ECE2021

Milestone III

December 5th, 2023
UNB Fredericton

Class Code: ECE2021
Document: Milestone III

Student Names: Samuel Woytiuk #3358161

Will Ross #3734692

Alexander Cameron #3680202

Benedict MacGillivray #3735161

Due Date: December 5th, 2023

Table of Contents

1.0 Background

Hydration is an essential part of everyday life. Staying hydrated has many benefits, including an increase in energy and improved brain performance, among many other health benefits. Due to these factors, it is especially important to stay hydrated when studying or working. A problem that many individuals may face is having to disrupt a work session to refill water. This has led to a discussion about implementing a solution so that efficiency and productivity may be improved.

One solution is an automatic water bottle filler. This device would determine when a water bottle is empty by measuring the weight of the contents and would then automatically refill it, as necessary. It would stand stationary on a desk, allowing users to remain in a workspace, and focus on a task without being disrupted. This would save a lot of time for the user and allow for more coherent and productive work sessions. It would lead to an overall increase in performance, and simply equip the user with a convenient device that may prove to be useful in other scenarios as well.

2.0 Customer Requirements

CR-N# (customer requirement need)
CR-W# (customer requirement want)

CR-N1: Detect a valid water bottle that functions with the device.
CR-N2: Constantly detect the weight of the water bottle while the bottle is present, and the device is powered on.
CR-N3: Automatically detect the water level of the bottle.
CR-N4: Refill the water bottle when an empty or partially empty water bottle is detected.
CR-N5: The source pumps water directly into the detected bottle.
CR-N6: Allow the user to manually refill the bottle.
CR-N7: The device will use micro-bit LEDs to indicate if the bottle has been correctly placed on the enclosure.
CR-N8: Display the water level.
CR-N9: Utilize LED lights to indicate when the bottle is done filling.

CR-W1: Temperature control.
CR-W2: Refill the bottle with filtered water.
CR-W3.1: The device will use a speaker to give an auditory confirmation that the bottle is filling up.
CR-W3.2: The device will use a speaker to give auditory confirmation that the bottle is finished filling up.
CR-W4.1: A red light notification will be displayed when filling starts.
CR-W4.2: The red light will change to blue once the water bottle is full, showing that it is full and playing the "power down" sound.
CR-W5: The device will also be connected to a web application to allow the bottle to be filled remotely.
CR-W6: The device will display the approximate temperature.
CR-W6.1: Indication when the water source is low.
CR-W6.2: Indication when the water source is empty.

3.0 Inputs/Outputs Conceptualization

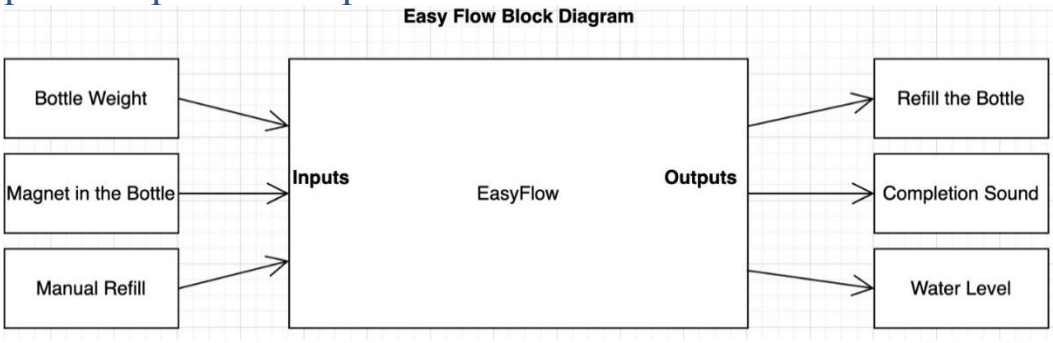

Figure 1: Output Block Diagram

3.1 Inputs:

Bottle Weight: The pressure will measure the weight of a specific cup to calculate if it needs to be refilled or not.

Magnet in the Bottle: The cup will have a magnet in the bottom for the UNB dev board to ensure that the correct bottle is being used.

Manual Refill: The device will use one of the buttons on the UNB dev boards if the user wants to manually fill the cup till full.

3.2 Outputs:

Refill the Bottle: The main output of the device will be water from an 18.9L water jug that is used in water coolers dispensed through water tubing into the specialized cup.

Water Level: The device will use the UNB dev board bar graph to display the current approximate water level, with the dev board bar graph having 5 lights the graph will approximate the levels in terms of fifths, and the light will blink if it is below the threshold and solid if it is above. The colour of the bar graph will be red if the water is below the set threshold and blue if it is above.

Completion Sound: The device will use a speaker to give an auditory confirmation that the cup is finished filling up. If the device uses the dev board's built-in speakers, it will play a series of sounds. If the speaker is external, then it will use preprogrammed phrases.

4.0 User Interface

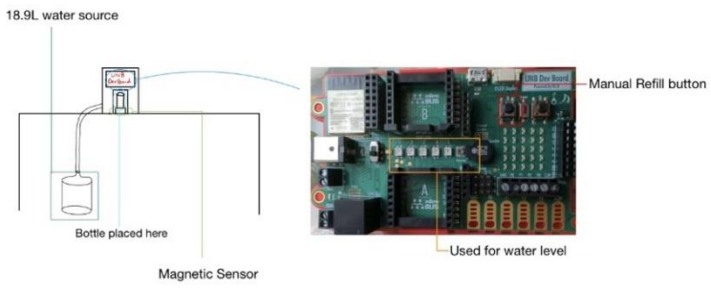

Figure 2: Easy Flow User Interface

The user interface will use the UNB Dev-board to take in and display information. The user will place the bottle 1L in the stand. It will use the 5 LED to show a rough estimate of the water level in fifths with the lights turning red when below half and blue when above. The light will be solid for the lights when it reaches the level and will blink in-between. It will use the A button on the dev board if the user wants to manually refill the bottle.

5.0 User manual

The Easy Flow is made to fill your water bottle to a certain level automatically and stop when it is full. It correctly determines the weight of the bottle and the water inside by using a pressure plate and a magnetic sensor. To make things simpler for the user, the pad includes a water pump, blue and red indicator lights, and alarms that can be heard.

Safety measures:

- This device is intended for indoor use only.
- Never interfere with the system's electrical components.
- Children should use this device under adult supervision.
- While the water pump is active, do not remove.

Installation Guide:

1. Set the pressure plate down on a solid, level surface.
2. Power on the pad
3. Connect a power source to the Easy Flow.
4. Insert the water tubing into a jug of water.
5. The device will start a self-test and a startup sound. The red light will indicate readiness if the water bottle is empty.
6. Now the easy flow is ready for use, enjoy.

Operation of Easy Flow:

Your water bottle should be placed on the pressure plate. It will be picked up by the magnetic sensor and pressure plate, the red light will turn on to show that it is prepared for filling. The bottle will start to fill thanks to the device. As soon as the bottle starts to fill, you will hear a "power up" sound, and the red light will stay on. The red light will change to blue once the water bottle is full, indicating that it is full and playing the "power down" sound. For more mobility of the easy flow, you can download our app. With the easy flow application, current water level, and manually fill the water bottle.

Maintenance and Troubleshooting:

- Keep the pressure plate clean and free of debris.

Specifications:

- Indicator Lights: Red (Starting fill), Blue (Finished filling)
- Sound Alerts: When filling starts, When full
- Magnetic Sensor: Detects presence of water bottle

6.0 Deliverables

You talk about how the product works and what it will comprise of, but the purpose of this section is to discuss what will be provided to the clients such as any schematics, drawings, prototypes, etc.
Based upon initial interaction with client group and completing a discussion of their product wants and needs, the agreed upon deliverables that our group will meet are as follows:

The product will be a system with the primary purpose of ensuring the client's 1-liter water bottle is maintained at an optimal content level and providing visual notifications. The product will be comprised of a 1L water bottle with a magnetic detection system. The docking unit will incorporate weight detection and the magnetic bottle monitoring system. The docking system will also contain LED indicators to advise the client that the bottle is correctly in place and detected and indicators for whether it is empty, low, or full.

When the docking system detects that the water bottle is correctly docked and below the required content weight, it will automatically commence the flow of water from the reservoir to refill the bottle. Once a specific weight of the contents of the bottle is met the system will stop the flow of water.

7.0 Engineering Requirements Data and Justification/Sources

Considerations for Water Level Detection and Refilling Water Bottle at Pre-determined Levels:

The following research was done to ensure that our customer requirements and wants were met regarding refilling and water level included the following: Calculations were completed to determine the necessary electric pump Wattage to overcome a 900mm vertical rise from the water source placed below the desk. The calculations can be seen in figures [figure 9, citation 28] and [figure 10, citation 28]. The figures were derived based on the assumption of overcoming hydrostatic pressure, distance, and frictional losses in 0.5" diameter vinyl PVC tubing. These calculations were made to match a flow rate of 1.0 gallons per minute which are standard comparable bottle refilling station flow rates, [2] like those found at UNB as well as standard sink "filtered water systems" [6]-[7]

The device will use pressure sensors and known weight values of the unfilled and filled bottle to determine the amount of water remaining in the water bottle. A switch will be triggered above a pre-determined value that will allow for a measurement of the water within the bottle, or another event to take place. [30]

Considerations for Auditory and Visual Notifications:

Visual:

To ensure that users will be able to view the visual LED notifications, several factors were considered. The optimal viewing angle when sitting at a desk is determined to be roughly 60-70 degrees below the optimal eye level as determined by research conducted by the CCOHS [15] conducted to determine the best "desk viewing angle". The device will be embedded with LEDs notifying the various states. Most standard LEDs will provide sufficient brightness for viewing in daytime home and office settings. See figures [figure 6, citation 26] and [figure 7, citation 11] which provide examples of LED brightness at various resistances [figure 8, citation 11] along with their physical footprints.

Auditory:

The following research will allow us to determine several factors outlined in our customer's needs and wants about auditory notifications. The UNB Development board is equipped with a speaker that is able to produce noise a volumes of up to 90Db @ 3V at a distance of 10cm. Based on a standard desk set up with a depth of roughly 600mm, based on calculations our device will be able to provide sufficient auditory notifications at this distance[figure 5, citation 28] at standard noise levels for home and/or office use.[21]. Research from "How to design a pleasant notification sound" [22] will allow us to supply useful but nonintrusive sound notifications.

Considerations for Temperature Control:

The standard R134a refrigerant should be used to regulate the temperature of the water. [2][31]. This refrigerant is commonly used in smaller temperature-regulating units such as air-conditioning units in automobiles. A capillary tube consisting of copper piping, with a diameter of 1.5mm, will be used to control the flow of the refrigerant. [32].

Considerations for Manual Usage:

- Power switch – Device will be equipped with a manual toggle to turn on/off the device.
- Manual fill button.
- Safety – Prevent overflow and spillage.
- Feedback – inform user when it is complete.

8.0 Engineering Requirements

Detection System:

- **ER-N1:** The device must detect that the water bottle fits the size requirements of a diameter of 89mm.
- **ER-N2:** The device must constantly use a weight sensor to detect the weight of the water bottle when it is stationary on the device.
- **ER-N3:** The device must be able to automatically detect the water level of the water bottle using the weight of the water as well as the weight of the bottle to a preassigned value.

Environmental System:

- **ER-N4:** The device must be able to refill the water bottle when it detects that there is less than 500ml of water left in the water bottle.
- **ER-N5:** The water source must pump water directly into the bottle at a flow rate of 1.1 GPM.
- **ER-N6:** The device must allow the user to manually refill the water bottle.

Interface System:

- **ER-N7:** The device must use micro-bit LEDs to indicate if the bottle has been correctly placed on the magnetic sensor.
- **ER-N8:** The water bottle must be able to display the water level of the water bottle accurately using the weight of the water bottle subtracted from the water mass to calculate the water level.
- **ER-N9:** The LEDs, to signal if the water bottle is finished filling, will be displayed at a brightness of 50lm and will be blue.

9.0 System Block Diagram

9.1.0 Detailed Design

9.1.1.0 Hardware

9.1.1.1 Block Diagram

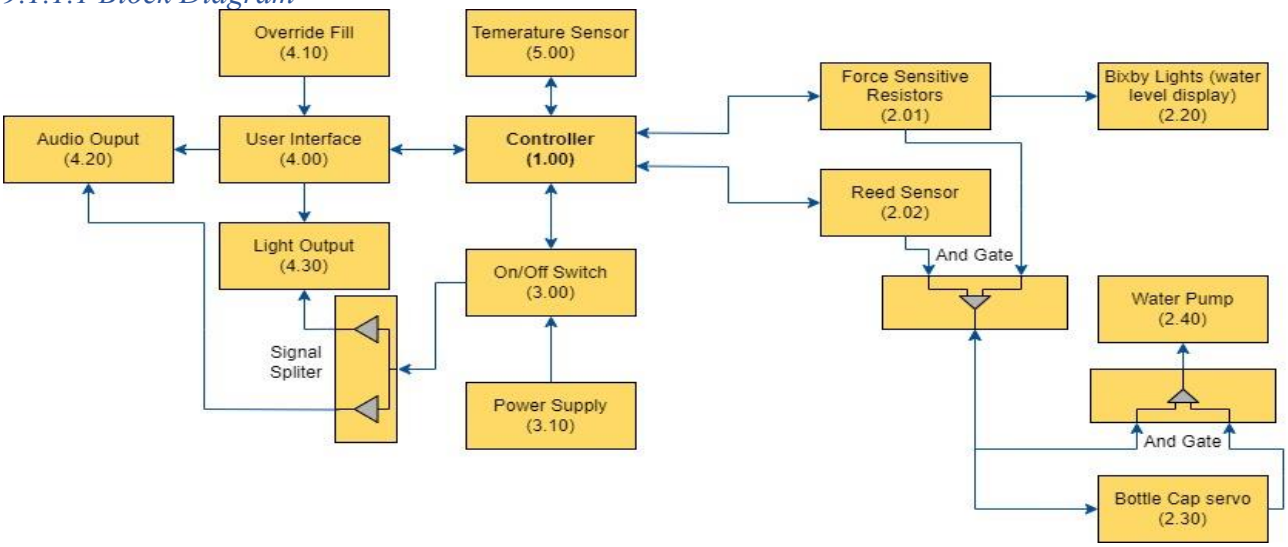

Figure 3: Original System Block Diagram

The original block diagram includes several features that were successfully used in the final product, while a few were ultimately removed in collaboration with client input.

Audio Output, Light Output, Override fill, Controller (UNB Devboard), Force Sensitive Resistors, Bixby Lights, Reed Sensor, and Water Pump were successfully implemented.

During the design process we encountered and resolved the following issues:

- The requirements in lighting output included testing single-color LEDs before finding and testing multiple RGB LEDs. This allowed for minimizing the total number of components while meeting client requirements. During testing of the original RGB LED it was determined that it was difficult to maintain a consistent color. Implementing a pre-built RGB LED with pins, provided consistent results.
- Issues arose while testing the Force Sensitive Resistor (FSR). Following the recommended resistor values provided by the FSR datasheet[33]. During testing, the Dev Board had difficulty maintaining consistent readings based on the position of the FSR and contact with the bottle. Eventually, after securing the FSR and providing a slight increase in height over the remainder of the resting surface, the FSR began providing more consistent results.
- The reed sensor implementation worked as intended during individual testing, but a faulty mikroBUS port was detected during the final construction stages. This required pivoting on implementation while still meeting client requirements.

After ongoing discussions with the clients, several “wants” were removed from the original product design due to practical limitations as well as available supplies. The temperature sensor was removed after discussing the difficulties in successfully implementing a temperature control system that, would work safely consistently. The Bottle Cap Servo was removed as we received one water pump motor and confirmed with the clients that using a more functional bottle and cap was acceptable.

9.1.1.2 Schematic

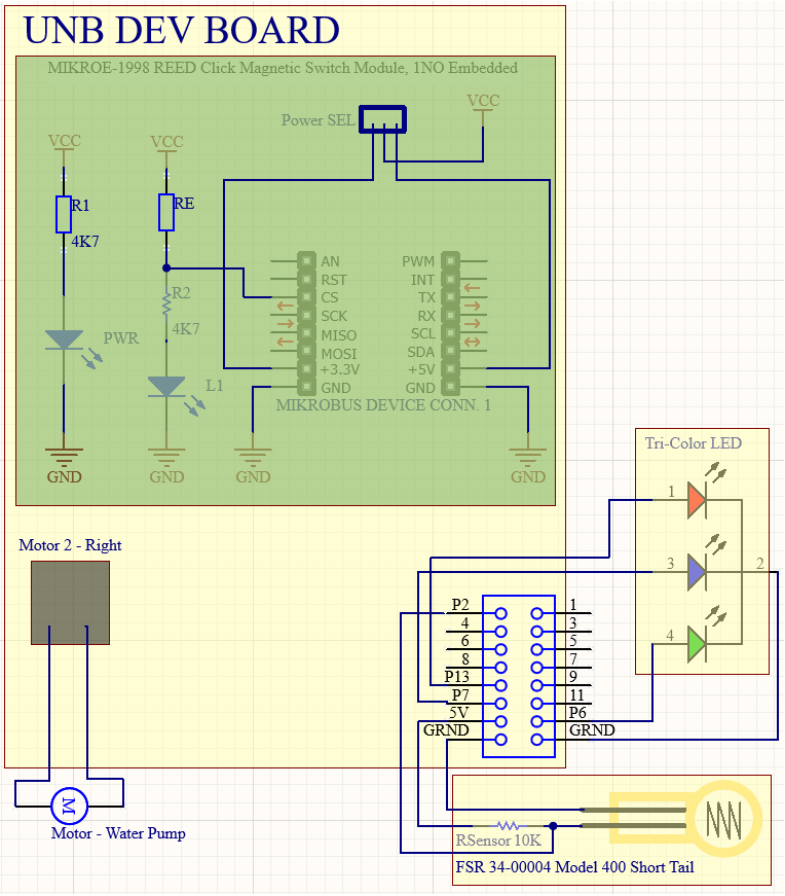

Figure 4: Altium diagram

Figure four represents the final Altium model implemented outside of the components included on the Dev Board. The figure highlights the mikroBUS-1998 REED sensor attached to one of the two provided mikroBUS ports, RGB Tri-Color LED connected to PINs P13, P6, P7, and ground, the submersible water pump attached to Motor two as well as the Interlink Electronics Force Sensitive Resistor Model-400 connected in parallel to the 5V source, Analog Pin P2 and ground. The omitted components including the speaker, Bixby lights, power input, and on/off switch are assumed to be on the yellow UNB Dev Board block.

9.1.2.0 Software

9.1.2.1 Flow Diagram

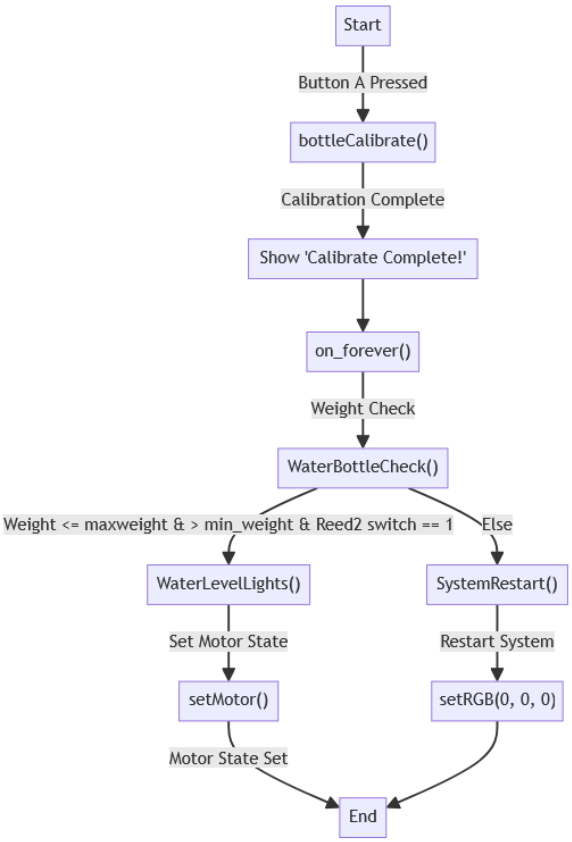

Figure 5: Flow Diagram

This flow diagram breaks down each section of the block design and shows its software counterpart. On start-up the user will recalibrate the system. The system will constantly be checking for a water bottle and either starting up to fill the bottle or do nothing.

9.2.0 Implementation

9.2.1 Hardware

Block 1.0 *Controller (Dev Board)*

During the integrated hardware implementation phase, the mikroBUS A port was sending a value to any mikroBUS connector and signalling that it was on without a magnetic signal. There needed to be a reed sensor to pick up a magnetic value from our cup, or else it would not be feasible. To fix that issue, the mikroBUS A port was used, and the reed sensor worked properly. Due to the magnetic sensor being further away from the cup, another issue arose, and the approach was changed. A magnet was used as a key to turn on and off the signal and allow the water bottle to be filled.

Block 2.01 *Force Sensitive Resistor*

When installing the force sensitive resistors (FSR), the biggest issue was getting the FSR to a good range that would work well with a water bottle being filled. To Accomplish this the FSR was set up in a voltage divider to get the values being sent out of the FSR to be sent to the circuit board. To do this a 10K ohm resistor was put in parallel with the FSR to get our value. Once done all the values are sent to the

circuit board by using the Dev Boards built in analog read using P2 and the function used to check the resistor value every 100 millisecond to ensure that the mass values were accurate.

Block 2.02 *Reed Sensor*

The reed sensor was used to detect if the proper water bottle was on the pressure plate to make sure it was not just another object with a similar weight. When implementing the reed senor, it was connected to the mikroBUS connector. When testing, the reed sensor was always on in mikroBUS port A. Once it was swapped to mikroBUS port B, the issue was solved. Next was testing the range of the sensor, which depended on the strength of the magnet.

Block 2.20 *Bixby Lights*

The Bixby lights were used to visualize the mass of the cup. One light represents an empty cup, and five lights represent a full one. As mass is added, the lights will go up depending on the set weight and the current volume of water.

Block 2.40 *Water Pump*

The water pump was connected to the right motor mount on the UNB Dev Board that will be turned on or off depending on mass and reed sensor values. It was set to 80% speed, and the tests showed that if the motor is set to 50% it is not powerful enough to go against gravity, and at 100% it was too fast.

Block 3.00 *On/ Off switch*

The on and off switch was built into the UNB Dev board. The dev board also would start any functionality unless both the reed sensor was true and the weight on the FRS was above or equal to the min weight and below the max weight.

Block 3.10 *Power Supply*

The power supply to provide all electronic components with power comes from a wall outlet that connects to the Dev Board through a barrel inlet.

Block 4.00 *User Interface*

The user interface was the UNB Dev Board showing Bixby lights for water level, A button for recalibration, the 5 x 5 light screen for seeing the current mass, and the built-in speaker for audio for powering up and powering down the system.

Block 4.10 *Override fill*

On the UNB dev board, button B can be pressed which will check and make sure that all requirements have been met except for the reed sensor and will solely rely on the mass of the machine while holding the B Button.

Block 4.20 *Audio Output*

Once the system starts up and the pump turns on it will play Brilliant Labs “Power Up” sound and when the system is complete it will play Brilliant Labs “Power Down” sound.

Block 4.30 *Light output*

Once the system starts up and the pump turns on the RGB light will turn red signalling not to move the cup and when the system is complete the RGB will display blue signalling that the fill is complete.

Originally, it was going to use three resistors at a value of 330 ohms but then there was a pre-built RGB light with the inputs Ground, Red, Green, and Blue.

9.2.2 Software

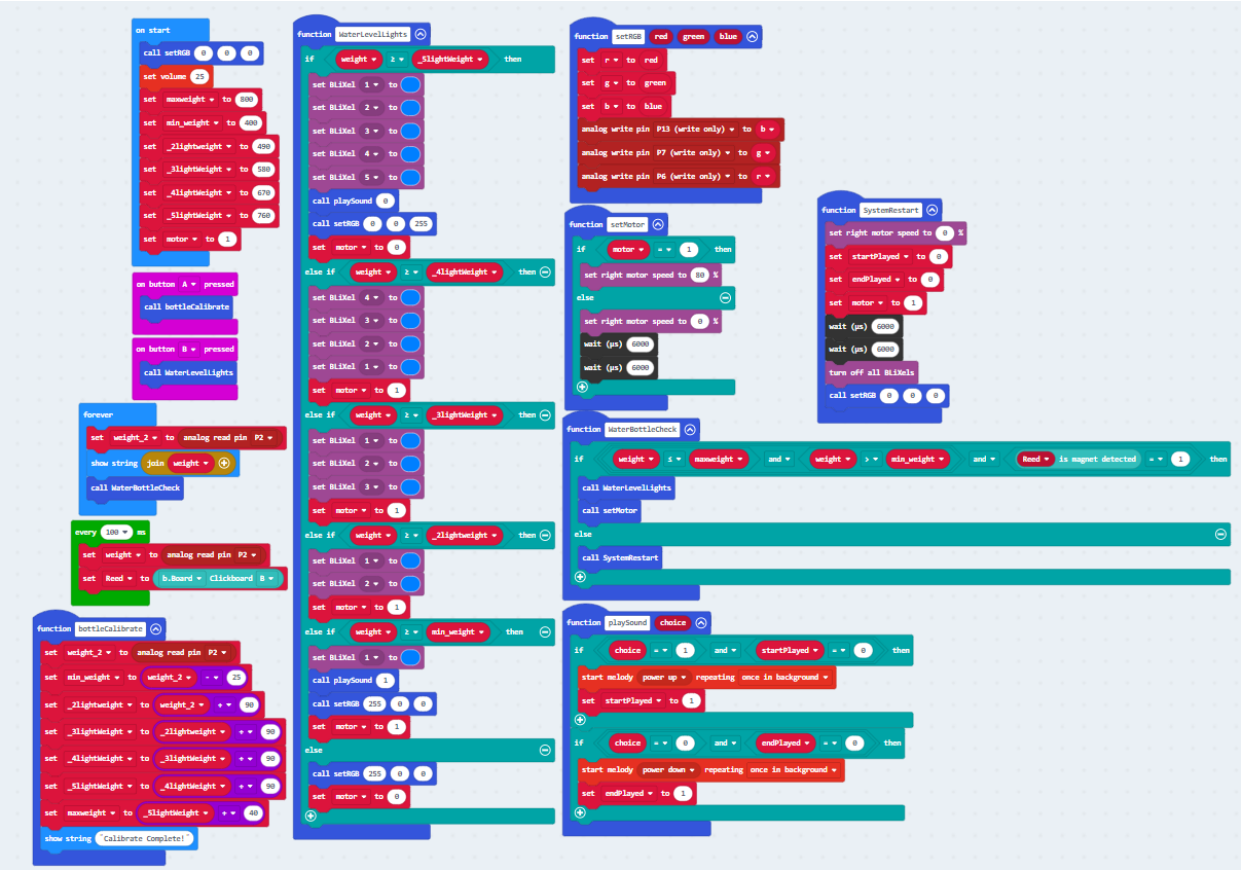

Figure 6: Block Code

Block 2.01 *Force Sensitive Resistor*

For the FSR, the variable “weight” was set equal to analog read pin P2. This is inside a while loop that loops every 100 milliseconds. It is used to read the measurements from the cup and get the added weight of the water.

Block 2.02 *Reed Sensor*

For the reed sensor, the variable Reed is set it equal to the dev board mikroBUS port B. This is inside a while loop that loops every 100 milliseconds.

Block 2.20 *Bixby Lights*

Our Bixby lights were shown to represent the water level in the cup. For software integration, the waterLevelLight() function checks the weight of the cup while it is being filled and will update the lights as the mass increases.

Block 2.40 *Water Pump*

To account for the water pump, a function was created to make sure the motor is responsive. SetMotor() will check if the variable motor is set to 1 or 0, 1 being true and 0 being false. If the motor equals 1 then the pumps are set to 80% speed. If not, the pump speed is set to 0. The motor variable gets its values from the waterLevelLights() function. The motor will be one unless the water is near or at the maximum mass.

Block 3.00 *On/ Off switch*

The waterBottleCheck() function checks if the mass is between the maximum mass and minimum mass. If it is between the two masses and the reed sensor is true it will turn on, and they will run through the waterLightLevel() function and then the setMotor() function.

Block 4.00 *User Interface*

The user interface was the UNB Dev Board showing Bixby lights for water level. The A button was used for recalibration, the 5 x 5 light screen for seeing the current mass, and the built-in speaker audio for power-up and power-down sounds.

Block 4.10 *Override fill*

Pressing the B button will override the waterBottleCheck(), meaning the system will only depend on the mass level and bypass the reed sensor.

Block 4.20 *Audio Output*

When the mass and reed sensor have been met inside waterLevelLights() it will call the playSound(choice) function. When choice = 1 the system will play Brilliant Labs “power up” melody, and when choice = 0 it will play Brilliant Labs “power down” melody. Just so the system would not play it every 100 milliseconds when the sound is active, it sets either startPlayed or endPlayed to 1. This was to ensure that the sound would only be played once.

Block 4.30 *Light output*

At the start and throughout the process from the water bottle being filled RGB light changes colour. To implement this, a function setRGB(red, green, blue) was created that takes 3 numerical values depending on the RGB value wanted. Inside this function, the three values are written to their designated pins on the circuit board.

11.3.0 Debug and Verify

The biggest part to debug was the pressure sensor as it took a while to find the value of the correct resistor to provide a proper sensitivity from the FSR. By testing multiple resistors and values the results indicated there was no linear pattern between 1k ohms to 10M ohms; the 10k ohm resistors were shown to provide the best sensitivity for the water bottle. The pressure sensor was very inconsistent for tests, it would not give the same mass reading for multiple tests in a row. By implementing a calibration button, the system can set mass values depending on the empty cup reading.

The magnetic sensor would only work with one of the mikroBUS ports. The mikroBUS A port would keep the reed sensor to a steady value reading, meaning there was no actual detection if there was a water bottle or not. The current prototype casing of the dev board could only fit in a certain orientation due to the location of the ports. This meant that the sensor could not detect the magnet on the water bottle. In

this prototype, it used a key on the front side to regulate if the system was allowed to start up, using a detached magnet and holding it to the side. If the mikroBUS port had been detected earlier, the casing could have had a new case layout to properly hold the Dev Board.

The water pump and the LED were not hard to implement, they just required more research. For the water pump, soldering more wire on and then heat shrinking was necessary to make it watertight. The motor tests provided a value to see what power level the pump should run at. The pump running at 80% was the most efficient, compared to 50% where the pump was not strong enough against gravity. The pump at anything over 80% was able to push it straight up through the tubing. For the RGB light, researching breadboards and power equations was needed. Once the circuit was complete, connecting the RGB light to the board was done using the analog pins.

The last step was to put all the components together. This provided challenges to the safety of the system components. The system casing had an opening in it to allow access to the dev board. This could have caused issues if the pressure sensor malfunctioned and caused an overflow. To cover up the dev board and still be able to see, a scored piece of plexiglass was acquired from the Engineering Maker Space. The tubing had an issue with the initial force of the pump; making it loose and shifting when water first entered. Designing a water delivery support structure to secure the tubing stopped the initial force of the pump.

10.0 System Integration and Testing

During the integration and testing phase, all components were assembled, and the project was thoroughly tested. The integration involved creating an enclosure using Fusion 360, which was fabricated with Maker Studios 3D printers. This enclosure served to house all electronic components as well as the water bottle.

The electrical setup included a UNB development board, wiring, a thin pressure sensor, a 10k ohm resistor, a breadboard, an alligator clip, a reed sensor, and a water pump. A significant challenge encountered was calibrating the pressure sensor to accurately read mass values with appropriate sensitivity. This was addressed by researching and implementing voltage dividers with pressure sensors, arranged in parallel. This configuration yielded a range of values, enabling the development board to accurately interpret the data. Various resistor values were evaluated; notably, a combination of a 2-million-ohm resistor and a 10k ohm resistor provided optimal results. However, intermediate values like a 500k ohm resistor resulted in inconsistent sensitivity and unreliable readings. Ultimately, the 10k ohm resistor in parallel with the pressure sensor was selected for its reliable performance and datasheet recommendation.

Incorporating the pressure sensor into the CAD design required strategic placement to ensure consistent readings. To maintain the sensor's position and enhance sensitivity, a small piece of paper was affixed atop the sensor, which was then securely taped to the base. The slight increase in height provided by the paper above the bottle resting plate allowed the sensor to detect the mass more accurately and consistently. The pressure sensor was programmed to scan for mass values every 100 milliseconds, enhancing accuracy and serving as a safety feature to prevent water spillage on the electronics.

An issue with the reed sensor was also addressed. MikroBUS port A on the development board was found to be faulty, causing continuous false readings. The sensor was subsequently moved to the mikroBUS B port, where it functioned correctly, using a magnet as a key to initiate the operation.

To accommodate different water bottles, fill levels, and placement variations, a function was implemented to recalibrate the bottle by pressing the A button. This recalibration adjusts the mass levels in six increments, adding 70 units to each level, and ensuring accurate filling without overflow.

The integration process also involved setting up a visual indicator using built-in Bixby lights on the development board. These lights indicate the water level, with one light representing the minimum weight and five lights indicating the maximum weight or completion of filling. Throughout filling the lights would illuminate as the water level rose. Audible chimes were programmed to signal the start and end of the filling process.

A significant challenge with the water pump was its erratic behavior during testing, where it failed to turn off upon receiving a signal. This was resolved in the software by establishing five distinct mass levels corresponding to the Bixby lights. The pump's operation was linked to these mass levels, ensuring it turned off at the appropriate weight or in the absence of a bottle.

A restart function was also developed. This function is activated once the desired water level is reached, as set by either the calibration or a predefined level. It resets the sound, lights, and pump settings, preparing the system for subsequent use.

The final integration of various components was facilitated by the user-friendly design of the UNB development board. Different pin inputs and outputs were utilized to connect all electronic components. The RGB light was controlled using ground and three pin outputs, while the pressure sensor and motor were connected using the analog pins. After modifications to the code and several tests to ensure functionality, the system was confirmed to operate as intended, in line with the established test cases and verification plan.

11.0 Completed Verification Plan

Engineering Requirements	Item Being Verified	Equipment Required	How to Verify	Expected Result	Actual Result	Pass/ Fail
ER-N1 Weight Sensor Test	Weight Sensor	-Bottle - Pressure Sensor -UNB Dev Board	Have a light connected to the sensor and place the bottle at varying fill levels	Light will only turn on when the weight is below the set threshold	Weight Sensor correctly measured when to turn light on or not	Pass
ER-N4 Start Weight Not Zero Measurement Test	Weight Sensor	-Bottle - Pressure Sensor -UNB Dev Board	Place the bottle with distinct levels of water	The bottle will only fill if below 500mL	The bottle only filled when below half of the weight of the water	Pass
ER-N9 Max Weight Measurement Test	Bottle filled LED	-Bottle -UNB Dev Board -Pressure Sensor -Magnetic Sensor	Have the bottle fill up	Light should turn on when bottle is full.	Light changed to the correct colour when full	Pass
ER-N6 Manual Button Fill Test	Manual Refill Button	-Bottle -UNB Dev Board	Have the button connected to manually refill the bottle and the bottle placed under the spout	When the button is pressed the Easy flow fills up the bottle no matter what water level it is and stops before it overflows	When button B was pressed and held water would flow bypassing Reed sensor requirement	Pass
ER-N3 Bottle Placement Test	Multiple weight sensors	-Bottle -UNB Dev Board	Place the bottle in different	The LED should turn on when the bottle is placed in the correct spot	Could not implement	Fail

		-Weight Sensor -Magnetic Sensor		activating both pressure sensor		
ER-N8 Water Display Test	Water level Display	-Bottle -UNB Dev Board -Weight Sensor -Magnetic Sensor	Place the bottle with various levels of water	Easy Flow correctly displays the approximate water level	Due to the sensor the water level can only accurately be displayed if it was calibrated in the same testing session.	Pass
ER-N5 Correct Water Pressure Test	Water Pressure	-Water Pump -UNB Dev Board	Have the water source low and the tubing connected to the vertical and have the pump pulling water up	The pump pulls the water through the tubing straight up	Water was able to pump through the tube into the cup	Pass
ER-N7 Magnetic Sensor Test	Magnetic Sensor	-Bottle -Magnetic Sensor -UNB Dev Board	Have a light connected to the sensor if it detects the bottle it will turn.	Light to be on only when the bottle is placed in the correct spot	Light turned on when magnet was in range	Pass
ER-N2 Continuous Weight Measurement Test	Constantly Detect Water Level	-Bottle -UNB Dev Board	Place an empty bottle and wait for it to fill	Should stop filling when the bottle reaches the max weight	Water stop filling when at desired weight	Pass

Appendix A - References

[1] Elkay, "Elkay In Wall bottle filling station," [Online]. Available: https://www.elkayfiles.com/spec-sheets/lvrc8wsk_spec.pdf.
[2] Elkay, "Bottle filler flow rate specification," [Online]. Available: https://pages.bottlefillingstations.com/wp-content/Product%20PDFs/Green/Elkay%20LZSTLG8WSLK_Spec%20Sheet.pdf.
[3] WaterAnywhere, "Flow rates for copper piping," [Online]. Available: https://wateranywhere.com/flow-rate-and-sizing-guide/.
[4] Elkay, "Elkay product list," [Online]. Available: https://www.elkay.com/products/category/drinking-water/fillers/on-wall.
[5] R. Allain, "How many water bottles could a filling station save?," Wired, [Online]. Available: https://www.wired.com/story/how-many-water-bottles-could-a-filling-station-save/.
[6] Brondell, "Filtered water sink system specification," [Online]. Available: https://images.homedepot.ca/pdf/1001002329_specification-pdf.pdf.
[7] Home Depot, "Brondell coral 3-stage undercounter water filteration system," [Online]. Available: https://www.homedepot.ca/product/brondell-coral-3-stage-undercounter-water-filteration-system/1001002329.
[8] Home Depot, "Filtered water sink comparable 2 specification," [Online]. Available: https://images.homedepot.ca/pdf/RO500-pdf.pdf.
[9] All About Circuits, "MCD to Lumens calculator," [Online]. Available: https://www.allaboutcircuits.com/tools/convert-mcd-to-lumens/.
[10] LED Supply, "LED information and specifications," [Online]. Available: https://www.ledsupply.com/blog/what-you-need-to-know-about-leds/.
[11] SparkFun, "Analysis on Color LED brightness and resistances," [Online]. Available: https://www.sparkfun.com/news/5461.
[12] SparkFun, "LED Brightness Adjuster Kit," [Online]. Available: https://www.sparkfun.com/products/21226.
[13] SparkFun Forum, "Discussion on required brightness for LED visible in daylight," [Online]. Available: https://forum.sparkfun.com/viewtopic.php?t=45989.
[14] Lumex, "LED specification sheet," [Online]. Available: https://www.mouser.ca/datasheet/2/244/lumex_lumx-s-a0003593211-1-1737925.pdf.
[15] Canadian Centre for Occupational Health and Safety, "Desk viewing angle," [Online]. Available: https://www.ccohs.ca/oshanswers/ergonomics/office/monitor_positioning.html.
[16] Cleveland Clinic, "Acceptable range of temperatures," [Online]. Available: https://health.clevelandclinic.org/cold-water-vs-warm-water/.
[17] The Ice Diet, "About Dr. Brian Weiner," [Online]. Available: https://theicediet.com/about_Dr.html.
[18] Wise Well, "Which water temperature is best for drinking," [Online]. Available: https://wisewell.ae/blogs/news/which-water-temperature-is-best-for-drinking.
[19] Amazon, "Heating cooling mechanism product example," [Online]. Available: https://www.amazon.ca/Coffee-Warmer-Cooler-Desktop-Drinks/dp/B081D1XYMN.
[20] Vat19, "USB Warmer & Cooler," [Online]. Available: https://www.vat19.com/item/usb-warmer-cooler.
[21] Hearing Health Foundation, "Normal noise ranges," [Online]. Available: https://hearinghealthfoundation.org/decibel-levels.

[22] Five Point Seven, "How to design a pleasant notification sound," [Online]. Available: https://medium.com/@fivepointseven/how-to-design-a-pleasant-alert-sound-2ddf7a9724de.
[23] Brilliant Labs, "UNB Dev Board documentation," [Online]. Available: https://www.brilliantlabs.ca/documents/bBoard/b.Board-Getting-Started-Sept_2021.pdf.
[24] Brilliant Labs, "b.Board specifications and Allegro Microsystems A3916GESTR-T product details," [Online]. Available: https://www.digikey.ca/en/products/detail/allegro-microsystems/A3916GESTR-T/6348448.
[25] Digikey, "Allegro Microsystems A3916GESTR-T product details," [Online]. Available: https://www.digikey.ca/en/products/detail/allegro-microsystems/A3916GESTR-T/6348448.
[26] CCOHS. "Monitor Positioning." Canadian Centre for Occupational Health and Safety, [Online]. Available: https://www.ccohs.ca/oshanswers/ergonomics/office/monitor_positioning.html.
[27] R. D. Knight, "Physics for Scientists and Engineers," 5th ed. Addison-Wesley, Jun. 30, 2021.
[28] Course Hero, Available: https://www.coursehero.com/tutors-problems/Physics/9842792-Please-answer-the-question-I-below-about-acoustics-Thank-you-It-was/.
[29] "DIY Water Filling Machine Using Flow Sensor & Arduino," How2Electronics, Available: https://how2electronics.com/diy-water-filling-machine-using-flow-sensor-arduino/.
[30] Mark Pallay, Shahrzad Towfighian, “A Combined MEMS Threshold Pressure Sensor and Switch,” October 30, 2019. https://ieeexplore.ieee.org/document/8956554
[31] Emzone, “134a A/C Refrigerant,” [Online]. Available: https://emzone.ca/product/134a-a-c-refrigerant/
[32] SWEP, “Refrigeration Handbook – 4.4. Capillary Tubes,” [Online]. Available: https://www.swep.net/refrigerant-handbook/4.-expansion-valves/adf4/
[33] Interlink Electronics, "Datasheet for Force Sensing Resistor (FSR)," Interlink Electronics, November 2017. [Online]. Available: https://cdn2.hubspot.net/hubfs/3899023/Interlinkelectronics%20November2017/Docs/Datasheet_FSR.pdf
[34] MikroElektronika, "Reed Click User Manual," MikroElektronika, v1.0.0, [Online]. Available: https://download.mikroe.com/documents/add-on-boards/click/reed/reed-click-user-manual-v100.pdf

Appendix B - Code

```
def playSound(choice: number):
    global startPlayed, endPlayed
    if choice == 1 and startPlayed == 0:
        music.start_melody(music.built_in_melody(Melodies.POWER_UP),
            MelodyOptions.ONCE_IN_BACKGROUND)
        startPlayed = 1
    if choice == 0 and endPlayed == 0:
        music.start_melody(music.built_in_melody(Melodies.POWER_DOWN),
            MelodyOptions.ONCE_IN_BACKGROUND)
        endPlayed = 1
def setRGB(red: number, green: number, blue: number):
    global r, g, b
    r = red
    g = green
    b = blue
    pins.analog_write_pin(AnalogPin.P13, b)
    pins.analog_write_pin(AnalogPin.P7, g)
    pins.analog_write_pin(AnalogPin.P6, r)

def on_button_pressed_a():
    bottleCalibrate()
input.on_button_pressed(Button.A, on_button_pressed_a)

def bottleCalibrate():
    global weight_2, min_weight, _2lightweight, _3lightWeight,
_4lightWeight, _5lightWeight, maxweight
    weight_2 = pins.analog_read_pin(AnalogPin.P2)
    min_weight = weight_2 - 25
    _2lightweight = weight_2 + 90
    _3lightWeight = _2lightweight + 90
    _4lightWeight = _3lightWeight + 90
    _5lightWeight = _4lightWeight + 90
    maxweight = _5lightWeight + 40
    basic.show_string("Calibrate Complete!")

def on_button_pressed_b():
    WaterLevelLights()
input.on_button_pressed(Button.B, on_button_pressed_b)

def WaterLevelLights():
    global motor
    if weight >= _5lightWeight:
```

```
        BLiXel.set_pixel_colour(BLiXel.blixel_index(BLiXelIndex.ONE),
0x007fff)
        BLiXel.set_pixel_colour(BLiXel.blixel_index(BLiXelIndex.TWO),
0x007fff)

BLiXel.set_pixel_colour(BLiXel.blixel_index(BLiXelIndex.THREE),
0x007fff)
        BLiXel.set_pixel_colour(BLiXel.blixel_index(BLiXelIndex.FOUR),
0x007fff)
        BLiXel.set_pixel_colour(BLiXel.blixel_index(BLiXelIndex.FIVE),
0x007fff)
        playSound(0)
        setRGB(0, 0, 255)
        motor = 0
    elif weight >= _4lightWeight:
        BLiXel.set_pixel_colour(BLiXel.blixel_index(BLiXelIndex.FOUR),
0x007fff)

BLiXel.set_pixel_colour(BLiXel.blixel_index(BLiXelIndex.THREE),
0x007fff)
        BLiXel.set_pixel_colour(BLiXel.blixel_index(BLiXelIndex.TWO),
0x007fff)
        BLiXel.set_pixel_colour(BLiXel.blixel_index(BLiXelIndex.ONE),
0x007fff)
        motor = 1
    elif weight >= _3lightWeight:
        BLiXel.set_pixel_colour(BLiXel.blixel_index(BLiXelIndex.ONE),
0x007fff)
        BLiXel.set_pixel_colour(BLiXel.blixel_index(BLiXelIndex.TWO),
0x007fff)

BLiXel.set_pixel_colour(BLiXel.blixel_index(BLiXelIndex.THREE),
0x007fff)
        motor = 1
    elif weight >= _2lightweight:
        BLiXel.set_pixel_colour(BLiXel.blixel_index(BLiXelIndex.ONE),
0x007fff)
        BLiXel.set_pixel_colour(BLiXel.blixel_index(BLiXelIndex.TWO),
0x007fff)
        motor = 1
    elif weight >= min_weight:
        BLiXel.set_pixel_colour(BLiXel.blixel_index(BLiXelIndex.ONE),
0x007fff)
        playSound(1)
        setRGB(255, 0, 0)
```

```
        motor = 1
    else:
        setRGB(255, 0, 0)
        motor = 0
def SystemRestart():
    global startPlayed, endPlayed, motor
    bBoard_Motor.motor_right_duty(0)
    startPlayed = 0
    endPlayed = 0
    motor = 1
    control.wait_micros(6000)
    control.wait_micros(6000)
    BLiXel.blixels_off()
    setRGB(0, 0, 0)
def setMotor():
    if motor == 1:
        bBoard_Motor.motor_right_duty(80)
    else:
        bBoard_Motor.motor_right_duty(0)
        control.wait_micros(6000)
        control.wait_micros(6000)
def WaterBottleCheck():
    if weight <= maxweight and weight > min_weight and
Reed2.get_switch() == 1:
        WaterLevelLights()
        setMotor()
    else:
        SystemRestart()
Reed2: Reed.Reed = None
weight = 0
weight_2 = 0
b = 0
g = 0
r = 0
endPlayed = 0
startPlayed = 0
motor = 0
_5lightWeight = 0
_4lightWeight = 0
_3lightWeight = 0
_2lightweight = 0
min_weight = 0
maxweight = 0
setRGB(0, 0, 0)
music.set_volume(25)
```

```
maxweight = 800
min_weight = 400
_2lightweight = 490
_3lightWeight = 580
_4lightWeight = 670
_5lightWeight = 760
motor = 1

def on_forever():
    global weight_2
    weight_2 = pins.analog_read_pin(AnalogPin.P2)
    basic.show_string("" + str(weight))
    WaterBottleCheck()
basic.forever(on_forever)

def on_every_interval():
    global weight, Reed2
    weight = pins.analog_read_pin(AnalogPin.P2)
    Reed2 = Reed.create_reed(BoardID.ZERO, ClickID.B)
loops.every_interval(100, on_every_interval)
```

Appendix C - Figures

To determine the SPL at a different distance, we can use the inverse square law, which states that the intensity of a sound wave decreases with the square of the distance from the source. The formula to calculate the change in SPL with distance is:

$$\Delta SPL = 20 \times \log_{10}\left(\frac{r_1}{r_2}\right)$$

Where:

- ΔSPL is the change in SPL in decibels (dB).
- r_1 is the original distance (10cm in this case).
- r_2 is the new distance (60cm).

Let's calculate the change in SPL:

$$\Delta SPL = 20 \times \log_{10}\left(\frac{10cm}{60cm}\right)$$

$$\Delta SPL = 20 \times \log_{10}(0.1667)$$

$$\Delta SPL \approx -15.6 \text{ dB}$$

The SPL will decrease by approximately 15.6 dB when moving from 10cm to 60cm.

Given the original SPL at 10cm was 90dB, the SPL at 60cm would be:

$$SPL_{60cm} = 90 \text{ dB} - 15.6 \text{ dB}$$
$$SPL_{60cm} \approx 74.4 \text{ dB}$$

So, at a distance of 60cm, the SPL would be approximately 74.4 dB.

3. **Water Density and Gravity**: For water, the density (ρ) is approximately 1000 kg/m^3, and the gravitational constant (g) is approximately 9.81 m/s^2.

Given these, the hydraulic power (P_hydraulic) required by the pump can be calculated using the formula:

$$P_{\text{hydraulic}} = \rho \times g \times h \times Q$$

Where:

- ρ = Density of water (1000 kg/m^3)
- g = Gravitational constant (9.81 m/s^2)
- h = Total head or pressure head in meters (equivalent to the pressure requirement, which is 9.49 kPa or about 0.967 meters of water column)
- Q = Flow rate in cubic meters per second (1.0 gpm is approximately 0.00006309 m^3/s)

Plugging in the values:

$$P_{\text{hydraulic}} = 1000 \times 9.81 \times 0.967 \times 0.00006309$$
$$P_{\text{hydraulic}} \approx 0.602 \text{ Watts}$$

Now, to account for pump efficiency:

$$P_{\text{motor}} = \frac{P_{\text{hydraulic}}}{\text{Efficiency}}$$

Assuming an efficiency of 70% (or 0.7):

$$P_{\text{motor}} = \frac{0.602}{0.7}$$
$$P_{\text{motor}} \approx 0.86 \text{ Watts}$$

So, you'd need a motor that can provide at least 0.86 Watts of mechanical power.

Lastly, ensure that this motor's electrical requirements (voltage, current, and power) are compatible with the motor driver's specifications (which can handle up to 1A of current at 2.7V to 15V). This will help ensure that the driver can adequately control and power the motor.

Figure 7: Speaker Sound Formula and Water pump power calculations

1. **Hydrostatic Pressure**:
   As calculated before, the hydrostatic pressure due to 3 feet of water is approximately:
   $$P \approx 8.973 \text{ kPa}$$
2. **Frictional Losses for 0.5" PVC at 1.0 gpm**:
   Friction loss tables for PVC typically provide values in terms of psi loss per 100 feet of pipe. For 0.5" PVC tubing with a flow rate of 1.0 gpm, a typical friction loss might be around 2.5 psi/100 ft.
   Given that the length of the tubing in your scenario is 3 feet, the frictional loss for 3 feet is:
   $$\text{Friction Loss} = \frac{3 \text{ ft}}{100 \text{ ft}} \times 2.5 \text{ psi}$$
   $$\text{Friction Loss} \approx 0.075 \text{ psi}$$
   Convert this to kPa (knowing 1 psi ≈ 6.895 kPa):
   $$\text{Friction Loss} \approx 0.075 \times 6.895$$
   $$\text{Friction Loss} \approx 0.517kPa$$
3. **Total Pressure Requirement**:
   Combining hydrostatic and frictional pressures:
   $$P_{\text{total}} = P + \text{Friction Loss}$$
   $$P_{\text{total}} = 8.973 \text{ kPa} + 0.517 \text{ kPa}$$
   $$P_{\text{total}} \approx 9.49 \text{ kPa}$$

So, the pump should be capable of delivering a pressure of at least 9.49 kPa to achieve a flow rate of 1.0 gpm through 0.5" PVC tubing elevated 3 feet.

Figure 8: Water pump calculations

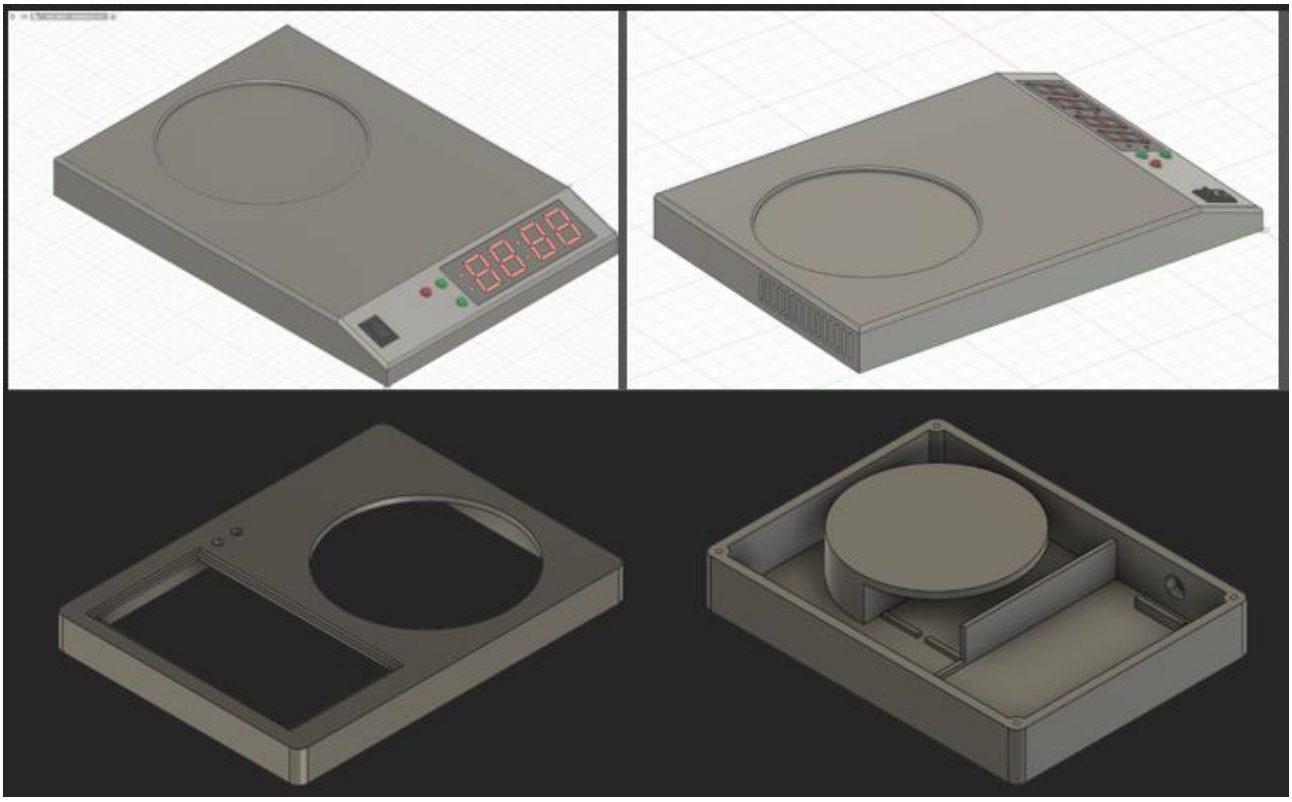

Figure 9: Prototype Version 1 and Final Implementation