Phase 3: Development Part 1_Smart Water Fountains

Creating a complete Python script for IoT sensors to send real-time water fountain status data to an IoT platform requires knowledge of the specific IoT platform you are using, the sensor hardware, and the communication protocol you want to implement (e.g., MQTT or HTTP). Below, I'll provide a general Python script template to help you get started. You'll need to adapt this script to your specific hardware and IoT platform.

I'll use the Paho MQTT library to send data to an MQTT broker. Please adjust the script according to your actual setup.

Python code:

```
import time

import random  # For generating dummy data (you should replace this with real sensor data)

import paho.mqtt.client as mqtt


# MQTT Broker settings (replace with your IoT platform details)

MQTT_BROKER_HOST = "your-broker-hostname"

MQTT_BROKER_PORT = 1883

MQTT_TOPIC = "water-fountain-status"

MQTT_USERNAME = "your-username"

MQTT_PASSWORD = "your-password"


# Initialize the MQTT client

client = mqtt.Client()


# Connect to the MQTT broker

client.username_pw_set(MQTT_USERNAME, MQTT_PASSWORD)

client.connect(MQTT_BROKER_HOST, MQTT_BROKER_PORT)
```

```
# Function to simulate sensor data (replace with real sensor data acquisition)
def read_sensor_data():
    # Replace this with actual code to read sensor data
    flow_rate = random.uniform(0.1, 10.0)  # Example: Random flow rate between 0.1 and 10.0 liters per minute
    pressure = random.uniform(20.0, 100.0)  # Example: Random pressure between 20.0 and 100.0 PSI
    return {"flow_rate": flow_rate, "pressure": pressure}

# Main loop to publish sensor data
while True:
    sensor_data = read_sensor_data()
    client.publish(MQTT_TOPIC, payload=str(sensor_data))
    print(f"Published data: {sensor_data}")
    time.sleep(5)  # Adjust the frequency of data transmission as needed
```

Here's a brief explanation of the code:

Import necessary libraries, including paho.mqtt.client for MQTT communication.

Set your MQTT broker settings (host, port, topic, username, and password). Replace these values with your IoT platform details.

Initialise the MQTT client and connect to the broker.

Create a function (read_sensor_data) that simulates sensor data. In practice, replace this with actual code to read data from your flow rate and pressure sensors.

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.

Temperature Sensor:

A water-proof temperature sensor is going to be used. Part number from sparkfun is: DS18B20 [6]. 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.

PH-sensor:

PH value is a valued indicator of water quality. This PH-sensor[7] 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.

Conductivity sensor:

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. [8]

Liquid Level Sensor:

This sensor [9] 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.