

PROJECT:8 SMART WATER FOUNTAIN

DEVELOPMENT PART-1

PROJECT DEFINITION:

The project aims to enhance public water fountains by implementing IoT sensors to control water flow and detect malfunctions. The primary objective is to provide real-time information about water fountain status to residents through a public platform.

TEAM MEMBERS:

1.JERWIN.N.H-962221106056

2.AKSHINTH.M-962221106017

3.DEEPENDRAN.P-962221106039

4.AAKASH.S-962221106001

PROJECT OVERVIEW:

This project includes defining objectives, designing the IoT sensor system, developing the water fountain status platform, and integrating them using IoT technology and Python.

Designing a foundation for a Smart Water Foundation involves creating a comprehensive framework that leverages technology, data, and innovation to address water-related challenges, promote conservation, and ensure access to clean and safe water for all. Here's a high-level design plan for such a foundation:

- **Mission and Goals:** Clearly define the mission and goals of the Smart Water Foundation. This might include objectives like ensuring clean and accessible water, promoting water conservation, and using technology for water management.

- **Governance Structure:** Establish a robust governance structure with a board of directors, advisory board, and executive team responsible for overseeing the foundation's operations and decision-making.
- **Funding Strategy:** Develop a sustainable funding strategy, which may involve partnerships, grants, donations, and potentially government funding. Ensure transparency in financial management.
- **Technological Infrastructure:** Implement advanced technology solutions to monitor, manage, and optimize water resources. This could include:
 - **IoT Sensors:** Deploy sensors to monitor water quality, water levels, and usage in real-time.
 - **Data Analytics:** Utilize big data analytics and machine learning to derive insights from collected data.
 - **Smart Water Meters:** Promote the installation of smart water meters for efficient consumption tracking.
 - **GIS (Geographic Information Systems):** Use GIS to map water sources, distribution networks, and potential areas of concern.
- **Public Awareness and Education:** Launch awareness campaigns and educational programs to inform the public about the importance of water conservation and responsible usage.
- **Water Quality Monitoring:** Develop a system for continuous water quality monitoring to ensure safe and potable water supply. Implement alert mechanisms for water quality breaches.
- **Water Distribution Optimization:** Use data-driven insights to optimize the distribution of water resources, reducing waste and leakage in distribution networks.
- **Community Engagement:** Engage local communities and stakeholders in water management efforts. Encourage community-led initiatives for water conservation.
- **Research and Innovation:** Fund research projects and innovation in water technology, treatment, and management. Collaborate with universities and research institutions.
- **Emergency Response and Disaster Preparedness:** Develop contingency plans for water-related emergencies and natural disasters. Ensure quick response and recovery mechanisms.

- **Policy Advocacy:** Advocate for policies and regulations that support sustainable water management and conservation at local, national, and international levels.
- **Data Security and Privacy:** Implement robust data security and privacy measures to protect sensitive information collected through technology solutions.
- **Metrics and Reporting:** Establish key performance indicators (KPIs) to measure the foundation's impact and regularly report progress to stakeholders and the public.
- **Collaboration and Partnerships:** Collaborate with other non-profits, governmental agencies, corporations, and NGOs to leverage resources and expertise for achieving common water-related goals.
- **Scaling and Replication:** Plan for scalability and consider replicating successful programs in different regions or countries to maximize impact.
- **Monitoring and Evaluation:** Continuously monitor and evaluate the foundation's projects and initiatives to identify areas for improvement and adaptation.
- **Sustainability Initiatives:** Promote sustainable water practices in agriculture, industry, and households to reduce water consumption and environmental impact.
- **Public Engagement Platforms:** Develop web and mobile applications to engage the public in water monitoring, reporting issues, and receiving information on water conservation.

Remember that the success of the Smart Water Foundation relies on strong leadership, collaboration, technology implementation, and ongoing commitment to its mission of sustainable water management. Additionally, adapting to changing circumstances and emerging technologies is essential for long-term success.

MATERIALS AND COMPONENTS:

- **Water Pump:** Choose a water pump that suits the size and flow rate you want for your fountain.
- **Container:** Select a container to hold the water, like a basin or pot.
- **Tubing:** You'll need tubing to transport water from the pump to the fountain head.
- **Fountain Head:** Select a fountain head or nozzle to create the desired water spray pattern.
- **Power Supply:** Ensure you have a suitable power supply for the water pump (usually a low-voltage power adapter).
- **Microcontroller:** Choose a microcontroller (e.g., Arduino, Raspberry Pi) to control the fountain.
- **Sensors:** Depending on your project, you may need various sensors, such as motion sensors, ultrasonic sensors, or microphones.
- **Relays or Transistors:** Use these components to control the water pump.
- **LEDs:** For added visual effects, consider using LEDs to illuminate the fountain.
- **Water Reservoir:** Have a separate water reservoir to maintain the water level.
- **Water Filter:** A filter can help keep the water clean and prevent clogs in the tubing.
- **Smart Module :** If you want to control your fountain remotely, consider adding a Wi-Fi or Bluetooth module.

PROCEDURE:

- **Design your fountain:**

Decide on the size, shape, and appearance of your water fountain. This will help you select the appropriate components.
- **Assemble the hardware:**

Set up the container as the water basin.

Connect the water pump to the tubing and place it in the water basin.

Attach the fountain head at the top of the tubing.

Connect the power supply to the water pump.

Set up any additional components like LEDs for lighting and a water filter.

➤ **Set up the control system:**

Choose a suitable microcontroller and set it up with the required software development environment.

Connect sensors (e.g., motion sensor, microphone) to the microcontroller if you want to create an interactive fountain.

➤ **Write the code:**

Write the code to control the water pump, sensors, and any other components.

Implement the logic for how the fountain responds to different inputs.

➤ **Test the system:**

Test your smart water fountain to make sure it functions as expected.

Check for any issues or bugs in your code and make necessary adjustments.

➤ **Optional remote control (smart feature):**

If you want to control your fountain remotely, integrate a Wi-Fi or Bluetooth module into your microcontroller and develop a smartphone app or a web interface for control.

➤ **Finalize the project:**

Secure all components in place.

Fill the water basin with clean water.

Power up the system and monitor it for a while to ensure it operates reliably.

➤ **Maintenance:**

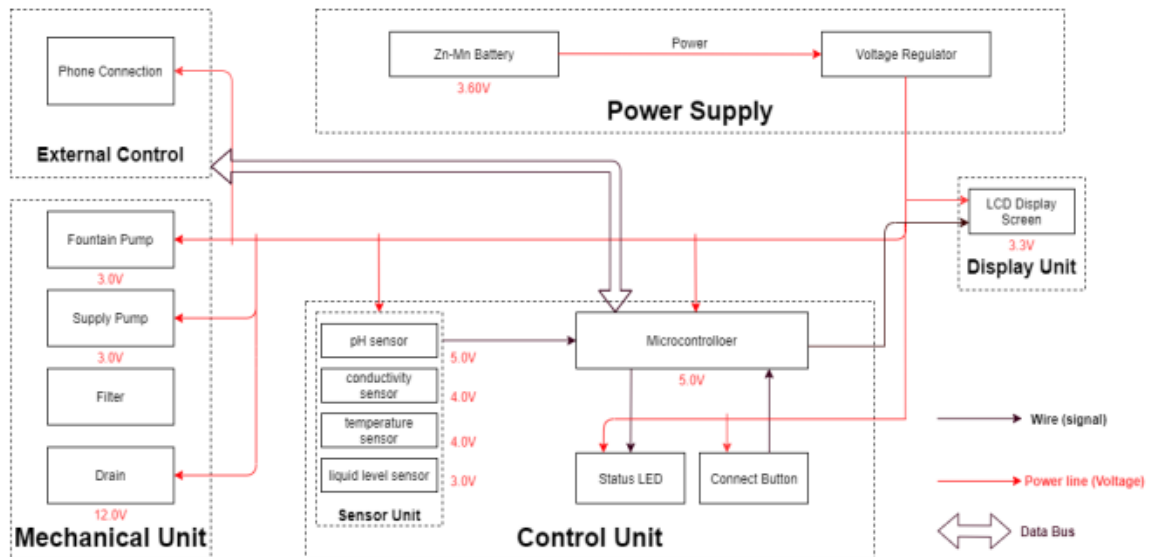
Regularly check and maintain your smart water fountain to keep it in good working condition. Clean the water reservoir, replace the water filter, and check for any clogs or malfunctions.

➤ **Enjoy your smart water fountain:**

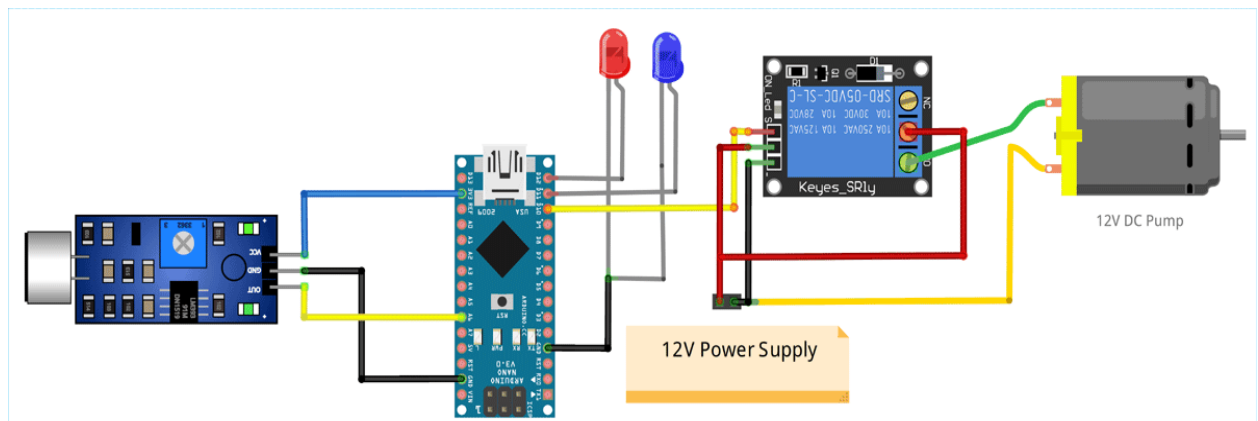
Once your project is complete, you can enjoy the soothing sight and sound of your custom-designed smart water fountain.

Remember to prioritize safety during the project, especially when working with water and electricity. Always follow safety guidelines for your specific components and electrical connections.

BLOCK DIAGRAM:



CIRCUIT DIAGRAM:



CODING:

```
// Include the necessary libraries
#include <Servo.h>

// Define the pins for the PIR sensor and water pump
const int pirPin = 2; // PIR sensor connected to digital pin 2
const int pumpPin = 8; // Water pump connected to digital pin 8
```

```
// Create a Servo object for controlling the water pump
Servo waterPump;
// Variables to store sensor state
int pirState = LOW;
int lastPirState = LOW;
void setup() {
  // Initialize the PIR sensor pin as an input
  pinMode(pirPin, INPUT);
  // Initialize the water pump as a servo
  waterPump.attach(pumpPin);
  // Turn off the water pump initially
  waterPump.write(0);
  // Serial communication for debugging (optional)
  Serial.begin(9600);
}
void loop() {
  // Read the PIR sensor
  pirState = digitalRead(pirPin);
  // Check if motion is detected
  if (pirState == HIGH) {
    // Turn on the water pump
    waterPump.write(90); // Adjust the angle to control the water flow
    delay(5000); // Run the pump for 5 seconds (adjust as needed)
    waterPump.write(0); // Turn off the water pump
    delay(1000); // Delay before rechecking for motion
  }
  // Save the current state for the next iteration
  lastPirState = pirState;
}
```

In this code:

- We use the Servo library to control the water pump, and a Servo object is created for this purpose.

- The PIR sensor is connected to digital pin 2, and the water pump is connected to digital pin 8.
- The program continuously checks the PIR sensor for motion. If motion is detected, it turns on the water pump for a specified duration (5 seconds in this example) and then turns it off.
- Adjust the angles and timing (e.g., the angles for controlling water flow and the delay times) to suit your specific water fountain setup.
- The program also includes serial communication for debugging. You can view sensor values and debug messages using the Arduino IDE's Serial Monitor.
- This is a basic example, and you can expand and customize the code to include features like LED lighting control, sound effects, remote control, and more, depending on the complexity of your smart water fountain project.

REQUIRED COMPONENTS:

- Raspberry Pi (or another microcontroller)
- Flow rate sensors (e.g., Hall-effect sensors)
- Pressure sensors (if necessary)
- Internet connection (Wi-Fi or Ethernet)
- Power supply for the Raspberry Pi
- Platform for receiving sensor data (e.g., a cloud-based platform or local server)
- [Step 1](#): Set up your Raspberry Pi

Assemble your Raspberry Pi and make sure it's connected to the internet, either via Wi-Fi or Ethernet.

Install Raspbian or another suitable operating system on your Raspberry Pi.

Install necessary Python libraries for sensor interfacing. For example, if you're using a Hall-effect flow rate sensor, you might need to install RPi.GPIO or any other relevant library for that sensor.

- [Step 2](#): Connect Sensors

Connect the flow rate sensors and pressure sensors (if required) to the GPIO pins of your Raspberry Pi. Make sure to follow the sensor's datasheet and pinout diagrams.

Ensure that the power supply for the sensors is appropriately connected.

➤ **Step 3:** Write Python Script for Sensor Data Collection

Write a Python script to collect data from the sensors and send it to a central platform. Here's a simplified example using Python:

PYTHON CODE:

```
import RPi.GPIO as GPIO
import time
import requests

# Define GPIO pins for your sensors
flow_sensor_pin = 17
pressure_sensor_pin = 18

# Initialize GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(flow_sensor_pin, GPIO.IN)
GPIO.setup(pressure_sensor_pin, GPIO.IN)

# URL of your platform to send data
platform_url = "https://your-iot-platform.com/api/data"

while True:
    # Read data from sensors
    flow_rate = GPIO.input(flow_sensor_pin)
    pressure = GPIO.input(pressure_sensor_pin)

    # Create a data payload
    data = {
        "flow_rate": flow_rate,
        "pressure": pressure,
        "timestamp": time.time()
    }

    # Send data to the platform
    response = requests.post(platform_url, json=data)

    # You can add error handling and other functionality here
    time.sleep(60) # Adjust the time interval for data collection
```

This script reads data from the flow rate and pressure sensors and sends it to your IoT platform at regular intervals.

➤ **Step 4: Set up the IoT Platform**

You need to set up an IoT platform to receive and store the data. Popular choices include AWS IoT, Google Cloud IoT, or platforms like ThingSpeak or Adafruit IO.

Create an account on your chosen IoT platform.

Set up the necessary channels or topics to receive and process the sensor data.

Obtain the platform's API endpoint and credentials and update them in your Python script.

➤ **Step 5: Run and Monitor**

Run your Python script on the Raspberry Pi. It will continuously collect and send sensor data to your IoT platform.

Monitor the platform for real-time water fountain status data. You can create visualizations, set up alerts, and perform analytics based on the received data.

Remember to ensure security practices, like encryption and authentication, when sending data to the platform, as this is crucial for IoT applications.

CONCLUSION:

In conclusion, a smart water fountain is a fascinating and interactive project that combines art, technology, and creativity to create a unique water feature. This project allows you to design a water fountain with various automated and customizable features that enhance its functionality and aesthetic appeal. Here are some key takeaways:

- **Integration of Technology:** A smart water fountain incorporates technology, such as microcontrollers like Arduino or Raspberry Pi, sensors, LEDs, sound systems, and more. These components can be programmed to control various aspects of the fountain's operation.
- **Customization:** With a smart water fountain, you can customize water flow patterns, colors, lighting effects, and even sound, allowing you to create a unique and ever-changing water display.
- **Interactivity:** Smart water fountains can be interactive. They can respond to user inputs, such as motion, sound, or even smartphone commands.

This makes them suitable for use in public spaces or as a captivating feature in your home.

- **Remote Control:** Some smart fountains can be controlled remotely through smartphone apps or other devices, giving you the ability to adjust settings and patterns from a distance.
- **Maintenance:** Proper maintenance is essential to keep the fountain in good working condition. Regularly check and clean the water reservoir, replace filters, and ensure electrical components are secure and waterproof.
- **Safety:** Safety is a primary concern when working with water, electricity, and electronic components. Ensure that all connections are secure and that there is no risk of electrical or water-related hazards.
- **Creative Expression:** Building a smart water fountain is an opportunity for creative expression. You can design the fountain to suit your personal aesthetic and functional preferences, making it a unique piece of art.
- **Educational Experience:** Creating a smart water fountain can be an educational endeavor, allowing you to learn about electronics, programming, and various components and sensors used in the project.

In summary, a smart water fountain is a captivating and engaging project that combines the beauty of water with the power of technology. Whether used as a decorative piece in your home or as a striking feature in a public space, a well-designed smart water fountain can provide a soothing and interactive experience for those who interact with it.