

ECE 4873 Project Summary

Project Title	Multichannel Audio Synthesizer
Team Members (names and majors)	Joseph Farnham Moradeke Olumogba
Advisor / Section	Dr. Whit Smith / ECE4873 sd20s01
Semester	Year/ <u>Semester</u> Circle: Either Intermediate (ECE4011) or Final (ECE4873)
Project Abstract (250-300 words)	<p>The two main goals of our project were to develop a device that could sample audio data simultaneously across several channels and to create software that would convert the raw audio data into a multichannel .wav file. While devices that sample audio data on multiple channels already exist, those devices are too expensive for many enthusiasts and hobbyists who would otherwise be interested in the device. Therefore, our project arose out of a desire to have a lower cost alternative to devices currently in the market. We were motivated to do this project because we knew we had the talent to successfully complete it, and we also wanted to pass the senior design course.</p> <p>While we made substantial progress, we unfortunately did not achieve our goals due to a hardware issue that prevented our device from properly capturing audio data. That being said, we were able to initialize all of our components properly, which included the ADC, the flip-flop, and the FTDI USB chip, and see that the ADC could transmit data through the flip-flop to the pins of the USB chip. We also verified that we could affect the sampling rate of the ADC by changing the sampling rate specified in our software. We were able to verify these functionalities by using a logic analyzer that we hooked up to a computer. Once the logic analyzer was hooked up to the computer, we used an open-source application named PulseView to control the logic analyzer, and view voltage traces from individual pins.</p> <p>In the end, while our design was not perfect, we understand the root cause of why our design failed and will detail our findings in our final report. We also used the schematic from the following group who completed this project in 2014: http://ece4012y202002.ece.gatech.edu/14spring/ECE4012L1A/ws2/</p>

Project Title	
List codes and standards that significantly affect your project. Briefly describe how they influenced your design.	<p>USB 2.0 – FTDI embedded the entire USB protocol on the FT232H chip of theirs that we used</p> <p>C++11 – Our code was written in C++ to leverage the driver that FTDI provides</p> <p>SPI – We used the SPI protocol to initialize the ADC from the USB chip</p>
List at least two significant realistic design constraints that applied to your project. Briefly describe how they affected your design.	<p>Cost – We were tasked with creating a device that was cheaper than comparable devices on the market. This affected our component selection.</p> <p>Number of channels needed – We were tasked with creating a device that could simultaneously sample eight channels of audio data. Therefore, we needed to select chips that could support transmitting and sampling at least eight channels of audio data.</p>
Briefly explain two significant trade-offs considered in your design, including options considered and the solution chosen.	<p>We chose to add a flip-flop to our design. While the flip-flop may not have been necessary and made the device more expensive, it ensured timing synchronization between the ADC and USB chip. We trusted our design to function better with a flip-flop included. The 2014 team also included a flip flop, but we wanted to make sure we understood why it was there before including it in the device we wanted to produce.</p> <p>We chose a USB chip that had the USB protocol embedded in it. We could have chosen a chip that did not embed the USB protocol. However, we decided to chose the USB chip we did because it significantly reduced our workload, and therefore time to complete the project.</p>
Briefly describe the computing aspects of your projects, specifically identifying hardware-software tradeoffs, interfaces, and/or interactions. <i>Complete if applicable; required if team includes CmpE majors.</i>	<p>Our software used the USB driver and API provided to us by FTDI. Given we did not have to handle any portions of the USB protocol, or create the driver for the chip, we could focus on the real-time hardware interaction between the separate subcomponents in our design.</p> <p>We also used the SPI protocol to transmit serial data from the USB chip to the ADC chip. We manually toggled pins on the USB chip through software to transmit data using SPI.</p>