Team D5
Eli Wirth-Apley, Sachit Goyal, Vedant Bhasin

SYMPHONYAI

Listening Experience

Use Case

Our product will allow users to experience an in house orchestra of AI generated music produced by a state-of-the-art generative neural network. Users will also be able to customize their listening experience by applying EQ and other effects individually to each generated instrument with a custom remote control. Our movable, distributed speaker system will immerse users in rich music experience of an orchestra hall. Whether you're an artist trying to find inspiration for new compositions, or just a passionate music listener looking to experience AI generated music, our product has something for you.

ECE areas: Signals, Hardware

https://teenage.engineering/products/choir

Use Case Requirements

Requirement	Goal
Size and weight of Controller:	8x4x2in, 250g
Controller Inputs:	3 Potentiometers and 3 function buttons
Remote response to user input:	~150ms
ML inference for generating music:	~100ms
Distributed networking to speakers:	1 Mbps Baud Rate
Battery Life: Remote	10 hours
Battery Life: Speaker	4 hours
% empty bars	30.00*

Benchmarked from Deng et al. MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment, with a 25% margin on piano performance

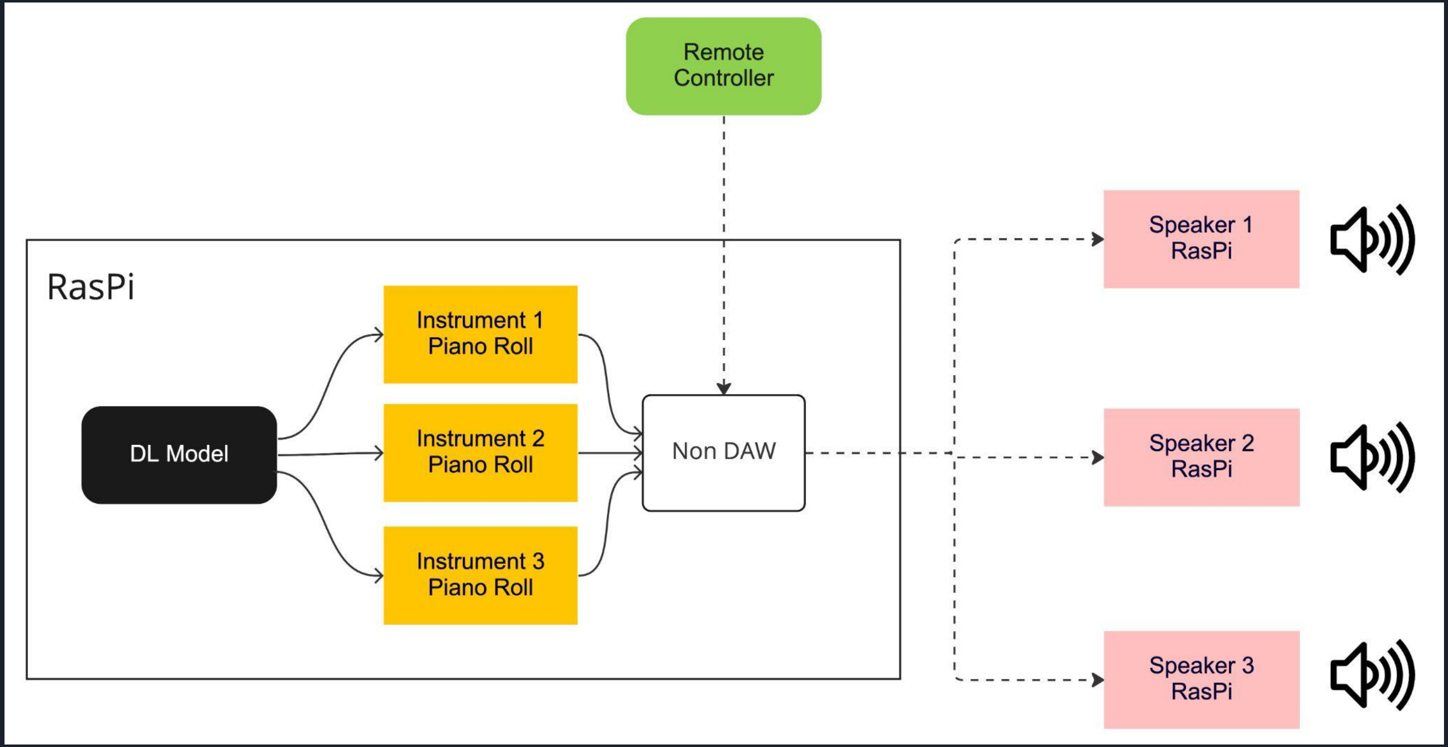
Qualitative Use Case Requirements

Requirement	Goal
Genre of music	Three distinct genres
Pleasing sound experience	User score of 4/5, above average
Easy of use: remote control	User score of 4/5, above average
Easy of use: distributed speakers	User score of 5/5, very easy
Similarity to human generated music	User score of 4/5, above average

Technical Challenges

- Generating music with multiple instruments is a complex design challenge which requires intricate networks to model dependencies across different instruments as well as previous melodies of the same instrument.
- Even with certain objective metrics to measure the quality of the generated music, it has hard to quantify if the model is producing good or bad music.
- Deep learning models can be memory intensive, requiring 100’s of MB to store the learned parameters.
- Achieving 1Mbps baud rate to distribute music files between Raspberry Pis using nRF24L01 transceiver modules.
- Designing a PCB which affords an easy to use controller interface.
- Seamless experience when applying effects to generated music and changing genre given multiple latencies of hardware, communication between chips.
- Compact, durable remote with multiple chips, power supplies will be tricky and will require careful design choices

Solution Approach



Solution Approach

- ML: The main advantage that a GAN has is that it learns by trying to make it's generations indistinguishable from a trained discriminator, thus to the human ear it should be indistinguishable from human made music.
- ML: The model will be trained on data from the Lakh Pianoroll Dataset (LPD), There will be a different model trained for each genre, to ensure coherence in the music.
- ML: The generator will be approx 300,00 parameters, which is significantly lower than what an attention based approach would have.
- Distributed: Using RasPi and nRF24L01 transceiver modules.
- Controller: Using RasPi for inference, Arduino for inputs housed in a custom PCB.

Testing, Verification and Metrics

- Each individual component i.e. ML model, remote control and distributed communication will be tested individually to meet requirements and then integrated.
- ML: The performance of the model will be evaluated based on Tonal Distance between the instruments, % empty bars and the user survey.
- Distributed communication: unit tested with different protocols and both latency and throughput will be measured. As music files are usually 3-4 Mb, testing will be based on sending large files infrequently.
- Test breadboarded hardware design with oscilloscope over full range of inputs to ensure signal propagates through potentiometers to arduino

=

Tasks and Division of Labor

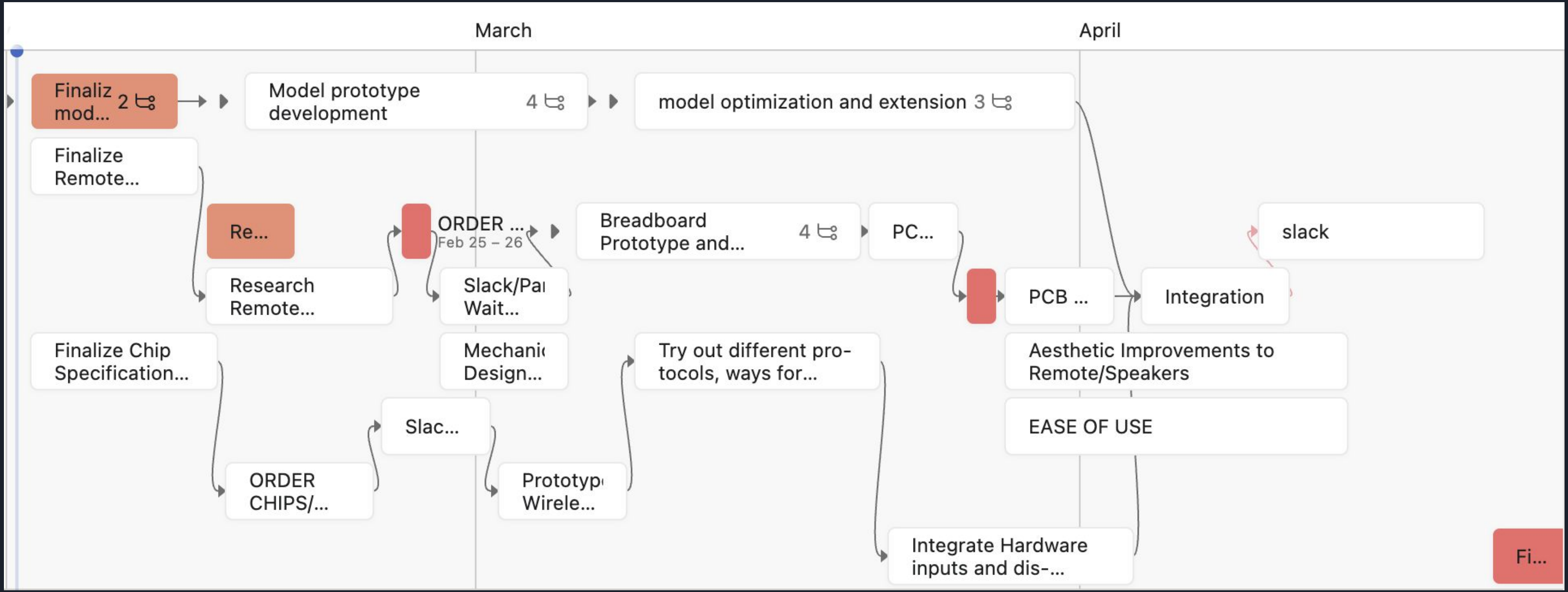
Remote control design: Eli

Distributed networking: Sachit

ML Model: Vedant

Communication between hardware and Arduino/RasPi: Sachit and Eli

Schedule



March
April
Finaliz mod... 2
Model prototype development 4
model optimization and extension 3
Finalize Remote...
Re...
ORDER ...
Feb 25 – 26
Breadboard Prototype and... 4
PC...
slack
Research Remote...
Slack/Pa
Wait...
PCB ...
Integration
Finalize Chip Specification...
Mechani
Design...
Try out different pro-tocols, ways for...
Aesthetic Improvements to Remote/Speakers
Slac...
EASE OF USE
ORDER CHIPS/...
Prototyp
Wirele...
Integrate Hardware inputs and dis-...
Fi...

Questions?