DESIGN AND IMPLEMENTATION OF AN AUTOMATIC IRRIGATION SYSTEM

PRESENTED BY

ONUCHE CARNEY ENYO-OJO

MATRIC NO: 18245009

A PROJECT REPORT

Department of Electrical/Electronics Engineering

Faculty of Engineering

University of Abuja, Nigeria

SUPERVISED BY:

ENGR. DR. I. BEBEJI

January 2024

ABSTRACT

Irrigation is a crucial practice in agriculture, aiding in crop growth, landscape maintenance, and soil rehabilitation, especially in arid regions or during periods of low rainfall. This project presents the design and implementation of an automatic irrigation system using a microcontroller-based approach. The system integrates hardware components including Arduino Uno, moisture sensors, LCD display, relays, and water control mechanisms to regulate water supply to plants based on real-time soil moisture readings. The system monitors three critical parameters: soil moisture level, reservoir water level, and overhead tank water level. When soil moisture drops below a predefined threshold and water is available in the overhead tank, the solenoid valve automatically opens to irrigate the field. Similarly, when the overhead tank level is low and the reservoir has sufficient water, the pump activates to refill the tank. A low-water alarm warns users when both water sources are critically low. The system was successfully tested and validated, demonstrating its effectiveness in automating irrigation while conserving water resources and reducing manual labor.

1. INTRODUCTION

1.1 Background

Agriculture remains the backbone of many developing economies, yet traditional farming practices often lead to inefficient use of water resources. Manual irrigation methods are labor-intensive, time-consuming, and frequently result in over-watering or under-watering of crops. With increasing water scarcity and the need for sustainable agricultural practices, automation in irrigation systems has become essential. Automatic irrigation systems utilize sensors and control mechanisms to deliver water precisely when and where it is needed, optimizing crop growth while conserving water.

1.2 Problem Statement

Manual irrigation requires farmers to regularly check soil moisture and operate water pumps and valves, typically in the morning or evening. This process is not only physically demanding but also prone to human error—farmers may forget to water crops or misjudge the appropriate amount of water needed. Furthermore, manual systems cannot respond to changing environmental conditions in real-time, leading to water waste and suboptimal plant growth.

1.3 Project Objectives

The primary objectives of this project are to:

- Design an automated irrigation control system using microcontroller technology
- Monitor soil moisture levels in real-time using electronic sensors
- Automatically control water supply based on moisture readings
- Implement water level monitoring for both reservoir and overhead tanks
- Reduce manual labor and optimize water usage
- Provide visual feedback through LCD display
- Alert users when water sources are critically low

2. SYSTEM DESIGN

2.1 System Overview

The automatic irrigation system consists of three main subsystems: the sensing subsystem, the control subsystem, and the actuation subsystem. The sensing subsystem monitors soil moisture and water levels using analog sensors. The control subsystem, built around an Arduino Uno microcontroller, processes sensor data and makes irrigation decisions based on predefined thresholds. The actuation subsystem consists of relays, a solenoid valve, and a water pump that physically control water flow.

2.2 Hardware Components

Component	Specification	Function
Arduino Uno	ATmega328P	Main controller
Moisture Sensors	3× Analog sensors	Soil and water level monitoring
LCD Display	16×2 Character LCD	Status display
Relay Module	2-Channel, 5V	Switching control
Solenoid Valve	12V DC	Water flow control
Water Pump	12V DC, 1A	Water circulation
Transistors	2× BC547	Relay drivers
Diodes	2× 1N4007	Back-EMF protection
Buzzer	5V Active	Low water alarm
Power Supply	12V, 2A	System power

2.3 Pin Configuration

The Arduino Uno interfaces with all components through its digital and analog pins. Analog pins A0, A3, and A5 read the three moisture sensors (reservoir, soil, and overhead tank respectively). Digital pins 5-8 and 10-11 control the LCD display in 4-bit mode. Digital pins 12 and 13 control the valve and pump relays respectively, while pin 1 drives the alarm buzzer.

3. IMPLEMENTATION

3.1 Control Algorithm

The system operates using a simple yet effective threshold-based control algorithm. The microcontroller continuously reads analog values from the three moisture sensors and converts them to percentage values (0-300% scale). These values are displayed on the LCD and compared against predefined thresholds. The control logic implements four main decision paths:

Irrigation Control: When soil moisture drops below 20% and the overhead tank level exceeds 30%, the valve relay activates, opening the solenoid valve to irrigate the field. The valve closes automatically when soil moisture reaches 100%.

Water Circulation: When the overhead tank level falls below 20% and the reservoir level exceeds 30%, the pump relay activates to transfer water from the reservoir to the overhead tank. The pump stops when the tank reaches 100% capacity.

Low Water Warning: If both the reservoir and overhead tank levels drop below critical thresholds (30% and 20% respectively), the system activates the buzzer in a beeping pattern and keeps the pump off to prevent dry running.

Normal Monitoring: When all levels are within acceptable ranges, the system continuously monitors and updates the display without taking action.

3.2 Circuit Design

The circuit design follows standard embedded systems practices. Each moisture sensor receives 5V power and returns an analog voltage (0-5V) proportional to moisture level. The relay driver circuits use BC547 NPN transistors with 1kΩ base resistors to provide sufficient current for the 5V relay coils. 1N4007 flyback diodes protect the transistors from voltage spikes when the relays switch. The LCD operates in 4-bit mode to minimize pin usage, with a 10kΩ potentiometer controlling contrast. A 220Ω resistor limits current through the buzzer and LCD backlight.

3.3 Software Development

The system software was developed using the Arduino IDE and programmed in C/C++. The code is structured with clear initialization in the setup() function and continuous operation in the loop() function. The LiquidCrystal library handles LCD communication. The program reads sensors every second, converts readings to percentages, updates the display, and executes control logic. Configurable threshold constants allow easy system tuning for different soil types and crop requirements.

4. TESTING AND RESULTS

4.1 Testing Methodology

The system underwent comprehensive testing in three phases: component testing, integration testing, and system validation. Component testing verified individual sensors, the LCD display, relays, and the buzzer. Integration testing confirmed proper communication between the Arduino and all peripherals. System validation tested the complete operational workflow under various simulated conditions.

4.2 Test Results

Sensor Accuracy: The moisture sensors showed consistent readings with approximately ±10% accuracy, which is acceptable for irrigation control. Dry soil registered readings of 800-1000, while saturated soil registered 200-400.

Response Time: The system responded to moisture changes within 1-2 seconds, activating or deactivating the valve and pump as expected. The LCD updated in real-time without noticeable lag.

Relay Operation: Both relays operated reliably, with audible clicking confirming activation. The pump and valve responded correctly to relay states.

Alarm Function: The low-water alarm activated correctly when both water sources were low, producing an intermittent beeping pattern that was clearly audible.

Overall Performance: The system successfully automated the irrigation process, maintaining soil moisture within the desired range and managing water resources efficiently.

5. CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This project successfully demonstrates the design and implementation of an automatic irrigation system using microcontroller technology. The system effectively monitors soil moisture and water levels, automatically controls irrigation based on predefined thresholds, and provides real-time feedback through an LCD display. By eliminating the need for manual intervention, the system reduces labor costs, optimizes water usage, and ensures consistent crop care. The use of readily available, low-cost components makes the system economically viable for small to medium-scale agricultural applications.

The project achieved all stated objectives: automatic moisture monitoring, autonomous water control, multi-level water management, visual status indication, and low-water alerting. Testing confirmed the system's reliability and effectiveness in maintaining optimal soil moisture conditions while preventing water waste.

5.2 Recommendations

Based on the experience gained from this project, the following recommendations are proposed for future enhancements:

Sensor Upgrade: Replace resistive moisture sensors with capacitive sensors for improved longevity and accuracy, as resistive sensors are prone to corrosion over time.

IoT Integration: Add WiFi capability using ESP8266 or ESP32 modules to enable remote monitoring and control via smartphone applications.

Data Logging: Implement SD card storage or cloud database integration to record moisture trends and water usage for analysis and optimization.

Weather Integration: Incorporate weather forecast data to adjust irrigation schedules based on predicted rainfall.

Multi-Zone Control: Expand the system to manage multiple irrigation zones with different moisture requirements.

Solar Power: Add solar panels and battery storage for off-grid operation in remote agricultural areas.

Advanced Control: Implement PID control algorithms for smoother, more precise moisture regulation instead of simple on-off control.

6. REFERENCES

[1] Arduino Official Documentation. (2023). Arduino Uno Rev3. Retrieved from https://www.arduino.cc

[2] Allen, R. G., Pereira, L. S., Raes, D., & Smith, M. (1998). Crop Evapotranspiration: Guidelines for Computing Crop Water Requirements. FAO Irrigation and Drainage Paper 56.

[3] Awati, J. S., & Patil, V. S. (2012). Automatic Irrigation Control by Using Wireless Sensor Networks. Journal of Exclusive Management Science, 1(6), 1-7.

[4] Jones, H. G. (2004). Irrigation Scheduling: Advantages and Pitfalls of Plant-based Methods. Journal of Experimental Botany, 55(407), 2427-2436.

[5] Khriji, A., et al. (2012). Remote Sensing and Control for Establishing and Maintaining Digital Irrigation. International Journal of Advanced Information Technology, 2(1), 11-25.

[6] Singh, S., & Sharma, N. (2012). Research Paper on Drip Irrigation Management Using Wireless Sensors. International Journal of Computer Networks and Wireless Communications, 2(4), 461-464.

[7] ATmega328P Microcontroller Datasheet. (2018). Microchip Technology Inc.

[8] HD44780U LCD Controller/Driver Datasheet. (1998). Hitachi.

APPENDIX: TECHNICAL SPECIFICATIONS

A. System Specifications

Parameter	Specification
Operating Voltage	5V (Logic), 12V (Actuators)
Power Consumption	< 25W (peak)
Sensor Type	Resistive moisture sensors
Sensor Range	0-1023 (10-bit ADC)
Display Update Rate	1 second
Control Method	Threshold-based
Response Time	< 2 seconds
Operating Temperature	0°C - 50°C
Number of Zones	1 (expandable)

B. Threshold Values

Parameter	Threshold	Action
Soil Moisture	≤ 20%	Activate valve (if water available)
Soil Moisture	≥ 100%	Deactivate valve
Overhead Tank	≤ 20%	Activate pump (if reservoir has water)
Overhead Tank	≥ 100%	Deactivate pump
Reservoir	≤ 30%	Activate alarm (critical low)
Reservoir	≥ 50%	Deactivate alarm