

EECS 2032 Alarm Clock Project

By: Jonah Ottini
ID#: 218945030

Section 1: Overview

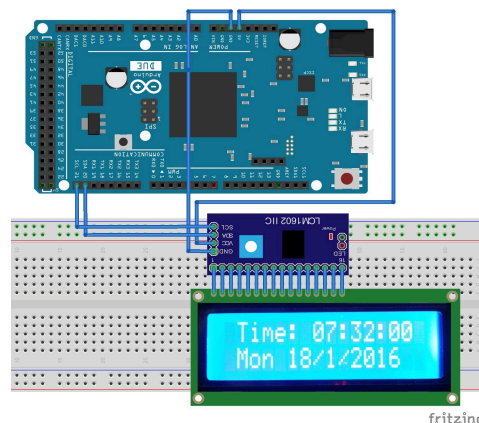
For my final project in this course, I made an alarm clock using the arduino due bored. I decided that for this project it would be easier to work by myself, so I am the only member of my group. Working by myself allows greater control over the project as well as a more manageable chance of error, and I think it's realistic for the scope of this project.

The general goal of this project would be to create a fully functioning alarm clock, with any potential features an alarm clock would need. Features like a way to display the current time and date, an alarm that could go off at a specified time, and a way to permanently or temporarily stop the alarm are absolutely necessary for an alarm clock. Being the only member in my group would mean I would have to implement all these features on my own.

Section 2: Features

Display

Creating a working clock display would be the foundation of this project, so clearly it was the first thing that had to be done. Luckily while researching ideas for a project I stumbled upon a forum called Electromania that detailed how to create a working clock display using the Arduino Due's built in Real Time Clock (RTC) functionality. Using that as a base I could hook up an LCD that met the Due's requirement and begin work on the other features of my alarm clock.



<https://www.theelectromania.com/2016/01/clock-and-calendar-using-arduino-due.html>

Alarm

The next most important thing for an alarm clock would be the alarm. Luckily the code base that I was using gave me an idea. When displaying the current time the program stored each time element in a separate variable. This means that hours, minutes and seconds each had their own unique variable that was constantly changing to reflect the current time. Therefore I could create a parallel set of variables that reflected the time I wanted the alarm to go off. Then if the two times ever lined up I could make it start playing the alarm. When that happens I can set a boolean value to true to allow the alarm loop to run until the stop button is pressed.

Stop and Snooze Buttons

For the buttons I had to do a little research to understand how to hook them up, but once I got them working it was pretty simple. I could create two if statements in the alarm loop that would check if either button was being pushed while the alarm ran. If either were the alarm would stop, and if the button pushed was the snooze button a set amount of time would be added to the alarm time value. If neither of the buttons were being pushed the alarm would play as normal.

Section 3: Conclusions

The content from this course greatly helped me in this project. The labs early on in the course helped me grasp how to use the board along with components and a breadboard to get things to run. The later lectures about coding in C helped me to actually write my programs to make my alarm clock function as I wanted it to.

Many problems had to be solved throughout this project. Many of the problems actually had to do with the base code that I was using. Firstly the RTC library the code used was outdated so replacing was one of the first things I did. Other issues included the date display being accidentally reliant on the minutes variable which caused it to print incorrectly, and the array of day names starting with Monday when 0 in the RTC related to Sunday. Another problem I had was that my original buzzer wasn't loud enough so I had to buy a new one before I was satisfied with the sound.

This project taught me a lot about working with both hardware and software to create embedded systems. Using code to manipulate physical components is very interesting and this course gave me a stronger understanding of how all the embedded systems people use day to day function.

I think that if I put a little more effort into the presentation of the project, and maybe added a few more features, it could result in a marketable product. Although modern day alarm clocks are already able to fit into smartphones as a small feature so I doubt the demand would be high.

In my opinion this project was very successful in reaching its goals. My original idea was very straight forward but as I kept working on it, I kept thinking of new cool features I wanted to implement into it. Things such as multiple alarm tones that can be swapped between, and the ability to set specific alarms based on the date. I have already been using it as my personal alarm clock for a while now and it works great. With the original intended features working perfectly, and the addition of extra things, I think this project was a massive success.