KARNATAK LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

(APPROVED BY AICTE, NEW DELHI)

Report On

project of the title

"PID Based Temperature Controller"

Submitted in the partial fulfillment for the academic requirement of

5th Semester B.E.

In

Electronics &Communication Engineering

SUBMITTED BY:

NAME	USN	Marks
Likhit Porwal	**2GI22EC066**	
Manasi Shilwant	**2GI22EC069**	
Pramod Gurav	**2GI22EC099**	
Rayyan Hanchanal	**2GI22EC115**	

GUIDED BY:

Dr. Saurav Mitra

Professor, Dept. of ECE, KLSGIT, Belagavi

2024-2025

Karnataka Law Society's
GOGTE INSTITUTE OF TECHNOLOGY
Udyambag Belagavi -590008
Karnataka, India.
Department of Electronics &Communication and Engineering

Certificate

This is to certify that the project titled **"PID Based Temperature Controller"** carried out by Likhit P, Manasi S, Pramod G, Rayyan H **USNs: 2GI22EC066, 2GI22EC069, 2GI22EC099, 2GI22EC115** respectively have submitted in partial fulfilment of the requirements for 5th semester B.E. in **ELECTRONICS AND COMMUNICATION AND ENGINEERING.** It is certified that all corrections/suggestions indicated have been incorporated in the report. The seminar report has been approved as it satisfies the academic requirements prescribed for the said degree.

Signature of the Guide
Dr. Saurav Mitra

Signature of the H.O.D
Dr. Supriya Shanbhag

<u>INDEX</u>

Introduction:

Temperature regulation is a critical requirement in various industrial applications, where maintaining precise thermal conditions ensures efficiency, safety, and optimal performance of machinery and processes. This project focuses on designing and implementing a PID-based temperature control system tailored for industrial heating applications.

The Proportional-Integral-Derivative (PID) control algorithm is widely used in automation and process control due to its ability to maintain stable and accurate control over dynamic systems. In this project, the PID control mechanism is implemented using operational amplifiers and discrete hardware, ensuring reliable and responsive temperature adjustments.

A thermocouple serves as the primary temperature sensor, offering high accuracy and durability in harsh industrial conditions. The ESP32 microcontroller is employed for system control and interfacing, enabling real-time monitoring and adjustments. The heating element is controlled via a relay module, which acts as a switch to regulate power based on the computed PID output.

Purpose:

The primary objective of this project is to develop a robust and efficient temperature control system that provides:

- Precise temperature regulation to maintain stability in industrial heating applications.
- Optimized performance and energy efficiency, reducing wastage and improving process consistency.
- Enhanced safety features, ensuring protection against overheating and thermal fluctuations.

By integrating hardware-based PID control with microcontroller automation, this system offers a scalable and adaptable solution for industrial environments requiring reliable temperature management.

Problem Statement:

The project aims to develop an intelligent temperature control system using Proportional-Integral-Derivative (PID) control. The system will be capable of continuously monitoring temperature, making real-time adjustments, and maintaining stability in industrial environments. By leveraging a thermocouple sensor for temperature measurement and an ESP32 microcontroller for processing and control, the system will ensure accurate and responsive regulation of industrial heating processes.

Objectives:

To achieve the desired functionality, the project is structured into several key objectives:

1. Study PID Control

- Understand the fundamentals of PID (Proportional-Integral-Derivative) control theory.
- Explore how PID is used in automation and process control.

2. Understand Thermocouple Technology

- Study thermocouple working principles and types.
- Analyze their industrial applications and selection criteria.

3. Tuning PID

- Learn various PID tuning techniques, including Ziegler-Nichols and manual tuning.
- Optimize PID parameters for precise temperature regulation.

4. Design a Temperature Controller

- Develop a control system capable of stabilizing temperature fluctuations.
- Ensure system reliability and accuracy in industrial settings.

5. Implement Hardware

- Integrate hardware components, including a thermocouple, heating element, and control circuitry.
- Assemble and test the physical system for robustness

6. Interface with ESP32 Microcontroller
   - Program the ESP32 for PID computation and system control.
   - Implement communication protocols for real-time monitoring.

7. Complete System Implementation
   - Finalize the design by integrating software and hardware components.
   - Test the system under real-world industrial conditions.

Block Schematic for Temperature controller using Op-Amps:

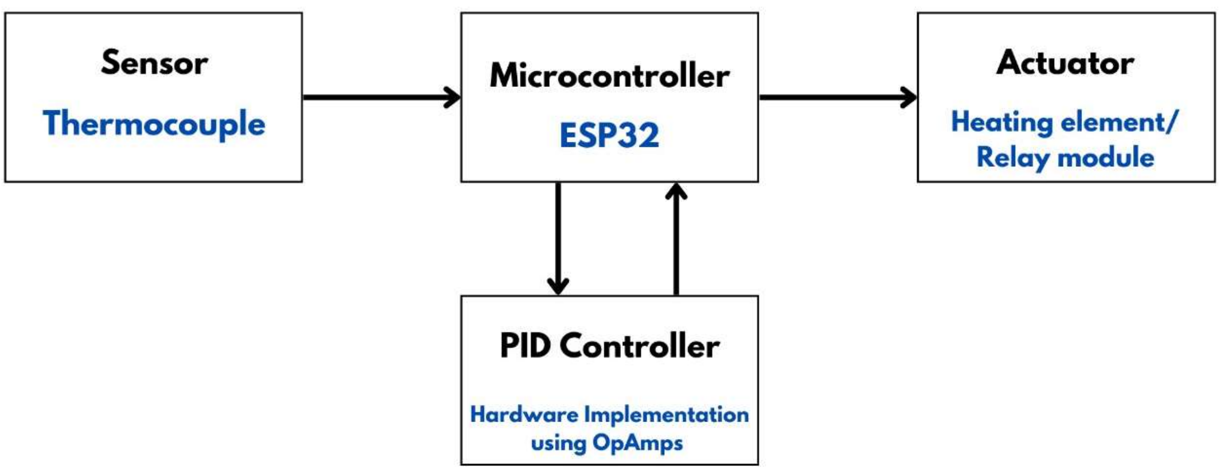


Fig 1: Block schematic for PID based temperature controller

Sensor Testing and Calibration:

Fig 2a

Fig 2b

Fig 2a and 2b: Sensor Testing and Calibration in Control system lab, KLS GIT

In this project, the RTD sensor was calibrated against a known reference temperature of 25° C using a thermometer as the standard. The calibration ensured that the RTD sensor's output accurately corresponded to the actual temperature.

Calibration Procedure:

1. Setup:

- The RTD sensor was connected to the signal conditioning circuit and the data acquisition system.
- A high-accuracy thermometer was placed in the same environment to act as the reference.

2. Reference Temperature:

- The environment was stabilized at 25° C, confirmed by the thermometer.

3. Output Comparison:

- The RTD's output resistance was measured and compared with the reference temperature reading from the thermometer.

4. Adjustment:

- Any discrepancies between the RTD's resistance and the expected value at 25° C (based on the RTD's resistance-temperature curve) were corrected by adjusting the offset in the processing system.

Results:

- After calibration, the RTD sensor's output was verified to accurately represent the temperature at 25° C.
- The calibration ensures improved accuracy and reliability in the system's temperature measurements, particularly in real-time applications.

Open loop System Response:

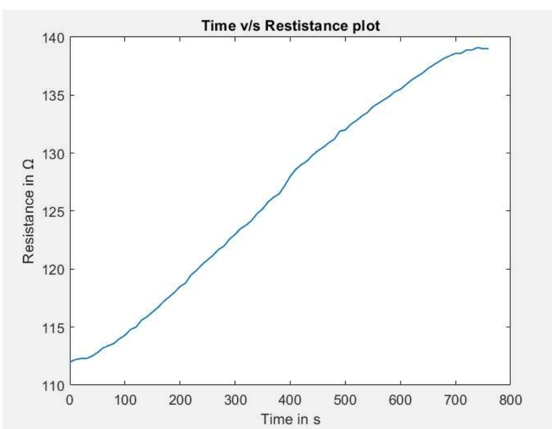


Fig 3a: Time vs. Resistance plot

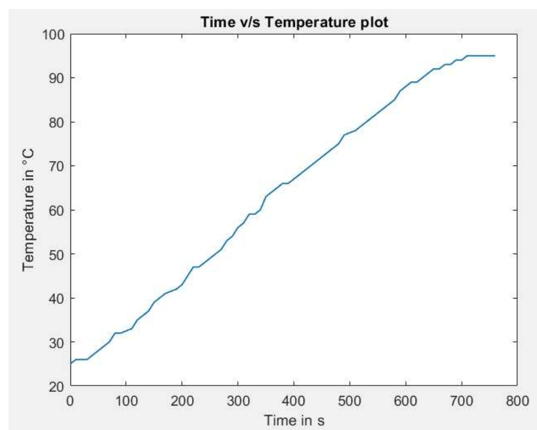


Fig 3b: Time vs. Temperature plot

Inference:

The readings obtained from sensor calibration and testing are plotted in MATLAB to analyze the response and verify performance.

Obtaining Process Reaction Curve:

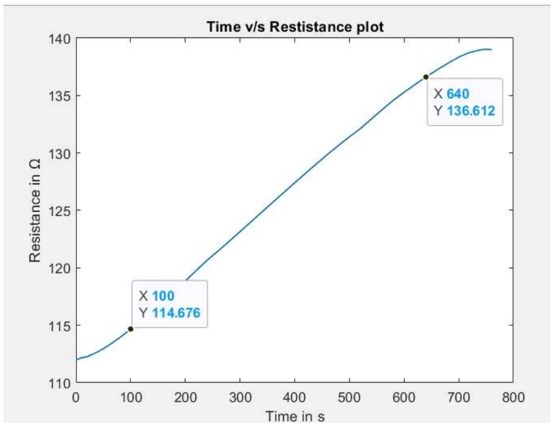


Fig 4a: Time vs. Resistance curve

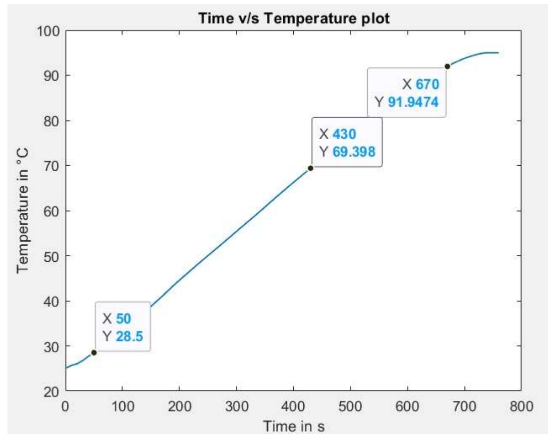


Fig 4b: Time vs. Temperature curve

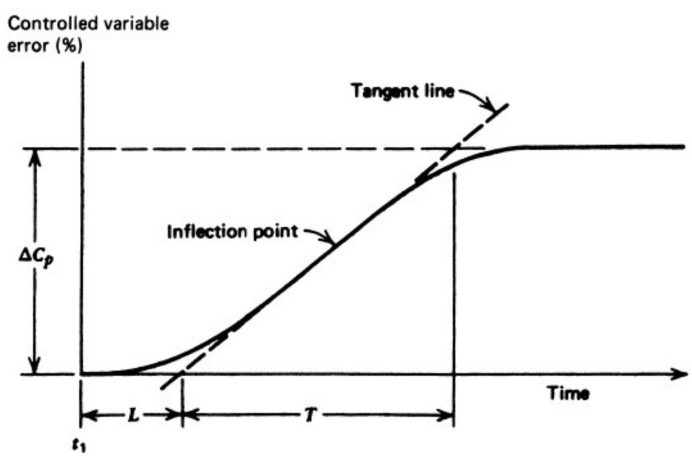


Fig 4c: Reference Process-reaction curve graph for loop tuning

By linearizing and smoothening the response (as in Fig 9a and 9b), the graph closely resembles the one in the reference (Fig 9c). Therefore, empirical formulas are used to determine the time constant, which is then utilized to derive the transfer function.

Calculations and Transfer Function:

For Time Constant,

63.2% x Change in output + Initial output

= (63.2% x 70) + 25

= 44.24 + 25

= 69.25 ~ 69.3 °C

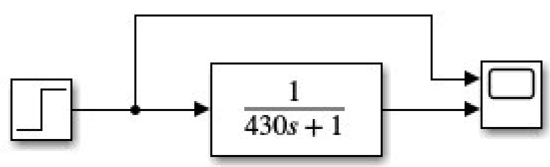
This value corresponds to T = 430s (from process reaction curve)

Therefore, τ = 430s

We know that,

$$Transfer\ Function,\ H(s) = \frac{1}{\tau s} = \frac{1}{430s+} \qquad \text{… (eqn 1)}$$

PID Tuning using Simulink:



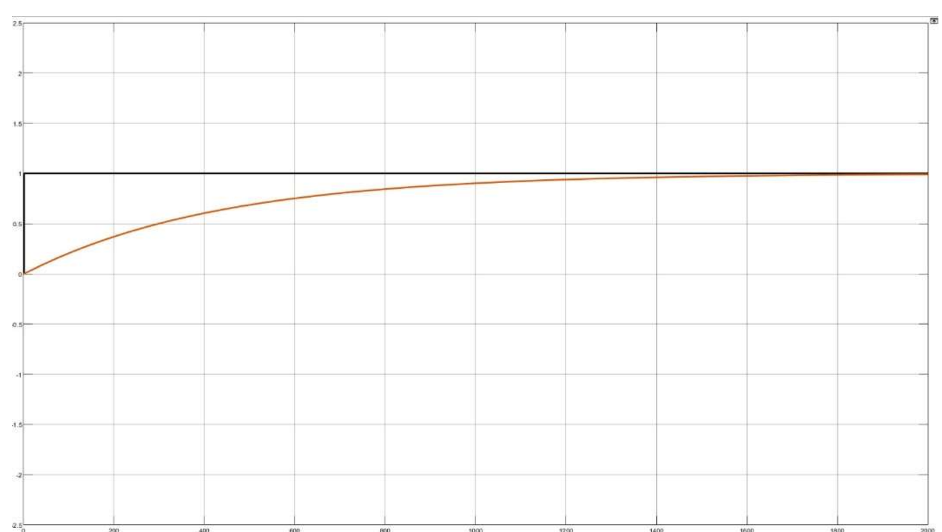


Fig 5a: Open Loop Process

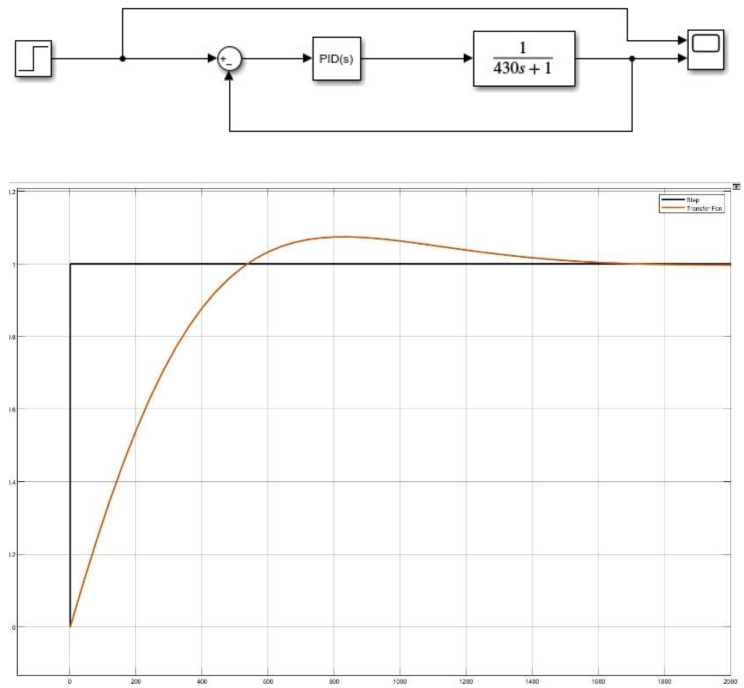


Fig 5b: Closed Loop Process

Through the tuning process, P= 1.337, I= 0.006, D= 0.187

Op-Amp Circuit:

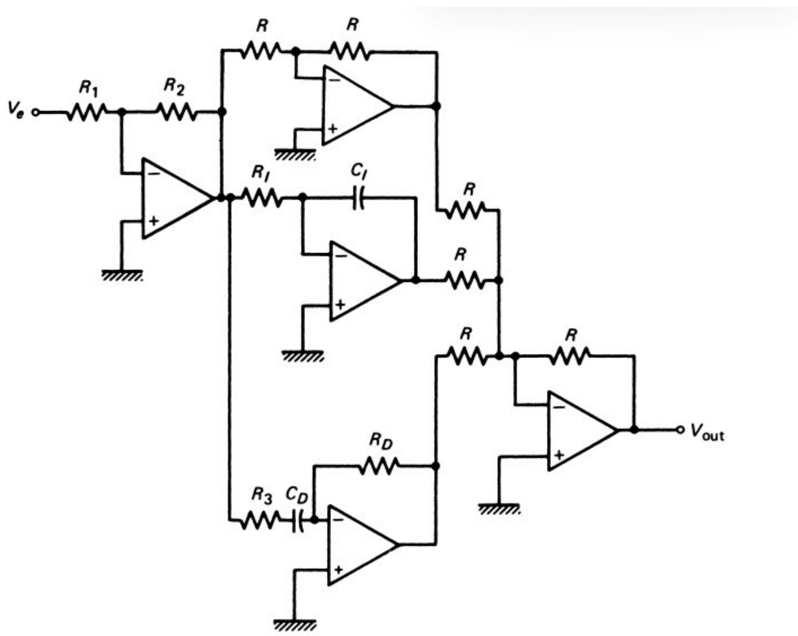


Fig. 6: Op-Amp based PID Controller

These circuits have shown that the direct implementation of controller modes can be provided by standard op amp circuits. It is necessary, of course, to scale the measurement as a voltage within the range of operation selected by the circuit. Furthermore, the outputs of the circuits shown have been voltages that may be converted to currents for use in an actual process-control loop. These circuits are only examples of basic circuits that implement the controller modes. Many modifications are employed to provide the controller action with different sets of components.

Software Simulation of PID:

Software simulation of Analog Op-Amp based PID circuit using Tinker-Cad simulation software.

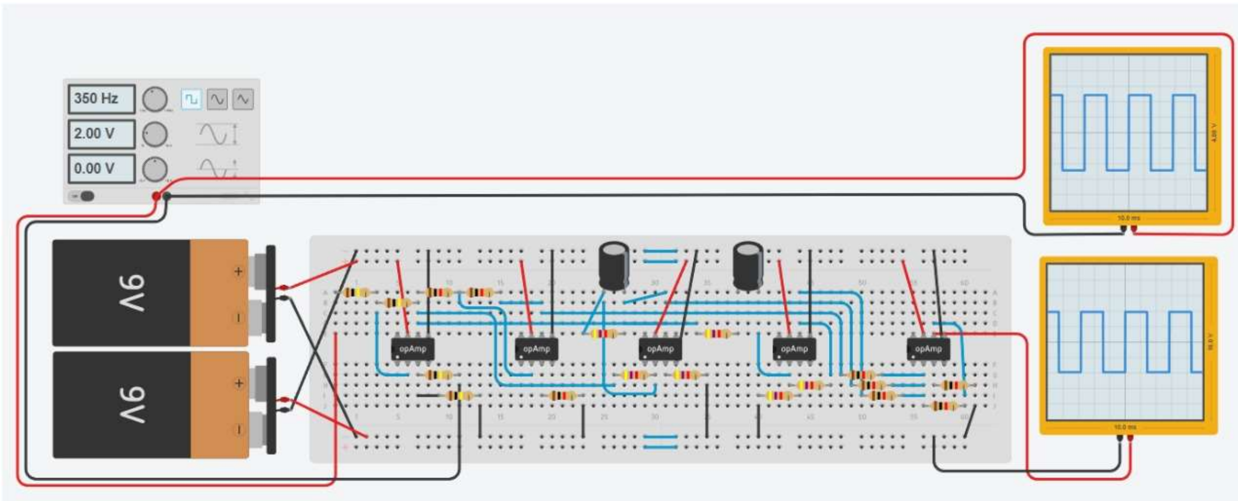


Fig. 7a: Output of P controller

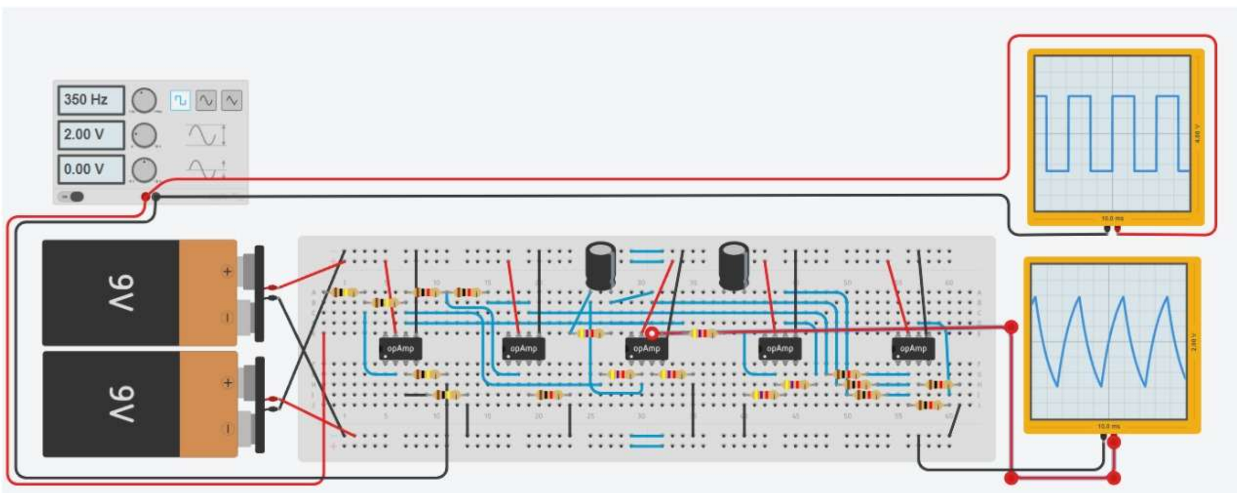


Fig. 7b: Output of I controller

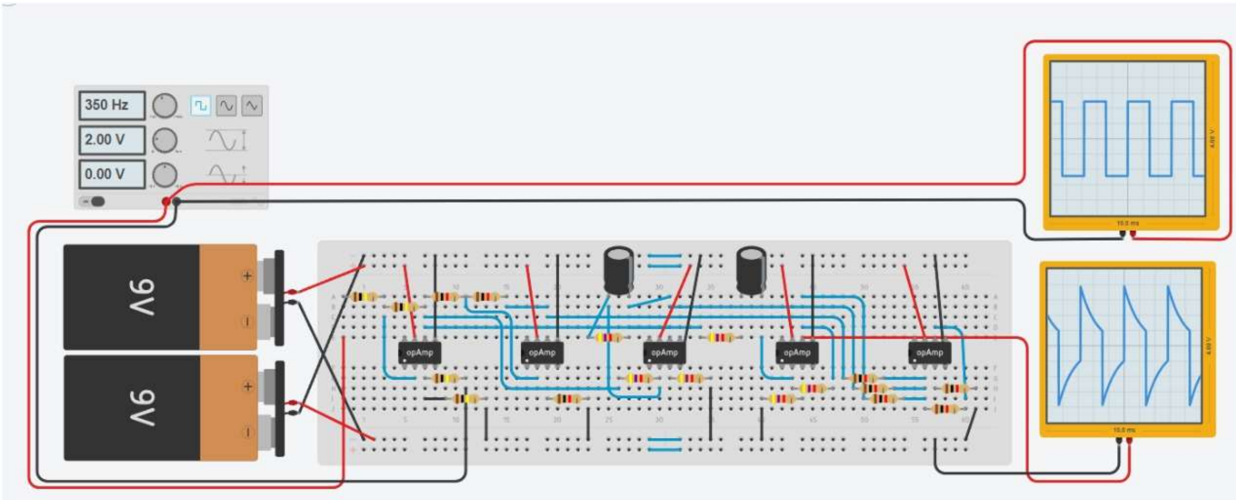


Fig. 7c: Output of D controller

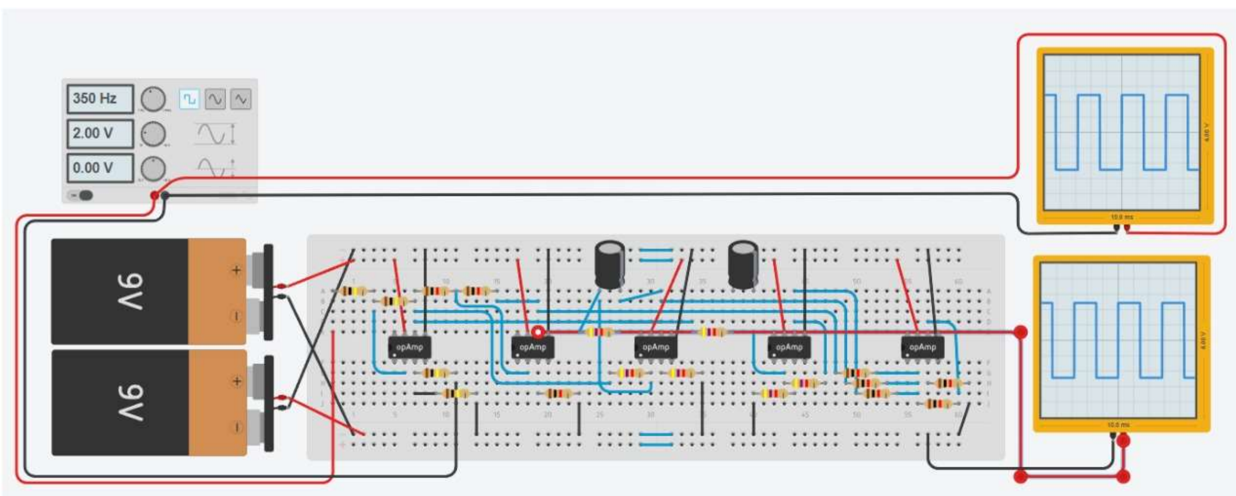


Fig. 7d: Output of PID controller

Block Schematic for Temperature Controller:

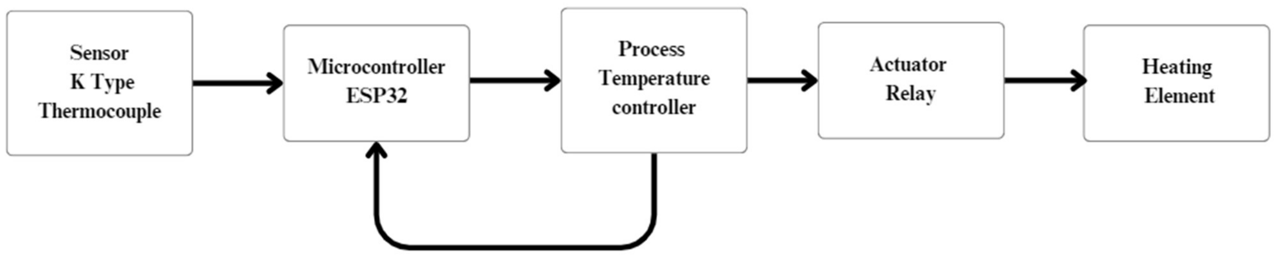


Fig 8: Block schematic for PID based temperature controller using ESP-32

This workflow explains how the ESP32 microcontroller performs PID control using a combination of hardware and software.

1. Temperature Sensing

- The thermocouple measures the system temperature and sends its voltage signal to the MAX6675 thermocouple amplifier.
- The MAX6675 converts the signal into a digital temperature value and communicates it to the ESP32 via the SPI protocol.

2. Error Signal Calculation

- The ESP32 compares the desired temperature (setpoint) with the current temperature from the thermocouple.
- The difference is computed as the error signal.

3. PID Algorithm Execution

- The ESP32 software processes the error signal using the PID formula:
  - Proportional (P): Adjusts based on the current error.
  - Integral (I): Eliminates accumulated errors.
  - Derivative (D): Predicts and dampens sudden changes.
- The calculated output determines the correction needed.

4. Actuator Control

- The ESP32 sends the PID output to a relay module, which controls the heating element's power.
- The relay switches the heater on/off or adjusts intensity to maintain the desired temperature.

5. Feedback and Monitoring

- The ESP32 continuously monitors the temperature and updates the PID computation in real time.
- It can display temperature data on an I2C LCD or transmit it wirelessly for remote monitoring via Wi-Fi.

6. ESP-32 Relevancy:

a) **Flexibility & Tunability:** ESP32 allows easy software-based PID tuning, while op-amp circuits require hardware modifications.

b) **Precision & Stability:** Digital calculations in ESP32 ensure better accuracy and long-term stability, unlike analog circuits that suffer from component drift.

c) **Sensor & Output Integration:** ESP32 directly interfaces with digital sensors and output devices (e.g., displays, relays, Wi-Fi), reducing extra circuitry.

d) **Data Logging & Remote Monitoring:** ESP32 enables temperature logging, Wi-Fi/Bluetooth connectivity, and remote control, which analog circuits cannot.

e) **Compact & Multi-Functional:** A single ESP32 can replace multiple analog components, reducing circuit complexity while supporting multi-zone control.

Methodology:

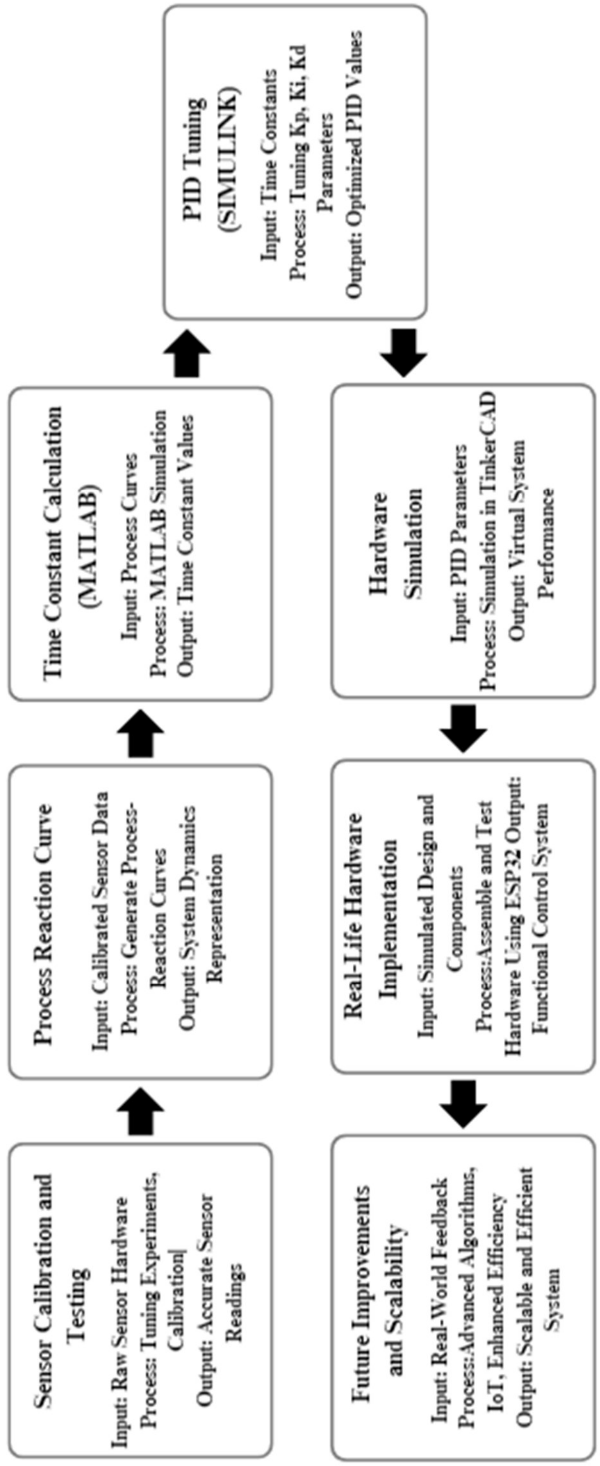


Fig 9: Flow diagram of Methodology for PID based temperature controller

1. **Calibration:**
   Ensures sensor accuracy using a known reference, providing reliable input for control processes.
2. **Curve Derivation:**
   Captures system dynamics to understand thermal response behavior.
3. **Time Constants:**
   MATLAB extracts key parameters for modeling system behavior.
4. **PID Tuning:**
   Simulink optimizes control parameters for stability and minimal error.
5. **Simulation:**
   Virtual testing validates performance before real-world application.
6. **Implementation:**
   Integrates hardware and PID for real-time temperature control.
7. **Improvements:**
   Adds scalability and advanced features like IoT for future readiness.

Hardware used:

1. Sensor (Thermocouple): Measures the temperature and sends it to the ESP32 microcontroller.
2. Microcontroller (ESP32): Processes the temperature data and collaborates with the PID controller to adjust the system.
3. PID Controller: A hardware-based control mechanism (using Op-Amps) calculates corrections to minimize temperature errors.
4. Actuator (Relay): Executes adjustments (e.g., heating or switching) to maintain the desired temperature.

The system uses feedback control to stabilize temperature efficiently.

System Components and Tools:

Data Sources used:

Thermocouple for temperature
measurement.

Language used:

Embedded C for ESP32
programming.

Methods Used:

PID control (Proportional-Integral-Derivative),
signal processing and feedback control.

Software used:

Arduino IDE.

Table 1:

Components
Resistors (1kΩ, 4.7kΩ)
Thermocouple
Thermocouple drivers (MAX6675)
Microcontroller (ESP32)
Breadboard
Rotary Encoder
i2c LCD Display
MOSFET (IRFZ44N)
BJT (C8050)
Wires

Hardware Implementation:

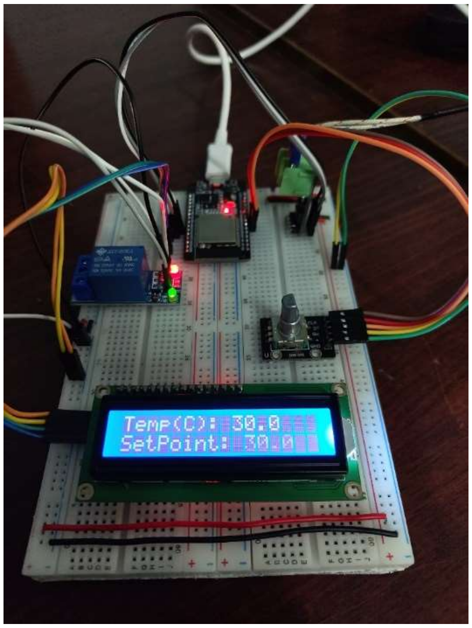


Fig 10a

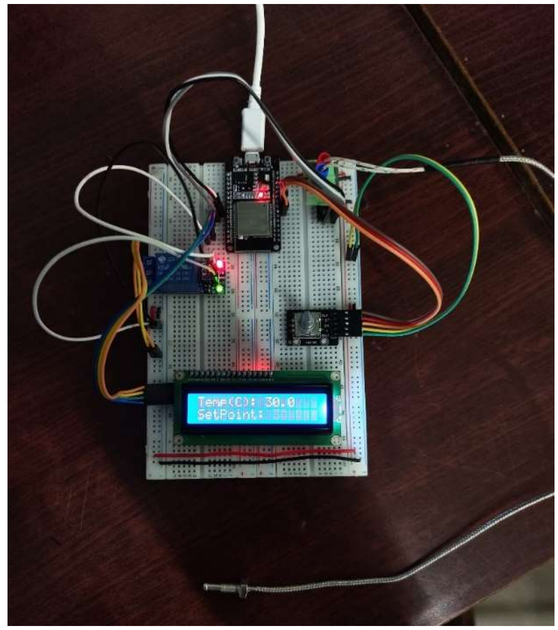


Fig 10b

Fig 12a and 12b: Hardware implementation of PID based temperature controller using ESP-32

Future Scope:

Future Scope of the PID-based Smart Temperature Monitoring and Control System:

1. Networked Control System with Intelligence

- Develop a networked control system that enables distributed control over multiple devices and sensors.
- Integrate machine learning algorithms to enhance decision-making and system efficiency in real-time.

2. Response Time and Stability Analysis

- Measure and analyze the response time of the control system.
- Improve stability by optimizing the PID control parameters and leveraging microcontroller capabilities for enhanced performance.

3. Communication Among Control Systems
   - Implement communication protocols such as F-bus, Modbus, and Digital Twin technology to enable seamless data exchange among various control systems.
   - Facilitate remote monitoring and control through communication interfaces.

4. CUDA and Parallel Computing
   - Explore the use of CUDA (Compute Unified Device Architecture) and parallel computing for executing control algorithms faster and more efficiently, especially for complex systems with multiple controllers.

5. Control of Multiple Processes with Varying Speeds
   - Extend the system to handle multiple processes with different response speeds and dynamics.
   - Adapt the PID control logic to work effectively with systems operating at varying speeds and conditions.

These advancements will enhance the system's capabilities and make it suitable for more complex and larger-scale industrial applications.

References:

1. Palaniyappan T K, Vaishali Yadav, Ruchira, Vijay Kumar Tayal, Pallavi Choudekar, "PID Control Design for a Temperature Control System", IEEE, April 2018

2. Vatia Fahrunisa Rahmadini, Alfian Ma’arif, Nur Syuhadah Abu, "Design of Water Heater Temperature Control System using PID Control", IEEE, 2023

3. C. D. Johnson, Process Control Instrumentation Technology, 8th ed., Pearson New International Edition, 2014.