

SMART WATER MANAGEMENT-DEVELOPMENT PART1

Designing a smart water management based on the ESP8266 using the WiKoWi application can be a fascinating project. This system would allow you to control and monitor the water fountain remotely through a mobile application. Here is a simplified roadmap for creating such a system:

****Components Needed:****

1. ****ESP8266 Microcontroller:**** The ESP8266 will be the core of the smart water fountain, handling control and communication.
2. ****Water Pump:**** Select an appropriate water pump to provide the necessary flow rate for your fountain.
3. ****Water Level Sensor:**** Use a sensor to monitor the water level in the fountain and ensure it doesn't run dry.
4. ****Relays or Transistors:**** To control the water pump and potentially other features like lighting or water jets.
5. ****Wi-Fi Module:**** ESP8266 has built-in Wi-Fi for communication.
6. ****Mobile Device:**** The WiKoWi application will be installed on a mobile device (smartphone or tablet) for remote control.

****Steps to Create the Smart Water Management : ****

1. ****Hardware Setup:****

- Connect the water pump, water level sensor, and any other components to the ESP8266 microcontroller.
- Ensure power and wiring are correctly set up for safe and efficient operation.

2. ****Programming the ESP8266:****

- Write code for the ESP8266 microcontroller, either using the Arduino IDE or the Arduino core for ESP8266.
- Program the ESP8266 to control the water pump and monitor the water level using the connected sensors.
- Set up a Wi-Fi interface for the ESP8266 to connect to your local Wi-Fi network.
- Create an interface to receive commands from the WiKoWi application, such as turning the fountain on or off, adjusting water flow, and receiving status updates.

****Programming****

```
#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
  }
}

```

3. **WiKoWi Application Integration:**

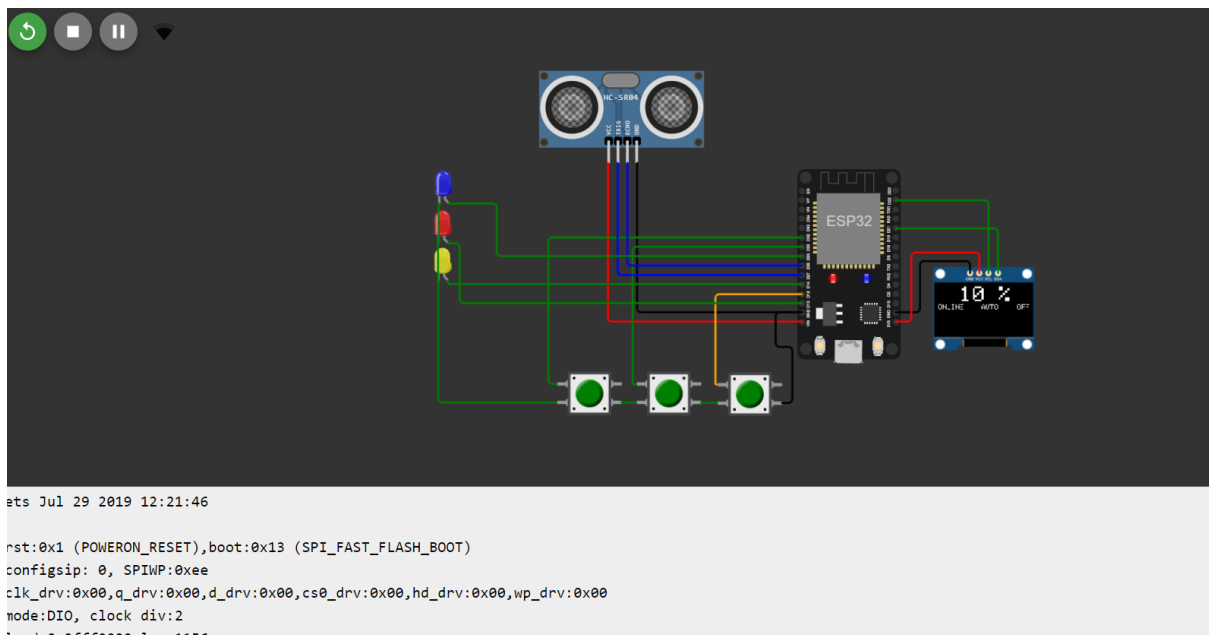
- Develop a mobile application using the WiKoWi SDK that communicates with the ESP8266.
- The application should provide a user-friendly interface for users to control the water fountain remotely.
- Use the WiKoWi SDK to send commands and receive status updates from the ESP8266.
- Implement real-time monitoring of water level and pump status.

4. **Security and Access Control:**

- Implement security measures to prevent unauthorized access to the water fountain system.
- Use authentication and authorization mechanisms for users who want to control the fountain.
- Secure communication between the mobile application and the ESP8266 using encryption.

5. **Testing and Deployment:**

- Thoroughly test the smart water fountain to ensure it functions as expected.
- Deploy the system with the water fountain and make it accessible through the WiKoWi application.



output

6. **Maintenance and Updates:**

- Regularly maintain and update the system to address any issues, improve performance, and incorporate user feedback.

7. ****Optional Features:****

- Depending on your project's requirements, you can add additional features like water lighting control, water jet control, or scheduling capabilities.

Remember that the complexity of this project can vary based on your specific requirements. You may also need to consider factors such as water source, water filtration, and the aesthetics of the water fountain. Always prioritize safety and waterproofing when designing and building a water feature like a fountain.