BATHROOM INTERACTIVE ENTERTAINMENT SYSTEM

By

Jie Wen

Jing Huang

Zeping Lu

Final Report for ECE 445, Senior Design, Spring 2022

TA: Uma Lath

May 2022

Team No.60

Abstract

For people who treat showers as the best time of their day, a bad song could easily ruin their mood. However, changing a song using a wet hand would only cause worse scenes with electric hazards. We design and build a terminal device that interacts with a wristband. Our product solves the issue by using a wristband that when activated, it will record the user's hand gestures and skip songs as needed. The terminal that is installed by the bathtub wall will also serve as a Bluetooth speaker with integrated mood light so the user doesn't need to carry many devices to enjoy their bath time.

There are not many comparable IoT (Internet of Things) technologies to interact with household appliances effectively according to the user's will in the market. We have more incentives to improve the project to develop with rapidly advancing technology of DID (decentralized identifiers). This will facilitate the house access while keeping privacy with PKI (public key infrastructure) identified.

Contents

1. Introduction

In this project, we aim to facilitate user experience in the bathroom entertainment process. Nowadays, it is a disaster for people who try to take their best time of the day in their bathroom to reach their cell phone during shower. In this report, we will break down the design and implementation process of our project, gesture controlled bathroom entertainment system, power subsystem, output terminal subsystem, user terminal subsystem, and the Bluetooth bypass single channel.

We sought to construct the system through the use of a master-slave HC-04 Bluetooth module allocated on both user interface and the output terminal, a WS2812b LED strip right around the bounding box, a microcontroller, a speaker, an inertial measurement unit MPU9250, and a LCD screen IPS STT7735.

Our major purpose is to enhance user interaction nimbleness with the bathroom and develop the future smart devices for all household appliances. This device to some extent guarantees user safety while enjoying the best time of the day.

2. Design

In this part, we are going to break down the general design into design procedures and design details of each subsystem. The proposed design concept consists of an output subsystem, a user interface, a controller subsystem, and a waterproof outlet. The output subsystem will react to the orders from the user interface through the controller module.

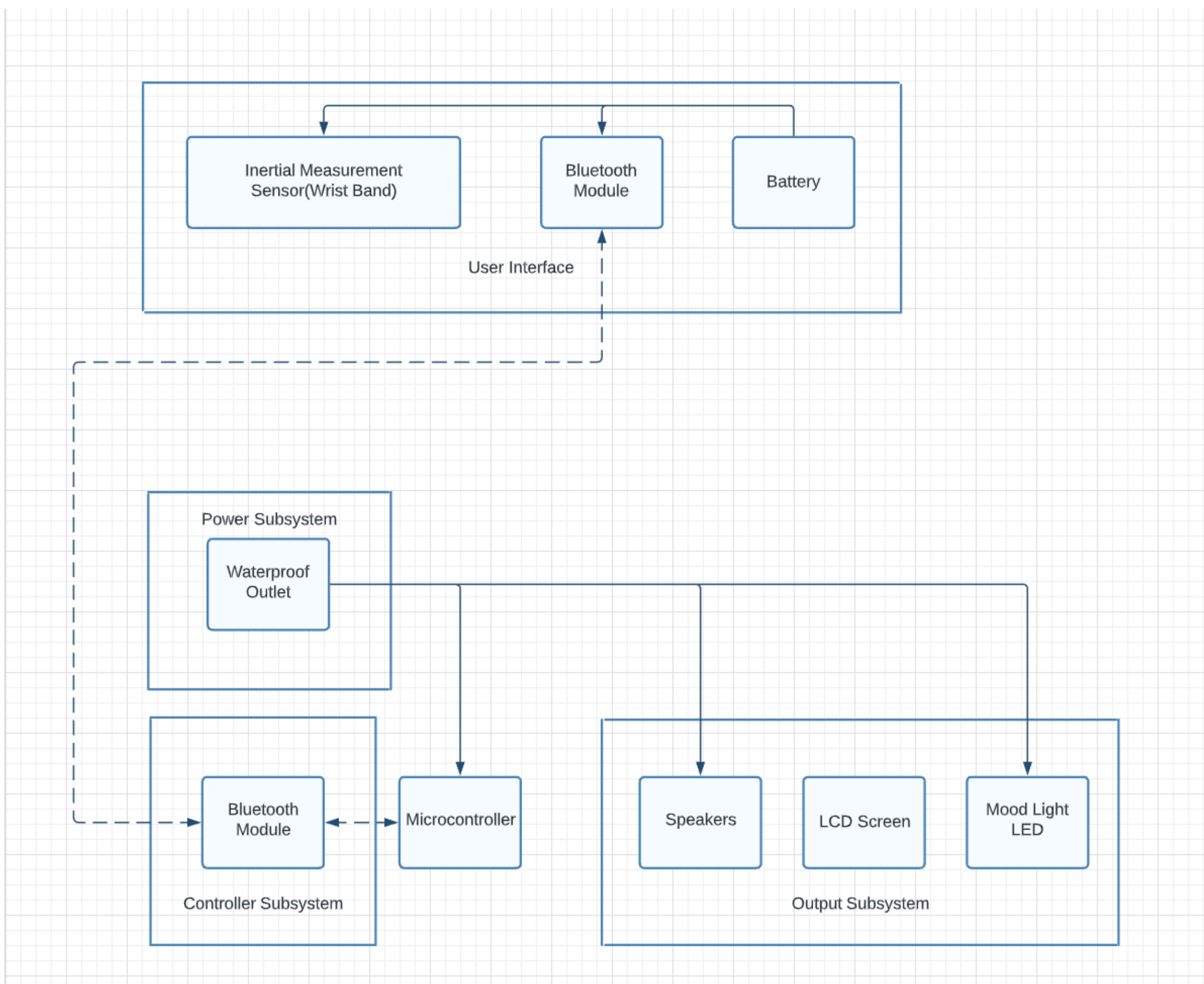

Figure 1 Block Diagram for the entire system

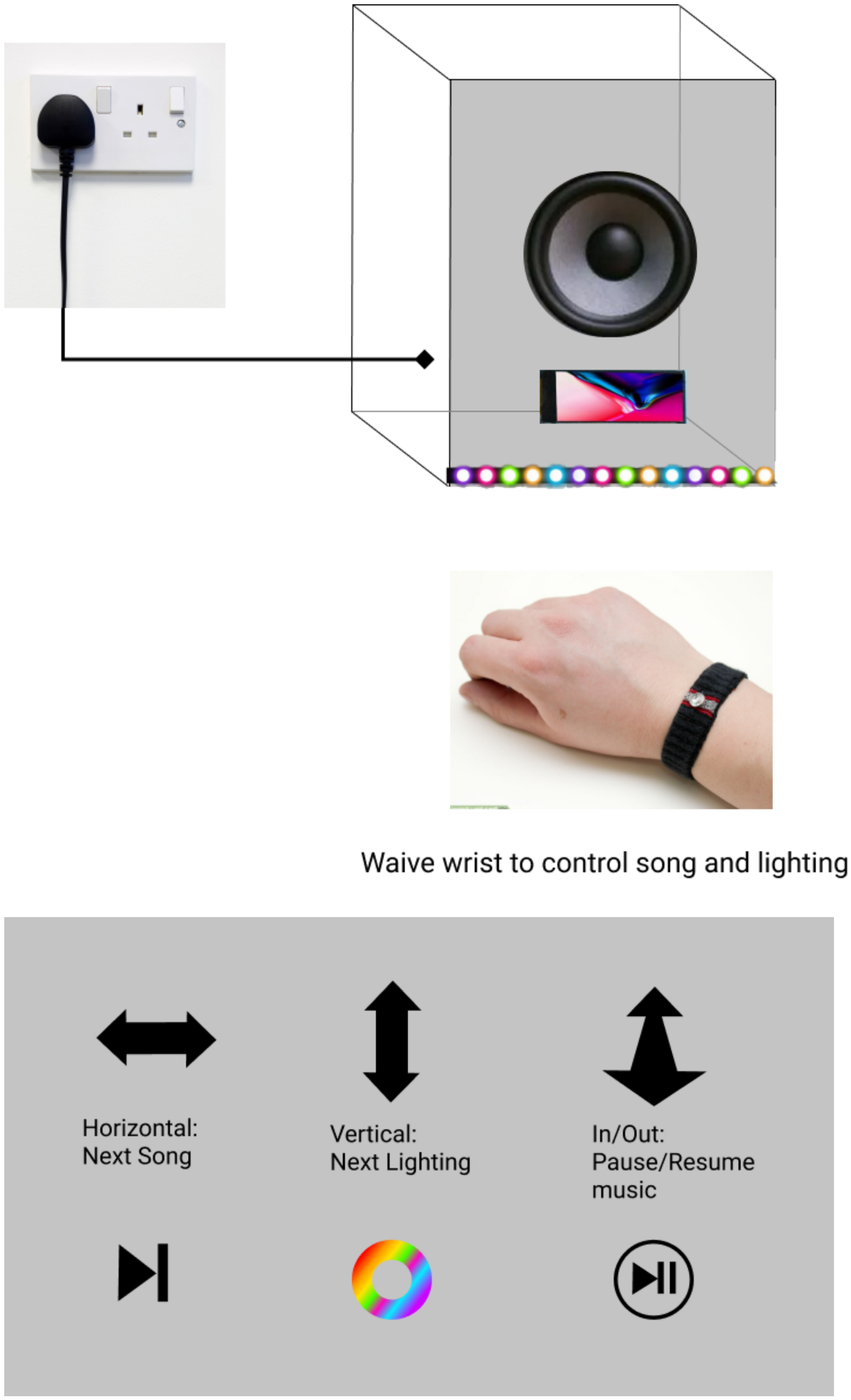

Figure 2 The concepts and outlook

2.1 [Output subsystem]

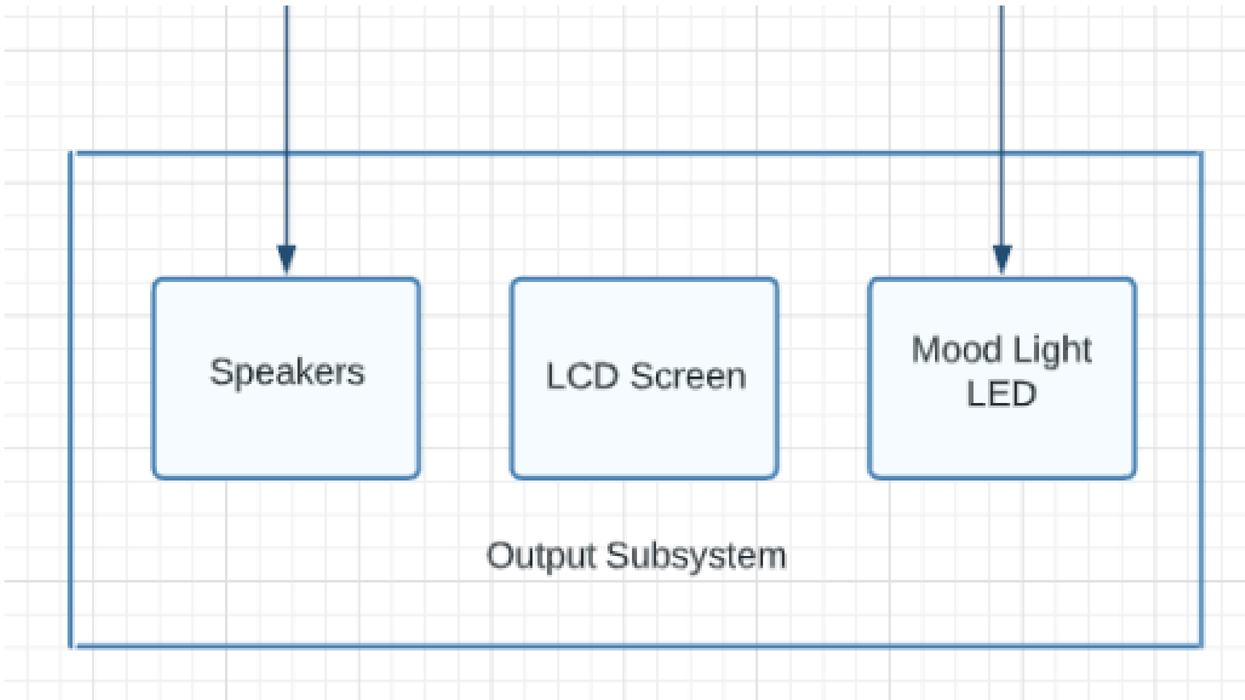

Figure 3 Output subsystem constitution

The output subsystem is capable of reacting to the user interface within a range of 3 meters. For basic functions, the speakers can play, shuffle or pause the music whenever the terminal recognizes a signal sent over through the user interface, i.e. wristband. The LCD screen will be able to demonstrate the command received from the wristband, and it will help along the debugging process. Using that information, the mood light LED will be able to display the specific colorway outside the bounding box, which surrounds the output subsystem, leaving alone a window for LED. In practice, this will indicate the exact light mode we preloaded into the microcontroller. Under different commands, there will be different sets of light and music on the list as preset and personal settings.

In order to accomplish this goal, several aspects must be considered. First, the LCD screen has to be chosen intricately so that the dot matrix of the display can present the information within limited space at low cost. Thus a representative HD44780 with standard controller chip LCD display. For 16 characters per line on 2 lines, there are 1280 dots to manipulate. It might be unrealistic to bring one separate control line for each dot for connection to a microcontroller. In addition, the method of powering the LCD board must be carefully considered so that the trade-off between a program with high capacity and propagation delay within the running time is compromised.

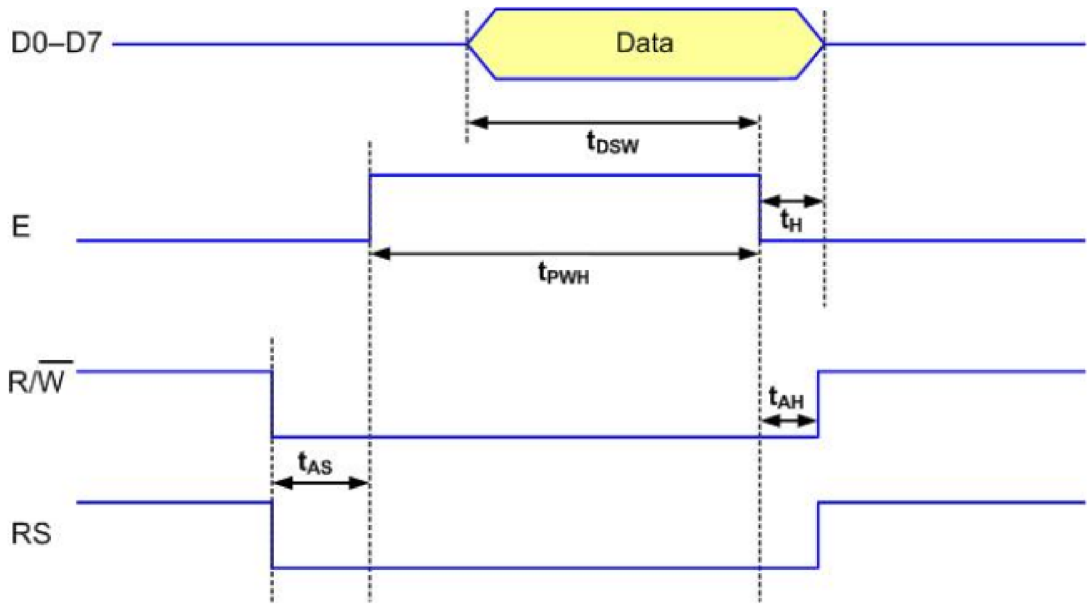

Figure 4 LCD write propagation delay diagram

We carefully reviewed the propagation diagram and decided to initialize the LCD display with dribble output, i.e. output with 4bits load per port. We package the dribble code package into a header file and optimize the code to respond to each line of printout after a full cycle of around one second. In order to achieve the responsive, accurate matchup, we set a delay for each read and write of two lines for three seconds. To make sanity checks, we will add some prevention of detection failure. We present an "error mode", which will be triggered when a wrong command has been detected due to weak transfer, switch bounce and poor memory capacity of the microchip.

VSS	VDD	V0	RS	RW	E	DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7	LEDA	LEDK
GND	5V	potentiometer	P3.6	GND	P3.7	/	/	/	/	P4.0	P4.1	P4.2	P4.3	5V	GND

Figure 5 tabulate the port layout of the board

2.2 [User Interface]

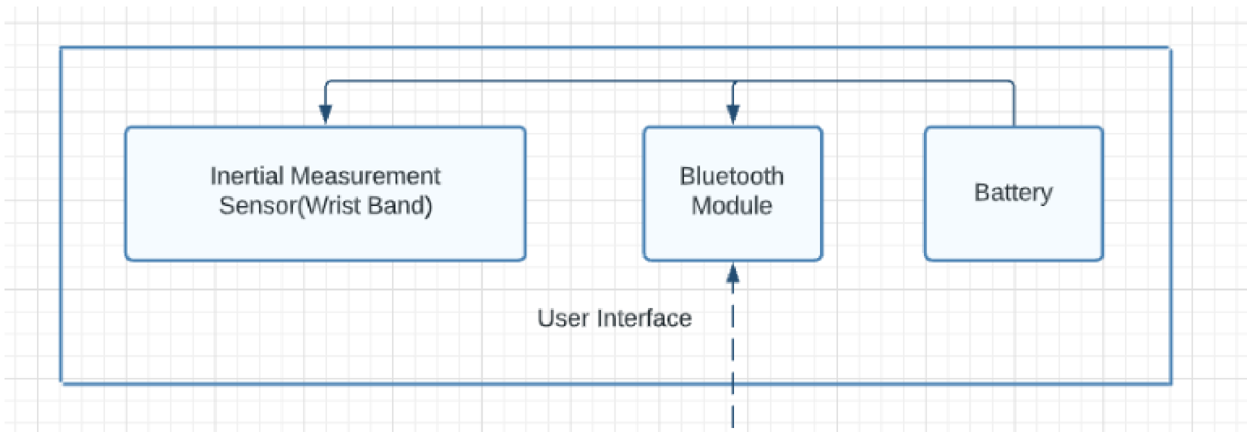

Figure 6 User Interface (wristband) constitution diagram

The user interface, wristband, is the other key portion of the entire system. The proposed design for the user interface consists of a powered microcontroller, which is capable of detecting, capturing and sending the gesture from user's end to terminal. The information will be able to accurately process through the microcontroller with the acceleration, gyroscope and magnetometers attached. Using that information, we picked an ESP32-PCIO microcontroller synchronized with RTC as the core, and an Inertial measurement unit MPU 9250. The node microcontroller is operating under 3.3V logic, lower energy cost as a remote device than ATMega line, more flash memory and compatible libraries online.

Meanwhile, we need to fit in all the materials with an exquisite bounding box. To achieve this, the IMU MPU9250, a much smaller size and more library functions than its comparable LSM9DS1 fits more of our demands.

Unfortunately, during the delivery period, we are unable to get the PCB designed and sent, and thus have to continue with an Arduino as a showcase.

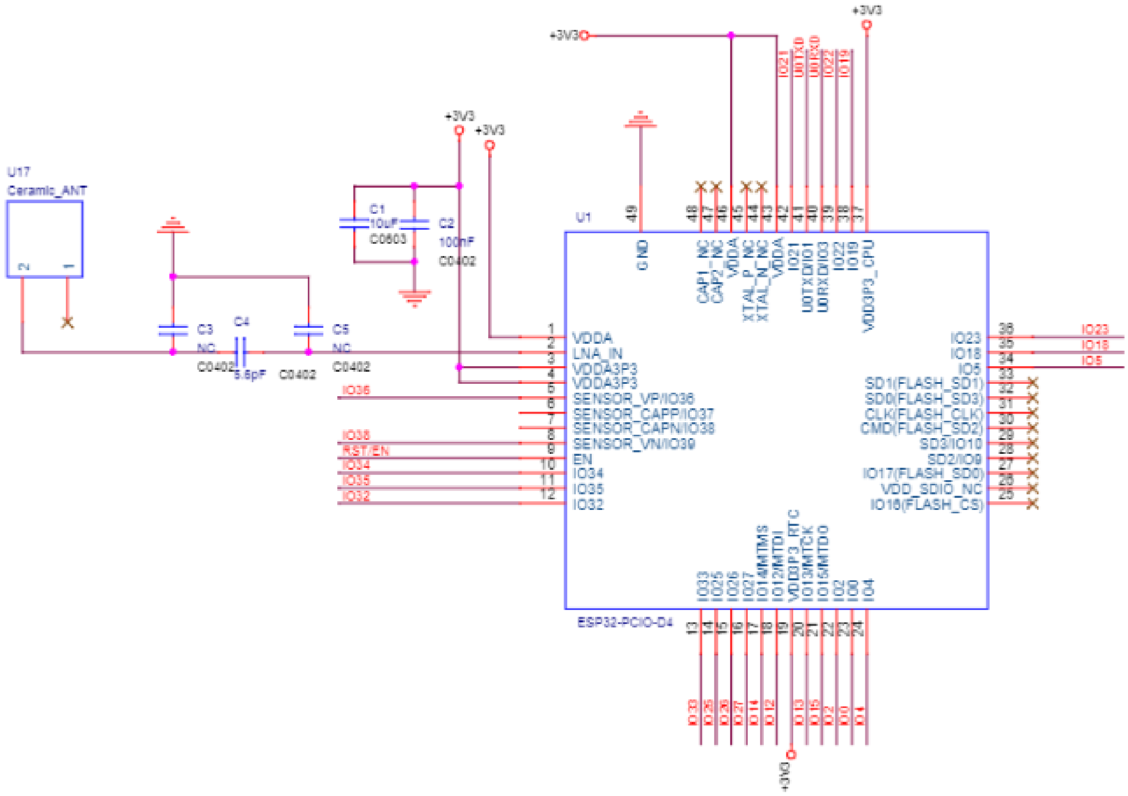

Figure 7 schematics for ESP32-PCIO microchip

2.3[Controller Subsystem]

The controller subsystem transfers the data carried in between the user interface and output terminal, while linking the speaker to the cell phone for music playlist. To transfer both music and gesture signal, the controller system has to be of two-channel bi-pass transferring module, consisting of either both Bluetooth and WIFI or two independent data transferring protocols.

Realizing the failure to carry out printed circuit board for wristband, we use single master-slave Bluetooth module HC-05 to conduct the task. HC-05 is the most popular module in the marketplace, and is easy to set up with AT mode.

3. Design Verification

We have designed several tests to verify our subsystem design. Then as a complete prototype, we tested the functionality of our design and checked if our high level requirements have been met.

3.1 [User Interface]

For our user interface, we connected our IMU with our microcontroller to first verify the functionality of our components. With a working library, our IMU successfully prints the current acceleration, gyroscope and magnetometer. However, the data being reported only reflects the current magnitude of the acceleration, which doesn't satisfy our initial design of directional gesture. Noticing the lack of data, we planned to calculate the location of the wristband by performing a series of integration with our recorded data. The lack of memory and processing speed of the microcontroller stopped us from continuing with this idea. Therefore, we decided to change our high level requirements of having directional gestures, instead we used horizontal, vertical and into the page movement to detect the user's gesture.

By setting a threshold of the acceleration of each axis, we can accurately detect if the user waived to any of the three axes that we are measuring and prints out the corresponding command that we wish to send to the terminal. We consider this as a successful working subsystem.

3.2 [Terminal]

The desired functionality of our terminal is to change lighting color, change song playing and pause/resume the music when corresponding commands are received. We first connected lighting strips, speaker and microcontroller to test our components. With our preset program, the lighting strips would change color to the RGB value that we set, the speaker would play tone with our tone set and the microcontroller would print out the current status of each component. When different values of RGB and tone are entered through the serial port of our microcontroller, our output components would correspond to them correctly.

We then tested our LCD screen with a set of default display messages. As we mentioned above, we need to use a set of delay inside our microcontroller to manipulate the display window of our LCD. We gave up the function of changing the brightness and contrast of the LCD to save extra ports on our microcontroller for bluetooth connection. The only conflict we noticed is when the LED strips are displaying a bright color, the current isn't sufficient enough for our LCD to display at maximum brightness. Since our microcontroller only outputs 5V voltage, to avoid this conflict, we changed the light setting so that we would only display a dimmer light effect.

3.3 [Bluetooth Connection]

Our bluetooth unit was first connected to the microcontroller to test if it allows any serial port communication with another paired device. Using the available library, we built a stable connection between an android device with the bluetooth module. To bind the bluetooth module into master and slave mode with our user interface bluetooth module, we enter the AT command mode of the device. By setting the address and baud rate, we successfully connected our two subsystems and it would automatically connect when powered on, meaning a success master and slave mode connection.

4. Costs

4.1 Parts

Table 1 Parts Costs

Part	Manufacturer	Cost ($)	Quantity	Total Cost ($)
ATmega32U4	Microchip Technology	$9.78	1	$9.78
MPU9250	TDK InvenSense	$6.99	1	$6.99
HC-05	Envistia	$7.49	2	$14.98
Arduino Uno	Arduino	$24.95	1	$24.95
WS2812B	WORLDSEMI	$8.54	1	$8.54
AST-03008MR-R	PUI AUDIO	$3.00	1	$3.00
NHD-0416BZ-FL-YBW	Newhaven Display	$21.42	1	$21.42
1571294-3 (Push button)	TE Connectivity ALCOSWITCH Switches	$1.00	1	$1.00
Total				**$90.66**

4.2 Labor

Labor cost for making this project from scratch:

3 people * 10 weeks * (15 hours/week) * ($30/hour) = **$13500**

Labor cost for manufacturing this device with the knowledge we have after this class:

3 people * 4 hours * ($30/hour) = **$360**

5. Conclusion

5.1 Accomplishments

We accomplished most of our high level requirements. The wristband would react when the button is pressed and store data within 3 seconds for the microcontroller to determine current command. The command is then sent through bluetooth to our terminal so that it would react correctly, i.e display correct messages on LCD, display music, and change light color.

5.2 Uncertainties

The master and slave mode of our bluetooth module becomes unstable when a long time has gone without powering up. We would have to reenter the AT mode to bind the two units, despite the status of the module being shown connected based on the built in led flashing.

5.3 Ethical considerations

- Contribution to society: Our project aims to make people's lives easier. Therefore, we believe this project contributes to society in a way that it simplifies lives and makes people's shower experience more enjoyable.

- Avoidance of harm: This device is designed to be used in a bathroom setting during shower, therefore it is crucial to be waterproof. Not being able to do so will not only result in failure of the project, but also arouse serious electricity safety concerns.

- Legal and Discrimination concerns: There is no legal or discrimination concern related to our project. Our project doesn't break any existing law; our project serves an audience of all gender, age, and racial groups.

5.4 Future work

There is still a long way to go to find out the final solution for an intelligent user interface that can carry our information securely and interact with the smart household appliances effectively. We will update the wristband with a predesigned dual-channel transferring module of Bluetooth and WIFI channels, or set up a new multi-channel method for simultaneous Bluetooth connection. Then we need to explore more of the decentralized identifier (DID) established on decentralized networks, nodes and the public key infrastructure (PKI) identification. All the smart household devices need to update accordingly to establish an entire ecosystem for a decentralized, secure, convenient life manner.

References

[1] *HD44780 (LCD) Datasheet,* Hitachi Semiconductor,Inc. Available at:

HD44780 Datasheet(PDF) - Hitachi Semiconductor (alldatasheet.com)

[2] *How to solve problem AT+INQ ERROR (1F), AT+INIT ERROR(17) on Bluetooth Module HC-05?*. Available at:

https://arduino.stackexchange.com/questions/50974/how-to-solve-problem-atinq-error-1f-atinit-error17-on-bluetooth-module-hc

[3] *How To Configure and Pair Two HC-05 Bluetooth Modules as Master and Slave | AT Commands*. Available at:

https://howtomechatronics.com/tutorials/arduino/how-to-configure-pair-two-hc-05-bluetooth-module-master-slave-commands/

[4] *How To Control WS2812B Individually Addressable LEDs using Arduino*. Available at:

https://howtomechatronics.com/tutorials/arduino/how-to-control-ws2812b-individually-addressable-leds-using-arduino/

[5] *MPU9250 (IMU),* datasheet, InvenSense, Inc., 2016. Available at:

PS-MPU-9250A-01-v1.1.pdf (tdk.com)

[6] *Play a Melody using the tone() function,* Available at:

https://docs.arduino.cc/built-in-examples/digital/toneMelody

[7] *Pro Micro & Fio V3 Hookup Guide*. Available at:

https://learn.sparkfun.com/tutorials/pro-micro--fio-v3-hookup-guide/all

[8] *Setting up Bluetooth HC-05 with Arduino*. Available at:

https://exploreembedded.com/wiki/Setting_up_Bluetooth_HC-05_with_Arduino

Appendix A Requirement and Verification Table

System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. User Interface a. Coin battery provides 3V DC voltage b. The accelerometer consumes around 0.1 mA c. Operated when humidity is 50% d. The accelerometer sends correct x,y,z coordinates when moving e. Bluetooth module connects with the terminal bluetooth receiver f. Data can be transferred through bluetooth to the terminal under 0.3s	**1. Verification** a. Use a voltmeter to determine the voltage of a coin battery and use datasheet to verify total capacity b. Measure the consumption of the accelerometer when powered on using a ammeter and calculate the total usage time from the battery capacity c. Power on the accelerator and move it while connected to a computer by testing different movements, must be able to give correct x,y,z axis data. d. Test the bluetooth module by connect it with the bluetooth module on the terminal and send different signals e. Measure the time required for the terminal to receive the data sent.	1a. N, the Bluetooth doesn't work with a coin battery 1b. Y 1c. N, we didn't have an environment to do the humidity test 1d. Y 1e. Y 1f. Y
2. User Interface a. The output system needs to be waterproof. b. The speaker operates under 8ohm and 0.4watt. It should have an output sound of maximum 80 of decibels. c. The LCD screen is able to display the song that is currently playing and show the current command being executed.	**2. Verification** a. Increase the humidity of the environment and make sure everything is still functioning with a humidity of 50%. b. Measure the output volume using a decibel examiner. If the speaker itself can't reach 80 decibels, an amplifier is required. c. The LCD screen should update within 0.5 seconds after a command is executed.	2a. N, we didn't have an environment to do the humidity test 2b. N, the tests were not conducted 2c. Y

d. The Mood LED should be able to perform any color with coded RGB values.	d. Input different values of RGB in the software and compare the output of the LED unit with the expected color. The colors have to be exactly matched (with human eyes) in order to pass the testing.	2d. Y
3. Power a. The power system provides 110V +/- 0.5% from a 110V AC source. b. An AC to DC converter is required to limit the power to +/- 5% of the total required power.	**3. Verification** a. Measure the voltage using an oscilloscope, ensuring that the output voltage stays within 5% of 110V. b. The total required power still requires calculation. But we will use an oscilloscope to ensure the final voltage stays within 5% of the required total power.	3a. Y 3b. Y
4. Controller Subsystem a. The controller is able to receive the signal from the wristband. b. The controller is able to process the received signal and control the output subsystem. c. Bluetooth module can be connected to both user's phone and the wristband and able to identify the input source	**4. Verification** a. Verify that the microcontroller received the x,y,z coordinates and is able to distinguish between movements of four directions. b. Manually verify that the correct operation is performed by the output system.	4a. Y 4b. Y 4c. N, the terminal can't connect to the phone and the wristband at the same time

Appendix B Code Appendix

Please find code in github with the following link: https://github.com/jingl3ling/ECE445project