LED CUBE EXPERIENCE +

Gustavo Correa

CS 122A Fall 2017

Phase I

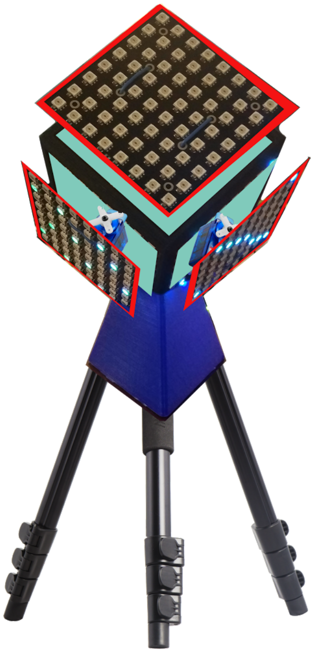
Introduction

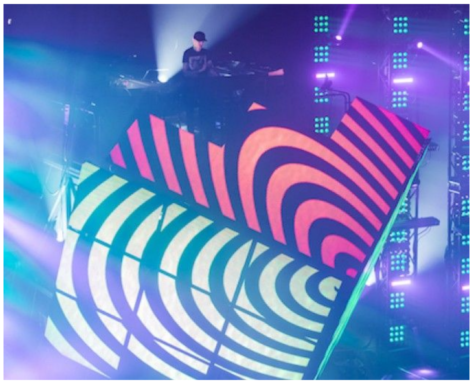
The first paragraph (3 - 5 sentences) should describe **why** you are building the project. Are you solving a problem? Is it a personal passion of yours? Did it sound like something fun? How did you come to decide to do this project.

- My goal for this project is to create an entertainment experience for anyone producing or playing music at a concert stage or venue. As a frequent concert goer and raver, I would like to redesign the concert/rave experience for the audience viewers and performers. My passion for art, music, and engineering has led me to create something that uses music to controls lights.

The second paragraph (3 - 5 sentences) will introduce what the project is. Describe how it could be used or where it would be "deployed." Include a description of how someone might interact with your project. Include a representative image of your project.

- I will be building upon what I created for my CS120B Project. In my CS120B project I displayed the amplitudes of sound picked up from an AUX cord and microphone using the LED Matrix Cube as the primary display device.
- What I would like to do now is modify the cube idea so that each LED Matrix panel is motorized and is able to expand and contract on the cube. My goal is to have the bass/hi-hat notes from music to synchronize with the lights and motors so that movement is visible on the cube. My inspiration came from Deadmau5's cube shown below.
- **The Cube will be designed to be deployed with a tripod at concerts/raves. This modular method of deployment follows the modular philosophy of Tait. It will allow for better transportation, support and stability.**
- **During a typical show both recorded music and live music that is played will display lighting/motor effects**
- **The music that is played on the keyboard will drive patterns on the LED displays**

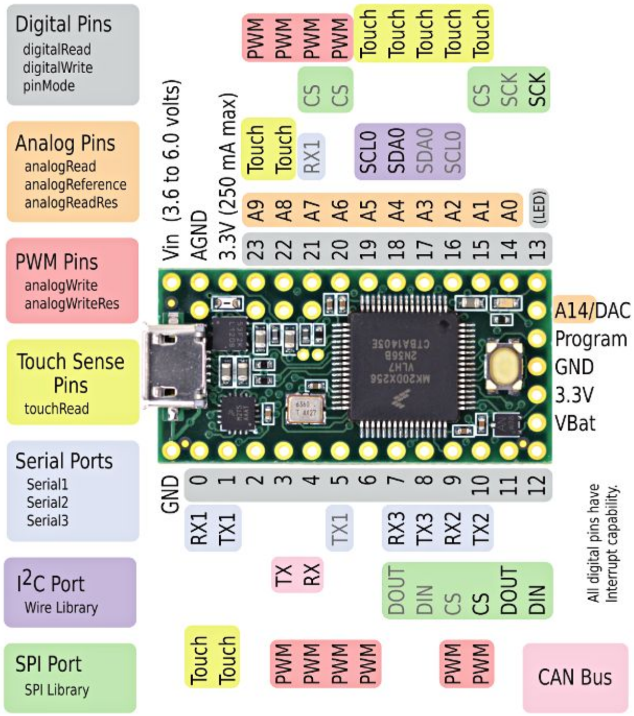




Components (Pin-out)

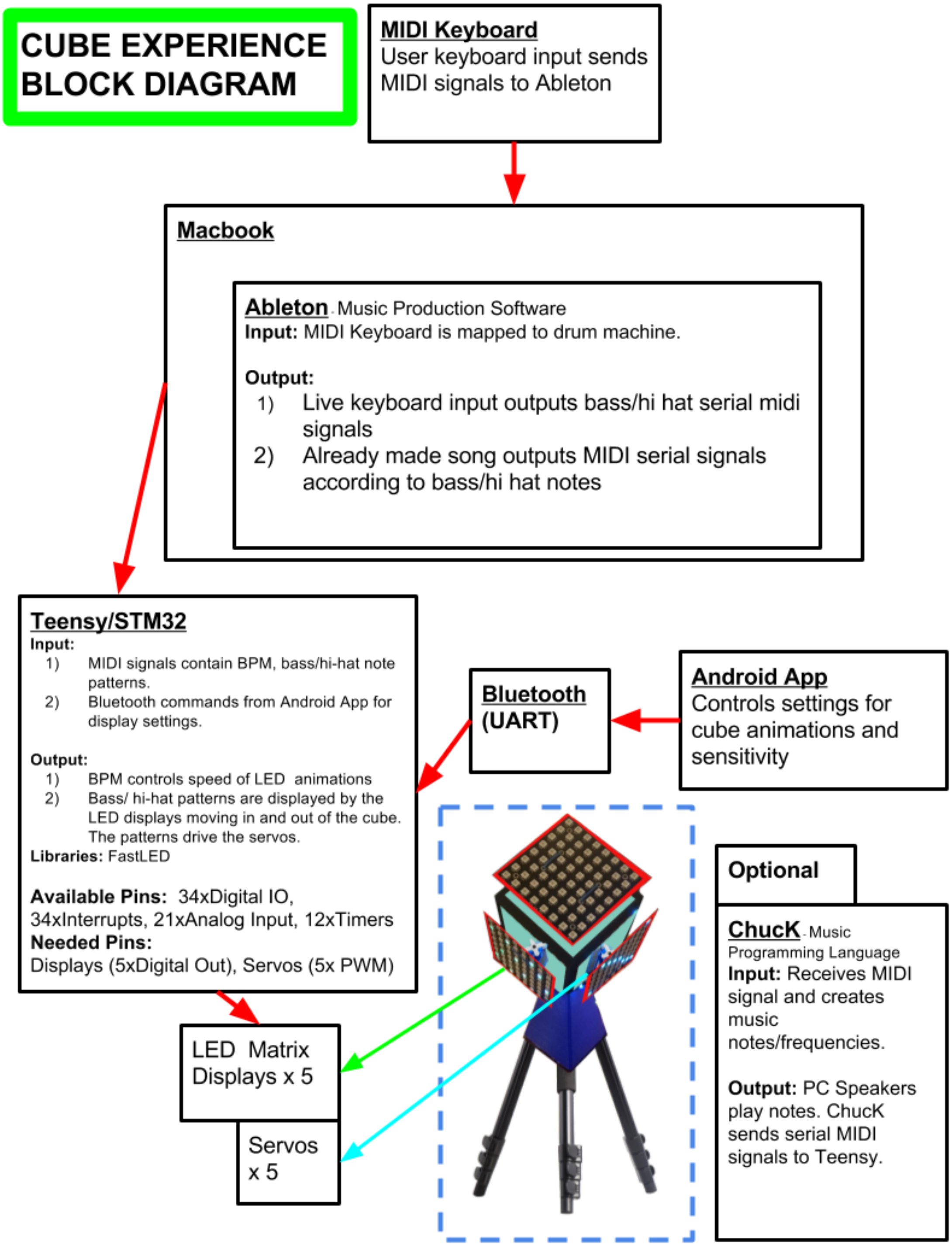
- **Inputs -** List of components used for input
  - **Teensy:**
    - MIDI input signals from Ableton Software contain BPM, bass/hi-hat note patterns
  - **MIDI Keyboard:**
    - User keyboard input sends MIDI signals to Ableton
  - **Ableton:**
    - MIDI keyboard is mapped to drum machine
  - **Android App:**
    - Accepts user input to control cube animation and sensitivity settings.
  - **Bluetooth Module:**
    - Accepts input via bluetooth from Android App.
- **Outputs -** List of components used for output
  - **Teensy:**
    - BPM controls speed of LED animations. Bass/ hi-hat patterns are displayed by the LED displays moving in and out of the cube. The patterns drive the servos.
  - **MIDI Keyboard:** Sends MIDI signals to Ableton
  - **Ableton:**
    - Live keyboard input outputs bass/hi-hat serial midi signals.
    - Already made song outputs MIDI serial signals according to bass/hi hat notes.
  - **Android App:**
    - Controls settings for cube animations and sensitivity.
  - **Bluetooth Module:**
    - Connects via UART to Teensy to accept input from Android App.
  - **LED Matrix Display:**
    - Displays animations that go with tempo.
    - Will be using the FASTLED library to output to the
  - **Servos:**
    - Move LED panels according to the beats of the music.
- **Internal components -** List of components that are used for “computation” but the user won’t interact with
  - Teensy, bluetooth module, led matrix displays, servos
- **Microcontrollers/Processors**
  - List of microcontrollers used (arduino, raspberry pi, FPGA’s, PLC’s, ATmega1284)
    - Teensy
  - For each microcontroller/processor that is **not** the ATMega1284 you will need to provide:

- Why do you need to use this processor (what is unique compared to the 1284)?
    - Given the complexity of the project I would like to use as many external libraries as possible to improve the data flow rate. If I were to implement the FASTLED library on my own I would have to recreate a library that has been improved over several years.
    - I would like to focus on completing the data path from MIDI Keyboard input all the way to LED Display and Servo output.
- Do you have experience working with this processor?
    - Yes, I have worked on it using the Arduino IDE
- Do you know where to go for help programming this processor?
    - IEEE UCR’s micromice use the Teensy for faster speeds, more memory, and more interrupts.



Include a block diagram on how all of the components will be connected. Include:

- How many IO pins are necessary for each connection
- How many IO pins each component has available
- Communication technologies used (TWI, SPI, USART, BT, etc.)



CUBE EXPERIENCE BLOCK DIAGRAM

MIDI Keyboard
User keyboard input sends MIDI signals to Ableton

Macbook

Ableton - Music Production Software
Input: MIDI Keyboard is mapped to drum machine.

Output:

1) Live keyboard input outputs bass/hi hat serial midi signals
2) Already made song outputs MIDI serial signals according to bass/hi hat notes

Teensy/STM32
Input:

1) MIDI signals contain BPM, bass/hi-hat note patterns.
2) Bluetooth commands from Android App for display settings.

Output:

1) BPM controls speed of LED animations
2) Bass/ hi-hat patterns are displayed by the LED displays moving in and out of the cube. The patterns drive the servos.

Libraries: FastLED

Available Pins: 34xDigital IO, 34xInterrupts, 21xAnalog Input, 12xTimers
Needed Pins:
Displays (5xDigital Out), Servos (5x PWM)

Bluetooth (UART)

Android App
Controls settings for cube animations and sensitivity

LED Matrix Displays x 5

Servos x 5

Optional

ChucK - Music Programming Language
Input: Receives MIDI signal and creates music notes/frequencies.

Output: PC Speakers play notes. ChucK sends serial MIDI signals to Teensy.

70 - 80 points project

What will you be doing to get your 70 - 80 implementation points. This is your base project to which you will be added additional functionality in terms of new software and/or hardware. What is the bare minimum you need to demo your concept.

- The bare minimum required to demo my concept will be to complete the path: **Keyboard Input → Ableton Drum Machine MIDI OUT to Teensy → Output to lights on the LED Displays and servo motor movement.**

80 - 90 points project

What will you be adding to reach that 80 - 90 points. What additional component (software or hardware) will you be adding, why is this sufficient to get you the additional "10" points.

- To receive 80-90 points the following is required:
    - **The Android App will be paired with the Teensy using Bluetooth/UART and will be used to control animation, brightness, and other settings for the Display and Motors.**
    - Android Apps are a perfect solution to provide a graphical user interface for portable projects/products. The app will complete the all in one package for music artists.

90 - 100 points project

What will you be adding to reach that 90 - 100 points. What additional component (software or hardware) will you be adding, why is this sufficient to get you the additional "10" points.

- To receive 90-100 points the following is required:
    - **The Cube will perfectly synchronize the bass/hi-hat notes in a song created in Ableton to lights/movement on the Displays.**
    - The entire goal for this project is to create a intuitive entertainment experience. In a live show, the data flow from music input to visual output has to be as seamless as possible and everything has to be synchronized together.

Phase II

Milestone

What is your target milestone? **When** is your intended milestone date (non-binding).

OBJECTIVES:

1. Order Servo motors, tripod adapter parts
2. Keyboard -> Ableton -> Teensy/Servo
3. Custom Song -> Cue Points -> Teensy/Lights
4. Teensy -> Servo/ LED Displays
5. Design Cube Frame
6. Android App -> Bluetooth -> Teensy
    a. **OR**: Web App -> Teensy
7. Testing

DATE	**MILESTONE (70-80) (80-90) (90-100)**	**STATUS**
Week 5 10/31	Order Servo Motors, lasers, tripod/mounts	
11/2	Keyboard -> Ableton -> Teensy/Servo	
Week 6 11/7	Teensy -> Servo/ LED Displays	
11/9 (Test)	Design Cube Frame	
Week 7 11/14	Android App -> Bluetooth -> Teensy **OR**: Web App -> Teensy	
11/16 (Test)	Finish App to Teensy Connection	
Week 8 11/21	Custom Song -> Cue Points -> Teensy/Lights	
11/23	Refine animations and servo responsiveness	
Week 9 11/28 (Test)	Hardware modifications	
11/30	Testing should be done	

Testing and Verification

What is your plan to test your invention?

- My plan to test my project is to have each milestone 100% working by the end of each due date. Given the way my system is designed, each module should integrate easily with one another. The following plan for the “point level” will go with the milestone schedule mentioned above.

How will you test each “point level” in your project?

- 70 - 80 point project
  - Tested by fellow IEEE officers
- 80 - 90 point project
  - Tested by freshman with small knowledge about arduino
- 90 - 100 point project
  - Tested by friends and mentors who have done projects like this before
  - Tested by random students around campus or in the library

When will you start testing each stage?

- 70 - 80 point project
  - **Week 6, 11/9**
- 80 - 90 point project
  - **Week 7, 11/16**
- 90 - 100 point project
  - **Week 9, 11/28**

If you notice the testers get more general as your project gets more mature. Engineers tend to have a better grasp of what these inventions are supposed to do, and are less likely to put in incorrect input. General majors are more likely to interact in a way that will illuminate bugs.

Form Factor

This is the non-engineering portion of the proposal/project. **IF** you were to make a case for your project, embed it in the environment, or “deploy” it in some fashion. What would that look like? Would you need to shrink the form factor down before deployment? Could you 3D print a case? Could you build one from wood?

NOTE: This is not required for this course, but presentation is still important, and it is something to think about doing before you take a project into an interview.

Features

The cube will be deployed onto a tripod type system.

This modular deployment method models the modular deployment method developed by Tait.

I will be designing a new frame for the cube and 3D printing it.

The new frame’s design will have mounts for the motors, displays, teensy, and screw hole for the tripod.