

# Project Title: Smart water fountain

## Phase 4 : development part 2

In this part you will continue building your project. Continue building the project by developing the water fountain status platform.

Use web development technologies (e.g., HTML,CSS, JavaScript) to create a platform that displays real-time water fountain status.

Design the platform to receive and display real-time water fountain data, including water flow rate and malfunction alerts.

# To create a javascript for lot sensors that send real-time smart Water Fountains :

- **Hardware Components:**

- Microcontroller (e.g., Arduino, Raspberry Pi)
- Water level sensor
- Motion sensor (PIR sensor)
- Water pump
- Water container or reservoir
- Tubing or pipes for water flow
- Power supply

# Software Development:

- **Set Up the Microcontroller:**
- Connect the microcontroller to your computer.
- Install the necessary development environment for the microcontroller (e.g., Arduino IDE for Arduino or Python libraries for Raspberry Pi).

## Program the Microcontroller:

Write the firmware (code) for the microcontroller to control the water pump and sensors.

Use java for Raspberry Pi or C/C++ for Arduino.

Read data from the water level sensor to monitor the water level in the container.

Use the motion sensor to detect motion near the fountain.

- **Control the Water Pump:**

- Write code to control the water pump based on the water level and motion sensor data.
- If the water level is low and motion is detected, turn on the pump to start the fountain.

- **Feedback Mechanism:**

- Implement feedback mechanisms to ensure the water pump doesn't run dry, which could damage the pump.
- Monitor the water level continuously and ensure the pump is turned off when the water level is too low.

- **Web Interface or Mobile App (Optional):**

- If you want remote control and monitoring, you can create a web interface or a mobile app.
- You'll need to establish communication between the microcontroller and the interface, possibly via Wi-Fi or Bluetooth.

- **Testing and Debugging:**
  - Test your setup thoroughly.
  - Debug any issues in hardware or software.
- **Assembly and Waterproofing:**
  - Assemble the hardware components and ensure that all electrical connections are waterproofed to prevent damage from water splashes.
- **Power Supply:**
  - Ensure you have a stable power supply to run the microcontroller and the water pump.
- **Documentation and User Instructions:**
  - Document your project, including the wiring diagram, code, and any setup instructions for end users.
- **Maintenance and Safety:**
  - Regularly check and maintain the system.
  - Implement safety measures, such as emergency shutoffs in case of malfunctions.

## JavaScript Program for Connecting mobile app with Smart water fountains IOT project:

- `Const five = require('johnny-five');`
- `const board = new five.Board();`
- 
- `board.on('ready', function() {`
- `// Define pins for sensors and water pump`
- `const waterLevelPin = new five.Pin(2); // Replace with your actual pin number`
- `const motionSensorPin = new five.Motion(3); // Replace with your actual pin number`
- `const waterPump = new five.Relay(4); // Replace with your actual pin number`
- 
- `// Initialize motion sensor`
- `motionSensorPin.on('calibrated', function() {`
- `console.log('Motion sensor calibrated');`
- `});`

- `// Event handler for motion detection`
- `motionSensorPin.on('motionstart', function() {`
- `console.log('Motion detected. Fountain is active.');`
- `waterPump.on();`
- `});`
- 
- `motionSensorPin.on('motionend', function() {`
- `console.log('No motion detected. Fountain is inactive.');`
- `waterPump.off();`
- `});`
- 
- `// Read water level`
- `waterLevelPin.query(function(state) {`
- `if (state === 0) {`
- `console.log('Water level is low. Turning on the water pump.');`
- `waterPump.on();`
- `} else {`
- `console.log('Water level is sufficient. Turning off the water pump.');`
- `waterPump.off();`
- `}`
- `});`
- `});`

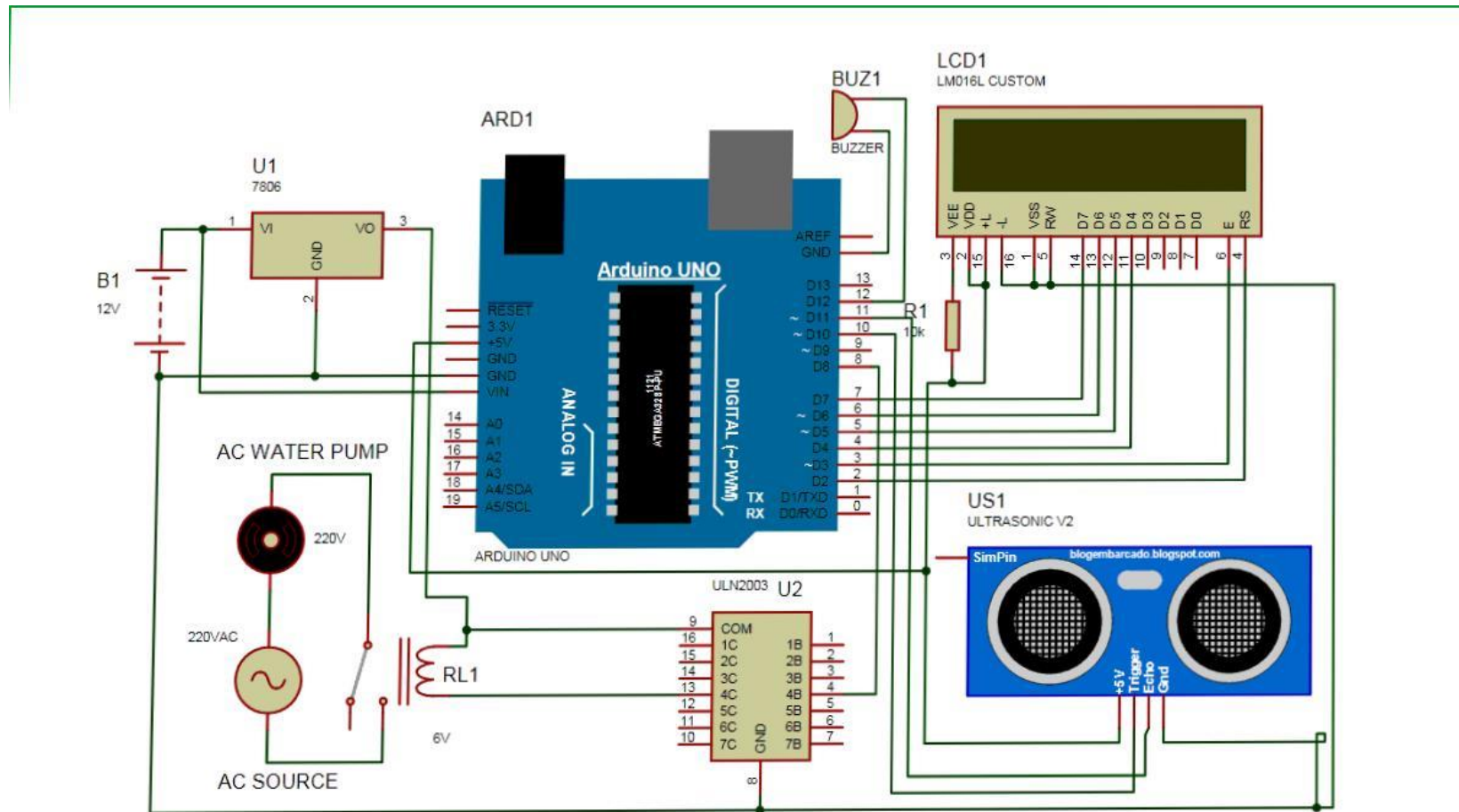
- **Front-end Development:**

- 
- **a. HTML:** Create the HTML structure for your web application. Define the layout, including elements
  - 
  - for displaying data.
  -
- **b. CSS:** Style your web application using CSS. Use CSS to make it visually appealing and user-friendly.
  -
- **c. JavaScript:** Use JavaScript to add interactivity and real-time updates to your platform. You can use
  - 
  - JavaScript libraries and frameworks to simplify this process, such as React, Angular, or Vue.js.



- In this Arduino program:
- Define the pins for the water level sensor, motion sensor, and water pump.
- In the setup function, set the pinMode for each pin and ensure the water pump is initially turned off.
- In the loop function, continuously read the status of the water level sensor and motion sensor.
- If the water level is low, turn on the water pump; otherwise, turn it off.
- Print messages to the serial monitor to indicate the status of the water level and motion.
- Adjust the delay to control how often the sensors are checked.

# Circuit diagram for Smart water fountains:



## Interfacing of the water level sensor with Arduino :

