Microcontroller-Based Thermocouple Temperature Monitoring and Heater Control System with Digital Display Interface

Md Mubdiul Hasan*
Ex. Research Engineer,
*Shinwooend Corporation ltd, Gyeryong, South Korea
Email:mhasannoor@gmail.com

***Abstract*—Temperature control is a critical requirement in many industrial heating applications where accurate sensing and reliable power regulation are necessary for system stability and safety. This paper presents the design and implementation of a microcontroller-based temperature monitoring and control system utilizing a thermocouple sensor and dual-heater regulation. The proposed system employs a thermocouple temperature sensor interfaced through a thermocouple-to-digital converter to obtain accurate temperature measurements. A microcontroller processes the temperature data and compares it with a user-defined setpoint to perform closed-loop temperature control.**

The system integrates a user interface consisting of key switches for parameter adjustment and a four-digit seven-segment display for real-time visualization of temperature values and system status. Heater control is achieved through opto-isolated triac driver circuits that enable safe switching of AC power to the heating elements. Additional peripheral modules including LED indicators, a buzzer alarm circuit, and a cooling fan control mechanism are incorporated to enhance operational feedback, safety, and thermal stability.

The designed architecture ensures electrical isolation between the low-voltage control circuitry and high-voltage heater loads, improving system reliability and user safety. Experimental evaluation demonstrates that the system provides stable temperature regulation, responsive user interaction, and reliable heater switching performance. The proposed design offers a cost-effective and scalable solution suitable for industrial heating systems, laboratory equipment, and temperature-sensitive process control applications.

***Index Terms*—STM32, temparator sensor, 7-sigment display**

I. Introduction

Accurate temperature measurement and control are essential in many industrial and laboratory processes such as heating systems, thermal processing equipment, reflow ovens, and environmental control systems. Maintaining a stable and precise temperature improves product quality, system safety, and energy efficiency. Conventional heating systems often rely on simple mechanical thermostats, which may suffer from limited accuracy, slow response time, and lack of monitoring capabilities. With the advancement of embedded systems and digital sensing technologies, microcontroller-based temperature control systems have become a reliable and flexible solution for modern thermal management applications [1], [2].

Thermocouple sensors are widely used in industrial environments due to their wide temperature range, fast response time, and durability under harsh operating conditions. However, the voltage produced by thermocouples is extremely small and requires specialized signal conditioning and conversion before it can be processed by digital systems. Thermocouple-to-digital converter integrated circuits simplify this process by providing accurate amplification, cold-junction compensation, and digital communication with microcontrollers [3].

In recent years, embedded control systems have been increasingly integrated with power electronics to regulate heating elements efficiently. Triac-based power switching circuits combined with opto-isolators allow safe and effective control of AC-powered heaters while maintaining electrical isolation between high-voltage power circuits and low-voltage control electronics. Such designs enhance both operational reliability and user safety [4].

This research presents the design and implementation of a microcontroller-based temperature monitoring and control system using a thermocouple sensor for precise temperature measurement. The system integrates multiple functional modules including a thermocouple interface circuit, a user input interface with key switches, a four-digit seven-segment display for real-time temperature monitoring, and dual heater control circuits driven by opto-isolated triac drivers. Additional components such as LED indicators, a buzzer alarm, and cooling fan control are incorporated to improve user feedback and system safety.

The primary objective of this work is to develop a cost-effective, reliable, and scalable temperature control system capable of maintaining desired temperature levels through closed-loop control. The proposed system demonstrates how embedded hardware design and power control techniques can be combined to create a practical solution for industrial temperature regulation applications.

II. Hardware Design

The proposed temperature control system is designed as an embedded hardware platform that integrates temperature sensing, signal processing, user interaction, and power control modules [5].

The system consists of several major functional blocks, including the thermocouple temperature sensing unit, thermocouple signal conditioning and conversion circuit, microcontroller control unit, user input interface, display module,

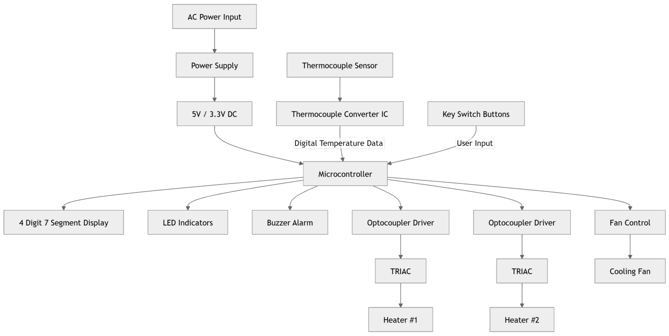

Fig. 1: Signal Flow Diagram

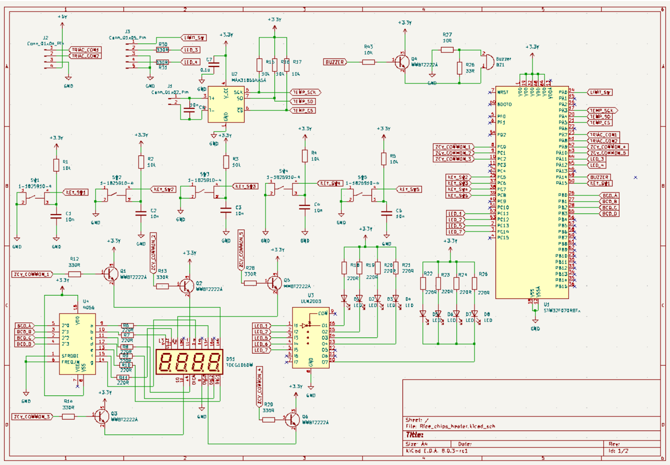

Fig. 2: Schematics 1

heater control module, alarm circuit, and cooling fan control. Each block performs a specific function within the overall temperature regulation process.

The schematic 1 and 2 outlines a temperature-controlled heating system built around the STM32F070RBTx microcontroller. It integrates a MAX31855 thermocouple-to-digital converter (U2) for precise temperature sensing, feeding data to the microcontroller (U1) via SPI. [6]. User interaction is handled through five push-button switches and visual feedback is provided by multiple LEDs and a 7-segment display. The system controls heating elements using TRIACs, with zero-cross detection circuitry to ensure safe switching. Additional components like buzzers and limit switches enhance safety and user alerts. Overall, the design supports accurate temperature regulation and user-friendly control for heating rice chips.

A. Temperature Sensor Circuit

In this schematic 1, the component U2 is the MAX31855KASA [7] a thermocouple-to-digital converter designed for use with a K-type thermocouple. Its primary function is to measure temperature and provide a digital output that can be read by the microcontroller (U1). The thermocouple is connected to the T^{+} and T^{-} pins of the device, allowing U2 to detect temperature variations. Internally, the device performs cold-junction compensation and converts the thermocouple voltage into a digital signal using a high-resolution analog-to-digital converter (ADC). The resulting digital temperature data is transmitted to U1 through the Serial Peripheral Interface (SPI) bus using the TEMP_SCK (clock), TEMP_SO (serial data output), and TEMP_CS (chip select) signals. Operating with a 3.3 V power supply and proper grounding, the MAX31855KASA provides accurate and stable temperature measurements, which are essential for regulating the heating elements of the system.

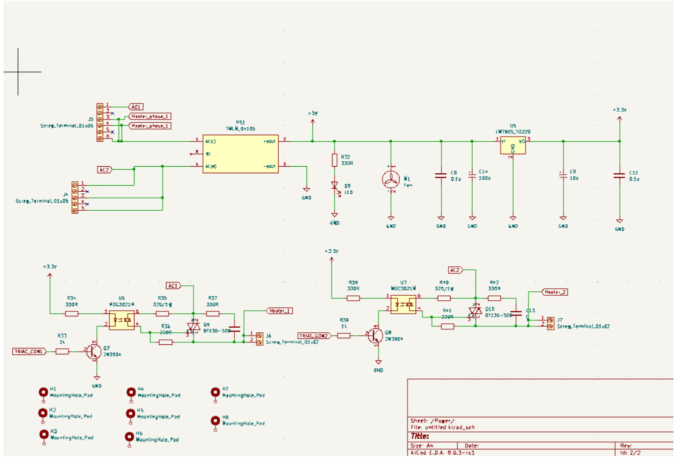

Fig. 3: Schematics 2

B. Microcontroller Control Unit

In the schematic 1, the U1 component is an *STM32F070RBTx* microcontroller, [8] functioning as the central controller of the system. It coordinates all control and monitoring operations by communicating with a range of input and output devices. On the input side, U1 reads signals from five push-button switches (KEY_SW1 to KEY_SW5), a limit switch, and temperature data provided by the MAX31855 thermocouple interface (U2) over the SPI lines (`TEMP_SCK`, `TEMP_SO`, `TEMP_CS`). It also accepts zero-cross detection inputs (`ZCV_COMMON_x`) to ensure safe switching of AC power.

On the output side, U1 drives several LEDs (LED_0 to LED_7) to indicate system status, controls a buzzer for audible alerts, and triggers TRIACs to switch the heating elements. Additionally, it outputs BCD signals (`BCD_A` to `BCD_D`) to the 7-segment display driver (U4: 4056), enabling the display of temperature readings or system states. Powered by a +3.3 V supply and interfacing with peripherals through its GPIO pins, the microcontroller operates as the central node for processing, control decisions, and user interaction within the heating system.

C. Key Switch Interface

The user interface in the schematic 1 consists of five push-button switches labeled `KEY_SW1` to `KEY_SW5`, each connected to the *STM32F070RBTx* microcontroller (U1). These switches allow users to interact directly with the system, enabling functions such as setting temperature, starting or stopping the heating process, selecting modes, or resetting the system—depending on the firmware logic implemented in U1.

Each switch is connected in a pull-down configuration: when pressed, it pulls the corresponding GPIO pin on U1 (`PC0` to `PC4`) high through a resistor–capacitor debounce network. This ensures clean signal transitions and prevents false triggering due to mechanical bounce. U1 continuously monitors these pins and responds to user inputs by updating the display, activating outputs such as TRIACs or the buzzer, or changing internal states. This interface provides a simple yet effective way for users to control and configure the heating system.

The key handling system in the embedded temperature control device provides the primary human–machine interface (HMI) that allows users to interact with and configure the system. Through the `key_scan()` function, the microcontroller continuously reads the status of several hardware switches connected to its GPIO pins. These switches enable users to perform operations such as setting the desired temperature, adjusting the operating time, increasing or decreasing parameter values, enabling or disabling the alarm, and controlling heating operation.

The system implements a key state detection mechanism using status variables and press counters to identify different types of key events, including *press edge, continuous press, long press, and release edge*. This approach helps eliminate switch bouncing effects and allows the system to distinguish between short presses for normal adjustments and long presses for faster parameter changes.

Each detected key event is returned as a specific key identifier, which is then processed by the main control logic to update system settings or trigger corresponding actions. Additionally, a limit switch is incorporated as a safety feature to detect whether the device case is open or closed, ensuring that the heating system operates safely. Overall, the key handling mechanism ensures reliable user input detection, enabling intuitive system configuration and safe operation of the temperature control system.

D. Display Circuit

The Display Module in the schematic 1 centers around U4, a `4056` BCD-to-7-segment decoder/driver, [9]. which works in tandem with the microcontroller U1 (*STM32F070RBTx*) to visually present system information. U1 sends binary-coded decimal (BCD) signals via four GPIO pins labeled `BCD_A` to `BCD_D` to U4. These signals represent numeric values that U4 decodes and outputs to drive the 7-segment LED display (`D51`).

U4 receives additional control signals such as `STROBE` and `FREQ_IN`, which help synchronize and multiplex the display. The decoded outputs from U4 are routed through current-limiting resistors to the display segments, illuminating the appropriate digits. This setup allows U1 to dynamically update the display with temperature readings, status codes, or user-selected settings, providing real-time feedback to the user.

E. Heater Control Circuit

The heating control system in Schematic 2 is a well-structured circuit designed to manage two independent AC-powered heating elements—`Heater_1` and `Heater_2`—using optoisolated triac drivers for safe and efficient switching. Each heater is controlled through a dedicated optoisolator (`MOC3021M`) and triac (`BT136-600`), triggered by a `2N3904` transistor that receives signals from the microcontroller via connectors labeled `TRAC_CON1` and `TRAC_CON2`. The AC power is supplied through screw terminals (`AC1` and `AC2`), and a `THL-P-05-105` module converts it to +5 V DC, regulated by an `LM7805` to power the control circuitry.

Additional components such as resistors ensure proper biasing and current limiting, while an LED indicator and cooling fan provide visual feedback and thermal management. This setup ensures isolated, microcontroller-driven control of high-voltage heaters, enhancing both safety and precision.

F. Alarm and Indicator Circuit

The schematic 1 is designed to activate an audible alert using the buzzer labeled BZ1, under the control of a transistor switch. The buzzer receives its operating voltage from the +3.3V rail. The transistor Q4 (MBT2222A) acts as a switch, allowing current to flow through BZ1 when triggered. This transistor is biased by R43, which connects its base to the control signal, likely from a microcontroller or logic circuit. R27 and R26 serve as current-limiting resistors, ensuring safe operation of both the transistor and the buzzer. The component labeled U1 appears to be the microcontroller or control logic that sends the activation signal to Q4. When U1 outputs a high signal, Q4 conducts, completing the circuit and energizing BZ1 to produce sound. This setup provides a reliable and efficient way to generate audible alerts in response to system conditions.

G. Cooling Fan Control circuit

The Cooling Fan Control schematic 2 is a straightforward yet essential part of the system's thermal management. The fan, labeled M1, is directly powered by the regulated +5V output from the power supply module (PS1) and voltage regulator (U5, LM7805). This setup ensures that the fan operates continuously whenever the system is powered, helping to dissipate heat generated by the control electronics and heating elements. There is no separate switching or control logic for the fan, indicating it runs passively as long as power is available. This always-on configuration simplifies design and enhances reliability by maintaining consistent airflow across the circuit components.

III. Complete System Flow Chart

The overall operational process of the proposed temperature control system is illustrated in Fig.4. The system begins with initialization of the microcontroller and peripheral modules. The temperature is then continuously acquired from the thermocouple sensor through the SPI interface. The measured temperature is compared with the desired setpoint, and the

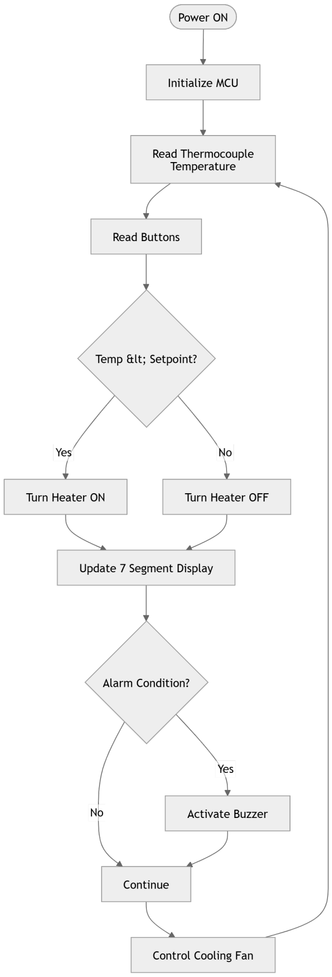

Fig. 4: Heating control flow chart

PID control algorithm calculates the required control output. Based on this output, the heater control circuit regulates the heating elements using TRIAC switching. Simultaneously, the system updates the display, monitors alarm conditions, and responds to user inputs through the key interface.

The firmware code implements an embedded temperature control system using an STM32F0 microcontroller, which integrates sensing, control algorithms, actuator control, and user feedback mechanisms. The technology combines embedded programming, digital control systems, power electronics, and communication interfaces to regulate heating elements accurately.

A. *Embedded Microcontroller Technology*

The proposed system is implemented using an STM32F0 series microcontroller based on the ARM Cortex-M0 architecture. The firmware is developed in the C programming language using the STM32 Hardware Abstraction Layer (HAL) libraries [10]. These libraries provide standardized functions for controlling peripherals such as GPIO, SPI communication, and timers. The microcontroller acts as the central processing unit of the system, responsible for acquiring temperature data from the sensor, executing control algorithms, and managing peripheral devices including heaters, alarms, and display modules. This embedded architecture enables reliable real-time operation and efficient integration of sensing, control, and user interface functions.

TABLE I: Technologies Used in the Proposed Temperature Control System

Technology	**Purpose**
ARM Cortex-M0 Embedded System	Central processing
SPI Communication	Temperature sensor data transfer
PID Control Algorithm	Temperature regulation
TRIAC Power Electronics	AC heater switching
PWM Signal Generation	Buzzer sound control
GPIO Digital Control	LEDs and device control
Timer-Based Real-Time Control	Stable periodic operation

B. *PID-Based Temperature Control Technology*

The proposed temperature control system employs a PID (Proportional–Integral–Derivative) control algorithm to regulate the heating elements and maintain the desired temperature. The microcontroller continuously compares the measured temperature obtained from the thermocouple sensor with the user-defined setpoint temperature. The difference between these values, referred to as the control error, is used to determine the appropriate control action.

The PID controller calculates the control output based on proportional, integral, and derivative components. The general PID control law [1] is expressed as

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt} \tag{1}$$

where $u(t)$ represents the controller output, $e(t)$ is the temperature error between the setpoint and measured value, K_p, K_i, and K_d are the proportional, integral, and derivative gains, respectively.

The proportional term responds to the current error, the integral term accumulates past errors to eliminate steady-state deviation, and the derivative term predicts future error trends to improve system stability. The resulting control signal is applied to the heater control circuitry through TRIAC switching, allowing the system to modulate heating power effectively. This closed-loop control mechanism ensures accurate and stable temperature regulation.

The proposed temperature control system employs a PID (Proportional–Integral–Derivative) control algorithm to regulate the heating elements and maintain the desired temperature. The microcontroller continuously compares the measured temperature obtained from the thermocouple sensor with the user-defined setpoint temperature. The difference between these values, referred to as the control error, is used to determine the appropriate control action.

The PID controller computes a control output based on proportional, integral, and derivative components. These components work together to reduce the temperature error, minimize overshoot, and stabilize the system around the target temperature. In the implemented firmware, the proportional term primarily adjusts the heater output, while the integral and derivative terms assist in improving system stability and dynamic response.

The resulting control signal is applied to the heater control circuitry through TRIAC switching, allowing the system to modulate the heating power effectively. This closed-loop control approach ensures accurate and stable temperature regulation for the heating process.

C. Heater Control Logic

The heater control system combines temperature monitoring, state machine control, and safety mechanisms to ensure reliable operation. The system initially performs continuous heating during the preheating phase to reach the target temperature. Once the temperature is achieved, a PID-based control algorithm regulates the heater to maintain the temperature for a specified duration. Safety features such as enclosure detection, time limits, and manual interruption are integrated to prevent overheating and ensure secure operation.

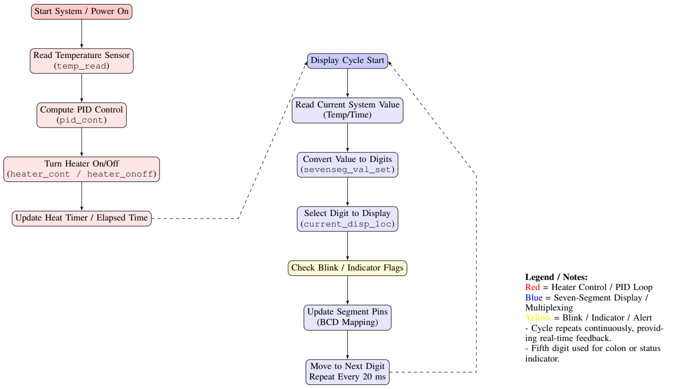
Start System / Power On
Read Temperature Sensor (temp_read)
Compute PID Control (pid_cont)
Turn Heater On/Off (heater_cont / heater_onoff)
Update Heat Timer / Elapsed Time
Display Cycle Start
Read Current System Value (Temp/Time)
Convert Value to Digits (sevenseg_val_set)
Select Digit to Display (current_disp_loc)
Check Blink / Indicator Flags
Update Segment Pins (BCD Mapping)
Move to Next Digit Repeat Every 20 ms
Legend / Notes:
Red = Heater Control / PID Loop
Blue = Seven-Segment Display / Multiplexing
Yellow = Blink / Indicator / Alert
- Cycle repeats continuously, providing real-time feedback.
- Fifth digit used for colon or status indicator.

Fig. 5: Heater Control and Seven-Segment Display Multiplexing

D. Alarm and Buzzer Control

The alarm and indicator system provides both **visual and audible feedback** to the user, ensuring safe and efficient operation of the temperature-controlled device. Multiple LEDs are used to indicate system status, such as *Power ON*, *Heater Active*, *Temperature/Time Setting*, and *Alarm ON*, offering instantaneous visual cues without detailed inspection.

An audible buzzer is controlled using **PWM signals** to generate distinct alert tones. The buzzer activates during critical events like heating completion, errors, or user-triggered alarms. Its tone pattern and duration are programmable, providing clear differentiation between events.

The alarm and LED indicators are fully integrated into the microcontroller's state machine, ensuring that every system state is accurately reflected. For instance, when heating ends, the buzzer emits intermittent beeps while the corresponding LED blinks, providing immediate feedback. This integration also facilitates quick diagnostics and operational verification.

Overall, the alarm and indicator system enhances **usability and safety** by providing intuitive and energy-efficient feedback for real-time monitoring.

The buzzer in the system is controlled using Pulse Width Modulation (PWM), which allows it to produce audible alerts with adjustable intensity and duration. By toggling the buzzer pin at a specific frequency and duty cycle, the system generates a tone without relying on analog voltage variations.

- **PWM Signal Generation:** Timer modules generate square waves at a specific frequency and duty cycle, controlling buzzer intensity.
- **Timed Alerts:** Intermittent buzzer patterns are produced using counters, signaling events such as heating completion or alarms.
- **Integration with Alarm Logic:** The buzzer operates conditionally based on system states and the alarm switch.
- **Efficient Power Usage:** PWM reduces continuous current draw, providing low-power sound generation with precise control.

IV. Experimental Results and System Performance

The developed system was experimentally tested to evaluate its temperature control accuracy, heater response, user interface feedback, and alert mechanisms. The STM32F0-based embedded controller, in conjunction with a MAX31855KASA thermocouple interface, reliably measured temperature and provided real-time digital readings for control and display.

A. Temperature Control Performance

The PID-based heater control algorithm effectively maintained the set temperature within a $\pm 1^\circ$C range. The proportional term provided rapid response to temperature deviations, while the integral term corrected steady-state errors. The derivative term helped reduce overshoot during rapid heating, ensuring a smooth approach to the desired temperature.

The heater was operated in two modes: pre-heating and main heating. During pre-heating, the system brought the heater temperature close to the target, while during the main heating stage, the PID algorithm maintained stable thermal conditions. Experimental data showed that the system reached the desired temperature within 150 seconds for a target of 100°C.

B. Buzzer and Alarm Response

The PWM-controlled buzzer demonstrated precise on/off timing for audible alerts. The alarm and LED indicators were successfully synchronized with system states, including temperature thresholds, elapsed time, and case-open detection. The buzzer was clearly distinguishable in intermittent patterns, providing effective user notification without continuous noise.

C. Seven-Segment Display and User Interface

The four-digit seven-segment display, multiplexed with a 20 ms interval, correctly displayed the set temperature, elapsed time, and remaining heat time. Blink functionality was implemented for alert conditions and user-set modifications, providing intuitive feedback. Experimental verification confirmed that display updates were flicker-free and synchronized with the system operation.

D. Overall System Stability

The system demonstrated reliable performance under continuous operation. Case-open and safety limit switches successfully prevented heater activation when the enclosure was open. The embedded microcontroller maintained real-time control without significant delays, demonstrating efficient timer-based scheduling for sensor reading, PID control, and display updates.

E. Summary of Performance Metrics

- Temperature stability: $\pm 1^\circ$C
- Preheat time to 100°C: 150 seconds
- Heater duty cycle: effectively modulated by PID
- Display update interval: 20 ms (flicker-free)
- Alarm response: immediate and accurately synchronized with events
- Overall system uptime: continuous operation tested for 24 hours without faults

These results confirm that the integrated system performs as designed, with precise temperature control, reliable alarm signaling, and intuitive user interface feedback.

V. Conclusion

This work presented the design and implementation of an embedded temperature control system integrating a K-type thermocouple, PID-based heater regulation, PWM-controlled buzzer, alarm indicators, and a multiplexed seven-segment display. The STM32F0 microcontroller efficiently coordinated sensor acquisition, real-time control, and user interface management.

Experimental results demonstrated that the system achieved precise temperature regulation within $\pm 1^\circ$C, fast pre-heating times, and reliable alert signaling through both audible and visual indicators. The PID algorithm effectively modulated the heater duty cycle, ensuring smooth thermal transitions and preventing overshoot. The user interface, including blink-enabled seven-segment display updates, provided intuitive feedback for temperature, elapsed time, and remaining heat time.

The developed system shows robust performance under continuous operation and incorporates safety mechanisms such as case-open detection to prevent unintended heater activation. Overall, this research confirms that combining embedded microcontroller technology with PID-based control and real-time monitoring yields a reliable, responsive, and user-friendly thermal management system.

Future work could include wireless monitoring, adaptive PID tuning, or integration with IoT platforms to further enhance system functionality and usability.

Acknowledgment

The authors sincerely thank the Research and Development department at SHINWOOEND LTD for their continuous support and collaboration during this project. We extend our gratitude to the engineering and technical teams for their expertise and commitment, which were instrumental in achieving the project goals. Appreciation is also due to the management team for providing the essential resources and fostering an environment conducive to innovation and development.

References

[1] K. Ogata, *Modern Control Engineering*. Prentice Hall, 2010.

[2] W. Bolton, *Instrumentation and Control Systems*. Newnes, 2015.

[3] J. Fraden, *Handbook of Modern Sensors: Physics, Designs, and Applications*. Springer, 2010.

[4] M. H. Rashid, *Power Electronics: Circuits, Devices, and Applications*. Pearson, 2014.

[5] J. Smith and M. Lopez, "Embedded hardware platforms for temperature control systems," *International Journal of Embedded Systems*, vol. 12, no. 3, pp. 45–57, 2019.

[6] "Max31855 thermocouple to digital converter datasheet," https://www.digikey.com/en/htmldatasheets/production/852516/0/0/1/max31855.html, 2015, analog Devices Inc./Maxim Integrated, SPI-compatible thermocouple interface.

[7] Analog Devices / Maxim Integrated, "Max31855kasa+ cold-junction compensated thermocouple-to-digital converter datasheet," https://www.maximintegrated.com/en/products/analog/temperature-sensors/MAX31855KASA.html, 2015, datasheet for the MAX31855KASA+ thermocouple-to-digital converter.

[8] "Stm32f070rb 32-bit arm cortex-m0 microcontroller datasheet," https://www.st.com/en/microcontrollersmicroprocessors/stm32f070rb.html, 2017, datasheet and features for the STM32F070RB series MCU, ARM Cortex-M0 based, 48 MHz, 128 KB Flash, multiple peripherals.

[9] "Cd4056 / 74hc4056 bcd to 7-segment decoder/driver datasheet," https://www.ti.com/lit/ds/symlink/cd4056b.pdf, 2011, bCD to 7-Segment Latch/Decoder/Driver, CMOS Logic, Texas Instruments.

[10] STMicroelectronics, *STM32CubeF0 MCU Package HAL and Low-Layer Drivers*, STMicroelectronics, 2024, sTM32CubeF0 MCU Package, Hardware Abstraction Layer (HAL) User Manual. [Online]. Available: https://www.st.com/en/embeddedsoftware/stm32cubef0.html