**SMART WATER FOUNTAIN**

**PHASE 3 – DEVELOPMENT PART 1**

First step of developing a smart water fountain is to simulate it using wokwi simulator. Basic components needed for the smart water fountain simulation are listed below:

1)NODE MCU ESP32 – it act as microcontroller .

2)water pump – To pump water from the tank to fountain.

3)relay module – To control the water pump.

4)Ultrasonic sensor(HC SR04) – To detect the water level in the fountain.

5)wokwi virtual components - for web interface and Simulation.

Then the circuit is build in Wokwi circuit editor by adding the components like NODE MCU ESP32,waterpump,relay module and ultrasonic sensor. A Button and range element is added from the virtual components which provide web interfaces control for the smart water fountain.

In the circuit, relay module and ultrasonic sensors are connected to the NODE MCU ESP32(Microcontroller). The water pump is connected to relay module to control the water flow in the pump.

Ultrasonic sensor was connected to the ESP32 to monitor the water level in fountain and update the user interface.

Then write the python script for all the components using aurdiuno code editor.

**CODING:**

#include <ESP32WiFi.h>

#include <WiFiClient.h>

#include < Ultrasonic.h>

const char\* ssid = "YourWiFiSSID";

const char\* password = "YourWiFiPassword";

const int trigPin = D2;

const int echoPin = D3;

const int relayPin = D1;

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();

}

float distance = ultrasonic.read();

if (distance < 10) {

// Water is low, update the web interface

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

}

}

Then the circuit was saved and simulated. With this setup, we can simulate a smart water fountain that can be remotely controlled, and water level is monitored.