

Krystopher Mandujano

Student ID: 63626130

EECS 113 Final Project Report

Final Project Description:

The project is a music game in which players compete against each to see who can correctly name, the title and the artist of songs played. The game starts with players selecting the number of participants. Then they select the length of the game. There are three different game lengths: Fast, Normal, and Slow. The Fast option features 2 rounds, the Normal option consists of 3 rounds, and the Slow option consists of 5 rounds. Each round consists of players listening to a 100 second sound clip that is broken up into 5 sections with each section being 20 seconds long. These sections are randomly selected snippets of songs. The Fast option features 10 randomly selected songs, the Normal option features 15 randomly selected songs, and the Slow option features 20 randomly selected songs. After selecting the game's length, the players will select the game's difficulty. The game has three different difficulties: Easy, Normal, and Hard. The Easy difficulty has the songs played unaltered; the Normal difficulty features some of the songs altered, and the Hard difficulty features all the song being altered. Altering the songs involves changing the song's pitch, the speed of the song or adding audio filters to alter some part of the song. After the clip finishes, players take turns trying to guess the song's title. Points are given to players who correctly give the song's title and artist. The goal of the game is to have the highest amount of points at the end of the game. At the end of each round, a round result information which includes the songs players missed and the number of points they gained that round. In addition, a scoreboard is shown that displays the point totals for all of the players.

Project Challenges:

One of the biggest challenges I faced with my project is programming the driver for the PMOD Keypad. I had to deal with choosing the right interface to use with the keypad. At first, I tried to use the GPIO interface for the keypad, however, I ran into issues with trying to read the input for the keypad. The problem was that the key pad required that the GPIO pins to be Input/Output able, however, the PYNQ-Z1 board did not allow GPIO pins to be both input and output.

Instead, I decided to use the SPI protocol to connect the keypad and board. The SPI protocol was prefect for the keypad since the keypad accepted SPI protocol. In addition, SPI allowed for serial communication between the board and keypad in a master-slave architecture. This architecture was useful for the keypad since it meant that I could read and send inputs to the keypad at my own pace. This made checking for a key press easier than with GPIO since it was harder to write the address to of the GPIO pins than with the SPI write functions. However due to time constraints, my keypad is not well test, thus it may not work. As a result, I created a second piece of code that use the computer for player input.

Another challenge I faced was programming the FFMPEG sections of the game. My problems involved with learning FFMPEG for the first time. This made looking for the correct code syntax harder because I did not what some of the filter functions required so I decided to stick the filters that had clearly define instructions.

Parts:

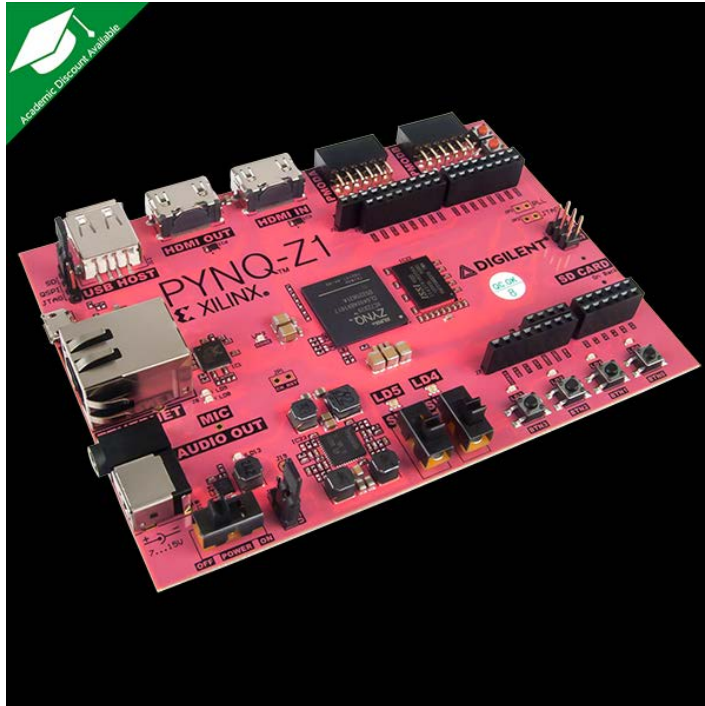
Grove OLED Display 0.96''



Digilent Pmod Keypad



PYNQ_Z1 Board



PYNQ Shield

