MDR: Amped Up!

An Augmented Guitar Amp Experience

Team

Aaron Hanley
(CompE)

Casey Massar
(CompE)

Remy Yoo
(EE)

Chris Caron
(EE)

Introduction: A Quick Review of Amped-Up

- There are a lot of different makes and models of amps, and custom tuning your parameters inefficient and limiting
- Tone shaping through the use of tonal parameters
- We hope to create an easy to use interface that attaches to virtually any amp to control those parameters in real time and recall saved presets

Project Goal

To create a system that attaches to an amplifier and controls its knobs in order to provide musicians with a more powerful interface to control it and create new sounds

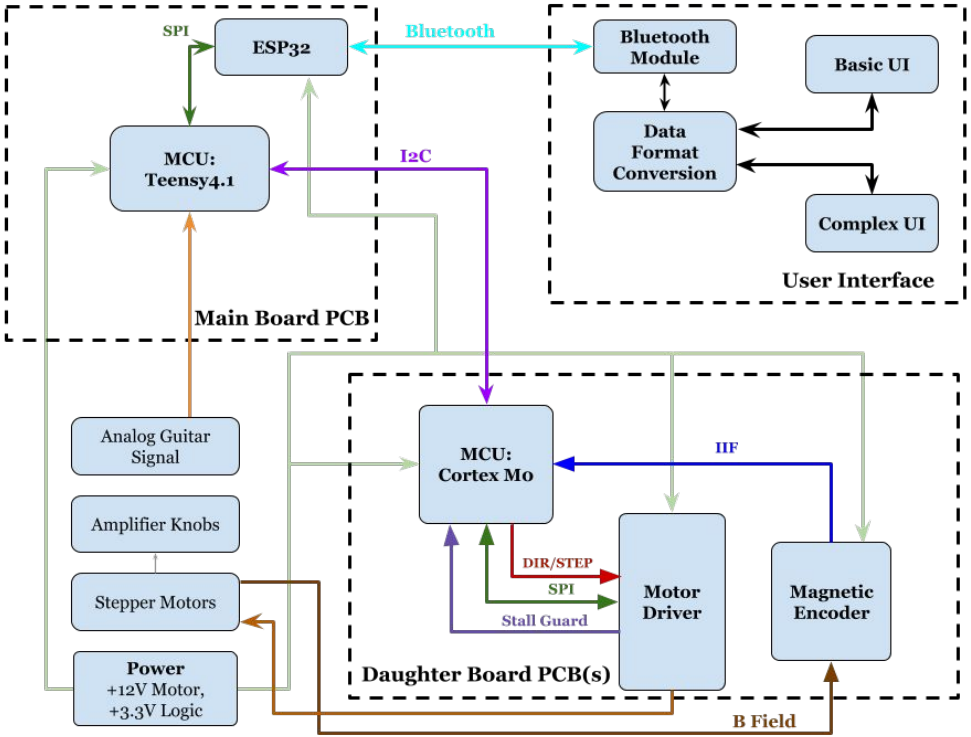
Use Case

		
High Notes (Solos)	Desirable tone (Clear, Mellow)	Harsh, distracting, too much echo (“wet”)
Low Notes (Rhythm)	Muffled, difficult to distinguish individual notes (“dry”)	Desirable tone (Bright, Nuanced)

System Specifications

- Is compatible with existing amplifiers, allows manual override
  - Electromechanically can control at least 8 knobs
  - *Able to turn from extreme to extreme in at most 80 ms*
- Controls tone shaping parameters for guitar sound, including:
  - Gain, EQ, Post-amp Effects, Volume, Channel Selection
- Supports locally saving and recalling user configurations (10+ configurations)
- Can produce dynamic sounds by modulating the amplifier's tone shaping parameters in real-time
- Supports arbitrary automations to control the amplifier parameters according to:
  - Guitar signal volume, Generic footswitch, Expression pedal, Mathematical functions, Guitar signal frequency
- Features user interface to interact with customizable functions
  - Must be robust enough to adequately control all features of the amplifier

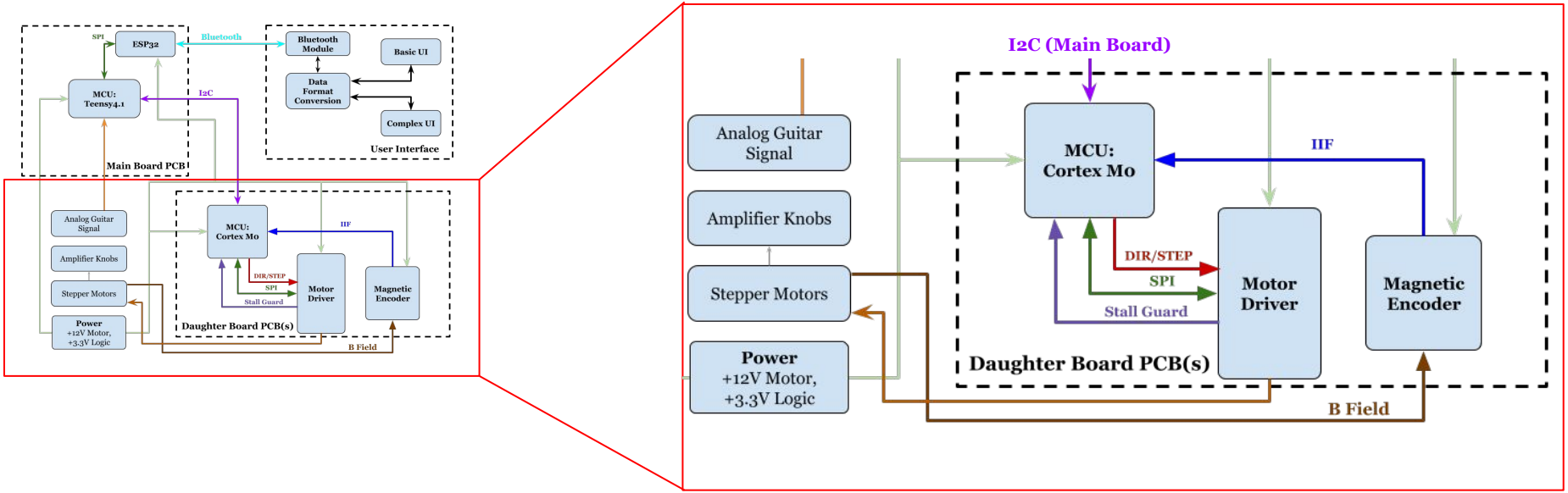
Updated Design Diagram



Challenges and Vital Systems

- Daughter Boards
  - Closed Loop Stepper Control System, Mechanical Design, Communication
- Main Board
  - Node Graph Implementation, Communication, Preset Storage
- User Interface
  - Basic and Complex GUI
  - Data Formats

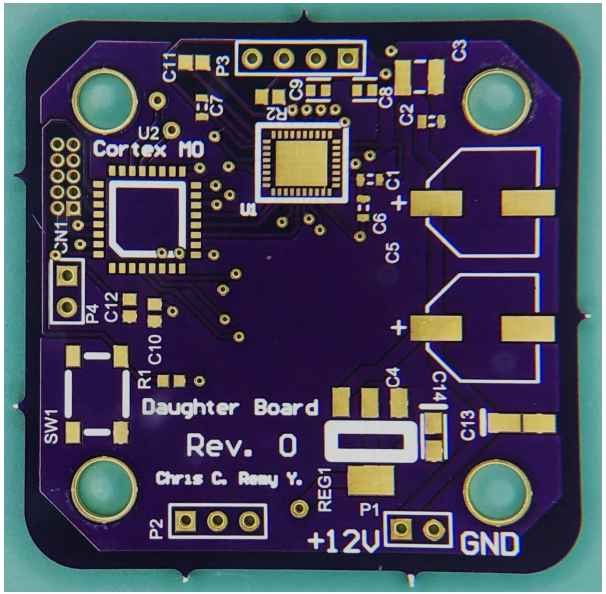
Part 1: Daughter Board and Motor Control

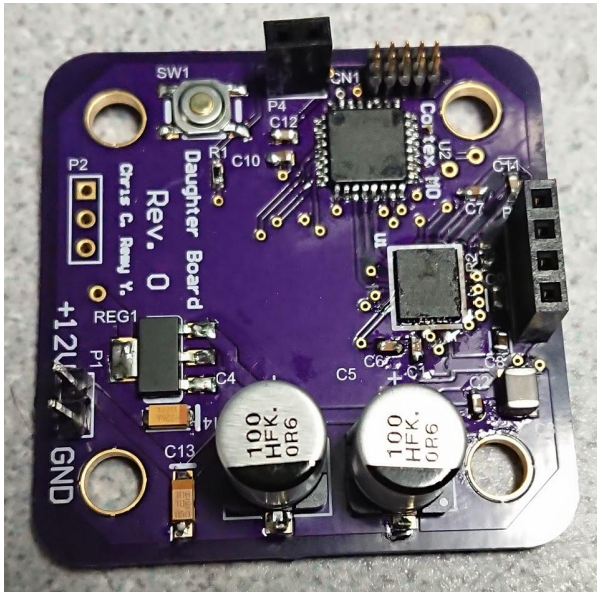


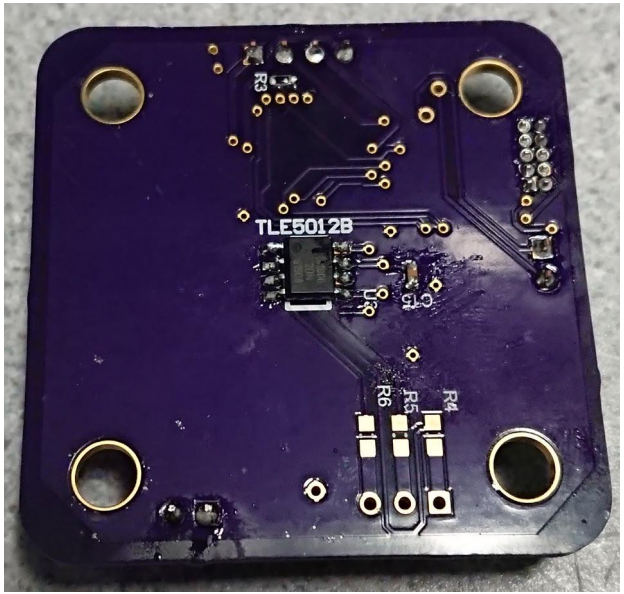
Main Functions

- Receives Desired Knob Position from the main board(over I2C), and uses a closed loop motor control circuit to solve for and move to position.
- Custom PCBA: Cortex M0, Trinamic Motor Driver, Magnetic Encoder, Voltage Regulator
- Mechanical Design: Stepper Motor Mounting Mechanism, Full Enclosure

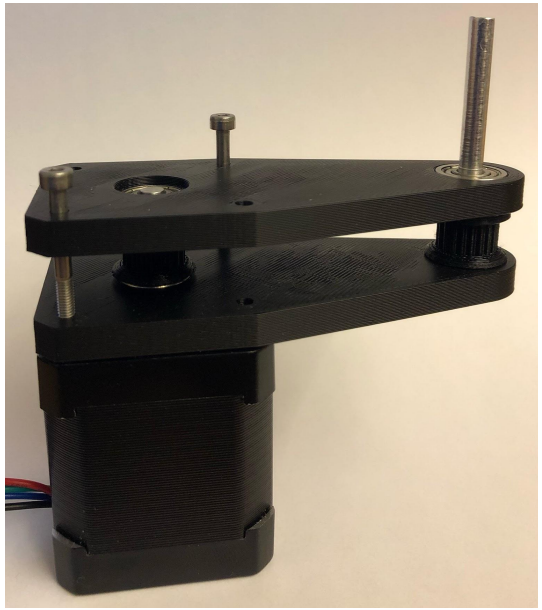
Daughter Board PCBA

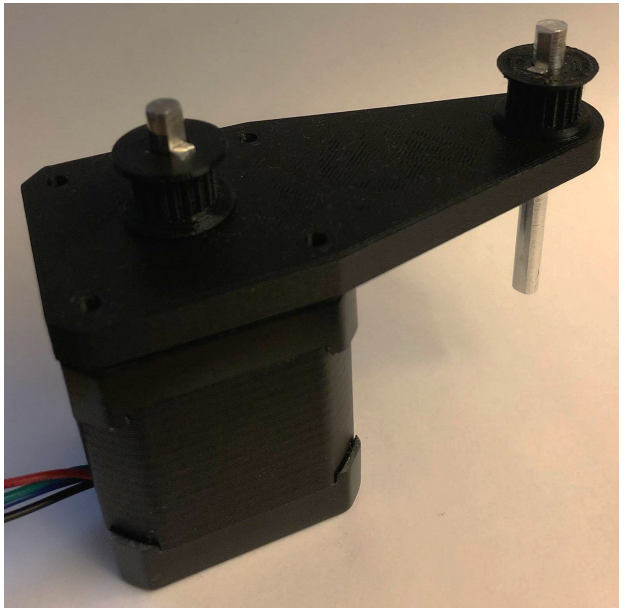




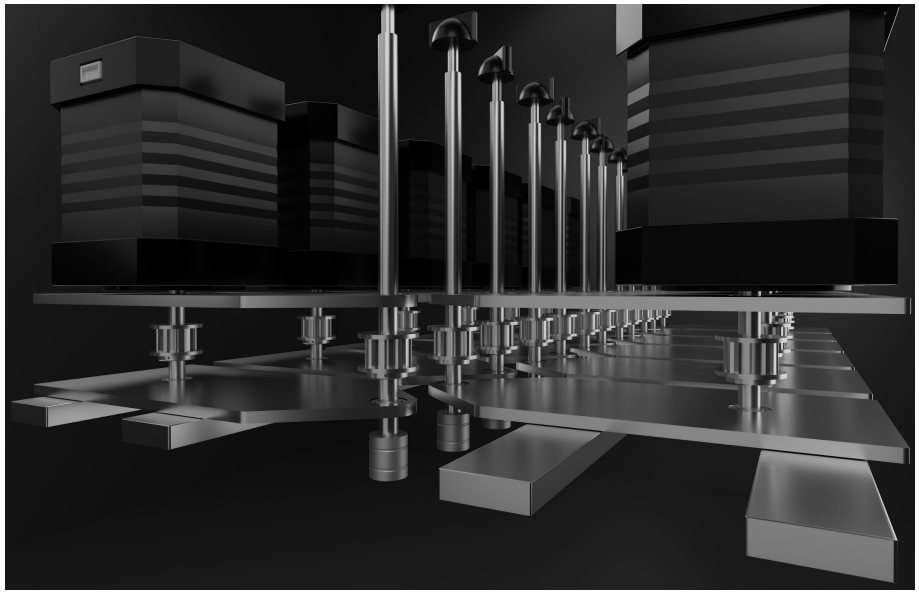


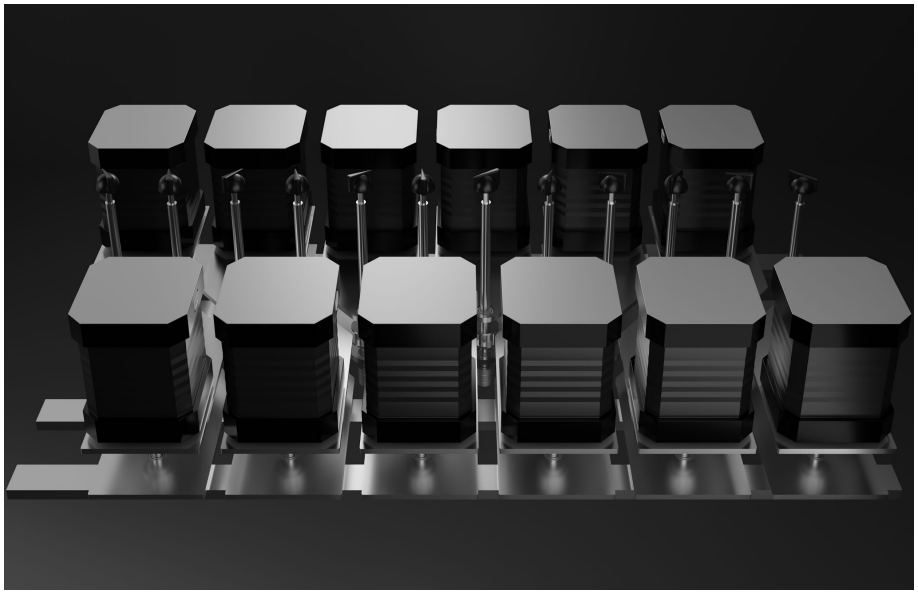
Mechanical Interface: CAD & 3D Printing



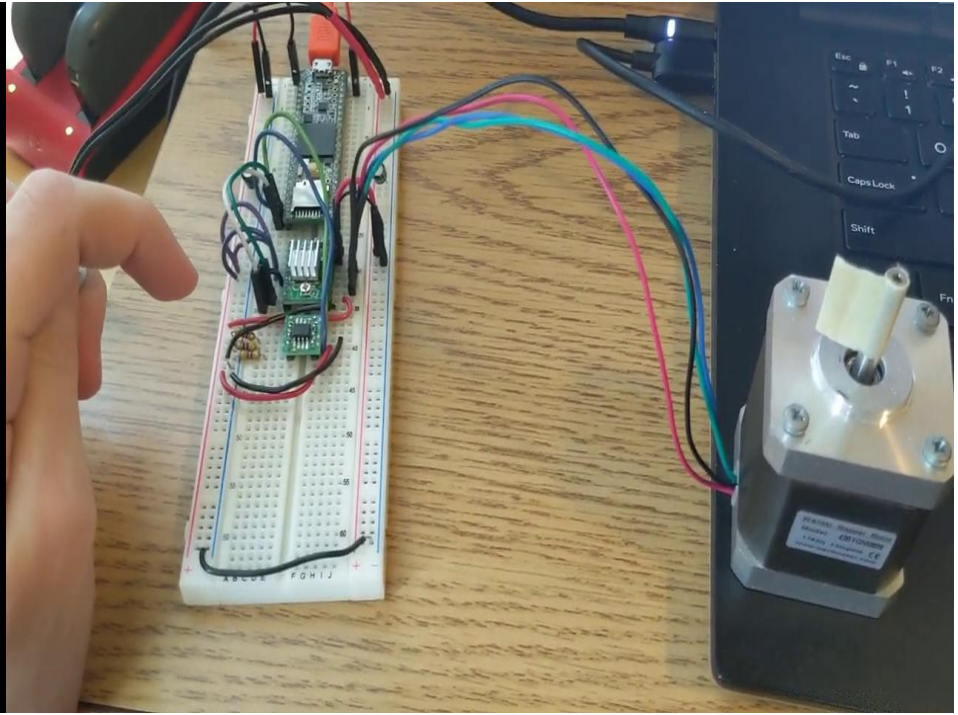


Mechanical Interface: Concept Renders

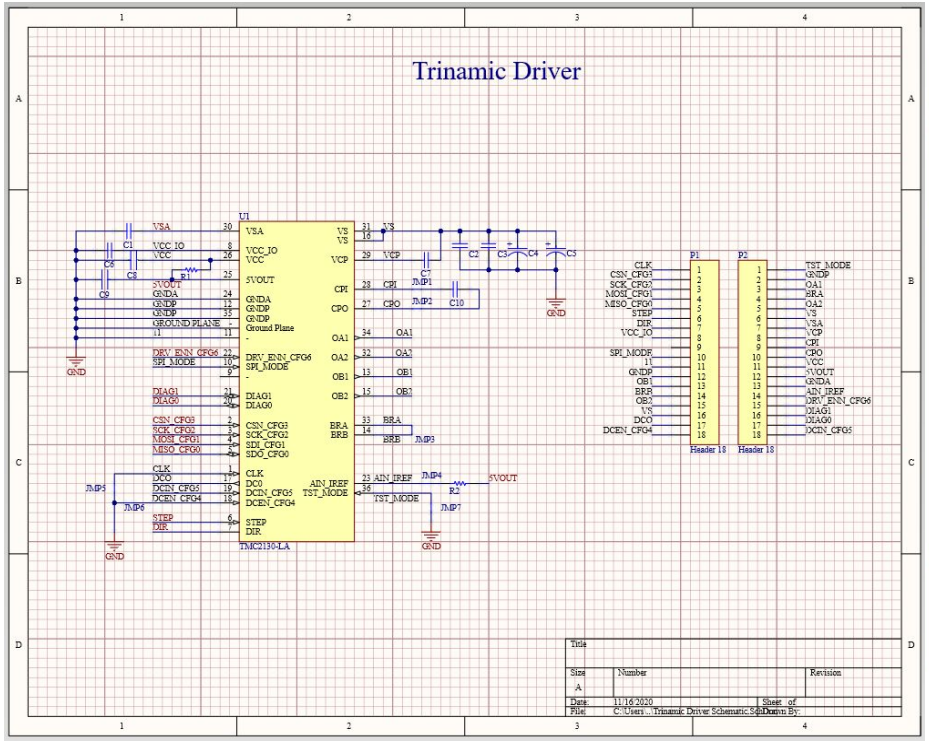


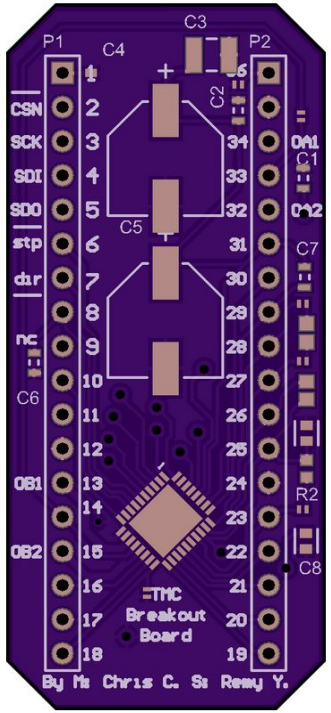


Motor Control and Magnetic Encoder Demo

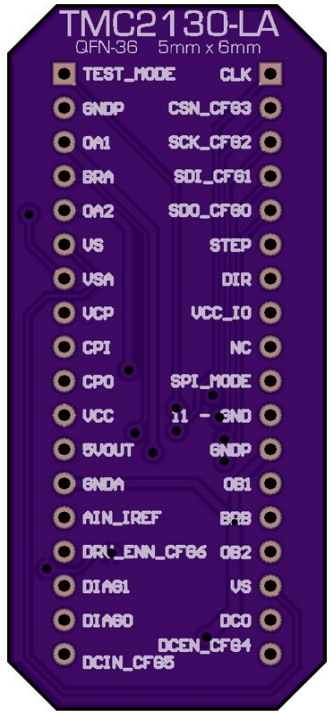


Closed Loop Control: Custom Breakout



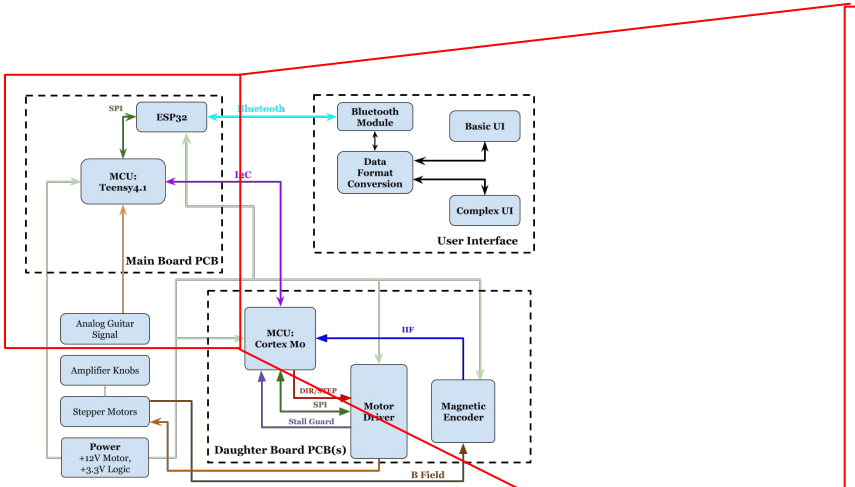


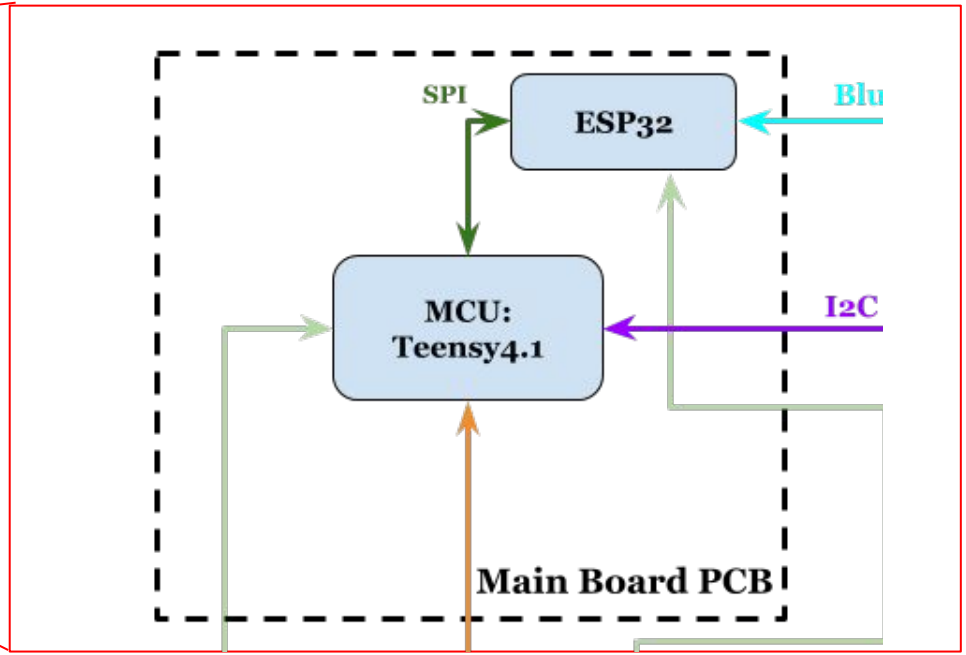
Top



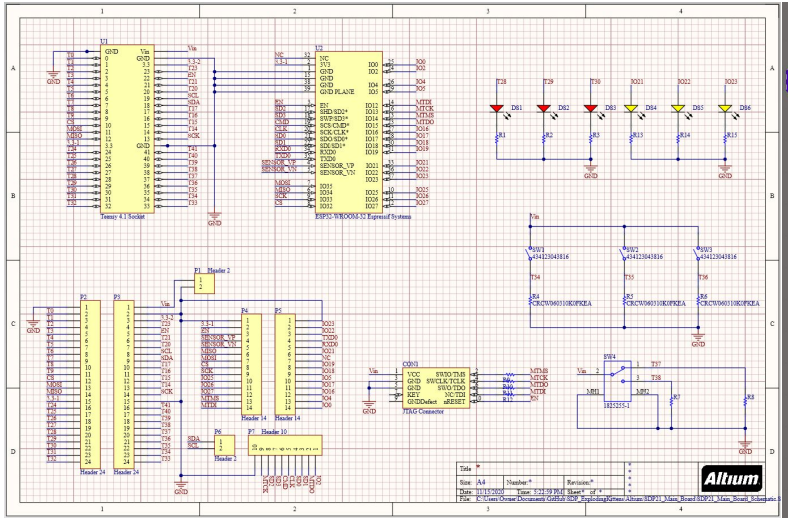
Bottom

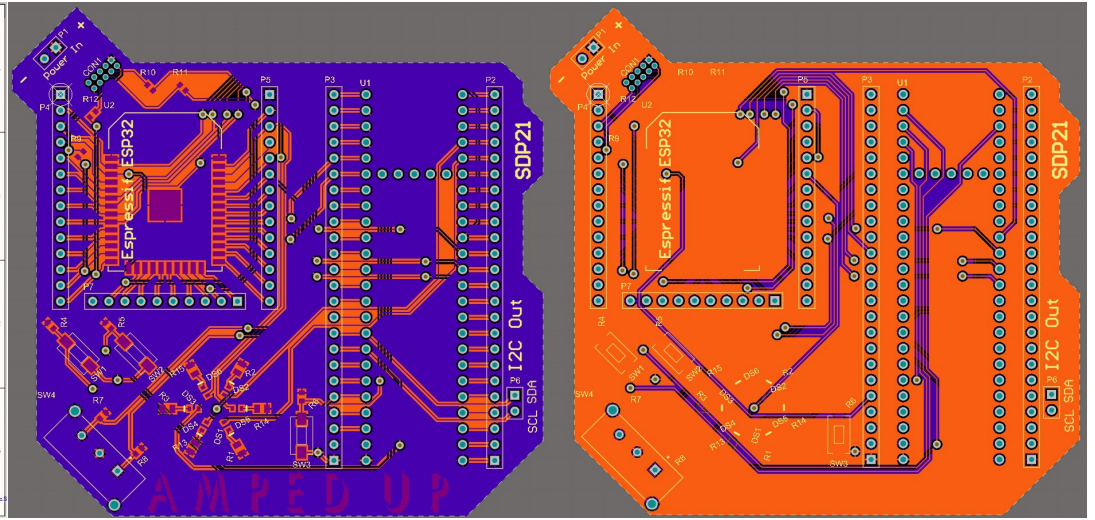
Part 2: Main Board and Communications





Main Board: Altium Design





Main Processor: Teensy 4.1

- Calculates knob positions based on node graph
- Sends updated knob positions to daughter boards
- Receives node graphs created by user using remote GUI

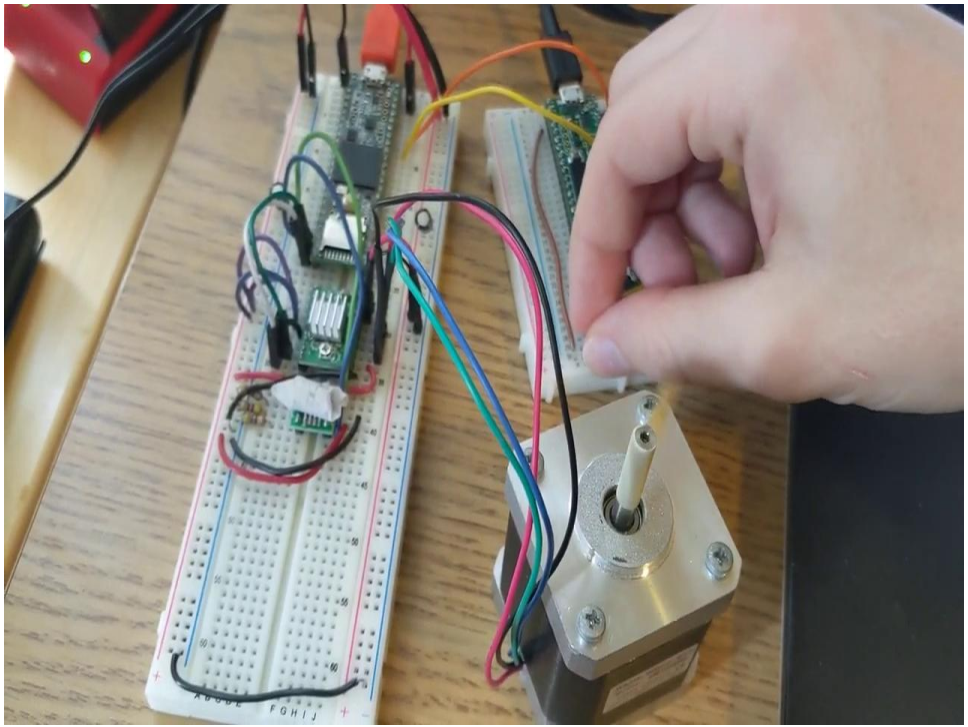
Communication with remote GUI

- Teensy communicates with remote GUI over Bluetooth
  - ESP32 acts as Bluetooth to SPI relay
  - SPP provides reliable byte stream

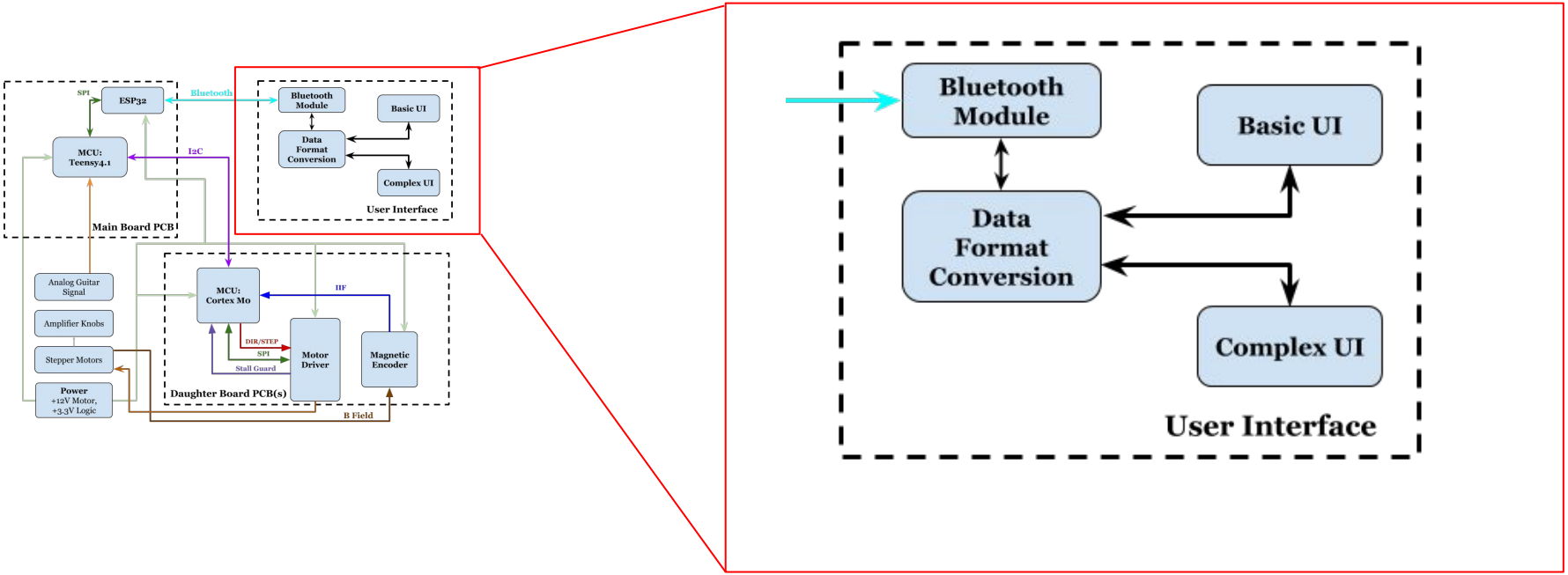
Communication with daughter boards

- Teensy communicates with daughter boards over I2C
  - Daughter boards are daisy-chained
  - Teensy dictates target knob positions to daughter board
  - Daughter board alerts Teensy if user grabs knob

Demo: I2C Communication



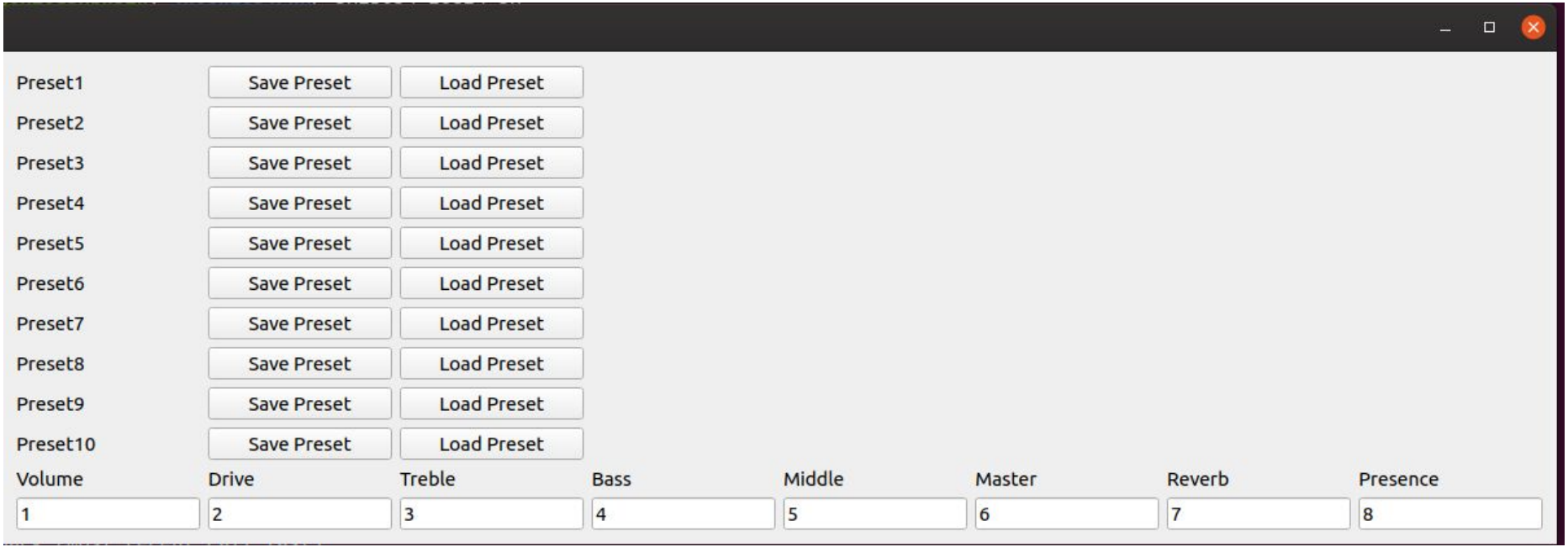
Part 3: User Interface and Data Formats



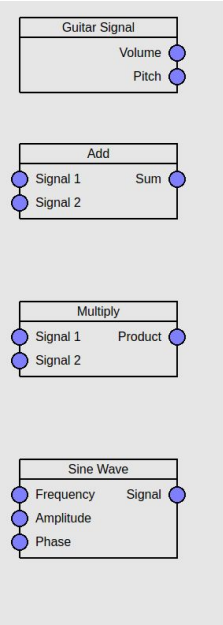
Main Challenges

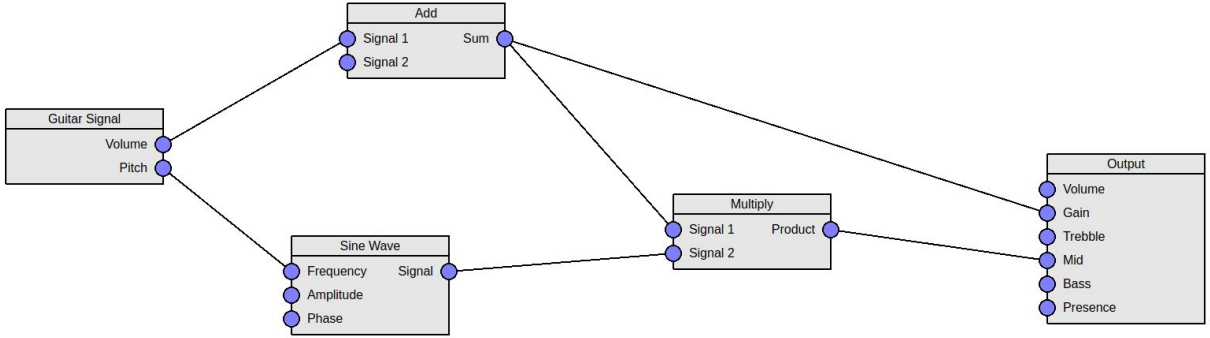
- User interface must be powerful but not confusing
  - Basic functionality is simple to access
  - Advanced functionality is easy to understand
- Complex information must be exchanged with main board
  - Need to send presets, node graphs both ways
  - Different languages, different processor architectures

Basic UI


Preset1 Save Preset Load Preset
Preset2 Save Preset Load Preset
Preset3 Save Preset Load Preset
Preset4 Save Preset Load Preset
Preset5 Save Preset Load Preset
Preset6 Save Preset Load Preset
Preset7 Save Preset Load Preset
Preset8 Save Preset Load Preset
Preset9 Save Preset Load Preset
Preset10 Save Preset Load Preset
Volume Drive Treble Bass Middle Master Reverb Presence
1 2 3 4 5 6 7 8

Node Graph Editor





Communication with Main Board

- Bluetooth SPP
  - Looks like a full duplex reliable byte stream
  - Achieved using PyBluez library
- Messages serialized using Flatbuffers
  - Object serialization protocol
  - Good middle ground between raw binary and ASCII formats

MDR Accomplishments

- Completed version 1 PCBA of daughter board and ordered main board PCB
- Assembled first iteration of mechanical design
- Knob Control Unit: Open Loop functionality
- Rudimentary “simple” UI design
- Initial Implementation of “advanced” GUI version

Hardware Plan for FPR

- Daughterboard Revision
  - Currently pending 7 changes as of 11/16/2020
  - Improved footprints for main IC's
  - Investigate optional programming through USB-C
- Motherboard Revision
  - Higher quality PCB fabrication
  - Remove testbench features
  - Remove breakout points
- Finalize Mechanical Design
  - Connect timing belts to pulleys
  - Create enclosure

Team Member Contributions

Chris:

- PCB Design, ElectroMechanical Design

Remy:

- PCB Design, PCB Routing, Prototype Hardware

Casey:

- Advanced UI Design, Data Format Creation, Teensy Programming

Aaron:

- Open Loop Stepper Control, I2C Communication, Basic UI Design

Budgeting

Current Expenditures

Category	Price
MCUs	$60.13
ICS	$64.79
Mechanical Components	$29.37
Daughter Board BOM	$16.20
Main Board BOM	$39.68
PCB Orders	$33.75
Total	$243.92

Planned Expenditures

Item	Quantity	Price	Totals
MB PCB Revisions	1	60	60
DB PCB Revisions	2	50	100
DB BOM	6	16	96
MB BOM	1	40	40
Rest of M0s	8	1.74	13.92
Stepper Motors	11	8	88
		Total:	397.92

https://docs.google.com/spreadsheets/d/1mysqH0vYbxG8X77n8beeuHevb0FEGBNMz5iD0I4eBcE/edit?usp=sharing

Unit Pricing

- Unit Price to Manufacture 1 Completely Functional Unit (1 unit price / 1000 order)

Item	Quantity	Price	Totals	1000 Units Total
DaughterBoard	8	$19.19	$153.52	$153,520
MotherBoard	1	$60.44	$60.44	$60,440
Enclosure	1	$100	$100	$100,000
		Total:	$313.96	$313,960

List of Hardware and Software

- Main Board(Custom PCB)
  - Teensy 4.1(C++)
  - ESP 32(C++)
- Daughterboards(Custom PCB)
  - Cortex M0(C)
  - Trinamic Driver(I2C)
  - Magnetic Encoder(IIF)
- GUI
  - Python(PyQt)
  - Google Flat-Buffers(Data Formats)