University of California Santa Cruz
Baskin School of Engineering
Electrical and Computer Engineering Department

Wet/Dry Cycling Final Report

Rafael Delwart, Cole Schreiner, Matthew Randall, Ryan Taylor & Ana Villa

ECE 129
Instructor: Stephen Petersen
TA: Evripides Nicolaides
26 May 2025

Table of Contents

1 Abstract

This report describes the design and implementation of an automated wet/dry cycling system to progress Professor Deamer's prebiotic biology theory. The system executes thermal control, fluid delivery, gas flow, mixing, and extraction. A block diagram defines subsystem responsibilities and informs team organization. Each subsystem was developed to perform a specific function within the overall cycling process. A web-based user interface was created to configure system parameters and send commands. Backend communication connects the interface to the hardware. A power budget was developed to support operation across all subsystems. The system met all functional requirements and will be used in ongoing experiments related to RNA synthesis.

2 Wet/Dry Cycling System Introduction & Motivation - Team

The prevailing scientific theory of the origins of life proposes that life began in the ocean, where hydrothermal vents provided the necessary energy and chemical gradients for the first biomolecules to form. However, Professor David Deamer is challenging this ocean-centric model by proposing an alternative hypothesis: that life may have originated in freshwater hydrothermal pools on land, where wet/dry cycles played a critical role in driving prebiotic chemical reactions. Unlike the deep ocean, these terrestrial environments would have allowed for periodic dehydration and rehydration, which Deamer argues is essential for the polymerization of RNA. Experimental progress into this theory has been limited by the lack of automated systems that can precisely simulate hydrothermal environments.

Through the development of an automated system to simulate hydrothermal environments, this project aims to progress the research in the role of wet/dry cycling in the theory of the origins of life. Under the guidance of Professor David Deamer, a researcher in prebiotic

chemistry, the team engineered a modular system capable of simultaneously processing multiple samples with environmental controls. The machine features automated fluid handling, temperature regulation, gas control, and mixing capabilities, with user control over each variable. This system provides researchers with control over environmental parameters including temperature, humidity, gas composition, and cycling frequency. By automating the process of wet/dry cycling, this project aim to create a high-throughput platform that advances origins-of-life research. In order to contribute productively to the project, each team member needed to fully understand the scope of the Professor Deamer's research.

3 Background on Biololgy & Theory - Rafael Delwart

3.1 RNA World Hypothesis

DNA holds genetic information that gets passed down through generations. It also uses RNA to make the proteins needed for life to continue. Because DNA has to both carry information and help make the tools (proteins) that support life, it's hard to imagine how life could have started with DNA alone—since that would mean early DNA already had to be very complex.

It has instead been proposed that life may have instead originated as RNA molecules, the so called RNA world hypothesis. RNA, unlike the more stable and inert DNA, was shown capable of carrying both copyable genetic information and enzymatic activity. This occurs as RNA strands (unlike DNA) can readily fold into complex structures capable of enzyme-like catalytic activities.

3.2 Prebiotic RNA Synthesis

RNA strands are currently produced from DNA templates through the act of transcription using RNA polymerase enzymes for their eventual translation into proteins. The RNA World

Hypothesis envisions the earliest form of replicating genetic material consisting of short RNA molecules capable of replication without assistance from encoded proteins but instead using inherent RNA catalytic functions such as the formation of chemical bonds linking nucleotides into RNA chains. Natural selection would then select the fastest RNA replicators folding into the shape most beneficial to its own amplification. It has been shown that RNA precursors, ribonucleotides of Guanosine, Adenosine, Cytosine and Uracil (GACU) can form short chains called oligonucleotides under certain abiotic, early earth, conditions.

3.3 Experimental Simulation

The designed wet/dry cycling system derives from the observation of Dr Deamer et al that such polymerization was greatly enhanced through cycles of dehydration and rehydration. During these cycles reagents are concentrated to such an extent that precursors are chemically activated and made more reactive then in strictly aqueous environments. The improved ability to produce short RNA strands open the way to impose selective pressures on abiotically synthesized random RNA chains to recapitulate early events in life's origin. In order to realize this project, a group of engineers was formed. This group was then split into subteams through the use of a top level block diagram.

4 Top Level Overview - Team

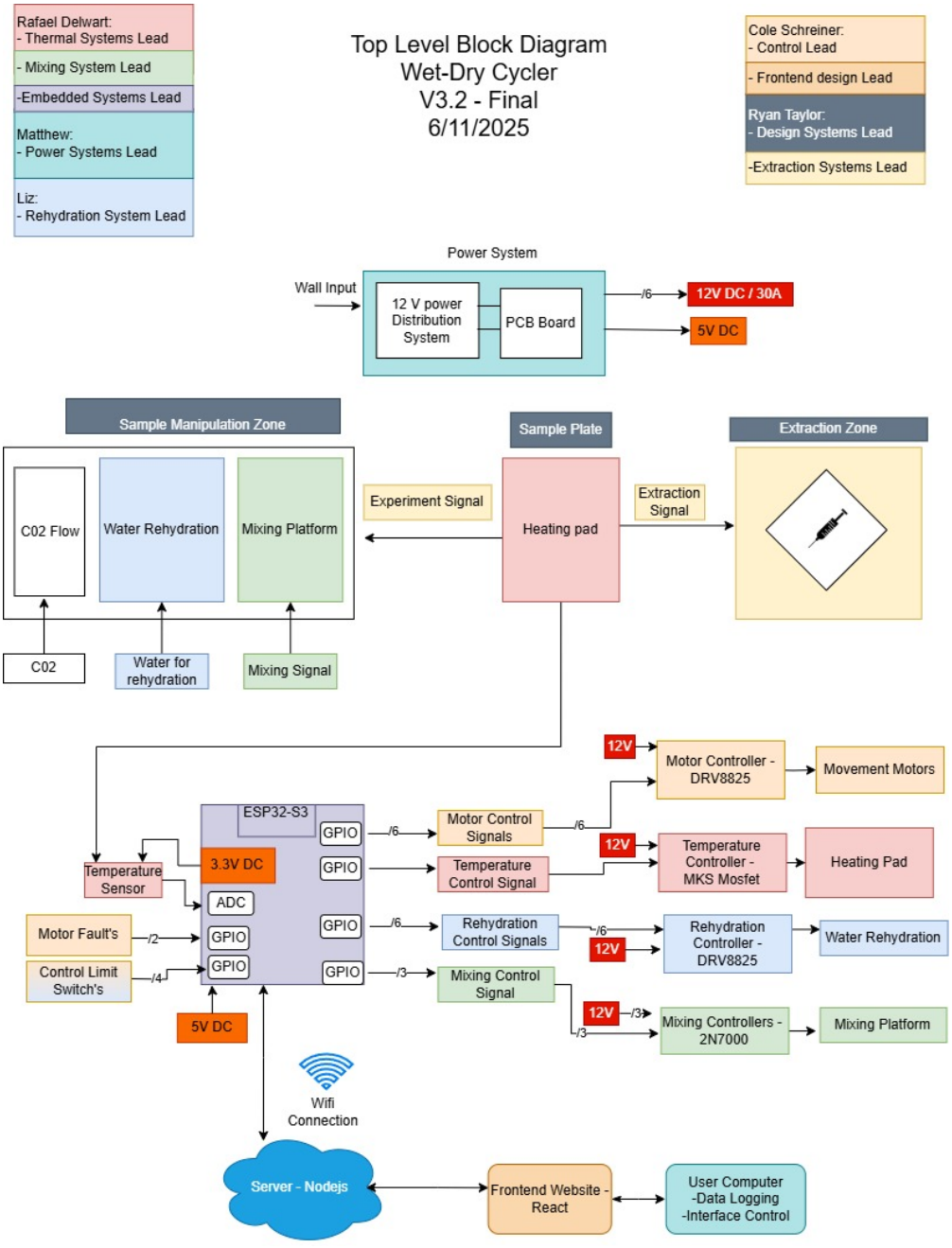


Figure 1: Overall system block diagram

The system block diagram, shown in Figure 1, provides a top-level visual representation of the system architecture. The system flow from the block diagram shows that first the samples

are heated using the silicon heating pad followed by sample rehydration and magnetic mixing. When the client is ready to extract the sample, the extraction button is pressed and the entire platform moves to the extraction zone. When extraction is finished the client then presses the button again and the samples go back into the wet/dry cycling. The block diagram breaks down the system into eight core functional subsystems; the temperature control module, fluid delivery system, CO_2/N_2 delivery unit, mixing mechanism, platform movement mechanism, power distribution, microcontroller design and user interface. Each block encapsulates a distinct operational domain, with standardized symbols and interconnecting arrows explicitly mapping the flow of control signals, materials, and data. This hierarchical decomposition of the overall system enabled the team to break down the project and assign sub-team leads.

The wet/dry Cycler project was organized into seven subteams: thermal systems, mixing systems, movement systems, embedded design, power delivery and electrical design, mechanical integration, and fluid delivery. Each subteam was responsible for a distinct subsystem that contributed to the overall functionality and reliability of the device.

The thermal systems subteam oversaw temperature regulation throughout the dry phase of the cycle. This included implementing a system capable of reaching and maintaining target temperatures with precision within 5 °C to support consistent sample drying.

The mixing systems subteam managed the design and operation of the agitation platform, which was responsible for uniformly mixing fluid into samples during the wet phase. The subteam goal was to ensure thorough and repeatable mixing to promote sample integrity prior to extraction.

The fluid delivery subteam designed the rehydration subsystem to deliver controlled volumes of fluid to the sample plate during the wet phase. Their work involved integrating fluid flow components with control hardware and coordinating timing with the mixing platform. The team prioritized consistent delivery, minimal leakage, and compatibility with the surrounding mechanical and electrical systems.

The movement systems subteam developed the actuation mechanisms that enabled sample

transport between the preparation and extraction zone. The subteam's responsibilities included establishing the motion sequence, designing the interface between the control unit and motors, and ensuring positioning within 10mm. This functionality was critical for automating transitions and enabling the continuous operation of the wet/dry cycle.

The embedded design team was tasked with making the central control layer for the wet/dry cycling system. This included enabling coordination of all mechanical, thermal, and fluid-handling subsystems. Its purpose was to implement a modular firmware architecture that could execute experimental protocols autonomously while maintaining real-time communication with the user interface.

The power and electrical subteam was responsible for the electrical infrastructure of the system, including power distribution, voltage regulation, and circuit design. The subteam was tasked with creating a power system to power every component reliably. The team also was tasked with creating an engineering schematic of the entire electrical system and designing a PCB to deliver all microcontroller signals reliably.

The mechanical integration subteam focused on the physical layout and structural assembly of the device. The subteam ensured that all subsystems—thermal, electrical, mixing, movement, and fluid—were securely mounted and spatially coordinated within the system enclosure. In addition to managing the overall structural framework, this subteam was responsible for designing and implementing the CO_2 delivery system, which introduced controlled amounts of gas into the chamber as part of the dry phase.

The successful integration of these diverse subsystems required adherence to established engineering standards. By grounding design decisions in recognized standards for electronics, mechanical systems, and fluid control, the team maintained consistency across subsystems developed in parallel. The following section outlines the engineering standards that informed the design and implementation of each subsystem.

5 Engineering Standards Employed - Team

The team adhered to a variety of engineering standards throughout the development of the automated hydrothermal simulation system. For electrical design and PCB layout, IPC-2221 standards were followed to guide trace spacing, via sizing, and overall board layout [6]. All software development adhered to standard modular design principles and was documented according to IEEE Std 829-2008 for software and test documentation [5]. For mechanical design, tolerancing and component clearances were guided by ANSI Y14.5 standards to ensure proper fit and manufacturability [4].

From a systems engineering perspective, the team used a block diagram and modular architecture to decompose the project into functional subteams, each adhering to specific design standards. These included the documented principles outlined in the *Technical Documentation Standards in Engineering Design* [9]. The individual standards followed by each subteam are discussed in their respective sections.

6 Thermal Subsystem - Rafael Delwart

6.1 Overview

Figure 2: Silicone heating pad rated for 12 V and 100 W, the pad includes a pre-installed NTC thermistor and adhesive backing

To facilitate sample dehydration shown in the block diagram Figure 1, a silicone heating pad shown in Figure 2 which is typically used for 3D printer heated beds was selected. While alternative methods such as Peltier elements and other resistive heating techniques were considered, the silicone heating pad was ultimately chosen due to its affordability, reliability, and the client's tolerance for a larger margin of temperature variation. To ensure even heat distribution across the vials and improve thermal stability, the heating pad was fixed to an aluminum plate shown in Figure 6. The pad is controlled via an off-the-shelf high current MOSFET shown in Figure 4, the heating controller is one commonly used in 3D printing applications to regulate silicon hotbeds.

6.2 Temperature Measurement

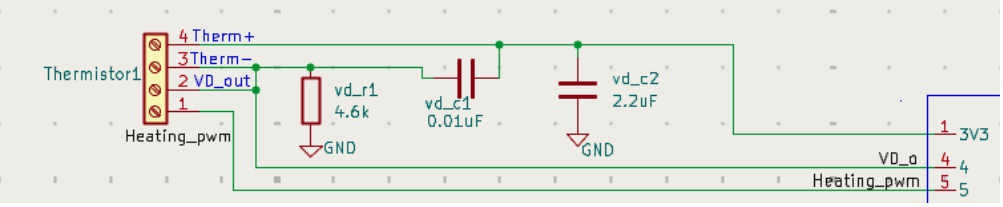


Figure 3: Voltage divider circuit, uses thermistor and known resistor of 4.6KOhm

The heating pad incorporates an NTC (Negative Temperature Coefficient) thermistor, meaning its resistance decreases as temperature increases. The thermistor is integrated into a voltage divider circuit, as shown in Figure 3. This thermistor has a beta coeffecient of $\beta = 3950\,\mathrm{K}$, and its resistance as a function of temperature is described by equation 1.

$$R(T) = R_0 \cdot \exp\left[\beta\left(\frac{1}{T} - \frac{1}{T_0}\right)\right] \tag{1}$$

Where $R(T)$ is the resistance at temperature T (in Kelvin), $R_0 = 100\,\mathrm{K\Omega}$ is the resistance at the reference temperature $T_0 = 298.15\,\mathrm{K}$ (which corresponds to $25\,^\circ\mathrm{C}$). Rearranging Equation 1 solves for temperature as a function of resistance:

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{\beta}\ln\left(\frac{R(T)}{R_0}\right) \tag{2}$$

The resulting temperature in Kelvin is then converted to degrees Celsius to meet the client's specified unit preference.

6.3 Power Switching Module: MKS MOSFET

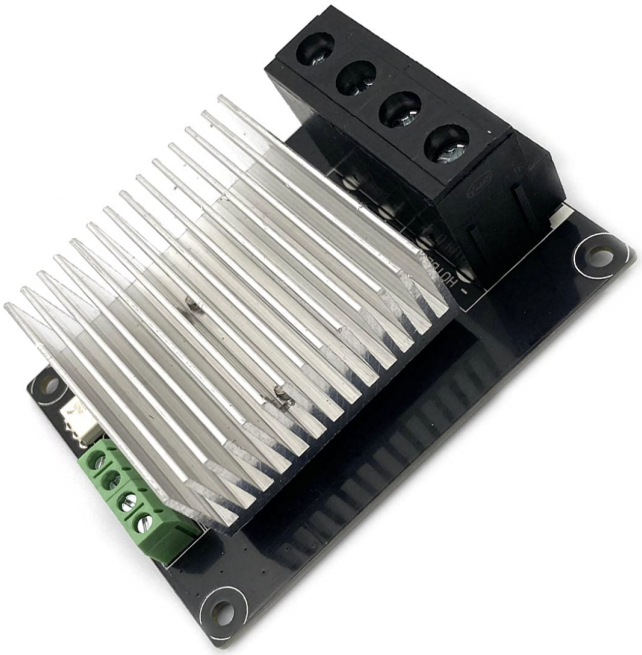


Figure 4: MKS MOSFET heated bed power module used to switch the silicone heating pad.

To control the high-current demands of the silicone heating pad, an MKS MOSFET heated bed power module shown in Figure 4 was integrated into the system. Originally designed for 3D printer hotbeds, this module is well-suited for driving large resistive loads such as the heating pad used in this project. The module accepts logic-level control signals from a microcontroller (such as the ESP32) and uses an onboard MOSFET to switch the high current needed by the pad without placing stress on the microcontroller's GPIO.

As seen in Figure 4 the MKS module features screw terminals for both power input and load output, simplifying integration with the 12V power supply and the heating element. It supports up to 25 A continuous current at voltages up to 24 V, well above the operational requirement of this system, providing both thermal and electrical headroom. An onboard indicator LED provides visual feedback when the load is active, which is useful during testing and debugging.

Internally, the module includes an opto-isolator between the control signal and the power switching circuit. This opto-isolation protects the low-voltage control electronics from transients and noise that may be generated during high-current switching events. The MOSFET itself is mounted on a large PCB copper plane and paired with a heatsink to assist with thermal dissipation during extended use.

This module was selected due to its reliability, widespread community validation in 3D printer applications, and its compatibility with both 12V systems and microcontroller-based control logic. Its inclusion significantly simplifies power routing and thermal management in the heating subsystem, allowing for safe and efficient switching without requiring custom circuitry.

6.4 Feedback Control for Temperature

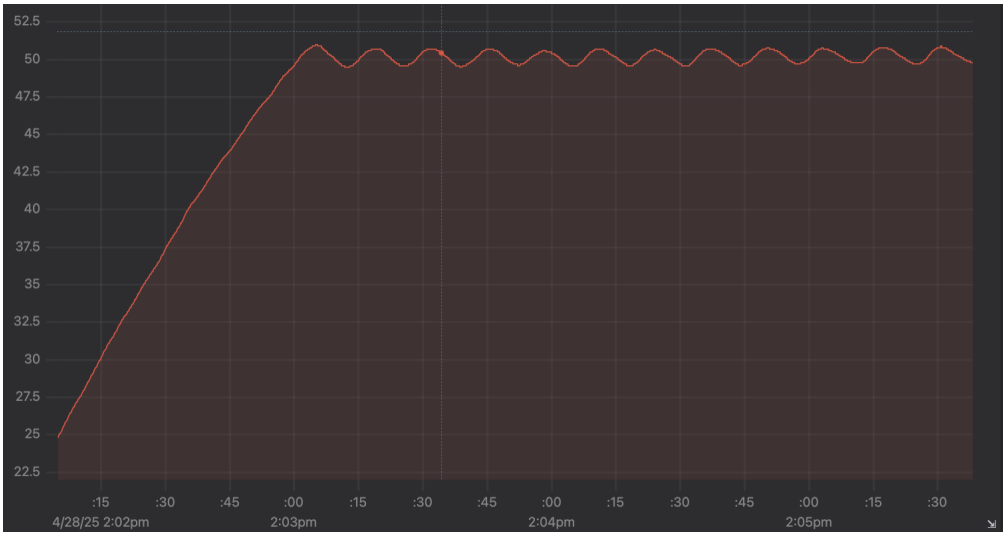


Figure 5: Temperature profile over five minutes, showing the heating pad warming from ambient to 50 °C.

The voltage drop across the thermistor in the voltage divider is read using the ESP32's ADC. From this voltage, the thermistor's resistance is computed, and the corresponding temperature in degrees Celsius is calculated. To regulate temperature, bang-bang control is employed; the high current MOSFET is toggled on or off to maintain the desired setpoint

within a tolerance of $\pm 2\,^{\circ}\mathrm{C}$, as illustrated in Figure 5.

6.5 Aluminum Plate to Hold Samples

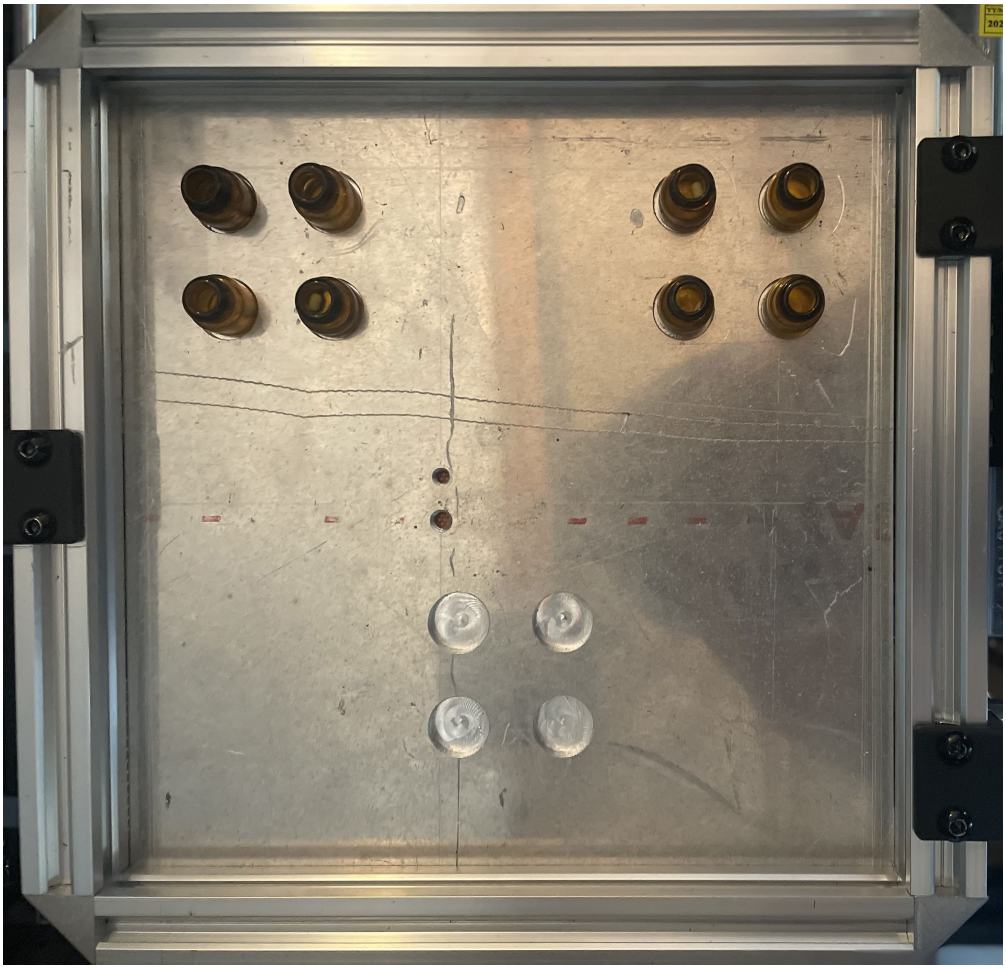


Figure 6: Machined 2024 aluminum alloy plate with slots for 12 vials.

Aluminum was selected for the sample holder due to its excellent thermal conductivity and uniform heat distribution properties. A 2024 aluminum alloy was chosen, offering sufficient thermal performance for the application. Its thermal conductivity of ($121\,\mathrm{W\,m^{-1}\,K^{-1}}$ to $130\,\mathrm{W\,m^{-1}\,K^{-1}}$) meets the required precision of $\pm 5\,^{\circ}\mathrm{C}$ for temperature control. To ensure efficient heat transfer, the silicone heating pad is fixed to the bottom of the aluminum plate using the pre-applied adhesive backing provided with the pad. The direct bonding minimizes thermal resistance between the heating pad and the plate.

The machined aluminum plate, shown in Figure 6, was designed to mount securely onto the device's movement mechanism shown in Figure 17. The aluminum plate uses engraved slots to hold 12 sample vials. Fabrication was completed using a ShopSabre CNC router at SlugWorks, located within the Baskin School of Engineering.

In addition to thermal control, gas flow plays a critical role in simulating the natural wet/dry cycles believed to occur in terrestrial hydrothermal environments. Controlled delivery of CO_2 gases over the samples supports dehydration and replicates atmospheric conditions relevant to prebiotic chemistry.

7 Gas Control Subsystem- Ryan Taylor

As shown in the system block diagram Figure 1, the gas delivery system is integrated into the Sample Manipulation Zone and plays a critical role in supporting the chemical environment required for the simulations of prebiotic reactions. The main objective of the gas control subsystem was to facilitate a controlled delivery of CO_2 over the samples.

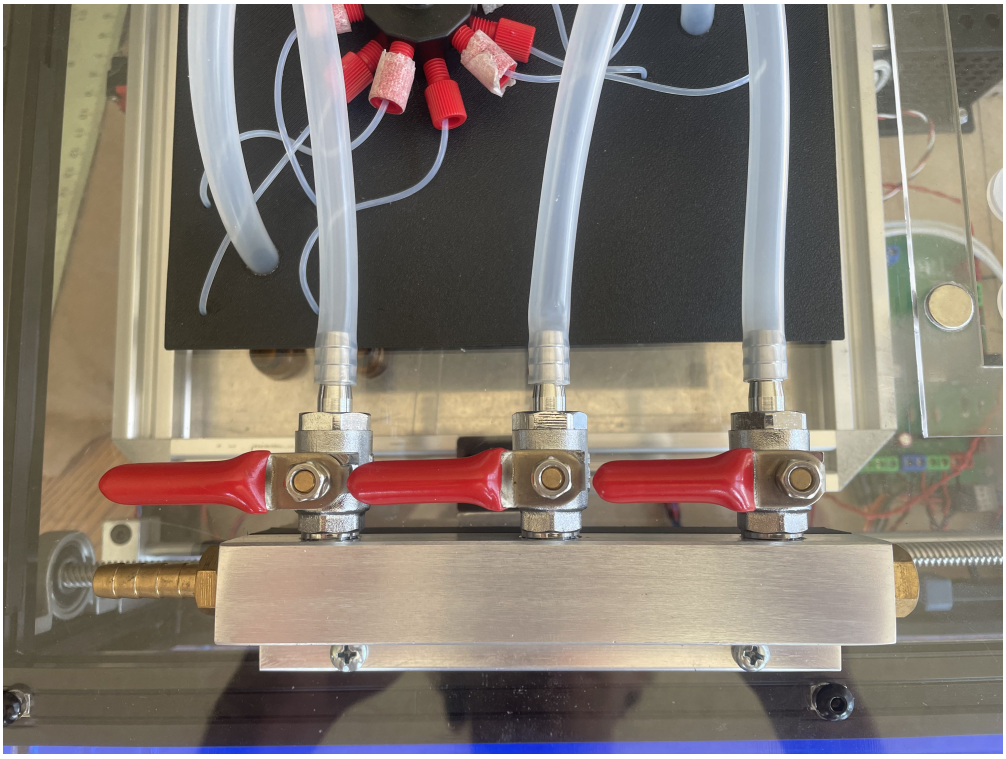


Figure 7: CO_2 manifold mounted on top of the device, controlling three output zones.

Figure 8: Commercial CO_2 splitter and manual valve system used in the manifold.

Figure 9: 0.35” OD / 0.28” ID silicone tubing used for gas routing to vial groups.

The gas delivery system consists of a main CO_2 supply line routed into a three-way manual valve manifold Figure 7. This manifold divides the incoming gas into three independently controllable output lines. Each line feeds a separate group of four vials that sit on the aluminum plate. This allows researchers to selectively activate or deactivate gas-exposure zones according to experimental requirements. This modular control supports more refined parameter sweeps and time-course studies within a single run.

Each output line is routed through guided tubing channels mounted to the top of the frame, ensuring accurate delivery to designated vial regions. Flexible silicone tubing with a 0.28-inch inner diameter and a 0.35-inch outer diameter, shown in Figure 9, was chosen to support sufficient volumetric gas flow while maintaining a tight seal on barbed connectors. The flexibility of the tubing also allows for efficient routing without introducing flow restriction or leakage.

This subsystem ensures that gas delivery is stable, adjustable, and compatible with the environmental control required for the experimental objectives of the device. Following the gas-driven dehydration phase, the system transitions into rehydration, where the system deliverers fluid to each sample.

8 Rehydration Subsystem - Ana Villa

The custom-built rehydration system evaluated in this experiment was developed using a design methodology inspired by commercial syringe pumps commonly used in biomedical laboratories. This approach ensured compatibility with standard lab practices while enabling reproducibility and scalability. The system, shown schematically in Figure 1, delivers water from a single syringe pump through a 13-port manifold to twelve individual outlets, each aimed at rehydrating a specific sample chamber. The design target was to deliver approximately 100 microliters of water per outlet per cycle.

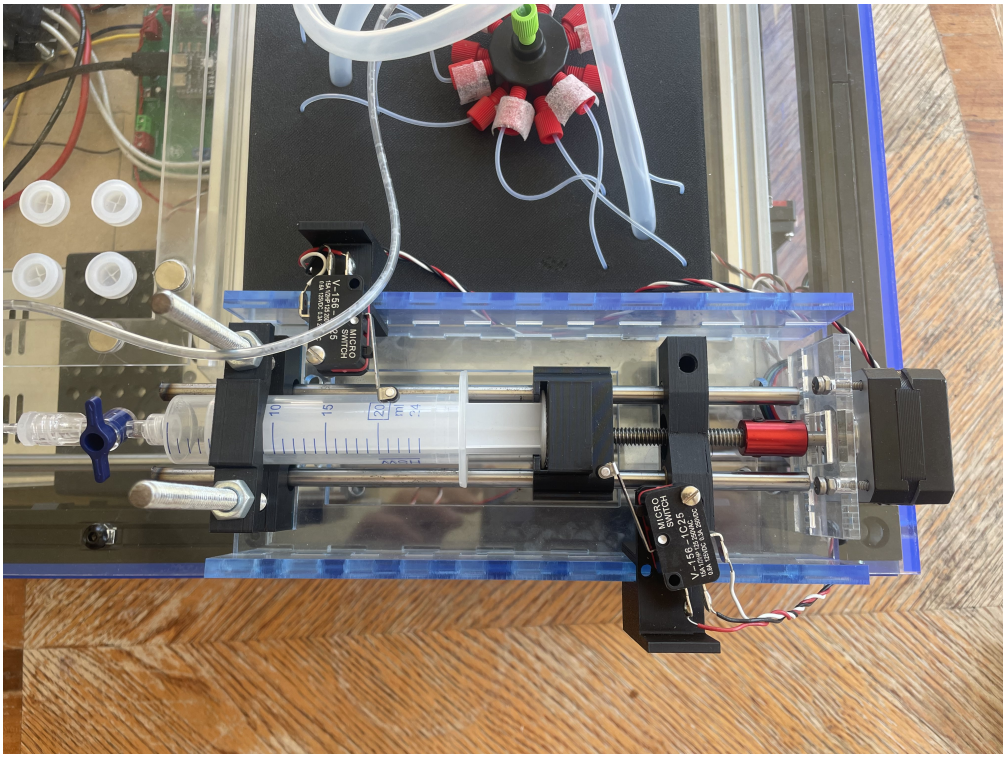


Figure 10: Syringe Pump

The syringe pump used in the system was built using a 25 mL syringe, mounted in a 3D-printed carriage and driven by an M8 threaded rod with a 1.25 mm pitch, connected to a NEMA 17 stepper motor.

To drive the syringe plunger, the torque T required by the M8 lead screw (1.25 mm pitch,

mean diameter $d_m = 7.5\,\text{mm}$) under a 30 N axial load was estimated using:

$$T = \frac{F \cdot d_m}{2} \left(\frac{l + \pi \mu d_m}{\pi d_m - \mu l} \right) \tag{3}$$

Assuming a friction coefficient $\mu = 0.2$, the required torque is $T \approx 0.112\,\text{N} \cdot \text{m}$.

Using a NEMA 17 stepper motor with holding torque $T_h = 0.45\,\text{N} \cdot \text{m}$ and rated current $I_r = 1.5\,\text{A}$, the estimated current to produce the needed torque is:

$$I \approx I_r \cdot \frac{T}{T_h} = 0.37\,\text{A} \tag{4}$$

As the motor rotates, the threaded rod translates this rotary motion into linear displacement through a lead screw mechanism. Each full rotation of the M8 rod advances the carriage holding the syringe plunger by 1.25 mm, pushing water out of the syringe barrel with controlled precision. By incrementally rotating the stepper motor in discrete steps, the system can finely regulate the volume of fluid dispensed, enabling reproducible delivery of small volumes on the order of microliters. Control is provided by an ESP32 microcontroller and a DRV8825 stepper driver as discussed in section 11, with limit switches placed at both the front and back of the carriage which signal when the syringe is empty or when the plunger has returned to its starting position.

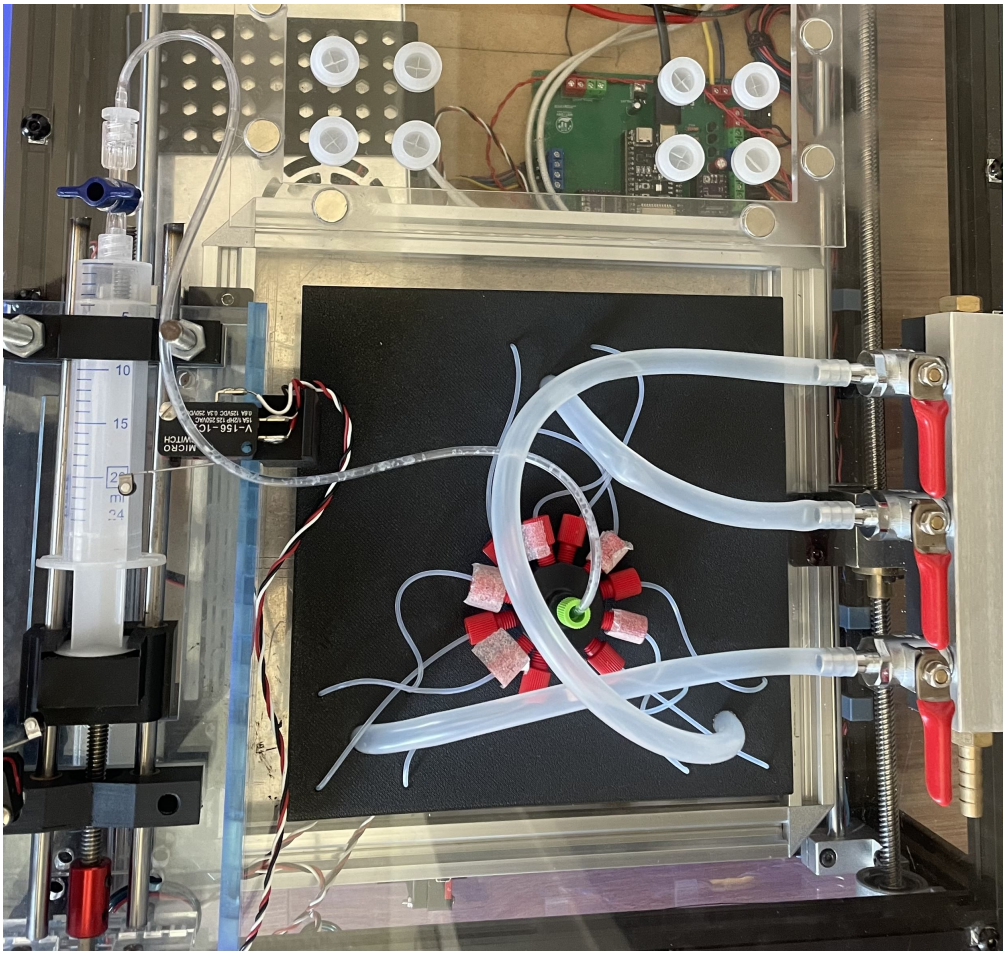


Figure 11: Syringe Pump with Manifold connection

The syringe pump feeds into a 13-port manifold consisting of one intake and twelve output ports. Each output port is connected to 1/16-inch outer diameter (OD) PTFE tubing with a manual shut-off valve to control flow to individual samples. The intake tubing was initially selected as 1/8-inch OD to support higher flow rates and reduce resistance during fluid draw. This larger diameter helps maintain sufficient pressure and minimizes strain on the pump, especially during the intake phase when fluid must be pulled quickly and evenly into the system. In contrast, the smaller 1/16-inch OD tubing on the output side was chosen to enable controlled delivery of fluid to each sample, as smaller internal volume reduces the chance of overshooting the target volume due to inertia or residual pressure. PTFE was selected for all tubing due to its widespread use in laboratory systems, offering excellent

chemical and thermal resistance. To mitigate pressure loss and water dripping caused by vacuum effects when changing syringes, a manual shut-off valve was installed between the syringe and the intake line. This allowed the intake to be temporarily sealed during syringe replacement, preserving system integrity.

8.1 Test and Results

Table2

5/12	1	2	3	4	5	# Average	Standard Dev	# Error %
T1	95	105	100	95	101	99.2	4.27	3.2
T2	100	100	105	90	97	98.4	5.50	3.6
T3	95	113	110	118	107	108.6	8.62	10.6
T4	105	106	85	100	108	100.8	9.31	6.8
T5	105	85	90	87	81	89.6	9.21	12.4
T6	110	105	111	115	113	110.8	3.77	10.8
T7	105	100	100	112	85	100.4	9.91	6.4
T8	110	95	112	100	108	105.0	7.21	7.0
T9	110	110	92	100	107	103.8	7.76	7.0
T10	110	100	95	100	112	103.4	7.27	5.4
T11	90	85	90	87	85	87.4	2.51	12.6
T12	118	110	117	103	110	111.6	6.11	11.6
								8.1

Figure 12: Water Port Test 1

Before testing, the manifold and tubes were pre-filled with water to eliminate air bubbles. To measure the volume delivered, a 20–200 µL manual pipette was used to measure 100uL. After each distribution cycle ($\tilde{1}$.2 mL total from the syringe), the volume of each of the 12 outputs was drawn and recorded individually. This was repeated across multiple trials. Across several trials, the average percent error was approximately 8%. The highest deviation from the target volume was 13%, while the lowest was 4%. Although volume measurements were performed manually using a pipette and are subject to slight user variability, they still offered a reasonable estimate of the system's performance. More accurate assessments would have required equipment such as analytical balances capable of detecting mass changes in the milligram range or high-resolution flow sensors. However, due to resource constraints and

the absence of such precision tools, low-cost measurement techniques were relied on. Despite this, the results provide meaningful insight into the system's accuracy under realistic testing conditions.

During testing, one of the primary issues observed was uneven water flow between ports. Some outputs delivered noticeably more water than others. To troubleshoot this, the subteam consulted members of the DuBois Lab, particularly Sara O'Rourke, who provided both materials and insight. She pointed out that a narrower intake tube diameter compared to the output tubes could restrict flow and cause pressure inconsistencies. At the time, the intake used a 1/16-inch outer diameter (OD) tube same as the output tubes. She recommended using a wider intake tube to increase pressure and promote more uniform distribution through the manifold. The system was then changed to a 1/8-inch OD tube while the output tubes remained at 1/16-OD.

Further guidance came from PhD students in the Schmidt Lab, who specialize in microfluidics. They emphasized that water will always follow the path of least resistance. This principle helped explain why certain outputs consistently delivered more fluid: those lines likely had shorter, straighter tubing with fewer bends, making them lower-resistance paths. To address this, they suggested increasing the number of turns or curves in the tubes of overproducing ports to artificially raise resistance, while underperforming ports should have straighter, more level paths to minimize gravity-driven or frictional losses. Another recommendation was to cut all output tubes to the same length, which would help standardize flow resistance and reduce variability.

Lastly, the subteam found that achieving tighter connections at all tube junctions was critical to maintaining consistent flow. Even minor leaks or loose fittings introduced asymmetry into the pressure distribution and affected the accuracy of the system. These collective insights from experienced lab members were instrumental in refining the design and improving the repeatability of fluid delivery.

8.2 Discussion

The results demonstrate that the rehydration system performs reliably within the of 100uL per sample for Professor Deamers application. Pre-filling the tubing and including valves at the outputs were crucial to achieving this level of consistency, as they eliminated major sources of variability such as trapped air and inconsistent pressure. Using a pipette for volume measurement was a pragmatic solution in the absence of a precision balance and enabled effective identification of variation between outputs.

The rehydration system, which uses a syringe pump and a 13-port manifold, demonstrated reasonably consistent water delivery of approximately 100 µL per port, with an average error of 8% and a range of 4% to 13%. While this level of accuracy is acceptable for many biological protocols, more sensitive procedures may require tighter control. The use of pre-filled tubing and manual shut-off valves helped reduce variability, and despite the lack of precision instrumentation, performance measurements using a pipette provided a practical and cost-effective solution. Future improvements could include automating the measurement process or incorporating flow sensors for real-time monitoring. To enhance mechanical performance and fluid control, adopting a more specialized pump—such as the SSP Series Syringe Pump with a 30 mm stroke—is recommended. Its actuation and stroke control make it well-suited for small-volume distribution tasks like those required in this system. Overall, the setup proved effective and adaptable, offering a solution for fluid delivery in resource-constrained laboratory environments.

Following the rehydration, the next phase of the protocol involved mixing the samples to ensure the water was evenly distributed.

9 Mixing Subsystem - Rafael Delwart

9.1 Overview

The mixing state shown in the block diagram Figure 1 is driven by the goal of ensuring that the water introduced during the rehydration phase is evenly distributed throughout each sample. To actuate this process, an appropriate mixing method was first selected, followed by the development of a control design to manage its operation.

9.2 Mixing Method

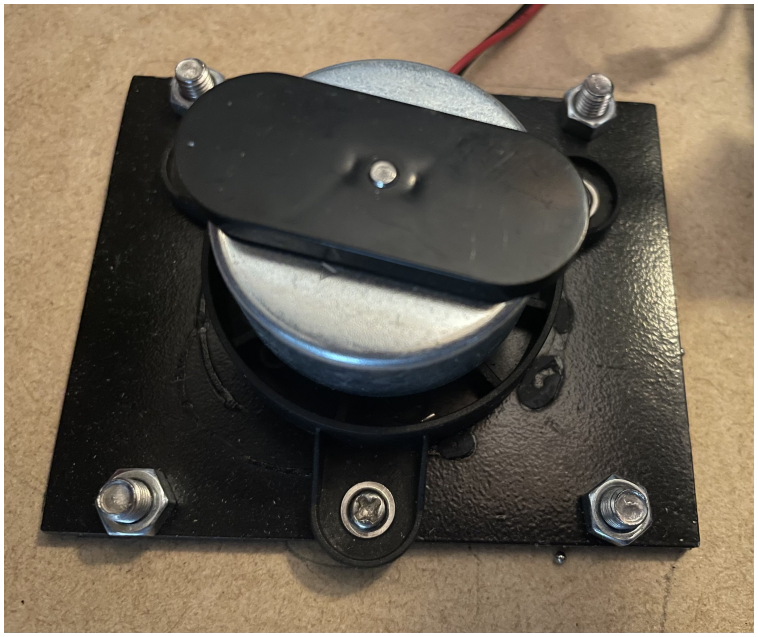


Figure 13: 12V DC motor with large magnet attached for magnetic agitation.

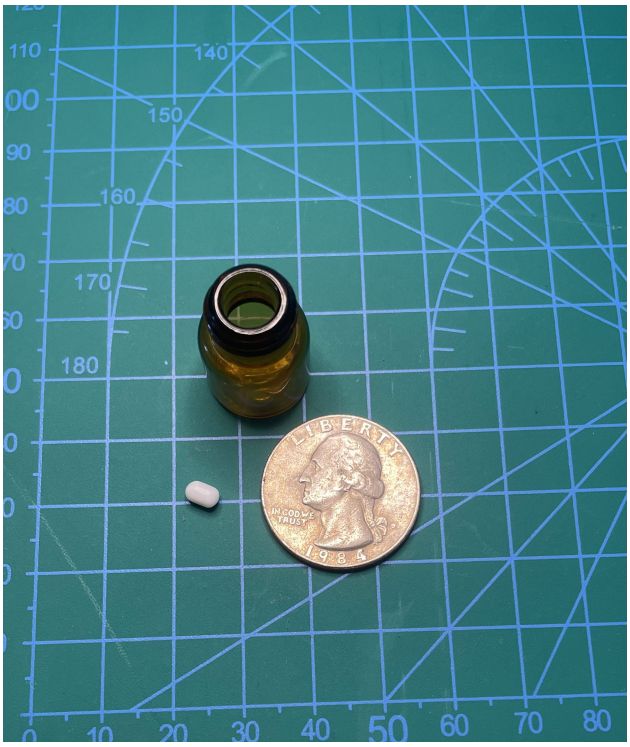


Figure 14: Stir Bar and Vial, shown next to a U.S. quarter for scale.

The mixing system designed uses three individually controlled DC motors shown in Figure 13, each repurposed from magnetic stirring plates. These motors use an attached permanent magnet and are positioned beneath groups of four sample vials shown in Figure 6. Inside each vial, a small magnetic stir bar shown in Figure 14 is placed. As the motor spins, it creates a rotating magnetic field that stirs the bar within the sample solution. This setup ensures gentle and consistent agitation without physical contact. The ability to individually control each section of vials and the duration of which they are mixed provides researchers with greater control over experimental parameters, allowing them to refine the conditions most conducive to RNA production.

9.3 Mixing Mosfet Control

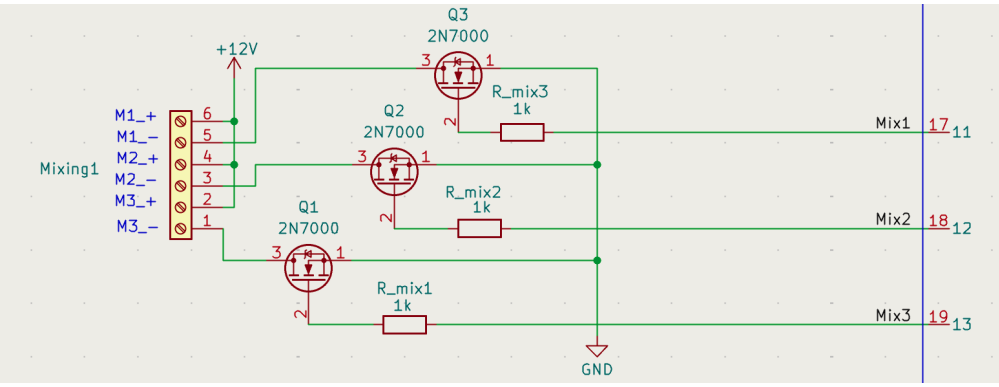


Figure 15: Mixing circuit: three 2N7000 N-channel MOSFETs, each controlled by an ESP32 GPIO pin, regulate power to individual DC mixing motors

The DC motors are powered by a 12 V supply and switched using 2N7000 N-channel MOSFETs, as shown in Figure 15. Each motor is controlled independently by a corresponding GPIO pin on the ESP32 microcontroller, enabling selective mixing across the three vial zones.

To ensure noise-free switching, 1 kΩ gate resistors are placed between the GPIO pins and the MOSFET gates. These resistors limit inrush current during switching. Without these resistors, the microcontroller could be exposed to brief but potentially damaging current spikes. The resistors also reduce the risk of signal ringing and unwanted oscillations. Each 2N7000 MOSFET is rated for up to 200 mA of continuous drain current. Measurements indicate that the mixing motors draw a maximum of approximately 150 mA during stall conditions, ensuring the components operate well within their safe limits.

9.4 Magnetic Field Interference

Figure 16: Magnetic field interference test using magnetic sand. The distinct, non-overlapping patterns confirm that each mixing motor generates an isolated magnetic field, preventing interference between adjacent vials.

A mixing field interference test was conducted using magnetic sand to verify that the rotating magnetic field of one motor does not interfere with adjacent vials. The results confirmed effective isolation between the three mixing zones as shown in Figure 16.

Once the samples have been thoroughly rehydrated and mixed to ensure uniform distribution of solutes, the cycle repeats. However, the researcher can hit the extraction button to move to the extraction stage. The extraction mechanism is designed to have the researcher remove the sample in the least intrusive way possible.

10 Sample Extraction - Cole Schreiner & Ryan Taylor

Extraction of samples during the wet/dry cycling process was a feature that enhanced experimental flexibility and control. By enabling researchers to remove individual vials for analysis at various points within a cycle, the system allowed for monitoring and manipulation of experimental parameters on a per-sample basis. This functionality provided researchers with the ability to conduct studies, compare the effects of different cycle lengths, and introduce interventions at specific stages, all within a single experiment. This level of control was not achievable in traditional batch processes, where all samples were treated identically and processed together, limiting experimental variability and insight.

10.1 Movement System - Cole Schreiner

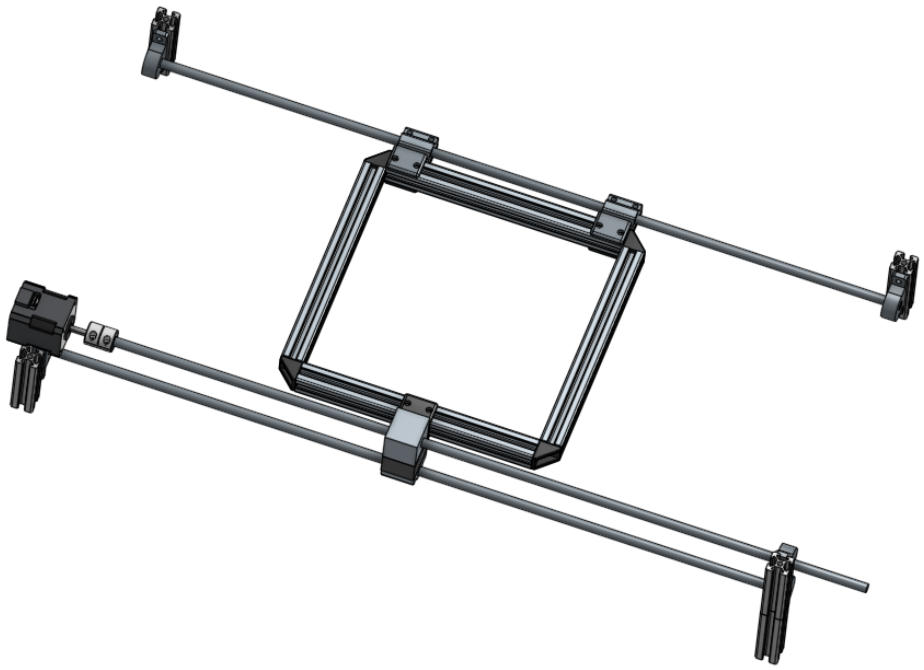


Figure 17: Extraction System Rendering

To achieve this functionality, the design methodology incorporated two distinct physical locations within the system: one for active wet/dry cycling and the other for extraction and analysis. The extraction system, shown in Figure 17, was engineered to move seamlessly between these locations, allowing a vial to be isolated for sampling or manipulation without disturbing the ongoing cycles of the remaining samples. This modular movement preserved the integrity of the experiment while ensuring that environmental conditions could be tightly controlled and replicated for each sample.

Design Evolution and Challenges

The development of the extraction mechanism underwent significant design iterations based on practical testing and reliability concerns. The extraction system needed to be structurally sound, provide stability for other subsystems, and remain heat resistant for extended periods of experimentation. The initial prototype, shown in Figure 18, featured a complex wheel-mounting platform system that utilized multiple mechanical components, including a lead screw drive coupled with a belt transmission system.

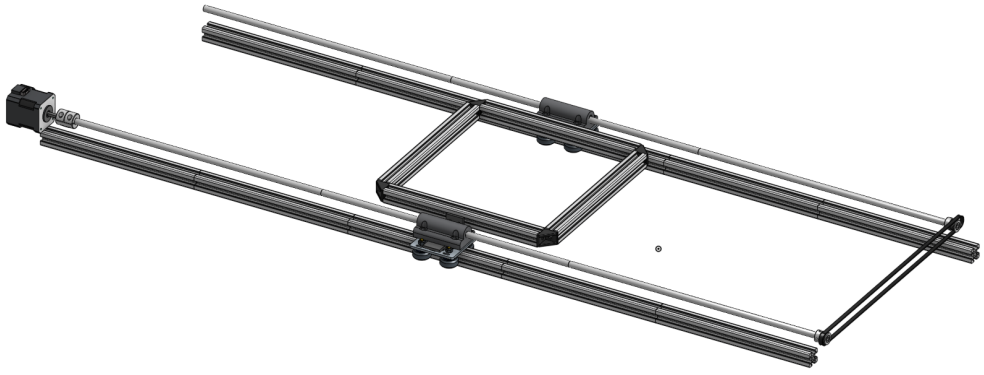


Figure 18: Original prototype extraction mechanism with wheel-mounting platform

Figure 19: Platform wheel mount used in the original prototype.

During testing, several critical failure modes became apparent with this initial design. The wheel mounting platform mechanism, shown in Figure 19, suffered from recurring issues with screws coming loose during operation, compromising the structural integrity of the system. The belt-driven component of the lead screw assembly introduced slipping problems that resulted in inconsistent positioning and reduced accuracy. Furthermore, the manufacturing tolerances of the various mechanical parts proved insufficient for operation, leading to misalignment of the system altogether.

Based on these observations, the design was simplified to eliminate failure-prone components. The final implementation utilized a direct lead screw drive system coupled with a ball bearing linear slide mechanism, shown in Figure 20 and Figure 21 below.

Figure 20: Lead Screw mounting system

Figure 21: Ball bearing pole mount

This configuration translated rotational motion from the stepper motor directly into linear motion through the lead screw. Furthermore, the ball bearings rode on a smooth guide rod, providing lateral stability and support for the extraction system. This simplified approach significantly reduced friction, improved reliability, and allowed for calculation of the power requirements for the motors.

Torque Specifications and Calculations

The extraction system used a horizontally oriented lead screw to translate rotary motion into linear motion. The relevant design and material properties are summarized below:

Parameter	**Value**
Lead screw length	600 mm
Lead screw diameter	8 mm
Pitch	6 threads/cm (i.e., 1.67 mm per revolution)
Material (screw)	Stainless Steel
Material (nut)	Brass
Coefficient of friction (steel on brass)	0.19
Efficiency (η)	0.30 (estimated)
Load (weight of platform, brackets, plates)	$\approx$ 10 lbs ($\approx$ 44.48 N)
Rolling friction factor	0.5

Table 1: Mechanical and material parameters for lead screw system.

The total downward force from the platform, brackets, and heat plate was calculated as:

$$F = (8.56 \text{ lbs}) + (0.132 \text{ lbs}) + (2.0 \text{ lbs}) = 10.692 \text{ lbs} \approx 44.48 \text{ N} \tag{5}$$

Assuming horizontal motion and taking into account rolling friction and lead screw efficiency, the torque required to move this load was given by:

$$\tau = \frac{F \cdot L \cdot p}{2\pi \cdot \eta} \tag{6}$$

where: $F = 44.48$ N (force), $L = 1.67$ mm (lead/pitch per revolution), $\eta = 0.30$ (lead screw efficiency), and p (rolling friction multiplier) $= 0.5$ (applied conservatively).

$$\tau = \frac{44.48 \text{ N} \cdot 1.67 \cdot 10^{-3} \text{ m} \cdot 0.5}{2\pi \cdot 0.3} \approx 0.0787 \text{ Nm} \tag{7}$$

This torque estimate was used to select a stepper motor rated for at least 0.1 Nm to ensure sufficient force under all operating conditions. Given this small torque requirement, a stepper motor capable of delivering a torque above the desired requirement while maintaining relatively low power consumption was chosen.

Stepper Motor Selection and Performance

To drive the lead screw, the Nema-17 stepper motor was selected, a widely used option in motion control applications such as 3D printers and CNC systems. Key specifications of this motor included:

Specification	**Value**
Available Torque	0.59 Nm
Step Angle	1.8°
Rated Current (per phase)	2.0 A
Phase Resistance	1.4 Ω
Inductance	3.0 mH
Recommended Voltage	12–24 V

Table 2: Electrical specifications for the selected NEMA 17 stepper motor.

The required torque to drive the lead screw under full load was calculated to be approximately:

$$\tau_{\text{required}} = 0.0787 \text{ Nm}$$

This value was well within the holding torque capability of the selected motor:

$$\frac{\tau_{\text{required}}}{\tau_{\text{motor}}} = \frac{0.0787}{0.59} \approx 0.133 \Rightarrow \text{only } 13.3\% \text{ of available torque used}$$

This provided a large safety margin, ensuring the motor could operate without overheating or stalling, even if the actual friction or weight increased slightly due to wear or misalignment.

Current Draw Estimate

While the motor was rated for 2.0 A per phase, the actual current draw under load was proportional to torque. Assuming a linear relationship between torque and current draw (a reasonable approximation for stepper motors operating below rated torque):

$$I_{\text{draw}} \approx \frac{\tau_{\text{required}}}{\tau_{\text{rated}}} \cdot I_{\text{rated}} = \frac{0.0787}{0.59} \cdot 2.0 \text{ A} \approx 0.267 \text{ A (per phase)} \tag{8}$$

Therefore, under normal operation, the motor drew approximately 270 mA per phase, a small fraction of its rated capacity. This low current requirement improved energy efficiency

and reduced heat generation.

In summary, the selected NEMA 17 stepper motor offered a torque capability more than 7 times the system's needs, safe thermal performance with modest current draw, compatibility with compact 3D-printer-style drivers, and a good balance between force, efficiency, and size for the extraction system.

10.2 Extraction Mechanism - Ryan Taylor

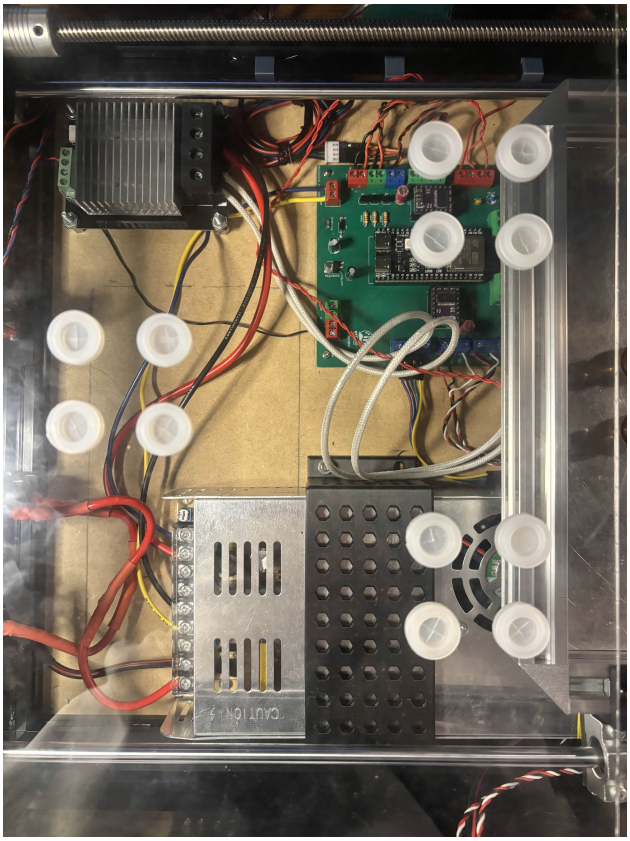


Figure 22: Physical extraction location on the housing where vials are accessed.

Figure 23: Silicone straw hole plugs used to maintain pressure seal while allowing syringe access.

Once the vial platform reaches the designated extraction location Figure 22, individual vials must be accessed for sampling without compromising the sealed environment within the chamber. To address this, a sealing mechanism that allows insertion of a syringe or pipette while maintaining pressure was chosen.

Commercial grade **silicone straw hole plugs**, shown in Figure 23, were selected as the sealing interface. These plugs are soft and elastic, capable of resealing after puncture by a needle or pipette tip. Their design allows the user to insert an extraction needle directly through the plug to withdraw sample fluid without needing to open the chamber or remove any housing panels.

Each silicone plug is fit into a precision-cut hole in the top acrylic panel. The elasticity of the material ensures a tight seal around the hole, and the natural self-healing property of silicone helps retain CO_2 pressure inside the chamber even after repeated use. This solution proved to be simple to implement and was compatible with standard lab pipettes and syringes.

This approach allows researchers to sample fluid from individual vials during or after experimental runs without exposing the whole system to outside air or disturbing other samples.

To enable control of the stepper motors responsible for both rehydration and extraction, the DRV8825 motor driver was selected for its compatibility and performance. The controller was configured to meet the system's voltage, current, and step resolution requirements.

11 Stepper Motor Controller Configuration and Setup - Rafael Delwart

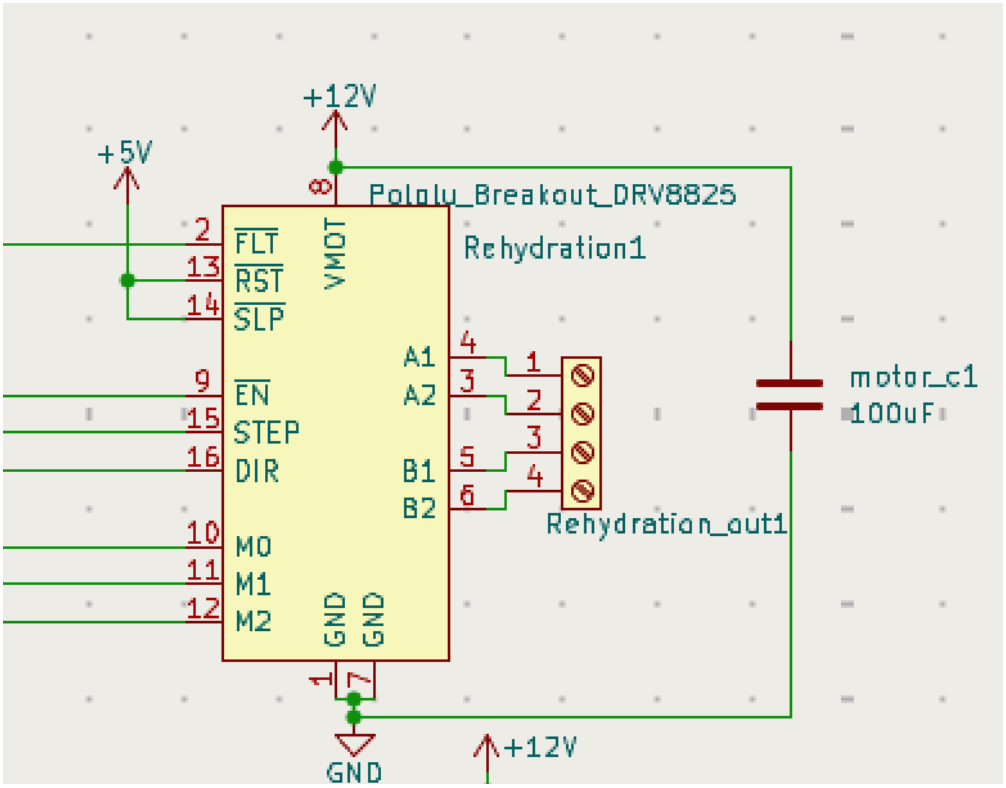


Figure 24: DRV8825 stepper motor driver circuit used for the rehydration syringe pump.

The Pololu DRV8825 stepper motor driver shown in Figure 24 was chosen for this project due to its affordability, high current handling capabilities, and flexible configuration options. Specifically, it supports up to 2.2 A per coil (with adequate cooling), features built-in fault protection, and enables microstepping down to 1/32 steps, which is critical for the control required in both the syringe pump and movement platform subsystems. Additionally, its

12 V capability allows it to operate within the system's existing power budget shown in Appendix A.

11.1 Driver Control

As shown in Figure 24, the DRV8825 is powered from a 12 V supply connected to the `VMOT` pin. A 100 µF electrolytic capacitor is placed near the power input to stabilize the supply voltage and suppress transient spikes, as recommended by the datasheet. The motor coil connections are wired to pins `A1`, `A2`, `B1`, and `B2`, while logic-level signals (STEP, DIR, EN) are controlled via the ESP32.

Stepper motion is achieved by sending pulses to the `STEP` pin, with each pulse requiring a minimum duration of 1.9 µs to register. To meet these timing requirements and maintain modularity, a custom motor control library was implemented. This library uses hardware timers to generate accurately timed pulses and to handle initialization, direction control, and current disable through the `EN` (enable) pin.

The `FLT` (fault) pin is also monitored by the microcontroller. If a fault condition occurs—such as over-temperature, under-voltage, or over-current—the pin is pulled low, allowing the firmware to detect and handle errors gracefully. This mechanism as well as the error handling shown in the embedded state machine in figure 30 assists users in diagnosing issues, such as power loss or wiring faults, in real time.

11.2 Microstepping Modes

The DRV8825 stepper driver supports multiple microstepping resolutions, allowing finer control of the stepper motor. This is essential for the syringe pump in the system, where fluid movement must be precise. Microstepping is controlled via three logic inputs: `M0`, `M1`, and `M2`. These pins have internal pull-down resistors, so when left unconnected they default to logic LOW. The pins are connected directly to the microcontroller's GPIO for software control.

M2	**M1**	**M0**	**Microstep Resolution**
Low	Low	Low	Full Step (1x)
Low	Low	High	Half Step (1/2x)
Low	High	Low	Quarter Step (1/4x)
Low	High	High	Eighth Step (1/8x)
High	Low	Low	Sixteenth Step (1/16x)
High	Low	High	Thirty-second Step (1/32x)
High	High	Low	Thirty-second Step (1/32x)
High	High	High	Thirty-second Step (1/32x)

Table 3: DRV8825 microstepping mode configuration using M0, M1, and M2 pins.

11.3 Heatsink and Cooling Considerations

Each DRV8825 module comes with a small aluminum heatsink. This heatsink are fixed to the underside of the board to the exposed thermal pad. Thermal adhesive tape from the manufacturer is used to attach the heatsink securely. This passive cooling method is sufficient for most applications where the current draw does not exceed 1.5 A per phase.

For higher current applications, additional cooling (e.g., forced airflow) may be required. In the system, the heatsink ensures thermal stability during continuous operation of the syringe pump and movement mechanism.

11.4 Mounting and Current Adjustment

The DRV8825 breakout boards are inserted into standard 16-pin female headers on the custom designed PCB. Proper orientation is critical thus the PCB has squares denoting the proper oreitation as shown in Figure 39

After insertion, the onboard potentiometer is used to adjust the current limit for the stepper motor. This is done by measuring the voltage at the `VREF` test point and using the

following formula:

$$I_{\text{limit}} = 2 \times V_{\text{REF}} \tag{9}$$

Where I_{limit} is the maximum current per coil, and V_{REF} is the reference voltage measured with respect to ground.

For example, with the movement motor the current was limited to 1.0 A, thus the potentiometer should be adjusted until $V_{\text{REF}} = 0.5\,\text{V}$. The data sheet recommends beginning with a lower current and slowly increase it while monitoring motor temperature and performance. Thus, for the movement motor where the required current was calculated as shown in this equation Equation 8, the potentionmeter was adjusted so that V_{REF} is (0.1V) rather than a calculated required voltage of (0.134V) and then slowly increased until the system operated nominally, this was found to be around (0.198V) or 0.4A this is slightly more than calculated but within an expected range. For the syringe pump stepper motor the process was repeated and the voltage and current draw were found be to 0.4 A and 0.2 V which is as expected when calculated as shown in Equation 4

With all major processes—heating, rehydration, mixing, gas flow, and extraction integrated, a centralized method for controlling and monitoring these subsystems became essential. To address this, a user interface was developed to provide the researchers with control over system parameters and real-time feedback during experimental operation.

12 User Interface - Cole Schreiner

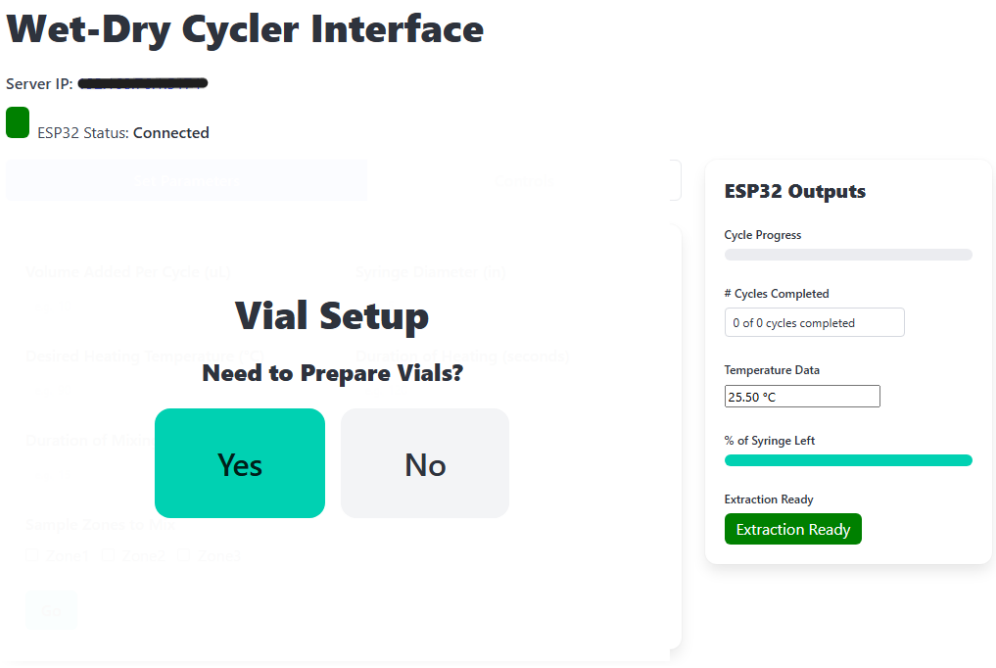


Figure 25: Initial Webpage Presented to the User

The web interface was a critical component of the wet/dry cycling system. It provided researchers with a user-friendly method of interacting with the hardware, bridging the gap between hardware operations and user control. Its primary purpose was to allow researchers to view, configure, and log experimental parameters related to RNA synthesis. By abstracting the complexity of the underlying hardware, the interface enabled experiments to be initiated, monitored, and controlled with ease.

12.1 Purpose and Role in System Architecture

The web interface served as the primary bridge between the user and all hardware systems, enabling real-time interaction during each experiment. It was designed to support the configuration of system states and experimental parameters, provide dynamic control over active processes, and ensure state preservation in the event of power interruptions

or disconnections. This design allowed experimenters to conduct experimental runs with precision and minimal direct hardware intervention.

Technological Stack

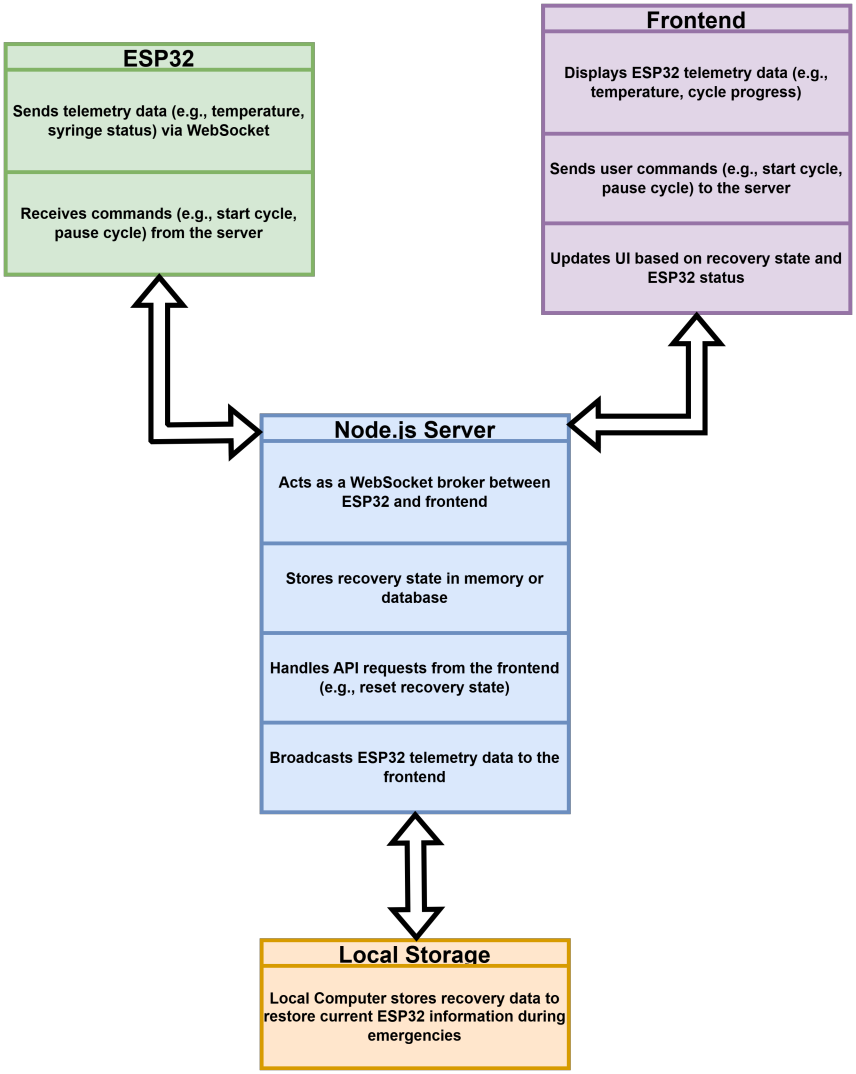


Figure 26: System Interaction Diagram

The web system integrated multiple tools and frameworks, as seen in Figure 26, to ensure seamless interaction between hardware and software components:

React.js (Frontend)

A component-based JavaScript library used to build an interactive, responsive interface. It dynamically updated the user interface based on ESP32 telemetry data and recovery state, ensuring real-time feedback and smooth user experience.

Node.js (Backend Server)

A JavaScript runtime that served the site and maintained a persistent WebSocket connection with both the browser and the ESP32 microcontroller. It acted as a broker for communication and handled API requests efficiently.

WebSocket Protocol

Enabled bidirectional, real-time communication between the frontend, backend, and ESP32. This protocol was essential for maintaining synchronization between user commands and system state, ensuring minimal latency.

ESP32-S3 (Hardware Controller)

A microcontroller that communicated via WebSocket with the Node.js server to receive instructions (e.g., start cycle, pause cycle) and report internal state (e.g., temperature, syringe status). Its integration ensured hardware control.

JSON-Based State Recovery System

A local file system mechanism used to record system state. This ensured state persistence across page reloads, disconnections, and microcontroller resets, allowing recovery during emergencies.

Local Storage

The local computer stored recovery data to restore current ESP32 information during emergencies, ensuring continuity in operation.

12.2 Key Features and Design Considerations

Throughout the development process, particular emphasis was placed on safety and clarity in the interface design. The following features were central to the final implementation:

State Machine Interface

A dynamic state machine was developed and embedded into the frontend logic. This state machine mirrored the internal state of the ESP32 and ensured that both systems agreed on the current operational mode (e.g., Setup, Heating, Mixing, etc.). The frontend transitioned between interface states only when a mutual confirmation from the ESP32 was received. This ensured synchronization and reduced the risk of invalid or conflicting user actions.

Tab-Based Input Access

To enforce a controlled workflow, the interface was split between two tabs. Each tab quarantined a different set of options corresponding to specific stages of the wet/dry cycling process. Inputs were conditionally locked or unlocked based on the current state of the experiment (see figures 25, 27, and 28).

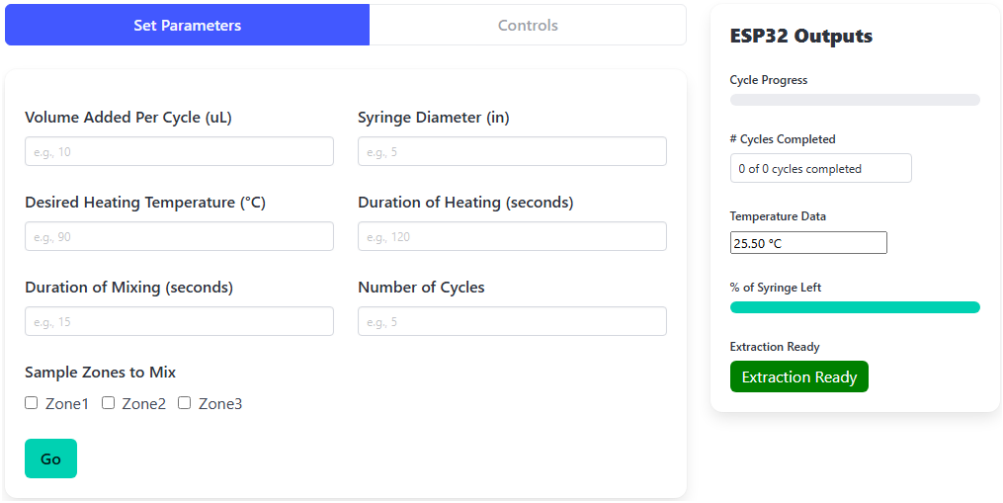


Figure 27: User Interface: "Set Parameters" Tab

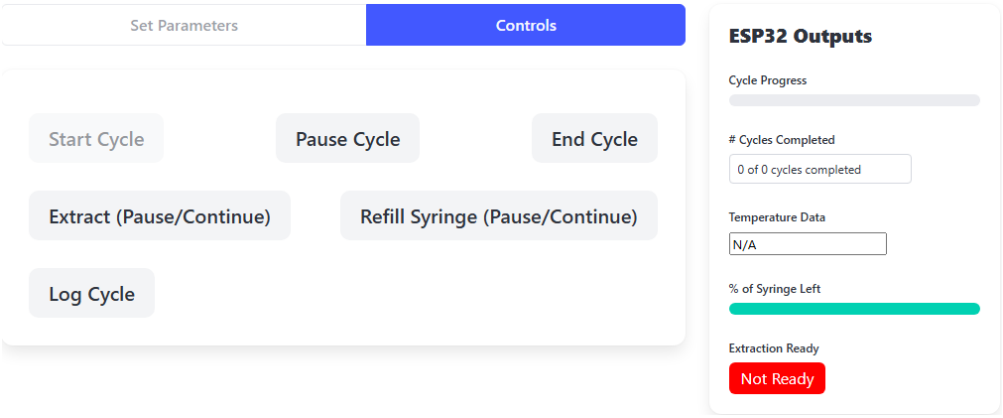


Figure 28: User Interface: "Controls" Tab

Parameter configuration was restricted to the setup state (Figure 27) and became locked once an experiment began. During active experimentation, user interaction was limited to the "Controls" tab (Figure 28), with individual control buttons disabled while others were in use. This quarantined-access design minimized user error and ensured state machine integrity across repeated cycles.

Recovery and State Persistence

One of the most important aspects of the system was its ability to persist state across disruptions. A JSON-based state recovery mechanism that recorded the current frontend

and backend state to disk was implemented. Upon reloading the web page, the frontend queried the backend for the last known state and re-entered the exact interface state present before reload or power loss. Similarly, the ESP32 retained its state through local storage and resumed operation coherently upon reconnecting to the backend. This design ensured that experiments were not disrupted by power interruptions or accidental browser closures.

Real-Time Feedback and Logging

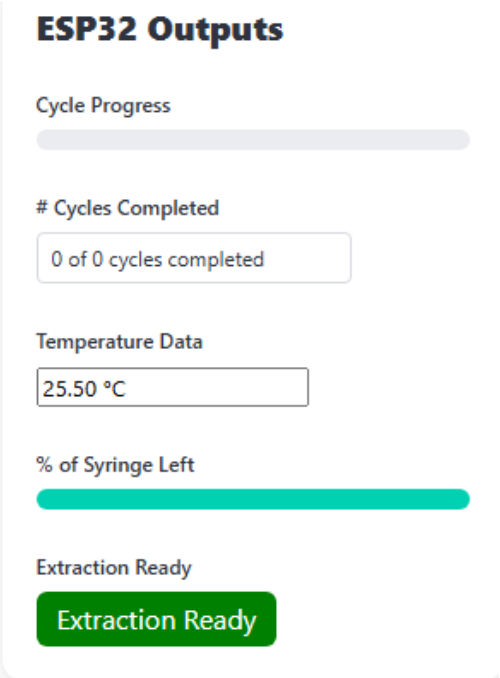


Figure 29: User Interface: Real-Time Feedback

The frontend continuously displayed the current state of the hardware and relevant telemetry, such as temperatures, timing cycles, and pump status (see Figure 29). Any change made by the user or ESP32 was immediately reflected in the system and logged. This real-time interactivity was enabled by the persistent WebSocket connection, which minimized latency and kept the interface in sync with the hardware.

State Management and Synchronization

Ensuring agreement between the frontend UI state, backend server state, and microcontroller state was a particularly challenging problem (see Figure 26). Multiple paradigms for shared state were explored, and the team eventually settled on a file-based persistence system that could be accessed by both frontend and backend on reload. Considerable effort was spent debugging edge cases where components could fall out of sync, especially during power loss scenarios or user-initiated reloads.

12.3 Design Evolution and Final Implementation

Creating an intuitive user experience proved to be as challenging as the technical backend. The UI evolved from a simple OLED screen into a visually appealing webpage. As the team's understanding of real-time systems deepened, the state machine framework was introduced, and testing was conducted to verify state integrity under all circumstances. The final interface was an essential integration for the overall system and was designed to be maintainable and extensible.

The web interface successfully met the goals of providing a platform for managing RNA wet/dry cycling experiments. It represented a substantial technical achievement for the team, requiring integration of full-stack development and hardware communication. The final system enabled experimental control, even under failure conditions, and served as a foundation for future extensions. Alongside the website, backend communication was developed to enable coordination between the user interface and the physical hardware.

13 Embedeed Architecture and Code Libraries - Rafael Delwart

13.1 Overview of the Codebase

The firmware is structured using a modular architecture for each subsection, this allows for execution of the wet/dry cycling process. Each component of the system; heating, mixing, fluid rehydration, and vial movement is abstracted into dedicated libraries, allowing for clean separation of code and simplified debugging. All software development adhered to standard modular design principles and was documented according to IEEE Std 829-2008 for software and test documentation [5], ensuring consistent structure, clarity, and traceability across all modules.

A finite state machine governs how the system transitions through operational phases based on internal flags, timers, and WebSocket commands. This structure is essential for synchronizing hardware activities and maintaining system stability during long-duration experiments.

Figure 30 shows the embedded system state machine that drives the execution logic. Each state corresponds to a defined physical process and is mapped to functions implemented in `main.cpp`, `globals.h`, and peripheral libraries. The full source code for the firmware, including all libraries and configuration files, is available on GitHub. See Appendix B for the link.

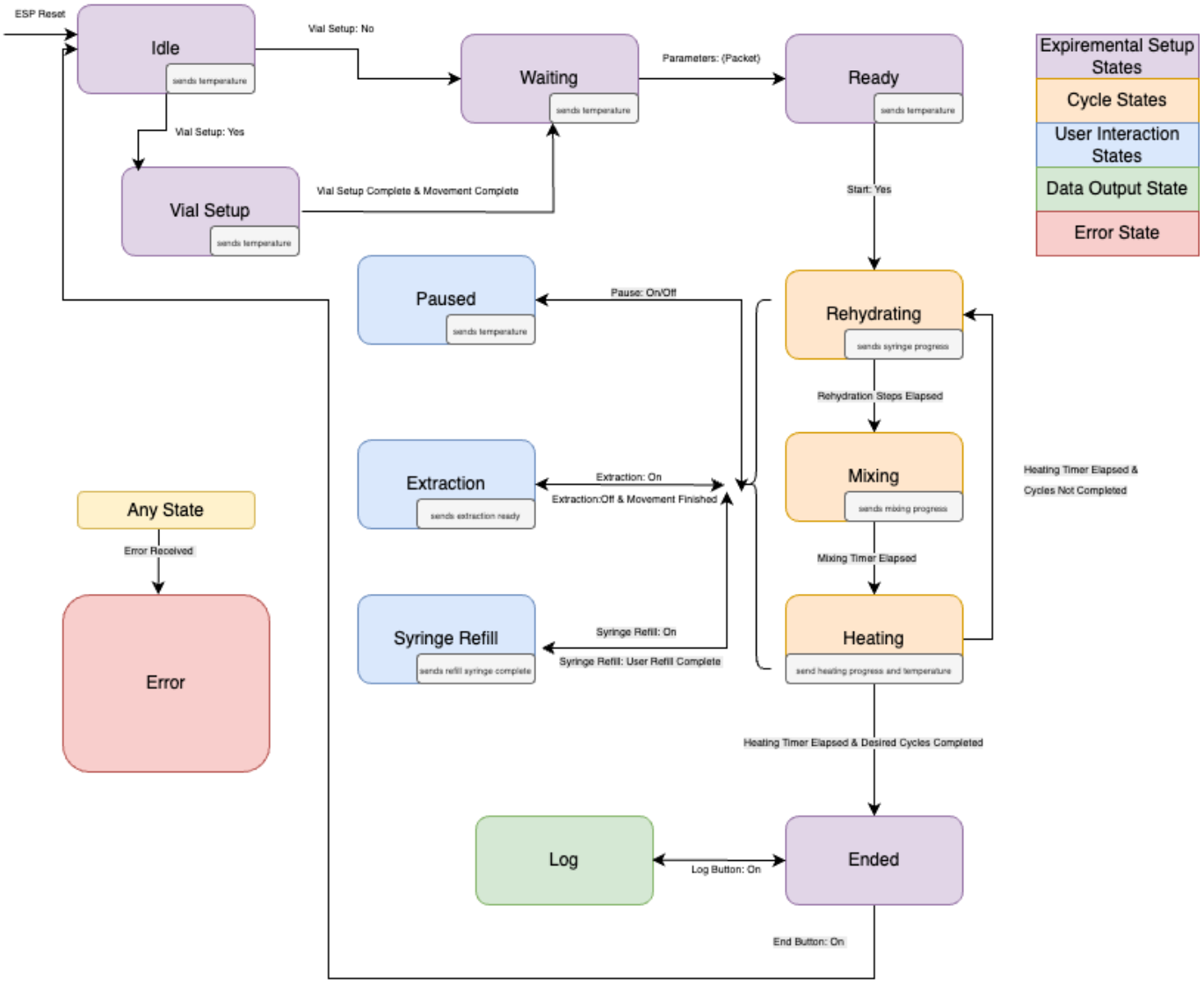


Figure 30: State machine diagram for the embedded control system. Arrows indicate valid transitions based on input events and subsystem completion.

13.2 System State Machine

The `SystemState` enumeration defines the system's operational phases. It begins in **IDLE**, awaiting vial setup instructions from the front-end, then transitions to **VIAL_SETUP** for positioning using bumper-based feedback. In **WAITING**, the system pauses for parameters input, after which it enters the **READY** state, signaling readiness to begin the wet/dry cycle. The **REHYDRATING** state triggers the syringe pump to dispense fluid, followed

by **MIXING**, where digital motors agitate the sample. **HEATING** activates the silicone heating pad for dehydration. At any point within the cycle the system can enter **REFILLING** to retract the syringe fully, or may switch to **EXTRACTING** for vial access. **LOGGING** records system state transitions and sensor readings, as illustrated in Figure 30, allowing researchers to review and analyze the data at a later time. The system can enter **PAUSED** to halt operation while preserving progress, **ENDED** upon successful cycle completion, or **ERROR** in response to safety violations.

State transitions are driven by timed events (e.g., heating duration), bumper-triggered interrupts, WebSocket commands from the user interface (e.g., parameter submission or pause), and internal flags such as `heatingStarted` or `mixingStarted`, which ensure state-specific actions occur only once per cycle.

This state machine structure allows the system to pause, resume, or recover mid-process, an essential feature for long-duration scientific experiments.

13.3 Initialization and WiFi Setup

The `main.cpp` file is the startup point for the firmware and is primarily responsible for initializing all subsystems and establishing network connectivity. Upon startup, it begins by configuring the ESP32's non-volatile storage (NVS) and initializing serial communication for debugging.

Next, the device attempts to connect to the local WiFi network using credentials the user-defined. Once connected, it initializes the WebSocket server to enable real-time communication with the frontend.

Following network setup, all major hardware modules are initialized. This centralized process ensures that peripherals start in a known state, and any initialization errors can be sent to the user for debugging.

13.4 Frontend Communication Handlers

To facilitate real-time interaction between the embedded system and the user interface, two primary sets of functions are implemented: `handleFunctions` and `sendFunctions`. These ensure bidirectional communication over WebSockets.

The `handleFunctions` are responsible for interpreting incoming packets from the front end. Each packet typically includes a command type and relevant data payload. Depending on the system state and the received instruction, such as `VIAL SETUP`, `START CYCLE`, or `PAUSE CYCLE`, these functions trigger corresponding actions within the system state machine.

Conversely, the `sendFunctions` package and transmit system updates back to the frontend. These include real-time state changes, sensor readings, and recovery data. This continuous feedback loop ensures the frontend remains synchronized with the system's current operation and allows the user to monitor and control the process remotely.

13.5 Mixing Control Module

The `mixing.h` header defines the interface for managing the three DC mixing motors, each assigned to a distinct vial section and controlled via dedicated GPIO pins on the ESP32. Upon system startup, the `initMixing()` function is called to configure the necessary GPIO modes and ensure that all motors are initially turned off. During operation, specific motors can be activated using `startMixing(index)`, which enables the appropriate MOSFET to begin mixing in the designated vial section. To stop mixing, the firmware calls `stopMixing(index)`, which deactivates the corresponding motor. For transitions or global halts, `stopAllMixing()` provides a utility to simultaneously disable all three motors, ensuring safe and synchronized control throughout the mixing phase.

13.6 Heating Control Module

The `heating.h` header facilitates control over the silicone heating pad used for sample dehydration. Initialization is handled by `initHeating()`, which sets up the required GPIO pins and configures the ADC channel for temperature monitoring via a thermistor. When heating is needed, the system runs `startHeating()` to enable the MOSFET supplying power to the heating pad. Conversely, `stopHeating()` turns off the heat by disabling the MOSFET. Temperature feedback is provided by `readTemperature()`, which samples the voltage divider and uses the thermistor's Beta equation to compute an estimated temperature. The function `setTemperature()` then checks the user-defined desired temperature, and runs startHeating once readTemperature returns the desired value the MOSFET is turned off using stopHeating. This cycle is repeated to implement the bang -bang control seen in figure 5.

13.7 Stepper Motor Driver Module

Defined in `DRV8825.h`, this module abstracts the low-level control logic required to operate the DRV8825 stepper motor driver, which supports both the movement system and the syringe-based fluid delivery mechanism. To prepare the driver, `initDRV8825(*motor)` configures GPIO pins for stepping, direction, enabling, and microstepping mode selection using a pointer to a motor struct which holds that specific motors pin numbers. Motor movement is executed using `stepMotor(*motor, steps, direction)`, which issues a defined number of timed step pulses. For dynamic enable/disable control, `enableMotor(*motor)` and `disableMotor(*motor)` allow the system to engage or disengage the motor when necessary. Directional control is established via `setDirection(*motor, direction)`, and microstepping behavior can be customized through `setMicrostepMode(*motor, mode)`. The higher-level interface `move(*motor, steps, direction, delayUS)` integrates these features to deliver precise movement using configurable step count, direction, and delay, enabling both fine and coarse control over syringe or sample plate motion.

13.8 Rehydration Module

The `rehydrationmotor.h` header provides an abstraction over the DRV8825 stepper motor driver to control a syringe pump responsible for fluid handling during the `REHYDRATING` and `REFILLING` states. Initialization via `initRehydrationMotor()` sets up the driver's GPIOs, enables microstepping (typically 1/16), and disables the motor by default to save power.

To dispense fluid, the `rehydrate(volume_uL)` function converts a user-specified target volume in microliters into motor steps. This conversion is based on the physical geometry of the system, using the following formula:

$$\text{Steps} = \frac{\text{Desired Volume (µL)}}{\text{µL per Step}} \tag{10}$$

The volume dispensed per step is computed using:

$$\text{µL per Step} = \left(\pi r^2 \cdot \frac{\text{Leadscrew Pitch (in/rev)}}{\text{Steps per Rev}}\right) \cdot 16387 \tag{11}$$

where:

- r is the syringe radius in inches,
- Leadscrew Pitch is the linear travel per revolution (inches),
- Steps per Rev accounts for both motor steps and microstepping,
- 16387 is the conversion factor from in^3 to µL.

This method allows the system to calculate step counts and call `DRV8825_Move()` to push a precise amount of water. To prepare for the next cycle, `refillSyringe()` retracts the plunger until the rear bumper switch is triggered. The motor is then disabled with `stopRehydrationMotor()` to reduce heat and power draw. Overall, this module provides intuitive volume-based control, abstracting away mechanical step details.

13.9 Movement (Extraction) Module

The `movement.h` header defines functionality for driving the stepper motor that positions the vial platform during vial setup and sample extraction. To move the platform to a particular vial position, the system uses `moveToPosition(position)`, which computes the required step count and calls `move()`. Homing is handled by `homeMovement()`, which drives the tray toward a physical switch to establish a known zero reference point. For situations where full positioning is unnecessary, `moveSteps(steps, direction)` offers direct, low-level step control. Finally, `stopMovementMotor()` disables the driver to avoid excess power draw and heat buildup when idle. Altogether, this module abstracts platform indexing and movement logic, enabling seamless coordination with other subsystems in the state machine.

13.10 Error Handling

Error handling is essential for ensuring system reliability and safety. The firmware includes basic fault detection mechanisms and communication faults to report issues to the frontend interface in real time.

13.10.1 Initialization Errors

During system startup, all subsystems—including WiFi, WebSocket server, motor drivers, sensors, and peripheral modules—are initialized sequentially. Each initialization function includes internal checks (e.g., pin configuration status, sensor availability). If any of these checks fail, an error flag is set, and a detailed message is sent to the frontend via the WebSocket connection. This allows the user to identify and resolve configuration issues before proceeding.

13.10.2 Runtime Monitoring

Throughout operation, the system continuously monitors critical parameters:

Temperature Out-of-Range: If the temperature read by the thermistor exceeds safe limits, the heating subsystem is automatically disabled, and an alert is sent to the frontend to prevent overheating or thermal damage.

DRV8825 Fault Detection: The DRV8825 stepper motor drivers feature a fault output pin that signals overcurrent, thermal shutdown, or undervoltage lockout. These fault lines are monitored by the ESP32. If a fault is detected on either the syringe pump or movement motor, the corresponding subsystem is immediately disabled, and a detailed fault report is sent to the frontend. This helps protect the driver and motor from damage.

Limit/Bumper Failures: If a movement or refill routine fails to trigger a bumper switch within the expected number of steps, the motor is stopped to prevent mechanical overrun. The system then enters a safe `PAUSED` state and notifies the user of the fault condition.

Communication Timeout: If the system detects a prolonged loss of communication with the frontend (e.g., due to WiFi dropout or browser closure), it halts all active subsystems. This ensures that no processes continue unsupervised, preserving experimental integrity.

13.10.3 Frontend Alerts

When an error is detected, a descriptive packet is sent to the frontend using `sendFunctions`, including the error type, subsystem affected, and suggested corrective action if applicable. These alerts are displayed to the user through the interface, providing immediate feedback and guiding troubleshooting steps.

This error handling framework ensures that both developers and researchers can detect, diagnose, and respond to system faults without interrupting workflow or compromising experimental integrity. A power system was implemented to supply consistent current and voltage to all hardware subsystems throughout operation.

14 Power System Design - Matthew Randall

The power subsystem was designed to meet the following primary objectives: ensure stable voltage regulation and consistent current supply to all components under varying load conditions. To implement this, the following top-down approach was realized: a power budget was created to understand the system's load requirements, a power system block diagram was developed to establish the overall architecture and an engineering schematic was produced to validate the design and support testing.

14.1 Engineering Standards Employed for Power System Design

The power subsystem design adhered to established engineering standards to ensure clarity, maintainability, and electrical safety. The following standards were used: technical documentation practices outlined by the UCSC ECE Department [9], power supply design principles described in National Semiconductor's Application Note 556 [8], and NEC ampacity guidelines for wire sizing based on current and length [7].

14.2 Power Budget and Load Requirements

Wet/Dry Cyclying Power Budget 24 hr cycle					Working Concept		V2.2 6/6/2025								
							State-based Current (mA)			Time in State (hr) 100% duty = 24hrs					
Item	Value	Quantity	Nominal Voltage (V)	Needed Input V Regulation	Output V Variation	Spectral Purity /Noise	high	low	off	high	low	off	Total Consumption (mA-hr)	Max Power (mW) (for design reference)	Total Energy/day (mW-hr)
12V Power Supply		1	12 output	N/A	14.8V max out 11.6V	N/A	22000	17500		1	23				
Microcontroller board	ESP32	1	5	4.75V min 5.25V max	3.3v: ±12%	0.003% RMS	500	14.6	8	24	0	0	12000	2500	60000
5V linear Regulator	LM117	1	12	7V min 15V max	5v ± 0.1V	10 uV (RMS)	800	5.00E+00	8.00E-01	0.25	23.25	0	316	9600	3795
Stepper Motor Driver	DRV8825	2	12V	8.2V min 45V max	N/A	N/A	2200	1.00E+03	5.00E+01	0.25					
Heating Pad	210W silcon heat pad	1	12	15V max 12V min	N/A	N/A	17500	0	80	22	0	2	385160	210000	4621920
Dc Stepper Motor	NEMA 17	2	12	11.6V min 24V max	N/A	2.3V (max peak to peak) noise	2000	1.00E+03	0.00E+00	1	0	23	4.00E+03	24000	4.80E+04
Heat Pad Controller		1	3.3	± 50mV	N/A	N/A	2	0	0.00E+00	22	2	0	4.40E+01	6.6	1.45E+02
Brushless Dc Motor		3	12	N/A	N/A	Not yet measured	200	46	1.00E-05	0	1	23	1.38E+02	2400	1.66E+03
Resistor	10000 Ohm	1	n/a										TOTAL: 4.0E+05	TOTAL: 2.5E+05	TOTAL: 4.7E+06

Figure 31: Power budget created to define the system load requirements

Figure 31 shows the created power budget spreadsheet to define the system load requirements.[1] The analysis considered each component's voltage requirements, current consumption across different states, and total energy usage over a 24-hour duty cycle.

The heating pad represents the dominant power load in the system, accounting for the majority of total energy consumption. Operating at 12 V, it requires 17.6 A, requiring consideration of both power regulation and thermal management. In addition, multiple motors—including NEMA 17 stepper motors and brushless DC motors—contribute significant current draw, ranging from 50 mA to 2 A, during active states. These components require high instantaneous current and introduce spectral noise that must be managed through proper isolation and grounding strategies. It's important to note that the two stepper motors for rehydration and movement systems will never be in high states simultaneously. These high current demands lead to the selection of a power supply able to maintain 12 V for high current draw.[2]

[1] See Appendix for linked power budget

[2] power supply specifications are shown in row 4 of power budget

14.2.1 Power Supply Selection

Due to the high current demands of components such as the heating pad, stepper motors, and motor drivers, a 12 V power supply was selected to serve as the primary voltage source. This choice was based on its ability to meet the system's load requirements while maintaining voltage within the acceptable input ranges. To ensure the correct power supply was selected, the pugh chart shown in Figure 32 was created.

12V Power Supply Pugh Chart V1.1 1/20/2025

Proposed Supply	Link	Cost (x2)	Number of Output Ports (x1)	Max Load Current (x4)	Ease of Use/Compatability (x1)	Efficency (x4)	Total
UniSource PS-1230 PS Switching Power Supply	https://www.valu	$333 (score 2/10)	1 (score 5/10)	30A (score 10/10)	(score 10/10)	73% (score 7/10)	87
DF-1763 Universal 110/220 Volt AC to 12V DC Power Converter, 10 Amps	https://www.volta	$60(score 6/10)	1 (score 5/10)	10A (score 2/10)	(Score 10/10)	80% (score 8/10)	67
Galggy 120 to 12V 350W supply	https://www.ama	$24 (score 10/10)	3 (score 9/10)	33A (score 10/10)	(score 6/10)	70-90% (Score 8/10)	112
Astron SS-30-AP	https://theantenr	$209(Score 4/10)	2 (score 7/10)	30A(score 10/10)	Score(8/10)	85% (Score 9/10)	99

Figure 32: Pugh chart created for power supply selection.

From Figure Figure 32, the Galggy 120 to 12 V supply emerged as the best choice with a total score of 112, driven by its low cost, high current capacity, and multiple output ports.

The 12 V power supply provided 12 V on three parallel ports with a maximum current draw of 33 A. The stability of the power supply was then experimentally tested by placing a volt meter on a parallel port, while the high current heat pad and motors were turned on. The power supply was measured to deliver stable a 12 V operating up to 25 A with 1.6% error in voltage. Once it was deemed that the power supply was suitable for load requirements of the system, a power system block diagram was created to give a high level overview of the power distribution.

14.3 Power System Block Diagram

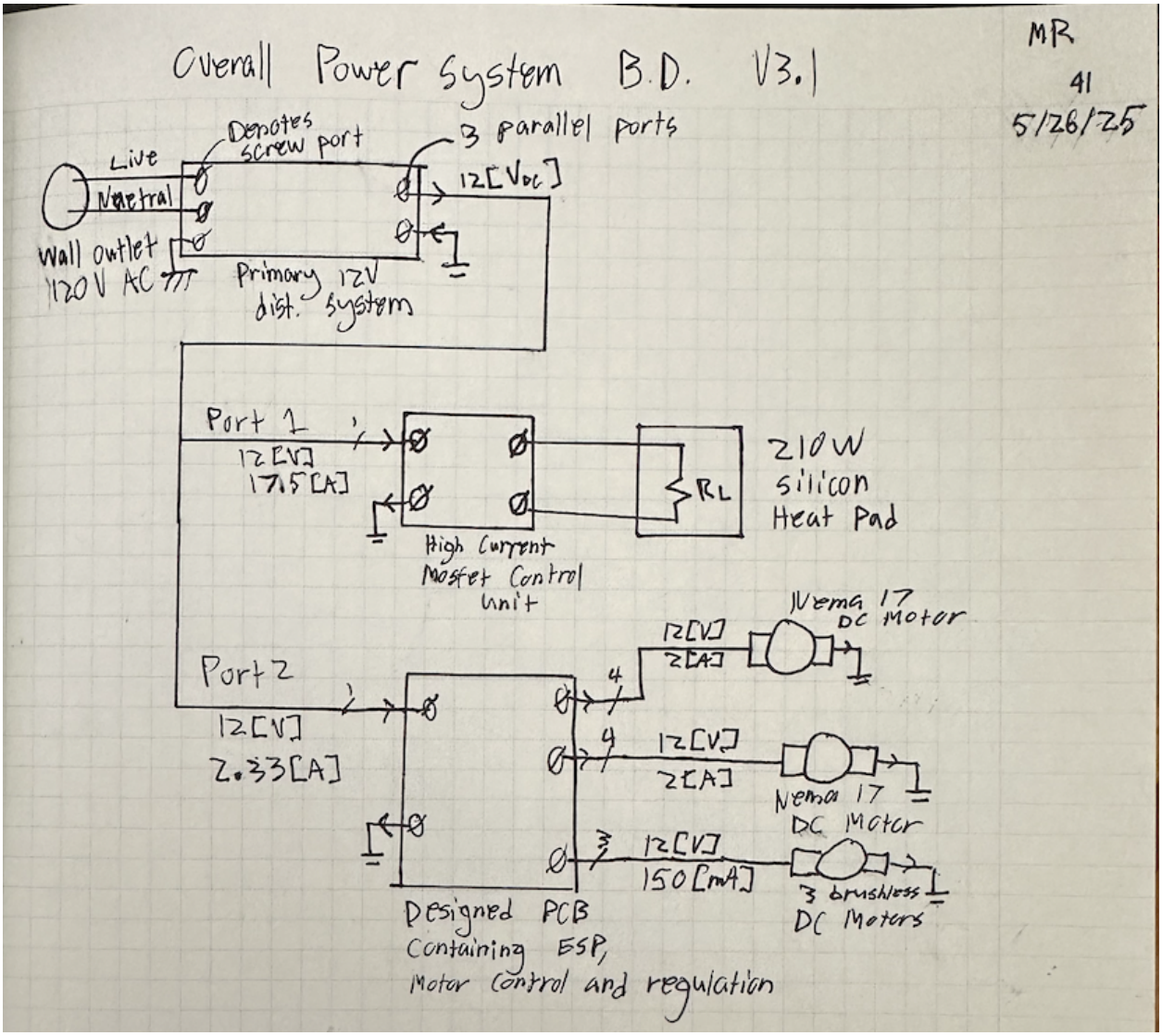


Figure 33: Power system block diagram V3.1

Figure 33 illustrates the high-level 12 V DC power distribution design. The system begins with the selected 12 V power supply, which gains power from the 120 V_{AC} wall outlet. From the 12 V distribution system, three parallel output ports are defined, each serving different subsystems with distinct load requirements.

Port 1 supplies 17.5 A at 12 V to a high-current mosfet control unit, which drives a 210 W silicon heat pad. Port 2 outputs 2.33 A at 12 V to a custom-designed PCB that contains an

ESP-based microcontroller responsible for all motor control. The PCB distributes power to two NEMA 17 DC motors, each consuming 2 A at 12 V, and an additional output supplies 150 mA at 12 V to three brushless DC motors. Notice that the loads on port 2 add up to more than 2.33,A, however the rehydration and movement stepper motors will never be on concurrently.

The diagram demonstrates a top-down structure with signal flow proceeding vertically. Voltage levels and current ratings are labeled for each component. Functional blocks are distinctly grouped, providing a self-documenting overview that aligns with UCSC engineering documentation standards.

Due to the high current demands, appropriate wire gauge selection and return paths must be defined. In order to better understand those requirements, the block diagram in Figure 33 was decomposed into a load analysis shown in Figure 34.

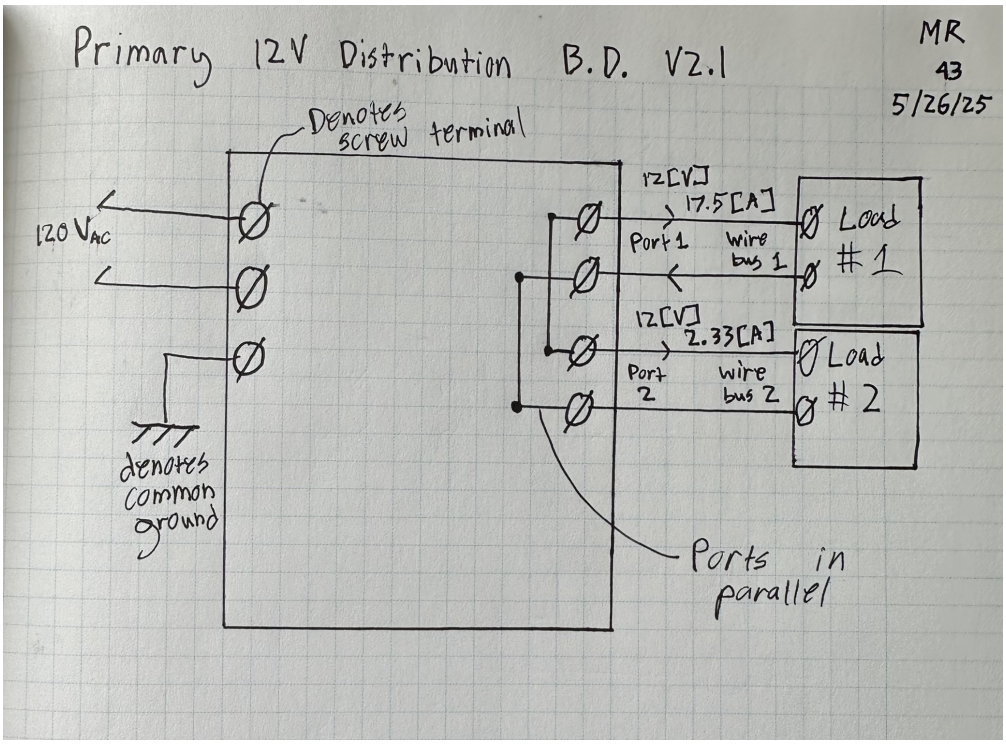


Figure 34: Block diagram of the 12 V power supply and load connections

Decomposing the overall power system into the power supply block diagram shown in Figure 34 supported two key design considerations: selecting appropriate wire gauges and

ensuring proper current return paths. Each wire bus was assigned a gauge based on expected current load and length. Wire gauge becomes a necessary parameter when significant voltage drops or excessive heat generation occurs. For the heat pad in particular, the primary concern was the high thermal load from dissipating 210 W.

Table 4 shows the parameters used to determine the appropriate wire gauge using the NEC standards for wire gauge [7].

Wire Bus	Current Draw	Length	Voltage Drop	Wire Gauge
1	17.5 A	3 ft	2%	12 AWG
2	2.33 A	2 ft	2%	18 AWG

Table 4: Criteria used to select appropriate wire gauge

Table 4 lists the parameters used to calculate the required wire gauge for each wire bus based on current draw, length, and allowable voltage drop. Although the NEC standards permitted smaller gauges, larger wire sizes were selected to provide additional safety margin.

The other key consideration derived from Figure 34 was the design of current return paths. It was determined that each load should have dedicated supply and return wires connected directly to the power supply, allowing the power supply's heatsink to dissipate the full current draw. Sharing ground paths between components can create uneven return currents and increase the risk of damage, so all grounds were routed directly back to the power supply for component safety.

Based on the power system block diagram in Figure 33, an engineering schematic was developed to clearly illustrate the connections and support system debugging.

14.4 Overall System Engineering Schematic

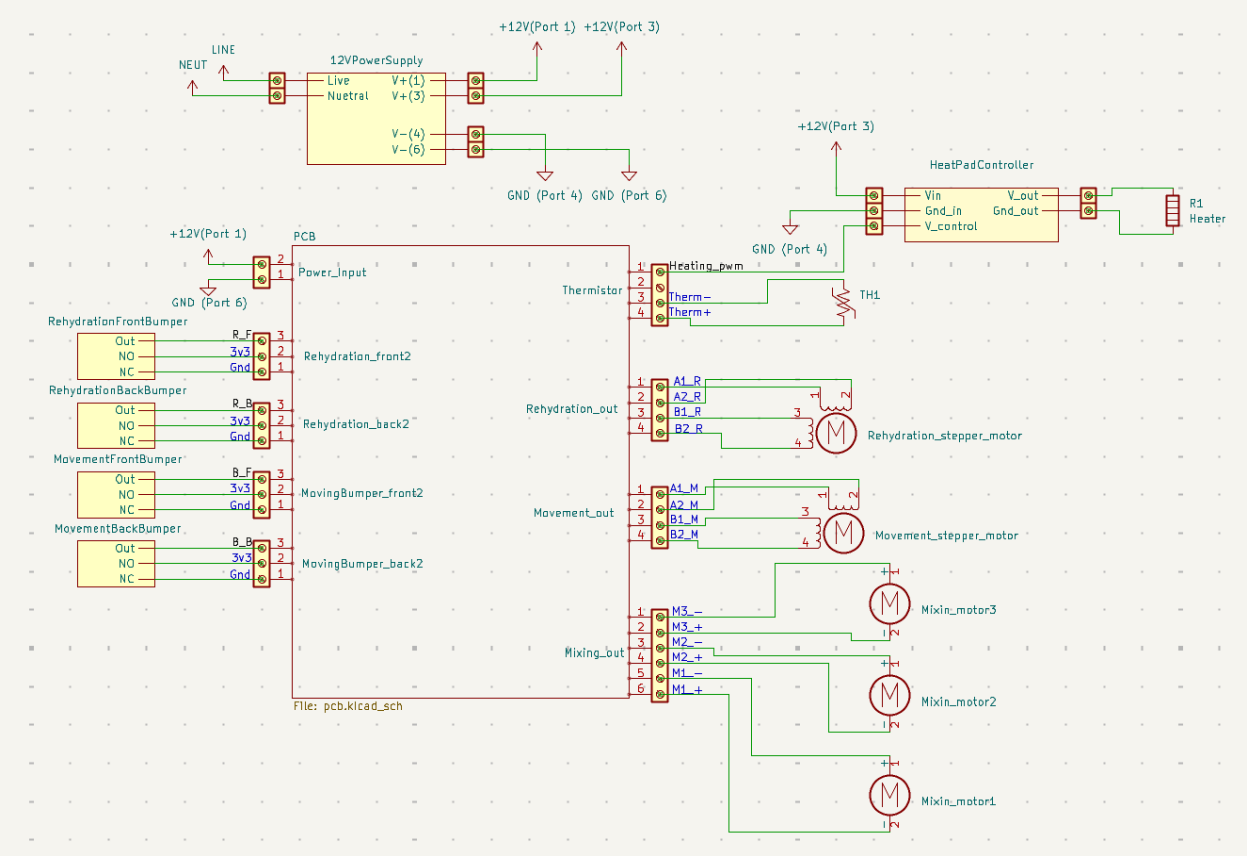


Figure 35: Top level engineering schematic of all electrical components

Figure 35 translates the power system block diagram into a detailed representation of component interconnections and signal flow. At the center is a 12 V power supply that distributes regulated voltage to all major subsystems, including the custom-designed PCB and the heat pad controller. The heat pad controller receives dedicated 12 V and GND lines to accommodate its higher power demands. The PCB is powered through a separate port and distributes appropriate voltage rails to the onboard microcontroller, limit switches, and DC motors.

Signal naming conventions such as `Heating_pwm`, `Rehydration_out`, and `Movement_out` are consistent with the firmware, which facilitates hardware-software integration. The screw terminals are depicted using KiCad's terminal block symbols, providing clarity for physical

wiring. This engineering schematic was primarily used to visualize power flow within the electrical system and to assist with system wiring and debugging. The following section will discuss the PCB design in greater detail to further illustrate the electrical architecture.

15 PCB Design, Matthew Randall & Ryan Taylor

To simplify the electrical system, the subteam designed a PCB to integrate the microcontroller, motor controllers, and voltage regulator. The PCB was developed with the goal of routing all required microcontroller signals and enabling controlled power delivery to the motors via the external power supply. The entire PCB design workflow was created using KiCad. In designing and implementing this PCB, several engineering standards were followed to ensure reliability and compatibility across system components.

15.1 Standards Employed for PCB Design

The PCB design adhered to established engineering standards to ensure performance and clarity of work. The design followed the standard workflow for PCB design. The design began with an engineering schematic that accurately captures all relevant components and interconnections in a self-documenting format, as described in the UCSC documentation guidelines [9]. Power and ground traces were routed using wide, low-inductance paths to minimize voltage variation due to transient current demands, consistent with good engineering practice for PCB layout [10]. Bypass capacitors were placed close to IC power pins to filter noise and stabilize voltage, mitigating the effects of parasitic inductance as detailed in Petersen's power distribution note [11]. Throughout the design process, schematic versions were archived and annotated to support traceability and ensure design changes could be clearly documented. These standards collectively supported a professional PCB design.

15.2 PCB Schematic

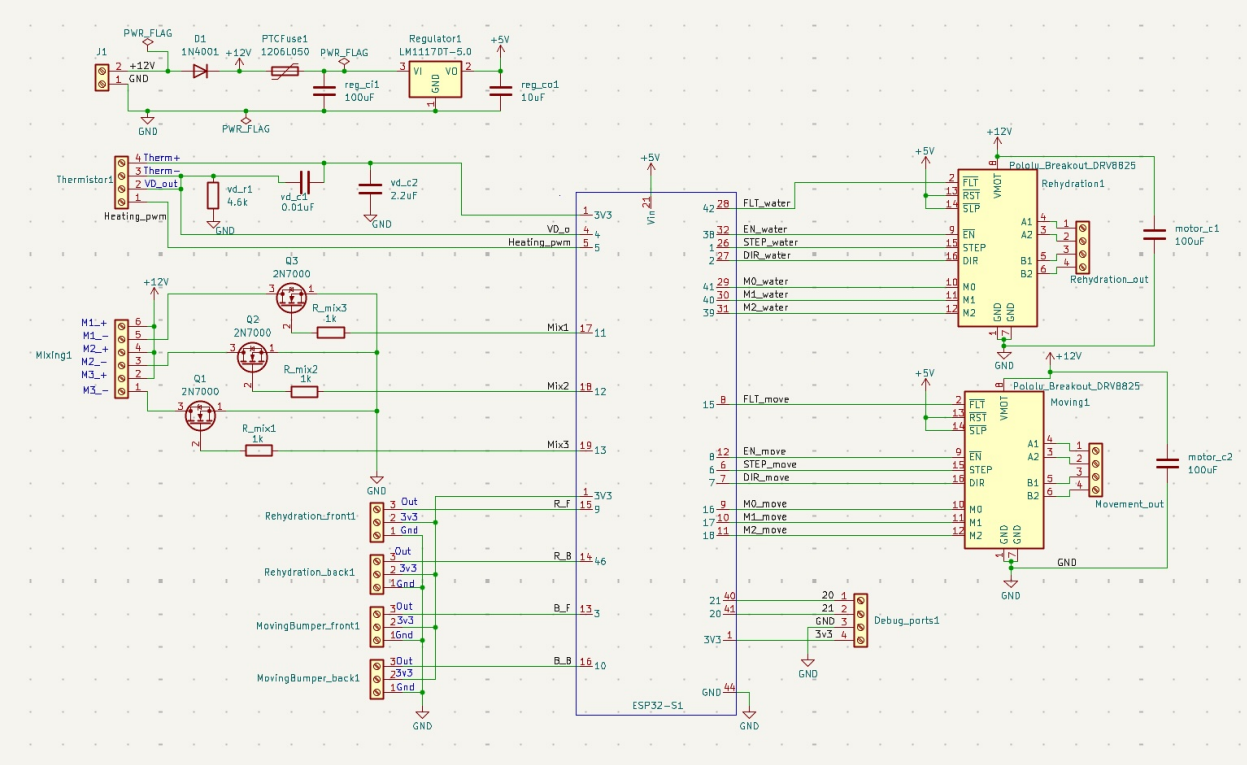


Figure 36: PCB schematic showing relevant components and interconnections

As previously mentioned, the PCB design process begins with the creation of an engineering schematic. This schematic must clearly and accurately represent all components and interconnections in a self-documenting format to support effective PCB layout. Figure 36 shows the schematic used for PCB design. The PCB receives 12 V and ground from the power supply, thermistor input for heating control, and limit switch signals for movement and rehydration system feedback. Outputs from the PCB include control signals for the movement and rehydration stepper motors, voltage signals for the mixing motor, a GPIO ON/OFF signal for heating control, and power to the limit switches. All input and output signals are fed through screw terminal blocks for ease of wiring. The debug ports were added to ensure proper operation of the 3.3 V rail and the microcontroller signals.

The 12 V input rail supplies power to the stepper motor controllers and DC mixing

motors. This 12 V input is regulated down via a linear regulator to generate a 5 V rail, which powers the microcontroller and pulls the reset and sleep pins high on the stepper motor controllers. The LM1117DT 5V linear regulator was selected for this step-down application due to its ability to supply up to 0.8 A of current, which meets the ESP32-S3's power requirement of 0.5 A during Wi-Fi transmission, as specified in the power budget. This provides overhead for the sleep and reset pins on the DRV8825, even though they consume negligible amount of current. On the input a safety diode is placed to prevent reverse voltage from damaging components. The diode chosen was the 1N4001 because it can safely handle 50V and can operate safely in the working temperature range. The last safety piece is a 1206L260 resettable PTC-fuse rated for 2.6 hold current at 12V, this is within specification for the PCB. Through the 5V regulator goes onto the ESP32-S3 breakout board with it's own 3.3V regulator, which is used for the thermistor voltage divider to monitor temperature and to power the limit switches.

The schematic in Figure 36 was then converted into a PCB layout using KiCad, which provided the necessary component symbols and footprints for both stages of the design.

15.3 PCB Layout

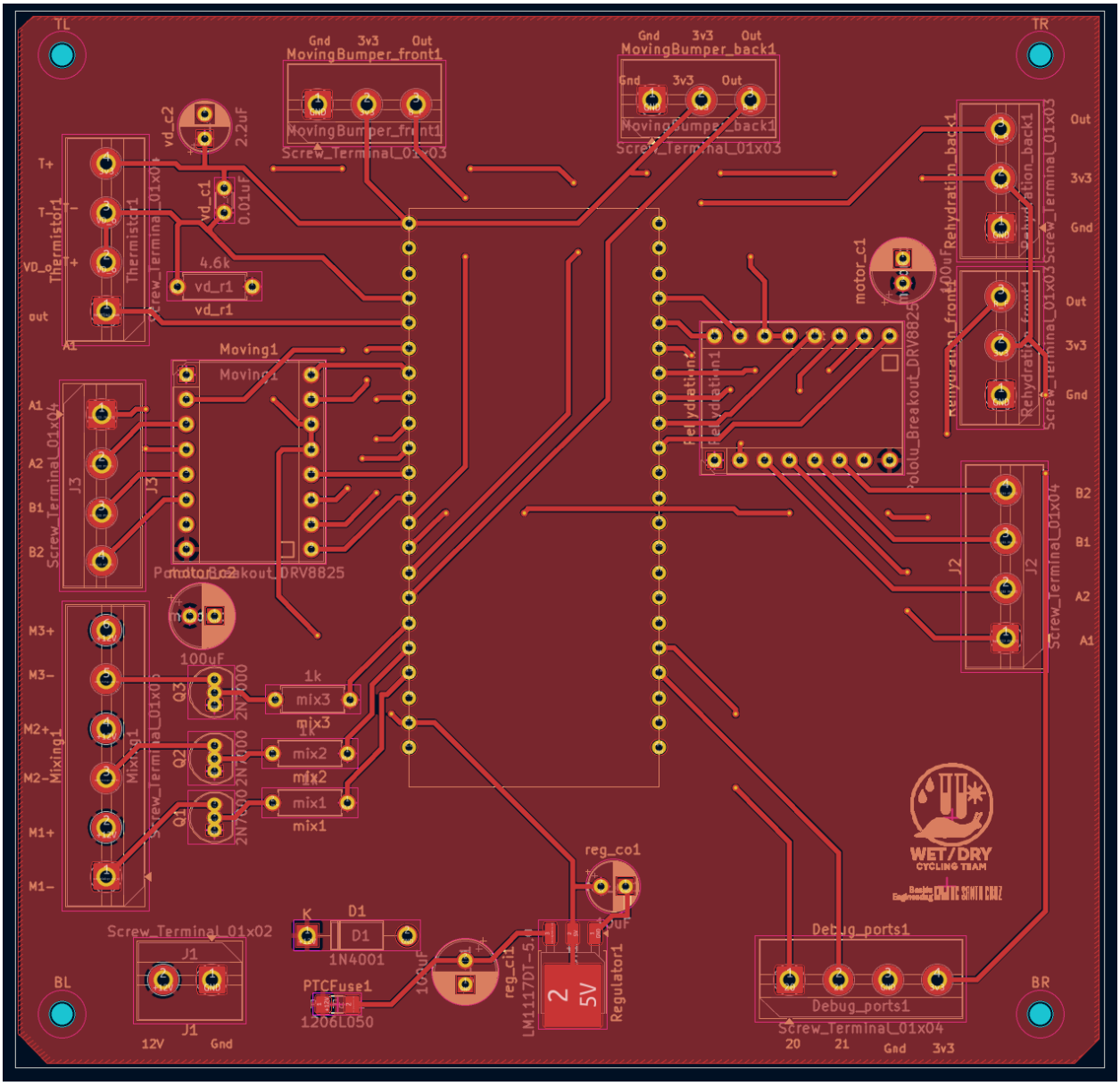


Figure 37: Top layer of designed PCB

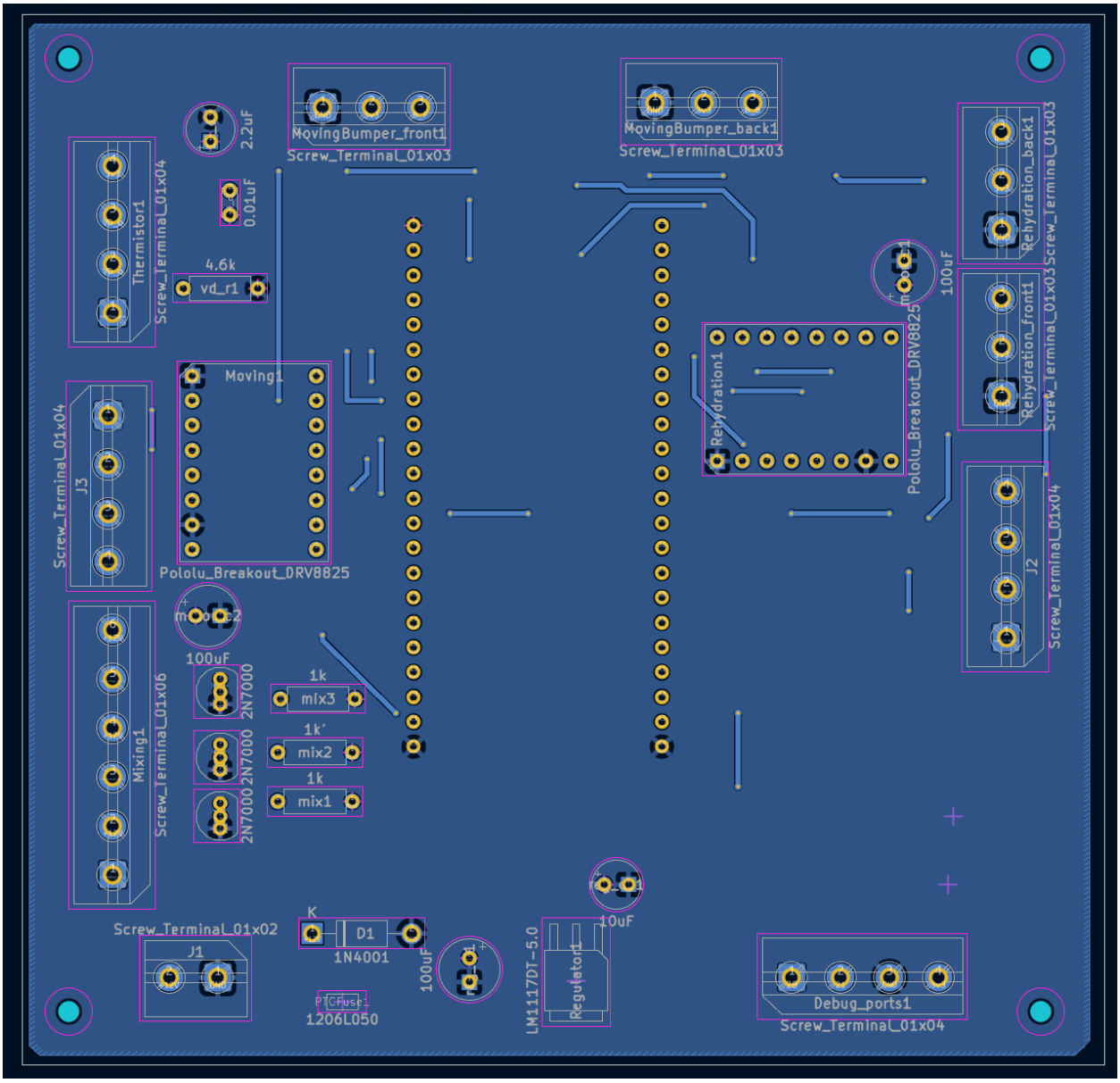


Figure 38: Bottom layer of designed PCB

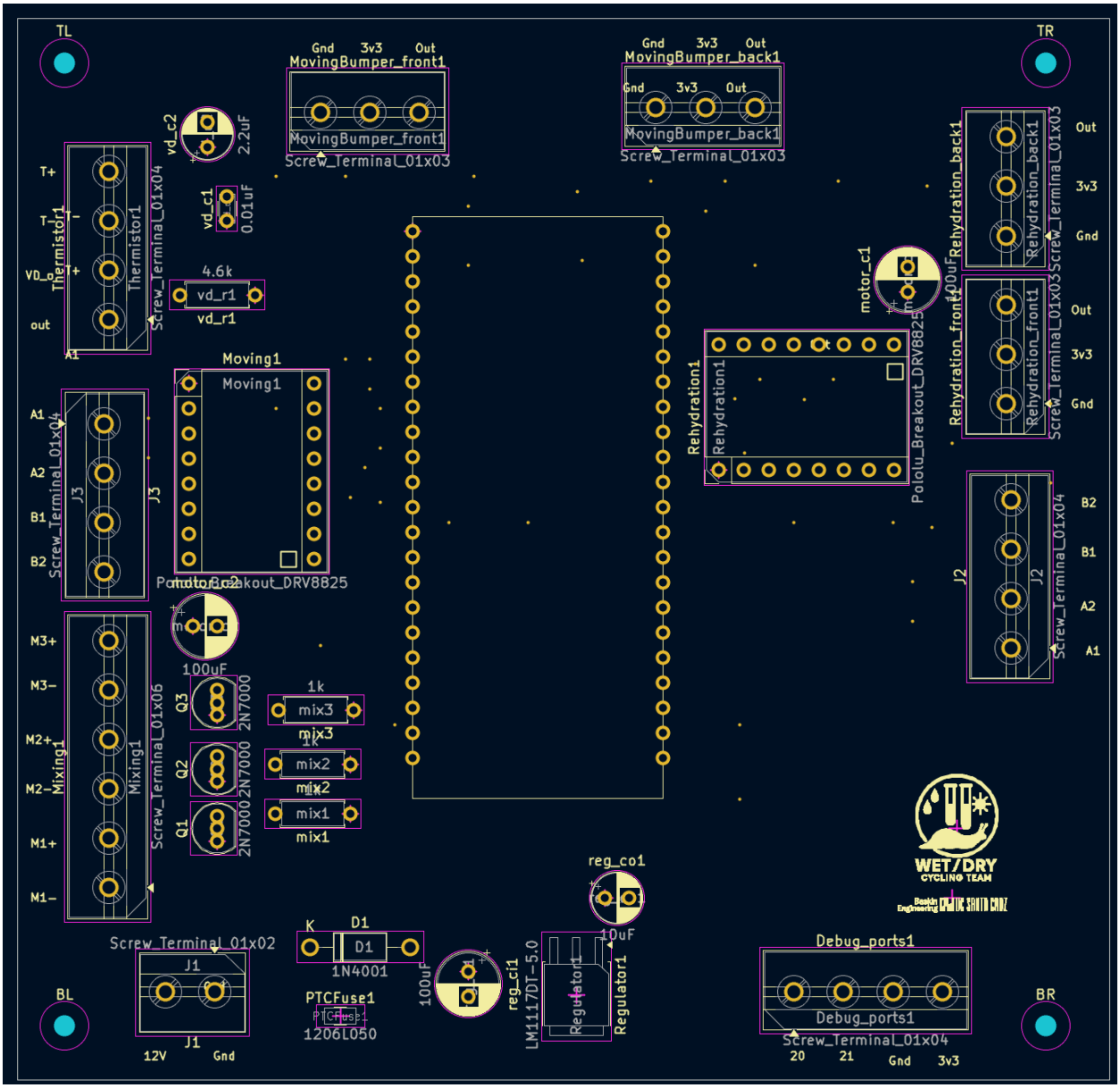


Figure 39: Silkscreen layer of designed PCB

Figure 37 shows the top layer of the designed PCB. This layer was used for routing all signal traces as well as distributing power through a copper pour. A 12 V plane was poured over the top layer to minimize voltage drop and reduce impedance in high-current paths, aligning with the engineering standards used. The components requiring 12 V were placed as close to the input as possible to minimize current paths. Component placement was based on their physical positions within the system rather than following a conventional

left-to-right signal flow. This component placement was done to simplify the system wiring. High-power components such as the DRV8825 motor drivers and LM7805 voltage regulator were placed to allow short power paths. Bypass capacitors were placed as close to their respective components as possible.

Figure 38 shows the bottom layer of the designed PCB. The bottom layer is implemented as a continuous ground plane. This plane provides a low-impedance return path for both signal and power currents. The ground plane reduces voltage noise and maintains stable reference potentials across the board. An unbroken ground plane minimizes parasitic inductance and ensures the shortest possible return current paths. It also prevents the occurrence of floating grounds, which can lead to differential voltage offsets and instability between subsystems. Additionally, the ground plane is used for via stitching to connect top-layer traces. The stitching of top layer traces was done while considering the flow of return currents to avoid introducing unintended return current loops.

Figure 39 shows the silkscreen layer of the designed PCB. The silkscreen serves to aid in the soldering and wiring of external components. Subsystems are grouped and named logically, and polarized components have proper orientation markings. Every screw terminal and it's input pins are also clearly labeled. Squares were added to the motor controller footprints to indicate potentiometer placement for easier tuning. The silkscreen aided in the construction and debugging of the PCB. For an image of the assembled PCB, see Figure 53 in the Appendix.

Signal and low-current traces were routed using a standard width of 20 mils, chosen as the largest possible trace width that still allowed for straightforward and efficient routing across the board. This width was verified to be sufficient by estimating the maximum expected current through signal lines and referencing IPC-2221 trace width guidelines, which confirmed that 20 mils can safely carry the anticipated currents without significant voltage drop or thermal rise [6]. Each subsystem connected to the PCB is mounted at a designated position within the mechanical structure. The following section discusses the physical layout

and mechanical design in detail.

16 Structural Housing and Frame – Ryan Taylor

16.1 Subsystem Overview

The Structural Housing and Frame mechanically integrates the extraction module, rehydration module, heating elements, electronics, and CO_2 delivery system referenced in the Block Diagram, shown in Figure 1.

In the Sample Manipulation Zone, the housing provides aligned mounting points for motors, limits, tubing, and syringe assemblies. These components require rigid geometry to ensure repeatable motion and fluid-handling performance.

While the housing is not sealed airtight, it provides semi-enclosed containment suitable for short-term CO_2 accumulation and environmental isolation. The design favors accessibility and modularity over full sealing, which aligns with the prebiotic synthesis use case.

The structural housing and frame subsystem was designed around four primary engineering constraints that governed all major decisions. First, structural stiffness was essential to ensure alignment consistency across repeated actuation cycles, minimizing the risk of performance drift due to mechanical deformation. Second, the system required a high degree of reconfigurability to support rapid prototyping, debugging, and iterative improvements throughout the development process. Third, although not fully sealed, the frame needed to provide a partial enclosure capable of retaining CO_2 and reducing the risk of environmental contamination—an important consideration given the biological sensitivity of the wet/dry cycling process. Finally, the layout had to remain accessible, allowing easy integration of tubing, electronics, and sensors without obstructing other components or complicating maintenance. These four constraints shaped the design philosophy of the entire subsystem.

The frame combines 2020 aluminum extrusion with 0.25-inch acrylic panels. Components were mounted using V-slot fasteners and custom brackets. Both materials were fabricated to

within ±0.1 mm tolerances—laser-cut for acrylic, and manually milled and caliper-verified for aluminum. This consistency preserved alignment across the system envelope.

Identical isometric angles are used for the CAD model Figure 41 and physical build Figure 40 to enable direct visual comparison.

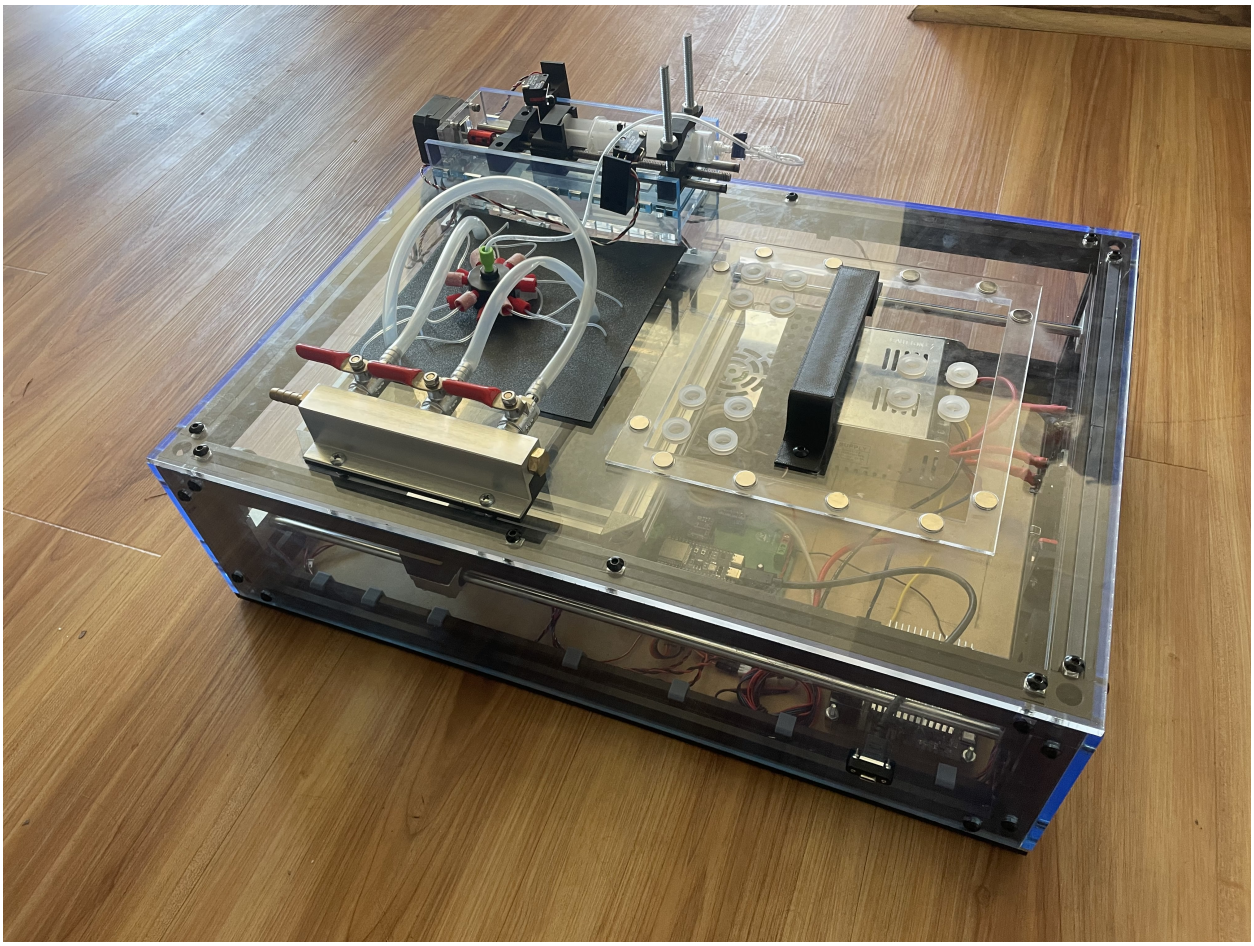


Figure 40: Physical build of the wet/dry Cycler.

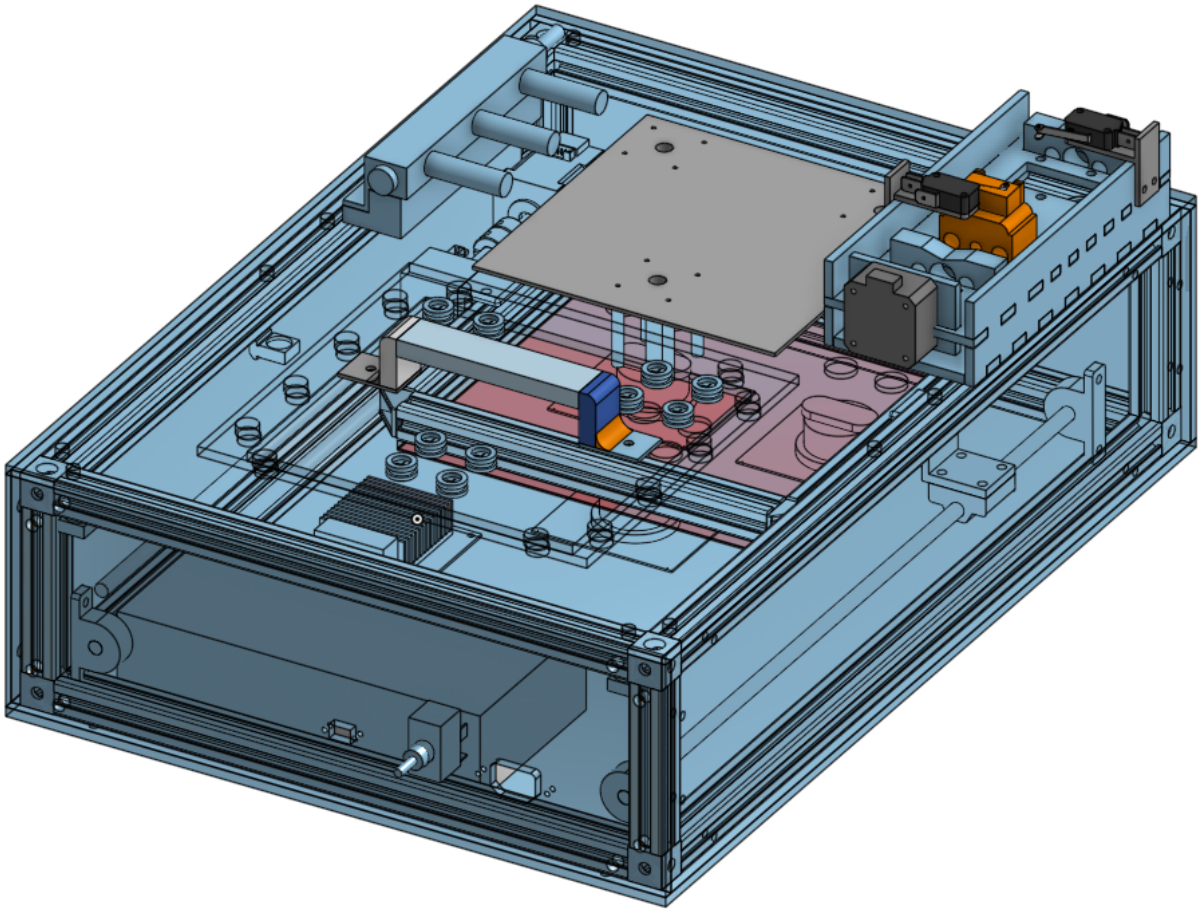


Figure 41: CAD model of wet/dry Cycler.

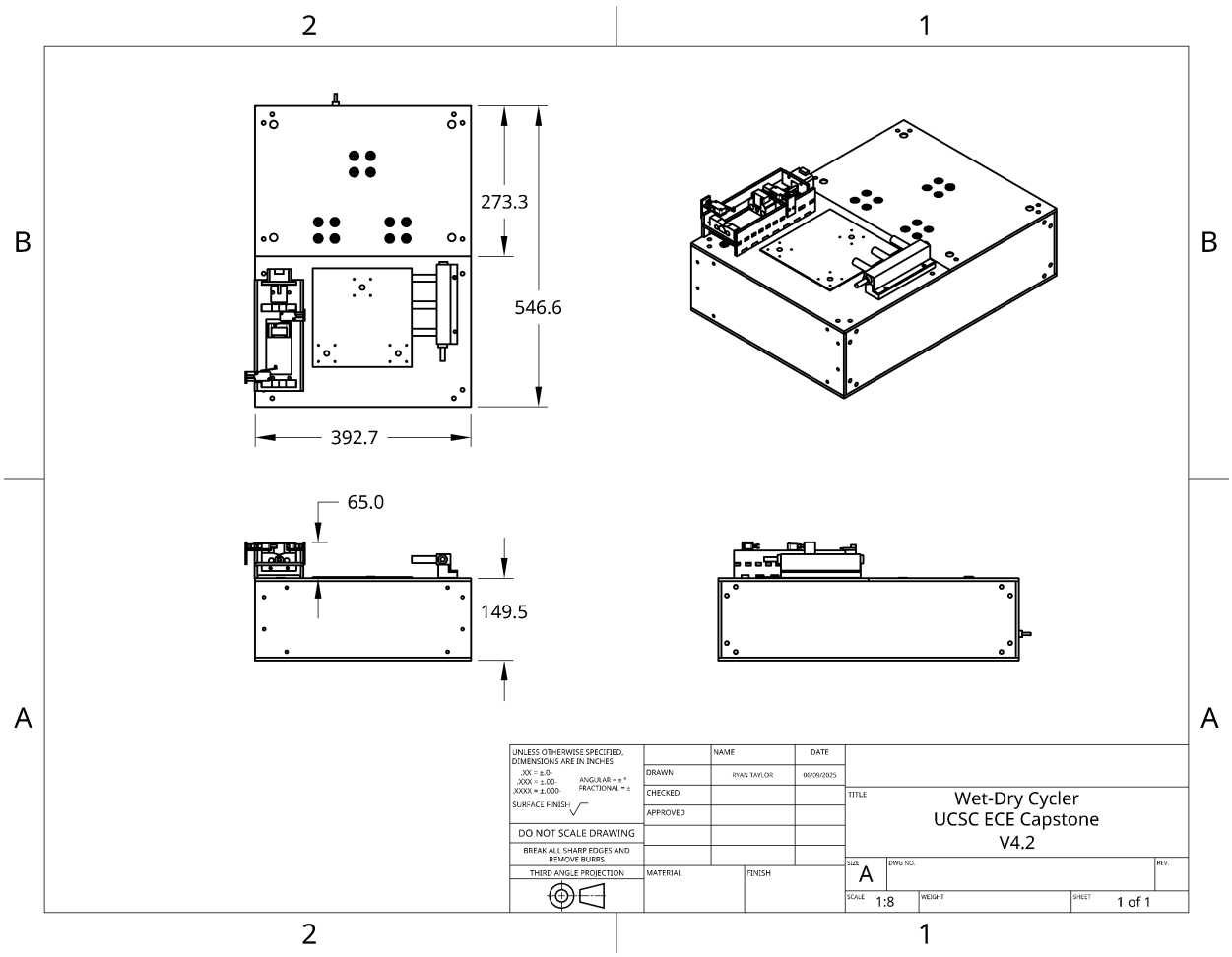


Figure 42: Dimensioned drawing of wet/dry Cycler housing with annotated panel and frame sizes (mm).

16.2 Material Selection – Acrylic Panels

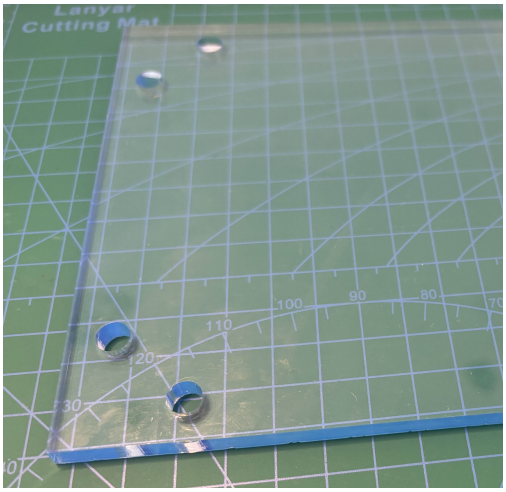


Figure 43: Clear acrylic panel used for housing.

Quarter-inch acrylic seen in Figure 43, was selected for the enclosure panels based on its transparency, rigidity, and compatibility with standard sterilization methods. Its thickness balances strength and is easily machined with a laser cutter, while staying lightweight. Clear acrylic helped with visual inspection during development and will help the researcher understand the state of the system while it is running.

Panels were laser cut to a nominal ± 0.1 mm tolerance. This precision allowed direct mating with slotted extrusion brackets without post-processing. While effective, future designs may shift to ABS panels for improved durability and heat resistance.

16.3 Material Selection – Aluminum Frame

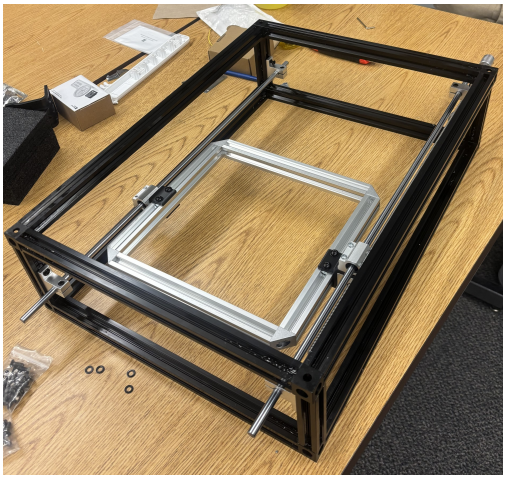


Figure 44: Aluminum Framing Connected through the cube connectors.

The aluminum frame seen in Figure 44, uses 2020 aluminum extrusion with V-slot profiles. This system offers a balance of rigidity, cost, and Versatility. Compared to 2040 profiles, 2020 was sufficient for expected loads defined in the load analysis(7) of the motor system , with easier routing in compact form factors.

Sliding nuts(55) and cube connectors(56) were used for fastening. The aluminum bars were cut and manually milled to ± 0.1 mm and verified with calipers. This ensured perpendicular joints and stable frame geometry.

16.4 Exploded View and Panel Integration

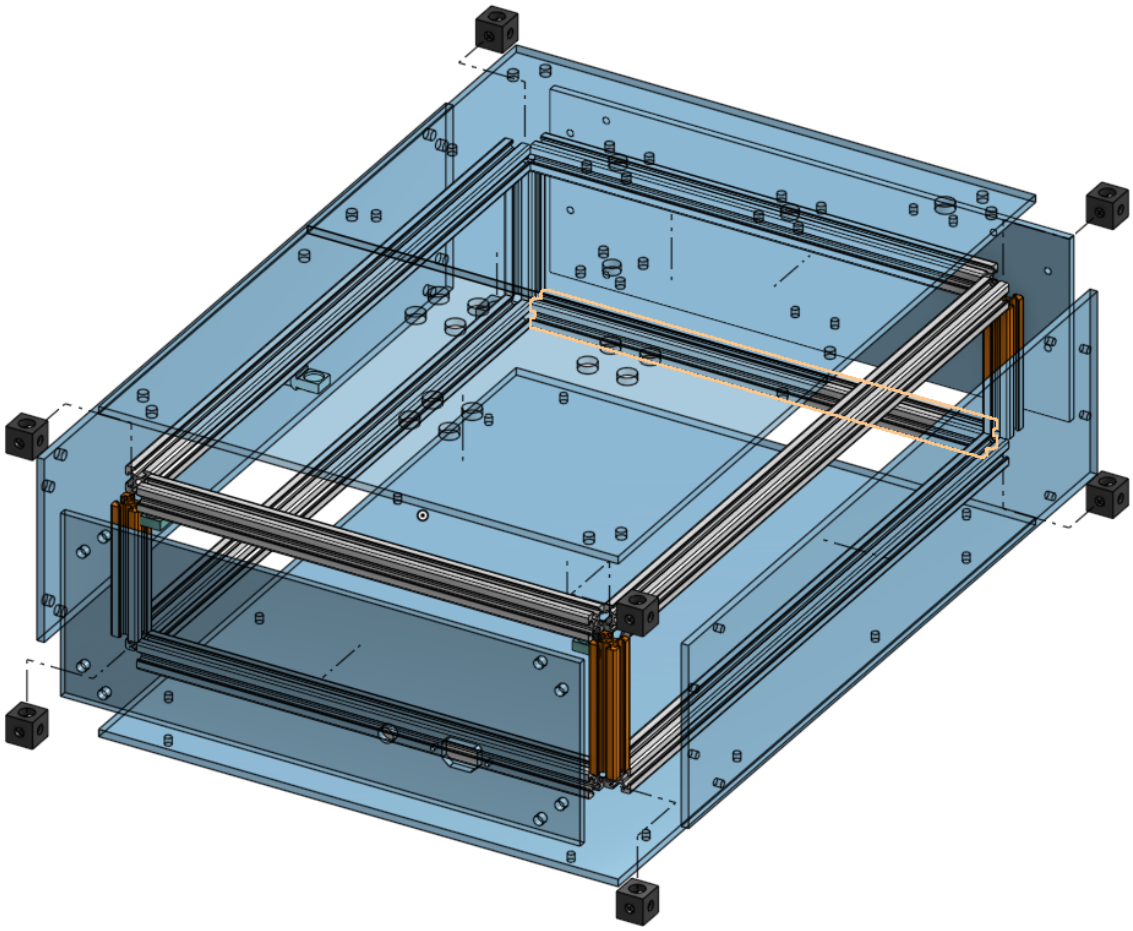


Figure 45: Exploded CAD view showing panel-frame interface and bracket positions.

Figure 45 shows an exploded CAD view of the panel-to-frame interface, detailing how panels are attached via corner brackets and M5 screws. This allowed rapid panel replacement or realignment without disassembling the entire frame.

16.5 Top Panel Access – Magnetic Lid System

Figure 46: Removable lid system, magnetically attached.

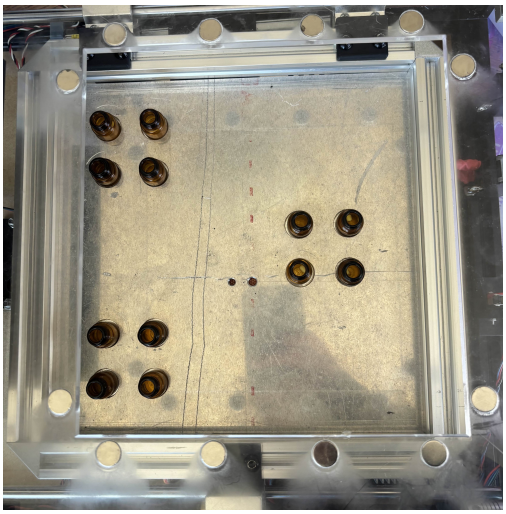


Figure 47: Removable lid system, open.

The top acrylic sheet is magnetically connected to the frame because it makes access to the vials in the extraction setup easier. When setting up the experiment this will reduce the time needed for setup of an experiment. However, there are also screw holes set up if the researcher decides it is best to keep the seal.

The magnets are 12mm in diameter by 2mm tall, these create a firm seal on that connects the parts together. These magnets fit into the engraved acrylic Figure 46. The magnets are

then super-glued for security. The corresponding top acrylic plate has engraved circles that mirror these magnet holders, holding the 2 planes together. There is then a 3D printed handle that allows for removability.

16.6 Stepper Motor and limit Mounting

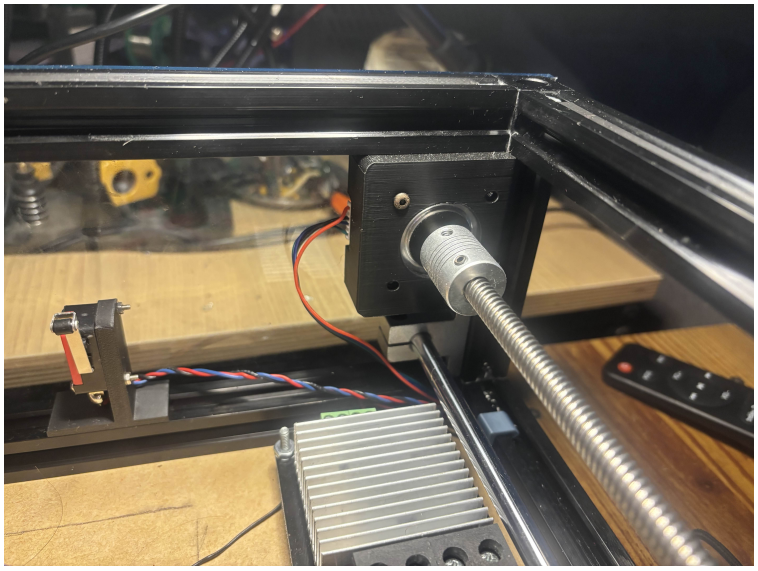


Figure 48: Stepper motor mount bracket.

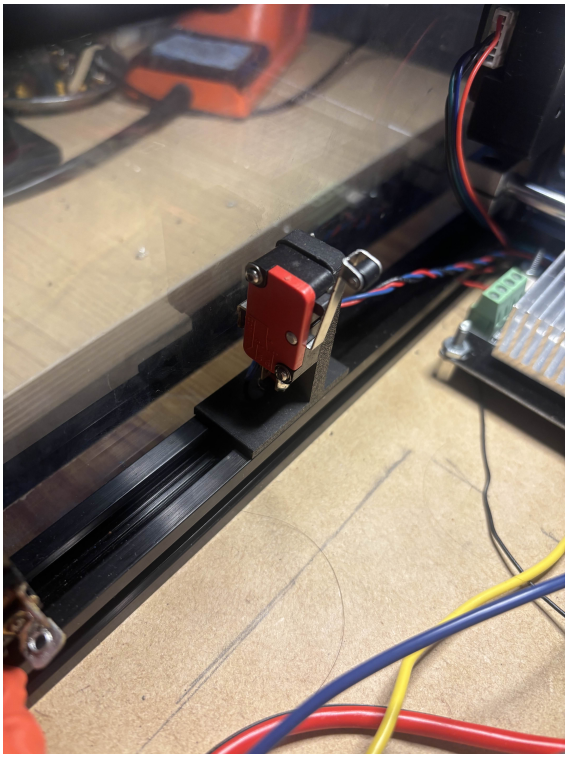


Figure 49: Limit bumper mounted to the aluminum frame.

The stepper motors shown in Figure 48 and the limit switches shown in Figure 49 were secured to the aluminum rails with a custom 3D-printed bracket that slots into the 2020. The limit switches sit on the horizontal rails on both sides of the device to detect the position of the plate.

16.7 Syringe Motor Mount

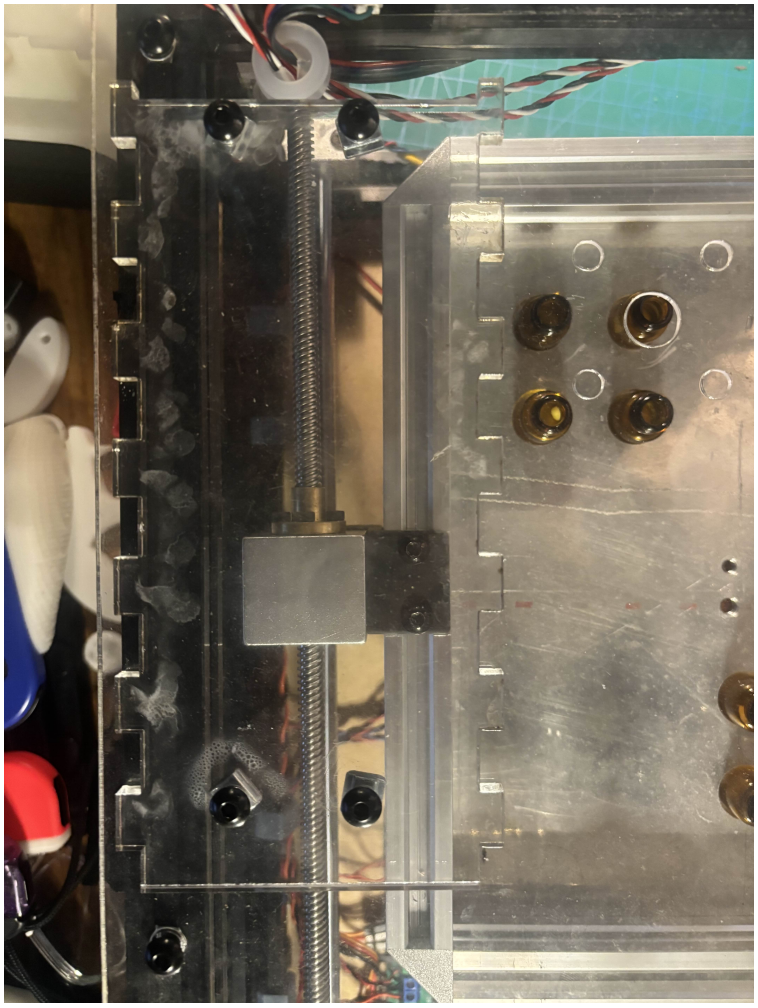


Figure 50: Syringe motor mounted onto top acrylic panel with sealed wire pass-through.

The syringe motor was mounted to the top panel using self-tapping screws. Using the same Silicon hole plugs as in the extraction Figure 23 provided a semi-sealed pass-through for wiring, maintaining basic CO_2 isolation. Initial Designs involved engraving a cut out in the acrylic for the syringe to sit in, however this was not solid enough therefore the bottom piece was designed in order to mount the syringe to connect to the overall system

16.8 Power Input Design

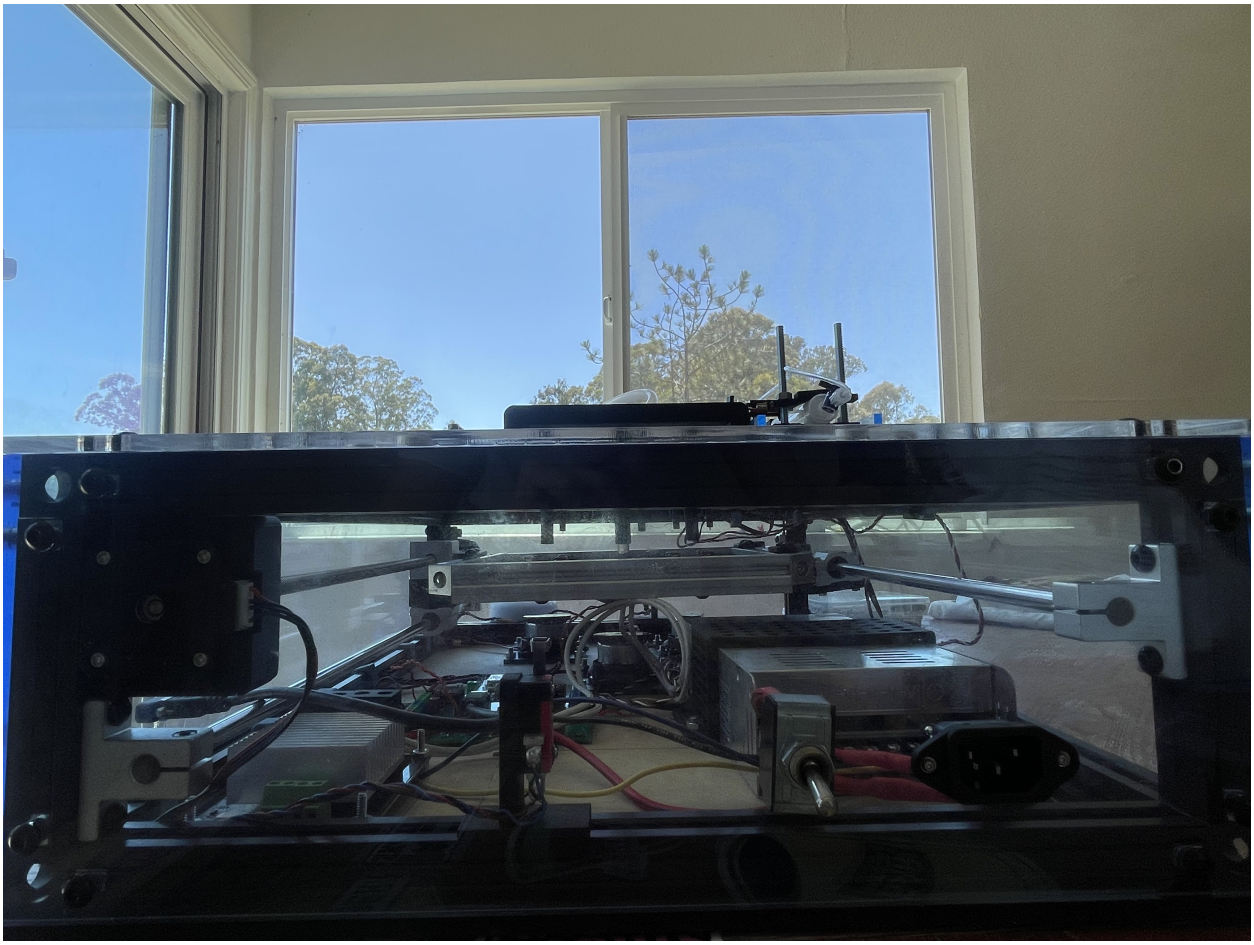
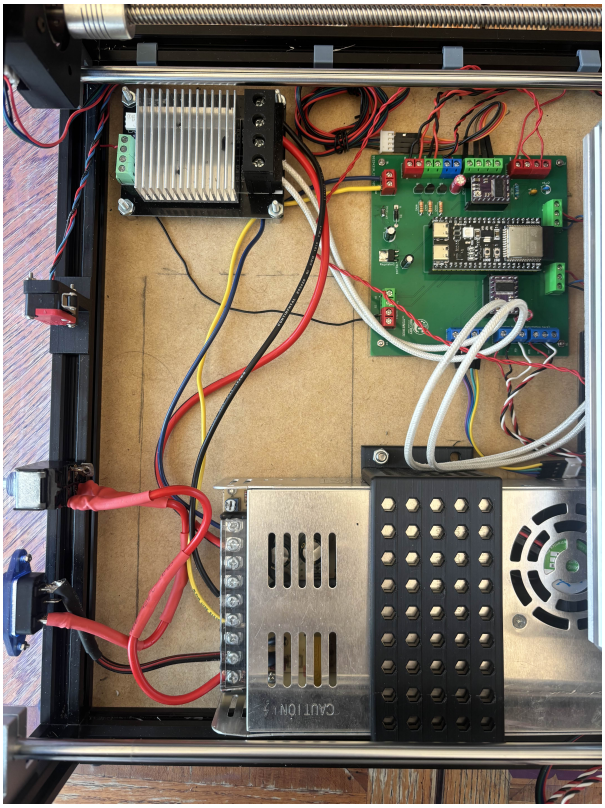
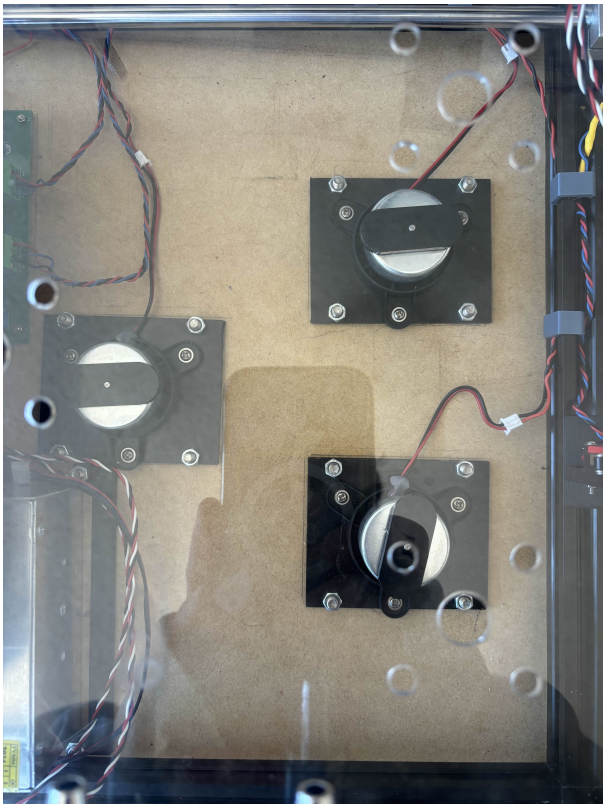


Figure 51: Front profile of the wet/dry Cycler, including power switch and indicator lights.

Power was routed through an Inlet Power Plug Socket, shown in Figure 58 and Figure 57 and toggle switch on the front panel. The inlet power plug socket is rated for 250V AC/10A, which means a maximum power supply rating of 2500 W. At 120V AC the power socket can output 21 A to the power supply. This is more than the expected output of 360 W, according to out power budget in Appendix A.



(a) Power supply, MOSFET, and PCB.



(b) DC mixing motors.

Figure 52: MDF baseplate assembly with electronics (left) and motors (right).

16.9 Baseplate Mounting and Wiring

Electronics were placed on the MDF baseplate and drilled mounting holes for to make the wiring neater. The DC mixing motors were placed directly under the center of the vial grouping they are associated with. Then the magnetic field test addressed in Figure 16 was conducted to verify that the magnetic field reaches the magnets inside the vials.

16.10 Conclusion and Reflections

The housing met core performance criteria: stiffness, reconfigurability, and integration. The frame ensured placement of hardware, and the acrylic panels provided lightweight enclosures without compromising usability.

Post-build observations revealed several areas for improvement. The syringe motor

mounting introduced slight deflection in the surrounding acrylic, suggesting that future iterations would benefit from the addition of a 3D-printed support bracket or guide rail to reinforce this region. Additionally, the CO_2 tubing routing was suboptimal. The original layout did not account for the fact that sharp bends in the tubing could lead to pinching and restricted gas flow. As a result, the current routing paths are more convoluted than necessary, weaving around components to avoid tight radii. In future designs, a more deliberate approach to tubing layout—potentially with dedicated routing channels or shielding—could significantly improve efficiency and clarity.

17 Conclusion & Future Applications

The Wet/Dry Cycler successfully automates prebiotic simulation experiments by integrating heating, fluid handling, gas control, and mechanical actuation into a modular system. The system supports control of experimental parameters and provides a user-friendly interface for researchers. Engineering decisions were guided by project deliverables established at the beginning of the project and informed by relevant engineering standards. The platform performs as intended and will be transferred to a PhD researcher for continued experimental use. The design reduces cost and increases throughput for RNA synthesis research, offering a practical tool for advancing origin-of-life studies.

RNA, due to many functions beyond coding genetic information, has numerous therapeutic applications—including *in vivo* genetic modifications (e.g., CRISPR-Cas9) and alteration of cellular RNA expression using short interfering RNA. A secondary function of the instrument is to address the high cost of therapeutic RNA molecules, which are currently synthesized enzymatically using expensive RNA polymerases. This process can cost upwards of $120,000, therefore this research can significantly reduce this cost because it strives to make RNA with just water, heat and chemical bases.

References

[1] David Deamer, Google Scholar Profile, https://scholar.google.com/citations?user=2RoLRE8AAAAJ&hl=en

[2] David Deamer, *Hot Springs and the Origin of Life*, Astrobiology, 2020, https://www.liebertpub.com/doi/full/10.1089/ast.2019.2045

[3] David Deamer, *Wet-Dry Cycling in Prebiotic Chemistry: A Critical Review*, Proceedings of the National Academy of Sciences, 2024, https://www.pnas.org/doi/abs/10.1073/pnas.2412784121

[4] ANSI. *Dimensioning and Tolerancing (ANSI Y14.5-2009)*. American National Standards Institute, 2009.

[5] IEEE. *IEEE Standard for Software and System Test Documentation (IEEE Std 829-2008)*. IEEE, 2008.

[6] IPC. *Generic Standard on Printed Board Design (IPC-2221A)*. IPC, 2003.

[7] USA Wire & Cable, *NEC Ampacity Guidelines*, based on National Electrical Code (NEC), https://usawire-cable.com/wp-content/uploads/nec-ampacities.pdf

[8] Ralph E. Locher, *Introduction to Power Supplies, AN-556*, National Semiconductor, November 1988.

[9] S.C. Petersen, T. Favaloro, *Technical Documentation Standards in Engineering Design*, Rev. 1.2, University of California, Santa Cruz, ECE Department.

[10] S.C. Petersen, *Standards of Good Engineering Practice for PCB Design*, Rev 3.0.

[11] S.C. Petersen, *Power Distribution, PCB Layout and Part Placement Note*, Rev 1.0.

18 Appendix

A Power Budget Spreadsheet

Power Budget Spreadsheet.

B GitHub Repository

Wet-Dry-Cycler GitHub Repository.

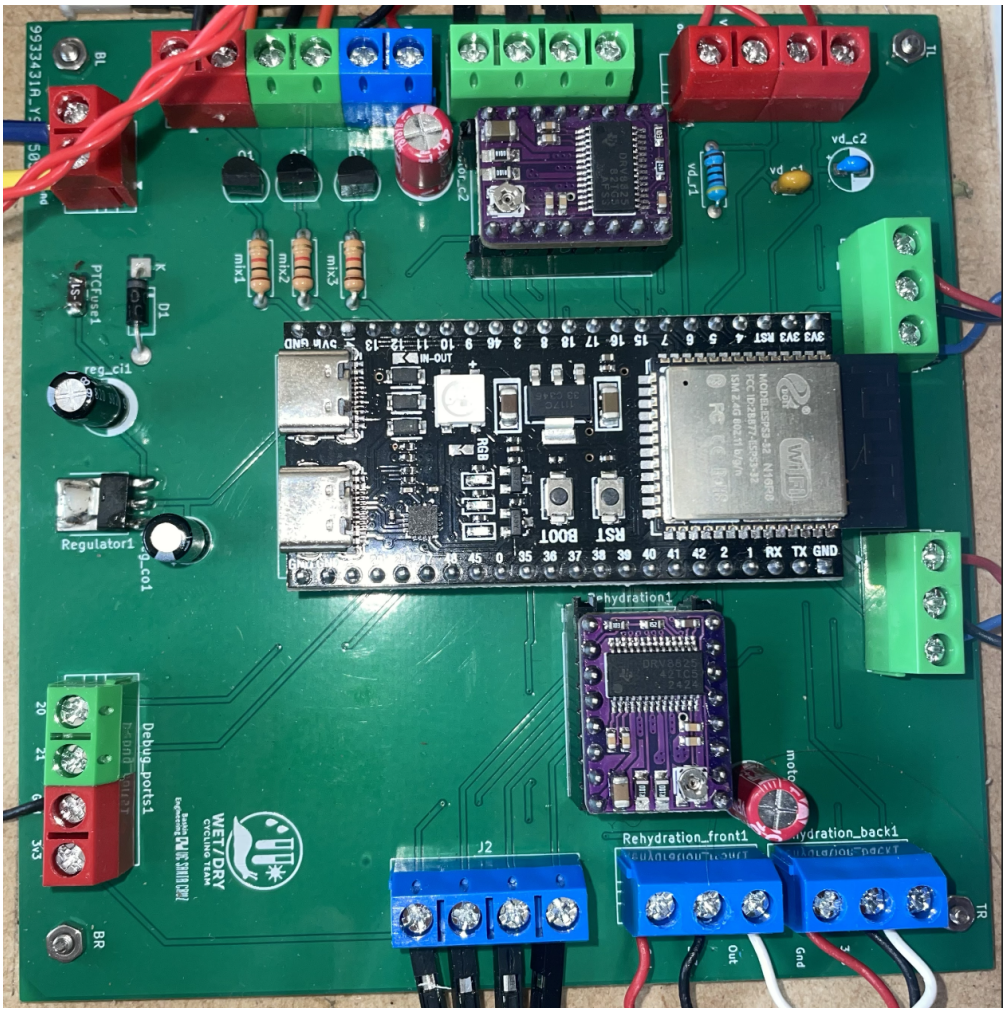


Figure 53: Photo of the assembled PCB

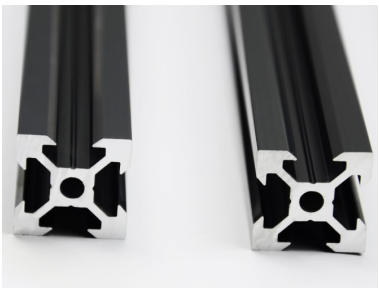


Figure 54: 2020 aluminum extrusion featuring a standard V-slot profile.

Figure 55: Sliding nut compatible with the V-slot, used for adjustable fastener placement.

Figure 56: Cube connector for joining aluminum beams at 90° angles.

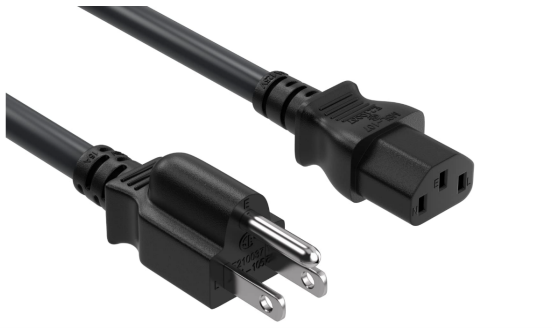


Figure 57: Panel-mounted male barrel jack connector for external power supply.

Figure 58: Matching female barrel jack plug for wall adapter.