TABLE OF CONTENTS

Owner:
Louis A. M.
https://github.com/TheVelourFog

Introduction

The purpose of this project is to design, plan, and build a prototype of an ambient temperature triggered alarm. The thermal sensor will be able to read the environment's temperature in Celsius and use a buzzer and LEDs to indicate when a threshold is passed. This device could be useful for people who live in hot climates by alerting of dangerously hot temperatures. It could be applied to things like machinery to alert of overheating help prevent it. It could also be integrated into temperature-controlled environments, such as large scale refrigerated storage, to prevent loss due to temperature fluctuation.

Owner:
Louis A. M.
https://github.com/TheVelourFog

Specification

1. Inputs:
    I. Temperature
2. Outputs:
    a. LCD
        I. Displays Binary State and Temperature
    b. LED
        I. Green When Safe State
        II. Red When Danger State
    c. Buzzer
        I. Pulsating Alarm
3. Functions:
    a. Continuously Store Temperature Read In
        I. Display Temperature On LCD
        II. Update State Based On Temperature
            i. If Temperature Is Below A Threshold
                ❖ Change State To Safe
                ❖ Power Off Buzzer
            ii. If Temperature Is Above A Threshold
                ❖ Change State To Danger
                ❖ Power Buzzer

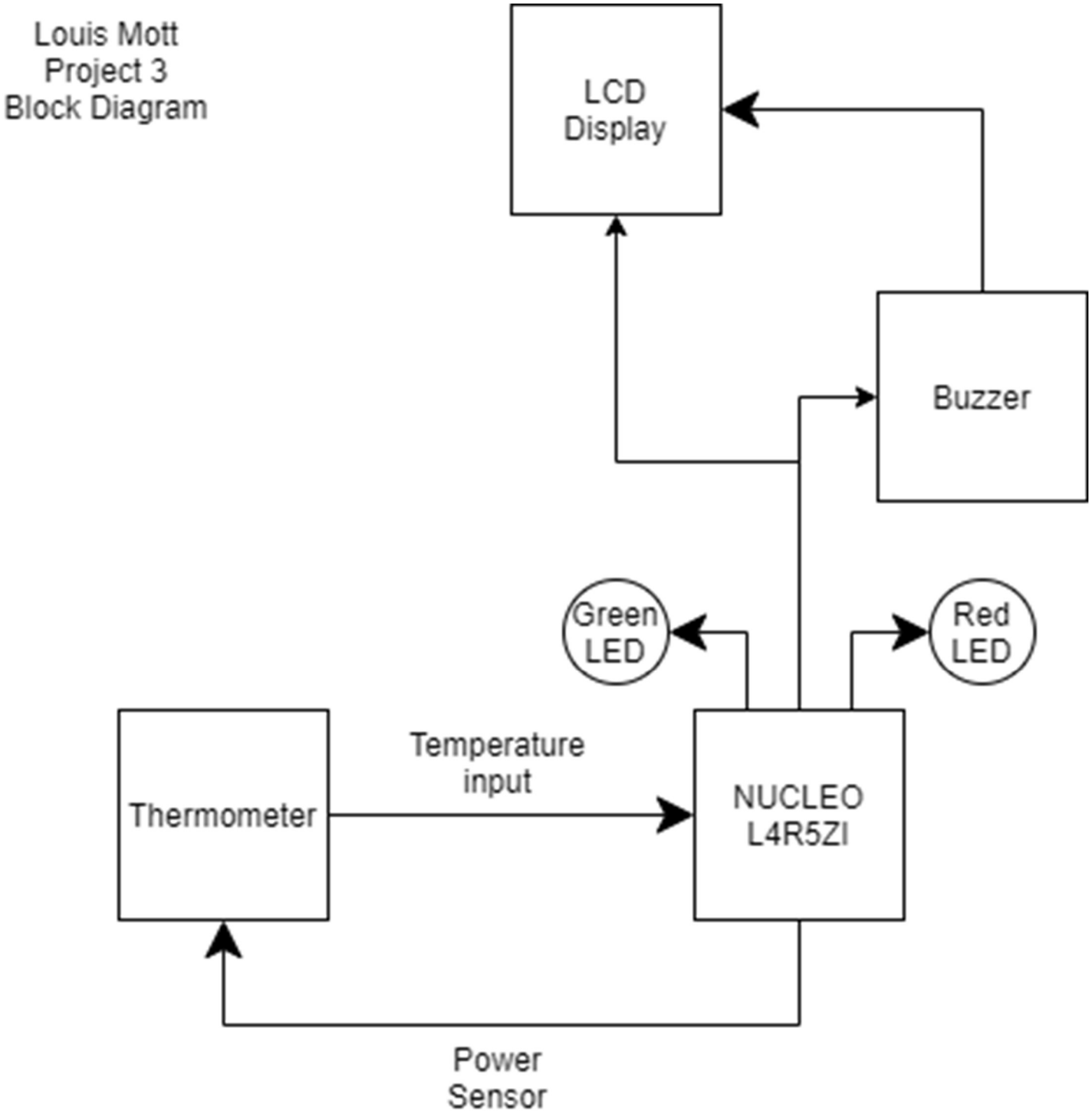
Owner:
Louis A. M.
https://github.com/TheVelourFog

Applications & Features Overview

This thermal alarm features a DHT11 ambient temperature sensor that is used to report temperatures to an integrated LCD. This display also displays Safe or Danger to report the current state. The current state is also indicated by a green or red LED for convenience. When the reported temperature enters the defined Danger zone, an alarm is triggered to audibly alert all those in the area. This thermal sensor could be easily applied to refrigerated transport vehicles, commercial refrigerated storage rooms, any ambient temperature control system, or it could be implemented as a heat alarm for public safety in extremely hot climates.

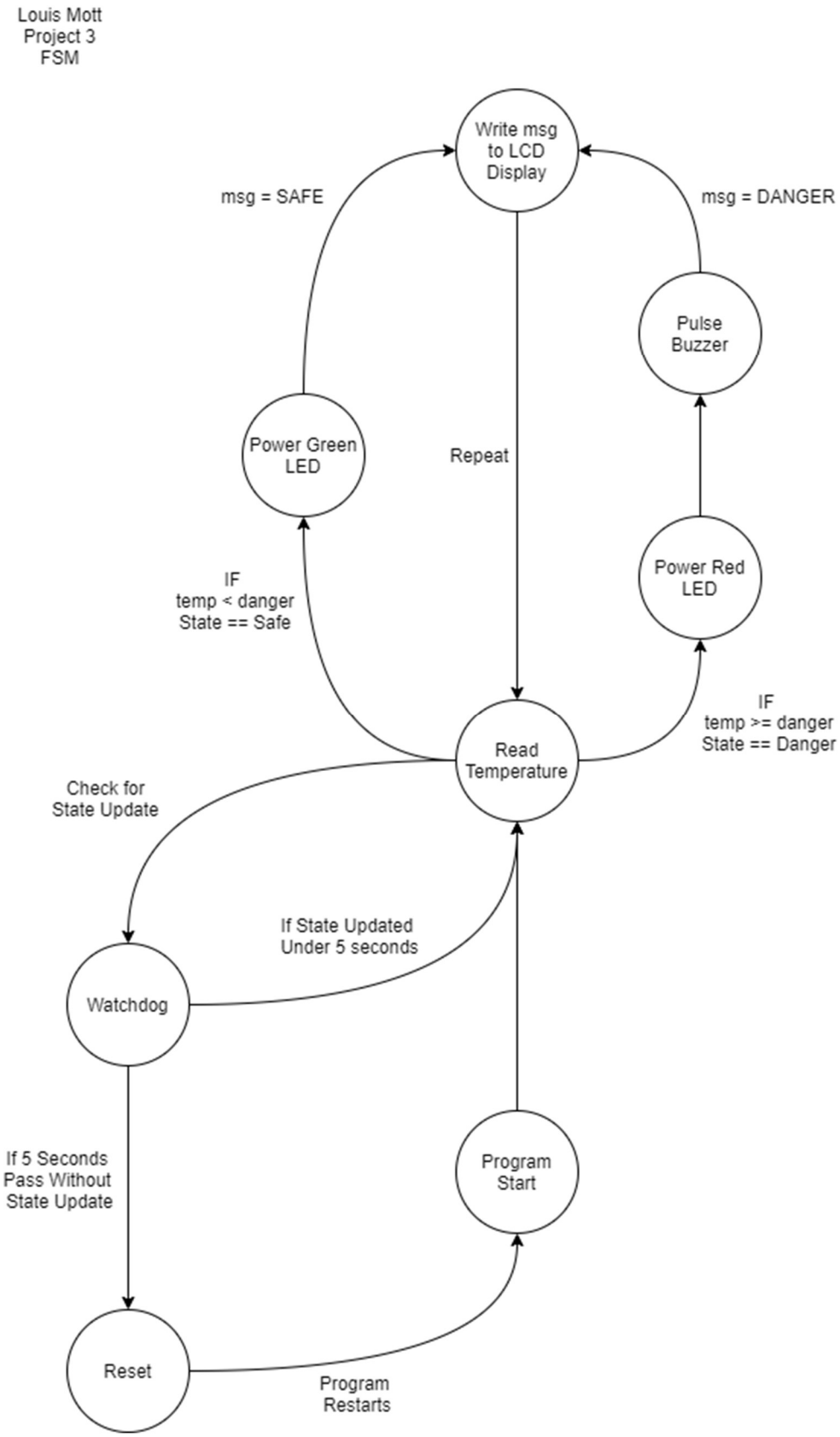
Owner:
Louis A. M.
https://github.com/TheVelourFog

Block Diagram



Owner:
Louis A. M.
https://github.com/TheVelourFog

Finite State Machine



Owner:
Louis A. M.
https://github.com/TheVelourFog

Bill of Materials

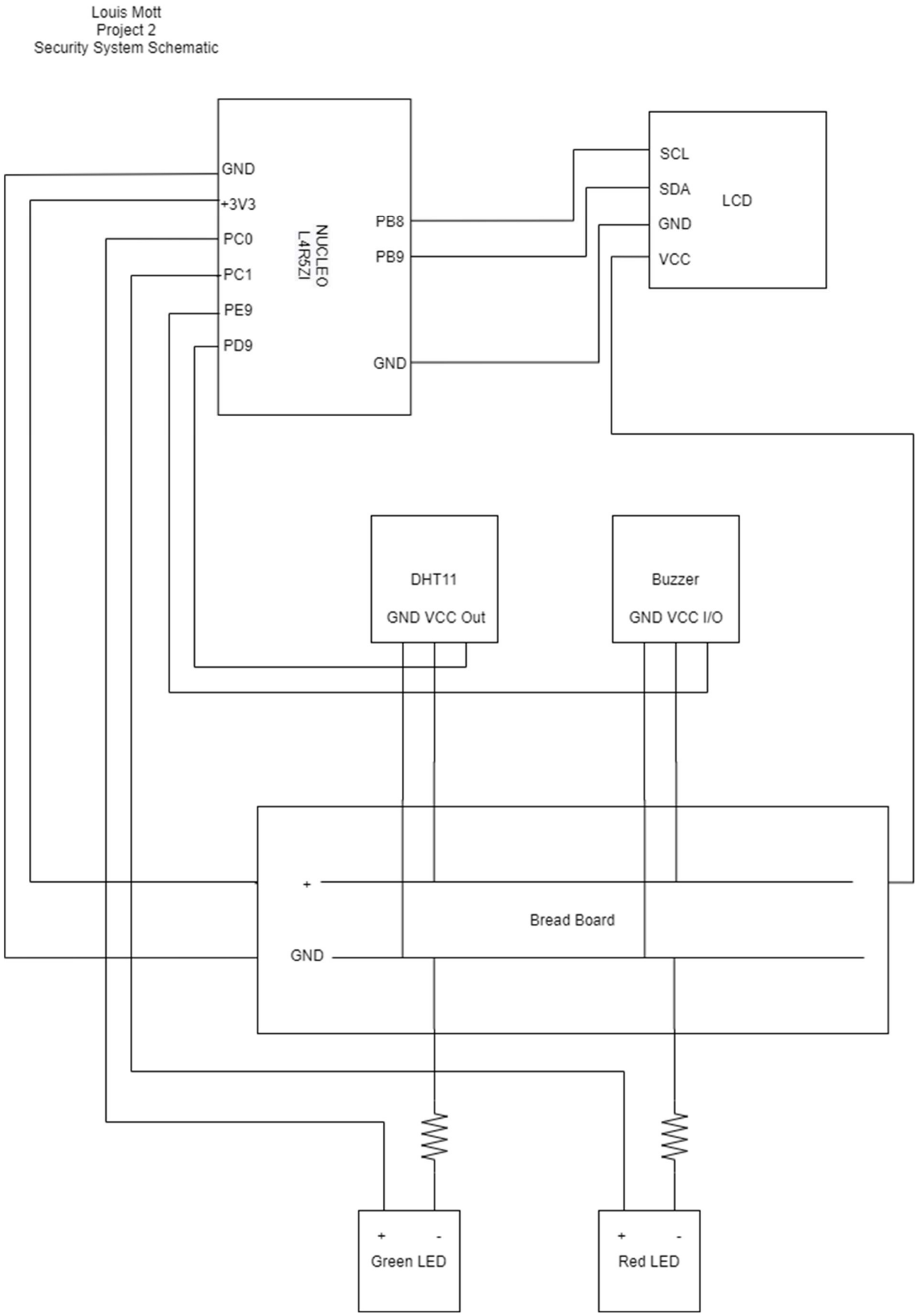
Item	Description	Link to purchase model you chose
1. Nucleo L4R5ZI	Embedded Processor Development Board	https://www.mouser.com/ProductDetail/STMicroelectronics/NUCLEO-L4R5ZI?qs=j%252B1pi9TdxUYHwRjgL7zLGg%3D%3D
2. Green LED	Single green light source	https://www.mouser.com/ProductDetail/SparkFun/COM-09592?qs=WyAARYrbSnZOqk%252BDI0izeg%3D%3D
3. Red LED	Single red light source	https://www.mouser.com/ProductDetail/SparkFun/COM-09590?qs=WyAARYrbSna4SgQTfntsTg%3D%3D
4. LCD	16 x 2 Display out	https://www.mouser.com/ProductDetail/Seeed-Studio/104020113?qs=0lSvoLzn4L9Che40BE0WKg%3D%3D
5. Buzzer	Alarm Sound Module	https://www.amazon.com/gp/product/B07MPYWVGD/ref=ppx_yo_dt_b_asin_title_o01_s00?ie=UTF8&psc=1
6. DHT11	Ambient Temperature and Humidity Sensor	https://www.amazon.com/gp/product/B07HF7CQHQ/ref=ppx_yo_dt_b_asin_title_o01_s00?ie=UTF8&psc=1
7. BreadBoard	Solderless Prototype Breadboard	https://www.amazon.com/DEYUE-Solderless-Prototype-Breadboard-Points/dp/B07NVWR495/ref=sr_1_10?dchild=1&keywords=breadboard&qid=1608718525&sr=8-10

Owner:
Louis A. M.
https://github.com/TheVelourFog

8. Male-to-Male Jumper	Wire with two male ends	https://www.amazon.com/GenBasic-Solderless-Dupont-Compatible-Breadboard-Prototyping/dp/B077N6HFCX/ref=sxts_sxwds-bia-wc-rsf-lq2a1_0?cv_ct_cx=male+to+male+jumper&dchild=1&keywords=male+to+male+jumper&pd_rd_i=B077N6HFCX&pd_rd_r=8c17b02f-32f1-4845-8435-5cc871a3a464&pd_rd_w=458ey&pd_rd_wg=hRfhn&pf_rd_p=52f9c563-bb87-44f4-9d9d-e1c03402d90f&pf_rd_r=WGW3SC94WF02E7TJ6QRA&psc=1&qid=1608718594&sr=1-1-d3e58e83-6458-471c-a87e-175495b96a10
9. Male-to-Female Jumper	Wire with male and female end	https://www.amazon.com/Elegoo-EL-CP-004-Multicolored-Breadboard-arduino/dp/B01EV70C78/ref=sr_1_3?dchild=1&keywords=male+to+female+jumper&qid=1608718624&sr=8-3
Carbon Film Resistor	1K Ohm Resistor for LED	https://www.amazon.com/uxcell-Carbon-Resistors-Tolerances-200pcs/dp/B07LG82FCT/ref=sr_1_16?dchild=1&keywords=carbon+film+resistor&qid=1608718646&sr=8-16

Owner:
Louis A. M.
https://github.com/TheVelourFog

Schematic



Instructions

Build:

1. Read and understand Schematic
2. Connect 3V3 power on Nucleo to positive rail on breadboard
3. Connect GND on Nucleo to negative rail on breadboard
4. Integrate LEDs
    a. Place red and green LEDs appropriately in breadboard
    b. Connect Green LED positive to pin PC0
    c. Connect Red LED positive lead to pin PC1
    d. Add resistor from ground to negative lead on each LED
5. Integrate DHT-11
    a. Plug sensor into breadboard appropriately
    b. Connect positive lead to power rail on breadboard
    c. Connect negative lead to ground rail on breadboard
    d. Connect output lead to pin PD9
6. Integrate LCD
    a. Connect VCC lead to power rail on breadboard
    b. Connect GND lead to ground rail on breadboard
    c. Connect SDA lead to PB9
    d. Connect SCL lead to PB8
7. Integrate Buzzer
    a. Plug buzzer into breadboard appropriately
    b. Connect positive lead in line with the positive lead of the Red LED
    c. Connect negative lead to ground rail on breadboard
    d. Connect I/O lead to pin PE9

Use:

1. Assemble Thermal Alarm
2. Connect to device that has Mbed Studio installed
3. Set Thermal Alarm codebase as active program
4. Click the build button to apply the program to the Nucleo device
5. Thermal Alarm will now function without outside interaction according to program

User Info:

1. The threshold variable can be modified to easily change the temperature threshold
2. The conditional in the checkTemp(void) function can be easily modified to change if threshold is seen as an upper temperature limit or lower temperature limit

Owner:
Louis A. M.
https://github.com/TheVelourFog

Testing Plan - Analysis

- **Temperature Sensor**
  - WHY: To ensure temperature readings are reported accurately. If this is not accurate. False readings could change the state when not necessary.
  - HOW: Change the temperature by opening or closing a nearby window, and use an analog thermometer to verify the temperature printed to the console is
- **Buzzer**
  - WHY: In order for this to be an alarm, it must have an auditory alert. It is also important to ensure it will turn on and off at the appropriate times.
  - HOW: When the state changes to "danger" ensure buzzer powers on for one second then off for one second repeatedly until the state changes to safe.
- **State Change**
  - WHY: Core necessity for the rest of the device's functionality.
  - HOW: Make sure state changes after condition is met by using onboard button to change state, and print the state to the terminal.
- **LCD**
  - WHY: This is used to display the state but mainly used to read the current temperature.
  - HOW: Ensure temperature being printed to console matches temperature displayed on LCD. Also make sure the displayed state changes when the red led is powered.
- **LED**
  - WHY: Some people are deaf, so it is important to have an easily identifiable visible queue for the state.
  - HOW: When the state is changed to "safe" the red LED should lose all power and the green LED should get steady power. When the state is changed to "danger" the red LED should get power and lose power and single second intervals, causing it to blink.

Owner:
Louis A. M.
https://github.com/TheVelourFog

Testing Instructions

1. LCD:
    a. Use heat source (hair dryer) to raise ambient temperature
    b. Verify temperature is updated in reasonable amount of time
    c. Verify state is updated after threshold breached

2. LEDs:
    a. Repeat 1.a
    b. Verify Green LED is powered when state is Safe
    c. Verify Red LED blink when state is Danger
3. DHT-11:
    a. Repeat 1.a
    b. Use store bought thermometer to verify temperature reported to LCD is accurate
4. Buzzer:
    a. Verify buzzer remains off while state is Safe
    b. Repeat 1.a
    c. Verify Buzzer pulses when state is Danger
    d. Allow to cool
    e. Verify buzzer turns off when state returns to Safe

Owner:
Louis A. M.
https://github.com/TheVelourFog

Development Timeline

CSE 321 Project 2 Plan

Owner:
Louis A. M.
https://github.com/TheVelourFog

Development Stage Summaries

Stage 1. Initially decided to make a device that would automatically power lighting when entering a room. This later changed to a temperature triggered alarm due to technology restraints.

Stage 2. Repository created on github, and initial project statement, constraints and specifications for project design.

Stage 3. Detailed initial inputs, outputs, and constrains. Also created a rough initial design plan. This plan included a bulleted list of objectives, FSM diagram, and gant chard detailing individual task timelines.

Stage 4. Progress report submitted detailing accomplished tasks, plans for the coming week, and possible foreseen concerns. Progress check-in with Dr. Winikus went well.

Stage 5. Final progress report submitted detailing accomplished tasks from the previous week, plans for finishing the project, and possible foreseen concerns. Report also included a bulleted plan for analyzing and testing implemented hardware and software

Owner:
Louis A. M.
https://github.com/TheVelourFog

Future Considerations

Shortfalls:

Does not utilize any watchdog utility. If the running program becomes hung and is not able to update the temperature or state, it will need to be manually reset. This poses a possibility for silent device failure.

Does not use threading or any synchronization technique. If these things were used, interrupts that check the incoming temperatures against the threshold could be handled on separate threads. This would improve efficiency.

DMA (Direct Memory Access):

Design plan did not account for DMA, and has no direct use for it.

General Improvement:

One feature that could be added to this device that would improve its usability would be the watchdog timer. The watchdog time would automate a restart of the program whenever an issue with state changes occurred. This would make it more appealing to the user because it would require less monitoring and cut down on false negatives/positives.

Owner:
Louis A. M.
https://github.com/TheVelourFog