Smart Water Fountains

Development Part-1

Date	**10-10-2023**
Team ID	**509**
Project Name	**Smart Water Fountains**

Table of Contents

1. Introduction

Optimizing the efficiency of smart water fountains is crucial to promote water conservation and responsible usage. In this document, we will outline a Internet of Things (IoT) project focused on maximizing the efficiency of smart water fountains. We'll define the problem statement, outline the steps involved, and discuss how an automated approach can benefit water management and sustainability efforts.

2. Problem Statement

It is hard for us to optimize the water fountain to minimize the water usage based on various factors. We going to develop a model that optimizes the water usage of smart water fountains based on various factors such as foot traffic, weather, and time of day.

3. Microcontroller Selection:

The Raspberry Pi Pico is a remarkable microcontroller board created by the Raspberry Pi Foundation, designed to meet the diverse needs of embedded systems and Internet of Things (IoT) projects. At its heart, it employs the powerful RP2040 microcontroller chip, featuring dual ARM Cortex-M0+ cores. This provides substantial processing capability while offering extensive GPIO pins, making it adaptable to a wide range of applications. Whether you are a

beginner or an experienced developer, the Pico accommodates your programming preferences by supporting both MicroPython and C/C++.

One of the standout features of the Pico is its exceptional input/output (I/O) capabilities, encompassing UART, SPI, I2C, and PWM, enhancing its utility for various projects. Its compact form factor, low power consumption, and cost-effectiveness make it an attractive choice for projects ranging from educational endeavors to hobbyist creations and even commercial applications.

The Pico's support for MicroPython simplifies programming and allows for rapid development. Furthermore, it benefits from the extensive Raspberry Pi community, providing access to a wealth of resources, tutorials, and projects. Whether you are creating a sensor-based IoT device, a robotics project, or an interactive art installation, the Raspberry Pi Pico stands out as a versatile and cost-efficient microcontroller board suitable for an array of innovative and creative undertakings.

3.2. Sensors Selection

The ultrasonic level sensor, a crucial component in water fountains, employs ultrasonic sound waves to measure the distance between the sensor and the water surface. Emitting high-frequency sound pulses and timing their return, it calculates the water level accurately. This non-contact technology ensures precision, reliability, and longevity in water management systems. It is ideal for maintaining water reservoir levels, preventing overflow, and enabling automated water refilling. The sensor's ability to provide real-time data contributes to efficient water fountain operations, offering both convenience and conservation. Its widespread use in water fountains exemplifies its significance in optimizing water features, making them not only visually appealing but also smartly managed.

A water fountain's purity sensor is a vital component that assesses the quality of the water it circulates. These sensors often employ various technologies like electrical conductivity or turbidity measurement to detect impurities, particles, or contaminants in the water. By continuously monitoring water quality, the sensor ensures that the fountain dispenses clean and safe water, making it suitable for drinking or aesthetic purposes. If the sensor detects impurities beyond acceptable levels, it can trigger maintenance or purification processes to maintain water purity. This feature is particularly valuable in public places and establishments where water quality is crucial

for both health and aesthetics, guaranteeing a refreshing and hygienic water experience

3.Code

Language Used: Micro Python

MicroPython is a compact and efficient Python programming language implementation designed for microcontrollers and embedded systems. When integrated with the Raspberry Pi Pico, a microcontroller board equipped with the RP2040 microcontroller chip, it forms a user-friendly platform for IoT, robotics, and embedded projects. The Raspberry Pi Pico's impressive capabilities, including its dual ARM Cortex-M0+ cores and ample GPIO pins, make it an accessible choice for developers of all skill levels. With MicroPython support, programming the Pico becomes straightforward, offering a familiar environment for Python enthusiasts. Its compact size, affordability, and low power consumption make it a preferred choice for educational, hobbyist, and commercial projects, catering to a wide range of applications.

Code

```
from machine import Pin, PWM, Timer, ADC

from ultra import DistanceSensor
#We have used a separate .py file to read inputs from sensor for
reference I’m giving the Wokwi link here
https://wokwi.com/projects/379535806439664641

from time import sleep

ds = DistanceSensor(echo=14, trigger=15)

Fountain_motor = 2

motor_num=27

pin = machine.Pin(Fountain_motor, machine.Pin.OUT)

motor=machine.Pin(motor_num,machine.Pin.OUT)

while True:
    distance_cm = ds.distance * 100
    distance=float(distance_cm)
    print(f"Distance: {distance_cm} cm")
```

```
#to refill water in the reservoir automatically----------------------
----------
    if distance < 10:
          #denotes the distance btw sensor and water
            print("full")
            pin.on()
           #Here this “pin” represents the fountain activity, the
fountain will be on when the reservoir is full other wise this
device will first fill reservoir using “motor” pump
            motor.off()
           #to turn motor off
    else:
            print("filling ")
            pin.off()
            motor.on()
            #to turn motor on

#-----------------------------------------------------------------
------
    sleep(0.1)

#This code is to make the work automatically based on water level we
can also integrate many sensors and control remotely that will be
done in upcoming development part-2 ------------------------------
```

3. Working Principle

Control and Monitor:

With the software in place, We can now control and monitor your smart water fountain from a smartphone or computer. we can turn the fountain on/off remotely, check water levels, and receive alerts if water levels drop too low or if there are temperature issues.

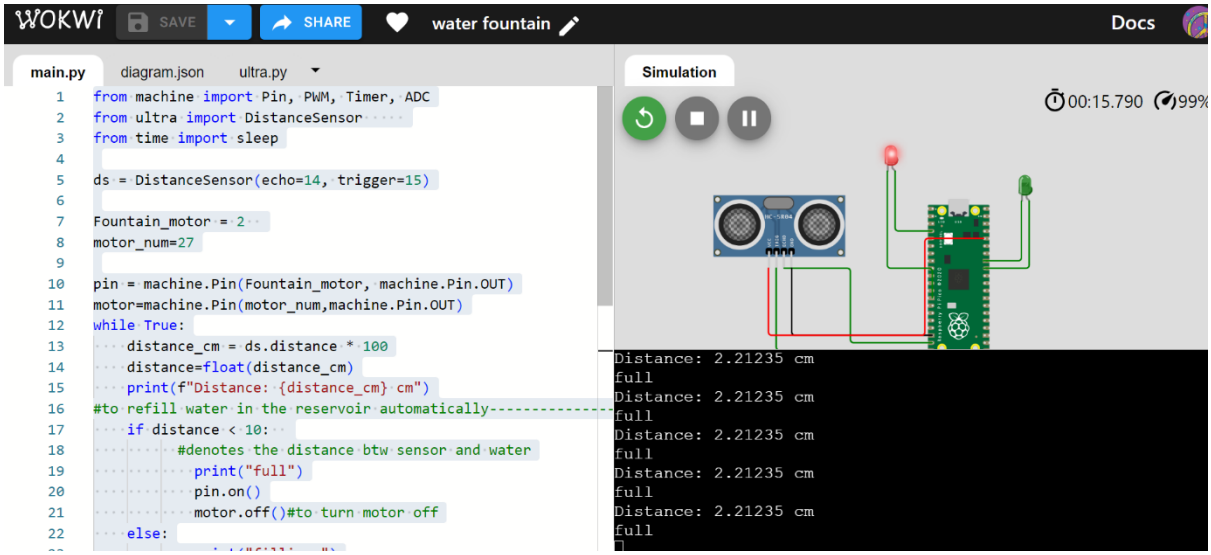
Automation:

We can automate our smart water fountain. For example, we could set schedules for the fountain to turn on and off automatically or create triggers based on environmental conditions.

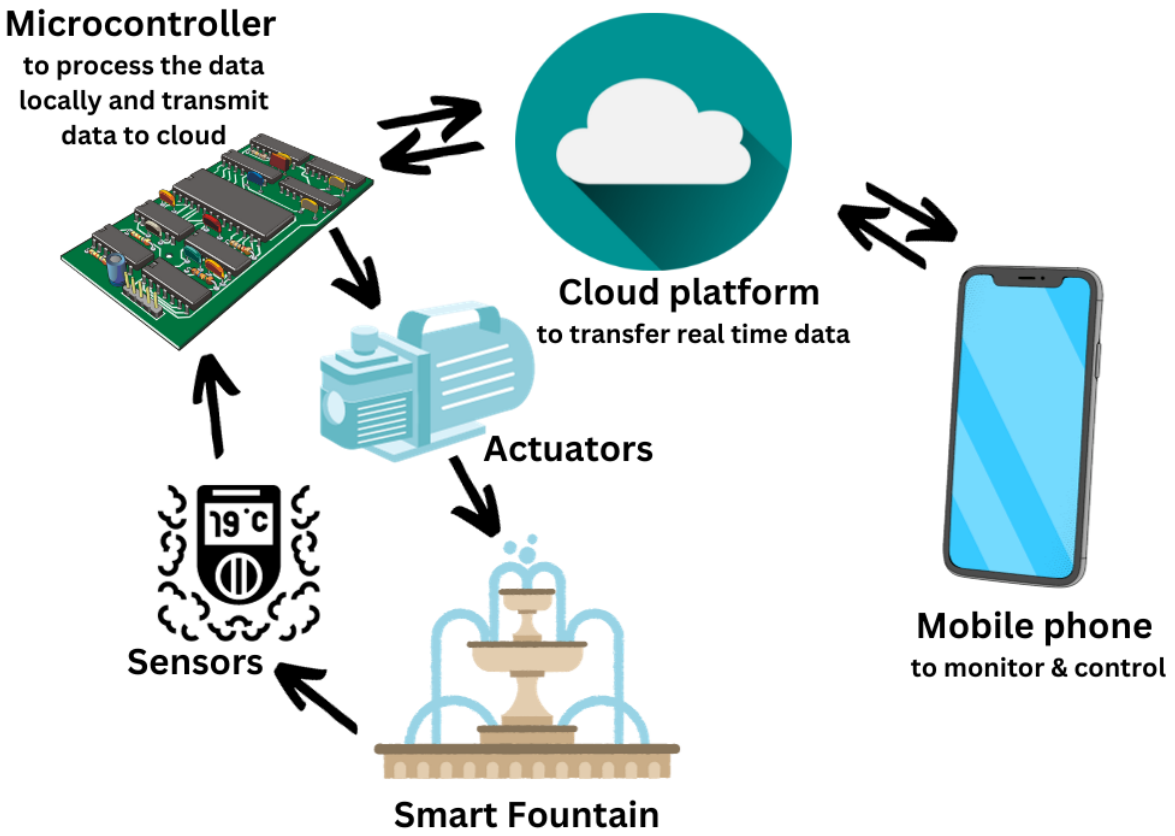
Enclosure and Maintenance:

Ensure that all components are housed in a waterproof enclosure to protect them from water damage. Regularly check and maintain the fountain, including cleaning the pump and sensors.

SIMULATION



3.7. Block Diagram



4. Conclusion

In this document, we have outlined a Internet of Things (IoT) project focused on optimizing water usage in smart water fountains to promote water conservation

and responsible resource usage. By following a design thinking approach, water management authorities and environmentalists can leverage automation, ultimately leading to efficient water usage, conservation, and a more sustainable future.