

**GOVERNMENT COLLEGE OF TECHNOLOGY,**

**COIMBATORE-13**

DEPARTMENT OF ECE

**SMART WATER FOUNTAINS**

TEAM MEMBERS **,**

AKSHAYA V A -71772114102

DEEPITA M J -71772114105

DHANVARTHINI B -71772114106

KRITHIKA S -71772114120

**INTRODUCTION**

* Water is an essential part in our day to day life.
* Smart water fountain is an automated system that provides a continuous flow of water in a controlled manner.
* It incorporates various technologies and features to enhance functionality, efficiency, and convenience.

**HARDWARE COMPONENTS**

* Raspberry Pi 3 Model B or later
* Water pump
* Relay module
* Water level sensor (optional)
* Power supply for the pump
* Tubing and fountain nozzle
* Waterproof container for the water reservoir
* Various cables, connectors, and a breadboard

**SOFTWARE COMPONENTS**

* Raspbian OS (or a suitable Raspberry Pi OS)
* Python for programming
* IoT platform (e.g., MQTT, AWS IoT, or Google Cloud IoT Core)
* Libraries for GPIO control (e.g., RPi.GPIO)
* Optional: Web server and HTML/CSS/JavaScript for a web-based user interface

**STEPS TO CREATE THE SMART WATER FOUNTAIN**

1. Set Up Raspberry Pi
2. Hardware Setup
3. Install Required Libraries
4. Code the Fountain Control
5. IoT Integration
6. Web Interface
7. Assemble and Test
8. Finalize and Deploy

**CODING**

# Define the code to run in the Arduino

arduino\_code = """

#include <Ultrasonic.h>

Ultrasonic ultrasonic(2, 3); // Trigger (pin 2), Echo (pin 3)

void setup() {

Serial.begin(9600);

}

void loop() {

float distance = ultrasonic.Ranging(CM);

Serial.println(distance);

// Send data to the computer (Python script)

Serial.print("D:");

Serial.println(distance);

delay(1000);

}

"""

# Upload and run the code in the simulation

simulation.run\_code(arduino\_code)

# Monitor the water level and send data to ThingSpeak

while True:

data = simulation.get\_serial\_data()

if data and data.startswith("D:"):

distance = float(data[2:])

print(f"Water level: {distance} cm")

# Send data to ThingSpeak

try:

response = requests.get(f"{thingspeak\_url}&field1={distance}")

if response.status\_code == 200:

print("Data sent to ThingSpeak successfully.")

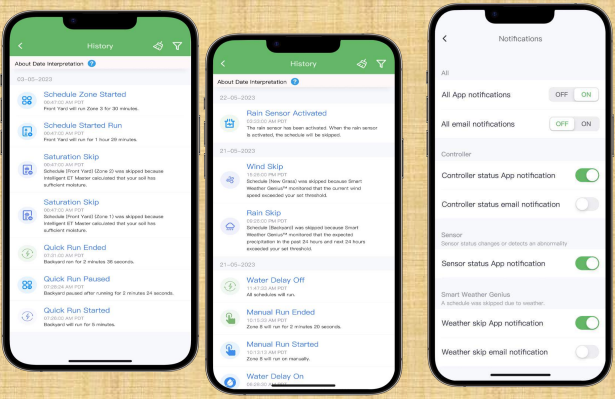
else:

print("Failed to send data to ThingSpeak.")

except Exception as e:

print("Error sending data to ThingSpeak:", str(e))

**DESIGN**

****

****

**DRAWBACKS OF WATER FOUNTAINS**

* Waste of water resources.
* Manpower is need to operate.
* Difficult to find errors.
* Different style
* Purpose of fountain to flow