**Smart Water Management**

**Project Proposal: IoT Water Consumption Monitoring System**

# **Project Overview:**

The IoT Water Consumption Monitoring System is designed to address the growing concern of water conservation in public places, such as parks, schools, and municipal facilities. This system aims to accurately measure and monitor real-time water usage using IoT sensors (flow meters) and transmit the data to a centralized data-sharing platform. The project will enable water management authorities to efficiently manage water resources and encourage responsible water consumption practices.

## **Objectives**

1. Develop a reliable IoT solution for monitoring water consumption in public areas.
2. Implement IoT sensors to measure and monitor water flow and consumption in real-time.
3. Create a Python-based script to collect, process, and transmit water consumption data to a data-sharing platform.
4. Provide an intuitive user interface for data visualization and reporting.
5. Enhance water resource management and encourage conservation efforts.

## Scope

* Deployment of IoT sensors in selected public areas.
* Development of a Python script for data collection and transmission.
* Integration with a data-sharing platform.
* User-friendly data visualization for water consumption monitoring.
* Testing and validation of the system's accuracy and reliability

# **Hardware**

**Microcontrollers:**

They are used to interface with the IoT sensors, process data, and facilitate communication with the data-sharing platform.

**Used:** Arduino,Esp32

**Connectivity:**

**Wi-Fi Module**: Use Wi-Fi for data transmission.

# **Sensors**

We have used the following sensors for our project:

Temperature

Humidity

Water Flow

Water Meter

Water Level

Leak Detection

## **Their uses**

**Temperature sensor:**

It is used to measure the temperature of water in real-time. They are useful in detecting changes in temperature that could indicate a problem with the water supply or distribution system.

**Humidity sensor:**

It is used to measure the amount of moisture in the air. They can be used to detect leaks or other issues that could lead to mold growth or other problems.

**Water flow sensor:**

It is used to measure the rate of water flow in pipes and other water distribution systems. They can be used to detect leaks, monitor water usage, and optimize water distribution.

**Water meter:**

It is used to measure the amount of water consumed by households or businesses. They can be used to detect leaks, monitor water usage, and optimize water distribution.

**Water level sensor:**

It is used to measure the level of water in tanks, reservoirs, and other storage containers. They can be used to detect leaks, monitor water usage, and optimize water distribution.

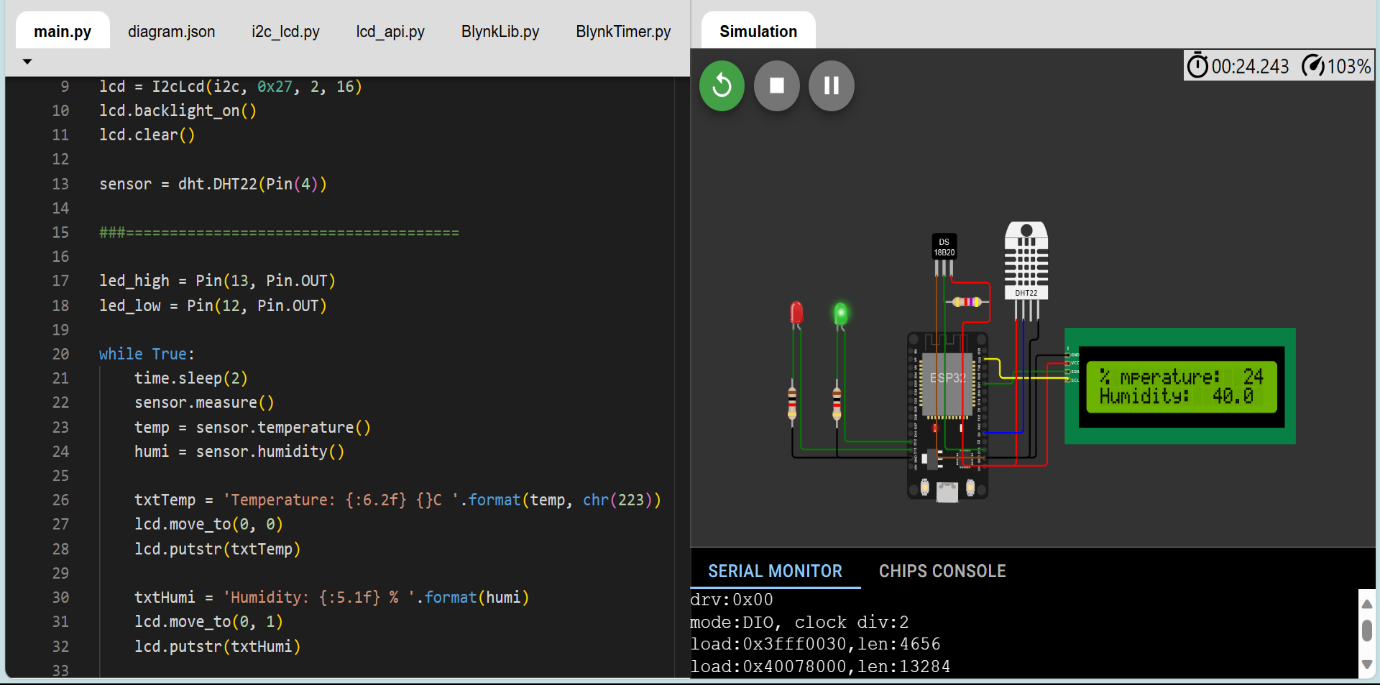
**Leak detection sensor:**

It is used to detect leaks in pipes and other water distribution systems. They can be used to prevent water loss and damage from leaks.

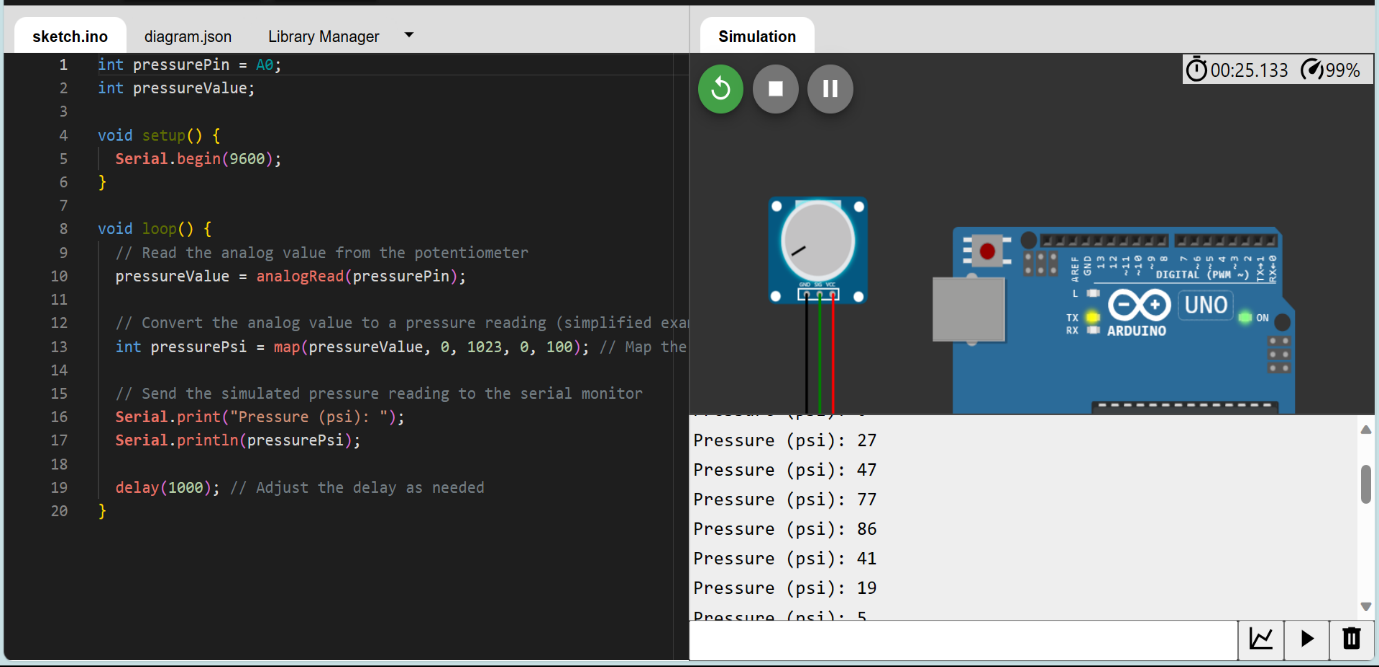
Configured Sensors:

Screenshots:

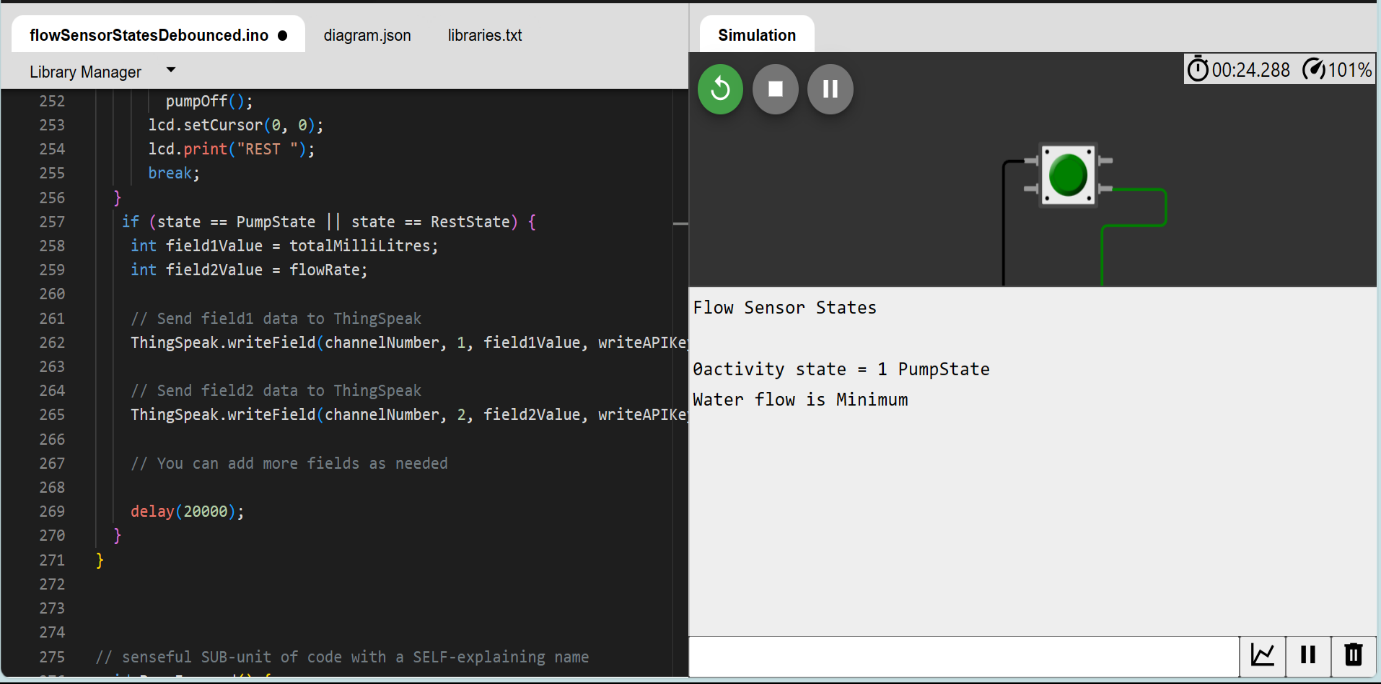
Temperature and Humidity



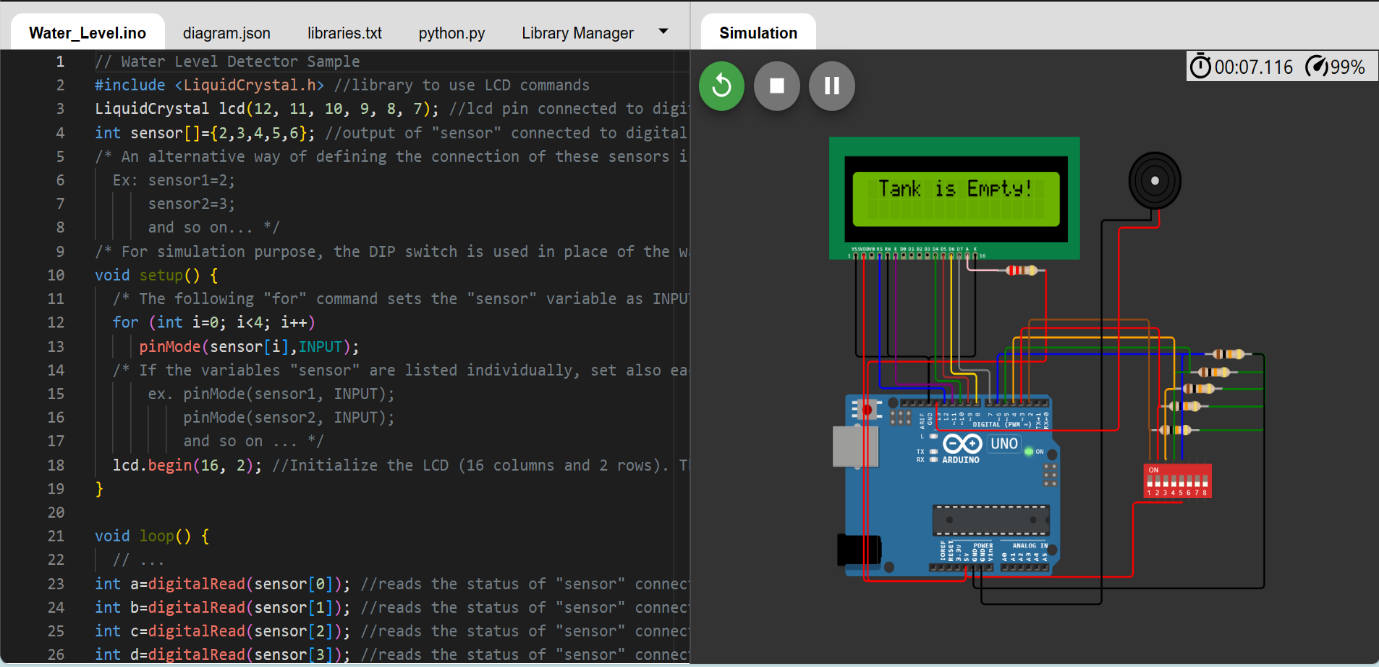
Pressure



Water Flow



Water Level



Codes used available here:

https://drive.google.com/drive/folders/1O8PMTsy5sHP-eAH0xma4c6pbRRohEGnz?usp=share\_link

# Script for data sharing

Python script for sharing data from Temperature and Humidity Sensor

import requests

import time

import json

thingspeak\_url = "https://api.thingspeak.com/update"

api\_key = "6EV4VJEM23TR6EO2"

ssid = "Wokwi-GUEST"

password = ""

DHT\_PIN = 15

TRIG\_PIN = 13

ECHO\_PIN = 12

def get\_distance():

  from machine import Pin

  import dht

  dht\_sensor = dht.DHT22(Pin(DHT\_PIN))

  while True:

    try:

        dht\_sensor.measure()

        temperature = dht\_sensor.temperature()

        humidity = dht\_sensor.humidity()

        distance = get\_distance()

        print("Temperature: {:.2f}°C, Humidity: {:.2f}%, Distance: {:.2f} cm".format(temperature, humidity, distance))

        data = {

            "api\_key": api\_key,

            "field1": temperature,

            "field2": humidity,

            "field3": distance

        }

        response = requests.post(thingspeak\_url, data=data)

        print("Data sent to ThingSpeak. Status code:", response.status\_code)

    except Exception as e:

        print("Error:", str(e))

    time.sleep(15)

Python script for sharing data from WaterFlow Sensor

import requests

import time

import random

channel\_id = "2306722"

write\_api\_key = "J7UB4P9UTY5Z206M"

thing\_speak\_url = f"https://api.thingspeak.com/update?api\_key={write\_api\_key}"

def simulate\_water\_flow\_data():

    return random.uniform(0, 10)

while True:

    try:

        water\_flow\_rate = simulate\_water\_flow\_data()

        data = {

            "field1": water\_flow\_rate

        }

        response = requests.post(thing\_speak\_url, data=data)

        print("Data sent to ThingSpeak. Status code:", response.status\_code)

    except Exception as e:

        print("Error:", str(e))

    time.sleep(15)

Python script for sharing data from Water Meter

import requests

import time

import random

channel\_id = "2306899"

write\_api\_key = "VMMZ4TEZVTV6UX8N"

thing\_speak\_url = f"https://api.thingspeak.com/update?api\_key={write\_api\_key}"

def simulate\_water\_meter\_data():

    return random.uniform(0, 1000)

while True:

    try:

        water\_consumption = simulate\_water\_meter\_data()

        data = {

            "field2": water\_consumption

        }

        response = requests.post(thing\_speak\_url, data=data)

        print("Data sent to ThingSpeak. Status code:", response.status\_code)

    except Exception as e:

        print("Error:", str(e))

    time.sleep(15)

Python script for sharing data from Water Level Sensor

import requests

import time

import machine

from machine import Pin

thingspeak\_url = "https://api.thingspeak.com/update"

channel\_id = "2306899"

api\_key = "VMMZ4TEZVTV6UX8N"

sensor\_pins = [2, 3, 4, 5, 6]

lcd = machine.LCD()

lcd.init()

def read\_sensor\_state():

    sensor\_states = [digitalRead(pin) for pin in sensor\_pins]

    return sensor\_states

def get\_water\_level(sensor\_states):

    if all(sensor\_states):

        return "Overflowing"

    elif sensor\_states[0]:

        return "Tank is Full"

    elif sensor\_states[1]:

        return "Tank is 75% Full"

    elif sensor\_states[2]:

        return "Tank is 50% Full"

    elif sensor\_states[3]:

        return "Tank is 25% Full"

    else:

        return "Tank is Empty"

while True:

    try:

        sensor\_states = read\_sensor\_state()

        water\_level = get\_water\_level(sensor\_states)

        lcd.text(water\_level)

        data = {

            "api\_key": api\_key,

            "field1": water\_level

        }

        response = requests.post(thingspeak\_url, data=data)

        print("Data sent to ThingSpeak. Status code:", response.status\_code)

    except Exception as e:

        print("Error:", str(e))

    time.sleep(15)

Python script for sharing data from Leak Detection Sensor

import requests

import time

import random

channel\_id = "2309000"

write\_api\_key = "28WR12W5X9OSKXY2"

thing\_speak\_url = "https://api.thingspeak.com/update?api\_key={write\_api\_key}"

def simulate\_leak\_data():

    return random.choice([0, 1])

while True:

    try:

        leak\_status = simulate\_leak\_data()

        data = {

            "field1": leak\_status

        }

        response = requests.post(thing\_speak\_url, data=data)

        print("Data sent to ThingSpeak. Status code:", response.status\_code)

    except Exception as e:

        print("Error:", str(e))

    time.sleep(15)

Python script for sharing data from Pressure Sensors

import requests

import time

import random

channel\_id = "2310757"

write\_api\_key = "OL8FZEGPB5KK8DCW"

thing\_speak\_url = f"https://api.thingspeak.com/update?api\_key={write\_api\_key}"

def simulate\_pressure\_data():

    return random.uniform(0, 100)

while True:

    try:

        pressure\_value = simulate\_pressure\_data()

        data = {

            "field1": pressure\_value

        }

        response = requests.post(thing\_speak\_url, data=data)

        print("Data sent to ThingSpeak. Status code:", response.status\_code)

    except Exception as e:

        print("Error:", str(e))

    time.sleep(15)