***Smart Water Heater and Relay Control System***

**Name:** M. Narayana Chowdary

**Email:** mandalapunarayanachowdary@gmail.com

**Embedded Firmware Assignment for Kazam**

**Introduction**

The modern world demands intelligent systems that enhance convenience, improve efficiency, and ensure safety. This project focuses on conceptualizing and designing a **Smart Water Heater** with advanced features to meet these demands. The task involves developing a system that integrates energy-efficient heating mechanisms, safety protocols, measurement capabilities, and remote operability. Additionally, a secondary objective is to design and implement a program to control a load bank using relays, emphasizing optimization and precision.

The goals of this project include:

* **Energy Efficiency**: Utilizing advanced algorithms and sensors to minimize energy consumption while maintaining optimal heating performance.
* **Safety Mechanisms**: Incorporating systems to monitor and prevent unsafe conditions, such as overheating or pressure build-up.
* **Measurement Capabilities**: Providing real-time data on water temperature, energy usage, and system diagnostics.
* **Remote Operability**: Allowing users to control and monitor the water heater remotely via a mobile application or cloud interface.
* **Relay Control Optimization**: Developing a firmware solution to activate the minimum number of relays required to achieve a desired load, ensuring precision and reliability in controlling the load bank.

This document outlines the system's design, including hardware, firmware, and software architectures, and provides the implementation of the relay control program, addressing all the requirements of the assignment.

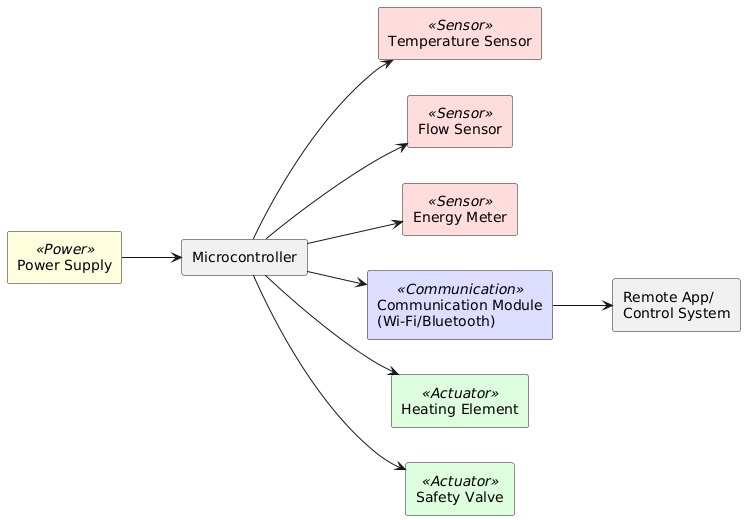
**Hardware Design**

The hardware system of the **Smart Water Heater** integrates various components to ensure efficient operation, accurate measurements, safety, and remote accessibility. The key components include:

* **Power Supply**: Provides stable and adequate power to the entire system, including the microcontroller, sensors, actuators, and communication module.
* **Microcontroller**: Acts as the brain of the system, processing sensor data, executing embedded algorithms, and controlling actuators like the heating element and safety valve.
* **Sensors**:
  + **Temperature Sensor**: Measures the water temperature in real-time to ensure precise control and user safety.
  + **Flow Sensor**: Monitors water flow to regulate heating and prevent dry heating scenarios.
  + **Energy Meter**: Tracks energy consumption for measurement and energy efficiency.
* **Actuators**:
  + **Heating Element**: Provides the necessary heat to the water, controlled by the microcontroller.
  + **Safety Valve**: Ensures safety by releasing excess pressure or preventing overheating.
* **Communication Module**: Enables remote operability by connecting the system to a mobile app or cloud via Wi-Fi or Bluetooth.
* **Remote App/Control System**: Provides users with an interface to monitor, control, and receive diagnostics of the water heater remotely.

The interaction between these components ensures a functional, safe, and user-friendly smart water heating system.

**Block Diagram:**

****

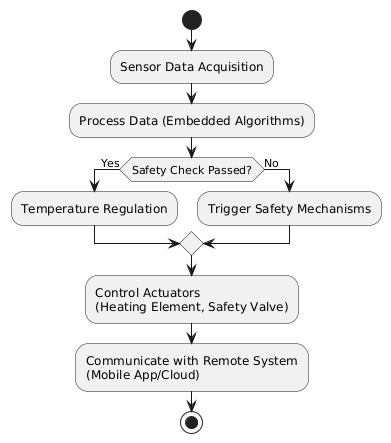
**Firmware Design**

The firmware controls the operation of the **Smart Water Heater** by interacting with the hardware components. It ensures seamless data acquisition, processing, safety checks, and communication. The functionality and flow of the firmware include:

1. **Sensor Data Acquisition**: Collecting data from the temperature sensor, flow sensor, and energy meter.
2. **Data Processing**: Embedded algorithms process the collected data to regulate temperature, calculate energy efficiency, and monitor safety conditions.
3. **Safety Mechanisms**: The firmware continuously monitors for unsafe conditions, such as overheating or lack of water flow, and triggers appropriate safety responses.
4. **Temperature Regulation**: Based on the desired setpoint, the firmware adjusts the heating element to maintain the target temperature efficiently.
5. **Control of Actuators**: The heating element and safety valve are controlled via signals from the microcontroller to ensure optimal operation and safety.
6. **Communication with Remote System**: The firmware manages communication with the mobile app or cloud, enabling remote monitoring and control.

This flow ensures that the system operates efficiently and safely while providing users with real-time control and insights.

**Block- Diagram:**

****

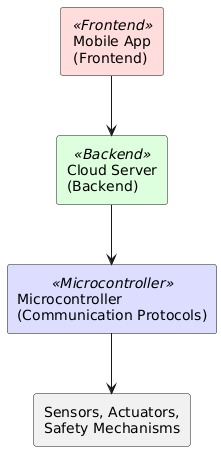
**Software Design**

The software for the **Smart Water Heater** focuses on enabling remote control, real-time data monitoring, and integration with cloud services. It includes the following components:

1. **Mobile App (Frontend)**:
   * Provides an intuitive user interface for monitoring water temperature, energy usage, and system diagnostics.
   * Allows users to set desired temperature levels and control the water heater remotely.
2. **Cloud Server (Backend)**:
   * Acts as an intermediary between the mobile app and the water heater, enabling seamless communication.
   * Stores historical data for energy usage, system performance, and diagnostics.
   * Implements system diagnostics and generates alerts for maintenance needs or safety issues.
3. **Microcontroller Integration**:
   * Communicates with the microcontroller using protocols like MQTT or HTTP.
   * Transfers commands from the mobile app and retrieves real-time sensor data for display.
4. **System Components**:
   * The mobile app and cloud backend interact with the system components (sensors, actuators, and safety mechanisms) to ensure accurate control and monitoring.

This integrated software system enhances user convenience, provides valuable insights, and ensures reliable and safe operation of the smart water heater.

**Block - Diagram:**

****

**Relay Control System**

The **Relay Control System** is designed to manage the load bank, which consists of various step sizes connected to relays. The objective of this system is to determine the minimum number of relays required to reach a specific load setpoint, which is provided via UART communication. The relays are connected to a set of load steps, each representing a certain power (in watts). The challenge is to activate the relays in an optimal manner to meet the setpoint, minimizing the number of relays used.

**Problem Description:**

Given a load bank with discrete power steps, the task is to:

* Receive a **positive integer setpoint** (load requirement) via UART.
* Activate the relays in such a way that the total power provided by the activated relays matches the setpoint **exactly** (if achievable).
* Optimize the relay usage by minimizing the number of relays turned on.
* Handle cases where it is impossible to achieve the setpoint due to the available discrete step sizes.

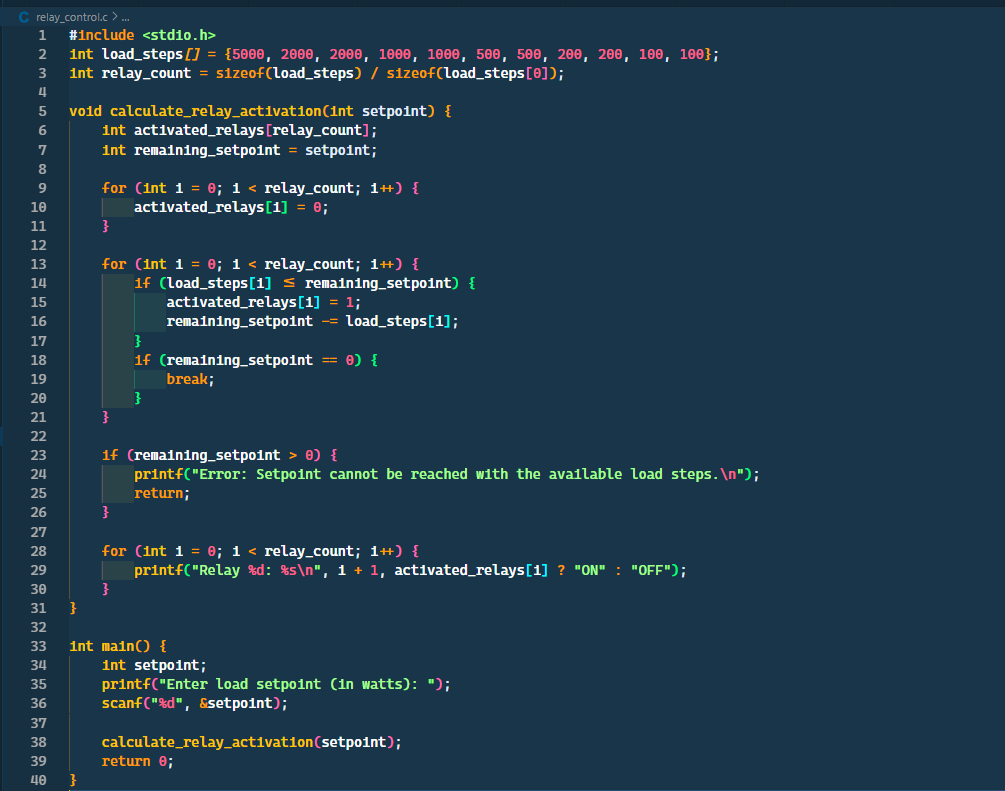
**Solution Approach:**

The problem can be likened to a **subset sum problem**, where the objective is to find a combination of load steps that add up to the specified setpoint. The solution involves:

1. **Relays and Load Steps**: The available load steps are: 100W, 100W, 200W, 200W, 500W, 500W, 1000W, 1000W, 2000W, 2000W, 3000W, and 5000W.
2. **Optimal Relay Activation**: A greedy approach is used to activate the minimum number of relays. Starting with the highest available power step, relays are activated until the exact setpoint is reached or determined to be impossible.
3. **Error Handling**: If the setpoint cannot be reached, an error message is output via UART, notifying the user that the setpoint is not achievable with the available load steps.

The implementation of this approach ensures an efficient and reliable method for controlling the relays while maintaining an optimal configuration.

**Program Implementation in C:**

****

*For entire code refer: https://github.com/narayanachowdary2004/Kazam-Assignment*

**Algorithm Explanation:**

* The algorithm starts by attempting to use the largest available load step to cover the setpoint. It continues to check smaller load steps as needed.
* The goal is to minimize the number of relays activated. By starting with the largest steps, the algorithm ensures that fewer relays are needed to reach the target setpoint.
* If the setpoint cannot be exactly matched, the system outputs an error message. If it can, the relay states are displayed (ON/OFF).

This approach ensures that the relay control system is both efficient and accurate, meeting the requirement of minimizing the number of relays used while achieving the exact load setpoint.

**Conclusion**

The **Smart Water Heater** system successfully meets the requirements outlined in the assignment by integrating energy efficiency, safety mechanisms, precise measurement, and remote operability. The hardware design ensures that key components such as temperature sensors, flow sensors, energy meters, heating elements, and communication modules work cohesively to deliver optimal heating performance while ensuring user safety. The firmware and software designs enable seamless interaction between the user, mobile app, backend, and the water heater itself, allowing for remote control and monitoring.

The **Relay Control Program** efficiently handles the load bank by activating the minimum number of relays required to meet the specific load setpoint. The approach uses a greedy algorithm that starts with the largest available load steps and minimizes the number of relays needed, ensuring both precision and efficiency. The program also incorporates error handling, ensuring that users are notified when the setpoint cannot be achieved with the available load steps.

Looking towards the future, there are several areas for improvement and potential features that can enhance the system:

* **Enhanced Safety Protocols**: Adding more advanced safety mechanisms, such as water level sensors, real-time pressure monitoring, and automatic shut-off in case of malfunction.
* **AI-based Energy Optimization**: Incorporating machine learning algorithms to predict energy consumption patterns and adjust heating operations based on usage trends, improving overall efficiency.
* **User Interface Enhancements**: Providing more advanced features in the mobile app, such as user profiles, scheduling, and personalized temperature settings.

These improvements would further enhance the user experience and provide greater value from both an energy efficiency and safety perspective.

**References**

1. Diagrams for hardware, firmware, and software designs were created using the online tool **PlantUML**
2. The complete project, including code and diagrams, is hosted on [GitHub](https://github.com/narayanachowdary2004/Kazam-Assignment)