Running Buddy

This is running buddy, a tool for counting steps and making runners visible at night. The step counter uses a high-pass filter and data queue to remove noise from the accelerometer and give the most accurate output. Logs for step count and button presses are recorded to an on-board 64gb SD card, which is enough storage for a lifetime of running data. The circuit is powered by a single 18650 3000 mAh battery, including the lights, which from testing lasts up to 8 hours with the lights on. The screen is positioned facing the user over their left shoulder, which is my preferred fanny-pack style when running. The screen shows two different views, for the current temperature and step count. These screens are toggled through left and right capacitive buttons located on the bottom part of the controller. The two buttons on the top of the controller toggle the lights. There are no markings on the controller housing for what the buttons do, but it is easy enough to remember.

The lights were sewn on to the bag, and additionally hot glued in place, which made assembly relatively straightforward. I tested running with the running buddy and nothing fell off or broke, and it seems fairly robust. After the run I downloaded the data from the sd card, which contains accelerometer data, push button data and temperature data (3 sensor inputs). This data could be post-processed and used to generate a GUI displaying step data over time, estimated distance traveled, and average temperature, but this would be out of scope for the given assignment. The feedback devices for the user include the main led light array, on-board LEDs on the microcontroller, and the LCD display.

When building the running buddy, I ran into several issues. I wanted to add a buzzer to give notifications when the user reaches a certain step count. The buzzer did not want to cooperate, however, and I was only able to get it to work with an on-board relay. There was not enough time to add this code to the project, but given another few days it would have made it in the project. I also ran into issues with tuning the capacitive buttons, and it is still too sensitive for what I had in mind. Next steps would include finding the right threshold through experimentation for the sensitivity of the buttons.

There were several features that I wanted to implement as well but were out of scope for the assignment. One was a 3D printed case housing the microcontroller. I created an initial mock-up, but due to time constraints was not able to create a polished final prototype. Another feature that would have been great to add is an automated system for collecting and saving the running data. A mobile app connecting to the mkr 1010 over bluetooth would have provided this functionality. Again, due to the constrained scope of the assignment I decided not to proceed with building this. Maybe for the final project I will revisit this idea.

The main limitation of the system is not having GPS data for recording where you actually ran. This is by design, as GPS would take much more power and the goal of the running buddy is

mainly to be visible at night. However, a low-powered GPS module would be invaluable when recording where you ran.

Here is a link to the code:
https://github.com/jschmidtnj/ubicomp/tree/main/running_buddy/embedded
This is a sample data file:
https://github.com/jschmidtnj/ubicomp/blob/main/running_buddy/embedded/data.csv
And this is a video showing running buddy's functionality: https://youtu.be/lsQlp0ld6Hg

Some pictures: