Group 8: IoT Based Smart Aquarium

Aryan Kumar Naveen Kumar Mathur Amit Kumar Singh
Prince Kumar Ahirwar Aman Raw
190185, 190535, 190117, 190646, 190108
{aryankmr, naveenkm, amitsgh, pprince, amanraw}@iitk.ac.in
Indian Institute of Technology Kanpur (IIT Kanpur)

Abstract

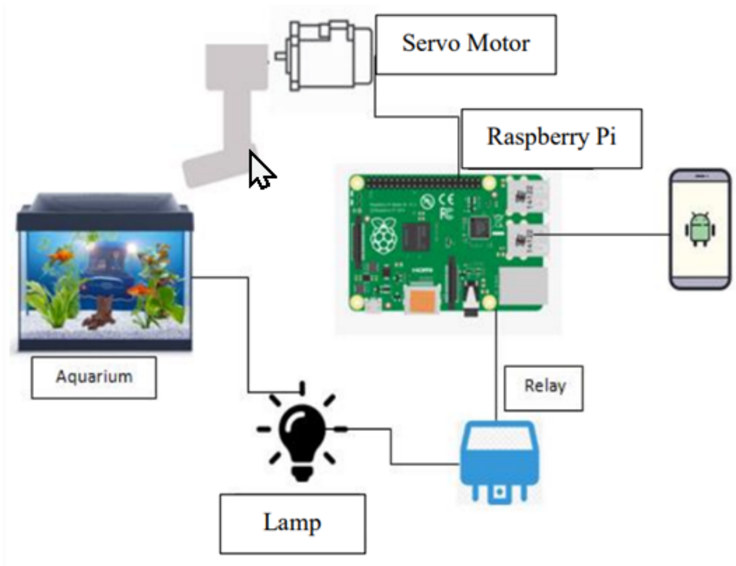
Nowadays, most pet owners choose to keep a fish as a pet. However, they need a significant amount of aquarium water care and maintenance. We presented an IoT-based Smart Aquarium Monitoring System in this research paper to provide a healthy fish living environment. We used a pH sensor and a light sensor to guarantee that the fish had enough light and a suitable pH for survival. The aquarium's condition was determined using sensors (LDR, pH). Methods from Python's socket library are used to create a connection between the Raspberry Pi as a server and another PC as a client. Depending on the sensor results, the system will automatically adjust the brightness of the aquarium and drain and replenish the water through a solenoid valve. The system may also be monitored via the web application that we developed. This system may also be developed to protect the health of marine life in bigger bodies of water.

1 Introduction

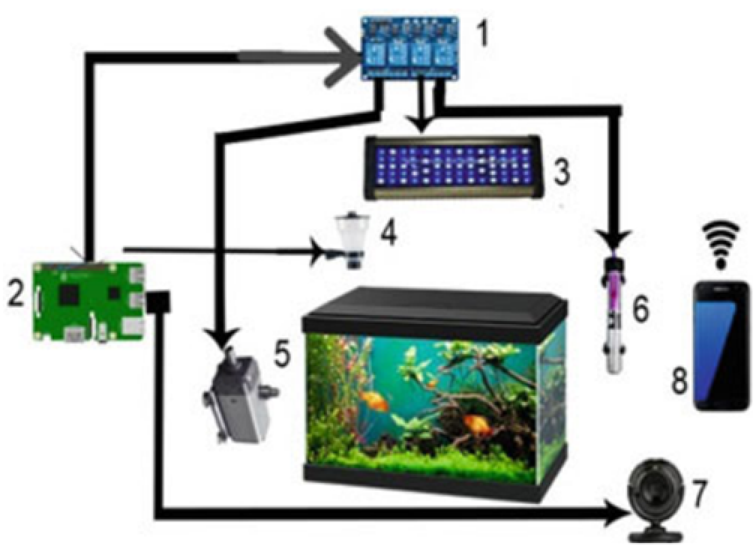
Having an aquarium might bring charm to the house, and there is a lot of fondness for keeping fish in the aquarium. But, it is equally tedious to maintain water and environment quality in the aquarium as per the requirements of the fish. There is a great deal of managing the aquarium's light, temperature, and water quality. And with time constraints, it is hard for aquarium owners to monitor and maintain the aquarium in real time.
In this report, we will walk you through the literature review we performed before beginning with the project as well as we will briefly discuss our project and the results we have achieved. We'll discuss the shortcomings and the future works which could further be implemented on the project.

2 Related Work

1. In this paper [1], a smart aquarium device was designed to feed aquaculture fish automatically; namely, Smart Aquarium Design Using Android-Based Raspberry Pi. It used a servo motor to drive the fish feeding valve and also uses a relay as an on/off aquarium decorative light. Here, Android is the interface between the aquarium and the tools. Through this paper, we learned about the function of the relay and its connections with the Raspberry Pi board.



2. Paper [2] proposed technology uses a Raspberry pi Webcam server to capture live video streams of an aquarium, feed fishes through servomotor using pulse-width modulation, relays are used to ON/OFF light, temperature control, filter control, and android app to allow the user to access various functionalities. We will not incorporate the webcam or filter control features because they are already implemented, and we won't add anything new and significant to the existing project.



3. In paper[3], we learned about the optimal freshwater temperature range for fish survival is 25-32 degrees Celsius. The temperature affects the digestive process and metabolism of the fish. The maximum turbidity that is safe for fish is 128 NTU. If the turbidity level of the water is more than the maximum value, it is necessary to drain the pond. This system uses Cayenne to design a software system. Cayenne is a third-party software-based Android provides application system. The software provides a graph of real-time water temperature measurement. Also, here, one of the solenoid valves works to drain half of the height of aquarium water. After that, another water solenoid valve opens to give access the clean water to refill the aquarium water.

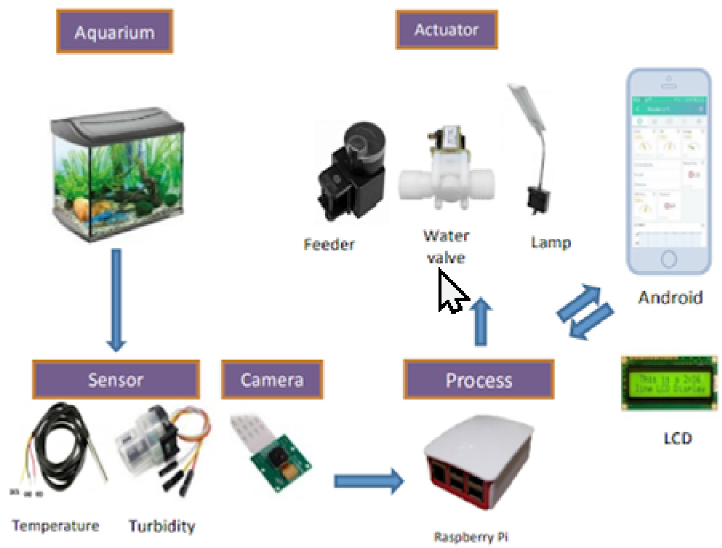
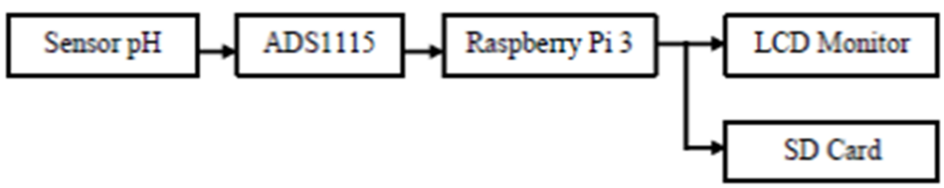


FIGURE 1. Hardware design

4. Here [4], we see ADS1115 functions as an analog-to-digital converter (ADC). This component is used to change the analog signal to digital so that it can be processed by the processor. Raspberry Pi functions as a processor, which in this device functions to process the signal processing from ADS1115. Raspberry Pi also functions as a GUI so that the pH reading can be displayed on an LCD monitor.



5. Here [5], the idea to use Socket Programming is implemented because it is an efficient way of communication for a client-server-based platform where we just need to export data as requested by the client. A socket is a communications connection point (endpoint) that you can name and address in a network. Socket programming shows how to use socket APIs to establish communication links between remote and local processes. Basically, the Raspberry Pi will act as a server and will provide the data collected by the sensors, and then we will establish a connection that will use UDP and TCP as the communication protocol between Raspberry pi and our pc over a local port and IP and after using the server's bind() method which binds it to a specific IP and port so that it can listen to incoming requests on that IP and port will exchange the data and continue further analysis.

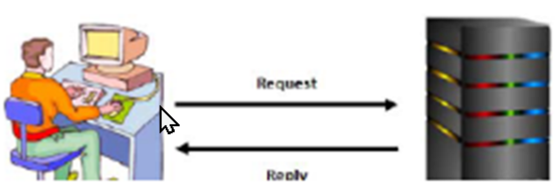
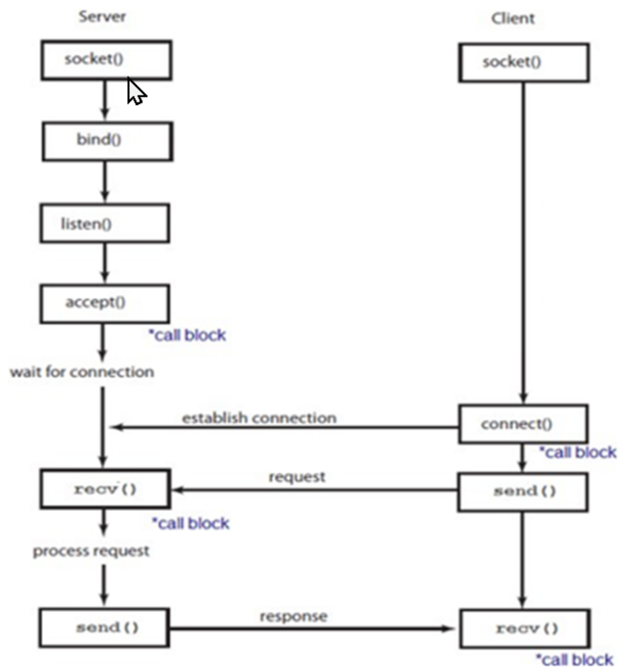


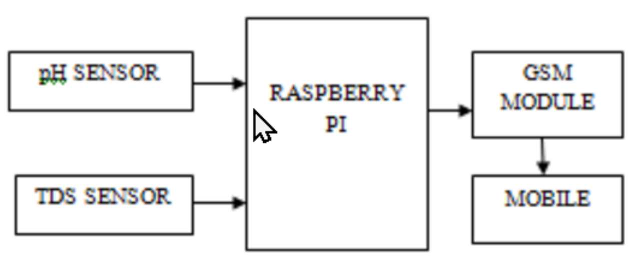
Fig 1: Client Server Communication

Fig 4: Established connections with the use of temporary ports assigned

6. This paper[6] uses tells us about the idea of using GUI with a web page/application.

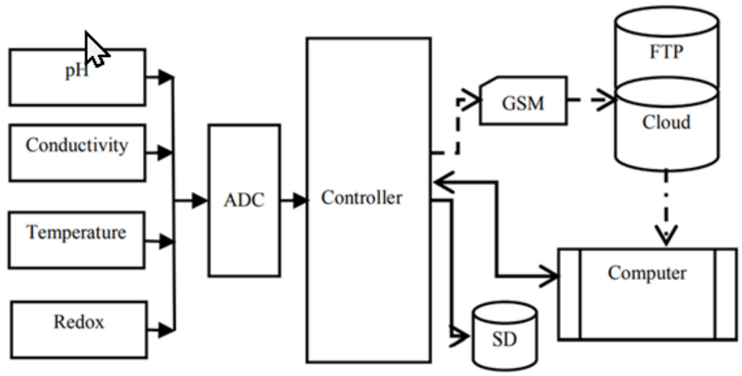


7. [7]The pH and TDS level in Aquarium are monitored and collected from corresponding sensors. pH sensor measures the activity of hydrogen ions in a solution, whereas TDS measures the concentration of Total Dissolved solids in parts per million (ppm). Here in this approach, Collected data is transferred to Raspberry Pi. It is programmed in such a way that if pH and TDS levels are ideal, the algorithm will stop. If it exceeds or falls below the ideal level, it sends an SMS alert to the user's mobile phone by using GSM Module. GSM (Global System for Mobile Communication) is a digital cellular technology used for transmitting mobile voice and data services. It uses the Time Division Multiple Access (TDMA) transmission method. It supports a data transfer speed of 9.6 Kbps allowing transmission of SMS.



8. The approach in the paper[8] considers that the parameters pH, oxidation-reduction potential

(ORP), conductivity, and temperature independently provide very little information in terms of how polluted the water is. Hence, the collective parameter is analyzed to generate a valid input, which is either polluted or not polluted. The system comprises sensors, an analog-to-digital converter (ADC), a microcontroller, an SD storage, and a GSM module. The experiment is performed on different water sources like tap water, seawater, surface water, and polluted water. The parameter references obtained from all the different water sources will be used to build classifiers which will be used to perform automated water analysis in the form of Neural Network Analysis.



9. We analyzed the solutions currently available for the implementation of smart aquariums. As we can see in the paper[9], there is the usage of apps that you can connect to using wifi inside homes to get the info and data gathered by the sensors. But, we mainly focused on extending accessibility from anywhere using socket programming's server-client-based communication. It will allow us to establish a server-client relationship to fetch data from the server hosted on the Raspberry Pi using a socket module in Python at a specific IP address and a port. We are also trying to make it more real-time by incorporating the data into a web page or app.

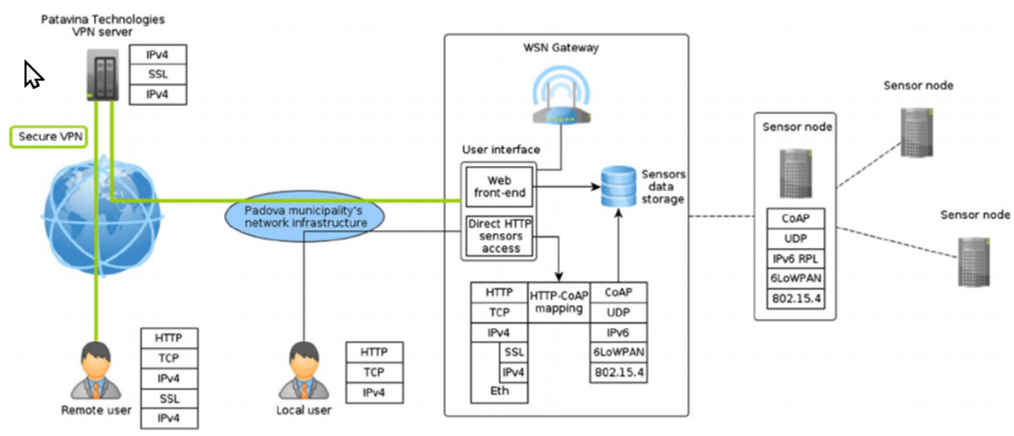


Fig. 3. System architecture of "Padova Smart City."

10. Here[10], the proposed system has measured TDS, Temperature, Turbidity, and pH values of water with the help of Raspberry Pi and various Sensors to make a cost and energy-efficient system. Here they used ThingSpeak API to store and retrieve data using the HTTP protocol over the Internet. ThingSpeak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates.

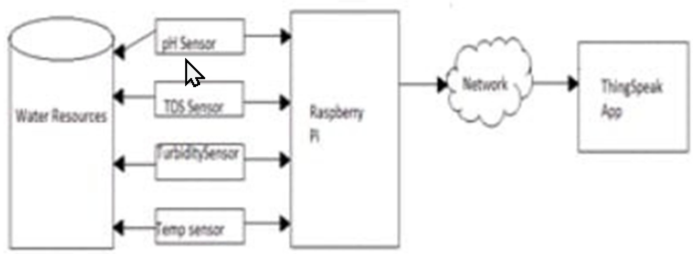


Fig-1: Block diagram proposed system

How will our approach be different?

1. We will use pH sensors to determine the pH level of the water and maintain the right pH level. The existing solutions are only monitoring the pH but not acting on unfavorable conditions automatically.
2. Brightness in the aquarium will be synchronized according to the surrounding brightness. These features are not pre-existing and are also significant.
3. We will use the Fill-or-Drain Water System approach to regulate the pH level of the aquarium water by using a solenoid valve.
4. We will use Socket programming for communication between Raspberry Pi and PC which is simple, and data can be retrieved from anywhere.
5. We will work on the encryption and decryption of the data to make it secure.
6. For the sake of simplicity, we will get the data displayed on the laptop rather than on the smartphone. This is because, for now, we are not planning to make any android app to display output data and control our IoT system.
7. In our project, we will only focus on monitoring pH because monitoring oxidation-reduction potential (ORP) and conductivity are not very necessary for water health monitoring in an aquarium. Even if we go for implement these monitoring parameters for high-precision readings from water, our solution would not be very cost-efficient and will take a lot of time to build it.

3 Proposed Idea

In our project, we intended to tackle the above problems mentioned in the introduction as well as problems with the pre-existing solutions so that owner doesn't have to worry about the maintenance of the aquarium. We will monitor the aquarium's water quality and the environment through our project IoT-based Smart Aquarium. pH level measurements will help us monitor and regulate water quality. There's a suitable range of water pH (approximately 6.5 to 9.0) at which fishes can thrive efficiently. The brightness in the fish tank is also crucial for the fish's health. The brightness readings will help us in the timely lumination of the tank. We have used an LDR sensor to detect the brightness level.

We have used Raspberry Pi 3 and Arduino controllers to collect the sensor data and send it to the local PC. We worked on the encryption and decryption of the data before sending it to ensure the security of our data. We have established the connection between Raspberry Pi as a server and another PC as a client using methods present in the Socket Library in Python.

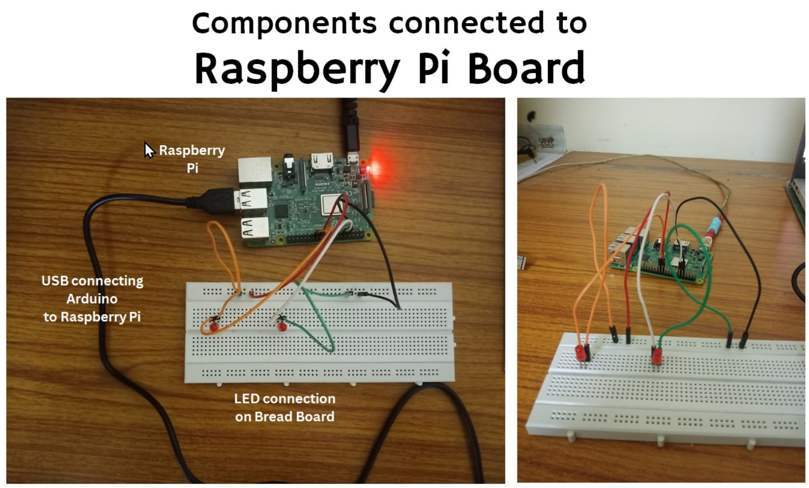
This sensor data in our PC terminal will decide which sort of actions should be taken. Our system will turn on or off the lights according to the light sensor values (light intensity in the room), and it will invoke the fill-drainage system if the pH level is alarming in the aquarium. Along with this, the real-time sensor data will be hosted on a web page so the owner can access it remotely.

4 Methodology

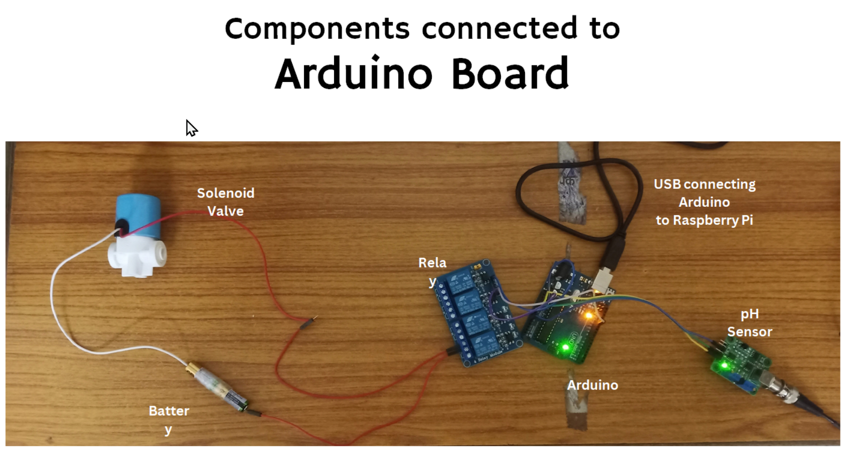
Hardware Requirements:

Raspberry Pi	Arduino	Relay
pH Sensor	Mobile Phone for LDR Sensor	Solenoid Valve
2 LEDs	Breadboard	Wires

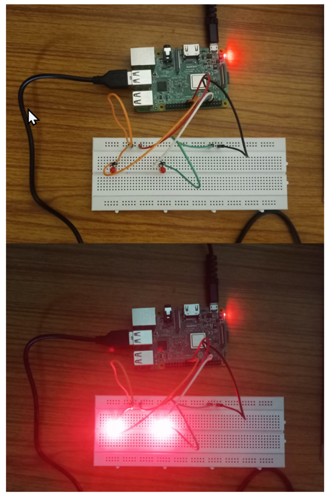
1. We started with the Raspberry Pi board setup and its connections. For setting up the Raspberry Pi, we used the steps illustrated on the official site. The following figure shows the connections:



2. Next, we proceeded with the Arduino Board connections. The following figure shows the connections:



3. Then, we carried out the LDR sensor connection. We used Light Sensor present in the Smart Phone for the project. We used PhonePI+ mobile app for streaming light values to Raspberry Pi through Flask-Socket communication. We connected 2 LEDs with Raspberry Pi, which automatically handles non-ideal light conditions in the aquarium surroundings. These 2 LEDs are connected to GPIO 17 (pin number 11) and GPIO 11 (pin number 23) of the Raspberry Pi. In a room, light values can vary between 0-100 lux (Illuminance). LEDs will turn ON for light values between 0-20 and turn OFF for more than 40.



4. Further, we proceeded with the pH Sensor.

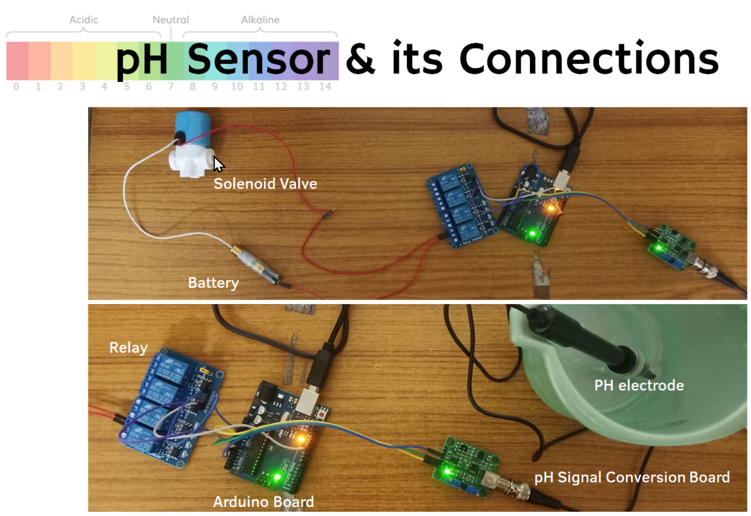
   Analog pH sensor is used to measure the pH value of the aquarium water and show the acidity or alkalinity of the water.

   The Nernst equation gives a relation between the cell potential of an electrochemical cell, temperature, reaction quotient and the standard cell potential. This equation is used to calculate the PH value of a solution as well. The glass electrode response is governed by the Nernst Equation can be given as:

$$E = E_0 - 2.3\frac{RT}{nF}lnQ$$

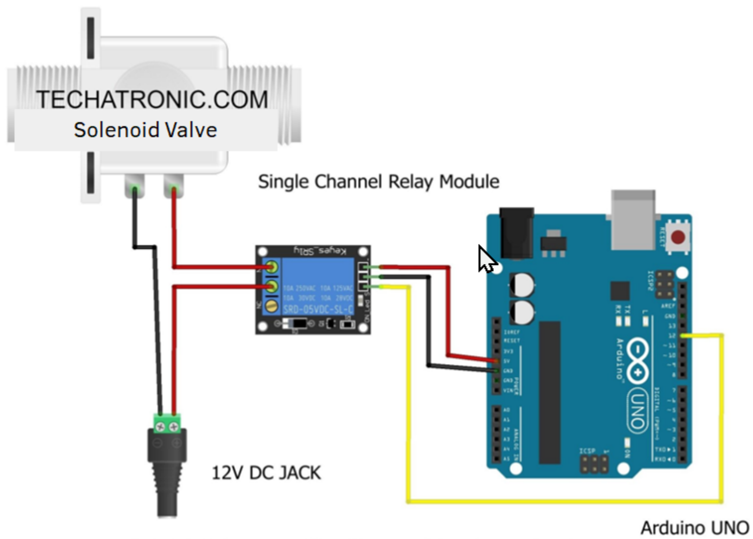
   Where:
   Q= Reaction coefficient
   E = mV output from the electrode
   E0 = Zero offset for the electrode
   R = Ideal gas constant= 8.314 J/mol-K
   T = Temperature in ^{o}K
   F = Faraday constant = 95,484.56 C/mol
   N = Ionic Charge

   We connected the pH sensor to the pH Signal Conversion Board which was then connected to the Arduino board to program it and get the readings.

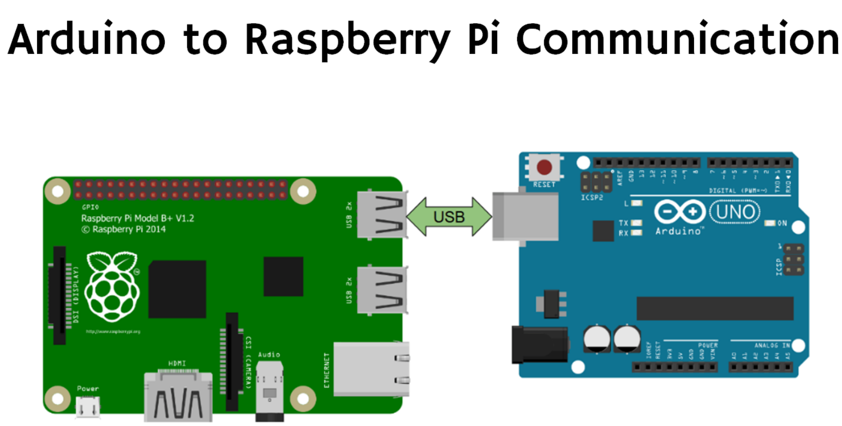


5. We attempt to regulate the pH of the aquarium water by adding or removing the water from it until we achieve the suitable pH range. Here, we used a solenoid valve, which is used for controlling the flow of any liquid. For this, we connected the solenoid valve with the Arduino board to which the pH sensor is already connected. Relay act as the middleman between the valve and the board. The connections are as follows:

   For non-ideal case: (pH>8 or pH<6.5), the relay will automatically turn ON the solenoid valve for 2 seconds.

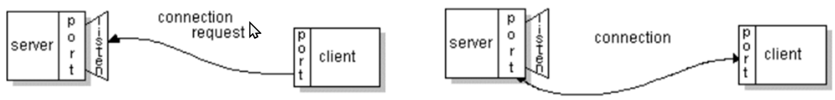


6. Then, we connected Raspberry Pi and Arduino Board with each other through Serial communication. On the Raspberry Pi side, we chose one of the 4 USB ports available on the board. For Arduino, we used the USB port that we used to upload code from your computer (with the Arduino IDE) to our board. We first connect your Arduino to our computer so that we can upload the code into the board. After that, we connected the USB cable to the Raspberry Pi. The Raspberry Pi will power the Arduino via this cable.

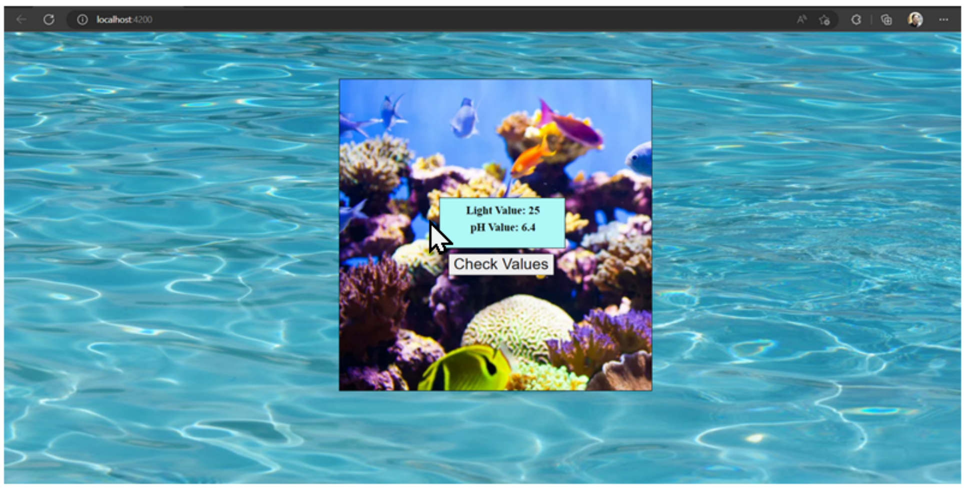


7. Then, the communication between Raspberry Pi and our PC was achieved through Socket Programming. Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while the other socket reaches out to the other to form a connection. The server forms the listener socket while the client reaches out to the server. Thes sockets work on two parameters. The first parameter refers to the address-family ipv4. and the other means connection-oriented TCP protocol. TCP

(Transmission Control Protocol) is a connection-oriented protocol, which means a connection is established and maintained until the applications at each end have finished exchanging messages.



8. Finally, we worked on the webpage through which we requested and received data on our PC. At the backend side (Node.js was used here), we created an API to fetch real-time sensor data from a local JSON file. On the frontend side (Angular was used here), we integrated the API to receive and display real-time sensor values on a button click.



9. Finally, we performed encryption and decryption of the data. While sending the JSON object, we just incremented the ASCII values, and while receiving, we decremented them back.
10. **Project Structure and Code description:**
   1) */Raspberry Pi code copy/requirements.txt* : Contains all the dependencies required for Raspberry Pi
   2) */Raspberry Pi code copy/sensor.py* : Receives real-time Light Sensor values from PhonePi+ app and pH values from Arduino, controls the LEDs, sends sensor data to Client through Socket, encrypts data before transmission
   3) */arduino code/ph/ph.ino* : Collects real-time pH values and sends it to the Raspberry Pi, controls the relay which turns ON/OFF the solenoid valve for water flow
   4) */socket/socket_communication.py* : Receives real-time sensor data from Raspberry Pi, decrypts the data and sends it to sensor_output.json
   5) */socket/sensor_output.json* : JSON containing light and pH values
   6) */backend/package.json* : Contains all the dependencies required for backend
   7) */backend/server.js* : API to fetch data from sensor_output.json
   8) */front end/package.json* : Contains all the dependencies required for frontend
   9) */front end/src/app/* : Contains files to integrate the API and display live data on a button click on a webpage.
   The files are present in this GitHub repository.
11. **Steps to Run:**
   1) Connect the Arduino (with all pH sensor and relay connections) to the PC using the USB cable. RUN and UPLOAD the */arduino code/ph/ph.ino* in the Arduino IDE. Make sure that it is printing the pH values in the Serial Monitor every 2 seconds. Remove the USB and connect it to the Raspberry Pi.
   2) Connect the Raspberry Pi (with all LED connections) to the PC using Type B cable. Connect

it to WiFi (The board used in this Project connects to a WiFi network with {Name: Rpi, Password:00000000}). Open the Raspberry Pi terminal and log in. (Login credentials of our board: {Username: pi, Password: iotians}).
3) Duplicate the files */Raspberry Pi code copy/requirements.txt* and */Raspberry Pi code copy/sensor.py* in any directory. (In our board, just navinate to */Desktop/Group*8 directory where files already exists).
4) Install all the dependencies using:

```
$ pip install -r requirements.txt
```

5) In the sensor.py enter Raspberry Pi IP and PC IP addresses in the variables server and client respectively.
6) Now in PC, in the */socket/socket_communication.py* enter PC IP address in the variable IP and run this file making it ready to receive data from Raspberry Pi. (Make sure to run it from the correct directory).

```
$ python3 socket_communication.py
```

7) In the Raspberry Pi, run the sensor.py. Now it waits for the Light Sensor in the PhonePi+ app to turn ON before printing and transmitting sensors data.

```
$ python3 sensor.py
```

8) Connect the mobile phone to the same network with which Raspberry Pi is connected. (In our case, WiFi network with {Name: Rpi, Password:00000000}). In the PhonePi+ app, enter the Raspberry Pi IP address, set frequency corresponding to LightSensor as 2000 and turn ON the light sensor. This will start streaming sensors' data on Raspberry Pi as well as sending it to the PC through socket communication. In the PC, this data is received and gets recorded in */socket/sensor_output.json* file in real-time.
9) Now, to view this data on the webpage, first start the backend by installing dependencies and running */backend/server.js* (Make sure to do it from the correct directory):

```
$ npm install
$ node server.js
```

10) Now start the frontend by navigating to the /front end/ directory, installing all dependencies and running:

```
$ npm install
$ npm start
```

11) Once its complete, open the browser, go to *http://localhost:4200/* and click the button "Check Values". This will display the light and the pH values.

5 Results

1. We successfully implemented brightness monitoring and adjustment features in our aquarium through LED lights. The ideal light range for the aquarium surroundings is taken to be 20 - 40 lumen. A normal 9W LED bulb can provide a light value of around 30-35 when the sensor is kept a few meters away from the bulb facing toward it. So we can use one or two 9 W bulbs. However, to avoid excess light and handle light conditions during both day and night, it's better to use more number of 3 W bulbs. The number of bulbs lighting up will depend on the current surrounding conditions, and thus, the light value will adjust itself within the ideal range in a better way.
2. We successfully implemented the pH monitoring and regulating feature in our Aquarium. There's a suitable range of water pH (approximately 6.5 to 9.0) at which fishes can thrive efficiently. Our solution helps in maintaining the pH within range. This is done by filling or draining water with the Solenoid Valve.
3. We successfully encrypted and decrypted the data. While sending the JSON object, we just incremented the ASCII values, and while receiving, we decremented them back.

4. All the data is displayed on the webpage.
5. Brightness and pH regulation are done automatically. Users do not have to perform these actions manually.

6 Discussion and Future Work

1. We can work on measuring the temperature of the aquarium water and regulate the temperature as well. Future work can cover this feature.
2. pH regulation can be more efficiently done. Currently, we are using one Solenoid Valve. For future work, we can use two Solenoid Valves to perform fill and drain functions simultaneously.
3. Not every user will be carrying their laptops with them all the time. Therefore, we can control and operate everything through Android App rather than PC in the extended version.

7 Conclusion

Having an aquarium may add elegance to the home, and many people like maintaining fish in aquariums. However, it is equally difficult to maintain the aquarium's water and ambient quality in accordance with the needs of the fish. There is a lot of management needed over the aquarium's light, temperature, water quality, and other parameters. Through our project, we monitor the pH level of the aquarium water, regulate it, and maintain it in the fish's favorable range. We also monitor the surrounding brightness and luminate the aquarium whenever it is dark. These functions are done automatically without any manual work. We also added the encryption feature, which will save valuable or expensive fish from any external attacks.

8 Individual Contributions

Members and contributions		
Name	Work	Contribution
Aryan Kumar	Worked on making all the presentations, report and helped with the research and connections.	20%
Amit Kumar Singh	Worked on connection and functioning of sensors, LED connections, serial communication between boards, socket communication, and web page.	20%
Naveen kumar Mathur	Worked on establishing connections with Rpi, encryption-decryption and socket programming.	20%
Prince Kumar Ahirwar	Working of solenoid valve and connections with Arduino device.	20%
Aman Raw	Worked on connections of raspberry pi and sending data from sensor to laptop.	20%

References

[1] Y. I. Khairunisa, Mardeni, "Smart aquarium design using raspberry pi and android based," in *Journal of Robotics and Control (JRC)*, vol. 2, pp. 368–372, 2021.

[2] S. Kori, S. Ayatti, V. Lalbeg, A. Angadi, S. Kori, S. Ayatti, and Lalbeg, *Smart Live Monitoring of Aquarium-An IoT Application.* 03 2020.

[3] Y. Afifah, C. H. Brillianto Apribowo, R. Rosadi, and M. Hafiz, "The smart monitoring and automation control system for fish aquarium based on internet of things technology," vol. 2097, p. 030018, 04 2019.

[4] E. P. F. D. Iswanto, Prisma Megantoro, "Ph meter based raspberry pi and user interface using python," in *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 9, 2019.

[5] R. L. Maata, R. Cordova, B. Sudramurthy, and A. Halibas, "Design and implementation of client-server based application using socket programming in a distributed computing environment," 12 2017.

[6] M. U. T. Amber Shah, Md Gulam Servar, "Realtime chat application using client-server architecture," in *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, vol. 10, 2022.

[7] T. Ramprakash, A. Mahalakshmi, M. Shamima, and T. Subhalakshmi, "Smart home automation for monitoring ph and tds level in aquarium and purifier," in *International Journal of Engineering Research and Technology (IJERT)*, vol. 6, 2018.

[8] A.N.Prasad, K. A. M. Kabir, F. R. Islam, and H. Haqva, "Smart water quality monitoring system," 12 2015.

[9] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *Internet of Things Journal, IEEE*, vol. 1, 01 2012.

[10] A. T, Bhaktia, C. R, and P. D, "Iot based low cost system for monitoring of water quality in real," 05 2018.

[11] pH Meter using Arduino Uno and LCD Display
[12] Control an LED with Raspberry Pi 4 and Python 3
[13] Raspberry Pi Arduino Serial Communication
[14] Communication between two computers using python socket
[15] PhonePi SampleServer
[16] SOLENOID VALVE INTERFACING WITH ARDUINO
[17] Java Tutotials- what is Socket?
[18] Transmission Control Protocol (TCP)