IOT BASED SMART WATER MANAGEMENT

A Project report submitted in partial fulfilment of the requirements for the degree of B. Tech in Information Technology

By

T. RAMYABHARATHI (513221205014)

Under the supervision of

Professor & HOD

Department of Information Technology

SMART WATER MANAGEMENT

PHASE 4 : DEVELOPMENT PART – 2

**Hardware Setup**

Before diving into the Hardware & Software setup, it is important to first understand what the project is intended to do? Way before starting my IoT & Robotics journey I always wanted to solve the **problem of overflowing water tanks.**

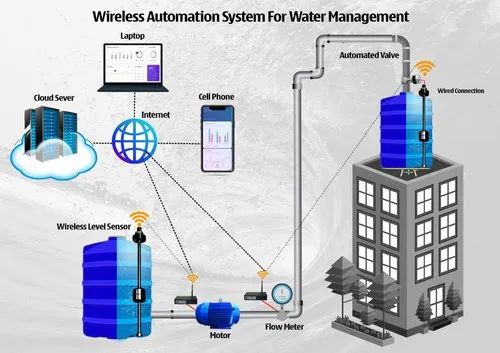
So, after understanding what do **Internet of Things & Machine Learning** means I thought of making a project which will eliminate this traditional problem to the full extent. But I was startled to see that there were already thousands of projects covering this area. Almost every sensor out there has been used in these projects.

I think all of the projects which were made earlier have the following drawbacks:

* In the real world scenario the water supply chain, from the water source (eg dam, water facility) to water tanks in our home, is a very long one and we have to admit that **the problem of water wastage starts from the initial end of the cycle and penetrates through the very last end of that cycle.**All of the previous projects only focus on the very last end of the cycle ie the project only deals with the water tanks of our homes.
* The projects were **almost passive**. By passive I mean that they will monitor the water level and will turn the water motor off when the water reaches a certain level. This whole computation would be done within the microprocessor and the **end-user can’t**get to know what is happening inside the code at the moment!
* **Less interactive** projects. If one can provide some motion or life to the immobile projects then it will greatly enhance the chances of implementing the project in real life.
* Very few of the projects use the concept of **Machine Learning.**Those who use the ML concepts are just using the pre-built easy structured graphs and anomaly detection things which actually don’t make much more sense!!
* Apart from the basic sensors, I have used many different sensors. Be it a **water flow sensor, solenoid valve,** **or an analog multiplexer IC (CD4051)**, all these sensors helped me to almost nullify the water wastage through the whole water supply chain. In this manner, I focused on the whole water supply chain.
* At every instance of the project execution, the end-user will be updated and informed about each major workflow. The project will talk with the user. Thus making the project **active**.
* In order to make this project more interactive, I dumped the old idea of integrating **IFTTT** or using the **CLI**. Even I have not used either the **blynk application** or the normal switches in my project. Instead of all this, I have provided **four** custom options to control the project which include a **dedicated website, NodeMCU based robot, customized cloud dashboard & voice control (I have not used any drag and drop feature of Blynk or IFTTT applications)**
* Rather than using **simple and pre-built** ML systems I used the **Iterative Dichotomiser 3** **algorithm** and implemented it via Python so that the robot can actually make decisions on its own after analyzing the dataset.

**Functioning of project**

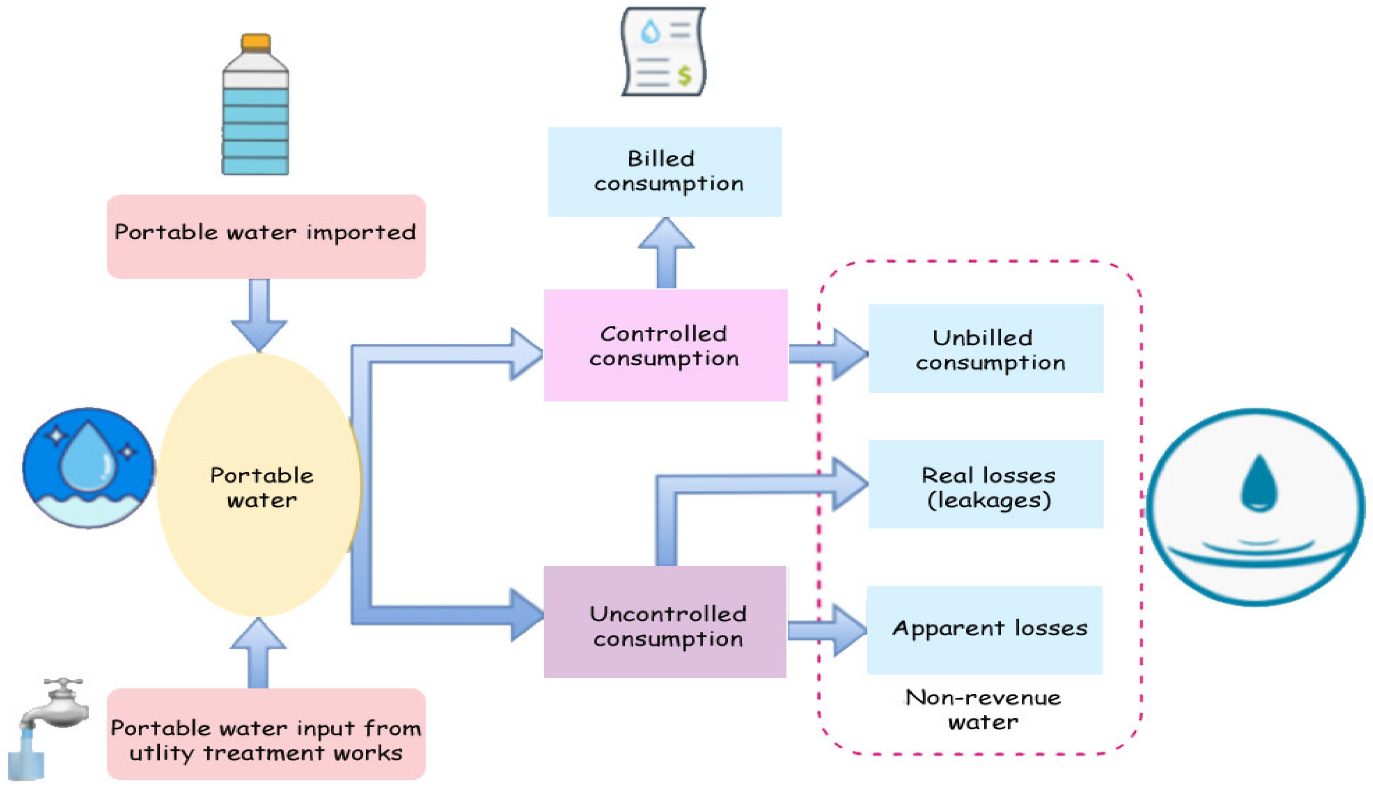
One can break down my project into two major parts. One is the immobile Boltduino/Arduino and the mobile NodeMCU based robot. I will discuss the functioning of each of both sections.



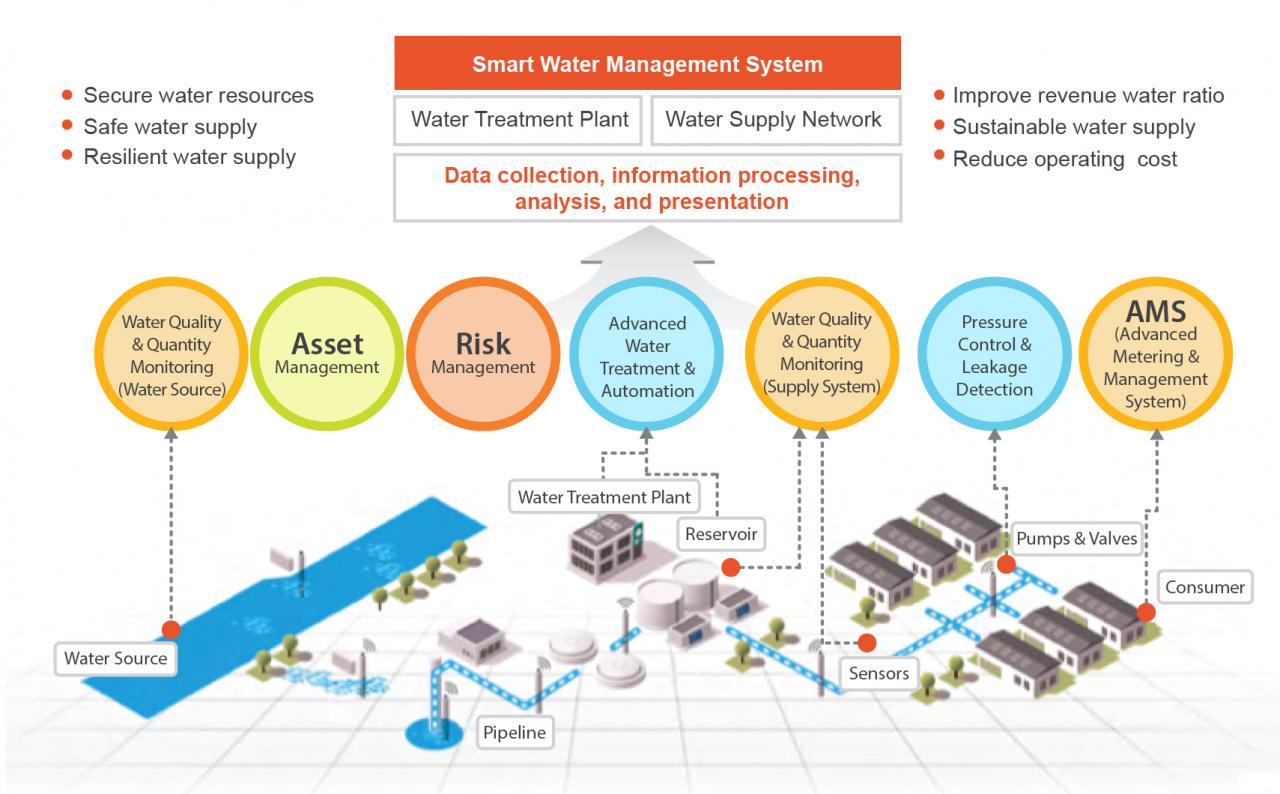
As you can see from the picture that there are 03 water tanks. You can assume the **first one**to be the **main water facility**, the second one to be the **nearest water tower,** and the third one to be the **water tanks in homes**. The main water facility will get the water from the dam and the **water lifting submersible pump** will be turned ON and at that point of time, the solenoid valve will be turned LOW. Whenever the water level in the nearest water tower will be less than 20% then the **solenoid valve** will turn ON and thus the water tower will get filled to a certain threshold (say 95%). When the water tower will be filled then the **DC water motor** will start pumping out the water from the tower to the water tanks in our home.

At the time of pumping the **water flow rate** will be calculated through the YS-401 water sensor. Then the total time to fill the water tank will be calculated and accordingly, the instructions will be sent to the robot so that it can reach the destination in time to inform the user.

**One major aspect of the Arduino setup** is that I have to make the Arduino and the python program communicate with each other. I have to do this as **Bolt Wifi Module can understand Python language, not Arduino’s language.**So I have to make the Arduino pass some certain keywords to the Python program and then the Python program will act accordingly. For this, I used **serial communication** through the **COM port**. This was a tricky part as the Arduino has to pass about 6 parameters to the python program and then python program has to proceed accordingly.



In order to make the passed variable understandable to the python code, I used the **decode and strip methods**. In this case, the decode() method converts the string from **UTF-8 encoding to binary encoding**. Another problem after using this decoding method was that say**for example the passed variable was “stop”** then the decode method will append some escape sequences to the variable and now the variable will look like **“stop/n/r”**. In order to remove the extra escape sequences and also not to affect the binary encoding, I used the **strip method as it removes any spaces or specified characters at the start and end of a string**.



There could be many possibilities in which the user may not be able to turn the motor off. In those scenarios, the Arduino will use the **TTF** (Time To Fill) to turn off the motor, when the TTF is about to be reached. So, **in this way, the project becomes quite interactive and the user also gets timely updates regarding the whole project.**

The first work for the robot is to fetch the **place** attribute so that it can find the place where it has to go. Once the robot fetches that value then the TCS3200 color sensor comes into the picture. Basically in the dataset, there are 03 values for place attribute which are **balcony, bedroom, and hall.** From the starting point of the robot to the final destination I had laid down 03 different colored lines so that the color sensor will follow a particular colored line after getting the predicted value. The place has the following colors:

1. **Red – Bedroom**
2. **Blue – Balcony**
3. **Green – Hall**

