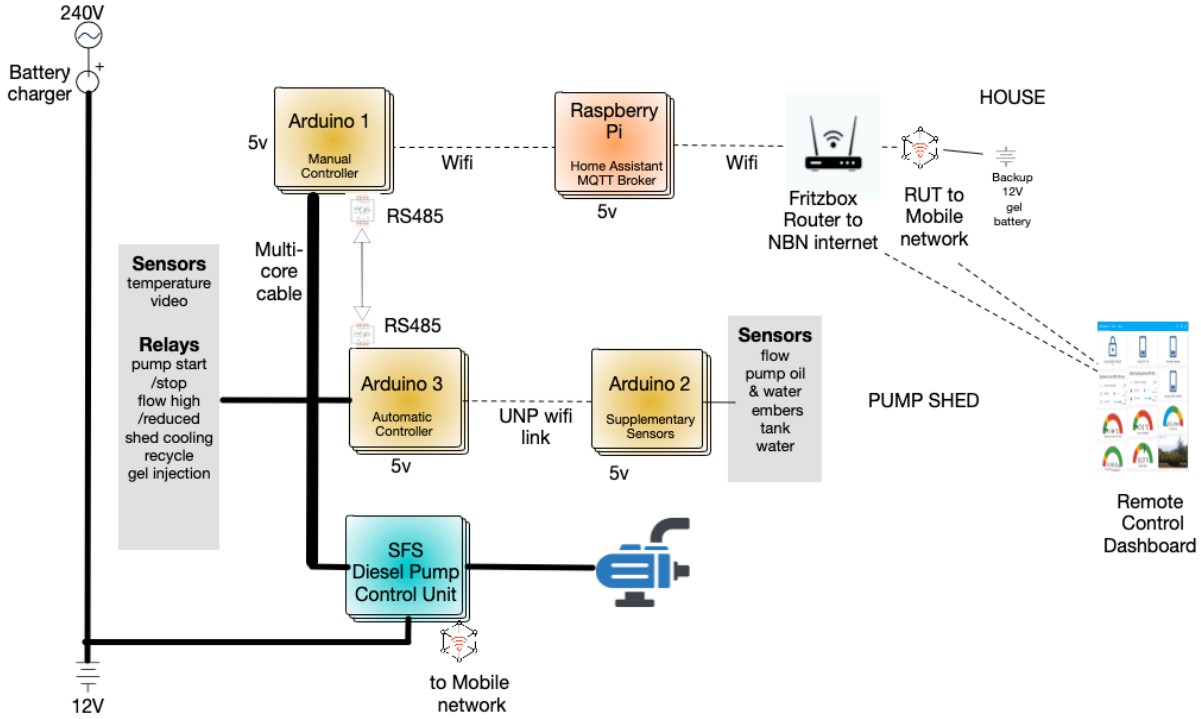
Software summary



The system is supported by a suite of four programs (ARD1, ARD2, ARD3 and RPI4) being one so-called ‘sketch’ for each Arduino, plus the software constructed for the Raspberry Pi.

The three Arduino sketches are tailored for specific tasks within an adaptive environmental management system, focusing on water management, peripheral control, and autonomous operation under different conditions. Here is a brief summary of each sketch's functionalities, the sensors they use, and how they interact within the system.

Sketch 1: Central Control and Peripheral Management (Arduino 1)

Functional Overview:

- Acts as the primary communication hub within the house, leveraging stable Wi-Fi connectivity.
- Manages peripheral devices through relay controls based on sensor inputs and external commands.

Sensors and Inputs:

- **Temperature Sensors:** Measures ambient temperatures to manage heating/cooling systems.
- **Pressure Switches:** Monitors oil and water pressure crucial for pump operations.
- **Ember Detector:** Provides early fire detection using an analog sensor input.

Control Actions:

- **Peripheral Relay Control:** Manages various relays for pumps and cooling systems, adjusting their states based on environmental data and preset conditions.
- **Heartbeat Signal:** Broadcasts a signal to Arduino 3 to monitor connectivity and control status.
- **MQTT Communications:** Sends and receives MQTT messages for system status updates and remote command execution.

Sketch Highlights:

- Implements various simulation modes for testing without actual sensor feedback.
- Utilizes EEPROM for storing state configurations to ensure persistence across reboots.

Sketch 2: Water Management and Flow Monitoring (Arduino 2)

Functional Overview:

- Dedicated to managing water levels and flow rates, crucial for effective water resource management.
- Located within the house to utilize the stable infrastructure shared with Arduino 1.

Sensors and Inputs:

- **Water Level Sensor:** Monitors water levels in tanks to prevent overflows or excessive depletion.
- **Flow Meter:** Calculates water inflow and outflow to maintain balance and detect anomalies.

Control Actions:

- **Data Reporting:** Regularly updates Arduino 1 with water level and flow data for broader system decisions.
- **Local Alarms:** Triggers alarms based on critical water levels or unexpected flow rates.

Sketch Highlights:

- Employs a simulation mode for flow and water level, allowing for testing without actual water usage.
- Integrates with MQTT for data communication, enhancing its role in the system-wide management.

Sketch 3: Autonomous Pump House Controller (Arduino 3)

Functional Overview:

- Operates independently in the pump house, ready to take control if connectivity with Arduino 1 is lost.
- Manages critical functions like pumping and emergency responses autonomously.

Sensors and Inputs:

- **External Temperature Sensors:** Monitors external temperatures for operational adjustments.
- **Pressure Sensors:** Similar to Arduino 1, used here for autonomous pump control.

Control Actions:

- **Autonomous Mode:** Engages when it stops receiving the heartbeat signal from Arduino 1, indicating a potential communication failure.
- **Pump and Peripheral Management:** Controls relays directly based on local sensor inputs and programmed logic.

Sketch Highlights:

- Monitors heartbeat signals to decide when to switch to autonomous control.
- Uses EEPROM for state persistence and configuration storage, ensuring reliability even after power disruptions.

System Integration and Communication

Data Sharing and Reliability:

- Arduino 1 and 2 share sensor data and operational commands to ensure coherent system behavior. This integration allows Arduino 1 to make informed decisions based on real-time water management data from Arduino 2.
- Arduino 3 acts as a fail-safe, ensuring critical operations continue even when the main communication links are down.

Operational Hierarchy:

- Under normal operations, Arduino 1 acts as the central node, processing data from Arduino 2 and controlling system-wide peripherals.
- In the event of a system failure or loss of connectivity, Arduino 3 takes over, ensuring that essential functions such as pumping and emergency cooling or heating continue without interruption.

Resilience and Redundancy:

- The autonomous capabilities of Arduino 3 are critical for maintaining operations during unexpected failures, showcasing the system's resilience.
- The use of EEPROM across devices enhances system reliability by preserving crucial configuration data.

In summary, these sketches collectively form what is designed as a robust, responsive, and resilient environmental management system capable of operating under various conditions, providing seamless control and ensuring continued operation through intelligent redundancy and autonomous capabilities. Some further programmatic details of each sketch follow below.