ECE 477 Midterm Review

Team 12 - R.A.C.H.E.L.

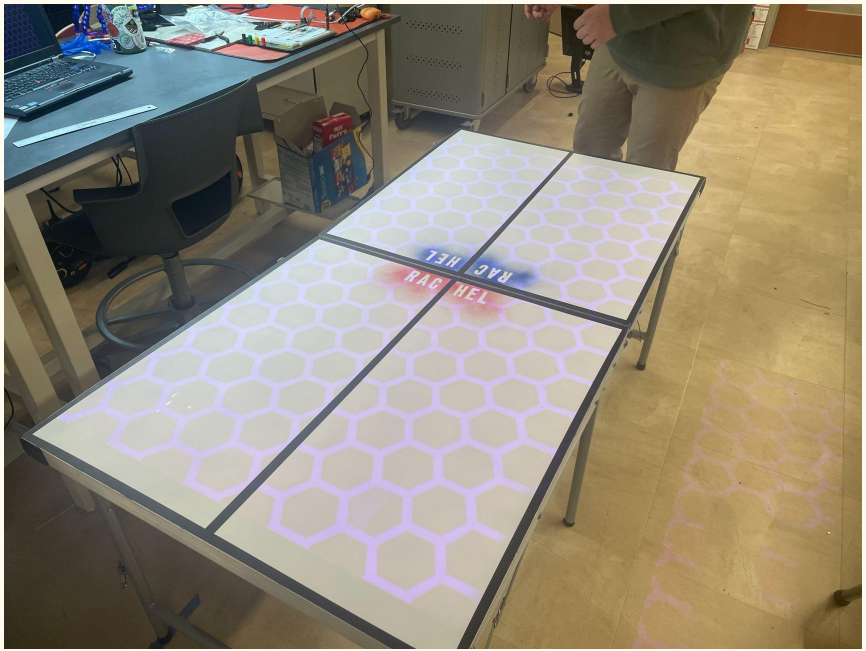
Bartosz Stoppel
Micah Morefield
James Hubbard
Jack Myers

Outline

- Project Overview
- Project-Specific Success Criteria
- Block Diagram
- Major Components
- Packaging Design
- Electrical Schematic
- PCB Layout
- Software Development Status
- Prototyping Progress
- Project Timeline
- Questions / Discussion

Project Overview

Project RACHEL is an enhancement to the original game of table tennis.



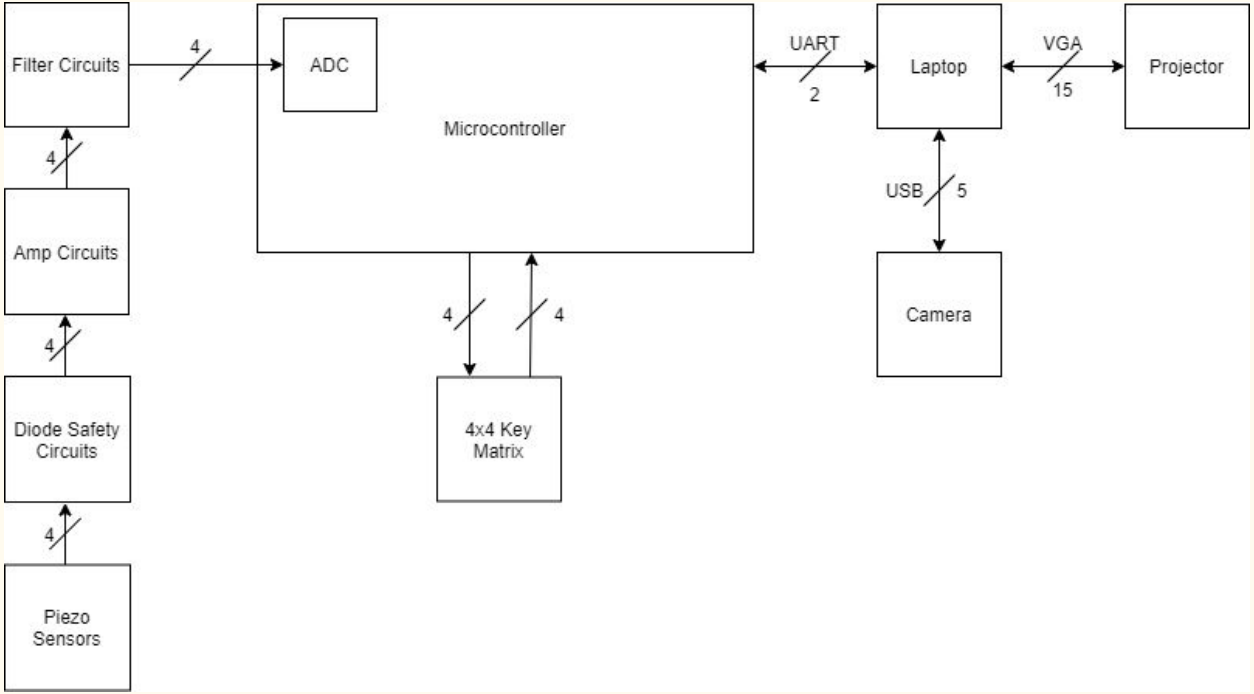
Real-time scorekeeping
- Using contact microphones
- Displayed by projector

Interactive game system
- User-modifiable parameters
- Unique game modes

Project-Specific Success Criteria

- An ability to handle 2-way communication between a microcontroller and a laptop through UART (HARDWARE)
- An ability to read key presses from a 4x4 keypad matrix (HARDWARE)
- An ability to amplify analog signals from contact microphones (HARDWARE)
- An ability to interpret microphone inputs to infer the score (SOFTWARE)
- An ability to read multiple inputs through 1 ADC controller with a circular buffer (SOFTWARE)

Block Diagram



Full System Block Diagram

Block Diagram (Cont.)

0x00 - sent on startup / boot
0x01-0x0F - invalid (bit 4 should be set)
0x10-0x1F - key press data
0x20 - 0x3F - blue bounce + key press data
0x40 - 0x5F - red bounce + key press data
0x60-0x7F - invalid (bit 7 should be set)

0x80-0xFF - errant reports...
0x8X - invalid packet
*0x9X - multiple press error
0xAX - double bounce on blue side
0xBX - indeterminate error type
0xCX - double bounce on red side
0xDX - indeterminate error type
0xEX - double bounce on both sides
0xFX - indeterminate error type

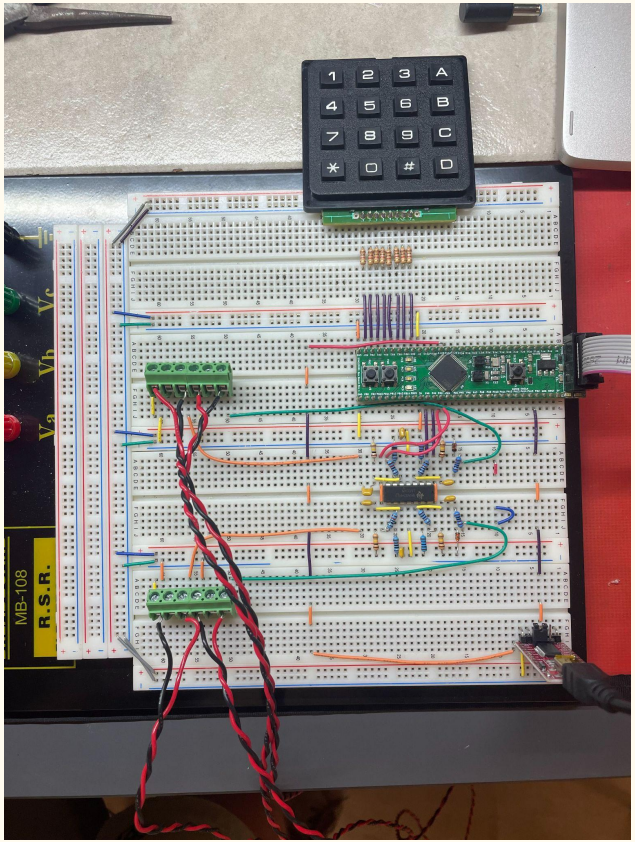
* still readable

UART Packet Structure - Microcontroller to Laptop

0x01 - Request Data Packet
0x02 - Force Software Reset
0x03 - Force Shutdown

UART Packet Structure - Laptop to Microcontroller

Major Components – Microcontroller



STM32F091RCT6

- 48 MHz Clock
- 12-bit, 1 MHz ADC
- 16 ADC channels
- 256 kB SRAM
- 8 UART ports
- Readily available!

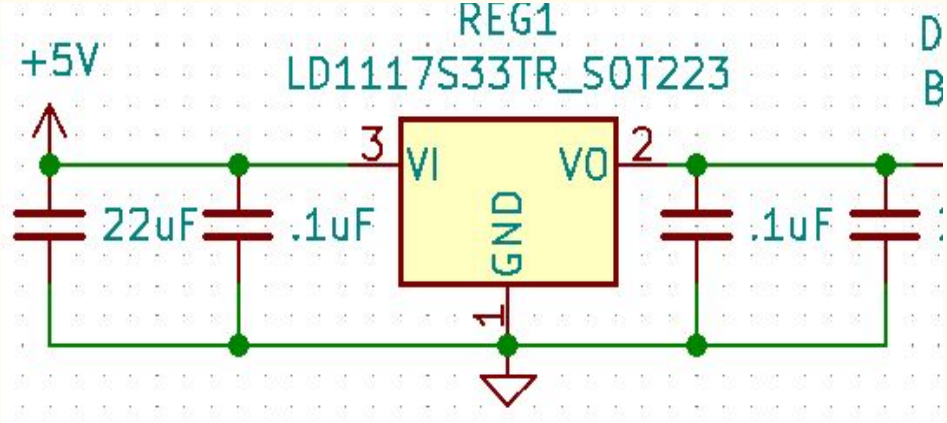
Handles button UI and bounce logic

Major Components - 5-to-3.3 V Regulator

FD1117S33TR

- Max current output: 800mA
- V(out) = 2.5V for V(in) = 4.5V
- Readily available!

Takes 5V from USB-C and provides 3.3 V to the rest of the circuit



Major Components - UART-to-USB

FT232RL

- USB Speed: Full (12 Mbps)
- I/O V: 1.8V to 5V
- Not readily available - LCSC was the only vendor with it

Allows communication between laptop and microcontroller

Major Components - Laptop

Lenovo ThinkPad T420

- Processor: 2 Core, 3.2 Ghz
- Intel HD Graphics 4000
- 8 GB RAM
- Needed IO: VGA, USB
- Readily available!
- Has RACHEL painted on it

Processes camera data, handles game logic, and sends graphics to projector

Major Components - Camera

ASUS Xtion PRO Live

- 5:4 Aspect Ratio
- 1280 x 1024 Pixels
- 60 FPS RGB Sensor
- 30 FPS Depth Sensor
- Readily available!

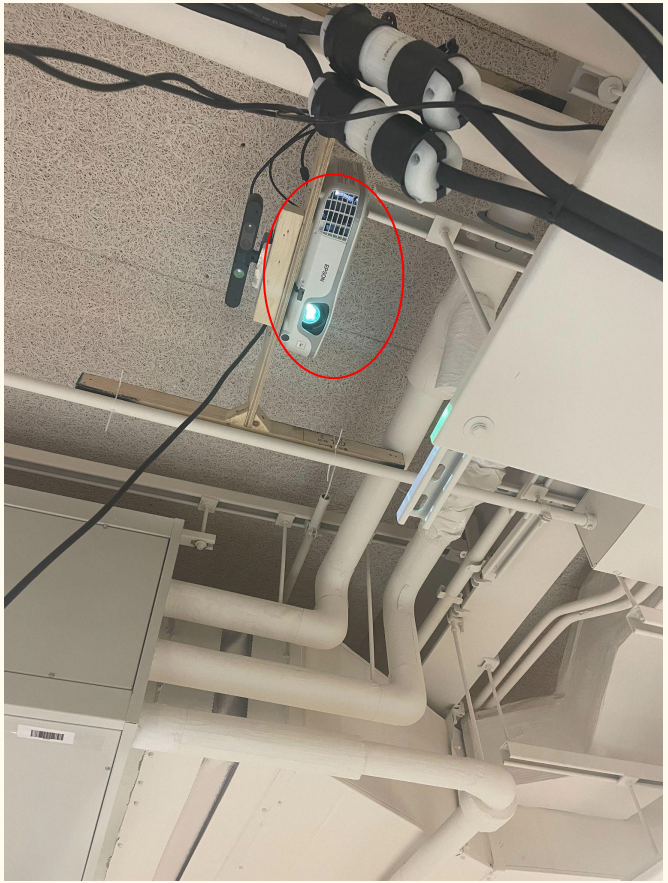
Determines the location of the ball

Major Components - Projector

Epson PowerLite X12

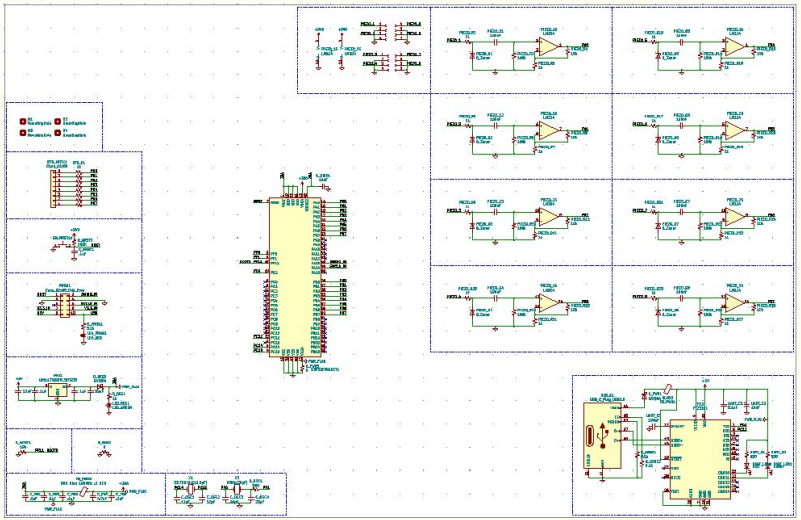
- Throw Ratio: 1.48-1.77
- Brightness: 2800 lumens
- Resolution: 1024 x 768
- Interface: VGA
- Readily available!

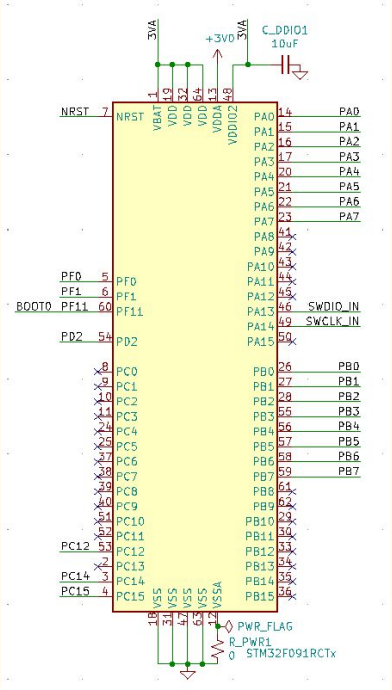
Projects the game on the table



Electrical Schematic

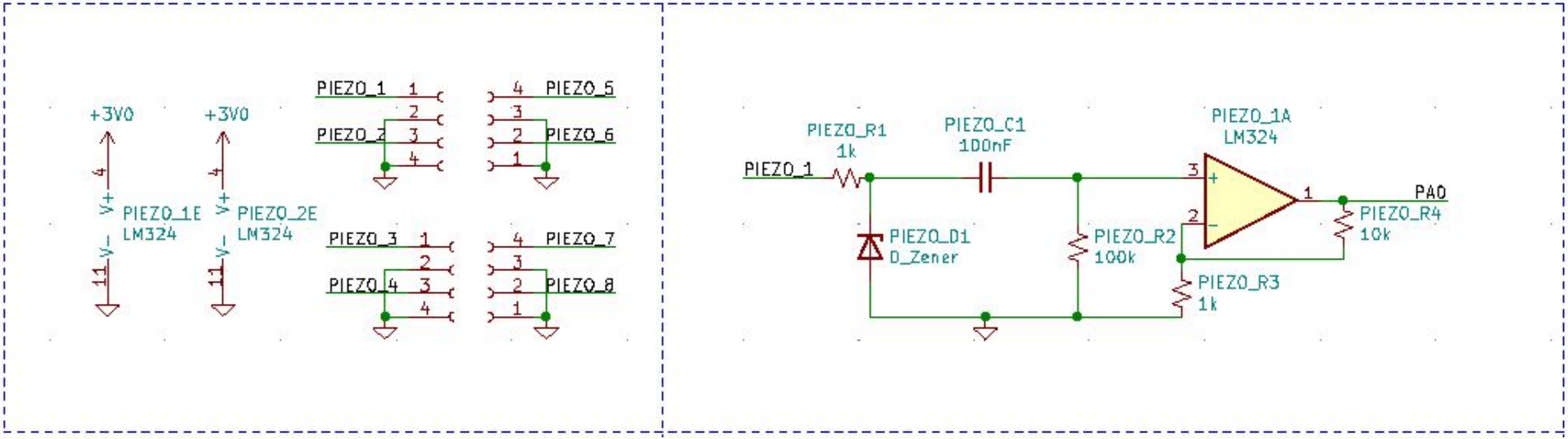
- Software: KiCAD
- Potential Vendors: JLC-PCB, OSH Park





MCU Schematic and all its utilized pins

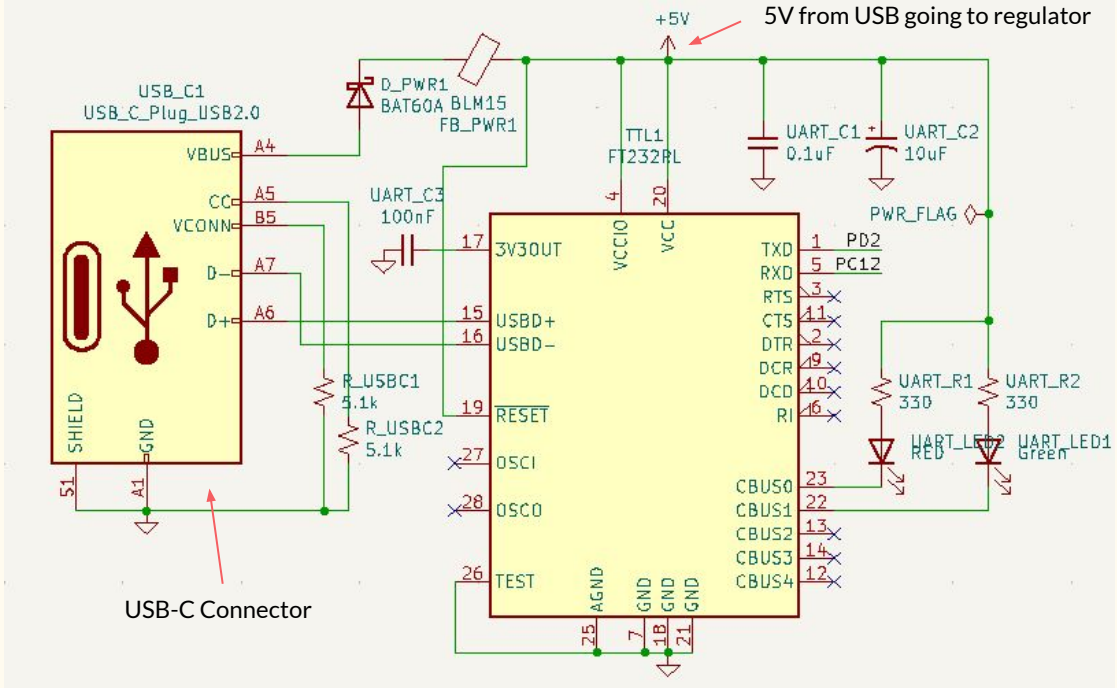
Electrical Schematic - Contact Microphones



High Pass Filter: 16Hz+; OpAmp Gain: 11

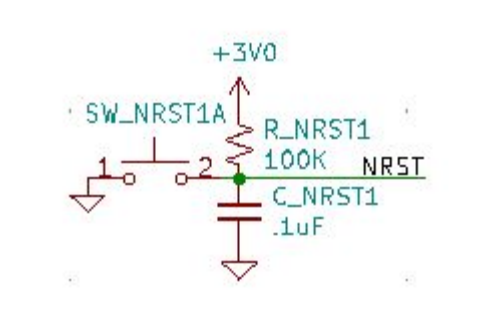
Left - 16 connections for 8 microphones; Right - Individual microphone circuit

Electrical Schematic - UART-to-USB

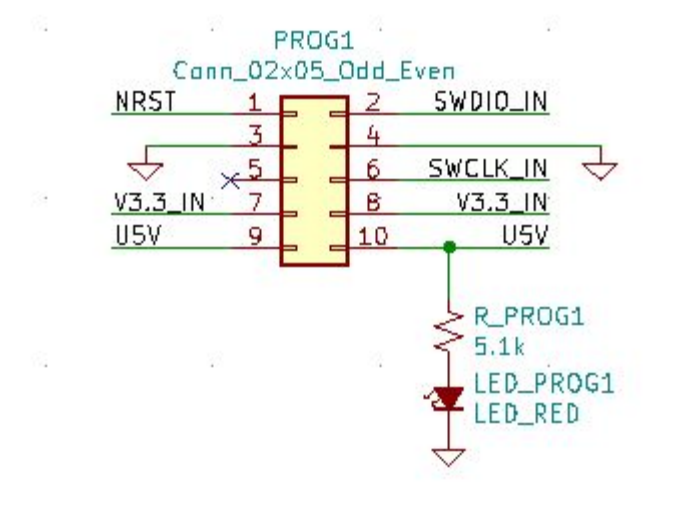


Utilizes a USB-C connector and FT232RL

Electrical Schematic - Programming



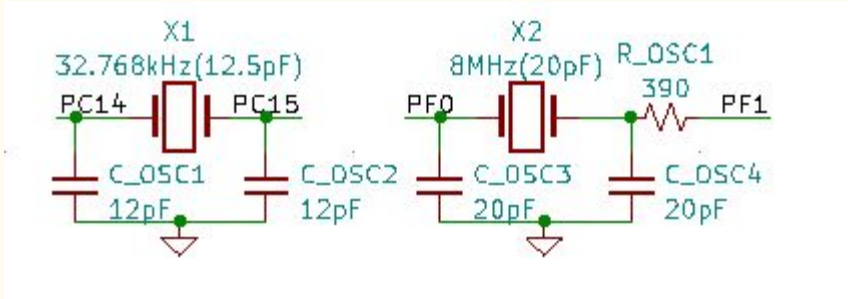
nRST switch



SWD Programming Header

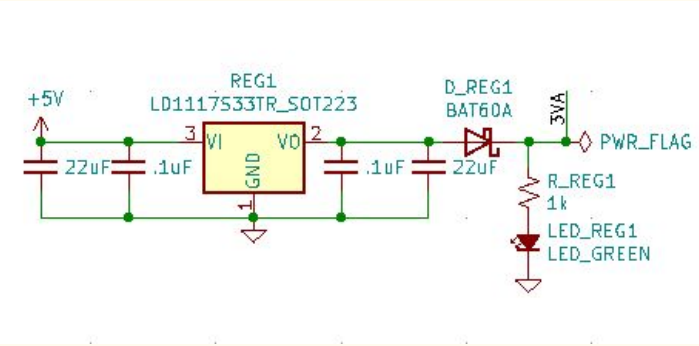
Note: 5V & 3V from the programming header are not connected to the rest of the circuit, they are left floating

Electrical Schematic - Clock Oscillators

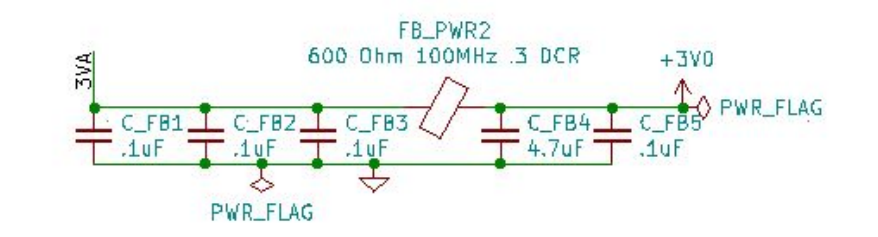


The choice of oscillators are based on ECE 362's development board

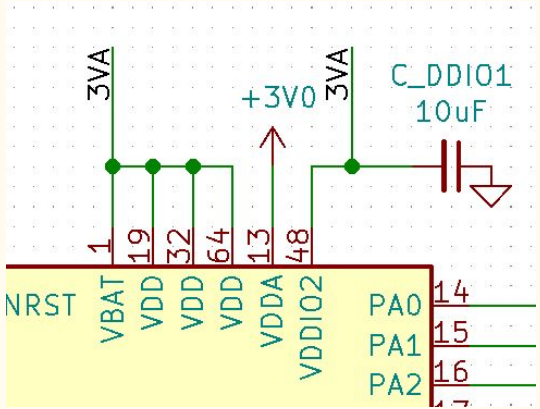
Electrical Schematic - Power



5to3V Regulator



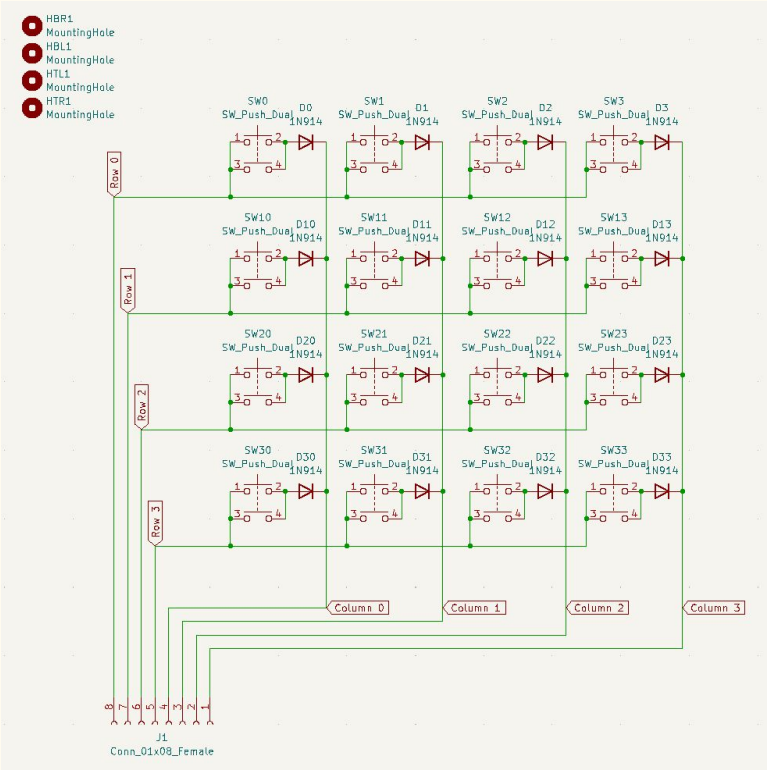
Decouple + Ferrite Bead



MCU Power Input

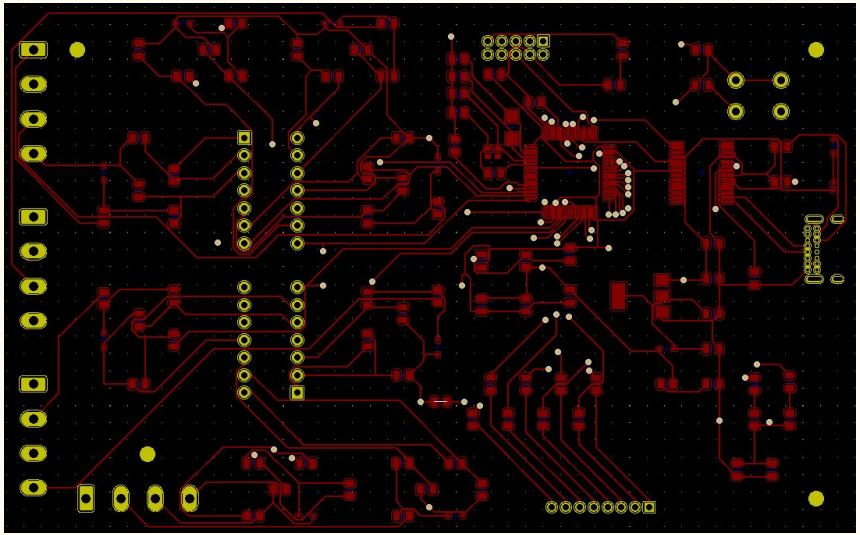
Power to VDDA is decoupled before going into the MCU

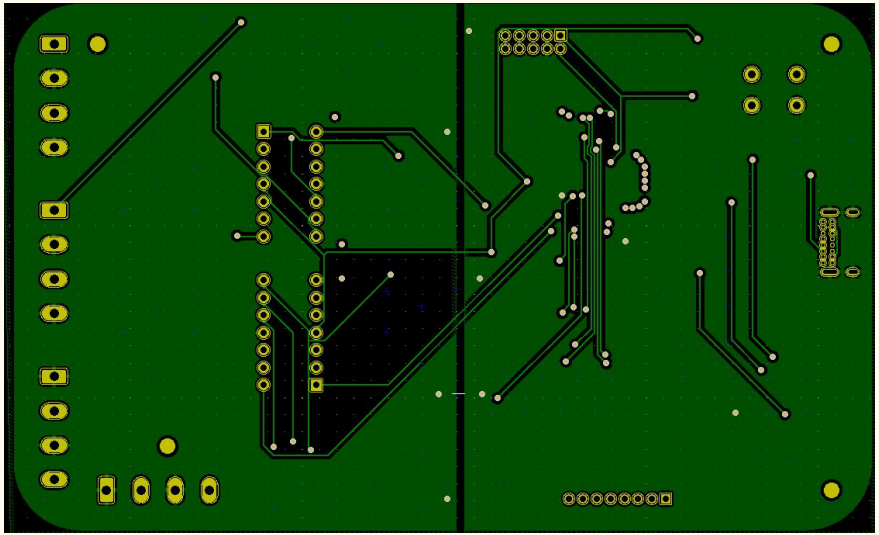
Electrical Schematic - Button Matrix



* Custom 4x4 Matrix with anti-ghosting diodes

PCB Layout - Top vs. Bottom

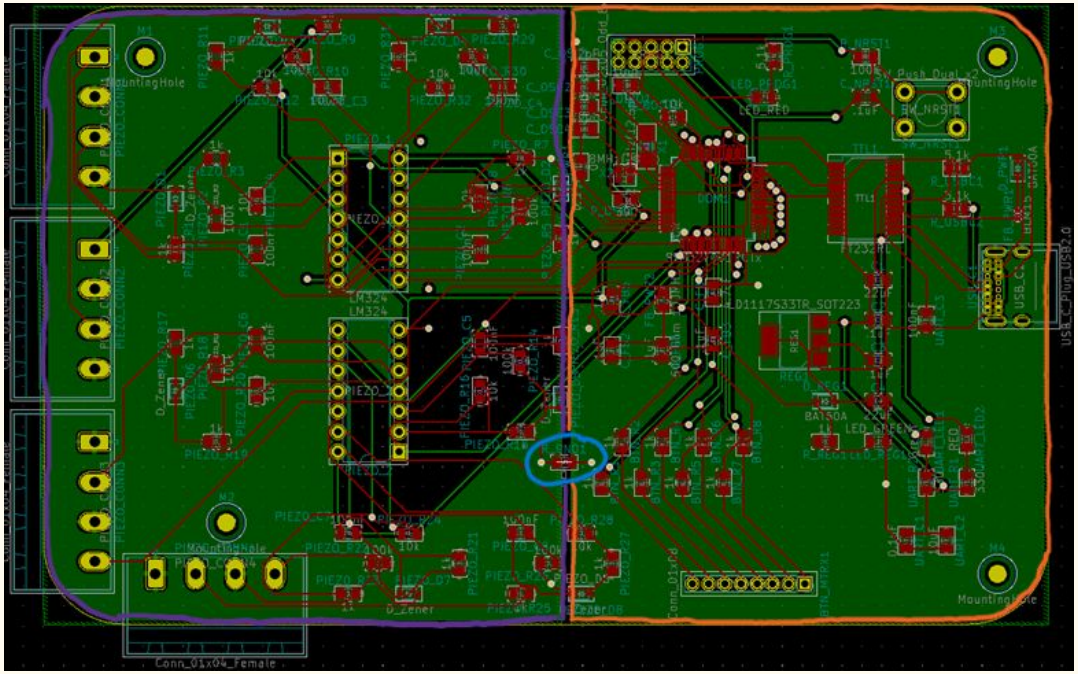




- 2 layer board
- ~125mm by ~80mm
- Bottom layer consists of mainly ground planes
- *We have a ground plane below clock oscillators

PCB Layout - Analog vs. Digital

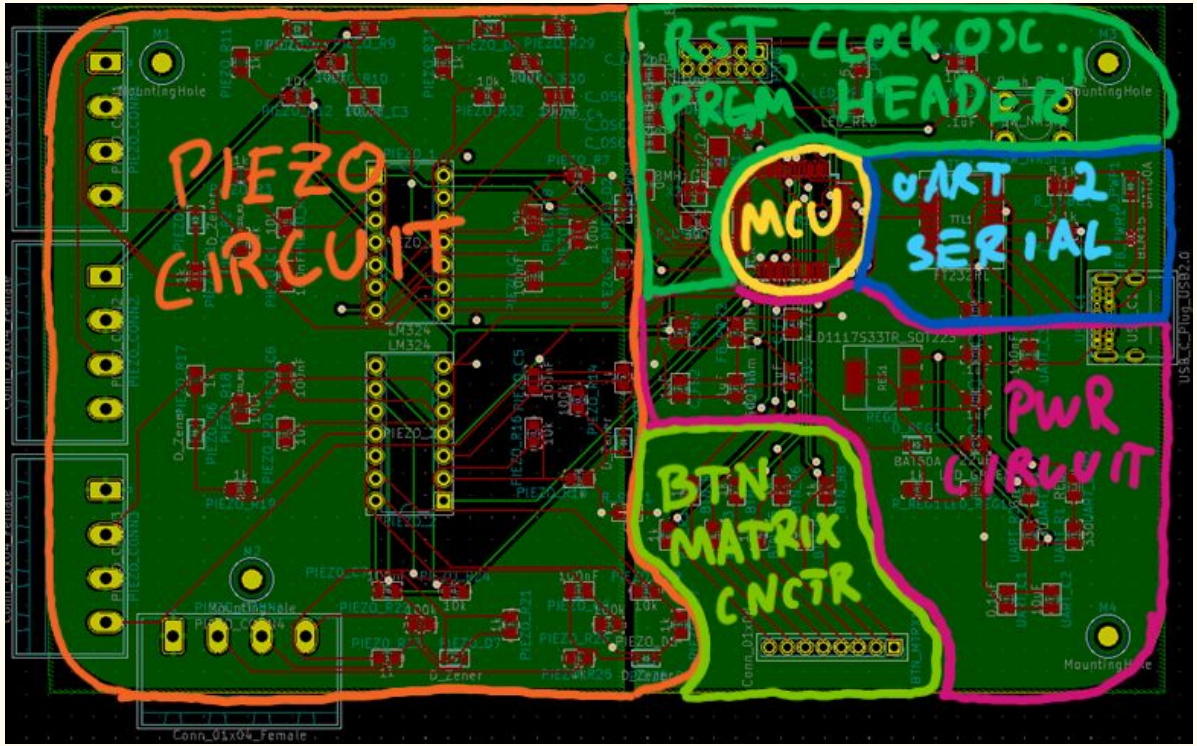
- First major decision on PCB placement was to put all analog on a separate ground plane
- Both ground planes are connected with a 0 ohm resistor



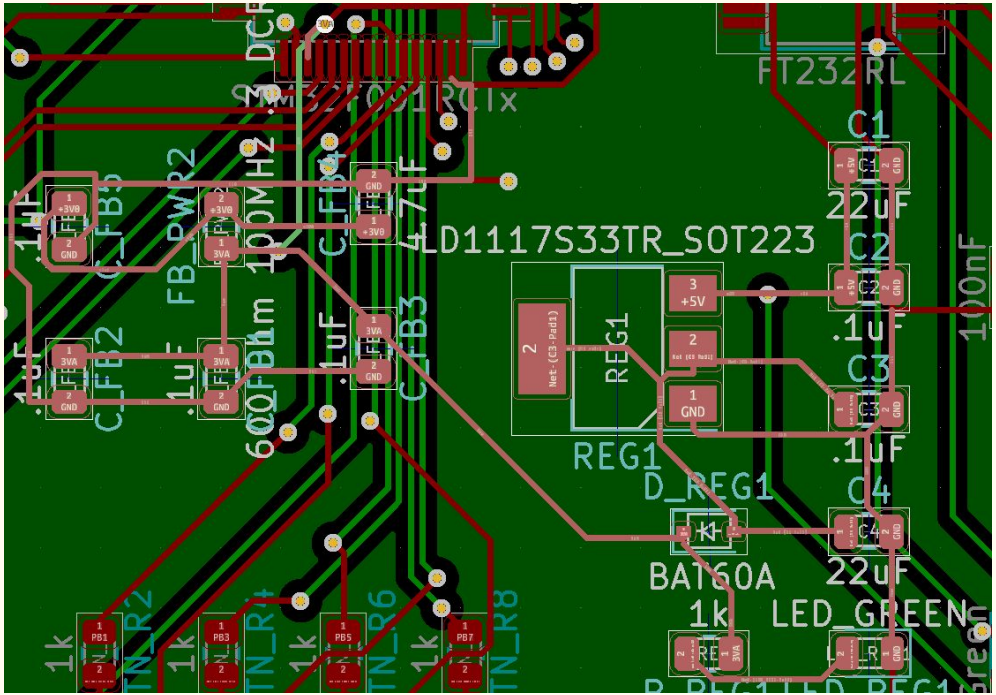
Left, purple outlined side is analog; right, orange outlined side is digital; blue circled 0Ω resistor connects both ground planes

PCB Layout - Regions

- Piezo Circuit - holds piezo connectors, high pass filters, and op amps
- Btn Matrix Connector - connection to the button PCB
- Pwr Circuit - 5 V USB-C input through 5to3V converter
- USB 2 Serial - communication with laptop method
- Other - clock oscillators, reset switch, programming header

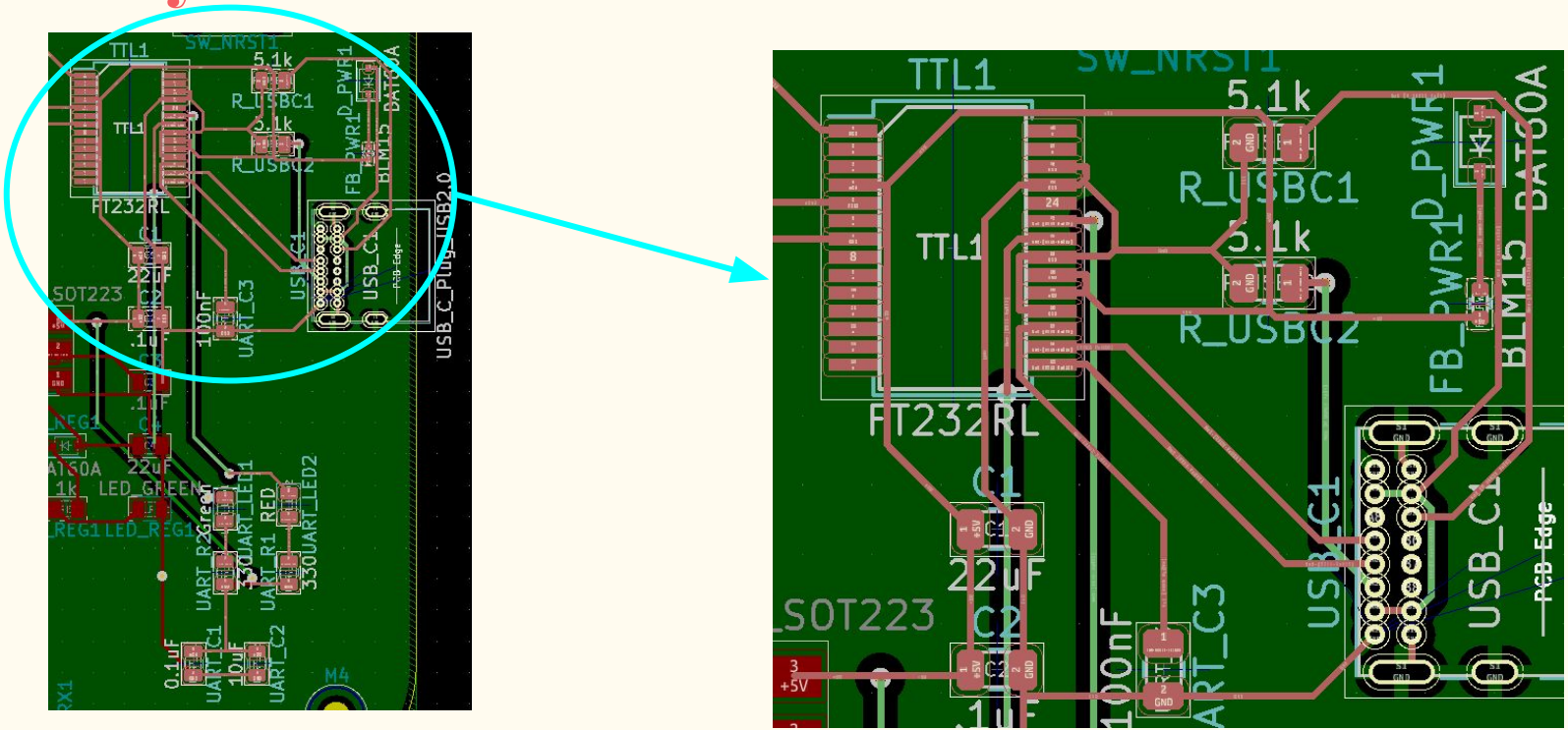


PCB Layout - PWR Circuit



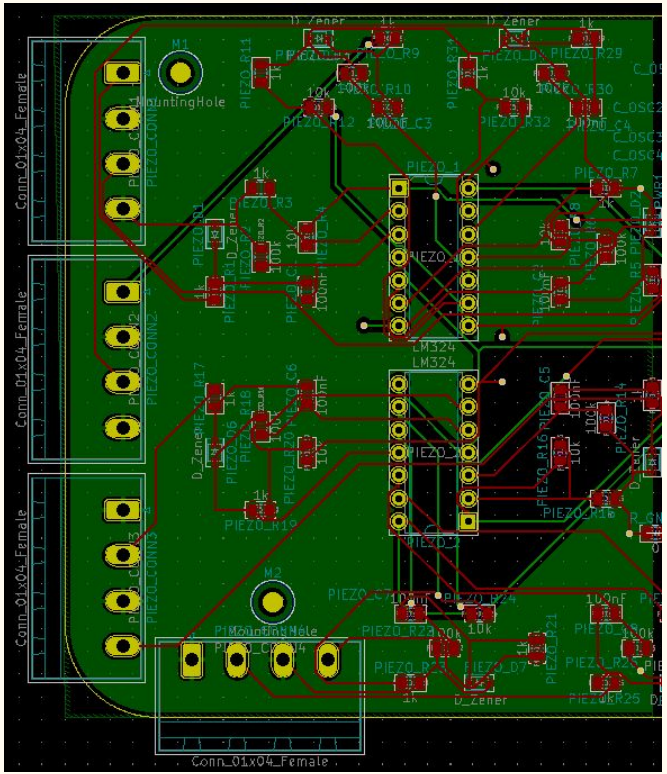
Regulator (Right) is fed 5V from USB-C connector and then decoupled further before going to the MCU (Top)

PCB Layout - UART-to-USB



FT232RL (top left) receiving input from USB-C (right); LEDs and decoupling caps on bottom

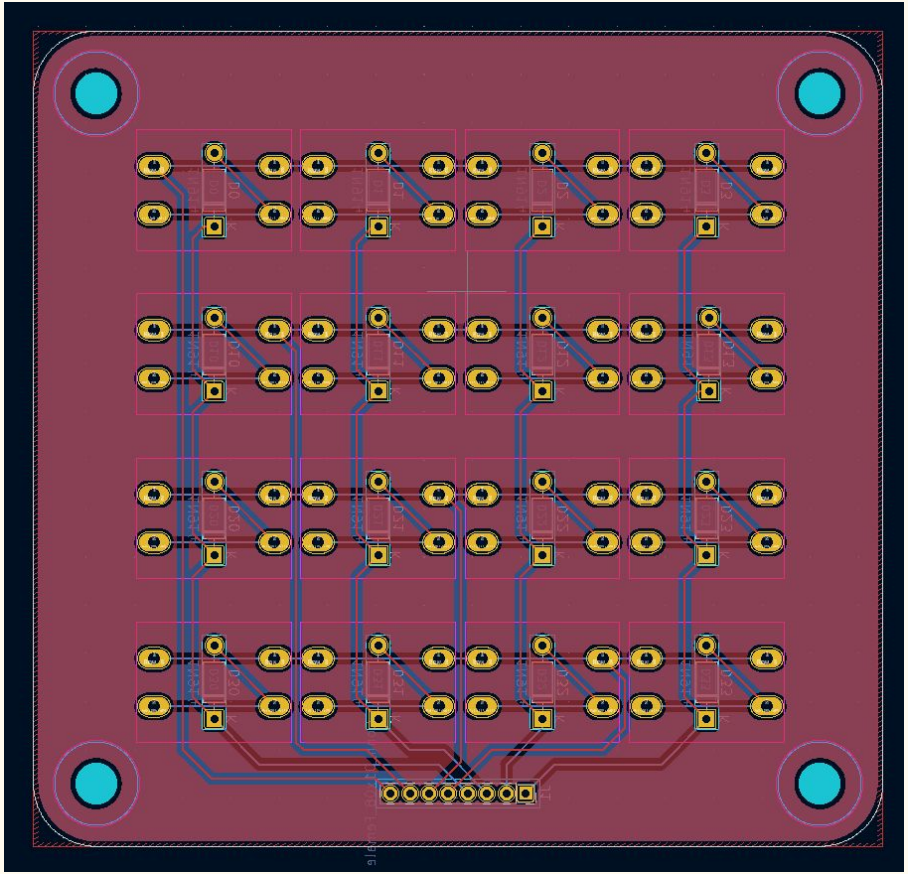
PCB Layout - Microphone Circuit



Connectors to Piezos and their corresponding circuits are in groups of 6 around the Op Amps

PCB Layout - Button Matrix

- All parts soldered with through hole joints
- Dimensions match silicone cover found in lab
- Each Button has corresponding anti-ghosting diode mounted to the back of the PCB
- 8 wide bus for connecting to main PCB



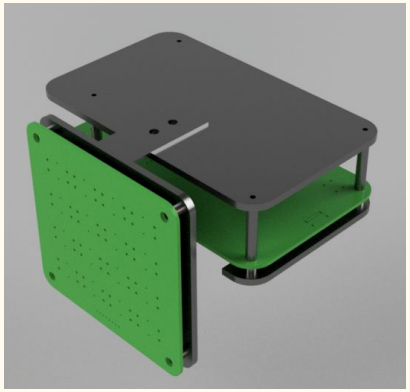
Packaging Design - Overview

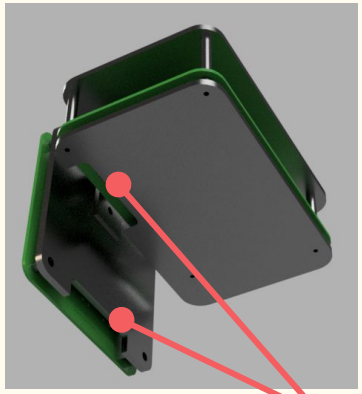
Our packaging is designed using a cut acrylic sheet, standard M3 and M4 standoffs, M3 and M4 screws and a single L bracket.

Advantages:

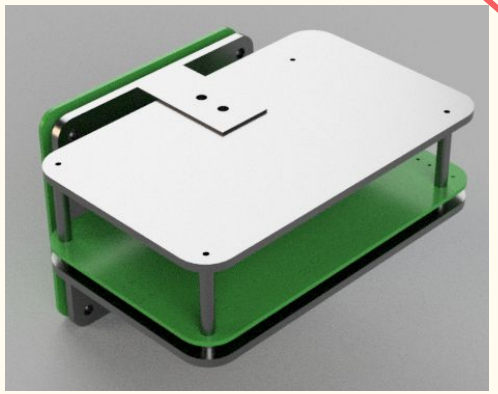
- No 3D printing required
- Full assembly incorporates main pcb and button matrix

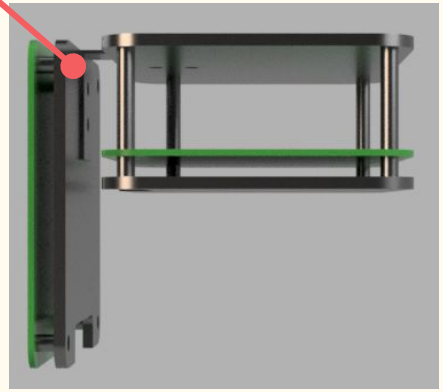
Table Mount Point



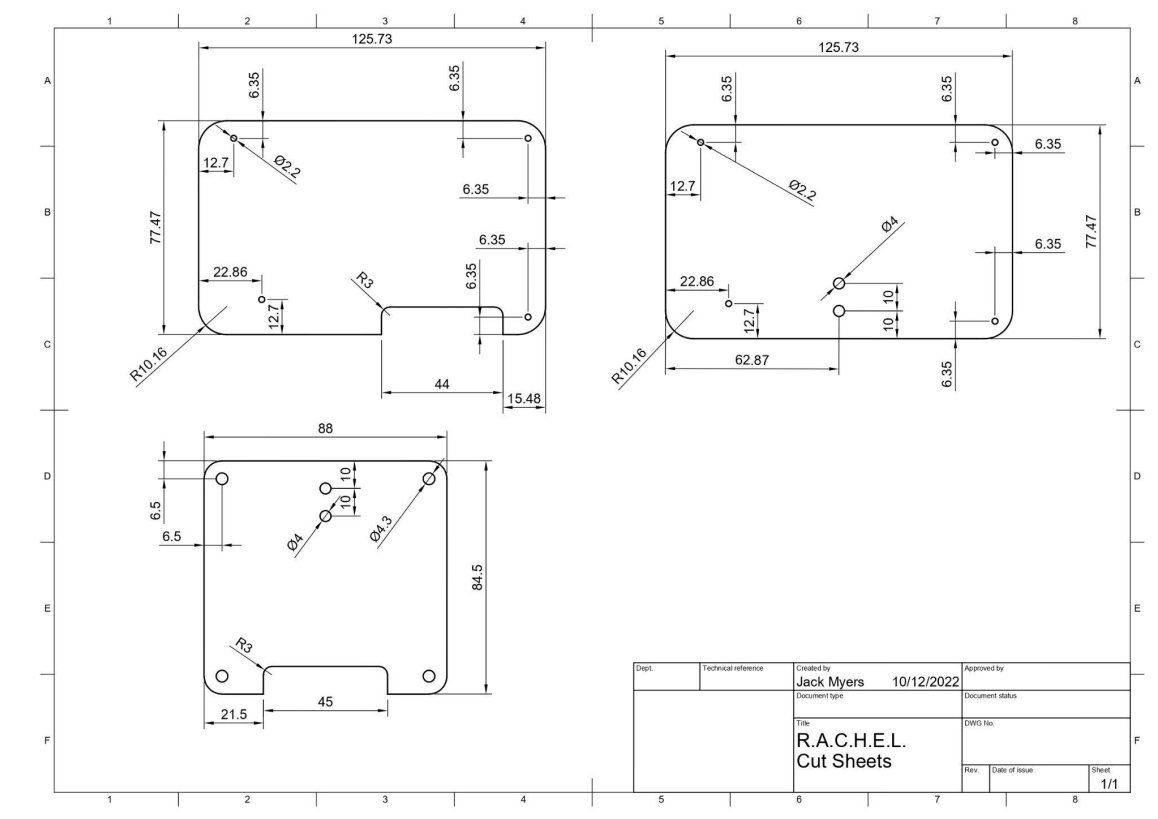


Cutouts for 8 Wire Bus



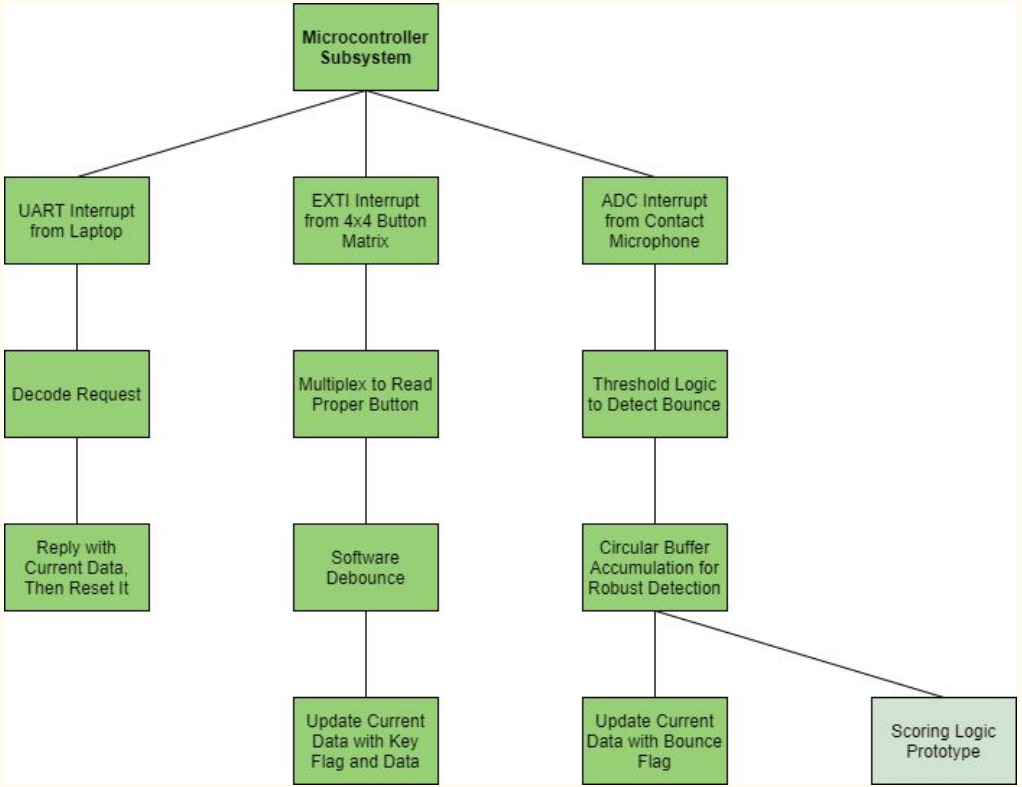


Packaging Design - Schematic

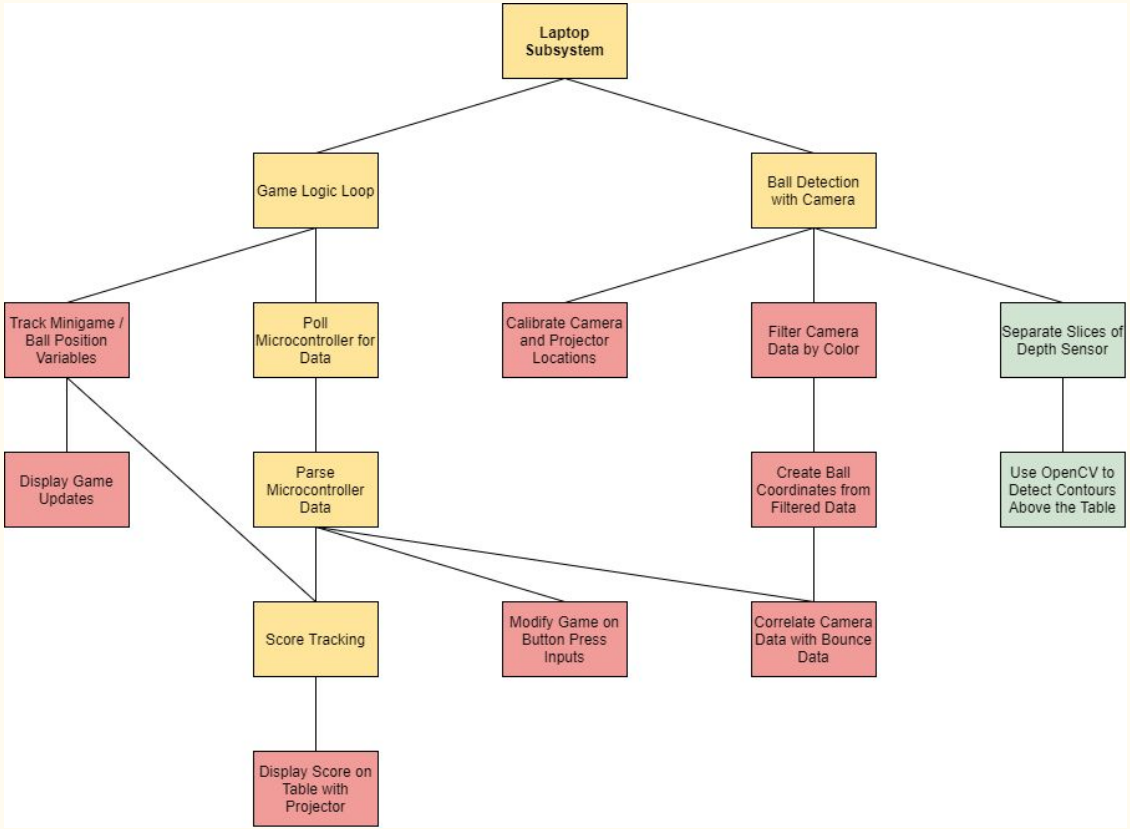


* We plan to fabricate these 3 sheets using a laser cutter. The rest of the hardware will be purchased.

Software Development Status



Software Development Status (Cont.)



Prototyping Progress - 4x4 Keypad Matrix

Periodically sets rows high and polls columns

Uses history buffer for software debouncing

EXTI interrupt on high columns

Sends pressed key to laptop via UART

Current implementation - not protected from ghosting

- To be resolved by custom PCB

Prototyping Progress - Contact Microphones

Periodically poll 4 ADC channels' conversions, circularly

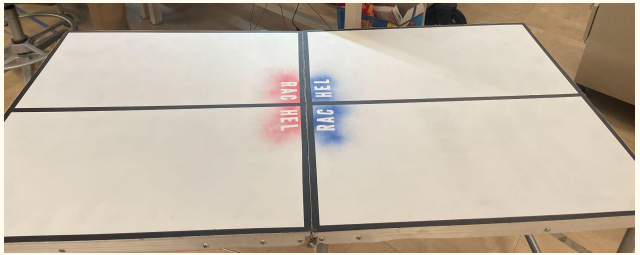
If conversion > ADC threshold,

- Sum the "red" channels for 40 rounds
- Sum the "blue" channels for 40 rounds
- The greater sum records a bounce

Send bounce data to laptop via UART

Currently using a hardcoded threshold (400)

- Automated calibration is WIP



Prototyping Progress - Game Logic

Simple scorekeeping by detecting double bounces

Works well for “normal” situations

Currently no support for some standard rules, e.g:

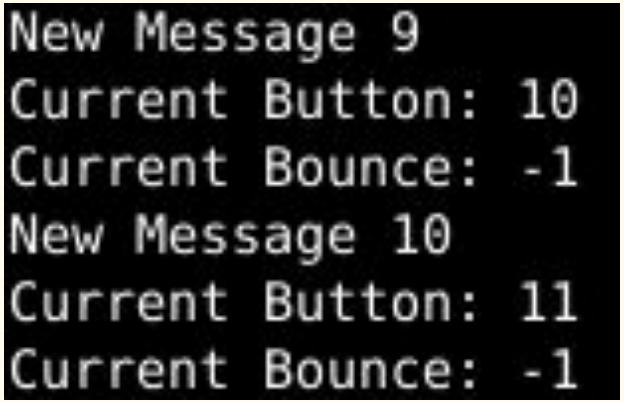
- diagonal serve (achievable)
- let (maybe)
- vertical edge exclusion (definitely not)

Prototyping Progress - UART-to-USB

Reading UART messages from the laptop requires us to read from a linux TTY file representing the USB connection ("/dev/ttyUSB0").

Using this interface and the C++ functions and definitions found in *termios.h* we can reliably configure, send and receive UART messages.

The messages sent contain little enough information to fit in a one byte message.



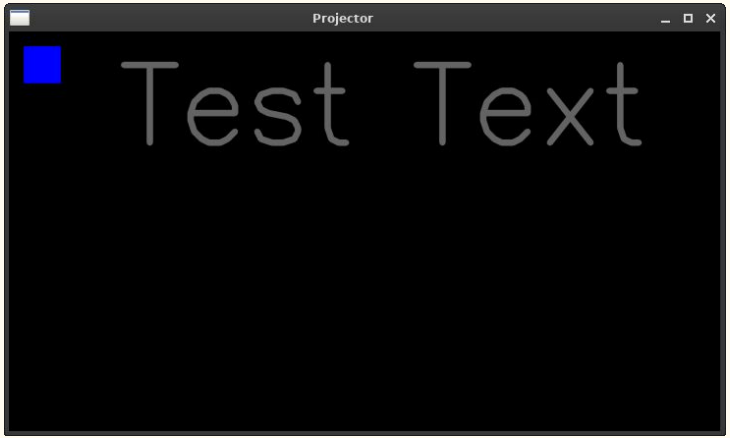
Read Message when Button Pressed

Prototype Progress - Projector

Ability to display to the projector using OpenCV.

Abstracted interfaces such as *writetext()* and *drawsquare()*, *refresh()*

This functionality should be expanded as we develop game graphics



Test Drawing Using openCV

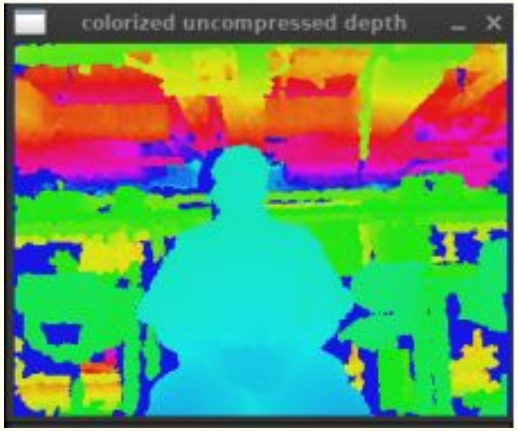
Prototyping Progress - Ball Tracking

Initially focused on depth data

Color is more reliable, easier to implement

Ball tracking is a low priority because it isn't a PSSC

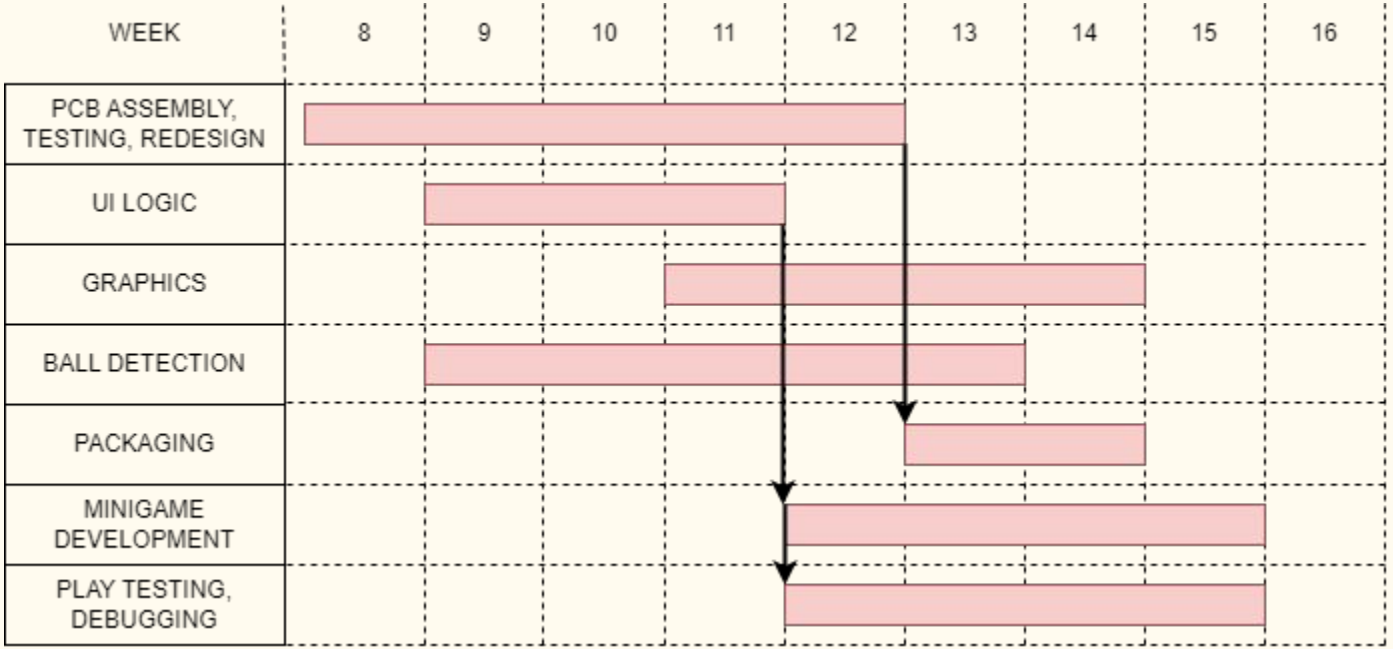
Future plans to incorporate camera data to assist with scoring in nuanced situations



Project Timeline

- Week 8: recheck PCB (potentially reorder), check/order button PCB, project score
- Week 9: ball detection, UI logic
- Week 10: ball detection, graphics
- Week 11: ball detection, methodically begin soldering
- Week 12: ball detection, debug circuit
- Week 13: ball detection, test circuit
- Week 14: minigame implementation, packaging
- Week 15: playtest, debugging
- Week 16: showcase

Project Timeline



Questions?