

OSCILLOSKETCH

By

Eric Vo

Josh Jenks

Final Report for ECE 445, Senior Design, Spring 2026

TA: Xiaodong Ye

05 May 2026

Project No. 22

Abstract

OscilloSketch is a handheld embedded system that generates coordinated X and Y analog voltages for oscilloscope XY mode, turning an oscilloscope into an interactive vector display. The final design supports four operating modes: Etch-a-Sketch drawing, preloaded vector shapes, Pong, and a music visualization mode with high-pass and low-pass filtering. The device uses rotary encoders and pushbuttons for user input, an ESP32-S3 for real-time control, a dual 12-bit digital-to-analog converter (DAC) for signal generation, and an analog output stage that converts 0–3.3 V DAC outputs into bipolar BNC outputs. It operates from a single USB-C input with regulated power rails and output protection. Testing showed ± 4.9 V X/Y outputs, 98 mA current draw, 19.76 ms input-to-output latency, and a 110 kHz coordinated update rate. The system met its main voltage, power, timing, and usability requirements.

Contents

1. Introduction.....	1
1.1 Problem.....	1
1.2 Solution.....	1
1.3 Functionality.....	1
1.4 Block Diagram.....	2
1.5 High-Level Requirements.....	3
2. Design.....	4
2.1 Mechanical Design.....	4
2.2 Hardware Design.....	4
2.2.1 Power Subsystem.....	4
2.2.2 Microcontroller Unit (MCU) Subsystem.....	6
2.2.3 User Input (UI) Subsystem.....	7
2.2.4 Digital to Analog Converter (DAC) Subsystem.....	8
2.2.5 Analog Output and Protection Subsystem.....	8
2.2.6 Additional Subsystem.....	9
2.3 Firmware.....	10
2.4 Design Alternatives.....	12
3. Design Verification.....	13
3.1 Power Subsystem Requirements & Verification.....	13
3.2 MCU / Firmware Subsystem Requirements & Verification.....	13
3.3 User Input Subsystem Requirements & Verification.....	14
3.4 DAC Subsystem Requirements & Verification.....	15
3.5 Analog Output Subsystem Requirements & Verification.....	15
4. Costs.....	16
4.1 Labor Cost Analysis.....	16
4.2 Parts Cost Analysis.....	16
4.3 PCB Cost Analysis.....	16
4.4 Grand Total.....	16
5. Conclusion.....	17
5.1 Accomplishments.....	17
5.2 Uncertainties.....	17
5.3 Ethical considerations.....	17
5.4 Future work.....	17
References.....	18
Appendix A Requirement and Verification Table.....	19

1. Introduction

This report first introduces the motivation and system-level goals for OscilloSketch, then describes the hardware, firmware, user input, DAC, power, and analog output subsystems. Later sections discuss verification results, cost, schedule, safety, standards, and ethical considerations. The main conclusion of the project is that the final device successfully met its core requirements: it produced approximately ± 4.9 V oscilloscope-compatible outputs, operated from USB-C at under 500 mA, achieved real-time user response, and updated the coordinated X/Y outputs fast enough to create a smooth and stable oscilloscope image.

1.1 Problem

Oscilloscope XY mode is useful for showing two-dimensional signal relationships, but it is often difficult for beginners to understand from static examples alone. Typical demonstrations may require multiple lab instruments, fixed waveform setups, or non-portable equipment, which limits hands-on interaction. This project addresses the need for a portable and intuitive system that can generate synchronized X and Y analog voltages for oscilloscope XY mode while responding to user input in real time.

1.2 Solution

OscilloSketch is a handheld embedded device designed to generate safe and stable bipolar X and Y voltages for oscilloscope XY mode while providing intuitive user interaction. The device functions like an Etch-a-Sketch: two rotary encoders control cursor position in real time, enabling continuous line drawing on an oscilloscope display.

The system is powered entirely from a single USB-C connection and outputs analog X and Y signals through BNC connectors for simple setup, portability, and easy interfacing with standard laboratory oscilloscopes. By letting users draw shapes, display preloaded patterns, play Pong, and visualize filtered audio, OscilloSketch helps demonstrate how changing two coordinated waveforms changes the resulting XY display. The design emphasizes deterministic timing, predictable user input, clean mixed-signal design, and protection at user-accessible connectors.

1.3 Functionality

OscilloSketch provides a portable way to generate the two analog signals needed for oscilloscope XY mode. A dual 12-bit digital-to-analog converter (DAC) creates high-resolution X and Y outputs, giving each axis 4096 possible voltage levels so drawings appear smooth instead of visibly stepped. Since the DAC only outputs 0–3.3 V, the op-amp output stage shifts and scales the signals to approximately ± 5 V while centering midscale near 0 V. This allows the oscilloscope trace to use a larger display area and makes the output suitable for XY visualization.

The device is powered from a single USB-C input, which supports the goal of making the system portable and easy to set up. Instead of requiring bench power supplies or multiple lab instruments, the user can power the device with a common USB-C cable and wall adapter. The X and Y outputs are sent through BNC connectors so the device can interface directly with standard laboratory oscilloscopes.

User interaction is handled through two rotary encoders and four pushbuttons. In Etch-a-Sketch mode, the encoders directly control the X and Y cursor position, allowing the user to draw on the oscilloscope in real time. This makes the system more engaging than a fixed waveform generator and helps users understand how changing the X and Y voltages changes the displayed position.

The final design also includes preloaded shapes, Pong, and music visualization modes. The preloaded shapes demonstrate how coordinated waveforms can form patterns, such as using two sine waves 90 degrees out of phase to create a circle. Pong demonstrates that the system can generate real-time interactive graphics in XY mode. The music mode allows users to hear and see the effect of low-pass and high-pass filtering, connecting signal processing concepts to both audio and visual changes on the oscilloscope.

1.4 Block Diagram

As shown in Figure 1 below, USB-C provides 5 V input power to the Power Subsystem and USB D+/D- signals for programming the ESP32-S3. The Power Subsystem generates the regulated 3.3 V digital rail for the microcontroller unit (MCU), DAC, encoders, and buttons, as well as the regulated +5 VA and -5 VA analog rails required by the op-amp output stages. The User Input Subsystem sends quadrature encoder signals and pushbutton inputs to the ESP32-S3, allowing the firmware to determine the selected mode and update the X/Y output coordinates in real time.

The ESP32-S3 processes user inputs and sends digital X/Y data to the dual 12-bit DAC over SPI. The DAC produces RAW_X and RAW_Y analog signals, which are shifted and amplified by the dual op-amp stage to create bipolar X and Y outputs for oscilloscope XY mode. These signals are routed to the BNC X and BNC Y connectors.

The additional subsystem supports audio visualization and Z-blanking. In audio mode, the RAW_X and RAW_Y signals from the DAC are routed into the audio op-amp stage, where they are conditioned for audio output and sent to the audio jack as left and right audio signals. For Z-blanking, the ESP32-S3 generates a control signal that passes through a 3.3 V to 5 V level shifter and is routed to the BNC Z connector, allowing the device to control the beam intensity on compatible oscilloscopes.

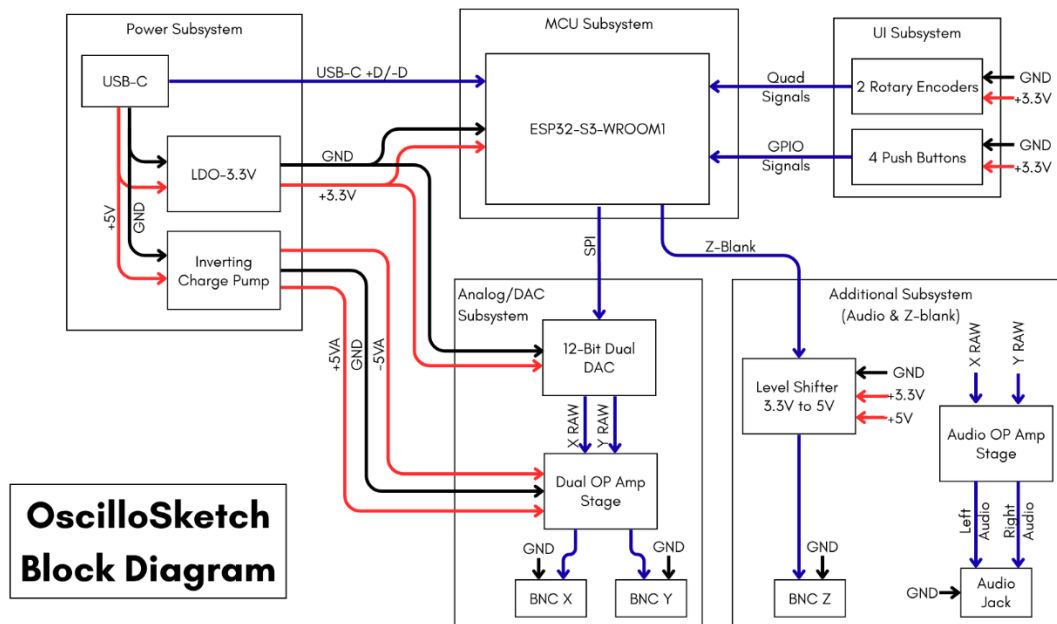


Figure 1: Block diagram of OscilloSketch

1.5 High-Level Requirements

The project is considered successful only if the system meets all of the following.

1. The system shall generate two independent continuous analog output voltages with at least 12-bit resolution and an output range near plus and minus 5 V suitable for oscilloscope XY mode.
2. The system shall operate entirely from a single 5 V USB-C connection and shall draw less than or equal to 500 mA while supporting both device power and firmware programming.
3. The system shall provide real-time Etch a Sketch style control, where two rotary encoders continuously control the X and Y cursor position with input to output latency less than or equal to 20 ms, and where motion appears smooth on the oscilloscope during normal use.
4. The design target is an output update rate of at least 10,000 coordinated X-Y points per second.

Together, these requirements ensure that the device meets its main goals: producing oscilloscope-compatible analog outputs, remaining portable and easy to power, and providing smooth interactive control for drawing and waveform visualization.

2. Design

2.1 Mechanical Design

The enclosure is 3D printed with 3 mm wall thickness and is designed to fit the printed circuit board (PCB) snugly while still leaving a 0.3 mm tolerance gap for assembly. As shown in Figure 2, the PCB layout on the left shows the placement of the external components, while the enclosure model on the right shows how the board should fit inside the housing. The PCB, approximately 98.5 mm × 83 mm, is secured using four mounting holes and standoffs to prevent board flex. The top of the enclosure exposes the rotary encoders and pushbuttons for user interaction, while the side cutouts align with the 3 BNC connectors, audio jack, and USB-C port. The BNC connectors are panel-mounted to reduce stress on the PCB during cable insertion and removal. An acrylic cover is used on top of the enclosure to protect the internal electronics while still allowing the board and components to remain visible.

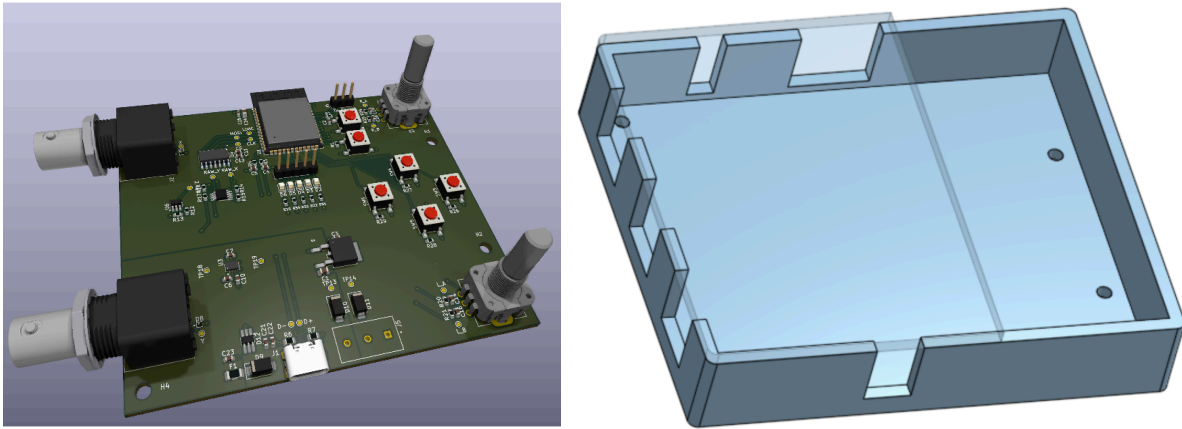


Figure 2: 3D-PCB layout v1 and 3D-printed enclosure v2 design. The left image shows the PCB with external controls and connectors, while the right image shows the enclosure with cutouts for board mounting and cover.

2.2 Hardware Design

The major component selections and design calculations were based on manufacturer datasheets and application notes for the ESP32-S3 module [1], MCP4922 DAC [2], LM27762 charge pump [3], OPA2192 and TLV9061 op-amps [4], [5], PEC11R rotary encoders [6], AZ1117C LDO [7], Tektronix 2225 oscilloscope [8], IEEE Code of Ethics [9], TI op-amp output-scaling guidance [10], and TI USB-C power guidance [11].

2.2.1 Power Subsystem

The power subsystem accepts 5 V from the USB-C input and generates the regulated supply rails required by the rest of the system. The 3.3 V rail powers the ESP32-S3, DAC, and digital logic, while the +5 V and -5 V analog rails power the op-amp output stage. These rails allow the system to produce stable bipolar X and Y outputs for oscilloscope XY mode.

As shown in Figure 3, the USB-C connector was chosen because it provides both a convenient 5 V power source and USB D+ and D- signals for programming the ESP32-S3. Since USB-C chargers and cables are widely available, the device can be powered without requiring a separate bench supply during normal use. The CC1 and CC2 pins use 5.1 k Ω pull-down resistors to identify the board as a USB-C power sink,

because the oscilloscope input is high impedance and the op-amps only need to drive small signal currents during normal operation. These ± 5 V rails allow the analog stage to swing above and below ground, which is required for the final ± 5 V oscilloscope outputs.

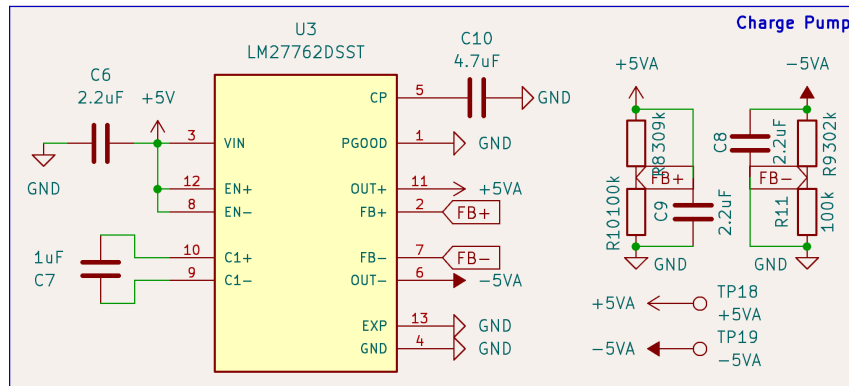


Figure 4: Schematic diagram of the Charge Pump IC and Circuit

As shown in Figure 5, the 3.3 V rail is generated using the AZ1117C-3.3 LDO, which regulates the 5 V input for the ESP32-S3, DAC logic, and user input circuitry [7]. An LDO was chosen because it provides a simple and low-noise regulation method for the digital rail. The AZ1117C can support up to about 1 A of output current, which provides more than enough current for the low-power digital components on the board, about 100 mA. The input capacitor helps support sudden load changes, while the output capacitor improves regulator stability and reduces output ripple.

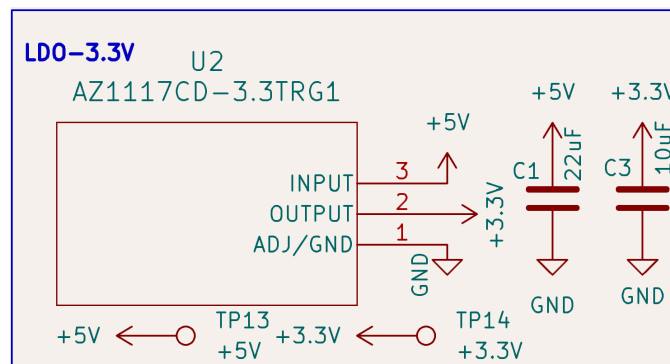


Figure 5: Schematic of LDO from 5 V to 3.3 V

2.2.2 Microcontroller Unit (MCU) Subsystem

The OscilloSketch uses the ESP32-S3-WROOM-1-N16 as the main microcontroller. This module was chosen because it provides a strong combination of processing speed, memory, native USB support, and available GPIO. The ESP32-S3 has a dual-core processor running up to 240 MHz, which provides enough processing headroom for reading user inputs, generating X-Y coordinate data, and updating the DAC at a high refresh rate [1]. The N16 version includes 16 MB of flash memory, giving the system extra space for firmware, stored drawing data, preset vector shapes, and future feature expansion [1].

As shown in Figure 6, the ESP32-S3 is powered from the regulated 3.3 V rail and connects to the rest of the system through user input, USB, and DAC control signals. Native USB allows the same USB-C connection to be used for programming and serial debugging, removing the need for a separate USB-to-UART programming chip. The microcontroller receives signals from the two rotary encoders and pushbuttons so the user can control the X and Y drawing position and switch between operating modes. It also communicates with the dual 12-bit DAC through SPI using SPI clock, MOSI, chip select, and LDAC signals. The SPI interface sends the digital X and Y values to the DAC, while LDAC allows both DAC channels to update together for coordinated oscilloscope XY output.

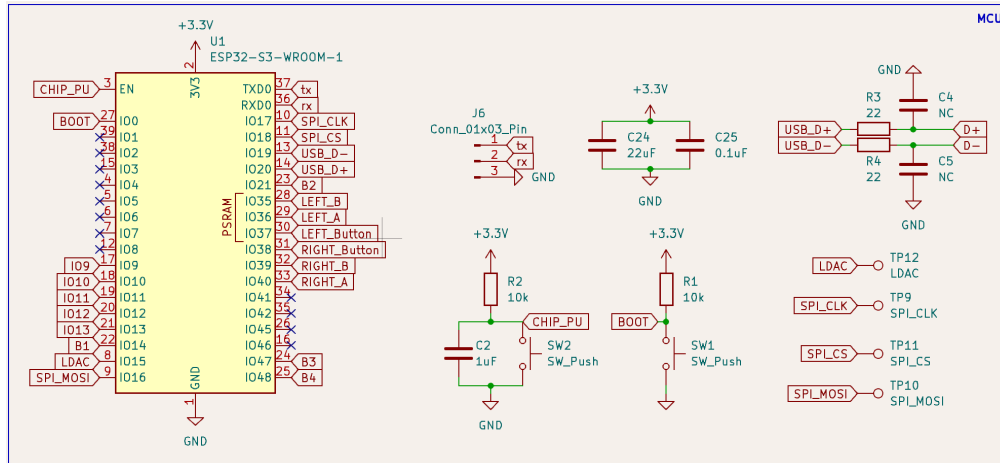


Figure 6: Schematic of the ESP32-S3-WROOM-1 and its connections

2.2.3 User Input (UI) Subsystem

Figure 7 illustrates the user input subsystem, which consists of two PEC11R rotary encoders and four pushbuttons [6]. The left side of Figure 7 shows the four active-low pushbuttons, labeled B1 through B4, which are used for functions such as clearing the drawing or switching operating modes. Each button uses a 10 kΩ pull-up resistor to 3.3 V, so the GPIO normally reads high and is pulled low when the button is pressed.

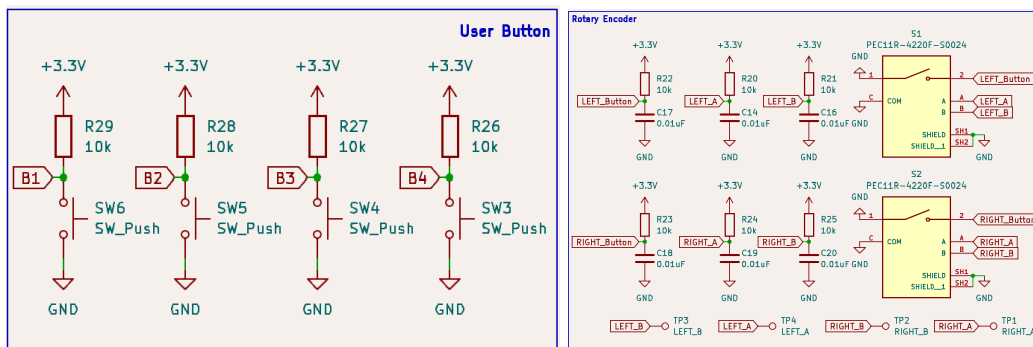


Figure 7: Schematic of four buttons (on left) and two rotary encoders (on right)

The right side of Figure 7 shows the two PEC11R rotary encoders, one for the X-axis and one for the Y-axis. Each encoder provides quadrature A/B signals, allowing the ESP32-S3 to determine both rotation direction and movement amount [6]. The encoder lines use 10 kΩ pull-up resistors and 0.01 μF capacitors to reduce contact bounce and noise before being read by the microcontroller. Each rotary encoder also includes an integrated pushbutton, which is used to change the operating modes of the OscilloSketch.

2.2.4 Digital to Analog Converter (DAC) Subsystem

Figure 8 shows the MCP4922 dual 12-bit DAC used to generate the raw X and Y analog signals. This DAC was chosen because it provides two independent 12-bit outputs in one package, allowing OscilloSketch to control both oscilloscope axes using a single device [2]. The DAC is powered from the regulated 3.3 V rail, producing raw output voltages from 0 V to 3.3 V. The MCP4922 also has a typical settling time of about 4.5 μs, which is much faster than the 100 μs period required for a 10 kHz update rate [2]. This helps ensure that the DAC output can settle quickly between updates and does not limit the refresh rate of the displayed XY drawing.

The ESP32-S3 communicates with the MCP4922 using SPI through the clock, data, and chip select signals. The two DAC outputs are labeled RAW_X and RAW_Y, representing the unamplified X and Y signals before the analog output stage converts them into the final ±5 V oscilloscope outputs. The LDAC signal is used to update both DAC channels at the same time. The ESP32-S3 first sends the new X and Y values over SPI, then pulses LDAC so both outputs change together. This is important for XY mode because it prevents one axis from updating before the other, helping keep the displayed drawing stable and coordinated.

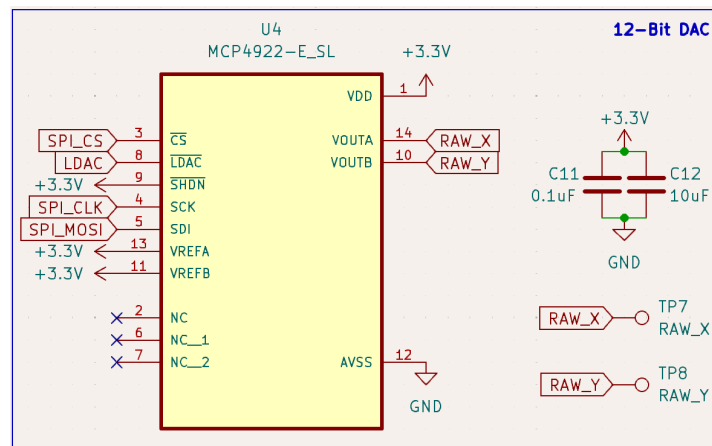


Figure 8: Schematic of the 12-Bit DAC

2.2.5 Analog Output and Protection Subsystem

Figure 9 shows the analog output and protection stage, which converts the MCP4922 raw DAC outputs, RAW_X and RAW_Y, from 0–3.3 V into approximately ±5 V signals centered near 0 V. The OPA2192 dual-precision op-amp is used so that one channel can process the X output and the other can process the Y output, with similar gain and offset behavior [4].

The output stage uses a level-shifted non-inverting amplifier based on TI's op-amp output-scaling approach [10], with the transfer function:

$$V_{out} = \left(1 + \frac{R_f}{R_{ref}}\right) \cdot V_{in} - \left(\frac{R_f}{R_{ref}}\right) \cdot V_{ref} = 3.0287 \cdot V_{in} - 4.996$$

Using $R_f = 19.8 \text{ k}\Omega$, $R_{ref} = 9.76 \text{ k}\Omega$, and $V_{ref} = 2.46 \text{ V}$, the level-shifted amplifier maps the 0–3.3 V DAC output to approximately -5 V to +5 V at the BNC output. Specifically, 0 V maps to about -5 V, 1.65 V maps to about 0 V, and 3.3 V maps to about +5 V. The 2.46 V reference is generated using a resistor divider and buffered by the TLV9061 [5]. Buffering the reference provides a stable, low-impedance voltage for the level-shifting circuit, which is important because any change in the reference directly affects the output offset.

Because the MCP4922 is a 12-bit DAC, the raw step size is $3.3 \text{ V} / 4096 = 0.806 \text{ mV}$. After the analog gain stage, the effective output step size is approximately 2.44 mV per code, giving smooth movement on the oscilloscope display.

Output protection is included at each BNC connector using a series resistor and a bidirectional TVS diode. The series resistor limits current if the output is accidentally connected to a low-impedance load, while the TVS diode protects the op-amp from ESD and transient events. Since the intended oscilloscope input is $1 \text{ M}\Omega$, the series resistor has very little effect during normal operation.

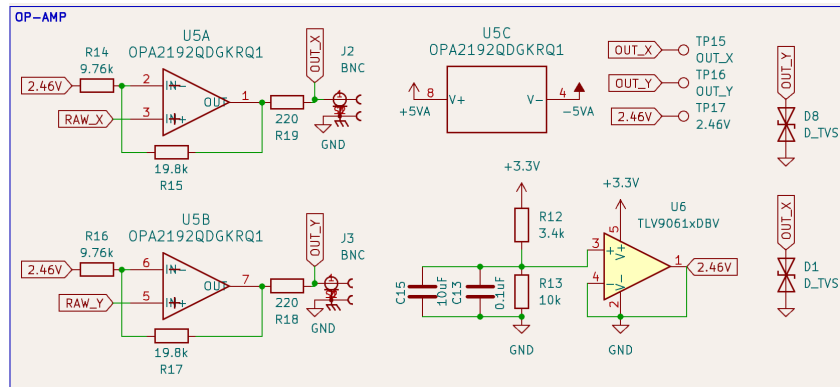


Figure 9: Schematic of the Analog Output and Protection

2.2.6 Additional Subsystem

In Figure 10 on the left, the audio subsystem provides an additional output path for the generated RAW_X and RAW_Y signals through the left and right channels of a 3.5 mm audio jack, respectively. Each signal first passes through a $10 \text{ k}\Omega / 10 \text{ k}\Omega$ resistor divider, reducing the 0–3.3 V signal range to approximately 0–1.65 V. For a full-scale sine wave, this corresponds to:

$$V_{RMS} = \frac{1.65}{2 \cdot \sqrt{2}} = 0.58 \text{ V}_{rms}$$

This value was chosen to stay near the lower end of typical line-level audio, which ranges from 0.5 V_{rms} to 1.2 V_{rms}, and reduce the chance of clipping. A $1 \mu\text{F}$ AC-coupling capacitor removes the DC offset

before the signal reaches the audio jack. A 68 kΩ series resistor is also included to limit current and provide protection if the output is shorted or connected to an external device.

The Z-blanking subsystem, as shown in Figure 10 on the right, provides an optional control signal for oscilloscope beam blanking. The blanking signal is generated by an MCU GPIO pin, but since the MCU uses 3.3 V logic, an SN74LV1T34DCK level shifter is used to translate the signal to 5 V logic for compatibility with the analog oscilloscope blanking input. The level shifter can operate at 50 MHz, so it provides sufficient timing margin for enabling and disabling the beam during XY drawing. A series resistor and TVS diode are included at the BNC output for current limiting and protection.

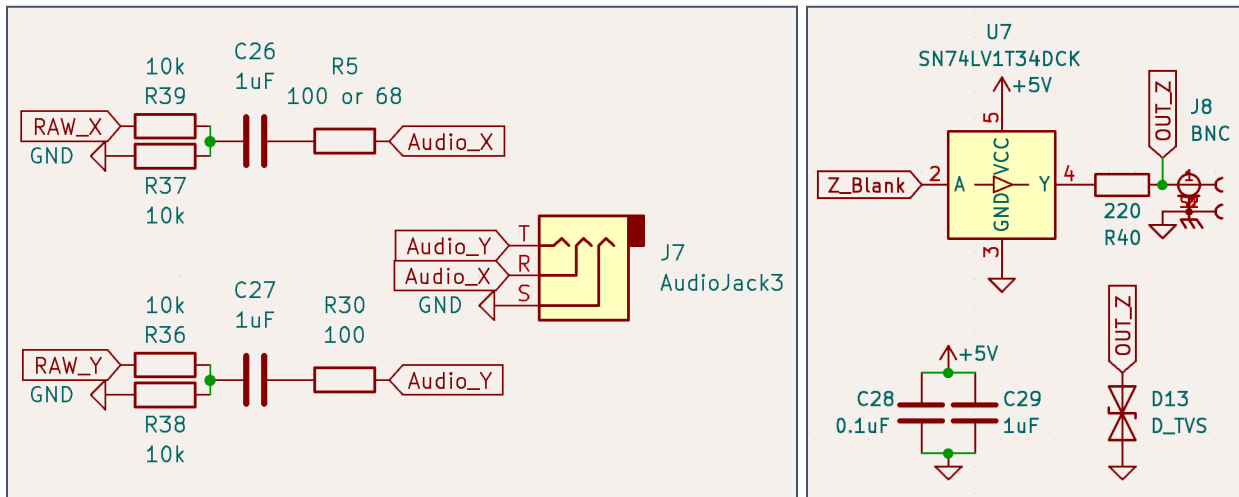


Figure 10: Schematic of the Audio Subsystem (on left) and Z-Blanking Subsystem (on right)

2.3 Firmware

The firmware controls the real-time behavior of OscilloSketch by reading user inputs, generating X/Y coordinate data, and continuously updating the dual DAC outputs. Since an analog oscilloscope does not store a digital frame buffer, the displayed image only remains visible if the firmware repeatedly redraws the generated vector points at a high rate. For this reason, the final firmware separates low-speed mode and input logic from the high-speed replay and DAC output path. The main program loop handles user input, mode switching, and coordinate generation, while a dedicated replay task on a separate CPU core continuously sends synchronized X/Y points to the DAC.

The firmware is organized around a shared replay structure used by all display modes. Each mode generates coordinate data differently, but the final output path is the same for every mode. Etch-a-Sketch mode stores user-generated X/Y points in a drawing path, Shape Demo mode generates predefined vector shapes, Pong mode renders a dynamic game frame, and Audio Mode generates X/Y audio visualization points. These points are passed to the shared replay layer, which selects the next point to output and sends it to the MCP4922 DAC. This modular structure prevents each mode from needing its own timing-sensitive DAC code and keeps the output behavior consistent across the entire system.

The high-speed replay task is the most important timing component of the firmware. Early firmware versions used a slower software timer approach, but this caused visible retracing after longer drawings because stored points could not be replayed fast enough. The final implementation uses the ESP32-S3 dual-core processor to separate replay from the main input and mode control logic. The replay task runs at approximately 110 kHz, meaning the system outputs about 110,000 coordinated X/Y points per second. For each replay step, the firmware writes the X and Y DAC codes over SPI, then pulses LDAC so both DAC channels update together. This synchronized update is necessary because oscilloscope XY mode depends on the X and Y voltages representing the same point in time.

Etch-a-Sketch mode is the main interactive drawing mode. The two rotary encoders control the X and Y cursor position, and the firmware appends the resulting positions to the stored drawing path. The cursor position is clamped within safe DAC bounds so the analog output stage does not operate too close to the supply rails. The firmware also interpolates between successive user positions before storing them, which makes drawn lines appear smoother instead of visibly jumping between encoder steps. The stored path is replayed continuously by the high-speed replay task so that the full drawing remains visible on the oscilloscope screen while the user continues drawing.

The additional display modes demonstrate that the same firmware architecture can support more than basic drawing. Shape Demo mode generates preset vector patterns, such as simple geometric shapes, to show how coordinated X/Y signals create stable oscilloscope figures. Pong mode uses real-time game logic to update paddle positions, ball motion, collisions, scoring, and vector frame rendering. This mode is more complex than the static shape mode because the displayed frame changes continuously based on user input and game state, but it still uses the same replay and DAC output path as the other modes.

Audio Mode uses a precomputed pulse-code modulation (PCM) song snippet stored on the ESP32-S3 rather than live serial audio streaming. The audio samples are converted into X/Y output points and replayed locally, which avoids the packet loss and underrun issues encountered during development of the live streaming implementation. The firmware also includes synthetic audio visualization presets, allowing the user to switch between the stored song and generated signal patterns. In Audio Mode, the rotary encoders control first-order low-pass and high-pass filter cutoff frequencies. These filters change the audio-derived X/Y waveform before output, allowing users to observe how filtering changes both the sound and the oscilloscope display.

The firmware also supports Z-blanking through replay metadata. Each replay step can mark whether the beam should be blanked before moving to the next point. When a blanking event is requested, the firmware briefly drives the Z-blanking GPIO signal, which is level-shifted to the oscilloscope's external Z input. This allows the device to reduce visible retrace lines between separated vector segments on compatible oscilloscopes. Although the visual effect was subtle during the final demo, the Z-blanking path was active and integrated into the replay system.

Overall, the firmware design turns the hardware into a flexible real-time vector display system. The main design decision was to keep application logic, point generation, and high-speed replay separate. This made it possible to support multiple operating modes while still meeting the timing requirement for

smooth oscilloscope output. The final 110 kHz replay rate exceeded the original 10,000 point/s target and gave the system enough update bandwidth for freehand drawing, preset shapes, Pong, and music visualization.

2.4 Design Alternatives

During the design process, most major issues were caught before final assembly. One issue was found in the original op-amp output-stage calculations. The first resistor calculations did not correctly map the DAC's 0–3.3 V output range to the desired ± 5 V BNC output range. This would have caused the oscilloscope output to be incorrectly centered or scaled. The amplifier equation was recalculated using TI's application report before the first PCB order was placed, and the final resistor values were selected and ordered to correctly map the DAC output to approximately ± 5 V at the BNC outputs [10].

A second issue was found during review of the first PCB layout. In the first board revision, one trace was routed too close to the encoder footprint and created a potential short between encoder signals. This was corrected in the second PCB revision by rerouting the trace and improving clearance around the encoder pins. Since the first PCB revision was never soldered, this issue did not affect the final assembled hardware. Overall, the major design inconsistencies were resolved before the final PCB revision, and no unresolved hardware issues prevented the system from meeting its main output requirements.

A major firmware design issue was the replay rate needed to keep oscilloscope drawings visually persistent. Early firmware versions used a software-timer replay approach, but this limited the practical update rate and caused visible retracing as drawings became longer. The replay system was first moved toward hardware-timed output, and the final implementation separated the high-speed replay task from the main user-input and mode-control logic using the ESP32-S3 dual-core processor. This corrected the timing bottleneck and allowed the system to reach approximately 110 kHz, well above the 10,000 point/s design target.

Another firmware design tradeoff occurred in Audio Mode. Real-time serial audio streaming was partially functional but suffered from packet loss and start-stop playback, so the final demonstration used a precomputed PCM audio snippet stored on the ESP32-S3 for reliable playback and filtering.

3. Design Verification

3.1 Power Subsystem Requirements & Verification

The power subsystem was verified by measuring the main regulated rails while the device was operating to ensure that each part of the subsystem produced the correct output voltage. As shown in Table 1, the 3.3 V digital rail measured 3.303 V, while the analog rails measured +4.998 V and -4.877 V. All measured values were within their required ranges, confirming that the subsystem provided stable supply voltages for both the digital logic and analog output stage.

The input protection requirement was also included in the R&V table, but the polyfuse was not tested by intentionally shorting the assembled board because this could have damaged other components. Instead, the protection was verified by schematic inspection, continuity testing, and review of the polyfuse datasheet, as stated in the Table 1 results. The resettable 1 A polyfuse is placed in series between the USB-C 5 V input and the board's 5 V rail, so any overcurrent event on the protected 5 V rail must pass through the fuse. If a short occurs on the board side of the 5 V rail, the fuse will increase its resistance and limit the current supplied from USB, thereby protecting it.

As part of the high-level requirements, the device also needed to operate from a single USB-C connection while drawing less than 500 mA. This requirement is related to the power subsystem because USB-C provides the main 5 V input used to generate the digital and analog supply rails. During testing, the USB-C input measured 5.23 V, which is within the expected USB supply range. The device current remained around 97 mA while powered at 5 V with a power supply, confirming that the system could be powered and programmed through USB-C without requiring an external power supply.

3.2 MCU / Firmware Subsystem Requirements & Verification

The MCU/firmware subsystem was verified by testing whether the ESP32-S3 could update both DAC outputs fast enough, communicate reliably over SPI, run continuously, and start in a safe output state. As shown in Table 2, the measured update rate was 110 kHz, which is well above the required 10,000 X-Y points/s. This confirms that the firmware timing and DAC update process were fast enough to produce a stable and smooth oscilloscope image.

The SPI communication was also verified using an oscilloscope or logic analyzer. The measured SPI clock was 20.0 MHz, as shown in Figure 11 on the left, which exceeds the required 5 MHz minimum. This ensured that both X and Y DAC values could be transmitted within each update period. Firmware reliability was verified by running the drawing or waveform mode continuously for at least five minutes, during which the system did not freeze or reset. Finally, startup behavior was tested by power cycling the device while measuring the BNC outputs. In Figure 11 on the right, a small voltage dip is visible during startup, but the output quickly returns to approximately 0.001 V. This shows that the firmware initializes the DAC outputs near 0 V rather than driving them to the positive or negative rail during power-up. Overall, the MCU/firmware subsystem met its timing, reliability, and startup safety requirements.

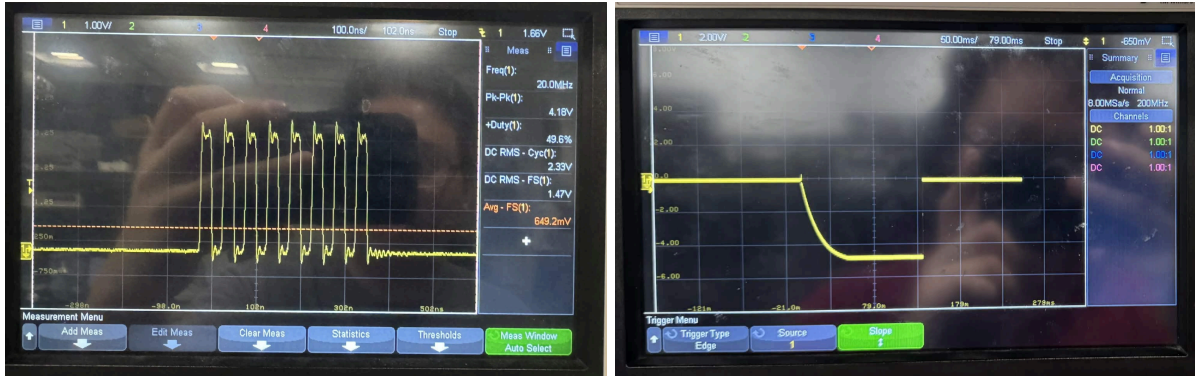


Figure 11: Oscilloscope measurements of the SPI clock at 20.0 MHz (left) and MCU startup output behavior (right)

3.3 User Input Subsystem Requirements & Verification

The user input subsystem was verified to ensure that the encoders and buttons provided intuitive real-time control of the XY drawing. The X and Y encoder directions were verified visually in oscilloscope XY mode by confirming that each encoder moved the trace along the correct axis and direction. As shown in Table 3, encoder scaling was then tested by rotating each encoder 20 detents and measuring the corresponding BNC output change. The X encoder produced a 4.06 V change, and the Y encoder produced a 3.86 V change, both within the required $3.9 \text{ V} \pm 0.2 \text{ V}$ range. This confirmed that the two encoders had similar voltage scaling and did not noticeably differ from each other.

Button response time was tested to confirm that user inputs felt real-time during operation. The required latency was 20 ms or less, which was chosen to be much faster than typical human reaction time, so the controls would appear immediate. As shown in Figure 12, the measured latency was 19.20 ms, meeting the requirement. Overall, the user input subsystem met its axis control, encoder scaling, and response-time requirements.



Figure 12: Oscilloscope measurement of user input latency, showing the delay between the input event and system response

3.4 DAC Subsystem Requirements & Verification

The DAC subsystem was verified to ensure that the MCP4922 produced the expected raw X and Y signals before the analog output stage. As shown in Table 4, the RAW_X and RAW_Y outputs were measured with a multimeter at minimum and maximum DAC codes. RAW_X measured 0.002 V to 3.30 V, and RAW_Y measured 0.001 V to 3.29 V, confirming that both DAC channels used nearly the full 0–3.3 V range.

The DAC outputs were also checked for monotonic behavior by sweeping the DAC codes upward and confirming that the output voltage increased smoothly without unexpected drops. This measurement is shown in Figure 13 on the left and confirms that the DAC output does not curve noticeably, drop, or become limited as the code increases. The LDAC timing was also verified to confirm that X and Y update at nearly the same time, as shown in Figure 13 on the right. The measured delay between RAW_X and RAW_Y updates was only 560 ns, which is small compared to the system update period. Overall, the DAC subsystem met its range, monotonicity, and simultaneous update requirements.



Figure 13: Oscilloscope measurements of the DAC code sweep showing smooth increasing output behavior (left) and RAW_X/RAW_Y update timing with a 560 ns delay between channels (right)

3.5 Analog Output Subsystem Requirements & Verification

The analog output and protection subsystem was verified by measuring the final BNC outputs after the DAC and op-amp stages. As shown in Table 5, the X and Y outputs were first checked at midscale code 2048 to confirm that the drawing would be centered on the oscilloscope. The measured offsets were 0.0039 V for X and 0.0022 V for Y, both well within the required ± 0.10 V range.

The output step size was also verified to ensure that small changes in DAC code produced smooth motion on the display. The measured value was 2.41 mV/code, which is close to the expected 2.44 mV/code and within the allowed tolerance. The output stage was verified with the intended 1 M Ω oscilloscope input, matching the high-impedance input configuration used for the Tektronix 2225 oscilloscope during XY-mode testing [8]. Overall, the analog output subsystem met its centering, step-size, update-rate, and stability requirements.

4. Costs

4.1 Labor Cost Analysis

The primary project cost is engineering labor. Using an estimated starting salary of \$80,000/year for a University of Illinois ECE graduate and 2080 work hours/year, the hourly rate is approximately \$38.46/hour. Per course guidelines, labor cost is calculated as:

$$(\$/\text{hour}) \times 2.5 \times (\text{hours worked}) = 38.46 \times 2.5 \times 112 \approx \$10,768$$

Assuming each partner contributed about 112 hours over the 16-week semester, the labor cost per partner is approximately \$10,768. For two partners, the total estimated labor cost is approximately \$21,538. As expected for a prototype development project, labor is the dominant cost and is much greater than the hardware material cost.

4.2 Parts Cost Analysis

The DigiKey cart total was \$58.83, including shipping and applicable tariffs, as shown in Table 6. Purchased parts included the MCP4922 DAC, OPA2192 op-amp, PEC11R rotary encoders, voltage regulation components, protection devices, and supporting passives. In the Component Cost table, parts without an asterisk were purchased through DigiKey, while asterisked parts were sourced from the ECEB Supply Center, eShop, or self-service lab inventory.

Although these course-supplied parts had no direct cost to students, their estimated retail value was included to better represent the true hardware cost. The estimated value of the eShop components was \$48.19, and the self-service connector/header components were estimated at \$12.53, giving a combined provided-parts value of \$60.72. These parts included passives, pushbuttons, LEDs, regulators, protection devices, pin headers, BNC connectors, and the ESP32 module.

Mechanical costs were estimated separately. The 3D-printed enclosure used well under 0.5 kg of filament, giving an estimated cost of \$5–10, and the small 3 mm acrylic top sheet was estimated at about \$3. Overall, the estimated hardware cost remained around \$132.55, excluding laboratory equipment.

4.3 PCB Cost Analysis

The PCB was fabricated through JLCPCB. Since the board is less than 100 mm × 100 mm, it qualifies for JLCPCB's low-cost prototype pricing, where the base fabrication cost is about \$4 for five boards. With shipping included, the total cost is typically around \$20–\$25, depending on the shipping method. Although fabrication was covered by course resources, this estimated market cost is included to reflect the true production cost of the prototype.

4.4 Grand Total

Overall, the estimated total project value is approximately \$21,696. This total includes \$58.83 for DigiKey-purchased parts, \$60.72 for eShop and self-service lab components, \$10 for 3D printing material, \$3 for the acrylic top sheet, \$25 for PCB fabrication, and \$21,538 for engineering labor

$$\$58.83 + \$60.72 + \$10 + \$3 + \$25 + \$21,538 = \$21,695.55$$

5. Conclusion

5.1 Accomplishments

OscilloSketch was a successful project that applied many areas of electrical and computer engineering, including op-amp design, PCB layout, DSP, embedded systems, operating systems, and RTOS concepts. We also built on experience from EV Concept and Illini Electric Motorsports in hardware design, debugging, and system integration.

The final device exceeded the original goals by reaching an update rate of about 110 kHz, using the ESP32-S3 dual-core architecture, and supporting four working display modes. Additional features such as audio output and Z-blanking were also completed. Overall, the project met its main requirements, earned an honorable mention, and provided valuable mixed-signal embedded system experience.

5.2 Uncertainties

OscilloSketch met its main requirements, but one remaining limitation was the live audio streaming. The audio output worked, and packets of audio could be heard, but there were gaps and losses between packets. Because audio was an extra subsystem rather than a main requirement, it was verified functionally but not fully optimized.

Another uncertainty was the maximum reliable update rate. The system achieved about 110 kHz, which exceeded the 10 kHz requirement, but the exact limit depends on firmware overhead, SPI timing, and the complexity of each display mode. Overall, there were no major unsatisfactory results, only minor limitations that could be refined in future versions.

5.3 Ethical considerations

This project emphasizes safety, honesty in performance claims, and professional responsibility. Output performance is stated with tolerances and under specified load conditions. Worst-case error analysis is included to quantify midscale offset and gain variation rather than relying solely on nominal calculations.

The design avoids overstating capabilities and clearly defines operating limits, including output current limits for 50 Ω loads. This approach aligns with the IEEE Code of Ethics by ensuring accurate reporting of system behavior and responsible engineering communication [9].

5.4 Future work

Future work for OscilloSketch could improve both the enclosure and firmware. The current enclosure is functional, but a future design could be more ergonomic, with rounded edges, better hand grips, and a controller-style shape for more comfortable handheld use. The buttons and encoders could also be repositioned for easier control during drawing or gameplay.

The firmware is also very expandable because the hardware already supports fast X/Y output, button input, encoder input, audio output, and Z-blanking. Future modes could include more oscilloscope games, Fourier series visualization, function generator waveforms, and musical note or piano modes. The live audio feature could also be improved with better buffering and optimized timing to reduce packet loss. Overall, the design provides a strong platform for adding new interactive oscilloscope features.

References

- [1] Espressif Systems, “ESP32-S3-WROOM-1 & ESP32-S3-WROOM-1U Datasheet,” v1.8, Mar. 2026. [Online]. Available: https://documentation.espressif.com/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf
- [2] Microchip Technology Inc., “MCP4902/4912/4922 8/10/12-Bit Dual Voltage Output DAC Converter with SPI Interface,” DS22250A, Apr. 2010. [Online]. Available: <https://ww1.microchip.com/downloads/en/devicedoc/22250a.pdf>
- [3] Texas Instruments, “LM27762 Low-Noise Positive and Negative Output Integrated Charge Pump Plus LDO Datasheet,” SNVSAF7C, Rev. C, Oct. 2025. [Online]. Available: <https://www.ti.com/lit/gpn/LM27762>
- [4] Texas Instruments, “OPAx192 36-V, Precision, Rail-to-Rail Input/Output, Low Offset Voltage, Low Input Bias Current Op Amp with e-trim Datasheet,” SBOS620E, Rev. E, Nov. 2015. [Online]. Available: <https://www.ti.com/lit/ds/symlink/opa192.pdf>
- [5] Texas Instruments, “TLV906xS 10-MHz, RRIO, CMOS Operational Amplifiers for Cost-Sensitive Systems Datasheet,” SBOS839M, Rev. M, Dec. 2024. [Online]. Available: <https://www.ti.com/lit/gpn/tlv9061>
- [6] Bourns, Inc., “PEC11R Series 12 mm Incremental Encoder Datasheet,” Rev. 04/26, Apr. 2026. [Online]. Available: <https://www.bourns.com/docs/Product-Datasheets/PEC11R.pdf>
- [7] Diodes Incorporated, “AZ1117C Low Dropout Linear Regulator Datasheet,” DS36676, Rev. 5-2, Sept. 2022. [Online]. Available: <https://www.diodes.com/datasheet/download/AZ1117C.pdf>
- [8] Tektronix, Inc., “2225 Oscilloscope Operators Manual,” Part no. 070629801, Aug. 2016. [Online]. Available: <https://www.tek.com/en/manual/2225-operators-manual>
- [9] IEEE, “IEEE Code of Ethics.” [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html>
- [10] Texas Instruments, “Output Voltage Scaling Using Op Amps and Precision Resistors,” SLOA097, Mar. 2010. [Online]. Available: <https://www.ti.com/lit/an/sloa097/sloa097.pdf>
- [11] Texas Instruments, “USB Power Delivery Controller Solutions Technical Reference,” SLYY228, 2021. [Online]. Available: <https://www.ti.com/lit/eb/slyy228/slyy228.pdf>

Appendix A Requirement and Verification Table

Table 1. Power Subsystem Requirements

Requirement	Pass Criteria	Verification	Result	P/F
Digital rail stays regulated, so ESP32-S3, DAC logic, encoders, and buttons operate reliably	3.3 V rail between 3.20 V and 3.40 V	Measure 3.3 V rail while the device is running	Voltage: 3.303 V	Pass
Analog positive rail stays near +5 V for the op-amp positive output swing	+5VA between +4.70 V and +5.30 V	Measure +5VA test point at idle and while drawing	Voltage: +4.998 V	Pass
Analog negative rail stays near -5 V for the op-amp negative output swing	-5VA between -4.70 V and -5.30 V	Measure -5VA test point at idle and while drawing	Voltage: -4.877 V	Pass
Input protection reduces the risk of board or USB source damage during a power fault	Current is limited after the fault and then recovers	Use current-limited bench supply & briefly test the fault	Datasheet (PTS12066V100) Poly-Fuse 5V 1A	Pass

Table 2. MCU / Firmware Subsystem Requirements

Requirement	Pass Criteria	Verification	Result	P/F
Updates the X and Y outputs fast enough to create a stable and smooth oscilloscope image	Update rate \geq 10,000 X-Y points/s	Measure LDAC/update pulse frequency on oscilloscope	Update Rate: 110 kHz	Pass
SPI shall be fast enough to send both X and Y DAC values within each update period	SPI clock \geq required 5 MHz	Probe SCK and CS with an oscilloscope or logic analyzer	SPI clock: 20.0 MHz	Pass
Firmware shall run continuously without crashing, so the device is reliable during use	No freeze/reset for 5 minutes	Run drawing or waveform mode continuously	Never froze or reset	Pass
Firmware starts in a safe state during power-up , so the input is not suddenly driven high	Startup output near 0 V, not stuck at ± 5 V rail	Power cycle while measuring BNC outputs	Starts up at 0.01 V	Pass

Table 3. User Input Subsystem Requirements

Requirement	Pass Criteria	Verification	Result	P/F
Encoders shall move the correct axis so the controls feel intuitive , like Etch-a-Sketch	X moves horizontally, Y moves vertically	Rotate X and Y encoders in XY mode	Correct movement	Pass
Encoder motion creates a predictable voltage change, so knobs are consistent	BNC changes to be $3.9\text{ V} \pm 0.2\text{ V}$ for 20 detents	Rotate each encoder 20 detents and measure the change in the BNC output.	X Voltage: 4.06 V Y Voltage: 3.86 V	Pass
Button inputs shall respond quickly enough that they feel real-time	Button press latency $\leq 20\text{ ms}$	Probe the physical button signal and the GPIO output	Latency: 19.20 ms	Pass

Table 4. DAC Subsystem Requirements

Requirement	Pass Criteria	Verification	Result	P/F
RAW_X and RAW_Y span the DAC range, so the analog stage can create the full output swing	Both outputs are approximately 0 V to 3.3 V	Check DAC's output when code is at 0 and 4095	RAW_X: 0.002 V to 3.30 V RAW_Y: 0.001 to 3.29V	Pass
DAC outputs shall be monotonic so drawing motion is smooth and predictable	Voltage does not decrease as code increases	Sweep codes upward and observe RAW_X/RAW_Y voltage	No observed voltage decreases	Pass
LDAC shall update X and Y together so the displayed point is not distorted	Both X and Y outputs change at the same update event	Output an X and Y coordinate and probe RAW_X & RAW_Y	560 ns	Pass

Table 5. Analog Output Subsystem Requirements

Requirement	Pass Criteria	Verification	Result	P/F
Updates the X and Y outputs fast enough to create a stable and smooth oscilloscope image	Update rate \geq 10,000 X-Y points/s	Measure LDAC/update pulse frequency on oscilloscope	Update Rate: 110 kHz	Pass
X and Y outputs shall be centered near 0 V, so the drawing appears centered	Both outputs within ± 0.10 V of 0 V	Command midscale code 2048 and measure both BNC outputs	X: 0.0039 V Y: 0.0022 V	Pass
Output step size shall match the expected analog gain for smooth motion	2.44 mV/code $\pm 15\%$	Step DAC by a known number of codes and calculate volts per code	Steps: 3.86V / 1600 2.41 mV/code	Pass
Output stage shall remain stable with the intended scope load and cable	No sustained oscillation.	Connect to a 1 M Ω oscilloscope cable and watch the output	No sustained oscillation	Pass
The device provides sufficient output voltage swing to create a usable XY display on the oscilloscope.	Each output reaches at least -4.5 V and +4.5 V	Command minimum and maximum DAC codes and measure outputs	X: +4.998 to -4.880V Y: +4.988 to -4.879V	Pass

Table 6. Components Cost Table

Description (* eShop)	Manufacturer	Qty.	Cost/Unit	Cost Total
Capacitor - 22 μ F / 6.3V (0805) *	KEMET / YAGEO	6	\$0.71	\$4.26
Capacitor - 1 μ F / 6.3V (0603) *	KEMET / YAGEO	6	\$0.10	\$0.60
Capacitor - 10 μ F / 20% / 10V (0603) *	Cal-Chip Electronics	15	\$0.10	\$1.54
2.2 μ F Capacitor (0603)	Samsung Electro-Mechanics	10	\$0.05	\$0.48
Capacitor - 4.7 μ F / 6.3V (0603) *	KEMET / YAGEO	3	\$0.50	\$1.50
Capacitor - 0.1 μ F 10% / 16V (0603) *	KEMET / YAGEO	15	\$0.05	\$0.78
Capacitor - 0.01 μ F / 16V (0603) *	KEMET / YAGEO	25	\$0.05	\$1.18
LED - GREEN LTST-C171GKT (0805) *	Lite-On	15	\$0.09	\$1.30
Diode - SP0503BAHTG (SOT143-4) *	Littelfuse	3	\$1.37	\$4.11

USB-C receptacle, USB4215-03-A	GCT	3	\$0.58	\$1.73
BNC connector *	Linx Technologies	4	\$2.21	\$8.84
Conn_01x05_Pin Connector (1x05 2.54mm header) *	Molex	3	\$0.69	\$2.07
Conn_01x03_Pin Connector (1x03 2.54mm header) *	Molex	3	\$0.54	\$1.62
Resistor - 10k Ω (0603) *	YAGEO	30	\$0.02	\$0.52
Resistor - 22 Ω / 1% / (1/10W) (0603) *	YAGEO	6	\$0.10	\$0.60
Resistor - 5.1k Ω 5%(1/8W) (0805) *	Stackpole Electronics	6	\$0.10	\$0.60
309k->316K Resistor (0603)	Stackpole Electronics	10	\$0.01	\$0.06
302k->309K Resistor (0603)	YAGEO	10	\$0.01	\$0.06
100k Resistor (0603)	Stackpole Electronics	10	\$0.01	\$0.06
3.4k Resistor (0603)	YAGEO	10	\$0.01	\$0.09
9.76k Resistor (0603)	YAGEO	10	\$0.01	\$0.07
19.8k Resistor (0603)	KOA Speer	10	\$0.07	\$0.66
47 ohm Resistor (0603) *	YAGEO	6	\$0.10	\$0.60
Resistor - 330 Ω (0603) *	YAGEO	15	\$0.02	\$0.36
PEC11R-4220F-S0024 Rotary Encoder/Switch	Bourns	5	\$1.89	\$9.45
Switch - Tactile *	C&K	12	\$0.31	\$3.74
Microcontroller - ESP32-S3-WROOM *	Espressif Systems	3	\$6.10	\$18.30
Voltage Regulator - AZ1117CD-3.3TRG1 (TO252-2) *	Diodes Incorporated	3	\$0.28	\$0.84
LM27762 regulated switched-capacitor voltage inverter	Texas Instruments	3	\$2.42	\$7.26
MCP4922 dual 12-bit SPI DAC	Microchip Technology	3	\$3.47	\$10.41
OPA2192 dual precision op-amp	Texas Instruments	2	\$5.55	\$11.10
TLV9061 single op-amp	Texas Instruments	3	\$0.61	\$1.83

Resettable PTC fuse	Bel Fuse	3	\$0.29	\$0.87
TVS DIODE 6VWM 15.6VC TVS diode	Littelfuse	10	\$0.12	\$1.17
Diode - CDBA540-HF (DO214AC) *	Comchip Technology	10	\$0.48	\$4.77
Diode - 82400102 TVS 5VWM 7.7VC (SOT23-6L) *	Würth Elektronik	3	\$0.86	\$2.58

Table 7. Schedule for ECE 445

Week	Eric's Work	Josh's Work
Feb 23 to Mar 1	Finalized hardware design, prepared design review material, ordered PCB and parts	Started firmware skeleton and helped prepare design review material
Mar 2 to Mar 8	Completed design review, applied hardware feedback, and reviewed PCB/design changes	
Mar 9 to Mar 15	Prepared breadboard demo with ESP32-S3 and two DACs	Wrote code for encoder-driven X/Y output updates for the demo
Mar 16 to Mar 22	Continued hardware planning and reviewed PCB/enclosure	Continued firmware work during spring break
Mar 23 to Mar 29	Assembled PCB and tested overall functionality while ordering a new PCB	Integrated firmware with the PCB and helped test raw DAC output behavior
Mar 30 to Apr 5	Both continued PCB testing with completed base firmware functionality for progress demo next week, and completed individual progress reports separately	
Apr 6 to Apr 12	Verified core hardware requirements for X/Y output range, centering, and update behavior	Improved firmware and tested update timing and control behavior
Apr 13 to Apr 19	Assembled the final PCB with audio and Z-blanking and did R&V testing	Increased the refresh rate using dual core, along with finishing four display modes
Apr 20 to Apr 26	3-D printed the PCB enclosure, and laser cut the acrylic panel. Finalized code and prepared for final demo, and created slides for final presentation.	
Apr 27 to May 4	Presented and went to the award ceremony. Worked on the final report, did lab checkout, and turned in lab notebooks.	