Room Temperature Based Fan Speed Control System

Prepared by: DEERAJ V

Roll Number: 23EE011

Branch: B.E/EEE

Abstract

This project demonstrates a simple and effective embedded control system that regulates fan speed based on ambient room temperature using an LM35 temperature sensor and Arduino, with its execution and simulation carried out in Scilab. The system automatically increases the fan speed as temperature rises, ensuring energy efficiency and comfort. This prototype serves as an excellent learning tool for understanding analog sensors, pulse-width modulation (PWM), and temperature-based control systems using Scilab for analysis and visualization.

Introduction

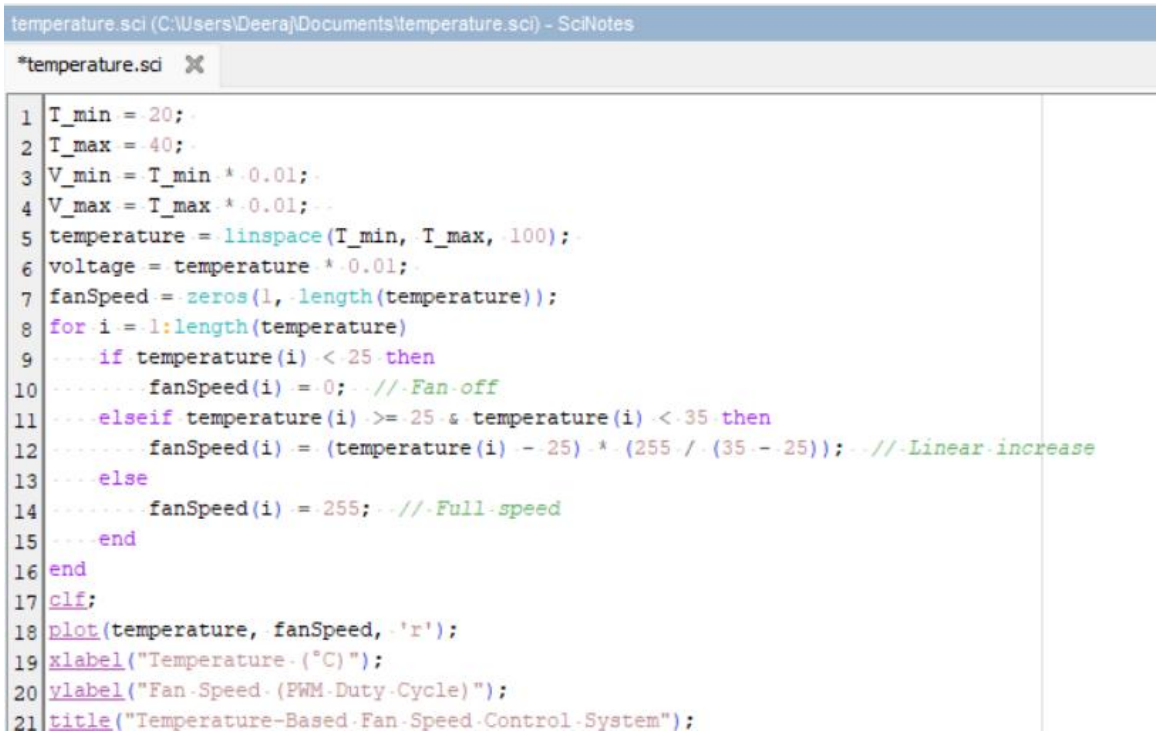
In many residential and industrial applications, controlling fan speed manually can be inefficient. Automating this process based on temperature can significantly improve convenience and energy savings. This project uses an LM35 temperature sensor to detect room temperature, and a transistor-controlled DC fan whose speed is modulated by Arduino's PWM output.

This project provides hands-on experience with embedded hardware interfacing, control logic, and basic electronics.

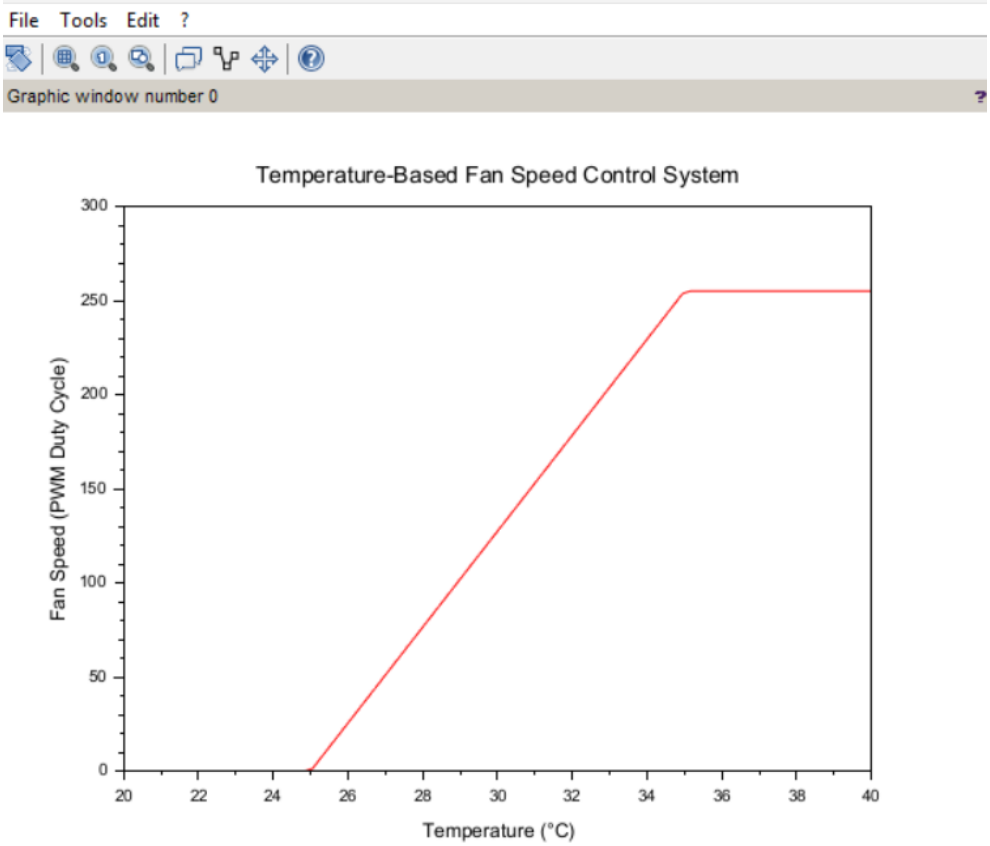
Working Principle

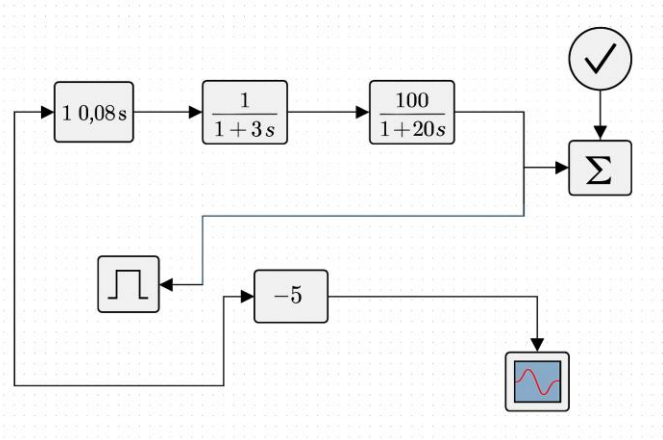
- The LM35 sensor outputs an analog voltage proportional to the temperature (10 mV/°C).
- Arduino reads this analog voltage, converts it into temperature, and controls the fan speed using PWM.
- If the temperature is below 25°C, the fan remains off. Between 25°C to 35°C, the speed increases linearly. Beyond 35°C, the fan operates at full speed.

SciLab Code


temperature.sci (C:\Users\Deeraj\Documents\temperature.sci) - SciNotes
*temperature.sci
T_min = 20;
T_max = 40;
V_min = T_min * 0.01;
V_max = T_max * 0.01;
temperature = linspace(T_min, T_max, 100);
voltage = temperature * 0.01;
fanSpeed = zeros(1, length(temperature));
for i = 1:length(temperature)
if temperature(i) < 25 then
fanSpeed(i) = 0; // Fan off
elseif temperature(i) >= 25 & temperature(i) < 35 then
fanSpeed(i) = (temperature(i) - 25) * (255 / (35 - 25)); // Linear increase
else
fanSpeed(i) = 255; // Full speed
end
end
clf;
plot(temperature, fanSpeed, 'r');
xlabel("Temperature (°C)");
ylabel("Fan Speed (PWM Duty Cycle)");
title("Temperature-Based Fan Speed Control System");

Simulation and Output





Applications

- Smart home automation
- Industrial HVAC systems

- IoT-based temperature control
- Embedded systems education and projects

Future Scope

- Replace DC fan with AC fan and use TRIAC for control.
- Add an OLED or LCD screen to show live temperature.
- Integrate with IoT platforms like Blynk or Firebase.
- Implement over-temperature protection alerts (buzzer/SMS).
- Enclose in a casing for real-world deployment.

Appendix

- **Step Input Block** – Provides temperature input to simulate environmental changes.
- **Gain Block** – Adjusts signal strength for processing.
- **Summation Block** – Combines input signals for control logic.
- **Transfer Function Block** – Models the dynamic behavior of the system.
- **Check Simulation Block (Validation)** – Validates output and ensures correct response.
- **Feedback Lines / Connectors** – Links the blocks to create a closed-loop system.

Conclusion and Application

This project successfully simulates a **Room Temperature-Based Fan Speed Control System** using **SciLab**, demonstrating how temperature variations can be regulated through **sensor-based feedback and control mechanisms**. The simulation provides a clear understanding of **embedded system dynamics and automation** in response to environmental changes.

Applications

• **Smart Home Automation** – Automatically adjusts fan speed based on room temperature.

• **Energy Efficiency Optimization** – Reduces unnecessary power consumption by controlling fan speed dynamically.

• **Educational Tool for Embedded Systems** – Helps students understand **sensor interfacing, PWM control, and feedback mechanisms**.

• **Industrial HVAC Systems** – Can be extended for **large-scale temperature regulation applications**.

• **IoT-Based Remote Monitoring** – Can integrate with platforms like **Blynk or Firebase** for **smart control and data analysis**.