# New Generation Addiction Control and Recovery Device System with Absolute Safety and Privacy

By

Adrian Santosh

Richawn Bernard

Leo Li

Final Report for ECE 445, Senior Design, Fall 2025

Professor: Cunjiang Yu

TA: Shenyang Liu

10 December 2025

Project No. 12

## Abstract

Our project aims at making a privacy-preserving, closed-loop nicotine-reduction device that senses user state and automatically titrates dose to taper dependence. The ESP32 control subsystem mixes nicotine, diluent, and an aversive additive via peristaltic pumps, adapting composition in real time while heart-rate inputs and fingerprint gating prevent misuse. Target performance is ±5% concentration accuracy with ~1 s control updates and mandatory inter-dose lockouts; safety monitors (leak/temperature/battery/sensor faults and abnormal vitals) trigger audible/visual alerts and software shutdown. Detailed logs are stored for temperature, liquid weights, and the actual mixture concentration at each run for debugging and to maintain healthcare records. To protect users, the system operates offline with SD-card data storage, and detailed histories require authentication. The entire system has been tested thoroughly, and all components meet the design requirements.

# Contents

# 1. Introduction

## 1.1 Problem

"First you take a drink, then the drink takes a drink, then the drink takes you." – F. Scott Fitzgerald. By "unconscious," I mean habits that sneak in slowly, built by small daily choices and cues, so people do not notice the slide until it is strong. Naming the harm is not enough; we need a clear plan to stop the slide. "It does not matter how slowly you go as long as you do not stop." – Confucius. The point is that steady effort matters more than speed, but pushes from stress and daily life keep the rock moving. Two significant gaps stand out in real life. The base logic for addiction recovery is usually step-by-step. It works like trying to reshape a Bouncy Ball. If we try to push too hard, it will cause a huge bounce back and eliminate most of the distance we have already covered.

All the current methods rely heavily on the self-regulation ability of the addict. This assumes strong willpower every hour, which is not realistic for most people. Life is messy; work shifts, kids, money, and stress can break the routine. Nicotine fades fast, so that a strong desire can spike many times a day. People end up guessing dose and timing, which causes under-dosing (white-knuckle cravings) or over-dosing (nausea, dizziness). In simple words, a system that senses, decides, and helps right when the urge hits. It should be easy to use, low-cost, and private. Today's tools are one-size-fits-all, slow to react, and too hard to follow under stress. We need a responsive loop: notice the trigger, then provide the right aid, check the effect, adjust, and do it again in time.

## 1.2 Solution

The solution is a smart nicotine reduction device that monitors and adjusts the amount of nicotine a user inhales with each use. This system will incorporate a programmable control unit that gradually decreases nicotine concentration over time. By lowering the dosage in careful steps, the device reduces the dependency and the "bounce back" effect of sudden withdrawal. To further discourage continued use, the system can introduce a safe bitterant, conditioning the brain to associate vaping with reduced satisfaction. This combination of gradual reduction and negative reinforcement helps break both physical cravings and the psychological appeal. The implementation centers on a PCB that controls the vape's motor and mixing components. The PCB adjusts the composition of the vapor in real time by mixing nicotine with a bitter compound and a dilutant such as water. The device will progressively lower its nicotine levels and increase that of the bitterant, allowing the users to adapt without severe withdrawal symptoms. The device checks the user's health status from their pulse rate and authenticates the user via fingerprint verification. On any safety violation, the alarm system is activated, and the device shuts down till the violation is resolved.

## 2. Design

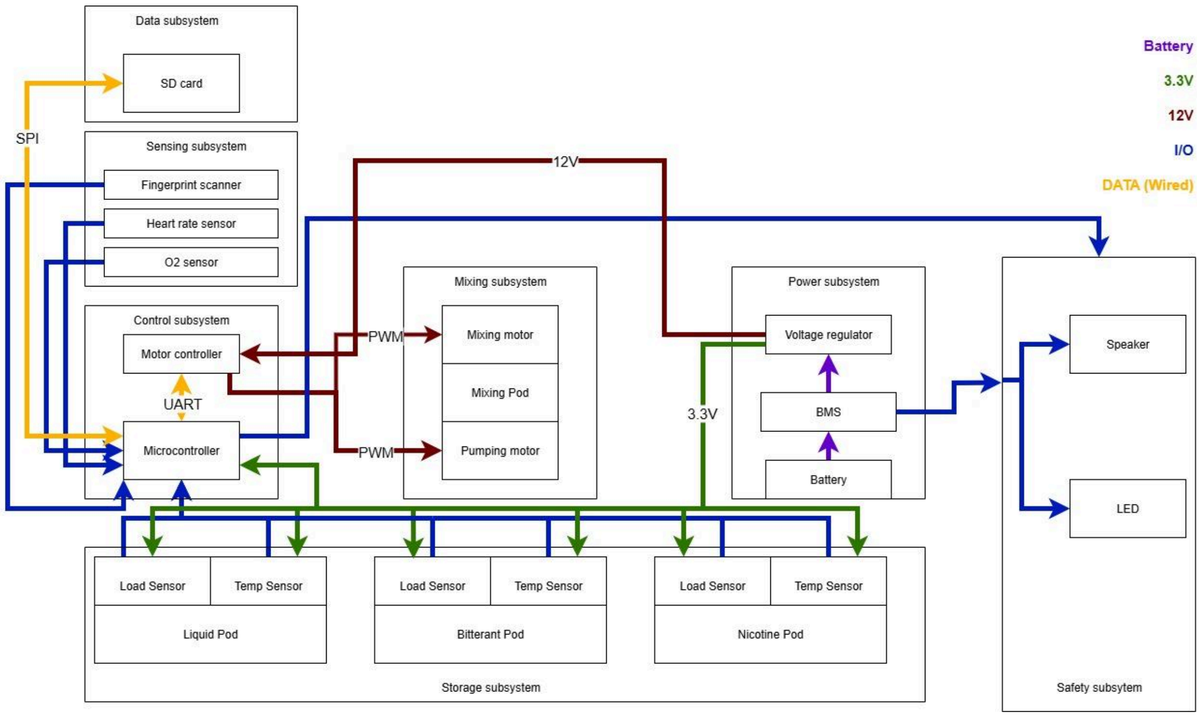


**Fig 1. System block diagram**

The system is divided into several subsystems, as shown in Fig. 1, that work together to ensure proper device functionality and safety. The Device starts by checking the user's status using the sensing subsystem. The fingerprint sensor verifies that the user has access to the Device. Next, the pulse sensor records the user's pulse to ensure it is within an acceptable range. The liquids are stored in the storage subsystem. It consists of nicotine, a dilutant, and a bitterant, which are mixed by the mixing subsystem into the mixing pod. After checking the user status, the load sensors and temperature sensors are polled to ensure that the integrity of the pods is not compromised. Once the pod health has been checked, the mixing subsystem is turned on. The pumps transport the liquid from the storage pods into the mixing pods according to the concentration set by the ESP32 microcontroller unit. The MCU controls the motor speed using the motor control circuit to prepare the mixture according to the correct concentration.
After the mixing pod is filled, the mixing motor spins the mixture to ensure that it is smooth and homogenous. Next, the temperature and weights of the liquid pods are checked to verify that the desired amount has been pumped into the mixing pod and that the temperature remains stable. The safety system provides alerts to the user at the device start, during the start and end of pumping, and during device shutdown. During device operation, regular logs from sensors, motors, and voltage regulators are written to an SD card in the storage subsystem for debugging and record-keeping. The entire system is powered by the power subsystem, which supplies 3.3V to the MCU and sensors and 12V to the motors. The power subsystem uses buck converters to convert the 12V battery output to the desired voltage levels and maintain current stability. We split the PCB design into three boards: the microcontroller PCB, which houses the ESP32 and peripherals; the Sensor and motor PCB, which houses the motor and sensors; and the power

PCB, which houses the power supply unit. The parts used in the system are given in Table 1 in Section 4.
The high-level requirements for each subsystem are as follows:

1. Sensor:
    a. The Sensor should function adequately with the microcontroller and receive immediate commands.
    b. The Sensor should be able to report results within a 5% tolerance when dealing with data.
    c. The physical setup for the sensors should be arranged in a reasonable way to avoid unexpected movement in space or unexpected detection of materials not considered in the design.
2. Control:
    a. The ESP32 Micro Controller should be working properly, with the certainty that the pins and power are set up correctly. Extreme cases should be considered and handled.
    b. The safety signal should be double-handled in the Control Module, with additional hardware to ensure proper operation.
    c. The Device calculates the requested content in 1s.
    d. When receiving input with extremely abnormal combinations, the Analysis should handle all cases not already handled by the other modules as a last resort.
3. Mixing:
    a. The Control Signals for the motor are correctly sent to the Mixing System and correctly handled.
    b. The motor device is working correctly with the correct power supply, operating frequency, and power.
    c. The system delivers the commanded dose concentration (using a safe test liquid) within 5% of the target, as measured on a scale, and enforces a minimum lockout time between doses.
4. Safety:
    a. All Safety Systems in each module are correctly connected and sending signals to the Safety signal handler at the correct time.
    b. If a red-flag condition occurs (very high heart rate, high oxygen level, sensor fault, low battery, liquid leak, high liquid temperature), a clear alert appears. The buzzer should ring, and the LED should light up.
5. Privacy:
    a. The Device should not present detailed information about past usage without a password or safety check.
    b. The Device works with no external network. All data is stored on an SD card, and the user can erase all local data from it.

## 2.1 Sensor Subsystem

The sensor subsystem receives real-time user data, such as heart rate. It also has a fingerprint sensor to capture and read fingerprint data for the device's secure operation and to prevent misuse. These signals feed into the microcontroller, which uses them to adapt the dosage and detect unsafe conditions. The sensors are soldered to the sensor PCB, which has ports for interacting with the microcontroller. The schematic and layout of the sensor PCB are shown in Fig. 2 and Fig. 3, respectively.

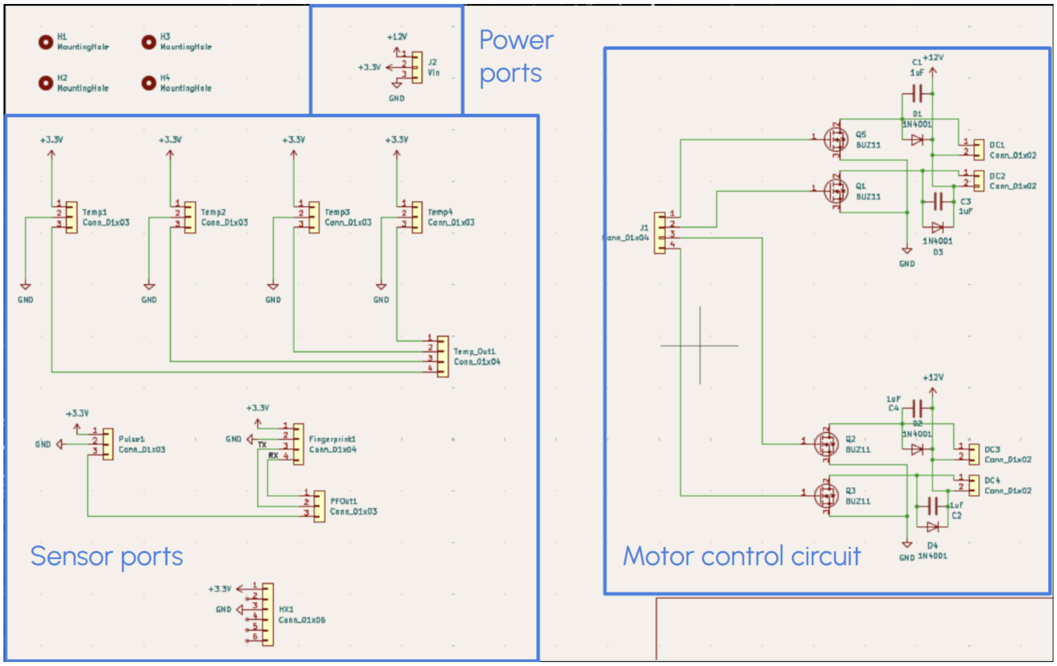


**Fig 2. Schematic of sensor and motor PCB**

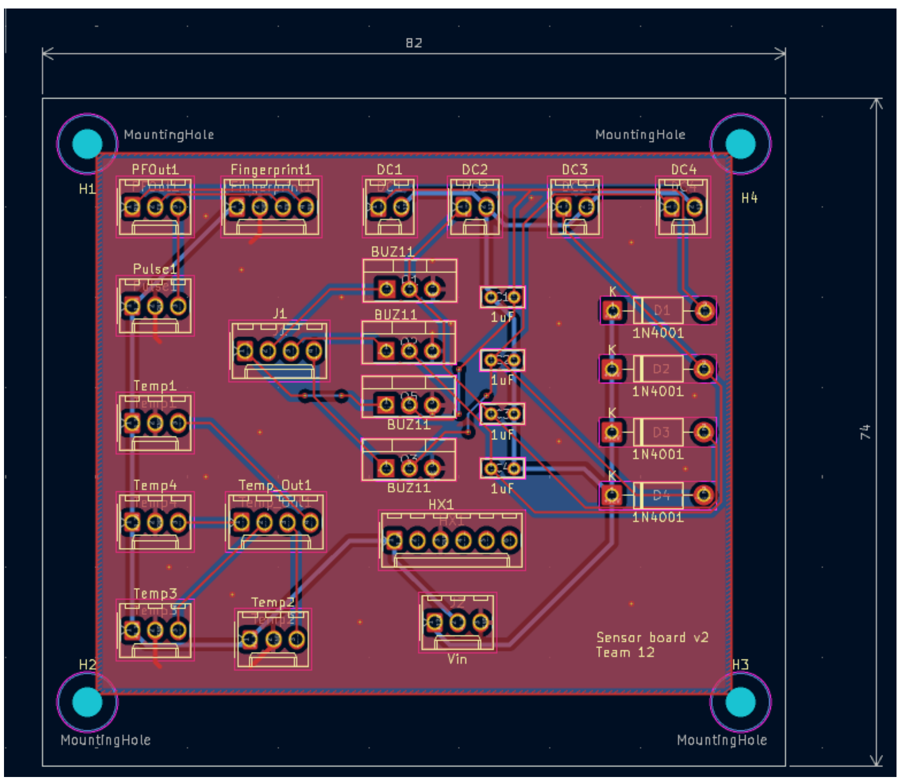


**Fig 3. Layout of sensor and motor PCB**

## 2.2 Control subsystem

The ESP32-S3-WROOM-1-N4 serves as the central processor, receiving data, implementing safety logic, and controlling the dosing concentration. It enforces system timing, gradual nicotine reduction, and overrides abnormal commands. It communicates with the mixing and tank systems to ensure safety cutoffs are maintained. The motor controller controls the pump and mixer to transport fluid from the tanks to the mixing pod and to maintain the liquid concentration in the mixing pod. The formula for controlling the pump timings is given in Equations (1), (2), and (3). The ESP32 and the programming circuit are on the microcontroller PCB, along with the ESP32's programming circuit. The schematic and layout of the microcontroller PCB are shown in Fig. 4 and Fig. 5, respectively.

$$\text{Nicotine pump time(s)} = \frac{\text{Nicotine concentration} * \text{total volume}}{\text{Nicotine pump speed (mL/s)}} \tag{1}$$

$$\text{Bitterant pump time(s)} = \frac{\text{Bitterant concentration} * \text{total volume}}{\text{Bitterant pump speed (mL/s)}} \tag{2}$$

$$\text{Dilutant pump time(s)} = \frac{\text{Dilutant concentration} * \text{total volume}}{\text{Dilutant pump speed (mL/s)}} \tag{3}$$

The speed of each motor is recorded by transporting a known mass of liquid and measuring the time taken.

An obstacle we had to overcome was correcting the driver code for some devices. Sometimes the code was correct, but the device was wrong, and sometimes the code was just outdated. Also, most of the time, the code was for an Arduino, not an ESP32.

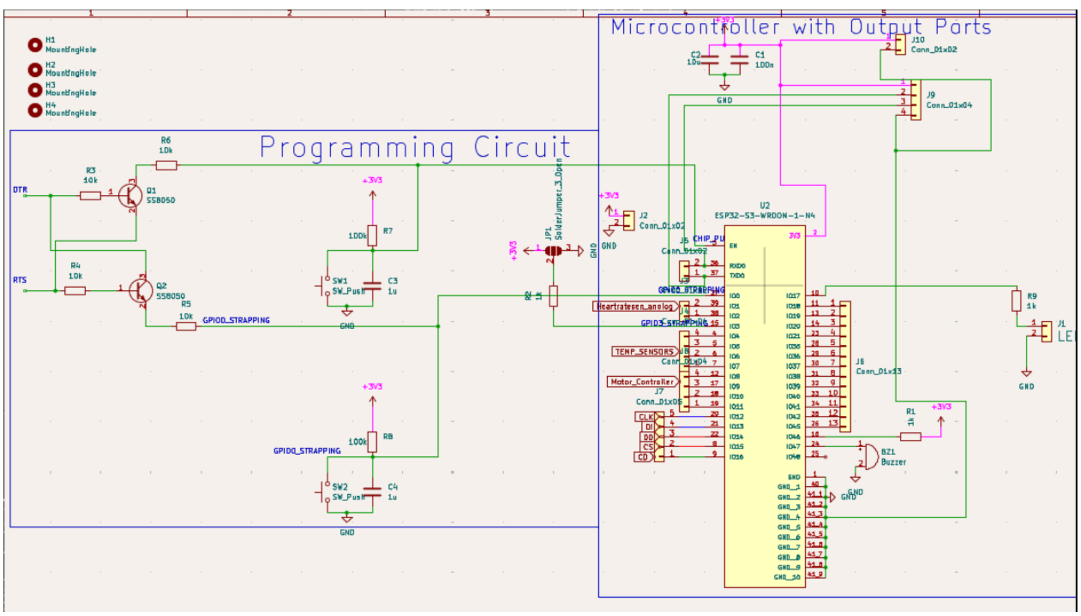


**Fig 4. Schematic diagram for the microcontroller PCB**

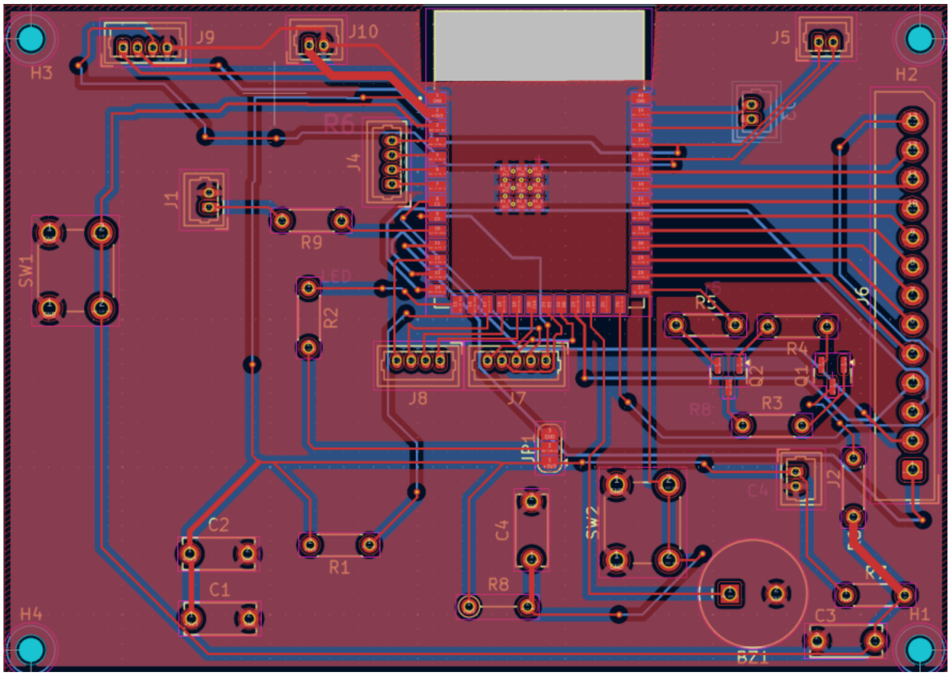


**Fig 5. Layout for the microcontroller PCB**

## 2.3 Mixing Subsystem

The mixing subsystem controls the fluid reservoirs in the storage subsystem. Each reservoir feeds fluid into the Mixing pod through a peristaltic pump. The mixing pod incorporates a stir mechanism driven by a DC motor that mixes the fluid to create a homogeneous solution prior to vaporization. This subsystem serves as a central point for the dose control and interacts directly with the control subsystem. The motor terminals are soldered to the motor PCB and connected to motor control circuits that turn the motor on and control its speed in response to signals from the control subsystem. The schematic and layout of the motor PCB are shown in Fig. 4 and Fig. 5, respectively.

The motor control circuit consists of a PWM signal from the microcontroller, a transistor, a diode, and a capacitor. Each of the four motors has its own motor control circuit. The source of the transistor is connected to ground, and the drain of the transistor is connected to the negative terminal of the motor. The positive end of the motor is connected to power. The microcontroller's PWM signal controls the transistor's gate. The PWM signal switches on the transistor, completing the motor circuit by connecting the motor's negative terminal to ground. The motor speed is controlled by the width of the PWM signal, which is set by the firmware running on the microcontroller. A diode is connected across the motor terminals to dissipate reverse current when the motor is turned off. We also connected a capacitor across the motor terminals to smooth voltage fluctuations caused by the motor's back EMF.

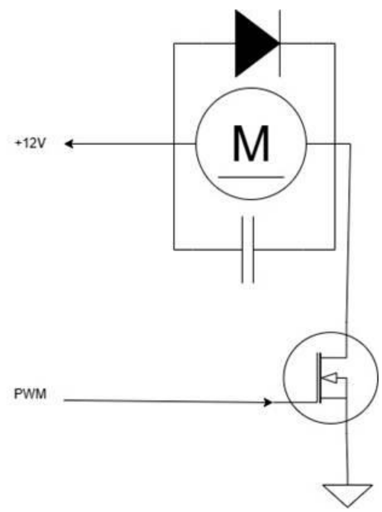


**Fig 6. Motor control circuit**

The average voltage applied to a motor via PWM is proportional to the duty cycle, and motor speed is approximately proportional to that average voltage as given in Equation (4).

For a PWM signal with a fixed frequency and variable pulse width:

$$V_{\text{avg}} = D \cdot V_{\text{peak}} \tag{4}$$

Where Vavg is the average voltage applied to the motor, D is the duty cycle (expressed as a decimal, e.g., 0.6 for 60%), and Vpeak is the peak voltage of the PWM signal (typically supply voltage)

This equation assumes the motor's inductance smooths the PWM waveform into a usable average voltage.

### 2.3.1 Motor Speed Approximation

For DC motors, speed is roughly proportional to the average voltage as shown in Equation (5),

$$\omega \approx k \cdot V_{\text{avg}} = k \cdot D \cdot V_{\text{peak}} \tag{5}$$

Where ω is the angular speed (rad/s or RPM), k is the motor speed constant (depends on motor design)

This linear relationship holds under light load and steady-state conditions. Under heavy load or with back-EMF, the actual speed may deviate.

### 2.3.3 Frequency Considerations

PWM frequency affects motor performance, but not the average voltage. A higher frequency provides smoother current, less audible noise, and better control, whereas a lower frequency produces more ripple and potential torque pulsations. Typical PWM frequencies for motor control range from 5 kHz to 25 kHz.

### 2.3.4 Example

If you apply a 12 V PWM signal with a 75% duty cycle $V_{\text{avg}} = 0.75 \cdot 12 = 9\text{ V}$. Assuming k = 100 RPM/V , the motor speed would be $\omega = 100 \cdot 9 = 900\text{ RPM}$

## 2.4 Power subsystem

The power system supplies regulated DC voltage (3.3V, 12V, and 5V, if needed) to sensors, motors, and the ESP32 using regular batteries. The power system should also include an emergency backup system in case one of the batteries is not working due to a fault or being out of charge. In this case, when the switch is on, the power subsystem should directly use the 12V battery to supply everything through a voltage adaptor. The power subsystem is on a separate PCB and is connected to the motor PCB and the microcontroller PCB through power ports. The layout and schematic of the power PCB are shown in Figs. 5 and 6, respectively.

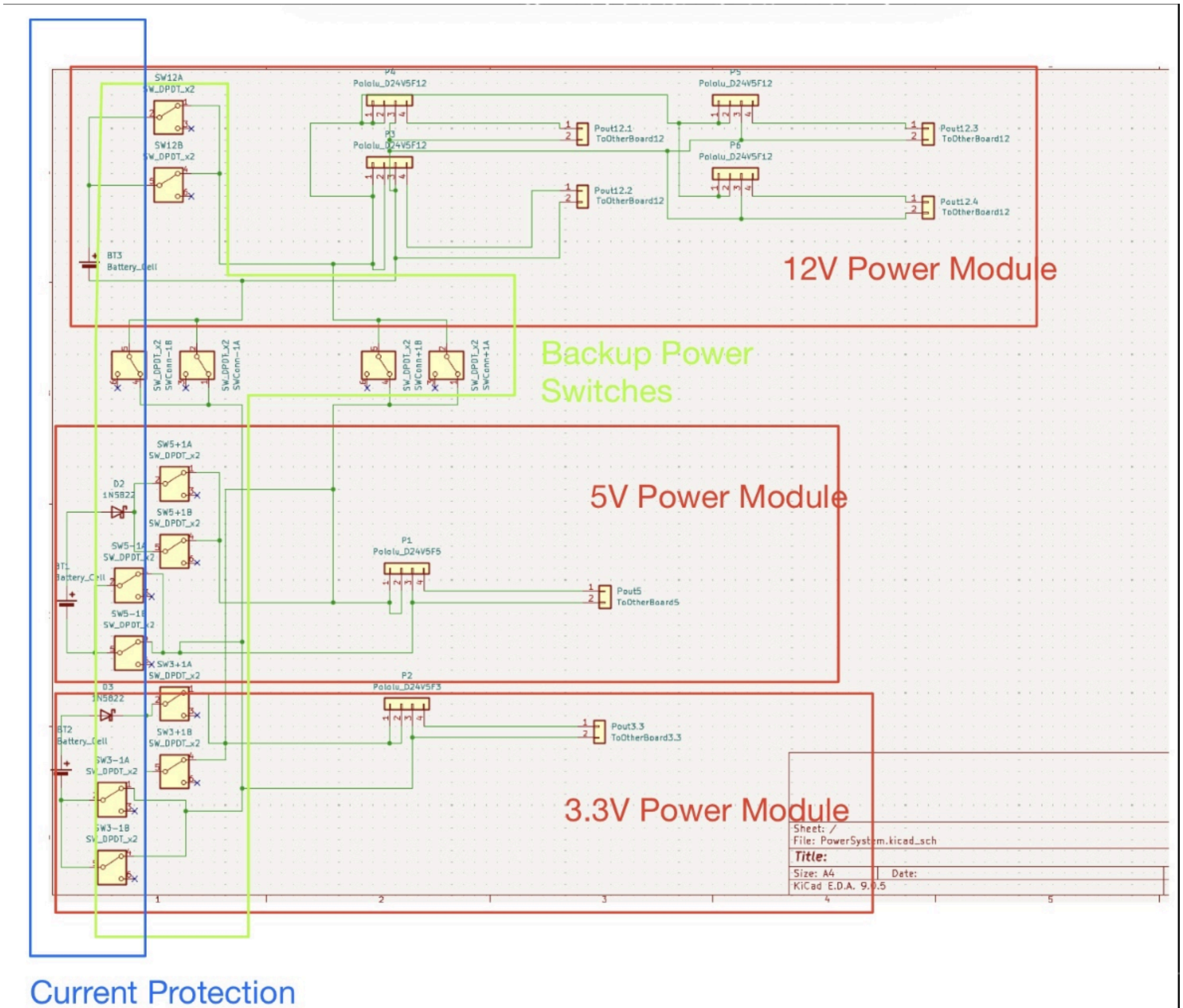


**Fig 7. Schematic of the power PCB**

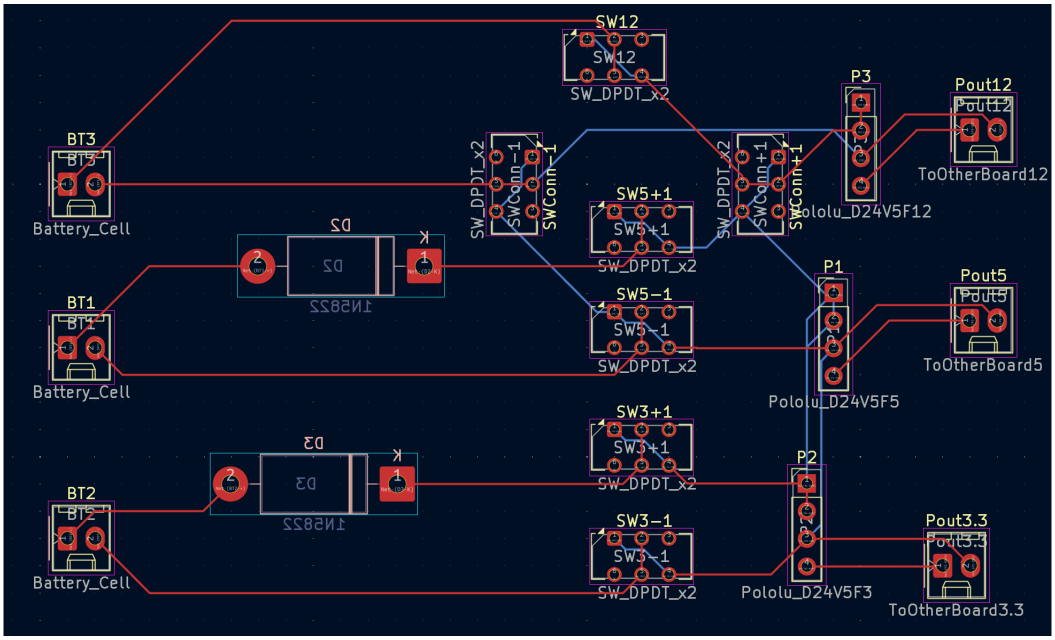


**Fig 8. Layout of the power PCB**

The biggest obstacle in designing the power subsystem was the failed attempt to use a linear regulator. The design process for the version using a linear regulator took more than 4 weeks and ended with total failure, so the design had to be switched to a buck regulator.

The amount of power being wasted as heat is roughly (Vin − Vout) × Iload = 11.7 V × I_load. At 500mA, it is 5.85W, a bunch of heat. Moreover, the efficiency here is super low; the equation is roughly Vout/Vin, which gives 3.3V/15V = 22%. Luckily, the buck regulator design works fine.

## 2.5 Storage Subsystem

This consists of three tanks: one stores the nicotine-containing fluid, another stores a liquid to dilute the nicotine fluid, and the third contains a bad-tasting fluid to decrease addiction. Each tank has a temperature and load sensor to monitor the liquid's state and detect temperature issues and leaks. Each tank should be carefully considered for device usage and the material used. Depending on the requirements, it can range from a simple glass beaker to a 3D-printed container.

## 2.6 Data subsystem

This system is used to protect users' data and enforce data security. Privacy is ensured by the SD card, with no wireless network connections; all communications are through SPI.

The Data Subsystem should not disclose details to anyone in most cases. The data subsystem is on the same board as the control subsystem to ensure signal integrity, as data is carried through PCB traces rather than wires.

## 2.7 Safety Subsystem

Safety Subsystems include all the Safety modules in each subsystem. When any one of them triggers any warning signals, the safety subsystem should take an appropriate action.

Safety Subsystem components include alarms, a flashing red LED, software shutdown logic, and fingerprint authentication for secure access. It logically connects both the microcontroller software and the physical safety hardware. The safety subsystem is located on the same board as the control subsystem and is exposed to the user.

## 3. Design Verification

All system requirements were checked off in accordance with the verification procedures outlined in Table 2 of Appendix A. All subsystems have been tested under realistic conditions, and data prove that the system behaves as expected. The fingerprint module accurately gated access with 90% true positives and no false positives, as shown in Fig. 9. Mixing substance casualty met all specifications, keeping the 50 mL and the 100 mL targets within their respective tolerances over repeated trials. Power rails were stable at 12 V, 5 V, and 3.3V under load, each within its respective voltage margin, as shown in Table 2. Safety overrides respond within 500 ms during fault simulation, and leak testing showed less than 5 g of unintentional drip. In all the tests conducted, the system stratified the verification criteria into dos concentration, temperature interlocks, and authentication requirements. Overall, each requirement was satisfied, showing that the device operated safely, is reliable, and functions according to specification.

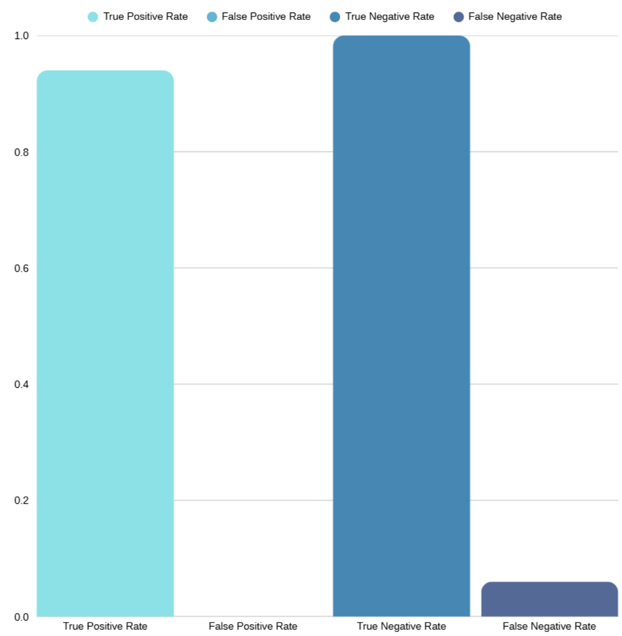


**Fig 9. Fingerprint sensor test data**

**Table 1 Power Circuit Data Table**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 12V Reg 1 | 12.044 | 12V Reg 3 | 12.160 | 3.3V Reg | 3.317 | 5V Reg | 5.0427 |
| | 12.045 | | 12.160 | | 3.317 | | 5.0428 |
| | 12.044 | | 12.160 | | 3.317 | | 5.0429 |
| | 12.044 | | 12.160 | | 3.317 | | 5.0425 |
| | 12.044 | | 12.160 | | 3.317 | | 5.0428 |
| | 12.044 | | 12.160 | | 3.317 | | 5.0427 |
| | 12.044 | | 12.160 | | 3.317 | | 5.0426 |
| | 12.044 | | 12.160 | | 3.317 | | 5.0426 |
| | 12.044 | | 12.161 | | 3.317 | | 5.0429 |
| | 12.043 | | 12.160 | | 3.317 | | 5.0427 |
| | 12.044 | | 12.160 | | 3.317 | | 5.0428 |
| | 12.044 | | 12.160 | | 3.317 | | 5.0426 |
| | 12.044 | | 12.162 | | 3.317 | | 5.0425 |
| | 12.044 | | 12.160 | | 3.317 | | 5.0427 |
| | 11.672 | | 12.160 | | 3.317 | | 5.0425 |
| | 12.044 | | 12.160 | | 3.317 | | 5.0428 |
| | 12.044 | | 12.160 | | 3.317 | | 5.0425 |
| | 12.044 | | 12.160 | | 3.317 | | 5.0429 |
| | 12.044 | | 12.160 | | 3.317 | | 5.0425 |
| | 12.044 | | 12.160 | | 3.317 | | 5.0427 |

# 4. Costs

## 4.1 Parts

**Table 2 Parts Costs**

| Part | Manufacturer | Retail Cost ($) | Bulk Purchase Cost ($) | Actual Cost ($) |
|---|---|---|---|---|
| Strain Gauge Load Cell - 4 Wires - 1Kg | Adafruit | 1*3.95$ | 1*$3.16 | 1*3.95$ |
| Adafruit HX711 24-bit ADC for Load Cells / Strain Gauges | Adafruit | 1*9.95$ | 1*$7.96 | 1*9.95$ |
| Temp Sensor | DFRobot | 4*6.9$ | 4*6.9$ | 4*6.9$ |
| Buzzer Piezo 12V 23.4mm Through Hole | Piezzo | 1*0.95$ | 1*0.95$ | 1*0.95$ |
| 3mm LED 3V Red | E-Shop Self Service | 1*4$ | 1*4$ | $0 |
| Pulse Sensor Amped | Adafruit | 1*25$ | 1*25$ | 1*25$ |
| MicroSD card breakout board+ | Adafruit | 1*7.5$ | 1*6.00$ | 1*7.5$ |
| Adafruit Industries Ultra-Slim Round Fingerprint Sensor and 6-pin Cable | Adafruit | 1*19.95$ | 1*19.95$ | 1*19.95$ |
| Linear Regulator | E-Shop Self Service | 1*1.45$ | 1*1.45$ | 0$ |
| 12V 2400mAh AA NI-MH Battery Pack | ECE Supply Center | 1*0.1$ | 1*0.1$ | 1*0.1$ |

| | | | | |
|---|---|---|---|---|
| Peristaltic Liquid Pump with Silicone Tubing - 12V DC Power | Adafruit | 3*24.95$ | 3*$19.96$ | 3*24.95$ |
| ESP32-S3-WROOM-1-N4 | DigiKey | 5*5.06$ | 5*$3.61964$ | 5*5.06$ |
| Total | | 200.6$ | 174.15$ | 195.15$ |

## 4.2 Labor

I need to give an approximate hourly wage estimate first. The Illinois minimum wage is $15 per hour, but we cannot apply it to our work since we are not working on a minimum-wage job. According to the Office of Student Financial Aid at the University of Illinois at Urbana-Champaign, the wage for a technical or professional job can range from $19 to $22 per hour. I will give a higher estimate, $21.50 per hour for each of us for technical work. However, it is not just like this because we are not doing KiCad or soldering PCBs every single second. For example, when doing some paperwork, I will set the wage at $16 per hour. When we are doing some other work, like finding parts, I will set the wage at 18$ per hour.

So, with that being said, each person is expected to do 2 hours of paperwork, 6 hours of random technical jobs, and around 18 hours of technical design, which makes the total cost of labor: 2*2.5*16+6*2.5*18+18*2.5*21.5=1317.5 $ per week. Moreover, we have 12 weeks to work, so that it will be around 1317.5 $*12 = 15810 $ per person. Thus, three people will cost 15810$ × 3 = 47430$.

# 5. Conclusion

## 5.1 Accomplishments

1) We turned a high-level idea into a working New Generation Safe Addiction Control and Recovery Device System with real hardware, real code, and a clear, modular design that future teams can build on.
2) We hit our safety goals for access control: the fingerprint sensor achieved an actual negative rate of 1.00 and a false positive rate of 0.00, so an unenrolled user cannot start the system. In contrast, enrolled users are accepted about 90% of the time across repeated trials.
3) We nailed dose control, which meets the 5% tolerance we set in our requirements.
4) We built a stable power system with three rails (12 V, 5 V, 3.3 V) and kept them very close to their targets (e.g., 5.00 V at ±0.05 V and 3.3 V at ±0.02 V), ensuring every subsystem runs reliably.
5) We proved our storage design works: after 144 hours of testing with different liquids, there was no loss in liquid mass in the storage pod.
6) We met key privacy and safety goals by keeping all data local on an SD card, allowing the user to erase it, avoiding network connections, and wiring the safety system so that a wrong fingerprint triggers a clear alert and prevents operation.
7) On top of all that, we provided clear, highly modular schematics and a verified design to the next group of people who could be working to enhance the device.

## 5.2 Uncertainties

1) The most significant uncertainty is still the disruptions. The current device relies too heavily on a stable lab environment, so additional noise protection is needed. For example, the fingerprint device should include a cleaning procedure and measures to prevent false access rejections; the load sensor should account for portable use; and the heart rate sensor should protect against noise from body movement and micro-shaking from wind.
2) Not enough consideration for micro-scale disruptions. We also did not have enough time to consider micro-disruptions, such as the magnetic field induced by the current, and during development, changing the wiring to go around the sensor already caused significant noise. There needs to be a more professional analysis that accounts for additional, more detailed factors in the future, to include more considerations.
3) Overnumbered Jump Wires. In our design, we split the PCB into separate sections to ease PCB wiring and used jump wires for power and signal delivery. Such actions served our purpose, but we used them too often in the design, which made the final result a mess. It caused more disruptions and uncertainty, as well as many safety concerns. Some unnecessary jump wires should be removed and combined into the PCBs.

## 5.3 Ethical considerations

We care about ethics above all else. We will focus mainly on these three aspects of ethics: privacy, care, academic integrity, and humanity. All of them have a different focus on different groups of people.

### 5.3.1 Ethics

1) Privacy:
   Different people hold different views on privacy, but we can estimate how people manage their privacy, especially the parts on which most people agree. Stoicism in Ancient

Greece is one of the great examples. According to [2], "The Stoics are determinists about causation, who regard the present as fully determined by past events, but who nonetheless want to preserve scope for moral responsibility by defending a version of compatibilism." On a Stoic view, the world is a network of physical causes, and each person is simply a bundle of "inner" parts bounded off from the rest of nature. Stoic ethics says that to live well is to live in agreement with nature, aligning our own reason with the rational order of the whole. Respecting the natural boundary between one person's inner processes and the external world is part of this harmony. When we pry into someone else's private thoughts or data, we try to cross that boundary and treat their inner life as something to control, which disrupts both their flourishing and ours. So respect for privacy is not just a social convention but a way of honoring the Stoic ideal of living according to nature: in others and in ourselves.

So, with that being said, we will follow the principle of utmost respect for privacy to ensure we are sufficiently respecting others, their "agreement with nature", and our own "agreement with nature".

2) Care:

I would also like to put strong emphasis on caring for each other, both in working together as a team and in caring for the product user who may use our design later.

As [3] was explaining, "Drawing conceptually from a maternal perspective, Noddings understood caring relationships to be basic to human existence and consciousness. She identified two parties in a caring relationship—“one-caring” and the “cared-for”—and affirmed that both parties have some form of obligation to care reciprocally and meet the other morally, although not in the same manner." As Noddings was talking about, caring relationships are as basic as a mother loving her children, and children loving their mother back, not just for the benefit.

Drawing on Nel Noddings’ ethics of care, we can see ourselves as always involved in relationships between “one-caring” and “cared-for,” with mutual obligations that are basic to being human. Caring means receiving others on their own terms and letting both our natural impulse to help and our “ideal self” guide us, rather than mechanically applying rules. In a team, that means we follow our shared agreements, but when someone is struggling, we offer support rather than treating them like a replaceable part. As designers, it means imagining the user’s vulnerabilities and refusing to build features that burden or harm them, even if they are technically successful. Centering care in this way keeps both our collaboration and our products genuinely humane.

On the other hand, we should also consider the problems the product's users may face. Many engineering successes are designed every year, but are still infamous for poor design in terms of “care”. We should put ourselves in the “ideal self” to carefully consider what users may need, so we never create functionalities that harm others.

3) Integrity:

Why do we care about academic integrity? We have an obvious answer there: it is wrong by the rules, but it is not the whole picture. Saying it is wrong not to follow academic integrity by rule is correct, but also not just correct by itself.

We care about academic integrity for more than just “because the rules say so.” In a Socratic sense, Integrity is a virtue: like bravery or helpfulness, it is about the wisdom to stay within the right range, avoiding both dishonesty and blind rule-following. For engineers, this virtue is crucial because our work affects other people’s safety and trust,

and the habits we form now will shape how we act when no one is watching. Practicing academic integrity means seeking help and using information wisely, mindfully, and carefully, so we learn to take real responsibility for our own work and its consequences. Academic Integrity is undoubtedly one of the most important, if not the most important, virtues for an Engineer. It matters in both ways. First, we have to be morally responsible to others. Second, this is the class U of I created for us to practice in a real design, and if we lack this exercise, then in the future, when nobody is doing precisely what we do, the harm will be directly back on us. We need to seek help and information wisely, mindfully, and carefully.
(Below is the set of integrity codes we need to follow.)
   a) https://studentcode.illinois.edu/article1/part4
   b) https://www.ieee.org/about/corporate/governance/p7-8

4) Humanity:
"Humans can never be used as the way toward a goal. Humans can only be the goal." (Philip Hillmer, ECE316)
As Professor Hillmer puts it, humans must never be used as a means to an end; each person has to be treated as an end in themselves. Even in ECE445, we should treat our project as if it could have a real-world impact and be mindful of every individual it might affect. We care about humanity and each person not only because we naturally care for others and share an ordinary human family, but also because our choices today shape the lives and feelings of future generations. So in our design decisions, we should always aim to promote the well-being of every person involved, never sacrificing anyone's dignity or safety for a goal.

### 5.3.2 Safety

Safety is also a big part that we need to ensure.
(This is a short version. The full version is in the Design Document.)

1. User safety: As a development team, we value user safety above all. We need to ensure users are completely safe when using our product.
2. Safety for society: We should always care about society and consider the various ways our project can affect it. We need to ensure that our project does not compromise anyone's safety.
3. Safety for the development team: We as a team should also care about our own safety.
4. General Rules (Full version on design document)
   a. No one is allowed to work in the lab alone.
   b. Safety measures should be enforced when working in a dangerous environment, such as high voltage or high temperature.
   c. The maximum continuous working time for a person is 6 hours, and the maximum daily working time is 8 hours.

## 5.4 Future work

1) Improve the fingerprint system so that it learns more users, runs faster, and achieves a higher actual positive rate while keeping the false positive rate at zero.
2) Add more sensors and logging (e.g., time-of-use, number of puffs, dose history), and make it easier for users to review or delete that data.
3) Shrink the hardware size and clean up the wiring and PCB so the system looks and feels closer to a real consumer device.

4) Do more extended storage and liquid tests (weeks or months) to see how different liquids behave over time, including leaks, evaporation, and contamination.
5) Refine the mixing and dose control to handle more dose levels and more types of liquids while keeping the error within the same tight range.
6) Design a user-friendly interface, add clearer status indicators, and simple controls so non-technical users can operate the system safely.
7) Run more user-style tests (simulated use with volunteers in a lab setting) to study comfort, ease of use, and how well the system might support addiction recovery plans.
8) Upgrade power management to explore higher-efficiency regulators, battery operation, and longer-term stress tests for all three voltage rails.
9) Shrink and polish the hardware. It is necessary to move from a lab prototype to a smaller, robust enclosure that looks and feels like a real consumer product.
10) Collect and negotiate with more individuals, as our project will likely be handed over to another group.

## References


[1] "Wages." Office of Student Financial Aid, https://osfa.illinois.edu/types-of-aid/employment/regulations/wages/. Accessed 11 Oct. 2025.

[2] Durand, Marion, Simon Shogry, and Dirk Baltzly. "Stoicism." The Stanford Encyclopedia of Philosophy, edited by Edward N. Zalta and Uri Nodelman, Spring 2023 ed., Metaphysics Research Lab, Stanford University, 2023, https://plato.stanford.edu/archives/spr2023/entries/stoicism/, Accessed 12 Oct. 2025.

[3] Sander-Staudt, Maureen. "Care Ethics." Internet Encyclopedia of Philosophy, https://iep.utm.edu/care-ethics/, Accessed 12 Oct. 2025.


## Appendix A : Requirement and Verification Table

**Table 3 System Requirements and Verifications**

| Requirement | Verification | Status (Y or N) |
|---|---|---|
| 1. Pulse sensors shall report values within ±20% of a reference at rest and under light exertion. | 1. Place pulse sensor on subject; count and start timer meanwhile. Record 2 min at rest, then 2 min after 30 s of step-ups. Log ESP32 readings | Y |
| 2. Fingerprint sensor shall gate all dosing; only enrolled prints unlock. | 2. (1) Enroll one finger. (2) Attempt unlock 10× with enrolled, then 20× with non-enrolled. (3) Try to dose without unlock. | Y |

| | | |
|---|---|---|
| 3. Safety overrides shall preempt dosing within ≤500 ms of a red-flag input (HR high, sensor fault, low battery). | 3. Induce each fault and measure time to pump-disable. | Y |
| 4. Commanded concentration should pump timing mapping shall achieve ±5% concentration when the Mixing/Storage are nominal. | 4. Request 10 setpoints, run dosing, measure actual concentration by gravimetric method.<br>Mean error <8% | Y |
| 5. Delivered dose concentration shall be within ±10% of target. | (1) Clean receiving cups. (2) Command dose at 10 targets. (3) Weigh each component pre/post to compute actual fractions. | Y |
| 6. No leaks: total unintended drip <0.1 g in 10 min after a dose. | Weigh pad under outlet before/after 10 min post-dose. | Y |
| 7. Provide regulated rails: 3.3 V, 5 V, 12 V within ±15% at max load; | Load each rail to design max; measure V (per schematic). | Y |
| 8. Tank temperature high (≥45 °C) shall trip alert and stop dosing. | Heat one tank slowly and observe | Y |
| 9. Materials compatibility: tubing/tanks shall show no visible degradation after 24 h exposure to test liquids. | Soak samples 24 h; inspect and weigh. | Y |
| 10. Detailed history requires authentication (password or fingerprint). | Open history UI without auth should expect deny; then with correct authentication should allow. | Y |
| 11. On any red-flag (HR high, sensor fault, battery low, tank leak, over-temp), system shall: (a) sound buzzer, (b) flash red LED | Trigger each red-flag one at a time. Then observe outputs of beep and red light. | Y |