Speaker Shower Head

ECE 445 Design Document

Team 19
Abhijun Gupta (abhijun2)
Bhavana Ambatipudi (bhavana4)
Manav Modi (manavm2)

TA:
Akshat Sanghvi

Senior Design Spring 2023
February 23, 2023

Table of Contents

1 Introduction

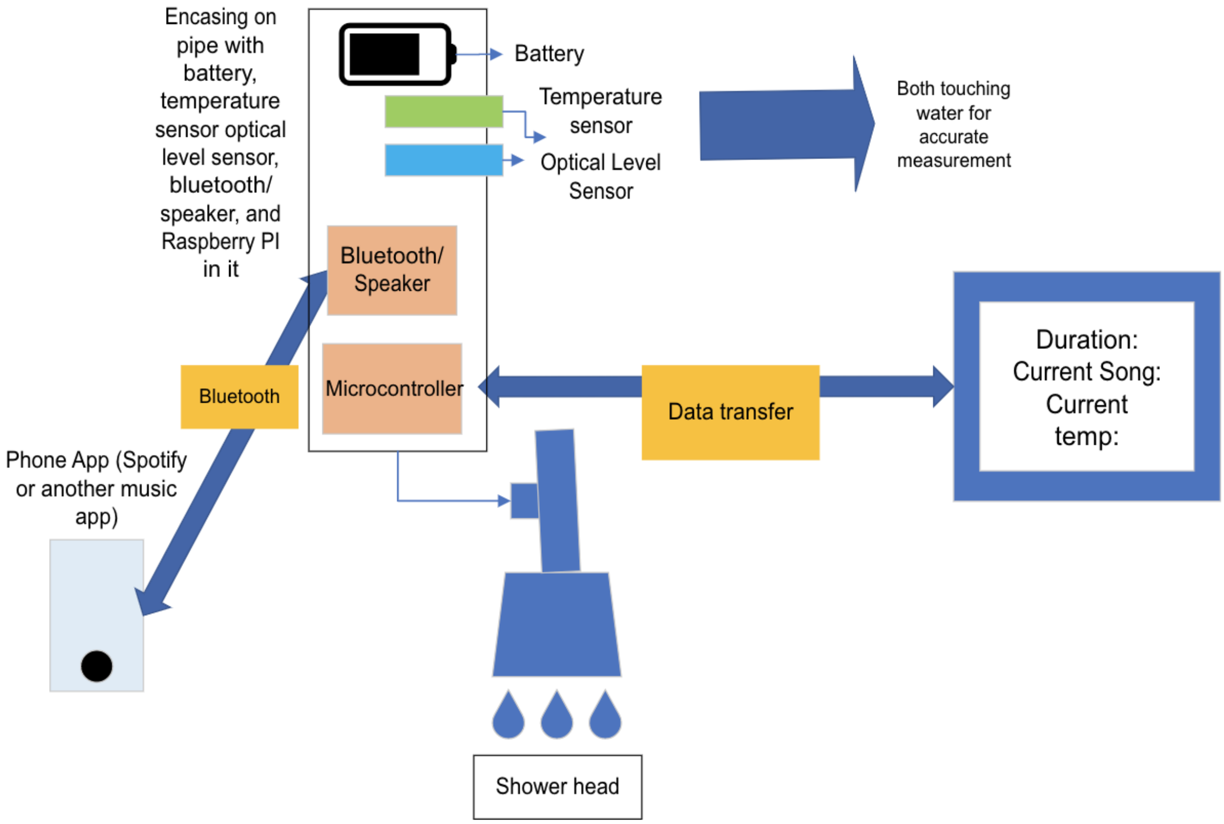
1.1 Problem

Showering can often become a monotonous chore with nothing to do but be left with their thoughts. Some people prefer to use their phone to play music while showering to have some to listen to or sing along to. The music is often blocked by the shower curtain and drowned out by the water leading to a lackluster listening experience. Additionally, in the shower you cannot control the song that is playing as the phone would get wet if used inside the shower. Another common problem in the shower is initially setting the proper temperature of the shower. Most people utilize the guess and check method to get the correct temperature for the shower. Although this method can work it leads to users standing outside for longer than necessary and sometimes having the temperature be just a bit off. Some people in the shower often lose track of time in the shower and use a lot more water than they need to. Having a timer in the shower to notify users of the duration of the shower will motivate people to waste less water and take shorter showers. Our goal is to enhance the shower experience by allowing users to control their music, monitor shower duration, as well as see the current temperature of the water to ensure that the temperature is just right every time.

1.2 Solution

Our objective is to address some of the challenges associated with showers including displaying the current water temperature, shower duration, and playing music. Our showerhead design will incorporate a temperature sensor to allow users to view the current water temperature so they can set the dials right to get the temperature exactly right before even stepping into the shower. The showerhead will also keep track of average shower duration by using a moisture sensor to motivate users to utilize less water. Additionally we will add a bluetooth speaker to the showerhead to allow control of music and better audio quality than a phone outside of the shower. All in all, the showerhead will incorporate sensors for detecting water temperature and if the shower is currently running, a bluetooth speaker to play music, a display for displaying sensor information, buttons to control the music being played by the shower, and a sturdy physical structure suitable for most shower setups to improve shower quality.

1.3 Visual Aid



1.4 High-level requirements

Temperature Sensing

1. Correctly displaying temperature of the water: one of the core requirements of our design is accurately measuring how hot or cold the water is for the user.

Shower Duration

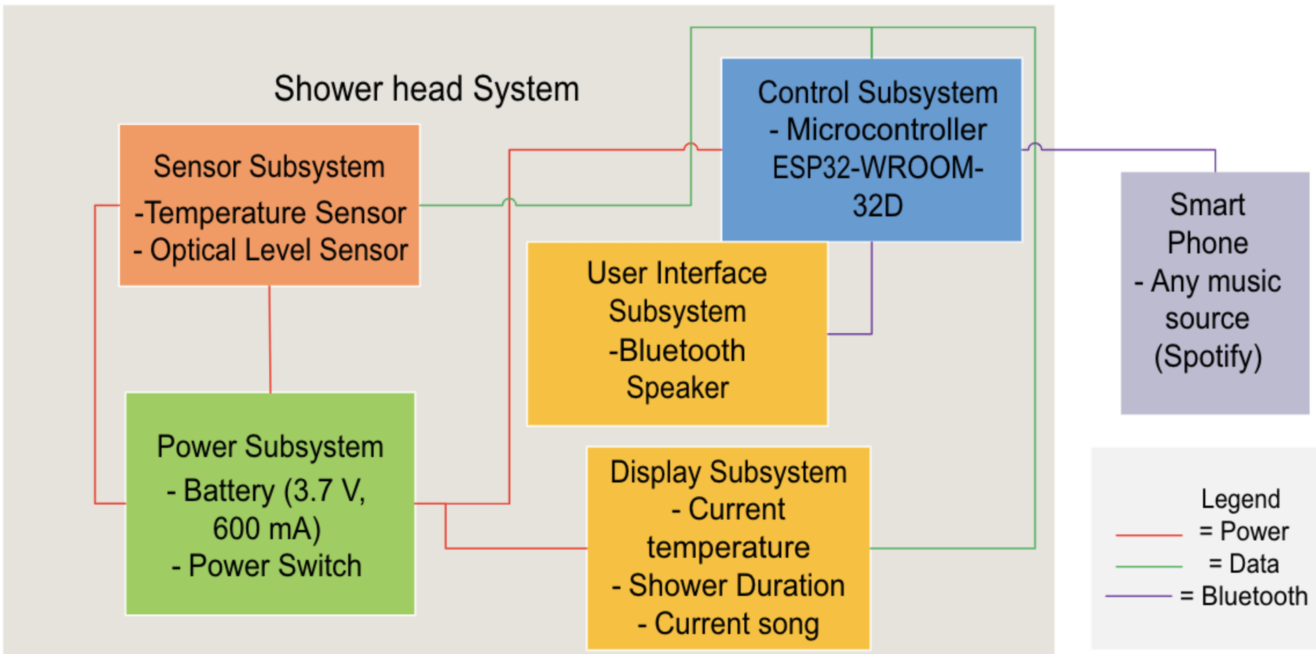
2. Correctly detecting if the shower is on: the display should accurately show the current shower duration as well as the average shower duration.

Speaker/Button Functionality

3. Have the ability to connect a phone to the speaker through bluetooth and play music through a speaker which can be controlled by the buttons on the shower remote.

2 Design

2.1 Block Diagram



2.2 Physical Diagram

Diagram 1: Shower Head Sensors

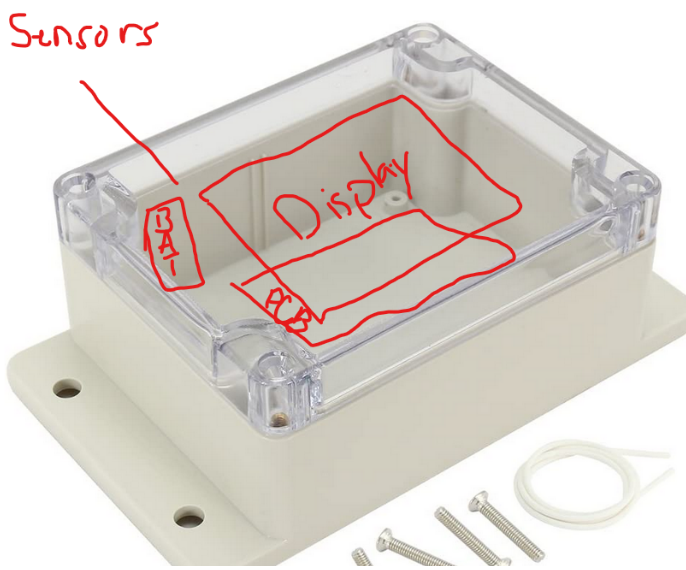


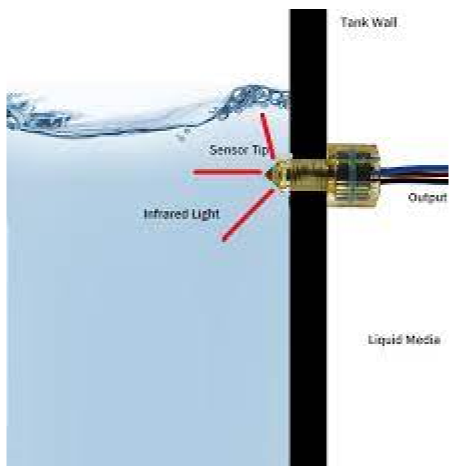
Diagram 2: Display Housing

The Physical Diagram shown above shows our plan for our additions for the shower. The sensors will be drilled and added to the top of the showerhead as shown in diagram 1. The sensors will be glued to the top of the showerhead using epoxy to ensure water does not escape from the holes drilled to the showerhead. Diagram 2 shows the waterproof housing for the PCB, battery, and display. This will be connected to the sensor output wires and attached to the shower wall for easy user access to the display and any changes needed.

2.3 Subsystem Overview and Block Diagram Requirements

Sensor Subsystem

The sensor subsystem is important in order to detect the temperature of the water and display it on the display. We will use a temperature sensor in order to accurately measure the temperature of the water. The temperature sensor will be inside of the box encasing on top of the pipe that connects to any shower head. With this encasing, we will have a tiny hole in which the temperature sensor can touch the water for the most accurate measurement. Our sensor subsystem will also need to check whether or not there is water running from the shower head. For this, we will use an optical level sensor. The optical level sensor we are using requires the tip to be touching the water. In order to do this, we will use the same method we have used for the temperature sensor and cut a hole in the pipe and the box encasing to allow the tip to touch the water. The image below shows an illustration of a hole being cut into the pipe and the tip of each sensor touching the water.



Both of these features will enhance the user's shower experience by allowing them to know the temperature of the water they want for future showers and knowing how long their shower experience is. A temperature sensor requires around 2.7-5.5 Volts in order to properly work and an optical liquid level requires around 2.7 V to properly work. In this case we would err on the side of caution and require 5 V for each device to work the way we intend.

R&V Table 1: Showerhead Sensor Subsystem	
Requirements	Verifications
Accurately sensing temperature of the shower water when the shower is on	● Use a thermometer in order to check if the temperature sensor is accurately

	measuring the water temperature with a margin of error around 1%
When the shower is detected to be on, the optical liquid level sensor will detect the water as running	● We will use visual aid in order to detect whether the shower is on ● If the shower is on but the optical level sensor is detecting it to be off, then we will adjust the hole for the optical sensor so that more of the tip is touching the water ● If adjusting the tip does not work, then we will replace our optical liquid level sensor
When the shower is detected to be off, the optical liquid level sensor will detect the water as not running	● We will use visual aid in order to detect whether the shower is off ● If the shower is off but the optical level sensor is detecting it to be on, then we know that there is liquid inside of the pipe that is not accounted for after the water is shut off ● If there is still liquid inside of the pipe, we will need to find a technique to dry the pipe after every shower use

Control Subsystem

The showerhead control system is used as a communication device to connect to the remote. In this case we will be using an ESP32 in both the shower head control system and the remote control system in order to send information regarding the temperature of the water and how long the optical liquid level sensor is on. In this case there will be a PCB using a bluetooth connection that will connect to the remote and send information gathered from the sensor subsystem and bluetooth speaker. The PCB will be placed inside the box placed on top of the pipe with all of the other requirements for the shower head system to work. The control subsystem will require around 5 V for the PCB to function and send information from the shower head to the remote.

R&V Table 2: Showerhead Control Subsystem	
Requirements	Verifications
Connect to the ESP32 and get I/O information	● Use commands to check if the ESP32

	is active and communicating effectively ● Once the PCB is plugged in check if it has power ● Connect the ESP32 to WiFi ● Will know if it is connected to another device also on the internet by using the “Ping” command
Send sensor subsystem information to the remote	● Will know if it is working if the information being inputted and outputted regarding the shower temperature and duration is accurate ● All of the accuracy verifications regarding the temperature and duration of the shower is mentioned in Table 1: Shower head Sensor Subsystem
Take button input information and control the speaker	● Button input information will involve the rewind, fast forward, and play/pause buttons. ● Have to make sure the ESP32 is also in control of the speaker by connecting the ESP32 and the speaker to Bluetooth so that whatever information the shower head ESP32 is receiving from the remote can be used to change the song information ● More info in Table 3 about how to verify above

Speaker Subsystem

The speaker subsystem will consist of a bluetooth speaker that will be able to connect to the user’s phone and be able to play music depending on the user’s choosing. The speaker will also connect to the communication device in the shower head using Bluetooth. This way, the information that is being received from the remote buttons can also be used to control the song that is being played. The bluetooth speaker will be put inside of the enclosed casing that is on top of the pipe. In this case, the speaker and bluetooth will be attached to the pipe that can connect to any shower head of the user’s choice. This allows the user to have music they can play of their own choice whenever they are in the shower.

R&V Table 3: Showerhead Speaker Subsystem	
Requirements	Verifications
Connect to phones with bluetooth capabilities	● Connect the phone to bluetooth by connecting the phone to the ESP32 ● Turn bluetooth speaker on ● Make sure that can play the song of choice once also connecting bluetooth speaker to ESP32
Connect to ESP32 with bluetooth capabilities	● Use Bluetooth to get WiFi configured ● Check if ESP32 is connected to shower head speaker by playing a song once the phone is also connected to bluetooth speaker
Be able to change/pause/play songs using the button subsystem (controlled by the showerhead control subsystem)	● Make sure the ESP32 on the remote is connected to the ESP32 on the shower head ● Use the buttons to rewind, fast forward, and play/pause the song currently playing ● If this does not work, check the ESP32 connections and refer back to the first requirement's verifications in Table 2 in order to properly pair both ESP32's

Power Subsystem

The system will all be powered by the power subsystem that will provide consistent voltage to all the components. We are planning on using a 3.7 volt lithium ion battery to connect to the components in the shower head (communication device, bluetooth speaker, and the sensors). However, in order to make sure that the 3.7 Volts are not overheating and overpowering many of the subsystems we have, we will be making sure there is a microcontroller in between the power and all of our devices. The power subsystem will have to be carefully placed to ensure no water can get to the battery since the battery cannot be waterproof and would be very close to the shower. Therefore, we will encase the battery in a waterproof housing that will be attached to the shower wall/ We will double encase the battery inside of the original encasing in order to accommodate the temperature and the optical liquid level sensors. In case the holes end up leaking into the box, we will have a separate compartment that will ensure that the battery remains dry. This will make sure that there is no water leaking onto the battery and that there is no power shortage.

R&V Table 4: Showerhead Power Subsystem	
Requirements	Verifications
Stable voltage and amperage throughout the showerhead and its subsystems.	• We will ensure there is stable voltage throughout the subsystems by using a voltmeter to check that the correct voltage is going into each device within a range of +/- 0.1 volts • We will ensure there is stable amperage throughout the subsystems by using an ammeter or a multimeter to check that the correct amount of amps is going into each device within a range of 5% of normal amperage
Double encasing surrounding the battery	• We will add a compartment in the box encasing that is above the pipe • This will ensure that even if there is leakage inside of the encased box through the holes made to put the temperature and optical liquid level sensors the battery will not be affected

Display Subsystem

The display subsystem will output information based on the data collected by the Raspberry PI as well as current information given from the sensor subsystem. The display subsystem will be a small 5-10 inch waterproof display with an input voltage of 5 volts to keep a common necessary voltage throughout the remote. A user-friendly interface will be created for the display on the ESP32. The interface will display important information for the user such as current water temperature, current shower duration, and the current song being played by the speaker. This information will constantly be updated based on the inputs from the sensors and controlled by the ESP32.

R&V Table 7: Display Subsystem

Requirements	Verifications
Displays information based on the current information gathered by the ESP32	● The ESP32 is displaying its home screen ● The ESP32 can create a user-friendly interface on the display ● The ESP32 is able to update the display with current information

2.3 Tolerance Analysis

2.3.1 Battery Power Analysis

One risk of the feasibility of our project is the battery power needed to power our components in the remote including a display, raspberry pi, and buttons. Our initial idea was to use our 9 volt battery with a voltage regulator to power our 3 components at 5 volts, but after doing the tolerance analysis on this aspect of our power distribution we found that a typical 9 volt alkaline battery is rated for 550 mAh while the ESP32 WROOM 32D requires 250 mA for normal operation. With all the components (sensors, display, and bluetooth functionality) the microcontroller will use much more amperage. All of these components combined will have an amperage of about 200mA. Assuming 550 mAh battery and an average consumption of 100mA the battery will only last,

$$550/200 = 2.75 hours$$

Which is decent, but not as much as we would have hoped. After considering this problem and weighing our options we decided a more powerful battery may be the best solution. After some searching we found a rechargeable 3.7 volt lithium ion battery rated for 2000 mAh. A DC-to-DC step up converter circuit also known as a boost converter would need to be added to this battery to output a consistent 5 volts which has a typical power efficiency of 85% giving us,

$$2000mAh * 0.85 = 1700mAh$$

Adding other real world inefficiencies such as heat, battery voltage variation, and raspberry pi amperage pull variation our efficiency would reach closer to approximately 70% on top of the boost converter,

$$1700\,mAh * 0.70 = 1190\,mAh$$

1190 mAh running at 5 volts will be able to power our raspberry pi for,

$$1190\,mAh/200\,mA = 5.95 hours$$

5.95 hours is a very reasonable time between charges since the device will only be on during shower usage. Most users will need to charge on average every 2 weeks. This is a much better alternative to having to plug into the remote while being used inside the shower and can allow the user to use the remote without fear of electrocution from a power wire connecting to the remote.

2.3.2 Waterproofing Analysis

Waterproofing would be the greatest risk to the feasibility of this project since the entire project is planned to be in the shower where water can get anywhere and mixing water with electronics can cause many issues with safety. Our project is still feasible because all of our components can be completely encased in housing that will not allow any water inside. The only things that would not be completely encased would be the sensors but we will use waterproof sensors that will mitigate the risk due to waterproofing.

The IP waterproof standard is useful to understand the waterproofing necessary for us and gives us a way to test if our device will withstand shower conditions. Our housing will be tested with standard IP testing to ensure that over time water does not leak into our electronics. IP x5 is our goal which gives protection against low pressure jetting water. This requires our housing to withstand a light spray of about 4psi from any angle. Our housing should also be IP x6 for the angle at which the water will flow through the pipe. IP x6 require the housing to withstand high pressure jetting water which can be tested with 15 psi of water. There are many housings that can withstand these conditions and many examples that we can use. The IP waterproof rating gives a great standard of protection that we will qualitatively follow for our testing to ensure that our showerhead will mitigate risk as much as possible.

IP Rating Number	Ingress of Solids (1st Number)	Ingress of Liquids (2nd Number)
0	No protection	No protection
1	Protected against objects > 50mm	Protected against vertically falling water drops or condensation (dripping water)
2	Protected against objects >12.5mm	Protected against falling drops of water up to a 15° angle
3	Protected against objects > 2.5mm	Protected against spraying water up to 60° angle
4	Protected against objects > 1.0mm	Protected against splashing water
5	Limited protection against dust (no harmful deposit)	Protected against low pressure jetting water
6	Total dust protection	Protected against high pressure jetting water
7	-	Protected against short periods of in up to 1 meter of water depth
8	-	Protected against water immersion at depths ≥1 meter
9	-	Protected against close-range high pressure, high temperature spray downs.

IP Rating Standard

3 Cost and Schedule

3.1 Cost Analysis

We can expect a salary of $40/hr×5 hr×75 = $15000 per team member. We need to multiply this amount with the number of team members, $15000× 3 = $45,000 in labor cost.

The following is the table with the information of the parts we require:

Description	Manufacturer	Quantity	Price ($)
Temperature and Humidity Sensor, T9602 Series	Amphenol Advanced Sensors	1	29.37
IAKLE High Pressure Shower Head with Hose,3-Setting Filtered Detachable Water Saving Jet Handheld Shower Heads Filter for Hard Water,1.6GPM(80in/2m Hose)	IAKLE	1	10.99
Temperature Sensor - Waterproof (DS18B20)	ROHS	1	10.95
Raspberry Pi 3 – Model A+ (PLUS) - 512MB RAM	Raspberry Pi	2	50
Duracell - 9 Volt Battery	Duracell	2	14
2.8" TFT Display with Resistive Touchscreen	Adafruit	1	14.95

Total Parts: $130.26
Total Cost of Labor: $45,000
Total : $45,130,26

3.2 Schedule

Week	**Task**	**Person**
Feb 27th - Mar 3rd	Order more parts like the display and others that might be required	Manav
	Work on design document	Everyone
	Talk to the Machine Shop	Manav
	Start Designing PCB	Everyone
Mar 6th - Mar 10th	Continue PCB desining	Abhi
	Start Board Assembly	Bhavana
	Work with machine shop to implement the showerhead with sensor	Bhavana
	Finalize all the sensors and get started on the remote	Everyone
Mar 13th - Mar 17th	Spring Break	Everyone
Mar 20th - Mar 24th	Order PCB	Abhi
	Finalize Microcontroller	Everyone
	Start integrating all the sensors with the showerhead	Manav
Mar 27th - Mar 31st	Work on communication with the showerhead and display	Bhavana
	Implement the display on the remote	Abhi
	Finish the sensor integration with the showerhead	Manav
	Revise the PCB and reorder it	Everyone
Apr 3rd - Apr 7th	Finish working on the remote	Bhavana
	Test the PCB	Abhi
	Test the showerhead	Manav
Apr 10th - Apr 14th	Test the remote	Manav
	Make bug fixes to the showerhead	Bhavana
	Prepare for demo	Everyone
Apr 17th - Apr 20th	Establish communicaton with remote and showerhead	Abhi
	Check everything and test it	Everyone
	Prepare for demo	Everyone
Apr 21st	MOCK DEMO	
Apr 26th	FINAL DEMO	

4 Ethics and Safety

The main IEEE ethical dilemma we are facing is the safety aspect of water with all of our equipment. Most of the safety features regarding our project involve water. As we are creating a showerhead that will interact directly with the water, we need to make sure that most of our equipment we are using is waterproof. We also need to consider the power source of our project being affected by water. We cannot have a battery inside of the shower reacting with the water as that will be dangerous. Another option we had was to keep the power source outside of the shower; however, we would need to have wires connecting to our device which would also pose a threat. We have decided to encase the power source and its connecting wires inside of a box in order to ensure no water can disrupt the system.

It is crucial to put safety first and take all required safeguards to protect both the user and the equipment when building electrical gadgets that interact with water. This entails giving considerable thought to the device's design, the materials it is made of, and its intended purpose.

The construction of a watertight box to contain the battery and sensor is a crucial step in ensuring safety. This will aid in preventing water damage, which can malfunction, short circuit, or even start fires, in the electronics. The box should be built to be totally impermeable to water and strong enough to sustain any pressure or impact during usage (Brown).

Also, it's critical to include detailed instructions on how to operate the gadget safely, highlighting the need of keeping it dry and preventing any contact with water. This might contain instructions on how to operate the gadget, how to use it safely, and what to do if it becomes accidentally submerged in water.

The IEEE's Code of Ethics, which describes the duties of engineers in their profession, should be borne in mind when creating such a device. In accordance with this code, it is crucial to put the public's safety and welfare first, be truthful and objective in your job, and abstain from any conflicts of interest or other unethical actions (IEEE Xplore).

5 References

"Optical Level Sensors: Research and Buy Optical Liquid Level Sensors." *SMD Fluid Controls*, 6 Oct. 2021, https://www.fluidswitch.com/2015/10/16/how-do-optical-level-sensors-work/

"How to Configure WIFI on Raspberry Pi 4." *BINARYUPDATES.COM*, 4 Aug. 2021, https://binaryupdates.com/how-to-configure-wifi-on-raspberry-pi-4/#:~:text=The%20first%20step%20is%20to,networks%20with%20all%20necessary%20information

Brown, Christian. "Nothing Gets in: Waterproof Enclosure Design 101 (and IP68)." *Fictiv*, https://www.fictiv.com/articles/nothing-gets-in-waterproof-enclosure-design-101-and-ip68

IEEE Xplore. https://ieeexplore.ieee.org/Xplore/home.jsp

Martin, Taylor. "How to Setup Bluetooth on a Raspberry Pi 3." *CNET*, CNET, 20 May 2016, https://www.cnet.com/tech/computing/how-to-setup-bluetooth-on-a-raspberry-pi-3/