**3118 -Smart Water Management System**

Water scarcity and water management topics are very closely related to each other. Improper water management leads to water scarcity and vice versa. Don’t assume that this project is also one of the 1000 similar projects based on water management issues.  
  
I am confident that if a person goes through this documentation, he will get to know why this project is unique and smart from other projects! Not only this project solves the old traditional problem of water management in a new way but it also provides a smart way of interacting with robots

## Things used in this project

**Hardware components**

* 1. **Sensors Used With Boltduino/Arduino**
     1. 5V Relay
     2. I2C LCD
     3. Boltduino
     4. 9V Battery
     5. Bolt Wifi Module
     6. IRF540 MOSFET
     7. Water Flow Sensor
     8. Ultrasonic Sensor X 2
     9. 1N4007 Rectifier Diode
     10. 12V DC Solenoid Valve
     11. Water Lifting Submersible Pump
     12. 4-way Capacitive Touch Switch Module
     13. 3-6 V Mini Micro Submersible Water Pump
     14. LM35 IC (Temperature sensor)
  2. **Sensors Used With Boltduino/Arduino**
     1. Nodemcu
     2. Piezo Buzzer
     3. IR Sensor X 2
     4. DC Motors X 2
     5. 12V DC Adapter
     6. TCS3200 Color Sensor
     7. Capacitive Touch Sensor
     8. ESP8266 Motor Driver Shield
     9. Analog Multiplexer IC – CD4051

**b) Software apps and online services**

* 1. Arduino IDE
  2. Bootstrap Studio
  3. Spyder (Anaconda)
  4. Twilio
  5. Canva
  6. Hostinger
  7. Integromat
  8. Mega Creator
  9. Pichon (Icons8)

### 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 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!!

So, I wanted to make a project which must be very different from the existing projects. Then I came up with the project Wheels 4 Water, a perfect amalgamation of **IoT, Machine Learning & Cloud.** Let me point the main highlights of my project, which make my project stand out from others.

* 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.