IBM NAN MUDHALVAN –INTERNET OF THINGS
(GROUP-4)

SMART WATER FOUNTAIN

PHASE-2 INNOVATION

DONE BY:

➢ ARUN KUMAR.R	(au82042106004)
➢ DHINESH.R	(au82042106012)
➢ HARIHARASUDHAN.S	(au82042106014)
➢ MILTON EMIRATE RAJ.A	(au82042106029)
➢ MUKESH.M	(au82042106030)

DESIGN INTO INNOVATION

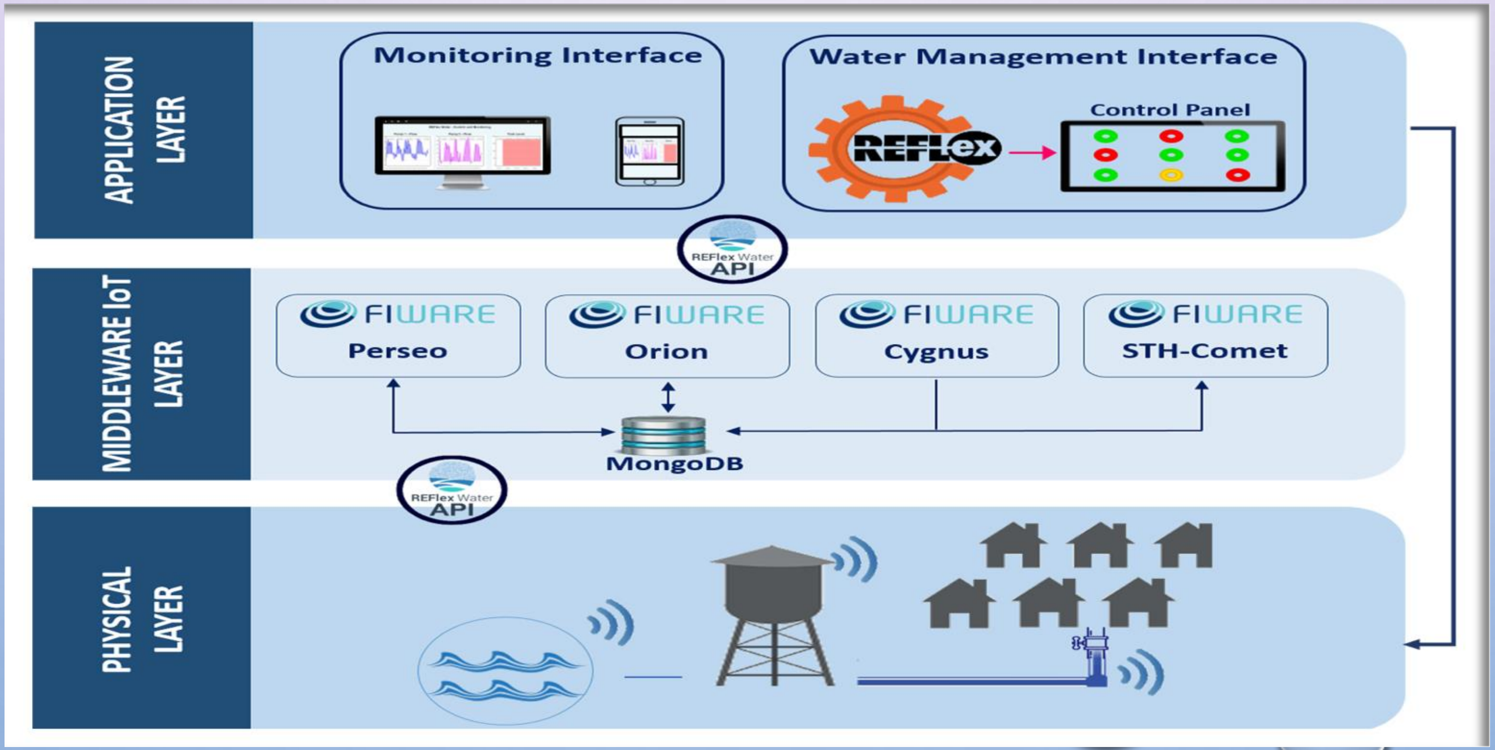
Converting a traditional water fountain into a smart innovation represents a practical approach to address modern challenges related to water conservation, hygiene, and efficiency. This transformation involves a strategic integration of sensors, automation, and connectivity, turning a mundane fixture into a sophisticated solution. This abstract provides an overview of the process, emphasizing the importance of careful planning, selection of components, and user-friendly features. The resulting smart water fountain not only promotes responsible water usage but also enhances user experience through touchless operation, water quality monitoring, and remote control capabilities. Furthermore, the smart fountain showcases the potential for the convergence of technology and sustainability, offering an exemplary case of how traditional infrastructure can adapt to the demands of a rapidly evolving world.

Converting a traditional water fountain into a smart innovation involves integrating technology and automation to enhance its functionality and usefulness. Here are the steps to turn a traditional water fountain into a smart innovation:

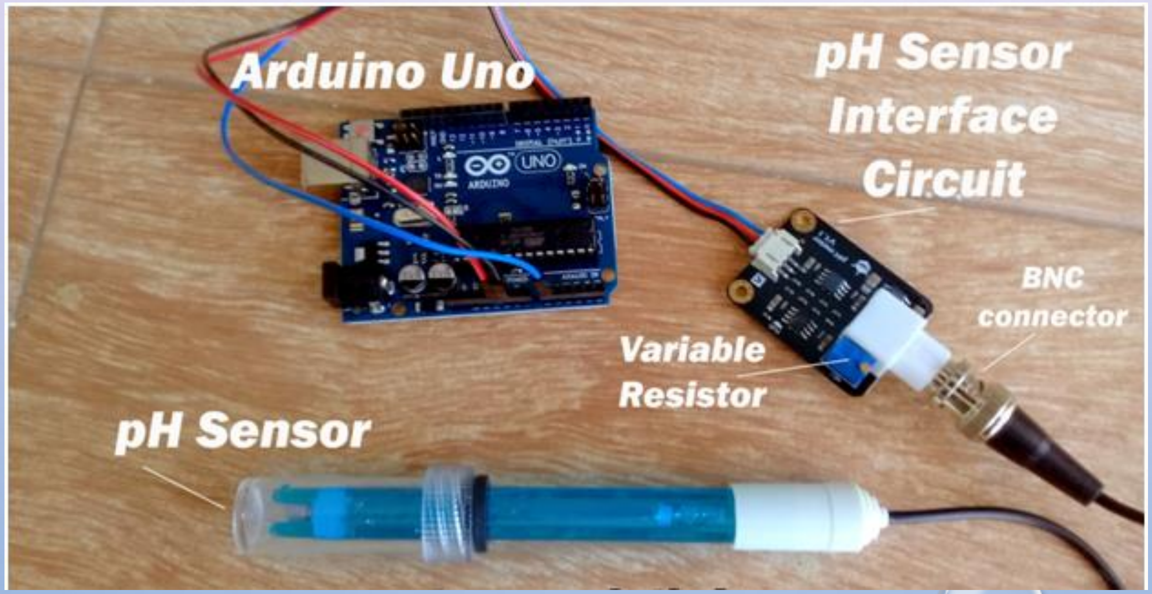
- **Water Quality Monitoring**
- **Touchless Operation**
- **Real-time Usage Tracking**
- **Water Filtration**
- **Solar-Powered**
- **Refill Stations**
- **Educational Displays**
- **Leak Detection**

Water Quality Monitoring

Water quality monitoring is a crucial aspect of smart water management systems. It involves the continuous assessment of various water parameters to ensure the safety and sustainability of water resources. Smart water management systems use technology and data analysis to optimize water distribution, reduce water waste, and respond to water quality issues in real time



This pH sensor kit is best for measuring the water quality and other parameters. It has an LED that works as the Power Indicator, a pH sensor, a BNC connector, a variable resistor that is used for the calibration which I will explain during the testing and PH2.0 sensor interface circuit.

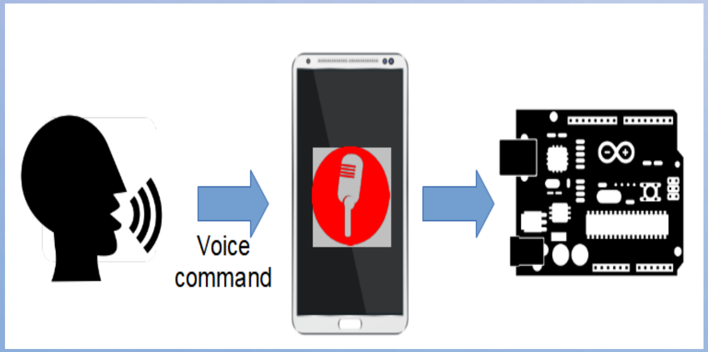


Touchless Operation

Implementing touchless operation in a smart water fountain is an excellent way to enhance user experience and promote hygiene, especially in light of concerns about the spread of germs and viruses. Here's how you can achieve touchless operation in a smart water fountain

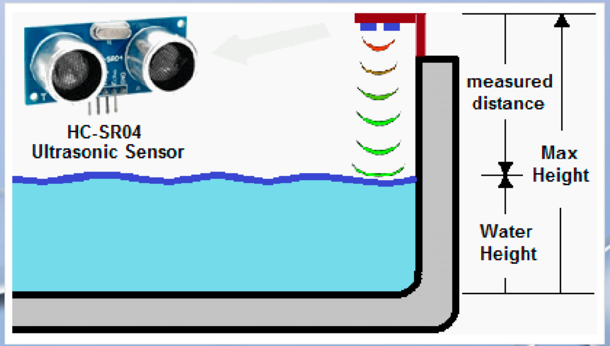
Voice Commands:

_Incorporate voice recognition technology to allow users to activate the water fountain with voice commands. For example, users can say "dispense water" to initiate the flow Use LED indicators to communicate the status of the water fountain. For example, a green LED could indicate that it's ready for u



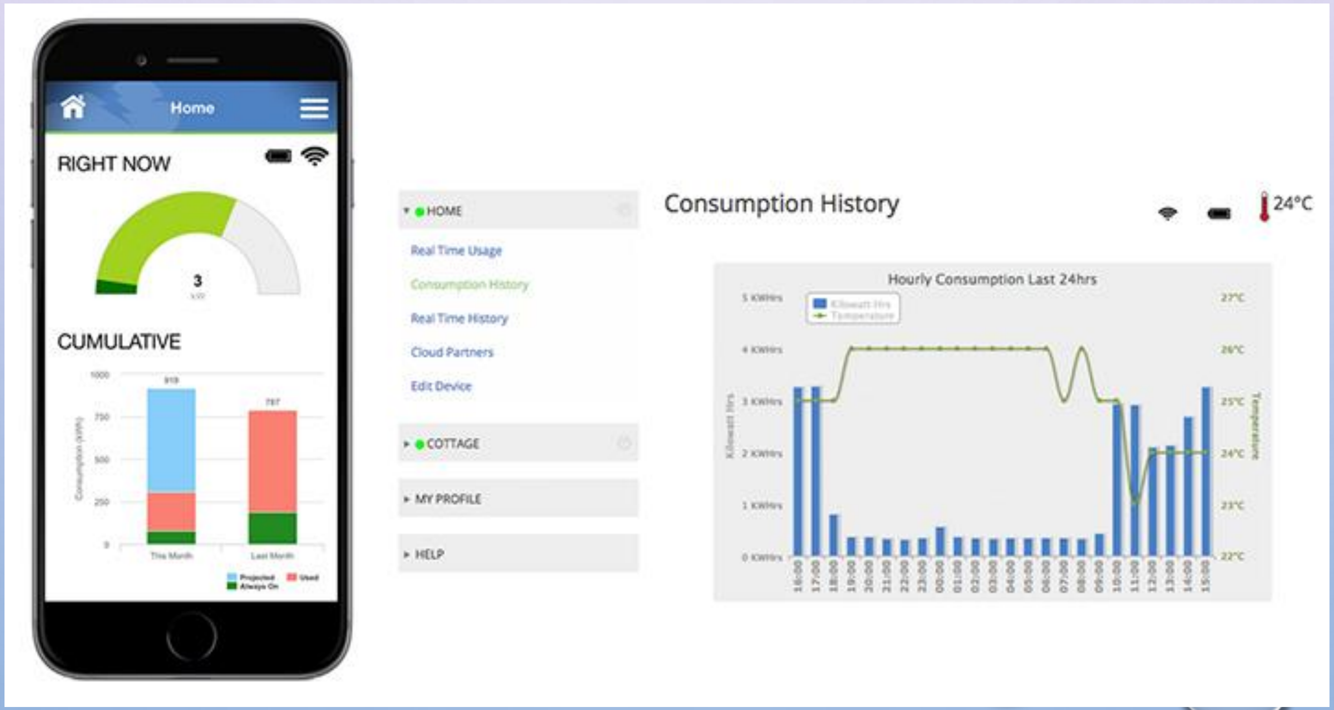
Ultrasonic Sensors:

Ultrasonic sensors can also be used to detect the presence of an object, such as a hand or bottle, in front of the fountain. These sensors work by emitting ultrasonic waves and measuring the time it takes for the waves to bounce back. When an object is detected, the water flow is triggered



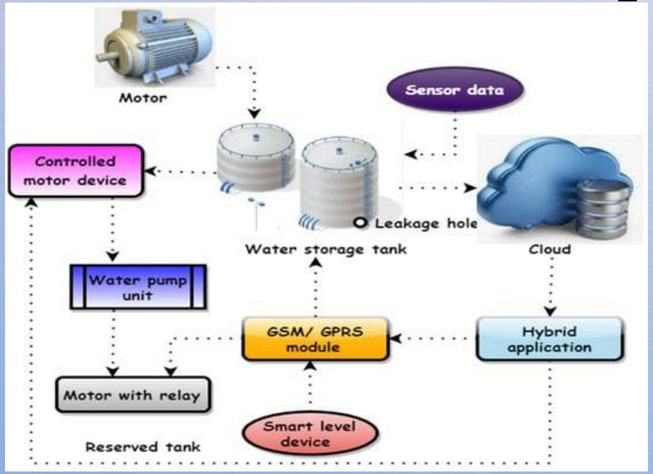
Real-time Usage Tracking

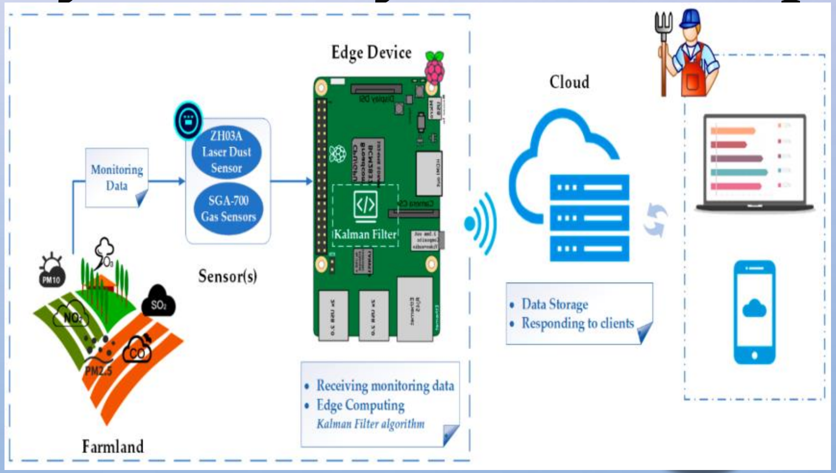
Real-time usage tracking in a smart water fountain is a valuable feature that helps monitor and optimize water consumption, promotes responsible usage, and contributes to sustainability efforts. Here's how you can implement real-time usage tracking in a smart water fountain



Water Filtration

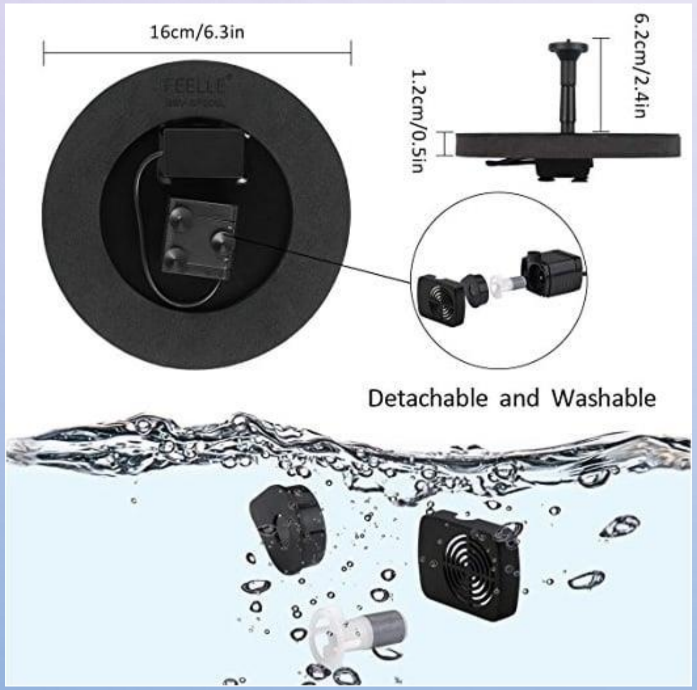
1.Predictive Maintenance: IoT-enabled water fountains can predict when the filters need replacement or maintenance based on usage patterns and water quality data. This reduces downtime and ensures optimal filtration performance.
2.Energy Efficiency: The IoT system can optimize the power consumption of the filtration system based on demand and water quality. For example, it can reduce power consumption during off-peak hours or when water quality is already within acceptable limits.

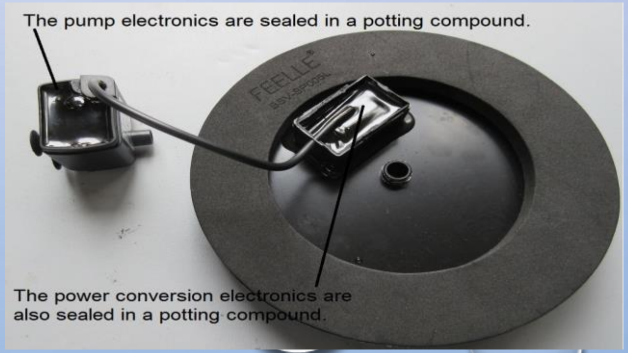




Solar-Powered

Feelle's Solar Powered Floating Water Fountain is a small, yet impressive, water fountain intended for a birdbath, pond, or pool and garden decoration. The diameter of the solar panel's disc measures about 6.3 inches; and while the thickness of the disc itself is about a ½ inch, the overall height of this little fountain is approximately 4 inches (from the bottom of the pump to the top of the fountain).




The pump electronics are sealed in a potting compound.
FEELLE
The power conversion electronics are also sealed in a potting compound.

Educational Displays

Incorporating educational displays into a smart water fountain can help raise awareness about water conservation and promote responsible water usage. Here's how to implement educational displays in a smart water fountain:

1. Digital Screens: Install digital screens or touchscreens near the water fountain. These screens can display educational content, interactive infographics, and videos about water conservation and its importance.
2. Content Rotation: Create a library of educational content that rotates periodically to keep users engaged and informed. Content can include tips on reducing water waste, the environmental impact of bottled water, facts about local water sources, and more.

RFID Attendance
OLED Display
WiFi Router
Computer Server
THE IOT PROJECTS

Leak Detection

1.Sensor Placement: Install water sensors at strategic locations where water leaks are likely to occur, such as near plumbing connections, water heaters, or appliances like washing machines and dishwashers.

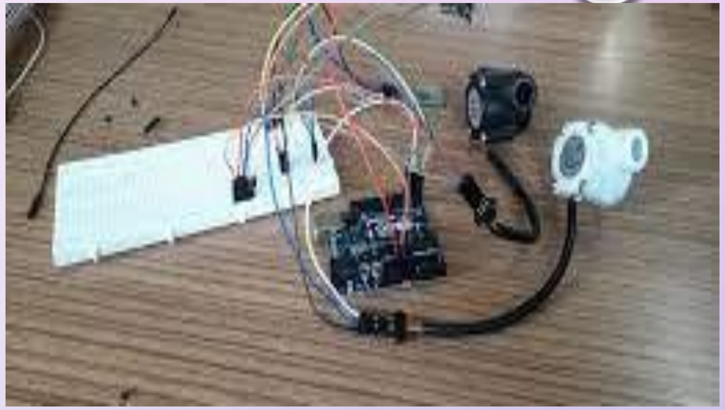
2.IoT Device Integration: Connect the water sensors to the IoT devices and configure them to continuously monitor for changes in moisture levels.

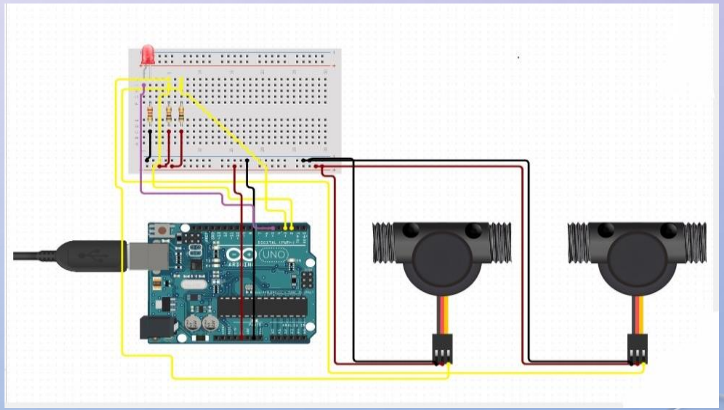
3.Data Transmission: Set up the IoT devices to send data to the central server or cloud platform at regular intervals. The data should include moisture levels, location information, and device status.

4.Real-time Monitoring: Implement real-time monitoring of the incoming data. You can use data analytics to detect anomalies, sudden spikes in moisture levels, or patterns indicative of a leak.

5.Alert System: Configure the system to send alerts when a potential leak is detected. Alerts can be in the form of SMS, email, mobile app notifications, or even automated phone calls.

6.Remote Control: If available, enable remote control of water valves or shutoff mechanisms through the IoT system to stop the water supply in the event of a confirmed leak.



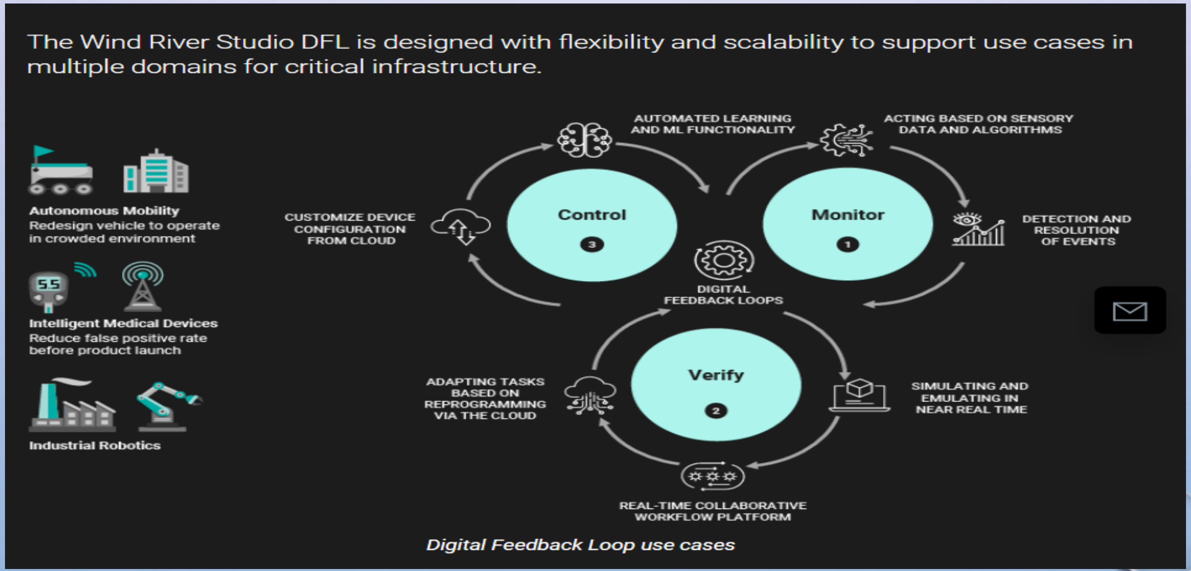


Feedback Mechanism

1.Determine Feedback Parameters: **Define what kind of feedback you want to collect. This could include user preferences, satisfaction ratings, environmental data, or any other relevant information.**

2.IoT Device Deployment: **Deploy IoT devices in the relevant locations or among users. These devices should be equipped with sensors and a user interface for feedback.**

3.Feedback Collection: **Encourage users to provide feedback through the IoT devices. They can do this by interacting with the user interface or allowing the sensors to collect data passively.**

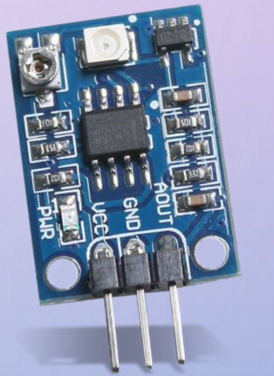


The Wind River Studio DFL is designed with flexibility and scalability to support use cases in multiple domains for critical infrastructure.

Autonomous Mobility
Redesign vehicle to operate in crowded environment
Intelligent Medical Devices
Reduce false positive rate before product launch
Industrial Robotics
AUTOMATED LEARNING AND ML FUNCTIONALITY
ACTING BASED ON SENSORY DATA AND ALGORITHMS
CUSTOMIZE DEVICE CONFIGURATION FROM CLOUD
Control
3
Monitor
1
DETECTION AND RESOLUTION OF EVENTS
DIGITAL FEEDBACK LOOPS
ADAPTING TASKS BASED ON REPROGRAMMING VIA THE CLOUD
Verify
2
SIMULATING AND EMULATING IN NEAR REAL TIME
REAL-TIME COLLABORATIVE WORKFLOW PLATFORM

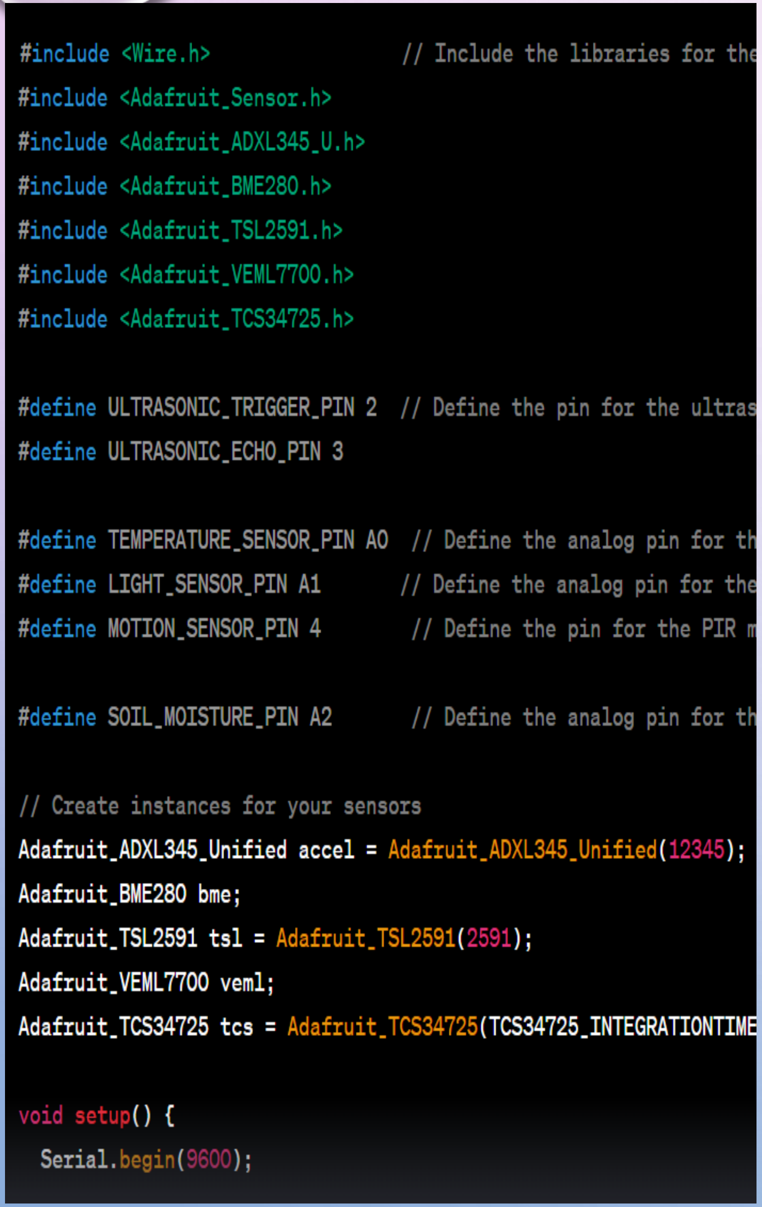
Digital Feedback Loop use cases

Creating an Arduino code for a water fountain using at least 5 sensors is a complex task that would require more information about the specific sensors and components you're using. However, I can provide you with a basic example code structure and explain how to integrate the sensors into the project. You can adapt this code to your specific sensor and hardware setup.
In this example, we'll use five different sensors:

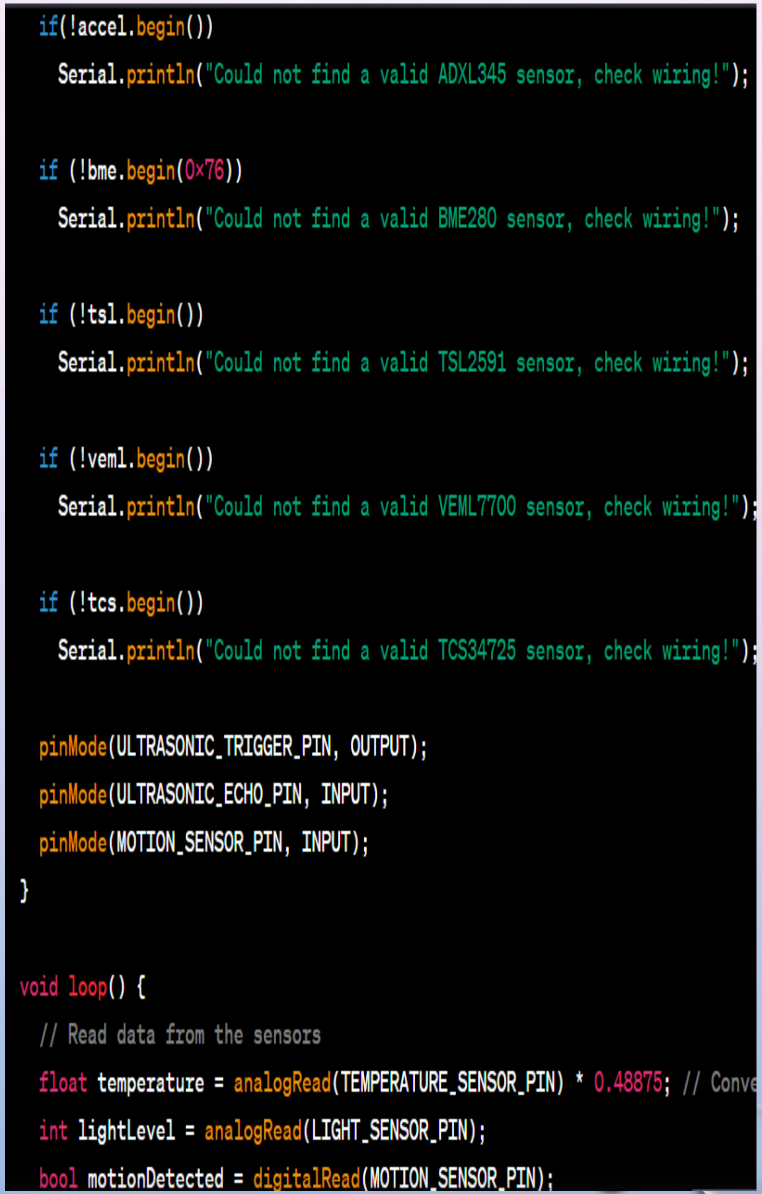


1. Ultrasonic sensor to detect water level.
2. Temperature sensor to monitor water temperature.
3. Light sensor to detect ambient light.
4. PIR motion sensor to detect motion.
5. Soil moisture sensor to check the soil moisture level.

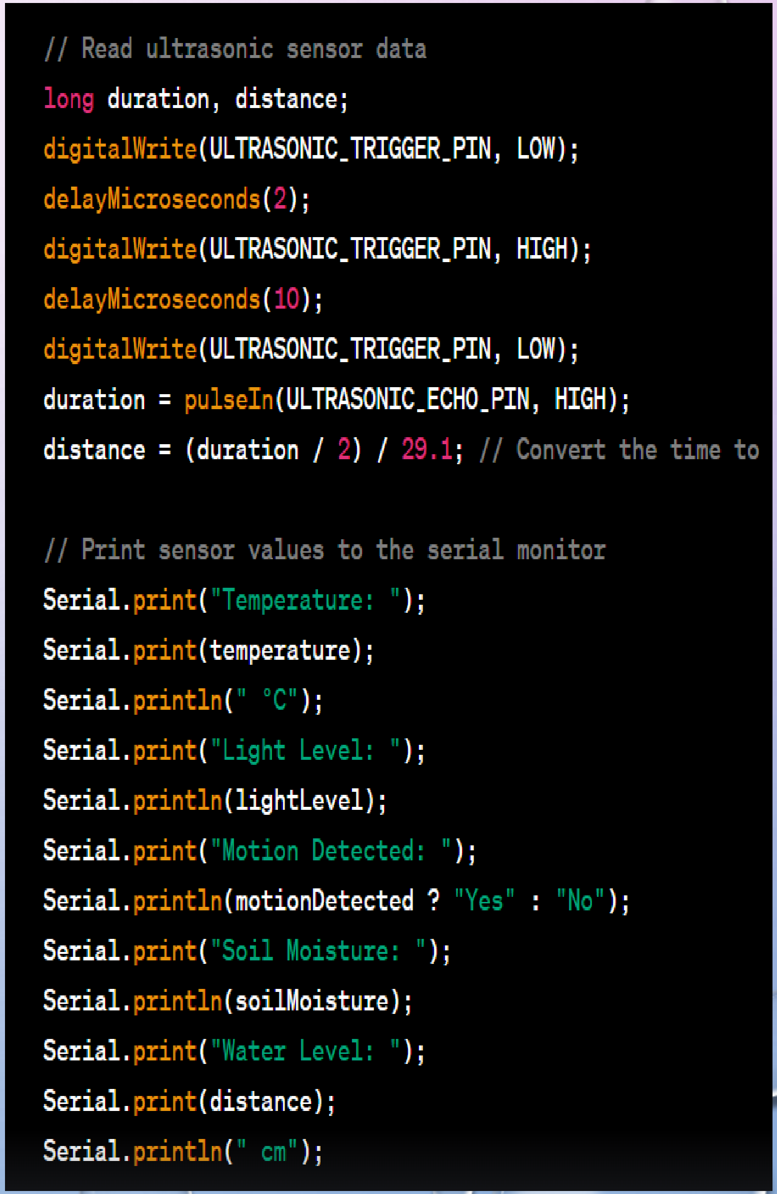
PROGRAM



```
#include <Wire.h>                // Include the libraries for the
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>
#include <Adafruit_BME280.h>
#include <Adafruit_TSL2591.h>
#include <Adafruit_VEML7700.h>
#include <Adafruit_TCS34725.h>

#define ULTRASONIC_TRIGGER_PIN 2  // Define the pin for the ultras
#define ULTRASONIC_ECHO_PIN 3

#define TEMPERATURE_SENSOR_PIN A0  // Define the analog pin for th
#define LIGHT_SENSOR_PIN A1       // Define the analog pin for the
#define MOTION_SENSOR_PIN 4        // Define the pin for the PIR m

#define SOIL_MOISTURE_PIN A2       // Define the analog pin for th

// Create instances for your sensors
Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(12345);
Adafruit_BME280 bme;
Adafruit_TSL2591 tsl = Adafruit_TSL2591(2591);
Adafruit_VEML7700 veml;
Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME

void setup() {
  Serial.begin(9600);
```



```
  if(!accel.begin())
    Serial.println("Could not find a valid ADXL345 sensor, check wiring!");

  if (!bme.begin(0x76))
    Serial.println("Could not find a valid BME280 sensor, check wiring!");

  if (!tsl.begin())
    Serial.println("Could not find a valid TSL2591 sensor, check wiring!");

  if (!veml.begin())
    Serial.println("Could not find a valid VEML7700 sensor, check wiring!");

  if (!tcs.begin())
    Serial.println("Could not find a valid TCS34725 sensor, check wiring!");

  pinMode(ULTRASONIC_TRIGGER_PIN, OUTPUT);
  pinMode(ULTRASONIC_ECHO_PIN, INPUT);
  pinMode(MOTION_SENSOR_PIN, INPUT);
}

void loop() {
  // Read data from the sensors
  float temperature = analogRead(TEMPERATURE_SENSOR_PIN) * 0.48875; // Conve
  int lightLevel = analogRead(LIGHT_SENSOR_PIN);
  bool motionDetected = digitalRead(MOTION_SENSOR_PIN);
```



```
// Read ultrasonic sensor data
long duration, distance;
digitalWrite(ULTRASONIC_TRIGGER_PIN, LOW);
delayMicroseconds(2);
digitalWrite(ULTRASONIC_TRIGGER_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(ULTRASONIC_TRIGGER_PIN, LOW);
duration = pulseIn(ULTRASONIC_ECHO_PIN, HIGH);
distance = (duration / 2) / 29.1; // Convert the time to

// Print sensor values to the serial monitor
Serial.print("Temperature: ");
Serial.print(temperature);
Serial.println(" °C");
Serial.print("Light Level: ");
Serial.println(lightLevel);
Serial.print("Motion Detected: ");
Serial.println(motionDetected ? "Yes" : "No");
Serial.print("Soil Moisture: ");
Serial.println(soilMoisture);
Serial.print("Water Level: ");
Serial.print(distance);
Serial.println(" cm");
```