**"IoT-Based Smart Irrigation System for Efficient Water Management in Agriculture"**

**221FJ01003**

**SIDDHARDHA AMIRISETTY**

**BCA 3RD**

1. **PROBLEM STATEMENT**

Describe the physical design of this IoT-based smart inigation system. Identify and explain the functions blocks of this lot system. What are the roles of each block, and how do they interact with each other to achieve efficient irrigation?

1. **INTRODUCTION**

Water scarcity and the need for sustainable agricultural practices have made efficient water management crucial in farming. Traditional irrigation methods often lead to water wastage and do not provide optimal conditions for crop growth. To address these challenges, this project presents the design and development of a Smart Irrigation System using Internet of Things (IoT) technologies.

The smart irrigation system is built around an ESP32 microcontroller, which serves as the central unit to collect data from various sensors and control actuators. Key components include a DHT22 sensor for measuring temperature and humidity, a soil moisture sensor to detect soil moisture levels, a relay module to control water pumps, and a servo motor to manage water flow. A Liquid Crystal Display (LCD) provides a user-friendly interface to display real-time data and system status.

By integrating these components, the system ensures efficient water usage, reduces human intervention, and provides a scalable solution for modern farming practices.

This document details the hardware and software requirements, the system's design and implementation, and the benefits of using an IoT-based approach for smart irrigation, which enhances both water efficiency and crop productivity.

1. **METHODOLOGY**

Physical Design of the IoT-Based Smart Irrigation System:

The physical design of the IoT-based smart irrigation system involves the integration of various sensors, actuators, and control units that work together to monitor environmental conditions and automate the irrigation process. The system is divided into several functional blocks, each playing a crucial role in ensuring efficient and automated irrigation.

***Functional Blocks***

Sensing Block

Soil Moisture Sensor: Measures the moisture level in the soil. This sensor is critical in determining whether the soil is dry and needs watering or if it has sufficient moisture.

DHT22 Sensor (Temperature and Humidity): This sensor measures the ambient temperature and humidity. The data from this sensor can be used to adjust irrigation schedules based on weather conditions.

Processing and Control Block

ESP32 Microcontroller: This microcontroller serves as the brain of the system. It collects data from the sensors, processes it, and makes decisions about when to activate or deactivate the irrigation system based on predefined thresholds. It also controls the actuators like the relay and servo motor.

Actuation Block

Relay Module: Controls the water pump by turning it on or off based on signals received from the ESP32. The relay acts as an electronic switch that can handle the high current required to operate the water pump.

Servo Motor: Controls the valve position to regulate water flow. The servo is controlled by the ESP32 to either start or stop water flow based on soil moisture levels.

Display and User Interface Block

Liquid Crystal Display (LCD): Provides real-time feedback to the user by displaying sensor readings such as soil moisture, temperature, humidity, and the status of the irrigation system. This allows for easy monitoring and manual intervention if necessary.

Power Supply Block

Power Supply Unit: Ensures that all components of the system receive a stable and sufficient power supply. This could include batteries or a direct power source depending on the installation requirements.

Roles and Interaction of Each Block

Sensing Block: Continuously monitors environmental parameters (soil moisture, temperature, humidity). The data is sent to the processing block for analysis.

Processing and Control Block: Analyzes the data received from the sensing block. If the soil moisture is below a certain threshold, the ESP32 microcontroller activates the relay to start the water pump and the servo motor to open the valve. If the moisture level is adequate, it turns off the pump and closes the valve.

Actuation Block: Directly interacts with the physical environment. The relay controls the power supply to the water pump, while the servo motor adjusts the valve. These components ensure that water is delivered only when necessary, conserving water and optimizing irrigation.

Display and User Interface Block: Provides the user with real-time information about the system’s operation. This helps in monitoring and allows manual control if needed.

Power Supply Block: Keeps all the other blocks operational by providing necessary power. The reliability and stability of the power supply are crucial for the uninterrupted operation of the system.

Here is how you can implement in wokwi software.

***Code:***

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <DHT.h>

#include <ESP32Servo.h>

// Pin definitions

#define DHTPIN 4

#define DHTTYPE DHT22

#define SOIL\_MOISTURE\_PIN 34

#define RELAY\_PIN 18

#define LED\_PIN 2

#define SERVO\_PIN 5

// Threshold for soil moisture (adjust as necessary)

const int moistureThreshold = 2000;

// Components

DHT dht(DHTPIN, DHTTYPE);

LiquidCrystal\_I2C lcd(0x27, 16, 2);

Servo waterPump;

void setup() {

  // Initialize Serial Monitor

**Serial**.begin(115200);

  // Initialize DHT Sensor

  dht.begin();

  // Initialize LCD

  lcd.begin(16, 2);

  lcd.init();

  lcd.backlight();

  lcd.setCursor(0, 0);

  lcd.print("Smart Irrigation Sys");

  // Initialize Servo

  waterPump.attach(SERVO\_PIN);

  waterPump.write(0); // Initial position

  // Initialize pins

  pinMode(SOIL\_MOISTURE\_PIN, INPUT);

  pinMode(RELAY\_PIN, OUTPUT);

  pinMode(LED\_PIN, OUTPUT);

  // Deactivate relay initially

  digitalWrite(RELAY\_PIN, HIGH);

}

void startWatering() {

  waterPump.write(90); // Assume 90 degrees is the pumping position

  lcd.setCursor(0, 1);

  lcd.print("Watering ON...");

}

void stopWatering() {

  waterPump.write(0); // Assume 0 degrees is the off position

  lcd.setCursor(0, 1);

  lcd.print("Watering OFF");

}

void loop() {

  // Read sensor values

  int soilMoistureValue = analogRead(SOIL\_MOISTURE\_PIN);

  float temperature = dht.readTemperature();

  float humidity = dht.readHumidity();

  // Display sensor readings on LCD

  lcd.setCursor(0, 1);

  lcd.print("Temp: ");

  lcd.print(temperature);

  lcd.print("C Hum: ");

  lcd.print(humidity);

  lcd.print("%");

  // Automatic watering based on soil moisture

  if (soilMoistureValue < moistureThreshold) {

    digitalWrite(RELAY\_PIN, LOW); // Activate relay for automatic watering

    startWatering();

  } else {

    digitalWrite(RELAY\_PIN, HIGH); // Deactivate relay

    stopWatering();

  }

  // LED indicator for watering status

  if (soilMoistureValue < moistureThreshold) {

    digitalWrite(LED\_PIN, HIGH);

  } else {

    digitalWrite(LED\_PIN, LOW);

  }

  // Add a small delay to avoid overwhelming the ESP32

  delay(2000);

}

Here Is what you need to connect or execute in wokwi software:

To implement the IoT-based smart irrigation system, we need specific hardware components and software tools to develop and run the code. Below is a detailed explanation of the software requirements, hardware components, and the functionality of each part of the code.

Software Requirements

Arduino IDE:

The Arduino Integrated Development Environment (IDE) is used for writing and uploading code to the microcontroller (ESP32). It provides a user-friendly interface to code, compile, and debug programs.

ESP32 Board Libraries:

The ESP32 board libraries are required for programming the ESP32 microcontroller using the Arduino IDE. These libraries provide functions to control the ESP32’s GPIO pins and to communicate with sensors and actuators.

Additional Libraries:

LiquidCrystal\_I2C: Used for interfacing the I2C LCD with the ESP32 to display sensor data.

DHT: Used for interfacing the DHT22 temperature and humidity sensor.

ESP32Servo: Used for controlling the servo motor.

Wire: A library for I2C communication, used by the LCD library.

Hardware Requirements

ESP32 Microcontroller:

The core processing unit of the system, responsible for reading sensor data, making decisions, and controlling actuators.

DHT22 Temperature and Humidity Sensor:

Measures ambient temperature and humidity, which can be used to monitor environmental conditions.

Soil Moisture Sensor:

Measures the moisture content in the soil. This sensor determines whether irrigation is needed based on soil dryness.

Relay Module:

Controls the water pump. The relay acts as an electronic switch that can turn the pump on or off based on signals from the ESP32.

Servo Motor:

Controls the water valve to regulate the flow of water. It is used to open or close the valve based on moisture readings.

Liquid Crystal Display (LCD) with I2C Interface:

Displays real-time data such as temperature, humidity, and soil moisture levels. It provides a visual interface for monitoring system status.

Water Pump:Pumps water to the plants when irrigation is needed.

Power Supply Unit:

Provides power to all the components.

1. **OUTPUT**

A computer screen shot of a device

Description automatically generated

1. **CONCLUSION**

The smart irrigation system is designed to automate the irrigation process efficiently by monitoring soil moisture and environmental conditions in real time. The system is a robust and scalable solution that leverages IoT technology to conserve water, reduce human intervention, and improve crop yield. By integrating sensors, actuators, and microcontrollers, the system intelligently manages water resources, making it a valuable tool for modern agriculture.

1. **REFERENCES**

[1] ESP32 Development Board: "ESP32-WROOM-32 Datasheet." Espressif Systems, 2024, <https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf>

[2] DHT22 Temperature and Humidity Sensor: Adafruit Industries. "DHT22 Temperature and Humidity Sensor Datasheet." 2024, <https://learn.adafruit.com/dht/overview>

[3] LiquidCrystal\_I2C Library: Arduino-Libraries. "LiquidCrystal\_I2C Library for Arduino." GitHub, 2024, <https://github.com/johnrickman/LiquidCrystal_I2C>

[4] ESP32Servo Library: "ESP32Servo Library Documentation." Arduino Libraries, 2024, <https://github.com/madhephaestus/ESP32Servo>

[5] Soil Moisture Sensor: Seeed Studio. "Grove - Capacitive Soil Moisture Sensor (Corrosion Resistant)." Seeed Studio, 2024, <https://wiki.seeedstudio.com/Grove-Capacitive_Moisture_Sensor-Corrosion-Resistant/>

[6] Microcontroller Programming: Monk, Simon. Programming Arduino: Getting Started with Sketches. McGraw-Hill Education, 2024.

[7] IoT in Agriculture: G. Kaur and D. Singh, "IoT Based Smart Irrigation System for Agricultural Applications," in International Journal of Advanced Research in Computer Science, vol. 11, no. 5, 2024.