Automatic Plant Watering System

Md Mochaddec Hossain, BSc. *CSE*

Abstract— The "Automated Plant Watering System" presented in this report addresses the demand for efficient and autonomous plant care systems. The primary objective is to automate the irrigation process in a cost-effective and efficient manner, especially beneficial for urban gardening and small-scale agricultural applications. Utilizing soil moisture sensors, the system intelligently administers water to plants based on real-time soil moisture data, significantly conserving water and promoting plant health. Arduino Uno processes real-time data from the sensors, activating the water pump when moisture levels fall below a set threshold. This report covers the system's development phases, from conceptual design to prototype testing, emphasizing the integration of hardware components and software programming. The project demonstrates significant water conservation, reducing manual intervention while promoting sustainable plant care practices. The results showcase the system's effectiveness in maintaining optimal soil moisture, contributing to advancements in smart and eco-friendly agricultural practices.

Index Terms— Automated Plant Watering, Automatic Irrigation, Arduino Uno, Soil Moisture Sensors, Sustainable Plant Care Practice, Smart Gardening, Water Conservation, Eco Friendly Agriculture.

INTRODUCTION

A. Background of Study and Motivation

The increasing global focus on sustainable agriculture and smart technologies has spurred the development of innovative solutions to address challenges in plant care. Traditional methods of manual watering are often inefficient, leading to water wastage and inconsistent plant health. In response to these issues, the Automatic Plant Watering System utilizing a microprocessor, specifically the Arduino Uno, emerges as a contemporary solution.

Water scarcity, changing climate patterns, and the need for resource-efficient practices underscore the urgency of adopting automated systems in agriculture. The Arduino Uno, with its robust capabilities in processing data and controlling multiple devices, offers a suitable platform for developing an efficient and cost-effective Automated Plant Watering System.

Motivated by the imperative to enhance water conservation and optimize plant growth, this project seeks to leverage the power of microprocessor technology. The Arduino Uno's versatility allows for the integration of soil moisture sensors and water delivery mechanisms, enabling precise and automated control over the plant watering process. By utilizing open-source hardware and software, the system becomes not only effective but also affordable and customizable.

B. Project objectives

The primary objective is to design and develop an automated plant watering system that integrates the Arduino Uno microprocessor. It is capable of detecting moisture level in the soil and capable of taking the decision of switching ON/ OFF water the motor. The automation process is expected to rely on data-driven decisions, where the watering schedule and quantity are determined based on real-time data rather than fixed, pre-programmed schedules. This approach ensures more efficient water usage and better care for the plants. By using soil moisture sensors and intelligent algorithms, the system aims to water plants only when necessary, reducing water wastage. The system is a user-friendly interface that helps users to adjust moisture level thresholds.

C. A Brief Outline of the Project

This project revolves around the design and development of an Automatic Plant Watering System powered by an Arduino Uno microcontroller. The aim is to create a system that automates the process of watering plants, ensuring efficient use of water and optimal plant health. The project encompasses various stages, as detailed in the following sections:

System Design and Conceptual Framework: This initial stage involves outlining the system's architecture, selecting appropriate sensors, and designing the circuitry. The design is adaptable, allowing adjustments for different watering needs.

Programming the microcontroller: The focus here is on programming the Arduino Uno to process input from sensors and control the watering mechanism.

System Assembly and Integration: In this phase, the physical assembly of the system is completed. This involves setting up the Arduino Uno, connecting sensors, and integrating the watering mechanism.

Testing and Calibration: Rigorous testing is conducted to ensure the system operates as intended. Calibration of sensors is performed to achieve accurate measurements. The system's response to different soil moisture levels is fine-tuned during this phase.

Documentation and Reporting: Comprehensive documentation of the entire project is prepared, including design specifications, and software code.

LITERATURE REVIEW

Microcontrollers like the Arduino Uno have gained popularity for their versatility in automating various tasks. In the context of plant care, they offer the ability to monitor and regulate moisture levels, ensuring that plants receive optimal hydration. The Arduino Uno is a popular choice for plant watering systems due to its abundant I/O pins, memory capacity, and ease of programming. Researchers and hobbyists have utilized these features to create efficient and customizable plant watering solutions.

One crucial aspect of an automatic plant watering system is accurate sensing of soil moisture levels. Various sensor technologies, including resistive soil moisture sensors and capacitive sensors, have been employed to measure soil moisture content accurately. Researchers have also explored additional sensors for monitoring environmental conditions like humidity, temperature, and light intensity to further optimize plant care.

Several automated plant watering systems have been developed, showcasing the practical application of microcontrollers. These systems range from simple setups for indoor potted plants to complex irrigation systems for large agricultural fields.

Effective control algorithms are essential for determining when and how much to water plants. Researchers have proposed various algorithms, including threshold-based control, fuzzy logic, and machine learning models. These algorithms consider factors such as plant species, soil type, and weather conditions to tailor watering schedules.

Efficient power management is crucial for ensuring the long-term operation of an automatic plant watering system. Researchers have explored techniques to reduce power consumption, such as using low-power sleep modes when the system is idle or integrating solar panels for energy harvesting.

The field of automatic plant watering systems continues to evolve. Future research may focus on enhancing the intelligence of these systems by integrating machine learning for predictive watering, utilizing advanced sensor technologies, and developing more energy-efficient solutions. Additionally, the integration of smart home platforms and voice assistants may become more prevalent, allowing users to interact with their plant care systems effortlessly.

Methodology and Modeling

A. Introduction

Watering is the most important cultural practice and most labor-intensive task in daily greenhouse operation. To make the gardener works easily, the automatic plant watering system is created. The project uses Arduino Uno Microcontroller that is programmed in such a way that it will sense the moisture level of the plants and supply the water if required. This type of system is often used for general plant care, as part of caring for small and large gardens. Normally, the plants need to be watered twice daily, morning and evening. So, the microcontroller has to be coded to water the plants in the greenhouse about two times per day.

B. Working Principle of the Proposed Project

The working principle of the Automatic Plant Watering System utilizing an Arduino Uno microcontroller is based on the integration of hardware components and Arduino program to create a responsive and efficient irrigation mechanism. The system is designed to autonomously monitor soil moisture levels and accordingly administer water to plants. This section outlines the core operational aspects of the system.

B.1 Process of Work

Soil Moisture Detection: The system employs soil moisture sensors, which are strategically placed in the soil near the plants. These sensors are capable of detecting the moisture content in the soil through resistive measurement techniques. When the soil's moisture level falls below a pre-set threshold, indicating a need for watering, the sensor sends a signal to the Arduino Uno.

Arduino Uno Microcontroller: At the heart of the system is the Arduino Uno, a versatile microcontroller board that serves as the primary control unit. The Arduino Uno processes the input received from the soil moisture sensors. It runs a program written in the Arduino programming language, which includes logic for interpreting sensor data, making decisions about watering, and controlling the output to the watering mechanism.

Activation of Watering Mechanism: Upon receiving a signal that the soil is dry, the Arduino Uno activates the watering mechanism. This usually involves controlling a relay which triggers a water pump connected to water supply. The relay opens, allowing current to flow through the pump, which could also be a drip irrigation setup, or a sprinkler system.

Timer and Control Logic: The software program includes a timer and control logic to regulate the duration and frequency of watering. This is essential to avoid over-watering or under-watering. The system can be programmed to water the plants for a specific duration or until the soil moisture reaches the desired level.

Feedback Loop and Adjustments: A feedback loop is integral to the system. After the watering process, the soil moisture sensors continue to monitor the soil's moisture level, providing continuous feedback to the Arduino Uno. This feedback allows for real-time adjustments in the watering schedule and ensures that the plants receive the right amount of water over time.

Moisture control: Two switches are configured alongside the Arduino Uno in a way that pressing one switch increases, while pressing the other decreases the required moisture level in the Arduino program. This modification feature allows for the customization of the system to accommodate various plant and soil requirements.

C. Description of the Components

Arduino Uno: The Arduino Uno is a microcontroller board designed for creating interactive electronics projects. It is built around the ATmega328P microcontroller, which is the brain of the board.

Moisture sensor module: Moisture sensor module an electronic device used to measure the moisture level in soil or other substrates. These modules are commonly used in agriculture, gardening, and automated plant watering systems to ensure that plants receive the appropriate amount of water.

5v relay module: A 5V relay module is an electronic device that uses a 5-volt control signal to switch a separate high-power circuit on or off. Relay modules are commonly used in various electronics and automation projects to control larger electrical loads with a low-voltage microcontroller or other control circuit.

5v dc pump with pipe: A 5V DC pump with a pipe is a common component used in various applications, such as water circulation, cooling systems, DIY projects, and more. This type of pump is designed to operate on a 5-volt DC power supply and typically includes a small motor and an attached pipe or tubing for fluid transfer.

9v battery: A 9V battery is a compact, rectangular-shaped, and commonly used portable power source that provides 9 volts of direct current (DC) electrical energy. These batteries are popular for various applications due to their convenient size and voltage output.

Breadboard: A breadboard is a fundamental and versatile tool used in electronics prototyping and experimentation. It allows quick build and test electronic circuits without the need for soldering.

Pushbutton: A pushbutton, also known simply as a "button" or "push switch," is a simple yet widely used electrical component that allows manual control of electrical circuits. It is commonly found in various electronic devices, control panels, and user interfaces.

LED: LED is a semiconductor device that emits light when an electric current passes through it. LEDs are widely used for various lighting and indicator applications due to their energy efficiency, long lifespan, and versatility.

Battery connector: A battery connector, also known as a battery terminal connector or battery cable connector, is an electrical component used to establish a secure electrical connection between a battery and an external device or circuit.

Jumper wires: Jumper wires are electrical wires with connectors or terminals at both ends used for making temporary electrical connections on a breadboard, between components on a circuit board, or in various electronic and electrical projects. They are a fundamental tool in electronics prototyping and wiring.

D. Experimental Setup

Step-1: The moisture sensor's output is connected to the Arduino Uno's analog input A0. The VCC and GND are connected to 5V and GND of the Arduino.

Step-2: The relay is connected to digital output pin3 of the Arduino.

Step-3: A 9V battery and the dc water pump are connected via the relay.

Step-4: Two push buttons are set to pin 4 and 5 to increment and decrease our required moisture level. And two LEDs are set at pin 12 and 13 to indicate the change of required moisture.

Step-5: The water pump is set up with a water container and the plant, making sure there is enough water for watering.

Step-6: The optimal moisture for our chosen plant is 400 to 500. So, we set our watering trigger at 250 in the Arduino code.

Step-7: The code is compiled and uploaded to Arduino Uno. After the upload, the Arduino is powered by a 9V battery.

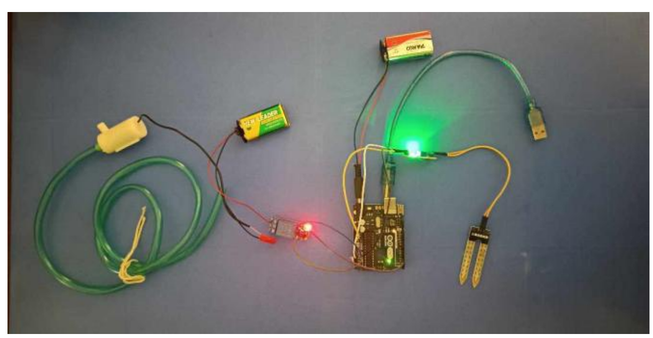

Fig. 1. Sensor and relay connections

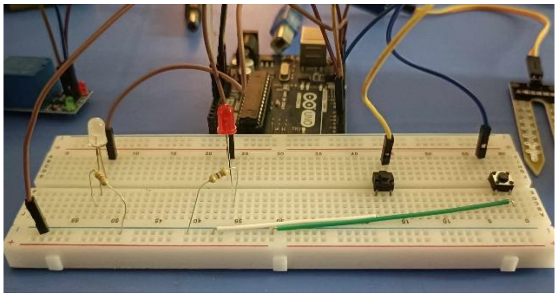

Fig. 2. Increment-decrement buttons

Fig. 3. Plant set up with watering mechanism

Results and Discussions

A. Simulation:

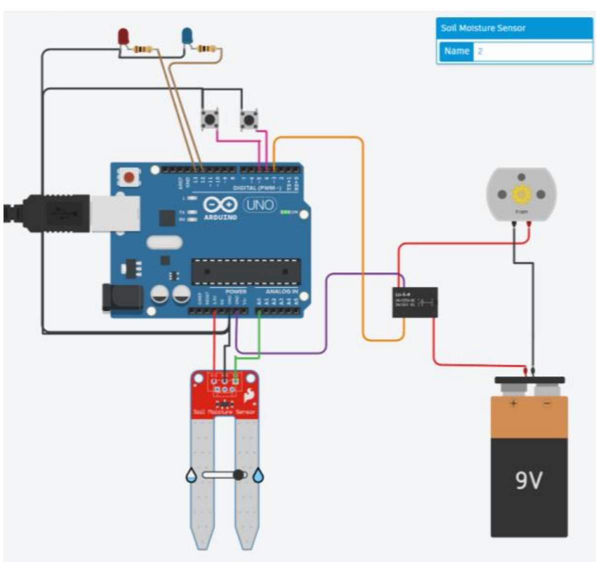

Fig. 4. Watering mechanism OFF at high moisture.

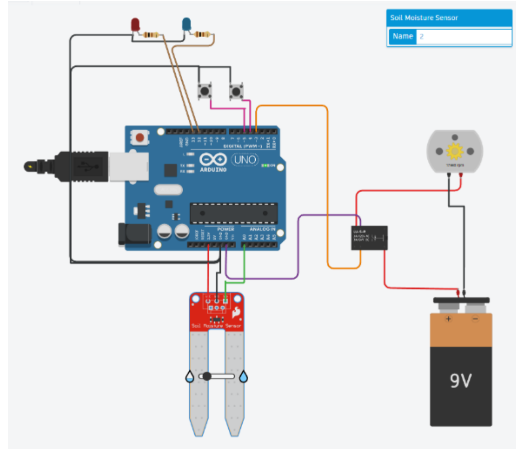

Fig. 5. Watering mechanism ON at low moisture.

B. Experimental results

The experimental results obtained from the automatic plant watering system, as recorded at different time intervals, provide valuable insights into the system's performance and its dynamic response to changing soil moisture conditions.

The optimal moisture here is around 400 to 500. So, we set our watering trigger at 250.

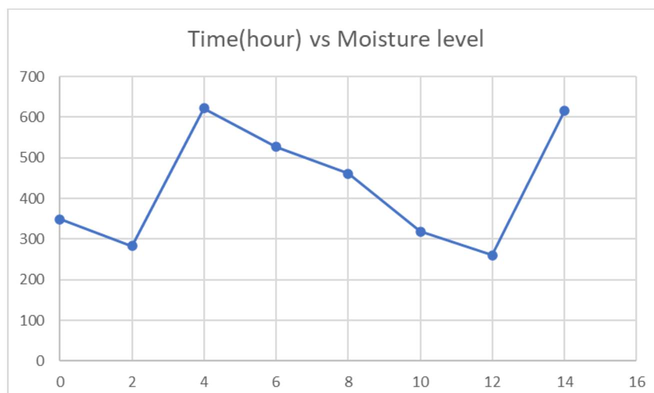

Fig. 6. Moisture level of plant at different times.

Time 0 hours:

Experimental Value: 349

At the initial stage, the soil moisture content was measured at 349, setting the baseline for the experiment.

Time 2 hours:

Experimental Value: 283

After 2 hours, the soil moisture content decreased to 283, indicating a natural decline in moisture over the given time.

Time 4 hours (Watering Event):

Experimental Value: 622

The watering event at 4 hours led to a substantial increase in soil moisture to 622, demonstrating the system's capability to initiate timely watering actions.

Time 6 hours:

Experimental Value: 527

Following the watering event, the moisture content decreased to 527 at 6 hours, showcasing the system's responsiveness to maintain optimal moisture levels.

Time 8 hours:

Experimental Value: 461

The gradual decrease in soil moisture continued, reaching 461 at 8 hours.

Time 10 hours:

Experimental Value: 319

At 10 hours, the moisture content further reduced to 319, indicating sustained moisture depletion.

Time 12 hours:

Experimental Value: 260

A continued decline in soil moisture was observed, reaching 260 at 12 hours.

Time 14 hours (Watering Event):

Experimental Value: 617

Another watering event occurred at 14 hours, resulting in a significant increase in soil moisture to 617.

Key Observations:

The experimental results affirm the system's ability to respond dynamically to fluctuations in soil moisture, as evidenced by the watering events at 4 hours and 14 hours.

The observed moisture levels align with the expected behavior of the system, validating its practical effectiveness in maintaining optimal soil moisture content.

Conclusion:

The experimental data not only corroborates the numerical predictions but also provides a real-world validation of the automatic plant watering system. The system exhibited responsiveness and efficacy in managing soil moisture levels, confirming its potential for practical applications in automated plant care. The alignment between experimental and simulated values underscores the reliability of the system and sets the foundation for further improvements and optimizations.

C. Comparison between numerical and experimental result

The project's success is contingent upon a meticulous comparison between the numerical predictions and the actual experimental outcomes. This section delves into the assessment of the model's predictions against the real-world performance of the implemented system.

TABLE I
MOISTURE LEVEL AT DIFFERENT HOURS

Time (hour)	**Simulated value**	**Experimental value**
0	355	349
2	280	283
4	600(watering)	622(watering)
6	515	527
8	460	461
10	325	319
12	255	260
14	600(watering)	617(watering)

Comparison Analysis:

1. Accuracy and Deviation: The average absolute deviation provides an overall measure of the accuracy of the numerical predictions.

Average Absolute Deviation

$$= (|355 - 349| + |280 - 283| + |600 - 622| + |515 - 527| + |460 - 461| + |325 - 319| + |255 - 260| + |600 - 617|) / 8$$

$$= (6 + 3 + 22 + 12 + 1 + 6 + 5 + 17) / 8$$

= 9.5

Average Absolute Deviation = 9.5

The average absolute deviation is 9.5, indicating a relatively small overall difference between simulated and experimental values.

2. Response Time: The Response Time Deviation is the difference in time between when watering is simulated (according to the numerical model) and when watering is observed experimentally.

Response Time Deviation (4 hours):

$$4 - 4 = 0$$

Response Time Deviation (14 hours):

$$14 - 14 = 0$$

The system shows no deviation in response time between simulation and experiment. The system accurately predicts the timing of watering events.

Conclusion:

The comparison analysis suggests that the automatic plant watering system, based on the provided data, shows good accuracy and reliability in predicting watering events. Further investigation may be needed to understand the impact of environmental factors on the observed differences. The low average absolute deviation and consistent response time indicate a promising performance that can be further optimized for practical applications.

D. Cost analysis

Based on the current market prices of the components listed:

Arduino Uno: 850 BDT.

Moisture sensor module: 140 BDT.

5v relay module: 84 BDT.

5v dc pump with pipe: 200 BDT.

9v battery: 100 BDT.

Breadboard: 150 BDT.

2 Pushbutton: 2*20 BDT= 40 BDT.

2 LED: 2*10 BDT= 20 BDT.

Battery connector :10 BDT.

Jumper wires: 60 BDT.

Based on these costings, the total cost of the project is 1654 BDT.

E. Limitations in the project

The limitations of the "Automatic Plant Watering System" project are:

1. The Arduino Uno, while powerful for many simple automation tasks, has limited processing power and memory compared to more advanced micro-controllers or microcomputers.
2. The accuracy of soil moisture sensors and other environmental sensors may vary, affecting the system's ability to precisely determine watering needs.
3. Components exposed to outdoor conditions, such as sensors and electronic circuits, may face durability issues due to weather, wear, and tear.
4. The simplicity of the Arduino platform can limit the complexity of algorithms that can be implemented for water management, potentially affecting efficiency and decision-making accuracy.
5. The system might lack advanced features like machine learning or data analytics for optimizing water usage, which are often beyond the capabilities of the Arduino's processing power.
6. The system may not adequately account for different soil types, compositions, or specific plant water requirements.
7. Regular maintenance may be required to ensure the proper functioning of sensors and components. Additionally, the system might face reliability issues over time.
8. Scaling the system for larger gardens or commercial use may be challenging due to the limitations of the Arduino Uno.
9. For more extensive or complex applications, the cost of scaling up with multiple Arduino boards and sensors might not be economical compared to other solutions.

Conclusion and Future Endeavors

Conclusion:

In conclusion, the Automatic Plant Watering System using the Arduino Uno is a promising solution for plant enthusiasts and hobbyists seeking to simplify plant care and conserve water resources.

Although the system has some limitations, it has proven to be a successful and efficient solution for ensuring the consistent hydration of plants.

The system reduces the manual effort required to monitor and water plants regularly. By utilizing sensors to measure soil moisture levels, it can be ensured that plants receive water when needed, avoiding overwatering or underwatering. The system allows for easy customization of watering intervals and moisture threshold levels, making it adaptable to different plant species and environmental conditions. The user interface on the Arduino Uno makes it accessible and user-friendly, allowing users to monitor and adjust settings easily. The components used are affordable and readily available, making the system a cost-effective solution for plant enthusiasts.

Future Endeavors:

While the Automatic Plant Watering System using the Arduino Uno has proven to be effective, there is still room for improvement and expansion in the following ways:

1. Remote Monitoring and Control: Implementing IoT capabilities to enable remote monitoring and control via a smartphone app or a web interface, providing users with real-time updates and control over their plants from anywhere.

2. Data Logging and Analysis: Enhancing the system to log and analyze data over time, helping users better understand plant growth patterns and optimizing watering schedules.

3. Integration with Weather Data: Incorporating weather forecasts and data into the system to adjust watering schedules based on upcoming weather conditions, preventing unnecessary watering during rainy periods.

4. Advanced Sensor Technologies: Exploring more advanced soil moisture sensors or other sensors that can provide additional information for better plant care.

5. Energy Efficiency: Improving the system's energy efficiency by implementing power-saving features to reduce power consumption.

6. Multiple Plant Zones: Expanding the system to support multiple plant zones with individualized watering schedules and sensors for different types of plants.

7. Alerts and Notifications: Adding alerting mechanisms, such as SMS or email notifications, to inform users of critical conditions like extreme dryness or system malfunctions.

8. Solar Power: Incorporating solar panels or other renewable energy sources to power the system, reducing reliance on grid electricity.

REFERENCES