Vicki Moran
Spencer Rosen
E155 Final Project

Project Proposal - MIDI Visualization

Overview

We plan to implement a system that receives input from a MIDI keyboard and plays the corresponding note on a speaker while displaying the MIDI visualization of the input on an LED matrix. This will involve receiving user input from the MIDI keyboard through the Raspberry Pi, outputting a square wave of appropriate frequency to a speaker, and sending data about which keys are pressed to the FPGA. The FPGA will control an LED matrix to display which keys are pressed over time. There will be a feature to record a pattern and play it back in a loop. There will be three modes: live mode, recording mode, and playback mode, which will be signified by a status LED. Only one key can be pressed, played, and displayed at a time.

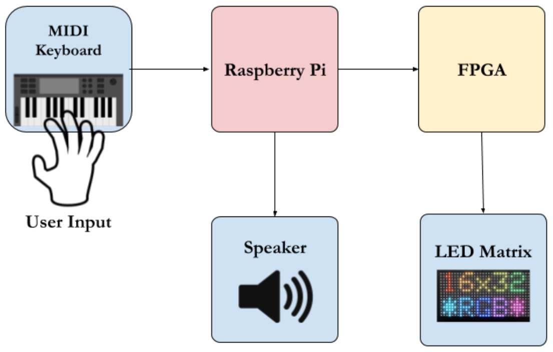

Figure 1: Block Diagram

Subsystems

MIDI Keyboard - the MIDI keyboard will plug into the Pi over USB and the Pi will identify which key is pressed using a modified open source library. In recording mode, the Pi will also store the duration each key is pressed.

Storing Patterns - in recording mode, each time a key is pressed or released, the Pi will store which note was pressed (including pauses) and the duration it was pressed in a data structure.

Playing Audio - the Raspberry Pi will generate square waves of a frequency corresponding to which note is to be played and pass them through a power amplification circuit connected to a speaker. In playback mode, the Pi will loop through a pattern of stored notes and durations and play each note for the duration it was pressed during recording mode.

SPI - the Pi will send data about which keys are pressed to the FPGA via SPI. The Raspberry Pi will be the master and the FPGA will be the slave in this communication protocol.

LED Matrix - the FPGA will set the RGB values of the LED matrix to display which notes are being played on the Pi. Each row of the matrix will be a constant color corresponding to the frequency of the note being played (a spectrum with red corresponding to high frequency notes and blue corresponding to low frequency notes).

New Hardware

We are using a MIDI keyboard. The keyboard plugs into the Pi via USB, and there are open source libraries to assist with reading the MIDI input to determine which key is pressed.

We are also using a 16x32 RGB LED matrix. The FPGA will send RGB data to the matrix to control the color of each pixel.

Raspberry Pi Functions

The Raspberry Pi will receive input from the MIDI keyboard, play the notes on a speaker, send data to the FPGA via SPI, and store a pattern of notes. In live mode, the Pi will read input from the MIDI keyboard, play the notes on the speaker, and send data to the FPGA via SPI. In recording mode, the Pi will perform all of the features of live mode while additionally storing which note was pressed and the duration of the press each time a key is pressed or released. In playback mode, the Pi will stop receiving input from the MIDI keyboard, and instead perform the features of live mode by looping over the stored notes from recording mode.

FPGA Functions

The FPGA will receive data from the Raspberry Pi via SPI and display a visual representation of which notes are played over time. The received data will tell the FPGA which keys are being pressed, and the FPGA will display which keys are being pressed on the LED matrix. There will be a push button for toggling between modes, a push button for reset, and an LED to display which mode the system is in.

Budget

midiplus MIDI Controller, 32-key (AKM320)

$37.12

https://www.amazon.com/midiplus-AKM320-MIDI-Keyboard-Controller/dp/B00VHKMK64

Medium 16x32 RGB LED Matrix Panel

$24.95

https://www.adafruit.com/product/420