Happy Little Timer (004-1 ECE342-W23)
Tyler Robertson
Gabe Ryley

Executive Summary

Our partnership was tasked with building a custom timer that fulfilled a variety of customer requirements. The timer needed to be accurate, intuitive, and safe. The timer also needed to have three configurable brightness levels, and an alarm at an easy-to-hear frequency. The timer, as per our own desired challenges, was to be settable through a user interface, and to be easy to use by hard-of-hearing users. With such a tall challenge ahead of us, we decided to develop the timer over three phases.

Our timeline was created in an excel spreadsheet, and is not suitable to be placed in a text document. The link can be found HERE.

The first key phase was our Design phase. In this phase, we needed to determine how we were going to implement the requirements, what devices we were going to use, and how we were going to link them all together. We first designed a “black box diagram”, abstracting our design to a very high level. With a limitation on how many blocks we could make, we decided to break the design into four modules: Enclosure, PCB, Arduino, and User Interface. We planned out our connections between the modules as well. In this phase, we also sketched out electrical schematics and our enclosure to prepare for construction.

The second phase was our Build phase. Each of us set out to construct our designated modules individually, but we constantly worked with each other for advice and assistance when needed. This is where we found our TM1637 and NeoPixel Ring devices, choosing them for their ease of use and integration with Arduino libraries. We also designed our specific voltage regulator and MOSFET audio driver circuits in this phase, electing to house them on our PCB. The bulk of our time was spent in this phase, as we developed the Arduino Code, constructed our electrical schematics onto our PCB, and built the enclosure and user interface.

The last phase of our development was our Integration phase. In this phase we implemented connectors to plug our User Interface and Arduino into our PCB, and securely placed them into our enclosure, creating a distinct, isolated system. In this phase, we performed copious amounts of testing to ensure that our engineering and customer requirements were both met.

Throughout the design process, we learned industry and interpersonal skills alike. Programming an Arduino and designing a PCB for the first time were both invaluable experiences. We also struggled with designing our enclosure, since exact measurements are hard to come by. Equally importantly, we learned the value of communication. Both of us had different expectations and visualizations of our project, and we worked together to match these distinct views. We also had to be accommodating for when things went wrong, and inform each other about decisions to change our designs.