Automatic Cocktail Dispenser

By

Benjamin Thuma

Caleb Kong

Carson Van Pelt

Final Report for ECE 445, Senior Design, Spring 2023

Professor: Viktor Gruev

TA: Prannoy Kathiresan

3 May 2023

Project No. 32

Abstract

This report describes the design and functionality of an automatic cocktail dispenser developed for the ECE Senior Design Lab at the University of Illinois. The design uses three solenoid valves, a load cell, and a stirring rod mounted on a gear motor. A user button input sets the weight of liquid dispensed from each valve, and the stirring rod mixes the drink after it is dispensed. This device is able to take ingredients and dispense precise quantities of each, making cocktail mixing easier and more precise for those inexperienced in the craft. After many trials, this device has been tested to dispense within 5% margin of error for each specified liquid quantity, and tests of the stirring mechanism have also shown successful homogenous mixing. This document will describe design steps throughout the project, issues that occurred, and final results proving its successful execution.

Contents

1. Introduction

1.1 Problem

Once people are 21 years old, enjoying unique and flavorful alcoholic beverages can be a fun social activity to do with friends and family. There are thousands of different concoctions that are ready to be tried, but it is up to the consumer to know what they want and how to make it. Going out to a cocktail bar is also an option, but going consistently would hurt financially, as the drinks tend to be very expensive (around $10 to $15 on average)[1]. In addition, purchasing drinks in a social environment where all your faith is in a stranger to mix your drinks poses potential dangers. Some bartenders may be interested in serving high-volume cocktails that consumers would not expect, which could lead to overconsumption. Overall, enjoying drinks in a large social setting where the alcohol volume content can vary is a slippery slope to impaired judgment and serious accidents.

Even when making drinks in their own home, most consumers are not skilled bartenders. This can also lead to similar issues discussed before such as adding more alcohol than intended, or even not enough. Finding the right balance of ingredients can be bothersome and dissatisfying when done incorrectly. High alcohol intake, cost, and consumer dissatisfaction are all key issues that can be avoided when using our automatic cocktail dispenser.

1.2 Solution

Our solution was to create an automated cocktail dispenser, one that can eliminate cost and complexity by introducing standardized ingredient portions, as well as providing different cocktail choices. The user is able to choose a cocktail from an online menu, then place the proper ingredients in the appropriate containers. Then, the user can select how much alcohol they want in their drink, and the correct quantities will be dispensed into a cup and mixed with a motorized stirring rod.

1.3 Visual Aids

Shown below in Figure 1 is the physical design of the automatic drink dispenser. The User Interaction panel can be seen on the left with the scannable QR code in the center. The drink ingredient containers are placed above the machine, and the solenoid valves are below each

container. Lastly, the stirring rod is placed directly above the cup and is enhanced with a spring ensuring flexibility to allow easy removal of the dispensed drink.

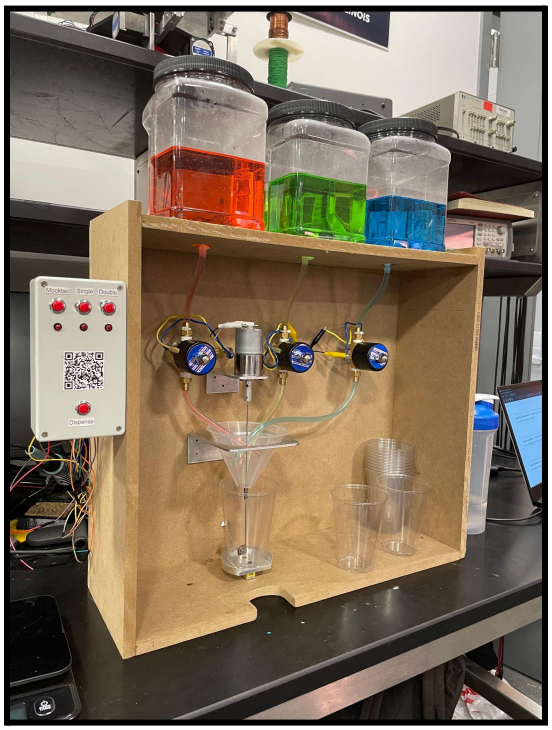

Figure 1: Physical Design

1.4 High-level Requirements

1. The product must be able to dispense the user's chosen drink with no selection error - normal and double-shot cocktails have ingredients from all three containers, and the mocktail only has liquid from one container.
2. The product must dispense the correct quantity of each ingredient for each respective cocktail within ~10% accuracy, measured by mass (e.g. +/- 4.1g for 41mL, one shot, of water). The baseline masses for: mocktail is 205g (0,0,205), single shot cocktail is 205g (41,82,205), and double shot cocktail is 205g (82,164,205).
3. The stirring rod properly mixes the drinks after a successful dispense (test with red, green, and blue food coloring water in each container, to see if the final cocktail color is a uniform brown).

2. Design

2.1 Block Diagram

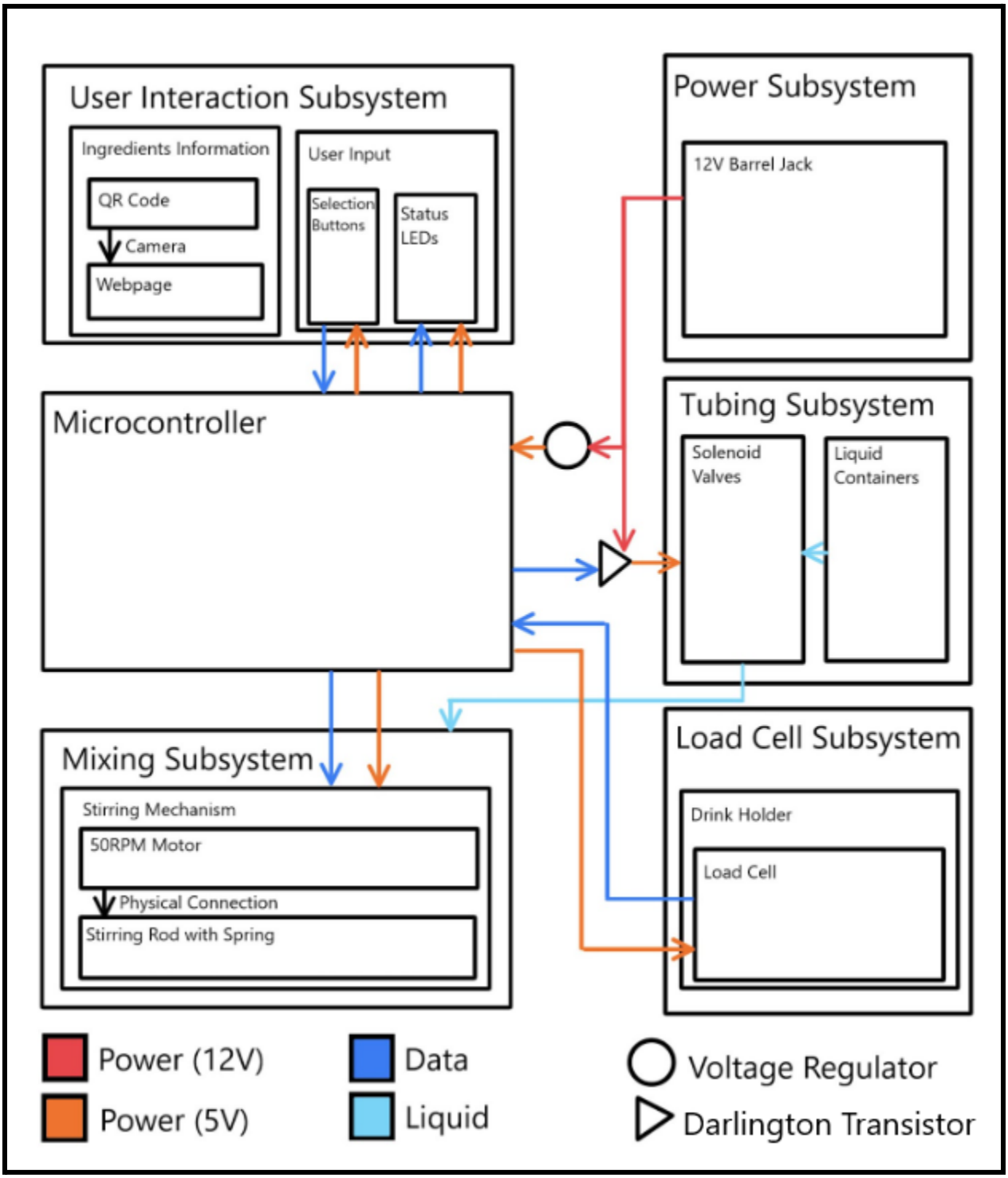

Figure 2: Block Diagram

2.2 User Interaction Subsystem

The user interaction subsystem encompasses all elements of the device that allow the user to select and learn about their drinks. The main on-device portion of this subsystem consists of three potency selection buttons and their corresponding LEDs, as well as a separate dispense button. Also included in this subsystem is a scannable QR code, which connects the user to an online menu that recommends drinks and ingredients, as well as providing information regarding their alcohol content and a picture of all drinks themselves. As we only have three ingredients containers on our device, we decided that the best way to give the user a choice regarding their drink would be the potency, so the potency select buttons are given as mocktail (no alcohol), single (standard alcohol content), and double (double the standard alcohol content). In our initial plans, we had imagined this as a touch screen where different drinks altogether could be selected. However, as the machine shop limited us to three ingredients, we felt buttons and a separate menu would be much more cost-effective as a full interface does not make much sense for so few options. The final application of our user interface can be seen below in Figure 3.

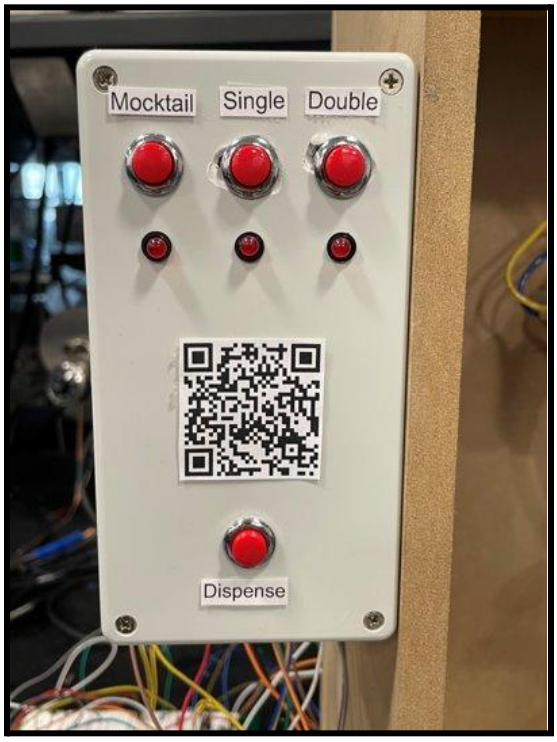

Figure 3: User Interaction Subsystem Application

2.3 Power Subsystem

The Automatic Cocktail Dispenser receives power through a 12V 1A wall adapter that can be connected via a barrel jack to the rest of the circuit. The 12V is regulated via a +5 voltage converter (LM7805ACT) with a 0.33µF capacitor at the input and a 0.1µF capacitor at the output, providing stable results to the rest of the circuit. The electric solenoids and gear motor operate at 12V allowing them to be connected directly to the 12V output of our barrel jack. Our microcontroller operates at 5V, so it takes the outputted +5V from the regulator, fitting within its specified operating voltage of 1.8V to 5.5V. Different from our original design, we have changed to wall power rather than batteries as we did not want to risk the inconsistency that batteries could put on our circuit. Batteries could possibly lead to data loss in the load cell or issues with the valves dispensing the ingredients and wall power prevents any of these issues from occurring. In a different implementation, 5V motors and solenoids could be used as well as the microcontroller, avoiding the need for regulation. Though, we felt our current implementation, seen in Figure 4, works perfectly for our current design.

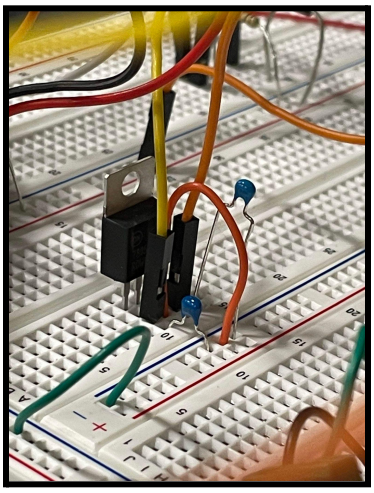

Figure 4: Power Subsystem Application

2.4 Microcontroller

The microcontroller subsystem consists of our dedicated chip, an ATMega328PU, and the input and output connections it provides to the rest of our blocks. The microcontroller takes input from the user interaction and load cell subsystems and outputs to the stirring and tubing subsystems. We chose this chip specifically as it has many input and output pins and each device was able to use its own dedicated pin for input and output control. If a revision were to occur with more inputs or outputs, a multiplexer could be used, however, for our current design, we felt

dedicated pins gave us a better sense of security with less room for failure. On top of its good pin layout, this chip also provided us with a substantial addressing space of 32KB, giving us more than enough room for any code we may write. The general pin layout of our microcontroller can be seen below in Figure 5, with its programming header listed as the AVR-ISP-6.

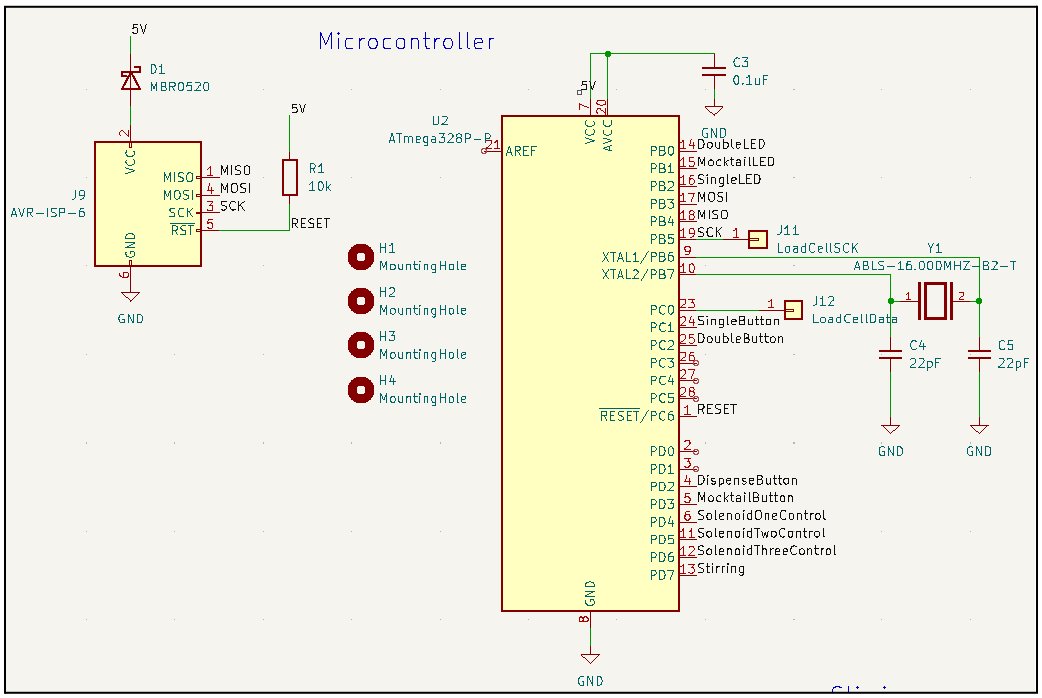

Figure 5: Microcontroller Pin Layout

2.5 Tubing Subsystem

The electric solenoid valves are controlled via the microcontroller and Darlington transistors. With a current-limiting resistor going into the base, we used a TIP122 to act as a switch for the solenoids which the microcontroller can open and close. It can handle the necessary voltages and high currents the circuit requires to ensure fast speeds and efficient switching. In addition, the TIP122 is able to take small currents provided by the ATMEGA328PU and amplify them enough for high-current devices like the solenoids. This effect happens because a Darlington transistor is built of two BJTs with the emitter of the first connected to the base of the second. The collectors of both transistors are tied together and this configuration causes greater current amplification than a single BJT can provide, perfect for our circuit design. Each solenoid is then connected between the collector of the TIP122 and the 12V power supply. Once the Darlington transistor is given sufficient current, the solenoids activate, dispensing the liquids from their respective containers. Additionally in this design is a flyback

diode in parallel with each solenoid, preventing voltage spikes from affecting the rest of the circuit. Circuit diagrams of the solenoids can be seen below in Figure 6.1 with the physical designs shown in Figure 6.2.

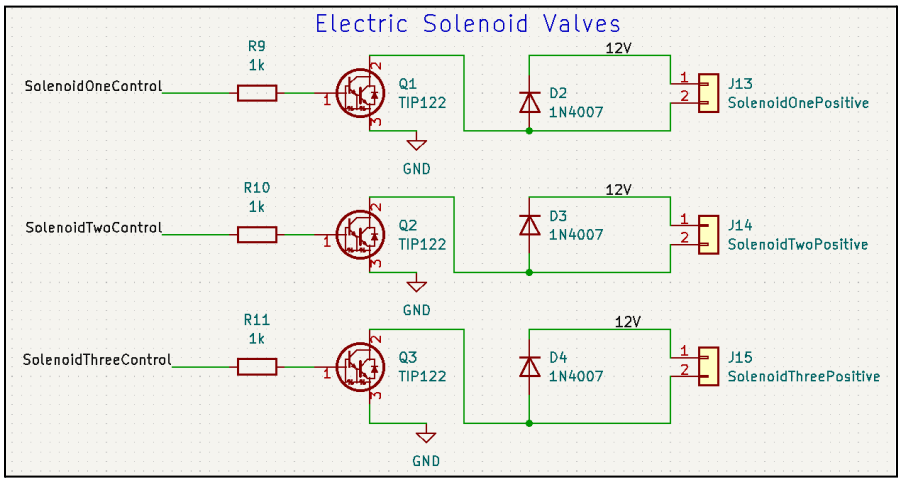

Figure 6.1: Tubing Subsystem Schematic

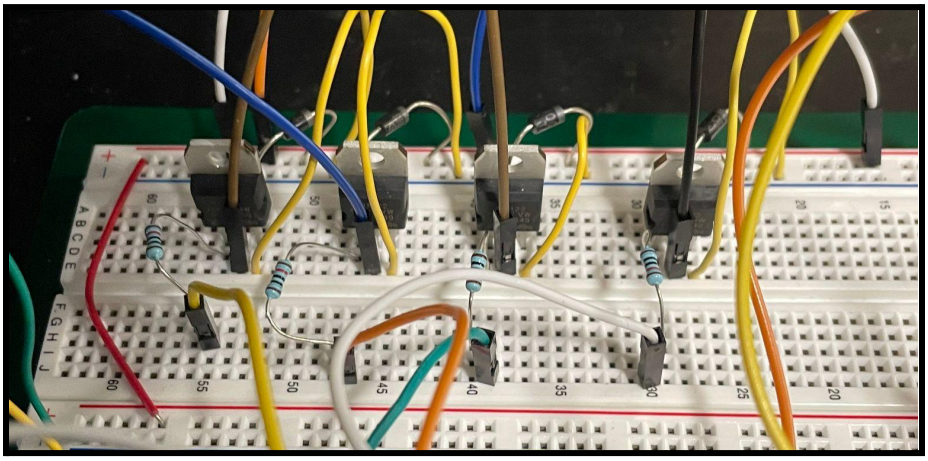

Figure 6.2: Tubing Subsystem Application

2.6 Mixing Subsystem

The mixing subsystem shares the same implementation as the tubing subsystem, being made up of a TIP122, a flyback diode, and a current-limiting resistor. Using the same implementation, the overall circuit is simplified while also minimizing the number of different parts required throughout the project. The main difference of this subsystem in comparison to only comes from is that it would only be utilized at the very end of the dispense when all the solenoids are closed. The mixing subsystem is responsible for stirring the container full of liquid once all the ingredients have been dispensed. Once the ingredients have been dispensed, the

microcontroller will power the 50 RPM gear motor with a rod attached to it and begin spinning for a predetermined set of time and then cease operation. The motor is rated at 12V and 600mA of current which is well within the boundaries of this design.

2.7 Load Cell Subsystem

The load cell subsystem consists of our Bolsen Tech 1KG load cell as well as an HX711 amplifier, seen in Figures 7.1 and 7.2, respectively. While fairly simple overall, this system provides all input to our microcontroller while dispensing is occurring, so its proper functionality and accuracy are paramount to the overall success of the project. The load cell itself is connected to a drink holding plate and its output pins are wired directly into the HX711 amplifier. This amplifier is standard among load cells and provides a single analog output which reports the voltage difference between the positive and negative terminals of a given load cell. This analog output is connected to one of the many analog inputs of our microcontroller and taring of the device occurs during the initial startup of the device, pre-dispense and post-dispensed, once a drink is removed from the plate. We had initially considered changing load cells, as this unit did not provide a strong data sheet, however, as we found it fairly accurate for our needs, and its size was perfect for our implementation, we decided to stay with this choice. For future revisions, we would most likely look for a similarly sized load cell with more in-depth documentation.

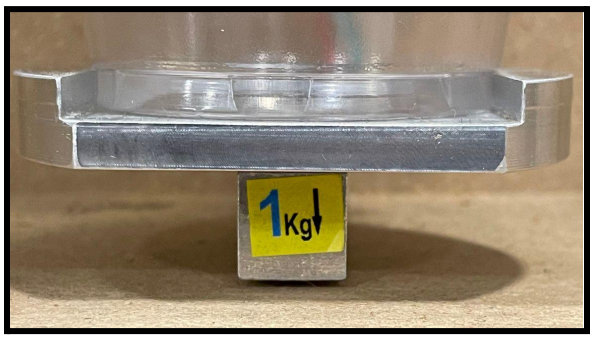

Figure 7.1: Load Cell with Drink Plate

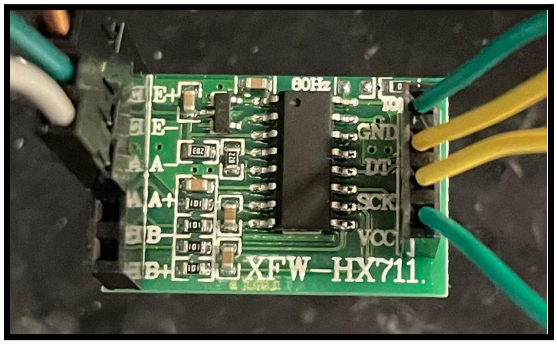

Figure 7.2: HX711 Amplifier

2.8 PCB Layout

When laying out the components of the PCB, we aimed for the connectors to be around the outside of the board for ease of access. The microcontroller was placed in the center to provide simple routing between its input and output devices. We also made sure to put specific subsystems in the same area so that it would be easier to recognize them and solder parts together as a group. For example, both the solenoid and stirring subsystems have been placed in the bottom left. They both use the same components, so it is easily recognizable when trying to figure out which parts go where. We also tried to leave a reasonable amount of space between each component to make soldering easier overall. The current PCB layout can be seen below in Figure 8.

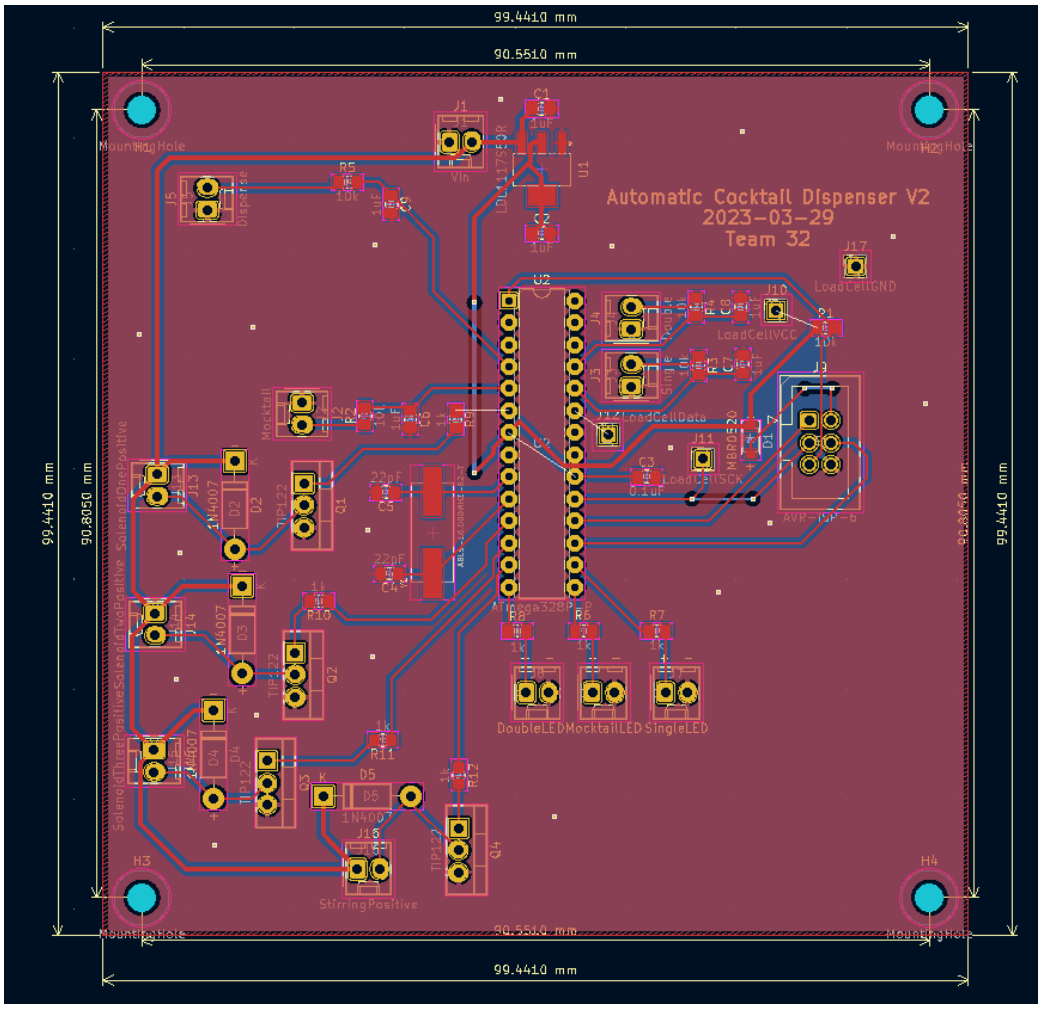

Figure 8: PCB Layout

3. Results and Verifications

3.1 User Interaction Subsystem

With any device that has user input, it is important to communicate the user's selection in some form. For our project, an LED is lit according to a drink potency selection. In the code, pressing a button will also select the mass dispensed from each solenoid, which the load cell will use. This was verified by measuring the liquid dispensed from each solenoid separately, and adjusting the constants in the code accordingly. See Section 3.3 for more details. For simultaneous button inputs of drink potency, multiple LEDs will light up, but only one LED will be on after the buttons are released, meaning only one drink selection can be made at a time. These LED results were verified visually, as after pressing any button the corresponding LED would be lit. Finally, the QR code was tested with a phone camera which successfully links to our webpage, loading a full drink menu with correct aspect ratios on any given device.

3.2 Power Subsystem

To test each element of the power subsystem, it was essential to first confirm that each element worked individually. We used the power supply to provide 12 V to the electric solenoids and gear motor. The solenoids opened, and the motor turned on, spinning at 50 RPM. Using a multimeter, we found that the LM7805ACT voltage regulator stepped the voltage down from 12.2 V to 5.01 V. This 5 V was needed to power the microcontroller and load cell, so having a functioning voltage regulator was necessary for a functioning project. See Table 5 in Appendix A for each requirement.

3.3 Microcontroller

Given that the microcontroller is arguably the most crucial factor in the overall success of our project, it was critical that each requirement for our ATMEGA328PU chip was met. To start, we needed to ensure we were receiving consistent readings from our load cell sensor without data loss. This was verified by observing the serial monitor on an Arduino using the same microcontroller as our project. After running the test code and observing the output, we viewed consistent readings from our load cell, implying the connection between it and the microcontroller was stable. This test ensured data readings from liquid quantities would be successfully read, enabling us to continue with more practical testing.

To then test the physical dispense quantities of our device, we initially ran three separate trials. Within each of these trials, we measured each step of the dispense cycle for each drink selection by adding stopping points in the underlying code. The initial values were expected to dispense baseline weights of 41g per shot with each cocktail having a total weight of 205g. Measuring these with a kitchen scale, however, seen below in Table 1, there was definitely room for improvement, especially for the "Single" user input. The first number in each cell represents how much liquid was dispensed, on average, over three trials, and the number in parenthesis depicts how much this quantity varied from the expected output. The coded values can be seen next to the selection names, listed as (first liquid weight, first and second liquid weight, and total weight).

AVERAGE	Solenoid 1	Solenoid 2	Solenoid 3
Mocktail (0,0,205)	0	0	203 (-2)
Single (41,82,205)	46.33 (+5.33)	85.33 (+3.33)	202 (-3)
Double (82,164,205)	85.67 (+3.67)	163.33 (-0.67)	203 (-2)

Table 1: Average of Initial Dispense Trials (3)

After analyzing these results, the code was adjusted accordingly for each solenoid and for each user input. Specifically, the values for solenoid 3 were increased to account for the under pouring and the values for solenoid 1 were decreased to account for over pouring. Solenoid 2 needed more catered adjustments, as it over and under-poured in different selections. We think the main reason for these variations was the physical distance between the funnel and current waterline of the drink. With a longer drop distance to the water, weight spikes can occur in the load cell ending the cycle before dispensing is fully completed. The newly adjusted code values, with these data points in mind, can be seen in the table below with their corresponding weights.

NEW	Solenoid 1	Solenoid 2	Solenoid 3
Mocktail (0,0,207)	0	0	205
Single (34.5, 78.5, 207)	41	82	205

Double (78,164.5,207)	82	164	205

Table 2: Updated Dispense Trial

As seen in the table above, after adjusting coded values, each dispense is now extremely accurate, properly reflecting the intended portions. With this complete, overflow protection was then added by adding a final blocking statement to the end of the dispense code block. Unlike other blocking statements used in our code, waiting for increased weight, this statement instead waited on a significant decrease in weight, implying the cup had been removed. After then testing many button inputs and selection with a full cup on the plate, we confirmed that this safety feature worked as intended and unwanted overflows would not occur.

3.4 Tubing Subsystem

The tubing subsystem itself is relatively simple - the difficult part was getting the control signals to operate the subsystem. Through the code we wrote for the microcontroller, we were able to ensure the solenoids dispensed serially. Other requirements we had were leak-free and stable tubing. When dispensing, there was minimal movement and no leakage, throughout the entire course of testing our project (upwards of 100 test runs). The tubing subsystem was fully functional.

3.5 Mixing Subsystem

Verification of the mixing subsystem was all about the stirring rod and gear motor operation. Most of the requirements are visual/physical, such as the stirring rod not interfering with the funnel it sticks through, not disturbing the balance of the cup, and fully mixing the cocktail. See Table 8 for more details on the requirements. All the requirements were verified - the stirring rod does not interfere with any other component and homogenizes the drink.

3.6 Load Cell Subsystem

The load cell, being the only sensor in the whole project, was key to dispensing the correct amount of liquid. For starters, we checked the input and output resistance of the load cell, to ensure there were no manufacturing defects. The input and output resistances were 0.999 kΩ,

which is well within 5% of the datasheet specified 1 kΩ. In order to measure mass correctly, we had to first calibrate the load cell, so we used a default calibration program given in the HX711 library in the Arduino IDE. However, the calibration constant that was returned was somewhat variable. In the end, we had to take an average of many calibration constants to use. This allowed us to measure the mass of objects correctly, within a 5% error. Finally, the load cell needed to operate correctly at 5 V, since that is what we powered it with. This was also measured and verified.

4. Costs

4.1 Parts

Description	Manufacturer	Quantity	Extended Price	Link
Bolsen Digital Load Cell Weight Sensor 1KG/5kg/10kg/20kg Portable Electronic Kitchen Scale + HX711 Weighing Sensors Ad Module for (1kg)	Bolsen Tech	1	$8.29	Link
HFS (R) 12v Dc Electric Solenoid Valve Water Air Gas, Fuels N/c - 1/4IN NPT Available (12V DC 1/4IN NPT)	HFS (Hardware Factory Store)	3	$41.13	Link
Greartisan DC 12V 50RPM Gear Motor High Torque Electric Micro Speed Reduction Geared Motor Centric Output Shaft 37mm Diameter Gearbox	Geartisan	1	$14.99	Link
Beer Nuts Bar Mix Containers	Beer Nuts	3	$26.37	Link
Solo Clear 10 Ounce Plastic Cups, 36 Count	Solo	1	$10.99	Link
RP3502MARED Push Button	E-Switch	4	$14.52	Link
5100H1 LED	Visual Communications Company - VCC	3	$8.01	Link
IC REG LINEAR 5V 1.2A SOT223	STMicroelectronics	1	$1.16	Link
LM7805ACT +5 Voltage Regulator	Onsemi	1	$0.63	Link
1N4002 100VOLT 1AMP RECTIFIER DIODE	Onsemi	4	$0.60	Link
NPN BIPOLAR 100 VOLT 5AMP POWER TIP122	STMicroelectronics	4	$2.56	Link
ATMEGA328P-P	Super Electronics	10	$20.46	Link

1N5819 40V SCHOTTKY DIODE	STMicroelectronics	1	$0.38	Link

Table 3: Component Costs

4.2 Labor

Assuming an hourly rate of $37 per hour for each partner, $37/hour x 2.5 x 150 hours = $13875 each. Totaling for each partner, the labor cost comes out to $41,625. In terms of overhead costs, electricity costs 13 cents/kWh and the time of operation per day could be approximated to 30 min per day. Using this, energy consumption comes out to 0.012 kWh which is $0.00156 per day. Assuming the design is used three times a week for a year, the total overhead costs come out to be $0.24. As seen in the figure above, the total cost in parts before tax ends up being $150.09. Assuming a sales tax of 6.25%, the total cost of components equals $159.47. The total of parts, labor, and overhead costs come out to be $41,784.71.

5. Conclusion

5.1 Accomplishments and Challenges

Since our project was successful, all of our high-level requirements (Section 1.4) were met. Being able to dispense the user's drink with no selection error (Requirement 1) means that we have proper solenoid control based on button inputs, so the user interaction subsystem communicates with the microcontroller correctly. To dispense the correct quantity of each ingredient (Requirement 2) means we need the load cell to be able to control the solenoid valves, which in turn implies that we were able to read and interpret the load cell data accurately. At the same time, this means that the voltage regulator worked, because the load cell, which operates at 5 V, was able to communicate with the solenoid valves, which operate at 12 V. Finally, our stirring rod was able to mix the drink uniformly (Requirement 3) after a drink was dispensed, which we can verify visually.

We also added some additional features to ensure safety and enhance the user experience, including a visual "dispensing" signal, overflow prevention, and default drink selection. For overflow, we used a blocking mechanism in the code. After a successful dispense and stir, the microcontroller polls the load cell until the cup is removed. If the cup is not removed, the machine is left in the dispensing state, where all the LEDs are turned on and the buttons are unresponsive. Once the drink is taken, however, the machine will reset and default to a mocktail drink selection. See Section 5.3 for more info about the default state.

5.2 Key Takeaways

The main takeaways from our project come in the form of time management improvements, more realistic overall expectations, and the benefits of thoughtful rather than quick decisions. Regarding time management, one of the main things we learned is that all calendars should have room for unexpected setbacks. One large setback of our project, for example, was a malfunctioning PCB. As we only allocated a small amount of time for PCB debugging, we were unable to identify the exact issue with our board in the allotted time. This resulted in us breadboarding our circuit instead, losing points for a functional PCB. If we had allotted an extra week or two in our timeline for setbacks, we most likely would have been able to solve this issue.

Continuing with realistic expectations, this was something we mainly learned from the machine shop. After introducing our original design to them, a much more complex mixer, the machine shop instantly denied the project. We were told we expected to house nearly five times the amount of ingredients the machine shop could do at a maximum and that our original plan for pumps would be far over budget. After redesigning our project, with these more realistic expectations in mind, we came up with our final design which made for a much more reasonable semester of work for both us and the machine shop.

Finally, regarding thoughtful decisions, one of the main issues that occurred in our project was ordering parts too early. Initially, while we were clamoring to get started on our project, we ordered many parts we were certain would be used in the final design. Though many of these were, we also found that we ended up wasting $30-$40 on unused parts. This is something that could've been avoided with more thoughtful decisions and careful planning. This project has helped us better understand design, time management, and decision-making as a whole, and will prove greatly beneficial to our future lives in the workplace.

5.3 Ethical Considerations

Our project does not pose significant ethical or safety concerns, though it is important to note that, while dealing with liquids, we must ensure that they will not directly interact with any circuitry. A mix of the two could cause a short circuit leading to an electrical shock or fire, which is a significant health risk to the device user. To avoid these concerns, all circuitry will be mounted away from any dispensing portions of the unit, and buttons, which may be mounted closer, will be ensured to be waterproof.

On a separate note, as the recommended ingredients on our webpage include alcoholic substances, it is crucial that we notify the user of the health risks and laws regarding the consumption of the recommended product. In order to uphold the standards of the IEEE Code of Ethics, which includes “[disclosing] factors that might endanger the public” and “[avoiding] unlawful conduct” [2] we have ensured that standard surgeon general warnings are included on our website in an obvious location at the bottom of the webpage. Furthermore, to address the issue of underage drinking, we have the preset drink option set to “mocktail.” As soon as the machine turns on, if the dispense button is pressed, the machine will pour a mocktail. After any

drink is selected and dispensed, the next drink (if no further selection is made), will also be a mocktail. This way, the user must make a conscious decision to pour and consume an alcoholic beverage.

5.4 Future Work

For future work on the project, the first modification we would like to make is adding a retractable stirring rod with a higher RPM motor. This would allow for easier removal of the cup after a dispensing cycle, while also better stirring any given drink; 50 RPM could be a bit slow at times. For better sanitary conditions, we would also like to make the stirring rod and tubing easily removable and replaceable. With this change, cleaning any part and ensuring the tubing remains hygienic, even after changing the current drink ingredients would be much easier. Finally, in a full redesign, we would like a touch display listing all drinks available that would modify the expected densities of the load cell depending on dispensed ingredients. Along with this, adding around two or three more holding containers could allow for many more drink options, giving the user more ease in both selecting their drink and finding one that they would enjoy.

References

[1]https://home.binwise.com/blog/alcohol-pricing#:~:text=Most%20restaurants%20are%20aiming%20for,is%20between%20%245%20and%20%2415.

[2]https://www.ieee.org/about/corporate/governance/p7-8.html

Appendix A: Requirements and Verification Tables

User Interaction Subsystem	
Requirements	**Verification**
• Buttons can select drink potency.	• After the user clicks a selection button, the corresponding LED will be lit and the proper alcohol content should be dispensed through the tubing system. We will confirm this by using the mocktail as a baseline which should not activate the alcohol solenoids at all and have an approximate weight of 135g. For a single-shot cocktail, the solenoids should activate, adding ~90g overall, and a double-shot cocktail should add ~180g to the total weight, with a margin of error of 10% for the total weight. **Verified!**
• QR Code can be scanned and the website is accessible.	• After the user scans the QR code with a compatible device, they will be sent to our website. There should be no errors displayed to the user (404, 500, etc.) and the content of the website should be legible on any device. **Verified!**
• Only a single LED is lit at a time so more than a single option can not be selected.	• If we select multiple buttons at once, the lesser alcohol content will be selected, and only that LED will be lit.

	If a selection button and the dispense button are selected at the same time, the selection will be made but the device will not dispense unless the dispense button is selected on its own. **Verified!**
• The LED indicator is lit up according to the correct button press.	• When a user chooses one of the alcohol content selection buttons, the associated LED should be lit up. This should be nearly instantaneous and will confirm to the user that their selection has been registered on our microcontroller. There will be no LED misfires and one LED should always be lit. **Verified!**

Table 4: User Interaction Subsystem - Requirements & Verification

Power Subsystem	
Requirements	**Verification**
• The electric solenoid valves respond to the supplied power and activate.	• Measuring the voltage across the solenoids should be within 10% of 12V. When supplied power, the valves should open and they should close again when returned to low voltage. **Verified!**
• The gear motor shows a response when supplied with power, and holds	• Measuring the voltage across the motor should be within 10% of 12V.

the correct RPM (50 RPM).	To test rpm, we will attach a rod or pencil to the motor and time it with a stopwatch for 12 seconds. If the motor is successfully reaching 50 RPM, there should be 10 rotations back to the start in that amount of time. **Verified!**
• Microcontroller is supplied with the proper voltage (5V) specified in its spec sheet.	• We will use a multimeter to confirm our current/resistance calculations are correct and the device is being supplied the proper voltage (within 10% of 5V). **Verified!**
• Voltage regulator correctly steps down the voltage from 12V to 5V.	• Using the oscilloscope, we will probe the nodes before and after the regulator on the PCB and ensure the voltages are as expected. **Verified!**

Table 5: Power Subsystem - Requirements & Verification

Microcontroller Subsystem	
Requirements	**Verification**
• Consistently receive data readings from the load cell sensor without any loss	• We will create a test program which uses the indicator LEDs as debug LEDs (or a separate debug LEDs) which will emit light when the load cell is detecting weight. We will continuously add weight to a cup on the load cell and the LED should stay lit the entire time. **Verified!**

• Correctly pick the order of ingredients to be dispensed.	• Solenoids should open in the order we expect and no two solenoids should ever be open at the same time. We can confirm this by viewing the flow of liquid (water dyed with food coloring) through our tubing system. **Verified!**
• Only allow the stirring rod to operate once the liquids are done dispensing	• While any solenoid is open the stirring rod should not be activated. The stirring rod should only activate after either the first (and only) ingredient is fully dispensed from the mocktail or all three ingredients are fully dispensed for either cocktail selection. Fully dispensed should mean no liquid is flowing through the tubing system, so this will require a slight delay in activation even after solenoids are closed. Liquid dripping from the valves is acceptable, but there should be no noticeable flow. **Verified!**
• Choose the correct quantity of liquid for the drink that was selected	• To test that the correct quantity of liquid was dispensed we will precisely measure the three options using our own measuring equipment. By pouring the dispensed drinks into a standard milliliter measuring cup, the three cocktails should fill 135ml, 225ml and 315ml respectively, within a margin of error of 10%. **Verified!**

• Accept user input and correctly recognize the drink selection	• Using a milliliter measuring cup, the button option for mocktail should dispense 135ml, single shot 225ml and double shot 315ml within a 10% margin of error. If any of these are consistently incorrect or flipped it is likely we have mounted the buttons and/or leds incorrectly and we will remedy this. **Verified!**
• The dispense button cannot be selected again when a filled drink resides on the weighing plate.	• This functionality should be limited in our microcontroller code, however, we will test it on the actual machine to confirm. We will dispense a drink and then try multiple options which may lead to failure. We will try removing and replacing the filled drink, changing the potency selection and multiple dispense button presses. As long as the drink resides on the plate, none of these should begin another dispensing cycle. **Verified!**

Table 6: Microcontroller Subsystem - Requirements & Verification

Tubing Subsystem	
Requirements	**Verification**
• Only one tube dispenses at a time, so when one valve is open the rest are closed	• Through simple test code, we can instruct the valves such that each one cannot open unless all others are

	closed as well. Test examples can be run with water as the liquid in order to observe the behavior of the valves and ensure they work properly. **Verified!**
● No leakage from the tubing when liquids are being dispensed	● This can be confirmed through several examples of running liquids through the tubes and observing any holes or gaps that the liquid could pass through. This requirement will be fulfilled once there appears to be no leakage for a good stretch of test examples (say, 20 test runs). **Verified!**
● Tubing is stable and remains firm while ingredients are being dispensed.	● Tubing should not shake or move an unreasonable amount while dispensing liquids. The accepted tolerance will depend on the type of tubing used by the machine shop, and we will verify this by observing movement (or lack thereof) over multiple test trials. **Verified!**

Table 7: Tubing Subsystem - Requirements & Verification

Mixing Subsystem	
Requirements	**Verification**
● The stirring rod starts operation once the last ingredient is fully dispensed (The last valve is closed).	● Stirring rod should not begin activation until all ingredients are fully dispensed. There should be a clear

	indication from the tubing system that all valves are closed before its activation (liquid is no longer leaving the containers) and a slight delay should occur between the closing of solenoids and activation of the motor to allow excess liquid in the tubing to enter the user's cup. **Verified!**
• Stirring rod operates for the correct amount of time.	• Stirring rod should run for the amount of time specified for the drink selection. We will confirm this by timing the activation of the motor with a stopwatch and its timing should remain accurate within 1s. **Verified!**
• Stirring rod does not interfere with the funnel it sticks through as well as the ingredients as they are being dispensed.	• This can be confirmed by running several test examples of dispensing the liquid through our funnel/stirring obstacle and keeping track of how many times there is any spillage. Adjustments can be made to the stirring rod when necessary to ensure the requirement is met. **Verified!**
• Stirring rod does not disturb the balance of the cup while in operation.	• This can be confirmed by having several examples of a sample cup full of liquid being stirred. Varying the speed of the motor and ensuring that 50 RPM is a reasonable rate should confirm this requirement. **Verified!**

• Cocktail is fully mixed	• Since clear cups are being used, one can see the concoction before and after the mix. Water with food coloring will be used to test this, and pictures will be taken before the drink is stirred and after as well. If the drink is the expected homogenous color after the mixing, the requirement is fulfilled. For example, mixing red, blue, and yellow liquid should result in a brown color. **Verified!**

Table 8: Mixing Subsystem - Requirements & Verification

Load Cell Subsystem	
Requirements	**Verification**
• The load cell operates at the specified operating voltage (2.6-5.5 V).	• Apply 5V at the VCC pin and observe whether the load cell reads data at this voltage. **Verified!**
• Accurately provides data to the microcontroller with 5% accuracy.	• With an Arduino, write test code that prints load cell reading to the console. • With objects of known weight, place them on the laid cell sensor and check that the values match. **Verified!**
• Load cell input and output resistance is within 5% accuracy of specified calibration in the datasheet.	• A load cell has 4 wires, two sets of positive and negative terminals. Connect an ohmmeter to measure the resistance of both pairs of wires (a

	positive + negative pair), and compare the values. **Verified!**

Table 9: Load Cell Subsystem - Requirements & Verification