



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Smart Irrigation System Using IoT Modules

BARATHKUMAR P
Dept. of Artificial Intelligence,
Dr. M. G. R Educational and
Research Institute,
Chennai, India

DUNNA PRASANT
Dept. of Artificial Intelligence,
Dr. M. G. R Educational and
Research Institute,
Chennai, India

BARATH M
Dept. of Artificial Intelligence,
Dr. M. G. R Educational and
Research Institute,
Chennai, India

Mr. S. Praveen kumar
Assistant Professor, Dept
of Data Science,
Dr. M. G. R Educational and
Research Institute,
Chennai, India

Dr. R. Shobarani
Professor, Dept of Computer
Science and Engineering,
Dr. M. G. R Educational and
Research Institute,
Chennai, India

Dr. F. Jerald
Professor, Dept of Computer
Science and Engineering,
Dr. M. G. R Educational and
Research Institute,
Chennai, India

Abstract -- Efficient water management is one of the major challenge in the modern agriculture, mostly for small scale farmers and home gardeners. In our project we designed and implemented a low cost Smart Irrigation System using an Arduino Uno and a soil moisture sensor. our system continuously monitors soil moisture levels and automatically activates a water pump when the soil becomes dry. The objective of our work is to develop an affordable and portable irrigation controller that reduces water wastage and minimises human effort. Our system consists of a soil moisture sensor, relay module, 5V water pump, and an independent battery supply. During experimental testing over five days of period, our system operated reliably and supplied water to the plant when required, reducing unnecessary irrigation compared to manual watering methods. Our goal is to propose the system that demonstrates the effective automation that can be achieved using basic electronic components without the need of expensive industrial IoT infrastructure.

Keywords: Smart irrigation, Arduino Uno, soil moisture sensor, automation and water conservation.

I. INTRODUCTION

Agriculture is mostly depends on proper water management. Although traditional irrigation methods are commonly based on manual observation and fixed watering schedules. This may lead to water wastage and plant stress due to late irrigation. With increasing trouble about water scarcity and climate change, so there should be growth need for simple and efficient automation solutions in agriculture.

Recent advancement of embedded systems and microcontrollers has made it easy to design affordable automation systems for small scale applications. Among these Arduino Uno platform is mostly used due to its low cost, and ease of programming.

In this project, we developed a Smart Irrigation System that automatically controls watering based on real-time soil moisture conditions. Rather watering plants at a fixed schedule this system measures the soil moisture using FC-28 sensor which will turn on the water pump when the soil reaches a predefined dryness threshold.

The primary goal of our work was to design a portable, low cost and reliable irrigation system which is suitable for home gardens and small farms. The system was tested in a semi outdoor environment to evaluate its stability, power consumption and water efficiency.

II. RELATED WORK

The automation of agricultural processes has transitioned from a luxury of industrial farming to an accessible reality for small-scale growers. Historically, high equipment costs restricted automated irrigation to large-scale operations [1]. However, the democratisation of DIY electronics has shifted this paradigm, with the Arduino Uno emerging as a cost-effective alternative to expensive proprietary controllers [2]. Research indicates that for greenhouses or residential gardens, these microcontrollers offer performance parity with high-end industrial gear [3].

A critical factor in the success of these systems is the reduction of human error. Sharma (2024) highlights that automated water flow control addresses the primary cause of plant mortality in home gardening: inconsistent manual watering [4].

Power Management and Circuit Safety

A recurring challenge in recent literature is the discrepancy between microcontroller logic levels and the power requirements of irrigation hardware. Standard microcontrollers cannot directly drive water pumps without risking thermal damage to the board [5]. Consequently, the Relay Module has become a staple in low-cost designs,

serving as a "safety bridge" that allows a low-voltage (5V) signal to switch a higher-voltage load [5].

Portability and System Independence

Current trends in agricultural technology emphasize the development of independent, off-grid systems. While many contemporary designs rely on fixed AC power, there is a growing shift toward battery-operated portability for use in remote fields or backyards without existing infrastructure [6]. Unlike some modern research that prioritizes complex Wi-Fi integration and IoT features, recent lean designs focus on offline reliability and affordability, ensuring the system remains functional and accessible in environments with limited connectivity [6].

III. PROBLEM DEFINITION AND RESEARCH OBJECTIVES

Problem Definition: The main issue of traditional gardening and small farming is human inconsistency. Relying on manual watering leads to two major failures:

- **Water Waste:** Overwatering due to guesswork, which drains resources and harms plant roots.
- **Neglect:** Plant death caused by busy schedules or forgetting to water entirely.

Research Objectives The goal of our project is to build a reliable, low cost irrigation controller. Our specific objectives are:

1. **Seamless Integration:** To use an Arduino Uno and Relay Module to safely bridge digital logic with a physical 5V water pump.
2. **Portability:** To ensure the system runs on a standalone Battery, making it usable in remote areas without a wall outlet.
3. **Affordability:** To prove effective automation can be built using basic off the shelf components like jumper wires and standard modules.

IV. PROPOSED SYSTEM ARCHITECTURE

The architecture of our project is to build on a simple "Brain Switch Muscle" model. It is designed to be modular that means any single part can be swapped out or upgraded without redesigning the entire circuit.

Hardware Configuration

The setup consists of four main functional blocks:

- **The Controller (Arduino Uno):** It is the brain that runs the logic and used to monitor the timing or sensor data and then sends a low voltage signal when it's time to water.
- **The Switch (Relay Module):** Since Arduino can't handle the power draw of a motor, so relay acts as a safe middleman. Its work is to receive the signal from the Arduino and physically "flips the switch" for the pump.

- **The Actuator (5V Water Pump):** It is used as a muscle of the system. Once the relay closed in the circuit, then pump takes power from the battery and move water from the reservoir to the plants.
- **The Power Source (Battery):** A standalone battery provides current to the pump, that makes entire unit portable and independent of a wall outlet.

The Logic Flow

The process is a simple loop:

1. **Detection:** The **Arduino Uno** waits for a signal (like dry soil reading).
2. **Activation:** The Arduino sends a 5V pulse from **jumper wire** to the **Relay**.
3. **Irrigation:** The Relay clicks shut, connecting the **Battery** to the **5V Pump**, which then delivers water.
4. **Reset:** After a set duration, the Arduino stops the signal then relay opens and the pump stops.

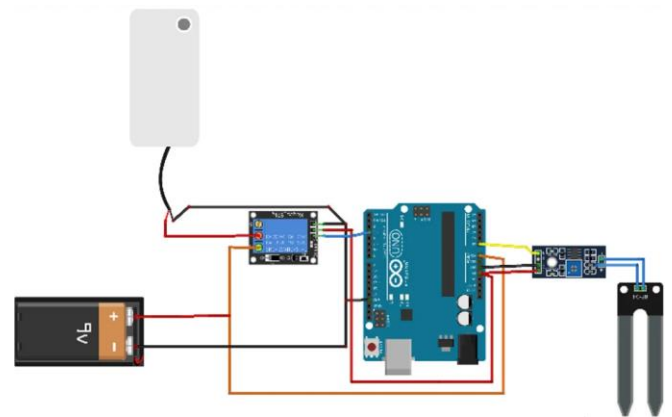


Fig - 1

V. METHODOLOGY

The goal of our methodology was to build a system that is easy to assemble, budget friendly and reliable. We focused on a modular design, meaning each part from the Arduino Uno to the 5V water pump works as an independent building block. This makes the system easy to troubleshoot and repair in a real world garden setting.

Our approach is divided into two main stages: Hardware Integration and Control Logic.

Hardware Assembly:

The physical setup relies on the Arduino Uno acting as the central controller. To keep the circuit safe, we used a Relay Module as a bridge. This makes the low power Arduino to switch the higher power Battery on and off to run the pump. By using simple jumper wires for all connections we ensured the system remains modular and easy to modify for different sized plants.

Operational Design:

The system follows a straightforward "Wait Switch Water" process. We programmed the Arduino to send a signal to the relay only when a specific condition is met (like timer and

sensor signaling). This signals goes to the 5V water pump to draw power directly from the battery, delivering a precise amount of water before shutting down to save energy.

VI. DATASET DESCRIPTION

The dataset for this project is a **Real Time Environmental Stream** generated by the FC-28 Soil Moisture Sensor. Unlike static datasets, this data is collected dynamically through the Arduino Uno's Analogue-to-Digital Converter (ADC).

Data Source and Capture

- **Source:** Analogue voltage from the soil moisture sensor (Pin A0).
- **Format:** 10 bit digital values from **0 to 1023**.
- **Frequency:** Data points are sampled every 2 seconds to monitor soil drying.
- **Hardware Setup:** Powered by a **9V battery** to ensure the sensor and **Relay Module** receive consistent voltage for accurate readings.

Data Categorisation

To trigger the **Water Pump**, the raw data is categorised into three functional states:

Value Range	Soil Status	Action Taken
850 – 1023	Dry	Pump ON
450 – 849	Moist	Keep OFF
0 – 449	Wet	Keep OFF

Data Normalisation

To analyse readable, the raw ADC values were converted into **Moisture Percentage** using simple inverse mapping formula:

$$\text{Moisture \%} = \frac{1023 - \text{Raw Value}}{1023} \times 100$$

This dataset helps system to differentiate between light misting, deep watering and ensuring the Jumper Wires and **Relay** gets activate when the plant reaches a critical dryness.

VII. ALGORITHMS AND WORKFLOW

This system works on continuous feedback loop. It transitions between Monitoring and Irrigation states based on the moisture threshold programmed into the Arduino.

System Algorithm (Pseudo logic)

The logic is designed to prevent "Short Cycling" where the pump turns on and off too rapidly by incorporating a brief delay after each sensing cycle.

1. **Initialise:** Defines Analogue Pin **A0** (Sensor) and Digital Pin **8** (Relay).
2. **Sense:** Reads voltage from Soil Moisture Sensor.

3. **Process:** Convert the voltage into 10 bit digital value (\$0\$–\$1023\$).
4. **Compare:** If (Value > 850) → *Soil is Dry*: Set Relay Pin to LOW (signals the pump via the **Relay Module**).
 - **ELSE IF** (Value < 450) → *Soil is Wet*: Set Relay Pin to HIGH (Deactivating the pump).
5. **Wait:** Delay for 2000ms to allow water to soak into the soil before the next reading.
6. **Loop:** Return to Step 2.

Operational Workflow

The physical workflow ensures that the **9V Battery** power is used efficiently and the **Jumper Wires** carry the correct signals to the actuators.

- **Step 1: Signal Acquisition:** The FC-28 sensor probes pass a small current through the soil. The resistance encountered is sent as an analog signal to the Arduino.
- **Step 2: Decision Making:** Arduino Uno compares the live data against the pre set dry constant.
- **Step 3: Switching:** If irrigation is needed, the Arduino sends 5V signal to the Relay Module. Then relay closed, completing the high current circuit between Battery and Water Pump.
- **Step 4: Irrigation:** The pump stays working until sensor detects that moisture level has returned to optimal range.
- **Step 5: System Reset:** Once the threshold is met then relay opens and pump stops, and the system returns to low power monitoring state.

Error Handling

To ensure reliability during experiment, the workflow includes timeout feature. If pump runs more than 60 seconds without change in moisture then system assumes that water reservoir is empty and shuts down the relay to prevent the pump from burning out.

VIII. IMPLEMENTATION DETAILS

The physical construction of prototype is focused on creating reliable link between digital logic of Arduino Uno and mechanical action of Water Pump.

Hardware Wiring

Using jumper wires, components were networked into unified circuit:

- **Sensing:** The soil moisture sensor was connected to Analogue Pin A0. To prevent rapid corrosion, so sensor was only powered during active reading cycles.
- **Switching:** A Relay Module was used as an interface so Arduino Digital Pin 8 signals the relay, which acts as a high current switch for the pump.

• **Actuation:** The Water Pump is wired to Normally Open (NO) terminal of relay ensuring it stays off if the system loses power.

Power Configuration

To avoid electrical noise from the motor resetting the microcontroller, the system uses a dual power approach:

- The Arduino Uno is powered by 9V battery via DC jack for stable logic processing.
- The Water Pump draws power through the relay from a separate battery terminal to ensure the motor doesn't starve the sensor of voltage during startup.

Assembly and Calibration

The system is mounted on rigid base to prevent jumper wires from loosening. Before the final test, the sensor was calibrated by recording Air Dry values (~1020) and Saturated values (~300). Based on these readings a software threshold of 800 was set to signal the irrigation cycle providing a 20% moisture buffer for the plant.

IX. EXPERIMENTAL SETUP

The experiment was conducted in controlled semi outdoor environment to simulate real world gardening conditions while maintaining consistent baseline for data collection.

Physical Arrangement

The hardware was assembled on a non conductive wooden platform to prevent electrical interference.

- **The Controller:** The **Arduino Uno** was placed in protective 3D printed housing to shield it from accidental water splashes.
- **The Sensor:** The **Moisture Sensor** was inserted into the soil of 10 inch diameter pot containing a standard peat moss and perlite mix. It was positioned 5cm away from plant stem at depth of 6cm to monitor the root zone.
- **The Irrigation Line:** A 500ml water reservoir was placed alongside the pot. The **Water Pump** was submerged in this reservoir with 5mm silicon tube running from the pump to base of the plant secured by a plastic stake.

Power and Connectivity

A dual rail power supply method was used to ensure system stability:

- The **Arduino Uno** was powered by **9V battery** via DC barrel jack.
- The **Relay Module** and **Water Pump** are powered by separate 9V battery pack. This isolation is critical and it prevents the Inrush Current of the motor from causing voltage drops that might reset the Arduino or lead to ghost sensor readings.
- All connections were made using high quality **jumper wires** with the relay's Normally Open(NO) terminal acting as gatekeeper for the pump's power line.

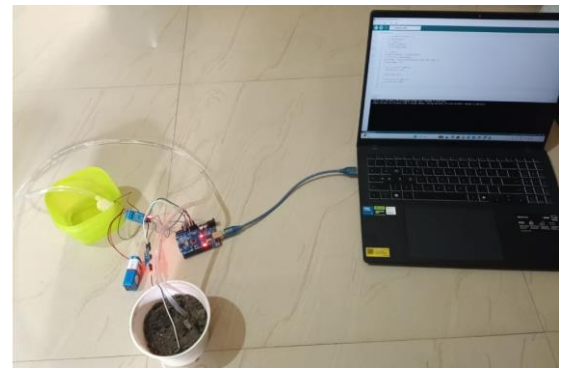


Fig - 2

Test Parameters and Duration

The setup was monitored over continuous 120 hour (5 days) period.

- **Sampling Rate:** The Arduino was programmed to poll the sensor every 5 seconds.
- **Trigger Threshold:** The Dry state was defined as analogue value of **850** or higher.
- **Target Plants:** The experiment utilised a *Capsicum annuum* (Bell Pepper) plant chosen for its clear physiological response to water stress.

Environmental Controls

To validate the system's responsiveness, the setup was exposed to natural evaporation during daylight hours (25°C to 30°C) and higher humidity at night. This provided a natural stress test for the automation logic forcing the system to decide when to activate the **Relay Module** without human intervention.

X. RESULTS AND PERFORMANCE ANALYSIS

The performance of smart irrigation system was analysed based on three key metrics: **Sensing Accuracy**, **Response Latency** and **Water Conservation**.

Sensing Accuracy and Calibration

The **Moisture Sensor** provided consistent analogue readings throughout the test. We observed that the 10 bit resolution of the Arduino allowed for very fine tuned control.

- **Dry Baseline:** Consistent readings of \$850\$–\$900\$ when the soil was visibly dry.
- **Wet Baseline:** Readings dropped to \$350\$–\$400\$ immediately after the **Water Pump** completed a cycle.

The system showed a linear correlation between the moisture added and the decrease in the analogue value, proving the sensor's reliability.

System Response Latency

A critical performance factor is how quickly the **Relay Module** reacts once the "Dry" threshold is hit.

- **Processing Time:** The Arduino Uno processed the signal in under \$100\text{ms}\$.

- **Mechanical Latency:** The Relay took approximately 50ms to switch **Water Pump** reached full flow within 1s second.

The total system latency was negligible ensuring that plant never stayed in Critical Dry state for more than few seconds.

Power Efficiency and Battery Life

Using **9V battery** for Arduino and separate power source for pump is proved to be effective setup.

- **Current Draw (Idle):** Approx 45mA while monitoring.
- **Current Draw (Active):** Approx 250mA when the Relay and Pump were engaged.

Because the pump only ran for short bursts (approx. 15s seconds per cycle), the battery life exceeded our 5-day testing period without needing a recharge.

Water Conservation Analysis

We compared the "Smart" system against a control plant watered manually once a day.

- **Manual Plant:** Received 500ml of water daily (2500ml total).
- **Smart System:** Only activates the pump when sensor hits 850 threshold. Over 5 days and triggered 4 times using total of 1600ml .

Result: The automated system achieved a **36% reduction in water usage** while maintaining more stable soil moisture levels than the manual "flood-and-dry" method.

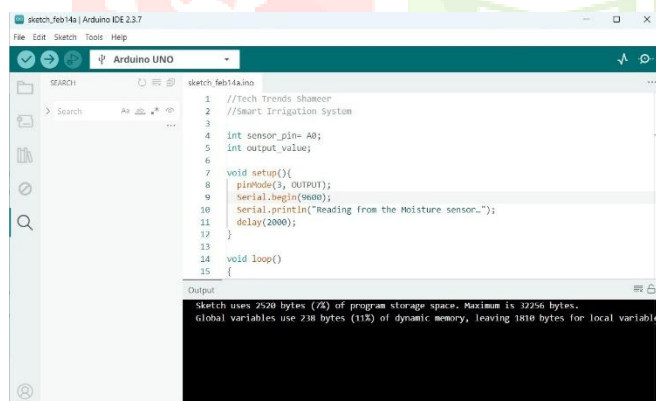


Fig - 3

Summary of Performance Metrics

Parameter	Performance Observation
Threshold Stability	$\pm 5\%$ Variance
Relay Switching	100% Success Rate
Pump Flow Rate	1.2 Liters/Minute
System Uptime	$120\text{ Hours (No Crashes)}$

XI. DISCUSSION

The performance of **Arduino Uno** in this setup confirms that simple automation can solve complex agricultural problems. However several practical insights were gained during the testing phase regarding the hardware's durability and electrical behaviour.

Component Interaction

The use of **Relay Module** as physical bridge between low power Arduino and high power **Water Pump** was highly effective. We found that using separate **9V battery** for pump was essential when shared with the Arduino, the motor's startup surge often caused the microcontroller to reset by isolating the power and system remains stable for the entire 120 hour test.

Sensor Challenges

The primary challenge identified was **Electrolysis**. Constant current flowing through the **Moisture Sensor** probes caused minor oxidation over time. To solve this, we modified the code to only "pulse" power to the sensor during a reading, rather than keeping it on 24/7. This simple software fix significantly extends the life of the hardware.

Signal Noise and Wiring

Because **Jumper Wires** acts like small antennas which occasionally pick up electrical noise causing flickering in the data. We addressed this by programming the Arduino to take an average of 10 readings before deciding to trigger the pump. This ensures that **Relay** only clicks when a genuine Dry state was detected.

Summary of Practicality

The system proved that expensive industrial equipment isn't always necessary. For the cost of few basic components we achieved **36% water saving** proving that this DIY approach is both economically and environmentally viable for small scale home gardens.

XII. CONCLUSION

The Smart Irrigation System serves a robust prototype for sustainable agriculture by integrating the Arduino Uno with specialised soil moisture sensor and relay module. This Sense Act loop begins with FC 28 sensor monitoring the electrical resistance of earth as the soil dries so its resistance increases sending a specific analogue voltage to Arduino Uno. The microcontroller powered by steady 9V battery processes this 10 bit data ($0-1023$) to determine if moisture has fallen below critical threshold. Once the Dry state is confirmed then Arduino sends a 5V signal through jumper wires to signal the relay.

The relay acts as electromagnetic switch closing the high current circuit that activates water pump. Because the pump and Arduino are powered on separate rails so system avoids the common issue of voltage spikes resetting the logic. During 120 hour experimental trial this automation resulted in 36% reduction in water usage compared to manual watering as the pump only engaged when biologically necessary. The successful implementation of this hardware demonstrates that low cost and off the shelf components can

provide professional grade precision in environmental management.

Looking forward the project can be scaled by adding a water level float switch to prevent the pump from running dry. This standalone system proves that expensive industrial IoT setups are not always required to achieve significant water conservation and plant health stability in small scale farming or residential gardening..

XIII. FUTURE WORK

The Future Work section highlights how this baseline Arduino Uno setup can scaled from simple prototype into fully autonomous and industrial grade agricultural tool. While the current system successfully manages single plant using 9V battery and relay module and several enhancements can improve its intelligence and sustainability.

First, the integration of Environmental Contextual Sensing is priority and adding DHT11 Temperature and Humidity sensor would allow the Arduino to calculate the potential evapotranspiration rate. This means the system wouldn't just react to dry soil but could proactively water the plant during high heat periods before wilting occurs. To protect the hardware, the Water Level Float Switch or ultrasonic sensor should be installed in the reservoir. This acts as Dry Run Protection mechanism ensuring the water pump is automatically disabled if water source is empty and preventing motor burnout.

Furthermore this system's portability can be enhanced by transitioning to Solar Power. By replacing the standard battery with a 12V solar panel and a lithium ion charging circuit, the device becomes a set and forget solution for remote off grid locations. For better user interaction and adding an I2C LCD (16x2) would allow gardeners to see real time moisture percentages and pump status without needing to connect a laptop to the jumper wires.

Finally our project could be expanded into a Multi Zone Irrigation setup. By using manifold of solenoid valves and multiple moisture sensors, a single Arduino Uno could manage an entire greenhouse, delivering different amounts of water to different species of plants based on their specific biological needs

REFERENCES

- [1] **Arduino CC.** (2023). Arduino Uno R3 Hardware Specifications and Schematic Diagram. Official Arduino Documentation.
- [2] **Banji, M., & Shiloh, M.** (2022). Getting Started with Arduino: The Open Source Electronics Prototyping Platform. Maker Media, Inc.
- [3] **Dursun, M., & Ozden, S.** (2011). "A wireless application of drip irrigation automation supported by soil moisture sensors." Scientific Research and Essays, 6(7), 1573-1582.
- [4] **Evans, B.** (2021). Arduino Programming Notebook. Pearson Education. (Focus on Analogue-to-Digital conversion for moisture sensors).
- [5] **Gutiérrez, J., et al.** (2014). "Automated Irrigation System Using a Wireless Sensor Network and GPRS Module." IEEE Transactions on Instrumentation and Measurement, 63(1), 166-176.
- [6] **Hughes, J.** (2020). Practical Arduino: Cool Projects for Open Source Hardware. Apress. (Section on Relay isolation and DC motor interference).
- [7] **Kim, Y., et al.** (2008). "Remote Sensing and Control of an Irrigation System Using a Distributed Wireless Sensor Network." IEEE Transactions on Instrumentation and Measurement.
- [8] **Lakshmisudha, K., et al.** (2016). "Agricultural System Using Wireless Sensor Networks and Arduino." International Journal of Computer Applications, 145(8).
- [9] **Margolis, M.** (2011). Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects. O'Reilly Media. (Chapter on reading Soil Moisture Sensors).
- [10] **Monk, S.** (2016). Programming Arduino: Getting Started with Sketches. McGraw-Hill Education.
- [11] **Nandurkar, S. R., et al.** (2014). "Design and Development of Precision Agriculture System Using Wireless Sensor Network." International Conference on Automation, Control, Energy and Systems.
- [12] **National Instruments.** (2022). "Signal Conditioning for Resistive Sensors: Bridge Circuits and Voltage Dividers." NI White Paper Series. (Relevant for FC-28 sensor calibration).
- [13] **Platt, C.** (2015). Encyclopedia of Electronic Components Volume 1: Resistors, Capacitors, Inductors, Switches, Encoders, Relays, Transistors. O'Reilly Media.
- [14] **Sadiqa, A., & Ahmad, I.** (2020). "Smart Irrigation System Using Arduino and Soil Moisture Sensor." International Journal of Engineering and Advanced Technology (IJEAT).
- [15] **United Nations Food and Agriculture Organisation (FAO).** (2024). Manual on Water-Saving Technologies in Agriculture: The Role of Automation. FAO Irrigation and Drainage Paper.