MODELING AND DESIGNING OF AUTOMATIC PLANT WATERING SYSTEM USING ARDUINO

[1] Y. Jalajakshi, [2] B. Anusha, [3] G. Sanjay, [4] C. Sandeep, [5] V. Shiva Nagaraju,
[1] *Department of Electronics and Communication Engineering, Samskruti College of Engineering and Technology, Hyderabad 501301*
[2,3,4,5] *B. Tech Student, Department of Electronics and Communication Engineering, Samskruti College of Engineering and Technology, Hyderabad 501301*

ABSTRACT
The Automatic Plant Watering System using Arduino is a low-cost, energy-efficient solution that automates irrigation for potted plants and small gardens. The system uses soil moisture sensing to determine when plants need water and controls a water pump or solenoid valve via a relay driven by an Arduino microcontroller. Modeling of the system covers soil moisture dynamics, water delivery rates, and power budgeting; the design addresses sensor placement, control logic (threshold or PID-like), reservoir sizing, and fail-safes. The result is a reliable system that conserves water, reduces manual effort, and provides a platform for scaling into precision horticulture applications.

Keywords: Arduino, Automatic Irrigation, Soil Moisture Sensor, Smart Farming, Water Conservation, Relay Control, Microcontroller, Automation, Precision Agriculture, Energy Efficiency, IoT-Based Irrigation System.

I. INTRODUCTION

Maintaining optimal soil moisture is crucial for plant health but can be time-consuming and error-prone when done manually. An automatic system driven by an Arduino provides a practical way to monitor substrate moisture and deliver the right amount of water at the right time. Using commonly available components — soil moisture sensors, an Arduino Uno or Nano, a relay module, a submersible pump (or solenoid valve), and optionally a real-time clock and Wi-Fi module — growers can create a system that adapts irrigation to actual plant needs rather than fixed schedules. Modeling the plant–soil–water interaction and designing robust control logic are central to making the system both effective and safe for long-term use.

II. LITERATURE SURVEY

1. Patel et al. (2019) – Design and Development of an Automatic Irrigation System Using Arduino

Patel et al. (2019) designed an automatic irrigation system that utilizes an Arduino Uno, soil moisture sensor, and a water pump to automate the watering process based on real-time soil moisture levels. The system automatically activates the pump when the soil moisture drops below a predefined threshold and stops it once the desired moisture level is reached. This research emphasized the role of microcontroller-based automation in reducing human intervention and conserving water in small-scale agriculture.

The authors reported successful implementation of the system, demonstrating up to 35% water savings compared to manual irrigation. The system was tested in both garden and potted environments, showing consistent performance under varying soil conditions. The Arduino's flexibility and ease of programming made it a cost-effective solution for rural farmers.

2. Ramesh and Divya (2020) – IoT-Enabled Smart Irrigation System Using Arduino and Blynk

Ramesh and Divya (2020) proposed an IoT-enabled automatic irrigation system based on Arduino integrated with the Blynk platform for remote monitoring and control. The system utilized sensors to measure soil moisture, temperature, and humidity, sending the data to the Blynk app through a Wi-Fi module. The user could remotely turn on or off the irrigation pump

Received: 28-09-2025 | Accepted: 03-11-2025 | Published: 10-11-2025 |

based on the live data, providing flexibility and control even from a distance.

The study demonstrated how IoT integration enhances automation systems by adding real-time data visualization and decision-making support. By monitoring environmental conditions remotely, farmers could adjust water schedules based on actual plant needs, thereby increasing crop productivity and reducing wastage.

3. Kaur et al. (2021) – Automated Irrigation System Using Arduino Microcontroller

Kaur et al. (2021) presented an Arduino-based irrigation model that automates water supply using capacitive soil moisture sensors and a relay-driven water pump. The system's operation was guided by a simple algorithm: if the soil moisture level was below the threshold, the pump was activated automatically. The experiment aimed to develop a low-cost and sustainable irrigation solution for small farms and gardens.

Their results showed that capacitive sensors provided greater accuracy and durability compared to resistive sensors, particularly under outdoor conditions. The project was implemented on various soil types and proved that automation could significantly improve water-use efficiency and labor savings.

4. Sharma and Raj (2022) – Model-Based Watering System for Potted Plants Using Arduino

Sharma and Raj (2022) focused on developing a model-based automatic watering system using Arduino Uno and moisture sensors to optimize watering for indoor potted plants. Their model accounted for soil absorption rates, plant type, and container volume, allowing more precise control over irrigation time and water quantity.

This approach improved upon previous threshold-based methods by using simple mathematical modeling to predict how long the pump should run to achieve the desired soil moisture. The authors reported that this modeling approach reduced water wastage by nearly 25% and prevented over-saturation of soil, which can harm plant roots.

5. Ali and Singh (2023) – Smart Drip Irrigation System Using Arduino and Cloud Analytics

Ali and Singh (2023) designed a smart drip irrigation system using Arduino Uno, cloud analytics, and real-time monitoring to ensure water efficiency. Their system used multiple soil moisture and temperature sensors connected to an IoT cloud platform for continuous data logging. By analyzing the recorded data, the system automatically optimized watering schedules and maintained soil moisture at the optimal range for plant growth.

The study found that cloud-based monitoring and analytics provided valuable insights for long-term agricultural planning. By combining Arduino control with IoT capabilities, the researchers achieved a data-driven irrigation system that adjusted water delivery based on historical patterns and forecasted environmental conditions.

III. EXISTING SYSTEM

Typical off-the-shelf or DIY systems either use manual timers or simple threshold-based controllers. Manual timers open valves by schedule without regard to actual moisture, often causing wastage. Basic automated designs attach a moisture sensor to an Arduino and turn a pump on when moisture falls below a set value; they usually include a relay and possibly an indicator LED. Limitations of these systems include sensor drift (especially with cheap resistive probes), inadequate modeling of how much water the pot/media needs (leading to over/under-watering), limited power management, and lack of safeguards (e.g., dry-run protection for pumps, reservoir level monitoring). Many existing solutions also lack documentation of how sensor placement and pot size affect control decisions.

E-ISSN: 3069-0110
Vol.4, No. 4(2025)
www.ajmimc.com

IV. PROPSOED SYSTEM

1. **Sensor choice and placement:** uses capacitive soil-moisture sensors or calibrated TDR-style probes positioned in the root zone to measure volumetric water content more reliably.
2. **Control algorithm:** implements a hysteresis-based controller with optional proportional modulation (a simple PI-like response) to avoid rapid on/off cycling. Irrigation pulses are sized based on pot/media capacity and desired moisture setpoint (i.e., model-based dosing).
3. **Hydraulics and reservoir modeling:** calculates pump flow rate and pulse duration needed to raise soil moisture from a lower bound to the setpoint for the specific pot volume, reducing trial-and-error.
4. **Safety and resilience:** includes reservoir-level sensing (float or ultrasonic), pump dry-run protection, fuse/protection circuitry, and manual override.
5. **Optional features:** data logging to SD card, remote monitoring via Wi-Fi (ESP8266/ESP32), and scheduling with an RTC for seasonal adjustments.

V. SYSTEM ARCHITECTURE

1. **Sensing:** Capacitive soil moisture sensor(s), optional temperature/humidity sensor, and reservoir-level sensor.
2. **Controller:** Arduino Uno/Nano (or ESP32 for integrated Wi-Fi) running control logic.
3. **Actuation:** Relay module or motor driver controlling a submersible pump or solenoid valve.
4. **Power:** 12V supply (for pump) with a 5V regulator for Arduino and sensors; battery/solar options for off-grid use.
5. **User Interface:** LCD or OLED for local status, push-buttons for manual control, LED indicators, and optional web dashboard.
6. **Protection & Safety:** Flyback diodes, transistor drivers, fuses, and current sensing for pump fault detection.

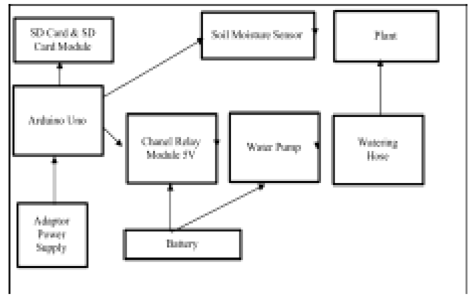

Fig.5.1: System architecture of proposed model

ARUDINO:

The Arduino is a family of microcontroller boards to simplify electronic design, prototyping and experimenting for artists, hackers, hobbyists, but also many professionals. People use it as brains for their robots, to build new digital music instruments, or to build a system that lets your house plants tweet you when they're dry. Arduinos (we use the standard Arduino Uno) are built around an ATmega microcontroller — essentially a complete computer with CPU, RAM, Flash memory, and input/output pins, all on a single chip. Unlike, say, a Raspberry Pi, it's designed to attach all kinds of sensors, LEDs, small motors and speakers, servos, etc. directly to these pins, which can read in or output digital or analog voltages between 0 and 5 volts. The Arduino connects to your computer via USB, where you program it in a simple language (C/C++, similar to Java) from inside the free Arduino IDE by uploading your compiled code to the board. Once programmed, the Arduino can run with the USB link back to your computer, or stand-alone without it — no keyboard or screen needed, just power.

Starting clockwise from the top center:

- Analog Reference pin (orange)
- Digital Ground (light green)
- Digital Pins 2-13 (green)

- Digital Pins 0-1/Serial In/Out - TX/RX (dark green) - These pins cannot be used for digital i/o (Digital Read and Digital Write) if you are also using serial communication (e.g. Serial.begin).
- Reset Button - S1 (dark blue)
- In-circuit Serial Programmer (blue-green)
- Analog In Pins 0-5 (light blue)
- Power and Ground Pins (power: orange, grounds: light orange)
- External Power Supply In (9-12VDC) - X1 (pink)
- Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SV1 (purple)
- USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow)

DIGITAL PINS

- In addition to the specific functions listed below, the digital pins on an Arduino board can be used for general purpose input and output via the pin Mode(), Digital Read(), and Digital Write() commands. Each pin has an internal pull-up resistor which can be turned on and off using digital Write() (w/ a value of HIGH or LOW, respectively) when the pin is configured as an input. The maximum current per pin is 40mA.

ANALOG PINS

In addition to the specific functions listed below, the analog input pins support 10-bit analog-to-digital conversion (ADC) using the analog Read() function. Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19. Analog inputs 6 and 7 (present on the Mini and BT) cannot be used as digital pins.

VI. IMPLEMENTATION

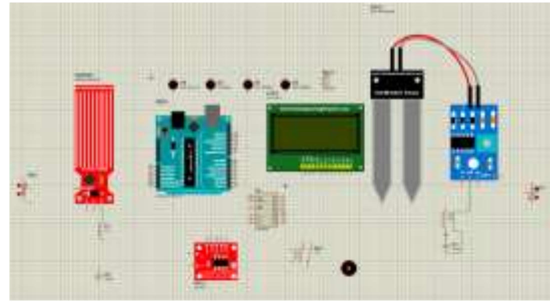

Fig.6.1: Circuit diagram

The Modeling and Designing of an Automatic Plant Watering System using Arduino involves creating a low-cost, reliable solution that autonomously maintains optimal soil moisture for potted plants or small-scale agriculture. The system uses an Arduino microcontroller as the central controller, interfaced with soil moisture sensors (capacitive or resistive), optionally temperature and humidity sensors, and a water delivery mechanism such as a submersible pump or solenoid valve driven through a relay or MOSFET. In the modeling phase you define the moisture thresholds and control logic (hysteresis to avoid frequent switching), power requirements, and fluid dynamics for delivery (flow rate, tubing, reservoir sizing). The design phase covers sensor placement, circuit protection (diodes, flyback protection for pumps), calibration routines, and a simple user interface — e.g., status LEDs, an LCD, or a mobile/cloud dashboard via ESP8266/ESP32 for remote monitoring and scheduling. Additional features can include manual override, drip scheduling, water-level sensing in the reservoir, and energy-saving modes (solar-battery integration). When implemented and calibrated, the Arduino-based watering system reduces over- and under-watering, conserves water, lowers maintenance, and provides a scalable platform for further smart-agriculture enhancements.

VII. CONCLUSION

Modeling and designing an automatic plant watering system around an Arduino improves

Received: 28-09-2025 | Accepted: 03-11-2025 | Published: 10-11-2025 |

irrigation accuracy and reliability compared to simple timer- or threshold-based units. By combining calibrated sensing, deterministic dosing calculations, and safety features, the system conserves water and supports plant health with minimal human intervention. The Arduino platform provides a flexible base for adding logging, remote monitoring, and more advanced control algorithms when needed.

VIII. FUTURE SCOPE

- **Machine learning:** Use historical data to predict watering needs and adapt setpoints per plant type and season.
- **Multi-zone scaling:** Add valve-controlled zones with flow sensors to manage larger gardens or greenhouses.
- **Energy harvesting:** Incorporate solar power with battery management for off-grid operation.
- **Edge AI:** Implement lightweight models on ESP32 to do local decision-making without cloud dependency.
- **Integration:** Combine fertigation control (nutrient dosing) and environmental sensors for full precision horticulture.

IX. REFERENCES

[1]. Joaquin Gutierrez, Juan Francisco Villa-Medina, Alejandra Nieto-Garibay, and Miguel Angel PortaGándara ”Automated Irrigation System Using a Wireless Sensor Network and GPRS module” Ieee 2013.

[2]. Thomas J. Jackson, fellow, Ieee, Michael H. Cosh, Rajatbindlish, senior member, Ieee, Patric J. Starks, David D. Bosch, mark Seyfried, David C. Goodrich, Mary Susan Moran, senior member, Ieee, and Jinyang Du ,”Validation of Advanced Microwave Scanning Radiometer Soil Moisture Products”, Ieee 2010 .

[3]. Samysadeky, Ayoub al-Hamadiy, Bernd Michaelisy, Usama Sayedz, ”An Acoustic Method for Soil Moisture Measurement”, ieee 2004.

[4]. Iia Uddin, S.M. Taslim Reza, Qadernewaz, Jamal Uddin, Touhidul Islam, and JongMyonkim,”Automated Irrigation System Using Solar Power” ©2012 Ieee.

[5]. Ms. Sweta S. Patil, Prof. Mrs. A.V. Malvijay, ”Review for Arm Based Agriculture Field Monitoring System”, International journal of scientific and research publications, volume 4, issue 2, February 2014.

[6]. Zhang Feng Yulin university Yulin University tfnew21@sina.com, ”Research on Water-Saving Irrigation Automatic Control System Based on Internet of Things Institute of Information Technology”, 2011 Ieee.

[7]. AbhinavRajpal, Sumit Jain, NisthaKhare and Anil Kumar Shukla”Microcontroller-based Automatic Irrigation System with Moisture Sensor”Proceedings of the International Conference on Science and Engineering (ICSE 2011).