Embedded Systems Design Document

Project: Basic Heater Control System

Applicant: Ranveer Singh Shekhawat

Date: August 2025

About the Project

This document outlines the design of a simple Heater Control System that automates heater actuation based on environmental temperature using a temperature sensor and microcontroller logic. The system is part of a simulation-based evaluation for upliance.ai, India's first AI-powered cooking assistant company.

1. Minimum Sensors Required

Temperature Sensor (Essential)

- Recommended: DHT22 / LM35 / DS18B20

- Function: Measures ambient temperature in real-time

- Justification: Critical to determine when to activate and deactivate the heater based on threshold logic.

DHT22

1. General Overview

- The **DHT22** is a low-cost digital sensor for measuring **temperature** and **humidity**.
- It offers higher accuracy and a wider range than its sibling, the DHT11.

2. Technical Specifications

Feature	**Specification**
Temperature Range	-40°C to +80°C
Temperature Accuracy	±0.5°C
Humidity Range	0% to 100% RH
Humidity Accuracy	±2–5% RH
Sampling Rate	0.5 Hz (1 reading every 2 seconds)
Supply Voltage	3.3V to 6V
Data Output	Digital (single-wire proprietary protocol)

Feature	Specification
Power Consumption	Low (static current ~1–1.5 mA)

Optional for Expansion

- Current Sensor (e.g., ACS712): To detect heater load draw.

- Smoke Sensor (e.g., MQ2): For overheating or fire detection in extended versions.

- Humidity Sensor: Could be used for climate-specific heating control in future versions.

2. Recommended Communication Protocol

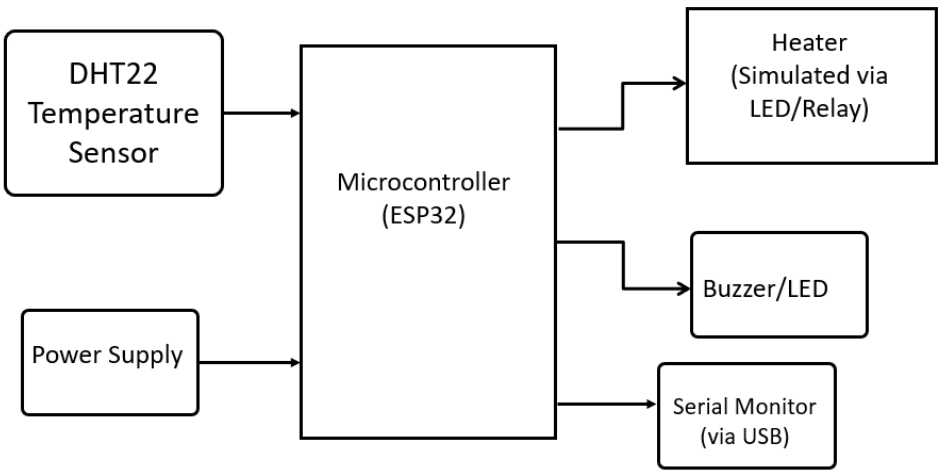
Protocol: I^2C (Inter-Integrated Circuit)

Why I^2C?

Factor	Justification
Simplicity	Requires only 2 wires: SDA (data), SCL (clock)
Multi-sensor support	Easily integrates multiple sensors/devices
Power efficiency	Suitable for low-power embedded systems
Common support	Supported by most microcontrollers (ESP32, Arduino, etc.)

In case of only one sensor, 1-Wire (used by DS18B20) is also acceptable, but I^2C is scalable.

3. Block Diagram



4. Future Roadmap

a) Overheating Protection

- Add a smoke sensor (MQ2) to detect dangerous levels of heat or burning.

- Add a temperature ceiling limit, beyond which the system forcibly disables the heater and raises an alert (e.g., buzzer or visual warning).

b) Multiple Heating Profiles

- Use EEPROM / Flash Memory to store user-defined heating profiles (e.g., Eco, Normal, Boost).

- User Interface via:
  - OLED display + buttons (local UI)
  - Bluetooth/Web dashboard (remote control)

- Profiles can define:
  - Temperature range
  - Runtime limits
  - Hysteresis margins

c) Connectivity & AI Integration

- Add Wi-Fi (via ESP32) to support:
  - Remote monitoring
  - AI-based heating schedule predictions

- Integrate with upliance.ai platform for advanced automation.

Summary

Element	Details
Minimum Sensor	DHT22 / LM35
Communication Protocol	I^2C (recommended), 1-Wire (optional)
Core Components	Sensor → Microcontroller → Heater
Expandable Features	Overheating detection, profiles, IoT

Link: https://wokwi.com/projects/438153436114816001