**Automated Room Temperature Regulator**

Ashwin B

CB.EN.U4CSE21606

Amrita School Of Computing

Coimbatore, India

[cb.en.u4cse21606@cb.students.amrita.edu](mailto:cb.en.u4cse21606@cb.students.amrita.edu)

Elson Kuriakose

CB.EN.U4CSE21616

Amrita School Of Computing

Coimbatore, India

[cb.en.u4cse21616@cb.students.amrita.edu](mailto:cb.en.u4cse21616@cb.students.amrita.edu)

Karthik R Krishnan  
CB.EN.U4CSE21630  
Amrita School Of ComputingCoimbatore, India  
[cb.en.u4cse21630@cb.students.amrita.edu](mailto:cb.en.u4cse21630@cb.students.amrita.edu)

This project focuses on the automation of a room temperature regulator using the Raspberry Pi Pico development board with an RP2040 microcontroller and a L298N Motor Driver. The RP2040 microcontroller is equipped with an internal temperature sensor, forming the core of a straightforward setup that employs the Proportional-Integral-Derivative (PID) algorithm. The PID algorithm is utilized to dynamically regulate the fan speed by adjusting the PWM (Pulse Width Modulation) duty cycle.

Keywords—Raspberry Pi Pico, DC Fan, L298N Module

# Introduction

In the dynamic landscape of embedded systems, the Raspberry Pi Pico, powered by the RP2040 microcontroller, stands as a robust platform primed for groundbreaking projects. This endeavor undertakes the intriguing task of automating a room temperature regulator, harnessing the capabilities of the Raspberry Pi Pico development board in tandem with the RP2040 microcontroller and a L298N Motor Driver. At the heart of this innovative setup lies the RP2040 microcontroller, equipped with an internal temperature sensor, forming the foundational element of a streamlined configuration. The system employs the Proportional-Integral-Derivative (PID) algorithm to dynamically regulate fan speed by meticulously adjusting the Pulse Width Modulation (PWM) duty cycle. This integration not only ensures precise temperature control but also optimizes energy efficiency. The Raspberry Pi Pico's GPIO pins play a pivotal role in interfacing with the internal temperature sensor and the L298N Motor Driver, facilitating seamless communication and control.

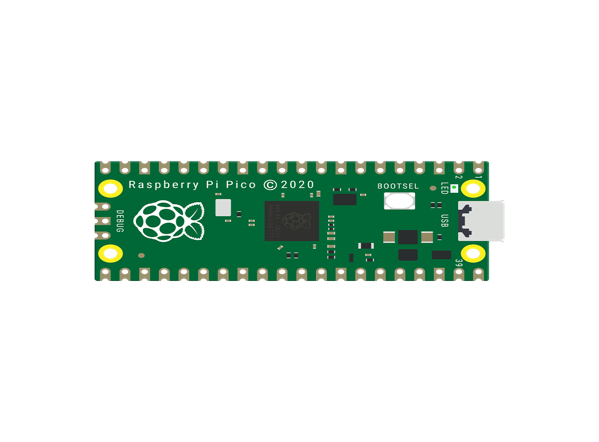
# Architecture

The automated room temperature regulator project introduces an innovative application of the Raspberry Pi Pico, showcasing its adaptability and efficiency in real-world scenarios. At the heart of this system is the Raspberry Pi Pico, chosen for its compact design, versatile GPIO pins, and processing capabilities. Serving as the central control unit, the Pi Pico orchestrates seamless interactions between various components. Specifically tailored for room temperature regulation, the project utilizes the RP2040 microcontroller within the Raspberry Pi Pico, equipped with an internal temperature sensor. This sensor forms the basis for a simple yet effective setup, showcasing the PID algorithm for precise fan speed control through PWM duty cycle adjustments. The use of PID coefficients allows for fine-tuning the system's performance, achieving effective temperature regulation. The comprehensive usage instructions guide users through the setup process, ensuring a smooth implementation of the project. In conclusion, this project demonstrates the Raspberry Pi Pico's capability to implement closed-loop PID control for practical applications, underscoring its potential in environmental regulation and serving as a foundation for broader industrial processes. The simplicity and effectiveness of the system pave the way for diverse applications of microcontrollers in real-world automation scenarios.

# Components

|  |  |  |
| --- | --- | --- |
| S.no | Components | Quantity |
| 1 | Raspberry Pi Pico | 1 |
| 2 | L298N Module | 1 |
| 3 | 12V DC FAN | 1 |
| 4 | Connecting Wires | 6 |
| 5 | AC Adapter | 1 |

## Raspberry Pi Pico



The Raspberry Pi Pico spearheads the room temperature regulator project, acting as the central brain and control nexus. With its compact design and GPIO pins, it seamlessly interfaces with the RP2040 microcontroller and L298N Motor Driver, facilitating an innovative room temperature regulation system. Harnessing the internal temperature sensor of the RP2040, this setup employs the Proportional-Integral-Derivative (PID) algorithm for precise control. The PID algorithm dynamically adjusts the fan speed through Pulse Width Modulation (PWM) duty cycle, ensuring efficient temperature management. The Pico orchestrates sensor data acquisition, algorithmic processing, and triggers actions to maintain the optimal room temperature.

## L298N Module

The L298N Motor Driver assumes a crucial role in the Automated Fan Control project, serving as the interface between the Raspberry Pi Pico RP2040 and the fan mechanism. Its robust design and dual H-bridge architecture make it an ideal choice for motor control applications. The L298N efficiently translates the PWM signals from the Pico into precise motor speed adjustments, ensuring the fan responds seamlessly to temperature variations. Equipped with built-in protection features, such as overcurrent and overtemperature safeguards, the L298N enhances the reliability of the system. This motor driver module becomes the linchpin for translating digital commands from the Pico into tangible and controlled fan operations, contributing to the overall effectiveness of the room temperature regulation project.

## Connecting Wires

The wiring process in the Automated Fan Control project revolves around creating a seamless connection between the Raspberry Pi Pico RP2040 and the L298N Motor Driver. Commence by identifying the appropriate GPIO pins on the Raspberry Pi Pico, specifically designating pins for motor control, power, and ground.

## 12V DC FAN

In the Automated Fan Control project, the 12V DC Fan is a pivotal component responsible for maintaining optimal room temperature. To seamlessly integrate the fan into the system, direct connections are established without the use of a breadboard. The fan is linked directly to the L298N Motor Driver, which serves as the interface between the Raspberry Pi Pico RP2040 and the fan mechanism. This straightforward connection method ensures a robust and reliable setup. The GPIO pins on the Raspberry Pi Pico are directly employed to establish connections for power, ground, and motor control, simplifying the wiring process and contributing to the efficiency of the temperature regulation system

## AC Adapter

## The AC adapter plays a vital role in powering the Automated Fan Control system, serving as a reliable alternative to the USB cable for supplying electrical energy to the Raspberry Pi Pico RP2040. In this setup, the AC adapter replaces the USB power source, providing a dedicated power supply for the project. The AC adapter typically features a standard power plug on one end, which connects to a power outlet. On the other end, it is equipped with a connector compatible with the power input requirements of the Raspberry Pi Pico. This direct power connection ensures a stable and continuous power source for the microcontroller and associated components. The absence of a USB cable in this context does not affect the functionality of data transfer, as the primary purpose of the AC adapter is to supply power, while communication and programming are facilitated through other means.

# Methodology

## Reading Temperature Sensor

The internal core temperature sensor of the RP2040 is used to measure ambient temperature. The sensor data is read via the ADC and converted to Celsius.

## PID Control Algorithm

A PID control loop is implemented in software, taking the current temperature as input and calculating the PWM duty cycle to adjust fan speed and regulate temperature.

## PWM Motor Control

The calculated PWM duty cycle is output via GPIO to the L298N motor driver to control power delivered to the fan, modulating its cooling effect.

## Tuning PID Coefficients

The proportional, integral and derivative coefficients of the PID loop are tuned empirically to achieve effective temperature regulation performance.

# Functionality

A. Temperature Sensing:

The internal temperature sensor embedded within the RP2040 microcontroller serves as the primary means of gauging the ambient room temperature. This sensor provides accurate and real-time temperature readings, forming the basis for the dynamic control of the room's climate.

B. PID Algorithm:

The heart of the Temperature controller fan is the PID algorithm, a sophisticated control technique. The Proportional, Integral, and Derivative components work in harmony to calculate the optimal fan speed required to bring the room temperature closer to the desired setpoint. This ensures a responsive and stable system that adapts to varying environmental conditions.

C. PWM Fan Speed Control:

I. Dynamic Fan Speed Adjustment:

PWM modulation is employed to dynamically adjust the fan speed. The PID algorithm output is translated into PWM duty cycles, allowing precise control over the fan's rotational speed. This ensures a proportional and energy-efficient response to temperature changes.

II. Energy Efficiency:

The PWM-based fan speed control contributes to the project's energy efficiency. By modulating the fan's operation rather than relying on fixed-speed settings, the Room Guardian optimizes energy consumption while maintaining optimal room conditions.

III. Real-time Feedback:

Users can monitor the system's performance through real-time feedback displayed on a connected terminal. This feedback includes PID variables, fan speed, and current room temperature, providing transparency into the Controller’s operation.

# Usage Instructions:

1. Circuit Setup:

Ensure all components are connected as per the provided circuit diagram. This typically involves connecting the temperature sensor, fan, and any other necessary components to the appropriate GPIO pins on the Raspberry Pi Pico.

2. Power Supply:

Power on the 12V power supply to activate the entire system. Make sure the power supply is properly connected, and there are no loose connections.

3. Observe PID Variables and Resulting Fan Speed/Temperature:

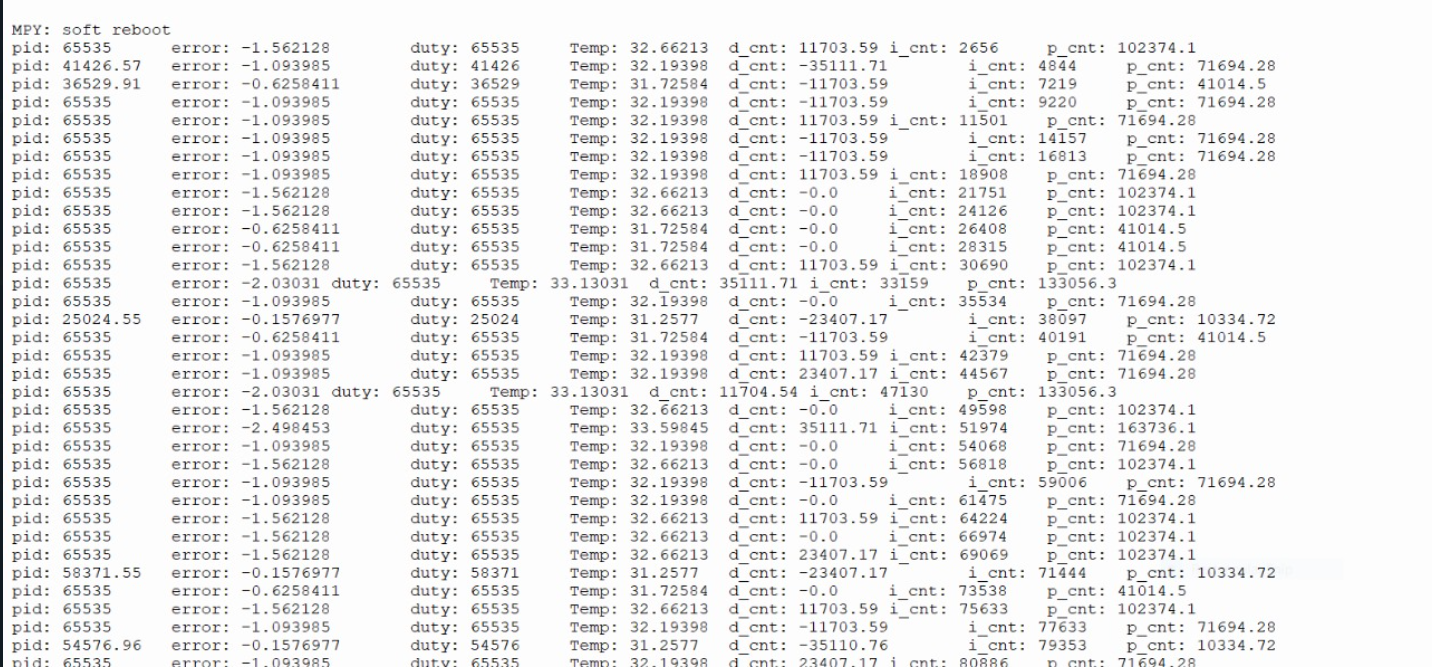
Monitor the terminal output for information related to the PID variables (Proportional, Integral, Derivative) and how they influence the fan speed and room temperature. This will give insights into the control loop's behavior.

# RESULTS AND ANALYSIS:

Thonny Python Output Visualization:

Terminal Output Display:

We have utilized Thonny Python as the terminal output display for the Controller. Showcase the real-time feedback provided by the system, including PID variables, fan speed, and current room temperature.



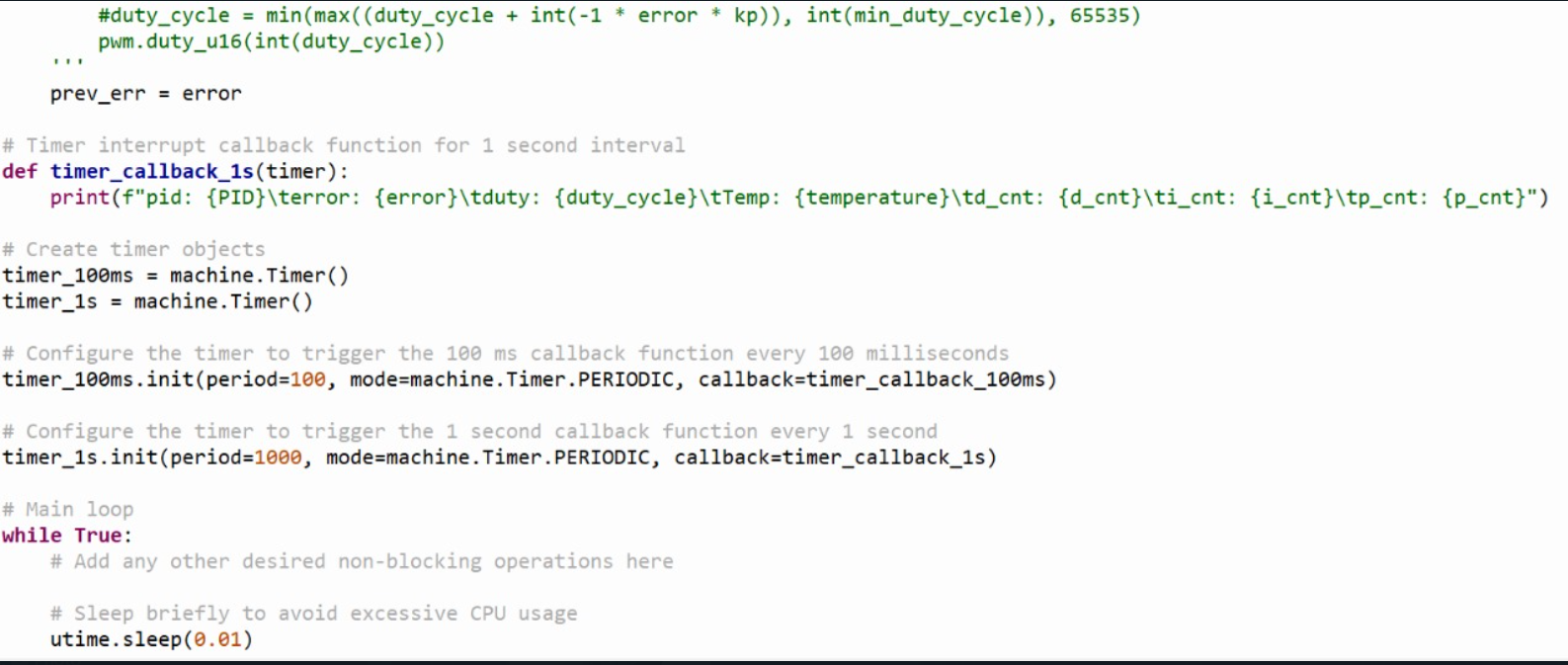


Fig. Code snippet

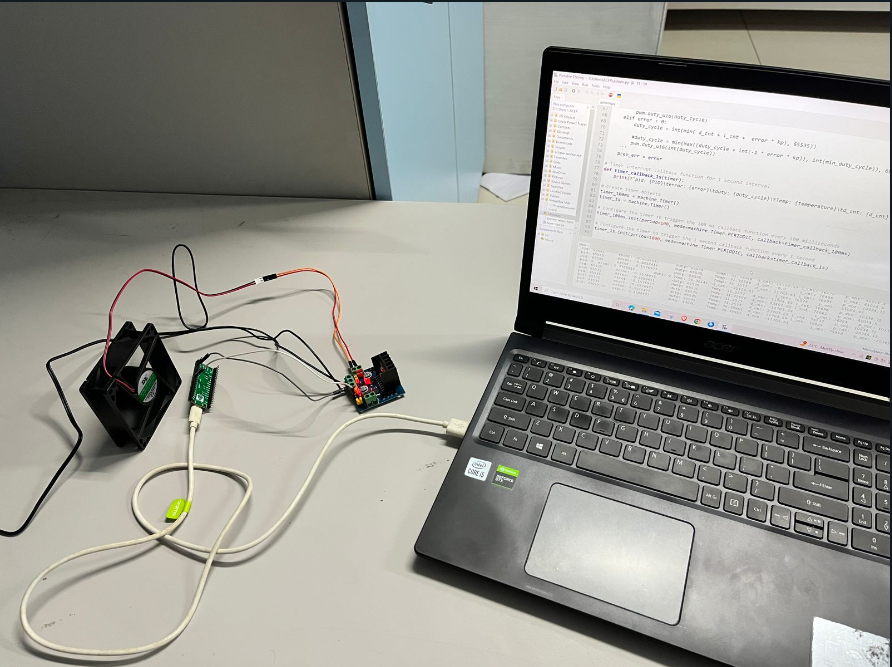


Fig. This image shows the complete circuit connection of the project

# Conclusion

In conclusion, the Room temperature fan controller project, powered by the Raspberry Pi Pico and driven by the RP2040 microcontroller, stands as an innovative solution to automate room temperature regulation. Leveraging the internal temperature sensor, this project seamlessly integrates a straightforward yet effective demonstration of the PID algorithm.

The PID algorithm, serving as the project's control mechanism, dynamically adjusts the fan speed through precise manipulation of the PWM duty cycle. This responsive and intelligent system ensures that as the room temperature fluctuates, the fan provides proportional cooling.

The simplicity of the setup not only highlights the efficiency of the Raspberry Pi Pico and RP2040 microcontroller but also underscores the accessibility of implementing advanced control techniques. This project not only showcases the potential of microcontroller-based solutions for home automation but also provides a practical illustration of how smart technology can enhance everyday living.

In essence, The temperature controller serves as a testament to the fusion of hardware capabilities and intelligent algorithms, presenting a tangible example of how automation can be harnessed for the betterment of our living spaces. The utilization of the PID algorithm for temperature regulation is not just a technological feat but a step towards creating more responsive and energy-efficient systems that cater to the evolving needs of modern living.

# Future Scope

1. LCD Display Integration:

Enhance the user interface by incorporating an LCD display to provide real-time feedback and information. The display can show the current room temperature, setpoint, and fan speed. This addition not only adds a visual element to the system but also offers users immediate insights into the status of their room environment.

2. WiFi Connectivity for Remote Control and Monitoring:

Integrate WiFi connectivity to enable remote control and monitoring of the Room Guardian. This feature allows users to adjust temperature settings and fan speed through a dedicated mobile app or web interface. Additionally, it provides the convenience of checking the room temperature and system status from anywhere with an internet connection.

3. Temperature Data Logging Over Time:

Implement a data logging feature to record temperature data over time. This historical data can be stored on an SD card or transmitted to a cloud server. Analyzing this data can provide valuable insights into temperature trends, system performance, and energy consumption patterns. Users can visualize the recorded data through graphs or charts, fostering a deeper understanding of their room climate.

4. Self-Tuning PID Algorithm:

Take the Temperature controller fan to the next level by implementing a self-tuning PID algorithm based on the collected temperature data. The system can analyze historical data to automatically adjust PID parameters, optimizing the control strategy for better performance and responsiveness. This self-tuning capability ensures that the Room Guardian adapts to changing environmental conditions and continuously improves its efficiency over time.

# Softwares Used

## Thonny

* *Usage:* It is used to monitor the output