# BeatBeam

**Daniel Bond, Brian Hickey,**
**Duncan Smith-Freedman, Brandon Sprague**
**Faculty Advisor: Prof. Michael Zink**

Daniel Bond | Brian Hickey | Duncan Smith-Freedman | Brandon Sprague

## Abstract

BeatBeam allows users with novice-level musical experience to easily create music in a collaborative fashion. The BeatBeam system is composed of four integrated subsystems that concentrate on providing the user with the necessary networking, web-based interface, audio-visual system, and signal interpretation system required to deliver a seamless musical experience.

The four components work together within a 3D printed central node to provide software generated music and lights that are designed to entertain users and provide a creative platform for those with no knowledge of music theory.

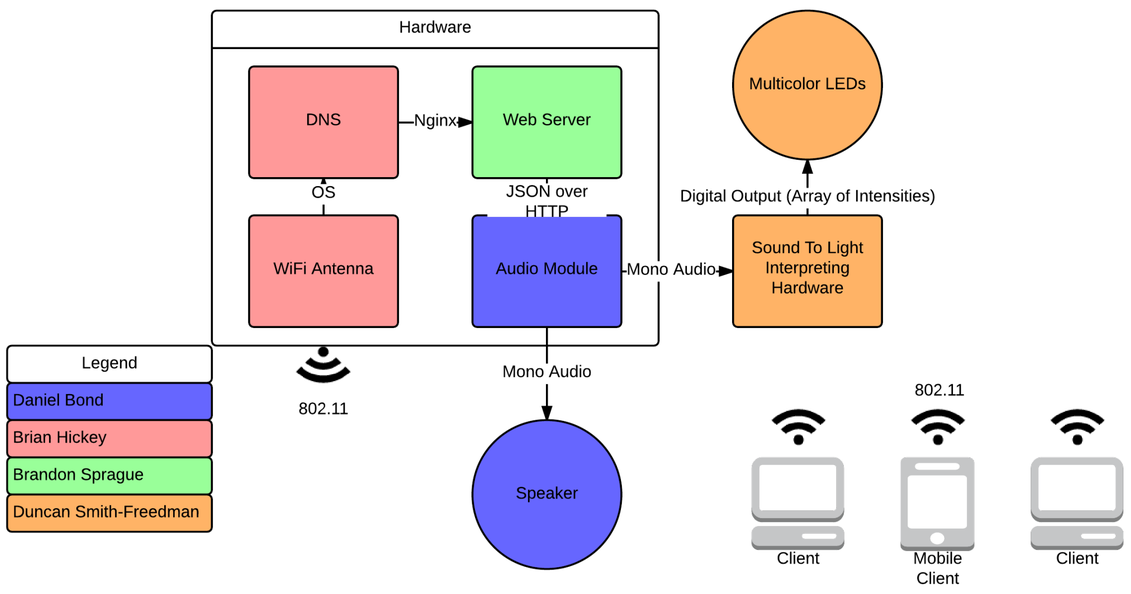
## System Overview

BeatBeam's core system function is the conversion of binary grid input from the end devices of users to pleasing music and light emanating from the central node.

The instrument grids and grid state are all maintained by the web server, which communicates with a Java music generation module over HTTP using JSON. The music generation module interprets each grid based on whether it is a tuned or untuned instrument, and then creates music using applied music theory and chord decomposition.

This output music signal is then analyzed by audio-to-light software and hardware to create an LED light show to accompany the created music.

## Block Diagram




## Results

BeatBeam provides an exciting and engaging experience for audiences of all ages to enjoy. The collaborative nature of the system fosters an environment where a large number of users can create music together that is pleasing to the ear while variable from a tempo and instrumentation standpoint.

## Specifications

Since the project was first conceived, our team has held BeatBeam to three strict specifications in order to provide a consistent user experience:

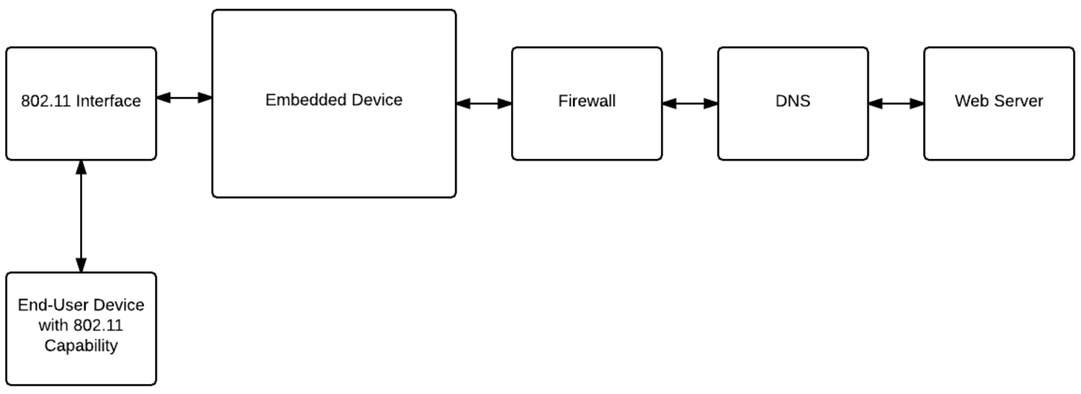
***1.)*** *Users with no prior musical experience will be able to make pleasing music more than 90% of the time.*
***2.)*** *Groups of 20 people will be able to simultaneously create music.*
***3.)*** *Less than 25ms delay across clients*

## Acknowledgement

A special thanks to our fearless advisor, Professor Mike Zink, who facilitated and provided meaningful advice related to the project.

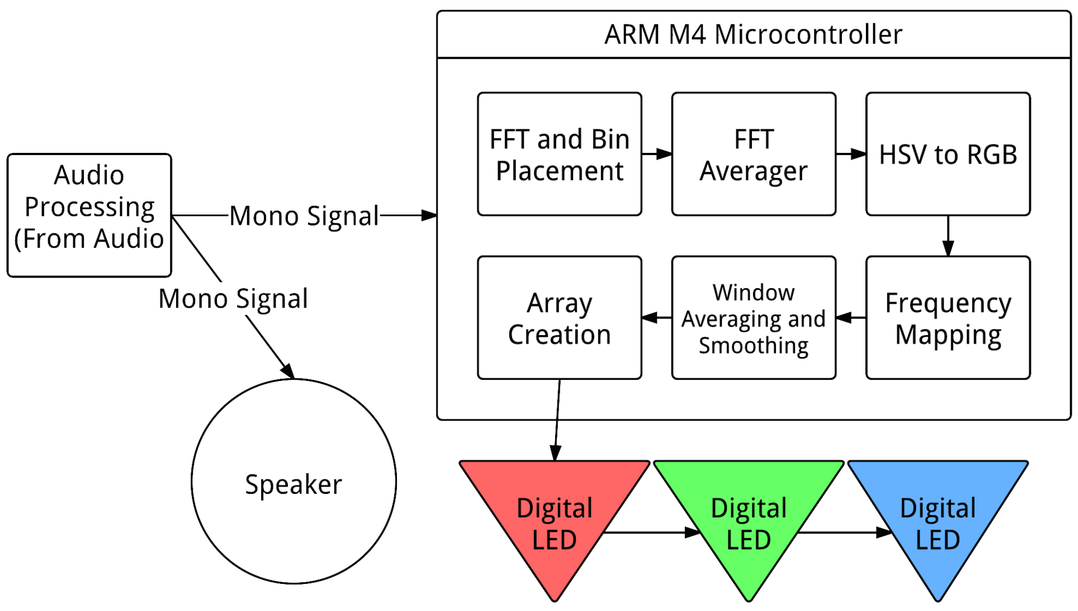
Further thanks to our evaluators, Professor Bardin and Professor Moritz, for their guidance and constructive criticism throughout the year.

# WiFi Routing




The networking module employs IEEE 802.11 for bidirectional communication between the end-user and the web server in the central node. A DHCP server facilitates addressing of clients, and DNS is employed in order to redirect all web requests to our HTTP server running the BeatBeam interface

# Light Generation




The light generation utilizes the Fast Fourier Transform (FFT) on the microcontroller to convert the music from time-based to frequency-based. The FFT is uniformly distributed among an array the length of the number of LEDs, and is then digitally output to the LED array.

# Cost

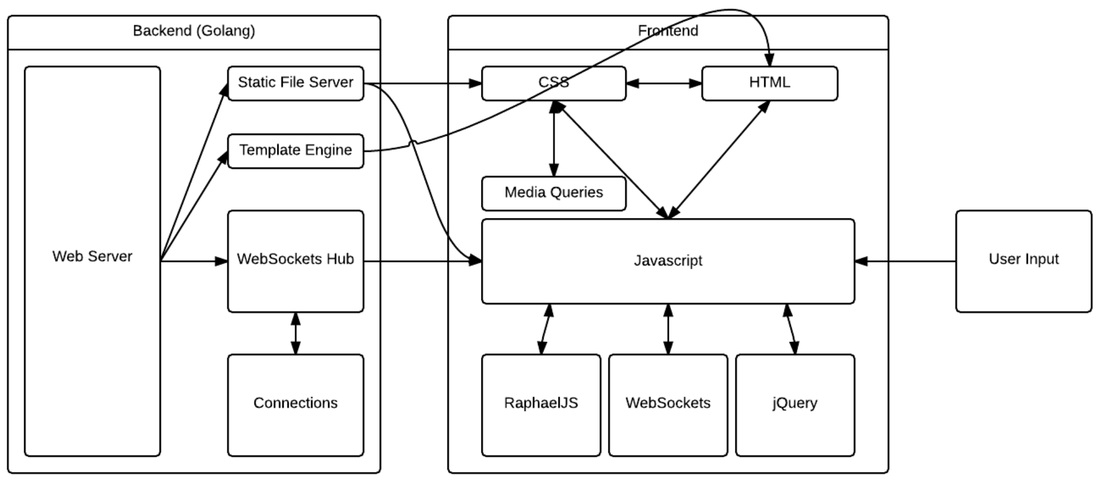
**Development**

| Part | Price |
|---|---|
| Raspberry Pi | 41.74 |
| Teensy | 24.01 |
| 3D Printed Housing | 50.00 |
| Audio Amp | 19.95 |
| Speakers | 43.90 |
| Power Supply | 39.95 |
| Total | 219.50 |

**Production**

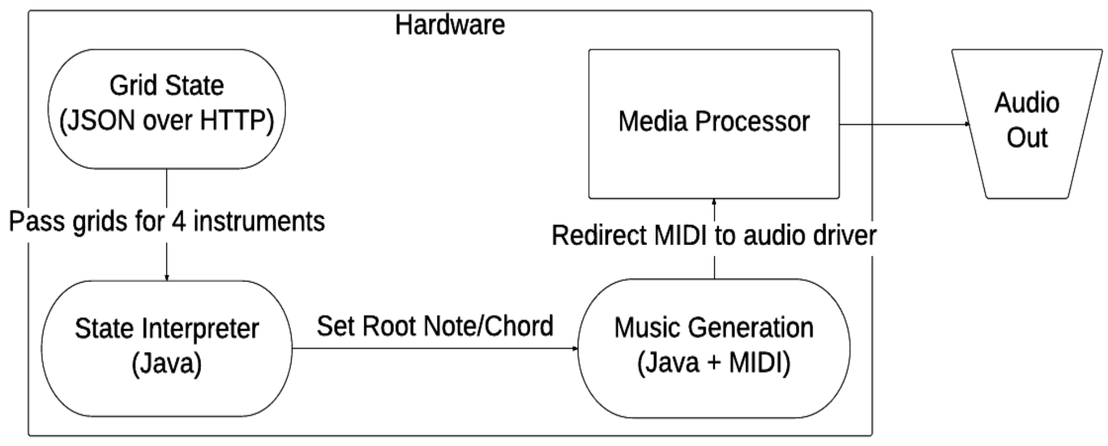
| Part | Price |
|---|---|
| Raspberry Pi | 39.95 |
| Teensy | 24.01 |
| 3D Printed Housing | 10.00 |
| Audio Amp | 3.75 |
| Speakers | 30.55 |
| Power Supply | 15.00 |
| Total | 123.26 |

# Web Server




The web server is written in the Go programming language. It accepts client requests, generates HTML pages, and maintains statistics and WebSocket connections.

# Music Generation




The music module is written entirely in Java, while leveraging the sound.midi package. After each music sequence, it communicates with the web server over HTTP using JSON to retrieve updated user grids and management settings.

# Experiment

Once the module was complete, it was given to different groups of UMass students and adults to test how it sounds and replayability. The size of these groups ranged from 3 to 8 people, with an age range of 18 to 45. Afterwards, each person was told to report on what her/she enjoyed, didn't enjoy, and give Beatbeam an overall grade.

Raters observed that while the project as a whole was an enjoyable experience, the cooperative nature of Beatbeam made it really fun. Making music with friends is most enjoyable when it can be done with ease and still sound pleasant.