Project Summary

Oscilloscopes are devices used to measure analog signals, and display them on a digital display. Oscilloscopes are used in applications where learning about specific characteristics of a signal is important. Such characteristics include things like peak to peak voltage and period/frequency. Modern day oscilloscopes have trigger levels, which freeze the current wave on screen when the signal reaches a certain level, in order to get measurements. These are all features that users have come to expect out of a modern day oscilloscope.

In this project, the goal was to create an oscilloscope from scratch that shares many of these same quality of life features that users have come to expect from modern day oscilloscopes. This project uses two BNC connectors in order to connect the analog signal generator to the oscilloscope itself. The BNC connectors are connected to a PCB which then wires the signal to an Arduino Uno microcontroller. The Arduino Uno collects the data displays on a laptop display. This system is convenient as only two connections have to be made in order for the user to start seeing the signal on the laptop screen: connecting the signal via the BNC connector, and connecting the Arduino Uno to the user's laptop via a USB connection.

To approach the project building process, the system was first split into six blocks: PCB, microcontroller hardware, laptop functionality, data acquisition code, data calculation code and trigger level. Each person was given two blocks to work on. Each block was expected to work with its given inputs, and create the expected outputs. This ensured that once the full system was connected, all of the blocks would connect and work in harmony with each other.

One of the main components required in a modern digital oscilloscope is the use of an analog-to-digital converter (ADC). This converts the incoming analog signal into a digital signal capable of interfacing with a digital device like a laptop. Not using the Arduino built in ADC would have been ideal, as having an external ADC would result in a higher sampling rate and a clearer image because of a higher bit rate. Unfortunately actually implementing an external ADC proved to be more difficult than first assumed, as in order to collect the data from the ADC, it needed to be wired up to the Arduino Uno using Serial Peripheral Interface (SPI). Using SPI would have required using multiple Arduino Unos to collect data from two channels. This would have complicated the stream of data, and was difficult to code. As such, we decided to pivot and use the internal ADC on the Arduino Uno as it was much easier to implement in the final system.

This project was a major learning experience of ups and downs. One important takeaway is that nothing is certain until it has been researched. Making assumptions about how to implement a component or do a certain task is often more complicated than originally thought. So do research early, and make sure that you have a good idea of how to implement something before starting work on it.

Lab 1: 1/13/23

Ethan Burton
Rabecka Moffit
Shale Travis

Course Timeline

Week 1 1/13 (Done by Friday of the week)

- Decide goals
- Decide timeline
- Course timeline due

Week 2 1/20

- Have block diagram finished
- Goals submission

Week 3 1/27
- Design PCB
- Block diagram due

Week 4 ~~1/27~~ 2/3
- Time for midterms
- Overflow week if needed
- First block check off

Week 5 ~~2/3~~ 2/10
- Obtain PCB (have it with us)
- Write programs needed

Week 6 ~~2/10~~ 2/17
- Assembly
- Code (if needed)

Goals

- The project should reflect cutting edge technology driving industry trends
    - System uses an FPGA for data acquisition & a high level OS for interfacing w/ user

- The system must have a start/stop function
    - The system must be able to stop reading the signal & then start again.

Week 7 ~~2/17~~ 2/24
- Testing & debugging
- Second block ✓-off

Week 8 3/3
- Testing & debugging

Week 9 3/10

- Project Due