

Physical Time Tracker for Meetings

Lisa (Hyuni) Ko

University of Texas at Austin, School of Information

I379C: Capstone

James Howison

April 21, 2024

Executive Summary

The I379C capstone project, “Physical Time Tracker for Meetings” at the University of Texas was undertaken with the goal of creating a physical time-tracking system to improve the interactivity and efficiency of meetings. Even though the project's initial goal was to create a fully functional time-tracking device, the focus shifted toward a more exploratory approach, emphasizing the development of hardware coding skills using Arduino uno. These elements formed the foundation for a sophisticated system designed to provide a visual and tangible representation of time, thereby improving communication and meeting management. This exploration has not only advanced my technical capabilities but has also reinforced the importance of adaptability and lifelong learning in technology development. The skills and insights gained from this project are instrumental in paving the way for my future career as a data analyst, equipping me with the tools necessary to innovate and excel in the field of technology.

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

This project initially aimed to design and build a physical time-tracking system for meeting environments, inspired by the Time Timer. However, it turned into an exploratory research with an emphasis on hardware coding for informatics and Arduino in particular. While the original goal was to revolutionize how time is managed during meetings to enhance

communication and interaction, the actual outcome was a comprehensive exploration of coding techniques and hardware integration. The physical time-tracking system was not fully developed but was conceptualized and represented through preliminary sketches. This shift in focus has provided valuable insights into the practical application of hardware and software in designing interactive tools.

Intro to Hardware Coding for Informatics Student Guide

I made a basic code guide and description for each code snippet I used for the time tracking. This manual could be used as a basic resource and reference for students who want to learn hardware programming, especially timer features displaying on 4-digit 7-segment boards on Arduino. It covers fundamental subjects like configuring the Arduino environment, understanding basic coding concepts, and providing detailed instructions for integrating basic sensors and outputs. ISchool students with little to no prior expertise with hardware projects, arduino experience will be able to easily understand the meaning of each code and hardware components.

Detailed design sketches for physical components and time tracker concept

Although the complete time-tracking system was not finalized, detailed design specifications were developed. These sketches show and outline the necessary components, such as the 4-digit 7-segment display, layout of the track and the positioning of the train, the arrangement of the streetlights showing the time chunk set for the meeting. In addition to developing detailed design specifications, I also created a concept sketch of a train time tracker tailored for the iSchool conference room. It visually represents the device's virtual design and functionality, providing a clear blueprint for potential implementation and further development.

Codebase for Arduino hardware integration.

The project has created a detailed codebase specifically designed for Arduino based timer. The Arduino programming environment uses C/C++ language. It includes initialization routines, main loop logic, and function definitions specifically optimized for low-power, high-performance operation on Arduino boards. The Arduino project within the codebase also uses digital and analog sensors, display, and user input devices such as buttons. The code facilitates communication between these hardware components, using Arduino's digital and analog input pins.

Comprehensive documentation of the project.

The project's comprehensive documentation includes research notes, design iterations, testing results, and reflections on the development process. This whole documentation process not only provided a repository of files, but also a narrative that captures all ups and downs of the project. Research notes provided initial explorations and researches of the existing time tracking methods during meetings, and insights gained from initial brainstorming sessions. Design sketches of the time tracker included early concepts and refined design iterations. Reflections I wrote every week captured every challenge I faced and solutions. Through writing reflections and making notes every week, I was able to trace down lessons I learned throughout the project, limitations I found and potential areas for future, further development. This documentation not only captures the project's evolution but also provides critical insights into the challenges and solutions encountered, offering a valuable learning tool for future projects.

Contribution to Career Goals

Although being primarily focused on hardware coding with Arduino, the capstone project experience has been crucial in helping me prepare for a career as a data analyst. In addition to improving my technical abilities, this experience has made me more aware of the need of ongoing education and flexibility, both of which are essential in the ever changing field of data science.

Even though hardware coding may not directly correlate with the typical job description and responsibilities of a data analyst, this project significantly enhanced my analytical skills and willingness to learn new things. Engaging with real-time timer code data and interactive Arduino coding, especially while working with an 4-digit 7-segment display on Arduino, allowed me to gain hands-on experience in merging two different codes and transforming open source code into actionable information for my project. I believe this practical experience is directly applicable to data analytics, where understanding existing data projects and manipulating code streams are important for generating data insights.

In addition to the technical skills I learned, this project also highlighted the value of adaptability and lifelong learning within the technology landscape. Given how fast new coding tools and techniques can be learned and applied, adaptability has become one of the important skills in the ever-changing world of technology. My learning of Arduino coding was a prime example of this, marking my very first encounter with hardware programming. Through this experience, I not only learned technical skills but also learned how to adapt to entirely new challenges and technologies.

When learning how to code on Arduino for the first time in my life, I learned not only the hardware coding skills but also how to adapt to new challenges and technologies I have never

faced before. In my future career as a data analyst, the ability to learn quickly and adapt to new analytical tools and methods will be valuable. Experience of learning new things, problem-solving strategies, mirroring the dynamic challenges will all help when I become a data analyst trying to solve problems and adjust to rapid acquisition of new data analytics skills.

Moreover, this project has helped me continually engage with new ideas and technologies, enabling me to practice my problem solving skills. As I learn more and work more on a career in data analytics, these skills will be important for navigating the complexities of big data environments and delivering insightful solutions out of different problems.