IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 8, August 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.625

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

PID Controller for Temperature Control System

Pritee Pawar, Tanuja Patil, Anushka More, Mohini Saredy (HOD)

Department of Electronics and Telecommunication, AISSMS IOIT, PUNE, India

ABSTRACT: This project aims to create and implement a Proportional integral-Derivative (PID) controller that can precisely manage temperature. The PID controller uses an Arduino Uno microcontroller in conjunction with a TB660 motor driver to dynamically modify the system's output in response to variations between the setpoint temperature and the measured temperature. By continuously monitoring the temperature, the PID algorithm improves the accuracy of temperature management by fine-tuning the output to minimize the error between the intended setpoint and the current temperature. The goal of this configuration is to attain precise temperature control, which is essential for several uses, from scientific research to industrial operations. Combining the PID controller with the TB660 motor driver and Arduino Uno provides a stable platform for obtaining ideal temperature control, which should increase the dependability and efficiency of temperature-sensitive systems.

I. INTRODUCTION

Many industrial and commercial processes depend on temperature regulation, which affects safety regulations, operational effectiveness, and product quality. For results in a variety of industries, including chemical manufacture, food preservation, and laboratory research, it is essential to maintain accurate and steady temperatures. Proportional-Integral-Derivative (PID) controllers are a fundamental solution for temperature management among the many techniques employed because of its dependability and flexibility in changing conditions. The creation and use of a temperature management system based on PID controllers is the main goal of this research project. Improving temperature control in various settings and sectors of the economy is the goal. PID controllers, TB660 motor operators, LM35 temperature sensor units, and pulse width modulation (PWM) signals are important parts that work together to form a complete system. Considering a designed temperature control system's performance, usefulness, and practicality is the major objective. In particular, the study intends to assess the efficiency of PID controllers in preserving temperature stability, examine the precision and dependability of LM35 temperature sensing units, comprehend the function of TB660 motor drivers in controlling heating or cooling mechanisms, and investigate the application of PWM signals in modifying power outputs to heating or cooling elements. This research attempts to improve temperature control technology by providing insights to optimize temperature control systems through methodical experimentation and analysis. The results of the study will be useful to the biotechnology, food processing, medical instruments, and manufacturing sectors because accurate temperature management is essential to process optimization and product quality assurance.

II. LITERATURE SURVEY

The study of using the Arduino Uno to create PID control algorithms for temperature control applications offers a thorough understanding of both the theoretical and practical elements. In the articles or papers listed below, an effective method for utilizing PID algorithms on Arduino to maintain accurate temperature control was presented. PID temperature control system setup is made easier with the help of step-by-step tutorials and guidelines that come with circuit schematics and sample code. These resources demonstrate the use of PWM signals to precisely and smoothly manage heating equipment and the use of Ziegler Nichols and trial-and-error techniques for manually modifying PID parameters based on system response.

A literature review on PID controllers for temperature control using an LM35 temperature sensor, TB6600 motor driver, and PWM for heat control would typically cover several key areas.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Sr.No	Paper	Methodology	Remark
1	A. Saravanan and S. S. Jagtap, "Temperature Control System Using Arduino and PID Algorithm," International Journal of Control Theory and Applications, vol. 9, no. 14, pp. 6515-6521, 2019.	Implementation of a PID algorithm for temperature control using Arduino.	Practical approach with a focus on real-world applications.
2	Circuit Digest, "PID Temperature Controller Using Arduino"	Step-by-step guide to build a PID-based temperature control system using Arduino.	Practical tutorial with circuit diagrams, code, and explanations.
3	M. H. Rashid, "Power Electronics: Circuits, Devices & Applications," Pearson, 4th edition, 2013	Explains the implementation of PID control using PWM for temperature regulation.	Detailed explanations of power electronics principles applied to PID control.
4	C. K. Dey et al., "Tuning of PID Controller using Ziegler-Nichols and Genetic Algorithm for Temperature Control," International Journal of Engineering Trends and Technology (IJETT), vol. 69, no. 7, pp. 156-160, 2021	Compares Ziegler-Nichols method and Genetic Algorithm for PID tuning.	Provides a comparative analysis, showing the effectiveness of different tuning methods.
5	C. A. Smith and A. B. Corripio, "Principles and Practice of Automatic Process Control," John Wiley & Sons, 3rd edition, 2006.	Describes the trial and error method for tuning PID controllers.	Provides practical examples and step-by-step guidance for manual tuning.
6	H. P. Huang and C. C. Wang, "Introduction to Feedback Control," Springer, 2015.	Covers the basics of PID control and tuning, including trial and error methods.	Comprehensive introduction suitable for beginners in control systems.

III. RESEARCH AND OBJECTIVES

1. How to Install a Temperature Control System Automatically: Create a PID (Proportional-Integral-Derivative) controller to automatically control a system's temperature.
2. Make sure the PID controller minimizes variations and changes in temperature by adjusting heating elements to maintain the temperature at a user-defined setpoint.
3. In order to improve control performance, optimize PID parameters: To find the ideal PID parameters (proportional, integral, and derivative gains) for the temperature control system, apply the Ziegler-Nichols tuning m e t h o d and the trial-and-error (fine-tuning) tuning approach.
4. To ensure a steady and responsive temperature, fine-tune the PID parameters to reduce temperature overshoot, oscillations, and settling time.

IV. METHOD AND MATERIALS

PID controller for controlling the temperature using Zigler Nicolo's method with Arduino uno tb660 motor delivery lm 35 temp gives all methodology Here is a step-by-step methodology for implementing a PID controller to control temperature using the Ziegler-Nichols method with an Arduino Uno, TB6600 motor driver, and LM35 temperature sensor:

➢ Assemble the following Components: ○ Arduino Uno board ○ TB6600 motor driver ○ LM35 temperature sensor ○ Power supply for the motor driver and heating element ○ Heating element

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

➢ Setup Hardware: Hardware setup: Join the Arduino Uno and the LM35 temperature sensor. Three pins are usually present on an LM35: VCC, GND, and OUT. Attach OUT to an analog input pin (such as A0), GND to GND on the Arduino, and VCC to 5V on the Arduino. Link the Arduino Uno and TB6600 motor driver together. Follow the datasheet or guidelines provided with the TB6600 for wiring instructions. Connect the necessary power source and the heating element to the motor driver.

➢ Install Arduino IDE: Download and install the Arduino IDE on your computer if you haven't already.

➢ Write the Arduino Code: Launch the Arduino IDE, then open up a blank sketch. Add the libraries that are required for the PID controller and LM35 sensor. If the PID library isn't already included in the Arduino IDE, you might have to install it. Set the PID controller's initial values for the setpoint temperature, the LM35 sensor's input pin, the motor driver's output pin, and the PID constants (Kp, Ki, and Kd). Utilize the Ziegler-Nichols approach to configure the PID controller. The Ziegler-Nichols method's ultimate gain and ultimate period are used to adjust the PID constants in this way. Utilizing analog Read(), retrieve the temperature from the LM35 sensor and use the sensor's formula to convert it to Celsius. Based on the variation between the setpoint and real temperatures, use the PID controller to determine the output control signal for the motor driver. To modify the heating element and keep the temperature at the desired level, send the motor driver a control signal.

➢ Upload and Test the Code: Start by using USB to connect your Arduino Uno to your PC. Launch the Arduino IDE and compile the PID controller's given code. Connect the Arduino board to the compiled code. Once uploaded, track the temperature measurements obtained from the LM35 temperature sensor and observe the TB6600 motor driver's control signal generated from the Arduino. Make sure that the temperature readings are steady and within the intended range to confirm that the PID controller is operating as intended. Optimizing temperature control performance may require adjusting the derivative, integral, and proportional terms.

➢ Adjust and Fine-Tune: To guarantee precise temperature readings, adjust the LM35 temperature sensor as necessary. This can need modifying the code's calibration factor to reflect the properties of the sensor. Adjust the PID controller's constants to get the required temperature control characteristics and real-world performance. To get the ideal balance between responsiveness and stability, this technique might need to be tested and adjusted repeatedly.

➢ Safety considerations: When handling electrical and thermal components, put safety first. Make sure that enough insulation and safety precautions are in place to avoid risks like electrical mishaps or overheating. Keep a close eye on the system while it's operating, and be ready to take action if something unexpected happens. This can entail lowering control parameters or turning off the machine entirely to stop harm or dangerous situations.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

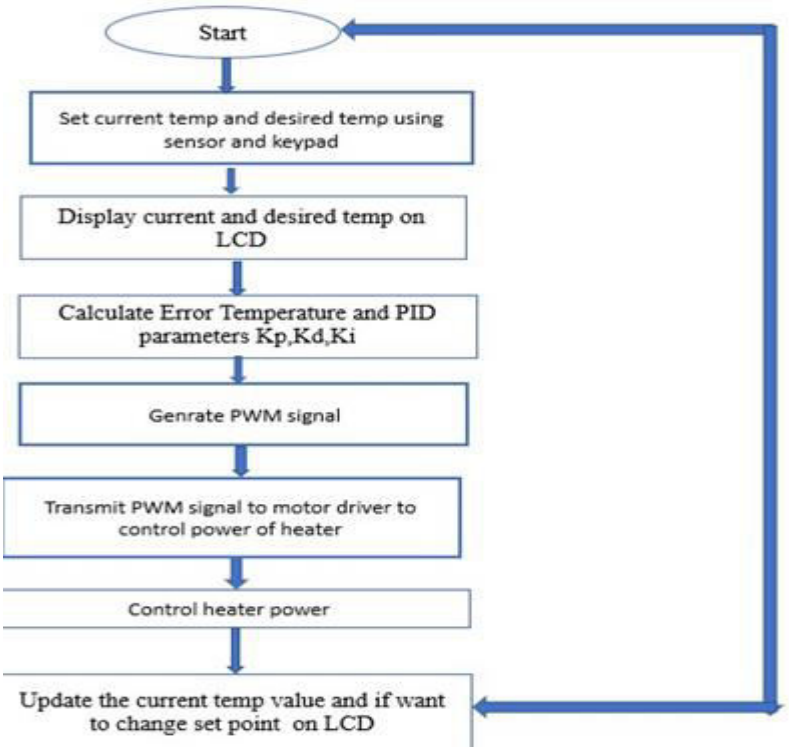
International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

V. WORKFLOW

The following describes a step-by-step process for integrating an Arduino Uno, TB6600 motor driver, and LM35 temperature sensor with a PID controller to regulate temperature using the Ziegler-Nichols method:

Start :The system starts up and gets ready to read inputs from the user and sensor data.



Decide on the desired and current temperatures.
Current Temperature: The temperature at this moment is determined by the LM35 sensor.
Desired Temperature: Using a keypad, the user enters the desired setpoint temperature.

➢ Determine the PID parameters and error temperature.

➢ Calculating Errors: It computes the difference between the actual temperature and the desired temperature.

➢ PID Settings: Based on the error, the PID algorithm's proportional (P), integral (I), and derivative (D) components are calculated. Each component's formula is as follows:

$P = Kp \times \text{error}$

$I = I + (Ki \times \text{error} \times \Delta t)$

$D = Kd \times (\text{error} - \text{previous error} / \Delta t)$

➢ PID Output: The control output is obtained by adding the P, I, and D components.

Output is equal to P + I + D.

➢ Produce a PWM Signal

A PWM signal is produced using the PID algorithm's control output. The power supplied to the heater is modulated by this PWM signal.

➢ Send PWM Signal to Motor Driver

A PWM signal is created and transmitted to the motor driver. The PWM signal is used by the motor driver to regulate the heater's power supply.

➢ Regulate the Power of the Heater

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

To control the temperature, the motor driver modifies the heater's power. To reduce temperature inaccuracy, the heater power is adjusted either way.

➢ Modify the Value of the Current Temperature

The LM35 sensor is used by the system to continuously update the current temperature measurement.

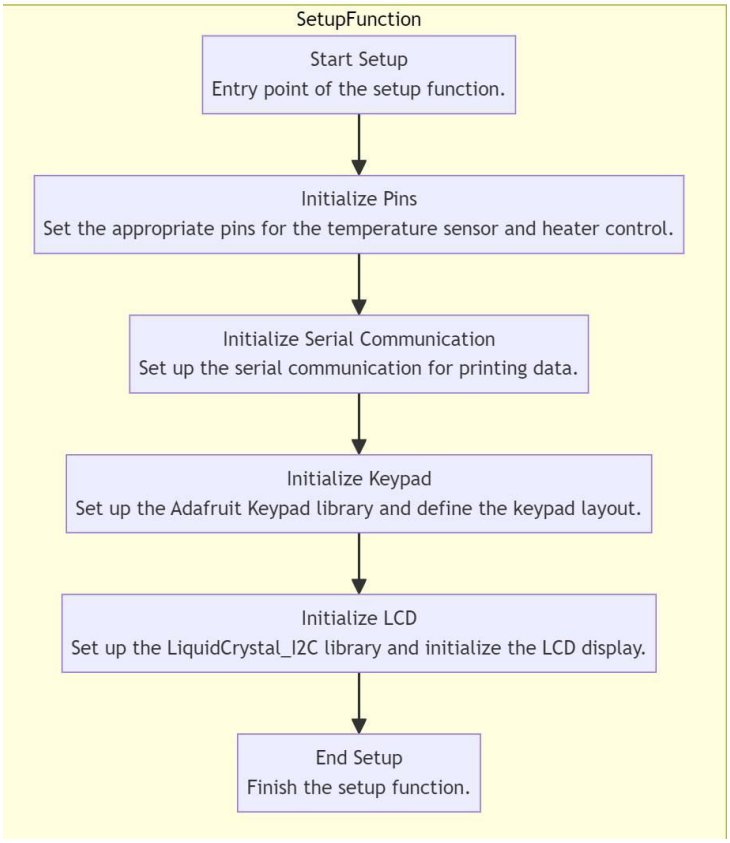
➢ Modify LCD Setpoint (if necessary)

The keypad provides the user with the ability to modify the setpoint. An LCD panel shows the updated setpoint.

➢ Rewind to the beginning

After reading the setpoint and current temperature once more, the process loops back to the beginning, where the PID control loop keeps the temperature at the target level.

Setup Function :



This is a thorough description of the Mermaid flow diagram along with an explanation of the Arduino code that is included.

Configuration Function

The Arduino code's setup function is shown in this subgraph.

It configures the keypad, initializes the LCD display, initializes the required pins, and establishes serial communication.

Primary Loop

The Arduino code's main loop is depicted in this subgraph.It records the data to the small monitor, refreshes the LCD display, reads the temperature, computes the PID control output, modifies the heater, and continuously checks for keypad input.

Before the following iteration, the loop is delayed by the designated sample time.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

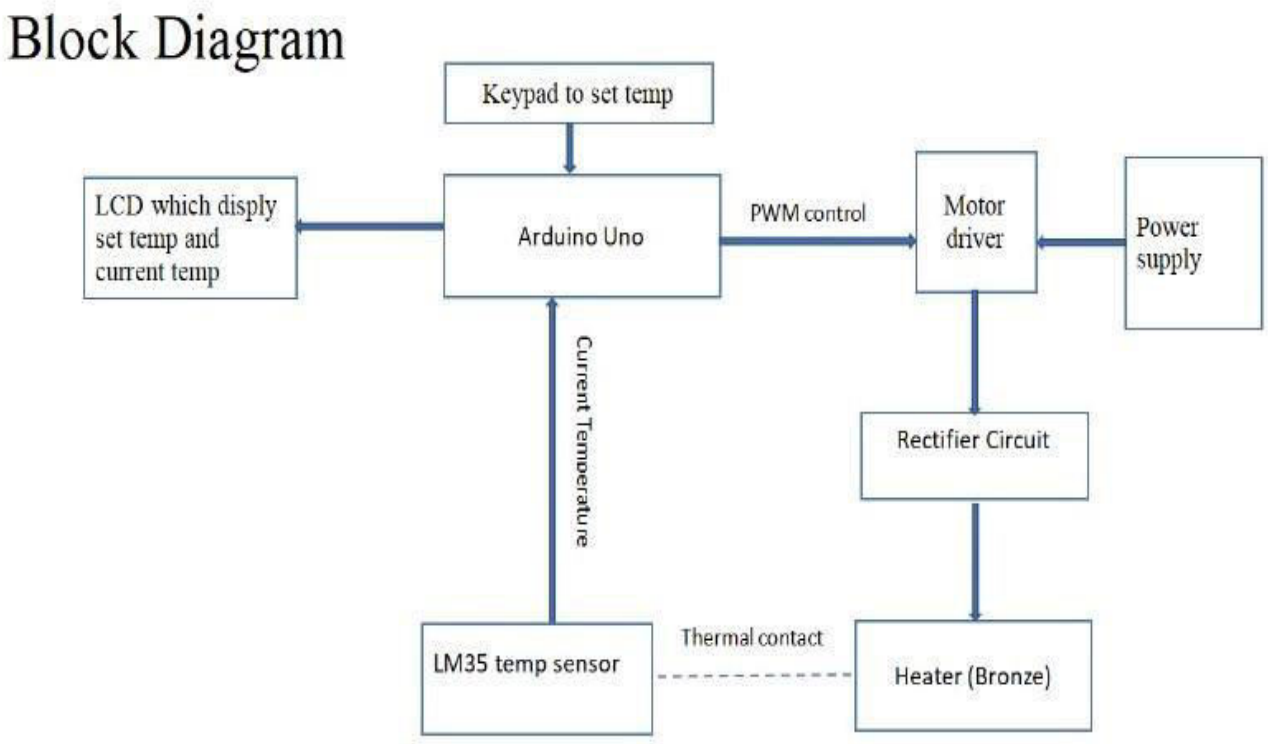
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Function for Reading Temperature
The temperature reading function is represented by this subgraph.It retrieves the temperature value determined from the temperature sensor's analog reading, converts it to a temperature in degrees Celsius, and then The logic and functionality of the code are thoroughly broken down in the flow diagram, which includes the following essential elements. handling of keypad input The code sets the required temperature setpoint by processing keypad input.

Measurement of temperature: The temperature is obtained from the sensor by the code, which then translates the analog value into a temperature in degrees Celsius. PID management Based on the temperature inaccuracy, the algorithm computes the PD control output and modifies the integral and derivative components.

Heater control:
To maintain the intended temperature setpoint, the code modifies the heater output in response to the PID control output.
ICD display:
The code modifies the ICD display to display the setpoint and current temperature. -Data logging: For monitoring and analysis, the code logs the temperature, setpoint, and PID parameters to the serial monitor.

VI. BLOCK DIAGRAM



Explanation of Block Diagram :

➢ LM35 Temperature Sensor: The temperature in Celsius is linearly proportional to the output voltage of this precise integrated-circuit temperature sensor. It measures the room temperature and generates an analog voltage output that the Arduino Uno can read.
➢ Arduino Uno: The PID control technique is implemented and data from the LM35 temperature sensor is processed using the Arduino Uno microcontroller board. It uses PWM (Pulse Width Modulation) signals to modify the heater

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

based on the error between the desired setpoint temperature and the actual temperature. It first receives the analog voltage output from the LM35 sensor.

➢ Heater: Using a motor driver circuit, the Arduino Uno regulates the heater. The Arduino modifies the duty cycle of the PWM signal delivered to the motor driver, which regulates the power supplied to the heater, based on the output of the PID control algorithm. This enables the system to modify the heater's power output in order to control the temperature.

➢ Circuit for Rectifier (Signal Diode): To guarantee that the motor driver receives only positive PWM signals, this circuit is placed in between the heater pin and the data pin. By rectifying the PWM signal, the negative PWM signal is eliminated.

➢ Motor Driver: The circuit that controls the motor driver is in charge of regulating the heater's power supply. It modifies the voltage and current delivered to the heater in accordance with the PWM signal it gets from the Arduino Uno. By serving as an interface, it enables the microcontroller to manage higher power devices such as motors and heaters from the Arduino.

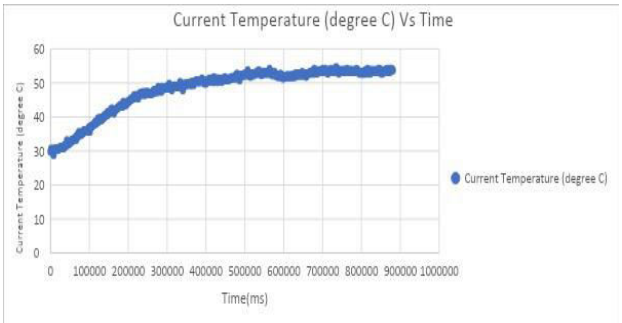
➢ LCD Display: An Arduino temperature control system that integrates a 20x4 LCD display with a keypad offers a user-friendly interface for setting temperature setpoints and keeping track of current temperature data. With this improvement, operators may enter desired temperatures straight into the

keypad, and the LCD screen gives them immediate response by showing the current temperature values. These elements are added to the system, making it more flexible and responsive to different user requirements, improving overall usability and effectiveness in temperature control applications.

VII. RESULT

Graph for Different value of PWM: We have produced PWM (Pulse Width Modulation) graphs for our temperature control system for a range of proportional parameter (P) values of the PID (Proportional-Integral-Derivative) controller. The relationship between temperature and time for various P values is seen in these graphs. The P value affects how quickly and effectively the system adapts to reach and maintain the target temperature setpoint, as well as how the controller responds to temperature variations.

1]



Objective
On X-axis = Time(ms) and
On Y-axis = current Temperature in degree celsius
Value of PWM is 27 (25%)

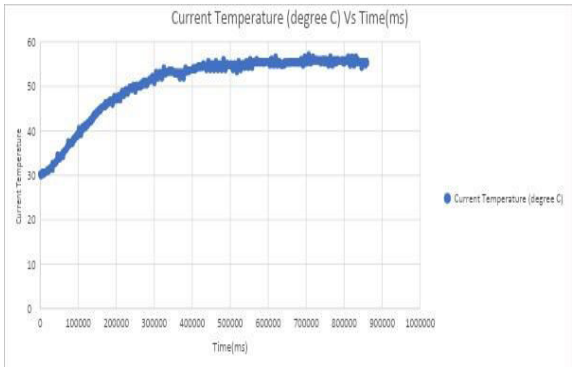
Conclusion
First temp increases exponentially then it increases linearly and saturate at temperature value 53 degree celsius.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

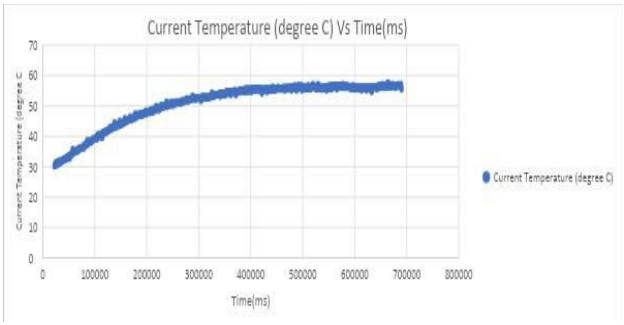
2]



Objective
On X-axis = Time(ms) and
On Y-axis = current Temperature in degree celsius
Value of PWM is 127 (50%)

Conclusion :
First temp increases exponentially then it increases linearly and saturates at temperature value 55 degree celsius .

3]



Objective On X-axis = Time(ms) and
On Y-axis = current Temperature in degree celsius
Value of PWM is 191 (75%)

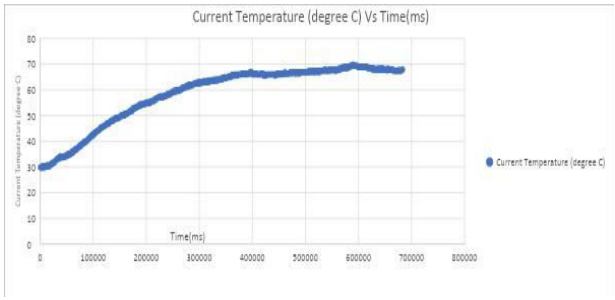
Conclusion :
First temp increases exponentially then it increases linearly and saturates at a temperature value 56 degree celsius.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

4]



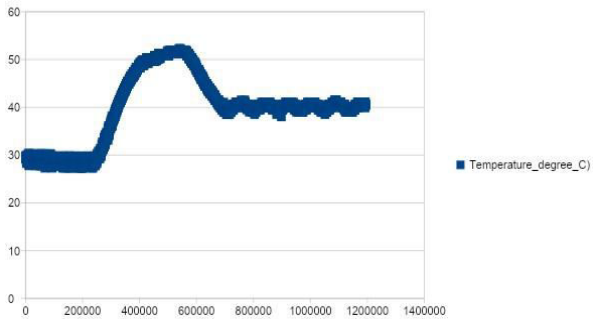
Objective On X-axis = Time(ms) and
On Y-axis = current Temperature in degree celsius
Value of PWM is 255 (100%)

Conclusion : First temp increases exponentially then it increases linearly and saturates at temperature value 67 degree Celsius

Graph for different value of Kp:
The following graph shows how a temperature control system responds to varying Kp values: Temperature fluctuations over time are displayed by each line, which relates to a particular Kp value. Faster temperature adjustments brought on by higher Kp values may cause oscillations or overshooting.Lower Kp levels, on the

other hand, result in smoother temperature swings, albeit with slower adjustments and a lower likelihood of instability.Examining these curves provides information on the behavior of the system at different proportional gain settings, which helps choose a suitable Kp value for efficient temperature control.

1. Kp=10



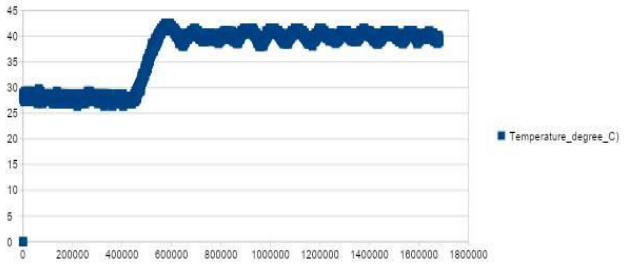
We noticed oscillations that were outside of our expected range, indicating instability in our system when the proportional gain (Kp) is adjusted to 10. This result suggests that the system oscillates around the target temperature setpoint instead of smoothly settling to it because the selected Kp value prompted an unduly aggressive response.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

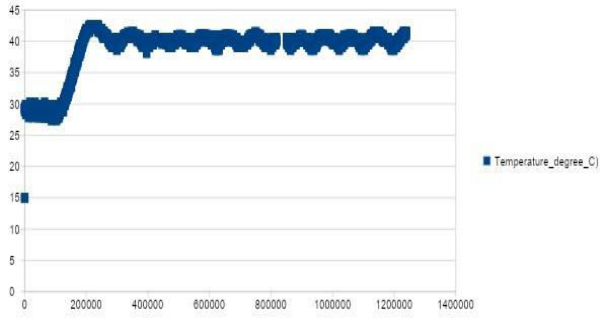
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

2. Kp=20



The PID regulator's commensurate term's control action was more successfully balanced with the other control parameters (integral and secondary terms) as evidenced by the higher stability at Kp = 20.As a result, the system oscillated less and got closer to the desired temperature setpoint with less accuracy and thickness

3. Kp=15



The larger corrective action that the proportional term exerts as Kp is raised is most likely the cause of the stability difference between Kp=15 and Kp=20. Kp=20 may cause the control system to react to temperature errors more forcefully, effectively reducing oscillations and encouraging more even temperature regulation.

Consequently, the system demonstrated reduced oscillations and more accurately and consistently approached the target temperature setpoint at Kp=20. The value of Kp for a sustained, regular oscillation is 20, and Ku=20.
The ziegler-nichols method is utilized in additional computations with this value of kp and ku.

Ziegler–Nichols Method :
First, a proportional-gain-only system is used in the procedure. P gain is increased until sustained (stable in terms of breadth) and regular (stable in terms of period) oscillations are observed in the system; the oscillation need not be centered at the setpoint. The sole tedious aspect of the Ziegler-Nichols system is this. The remainder is merely computation.

As you can see, the columns labeled TI and TD, rather than KI and KD, are the integral and derivative. The time constant that is utilized to compute the integral or derivative is denoted by these "T" variables. We apply the following relationships for our discrete-time controller in order to calculate KI and KD:

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

	K_P	T_I	T_D
P-only control	$K_U/2$		
PI control	$K_U/2.2$	$P_U/1.2$	
PID control	$K_U/1.7$	$P_U/2$	$P_U/8$

Ki=KD=KP(TD/T) Ki=KP(T/TI)
where T is the PID interval; in our system, this means that **T = 117 sec.**

As a result, as we can see from the results, when various graphs are used to analyze the proper Kp value for our stable system, the system oscillates less and gets closer to the desired temperature setpoint at Kp=20 and Pu = 96 seconds, the ultimate period between two periodic oscillations.

Calculations For Kp(Proportional gain),Ki(integral gain, Kd(Derivative Gain)
We have Ku= 20 , Pu = 96 sec
For P Controller , **Kp = 20 / 2 = 10**
For PI- Controller , **Kp = 20 / 2.2**
For PID Controller , **Kp = 20 / 1.7= 11.76**
We have Pu= 96 sec , T=117 sec
For PI- Controller , **Ti =** Pu / 1.2 = 96/1.2 **= 80 sec** Ki=Kp(T/Ti) = 11.7(117/80)
Ki = 17.11
For PID Controller , **Ti =** Pu / 2 = 96/2 **= 48 sec**
Ki=Kp(T/Ti) = 11.7(117/48)
Ki = 28.51
Td = Pu/ 8= 96/8 **= 12 sec**
Kd=Kp(Td/T) = 11.7(12/117) **Kd=1.2**
Table:

	Kp (Proportional Gain)	**Ki (Integral Gain)**	**Kd (derivational Gain)**
P-Only Controller	**10**		
PI-Only Controller	**9.09**	**17.11**	
PID Controller	**11.76**	**28.51**	**1.2**

Output=*P+I+D*

VIII. CONCLUSION

Temperature regulation was significantly improved by implementing a PID controller for temperature management utilizing the Ziegler-Nichols approach using an Arduino Uno, TB6600 motor driver, and LM35 temperature sensor.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The Ziegler-Nichols technique made it possible to identify the ideal PID controller parameters, which improved stability and sped up response times for preserving the target temperature setpoint. The TB6600 motor driver effectively controlled the heating/cooling element to accomplish exact temperature adjustments, while the Arduino Uno offered a versatile and configurable platform for applying the PID control algorithm. With exact temperature feedback from the LM35 temperature sensor, the PID controller could make well-informed control decisions. All things considered, the combination of these elements and the use of the Ziegler-Nichols technique produced a strong and efficient temperature control system appropriate for various industrial and research applications.

ACKNOWLEDGMENTs

We would like to express our sincere gratitude to all the people and organizations that helped us finish this research project on PID controllers for temperature control systems. First and foremost, we would like to sincerely thank Dr. M.P. Sardey Mam and Prof. Garde Sir whose advice, knowledge, and constant support have been extremely helpful at every turn during this study project. Their enlightening comments, helpful critiques, and support have not only influenced the course of this research but also promoted our academic advancement. We also like to express our sincere gratitude to the instructors and personnel of the AISSMS Institute of Information Technology's Electronics and Telecommunication Department for their ongoing support and help. Their dedication to academic brilliance, eagerness to share knowledge, and commitment to building a favorable research environment have significantly contributed to the success of this project. Finally, we would like to express our gratitude to all the people, groups, and establishments that have contributed materials and spaces to this project. Their assistance has been crucial to the accomplishment of this research's achievement.

To sum up, we would like to sincerely thank everyone who has helped to make this research endeavor a reality, whether directly or indirectly. We truly appreciate the chance to conduct this study and add to the body of knowledge in our profession, and your encouragement and support have been invaluable.

REFERENCES

The following references offer information on using Arduino microcontrollers to implement PID control algorithms for temperature control applications. They cover a range of topics, including design, implementation, and experimental findings, all of which are beneficial for comprehending and constructing your own Arduino-based PID temperature control system.

1. A. Saravanan and S. S. Jagtap, "Temperature Control System Using Arduino and PID Algorithm,"International Journal of Control Theory and Applications, vol. 9, no. 14, pp. 6515-6521, 2019.
2. Kiam Heong Ang, and G. Chong. “PID control system analysis, design, and technology.” IEEE Transactions on Control Systems Technology, vol. 13, no. 4, July 2005, 2005, pp. 559 - 576. ieeexplore, https://ieeexplore.ieee.org/document/1453566.
3. J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers", Trans. ASME, vol. 64, pp. 759-768, 1942. (K.J. Åström and T. Hägglund)
4. M. S. Mahmoud, "PID Control of Temperature Process Using Arduino," International Journal of Engineering Research & Technology (IJERT), vol. 4, no. 5, pp. 354-359, 2015.
5. T. G. Hughes, "PID Control for Temperature Regulation Using Arduino and PWM," International Journal of Engineering Research & Technology (IJERT), vol. 6, no. 4, pp. 213-218, 2017.
6. Madavan, V., Thangavel, K., & Rajendran, V. (2021). “Implementation of PID Temperature Control System using Arduino”. International Journal of Control and Automation, 14(3), 467-476.
7. A. K. Singh, "Microcontroller-Based PID Temperature Controller Using PWM," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE), vol. 4, no. 8, pp. 6785-6790, 2015.
8. T. G. S. Kiran et al., "Optimization of PID Controller for Temperature Control System using Simulated Annealing," International Journal of Control Theory and Applications, vol. 11, no. 22, pp. 109-115, 2018.
9. Liu, J., Wu, J., & Zhao, D. (2019). “Research on temperature control systems based on the PID algorithm in mobile water treatment devices”. In 2019 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA) (pp. 348-351). IEEE.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

10. Singh, S., Chaudhari, K., Singh, R., & Kumar, D. (2020). “Temperature Control System Using PID Controller”. In 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 1-5). IEEE.
11. T. G. S. Kiran et al., "Optimization of PID Controller for Temperature Control System using Simulated Annealing," International Journal of Control Theory and Applications, vol. 11, no. 22, pp. 109-115, 2018.
12. Saleh, M. S., Abdel-Magid, Y. L., & Mahmoud, K. H. (2021).” PID Control of a Temperature Control System Using Gravitational Search Algorithm and Genetic Algorithm”. Journal of Control, Automation and Electrical Systems, 32(2), 193-203

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details