Project Summary

The criteria our Oscilloscope needs to solve includes having two functional channels, a sampling rate fast enough o measure a 1 MHz signal, probes that connect/disconnect using robust connectors, a configurable trigger, adjustable time and voltage axis, and button presses that respond under 20 milliseconds. An additional feature the team aimed to solve is allowing users to save a waveform as a .CSV file onto an SD card, and plot on another device.

Starting with the design process, we knew we wanted a way of solving for the high sampling rate. An Arduino UNO wouldn't be able to normally get a sampling rate as fast enough to measure a 1 MHz signal, which leads to talks with peers. From there, we discovered the Teensy 4.1, which also functioned with Arduino code. The BNC connectors were how we wanted to approach the robust connections between the ports. This worked well, as our probes connect to BNC ports, which is the case for most oscilloscopes. An interesting development was implementing the SD card functionality, which was figured out when reading further into the Teensy 4.1 datasheet. Using C code, we were able to implement a wave capture that acts similarly to that of an average oscilloscope.

To fully utilize the capabilities of the Teensy 4.1 fast sampling adcs' we utilized a buffer library that writes the sampled values to direct access memory in the Teensy. This approach was chosen because it allows for a faster collection of data than using a for loop that uses analog read to assign the values to an array in the script. To implement the triggering, voltage scaling, and frequency scaling we read the data from direct access memory, manipulate it and then store the values in a buffer we can use for plotting. For voltage scaling we simply multiply the values against a scalar, and for frequency scaling we zoom in by skipping samples within the direct access memory when transferring it to a buffer.

To display the results we used Processing, and had it communicate with the Teensy via serial communication. To establish a connection, tell Processing what channels are on, and to tell the Teensy when more data is needed a protocol was created. The protocol involves using the byte associated with "A" to tell the Teensy and Processing when a connection has been established and when more data can be sent. In addition, the protocol involves using the first byte sent from the Teensy to Processing to communicate which channels are on, allowing Processing to correctly interpret the values being sent.

A big part of this experience was the growth and learning done from the whole design process. One thing the team reflected that we wished we had done was more prototyping. With heavy courses accompanying Junior design, making time to experiment with new ideas wasn't always an option for the team. Additionally, the PCB design was a new experience for the team. While we managed to get a PCB on time, there were some problems with our design philosophy that we hadn't re-evaluated. Communication was a big key for this project, and something the team really values from the whole experience.