Phase 3: Development Part

Here are some steps to help for deploying IOT devices get started:

1.Define Project Goals: Determine the specific functionalities you want your smart water fountain to have. This could include monitoring water flow, detecting malfunctions, or implementing automated features.

2.Choose the Right Sensors: Select sensors based on your project requirements. Flow rate sensors and pressure sensors are crucial for monitoring water flow and pressure accurately.

3.Select a Microcontroller: Choose a microcontroller board compatible with your sensors and connectivity requirements. Arduino and Raspberry Pi are popular choices for IoT projects.

4.Connectivity: Decide on the communication protocols (Wi-Fi, Bluetooth, LoRa, etc.) for connecting your devices to the internet and to each other.

5.Programming: Write code to collect data from sensors, process it, and send relevant information to a cloud platform or a local server for further analysis.

6.Cloud Integration: If you want to monitor your smart water fountain remotely or store data for analysis, integrate your IoT devices with a cloud platform like AWS IoT, Google Cloud IoT, or IBM Watson IoT.

7.Data Analysis and Visualization: Implement data analysis and visualization tools to interpret the data collected by your sensors. This can help in identifying patterns and trends.

8.Testing and Iteration: Test your setup thoroughly to ensure all components are working as expected. Iterate on your design if necessary to improve functionality and efficiency.

Sensor Unit:

This block contains the four 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 quantity.

For the PH-value sensor, temperature sensor and conductivity sensor, values will be retrieved and calculated to determine the overall water quality level. When poor water quality is determined, the water replacement procedures will take place. The weight sensor readings will be used to determine the amount of fresh water left in the water tank.

These sensors can be explained as follows,

<u>1.Temperature Sensor:</u>

A water-proof temperature sensor is going to be used. Part number from sparkfun is: DS18B20. This temperature sensor is compatible with a relatively wide range of power supply from 3.0V to 5.5V. The measured temperature ranges from -55 to +125 celsius degrees. Between -10 to + 85 degrees, the accuracy is up to +-0.5 degrees. This sensor can fulfill all requirements needed for this project.

<u>2.PH-sensor:</u>

PH value is a valued indicator of water quality. This PH-sensor works with 5V voltage, which is also compatible with the temperature sensor. It can 6measure the PH value from 0 to 14 with an accuracy of +- 0.1 at the temperature of 25 degrees.

<u>3.Conductivity sensor:</u>

Conductivity sensor is also part of the water quality assessment. The input voltage is from 3.0 to 5.0V. The error is small, +-5%F.S. The measurement value ranges from 0 to 20 ms/cm which is enough for water quality monitoring.

<u>4.Liquid Level Sensor:</u>

This sensor is responsible for reflecting how much freshwater is left in the water tank. When the water level is low, fresh water will be pumped to the water tank to ensure the water fountain keeps running with freshwater. This sensor is 0.5 Watts. For water level from 0 to 9 inches, the corresponding sensor outputs readings from 0 to 1.6. From that, the quantity of freshwater left can be determined.

Python Script:

```
import paho.mqtt.client as mqtt

import random

import time

# MQTT Broker (Platform) Details

broker_address = "mqtt.yourplatform.com"

port = 1883

topic = "water-fountain-status"

# Dummy IoT Sensor Data Generation

def generate_sensor_data():

    # You can replace this with actual sensor data collection logic

    fountain_status = random.choice(["on", "off"])

    return fountain_status

# MQTT on_connect Callback

def on_connect(client, userdata, flags, rc):

    print("Connected with result code "+str(rc))

    client.subscribe(topic)

# MQTT on_publish Callback

def on_publish(client, userdata, mid):

    print("Message Published")

# Main MQTT Client
```

```
client = mqtt.Client()

client.on_connect = on_connect

client.on_publish = on_publish

# Connect to MQTT Broker

client.connect(broker_address, port, 60)

try:

    while True:

        # Get Water Fountain Status

        fountain_status = generate_sensor_data()

        # Publish Data to Platform

        client.publish(topic, fountain_status)

        print("Water Fountain Status Published: " + fountain_status)

        # Wait for Some Time (e.g., 1 minute)

        time.sleep(60)

except KeyboardInterrupt:

    print("Script Terminated")

    client.disconnect()
```