KONGU ENGINEERING COLLEGE

(Autonomous)

Perundurai, Erode-638 060.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SMART IRRIGATION SYSTEM

IoT PLATFORM DESIGN METHODOLOGY

FOR

INTERNET OF THINGS AND CLOUD COMPUTING (22CST51)

SUBMITTED BY

ASHOK J (23CSR025)

ABISHEK MAHESH (23CSR011)

ARAVINTH L (23CSR021)

ATHISH M (23CSR029)

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI, ERODE – 638060

NOVEMBER 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

ASHOK J (23CSR025)

ABISHEK MAHESH (23CSR011)

ARAVINTH L (23CSR021)

ATHISH M (23CSR029)

Course Code: 22CST51

Course Name: INTERNET OF THINGS AND CLOUD COMPUTING

Semester: V

Certified that this is a Bonafide record of work for application projects done by the above students for **22CST51- INTERNET OF THINGS AND CLOUD COMPUTING** during the academic year 2024 - 2025.

Faculty In-Charge

ABSTRACT

The **Smart Irrigation System** is an IoT-based automation solution designed to optimize water usage in agriculture by intelligently controlling irrigation based on real-time soil moisture levels. This system employs an **Arduino UNO microcontroller** as the core processing unit, interfaced with a **soil moisture sensor** to continuously monitor soil conditions. When the soil becomes dry beyond a predefined threshold, the system automatically activates a **motor pump** through a **relay module**, ensuring that crops receive adequate water only when necessary. Once optimal moisture is restored, the pump is switched off, thereby conserving water and preventing over-irrigation.

By integrating simple yet effective components such as **transistors, resistors, LEDs, and jumper wires**, the system provides clear operational feedback — with indicator LEDs representing system states like pump activity and soil dryness. The setup is compact and user-friendly, suitable for small-scale farms, gardens, and educational demonstrations. The inclusion of IoT concepts enables future scalability for **remote monitoring, mobile notifications, and scheduled irrigation control** through cloud-based or app-integrated interfaces.

Developed using the **Arduino IDE**, this project demonstrates the fusion of embedded systems and IoT for sustainable agriculture. It significantly reduces human effort, minimizes water wastage, and enhances crop health by ensuring an optimal irrigation cycle. The system's design supports both **manual override and automated scheduling**, allowing flexibility for different environmental and crop conditions.

In essence, the **Smart Irrigation System** redefines modern farming practices through automation, precision, and environmental responsibility. It serves as a cost-effective and scalable model that promotes resource-efficient agriculture. Future developments could include **integration with weather forecasting APIs**, **fertilizer monitoring**, or **mobile app-based data visualization**, transforming it into a fully intelligent irrigation ecosystem. This project not only exemplifies innovation in smart agriculture but also contributes toward achieving sustainable development goals by leveraging technology for environmental conservation and food security.

CONTEXT:

INTRODUCTION

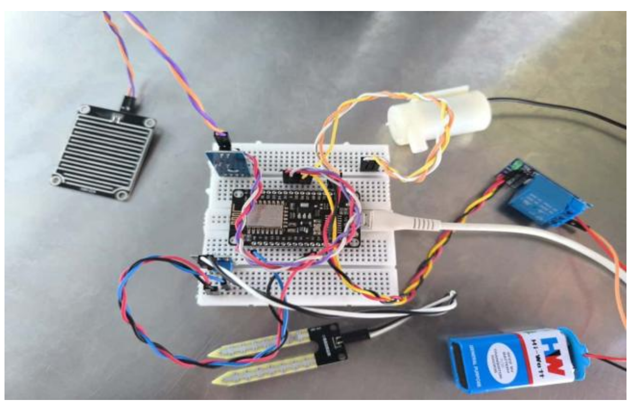
The **Smart Irrigation System** is an innovative **IoT-based automation solution** designed to revolutionize traditional agricultural practices by optimizing water usage and minimizing human intervention. Conventional irrigation methods often lead to excessive water consumption, uneven watering, and high dependency on manual monitoring, all of which contribute to decreased crop productivity and resource inefficiency. This project addresses these challenges by integrating **smart sensing technology** and **microcontroller-based automation** to create an efficient and intelligent irrigation process.

At the heart of the system lies the **Arduino UNO microcontroller**, which serves as the control unit, continuously processing real-time data received from a **soil moisture sensor**. The sensor measures the moisture content of the soil, and based on this input, the system automatically controls a **motor pump** via a **relay module**. When the soil moisture falls below a defined threshold, the motor pump is activated to irrigate the field, and once the optimal moisture level is achieved, the pump is automatically turned off. This ensures that plants receive adequate water while preventing wastage and over-irrigation.

The system also incorporates **LED indicators** to display operational states such as watering status and system activity, providing a simple yet effective user interface. Designed with scalability and affordability in mind, the Smart Irrigation System can be deployed in **agricultural farms, greenhouses, and home gardens**, making it suitable for both small- and large-scale applications. By combining embedded systems with IoT principles, the project demonstrates how technology can promote **sustainable agriculture** and support **environmental conservation** through intelligent automation.

This project not only enhances **crop health and yield** but also reduces **manual labor** and **resource wastage**, contributing to a more efficient and sustainable agricultural ecosystem. Future advancements can include features such as **wireless data transmission**, **cloud-based monitoring**, **integration with weather prediction APIs**, and **mobile app control**, transforming it into a fully automated and remotely accessible irrigation solution. The **Smart Irrigation System** thus represents a significant step toward achieving smart farming through innovation, precision, and environmental responsibility.

OUTPUT:



PURPOSE AND REQUIREMENT SPECIFICATION

1. **Automated Water Management**

   The Smart Irrigation System aims to optimize water usage by automatically monitoring soil moisture levels using a soil moisture sensor. This ensures that crops receive adequate water only when needed, eliminating the inefficiencies of traditional manual irrigation methods.

2. **Real-Time Monitoring and Control**

   The system uses the Arduino UNO microcontroller to process real-time data from the soil moisture sensor. Based on this data, it controls the motor pump via a relay module to start or stop irrigation automatically, providing continuous and intelligent management of soil hydration.

3. **Energy Efficiency**

   Designed for minimal energy consumption, the system activates the water pump only when the soil is dry, conserving both power and water. This feature ensures sustainable operation, making the system suitable for long-term agricultural use.

4. **User-Friendly Operation**

   The inclusion of LED indicators allows users to easily understand the system's status, such as whether irrigation is active or idle. The simple circuit design and automation make it highly accessible even for users with limited technical knowledge.

5. **Cost-Effective Design**

   Built with affordable and readily available components like the Arduino UNO, soil moisture sensor, and relay module, the system offers a low-cost automation solution for farmers and gardeners. Its modular nature allows easy assembly, maintenance, and expansion.

6. **Scalability and Customizability**

   The system can be scaled to cover larger agricultural fields or connected to multiple sensors for improved accuracy. It can also be enhanced with IoT connectivity for remote monitoring, data logging, and weather-based irrigation control, making it adaptable for diverse applications.

7. **Educational and Practical Significance**

   The project demonstrates the real-world application of IoT and embedded systems in sustainable agriculture. It serves as an excellent educational model for understanding how automation and sensor technology can improve efficiency, reduce manual labor, and conserve natural resources.

REQUIREMENTS FOR SMART IRRIGATION SYSTEM

- **Soil Moisture Detection and Monitoring**

The system must accurately detect the soil's moisture level using a soil moisture sensor. It should differentiate between dry and adequately moist soil to ensure precise irrigation control and prevent unnecessary watering.

- **Automatic Water Pump Control**

Upon detecting that the soil moisture level has dropped below a predefined threshold, the system must automatically activate the water pump. Once the desired moisture level is restored, the pump should automatically turn off, maintaining optimal soil conditions for plant growth.

- **Real-Time Operation and Response**

The Arduino UNO microcontroller must process sensor data in real time and control the relay and pump accordingly. This ensures immediate system response to changing soil conditions, improving water efficiency and crop health.

- **Hardware Reliability**

Components such as the Arduino UNO, relay module, soil moisture sensor, and motor pump must operate reliably over extended periods in outdoor or greenhouse environments. The circuit connections should be secure and weather-resistant for long-term performance.

- **Energy Efficiency**

The system should minimize power consumption by running the pump only when necessary. Efficient energy usage ensures sustainability and cost savings, making the system suitable for continuous operation in remote or low-power agricultural setups.

- **User-Friendly Indicators**

The system must include LED indicators to display operational status, such as when the soil is dry, irrigation is in progress, or the system is idle. This allows users to easily monitor functionality without complex interfaces.

- **Integration and Scalability**

The system should support the addition of multiple sensors and pumps to cover larger agricultural fields. Its modular design allows integration with IoT platforms or mobile applications for remote monitoring and control, ensuring adaptability for various scales of farming.

- **Data Accuracy and Communication**

Sensor readings and control signals must be stable, accurate, and consistent. In future IoT-integrated versions, secure communication protocols should be implemented to ensure reliable data transmission and protect user information.

- **Ease of Maintenance and Calibration**

The system design should allow easy calibration of the soil moisture sensor and replacement of components. Maintenance procedures and circuit documentation must be clear to ensure long-term usability and quick troubleshooting.

- **Adaptability for Various Applications**

The Smart Irrigation System must be versatile enough to be implemented in agricultural farms, home gardens, greenhouses, and nurseries. Its flexible design ensures compatibility with future upgrades, such as weather-based irrigation and fertilizer monitoring.

DOMAIN MODEL SPECIFICATION

The Smart Irrigation System consists of three primary entities, each playing a vital role in the efficient operation and automation of the irrigation process:

1. **Users:**
   • Primary stakeholders who monitor and manage the irrigation system.
   • Users may include **farmers, gardeners, or agricultural supervisors** who oversee the watering process.
   • They interact with the system to observe soil moisture levels and manually control the pump if required.

2. **System Hardware:**
   • Comprises the **Arduino UNO microcontroller**, **soil moisture sensor**, **relay module**, **motor pump**, **LED indicators**, and supporting components such as transistors and resistors.
   • The soil moisture sensor detects the moisture level of the soil and sends real-time data to the Arduino.
   • Based on this data, the Arduino automatically controls the water pump through the relay to maintain optimal soil conditions.

3. **Control Interface (Manual / IoT Integration):**
   • Acts as the bridge between the user and the system hardware.
   • Provides visual feedback through LEDs to indicate soil status (dry or moist) and pump activity.
   • In future IoT-integrated versions, this interface can be extended to mobile or web applications for **remote monitoring, irrigation scheduling**, and **data visualization**.

INFORMATION MODEL SPECIFICATION

a. Soil Moisture Sensor

Represents the sensor used to detect the moisture level of the soil for automated irrigation control.
Attributes:
• Sensor ID: Unique identifier for each soil moisture sensor.
• Moisture Threshold: Predefined limit to determine when irrigation should start or stop.
• Status: Operational state of the sensor (e.g., Active, Malfunctioning).
• Location: Physical placement of the sensor (e.g., Field Section A, Garden Zone 1).

b. Arduino UNO Microcontroller

Serves as the central control unit that processes data from sensors and operates the motor pump through a relay.
Attributes:
• Device ID: Unique identifier for the Arduino unit.
• Processing Capability: Ability to handle real-time sensor input and control signals.
• Connectivity: Communication interfaces used (e.g., wired connections, IoT integration in future).
• Power Source: Type of power supply (e.g., DC Adapter, Solar Power).
• Status: Operational condition (e.g., Active, Standby, Fault).

c. User

Represents the individual responsible for monitoring and managing the irrigation system.
Attributes:
• User ID: Unique identifier for each user.
• Name: Name of the user or farmer.
• Role: User's role (e.g., Owner, Operator, Technician).
• Contact Information: Email or phone number for receiving system updates or alerts (in IoT-integrated versions).

d. Motor Pump Module

Controls the flow of water to the crops based on soil moisture readings, managed through the relay circuit.
Attributes:
• Pump ID: Unique identifier for each motor pump.
• Type: Type of pump used (e.g., DC Pump, Submersible Pump).
• Status: Operational state (e.g., On, Off, Fault).
• Flow Rate: Amount of water delivered per minute during operation.

e. Relay Control Unit

Acts as a switching mechanism between the Arduino and the motor pump, enabling automatic irrigation.
Attributes:
• Relay ID: Unique identifier for the relay module.
• Voltage Rating: Input voltage capacity (e.g., 5V, 12V).
• Status: Operational condition (e.g., Active, Inactive).
• Control Signal: Input signal from Arduino that determines switching actions.

Attributes:

• **Interface ID:** Unique identifier for the control interface instance.

• **Real-Time Notifications:** Alerts or indicators showing soil dryness or irrigation activity.

• **Control Features:** Options to manually start or stop the motor pump and adjust soil moisture thresholds (in IoT-integrated versions).

• **Status Display:** Provides real-time updates of soil moisture levels, pump status, and overall system activity.

RELATIONSHIPS

a. **Soil Moisture Sensor – Arduino UNO Relationship**
Description: The soil moisture sensor continuously measures the soil's moisture content and sends real-time data to the Arduino UNO for processing and decision-making.

b. **Arduino UNO – Relay Module Relationship**
Description: The Arduino UNO processes the sensor data and controls the relay module to automatically switch the motor pump ON or OFF based on the soil's moisture level.

c. **Relay Module – Motor Pump Relationship**
Description: The relay acts as an intermediary switch between the Arduino and the motor pump, controlling the pump's operation according to the signals received from the microcontroller.

d. User – Interface Relationship
Description: The user interacts with the system through LED indicators or an IoT-based dashboard (future integration) to monitor the current soil condition, irrigation status, and system performance.

e. **Arduino UNO – User Interface Relationship**
Description: The Arduino updates the user interface with real-time status information, such as soil dryness or pump activity, and may also receive manual override commands from the user (in IoT-enabled versions).

ACTIONS

a. Monitor Soil Moisture
Continuously detect and display real-time soil moisture levels through the sensor and provide updates via LEDs or an IoT interface.

b. Automatic Irrigation Control
Activate or deactivate the motor pump automatically based on moisture readings to ensure optimal watering and resource efficiency.

c. Manual Control (Override)
Allow users to manually control the pump or adjust system thresholds through buttons or a connected app (in IoT-integrated systems).

d. Visual Feedback
Provide real-time visual indicators through LEDs showing system states — such as "Soil Dry," "Pump ON," or "Moisture Adequate."

e. Data Logging and Analysis
Record soil moisture readings, pump operation duration, and user actions for performance analysis and future optimization (in extended IoT versions).

CONSTRAINTS

a. Data Accuracy
Ensure precise soil moisture measurement to prevent incorrect irrigation actions and maintain optimal soil conditions.

b. Resource Efficiency
Minimize water and power consumption by activating the motor pump only when necessary, ensuring sustainable system operation.

c. **System Reliability**
Maintain continuous and reliable communication between the soil moisture sensor, Arduino UNO, and relay module to ensure consistent system performance.

SECURITY

a. **Access Control**
Restrict control and monitoring access to authorized users, especially in IoT-integrated versions, to prevent unauthorized operation of the irrigation system.

b. **Data Protection**
Secure data transmission between sensors, microcontroller, and cloud-based interfaces using standard IoT communication protocols to protect sensor data and user configurations.

INTERFACES

a. **User Interface**
LED indicators or a mobile/web-based IoT dashboard (in advanced versions) for monitoring soil moisture levels, pump activity, and overall system status. The interface also allows manual control of irrigation when required.

b. **Backend**
The Arduino UNO firmware processes sensor data, executes irrigation control logic, and manages real-time responses such as pump activation, LED signaling, and future IoT communication.

INTEGRATION

a. **External Tools**
Integration with cloud-based platforms such as ThingSpeak or Firebase (for IoT versions) can be implemented for storing historical soil data, logging irrigation events, and providing advanced analytics and notifications.

SCALABILITY AND PERFORMANCE

a. **Device Scalability**
Allow for the addition of multiple soil moisture sensors, relay modules, or motor pumps without requiring major reconfiguration. This ensures that the system can be easily expanded for larger agricultural fields or multiple irrigation zones.

b. **Real-Time Processing**
Ensure immediate detection of soil dryness and prompt activation of the motor pump. The Arduino UNO should efficiently process sensor input and respond instantly to maintain consistent soil moisture levels.

AUDIT AND LOGGING

a. **Irrigation Logs**
Maintain detailed logs of irrigation events, including timestamps, soil moisture readings, and pump activation cycles, to support analysis of water usage and crop health trends.

b. **Troubleshooting Logs**
Record sensor malfunctions, connection errors, or pump failures to assist in debugging and maintenance, ensuring long-term reliability of the system.

COMPLIANCE

a. **Regulatory Compliance**
Ensure adherence to IoT and environmental sustainability standards related to agricultural automation and smart water management systems.

b. **Data Privacy**
Protect user and environmental data, especially in IoT-integrated versions, by following standard data privacy and storage regulations to ensure secure cloud communication and user confidentiality.

SERVICE SPECIFICATION

1. **Soil Moisture Detection Service**
Continuously monitors soil moisture levels and sends data to the Arduino UNO for real-time processing.

2. **Irrigation Control Service**
Activates or deactivates the motor pump via the relay module based on soil moisture readings to maintain optimal soil hydration**.**

3. **Visual Feedback Service**
Controls LED indicators to provide real-time feedback on system states, such as “Soil Dry,” “Pump ON,” or “Moisture Adequate.”

4. **Manual Override Service**
Allows users to manually control the pump or adjust moisture thresholds through local buttons or a future IoT-integrated dashboard.

IoT Level Specification for Smart Irrigation System

This system can operate at IoT Level 2–3, focusing on device-level automation and sensor-based control. The Arduino UNO handles local processing and controls the pump automatically based on soil moisture. For advanced IoT integration, data can be transmitted to a cloud platform (e.g., Firebase or ThingSpeak) for remote monitoring, analytics, and scheduling. As a standalone system, it efficiently manages irrigation without requiring constant human intervention, while remaining scalable for future smart farming solutions.

FUNCTIONAL VIEW SPECIFICATION

The core functionalities of the Smart Irrigation System include:

1. **Continuous Soil Monitoring**
The soil moisture sensor constantly checks the water content of the soil and sends readings to the Arduino UNO.

2. **Automated Irrigation Response**
When soil moisture falls below a predefined threshold, the system automatically activates the water pump via the relay to irrigate crops, and stops once optimal moisture is restored.

3. **Visual Status Indication**
LED indicators provide real-time feedback on system states, allowing users to quickly assess whether irrigation is in progress or the soil is sufficiently moist.

4. **Manual Control and Configuration**
Users can manually control the pump or adjust the soil moisture threshold settings.

OPERATIONAL VIEW SPECIFICATION

1. **Initialization:**
   • The Arduino UNO initializes all connected components, including the soil moisture sensor, relay module, motor pump, and LED indicators.
   • For IoT-enabled versions, it establishes a connection with the cloud dashboard or mobile app for remote monitoring.

2. **Monitoring Mode:**
   • The system continuously monitors soil moisture levels.
   • The green LED indicates that soil moisture is within the optimal range and the system is idle.

3. **Irrigation Activation:**
   • When soil moisture falls below the defined threshold, the system automatically activates the motor pump via the relay module.
   • The red LED illuminates to indicate active irrigation.

4. **User Response and Manual Override:**
   • Users can manually stop the pump or adjust moisture thresholds using local buttons or, in IoT versions, via a mobile/web dashboard.
   • The system updates the interface with real-time soil status and pump activity, allowing users to monitor and control irrigation efficiently.

DEVICE AND COMPONENT INTEGRATION

1. **Arduino UNO:**
   Acts as the central controller, processing input from the soil moisture sensor and controlling outputs to the relay, pump, and LEDs.

2. **Soil Moisture Sensor:**
   Measures the soil's water content continuously and sends real-time data to the Arduino for decision-making.

3. **Motor Pump:**
   Delivers water to crops automatically when activated by the relay, ensuring optimal irrigation.

4. **Relay Module:**
   Acts as a switching interface between the Arduino and the motor pump, allowing automated control of water flow.

5. **LED Indicators:**
   Provide visual feedback: green LED for adequate soil moisture, red LED when irrigation is active.

6. **IoT Dashboard / App (Optional):**
   Enables remote monitoring and control of the system, including real-time updates on soil moisture, pump status, and threshold adjustments.

7. **Arduino Code / Firmware:**
   Manages sensor readings, executes irrigation logic, controls LEDs and pump operation, and communicates with the IoT dashboard if integrated.

APPLICATION DEVELOPMENT

The application for the Smart Irrigation System is developed using the Arduino IDE for firmware programming and Python (optional) for advanced data processing or IoT integration. Key features include:

1. **Sensor Integration:**
   Python scripts or Arduino code process real-time data from the soil moisture sensor to determine when irrigation is required.

2. **Irrigation Control Management:**
   The Arduino UNO automatically activates or deactivates the motor pump via the relay based on sensor readings, ensuring optimal watering cycles and efficient water usage.

3. **User Interface:**
   The system provides an intuitive interface through LED indicators for local monitoring. In IoT-enabled versions, a mobile or web application delivers real-time updates, displays soil moisture levels, pump status, and allows users to manually control the irrigation system or adjust thresholds.

CODE:

```
#include <DHT.h>
#include <DHT_U.h>  // optional but safe to include

// Define pins
#define DHTPIN D6        // DHT data pin connected to D6
#define DHTTYPE DHT11    // Sensor type: DHT11 or DHT22
#define SOIL_PIN A0      // Soil moisture analog pin
#define RAIN_PIN D5      // Rain sensor digital output pin
#define RELAY_PIN D7     // Relay control pin

// Create DHT object
DHT dht(DHTPIN, DHTTYPE);

// Variables
int soilValue;
int rainValue;
float humidity, temperature;

void setup() {
  Serial.begin(115200);
  dht.begin();

  pinMode(SOIL_PIN, INPUT);
  pinMode(RAIN_PIN, INPUT);
  pinMode(RELAY_PIN, OUTPUT);

  digitalWrite(RELAY_PIN, HIGH); // Relay OFF initially (HIGH = OFF for active LOW relays)
  Serial.println("Smart Irrigation System Started...");
}

void loop() {
  // Read sensors
  soilValue = analogRead(SOIL_PIN);   // 0–1023
  rainValue = digitalRead(RAIN_PIN);  // HIGH or LOW
  humidity = dht.readHumidity();
  temperature = dht.readTemperature(); // Celsius

  // Check if DHT read failed
```

```
  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed to read from DHT sensor!");
  } else {
    Serial.print("Humidity: "); Serial.print(humidity); Serial.println(" %");
    Serial.print("Temperature: "); Serial.print(temperature); Serial.println(" °C");
  }

  // Print values
  Serial.print("Soil Moisture: "); Serial.println(soilValue);
  Serial.print("Rain Detected: "); Serial.println(rainValue == LOW ? "YES" : "NO");

  // Logic:
  // Dry soil (value > 500) AND no rain (HIGH) => turn ON pump
  if (soilValue > 500 && rainValue == HIGH) {
    digitalWrite(RELAY_PIN, LOW); // Relay ON (active LOW)
    Serial.println("Pump ON (Irrigating...)");
  } else {
    digitalWrite(RELAY_PIN, HIGH); // Relay OFF
    Serial.println("Pump OFF");
  }

  Serial.println("---------------------------");
  delay(2000); // Wait 2 sec
}
```

CONCLUSION

The **Smart Irrigation System** exemplifies how IoT and embedded technologies can transform traditional agricultural practices. By integrating a **soil moisture sensor** with the **Arduino UNO** and optionally leveraging a mobile or web dashboard, the system ensures real-time monitoring and automated irrigation control. Its simplicity, cost-effectiveness, and scalability make it a practical solution for farms, greenhouses, and home gardens. Future iterations could include **cloud-based monitoring**, **AI-driven irrigation scheduling**, and **multi-sensor integration** to further optimize water usage, crop health, and operational efficiency.