International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Autonomous Irrigation Controller Leveraging Soil Hydrology Analytics

[1]Ahil Nirranjan D , [1] Arun E , [1] Logeshwaran S, [2] Keerthana G

[1]Student,[2] Assistant Professor
Department of Electronics and Communication Engineering
K.Ramakrishnan College of Technology, Trichy,Tamilnadu, India.

Abstract : Watering based on fixed schedules or visual inspection may waste water, create fluctuating root-zone moisture, and fail to protect plants when users are away. This paper presents AquaPulse, a proposed autonomous irrigation controller that combines calibrated soil-moisture sensing, temperature monitoring, reservoir-level detection, water-flow verification, embedded control, and Internet of Things (IoT) telemetry. The design interprets moisture readings in the context of substrate calibration rather than treating raw sensor percentages as universal indicators of plant water availability. A local microcontroller applies filtering, persistence checks, hysteresis, soak intervals, maximum-run limits, and fault interlocks to determine when watering may begin and when it must stop. Reservoir and flow checks help prevent dry running and identify blocked tubing or unsuccessful delivery. A mobile dashboard provides current status, historical trends, alerts, and time-limited manual control, while essential irrigation logic continues during network outages. The paper details the system architecture, hardware interfaces, software state machine, data model, calibration procedure, and an experimental protocol for comparison with fixed-timer and single-threshold baselines. Evaluation metrics include delivered water volume, moisture-band compliance, pump runtime, energy use, false triggers, and fault-detection performance. An optional short-horizon moisture estimator is described as an advisory feature that cannot override safety rules. No prototype measurements or field dataset were supplied; therefore, the manuscript does not claim measured water savings, improved plant growth, or prediction accuracy. The contribution is a reproducible engineering design and validation framework for small-scale smart irrigation, suitable for further implementation and controlled

Key Terms : autonomous irrigation, soil moisture, soil hydrology, Internet of Things, precision agriculture, water-flow sensing, embedded control.

INTRODUCTION

Water is essential for plant growth, but irrigation is beneficial only when its timing and amount match the plant's demand and the storage characteristics of the root zone. A fixed timer applies water according to a schedule even when the substrate is already wet, while manual watering depends on the user's availability and judgement. Both approaches can lead to overwatering, underwatering, nutrient leaching, and inconsistent plant conditions. These limitations are especially relevant to indoor planters, home gardens, nurseries, and small agricultural plots where the operator may not be present throughout the day. Low-cost sensors and microcontrollers have enabled irrigation systems that observe soil conditions and switch pumps or valves automatically. Internet connectivity adds remote monitoring and notifications, but connectivity alone does not guarantee a reliable irrigation decision. A robust design therefore combines agronomic interpretation, local control, and fault-aware actuation.AquaPulse is proposed as a compact irrigation appliance with a water reservoir, a planter moisture probe, temperature sensing, a pump or valve, a flow sensor, a tank-level sensor, and a phone dashboard. Its purpose is not simply to automate a timer; it is to make each irrigation event conditional on measured substrate moisture and system safety. The controller is intended to operate locally so that loss of Wi-Fi does not stop the core watering logic.

RELATED WORK

IoT irrigation research typically combines environmental sensors, a microcontroller, a communication link, and an actuator. The literature includes domestic smart-irrigation systems that support automatic and user-assisted watering, as well as precision-agriculture systems that transmit moisture data and actuate irrigation equipment [1], [2]. These designs demonstrate the practicality of sensor-based irrigation. However, a controller's performance depends on calibration, sensor placement, water-distribution characteristics, and the crop or plant profile used to set its target range. Predictive approaches attempt to estimate future soil moisture from time-series measurements and environmental context. An ARIMA-based IoT watering approach has been described for moisture prediction in smart agriculture [3]. Forecasting may help anticipate a moisture deficit, but a model can be unreliable when data are sparse, sensors drift, or conditions change. Consequently, AquaPulse treats prediction as an optional advisory layer. It cannot bypass tank-level checks, maximum pump duration, or other safety interlocks.Soil-water physics provides a further reason to calibrate moisture sensors. Soil-water retention describes the relationship between water content and the energy with which water is held in a porous medium. Van Genuchten's model is a widely cited representation of the soil-water retention curve and related hydraulic behaviour [4]. The proposed controller does not need to solve a full hydraulic model at every sampling interval, but the principle is important: equal raw readings from different soils or potting mixes may not correspond to equal plant-available water.

The main design distinction in AquaPulse is the combination of moisture-based control with delivery verification and bounded operation. A sensor-only system may report that the soil is dry and switch on a pump, yet the tube could be blocked or the reservoir empty. Adding flow and level checks enables the system to identify that the requested action did not occur as expected. This improves diagnosability even when it does not directly improve the moisture estimate.

REQUIREMENTS AND DESIGN OBJECTIVES

The system is designed around functional, safety, usability, and maintainability requirements. Functional requirements define the observations and actions needed for automatic watering. Safety requirements limit operation under invalid or hazardous conditions. Usability requirements make the system state understandable to a non-specialist. Maintainability requirements support calibration, replacement of sensors, and investigation of faults.

A. Functional requirements

- Measure substrate moisture at a repeatable depth and location, and retain both raw and calibrated readings.
- Measure temperature as contextual information; do not use temperature alone as a reason to water.
- Detect a low reservoir level before the pump runs and stop operation if the level becomes unsafe.
- Verify water delivery using a flow switch or calibrated flow meter during each watering event.
- Apply a lower moisture threshold to start watering and a separate upper threshold to stop it.
- Display current measurements, pump status, tank status, recent irrigation events, and fault messages.
- Support time-limited manual watering without allowing users to bypass critical safety interlocks.

B. Non-functional requirements

The local controller should continue its basic control loop during internet or cloud outages. Firmware should default the pump to OFF after reset or uncertain state, impose a maximum run duration, and log a reason for every interrupted cycle. Sensor and actuator connections should be serviceable, and the enclosure should protect electronics from splashes. The interface should use clear units and timestamps, while remote commands should be authenticated and acknowledged by the device.

C. Design assumptions

The baseline application is a single pot or small planting zone with a low-voltage pump and a finite reservoir. Larger installations may use valves and a pressurized supply, but would require additional hydraulic design. The exact moisture thresholds, sample interval, soak time, and dose limit depend on the selected substrate, plant, container geometry, and pump capacity; these parameters must be determined through calibration and testing rather than assumed to be universal.

SYSTEM ARCHITECTURE

AquaPulse is divided into four cooperating layers: sensing, local decision and actuation, communication and data management, and the user interface. The sensing layer converts environmental and hydraulic conditions into electrical signals. The local controller validates and filters those signals, evaluates the state machine, and drives the pump. The communication layer sends telemetry when a network is available. The interface allows observation and configuration but is not part of the safety-critical feedback loop.

Subsystem	Proposed implementation	Role and design notes
Controller	ESP32-class MCU or equivalent	Samples sensors, runs state machine, maintains local settings and event log.
Moisture probe	Capacitive sensor	Measures substrate response; calibrate for each substrate and installation depth.
Temperature sensor	Digital sensor	Provides contextual temperature readings and supports alerts.
Tank sensor	Float switch or level sensor	Inhibits pump when reservoir is too low.
Flow sensor	Hall-effect meter or flow switch	Checks that water is moving; meter can estimate delivered volume after calibration.
Actuator	DC pump or solenoid valve	Applies water under bounded timing and volume rules.
Driver stage	MOSFET or suitable relay with protection	Switches actuator current; must tolerate startup current and inductive transients.
Connectivity	Wi-Fi and secure API/MQTT option	Transmits status and receives permitted settings or commands.
Dashboard	Mobile or web interface	Shows readings, trends, alerts, and manual controls.

A typical control cycle begins by sampling the moisture probe and other sensors. The firmware checks whether the readings are valid, applies filtering, and updates the state machine. If the moisture deficit is persistent and the system is safe, the pump starts. The flow sensor must register expected flow within a defined timeout. While watering, the controller checks moisture, runtime, delivered volume, and tank level. When a stop condition occurs, the pump is switched off and the controller enters a soak period. After the soak delay, the substrate is sampled again because water may need time to spread through the root zone. The physical layout matters. The moisture probe should be placed in the active root zone but not directly beside the water outlet, where a local

wet spot could produce an unrepresentative reading. The flow sensor should be installed in the correct direction and within its rated flow range.

SOIL HYDROLOGY AND SENSOR CALIBRATION

A soil-moisture sensor does not directly measure plant health. Most inexpensive probes report an electrical quantity related to the dielectric properties or conductivity of the surrounding medium. The relationship between sensor output and water content is affected by soil texture, organic matter, salinity, temperature, packing density, probe geometry, and contact with the substrate. A raw reading such as "62%" should therefore be treated as a device-specific scale unless it has been calibrated against a defined moisture measurement.

A. Gravimetric calibration

A practical calibration can be performed using representative samples of the intended potting mix. Record the mass of a dry sample, add a measured amount of water, allow moisture to equilibrate, and record the sensor output and wet mass. Repeat at several moisture states from relatively dry to near the normal upper operating range. The gravimetric water content is calculated as:

$w = (m_wet - m_dry) / m_dry$, where m_wet is the wet sample mass and m_dry is the dry sample mass. If dry bulk density is known, volumetric water content can be approximated as $\theta v = w\rho b/\rho w$, where ρb is dry bulk density and ρw is water density. For mixed commercial substrates, the result is an operational calibration and should not be interpreted as a universal soil-water characteristic curve.

The calibration curve can be implemented as a lookup table or a fitted linear or nonlinear mapping. A lookup table is often preferable when the probe response is nonlinear because it avoids assuming a simple linear relationship. Calibration points should cover the range where the controller is expected to operate. Repeating selected points helps estimate sensor repeatability. The probe should be rechecked after moving it, changing the substrate, or observing a persistent discrepancy between sensor readings and visible substrate condition.

B. Threshold selection and hysteresis

Let θ represent calibrated moisture, θL the lower threshold, and θH the upper threshold, with θH greater than θL. Watering becomes eligible when filtered moisture remains below θL for a persistence interval. The pump is stopped when moisture reaches θH or another stop condition occurs. This separation creates hysteresis: after the controller starts watering, small fluctuations around the lower threshold do not cause repeated start-stop cycling. Thresholds should be selected by observation and plant-specific guidance. A sensible calibration trial records the moisture range associated with healthy growth and checks how quickly the pot dries under typical light, airflow, and temperature. A single probe measures only a local area, so multiple probe positions or repeated manual checks may be needed to ensure that the target range represents the whole root zone. The controller should avoid maintaining the substrate continuously near saturation, because persistent waterlogging can reduce oxygen availability around roots.

C. Filtering and plausibility checks

A median filter can reject isolated spikes, while an exponential moving average can smooth normal variation. If x_k is the latest reading and y_{k-1} the previous filtered value, an exponential filter is $y_k = \alpha x_k + (1-\alpha) y_{k-1}$, where α between zero and one determines responsiveness. A low α produces smoother output but reacts more slowly to genuine changes. Filtering must not hide a disconnected sensor or freeze the control loop; stale readings, out-of-range values, and implausibly abrupt jumps should be flagged separately.

CONTROL ALGORITHM AND STATE MACHINE

The controller is implemented as a finite-state machine to make behaviour predictable and testable. The primary states are IDLE, WATERING, SOAK, and FAULT. IDLE means that sensors are being evaluated and no water is being delivered. WATERING means the pump is active and all run-time limits are monitored. SOAK means the pump is off while water redistributes through the substrate. FAULT means an abnormal condition has disabled automatic actuation.

State	Entry condition	Required behaviour
IDLE	Normal operation; no active irrigation	Sample and filter sensors; check persistent moisture deficit and minimum off-time.
WATERING	Eligible request and all interlocks pass	Run pump; verify flow; monitor moisture, tank level, dose, and maximum duration.
SOAK	Normal watering stop condition	Keep pump off for a configured interval, then reassess moisture.
FAULT	Invalid sensor, low tank, flow failure, or emergency stop	Pump OFF, log fault, notify user, require recovery conditions before restart.

A. Decision sequence

- Acquire moisture, temperature, tank level, and flow measurements with timestamps.
- Validate measurements. If a required sensor is invalid or stale, prevent automatic watering and record the reason.
- Check the tank level and emergency-stop input before any pump command.
- Apply filtering and increment a persistence counter only while moisture remains below the lower threshold.
- Start watering only when the persistence count, minimum off-time, and safety conditions are all satisfied.
- Verify flow within a startup timeout. If flow is absent, stop the pump and record a delivery fault.
- During watering, stop at the upper moisture threshold, maximum runtime, maximum delivered volume, low tank, or any detected fault.
- Enter SOAK after a normal stop, then return to IDLE and reassess the substrate.
- Publish data when connected; if offline, continue local control and store a bounded number of records.

The controller should define priority explicitly: emergency stop and safety interlocks override all watering requests, including manual commands. If a moisture sensor fails, a conservative implementation should disable automatic watering rather than infer dryness from a missing value. A separate, clearly indicated user action may be provided for supervised troubleshooting, but it must still respect pump runtime and tank protection.

HARDWARE IMPLEMENTATION AND ELECTRICAL SAFETY

A microcontroller board with digital inputs, analog inputs, a hardware timer, and Wi-Fi is adequate for a single-zone prototype. The moisture probe may use an analog input, while tank and flow sensors may provide digital switch signals or pulse trains. Temperature can be measured by a digital sensor. Firmware should sample sensors on a fixed schedule and separate sensor acquisition from communication so that a slow network request cannot delay the control loop. The pump driver must be selected according to the actuator voltage, normal current, and startup or stall current. A logic-level MOSFET is suitable for many low-voltage DC pumps when correctly rated and wired; a relay may be used when its contact rating and switching life are appropriate. Inductive loads require suitable suppression, and the circuit should include a fuse or current-limiting protection. The power supply must support simultaneous controller and pump demand without causing brownouts. A hardware default-off state is recommended so that a reset or disconnected control signal does not unexpectedly energize the pump. Water and electronics should be physically separated. Use drip loops, strain relief, insulated connectors, and an enclosure suitable for the expected humidity and splashing.

Any mains-powered supply must be enclosed and installed in accordance with local electrical safety requirements. Testing should begin with the pump disconnected or replaced by a safe indicator load, then proceed to water tests under supervision. The reservoir should be placed where overflow cannot damage nearby equipment.

A. Flow measurement

A flow meter provides evidence that water is moving, whereas a pump status bit only confirms that the controller issued a command. If a pulse-output meter is used, delivered volume can be estimated by $V = n/Kf$, where n is the number of pulses and Kf is the calibrated pulse count per litre. Kf should be determined by collecting a known volume at the intended operating flow rate. The estimate may be less reliable at very low flow rates, and the firmware should treat implausible readings as a possible sensor or plumbing fault.

B. Reservoir-level protection

A low-level switch should be placed above the pump intake's minimum safe level. The controller should check the switch before starting and throughout watering. A tank sensor that reports continuous level may support a remaining-water display, but a simple float switch can be sufficient for dry-run protection. The dashboard must distinguish a measured level from an estimated percentage derived from tank geometry.

FIRMWARE, IOT COMMUNICATION, AND USER INTERFACE

Firmware should be separated into modules for sensor acquisition, calibration, filtering, state-machine control, actuator output, event logging, communication, and configuration. This modular structure simplifies testing and allows the moisture sensor or pump driver to be changed without rewriting the complete application. Configuration parameters should be range-checked before being stored. On reboot, the controller should load validated settings and initialize the pump output to OFF. Each telemetry record should contain a device identifier, timestamp, raw and calibrated moisture, temperature, tank status, flow state or volume, pump state, operating mode, and fault code. Units should be explicit. The system should distinguish missing data from a zero value and should mark records as stale when communication has stopped. Historical records enable the user to identify drying patterns, repeated short cycles, or changes in water use over time. The mobile or web dashboard should have three main views. The current-status view presents moisture, temperature, tank level, flow, and pump state. A history view plots readings and marks watering events. A configuration view allows authorized users to adjust plant profile, thresholds, soak interval, maximum runtime, and alert preferences. Manual watering should require a deliberate action, show a countdown, and stop automatically when the requested duration expires. Remote communication must not be required for safe local operation. When Wi-Fi is unavailable, the device should continue sampling and applying local rules, store a bounded event queue, and transmit queued records after reconnection. Remote commands should include authentication, validation, and a response indicating whether the device accepted and executed the request. Credentials should not be displayed in logs, and firmware updates should be performed through a controlled process. These measures reduce the risk that an internet or application failure disrupts watering or exposes control of the pump.

EXPERIMENTAL METHODOLOGY

The design should be evaluated using a controlled comparison rather than relying on the dashboard's appearance or a few successful watering events. The minimum comparison includes a fixed-time baseline (B1), a single-threshold moisture controller without delivery verification (B2), and the full AquaPulse controller (A3). Where feasible, use multiple pots with the same plant type, container size, substrate mass, light exposure, and initial moisture condition. Randomize pot positions or rotate them periodically to reduce location effects.

A. Test procedure

- Document the plant species, pot dimensions, substrate composition, sensor model, probe depth, pump rating, and reservoir volume.
- Calibrate the moisture sensor using representative samples and record the calibration curve and uncertainty.
- Choose lower and upper moisture thresholds and document how they were selected.
- Run each controller for the same observation period and record moisture, temperature, water volume, pump runtime, tank status, and faults.
- Use a calibrated measuring container or flow meter to verify delivered volume independently.
- Repeat trials and report the number of replicates, trial duration, missing samples, and any exclusions.
- Inject safe faults, including low reservoir, blocked tubing, disconnected moisture probe, flow-sensor failure, and network loss.
- Compare water use and moisture compliance together so that apparent savings are not achieved by leaving the plant too dry.

B. Evaluation metrics

Metric	Definition	Interpretation
Water use	Total delivered water volume (L) per pot and trial	Primary measure of applied water.
Water-use reduction	$100 \times (VB - VA) / VB$ (%)	Relative reduction against a named baseline.
Moisture compliance	Time inside target band / valid observation time × 100%	Measures how consistently the root zone remains in range.
Overshoot duration	Time above the upper moisture threshold	Indicates excess wetness after watering.
Pump runtime	Sum of pump-on durations	Supports comparison of operating demand.
Energy use	Measured energy (Wh) over the trial	Quantifies electrical demand.
Fault-detection rate	Correctly detected injected faults / all injected faults	Assesses the monitoring and interlocks.
False-trigger rate	Unnecessary watering events per day	Indicates sensitivity to noise or poor thresholds.
Prediction MAE/RMSE	Error on held-out data	Applies only if a moisture prediction model is tested.

For actual moisture θ and predicted moisture $\hat{\theta}$ over n samples, mean absolute error is $MAE = (1/n)\Sigma|\theta - \hat{\theta}|$, and root-mean-square error is $RMSE = \sqrt{[(1/n)\Sigma(\theta - \hat{\theta})^2]}$. These metrics should be calculated on a chronologically held-out test period, not on the same records used to fit a prediction model. The report should include sampling frequency, calibration method, sensor placement, number of plants, and summary statistics such as mean and standard deviation. Confidence intervals or statistical tests may be used when the sample size and design justify them.

C. Fault-injection testing

For a low-tank test, lower the water level to the sensor trigger point and confirm that the pump cannot start or stops promptly. For a blocked-tube test, simulate a blockage safely and verify that the flow timeout turns the pump off. For a disconnected moisture sensor, confirm that the firmware marks the reading invalid and does not interpret it as a dry substrate. For a network-loss test, disable connectivity and verify that local control continues.

Each test should record the initial condition, expected response, observed response, and recovery procedure. Fault tests must be performed with appropriate supervision to prevent spills or pump damage.

RESULTS AND DISCUSSION

No physical prototype measurements or experimental dataset were supplied for this manuscript. Consequently, numerical water savings, growth improvements, sensor accuracy, and fault-detection rates are not reported as achieved results. Once testing is completed, this section should be replaced or extended with actual tables and plots. Each result should identify the baseline, trial duration, number of replicates, units, uncertainty, and any missing data. The most important comparison is water use alongside moisture-band compliance. A controller may use less water simply because it fails to irrigate enough; that outcome should not be described as an improvement. Conversely, a controller that maintains moisture but runs the pump excessively may indicate poor

threshold selection, sensor placement, or leakage. Flow measurement can help distinguish low consumption caused by efficient irrigation from low consumption caused by a blocked line.The proposed architecture is expected to improve fault visibility because it monitors tank and flow conditions in addition to soil moisture. Hysteresis and persistence filtering are intended to reduce rapid cycling due to sensor noise. Local operation is intended to preserve watering during internet outages. These are engineering hypotheses that require tests. Sensor drift, root-zone heterogeneity, uneven water distribution, and the limited coverage of a single probe remain important sources of error.Temperature is contextual telemetry, not a direct substitute for soil-moisture measurement. High temperature may increase evaporation and plant demand, but it does not prove that the root zone is dry. Similarly, applying water may affect root-zone temperature, but cooling should not be claimed as a validated benefit without measurements. The primary objective remains maintaining a calibrated moisture band while controlling water delivery safely.

CONCLUSION

This paper has presented AquaPulse, a proposed autonomous irrigation controller that combines calibrated soil-moisture sensing, temperature telemetry, reservoir-level protection, flow verification, local pump control, and IoT-based monitoring. The design uses a moisture band with hysteresis, persistence filtering, soak periods, maximum runtime and volume limits, and fault-aware state transitions. It also specifies the hardware interfaces, telemetry model, calibration process, and controlled evaluation methodology needed to assess the system objectively. The central contribution is a practical design that treats irrigation as a feedback-control problem rather than a fixed timer task. Its benefits remain hypotheses until supported by measured results. A completed prototype should therefore be tested against clear baselines and reported with transparent methods, units, uncertainty, and reproducible data.

REFERENCES

[1] J. Gutiérrez, J. F. Villa-Medina, A. Nieto-Garibay, and M. Á. Porta-Gándara, “Automated irrigation system using a wireless sensor network and GPRS module,” IEEE Transactions on Instrumentation and Measurement, vol. 63, no. 1, pp. 166–176, Jan. 2014, doi:

[2] I. Rajapaksha, T. Gamage, I. Eranga, A. Dharmasena, and K. Gamage, “Smart irrigation system for domestic homes,” in Proc. IEEE International Conference on Smart Computing and Systems Engineering (SCSE), 2022, doi:

[3] “An IoT-Based Soil Moisture Management System for Precision Agriculture: Real-Time Monitoring and Automated Irrigation Control,” in Proc. 4th International Conference on Smart Electronics and Communication (ICOSEC), 2023, doi:

[4] “IoT-Enabled Adaptive Watering System With ARIMA-Based Soil Moisture Prediction for Smart Agriculture,” IEEE Access, vol. 13, pp. 27714–27728, 2025, doi:

[5] “IoT and Machine Learning-Based Smart Soil Irrigation Farming Systems,” in Proc. IEEE International Students’ Conference on Electrical, Electronics and Computer Science (SCEECS), 2024, doi:

[6] “The Smart IoT Based Automated Irrigation System Using Arduino UNO and Soil Moisture Sensor,” in Proc. 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), 2022, doi:

[7] “Smart Plant Irrigation System with Arduino Uno and Soil Moisture Sensor,” in Proc. 2nd International Conference on Computational Intelligence, Communication Technology and Networking (CICTN), 2025, doi:

[8] T. J. Schmugge, T. J. Jackson, and H. L. McKim, “Survey of methods for soil moisture determination,” Water Resources Research, vol. 16, no. 6, pp. 961–979, Dec. 1980, doi:

[9] M. Th. van Genuchten, “A closed-form equation for predicting the hydraulic conductivity of unsaturated soils,” Soil Science Society of America Journal, vol. 44, no. 5, pp. 892–898, 1980, doi:

[10] M. Pramanik, M. Khanna, M. Singh, D. K. Singh, S. Sudhishri, A. Bhatia, and R. Ranjan, “Automation of soil moisture sensor-based basin irrigation system,” Smart Agricultural Technology, vol. 2, Art. no. 100032, 2022, doi: