

Smart Water Fountain

IOT Sensor to find monitor water flow and detect malfunction:

Flow rate sensors, pressure sensors, leak detection sensor.

Python script for flow rate sensor:

```
import RPi.GPIO as GPIO

import time

import paho.mqtt.client as mqtt

FLOW_RATE_PIN = 17

GPIO.setmode(GPIO.BCM)

GPIO.setup(FLOW_RATE_PIN, GPIO.IN,
pull_up_down=GPIO.PUD_UP)

MQTT_BROKER = "your_mqtt_broker_address"

MQTT_PORT = 1883

MQTT_TOPIC = "water_fountain/flow_rate"

client = mqtt.Client()

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

    print("Connected to MQTT broker with result code " + str(rc))
```

```
client.on_connect = on_connect

client.connect(MQTT_BROKER, MQTT_PORT, 60)

flow_rate = 0

total_volume = 0

last_pulse_time = time.time()

def pulse_callback(channel):

    global flow_rate, total_volume, last_pulse_time

    current_time = time.time()

    pulse_duration = current_time - last_pulse_time

    flow_rate = 1.0 / pulse_duration # Flow rate in pulses per second

    last_pulse_time = current_time

    total_volume += 1 # Adjust for your specific flow rate calculation

GPIO.add_event_detect(FLOW_RATE_PIN, GPIO.FALLING,
callback=pulse_callback)

try:

    while True:

        client.publish(MQTT_TOPIC, f"Flow Rate: {flow_rate} pulses per
second")

        print(f"Flow Rate: {flow_rate} pulses per second")

        time.sleep(5) # Adjust the frequency of updates as needed

except KeyboardInterrupt:
```

```
GPIO.cleanup()
```

```
client.disconnect()
```

Python script for pressure sensor:

```
pip install RPi.GPIO paho-mqtt
```

```
import RPi.GPIO as GPIO
```

```
import paho.mqtt.client as mqtt
```

```
import time
```

```
PRESSURE_SENSOR_PIN = 17
```

```
GPIO.setmode(GPIO.BCM)
```

```
GPIO.setup(PRESSURE_SENSOR_PIN, GPIO.IN)
```

```
MQTT_BROKER = "your_mqtt_broker_address"
```

```
MQTT_PORT = 1883
```

```
MQTT_TOPIC = "water_fountain/pressure"
```

```
client = mqtt.Client()
```

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

```
    print("Connected to MQTT broker with result code " + str(rc))
```

```
client.on_connect = on_connect
```

```
client.connect(MQTT_BROKER, MQTT_PORT, 60)
```

```
try:
```

```
    while True:
```

```
        pressure_value = GPIO.input(PRESSURE_SENSOR_PIN
```

```

        status_data = {
            "pressure": pressure_value,
            "timestamp": time.time()
        }

        client.publish(MQTT_TOPIC, json.dumps(status_data))
        print("Published data to MQTT:", status_data)
        time.sleep(5)
except KeyboardInterrupt:
    GPIO.cleanup()
    client.disconnect()

```

Python script for leak detection sensor:

```

import RPi.GPIO as GPIO

import paho.mqtt.client as mqtt

import time

LEAK_SENSOR_PIN = 17

GPIO.setmode(GPIO.BCM)

GPIO.setup(LEAK_SENSOR_PIN, GPIO.IN)

MQTT_BROKER = "your_mqtt_broker_address"

MQTT_PORT = 1883

MQTT_TOPIC = "water_fountain/leak_status"

client = mqtt.Client()

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

```

```
print("Connected to MQTT broker with result code " + str(rc))

client.on_connect = on_connect

client.connect(MQTT_BROKER, MQTT_PORT, 60)

try:

    while True:

        if GPIO.input(LEAK_SENSOR_PIN) == GPIO.HIGH:

            leak_status = "Leak Detected"

        else:

            leak_status = "No Leak"

        status_data = {

            "leak_status": leak_status,

            "timestamp": time.time()

        }

        client.publish(MQTT_TOPIC, json.dumps(status_data))

        print("Published leak status to MQTT:", status_data)

        time.sleep(5)

except KeyboardInterrupt:

    GPIO.cleanup()

    client.disconnect()
```

Conclusion:

Thus the python script for flow rate sensor, pressure sensor, leak detection sensor are implemented successfully

