

**Smart Irrigation System using IoT and Sensors**

**A PROJECT REPORT**

**Submitted by**

***K surjith babu(192212349)***

*Under the guidance of*

**Dr R DHANALAKSHMI**

*in partial fulfilment for the completion of course*

**CSA0279 – C PROGRAMMING FOR BRGINNERS**

**TITLE**:

**Smart Irrigation System using IoT and Sensors**

**PROBLEM STATEMENT:**

Efficient water management is essential in agriculture to maximize crop yield and minimize water wastage. Traditional irrigation methods often result in overwatering or underwatering, affecting crop productivity and increasing resource consumption. The lack of precision in these methods highlights the need for a smarter approach. A system that dynamically adjusts water supply based on real-time environmental data is critical. This project aims to address these challenges with a sustainable and automated irrigation solution using IoT and sensors.

**Tasks:**

* Design and implement IoT-based sensors to monitor soil moisture, temperature, and humidity.
* Develop an automated system to control irrigation based on sensor and weather data.
* Create a mobile or web application for remote access and control.
* Integrate weather data for predictive irrigation planning.
* Enable water usage tracking and provide analytics to optimize irrigation patterns.

**Outcome** :

The Smart Irrigation System will enhance agricultural productivity by ensuring efficient water management. Farmers can monitor soil and environmental conditions in real time through IoT sensors. The system will automate irrigation, reducing water wastage and improving crop health. Remote access through mobile or web applications will allow convenient control. Integration with weather data will prevent unnecessary watering during rain. Water usage tracking and analytics will help optimize irrigation schedules. Alerts will ensure timely manual intervention if needed. Ultimately, the system will promote sustainable farming practices and reduce costs for farmers.

**AIM**

The Smart Irrigation System using IoT and Sensors aims to revolutionize agricultural practices by optimizing water usage through advanced technology. The system intends to monitor soil moisture, temperature, and weather conditions in real-time to adjust irrigation schedules efficiently. By integrating IoT sensors, the system can provide precise water distribution, reducing wastage and promoting sustainable farming. It aims to enhance crop health by delivering the right amount of water at the right time, thus improving yield and minimizing resource consumption. The system will also incorporate weather forecasts to anticipate rainfall and adjust watering accordingly.

**ABSTRACT**

Efficient water management is a critical challenge in agriculture, where traditional irrigation methods often lead to overwatering or underwatering, reducing crop productivity and increasing resource wastage. This project focuses on developing a Smart Irrigation System using IoT and sensor technology to optimize water usage and improve agricultural outcomes. The system leverages IoT-based sensors to monitor real-time soil moisture, temperature, and humidity levels, combined with weather forecasts, to automate irrigation.

A mobile or web application provides remote access, allowing farmers to control and monitor the system conveniently. Additionally, the system tracks water usage and provides analytics for optimizing irrigation schedules. By integrating local weather data, it prevents unnecessary watering during rain and sends alerts for manual intervention when needed.

Scalable to different farm sizes and crop types, this solution addresses the need for sustainable agriculture. It enhances crop health, minimizes water wastage, reduces costs, and empowers farmers with data-driven insights, contributing to environmentally friendly farming practices.

**INTRODUCTION**

A Smart Irrigation System using IoT and Sensors is an advanced agricultural technology designed to optimize water usage and improve irrigation efficiency. It combines Internet of Things (IoT) technology with various sensors, such as soil moisture, temperature, humidity, and rain sensors, to automate and regulate the watering process.A Smart Irrigation System using IoT and Sensors is an advanced agricultural technology designed to optimize water usage and improve irrigation efficiency. It combines Internet of Things (IoT) technology with various sensors, such as soil moisture, temperature, humidity, and rain sensors, to automate and regulate the watering process.

Key functionalities of the system include:

* Sensors collect data about soil moisture, temperature, and environmental conditions.
* The system determines when and how much water is needed, preventing over-irrigation or under-irrigation.
* IoT integration allows users to monitor and control the irrigation system remotely through smartphones or computers.
* By delivering water precisely where and when it is needed, the system conserves water and reduces energy consumption.
* Historical data helps in understanding irrigation patterns and optimizing water usage for future cycles.

This system helps tackle global challenges like water scarcity and promotes sustainable agricultural practices. It is particularly beneficial in regions with limited water resources or irregular rainfall patterns.

**CODE IMPLEMENTATION**

**Modules:**

**Complaint Registration:**

1. Allows users (farmers, technicians, or system administrators) to register issues or malfunctions in the irrigation system, such as sensor failures or water supply interruptions.
2. Users can submit complaints through a mobile app or web interface, providing detailed descriptions and attaching images if necessary.

**Complaint Categorization:**

1. Categorizes registered complaints based on severity, type (sensor issue, water flow issue, power failure, etc.), and priority level.
2. The system uses automated classification algorithms or human intervention to categorize complaints, allowing for efficient handling.

**3. Assignment:**

a. Assigns the complaints to the appropriate team member or technician based on expertise, workload, and location.

b. The system automatically matches complaints to technicians with the right skills and availability, or manual assignment can be performed by an administrator.

**4. Status Tracking:**

a.Tracks the progress of complaint resolution, providing real-time updates to users and administrators.

b. Once a complaint is assigned, the system updates its status (e.g., "In Progress," "Resolved," "Pending") and sends notifications to relevant parties.

**5. Feedback Collection:**

a. Collects user feedback after the complaint is resolved to evaluate the quality of service and system performance.

b.A survey or feedback form is sent to the user post-resolution, where they can rate the service and provide comments for improvement.

**Data Storage**

* Stores real-time data from all sensors in a cloud-based or local database.
* Desired soil moisture levels,Seasonal irrigation schedules.

**Flow**

Soil, temperature, and humidity sensors gather environmental data.

Sensor data is sent to the control module via communication protocols.

**Code Outline:**

1. **Sensor Data Collection:**
2. **Decision-Making and Control:**

* Controlling irrigation systems
* Receiving alerts about critical events
* Water levels or maintenance needs

**PROGRAM**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int readSoilMoisture() {

return rand() % 101;

}

// Function to read temperature (Simulated)

int readTemperature() {

return rand() % 51;

}

int readHumidity() {

return 50 + rand() % 51;

}

void controlPump(int soilMoisture, int moistureThreshold) {

if (soilMoisture < moistureThreshold) {

printf("Pump Status: ON\n");

} else {

printf("Pump Status: OFF\n");

}

}

void displaySensorData(int soilMoisture, int temperature, int humidity) {

printf("Soil Moisture: %d%%\n", soilMoisture);

printf("Temperature: %d°C\n", temperature);

printf("Humidity: %d%%\n", humidity);

}

void checkWeatherPrediction() {

int weather = rand() % 3; // Randomized prediction

if (weather == 0) {

printf("Weather Prediction: Rain Expected\n");

} else if (weather == 1) {

printf("Weather Prediction: Cloudy\n");

} else {

printf("Weather Prediction: No Rain\n");

}

}

void saveDataToDatabase(int soilMoisture, int temperature, int humidity) {

printf("Data Logged: SoilMoisture=%d%%, Temperature=%d°C, Humidity=%d%%\n",

soilMoisture, temperature, humidity);

}

void analyzeHistoricalData() {

printf("Analyzing historical data...\n");

int averageMoisture = 40; // Example average from database

int averageWaterUsage = 100; // Example water usage in liters

printf("Average Soil Moisture: %d%%\n", averageMoisture);

printf("Average Water Usage: %d liters per day\n", averageWaterUsage);

}

void displayUserMenu() {

printf("1. View Sensor Data\n");

printf("2. Simulate Weather Update\n");

printf("3. Log Data\n");

printf("4. Analyze Historical Data\n");

printf("5. Configure Moisture Threshold\n");

printf("6. Exit\n");

}

int main() {

srand(time(0)); // Seed for random values

int choice, soilMoisture, temperature, humidity;

int moistureThreshold = 40; // Default moisture threshold

while (1) {

displayUserMenu();

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

soilMoisture = readSoilMoisture();

temperature = readTemperature();

humidity = readHumidity();

displaySensorData(soilMoisture, temperature, humidity);

controlPump(soilMoisture, moistureThreshold);

break;

case 2:

checkWeatherPrediction();

break;

case 3:

soilMoisture = readSoilMoisture();

temperature = readTemperature();

humidity = readHumidity();

saveDataToDatabase(soilMoisture, temperature, humidity);

break;

case 4:

analyzeHistoricalData();

break;

case 5:

printf("Current Moisture Threshold: %d%%\n", moistureThreshold);

printf("Enter new threshold: ");

scanf("%d", &moistureThreshold);

printf("Threshold updated to %d%%\n", moistureThreshold);

break;

case 6:

printf("Exiting the system.\n");

exit(0);

break;

default:

printf("Invalid choice. Try again.\n");

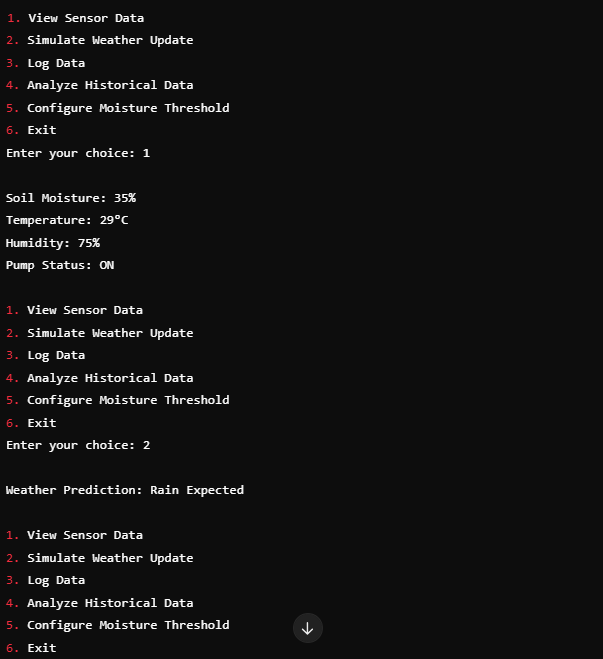
}

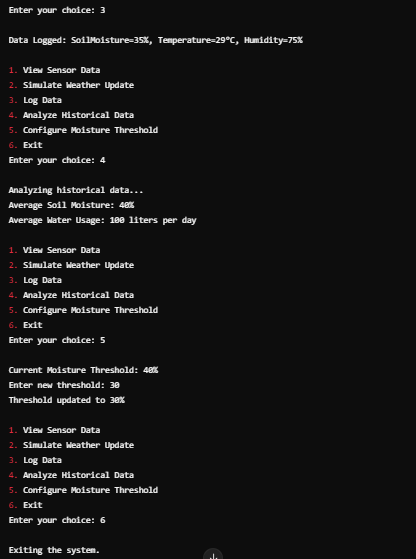
printf("\n");

}

return 0;

}

**RESULT**

****

**ENGINEERING STANDARDS**

1. C Language Standards
2. Software Development Standards
3. Data Security and Privacy Standards
4. Usability and Accessibility Standards
5. System Performance Standards
6. Interoperability and Integration Standards
7. Ethical and Social Responsibility Standards
8. Documentation and Reporting Standards
9. Scalability and Future Readiness

By adhering to these standards, the development process of the Smart Irrigation System using IoT and Sensors will be systematic, ensuring that the system operates efficiently, with minimal bugs and maintenance issues, and can be easily integrated with other agricultural technologies. The software will be reliable, ensuring accurate real-time data collection and irrigation adjustments. Furthermore, it will be secure, protecting user data and preventing unauthorized access to system controls. These standards will help deliver a product that aligns with best industry practices, ensuring efficiency, sustainability, and user satisfaction.

Key standards include, I.ISO/IEC 30141:2018 which outlines a structured approach to software development, ensuring every phase is well-organized and documented.IEEE 802.11 (Wi-Fi)focuses on testing documentation to validate the software’s performance and identify any issues before deployment. Additionally, and it emphasizes software quality, ensuring functionality, reliability, usability, and security.

By adhering to these standards, I aim to develop a system that not only meets its intended purpose of handling consumer complaints effectively but also upholds high-quality benchmarks, ensuring customer trust and satisfaction.

**C Language Standard**

**1.ISO/IEC 30141:2018**

IoT Reference Architecture

Device and Network Interoperability

Data Handling and Security:

Scalability and Flexibility

Device Management and Monitoring

Integration with Cloud and Big Data

2.**IEEE 802.15.4**

Cost-Effective

Mesh Networking Capability

Short Range

Low Data Rate

Reliable Communication

Low Power Consumption

**3.IEEE 802.11 (Wi-Fi)**

Wireless Communication

Real-time Data Transmission

Integration with Other Smart Systems

Power Efficiency

Scalability

Remote Monitoring and Control

4.**ISO/IEC 9899:2011 (C11)**

Long-term Performance and Maintenance

Protection from Dust and Water

Temperature and Humidity Testing

Vibration and Shock Resistance

Durability and Reliability

**5.ISO 80601-2-56**

Safety and Reliability

Data Integrity and Accuracy

Automation and Control

System Interoperability

Security and Data Privacy

Performance Standards

1. **Software Development Standards**

1. **ISO/IEC 12207:** Software outlines lifecycle processes

* providing a structured framework for developing the Smart Irrigation System. It covers everything from initial planning and requirements gathering to design, development, testing, and maintenance.
* The standard emphasizes proper documentation at every phase, which helps in tracking progress and ensuring that the system is built according to the initial specifications.

2.**IEEE 829:** Test Documentation Standard

* IEEE 829 ensures that comprehensive test plans, test cases, and test results are documented throughout the development cycle.
* By adhering to IEEE 829, developers can test the system in real-world conditions, such as varying soil moisture and weather conditions.

3.**ISO/IEC 25010:** Software Quality Models

* This standard is key in defining and maintaining high-quality software. It provides criteria for assessing the software's functionality.
* It emphasizes the need for secure data handling and a user-friendly interface. Security standards help protect sensitive data.

4.**IEEE 830:** Software Requirements Specification (SRS)

* IEEE 830 provides guidelines for creating a comprehensive Software Requirements Specification (SRS), which is crucial for defining all system functionalities.
* The SRS helps in documenting both functional requirements (e.g., automatic water adjustment based on soil moisture levels) and non-functional requirements.

5**.ISO/IEC 15504:** Software Process Assessment (SPICE)

* SPICE allows for benchmarking the development practices against global standards and best practices, fostering a culture of excellence.
* It ensures that the Smart Irrigation System undergoes regular reviews to assess the maturity of development processes, ultimately resulting in improved software quality.

**FUTURE SCOPE**

The Smart Irrigation System using IoT and sensors has immense potential for future advancements in agriculture. It can be enhanced by integrating advanced AI and machine learning algorithms to predict crop water requirements more accurately based on historical and real-time data. The system can also incorporate solar-powered energy solutions to make it more sustainable and energy-efficient. Expansion to include nutrient sensors could enable precise fertilizer application, improving soil health and crop productivity.

**1.Sustainability and Resource Management**

* A Smart Irrigation System aims to minimize the environmental footprint by efficiently utilizing water resources. By incorporating sensors that monitor soil moisture levels, temperature, and other environmental factors.
* The system can ensure that irrigation occurs only when necessary, preventing water wastage. Additionally, real-time data allows for efficient water usage, promoting sustainability in agricultural practices and reducing water consumption.

**2.Integration with Smart Farming Systems**

* The integration of a smart irrigation system with other smart farming technologies, such as climate control, automated machinery, and crop monitoring systems, enhances overall farm management.
* Data collected by the irrigation system can be shared with other components of the farming system, leading to optimized workflows, better decision-making, and a more efficient agricultural operation.

**3.Leak Detection and Maintenance Alerts**

* IoT-enabled smart irrigation systems can detect irregularities, such as leaks or malfunctioning equipment, in real-time.
* Sensors in pipes, valves, and sprinklers send continuous data to the central system, which uses analytics to identify anomalies.

**4.Predictive Analytics for Crop Health**

* Predictive analytics is a key feature that uses historical and real-time data (e.g., weather patterns, soil conditions) to forecast potential issues, such as pest infestations, diseases, or nutrient deficiencies.
* The system can then recommend or initiate preventative measures, helping farmers maintain healthy crops and optimize yield.

**5.Scalable System Design**

* The scalability of a Smart Irrigation System allows it to be adapted to different farm sizes and types, from small family farms to large industrial-scale operations.
* A scalable design ensures that the system can be customized to fit the specific needs of each user, making it a flexible solution for various agricultural settings.

**6.Nutrient and pH Monitoring**

* Smart irrigation systems can also integrate sensors that measure soil pH and nutrient levels.
* This data helps farmers determine whether the soil is acidic or alkaline and whether the crops are receiving sufficient nutrients.

**7.Energy-Efficient Operation**

* Smart irrigation systems are designed to be energy-efficient, reducing operational costs and minimizing environmental impact.
* The system can adjust watering times to avoid peak energy demand hours, making use of renewable energy sources when available.

**8.Mobile/Web Application Control**

* A mobile or web application allows farmers to remotely monitor and control the smart irrigation system.
* The application provides real-time data on soil moisture, weather forecasts, and irrigation status.

**9.Automated Watering System**

* The core of a smart irrigation system is its ability to automatically water crops based on real-time data from sensors.
* The system evaluates soil moisture levels and weather conditions to determine the ideal time and amount of water to deliver.

**10.Weather-Integrated Irrigation**

* Weather data is integrated into the smart irrigation system to adjust watering schedules based on real-time and forecasted weather conditions.
* By considering factors like temperature, humidity, and precipitation, the system adapts to changing weather patterns, improving efficiency and promoting water conservation.

**11.Real-Time Soil Monitoring**

* Sensors placed in the soil continuously measure parameters such as moisture levels, temperature, and electrical conductivity.
* The real-time data collected from these sensors helps the system make instantaneous irrigation decisions.

**CONCLUSION**

The Smart Irrigation System using IoT and Sensors developed using C programming serves as a foundational application for managing customer grievances in an organized and efficient manner. By enabling the controlling irrigation systems, receiving alerts about critical events, water levels or maintenance needs the system addresses key challenges in such as delayed responses and lack of transparency.

This project highlights the practical use of C programming for solving real-world problems, leveraging its file handling and structured programming capabilities. While the current system provides basic functionalities, it demonstrates the importance of automation in reducing manual effort, improving operational efficiency, and enhancing customer satisfaction.

The Smart Irrigation System using IoT and Sensors presents a transformative solution to modern agricultural challenges. By leveraging real-time data from soil moisture, temperature, and weather sensors, the system ensures efficient water usage, reducing wastage and promoting sustainable practices. Automation, coupled with advanced analytics and mobile access, offers farmers precise control over irrigation, improving crop productivity while lowering costs. With the integration of weather forecasts and energy-efficient technologies, the system supports environmentally responsible farming.