Project Overview

CPE 301 Final Project
Team: Exactly One Gallon of Milk

Keaton Clark, Chris Zinser, Jann Arellano, Lloyd Gonzales

https://github.com/Keaton-Clark/FinalProject

Table of Contents

Design

Overview

Over the course of several weeks, we developed a swamp cooler built out of the casing of an old refrigerator, with components sourced from a mix of the Arduino kit and old parts. The physical attributes of the swamp cooler adheres to the project requirements, with the additional changes:

- Uses a computer fan instead of the included 3-6V motor and fan blade
- RGB LED used instead of four separate LEDs
- Uses an I2C TWI module to interface with the LCD display to reduce the total number of wires necessary and simplify communication
- Powered using an external 12V source when in operation
- Sensors and other devices soldered onto solderable breadboards where possible; ribbon cables used to connect devices that are either physically separated from each other or must be freestanding
- A single button is used to handle all manual state changes (since exactly one of the start, stop, or reset buttons is "active" at any point in time)
- All components secured to the refrigerator casing

This followed development from a breadboard prototype, some pictures of which are included in this document.

The underlying codebase is written in C, using the AVR libraries and development environment (avr-gcc, avrdude) rather than the Arduino IDE. Libraries were handwritten for this project to interface with the DHT11, LCD display, stepper motor, and DS3231/DS1302, adhering to the communication protocols defined in each of these components' datasheets. In addition, a general-purpose I/O library was developed to abstract GPIO, ADC, UART, and TWI functionality.

At a high level, the main program works as follows:

- At initialization:
    - Enable (general) interrupts and set up the timer and external button interrupts.
    - Start the UART.
    - Start the LCD TWI.
    - Start the ADC.
    - Initialize the structs representing each component, performing initial readings if necessary.
    - Transition to DISABLED.
- In the main loop:
    - Call the relevant state handler for the current state.
    - Update the LCD.
    - Check if a state change to another state is necessary by reading the temperature and water level, and taking action if a transition exists based on these conditions.

The main program contains two ISRs:

- one to handle external interrupts, produced by the sole button handling manual state changes (INTx_vect with the EICRx, EIMSK registers for configuration)
- another to reload an internal clock that interrupts once per second, used in determining when to poll the DHT11 or update the LCD (a design choice made over polling the RTC module)

A makefile has been provided to both generate the binaries and flash a connected AVR microcontroller (assuming a specific dev port and an ATmega2560), with instructions for WSL development described in the README.

Components

The project used the following major components from the board (with their corresponding datasheets linked):

- DS1307 (real time clock module): https://www.sparkfun.com/datasheets/Components/DS1307.pdf
- DHT11 (humidity/temperature sensor): https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf
- HD44780U (2x16 LCD display): https://www.sparkfun.com/datasheets/LCD/HD44780.pdf
    - TWI board: https://www.nxp.com/docs/en/data-sheet/PCF8574_PCF8574A.pdf
- Water sensor: https://curtocircuito.com.br/datasheet/sensor/nivel_de_agua_analogico.pdf
- Stepper motor: https://www.mouser.com/datasheet/2/758/stepd-01-data-sheet-1143075.pdf
- ULN2003 stepper motor driver board: https://www.st.com/resource/en/datasheet/uln2001.pdf
- DC motor: https://bc-robotics.com/shop/3-6v-dc-hobby-motor-260/ (store link)
- Button: https://www.arduino.cc/documents/datasheets/Button.pdf
- LED (one-color): https://cdn-shop.adafruit.com/datasheets/WP7113SRD-D.pdf
- RGB LED: https://www.arduino.cc/documents/datasheets/LEDRGB-L-154A4SURK.pdf
- 10k potentiometer: https://components101.com/sites/default/files/component_datasheet/potentiometer%20datasheet.pdf (closest match)
- NPN transistor PN2222: http://users.ece.utexas.edu/~valvano/Datasheets/PN2222-D.pdf
- Diode rectifier: https://www.vishay.com/docs/88503/1n4001.pdf

Note that some of these components were replaced with functionally equivalent components when moving from our “prototype” design using only the components in the kit to the “final” design. Some of these components did not have datasheets available, but are *at least* as good as the kit-equivalent components with respect to operating conditions..

Parameters

(For a description of the pins used, refer to the Fritzing diagram under the Pictures section.)

The system has parameters that can be changed programmatically, most easily by redefining the macros in main.c:

- **VENT_DIRECTION_THRESHOLD = 50** - The change in analog signal necessary (from the last time the stepper motor was moved) for the stepper motor controlling the vent to be moved. This is used to help prevent the stepper motor from constantly moving due to natural noise on the analog line, which additionally reduces the number of times the blocking stepper_rotate. **Defaults to 50** (a change of approximately 0.25 V on the line).
- **STEPS_PER_ANALOG_UNIT = 0.5** - The number of stepper motor steps to take per unit difference between the last "move" and this "move", based on ADC readings at each iteration of the main loop (where the ADC readings are stored in a 16-bit integer).
- **WATER_LEVEL_THRESHOLD = 256** - The analog threshold of the water sensor before a transition from IDLE or RUNNING to ERROR.
- **TEMPERATURE_THRESHOLD = 0** - The temperature threshold in degrees Celsius needed for the swamp cooler to transition from IDLE to RUNNING.

Constraints

The system operates on a 12V external power supply (which has been measured to output 17V in certain cases).

The aggregate operating constraints for the system are (approximately) as follows:

- Maximum water reservoir size: 120x160x40mm (detection height of water sensor is 40mm, base of reservoir is up to 120x160mm)
- Operating humidity: 20 - 90% (limited by detection region of DHT11 and operational conditions of some sensors)
- Max operating temperature of 30C (limited by the water sensor)
- Minimum operating temperature of 10C (limited by the water sensor)
- Maximum water reserve capacity of 768 milliliters or 0.768 Liters or .768 soda bottles in non-freedom units.

Although most components can handle operating environments well beyond the conditions stated above, the conditions above describe the environment in which *all* components are known to work. An operating temperature of 25C has been noted to work best.

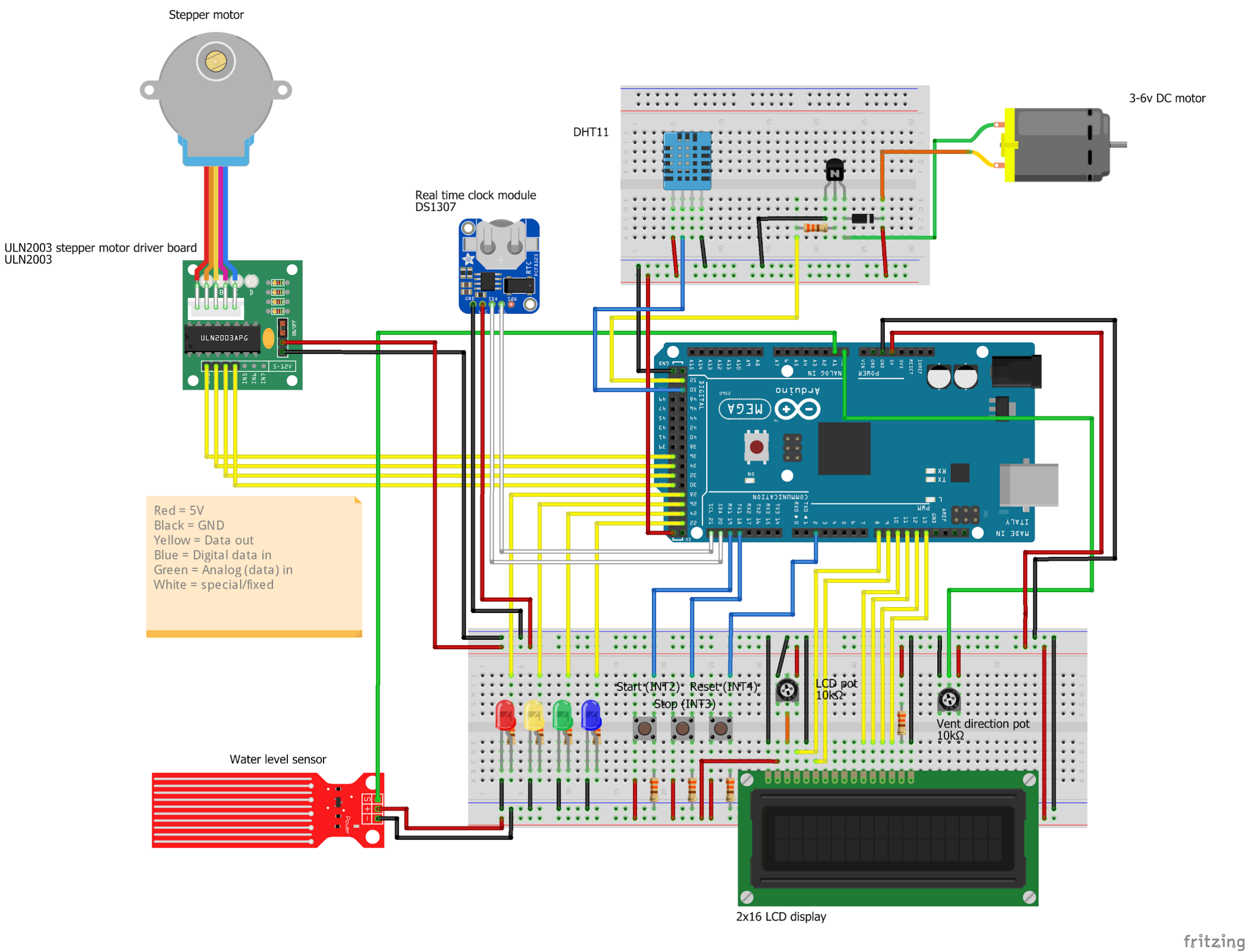
Individual components have the following (major) operating conditions and sensor ranges:

- DS1307 RTC: Operating temperature of -40C - 85C.
- DHT11: Sensing range of 20 - 90% RH and 0 - 50C. Accuracy is within 5% for humidity, and within 2C for temperature. Most accurate at 25 C.

- Rectifier: Operating temperature of -50 - 150C.
- Transistor: Operating temperature of -55 - 150C.
- Potentiometer: Noise of +/- 3%, operating temperature of -10 - 85C. 2 million cycles before degradation.
- LED: Operating temperature of -40 - 85C.
- Driver/stepper: Operating temperature of -40 -85C; storage temperature of -55 -150C.
- Water sensor: Operating temperature of 10 - 30C, 10% to 90% operating humidity, 40mm detection area (for a maximum detectable volume of 120x160x40mm).

Pictures and Videos

Schematic diagram



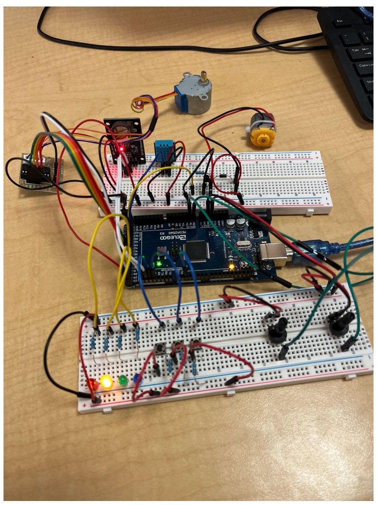
Note that this is the schematic diagram used for prototyping. The actual project, after switching out kit components for custom components, using solderable breadboards, and mounting components had several differences:

- DHT11 to digital pin 7

- Stepper motor to pins 25, 27, 29, 31
- Water level sensor to analog pin 14
- Potentiometer to analog pin 12
- Button to digital pin 2
- RGB LED to digital pins 4, 6, and 8
- Computer fan to digital pin 3

Prototype

Working Prototype: https://youtu.be/_qRYFwZBqhI
Picture:



Final design

Video: https://youtu.be/hOVqMPYb08s
Picture: