ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Drink Dispensing Robot

Team #23

ANDREW JUNG
(crjung2@illinois.edu)
ETHAN CAO
(ecao4@illinois.edu)
MEGAN CHENG
(mxcheng2@illinois.edu)

TA: Frey Zhao

December 10, 2025

Abstract

The drink dispenser robot provides automated retrieval dispensing and delivery of a mixed beverage using three different subsystems: a coaster, a two-liquid dispenser, and a robot. The user specifies the drink ratios through the coaster, afterwards the robot will deliver the cup to the dispenser. The dispenser will fill the cup and the robot will then deliver it to the user. All subsystems achieved full standalone functionality and communicated reliably through ESP-NOW during testing. The robot consistently completed grabbing the liquid and transporting the cup. The robot demonstrated stable movement and interacted with the dispenser and coaster dependably. Although the system operated well, the project revealed the limitations in the accuracy of IR sensing which affected precision. Despite this constraint, the system successfully showed that a drink dispenser ecosystem is feasible and this project provides a strong foundation for future expansions and improvements.

Contents

1 Introduction

1.1 Motivation

The growing investment and successes of the service robotics industry demonstrates the increasing demand for fast and convenient refreshment delivery. Cafes such as those built by Yummy Future effectively compete against tradition cafes by combining quality beverages with novel experiences. While fully featured robot baristas can only be deployed in commercial settings, we seek to provide a basic, at home solution.

1.2 Problem

Too often, we're tired or distracted and put off getting a drink of water. Those small delays add up, leaving us dehydrated and drained without even realizing it. In social situations, the host tends to be responsible for providing beverages to a group, distracting them from conversations. Additionally, many users may get tired of drinking water and would prefer flavored drinks. Dehydration impacts focus, energy, and overall well-being, yet it happens so easily in our daily lives. Also, the additional fun aspect of having a personal bartender and a mini robot bringing your drinks encourages you to keep drinking, staying hydrated.

1.3 Solution

The solution is to create a drink delivery ecosystem that seamlessly connects a mobile robot, a drink dispenser hub, and a callback system for user interaction. The robot is responsible for navigating the environment, locating both the dispenser hub and the user's cup, and safely transporting beverages. As this is an at-home bar-like experience, it is assumed that the robot will be on a table setting with no complex obstacles. The hub acts as the liquid source, with the ability to dispense various ratios of drinks, allowing users to choose of mixed drink. The callback system enables the user to request service and call the robot without needing to approach the dispenser themselves, and allows the user to customize the drink ratio they want. Together, the robot docks at the hub, the hub dispenses the desired beverage into the cup, and the robot delivers the drink back to the user. Specifically, the robot integrates multiple sensors which are connected to an ESP32 microcontroller. On the dispenser side, the pump subsystem controls two liquid channels via motor-driven pumps. A precision docking subsystem with IR transmitters to aligns the robot under the dispenser nozzle to prevent spillage. Finally, the callback subsystem allows the user to call the robot and adjust set a custom drink mixture. It is in the shape of a coaster so that it can blend in with the environment flawlessly. This ecosystem ensures hydration is made convenient, safe, and customizable with minimal user effort.

1.4 High Level Requirements

To consider our project successful, our system must fulfill the following:

- The robot will retrieve the drink within 90 seconds on a clear tabletop without any obstacles. The introduction of obstacles shall not inhibit functionality but may increase service time.
- The dispenser shall be able to fill a standard red solo cup. No more than 1 mL of liquid will be spilled on the table for every 5 drink retrieval cycles.
- The robot will not get stuck on the dispensing station/coaster or fall off of the table.

1.5 Block Diagram

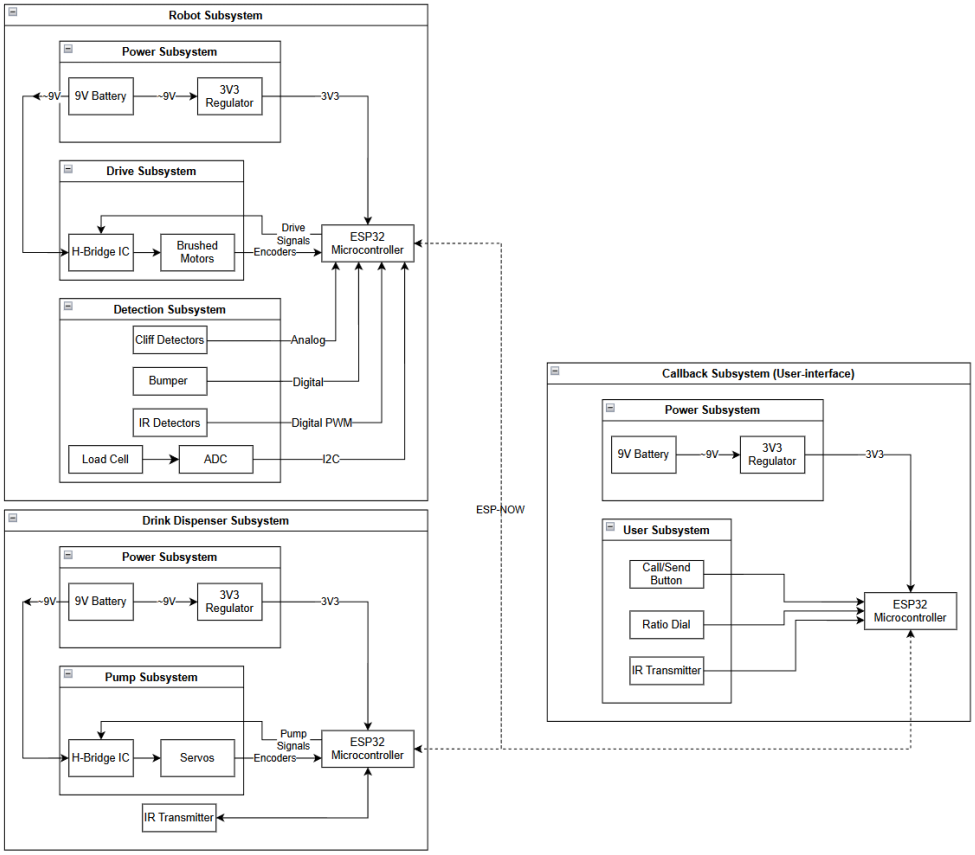

Figure 1: Overall Block Digram

The drink dispenser system includes 3 major components: The coaster to receive user input, the dispensing station to provide liquids, and the robot to transport cups between the coaster and dispensing station. One major change to the block diagram made over the course of the semester was the removal of discrete sensor PCBs, as breadboard prototypes allowed us to finalize sensor placements before ordering the PCBs

2 Design

2.1 Design Procedure

2.1.1 Driving Robot vs. Robotic Arm

When choosing between a driving robot versus a robotic arm we chose to use a driving robot. This choice was made not only to make it mechanically simpler but also to lower the cost. A robotic arm contains multiple joints and several degrees of movement instead of being on a 2D plane, this would greatly increase mechanical complexity, software complexity and require calibration. In contrast, using a pair of wheels and a simple set of DC motors is far more straightforward than designing and controlling a multi-axis servo system. Additionally, covering a larger area would mean a larger robotic arm while a mobile robot can remain the same size.

Another reason we chose to have a driving robot was because it is easier to drive around a two-wheel drive vehicle rather than a rotating mechanical arm. The parallel plane that the driving robot has with the ground would mean that the cup, which will always be level, is less prone to sloshing and spilling.

2.1.2 Mobile vs. Stationary Dispenser

We decided against using a mobile dispenser because it adds a significant amount of weight for the robot to carry around. This additional weight would require stronger motors and would increase power consumption, which is not desirable given our entire system is powered by 9V batteries. A moving dispenser would drain the batteries quickly and reduce the operating time as well as potentially needing a higher voltage system.

Another reason to not choose a mobile dispenser is that the sheer size of it would produce a robot that is larger. Since the robot has to navigate around obstacles a larger and bulkier design might prevent the robot from passing through tight spaces, which would hurt usability.

2.1.3 Tabletop vs. Floor

We choose to go with a tabletop implementation of our project because this avoids the obstacles that might be moving on the ground like humans and eliminates the need for us from having to build a 3 foot tall robot to make it convenient for users to grab the drinks. If the dispenser and the robot were both placed on the ground, the coaster would also need some way to transmit IR to the ground which mayme more than 2ft above the ground. This would significantly reduce the reliability of IR dues to the distance horizontally.

Another reason is that if the robot was still the same size it is right now but on the ground the user would need to bend down every time to retrieve/put the drink on the robot. This goes against our goal to make the product easy to use for someone who is lazy. The

solution to doing this would be to make a robot that was as tall as the user reach but having a robot of that scale for a budget of $150 would be unfeasible.

2.1.4 User Tracking Methods

We chose to use IR transmitter and receiver for the robot tracking where the user and dispenser. This was because it was the simplest and a reliable implementation within the constraints of our already complex multi-ESP32 system. Also we could use a modulated 38kHz signal that would filter out any noise from light and would be ideal for working indoor. The modulated signal allows the receiver to filter out sunlight and artificial light ensuring reliable detection. The hardware integration of IR requires only a single GPIO-driven PWM signal and a low-cost detector, allowing the robot to determine direction purely from relative signal intensity. This avoids the substantial additional hardware, calibration, and complications UWB would require (such as synchronization and an external radio module). BLE was another option we had, but decided against that idea because BLE has limited update rates compared to IR detection. BLE also requires pairing and would be noisy and inconsistent because it relies heavily on orientation. In contrast, IR is low-power, inherently directional, and easily integrated into the coaster's compact ring of transmitters. Overall, IR offers the most practical and minimal-complexity solution for our system.

2.2 Design Details

The following components and circuits provide functionality to our system. Full schematics for the robot (Fig. B.1), dispenser (Fig. B.2), and coaster (Fig. B.3) can be found in appendix **??**. Full PCB images for the robot (Fig. B.1), dispenser (Fig. B.2), and coaster (Fig. B.3) can be found in appendix **??**.

2.2.1 Physical Design

For our robot chassis, we will be using a combination of 3D printed and off-the-shelf parts. The chassis consists of two driven wheels with encoders and a caster wheel attached to a load cell. We will be using an N20 motor with an integrated encoder for the driven wheels and 3D print the caster wheel. A 3D printed cup-holder will be used to help the user place the cup in a repeatable manner. The bumper mechanism utilizes a mechanical keyboard switch and includes a bearing and rail to prevent binding during angled collisions (Fig. 2 and Fig. 3).

Additionally, when operating with liquid, a safety hat can be attached to the robot to prevent spills from reaching sensitive electronics (Fig. 4 and Fig. 5).

Initial testing found that 3D printed plastic wheels often slipped and, therefore, did not allow the robot to move in a repeatable manner. To increase friction, robot wheels were printed with an internal structure (Fig. 6), which allowed a urethane rubber tread to be molded around it. [1]

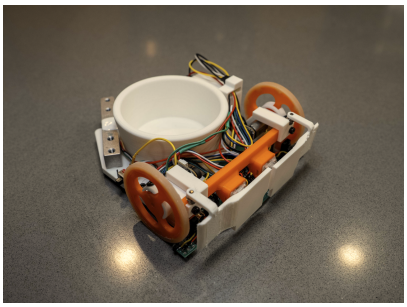

Figure 2: Robot

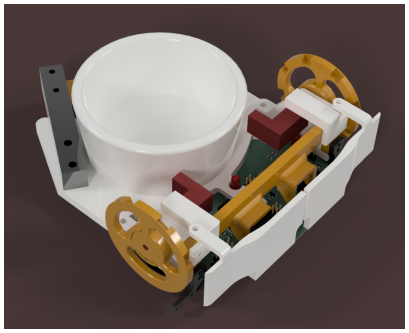

Figure 3: Robot Render

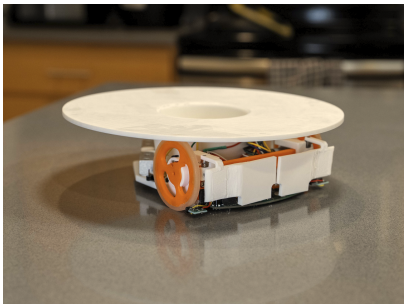

Figure 4: Robot with Hat

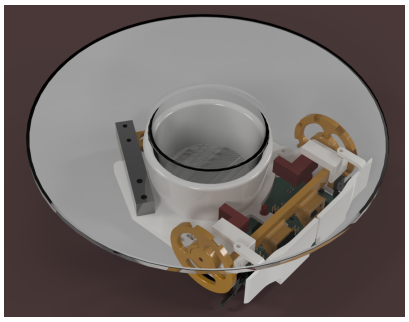

Figure 5: Robot with Hat Render

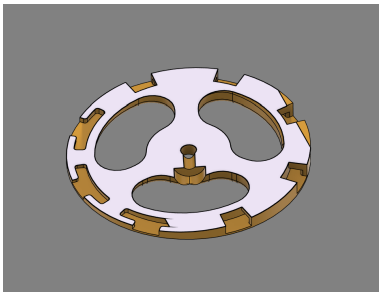

Figure 6: Cross Section View of Robot wheel

For our drink dispensing hub, we have two chambers to add liquid in as well as an overhang section to allow for routing of the tubing from the pumps to the where the robot will be sitting with the cup. As shown in the figure, the PCB will sit above these two tubes for the liquid to prevent any shorting of the circuit components as it is elevated above the water (Fig. 7 and Fig. 8). This also allows for centralized wiring of the pumps, IR LED, and the battery.

The dispenser hub is purposely made so that the overhanging part has no walls to allow the robot to move freely underneath it. The hub also has the IR LED at the center of the back panel to allow for full infrared range so that the robot can easily navigate to the dispenser.

The pumps chosen for the design are submersible 3V motor pumps. They are supplied by

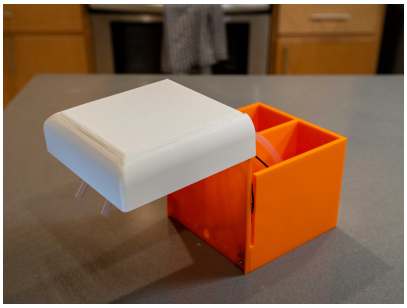

Figure 7: Drink Dispenser

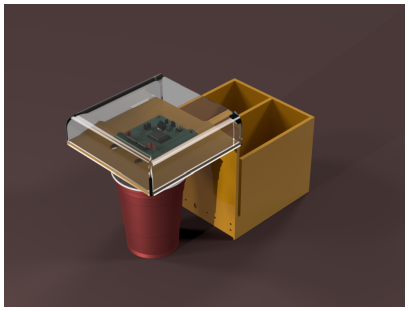

Figure 8: Drink Dispenser Hub Render from Fusion 360

Adafruit and can be placed in any container that holds water. When the pump is turned on, water is cycled through the plastic body surrounding the motor and then through a tube connected to the output of the pump leading to the cup. [2]

Initially, prototyping was done for the pump to see if DIY could be a potential option. Two different iterations were attempted, but none of them were successful. The first design tested was using a servo and 3D printed material, tubing, and ball bearings to push liquid out. The overall design did not work with the existing hobby servo we had on hand because there was too much torque from the ball bearings that rubbing against the grippy rubber tubing. [3]

Another prototype was done using a syringe and a small hobby motor. This design was a little tricky to make and without the proper sealant ordered to seal the water off from the motor and hold it stable, it was not stable enough to be utilized.[4]

In the end, the submersible pump was chosen. This method was compact and needs no additional mounting to any walls of the drink dispenser as it can just sit inside the liquid it is dispensing. It is also powerful enough to fill the cup within 10 seconds allowing the total cycle of user to robot to drink dispenser and back again to fall within our requirement of 90 seconds.

Figure 9: Coaster

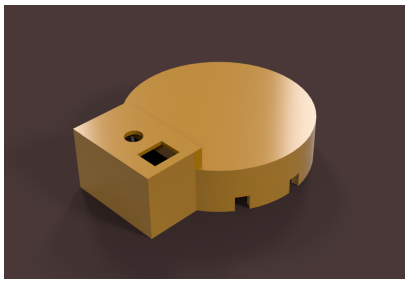

Figure 10: Coaster Render

The coaster (Fig. 9 and Fig. 10))was designed to be a compact, intuitive user interface that fits naturally into a tabletop environment and act as a coaster, while hiding its electronics. A centralized user-input with the tactile call/send button and the 3D-printed covered po-

tentiometer dial allows the user to adjust drink ratios and control robot movement. The coaster is surrounded with radial array of seven IR transmitters that emit in 38 kHz modulated beacon, ensuring the robot can locate the user regardless of coaster orientation. The PCB and physical design is optimized to remain as compact as possible while still housing the ESP32, power system, IR modulation, and input components. This creates a user-friendly, aesthetically unobtrusive device that functions both as a drink coaster and as the primary control hub of the system.

2.2.2 Robot Hardware

Cliff Detection

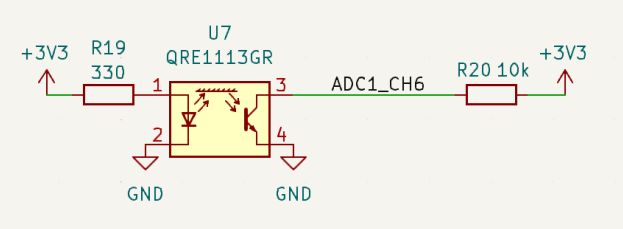

Figure 11: Cliff Detection Circuit

Different cliff detection approaches were compared. The initial idea was to use switches that would be pressed against the table. Although easy to implement, this creates more friction when driving and risks getting stuck in imperfections such as gaps between adjacent tables. Another option would be to use all-in-one time of flight sensors similar to those provided by ST Microelectronics [5]. These sensors would produce the best results. However, they did not fit our budget. The QRE1113 Sensor [6] was chosen because of its low cost and easy implementation. This sensor is often used as a color sensor on line following robots but can be utilized as a distance sensor, as objects which are further away create weaker reflections. The downside of this sensor is that it will not differentiate between cliffs and dark surfaces. The final implementation shown in figure 11 uses the sensor's integrated BJT to create a common-emitter amplifier to amplify the detected IR photons into an analog signal.

IR Detection

The selected IR sensor was the TSSP58P38, this sensor was selected because of its ability to receive a modulated signal and determine its intensity. Our system uses a 38kHz modulated signal to filter out background IR noise from sources such as the sun or incandescent lighting [7]. In addition, this sensor can detect the intensity of light received. When using two or more TSSP58P38 sensors, the direction of a light source can be determined using intensity data. The sensor is attached to an interrupt capable pin that is used to process the pulse width signal that the sensor produces (Fig. 12).

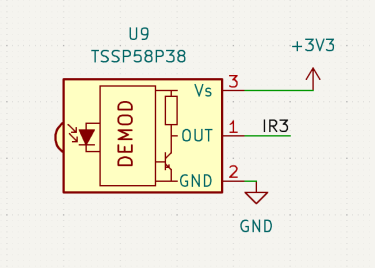

Figure 12: IR Detection Circuit

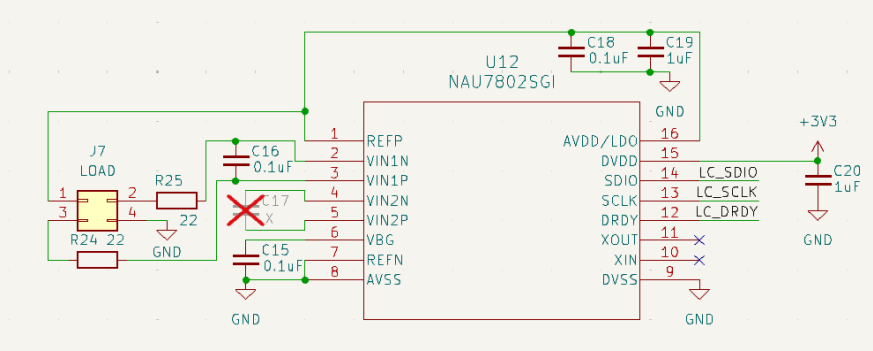

Figure 13: Load Cell Circuit

Load Cell

A load cell is needed to determine how full a cup is when it is placed on the robot. This prevents the dispenser station from overfilling a cup if the user places a partially filled cup on the robot. The load cell utilizes a Wheatstone bridge to determine the deflection of a metal beam. The output of the Wheatstone bridge must be amplified in order to detect changes, as the raw voltages are extremely low [8]. A NAU7802SGI analog to digital converter is used to interface between the load cell and microcontroller and is implemented (Fig. 13) according to the reference circuit in the datasheet [9].

Drive

The drive system utilizes a dual H-bridge IC to control two motors, one for each side of the robot (Fig. 14). The integrated wheel encoders communicate movements using a quadrature signal and the H-bridge receives commands via enable and direction signals. The L298N IC was chosen as it was provided by the ECE self service shop. Unlike the rest of our circuit, this IC utilizes 5V logic so an additional LDO was added to provide logic power. The datasheet [10] was consulted to determine that a logic level converter would not be needed, as the high state voltage of the input pins was below 3.3V (Fig. 15).

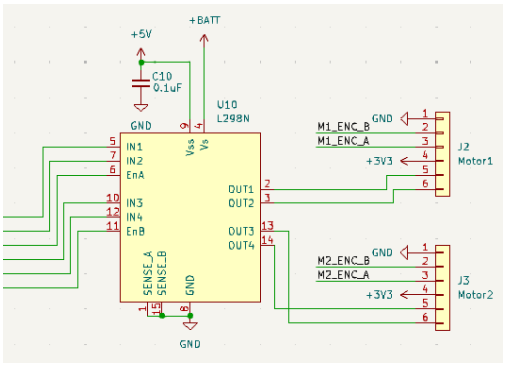

Figure 14: Dual H-Bridge Motor Driver Circuit

V_{iL}	Input low voltage (pins 5, 7, 10, 12)		–0.3		1.5	V
V_{iH}	Input high voltage (pins 5, 7, 10, 12)		2.3		V_{SS}	V
I_{iL}	Low voltage input current (pins 5, 7, 10, 12)	V_i = L			–10	µA
I_{iH}	High voltage input current (pins 5, 7, 10, 12)	V_i = H ≤ V_{SS}–0.6V		30	100	µA
V_{enL}	Enable low voltage (pins 6, 11)		–0.3		1.5	V
V_{enH}	Enable high voltage (pins 6, 11)		2.3		V_{SS}	V

Figure 15: Logic Levels from L298N Datasheet

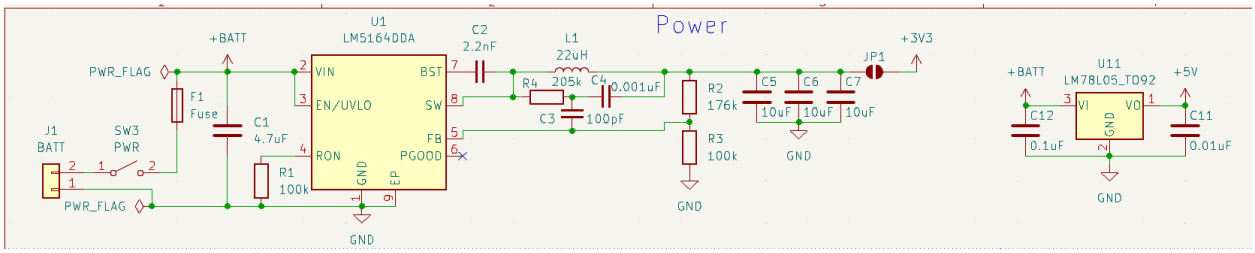

Figure 16: Power Regulation Circuit

Power

A buck converter was chosen for the 3.3V power source, as the ESP32-S3 microcontroller can consume a significant amount of power when wireless functionality is enabled, so an LDO is insufficient. This buck converter was designed with an example schematic provided by TI's power designer tool [11] and its impelmentation is shown if figure 16. The LM5164 buck converter [12] is an adjustable converter and the feedback pin is used to determine the output voltage. The regulator will adjust its output voltage to bring the voltage at the feedback pin to match its internal reference of 1.2V. Equation 1 is used to determine suitable resistor values.

$$R_{FB3} = \frac{1.2}{V_{\text{out}} - 1.2} \times R_{FB2} \tag{1}$$

Microcontroller

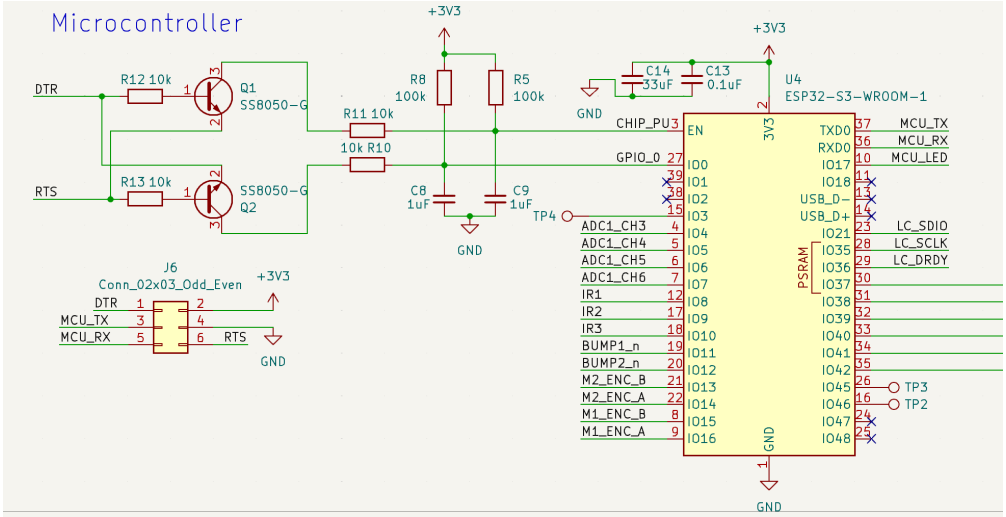

Figure 17: Microcontroller and Programming Circuit

The microcontroller chosen was a ESP32-S3-WROOM-1 [13] due to its availability from the ECE supply center and wireless capabilities. The programming circuit is derived from the circuit provided in the ECE 445 Wiki [14].

2.2.3 Dispenser Hardware

Pump

Please refer to figure 14 and the Drive portion under Robot hardware. The design of the hardware is entirely the same with the exception that the submersible pumps do not use the encoders that are wired and instead only use the PWM output of the driver to control when the pumps will be on.

Power

Please refer to Fig. 16 and the Power portion under Robot hardware.

Microcontroller

Please refer to Fig.17 and the Microcontroller portion under Robot hardware.

IR

The IR LED used in our design is an IR204A procured from the ECE Service Shop. Because the LED must be driven with a high frequency pulsed current to achieve the best transmission range, it is connected to an IRFML8244TRPBF NMOS transistor, which is

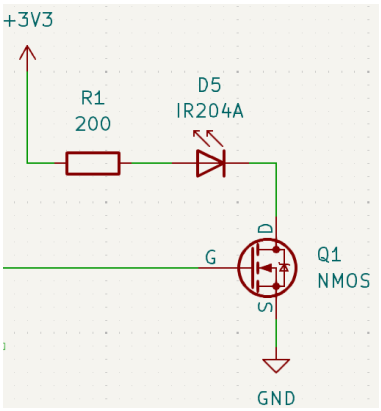

Figure 18: IR LED Schematic for Dispenser

controlled by a GPIO pin on the microcontroller. This configuration allowed the microcontroller to generate a clean stable 38kHz modulate waveform while ensuring that the LED current remains clean and stable. The NMOS stage also provided a fast switching speed and improved signal integrity which protects the GPIO from excess current draw. This results in a reliable IR output for detection by the robot's sensor.

2.2.4 Coaster Hardware

IR

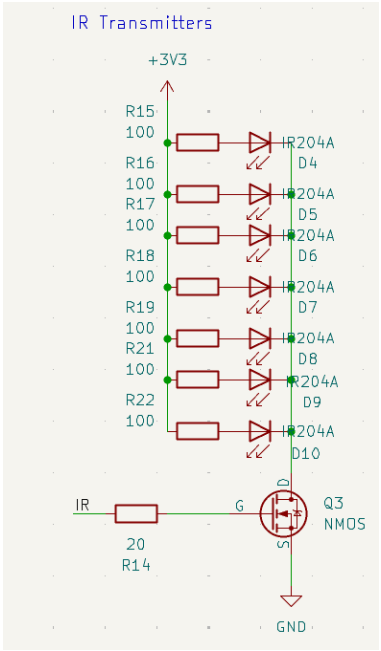

Figure 19: IR LED Schematic

Please refer to the IR portion under Dispenser hardware. For the specific coaster implementation of IR transmitter, it is connected to the GPIO18 for clean, flexible output for a

timed waveform. There are 7 transmitters in total so that the user has directional freedom when placing the coaster. The specific resistance values are calculated accounting the 7 transmitters. The IR204A has a forward voltage of $V_f = 1.2\,\text{V}$ at $20\,\text{mA}$, meaning each IR LED should use approximately a $100\,\Omega$ series resistor. The resistor value is computed as

$$R = \frac{3.3\,\text{V} - 1.2\,\text{V} - 0.05\,\text{V}}{0.02\,\text{A}} \approx 100\,\Omega.$$

Power

Please refer to Fig. 16 and the Power portion under Robot hardware.

Microcontroller

Please refer to Fig.17 and the Microcontroller portion under Robot hardware.

Dial/Potentiometer

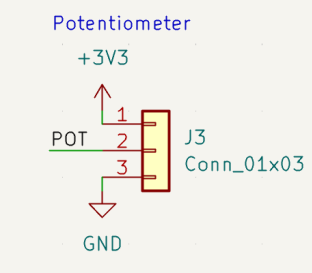

Figure 20: Potentiometer Schematic

Used a 10K SINGLE TURN 2W POTENTIOMETER from the ECE supply store. It is connected to the GPIO4 for a stable analog reading (ADC $0 \sim 4095$), and connected to $330\,\Omega$ Resistor limits charge spikes into the ADC sample cap.

Tactile Swith

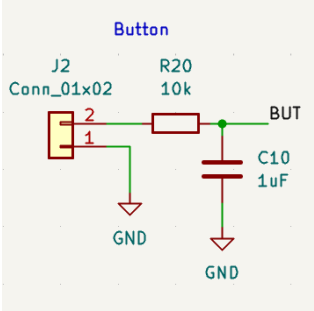

Figure 21: Tactile Switch Schematic

The ESP32 reads the state of the button (on/off) connected to the GPIO4 for digital reading.

2.2.5 Robot Software

Sensing

Since the robot's cliff detectors were analog sensors, we designated a brightness cutoff where the surface would be considered a cliff. If we were to create a second revision of the design, we would likely modify our hardware so that the cliff detectors could drive a digital interrupt. The IR detectors produced digital, pulse width outputs which could be analyzed using system interrupts and timers. For load cell readings, we used the Adafruit NAU7802 library [15].

Motion

The robot's motion is driven by two main software systems. The wheel velocity control loop implemented at each motor and the motion planning queue.

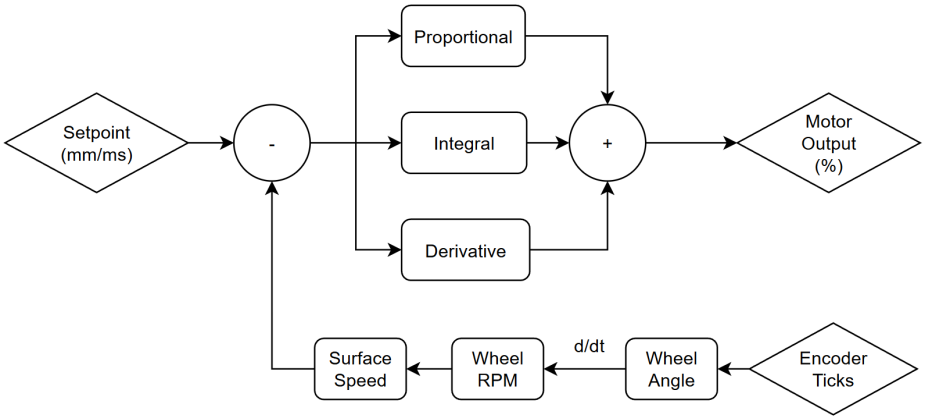

Figure 22: Robot Wheel Velocity Control System

Figure 22 shows the control loop which drives each robot wheel. To optimize encoder data collection without slowing down other processes, the ESP32Encoder [16] library was used to outsource counting to a hardware pulse counter peripheral. The setpoint is set by the motion queue and surface velocity is determined using encoder counts and equations 2, 3, and 4. Approximate K_p, K_i, and K_d constants were determined using the Ziegler-Nichols method [17] with a control type of minimum overshoot.

$$\theta_{wheel}(t) = \frac{2\pi}{1400\ (ticks\ per\ rev)}\ \text{ticks}(t) \tag{2}$$

$$\text{RPM}(t) = \frac{60}{2\pi} \frac{d\theta_{wheel}(t)}{dt} \tag{3}$$

$$v(t) = \text{RPM}(t) \cdot \frac{2\pi r}{60} \tag{4}$$

Motion planning queue is a custom double ended queue implementation which allows normal movements to be appended to the end of the queue and evasive movements to be pushed to the front of the queue. Individual queue items are formatted with a distance to cover over a requested time. The requested time is used to calculate wheel velocity and used in our timeout error calculations but is not strict.

Obstacle Avoidance

Obstacles including cliffs and objects are avoiding using the cliff detection and bumper subsystems. Each loop, these sensors are polled and evasive movements are pushed to the front of the queue if needed.

IR Tracking

IR tracking is done with a combination of the signal strengths of the 3 forward facing IR detectors. The difference in IR strength on the left and right of the robot is the error fed into a proportional controller which determines wheel power output. The center sensor is used to determine if the target has been reached.

2.2.6 Dispenser Software

The dispenser software is split into two main portions. The first of which being the receiving of the signals and processing of the pump ratio. The second portion is the communication of the dispenser status to the coaster and robot.

Pump Ratio

The pump ratio is controlled using a time-based approach, where the pump runs for 10 seconds to fill a cup. After dispensing, a short delay is added to account for any remain liquid from the pump tubing to help maintain the accuracy and reduce potential spilling and dripping.

Pump Status

To ensure safe and reliable operation of the dispenser, it is essential to transmit status message so the system can properly coordinate actions and prevent spills from happening. There are three different status messages that can be sent which are a message of 0 (IDLE), 1(WAITING), and 2(DISPENSING).

2.2.7 Coaster Software

The coaster software is responsible for handling user input (button and dial), generating drink-ratio, transmitting a modulated IR, and controlling robot movement.

Button Status bit[1:0]	Description
0	First press calls the robot to the coaster (once on startup).
1	Sends robot to dispenser; communicates selected ratio value.
2	Dispense sequence begins: pump on $\rightarrow$ pump off $\rightarrow$ wait 4 s $\rightarrow$ robot returns to coaster.

Table 1: Coaster Button FSM Behavior

- **Button FSM**
  - Initializes in state 0 (Table 1) on startup.
  - States 1 and 2 (Table 1) alternate based on user presses (e.g., $0 \rightarrow 1 \rightarrow 2 \rightarrow 1 \rightarrow 2 \ldots$).
- **Button Handling**
  - Implements $50\,\mathrm{ms}$ software debouncing.
  - Uses latched-press detection to prevent double triggers.
- **Ratio Selection (4-bit)**
  - Reads the potentiometer through a 12-bit ADC (0–4095).
  - Maps ADC input to a 4-bit drink ratio value (0–15).
- Uses LEDC PWM at $38\,\mathrm{kHz}$.
- IR beacon is pulsed with $40\,\mathrm{ms}$ ON / $60\,\mathrm{ms}$ OFF timing.

2.2.8 Wireless System Integration

To enable wireless coordination between the robot, dispenser, and coaster, the system uses a custom communication protocol built on ESP-NOW. This protocol provides low-latency, connectionless messaging suitable for real-time command and status updates.

Custom Message Protocol

Each message consists of a 2-byte frame: The upper byte contains 2 bits for the *Sender ID* and 2 bits for the *Message Type*. The lower byte provides a 1-byte data field used for status codes, ratio values, or control flags.

Device identifiers are assigned as follows: Robot -¿ `0`, Dispenser-¿ `1`, and Coaster-¿ `2`

Key Message Types

- **STATUS**: Reports device state, IR detection state, and location information.
- **CALL**: Requests robot movement commands (`0x00`, `0x01`, `0x02` depending on state).
- **REQ_FILL**: Sends a 4-bit drink ratio from the coaster to the dispenser.
- **CMD_IR**: Signals the start of IR-beacon transmission on the dispenser or coaster.

All ESP-NOW packets are broadcast to:

`FF:FF:FF:FF:FF:FF`

Message Structure

The following code defines the message format and enumerations used by all three devices:

```
typedef struct {
    deviceIDs deviceID;   // sender ID
    msgTypes msgType;     // message type
    byte     data;        // data
} msgStruct;

typedef enum {
    MSG_TYPE_STATUS   = 0b00,
    MSG_TYPE_CALL     = 0b01,
    MSG_TYPE_REQ_FILL = 0b10,
    MSG_TYPE_CMD_IR   = 0b11,
} msgTypes;

typedef enum {
    DEVICE_ID_ROBOT     = 0x00,
    DEVICE_ID_DISPENSER = 0x01,
    DEVICE_ID_COASTER   = 0x02,
} deviceIDs;
```

3 Design Verification

3.1 Robot Battery Life

Assuming a battery capacity of 500 mAh [18], we can use measured current data and equation 5 to calculate expected runtime.

$$Runtime(hours) = \frac{Capacity(mAh)}{I_{\text{load}}(mA)} \tag{5}$$

Operating State	Current (mA)	Battery Life (hours)
Idle	67	7.46
Suspended Motors	140	3.57
Stalled Motors	250	2.00

Table 2: Current Consumption and Estimated Battery Life for a 500 mAh Battery

3.2 Power Supply

The power supply was verified by powering a disconnected buck converter implementation with 9V and probing the 3.3V output with an oscilloscope (Fig. 23). The measurement found a maximum output of 3.59V and a minimum output of 3.28V. This is within the acceptable range for powering our ESP32-S3 Microcontroller according to its datasheet [13].

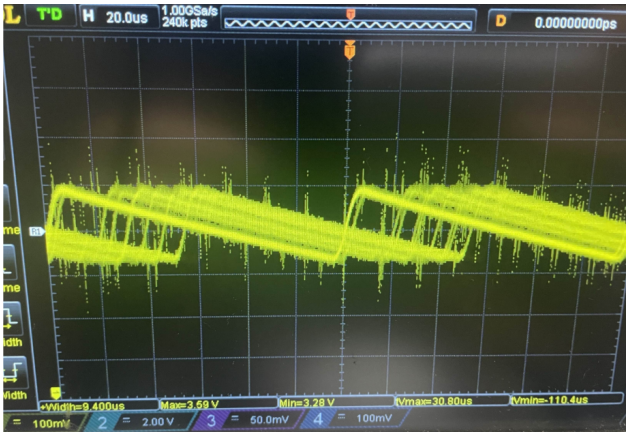

Figure 23: Oscilloscope Measurement of 3.3V Regulator Unloaded

3.3 Motor Driver

The motor driver was used for two different parts of the project. One motor driver was used for the movement of the robot and the other was used to drive the two pumps. According to our requirements and verifications table it is stated that the pump system will not consume over the current limit of the h-bridge, motor or battery. After measuring the total current draw of the entire drink dispenser PCB the current output was a maximum of 250mA which is well below the tolerance of the H-Bridge IC as well as the battery. C.5

Another requirement was that both pumps must run at the same speed. This was accomplished and successful shown by setting the dial to 50% on the coaster and allowing the pump to dispense equal amounts of liquids. The liquids were then weighed and measured. The liquids both measured within 167-170 gram range. This verifies that the pumps ran at equal speeds at equal speeds and dispensing the correct ratio of liquids. C.5

3.4 Cliff Detection

The cliff detection subsystem was verified by placing the robot on a table and commanding it to drive forward at a constant speed. We found that the robot successfully reverses and turns upon reaching a cliff. Further testing found that extremely dark surfaces which absorbed the majority of infrared light would not work, as it was impossible to differentiate between the tables surface and no reflection with our current sensor setup. C.1

3.5 IR Tracking

Infrared tracking testing was done by placing the coaster on the ground and using the robot's IR sensors to locate the coaster. Initial testing looked promising, as the robot could determine which direction received higher intensity IR light. However, extensive testing showed that the robot could not accurately find the light and consistently had an error of around 400 millimeters. We determined that this was most likely because our sensors did not have a high enough dynamic range to determine if the target had been reached. Even when far away from the IR source, the sensor would read maximum intensity. When we reduced the intensity of our IR source, the sensor would only work from a few inches away. Ultimately, we deemed this insufficient and did not incorporate it into our final design. C.1

4 Costs

Part	Manufacturer	Retail Cost ($)	Actual Cost ($)
Robot Motor	MECCANIXITY	10.00	20.00
1528-4540-ND	Adafruit	3.95	3.95
LM5164DDA	Texas Instruments	4.06	12.18
TSSP58P38-ND	Vishay Semiconductor	1.84	5.52
QRE1113GR	onsemi	1.09	4.36
NAU7802SGI	Nuvoton	1.62	1.62
FC-203 BRIGHT TIN	Bel Fuse Inc.	0.10	0.36
02171.25HXP	Littelfuse Inc.	0.51	2.10
IR204A	Everlight Electronics Co	0.09	0.72
LSXBH8080YKL220M	Taiyo Yuden	0.40	1.64
ESP32-S3 WROOM 2	Espressif Systems	10.67	32.01
732-5016-ND	Würth Elektronik	0.15	6.98
P120PK-Y25BR10K	TT Electronics/BI	1.42	1.42
Y41A00421FPLFS	C&K	1.61	1.61
MX1A-E1NW	Charry Americas LLC	1.25	2.50
L298N	STMicroelectronics	11.78	23.56
609-3234-ND	Amphenol ICC (FCI)	0.49	1.47
68000-220HLF	Amphenol ICC (FCI)	1.28	1.28
SS8050-G	Comchip Technology	0.24	1.48
IRFML8244TRPBF	Infineon Technologies	0.43	0.86
233	Keystone Electronics	0.54	1.62
ALL SMD Components	Omitted for Simplicity	N/A	5.00
Total			133

Table 3: Parts Costs

The labor costs were estimated by taking the average yearly salary of an electrical engineering graduate and calculating the average hourly rate. Then it was estimated that there would be about 6 hours of work done a week for 14 weeks so that would result in around a $10k labor cost with a parts cost of around $130 (Table 3). This results in a cost of around $10130 spent for the production of this drink dispenser hub

5 Conclusion

Our project successfully met all major requirements. Each subsystem satisfied its individual verification points, though integration introduced some challenges.

Successes

The drink-dispenser robot reliably modulated IR signals, avoided obstacles, and communicated across all subsystems using ESP-NOW. The IR receiver consistently detected modulated signals even with multiple transmitters in the coaster subsystem. The robot's movement stayed stable and prevented spills both in normal operation and when detecting cliffs or obstacles. The wireless queue with custom message handling introduced minimal latency, allowing seamless user interaction and accurate dispensing ratios. Power delivery also remained stable, with the 3.3 V and 5 V rails and motor drivers meeting all current requirements.

Challenges

Challenges that we faced included having to go through multiple iterations of the pump design and IR accuracy. We did successfully choose a working and robust pump version in the end. For the IR accuracy, we had issues with the low dynamic range that the sensor used provided. It was hard to determine exactly which direction the IR was being detected from as the IR range would go from, no IR detected to full IR detected. Purchasing a sensor with a wider dynamic range could be on solution to the issue. Another solution to this issue could be to combine IR with other methods of location tracking, such as the inclusion of line following when getting close enough to the dispenser hub.

Future Work

Future improvements include adopting a more precise short-range location-finding method or calibrating the robot to maintain a map of the dispenser's location. Additional coaster or call-station devices could support multi-user environments. Accuracy could also be improved with a real-time operating system .

Ethics

Throughout development, we considered electrical, mechanical and food-safety risks. We minimized short-circuit and shock hazards by elevating PCBs, separating liquids from electronics, adding 3D-printed protective features and using low-voltage systems. These decisions align with the IEEE Code of Ethics, emphasizing public safety and responsible engineering. All liquid-contact components were chosen to comply with FDA food-safety guidelines. Altogether, these practices demonstrate our commitment to safe and ethical engineering.

References

[1] S. Schaffer. "How to Make Custom Wheels for Your Robot - Molding/Casting Polyurethane," Accessed: Nov. 10, 2025. [Online]. Available: https://www.youtube.com/watch?v=U-_xxI6qvlg.

[2] Adafruit. "Submersible 3V DC Water Pump - 1 Meter Vertical Type," Accessed: Oct. 25, 2025. [Online]. Available: https://www.adafruit.com/product/4547.

[3] Thingiverse. "Thing 5170737," Accessed: Sep. 23, 2025. [Online]. Available: https://www.thingiverse.com/thing:5170737.

[4] DIYtouch. "Create Your Own Submersible Water Pump," Accessed: Oct. 20, 2025. [Online]. Available: https://www.youtube.com/shorts/7UcwS6lWxI0.

[5] STMicroelectronics. "Time-of-Flight Sensors." Accessed: Nov. 5, 2025. [Online]. Available: https://www.st.com/en/imaging-and-photonics-solutions/time-of-flight-sensors.html.

[6] ON Semiconductor. "QRE1113 Optical Reflective Sensor Datasheet. "[Online]. Available: https://www.onsemi.com/download/data-sheet/pdf/qre1113-d.pdf.

[7] Vishay Semiconductors. "TSSP58P38 Proximity Sensor Module Datasheet. "[Online]. Available: https://www.vishay.com/docs/82476/tssp58p38.pdf.

[8] Load Cell Central. "Load Cell and Strain Gauge Basics," Accessed: Dec. 8, 2025. [Online]. Available: https://www.800loadcel.com/load-cell-and-strain-gauge-basics.html.

[9] Nuvoton. "NAU7802 24-bit ADC Datasheet. "[Online]. Available: https://www.nuvoton.com/export/resource-files/NAU7802%20Data%20Sheet%20V1.7.pdf.

[10] STMicroelectronics. "L298 Dual Full-Bridge Driver Datasheet. "[Online]. Available: https://www.st.com/resource/en/datasheet/l298.pdf.

[11] Texas Instruments. "WEBENCH Power Designer. "[Online]. Available: http://webench.ti.com/appinfo/webench/scripts/SDP.cgi?ID=3912C433EB0814E8.

[12] Texas Instruments. "LM5164 Synchronous Buck Converter Datasheet. "[Online]. Available: https://www.ti.com/lit/ds/symlink/lm5164.pdf.

[13] Espressif Systems. "ESP32-S3-WROOM-1/1U Datasheet. "[Online]. Available: https://documentation.espressif.com/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf.

[14] University of Illinois Urbana-Champaign. "ECE 445: ESP32 Example. "[Online]. Available: https://courses.grainger.illinois.edu/ece445/wiki/#/esp32_example/index.

[15] A. Industries. "Adafruit_NAU7802 — Arduino library for the NAU7802 24-bit ADC," Accessed: Dec. 8, 2025. [Online]. Available: https://github.com/adafruit/Adafruit_NAU7802.

[16] madhephaestus. "ESP32Encoder — Arduino library for ESP32 encoder handling," Accessed: Dec. 8, 2025. [Online]. Available: https://github.com/madhephaestus/ESP32Encoder.

[17] Wikipedia. "Ziegler–Nichols method," Accessed: Dec. 10, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Ziegler%E2%80%93Nichols_method.

[18] Energizer. "MAX–EU 9V," Accessed: Dec. 8, 2025. [Online]. Available: https://data.energizer.com/pdfs/max-eu-9v.pdf.

Appendix A Circuit Schematics

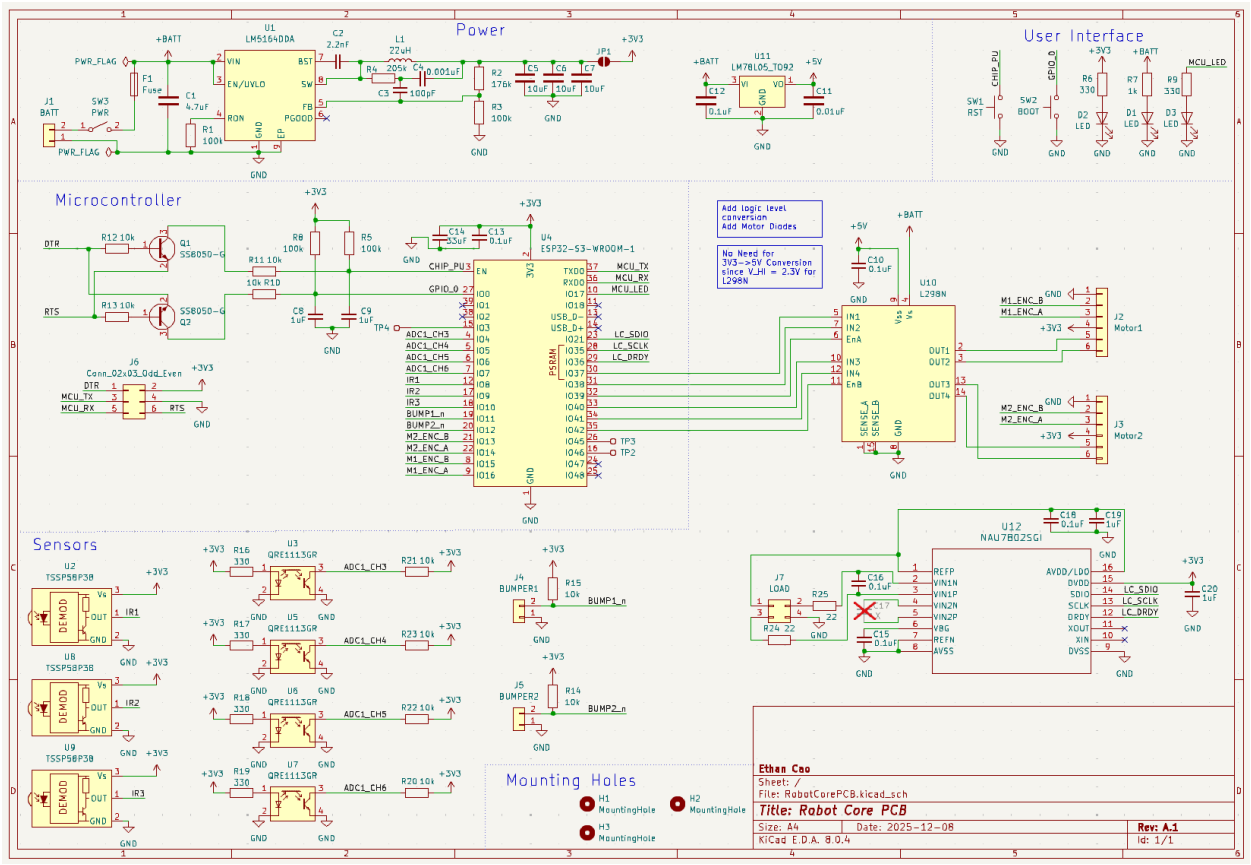

Figure B.1: Robot Schematic

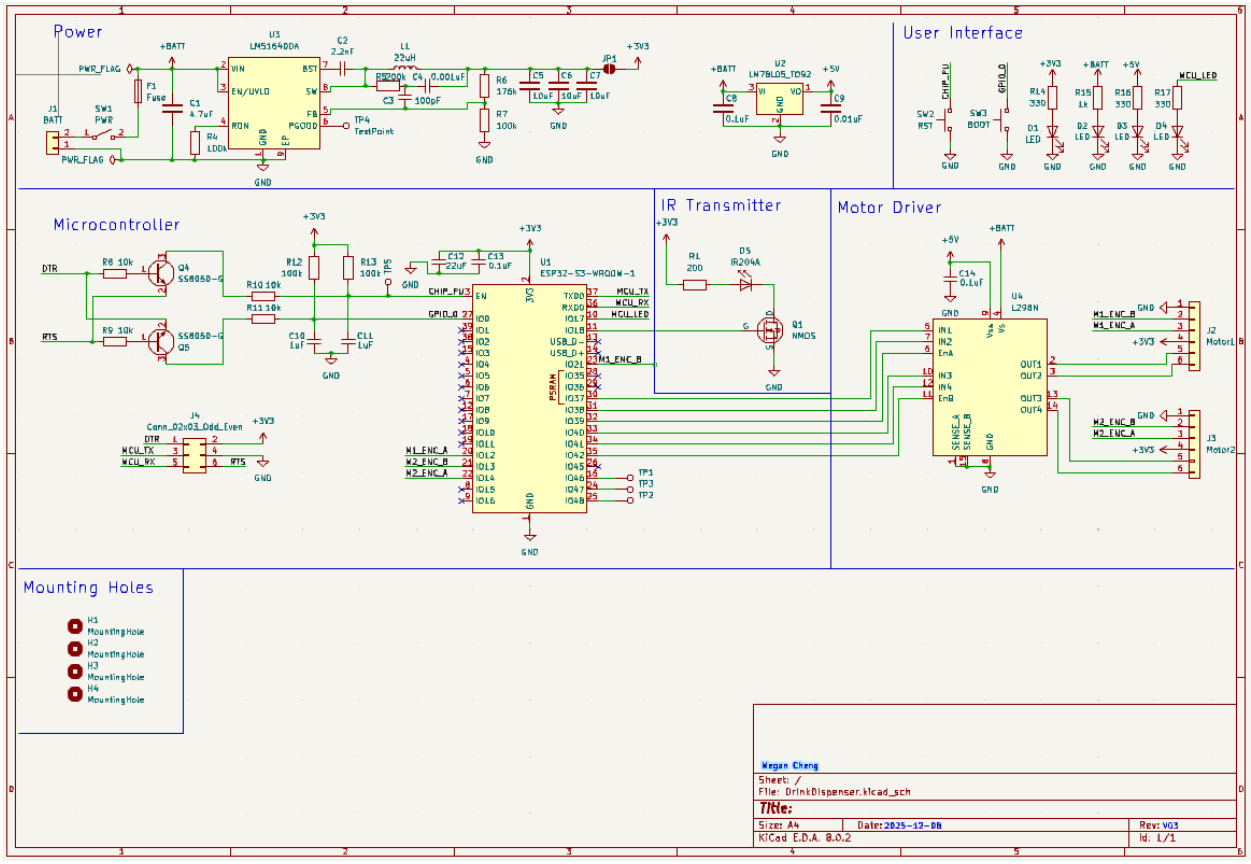

Figure B.2: Drink Dispenser Schematic

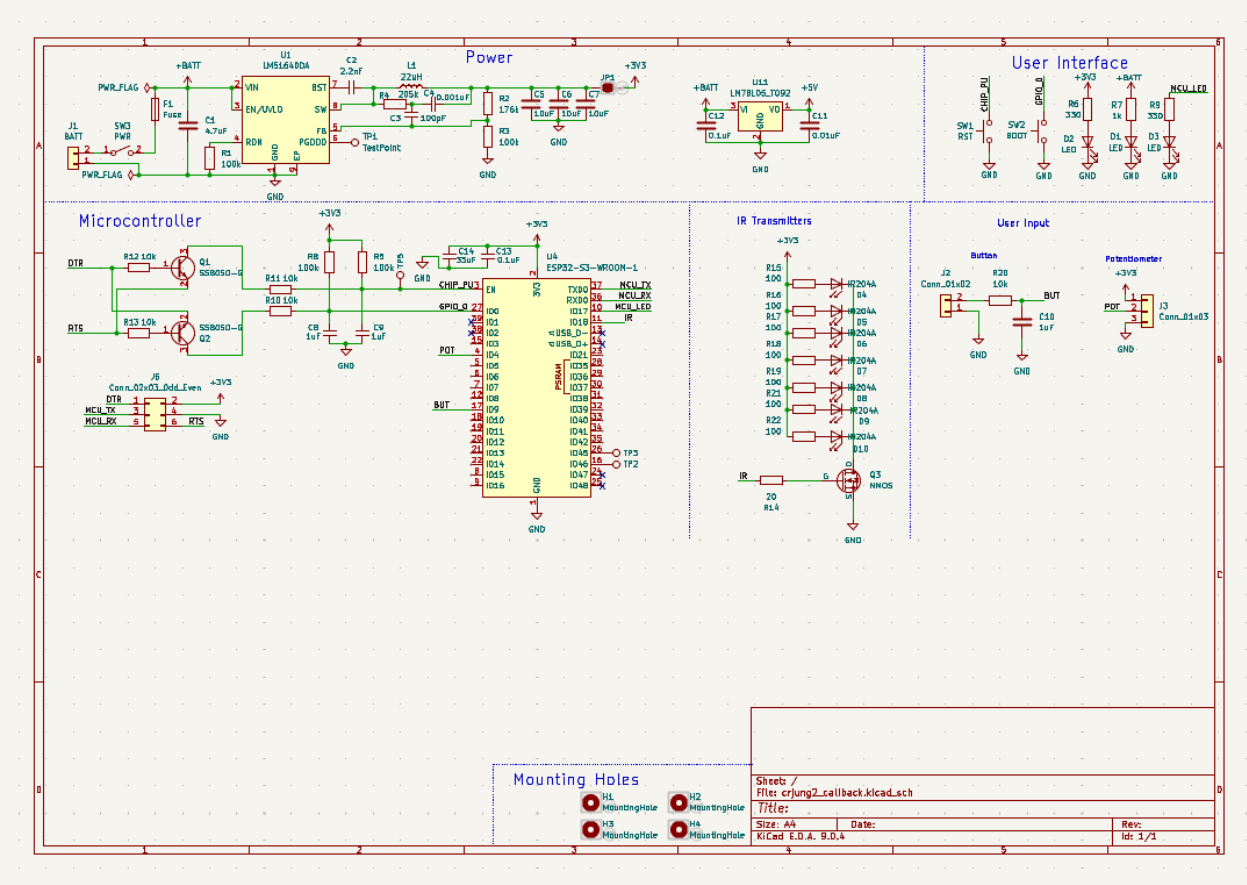

Figure B.3: Coaster Schematic

Appendix B PCB Layouts

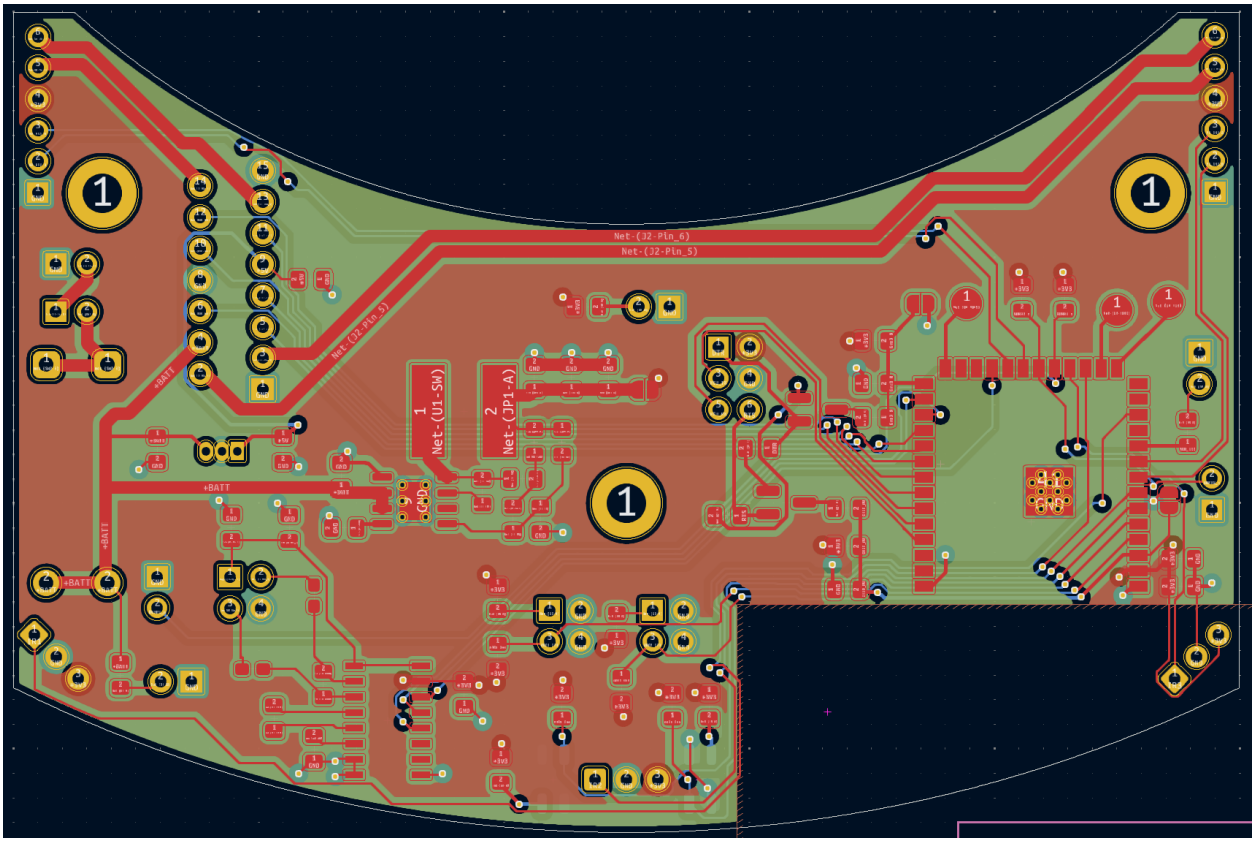

Figure B.1: Robot PCB

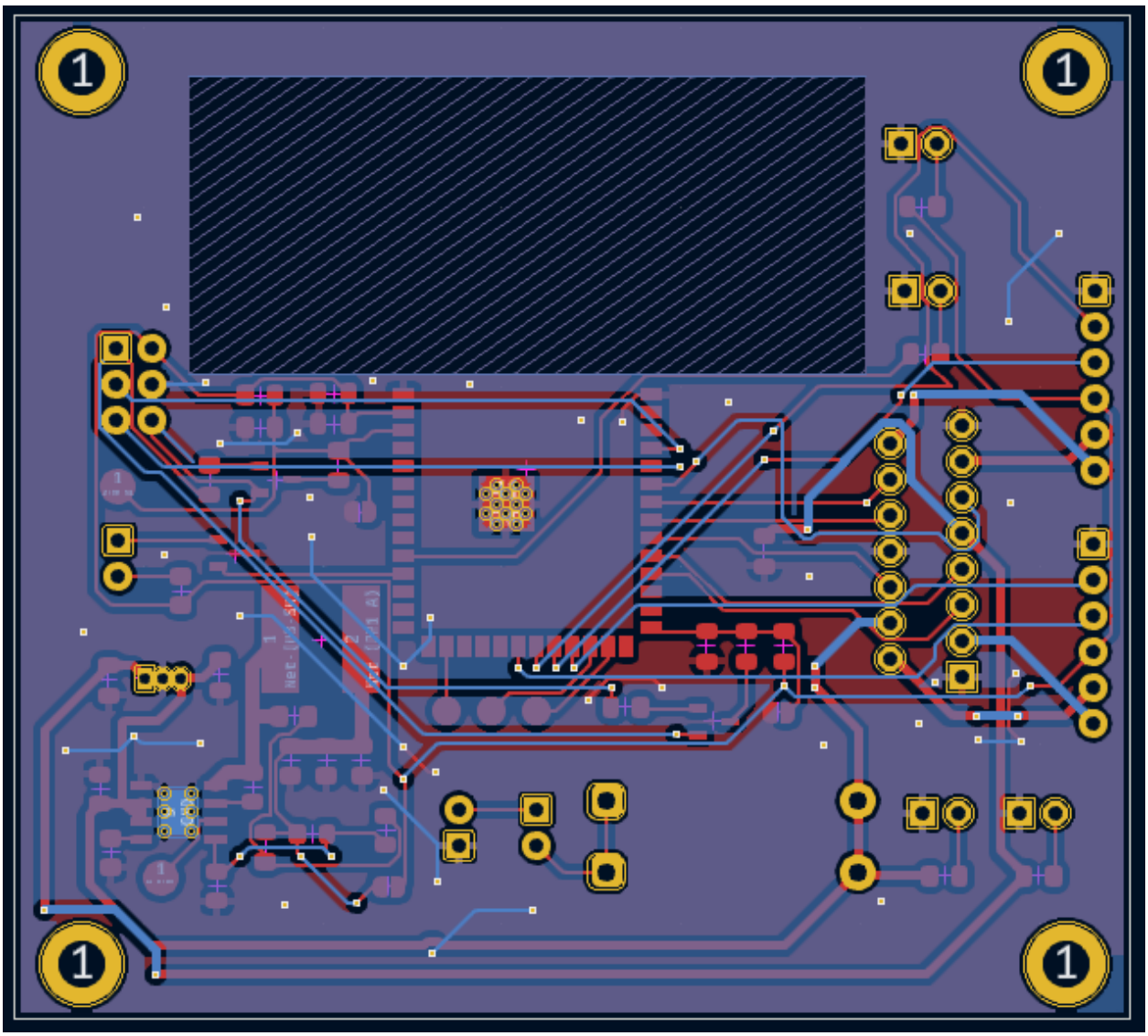

Figure B.2: Drink Dispenser PCB

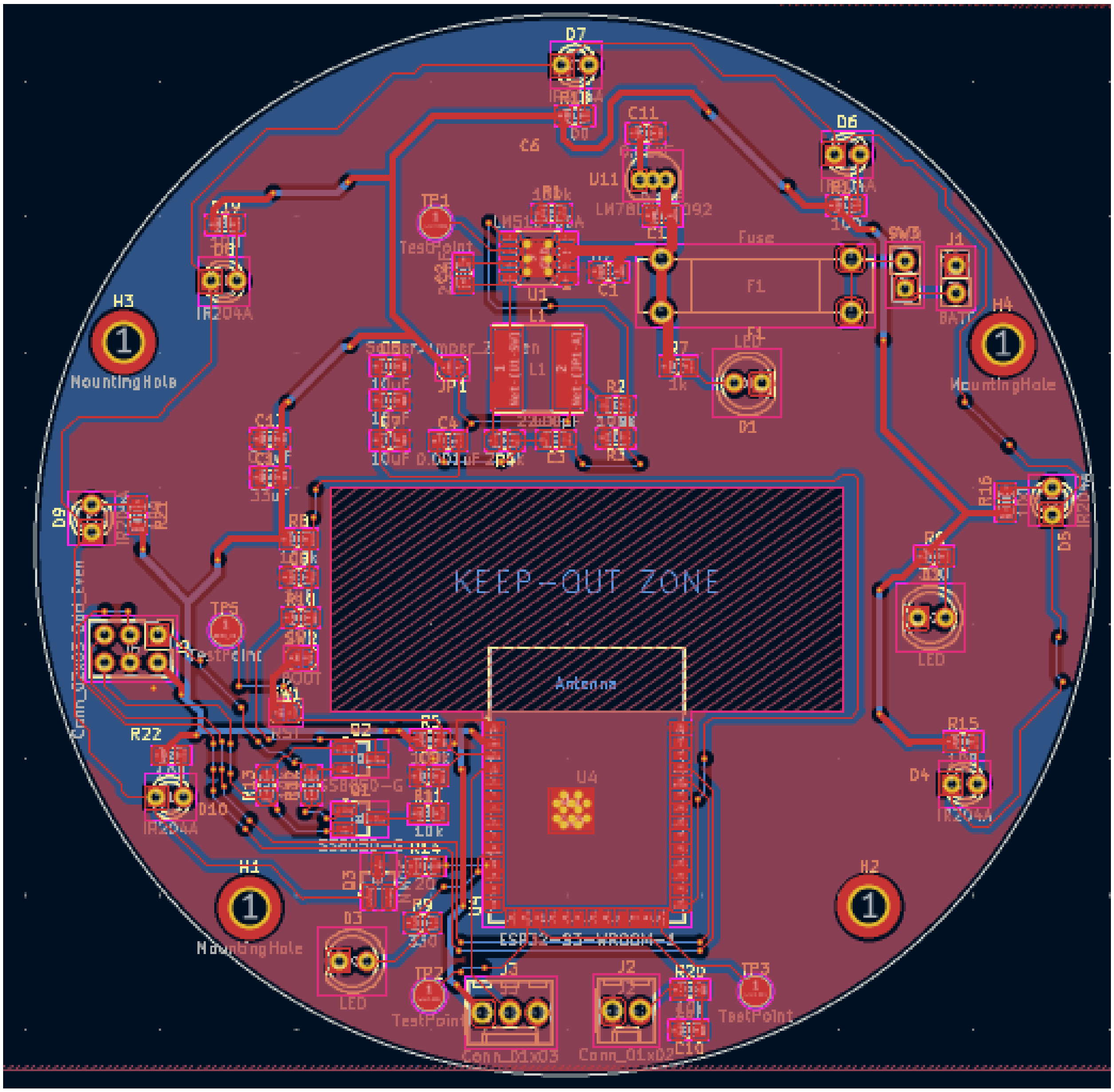

Figure B.3: Coaster PCB

Appendix C Requirements and Verification Tables

Table C.1: Detection System Requirements and Verification

Requirement	**Verification**
The bumper system shall be able to detect a collision anywhere on the front surface of the robot and determine which side (front left or front right).	Place objects in the path of the robot and ensure that it stops and paths around the object.
The cliff detection system shall detect any drop greater than 1in.	Place a coaster outside of the table. Send a request and ensure the robot does not fall off the table while attempting to reach the coaster.
The IR detector shall determine the intensity of the IR light received.	Place a coaster to the front left/right of the robot and check that the robot turns toward the coaster.

Table C.2: Drive Subsystem Requirements and Verification

Requirement	**Verification**
The drive subsystem shall drive at over 1ft/s unloaded and 0.5ft/s loaded with a cup.	Write test code to drive forward 1ft and backward the same distance. Use a stopwatch and compute speed for both loaded and unloaded cases.

Table C.3: Power Subsystem Requirements and Verification

Requirement	**Verification**
The power system shall provide up to 500mA continuously and 750mA bursts at 3.3V.	Disconnect the buck from the rest of the circuit. Use a resistive load to test maximum current.
The power system shall protect against overcurrent events using a fuse.	Disconnect the buck converter from the robot. Short the 3.3V output to ground and connect power.
The battery shall be replaceable with basic hand tools.	Provide a user with a screwdriver and Allen key. Have the user replace the battery.

Table C.4: Microcontroller Requirements and Verification

Requirement	**Verification**
The microcontroller shall not exceed 80°C.	Use a thermal probe while the robot is running. Ensure max temperature stays under the limit.

Table C.5: Pump Subsystem Requirements and Verification

Requirement	**Verification**
The pump subsystem shall not consume current above the maximum rating of the H-bridge, motor, or battery.	Measure current draw with a multimeter during operation. Verify peak current is within rated limits.
Both pumps must dispense liquid at the same speed.	Operate both pumps simultaneously and collect liquid over a fixed interval. Verify volumes match within tolerance and do not spill.

Table C.6: Docking Subsystem Requirements and Verification

Requirement	**Verification**
When the robot microcontroller wants to find the drink dispenser, it shall drive the IR LED at 38 kHz.	Probe the IR LED control signal with an oscilloscope and confirm a 38 kHz square wave when triggered.

Table C.7: User Subsystem Requirements and Verification

Requirement	**Verification**
The Call and Send buttons shall register a single press with no false triggers.	Rapidly press the button 10 times. If the ESP reports exactly 10 presses, debounce is correct.
Reported mix ratio shall be accurate to ±3%FS after 1-point calibration.	Mechanically check 10 evenly spaced dial positions vs. computed ratio. Pass if error $\leq$ ±3%FS.

Table C.8: Coaster IR Subsystem Requirements and Verification

Requirement	**Verification**
Carrier frequency 38kHz ± 1%; duty cycle 33% ± 5%.	Scope the MOSFET gate waveform for 60 sec. Pass if frequency and duty are within limits and jitter 2%.