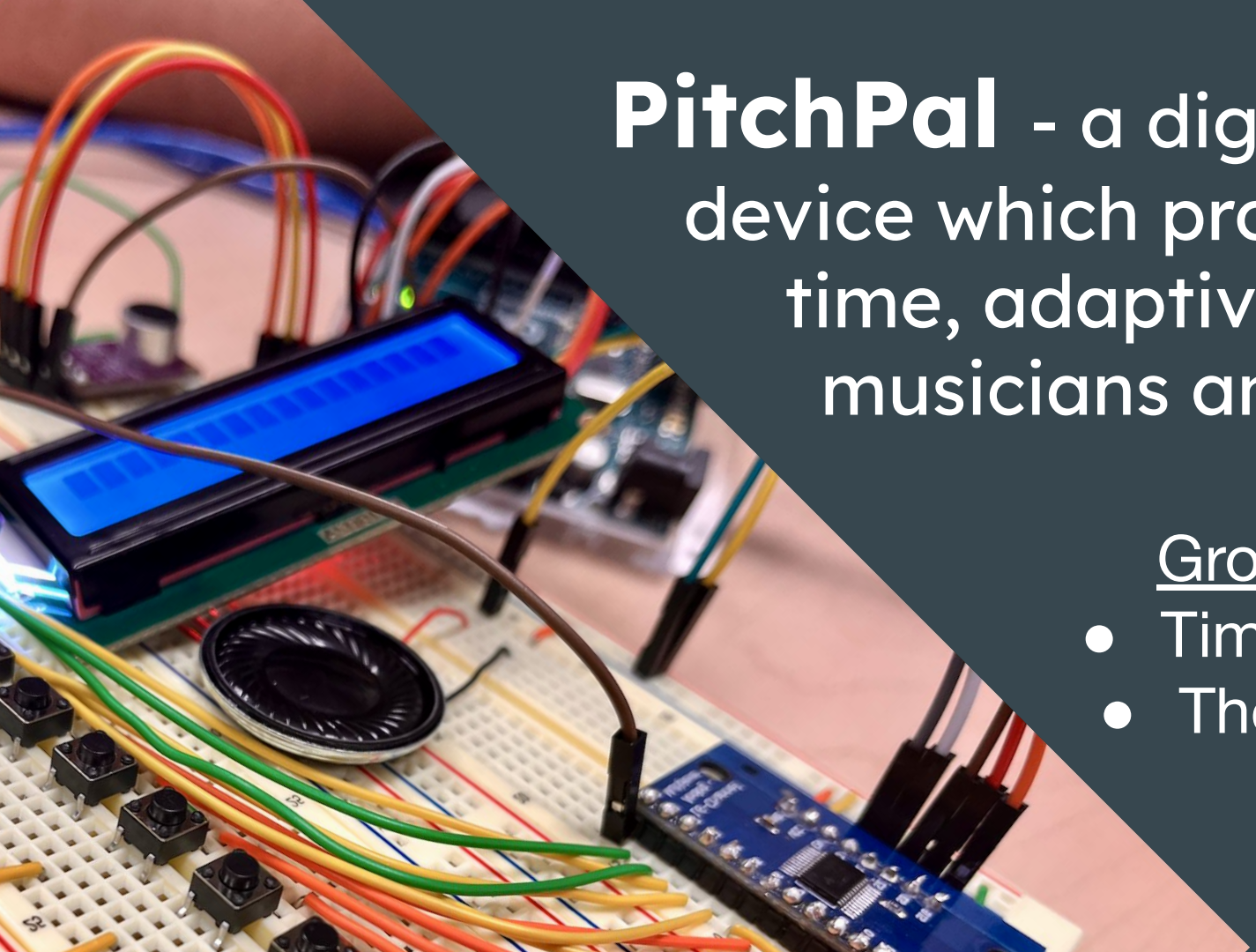


PitchPal - a digital tuning device which provides real time, adaptive feed, for musicians and vocalist

Group members

- Timothy Scholtz
- Thomas Mahok

Project Features

- **Pitch Feedback:**
  - PitchPal is able to give real time feedback about a person pitch accuracy
- **Note Identification:**
  - PitchPal is able to listen real world audio, and identify what pitch is being played
- **Audio feedback:**
  - PitchPal is able to play notes to give a user a refrence point for the practice

Problem Statement:

To develop a user friendly device to assist musicians at all skill levels by providing a reference pitch and feedback on their accuracy, addressing the challenges of traditional ear tuning, especially in the context of internet-based music recording where precise pitch alignment is crucial.

Phase 1

Project Milestones and Decisions

- Chose problem statement
- Came up with a draft of the physical device
- Created flowchart of device use case

State of the project

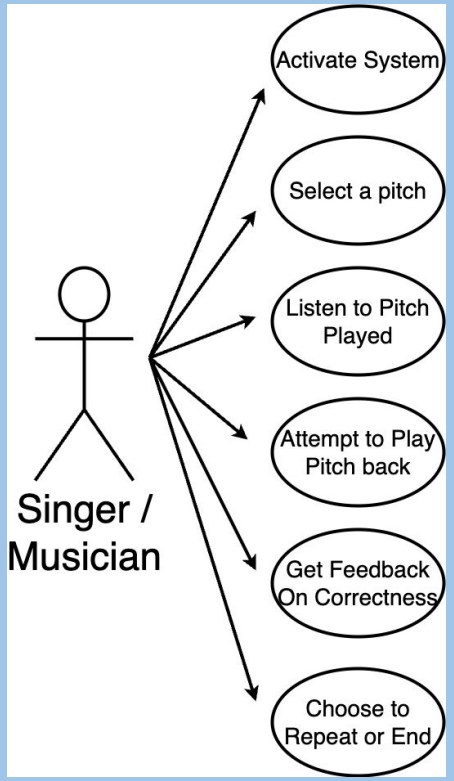
We gathered all the components required to build our project, as well as starting testing out arduino programming

Problems

We needed to find a way to get 12 buttons for input, as well as learn how to process audio

Changes to original desig n

We reduced our project scope, which originally had envision using a led based system for providing dynamic and colourful feedback



Phase 2

Project Milestones and Decisions

- Achieved the basics requirements, was able to: get user input, record and analyses audio, and give feedback

State of the project

We got all the individuals tasks working, which included: user input, user output, recording audio, processing audio, and playing audio

Problems

We struggled to get a systems which contained all of the parts and had a usable user interface working

Phase 3

Project Milestones and Decisions

- Created a working demo of device which was able to complete all original goals

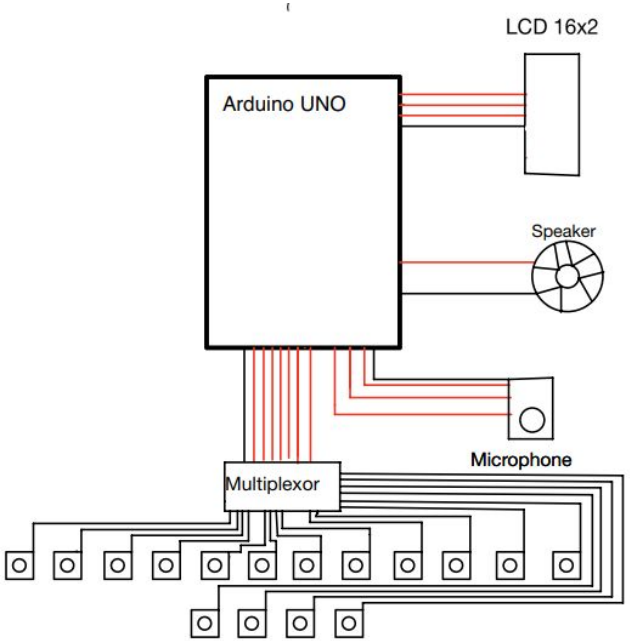
State of the project

We successfully implemented all features of the project. This was all integrated with a user friendly menu system.

Problems

We struggled to have some of the hardware and software act correctly in tandem. Our biggest issue being crashes due to memory overflow when trying to calculate the dominant frequencies the user would input.

Project Schematic



Technologies used in Project:

- Microphone
- Speaker
- Multiplexer
- LCD Screen
- Buttons
- Arduino Uno

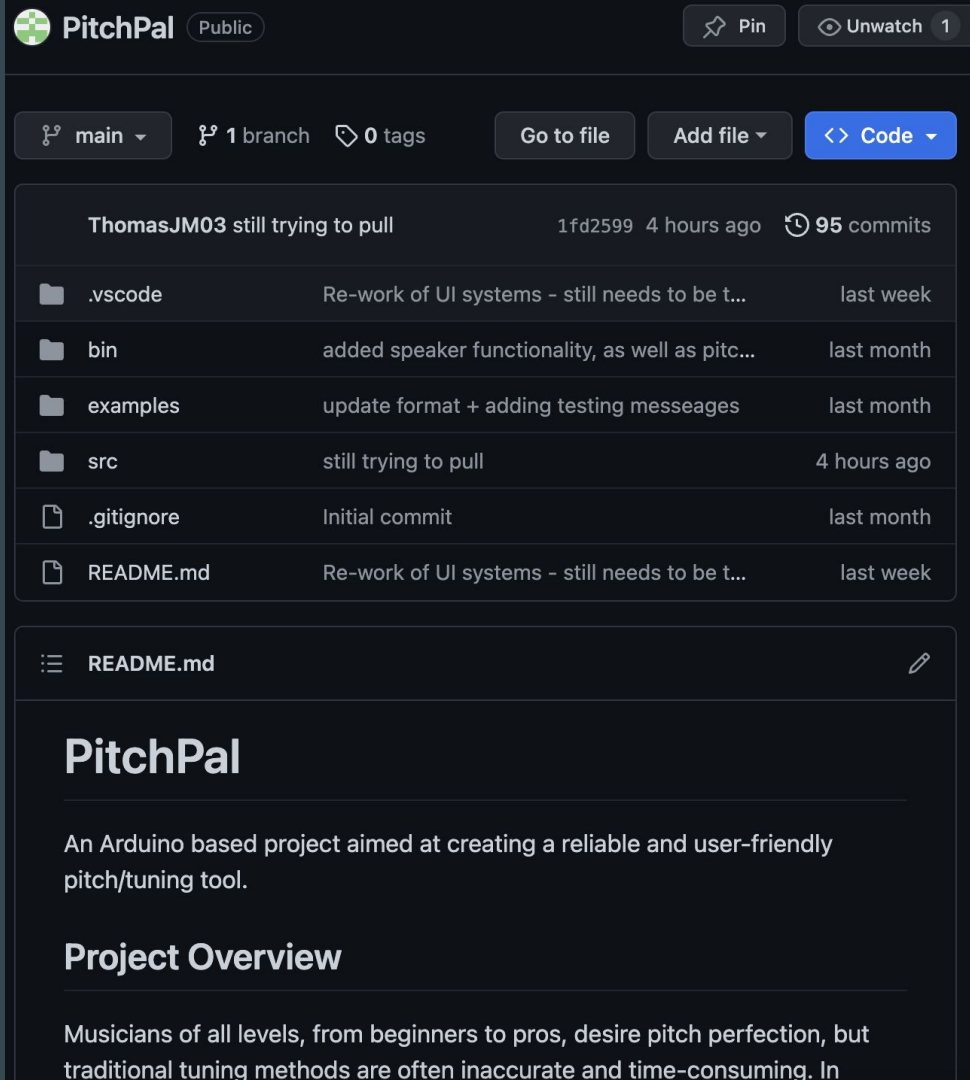
The Codebase

We followed declarative and state design principles when creating out project.

Our project consists of 3 main file:
Main.cpp
Input.cpp
Output.cpp

With relevant header files, as well as header files to store constant values.

Location of code base:
https://github.com/t-scholtz/PitchPal



The Codebase

Main.cpp

The purpose of Main is to **initialize** the device when it starts, and it also runs a **state machine**.

The state machine is built from an infinite **loop** and **switch case**.

Each state is a seperate function, and to transition between states, a function returns the number of the state it wishes to transfer control to.

```
/* Main Loop for Project
   State machine
   case 1: Main Menu
   case 2: Pitch Practice
   csee 3: Play Note
   case 4: Indentify Pitch
   Defualt - Go to state 1
 */
int state = 1; // starting values
void loop()
{
  switch (state)
  {
  case 1: // Menu Page
    reset();
    state = stateSelector();
    break;
  case 2: //User choses note and gets feedback
    state = pitchPractice();
    break;
  case 3: //User chooses a note and speaker plays it
    state = notePlaying();
    break;
  case 4: //Listens to user audio and determines
          //what note is being played
    state = pitchFind();
    break;
  default:
    state = 1;
    break;
  }
}
```

The Codebase

Input.cpp

Input handles collecting and processing input collected by the arduino

Notable functions are getMicFrequency, which handles audio processing, as well as the functions related to grabbing buttons input.

```
1   // input.h
2   #ifndef INPUT_H
3   #define INPUT_H
4
5   String noteToString(int note);
6   int noteArray(int a, int b);
7   String noteStrArray(int a, int b);
8   int muxChannel(int a, int b);
9   float getMuxInput(int channel);
10  void micSetup();
11  double getMicFrequency();
12  int checkForButtonPress();
13  int waitForUserInput();
14  int waitScrollingText(String text);
15  int selectNote();
16  int selectOctave();
17  String noteFinder(double freqOfNote);
18  #endif
```

The Codebase

Output.cpp

Output handles formatting and outputting data.

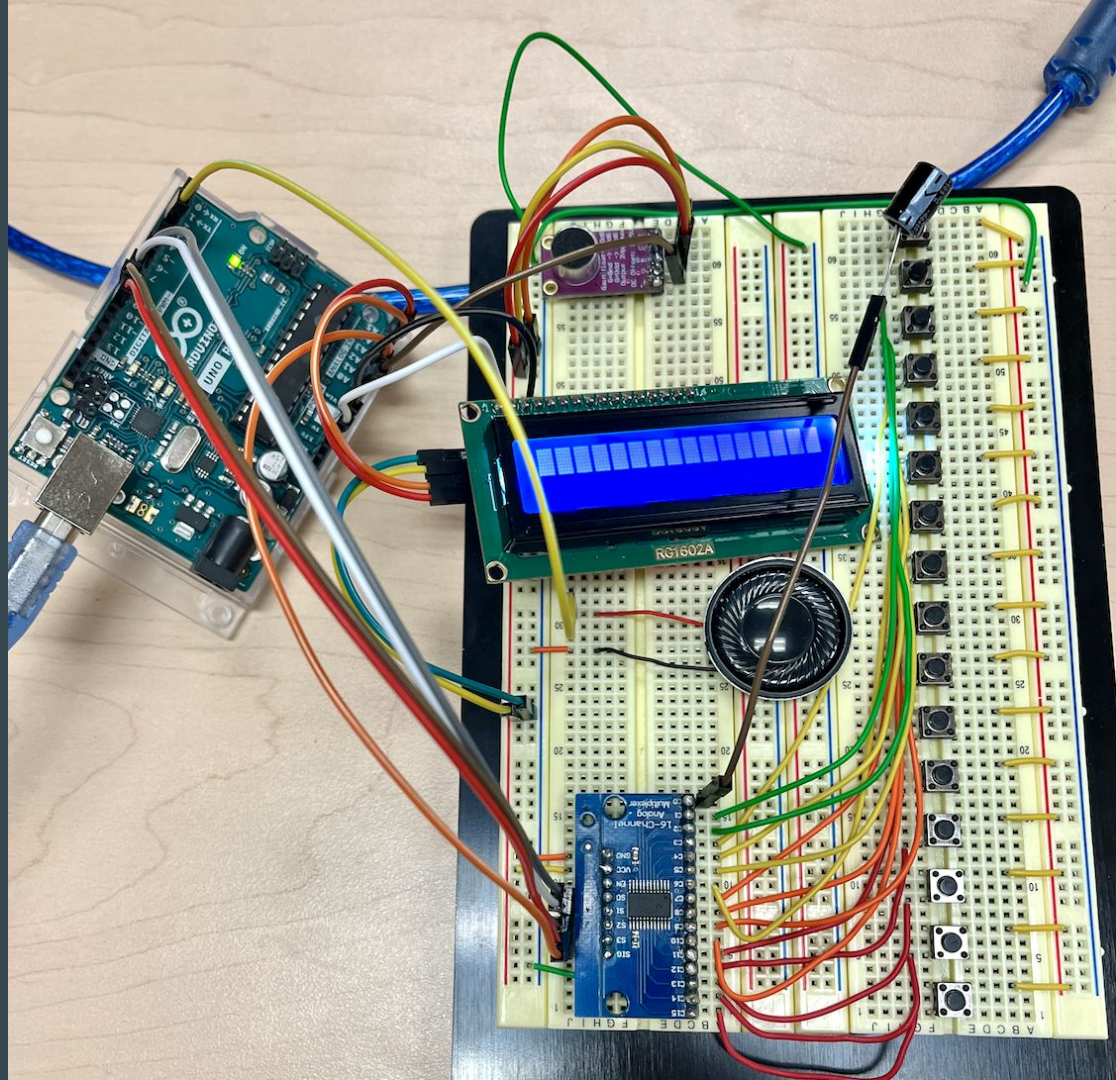
This includes speaker and LCD screen handling.

```
1   #ifndef OUTPUT_H
2   #define OUTPUT_H
3
4   void initializeLCD();
5   void lcdClear();
6   void lcdSetCur(int x, int y);
7   void lcdPrint(String A, String B);
8   void playNote(int note, int length);
9   void startUpAnim();
10
11  #endif
```

Project Demo

Project Demo

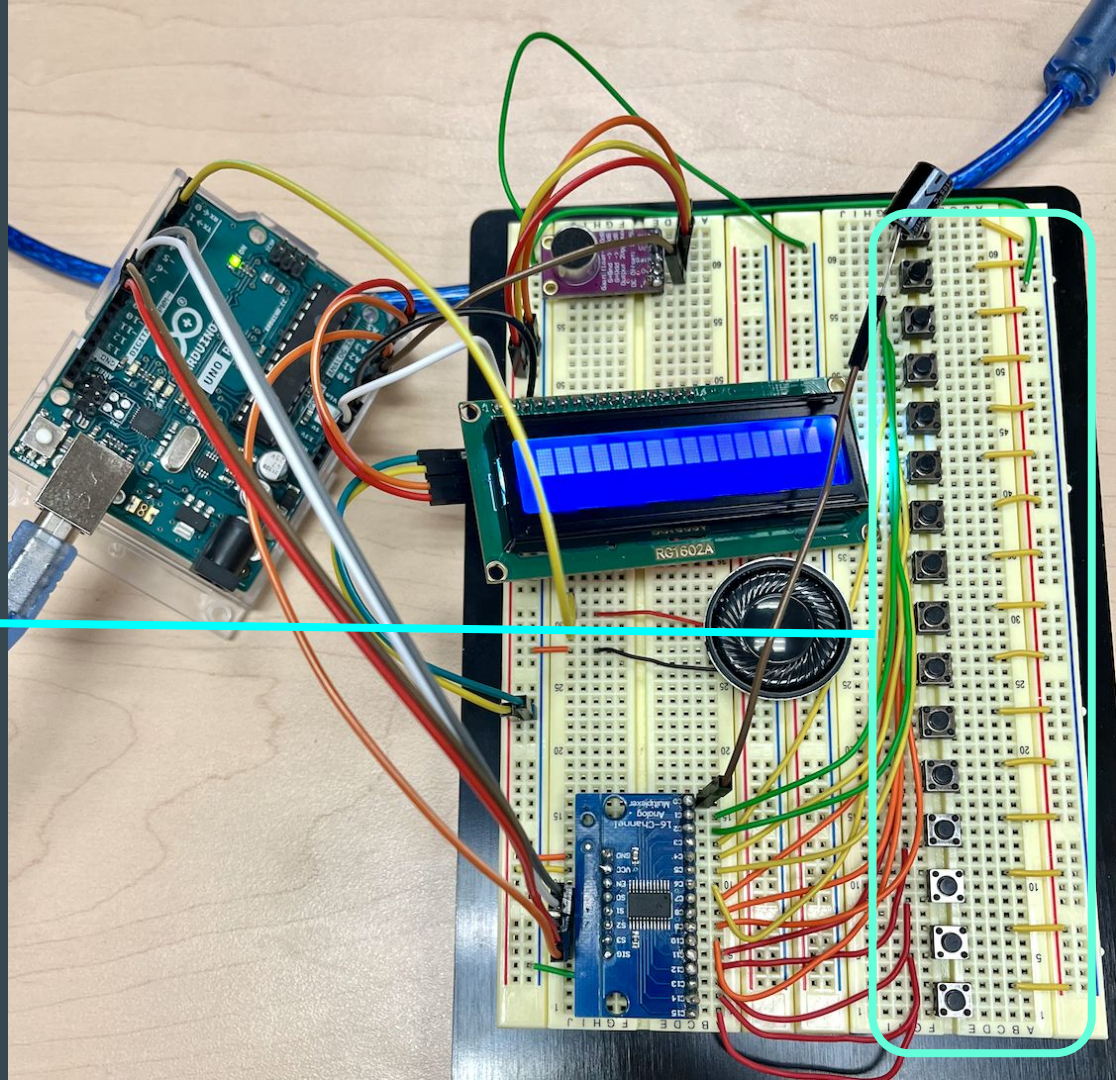
Project Breakdown



Project Demo

Project Breakdown

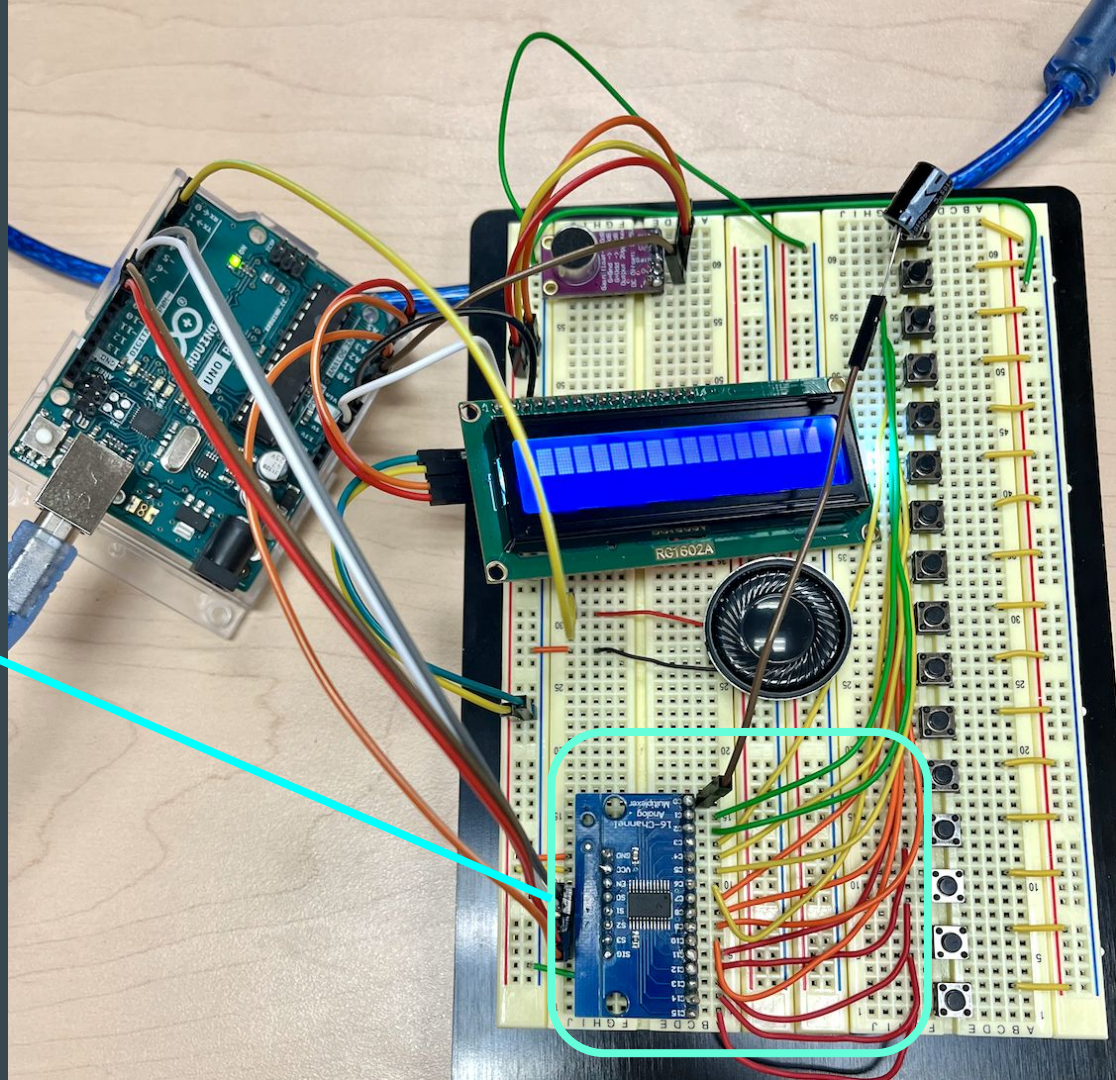
16 button input



Project Demo

Project Breakdown

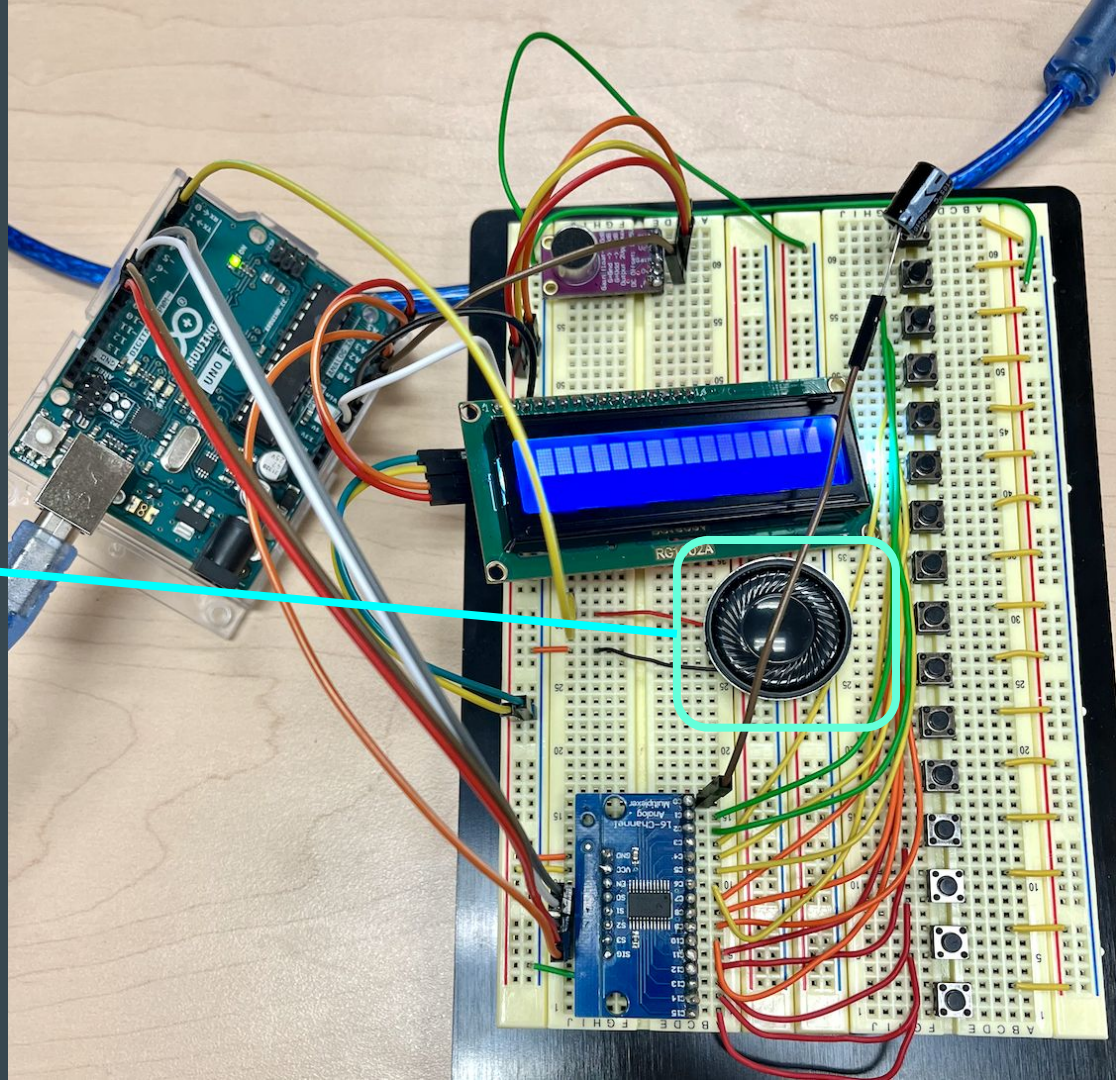
16 to 1 Mux



Project Demo

Project Breakdown

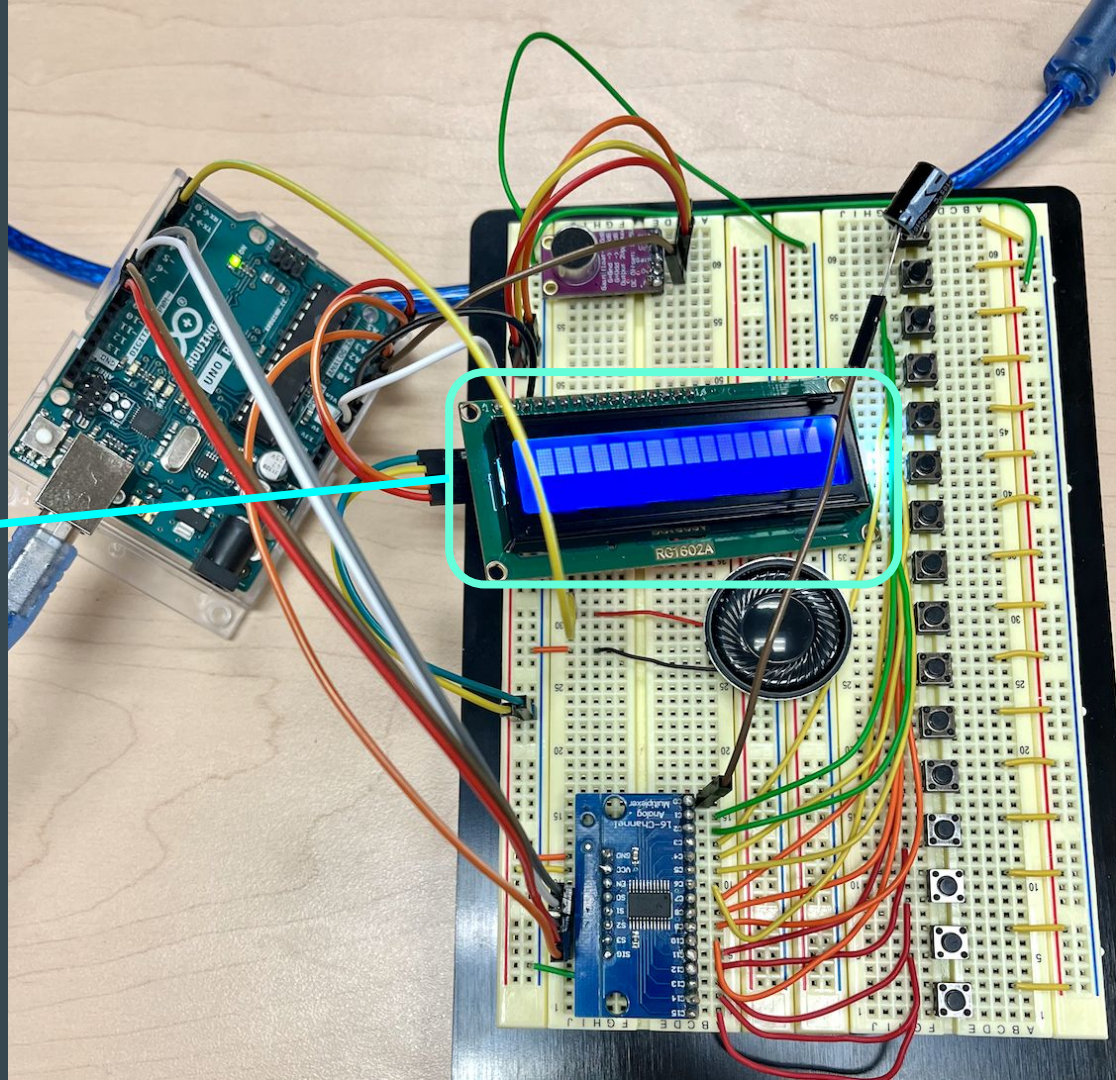
Speaker



Project Demo

Project Breakdown

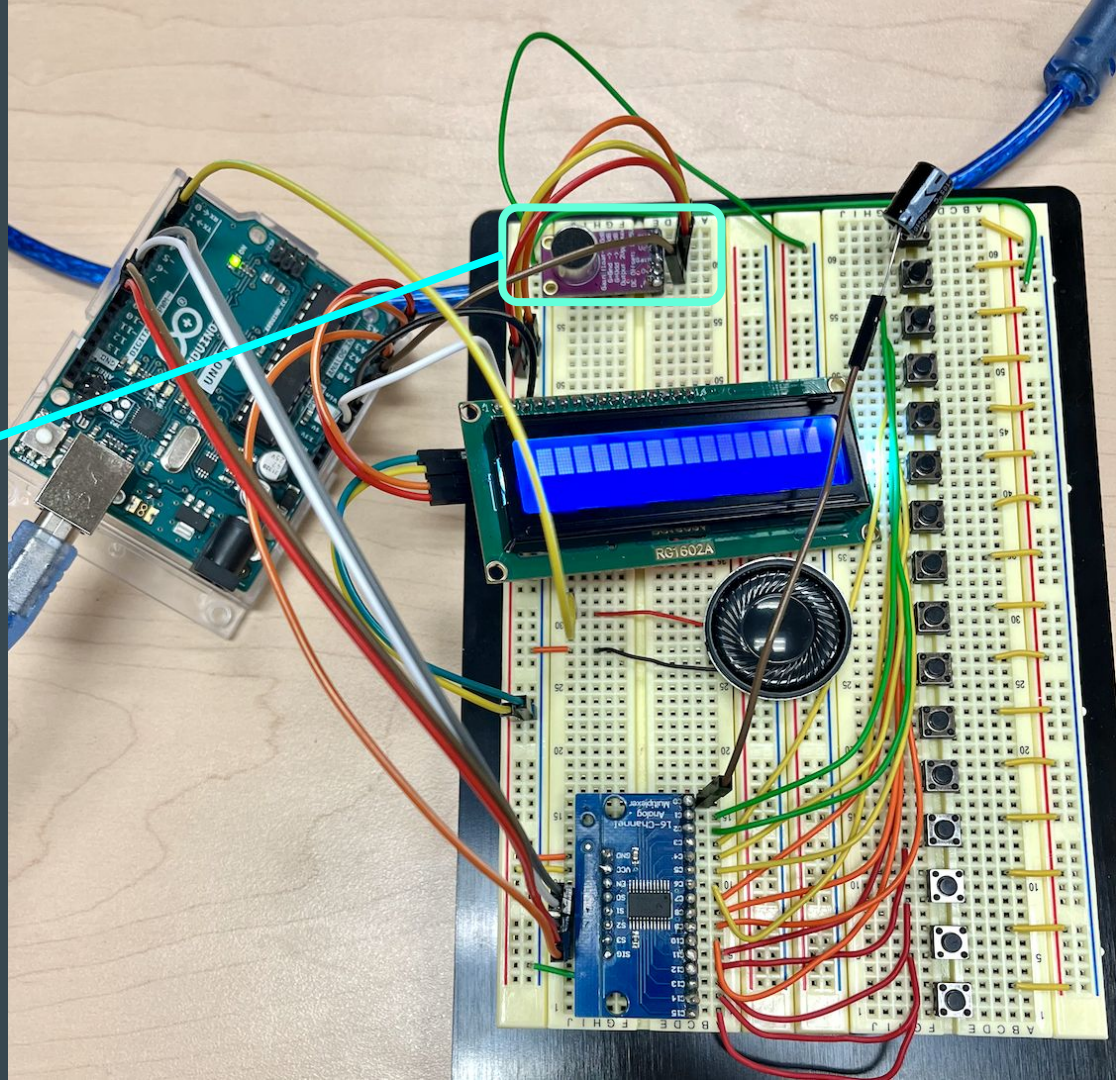
LCD Screen



Project Demo

Project Breakdown

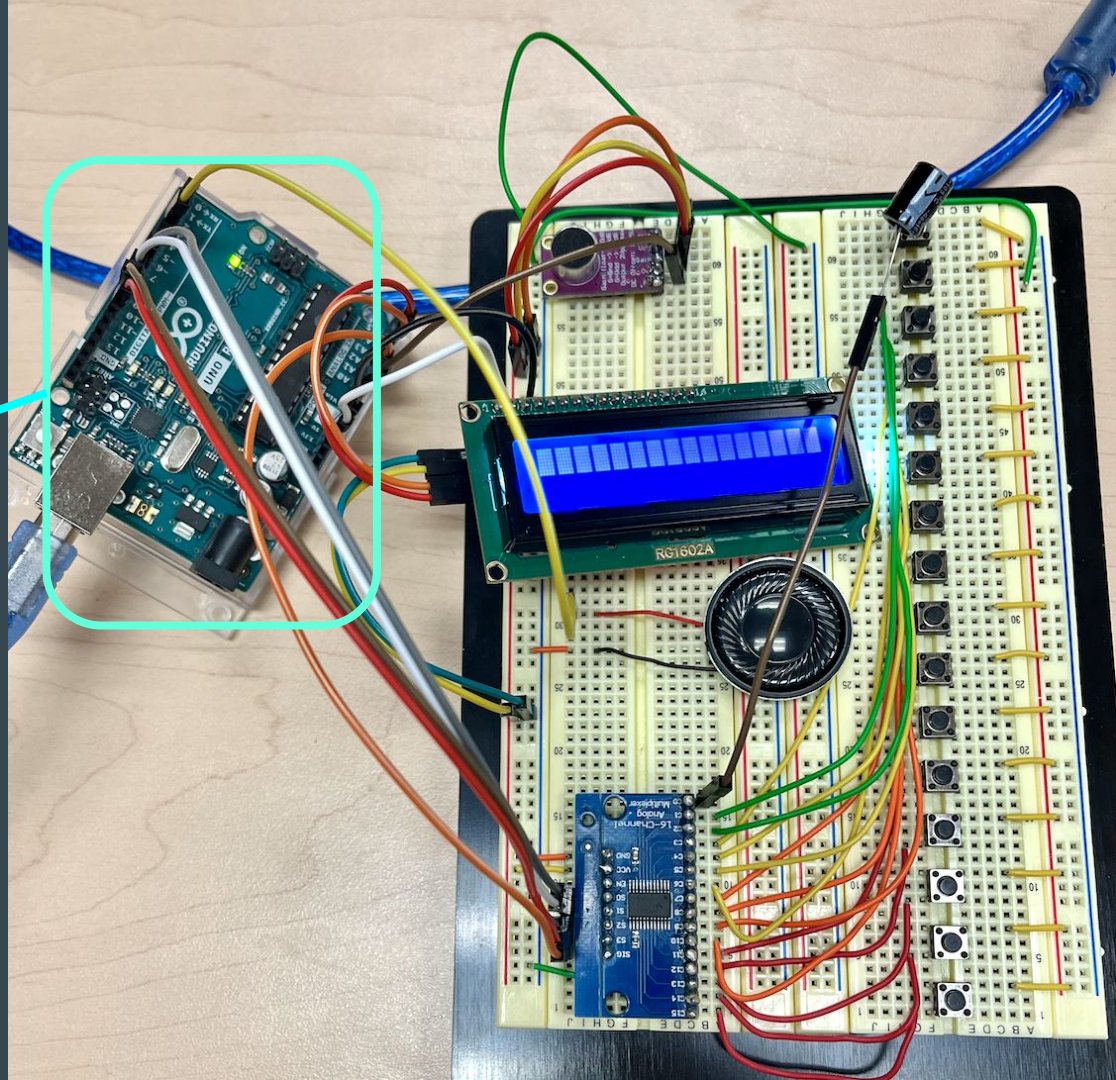
Microphone



Project Demo

Project Breakdown

Arduino Uno

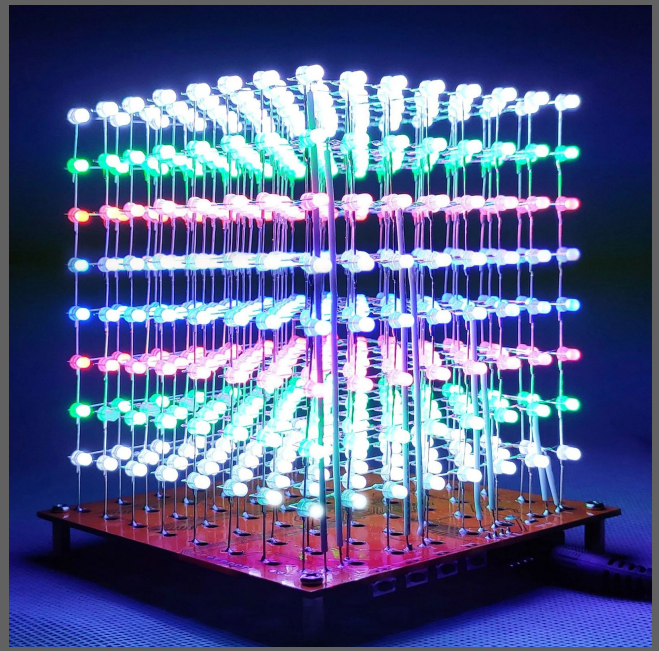


Challenges and Learning outcomes

Developing an interactive and visual user interface turned out to be more time-consuming than initially anticipated. Additionally, memory constraints posed several challenges, prompting us to optimize some of our more data-intensive calculations and handling. Debugging optimization issues consumed a significant portion of our time, overshadowing the actual coding process.

In the course of this project, we delved into audio processing and music theory to meet our objectives. Beyond technical skills, we discovered the value of friendship and teamwork, realizing their profound impact on our collaborative efforts.

Future work



We plan on developing an enclosure for the device, relocating it from the breadboard, and soldering it into a more compact design tailored to fit snugly into the enclosure. Additionally, we intend to incorporate an LED matrix to provide more engaging user feedback.