

THE DESIGN AND DEVELOPMENT OF A MOBILE APP TO REMOTELY CONTROL A TEMPERATURE CONTROL SYSTEM

Răzvan Constantin Niță¹, Bogdan Ciprian Doicin¹ 

¹ Automation, Computers and Electronics Department, Mechanical and Electrical Engineering Faculty,
Petroleum-Gas University of Ploiești, Romania

* email (corresponding author): bogdan.doicin@upg-ploiesti.ro

DOI: 10.51865/JPGT.2026.01.16

ABSTRACT

A temperature control system is one of the most widely used control systems nowadays. It is used in many environments, including people's houses. The advantages of such a system in a house are plenty, a couple of them being an increased comfort and long-term energy savings comparing to other temperature control systems. Another technology which is really gaining traction in the last years is the Internet of Things (IoT). This technology aims to connect every device to another one through the internet.

The present paper aims to present a temperature control system, designed for home use, which can be controlled remotely, by a mobile phone app and using the Internet of Things technology. The main advantage is that the heating system can be turned on or off pre-emptively, so that the user reaps immediately the benefits. As an example, the user can turn on the heating system in his home long before he comes home, thus having from the start the desired temperature. The proposed solution can be adapted and customized, according to the user's wishes.

Keywords: temperature control system, mobile app, Bluetooth communication, Kotlin, Arduino board

INTRODUCTION

The integration of the IoT technologies in the residential environment became a major development direction in the latest years, because of an increase of the interest in intelligent solutions to improve the comfort, the energetic efficiency and the users' safety. In this context, temperature control represents one of the most important functions of the Smart Home systems, with a direct impact on the energy consumption and the inner environment quality.

The recent studies emphasize that the users are more interested in accessible and easy to use solutions that allow the monitoring and control of the environment parameters through mobile devices. However, the most systems have limitations about their interoperability, high costs and technological complexity, which decreases their adoption scale from the home users.

In Romania, the pilot projects from the Smart Home field show that the users have an interest in the intelligent temperature control systems, but they have difficulties in using commercial systems, either because of the high costs or because of the lack of flexibility. This situation justifies the necessity of the development of customisable solutions that are easy to access and to integrate, allow the remote control of the execution elements and real-time parameter monitoring.

The last years' research shows that the interest for integrating IoT technologies in temperature control intelligent systems is increasing. The scientific literature emphasizes different approaches, from architectures which are dedicated to the HVAC control (HVAC means Heating, Ventilation and Air Conditioning) up until solutions that use machine learning for energy consumption optimization [1].

A significant number of studies have been focused on the role of intelligent sensors in thermal parameters monitoring. Works like Fraden show that the evolution of the temperature sensors allowed an increase in measurements' precision and reliability [2]. In the Romanian literature, Popescu and Iliescu emphasize the applicability of the intelligent sensors in IoT systems, where they become essential elements for thermal processes remote control [3]. Something similar can be found in [4].

An important direction of nowadays research is given by the usage of artificial intelligence and machine learning for thermal behaviour prediction. Zhao and Wen show that the neural networks-based models, advanced regression or deep learning algorithms can anticipate the evolution of a room's temperature, based on the existent history, ambient conditions and occupation patterns [5]. These approaches allow the systems to proactively react, reducing the energetic consumption and improving the thermal stability. Rawlings, Mayne and Diehl offer solid theoretical fundamentals for the predictive control, which is used more frequently in intelligent HVAC applications [6].

Although the modern temperature control solutions based on IoT and Smart Home technologies have clear advantages regarding comfort and energy efficiency, they also have a series of limitations and challenges.

A first limitation is the acceptance and adoption degree of IoT technology by the Romanian users. The study performed by Micu et al. shows that, although there is an increasingly higher interest for the Smart Home technologies, the perception over their usefulness and value is influenced by factors like fear of technological complexity and data security, fears that can delay the large-scale acceptance of the temperature control smart devices. Moreover, the component interoperability keeps being a practical problem in the Smart Home ecosystem implementation. Because of the various standards and the used protocols' differences, the sensor integration can be difficult, which can lead to fragmentation of the offered solutions. In time, they become difficult to manage.

Another challenge identified in the literature is data integration and the interoperability exchange of the IoT systems. Research regarding HVAC systems' monitoring and control through IoT shows that, although the technology allows real-time data collection, the integration and the efficient exchange of information between the different solutions remain difficult to implement, impairing the general performance of the intelligent system. While the modern temperature control system offers obvious advantages, they are still affected by the challenges regarding technological acceptance, data security, component interoperability and implementation costs. These limitations justify continuing the research and development of better solutions, which are more secure and easier to integrate in the existent home infrastructure [7].

The present research status shows a clear convergence between the intelligent IoT sensors, predictive control and cybersecurity, direction which transforms the temperature control systems into essential components of the modern infrastructure, efficiently energetic and safe [8].

IoT systems for temperature monitoring have been also the scope of many research projects. One of these projects was published in [9] and describes a IoT temperature monitoring system based on a Wi-Fi-connectable microcontroller (ESP 8266) and broadcasting the acquired data to a cloud platform. The user can view the temperature values through a web interface of an internet application. The system is mostly guided towards remote monitoring, with the possibility of report and/or alert generation. In this case, the effective control is mostly of a secondary importance, and the complexity of the system is skewed towards comm infrastructure and data management [10].

Another project can be found in [11] and it's called "IoT Temperature Monitoring System with Mobile App Integration". This project uses an ESP-32-type microcontroller and a DHT11 temperature sensor, which communicates via Wi-Fi with a database or cloud service. The mobile app, developed in Flutter, allows users to view the data in real time and to access the measurements history in a graphic format. A similar project can be found in [12].

This type of project illustrates a complete IoT approach, where the system is conceived to work in a permanently online network, which can expand towards advanced analysis or integration with other services.

Another system is presented in [13]. It describes a monitoring system for the fan speed control, which is designed for the industrial and semi-industrial applications. The system uses advanced microcontrollers (ESP 32, STM 32 or similar), temperature sensors and PWM-based control algorithms (PWM comes from Pulse Width Modulation) for fan speed fine tuning. The data are broadcasted to a professional IoT platform, where they can be monitored, analysed and correlated with other parameters.

In this case, the mobile app or the web interface are only components of a larger ecosystem, where the accent is on reliability, scalability and integration into existent infrastructures.

Finally, an IoT temperature monitoring system is presented in [14]. It uses two DHT22 sensors for continuous temperature measurement. The data are displayed on a liquid crystal display, and the monitoring logic is implemented in an Arduino Uno mainboard.

Integrating the predictive analysis techniques and machine learning algorithms in temperature monitoring became a highly intensely studied field, because it allows consumption optimization without impairing the occupants' comfort [15]. Recent studies show that using predictive control algorithms can lead to significant energy savings [16],[17],[18].

This paper answers the above needs by designing and implementing a complete system, based on accessible technologies and a mobile application, which offers the user a complete control over the heating and cooling systems. Comparing to the presented solutions, the one from this paper has a medium complexity and low cost, being easy to reproduce and adapt in educational or residential settings, unlike the more complex and expensive industrial counterparts.

In the oil and gas industry, the presented method is an excellent and useful way of controlling the temperature remotely. First, there is no need for the human to physically go to the installation, if he can do the same thing from afar, without losing time. The modern technologies can be integrated in the control systems, which are highly relevant for improving operational efficiency and monitoring in industrial environments. Second, the principles presented therein (remote monitoring, automation and real-time control) can be extended to the industrial systems.

The main purpose of this paper is to present a method to develop an intelligent, modular and accessible system, which monitors the system's temperature and humidity, processes the data locally and can be remotely controlled, through a Bluetooth interface. The method presented therein can be improved and adapted to suit everybody's needs.

MATERIALS AND METHODS

This paper aims to present an intelligent, modular and highly accessible system, which monitors the environment's temperature and humidity, locally processes the data, through an Arduino UNO R3 microcontroller and allows the remote control of its elements through a Bluetooth connection. The user interacts with the system through a mobile app, developed in Kotlin.

The used hardware and its main characteristics are presented below. The Arduino UNO R3 mainboard. This mainboard is the most important part of the presented solution, because it processes and coordinates the functioning of the entire system. It is based on the ATmega328P and offers a mature and stable platform. The hardware scheme of the Arduino mainboard is presented in **Error! Reference source not found.** [19].

The main characteristics of the mainboard are as follows: the microcontroller is ATmega328P, the operating voltage: 5V DC, the recommended power voltage: 7-12 V DC. Also, the number of digital I/O pins is 14, with 6 analogue pins. It has a flash memory of 32 kb, SRAM: 2 kb, EEPROM: 1 kb, clock frequency: 16 MHz and, as interfaces: UART, SPI, I2C. Although it's cheap and widely available, this mainboard has its limitations. The biggest limitation is the available memory. 32 KB of flash memory and 2 KB of RAM (or 8, for more advanced versions) can only take the application so far. Its low processing power can sometimes lead to sluggish sensor reading and display updates and cannot handle multitasking. It also has a high power consumption: about 50 mA continuously at 16 MHz [20].

The temperature and humidity sensor DHT11. This sensor can measure the temperature and humidity with an adequate precision, being a good fit for medium level applications. The used sensor communicates through a simple digital protocol, being easily integrated with the Arduino mainboard. Its purpose is to provide the necessary data to analyse the environment conditions and to start the control actions. The hardware scheme of the sensor is presented in Figure 2 [21],[22].

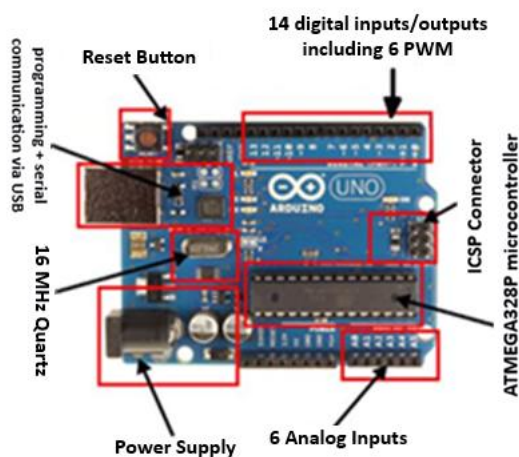


Figure 1. The Arduino mainboard.

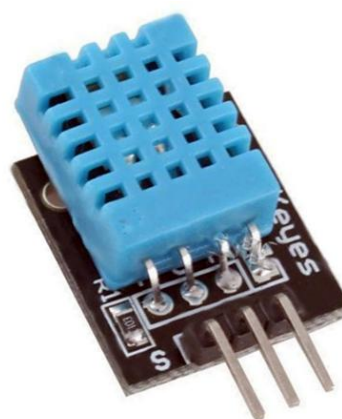


Figure 2. The DHT11 sensor.

The main sensor specifications are as follows: the temperature measurement interval: 0 – 50°C, the temperature measurement accuracy: $\pm 2^\circ\text{C}$ and the humidity measurement interval: 20-90%. Also, the power voltage is between 3.5 – 5.5 V DC, the electricity consumption: < 2.5 mA and the comm protocol: digital, one wire.

However, this sensor has its limitations. One of them is that it has a very slow sampling rate. It reads data once every two seconds, which makes unsuitable for real-time or fast changing environments. Its accuracy ($\pm 2^\circ\text{C}$) is low, and the measurement range is narrow (0-50°C). Moreover, extreme temperatures and humidity affect the sensor's performance and it's susceptible to electrical noise [23].

The HC-06 Bluetooth module. This module ensures the wireless communication between the Arduino mainboard and the mobile app. It works in slave mode and allows the bidirectional data sending through a serial protocol. The main advantages of this model stem in stability, low consumption and excellent compatibility with Android devices. The hardware scheme of this Bluetooth model is presented in Figure 3 [24].

The main module specs are as follows: power voltage: 3.3V DC, current consumption: 50 mA, frequency: 2,4 GHz. Also, the operating distance is 10 meters in open field, the baud rate: 9600 – 460800 bps, transmission power: +4 dBm, the reception stability: -84 dBm at 0.1% BET and the operating mode: slave.

This device is equipped with an integrated status led, which is very important for connection diagnosis and module behaviour monitoring. The Bluetooth communication device has three modes:

- Not connected. In this mode, the status led blinks fast. In this status, the module is powered on, the Bluetooth connection is active, the module is discoverable and pairable, but it is waiting for an external connection.
- Connected. In this mode, the status led is continuously on or blinks slowly. In this mode, the Bluetooth connection is established, the serial communication is active, and the module can exchange data with the paired device.
- Active data broadcast. In this mode, the status led is continuously on but sometimes it has some micro-oscillations of the light intensity. In this mode, the data exchange is active.

The LCD1602A liquid crystal display. This device is an alphanumeric display, with 16 columns and 2 rows, which is used for real-time display of the humidity and the system status. It offers a local visual interface, which is independent of the mobile app, facilitating the real-time system monitoring. The hardware scheme of the used LCD display is presented in **Error! Reference source not found.** [25].

The LCD characteristics are as follows: display type: LCD alphanumeric, dimension: 16*2 characters, controller: HD44780. As power voltage: 4.7-5.3 V, and the current consumption: 2 mA, without backlight. It also has two modes: 4 bit/ 8 bit and adjustable contrast.

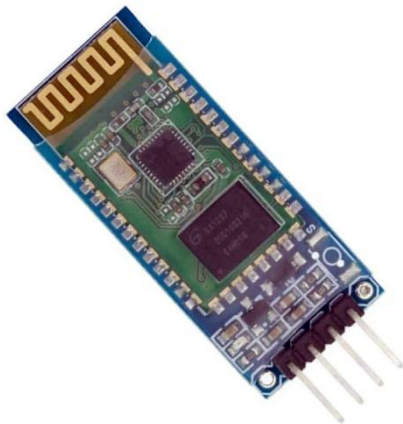


Figure 3. The HC-06 Bluetooth module.

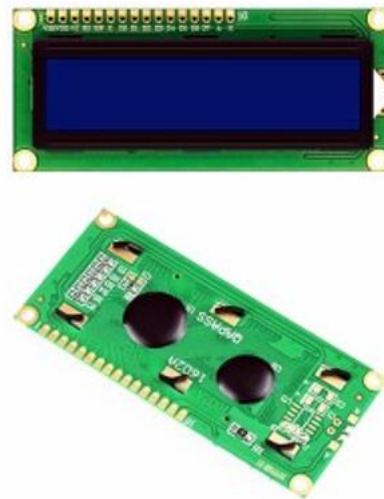


Figure 4. The LCD1602A display.

The MOSFET IRLZ44N transistor is used for operating the 12V elements like the fan and the light bulb. It allows the conversion of the higher currents using logic-level signals that come from the Arduino mainboard. Its characteristics (low internal resistance, high current capacity and logic-level compatibility) make it ideal for the execution elements control. The hardware scheme of the transistor is presented in Figure 5.

The main transistor characteristics are as follows: type: N-channel MOSFET logic-level, maximum current: 55 V DC, maximum current intensity: 47 A. The resistance (when turned on): 0.022 Ω , the dissipated power: 94 W, necessary gate tension: 5V and is ideal for PWM switching.

The BFB1012H fan. This fan is used to cool the environment when its temperature goes above the set threshold. It is controlled by the transistor, and it can be manually automatically activated, through the mobile app. An image of the fan is presented in Figure 6.

The fan's characteristics are as follows: the nominal tension: 4 – 13 V DC, maximum current: 0.8 A and the operating temperature: -10 – 60°C (14 – 140°F).

Hella light bulb P21/4W 12 V. The purpose of this light bulb is either as a heating device or as a signalling device, depending on the implemented logic. The bulb works at 12 V, and its moderate current consumption make it a perfect fit for being electronically controlled by the transistor. The image of the P21 light bulb is presented in Figure 7.

The light bulb's characteristics are as follows: nominal tension: 12 V, nominal power: 21 W for the main thin thread and 4W for the secondary thin thread, base type: BAY15d (with two delayed side pins), number of thin threads: 2. As light stream: approximately 460 – 500 lumens for the main thin thread and approximately 35 – 50 lumens for the secondary thin thread. Also, the consumed current is ≈ 1.75 A for the main thin thread and ≈ 0.33 A for the secondary thin thread. Its lifetime is 300 – 500 hours, depending on the producer and conditions. The functioning temperature: high and its material: heat-resistant glass and metal base BAY 15d.



Figure 5. The MOSFET transistor.



Figure 6. The BFB1012H fan.



Figure 7. The P21 bulb.

The development of the system within this paper can be synthesized in a couple of essential points:

- Local architecture, without using the cloud technology. All the decisions are made by the microcontroller, and the user communication is made directly via Bluetooth, without any intermediary servers or external databases.
- The technology is accessible and well-known: Arduino UNO R3, a DHT11 sensor, a Bluetooth module, an LCD display, transistor and fan.
- A native mobile app. The user interface is developed in Kotlin, using a clear architecture, which allows a clear integration with the Android OS.
- Mixed control (manual and automatic). The system allows both manual and automatic control over its parts.

The general system architecture is designed on four levels:

- Data acquisition.
- Processing and control.
- Actioning.
- User interface.

The system architecture is conceived so as it allows temperature monitoring and remote control by using a mobile interface. The proposed system is made of three main components:

- The hardware subsystem, which is based on the Arduino platform.
- The software subsystem, which is implemented on the microcontroller.
- The mobile App, developed in Kotlin, using the IntelliJ IDEA Integrated development editor [26],[27].

Functionally, the system architecture can be divided into the layers presented as follows.

The data acquisition layer. This layer is responsible for environment parameters measurement. The DHT11 sensor provides real-time values of the environment temperature and humidity, which are received by the Arduino UNO R3 mainboard, through a digital interface. The collected data are used for both display and decision-making purposes.

The processing and control layer. This layer is implemented by the Arduino mainboard. The mainboard executes the control logic, analyses the values received from the sensor and determines how each hardware piece will run. This layer also manages the Bluetooth communication, which receives the user commands and sends them to the mobile app.

The action level. This level is made from all the components that do real physical actions. For the proposed solution, these components are:

- A 12V fan, used for the environment cooling.
- A light bulb used as either heating or signalling.
- A LCD1602A liquid crystal display screen, which displays the system status to the user.

The control over the presented elements is made through the MOSFET IRLZ44N transistor, which allows task switching by using logical signals that came from the Arduino mainboard.

The user interface level. This level consists of the mobile app, which offers the user the possibility to visualize the system parameters and to send control commands back to the monitoring system. The GUI allows:

- Real-time temperature and humidity display.
- Enabling or disabling the light bulb or the fan.
- Setting some temperature thresholds.
- System status monitoring.

The connection between the app and the Arduino mainboard is made by Bluetooth, using the HC-06 module, which ensures that the connection is stable and it's easy to integrate in the mobile apps in general.

The component interaction can be described as follows:

- The DHT11 sensors measures temperature and humidity.
- The Arduino mainboard receives the data and processes them according to the control algorithm.
- Depending on the measured values or the commands that are received through the Bluetooth connection, the mainboard enables or disables the execution elements.
- The updated data are displayed on the LCD screen and broadcast to the mobile app.
- The user can intervene at any time through the app, sending commands through the system.

The block diagram of the entire solution is presented in Figure 8. From the diagram in Figure 8, the steps of the presented project are:

- The mobile app sends commands to the microcontroller, through the HC-06 Bluetooth connection.
- DHT11 measures the environment temperature and humidity and sends the data to the Arduino mainboard.
- The microcontroller will send the data towards HC-06 then to the phone, for the app to display the temperature and humidity. The same data will be displayed also on the LCD screen.
- Depending on the commands that are send from the phone, the Arduino mainboard will send control signals to the transistor, to start or stop the light bulb and/or the ventilator.

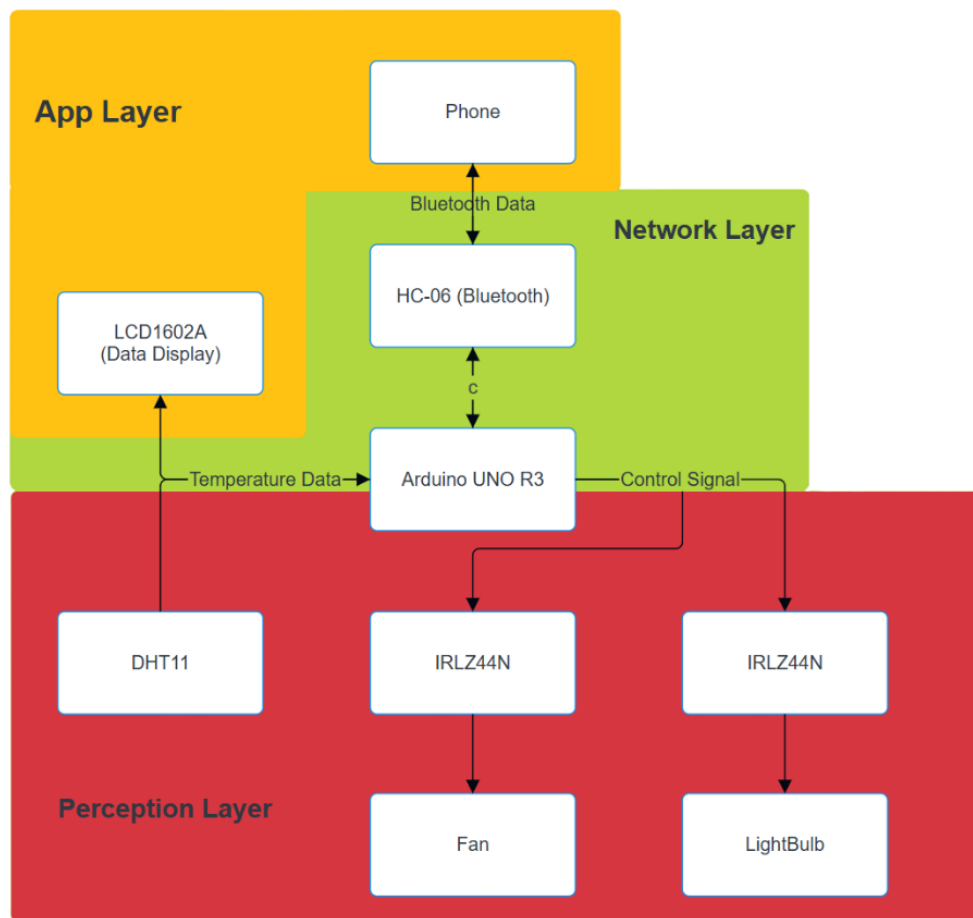


Figure 8. The solution's block diagram.

The electrical diagram of the solution is presented in Figure 9. The mobile app was conceived to be modular, so as each software component has a well-determined role within the communicating with the hardware process. The application is structured around three main modules: the GUI, the Bluetooth connection management and the data processing module. This clear separation allows for a clear, scalable and maintainable development.

The GUI layer is the visible component of the mobile application and facilitates the user's interaction with the temperature monitoring system's functions. The interface has two screens, accessible by swiping. Each of these pages is represented by a distinct fragment, responsible for a certain type of functionalities.

Through this layer, the user has access to the following features:

- Turning on the heat if the temperature is within or below a given range.
- Turning off the heat if the temperature is above a given range.
- Turning on the fan if the temperature is within or above a given range.
- Turning off the fan if the temperature is below a given range.
- Switching between automatic and manual control. When the application is on automatic control, the temperature is monitored automatically, between a given range. When the application is on manual control, the user decides when to turn on or off the heating.

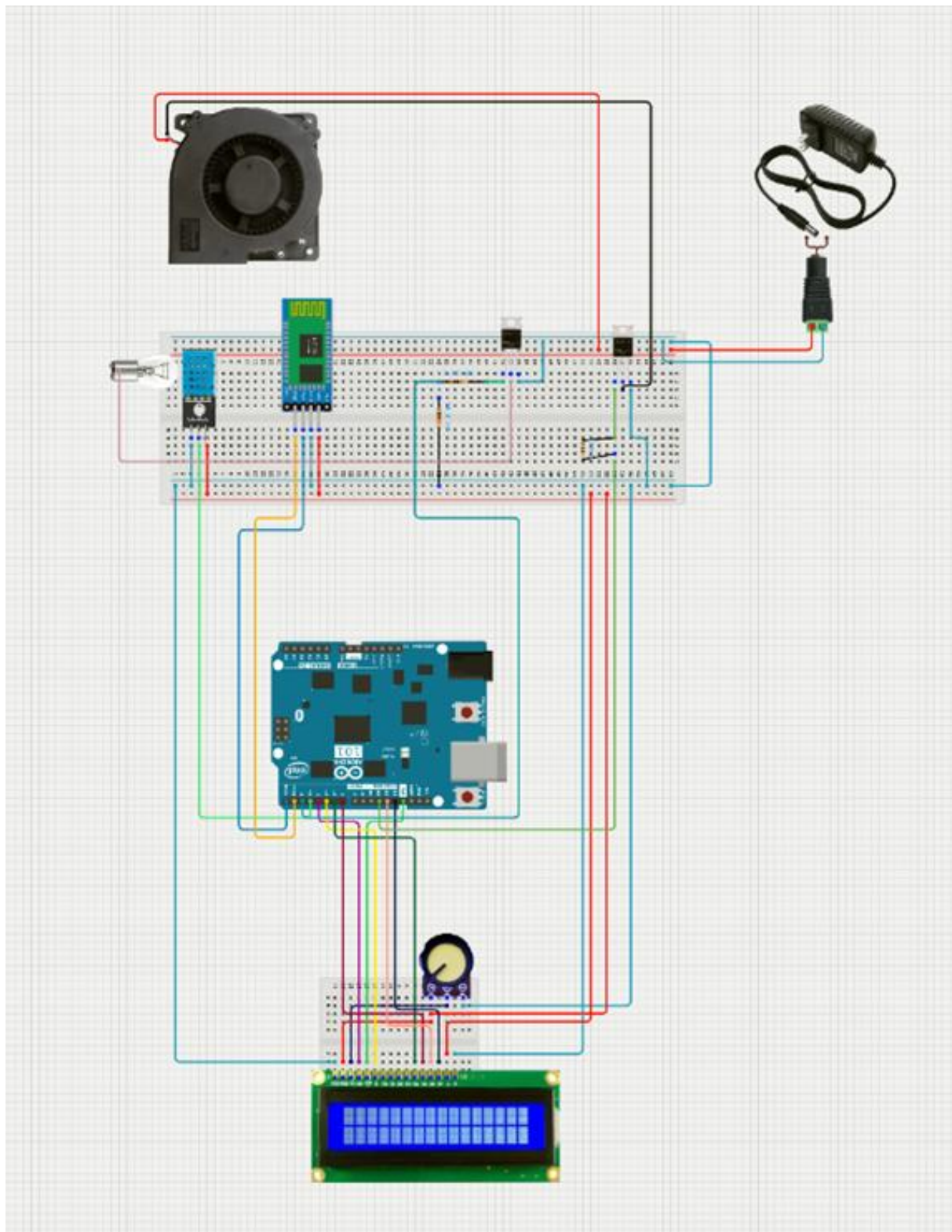


Figure 9. The solution's electric diagram.

The Bluetooth communication management layer is the responsible component for data sending and receiving between the mobile app and the temperature monitoring system. This module ensures the real-time link between GUI and the hardware device.

The data processing module is the intermediary component between the Bluetooth communication and the GUI. Its purpose is to interpret the information received from the microcontroller, to validate the existent data (including the user set values) and to generate the necessary commands to control the heating and ventilation system. The module ensures that the application functions coherently and prevents erroneous or incomplete data broadcasting towards the hardware.

This module has 4 main functions:

- Interpreting the data received from the microcontroller.
- Validating the data the user entered in the GUI.
- Generating the corresponding commands in a format accepted by the microcontroller.
- Permanent synchronization between the UI, local memory and hardware.

By separating the above functions, the application becomes more robust, easier to maintain and less error prone.

The designed application also allows the user to set his own preferences. These are stored locally, in the phone where the app is installed.

Finally, the designed app was tested from three different points:

- Hardware component testing. The following tests were made:
 - The sensor was tested through repeatedly reading the temperature and humidity, following the measurement stability and its behaviour towards temperature-controlled variations.
 - The Bluetooth module was tested by inducing all the three stages it could have and seeing the response of the status led. The connection with the mobile device was established in 1-3 seconds and the transmission tests showed that the module can receive and broadcast packages without data loss.
 - The transistor was tested by switching between the fan and the light bulb. The behaviour at quick on and off commands was followed. The transistor functioned correctly and stable, without variations.
 - The fan and the light bulb were tested in both manual and automatic command regime. They responded correctly to every performed test.
- Control algorithm testing. The following tests, which all passed, were performed:
 - Setting the temperature under the lower heating threshold. The heating was turned on.
 - Setting the temperature above the higher cooling threshold. The cooling was turned on.
 - Setting the temperature between the lower and the higher threshold. Both heating and cooling were turned off.
- Full system testing. The following tests were performed, which all passed:
 - Modifying the ambient temperature and watching the app updating the changes.

- Modifying the temperature thresholds from the app and watching the hardware respond to the modifications.
- Sending manual commands from the app and watching the instantaneous response of the ventilator and the light bulb.
- System's behaviour on extended periods of functioning.

RESULTS AND DISCUSSIONS

The entire solution is controllable by the mobile phone app. By using the manual control, the app's screen is presented in Figure 10.

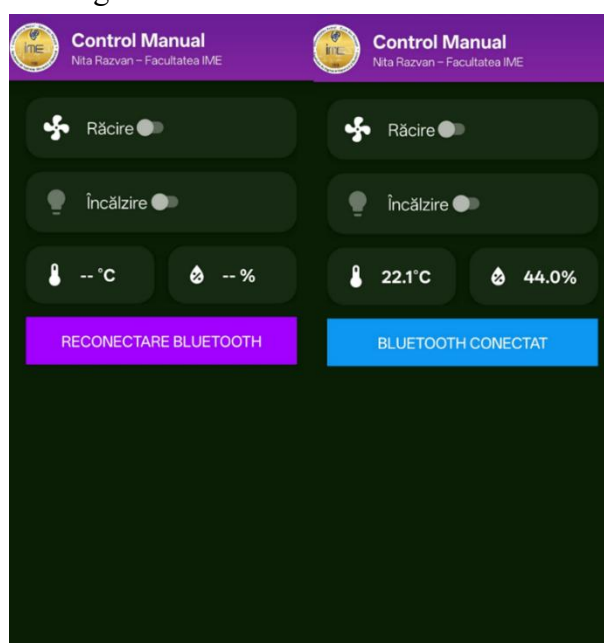


Figure 10. The app's manual control interface.

In the Figure 10, the user can manually choose the heating or the cooling function, by tapping the switches in the upper part of the screen. The switches are mutually exclusive. In the bottom part of the screen, there is a button which allows the app to connect to the system via Bluetooth. The button's caption reflects the connection status at this point. The automatic control of the app allows the user to choose a lower and a higher threshold for the desired environment temperature. The system automatically heats or cools the environment, based on the user settings. The app control screen is presented in Figure 11. Also, the Figure 11 shows the button in the bottom of the screen, which allows the user to turn on or off the automatic control. If the automatic control is turned off, the manual control is automatically turned on and vice-versa.

The history of the monitored temperature and humidity is stored locally and can be seen from a third app screen, which is presented in Figure 12. The graphs can be zoomed in or out and can be panned, as the user wishes.

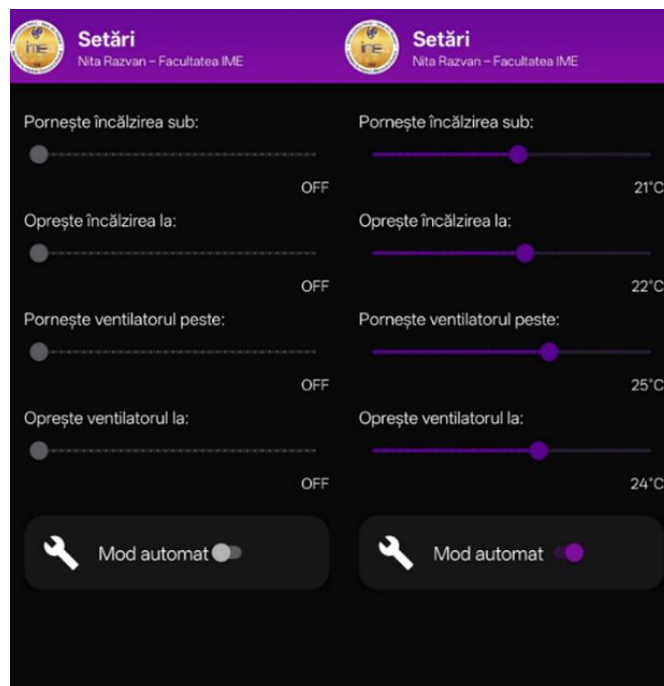


Figure 11. The app's automatic control interface.

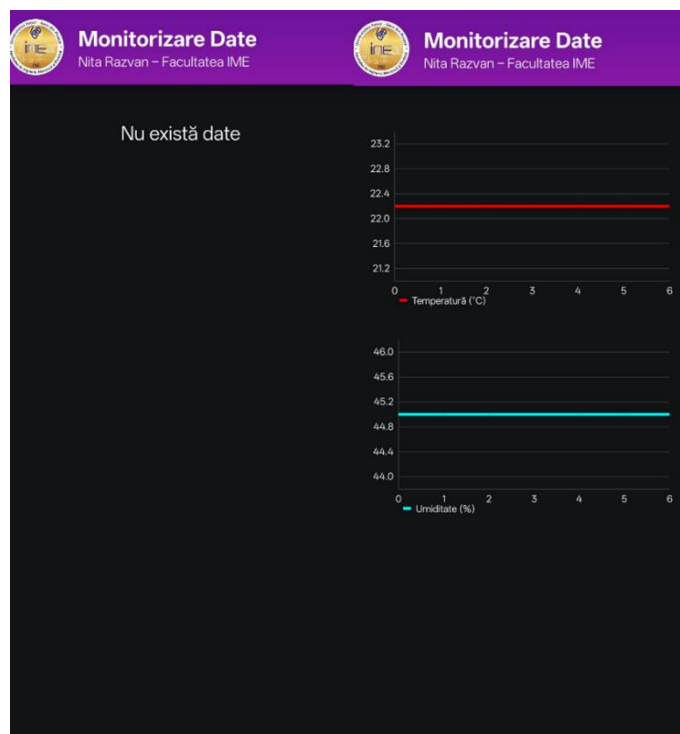


Figure 12. The app's history interface.

The proposed solution worked as intended, on both automatic and manual modes, offering the user the desired flexibility, according to his needs. The app's flowchart is presented in Figure 13.

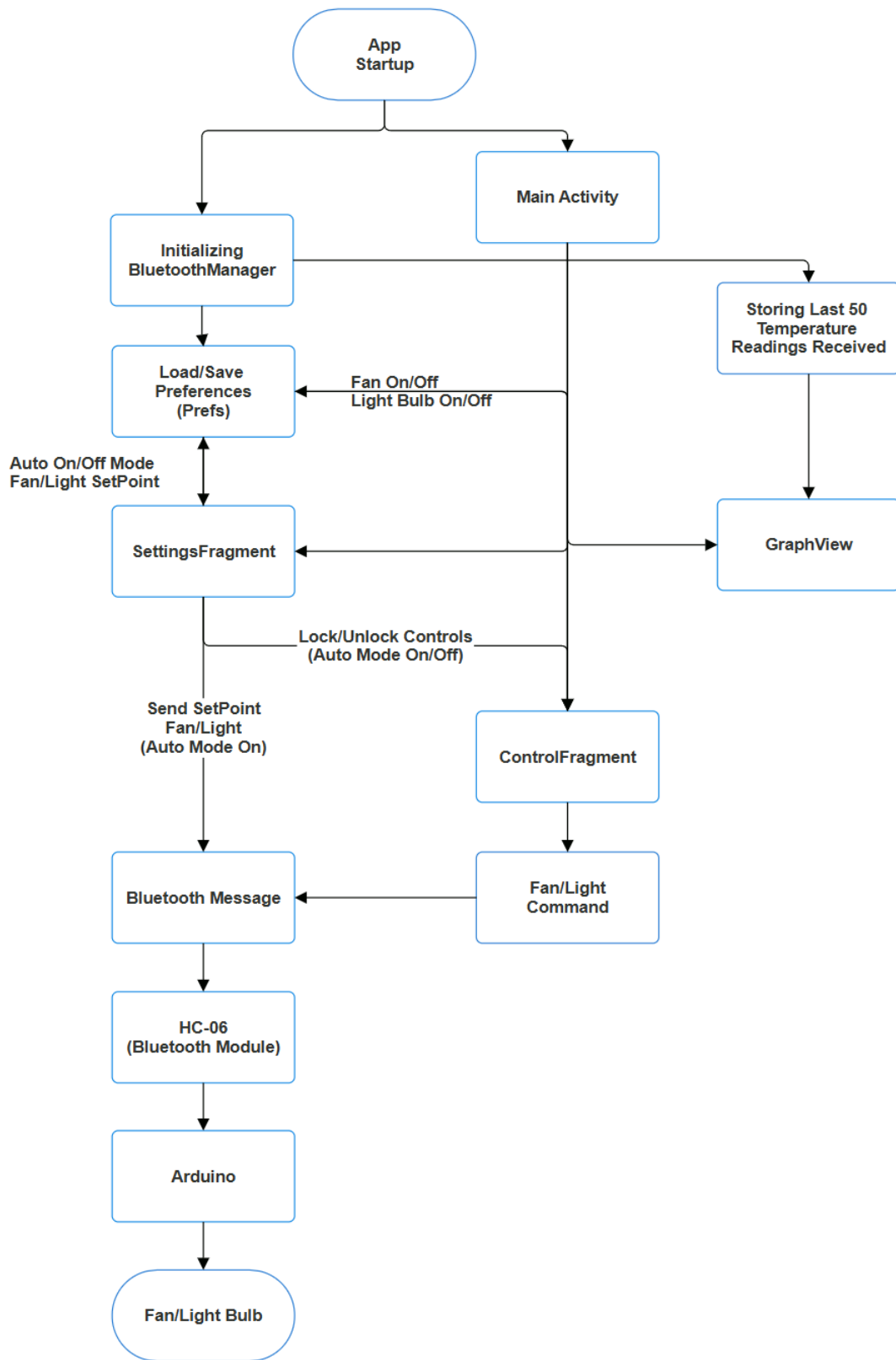


Figure 13. The app's flowchart.

CONCLUSIONS

The proposed solution offers a flexible approach to setting the environment temperature according to the user's needs. The main advantages of this approach are:

- Intelligent automatic mode. The system can work autonomously, based on the user's pre-set values. This approach reduces the energy consumption, maintains the desired temperature without human intervention and eliminates the overheating risk. The behaviour is akin of an intelligent thermostat.
- Real-time Bluetooth communication. The app sends and receives real-time data. This means that the updates are not delayed at all, which offers a fluid and smooth experience.
- Modern and intuitive interface. The app design is easy to use even for the non-tech users. Moreover, the dark theme reduces the eye strain.
- Displaying temperature and humidity history. The graphs and the history are displayed in real-time, making parameter monitoring easier. It's a huge advantage for optimization and diagnosis.
- Error preventing and hardware protection. The app verifies the data correctness and consistency before sending anything to the physical system. Moreover, the user cannot give contrary commands (like turning on at the same time the bulb and the ventilator).
- Modular and scalable architecture. The app is divided in modules, that allow for easy scaling and customizing according to the user's needs.

The hardware implementation proved that an ensemble of simple components can ensure the stable functioning of a thermal control system. The tests performed on every hardware piece that was used, confirmed their reliability, both individually and as a whole system.

The mobile app offers an intuitive interface, which is modular, scalable and easy to use.

This paper proves that the low-cost technologies can be integrated into a complete intelligent system, fully functional, able to control and monitor the real-time temperature, offering both flexibility and safety. The system can be easily extended and/or customized, according to everybody's wishes. This thing transforms the proposed solution into an excellent platform for further IoT developments, having potential for turning into complex systems.

REFERENCES

- [1] Oancea, G., Reglarea automată a proceselor industriale – ediția revizuită. Editura Universității Transilvania, Brașov, 2020.
- [2] Fraden, J., Handbook of Modern Sensors: Physics Designs and Applications (6th edition), Springer Publishing House, 2021.
- [3] Popescu, D., Iliescu, S., Instrumentație și măsurări în sisteme inteligente. Editura MatrixRom, 2019.
- [4] Dobrescu, R., Sisteme de automatizare pentru clădiri inteligente. Editura Politehnică, București, 2019.
- [5] Zhao, Y., Wen, J., Wang, S., Sensor-based fault detection in HVAC systems: Energy and Buildings. 2019.

-
- [6] Rawlings, J.B., Mayne, D.Q., Diehl, M., Model Predictive Control: Theory, Computation and Design (2nd edition), Nob Hill Publishing House, 2020.
- [7] Gubbi, J, Buyya, R., Marusic, S., Palaniswami, M., Internet of Things: A Comprehensive Guide. Wiley Publishing House, 2021.
- [8] Korodi, A., Silea, I., Senzori inteligenți și aplicații industriale. Editura Politehnică Timișoara, 2020.
- [9] <https://www.ijraset.com/research-paper/design-smart-home-control-system-based-on-android-using-iot>
- [10] Madakam, S., Lake, V., Lake, V., Internet of Things (IoT): A Literature Review, Springer Publishing House, 2020.
- [11] <https://github.com/Jayesh-JainX/IoT-Temperature-Monitoring-System-with-Mobile-App-Integration>
- [12] <https://www.hashstudioz.com/blog/iot-temperature-based-fan-speed-control-monitoring-system/>
- [13] Ionescu, C., Automatizări distribuite și control remote în sisteme industriale. Editura Universității Tehnice din Cluj Napoca, 2023.
- [14] <https://ocw.cs.pub.ro/courses/pm/prj2021/amocanu/statie-monitorizare-temperatura>
- [15] Yang, R., Newman, M.W., Learning from a Thermostat: Designing with People’s Thermal Comfort in Mind. UbiComp Conference, 2013.
- [16] Afram, A., Janabi-Sharifi, F., Theory and applications of the HVAC control systems – A review of the model predictive control (MPC), Building and Environment, 2014.
- [17] Kwak, S., Kim, J., Kim, Y., Energy-saving potential of smart thermostat. Energy and Buildings, 2015.
- [18] Buyya, R., Dastjerdi, A.V., Internet of Things, Principles and Paradigms, Morgan Kaufmann Publishing House, 2019.
- [19] <https://docs.arduino.cc/hardware/uno-rev3/>
- [20] <https://www.omo-ic.com/blog/2023-12/dht11-common-usage-problems-working-principles-and-application-scenarios.html>
- [21] <https://learn.adafruit.com/dht>
- [22] Bălănescu, M., Senzori și traductoare pentru aplicații embedded. Editura Universității Politehnica București, 2021.
- [23] <https://www.omo-ic.com/blog/2023-12/dht11-common-usage-problems-working-principles-and-application-scenarios.html>
- [24] <https://www.martyncurrey.com/tag/hc-06/>
- [25] <https://cdn.sparkfun.com/assets/9/5/f/7/b/HD44780.pdf>
- [26] https://en.wikipedia.org/wiki/IntelliJ_IDEA
- [27] <https://en.wikipedia.org/wiki/Kotlin>