Journal port Science Research
https://jpsr.uruk.edu.iq/index.php/jport
Volume 9, issue 2. 2026

Real-Time Temperature Monitoring and Automatic Control Using Embedded System

Esraa E.Raoof

Scientific Research Commission, Baghdad, Iraq

ARTICLE INFO

***Keywords**:*

Microcontrollers, Real-time temperature monitoring, Embedded systems, Cooling and heating systems

ABSTRACT

Embedded system-integrated technology for the greenhouse protects the system through self-regulation and temperature control. Sensors and a microcontroller measure the temperature and activate the heating or cooling components accordingly. These services are provided in private housing, agricultural, and industrial facilities, and are intended to enhance energy efficiency and reduce energy consumption. The temperature regulation project, developed on the Arduino platform with an integrated PID controller, is an efficient and effective temperature control tool. The system automatically operates the heating and cooling mechanisms using a PID algorithm. Bluetooth-enabled remote XY switches can be used to monitor the temperature. The system adjusts the temperature sensor settings, ensuring continuous control of the PID indicator. This is the ideal way to regulate temperature while optimizing energy consumption and improving system efficiency.

INTRODUCTION

One essential instrument for controlling temperature with little human involvement is a temperature controller this is an extremely low temperature, minimizing human contact. Errors in the data transmitted from sensors, such as thermocouples or thermometers, are detected via the heating or cooling component. This will make it easier to recharge your power supply and find a way to train and dry it. Digital temperature readings are used in homes and in winter for HVAC systems, as well as for refrigeration equipment, food processing, and specialized industries. In this work, a PID (proportional-integral-differential) Arduino related to temperature is not necessary. This type of preheating is associated with heat treatment, improving recycling processes and accelerating compensation times.

PID microcontrollers designed for industrial consumer sectors also improve performance. This technology involves the use and installation of PID temperature protection systems, the overall system performance, operational efficiency, and environmental management are improved (1).

***Figure (1)** digital temperature controller*

E-mail address:
esraa.e.raoof@src.edu.iq

Received 4th February 2026 Accepted 18th March 2026

10.36371/port.2026.2.8

Temperature measurement systems incorporate sophisticated technological sensors, leading to a greater number of temperature controllers and resulting in more precise temperature control and regulation within the systems they serve. These systems integrate microcontrollers, sensors, actuators, regulatory algorithms, and temperature measurement to achieve unified temperature control and understanding. Readings, security, and energy are all managed on a single panel for industrial consumption and production, ensuring optimal thermal protection. With a single, specialized application for dynamic regulation and management, modern regulatory strategies such as fuzzy logic, PID controllers, and machine learning algorithms can be employed. Typical applications include industrial 3D printers, automated production lines, intelligent ecosystems, and thermal management for selected electronics.The system architecture will be capable of providing specialized services such as temperature, stability, and monitoring. Regulatory algorithms like on/Off, Proportion, and PID Proportional, or PID, are selected and fine-tuned to optimize temperature regulation while minimizing undesirable transient behaviors (1). The temperature controller system has trouble getting to the temperature. The temperature controller system might not get close enough to the temperature. The temperature controller system keeps swinging around the temperature. This inaccuracy with the temperature controller system can be caused by things like the room getting hotter or colder or problems with the power supply.The temperature controller system can swing a lot around the target temperature. It can take some time for the temperature controller system to get to the target temperature after you change the temperature or something disturbs the temperature controller system. This long wait can make the temperature controller system inefficient and waste energy. It is hard to get the PID control settings right to make the temperature controller system work the way you want the temperature controller system to. The purpose of this project is to make a temperature controller system using an Arduino chip to monitor and maintain the temperature at a range. This project displays the temperature on a LCD display and via Bluetooth HC06 to determine if the cooling system is on or the heating system is, on which both contain a fan to regulate the temperature and keep it balanced.

METHODOLOGY:

The temperature regulation system uses a Proportional-Integral- controller on the Arduino platform to keep the temperature stable. This controller always checks the difference between the temperature we want and the temperature it actually is. It gets this information from a temperature sensor. The controller then sends a signal to the part that changes the temperature. This signal is based on the difference between the two temperatures.it use a formula to figure out this signal. This formula has three parts that we call integral and derivative gains. A kind of controller called a discrete PID controller, because the Arduino is digital. It checks the temperature at times and then sends a new signal to the part that changes the temperature. It had to try a few times to get the right settings for the controller so the temperature stays the same. The signal from the controller goes to the part that changes the temperature through a module. This allows the system to change the temperature by itself. Even though PID controllers are used a lot in Arduino projects this system is special because it is designed for a purpose. Most projects do not think about how heat affects the system. We did. We made sure the controller settings are just right for the temperature we want. Also made sure the system does not get too hot or cold. It does not waste energy. The system can sense the temperature control the part that changes the temperature, and change the temperature by itself. This is a system that can be used in a real greenhouse or laboratory, not just a simple project to demonstrate how it works. The temperature regulation system uses a Proportional-Integral- controller to keep the temperature stable. The temperature regulation system is special because it uses the Arduino platform and a Proportional-Integral-Derivative controller to keep the temperature stable. The Proportional-Integral-Derivative controller is a part of the temperature regulation system. The temperature regulation system is designed to keep the temperature stable in a greenhouse or laboratory. The temperature regulation system uses the Proportional-Integral- controller to keep the temperature stable and it is a complete system.

EXPERIMENTAL SETUP:

The creation of an inexpensive automatic greenhouse environmental control system is presented in this paper. Installation, software configuration, integration, maintenance, and crop control strategy implementation are among the processes covered by the system. Its foundation is a dispersed network of sensing and control nodes that uses inexpensive, low-power microcontrollers to integrate cable and wireless communications. Energy efficiency, distributed control network fault tolerance, and economical hardware design are some of the main issues that have been addressed. A multi-tier design was used to create a suite of software applications focused on growers. The system's practical implementation in typical greenhouse settings served as validation (2).Using embedded system technology; a clever, automated model for predicting air conditioning system temperature has been created and put into use. Along with monitoring the room's overall temperature, the system tracks and evaluates the temperature of objects—like people—in the space. An algorithm that can forecast the ideal room temperature was created using this data and a mathematical model. To decide whether to turn on or off the air conditioning system automatically, the program uses data from two temperature sensors.(3)For greenhouse applications, the proposed system is intended to maintain a consistent temperature at the targeted level. Temperature data are continuously monitored and contrasted with user-specified thresholds. The system automatically turns on hardware components to restore ideal conditions if the temperature rises above the threshold. By connecting the system to a personal computer over a LAN or Intranet, this research investigates the interactive integration of temperature and humidity control. Accurate temperature and humidity monitoring is crucial in industrial and research settings. Traditional configurations often rely on LCD screens and push buttons, limiting display options, ease of use, and remote access, even in highly efficient microcontroller-based systems. The proposed method overcomes these limitations, improving control efficiency and ease of use. (4).The use of commercial embedded systems to construct an automated temperature control system for a prototype greenhouse is covered in this paper. Temperature sensors are incorporated into the system to track the temperature of the air in the growing area and at the heating and cooling actuator outlets. A cascaded PID control algorithm installed on an embedded platform processes these readings and uses a Human-Machine Interface (HMI) to modify actuators and fans in order to maintain user-defined temperature set points (5).

Electronic Circuit Components:

1. Arduino UNO
2. Temperature sensitive (LM35)
3. Relay 12 volt
4. Power supply 12 volt (10 A)
5. LCD 12C 20*4
6. L298N Motor Driver
7. FAN
8. Peltier
9. PTC- Fan-Heater
10. Bluetooth (HC-06)

ArduinoUNO:

Arduino is a low-cost, user-friendly microcontroller that supports sensor integration and physical computing. It uses simple C-based programming and offers broad functionality, making it popular for various applications.(6) Based on the ATmega328P, the Arduino Uno is a small and functional microcontroller board. It runs at 16 MHz and has 6 analog inputs and 14 digital I/O pins. It has 1 KB of EEPROM, 2 KB of SRAM, and 32 KB of flash memory. Up to 40 mA per I/O pin can be delivered when powered by an external source (7V–12V) via USB. In contrast to previous variants, the Uno replaces the FTDI chip with an ATmega8U2/16U2 chip for USB-to-serial conversion. It is perfect for projects that can be expanded because it still works with a lot of shields that are currently in use (7).

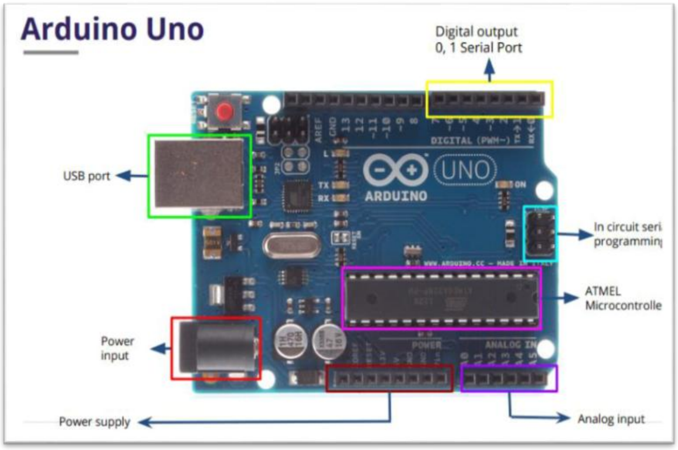

Figure (2) *Arduino UNO*

Temperature sensitive (LM35):

The LM35 is a high-precision temperature sensor that outputs a voltage linearly proportional to degrees Celsius (˚C).

- Operating Voltage: 4V to 30V
- Current Consumption: ~60 µA
- Accuracy: ±0.5˚C at 25˚C
- Output Scale: 10 mV per ˚C
- Temperature Range: –55˚C to +150˚C
- Low Self-Heating: Less than 0.1˚C
- No External Calibration Required

The sensor performs best under stable temperature and humidity conditions. It has several variants such as LM35C, LM35CA, and LM35D, all based on the same working principle but differing in accuracy and operating temperature ranges.(8)

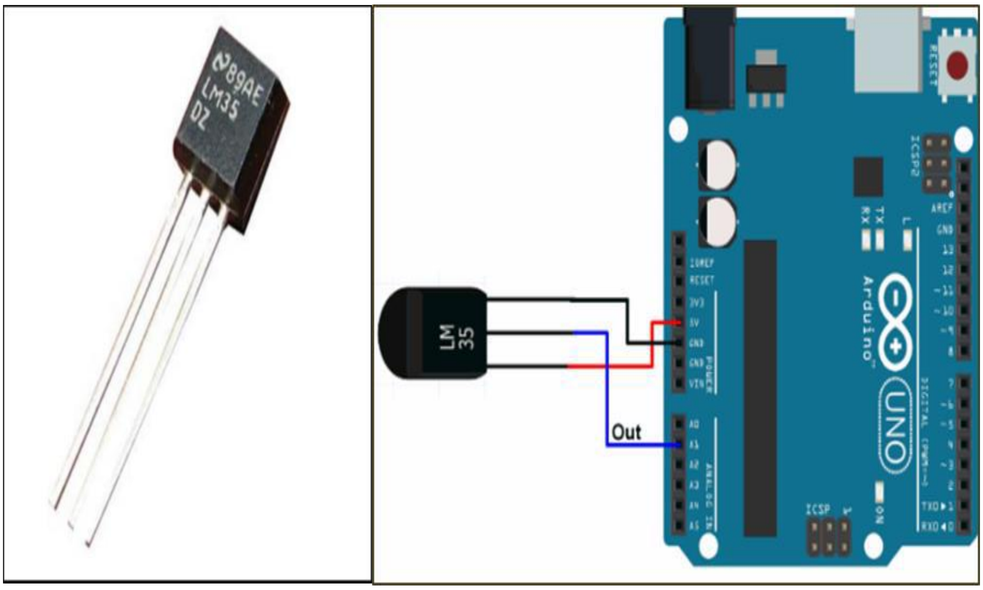

Figure (3) *views of LM35 sensor and LM35 sensor connected with UNO Arduino*

Relay 12 volt :

A relay is a simple electronic component made of a coil and a switch operated by a magnetic field. It allows a small electric current to control a larger current. Relays were used in early electronic devices and computers before microprocessors existed. Today, they are still widely used in vehicles and machines to control current, voltage, and sequential operations.(9)

Figure (4) *Relay*

To power an Arduino, a power supply transforms electrical voltage; typically, this voltage is 5 volts, but 12 volts can be used sparingly. With its 4-bit mode, which connects with fewer pins, contrast control, and backlight for improved visibility, a 16-pin LCD operates on 5 volts.

L298N Motor Driver:

Using TTL logic signals, the L298 driver is a high-voltage, high-current dual full-bridge driver made to regulate inductive loads such as relays, DC motors, and stepper motors. Its outputs are set up as bridges to drive loads in either differential or common mode, and it has two enable inputs for separate control. It has a built-in 5V regulator, motor direction LEDs, Schottky diodes for protection, screw terminals for simple connections, and support for up to 46V and 4A.(10)

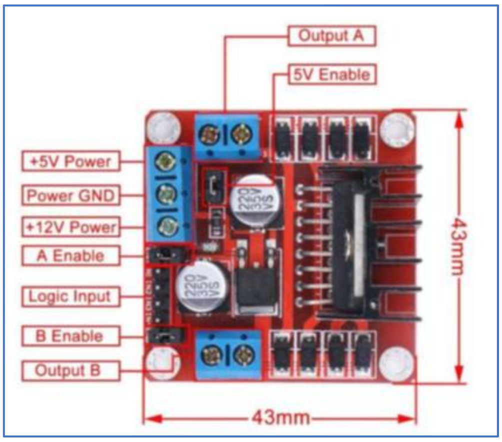

Figure (5) *L298N Motor Driver pins*

FAN, Peltier, PTC- Fan-Heater, Bluetooth (HC-06):

Fan: A turbo machine that overcomes flow resistance by creating a pressure differential, which transforms energy into gas flow—typically air—through a system. Peltier Module: A thermoelectric device that operates at 15.4 V and 15 A and uses the Peltier effect to deliver heating or cooling; it needs a heat sink and thermal paste for protection.(12)
PTC Heaters: Superior to conventional heaters in terms of performance and safety, these cutting-edge ceramic-based heaters provide safe, energy-efficient heating in fin or honeycomb shapes.(11)
HC-06 Bluetooth Module: Usually used as a slave device for wireless serial communication with smartphones, this short-range wireless Bluetooth SPP module has a data throughput of 3 Mbps at 2.4 GHz.

Design and Implementation:

After we know all hardware components, we connect all components together shown in figure below

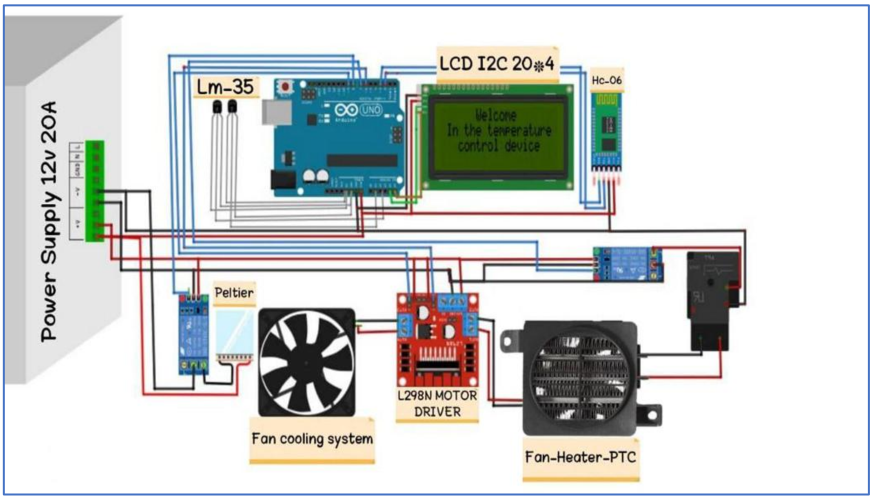

Figure (6) *project implementation*

Code (called sketches) can be written, edited, and uploaded to Arduino boards using the Arduino IDE. It has a terminal, toolbar, message area, and text editor. ".ino" is the extension used to save sketches. The entire system was put together after each component had been properly programmed and tested. All components function perfectly. The cooling system (fan, heat sink, and Peltier element) activates once the temperature exceeds a preset value (defined via HC-06 Bluetooth technology). The heating system (fan and heating element) activates when the temperature falls below the preset value. The measured temperature is converted to Celsius using a specific formula.

$$T\_avg = (T1+T2)/2 \times (100)/Vref$$

Where:

T1: the temperature reading from the first sensor

T2: the temperature reading from the second sensor

T_avg: Average temperature in Celsius

V_ref: Reference voltage

The sensor readings T1T_1T1 and T2T_2T2 are obtained as **ADC counts** from the Arduino board (0–1023), and Vref is the reference voltage of the board (typically 5V). To find the temperature in Celsius we take the average of the two readings. Then we convert that average to voltage. After that we use the sensors details to convert the voltage to temperature like the equation, above shows. This makes it clear how to turn ADC readings into the temperature.

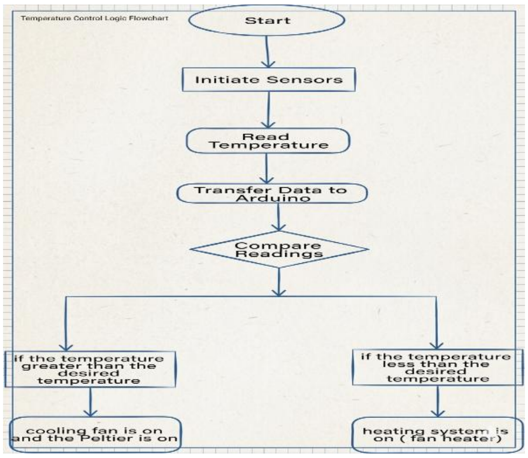

Figure (7) *Flow chart*

Results and Discussion

This system is crucial in this field due to its precise temperature control, which improves processes. By reducing manual intervention, automation through proportional-integral-differential (PID) control contributes to increased process consistency and efficiency. With a nearly linear response to heat and a shift of 0.5°C each read, the temperature sensor showed quick response and low power usage. It confirmed high sensitivity by rapidly cooling down after going from low to high temperatures in 22 seconds. Following individual testing and integration, the entire system operated without a hitch. Based on variations from the target temperature, the controller precisely turned on the heating or cooling system. The sensor sent real-time data to the display. Now that the testing phase is done and each and every component has been programmed, connected, and tested individually. It's time to connect the system's components together and upload the final code, and see the system fully operating. All the components in the system were working together simultaneously, and no component was interfering with any other component. The temperature sensor was sending the value, which was displayed on the screen. Every time the value would be abnormal, the controller would activate the suitable system (cooling fan or heater fan). Now that the testing phase is done and each and every component has been programmed, connected, and tested individually. It's time to connect the system's components together and upload the final code, and see the system fully operating. All the components in the system were working together simultaneously, and no component was interfering with any other component. The temperature sensor was sending the value, which was displayed on the screen. Every time the value would be abnormal, the controller would activate the suitable system (cooling fan or heater fan)

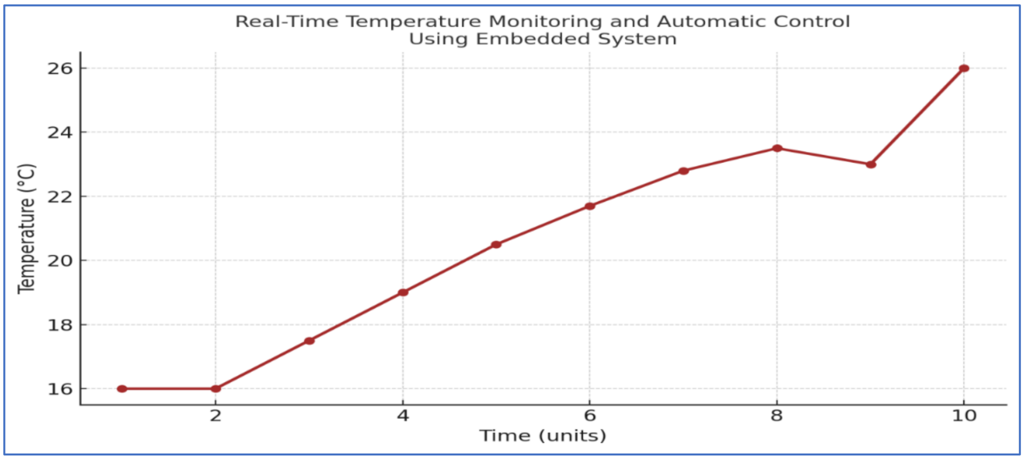

Figure (8) *Flow chart*

The graph illustrates the process of real-time temperature monitoring using an embedded system equipped with automatic control. As shown, the temperature remains stable at 16°C during the first two time units, indicating an initial state of equilibrium. The temperature fluctuations between 8 °C and 23.5 °C resulted from a single reaction triggered either by radiation or the activation of hot components. In unit 9, the heating element is exposed to a very low temperature, while in unit10, a temperature of 26 °C emerges from one of the columns. These abrupt reactions can be caused by system interactions with a heat source or one of the disturbance systems. This data underscores the importance of integrated systems on a larger scale, where the temperature is automatically controlled and adjusted over a defined period.

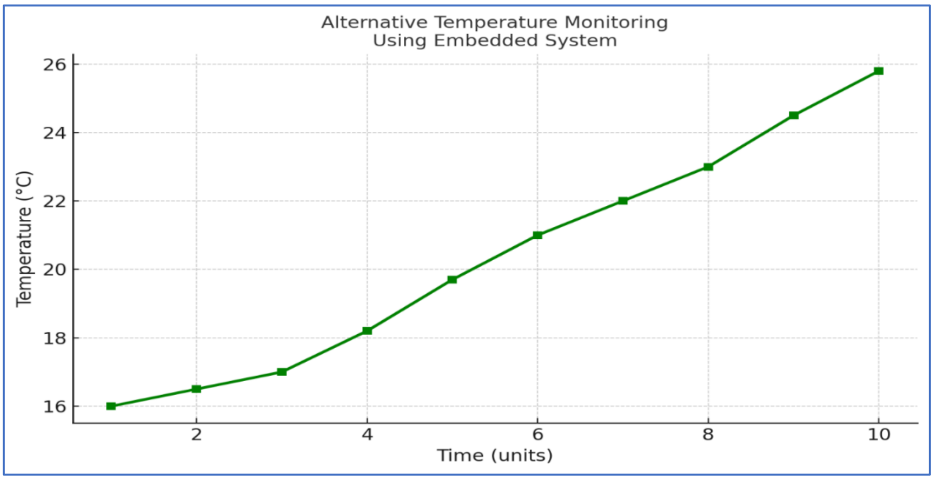

Figure (9) *Alternative Temperature Monitoring*

The graph shows a steady increase in temperature over time. Starting at 16°C in the first time period, the temperature rises continuously over ten time periods, eventually reaching approximately 26°C. This smooth, gradual curve indicates a controlled heating process, likely driven by the integrated system. The absence of sudden fluctuations or drops suggests stable environmental conditions and efficient sensor performance. Overall, the graph reflects the system's effectiveness in continuous temperature monitoring and supports its use in various scenarios.

Conclusions:

The new temperature control system, based on the Arduino platform and utilizing PID technology, optimizes temperature control. The system continuously monitors cooling and heating settings using PID algorithms, resulting in optimal cooling settings. This study has demonstrated the effectiveness of dynamic and efficient temperature control systems, offering a significant advantage in improving heating systems and environmental management. This system features rapid response to high temperatures, enhancing stability and optimization. The configuration, incorporating a flexible cooling module, provides flexible protection and variable temperature settings. Ultimately, this study highlights the effectiveness of dynamic and efficient temperature control systems.

REFERENCES

Ekundayo, J. (2011). *Temperature control system* (Doctoral dissertation, Babcock University).

Sahu, K., & Mazumdar, S. G. (2012). Digitally greenhouse monitoring and controlling of system based on embedded system. International Journal of Scientific & Engineering Research, 3(1), 41–48.

Shamrat, F. M. J. M., Allayear, S. M., Alam, M. F., Jabiullah, M. I., & Ahmed, R. (2019). A smart embedded system model for the AC automation with temperature prediction. In International Conference on Advances in Computing and Data Sciences (pp. 343–355). Springer.

Baruah, S. (2011). Designing and implementing a computer interactive network embedded system for monitoring and controlling of humidity and temperature for using in a greenhouse. In Proceedings of the International Conference & Workshop on Emerging Trends in Technology (pp. 1247–1250).

Alsharakhat, A., Almoalem, M., AlHawaj, N., Papageorgiou, C., & Sadriwala, A. (2017). Temperature control of a greenhouse system using embedded systems technology. In 2017 9th IEEE-GCC Conference and Exhibition (GCCCE) (pp. 1–9). IEEE.

Louis, L. (2016). Working principle of Arduino and using it as a tool for study. International Journal of Control, Automation, Communications and Systems, 1.

Ahmed, S. M., Marhoon, H. M., & Nuri, O. (2018). Implementation of smart anti-theft car security system based on GSM. International Journal of Engineering & Technology, 7(4), 5261–5265.

National Semiconductor (TI). (n.d.). LM35 precision centigrade temperature sensors [Datasheet]. Texas Instruments. https://www.ti.com/lit/ds/symlink/lm35.pdf.

Tjandi, Y., & Kasim, S. (2019). Electric control equipment based on Arduino relay. Journal of Physics: Conference Series, 1244(1), 012028. IOP Publishing. https://doi.org/10.1088/1742-6596/1244/1/012028.

Rahman, Z. S. A., & Hussain, F. S. A. (2017). Smart incubator based on PID controller. International Research Journal of Engineering & Technology, 4(3), 2501–2509.

Shin, Y. H., Ahn, S. K., & Kim, S. C. (2016). Performance characteristics of PTC elements for an electric vehicle heating system. Energies, 9(10), 813. https://doi.org/10.3390/en9100813.

Rambang, K., Fitriawan, H. F., Arinto, F. X., & Murdika, U. (2021). Design of cooling and heating tool using thermoelectric Peltier based on Arduino Uno. ELKHA: Jurnal Teknik Elektro, 13(1), 41–47.