

Heater Control System - Arduino Implementation

This project demonstrates a simple heater control system using Arduino, simulating heating using a temperature sensor (TMP36 or DS18B20), visual feedback (LEDs), and safety mechanisms like a buzzer and relay logic for overheat protection.

Components Used

- Arduino UNO - Main controller
- TMP36 / DS18B20 - Temperature sensors
- 5 LEDs - State indicators (Blue, Green, Yellow, Orange, Red)
- Buzzer - Overheat alert

Minimum Sensors Required

- Temperature Sensor (e.g., TMP36 or DS18B20): Required to measure ambient or system temperature.
- Optional Overheat Sensor: A secondary temperature sensor to measure heater surface or ambient safety thresholds.
- Future Option: Ambient temperature sensor to adjust heating profile based on environmental conditions.

Recommended Communication Protocol

For Arduino-based systems, analog input or One-Wire (for DS18B20) is used. For scalable or ESP32-based systems, I2C is recommended for multi-sensor support. One-Wire is preferred for DS18B20 due to its minimal wiring and simplicity.

System States

- IDLE: Blue LED - System is powered but not heating.
- HEATING: Green LED - Heater is ON and temperature is rising.
- STABILIZING: Yellow LED - Near target temperature.

- TARGET REACHED: Orange LED - Heater OFF, temp maintained.
- OVERHEAT: Red LED + Buzzer - Safety cutoff triggered.

Simulation Process

The development was done in two phases:

- Phase 1: TinkerCAD for initial logic and schematic simulation.
- Phase 2: Wokwi for full component integration and testing.

Simulation Links:

TinkerCAD: <https://www.tinkercad.com/things/bLwauZPdw8U-tmp36-temperature-sensor-with-arduino>

Wokwi: <https://wokwi.com/projects/430372342260459521>

GitHub Repository:

<https://github.com/sarthak30102003/Basic-Heater-Control-System>

Future Roadmap

- Port to ESP32 using ESP-IDF.
- Add BLE advertisement of heating state.
- Implement FreeRTOS for non-blocking multitasking.
- Integrate real sensor feedback instead of simulated variable.

Schematic

