# Smart water fountains

**Objectives:**

smart water fountains using IoT include conserving water resources by dispensing water only when needed, monitoring and maintaining water quality, offering remote control and monitoring, tracking usage patterns, optimizing energy efficiency, ensuring timely maintenance, enhancing user interaction, leveraging data analytics for continuous improvement, integrating with public utilities for resource efficiency, enabling emergency response mechanisms, providing real-time water quality alerts, promoting sustainability through user incentives, and prioritizing data privacy and security to create a system that improves water accessibility and conservation while ensuring safety and efficiency.

**Iot Device setup:**

**Pressure sensor:** A pressure sensor is a device or transducer that measures the force or pressure applied to it and converts this physical quantity into an electrical signal that can be read, recorded, or processed. Pressure sensors are commonly used to monitor and quantify pressure levels in various applications, including industrial processes, automotive systems, medical devices, and environmental monitoring.

**Flow sensor :** A flow sensor is a device designed to measure the rate of fluid flow, whether it's a liquid or gas, through a conduit or pipe. Flow sensors are used to monitor and quantify the volume or mass of fluid passing through a specific point in a system over a specified period of time. These sensors are crucial for applications where it's essential to control, regulate, or monitor the flow of fluids, such as in industrial processes, environmental monitoring, healthcare, automotive, and HVAC systems.

**Platform :**

**Remote Monitoring and Control:**

Implement remote monitoring and control features, allowing users to check fountain status and adjust settings via the app.

**Alerts and Notifications:**

Set up alerts for low water levels, filter replacement, or other critical events.

Send notifications to users via email, SMS, or push notifications.

**Particle:**

Particle offers a comprehensive IoT platform with hardware, connectivity, and cloud services. You can use their devices and cloud infrastructure to monitor water levels, quality, and control the fountain's operation.

**Losant:**

Losant is known for its flexibility and powerful workflow automation. You can use it to monitor water quality and control the fountain's features, such as water flow and lighting, based on sensor data.

**Ubidots:**

Ubidots is an easy-to-use platform for data visualization and control.It's suitable for monitoring and controlling your smart water fountain and can provide real-time insights into water conditions.

**Azure IoT:**

Microsoft's Azure IoT platform can be used to connect and manage water quality sensors, as well as control the fountain's features remotely. Azure IoT Hub and Azure IoT Central are good options for this purpose.

**Explaination:**

A smart water fountain with IoT sensors is a modern and connected solution that leverages Internet of Things (IoT) technology to monitor, manage, and enhance the functionality of a water fountain.

a smart water fountain with IoT sensors offers real-time monitoring, control, and data-driven insights into the fountain's operation. It enhances user experience, conserves resources, and enables efficient maintenance, making it a valuable solution for public spaces, parks, and commercial settings.

**Arduino coding:**

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <Ultrasonic.h>

Const char\* ssid = “YourWiFiSSID”;

Const char\* password = “YourWiFiPassword”;

Const int trigPin = D2; // Trigger pin of Ultrasonic Sensor

Const int echoPin = D3; // Echo pin of Ultrasonic Sensor

Const int relayPin = D1; // Relay module control pin

Ultrasonic ultrasonic(trigPin, echoPin);

WiFiServer server(80);

Void setup() {

pinMode(relayPin, OUTPUT);

digitalWrite(relayPin, LOW);

Serial.begin(115200);

WiFi.begin(ssid, password);

While (WiFi.status() != WL\_CONNECTED) {

Delay(1000);

Serial.println(“Connecting to WiFi...”);

}

Server.begin();

}

Void loop() {

WiFiClient client = server.available();

If (client) {

String request = client.readStringUntil(‘\r’);

If (request.indexOf(“/on”) != -1) {

digitalWrite(relayPin, HIGH); // Turn the pump on

delay(2000); // Run the pump for 2 seconds

digitalWrite(relayPin, LOW); // Turn the pump off

}

Client.flush();

}

// Check water level

Float distance = ultrasonic.read();

If (distance < 10) {

// Water is low, update the web interface

// You can send an HTML response to the client here

}

}

**Project design:** 