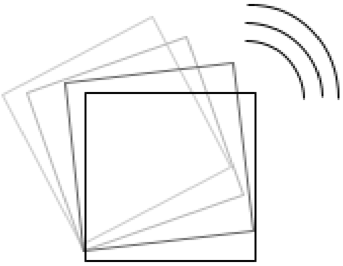
Music Cubes D23

Adam Kalayjian and Jeff Davis



Overview

This project aims to create an intuitive MIDI controller that utilizes the orientation of a cube in order to trigger samples, control effects, or control another MIDI instrument.

Progress

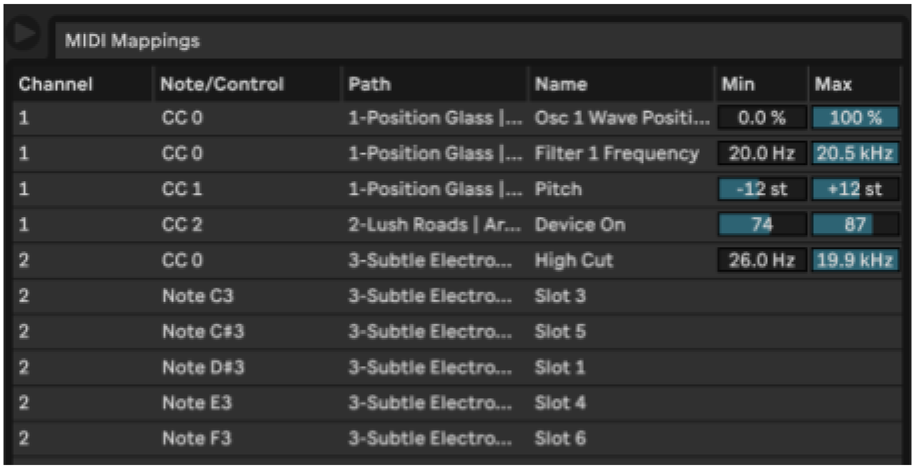
This term, we focused on implementing changes to the previous software and hardware of the cube to streamline usage and increase user-friendliness.

Software

The goal of this project's software is to transform data from the Seeed Studio XIAO nRF52840's Internal Measurement Unit (IMU) into usable MIDI data on a DAW, like Ableton Live.

MIDI over Bluetooth

The C23 group accomplished the Bluetooth Low Energy (BLE) connection to a computer utilizing Bluetooth advertising keys and a Python client. Instead, our solution utilizes the MIDI over Bluetooth protocol in order to directly interface with a computer as a MIDI device. We used the `bluefruit` and `MIDI.h` libraries to send MIDI Note and Control Change signals to the computer, removing the need for client code to interpret BLE data. Additionally, we utilize the `bluefruit` library with `base64.h` to create a unique ID for each cube, which is visible when you connect to the cube.



MIDI Mappings

Channel	Note/Control	Path	Name	Min	Max
1	CC 0	1-Position Glass |...	Osc 1 Wave Positi...	0.0 %	100 %
1	CC 0	1-Position Glass |...	Filter 1 Frequency	20.0 Hz	20.5 kHz
1	CC 1	1-Position Glass |...	Pitch	-12 st	+12 st
1	CC 2	2-Lush Roads | Ar...	Device On	74	87
2	CC 0	3-Subtle Electro...	High Cut	26.0 Hz	19.9 kHz
2	Note C3	3-Subtle Electro...	Slot 3		
2	Note C#3	3-Subtle Electro...	Slot 5		
2	Note D#3	3-Subtle Electro...	Slot 1		
2	Note E3	3-Subtle Electro...	Slot 4		
2	Note F3	3-Subtle Electro...	Slot 6		

Example of mapping MIDI CC and Note to effects/samples in Ableton Live.

Pose Estimation

In order to accurately estimate the pose of the cube using the IMU, we used a Madgwick filter using the `MadgwickAHRS.h` library to estimate the pitch, yaw, and roll of the cube. These values are converted into MIDI Control Change signals.

Workflow

Our solution implements two methods of usage, Dice mode and Tilt mode. When the cube is in motion, the precise pitch, yaw, and roll are transmitted as MIDI Control Change signals (Tilt mode). When the cube is "rolled" or stops moving, a MIDI Note On signal is generated, indicating which side the cube has landed on. A more traditional method of using the accelerometer is used to identify which side the cube is on, instead of the Madgwick filter. The cube transitions between these two modes using the raw IMU data.

Next Steps

Currently, each cube is hard-coded to a MIDI channel, meaning that signals can interfere if cubes are coded to the same channel. We attempted to create an auto-assigning channel system using MIDI over Bluetooth, but it was difficult to come up with a solution that doesn't require code or Max For Live on the computer side. We believe that another Bluetooth service must be created to identify other cubes and assign corresponding MIDI channels, but that this should not be too difficult using the `bluefruit` library. Additionally, some of the MIDI outputs can be finicky depending on the orientation of the cube (for example, the yaw value is pretty noisy in one particular orientation) but these issues can probably be solved with some additional Arduino code.

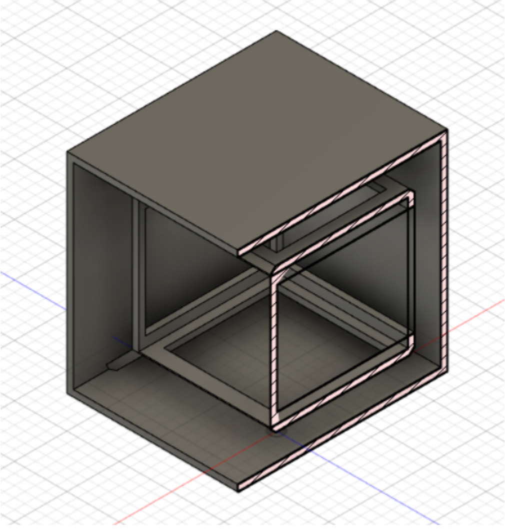
Hardware

The goal for hardware upgrading is focused on the outer casing of the Cube. The original casing of the cube has all of the components within a casing with a 1inx1inx1in. While this results in a convenient form factor, it makes it difficult to access the tightly packed components of the cube and leaves no room for additional components or future upgrades. In addition, we wished to add some form of lighting to the cube in order to convey information to the user, which the compactness of the original casing wouldn't allow.

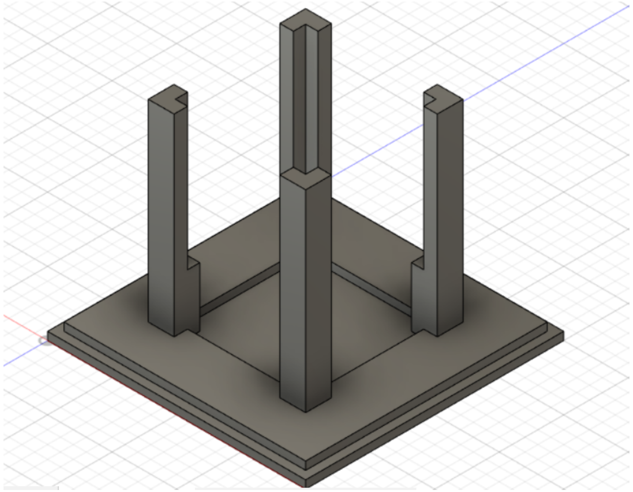
Outer Shell

The first change made to the outer shell of the cube is the size. In order to help facilitate easier access to the inner components, the outer profile of the cube has been increased to 2inx2inx2in. While this is a comparatively large increase, effectively doubling each dimension of the cube, the size remains manageable to be manipulated with one hand.

The second significant change made to the outer shell is the material that the shell is intended to be made of. The outer shell is now intended to be printed in a clear filament, which would allow informational LEDs to be seen through the walls of the cube, also lending to an aesthetic glowing effect.



Component Holder



In order to address the serviceability of the cube, a design was made to be able to easily remove the components from the outer shell as a separate piece. This piece slides into a central slot of the outer shell, holding the components in the center of the cube to keep the cube consistently weighted. The volume of the inside of this central slot is the same as the dimensions of the original shell as it is known that the components can fit within an area of that size. While this volume is the same as the original cube, not requiring the components to be covered will increase serviceability greatly. This piece also acts as a cover for the last side of the cube to create a seamless geometry with no holes.

Next Steps

Currently, there is no hardware setup for LEDs inside the cube. In addition, we are currently using the original cubes as the center “core” for the outer shell. While this is functional, it defeats the purpose of increasing the cube’s serviceability. Future teams can work on creating a new core for the cubes that house the electronics in a non-enclosed space with the addition of LEDs.