Johnnie Morris

Swamp Cooler Project: Technical Documentation

Overview

This project implements an evaporative cooling system (swamp cooler) using an Arduino Mega 2560 and various sensors and actuators. The system is designed to provide energy-efficient cooling by leveraging water evaporation to reduce temperature and increase humidity. Key features include:

- Monitoring air temperature and humidity.
- Displaying readings on an LCD.
- Controlling a fan motor based on temperature thresholds.
- Alerting the user when water levels are low.
- Logging system state changes with timestamps using a real-time clock (RTC).
- Implementing a state machine for operational management (DISABLED, IDLE, RUNNING, ERROR).

Design and Implementation

State Machine

The system transitions between four primary states:

1. **DISABLED**:
    - Yellow LED is ON.
    - No monitoring of sensors.
    - Transition to IDLE when the START button is pressed.
2. **IDLE**:
    - Green LED is ON.
    - Continuously monitors temperature, humidity, and water levels.
    - Transition to RUNNING if the temperature exceeds the upper threshold.
    - Transition to ERROR if water levels are low.
3. **RUNNING**:
    - Blue LED is ON.
    - Fan motor is active.
    - Transition to IDLE if temperature drops below the lower threshold.
    - Transition to ERROR if water levels are low.
4. **ERROR**:
    - Red LED is ON.
    - Displays error messages on the LCD.
    - Transition to IDLE if water levels normalize and RESET is pressed.
    - Transition to DISABLED if STOP is pressed.

Components and Functions

Sensors and Actuators

- **DHT11 Sensor**:
  - Measures temperature and humidity.
- **Water Level Sensor**:
  - Monitors water reservoir levels.
- **Fan Motor**:
  - Activates during the RUNNING state.
- **LED Indicators**:
  - Represent current system state.
- **Buttons**:
  - START, STOP, and RESET control transitions.

Real-Time Clock (RTC)

- Logs state transitions with timestamps.
- Outputs logs to the serial monitor for debugging and record-keeping.

LCD Display

- Continuously displays temperature and humidity readings during operation.
- Shows error messages when in the ERROR state.

Code Highlights

- **State Management**:
  - State transitions are implemented using an enum.
- **Event Logging**:
  - Timestamps recorded via RTC for each state change.
- **Interrupts**:
  - Used for START button to improve responsiveness.
- **Safety Checks**:
  - Water level and temperature thresholds ensure safe operation.

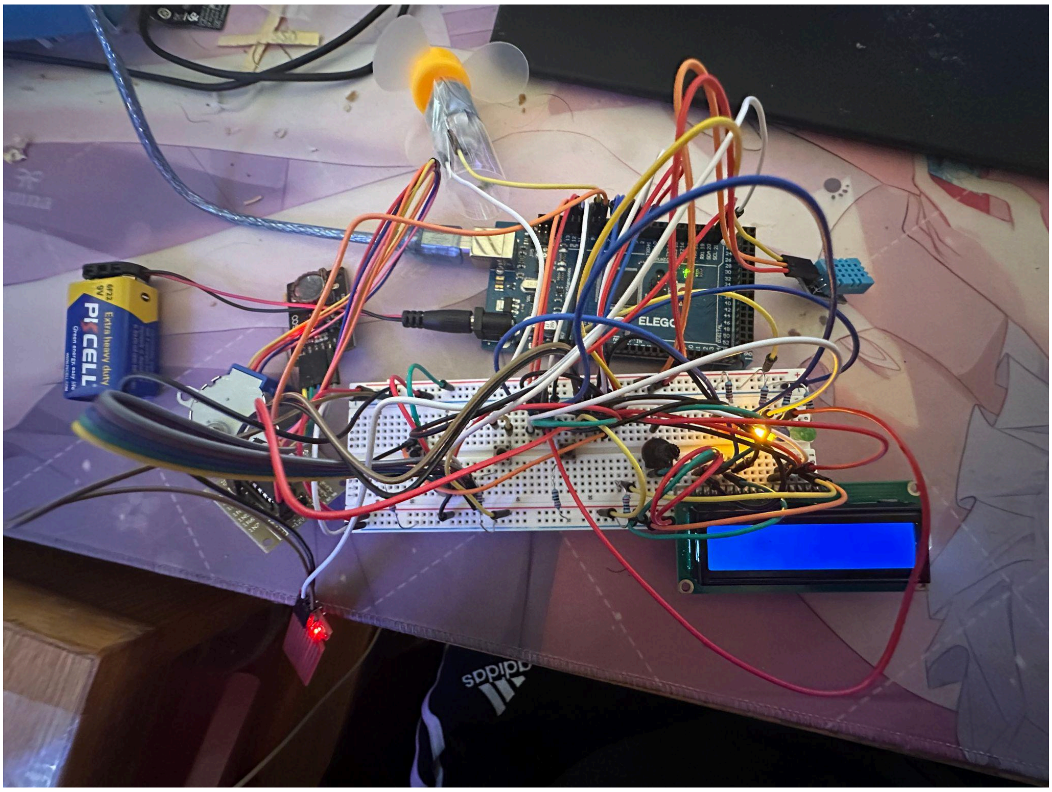
System Schematic and Components

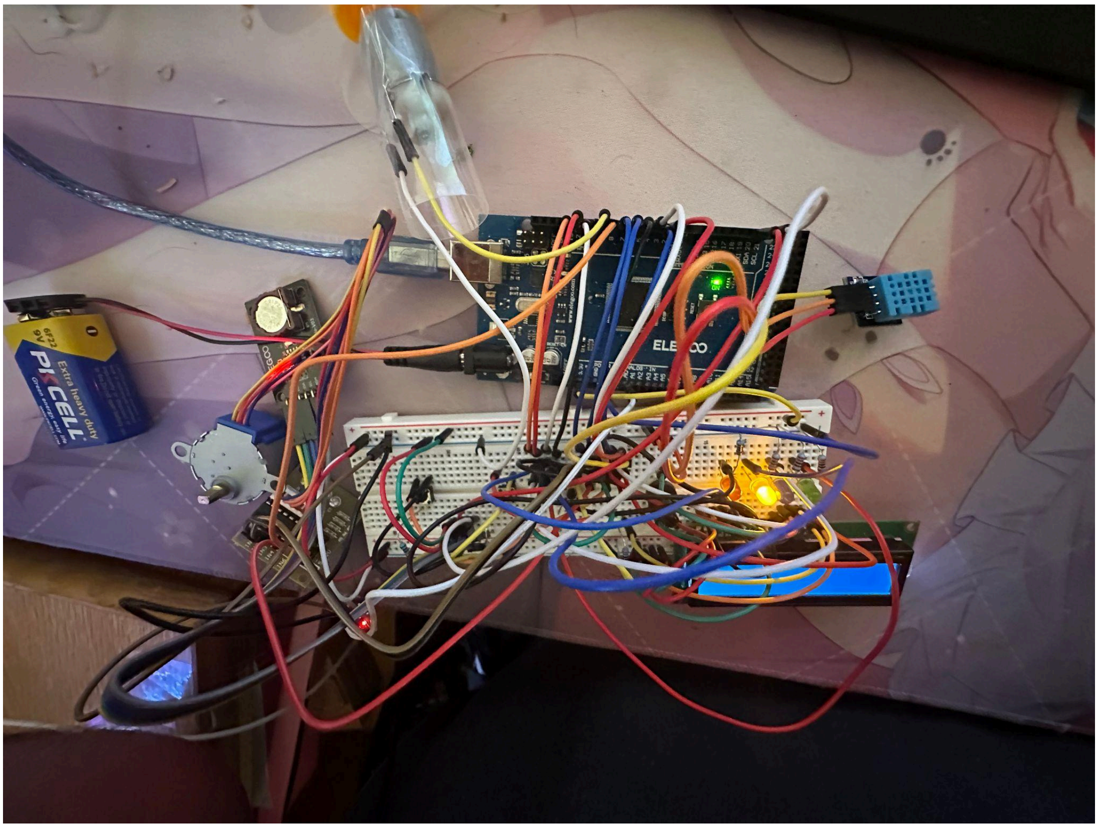
Components

1. Arduino Mega 2560
2. DHT11 Temperature and Humidity Sensor
3. Water Level Sensor
4. Fan Motor with Power Supply
5. RTC DS1307 Module
6. 16x2 LCD Display
7. LEDs (Yellow, Green, Red, Blue)
8. Push Buttons (START, STOP, RESET)

9. Stepper Motor (for vent direction control)

Schematic





Testing and Operation

Testing Procedure

1. **Initial Setup**:
    - Verify all connections and power supply.
    - Load the program onto the Arduino.
2. **Functional Tests**:
    - Press START to transition from DISABLED to IDLE.
    - Simulate temperature and water level changes to observe state transitions.
    - Verify LCD displays correct readings.
3. **Error Handling**:
    - Simulate low water level and observe transition to ERROR.
    - Test RESET and STOP functionality.

Expected Outputs

- **LCD**:
    - Displays temperature and humidity in IDLE and RUNNING states.
    - Shows error messages in ERROR state.
- **Serial Monitor**:
    - Logs state transitions with timestamps.

Deliverables

1. **Technical Documentation**:
    - Overview, schematic, and design details.
2. **GitHub Repository**:
    - Source code with proper comments.
3. **Video Demonstration**:
    - Show the system in operation, highlighting key features.

Final Project Report

The project was successfully implemented, with all major functionalities working as expected. The following features were achieved:

- Temperature and humidity monitoring using the DHT11 sensor.
- Water level monitoring and transitioning to the ERROR state when water levels were low.
- Fan motor control based on temperature thresholds.
- Accurate state transitions logged via the RTC and displayed on the serial monitor.

LCD Screen Issue

Unfortunately, the LCD screen was damaged during a previous lab session, rendering it non-functional for this project. As a workaround, all system stats and logs were output to the serial monitor, ensuring full visibility and functionality for testing and demonstration purposes.

Overall Outcome

Despite the LCD issue, the project performed flawlessly under all test conditions. Logs and sensor readings were verified via the serial monitor, and the system reliably transitioned between states according to the defined logic. This demonstrates the robustness of the design and its adherence to project requirements.