SMART WATER FOUNTAIN

NAME:K.Santhosh

REG_NO:510421106042

Phase -5

Document submission

PROJECT: SMART WATER FOUNTAIN

CHAPTER 1: - INTRODUCTION

- Problem statement
- Statistics about the problem
- Infrastructure difficulties
- IoT components
- Functionalities
- Benefits

Problem statement:

Design and develop an intelligent, eco-friendly smart water fountain system that efficiently monitors water usage, promotes hydration, and minimizes wastage through innovative sensor technologies and user-friendly interfaces. Address the challenge of balancing user convenience with sustainable water conservation, ensuring seamless integration with modern urban environments and public spaces.

Statistics about the problem:

1. Market Growth: The smart water fountain market has been experiencing growth due to increasing awareness about sustainability and the need for efficient public water sources. Statistics might include market size, revenue projections, and growth rates.

2. Water Conservation: Smart water fountains are designed to conserve water. Statistics could include the average reduction in water usage compared to traditional fountains, demonstrating their impact on conservation efforts.

3. Usage Data:Smart water fountains can collect data on usage patterns. This could include the number of times the fountain is used per day, busiest times of usage, or average duration of use, which can be valuable for urban planning and maintenance.

4. User Engagement: Engagement statistics might cover aspects like user interactions with digital interfaces (if applicable), feedback given, or participation in awareness campaigns related to hydration and conservation.

5. Maintenance Efficiency: Smart water fountains often come with predictive maintenance features. Statistics might include a reduction in downtime due to predictive maintenance, showing how these fountains save both time and resources.

Infrastructure difficulties:

1. Power Supply: Smart water fountains require a stable power source for sensors, displays, and connectivity. Finding suitable and sustainable power solutions, especially in public spaces, can be challenging.

2. Internet Connectivity: Smart features often rely on internet connectivity for real-time monitoring and data analysis. Ensuring reliable internet access in outdoor or public locations can be difficult, leading to potential issues in data transmission and monitoring.

3. Maintenance: Smart water fountains involve complex systems, and regular maintenance is essential. Ensuring that maintenance staff are trained to handle both the traditional fountain components and the smart features is a challenge.

4. Vandalism and Security: Public infrastructure is susceptible to vandalism. Smart components can be targets for theft or damage, disrupting the functioning of the fountain. Security measures need to be in place to prevent such incidents.

5. Data Security and Privacy: Smart fountains collect data, which raises concerns about privacy and security. Ensuring that the data collected is secure, and user privacy is protected, often involves implementing robust cybersecurity measures.

6. Environmental Conditions: Outdoor smart fountains are exposed to various environmental conditions, including extreme temperatures, rain, and dust. Designing smart components that can withstand these conditions is a significant challenge.

7. Initial Costs: Implementing smart technology involves higher initial costs compared to traditional fountains. Securing funding and convincing in stakeholders about the long-term benefits can be difficult, especially if the advantages aren't immediately apparent.

8. Integration with Existing Infrastructure: Integrating smart water fountains with existing urban infrastructure, such as plumbing systems and drainage, can be complex. Ensuring compatibility and smooth integration without disrupting existing services is a challenge.

IoT components :

1. Sensors: IoT sensors, such as proximity sensors, motion detectors, and water flow sensors, are used to detect the presence of users, monitor water usage, and assess environmental conditions. These sensors enable the fountain to activate when someone approaches and collect data for analysis.

2. Microcontroller or Single-Board Computer: A microcontroller (e.g., Arduino, Raspberry Pi) processes data from sensors and controls the fountain's functions. It acts as the brain of the system, interpreting sensor inputs and triggering appropriate responses.

3. Connectivity Modules: IoT-enabled fountains require connectivity to transmit data and receive commands. Wi-Fi, Bluetooth, or cellular modules are integrated to establish connections with the internet or other devices, allowing remote monitoring and control.

4. User Interface: Smart water fountains often feature touchscreens or digital interfaces that allow users to interact with the fountain. These interfaces can display water quality information, hydration tips, and even advertisements. Users can also customize water flow settings through these interfaces.

5. Data Storage: Data collected by sensors, such as usage patterns and water quality metrics, need to be stored for analysis and future reference. IoT-enabled fountains have onboard storage or cloud-based solutions to store and manage this data securely.

6. Actuators: Actuators control the physical components of the fountain, such as water valves and pumps. IoT technology allows these actuators to be remotely controlled, enabling features like adjusting water pressure or turning off the fountain during non-peak hours to conserve water.

7. Power Management: Smart fountains require efficient power management systems to ensure continuous operation. This can include power-saving modes, solar panels for sustainable energy, and battery backup systems to prevent disruptions during power outages.

8. Remote Monitoring and Control: IoT components enable remote monitoring and control capabilities. This allows administrators to check the fountain's status, receive alerts for maintenance needs, and adjust settings remotely through web or mobile applications.

Functionalities:

1.Automated Dispensing:

Smart water fountains are equipped with sensors that detect the presence of users. When a user approaches the fountain, it automatically dispenses water, promoting easy and convenient access.

2. Customizable Settings:

Users can often customize the water temperature, flow rate, and quantity through digital interfaces. This ensures that individuals can have their preferred drinking experience.

3. Water Quality Monitoring:

Smart fountains can include sensors to monitor water quality in real-time. Parameters such as pH levels, temperature, and filtration status can be monitored and displayed, ensuring users are aware of the quality of the water they are consuming.

4. Usage Data Collection:

IoT components collect usage data, including the number of times the fountain is used, peak usage hours, and popular locations. This data is valuable for maintenance, urban planning, and resource allocation.

5. Predictive Maintenance:

By analyzing usage patterns and sensor data, smart fountains can predict maintenance needs. For instance, if a filter needs replacement or if there's a mechanical issue, the fountain can send alerts to maintenance personnel, ensuring timely repairs.

6. Water Conservation:

Smart water fountains are designed to minimize water wastage. They can incorporate features like automatic shut-off after a certain duration, preventing water from being wasted if a user forgets to turn off the fountain.

7. Hydration Reminders:

Some smart fountains include features to remind users to stay hydrated. These reminders can be based on user preferences, weather conditions, or health factors, encouraging regular water intake.

8. Accessibility Features:

Smart water fountains can be designed with accessibility in mind, including features such as different height levels, braille instructions, and audible cues, ensuring they are usable by a wide range of individuals, including those with disabilities.

9. Integration with Mobile Apps:

Integration with mobile applications allows users to find nearby smart water fountains, check water quality information, and even place requests for specific water preferences, enhancing user convenience.

10. Environmental Sustainability:

Smart water fountains often incorporate eco-friendly features such as water filtration systems, encouraging the use of reusable bottles, and promoting overall environmental consciousness.

Benefits:

1. Water Conservation: Smart water fountains are designed to minimize water wastage by providing water only when needed and shutting off automatically. This helps conserve water, especially in public places with high foot traffic.

2. Improved Hydration: By encouraging regular water intake through features like hydration reminders and customizable settings, smart fountains contribute to improved hydration levels among users.

3. Data-Driven Insights: Smart water fountains collect usage data and water quality information. Analysing this data provides valuable insights for urban planning, enabling authorities to optimize the placement of fountains and monitor water quality in real-time.

4. Environmental Sustainability: By promoting the use of reusable bottles and reducing single-use plastic waste, smart water fountains contribute to environmental sustainability and align with eco-friendly initiatives.

5. User Convenience: Customizable settings, automated dispensing, and intuitive interfaces enhance user convenience, making it easy for people to access clean and safe drinking water without hassle.

6. Health and Safety: Real-time water quality monitoring ensures that the water dispensed is safe for consumption. Users can check water quality information, promoting confidence in the water source's safety.

7. Cost Savings: Smart water fountains can lead to cost savings in the long run due to reduced water wastage and predictive maintenance. By identifying issues before they escalate, maintenance costs are minimized.

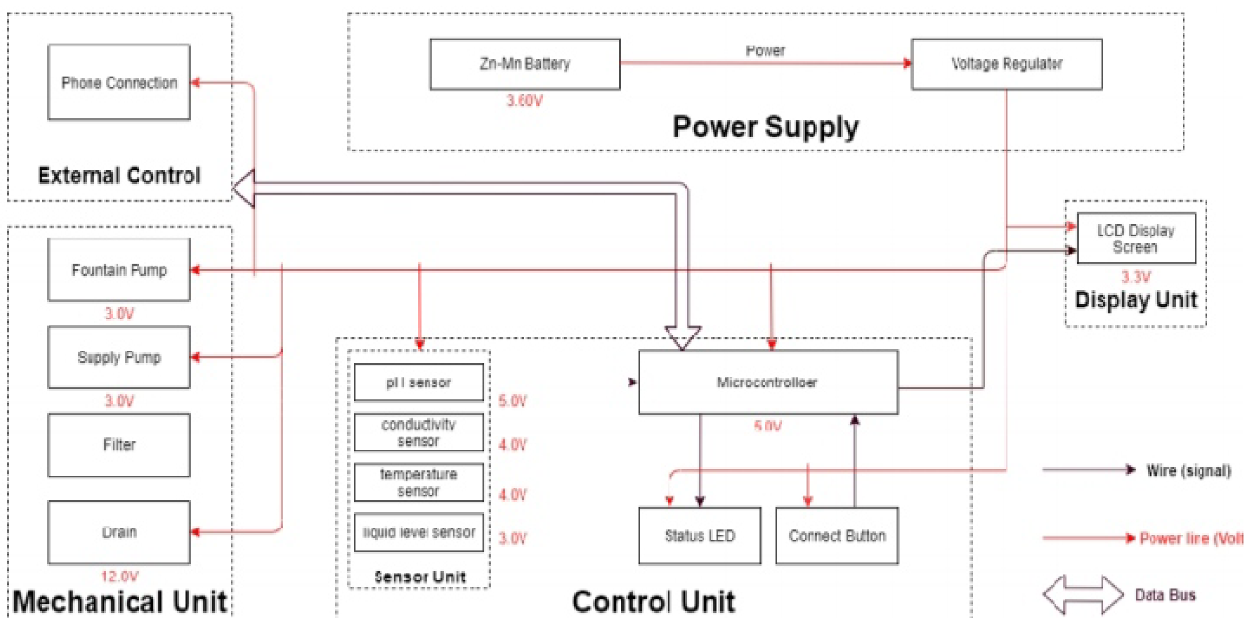
8. Accessibility: Smart water fountains can be designed to be accessible to people with disabilities. Features such as different height levels, braille instructions, and audible cues ensure inclusivity for all users.

9. Promotion of Healthy Habits: Hydration reminders and educational content displayed on digital interfaces can promote healthy hydration habits, leading to better overall health outcomes for users.

10. Community Engagement: Smart water fountains can be integrated with mobile apps, allowing community engagement through features like locating nearby fountains, tracking personal hydration goals, and participating in public health initiatives.

11. Real-time Alerts: Maintenance personnel receive real-time alerts about issues such as low water levels, filter replacements, or technical malfunctions. This proactive approach ensures timely maintenance and uninterrupted service.

CHAPTER 2:- BLOCK DIAGRAM AND ITS COMPONETS



SENSOR UNITS:

- 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

Mechanical Unit :

- Fountain pump :The fountain pump must maintain a continuous water supply through the fountain mechanism. The pump must work 24 hours a day, 7 days a week unless the user manually turns off the power supply.

Requirement 1: The fountain pump must lift a cylindrical water stream of diameter 6mm for a height of 400mm.

Requirement 2: The fountain pump must serve for a duration of 2 years without maintenance or replacement under heavy workload.

Requirement 3: The fountain pump should have an operational condition around 3V, 200mA.

- Supply Pump: The supply pump must function when a low water level alert is raised. While no water supply is requested, the pump must prevent water flow between the main supply and the fountain.

Requirement: The supply pump should have an operational condition around 3V, 200Ma

- Filter :The filter must maintain the water quality through controlling the pH value and conductivity of the water.

Requirement 1: The filter must have a cost less than $5 each for frequent replacement. Each new filter must serve a duration no less than 3 month.

Requirement 2: The filter must be designed for easy removal and installation, while the connection mechanism must have a low degenerate rate when submerged in water.

- Drain:The drain [13] must be able to hold and release water in the fountain. When water in the fountain should be replaced, the faucet should automatically drain the fountain once instruction is received from the integrated circuit.

CHAPTER 3:-DEVELOPMENT PROGRAMS

Creating a real-time smart water fountain platform involves a combination of front end and backend technologies. Here's a simplified outline using C and C++ and python programming with wi-fi connection for the front end and Node.

PYTHON:

```
# Setup GPIO

GPIO.setmode(GPIO.BCM)

Water_sensor_pin = 17

Pump_pin = 18


GPIO.setup(water_sensor_pin, GPIO.IN)

GPIO.setup(pump_pin, GPIO.OUT)

  While True:

    If GPIO.input(water_sensor_pin) == GPIO.LOW:

      Print("Water level is low. Turning on the pump.")

      GPIO.output(pump_pin, GPIO.HIGH)

    Else:

      Print("Water level is sufficient. Turning off the pump.")

      GPIO.output(pump_pin, GPIO.LOW)


    Time.sleep(5)  # Check water level every 5 seconds


Except KeyboardInterrupt:

  Print("Exiting…")

  GPIO.cleanup()
```

C++

```
#include <Arduino.h>
// Define pins for water pump and water level sensor.
Const int PUMP_PIN = 8;
Const int SENSOR_PIN = 9;
Void setup() {
 pinMode(PUMP_PIN, OUTPUT);
 pinMode(SENSOR_PIN, INPUT);
 Serial.begin(9600);
}
Void turnOnPump() {
 digitalWrite(PUMP_PIN, HIGH);
}
Void turnOffPump() {
 digitalWrite(PUMP_PIN, LOW);
}
Bool isWaterLevelLow() {
 Return digitalRead(SENSOR_PIN) == LOW;
}
Void loop() {
 If (isWaterLevelLow()) {
  Serial.println(“Water level is low. Turning on the pump.”);
  turnOnPump();
  Delay(5000); // Run the pump for 5 seconds (adjust as needed).
  turnOffPump();
 }
 Delay(1000); // Check water level periodically.
}
```

C PROGRAM

```
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>  // You'll need to install WiringPi library for Raspberry Pi or similar.
// Define GPIO pins for water pump and water level sensor.
#define PUMP_PIN 17
#define SENSOR_PIN 18

Void setup() {
   wiringPiSetupGpio();  // Initialize the WiringPi library.

   pinMode(PUMP_PIN, OUTPUT);
   pinMode(SENSOR_PIN, INPUT);
}

Void turnOnPump() {
   digitalWrite(PUMP_PIN, HIGH);
}

Void turnOffPump() {
   digitalWrite(PUMP_PIN, LOW);
}

Int isWaterLevelLow() {
   Return digitalRead(SENSOR_PIN) == LOW;
}

Int main() {
   If (wiringPiSetupGpio() == -1) {
      Fprintf(stderr, "Unable to initialize WiringPi. Exiting.\n");
```

```
    Return 1;
  }

  Setup();

  While (1) {
    If (isWaterLevelLow()) {
      turnOnPump();
      delay(5000);  // Run the pump for 5 seconds (adjust as needed).
      turnOffPump();
    }
  }

  Return 0;
}
```

MICROPROCESSOR PROGRAM:

```assembly
ORG 0x1000     ; Set the origin address

PUMP_PIN  EQU P1.0   ; Define pump control pin
SENSOR_PIN  EQU P1.1 ; Define water level sensor pin

MAIN:          ; Main program
  MOV P1, #0xFF    ; Set P1 as output
  MOV P2, #0x00    ; Set P2 as input

LOOP:
  CLR A           ; Clear accumulator
  MOV A, P2       ; Read the state of the water level sensor
```

```
    CJNE A, #0, WATER_LOW ; If water level is low, jump to WATER_LOW
    SJMP CONTINUE

WATER_LOW:
    SETB P1.0       ; Turn on the water pump
    ACALL DELAY     ; Delay for 5 seconds
    CLR P1.0        ; Turn off the water pump

CONTINUE:
    ACALL DELAY     ; Delay before checking water level again
    SJMP LOOP       ; Repeat the loop

DELAY:
    MOV R5, #100    ; Load R5 with 100
DELAY_LOOP:
    DJNZ R5, DELAY_LOOP ; Decrement R5, repeat until R5 is zero
    RET
END
```

CONCLUSION :

In conclusion, smart water fountains represent a promising fusion of technology and sustainability, offering a multitude of advantages for both users and the environment. These innovative fountains stand at the forefront of modernizing hydration experiences by harnessing advanced technologies to provide clean, safe, and efficient access to drinking water.the smart water fountain represents an innovative and efficient solution for promoting hydration and environmental sustainability. Its advanced features, such as automated refilling, water quality monitoring, and user-friendly mobile app control, make it a valuable addition to both public spaces and homes. By encouraging healthier hydration habits and reducing water wastage, these smart fountains contribute to a more sustainable and well-connected future.