The BopBot

TEAM 21

Meet the team

Matthew Cierpial
CSE

Max Jaffe
EE

Austin Reilly
CSE

Vee Upatising
CSE

Advisor: Prof. Moritz

Problem Statement

The most memorable songs often feature a compelling (vocal) melody. Musicians often struggle to write melodies that are catchy and unique. The BopBot can inspire this creative process by offering melodies based on what you play into the machine. After defining timing parameters, the user will be prompted to play a chord progression. The BopBot will then generate a cohesive and interesting sequence of notes using its recurrent neural network. The user can then playback the chord progression and melody together to inspire different musical motifs to use in their own melodies.

NEW System Specifications

1. Can fit onto a musician's pedal board. (Typically 170x138mm)
2. Completes melody generation in an amount of time that is conducive for live music. Less than 5 seconds is acceptable.
3. 4 musical genres to choose from for models: Rock, Blues, Classical, Pop.
4. Powered by a typical music pedal power supply (9V, 1700mA max).
5. Neural network design must have less than 23,000 neurons in each of the 3 layers (fewer if using 4 layers) in order to generate a melody within 5 seconds on a 1 GHz processor.
    - The design of the neural network is based on successful existing music-generation RNN's: https://pdfs.semanticscholar.org/c933/79a401dd159fc0c90eab44c43d07286b227e.pdf
5. Each neural network must be smaller than 200 MB such that the microprocessor has enough data memory (SDRAM).

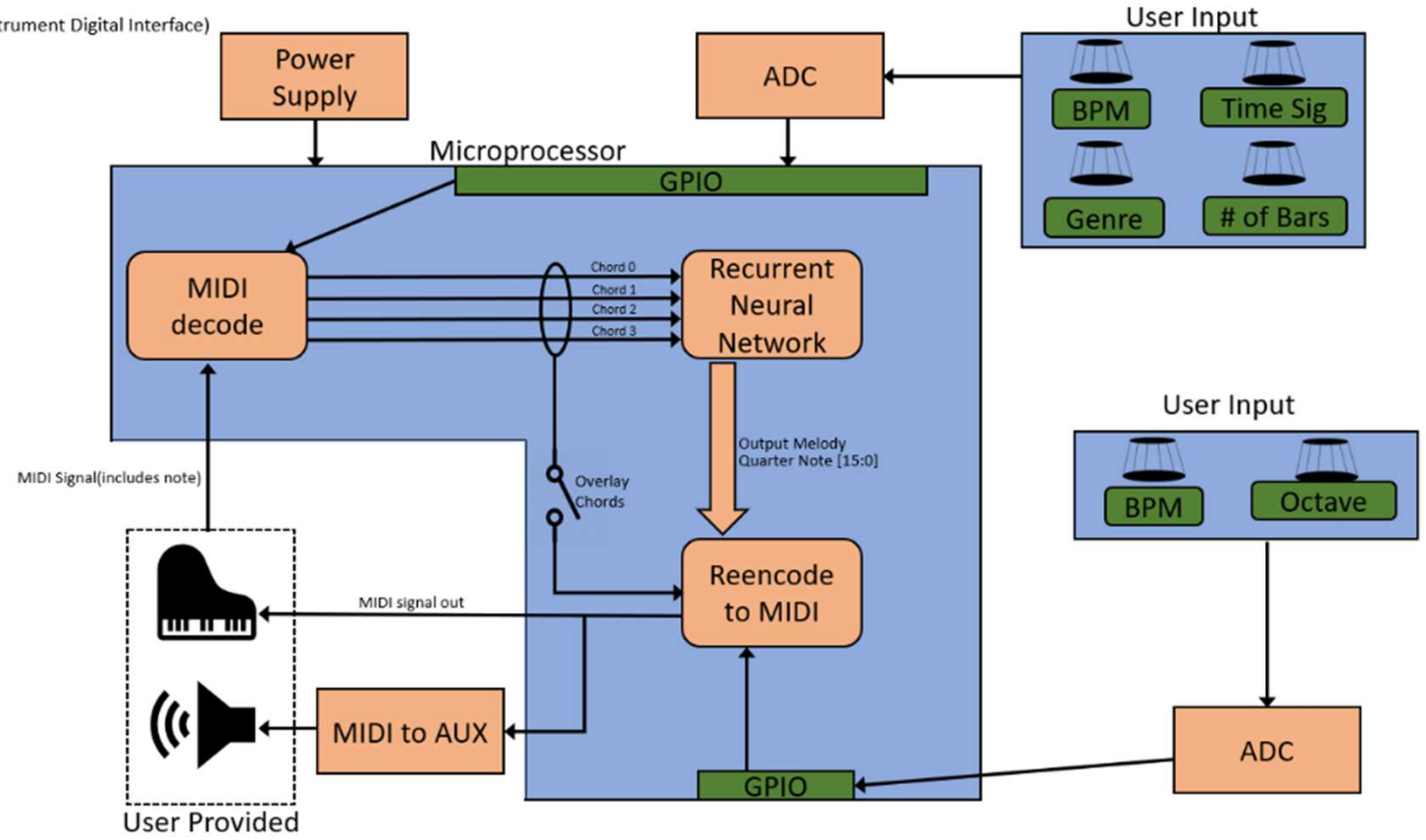
Team roles

- Vee: Supervised machine learning network design, testing, and implementation
- Matt: Encode output from machine learning network as a playable MIDI/Analog signal and implement playback control based on user input. (BPM/Octave)
- Austin: Decoding MIDI signal input into a form that can be used by the machine learning network.
- Max (Proj. Manager): Supervised machine learning network design, testing, and implementation. PCB layout and assembly.

Everyone will be working on data collection for model training

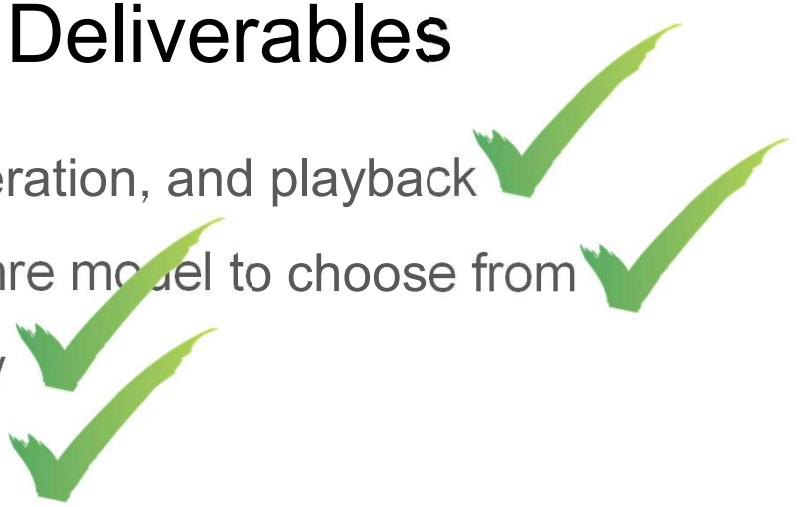
NEW Block diagram

-Each chord is represented by 8 bits.
-Each Quarter Note is also represented by 8 bits.
-MIDI(Musical Instrument Digital Interface)



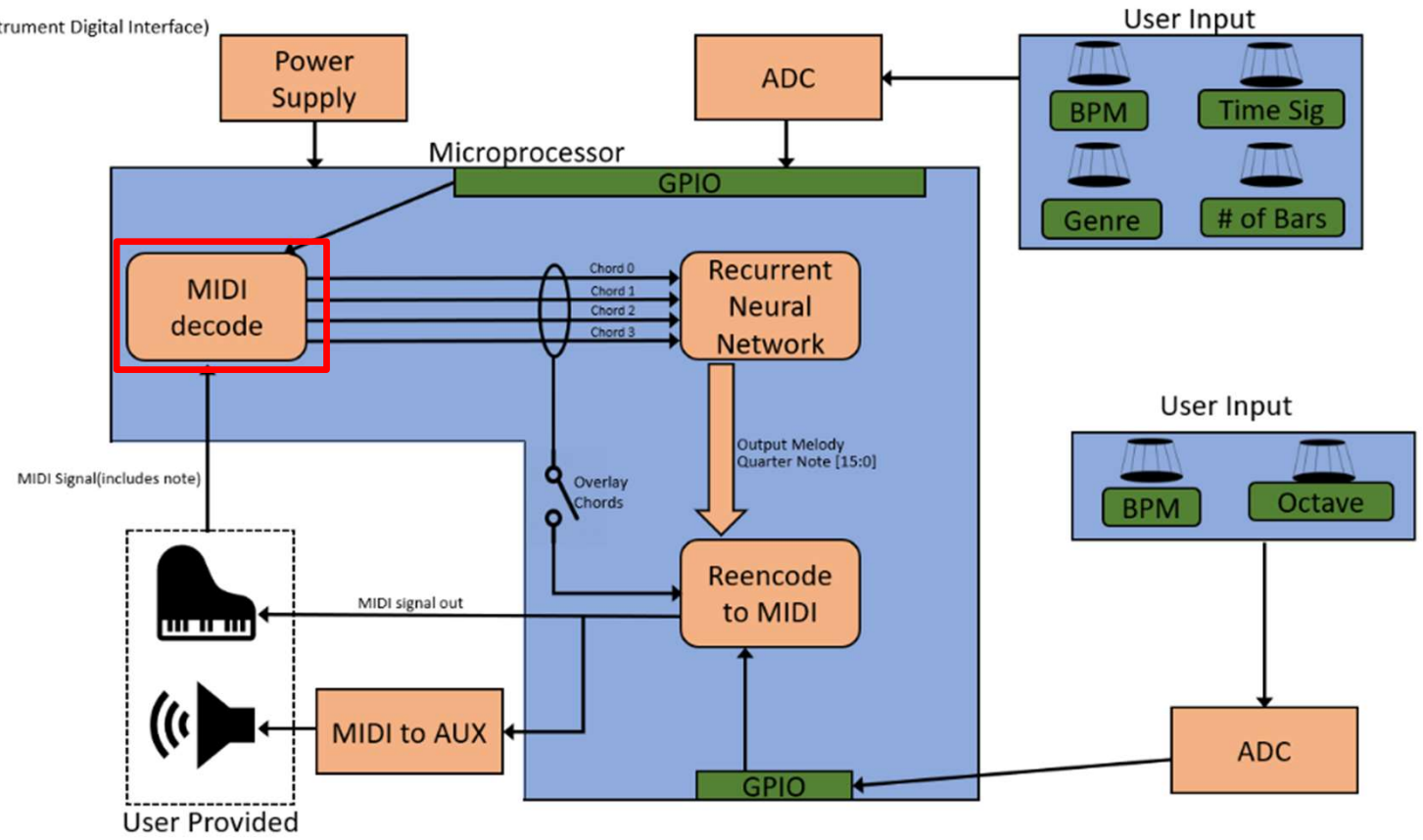
Our proposed MDR Deliverables

- Live recording, melody generation, and playback
- More than one available genre model to choose from
- Input will be in a specific key
- Playback with correct timing
- All done using the De1-SoC
  - Solution: Raspberry Pi Model 3 B

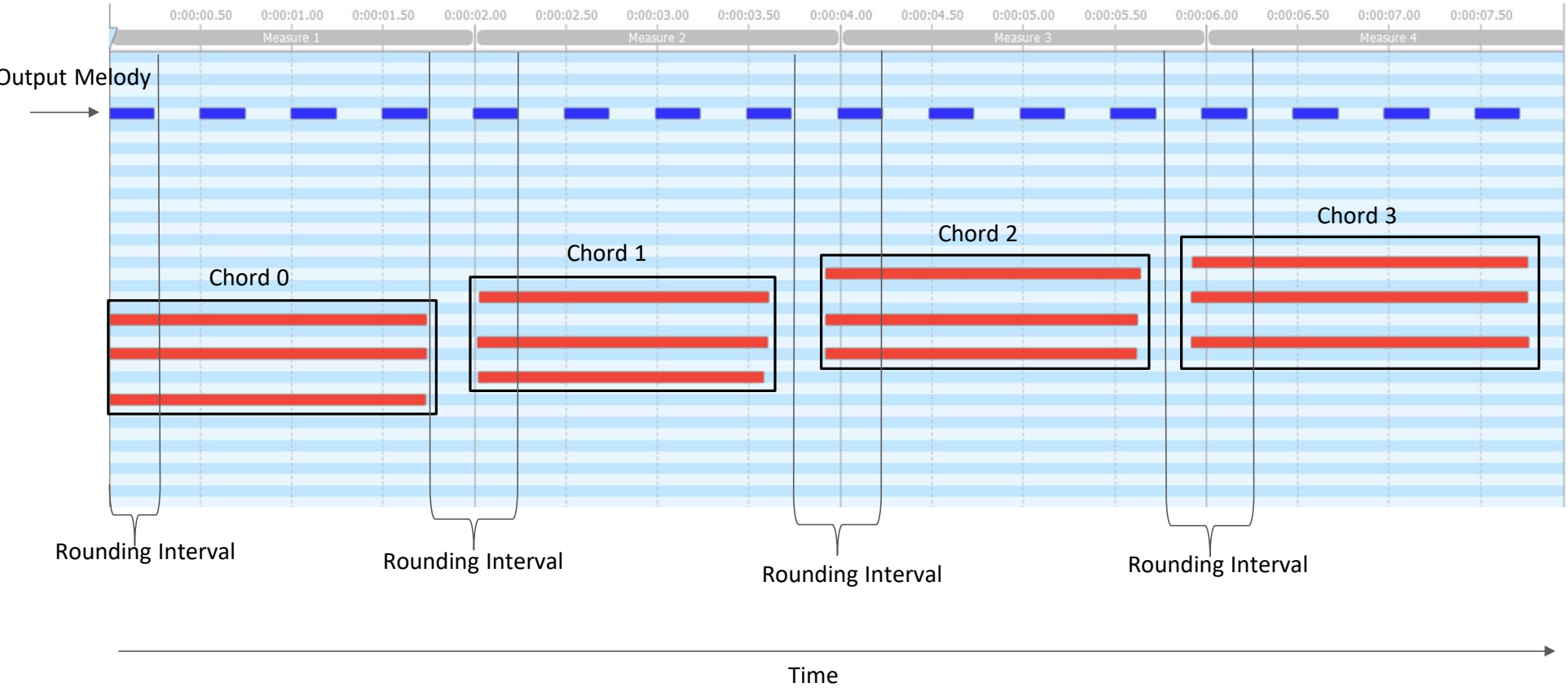


NEW Block diagram

-Each chord is represented by 8 bits.
-Each Quarter Note is also represented by 8 bits.
-MIDI(Musical Instrument Digital Interface)

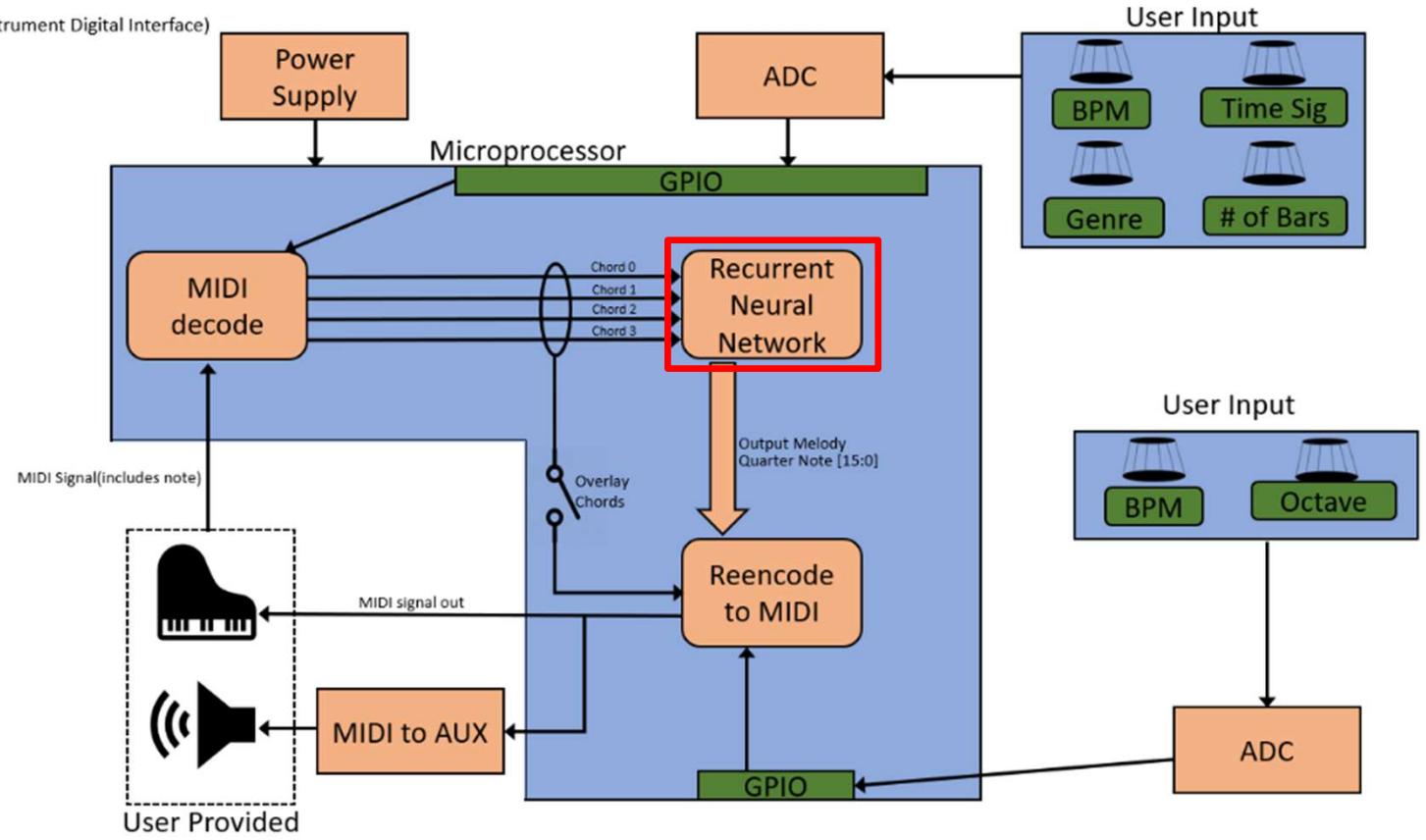


MIDI Decode - I/O Visualization

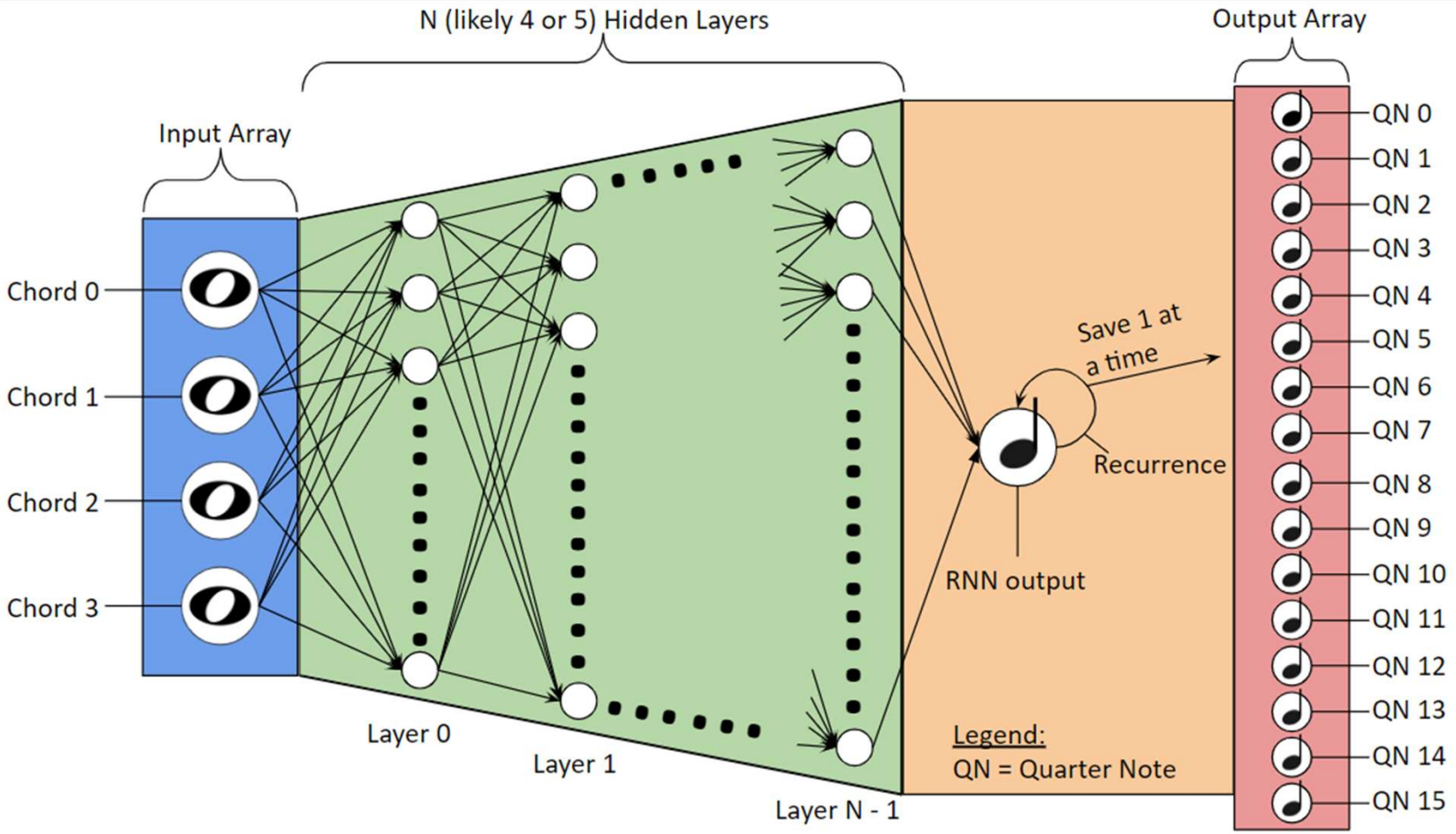


NEW Block diagram

-Each chord is represented by 8 bits.
-Each Quarter Note is also represented by 8 bits.
-MIDI(Musical Instrument Digital Interface)



RNN Block diagram



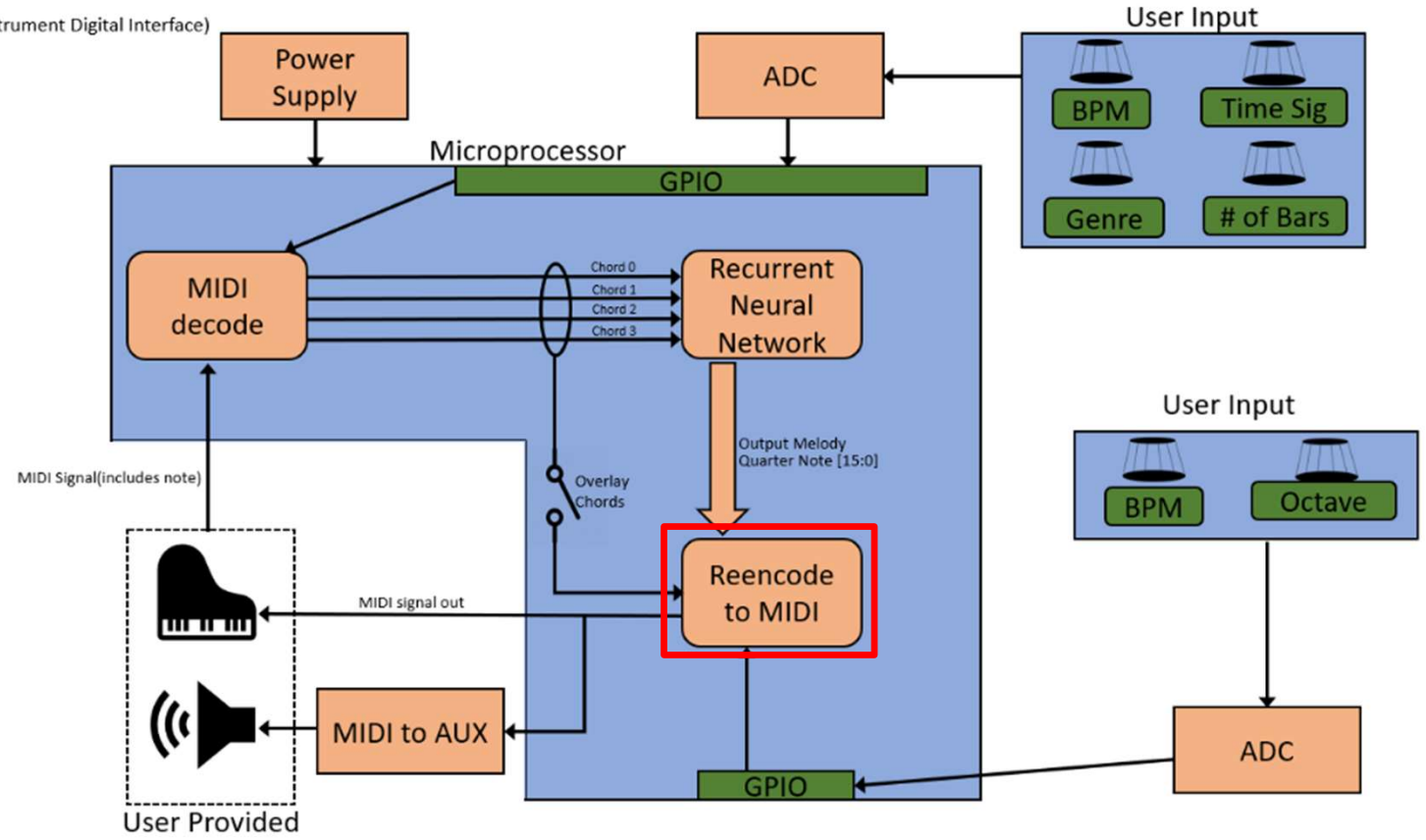
Neural Network Design Choices

- Recurrent Neural Network:
  - The sequential ordering of the notes matters
  - The value of each note is directly dependent on the values of the notes played before it
- Neural Network Structure (subject to change):
  - We chose 3 hidden layers with 128 nodes at each layer
  - This was a good optimization between performance vs. cost
  - After research, it was found that there are diminishing returns beyond 4 hidden LSTM layers

Design choices inspired from previous research: https://pdfs.semanticscholar.org/c933/79a401dd159fc0c90eab44c43d07286b227e.pdf

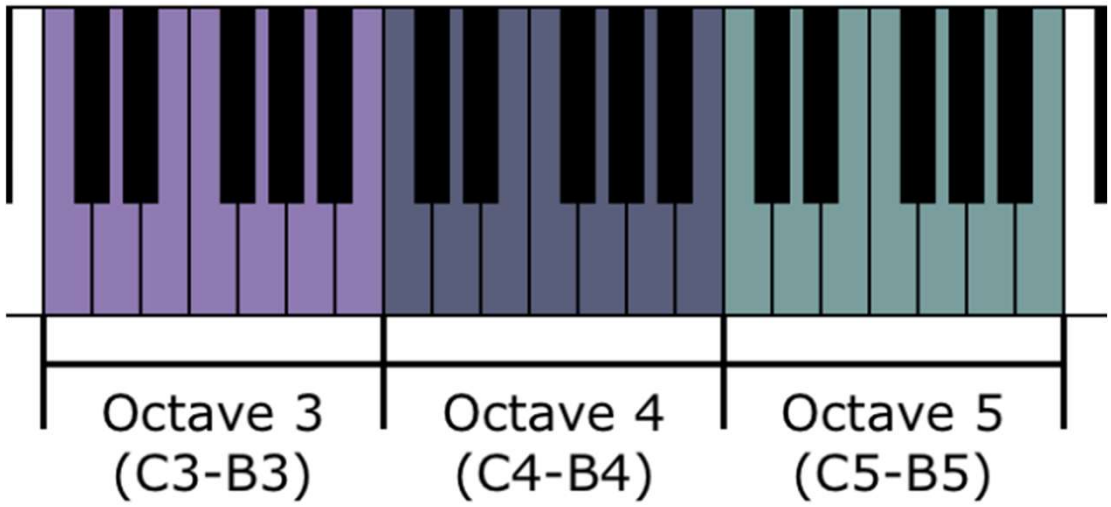
NEW Block diagram

-Each chord is represented by 8 bits.
-Each Quarter Note is also represented by 8 bits.
-MIDI(Musical Instrument Digital Interface)



Reencode to MIDI

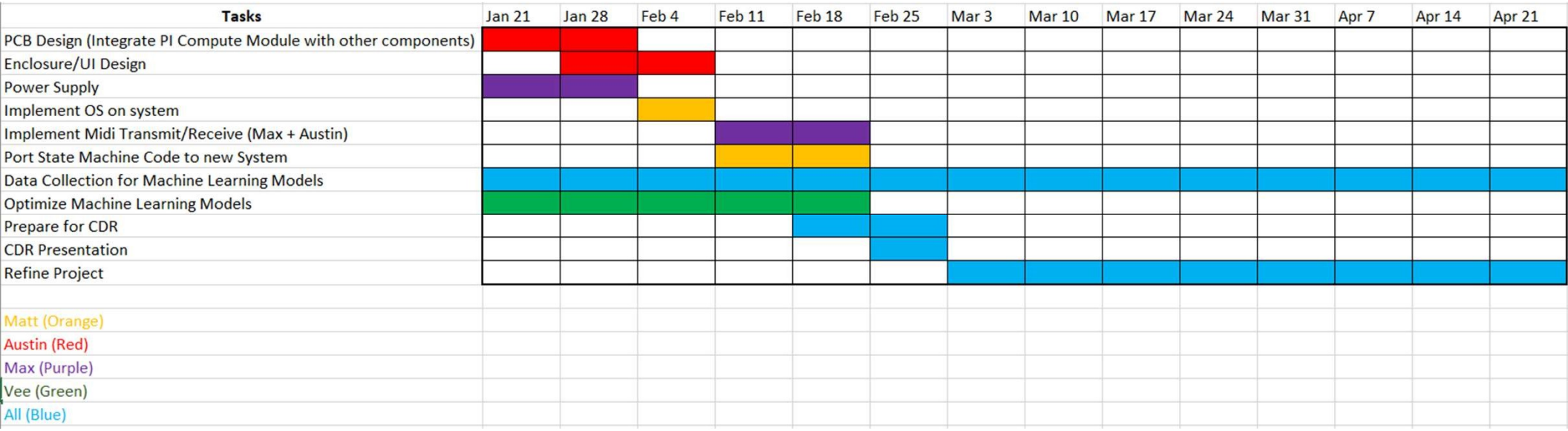
- Post-RNN processing
- Overlaying chords and melody



CDR Deliverables

- Model improvements
  - DATA DATA DATA!
  - Trial and error
  - An analysis of model improvements
- Raspberry Pi Compute Module M3+
- PCB Design

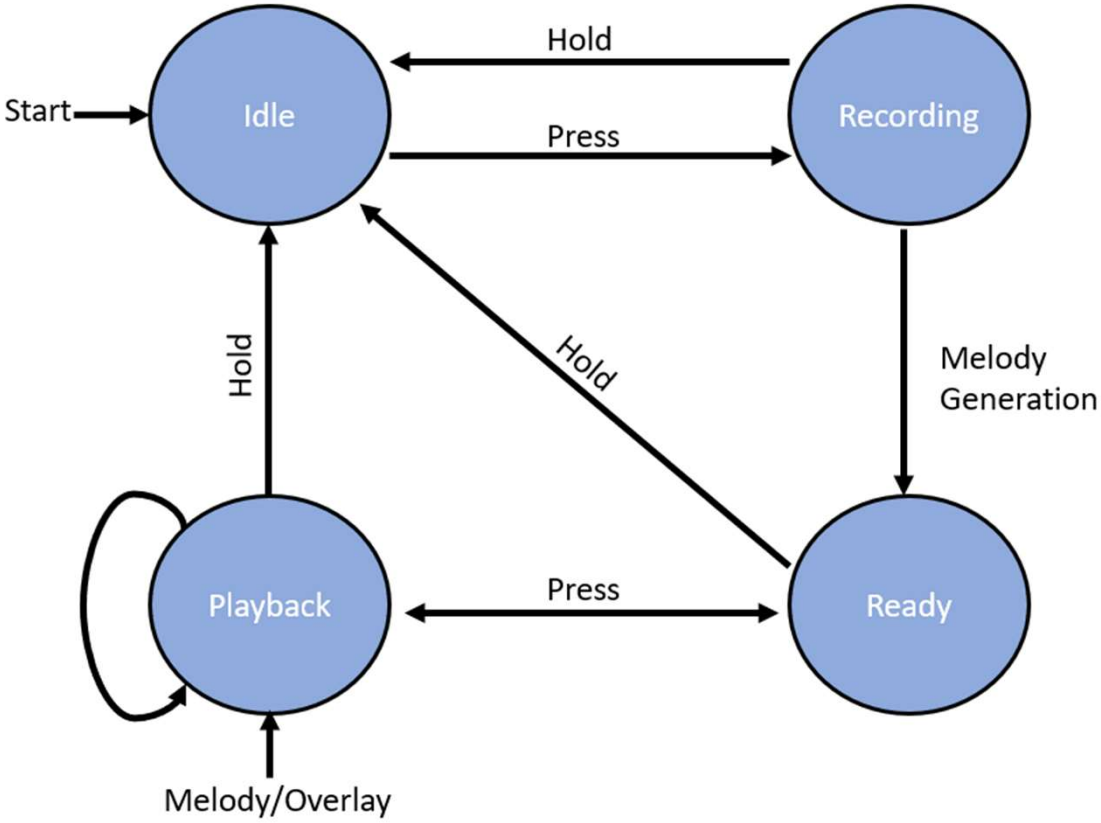
Gantt Chart



Tasks	Jan 21	Jan 28	Feb 4	Feb 11	Feb 18	Feb 25	Mar 3	Mar 10	Mar 17	Mar 24	Mar 31	Apr 7	Apr 14	Apr 21
PCB Design (Integrate PI Compute Module with other components)														
Enclosure/UI Design														
Power Supply														
Implement OS on system														
Implement Midi Transmit/Receive (Max + Austin)														
Port State Machine Code to new System														
Data Collection for Machine Learning Models														
Optimize Machine Learning Models														
Prepare for CDR														
CDR Presentation														
Refine Project														

Matt (Orange)
Austin (Red)
Max (Purple)
Vee (Green)
All (Blue)

Demo


Start
Idle
Hold
Recording
Press
Hold
Hold
Melody
Generation
Playback
Press
Ready
Melody/Overlay

Questions?