Physical Time Tracker

Lisa (Hyuni) Ko

University of Texas at Austin, School of Information

I379C: Capstone

James Howison

March 22, 2024

Executive Summary

150-200 words

Project Overview

This project is dedicated to design and build a physical time-tracking system, specifically for meeting environments. Drawing inspiration from the principles behind the Time Timer, this system aims to introduce an innovative approach to monitoring and managing time during meetings through a visible and tangible representation of elapsed time. The time tracker's objective is to revolutionize the way time is controlled during meetings, which will improve communication, and encourage a more interactive approach to time tracking. The project Aims:

- Researching the existing of time-management tools in order to collect insights that will direct the creation of our time-tracking system
- Designing a physical system with movable time segments that reflect the meeting environments
- Experimenting and putting into practice coding techniques on hardware platforms in order to bring the conceptualized design, ensuring functionality and user engagement.

Project Setting

This project focuses on an exploratory research and development framework, with the use of Arduino technology for hardware coding.

- Research and Design: Investigating existing time-tracking devices and methodologies, defining detailed requirements and features for the system, and conceptualizing the physical components and interaction mechanisms.

- Prototype Development: Building functioning prototypes using the Arduino platform, integrating sensors, actuators, and indicators, and testing individual components for functionality.
- Coding and Testing: Developing the logic for time tracking, implementing code to trigger events based on elapsed time, conducting comprehensive testing of the entire system, and refining the system based on testing results.
- Documentation: Documenting hardware connections, code structure, and troubleshooting steps, creating a user manual with clear instructions on operating the system, and producing comprehensive documentation of the project.

In alignment with the project's goals and stakeholder requirements, the time-tracking system's design is subject to specific critical conditions:

- Timer should be intersubject: The system is tasked with promoting an environment of mutual time awareness, where both presenters and attendees are continuously informed about the time spent and the time remaining for each agenda item. This feature underscores the significance of real-time feedback, empowering participants to allocate their time wisely and adjust their input to fit the meeting's flow.
- Innovative Time Visualization: Instead of following conventional time display approaches, the system is intended to visually represent the passing of time. The system will use creative visual signals to indicate the passage of time within the framework of traditional clock faces, so that the user understands the time chunk. The goal of this approach is to provide a new angle on time tracking by simplifying and enhancing the

visual impact of the meeting segments.

Project Timeline

- Week 1-4: Research existing Arduino projects related to time tracking, define detailed requirements and features for the system
- Week 5: Set up the development environment and familiarize with the arduino environment.
- Week 6: Develop the initial code to establish basic functionality, focusing on the integration and operation of the system's physical elements.
- Week 8-9: Create the core time-tracking logic, implementing code that triggers specific events as time advances. Initiate the integration of this logic with the physical components assembled in the prototype.
- Week 10-11: Conduct comprehensive testing, gather feedback, refine the system, document hardware connections, code structure, and troubleshooting steps, and create a user manual.
- Week 12-13: Finalize the poster for submission and work on the final report.

Relevance of UT iSchool Informatics Courses

The information and abilities gained from three essential informatics courses helped create a physical time tracker.

I304: Programming for Informatics

By utilizing the programming language (python) learned in “Programming for Informatics”, I was able to enable the creation of custom scripts, which form the foundation of the time tracking logic. Although the time tracker project utilized Arduino as its hardware, which primarily uses C/C++, Python is a very useful language for activities like simulation, data processing, and communication in the time-tracker project. Also, the debugging skills learned throughout the course were important in ensuring the reliability of the time-tracking system. The implementation of analytical debugging techniques helped promptly detect logical mistakes or unforeseen patterns in the Arduino coding. Debugging involves not only correcting mistakes but also confirming and validating the functionality of the system. The course's instruction on programming was crucial in combining the time-tracking system's hardware and software components. It enables the system to be capable of taking in data, processing it, and producing hardware-compatible output signals.

I320D: Open Source Software Development

The "Open Source Software Development" course offered insights and practical experience in utilizing open-source tools and resources when getting the Arduino hardware code for time tracking logic. The nature of open-source software development encourages creativity by allowing developers to build upon existing projects and customize solutions to meet specific needs. Including pre-existing Arduino open-source code into the time-tracking project is an effective way to increase functionality and development. This approach offers a chance to build on other people's work and give back to the ecosystem, which is exactly in line with the philosophy of open-source software development. Along with the codebase, the project's documentation—which includes wikis, various Markdown documents, and the README file—is also hosted on GitHub. All stakeholders, including the project supervisor, can learn about the project's objectives, architecture, setup procedures, and usage from this documentation. During my project, the Open Source Software Development course's practical application of concepts and abilities is shown by utilizing pre-existing open-source Arduino code to control LED strips and create timers. This highlights the course's relevance to the project.

I310U: Introduction to User Experience Design

The fundamental design and interface elements of the time-tracking system were designed and refined with the help of the "Introduction to User Experience Design" course, which provided insights into user-centered design principles and processes. In terms of user requirements and research, understanding user needs and preferences through research and analysis is a key component of design. One of the most important steps in the design process is collecting and analyzing user needs and preferences through detailed research and study. The principle of user experience design was carefully considered during the time tracking system's design process to make sure the time tracker meets the needs and preferences of its users. Also, prototyping design plays a huge role in the time-tracker project since it can transform concepts into concrete solutions. The concept evolved from initial sketches to a tangible form that could be interacted with, evaluated, and improved using LED strips and Arduino hardware programming. In summary, the "Introduction to User Experience Design" course offered significant knowledge and abilities in the principles and methods of user-centered design. The alignment of my project with these UX design principles is clear, underscoring its dedication to developing a user-focused, physical time-tracking system.

Deliverables

500-600 words (due final)

- Detailed design specifications for physical components.
- Functioning prototypes utilizing Arduino
- Finalized time-tracking system with interactive features.

- Codebase for microcontroller integration.
- User manual for operating and customizing the system.
- Comprehensive documentation of the project.

Contribution to Career Goals

350-500 words

discuss how this capstone experience contributes to my career goals (leaving it for final report)