

# Audio Compressor By Good Vibrations Sponsor: SDSU

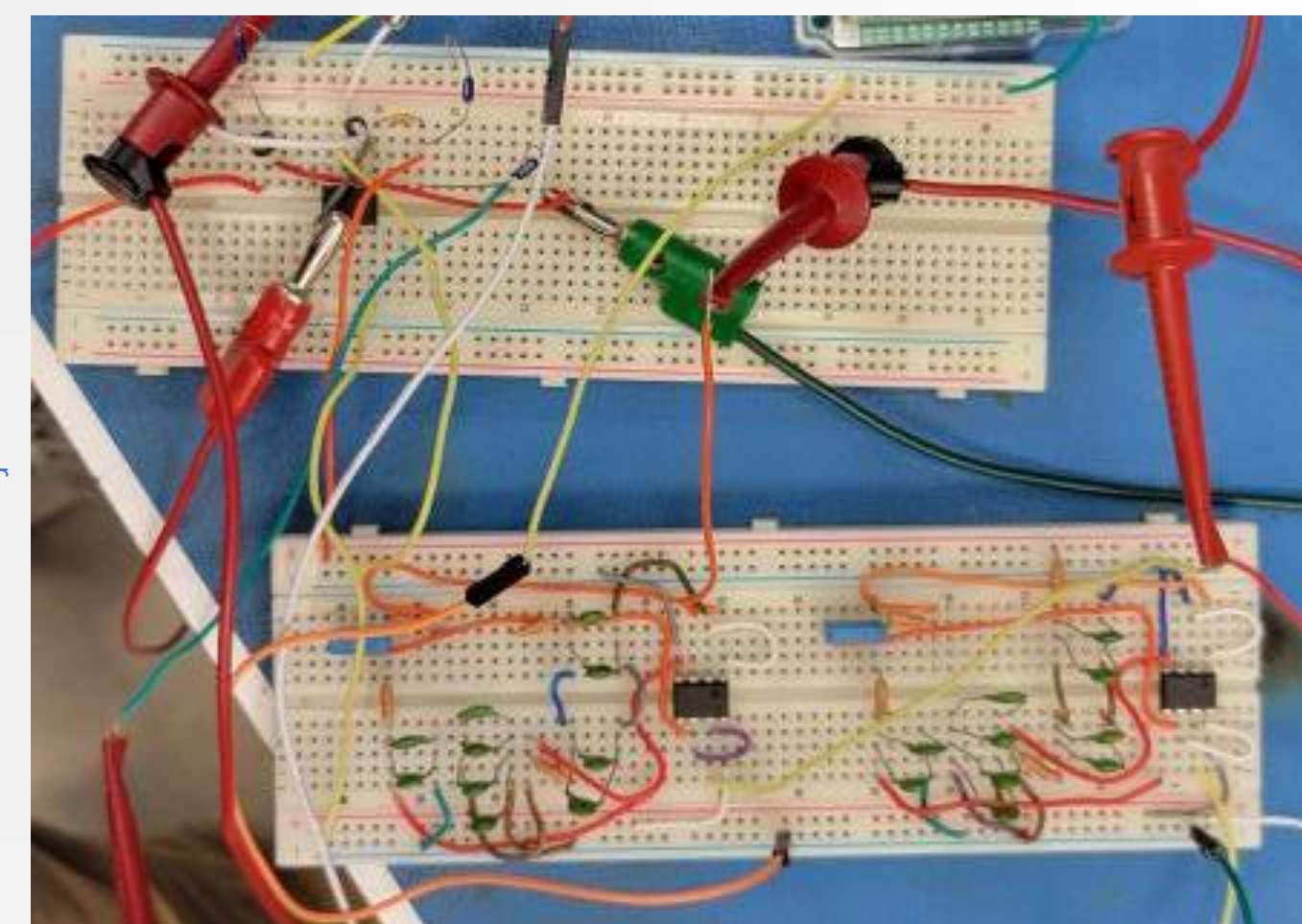


SAN DIEGO STATE  
UNIVERSITY

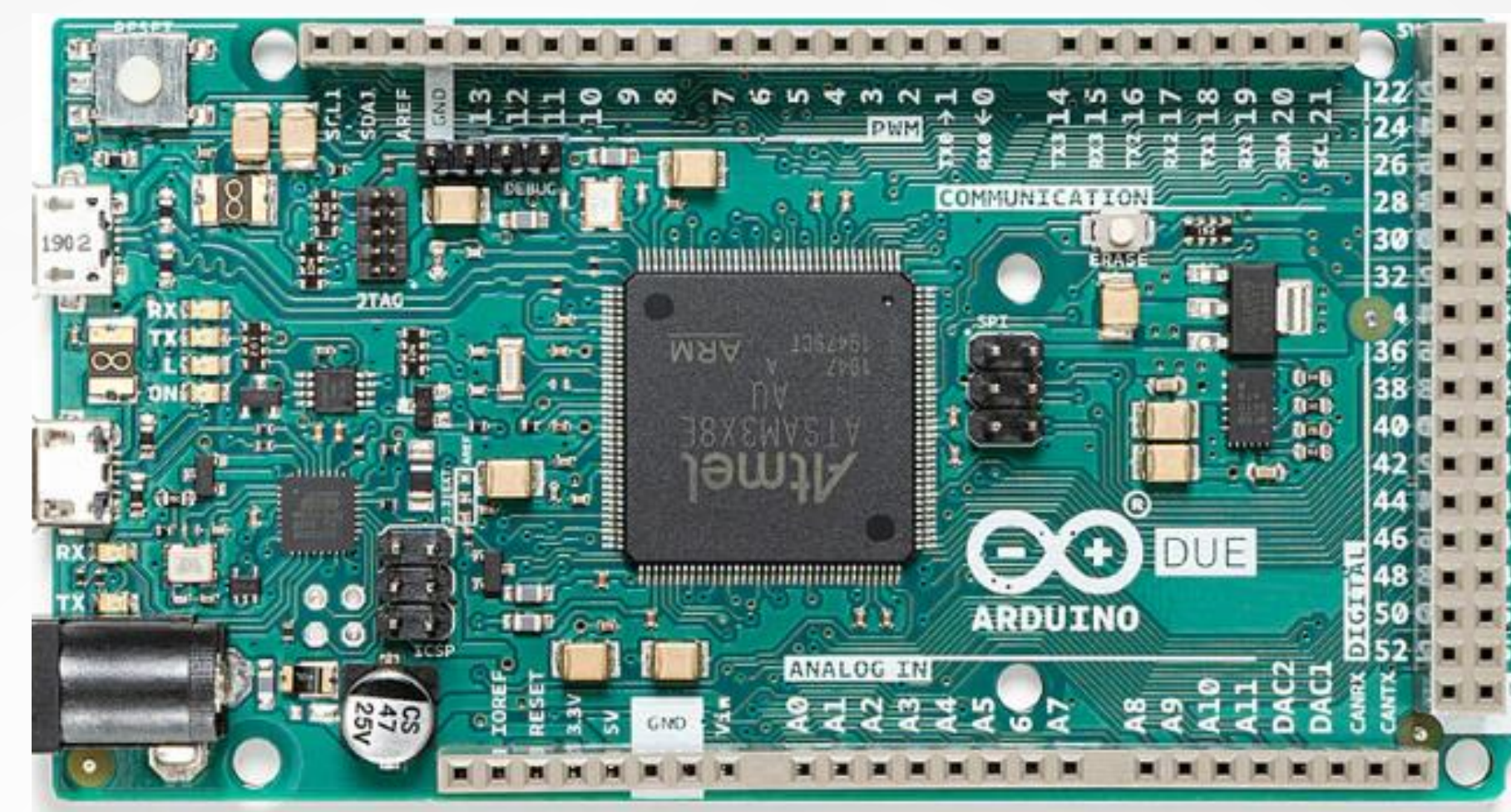
Electrical and Computer Engineering  
College of Engineering

## Problem Statement

There are 52.4 million Americans 65 and older representing about 16% of the population. Approximately one of three people between the ages of 64 and 74 have hearing loss and half of those older than 75 have difficulty hearing. For these people, enjoying music can be frustrating or impossible because they can't hear the soft parts, and the loud parts cause pain for them (and those around them!). For this project, the team will integrate software provided by Professor Dorr on a hardware platform that includes a multifunction switch, display, enclosure, connectors, and power supply. The audio compressor will come with multiple selectable profiles to accommodate for multiple different users.

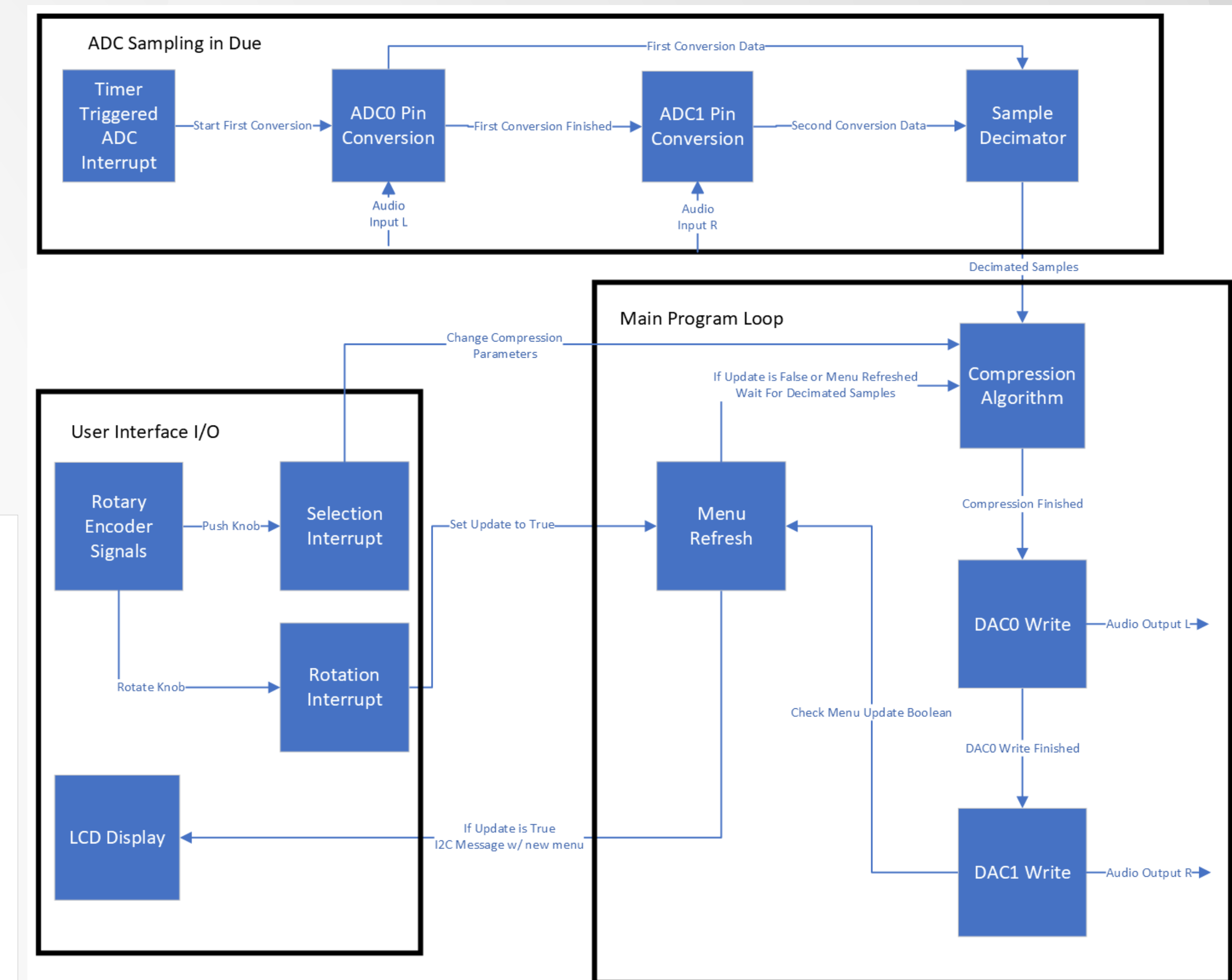


## Key Components



From Left to Right, Top to bottom: LCD Display, Anti-Aliasing, and Reconstruction Filter.  
LCD Display, displays the GUI menu which includes profiles to choose from.  
Arduino Due, used to run the main audio compression code and output the compressed audio. The Due also interfaces with the LCD and other peripherals to dynamically change compression profiles  
Anti-Aliasing filter, to prevent high frequency noise at the input and the Reconstruction Filter is used to filter out high frequency noise from the Arduino Due.

## Software Flow Diagram

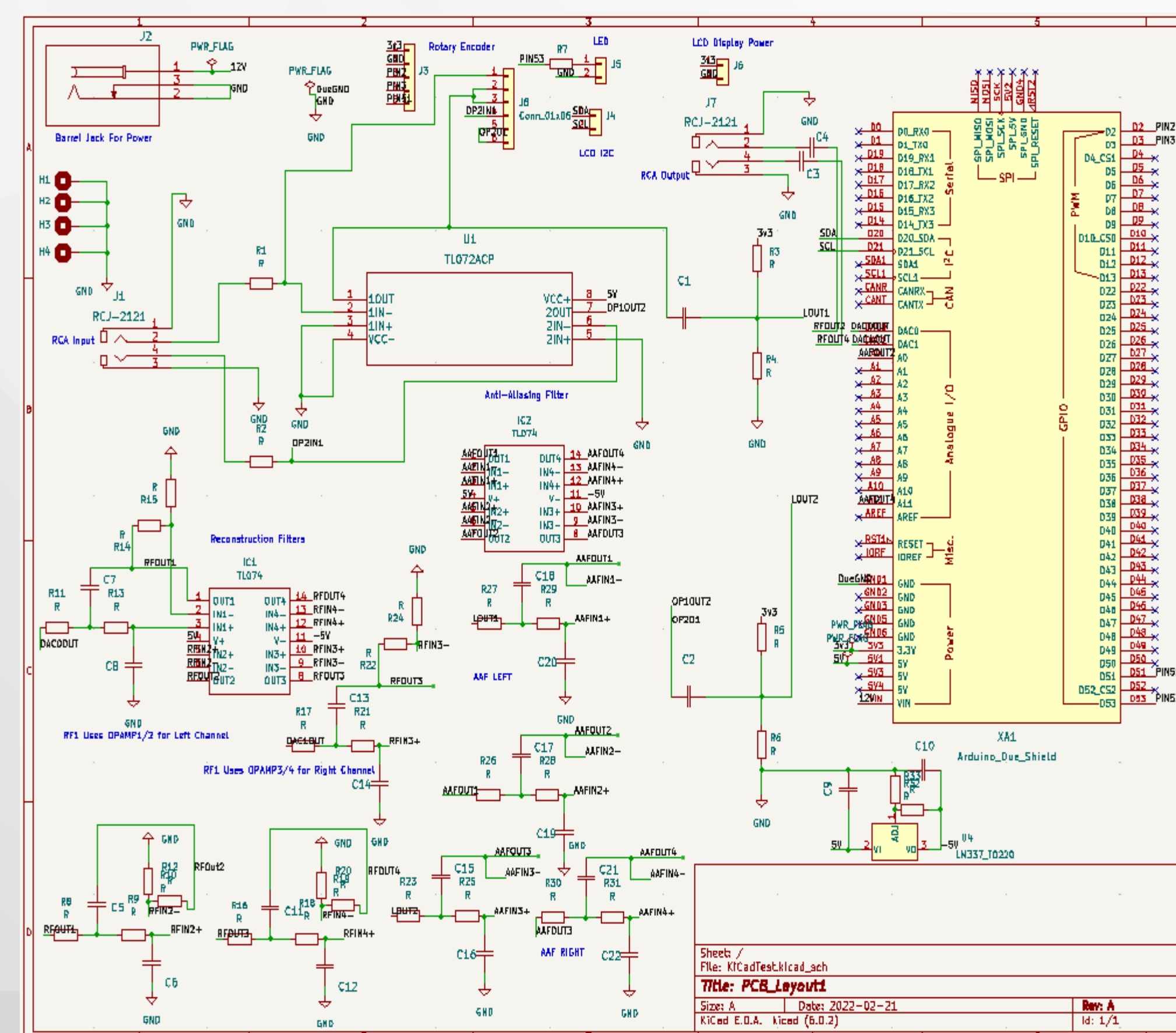


## The ECE Team



Team Members: Ethan Cua(COMPE), Hugo Rivera Garza(EE), Alex Belmar(EE), Jonah Andres(COMPE), Ninos Younamdam(COMPE)  
(Left to Right)

## PCB SCHEMATIC



## ENCLOSURE



The TinkerCAD model of the enclosure shows the user interface available to the user. The rotary encoder and LCD are used to display the current selection of compression profiles, as well as allow the user to select a new one while the device is in operation. The knob on the front is the input gain adjuster, to control the input volume, which includes a clipping LED to notify the user when they have adjusted their audio to the upper allowable limits.

The software was divided into three logical groups:  
The audio sampling code is triggered by a timer interrupt which triggers new ADC conversions, saving the values of previous conversions to be output after a delay, with gain.  
the user interface triggered by signals generated from the multifunction knob, which tells the device which compression configuration is in use, dictating how severe the compression is, and the volume range that should remain uncompressed.  
and the main body of the compression code that synthesizes the results of both to determine the gain that should be applied to the input audio signal, and use that gain to generate DAC output

## Acknowledgements

Professor Dorr  
SDSU College of Engineering

Spring 2022