Landhopper: GPS Watch

Gary Bailey Twain Byrnes Carson Swoveland

Carnegie Mellon University, 18-500 ECE Project Experience, Team B8

04 May 2024

***Abstract* — We created a GPS watch tailored for active users who value control over access to their location data. Compared to existing solutions, our creation ensures prolonged, uninterrupted use during outdoor activities by utilizing solar energy harvesting. By streamlining the feature set, the watch can offer users an extended and reliable tracking experience.**

***Index Terms* — Energy harvesting, GPS, smart wearables**

1 Introduction

The Landhopper GPS watch is designed as a streamlined, minimal, and convenient device for personal location tracking and navigation. The primary use case for our GPS watch centers around empowering individuals engaged in activities such as remote week-long backpacking trips or urban walks, keeping track of their location no matter where they may be. In an era where people increasingly seek active and adventurous lifestyles while simultaneously feeling a stronger urge to protect their own privacy, the need for a reliable, long-lasting GPS watch that prioritizes privacy has become more pronounced.

Figure 1: The finished Landhopper watch.

Our watch holds its importance from four facets: its reliable tracking, its extended battery life, its user privacy concerns, and its contribution to user safety. Our watch is necessary for those who need location tracking in any GPS-friendly environment: it works even in rural areas where location tracking is typically much worse. Its extended battery life makes it indispensable for those who do not have constant access to charging, such as in hiking settings. Landhopper's design holds user privacy is held in high regard, and empowers users by leaving them complete control over their own information. Finally, in case of emergency, a user could use our watch to follow their previously-taken path to find a way back to safety, potentially saving lives.

In a landscape filled with various tracking technologies, our approach stands out due to its combination of an extended battery life and its offer of user privacy. While most other watches providing location data also upload this information to the cloud and sell it to private companies, our GPS watch prioritizes privacy of its users. We guarantee this by only storing the user's location data locally, making the data available exclusively to them, while also increasing the battery life of the watch. Additionally, most existing smartwatches must be charged every one to two days. Our watch utilizes energy harvesting to ensure a battery life that will make it usable by hikers who need location tracking on long expeditions. Our watch provides individuals with a powerful, reliable, and secure tracking device that seamlessly integrates into their active lifestyles, enhancing their outdoor experiences while contributing to the broader goals of public health and safety.

2 Use-case Requirements

We can sum up the potential needs of the user in one working example: a week-long backpacking trip. Many go on such trips without a GPS watch, since most watches providing GPS do not have a long enough battery life. Framing our use case requirements around this scenario aligns with public health and safety, ensuring that more people can go on such trips and allowing them to prioritize their physical health in hiking and their mental health in an escape from the stress and hubbub of modern society, while offering them safety and peace of mind through GPS tracking. Additionally, while the watch's agnostic approach to network existence is helpful for a user going on a backpacking trip, it is also important from a more global perspective, as this feature makes it a much more useful tool for those who need to know the time and their location but may never have consistent access to power or a network connection.

First, we determine the standard use case for a backpacker who may be interested in this watch. Since we are creating a watch that charges itself with energy harvested from its environment, we must define the environment of our use-case, and since this is a wearable device, this means defining the activity of our user. We define our user to be one who is outdoors and active for six hours per day. We pick this time since two miles per hour is a reasonable backpacking pace, and backpackers often walk 8-16 miles per day. Taking the average of these distances, we end at about 6 hours per day.

Since many backpacking trips last about a week, an active user should be able to sustain power to their watch for at least a week, starting from a full battery. We should be able to store this location data locally, equivalent to one week's worth of data. To be useful for tracking progress, the watch should be able to provide a total distance estimate with an error of not more than 10%. Finally, the user must actually be able to wear the watch for an extended period of time: that is, the watch should be no heavier than 0.5lb in order to allow for constant or consistent wear.

3 System Architecture

The most stringent requirements are the long battery life and the small size. Our design has been primarily informed by the former. Our design consists of two main subsystems: the energy harvesting system and the navigation system.

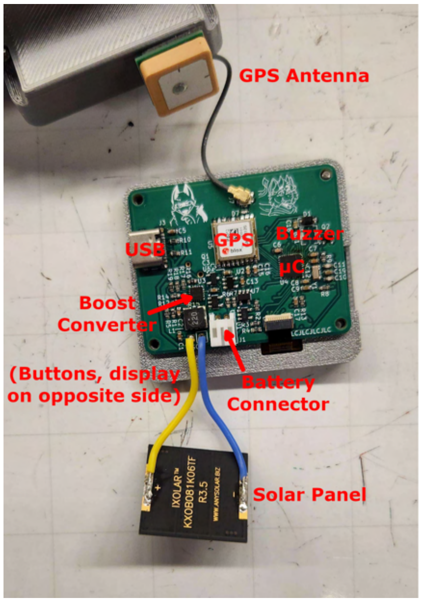

Figure 2: Annotated photo of the internals of the final version of Landhopper

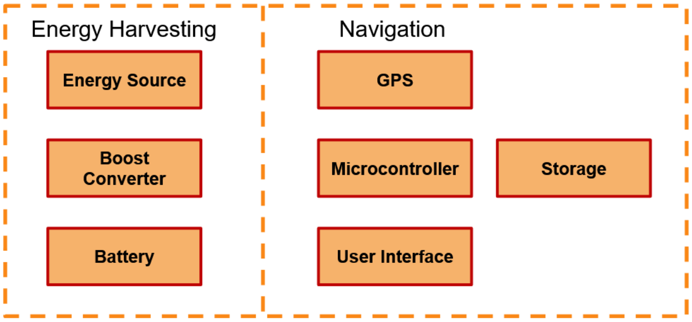

Figure 3: Block diagram of Landhopper's overall architecture. See System Implementation for a more detailed view.

The energy harvesting system converts, stores, and regulates energy obtained from the motion of the watch. The navigation system receives the GPS location, keeps a log of the user's path, and runs the user interface.

The fundamental operation of the energy harvesting component is fairly standard among commercial smartwatches. We spent a large amount of time considering various energy harvesting solutions, which is discussed in 5.1, and in the end we decided on the solar cell. Our solar cell produces a fluctuating DC voltage. This voltage is too unstable and too low to directly charge a battery, so we then use a boost converter to increase and regulate the voltage. Because the voltage of the battery is not necessarily appropriate for the rest of the system, we then pass it through a voltage regulator to get our main system voltage. The battery can also be charged using the USB port when the user has access to power.

The navigation component intentionally has the minimum feature set to meet our use-case requirements. A single microcontroller interfaces with the display, the controls, the GPS receiver, and a USB port. By default, the microcontroller and GPS receiver stay in sleep mode to conserve power. A timer in the GPS receiver wakes the receiver when a new sample is required, and the transmission of data wakes the microcontroller so that it can update the display and then return to sleep. During sleep, the display maintains its current state, and the GPS receiver keeps track of its previous fix information. This makes the process of updating the overall state of the watch very fast, as the GPS receiver can get a "warm start" fix and we can update only the lines of the display that have actually changed.

Our storage component is an SD card. The microcontroller interfaces with this in order to store a user's logged location. The stored paths can then be read by a computer and loaded into common mapping software to be visualized. Additionally, SD card stores map data which can be viewed on the watch itself. The watch can be set to a mode where, rather than displaying the time, it displays a simple street map of the area near the user. This map updates in real time as the user moves, keeping their location in the center.

3.1 Principles

We adhere to the engineering principle that less is better by intentionally choosing a small set of features and using only the minimum hardware required to implement those features. This reduces cost, improves energy efficiency, and improves reliability.

While testing the kinetic energy harvesting mechanism (discussed later), we varied several different parameters to find the optimal size for power output. The parameters included gear ratio, gear size, bearing size, and number of planetary gears.

To render the map, we use the Web Mercator projection. This converts the Earth's spherical shape into a 2D approximation, with the exchange being that positions very near the North or South pole cannot be represented. As Landhopper is not designed to be rugged enough for (Ant)arctic expeditions, we consider this to be an acceptable trade-off.

3.2 Changes From Design Report

Our system architecture remains largely the same as in the Design Report. However, we have finalized on using a solar cell as our energy source and changed how the watch interfaces with the computer (SD card instead of USB).

4 Design Requirements

4.1 Accuracy

Many remote trails are characterized by frequent turns and switchbacks caused by uneven terrain and steep slopes. In order to provide users with accurate distance measurements, we need to be able to measure these complex paths. Taking the Kit Carson Peak trail in Colorado as an example of a moderate-to-advanced route, we can see that there are frequent switchbacks that are no more than 150 feet across [4]. Measuring a straight-line distance across these switchbacks will dramatically underestimate the traveled distance. Thus, we need a resolution far below the size of the switchbacks; a maximum error of 50 feet should be sufficient. Furthermore, we need to sample frequently enough to make sure we do not miss any turns. An endurance-hiking pace of 1 mile per hour would take about 2 minutes to cross one of these switchbacks, so we need to sample at a faster rate. One sample per minute will meet this goal.

4.2 Battery Life

Our primary concern with this watch (as with most portable electronics) is battery life. Our requirements determine that our sample period (denoted by T) must be 5 minutes; our overall lifespan (denoted by L) must be 1 week; and that we do not have space or weight for a battery larger than 500mAh, which results in a capacity (denoted by B) of approximately $3.3\text{V} * 500\text{mAh} = 99$ Joules. The overall battery life of the watch is then determined by our average harvested power (denoted by P_H); active power (denoted by P_A), which is the power draw when taking a sample; sleep power (denoted by P_S), which is the power draw when not taking a sample; and the time required to take a sample (denoted by t). The relationship between these variables is given by:

$$L \leq \frac{B}{\frac{t}{T}P_A + P_S - P_H} \tag{1}$$

If we plug in and rearrange based off of the use-case requirements, we find:

$$\frac{t}{60\text{s}}P_A + P_S - P_H \leq 163\mu\text{W} \tag{2}$$

This equation has four variables, but we can add educated constraints based on what we expect to be achievable. Many commercial GPS modules can get a fix from a warm start in approximately a second [11], so we can set t to 1 second (the overhead of updating the display is negligible). Additionally, the lowest-power GPS modules on the market use about 15mW active, and the GPS receiver is the most power-hungry component, so we can pessimistically assume that our entire system will draw 60mW when active. Most microcontroller-oriented devices draw only a few hundred microwatts, so we can pessimistically assume that the sleep power draw will be 1mW. We can then determine the average power requirement of our energy harvesting scheme:

$$P_H \geq 1.837mW \tag{3}$$

5 Design Trade Studies

5.1 Energy Harvesting

There are several different sources of energy used in contemporary smartwatches, including electromagnetic, piezoelectric, photovoltaic, and triboelectric approaches [6]. Our primary design consideration here was ease of fabrication or sourcing: any component that we cannot find or make for less than $100 using materials available in CMU makerspaces is immediately non-viable. We considered each of these different types of energy harvesting:

Triboelectric approaches are immediately non-viable, as triboelectric energy harvesting modules are not available off-the-shelf and manufacturing them requires very advanced material science work [3].

Piezoelectric approaches work even without sunlight and have lower complexity compared to other mechanical approaches, and have demonstrated usefulness in wearable energy harvesting. We attempted to assemble a piezoelectric energy harvesting setup following work by Wang et al. [6], but the piezo elements in the paper were highly expensive, and we could not find a robust way to attach the piezo elements we could find. Thus, we rejected the piezoelectric strategy.

Kinetic approaches are inspired by mechanical watches, and involve using an offset weight to spin a generator. To get usable voltage levels, the speed of the rotation must be

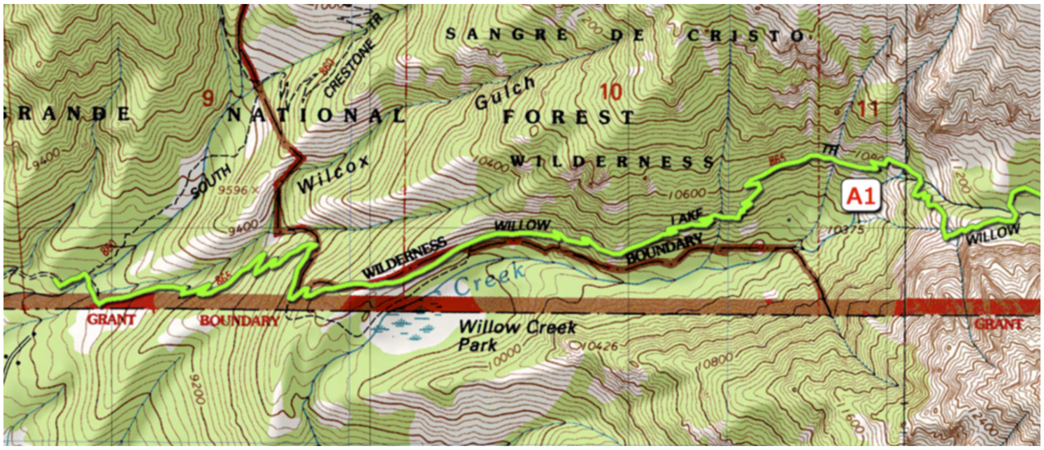

Figure 4: An excerpt from a trail map for Kit Carson Peak, showing jagged and frequent turns. The watch must be able to accurately track these details [4].

increased significantly, which requires adding a gearbox between the weight and the generator. Roughly following the approach by Cai et al. [1], we designed several iterations of a planetary gearbox attached to an off-the-shelf brushless gimbal motor from SparkFun [9]. We found the best performance at a 4:1 speed increase, where we were able to attain 10mW of peak power. This is definitely sufficient to power the watch, but the assembly was bulky and heavy, so we also rejected this strategy.

Photovoltaic (solar) energy harvesting only works in bright sunlight, but has a number of advantages. It has high power output, is easy to use, doesn't require rectification, and can be very compact. In bright sunlight, we found that a solar panel that could fit on the watch band produced up to 50mW of continuous output, far exceeding the required amount. This energy output still remained above 2mW even at very sharp angles to the sunlight.

The comparison between the energy harvesting approaches is summarized in Table 1. We decided on solar power due to its ease of use, high power output, and small size.

5.2 Battery

There are several different battery technologies that could be used in a watch, such as nickel-metal hydride (NiMH), lithium ion (LiIon) / lithium polymer (LiPo), and newer technologies like lithium iron phosphate (LiFePO4). We chose to use a very standard lithium polymer battery, since they are inexpensive, have a high energy density, and charge controllers for them are readily available and easy to integrate. Lithium polymer batteries are not considered as safe as some of the other technologies discussed, but when operated within specifications (i.e. not overcharged or overdischarged), they are known to be very reliable. We previously set our battery size limit to 500mAh when calculating for our energy harvesting requirement, so this is the capacity we used in the final version.

Table 1: Comparison of Energy Harvesting Methods

	Piezo	Kinetic	Solar
Can Build?	No	Yes	Yes
Power	N/A	10mW	50mW
Size	Tiny	Large	Small

5.3 Display

There is a large number of display technologies which could be used on a watch. Our primary consideration here is power consumption, so this immediately rules out displays that require a backlight or active emission, like OLED displays. E-Ink is a promising choice due to having near-zero quiescent draw, but they have extremely poor refresh rates and require complicated driver circuitry [2]. LCD displays have excellent refresh rates and low power draw, but likewise often require complex driving circuitry. Fortunately, Sharp produces an LCD display that integrates memory and driving circuitry, so it has all the benefits of an LCD while being easy to interface with [14]. This is the option we went with on Landhopper.

5.4 GPS Module

There are many different manufacturers and models of GPS and GNSS modules aimed at different applications. The most important consideration when choosing one was once again power consumption. There are several manufacturers of low-power GPS modules, such as U-blox, Quectel, and STMicroelectronics. However, out of all of these, U-blox makes what was from our research the lowest-power GPS module on the market, the MAX-M10S [11]. This module can maintain GPS tracking while consuming as little as 15 mW, consumes less than 1 µW in standby mode, and has a time-to-first-fix of 1 second from standby, which is important for regaining accuracy after a user exits a building. Another important consideration made when choosing the GPS module was the protocol which it uses to communicate with the microcontroller. The standard protocol used to communicate GPS data is NMEA 0183, but this protocol is needlessly verbose and difficult to parse due to being based on ASCII text. Many of the GPS modules from U-blox support a different protocol called UBX, which is a binary protocol designed specifically for ease of parsing. Using this protocol instead of the standard NMEA 0183 protocol allows our GPS data parser to be significantly simpler, saving code space and processing time [17].

5.5 Microcontroller

Like with the GPS module and the display, many options exist for the primary microcontroller for our system. The primary consideration was power consumption, as before, but here we also wanted to ensure ease of programming, as firmware and UI constitute a major portion of our design. For this latter reason, we considered several different microcontroller series, such as the Atmel AVR, Atmel SAM, and STMicroelectronics STM32 series [15], as various members of our team have experience with them. In the end, we chose the STM32 series, specifically the STM32L4 subseries, for several reasons: it is the series that the most of team members have experience with, and it has an ARM Cortex-M4F processor core, allowing for us to use floating-point numbers in our firmware, while still consuming very little power: at the default clock speed, it uses less than 2 mW in run mode, which is reduced to less than 0.5 mW in sleep mode and further reduced to just a few microwatts (while still retaining memory contents) in stop mode.

5.6 Firmware Language

There's very few options in terms of programming language for the STM32 platform: the options are pretty much just assembly, C, C++, and Rust. We chose to implement the firmware for Landhopper in the Rust programming language for several reasons:

- Rust's language design stops many memory-related bugs at the compilation stage. This is especially important on an embedded system, where there's no operating system to catch accesses to memory outside of expected regions.
- The Rust Embedded ecosystem is designed so that many of the libraries are generic and applicable to any embedded platform. This gives us a large set of libraries at our disposal for tasks such as SD card reading and task scheduling.
- Several of our team members have worked with STM32 devices in the past and have found the C libraries for this platform irritating.

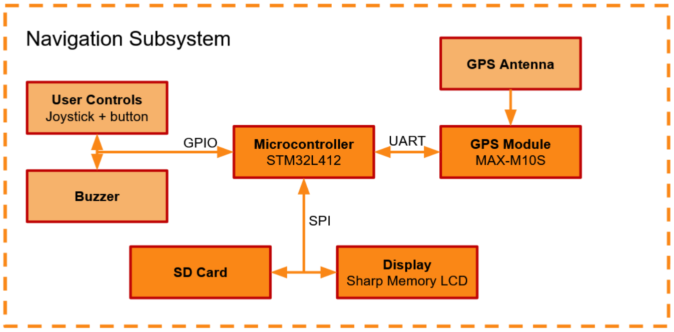

Figure 5: Navigation subsystem block diagram. (Yellow: passive components, orange: active components)

6 System Implementation

6.1 Navigation

The implementation of the navigation subsystem is built around two core parts: the U-blox MAX-M10S GNSS module [11], and the STMicroelectronics STM32L412KC microcontroller [16]. See the design trade studies for the reasoning behind our choice of modules. The GPS module and the microcontroller communicate over UART at 9600 baud (which is the default for the GPS module), using U-blox's UBX protocol [17]. Since the GPS module does not contain any non-volatile configuration memory, the microcontroller configures its settings at boot time. For displaying information to the user, a Sharp LS013B7DH03 display [8] is used, which was chosen for its ultra low power consumption and ease of control. The display is connected to the STM32 microcontroller over unidirectional SPI. For saving locations logs and loading map data, the STM32 microcontroller also connects to a removable SD card over bidirectional SPI. User-interface components, specifically the 5-way button joystick and buzzer, are connected to the microcontroller over general-purpose I/O.

6.2 Power and Energy Harvesting

As discussed in the trade studies section (5.1), we chose to pursue a photovoltaic (solar-cell) based design for our energy harvesting. The solar cell, an ANYSOLAR G3 [10], is connected to an ADP5090 boost converter [7], which boosts the fluctuating voltage output by the cell up to levels usable to power the system and charge the battery. The ADP5090 also performs maximum power point tracking to maximize the efficiency of the conversion. The ”system” voltage output by the ADP5090 is then fed into a 3.3V low-dropout voltage regulator [13], which outputs the 3.3V power rail used by the rest of the system. We use a linear regulator to minimize size, and specifically a low-dropout regulator to minimize power losses. In addition to the energy harvesting, we provide a USB-C port to charge the battery directly, via a MCP73831 LiPo charging regulator [12]. When the USB-C cable is connected, the ADP5090 stops charging the

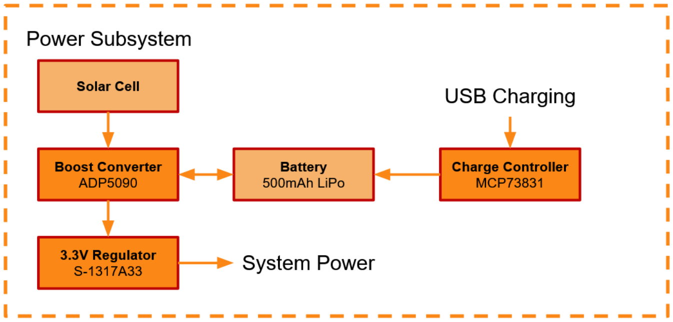

Figure 6: Power subsystem block diagram. (Yellow: passive components, orange: active components)

battery from the solar cell and directly powers the system off of the USB power.

6.3 Firmware Architecture

In order to maintain responsiveness, we needed to be able to process several concurrent tasks. We used the RTIC framework to manage task scheduling and provide asynchronous channels. If a task has no input data or is blocked on publishing data, it will pause and other tasks will take its place.

The display task is the core task that handles user input and the UI. Events from other tasks are shown on the display, possibly causing a state change (e.g. switching to the map). As the user moves, the display task makes requests for map tiles to be redrawn in the background. The GPS task handles data received from the GPS over UART, parses the received messages, and sends the position to the SD card task for logging. The SD card task logs GPS points and loads map tiles on demand, sending them back to the display task once loaded.

The hardware drivers for the GPS receiving and display were written from scratch, and the display driver acts as a hardware layer for the embedded-graphics library, providing a concise API for drawing primitives and text to the screen. The firmware is all written without any heap allocation, to ensure long-term stability. For debugging, the microcontroller interfaces with our ST-Link debugger via test points and makes use of the "defmt" library for low-overhead logging, transmitted over the RTT protocol. ST-Link also allows us to use GDB for conventional step-through debugging. See the "Code Used" citations for links to all of the libraries and example code we used in the firmware.

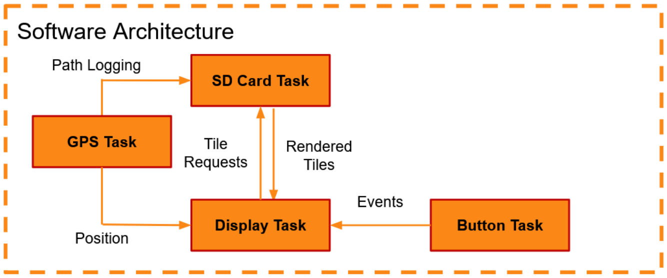

Figure 7: Firmware architecture block diagram

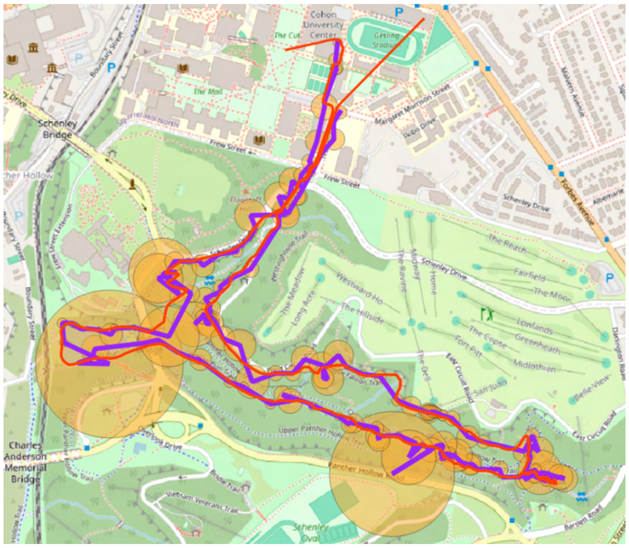

Figure 8: Comparison between Landhopper data (purple) and true position (red), showing error (orange circles)

7 Testing and Validation

7.1 Navigational Accuracy

To validate the accuracy of our GPS tracking, we brought Landhopper on a 1.5 hour walk through nearby Schenley Park and compared it against a "ground truth" GPS module. We used a ZED-F9P GPS-RTK module for known-good position data, and attempted to keep it within a few feet of Landhopper at all times. See Fig. 8 for the comparison between both paths.

After pairing points from the two datasets based on their GPS time-of-week (the internal GPS timekeeping system), we determined that Landhopper has an average positional error of approximately **37 feet**, which is significantly less than our desired average error of 150 feet.

7.2 Battery Life

To validate the battery life of Landhopper, we also recorded battery voltage during the walking test in 7.1. During the first third of the walk, the watch was exposed to bright sunlight. After that, we entered a more densely-wooded area and there was no direct sunlight. As shown in Fig. 9, the average does not decrease during the first third of the graph. This indicates that the solar panel is charging the battery and that we have achieved **unlimited battery life**. During the second half of the graph, the slope indicates approximately 48 hours of battery life. However,

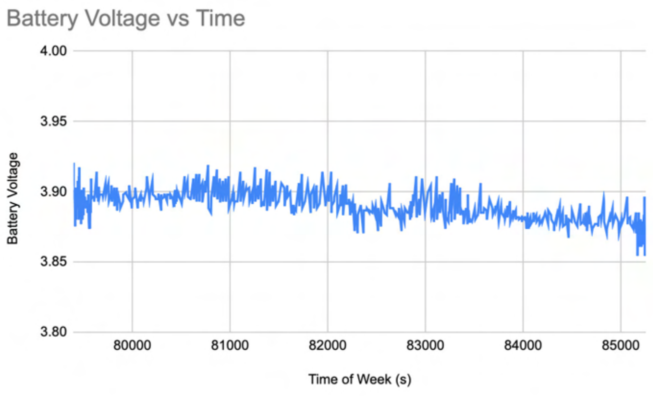

Figure 9: Graph of Landhopper's battery voltage throughout the hike

during this test, the GPS was actually sampling once every 4 seconds rather than once every 60 seconds, meaning that power consumption was dramatically higher than expected. If we adjust for a lower sampling rate (which only draws approximately 1/5th the power, due to idle draw) we attain a battery life of **10 days**, which exceeds the one week requirement.

7.3 Comfort Test

To validate the comfort of the watch, we asked a total of 7 people to briefly wear the watch and answer questions. **100%** of respondents said they could wear the watch regularly, regardless of whether or not they normally wear a watch. When asked to rate the comfort of the watch from 1 to 10, the average response was **7.1**. Feedback was directed at making the watch smaller, improving the screen, or securing the dangling solar panel. Even so, we consider an average score of 7.1 to be a success.

7.4 Weight and Size Test

The main body of Landhopper is **55mm** wide, **49mm** tall, and **15.4mm** thick. When compared with a standard smartwatch (in our test, the Samsung Gear S3), Landhopper is almost exactly the same width and height and about 3-5mm thicker. Landhopper's bulky appearance (as noted in Section 7.3) stems from its rectangular shape. Removing the square corners and making the main body more circular would likely improve the appearance.

Landhopper weighs approximately **60 grams**, substantially less than our 0.5lb target.

8 Project Management

8.1 Schedule

The schedule is essentially the same as in the design report, but pushed back about two weeks to accommodate for delays designing the final PCB.

The schedule is attached to the end of this document in Fig. 10.

8.2 Team Member Responsibilities

Gary handled the initial firmware and the design of the final PCB. Twain handled much of the energy harvesting experimentation, the PCB assembly, and documentation. Carson handled the initial PCB design and the energy harvesting setup, as well as some later firmware tasks.

8.3 Bill of Materials and Budget

We have included the bill of materials of a single Landhopper watch as Table 2. Additionally, we've also included the list of all items we purchased throughout the development of Landhopper as Table 3. Note that this latter table does not include shipping costs, which bring our total cost much closer to $600.

8.4 Risk Management

We had a few risks during design. The first risk came with the energy harvesting. Two of the three approaches came from academic papers we found, so we tried all three approaches to ensure that even if we couldn't replicate the first two, we would still have a fallback option.

We also made our prototype board have all of the necessary features of the final design, but in a much larger and expanded form that is easier to bodge and debug. This allowed us to get a design in early knowing that there would be mistakes that could be fixed. This gave us two things: it meant that in the worst case we could use a bodged version of the prototype for our final demo. It also meant that we could validate our design choices in the real world, before we designed the final (much more time-consuming) PCB.

9 Ethical Issues

Some may have privacy concerns about the watch, but it is almost entirely secure. There is no way to hack into the watch, since the only communication that it does is receiving GPS satellites. It is technically possible to steal a user's location data when they transfer the data to their computer if the bad actor has access to the user's computer, but that cannot be seen as the fault of the watch.

It is possible that, if a person trusts Landhopper to tell them where they are accurately and Landhopper is too far off, that they will be misguided, and the resource that they are relying on will fail them, and could cause them to navigate into unsafe areas or away from the path they may have lost.

The most harmful thing that could be possible is that a bad actor could take Landhopper, place it on a person without their knowledge (say, attached to their car or backpack somehow), and then retrieve Landhopper later and download the person's location data. There are no ways that we could mitigate this risk that would not interfere with the normal use of the watch, but given that there are much more convenient ways of stalking people that do not involve retrieving the device, we find it unlikely that someone will turn to Landhopper for their stalking purposes. For example, people have been known to use Apple AirTags, which can be located remotely [5].

10 Related Work

There are already existing smartwatches with GPS capabilities, though very few of them integrate energy harvesting technology. Notably, the Garmin Instinct 2 utilizes photovoltaic cells built into the screen in order to charge itself. However, at a price point of $400, the Instinct 2 has a significantly higher price point than our materials cost. At the end, the parts total cost of a single Landhopper watch is less than $75 (see 2).

11 Summary

Our system was able to meet the design specifications. We wanted the battery to last for a week with the energy harvesting active for 6 hours a day, and it ended up lasting for 10 days without the energy harvesting working at all. While the solar panel is in the sun, the battery life is indefinite. We wanted the location data to be able to be stored locally, and found that, with a map the size of Pittsburgh, we could still store location data for 25-35 years, which is significantly longer than a week. We wanted the total distance estimate to be within 10% of the real distance for our use case requirement, and our design requirement suggested data points logging within 150 feet. We found that, on our hike through Schenley Park with Landhopper, our average error was around 40ft, which is significantly less than 150ft, and will thus, in many cases, place the user within 10% of their total distance traveled. Finally, we needed the watch to be under half a pound for constant comfortable wear. It ended up being less than 60g, which is well within that bound.

11.1 Future work

We plan to continue working on this project beyond this semester and intend to add some more useful features. Right now, we have a buzzer on the PCB, but no capability to set alarms, so the buzzer is currently unused. Additionally, Landhopper only works outdoors, in GPS-friendly locations. It might be nice to add indoor localization by incorporating Wi-Fi or Bluetooth into our system. Though we did not do this to prioritize battery life and watch size, it may be interesting to see if this can be incorporated into the system and able to be turned off for a "low-power mode" for hiking purposes. Additionally, we will attempt to shrink the PCB further and improve the case aesthetics and comfort. One very big improvement to convenience would be getting USB functionality to work: being able to connect Landhopper directly to a laptop to both charge it and upload/download data would be very convenient.

11.2 Lessons Learned

Not really specific to this application, but it's important to remember to double check PCB schematics before sending them off: while soldering parts onto our initial PCB, we realized that two of the connections were backwards from how we thought they should be (including the SD card slot facing inwards, and the screen which we thought was backwards but turned out to be an ambidextrous connection). Our final PCB design also had some sticking points, where we wish we had added more debugging features to it (more test points and, most of all, a reset button).

We wasted a bit of our time with being stubborn on wanting the piezoelectric solution to work instead of exploring other energy harvesting sources alongside the piezoelectric tiles, and it would have been less stressful had we parallelized some of that work.

Glossary of Acronyms

- GDB: GNU Debugger (GNU is a software project)
- GNSS: Global Navigational Satellite System, which includes GPS
- GPS: Global Positioning System
- GPS-RTK: GPS with Real-time Kinematics
- LDO: Low-dropout voltage regulator
- PCB: Printed Circuit Board
- RTT: Real Time Transfer, a debug I/O protocol developed by Segger
- SPI: Serial Peripheral Interface, a common serial protocol
- UART: Universal Asynchronous Receiver-Transmitter, a common serial protocol

References

[1] Mingjing Cai and Wei-Hsin Liao. "A Coaxial Wrist-Worn Energy Harvester for Self-Powered Internet of Things Sensors". In: *IEEE Internet of Things Journal* 10.1 (Jan. 2022), pp. 133–143. DOI: `10.1109/JIOT.2022.3202166`.

[2] "Is an E-Ink Display Right for Your Next Project?" In: *Maker.io by DigiKey* (2022). URL: `https://www.digikey.com/en/maker/blogs/2022/is-an-e-ink-display-right-for-your-next-project`.

[3] Dong Wook Kim et al. “Material aspects of triboelectric energy generation and sensor”. In: *NPG Asia Materials* 12.6 (Jan. 2020). DOI: `10.1038/s41427-019-0176-0`.

[4] *Kit Carson Peak via Challenger Point.* 14ers.com, 2024. URL: `https://www.14ers.com/route.php?route=kitc2`.

[5] Gordon Ung. “I used Apple AirTag to track my wife and kids. Here’s what I learned”. In: *PC World* (2022). URL: `https://www.pcworld.com/article/608037/i-used-apple-airtag-to-track-my-wife-and-kids-heres-what-i-learned.html`.

[6] Biao Wang et al. “Characterization of Wrist Motions and Bionic Energy Harvesting for Wrist Wearables”. In: *IEEE Internet of Things Journal* 9.21 (Nov. 2022), pp. 21147–21156. DOI: `10.1109/JIOT.2022.3178719`.

Datasheets

[7] *ADP5090 Datasheet.* Analog Devices Inc. 2015. URL: `https://www.analog.com/media/en/technical-documentation/data-sheets/ADP5090.pdf`.

[8] *Device Specification for LS013B7DH03.* Sharp Corporation. 2014. URL: `https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/929/LS013B7DH03_Spec.pdf`.

[9] *EM3215D2450Y1R Specification.* Top Once Motor Manufacture. 2022. URL: `https://cdn.sparkfun.com/assets/e/b/2/8/3/OT-EM3215_DC_Brushless_Gimbal_Motor_3_Phase_Datasheet.pdf`.

[10] *KXOB081K06TF Datasheet.* Anysolar Ltd. 2023. URL: `https://waf-e.dubudisk.com/anysolar.dubuplus.com/techsupport@anysolar.biz/O18Azec/DubuDisk/www/Gen3/KXOB081K06TF%20DATA%20SHEET%2020210127.pdf`.

[11] *MAX-M10S Datasheet.* U-blox AG. 2023. URL: `https://www.u-blox.com/sites/default/files/MAX-M10S_DataSheet_UBX-20035208.pdf`.

[12] *MCP73831/2 Datasheet.* Microchip Techonlogy Inc. 2020. URL: `https://ww1.microchip.com/downloads/en/DeviceDoc/MCP73831-Family-Data-Sheet-DS20001984H.pdf`.

[13] *S-1317 Series Datasheet.* ABLIC Inc. 2016. URL: `https://www.ablic.com/en/doc/datasheet/voltage_regulator/S1317_E.pdf`.

[14] *Sharp Memory LCD Suite.* Sharp Microelectronics. 2015. URL: `https://www.mouser.com/datasheet/2/365/Sharp_Memory_LCD_Brochure_2015-746065.pdf`.

[15] *STM32L series.* STMicroelectronics NV. 2022. URL: `https://www.st.com/resource/en/brochure/brstm32ulp.pdf`.

[16] *STM32L412xx Datasheet.* STMicroelectronics NV. 2022. URL: `https://www.st.com/resource/en/datasheet/stm32l412kb.pdf`.

[17] *U-blox M10 standard precision GNSS firmware, Interface Description.* U-blox AG. 2021. URL: `https://content.u-blox.com/sites/default/files/u-blox%20M10-SPG-5.00_InterfaceDescription_UBX-20053845.pdf`.

Code Used

[18] James Waples et al. *Embedded-Graphics and Embedded-Text.* URL: `https://github.com/embedded-graphics/embedded-graphics`.

[19] Lokathor et al. *Tinyvec.* URL: `https://github.com/Lokathor/tinyvec`.

[20] Scott Mabin et al. *STM32L4 Drivers.* URL: `https://github.com/stm32-rs/stm32l4xx-hal`.

[21] RTIC Developers. *RTIC, RTIC-Sync, and RTIC-Monotonics.* URL: `https://github.com/rtic-rs/rtic`.

[22] Rust Embedded Working Group. *Cortex M Drivers.* URL: `https://github.com/rust-embedded/cortex-m`.

[23] Ferrous Systems. *Defmt and Defmt-BRTT.* URL: `https://github.com/knurling-rs/defmt`.

[24] Ferrous Systems. *Knurling App Template.* URL: `https://github.com/knurling-rs/app-template`.

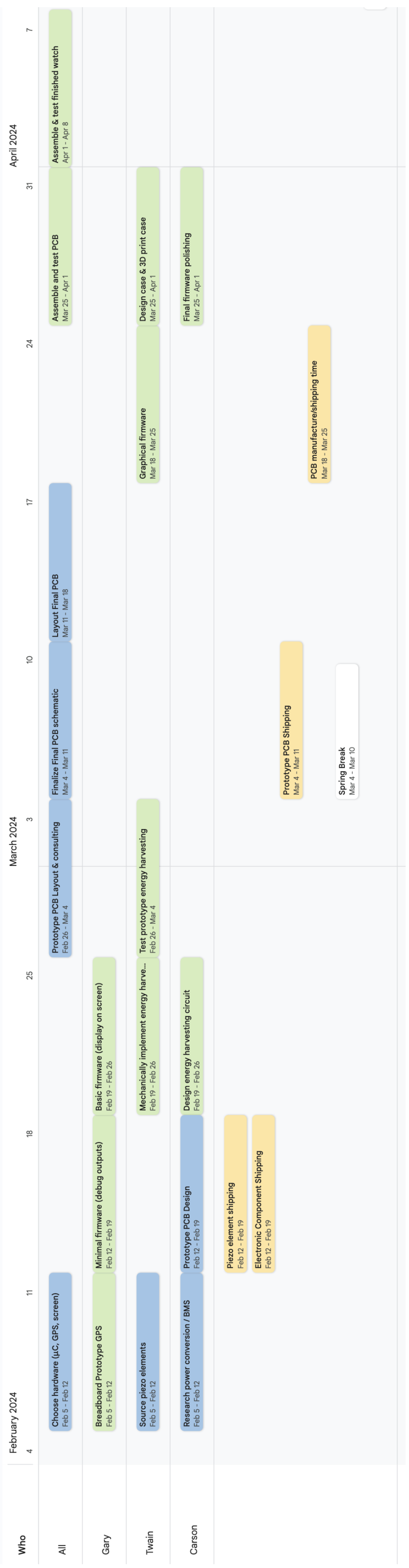

Figure 10: Gantt Chart

Table 2: Bill of materials for a single Landhopper watch

Name	Qty	Unit Price	Total Price	Manufacturer	Link
Final PCB	1	\$25.27 for 5	\$5.05	JLCPCB	
Sharp Memory Display	1	\$11.91	\$11.91	Sharp	Link
STM32L412KBUx Microcontroller	1	\$4.20	\$4.20	STMicroelectronics	Link
MAX-M10S GPS Module	1	\$21.00	\$21.00	U-Blox	Link
5-way Joystick	1	\$3.15	\$3.15	E-Switch	Link
4mm Buzzer	1	\$1.00	\$1.00	(generic)	(stock)
Solar cell	1	\$5.92	\$5.92	ANYSOLAR Ltd	Link
MCP73831 LiPo charger	1	\$0.76	\$0.76	Microchip Technology	Link
ADP5090 Boost Converter	1	\$6.49	\$6.49	Analog Devices, Inc.	Link
3.3V LDO Regulator	1	\$0.98	\$0.98	ABLIC Inc.	Link
32.768kHz Crystal	1	\$0.72	\$0.72	NDK America, Inc.	Link
2N7002 NMOSFET	1	\$0.15	\$0.15	Diotec Semiconductor	Link
Si2319CDS PMOSFET	1	\$0.55	\$0.55	Vishay Siliconix	Link
MCP6401 Op-amp	1	\$0.46	\$0.46	Microchip Technology	Link
PMEG4005 Diode	1	\$0.23	\$0.23	Nexperia Inc	Link
ESD protection diode	1	\$0.43	\$0.43	Littelfuse, Inc.	Link
74AHC1G04 Inverter	1	\$0.25	\$0.25	Texas Instruments	Link
5.1kΩ Resistor	2	\$0.05	\$0.10	(generic)	(stock)
33kΩ Resistor	1	\$0.05	\$0.05	(generic)	(stock)
10kΩ Resistor	4	\$0.05	\$0.20	(generic)	(stock)
20kΩ Resistor	1	\$0.05	\$0.05	(generic)	(stock)
82kΩ Resistor	1	\$0.05	\$0.05	(generic)	(stock)
1MΩ Resistor	2	\$0.05	\$0.10	(generic)	(stock)
3.59MΩ Resistor	1	\$0.05	\$0.05	(generic)	(stock)
4.06MΩ Resistor	1	\$0.05	\$0.05	(generic)	(stock)
4.47MΩ Resistor	1	\$0.05	\$0.05	(generic)	(stock)
5.55MΩ Resistor	1	\$0.05	\$0.05	(generic)	(stock)
6.08MΩ Resistor	1	\$0.05	\$0.05	(generic)	(stock)
6.45MΩ Resistor	1	\$0.05	\$0.05	(generic)	(stock)
6.50MΩ Resistor	1	\$0.05	\$0.05	(generic)	(stock)
16.9MΩ Resistor	1	\$0.05	\$0.05	(generic)	(stock)
4.3pF Capacitor	2	\$0.05	\$0.10	Yageo	Link
100nF Capacitor	5	\$0.05	\$0.25	(generic)	(stock)
4.7uF Capacitor	8	\$0.04	\$0.32	Samsung Electro-Mechanics	Link
10uF Capacitor	1	\$0.05	\$0.05	(generic)	(stock)
10nF Capacitor	1	\$0.05	\$0.05	(generic)	(stock)
47uF Capacitor	1	\$0.45	\$0.45	Murata Electronics	Link
22uH Inductor	1	\$1.53	\$1.53	Würth Elektronik	Link
U.FL connector	1	\$0.75	\$0.75	Adafruit Industries LLC	(stock)
USB-C Port	1	\$0.81	\$0.81	GCT	Link
Battery connector	1	\$0.14	\$0.14	JST Sales America, Inc.	Link
Screen connector	1	\$0.94	\$0.94	Panasonic	Link
MicroSD Slot	1	\$1.20	\$1.20	GCT	Link
Watch band	1	\$8.99 for 3	\$3.00	Relting	Link
Grand Total			\$73.79		

Table 3: All purchases made during development, including parts unused in the final version

Name	Qty	Unit Price	Total Price	Manufacturer	Link
GPS Module Breakout	1	$44.95	$44.95	SparkFun Electronics	Link
Chip Antenna 1	2	$4.95	$9.90	Adafruit Industries LLC	Link
Chip Antenna 2	2	$3.95	$7.90	Adafruit Industries LLC	Link
uFL to SMA antenna	2	$6.03	$12.06	Molex	Link
Gimbal Motor	2	$29.95	$59.90	SparkFun Electronics	Link
Boost Converter	4	$6.49	$25.96	Analog Devices Inc.	Link
Active Rectifier	5	$2.63	$13.15	Texas Instruments	Link
1:1 1mH transformer	5	$0.95	$4.75	Bourns Inc.	Link
Schottky Diode	20	$0.23	$4.60	Nexperia USA Inc.	Link
3.3V LDO	5	$0.98	$4.90	ABLIC Inc.	Link
15mm width Piezo Tile	3	$6.49	$19.47	uxcell	Link
Metal Sheet	1	$30.89	$30.89	Tsmhisd	Link
Watch band	1	$8.99	$8.99	Relting	Link
2-Part Epoxy	1	$12.20	$12.20	J-B Weld	Link
Metric Machine Screws	1	$18.99	$18.99	HVAZId	Link
Ball Bearing Kit	1	$9.99	$9.99	weideer	Link
1Mohm trimmer potentiometer	5	$1.77	$8.85	Bourns Inc	Link
22uH inductor	3	$1.53	$4.59	Würth Elektronik	Link
LiPo battery charger	4	$0.76	$3.04	Microchip Technology	Link
GPS module	2	$21.00	$42.00	U-blox	Link
Sharp Memory Display	2	$11.91	$23.82	Sharp	Link
STM32L4 Microcontroller	4	$4.62	$18.48	STMicroelectronics	Link
USB-C Female Connector	4	$0.67	$2.68	GCT	Link
Crystal Oscillator	4	$0.72	$2.88	NDK America, Inc.	Link
4.3pF capacitor	10	$0.25	$2.52	KEMET	Link
Solar	2	$5.92	$11.84	ANYSOLAR Ltd	Link
Screen Connector	2	$0.94	$1.88	Panasonic	Link
Battery Connector	10	$0.14	$1.36	JST Sales America, Inc.	Link
MicroSD Slot	2	$1.20	$2.40	GCT	Link
Prototype PCB	5		$18.90	JLCPCB	
ESD Protection Diode	5	$0.43	$2.15	Littelfuse, Inc.	Link
Motor Connector	5	$0.27	$1.35	Molex	Link
USB-C Female Connector but better	5	$0.81	$4.05	GCT	Link
P-channel mosfet	5	$0.55	$2.75	Vishay Siliconix	Link
N-channel mosfet	5	$0.15	$0.75	Diotec Semiconductor	Link
Slide switch	5	$0.67	$3.35	C&K	Link
Inverter	10	$0.25	$2.51	Texas Instruments	Link
Op-amp	5	$0.46	$2.30	Microchip Technology	Link
Final PCB	5		$25.27	JLCPCB	
5-way Joystick	3	$3.15	$9.45	E-Switch	Link
STM32 Microcontroller in QFN package	3	$4.20	$12.60	STMicroelectronics	Link
Sharp Mem Display spares	2	$11.91	$23.82	Sharp	Link
MicroSD Slot spares	2	$3.11	$6.22	Wurth	Link
GPS module spare	1	$21.00	$21.00	U-blox	Link
Boost Converter spares	3	$6.49	$19.47	Analog Devices Inc.	Link
Op-amp spares	5	$0.46	$2.30	Microchip Technology	Link
Antenna spares	2	$3.95	$7.90	Adafruit Industries LLC	Link
Solar cell spares	2	$5.92	$11.84	ANYSOLAR Ltd	Link
4.3pF caps in 0603	10	$0.05	$0.49	YAGEO	Link
4.7uF caps in 0603	10	$0.04	$0.41	Samsung Electro-Mechanics	Link
47uF caps in 0603	5	$0.45	$2.25	Murata Electronics	Link
500mAh battery	1	$7.99	$7.99	Liter Energybattery	Link