Temperature Monitoring and Control System using ATmega328P and LM35

Abstract

This project presents the design and implementation of a real-time temperature monitoring and controlling system using the ATmega328P microcontroller and LM35 temperature sensor. The LM35 provides an analog voltage proportional to temperature which is converted into digital data using the internal ADC of the microcontroller. The system continuously monitors ambient temperature and automatically activates a cooling device (fan/relay) when the temperature exceeds a predefined threshold. The circuit was designed using KiCad and simulated in Proteus to verify real-time performance and reliability. The system provides an economical and efficient solution for environmental monitoring applications such as industrial automation, server rooms, and home appliances.

Objectives

- To measure temperature using LM35 analog temperature sensor
- To convert analog voltage into digital temperature data using ADC
- To display and monitor temperature in real time
- To automatically control a fan/relay based on temperature threshold
- To design circuit schematic and PCB layout using KiCad
- To simulate the system using Proteus software

Components Required

Component	Quantity	Purpose
ATmega328P / Arduino UNO	1	Microcontroller
LM35 Temperature Sensor	1	Temperature sensing

Component	Quantity	Purpose
Relay module / Fan	1	Temperature control
LCD 16x2 (optional)	1	Display temperature
Resistors	Few	Current limiting
Capacitors	Few	Noise filtering
Power supply 5V	1	Circuit operation

Block Diagram

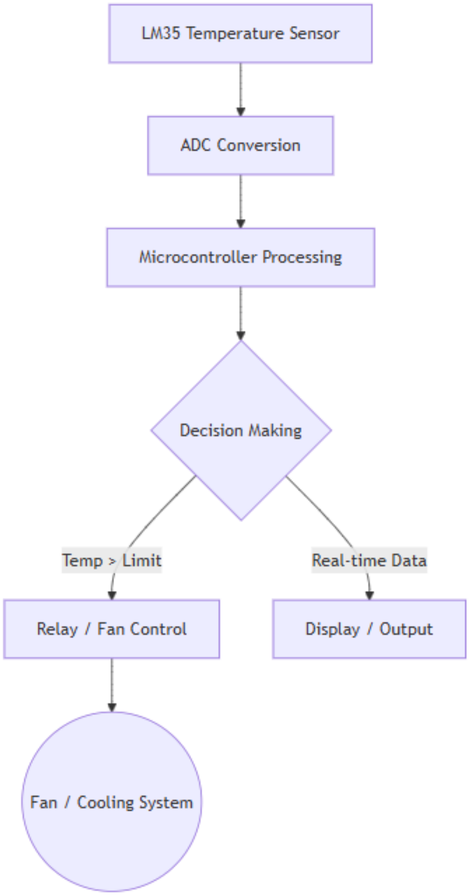

fig.Block Diagram

Working Principle

The LM35 sensor produces an output voltage linearly proportional to temperature at a scale factor of 10 mV/°C.

$$Temperature(°C) = V_{out} \times 100$$

The ATmega328P reads this voltage through its 10-bit ADC:

$$ADC\ Value\ =\ \frac{V_{in}}{5V} \times 1023$$

The microcontroller converts the ADC value into temperature and compares it with a predefined threshold.

- If Temperature > Threshold → Relay ON (Cooling Activated)
- If Temperature ≤ Threshold → Relay OFF

Schematic diagram

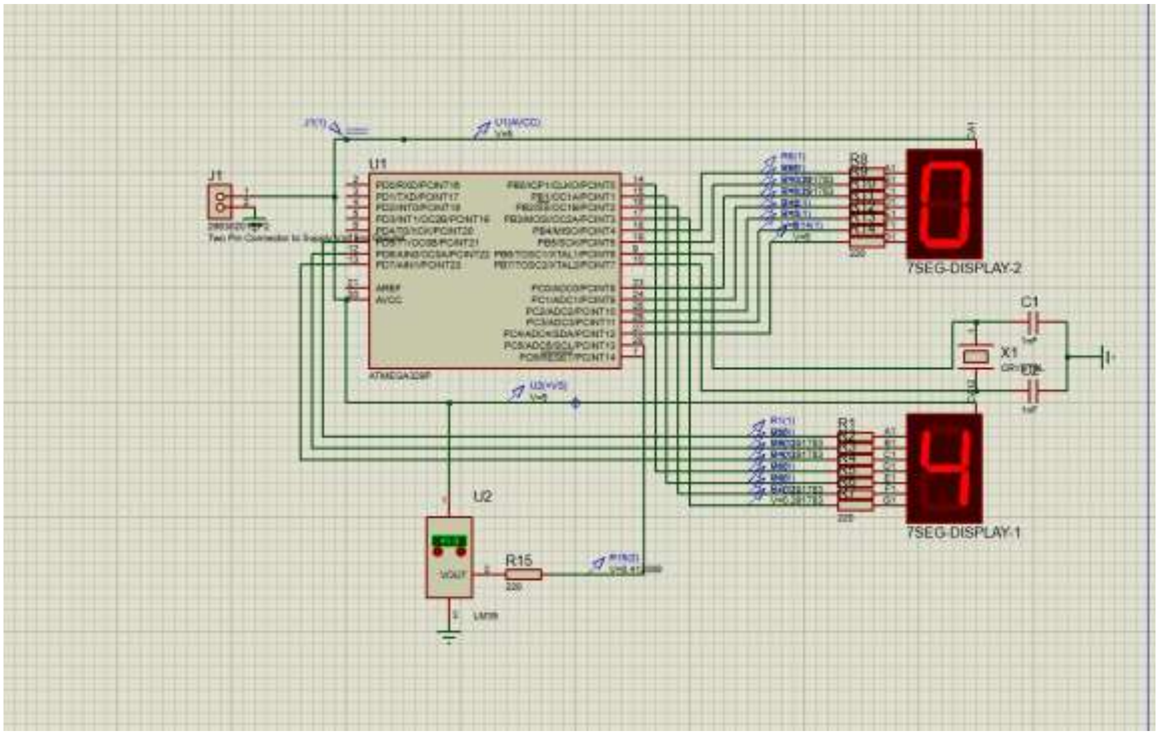

fig . Schematic diagram

Software Description

The firmware is developed using Arduino IDE in Embedded C.
It performs three major operations:

1. ADC reading from LM35
2. Temperature calculation
3. Automatic control of cooling device

Algorithm

1. Start system
2. Read analog voltage from LM35
3. Convert ADC value into temperature
4. Compare temperature with threshold
5. If temperature high → turn ON fan
6. Else → turn OFF fan
7. Repeat continuously

Flowchart

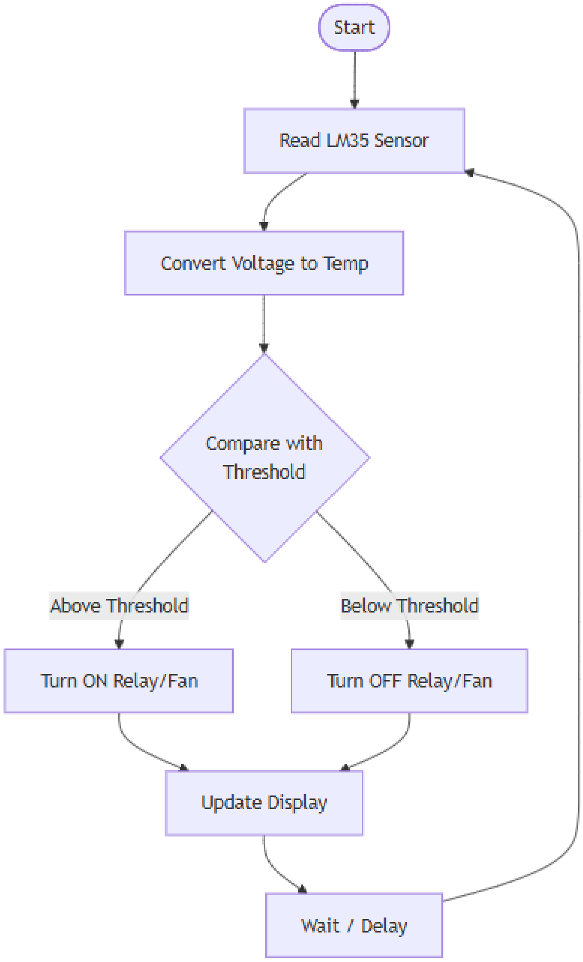

Fig.flow chart

PCB Design

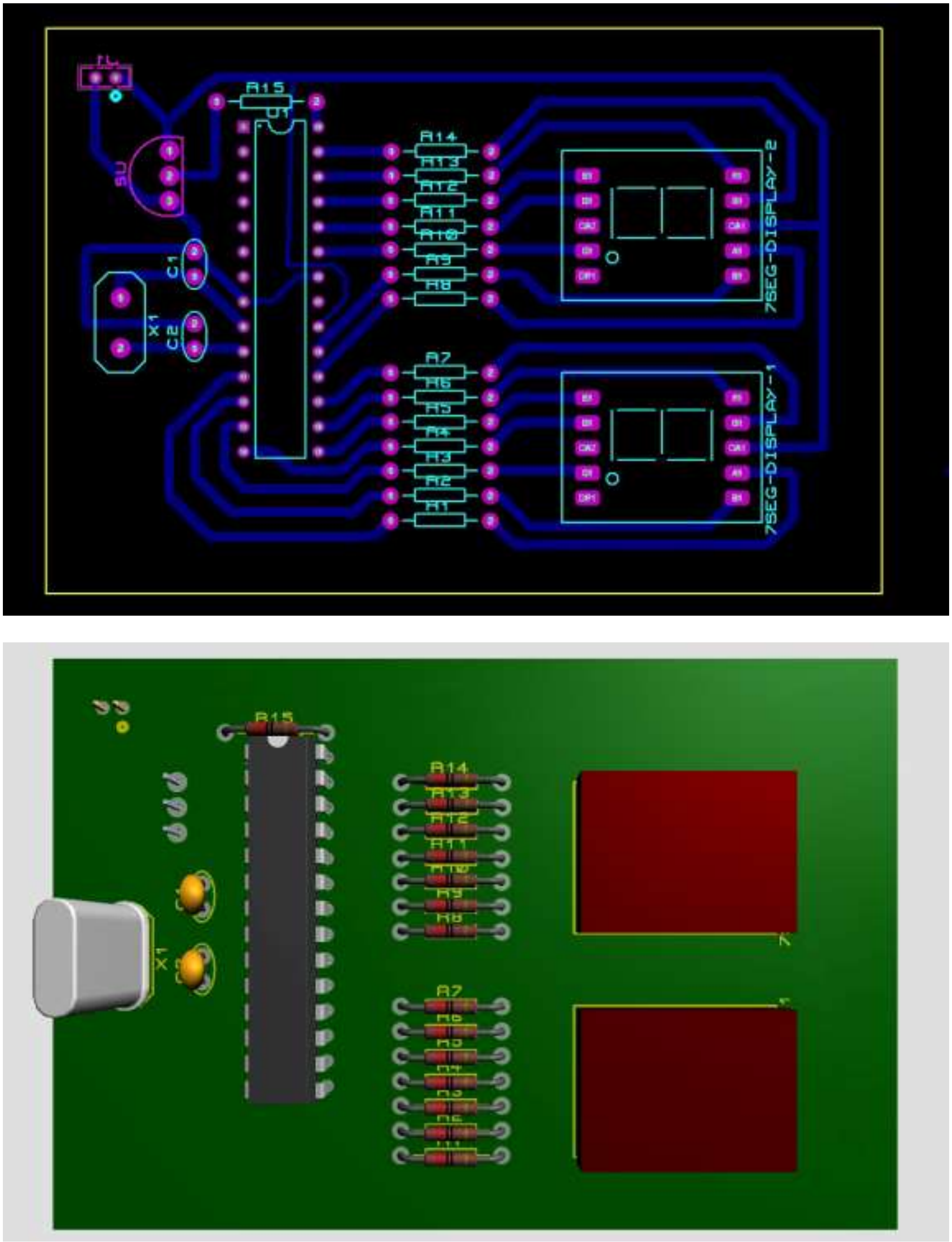

Fig.PCB design

Applications

- Industrial temperature monitoring
- Server room cooling control
- Smart home automation
- Incubators and storage systems
- HVAC systems

Advantages

- Low cost and simple design
- Accurate measurement
- Real-time automatic control
- Low power consumption
- Easy to implement

Limitations

- Limited sensing distance (local measurement)
- Requires calibration for high precision environments
- Single parameter monitoring

Future Scope

- IoT cloud monitoring using ESP8266/ESP32
- Mobile app notification system
- Multiple sensor network monitoring
- Automatic heater + cooler combined system

Conclusion

The temperature monitoring and control system was successfully designed and implemented using the ATmega328P microcontroller and LM35 sensor. The system accurately measures temperature and automatically controls a cooling device based on environmental conditions. The project demonstrates reliable real-time performance, low cost, and suitability for industrial and domestic automation applications.

References

- ATmega328P Datasheet
- LM35 Temperature Sensor Datasheet
- Arduino IDE Documentation
- Proteus Design Suite Manual
- KiCad PCB Design Guide