THE OHIO STATE UNIVERSITY

Project Name: The Clever Cleaner Container

ECE 4900: Capstone Design II

Autumn 2021

Final Design

The C^3

Tue, Thu 3:55 PM

Assignment Due Date: 12/7/21

Team members: Daniel Welsh,
Bryce Goldman, Eric Killian,
Yu Chengfeng, Zun Yang

Table of Contents

List of Figures & Tables:

Executive Summary

It is often the case that people have to store several cleaning liquids in their kitchens, bathrooms, laundry rooms, or garages. This can lead to clutter and disorganization. The primary objective of this project is to make the storage and usage of cleaning liquids more convenient.

This device is able to hold up to four different cleaning liquids at once, with a separate tank for each within a singular outer container. The user will be able to electronically select one of the four liquids and choose an amount to dispense.

There are several requirements that must be met in order for this device to be practical. It must hold enough liquid to avoid the hassle of constantly needing to refill it, but must also be small enough to fit on a small counter and in an average person's hands. It must also be light enough, even when filled, to be carried around by an average person with ease.

After the initial design idea was accepted, brainstorming began for the device itself. Features including the nozzle, power supply, and user interface were selected according to what seemed most reasonable for our specific implementation. Other design concepts with various features were also considered to implement our solution, which was to create a device that can hold and dispense multiple liquids separately where the liquid being dispensed is selected electronically.

Several design constraints had to be considered in the conception and development of this project. With a tight budget, short timeline, and limited experience among the group members, the potential solutions were limited. Constraints regarding the electrical and mechanical components of the solutions were considered as well. Due to the use of electrical components as well as fluids in the project, engineering standards regarding both fluids and electricity were also taken into account.

The work for this project was divided into several parts and organized in a Gantt chart. First, individual components of the device were tested in order to ensure they were not defective. Next, the software for the liquid selection and the hardware for the device was developed and tested. Finally, the whole device was assembled and tested to eliminate any lasting issues. The final testing of the device was performed to ensure its compliance with the requirements specified in the Requirement Traceability Matrix.

The length and detail involved in this semester-long project warranted a detailed plan. Thus, a Gantt chart was created to outline the various tasks for the group and when they were supposed to be completed. A timeline and work breakdown structure were also developed to keep the group on track and the workload even.

Several potential risks throughout the duration of the project were considered. The considered risks were both technical and management related. Technical risks involved things not working as expected as well as imperfections in the function of the device. Management risks included lack of communication between group members, deadlines not being met, and unforeseen tasks arising and interfering with ongoing tasks. Unforeseen challenges led to unforeseen lessons being learned by the group several times throughout the semester.

Overall, this device should serve as a way for individuals to make their lives at home more convenient. Nearly everybody has many bottles of liquid in their kitchens, bathrooms, garages, and even elsewhere in their homes. This device attempts to remove some of the clutter created by all these bottles. Instead of having to search for four different liquids, the user can simply select a liquid and dispense it. Selecting a custom amount will also help the user save liquid, and therefore money, in the future. Potential future enhancements for this product include an attachment to mount it under a sink, different types of nozzles for different sprays, and wireless capability for charging and communicating with the user's smartphone.

Introduction

Purpose

The purpose of this report was to first describe how cleaning is time consuming and cumbersome. The common household has several small bottles of cleaning sprays and liquids stored in cabinets that take up time finding and switching between while cleaning. This report will outline the solution and implementation plan that Capstone Design II Electrical & Computer Engineering students at Ohio State University came up with to create a more organized and quicker device to store and use these cleaning supplies while saving money from buying cleaners in bulk. This document describes in detail the exact steps and procedure the group went through implementing the Cleaning Device to a level such that another team of Engineers could easily replicate this product by going through this report.

Background and Motivation

Cleaning is an arduous and daunting task that millions of household people and workers struggle with each year. The average person spends 13 minutes per day cleaning up the kitchen after meals [1]. According to the 2018 ACI National Survey Stats on Cleaning Habits, 74% perform light cleaning most often [2]. Small cleaning bottles also cost more than larger cleaning jugs, for example Fabuloso® Lavender Multi-Purpose Cleaner 128 oz costs 6.79 and Fabuloso Lavender Multi-Purpose Cleaner 16.9 oz costs 1.39 [3]. People do small cleaning throughout the day often, and waste time carefully measuring out cleaning amounts and searching for their cleaners. Dish soap, countertop cleaner, and disinfectant are routinely needed and can be annoying to go out of your way to find. We live in a technology-driven society in which everyone is trying to make their lives as easy and as convenient as possible while maximizing productivity. A device is needed to allow easy access to various cleaning liquids and to switch quickly between to save time and make cleaning easier.

Problem Statement

The objective of the Clever Cleaner Container (the C^3) was to provide people with a quick and easy way to access their cleaning liquids. This device could be used in the kitchen, in the bathroom, for cleaning in restaurants, and other endless situations. The device will have four refillable cleaning supply tanks for various cleaning products, as well as a programmable dispenser nozzle that allows quick and easy access to the four cleaning supplies at the push of a button. The interface on the Clever Cleaner Container nozzle will contain an easy to use array of buttons to access the various cleaners and program the amounts of cleaners needed. The Clever Cleaner Container will decrease the tediousness of this mundane task for the user, allowing them to spend time and effort in more productive ways.

Overview

This report will first outline the problem, and describe the basic motivation behind the device. Then the report includes requirement specifications including a requirement traceability matrix and the initial thoughts on the project of the group. The report then covers some of the initial features for the product that the group had and rejected, before covering the final design the group made. The report then covers some alternative designs that the group had. The report then covers how the prototype was constructed and designed. Then the constraints of the project are discussed to show some of the limitations the group faced when working on this project. The report then covers the standards that impacted this project transitioning into the validation steps that were used to test that this product performs as intended. Next the report goes over the project management aspects including the Work Breakdown Structure and Attribute, Gantt Chart, Timeline, Budget and Expenses, Risks, Challenges, Problems, and Lessons Learned. Lastly the report summarizes the proposed benefits, future enhancements, and next step recommendations.

Technical Content

Concept Overview

Additional description of concept of the product and solution (if needed beyond introduction)

The C^3 (Clever Cleaner Container) was a quick and easy way to dispense various precise amounts of cleaning supplies quickly. The device consisted of four refillable cleaning supply containers, a programmable smart dispense nozzle, four electric DC pumps, a microcontroller, and a rechargeable battery. The Clever Cleaner Container could be used in a kitchen, a restaurant, at a store, or in many applications and makes cleaning quick and easy.

Requirement Specifications

Requirement infromation				Relationship Traceability				
ID	Requirment	Component	Priority	Need Objective	Owner	Validation Test	Acceptance Critera	Current Status
1	The device should be able to pump out the chosen cleaner between 16 ml to 17 ml in 10 seconds for each of its 4 pumps	Pumps	2	The device needs to pump out cleaner at steady rate close to its designed throughput (100ml/min)	Daniel	T1: Program pump A to dispence 15 mL and time how long it takes to dispences. T2: Repeat this for the other 3 pumps (B, C, & D)	T1: The total amount of cleaner that comes out should be close to 15 or 15 ml . T2: The total amount of cleaner that should dispence in 10 sec.	Closed
2	The device should be able to hold from 24 oz to 28 oz for a cleaner liquid in each of its 4 tanks	cleaner containers	2	The device should be able to hold enough cleaner so the user doesn't have to refill it frequently	Daniel	T1: Fill in 22 oz of one liquid cleaner into container A. T2: Repeat this for the other 3 containers (B, C, & D)	In both tests (T1 and T2) the container should be able to hold the liquid cleaner in that test.	Closed
3	The device should be charged easily through a standard wall outlet or some standardized power source	Control board	1	The device, which uses several DC motors, needs to operate through a control board that can be powered by a standard AC outlet	Bryce	Plug in the device	Each DC motor works, indicating it is receiving the correct DC power from the control board	Closed
4	The empty device needs to weigh under 10 pounds with no cleaner	All	3	The device needs to be lightweight and portable	Eric	Weigh the device while it is empty (no liquids in the container).	The empty device weighs under 10 pounds	Closed
5	The body (main part of the device) should be less than 10 and more than 8 inches for length, less than 6 inches and more than 5 inches for width, less than 12 inches but more than 8 inches for height	The main body	1	The device should be able to hold all motors, control board and containers	Zun	Measure the lenght, width and hight of the main body of the device	The main body's dimensions should measure within the specifed length witdth and height	Closed
6	The length of the tube between the nozzle and the main part should be greater than 1 foot but less than 4 feet	Tube	2	The length of the tube needs to be long enough for the user to handle different house cleaning	Chengfeng	Measure the lenght of the tube from the top of the nozzle to the bottome of the tube	Measurement should be within the specified lenght of the tube	Closed
7	One button on the 4x4 matrix keypad needs to trigge/dispense the output of the corresponding cleaner	4*4 matrix keypad	1	The device needs to output the chosen cleaner	Chengfeng	T1: Push button X (one of the A, B, C, D) then press the trigger button	T1: container X should output liquid based on the user input	Closed
8	One button on the keypad is the program button	4*4 matrix keypad	2	The user could change the defult volume of the chosen cleaner	Eric	Press the program button to allow the user input numbers	The device will dispenses the input numbers of cleaner volume from the user	Closed
9	The device needs a numeric keypad with digitis 0-9	4*4 matrix keypad	2	The user will input an amount of cleaner to be dispensed	Zun	User selects 1 mL, 52 mL, and 100 mL (dispensing between each selection).	Amounts that are dispensed are 1 mL, 52 mL, and 99 mL, respectively	Closed
10	The device needs four push buttons to select the active cleaner that you want to dispense fluid from	4*4 matrix keypad	1	The user needs to be able to select which cleaner to use	Bryce	Fill each container with a different liquid then push button for each cleaner (one at a time) and press the dispense button.	The correct liquid dispenses for each selection.	Closed
11	The device must be able to handle a maximum viscosity a peristaltic pump can handle is about 2000 cps	Pumps	2	The user needs to select cleaners under this viscostiy for pumps to have accurate throughput	Daniel	Cleaning liquids with viscostiy up to this limit will be pumped though the pumps and dispensed into meaurable container to test throughput	Pumps will dispence cleaning liquid with accurate thoughput with liquids under 2000cps	Closed
12	The device must indicate which cleaner tank is selected and how much cleaner has been programed	OLED screen	2	The user should be able to tell which cleaner they are using and how much they are using to avoid confusion and increase usability	Bryce	Select a cleaner tank with key pad and look at the OLED screen to indicate the selection and programed amount	The OLED screen should display A if cleaner tank A is selected and the Correct amount of ml of cleaner programed to dispense. Repeat this for selections B, C, & D	Closed
13	The device must have some way to indicate which cleaners are in which tanks by allowing the user to write or erase the names of cleaners on the lables of tanks A through	The main body	4	Many cleaners look similar and the user should have little difficulty telling them apart	Zun	Write on each of the labes on the tanks and erase them after 3 min	Each lable should be able to be written on. After 3 min erase what was written and the lables should be completly clear	Closed
14	The device should be able to hold a charge for at least 30 mins of use after full charge	Power supply/battery	5	Users may need to clean areas not near a wall outlet around the house. Being able to work for 30 min without needing pluged in would decrease the user's cleaning time	Zun	Charge the device	The device still has power	Closed
15	The device should be able to output at least 2500mL of liquid on a full charge (device will consume a small amount of power running idle)	Power supply/battery	5	The device needs to be able to output 2500mL of liquid in a single charge	Chengfeng	Charge the device	The device still has power	Closed
16	FIll each tank with cleaner. Once pumps are primed to the tip, each tank (A-D) should be able to ouput 1 to 99 ml with +/-5ml of accuracy. using 20ml, 40ml, 60ml, & 80ml to test each tank. Repeat each 3 times and take average.	pumps/code	1	User is able to have a range of volume to select and dispence semi accurately	zun	Prime pumps, program amounts, disepnce and measure	Average from each tank should be within +/- 5ml for each test	Closed

Table 1: Requirement Traceability Matrix

The group chose the requirements listed above to make sure that the Clever Cleaning Container was made to the right specifications. The group wanted the device to be portable, accurate, and easy to use which went into many of the decisions made. For example, the device needed an easy to use screen, a programmable interface, and the device needs to be able to pump between 1 and 100mL upon request by the user.

Rejected Alternate Features

Aspect	Solutions 1	2	3	4
Spray chosen cleaner	Nozzle gun	Trigger sprayer	Selection nozzle	
User Input	A 4 * 4 matrix keypad	Individual Button Inputs	Touch screen	Smartphone App
Body shape	Cylinder	Rectangle		
Power supply	Battery supply	Wired supply		
Output control	Smartphone App	Knob to control the output rate	A 4 * 4 matrix keypad	
System setup	Arduino board	NodeMCU board		
Pump	12 V DC Pump			
User Display	OLED	No Display	LCD	LED
Microcontroller	STM32	Arduino	ESP32	MSP430FR

Table 2: Design Concepts Table; potential solutions to different problems are listed

As seen above the group considered many other approaches before converging on the final design. Some alternate features that were considered included various other microcontrollers, a Smart-Phone application that could interact with the device, a knob to control the flow rate for each dispense mechanism. The group ended up deciding that many of these options were simply unnecessary or would violate time and experience constraints faced by the group. For additional Rejected Features see morph chart above.

Design

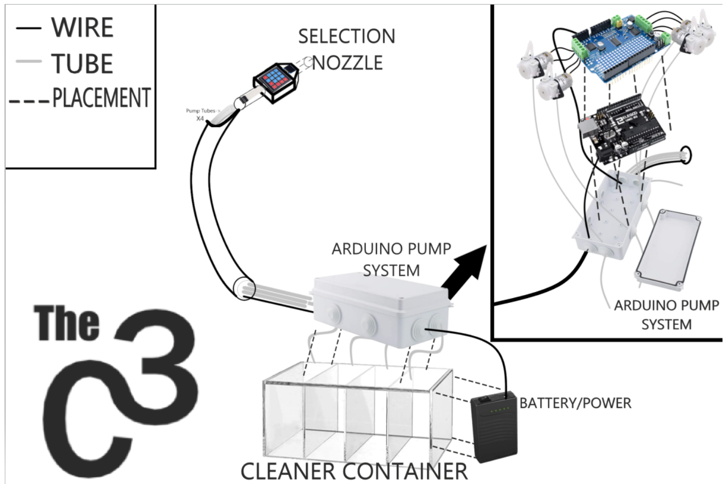

Figure 1: Diagram of Final Design

<u>Final Design</u>: Nozzle gun, A 4*4 matrix keypad. Rectangular container, Battery supply, A 4*4 matrix keypad, Ardunio board, 12 V DC Pump, OLED screen, Simple Arduino Microcontroller

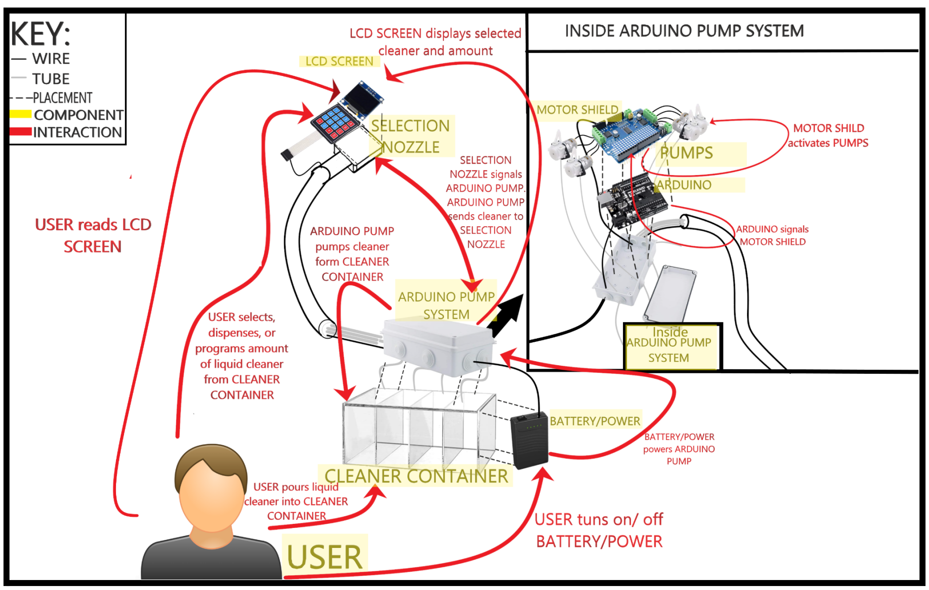

Figure 2: Component placement and interaction diagram

From the diagram, the cleaner container will be the base in which all the components are mounted or connected too. The 12v battery/power supply will be mounted to the side using velcro so that it can be removed to be charged or replaced, however, it can be charged while connected to the base. The C^3 can still operate while the battery is charging so long as it is switched on.

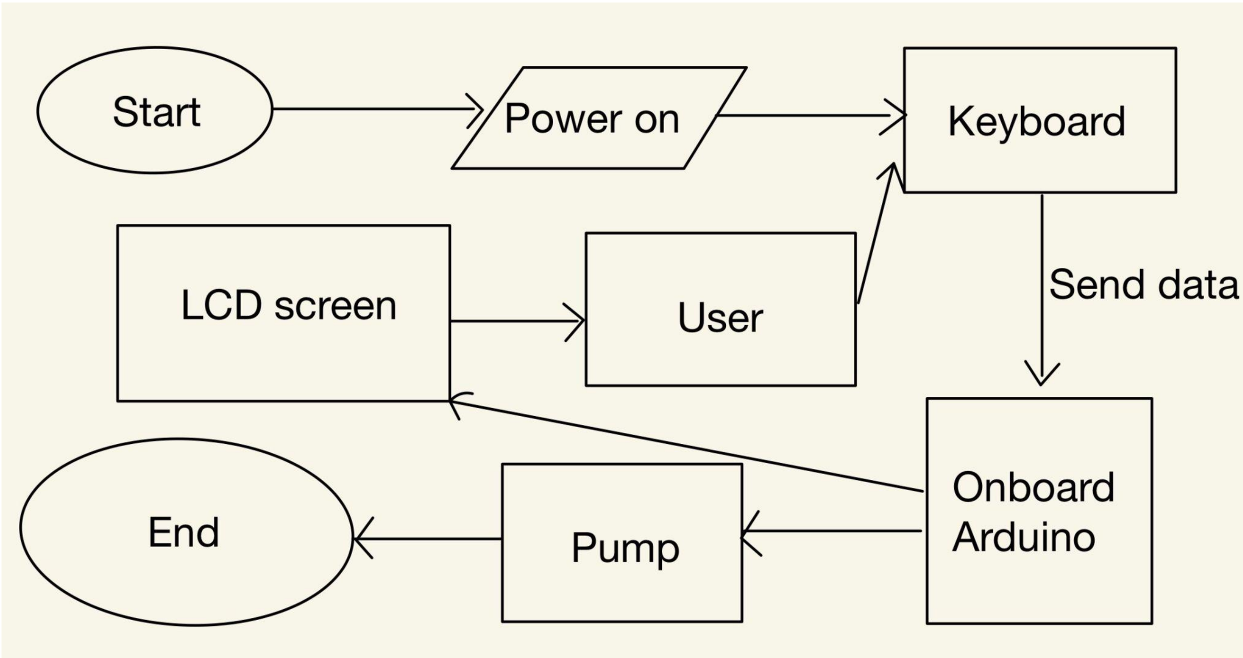

Figure 3: Basic Functionality Flow Chart Diagram

On the base of the project was the arduino pump system. This was a waterproof which holds the main electronic components which run the pump system. The components include Arduino Uno board, motor shield, and four 12V DC pumps. The Arduino Uno board was the CPU which was programmed to run the pumps based on the user's cleaner selection and programmed quantities. The board sent the signal to the motor shield which activated one of the four pumps to pump the cleaning liquid from the cleaner container, through the tubing, to the selection nozzle.

The selection nozzle was where the user had most control and where they operated the pump system. The user used the keypad to select their cleaner from tanks A through D, programmed the quantity in milliliters, and dispensed the liquid. Each pump was selected one at a time using the A-D keys. The user programed their selection by pressing the '*' key, entered their quantity from 1-99ml and confirmed their selection with the '*' key. The user then used the '#' key to dispense the selected cleaner.

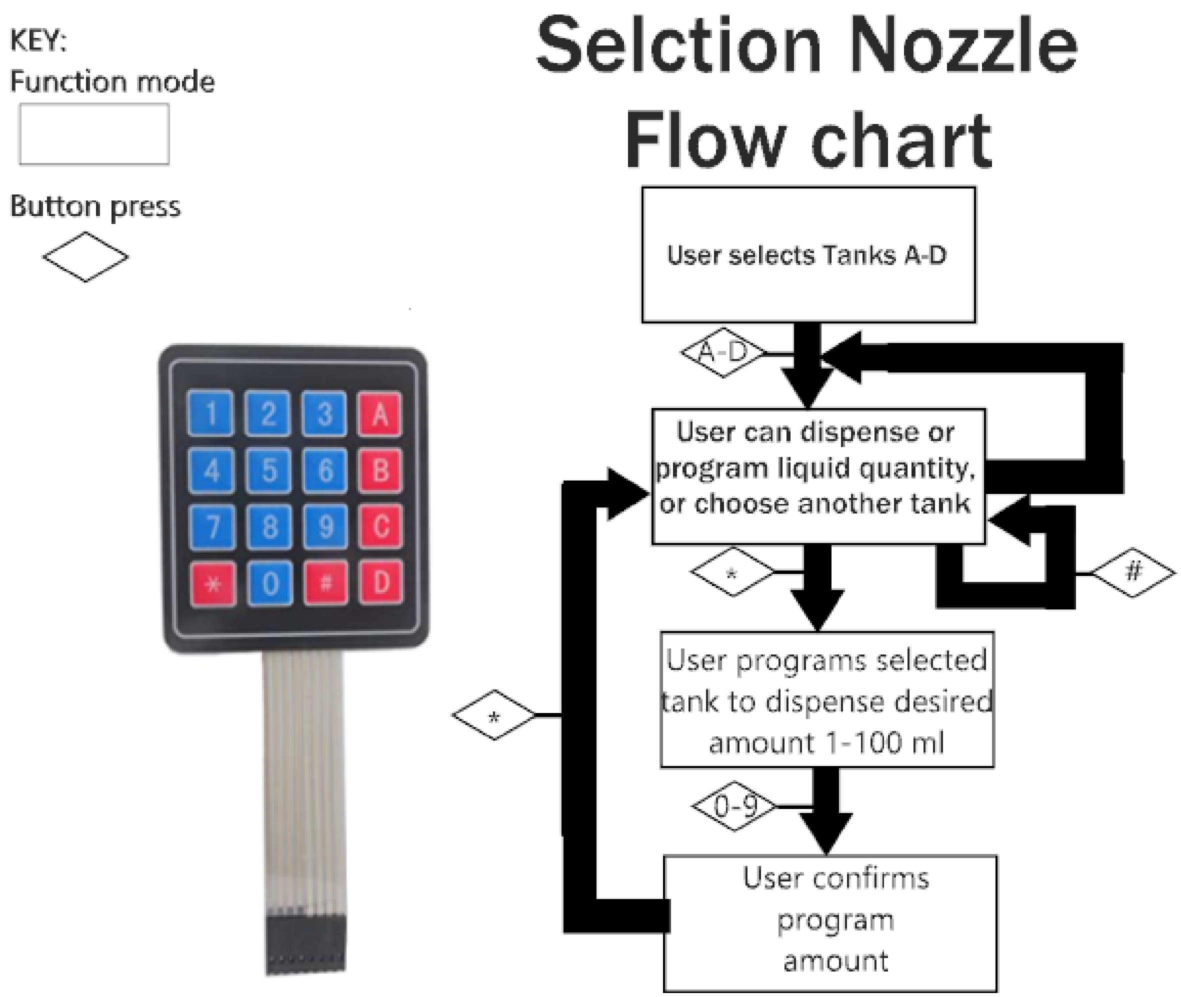

Figure 4: Selection Nozzle User Flow chart

The user saw their selection and quantity from the OLED screen mounted on the selection nozzle. The programmed amount was stored in the pump system's memory. This amount was pumped from the selected pump to the selection nozzle.

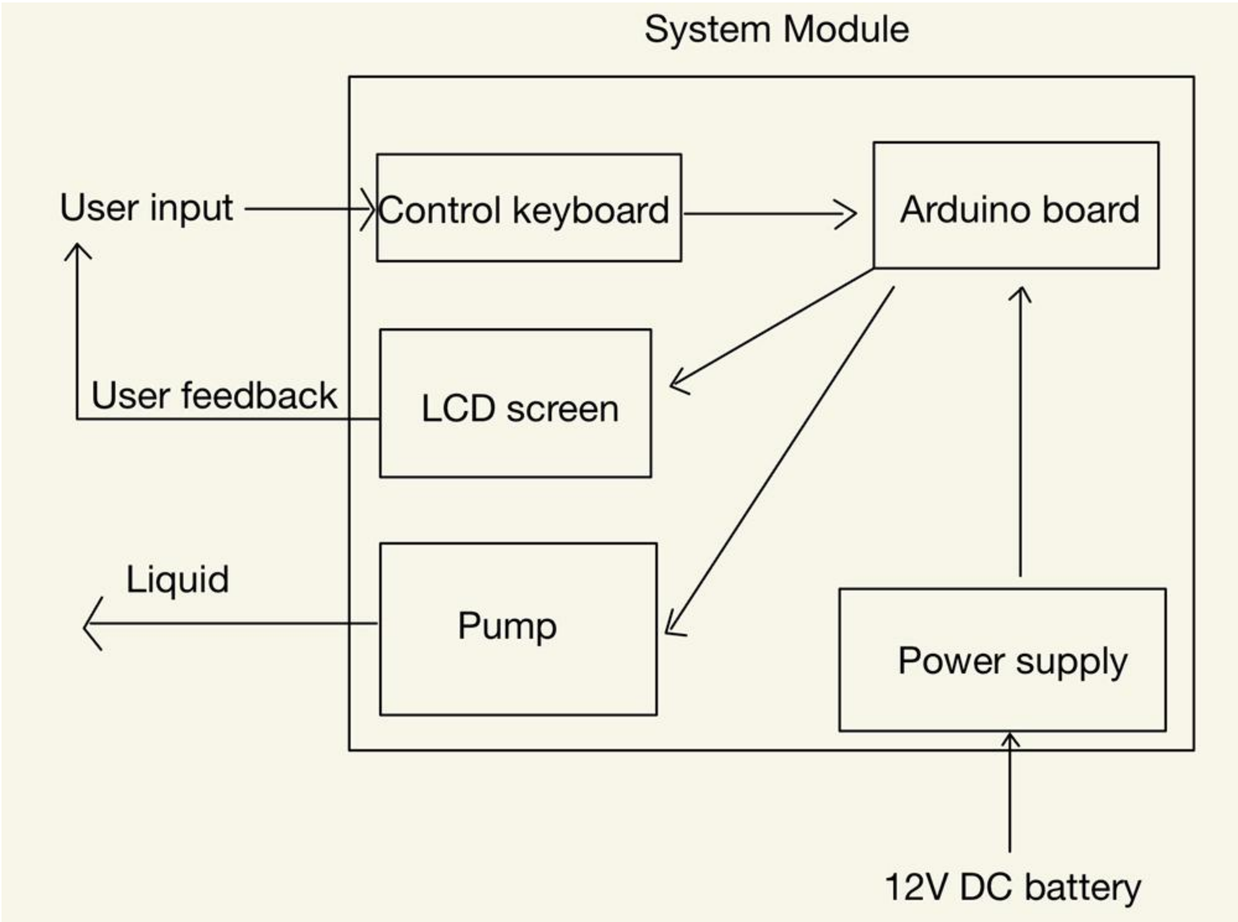

Figure 5: System Module Flow Chart

The tubing from the four pumps was run into the nozzle's handle through the hollowed out middle tunnel, and out the end. Meanwhile, the wiring for the screen and keypad was run down through this area and back to the pump system. The design for the selection nozzle can be seen later in the report. The parts were mounted and sealed with a clear silicon liquid seal and adhesive foam rubber to prevent leakage of cleaner to other components as this may be a risk.

Rejected Alternate Designs

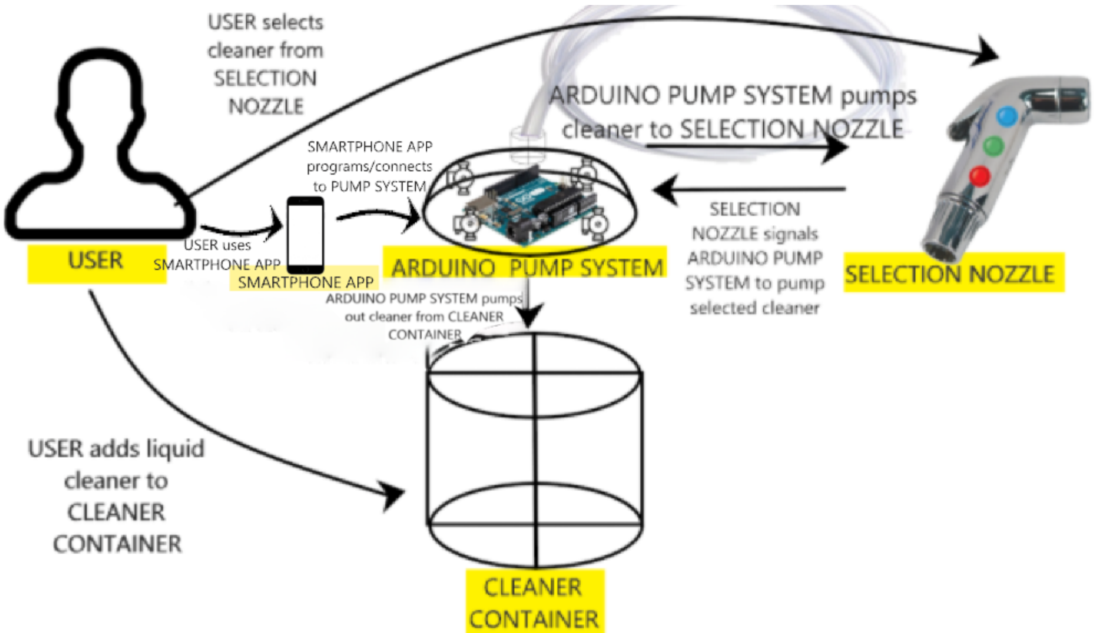

Figure 6: Draft drawing of design concept 1

<u>Rejected Design concept 1:</u> Nozzle gun, Individual Button Inputs, Cylinder, Battery supply, Knob to control the output rate, Arduino board, LED, Arduino

The first design concept included a selection nozzle that sprays detergent at a user-controllable speed (controlled by a knob). The device featured a waterproof cylinder body part container with a semicircle shape cap containing the Arduino board, the pump system and the wires between them. The waterproof main body was cylinder form, 10 ~ 12 inches long, 2.5 ~ 3 inches for radius and 8 ~ 10 inches high. On the selection nozzle gun, there will be four buttons and a knob for connecting the motor to the Arduino board. There was to be an LED screen on the waterproof body part that showed a letter that represented the chosen cleaner. The four buttons were to allow the user to select different types of cleaners chosen by the user. The user was to press one of the letter buttons to select a cleaner, and adjust the knob to control the flow rate of the cleaner. The last button on the surface was the one to control the start and end of the spraying liquid. This design used a smartphone app thus it required programming skill to make connections between the Arduino board and smart phone through Bluetooth.

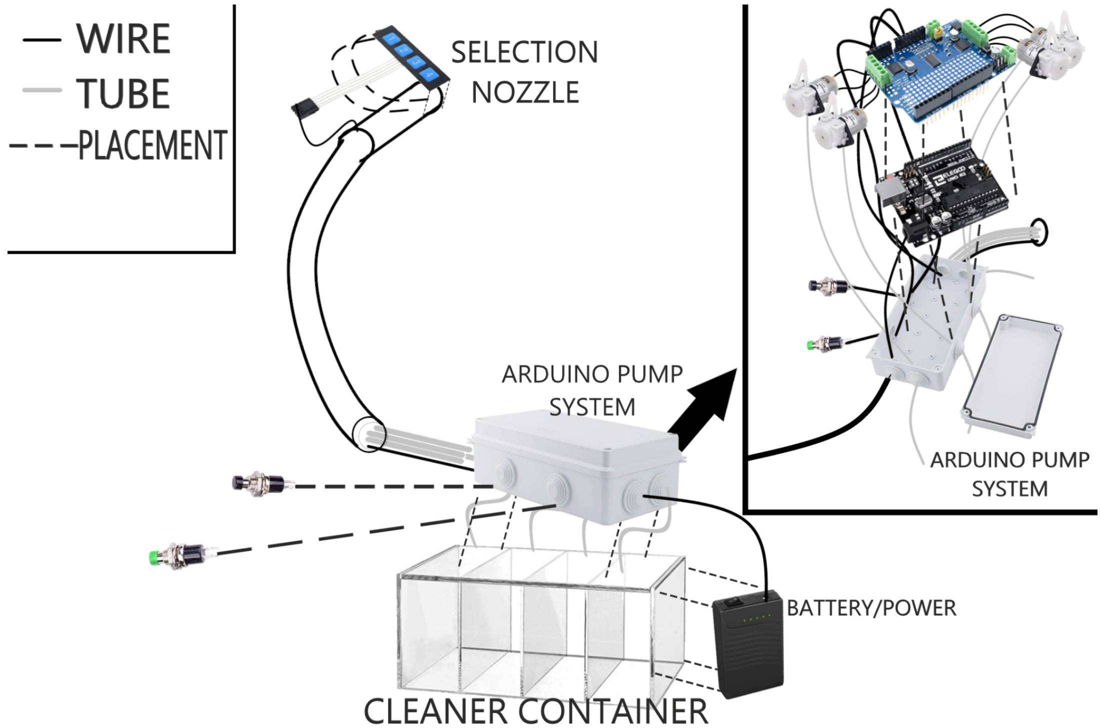

Figure 7: Draft drawing of rejected design concept 2

<u>Rejected Design concept 2:</u> Selection nozzle, Smartphone App, Rectangle, Battery supply, Knob to control the output rate, Arduino board, No Display, Arduino

The second design concept included using a selection nozzle that sprays cleaner at the user control rate. The user was to install an app on their cell phone to connect to the device through Bluetooth. In that app, the user would have been able to choose and name each cleaner. The user could also have inputted a number to change the output rate of a chosen cleaner. The body part included four cleaner containers designed to be a rectangular form to hold more cleaner and save the user's time from filling cleaners. The Arduino board and pump system were to be placed in a box that was to be placed above the body part to avoid a cleaner leakage problem. The device would have had a rechargeable battery supply and lasted for at least 15 minutes using time. The Arduino board would have copied the user input and stored the number into the memory for future use. The app on the user's phone would have shown the remaining cleaner in percentage.

Description and implementation of prototype

The first stages of implementing this product consisted of running basic tests and getting the four motors to run with as well as receiving input from the keypad matrix and printing out the results over a serial connection to the ATmega32U4 (See Appendix for code used to perform these tests).
After adequate tests were performed and the group was confident that the keypad and DC motors worked, the group worked on the 3D printed parts for the assembly of the device and the final code that would run on the device.

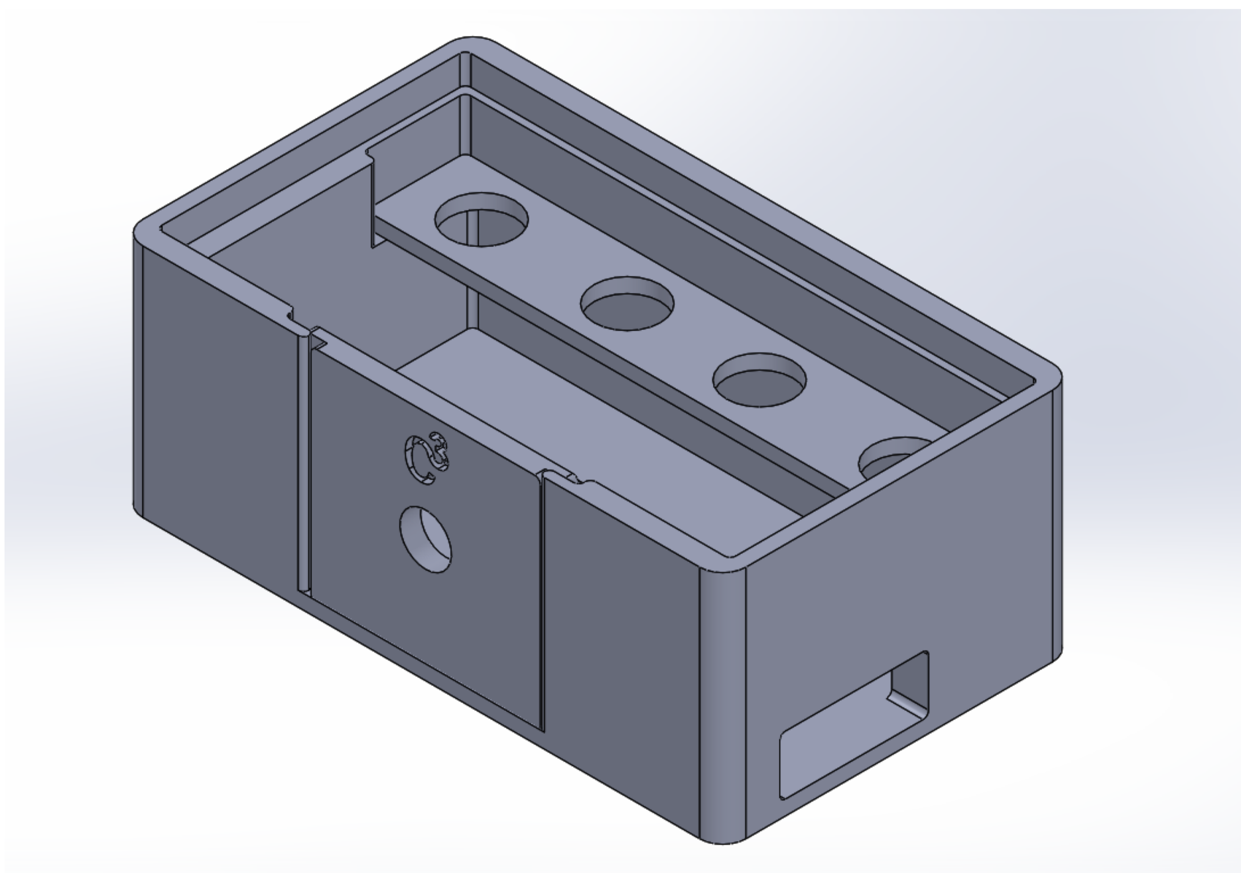

Figure 8: Solidworks Part Design for Arduino Pump System (Hose Mount, Pump Mount, and Case)

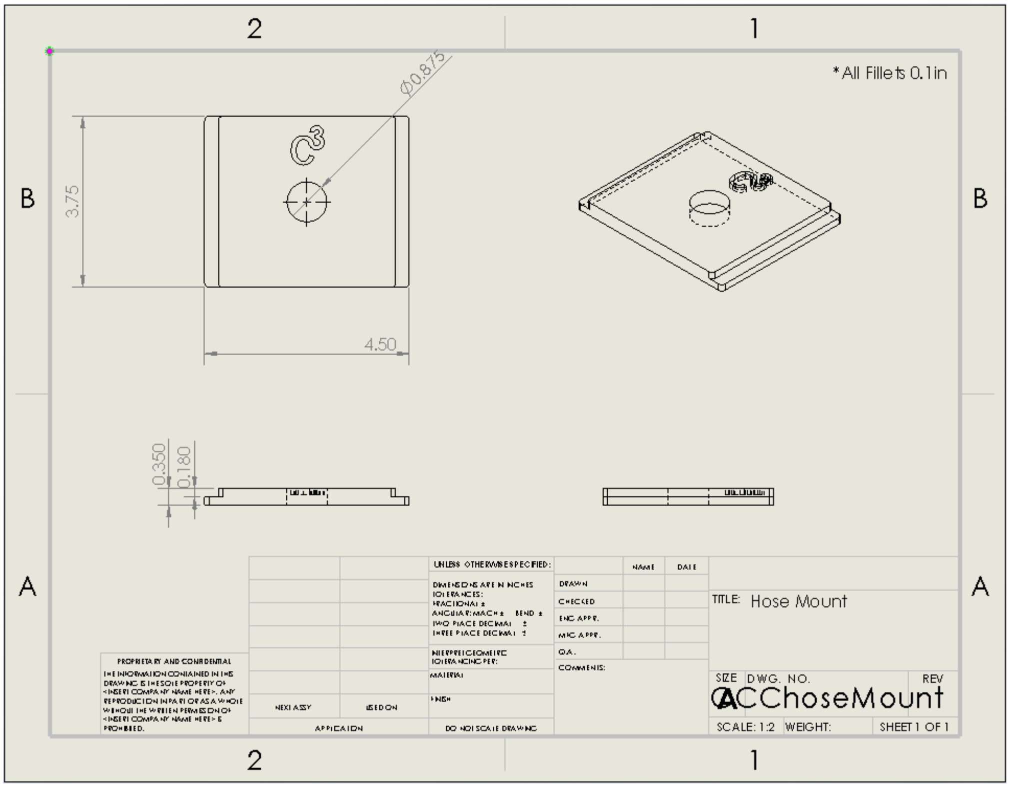

Figure 9: Orthographic view of the Hose Mount

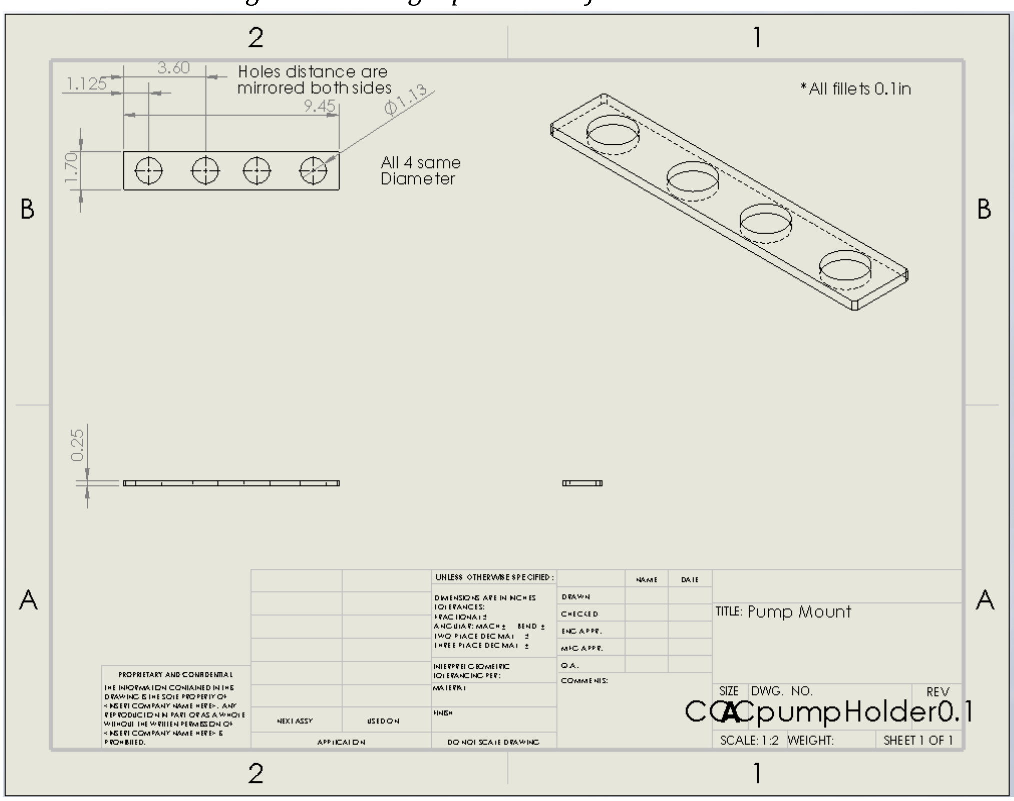

Figure 10: Orthographic view of the Pump Mount

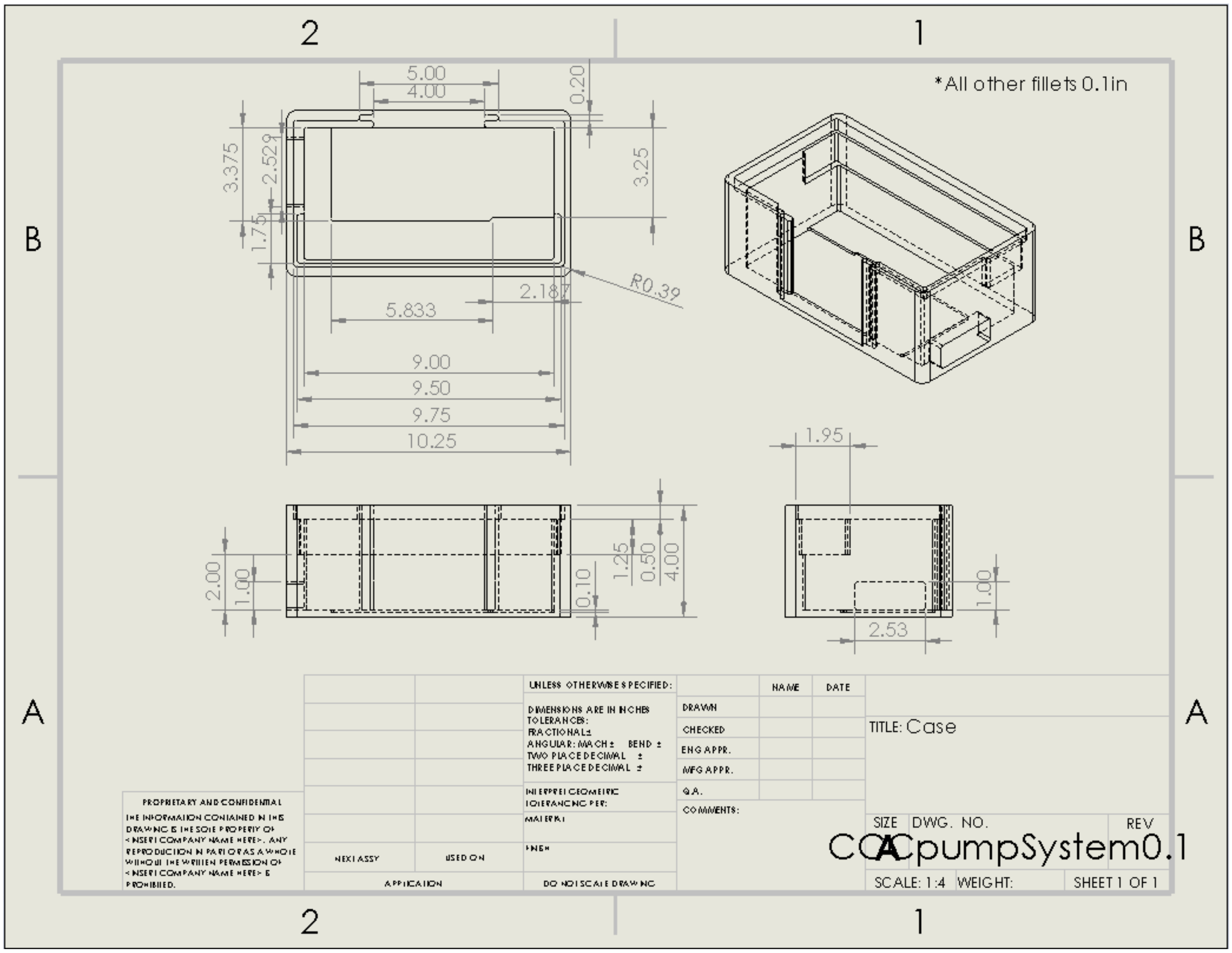

Figure 11: Orthographic view of the Case

The construction of the Clever Cleaning Container consisted of a 3D printed base unit. The purpose of this 3D printed part was to house the battery, microcontroller, DC pumps, as well as all the cables and fluid tubing for the selection nozzle. This base section was designed as three parts, one being a removable piece that housed the hole for the tubing (*Hose Mount)*, one being a removable piece that held the four DC motor pumps in place (*Pump Mount)*, and the other being the main base unit (*Case)*.

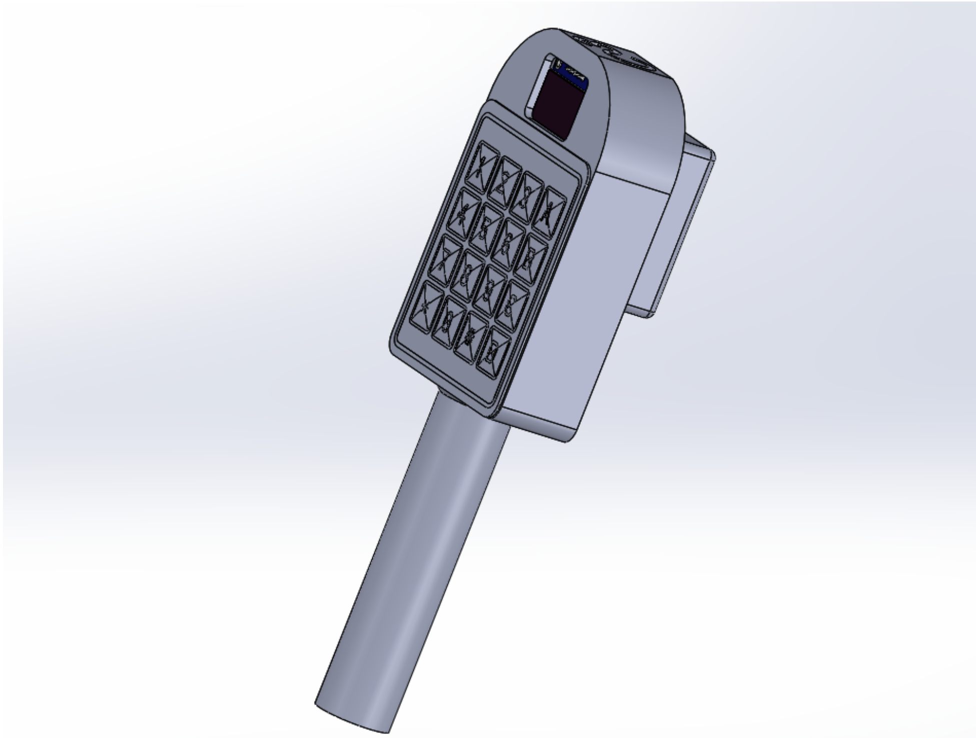

Figure 12: Solidworks Part Design for the Selection Nozzle Assembly (Nozzle Top, Handle, Nozzle Cover)

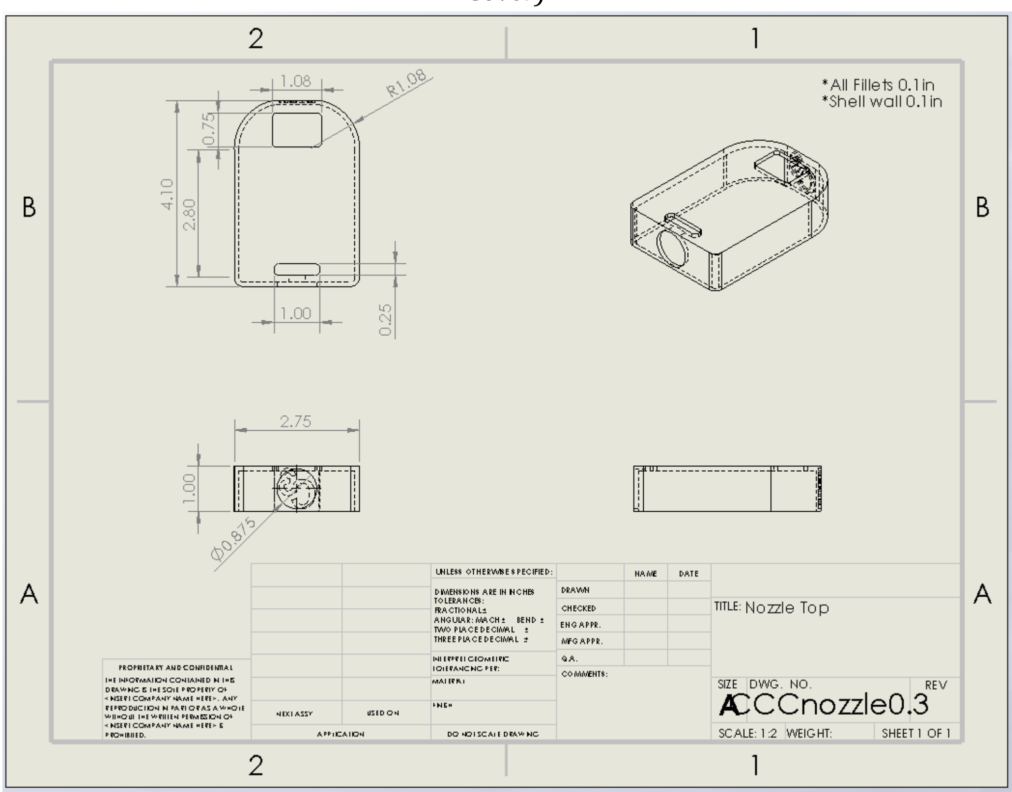

Figure 13: Orthographic view of the Nozzle Top

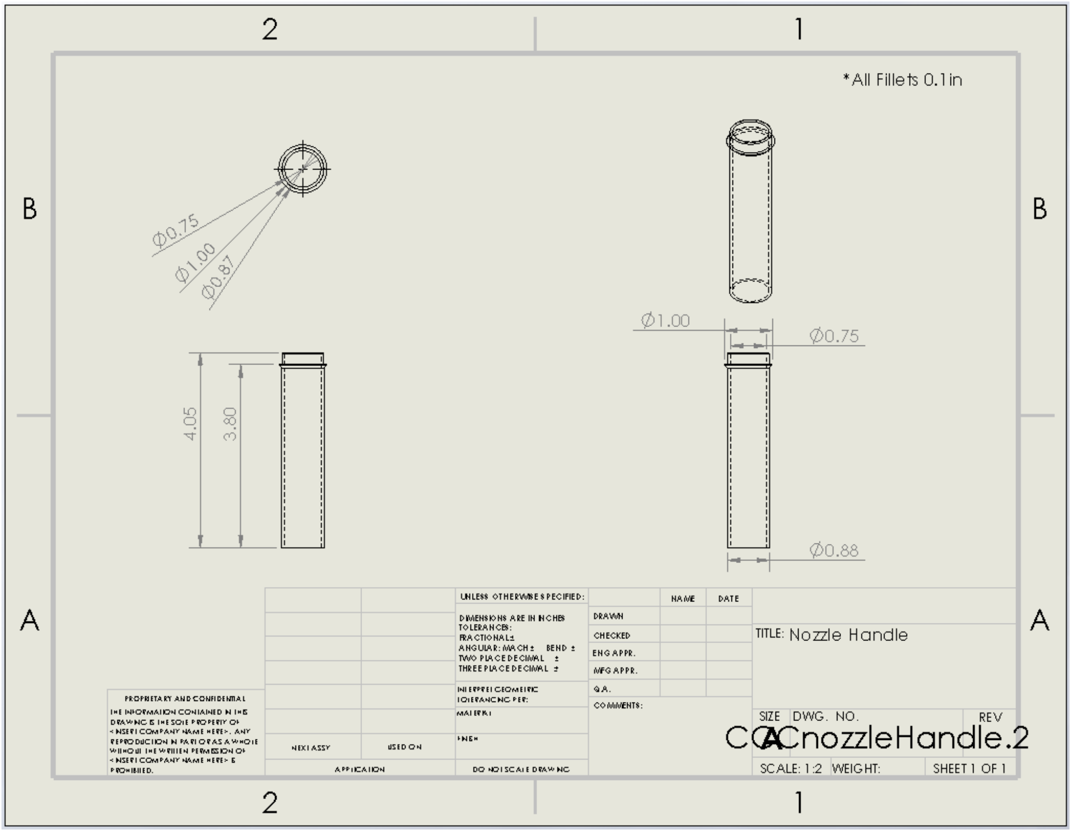

Figure 14: Orthographic view of the Handle

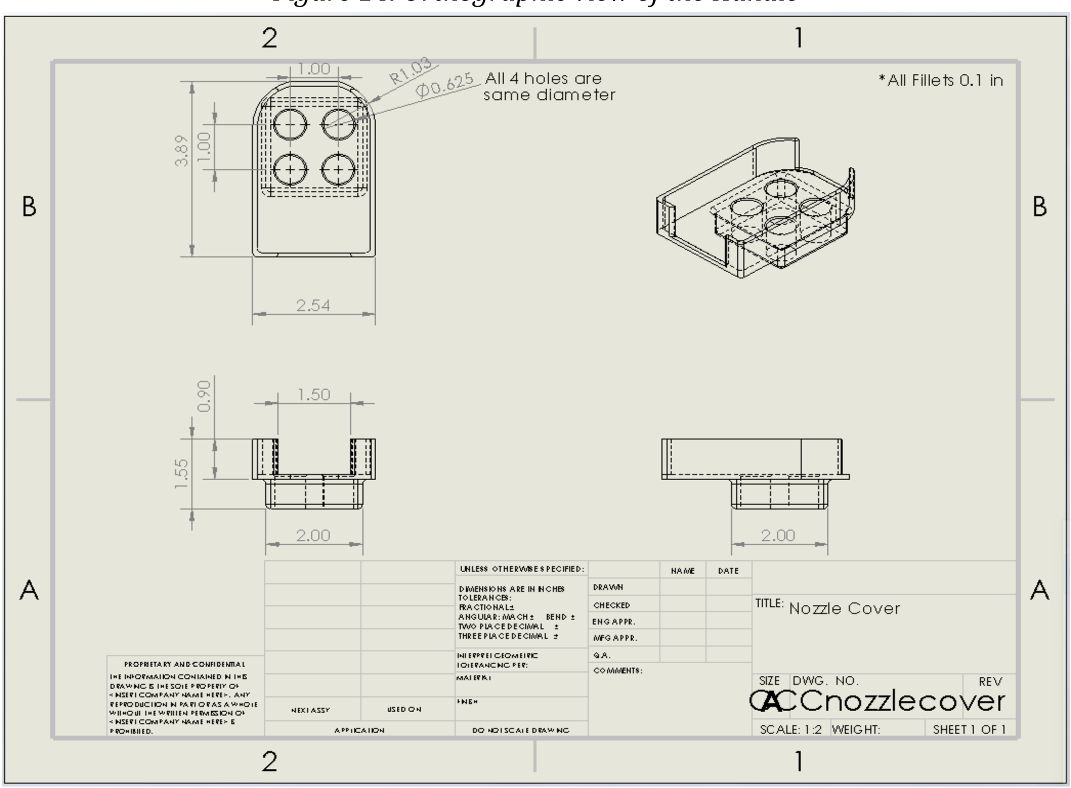

Figure 15: Orthographic view of the Nozzle Cover

The selection nozzle was constructed to provide the user an easy way to dispense cleaners. The selection nozzle was constructed as three pieces that connect together. The handle consisted of a hollow tube for the wires and fluid tubes to run through that connected to the other two parts of the selection nozzle. The Nozzle Cover is the piece on the back of the selection nozzle and houses the spraying units for the cleaner. The Nozzle top connects to the Nozzle cover and contains a slot for the OLED screen and the keypad matrix.

To simplify cable management for the keypad matrix and OLED screen the group soldered long wires with color-coded ends so each cable could be routed in the correct pin on the Arduino Uno Development Board. The cables were then combined together and routed through a long white tube that connected the base station to the handle of the selection nozzle.

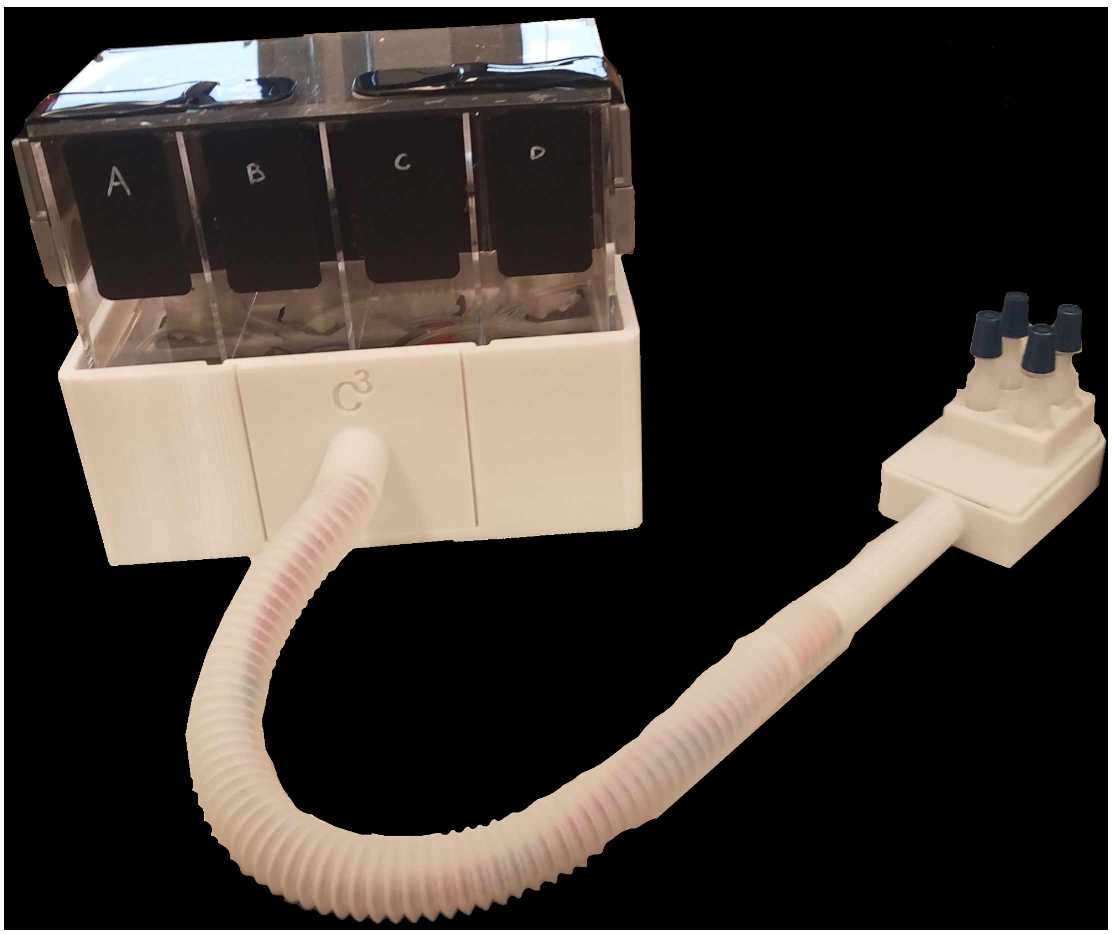

Figure 16: Final assembly with 3D printed Base and Nozzle assembled

The figure above shows an assembled picture of the Clever Cleaning Container. To Construct the device, the arduino and battery back were first glued into the base unit and connected together. The wires were then fed through the tubing that was purchased and connected to the selection nozzle and the hose mount. All pieces were secured and glued into place to secure them.

To connect the tubing to the rectangular containers a small hole was drilled in the center of the bottom of each cleaning container. The group then put the silicone tubing for each pump in the holes and sealed them with water resistance sealant.

Once holes were drilled into the container and the water sealant dried, the entire container was glued onto the arduino pump station. The arduino pump system was moved to the base of the device to allow the user to more easily access the four different compartments for the cleaning liquid. This led to more risk of leaks damaging electrons but increased usability.

Analysis and applicability of constraints

As with all Engineering projects the quality of the end product is limited by the constraints the group that is making the project faces. To become aware of these constraints the group decided to outline some of the major constraints that would be the limiting factors when making the Clever Cleaning Container. The group settled on six major constraints when designing the cleaning device:

1. Budget (Economic)
2. Group Experience
3. Time
4. Equipment
5. Electrical limitations
6. Mechanical limitations (Manufacturability)

One constraint the group had was Budget. As per requirement in ECE 4900 the group was not allowed to spend over $500 to complete the project. In order to work best around this constraint the group was smart about the components selected and worked hard to stay within the required budget constraint.However, the project went over budget and ended in a total of $643.36 due to 3D printing costs. All groups had a difficult time trying to find a way to 3D print parts. The organization that ECE 4900 used to work with for printing no longer offered their service so the professor had to find an alternative. Finally Professor Wolf offered his services on 11/11/21 as an advisor and part printer. After parts were sent for advice and redesign, the final cost of all parts was found to be $428.80 on the date 12/2/21. With no time to redesign or use cheaper methods the arduino pump system and selection nozzle pieces were printed and the team went over budget. Given more time the team could have used laser cutting plastic to create parts, or bought generic components to fashion into the necessary parts. These may have led to a cheaper design and allowed the team to stay under budget.

Group experience was another constraint that the group had. Our capstone group consisted of all students, not any company representatives working in the industry so all elements of the project had to be within the scope of the expertise of the group. For example the group considered adding capabilities to connect the device to a smartphone app however none of the group members had relevant experience in mobile app development so it fell out of the expertise of the group. Our group did have a fair amount of experience in electrical and mechanical design, as well as experience in embedded programming. The group added functionality to the device based on the experience levels of the team. Making the project watertight was difficult considering the team was inexperienced with plumbing and liquid physics. Redesign of container and tubing was done several times to correct leakage.

Time was one of the biggest constraints the group faced. The group had only one semester to come up with a design and complete the project, therefore the group had to move fast and create a detailed plan as to how the project would become completed. Time also constrains the amount of testing time the group would have when working on the project. Being able to do long term reliability tests on the product is hard when the group only had one semester to complete the project. To combat this the group put together a gantt chart describing what needed to get done when. The group made sure to leave lots of time for testing to compensate for the lack of time the group faced.

Equipment limitation constrained the group as well. When it comes to the manufacturing of the Clever Cleaning Container the group lacked the ability to manufacture lots of the parts and equipment needed to build a completely unique product from scratch. The group did have access to 3D printers, however the cost of 3D printing the entire project would consume a significant amount of time and money for the group. To counter this constraint the project consisted of many pre-made parts ordered off Amazon with some crucial 3D printed parts since the group lacked a large-scale manufacturing site for the project.

The group also made note of many Electrical constraints of components that would affect the end project. This included the speed and EEPROM non-volatile memory of the arduino microcontroller, the refresh rate of the OLED screen, and the programming limitations of the 4x4 keypad matrix and the voltage compatibility for the rechargeable battery. The group faced a big constraint of the arduino only having limited dynamic memory. This constraint was avoided by improving efficiency in the code.

Finally the group considered many Mechanical constraints for components. This included the weight of the battery, size of the cleaner holding cells, as well as mechanical limitations for how the device can be constructed. When designing the 3D printed enclosures the group did not originally realize that there were limitations to how 3D printed parts could be made. The group ended up having to print multiple parts for each 3D printed piece and make sure that the parts were designed in a way that could be printed correctly.

By adhering to all of these constraints the group was able to create a clear definition for the scope of the project and better understand how to move forward with creating the Clever Cleaning Container.

Standards and Regulatory issues
In order to make sure the Clever Cleaning Container is compatible and safe for all people, the group looked at various engineering standards that would ensure the C^3 would be a viable product. The group first looked at outlet standards defined by NEMA to make sure the cleaner could be used in North America. Our cleaning device needs to be able to be charged from a standard North American NEMA 5-15 110V AC outlet to ensure ease of use in the US.

Figure 17: NEMA 5-15 Standard Outlet Compatible

The group looked into standards for the hose in which the electrical and fluid lines would be routed in. The PVC NSF-51 standard for Standard Wall PVC Food and Beverage Hose seemed to align the most with the intentions of the group for the project. The hoses under specification of the standard are required to be non-conductive (Silicone-Free), Light Weight, Constructed with Non-Toxic Compounds, and suited for air and water-lines. This applies to the Clever Cleaning Container by defining a clear cut specification for how the materials and attributes of the hose that extends out of the device should be.

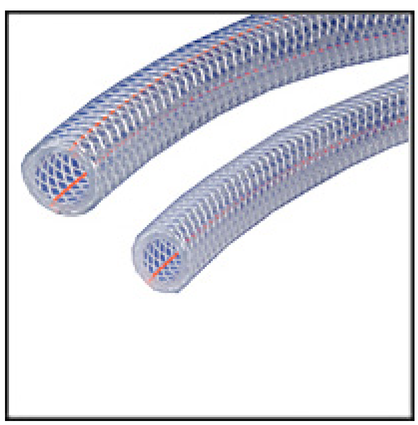

Figure 18: PVC NSF-51 Standard Wall PVC Food and Beverage Hose *[4]*

Additionally the group looked into formal safety specifications for the safety of electronics, protected enclosure, and electric cables submerged in water. The standards the group settled on included the IEEE 1019-2013 IEEE Recommended Practice for Specifying Electric Submersible Pump Cable—Polypropylene Insulation, the IEC 60529, and the general IEEE NESC - 2007 National Electrical Safety Code. These standards are important so that our device can be used around water and so that our device is as safe as possible.

Formal Engineering Standards
1. IEEE 1019-2013 - IEEE Recommended Practice for Specifying Electric Submersible Pump Cable—Polypropylene Insulation
2. IEC 60529 - Classification of Degrees of Protection Provided by Enclosures (outlines standards surrounding hazards for electronics near fluids)
3. IEEE NESC - 2007 National Electrical Safety Code (Water lines covered in Section 321B Part 4)
4. PVC NSF-51 Standard Wall PVC Food and Beverage Hose

These standards facilitated the development of the Clever Cleaning Container by providing a clear set of guidelines for how the group chose the silicon tubing for the cleaners, as well as the design of the Clever Cleaning Container making sure it was safe to use.

Validation

Container System:

The container system was quite simple, as its only function was to hold the liquids to be dispensed. The system consists of four separate tanks that each hold between 24 and 28 oz of liquid. It was measured to meet the size requirement of 8-10" length x 5-6" width x 8-12" height. It was also weighed to ensure that it was below the weight requirement of 10 pounds. Finally, each of the four tanks of the container system was tested to ensure that there were no leaks. Water was poured into each tank and allowed to sit overnight. There were four holes drilled on the bottom of each tank while motors and the Arduino board were placed under the container, so it was necessary to ensure the container was water-sealed. The volume of the liquid was the same as when it was poured into the tank the previous night, and there was no visual evidence of a leak, thus the particular tank was water-sealed.

Arduino Pump System:

The Arduino Pump System was one of the two electrical components of the device, the other being the keypad interface. This system consists of the Arduino board to implement the selection program, four pumps to dispense each liquid in the container, a motor shield to connect to and activate the pumps, and a plastic case to hold each of these components. The primary objective of testing this system involved the code since the motor was always running at the maximum speed, and the code decided the time that each motor ran. The first part of testing this system was a simple visual inspection to ensure that the components were not broken and fitted in the plastic case. Further testing followed:

1. Circuit/Power Test
   a. The Arduino board was connected to the motor shield such that the Arduino was providing power to the motor shield.
   b. The motor shield was connected to a pump such that the motor shield was powering the pump. The motor was receiving the correct amount of power.

c. The pumps were working, thus the circuit was complete and its components were working as expected.

2. Individual Pump Test
   a. Kept the same circuit as the previous test, adjusted the speed of the pump using the motor shield. Tried this for two other speeds, and the pump's speed changed as expected.
   b. Repeated (a) for the remaining three pumps. Several liquids with different viscosities were used(below the maximum requirement of 2000 cps).

3. Full Circuit Test
   a. The same circuit was kept as the previous test, connected all four motors to the motor shield.
   b. Each motor was tested individually at three different speeds. Everything behaves as expected, thus the mechanical components of the circuit were all functioning correctly.

Keypad Interface:
The numeric keypad was attached to the selection nozzle, but it had its own section because the testing involved was primarily separate from the nozzle. The function of the keypad was to select the liquid and an amount to dispense, which was a direct implementation of the code written for the project. This interface also consisted of an LCD that displayed the liquid and the amount being dispensed. The testing of the keypad interface was primarily concerned with the code and broken down into several functionalities tests:

1. Simplified Dispense Test:
   a. The primary objective of this test was to get the keypad to dispense a liquid when the dispense button was pressed. Specifying a cleaner and an amount is not part of this test.
   b. Written code that caused a specific amount (defined in the code) of liquid to be dispensed when the dispense key was pressed.
   c. The circuit was set so that the Arduino and motor shield powered a single pump and connected the keypad to the Arduino.
   d. Loaded the code onto the Arduino and pressed the dispense key. The pump dispensed the correct volume of liquid.

2. Selection Test:
   a. This test extended on the previous test, added the feature of selection.
   b. Written code allowed the user to select one of the four liquids in the container using the keypad's A, B, C, and D keys. The default amount was the same as the previous test.
   c. The other three motors were connected to the motor shield and loaded the code onto the Arduino.
   d. Pressed either A, B, C, or D to select a liquid, then pressed the dispense key to dispense it. Repeated this several times for each of the four liquids.

3. Volume Test:
   a. This text extended on the previous test but allowed the user to select a volume of liquid to be dispensed.
   b. Written code allowed the user to change the amount of liquid to dispense by pressing the program key. After the program key was pressed, the user inputted a numerical value between 1 and 99 (mL) then the program key was pressed again to set the amount.
   c. The code was loaded into the Arduino board. Selected a liquid (with A, B, C, or D) and hit the program button, inputted a volume, then pressed the program button again to set this volume. Pressed dispense and then measured the volume dispensed.
   d. The other four liquids were used with several different volume values.

4. OLED Screen Test:
   a. Codes that displayed the liquid and amount being dispensed had been written (the default value defined in the code before the user changes it). The user inputted an amount of liquid to be dispensed, and the correct number was displayed on the screen.
   b. The code was loaded into the Arduino again to run the selection and volume tests. The OLED screen displayed the letter and amount of the cleaner being dispensed.

Selection Nozzle/Full Device:
The selection nozzle was attached to the end of the hose where the liquid was dispensed. It was a 3-D printed part that the user held to dispense the liquid in the desired location. The rest of the device was functional at this point in the testing, so the selection nozzle tests involved the entire device. Once the nozzle was attached, it was tested with all four liquids. Each liquid was dispensed several times, then removed the nozzle to check for leaks. The nozzle appears to be working correctly and there was no evidence of leaks or mixing of liquids. The full device was ready for testing.

Full Device Test:
   a. All the device components were connected and loaded the written code.
   b. Performed the selection test, the default amount of each of the four liquids was dispensed when its corresponding button was pressed (followed by the dispense key).
   c. Performed the volume test, manually inputted a volume amount and dispensed it. Did this for all four liquids and various values of volume.
   d. Tested the OLED display was correct through each of the previous tests.
   e. The device was tested with invalid volume values and button sequences, returned to its previous state (before the invalid action), and did not crash or function abnormally.

Battery Test:
The device was battery-powered and connected to a standard AC outlet to charge, but it could be used while it was charging. The battery was tested once the device was fully functional while connected to power. After the battery was fully charged, the device was unplugged from the wall and dispensed liquid until the battery life ran out. The volume of liquid was then measured to ensure it fitted the requirement of 2500 mL per charge. This test was repeated several times to validate the results.

OUTPUT TEST	TANK A	TANK B	TANK C	TANK D
20 ml test1	21	22	22	21
20 ml test2	20	21	22	20
20 ml test3	20	21	20	20
Average	20.33333333	21.33333333	21.33333333	20.33333333
40 ml test1	41	42	41	40
40 ml test2	41	42	42	39
40 ml test3	40	41	42	40
Average	40.66666667	41.66666667	41.66666667	39.66666667
60 ml test1	59	60	64	60
60ml test2	60	60	62	59
60 ml test3	59	59	63	60
Average	59.33333333	59.66666667	63	59.66666667
80 ml test1	77	80	82	81
80ml test2	78	81	85	80
80ml test3	77	82	84	80
Average	77.33333333	81	83.66666667	80.33333333

Table 3: Liquid pump output test

As seen above, the project was tested on its requirement traceability matrix to output specific amounts based on the user's input. The output was to be produced accurately with +/-5ml of error. Based on the test the Cleaver Cleaner Container never fell outside of this marin. During this test the battery requirements were tested simultaneously. Over an hour of use and over 2500ml dispensed and the device still retained power. The pump's throughput of 100ml/min was found to be accurate. The device's dimensions after construction were measured to be 10in,Length 10in, and Width 6in and its weight only slightly over 10lbs. All other requirements were tested and found to be within a tolerable range or better.

Project Management

Work Breakdown Structure and Attributes

ID	Activity	Description	Duration [weeks]	Predecessor Activities	Resources: People	Resources: Components/Software Needed
0	**Formation of team and brainstorming**					**None**
0.1	Team selection	Team members are selected	2			
0.2	Team organization	Team members create guidlines and meeting times	2	0.1	Chengfeng Yu, Eric Killian, Bryce Goldman, Zun Yang, Daniel Welsh	
0.3	Team Brainstorming	Team members brainstorm and form ideas for the project	2	0.2	Chengfeng Yu, Eric Killian, Bryce Goldman, Zun Yang, Daniel Welsh	
0.4	Team Project decision and initial preperations	Team decided on a project and started looking into ideas and first steps	2	0.3	Chengfeng Yu, Eric Killian, Bryce Goldman, Zun Yang, Daniel Welsh	
1	**Selecting and Ordering Parts**					**Computer to find parts online**
1.1	Arduino (Compatible Boards/Motor Shields) Part	Using guidelines, chose a Arduino (Compatible Boards/Motor Shields) that fits required specifications.	2		Chengfeng Yu, Eric Killian, Bryce Goldman, Zun Yang, Daniel Welsh	
1.2	Select Motors, tubing, and dispenser containers	Configure the motors to run and make sure everything works as expected. Make sure that everything can be powered correctly and runs fine with no problems.	2	1.1	Chengfeng Yu, Eric Killian, Bryce Goldman, Zun Yang, Daniel Welsh	
1.3	Select wire, buttons, nozzle, other electronic parts needed	Configure the motors to run and make sure everything works as expected. Make sure that everything can be powered correctly and runs fine with no problems.	2	1.2	Chengfeng Yu, Eric Killian, Bryce Goldman, Zun Yang, Daniel Welsh	
1.4	Order parts	Order all parts online	1	1.1, 1.2, 1.3	Chengfeng Yu, Eric Killian, Bryce Goldman, Zun Yang, Daniel Welsh	
2	**Test Basic Functionality of Parts**					**Arduino board, Motors, Battery, Power Cable, Buttons, Wire, Arduino software**
2.1	Motor Functionality Testing	Configure the motors to run and make sure everything works as expected.	5	1.4	Zun Yang	
2.2	Power distribution Testing	Make sure that everything can be powered correctly and runs fine with no problems.	5	2.1	Zun Yang	
2.3	Basic code for motor testing, and button functionality	Write some basic code that runs the motors and tests to make sure button input can be detected	5	2.2	Zun Yang	
3	**Code Generation**					**Arduino board, Motors, Battery, Power Cable, Buttons, Wire, Arduino software**
3.1	Pseudocode (Software Plan)	Develop an algorithm and write pseudocode for the main program that will run the device.	1	2.3	Daniel Welsh, Chengfeng Yu, Eric Killian, Bryce Goldman	
3.2	Selection Code	Write code that dispenses various liquids for different selected buttons	2	3.1	Chengfeng Yu, Eric Killian, Bryce Goldman, Zun Yang	
3.3	Dispense Amount Code	Write code that allows the user to change the dispense amount for each liquid (amount at this point is hardcoded)	2	3.2	Chengfeng Yu, Eric Killian, Bryce Goldman, Zun Yang,	
3.3	Programmable Amount Code	Write code that allows the user to change the dispense amount for each liquid through the keypad interface	2	3.2	Chengfeng Yu, Eric Killian, Bryce Goldman, Zun Yang,	
3.4	LCD Screen coding	Write code to display amounts, and selection on an LCD screens	2	3.3	Chengfeng Yu, Eric Killian, Bryce Goldman, Zun Yang	
4	**Hardware interface and placement**					**All components for the device**
4.1	Container Design	Study the various parameters of the container and select several suitable models. Using guidelines, chose a container that fits required specifications.	2	2.3	Daniel Welsh, Bryce Goldman	
4.2	Project Box Design	Study the various parameters of the project box and select several suitable models. Using guidelines, chose a project box that fits required specifications.	2	2.3	Daniel Welsh, Bryce Goldman	
4.3	3D Selection Nozzle Design	Study the various parameters of the selection nozzle and select several suitable models. Using guidelines, make fits required specifications to fit OLED and Key pad.	3	2.3	Daniel Welsh, Bryce Goldman	
4.4	Input keyboard Design	Study the various parameters of the input keyboard and select several suitable models. Using guidelines, chose a input keyboard that fits required specifications.	2	2.3	Daniel Welsh, Bryce Goldman	
4.5	Tube Design	Study the various parameters of the tube and select several suitable models. Using guidelines, chose a tube that fits required specifications.	2	2.3	Daniel Welsh, Bryce Goldman	
4.6	OLED Screen	Study the various parameters of the OLED screen and select several suitable models. Using guidelines, chose a OLED screen that fits required specifications.	2	2.3	Daniel Welsh, Bryce Goldman	
4.7	Integration	Assemble the different components to prepare for the test.	5	1, 2, 3, 4	Daniel Welsh, Bryce Goldman	
5	**Testing**					**Arduino Code, Prototype build**
5.1	Test Code Function	Input the code into the Arduino board and test it.	6	3.4 and 4.6	Chengfeng Yu, Eric Killian, Bryce Goldman, Zun Yang, Daniel Welsh	
5.2	Complete Project Test	Test complete project under all possible circumstances. and modify if needed	6	5.1	Chengfeng Yu, Eric Killian, Bryce Goldman, Zun Yang, Daniel Welsh	

Table 4: Work Breakdown Structure

The group divided the tasks appropriately based on background and skill level. Coding was handled by Eric Killian and Zun Yang while construction and designing were handled by Daniel Welsh, Bryce Goldman, Yu Chengfeng. Testing was completed by everyone based on their familiarity with their part on the project.

Gantt Chart

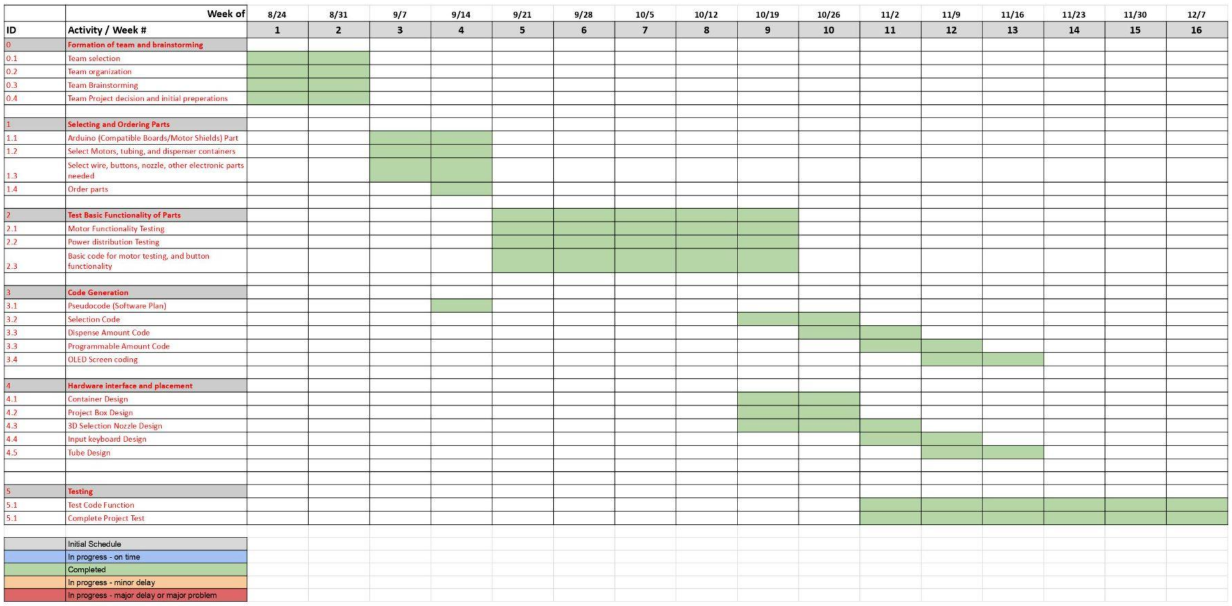

Table 5: Gantt Chart

Timeline

Task	Date
Initial parts are selected and ordered	9/14/21
All parts have arrived and none have obvious defects	9/21/21
Electronics connect together, initial prototype circuit is formed	9/21/21
Sample code written and tested on Arduino	9/21/21
First draft of code for the device is completed	11/2/21
Verify that the electronics are working as intended and no bugs are detected	11/25/21
First full prototype (includes mechanical components as well)	12/2/21
First full test run of the device	12/2/21
Device is finished mechanically and there are no leaks or missing parts	12/2/21
Final test run before presentation	12/6/21

Table 6: Timeline

The team was on or ahead of schedule nearly the entire duration of the project. The one time constraints that limited the group was when 3D parts were needing to be submitted for analysis and cost. As stated before this was an unforeseen issue since the previous printing company couldn't provide its services at the time. Another time constraint occurred when ordering parts. Amazon used to be the university’s easiest and quickest way to obtain parts, however, this semester the professor had to change how parts were ordered. A bill of materials was submitted to the professor, and after the professor had to send out the order to another person in the university to order them. Despite these issues the project's functionality and construction were barely affected.

Over this time the final design changed slightly. The Arduino Pump System had a container designed and 3d printed for it to better fit the parts and organize the components, although this drastically increased the cost. The Arduino Pump System was moved to the base of the device due to its weight and allowing easier access to the lid of the cleaner container. The code was changed to program 1-100ml to 1-99ml. This was done to simplify the code and alleviate the issues with memory on the arduino.

Required Hardware/Components/Equipment/Facilities

Required Hardware & Components:

1. PC
2. Solidworks
3. Arduino Uno
4. Wires & Connectors
5. Heat Shrink
6. 3D printer filament
7. Breadboard
8. Digital Keypad
9. Four 12V DC Pumps
10. Plastic Fluid Tubes
11. OLED Screen
12. Waterproof Box
13. Rechargeable Battery and Charging Cable
14. Chalkboard Stickers
15. Hose Tubing
16. Motor Shield

Required Equipment:

1. 3D printer
2. Wire cutters
3. Drill & bits

Required Facilities:

1. Capstone Laboratory (Dreese Laboratories)

Budget and Expenses

Item Name	Quantity	Cost per unit	Total cost for item	Sales Tax 5.75%	Budget Spent before tax	Budget Spent After tax
Arduino Uno REV3	1	$22.44	$22.44	$1.29	$631.69	$643.36
Arduino Motor Shield	1	$14.39	$14.39	$0.83		
12 volt rechargeable power supply	1	$38.39	$38.39	$2.21	Total Budget	Net Budget
Cleaners container	1	$27.99	$27.99	$1.61	$500	-$143.36
Black Foam Sheet Self Adhesive Rubbe	1	$15.99	$15.99	$0.92		
12v DC pump throughput 100ml/min	4	$7.45	$29.80	$1.71		
Arduino Keypad, 4x4	1	$5.88	$5.88	$0.34		
Silicone Tubing, 3mmIDx5mmOD, 16.4ft	1	$9.99	$9.99	$0.57		
120pcs Multicolored Dupont Wire	1	$6.98	$6.98	$0.40		
Clear Silicone Waterproof Sealant	1	$3.20	$3.20	$0.18		
OLED screen	1	$6.99	$6.99	$0.40		
Safety 1st Multi-Purpose Appliance Lock	2	$4.87	$9.74	$0.56		
Stretch Tube Tools Stress 6 pack	1	$5.99	$5.99	$0.34		
VELCRO	1	$3.78	$3.78	$0.22		
Chalkboard Labels	1	$1.34	$1.34	$0.08		
3D Printed Parts	1	$428.80	$428.80	$0		

Table 7: Final Project Budget

Risks

Technical Risks

Container:

- Difficult to move when full
  - Mitigation Strategy: Handles on the sides of the container for easy transport
- Leaking of liquids between the four tanks
  - Mitigation Strategy: Use a sealant and foam rubber to plug any holes between the tanks

Arduino Pump System:

- Electronics (pumps, Arduino board, motor shield) getting wet from the cleaning liquids inside the device
  - Mitigation Strategy: The plastic housing for the Arduino Pump System can protect the electronics from moisture. Further measures can also be taken to isolate these electronics from the liquids.

Selection Nozzle:

- Liquid dribbles out of the nozzle instead of spraying
  - Mitigation Strategy: Decrease the size of the hole in the nozzle to increase the pressure of the liquid coming out.
- Mixing of chemicals in the nozzle
  - Mitigation Strategy: Use four separate dispensers on the nozzle that do not touch each other.
- Liquid leaking where the hose meets the nozzle
  - Mitigation Strategy: Reinforce the connection between the hose and nozzle with a sealant or gasket.

Keypad Interface:

- Invalid or unexpected key sequences
  - Mitigation strategy: The code has error detection, resetting the system to the last known state prior to the invalid sequence.

The technical risks of this project go beyond the ones listed above. There is always a risk for something not working as expected, and it is very likely that several of these issues are encountered throughout the duration of the project. For broken components that are not able to be fixed, the entire team must be notified and a replacement must be ordered. The budget must be modified to account for this as well. For issues that appear fixable, the group must be consulted if the person that ran into the issue cannot figure out how to fix it himself. Technical issues that are fixed by one person should also be reported to the group and documented accordingly.

Management Risks

- A task is a lot more involved than the group previously realized, giving one person an unfair amount of work compared to the others
  - Mitigation Strategy: The division of work is modified such that this person's workload is reduced and everybody is doing roughly the same amount of work.
- A group member is not contributing their fair share
  - Mitigation Strategy: The group addresses the specific tasks that the lagging teammate has failed to accomplish.
- The duration of a task is longer than predicted on the Gantt chart
  - Mitigation Strategy: Extend the deadline of the task and adjust other tasks if needed.

Planning the entire project in advance is helpful but must also be flexible. It is very unlikely that everything will take exactly as long as expected and be finished on time. Many of the management risks involved with this project can be avoided if everybody on the team has the right mindset. Team members must not be resistant to changing the schedule or taking on more work if needed. Communication between team members must be effective in order to divide work in a way that is fair for everybody. There will almost certainly be disagreements and the project will be stressful at times. However, these management issues can be minimized if everyone on the team communicates well and is flexible with work.

Challenges, Issues, Problems and Discovered Solutions

The process of programming the Arduino went smoothly for the most part, with the exception of a conflict between the OLED display and the four motors. Individually, the OLED could function as expected without the motors connected to the circuit. Similarly, the motors worked as expected only when the OLED was not connected. After some unsuccessful troubleshooting into the communication interface of the Arduino, it was discovered that it did not have enough memory to properly execute the code. The code was modified to remove switch statements, which use up a significant amount of memory in Arduino. This resolved the issue, as the OLED and the motors could work simultaneously with the updated code.

The manufacturer of the container used to hold the liquids did not design it to be completely waterproof. Before making any modifications to the container, leaks between the container's four tanks as well as between the inside and outside of the container were observed. Initially, water sealant was applied to the top of the container and left to dry with the goal of creating a more pliable and watertight layer whenever the container was closed. After this failed and the issue of leaking between the tanks still requiring attention, rubber foam adhesive sheets were ordered. A sheet was attached to the underside of the container lid and cut appropriately to fit the shape of the container and the tanks. Then an additional foam sheet was used to cut four sections to be placed at the back of each of the tanks of the container. The layers of the rubber foam at both the top and sides of the container mitigated most of the leaking both between the tanks and outside of the container. The remaining leaks were small and able to be fixed with silicone water sealant.

Lessons Learned

It is nearly impossible to predict the future, especially in the case of engineering. Some things will take much longer than expected, and some things will be much easier than expected. Figuring out the algorithm and implementing it into code was expected to take weeks, but was done in a matter of hours. However, it took much longer than this to actually get the code to work on the device. The OLED display and the motors were not able to function correctly at the same time. This was an issue that could not be anticipated, and it took longer to fix than it did to write the code itself.

Spending too little money should not be a concern. For the bulk of the project, we were significantly under budget. Less than half of the budget had been spent before the 3-D printed parts were ordered. These parts cost over four hundred dollars, putting us slightly over budget for only a few additional parts. It is impossible to predict every part that is needed for an engineering design, and sometimes it will cost a lot less than initially expected. This is completely okay, as it is sometimes the case that parts will cost a lot more than expected.

Conclusion

Summary and Benefits

Overall, this project aims to simplify the hassle of cleaning making it effortless. The Clever Cleaning Container should dispense cleaning liquids instantaneously and save people time. Small cleaning jobs are done every day and time can be saved. The goal for this project is to do that while still meeting all the requirements the group outlined. The product needs to be a handheld device that can be carried around with an easy to use nozzle. The device needs to be easily programmed to dispense the correct amounts of liquid that the user wants on the push of a button, and the device needs to be most importantly safe. The group came up with many possible designs that would all meet these basic requirements. This included buttons placed in different arrangements, different cleaning compartment concepts, and various designs for the device. In the end the group chose a simple Arduino Board with a motor driver, with 4 DC pumps all connected to an output nozzle. The Arduino would be controlled from a 4x4 keypad that would be on the selection nozzle of the device. The group believes that this project is very realistic and it's easily possible to be done under budget leaving the group lots of room for flexibility. The group tried to be ahead of schedule on time management as well, planning ahead for possible struggles and complications with the project. The group agreed to put forth its highest effort to make the best project in the allocated time.

Future Enhancements

When working on the design of the Clever Cleaning Container, the group thought of many possible future enhancements that would make this device even better. This included integrating the device into a sink or shower nozzle for even more ease of use, adding a wireless charging dock station for the device, and even adding wireless communication to the device so that it could communicate with a smartphone application. Another idea the group came up with was adding the ability to put pictures of the cleaners being used on the OLED screen. In the end the group was not able to actually go through with these enhancements due to group constraints (Budget, Group Experience, Time, Equipment, etc.), the main constraint being time.

Next Step Recommendations

The group planned to improve the device by using another OLED screen that adapted better to the chosen Ardunio board. There was a memory conflict between the used OLED screen and the Arduino board caused by different memory formats. Another problem was that the chosen OLED screen only outputted two numbers when showing the output volume. A 3D printed container could replace the container to save the team's budget. The chosen container had a water leakage on the top lid, thus creating a problem while the user moved the device around in the house. Furthermore, the bottom box can be modified by adding a layer between motors and the Arduino board to prevent potential water leakage on tubes that connect motors and tubes.

Bibliography and References

[1] "How much time does the average person spend cleaning house?," Signature Maids, 20-Jun-2017. [Online]. Available: https://www.signaturemaids.com/our-blog/how-much-time-does-the-average-person-spend-cleaning-house/. [Accessed: 05-Sep-2021].

[2] "Survey results: Americans spend nearly six hours each week cleaning but wonder, is it enough?," *Survey Results: Americans Spend Nearly Six Hours Each Week Cleaning But Wonder, Is It Enough? | Business Wire*, 13-Nov-2018. [Online]. Available: https://www.businesswire.com/news/home/20181113006110/en/Survey-Results-Americans-Spend-Nearly-Six-Hours-Each-Week-Cleaning-But-Wonder-Is-It-Enough. [Accessed: 05-Sep-2021].

[3] "Fabuloso Lavender Multi Purpose Cleaner," *Kroger*. [Online]. Available: https://www.kroger.com/p/fabuloso-lavender-multi-purpose-cleaner/0003500053058?&cid=shp_adw_shopg_.kroger_g_pla_shop_acq_evgn_ship_low%2Bspend_cleaning%2Band%2Bhousehold_NEW0821&gclid=Cj0KCQjwm9yJBhDTARIsABKIcGZRabeDUo_ZznVq07hFmGFCvxQm0577QTD1Qf3UMJpiFHBpmfoySUoaAuoOEALw_wcB&gclsrc=aw.ds. [Accessed: 07-Sep-2021].

[4] "PVC NSF-51 hose: Hose and fittings source," *Hose and Fittings Source | Hydraulics & Pneumatics*, 30-Oct-2017. [Online]. Available: https://www.hoseandfittings.com/bah-pvc-nsf/. [Accessed: 26-Sep-2021].

Appendices

https://standards.ieee.org/standard/1019-2013.html
https://standards.globalspec.com/std/14354011/ansi-iec-60529
https://ieeexplore.ieee.org/document/4116784
https://www.hoseandfittings.com/bah-pvc-nsf/

Final Software Source Code on the ATmega32U4

```
/*
  Clever Cleaning Container (C^3)
  This code processes input from a 4x4 keypad matrix
  (Teyleten Robot 4 x 4 Matrix Array 16 Key Membrane Switch Keypad Keyboard for Arduino AVR PI C)

https://www.amazon.com/Teyleten-Robot-Membrane-Keyboard-Arduino/dp/B0993D2VLR/ref=sr_1_23?dchild=1&keywords=keypad+for+arduino&qid=1631044015&s=industrial&sr=1-23
  The code then processes the input to control 4 DC Pumps through a motor sheild
  (HiLetgo TB6612 Mosfet for Arduino Motor Shield Standard IIC I2C TB6612FNG Stepper Motor PCA9685 PWM Servo Driver Shield V2 Robot PWM Uno Mega R3 Replace L293D)

https://www.amazon.com/HiLetgo-TB6612FNG-Stepper-Motor-Replace/dp/B073SP76MC/ref=sr_1_3?dchild=1&keywords=ARDUINO+V2&qid=1630451092&s=industrial&sr=1-3
  Pinouts:
  * list the components attached to each input
  * list the components attached to each output
  Created 10/12/2021
  By Eric Killian
  Modified 10/12/2021
  Eric Killian
*/
#include <EEPROM.h>
#include <Keypad.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_MotorShield.h>

#define SCREEN_WIDTH        128
#define SCREEN_HEIGHT       64
#define OLED_RESET          4
#define SCREEN_ADDRESS      0x3C
#define MAX_DIGITS          3
#define SELECTED_MOTOR_ADDR  0
#define AMOUNTS_ADDR        2

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};
const byte row_pins[ROWS] = {2, 3, 4, 5}; // Pins used for the rows of the keypad
const byte col_pins[COLS] = {6, 7, 8, 9}; // Pins for the columns of the keypad
```

```
// Motor Sheild Initialization, uses default I2C address
Adafruit_MotorShield AFMS = Adafruit_MotorShield();
Adafruit_DCMotor *motors[4] = {AFMS.getMotor(1), AFMS.getMotor(2), AFMS.getMotor(3), AFMS.getMotor(4)};

// Keypad Initialization
Keypad keypadMatrix = Keypad(makeKeymap(keys), row_pins, col_pins, ROWS, COLS);

// motor selection variables
int selected_motor;

// program feature variables
bool program_mode = false;
char program_number[MAX_DIGITS+1];
int program_index = 0;

// time in ms for each motor
unsigned int amounts[4] = {100, 100, 100, 100};

#define LOGO_HEIGHT   40
#define LOGO_WIDTH    40
static const unsigned char PROGMEM c3_logo[] =
{ 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0xfc, 0x00,
  0x00, 0x00, 0x03, 0xfe, 0x00,
  0x00, 0x00, 0x07, 0xff, 0x00,
  0x00, 0x00, 0x0f, 0xff, 0x80,
  0x00, 0x00, 0x0f, 0x07, 0x80,
  0x00, 0x00, 0x0f, 0x07, 0x80,
  0x00, 0x00, 0x03, 0x07, 0x80,
  0x00, 0x00, 0x00, 0x07, 0x80,
  0x00, 0x00, 0x00, 0x7f, 0x00,
  0x00, 0x07, 0xe0, 0xfe, 0x00,
  0x00, 0x0f, 0xf8, 0xff, 0x00,
  0x00, 0x1f, 0xfc, 0x7f, 0x80,
  0x00, 0x3e, 0x7c, 0x07, 0xc0,
  0x00, 0x7c, 0x3e, 0x07, 0xc0,
  0x00, 0x78, 0x1e, 0x07, 0xc0,
  0x00, 0x78, 0x1f, 0x07, 0xc0,
  0x00, 0x78, 0x0f, 0x8f, 0xc0,
  0x00, 0x78, 0x0f, 0xff, 0x80,
  0x00, 0x78, 0x07, 0xff, 0x00,
  0x00, 0x78, 0x01, 0xfe, 0x00,
  0x00, 0x78, 0x00, 0x00, 0x00,
  0x00, 0x78, 0x00, 0x00, 0x00,
  0x00, 0x78, 0x1c, 0x00, 0x00,
  0x00, 0x7c, 0x3e, 0x00, 0x00,
  0x00, 0x7e, 0x7e, 0x00, 0x00,
  0x00, 0x3f, 0xfe, 0x00, 0x00,
  0x00, 0x1f, 0xfc, 0x00, 0x00,
  0x00, 0x0f, 0xf8, 0x00, 0x00,
  0x00, 0x07, 0xf0, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00,
```

```
  0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00 };

void setup() {
  Serial.begin(9600);

  // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
  if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }

  if (!AFMS.begin()) {        // create with the default frequency 1.6KHz
    Serial.println("Could not find Motor Shield. Check wiring.");
    while (1);
  }

  for (int i=0; i<4; i++) {
    motors[i]->run(FORWARD);
    motors[i]->setSpeed(0);
    amounts[i] = (int) readIntFromEEPROM(AMOUNTS_ADDR+i*2);
  }

  selected_motor = (int) EEPROM.read(SELECTED_MOTOR_ADDR);

  display.clearDisplay();
  display.drawBitmap(
    (display.width()  - LOGO_WIDTH ) / 2,
    (display.height() - LOGO_HEIGHT) / 2,
    c3_logo, LOGO_WIDTH, LOGO_HEIGHT, 1);
  display.display();
  delay(3000);


  print_selected_motor();
}

void loop() {
  // put your main code here, to run repeatedly:
  char button = keypadMatrix.getKey();

  if (button)
  {
    if (program_mode) {
      if (isdigit(button) && program_index < MAX_DIGITS) {
        program_number[program_index] = button;
        program_number[program_index+1] = '\0';
        program_index++;
      }
      print_program_num();
      if (button == '*') {
        program_mode = false;
        int amount = atoi(program_number);
        if (amount != 0) {
          amounts[selected_motor] = amount;
          writeIntIntoEEPROM(AMOUNTS_ADDR+selected_motor*2, amount);
        }

        print_selected_motor();
      }
```

```
    } else {
      if (button >= 'A' && button <= 'D') {
        selected_motor = ((int) button) - 65;
        EEPROM.write(SELECTED_MOTOR_ADDR, (byte) selected_motor);
        print_selected_motor();
      } else if (button == '#') {
        // runs the motors
        motors[selected_motor]->setSpeed(255);
        delay(600*amounts[selected_motor]);
        motors[selected_motor]->setSpeed(0);
      }
      else if (button == '*') {
        program_mode = true;
        program_index = 0;
        program_number[0] = '\0';
        program_number[1] = '\0';
        program_number[2] = '\0';
        program_number[3] = '\0';
        program_number[4] = '\0';
        print_program_num();
      }
    }
  }
}

void print_selected_motor() {
  display.clearDisplay();
  display.setTextColor(WHITE);
  display.setTextSize(2);
  display.setCursor(32, 0);
  display.print(amounts[selected_motor]);
  display.print("mL");
  display.setCursor(54, 24);
  display.setTextSize(4);
  display.print((char) (selected_motor+65));
  display.display();
}

void print_program_num() {
  display.clearDisplay();
  display.cp437(true);
  display.setTextSize(2);
  display.setCursor(0, 0);
  display.print("Program");
  display.setTextColor(WHITE);
  display.setCursor(0, 24);
  display.setTextSize(2);
  display.print((char) (selected_motor + 65));
  display.print(": ");
  display.print(program_number);
  display.print("mL");
  display.display();
}

void writeIntIntoEEPROM(int address, unsigned int number) {
  EEPROM.write(address, (byte)(number >> 8));
  EEPROM.write(address + 1, (byte)((number & 0xFF)));
}
```

```
unsigned int readIntFromEEPROM(int address) {
  unsigned int byte1 = EEPROM.read(address);
  unsigned int byte2 = EEPROM.read(address + 1);
  return ((byte1 << 8) + byte2);
}
```

Basic Keypad Print Code

```
const byte ROWS = 4;
const byte COLS = 4;

char keys[ROWS][COL] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
}

const byte row_pins[ROWS] = {1,2,3,4} //Pins used for the rows of the keypad
const byte col_pins[COLS] = {5,6,7,8} // Pins for the columns of the keypad

// Initialise the Keypad
Keypad keypadMatrix = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

void setup() {
  // put your setup code here, to run once:

  Serial.begin(9600);   // Initialise the serial monitor
}

void loop() {
  // put your main code here, to run repeatedly:
  char button = customKeypad.getKey();

  if (button) {
    Serial.println(button);
  }
}
```

Basic Motor Testing Code

```
/*
This is a test sketch for the Adafruit assembled Motor Shield for Arduino v2
It won't work with v1.x motor shields! Only for the v2's with built in PWM
control
For use with the Adafruit Motor Shield v2
---->  http://www.adafruit.com/products/1438
*/

#include <Adafruit_MotorShield.h>

// Create the motor shield object with the default I2C address
Adafruit_MotorShield AFMS = Adafruit_MotorShield();
// Or, create it with a different I2C address (say for stacking)
// Adafruit_MotorShield AFMS = Adafruit_MotorShield(0x61);

// Select which 'port' M1, M2, M3 or M4. In this case, M1
Adafruit_DCMotor *myMotor = AFMS.getMotor(1);
// You can also make another motor on port M2
//Adafruit_DCMotor *myOtherMotor = AFMS.getMotor(2);

void setup() {
  Serial.begin(9600);           // set up Serial library at 9600 bps
  Serial.println("Adafruit Motorshield v2 - DC Motor test!");

  if (!AFMS.begin()) {         // create with the default frequency 1.6KHz
  // if (!AFMS.begin(1000)) {  // OR with a different frequency, say 1KHz
    Serial.println("Could not find Motor Shield. Check wiring.");
    while (1);
  }
  Serial.println("Motor Shield found.");

  // Set the speed to start, from 0 (off) to 255 (max speed)
  myMotor->setSpeed(150);
  myMotor->run(FORWARD);
  // turn on motor
  myMotor->run(RELEASE);
}

void loop() {
  uint8_t i;
  myMotor->run(FORWARD);
  myMotor->setSpeed(255);
  delay(1000);
  myMotor->run(RELEASE);
  delay(1000);
}
```