Final Report for ECE 445, Senior Design, Spring 2023

TA: Akshat Sanghvi

3rd May 2023

Project No. 19

Bhavana Ambatipudi

Abhijun Gupta

Manav Modi

By

SPEAKER shower head

**Abstract**

The report talks about the Speaker Shower Head project. The project focuses on enhancing the user's shower experience by displaying the shower temperature, shower length, average shower length and the song playing on the waterproof Bluetooth speaker to the user. This project was created to help solve problems in the shower such as temperature variability and ensuring users can keep track of their shower time. We were able to accomplish our goals and create a user-friendly waterproof shower head that displayed accurate information

Contents

[1. Introduction 1](#_Toc134044928)

[1.1 Solution 1](#_Toc134044929)

[1.2 High Level Requirements 1](#_Toc134044930)

[2 Design 2](#_Toc134044931)

[2.1 Visual Aid 2](#_Toc134044932)

[2.2 Block Diagram 4](#_Toc134044933)

[2.3 Subsystems 5](#_Toc134044934)

[2.3.1 Sensor Subsystem 5](#_Toc134044935)

[2.3.2 Control Subsystem 5](#_Toc134044936)

[2.3.3 Power Subsystem 6](#_Toc134044937)

[2.3.4 Display Subsystem 6](#_Toc134044938)

[2.3.5 Speaker Subsystem 6](#_Toc134044939)

[3. PCB Design and ESP32 Programming 6](#_Toc134044940)

[3.1 Microcontroller 8](#_Toc134044941)

[3.2 Programming the ESP32 8](#_Toc134044942)

[4. Costs 9](#_Toc134044943)

[4.1 Parts 10](#_Toc134044944)

[4.2 Labor 10](#_Toc134044945)

[4.3 Total Cost 10](#_Toc134044946)

[5. Conclusion 11](#_Toc134044947)

[5.1 Accomplishments 11](#_Toc134044948)

[5.2 Ethical considerations 11](#_Toc134044949)

[5.3 Future work 12](#_Toc134044950)

[References 13](#_Toc134044951)

[Appendix A Requirement and Verification Table 14](#_Toc134044952)

# 1. Introduction

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.

## 1.1 Solution

Our objective was 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 added 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.

## 1.2 High Level Requirements

Temperature Sensing

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

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

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 Visual Aid

|  |  |
| --- | --- |
| Our design changed quite a lot from what we had originally planned. Our original visual aid consisted of there being a box placed on top of the showerhead that would allow the sensors to connect to the showerhead as well as encasing all our electrical wires. The original visual aid is shown down below:    Diagram 1: Original Visual Aid |  |
| After completing the design review, we realized we could not keep our design outline the same. Instead of having a box connected on the showerhead pipes, we instead decided to put the box on the side of the shower. Our new visual aid is shown below:  Diagram 2: Final Visual Aid  Below are the sensors subsystem and the display subsystem wirings: |  |

Diagram 3: Showerhead Design with Sensors

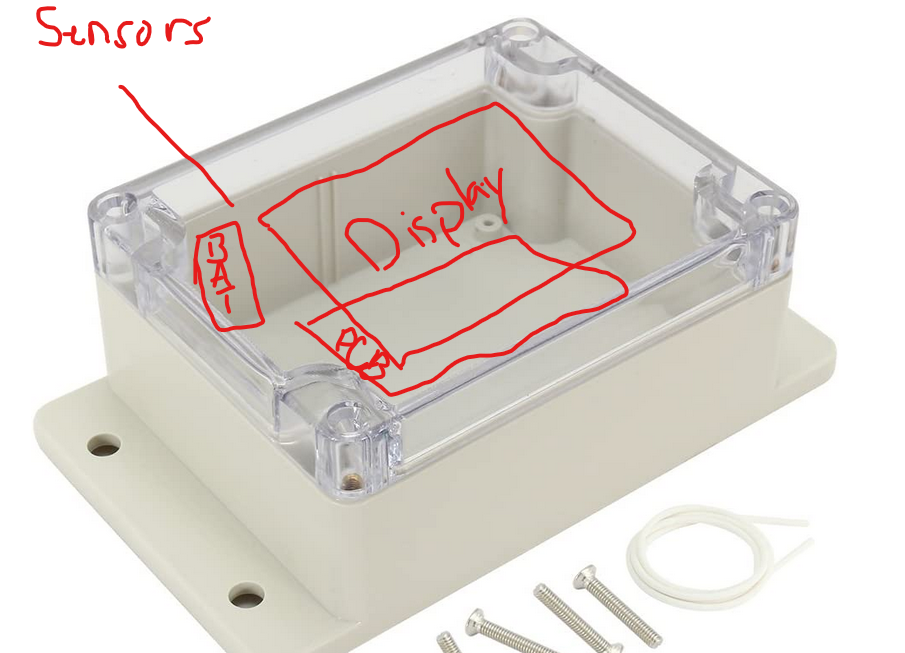


Diagram 4: Display Box

## 2.2 Block Diagram

Below is our block diagram. There are five main subsystems that made our entire design possible: the control, user interface, display, sensor, and power subsystems.

Diagram

Description automatically generated

Diagram 5: Block Diagram

## 2.3 Subsystems

### 2.3.1 Sensor Subsystem

The sensor subsystem is important to detect the temperature of the water and display it on the display. We will use a temperature sensor 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 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 [1]. 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.

Diagram

Description automatically generated

Diagram 6: Optical sensor touching water

Both 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 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.

### 2.3.2 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-WROOM-32D 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 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.

### 2.3.3 Power Subsystem

The system will all be powered by the power subsystem that will provide consistent voltage to all the components. We used a 9-volt alkaline battery to connect to the components in the shower head (communication device, bluetooth speaker, and the sensors). However, to make sure that the 9 Volts are not overheating and overpowering many of the subsystems we have, we will be making sure there is a voltage regulator in between the power and all 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 to accommodate the temperature and the optical liquid level sensors [5]. 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.

### 2.3.4 Display Subsystem

The display subsystem will output information based on the data collected by the microcontroller 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.

### 2.3.5 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.

# 3. PCB Design and ESP32 Programming

To get our entire system working, we needed to create a PCB that would house our microcontroller and connect it to our display and our sensors. We also had to add components that would allow us to program the microcontroller and components that would allow us to have proper voltage to all of the necessary components.

A screenshot of a video game

Description automatically generatedDiagram, schematic

Description automatically generated

Diagram 7: Final PCB Schematic

Diagram 7: Final PCB Design

## 3.1 Microcontroller

For our microcontroller, we used the ESP32-WROOM-32D. We chose this microcontroller as it would allow us to use Wi-Fi and Bluetooth that we needed to get our speaker subsystem working. Our microcontroller also had many GPIO analog and digital pins that would allow us to connect our temperature sensor, our optical liquid level sensor, and our display.

## 3.2 Programming the ESP32

To program the ESP32-WROOM-32D, we needed to add a BOOT option and a UART connection. For the BOOT option, we created a 2x1 pin connector that connected to GPIO pin 1. When programming the ESP32, we needed to set the BOOT option to low and ground it for the chip to receive data. We also needed to add a UART connection that would allow us to connect to our computers and use Arduino IDE to upload the code to the ESP32, so it understood what to make of the information received from the sensors. You can see some part of the code below:

Text

Description automatically generated

*Code 1: It was responsible for getting the temperature sensor information and showing it on the display.*

*Text

Description automatically generated*

*Code 2: It was responsible for calculating the shower length that the person is taking as well as average shower length.*

*Text

Description automatically generated*

*Code 3: It connects to the backend server to get the song name and show it on the display.*

# 4. Costs

For our project, we ended up spending a little more than the budget we were given of $150.

## 4.1 Parts

|  |  |  |
| --- | --- | --- |
| **Part** | **Manufacturer** | **Actual Cost ($)** |
| Temperature Sensor | ROHS | $10.95 |
| Showerhead with Hose | IAKLE | $10.99 |
| 9 Volt Battery | Duracell | $7.00 |
| Waterproof Encasing | Zulkit | $14.99 |
| Optical Liquid Level Sensor | EPTTECH | $5.20 |
| 2.8” TFT Display | Adafruit | 14.95 |
| Various Connectors | Digikey/Sparkfun | $10.85 |
| PCB | PCBWay | $5.00 |
| Voltage Regulators | Mouser Electronics | $5.25 |
| ESP 32 WROOM 32d | Digikey | $2.50 |
| Various Resistors and Capacitors | Digikey | $3.69 |
| Waterproof Bluetooth Speaker | Amazon | $14.99 |
| **Total** |  | **$106.36** |

## 4.2 Labor

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. We reached the salary of $40/hr based on an average of how much we will all be making in our jobs next year.

## 4.3 Total Cost

Including the labor costs with the individual parts cost our total cost for designing this product comes out to $45,106.36 which is mostly made up of the labor costs but shows just how much it takes to design a new product. Even with such a high cost for designing I think our product has a lot of worthwhile considerations and could eventually recuperate the initial high cost.

# 5. Conclusion

## 5.1 Accomplishments

We had several accomplishments throughout the project. Some of the followings were:

* Creating a working PCB that was able to bring together all our different working components onto one board.
* Using our optical liquid level sensor and water temperature sensor effectively to get all the values from our showerhead.
* Creating Arduino code that was able to display all the essential information on the display.
* Using the ESP32 to connect to the wifi to read the song name from the backend.

## 5.2 Ethical considerations

The main IEEE ethical dilemma we faced is the safety aspect of water with all of our equipment [4]. Most of the safety features regarding our project involve water. As we are creating a showerhead that will interact directly with the water, we needed to make sure that the equipment we used was waterproof. We also needed to consider the power source of our project being affected by water. We could not have a battery inside of the shower reacting with the water as that would be dangerous [4]. 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 decided to encase the power source and its connecting wires inside of a box to ensure no water can disrupt the system.

Text

Description automatically generatedFor our final demo, we also created a safety considerations checklist for the user. This would ensure that if there was something wrong with the system, the user would have instructions on what to fix and what to let a professional fix.

Diagram 8: Safety Checklist for User

## 5.3 Future work

In the future, we would like to add certain features while increasing budget costs to get better equipment. We would like to add a switch to turn the battery on or off so that we do not waste the battery life. This would ideally be added to the box with the PCB so that we could connect the battery to the switch on the PCB. We also would like to save the shower times on the ESP32 flash memory so that even if the display turns off, the user is able to see their average shower times displayed. Another future consideration for the project would be to add a flow meter to the device so that we can measure how much water the user is using. This would be helpful in places that experience a lot of droughts to limit people’s water usage.

# References

[1] “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/>

[2] “How to Configure WIFI on ESP32.” *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>

[3] 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>

[4] *IEEE Xplore*. <https://ieeexplore.ieee.org/Xplore/home.jsp>

[5] 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/>

# Appendix A Requirement and Verification Table

|  |  |  |
| --- | --- | --- |
| **Table 1: System Requirements and Verifications** | |  |
| Requirement | Verification | Verification status  (Y or N) |
| Sensor Subsystem:  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% | Y |
| Sensor Subsystem:  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 | Y |
| Sensor Subsystem:  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 | Y |
| Control Subsystem:  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 | Y |
| Control Subsystem:  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 | Y |
| Speaker Subsystem:  Connect to phone with bluetooth capabilities | * Use Bluetooth to get ensure speaker is playing correct song * Check if ESP32 can get information from the web server and find the correct song that is currently playing | Y |
| Speaker Subsystem:  Be able to change/pause/play songs and have updates change the Web Server | * Make sure the web server updates when a new song is played * Ensure that the changes on the web server can be read by the ESP32 as well * If this does not work, recheck code for the web server and Spotify API | Y |
| Power Subsystem:  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 | Y |
| Power Subsystem:  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 | Y |
| Display Subsystem:  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 | Y |