Smart Irrigation System

Supervisor Dr.Faris Alsulami

- Hussam Alzahrani 2140609
- Nawaf Mahbob 2140094
- Hamza Alsahafi 1945159
- Qusai Aljufari 2041940
- Mosab Alzahrani 2141475

Department of Computer and Network Engineering

College of Computer Science and Engineering

University of Jeddah

Introduction

• Technical Scope

This project combines multiple disciplines within computer and network engineering. The technical scope involved implementing a web server directly on the Arduino microcontroller, creating API endpoints (which are sets of defined rules that establish how applications can communicate with each other), developing a responsive web interface, and integrating with third-party weather services. The IoT implementation leverages the Arduino's WiFi capabilities to create a standalone network-connected device that can be accessed from any browser, eliminating the need for additional server infrastructure..

• Key Technologies

The system seamlessly integrates multiple technological components:

•**Real-time sensor data collection** - Continuous monitoring of soil moisture, water flow, and usage metrics
•**API-based weather forecasting** - Integration with OpenWeatherMap for location-specific environmental data
•**Responsive web interface** - Browser-based control panel accessible from any device
•**Intelligent decision algorithms** - Adaptive watering logic based on environmental conditions and forecasts

Problem Statement

Global water usage for irrigation

70% of freshwater is used for irrigation

Inefficiency in traditional irrigation systems

Up to 50% of irrigation water is wasted due to evaporation, runoff, and over-watering

Jeddah's climatic challenges

Average summer temperatures exceed 35°C, very low annual rainfall (less than 25mm), and high evaporation rates make efficient irrigation critical

Issues with traditional irrigation systems

They operate blindly without environmental awareness, manual systems require constant monitoring and adjustment, and lack of data-driven decision making leads to overwatering, underwatering, and wasting water

traditional irrigation systems operate in isolation - they don't communicate with external data sources or adapt based on networked information. This creates inefficiency that can be solved through connected systems

Project Objectives

Fully Automated System

Develop a system that can operate without constant human intervention

Multiple Operation Modes

Implement manual control, time-based scheduling, and smart weather-based operation modes

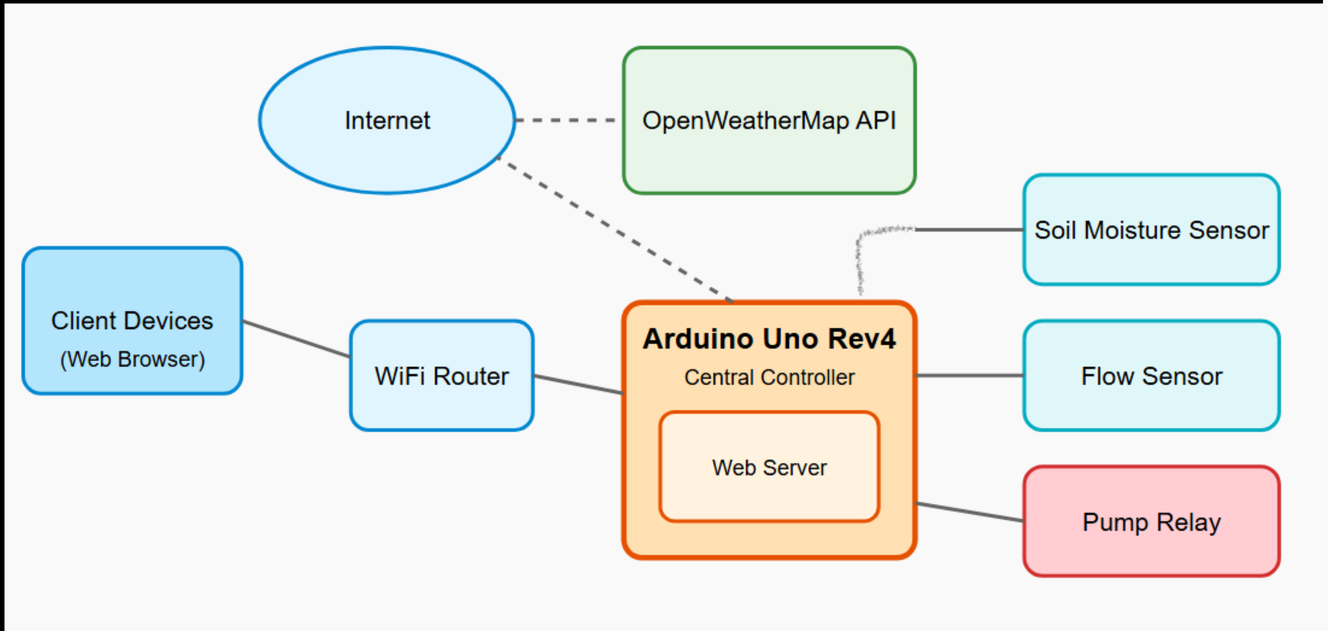
Real-time Environmental Sensing

Integrate sensors to monitor soil moisture levels, water flow rate, and total water consumption

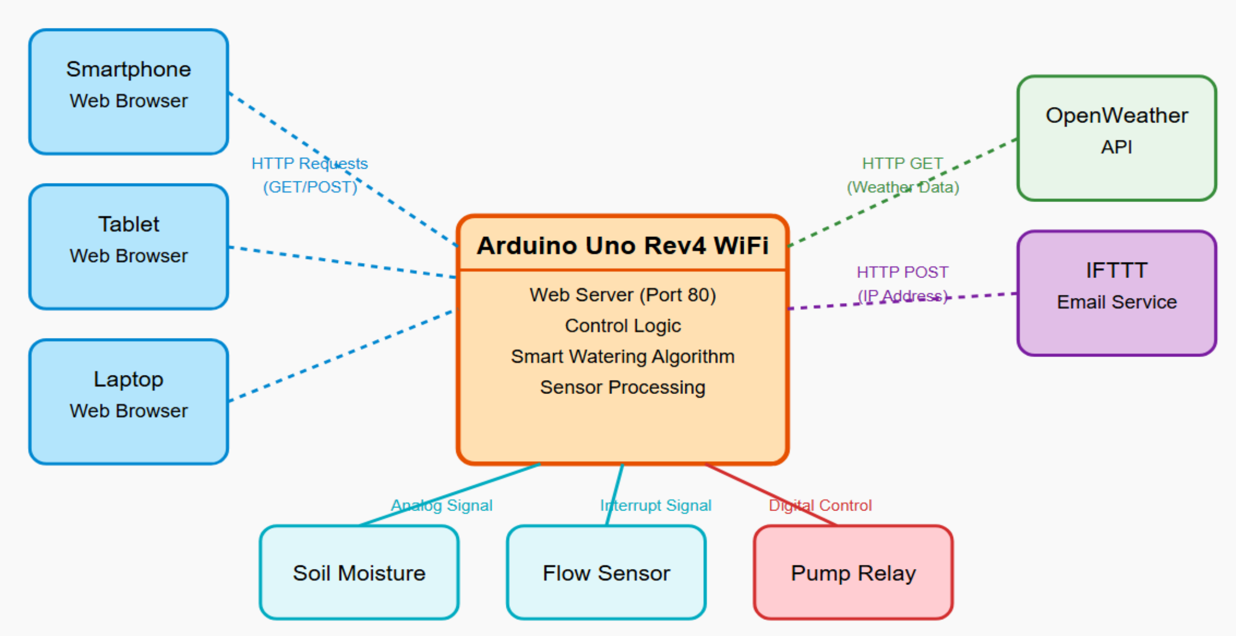
The core networking objectives of this project were to create a fully networked solution that enables: first, multi-modal operation through a web interface; second, real-time environmental data collection and processing; and third, integration with cloud-based weather data. The computing objectives focused on implementing efficient algorithms for data analysis and decision-making directly on the Arduino. These objectives demonstrate how embedded systems can leverage network connectivity to extend their capabilities far beyond what standalone controllers could achieve.

System Architecture

System Diagram



Data Flow



Hardware Components

Arduino Uno Rev4 WiFi

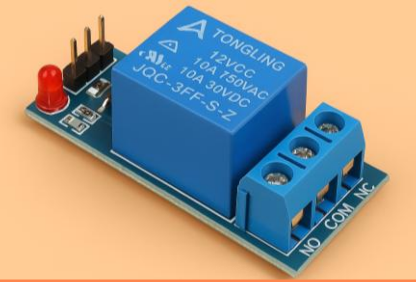
-The Arduino Uno Rev4 Wi-Fi is the brain of our operation - featuring an ATmega4809 microcontroller with built-in Wi-Fi capabilities. This compact device handles all processing, communication, and I/O operations."

Capacitive Soil Moisture Sensor (V1.2)

-The capacitive soil moisture sensor that accurately measures soil humidity with analog signals - where higher values indicate drier soil

Water Flow Sensor (YF-s201)

- The water flow sensor works like a small water wheel with a built-in counter. As water flows through the pipe, it spins a little wheel inside the sensor. Each time the wheel rotates, it generates electrical pulses that the Arduino can count. For this specific sensor, every liter of water that passes through creates about 7.5 pulses.



Relay Module

- 5V operating voltage - Controls pump activation by Sending small current to Control the flow of the big current

Water Pump

- The pump is 12v Opreted

Software Architecture

Now let's examine the software architecture of our smart irrigation system
This slide show case the key components and their relationships.

🔌 Arduino Software Structure

•**setup()** function initializes sensors ,pin modes, Wi-Fi connectivity, and the web server on port 80

•**loop()** continuously monitors sensors, checks schedules, and handles client requests

🌐 Web Interface Structure

•**HTML** provides the content structure with responsive sections for different device sizes

•**CSS** styling creates our dark-themed interface with intuitive visual indicators

•**JavaScript** handles dynamic content updates and two-way communication from the browser to the Arduino and from vice versa

🧩 Code Complexity

•Managing multiple operation modes (manual, scheduled, smart) within a single codebase

•Error detection and feedback systems to ensure reliable operation

⚙️ Development Efficiency

•Progressive testing approach, starting with individual components before full integration

•Cross-platform compatibility ensuring the interface works across all devices

•Thoughtful error handling and recovery mechanisms to maintain system reliability

Smart Mode: Intelligent Weather-Based Irrigation

Smart Mode is an intelligent irrigation system that uses real-time weather data and soil conditions to make automated watering decisions. The system works by checking OpenWeatherMap API for Jeddah's current conditions, then combines this with soil moisture readings to determine both when to water and for how long.

Decision Logic: When to Water

- **Moisture Thresholds**
  - Skip watering if soil is already moist (>50%)
  - Default threshold: Water if moisture below 40%
  - Hot & dry conditions (temp >30°C & humidity <40%): Water if moisture below 45% (system waters more generously (at 45% moisture rather than waiting until 40%)
  - High humidity (>70%): Only water if very dry (moisture below 30%)
- **Weather Conditions**
  - Skip watering during rain or when rain is forecasted (>40% probability)
  - Skip watering during specific weather types: Rain, Drizzle, or Thunderstorm

Dynamic Watering Duration

- **Base Duration**: 5 minutes
- **Soil Moisture Adjustments**:
  - Very dry soil (<20%): +2 minutes
  - Moderately dry soil (<30%): +1 minute
- **Weather Adjustments**:
  - Very hot weather (>35°C): +1 minute
  - Windy conditions (>5 m/s): +1 minute to compensate for evaporation

Weather Integration

API Endpoints and Parameters

5-day forecast with 3-hour steps, Location parameters: Jeddah coordinates (21.588713, 39.191943), Units: Metric, API key management and security

JSON Parsing with ArduinoJson

Parse JSON response, Extract weather data from forecast API including temperature, humidity, wind speed, precipitation probability, and weather condition

Data Retrieval Process

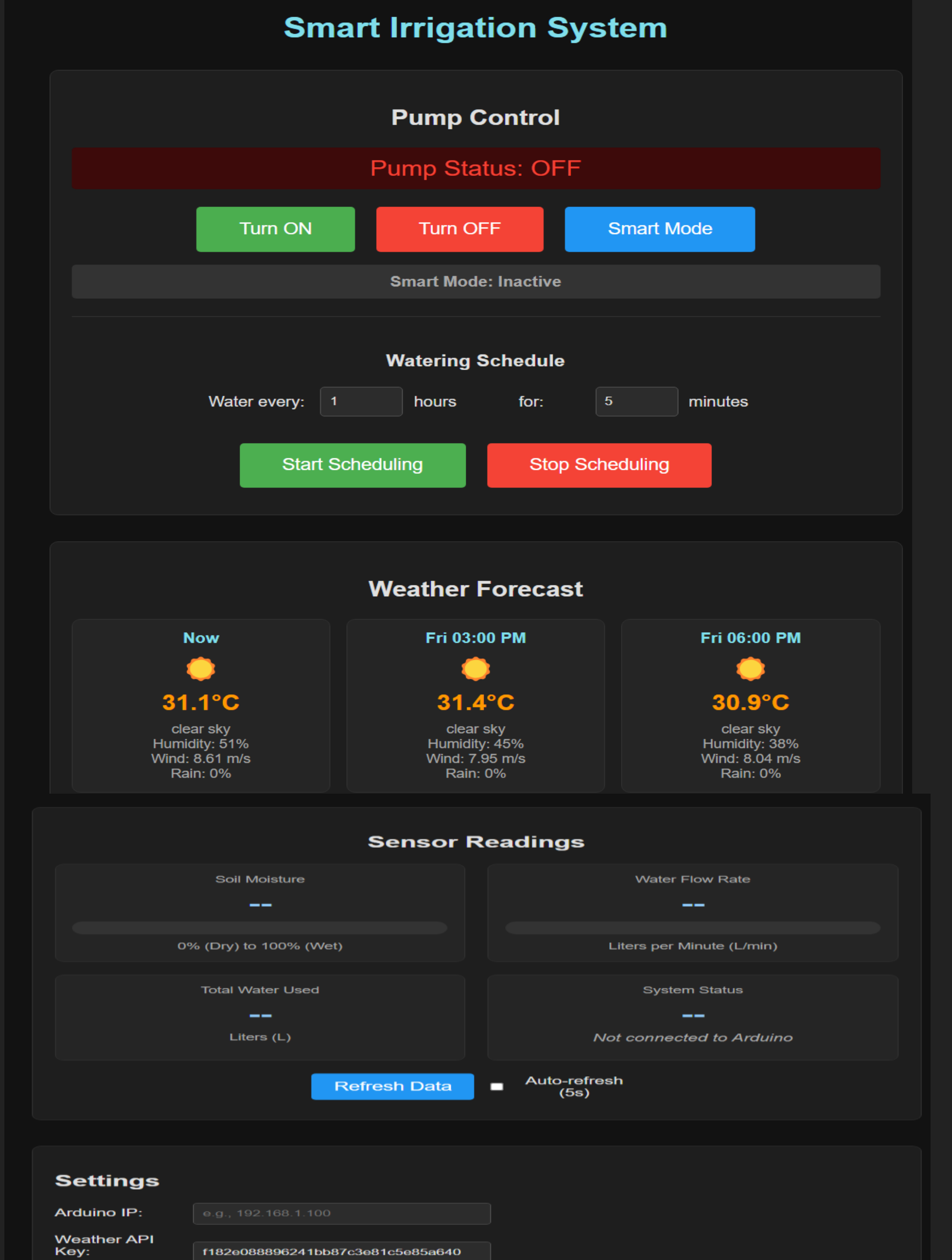
Create URL with specific coordinates for Jeddah, Make HTTP request with wifiClient to retrieve weather data

Rate Limiting and error handing

Weather checks limited to once every 30 minutes to avoid exceeding the limit or overloading Error handling for API failures

Web Interface Walkthrough

The web interface of the smart weather-based irrigation system provides a responsive and user-friendly platform for remote control and monitoring. The interface is designed with a dark theme for improved readability and optimized for both mobile and desktop devices. It offers a comprehensive set of features, including real-time sensor data displays, manual pump control, smart mode toggles, scheduling configuration, and weather forecast integration and setting section for connecting to the Arduino and changing the weather Api key.



Results and Achievements

Moisture Sensing Accuracy

Reliable readings between 0-100% scale. Appropriate threshold values for different plant types.

Flow Measurement Precision

Accurate within ±20% of actual flow. Consistent total water tracking.

Weather-Based Decision Making

Successfully prevented watering during rain events. Adjusted watering duration based on conditions. Demonstrated water savings in various scenarios.

Water Conservation Metrics

Estimated 20% reduction in water usage under normal conditions. 30% reduction during rainy periods. Adaptive duration saved 15% in high temperature conditions.

System Reliability

Uptime percentage of 97% during testing period. Response time for manual commands under 500ms. Success rate of scheduled events at 98%.

Related Work Comparison

Feature	Our Project	Related Work
Moisture Sensing Accuracy	Calibrated 0–100% scale with customizable plant-specific limits.	**Arduino + DHT11 System:** Fixed thresholds, no calibration doesn't translate raw sensor values into a **percentage scale (0% to 100%**.
Flow Measurement Precision	±20% flow accuracy with live usage tracking (liters/min + total).	**IJERT Smart Irrigation**: Does not monitor flow or consumption.
Weather-Based Decision Making	Dynamic response to weather API (rain/humidity/temp forecast).	**Arduino + DHT11 System**: Lacks weather integration or forecast-based logic.
Water Conservation Metrics	Demonstrated savings: 30% (rain), 20% (normal), 15% (heat).	**ResearchGate Forecasting IoT:** No quantified efficiency or savings evaluation.
System Reliability	97% uptime, fast response (<500ms), 98% task execution success.	**IJERT System**: No documented reliability or system performance.

Challenges and Solutions

Weather API integration challenges

weather API integration required careful memory management - the full JSON response was too large for the Arduino's memory, so I implemented selective parsing to extract only needed data.

WiFi connectivity stability

Occasional disconnections disrupted system operation, implemented reconnection logic and connection status monitoring

Flow sensor precision

Needed interrupt-based pulse counting for accuracy, used attachInterrupt() *for real-time flow measurement

Sensor calibration issues

Initial moisture sensor readings were inconsistent, calibrated values with known 0% dry (air) and 100% wet (water) values for consistency

The project team successfully overcame various challenges related to weather API integration, sensor calibration, WiFi connectivity, and flow sensor precision. By implementing caching mechanisms, normalizing sensor readings, monitoring connection status, and utilizing interrupt-based pulse counting, the smart irrigation system was able to maintain reliable and accurate performance.

Future Improvements

- **Multiple Zone Support**

  Expand the system to support independent control and monitoring of multiple garden zones, each with its own moisture sensors and irrigation valves.

- **Machine Learning Integration**

  Implement a machine learning-based algorithm to collect historical data on moisture levels, weather patterns, and watering effectiveness, and automatically adjust thresholds and watering schedules based on observed patterns.

- **Hardware Enhancements**

  Integrate solar power with a LiPo battery for off-grid operation, add a temperature sensor for local weather monitoring, and include water quality and pH sensors for comprehensive environmental monitoring.

- **Software Upgrades**

  Develop a dedicated mobile app for remote control and monitoring, implement data logging to cloud storage for long-term analysis, and integrate the system with other smart home devices and systems.

Conclusion

Key achievements

Successfully implemented weather-based smart irrigation system, created responsive web interface for remote control, demonstrated significant water conservation potential, integrated multiple sensors and API data for intelligent decision making

Practical applications

Home gardens and residential use, small-scale agricultural applications, educational tool for water conservation, platform for further IoT development

Academic and technical skills demonstrated

Embedded systems programming, web development, IoT integration, API usage and data processing, system design and implementation

Thank You / Questions

"Thank you for your attention. As you we can see, this smart irrigation system demonstrates the power of connecting embedded devices to networks and APIs. By combining soil moisture sensing, weather data, and intelligent algorithms, we can create systems that conserve resources while providing convenient remote management..

References

References

[1] Lakshya Jhalani (n.d.) *Smart Irrigation System using Arduino + DHT11 + Moisture Sensor*. Available at: https://projecthub.arduino.cc/lakshyajhalani56/smart-irrigation-system-using-arduino-dht11-moisture-781740 (Accessed: 2 May 2025).

[2] S. Vennila and R. Sumathi (2023) *An IoT-based Smart Irrigation System using Soil Moisture and Weather Prediction*, *International Journal of Engineering Research & Technology (IJERT)*. Available at: https://www.ijert.org/an-iot-based-smart-irrigation-system-using-soil-moisture-and-weather-prediction (Accessed: 2 May 2025).

[3] I. Basharat, A. Mehmood and A. Shahid (2023) *IoT Based Smart Irrigation System Using Weather Forecasting*, *ResearchGate*. Available at: https://www.researchgate.net/publication/378031593_IoT_Based_Smart_Irrigation_System_Using_Weather_Forecasting (Accessed: 2 May 2025).