

SMART WATER FOUNTAINS

Phase 3: Development Part 1

Name: Manikandan S

ID:aut2221110065

Project Title: Smart Water Fountains.

Abstract

In our fast paced world, the water fountain will be acting as the natural humidifiers and it also adds moisture to a dry room .

This is an overview of smart water fountain that jets the water into the air in a decorative and dramatic effect.

Smart water fountain are commonly used to promote peace and well being. This smart water fountain uses an arduino which is an open source electronics platform.

The fountain works fully automatically but can also be set up remotely via a mobile app for your convenience.

Our smart water fountain will be used with a help of a laptop.

Flow of water from high gravitational potential energy to low gravitational potential energy causes a water fountain to form, due to increasing pressure on the inside of the system.



Materials needed:

- 4x 12v mini fountain
- 4 relay module
- Arduino
- 12v power supply for the mini fountain
- Female/Male to jumper cables

Codes:

```
#define RELAY1 9
```

```
#define RELAY2 10
```

```
#define RELAY3 11
```

```
#define RELAY4 12
```

```
void setup()
```

```
{
```

```
  OUTPUT
```

```
    pinMode(Relay1,OUTPUT);
```

```
    pinMode(Relay2,OUTPUT);
```

```
    pinMode(Relay3,OUTPUT);
```

```
    pinMode(Relay4,OUTPUT);
```



```
}  
  
void loop()  
{  
  digitalWrite(RELAY1,LOW);  
  delay(500);  
  digitalWrite(RELAY4,HIGH);  
  
  digitalWrite(RELAY2,LOW);  
  delay(500);  
  digitalWrite(RELAY1,HIGH);  
  
  digitalWrite(RELAY3,LOW);  
  delay(500);  
  digitalWrite(RELAY2,HIGH);  
  
  digitalWrite(RELAY4,LOW);  
  delay(500);  
  digitalWrite(RELAY3,HIGH);  
}
```



The fountain works fully automatically but can also be set up remotely via a mobile app for your convenience.

Our smart water fountain will be used with a help of a laptop.

Flow of water from high gravitational potential energy to low gravitational potential energy causes a water fountain to form, due to increasing pressure on the inside of the symstem.

- **Arduino**



- **4 relay module**



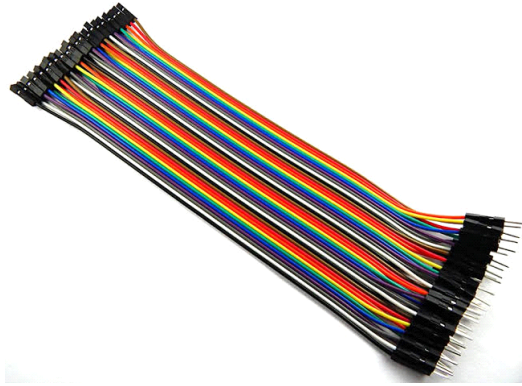


- 4x 12v mini fountain



- Female/Male to jumper cables





Certainly! Let's brainstorm a design and innovation solution to improve smart water fountains to address common issues and provide added benefits:

Problem: Existing smart water fountains may have issues with water wastage, hygiene, and user engagement.

Solution:

Sensors and AI Integration: Incorporate advanced sensors and AI algorithms to detect when a person approaches the fountain. The AI can analyze foot traffic data to predict peak usage times and adjust water flow accordingly, reducing wastage during off-peak hours.



Edit with WPS Office

Hygiene and Safety: To enhance hygiene, use UV-C sterilization technology within the fountain's nozzle area. The UV-C light can automatically disinfect the spout after each use, ensuring clean and safe drinking water.

User-Friendly Interface: Implement a user-friendly touch screen interface with multilingual options, making it easy for people of diverse backgrounds to understand and use the fountain. It can also display water quality metrics in real-time.

Customizable Water Temperature: Allow users to choose between cold, ambient, and warm water options, catering to different preferences and weather conditions.

Reusable Water Bottles: Design a specialized slot for reusable water bottles to encourage eco-friendly practices. Users can place their bottles under the spout, and the fountain can automatically fill them with the desired amount of water.

Mobile App Integration: Develop a companion mobile app that allows users to locate nearby smart water fountains, check water quality, and even pre-set their water temperature and quantity preferences. The app can also gamify water consumption to promote healthy hydration habits.

Sustainability Features: Incorporate a mechanism to collect and



filter rainwater, utilizing it as a source for the fountain during rainy seasons, thereby conserving municipal water resources.

***Accessibility:** Ensure the fountain is ADA-compliant with features like accessible height adjustments, voice commands, and Braille instructions.

***Data Analytics:** Gather usage data to identify trends and inform maintenance schedules. Proactively address issues and reduce downtime.

***Solar Power:** Make use of solar panels to power the fountain, reducing energy costs and environmental impact.

***Education and Awareness:** Use the fountain as an educational tool by displaying facts about water conservation and the environmental impact of single-use plastic bottles.

***Aesthetic Design:** Create an appealing, modern, and iconic design for the fountain, encouraging more people to use it and making it a focal point in public spaces.

By addressing these aspects of design and innovation, smart water fountains can become more efficient, hygienic, user-friendly, and environmentally conscious, ultimately improving the drinking water experience for people in public spaces.



Simulation:

```
import machine
import time

# Pin assignments for the ultrasonic sensor
TRIGGER_PIN = 23 # GPIO23 for trigger
ECHO_PIN = 22    # GPIO22 for echo

# Pin assignment for the LED
LEAK_LED_PIN = 19 # GPIO19 for the LED

# Set the pin modes
trigger = machine.Pin(TRIGGER_PIN, machine.Pin.OUT)
echo = machine.Pin(ECHO_PIN, machine.Pin.IN)
leak_led = machine.Pin(LEAK_LED_PIN, machine.Pin.OUT)

# Function to measure distance using the ultrasonic sensor
def measure_distance():
    # Generate a short trigger pulse
    trigger.value(0)
    time.sleep_us(5)
    trigger.value(1)
    time.sleep_us(10)
    trigger.value(0)
```



```

# Measure the echo pulse duration to calculate distance
pulse_start = pulse_end = 0
while echo.value() == 0:
    pulse_start = time.ticks_us()
while echo.value() == 1:
    pulse_end = time.ticks_us()

pulse_duration = pulse_end - pulse_start

# Calculate distance in centimeters (assuming the speed of sound is 343 m/s)
distance = (pulse_duration * 0.0343) / 2 # Divide by 2 for one-way travel

return distance

# Function to check for a water leak
def check_for_leak():
    # Measure the distance from the ultrasonic sensor
    distance = measure_distance()

    # Set the threshold distance for detecting a leak (adjust as needed)
    threshold_distance = 10 # Adjust this value based on your tank setup

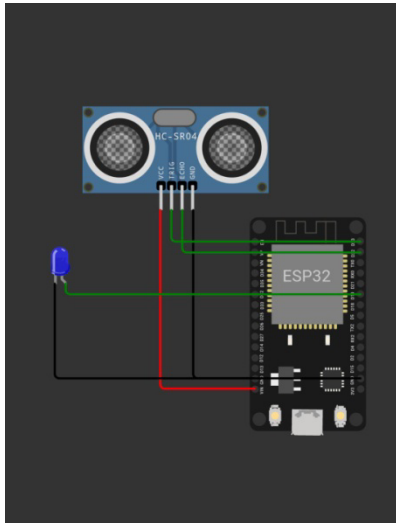
    if distance < threshold_distance:
        # If the distance is less than the threshold, a leak is detected
        return True
    else:
        return False

# Main loop
while True:
    if check_for_leak():
        # Blink the LED to indicate a leak
        leak_led.value(1) # LED ON
        time.sleep(0.5)
        leak_led.value(0) # LED OFF
        time.sleep(0.5)
    else:
        leak_led.value(0) # LED OFF

    time.sleep(1) # Delay between measurements

```





Effective data preprocessing is crucial for accurate analysis and maintaining the functionality and efficiency of smart water fountains. It enables proactive maintenance, resource conservation, and improved user experiences.

