



SmartWaterPressureMonitor

Preliminary Report V1

01.07.2021

Nauman Shakir

<https://NaumanShakir.com>

<https://3STechLabs.com>

Overview

This preliminary report aims at defining an architecture for SmartWaterPressureMonitor systems. The goals of this report are mentioned below

Goals

1. The system should allow centralized and real-time tracking to the user and administrator.
2. The system should monitor and log the parameters of water pressure and the pump on/off times.
3. The system should work wirelessly using over WiFi network.

Specifications

The system is divided into 3 different layers which should communicate with each other in real-time.

Layers

- Hardware - Sensor Nodes
- Processing Layer - Server {Ubuntu Server 18.04}
- Application Layer - Mobile and Web Apps

Components Required

- Gravity Water Pressure Sensor(Range 0~1.6 Mpa)
 - <https://www.dfrobot.com/product-1675.html>
- ADS1115 ADC
 - https://www.amazon.com/HiLetgo-Converter-Programmable-Amplifier-Development/dp/B01DLHKMO2/ref=sr_1_3?dchild=1&keywords=ads1115&qid=1625115935&sr=8-3
- ESP32 with SIM7000 GPRS
 - <https://www.banggood.com/LILYGO-TTGO-T-SIM7000G-ESP32-Wireless-Communication-Module-Small-Card-Development-Board-p-1798682.html?currency=CN&rmmds=buy>
- 1x High Current Rated Relay
 - https://www.amazon.com/Artshu-Single-SSR-50DA-H-Resistance-Regulator/dp/B07JZHXPB/ref=sr_1_16?dchild=1&keywords=high+current+relay&qid=1625202266&sr=8-16

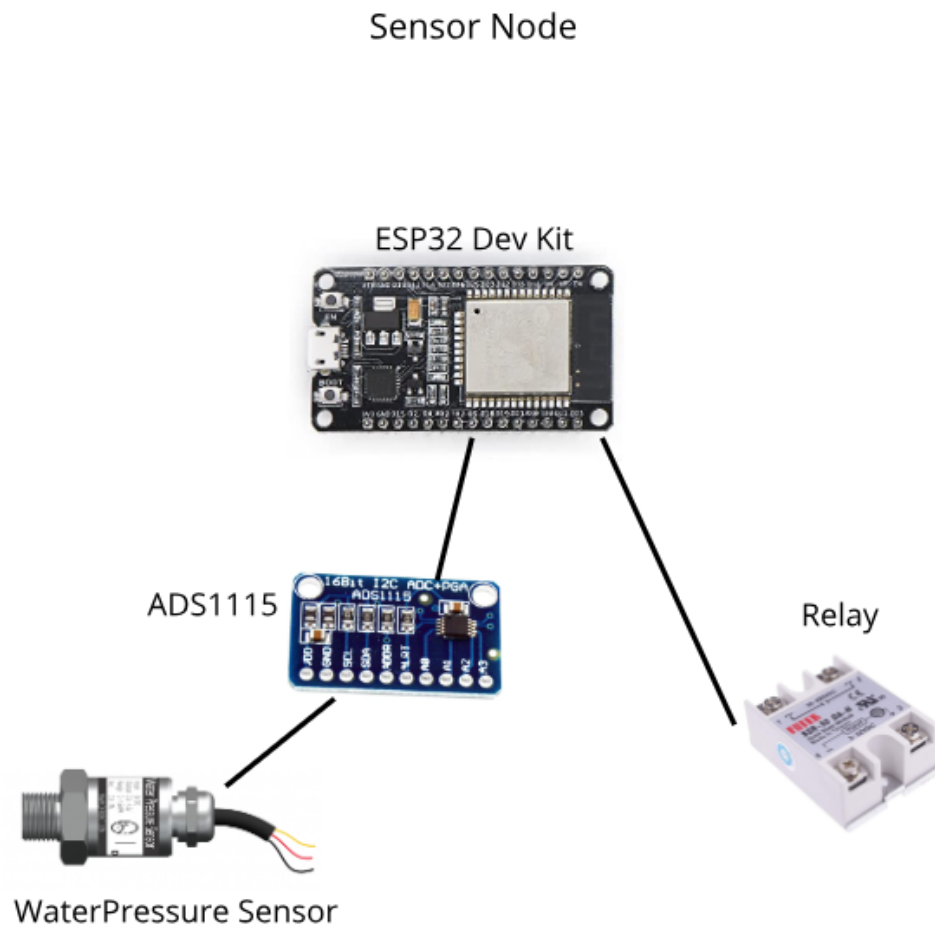
Architecture

The complete project has multiple components

- Sensor Nodes
- IoT Server

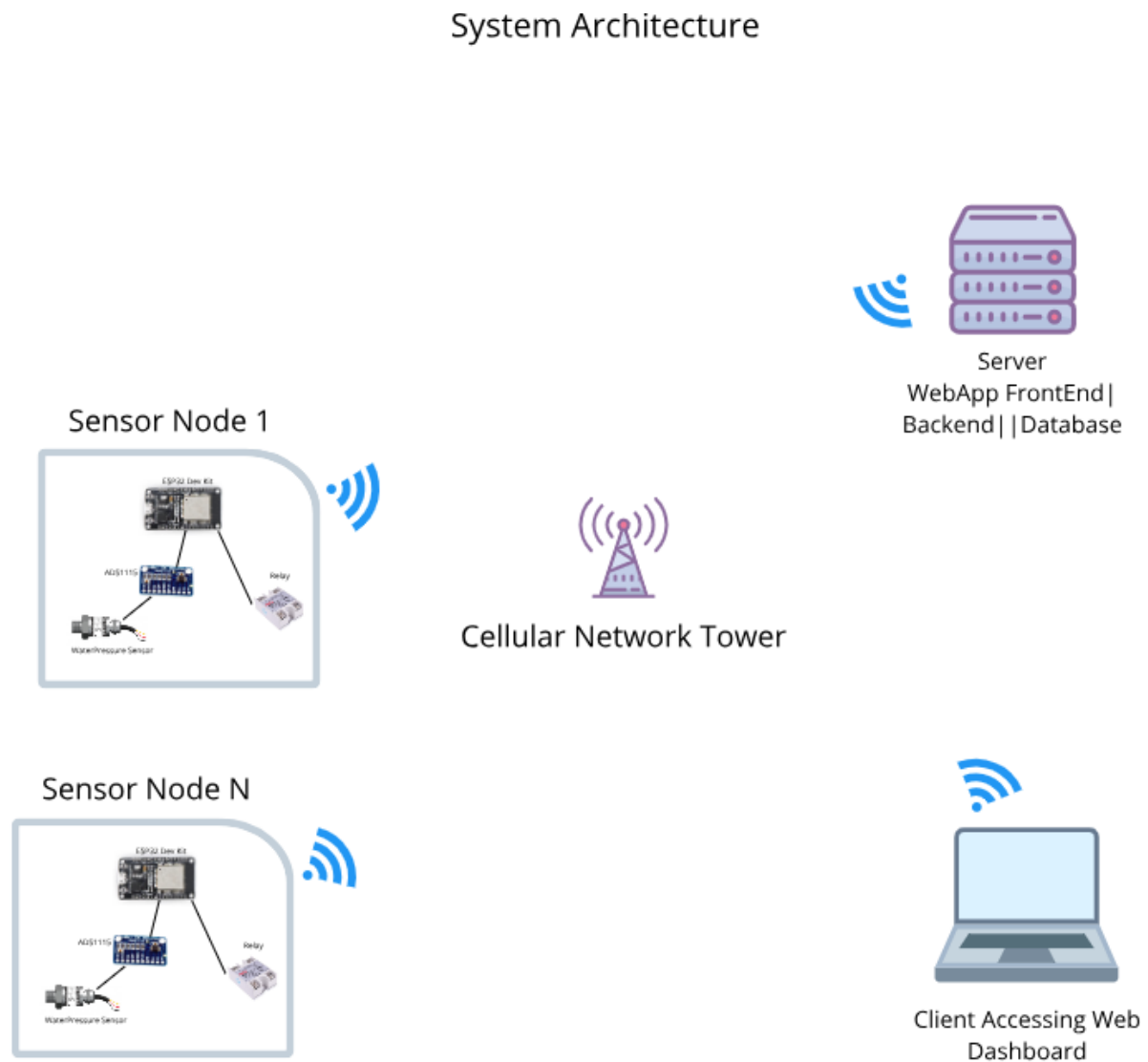
Sensor Nodes contains esp32 to which a high-quality water pressure sensor is connected along with a high current rating relay. While IoT Server is the central authority to which esp32 and web app communicate.

Sensor Nodes Architecture



Above is the Sensor Nodes architecture. Each sensor node contains an ESP32, a pressure sensor and a high current rating relay. The Water pressure sensor is connected to the ESP32 using ADS1115(external ADC) to get accurate pressure readings.

System Architecture



The above shown is the complete system architecture of the whole product. The Sensor Node(s) are connected to the Ubuntu Server running WebApp(frontend, backend and the database). The data from the sensor nodes share real-time with the server and there could be an unlimited number of sensor nodes present in the system. The Client Devices like laptops could access the WebApp from the server and can check the logs and real-time data.

Software Architecture

Multiple Nodes management

The nodes should have unique identifiers which are the MAC address numbers of ESP32 which will be used to uniquely identify each Node.

SensorNodeID:[DataString]

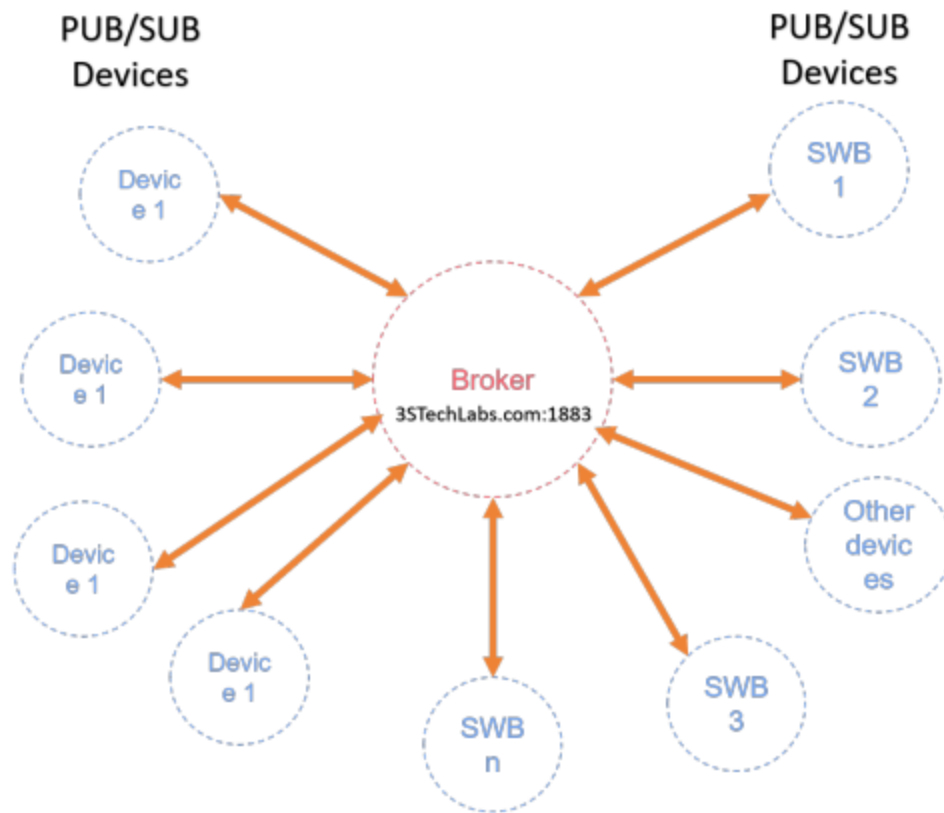
- SensorNodeID is a unique ID of Sensor Node
- [DataString] contains the sensor value(Status)

The communication is done over Cellular Network and data is shared to the server over MQTT communication protocol to make the data delivery fast and responsive.

Web Server Integration

The communication should ideally be done using MQTT because of its smaller data packets and fast response. Your web app will communicate with the IOT Server instead of communicating directly with the Sensor Nodes as this can make the communication much faster without affecting the functionality of the sensor node.

What is MQTT?



Take an example of a system in which there are hundreds of people having smart bands that can display information of a person's surroundings. And then there are Android, iOS and Windows devices that can be used to monitor smart bands to define set of parameters for bands.

So in a scenario where there are mixed types of devices including hardware platforms, the best communication protocol is MQTT.

It can handle two-way and parallel communication and the number of devices that can be connected and communicate via MQTT are limitless, the only limit is server resources. MQTT is also known as pub/sub protocol.

Hence the protocol of choice here is MQTT.

WebApp

The Webapp will allow

- Admin login and signup
- Monitoring real-time pressure sensor data and the relay state.
- History of the events(water pumps turned on/off events with timestamps)
- Listing all the available sensor nodes in the system.
- Ability to download the logs in CSV format.

The WebApp will be made using VueJS and nodejs.

Enclosure Design

The sensor node can be put into an enclosure to save it from the environmental variations. The design could be 3d printed. Below is the example of such an enclosure.



The actual design will depend on the size of sensor node and the ports that are required to be available for water pressure sensor and the relay.

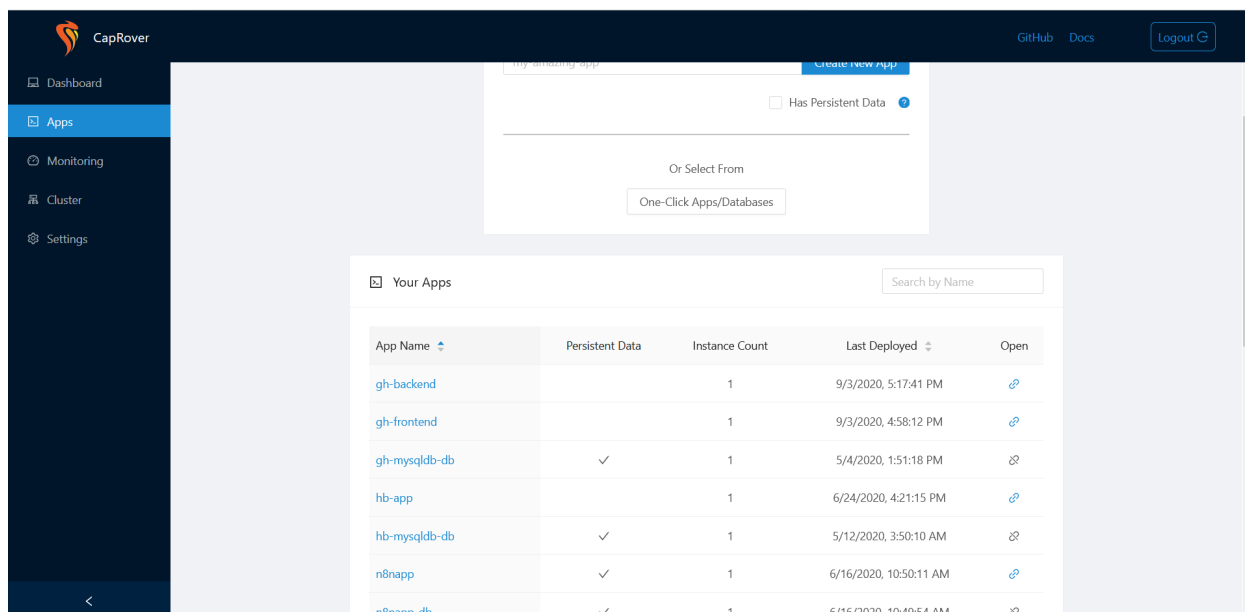
Management and CI/CD Pipeline

We will use Github for delivering the code. The developers from my team will be working on Hardware, Firmware, Backend, Frontend, and apps simultaneously and for modularity we will dockerize different components of the app.

For Continuous Integration and Delivery, we will be using CapRover running on a bare-metal Ubuntu 18.04 instance and each component of the project's GitHub repository will be linked to the respective Containers running in the Caprover. It will allow fully automated delivery.

We will use our own company's production server running on AWS for testing and delivery during the development time. After that we can transfer the files to your own AWS and details of migration are mentioned in the Terms and Condition section of this report.

My automation engineer will work on this.



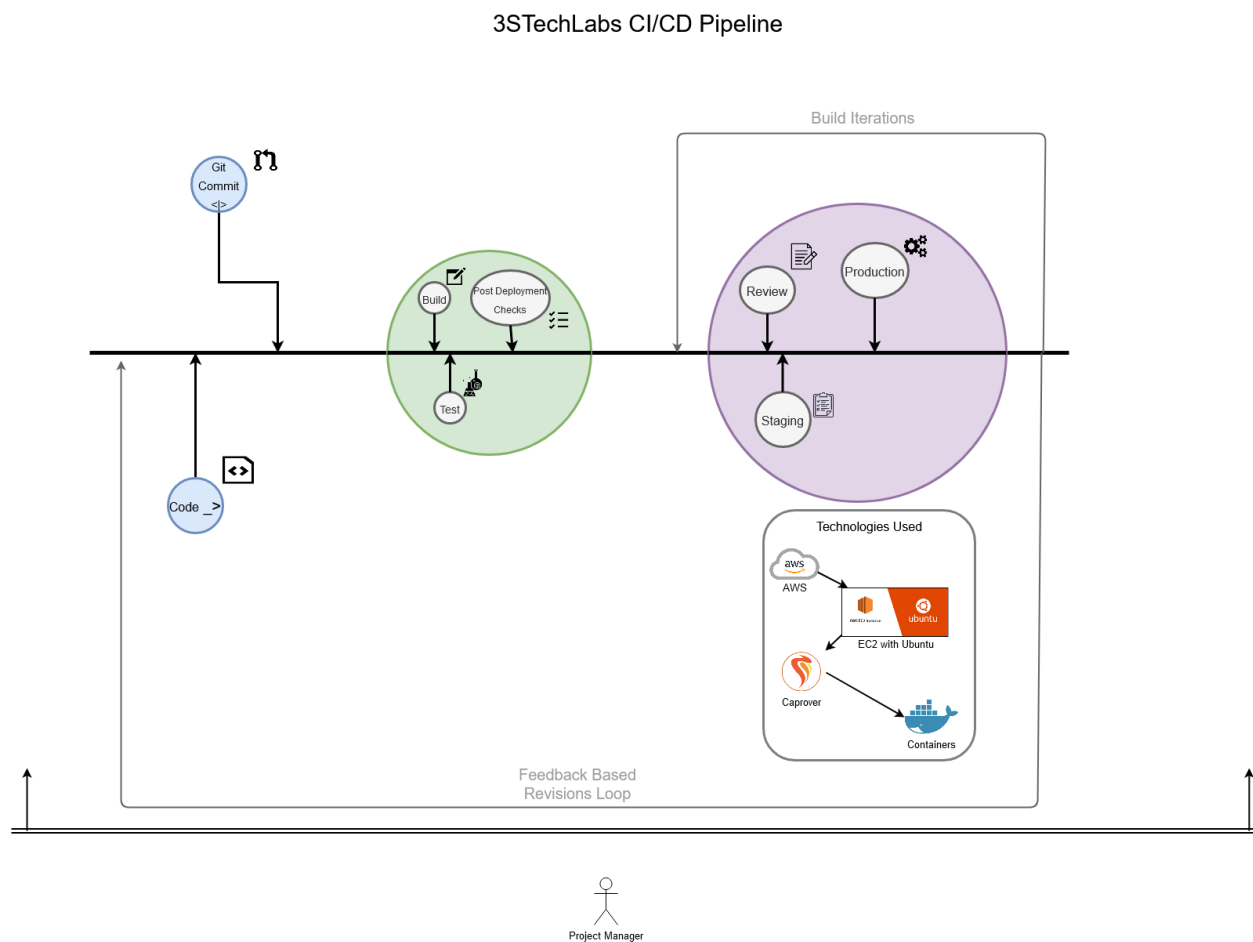
The screenshot shows the CapRover dashboard interface. On the left is a dark sidebar with navigation links: Dashboard, Apps (selected), Monitoring, Cluster, and Settings. The main area displays a 'Create New App' modal window at the top, which includes a 'Has Persistent Data' checkbox and a 'One-Click Apps/Databases' button. Below this, a 'Your Apps' section contains a table listing deployed applications.

App Name	Persistent Data	Instance Count	Last Deployed	Open
gh-backend		1	9/3/2020, 5:17:41 PM	Open
gh-frontend		1	9/3/2020, 4:58:12 PM	Open
gh-mysqldb-db	✓	1	5/4/2020, 1:51:18 PM	Open
hb-app		1	6/24/2020, 4:21:15 PM	Open
hb-mysqldb-db	✓	1	5/12/2020, 3:50:10 AM	Open
n8napp	✓	1	6/16/2020, 10:50:11 AM	Open
n8napp-db	✓	1	6/16/2020, 10:49:54 AM	Open

From the picture above you can see that all of the services like backend, frontend and database etc are running in separate containers allowing a smooth delivery pipeline.

3STechLabs' CI/CD Standard Pipeline

Below is our well-tested CI/CD Pipeline for project management and delivery. This has been proven to work for our 10+ Full-Stack IoT Projects and Products.



(For a detailed look, please open the image file of above diagram)

Questions and Their Answers

1. Could you develop a prototype for us?

- Do you want me to send you the completed prototype to your address? If yes, unfortunately, I can't do this at this moment and neither can I commit anything like that because of the current Covid-19 situation the flights are delayed eventually delaying the shipments and in some cases even missing the delivery altogether.
- On other hand, I will provide you
 - Complete Code
 - Circuit diagrams
 - A complete report on how to setup and configure things.
- You can use all the delivered files and documentation to assemble. the prototype yourself.

Milestones Breakdown

1. Sensor Nodes Design - **Milestone 1** **\$350**
 - a. Hardware and Firmware + Setup Document(Breadboard Based)
 - b. Development Time 15 Days
2. WebApp and IoT Server Design and Configuration - **Milestone 2** **\$600**
 - a. WebServer backend, Frontend, and other related things + Setup Document
 - b. Development Time 20 Days
3. Enclosure Design - **Milestone 3** **\$200**
 - a. 3D Printable Enclosure Design
 - b. Fusion360 will be used for the design
 - c. Development Time 6 Days

Terms and Conditions

- We will not ship anything physically unless otherwise decided, instead, you will be provided with a setup document with the completion of each milestone which will be easy to follow and will contain all of the necessary information.

In this document, the company refers to 3STechLabs.

If you agree to this report please fill-out the form below(optional).

<https://naumanshakir3s.typeform.com/to/jZlIJp>

Profile

Name: Nauman Shakir

Company: 3STechLabs

Designation: Founder and Program Manager

Email Address: NaumanShakir3S@gmail.com

Portfolio: <https://NaumanShakir.com>

I'm a Full-Stack IoT Developer and have done more than 150 hardware projects and running an IoT and Hardware Design House

<https://3STechLabs.com>

<https://facebook.com/3STechLabs>

<https://Linkedin.com/company/3STechLabs>

Freelancing Profiles

<https://www.fiverr.com/naumanshakir>

<https://www.upwork.com/fl/naumanshakir3s>