SMART WATER SYSTEM

Abstract:

This paper represents an IoT (Internet of things) based smart water quality monitoring (SWQM) system that aids in continuous measurement of water condition based on four physical parameters i.e., temperature, pH, electric conductivity and turbidity properties. A serious drop in ensuring the water quality in the distribution system is a factor that affects public health. This could lead to increase in biological and non-biological contents, change in color and odor of the water. These contaminants cause a serious threat to the whole water ecosystem. The conventional methods of analyzing the water quality requires much time and labor. So, there is a need to monitor and protect the water with a real time water quality monitoring system in order to make active measurements to reduce contamination. The growth of the technology had helped in developing efficient methods to solve many serious issues in real time. Internet of things (IoT) has achieved a great focus due to its faster processing and intelligence. This paper focus on discussing the architecture, applications and need of IoT in water management system.

Building Of Smart Water System:

A smart building uses its intelligence to collect actionable data from user devices, sensors, systems, and services on the premises. Applying that data using artificial intelligence and machine learning (AI/ML) makes the building both programmable and responsive to the needs of the users and the building manager.

Smart water systems allow the collection, treatment, distribution and recycling of water. These systems, often deployed underground, can leak, freeze, or breakdown. These systems are widely deployed on infrastructures nowadays.

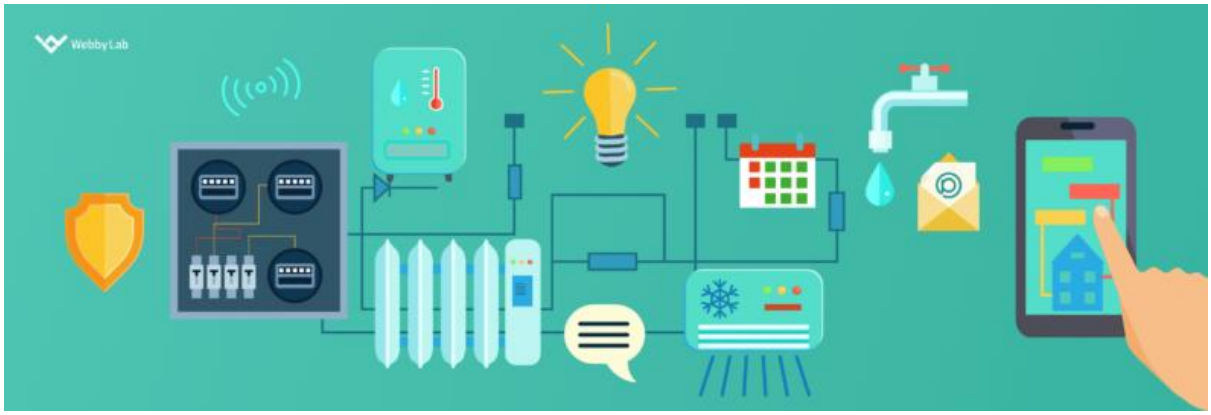
IoT water management systems leverage numerous sensors that collect real-time insights on how resources are used. These devices transmit the gathered data to the user's application online. This information empowers analysis of consumption patterns and encourages more rational water consumption.

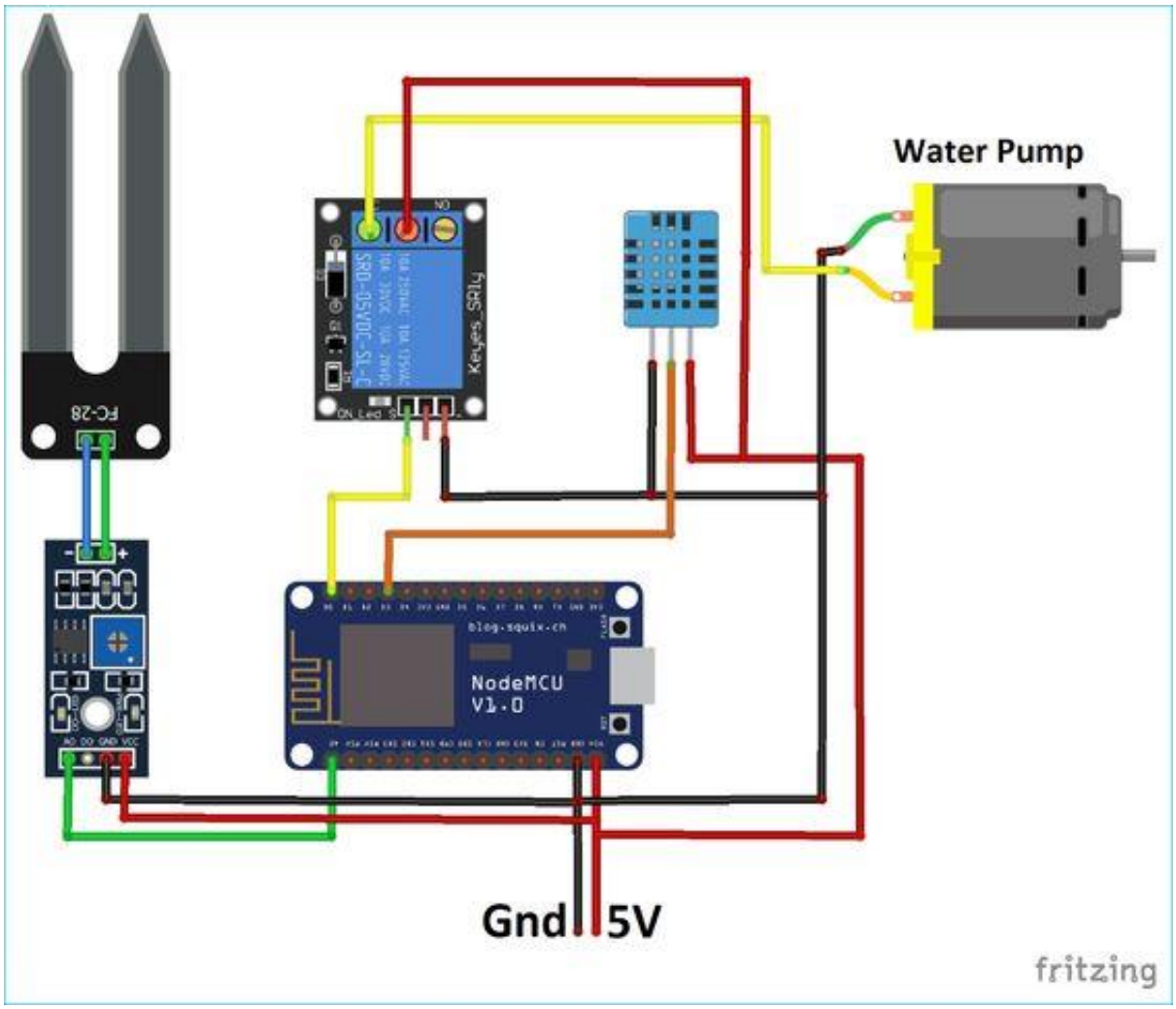
Smart water management aims to guide the utilisation of water in a manner that drives efficiency, sufficiency, and sustainability by integrating innovative technologies such as sensors, smart water metering, information systems, data acquisition and decision support systems.

A connected building management system (BMS) to control aspects such as heating and lighting. Sensors to monitor key factors like indoor air quality, occupancy and energy. A platform to access and visualize all this data.

A smart building offers the following features:

- Connecting building systems. ...
- Connecting people and technology. ...
- Controlling the building's expenditure. ...
- Centralized control of building behaviour. ...
- Connecting the building to a smart grid. ...
- Intelligent sensors IoT. ...
- System integration platform.





Python Script:

import RPi.GPIO as GPIO

import time

```
# Set up GPIO pins (for Raspberry Pi)
TRIG = 23
ECHO = 24

def setup_sensor():
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(TRIG, GPIO.OUT)
    GPIO.setup(ECHO, GPIO.IN)

def read_water_level():
    try:
        GPIO.output(TRIG, True)
        time.sleep(0.00001)
        GPIO.output(TRIG, False)

        while GPIO.input(ECHO) == 0:
            pulse_start = time.time()

        while GPIO.input(ECHO) == 1:
            pulse_end = time.time()

        pulse_duration = pulse_end - pulse_start
        distance = (pulse_duration * 34300) / 2  # Speed of sound = 343
m/s

        water_level = 10 - distance  # Adjust as needed based on your
sensor placement
```

```
        return water_level
    except KeyboardInterrupt:
        GPIO.cleanup()

if __name__ == "__main__":
    setup_sensor()
    while True:
        water_level = read_water_level()
        print(f"Water Level: {water_level} cm")
        time.sleep(10)  # Adjust the interval as needed
import RPi.GPIO as GPIO
import time

# Set up GPIO pins (for Raspberry Pi)
FLOW_SENSOR_PIN = 17

def setup_flow_sensor():
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(FLOW_SENSOR_PIN, GPIO.IN,
pull_up_down=GPIO.PUD_UP)

def count_water_usage():
    count = 0
    last_state = 0

    def flow_callback(channel):
        nonlocal count
        count += 1
```

```
    GPIO.add_event_detect(FLOW_SENSOR_PIN, GPIO.FALLING,
callback=flow_callback)


    try:
        while True:
            time.sleep(1)  # Count flow for 1 second intervals
            print(f"Water Usage (liters): {count}")
            count = 0
    except KeyboardInterrupt:
        GPIO.cleanup()

if __name__ == "__main__":
    setup_flow_sensor()
    count_water_usage()
```