Term-1 Project Review

Temperature Sensors for Veterans

sdmay23-07

Introduction

- Client: Adaptive Adventures
- Users: Individuals suffering loss of sensation to body appendages
- Goal of Project: Provide the ability to monitor skin temperature, receive alerts for dangerous body temperatures, communicate temperature data to both the user and supervising individual, and historize temperature data to provide users better assistance in making corrective actions for future occurrences

Hardware Implementation Architecture

Temperature sensing circuit

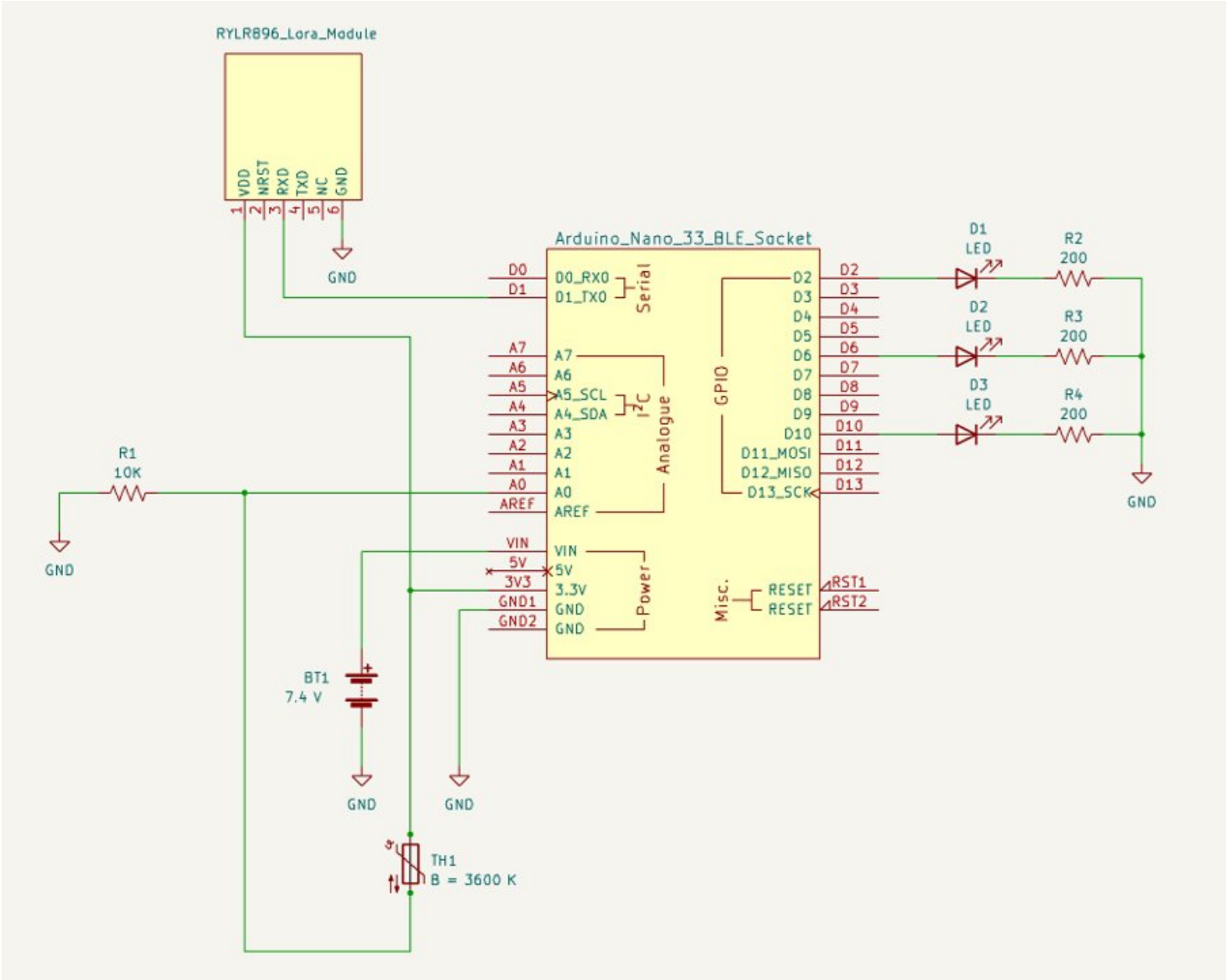
Bluetooth communication for short range data transmission and receival

Radio frequency module for long range communication

Push button power control, battery charging circuit, 7.4V power supply

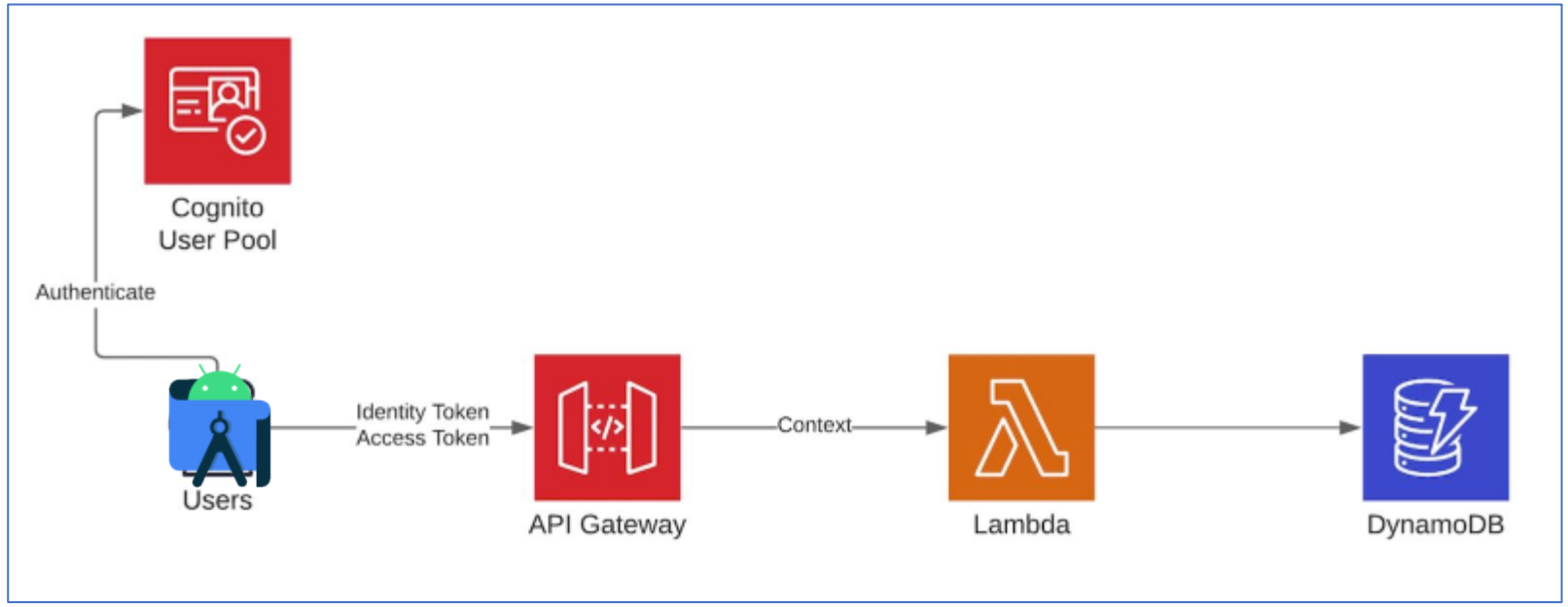
3 LEDs to alert for low battery level, temperature range exceeded, and loss of connection

Hardware Implementation Architecture Schematic



Software Backend Implementation Architecture

Backend:
Users' login/identification through Cognito, data flows bi-directionally through API gateway, Lambda, and DynamoDB.

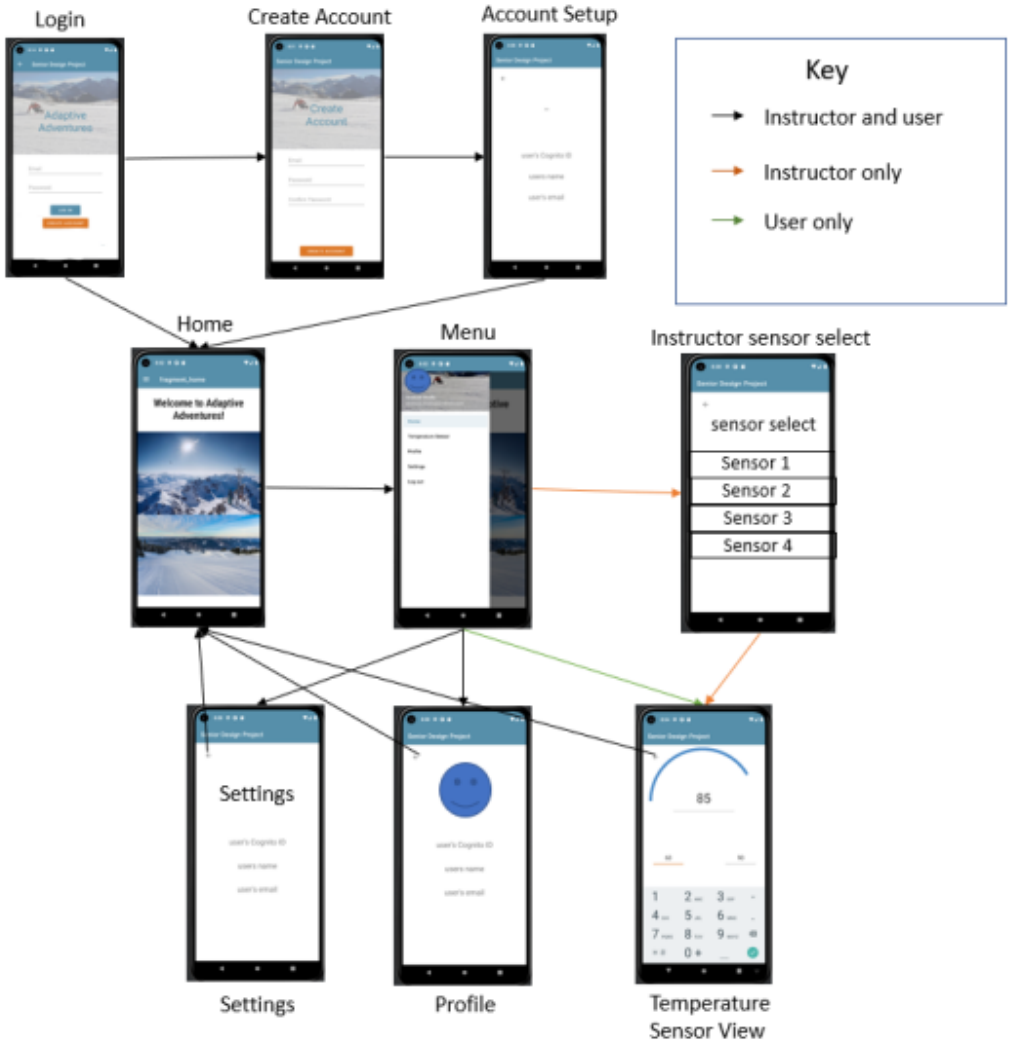


Software Implementation Architecture

- Login/Create Account: Allows user to access app securely using email and password.
- Account Setup: Upon completion of creating a username and password, the user must give more information to complete account setup.
- Home: Acts as a welcome page and access to a menu of access options.
- Instructor sensor select: Instructors can access multiple individuals' body temperature monitoring, this allows the instructor to pick which user to monitor.
- Settings/Profile: Allows users to change settings and profile information for the app
- Temperature Sensor View: Shows the current limb temperature and compares that temperature to set boundaries. The closeness to the boundaries is shown visually in a half circle progress bar, and a warning will go off if the body temperature reading is outside of the boundaries.

Frontend

Development in Android Studio, app written in Java



Bluetooth Implementation Architecture

Software

- Android studio implementation of the Bluetooth connection to the Arduino Bluetooth module will start by configuring Android studio to the Arduino's Bluetooth module using a specific UUID and MAC address of the Arduino nano. Also, there are functions for sending and receiving data from the Bluetooth module which will be used accordingly. Once the android application is closed, the Bluetooth connection will be terminated until opened again.

Hardware

- The Arduino nano will be able to connect to our Android application by advertising itself with a built-in function from Arduino, unitizing a UUID.
- During this process, the Arduino code will need to advertise specific services such as a ledService or pinService for analog values.
- A characteristic will need to be added to this service, along with a custom UUID that specifies what action will be taken, to provide read/write abilities
- Once this has been done, the phone app will be able to read or write data to the Arduino. The implementation of this data will then be design specific for both the Android application and Arduino functions

Hardware Work Progress

Milestone 1: Complete all pre-work necessary for project

- Task 1: Obtain, verify, and document all requirements for the project from the Client and Potential Users
  - Percent Complete: 100%
- Task 2: Complete necessary project research
  - Percent Complete: 100%

Milestone 2: Develop preliminary designs

- Task 1: Sketch preliminary hardware designs
  - Percent Complete: 100%
- Task 2: Acceptance of preliminary designs by users
  - Percent complete: 100%
- Task 3: Purchase all required materials
  - Percent Complete: 100%

Hardware Work Progress Cont.

- Milestone 3: Complete interface testing of preliminary designs
  - Task 1: Identify all interfaces of preliminary designs
    - Percent Complete: 100%
  - Task 2: Develop testing criteria and procedures for each interface
    - Percent Complete: 100%
  - Task 3: Complete and document testing of each interface
    - Percent Complete: 70%
- Milestone 4: Complete overall system testing
  - Task 1: Integrate multiple interfaces and ensure compatibility
    - Percent Complete: 20%
  - Task 2: Combine all interfaces of overall system
    - Percent Complete: 0%
  - Task 3: Develop testing criteria and procedure of overall system
    - Percent Complete: 100%
  - Task 4: Complete and document testing of overall system
    - Percent Complete: 0%

Work Progress

- Milestone 5: Complete Preliminary Prototype
  - Task 1: Develop PCB for System
    - Percent Complete: 0%
  - Task 2: Develop enclosure for device
    - Percent Complete: 10%
  - Task 3: Build overall prototype
    - Percent Complete: 0%
  - Task 4: Complete system testing of overall prototype
    - Percent Complete: 0%
- Milestone 6: Complete User Testing
  - Task 1: Provide prototype to potential users
    - Percent Complete: 0%
  - Task 2: Obtain feedback on prototype
    - Percent Complete: 0%
  - Task 3: Make adjustments as necessary
    - Percent Complete: 0%
  - Task 4: Repeat process until no adjustments needed
- Milestone 7: Submit Final Prototype

Software Work Progress

- Milestone 1: Complete all pre-work necessary for project
  - Task 1: Obtain, verify, and document all requirements for the project from the Client and Potential Users
    - Percent Complete: 100%
  - Task 2: Complete necessary project research
    - Percent Complete: 100%
- Milestone 2: Develop preliminary designs
  - Task 1: Sketch preliminary software designs
    - Percent Complete: 100%
- Milestone 3: Complete interface testing of preliminary designs
  - Task 1: Identify all interfaces of preliminary designs
    - Percent Complete: 100%
  - Task 2: Develop testing criteria and procedures for each interface
    - Percent Complete: 100%
  - Task 3: Complete and document testing of each interface
    - Percent Complete: 30%

Software Work Progress

- Milestone 4: Complete overall system testing
  - Task 1: Integrate multiple interfaces and ensure compatibility
    - Percent Complete: 20%
  - Task 2: Combine all interfaces of overall system
    - Percent Complete: 0%
  - Task 3: Develop testing criteria and procedure
    - Percent Complete: 100%
  - Task 4: Complete and document testing of overall system
    - Percent Complete: 0%
- Milestone 5: Complete Preliminary Prototype
  - Task 1: Develop frontend interfaces
    - Percent Complete: 70%
  - Task 2: Develop AWS backend
    - Percent Complete: 70%
  - Task 3: Create Bluetooth connection in application
    - Percent Complete: 50%
  - Task 4: Connect software with hardware
    - Percent Complete: 0%
  - Task 5: Complete system testing of overall prototype
    - Percent Complete: 0%

Software Work Progress

- Milestone 6: Complete Application User Testing
  - Task 1: Provide application to potential users
    - Percent Complete: 25%
  - Task 2: Obtain feedback on application
    - Percent Complete: 25%
  - Task 3: Make adjustments as necessary
    - Percent Complete: 0%
  - Task 4: Repeat process until no adjustments needed
- Milestone 7: Complete System Prototype User Testing
  - Task 1: Provide system prototype to potential users
    - Percent Complete: 0%
  - Task 2: Obtain feedback on system prototype
    - Percent Complete: 0%
  - Task 3: Make adjustments as necessary
    - Percent Complete: 0%
  - Task 4: Repeat process until no adjustments needed
- Milestone 8: Submit Final Prototype

Key Contributions

- Michael: Developed power control and management system
- Caleb: Calibrated and tested thermosensing hardware and code, designed project schematic.
- Max: Work with Communications via RF
- Jared: Work with Communications via BLE
- George: Work on the SW BLE connection
- Bridget: Developed frontend interfaces and temperature sensor visuals.
- Jamie: Implemented AWS backend (login features, databases, etc.)

Hardware Challenges and Solutions

- Difficulty in creating controlled testing environments
  - Using available applications and environments to emulate required conditions to the best of our ability
- Lead times of materials
  - Researching necessary hardware and placing orders well in advance
- Secondary alert system
  - Explaining difficulties to the Users and coming to an agreement on alerts – Visual and Vibratory
- Arduino code having to be in a continuous loop
  - Having a counter variable that allows certain actions such as sending data via BLE and RF to only happen at specific time intervals
- Thermistor temperature reading being inaccurate with basic B value equation
  - Implemented Steinhard-Hart equation to optimize temperature readings
  - Continued calibration in controlled temperatures using precision cooker
- Loss of RF communication (LoRa just sends out data – it does not check if there is something available to receive the data)
  - Made it so that each RF module is both a transmitter and receiver

Software Challenges and Solutions

- Outdated resources and documentation
  - Read new documentation if available, review old if not. Find new resources/libraries to utilize.
- Temperature status bar freezing after exceeding high-alert
  - Find corrupted reference in xml and delete.
- Databases unable to be queried directly
  - Act through API Gateway/Lambda, directly code transitions.

Remaining Work – Hardware

- Complete RF module range testing and final code implementation for User's device and Instructor's device
- Finalize BLE communication between Arduino and Phone application
- Complete testing on push button power control, battery charging capability, and low battery level indication
- Complete final testing to determine accuracy of temperature sensor and calibrate as necessary
- Complete design and testing of alert systems, both visual and vibratory
- User Testing

Remaining Work - SW

- Software
  - BLE
  - Implement API calls
  - Finish all interfaces
  - Implement Client vs Instructor profiles
  - Update profile abilities
  - Menu fragmentation
  - Connect to Hardware
  - User testing
  - Test Front-to-Back and Back-to-Front
  - System testing with backend

Conclusion

Current Status:

After gathering user-needs, functionality, creating the initial design and completing all related documentation last semester we have begun the implementation process for our design. We have gathered necessary hardware to begin building the physical protype. At this point in the hardware design, we have laid out testing criteria and are performing said tests. The android app is being developed in parallel with the hardware. We have implemented initial app design and have proposed our interface and functionality with the user and client. Front and backend development are well underway, with communications being worked as well.

Plan of action:

Plan of action at this point for our team is to finish individual component testing, then begin to test the prototype as a whole unit and see satisfactory results from the start of the system pipeline to the end. Temp sensor to a reading on the app. Once this is done, we can design the PCB and encloser and work on finalizing our hardware design. For the app development, we are continuing to finish the UI and back end. Testing the app with the hardware can begin soon. When our testing is complete, we will finalize our design based on the results.