SMART WATER FOUNTAIN

PHASE-05 DOCUMENTATION AND SUBMISSION

~SUBMITTED BY,

A.NITHIKA

1.OBJECTIVE:

The objective of this project is to create a Smart Water Fountains system that leverages Internet of Things (IoT) technology to monitor and manage water resources efficiently. The project aims to reduce water wastage, prevent leaks, and ensure the sustainable use of water in urban and rural environments.

PROJECT DEFINITION:

The project aims to enhance public water fountains by implementing IOT sensors to control water flow and detect malfunctions. This project includes defining objectives , designing the IOT sensor system, develop the water fountain status platform, and integrating them using the IOT technology and python.

DESIGN THINKING :

The block diagram below is a general design of our solution. We divide our design into four modules, including Power Supply, Control Unit, External Control, and Mechanical Unit. Details of each unit is presented in the diagram and described in the next section.

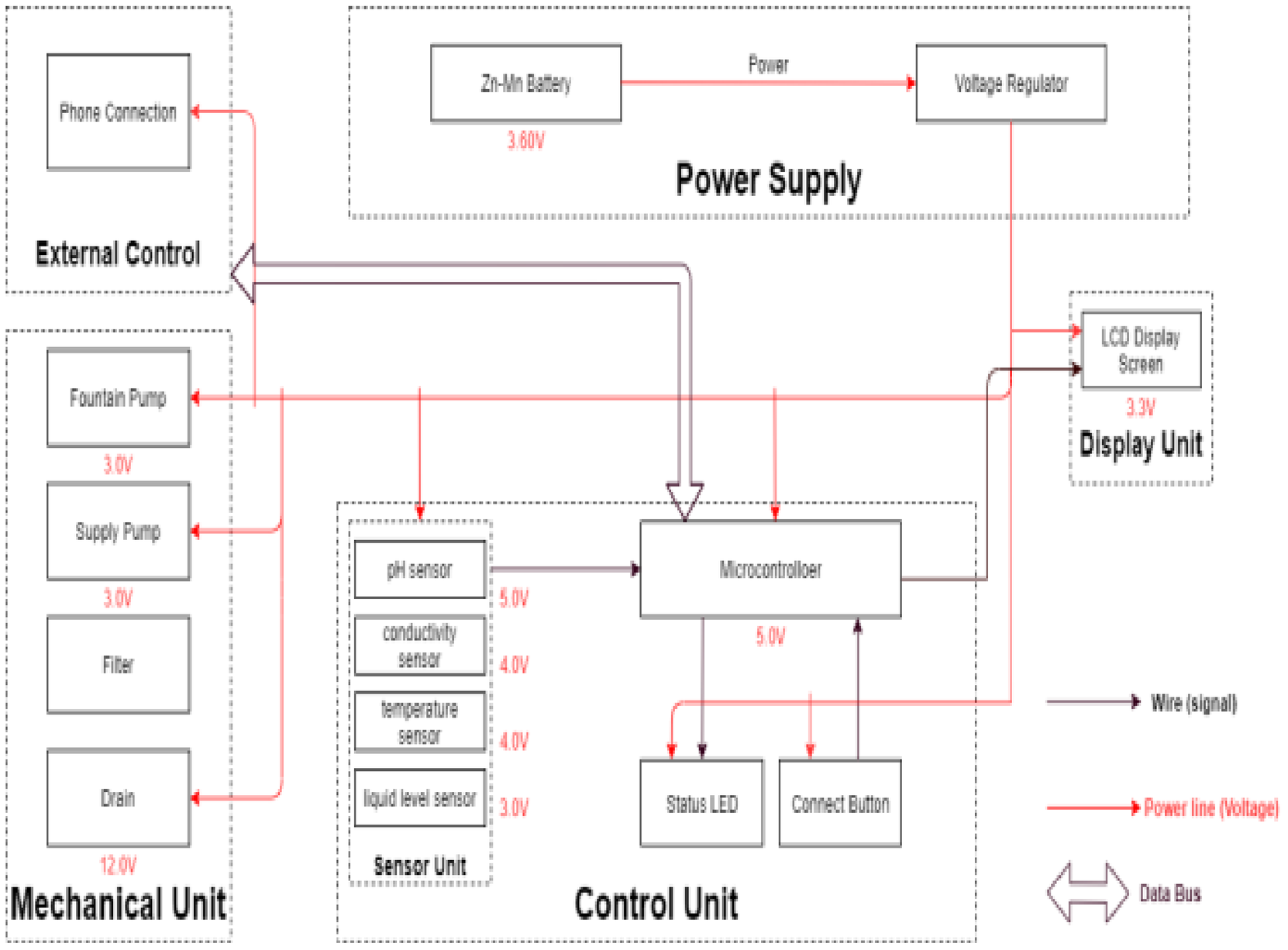
IOT DEVICE SETUP:

The project involves the deployment of various IoT devices for data collection, control, and communication. Here are the key components of the IoT device setup.

FAULT DETECTION SENSOR:

Sensors used in the monitoring of machinery with the object of identifying fault.

BLOCK DIAGRAM OF A DESIGN OF SMART WATER FOUNTAIN:



SENSOR UNITS:

The block contains the four types of sensors. The data acquired from the sensors will be transmitted to the control unit. Control unit will then have some logic designed to send corresponding signals to control other blocks of the water fountain. At the same time the display screen on the water fountain will display the readings along with the determined water quality level and remaining water quality.

TEMPERATURE SENSOR:

A water -proof temperature sensor is going to be used. This temperature sensors is compatible with a relatively wide range of power supply from 3.0V to 5.5V. The measured temperature ranges from -55 to +125celsius degrees. Between -10 to +85 degrees, the accuracy is up to +- 0.5 degrees. This sensor can fulfil all requirements needed for this project.

PH-VALUE:

It is the valued indicator of water quality. This PH -sensors works with 5V, which is also compatible with the temperature sensor.

FLOW SENSORS:

Flow sensors are placed in water distribution pipelines to measure water flow rates and detect any irregularities or leaks.

ULTRASONIC SENSORS:

For water level measurement and flow meter for flow rate measurement.

Checking the water supply and water quality:

Ensure there is enough water in the reservoir. If the water level is low, it can affect the fountain’s performance.

Filtration of the water:

Water fountain is also designed to self- filter the water every time when water is pumped through the submersible water pump.

2. PLATFORM DEVELOPMENT:

The project's platform development involves creating a cloud-based system for data analysis, visualization, and control. The platform consists of the following components:

CLOUD DATA STORAGE:

Data collected from the IoT devices is stored in a cloud database for realtime and historical analysis. Services like AWS, Azure, or Google Cloud can be used for this purpose.

DATA ANALYSIS AND PREDICTION:

Machine learning models are developed to analyse the data and predict water quality, consumption trends, and leak detection. These models help in making informed decisions.

CONTROL SYSTEM:

The platform can send commands back to the IoT devices to control water flow, shut off supply in case of emergencies, or trigger maintenance alerts.

3.EXPLAIN IN DETAIL:

A. Arduino Based Automated Plant Watering System with Message Alert:

The block diagram gives the short illustration of what the framework will do in this specific system. **Arduino microcontroller** is used to control the entire procedure of this Plant Watering System. The use of soil sensor circuit is straight forwardly associated with a computerized soil sensor stick with

digital pins connected to Arduino. An idea for GSM module in this project is to notify the user by sending SMS.

B. Real-Time Clock Controlled Herbal Garden Watering System:

Hygrometer sensors are usually used in smart garden watering system. However, they are known for corroding quickly, particularly if watering is required every day. This system takes advantage of RTC to trigger the watering of an herbal garden. The major components used in this project as shown in figure 2 are: 12V power pack, real time clock, relay, buzzer, and Arduino Nano. An Arduino Nano is used because it is smaller and takes less space. The real-time clock module is connected to the Arduino. The Arduino is programmed to turn ON the pump at a designated time. The pump is connected to the relay and 12V power through a 12mm PVC tube. The water supply is enabled through a 20 litre tank, enough to keep the watering for a few weeks. A float switch is connected to the bottom of the tank, that triggers an alarm if the water level goes below a critical level for the pump [5].

C. Fountain Auto Refill System:

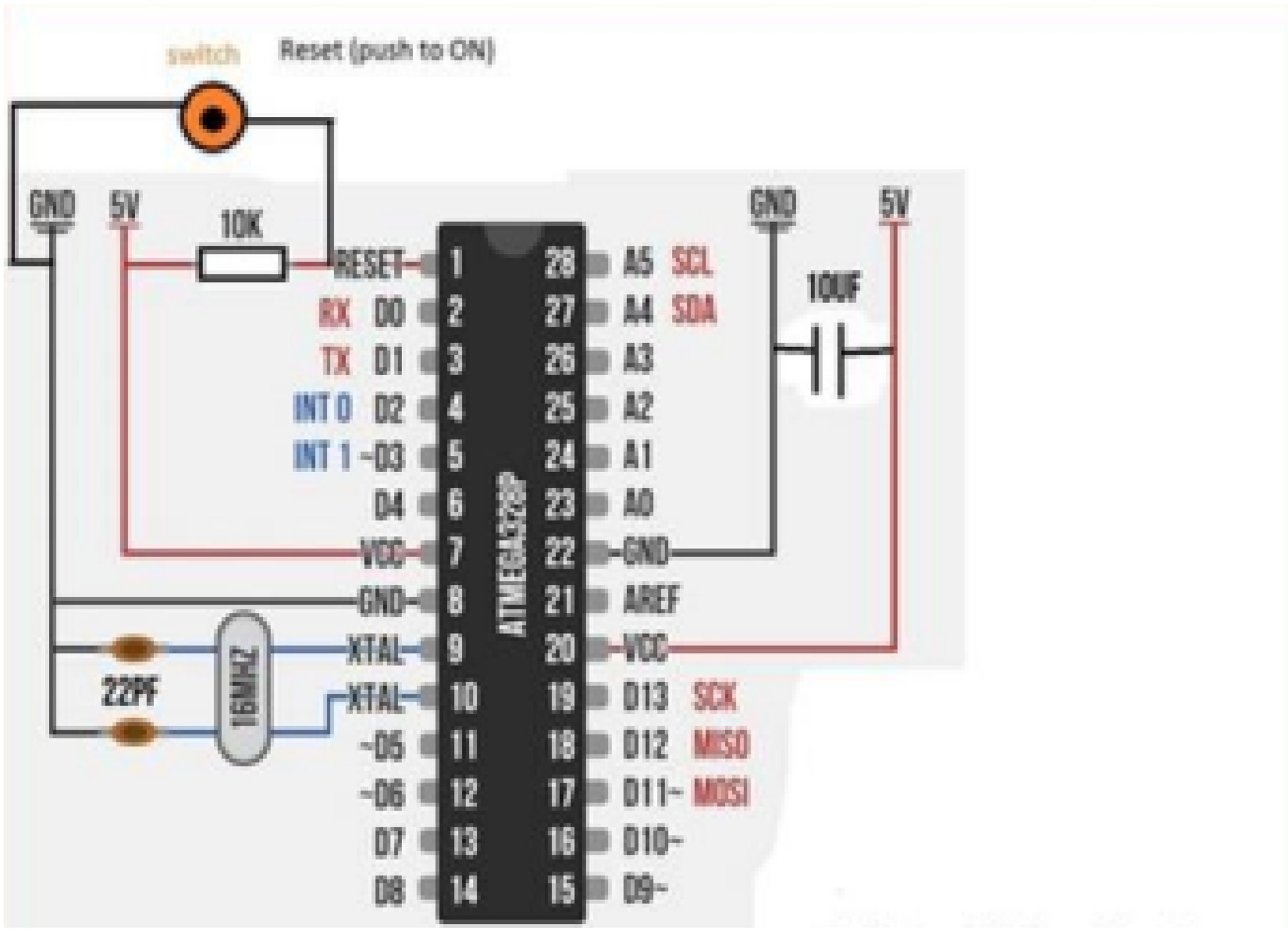
Automatic refill system provides a simple method to refill a fountain or an aquarium automatically when the water level is below a desired threshold. The reduction in water level might be caused by evaporation or water splash. The system consists of an electronic water pump, a float-switch and a microcontroller. When the water gets too low, the float switch sends a pulse to the microcontroller, and the microcontroller activates the pump. To operate a pump using an Arduino microcontroller, a relay is needed to switch on and off a separate power supply. Aquarium auto refill with Arduino is a simple example of how to use a float switch, small liquid pump and a relay to refill an aquarium once the water level gets too low. But this same technique can be used for pet dishes, water fountains, or any other number of similar applications.

METHODOLOGY (DESIGN AND CONSTRUCTION):

The design procedure involves determining the appropriate materials and components to be used in the circuit as well as their properties; which includes components values, voltage ratings and maximum current ratings. These could be actualized easily by consulting the data sheet provided by component's manufacturer. The analysis of each unit will be carried out in detail, for example, the value of capacitor in the power supply can only be determined by mathematical analysis where a formula is used and some important parameters such as total load current and frequency of mains voltage are used to calculate for the capacitance.

REAL-TIME CLOCK (DS3231)

ATmega328P Microcontroller IC 25



Soil Moisture Sensor :

The soil moisture sensor module is used to detect the moisture level of the soil. It measures the volumetric content of water inside the soil and gives the moisture level as output. The moisture sensor consists of two probes that are used to detect the moisture of the soil. These two probes are used to pass the current through the soil to the LM393 comparator IC, and then the sensor reads the resistance to get the moisture values. This Moisture sensor module consists of a moisture sensor, Resistors, Capacitor, Potentiometer, Comparator (LM393 IC), Power and Status LED in an integrated circuit. The moisture sensor probes are coated with immersion gold that protects Nickel from oxidation.

Realtime Clock (DS3231):

The DS3231 is a low-cost, extremely accurate I2C real-time clock (RTC) with an integrated temperature compensated crystal oscillator and crystal. The device incorporates complementary metal oxide semiconductor (CMOS) battery to maintains accurate timekeeping when main power to the device is interrupted. The integration of the crystal resonator enhances the long-term accuracy of the device. A precision temperature-compensated reference voltage and comparator circuit monitors the status of VCC to detect power failures, to automatically switch to the backup supply when necessary and to provide a reset output signal. Additionally, the reset pin is monitored as a pushbutton input for generating a reset externally. The pictorial view of the RTC (DS3231) module.

IoT Gateway Code:

The gateway has code to aggregate data, perform preprocessing, and securely transmit it to the cloud. Communication protocols like MQTT or HTTP may be used.

Cloud Data Storage Code:

Setting up and managing databases can be done using cloud providers' services and APIs.

PYTHON SCRIPT:

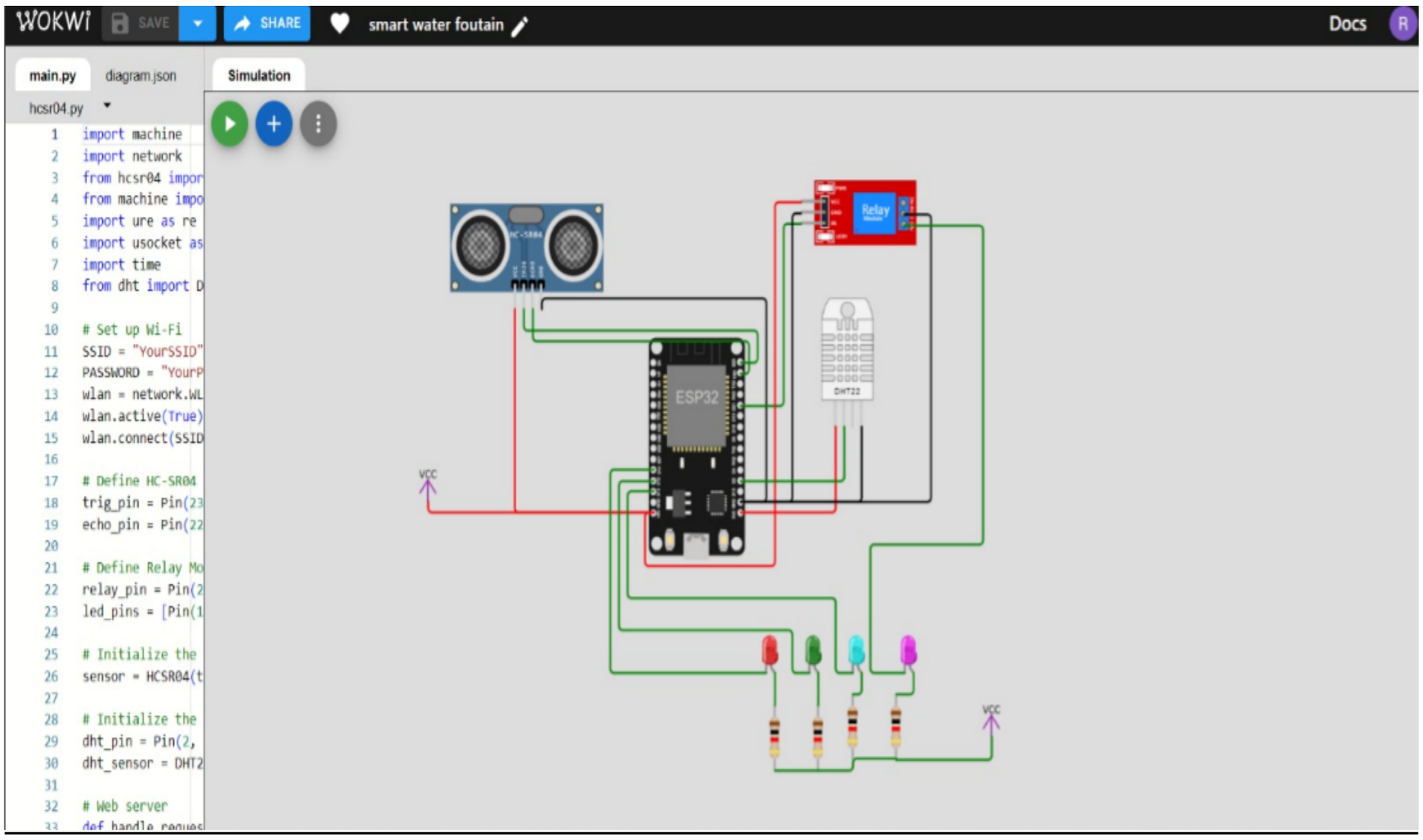
```
import machine
import network
from hcsr04 import HCSR04
from machine import Pin
import ure as re
import usocket as socket
import time
from dht import DHT22
# Set up Wi-Fi
```

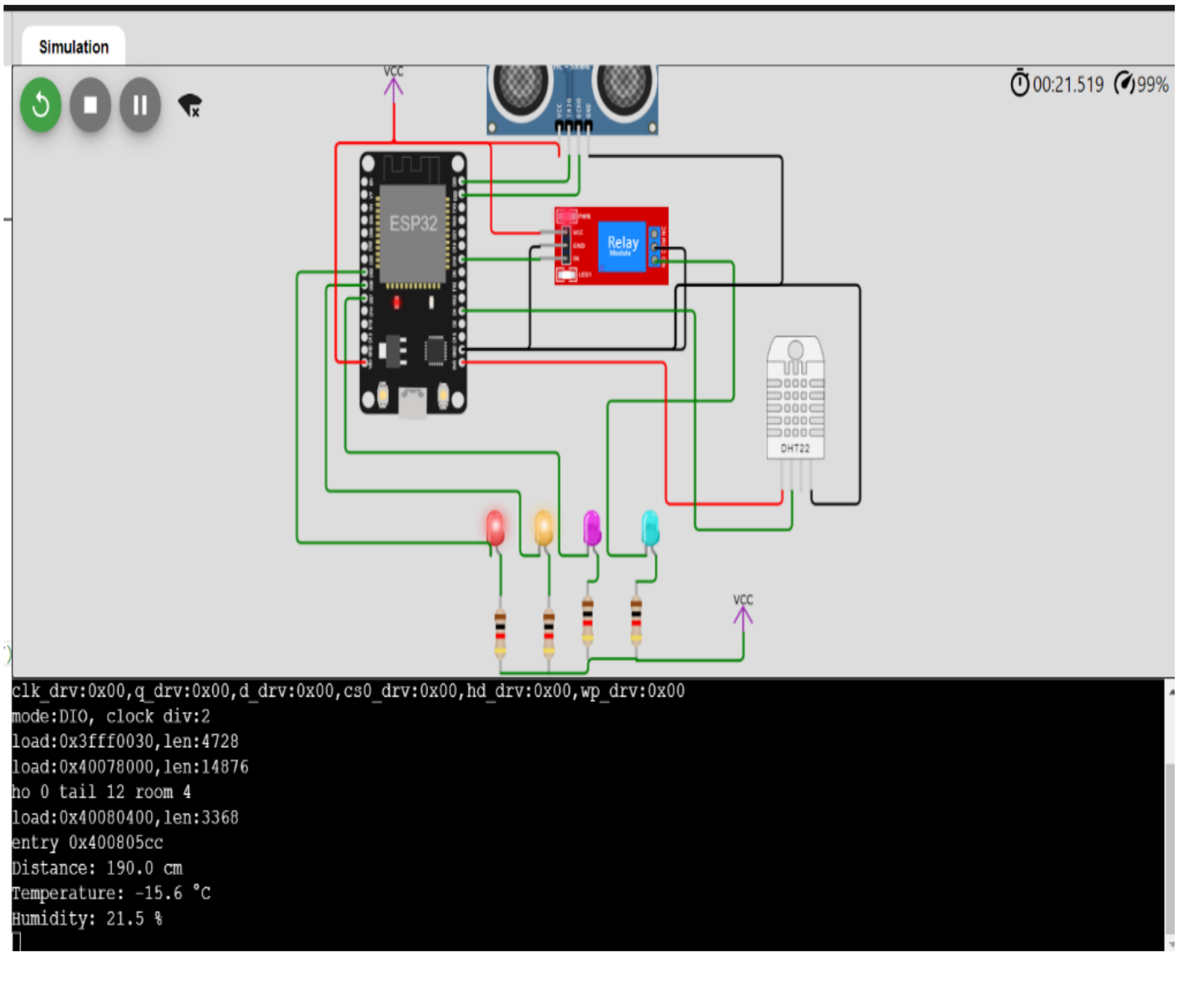
```
SSID = "YourSSID"
PASSWORD = "YourPassword"
wlan = network.WLAN(network.STA_IF)
wlan.active(True)
wlan.connect(SSID, PASSWORD)
# Define HC-SR04 pins
trig_pin = Pin(23, Pin.OUT)
echo_pin = Pin(22, Pin.IN)
# Define Relay Module and LED pins
relay_pin = Pin(18, Pin.OUT)
led_pins = [Pin(25, Pin.OUT), Pin(26, Pin.OUT), Pin(27, Pin.OUT)]
# Initialize the HC-SR04 sensor
sensor = HCSR04(trigger_pin=trig_pin, echo_pin=echo_pin)
# Initialize the DHT22 sensor
dht_pin = Pin(4, Pin.IN)
dht_sensor = DHT22(dht_pin)
# Web server
def handle_request(client):
    request = client.recv(1024).decode('utf-8')
    if 'GET /on' in request:
        relay_pin.on()
    elif 'GET /off' in request:
        relay_pin.off()
   distance = sensor.distance_cm()
    water_level = "High" if distance < 10 else "Low"
    dht_sensor.measure()
    temperature = dht_sensor.temperature()
    humidity = dht_sensor.humidity()
    response="HTTP/1.1200OK\r\nContent-Type: text/html\r\n\r\n"
            response+=f"<html><body><h1>Water Level and Temperature/Humidity
Monitoring</h1>"
```

```
response += f"<p>Distance: {distance} cm</p>"
response += f"<p>Water Level: {water_level}</p>"
response+=f"<p>Temperature:{temperature}°C</p>"
response+=f"<p>Humidity:{humidity}%</p>"
response+="<p><a href='/on'>TurnPumpOn</a></p>"
response+="<p><a href='/off'>TurnPumpOff</a></p>"
  response += "</body></html>"
  client.send(response)
 client.close()
def run_server():
   s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
   s.bind(('', 80))
   s.listen(5)
   while True:
      client, addr = s.accept()
      handle_request(client)
# Main loop
while True:
   distance = sensor.distance_cm()
   dht_sensor.measure()
   temperature = dht_sensor.temperature()
   humidity = dht_sensor.humidity()
   print("Distance:", distance, "cm")
   print("Temperature:", temperature, "°C")
   print("Humidity:", humidity, "%")
 # Control the water pump based on distance
 if distance < 10:
    relay_pin.on()
else:
    relay_pin.off()
```

```
    # Indicate water level using LEDs
if distance < 10:
        for i in range(3):
            led_pins[i].on()
    else:
        for i in range(3):
            led_pins[i].off()
# Run the web server
run_server()
# Delay for a while to avoid excessive measurements
time.sleep(2)
```

OUTPUT OF SIMULATION:


WOKWi
SAVE
SHARE
smart water foutain
Docs
main.py
diagram.json
hcsr04.py
Simulation
ESP32
Relay
DHT22
VCC


Simulation
00:21.519
99%
ESP32
Relay
DHT22
VCC
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:4728
load:0x40078000,len:14876
ho 0 tail 12 room 4
load:0x40080400,len:3368
entry 0x400805cc
Distance: 190.0 cm
Temperature: -15.6 °C
Humidity: 21.5 %

HTML CODE:

```
<!DOCTYPE html>
<html>
<head>
    <title>Water Fountain Status</title>
</head>
<body>
    <h1>SMART WATER FOUNTAIN</h1>
    <section id="water-flow">
        <h2>Water Flow Rate</h2>
        <p id="flow-rate">Loading...</p>
    </section>
    <section id="malfunction-alert">
        <h2>Malfunction Alert</h2>
        <p id="alert-status">No Alerts</p>
    </section>
</body>
</html>
```

JAVASCRIPT CODE:

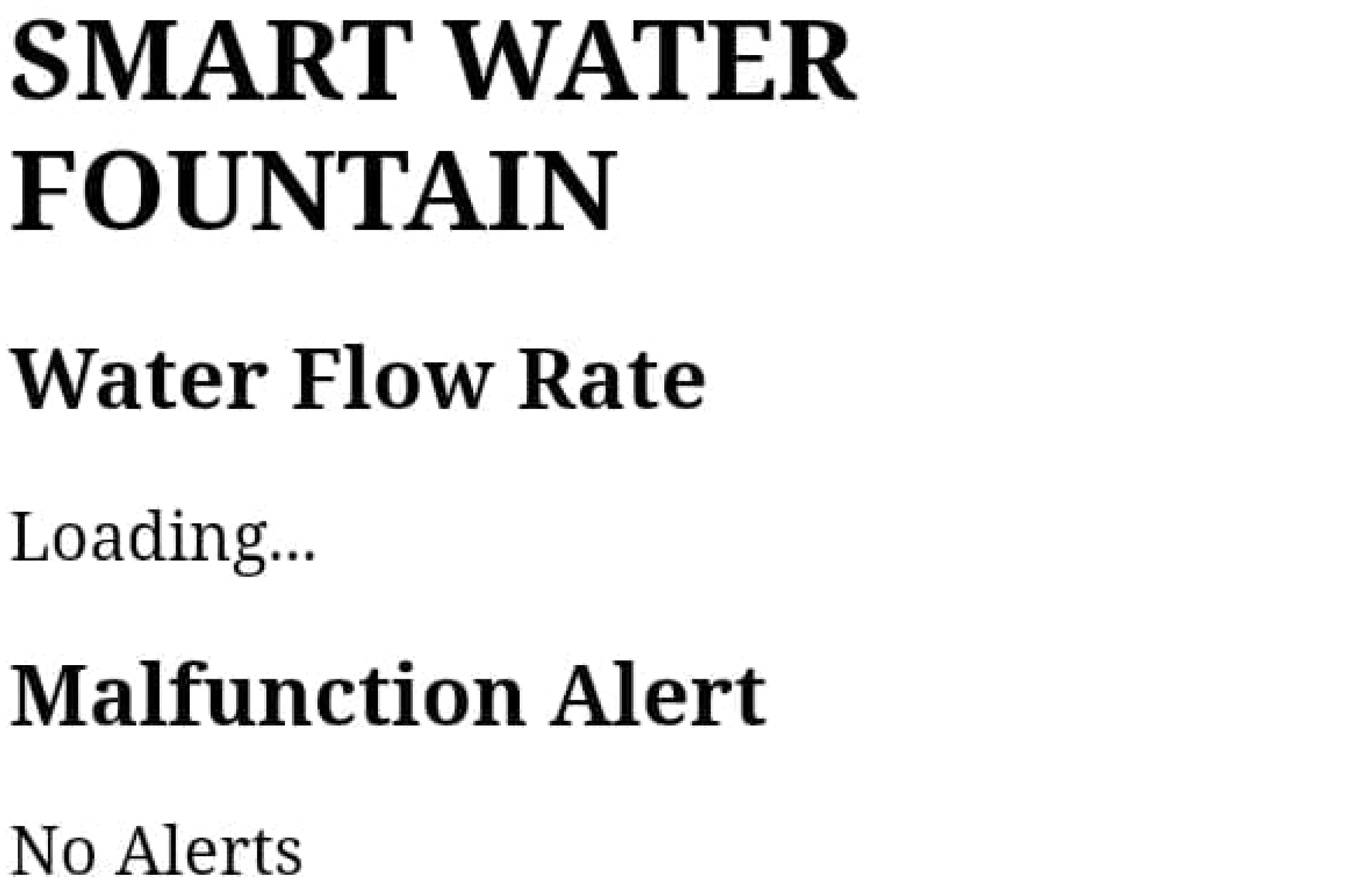
```
// Function to update water flow rate function updateFlowRate() {
fetch('water-fountain-api/flow-rate')
    .then(response => response.json())
    .then(data => {          document.getElementById('flow-rate').textContent = data.flowRate + '
GPM'; // Gallons per minute
    }); }

// Function to update malfunction alerts function updateAlertStatus() {
fetch('water-fountain-api/alert-status')
    .then(response => response.json())
    .then(data => {          document.getElementById('alert-
status').textContent = data.alert ? 'Alert' : 'No Alerts';
```

```
        });
}
```

// Update data every 5 seconds setInterval(updateFlowRate, 5000); setInterval(updateAlertStatus, 5000);

OUTPUT FOR THE PLATFORM:



CONCLUSION:

Keeping the record of usage of water using this experiment will help in solving the water deficiency problem. This paper represents the model design and scope of the SMART WATER FOUNTAIN [FLOW RATES] for solving the water deficiency problem. The volume of water was successfully calculated using an ultrasonic sensor which is used to calculate the variable height of water through the pipe.

A GSM module sent the signal once the threshold volume was reached in the form of SMS. The proposed system helps every house and society to reduce the wastage of water significantly and hence the water scarcity.

FUTURE SCOPE :

The proposed system can solve almost all the problems faced by the existing water meters. There is also scope for adding a feature which will automatically limit the flow of water into particular house or public fountain For this feature a solenoid valve can be installed inside water flow meter so that when it rotates inside the meter then flow of water can be minimized.

THANK YOU