11. San, M.M.; Win, C.K.; Mon, K.Z. Design and Simulation of Fan Speed Control using Arduino UNO and LM35DZ. *Int. J. Eng. Res. Adv. Technol. (IJERAT)* **2020**, *6*, 6. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.