

IBM PROJECT

Project name: Smart Water Fountain system

Project Title : Smart Water Fountain system

Name : J.Ida Selvam Magdaline
Reg No. : 950421104019
Department : Computer Science And Engineering
College Name : Dr. G.U. Pope College of Engineering
College Code : 9504
Course Name : IoT (Internet of Things)

Phase 3

```
import BlynkLib  
import BlynkTimer  
import random
```

```
BLYNK_AUTH = 'Your_Blynk_Auth-Token' # Replace with your  
Blynk authentication token
```

```
# Initialize Blynk and BlynkTimer  
blynk = BlynkLib.Blynk(BLYNK_AUTH)  
timer = BlynkTimer.Timer()
```

```
# Virtual pins for Blynk app  
WATER_PUMP_CONTROL = 1  
DISTANCE_SENSOR_READING = 2
```

```
# Simulated distance sensor reading  
def simulate_distance_sensor():  
    return random.randint(5, 50) # Simulated distance (5 cm to  
    50 cm)
```

```

# Send simulated distance sensor reading to Blynk
@timer.register(vpin=DISTANCE_SENSOR_READING)
def send_distance_sensor_reading(vpin):
    distance = simulate_distance_sensor()
    blynk.virtual_write(DISTANCE_SENSOR_READING, distance)

# Control the water pump based on Blynk button
@blynk.handle_event('write V1')
def water_pump_control(pin, value):
    if int(value[0]) == 1: # If the button is pressed
        print("Water pump is ON")
    else:
        print("Water pump is OFF")

# Start the Blynk Timer
timer.set_interval(1000, send_distance_sensor_reading)

try:
    while True:
        blynk.run()
        timer.run()
except KeyboardInterrupt:
    Pass

```

Explanation:

This script simulates the control of a water pump and the measurement of water levels. However, this code is for simulation purposes and uses random values for distance readings.

1. It imports the required libraries, including BlynkLib and BlynkTimer.

2. Replace ``_Blynk_Auth-Token`` with the actual authentication token get from the Blynk app. This token is used to authenticate device with the Blynk cloud platform.
3. The script sets up a Blynk object and a timer for periodic tasks.
4. Virtual pins are defined for Blynk communication. IV1 is used for controlling the water pump, and V2 is used for sending distance sensor readings.
5. ``simulate_distance_sensor()`` is a function that simulates distance readings between 5 cm and 50 cm. In a real-world application, replace this function with actual sensor readings from an ultrasonic distance sensor.
6. The ``send_distance_sensor_reading`` function is registered with the Blynk timer to periodically send the simulated distance readings to the Blynk app.
7. The ``water_pump_control`` function is an event handler that listens for changes on V1 in the Blynk app. When the button widget in the app is pressed (`value[0] == 1`), it prints "Water pump is ON." When the button is released (`value[0] == 0`), it prints "Water pump is OFF." In a real-world application, you use this function to control an actual water pump.
8. The timer is started with a 1-second interval to send simulated distance sensor readings to the Blynk app.
9. The script enters a loop where it continuously runs the Blynk and timer processes. This allows the script to interact with the Blynk app and handle sensor readings and water pump control.

