**SMART WATER FOUNTAIN :**

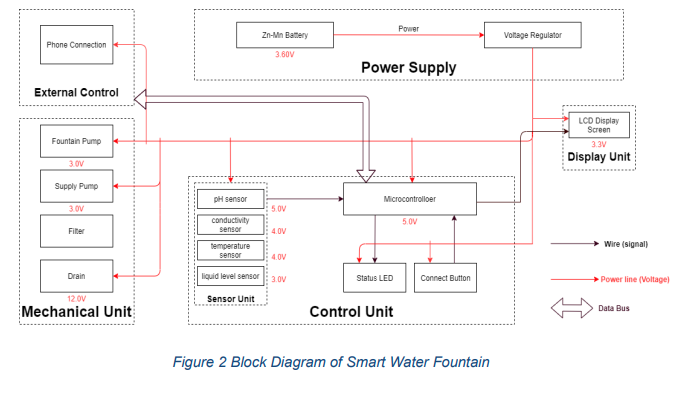
**Materials and Components:**

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

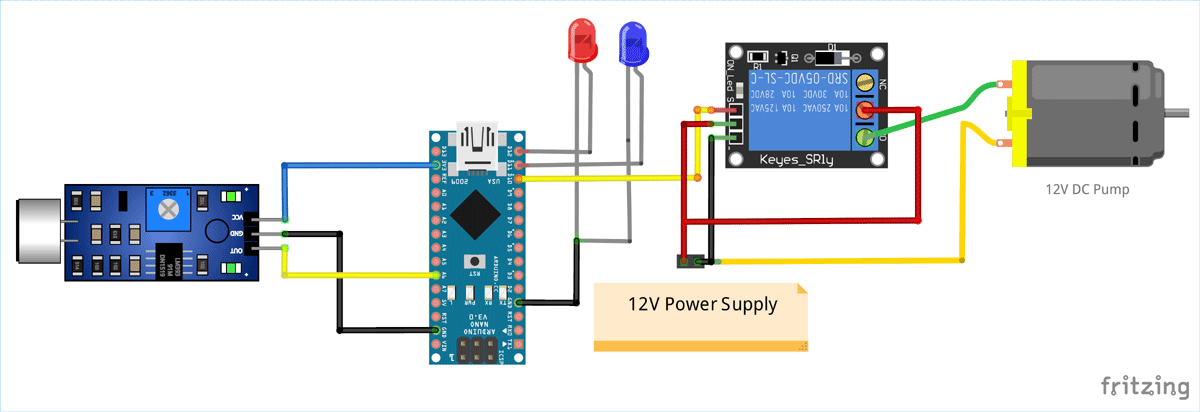
**Procedure:**

1. **Design your fountain:** Decide on the size, shape, and appearance of your water fountain. This will help you select the appropriate components.
2. **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.
3. **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.
4. **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.
5. **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.
6. **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.
7. **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.
8. **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.
9. **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.

**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:

1. **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.
2. **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.
3. **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.
4. **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.
5. **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.
6. **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.
7. **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.
8. **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.

**Top of Form**