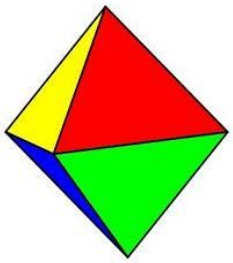
MIE438 Project Proposal

HourCache

Group 20

Cameron Witkowski - 1005533166
Mingshi Chi - 1004881096
Wei Hai Cui - 1005337324
Xing Hao Li - 1004906943
Yannis He - 1004769707

Introduction:



The pandemic has thrown a curveball into our lives - with everything moving virtually, our daily routines have been disrupted, and it's hard to keep track of time when we're stuck indoors all day. As a result, we wanted to create a method of tracking how much time we're spending on different activities during the day. Drawing inspiration from a simple sand hourglass, we decided on creating HourCache.

Project Description:

HourCache is a hardware based time-tracking octahedron that automatically tracks activities, such as studying or socializing. When placed on a side, the HourCache starts tracking the time spent in that state and records it so that we can see exactly how our time is being spent.

Unlike the traditional time tracking method, which requires people to open a software or notebook and manually log the time spent, simply flipping the HourCache is all you need to do for tracking a variety of tasks. HourCache reduces the steps and conscious effort to track an activity. By easing the time tracking process, people are more willing to record their time usage that used to be considered as "negligible", such as replying to messages on their phone in the middle of work. People easily get distracted and spend hours on the internet while thinking "I can reply to this message in 20 seconds".

The data will be synced through WiFi or Bluetooth and can be visualized on any device, helping people monitor their usage habits and see exactly where their time goes.

**data visualization requires high-level software engineering and data analysis, which is beyond the scope of this course*

Timeline:

Task	**Due Date**	**Task Description**
Finalize design and order parts	Feb 6th	Order parts on Aliexpress or Amazon depending on expected shipping dates
Prototype circuitry	Feb 24st	Design all wiring and battery optimization before the ordered components arrive
Prototype software communication	Feb 24st	Write code that corresponds to circuit prototype to control the hardware and implement communications
Design hardware placements	Feb 28th	CAD the outer casing to hold the components in

and outer display		place
Make physical prototype	March 7th	3D print the prototype casing
Testing combined hardware and software components	March 8th → March 28th	Combining all parts to test for functionality and improve on errors
Make final product	March 28th	Design & assemble into the final deliverable
Final Report	April 4th → due April 7th	Write up the report and provide documentation on the process

Component Specifications and Considerations:

Microcontroller

We narrowed down our alternatives to ESP32 board [1] and Arduino MKR1010 [2]. Both of these boards have a small size, built-in wifi and Bluetooth modules, and options for connecting to LiPo batteries. The main advantage of the ESP32 board is the computing power it offers. It surpasses Arduino in both clock speed, flash memory, as well as SRAM. The advantage of Arduino is it has a wide range of online community support, including full tutorials and libraries.

Since our project will likely be powered by battery for the majority of the time, one key component would be to identify the power consumption of the microcontroller, which would depend on the intensity of the wifi/bluetooth usage. This would be something we need to test ourselves and use that to determine the size of the associated battery.

Orientation Detection

At a high level, we must design a way of robustly measuring which side the HourCache. We are currently in the process of deciding between two systems for doing this.

1. Accelerometer

An accelerometer would be used to measure static forces acting on the HourCache. This would be used at rest (the normal operating state) when the only static force is gravity. A calibration process would be needed to classify this 3D force vector as representing one of eight sides. A promising option is an ADXL345 chip [3]. This chip costs only $15 from Amazon and only one would be needed for the complete product. It utilizes the I2C communication protocol and outputs force readings along 3 axes.

2. Pressure pads

Another way of detecting the HourCache's orientation is to place a pressure pad on each side, which will send a signal to the microcontroller when pressure is applied. A good example is the RP-S40-ST [4]. This sensor works like a voltage divider, and is exceedingly simple in its operation—see appendix [5]. The upside of this strategy is the one-hot translation of orientation data, presumably leading to greater robustness, and no calibration required. The downside is that 7 sensors are required (compared to one accelerometer), each needing one analog pin on the microcontroller and costing $15.

Communication

We have converged on using Bluetooth communication from Arduino to phone after considering other alternatives such as LCD display screen, and wifi communication through a web app. Appendix [5] shows the benefits and drawbacks of each method of communication. A simple APK android app will be made to display processed data from the HourCache to the user.
The current plan is to sync data to other external devices, such as PCs and phones, through either Bluetooth or WiFi protocol. However, SD cards could be used to store data as a backup plan if wireless data transformation is too challenging.

Battery Efficiency

In order to maximize battery life, the software will let the microcontroller go to sleep between states, and wake once again when the state changes. To keep track of time while sleeping, a real-time clock module is connected to the microcontroller over I2C with some featuring a standalone battery life of over 9 years [6]. Since most microcontrollers only feature 1 I2C input, we may use a multiplexer to connect both the RTC and the accelerometer.

User Feedback

With haptic feedback through a vibration motor, as well as auditory feedback, the octahedron itself will let users know when a state has been changed. The vibration motor and buzzer can simply be connected to the digital pins of the microcontroller.

Packaging & Misc Components

In order to fit all the components together, we've decided on a 3D printed shell that houses everything. A key consideration is the center of mass - since the octahedron can be used in different orientations, the center of mass should be as close to the geometrical center of the octahedron as possible. Considering repairability, the components will be fixed with machine screws, while the microcontroller and other standalone modules will attach to a solderable breadboard using headers.

Conclusion:

Just as a ship at sea is quickly lost without a compass, it is not easy to focus amid a storm of notifications, headlines, and distractions in our insulating, information-saturated world. To this end, the value of effective productivity tools cannot be understated and is only increasing. While the HourCache is rather simple in concept, it offers a powerful framework for time management if used diligently. By tracking one's time spent across different activities in an intuitive and user-friendly way, the HourCache serves as a rewarding companion to a regimented lifestyle. After all, a storm is no match for a captain with the right mentality and set of tools.

Appendix

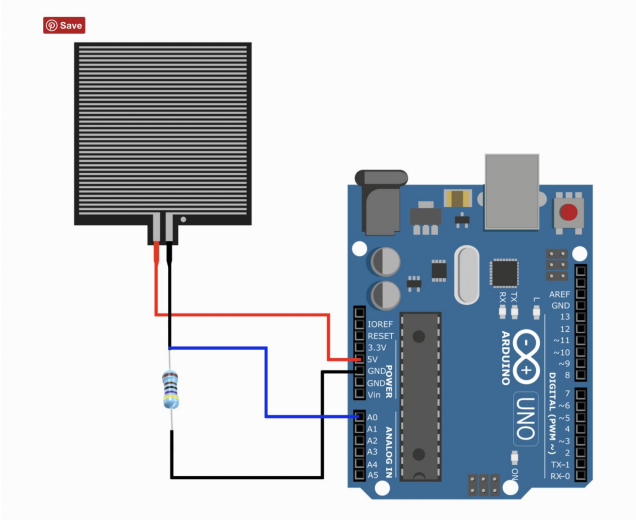
[1] SparkFun ESP32 Thing
https://www.sparkfun.com/products/13907

[2] Arduino MKR Wifi 1010
https://store.arduino.cc/usa/mkr-wifi-1010

[3] Accelerometer IC.
https://www.amazon.ca/ADXL345-Digital-Three-axis-Acceleration-Transmission/dp/B081VBXTJP/

[4] Pressure Pad
https://www.amazon.ca/RP-S40-ST-Accuracy-Pressure-Intelligent-Induction/dp/B07FCLV5BJ/

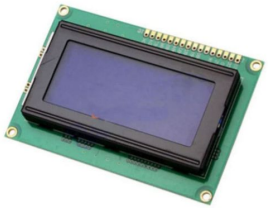
[5] Pressure Pad Operation



[5] Pros and Cons of different methods of communication

Property	**Wifi through a web app**	**Bluetooth**	**LCD screen**
Scalability	High scalability as different features can be added onto the web app to improve upon the design features	Can create a simple app that connects to Bluetooth and receives info from the Arduino. (XML code and Java Class)only works on Android tho for an app Need to make an APK regardless it seems. iOS also compatible but takes a bit of work to	Hard to scale if the screen has low pixel density. Limited by amount of info we can show at a time This would also mean that a side or a portion of the device is taken up by a screen

		get it to run on iOS Scalability is high as we can add different commands and features on the app	
Risk factor	High risk. Web development takes time. There is an option to use a preset website for Arduino communication but no member has experience with this Another possibility is to make our own website to host the app which will take a lot of time (High risk in time management)	mid risk. All members have had experience with Bluetooth using Arduino before. However, no one has experience with making an Arduino to phone APK. Some have experience with android development	Low risk as many of the team has worked with LCD screens before for displaying data
Ease of connectivity	Hard (must manually set IP addresses)	Easy to connect	Easy to connect
Amount of coding and setup	A lot of JS and set up. Server set up, IP address. Static or permanent. Better for letting all team members use easily	The bulk of the setup will be learning how to make the APK. all members already have coding Bluetooth to Arduino communication experience.	Very minimal. Many screens already have their own libraries as well as Arduino premade libraries that allow for customization of the display
User experience	Highly promising in user experience as users from all operating systems can use this. Can also design for user interface	The app design is dynamic which means we can design it nicely with a good user interface	The user must pick up the device and orientate it to look at the data which is not ideal for calibration or user experience Less information can be shown on the screen if the pixel density is smaller. The chosen screen, if the device is to be user friendly with a good interface would have to be similar to a

			TFT LCD module such as this  A typical LCD display screen although can be optimized to show all the necessary data would be hard on the eyes to view 

[6] Real Time Clock module example from sparkfun - https://www.sparkfun.com/products/12708