Automated Bartender

Department of Electrical Engineering and Computer Engineering

University of Central Florida

Dr. Lei Wei, Dr. ChungYong Chan

Group 38

Luis Rodriguez-Sandoval (EE)
Pedro Fernandes (EE)
Somal Surujnarine (EE)
Tommaso Dieorio (EE)

Committee Members:
Dr. Avra Kundu
Dr. Sonali Das
Dr. Suboh Suboh

Table of Contents

1.0 Executive Summary

Depending on who you ask, some people when asked would like to unwind at the end of the day by enjoying a nice drink and not having to put too much thought and effort into making that drink. With the growing demand for innovative solutions in the restaurant food & services industries our senior design group has created an automated bartending platform as the main focal point of our senior design project. This project will aim to innovate on how beverages are traditionally prepared and served as well as enhance the efficiency, precision, and the customer experience of bartending. The Automated Bartender will employ various technologies combined with a robust user-friendly interface to provide a simple and intuitive bartending experience that will streamline bar operations and maintain drink consistency. The Automated Bartender is where software and hardware will be meshed together to create a dispensing unit that will turn a time-consuming and monotonous task and boil it down to simply pressing some buttons on a screen. The Automated Bartender will be a project that will capture the common interest of each group member and is where we will take our common interest of engineering and crafting the perfect cocktail and create a device that will embody all of our common interests. The context of this project is for the user to responsibly enjoy alcoholic beverages in a safe and legal manner therefore this project will adhere to drinking guidelines set by the local and federal government.

The Automated Bartender project will have customizable drink options where the user can specify the drink portions they would like on the touchscreen; if the user would like to select from pre-selected drinks this option will be available via the touchscreen display as well. The Automated Bartender will be catered to individuals with a wide range and preferences. Some users will seek a classic cocktail while some users would like to experiment and create new concoctions to enjoy. This project will cater to these individuals. Along with these drink features a maintenance mode will be implemented to make maintenance and calibration of the Automated Bartender a simple and straightforward task. Maintenance mode will allow the user to troubleshoot the hardware and software systems without having to restart the Automated Bartender.

The Automated Bartender will dispense drinks intended for human consumption and will therefore adhere to the various safety standards discussed in this document. This project will execute these drink options with precision using a peristaltic pump that is created for the main purpose of precise fluid output. Along with the fluid level notifications on the touchscreen display there will be indicator LEDs to show the statuses of the reservoirs at a glance. The

Automated Bartender will be equipped with load cell sensors to keep track of the liquid remaining in its reservoirs and will notify the user when it's time to refill. Additionally, there will be load cell sensors integrated into the dispensing platform to track how much liquid has been dispensed to ensure precise drink portions. The Automated Bartender will operate on wall power creating a semi-portable device that can operate wherever a power outlet is available. This project would not be of any use if it did not have a reliable powersource. Powering the Automated Bartender platform will be a custom built power supply paired with various DC/DC power converters that have been researched and designed to fit the power needs of the automated bartending system and will allow for ease of upgrading and adding new systems and features in the future. This custom power delivery system will serve as the heartbeat of the Automated Bartender platform.

This document will be used to capture the design methodology of the Automated Bartender including the critical technologies and the justification behind the design choices made during this project. The description, motivations, house of quality and engineering requirements will be described for the Automated Bartender at the beginning of this document along with the various critical technologies. The core components and technologies will be researched, discussed, and chosen for this project along with various tables and illustrations to help justify the design choices made. Real-world design constraints, safety standards, and sociopolitical factors surrounding the project will be discussed and how they will affect the development of the Automated Bartender will be described and taken into account. High-level schematics and diagrams showcasing the various subsystems of the Automated Bartender and how they are connected will be provided along with low-level schematics highlighting the various technologies each subsystem is composed of. Software design attributes of the project will be discussed along with the corresponding flowcharts to illustrate and highlight the processes and decision making of the Automated Bartender. Detailed steps regarding the testing of the various subsystems will be described and detailed discussions and illustrations regarding the hardware and software design will be included. Details about the fabrication plan, design and the layout of the PCB will be included along with illustrations showing the design and layout of the PCB. Administrative aspects of this project including the build of materials (BOM), budget estimates, project milestones and the work distribution of this project will be discussed towards the end of the document prior to final conclusion of the project. Appendices including a list of the various figures and tables and the works cited pages will be provided as well.

2.0 Project Description

2.1 Project Motivation and Goals

Senior Design is an opportunity to fully realize an idea from conception to production, with the backing of other qualified engineers for collaboration and technical expertise and a very important grade for a drive to completion. It is the quintessential display of our understanding and prowess as students. Therefore, we have decided to create a device that shows the interests of each member, one that will offer a service to both us and our committee come the time of examination. We will brainstorm, prototype, construct, and demo our very own Automated Bartender.

All members of this group find bartending a spectacular display of finesse and showmanship, a skill largely lacking in circles that are not high-class or inexpensive. Access to a bartender of high skill is barred by the very low financials of most college students, and to learn how to bartend at a very high skill is a feat sunken with time and failure an engineering student cannot spare. It is these obstacles that drive us, Group 38, to create a machine that emulates the skill and finesse of an experienced bartender at a fraction of the cost, time, and tears.

There are some previous senior design groups that have done similar projects. One of the groups we have taken inspiration from was the Drink Wizard, which was done in the Summer of 2015 this group made an Automated Bartender system that dispenses various drink combinations based on the user's input. This system uses an ultrasonic sensor to detect if a cup has been placed in the proper spot to allow dispensing to begin and to detect how full the cup is. Our group is using this as inspiration on how our system is going to track the amount of liquid in the cup. Instead of using an ultrasonic sensor, our group will employ the use of a pressure sensor to track the amount of liquid being dispensed into the cup to prevent overflowing.

Another related project that we have taken inspiration from is the Under the Sun Drink Mixer, completed in the Fall of 2013. This group created an Automated Bartender system that dispenses various drink combinations. This group made a system that is portable enough to be taken to cookouts and tailgates. The inspiration that our group has taken from this project is that our device will be in a semi-portable form factor that can be taken from place to place with ease and be used if there is a power outlet available.

2.2 Objectives

What we expect to deliver to our committee is a custom-built machine that dispenses drinks at the request of its user on a low-power, easy-to-use human-machine interface. This will be achieved through the implementation of a touchscreen which will allow the user to graphically select the amount of liquid poured accurately from one of four different sources to a maximum of 12 ounces or to select one of many preset drink options. Each liquid will be housed in a separate reservoir, connected to a peristaltic pump for controlled liquid flow, with low-cost pressure sensors to detect the liquid level inside each reservoir as well as different colored LEDs to indicate that liquid level to the administrator/user. The peristaltic pump allows for precise control of the amount of liquid dispensed while maintaining a closed system; that is, liquid never needs to leave the food-safe tubing to run through the pump. Switching between reservoirs should be fast and seamless; to achieve this, solenoid valves will be used to close and open reservoirs according to need. These valves are fast, reliable, and allow for naturally closed options to reduce the risk of potential spillage. All of these peripherals will be controlled with a microcontroller.

What we hope to eventually deliver, through prototyping and revisions, is an experience of true bartending: Showmanship displays from our machine, automated mixing capabilities, and more. These design hopes come from our own wishes, as patrons of bars and coveters of entertainment, that we hope to have fully automated for our convenience and awe. The most important function of any bartender, though, is the ability to deliver a drink at the exact request of the customer, and that is our ultimate goal. Even the slightest over-pour can, over time, cost hundreds of dollars.

Below are our summarized goals and objectives for our project as well as how we intend to achieve them:

Main Goals:

- Build a machine that can offer the same or a better experience as getting a drink made by a bartender, at a significantly reduced cost, time, and skill requirement.
- Provide customers with the option to customize their own drinks and order preset drinks
- Create a product that is as user-friendly as possible, accessible to anyone.

- Create a highly accurate machine that enables users to customize the precise quantities of up to four drink options, with a maximum combined volume of 12 oz for the final product.
- Develop a product that targets both businesses and home users.

Touch Screen System:
- Allow customization of liquid selection and quantity, up to 12 oz. total.
- Enable users to order preset drinks of multiple liquids at set quantities.
- Provide the option to cancel or confirm selections to avoid mis-inputs.

Liquid Dispensing System:
- Enable the use of a single peristaltic pump with tubes connected to various reservoirs to reduce project costs
- Facilitate quick response from reservoirs via opening a solenoid valve attached to each reservoir's tube, optimizing machine performance.
- Control the liquid flow rate and accuracy via pump's rotary motion for throughput
- Protects the system by preventing spillage from the pump to the nozzle.

Feedback System:
- Visually indicate using LEDs the liquid levels in the reservoirs controlled by pressure sensors under each reservoir
- Enhance the visual experience of drink preparation by illuminating during the pouring process

Stretch Goals:
- Develop a mixing mechanism to expand the range of customizable choices available to customers.
- Bartender vs. Customer mode (Professional vs. Layman mode) to trade ease-of-use with responsiveness based on the user's comfortability with the system.
- Creative sprites/animations on touch screen to give user feedback of system's progress.
- Conveyor/push-arm/surface that slides drink toward the user once finished dispensing.

2.3 Project Specifications and Requirements

Requirement	Specification	Description	Priority
Total Premade	≤ 12 seconds	Total time to dispense a drink	High

Dispense Speed		is within parameters	
Liquids	≥ 4 Liquids available	Number of different liquids available to the machine for use at any one given time	Med
Single Dispense Speed	≤ 5 seconds	No one liquid should take more than 5 seconds to dispense within the normal volume required for mixing	High
Dispense Mechanism Size	10” x 10” x 10” or smaller	Dispenser to be placed over glass should be reasonably handheld-sized	Low
Selection Latency	≤ 100 ms	Latency between selection and beginning of dispensing, constraining logic time	Med
Dispense Tolerance	≤ 0.1 oz	Dispenser should be accurate within the specified tolerance	High
Robust Interface/Ease of Use	≤ 1 sub-critical error ; 0 critical errors	Device UI will be easy to use, and software will be able to withstand various user inputs	Med
Power Supply	Operate on 120VAC	Dispenser will operate on power from the wall socket	High
Device weight (without bottles)	≤ 35 Pounds	Minimize the weight of dispenser for the ease of portability	Med

Table 1: Design Specifications

The constraints laid out in the above table are intended as an assurance of the quality function of the designed machine. While there may be other areas of optimization possible for the design, we have identified the above as the most critical areas of optimization to deliver a quality product. These optimization (time-based) constraints are to preserve the speed of bartending as should our machine not function with comparable speed to a human bartender, the technology only complicates the process of drink mixing as opposed to providing an easier alternative.

The demonstrable specifications that will be presented are requirements 1, 2, and 6, as these showcase the critical elements of speed, variety, and accuracy respectively.

2.4 Hardware Diagram

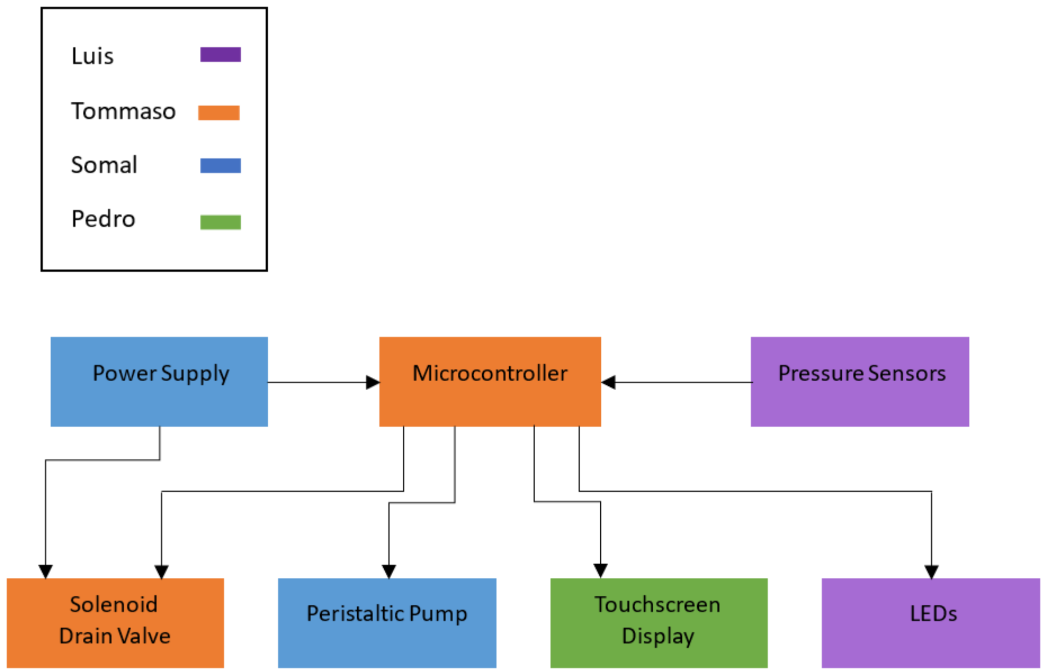


Figure 1: Hardware Block Diagram

Status of each block:

- Power Supply: Researched and designed
- Pressure Sensors: Researched and acquired
- Microcontroller: Researched and acquired
- Solenoid Drain Valve: Researched and acquired
- Peristaltic Pump: Researched and acquired
- Touchscreen Display: Researched and acquired
- LEDs: Acquired

2.5 Software Diagram

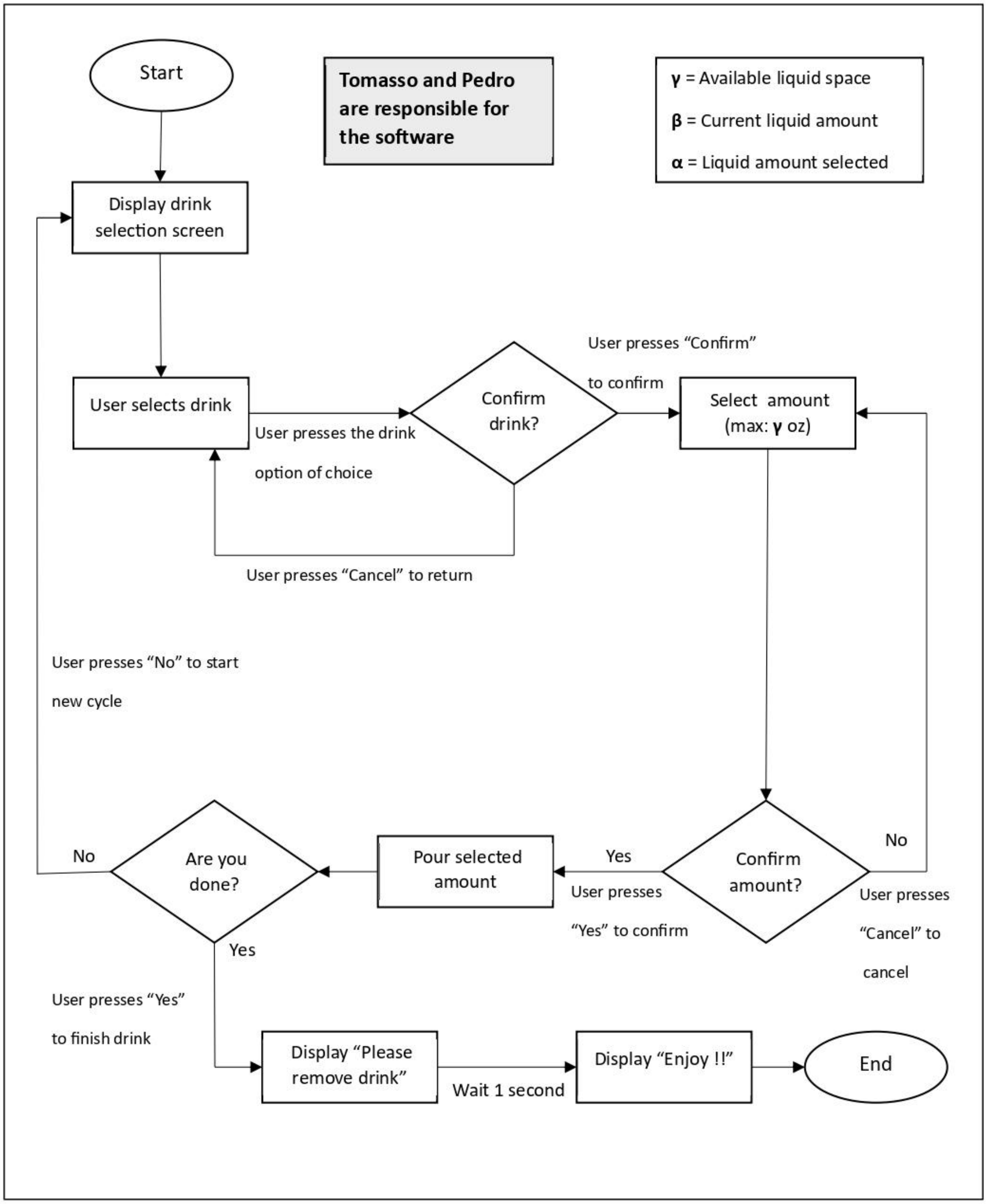
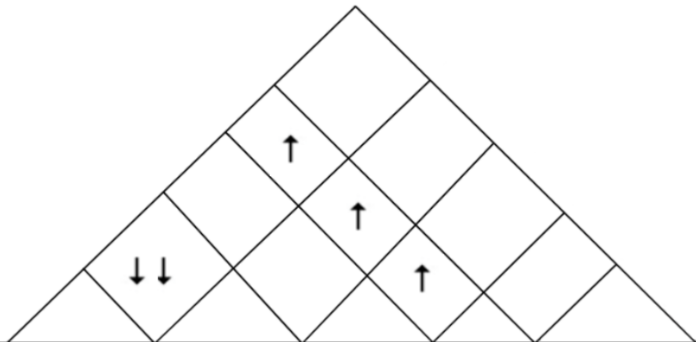


Figure 2: Software Block Diagram

2.6 House of Quality

The below house of quality summarizes the effects that each of the marketing and engineering requirements have on one another. Additionally, it compares how the engineering requirements interact as well.



			Dispense Speed	Accuracy	Liquids Available	Cost	UI Robustness
			+	+	+	-	+
Marketing Requirements	Cost	-	↑	↓	↑	↑↑	
	Ease of Use	+			↑		↑↑
	Overall Speed	+	↑↑	↓↓		↑	
	Indicators Lights	+				↑	↑
	Showmanship	+	↑	↓		↑	
	Target Engineering Requirements		≤5s single ≤12s premade	± 0.1 oz accuracy	≥ 4 liquids	≤ $500	≤ 1 subcritical error No critical

Figure 3: House of Quality Engineering vs Marketing

2.7 Existing Technology

The main idea of an automated drink dispensary has become more prevalent as the culture around drinking continues to relax. With this laid back atmosphere, technology surrounding pouring and mixing drinks for laymans is entering the market in large quantities. These technologies range greatly in similarities to the Automated Bartender, from a light-up coaster to another automatic dispenser.

One very simplistic technology akin to the Automated Bartender is the Barsys Smart Coaster. This coaster uses a weight sensor that measures the amount of liquid poured into it by the user, who is following a recipe from the Barsys mobile app to create their own cocktail. The LED lights within the Smart Coaster change colors to indicate progression and completion of each pour. This device is essentially the planned electronic feedback system, weight sensors and LEDs, of the Automated Bartender retooled into its own package and integrated with a mobile app for smart capabilities.

This technology is lightweight and robust in design, but lacks the automated portion that pours the drink for the user, making this more of a novelty device than a fully-fledged automated dispensary system. Barsys provides a more automated system in the Barsys 2.0+, but the system works primarily using gravity to pour the liquids and the mobile app to dictate the amount to pour. While this is nearly identical to our own device, it includes different technologies to ensure pour amounts, comes at a much steeper price at $1500, and the current Barsys 2.0+ setup does not allow for expansion with more drink options.

Figure 4: The Barsys Smart Coaster (left) and Barsys 2.0+ (right)

More similar to the Automated Bartender is Smart Bar USA’s Smartender Self Serve. This technology provides a self-serving experience, with up to 8 different liquids, ice dispensers, carbonator, and more. Users select what drinks they would like and the Smartender pours the liquid(s) through the same nozzle. This technology is truly an all-in-one system.

The main differences between the Automated Bartender and the Smartender Self Serve is the size. The Smartender Self Serve, in its smallest form, is a 112.25 x 41 x 74.5 inch movable cart. This product is meant to be a full bar replacement: There is no bartop or bartender in the implementation of the Self Serve. Instead, this cart has everything needed for users to serve their own drinks, similar to a standard self-serve drink machine at fast food restaurants.

Where the Smartender Self Serve aims to replace the bar and allow full customer control, the Automated Bartender is intended to supplement the bartender in their role and create drinks on the side to ease the drink queue load, or for simple cocktails to be made at home. The Automated Bartender is designed to have a much smaller package compared to the Smartender Self Serve, which does tradeoff some of the capabilities, such as carbonation and ice dispensers; however, most of these features are already standardized within the barspace, making them redundant for the Automated Bartender's design. While the Smartender Self Serve aims to substitute, the Automated Bartender aims to support.

Figure 5: Smartender Self-Service product

3.0 Research

Component investigation is a critical aspect of any engineering project. When researching components for the Automated Bartender we must find components that will perform as expected at a reasonable price. We have considered a wide range of technologies that will fit our constraints while still meeting our goals. All of these components have been thoroughly researched to ensure they will meet the project needs of the Automated Bartender platform.

3.1 Microcontroller Selection

There are several critical specifications which we consider imperative when selecting the proper microcontroller, those being CPU clock speed, I^2C and SPI connectivity, and memory (both ROM and RAM). CPU speed will ultimately factor into the speed at which the machine can function; for elements such as logic, this time is critical as this task is very easily optimizable with a faster CPU as there are no physical constraints on the algorithms used. However, when interfacing with elements such as the solenoids or peristaltic pumps, there are limiting physical factors such as maximum operation speed which will limit the effect a faster CPU could have on the design. The only foreseeable borderline hard real-time requirements our system would need to meet is the reading of the pressure sensor beneath the cup being filled as, should this device miss its scheduling window, the cup could overflow, resulting in possible damage to the machine or its electronics. Other firm real-time requirements include device feedback and response time on the touchscreen and movement of the motors for the peristaltic pumps.

I^2C connectivity and transmission speed will determine the speed at which the MCU can interface with the various sensors in the design. For the pressure sensor under the cup being dispensed into, this speed becomes rather critical as it is essential for measuring the amount of liquid in the container to compare to its maximum capacity; therefore, to prevent spills, this communication should be fast and efficient. For the other interfacing sensors, speed becomes less important, but the MCU selected must still be able to interface via I^2C. In regard to memory, because the programming is likely not going to be intensive when considering the design, this could be a critical place to decrease the cost of the device. However, it is still critical that the system can operate to its fullest speed and capacity which will likely require more RAM.

Other less critical factors to be considered include the number of I/O pins available for elements such as LED's, the number of channels available for ADC for the pressure sensors, ADC resolution, and ease of software development.

3.1.1 Atmega328p

The first processor we are considering is the Atmega328p. This is by far the least powerful option being presented in our research and would likely be the "bare minimum" processor to achieve our goals. That is, when considering the features of the device, while it is certainly possible to meet the prescribed goals of this project, there are certain elements of the processor which complicate its use, making it the least complex but a more difficult option to implement the features we would like to use to meet our goals.

The microcontroller is equipped with an 16 MHz CPU clock which would be more than adequate for the applications of this project and 3 peripheral clock/counter modules (2 8-bit and 1 16-bit) which can be configured to a low-frequency crystal oscillator to drive the serial interface for the two-wire module. This two-wire module, while adequate for the implementation of I^2C, is not by default an I^2C module. This creates difficulty in its use as the two-wire interface (TWI) that is included on the Atmega, while very similar to an I^2C module, does not meet all the necessary specifications or include all the same features of an I^2C module. While this does not rule out the possibility of the use of the Atmega, this is a serious dissuading factor as I^2C is one of the most important requirements for this project. However, should this processor be used, the TWI can be configured to a 400kHz 7-bit address I^2C module. The other critical module for our project is the Analog to Digital Converter (ADC) module. This processor has a 10-bit resolution 8-channel ADC module which should be more than sufficient for these applications as each pressure sensor will require analog to digital conversion, making the minimum number of required channels five.

The Atmega has 32kB of Flash memory available, 1kB of EEPROM available, and 2kB of SRAM available. While this amount of memory should be adequate for our applications, due to its relatively small size, it could require some optimization of memory allocation and program flow which may cause software implementation to become more complex (ie. the requirement to optimize some sections of code in assembly). While this again does not disqualify the Atmega from use, it does create a lack of expandability in this proof-of-concept design, disauding us from this application.

While the Atmega328p could certainly be used for the purposes of this project, it is likely not the best option. While its cost is very low at $2.89, its drawbacks prevent it from being a good option. The small amount of memory limits the expandability of this project and could prevent some memory-intensive stretch goals, such as creating multiple operating modes, from being possible. Additionally, the unfamiliar TWI module could create issues interfacing with I^2C devices and its different features could make implementation more difficult in software development. Therefore, the Atmega is likely not the best MCU for the purposes of this project.

3.1.2 MSP430FR6989

The second processor in consideration for this project is the MSP430 FR6989. This processor presents a unique balance of power, while solving some of the issues with the Atmega328p. The device has all the necessary features for the applications of this project as well as some promising elements when considering the expandability of this project and the stretch goals we have set forth.

The MSP430 contains a CPU configurable to a clock speed between 8 and 16MHz which again is adequate for the speed applications of this project. The processor also has two timer modules with multiple channels for general use. An I^2C module is included in the eUSCI interface of the chip, configurable for between 100 kbps and 400 kbps with 7 or 10-bit addressing. While one I^2C module is certainly adequate for our applications, a careful look will need to be taken at scheduling of reading from the pressure sensors to take into consideration the high priority with which the sensor beneath the cup needs to be read in comparison to the other bottle status sensors configured using I^2C. This scheduling problem becomes especially important when considering the possibility of expanding the system to be compatible with more than 4 bottles. While it is certainly possible to achieve this with proper scheduling, the question becomes how often it is necessary to read from the under-cup sensor in comparison to the other configured sensors and how this factors into the ADC conversion that will take place. For this proof of concept, however, this I^2C module should be sufficient.

The ADC module included on the MSP430 is much more powerful than the Atmega's, offering a 12-bit resolution and 16-channel ADC module. This again helps with expandability of the project, far surpassing the five necessary channels for the current specifications

The MSP430 is equipped with 128kB FRAM unified memory which is used as both RAM and ROM to store the program, short-term data, and longer-term storage. This amount of memory is enough to accomplish the goals of this project. While the speed of access is a bit slower with FRAM in comparison with other memory types, that is likely not a concern for this project as on the nanosecond-scale, these differences are negligible.

As a whole, the MSP430 FR6989 is a very solid processor for the applications of this project. While there are some features which would hinder some stretch goals and expandability, for the sake of the goals of this proof of concept, the MCU is more than adequate. Some other advantageous features of the MSP430 include the large number of general purpose input/output (GPIO) pins available to the user, which can be helpful in reaching stretch goals such as the implementation of a conveyor system to move the drink toward the user or mixing mechanism for finished drinks to make them more homogenous. Additionally, the development environment created by Texas Instruments for these processors makes programming easier. The MSP430 is a great option for this project and could prove to be an economic option which meets all requirements necessary and exceeds some, making the goals of the project not only achievable, but exceedable.

3.1.3 ESP32-WROOM-32e

The final processor under consideration for this project is the ESP32-WROOM-32e. This processor is far-and-above the most powerful under consideration and would likely far exceed the needs and goals of this project while allowing plenty of overhead for possible stretch goals and expandability of the system. Additionally, the added wifi and bluetooth functionality of this chip allow for further expansion of the horizons of this project.

The ESP32 contains an Xtensia dual-core 32-bit lx6 CPU configurable up to 240 Mhz. This operating frequency is far above the requirements for this project. The dual core introduces new possibilities in the realm of optimization of the project as one core could be used for the human-machine interface, freeing the other core to monitor essential processes such as the fill level of the cup and the proper function of pumps and valves. However, the use of two cores for this project is far from necessary. The processor is equipped with two I^2C modules configurable for between 100 and 400 kbps with 7 or 10-bit addressing. This is adequate for the applications of this project, and the presence of two distinct I^2C modules adds ease to task scheduling as essential tasks can be monitored more frequently or even be configured to their own module to completely negate the possibility of a scheduling conflict causing a critical error.

The processor also includes 2 12-bit resolution 18-channel ADC modules. This again is likely far more channels than necessary for the applications of this project, but, similar to the applications of two I^2C modules, the use of two ADC modules could allow for some interesting and helpful optimization opportunities when considering critical conversion tasks. Critical conversion tasks could be performed by one dedicated ADC module, limiting the number of conversions needing to be done by said module, increasing the sampling rate, while the other module handles subcritical tasks which do not need to be sampled as frequently.

The ESP32 contains 448kB ROM and 520kB RAM. This amount of storage presents almost no necessity for storage optimization in regard to program size or dynamic memory allocation and is far beyond the amount this project likely requires. This amount of memory also aids in the achievement of our stretch goal to create multiple operating modes for different applications (Differing user interfaces and options for bartender operation or customer operation).

As a whole, the ESP32 far exceeds any hard requirements for meeting the goals of this project. All of its features far surpass the necessary features to achieve this project and, as a result, make the achievement of these goals far easier when considering software development. The primary reason for its consideration for this project is its high value as the cost of this processor is only $2.80. When considering the wide range of features available on the ESP32 and the low cost of the MCU, it would make a great option for this project. Most of the extra features and peripherals available on the ESP32 aid in the meeting of stretch goals. The large number of ADC channels aid in the ability to add more bottles, the large memory allows for many different interfacing options to be created, and the dual core with two ADC and I^2C modules create an opportunity for easy scheduling and separating of high-priority tasks from standard-priority tasks to decrease the probability of a critical failure.

3.1.4 Microcontroller Comparison

Feature	Atmega328p	MSP430 FR6989	ESP32-WROOM-32e-N8
CPU Speed	8-16MHz Single core	8-16MHz Single core	Up to 240MHz Dual core

I^2C Module	1 TWI module 400 kbps 7-bit address	1 module 100-400 kbps 7 or 10-bit address	2 modules 100-400 kbps 7 or 10-bit address
ADC Module	1 module 10-bit resolution 8-channel	1 module 12-bit resolution 16-channel	2 modules 12-bit resolution 18-channel
Memory	32kB Flash 1kB EEPROM 2kB SRAM	128kB unified FRAM	448kB ROM 528kB SRAM
GPIO	23	83	24
Cost	$2.89	$10.77	$2.80

Table 2: Microcontroller Comparison

The primary selection criterion in choosing the correct MCU is the value: the chip which gives the most features with the fewest drawbacks. Another important consideration is the necessity of the features. While it is likely unnecessary to use a dual-core processor for this project, the value of the ESP32 when considering the cost, is far above the single-core option of the MSP430. Again, while the Atmega could be applied to this project, its value is far below that of the ESP32. Because of the expandability, power, and ease of software development that the ESP32 offers coupled with its low cost and high value for that cost, we will use the ESP32. We believe that the extra features of the ESP32 greatly widen the possibilities for improvements to this project, and while they are not a necessity, it is logical to have them available even if we do not need them.

3.2 Pumps

There are many different ways to transfer fluids from point A to point B—the most common method of transferring a fluid is by using a pump. Pumps can be driven mechanically or electronically. No matter the method that the pump uses to transfer liquids, they all have the same common goal: to move a fluid from point A to point B. Both the material and size of tubing used in the pumping process play a pivotal role in the pump design and operation. The most common types of pumps used today are centrifugal pumps, double diaphragm pumps, and peristaltic pumps.

3.2.1 Centrifugal Pump

The first common type of pump is the centrifugal pump. These pumps move fluids by using an impeller connected to a drive shaft inside the pump's housing. When the impeller rotates, it will transfer rotational energy from the impeller to the fluid, causing the fluid to move inside the pump chamber. When the impeller is spinning, it will move the fluid via centrifugal forces along the blade of the impeller. As the impeller is moving fluids, the pressure and the velocity of the fluid will increase.

Centrifugal pumps cannot operate efficiently when the liquid is too viscous, and they must be primed to operate efficiently. When the pump is not primed, the negative pressure created from the impeller spinning will not be enough for fluids to be drawn into the eye of the impeller. The biggest downside of using a centrifugal pump is that the fluid being pumped will have to touch the internal components of the pump, which can pose a problem for food safety and sanitation reasons.

3.2.2 Double Diaphragm Pump

The second common type of pump is the double diaphragm pump. These pumps move fluids by utilizing two flexible diaphragms that are controlled by compressed air. When compressed air is applied to the system via a central inlet port system, the diaphragm will reciprocate back and forth, allowing each chamber in the pump to operate independently of the other. When a piston is pushing against the diaphragm, this will cause the fluid contained in the chamber to be expelled into the outlet of the pump. When one diaphragm is compressed, the other diaphragm is relaxed, creating a vacuum that allows the second chamber to be filled up quickly. This constant back-and-forth reciprocation of the diaphragm allows fluids to be pumped efficiently through the mechanism. Each chamber of the pump has two balls (one at the top and one at the bottom of the chamber) that act as a valve. When the chamber is not being compressed by the diaphragm, the ball at the top of the chamber will be pulled down by the vacuum of the empty chamber, allowing fluids to flow into the empty chamber. When the fluid-filled chamber is being compressed by the diaphragm, the ball at the bottom of the chamber will be pushed down, preventing fluids from flowing into the suction chamber. Because of the unique design of the double diaphragm pump, it does not need to be primed before operation. The drawbacks of using this type of pump are that it has a slower flow rate compared to other pumps due to its mechanically heavy design, and the fluids flow out of the pump as pulses instead of a steady stream.

3.2.3 Peristaltic Pump

The third common type of pump is the peristaltic pump; this device is classified as a positive displacement pump. These pumps move fluids through flexible tubing that is constantly being compressed and decompressed by rollers on a rotor attached to the pump. The pumping principle that these devices use to operate is called peristalsis, which can be observed in biological systems. When the rollers compress a region of the tubing, the fluid contained in that region will be forced to flow through the tubing to the outlet side of the device. Regions of the tubing that are decompressed will have a lower pressure, causing a vacuum to draw fluid into the suction side of the pump, as seen in Figure 6.

The constant rotation of the rollers creates a system that pumps fluid through the tubing without the fluid having to come in contact with any components of the pump since the fluid is confined to the tubing. This characteristic of a peristaltic pump makes it a good choice for applications involving consumables. Peristaltic pumps are suitable devices for applications that require precise fluid output. The fluid output depends on revolutions per minute of the rollers, which are dependent on the input voltage of the DC motor and the diameter of the tubing used.

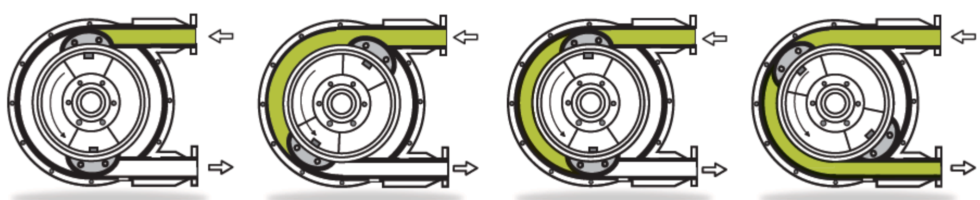


Figure 6: Peristaltic Pump

The design of peristaltic pumps does not come without its imperfections. Due to the constant compression and decompression of the tubing inside the pump. The tubing will wear out over continuous use of the pump. If the tubing is allowed to deteriorate to such an extreme, this can cause a decreased flow rate of the pump and increase the chances of the tubing to start leaking. The design of the peristaltic pump allows the internal tubing to be changed easily. As long as the tubing is changed, this disadvantage of the peristaltic pump can be mitigated. The peristaltic pump in this project will be operated for minutes at a time since there is not a constant operation of the pump, the life expectancy of the tubing inside the pump will be significantly increased.

3.2.4 Pump Selection

The use of a peristaltic pump will be an ideal choice for this project. Peristaltic pumps will be in small form factors and will be relatively cheap. Peristaltic pumps have a significant advantage when compared to the centrifugal and double diaphragm pumps; this advantage is that the flow rate can be easily controlled. The flow rate of the peristaltic pump can be adjusted by changing the voltage applied to the pump's input terminals. The self-priming features of the pump will allow the pump to operate in various environments compared to a pump that needs to be primed before use. When using Peristaltic pumps, the fluid inside of the tubing will not come into contact with the components of the pump; this feature will allow the pump to be implemented into food applications easily. While the other two pumps could theoretically be used in this application the peristaltic pump would be most logical due to its simplicity when compared to the other pumps.

Pump	Size	Complexity	Self-Priming	Cost
Centrifugal	Large	High	No	High
Double Diaphragm	Large	High	Yes	High
Peristaltic	Small	Low	Yes	Low

Table 3: Pump Comparison

The three peristaltic pumps researched for the Automated Bartender are the Kamoer high-flow peristaltic pump, Kamoer KHPP260 peristaltic pump, and the Kamoer KPHM100 peristaltic pump. All of these peristaltic pumps will be evaluated by their flow rates, input voltages, and price.

The first pump is the Kamoer high-flow peristaltic pump; this device interfaces with a brushed DC motor, uses 3 rotors for fluid displacement, and is priced at $40. This pump has a flow rate of 520 ml/min at 12V and a flow rate of 550 ml/min at 24V. The product dimensions are 112 x 66.6 x 91.7 mm. This pump uses a planetary gear transmission with a gear reduction ratio of 1:8 and has a noise rating of ≤ 80 dB.

The Kamoer KHPP260 peristaltic pump, priced at $18, serves as the second pump in our component selection. It operates through a connection to a brushed DC motor, employing three rotors for fluid displacement. This pump delivers a flow rate of 140 ml/min at 12V, which increases to 220 ml/min at 24V. Its product dimensions measure 54.5 x 76.1 x 74 mm, and it maintains a noise rating of ≤ 60 dB.

The third pump is the Kamoer KPHM100 peristaltic pump; this pump is priced at $12 and is interfaced with a brushed DC motor employing three rotors for fluid displacement. This pump has a flow rate of 100 ml/min at 12V and a flow rate of 140 ml/min at 24V. Its product dimensions measure 42.5 x 79 x 65 mm, and it maintains a noise rating of ≤ 60 dB.

Pump	Price	Flow Rate	Size
Kamoer high-flow peristaltic pump	$40	520 ml/min at 12V 550 ml/min at 24V	112 x 66.6 x 91.7 mm
Kamoer KHPP260	$18	140 ml/min at 12V 220 ml/min at 24V	54.5 x 76.1 x 74 mm
Kamoer KPHM100	$12	100 ml/min at 12V 140 ml/min at 24V	42.5 x 79 x 65 mm

Table 4: Peristaltic Pump Specs

The peristaltic pump selected to be used in the Automated Bartender will be the Kamoer KHPP260 peristaltic pump. This pump will provide us with the features we need for this project, and it is the best in terms of performance and cost and it will come in a size that would be easy to implement into this project.

3.3 Tubing

The primary goal of the Automated Bartender is to dispense drinks for user consumption. Since this project involves consumables, the tubing that the liquids flow through must be food-grade, contain no BPA (Bisphenol A), and be taste and odor-free. The tubing used in the Automated Bartender will have to be durable to withstand the constant compression and decompression of the peristaltic pump. There are many different solutions available today; the most common type of tubing for food-grade applications is Tygon tubing. This type of tubing is lightweight, flexible, and clear.

The most common variation of Tygon tubing used in drink applications is the Tygon E-65-F Food & Beverage Dispensing Tubing. This type of food-grade tubing follows the strict guidelines of FDA standards and NSF-51 standards, meaning that it is safe for applications involving food. The price of this tubing will cost about one dollar per foot, and the price will change depending on the size and diameter of the tubing. All the tubing will be connected using barbed tube fittings to ensure a tight seal and reduce the chances of a leak occurring.

3.4 Valve Selection

When considering the selection of the valves to be used throughout the device, the primary considerations include the robustness, longevity, and precision of the valve. The valve type must be able to reliably open and close despite any adverse exterior conditions, must not quickly wear out or degrade, and must open and close quickly to precisely control the amount of liquid released. While ultimately the liquid released will be controlled by the pump cycling, this precision in measurement can be greatly aided by the precision of the valve selected.

3.4.1 Solenoid Valve

The first option under consideration for this project is the solenoid valve. This valve functions utilizing a plunger pushed in place by a spring to plug the junction between two connected tubes. This plunger is controlled via a coil which when energized creates a magnetic field, overcoming the force of the spring, pulling the plunger out of its resting position to change the state of the valve. For the confines of this project, a normally closed valve, meaning the default position of the plunger is closing the connection off, will be considered (illustrated in Figure 7).

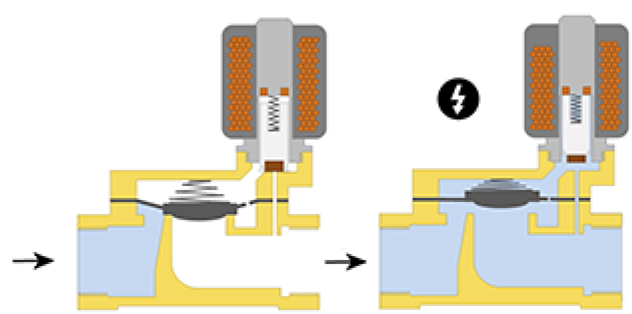


Figure 7: Normally Closed Solenoid Valve

Additionally, this valve has the added advantage of having multi-directional options, as the plunger can be fitted to allow flow past it in a closed state instead of simply blocking all flow.

This type of valve is very robust as it eliminates the majority of mechanical elements involved with fluid flow as the only real mechanical elements are the spring and plunger system. Also, as there are no real opportunities for wear, the longevity of this valve is also superior. Finally, because it is electrically controlled and magnetically opened and closed, the response of the valve is very fast; for a small direct-action valve the average response time is 30ms. The largest possible issue from these valves is power consumption as creating a magnetic field can consume large amounts of power, especially considering that the coil must be powered the entire time the valve is opened.

This could be mitigated by using a latching solenoid valve, which makes use of a permanent magnet to hold the valve open. This would prevent the necessity of using power to keep the valve open, but is not viable for applications with high-frequency operation, thus hurting the robustness of these valves. Another critical consideration is known as water hammering, which is an effect where the rapid closing of a valve results in waves in the reverse direction of prior flow. Because of the fast operation of these valves, this becomes a critical concern and could create pressure differential issues in the system.

3.4.2 Pinch Valve

The next option being considered for this project is a pinch valve. This valve functions by passing fluid through a malleable envelope which can be “pinched” with a solenoid or motor. This pump type functions via a motor or solenoid which pushes a pinch bar to stop flow through the envelope.

The primary advantage of this valve is the ease of maintenance as the envelope is often made of rubber and is easily replaceable. Therefore, should any issues arise with the integrity of the pump diaphragm, it is easy to replace. Additionally, there is no rerouting of fluid to run through the pump – the flow path is direct and straight – so there is decreased risk of a clog. The primary disadvantage of this valve comes from the properties of the often elastic sleeve used as the pinch diaphragm. Because this material must be elastomeric, negative pressure can cause the valve to close which, in our application, could become an unintended issue as the use of peristaltic pumps requires the use of a negative pressure differential to draw liquid through the tubing.

3.4.3 Ball Valve

The final option being considered for the applications of this project is the ball valve. By far the most universal valve being considered, the ball valve functions by the rotation of a hollow ball with a hole drilled through the center.

As the handle is rotated, the opening in the ball rotates to be perpendicular to the flow through the valve, closing off flow. The primary advantage of this option is the low power consumption. That is, while the solenoid and pinch valves rely on electromagnetics to change the state of the valve, these can be operated by a simple electric actuator. Additionally, ball valves are very reliable due to the robustness of their design, the lack of complex mechanical elements, and the direct flow path through them. However, the shortcoming of this valve type is the operation time, which is very slow in comparison to a solenoid-operated valve, and can even exceed 3 seconds.

3.4.4 Valve Comparison

Feature	Solenoid Valve	Pinch Valve	Ball Valve
Open/Close Speed	Fast ~30ms	Fast ~30ms	Slow >1s
Robustness	Robust	Not very robust	Very Robust
Negative Pressure Resistance	Resistant	Not very Resistant	Very Resistant
Lifetime	1-3 years	6 months - 3 years	8-10 years

Table 5: Valve Comparison

When considering the function of the Automated Bartender, precision and response time of the valves is one of the most important aspects: A response time of greater than one second is unacceptable for our applications. Therefore, despite the robustness and reliability of the ball valve, it is likely not the best choice for this project. While both the pinch valves and solenoid valves are suitable for the applications of this project, we will make use of solenoid valves to avoid any issues resulting from the negative pressure generated by the peristaltic pumps.

3.5 Power Components

The Automated Bartender will be powered by wall power, the voltage will be stepped down using transformers, and the current will be rectified to direct current. Further stepping down the voltage will be achieved by using a buck converter (DC to DC converter) the voltage will be stepped down to 3.3 volts, 5 volts, and 12 volts. The power supply of the automated Bartender will be an AC/DC converter since this device will be able to rectify AC to DC and provide a smooth DC output. This AC/DC converter will be mounted on a PCB.

3.5.1 Voltage Regulator

The Automated Bartender will have various critical elements ranging from microcontrollers, sensors, and pumps that will be powered by various voltage levels. It is imperative to ensure that the Automated Bartender platform will have a source of stable and regulated voltages to ensure a safe and stable operation. Voltage regulators will help to protect the more sensitive and vulnerable components from voltage spikes and overvoltage. These voltage regulators will play a role in mitigating any voltage spikes and fluctuations that can come from the power supply that can cause harm or inconsistencies throughout the project.

3.5.1.1 Linear Voltage Regulator

Various voltage levels will power the Automated Bartender. The microcontroller unit will be powered by lower voltages compared to the peristaltic pumps and the solenoid valves. To achieve lower voltages, we could employ the use of linear voltage regulators like the LM2940 and the L7885CV, but these types of regulators have been proven not to be remarkably power-efficient when dealing with regulating higher voltages. These linear voltage regulators will drop excess voltage from the circuit by dissipating the excess voltage as heat.

When there is a larger voltage drop from the input voltage to the output voltage, these linear regulators can generate a lot of heat, which in turn is more energy wasted. The increased heat radiated from linear regulators could affect other components near the regulator. Linear voltage regulators are easy components to work with and implement but they would not be the best choice for this project due to their low energy efficiency and high operating temperatures.

3.5.1.2 Switching Regulator

A switching regulator, also known as a switch mode power supply (SMPS), is a type of regulator used to provide a stable output voltage. SMPS are very efficient devices to regulate power compared to a linear voltage regulator. Linear regulators will just drop excess voltage as heat, leading to more energy being wasted. A switching regulator will use a switching circuit composed of MOSFETs to control the current flowing through an inductor. The working principle of a switch mode power supply is when the switch is closed, current will flow through the inductor, creating a magnetic field, and when the switch is open, the current will stop flowing. The magnetic field will collapse, releasing the stored energy into the output of the circuit.

Switching regulators are very efficient devices, due to the switching aspect of the circuit, the current will only flow when the switch is closed, therefore reducing the amount of wasted energy. The frequency of the switching action can be adjusted by changing the duty cycle of the PWM signal applied at the gate of the MOSFET. Switching regulators serve as the fundamental operating principle that many types of DC/DC and AC/DC converters operate on.

3.5.1.3 Regulator Selection

The regulator that will be used in the Automated Bartender platform will be the switching regulator. This type of regulator will be the best choice since it is a higher-efficiency regulator compared to linear regulators. Switching regulators will convert voltages with minimal power loss and dissipate less heat, which will in turn improve the lifespan and the reliability of the Automated Bartender. Linear regulators can come in large form factors that will take up unnecessary space, whereas the switching regulators can be placed on a PCB. By opting for a switch-mode regulator, we aim to maximize energy efficiency, minimize heat dissipation, and accommodate for a more compact and portable design of the project, ensuring an optimal balance between performance and resource utilization.

Regulator	Efficiency	Size	Complexity	Operation
Linear	Low (at higher voltages)	Small (usually comes on an IC)	Low - Medium	Drops excess voltage as heat

Switching	High	Medium to large (usually comes on a PCB)	Medium - High	Using switches to generate magnetic fields

Table 6: Regulator Technology Comparison

3.5.2 Power Supply

An AC/DC converter will power the Automated Bartender; this system will serve to rectify and step down the 120VAC wall power to a more suitable and easier-to-work-with DC voltage level. AC/DC converters use a transformer to step down the input voltage which is more appropriate for common applications and projects. Stepping down the voltage will prevent the device from being overloaded by power and reduce the chances of damaging the devices being powered. When the voltage is stepped down using a transformer, the output current will increase due to the conservation of energy. After the voltage has been stepped down, the signal must be rectified from AC to DC. Various converter topologies exist that can be employed to achieve this output.

3.5.2.1 Flyback Converter Topology

The flyback converter is a type of switching circuit topology that can be used in both AC/DC and DC/DC conversions. The working principle of this converter is that energy is stored in a magnetic field in the air gap between the transformers when a switch is closed, and then the energy stored in the magnetic field is transferred when the switch is open. When the switch is closed and in the "on" position, current will flow, the magnetic field on the primary side will be created, and a negative voltage will be induced on the secondary side. The diode on the secondary side will be reversed biased, and the output capacitor will supply power to the load. When the switch is open and in the "off" position, the magnetic field will start to collapse. The primary side will have a negative voltage, and the secondary side will have a positive voltage. The diode on the secondary side will be forward biased, and the energy from the transformer will charge the output capacitor and supply power to the load. The opening and closing of the switch is achieved by using a MOSFET; the voltage can be regulated by controlling the duty cycle applied at the gate of the MOSFET. This type of converter will use transformers that are mutually coupled. Flyback converters can be used in applications to step up or step down voltages. This circuit topology is usually composed of transformers, diodes, transistors, and various energy storage devices, as seen in Figure 8. Flyback converters are a type of isolated power converter, meaning that the input and the output of the circuit are isolated. This isolation is achieved by using a transformer since the primary and the secondary side of the coil are separated.

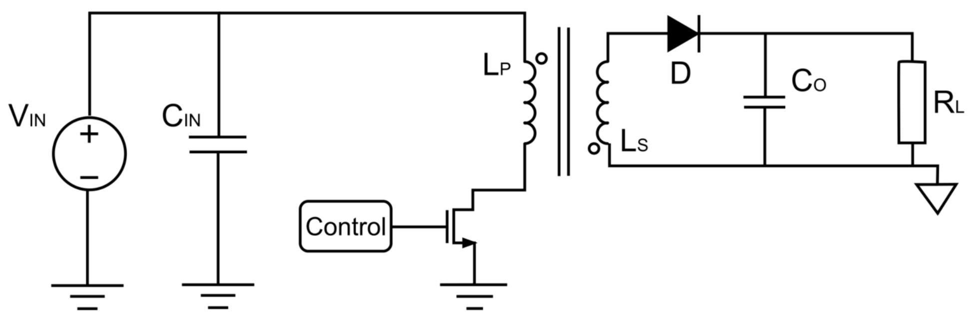


Figure 8: Flyback Converter Topology

3.5.2.2 Forward Converter Topology

The forward converter is a type of converter that uses transformers to either increase or decrease the output voltage. The load connected to the converter is protected by galvanic isolation between the transformers. The topology of the forward converter looks similar to the flyback converter, but these two converters operate very differently. The forward converter will transfer energy from the primary side of the transformer to the secondary side when the switch is closed, unlike the flyback, which operates in the reverse. When the switch is closed, and in the “on” state, there will be current flowing through the primary side of the transformer, which will, in turn, create a magnetic field in the transformer's core. The magnetic field will be coupled with the secondary side of the transformer, which will induce a voltage on the secondary side of the transformer.

When this coupling happens, this is when the transfer of energy will happen, and current will flow at the same time across the primary and secondary sides of the transformer. Once the switch is open and in the “off” state, the magnetic field in the transformer's core collapses. As the field is collapsing, this will induce a voltage on the secondary side of the transformer that will have a flipped polarity; this will then be rectified by diodes. The voltage can be regulated by using a PWM signal at the gate of the MOSFET; this will control the opening and closing of the switch and will determine how long voltage is applied to the primary side of the transformer, which will, in turn, affect the intensity of the magnetic field.

Forward converters are the most common and trusted type of converter. Still, this topology comes with the downside that it is more complex and expensive compared to other converter topologies.

3.5.2.3 LLC Resonant Converter Topology

LLC Resonant converters are a type of switch mode power supply that combines the use of a switch mode power supply (SMPS) and a resonant. This SMPS uses inductors and capacitors, which are vital components of the converter; these components will oscillate at a resonant frequency. LLC resonant convert operates by taking advantage of the resonant behavior of an LC circuit. The converter is made up of 4 different blocks; these blocks are the power switches, resonant tanks, transformer, and diode rectifier; all of these blocks will perform specific functions vital to the performance of the converter. The first block is made of switches that will convert the input signal into high-frequency square waves; this switching is performed by employing the use of MOSFETS. These high-frequency square waves will be fed into the resonant tank; this circuit is composed of LC components, and this circuit will remove the harmonics of a square wave and convert it into a sine wave. This happens because a square wave contains the fundamental sine wave and odd harmonics. Harmonics that are close to the resonant frequency of the resonant tank circuit will be passed, and the harmonics that are not close to the resonant frequency will be filtered out, leaving a sine wave at the output of the resonant tank circuit. The output sine wave will be sent to a transformer, where it will be either stepped up or down, depending on the application. The sine wave is then rectified by diodes to output a DC signal.

3.5.2.4 Push-Pull Converter Topology

Push-pull converter topology consists of a transformer that is center-tapped and two power MOSFETs that are controlled by a PWM signal. The MOSFETs are connected to each end of the primary windings of the transformer. The secondary winding of the transformer is attached to the load and a diode bridge for rectification. This converter will operate in two different stages, known as the push and pull stages. In the push stage, energy will be stored in the transformer. This occurs when the first power MOSFET is turned on and the second power MOSFET is off, and current is allowed to flow through the first half of the windings of the primary side of the transformer. This will induce a magnetic field to store energy.

In the push stage, the first MOSFET is turned off, and the second MOSFET is turned on. When in the push stage, the current will be allowed to flow to the second coil on the primary side of the transformer and this will collapse the magnetic field inside the transformer, releasing energy into the load connected to the converter. With the constant turning on and off the MOSFETs this will create a push and pull effect on the transformer that will supply power to the attached load. The push-pull topology is a topology that is highly efficient due to the unique way it utilizes a transformer. By using a transformer that is center-tapped, this will reduce the copper loss of the transformer and increase its efficiency. While this topology is very efficient, it can be a complex design to work with since the timing of the MOSFETs have to be precise to reduce the risk of both MOSFETs being turned off or on at the same time, which can lead to unwanted effects in the circuit. If the timing of the MOSFETs are incorrect, this can lead to an overall reduced efficiency of the converter. Additionally, the price of this topology can become expensive due to the design requirements of the transformer.

3.5.2.5 Converter Topology Selection

The Flyback converter topology will be the topology that will be used for creating the power supply of the Automated Bartender. This topology will create a high-efficiency AC/DC converter at a reasonable size. The primary factor that motivated the choice of the flyback converter is the simplicity and the cost-effectiveness of the topology making this an attractive option for this project where the group is providing their own funding and how the regulation can be adjusted by varying the PWM signal at the gate of the switching MOSFET. Using the flyback topology will provide electrical isolation between the input and output of the circuit. Having this isolation will help to increase the safety of the converter by providing protection against voltage spikes from occurring. The flyback topology will provide a base circuit topology that various circuits in the Automated Bartender can utilize. Overall the flyback topology will be a choice that will provide a balance of performance, cost-effectiveness, safety, and versatility. The converter will be mounted on a PCB and provide power to the entire system.

Converter	Operation	Cost	Efficiency	Complexity
Flyback	Energy stored when switch is closed	Low	High	Low

Forward	Energy transferred when switch is closed	Medium	High	Medium
LLC Resonant	Energy transferred via resonance	High	High	High
Push-Pull	Energy transferred via switches opening and closing	Medium	High	High

Table 7: Converter Topology Comparison

3.5.3 DC-to-DC Step-Down Converters (Buck Converter)

A DC-to-DC step-down converter, also known as a buck converter, is a circuit that takes a DC voltage and steps it down to a lower DC voltage, most commonly 3.3V, 5V, or 12V. These converters are most commonly placed on a PCB and they all have a DC-to-DC converter IC combined with various RLC components to achieve the desired output voltage. Most buck converters are synchronous converters, meaning that they use MOSFETs to rectify a signal instead of the traditional diodes. By using the MOSFET, it would be easier to control the switching timing for better regulation, and the switching MOSFET will be actively controlling the flow of current. In synchronous buck converters, the MOSFETs will have a lower voltage drop across the components compared to a traditional diode, which would lead to less energy wasted and increased overall efficiency.

3.5.3.1 3.3V DC-To-DC Converter IC Research

Buck converters will serve as an alternative to linear voltage regulators. The researched buck convert ICs are the TPS56339, TPS563300, TPS62932, and TPS62933F. These buck converters will have to take an input voltage step it down to 3.3 VDC and have an output current of at least 1 amp.

The first buck converter researched is the TPS56339 by Texas Instruments. This converter is a high-efficiency synchronous buck converter. This converter has an input voltage range of 4.5V to 24V and has a maximum safe continuous output current of 3 amps. This device comes in the SOT-23 package.

The second buck converter researched is the TPS563300 synchronous buck converter by Texas Instruments. This converter has an input voltage range of 3.8V to 28V and has a maximum safe continuous output current of 3 amps. This device comes in the SOT-583 package.

The third buck converter researched is the TPS62932 synchronous buck converter by Texas Instruments. This converter has an input voltage range of 3.8V to 30V and has a maximum safe continuous output current of 2 amps. This device comes in the SOT-583 package.

The fourth buck converter researched is the TPS62933F. This converter has an input voltage range of 3.8V to 30V and has a maximum safe continuous output current of 3 amps. This device comes in the SOT-583 package.

Converter	Price	$Vout_{MAX}$	$Iout_{MAX}$	Size
TPS56339	$1.00	16 V	3 A	SOT-23 (1.60 mm × 2.90 mm)
TPS563300	$0.63	22 V	3 A	SOT-583 (1.60 mm × 2.10 mm)
TPS62932	$0.66	22 V	2 A	SOT-583 (1.60 mm × 2.10 mm)
TPS62933F	$0.70	22 V	3 A	SOT-583 (1.60 mm × 2.10 mm)

Table 8: ICs for 3.3V Buck Converter

All of these buck converter ICs have a $Vout_{MAX}$ that is much greater than the desired voltage levels. When the chip is in the proper buck converter configuration, the desired output voltage will be achieved.

3.5.3.2 5V DC-To-DC Converter IC Research

The second type of DC-to-DC converter needed will be a 5V buck converter. The purpose of the converter will be to step down the input voltage to 5V. Therefore, components needing 5V to operate will be supplied with the appropriate voltage. The buck converter ICs researched are the LMR51420, LMR51430, and TPS629330.

The first buck converter researched is the LMR51420 synchronous buck converter by Texas Instruments. This converter has an input voltage range of 4.5V to 36V and has a maximum safe continuous output current of 2 amps. This device comes in the SOT-23 package.

The second buck converter researched is the LMR51430 synchronous buck converter by Texas Instruments. This converter has an input voltage range of 4.5V to 36V and has a maximum safe continuous output current of 3 amps. This device comes in the SOT-23 package.

The third buck converter researched is the TPS629330 synchronous buck converter by Texas Instruments. This converter has an input voltage range of 3.8V to 30V and has a maximum safe continuous output current of 3 amps. This device comes in the SOT-583 package.

Converter	Price	$Vout_{MAX}$	$Iout_{MAX}$	Size
LMR51420	$0.32	34.2V	2 A	SOT-23 (1.60 mm × 2.90 mm)
LMR51430	$0.37	34.2V	3 A	SOT-23 (1.60 mm × 2.90 mm)
TPS629330	$0.18	22V	3 A	SOT-583 (1.60 mm × 2.10 mm)

Table 9: ICs for 5V Buck Converter

3.5.3.3 12V DC-To-DC Converter IC Research

The third type of DC-to-DC converter needed will be a 12V buck converter. The purpose of the converter will be to step down the input voltage to 12V. Therefore, components needing 12V to operate will be supplied with the appropriate voltage. The 12V buck converter can be created using all of the same buck converter ICs discussed in the previous sections relating to 3.3V and 5V buck converters. Being able to use the same buck converter IC across multiple converters will help to reduce the overall cost of the various DC/DC converters needed.

3.6 Touchscreen Display

The project demands the addition of a touchscreen in order to develop a completely automated system that allows consumers to choose and customize their drinks while offering a user-friendly interface. This feature enables the display of visuals, dynamic user interface, and real-time information, making the experience more interactive and enriching. The most important considerations when choosing the touchscreen model include technology interface, cost, size, support for DC voltage, power consumption, compatibility with either SPI or I^2C protocols for data transmission, ease of implementation to project, rapid response time, a simplified configuration process, and availability of software libraries. Additional factors that should be considered, although they may not be the primary determinants for selection, include color range, resolution quality, vibrancy, audio feedback, and warranty. Considering the specific application requirements, size constraints, and budget considerations for the touchscreen display in the project, the most suitable and commonly available touchscreen monitor technologies on the market are the 5-wire resistive technology and surface capacitive technology.

A 5-wire resistive touch technology detects a touch input through a simple continuity test. As shown in Figure 9, the monitor has two thin metal layers separated by a few insulating spacers, with an air gap between the metal layers. When the screen is pushed, the two conductive material plates come into contact with each other, resulting in a change in resistance at that contact point, which is how the touch input is detected. In addition, this type of display works with both fingers and stylus pens, since the touch position is detected where pressure is applied on the screen. This technology is desirable for our project since it has various advantages, including relatively cheap cost, low power consumption, and resistance to moisture and liquids. However, there are certain disadvantages that must be addressed. For instance, due to its basic technology, the accuracy of touch is inversely related to the size of the monitor; thus, it might be an issue for bigger displays; and it has the lowest screen clarity of all others.

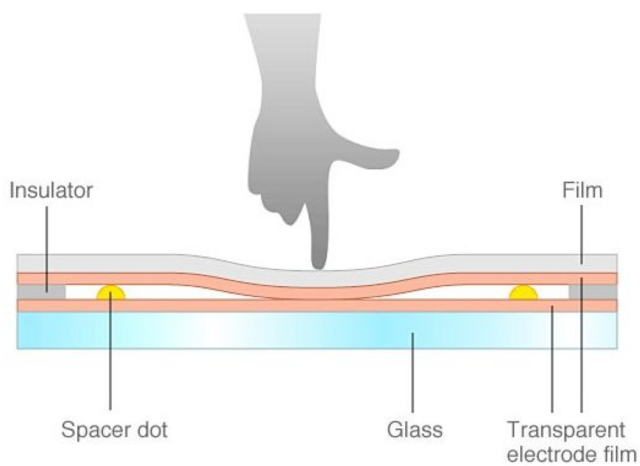


Figure 9: 5-Wire Resistive Wire Touchscreen

The input touch for surface capacitive touch technology, as the name implies, is determined by the change in capacitance captured by corner sensors that constantly measure the level of capacitance. According to Figure 10, when the screen is touched, an electrical charge transfer between the finger and the charged electrode occurs; as a result, the controller can accurately pinpoint the location of the touch because it is where the sensors detected the capacitance drop. This technology option is particularly compelling as it provides a higher level of accuracy in determining the touch input position and superior image clarity when compared to the 5-wire resistive technology. However, it comes at a higher cost, does not work if wearing gloves or any non-conductive pen, and is susceptible to Electro-Magnetic Interference (EMI) and Radio Frequency Interference (RFI). Consequently, touch input may be affected depending on the environment around the project.

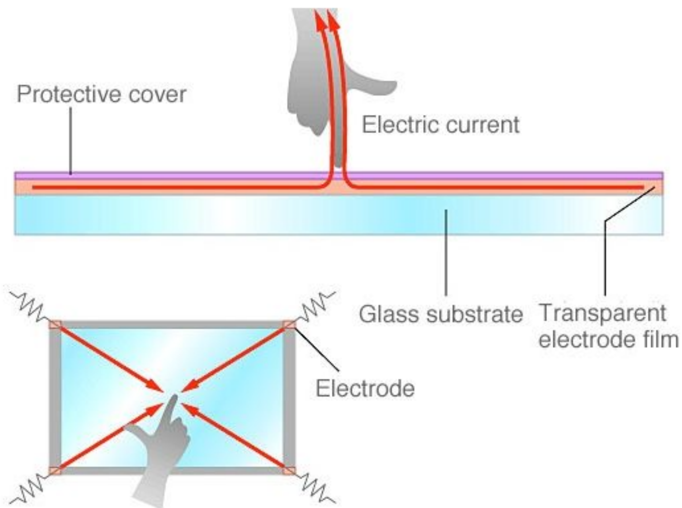


Figure 10: Surface Capacitive Touchscreen

Another important factor to consider when selecting an appropriate touchscreen is how the device will be implemented in the project. Because the project will already have a complex PCB design to power all of the components and allow the microcontroller to interact with the components, the design criteria for PCB in our project is expected to be achieved. Our faculty advisor then recommended and accepted that for the implementation of our touchscreen, we could use solutions that did not require us to design the PCB for the implementation of the touchscreen display. Therefore, with that in mind, it leaves two potential options. The first option involves using a touchscreen display module with a designed pinout for a specific communication protocol and a built-in SD card for program storage. This option would then require female-to-female jumper wires to interface with a microcontroller. The second option is to have a touchscreen with HDMI, USB, or USB-C cable capabilities for use with development boards, which offers a 'plug-and-play' connection for programming and power. However, since option two involves significantly more expensive solutions, the comparison will focus exclusively on the components from option one.

3.6.1 Touchscreen Module Comparison

This option allows the touchscreen display to be driven by the same microcontroller that is in charge of controlling the entire project. Based on our microcontroller selection being the ESP32, its dual-core 32-bit allows the microcontroller to communicate with multiple devices using different communication protocols. Therefore, the plan is to have one core to interface with all the components, excluding the touchscreen, using the I2C protocol. And to prevent interference with the others, ideally, another core should be used with the SPI protocol for the touchscreen. While there are preferences, the options to be considered below will be checked for multiple microcontrollers and communication protocol compatibility for a safe and complete evaluation of potential display solutions.

	Hosyond 3.2 Inches TFT LCD SPI Module ILI9341	Hosyond 3.5 Inches TFT LCD SPI Module ILI9488	4 Inches TFT LCD SPI Module ILI9486
Price	$15.99	$16.99	$20.99
Touch Technology	Resistive (not specified)	Resistive (not specified)	Resistive
Voltage	2.5V - 3.3V	2.5V - 3.3V	3.3V - 5V
Size	3.2 inches (diagonal)	3.5 inches (diagonal)	4 inches (diagonal)
Resolution	240 X 320	320 X 480	320 X 480
Color Range	Full RGB	Full RGB	Full RGB
Development Board	ESP32, Arduino UNO/MEGA2560, STM32,	ESP32, Arduino UNO/MEGA2560, STM32,	ESP32

	STC89/STC12	STC89/STC12	
Communication Protocol	4-wire SPI	4-wire SPI	4-wire SPI
Well-Documented	Yes	Yes	Yes
Library Accessibility	Large	Large	Moderate
Community Support	Great	Good	Acceptable

Table 10: Touchscreen Comparison

3.6.2 Hosyond 3.2 Inches TFT LCD SPI Module ILI9341

The main focus of this touchscreen option is the ILI9341 IC Driver, which contains many beneficial operating features that can be used in our project, such as 3-line/4-line SPI serial interface, also it has a very good low-power consumption functionality since it can operate with 2.5V - 3.3V, which is also another good thing since this actually matches the operating voltage for ESP32, and can be programmed to have sleep and stand-by modes for even better power saving capabilities. Another consideration is that this IC driver is very well supported by the community, contains easily accessible documentation and library options. The type of technology was not specified on the specification sheet, but it is possible to determine that it is resistive touch technology since it is compatible with a stylus/touch pen. For the project 3.2 inches might be considered a little small, but this is the biggest monitor that could be found for this model.

3.6.3 Hosyond 3.5 Inches TFT LCD SPI Module ILI9488

This ILI9488 option is quite similar to the ILI9341. It shares the same advantages of low power usage, sleep modes, and standby modes. The main differences to consider are a slightly bigger 3.5-inch diagonal display size, higher resolution, and higher price. The key factor to think about is library support for this module. Compared to the ILI9341, the ILI9488 has fewer community resources available. However, it can still be programmed to be driven by an ESP32 using the TFT_eSPI library designed for ILI9341 in the Arduino IDE, it only requires modifying the code to match the initialization sequences, display settings, and graphics functions for this module.

It is worth mentioning that there is another, more appealing option with the same IC driver available at a lower price and equipped with capacitive touch technology, found on the Walmart website. This could potentially be a budget-friendly choice with the prospect of better image quality and touch detection. However, this option is shipped and sold through Walmart but originates from a third-party vendor in China, which does not have an official website. Consequently, obtaining the specification sheet directly from the manufacturer or any specific official documentation for the device is very difficult, and the product information remains unverified. Therefore, it was not included in the comparison table.

3.6.4 4 inch TFT LCD SPI Module ILI9486

The IC driver for this module offers similar functionalities and power-saving benefits as the others. What sets this option apart is its larger 4-inch screen, a higher price tag, and requires a different library compared to the other two. It was discovered that this ILI9486 module can be programmed using SquareLine Studio software along with the LVGL graphics library. However, there are some drawbacks to consider. This option is sourced from a third-party vendor, and no documentation directly from the manufacturer could be located. This reliance on a third-party source might pose a risk, especially if the advertised product information turns out to be inaccurate.

3.6.5 Touchscreen Final Selection

The Hosyond 3.5 Inches TFT LCD SPI Module ILI9488 was the final pick for the touchscreen display. The reasons for that choice are that it delivers the best value when compared to the other options in terms of size, image quality, and interface technology. In addition, of all the available options, this one was the quickest to be delivered. This helps a lot because it gives enough time to test and validate this touchscreen option to make sure it satisfies the project's requirements and matches the product information.

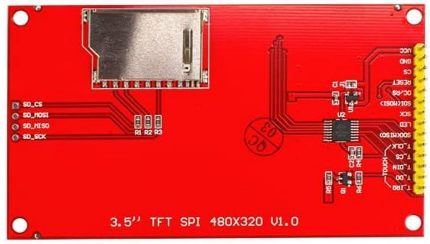


Figure 11: Hosyond 3.5 Inches TFT LCD SPI

3.7 Weight Sensors

An important part of any pour is making sure you get the right amount. Humans have eyes to detect the amount of liquid to be poured, albeit not very accurate alone. To this degree, we create reservoirs and manifolds to help measure out liquids for the perfect pour, still using their eyes along with their measuring devices. Our Automated Bartender will not have eyes.

We must create systems that allow our bartender to know how much liquid is poured, not relying simply on measures that will vary between each pour: Time, voltage and current to the motor, and resistance within the pump are all things that can vary between each pour. But the weight of liquid per unit will always be the same for a specific reservoir, and it is this metric that we will measure with our electronic feedback system with the help of electric weight sensors.

The weight sensors, as part of the Automated Bartender's electronic feedback system, are located beneath each liquid reservoir and at the depositing site. These weight sensors are used to feed information back to the microcontroller to identify the amount of liquid poured via weight change from reservoir to deposit site and detect how much liquid remains within a given reservoir. This information will determine the accuracy of pours and work in tandem with the LED system to notify how much liquid is left within a reservoir based on remaining weight.

Weight sensors in all generality are a form of transducer, a device meant to convert one type of energy to another, typically compression to electrical for weight sensors. There are many different configurations of weight sensors that achieve this goal, and many that are applicable to the needs of the Bartender.

3.7.1 Capacitive Sensors

Capacitive based sensors, specifically proximity sensors, use a system of dielectrics plates to emit an electric field and measure the change in the capacitance and resistance of the field through the sensor, which will change the reading to the microcontroller accordingly.

This sensor type can be used in our system by applying multiple sensors at different heights next to each reservoir on a cage and detecting the change within the electric fields to determine the liquid level within each reservoir. This setup would allow a sensor's capacitive value to be tied to an LED, where a specific threshold value from the sensor will toggle the feedback LEDs on or off.

3.7.2 Hydraulic Sensors

A hydraulic pressure sensor uses the gravitational force of pressure of a fluid pushing down to determine the weight on the sensor itself. Uses of this sensor include power steering systems, hydraulic lifts, hydraulic presses, and more. This type of sensor could be applied within the bottom of a reservoir and would provide a reliable value due to the constant, stable force of pressure in a static liquid in between pours.

Using a hydraulic pressure sensor will require specially designed reservoirs that allow the hydraulic pressure sensor to be mounted to the base of the reservoir to allow the sensor to be under pressure at all liquid levels. When the decoded information from the sensor reaches the microcontroller, they could be compared against custom value ranges to determine the amount of liquid left and update the feedback LEDs accordingly. However, this system would be difficult to implement within the depositing site, unless there is a secondary container the liquid pours into before reaching the depositing point.

3.7.3 Strain Gauge Sensor

Finally, a strain gauge sensor works based on the compression or tension of a gauge within the sensor. As the gauge compresses, resistance decreases; as it tensions, resistance increases. There are many different types of strain gauge sensors with different applications, some of which can be applied via placing the reservoir directly onto the strain gauge sensor and measuring its compression, such as a weight scale.

Using a type of strain gauge sensor, a load cell, would work by placing the sensor beneath the location of each reservoir over a mat or other surface. This surface would need to balance on top of the load cell to achieve accurate values for the remaining liquid level. These values, similar to the hydraulic pressure sensor, would be divided into ranges for the feedback LEDs to be toggled on and off.

Figure 12: A Straight Bar Load Cell, a Type of Strain Gauge Sensor

3.7.4 Weight Sensor Selection

With these sensors described, it is important to define how a sensor will be chosen. The main focuses come from the amount of sensors needed per reservoir, which controls the space constraints, the implementation at the reservoir and at the deposit site, which controls space constraints and complexity, and the accuracy of each sensor, primarily the “creep” specification.

Creep for a sensor is the change detected within a static load over a period of time, meaning the rate of change in a load that has not been removed from the sensor. Considering the primary focus of the sensor is to measure the reservoirs, creep is the most important specification for accuracy within our system. Creep is generally measured as a percentage of the full-scale output (FSO) or rated output (RO) per a period of time, which itself is rated in mV/V. This, along with the amount of sensors per reservoir and their implementation into the system can be seen below.

Weight Sensor Type	Sensors per Reservoir	Implementation for Reservoir	Implementation for Deposit Site	Accuracy (Creep %FS/t)
Capacitive Weight Sensor	Multiple - 3	Cage with side mounts to reservoirs for	Bar/cage with side mounts to depositing cup	Reliant on placement, environme

		each capacitive sensor	for sensor	nt, etc.
Hydraulic Pressure Sensor	Single	Custom-made reservoirs with hole in base to mount pressure sensor	Separate reservoir or container that measures pressure before depositing to site	+/- 0.1
Strain Gauge Sensor (Load Cell)	Single	Surface plate attached to load cell sensor that reservoirs rest on	Surface plate attached to load cell sensor that depositing cup rests on	+/- 0.1

Table 11: Weight Sensor Comparison

Based on the required implementation of the sensors at both the reservoir and depositing sites and the accuracy of the sensor, it leads to the choice of the strain gauge sensor via load cells. The load cells require the least amount of additional construction and space constraints for the placement of the sensors and provide an accurate and fairly consistent reading. Since the optimal location of these sensors is located directly beneath the reservoirs and depositing site, they can be placed underneath a mat or similar surface with the reservoir resting above it, providing a consistent, repeatable value for the microcontroller to decode.

There is the requirement that the load cells be integrated in a way such that each load cell bears all the weight of the reservoir above it regardless of its placement, meaning it must be integrated in a way that nothing else can accidentally support the reservoir's weight.

Comparing the three sensors, the implementation of the capacitive weight sensor would create a lot of additional construction and space constraints with the amount of sensors and infrastructure needed to accommodate their placement. Additionally, while it is possible to set up the feedback LEDs to tell the rough liquid level of the reservoir, the capacitive sensor would not provide a solid finite, repeatable value due to the need for precise placement of a reservoir to achieve the same impact on the sensor's electric field for the same value, in addition to proper ventilation to keep the space around the sensors clear of dust or other particulates. Circumventing these drawbacks would require custom cages for each reservoir and at the deposit site, which complicates the design.

With the hydraulic pressure sensor, the largest issue would be measuring the liquid at the deposit site. A new container would need to be introduced between the reservoir and deposit site, which would require additional construction and space constraints and still create some uncertainty in the amount of liquid that reaches the actual cup at the deposit site.

The specific load cell selected is the TAL220B straight bar, which comes in a wheatstone bridge configuration with four wires for each node of the load cell. This straight bar is rated for 5 kg (11 lbs.), more than enough to support each individual reservoir. The dimensions of this load cell is 1.27x 1.27 x 5.5 cm. With such a compact size, its integration into the system is straightforward: Place the cell in a small divot or hole beneath the reservoir and allow the reservoir to balance atop of it. If done right, the sensor will bear the full weight of the reservoir and output consistent, reliable values for the microcontroller to read.

3.7.5 Sensor Integration

While the values of the sensor are sent to the microcontroller, the information sent is not readily understood by the system, and without a type of guide or scale for the value the microcontroller is receiving, it would be difficult to create a scale for each value we receive. To this end, a device that can translate these inputs into a usable value for us as users to interpret is needed.

This requirement can be met with the HX711 Integrated Circuit component. The HX711 Load Cell Amplifier decodes the change of resistance within a gauge, allowing the user to code, interpret, and analyze the weight of a load at all times. The weight can be coded to different units as well. This will be beneficial for allowing our device to work in different measuring systems. Depending on the HX711 board, it can connect to up to two load cells per HX711 board. These boards, along with the TAL220B straight bars at the reservoirs and deposit site, will provide ample feedback for the reliability of each pour.

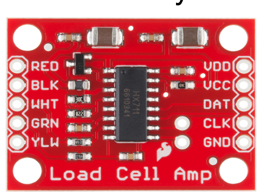


Figure 13: An HX711 Load Cell Amplifier

The integration of the weight sensors to the microcontroller is achieved using the inter-integrated circuit serial communication bus. The microcontroller chosen, ESP32, has inter-integrated circuit capability, allowing the weight sensor peripherals to connect as slaves to the microcontroller. Inter-integrated circuit communication works well in this scenario considering the amount of sensors we will need to connect: Inter-integrated circuit communication allows many peripherals to connect to it thanks to the unique style of labeled addressing that allows peripherals to be chained across the same data lines without loss of information or communication.

On top of this, both master and slave can send and receive information across these lines, though not at the same time; this setup is commonly referred to as a half-duplex channel. The communication is done using two serial lines, a serial data line (SDA) and serial clock line (SCL), that the master controls and shares with all slaves.

A downside to using I^2C is the relatively slower speed that information travels to and from a specific slave compared to using other communication types on a microcontroller, such as universal asynchronous receive-and-transmit (UART) or serial peripheral interface (SPI); however, considering the application of weight measurement between pours, this “slower” pace of data communication is still more than responsive enough for updating the weight values and providing feedback to the microcontroller in time for the data to be usable.

3.8 LEDs

While the weight sensors allow the microcontroller to track the reservoir liquid levels, there is nothing that directly indicates this information in a way for the user to see. While not a crucial portion of the system, having an informed user greatly simplifies the process of placing an order on drinks that use specific liquids. Adding simple devices that react to the reservoir levels’ data would improve the efficiency and reliability of the ordering process. In response to this, light emitting diodes provide the low-cost, low-complexity integration that communicates the liquid level information to users.

Light emitting diodes, or LEDs, are a semiconductor device that emits photons via current flow. Electrons separate from their “holes”, generating electron-hole pairs, and flow from the cathode side to a metal contact into a positively charged region called the p-type region, a neutrally charged active region, a negatively charged region called the n-type region, and then out another metal contact to the anode side.

These regions are generally a silicon (Si) substrate base that is then doped with either group 13 (+) elements of group 15 (-) elements on the periodic table to change the intrinsic silicon to a p-type or n-type respectively. Other methods, such as subtraction and lithography, are used to form the silicon into the desired shape and composition.

The diode portion of the system, made of the oppositely charged semiconductor materials and the neutral depletion region, form a PN junction, which allows electrons to flow primarily in only one direction from the p-type to n-type region, recombining with the separated “holes” in a process called radiative recombination. When crossing the depletion region between the two oppositely-charged materials, energy is released in the form of light. It is this process that causes visible light to be emitted from the LED. The emitted color can be changed either through a colored transparent plastic case or by changing the depletion region width of the PN junction based on the wavelength of desired color.

LEDs come in many forms, primarily as either illuminative and indicator LEDs, with different functions. Illuminative LEDs have a high lumen count, generally in the thousands, made for actual lighting uses such as LED flashlights and lightbulbs. Indicator LEDs generally have a low lumen count and are used in systems that require some indication of a state or process, such as power buttons on electronics or industrial machines.

For the Automated Bartender, indicator type LEDs will provide the required function and connectivity with the microcontroller, as the low lumen count will allow users to understand the liquid level without being overwhelmed by bright light and draw less voltage and power from the microcontroller compared to an illuminative LED.

From within indicator LEDs, there are many different choices as to what LED type to use. The most common LEDs within this category include single-color LEDs and RGB (red, green, blue) LEDs. The feedback system is meant to mimic a stop light, a device setup that most are familiar with, especially at legal drinking age. This means that the three colors are used for the LEDs, red, yellow, and green.

3.8.1 Single-Color LED

Single-color LEDs are the most basic of semiconductor devices. They come in the form of a diode with a narrow range of wavelengths (and therefore, bandgap), and with a translucent cap that generally matches their emitted light for refraction. There is a turn-on voltage that is required, generally about a 0.7 volt potential difference from anode to cathode, allowing the LED to function on very low voltages. The 3.3 V output from the ESP32 microcontroller is enough in this instance.

Implementing a single-color LED can be as simple as connecting an LED to a pin from the ESP32 and outputting voltage to it at certain values the load cell amplifier sends to the microcontroller. This will require different colored LEDs to distinguish the level of the liquid, possibly in three tiers: From 100% to 60%, a green LED is powered. From 60% to 30%, a yellow LED is powered. From 30% to 0%, a red LED is powered. These three LEDs will be placed in front of each reservoir, totalling up to 12 LEDs for the reservoirs alone. An LED at the depositing site may be implemented to show when it is safe to remove the cup for consumption.

3.8.2 RGB LED

RGB LEDs are three diodes with different band gaps (with wavelengths of red, green, and blue light) tied to a common anode or cathode that allows a multitude of colors to be displayed based on the voltage and resistance through each leg of the diode. Where traditional LEDs have two contacts, one for the cathode and anode, RGB LEDs have four, one for the common and one for each diode that represents the red, blue, or green LED, though there are RGB LEDs with two legs and an on-board controller. An RGB LED can come in many forms itself, either in the two or four leg configurations, or in a strip, which has many RGB LEDs connected in series.

Using RGB LEDs would allow more customization and guest-facing entertainment, something that a bartender strives for in their service. The implementation for the two-pin or four-pin RGB LED would be the same as the single-color LED, except that multiple pins on the ESP32 would be on at the same time for the four-pin RGB, specifically for emitting a yellow light considering that two of the diodes within the RGB LED are already red and green. Yellow is produced by red and green in equal intensity, meaning turning the pins for the red LED and green LED on at the same time should produce yellow, making this configuration work on DC power.

Since an RGB LED can produce multiple colors itself, only one RGB, or one strip, will be needed for each reservoir. Similar to the single-color LED, there is a turn-on voltage of about 0.6 volts per diode, which the microcontroller can output on each pin. Since the RGB LEDs' diodes will be the only components powered from the ESP32 pin in that circuit, a common cathode is preferred to a common anode to forward bias the diodes. Each diode has its own voltage source, an ESP32 pin to the diode, and will always be forward biased when the pin is outputting voltage, since all three diodes are connected to ground through the common cathode terminal. With the color configuration being red, yellow, and green, and the common cathode connecting the LED to ground, the RGB LED is a contender for the LED of the electronic feedback system.

3.8.3 LED Selection

The selection of the LED comes down to one core specification: pin usage. The microcontroller will be directly communicating and powering all LEDs, weight sensors, and the touchscreen. Pin management is vital for the sake of device connections to the ESP32.

LED Type	**Single-Color LEDs**	**RGB LEDs - Common Cathode Pins**	**RGB LEDs - Common Cathode Strip**
Number of LEDs per Location	3 per Reservoir - Green, Yellow, Red 1 (or 2) for Deposit Site - Green (and Red)	1	1
Number of Pins (per LED)	1	1 or 3	3
Voltage Draw (per diode)	3.3 V	3.3 V	3.3 V
LED Cost (per LED)	$0.05	$0.95	$1.85

Table 12: LED Comparison

In comparing the types of LEDs, the single-color LEDs will work best for the prototyping stage. Controlling one LED per pin is preferred within the test and design phase for building the code behind the electronic feedback system. Using RGB LEDs will require more voltage overall because of the yellow light requiring the red and green LEDs to be powered simultaneously.

Additionally, single-color LEDs only require one pin from the microcontroller to be active, versus the RGB LED's three pins for three diodes (or one from the two-pin, but separate coding must be done on the on-board controller for that RGB LED), meaning all choices will take up the same amount of pins from the microcontroller at the reservoirs. If implementation of the LEDs is added to the deposit site, however, the single-color LED is the best choice. The LED needed for the deposit site is the green LED at minimum, and the green and red LEDs at most. Using the RGB LEDs, strip or pins, would be inefficient from a pin perspective.

When the software design for all the systems are in place, it may be a goal to change from the single-color LED configuration to one of the RGB LED configurations, at least for the reservoirs. Since the blue LED is never used for the purpose of the electronic feedback system, it can be disconnected entirely, saving a pin. The change is dependent on the testing phase though, whether or not the microcontroller can sustain additional voltage outputs for an extended amount of time when the yellow light needs to be produced.

Alternatively, having an RGB LED would allow for more show elements for the Automated Bartender, something that is important to the idea of this creation. Disregarding pin management on the ESP32, having a full range of colors would be a nice-to-have addition for the sake of showmanship from the Automated Bartender. When pouring, the RGB LEDs could be set to flash randomly via the pins turning on and off, and capacitors could be used within the circuit of the RGB LED to cause soft transitions between colors, allowing for more variability. Testing must be done first, though, to see if it is worth including in the Automated Bartender.

3.9 Software

The software application on the Automated Bartender is responsible for managing multiple sensors, performing calculations, controlling the flow of liquid through valve switching, displaying real-time data, and more. With that said, selecting the most suitable software development environment is crucial to write the optimal program that meets the specified requirements for these functionalities since it directly affects the drink dispensing time, the volume of liquid poured, and the response time of the touchscreen display. Factors to consider for the effectiveness of the application environment include:

- Programming Language
- Development Environment

3.9.1 Programming Language

The programming language that is to be utilized has a few critical considerations. The primary consideration which needs to be taken for the language is its compatibility with embedded applications. For this, proximity to the hardware becomes critical. That is, the language we select should be just as high-level as necessary to adequately achieve the goals of the project. This will allow us to limit memory usage which is a valuable resource in embedded systems. Additionally, hardware adjacency allows us the opportunity to optimize sections of code more specifically to the hardware. For example, if we were to develop in the C language, we could optimize the execution of any time intensive functions by writing them in assembly language instead of C. Another consideration made when selecting a programming language is the library support and availability. For many of the components – namely the touch screen – being used external libraries will greatly aid in the development of software. Other less-than-critical considerations include language complexity, community support, compilation/execution time, and language familiarity.

3.9.1.1 C

The C programming language is a common base language for many other languages, including C++, C#, and others. Because of its procedural structure and ease of use, it is the first programming language taught to the great majority of programmers and engineers, including our group members. This is a popular choice because it is a low-level programming language, which means it is machine-independent and compatible with all types of operating systems; it has a large and diverse library for being one of the oldest and most widely used languages in the world; it is a fast executable language when compared to others, such as Java and Python; and it is a simple and versatile language. However, due to its adjacency to hardware, C can result in longer, more complex sections of code. All things considered, C is an excellent general-purpose language that can be used for nearly all types of applications.

3.9.1.2 C++

Since C++ is a programming language extension for C, it has all of the great characteristics that were previously mentioned. The primary distinction is that C++ is an object-oriented language, which means it supports classes and objects. Thus, it means that inheritance is possible, allowing for the utilization of previously created objects in code or for the implementation of new objects by modifying their features. Furthermore, C++ is a high-level language, meaning that commands are similar to human-readable language, making it more straightforward to write instructions for the computer without having to modify them for the particular computer on which they are being processed. These factors make this programming language popular for GUI and embedded system applications, which are the two primary components for the application of the Automated Bartender.

3.9.1.3 MicroPython

Python is a programming language known for its easy code readability and simple syntax; these two characteristics are achieved through the Python language's unique compilation process, which uses an interpreter to read the high-level language program, evaluate it, and convert to machine code; this is what differentiates this programming language from C or C++ because they are precompiled. As a result, it is a more readable and friendlier programming language, but it is slower than the other alternatives as a result to the compilation process. The option in question is MicroPython, which is essentially a software implementation of the Python 3 programming language intended for microcontrollers; thus, it is designed to implement high-level language into low-level hardware devices, making it a great choice for object-oriented and embedded programming. The disadvantage of this option is that installing the MicroPython firmware on Windows can be quite hard and its execution time is slower than the other options.

	C	**C++**	**MicroPython**
Language Complexity	Moderate	Moderate	Easy
Execution Time	Fast	Fast	Slow
Memory Usage	Low	Moderate	High
OS Compatibility	Large	Large	Large

Library Accessibility	Large	Large	Medium
Community Support	Large	Large	Medium

Table 13: Programming Language Comparison Table

3.9.1.4 Programming Language Selection

C was chosen as the programming language for creating the Automated Bartender application. The deciding factors for this choice are that the two best development environments for programming the ESP32 board are the FreeRTOS-based ESP-IDF and Arduino IDE both support C; therefore, it offers many well-documented libraries and code examples that can be referenced to help the development of our application, and it is also the programming language in which all members of the group are most familiar and have more experience compared to all other options discussed.

3.9.2 Development Framework

Based on the programming language selection, the development framework options to be compared, all of which make use of the FreeRTOS-based Espressif ESP-IDF , are making use of the IDF directly, Arduino IDE and Visual Studio Code.

3.9.2.1 ESP-IDF

The ESP-IDF is Espressif's official IoT Development Framework for the ESP32 microcontroller series. This option offers a software development kit that is excellent for any general application development, but it is particularly well suited for IoT projects due to its wide variety of tools, APIs, features, and libraries that are wanted for these types of applications. Because it completely supports both the C and C++ programming languages, this development framework is intended to provide a comprehensive and versatile platform. Some of the features that make this an appealing choice for the development framework for the development of the Automated bartender are that it provides a wide range of drivers that support different sensors, peripherals, and communication protocols, which simplifies the connection and programming of the most important parts of the project. Other noteworthy features include Wi-Fi and Bluetooth connectivity, as well as support for over-the-air (OTA) updates and secure boot, allowing for cloud-connected devices. Because ESP-IDF is open-source and publicly available on GitHub, there are many code examples that can be used to verify component functionality and connectivity, as well as referenced to develop the full code program.

3.9.2.2 Arduino IDE

The Arduino Integrated Development Environment, or Arduino IDE, is another open-source development platform with a user-friendly interface for beginners. This is a software development platform that allows communication and program transfer to any Arduino hardware; however, the ESP32 microcontroller can be programmed using this development framework via a simple configuration process that requires installing an additional board URL developed by Espressif. It also provides all of the benefits listed in the ESP-IDF, such as a wide range of tools, libraries and drivers which help in the programming of numerous hardware components, and enable the use of the ESP32 WiFi and Bluetooth functionalities. The main difference is that this option only supports a simplified version of C and C++ that is tailored for the needs of the microcontroller; thus, depending on the complexity of the program to be developed for the Automated Bartender, this option could restrict the functionality and flexibility of the code.

3.9.2.3 Visual Studio

This development environment is among the most commonly used options for programming. Visual Studio Code is a free coding editor developed by Microsoft for Windows, Linux, and macOS. What makes it highly appealing is its versatility—it can be used for programming in any language. It offers features that support various development operations, including debugging, task running, and version control. For programming an ESP32 microcontroller, Visual Studio Code becomes an even more powerful tool with the addition of an official extension provided by Espressif. This extension can be downloaded from the VS Code Marketplace. It incorporates all the features discussed previously for the ESP-IDF option. These features include the ability to build, flash, monitor, debug, trace, configure the GUI, and more. Considering these capabilities, Visual Studio Code stands out as an excellent all-purpose development framework.

	ESP-IDF	Arduino IDE	Visual Studio
ESP32 Configuration	Easy	Moderate	Moderate
C Language Support	Full	Limited	Full
Flexibility	More flexible	Less Flexible	More flexible
Drivers Compatibility	Large	Large	Large
Library Accessibility	Large	Large	Large
Community Support	Large	Large	Large

Table 14: Development Framework Comparison Table

3.9.2.4 Development Environment Selection

Overall, all three development framework options are excellent for programming the ESP32 microcontroller for the Automated Bartender. When considering the Arduino IDE option, it presented significant advantages, particularly its extensive library and strong community support. However, its limitation in fully supporting C became a drawback, especially given the complexity anticipated for the code. Additionally, the Arduino IDE has very limited debugging and development support when considering the overall size of the project being taken on. That is, there are unclear and limited errors and explanations, especially for development in C, and while community support can often help debug and interpret these error messages, there is no guarantee that the error which we encounter has been encountered and solved before. Therefore, Arduino IDE is not well suited for the applications of this project.

The decision then came down to choosing between ESP-IDF by Espressif and Visual Studio with the Espressif extension. Both options offered similar support for code development. An additional element that was considered when selecting where to develop this project was the support and extensions. When considering the vast support offered on Visual Studio for debugging and software development, especially those tools related directly to the ESP32 platform, we deemed that Visual Studio would likely be the easiest environment to develop in. However, Visual Studio is not inherently set up as an embedded development IDE. Especially when compared to development platforms such as TI CCStudio, the built-in support for embedded development is less than desirable. However, due to Visual Studio's overwhelmingly widespread use, there are many packages which help to bridge this gap to make it a more suitable environment for embedded software development, namely packages like Platform IO which creates projects specifically for the board selected. When combined with the Espressif-IDF these packages convert Visual Studio into a more than sufficient embedded development environment. Ultimately, the choice hinged on the ease and comfort of programming for the group members. Considering the diversity of computer systems which our group members utilize, including Windows and macOS, Visual Studio emerged as the preferred option due to its flexibility and user-friendly interface.

4.0 Standards and Design Constraints

For the applications of this project, it is necessary to consider the relevant standards applicable to the project. Especially in consideration of the contact with food, standards become very important to reference for the safety of use of the machine.

4.1 Related Standards

4.1.1 IEEE 829-2008

This standard lays out procedures and documentation for testing of any software-based system. The standard lays out the need for a Master Test Plan (MTP) which is then supported by a series of Level Test Plans (LTPs) which test individual elements or levels of the software or device requirements. From these documents come a variety of others which include test design, scope, criteria for success, procedures, logs, anomalies, and finally, reports on the success of each test performed. The purpose of theses standards is that the user would determine the overall functionality, safety, and success of a software-based system through the determination of the integrity level of individual tasks, the robustness of the tests used, detailed criteria for each testing task including procedure and success criteria, the level of integrity these testing levels suggest, and the documentation related to these tests.

For the applications of this project, our software testing will begin at the lowest level: simple control and communication with each individual element. Items such as motors for each pump and communication with pressure sensors, touchscreen, and LEDs will need to be tested to ensure that the software can properly control each element individually. This helps to rule out the possibility of configuration issues as the software begins to integrate these features together to create a more complete product. This phase of development will need to be mid-to-high-integrity as success at a higher level of robustness here helps to guarantee any possible failure as the software is developed lies in the integration of technologies together, not in the configuration of any one technology individually. In this, we will need to test the precision of control of the motors for the pumps to determine the accuracy of each pump controlled. Coupled with stringent testing of the valve systems to further ensure accuracy of the liquid dispensed, this will likely be the most stringent string of tests as proper measurement of liquid dispensed is paramount to the success of this project. Additionally, the accuracy and precision of the pressure sensors being used as well as their interfacing with the ADC on the MCU will need to be tested to further ensure accuracy of the liquid dispensed as well as the safety of the system to prevent possible overflow failures. Also, we will need to test the accuracy of the touchscreen system to ensure that each user input will properly be recorded. After each of these systems are tested, we will record the results and, if needed, generate new test plans and criteria.

Once each subsystem is tested individually, their integration for the sake of safety will need to be tested together. This includes the combination of the pump systems, load cells, and valve systems to ensure liquid dispensing accuracy, the combination of load cells and LEDs to showcase liquid levels remaining in the various containers, and the combination of the touchscreen system and the pump systems, to ensure there is a proper response to user inputs. The results of these tests will be recorded and first evaluated at the integration level; that is, it will be determined if the combination of these subsystems meet the necessary criteria for the project. Should they not, it will then be determined if the issue lies in the integration of the systems or an individual subsystem. For the former, a new test plan and criteria will be developed and the software will be modified. For the latter, we will return to the previous lower-level testing of a system to evaluate the success of the individual subsystem at fault.

Finally, a test plan will be generated for the entire system success according to the criteria laid out in the project objectives, specifications, and requirements sections. Below is a summary of this plan for testing.

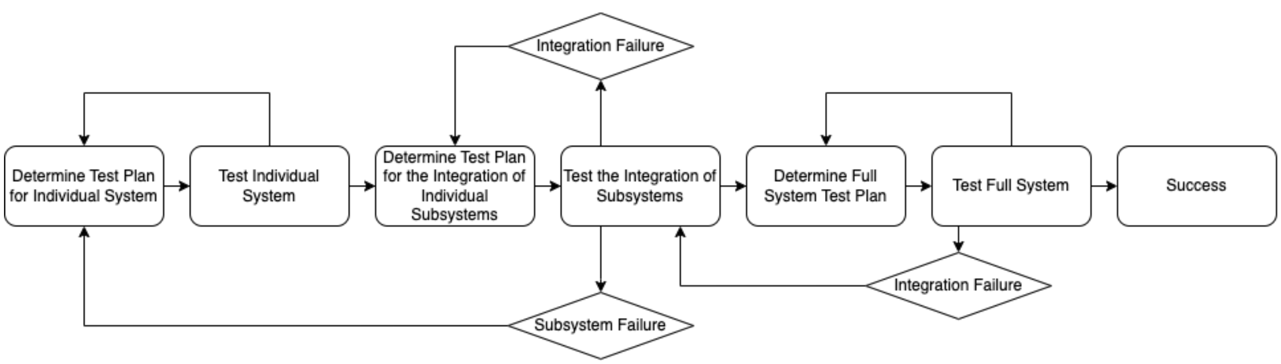


Figure 14: Test Plan Flowchart

4.1.2 Inter-Integrated Circuits (I^2C)

While standards for I^2C are de facto, due to its widespread use in this project, it is beneficial to discuss its standard features and function. The NXP UM 10204 lays out the specifications of I^2C technology as well as how to use it. The document covers the standard configuration of the technology using serial clock (SCL) and serial data (SDA) lines to interface different elements together, the function of slave and master devices, the start (S), stop (P), acknowledge (ACK), and not acknowledge (NACK) signals, the order standard signals are used to send and receive data, addressing details, and more complex standards (multi-master systems, arbitration, clock stretching, etc).

For the applications of this project, this standard is most helpful with the configuration of the I^2C module and its standard functions which will need to be developed in software. For example, following standard protocol in reading sensor data or utilizing standard settings when selecting the mode . Additionally, the manual lays out the various operation speeds available in I^2C, of which, our processors considered can utilize both standard and fast mode.

4.1.3 Serial Peripheral Interface (SPI)

Serial Peripheral Interface (SPI) is another de facto standard communication protocol that will be utilized in this project for the integration of the touchscreen device. Similar to I^2C, SPI has a slave-master configuration, but functions via a series of shift registers to shift data between devices. There are many variations of the configuration of SPI, but some important configuration points include the directionality of communication (full-duplex, half-duplex, or simplex), clock polarity and clock phase modes, and multiple device integration (individually selected or daisy chain). While the multiple device integration is not necessary for the applications of this project, the operation mode and communication directionality are important considerations. For a touchscreen a full or half duplex configuration will likely be necessary to control what data is displayed as well as receive feedback from the touchscreen.

4.1.4 Food and Safety Standards

The success of the Automated Bartender relies on a comprehensive analysis of various health and safety regulations that need to be researched and incorporated into our design. This review and assessment of these food and safety guidelines are essential to the design of the Automated Bartender. These standards will dictate the materials and methods that can be used in the design of this project. Below is an overview of the critical food and safety standards the Automated Bartender must adhere to.

4.1.4.1 NSF/ANSI 61

The first health standard reviewed for the Automated Bartender is the NSF/ANSI 61. This health standard, developed in 1989, establishes the minimum standards for chemical contaminants, components, and products that will come into contact with drinking water. This standard establishes the requirements that products must adhere to that come into contact with drinking water and do not introduce harmful substances into the water supply. The scope of the materials and products covered by this standard includes protective barrier material, joining and sealing materials, mechanical devices, and process media. All of the guidelines described in this standard cover products that are created by the use of good manufacturing processes and generally recognized manufacturing processes. This standard will govern the type of silicon tubing, pumps, containers, and valves that can be used in the Automated Bartender project. If this standard is not followed, it can cause other people to get sick when using this product and cause costly last-minute changes to the design of the Automated Bartender.

4.1.4.2 NSF/ANSI 25

The second health standard reviewed for the Automated Bartender is the NSF/ANSI 25. This standard, developed in 1989, established the essential food safety and sanitation guidelines for the design, manufacturing, and materials used in vending machines that can dispense foods and beverages. The design aspect of this standard calls for vending machine materials to be able to resist wear and the penetration of vermin into the machine and prevent cleaning materials and chemicals from coming into contact with consumables. If our group ever decides to sell this product commercially, the Automated Bartender will have to be evaluated to ensure it meets the strict guidelines of this standard. If the Automated Bartender does not meet these guidelines, alternative and approved components will have to be procured.

4.1.4.3 NFPA 170

The third standard reviewed for the Automated Bartender is the NFPA 170. This standard outlines and establishes various symbols to communicate fire safety, emergency and associated health information. The symbols used in NFPA 170 are uniform and easy to interpret to reduce confusion and make it easier to communicate the symbol's meaning. The Automated Bartender will contain fluid-filled lines and containers of flammable alcohol and electrical wires that can cause a shock. Due to these circumstances, the Automated Bartender will have to display symbols that warn of the potential electrical shock if not handled properly. The next set of symbols to be displayed are a flammable liquid and a no-smoking symbol.

4.1.4.4 NSF/ANSI 51

The fourth standard reviewed for the Automated Bartender was the NSF/ANSI 51. This standard establishes guidelines for plastic materials and components that are being used in food equipment. This standard also establishes guidelines on what materials can come in contact with food and beverages. This guideline will cover components ranging from tubing, sealants, gaskets, and valves. These guidelines will dictate the type of tubing and connectors that can be used in the Automated Bartender.

4.2 Design Constraints

When it comes to developing any project with extensive hardware and software needs, there will have to be realistic constraints and limitations in place during the development cycle. The Automated Bartender will have several key design constraints including economic, ethical, health & safety, manufacturability, time, sustainability, environmental, social, testing, and political. Our group will be actively working to optimize these constraints during development to fit the scope of the Automated Bartender best and to produce a successful and functioning product. These design constraints will serve as guidelines for us to follow during this project.

4.2.1 Economic/Budget Constraints

The cost of the Automated Bartender is the biggest concern of the group since this project will be completely funded by its four group members. Various Automated Bartender platforms similar to our project can be found for sale on the internet, varying from $300-$500. Our group will try to stay in the price range of the prebuilt bartending platforms. It is estimated that each group member will contribute approximately $125 to the development and building of the Automated Bartender. One of the criteria for selecting parts in the component research section is to find parts that will give us the best value without having to compromise performance. We will leverage components with large community support behind them by using open-source hardware and software to reduce development costs and improve the speed of development times.

Extensive testing will be performed at every stage of the development of the Automated Bartender. This extensive testing will help to identify any possible areas of the project where we could save money and/or improve the overall quality and performance of the system. For any specialty parts that can be made "in-house," we will employ a DIY philosophy to further aid with budget constraints and reduce the overall project cost. Some of the available options to aid with the DIY philosophy are employing the use of 3D printers to manufacture structural aspects of the project, for example, the frame of the unit or conduits to run electrical wires and other cables.

4.2.2 Health, Safety and Ethical Constraints

Alcohol is one of the main elements of this project; due to the overall nature of alcohol, this will impose significant health, safety, and ethical constraints. The Automated Bartender will follow strict health standards since this project will dispense drinks for human consumption. All of the components that will come into contact with the fluids will be thoroughly examined and researched to ensure it is food-safe. The Automated Bartender will follow various health and safety standards to ensure its drinks are safe for human consumption. The standards followed are the NSF/ANSI 61, NSF/ANSI 51, NSF/ANSI 25, and the NFPA 170 more information on these standards can be found in section 4.1.4.

Depending on who you ask and their personal values, some individuals will either agree or disagree with creating a machine whose main purpose is to dispense alcohol beverages and that the time and effort spent developing this machine can be spent benefiting society. Society as a whole agrees that abusing alcohol or any controlled substance can cause negative effects on that person using or on the people around them. The Automated Bartender is a platform that will dispense alcoholic beverages to consumers who are exclusively of legal drinking age, and they provide explicit consent to consume alcoholic beverages. When this project was first pitched, our group as a whole had to decide where we stood ethically and morally with respect to alcohol consumption and building a device whose primary purpose was to dispense alcoholic drinks.

The ethical constraints of this project are directly related to the effects that alcohol can have on a person's health and their personal and professional life.

Alcohol consumption in excess can cause adverse health effects and unfavorable effects on a person's personal and professional life. Excess alcohol consumption can cause a person to develop hepatic steatosis, also known as fatty liver disease. This disease can start out as not a serious issue, but over time, it can lead to total liver failure or cancer of the liver. If a person were to stop consumption of alcohol and seek medical treatment, this condition may be reversible. If a person continues to consume alcohol with hepatic steatosis, this can cause cirrhosis of the liver, which is when the liver develops scar tissue, therefore killing off the healthy cells of the liver.

Another adverse health effect of excessive alcohol consumption is an increased risk of developing alcoholic hepatitis. This occurs when a person consumes more alcohol than their liver can process. When this occurs, the liver will start to become damaged, and the person will start to develop hepatitis C. Development of hepatitis C can cause its own complications that can propagate through the body. Alcoholic hepatitis can be reversed if a person discontinues alcohol consumption. If a person continues to consume alcohol with alcoholic hepatitis, this will lead to cirrhosis of the liver, which is not reversible.

Excessive alcohol consumption can cause serious effects in an individual's personal life; these effects can range from impacts on personal relationships to work to impacts on personal work performance. If these issues are allowed to escalate this can lead to loss of personal relationships with loved ones and friends and loss of employment. The Automated Bartender cannot dictate all of the possible ways that alcohol will affect a person. It would be very difficult to inform the possible user of this platform of all of the possible effects that are associated with alcohol consumption. The United States Government is the entity that dictates the legal drinking age. The legal drinking age was set to 21 when Congress passed the National Minimum Drinking Age Act in 1984. The government is the entity that controls how alcohol can be legally consumed and disturbed. The Automated Bartender is a platform that will be operating within those guidelines set by the United States government. All warnings and possible negative effects of alcohol can be summarized by the Surgeon General's warning, which will be posted on the exterior of the Automated Bartender platform to mitigate these possible constraints.

4.2.3 Testing Constraints

Extensive testing will be performed at every stage of the development of the Automated Bartender. This extensive testing will help to identify issues as they happen and prevent those issues from propagating further into the development process. When components are initially received, they will be thoroughly tested to ensure they perform as expected. Doing this initial testing early will allow for modifications to easily be made if necessary. The results of the extensive testing will serve as checkpoints that can be referred to throughout the development process. These checkpoints will capture component configuration and design parameters at that point in time. These checkpoints will be used in troubleshooting and the planning of future features of the Automated Bartender. During the testing and prototyping of the Automated Bartender, all of the instances where alcohol may be used will be substituted for colored water.

4.2.4 Social/Ease of Use Constraints

When using the Automated Bartender platform, it is imperative that the user experience is easy to follow and straightforward. When using the software on the Automated Bartender, we must actively try to mitigate any possible cases that may cause confusion when someone is using this platform. During the development of the software and integration of the hardware, we will give extra attention to the user experience to reduce any possible confusion that may arise and make the experience of using the Automated Bartender platform easy and enjoyable. One of the steps that we will take to mitigate these possible concerns is to make all of the text on the touchscreen display appear in a clean and organized fashion.

4.2.5 Manufacturability Constraints

There are many considerations that need to be made when considering the manufacturability of this project. Among these constraints is the availability of parts selected. It is possible for parts not to be in stock that we are using in our design as well as for parts to only be available in bulk quantities, which is not applicable for this project. These kinds of factors will put constraints on part selection and usage as we can not make use of a part which is unavailable. Another critical factor which could constrain the manufacturability of this project is shipping times. Part suppliers may have items in stock or on order, but these parts could have too great of a lead time for the applications of this project. Therefore, we will need to consider lead time as well.

These manufacturability constraints could pose a detriment to other constraining factors as well, namely economic and time constraints. For example, if a part is out of stock, we could be forced to purchase a more expensive alternative in order to guarantee the manufacturability of our project. Additionally, when considering the possible lead times for parts delivery, while a desired part could have a longer than average lead time, this will need to be weighed against a possible alternative which could have a shorter lead time or wider availability.

4.2.6 Time Constraints

Time is another critical constraining factor which needs to be considered for this project as we only have two semesters to research, design, construct, test, and troubleshoot a functional proof-of-concept design. With that a careful consideration will need to be made when considering the possible features for the Automated Bartender. That is, certain features of the design may need to be revised or removed from consideration in response to these time constraints. For example, while a feature such as a robotic arm would certainly aid in meeting the goals of this project, adding to the authenticity and showmanship of the design, when considering the time constraints of this project, it is likely not the best use of time.

Perhaps the most important constraint that time poses on this project is the feasibility of features which we have determined as possible for our project. While certain features have passed the theoretical phase of development, when it comes to proper implementation of these features, we are prepared to make any design changes necessary to meet the time constraints for this project. For instance, one goal of this project is to make use of load cells to indicate liquid level remaining in the storage receptacles. While this in theory passed the feasibility requirements (unlike features like the robotic arm mentioned above), should this prove too time-intensive to implement, another path will be utilized to determine the remaining liquid levels in a storage receptacle.

Additionally, these time constraints impose the necessity of considering priority of features as well. Features such as the load cell weighing system or dispensing system are paramount to the proper functionality of the project, therefore, with respect to the time constraints, these features will need to have a higher degree of development time allocated to them. Whereas, a less critical feature such as the aesthetics of the human-machine interface will need to be allocated less development time due to the constraints. While features like this are still important to dedicate development time to and certainly are important to the success of the project, they are not function-critical aspects of the project.

4.2.7 Sustainability Constraints

This project is designed to operate reliably for upwards of one year, therefore, it is necessary to consider the sustainability of function of the project. Should parts require routine maintenance, this severely hurts the sustainability of this project. While these constraints are not paramount to the success of this project, they are critical elements in quality assurance for the product being delivered. To mitigate the impact of these factors, we are not considering any battery-operated or dependent systems which could create a need for routine replacement. Additionally, we are considering components which have an average lifetime of greater than one year of near-constant operation, and, while we do not anticipate near-constant operation of our project, this should help mitigate the sustainability constraints imposed.

4.2.8 Environmental Constraints

Similar to sustainability constraints, environmental constraints likely do not pose a major impact on the success of this project. However, it is beneficial to consider external factors surrounding this project. While this project will not face environmental constraints in the conventional sense (outdoor conditions such as heat), one critical environmental constraint imposed upon this project is its proximity to liquids. Being designed to control liquids, this project will involve electronics in close proximity to open liquid containers often. To mitigate the risk this poses, careful consideration will need to be taken for the placement of electronic components as well as possible waterproofing of components or the mounting of the entire electrical system. For example, electronics will need to be mounted high and away from the open cup being dispensed into, and components near storage receptacles, the open container, and the surface on which the machine is set, need to be protected from water damage.

4.2.9 Political Constraints

Political constraints will arise when our product can be used to possibly alter or influence the political preferences or personal beliefs of an individual. This product does not have any design aspects that could possibly be used to alter personal and political beliefs. Therefore, we have deemed as a group that this project will not have any relevant political constraints.

5.0 Comparison of ChatGPT and other Similar Platforms

ChatGPT is the most widely recognized AI platform because it was the first to be launched back in November 2022 by OpenAi; however, many other similar platforms have been announced since then containing different features; thus, the following discussion will compare ChatGPT with some popular alternatives and look at limitations, advantages, and disadvantages that each has, as well as describe how any of these platforms can affect the research process for the Automated Bartender.

5.1 ChatGPT

Starting with ChatGPT , its name precisely describes the service provided by this platform; it stands for Chat-Based Generative Pre-Trained Transformer, which essentially means that it is an artificial intelligence trained with an extensive amount of publicly available data on the internet by the end of 2021. Its key features include the ability to respond to any question asked in real-time in a chat manner using simple language in a conversational tone, and it can also generate anything based on a prompt such as creating stories, writing essays, translating texts, proofreading, prompt programming, and much more.

Even though it can generate any response for any input from the user, there are some limitations that come with using this platform, such as the fact that the information shared on the response given by ChatGPT may be incorrect or outdated because its databank is limited until the end of 2021, and the platform may be overloaded with the number of users and go out of air at times. The use of this platform for learning purposes has both advantages and disadvantages. For example, it allows the user to ask tailored questions on any topic, such as questions that the student came up with while studying the topic, and it can prompt the chatbot to explain in simpler terms some definitions and concepts taken from books or scientific articles.

The drawbacks include the uncertainty that the information provided by the platform may not be entirely true. The fact that it cannot browse the internet and thus is unable to reference back to its source or provide any links, and the fact that if the platform is used but not credited, it is considered plagiarism may result in the user committing an act of academic dishonesty.

5.2 Bard AI

Google's Bard AI is a really interesting alternative to ChatGPT. The most desired feature of the platform is that it is integrated with Google search; thus, it solves the biggest limitation with ChatGPT by having up-to-date available data and being able to have links of the web sources where the information used from the chatbot was collected from. This solves one of the negative aspects of using ChatGPT for learning purposes. In addition, this AI tool can be very helpful for component selection and research, as it can access customer comments from the internet to help answer specific questions about the product or the overall customer experience with the item in question. This is a feature that needs to be highlighted because of its potential benefits for the research process during senior design.

5.3 Elicit.org

Elicit is essentially an AI platform for research help; because its knowledge is based entirely on validated sources such as books, journals, research papers, and published theses, it has the capability to be an incredibly effective learning tool. This platform detects relevant papers on a given topic by interpreting the prompt through key words and context, generating full summaries and a list of crucial points for each document collected; these are some of its strong aspects. However, because this platform is not as complex and cannot be trained with as much data as ChatGPT, it is probably a good practice to use this AI tool in conjunction with ChatGPT. This way, anything generated by ChatGPT can later be verified with the help of Elicit, and the sources that prove the information can be accessed and cited later.

5.4 Consensus

Consensus is another useful artificial intelligence tool for research. Unlike Elicit, Consensus chat interface works based on yes or no questions. The chatbot will respond with a yes, maybe, or no response, but its overall answer is based on the most agreed and cited response from a range of academic sources analyzed by the tool. In this case, it will determine the percentage of papers that agree with the answer being yes, as well as the percentage of papers that believes it should be no. This platform is helpful for asking direct inquiries and displaying highly regarded publications that support the chatbot's response depending on how many times the paper has been mentioned. However, some of its drawbacks include the fact that it may only provide a true or false response to a prompted question, which means that its response may be biased based on the research papers in its database and may not provide the user with sufficient or objective information. Similar to Elicit, this platform makes it simple for users to access and cite the sources analyzed for the generation of the chatbot's answer.

	ChatGPT	**Bard AI**	**Elicit.org**	**Consensus**
Applications and Use Cases	Creative writing, prompt programming, research	Creative writing, prompt programming, research	Research	Research
Price and Access	ChatGPT 3.5 is free; ChatGPT 4 (Plus) is $20 per month	Free	Free	Free
Pros	Most well know platform Extensively trained and tested Very reliable	Connected to the google search engine, allows to prompt any query Information is retrieved from various websites, offering real-time and accurate answers to the prompt	Fast and efficiently extracts information from various relevant credited sourcers Provides concrete, detailed and well-explained summaries of collected sourcers	Great platform for direct yes or no answers to questions Final answer is supported by the overall conclusion of the most used sources on the topic asked
Cons	Inaccurate and outdated information might be provided due no internet	Very new platform, it shows some difficulty understanding and answering	Limited to only research questions No answers offered, only	It answers only to research questions where it can only be

	real-time search	some prompts It can be slow, and biased sometimes	abstract summaries of the relevant sources found on the topic prompted	answered directly as yes or no

Table 15: AI Platforms Comparison Table

5.5 AI Tools Utilization

While there are benefits to having a smart tool such as Chat GPT to condense, reiterate, and provide information, there is a cost with how it can be used. Students especially find a use for Chat GPT and AI tools to condense sections or whole books to relevant information, and while not inherently malicious at this point, can take this condensation as their own and attempt to submit it. Besides essay writing, it is possible to solve assignments using specialized AI tools. To reiterate, it is not inherently malicious to use AI in this manner, if used as a tool to supplement study and assist in guidance to solving a problem; it begins to be a problem when the student copies the answer with no intent to learn the topic behind it.

As much as AI can be a tool to provide information, it also collects information submitted to it, saving it to servers to be another piece of comparison data. This can and has compromised a lot of concealed documentation, and many companies no longer allow external AI tools such as Chat GPT any access to these documents. In response to this, companies are creating AI tools that exist solely on internal servers, allowing a secure, tailored AI tool to the work and businesses of that company.

Additionally, tech companies such as Intel are creating next generation tools and components that allow a local AI on personal computers and laptops. Similar to company-specific AI, this can provide secure comparison data on a local space that the user can control. Intel with its 14th generation chips will include a new type of unit, a neural processing unit (NPU), that will house this local AI. One test case given specifically is a lawyer scanning a new case and comparing it to previous ones they have done. This allows optimal comparison to data that the user is familiar with and can trust without the risk of a breach of security on an online AI.

Apple is also investing heavily into AI, with the new flagship M3 chips allowing for AI capability within their Mac products. Additionally, a glance at the current hiring trends within the company shows a focus on Artificial Intelligence / Machine Learning (AI/ML) career positions, hinting at the company's focus in research and development of AI capabilities. AI development is a big push in the tech industry now, and its priority across many companies shows how integral AI is intending to be with technology to come in the next generation.

Many different industries see the use of AI as a way to assist in the day-to-day work of employees, analyze large amounts of business data and identify trends, hence the strong push for localization of AI. In some instances, though, industries have attempted to circumvent human initiatives and concerns. The film industry, for example, has attempted to circumvent the Writers Guild of America (WGA) strike by using AI to write the scripts of new movies, shows, and other media. AI in this instance is harmful to the industry, removing opportunities for writers to find work and suppressing their right to unionize for improved conditions. In recent negotiations, the use of AI was limited for use in script writing to protect the writers and to avoid the legal litigation on the ownership of ideas that come from AI tools.

With AI seeing much more widespread use, companies are looking to innovate AI tools in many ways to meet market demands and concerns. What third-party sources cannot produce themselves, companies will produce themselves for their needs. AI is making a rapid pace to become an industry standard across all fields for the betterment of production and innovation.

5.6 AI Applications to the Project

When considering the applications of this project, AI tools could prove beneficial in the research and development of this project. There are a variety of unique systems and processes being combined to create the Automated Bartender, and the use of AI would definitely streamline the research of the non-electrically based systems that would be integrated. However, we made very limited use of these AI tools to avoid many of the inaccuracies and difficulties that they pose in their current phase within the market. While these tools have applications surrounding this project, they were not used for research or sourcing any components or processes implemented in the Automated Bartender.

5.6.1 Software Development

For the applications of this case study, we will consider the use of AI as it relates to software development, specifically to applications in debugging and error recognition. The primary issue with software development when it comes to an embedded environment is the proper classification of errors; often, error messages given by an IDE or compiler are not helpful to the development of software as they are auto-generated based on a set number of possible outcomes, requiring the developer to dig around the internet and through the lines of code surrounding the error to find the true error.

With the help of AI tools, specifically ChatGPT due to its generative, pre-learned nature, these errors can be interpreted into plain English, going beyond the predetermined error message and explaining the possible natures of the error and how that corresponds to the code written to aid in the development of the software. Additionally, ChatGPT can function as a debugger for smaller sections of code, accurately identifying syntax errors and possible compilation and application issues in a section of code. This is greatly beneficial to the experience in senior design as we will spend a large amount of time with the software being developed, and, after an extended period of time, an extra set of highly trained eyes can be highly beneficial to software development but may not always be available.

However, it is important to note that AI, especially ChatGPT does not always have accurate information, and, while the platform is trained extensively, may not be able to properly diagnose errors in the software development process. Additionally, once a program has been input to ChatGPT, there is no limitation to its subsequent use, even if it is unintentionally reused by the AI software to solve another user’s problem. Artificial intelligence, while not perfect in its applications to software development could be used in this project for the sake of debugging and error handling.

To further illustrate this a sample code block with known errors was input into first an IDE and the errors were identified by the pregenerated responses provided for error handling in that IDE. Then, those errors were written into ChatGPT and asked to be explained. Finally, the code itself was input into ChatGPT, and it was asked to identify errors. The below table summarizes the results

Error	IDE Response	ChatGPT Explanation	ChatGPT Response
Correct: int n=6; Incorrect: int * n=6;	Incompatible integer to pointer conversion	“typically occurs when you're trying to assign an integer value to a pointer variable without proper type casting” Included proper examples	“The issue with your code is that you're trying to assign an integer value directly to a pointer, which is not allowed in C. “ Included an explanation with examples and the corrected code.
Header file not included in directory	“test.h” not found	“typically indicates that the C compiler cannot locate the header file named "test.h" that your source code is attempting to include.” Included possible underlying issues watch could cause this error	“You have an #include "test.h" statement, but you need to make sure that the header file "test.h" is present in the same directory as your source code file or that the correct path is provided.” Includes a corrected code.

(All quoted responses: OpenAI's ChatGPT, private communication, 29 November 2023)

Table 16: ChatGPT Software Case Study Results

5.6.2 Topical Development for Research

One of the most applicable test cases for this project is how AI technology relates to research, especially its usefulness in considering what material to research. While we, as developing engineers, have a wealth of knowledge surrounding the components of this project and have the means to find and recall the knowledge necessary to develop this project, given that we have not developed anything of this nature and on this scale before, there are likely gaps in the considerations made in our research to ensure the full scope of a component, software strategy, or design tactic is considered.

AI tools are very useful in this instance to help us understand possible areas for deeper research. ChatGPT as a tool can be helpful due to its generative model. For example, when considering valve types, we could ask for types of valves to consider, their applications in an embedded system, factors to consider in electronically controlled valves, and various other elements surrounding valves. These considerations can then be coupled with Elicit to find articles surrounding the factors mentioned by ChatGPT. The primary function of these AI tools with respect to research is brainstorming and discovery of resources. In these applications, AI tools can be very beneficial to making the proper considerations surrounding technology selection and component selection for this project.

However, this application could also prove harmful for this project, as these AI tools have no way of identifying the importance of factors surrounding the components and technologies being researched. That is, we as the researcher must be able to identify what elements of the AI output are actually important to the applications of this project and what elements are not directly applicable to our research. Should we fail to do this, we may research technologies which are not applicable to this project, be misled in the feasibility of components, or collect irrelevant information surrounding components being researched. This means that in isolation, AI tools are likely more harmful than helpful in the brainstorming and idea development process. Instead, we must couple the output of AI tools with our applicable knowledge and independent research to determine a hierarchy of the responses provided to us.

5.6.3 English Language

Another beneficial application of AI tools for this project is the use of their understanding of the English language and proper grammar. This is especially important when considering that one of our group members does not speak English as a first language and that none of us are actively practicing proper grammar. Additionally, many of our group members are bilingual and do not always speak or use English. Therefore, AI tools which go far beyond spell check could prove very valuable for checking and correcting errors in our research.

One of the most important elements of this project is the ability to accurately communicate the design and the logic behind that design utilizing English. Clarity is of the utmost importance, and AI tools can be utilized to ensure we clearly articulate the purposes of this project. Additionally, AI tools can go beyond just interpreting writing and correcting grammatical mistakes, it can evaluate the overall flow of a paper and ideas. We could provide AI with a list of headings to ensure that the research flows and does not detract but aids in the clarity of communication we desire to have.

However, AI tools are not the perfect English language experts. It is easy, just as in day to day language, for AI to misinterpret the meaning of a word or phrase in the context of a written piece. This is especially true when considering the context of this paper which is highly technical in nature. This is even apparent with non-AI tools such as spell check or grammar check, which often misinterprets acronyms or technical words or phrases and suggests their replacement. While in theory, AI tools will have a better understanding of these technical elements of this research than non-AI tools, it is certainly important to consider that the editing of a paper with AI tools can very easily result in inaccuracies in the written research when compared to the original intent of the statement. Therefore, caution should always be exercised when using AI in this way. Grammatical changes will not be copied directly from AI tools but will be verified before their inclusion in our research.

6.0 Hardware Design

For the Automated Bartender Platform, the aspect of hardware design will serve as the cornerstone that will bring together a functional prototype. This section will cover the core circuits and their schematics that will be used in this project and provide a visual representation of the various subsystems and how they connect and a structural illustration of the Automated Bartender platform.

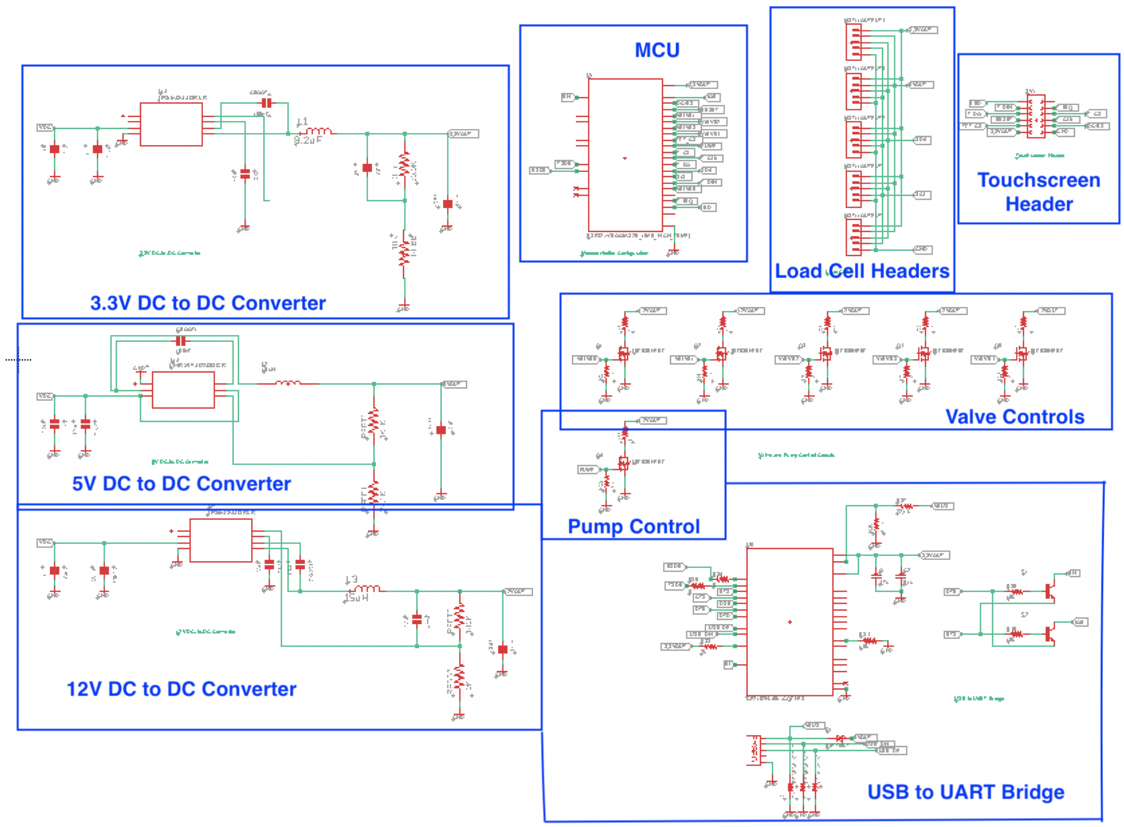


Figure 15: Full Main Schematic

6.1 Subsystem Block Diagram

This block diagram of the various subsystems of the Automated Bartender gives a high-level overview of the various subsystems that are in this but more importantly how those systems will connect with each other to form the final prototype. Each subsystem will be comprised of various circuitry and software that have been integrated together to achieve the desired functions. The Automated Bartender will be comprised of various subsystems that relate to power delivery, processing/control unit, user interaction/interface, fluid delivery, and force sensing components. All of these components will make up the automated bartending platform. At the core of this project is the microcontroller that will serve as the brain of this machine and it will communicate with all of the other subsystems. The fluid delivery subsystem will include the peristaltic pump and the solenoid valve; this subsystem will be responsible for pumping liquids out of the reservoirs into the users cup and opening and closing the valves to the appropriate reservoirs additionally this subsystem will be controlled by the microcontroller. The user interface subsystem will be comprised of the touch screen display and LEDs; this is where the user will interact with the automated bartending platform. This subsystem will allow the user to customize their drinks and this is where any messages or warnings will be displayed to the user.

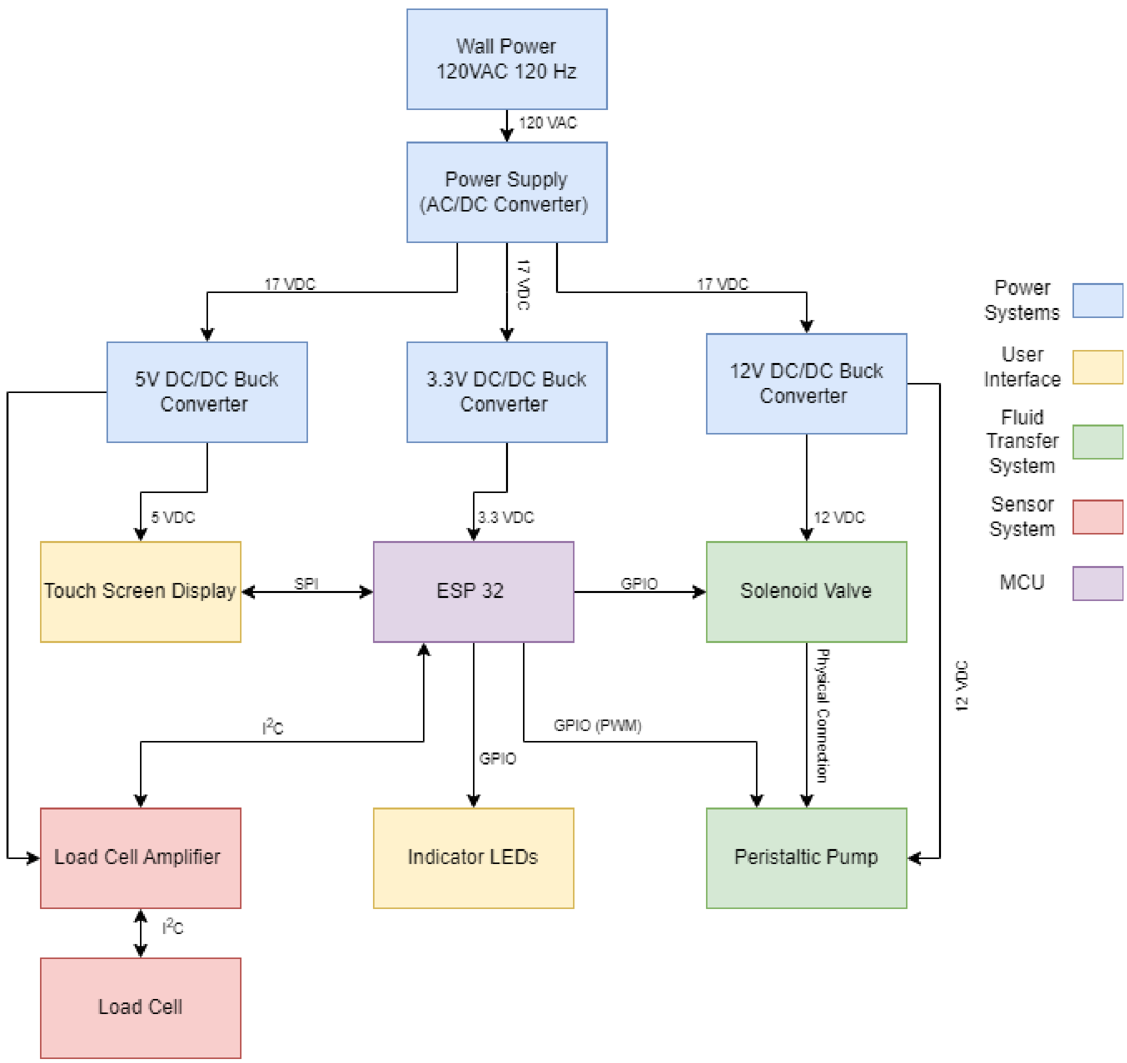


Figure 16: Subsystem Block Diagram

The LEDs will be used to indicate the liquid levels in the reservoirs. The sensor subsystem will comprise the load cell; this is how the automated bartending platform will track the amount of liquid dispensed and the amount of liquid remaining in the reservoirs. Finally, the power subsystem will consist of an AC/DC converter and various DC/DC converters. This subsystem is responsible for providing power to the entire project. This subsystem will take 120VAC wall power and step down and rectify then distribute into the appropriate power bus.

6.2 Converter Schematics

The Automated bartender will be composed of various components and subsystems that will require different levels of power to operate. It is imperative for these components to be supplied with the appropriate power to ensure proper functionality. All of the DC/DC and AC/DC converters discussed in this section are all in the class of switch-mode converters. Below are the various schematics that will be used to achieve these goals, these converters will be DC/DC and AC/DC converters, these converters will be placed onto a PCB.

6.2.1 3.3V DC-to-DC (Buck) Converter Schematic

The buck converter schematics were created using the Texas Instruments Webench online service. This valuable service generates various DC-to-DC converters based on specific design constraints set by the user, such as the input voltages and desired output voltage and current. Webench will generate schematics based on the inputted design constraints. The design of the converter can be further customized based on cost, efficiency, and footprint of the circuit. All of the DC-to-DC converter circuits have been created to have a balanced design between cost, efficiency, and footprint of the circuit, and they are created with common components that are readily available to ship. All of these converters will be using a buck converter topology to step down the input DC voltage into the appropriate output voltage.

The first schematic created was the 3.3V DC-to-DC converter, as seen in Figure 17. This converter will have a steady state efficiency rating of 92.4% and will have a build-of-materials cost of $0.88 and a footprint of 493 mm^2. The input voltage of this converter will be 14V to 22V, and the output will be 3.3V with a steady max output of 2A. This converter will operate by using the TPS62932DRL chip and various RLC components. This IC will employ a forced continuous current modulation which will help to reduce output ripples when connected to a load. A 32nF capacitor is connected to pin 7 of the IC to set the minimum soft start time of the IC to approximately 1ms. The enable pin is left floating this will enable the IC. The feedback pin is connected to a voltage divider this will be used to achieve the desired 3.3V output of the DC/DC converter. Input of this converter will be connected to the output of the AC/DC converter and the output will be used to create a 3.3V power rail.

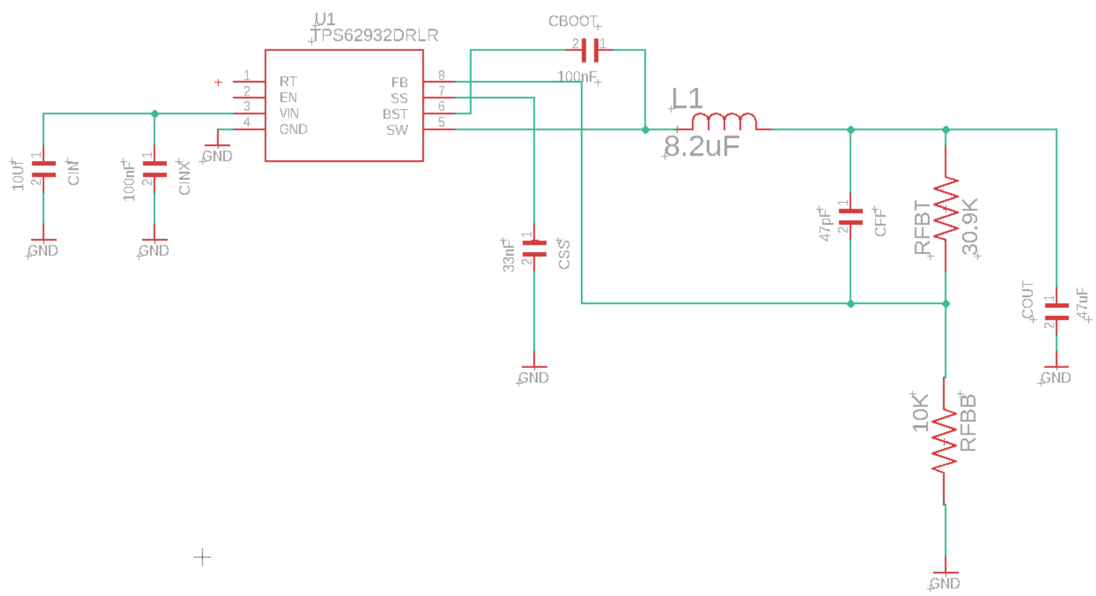


Figure 17: 3.3V DC-to-DC Converter Schematic

6.2.2 5V DC-to-DC (Buck) Converter Schematic

The second schematic created was the 5V DC-to-DC converter, as seen in Figure 18. This converter will have a steady-state efficiency rating of 93.1%, a build-of-materials cost of $1.05, and a footprint of 502 mm^2. The input voltage of this converter will be 12V to 22V, and the output will be 5V with a steady max output of 2A. This converter will operate by using the LMR51420DDC chip and various RLC components. This circuit will have various safety features that will be implemented internally onto the IC. These safety features will include a current limiter, short-circuit protection and a thermal shutdown function in case of the event of too much heat being dissipated across the chip. The enable pin on this IC will be tied to Vin this will keep the device enabled. The feed back pin is connected to a voltage divider that will be used to achieve the desired 5V output of the DC/DC converter. This IC will have a soft start and various compensation circuits that are implemented internally on the device to allow this IC to operate with the minimum amount of external components. Input capacitors Cin will serve as a high frequency decoupling capacitor for the converter and the capacitor Cinx will be placed close to the device pins and will serve as a bypass capacitor. An LC filter is used at the output to filter any unwanted voltage or current spikes that can be caused by the transient of the load currents. Using these features will help to drive down the cost of the converter. Input of this converter will be connected to the output of the AC/DC converter and the output will be used to create a 5V rail to power various components needing a 5V source.

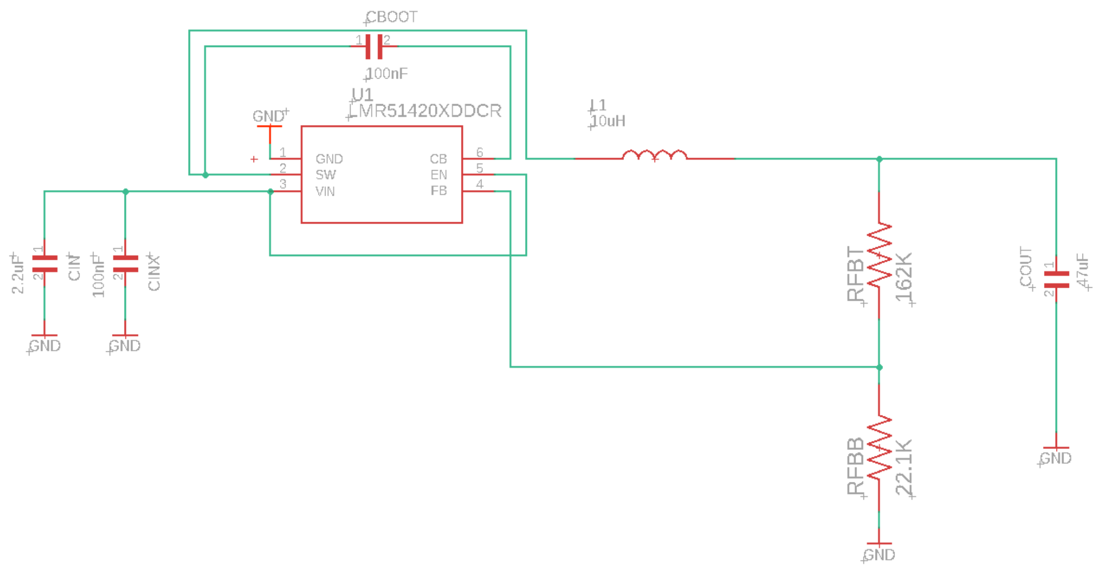


Figure 18: 5V DC-to-DC Converter Schematic

6.2.3 12V DC-to-DC (Buck) Converter Schematic

The third schematic created was the 12V DC-to-DC converter, as seen in Figure 19. This converter will have a steady-state efficiency rating of 97.3%, a build-of-materials cost of $1.45, and a footprint of 259 mm^2. The input voltage of this converter will be 14V to 24V, and the output will be 12V with a steady max output of 2A. This converter will use the TPS62932DRLR chip. The 12V DC-to-DC converter will come with the same safety features and protections as the 3.3V DC-to-DC converter because they both utilize the same chip. The input of the converter will be connected to the output of the AC/DC converter and the output will be used to create a 12V power rail to power the various components in the Automated Bartender. The output of the converter will have a smoothing circuit that will smooth out any ripples that might occur from the IC or from the result of transients caused by load changes.

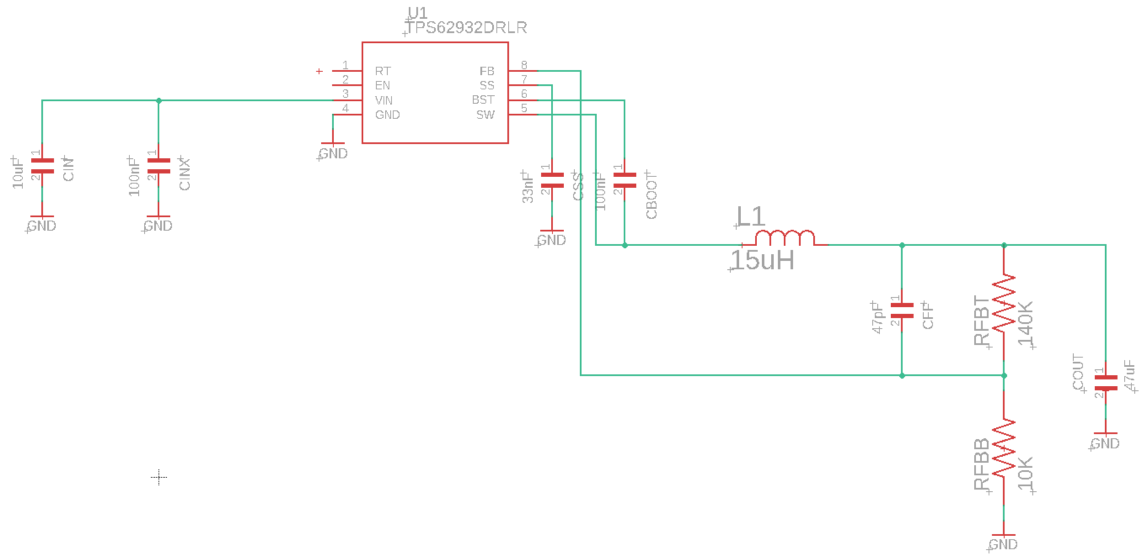


Figure 19: 12V DC-to-DC Converter Schematic

6.2.2 AC-to-DC Converter Schematic

The AC/DC converter schematic was created using the Texas Instruments Webench online service. When deciding on the design of the converter the size, efficiency, and topology of the converter were taken into strong consideration. The Automated Bartending platform will operate on a Flyback AC/DC converter topology due to its high efficiency and simplicity as discussed in section 3.x.x. This converter was chosen with parts that are readily available to be purchased and shipped. The only part of this converter that is not readily available to be purchased is the flyback transformer; this component will have to be custom made for this specific configuration. The specifications of the transformer have been provided by the Texas Instruments Webench online service.

This converter will have a steady state efficiency rating of 79.1%, and a build-of-materials cost of $9.45 and a footprint of 3322 mm^2. The input voltage of this converter will be 85VAC to 265VAC and the output will be 18VDC with a steady max output current of 2A. This converter will be placed on a PCB and will serve to power the entire project, the AC/DC converter will be placed on its own PCB to reduce the chances of component interference from happening due to the high frequency switching of the converter. This design will include a 5.11kΩ bleed resistor as a safety feature to ensure all energy storage devices will be completely discharged when the circuit is turned off. A snubber circuit is implemented into this design to handle voltage spikes that may occur from the high frequency switching. The snubber circuit will be a safeguard to ensure other components will not be affected by voltage spikes. This converter will come with some built-in safety protections like overload protection and under voltage protection. This converter will take 120VAC wall power, and it will step down and rectify wall power into an 18V DC signal that will then feed into the various DC-to-DC converters on the PCB. The rectification of the AC signal will be performed by the full bridge rectifier at the input and the voltage will be stepped down via the flyback transformer.

The output of this AC/DC converter will be an isolated output, meaning that there will be electrical isolation between the input and the output of the converter. This electrical isolation will be achieved by the use of the flyback transformer, the transformer will have a laminated metal core. Since this subsystem will involve working directly with wall power it will be imperative that we take the proper safety steps and precautions, during the development and testing of the AC/DC converter, we will seek guidance from various professionals to ensure our safety.

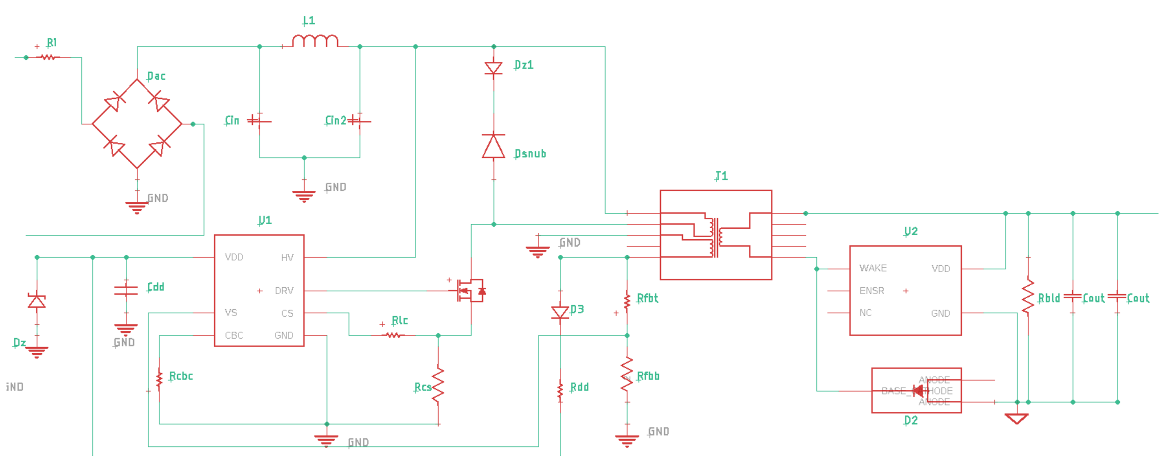


Figure 20: AC-to-DC Converter Schematic

6.3 MCU Schematic

The primary goal of the MCU schematic design is to establish a secure connection with all the various peripheral elements of the device. For elements such as the touchscreen, a pin header will be used; however, for most elements, the pin used for control will be routed on the PCB. The other critical consideration is the USB to UART bridge which is required to program the device. For the considerations of this project, the same bridge as used on the development board will be utilized. Including a picture of the MCU pinout would be superfluous for the applications of this paper; therefore only the USB to UART bridge is shown below.

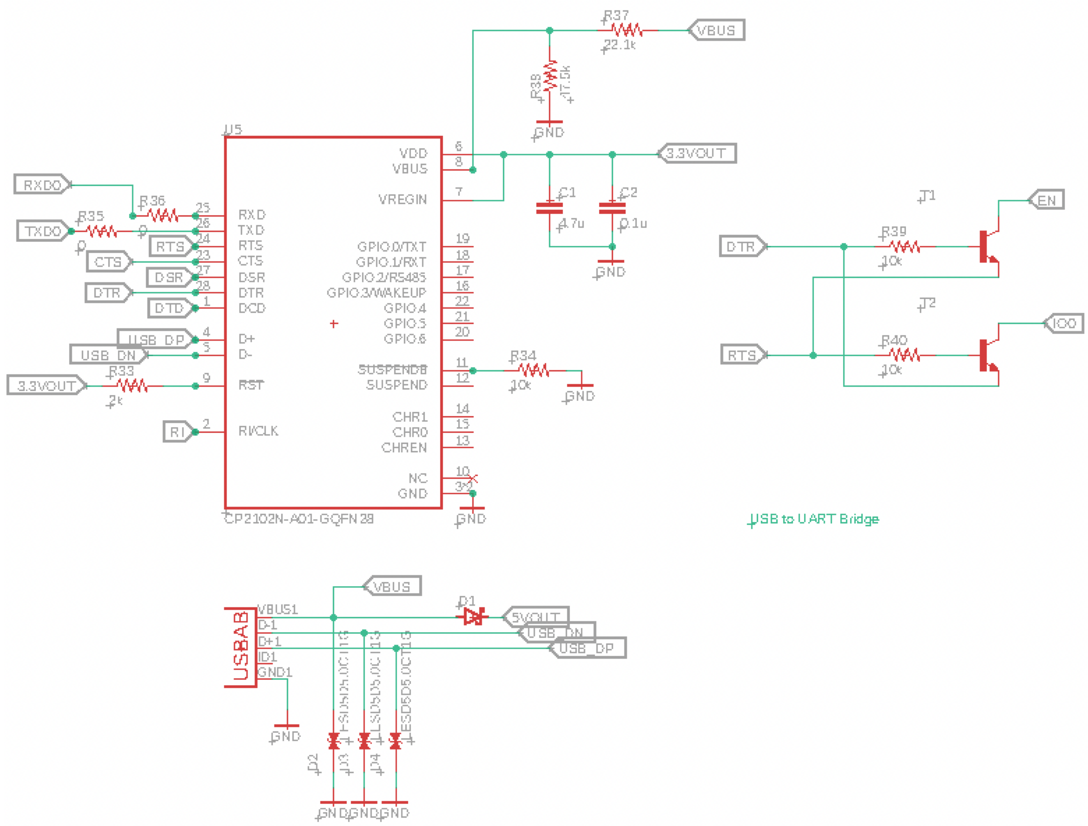


Figure 21: USB to UART Bridge

The purpose of this section of circuitry is to allow the processor to properly communicate with the computer being used to program via usb. Because the ESP32 has no built-in JTAG or bridging device, the signal from the USB jack – USB_DN and USB_DP is converted to a readable UART signal for the ESP32 through the CP2102N. Additionally, this configuration handles some typical boot functions to allow proper communication with the computer.

6.4 Touchscreen Schematics

For the touchscreen schematic, it is crucial to ensure that the correct display pins are connected to the designated GPIOs on the ESP-32 microcontroller, enabling the functionalities required for both displaying images and touch capability. The touchscreen module is a TFT LCD, where displaying images involves controlling an array of liquid crystal segments, each equipped with RGB filters to showcase desired formats and colors. Communication with the multiplexing columns and rows, responsible for turning individual pixels on and off, is facilitated by the IC driver built into the display—specifically, the ILI9488 IC driver in this case.

Communication with this display must occur through a 4-wire serial interface, utilizing SCK (Clock), SDI (MOSI), SDO (MISO), and CS (Chip Select). The SCK pin serves as the clock necessary to synchronize data transfer between the microcontroller and the IC driver; SDI (MOSI) is the pin used to send data from the microcontroller to the IC driver; SDO (MISO), the opposite of the previous pin, is used for receiving data from the IC driver to the microcontroller, and CS (Chip Select) is the pin to select the driver IC.

To connect the ESP32 to these display pins, the microcontroller's SPI peripherals are employed. GPIO 18 is connected to the SCK pin (VSPI_CLK function), GPIO 23 is linked to the SDI (MOSI) pin (VSPI_MOSI function), and GPIO 15 is connected to the CS pin (HSPI_CS function). Notably, the SDO (MISO) pin from the display is intentionally left unconnected to prevent the IC driver from sending data to the microcontroller. Instead, GPIO 23 is connected to the T_DIN of the display to receive data from touch input. Common pins like VCC and GND handle power supply and grounding. The RESET pin resets the IC driver, the LED pin controls backlight, and the DC/RS (Data/Command/Register Select) pin distinguishes between sending data and commands to the display. Pins T-IRQ, T_DO, T_DIN, T_CS, and T_CLK are responsible for configuring the touch functionality of the display. The image below illustrates the wiring connections used for the implementation between the ILI9488 TFT LCD display and the ESP-32 microcontroller.

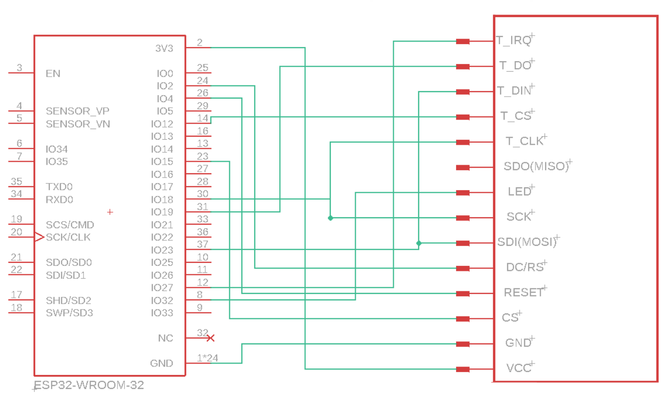


Figure 22: Touchscreen Schematic

6.5 Load Cell System Schematics

The following schematics comprise the load cell portion of the electronic feedback system. These schematics include the load cells, load cell amplifiers, and microcontroller, along with the proper connections to the pins for I^2C applications. The load cell amplifier has its own schematic in focus to reduce the overall size of the total schematic; this is to avoid losing details from components being too small within the schematics to be legible.

6.5.1 Load Cell Amplifier Schematic

The load cell amplifier schematic shown below is the piece integrating between the TAL220B load cell and the microcontroller. On the left hand side is the 5-pin header to the load cell, which connects to each node of the load cell's wheatstone bridge. The wire to each node is color coded, and represents a specific pin on the load cell amplifier: Excitation (+), Excitation (-), Signal (+), and Signal (-). On the right hand side is the 5-pin header to the microcontroller, which consists of VCC (3.3 V), serial clock (SCK), serial data (DAT), and ground (GND). These connections can be seen in Figure 24 on the HX711 Output 5-pin header next to the microcontroller.

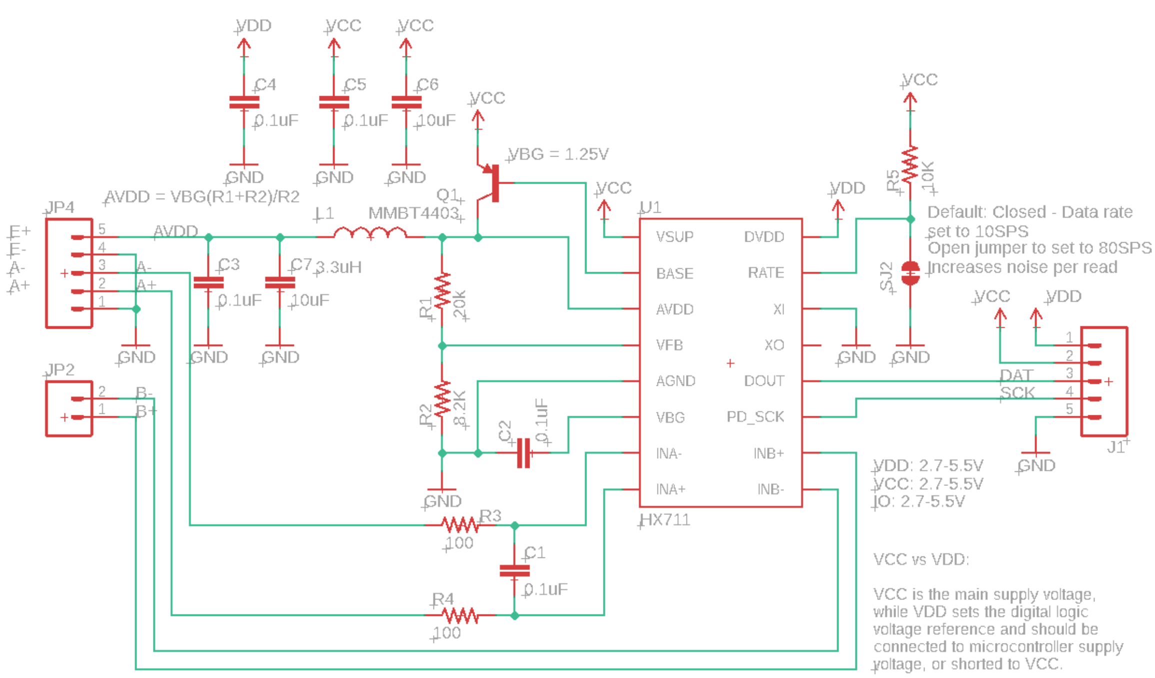


Figure 23: HX711 Load Cell Amplifier Schematic (From SparkFun [69])

6.5.2 Load Cell Schematic

The load cell schematic shows the load cell on the left, shown in its wheatstone bridge configuration, a simplified load cell amplifier schematic with its input and output pin headers, and the connections from the load cell amplifier output pins to the microcontroller. The input resistance for the load cell is measured around 1000 ohms, with an output resistance around 1050 ohms. On the HX711 output header, output 1 is ground, output 2 is serial clock (SDI), output 3 is serial data (SDO), and output 4 is VCC (3.3 V). See Figure 13 for HX711 load cell amplifier product figure.

Using I^{2}C communication, the load cells can be connected in parallel to one another, which reduces the amount of used pins. The connection to VCC and VDD is imperative for the load cells to ensure the data from each load cell is constantly updated for the microcontroller to update the LEDs.

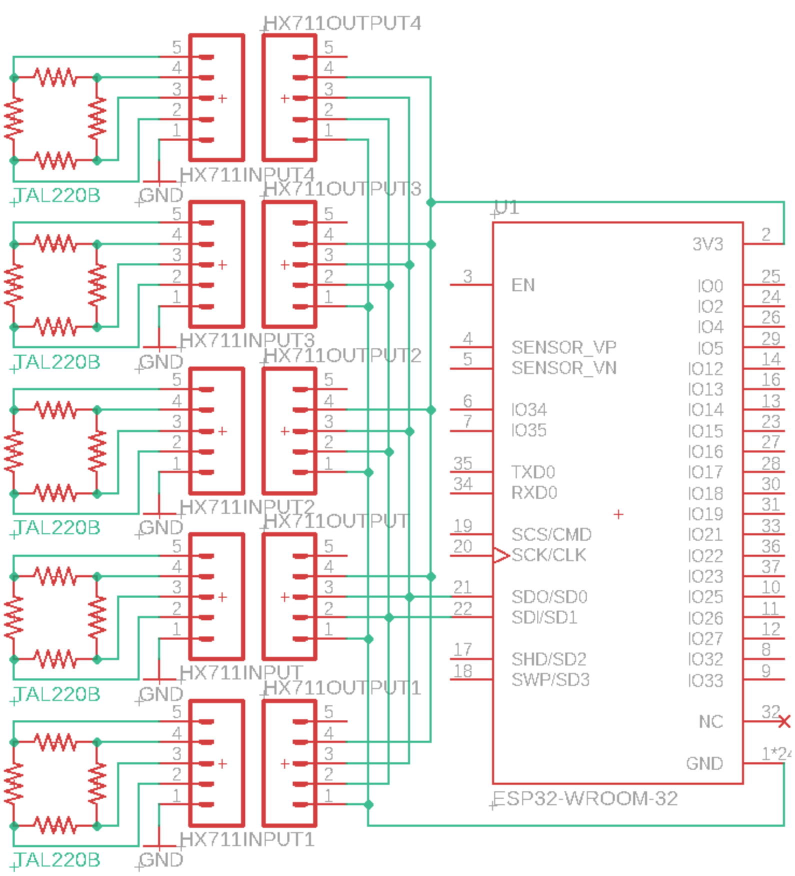


Figure 24: Load Cell Schematic with Simplified Load Cell Amplifier

For a more detailed view of the HX711 Load Cell Amplifier schematic, refer back to Figure 23.

6.6 LED Schematic

The LED schematic consists of the microcontroller and 12 LEDs, four red, four yellow, and four green, with a 220 ohm resistor on each node. With the 3.3 V output from the microcontroller, this should allow for 15 mA of current to run through the LED, providing enough illumination to give indication without the LED being difficult or uncomfortable to look at.

This configuration of LEDs takes up most of the microcontroller's pins, which causes concern for other devices that will rely on the microcontroller to activate. There are other planned configurations in the event the amount of LED pins needs to be minimized for other device changes, such as removing the green LED for nominal levels and only leaving yellow and red for consumption monitoring or even removing both green and yellow and using the red LED for low level warning and empty alerting in solid and blinking modes respectively. Note that LEDs omitted from the overall schematic until the more critical functions are verified and excess pins are determined.

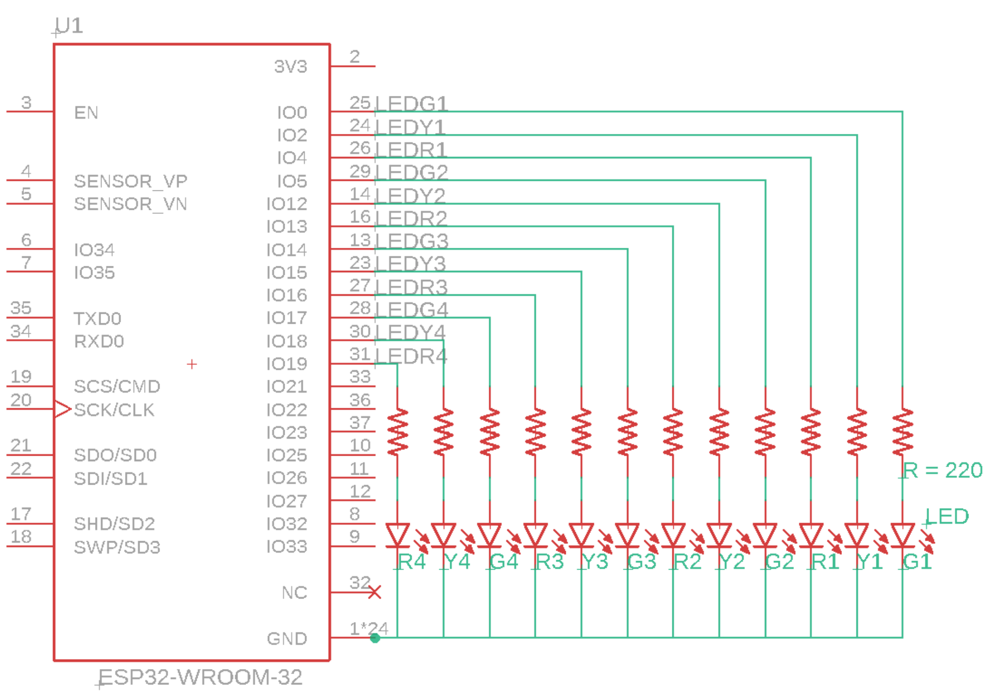


Figure 25: LED system connected to ESP32 Microcontroller

6.7 Pump and Valve Controls

The planned design for the pump and valve control systems is to make use of a MOSFET as a logic-controlled switch. The circuitry is designed such that when the MCU outputs a logic high signal to the gate of the MOSFET, the higher voltage output for the pump and valve systems will be in series with those components, causing them to activate. This is done in lieu of a relay for the sake of size. For the valves, this signal will always be constant logic high or logic low. For the pump subsystem, this circuit topology allows us the unique advantage of controlling the pump speed by sending a PWM signal to the gate of the MOSFET.

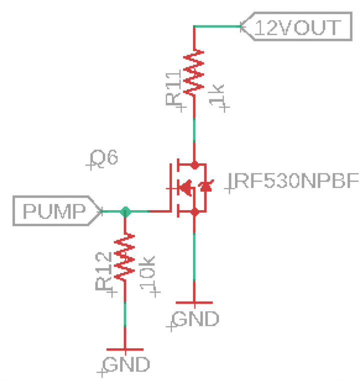


Figure 26: Pump and Valve Control Circuits

7.0 Software Design

The software design of this system is essentially a finite state machine, similar to the function of a vending machine, moving through various options and processes which allow for user customizability of the dispensed drink. While various states may call the same functions, the state the machine is in will affect the response of that function. For example, the dispense liquid function will be the same regardless of whether it is a manual dispense amount or a premixed option, the state of the machine, however, will differentiate the subprocesses necessary for each different state the machine may be in.

For a premixed option, the machine will refer to the various dispense amounts defined in the code whereas for custom dispense amounts, the program will need to pass the dispense amounts and units set by the user instead of the preset amounts. In both instances, the function to dispense will be the same. Because of this, a state machine may not be the perfect description as the function of the machine will not solely depend on the state it is in, but will also have a dependency on the previous state(s) selected. These can be stored as binary variables to determine which states have been passed through. In the previous example, there will be a variable which can be set or reset to determine whether the dispense function/state was preceded by the premixed state or the custom-dispense state. For the diagram below, diamonds represent the set or reset of a binary variable to track previous states while ovals represent the actual states being accessed.

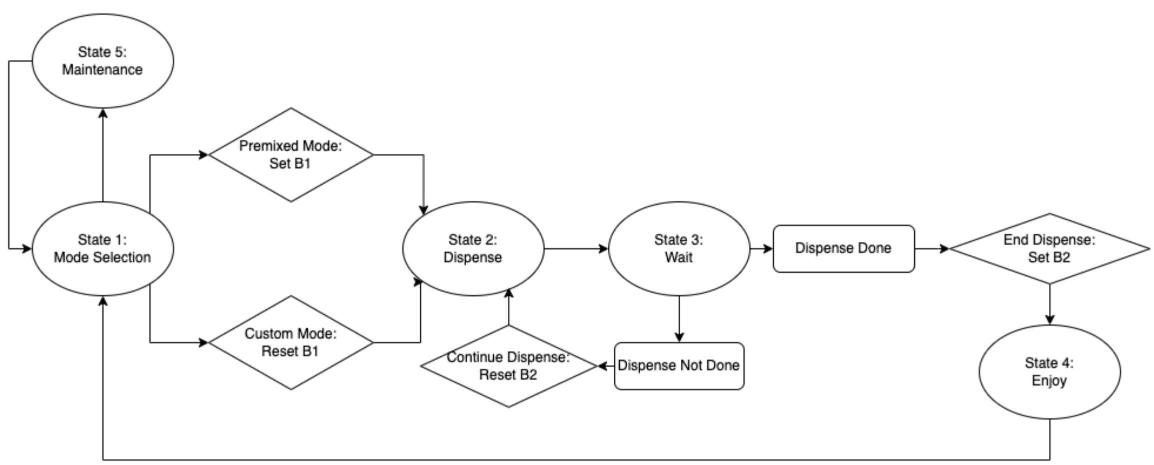


Figure 27: Simple State Diagram

The primary purpose of the first state is to allow the user to select the dispense mode. This will also serve as the home screen and most common state for our machine. It will employ the touchscreen subsystem and the LED subsystem. In this state, a cup must be placed underneath the dispense mechanism, sensed by the load cell under the cup, to proceed. Additionally, when this state is returned to from states four or five, the binary variables will all be reset to prepare the system for possible movement to state two.

The second state is the most important state as it will involve the touchscreen, LED, load cell, pump, and valve subsystems. It will be responsible for the accurate dispensing of liquids according to various premixed amounts or user input depending on the state of the binary variable B1. Additionally, this state will need to monitor the liquid levels in both the under-cup load cell and the level-monitoring load cells beneath the various reservoirs (updating the LED system in real time).

The third state is a primarily logic-based state which will determine whether the drink is finished dispensing or if there are other liquids which need to be dispensed. In premixed mode, this will solely be a logic driven task, and will not change the display for the user. However, in manual dispense mode, the user will need to be prompted to dispense another liquid type or complete the drink. Note that this will also set or reset another binary variable B2, which will have little effect on the next states. This variable is primarily for error handling and will serve as a safeguard to ensure the dispensing has finished properly: The setting of this variable will temporarily disable the pump and solenoid systems to ensure that there is no accidental dispensing.

The fourth state will be primarily to allow the machine to reset, displaying an “Enjoy” screen and preventing the user from beginning to mix another drink before the machine is ready. In this state, the load cell system will be employed one last time to verify liquid levels remaining in each reservoir and to ensure the cup has been removed before allowing a new drink to be mixed as a method of preventing overflow.

Finally, the fifth state will be a maintenance state which can be used for settings adjustment, system lockout, or any needs we see fit as developing the software.

State	Valve System	Pump System	Load Cell System	LED System	Touchscreen System
1: Mode	Not in use	Not in use	In use for cup	In use for	In use for

Select			placement sensing	display only	display and touch
2: Dispense	In use	In use	In use for cup monitoring and reservoirs	In use for editing	In use for display only
3: Wait	Not in use	Not in use	Not in use	In use for display only	In use for display in premixed mode; in use for display and touch in custom mode
4: Enjoy	Not in use: Locked out	Not in use: Locked out	In use for cup removal sensing and reservoir check	In use for display and edit	In use for display only
5: Maintenance	Available for use	Available for use	Available for use	Available for use	In use for display and touch

Table 17: Summary of State-System Interaction

Each of these states coupled with the various binary variables will directly determine the functionality of each subsystem. Because of this, each function call for the subsystems will need to be further subdivided into state responses to ensure proper function. This will likely be implemented in further function calls, unless the functionality is simple such as disabling the solenoids from opening or enabling the pump.

7.1 Pump Subsystem

For each state, the pump subsystem has two possible functions, either off (locked out) or on. In the off state, the output pin will be reconfigured as an input to effectively lock out the system and prevent any possible accidental spillage or negative pressure issues. This is critical as spillage could cause direct liquid contact with electronic elements and negative pressure could unintentionally draw liquid from reservoirs, put pressure on the valve system, or pinch off the tubing, effectively closing off tubes. Because of this, it is imperative that the pump does not have even the ability to activate in its off states.

In its on state, the pump system will need to operate in three different modes. The first mode is full on, which will constantly supply a logic high signal to the MOSFET controller by configuring the pin to GPIO and enabling a signal. The second mode is PWM mode, which will require the pin to be reconfigured to PWM. Then, a PWM signal will be output to the MOSFET controller to slow the operation of the pump motor. By providing an intermittent signal to the pump motor, the average voltage seen by the motor will decrease, resulting in a decrease of operation speed, allowing for more precise control of the liquid dispensed. The final mode is an on-but-not-pumping mode which will not output any signal to the pump system, but will keep the pin configured as an output.

When the pump system is requested for use, the program will first check for the current state; should the current state indicate the pump system should not be in use, the system will return an error and not activate the pump. The function will need the arguments of the amount of liquid to be pumped and the units of the liquid (should different unit options be implemented). The pump system will run in full-on mode initially until it reaches a threshold value which is within a certain amount of liquid of the target dispense amount. When this is hit, the motor will change operation modes to PWM mode to slow the flow of liquid as it reaches the desired amount. When the sum of amount of liquid remaining in the pump system manifold and the amount of liquid already dispensed is equal to the desired amount to be dispensed, the valve to the reservoir will be closed, the air valve will be opened, and the remaining liquid will be purged to avoid any cross contamination.

The final pertinent security information for the pump system is its redundancy check with the load cell system. The function for dispensing liquid via the pump will occasionally take a reading from the under-cup load cell to ensure that the amount dispensed as measured by the pump's timing is within a reasonable degree of the amount of liquid measured by the load cell. Below is a flowchart to showcase the function of the pump system.

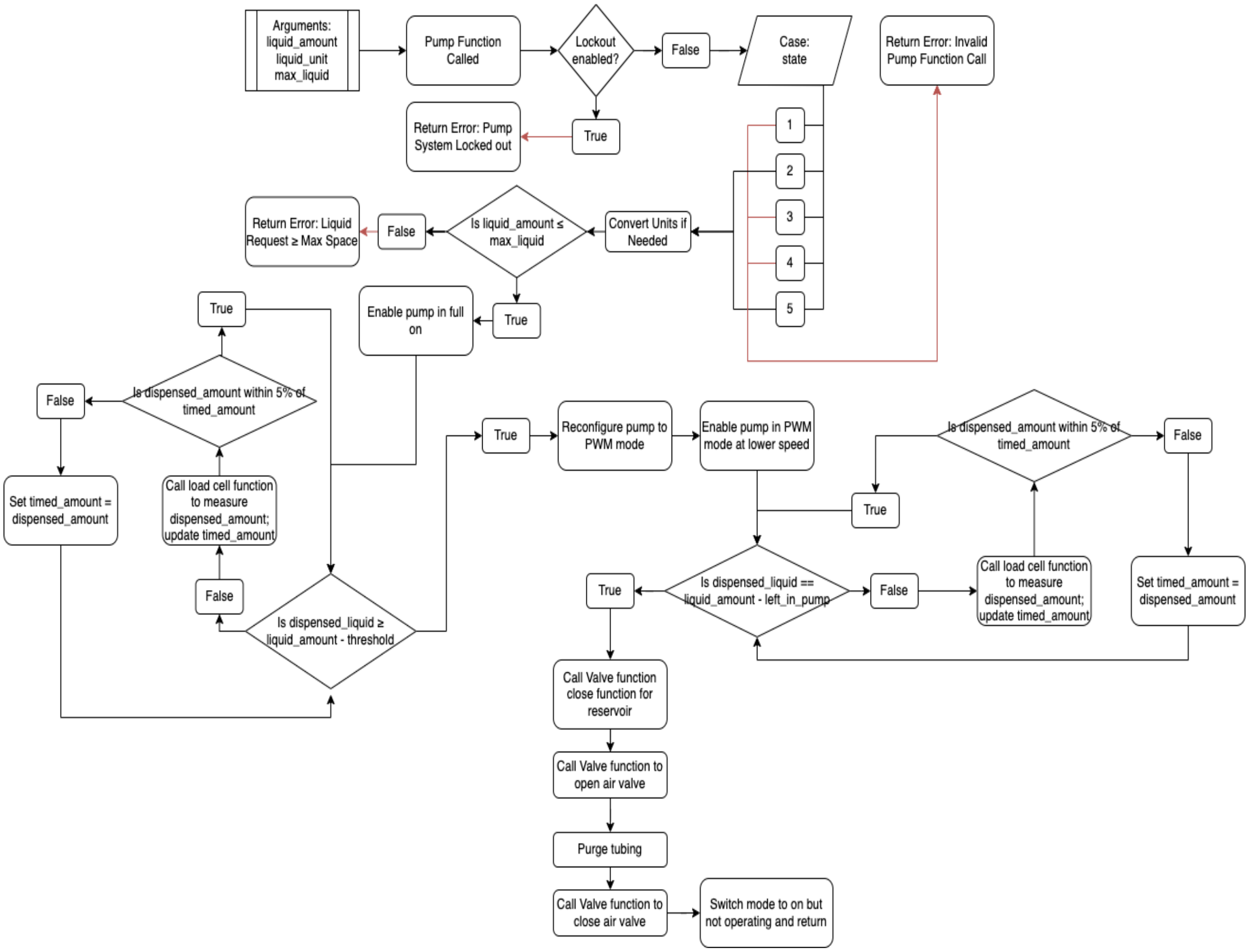


Figure 28: Pump Subsystem Block Diagram

Label	Constant or Variable?	Description
liquid_amount	Variable	Desired amount of liquid to be in cup at the end of dispensing
liquid_unit	Variable	The unit of measure being used; Global declaration
max_liquid	Variable	The maximum amount of liquid that can be dispensed in the cup, considering all liquids already in the cup

timed_amount	Variable	The amount of liquid that should be in the cup based off the timing of the pumps
dispensed_amount	Variable	The amount of liquid in the cup as measured by the load cell
left_in_pump	Constant	The amount of liquid that is past the valve and needs to be purged by the system before a new liquid is dispensed
threshold	Constant	The threshold at which the pump will switch from constant operation to PWM operation

Table 18: Pump Variables Summary

7.2 Valve Subsystem

The valve system will function in tandem with the pump system to control the flow of liquid. Because of the similar hardware design of the valve system to the pump system, it will have the same two functions: on mode, and off mode, where the pin is configured to prevent an input from being processed.

In the on state, the valve system will only be able to open or close; however, because of the direct effect they have on the flow of liquid, there will need to be extra security measures taken to ensure that no two valves are open at the same time and that no valve is unintentionally opened. To do this, an array of global variables will be declared to track the state of each valve. Before any valve is opened, the global array will be checked to ensure that all valves are closed. If all valves are closed, then the desired valve will be able to have its state altered. Coupled with the same security feature implemented in the pump subsystem to ensure the current machine state is a state in which the valve system is active, this should be adequate to prevent any unintended errors.

During pump usage, the valves will need to be opened via a logic high signal to the MOSFET controller, and when the sum of the remaining liquid past the valve and the liquid already dispensed reaches the desired amount, the valve will close by sending a logic low signal to the MOSFET controller. The below diagram shows the software function of the valve system, where the argument valve_number is the valve to be accessed and open_or_close is a binary variable which indicates if the valve should be opened or closed.

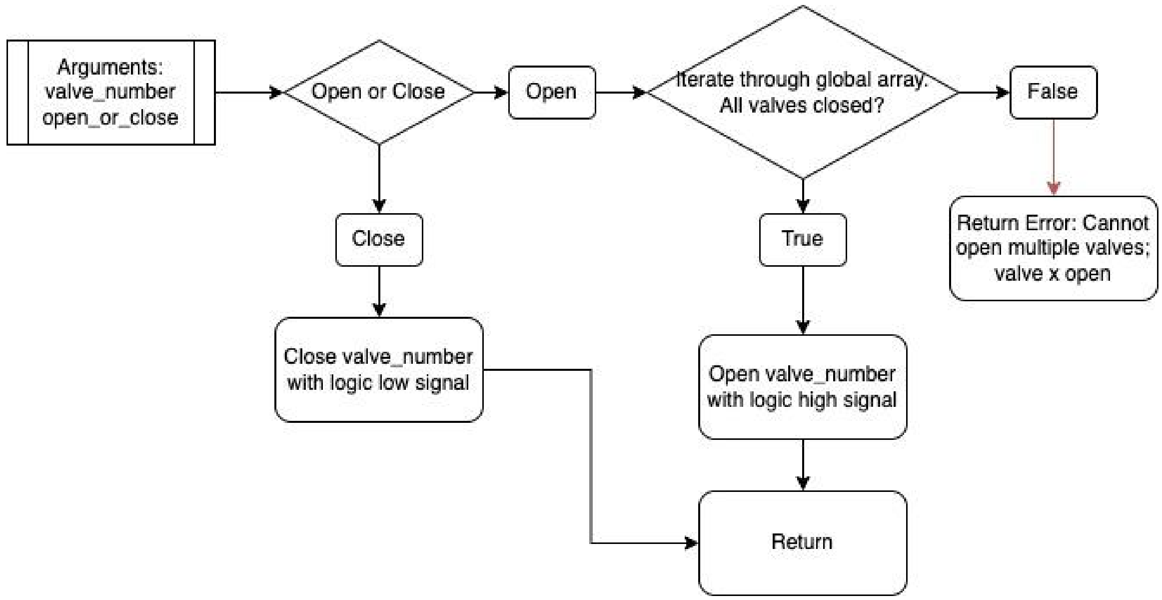


Figure 29: Valve Subsystem Block Diagram

7.3 Load Cell Subsystem

The load cell system is an integral sensor-based system which will interface to the ESP32 making use of I^2C connectivity. While the ESP32 is equipped with ADC capabilities, it is important to note that for accuracy and precision, we will make use of the HX711 amplifier circuit, which will convert the amplified signal separate from the ESP32. Therefore, the software design is simpler, involving only the configuration of the I^2C connection, but not the analog to digital conversion.

Upon startup, the I^2C ports will be enabled via a function call. This will configure the module in master mode, establish the serial data and serial clock ports, select the clock utilized for the module, and configure any pullup resistors. Once the module is configured, each load cell will be run through a brief test to confirm their functionality. This will be achieved by requesting the address of each load cell amplifier to be sent to the ESP32 one by one. This ensures proper communication protocol has been established.

Once communication is verified, the state will be tested via a case statement to determine the function of the load cells. In state one, a threshold value will be used to sense when the under-cup load cell goes from unloaded to loaded. The pump and valve systems will be locked out until the load cell senses a cup is placed beneath the mechanism, allowing for the safe dispensing of liquids. Additionally, the weight of this empty cup will be stored for use in calculations of liquid dispensed. In state two, the under-cup load cell will be utilized intermittently to measure an approximate liquid dispensed. Utilizing the difference between the empty cup weight measured in state 1 and the live measured weight as well as the density of water the approximate ounces dispensed into the cup can be calculated. Additionally, occasionally after this measurement and calculation occurs, the reservoir load cells will be read to update their current status. This process will be optimized to primarily read the reservoir currently being emptied, but will also measure the other reservoir levels occasionally for accuracy. Should any reservoir drop below the yellow or red threshold, this will raise a global flag variable which is checked at the end of each pour in state 4. In state 3, no load cells are in use. In state 4, the undercup load cell is used to determine that the cup is removed from the machine before the reset begins. Additionally, the reservoir levels are read one final time to verify their levels are accurate. In state 5, the load cells will be available for calibration. Essentially, the purpose of this mode is to remove any items and to “zero-out” or reset the load cell sensors to remove any inaccuracy in the measurement. This will likely need to be occasionally performed.

7.4 LED Subsystem

The function of this subsystem is simple as the updates will be directly linked to a function call which will read the status of global flag variables which will track the correct color LED which should be lit. These flags will be stored as an integer value where 1 corresponds to green, 2 to yellow, and 3 to red. When the function is called, no argument will need to be passed as these flags are global. The function will read the flag integer for each reservoir system and update the corresponding LEDs to match the status of the reservoirs.

7.5 Touchscreen Subsystem

The GUI is a crucial component in the design as it dictates how the Automated Bartender will function based on the choices made by the user through the touchscreen. The primary goal of the software design for the GUI is to create a very user-friendly interface while constantly monitoring the status of the Automated Bartender.

In the project narrative, the two main features emphasized are the ability to select from a list of pre-programmed drink options, each coded with fixed liquid amounts required for the requested drink, making this option considerably faster. The second feature is a custom drink mode where the user is free to mix any of the four liquid options in any amount, provided it does not exceed 12 ounces. With that in mind, the image below is a draft illustrating how the homepage will appear.

Figure 30: Homepage Draft

In addition to the main option to initiate drink selection with the two mode options, other features planned for display on the homepage include "settings," referred to as the maintenance mode. This section can be utilized for adjusting settings, implementing system lockout, or addressing any needs that may arise during software development. A "sleep mode" will be incorporated, allowing users to manually set the device to sleep for power-saving purposes. Additionally, the screen will automatically enter sleep mode after 5 minutes of no input.

The liquid status displayed on the right side of the screen is designed to show either the actual measured amount or a percentage. This reading is directly derived from the HX711 Load Cell Amplifier, which can be programmed to return values measured by the connected sensor in the desired unit for display.

The decision to feature this liquid status on the homepage is for two main reasons: firstly, it keeps users informed about the remaining quantity of each liquid, and secondly, since the amounts are measured in real-time, a global variable flag is checked each time the homepage is initialized. This flag assesses if any liquid has less than 12 ounces available for another pour. If the liquid level in any reservoir falls below this threshold, the flag is raised, triggering the display of a warning message. The warning page includes a button called "run diagnostics" so that after the user refills the reservoir, they can manually prompt the system to check the load cell reading. If the reading is above the threshold level, the system will reset the global flag variable, stop displaying the message, and allow the user to return to the homepage.

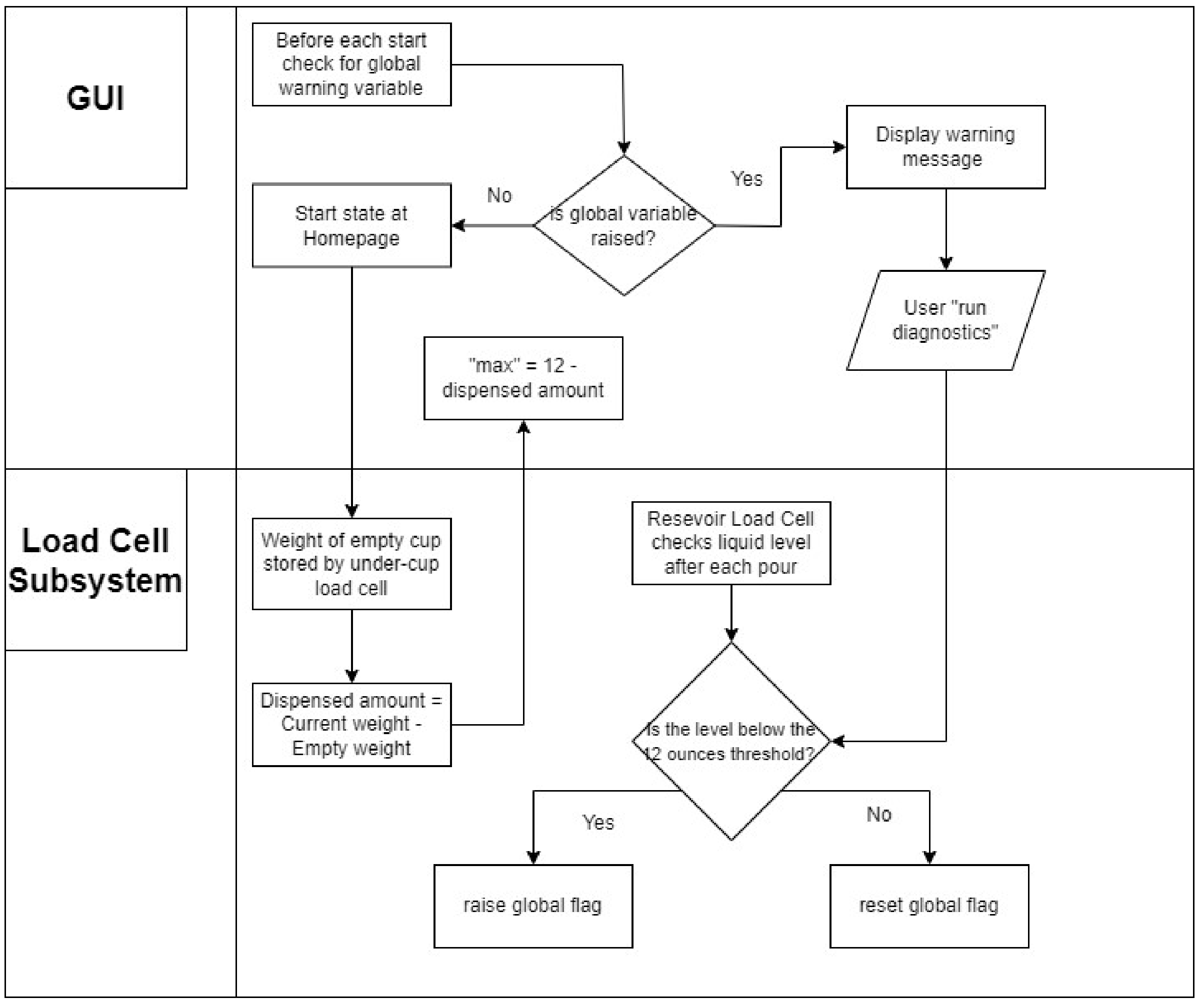


Figure 31: GUI and Load Cell Subsystem Integration

The way the user-interface will work is that when selecting the "pre-programmed" option it will set a variable like "selection_mode" to 1 and go to the next page of the hard coded options. Because of the size of the touchscreen, it limits the amount of option boxes we can display on it; therefore, to keep a good font size for the writing of the options and clear visual it is limited to 4 options only to be available on that page, but it will be further evaluated for the possibility to have 6 options available. In addition, the page will have a "return" option which by selecting it resets "selection_mode" to an idle state or to 0, and it goes back to the homepage screen. When any of the options from the pre-programmed list is selected since all the measurements are fixed and pre-defined on the code, the next page it will appear is the "wait" page and follow up with the "enjoy" page after the drink is done being made.

For the option when selecting "custom", it sets the variable "selection_mode" to 2 which is what will force the code to take the user to the next page with the option to select any of the 4 liquids available in the reservoirs, and for this page it also has the "return" option to go back to the homepage. After selecting any of the options available, the user will be transferred to a page which will be able to customize the amount it wants to pour of the selected liquid up to 12 ounces, with the option to either confirm the amount and drink selection or cancel all of it and return to liquid options page again. In this mode, when confirming the amount and drink selection it will display the "wait" page and follow up with a page asking if the user is done, so by selecting "no" it will take back to the options to add another liquid, and if "yes" them the "enjoy" message page will be displayed, but this option will only show up if any of the previous liquid selection does not add up to the maximum capacity of 12 ounces, if when selecting any of the drinks and reaching the maximum capacity after dispensing is done, then it will go directly to end of drink page. It is important to point out that the software will be designed so that the available liquid amount input by the user will be limited in real-time with what the cup has of availability left, with a variable "max" dictating how much it can be selected by the user.

More about the design of the "wait" page while the drinks are dispensed, the time the page will be displayed on the monitor is directly related to the amount of liquid being dispensed into the cup because the time the valves are opened determines the flow rate and cycles for the pumps to pour the ordered amount by the user; therefore, the time the "wait" screen stays on display will be exactly the same it takes for the system to pour the drink plus a delay of 1 second or less for safety and to guarantee that drink indeed is done.

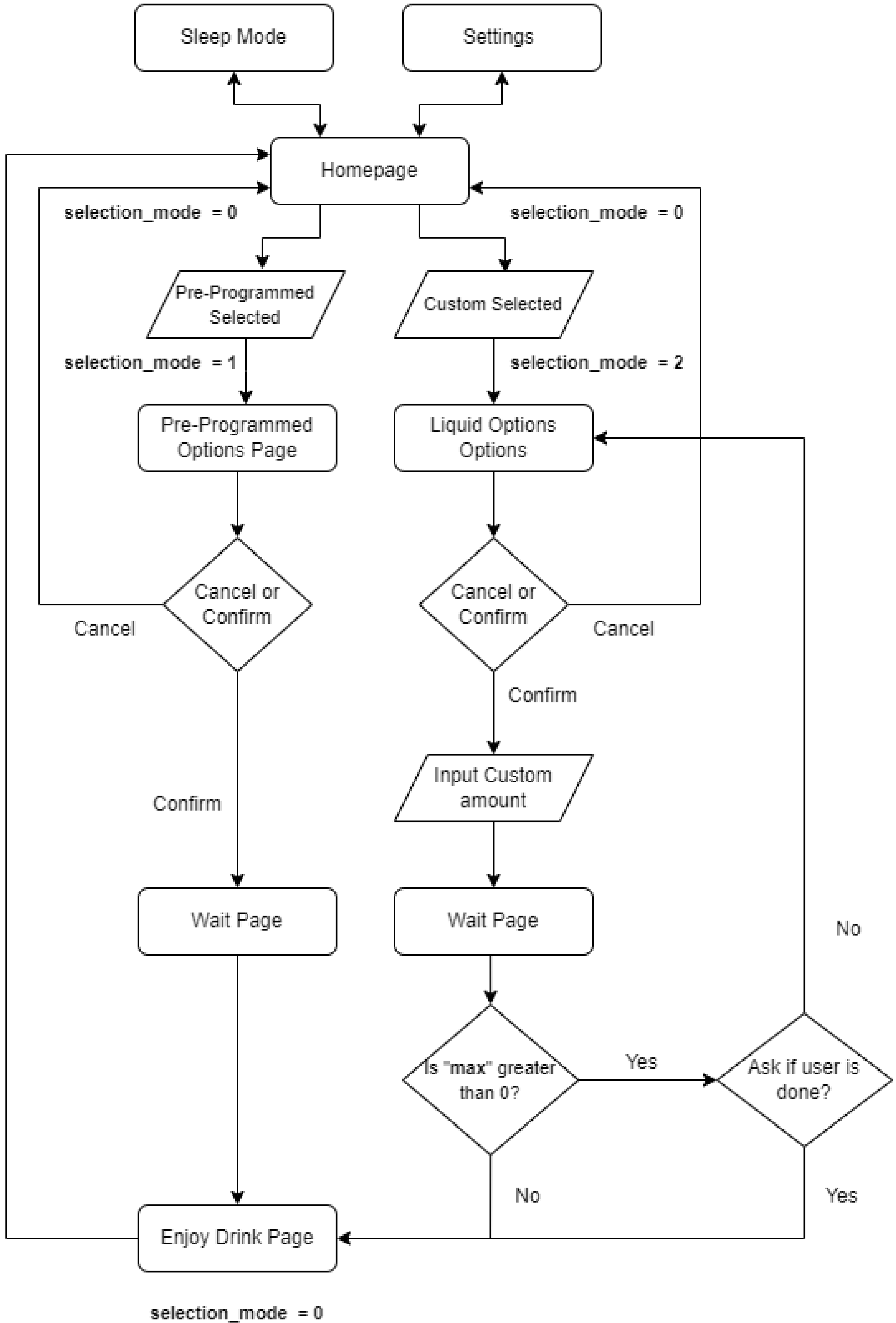


Figure 32: Touchscreen Flowchart

The touchscreen technology, as discussed in chapter three, may experience some inaccuracies due to its basic method of detecting touch input. This method relies on changes in resistance when two conductive plates come into contact in the area where touch pressure is applied, and the coordinates are then detected. To address this potential issue, a fine-tuning of the touchscreen will be implemented.

This fine-tuning involves defining minimum and maximum coordinate values so that only specific areas of the display, which will contain the selection boxes, will accept and read user touch input. For areas outside of these defined regions, no action will be triggered by touch input. The implementation of this feature aims to enhance the user interface experience by reducing the likelihood of the system reading a wrong input by the user accidentally clicking in a blank space near a selection box.

7.6 Maintenance Mode

The primary purpose of maintenance mode is to allow us as the designers to troubleshoot the software systems without having to either restart the software or develop different software. This mode will allow us to manually control all aspects of the Automated Bartender to troubleshoot any issues which may arise. We will have the ability to purge the pump line, activate or deactivate any valves, recalibrate the load cells, and recalibrate the touchscreen. These are all actions which should be performed upon boot-up, but it is unrealistic for these sensors and peripherals to remain accurate over an extended period of time. Considering that we anticipate the Automated Bartender to be active for extended periods of times, it is critical that we have the ability to recalibrate these peripherals.

Due to the highly manual nature of this mode, it is paramount that extra precaution be placed on the utilization of the peripherals in this mode. For example, if multiple valves are opened simultaneously, cross contamination occurs up into the reservoirs as the liquids diffuse through the common meeting point. This simply cannot happen; for commercial uses, this function would likely need to be password protected to ensure that only qualified personnel would be able to access these controls. However, for our applications, this is likely not necessary as the designers will be the only ones to make use of this mode. Therefore, we will likely not place password protections, but to avoid any possible errors in use, there will be a multitude of error messages asking the user to confirm actions such as opening multiple valves and purging the pump with no receptacle sensed at the output point. The functions of maintenance mode are summarized below. Note that as required other functions may be added.

Function	Purpose/Description	Security
Purge Pump	Empty the pump and common diaphragm of all liquids	Under-cup load cell read for cup; confirmation message
Open Valve	Functional testing of valves; allows for complete control of all valves simultaneously	For one valve, none; for more than one valve open at a time, confirmation message
Recalibrate Load Cells	Should a load cell not be properly zeroed over time	Instructions to remove all items from the cell; confirmation message
Touch Screen Calibration	Over time, it is possible that the resistive touch screen will need to be recalibrated	Confirmation message
Stretch: Unit conversion/ operating modes	Allow for the editing of units the machine uses from oz to mL; Allow for the change of operating modes from patron to bartender	Units will likely need a confirmation; change of mode would require a password.

Table 19: Maintenance Mode Functions

8.0 System Fabrication/Prototype Construction

8.1 PCB Layout

The PCB layout will consist of the main PCB, which will hold the majority of the components for the Automated Bartender including the three DC to DC converters, the MCU, the pin headers for the load cells, the pin header for the touchscreen, the USB to UART bridge, the valve control circuits, and the pump control circuit, and the Power Supply PCB, which will contain all the necessary circuitry for AC to DC conversion. This is done to prevent any possible interference between the switching components and electromagnetic elements of the Power Supply PCB. Trace width and properties will have to be taken into account for the design of the PCB due to larger power requirements of some of the components.

8.1.1 Main PCB Layout

The primary considerations taken with the main PCB layout were the proximity to the ESP32 unit and the logical layout of each system to prevent the need for long traces. Unlike the power supply PCB, there is no requirement for separate ground planes as there is no device which could cause interference (like the AC ground plane and DC ground planes interfering). While there are not two separate ground planes on the main PCB layout, there are still two ground planes, one on the top and one on the bottom, to allow for more surface area for return current to ground; this design choice will reduce the chances of ground loops from developing. All of the DC/DC converters will be placed away from any components that may be sensitive to noise. These converters will have large inductors that can possibly cause interference with other other components if they are placed too close in proximity.

Additionally, the board will make use of a two ounce copper pour for traces to account for a larger current, especially in the consideration of the current draw of the pumps and solenoids. While this is likely unnecessary for these elements still, for the sake of robustness we will still make use of these heavier pours to avoid any possible thermal or overload issues.

Another design requirement under consideration is the thermal distribution of the components. While none of the components in use are particularly thermally dissipative, there is the possibility for the processor to heat up. Should this occur and the Main PCB be placed in too close proximity to the Power Supply

PCB, this could create thermal issues. Thus, this should be avoided. Center ground pins on the ESP-32 MCU will not be soldered to the main board; these ground pins will serve as thermal relief for the MCU. Center ground pins of the MCU can be soldered to the board but is not recommended by the manufacturer.

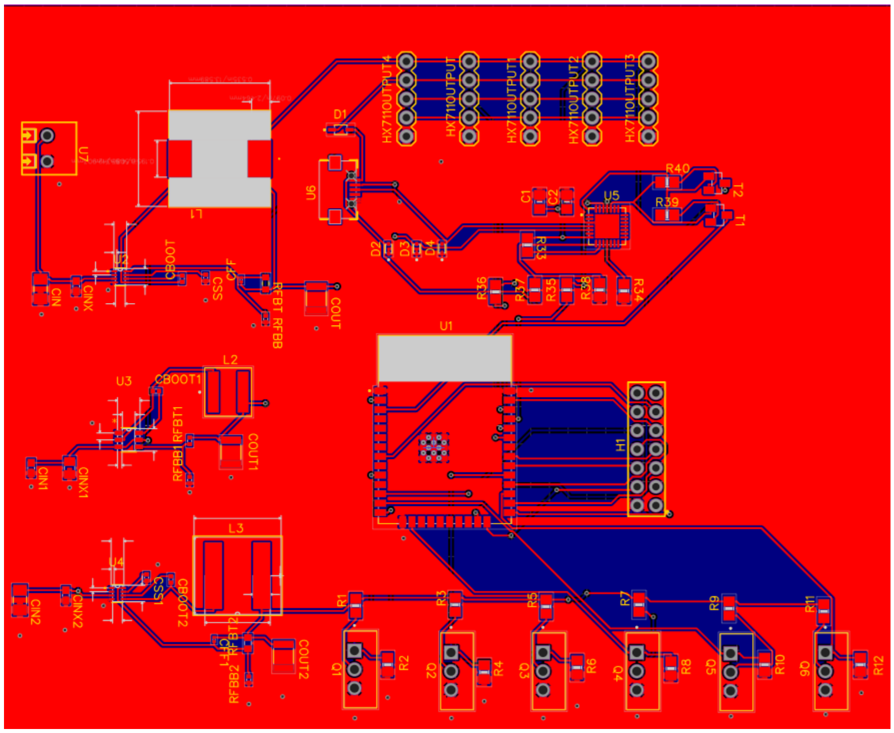


Figure 33: Main Board Layout

8.1.2 Power Supply PCB

The PCB layout of the switch mode power supply was created to minimize noise and increase signal integrity effectively. This layout includes critical components of the power supply, like the inductors, flyback transformer, and the various ICs needed for operation. The PCB is designed as a two-layer PCB with a copper pour on the top and bottom layer to act as two different ground planes. The first ground plane will serve as the ground for the primary (input) side of the transformer and the second ground plane will be the ground for the secondary (output) side of the transformer. The copper for the ground plane of the primary side of the transformer will be poured on the top of the PCB (red regions) and the ground plane for the secondary side of the transformer will be poured on the bottom side of the PCB (blue regions). Using a large surface area for the ground planes will reduce the probability of ground loops developing and causing harm and disruptions to the power supply. The width of the traces on the PCB have been adjusted to accommodate the larger current traveling through this PCB. A 2 oz. copper pour will be used for the traces to accommodate the larger current flowing through the board. A trace with a heavier copper pour will have an increased cross-sectional area allowing for more current to pass with the increased cross sectional area. Resistance of the trace will decrease because of the increased cross sectional area of the copper trace therefore less heat will be generated from the traces reducing the I^2R losses of the PCB.

By using a heavier copper pour compared to the standard 1 oz. pour for the traces it will allow for the width of the traces to be smaller while still being able to handle higher currents. The spacing between the traces and the poured copper on the surface of the PCB has been increased to prevent unintentional electrical connections on the board. The copper pour surrounding the traces will also act as a shield to reduce the risk of signal interference like cross talk from other traces and reduce the possibility of traces from possibly becoming coupled together.

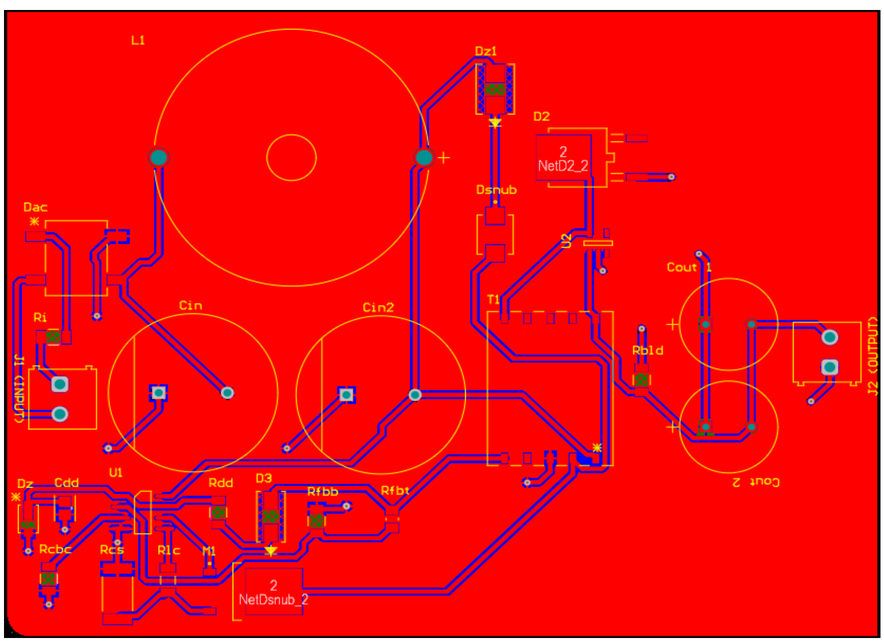


Figure 34: Power Supply PCB Layout

8.2 Structure Construction

The planned design of the Automated Bartender consists of the pumps, solenoids, and microcontroller being contained within a case with flat sides, an open back for accessing the components, and a flat front with an overhang for the output nozzle and holes for the LEDs to be mounted on. The exact dimensions will vary depending on the preference to ergonomics and tabletop footprint, but the minimum needed to fit all of the internal components is planned to sit at 18” x 18” x 18.”

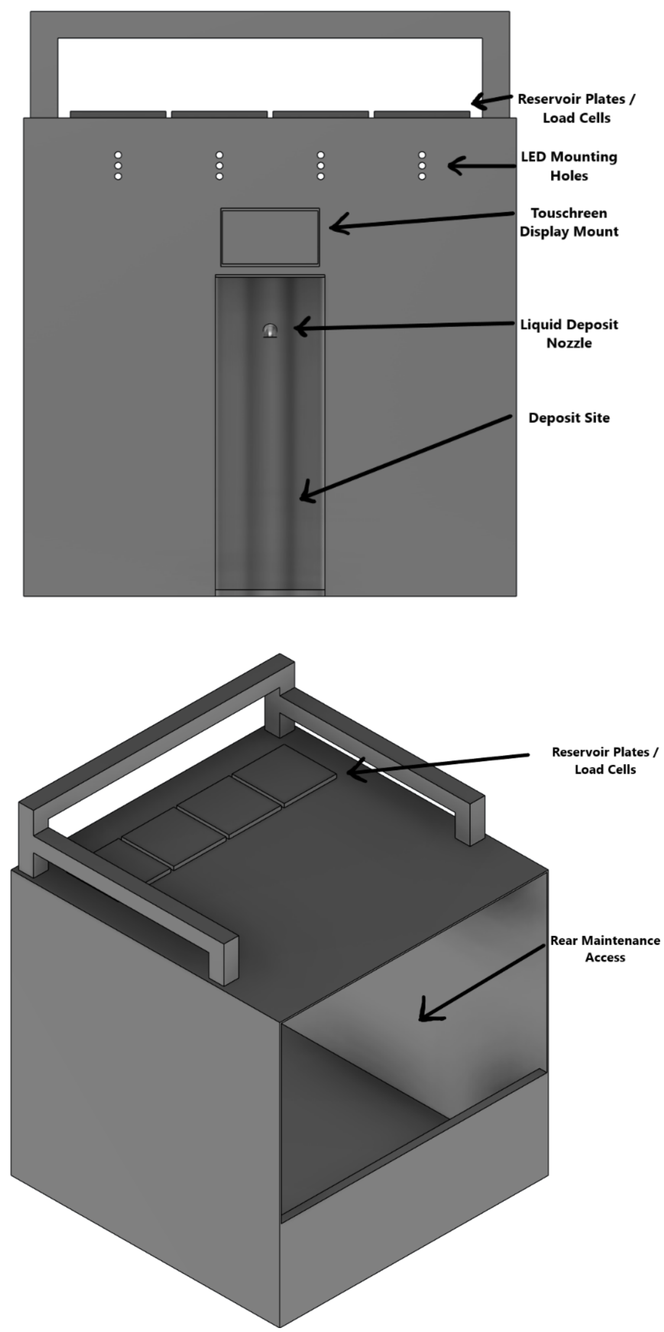


Figure 35: Structural Illustration Front (Top) and Isometric Back (Bottom)

These two views include all the notable features on the planned outside of the Automated Bartender. On the front view, from top to bottom are the reservoir plates, LED mounting holes, touchscreen display mount, liquid deposit nozzle, and deposit site. For the reservoir plates, this is where the reservoirs will be placed when being used by the Automated Bartender. Each raised plate has a load cell underneath to measure the fullness of each reservoir. The LED mounting holes are the locations for the green, yellow, and red LEDs for user feedback and visual enjoyment. The touchscreen display mount is where the touchscreen will be placed, with space in the back to wire to the microcontroller and full support from the frame to make the touchscreen sturdy, allowing users ease when selecting options. The liquid deposit nozzle is where the liquid(s) selected will dispense from, with the deposit site right below to allow the user to rest their cup in while the liquid is dispensing, preventing spillage.

On the back view, we have an isometric view to see the reservoir plates from a better angle on top of the Automated Bartender structure and showcase the rear access for maintenance and internal component fitting. The rear maintenance access is currently planned to be fully uncovered while in construction and testing phases, with a possible clear covering, such as plastic or glass, to come when assembly and testing is complete. With this covering, the internal components are protected from unauthorized access and elemental concerns such as dirt and other unclean items, and acts as shield for the exterior from accidental spills from inside the Automated Bartender.

9.0 System Testing

This chapter will contain all of the initial testing of the various hardware and software components of the Automated Bartender. This step is important to ensure that all of the major components that have been ordered work and they perform as expected. Performing this initial testing will help us to identify any possible deficits in the design of the project. This component testing will correlate with our testing constraint in section 4.2.3. Each component will be thoroughly tested and the steps taken during testing will be explained. It is imperative that during testing of these hardware and software components, the proper steps are taken to document the findings of each test. This documentation will serve as a reference throughout the design and development process of the Automated Bartender.

9.1 Hardware Testing

In this section, we will be testing the various major hardware components of the Automated Bartender. The components being tested will include the peristaltic pumps, load cells, power supply, touchscreen displays, and LEDs. This hardware testing section will be a critical phase in the development cycle of the Automated Bartender. During hardware testing, each component will be tested to ensure they operate as expected and to validate the quality of each component and to possibly identify any defects in the product.

9.1.1 Peristaltic Pump Testing

Testing of the peristaltic pump is a simple and straightforward process. The peristaltic pump is an essential part of the main design of the Automated Bartender. Testing this component involves applying a voltage to the input terminals of the pump and observing the effect. Once the output of the DC power supply is connected to the input terminals of the peristaltic pump, a 12VDC signal is initially applied at the input and the pump turns on. The input voltage is increased at 1V increments and all the way up to 17V as the input voltage is increased the speed of the pump increases. The current of the DC power supply was limited to 400 milliamps for the purpose of testing to ensure that the peristaltic pump does not become overloaded and burn out during testing. Adjusting the input voltage and observing the response of the pump will help the team later in the development process with calibrating the peristaltic pump to achieve the desired flow rate of the pump. Additionally, the self-priming capabilities of the peristaltic pump was tested by attaching silicone tubing to the nozzles of the pump and applying a voltage to ensure the pump is able to self-prime properly.

After the initial performance of the peristaltic pump was tested a PWM signal was applied to the control circuit of the pump and the input terminals were connected. Using a PWM signal to help to regulate the speed of the pump – by adjusting the duty cycle of the input signal – the speed of the pump can be calibrated to deliver a precise amount of liquid. As the duty cycle of the input signal is varied, the behavior and the flow rate of the pump is observed to ensure the pump is performing as expected.

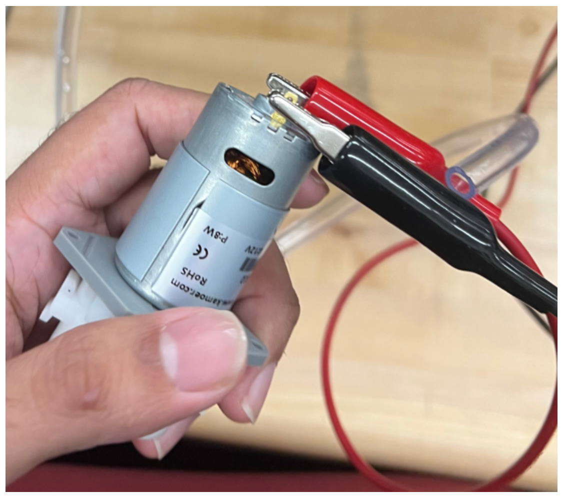


Figure 36: Peristaltic Pump Testing

9.1.2 Load Cell Testing

Testing the TAL220B load cells was done using the HX711 load cell amplifiers connected to the ESP32 microcontroller. Using the load cells with the amplifiers allows the microcontroller to detect and output values to the serial monitor on the IDE, allowing visual feedback to confirm connection and data travel from the load cell to the microcontroller. Much of the load cell usage for the Automated Bartender requires embedded programming to refine the data from the load cell to become usable for the LED system and to ensure accuracy for each pour.

The first step in properly testing the load cells is the construction of the apparatus meant to hold the load cell in place and balance the reservoir or other load. It is important for the load cell to receive all of the force of the load for accurate data. The apparatus planned is the load cell placed between two plates with spacers on opposite sides and ends of the cell to allow compression of the strain gauges. This will change the resistance values of the wheatstone bridge configuration of resistors and send different strain values through the load cell amplifier to the microcontroller.

Once the load cell is placed securely in its apparatus and the proper software is implemented for calibration, known weights will be placed onto the load cell to calculate the load cell's specific value outputs. This must be done for each individual load cell in order to calibrate the load cell based on each load cell's unique construction and to allow them to properly display the true value of the weight. Once the load cells are calibrated to this weight, other known weights will be measured to ensure concise data across all load cells. Additionally, low weights and weights near 5 kg. (11 lbs.) will be used to test the tolerances at the edges of the allowable weight limit, which appears to be a combined error of +/- 0.05% full-scale output (FSO).

In addition to the accuracy of instantaneous weight measurements, weights will be placed and left for 30-60 minutes at a time to test the creep of static weight and ensure it is within acceptable parameters. According to the datasheet, a creep no greater than +/- 0.1% FSO is expected per 3 minutes. With such a small value of percentage error, an elongated test is needed to properly measure the creep the load cell is experiencing.

With the testing of the load cells, value ranges for the LED feedback system can be obtained and implemented into the software. The LEDs rely directly on the value the microcontroller reads from the load cell and load cell amplifier, making the accuracy of these load cells vital to the user experience and throughput of the Automated Bartender. During testing, the values of each reservoir's LED thresholds will be assigned, considering that each reservoir will have a unique shape and weight.

9.1.3 Power Supply Testing

Testing the power supply of the Automated Bartender will be a straightforward and simple process. Testing of the power supply will involve measuring the output voltage of the power supply and observing how the power supply will behave when a load is attached. Before the power supply is turned on, there is a visual inspection of the power supply to ensure that there are no missing or damaged components. If the power supply was turned on and there were missing or damaged components, this would cause the power supply to fail and/or create a safety hazard for the team.

Once the power supply is turned on and allowed to reach a steady state, the output voltage of the power supply will be measured and compared to the expected value of 18V. If the expected values are not obtained, the power supply will be turned off, and further troubleshooting steps will take place. Once the output voltage of the power supply is at the proper level, various loads will be applied to the power supply to measure the efficiency. One of the loads applied to the power supply will be a light bulb once the light bulb is connected, a visual inspection of the power will be performed to ensure there are no components that are damaged, and the temperature of the power supply will be monitored to ensure it is not overheating.

The transient response of the power supply will be observed by applying sudden changes to the load and observing the response time and stability of the power supply. All of the initial testing of the power supply will be done with the supervision and guidance of a professional since this subsystem involves working closely with wall power.

9.1.4 Valve Testing

Testing the function of the valves was a fairly simple process. Before engaging in any complex testing, the function of the solenoid valve was tested by supplying its rated threshold voltage across the terminals in order to test its proper function. Additionally, the voltage was decreased to determine the minimum possible toggling voltage for the valve. While the rated voltage will be used throughout the applications of this project, it could be useful to have accurate information for this should any issues occur where the valve does not trigger when it should.

First, the MOSFET controller circuit was constructed on a breadboard making use of through-hole components. The solenoid valve was wired in series with a constantly-on 12V signal generated by DC PSU and the drain of the MOSFET. The source was grounded, and the gate attached to a pin on the ESP32. The ESP32 was programmed to output a logic high signal which alternates over a known time period. This circuit, in theory, provides enough voltage across the gate to open the MOSFET, allowing the circuit to complete, supplying the 12V to the solenoid valve, and toggling the valve every timed interval.

The criteria for success for this test was first, the proper response and function of the valve. That is that the valve opens and closes according to the signal provided by the ESP32 and that the timing of the toggling of the valve is approximately accurate. Additionally, the valve should open quickly. This was determined approximately by timing the pauses between the toggling of the valve, which is made evident by the sound of the opening and closing. When tested the reaction timing of the valve was both approximately accurate and consistent. While this method of testing is not surefire as it introduces the possibility of human error in timing the pause between toggles, the primary purpose of this test is not to determine the accuracy of the timing, but the accurate response of the circuit and subsequently the valve as well as the proper response to continued activation and deactivation of the valve via a toggling logic signal. Therefore, any slight timing delay in response to the toggling signal is likely not a critical issue for this system.

The final test which was applied to the valve system was a toggling speed test to determine how quickly the valve can be opened and closed. To achieve this, the same circuit used for prior testing was built, but the ESP32 will be coded to supply a short signal to open the valve. This signal was then shortened until a scenario was created where the pulse became too short to open the solenoid valve. The purpose of this test was to determine an approximate minimum open and close time of the valve to aid in further software design. Success in this test was a sufficiently fast open and close time of less than 150 ms.

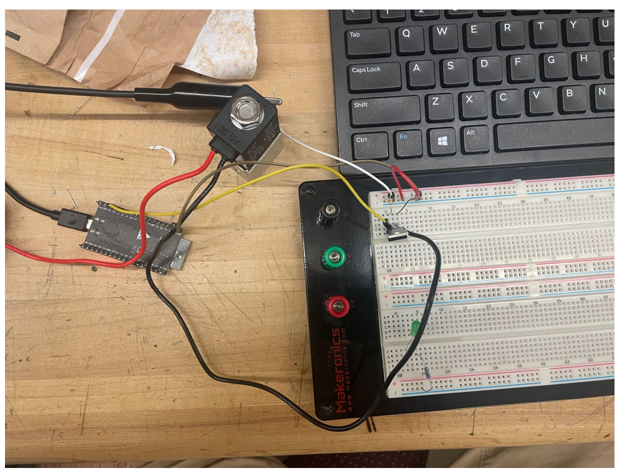


Figure 37: Solenoid Valve Testing Circuit

9.1.5 Touchscreen Testing

To test the main functionalities of the touchscreen display, the decision was made to use Arduino IDE example codes, deviating from the chosen development framework for programming the GUI, Visual Studio Code with the ESP-IDF extension. This choice was motivated by the existence of a widely approved community library called "TFT_eSPI," developed by Bodmer on GitHub. This library offers several advantages for testing the ILI9488 TFT LCD display using an ESP32 development board.

The TFT_eSPI library includes a variety of example codes. Two main tests were performed. The first involved a simple display test to check for display initialization, calibration, and various graphics, including text, lines, circles, etc. It also examined color range and backlight control. The second test focused on touch calibration, involving touching arrows on the corners of the monitor, followed by a simple "on" and "off" button page to assess the touch ability of the display.

However, before conducting these tests, significant configuration changes were necessary in the library. TFT_eSPI is a general-purpose library adaptable to various microcontrollers such as RP2040, STM32, and ESP8266, as well as compatible with many IC driver modules on TFT SPI screens (e.g., ILI9341, HX8357D, ILI9486). Adjustments in the "User_Setup.h'' file were made, commenting out configurations defined for ILI9341 and uncommenting configurations for the ILI9488 module. Pin configuration had to be aligned to meet with those wired between display SPI pins and the ESP-32 GPIOs, which required jump-wires and male-to-female wires for GPIOs that needed to be connected to multiple pins on the touchscreen display, and frequencies for the SPI interface, read, and touch were adjusted.

Following these adjustments, the "User_Setup_Select.h" file was configured to activate the specific file called "Setup21_ILI9488.h," the setup file for ESP32 and ILI9488 SPI bus TFT implementation. To run the code example on the ESP-32, the "esp32" board developed by Espressif needed to be downloaded to the Arduino IDE. Selecting the board and port, the code examples for testing were executed by navigating to "File," "Examples," "TFT_eSPI," and then uploading the desired option. For the display test, "480 x 320" was chosen, running "TFT_graphictest_one_lib," and for the touch test, "Generic" and "On_Off_Button" were selected. Holding the bootup button on the ESP-32 until the Arduino IDE connected to the board allowed the successful execution of both tests. The display exhibited vibrant colors, smooth transitions, and clear graphics, while the touch functionality worked well, enabling easy selection between the "on" and "off" buttons with visible changes.

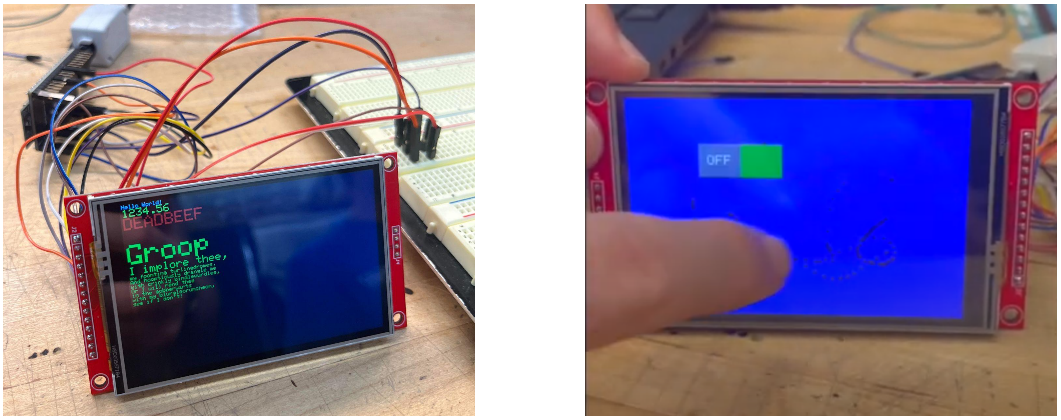


Figure 38: Display Test (left) and Touch Test (right)

9.2 Software Testing

Software testing will follow the criteria laid out in section 4.1.1 regarding software testing standards. Initially, each subsystem's hardware will be tested to ensure its compatibility with the software being utilized. The primary function of these tests (laid out in section 9.1) is to determine the proper function of the hardware, but also their proper response to the basic functions which the software will need to carry out. For example, the valve subsystem tests in section 9.1.4 serve to test the proper function and design of the hardware created for the control of the solenoid valves, but the added element of utilizing the ESP32 to control the function of these valves simultaneously tests the controller's ability to interface the hardware properly with the software. These initial tests allow for the establishment of the software's primary control of the system. The table below showcases the various primary functions to be tested for each system.

Subsystem	Test	Success Criteria
Peristaltic Pump	On-off control	The pumps properly toggle on and off when dictated by software
Peristaltic Pump	PWM control	The pumps properly ramp up and down when dictated by software
Load Cells	Load Cell Read Test	The weight measured by the load cell is properly read, amplified, and converted when requested

Load Cells	Multiple Load Cell Read	Load cells can be selected and read. The proper cell is read
Solenoid Valves	Open-close Control	The valves properly toggle when dictated by software
Solenoid Valves	Open-close Speed	The open-close speed is accurately measured by the software
Touchscreen	Picture Display	The proper picture is displayed and changed when dictated by the software
Touchscreen	Touch Response	Touch is accurately recorded and relays information to the software

Table 20: Primary Software Tests

Once each primary software element is deemed to be properly functioning, more goal-adjacent hardware will be tested, meaning the integration of these primary functions will be compiled to create functions which will work directly to achieve the goals set out for this project. For example, testing the touch screen HMI to ensure the machine moves seamlessly through states will help achieve goals regarding ease-of-use and proper function. It is important to note that as laid out in section 4.1.1, there are two possible error types for these systems. The first of which being integration errors – errors which result due to the compatibility of two primary functions and their simultaneous operability – and the second being primary-secondary complexity errors – errors which surface with the primary function being utilized in a secondary function which are a result of the configuration of the primary function itself instead of the compatibility of two primary functions cooperating.

This stage of software development is where the majority of the work lies, and, therefore, will be where the majority of issues will likely take place. In the case of compatibility errors, the secondary function will need to be recursively edited and tested to help the features cooperate. In the event of a primary-secondary complexity error, the primary function will need to be revisited and edited to solve the issue before moving back to secondary functionality.

The final stage of software testing will be the finished product, which will involve verifying that we meet our goals and engineering criteria laid out in Chapter 2. In this stage, integrative functions may be recursively reevaluated to ensure that the final goals of the project are fully met. Below is a summary of the integrative (secondary) software tests. No summary of the final stage of testing is provided as the tests will simply be the fulfillment of the design specifications and project objectives.

Integrative Test	**Subsystems Involved**	**Success Criteria**
HMI Function	Touchscreen	HMI functions properly and updates corresponding state variables
Dispensing Function	Valves, Pumps	Dispenser functions properly within speed requirements; Lockouts prevent dispensing
Reservoir Monitors	Load Cells, LEDs	Reservoir levels are accurately tracked and displayed
Cup Fill Measurement	Load Cells, Pumps	Amount of dispensed liquid is accurately tracked to 0.1oz
Maintenance	All	Maintenance mode allows control and calibration

Table 21: Integrative (Secondary) Software Tests

9.3 Plan For Senior Design 2

Being able to plan for senior design two ahead of time will be an imperative task since senior design two will be filled with many challenges that we as a group will have to overcome together. We will have to assemble the various PCBs and test them to ensure they are working as expected and finalize the software that the Automated Bartender will operate on. All of the hardware will have to be seamlessly integrated with the software to create a functioning prototype. As we are working on assembling the final prototype we will have to prepare for the final presentations to the review committee in parallel.

9.3.1 PCB Assembly and Testing

PCB assembly will be a major part of our workload in senior design two. There are currently two different PCBs that will have to be created and tested during senior design two. The first PCB will be the main PCB that all of the major systems will be housed like the voltage regulators and the microcontroller unit and the second PCB is where the power supply will be housed. The final design and verification of each PCB will be completed towards the end of SD1 and the PCBs will be ordered over christmas break. The initial assembly of the PCBs will start over christmas break to give the group a head start when going into senior design two. Initial testing of the main PCB will start over Christmas break and the initial testing of the power supply will have to be done at the very start of senior design two with the supervision of a professional for safety reasons since this PCB will be using wall power. Extra components will be ordered for each PCB since we expect some parts to run the risk of being damaged during the testing process. Most of the components on PCBs will be soldered on by the use of solder paste and stencil and any through hole components will be soldered using solder wire and an iron. The initial construction and testing of the PCBs will play a vital role to the timeline in senior design two since these will be the most important aspects of the development of the Automated Bartender. If these PCBs do not work a lot of time will have to be spent troubleshooting, and we will run the risk of possibly having to redesign the PCB. All of this time spent troubleshooting will subtract from our overall time to assemble the Automated Bartender.

9.3.2 Software Modifications and Testing

The primary goal of senior design two from a software standpoint is developing a fully functioning prototype software. Because the majority of primary functions will be established in the testing cycle of senior design one, senior design two will likely be devoted mostly to the integrative function of the software to create a functioning product from the cooperation of the individual systems and eventually creating a fully functioning product. In the development of these integrative functions for the Automated Bartender, we anticipate needs for improvements in robustness of configuration, timing of various elements, and even possibly in the general flow of the software as a whole. For example, taking the pump and valve subsystem. While they each currently have been tested to ensure their proper functionality and controllability by the ESP 32, there may be subtleties in the configuration or operation of the pumps or valves individually which may require the configuration to be adjusted such as the PWM timing needing to be increased to achieve an accurate pour or extra delay being needed between the opening or closing of valves and the activation of a pump. While this is just one possible error which could arise in this next phase of development, we anticipate many more to arise as the software grows more complex and integrates all the subsystems.

Any additional software development time will be put into the development of sprites for the touchscreen or possible stretch goals which could be integrated into the software.The first draft of a complete software system will be developed as a “bare bones” HMI which will help ensure that there are no functional errors with the software. That is, the pump, valves, load cells, LEDs, and touch screen should be functioning in a complete rough draft prototype software before any intricacies with user development or optimizations are taken into consideration. After the software is optimized for timing and functionality, the HMI will be built up in robustness, attractiveness, and ease-of-use to help meet the goals prescribed in chapter two. We will also be prepared to make any alterations to the software diagrams presented in chapter seven to achieve the goals and specifications we have laid out for this project.

9.3.3 Assembly of the Exterior

The assembly of the exterior of the Automated Bartender is where the vision of the Automated Bartender will start to become a reality. The design of the exterior of the project will be finalized at the end of senior design one and the initial assembly and realization of the project will start within the first month of senior design two. We will have a 3D sketch that we will follow for the assembly, and we will constantly be updating with design changes as development continues. We will create a small physical model of the exterior during the first month of senior design two and this is where we will be able to visualize the Automated Bartender in real space. The exterior will be created using laser cut sheets of acrylic, and if the use of acrylic does not fit into our budget, we will make use of balsa wood that has been sealed to be water-resistant. All of the smaller miscellaneous components will be created via 3D printing (PLA) if they cannot be easily purchased, like the mount for the PCB.

10.0 Administrative Content

10.1 Initial Budget Allocation

Allocation Description	Estimated Budget	Items
Development	$30	MCU Devkit, jumpers, breadboarding materials
Pump System	$50	Peristaltic Pump, food safe tubing, barbed fittings
Valve System	$40	Solenoid Valves
Load Cell System	$120	Load Cells, HX711 Amplifiers
Touchscreen System	$30	Touchscreen
Power Systems	$20	Converters, Regulators
Construction	$60	Building materials, PCB printing
TOTAL	$350	

Table 22: Budget Estimates

10.2 Bill of Materials

Quantity	#/Package	Description	Unit Price	Total Price
1	1	Kamoer KPHM100 peristaltic pump	$18	$18
5	1	2-Way NC Solenoid Valve	$11	$55
1	1	Tygon E-65-F Food & Beverage Dispensing Tubing	$10	$10
1	1	ESP32-WROOM-32E	$3	$3
2	5	⅛'' to ⅛'' Threaded to barbed fittings	$7	$14

5	1	SEN-14729 5kg Load Cells	$13.12	$65.60
5	1	HX711 Load Cell Amplifiers	$10.70	$53.50
1	1	3.5 Inches SPI TFT LCD ILI9488 Touch Screen	$16.99	$16.99
6	1	Acrylic Sheets 2'x1'	$10	$60
2	1	ESP32-WROOM-32E Dev Module	$10	$20
1	N/A	Various Electronic Components	$30	$30
1	1	PLA Filament	$10	$10
4	1	Food Grade Reservoir	$4	$16
1	1	Main PCB	$30	$30
1	1	Power Supply PCB	$20	$20
2	1	TSP 62932 DRDL Converter	$0.19	$0.38
1	1	LMR 51420 DDCR Converter	$1.33	$1.33
1	1	BAT760-7 Schottky Barrier Diode	$0.53	$0.53
3	1	LESD5D5.0CT1G Bidirectional	$0.87	$2.61
1	1	CP2102N-A01-GQFN28 USB to UART	$4.66	$4.66
4	1	Red LEDs	$0.34	$1.36
1	1	Various Construction/Assembly Components	$15	$15
TOTAL				$447.96

Table 23: Bill of Materials

10.3 Project Milestones

Task Description	Start	End
Initial senior design start	August 22nd, 2023	August 27th, 2023
Project Selection	August 28th, 2023	September 7th, 2023
Initial draft of Divide and Conquer Document. Attend the Divide and Conquer meeting with Dr.Chan	September 8th, 2023	September 21th, 2023
Revise Divide and Conquer Document	September 22th, 2023	September 24th, 2023
Begin working on 60 Page draft	September 25th, 2023	October 30th, 2023
Review and submission of 60-page document	October 31st, 2023	November 3rd, 2023
Start 120-page final document + start sourcing components and PCB design	November 4th, 2023	December 1st, 2023
Review and submission of final 120-page document	December 2nd, 2023	December 5th, 2023

Table 24: Senior Design 1 Milestones

Task Description	Start	End
Initial PCB Assembly + Testing	December 15th, 2023	January 25th, 2024
PCB redesign + Testing	January 20th, 2024	February 15th, 2024
Begin Final prototype assembly and functional	February 15th, 2024	March 1st, 2024

checkout		
Final Prototype testing and functional checkout	March 1st, 2024	April 25th, 2024
Demo Video and Website Updates	March 15th, 2024	April 15th, 2024
Final Presentation	TBD	TBD

Table 25: Senior Design 2 Milestones

10.4 Work distribution

The work for this project has been and will be equally divided amongst the group members. While typically a group will contain at least one computer engineering student, we have four electrical engineering students, and, therefore, will need to be diligent in assigning and separating work as there is no clear logical barrier as to who should be responsible for software. We have designated the software work primarily to Pedro and Tommaso while the hardware will be the primary responsibility of Luis and Somal.

However, it is important to note that there will be a large overlap in responsibilities as we counteract our lack of computer engineering expertise with manpower and research. Additionally, in general, each group member will be responsible for being experts in a chosen subsystem of the design: Somal is focusing on the power supply and pumps, Tommaso is focusing on the valves and microcontroller, Pedro is focusing on the touchscreen and SPI connections with the microcontroller, and Luis is focusing on the load cells and I^2C connections with the microcontroller. For each of these, the group member responsible will take lead in explaining, integrating, and interfacing the subsystem properly into the Automated Bartender. Below shows the distribution from the research and testing above.

Chapter 1:

- Somal wrote the executive summary

Chapter 2:

- Luis wrote the project motivations and goals
- Luis wrote the objectives
- Tommaso wrote the specifications

- Pedro wrote the hardware and software diagrams
- Tommaso wrote the house of quality
- Luis wrote the existing technology

Chapter 3:
- Tommaso wrote the microcontroller selection
- Somal wrote the pump selection and tubing
- Tommaso wrote the valve selection
- Somal wrote the power components
- Pedro wrote the touchscreen display
- Luis wrote the weight sensors
- Luis wrote the LEDs
- Luis wrote the load cell amplifier
- Pedro wrote the programming language selection
- Pedro wrote the software development framework

Chapter 4:
- Tommaso wrote on testing standards and technical standards
- Somal wrote on food safety standards
- Both Tommaso and Somal wrote on various constraints

Chapter 5:
- Pedro wrote on the various types of AI and their usefulness
- Luis wrote on the applications of AI
- Pedro, Luis, and Tommaso wrote on the usefulness of AI for this project

Chapter 6:
- Somal created subsystem block diagram
- Somal created 3.3V,5V,12V DC/DC converter schematics
- Somal created AC/DC converter schematics
- Pedro created and wrote on the touchscreen schematic
- Tommaso wrote on the USB to UART bridge
- Luis referenced load cell amplifier schematic from Sparkfun
- Luis created load cell schematic
- Luis created LED schematic
- Luis created Structural Illustration of the Automated Bartender

Chapter 7:
- Tommaso wrote on the pump and valve software design
- Tommaso wrote on the load cell and LED software design
- Pedro wrote on the touchscreen software design

Chapter 8:

- Tommaso wrote on the main PCB layout
- Somal wrote on the power supply PCB layout

Chapter 9:

- Somal wrote on pump testing
- Luis wrote on load cell testing
- Somal wrote on power supply testing
- Tommaso wrote on valve testing
- Pedro wrote on touchscreen testing
- Tommaso wrote on the software testing plan for senior design one
- Somal wrote on the hardware and physical design plans for senior design two
- Tommaso wrote on the software plans for senior design two

Chapter 10:

- Tommaso wrote the initial budget allocation
- Tommaso wrote the Bill of Materials
- Somal wrote the project milestones
- All members wrote the work distribution

Chapter 11:

- Tommaso wrote the conclusion

11.0 Conclusion

In summary, our goal for this project is to demonstrate our engineering knowledge through the design, development, and testing of an Automated Bartender system. For which, the ESP32-WROOM-32e will serve as the primary processor for controlling and interacting with the necessary peripheral systems. Through the use of a user-friendly HMI, the user will trigger the ESP32 to be able to control the valves and pumps to dispense to various reservoirs, the load cells for safety and status monitoring, the LEDs for indication, and the touchscreen for feedback and user input. Upon initial testing, these individual systems can function accurately and quickly under the control of the ESP32. However, the challenge now is to integrate these controls into a proper sequence of function and priority to achieve the goals prescribed in chapter two. By carefully considering the priority and timing of individual tasks, our hope for continued development is to interface these systems into a functioning prototype and to iteratively implement optimization techniques which will help us achieve the design specifications we have set. Additionally, time and consideration will be put into stretch goals, their overall ability to aid in the overall objectives of the project, and their feasibility in implementation given the time constraints.

Throughout the course of research for this project, we have gained a vast amount of knowledge surrounding the real-world applications of the general knowledge from classes. First, the depth of research and development required for this project, which is relatively large in scale for students but not large-scale when compared to industry, was overwhelmingly more than originally anticipated. We needed to account for as many possible factors surrounding every element of the design to find the technology and parts necessary for the construction of this project. What's more, we have to consider the possibility that we still could have incorrectly accounted for factors surrounding various components and technologies selected to achieve our goals. This iterative learning process proved as a new experience which we believe will be highly beneficial not just going forward with the second half of this project but also as we enter the industry. Second, the breadth of knowledge we gained in the real applications of circuitry we have studied was above our expectations. For example, we had studied simple control circuits like those implemented for the pumps and valves, but now we have an understanding of a possible real application of this technology. Finally, we learned a vast amount about PCB development. While we had engaged in PCB design previously, we had never developed a board which would actually need to be properly functioning. Therefore, we had to gain a large amount of knowledge surrounding PCB layout, construction, traces, vias, and ground planes. Additionally, our horizons were broadened beyond previous considerations into special features which need to be implemented in PCBs containing both AC and DC elements.

Over the past semester, we have recognized the challenges associated with a project of this scope, primarily the balance of communication and accountability, both individually and as a group, which we have learned to manage professionally. We found that communication was a principal element for our success and how we as a group can account for one anothers mistakes and shortcomings, which we all have. We learned how a team dynamic is critical to success and how to function as a true team to engage in the creative design process together. Also, we have recognized that there is a depth of complexity to a system which initially seemed simple. Moving forward, we desire to continue to learn to work together more efficiently to achieve the goals we have set for ourselves and to continue to dig deeper into the understanding of every component of the Automated Bartender to better understand how we can leverage our knowledge of engineering to create a functioning proof-of-concept for this idea. We have set a healthy precedent of routine meeting, questioning, prototyping, and improving, which will undoubtedly carry over to the next semester as we strive to complete this project.

Our primary objective moving forward is to utilize this document as a comprehensive guide to the design, construction, troubleshooting, time management, and testing of the Automated Bartender, understanding that development has only just begun. This base of knowledge we have accumulated and applied should serve as the primary source for the Automated Bartender moving forward. Our hope is that there will be no further design considerations necessary beyond those laid out in this document; however, we recognize that there may need to be changes to the design laid out, and that this is a part of the iterative creative design process in which we are engaging. We are prepared to make any necessary changes to the implementation of the Automated Bartender system to achieve the goals laid out in this document as the ultimate goal of this project is a functional and complete proof of concept created by our ultimate deadline of the end of next semester.

Appendix A: Works Cited

[1] *ESP*. Available at: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/lcd.html .

[2] *ESP32 TFT display library - ESP32 forum*. Available at: https://www.esp32.com/viewtopic.php?t=1937 .

[3] *Ili9341 DS v1.13 20110805 - adafruit industries*. Available at: https://cdn-shop.adafruit.com/datasheets/ILI9341.pdf .

[4] *3.5 inch SPI module ILI9488 Dropbox*. Available at: https://www.dropbox.com/sh/ucxjkx90pnn98f8/AAAXOFX68ouy7Yrz6J1-0Woxa?dl=0 .

[5] Damirchi, W. by M. (2023) *Interfacing 2.8 inch TFT LCD touch screen with ESP32*, *Electropeak*. Available at: https://electropeak.com/learn/interfacing-2-8-inch-tft-lcd-touch-screen-with-esp32/ .

[6] *How to create a simple touchscreen gui || Arduino LCD & touchscreen tutorial* (2018) *YouTube*. Available at: https://www.youtube.com/watch?v=SIo_Gv7K7Fo.

[7] Kuma (2022) *What are the different types of touchscreens and how are they used?*, *Noritake Co., Inc. Electronics Division*. Available at: https://www.noritake-elec.com/featured-articles/different-types-touchscreens-used#:~:text=There%20are%20two%20types%20of,applied%20to%20the%20conductive%20film.

[8] *LVGL/lvgl_esp32_drivers: Drivers for ESP32 to be used with LVGL*, *GitHub*. Available at: https://github.com/lvgl/lvgl_esp32_drivers.

[9] Sjennings (2022) *Top 5 touchscreen interfaces*, *Foxguard*. Available at: https://foxguardsolutions.com/top-5-touch-screen-interfaces/#:~:text=But%20did%20you%20know%20there,(SAW)%2C%20and%20infrared.

[10] *Touch screen basics* (2023) *TRU*. Available at: https://tru-vumonitors.com/touch-screen-basics/.

[11] ByteSnap (2023) *FreeRTOS vs linux for embedded systems*, *ByteSnap*. Available at: https://www.bytesnap.com/news-blog/freertos-vs-linux-embedded-systems/.

[12] *Introduction to C*. Available at: https://www.w3schools.com/c/c_intro.php.

[13] *IDF getting started: Espressif Systems ESP*. Available at: https://idf.espressif.com/#:~:text=ESP%2DIDF%20is%20Espressif’s%20official,ESP32%2DC%20series%20of%20SoCs.

[14] The JFrog Security Research Team August 15 and Team, T.Jf.S.R. (2022) *Choosing between linux and rtos for IOT projects - jfrog connect*, *JFrog*. Available at: https://jfrog.com/connect/post/how-to-choose-between-linux-and-rtos-for-iot-project/#:~:text=Does%20your%20application%20require%20more,%E2%80%9D%20%E2%80%93%20take%20this%20into%20consideration.

[15] *Kernel in operating system* (2023) *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/kernel-in-operating-system/.

[16] Learning, G. (2022) *Difference between C and C++*, *Great Learning Blog: Free Resources what Matters to shape your Career!* Available at: https://www.mygreatlearning.com/blog/difference-between-c-and-c/#:~:text=from%20C%20language%3F-,C%20is%20a%20structural%20or%20procedural%20programming%20language%20that%20was,%2C%20Inheritance%2C%20Polymorphism%2C%20etc.

[17] *IDF (IOT development framework) vs Arduino core in 2023*, *ESPBoards*. Available at: https://www.espboards.dev/blog/esp-idf-vs-arduino-core/.

[18] *ESP-IDF*, *ESP IoT Development Framework | Espressif Systems*. Available at: https://www.espressif.com/en/products/sdks/esp-idf#:~:text=ESP%2DIDF%20is%20Espressif's%20official,ESP32%2DH%20series%20of%20SoCs.

[19] *Step-by-step guide: How to program ESP32 with Arduino Ide?* (no date) *m5stack*. Available at:https://shop.m5stack.com/blogs/news/step-by-step-guide-how-to-program-esp32-with-arduino-ide.

[20] *Arduino Integrated Development Environment (IDE) V1*, *Arduino Documentation*. Available at: https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics.

[21] *Isaac Computer Science*. Available at: https://isaaccomputerscience.org/concepts/sys_proglang_high_level?examBoard=all&stage=all.

[22] *10 Best CHATGPT Alternatives in 2023 (free and paid)* (2023) *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/best-chatgpt-alternatives/.

[23] Agnihotri, N. *What is MicroPython?*, *Engineers Garage*. Available at: https://www.engineersgarage.com/what-is-micropython/.

[24] *Difference between python and C++* (2023) *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/difference-between-python-and-c/.

[25] *How to explain object-oriented programming to kids* (2020) *FunTech Blog*. Available at: https://funtech.co.uk/latest/explain-object-oriented-programming-to-kids#:~:text=The%20simplest%20way%20to%20explain,size%20and%20year%20as%20attributes.

[26] *Introduction to python* (2023) *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/introduction-to-python/?ref=roadmap.

[27] *MicroPython Python for microcontrollers*. Available at: https://micropython.org/.

[28] *Python 3 basics* (2023) *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/python-3-basics/.

[29] *Python interpreter Computer Science*. Available at: https://cs.stanford.edu/people/nick/py/python-interpreter.html#:~:text=You%20write%20your%20Python%20code,is%20called%20an%20%22interpreter%22.

[30] *These AI tools could help boost your academic research euronews*. Available at: https://www.euronews.com/next/2023/08/07/best-ai-tools-academic-research-chatgpt-consensus-chatpdf-elicit-research-rabbit-scite.

[31] *Sparkfun load cell amplifier - HX711*, *SEN-13879 - SparkFun Electronics*. Available at: https://www.sparkfun.com/products/13879.

[32] *Sen-14729 Sparkfun Electronics | Sensors, transducers | DigiKey*. Available at: https://www.digikey.com/en/products/detail/sparkfun-electronics/SEN-14729/9555603.

[33] *0 PSI to 15 PSI, 4 to 20ma DC, pressure Transmitter Grainger*. Available at: https://www.grainger.com/product/DWYER-Pressure-Transmitter-0-psi-2HLR2.

[34] I am an embedded c software engineer and a corporate trainer (2022) *I2C protocol,bus and Interface: A brief introduction*, *Aticleworld*. Available at: https://aticleworld.com/i2c-bus-protocol-and-interface/.

[35] *Hydraulic system of pressure sensor PressureSensorsnet*. Available at: https://www.pressuresensors.net/hydraulic-system-of-pressure-sensor#:~:text=Examples%20of%20hydraulic%20applications%20include%20vehicle%20braking%20systems%2C,lifts%20and%20industrial%20machinery%20such%20as%20hydraulic%20presses.

[36] Kevin ChenFounder / Writer at Rantle East Electronic Trading Co. (2023) *Everything you need to know about weight sensors*, *icrfq.net*. Available at: https://www.icrfq.net/weight-sensors/.

[37] SAVAGE, A.D. (2016) *Amazon*. Available at: https://www.amazon.com/dp/B0BWJHK4M6/ref=twister_B0CLH239GM?_encoding=UTF8&th=1.

[38] Stykemain, A. (2023) *How capacitive sensor work - explained: Types & applications*, *The Easiest Way to Learn Industrial Automation*. Available at: https://realpars.com/capacitive-sensor/.

[39] *University of Central Arkansas: UCA Center for Excellence in Teaching and Academic Leadership*. Available at: https://uca.edu/cetal/chat-gpt/#:~:text=One%20of%20the%20key%20features,stories%2C%20essays%2C%20and%20poems.

[40] *Weight sensor, compact single point load cell 200kg 100kg 60kg 40kg 20kg weight sensor, quality weight sensor, weight sensor supplier Forsentek Co., Limited*. Available at: https://www.forsensor.com/sale-15383724-compact-single-point-load-cell-200kg-100kg-60kg-40kg-20kg-weight-sensor.html.

[41] *HX711 weight / load cell 2 channel pressure sensor amplifier module for Arduino / DIY / Pic Envistia Mall*. Available at: https://envistiamall.com/products/hx711-weight-load-cell-2-channel-pressure-sensor-amplifier-module-for-arduino.

[42] *Load cell accuracy* (2019) *Understanding Accuracy When Choosing a Load Cell | Load Cell Central*. Available at: https://www.800loadcel.com/blog/load-cell-accuracy.html.

[43] *North Ridge Pumps*. Available at:
https://www.northridgepumps.com/article-211_air-operated-double-diaphragm-pumps.

[44] *The 1984 national minimum drinking age act National Institute on Alcohol Abuse and Alcoholism*. Available at:
https://alcoholpolicy.niaaa.nih.gov/the-1984-national-minimum-drinking-age-act#:~:text=The%201984%20National%20Minimum%20Drinking%20Age%20Act%2C%20%5B23%20U.S.C.%20%C2%A7,of%20receiving%20State%20highway%20funds.

[45] *AC adapter AC adapter - Energy Education*. Available at:
https://energyeducation.ca/encyclopedia/AC_adapter#:~:text=AC%20adapters%20convert%20a%20higher,voltage%20(within%20a%20tolerance).

[46] *Alcohol's effects on the body National Institute on Alcohol Abuse and Alcoholism*. Available at:
https://www.niaaa.nih.gov/alcohols-effects-health/alcohols-effects-body#:~:text=Liver%3A,Alcoholic%20hepatitis.

[47] *Different types of pumps* (2022) *HAOSH Pump*. Available at:
https://www.haoshpump.com/different-types-of-pumps/.

[48] *Fatty liver disease: Nonalcoholic & alcoholic steatohepatitis (NAFLD/AFLD) WebMD*. Available at: https://www.webmd.com/hepatitis/fatty-liver-disease.

[49] *How do peristaltic pumps work? albin*. Available at:
https://www.albinpump.com/en-us/news/how-peristaltic-pumps-work.

[50] Kelechava, B. (2023) *NSF/ANSI 25-2023: Vending Machines for food and beverages - ANSI blog*, *The ANSI Blog*. Available at:
https://blog.ansi.org/nsf-ansi-25-2023-vending-machines-food-beverages/#gref.

[51] *NFPA 170 NFPA 170: Standard for Fire Safety and Emergency Symbols*. Available at:
https://www.nfpa.org/codes-and-standards/all-codes-and-standards/list-of-codes-and-standards/detail?code=170.

[52] *NSF/ANSI 61: Drinking water system components – health effects NSF*. Available at:
https://www.nsf.org/knowledge-library/nsf-ansi-standard-61-drinking-water-system-components-health-effects.

[53] *Peristaltic pumps How Do Peristaltic Pumps Work | Verderliquids.com - Verder Liquids*. Available at:
https://www.verderliquids.com/us/en/pump-technologies/how-do-peristaltic-pumps-work.

[54] *Useful information on centrifugal pumps Michael Smith Engineers ltd*. Available at:
https://www.michael-smith-engineers.co.uk/resources/useful-info/centrifugal-pumps.

[55] I2C-bus specification and User Manual - NXP Semiconductors,
https://www.nxp.com/docs/en/user-guide/UM10204.pdf.

[56] Uberuser, "The average ball valve life expectancy & the best replacements," Everlasting Valve Company,

https://www.everlastingvalveusa.com/average-ball-valve-life-expectancy/#:~:text=A%20ball%20valve%20is%20a,about%20eight%20to%20ten%20years.

[57] Charlotte, "Determing the lifetime expectancy of a pinch valve sleeve," AKO UK Pinch Valves, https://www.pinch-valves.com/2021/10/22/determing-the-pinch-valve-sleeve-lifetime-expectancy/.

[58] "Solenoid valve maintenance - an overview," Beta Valve, https://www.betavalve.com/Wiki/42/Solenoid-Valve-Maintenance---An-overview#:~:text=Valves%20that%20have%20to%20work,then%20replacing%20the%20affected%20components.

[59] C. Kolstad, "Ball valve - how they work," Tameson.com, https://tameson.com/pages/ball-valve-introduction.

[60] C. Kolstad, "Pinch valve - how they work," Tameson.com, https://tameson.com/pages/pinch-valve.

[61] J.-W. Pustjens, "Solenoid valve response time," Tameson.com, https://tameson.com/pages/solenoid-valve-response-time.

[62] J.-W. Pustjens, "What is a solenoid valve and how do they work," Tameson.com, https://tameson.com/pages/solenoid-valve-types .

[63] "IEEE Standard for Software and System Test Documentation - Redline," in IEEE Std 829-2008 (Revision of IEEE Std 829-1998) - Redline , vol., no., pp.1-161, 18 July 2008.

[64] "New recipes," Barsys. Available at: https://thebarsys.com/pages/new-recipes.

[65] "Smartender self serve," *Smart Bar USA*. Available: https://www.smartbarusa.com/smartender-selfie.

[66] *"It's the opposite of art": Why illustrators are furious about ai* (2023) *The Guardian*. Available at: https://www.theguardian.com/artanddesign/2023/jan/23/its-the-opposite-of-art-why-illustrators-are-furious-about-ai.

[67] Matan, "Push-pull converter: How it works, application & advantages," Electricity, https://www.electricity-magnetism.org/push-pull-converter/.

[68] "How to design a push pull converter – basic theory, construction, and demonstration," Circuit Digest - Electronics Engineering News, Latest Products, Articles and Projects, https://circuitdigest.com/tutorial/how-to-design-a-push-pull-converter.

[69] "Design by: V11 - SparkFun Electronics," SparkFun Electronics, https://cdn.sparkfun.com/assets/f/5/5/b/c/SparkFun_HX711_Load_Cell.pdf.

[70] *Getting started with VS code IDE ESP*. Available at: https://docs.espressif.com/projects/esp-idf/en/v4.3-beta3/esp32/get-started/vscode-setup.html .

[71] Microsoft (2021) *Get started with visual studio code*, *RSS*. Available at: https://code.visualstudio.com/learn#:~:text=Visual%20Studio%20Code%20is%20a,%2B%2B%2C%20JavaScript%2C%20and%20more.

[72] *3.5" TFT color touch display-driver Ili9488 for NODEMCU ESP32* (2023) *YouTube*. Available at: https://www.youtube.com/watch?v=NidjZOxMMxo.

[73] *Bodmer/TFT_eSPI: Arduino and platformio IDE compatible TFT library optimised for the Raspberry Pi Pico (RP2040), STM32, ESP8266 and ESP32 that supports different driver chips*, *GitHub*. Available at: https://github.com/Bodmer/TFT_eSPI.

[74] *ESP32 pinout: How use GPIO pins?* (no date) *uPesy*. Available at: https://www.upesy.com/blogs/tutorials/esp32-pinout-reference-gpio-pins-ultimate-guide#.

We, Group 38, hereby declare that we have not copied any material from Chat GPT or similar AI platforms; we have simply used that platform as a tool for planning and proofreading purposes.

Appendix B: List of Tables

Appendix C: List of Figures