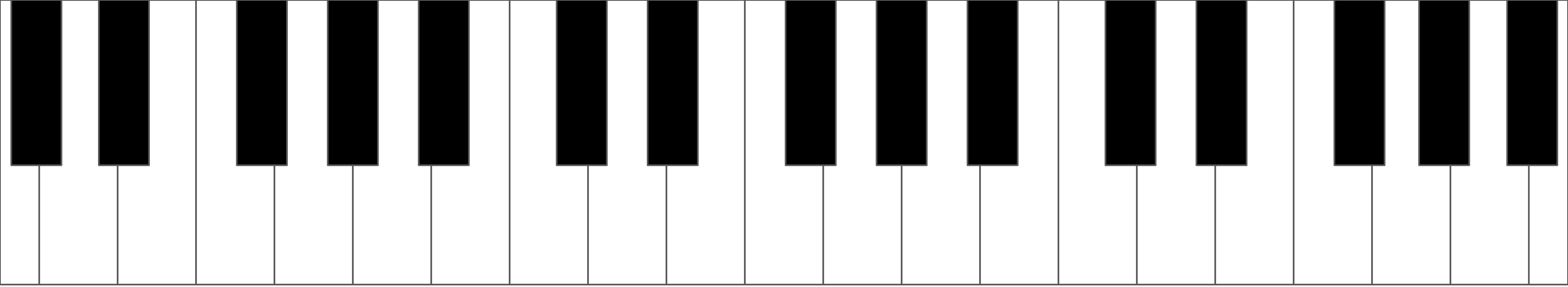


ac·com·pa·ny·Bot

/əˈkəmp(ə)nē bät/ *noun*

A piano playing robot that scans and parses sheet music and reproduces the notes by pressing keys on a piano

Made by Aden Fiol, Rahul Khandelwal, and Nora Wan

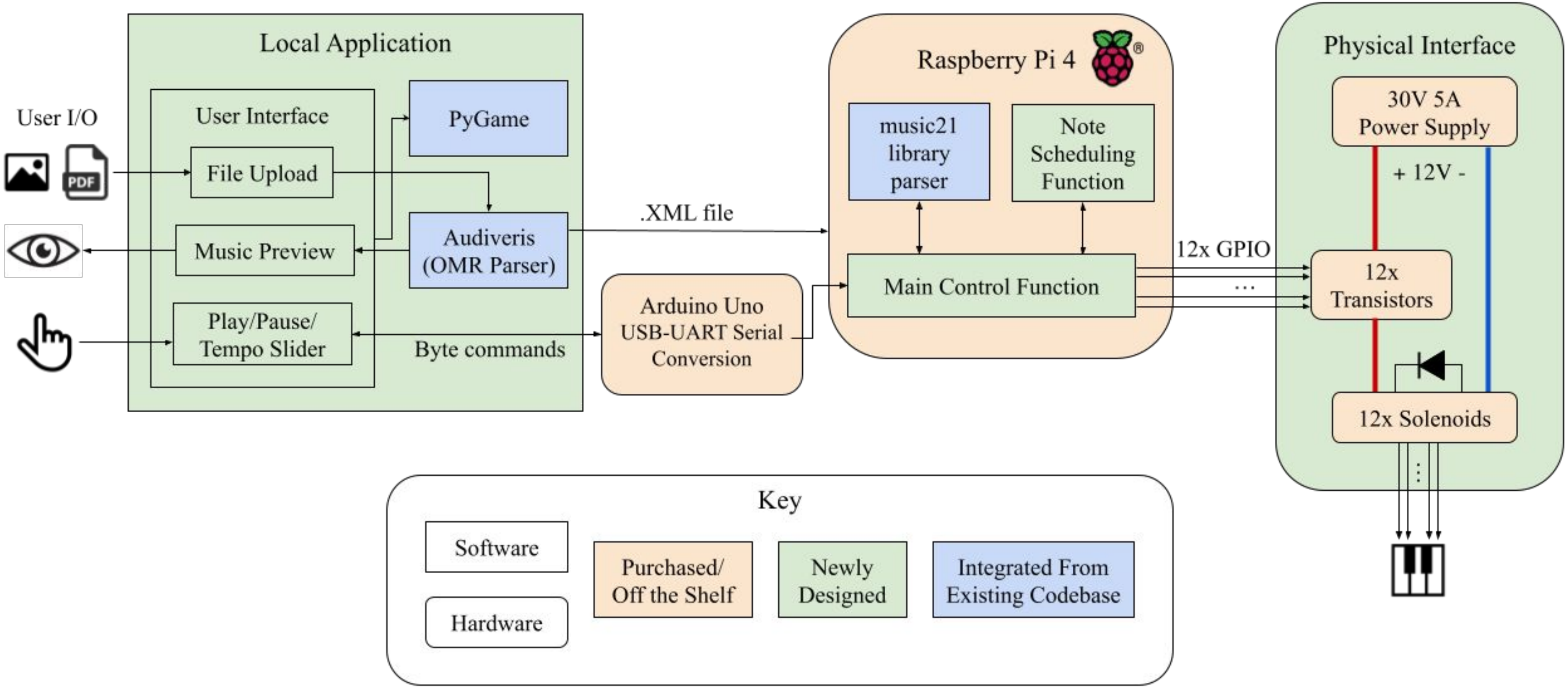
Use-Case Requirements

- **Note Playing Accuracy:** System should not make noticeable mistakes
- **Tempo Variability:** Ability to speed up/slow down tempo from application
- **Tempo Accuracy:** Ability to play music with the exact BPM specified
- **Low latency between UI and piano player:** Have piano player respond to user inputs within the average human response time to auditory stimuli

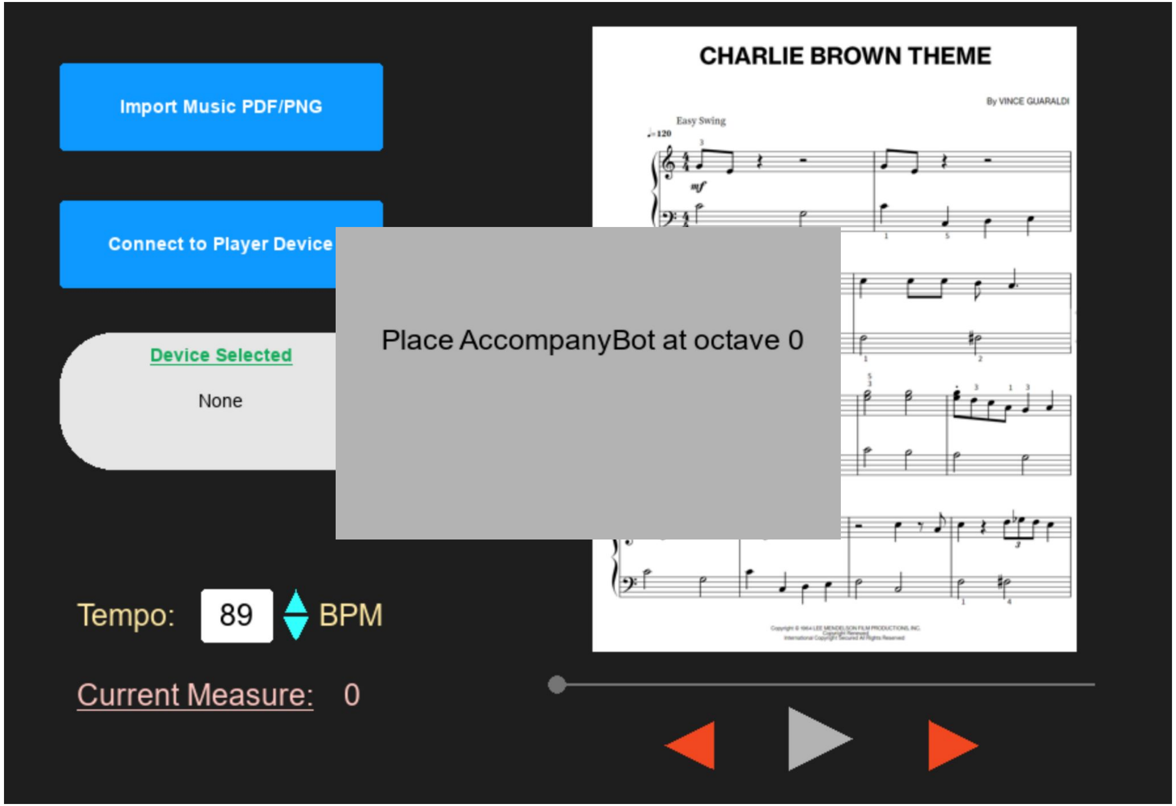
Quantitative Design Requirements

- >95% OMR Parser Accuracy
- 100% tempo accuracy
- <150ms latency between user action and piano player response
- <60W average power

Solution Approach



Complete Solution - Software



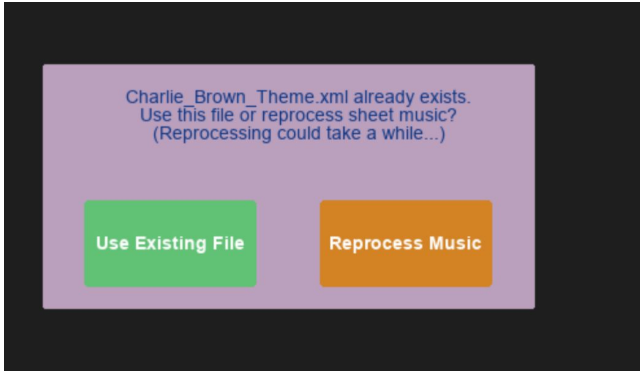
Place AccompanyBot at octave 0

Device Selected

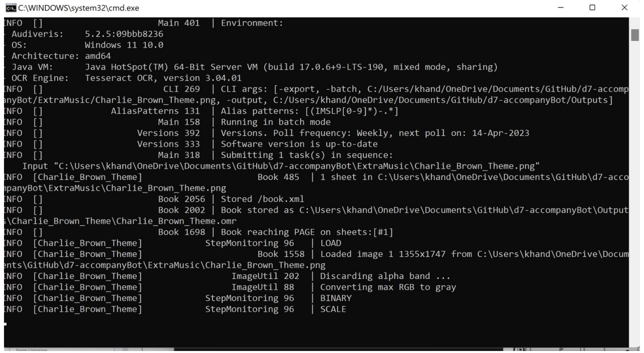
None

Tempo: 89 BPM

Current Measure: 0

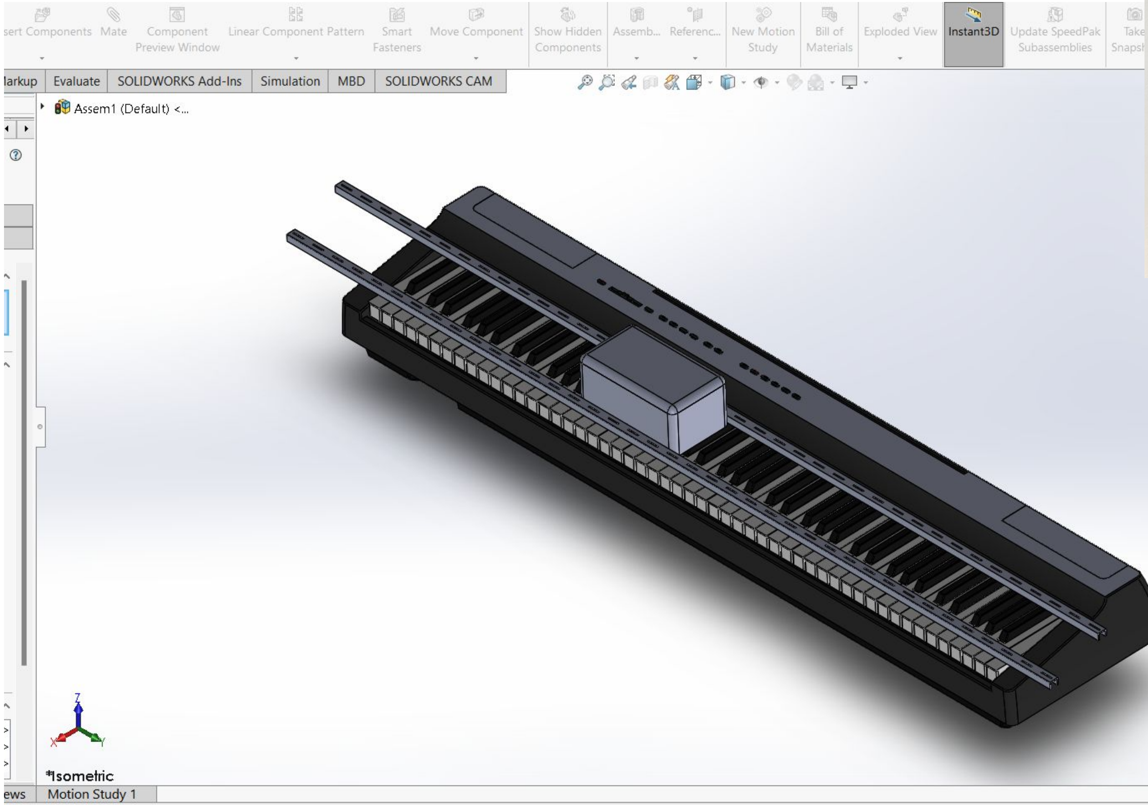


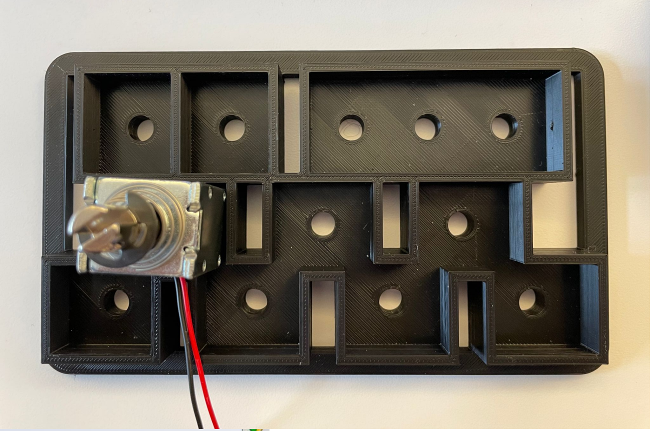
Charlie_Brown_Theme.xml already exists.
Use this file or reprocess sheet music?
(Reprocessing could take a while...)



```
INFO  []                        Main 401  | Environment:
- Audiveris:    5.2.5:09bbb8236
- OS:           Windows 11 10.0
- Architecture: amd64
- Java VM:      Java HotSpot(TM) 64-Bit Server VM (build 17.0.6+9-LTS-190, mixed mode, sharing)
- OCR Engine:   Tesseract OCR, version 3.04.01
INFO  []                         CLI 269  | CLI args: [-export, -batch, C:/Users/khand/OneDrive/Documents/GitHub/d7-accomp
anyBot/ExtraMusic/Charlie_Brown_Theme.png, -output, C:/Users/khand/OneDrive/Documents/GitHub/d7-accompanyBot/Outputs]
INFO  []               AliasPatterns 131  | Alias patterns: [(IMSLP[0-9]*)-.*]
INFO  []                        Main 158  | Running in batch mode
INFO  []                    Versions 392  | Versions. Poll frequency: Weekly, next poll on: 14-Apr-2023
INFO  []                    Versions 333  | Software version is up-to-date
INFO  []                        Main 318  | Submitting 1 task(s) in sequence:
   Input "C:\Users\khand\OneDrive\Documents\GitHub\d7-accompanyBot\ExtraMusic\Charlie_Brown_Theme.png"
INFO  [Charlie_Brown_Theme]                        Book 485  | 1 sheet in C:\Users\khand\OneDrive\Documents\GitHub\d7-acco
mpanyBot\ExtraMusic\Charlie_Brown_Theme.png
INFO  []                        Book 2056 | Stored /book.xml
INFO  []                        Book 2002 | Book stored as C:\Users\khand\OneDrive\Documents\GitHub\d7-accompanyBot\Output
s\Charlie_Brown_Theme\Charlie_Brown_Theme.omr
INFO  []                        Book 1698 | Book reaching PAGE on sheets:[#1]
INFO  [Charlie_Brown_Theme]           StepMonitoring 96   | LOAD
INFO  [Charlie_Brown_Theme]                        Book 1558 | Loaded image 1 1355x1747 from C:\Users\khand\OneDrive\Docum
ents\GitHub\d7-accompanyBot\ExtraMusic\Charlie_Brown_Theme.png
INFO  [Charlie_Brown_Theme]                   ImageUtil 202  | Discarding alpha band ...
INFO  [Charlie_Brown_Theme]                   ImageUtil 88   | Converting max RGB to gray
INFO  [Charlie_Brown_Theme]           StepMonitoring 96   | BINARY
INFO  [Charlie_Brown_Theme]           StepMonitoring 96   | SCALE
```

Complete Solution - Hardware



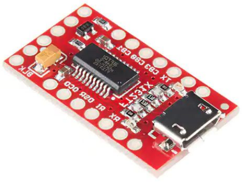


Design Limitations

- Uploaded files must be high quality
  - OMR parser functions best with black/white scores (compared to grayscale files)
  - Accuracy guaranteed when input scores have resolution above 400 DPI
- Due to scope of project, only plays one octave
  - During note scheduling, determines the octave with highest note count
  - Communicates octave number back to application
  - User must move the accompanyBot over the correct octave
- Only covers C-to-C octave range
  - Black key spacing is irregular
  - Prevents the design of a chassis that can fit over any octave of keys
- Physical solenoids cannot play faster than movement threshold
  - During note scheduling, determines max tempo playable
  - Communicates the max tempo supported back to the application

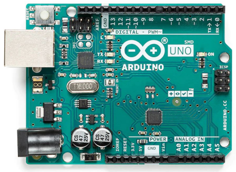
Design Tradeoffs

- Limited range of keys to one octave black and white keys
  - Save on costs and power consumption
- Chose to use 25N solenoids
  - Larger and more expensive, but generated more force and has a longer stroke length than alternative
- Used an Arduino Uno to aid in serial communication
  - Needed a way to convert between USB and UART
- Wireless instead of wired file transfer
  - scp to the RPi
  - Serial transmission of one page >13 seconds



vs.

vs.



Test, Verification and Validation

- **OMR Accuracy** → Reconvert back to pdf and compare visually
- **Tempo accuracy** → Measure tempo with a metronome
- **Computer/RPi Latency** → Function records system time when start/stop signals are sent and received
- **Power Consumption** → Measure DC power supply output voltage and current to calculate power

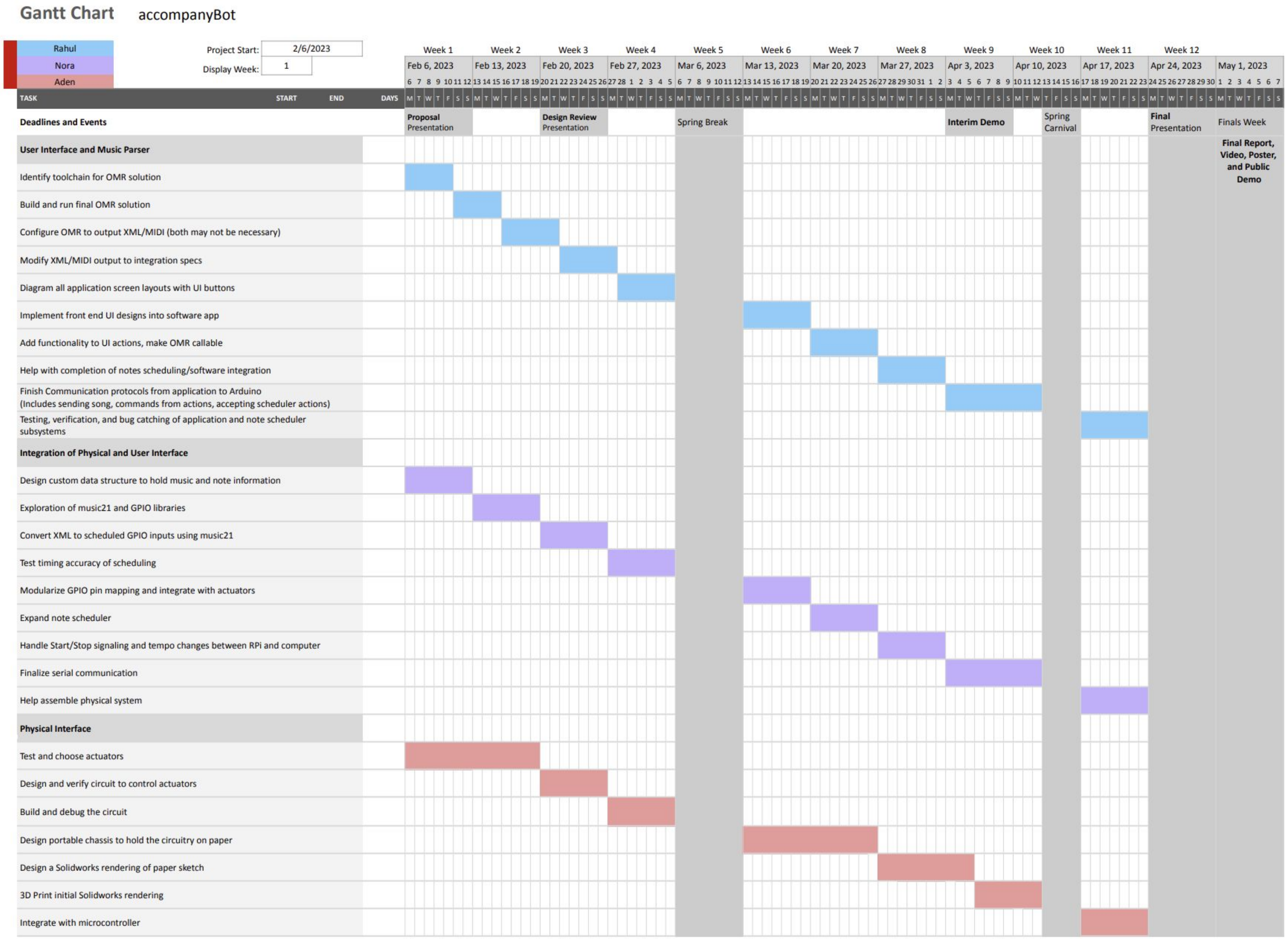
Specifications and Performance

Design Requirement Metric	System Performance
> 95% OMR accuracy of note pitches and note values for 400+ DPI music scores	~99% accuracy for easy to moderate playing difficulty scores of 240 DPI ✓
100% tempo accuracy	100% accurate between 30BPM-150BPM ✓
Start/stop playing within 150 ms of pressing the start/stop button	Round trip communication latency of ~70ms ✓
< 60W average power	Max power of 46.8W ✓

Specifications and Performance

Use Case Requirement Metric	System Performance
Note Playing Accuracy	**In Progress** Once full system is constructed, compare audio played by accompanyBot to MIDI output
Tempo Variability	**Pass** Able to update tempo via UI
Tempo Accuracy	**Pass** Solenoids keep pace with metronome
Latency	**In Progress** Determine whether noticeable delay exists between user actions on app and keyboard audio

Schedule



Gantt Chart accompanyBot

Rahul
Nora
Aden

Project Start: 2/6/2023
Display Week: 1

Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	
Feb 6, 2023	Feb 13, 2023	Feb 20, 2023	Feb 27, 2023	Mar 6, 2023	Mar 13, 2023	Mar 20, 2023	Mar 27, 2023	Apr 3, 2023	Apr 10, 2023	Apr 17, 2023	Apr 24, 2023	May 1, 2023
6 7 8 9 10 11 12	13 14 15 16 17 18 19	20 21 22 23 24 25 26	27 28 1 2 3 4 5	6 7 8 9 10 11 12	13 14 15 16 17 18 19	20 21 22 23 24 25 26	27 28 29 30 31 1 2	3 4 5 6 7 8 9	10 11 12 13 14 15 16	17 18 19 20 21 22 23	24 25 26 27 28 29 30	1 2 3 4 5 6 7
M T W T F S S	M T W T F S S	M T W T F S S	M T W T F S S	M T W T F S S	M T W T F S S	M T W T F S S	M T W T F S S	M T W T F S S	M T W T F S S	M T W T F S S	M T W T F S S	M T W T F S S

TASK START END DAYS

Deadlines and Events: **Proposal** Presentation; **Design Review** Presentation; Spring Break; **Interim Demo**; Spring Carnival; **Final** Presentation; Finals Week

Final Report, Video, Poster, and Public Demo

User Interface and Music Parser

- Identify toolchain for OMR solution
- Build and run final OMR solution
- Configure OMR to output XML/MIDI (both may not be necessary)
- Modify XML/MIDI output to integration specs
- Diagram all application screen layouts with UI buttons
- Implement front end UI designs into software app
- Add functionality to UI actions, make OMR callable
- Help with completion of notes scheduling/software integration
- Finish Communication protocols from application to Arduino (Includes sending song, commands from actions, accepting scheduler actions)
- Testing, verification, and bug catching of application and note scheduler subsystems

Integration of Physical and User Interface

- Design custom data structure to hold music and note information
- Exploration of music21 and GPIO libraries
- Convert XML to scheduled GPIO inputs using music21
- Test timing accuracy of scheduling
- Modularize GPIO pin mapping and integrate with actuators
- Expand note scheduler
- Handle Start/Stop signaling and tempo changes between RPi and computer
- Finalize serial communication
- Help assemble physical system

Physical Interface

- Test and choose actuators
- Design and verify circuit to control actuators
- Build and debug the circuit
- Design portable chassis to hold the circuitry on paper
- Design a Solidworks rendering of paper sketch
- 3D Print initial Solidworks rendering
- Integrate with microcontroller

Conclusions

Challenges

- Costs of materials became a bottleneck
- Open source tools had niche issues that arose deeper into the project

Lessons Learned

- Integration *takes time*
  - 3 weeks of slack time still was not enough
- Prototyping *takes time*
  - 3D print smaller parts before scaling up