Project Summary

Oscilloscope Group 002-3: Nick Kim, Nathan Henry, Liam Warner

The oscilloscope we built needed to have two input channels, a capable user interface, a high sample rate, modularity, sturdiness, portability, responsiveness, and the ability to be used with high voltages between -10 and +10 Volts.

When tasked with this problem, we first decided what device we would need to read, store, and process samples. This board also needed to handle display and user interface code. We chose the Arduino Due due to its useability and high clock speed. The attenuation circuit, button, and knob were all designed to be put on a custom-made PCB which would interface with the Arduino. The attenuator scales the +/- 10 V signal to a 0 to 3.3 V signal, according to the Arduino's ADC specification. The display portion of the project used a VGA display library on the Arduino Due. The functions provided in the library made it easy to design a custom GUI and implement voltage and time axis scaling adjustments. The sampling rate and trigger functionality were handled by the data acquisition and processing portion of the project. The data acquisition was first designed to use a DE-10 Lite FPGA, which met the engineering requirement to have a 1MHz sampling rate. However, integrating the FPGA with the Arduino Due would have created many issues with integration, so the team decided to have the Arduino handle all the computing for the project. The Arduino's ADC was set in freerunning mode, which allowed for a maximum sampling rate of 0.1 MHz after the project was fully integrated. This fell short of the 1 MHz engineering requirement, meaning that the oscilloscope cannot accurately measure signals higher than around 10 kHz. The user interface code was designed to detect when the button or knob is changed and updates the appropriate variables in the program. The VGA code is designed to display voltage and time axes, both input signals, and a menu that shows which adjustment mode is currently selected. The value of each adjustment mode, voltage, time, and trigger, are shown on the screen, and turning the knob scales the respective axis or adjusts the trigger level when the appropriate mode is selected. Finally, the PCB, knob, button, VGA port, and Arduino Due were securely fastened to a CAD-designed 3D-printed enclosure using standoff screws. The enclosure includes a raceway lid and TPU corner caps which offer additional protection, making the system able to withstand a 3-foot drop test. After all the individual blocks were designed, the team spent several weeks integrating, revising, and debugging each aspect of the project to make a cohesive final product. The final version of the oscilloscope is portable, sturdy, responsive, and easy to use.

All of us learned some valuable lessons throughout this project. One lesson learned is how valuable research during the design phase is. Our individual blocks performed well on their own but proved difficult to integrate with one another since we could not see potential issues beforehand. Another lesson learned during this project is knowing when to revise aspects of the project's design. There were instances during the build phase where portions of individual blocks needed to be cut. The peak-to-peak voltage and frequency calculations made the sample rate slower and disrupted the VGA output. Removing them made the debugging process faster, allowing us to focus on more important aspects of the project during the integration phase. Overall, we learned that it is important to trust in each other's abilities and to be kind to one another during all stages of the project.