IOT BASED SMART WATER FOUNTAINS…

Date : 29.10.2023

Smart water fountain in IoT (Internet of Things) aims to address various challenges related to water conservation, distribution, and quality monitoring. Some key objectives for implementing IoT in water fountain:

Project objectives:

- Efficient Water Usage: Optimize water consumption by employing IoT devices to monitor usage patterns, detect leaks, and manage irrigation systems based on real-time data. This can help in reducing water wastage and enhancing overall efficiency.

- Leak Detection and Prevention: Implement IoT sensors to detect leaks in water pipelines or distribution systems promptly. Real-time monitoring can enable quick responses to mitigate losses and prevent significant damage.

- Remote Monitoring of Water Quality: Employ IoT devices to continuously monitor water quality parameters such as pH levels, turbidity, and chemical composition. This ensures the early detection of contaminants and helps in maintaining safe and clean water supplies.

- Predictive Maintenance: Use IoT sensors to collect data on the condition of water infrastructure such as pumps, valves, and treatment systems. This enables predictive maintenance, reducing downtime and ensuring the proper functioning of equipment.

- Optimized Irrigation: Utilize IoT-enabled sensors to monitor soil moisture levels and weather conditions. This data can help in optimizing irrigation schedules, ensuring that water is used efficiently in agriculture and landscaping.

- Data-Driven Decision Making: Analyze the vast amount of data collected by IoT devices to gain insights into water usage patterns, demand fluctuations, and system performance. These insights can help in making informed decisions to improve overall water fountain strategies.

- Smart Water Distribution Networks: Implement IoT to create smart grids for water distribution. This involves real-time monitoring of demand, pressure, and flow rates within the network, allowing for dynamic adjustments and more effective fountain of water supply.

- Consumer Engagement and Awareness: Utilize IoT technologies to provide consumers with real-time information about their water usage. This can promote awareness and encourage responsible water consumption practices.

- Integration with Smart City Initiatives: Integrate smart water fountain systems with broader smart city initiatives to create a more connected, efficient, and sustainable urban environment.

- Regulatory Compliance and Reporting: Use IoT data to ensure compliance with water quality standards and regulations. Automated reporting systems can streamline the process of regulatory compliance.

These objectives collectively aim to leverage IoT technology to revolutionize water fountain by enhancing efficiency, ensuring sustainability, and improving the overall quality of water resources.

IOT Sensors:

The IOT sensor design plays a crucial role in the success of this project. It involves the deployment of various sensors and data collection devices throughout the public places. These sensors should include:

 Water pH Sensors:

pH sensors measure the acidity and alkalinity of water, which can affect water quality and aquatic ecosystems.

 Water Flow Sensors:

Flow sensors measure the rate of water flow in pipes or channels. They are used to monitor water consumption and detect leaks.

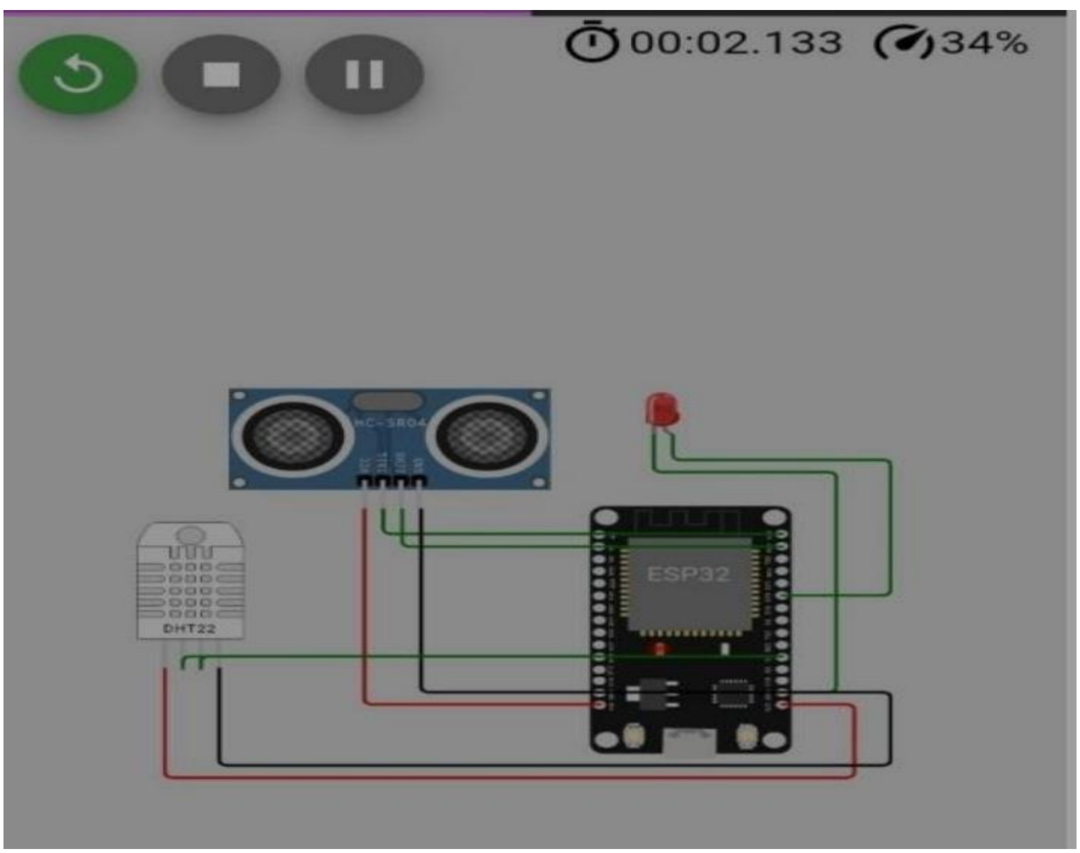
- Water Level Sensors:

Water level sensors measure the height or depth of water in tanks, reservoirs, or bodies of water. They are essential for managing water storage and flood monitoring.

- Water Quality Sensors:

These sensors measure various water quality parameters, including pH, turbidity, conductivity, dissolved oxygen, and contaminants such as chlorine or heavy metals.

Integration:



SOURCE CODE:

```
#include <DHT.h>

#include<WiFi.h>

//Dummy WiFi credentials(for simulation)

Const char* ssid=”your_network_name”;

Const char* password=”your_network_password”;

//DHT sensor configuration

#define DHTPIN 4 // Define the GPIO pin to which the DHT22 is connected

#define DHTTYPE DHT22 // Define the sensor type (DHT11 or DHT22)
DHT dht(DHTPIN,DHTTYPE);

Void setup() {

Serial.begin(115200);

//connect to wifi(simulated)

Serial.println(“connecting to WiFi(simulated)…”);

Delay(1000);
```

Serial.println(“connected(simulated) to WiFi”);

//Intialize the DHT sensor

Dht.begin();

}

Void loop() {

//read temperature and humidity

Delay(2000); // Delay between readings

Float temperature = dht.readTemperature(); // Read temperature in Celsius

Float humidity = dht.readHumidity(); // Read humidity

If (isnan(temperature) || isnan(humidity)) {

Serial.println(“Failed to read from DHT sensor!”);

} else {

Serial.print(“Temperature: “);

Serial.print(temperature);

Serial.println(“ °C”);

Serial.print("Humidity: ");

Serial.print(humidity);

Serial.println(" %");

}

Delay(2000);//delay for 2 seconds before the next reading(adjust as needed)

}

Real – time water consumption:

Code implementation:

HTML:(index.html)

```
<!DOC TYPE html>
<html>
<head>
<title>Smart Water Fountain Control</title>
</head>
<body>
 <h1>Control the Smart Water Fountain</h1>
 <button id=”startButton”>Start Fountain</button>
 <button id=”stopButton”>Stop Fountain</button>
 <script>
 // JavaScript code for controlling the fountain goes here

Document.getElementById(“startButton”).addEventListener(“cl
ick”, function() {


 // Code to start the fountain


 });
```

Document.getElementById(“stopButton”).addEventListener(“cli
ck”, function() {

// Code to stop the fountain

});

</script>

</body>

</html>

CSS :(styles.css)

Body{

Display: flex;

Justify-content: cenr;

Align-items: center;

Height: 100vh;

Margin: 0;

Background-color: #87CEEB; /* Set your desired background color */

}

.fountain {

Width: 200px;

Height: 200px;

Background-color: #1E90FF; /* Fountain base color */

```
 Border-radius: 50%;
 Position: relative;
 Animation: fountainAnim 1s infinite;
}
.water {
 Width: 60px;
 Height: 60px;
 Background-color: #00F; /* Water color */
 Border-radius: 50%;
 Position: absolute;
 Bottom: 0;
```

```
 Left: 70px;

 Animation: waterAnim 1s infinite;

}

@keyframes fountainAnim {

 0% { height: 200px; }

 50% { height: 100px; }

 100% { height: 200px; }

}

@keyframes waterAnim {

 0% { bottom: 0; }
```

50% { bottom: 100px; }

100% { bottom: 0; }

}

Const five = require('johnny-five');

Const board = new five.Board();

Board.on('ready', () => {

JavaScript: (script.js)

Const waterLevelSensor = new five.Sensor('A0');

Const pump = new five.Relay(10);

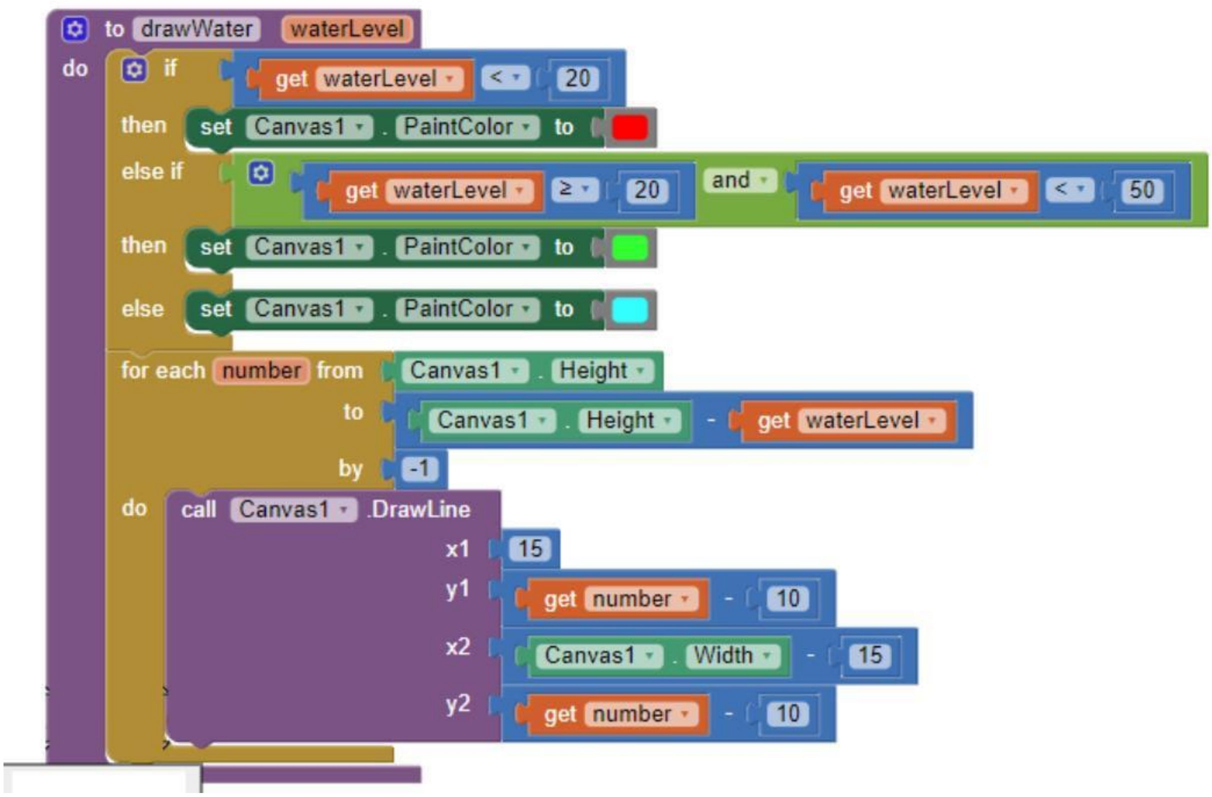
// Set a water level threshold

Const lowWaterLevelThreshold = 300;

```
waterLevelSensor.on('change', () => {

const waterLevel = waterLevelSensor.value;

if (waterLevel < lowWaterLevelThreshold) {

// Turn on the water pump

Pump.close();

} else {

// Turn off the water pump

Pump.open();

}

})
```

MIT APP INVENTOR :

App implementation:



Water conservation and sustainable practices:

1. Awareness and Behavioral Change:

Real-Time Feedback: Users receive immediate information about their water consumption. This direct_feedback can prompt behavioral changes as people become more conscious of their usage patterns. Seeing the impact of their actions in real-time encourages individuals to be more mindful of water wastage.

Usage Insights: Detailed data on water consumption patterns, trends, and peak usage times provided by the system can help users understand their habits better. Armed with this information, they can make informed decisions on how and when to use water more efficiently.

2. Early Leak Detection and Prevention:

Anomalies and Alerts: The system can detect anomalies, such as continuous flow when no water-usage is expected. Prompt alerts about potential leaks or abnormal consumption patterns allow quick response to rectify issues, reducing water loss.

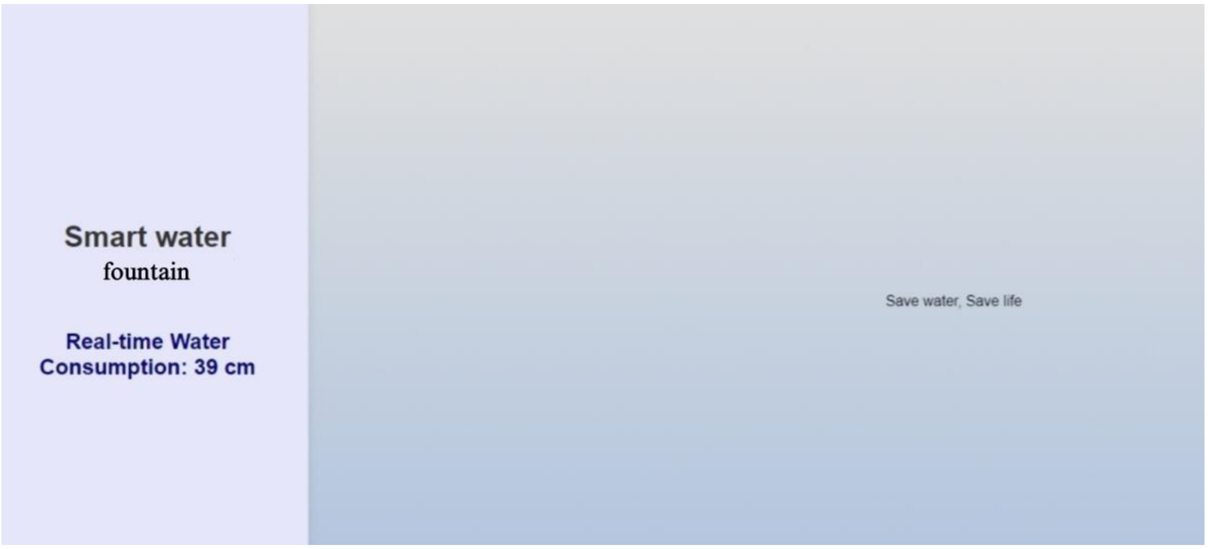
Preventative Maintenance: Identifying leaks early prevents significant water wastage and associated damage to infrastructure, supporting a sustainable water distribution system.

3. Efficient Resource Management:

Optimized Usage: With data insights, users can adjust their habits to reduce unnecessary consumption. For instance, they can identify peak usage times and try to limit water use during those periods, thereby reducing strain on the water supply system.

4. Community and Utility Awareness: Comprehensive data on usage patterns and demands help utility companies plan infrastructure upgrades or expansions .more accurately. They can allocate resources based on real-time needs, reducing unnecessary resource consumption. Infrastructure Planning: Comprehensive data on usage patterns.

Sample output:



Instructions to replicate the project:

1. IoT Sensor Deployment
2. Sensor Configuration
3. Code Development
4. Transit Information Platform Development:
5. Integration using python:
   Python to fetch data from the IoT platform (previously sent by sensors) through the platform's API or library.