PROJECT TITLE: SMART WATER FOUNTAIN

PHASE 3: DEVELOPMENT PART-1

INTRODUCTION:

A smart water fountain using IoT, or the Internet of Things, is a modern and innovative device designed to provide a more efficient and convenient way to manage and monitor the water supply in various contexts. Here's a brief introduction to the concept:

1. *IoT Integration*: The heart of a smart water fountain is its integration with IoT technology. This enables the fountain to be connected to the internet and to other devices, allowing for remote control and data collection.

2. *Remote Monitoring*: With IoT, you can monitor the status of the water fountain in real-time from anywhere, using a smartphone or computer. This makes it easier to ensure the fountain is working correctly and efficiently.

COMPONENTS REQUIRED:

RASPBERRY Pi

Ultrasonic Sensor

Water pump

LED

Resistor

HARDWARE DEVELOPMENT:

Developing the hardware for a smart fountain using IoT involves integrating various components to enable connectivity, monitoring, and control. Here's an overview of the key hardware elements required for such a project:

1. *Water Fountain Unit*:

- *Fountain Structure*: This is the physical structure of the fountain, which includes the reservoir, pipes, and nozzles.

- *Pump*: A water pump is used to circulate and deliver water to the nozzles.

- *Water Filter*: To ensure clean water, a filtration system is essential, and this may include UV sterilization or other purification methods.

- *Sensors*: Water level sensors, temperature sensors, and water quality sensors (pH, turbidity) are integrated to monitor various parameters.

2. *Microcontroller/Processor*:

- A microcontroller (e.g., Arduino, Raspberry Pi) serves as the brain of the system. It manages sensor data, controls actuators, and connects to the internet.

3. *IoT Connectivity*:

- *Wi-Fi/Bluetooth Module*: To enable wireless communication, you can integrate Wi-Fi or Bluetooth modules.

- *IoT Communication Protocols*: MQTT, HTTP, or CoAP can be used to transmit data to the cloud.

4. *Power Supply*:

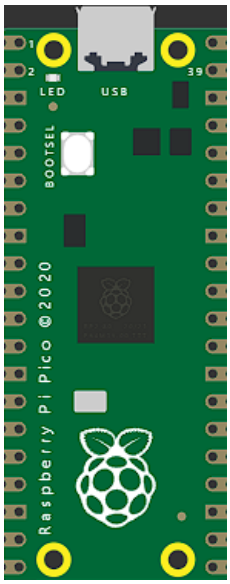
- Ensure a reliable and safe power source for the fountain and the IoT components. This may involve using a combination of mains power and battery backup.

5. *Actuators*:

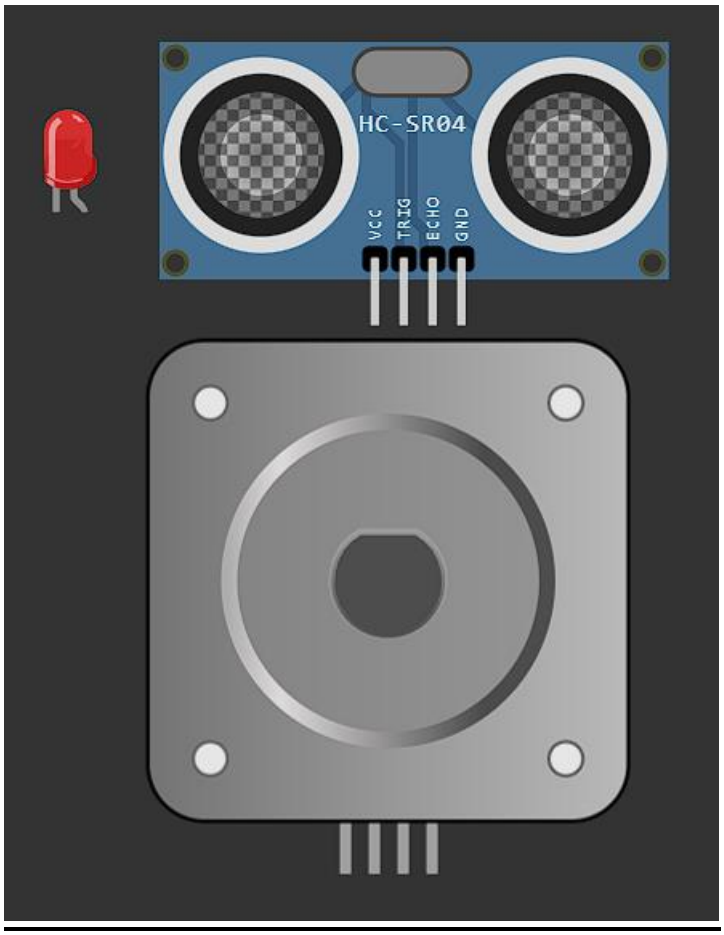
- *Valves and Solenoids*: These are used to control water flow and direction within the fountain.

- *LED Lights*: For decorative and lighting effects, integrated LED lights can be controlled via IoT.

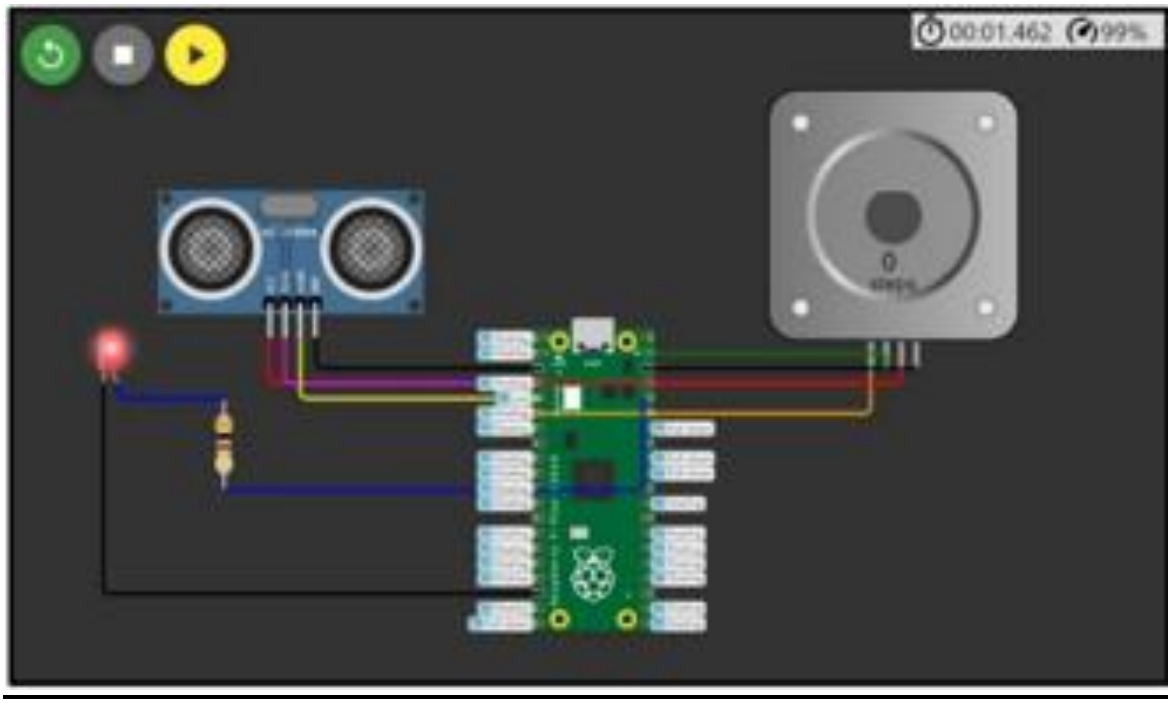
STEP-1:



STEP-2:



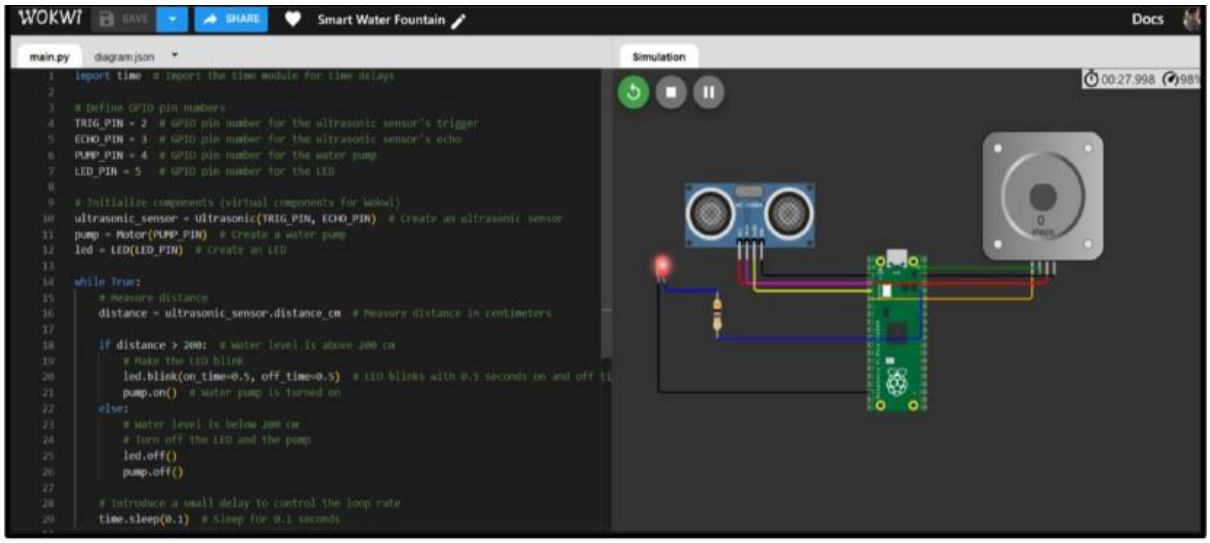
STEP-3:

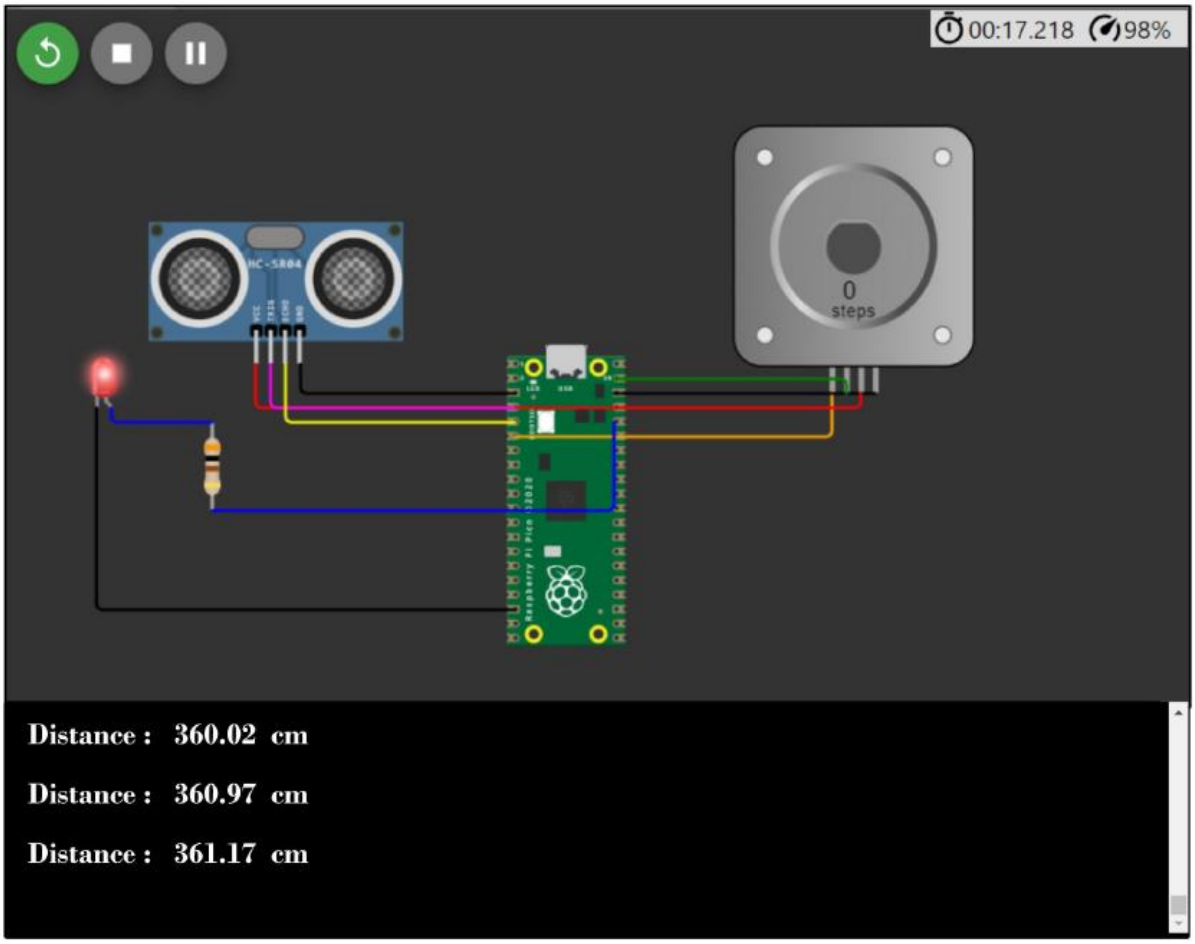


CODE:

```
import time
TRIG_PIN 2
ECHO PIN=3
PUMP PIN=4
LED PIN=5
ultrasonic_sensor = Ultrasonic(TRIG PIN, ECHO PIN)
pump Motor (PUMP_PIN)
led LED(LED_PIN
while True:
distance = ultrasonic_sensor.distance_cm
if distance> 200:
led.blink(on_time=0.5, off_time=0.5)
pump.on() # Water pump is turned on
else:
led.off()
pump.off()
time.sleep(0.1)
```

OUTPUT:


WOKWI
SAVE
SHARE
Smart Water Fountain
Docs
main.py
diagram.json
Simulation
00:27.998
import time # Import the time module for time delays
# Define GPIO pin numbers
TRIG_PIN = 2 # GPIO pin number for the ultrasonic sensor's trigger
ECHO_PIN = 3 # GPIO pin number for the ultrasonic sensor's echo
PUMP_PIN = 4 # GPIO pin number for the water pump
LED_PIN = 5 # GPIO pin number for the LED
# Initialize components (virtual components for Wokwi)
ultrasonic_sensor = Ultrasonic(TRIG_PIN, ECHO_PIN) # Create an ultrasonic sensor
pump = Motor(PUMP_PIN) # Create a water pump
led = LED(LED_PIN) # Create an LED
while True:
# Measure distance
distance = ultrasonic_sensor.distance_cm # Measure distance in centimeters
if distance > 200: # Water level is above 200 cm
Make the LED blink
led.blink(on_time=0.5, off_time=0.5) # LED blinks with 0.5 seconds on and off ti
pump.on() # Water pump is turned on
else:
# Water level is below 200 cm
# Turn off the LED and the pump
led.off()
pump.off()
# Introduce a small delay to control the loop rate
time.sleep(0.1) # Sleep for 0.1 seconds


00:17.218
98%
HC-SR04
0
steps
Distance : 360.02 cm
Distance : 360.97 cm
Distance : 361.17 cm