

Our task was to create a programmable timer that would generate an alarm tone after counting down to zero. The alarm had to operate at 440 Hz, the display had to be dimmable with three distinct brightness levels and the display had to count both minutes and seconds. The system also had to have buttons so that the users could operate the display and program it to their desired time.

Creating a timer using a microcontroller was not difficult but driving 4 seven segment displays with a single microcontroller is. The limited number of I/O pins meant that we would have to be resourceful when deciding our pin assignments. We utilized an ATmega328P because of its fairly high number of I/O pins and its similarity to the Arduino Uno platform. This allowed us to test the design using the Arduino IDE and the Arduino Uno platform, simplifying the development phase.

To solve our pin problem we introduced multiplexing into our design. A 4 digit number representing our 4 digit counter was stored into an array 4 digits wide and the program cycled through each element of that array every 800 microseconds. Every segment on our display was wired in parallel to the respective segments in the other display. All the digits were wired to one I/O pin. The digit was only ignited when the program had selected its respective element in the array.

Once we figured out how to display a 4 digit value on the timer the rest of the design was fairly simple. There was some difficulty with mounting our display to its enclosure. We chose to use an aluminum box because we had one lying around from a previous project but the segments were grounding to the aluminum and not functioning properly. Soldering the display onto a protoboard and then mounting that protoboard to the enclosure solved that problem. In the future we would use something made of plastic and 3D printed so it would be easier to mount components to.

The biggest lesson we learned was time management. Coordinating as a team was difficult due to our conflicting schedules and we frequently did things asynchronously. Utilizing applications like discord helped us keep in contact with each other and communicate while working independently.

The other lesson we learned was documentation. This class required far more documentation than other engineering classes we have taken and documenting work became more important as we were working independently. It became very important to document our progress so we both knew what was going on.