IMPLEMENTATION OF AN INTELLIGENT ROASTING MACHINE USING A SOFTWARE-BASED PID ALGORITHM AND MICROCONTROLLER

Jival, A.* and Oriolowo N.Z.

Department of Electrical and Electronics Engineering, Modibbo Adama University, Yola

*Correspondent email: jivalangeti@gmail.com

ABSTRACT

The demand for precision and automation in temperature-sensitive processes such as roasting has led to the development of intelligent control systems. Traditional methods that rely on thermostats and relays often suffer from inaccuracies, mechanical wear, and limited adaptability. This project addresses these limitations by implementing a microcontroller-based roasting machine equipped with a software-based PID (Proportional-Integral-Derivative) control algorithm to achieve accurate and consistent temperature regulation. The system is built around the AT89S52 microcontroller, which interfaces with essential hardware components including a temperature sensor (LM35), an analog-to-digital converter (ADC), a solid-state relay, and a user input/output interface consisting of a keyboard and an LCD display. The PID control logic was implemented entirely in software using the AT89S52 instruction set. The methodology involved designing both the hardware and software components, integrating them into a complete closed-loop control system. Experimental tests were conducted in both open-loop and closed-loop configurations to evaluate system performance. The results showed that the software-based PID controller was highly effective in maintaining the desired roasting temperature, with steady-state errors ranging from 0.1°C to 0.6°C. These outcomes confirm the reliability, accuracy, and responsiveness of the intelligent control system, demonstrating its suitability for applications where precise temperature management is essential. The system demonstrated reliable temperature regulation, accurately tracking set points with minimal steady-state error.

Keywords: Intelligent Roasting, Microcontroller, PID Algorithm, Temperature Control, AT89S52

1 INTRODUCTION

Modern electrical devices such as ovens, incubators, and roasting machines require precise internal temperature regulation to operate efficiently within a specified range [1]. These appliances serve important functions in households, hospitals, and food processing industries. An oven, for instance, is a closed chamber used for processes like heating, baking, or drying [2]. In hospitals, ovens are essential for sterilizing medical tools, while in homes they are commonly used for cooking, baking, and roasting. In the food industry, maintaining the correct temperature is especially vital for activities like culturing bacteria and other microorganisms [3]. For effective performance, the internal temperature of such devices must align with user-defined preferences. Traditionally, temperature regulation in such systems has relied on thermostats or electromechanical relays. Relays act as electrical switches but are prone to degradation due to spark generation during switching, which can eventually cause system malfunctions. Thermostats operate using bi-metallic strips that respond to temperature changes based on the thermal expansion of materials. However, the physical properties of these materials, such as their thermal expansion coefficient, tend to change over time, which reduces the thermostat's accuracy and consistency [4]. In today's advanced applications, ovens are expected to do more than maintain a constant temperature—they must also follow predefined temperature schedules and offer better usability. For example, in a supermarket setting, an oven may be required to begin roasting meat at high temperatures and later lower the heat automatically to keep the contents warm [5]. Additionally, the ability to evenly distribute heat within the oven chamber has become a key performance expectation. These new demands introduce complex control challenges that conventional devices like relays and thermostats are no longer sufficient to handle [6]. To meet these new requirements, a more sophisticated control approach is needed—one that delivers greater precision, reliability, flexibility, and ease of use. A top-down design strategy is a suitable method, where individual functional units are developed separately and later integrated into a complete system. This study focuses on the design and implementation of a microcontroller-based temperature control system for a chicken roasting machine using a software-based PID algorithm. The PID controller integrates three components: the proportional part speeds up the correction of temperature errors, the integral component ensures that steady-state errors are eliminated, and the derivative part functions as a damping mechanism to prevent temperature overshoot [7].

2 SYSTEM DESCRIPTION

Figure 1 illustrated the complete system consists of a microcontroller, a keyboard, a display unit, a heating element, actuators, and an analog-to-digital converter (ADC). It is built around the AT89S52 microcontroller, which serves as the central control unit. When the system is turned on, the microcontroller activates the heater using a solid-state relay. Temperature inside the oven is detected by an LM35 sensor, which converts the heat level into a small analog voltage signal. Because this signal is too weak for the ADC to process directly, an LM4558 amplifier is used to boost the voltage [8]. The amplified signal is then passed to the ADC, which converts it into a digital value that the microcontroller can read. The microcontroller compares this actual temperature value with the user-defined target temperature [9]. If the actual temperature is lower than the desired value, the microcontroller sends a signal to the solid-state relay to increase power to the heater, raising the oven's temperature. If the actual temperature is higher than the target, the microcontroller reduces the heater's power, lowering the temperature. Users can input the desired temperature using the keyboard. Both the actual and set point temperatures are displayed in real time on a Liquid Crystal Display (LCD), allowing easy monitoring and control.

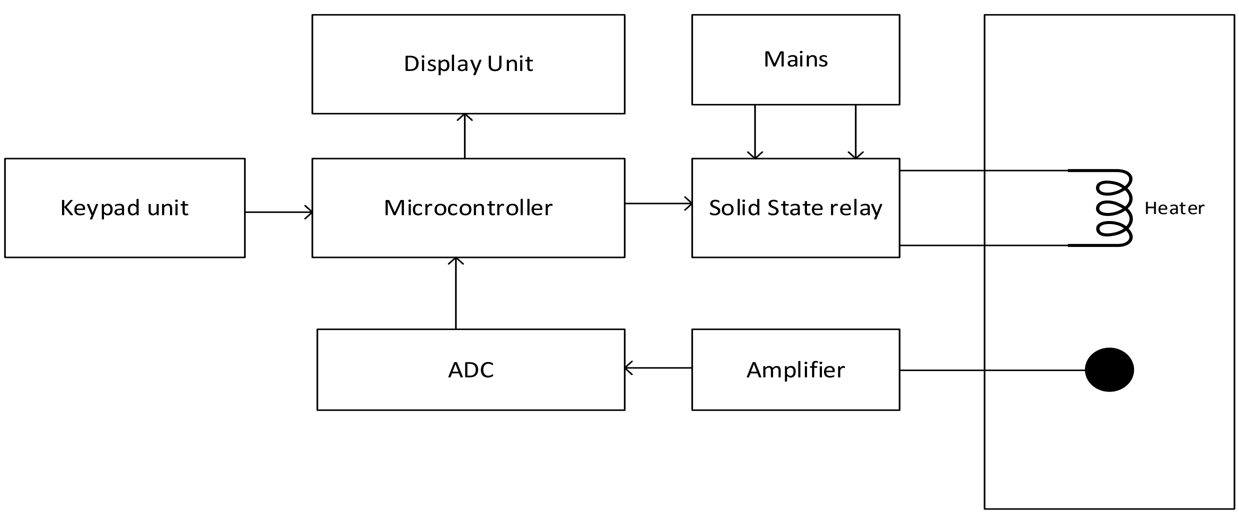

Figure 1. Complete System Block Diagram

3 DESIGN PROCEDURE

This research project was implemented in two primary stages: hardware development and software programming. The hardware stage entailed designing and assembling the different modules that constitute the system, with each functional block constructed and tested accordingly. The software component involved writing the control algorithm using assembly language, which was used to coordinate and manage the system's operations.

3.1 Power Supply Unit

The system generates a 5V DC power supply using an LM317 voltage regulator. According to the block diagram illustrated in Figure 2 above, a step-down transformer is first used to convert the 220V AC mains supply to 15V AC. This alternating voltage is then rectified into direct current (DC) and passed through a filter to smooth out fluctuations. The filtered voltage is subsequently regulated by the LM317 to produce a stable 5V DC output suitable for powering the circuit.

Figure 2. Block diagram of power supply unit

3.2 Solid state Relay and Microcontroller Units

This section of the system controls, the heater through pin 3.6 of the microcontroller illustrated in Figure 3. When the microcontroller sends a logic low ('0') to this pin, the heater is turned off; when a logic high ('1') is sent, the heater is activated. A resistor connected to the base of an NPN transistor is used to properly bias the transistor, which in turn functions as a switch for the solid-state relay. The microcontroller used, the AT89S52, is a low-power, high-performance 8-bit CMOS microcontroller featuring 8KB of in-system programmable flash memory [10]. Manufactured using Atmel's high-density non-volatile memory technology, it is fully compatible with the industry-standard 80C51 instruction set and pin configuration.

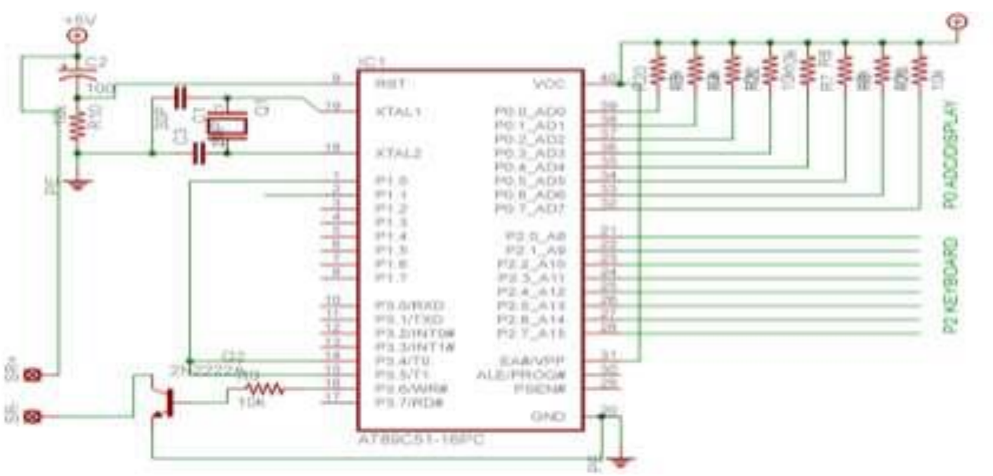

Figure 3**.** Microcontroller Schematic Diagram with Solid State Relay

3.3 Analog to Digital Converter Unit

The Analog-to-Digital Converter (ADC0804) operates using only a reference resistor and clock capacitors, as recommended by the manufacturer. A thermistor, which detects the oven's internal temperature, is connected to the ADC through a 5kΩ variable resistor. The ADC receives the sensed temperature as an analog voltage and converts it into a digital signal (in binary form) that the microcontroller can interpret. This digital output is then transmitted to the microcontroller through port zero. Figure 4 shows Analog to Digital Converter in cooperated with an Amplifier.

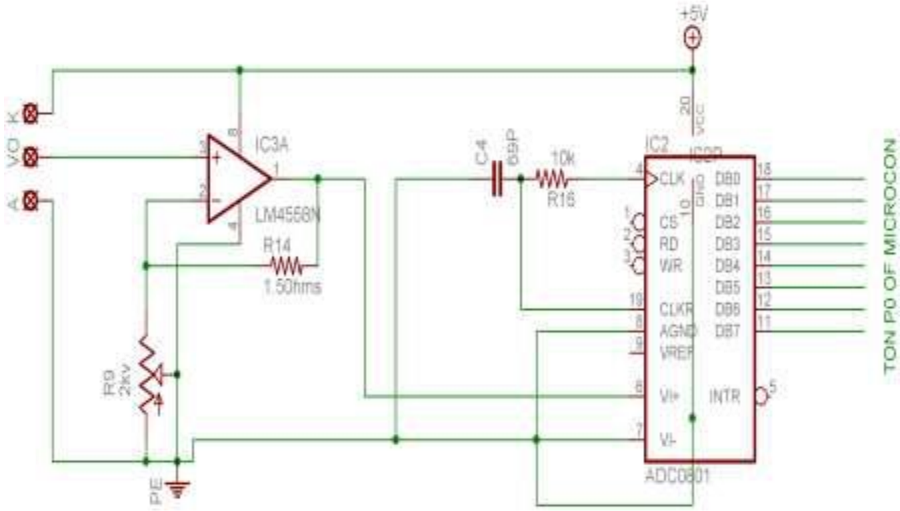

Figure 4. Analog to Digital Converter in cooperated with an Amplifier

4 PROCESS MODEL

The experimentally obtained step – response is shown in Figure 6: the process has a dead time and it is damped. Therefore, the step – response can be fitted into a simple first order model with dead time [10].

$$G(s) = \frac{K\ell^{-\theta S}}{\tau s + 1} \quad (1)$$

Where K is the process gain, □ is the death time, □ is the time constant. Referring to the graph in Figure 6:

Therefore:

$$G(s) = \frac{0.07\ell^{-17S}}{1090s + 1} \quad (2)$$

5 SOFTWARE-BASED PID ALGORITHM

A software-based PID (Proportional-Integral-Derivative) control algorithm is a digital implementation of the classical PID controller, used to regulate process variables such as temperature, speed, or pressure. Instead of using analog components, this algorithm is programmed into a microcontroller, allowing it to perform real-time control decisions based on sensor feedback [11]. In this system, the PID controller is used to maintain a desired oven temperature. The microcontroller continuously reads the current temperature (measured via a sensor and converted to digital form by the ADC), compares it with the user-defined set point, and calculates the error [12]. The PID algorithm then computes a control output based on the following three components: Proportional (P): Reacts to the present error. It adjusts the output in proportion to the difference between the desired and actual temperature. Integral (I): Accounts for the accumulation of past errors over time. It helps eliminate steady-state error by integrating the error signal. Derivative (D): Predicts future error based on the rate of change of the error. It dampens the system response to reduce overshoot and oscillation. The control output is calculated using the formula [13].

$$\text{Output} = K_p . e(t) + K_i . \int e(t)dt + K_d . \frac{de(t)}{dt} \quad (3)$$

Where: $e(t)$ = error = set point − actual temperature, $K_{p,} K_{i,} K_d$ = tuning constants for proportional, integral, and derivative terms. In the microcontroller, this equation is implemented in discrete

form due to the digital nature of the system. The calculated output is then used to adjust the heating element via a solid-state relay [14]. This approach provides accurate, real-time control, and its flexibility allows for easy adjustment of tuning parameters to suit different operating conditions or control objectives.
The transfer function of the PID controller is given as in [15]:

$$G(s) = K_p + \frac{K_i}{S} + K_d S \tag{3}$$

From Ziegler and Nichols PID controller is given as:

$$K_p = \frac{1.2\tau}{K\theta} \tag{4}$$

$$K_i = \frac{1}{2\theta} \tag{5}$$

$$K_d = 0.5\theta \tag{6}$$

Using the above data's obtained from the graph: K_p, $K_{i,}$ and K_d becomes. $K_p = 1099.15$, $K_i = 0.03$, $K_d = 8.5$. By experimental tuning the values of the PID above become: $K_p = 200$ $K_i = 0.25$ $K_d = 50$. In digital mode PID controller Becomes:

$$D_z = \frac{U_z}{Ez} = K_p + \frac{K_i Ts}{2}\left[\frac{Z+1}{Z-1}\right]E_z + \frac{K_d}{Ts}\left[\frac{Z-1}{Z}\right] \tag{7}$$

$$U_Z = K_P E_Z + \frac{K_i Ts}{2}\left[\frac{Z+1}{Z-1}\right]E_z + \frac{K_d}{Ts}\left[\frac{Z-1}{Z}\right]E_Z \tag{8}$$

$$U_Z + [z-1]U_z + K_p E_z - K_p[z-1]E_z + \frac{K_i Ts}{2}E_z + \frac{K_i Ts}{2}[z-1]E_z + \frac{K_d}{Ts}[z-2]E_z \tag{9}$$

Take the inverse z-transform of the equation above, we have the following:

$$U(k) = U(k-1) + K_p E(k) - K_p E(k-1) + \frac{K_i Ts}{2}E(k) + \frac{K_i Ts}{2}E(k-1) + \frac{2K_d}{Ts}E(k-1) + \frac{K_d}{Ts}E(k-2) \tag{10}$$

Substituting the values of PID in the equation.

$$U(k) = U(k-1) + 200E(k) - 200E(k-1) + \frac{0.25\times 10}{2}E(k) + \frac{0.25\times 10}{2}E(k-1) + \frac{50}{10}E(k) - \frac{2\times 50}{10}E(k-1) + \frac{50}{10}E(k-2) \tag{11}$$

$$\mathrm{U(k)} = \mathrm{U(k-1)} + 206E(k) - 209E(k-1) + 5E(k-2) \tag{12}$$

The control equation can be re-written in Hexadecimal as:

$$UH = U(k-1)H + (CE)HE(k) - (DI)HE(k-1) + 5E(k-2) \tag{13}$$

Data to be loaded into the microcontroller is calculated as follow:

$$\mathrm{DL} = \frac{Ts}{\frac{V^2}{R}}\left[\frac{10000}{10}\right]U \tag{14}$$

Where: $Ts = 10s$ Sample time. $\frac{V^2}{R} = 3500W$ Power of the Heater.

$$DL = \frac{10}{3500}\left[\frac{10000}{10}\right]U \tag{15}$$

$$DL = 3H \tag{16}$$

6 EXPERIMENTAL RESULTS

6.1 Open loop Test

Temperature readings from the oven were recorded using a MAS-34X digital multimeter connected to a PC. The device was set to monitor and log temperature data over a 30-minute duration, capturing values at 10-second intervals. The resulting data, which reflects the oven's temperature behavior without the control system, was used to generate the plot shown in Figure 5. The graph in fig 6 illustrates that the oven temperature rises steadily over time, eventually stabilizing at approximately 270°C.

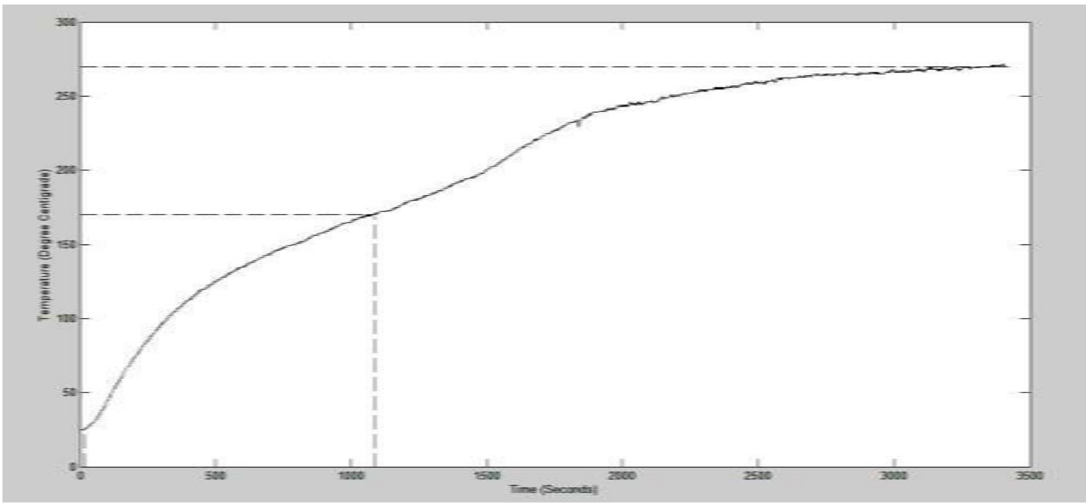

Figure 5. Graph of Temperature against Time

6.2 Close Loop Test

The system's ability to control temperature was evaluated by setting temperature values within the range of 30°C to 140°C. These set points were then compared with the corresponding measured steady-state temperatures, as presented in the Table 1.

Table 1. Close loop results with the steady state Error

S/N	Desired Temperature (^0C)	Actual Temperature Measured (^0C)	Steady State Error (^0C)
1	30	30.40	0.4
2	40	40.40	0.4
3	50	50.50	0.5
4	60	60.00	0.0
5	70	70.00	0.0
6	80	80.60	0.6
7	90	90.30	0.3
8	100	100.0	0.0
9	110	110.0	0.0
10	120	120.2	0.2
11	130	130.1	0.1
12	140	140.5	0.5

7 DISCUSSION

The experimental results from the oven temperature measurements reveal a clear distinction between open-loop and closed-loop behavior. Without the control system, the oven's temperature increased steadily and eventually plateaued at around 270°C, a typical response in systems lacking feedback regulation. However, after implementing a software-based PID control algorithm on the AT89S52 microcontroller, the system demonstrated effective temperature management. It was able to maintain user-defined set points within the 30°C to 140°C range with minimal steady-state errors between 0.1°C and 0.6°C. This confirms the system's reliability and the capability of the PID controller to manage temperature dynamics accurately. When compared with related works, such as Zhang et al. (2022), who used an STM32-based PID control system with ±0.5°C precision, the performance of this AT89S52-based system is competitive. Despite its simpler architecture and lower processing power, the AT89S52 microcontroller—when programmed efficiently—delivered comparable accuracy. This demonstrates that cost-effective 8-bit microcontrollers can still be highly effective in precision thermal control applications, provided the control algorithm is well-designed and properly tuned. The overall system setup is shown in Figures 6 and 7.

Figure 6**.** Front view of a roasting machine

Figure 7**.** system circuitry

8 CONCLUSIONS

This research, successfully developed a microcontroller-based roasting machine using a software-implemented PID control algorithm on the AT89S52. The system demonstrated reliable temperature regulation, accurately tracking set points with minimal steady-state error. Experimental tests confirmed its effectiveness, making it a cost-effective and efficient solution for automated roasting applications. The design is suitable for small-scale industries and can be further improved with features like data logging and wireless control.

REFERENCES

1. Singh, Y.K., Kumar, J., Pandey, K.K., Rohitha, K. and Bhargrav (2016). Temperature control system and its control using PID controller. *International Journal of Engineering Research and Technology*, vol. 4, pp. 4-6.
2. David, M., Anyebe, A.O. and Adegboyega, M.A. (2013). A microcontroller based electric

   cooker/oven with temperature and time control for the developing countries. *International Journal of Research and Applications*, vol. 3, no. 3, pp. 1082-1084.

3. Fatima, M.S. (2014). Comparison of PID controller tuning techniques for temperature process. *International Journal of Innovative Research in Electronics, Instrumentation and Control Engineering*, vol. 2, no. 8, pp. 2-5.

4. Ezema, C.N. and Umezinwa, C.B. (2016). Design and implementation of a microcontroller based digital time aware oven temperature controller. *International Journal of Multidisciplinary Research Module of Education*, vol. 2, no. 1, pp. 731-739.

5. Nisha, R. and Madhusoodanan, K.N. (2014). PC based PID temperature controller. *International Journal of Engineering and Advanced Technology Research*, vol. 2, no. 4, pp. 16-24.

6. Xie, D., Chang, X., Shu, X., Li, L., Wang, J. and Mei, L. (2015). High precision temperature control system for an oven-controlled crystal oscillator. *Open Automation and Control Systems Journal*, vol. 7, pp. 1690-1697.

7. Sharma, B.B., Banshwar, A., Pathak, M., Sharma, N.K. and Joshi, A. (2021). Microcontroller-based PID temperature controller using genetic algorithms. In *Adv. in Renewable Energy and Sustainable Environment*, L. Dewan *et al.*, Eds., vol. 667, Singapore: Springer, 2021, pp. 277–283. doi:10.1007/978-981-15-5313-4_26.

8. Turkey, Y. (2021). Fuzzy logic-based household oven design," *Electron. Lett. Sci. Eng.*, vol. 17, no. 2, pp. 165-175.

9. Chu, C.W., Zhu, Z.C., Bian, H.T. and Jiang, J.C. (2021). Design of self-heating test platform for sulfide corrosion and oxidation based on fuzzy PID temperature control system. *J. Test Platform Design*, vol. *, no.1 pp 543-543.

10. A novel hierarchical temperature control method of an electric oven and its experimental evaluation. *Robotica & Management* (Emerald Insight), 2022. (Hierarchical multi-region PI controllers implemented on Arduino Nano), pp. 44-52.

11. Zhang, J. X. *et al.*, (2022). Design of PID temperature control system based on STM32. *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 322, 2022, Art. no. 072020.

12. Nguyen, T.A.V. *et al.*, (2023). ID control for heating oven by Matlab-embedded Arduino," *Robotica & Management*, vol. 28, no. 1, pp. 36-42.

13. Borase, R.P. *et al.* (2022). Thermal investigation of a thermoelectric cooler based on Arduino and PID control approach. *Case Stud. Therm. Eng., (PID-Arduino for precise thermal control)*, vol. 36, pp. 102-249.

14. Yuwono, S., Rana, M.U., Schwung, D. and Schwung, A. (2025). Real time self-tuning adaptive controllers on temperature control loops using event-based game theory. arXiv*, Jun. 2025. (Self-learning PID gain adjustment applied to industrial temperature loop)*.

15. Chowdhury, M.A. and Lu, Q. (2022). A novel entropy-maximizing TD3-based reinforcement learning for automatic PID tuning. arXiv*, Oct. 2022. (RL-based automated PID tuning for second-order temperature systems)*.