Lab Tool Kit Data gathering for Scientist, Lake Analysis

Ali Ahmed Al Khusaibi, Zeid Haddadin, Homero Rodriguez, Ross Springstead, Mina Younes

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — Floating Lab Tool Kit, dedicated to the conservation and understanding of ponds and lakes worldwide. With a significant global count of 307 million lakes, as determined by an international team of scientists, the criticality of these water bodies becomes apparent. Our team focuses on the development of a compact floating lab capable of gathering essential data. Equipped with sensors for parameters such as pH, temperature, and air quality, our Lab Tool Kit enables researchers and scientists to conduct on-site analysis.**

I. Introduction

Understanding and monitoring the environmental status of lakes and ponds need efficient and reliable data collecting. In this project, we offer the Lab Tool Kit, an innovative floating data collection technology designed to enable real-time monitoring of static bodies of water. The Lab Tool Kit revolutionizes the area of environmental data collecting and analysis by integrating cutting-edge technology, mobility, and powerful sensors, delivering essential information into the dynamic behavior of aquatic ecosystems. The Lab Tool Kit is an all-in-one solution for collecting real-time data from lakes and ponds. The platform's fundamental component is a floating lab equipped with a variety of sensors and equipment. Real-time pH sensors, for example, enable continuous monitoring of water acidity levels, allowing researchers to understand changes in the health of water.

The Lab Tool Kit also includes a data storage and transmission system that enables smooth data transfer to cloud-based platforms, allowing for quick access to acquired data for analysis and monitoring. The Lab Tool Kit provides a unique perspective and the opportunity to collect data from various areas by traversing lakes and ponds, resulting in a more thorough understanding of environmental conditions.

The floating Lab Tool Kit, which combines innovative sensor technology with extensive data processing capabilities, provides researchers with an unparalleled chance to monitor and analyze critical environmental data. This comprehensive method enables scientists to anticipate possible hazards, investigate aquatic species behavior patterns, and devise conservation and protection plans. Finally, the floating Lab Tool Kit promises to transform our understanding of water ecosystems and contribute to the long-term sustainability of ponds and lakes for the benefit of both humans and the environment.

II. Design Detail

Fig. 1. Design of the the Lab Tool Kit

The design in the figure above is of the floating platform for the Lab Tool Kit on a surfboard. The image illustrates the integration of an RC boat motor and controllers, connected to the backend bottom of the surfboard. The system will be mounted on top of the floating platform with sensors, nets, and a solar panel incorporated into the design.

The design of the floating Lab Tool Kit on a surfboard is a remarkable combination of innovation and practicality. By connecting an RC boat to the backend bottom of a surfboard, researchers gain the ability to control the device remotely using a remote controller. This design enables precise navigation and maneuverability, facilitating targeted data collection in various water environments.

The Lab Tool Kit boasts a comprehensive set of sensors, including water temperature, pH, air temperature and humidity. These sensors provide real-time measurements, allowing for the monitoring and analysis of key environmental parameters. By capturing accurate data, researchers can gain insights into water quality, track changes over time, and identify potential environmental hazards.

The inclusion of a solar panel on the top surface of the Lab Tool Kit demonstrates a commitment to sustainability. The solar panel harnesses the power of the sun, converting sunlight into renewable energy to power the device's systems and sensors. This eco-friendly power source ensures continuous operation, reducing reliance on non-renewable energy sources and minimizing the environmental footprint of the project.

To facilitate bug collection, the floating Lab Tool Kit features two nets positioned on the sides of the device to capture bugs on the water's surface, as well as an additional net placed on top to collect flying bugs. This design allows researchers to study insect populations, their behavior, and their impact on the ecosystem, contributing to a comprehensive understanding of the aquatic environment.

Overall, the design of the floating Lab Tool Kit on a surfboard showcases the integration of advanced technology and sustainable practices. With its remote control capabilities, sensor data object, solar panel, and bug collection nets, this innovative design provides researchers with a versatile and efficient platform for environmental monitoring and research.

III. Software Detail

In this section, we will look into details about the software side of our Lab Tool Kit project.

A. *UI design of the website*

When it comes to the software side of our project, we can start by talking about the user interface of our website, which was designed using both HTML and CSS.

The increasing importance of the internet and digital platforms has made websites a crucial element for businesses, organizations, and individuals. As websites serve as the virtual face of an entity, creating a positive user experience has become a top priority. User interface (UI) design plays a pivotal role in achieving this objective, as it directly impacts how users interact with a website. A well-crafted UI helps by delivering a positive user experience, enhancing user engagement, and achieving business goals by keeping the user happy. User interface design refers to the process of crafting the visual and interactive elements of a website or application. It involves creating a user-friendly and visually appealing interface that allows users to interact seamlessly with the digital product. If the UI is not well crafted, users might not use the designed product and turn to a competitor with a better UI that makes the user's experience better. The components of UI design include layout, navigation, typography, color scheme, images, forms, and other interactive elements. Multiple elements contribute to making the visual appeal of the UI to the user as best as possible; a well-defined visual hierarchy helps in bringing attention to certain areas that may be more important than others. Readability is another element, which can be achieved by making sure the website is legible with the selected color schemes.

For our website, we approached the design with a "dark theme" in mind, meaning the background is black, and the elements present on the website should be dark oriented. For instance, some input fields have a gray background to fit in the dark theme. A dark theme was selected because of its aesthetic appeal, improved readability, reduced eye strain, and more.

The layout of the pages on our website was another important aspect to consider in the design of our UI. The placement and size of some elements like the title of the page, buttons, and more is crucial to make the UI as best as possible. Working on improving the layout takes multiple iterations as the first one is not usually the best design. We began with coding the first iteration. Then, we asked every member of the team for feedback on what they would improve to make their experience as best as possible. Seeking feedback and input from each member allows for diverse perspectives and ideas, leading to innovative solutions and improvements in the UI design. With the collected feedback, we improved the design and repeated the feedback process. The layout design went through multiple iterations that way until we were all satisfied with the result achieved.

The whole process of design can be represented by figure 2 below, which describes how after a design is implemented, feedback is collected to improve the next iteration of design. Additionally, after each iteration of design, every element of our website was tested to make sure that everything is working like intended.

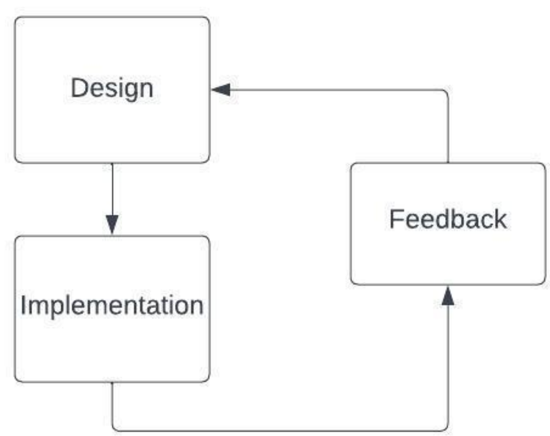

Fig. 2. Design process of the website with multiple iterations

B. Hardware programming using software

The hardware used in this project like the ATmega328 microcontroller as well as the NodeMCU dev board were programmed using Arduino IDE.

Arduino has become a popular platform for prototyping and developing projects involving microcontrollers and development boards. The Arduino IDE (Integrated Development Environment) provides a user-friendly interface that simplifies the programming process and facilitates the creation of diverse applications. Arduino IDE was used to program the ATmega328 microcontroller and the NodeMCU development board. It provides a user-friendly interface, a code editor, and a compilation and uploading mechanism, making it accessible to beginners and experienced developers alike. Our software team consists of 2 computer engineers who did not have any prior experience of coding using this IDE. However, thanks to this IDE being accessible to beginners, both coders were able to successfully code the ATmega328 as well as the NodeMCU to complete the wanted tasks (which will be talked about in this section). To program the ATmega328 microcontroller using Arduino IDE, the necessary hardware connections must be established. This involves connecting the microcontroller to the computer via a USB-to-serial adapter or Arduino board. Additionally, the appropriate drivers may need to be installed to enable communication between the microcontroller and the IDE. Arduino IDE provides support for programming the NodeMCU board through the ESP8266 core. The core adds the required libraries and tools to enable communication with the board. Once the codes are written for both the ATmega328 as well as the NodeMCU, they can be compiled and uploaded to the ATmega328 and the NodeMCU board using the "Upload" button in the IDE.

Moving on to what the ATmega328 microcontroller was programmed to do using Arduino IDE. The ATmega328 microcontroller plays a pivotal role as the brain of many projects, and it does in our project as well. We programmed the ATmega328, specifically focusing on integrating various sensors into the project and sending it as a JSON object to the NodeMCU. By leveraging Arduino IDE's features and libraries, our team effectively used its libraries as well as the programming language C++ to be able to collect the readings of the various sensors included in our project, making the ATmega328 a versatile and powerful controller. The advantage of using those libraries is that it offers sample code for many popular sensors, which when tweaked for our project, simplified the integration process.

Moving on to what the NodeMCU board was programmed to do using Arduino IDE and the programming language C++. Arduino IDE allows for seamless integration of the NodeMCU board, which features built-in Wi-Fi capabilities. We have leveraged Arduino IDE's libraries and functions to establish a connection between the NodeMCU and a Wi-Fi network present around us. This involves configuring the network credentials and using the appropriate library functions to establish a successful WiFi connection. In addition to the WiFi connection feature, the NodeMCU board was programmed to collect sensor readings from the ATmega328 using serial communication, and it was programmed to send the collected data as a JSON object to a database at specified intervals. This approach allows for seamless data transfer and opens up opportunities for real-time monitoring and analysis since our website will display those objects stored in the database to the user on the main page. Thanks to Arduino IDE, we were able to facilitate the creation of JSON (JavaScript Object Notation) objects using libraries like ArduinoJson. An HTTPS (Hypertext Transfer Protocol Secure) connection is then established between the NodeMCU connected to WiFi and our database (managed by MongoDB). Using HTTPS over HTTP ensures data security by encrypting the sent data so that it remains confidential and secure during transmission.

In other words, Arduino IDE and the programming language C++ were used to make the PCB containing the ATmega328 microcontroller collect sensors data, send the data as a JSON object to the NodeMCU connected to WiFi or hotspot, and have the NodeMCU send the JSON object to MongoDB using a secure HTTPS request.

This process can be represented by figure 3. Additionally, the data transmission process using JSON will be explained in more detail in the section below.

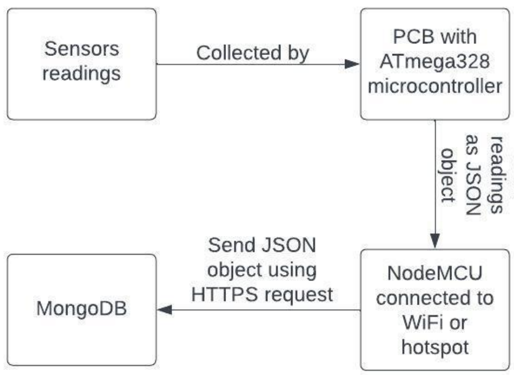

Fig. 3. Summary of software usage for both the PCB containing ATmega328 and the NodeMCU board

C. Software Stack

For our project we designed a full stack web application. This full stack web application was included for displaying the results of our sensor data to the end user. A full stack web application is the end-to-end development of applications. It includes both the front end and back end of an application.

With our project the front end is accessed by a client who is intending to use our Lab Tool Kit project. And is what the user sees. When designing our front end we decided to use the React framework. React is an open-source JavaScript library developed by Facebook that has gained significant popularity in the web development community due to its efficiency, scalability, and component-based architecture. With its virtual DOM implementation, React optimizes the rendering process by efficiently updating and rendering only the necessary components, resulting in improved performance and a seamless user experience. React's declarative syntax allows developers to describe the desired UI state, making it easier to understand and maintain code, while also facilitating code reusability through the creation of reusable UI components.

To send our data to our front end we have to take advantage of API's. APIs which stand for Application Programming Interfaces are sets of rules and protocols that allow different software applications to communicate and interact with each other. They serve as intermediaries, enabling developers to access and utilize the functionality and data of other applications or services. API's provide a standardized way for applications to request and exchange information, empowering developers to integrate various services, extend functionalities, and create innovative applications that leverage the capabilities of other systems With our project we use REST APIs (Representational State Transfer Application Programming Interfaces) which are a set of architectural principles used for designing web services that follow a client-server model. They utilize the HTTP protocol and standard HTTP methods (GET, POST, PUT, DELETE) to enable communication and data exchange between client applications and server resources in a stateless manner.. With our project we wrote API's to connect the front end to our MongoDB and we used GET, POST, and PUT methods. We are also using two outside API's in our project. The first with our use of SendGrid to send emails to our clients. And with our second one being with Google Charts to create a chart with our pond data.

The back end forms the core of the application where the business logic is applied. Back end development involves the creation and maintenance of the server-side components of a web application. It encompasses tasks such as designing the application's architecture, implementing server logic, managing databases, and integrating with external services. With our project we decided to use a version of Node.js called Next.js.

Next.js is a popular React framework used for building server-side rendered and statically generated web applications. It provides a powerful development environment with features like hot code reloading, automatic routing, and server-side rendering out of the box. Next.js optimizes web performance by generating static HTML files that can be served from a CDN (content delivery network), resulting in faster page loads. It supports serverless functions, allowing developers to create API endpoints directly within their Next.js applications. With its strong ecosystem and seamless integration with React, Next.js is a go-to choice for building modern and scalable web applications.

A benefit of using Next.js as our web development framework is that deploying our code to our domain was easy. With their built in service Vercel, all that was needed was a copy of our github repository and a list of the dependencies and our web application was live.

Since we used both React and Next.js all of our coding for our web application was done in JavaScript. In our project, JavaScript played a crucial role in both the front end and back end development. On the front end, JavaScript was essential for implementing interactivity and dynamic behavior, allowing us to manipulate HTML elements and handle user interactions. We leveraged JavaScript's capabilities to create a responsive and intuitive user interface, enhancing the overall user experience. Additionally, on the back end, we utilized JavaScript with technologies such as Node.js, specifically the Next.js framework. This combination allowed us to build a powerful and scalable server-side environment, enabling seamless integration with databases like MongoDB and the creation of API endpoints to handle data transmission and retrieval. JavaScript's versatility and extensive ecosystem contributed significantly to the success of our full-stack web application.

MongoDB is a leading NoSQL database management system that has gained widespread adoption in the industry. It provides a flexible and scalable solution for handling large volumes of unstructured and semi-structured data. Unlike traditional relational databases, MongoDB follows a document-oriented data model, where data is stored in flexible JSON-like documents, allowing for dynamic schema design and easy integration with modern application development frameworks.Another notable feature of MongoDB is its support for flexible data modeling. With its schema-less design, MongoDB allows for the storage of documents with varying structures and fields. This flexibility is particularly beneficial for agile development environments where data requirements evolve rapidly. MongoDB's document model also enables seamless integration with object-oriented programming languages, simplifying the development process and reducing the impedance mismatch between application code and the database.

In our implementation of MongoDB we had one cluster with three separate collections. Where one collection was for our user data, one for tokens, and one for our pond data.

For transmitting our data we used JSON. JSON is a lightweight data interchange format commonly used for transmitting and storing structured data. It consists of key-value pairs enclosed in curly braces and supports various data types such as strings, numbers, arrays, and objects. JSON is widely adopted due to its simplicity, readability, and compatibility with multiple programming languages. We use JSON objects in each aspect of data transmission for our project. It first starts when the data is being sensed in the pond. After our sensors have read the data we create key value pairs for the humidity, pH, air temperature, and water temperature with the respected data we collected. Once this data has been collected we serialize the JSON into one object and send it to the NodeMCU. Since we are using MongoDB as our database we are able to send JSON directly into our database with an HTTPS Post request to a data endpoint. This works by taking advantage of MongoDB's App Service feature which allows us to create a function that is called inside of our database to post the JSON object with the inclusion of a timestamp.. And once this data has been successfully transmitted from the sensors to the database, a simple GET API can be used to access it to display on our web application.

The features of our webstack include a multiple of things. We have a signup and login system that creates accounts for new users and manages existing users. An email is also sent out when a new account is created. The user also has the ability to reset their password if they forgot it by sending a token to the user's email where they then can prove their ownership of the account and reset their password. Once logged in users can add a Toolkit to their account by navigating to the add toolkit page. The pond data is displayed in table form showing the time stamp and all the sensor data points. The user then has the ability to search for specific data points with the search bar and drop down menus. The user also has the ability to update their password while signed in and to sign out of their account. We also have included authentication features to make sure that to access the data pages you are required to be signed in. Also, we have a charts page which allows the user to graph whichever data point they want onto the X and Y axis. And lastly we also have hidden easter eggs that can be accessed in different ways.

IV. Hardware Design

This section describes how the device's different hardware components work together to form a functional Lab Tool Kit

A. *Overall Design*

When it comes to the overall design of the hardware of our project, it can be represented by the following figure.

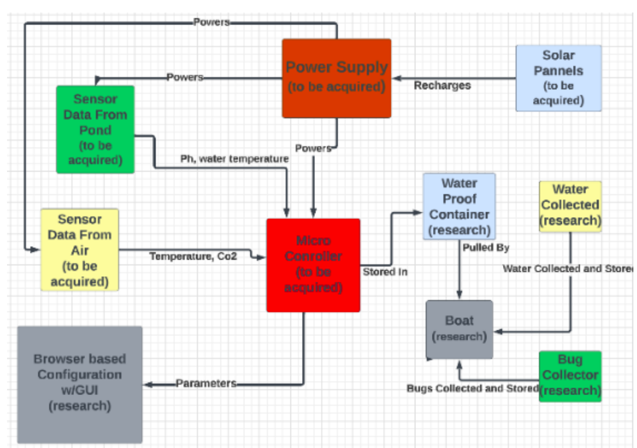

Fig. 4. Block diagram of all hardware components (shows a hardware schematic with visible connections between the components)

B. PCB Design

The PCB provides electrical support for the toolbox on the floating lab. Multiple websites were utilized to create the circuit board and its layout. JLCPCB printed it after that. The circuit board had two revisions. The first attempt was to change the layout and make a smaller board. The second edition was created after the initial board was tested and certain electrical difficulties with the material composition and layout of the board were discovered.

C. Testing

Each component was examined independently for hardware testing to confirm correct operation and to get information regarding the project's power needs. Each PCB was checked for maximum voltage and current capabilities once it arrived to ensure that all components could function simultaneously.

V. Integration

The integration of the different parts was an essential phase in the development process. When specific components were incompatible and required to be updated or replaced, the integration phase frequently took longer than the initial development phase.

A. Software Integration

The software end-to-end was originally tested on a local host before being transferred to the cloud computing engine and tested on multiple devices. To avoid limiting data transmission consumption, effort was made to ensure that the WiFi was responsive and could be linked on many different devices.

B. Hardware Integration

The PCB was designed with hardware integration in mind. Hardware integration could not be accomplished without this component. This caused problems due to the multiple delays and changes that were made to the PCB. The summer semester is about 4 to 5 weeks shorter than a normal semester, which is especially challenging for hardware design considering time delays in PCB fabrication and shipping. The system was proven to work perfectly as expected from design to breadboard and then to the soldered board.

C. Software and Hardware Integration

Integration of software and hardware did not come without challenges. The hardware required needs to interact with the software side in order to correctly combine these two pieces. This was accomplished by connecting the hardware to the database over WiFi.

D. System Testing

Both the hardware and software parts of the device were extensively tested before and after integration. Constant voltage tests were performed on the hardware during the PCB and component integration to confirm that the PCB was functioning properly and that it was giving the correct current and voltage to the various components. A problem was observed while uploading the code to PCB in the USB communication module which prevented it from reading the sensors, but the system was proven to work extensively on breadboard and also in the soldered board that is currently running the whole project.

For the software testing, extensive tests were done to the website. Extreme scenarios, such as inaccurate data, were tested to verify that the website would not crash. Both the functioning and the aesthetics were tested. Assuring that the website received data accurately and that the functionalities and components flowed logically.

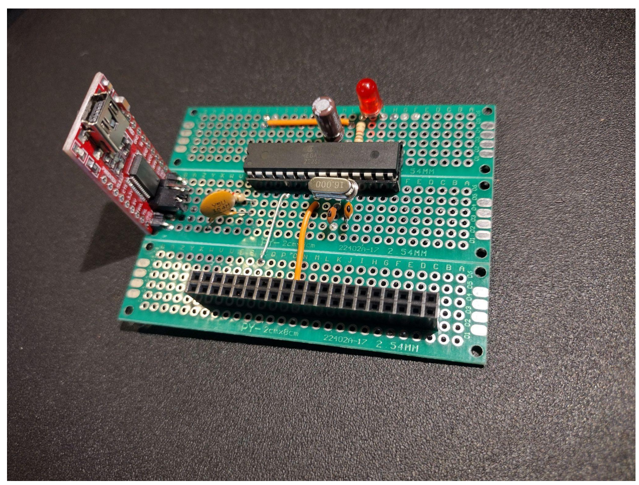

Fig. 5. Soldered breadboard circuit

The soldered board shown above comprises the whole network of ATmega MCU and USB connection. It was successfully tested multiple times, showing that it is able to manage the whole project from the hardware perspective. It is able to receive code to MCU, read all the sensors, send data packages to ESP MCU and finally have the data sent to the database which was successfully displayed for users as it was expected. The project worked properly from this soldered board which is a design that would work as well when implemented in PCB.

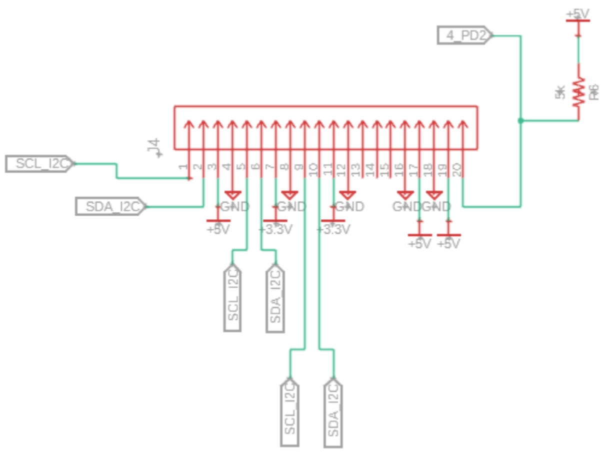

Fig. 6. Specialized pin socket for sensors

The schematic shown above is a specialized pin socket that provides power, ground and ports for two I2C sensors and 1 One Wire communication sensor. It can be observed that the pins for SCL and SDA are internally connected and there is space for debugging clock and data with the oscilloscope. The 5KΩ pull-up resistor was mounted in the PCB according to the specific requirements of the 1 wire communication sensor. This molecule would make the PCB more compact, efficient and specialized for the task of managing specific sensors.

The PCB main functionality is to control and gather the required data from the onboard sensors. The PCB consists of ATmega328P-PU, which is the microcontroller responsible for all the main functionality in the PCB. Also, the PCB has a UART interface to allow the user to program the ATmega328P-PU using the USB Mini-B cable. The operating voltage for the microcontroller on the PCB is 5V, so it was necessary for our application to add an onboard liner voltage regulator.

The linear voltage regulator in the PCB uses two capacitors 47uF at the input and 10uF at the output.

The ATmega328P-PU microcontroller is an 8-bit microcontroller from the AVR family. It offers a range of features and functionality that make it suitable for our project.

The most common type of oscillator used with the ATmega328P-PU is a crystal oscillator. A crystal oscillator consists of a crystal resonator connected to the microcontroller, providing a stable and accurate frequency reference. The ATmega328P-PU uses a 16 MHz crystal oscillator, which allows for precise timing of instructions and ensures reliable operation of the microcontroller.

The 22pF capacitors connected to the 16MHz crystal oscillator are an integral part of the crystal oscillator circuit used with the microcontroller ATmega328P-PU. These capacitors serve an important purpose in ensuring the stable and accurate operation of the crystal oscillator. Some of the roles of these capacitors include decoupling capacitors, which play a crucial role in the design and functionality of a custom PCB. These capacitors are specifically used to stabilize the power supply and ensure reliable and accurate performance of the microcontroller and other integrated circuits on the board. The capacitors added on the PCB are responsible for voltage stabilization, noise filtering, and component protection.

The FT232RL is a popular USB-to-serial converter chip manufactured by FTDI (Future Technology Devices International). It plays a crucial role in the PCB design, especially for projects involving communication with a computer or other USB-enabled devices.

Adding the FT232RL module to the PCB is necessary to be able to program the ATmega328P-PU. The FT232RL has two pins TX_FT and RX_FT. The transmitting and receiving pins of the FT232RL are connected to receiving and transmitting pins of the microcontroller ATmega328P-PU RX and TX.

The other functionalities of the FT232RL USB module consist of the following:

1- USB Connectivity: The FT232RL chip enables USB connectivity for the PCB. It provides a bridge between the UART (Universal Asynchronous Receiver-Transmitter) interface of the microcontroller and the USB interface of a computer or USB-enabled device. This allows the PCB to communicate and exchange data with a host device using the USB protocol.

2- Serial Communication: The FT232RL chip is primarily used to convert the serial data format used by microcontrollers (UART) into a format compatible with USB. It handles the conversion of the data between parallel and serial interfaces, ensuring seamless communication between the PCB and the host device. It supports various standard baud rates and data formats, making it versatile for different communication requirements.

3- Driver Support: FTDI provides driver support for the FT232RL chip across different operating systems including Windows. We found and installed the appropriate drivers for our device, allowing it to recognize and communicate with our PCB. The availability of reliable and well-maintained drivers simplifies the integration process and ensures broad compatibility.

4- Plug-and-Play Functionality: The FT232RL chip supports plug-and-play functionality, which means that the user can connect their PCB to a computer or USB-enabled device. It is automatically detected and configured without the need for manual driver installation or configuration. This feature enhances user convenience and eliminates the hassle of driver compatibility issues.

5- UART Interface Flexibility: The FT232RL chip provides a versatile UART interface that can be easily integrated with different microcontrollers and other devices. It supports various UART voltage levels, allowing compatibility with a wide range of microcontroller families. This flexibility makes it suitable for the PCB design, enabling seamless communication between the microcontroller and the host device.

V. Results

Throughout the testing phase of our project, we have achieved remarkable results in fulfilling our engineering specifications. The system consistently measures the water's pH at a minutely interval which would be scaled to hourly in actual production. Enabling us to closely monitor and analyze water quality trends. Additionally, our system effectively collects bugs present in the water, providing valuable insights into aquatic ecosystems and potential ecological impacts. Moreover, the remote controller functionality has proven to be highly responsive, allowing researchers to navigate the lab with exceptional precision and agility. These accomplished specifications underscore the success of our project and highlight its potential to revolutionize environmental monitoring and contribute to a better understanding of our natural world.

VI. Conclusion

In conclusion, the Lab Tool Kit project has been a remarkable journey of innovation, collaboration, and environmental stewardship. Through the diligent pursuit of our engineering specifications, we have successfully developed a cutting-edge platform that enables real-time monitoring of water parameters, bug collection, and remote-controlled mobility. The results obtained during the testing phase demonstrate the feasibility and effectiveness of our system in addressing the challenges of environmental monitoring and hazard detection. This project represents a significant step towards a more sustainable future, where technology and scientific inquiry intersect to protect our precious water resources. As we move forward, the Floating Autonomous Biological Control Lab holds tremendous potential for expanding our knowledge, fostering ecological conservation, and inspiring further advancements in environmental research and control strategies. We are excited to witness the positive impact that this project will have on both scientific understanding and the preservation of our planet's delicate ecosystems.

Acknowledgement

The authors wish to acknowledge the assistance of ChatGPT in writing this conference paper by introducing relevant information about the project to its AI input.

Additionally, the authors would like to thank the professors who kindly agreed to review our Senior Design project.

References

OpenAI. (2023). ChatGPT (May 24 version) [Large language model]. https://chat.openai.com/chat