**Team Name:**

The C-Team

**Team Members:**

Dow Cox, Cole Wilson, Austin DuCrest, Nolan Magee, Bernie Friesel

**Class Number and Name:**

ECE 3130-001 Microcomputer Systems

**Submission Date:**

4-30-2025

**Table of Contents:**

Title page 1

Table of Contents 2

Introduction 3

Specifications and Description 4

Detailed Implementation

Interface Design

Microcontroller Resource Utilization

Software

Hardware

Analysis

Testing

Public Safety

Global/Economic Factors

Teamwork Experience

Collaboration and Inclusivity

Leadership, Goals, and Planning

Appendix

**Introduction**

We will be implementing a soundboard, i.e. a synthesizer. This will be accomplished by using the on-board buzzer.

We will be assigning the buttons of the keypad and the switch buttons to an associated frequency. Each frequency has an associated musical note it makes the sound of, and we can use those sounds to make songs.

The user can play octaves 1 through 6 for musical notes C, D, E, F, G, A, B. The user switches through the octaves by using the pushbutton SW2.

**Specifications and Description**

Block Diagram of what the board is doing:

A screen shot of a graph

AI-generated content may be incorrect.

Whenever we press a certain input button using a pushbutton, it tells the program to deliver a certain frequency to the buzzer, which is accomplished by converting the frequency to period and inputting that value to the code controlling the buzzer.

We got the general values for the frequencies from an online chart and an audio engineering textbook, and we adjusted these frequencies as needed to get the right sounds from our buzzer.

Excel Spreadsheet of specifically what musical note each input button generates the Buzzer Sound Outputs:

A screenshot of a calendar

AI-generated content may be incorrect.

Note: these are color coded to the LED colors we used for the hardware

Note: an octave is 8 notes, the underlined notes indicate they are used for another octave

These interface directly with frequency matrix:

A black background with numbers

AI-generated content may be incorrect.

**Detailed Implementation**

**Interface Design**

The LCD provides a human machine interface (HMI) to tell the user what the keypad and switches do.

The keypad is a bunch of pushbuttons, and these are used by the user to generate musical notes/sounds as if they were playing an instrument.

The switches are used to change the boards operating modes.

The external LEDs implemented on the breadboard provide visual feedback to the user on what octave is being played.

The external buzzers implemented on the breadboard allow the user to play

**Microcontroller Resource Utilization**

We ran into some polling issues while programming the buzzer. To fix this, we utilized the microcontroller’s pulse width modulation (PWM). The PWM allows us to asynchronously switch the buzzers output sounds without having to use delays. The PWM is essentially one. Delays were not preferable since we our buzzer and music relied on exact timings.

So for example, in our buzzer.c:

A computer screen with text

AI-generated content may be incorrect.

**Software**

We implemented the code using Keil uVision and VS Code coding environments and programmed primarily in C-language. We used Keil whenever downloading program to the board. We utilized GitHub to share our main program and other files online with each other. The code was quite lengthy with lots of initializations for the keypad and such, and you can find the code in our GitHub:

GitHub: <https://github.com/dwc42/ece3130_project>

README: Dow explain your complicated code

**Hardware**

We used a STM32-NUCLEO-L476RG microcontroller connected to an EDUBASE-L452 1173-H2. This board was provided to us for purposes of the class/lab.

We utilized 6 external LEDs on the breadboard. The specific pin connections we utilized are detailed here:

A red and blue rectangle with black text

AI-generated content may be incorrect.

Note: these octaves match those in the specifications and description (p. 4).

We utilized 4 external buzzers on the breadboard. The specific pin connections are as follows:

A white grid with black lines

AI-generated content may be incorrect.

Full view of board:

README: add pictures of board with hardware working

**Analysis**

**Testing**

We tested the functionality of the input buttons (keypad) by pressing each button and listening for the sound the buzzer made. This was fairly easy to test, since if we hit button and it made a noise that was a good thing.

The actual tones / musical notes being played were tested by our musical experts Bernie and Austin to know if the buzzer was making the right noise and if we needed to adjust anything for a better output. The adjustments were rather simple as we would just adjust the frequencies in our code.

We ran into an error whenever we pressed multiple buttons at once because the buzzer got confused. We fixed this by adding a few functions to know exactly

**Public Safety**

This system utilizes a low voltage board without much physical danger to the user in terms of being electrocuted or shocked.

However, frequencies under 20 Hz and above 20 kHz are beyond the human range of hearing and can be dangerous to listen to. Accordingly, we avoided inputting any frequencies that exceed the human range of hearing. For example, the musical note C0 has a frequency of 16.35 Hz so we avoided programming that note.

It should also be noted that this board contains solder, which often contains lead which is carcinogenic. Anytime you handle this board you should wash your hands afterward.

**Global/Cultural/Economic Factors**

Music remains one of the great innovations of mankind, and every culture around the world has some form of music. Music has become synonymous with Tennessee through Nashville and great country music stars, and people have made great economic profit in the music industry.

This is made possible by innovative technologies like electric guitars and keyboards. Our board is a relatively simple implementation of a soundboard but serves as a good demonstration of what can be accomplished by engineers in a short amount of time.

The educational board we used cost about $125 for educational use, and the microcontroller we used cost about $25. The external LEDs and buzzers we implemented only cost about $10 each for packs of these, so the overall cost was about $170.

This overall cost is relatively inexpensive, and you may be able to implement it cheaper if you used only the necessary components (buttons, buzzer, LCD, microcontroller). But for our purposes the trainer board and microcontroller were provided for free, so they were the best options for us.

**Teamwork Experience**

**Collaboration and Inclusivity**

We maintained a co-learning environment where if one of us did not understand something the others would explain why or how we needed to do something. This allowed us to work efficiently with each other while everyone still learned something.

Dow has the most experience with coding and he took the lead on the project with writing most of the software, and took on a managerial role with the other team to help them implement some of the other features.

Austin and Bernie have the most music experience. Austin knows how to play the piano and Bernie is an audio engineer, so they contributed a lot to knowing the different music notes and how to accomplish the makings of the songs.

Nolan worked primarily on implementing external LEDs on the breadboard to give the user feedback on what octave is currently being played. He implemented the necessary resistors to keep the LEDs from burning and interfaced the LEDs with the associated code.

Cole primarily worked on writing the project report to communicate the work done by the team. Cole made the structure for the report and made most of the rough draft with Nolan editing the report and implementing a PowerPoint presentation.

**Leadership, Goals, and Planning**

Our team shares many classes together, so we brainstormed ideas among ourselves for about a week until having a meeting to finalize our plan and get an idea of where we needed to start and what we wanted to work on.

We implemented weekly group meetings that lasted about an hour to keep the project on track and clear up any confusion we had from working asynchronously.

Dow quickly took the lead while working on the project. As group figured out what we wanted to accomplish, Dow often knew the best way to implement that. He used a hands-on leadership style and directed the other group members on what we needed to accomplish for specific tasks.

We knew the buzzer and the buttons would be the most important parts of the project, so figuring out how to use those was the main priority. We made small goals to build momentum like just making the buzzer make noise or making the pushbuttons blink an LED. Then once we accomplished that we moved on to new goals like pressing multiple buttons at once and adding a button to adjust the frequencies being played.

**Appendix**

Musical Notes and their frequencies:

[1] “Note Frequency Chart (Pitch to Note),” *muted.io*. https://muted.io/note-frequencies/

Cost Estimate for Microcontroller:

[2] “Amazon.com: STM32 Nucleo-64 Development Board with STM32L476RG MCU NUCLEO-L476RG: Electronics,” *Amazon.com*, 2025. https://www.amazon.com/STM32-Nucleo-64-Development-STM32L476RG-NUCLEO-L476RG/dp/B01IO3N646 (accessed Apr. 07, 2025).

Cost Estimate for Board:

[3] “EduBase-V2,” *Trainer4edu.com*, 2025. https://trainer4edu.com/edubase\_v2/index.html (accessed Apr. 07, 2025).