Water Flow Monitoring

Team 3 (DoW) - Members

Name	Roll Number
Keyur Ganesh Chaudhari	2020101100
Srikar Bhavesh Desu	2020101003
Sambasai Reddy Andem	2020101014
Arjun Muraleedharan	2020101099

About the Project

Date of Presentation	24 September 2022
Faculty Mentor	Professor Sachin Chaudhari
Teaching Assistant	Nilesh Bawankar

I. Motivation

In today's world, monitoring water flow in a specific area of interest (such as a water cooler or a pipe leading to the house's water tank) has tremendously useful applications. Measuring metrics such as the flow rate of water through a pipe or the total volume obtained per day can:

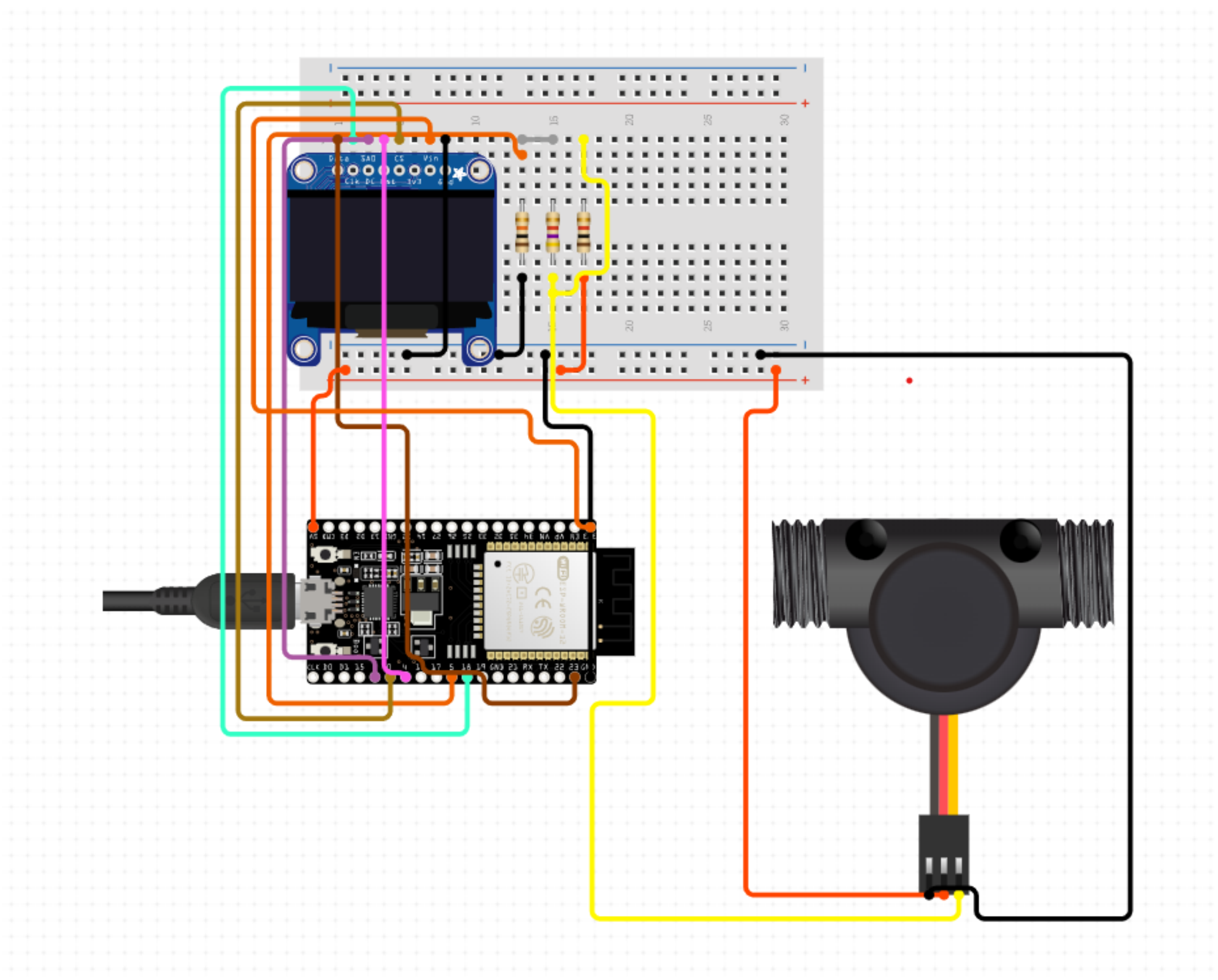
- Improve our lives by helping us understand water consumption patterns in our surroundings
- Promote conservation of water by keeping track of the aforementioned water consumption patterns.

For example, an abnormally low flow rate reading could be telling us that there is a leak in the pipe leading to wastage of water. Moreover, by analyzing the data, we are in a position to predict future patterns based on existing ones. This proves to be highly useful since it can not only tell us when to anticipate certain rates of water supply but can be the first step in understanding how external conditions could possibly affect water flow.

The **major goal** of our project is to develop an IoT-based system to monitor the water flow of the water cooler on the ground floor of the Vindhya building (A2 entrance). This system would consist of a water flow sensor measuring the flow rate and sending the obtained readings to an ESP32 microcontroller. The ESP32 would then interact with the cloud to store this data, which would then be accessed by a dashboard website, which is the point of interaction of the end user with the system. Specifically, our objectives throughout the project are to:

- Build a hardware system incorporating the sensor and microcontroller such that data is detected and sent in a consistent and reliable manner.
- Monitor and store this sensor data on the cloud (via Thingspeak and OneM2M) for the purpose of accessing it for viewing and analytics.
- Build a dashboard website so that the data is viewable publicly, visualize the data to better obtain results from it, and derive useful inferences from the data.
- Ensure interoperability of the system, since the user must be able to interact with the system, no matter which hardware and software they are working with.

II. Circuit Diagram and Hardware Specs



Hardware Specifications

1. Microcontroller

The microcontroller we are using is the Espressif ESP32-WROOM-32D. It is a potent, all-purpose Wi-Fi+BT+BLE MCU module that is designed to work with many different applications. To exchange data with the OM2M and Thingspeak servers, we use the ESP32 Wi-Fi Module. On the microcontroller, we make use of the following pins:

- PIN GND: for connecting the OLED display to the ground.
- PIN 3V3: for connecting the OLED display to VCC.
- PIN D21, D22: for connecting OLED display's SCL and SDA.
- PIN GND: for connecting the water flow sensor to the ground.

- PIN VIN: for connecting the water flow sensor to VCC.
- PIN D27: for reading the pulse output from the sensor.

2. Water Flow Sensor

We are using the YF-S201 SEA water flow sensor. The water flow sensor consists of a plastic valve body, a water rotor, and a hall-effect sensor. When water flows through the rotor, the rotor rolls. Its speed changes with different rates of flow. The hall-effect sensor outputs the corresponding pulse signal. This one is suitable to detect flow in water dispensers or coffee machines. It has the following specifications:

- Working Voltage: DC 4.5V~24V
- Normal Voltage: DC 5V~18V
- Max. Working Current: 15mA (DC 5V)
- Load capacity: ≤ 10 mA (DC 5V)
- Flow Rate Range: 1~30L/min
- Load Capacity: ≤10mA (DC 5V)
- Operating Temperature: ≤80℃
- Liquid Temperature: ≤120℃
- Operating Humidity: 35%～90%RH
- Allowing Pressure: ≤1.75MPa
- Storage Temperature: -25～+ 80℃
- Storage Humidity: 25%～95%RH
- Electric strength 1250V/min
- Insulation resistance ≥ 100MΩ
- External threads: 1/2”
- Outer diameter: 20mm
- Intake diameter: 9mm
- Outlet diameter: 12mm

3. OLED Display

We are using the 1.3 Inch I2C IIC 128x64 OLED Display Module (4 Pin). This 1.3 I2C OLED Display is an OLED monochrome 128x64 dot matrix display module with I2C Interface. It is perfect when you need an ultra-small display. Compared to LCD, OLED screens are way more competitive and have a number of advantages such as high brightness, self-emission, high contrast ratio, slim outline, wide viewing angle, wide temperature range, and low power consumption. It is compatible with any 3.3V-5V microcontroller, such as Arduino. It has the following specifications:

- Display Size - 1.3 inch
- Resolution - 128 X 64 pixels
- Controlling Chip - SSH1106
- Display Area - 34.5 X 19mm
- Driving Voltage - 3.3-5V
- Operating Temperature - 40-70 Celsius
- Interface type - IIC
- Pixel Color - White
- Length(mm) - 35.5
- Width(mm) - 33.5
- Height(mm) - 4
- Weight(gm) - 7

4. 3D Enclosure

We had custom designed a 3D enclosure suitable for on-field deployment. This 3D design was then printed using a 3D printer.

III. Methodology

This project required working on different components during different phases of the project timeline. Initially, after we were given the components required for the project, we built a starter circuit to analyze how each component works, based on which, we got an idea of how to use these sensors. A naive hardware circuit was then set up with a startup code to record the values by the Water Flow sensor and display it on the OLED. The next phase was sending data to ThingSpeak and oM2M. This data was then fetched by the dashboard and displayed in the form of charts and graphs. The data which was sent on to ThingSpeak was encrypted using an XOR cipher and decrypted in the Dashboard. The hardware setup was then soldered onto a PCB and fixed into a 3D enclosure which was then deployed at the Vindhya A2 Water cooler.

IV. Deployment Campaign

- **Hardware deployment**

The deployment campaign involved the soldering of the hardware circuit onto a PCB and then fixing it inside a 3D box to make it deployment-ready. A 3D box had to be designed meticulously to hold the circuit and the OLED. This 3D enclosure was then deployed at the Vindhya A2 water cooler.

- **Software deployment**

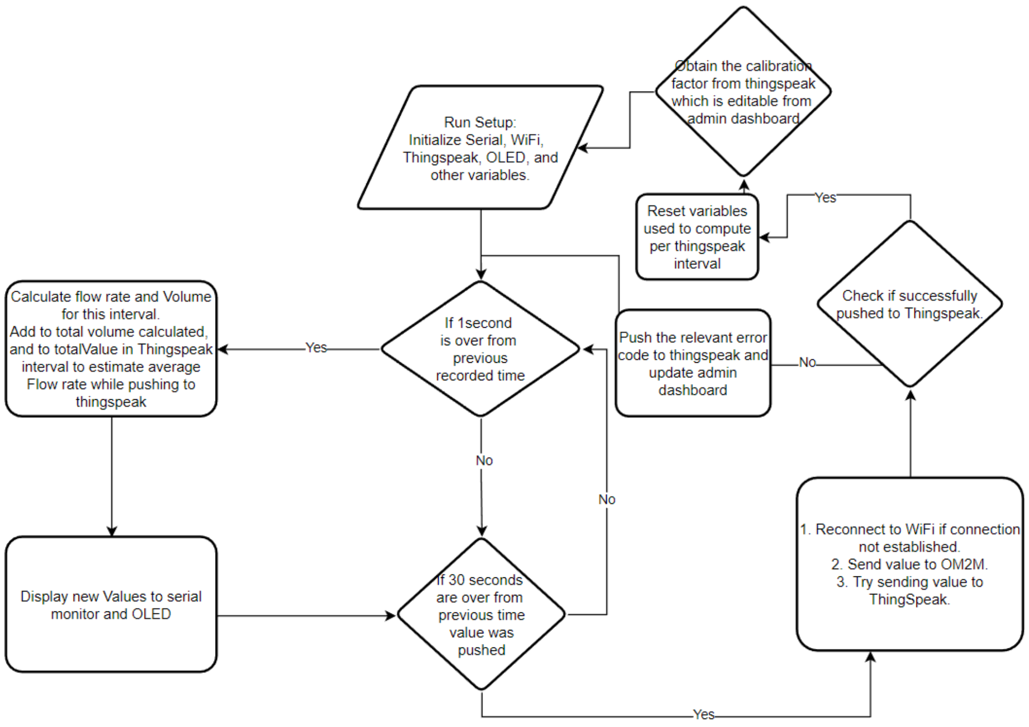
This project involved the development of both a front-end website (the dashboard) and a back-end server on the software side. We required a solution that would take care of software deployment instantly, ensure automatic scalability, and need no additional supervision so that the software deployment process was as seamless as possible.

Our solution for both the front-end and back-end was **Vercel**. Vercel is a cloud platform for website deployment and enabling server-less functions that ticked all of the above boxes. It ensured that the process of production deployment was easily done.

V. Data Validation

Data validation was done physically by checking out the data in real time and thus measuring the amount of liquid flowing. The water that was flowing was measured both using the water flow meter sensor and a bottle and the data were compared to check if both the readings were consistent with each other. Data was also encrypted and decrypted. The data that was decrypted was found to be consistent with the data that was sent thus verifying the encryption-decryption algorithm used.

VI. Flow Diagram of Pipeline


Obtain the calibration factor from thingspeak which is editable from admin dashboard
Run Setup: Initialize Serial, WiFi, Thingspeak, OLED, and other variables.
Reset variables used to compute per thingspeak interval
Yes
Check if successfully pushed to Thingspeak.
Calculate flow rate and Volume for this interval. Add to total volume calculated, and to totalValue in Thingspeak interval to estimate average Flow rate while pushing to thingspeak
Yes
If 1second is over from previous recorded time
Push the relevant error code to thingspeak and update admin dashboard
No
No
No
1. Reconnect to WiFi if connection not established. 2. Send value to OM2M. 3. Try sending value to ThingSpeak.
Display new Values to serial monitor and OLED
If 30 seconds are over from previous time value was pushed
Yes

VII. Communication to ThingSpeak and oM2M

The data from the Water Sensor is sent to both ThingSpeak and oM2M.

- **oM2M**

The oM2M server is hosted on the IIIT-H servers and it serves as a medium of exchange between the ESP32 and the Dashboard. The data that is recorded by the ESP32 is pushed onto the oM2M server instead of storing it locally.

- **ThingSpeak**

The data that we fetch from the water flow sensor is also pushed onto ThingSpeak. On ThingSpeak, we have 5 fields denoting average flow rate, net volume pushed, oM2M status, ThingSpeak status, and also the calibration factor. The data is encrypted at the ESP32 side using an XOR Cipher. This data is then retrieved by the dashboard using the ThingSpeak API and decrypted on the dashboard using a decryption key. This data is then plotted and visualized for further use.

VIII. Development of Dashboard

1. Technology Stack

Our choice of technologies used for the dashboard was:

- React + Next.js (for the dashboard)
- Express (for the back-end server)
- MongoDB (specifically, MongoDB Atlas for remote database services**)**

1. **React** allowed us to quickly set up our front end via a component-based setup that allowed for reusability and modularity. Moreover, its support for the ApexCharts library, which we used for data visualization, as well as other commonly required libraries was a deciding factor. We used **Next.js** along with React for its ability to provide a small back-end server integrated with React to hide private code and variables (such as our decryption code and the secret keys).
2. **Express** was used to set up an additional back-end server to provide features of user authentication for our admin dashboard, which was difficult to set up in the front end for privacy reasons and certain conflicting issues.
3. **MongoDB Atlas** was our choice of database service to store user information required for authentication. It integrates well with JavaScript-based tools like React and Express and is very easy to set up.

2. Data Fetching

To access all flow rate and volume readings, we used the data posted to ThingSpeak. We sent different parameters to different fields of a dedicated channel and used HTTPS GET requests to access this data. This HTTPS-based client-server architecture ensured flexibility and interoperability.

3. Data Visualization

Our dashboard served the purpose of presenting the data in an attractive, visualized format in the form of charts, tables, and plots. We set up a “public” dashboard that anyone could access, and view the data, adding the graphs in the form of screens. In addition, we computed metrics such as average flow rate per day and total volume per day, which can be highly useful.

In order to display and develop the charts, we used the **ApexCharts** library.

Moreover, we added user-friendly features like filtering data by time range, so that the user could view data specific to certain time periods.

4. Security

We set up an encryption scheme while communicating from the ESP32 to ThingSpeak in order to prevent eavesdropping. This was integrated with the dashboard as well, decrypting the data fetched from ThingSpeak.

5. Admin Dashboard

We went a step further and thought about what data an admin of the system would want to have access to, namely:

- WiFi downtime
- Request status for ThingSpeak and oM2M
- Setting calibration factor remotely

We set up an admin dashboard specifically for this purpose, protecting it via token-based authentication.

6. Deployment

We deployed both front-end and back-end to Vercel.

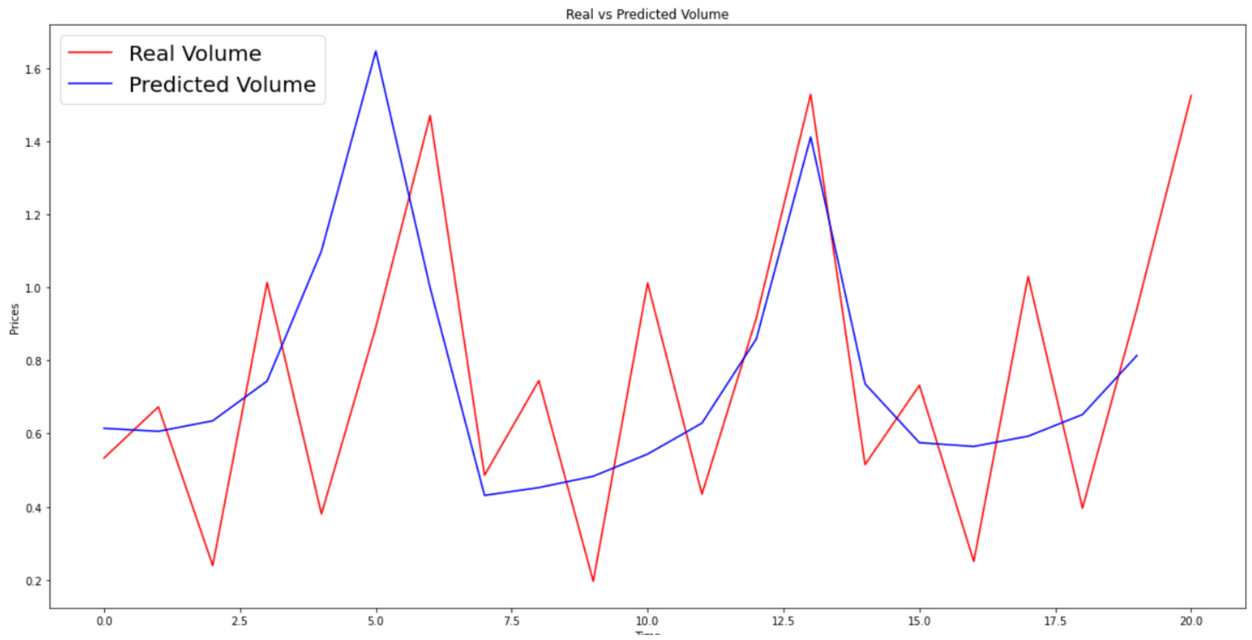
IX. Use of IoT Security Tools

The IoT Security tools used are as follows:

- HTTPS requests are made to oM2M to ensure the requests are encrypted.
- User authentication is done in the Admin dashboard to ensure access only to the admins.
- Data is encrypted before sending it to ThingSpeak for data integrity and credibility.

X. Data Visualization and Analysis

https://colab.research.google.com/drive/1huFcPSHLATW4ePf8WkVU9S5JmPqiMOvq?usp=sharing#scrollTo=ITBmIBA0Th4b



XI. Final Results and Conclusion

- Successfully found out the flow of water through the Water cooler and thus analyzed the volume of water flowing throughout the day during the filling of the Water Cooler.
- This data was further analyzed to make an ML Model that predicts the amount of water consumed on a daily basis.
- By finding out the amount of water consumed on a daily basis, it is easy to predict if the students are drinking enough water, thereby alerting the students of the college to drink more water in case there is lesser water consumption.

XII. Final Code Base

- Deployed link of the dashboard: https://water-flow-monitoring-dashboard.vercel.app
- GitHub repository link: https://github.com/coniferousdyer/Water-Flow-Monitoring-System