College of Engineering and Computing
VOLGENAU SCHOOL OF ENGINEERING
George Mason University

ECE 493 Senior Design Project

Affordable USB Multi-Function Lab Instrument

Final Project Report

Team Members:

Braden Camp

Timothy Kolos

Anna Mai

Erick Orellana

Anh Nhat Van

Faculty Supervisor: Dr. Jens-Peter Kaps

Associate Faculty: Dr. Craig Lorie

11 May 2025

Table of Contents

1. Abstract 6
2. Problem Statement 6
2.1 Motivation and Identification of Need 6
2.2 Market Review 6
3. Project Approach 8
3.1 Background Knowledge 8
3.1.1 MCU 8
3.1.2 Oscilloscope 8
3.1.3 Arbitrary Waveform Generator 9
3.1.4 Logic Analyzer 10
3.1.5 GUI 11
3.2 Problem Analysis 11
3.3 Our Approach 11
3.3.1 MCU 11
3.3.2 Hardware 12
3.3.3 GUI 12
3.4 Alternative Approaches 13
3.5 Project Requirements Specification 14
3.5.1 Mission Requirements 14
3.5.2 Operational Requirements 14
4. System Design 15
4.1 Functional Decomposition 15
4.2 Physical Architecture 16
4.3 System Architecture 17
4.4 Microcontroller Details and Design 17
4.4.1 Program 17
4.4.2 Oscilloscope 17
4.4.3 Logic Analyzer 19
4.4.4 Waveform Generator 20
4.4.5 USB Communication 20

4.4.6 Microcontroller Diagram 24
4.4.7 MCU Recommendations 24
4.5 PCB Design 25
4.5.1 Layout Design 25
4.5.2 PCB Assembly 27
4.6 GUI Details and Design 28
4.7 Case Design 32
5. Experimentation and Testing 34
5.1 Preliminary Experiment 34
5.2 Experimentation 35
5.2.1 MCU Experimentation 35
5.2.2 Analog Front-End Experimentation 37
5.2.3 GUI Experimentation 40
6. Technical Background 40
6.1 Analog Front-End 40
6.1.1 Signal Conditioning and Attenuation 41
6.1.2 Voltage Gain Amplification and Op-Amp Upgrade 41
6.1.3 DC Offset Injection and Buffering 42
6.2 Arbitrary Waveform Generator (AWG) Background 43
6.2.1 Frequency and Amplitude Range 43
6.2.2 Waveform Resolution 43
6.2.3 Digital-to-Analog Converter (DAC) 43
6.2.4 Operational Amplifier 44
6.3 Logic Analyzer 44
6.3.1 Sampling Rate and Time Resolution 44
6.3.2 Memory Depth and Record Length 44
6.3.3 Digital Signal Levels and Accuracy 44
6.3.4 Triggering and Decoding Capabilities 44
7. Schematic Design – Analog Front End 45
7.1 Oscilloscope 45
7.1.1 Input Stage 45
7.1.2 ADC and Signal Conversion 45

7.1.3 Amplification and Op-Amp Stage 46
7.2 Arbitrary Waveform Generator (AWG) 47
7.2.1 Frequency and Amplitude 47
7.2.2 Waveform Resolution and DAC 48
7.2.3 Amplification Stage 48
7.2.4 Dual-Channel Operation 48
7.3 Logic Analyzer 49
7.3.1 Sampling Rate and Channels 49
7.4 Power Supply 50
8. Administration 53
8.1 Allocation of Responsibilities 53
8.2 Funds Spent 53
8.2.1 Prototyping Costs 53
8.2.2 Final Product Cost Breakdown 54
8.3 Man-Hours Devoted to the Project 54
8.3 Teaming Experience 54
8.3.1 Team Communication and Dynamics 54
8.3.2 Project Management 55
9. Problems Faced 55
9.1 PCB Design and Assembly 55
9.2 Signal Integrity and Noise 55
9.3 Overload and Signal Clipping 56
9.4 MCU 56
9.5 GUI 57
10. Lessons Learned 57
10.1 Knowledge Gained 57
10.1.1 Front End 57
10.1.2 MCU 57
10.1.3 GUI 57
10.1.4 CAD Software and PCB 58
10.2 Key Takeaways 58
10.2.1 Trust but Verify 58

10.2.2 Prototype Early 58
10.2.3 Prioritize Performance Over Budget Early 58
11. References 59
Appendix: Proposal (ECE 492) 59

1. Abstract

The purpose of this project has been to design and construct an affordable multi-function lab instrument that will satisfy the needs of the Electrical and Computer Engineering (ECE) department's lab curricula. To accomplish this, the lab instrument includes a dual channel oscilloscope, arbitrary waveform generator, and logic analyzer. Ultimately, the device has similar features to other multi-function lab equipment on the market while being available at a lower cost. This project will be the culmination of the combined efforts of three past ECE senior design teams, integrating their separate projects into a single device.

2. Problem Statement

2.1 Motivation and Identification of Need

The ECE department at George Mason University currently requires students to purchase multi-function lab equipment such as the Analog Discovery 2 (AD2) or Advanced Active Learning Module (ADALM 2000). These devices are necessary for students to complete lab work for various classes in the ECE curriculum. The devices support students in the completion of coursework and projects outside of the classroom should an on-campus laboratory be unavailable, such as in the case of distance learning or during times outside open lab hours.

While prices on multi-function lab instruments have stabilized since the supply constraints of the pandemic, they can still be quite expensive today. The ADALM2000's price is roughly $230 at the time of writing. The more capable, and now retired, AD2 has been replaced by the AD3 with a current price of $379. Not all of the functions and capabilities of these devices are required for the ECE curriculum, and there are currently no budget-friendly variations of these devices.

This senior design project created a purpose-built USB multi-function lab device at a much-reduced price compared to the AD2, AD3 and ADALM2000. The device has two ADC channels for the oscilloscope with a 12-bit resolution and has a sampling rate of 2.5MS/s. There are also two 12-bit DAC channels sampling at around 2.5MS/s for the waveform generator and the necessary GPIO for the logic analyzer. The device receives power and transmits data over a USB interface, with the user able to control the device's functions through a custom GUI on the host system. The capabilities and accuracy of this device are adequate for the needs of ECE students at an affordable price point.

2.2 Market Review

Model	Analog Discovery 3	ADALM 2000	Pico 2204A BASIC	BitScope Micro BS05	Our Device
					

Price (USD)	$249	$210	$129	$145	$50
Oscilloscope					
Sample Rate (MS/s)	125	100	50	N/A	2.5
Number of Channels	2	2	2	2	2
Sampling Resolution	14-bit	12-bit	12-bit	N/A	12-bit
Waveform Generator					
Sample Rate (MS/s)	125	100	20	N/A	2.5
Number of Channels	2	2	2	N/A	2
Voltage Range	±5V	±5V	±2V	N/A	±5V
Logic Analyzer					
Channels	16	16	N/A	8	16
Logic Level	3.3V CMOS	3.3V CMOS	N/A	3.3V CMOS	3.3V CMOS
Sample Rate (MS/s)	125	100	N/A	40	5

Table 1: Current Available Alternative Multi-Function Lab Equipment

3. Project Approach

3.1 Background Knowledge

3.1.1 MCU

The microcontroller is a simple processor with expanded GPIO and analog capabilities included in its design for embedded applications. It is connected to an array of GPIO pins with differing purposes and capabilities. This project uses both ADCs and DACs. ADCs (analog-to-digital converters) take in an analog electrical signal from something like a thermometer or a photoresistor and convert it into a digital signal that can be processed by the microcontroller. DACs (digital-to-analog converters) perform an inverse function. They are supplied with a digital signal from the microcontroller and convert it into an analog electrical signal.

The CPU core onboard the microcontroller is responsible for executing instructions in the user's program. Memory on the microcontroller is broken into two categories, RAM and flash. Flash is slow to read and write, but is nonvolatile, meaning that it retains its state when power is lost. Flash memory is often used to store program data. RAM is faster but volatile, meaning that it is reset when the system powers down. It is also usually smaller in capacity due to its higher price. RAM is used to store temporary data and program variables so that the CPU can access them more easily. Timers in the microcontroller are used to time events, such as processing data from an ADC or enabling a clock signal for the microcontroller to communicate with the host system over USB.

3.1.2 Oscilloscope

An oscilloscope is an essential instrument used to measure fluctuations in voltage or current and provides a visualization and analysis of electrical signals over time. The primary function of an oscilloscope is to plot voltage against time, enabling users to observe waveform properties such as amplitude, frequency, rise time, and distortion.

Bandwidth

Bandwidth refers to the range of input frequencies that can be accurately measured and captured by an oscilloscope. To ensure proper signal capture, the bandwidth of the oscilloscope must be higher than the frequency of the signal being measured. The Rule of Five is commonly used, which suggests that the oscilloscope bandwidth should be a minimum of five times the highest frequency component of the signal. This minimizes errors due to bandwidth limitations, keeping errors within a ±2% tolerance. For example, an oscilloscope with a 2MHz bandwidth can effectively measure signals with frequencies up to 500kHz. This is the bandwidth specification we are aiming to meet with our device.

Rise Time

Rise time is the oscilloscope's ability to recognize and capture the rising and falling edges of a signal. It is closely related to bandwidth and can be approximated using the formula

$$t_R = \frac{0.35}{B}$$

where *B* is the bandwidth of the oscilloscope. A faster rise time ensures more accurate signal presentation, particularly for digital and high-speed signals, where rapid transitions are critical.

Sample Rate

The sample rate is the number of data points, or samples, the oscilloscope captures per second. An adequate sample rate is crucial to accurately visualize the signal on the screen. The sample rate should be at least 2.5 times the highest frequency component of the signal to avoid under-sampling, which can lead to aliasing and inaccurate results.

Number of Channels

Oscilloscopes typically come with multiple channels, allowing users to simultaneously measure and compare multiple signals. Oscilloscopes with two or four channels are the most common, offering flexibility for various measurements scenarios in circuit testing and debugging.

Vertical Resolution

Vertical resolution refers to the oscilloscope's ability to distinguish between different voltage levels, and it is determined by the resolution of the oscilloscope's ADC. Typically measured in bits, the vertical resolution defines the number of discrete levels the oscilloscope can use to represent the input signal. With the STM32-H562RGT6 microcontroller, a 12-bit ADC channel provides 4096 distinct levels of resolution, which means the signal is divided into 4096 possible voltage values. A higher vertical resolution allows for more precise measurements of small signal variations, making it essential for capturing fine details in low-amplitude signals or when observing small changes in signal voltage.

3.1.3 Arbitrary Waveform Generator

Frequency and Amplitude Range

The frequency range of an AWG defines the spectrum of frequencies it can generate, while its amplitude range determines the maximum and minimum voltages the device can output. These two parameters are crucial for testing and simulating various real-world scenarios. Typical AWGs are capable of generating frequencies ranging from a few Hz to several MHz, enabling the testing of both slow and high-speed circuits.

Waveform Resolution

Waveform resolution refers to the bit depth of a digital-to-analog converter (DAC) within the MCU, which determines the granularity of the waveform it generates. A higher bit depth, such as 12-bit or 14-bit, allows the AWG to generate smoother, more accurate output

waveforms. In the case of this project, the 12-bit DAC onboard the MCU can resolve 4096 distinct voltage levels, providing sufficient accuracy for most applications.

Digital-to-Analog Converter (DAC)

The DAC is a critical component in the design of the AWG, converting digital waveform data into analog signals. The DAC determines the accuracy of the output waveform based on its resolution and sampling rate. The sampling rate of the DAC should be at least ten times the frequency of the waveform being generated to ensure high-fidelity waveform reproduction. Higher DAC resolutions and sample rates enable the AWG to output complex, high-quality waveforms suitable for a wide range of testing scenarios.

Operational Amplifier (OpAmp)

After the waveform is generated by the DAC, it is passed through an operational amplifier (opamp) to amplify the signal, adjusting the output voltage to desired levels. The opamp ensures that the waveform has the correct amplitude to drive external circuits. The gain of the opamp can be adjusted to provide a wide range of output voltages, such as ±5V to ±12V depending on the project requirements. For our project, the opamp circuit needs to be capable of outputting a large signal that can drive up to a 20mA load.

Number of Channels

Similar to oscilloscopes, AWGs often come with multiple output channels. In our case, we have included two 12-bit output channels in our design. This allows users to generate two independent waveforms simultaneously, which can be useful in testing dual-input systems or performing synchronous signal testing.

3.1.4 Logic Analyzer

Sampling Rate

The sampling rate of a logic analyzer defines how frequently it captures the state of the digital input signals. The sampling rate should be higher than the input digital signal itself for accurate performance. A higher sampling rate allows for more detailed observation of fast digital transitions and events.

HAL Drivers

The Hardware Abstraction Layer (HAL) drivers provide a simplified interface for managing the peripherals of the microcontroller, making it easier to configure hardware components like GPIO, timers, and interrupts. For the logic analyzer, the GPIO HAL driver is essential for configuring the input channels to capture digital signals, ensuring they are correctly set up for input with appropriate voltage levels (3.3V or 5V). Additionally, the timer HAL drivers play a crucial role in defining the sampling rate, ensuring that digital signals are sampled at regular intervals based upon user-defined settings.

Direct Memory Access (DMA)

DMA is vital for efficiently handling high speed data transfers in the logic analyzer. Instead of relying on the CPU to manage every sample, DMA allows digital signals captured by the GPIO to be automatically transferred to a memory buffer without interrupting the microcontroller's main tasks. This enables the system to capture high-frequency signals across multiple channels without data loss. DMA also allows for techniques like double-buffering, where one memory buffer is being filled while the other is processed, ensuring continuous data capture. This signicantly reduces CPU overhead, allowing the microcontroller to focus on other critical tasks such as managing the oscilloscope and AWG functionalities.

3.1.5 GUI

GUI is an abbreviation for Graphical User Interface. A GUI is the interface that a user interacts with to control the device. Accordingly, a GUI must have some set of input methods such as buttons, textboxes, and checkboxes to indicate allowable inputs, as well as output methods such as text, graphs, and sound to provide feedback to the user on the result of their inputted actions. For our purposes, the GUI must be able to control the oscilloscope, the arbitrary waveform generator, and the logic analyzer from the host system by communicating with the microcontroller over USB connection. The GUI will allow users to interact with waveform parameters and view their associated plots in real-time.

3.2 Problem Analysis

This problem mainly concerned the functionality of combining the three systems together and keeping the cost of the system low. The team needed to ensure there were no conflicts within each system and fix any hardware or software issues that the prior teams left unresolved. Lastly, the GUI needed to be redesigned to accommodate the additional functionality required.

3.3 Our Approach

3.3.1 MCU

The prior teams' choice of MCU was the STM32-F303RE. This microprocessor is equipped with a 72MHz ARM Cortex-M4 CPU, 512kB flash, 80kB RAM, two 12-bit DACs, and four 5MS/s ADCs. The team who worked on the oscilloscope noted that the MCU would need more than 64kB RAM to store data from the ADC. This left only 16kB at best for the DACs needed for the waveform generator and for the logic analyzer. Additionally, the F303 has a single 12-bit DMA to reduce the impact of the ADC and DAC on the CPU. Considering prior projects only had to handle a single device, it may be wise to double this for implementing both of these. As such, our team was concerned about the performance of this MCU with all three systems implemented.

We decided instead that our device will instead be based on the STM32-H563VGT6. This processor features a much faster 250MHz ARM Cortex-M33 CPU, four times the flash memory, 640kB RAM, the same DACs, two 5MS/s ADCs, and two 12-bit DMAs. With these

improved specifications we saw fewer constraints related to the performance of the MCU when running all three functions.

3.3.2 Hardware

To ensure the quality and accuracy of the signals being captured, we are using an analog front-end that includes attenuators, amplifiers, and low-pass filters. These components help condition the incoming signals, making sure they are within the proper voltage range and free from excessive noise. By leveraging the MCU's DMA channels, we can efficiently transfer the data from the ADCs to memory without overloading the CPU, allowing for smooth real-time display. For the AWG, we are utilizing the two 12-bit DACs built into the STM32-H563VGT6 to generate precise, customizable waveforms, like sine, square, and triangle waves, etc. After the DACs, the signals are fed into op-amps to amplify and offset the voltage to the levels required for external circuits.

There were some minor problems within the current systems of the oscilloscope and AWG which we have addressed. The first involved the frequency response of the oscilloscope. At higher magnitudes (dB), the frequency dropped off too quickly. We believed this to be due to the opamp used in the gain amplifier and DC offset. We replaced it with an opamp with a higher gain bandwidth product to resolve this problem. The other problem lies within cost. A number of components chosen by the previous teams were unnecessarily expensive. To lower these costs, we decided to replace them in our design with equivalent parts that were cheaper.

3.3.3 GUI

The main purpose of a GUI is to present the possible actions of a system to the user in a manner that makes the functionality of the system intuitive and easy-to-use for the user. Each of the previous projects had an individual GUI that controlled their respective components, so the main task of this project is to create a GUI that integrates all three parts. Our GUI is designed using Python and the following libraries:

- PyQt6 – GUI application toolkit for graphics and interaction
- PyQtGraph6 – Fast, live, dynamic waveform plotting
- PySerial – USB serial communication
- NumPy – Data processing into arrays

Following past design teams' implementations, we are using the same libraries for easier integration into one project. The PyQt6 library provides us with the necessary graphics and interactive user-input features that make up the interface.

For live waveform plotting, we utilized PyQtGraph for a few reasons. First, PyQtGraph is based on the higher-level Python programming language, which allows the GUI software to be more portable across different operating systems. Secondly, PyQtGraph is faster than other graphing libraries such as Matplotlib, which is vitally important for live, real-time graphing, such as when plotting oscilloscope measurements. Finally, all previous teams have used PyQtGraph as well, which allowed us to leverage components of their GUIs for our design.

To process and display the waveforms on live graphs, the data must be compatible with PyQtGraph's data processing methods. We utilized the NumPy Python library to process the waveform data into arrays that we can manipulate and that PyQtGraph can plot. Like previous teams, we used PySerial to facilitate the bidirectional communication between the GUI on the host computer and the MCU in the device.

Below is the original planned prototype for our GUI to view live waveforms, which ultimately formed the basis of our final design in this project.

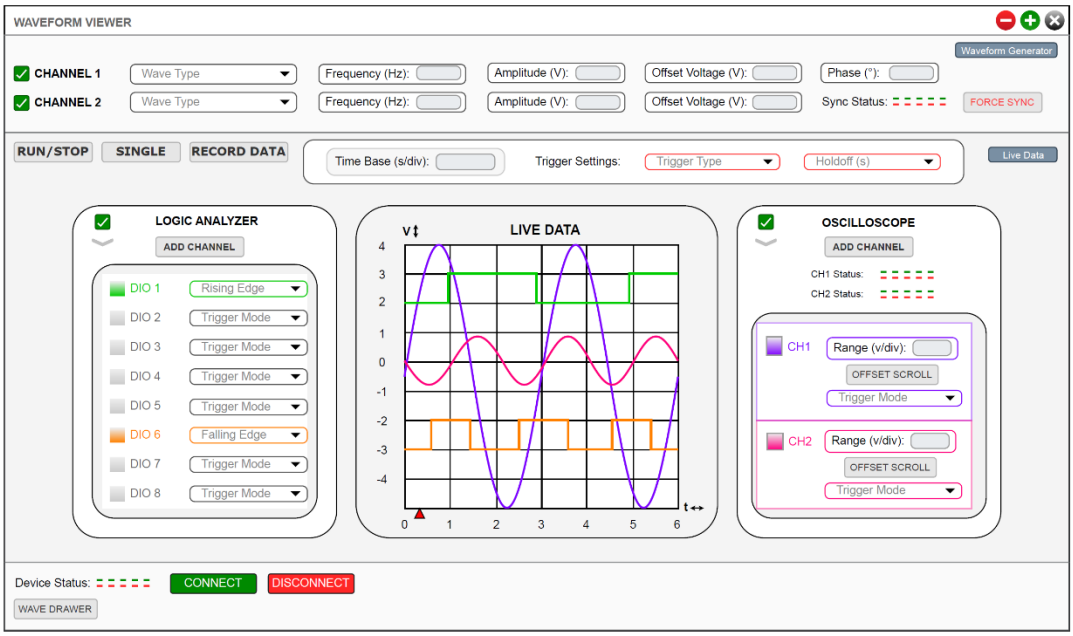


Figure 1: GUI Prototype

The GUI prototype integrates the designs and functions of previous teams' projects to operate cohesively in one main application. The top of the window houses the controls for the arbitrary waveform generator, while the oscilloscope and the logic analyzer are controlled and viewed in the main part of the window. The GUI prototype was initially designed with the intention of plotting the oscilloscope and the logic analyzer waveforms on one plot when run simultaneously. However, over time it was discovered that separate plots would suffice and perhaps provide easier visualizations for students. Additionally, to make the GUI more straightforward and to avoid unnecessary clutter, it will allow users to deactivate and/or collapse entire functions (waveform generator, oscilloscope, logic analyzer) or any unused channels in those functions. The goal of our GUI has been to provide users with an intuitive and easy to use interface that still delivers the capability and accuracy needed by ECE students.

3.4 Alternative Approaches

In coming to the decision to use the STM32-H563VGT6, we considered other STM32 MCUs, particularly the G474RE, H723VGT6, and H533RET6. The G474RE was dropped from consideration due to the need to interleave its ADCs to reach the required 5MS/s. While the

H723VGT6 has the fastest CPU by far, as well as excellent analog and digital components, the RAM is somewhat lacking compared to our choice MCU and it is roughly double in price. The H533RET6 was also considered due to the availability of relatively inexpensive development boards. However, the H563VGT6 proved to be the most well-rounded MCU choice, and we were able to source development boards for testing purposes. The final PCB design does not rely on the development board but has the H563VGT6 integrated onto the PCB itself.

3.5 Project Requirements Specification

3.5.1 Mission Requirements

The project shall develop an affordable USB multi-function lab instrument with two oscilloscope channels, two arbitrary waveform generators, and eight or sixteen logic analyzer channels to satisfy the requirements for the lab exercises of the ECE department. All systems used in this device shall be incorporated into a single custom PCB with the device being controlled via a GUI with communication occurring over USB.

3.5.2 Operational Requirements

Input/Output Requirements

- The device shall have two analog output channels for the AWG.
- The device shall have two analog input channels for the oscilloscope.
- There shall be at least eight logic analyzer channels.

External Interface Requirements

- The device shall be powered using the 5VDC supplied by a standard USB connection.
- The device shall communicate with a host computer over a standard USB connection.

Functional Requirements

<u>Oscilloscope</u>

- The oscilloscope will have a sampling rate of 5MS/s.
- The bandwidth shall be 2MHz.
- The maximum supported frequency shall be 500kHz.
- There shall be various trigger options available.

<u>AWG</u>

- The AWG will have a sampling rate of 5MS/s.
- The AWG will output 10V peak-to-peak.
- The peak load of the AWG will be 20mA.
- There shall be the ability to offset by ±5V.

<u>Logic Analyzer</u>

- The sample rate will be a maximum of 5MS/s.
- The device will support 3.3 and 5V logic.
- The device shall decode protocols such as I2C, SPI, UART, and CAN.

Technology and System-Wide Requirements

- The device should be implemented on a single custom PCB.
- The microcontroller will have at least two 12-bit ADCs and two DACs.
- The device shall be controlled via a GUI on a computer connected over USB.
- The GUI program shall be compatible with Windows, Linux, and macOS.
- The total cost of the device should not exceed $50 per unit for 1000 units.

4. System Design

4.1 Functional Decomposition

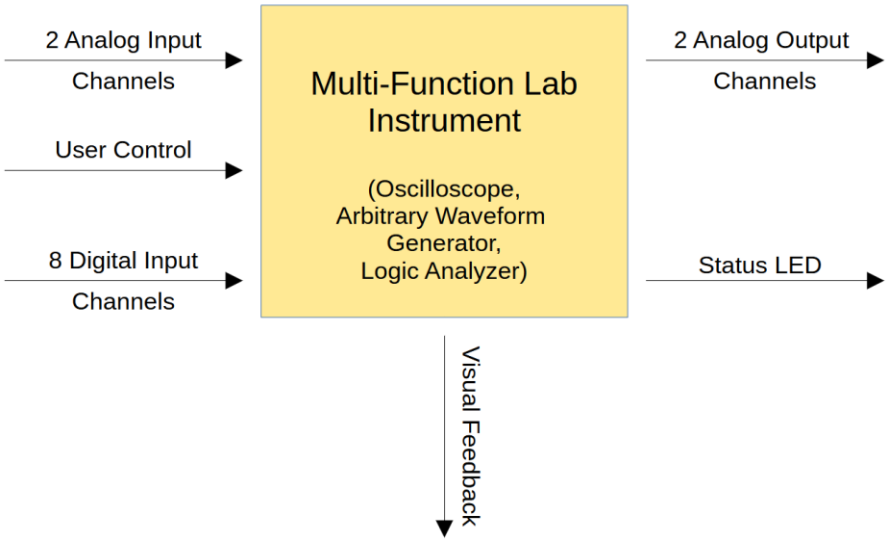


Figure 2: Level 0 Decomposition

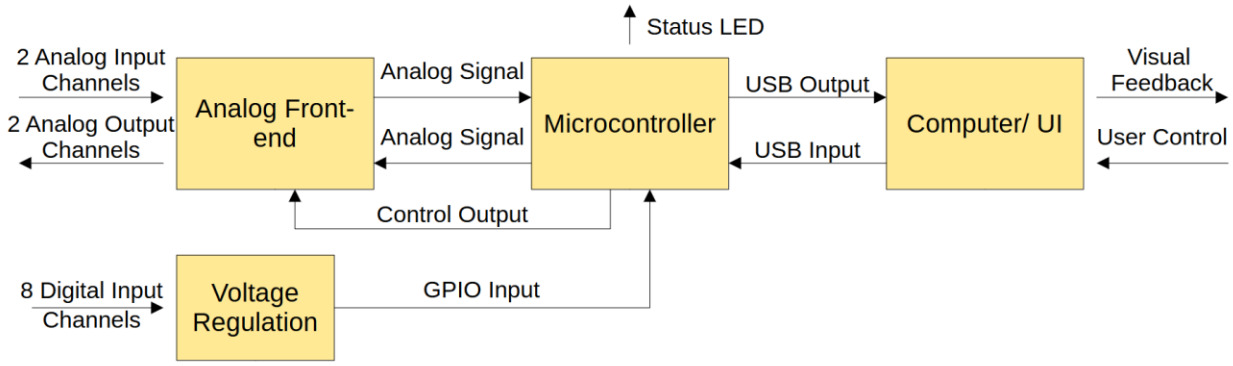


Figure 3: Level 1 Decomposition

4.2 Physical Architecture

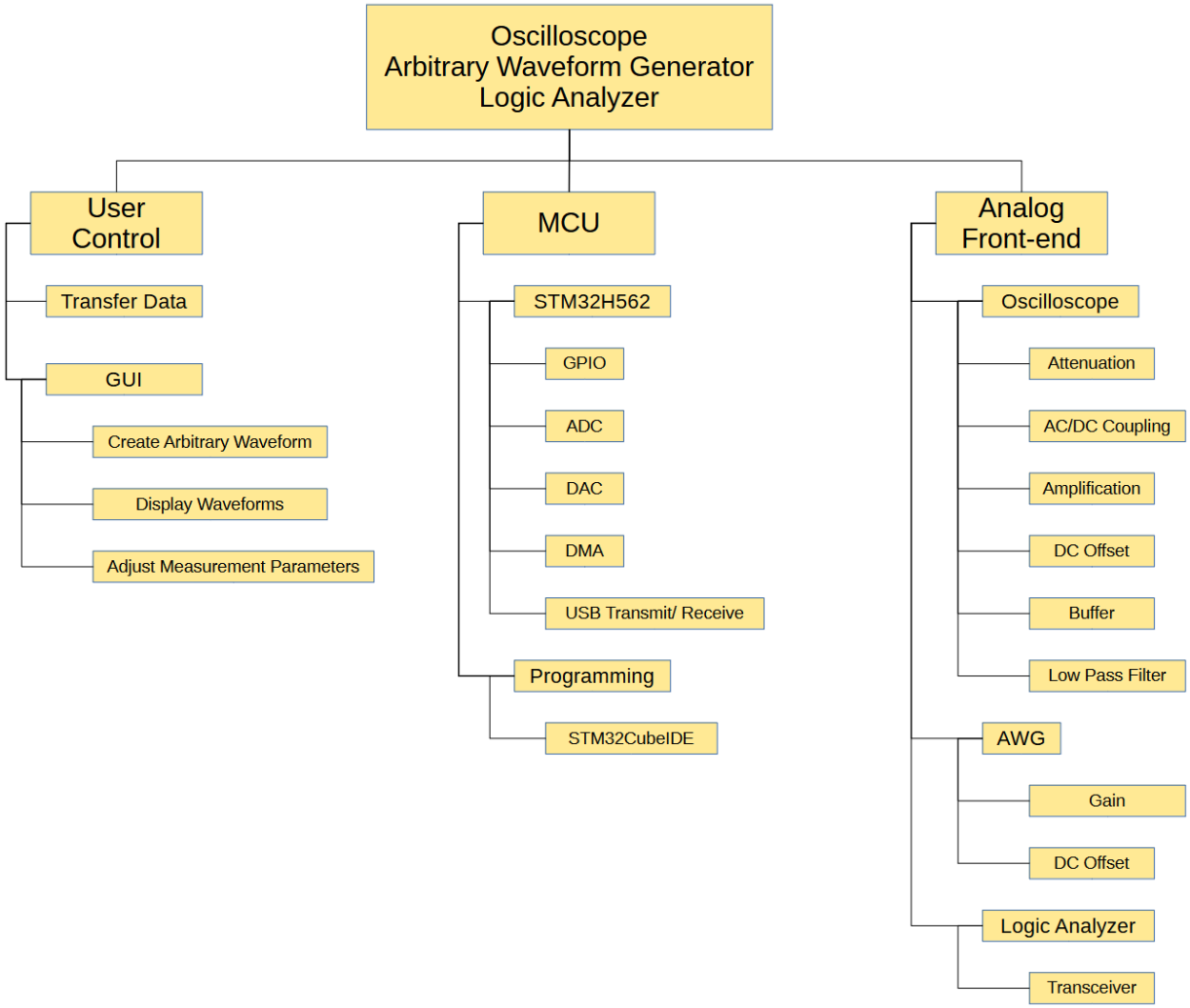


Figure 4: Physical Architecture

4.3 System Architecture

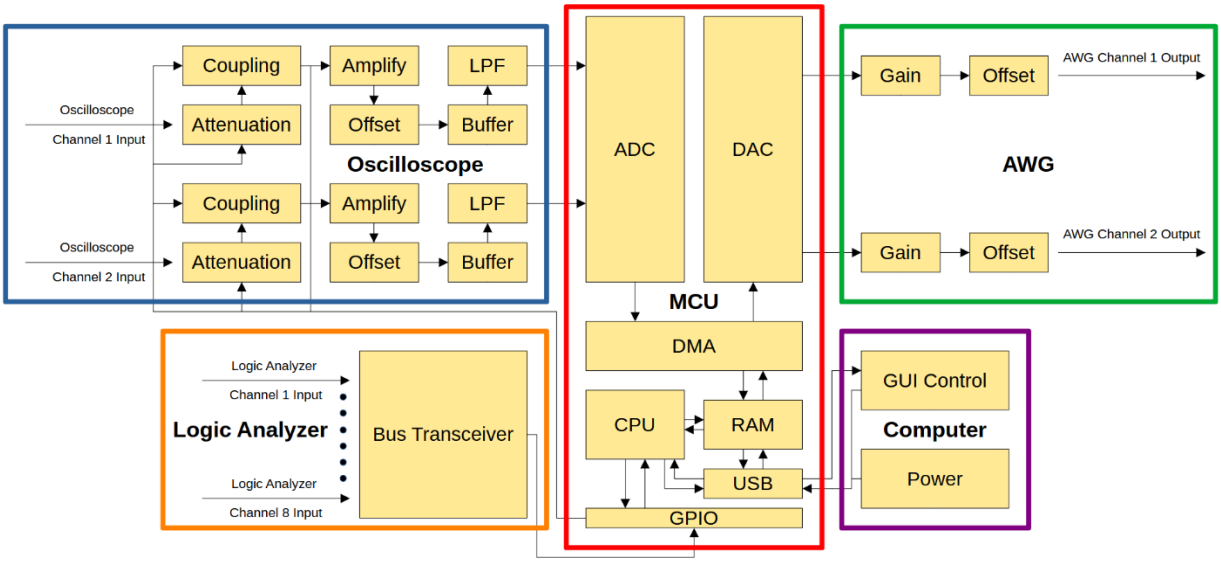


Figure 5: System Architecture

4.4 Microcontroller Details and Design

The microcontroller selected was the STM32H563VGT6. Any microcontroller within the H562/ H563 subfamilies can be substituted for this microcontroller. Several features of the microcontroller are used in this project including USB, GPDMAs, ADCs, DACs, Timers, and GPIO.

4.4.1 Program

The oscilloscope and logic analyzer relies on a series of states to determine what process the device is currently in. These are checked sequentially before determining which data to send, of which the process loops. Considering the waveform generator does not need to send any data back to the computer, it does not have a series of states nor is it included in this sequential process occurring in the main while loop. This sequential process is depicted in Fig 6.

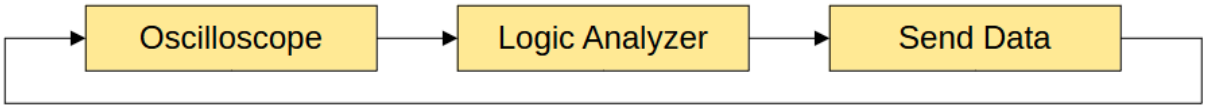


Figure 6: Main Process

4.4.2 Oscilloscope

The oscilloscope has the following states: noTrigger, preTrigger, triggerState, and postTrigger. The logic within the noTrigger state is depicted in the flow chart in Fig 7. The process begins by starting the ADC and the timer used for sampling the ADC. The check to determine if the DMA buffer is filled is fulfilled by a callback function which is triggered by an

interrupt. A timer runs independently of this to determine if 100 ADC samples have elapsed. If this is true it raises a flag to transmit the ADC data. A pointer is used to signify where in the buffer the data is being read from. When the DMA buffer is filled, the callback function stops the ADC to prevent data from being overwritten and transmits all remaining values. When this is done the process repeats.

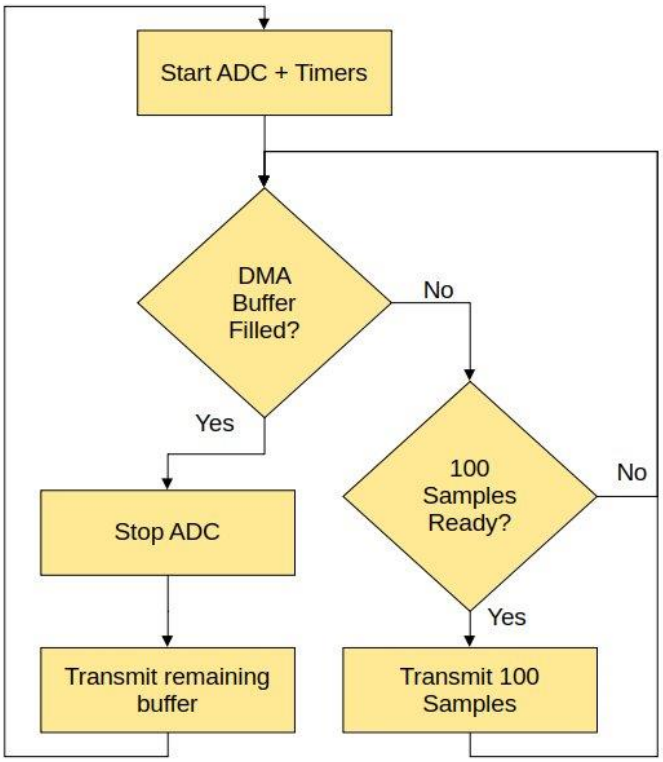


Figure 7: Oscilloscope No Trigger Process

If there is a trigger, the oscilloscope will use the preTrigger, triggerState, and postTrigger states during its runtime. Fig 8. is the flow chart for this process. It begins in the same way with starting the ADC and timers. The trigger detection is done using analog watchdogs. These are configured based on user values and two are needed to determine if the edge is rising or falling. Both watchdogs triggering consecutively indicates that the correct trigger has occurred. A trigger timer is started to sample after the trigger, again being user specified. The interrupt for the trigger timer indicates the end of sampling hence the ADC is stopped, and the entire buffer is transmitted.

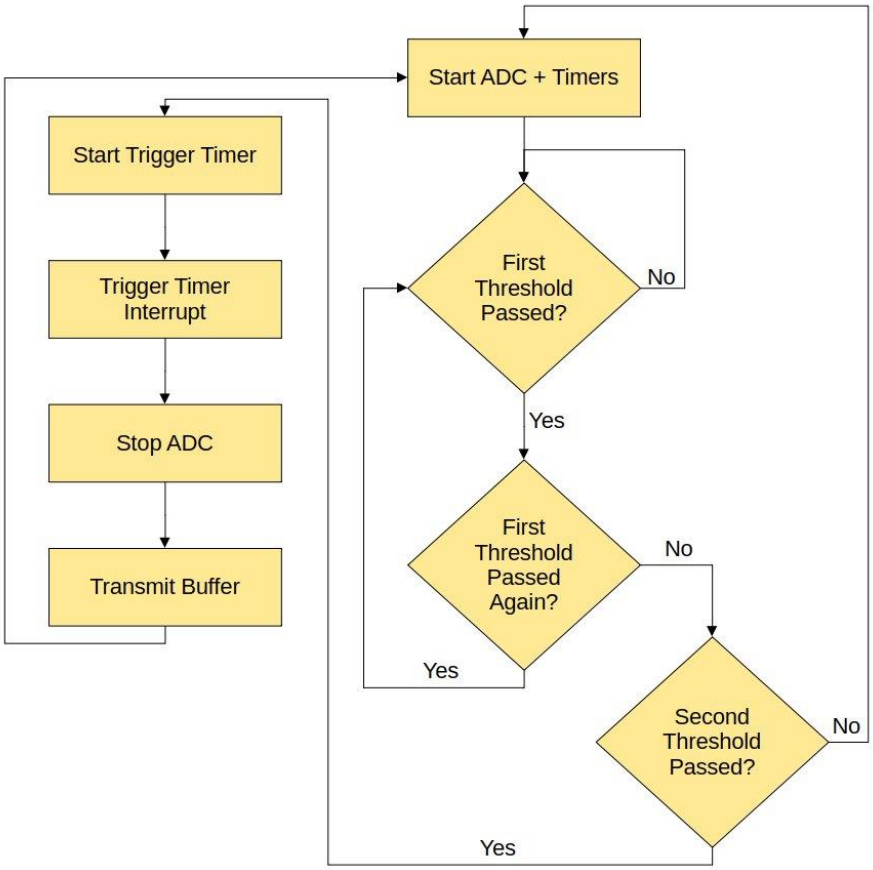


Figure 8: Oscilloscope Trigger Process

The oscilloscope relies on the ADC, timers, GPIO, and GPDMA to function. Both ADCs are being used, specifically the fast channels to allow for the highest sampling rate. Each channel has a buffer which is linked directly to the peripheral via the GPDMA to reduce load on the CPU. The size of the buffer cannot exceed 64K bytes due to a GPDMA limitation. Each ADC has a GPDMA channel. Several timers are used for each channel of the oscilloscope. There are several GPIO outputs to control parameters such as attenuation, offset, and amplification. In addition to these GPIO outputs, two timers acting as PWM serve as outputs for the analog frontend. For the rest of the timers, one is used to trigger ADC conversions, another which runs at $1/100^{th}$ of the timer used to trigger conversions to indicate when 100 samples are ready. The final timer for the ADC is used to determine how long after a trigger the ADC should continue sampling.

4.4.3 Logic Analyzer

The logic behind the logic analyzer is similar to that of the ADC with a trigger as seen in Fig 9. Unlike the ADC, the GPIO cannot feed directly into a GPDMA thus it must be sampled, checked, and stored via an interrupt with a sampling timer. The GPIO timer interrupt occurs, and the program checks to see if the current GPIO value for the user specified pin is different from the prior value of that same pin. If it is, a trigger timer is started to sample after the trigger

occurs. That trigger timer interrupts, and the sampling timer is stopped. The buffer is transmitted all at once as the transmit flag is raised.

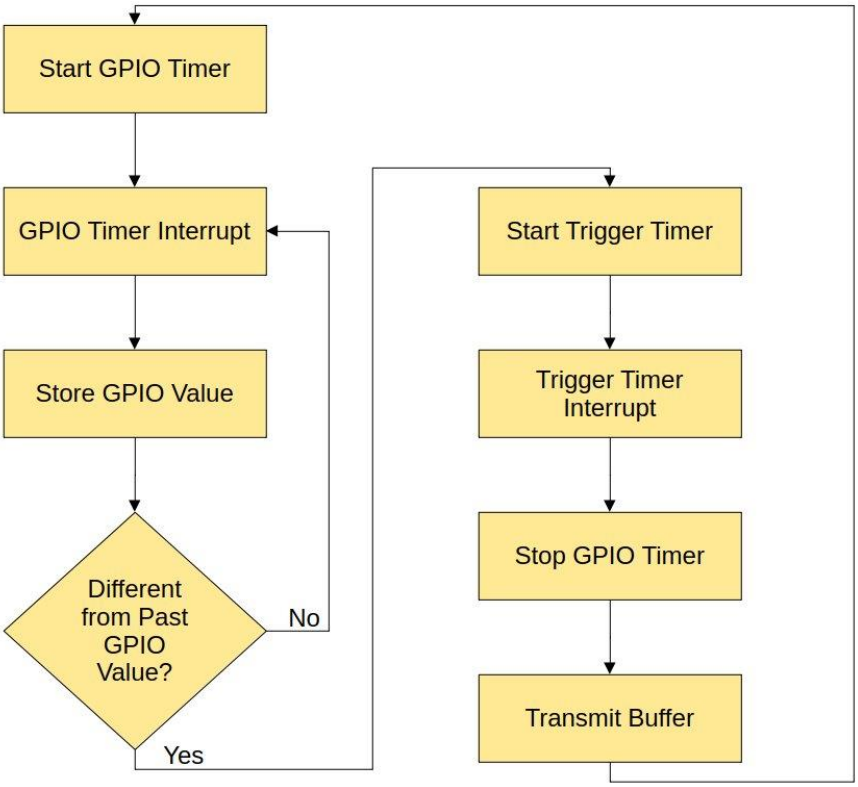


Figure 9: Logic Analyzer Trigger Process

The logic analyzer is rather lightweight for peripherals. It only uses two timers. One for the sampling and another for the trigger. It uses 16 GPIO inputs on a single GPIO bus for data collection.

4.4.4 Waveform Generator

The waveform generator is simple when it comes to logic. Considering it is solely an output; it only needs to be set once when the user desires. Thus the only time the waveform generator is addressed is during the reception of USB data, not in the main process. The waveform generator uses both DAC channels as outputs, two timers for PWM, and two additional timers as triggers for the DAC channels.

4.4.5 USB Communication

The USB communicates bi-directionally. It is constantly in the state of sending data which pauses in the event that data has been received. All data is sent or received in packets of 64 bytes. The breakdown of data sent is shown in Fig. 9. This is the data packet constantly transmitted over USB. For simplicity it is all combined into a single packet rather than dealing

with separate logic analyzer and oscilloscope packets. A simple way to determine whether the data from either the logic analyzer or oscilloscope is ready to be sent is by checking to see if the buffer position is less than 30000. If it isnt then that indicates the data is not ready and it is skipped over in the GUI. For example, if you are reading data from the oscilloscope and also waiting for a trigger with the logic analyzer, the oscilloscope data will have adcpos between 0 and 30000 while the logicpos will be at 40000, indicating that the data from the logic buffer can be discarded. If the whitespace bytes are not present, there may be issues with the computer decoding the transmission.

Offset	Name	Type	Value	Length	Note
0	Packet Type	u8	5	1	Indicates packet type
1	adcpos	u16	0-40000	2	Indicates position in the ADC buffer
3	logicpos	u16	0-40000	2	Indicates position in the logic buffer
5	oscch1	u16[]	x	16	8 values from ADC1 starting at adcpos
21	oscch2	u16[]	x	16	8 values from ADC2 starting at adcpos
37	logicpos	u16[]	x	16	8 logic analyzer values starting at logicpos
53	whitespace	u8[]	x	11	whitespace to make it 64 bytes in length

Figure 10: Data Packet

Carried over from prior waveform generator code are the acknowledgement packet and handshake packet. The handshake packet is redundant and not used though it is still present in the code. The purpose of the handshake packet is to keep the connection between the computer and microcontroller alive and prevent a timeout. The acknowledgement packet is sent every time a command is received to ensure that said command was received. The structure of the acknowledgement packet is shown in Fig 10.

Offset	Name	Type	Value	Length	Note
0	Packet Type	u8	2	1	Indicates packet type
1	Magic String	u8	"STM32AWG"	8	Magic String, no null terminator
9	Padding	u8	0	55	Zero Padded Bytes

Figure 11: Ack Packet

The remaining packets are for data being sent to the microcontroller for the waveform generator, oscilloscope, and logic analyzer. Fig. 11, 12, and 13 are the packets for the waveform generator, oscilloscope, and logic analyzer. Each packet again has a length of 64 bytes with any extra taken up with zero padded bytes. These packets contain all the information needed to modify the GPIO, timers, and control the logic needed to make these peripherals function.

Offset	Name	Type	Value	Length	Note
0	Packet Type	u8	1	1	Indicates packet type
1	Channel	u8	x	1	Channel to configure, 0 or 1
2	Gain	u8	x	1	Gain to use, 0 = High, 1 = Low
3	PSC	u16	x	2	Prescaler Value for channel Timer
5	ARR	u16	x	2	Divider Value for channel Timer
7	CCR_Offset	u16	x	2	CCR Register value for PWM Offset Gen.
9	numSamples	u16	x	2	Number of Samples
11	phaseARR	u16	x	2	Phase offset of wave in clock cycles
13	Padding	u8[]	{0, 0, …}	51	Zeroed Padding Bytes
64	Samples	u16[]	x	v	The samples for the wave. Length is numSamples*2 rounded up to nearest multiple of 128

Figure 12: Waveform Generator Packet

Offset	Name	Type	Value	Length	Note
0	Packet Type	u8	2	1	Indicates packet type
1	Channel	u8	0-1	1	0 or 1 to select channel
2	adcmode	u8	0-11	1	12 options to set ADC sampling rate
3	triggermode	u8	0-2	1	Indicates rising, falling, or no trigger
4	triggerval	u16	0-4095	2	What value the trigger should occur
6	triggerPeriod	u16	x	2	Period for trigger timer
8	triggerPrescaler	u8	0-7	1	Prescaler for trigger timer
9	sampletime	u8	0-7	1	8 options to change the sampling time
10	offset	u8	0-1	1	AC/ DC offset
11	attenutation	u8	0-1	1	Attenuator
12	amp10	u8	0-1	1	1:10 AMP
13	amp5	u8	0-1	1	1:5 AMP
14	amp2_5	u8	0-1	1	1:2.5 AMP
15	amp1	u8	0-1	1	1:1 AMP

Figure 13: Oscilloscope Packet

Offset	Name	Type	Value	Length	Note
0	Packet Type	u8	3	1	
1	Control	u8	0-1	1	Whether to enable or disable the logic analyzer
2	triggerPin	u16	0-65536	2	What pin to watch for trigger (2^pin#)
4	triggeredge	u16	0-1	2	Rising or falling edge for the trigger
6	period16	u16	x	2	trigger timer period
8	prescaler	u16	x	2	trigger timer prescaler
10	period32	u32	x	4	sampling timer period

Figure 14: Logic Analyzer Packet

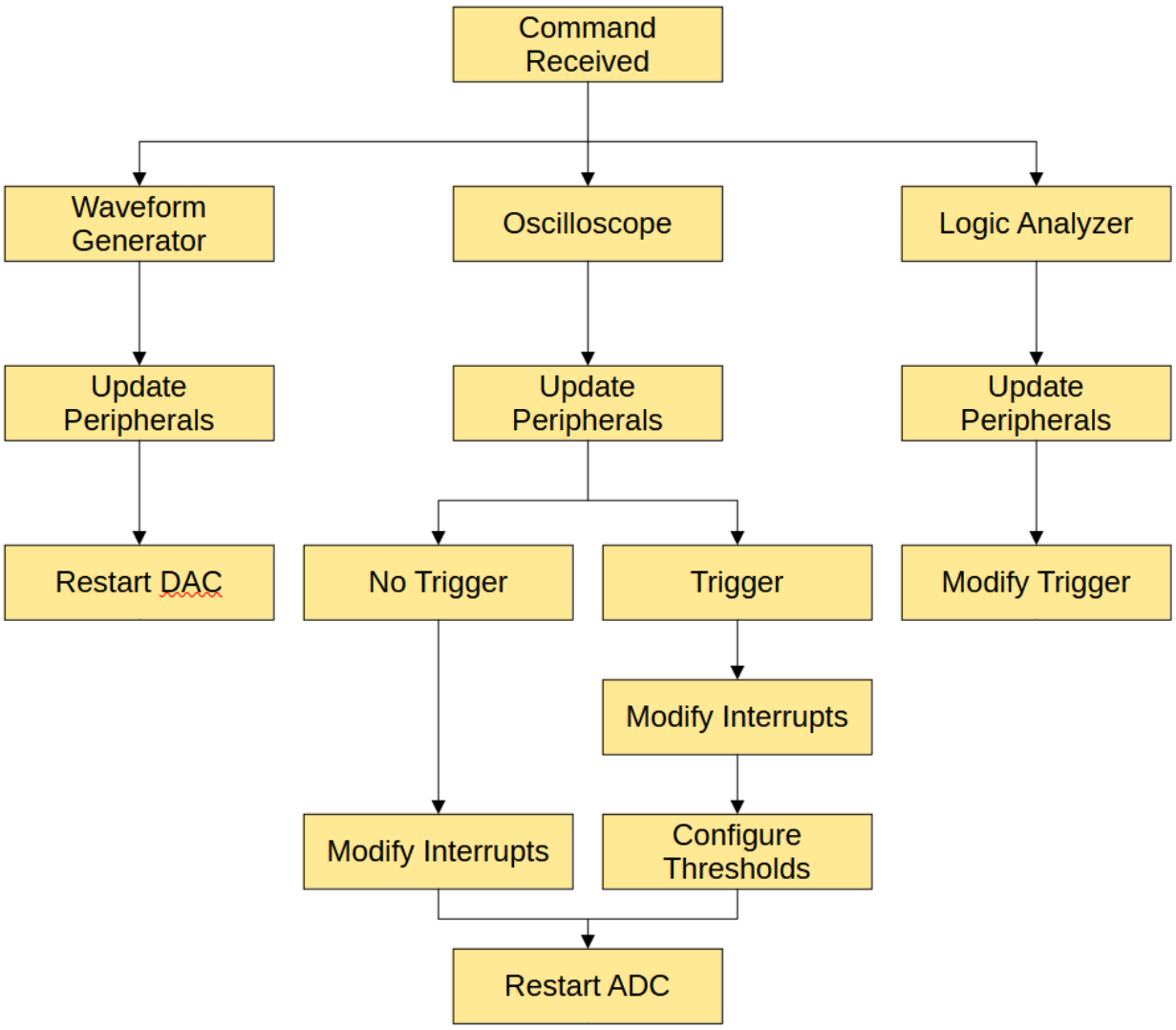


Figure 15: USB Command Tree

The command tree in Fig. 14 is what happens when the microcontroller receives a command. It first determines which part of the device needs to be updated. Then peripherals are updated such as GPIO, timers, and various variables for triggers and other logic.

4.4.6 Microcontroller Diagram

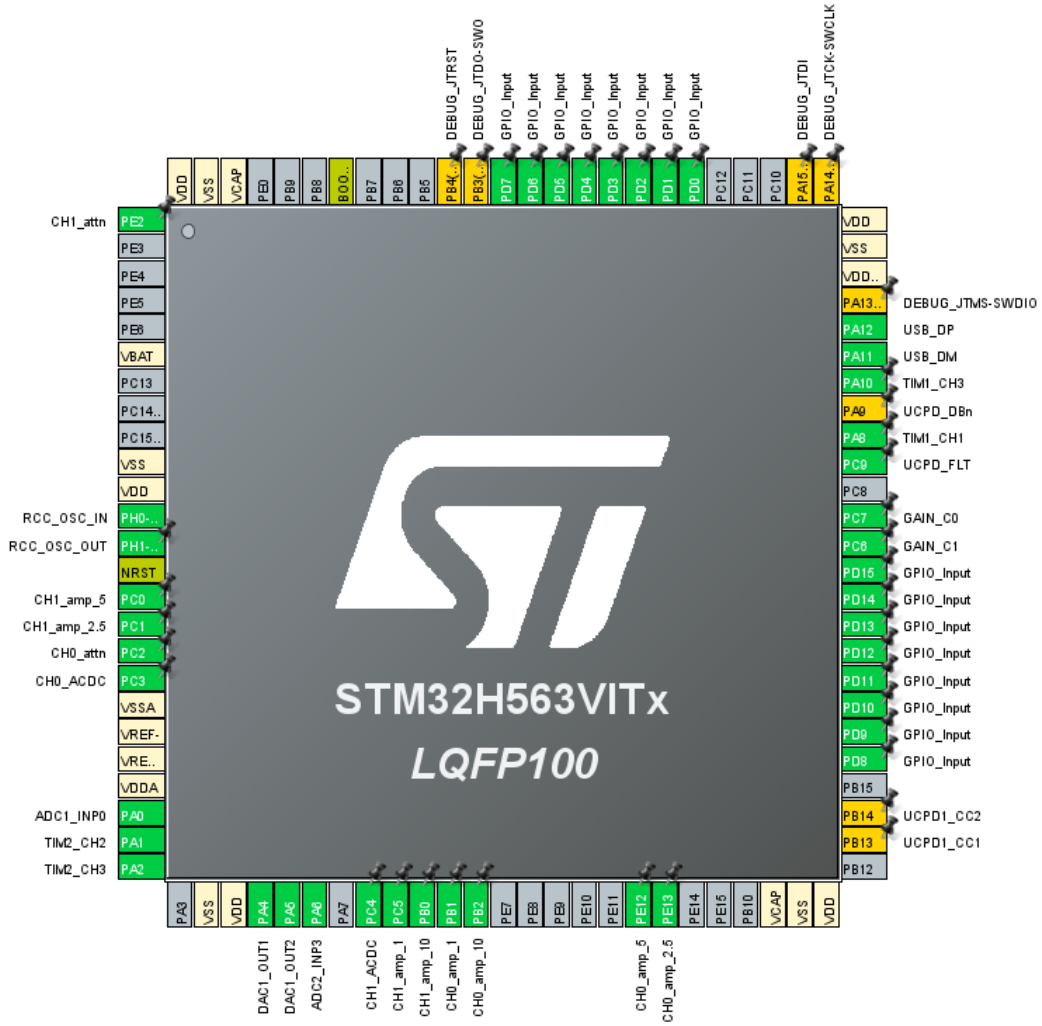


Figure 16: Microcontroller Diagram

Figure 15 shows the microcontroller pinout diagram for the H563VIT6 present on the latest PCB iteration. The 16 GPIO Inputs are all for the logic analyzer as those must be on a single GPIO bus (in this case GPIO D). In the event of modification, some pins can be easily changed such as the various GPIO outputs while others like the DAC outputs cannot be changed at all. The debug, USB, timer, ADC, DAC, and RCC GPIO all need careful consideration if they will be changed.

4.4.7 MCU Recommendations

Going forward it would be best to switch microcontrollers. The problems faced with the H5 family are expanded upon later. The H7 family would provide more processing power and slightly faster ADCs. This would solve the triggering issue. Furthermore any H7 MCU supports the classic USB firmware which the H5 series lacked.

4.5 PCB Design

4.5.1 Layout Design

For a physical implementation, the previous schematics were implemented into a PCB layout using KiCad. Two prototype PCBs were ordered as a part of this project, but they share similar design characteristics. A trace width of 0.2mm was used for most digital traces, and 0.4mm was used for most analog or power traces. Due to its complexity, the PCB was routed as a four-layer board with one ground layer. Additionally, we implemented a separate analog and digital ground on this layer to reduce noise. Unfortunately, even with a four-layer board, issues with the routing nonetheless precluded adding a dedicated power layer. In future designs, it would be recommended to move to an eight-layer construction to mitigate these issues and reduce noise.

The two PCB layouts, as mentioned, share similar design characteristics. We created the second layout to correct certain issues that prevented the first boards from working without modification; namely, errors in the USB design, clock crystal, inductors, and component orientation. Additionally, the second layout has many routing optimizations compared to the first, and four mounting holes instead of only two.

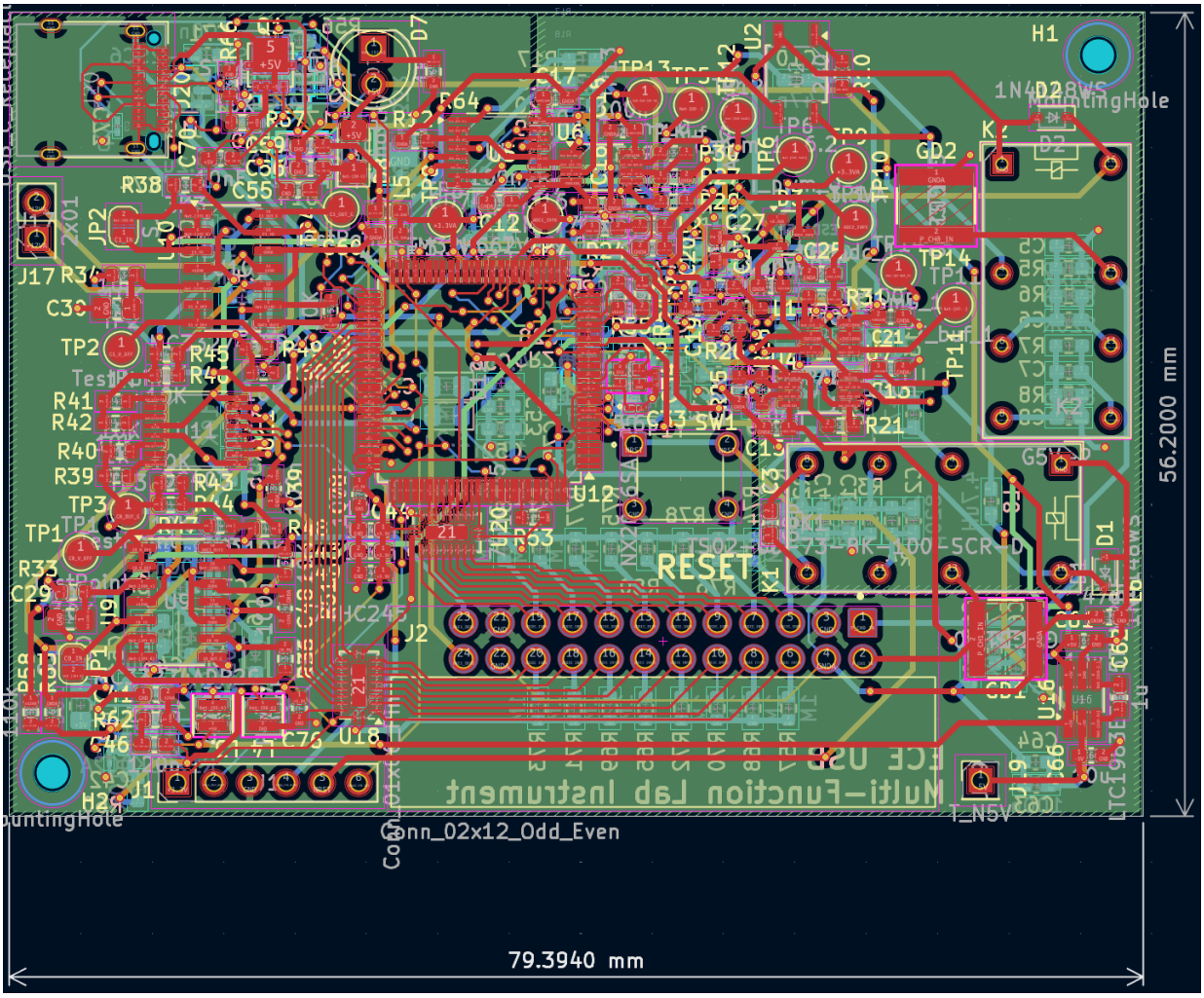


Figure 17: PCB V1

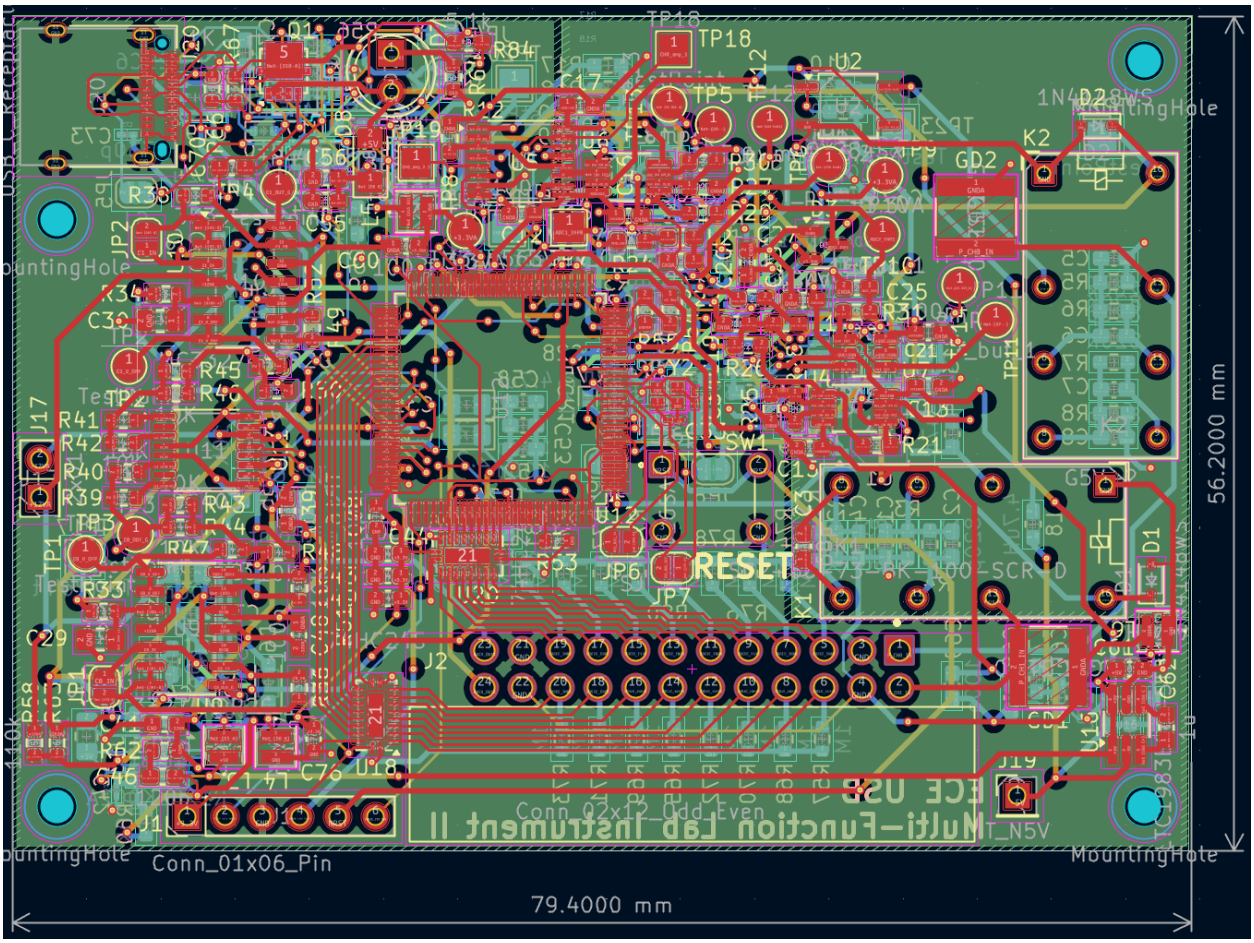


Figure 18: PCB V2

4.5.2 PCB Assembly

Both PCB prototypes were ordered through JLCPCB for manufacturing, since they offered assembly services for ease of prototyping. Nonetheless, not all components were stocked by JLCPCB, so a parts order needed to be placed through DigiKey as well and the missing components soldered by hand.

In the process of assembly, we determined that different soldering methods work better for different components, depending on their packaging style. We started by using exclusively solder paste and heating it with an electric heat gun, however this produced a large number of solder balls due to the composition of the paste. Without the use of a stencil, it was also difficult to move the paste evenly onto the pads such that bridges between pads did not occur, especially for smaller pads that were close together.

This method was still the easiest for some of the components. However, through the soldering process, we found that it was often better to tin the pads with solder, being sure to use a

liberal application of flux. Once its pads had enough solder, but not too much, we soldered the component using either a heatgun or soldering iron, again adding more flux to prevent bridging.

Regardless of the soldering method used, cleanup of the board was nonetheless a priority following assembly before testing the finished board. The soldering process leaves a large amount of debris on the board, such as flux residue and solder balls. Because solder balls are conductive, it was especially necessary to search the board for them and remove any that were found to prevent bridging. For this project, this was done by manual inspection and cleaning only; however, an ultrasonic cleaner would be useful for this purpose.

4.6 GUI Details and Design

The primary concern of the GUI was to integrate the separate functionality of the arbitrary waveform generator, oscilloscope, and logic analyzer into one package. As the eventual MCU communication format would take some time to develop, initial development of the GUI focused on creating the interface with internally generated signals.

An early version of the GUI is shown in Figure 18. This version used a strong color scheme to assist in placement of various GUI elements. Channel one of the AWG controls (at the top of the figure in yellow) controls the blue line in the displayed chart. The red dashed line is created by appending a random value within the range of the AWG's channel one's amplitude. This line in Figure 18 displays the change in amplitude over time of channel one. Additionally, the offset of channel one can be changed.

Figure 19: Early Development Iteration of GUI

Figure 19 displays an intermediate stage in development. The logic analyzer graph can be hidden by unchecking the main checkmark in the logic analyzer checkmark. Individual oscilloscope and logic analyzer channels can be enabled or disabled via individual checkmarks. The type, frequency, amplitude, offset, and phase of the AWG channels can be configured, and the resulting waves are shown on the corresponding oscilloscope channels. A pair of hardcoded signals are displayed on the logic analyzer.

Figure 20: Mid Development of GUI

Figure 20 displays the final version of the GUI in oscilloscope mode. The time base is set with a dropdown instead of a textbox to also serve as a way to set the MCU's sample rate. The primary innovation in our GUI over previous projects is now functional: the ability to manipulate the visuals of the oscilloscope channels. Both channels are locked in the horizontal (time) dimension, but the manipulation of the vertical (voltage) dimension allows for a clearer view of specific portions of the waveform and for the user to see waveforms of significantly different amplitudes simultaneously. In Figure 20, channel one displays a small amplitude sine wave, while channel two has a large amplitude sawtooth wave. This is achieved by plotting channel two on a separate plot and overlaying it on top of channel one's plot. The horizontal lines that represent the voltage divisions are not drawn by PyQtGraph, but are placed manually based on the current vertical range of channel one, which places the lines in the same position regardless of the zoom on channel one. This also allows for the user to pan channel one up or down to align the waveform with the voltage division. Channel two's voltage division and position are achieved simply through being overlaid as described before. Figure 21 is the final version of the GUI with all functionalities enabled. The AWG, oscilloscope, and logic analyzer graphs can be hidden and shown independently, although vertical space makes having all three enabled cumbersome.

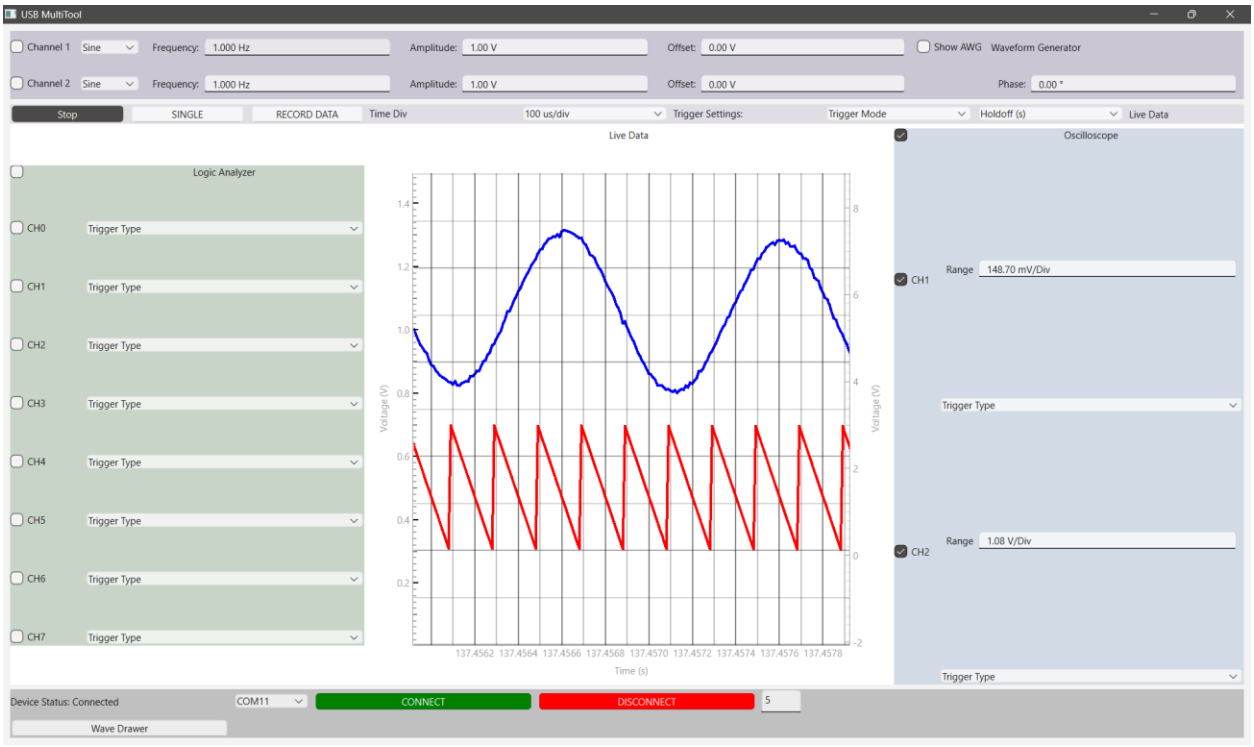


Figure 21: Final GUI

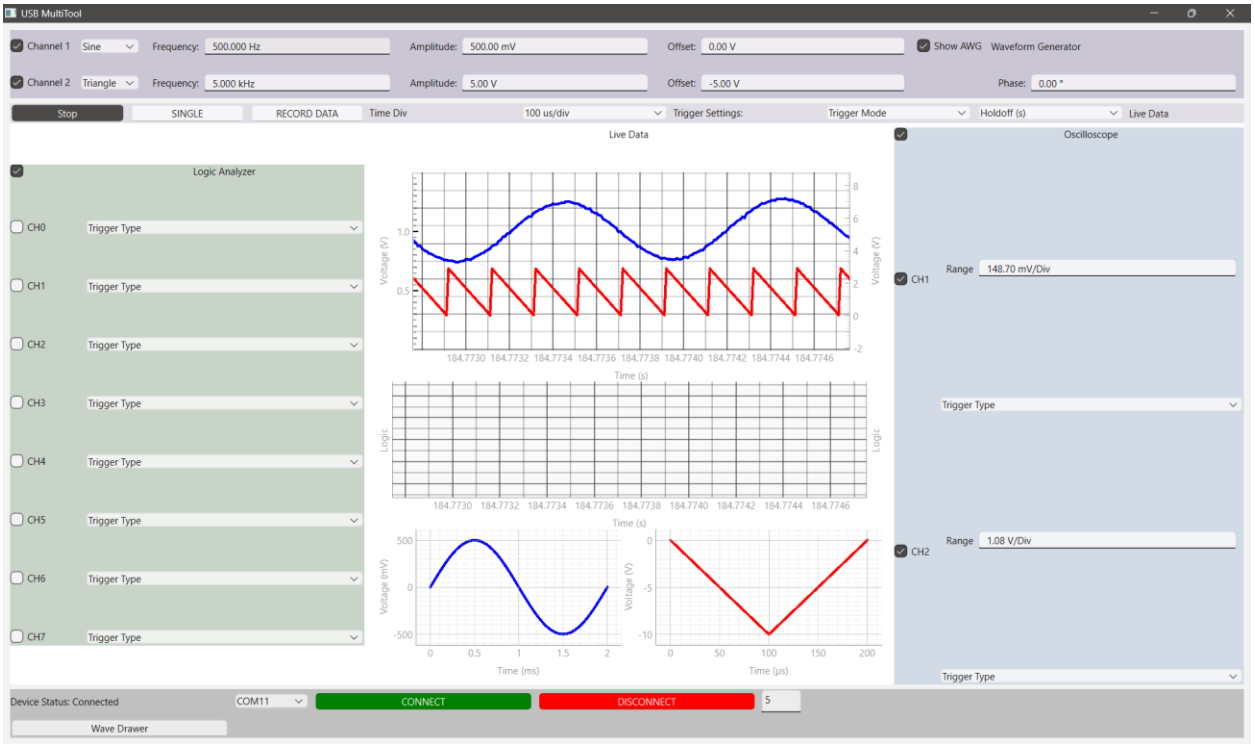


Figure 22: Final GUI with all functions enabled

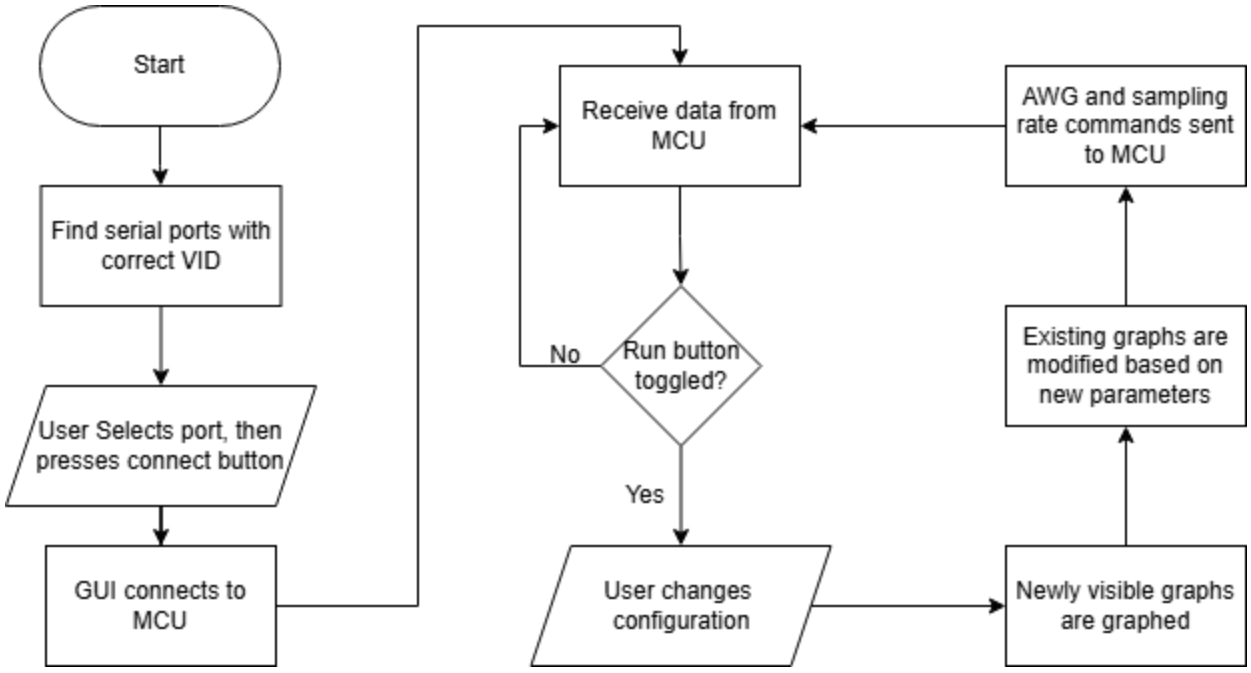


Figure 23: GUI Flow chart

Upon the user connecting the device to a host computer, the GUI identifies the available serial ports with the correct Vendor ID (VID) and adds them to the display. This allows for the user to choose the COM port and connect the host computer to the device, which permits our device to be used at the same time as another STM32 microcontroller project. This connection allows the GUI and MCU to communicate bidirectionally. Additionally, because our GUI continuously reads packets from the MCU, it can display live waveforms and is ready for live analysis upon request from the user. The GUI continuously checks the state of the “Run/Stop” button to determine when to display live data. Once the user selects the “Run” button, the GUI begins displaying the requested live waveform data until the “Stop” button is selected, whereupon the previously graphed data can be inspected in more detail. The live analysis can be resumed by selecting the “Run” button.

The user is able to adjust the waveform parameters or the testing configurations in real-time and the changes will be immediately reflected in the live graphs. This includes features such as enabling/disabling entire functions or channels and updating or modifying the function parameters in either newly created graphs or in already existing graphs. These changes are looped back to the MCU via event callbacks from user-input elements in the GUI, such as buttons and textboxes.

4.7 Case Design

The first case to house the PCB was made with a 3D printer with ABS. The enclosure houses and protects the device from damage. It is comprised of two 3D printed parts and uses friction to hold both parts together along with keeping the PCB snug inside. Ports necessary for the user to operate the device are accessible through openings in the case so that the device does not need to be removed from its case to use it. This was a rushed design and lacks the polish a final case design would have. The first case iteration design is shown in both Fig. 24 and 25. In order for the LED to be visible, thickness of the top cover must be reduced over the LED.

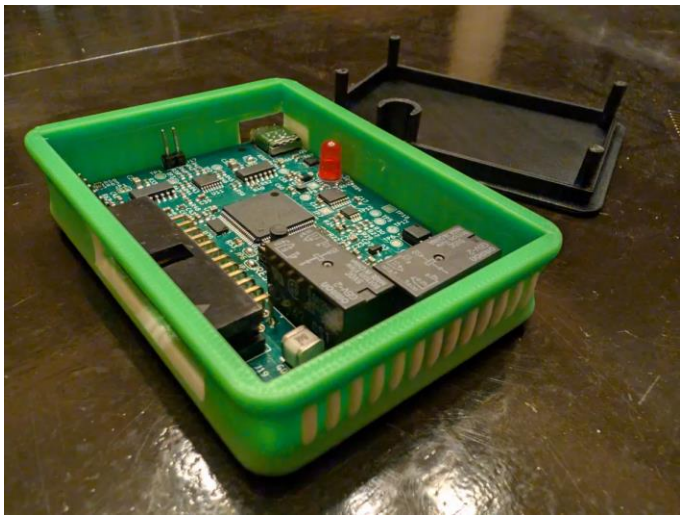


Figure 24: First Case Disassembled

Figure 25: First Case Assembled

The second case design has taken a more realistic approach to the project. Assuming this device would be mass produced; the viability of 3D printing will reduce as more efficient

methods such as injection molding become cheaper, quicker, and stronger. This second design is comprised of three parts rather than two. It follows the same nature of having two halves joined via friction with a third being a plastic liner to protect the PCB from making contact with the aluminum bottom half. This bottom half has been prototyped with the result after the first CNC operation shown in Fig 26. This is simply machined out of 6061 Aluminum. The top part of the case will be produced using injection molding and would likely be a double shot part, allowing for two colors to be used in the plastic top part.

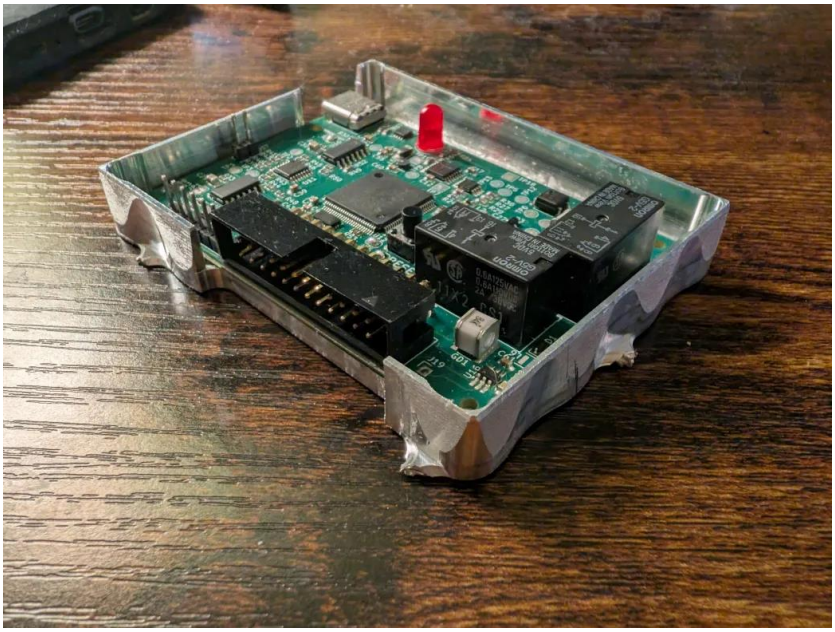


Figure 26: Bottom Half Prototype

Insert more here

A better case design would be aided by a better PCB design. The current PCB design poses a few problems for a case. The first and most difficult problem to work around is the port positioning. A future PCB revision should have both the USBC and 24-pin header protruding by whatever wall thickness is determined to be needed for the case in the end. That way, the ports can sit flush with the case rather than needing large cutouts to allow access. Additionally, larger holes for screw mounts would be good on a new case design. The current holes are only large enough for M2 screws and the placement of these holes could be more thought out.

5. Experimentation and Testing

5.1 Preliminary Experiment

Testing for the device was divided into three parts: the MCU, analog front-end, and the GUI. The GUI and MCU were developed primarily with the use of the development board to

ensure full functionality of all parameters with a reliable testbed. The MCU on the development board was tested to ensure the ADCs, DACs, GPIO all work with the GUI as expected. Finally, we tested the functionality of the PCBs themselves, fixed the problems that arose in our design, and compared them with the performance of the development board.

5.2 Experimentation

5.2.1 MCU Experimentation

Test 1: Confirm Communication with the MCU

The main program running on the MCU can be used to confirm that communication is working properly. This is simply printing out some of the data that is normally sent to the GUI to be displayed. With the terminal output showing the packet type, buffer positions, and some data from each, the test is successful.

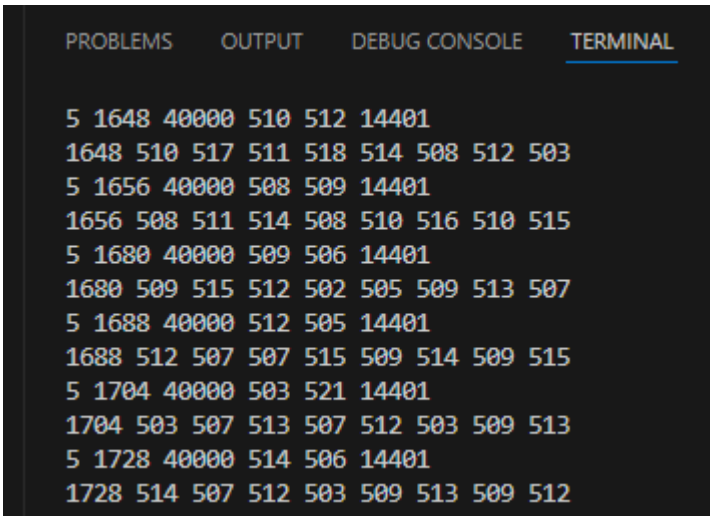


Figure 27: USB Output

Test 2: GPIO / PWM Testing

Using a test program, a PWM signal was output through GPIO. The resulting waveform was recorded with the Analog Discovery 2. This is indicative of what the oscilloscope and waveform generator need to function.

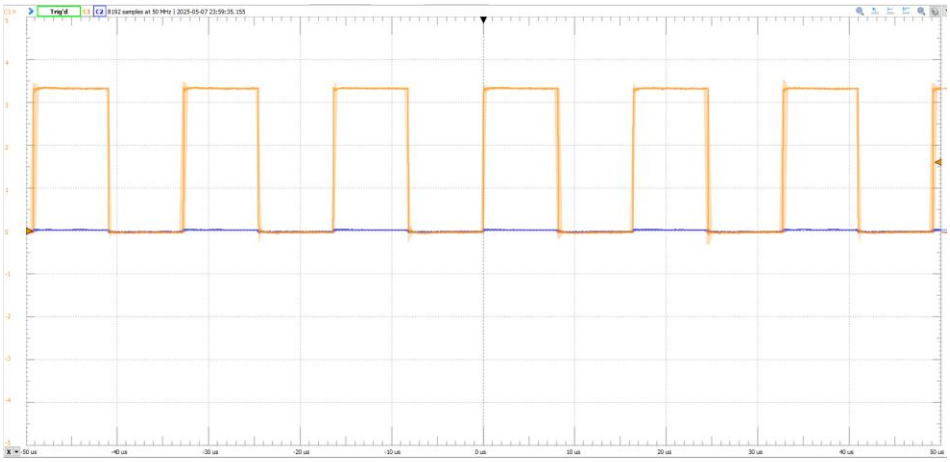


Figure 28: PWM GPIO Output

Test 3: ADCs

The onboard ADCs will be tested to evaluate their functionality and accuracy. A reference voltage will be supplied using a suitable power source, and a test program will display the corresponding voltage detected by the ADCs. The input voltage will be adjusted through the range of the MCU's onboard ADCs, and the ADC readings recorded and analyzed at each step.

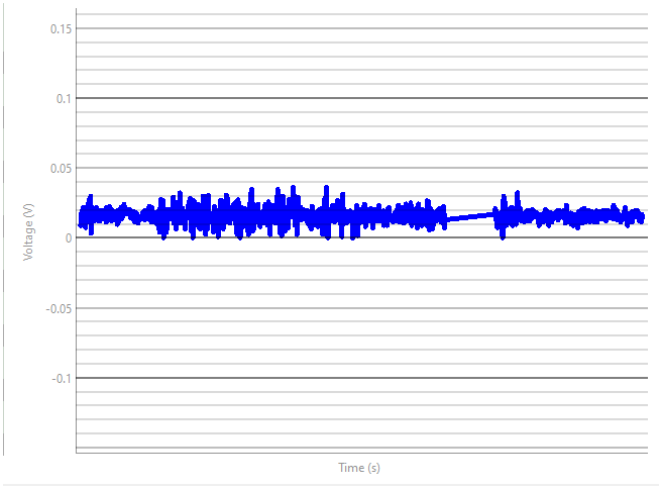


Figure 29: 0V Input

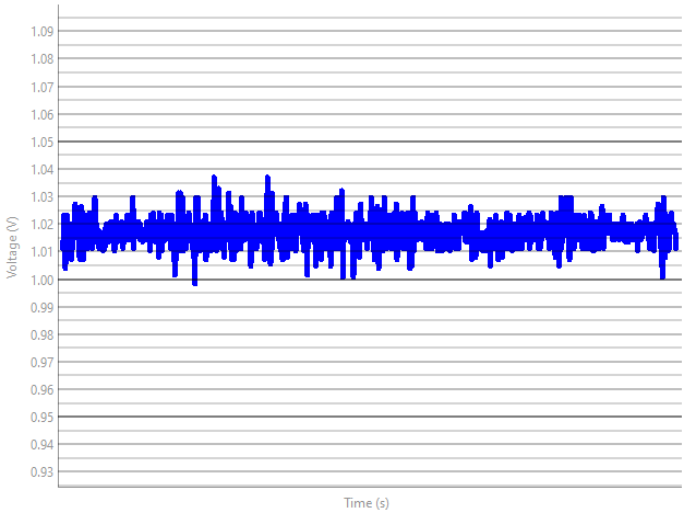


Figure 30: 1V Input

The testing using the Nucleo board shows that with no input the voltage reads between 0 and 0.05V. With 1V input the range is again between 1.0 and 1.05V. At 3V this increases to 2.95V to 3.05V. This testing was done on the Nucleo board as the noise on the PCB is terribly high.

Test 4: DACs

The onboard DACs will be tested to evaluate their functionality and accuracy. A test program will have each DAC sweep through its range of output voltages. The output will be measured using an oscilloscope; the resulting waveform should resemble a sawtooth wave.

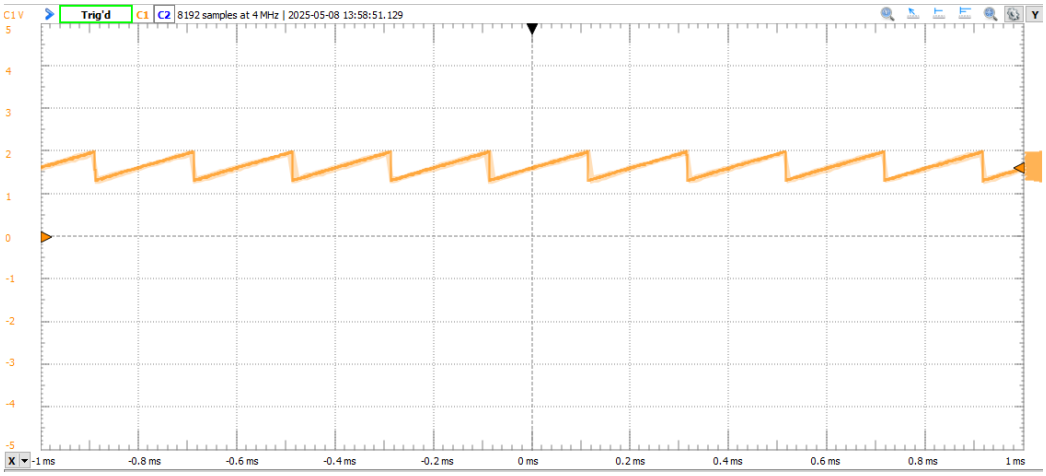


Figure 31: Sawtooth Output

This test could not be conducted on the PCB due to high noise levels. As such, it had to be done on the Nucleo board which can only output from 0 to 3.3V and lacks the amplification necessary.

5.2.2 Analog Front-End Experimentation

Test 1: Amplification Test

This test ensures that low-voltage signals are amplified correctly to levels suitable for the ADC. The amplified output should match the expected value within 1% accuracy, and the amplification should remain consistent across the tested input range.

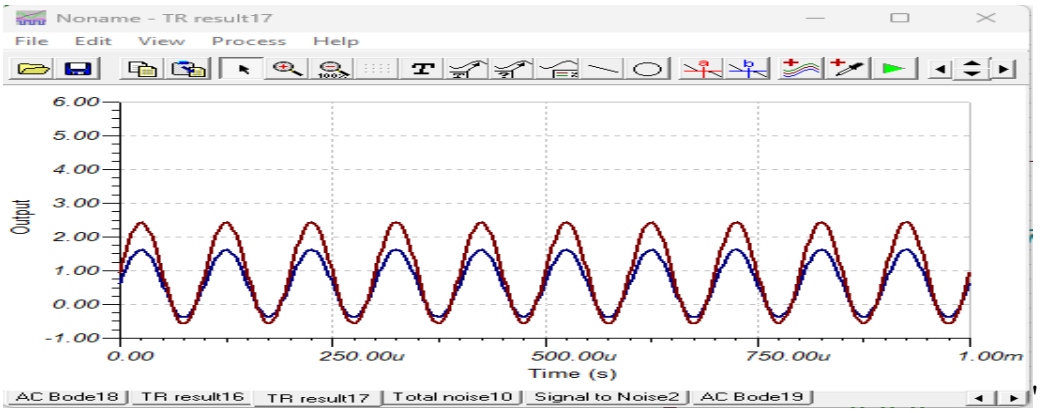


Figure 32: Amplification Test Result

Test 2: Low-Pass Filter Test

This test confirms the low-pass filter removes high frequency electrical noise while passing signals within the desired bandwidth. Signals below the cutoff frequency should pass through with minimal attenuation, while signals above the cutoff should be attenuated by at least 3dB at the cut off, with attenuation increasing with higher frequencies.

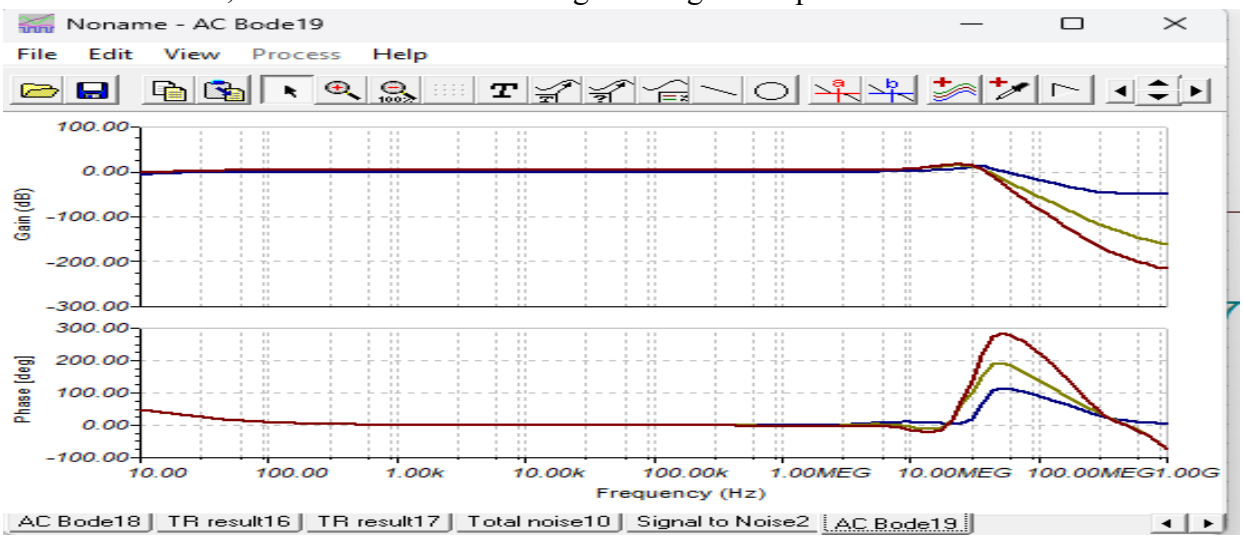


Figure 33: Low-Pass Filter Test Result

Test 3: Total Harmonic Distortion Test

This test will measure the distortion level introduced by the analog front end, particularly the amplifiers and filters. The total harmonic distortion should be within acceptable limits across the range of tested frequencies and amplitudes.

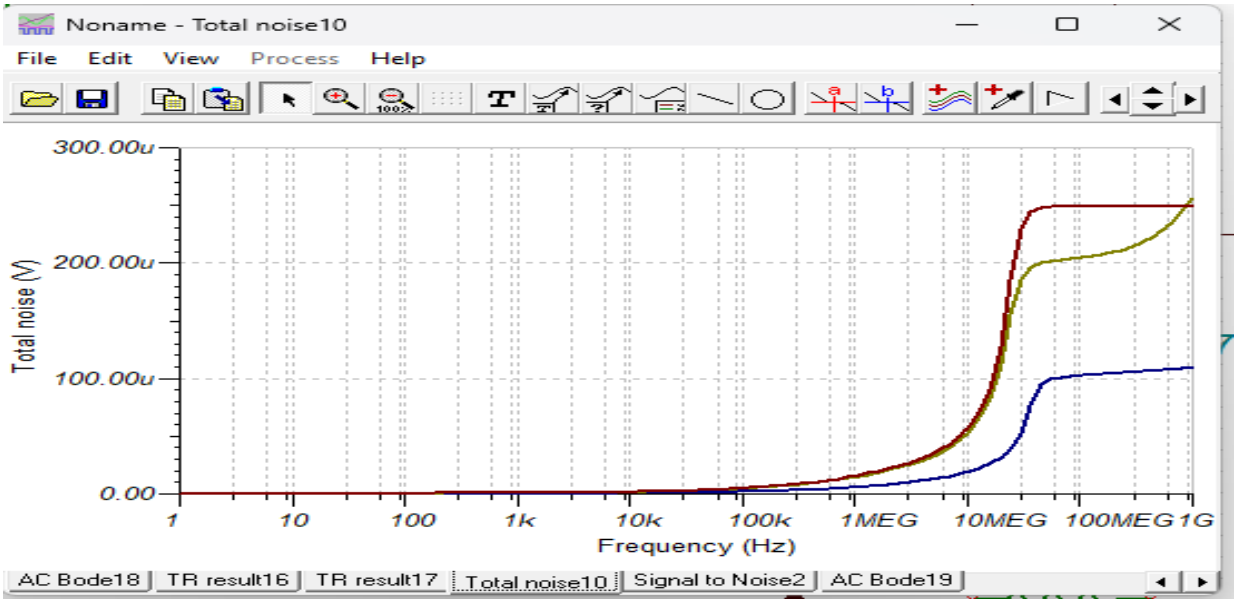


Figure 34: Total Harmonic Distortion Test Result

Test 4: Frequency Response Test

In this test, the overall frequency response of the analog front-end will be evaluated to ensure that it meets the required bandwidth specifications for the oscilloscope. This will be accomplished by applying a sweep of input frequencies and measuring the response through a test program. The test program will plot the frequency response on a graph. The frequency response should remain mostly flat within the specified bandwidth, with a roll-off that matches the design specifications beyond the cutoff frequency.

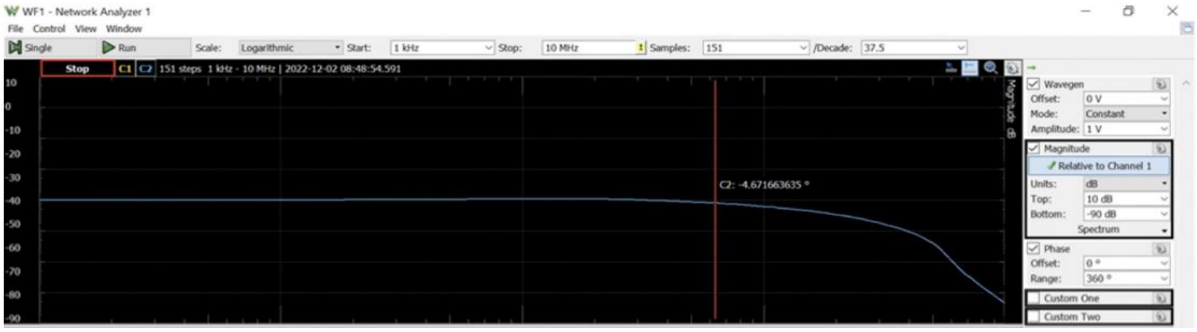


Figure 35: Frequency Response Test Result

The result ensures that the analog front-end preserves the integrity of the input signals without introducing excessive noise or distortion. A known clean, low-noise signal will be introduced as an input and a test program acting as an oscilloscope will monitor the signal-to-

noise ratio to ensure minimal interference.

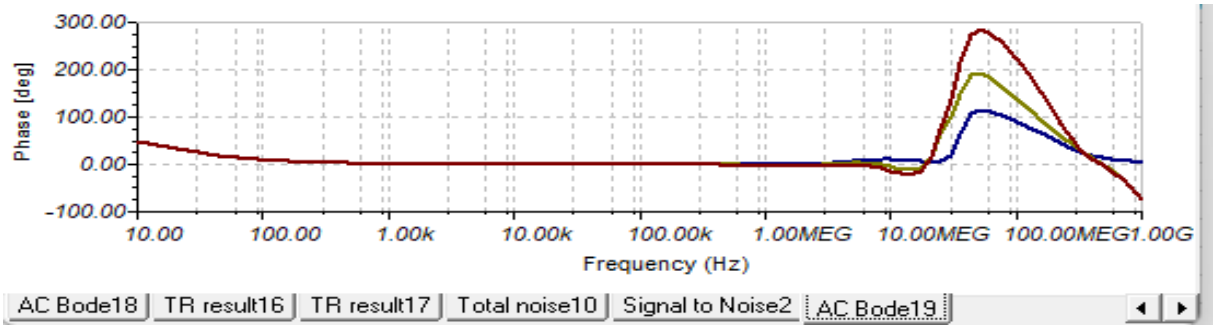


Figure 36: Frequency Response Test Result

5.2.3 GUI Experimentation

Test 1: Communication Test

This test ensured that the GUI control system running on the host computer is capable of communicating with the code running on the MCU. This was achieved partway through development by using a simple program to command the MCU to turn on a few LEDs, and to receive packets from the MCU.

Test 2: Interface Test

This test ensured that the control interface contains all required controls for the implemented wave generator and oscilloscope functions. Additionally, the usability of the UI was examined to ensure that it is understandable to a new George Mason electrical or computer engineering student with limited experience and little to no instruction.

Test 3: Control Test

This test verified that the GUI is capable of controlling the oscilloscope and waveform generator. The interface was able to issue commands to control the functionality of the oscilloscope and waveform generator. The MCU sent back data obtained from the oscilloscope, in the packet format described, which could then be graphed on the GUI.

Test 4: Compatibility Test

This test ensured that the GUI software can run on Windows, Linux, and macOS without issues. The GUI software was loaded onto computers running up-to-date versions of each operating system and the performance and stability of the software as well as its ability to interface with the hardware was evaluated on each platform.

6. Technical Background

6.1 Analog Front-End

The analog front-end of our USB oscilloscope is designed to accurately condition input signals for digital sampling by the MCU's ADC. The overall design builds upon our original approach, integrating multiple key functions such as attenuation, AC/DC coupling, amplification, DC offset injection, and buffering. The final design retains the original circuit topology, with the sole modification being the replacement of the op-amp in the amplification stage.

6.1.1 Signal Conditioning and Attenuation

The input signal is first routed through a resistor network that provides selectable attenuation, following the strictly 2 equations below.

$$R_1 \cdot C_2 = R_2 \cdot C_1$$

$$V_2 = V_s \cdot \frac{R_2}{R_1 + R_2}$$

This allows the oscilloscope to safely measure signals across a wide voltage range. In addition, an analog switch enables the selection between AC and DC coupling, ensuring that the input signal is correctly conditioned for subsequent processing.

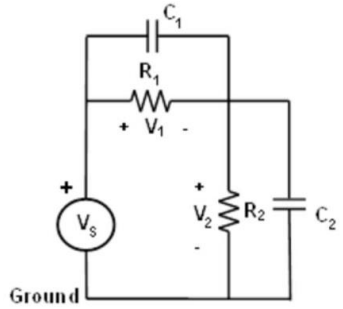


Figure 37: A Frequency Compensated Voltage Divider

6.1.2 Voltage Gain Amplification and Op-Amp Upgrade

A voltage amplifier is an electronic circuit that boosts the amplitude of a time-varying voltage or current signal. It draws electrical power from a source to increase the input signal's amplitude, delivering a higher-amplitude output. In our design, this amplification stage is configured as a non-inverting amplifier, where the gain is determined by the ratio of the feedback resistor (R_f) to the input resistor (R_{in}). Mathematically, the voltage gain *Av is* given by:

$$A_f = 1 + \frac{R_f}{R_z}$$

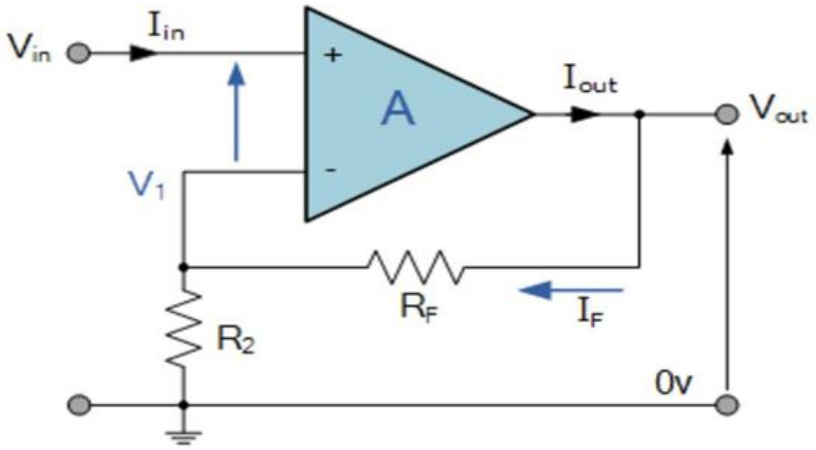


Figure 38: Voltage Gain Amplifier

For the final manufactured device, we replaced the original op amp with a higher-performance, rail-to-rail model featuring a greater slew rate and wider bandwidth. This upgrade ensures that the amplifier provides a cleaner, more accurate output signal across a broader frequency range, which is especially important for capturing fast transients and high-frequency components in oscilloscope measurements.

6.1.3 DC Offset Injection and Buffering

To ensure the entire waveform remains within the ADC's 0–3.3 V range, a PWM-generated DC offset is introduced through a low-pass filter. This offset adjustment maintains signal integrity during digitization.

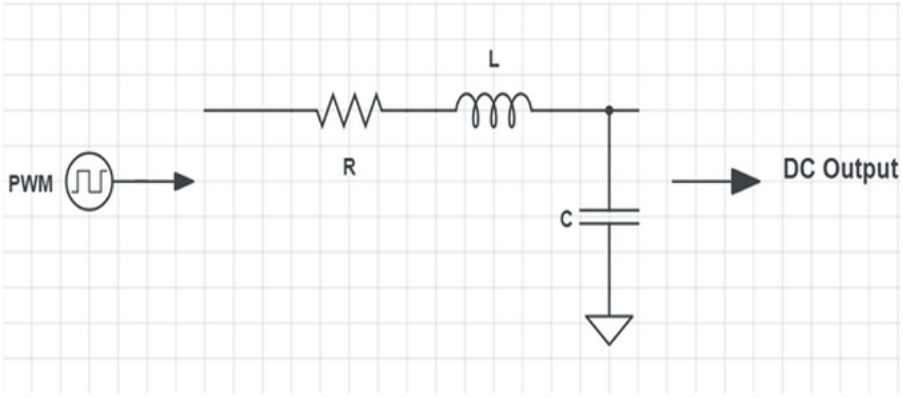


Figure 39: DC Voltage by using PWM

$f_c = \frac{1}{2\pi\sqrt{LC}}$ $z = \sqrt{(X_L - X_C)^2 + R^2}$ $R = \frac{1}{2\pi f C}$

$X_C = \frac{1}{2\pi f C}$ $X_L = 2\pi f L$ $C = \frac{1}{2\pi R f_C}$ $V_{out} = \frac{A \cdot X_C}{z}$

Finally, a unity gain buffer isolates the conditioned signal from the ADC input, minimizing loading effects and preserving the waveform's accuracy.

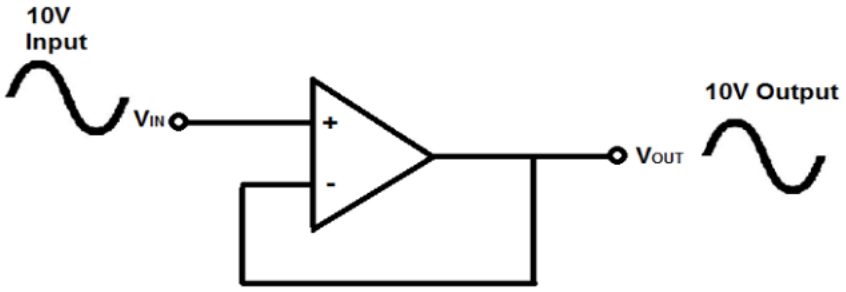


Figure 40: Unity Gain Buffer Configuration

6.2 Arbitrary Waveform Generator (AWG) Background

The AWG module in our USB multi-function lab instrument is designed to generate precise and customizable waveforms for testing and simulation. Its design builds upon the original proposal while leveraging the MCU's integrated DACs.

6.2.1 Frequency and Amplitude Range

The AWG is capable of generating signals across a wide frequency range—from a few Hz up to several MHz. This flexibility allows the AWG to simulate both slow-changing and high-speed signals. Its amplitude range is engineered to meet typical testing scenarios, ensuring that the generated voltage levels are sufficient for driving external circuits.

6.2.2 Waveform Resolution

The granularity of the generated waveforms is determined by the DAC's bit depth. With a 12-bit resolution, the DAC can produce 4096 distinct voltage levels. This high resolution allows the AWG to output smooth and accurate waveforms, which is crucial for simulating real-world signals with fine details. For an n-bit DAC operating at full scale with a reference voltage V_{ref} , the smallest change in output voltage (step size) is given by:

$$\Delta V = \frac{V_{ref}}{2^n - 1}$$

This formula shows that with higher resolution (larger nmn), the DAC can produce finer increments in voltage.

6.2.3 Digital-to-Analog Converter (DAC)

The DAC is a critical component that converts digital waveform data into an analog signal. Its performance-dictated by both resolution and sampling rate-ensures that the output waveform closely matches the intended design. In our system, the DAC's sampling rate is

maintained at 5 MS/s, which is well above the minimum requirement to avoid aliasing, thus enabling high-fidelity waveform reproduction.

6.2.4 Operational Amplifier

After digital-to-analog conversion, the resulting signal is passed through an op-amp stage. This op-amp amplifies and adjusts the waveform to the desired voltage levels, ensuring compatibility with external circuits.

$$A_v = 1 + \frac{R_f}{R_{in}}$$

The amplifier stage is configured to provide a wide range of output voltages (for instance, ±5 V to ±12 V) and is designed to drive loads up to 20 mA. This ensures that the AWG can generate robust signals even under demanding load conditions.

6.3 Logic Analyzer

6.3.1 Sampling Rate and Time Resolution

The core function of the logic analyzer is to sample digital signals at high speed. With a sampling rate of 5 MS/s, the instrument achieves a time resolution of 200 ns, allowing it to capture fast transitions and subtle timing details in digital waveforms. This high time resolution is essential for accurately reconstructing the state changes of digital signals, ensuring that even transient events are reliably recorded.

6.3.2 Memory Depth and Record Length

The performance of the logic analyzer is also defined by its memory depth, which determines the maximum duration of a captured signal trace. The record length, given by the total number of samples per channel divided by the sampling rate, allows for extended observation periods. This capability is critical when debugging complex digital systems or monitoring long sequences of events, such as bus transactions or protocol exchanges.

6.3.3 Digital Signal Levels and Accuracy

The analyzer is optimized for digital inputs operating at standard logic levels (3.3 V and 5 V). High accuracy in level detection is achieved by using comparators with well-defined thresholds, ensuring that even small variations in digital voltage are correctly interpreted. This precision is crucial for detecting glitches, metastability, or subtle timing issues that might otherwise be missed.

6.3.4 Triggering and Decoding Capabilities

Advanced triggering schemes form an integral part of the logic analyzer's functionality. Users can set triggers based on edge detection, specific logic patterns, or complex conditions across multiple channels. Once triggered, the captured data can be decoded-often by dedicated

software-to reconstructing digital protocols or to provide timing analysis. This flexibility allows for targeted data acquisition and simplifies the analysis of digital circuits and communication protocols.

7. Schematic Design – Analog Front End

7.1 Oscilloscope

The oscilloscope section of our lab instrument is developed to accurately measure and visualize electrical signals. It captures varying voltages over time, transforming analog signals into digital data through Analog-to-Digital Converters (ADCs).

7.1.1 Input Stage

The input stage employs an attenuation network, allowing measurement of both small and large signals without distortion or damage to the circuitry. Adjustable attenuation settings enable flexibility, extending the instrument's versatility across various measurement scenarios.

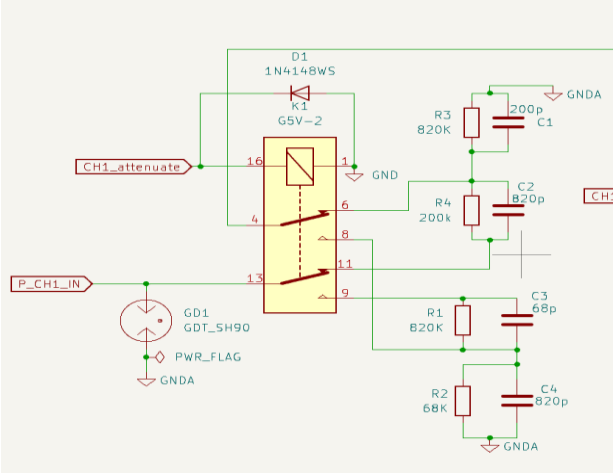


Figure 41: Attenuator

7.1.2 ADC and Signal Conversion

Utilizing a 12-bit ADC, the oscilloscope can discern 4096 discrete voltage levels, offering a detailed representation of the input signal. With a sampling rate of up to 5MS/s, the ADC provides sufficient speed to accurately capture fast-changing signals, ensuring precise waveform reproduction.

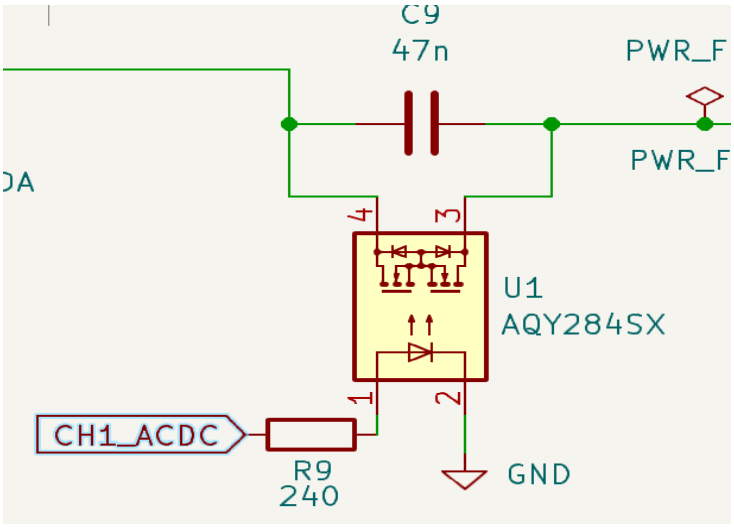


Figure 42: AC/DC Coupling

7.1.3 Amplification and Op-Amp Stage

The amplified signal passes through an operational amplifier (OpAmp) stage, significantly enhancing the signal's clarity and strength for accurate analog-to-digital conversion. Our current design incorporates improved op-amps over those used by the previous team, to minimize noise and distortion. This improves the oscilloscope's measurement reliability. The op-amp stage is configured with adjustable gain settings to accommodate varying input levels, providing a robust and adaptable measurement capability.

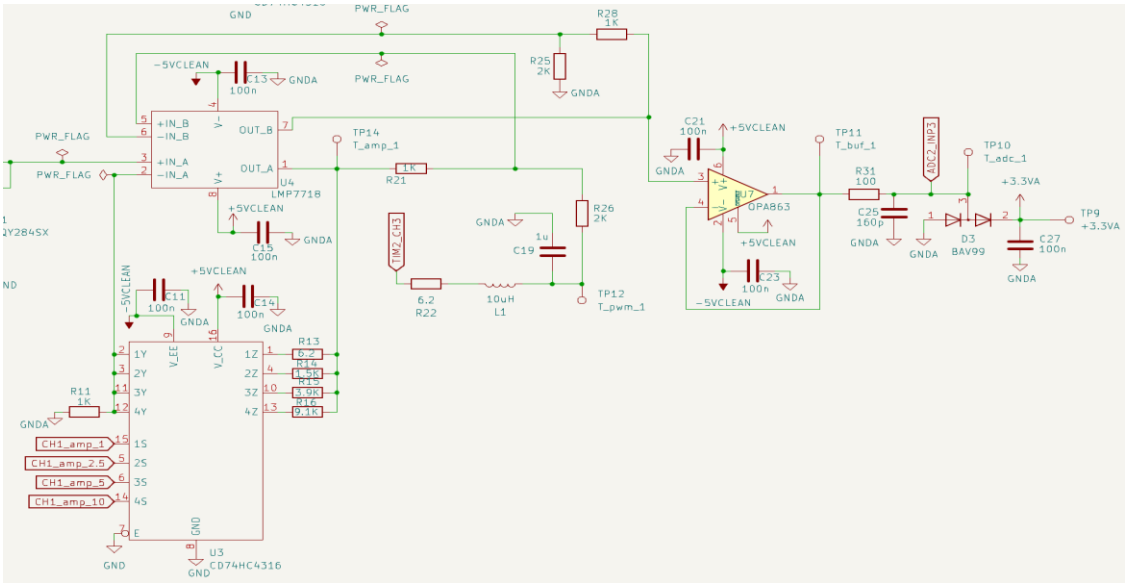


Figure 43: Gain Amplification, Offset, PWM, and Buffering

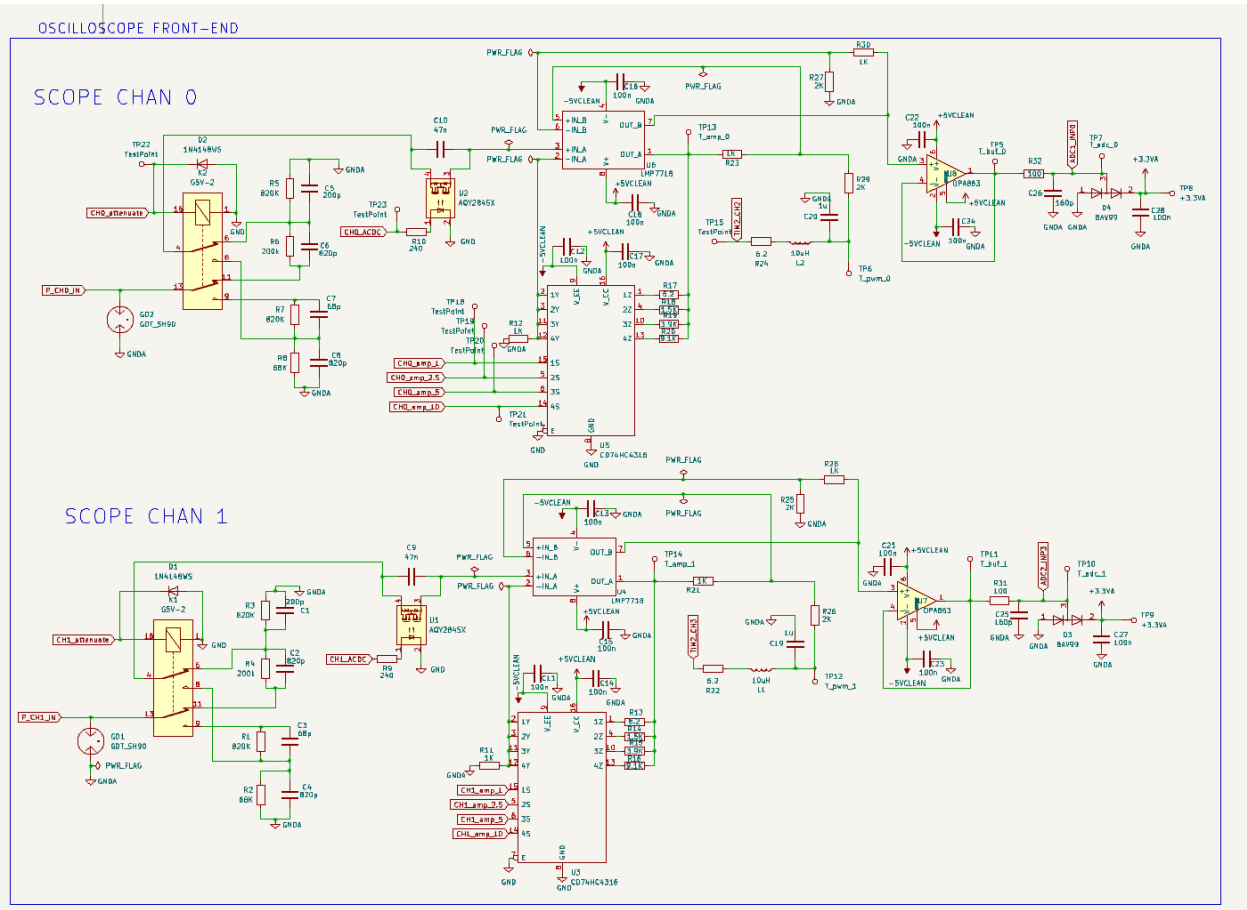


Figure 44: Oscilloscope Final Design

7.2 Arbitrary Waveform Generator (AWG)

The AWG module is designed for precise waveform generation, essential for simulation and testing applications. It utilizes the MCU's built-in Digital-to-Analog Converters (DACs) to deliver high-quality, customizable signals.

We still use the same Quad Op-Amp, MC3327ADR2G, that was previously used since it offers a much better price-to-performance than anything else; however, it is recommended that if there is no constraint in the budget, then the better course of action would be to use the same Op-Amp, LMP7718, that is used in the Oscilloscope design. Such streamlined component choices would make production a whole lot easier, and most importantly, increasing bandwidth of the device up to 10MHz from 2Mhz of the MC3327ADR2G Op-Amp.

7.2.1 Frequency and Amplitude

The AWG supports a wide frequency range, enabling comprehensive testing across diverse scenarios. It is capable of producing signals with varying amplitudes suitable for driving external circuits effectively.

7.2.2 Waveform Resolution and DAC

Waveform accuracy is governed by the DAC's 12-bit resolution, which allows for 4096 distinct voltage levels. The DAC's high sampling rate of 5MS/s ensures fidelity, avoiding signal aliasing and providing clear and smooth signal reproduction.

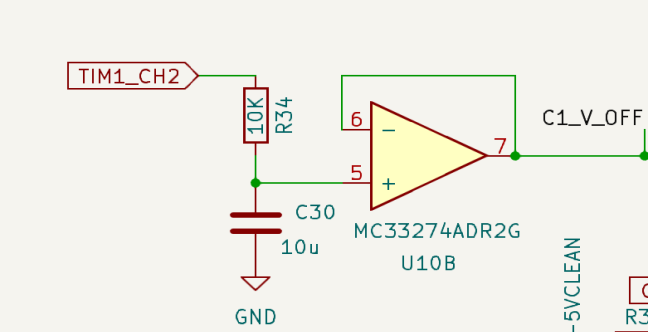


Figure 45: DAC to PWM/DC Offset

7.2.3 Amplification Stage

Post-DAC amplification is achieved through an OpAmp, adjusting the waveform to appropriate output voltage levels. This stage is specifically designed to deliver a broad range of voltages, typically from ±5V to ±12V, and can drive loads up to 20 mA. This capability ensures signal integrity even under rigorous conditions.

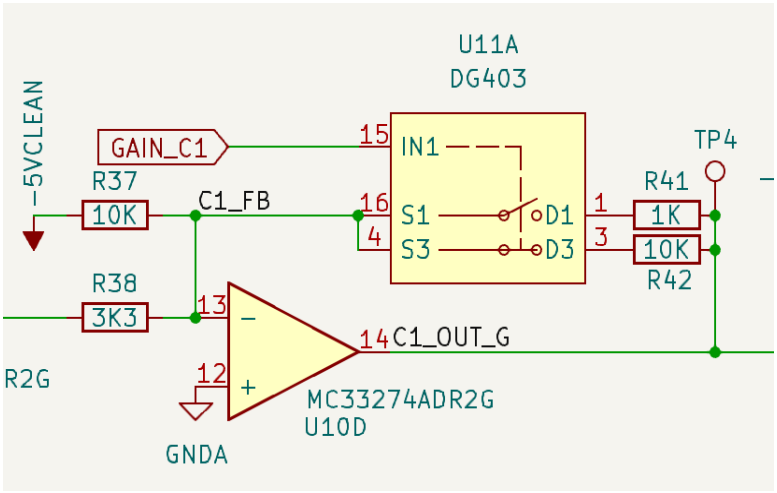


Figure 46: Supplied Wave and Gain Adjustment

7.2.4 Dual-Channel Operation

Our AWG incorporates two independent channels, enabling simultaneous generation of distinct waveforms. This functionality is particularly advantageous when testing circuits requiring multiple synchronized or varied input signals.

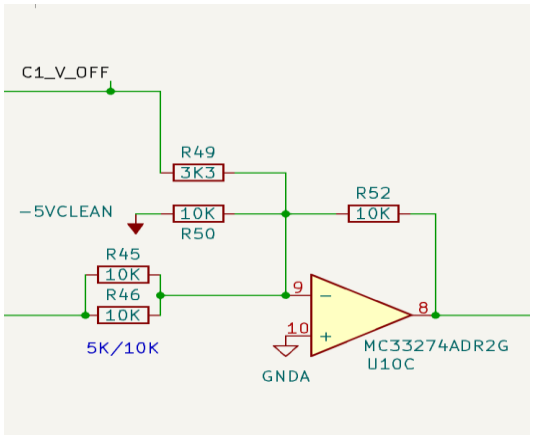


Figure 47: PWM Offset

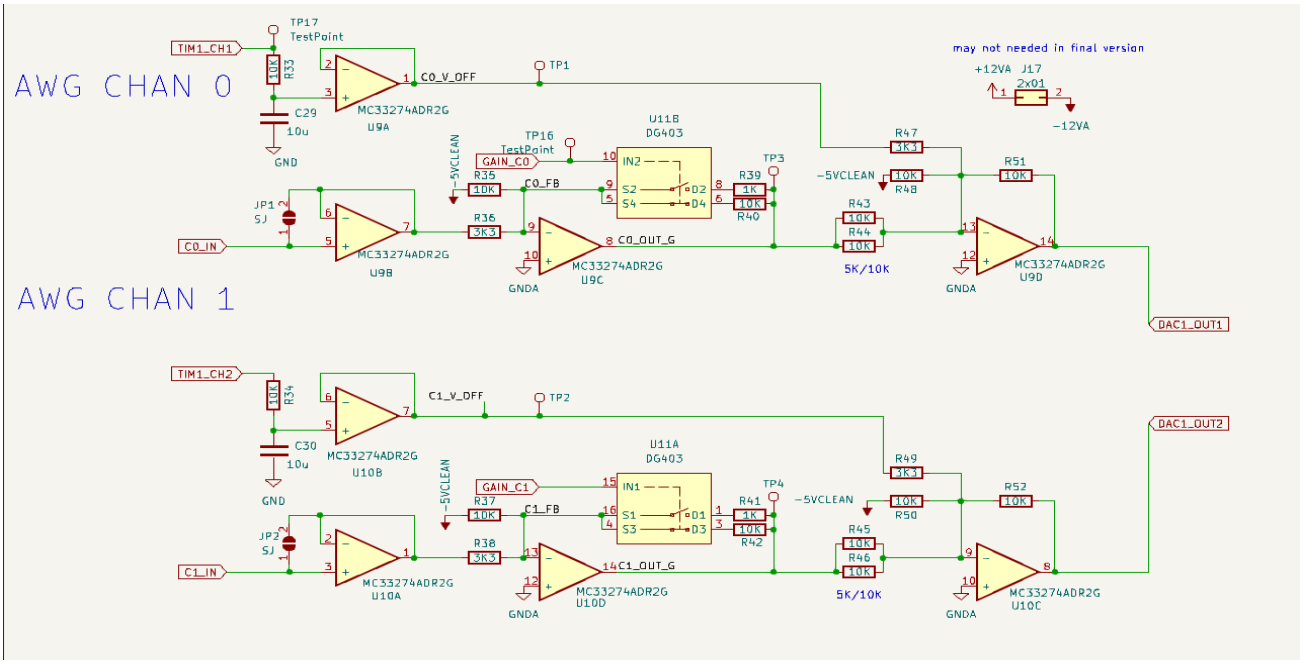


Figure 48: AWG Final Design

7.3 Logic Analyzer

The logic analyzer component of the instrument captures digital signals from embedded systems, facilitating debugging and testing tasks. It samples multiple digital lines simultaneously, translating complex digital interactions into clear visual representations.

7.3.1 Sampling Rate and Channels

Capable of handling sampling rates up to several MHz, the logic analyzer precisely records rapid digital transitions. The system supports multiple input channels, providing comprehensive coverage for analyzing parallel digital communications and synchronous systems.

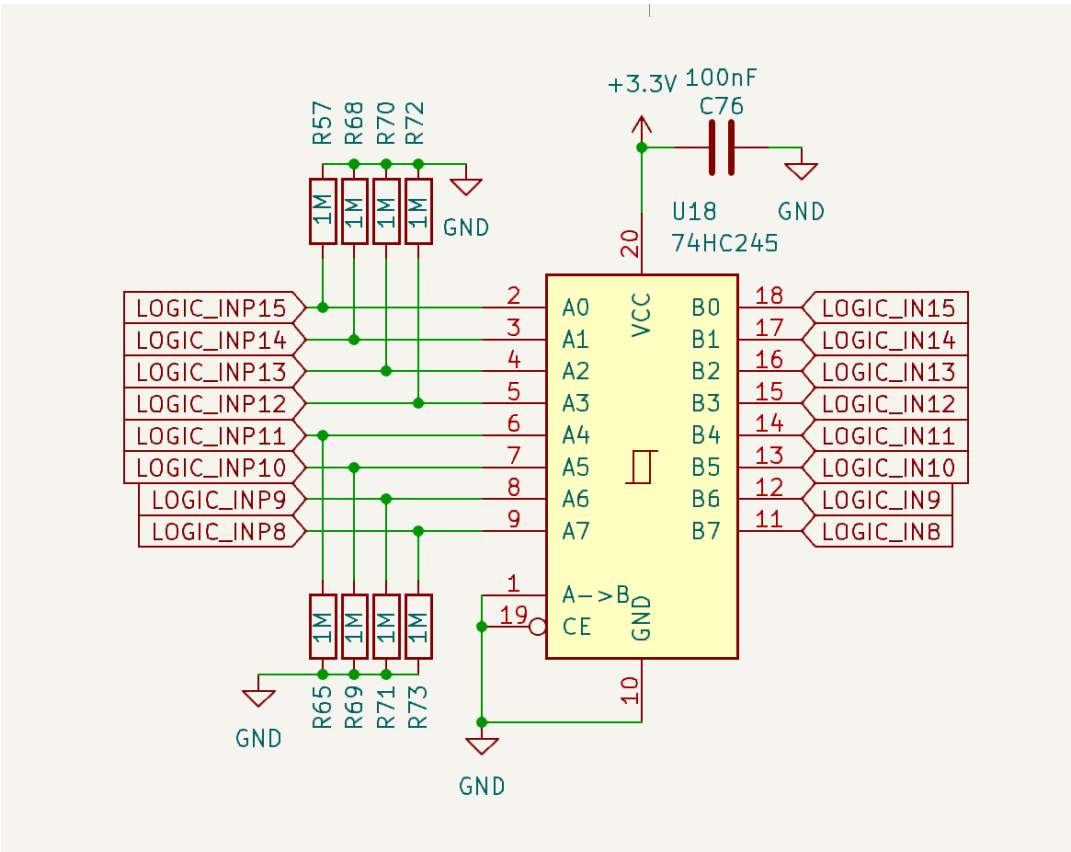


Figure 49: Input/Output with Transreceiver

7.4 Power Supply

The power supply subsystem ensures stable and clean power delivery across all modules. Utilizing a dual-rail ±12 V DC-DC converter and dedicated linear regulators for critical voltage rails such as +3.3 V and -5 V, the design provides consistent and isolated power. Special attention is given to noise minimization, achieved through the strategic placement of inductors, capacitors, and ground separation techniques. This careful power management guarantees reliable operation across all functional blocks.

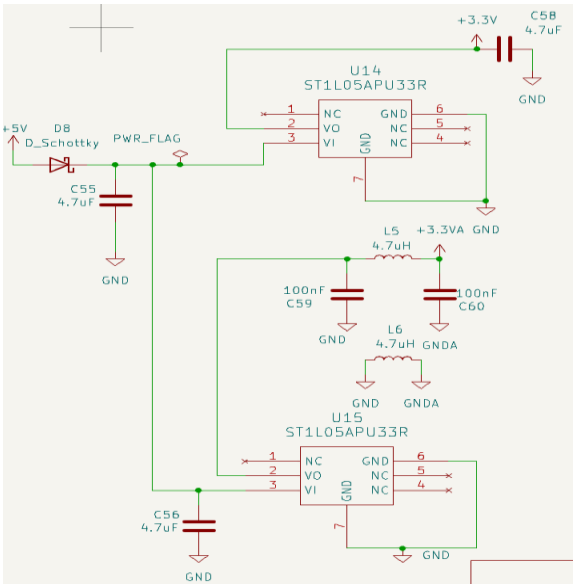


Figure 50: 3.3V and 3.3VA rails

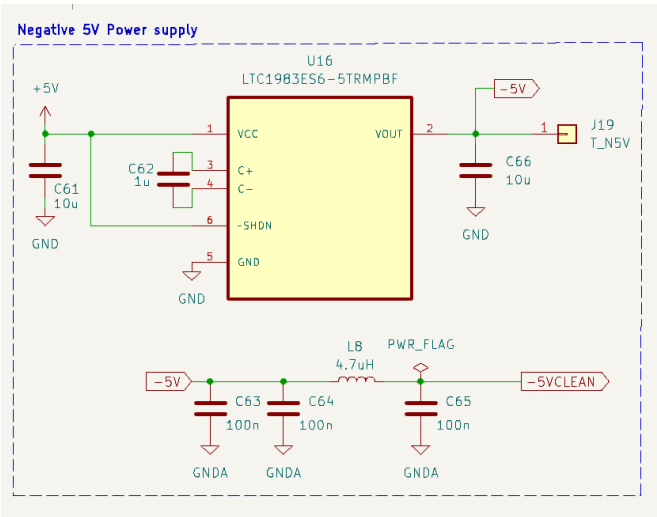


Figure 51: -5V and –5VCLEAN voltage rail

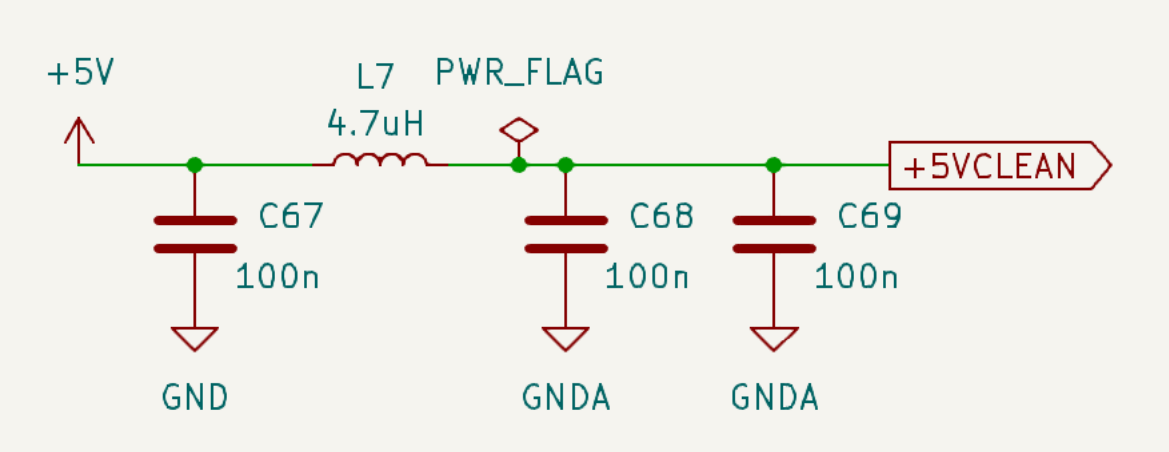


Figure 52: +5VCLEAN (+5V analog) rail

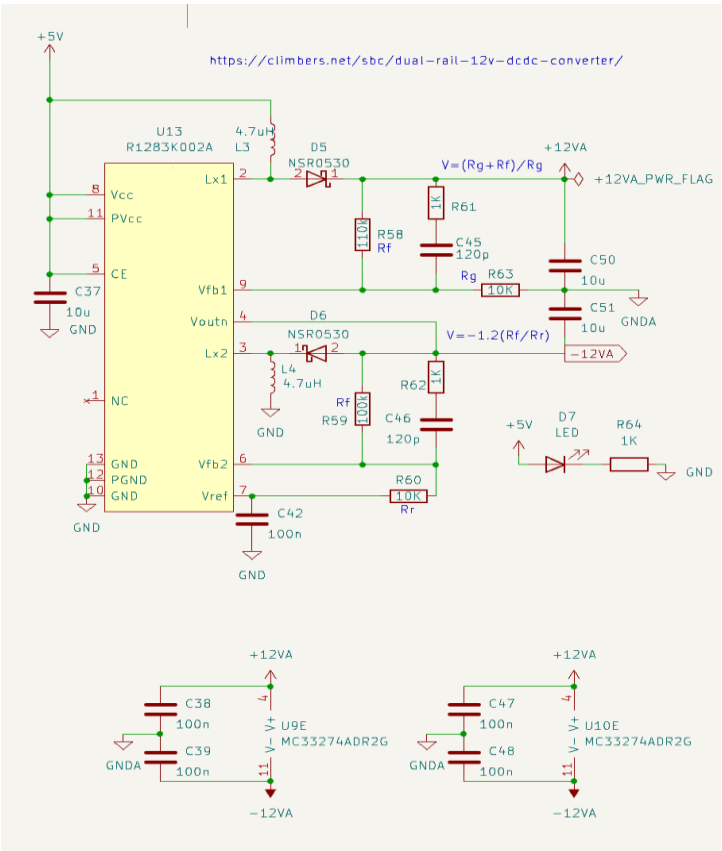


Figure 53: ±12V Analog Voltage Rail Schematic

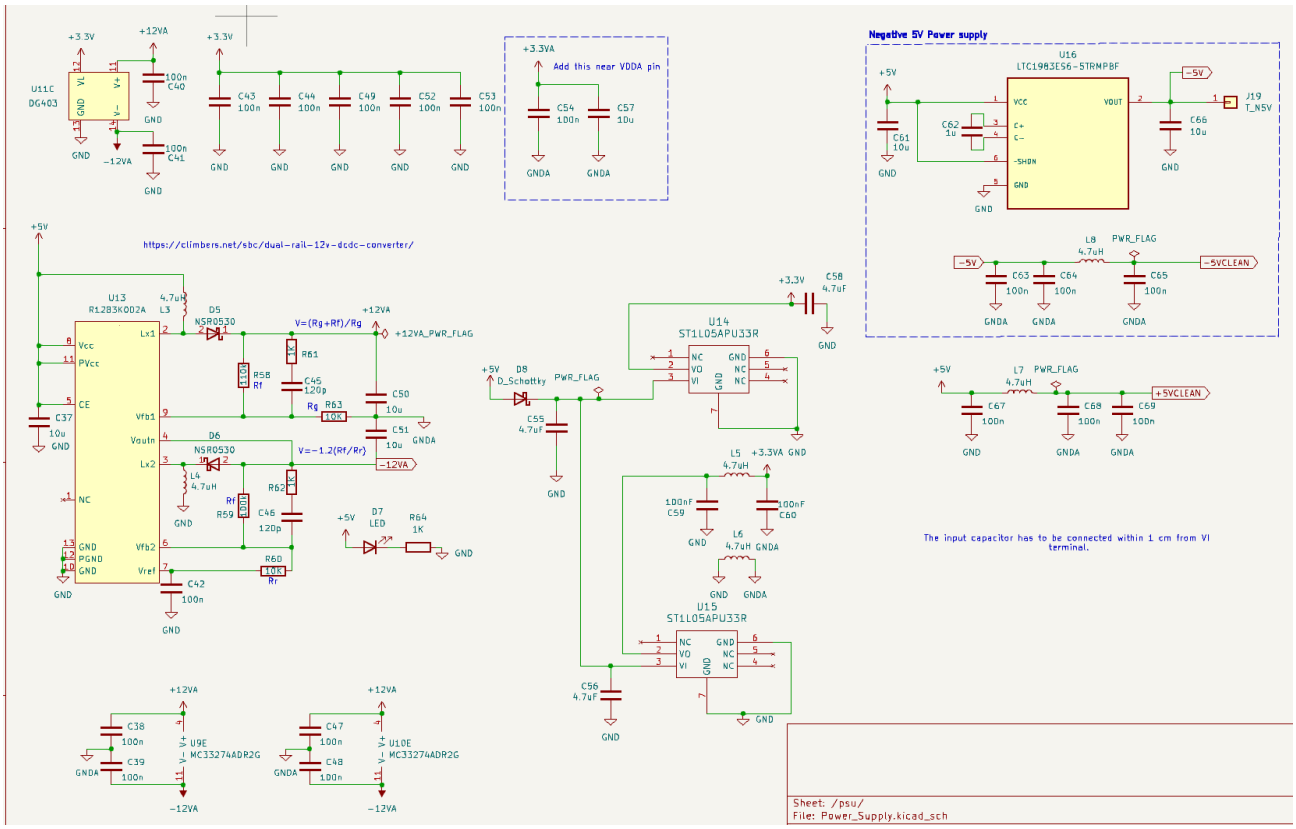


Figure 54: PSU Design

8. Administration

8.1 Allocation of Responsibilities

Team Roles	
Anh Nhat Van	Analog Systems PCB Schematic Design PCB Assembly
Braden Camp	Team Lead PCB Design Lead PCB Assembly FS Contact (Spring ‘25)
Timothy Kolos	MCU Programming Case Creation
Anna Mai	GUI Lead (Fall ‘24) GUI Programming
Erick Orellana	GUI Lead (Spring ‘25) GUI Programming FS Contact (Fall ‘24)

8.2 Funds Spent

8.2.1 Prototyping Costs

Item	Cost
5x Nucleo Development Boards	$175.00
PCB v1 DigiKey Order	$126.07
PCB v1 JLCPCB Order	$274.16
PCB v2 DigiKey Order	$52.98
PCB v2 JLCPCB Order	$337.62
Total Prototyping Costs	**$965.83**

8.2.2 Final Product Cost Breakdown

Item	Cost per Unit*
Components	$30.15
PCB	$0.62
Case	$0.57
Total Cost	**$31.34**

* assuming order of 1000 units

8.3 Man-Hours Devoted to the Project

Team Member	Hours Worked
Braden Camp	140
Timothy Kolos	142
Anna Mai	156
Erick Orellana	155
Anh Nhat Van	140
Total	**733**

8.3 Teaming Experience

8.3.1 Team Communication and Dynamics

To facilitate team communications, we utilized an online Discord group chat for convenient communication. This also served as a hub for the project, with “channels” set up to hold links to important team documents and information as well as for conversations related to each different section of the project. We were also able to hold virtual meetings on this platform. This allowed close and effective communication between members throughout the project.

Despite being able to meet virtually, one of the key things we prioritized was regular in-person team meetings that also included our faculty supervisor. This allowed us to receive regular feedback on our progress as well as to answer questions and coordinate tasks more effectively than a virtual meeting would allow. We made an effort to meet in-person weekly during the semester, and these meetings proved to be critical to our success as a team. For future reference, the details and proceedings of the meetings as well as upcoming action items were recorded in meeting minutes.

8.3.2 Project Management

One of the most challenging parts of a group project is coordinating members' efforts and schedules to make progress as a team. We utilized various systems to make it easier to schedule meetings, stay on task, and minimize confusion. The primary system we used was the Discord chat described in the previous section. We also maintained a calendar with the class schedules of each member recorded together, to show when team members would be available. Additionally, we created and maintained various spreadsheets to record team information, such as a project schedule with important tasks and deadlines recorded as well as who should complete them, a BOM for the project, and an inventory spreadsheet of relevant items owned or purchased by team members for the project. Finally, most of our documents were kept in a OneDrive folder shared with each member of the team, and code was maintained on a shared Git repository. This made the current versions of all files accessible for each team member at any time.

9. Problems Faced

9.1 PCB Design and Assembly

The design of the PCB is complex since it must accommodate mixed analog and digital signal components which can complicate routing. Additionally, the physical placement of components is critical to reducing crosstalk, noise, and ground loop issues. Managing high-frequency signals on a shared PCB can be particularly challenging, as a poor PCB layout can lead to performance degradation, increased electromagnetic interference (EMI), and inaccurate measurements.

Due to our team's limited experience with PCB design as well as the limitations of the KiCad software, the PCB routing was likely suboptimal and could have diminished the performance of our device, introducing unwanted electrical noise into the system. An improved PCB design in the future would make better use of power and ground planes, as well as additional layers, to improve electrical isolation and reduce noise.

We also faced difficulties with the assembly of the PCB, since while our PCB manufacturer of choice offered assembly services, they lacked stock of many of the parts used in our design. Therefore, we needed to order those parts separately and solder them onto the board ourselves. Due to our team's initial lack of experience with SMD soldering, while we were ultimately successful in assembling our prototypes, some errors were made during this process which made testing of the boards more difficult. Our initial prototype boards required some modifications to work properly due to errors in the PCB design itself, adding to the problems faced with assembly.

9.2 Signal Integrity and Noise

Since the device handles both high-frequency analog signals (for AWG and oscilloscope) and digital signals (for the logic analyzer), there is a risk of signal interference and noise between

different subsystems. Additionally, an inadequate or faulty opamp can generate significant electrical noise and distortions, which can interfere with the functionality of different systems.

9.3 Overload and Signal Clipping

High amplitude signals may exceed the input range of the ADC or maximum output range of the DAC, even with low-pass filters in place. This leads to signal clipping or distortion. Signal clipping can distort the waveform, leading to inaccurate measurements and faulty analysis. Overloading the inputs could also damage sensitive components.

9.4 MCU

While the STM32-H563VGT6 has met all requirements and is a very capable MCU, it uses a different ARM processor core than the F303 MCU utilized by the previous team, among other differences. This presented challenges with testing of the existing code, and the electrical layout and design of the PCB also needed to be changed to accommodate the distinct electrical and physical characteristics of the new MCU.

Additionally, unlike the previously used F303, the limitations of this microcontroller with respect to the project were not evaluated by previous teams. While the specifications of the microcontroller meet or exceed the project requirements on paper, practical testing presented unexpected challenges.

The first problem encountered was to do with USB. Given the H563 is a newer MCU, STMicro dropped support for the classic USB library used by the F303. The only USB library officially supported required the use of a RTOS. While it was possible to get the classic USB library working on the H563ZIT6, this may be a reason that the H562/3 are not working properly on the PCB.

The clock for the ADC/DAC1 presented the second issue. If only one of the peripherals are used, the clock speed can exceed 125MHz. However, this project required the use of both, hence the speed had to be below 125MHz. This would not have been an issue had the clock prescaler worked properly. To achieve the desired sampling frequency of 5MS/s, the clock needed to be set at 75MHz with a prescaler of 1. Unfortunately, there was an issue with the data from the ADC seemingly not being ready when the prescaler was set to 1 resulting in the 4 LSBs being lost. This behavior occurred even at extremely low ADC/DAC1 clock speeds (5MHz). This necessitated halving the sampling rate by increasing the prescaler to 2, effectively halving the sample rate as the ADC/DAC1 clock could not be increased to 150MHz.

The final problem encountered in regard to the ADC was with the analog watchdogs. The analog watchdogs were used to detect the presence of a trigger. Two were used per ADC to distinguish rising from falling edge and both triggered whenever the ADC was restarted, resulting in the trigger mode for the ADC not working.

The microcontroller on the PCB exhibited an issue where it would become unstable if the SYSCLK was set to the maximum rated speed (250MHz). This behavior was not exhibited when

halving SYSCLK to 125MHz. Other speeds were not tested. This issue may be related to the USB library as it frequently hung on initialization of the CDC Class.

9.5 GUI

Although the GUI team members had some experience in creating GUIs, none had experience working with PyQt6 or PyQtGraph. Additionally, since the original plan was to utilize GUI code created by the previous teams, bugs in the preexisting code were inherited into the project. The reused code also proved to be difficult to integrate for use with all three functions (AWG, oscilloscope, logic analyzer), which necessitated abandoning this approach and redesigning much of the GUI from the ground up.

10. Lessons Learned

10.1 Knowledge Gained

10.1.1 Front End

During the development of the analog front-end, our team acquired significant practical knowledge about conditioning input signals for accurate digital sampling. We gained a deeper understanding of designing adjustable attenuation networks and frequency compensated voltage dividers to handle a variety of input voltages safely and effectively. Upgrading the operational amplifiers was a crucial step which enhanced our device's performance significantly, especially in terms of bandwidth. In addition, streamlining our component sizes and variations would have been helpful with production and inventory management.

Implementing DC offset injection via PWM and buffering stages was critical for ensuring that signals remain within measurable ranges. We also gained valuable hands-on experience in troubleshooting noises and signal integrity issues through careful PCB layout, grounding techniques, and selective component upgrades. Comprehensive testing reinforced the importance of validating theoretical expectations against practical results, which will shape our approach to future analog design projects.

10.1.2 MCU

Programming on the STM32H563 is quite different from the MSP430 used in GMU classes. It is more complex and capable than the MSP430 and the development tools are likewise complex. There was much learning needed to understand the functions of STM32CubeIDE as well as the libraries which would be best suited for this project. Later development utilized debugging tools to determine where memory problems were occurring and why.

10.1.3 GUI

Development of the GUI involved extensive knowledge of PyQt and PyQtGraph and with the sparce resources available online, required us to solve any errors or questions amongst our team's collective knowledge and experience. Additionally, GitHub was used to maintain a shared

code repository amongst members and our faculty supervisor. This gave us experience with version control in a collaborative software development environment, which is not covered in the GMU ECE curriculum.

10.1.4 CAD Software and PCB

Our project required us to become familiar with the KiCad design suite, which we utilized to create our PCB. Since those among us who worked on the PCB had taken ECE 436, which teaches the fundamentals of PCB design using KiCad, we were able to leverage that past experience for our project. Nonetheless, the PCB designed in this project was more complex than any covered in the ECE 436 lab, so the project was still a valuable learning experience in this area that gained us significantly more experience with using the KiCad software and PCB design in general.

We also gained experience with SMD soldering in the process of assembling the PCB, something none of us had attempted previously. While this posed some challenges, through the experience we were able to learn enough to assemble a few prototypes successfully, and should be able to use this experience again in future endeavors.

10.2 Key Takeaways

10.2.1 Trust but Verify

During the design and testing of our device, we uncovered a significant number of issues with the previous teams' designs, many of which were unfortunately not well-documented. While one should assume good-faith intentions, mistakes are possible and therefore it is better to trust but verify others' designs before incorporating them into further work.

10.2.2 Prototype Early

Due to time-management blunders on our part there were significant delays with beginning the prototyping phase of the project. We did not anticipate that this would be an issue, however the prototyping and ordering took longer than we expected, and we were forced to rush to complete it, forfeiting the chance to potentially redesign the board in a better way than our first attempts. It would have been wiser to start the prototyping as early as possible to allow the most time for hardware and software debugging and revision.

10.2.3 Prioritize Performance Over Budget Early

We made many design choices early on to stay within the limited project budget outlined earlier. However, in the end we discovered that the total manufacturing cost of our design unexpectedly fell well short of the budget ceiling for this project, meaning that we could have spent more on improved components to achieve better performance, or designed a larger PCB that would have been easier to test and less complex to route. Therefore, we realized that it would have been better to be less conservative in the prototyping stage and then make any

cutbacks that were needed in the final design to meet the budget, than to be overly cautious from the start.

11. References

[1] P. Dong, D. Luong, J. A. J. M. Alsaei, B. Phan, G. Nguyen, T. V. M. Nguyen. “Affordable USB Oscilloscope Final Project Report,” B.S. thesis, Dept. Electrical and Computer Engineering, George Mason University, Fairfax, VA, USA, 2023.

[2] J. Schaeffler, G. Kuznetsov, H. Zainal, V. Arunachalam, Y. Gao, W. Denham. “Arbitrary Waveform Generator Final Project Report,” B.S. thesis, Dept. Electrical and Computer Engineering, George Mason University, Fairfax, VA, USA, 2023.

[3] “BitScope Micro Model 5: A tiny USB Mixed Signal Oscilloscope.” BitScope Designs. https://www.bitscope.com/product/BS05/ (accessed September 27, 2024).

[4] “Analog Discovery 3: USB Oscilloscope and Logic Analyzer.” Digilent. https://digilent.com/shop/analog-discovery-3/ (accessed September 27, 2024)

[5] A. Suciu. “ADALM2000 Overview.” Analog Devices Wiki. https://wiki.analog.com/university/tools/m2k (accessed September 27, 2024).

[6] Pico Technology Ltd, St. Neots, Cambridgeshire, United Kingdom. *Picoscope 2000 Series Oscilloscope Data Sheet,* 6th ed (2019). Accessed: September 27, 2024. [Online]. Available: https://www.picotech.com/download/datasheets/picoscope-2000-series-data-sheet-en.pdf

[7] "Using analog offset to maximize oscilloscope resolution," Pico Technology Ltd. https://www.picotech.com/library/application-note/using-analog-offset-to-maximize-oscilloscope-resolution (accessed September 24, 2024).

[8] “Analog Discovery 2.” Digilent. https://digilent.com/reference/test-and-measurement/analog-discovery-2/start (accessed September 27, 2024).

[9] A. S. Sedra, K. C. Smith, T. C. Carusone, and V. Gaudet, *Microelectronic Circuits*, 8th ed. New York, NY, USA: Oxford University Press, 2020.

[10] *ECE 334 Linear Electronics Lab I Laboratory and Experimental Guide*, Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA, 2015.

[11] *ECE 434 Linear Electronics Lab II Laboratory and Experimental Guide*, Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA, 2015.

[12] “OpenScope MZ: Getting started.” Digilent. https://digilent.com/reference/test-and-measurement/openscope-mz/start (accessed September 27, 2024).

Appendix: Proposal (ECE 492)

College of Engineering and Computing
VOLGENAU SCHOOL OF ENGINEERING
George Mason University

ECE 492 Senior Design Project

Affordable USB Multi-Function Lab Instrument

Project Proposal

Team Members:

Anh Nhat Van

Braden Camp

Timothy Kolos

Anna Mai

Erick Orellana

Faculty Supervisor: Dr. Jens-Peter Kaps

8 October 2024

Table of Contents

1. Executive Summary 4
2. Problem Statement 4
2.1 Motivation and Identification of Need 4
2.2 Market Review 5
3. Approach 6
3.1 Problem Analysis 6
3.2 Our Approach 6
3.2.1 MCU 6
3.2.2 Hardware 6
3.2.3 GUI 7
3.3 Alternative Approaches 8
3.4 Background Knowledge 8
3.4.1 MCU 8
3.4.2 Oscilloscope 8
3.4.3 Arbitrary Waveform Generator 10
3.4.4 Logic Analyzer 11
3.4.5 GUI 11
3.5 Project Requirements Specification 12
3.5.1 Mission Requirements 12
3.5.2 Operational Requirements 12
4. System Design 13
4.1 Functional Decomposition 13
4.2 Physical Architecture 14
4.3 System Architecture 15
5. Preliminary Experimentation and Testing Plan 15
5.1 Preliminary Experiment 15
5.2 Experimentation Plan 15
5.2.1 MCU Experimentation 15
5.2.2 Analog Front-End Experimentation 16

5.2.3 GUI Experimentation 17
6. Preliminary Project Plan 18
6.1 Overview 18
6.2 Allocation of Responsibilities 19
6.3 Detailed Project Timeline (ECE 492) 19
6.4 Estimated Project Timeline (ECE 493) 20
7. Potential Problems 20
7.1 PCB Design 20
7.2 Signal Integrity and Noise 20
7.3 Overload and Signal Clipping 20
7.4 MCU 20
7.5 GUI 21
8. References 21

1. Executive Summary

The purpose of this project is to design and construct an affordable multi-function lab instrument that will satisfy the needs of the ECE department's lab curricula. To accomplish this, the lab instrument will include a dual channel oscilloscope, arbitrary waveform generator, and logic analyzer. Ultimately, the device should have similar features to other multi-function lab equipment on the market while being available at a lower price.

This project will be the culmination of the combined efforts of three other senior design teams, integrating their projects into a single device. The two teams who worked on the oscilloscope and arbitrary waveform generator have finished senior design and completed their projects. The third team is actively developing the logic analyzer component, which is expected to be complete by the end of the Fall 2024 semester.

To implement the various components of this design, our team will be partitioned into three parts. One to develop the GUI, another for the microcontroller and digital subsystems, and the third dedicated to the analog front-end circuitry. Given that only the oscilloscope and waveform generator projects are fully complete as of this proposal, the goal for the end of ECE 492 is to have both of those systems integrated and fully functional. The GUI will be redesigned to accommodate both systems, and issues present in the prior projects will be resolved. The primary goals for ECE 493 will be to complete the implementation and testing of the logic analyzer and to address any unresolved issues from ECE 492. This proposal outlines our plan for this project, including its functional requirements, budget, and planned testing procedures.

2. Problem Statement

2.1 Motivation and Identification of Need

The ECE department currently requires students to purchase multi-function lab equipment such as the Analog Discovery 2 (AD2) or Advanced Active Learning Module (ADALM 2000). These devices are necessary for students to complete lab work for various classes. They support them with coursework and projects outside of class should an on-campus laboratory be unavailable, such as in the case of distance learning or times outside open lab hours. The COVID-19 pandemic rendered the lab equipment on campus inaccessible, and students needed to default to these consumer-level multi-function lab devices. This created a large demand for such products that could not be met by a supply chain weakened by the effects of the pandemic. As a result, the availability of such parts became limited, and their prices rose. This made getting into electrical and computer engineering much more difficult for students due to difficulties in sourcing necessary tools.

While prices on multi-function lab instruments have stabilized since the supply constraints of the pandemic, they can still be quite expensive today. The ADALM2000's price is roughly $230. The more capable and now retired AD2 has been replaced by the AD3 with a price

of $379. Not all the functions on these devices are required for the ECE curriculum. This project will create a purpose-built USB multi-function lab device at a much-reduced price compared to the AD2, AD3 and ADALM2000. The device will have two ADC channels for the oscilloscope with a 12-bit resolution, and have a sampling rate of 5MS/s. There will also be two 12-bit DAC channels sampling at around 5MS/s for the waveform generator and the necessary GPIO for the logic analyzer. It will receive power and transmit data over a USB interface, with the user able to control the device's functions through a custom GUI on the host system. The capabilities and accuracy of this device will be adequate for the needs of ECE students.

2.2 Market Review

Model	Analog Discovery 3 	ADALM 2000 	Pico 2204A BASIC 	BitScope Micro BS05 
Price (USD)	$249	$210	$129	$145
Oscilloscope				
Sample Rate (MS/s)	125	100	50	N/A
Number of Channels	2	2	2	2
Sampling Resolution	14-bit	12-bit	12-bit	N/A
Waveform Generator				
Sample Rate (MS/s)	125	100	20	N/A
Number of Channels	2	2	2	N/A
Voltage Range	±5V	±5V	±2V	N/A
Logic Analyzer				
Channels	16	16	N/A	8
Logic Level	3.3V CMOS	3.3V CMOS	N/A	3.3V CMOS
Sample Rate (MS/s)	125	100	N/A	40

Table 1: Current Available Alternative Multi-Function Lab Equipment

3. Approach

3.1 Problem Analysis

This problem mainly concerns the functionality of combining the three systems together and keeping the cost of the system low. The team will need to ensure there are no conflicts within each system and fix any hardware or software issues that the prior teams lacked the time to fix. Lastly, the GUI will have to be revised to accommodate the functionality required.

3.2 Our Approach

3.2.1 MCU

The prior teams' choice of MCU was the STM32-F303RE. This microprocessor is equipped with a 72MHz ARM Cortex-M4 CPU, 512kB flash, 80kB RAM, two 12-bit DACs, and four 5MS/s ADCs. The team who worked on the oscilloscope noted that the MCU would need more than 64kB RAM to store data from the ADC. This left only 16kB at best for the DACs needed for the waveform generator and what little may be needed for the logic analyzer. Additionally, the F303 has a single 12-bit DMA to reduce the impact of the ADC and DAC on the CPU. Considering prior projects only had to handle a single device, it may be wise to double this for implementing both of these. As such, our team is concerned about the performance of this MCU with all three systems implemented.

We plan that our device will instead be based on the STM32-H562RGT6. This features a much faster 250MHz ARM Cortex-M33 CPU, four times the flash memory, 640kB RAM, the same DACs, two 5MS/s ADCs, and two 12-bit DMAs. With it being a significant upgrade over the F303RE, we do not expect to be constrained by an underpowered MCU.

3.2.2 Hardware

To ensure the quality and accuracy of the signals being captured, we are using an analog front-end that includes attenuators, amplifiers, and low-pass filters. These components help condition the incoming signals, making sure they're within the proper voltage range and free from excessive noise. By leveraging the MCU's DMA channels, we can efficiently transfer the data from the ADCs to memory without overloading the CPU, allowing for smooth real-time display. For the AWG, we are utilizing the two 12-bit DACs built into the STM32-H562RGT6 to generate precise, customizable waveforms, like sine, square, and triangle waves, etc. After the DACs, the signals will be fed into Op-Amps to amplify and adjust the voltage to the levels required for external circuits.

There are some minor problems within the current systems of the oscilloscope and AWG which we will address. The first involves the frequency response of the oscilloscope. At higher magnitudes (dB), the frequency drops off too quickly. We believe this to be due to the opamp used in the gain amplifier and DC offset. We plan to replace it with an opamp with a higher gain

bandwidth product to resolve this problem. The other problem lies within cost. A number of components chosen by the previous teams are unnecessarily expensive. These will be replaced with equal but cheaper alternatives.

3.2.3 GUI

The main purpose of a GUI is to present the possible actions of a system to the user in a manner that makes the functionality of the system understandable and pleasant for the user. Each of the previous projects had an individual GUI that controls their respective components, so the main task will be creating a GUI that integrates all three parts. We will be using PyQtGraph for a few reasons. First, PyQtGraph is based in the higher-level Python programming language, which allows us to more easily achieve the goal of functionality on all major operating systems. Secondly, PyQtGraph is faster than other graphing libraries such as Matplotlib which is vitally important for real-time graphing such as in our oscilloscope requirements. Finally, both previous teams have used PyQtGraph as well, which allows us to leverage components of their GUIs for our design. Below is the planned prototype for our GUI to view live waveforms.

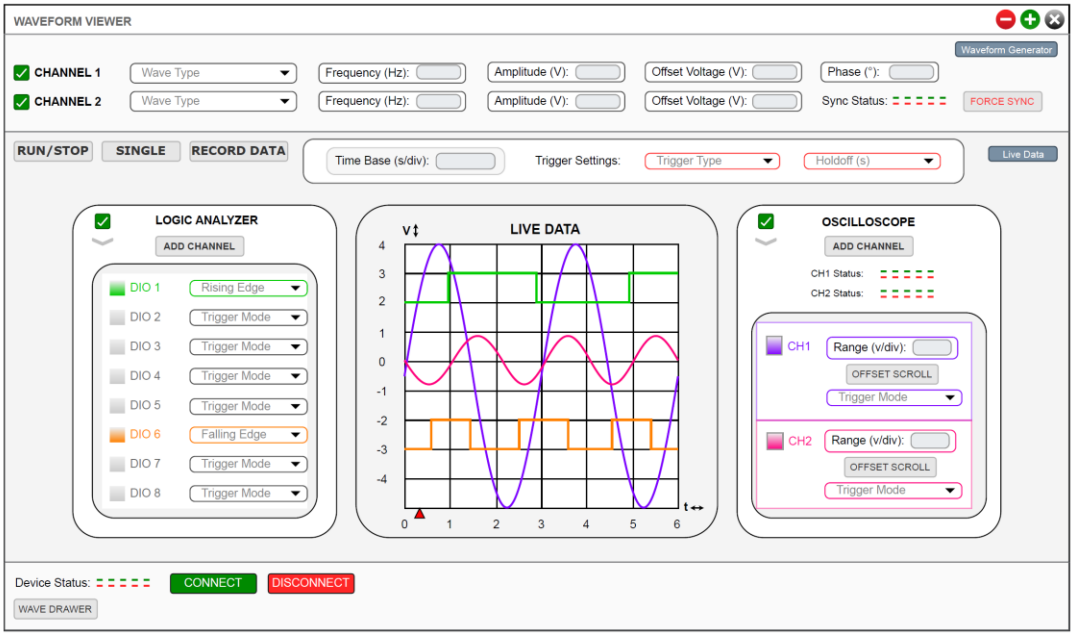


Figure 1: GUI Prototype

The top of the window houses the controls for the arbitrary waveform generator, while the oscilloscope and the logic analyzer are controlled and viewed in the main part of the window. The GUI will integrate the designs and functions of previous teams' projects to operate in one main application. When both the logic analyzer and the oscilloscope are run simultaneously, their waveforms will appear on one plot, allowing for more meaningful analysis of the relationship between the different waveforms. Additionally, to make the GUI more straightforward and to avoid unnecessary clutter, it will allow users to deactivate and/or collapse entire functions (waveform generator, oscilloscope, logic analyzer) or any unused channels in

those functions. The goal of our GUI is to provide users with an intuitive and easy to use interface that still delivers the capability and accuracy needed by ECE students.

3.3 Alternative Approaches

In coming to the decision to use the STM32-H562RGT6, we considered other STM32 MCUs, particularly the G474RE, H723VGT6, and H533RET6. The G474RE was dropped from consideration due to the need to interleave its ADCs to reach the required 5MS/s. While the H723VGT6 has the fastest CPU by far, as well as excellent analog and digital components, the RAM is somewhat lacking compared to our choice MCU and it is roughly double in price. The H533RET6 was also considered due to the availability of relatively inexpensive development boards. However, unlike the other teams, we have decided to instead purchase the MCU as a standalone part and solder it to the same board as the analog components.

3.4 Background Knowledge

3.4.1 MCU

The microcontroller is a simple processor with expanded GPIO and analog capabilities included in its design for embedded applications. It is connected to an array of GPIO pins with differing purposes and capabilities. This project uses both ADCs and DACs. ADCs (analog-to-digital converters) take in an analog electrical signal from something like a thermometer or a photoresistor and convert it into a digital signal that can be processed by the microcontroller. DACs (digital-to-analog converters) perform an inverse function. They are supplied with a digital signal from the microcontroller and convert it into an analog electrical signal.

The CPU core onboard the microcontroller is responsible for executing instructions in the user's program. Memory on the microcontroller is broken into two categories, RAM and flash. Flash is slow to read and write, but is nonvolatile, meaning it retains its state when power is lost. Flash memory is often used to store program data. RAM is faster but volatile, meaning it is reset when the system powers down. It is also usually smaller in capacity due to its higher price. RAM is used to store temporary data and program variables so that the CPU can access them more easily. Timers in the microcontroller are used to time events, such as processing data from an ADC or enabling a clock signal for the microcontroller to communicate with the host system over USB.

3.4.2 Oscilloscope

An oscilloscope is an essential instrument used to measure fluctuations in voltage or current and provides a visualization and analysis of electrical signals over time. The primary function of an oscilloscope is to plot voltage against time, enabling users to observe waveform properties such as amplitude, frequency, rise time, and distortion.

Bandwidth

Bandwidth refers to the range of input frequencies that can be accurately measured and captured by an oscilloscope. To ensure proper signal capture, the bandwidth of the oscilloscope must be higher than the frequency of the signal being measured. The Rule of Five is commonly used, which suggests that the oscilloscope bandwidth should be a minimum of five times the highest frequency component of the signal. This minimizes errors due to bandwidth limitations, keeping errors within a ± 2% tolerance. For example, an oscilloscope with a 2MHz bandwidth can effectively measure signals with frequencies up to 500kHz. This is the bandwidth specification we are aiming to meet with our device.

Rise Time

Rise time is the oscilloscope's ability to recognize and capture the rising and falling edges of a signal. It is closely related to bandwidth and can be approximated using the formula

$$t_R = \frac{0.35}{B}$$

where *B* is the bandwidth of the oscilloscope. A faster rise time ensures more accurate signal presentation, particularly for digital and high-speed signals, where rapid transitions are critical.

Sample Rate

The sample rate is the number of data points, or samples, the oscilloscope captures per second. An adequate sample rate is crucial to accurately visualize the signal on the screen. The sample rate should be at least 2.5 times the highest frequency component of the signal to avoid under-sampling, which can lead to aliasing and inaccurate results.

Number of Channels

Oscilloscopes typically come with multiple channels, allowing users to simultaneously measure and compare multiple signals. Oscilloscopes with two or four channels are the most common, offering flexibility for various measurements scenarios in circuit testing and debugging.

Vertical Resolution

Vertical resolution refers to the oscilloscope's ability to distinguish between different voltage levels, and it is determined by the resolution of the oscilloscope's ADC. Typically measured in bits, the vertical resolution defines the number of discrete levels the oscilloscope can use to represent the input signal. With the STM32-H562RGT6 microcontroller, a 12-bit ADC channel provides 4096 distinct levels of resolution, which means the signal is divided into 4096 possible voltage values. A higher vertical resolution allows for more precise measurements

of small signal variations, making it essential for capturing fine details in low-amplitude signals or when observing small changes in signal voltage.

3.4.3 Arbitrary Waveform Generator

Frequency and Amplitude Range

The frequency range of an AWG defines the spectrum of frequencies it can generate, while its amplitude range determines the maximum and minimum voltages the device can output. These two parameters are crucial for testing and simulating various real-world scenarios. Typical AWGs are capable of generating frequencies ranging from a few Hz to several MHz, enabling the testing of both slow and high-speed circuits.

Waveform Resolution

Waveform resolution refers to the bit depth of a digital-to-analog converter (DAC) within the MCU, which determines the granularity of the waveform it generates. A higher bit depth, such as 12-bit or 14-bit, allows the AWG to generate smoother, more accurate output waveforms. In the case of this project, the 12-bit DAC onboard the MCU can resolve 4096 distinct voltage levels, providing sufficient accuracy for most applications.

Digital-to-Analog Converter (DAC)

The DAC is a critical component in the design of the AWG, converting digital waveform data into analog signals. The DAC determines the accuracy of the output waveform based on its resolution and sampling rate. The sampling rate of the DAC should be at least ten times the frequency of the waveform being generated to ensure high-fidelity waveform reproduction. Higher DAC resolutions and sample rates enable the AWG to output complex, high-quality waveforms suitable for a wide range of testing scenarios.

Operational Amplifier (OpAmp)

After the waveform is generated by the DAC, it is passed through an operational amplifier (opamp) to amplify the signal, adjusting the output voltage to desired levels. The opamp ensures that the waveform has the correct amplitude to drive external circuits. The gain of the opamp can be adjusted to provide a wide range of output voltages, such as ±5V to ±12V depending on the project requirements. For our project, the opamp circuit needs to be capable of outputting a large signal that can drive up to a 20mA load.

Number of Channels

Similar to oscilloscopes, AWGs often come with multiple output channels. In our case, we plan to include two 12-bit output channels in our design. This allows users to generate two independent waveforms simultaneously, which can be useful in testing dual-input systems or performing synchronous signal testing.

3.4.4 Logic Analyzer

Sampling Rate

The sampling rate of a logic analyzer defines how frequently it captures the state of the digital input signals. The sampling rate should be higher than the input digital signal itself for accurate performance. A higher sampling rate allows for more detailed observation of fast digital transitions and events.

HAL Drivers

The Hardware Abstraction Layer (HAL) drivers provide a simplified interface for managing the peripherals of the microcontroller, making it easier to configure hardware components like GPIO, timers, and interrupts. For the logic analyzer, the GPIO HAL driver is essential for configuring the input channels to capture digital signals, ensuring they are correctly set up for input with appropriate voltage levels (3.3V or 5V). Additionally, the timer HAL drivers play a crucial role in defining the sampling rate, ensuring that digital signals are sampled at regular intervals based upon user-defined settings.

Direct Memory Access (DMA)

DMA is vital for efficiently handling high speed data transfers in the logic analyzer. Instead of relying on the CPU to manage every sample, DMA allows digital signals captured by the GPIO to be automatically transferred to a memory buffer without interrupting the microcontroller's main tasks. This enables the system to capture high-frequency signals across multiple channels without data loss. DMA also allows for techniques like double-buffering, where one memory buffer is being filled while the other is processed, ensuring continuous data capture. This signicantly reduces CPU overhead, allowing the microcontroller to focus on other critical tasks such as managing the oscilloscope and AWG functionalities.

3.4.5 GUI

GUI is an abbreviation for Graphical User Interface. A GUI is the interface that a user interacts with to control the device. Accordingly, a GUI must have some set of input methods such as buttons, textboxes, and checkboxes to indicate allowable inputs as well as output methods such as text, graphs, and sound to provide feedback to the user on the result of their inputted actions. For our purposes, the GUI must be able to control the oscilloscope, the arbitrary waveform generator, and the logic analyzer from the host system by communicating with the microcontroller over USB.

3.5 Project Requirements Specification

3.5.1 Mission Requirements

The project shall develop an affordable USB multi-function lab instrument with two oscilloscope channels, two arbitrary waveform generators, and eight or sixteen logic analyzer channels to satisfy the requirements for the lab exercises of the ECE department. All systems used in this device shall be incorporated into a single custom PCB with the device being controlled via a GUI with communication occurring over USB.

3.5.2 Operational Requirements

Input/Output Requirements

- The device shall have two analog output channels for the AWG.
- The device shall have two analog input channels for the oscilloscope.
- There shall be at least eight logic analyzer channels.

External Interface Requirements

- The device shall be powered using the 5VDC supplied by a standard USB connection.
- The device shall communicate with a host computer over a standard USB connection.

Functional Requirements

Oscilloscope

- The oscilloscope will have a sampling rate of 5MS/s.
- The bandwidth shall be 2MHz.
- The maximum supported frequency shall be 500kHz.
- There shall be various trigger options available.

AWG

- The AWG will have a sampling rate of 5MS/s.
- The AWG will output 10V peak-to-peak.
- The peak load of the AWG will be 20mA.
- There shall be the ability to offset by ±5V.

Logic Analyzer

- The sample rate will be a maximum of 5MS/s.
- The device will support 3.3 and 5V logic.
- The device shall decode protocols such as I2C, SPI, UART, and CAN.

Technology and System-Wide Requirements

- The device should be implemented on a single custom PCB.
- The microcontroller will have at least two 12-bit ADCs and two DACs.
- The device shall be controlled via a GUI on a computer connected over USB.
- The GUI program shall be compatible with Windows, Linux, and macOS.
- The total cost of the device should not exceed $50 per unit for 1000 units.

4. System Design

4.1 Functional Decomposition

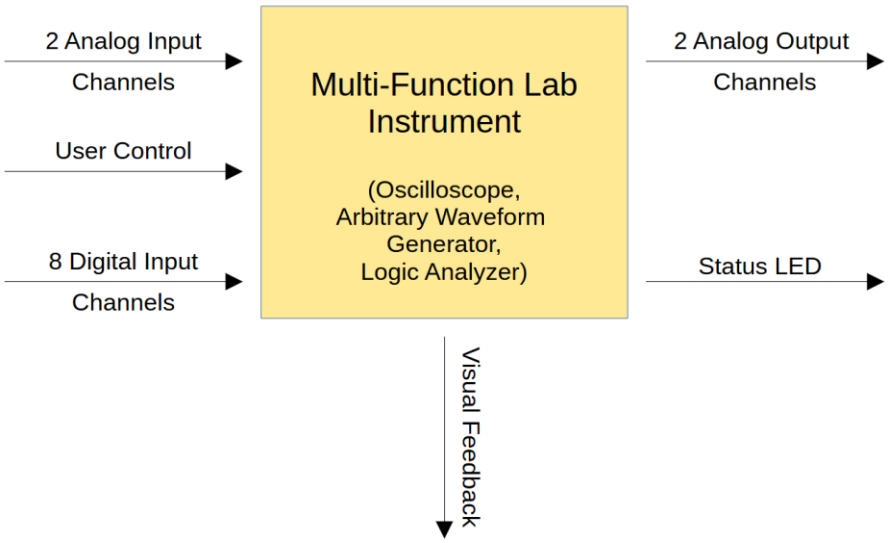


Figure 2: Level 0 Decomposition

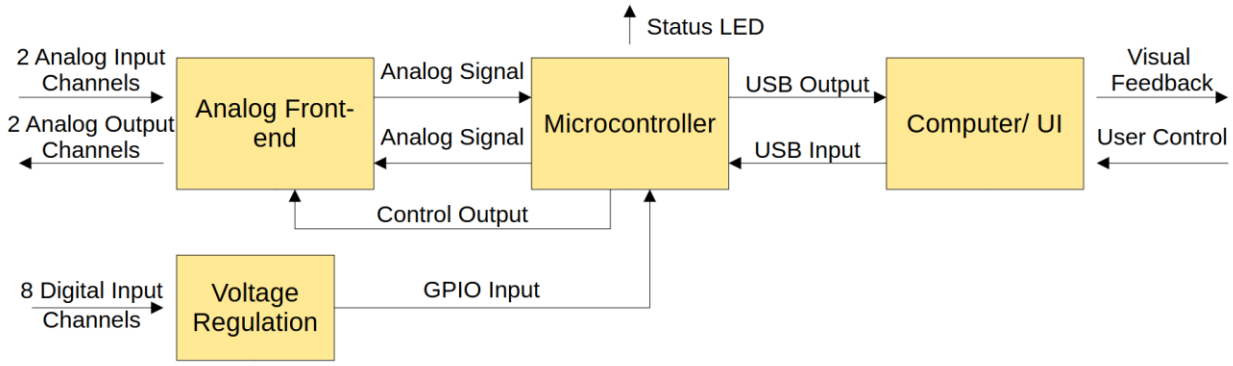


Figure 3: Level 1 Decomposition

4.2 Physical Architecture

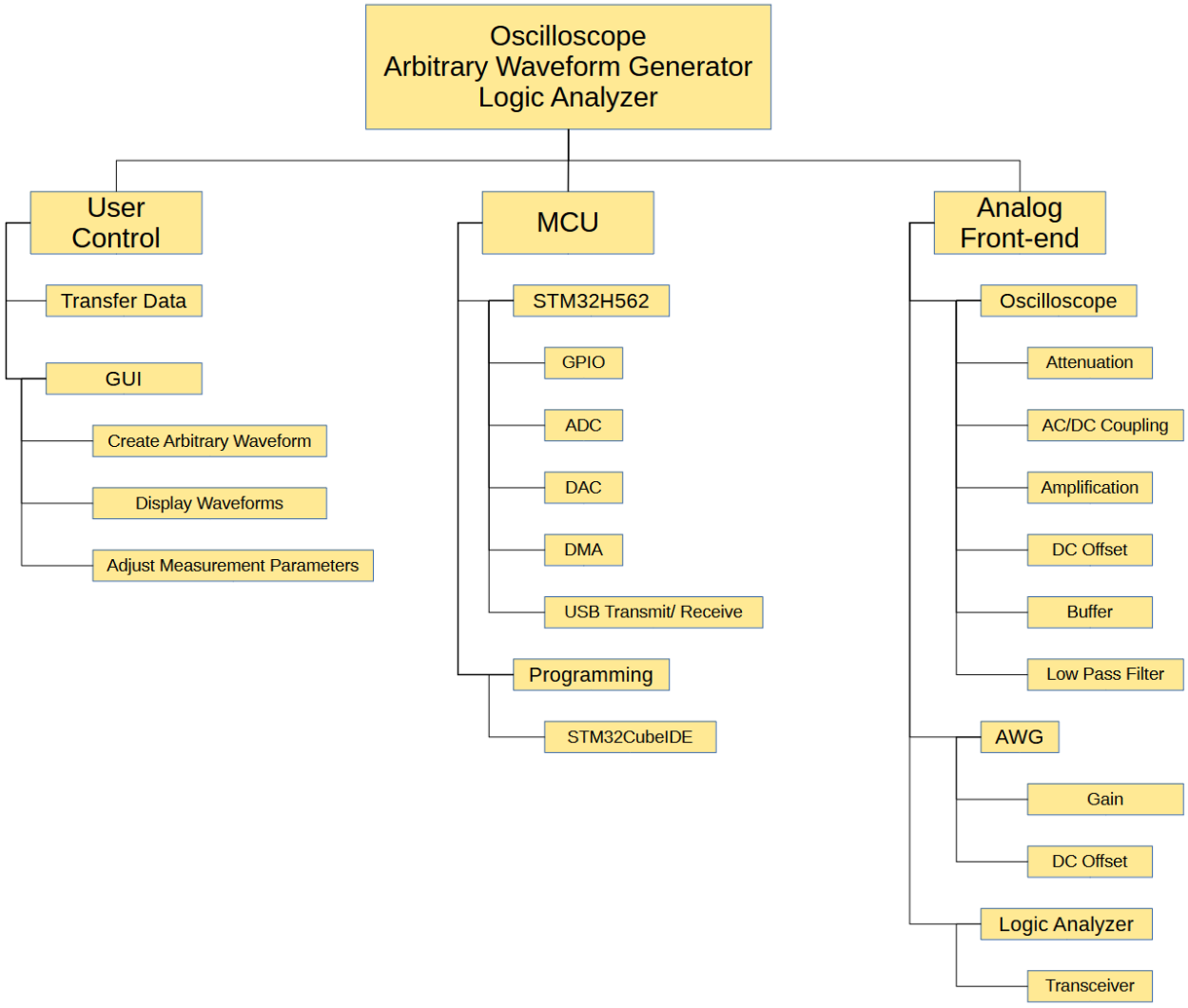


Figure 4: Physical Architecture

4.3 System Architecture

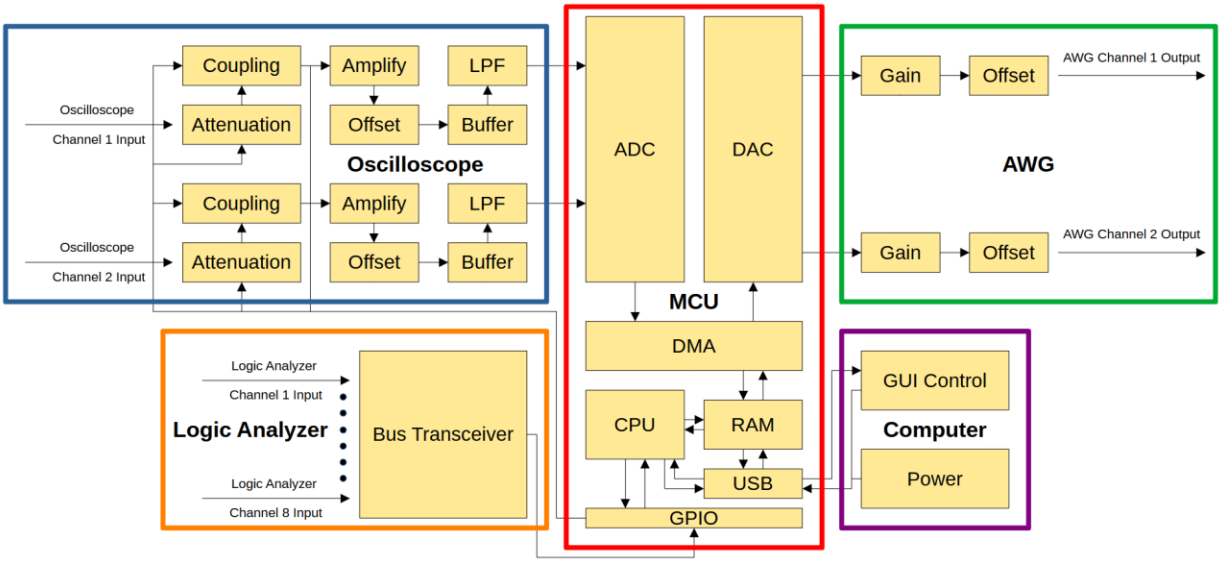


Figure 5: System Architecture

5. Preliminary Experimentation and Testing Plan

5.1 Preliminary Experiment

Testing for the device will be divided into three parts, the MCU, analog front-end, and the GUI. The GUI will be tested to ensure full functionality of all parameters. The MCU will be tested to ensure the ADCs, DACs, GPIO all work accordingly. Additionally, we will test the oscilloscope, waveform generator, and logic analyzer in accordance with the previous teams' tests to verify their functionality.

5.2 Experimentation Plan

5.2.1 MCU Experimentation

Test 1: Establish Communication with the MCU

A simple test program will be written and run on the MCU that performs a trivial operation and reports back to the host system. This will confirm, as a baseline, that the MCU as well as its USB link to the computer are working properly.

Test 2: GPIO Testing

Using a test program, relevant GPIO lines on the MCU will be tested. High and low logic levels will be evaluated using a logic analyzer and the behavior of associated components on the PCB will be compared with expected results.

Test 3: ADCs

The onboard ADCs will be tested to evaluate their functionality and accuracy. A reference voltage will be supplied using a suitable power source, and a test program will display the corresponding voltage detected by the ADCs. The input voltage will be adjusted through the range of the MCU's onboard ADCs, and the ADC readings recorded and analyzed at each step.

Test 4: DACs

The onboard DACs will be tested to evaluate their functionality and accuracy. A test program will have each DAC sweep through its range of output voltages. The output will be measured using an oscilloscope; the resulting waveform should resemble a sawtooth wave.

Test 5: Timers/PWM

A test program will utilize hardware timers onboard the MCU to generate a PWM signal. Its accuracy will be evaluated using an oscilloscope.

5.2.2 Analog Front-End Experimentation

Test 1: Signal Attenuation Test

The correct operation of the attenuation stage in the analog front end will be verified, ensuring that the high-voltage signals are properly scaled to levels suitable for the ADC. The output voltage after attenuation should match the expected scale value, with no more than 1% deviation.

Test 2: Amplification Test

This test will ensure that low-voltage signals are amplified correctly to levels suitable for the ADC. The amplified output should match the expected value within 1% accuracy, and the amplification should remain consistent across the tested input range.

Test 3: Low-Pass Filter Test

This test will confirm the low-pass filter removes high frequency electrical noise while passing signals within the desired bandwidth. Signals below the cutoff frequency should pass through with minimal attenuation, while signals above the cutoff should be attenuated by at least 3dB at the cut off, with attenuation increasing with higher frequencies.

Test 4: Voltage Protection Test

This test will validate that the voltage protection circuit prevents over-voltage from reaching the sensitive ADCs. The voltage protection circuit should successfully clamp the output voltage to the specified safe level, preventing any signal above the ADC's limit from reaching it.

Test 5: Total Harmonic Distortion Test

This test will measure the distortion level introduced by the analog front end, particularly the amplifiers and filters. The total harmonic distortion should be within acceptable limits across the range of tested frequencies and amplitudes.

Test 6: Frequency Response Test

In this test, the overall frequency response of the analog front-end will be evaluated to ensure that it meets the required bandwidth specifications for the oscilloscope. This will be accomplished by applying a sweep of input frequencies and measuring the response through a test program. The test program will plot the frequency response on a graph. The frequency response should remain mostly flat within the specified bandwidth, with a roll-off that matches the design specifications beyond the cutoff frequency.

Test 7: Signal Integrity Test

Ensure that the analog front-end preserves the integrity of the input signals without introducing excessive noise or distortion. A known clean, low-noise signal will be introduced as an input and a test program acting as an oscilloscope will monitor the signal-to-noise ratio to ensure minimal interference.

5.2.3 GUI Experimentation

Test 1: Communication Test

This test will ensure that the GUI control system running on the host computer is capable of communicating with the code running on the MCU. The connection will be made with a serial communication protocol such as UART.

Test 2: Interface Test

This test will ensure that the control interface contains all required controls for the implemented wave generator and oscilloscope functions. Additionally, the usability of the UI will be examined to ensure that it is understandable to a new George Mason electrical or computer engineering student with little to no instruction and limited experience.

Test 3: Control Test

This test will verify that the GUI is capable of controlling the oscilloscope and waveform generator. The interface should be able to issue commands to control the functionality of the oscilloscope and waveform generator. The MCU should send back data obtained from the oscilloscope, which should be graphed in the GUI.

Test 4: Compatibility Test

This test will ensure that the GUI software can run on Windows, Linux, and macOS without issues. The GUI software will be loaded onto computers running up-to-date versions of each operating system and the performance and stability of the software as well as its ability to interface with the hardware will be evaluated on each platform.

6. Preliminary Project Plan

6.1 Overview

This project will be conducted over the course of two semesters, with the project workload and tasks being partitioned into three main categories to allow team members to focus on a specific component. These categories are the microcontroller, analog front-end, and graphical user interface. Nonetheless, due to the size of the team, there may be some overlap in members' responsibilities. By the end of ECE 492, we expect to have developed a prototype PCB, with the oscilloscope, arbitrary waveform generator, and microcontroller being functional.

For the microcontroller, the team members will work on combining the code of the three projects and implementing them on the STM32-H562RGT6. This will involve becoming familiar with the previous teams' code as well as solving any conflicts that may exist with overlapping GPIO pins. The IDE of our choice for this project will be the STM32CubeIDE. Optionally, there is the Nucleo-H563ZI evaluation board with the same microcontroller which could be used for early testing and verification of systems during manufacturing of the custom PCB.

The work on the analog front-end will be going towards designing a custom PCB for the microcontroller, oscilloscope, waveform generator, and logic analyzer. Additionally, due to changes in components, the circuits may need to be re-evaluated in software such as Cadence PSPICE. These changes will be made due to the lack of availability of certain parts, cost-cutting measures to ensure the device stays under budget, and fixing problems causing the device to fail to meet the goals of the project. Like the previous teams, the new board will be created in KiCAD.

The GUI will be created using PyQtGraph, since it is easily adaptable to other operating systems and was used by the prior teams. The members working on this part will analyze the prior GUIs created by the other teams, and gain insight from the software created by industry leaders for existing products such as the AD2 and ADALM2000. From there, a new user-friendly GUI will be developed with all components and features integrated.

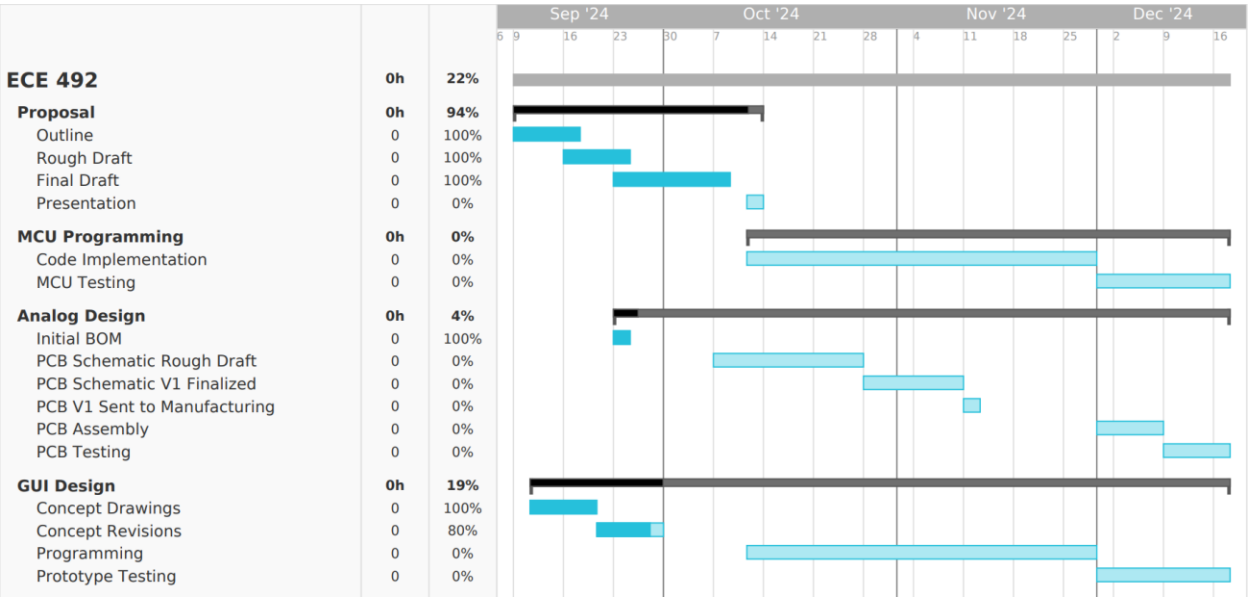
In ECE 493, any issues that may remain unresolved from ECE 492 will be addressed, and the logic analyzer functionality will be fully integrated into the device. The reason the logic analyzer will not be implemented in ECE 492 is that for the Fall 2024 semester it is in active development by a team in ECE 493. At the conclusion of this senior design project, we expect to

have a complete multi-function lab instrument on a custom PCB that is capable of meeting the needs of the ECE department.

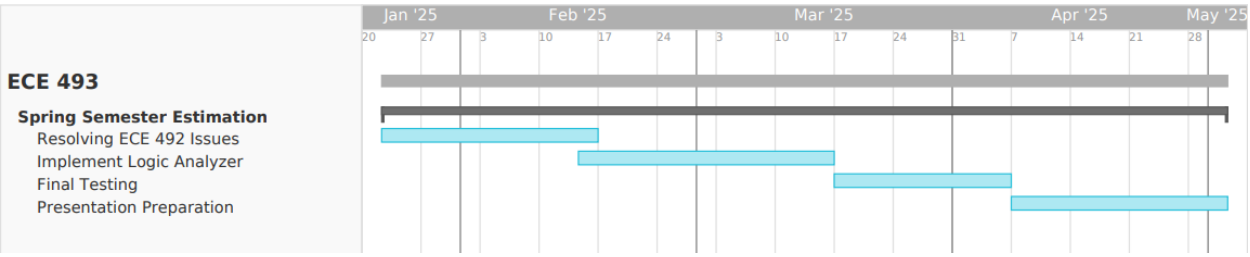
6.2 Allocation of Responsibilities

Team Roles	
Anh Nhat Van	Analog Systems Lead MCU PCB
Braden Camp	Team Lead MCU PCB Analog Front End
Timothy Kolos	MCU Lead AWG/Oscilloscope/Logic Analyzer Contact Analog Front End
Anna Mai	GUI Lead
Erick Orellana	GUI FS Contact

6.3 Detailed Project Timeline (ECE 492)



6.4 Estimated Project Timeline (ECE 493)



7. Potential Problems

7.1 PCB Design

The design of the PCB is complex since it must accommodate mixed analog and digital signal components which can complicate routing. Additionally, the physical placement of components is critical to reducing crosstalk, noise and ground loop issues. Managing high-frequency signals on a shared PCB can be particularly challenging, as a poor PCB layout can lead to performance degradation, increased electromagnetic interference (EMI), and inaccurate measurements.

Every aspect of the PCB design will need to be carefully considered before it is sent to manufacturing; potential problems that lie in the PCB will be further compounded by the fact that each PCB revision takes time to redesign, manufacture, ship, and assemble, adding to the project budget and timeline.

7.2 Signal Integrity and Noise

Since the device handles both high-frequency analog signals (for AWG and oscilloscope) and digital signals (for the logic analyzer) there is a risk of signal interference and noise between different subsystems. Additionally, an inadequate or faulty OpAmp can generate significant electrical noise and distortions which can interfere with the functionality of different systems.

7.3 Overload and Signal Clipping

High amplitude signals may exceed the input range of the ADC or maximum output range of the DAC, even with Low-pass filter in place, leading to signal clipping or distortion. Signal clipping can distort the waveform, leading to inaccurate measurements and faulty analysis. Overloading the inputs could also damage sensitive components.

7.4 MCU

While the STM32-H562RGT6 has met all requirements and is a very capable MCU, it uses a different ARM processor core than the F303 MCU utilized by the previous team, among other differences. This may present challenges with testing of the existing code, and the electrical

layout and design of the PCB may also need to be changed to accommodate the electrical and physical characteristics of the new MCU.

Additionally, unlike the previously used F303, the limitations of this microcontroller with respect to the proposed project have not been evaluated by previous teams. While the specifications of the microcontroller meet or exceed the project requirements on paper, practical testing may present unexpected challenges.

7.5 GUI

Although the GUI team members have some experience in creating GUIs, none have experience working with PyQtGraph specifically. Additionally, since the plan is to utilize GUI code created by the previous teams, bugs in the preexisting code may be inherited into the project. The reused code may also prove to be difficult to work with, which would necessitate abandoning this approach.

8. References

[1] P. Dong, D. Luong, J. A. J. M. Alsaei, B. Phan, G. Nguyen, T. V. M. Nguyen. “Affordable USB Oscilloscope Final Project Report,” B.S. thesis, Dept. Electrical and Computer Engineering, George Mason University, Fairfax, VA, USA, 2023.

[2] J. Schaeffler, G. Kuznetsov, H. Zainal, V. Arunachalam, Y. Gao, W. Denham. “Arbitrary Waveform Generator Final Project Report,” B.S. thesis, Dept. Electrical and Computer Engineering, George Mason University, Fairfax, VA, USA, 2023.

[3] “BitScope Micro Model 5: A tiny USB Mixed Signal Oscilloscope.” BitScope Designs. https://www.bitscope.com/product/BS05/ (accessed September 27, 2024).

[4] “Analog Discovery 3: USB Oscilloscope and Logic Analyzer.” Digilent. https://digilent.com/shop/analog-discovery-3/ (accessed September 27, 2024)

[5] A. Suciu. “ADALM2000 Overview.” Analog Devices Wiki. https://wiki.analog.com/university/tools/m2k (accessed September 27, 2024).

[6] Pico Technology Ltd, St. Neots, Cambridgeshire, United Kingdom. *Picoscope 2000 Series Oscilloscope Data Sheet,* 6th ed (2019). Accessed: September 27, 2024. [Online]. Available: https://www.picotech.com/download/datasheets/picoscope-2000-series-data-sheet-en.pdf

[7] "Using analog offset to maximize oscilloscope resolution," Pico Technology Ltd. https://www.picotech.com/library/application-note/using-analog-offset-to-maximize-oscilloscope-resolution (accessed September 24, 2024).

[8] “Analog Discovery 2.” Digilent. https://digilent.com/reference/test-and-measurement/analog-discovery-2/start (accessed September 27, 2024).

[9] A. S. Sedra, K. C. Smith, T. C. Carusone, and V. Gaudet, *Microelectronic Circuits*, 8th ed. New York, NY, USA: Oxford University Press, 2020.

[10] *ECE 334 Linear Electronics Lab I Laboratory and Experimental Guide*, Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA, 2015.

[11] *ECE 434 Linear Electronics Lab II Laboratory and Experimental Guide*, Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA, 2015.

[12] "OpenScope MZ: Getting started." Digilent. https://digilent.com/reference/test-and-measurement/openscope-mz/start (accessed September 27, 2024).