Smart Radiator Thermostat Project

1. Project Description

The goal of this project is to develop a smart thermostat that can automatically control a traditional radiator.

The system will measure the room temperature, allow users to set a target temperature, and adjust the radiator valve accordingly using a motor.

A built-in display will show real-time temperature data and allow users to configure settings, including scheduled temperature changes at specific times.

One of the key challenges will be designing a universal motor attachment mechanism that can work with different radiator valve types.

Additionally, since turning the valve does not directly correspond to a fixed temperature change, the system will need an auto-calibration feature.

2. Required Hardware & Costs

- Microcontroller: LilyGO TTGO T3 LoRa32 868MHz V1.6.1 ESP32 (21.50 EUR)
- Temperature Sensor: DHT22 or DS18B20 (5.00 - 10.00 EUR)
- Servo Motor or Stepper Motor (for valve control): (5.00 - 15.00 EUR)
- Display: OLED or LCD screen (10.00 - 20.00 EUR)
- Buttons/Rotary Encoder (for setting temp & time): (3.00 - 7.00 EUR)
- RTC (Real-Time Clock) Module: DS3231 (3.00 - 8.00 EUR)
- Power Supply: 5V/12V adapter or battery (5.00 - 15.00 EUR)
- 3D-printed or adjustable mount for the radiator valve (if needed): (10.00 - 30.00 EUR)
- Breadboard, jumper wires, resistors, etc.: (10.00 EUR)

Estimated total cost: 50 - 120 EUR (depending on component choices)

3. Explanation of the Project

1. Temperature Measurement & Display

- The system continuously reads temperature data from a sensor (DHT22/DS18B20) and displays it on an OLED/LCD screen.

2. User Input & Configuration

- The user can set a desired temperature using buttons or a rotary encoder.
- A timer function allows scheduling temperature changes (e.g., "Set temperature to 22C at 8:00 AM").

3. Radiator Valve Control

- A servo/stepper motor is attached to the radiator valve and adjusts it based on the temperature difference.
- The system will perform auto-calibration by detecting the valve's range of motion and mapping it to temperature changes.

4. Communication & Processing

- The ESP32 microcontroller processes the temperature data and user inputs.
- It controls the motor based on predefined logic, adjusting the radiator valve to maintain the target temperature.
- A real-time clock (RTC) module keeps track of time for scheduled temperature adjustments.

5. Power & Hardware Integration

- The system is powered via a 5V/12V adapter or a battery.
- The components will be mounted in a casing that can be attached to a radiator.

4. Key Challenges & Solutions

- Universal Fit: The motor attachment must be adjustable to fit different radiator types.
  Possible solutions include 3D-printed mounts or adjustable clamps.
- Auto-Calibration: The system needs to determine how much the valve should be turned to reach the desired temperature.
  This may require an initial learning phase.
- Power Efficiency: If battery-powered, the system should enter a low-power mode when not actively adjusting the radiator.

5. Learning Outcomes

- Integration of sensors, motors, and displays in an embedded system.
- Implementation of user input handling and scheduling functions.
- Development of an auto-calibration algorithm for precise control.
- Practical experience in mechanical mounting solutions for electronics.