iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: IV Month of publication: April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79718

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

IOT Based Automatic Soil Moisture Monitoring and Irrigation System

Tejas Bhadwale[1], Chaten Aher[2], Omkar Vengurlekar[3], Deepak Maurya[4], Sandeep Lalji[5], Swapnil Jadhav[6]
Electrical Engineering

Abstract: Agriculture plays a vital role in the economy, especially in developing countries where efficient water management is essential. Traditional irrigation methods often lead to excessive water usage, uneven distribution, and increased labor requirements. The IoT Based Automatic Soil Moisture Monitoring and Irrigation System aims to overcome these limitations by providing an intelligent and automated solution for irrigation control. The system continuously monitors soil moisture and controls a water pump via an IoT-enabled microcontroller, logging data to a cloud platform and enabling remote monitoring and manual override through a mobile application. The methodology involves sensing soil moisture in real time, comparing it with a predefined threshold value, and automatically switching the irrigation pump ON or OFF. This closed-loop system ensures optimal water usage, reduces human intervention, and enhances crop productivity. The proposed system is cost-effective, energy-efficient, and suitable for small-scale as well as large-scale agricultural applications.
Keywords: Automatic Irrigation, Soil Moisture sensor, Internet of Things, smartphone.

I. INTRODUCTION

Quality improvement in agricultural irrigation is essential to reduce crop failure, conserve water, and lower labor demands. Manual inspection and fixed-schedule watering are inefficient for many farms and gardens, particularly those located at a distance from the owner or in areas with variable climatic conditions. The IoT Based Automatic Soil Moisture Monitoring and Irrigation System addresses these limitations by integrating continuous soil moisture sensing with local decision making, reliable actuation, and remote oversight. At the heart of the system is an ESP32 microcontroller that performs edge processing of sensor data, applies calibrated mappings to raw readings, and executes control logic to operate a relay-driven pump only when soil moisture falls below a configurable threshold. This local intelligence reduces unnecessary actuation and ensures that irrigation events are driven by actual soil water needs rather than by rigid timetables.

Telemetry and operational state are transmitted to a cloud platform, where data are archived for historical analysis and presented through a mobile or web interface that supports secure setpoint management and manual override. The cloud component enables remote monitoring of multiple nodes, centralized configuration, and the generation of time-stamped logs that can inform irrigation planning and water budgeting. To enhance robustness in real-world deployments, the design incorporates software practices such as local buffering and retry mechanisms to tolerate intermittent connectivity, and sensor calibration routines to compensate for soil heterogeneity and sensor drift. These measures, together with the use of low-cost, modular hardware, strike a balance between affordability and operational reliability, making the system suitable for small to medium plots, greenhouses, and home gardens where Wi-Fi or cellular connectivity is available.

Beyond immediate operational benefits, the platform is intentionally extensible: additional sensors, nodes, or control algorithms can be integrated without fundamental redesign, and the same architecture supports scaling from a single garden node to coordinated multi-node installations across larger farms. The recorded moisture and actuation history provide a valuable dataset for refining control logic, evaluating crop responses, and, in future iterations, incorporating predictive inputs such as weather forecasts or evapotranspiration models to enable adaptive irrigation strategies. By automating routine tasks, reducing water waste, and enabling data-driven decision making, the system lowers labor requirements, improves moisture uniformity across cultivated areas, and contributes to more sustainable water stewardship in agricultural practice.

II. RELATED RESEARCH

Contemporary research on IoT-enabled irrigation systems consistently adopts an architecture that integrates local sensing and edge control with networked telemetry and cloud-based visualization. Numerous studies document practical prototypes built from low-cost microcontrollers such as ESP8266, ESP32, and Arduino, which acquire soil moisture data and actuate pumps or valves while transmitting telemetry to platforms like Firebase, ThingSpeak, or Blynk for remote monitoring and control.

Where local aggregation or heavier processing is required, single-board computers such as Raspberry Pi and Banana Pi are employed as gateways, enabling hybrid deployments that combine lightweight sensor nodes with more capable edge devices. Across these implementations, sensor selection emerges as a pivotal design decision. Low-cost resistive probes (for example, the FC-28) are common in proof-of-concept systems because of their affordability and simple interface; however, field studies repeatedly report issues with corrosion, drift, and nonlinear response that necessitate frequent recalibration. Capacitive moisture sensors are increasingly recommended for long-term deployments because they deliver more stable readings and resist degradation in wet soils, while high-accuracy techniques such as time-domain reflectometry and frequency-domain reflectometry are typically reserved for precision applications where the additional cost and complexity are justified.

Communication strategies vary according to deployment scale and connectivity constraints. In urban and peri-urban contexts with reliable broadband, Wi-Fi-based nodes using ESP32/ESP8266 modules offer low-latency telemetry and straightforward integration with cloud services and mobile dashboards. For remote or widely distributed farms, cellular links (GSM/GPRS) remain a practical option despite recurring data costs, and low-power wide-area networks such as LoRa and NB-IoT are gaining traction for their long range and energy efficiency. Comparative analyses in the literature indicate that Wi-Fi is the most convenient choice for small, localized systems, GSM provides ubiquitous coverage at a recurring cost, and LPWAN technologies are preferable when node density and geographic spread make per-node connectivity expensive or impractical. Robust local buffering and retry logic on the node are frequently emphasized as essential design elements to tolerate intermittent connectivity, and many implementations adopt lightweight protocols such as MQTT or compact HTTP clients to minimize bandwidth and improve reliability.

Control strategies reported in the literature span from simple threshold-based activation to more sophisticated adaptive and predictive approaches. Threshold control—where irrigation is triggered when measured moisture falls below a configurable setpoint—remains the most widely used method due to its simplicity, deterministic behaviour, and suitability for resource-constrained microcontrollers. More advanced research explores fuzzy logic controllers, model predictive control, and machine learning models that incorporate weather forecasts, evapotranspiration estimates, crop growth stage, and historical soil moisture patterns to optimize irrigation timing and volume. These adaptive approaches have demonstrated additional water savings and improved crop outcomes in experimental studies, but they require more data, computational resources, and careful model training, which can complicate deployment and maintenance in low-resource settings.

Cloud platforms and data handling practices are another recurring theme. Rapid prototyping commonly leverages managed services such as Firebase, Blynk, and ThingSpeak because they provide real-time databases, authentication, and visualization with minimal setup. For production systems, the literature recommends secure device authentication, encrypted communication channels, and local data persistence to prevent data loss during outages. Case studies report telemetry latencies on the order of a few seconds under favourable network conditions and stress that user experience and control responsiveness depend heavily on local network quality and service provider performance. Open-source repositories and community projects often accompany academic work, providing reusable firmware, dashboard templates, and integration examples that accelerate development.

Finally, the body of research highlights persistent practical challenges and deployment lessons. Sensor calibration and long-term reliability remain primary concerns; site-specific calibration and periodic validation are necessary when using resistive probes. Power management is critical for off-grid installations, and solar energy combined with battery management is frequently proposed as a resilient solution. Security, maintainability, and modular hardware design are emphasized for systems that scale beyond hobby projects: secure firmware update mechanisms, easy replacement of degraded sensors, and clear maintenance schedules improve longevity. Collectively, these findings suggest a pragmatic development pathway: adopt low-maintenance sensors, implement a robust threshold control loop on an ESP32 or similar node, use Wi-Fi and a managed cloud backend for rapid deployment where connectivity exists, and progressively incorporate predictive control, LPWAN connectivity, or renewable power as operational scale and requirements evolve.

III. METHOD

The system implements a closed-loop control paradigm that continuously senses soil moisture, makes local decisions at the edge, actuates irrigation hardware when required, and logs operational data for remote monitoring and analysis. The overall architecture couples an ESP32 DEV KIT V1 microcontroller with field sensors and an actuator chain, and links the field node to a cloud backend and a mobile/web interface. Sensor readings are sampled at configurable intervals, converted into calibrated moisture values, and evaluated against user-defined thresholds. When the measured moisture falls below the configured setpoint, the controller energizes a relay to run the water pump until the moisture returns to acceptable levels or until a maximum run time is reached. All events, sensor values, and status changes are timestamped and transmitted to the cloud for visualization, historical review, and remote configuration.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

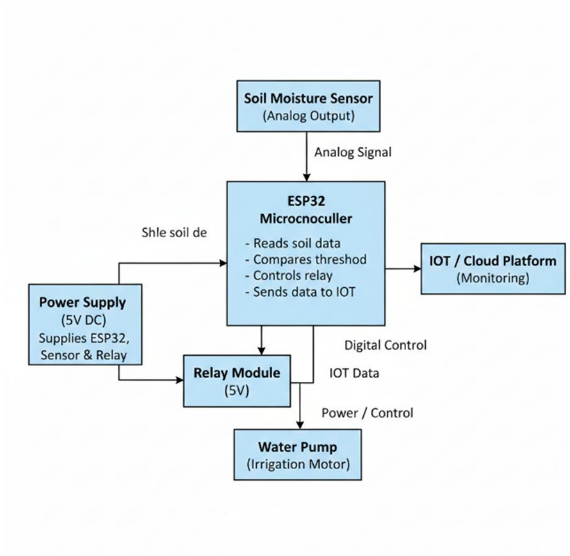


Fig.1 Flowchart of IOT Based Automatic Soil Moisture Monitoring and Irrigation system

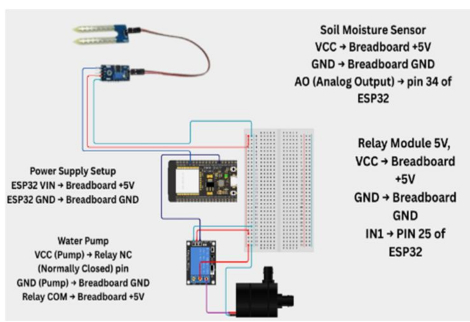


Fig.2 Output

A. Hardware Design

The hardware platform centers on the ESP32 DEV KIT V1, chosen for its integrated Wi-Fi capability, sufficient analog input resolution, and low power modes that facilitate both mains-connected and battery-backed operation. Soil moisture is measured by probes installed in the root zone; while the prototype uses the FC-28 resistive probe for cost-effective testing, the design anticipates the use of capacitive sensors for production deployments because of their superior stability and resistance to corrosion. The sensor's analog output is routed to the ESP32 ADC input and conditioned through a simple signal path that includes decoupling and, where necessary, a voltage divider to match ADC input range. Actuation is performed by a 5 V relay module rated for 5 A, which isolates the microcontroller from the pump's power circuit and provides a safe switching interface for the 5 V centrifugal submersible pump. Power is supplied by a regulated source; the prototype demonstrates operation from two 18650 Li-ion cells with voltage regulation provided by a 7805 regulator for the pump and an AMS1117 regulator for the ESP32, but the design supports alternative power arrangements including direct 5 V supplies or solar-charged battery systems with appropriate charge controllers. Peripheral wiring is organized on a prototyping breadboard for development, with the expectation that a field unit will migrate to a sealed enclosure and a printed circuit board for reliability. Mechanical considerations such as waterproofing, cable strain relief, and pump plumbing are addressed in the field layout to ensure long-term operation and to protect electronics from environmental exposure.

The conceptual block diagram places the ESP32 at the center of a local control loop: the sensor provides analog measurements to the ESP32, the ESP32 executes control logic and drives the relay to operate the pump, and the ESP32 also manages network connectivity to the cloud. The mobile application and cloud backend form the remote management layer, enabling operators to view real-time values, adjust setpoints, and issue manual overrides. Safety features are incorporated at the hardware level through relay isolation, fuse protection on the pump supply, and watchdog timers on the microcontroller to recover from software faults.

B. Software Design

The software stack is divided between firmware running on the ESP32 and cloud services that provide storage, visualization, and remote configuration. The firmware is responsible for sensor acquisition, calibration, decision logic, actuation control, telemetry packaging, and fault handling. Raw ADC samples are averaged and mapped to calibrated moisture percentages using a site-specific calibration curve to compensate for soil type and sensor characteristics. The control algorithm compares the calibrated moisture value to a configurable setpoint retrieved from the cloud; when the value falls below the setpoint the firmware engages the relay to start the pump and monitors moisture and elapsed time to determine when to stop. To prevent excessive cycling and to protect the pump, the firmware enforces minimum off and on intervals and a maximum continuous run time. Telemetry messages include the calibrated moisture value, raw ADC reading, pump state, battery or supply voltage, and a timestamp; messages are serialized in a compact JSON format and transmitted to the cloud over HTTPS or MQTT depending on the chosen backend.

Resilience to network interruptions is achieved through local buffering and retry logic. When the node cannot reach the cloud, telemetry and event records are stored in non-volatile memory and retransmitted when connectivity is restored. The firmware also supports over-the-air firmware updates, secure device authentication, and configurable logging levels to facilitate maintenance and troubleshooting. On the cloud side, a real-time database stores incoming telemetry, maintains device configuration parameters such as setpoints and timing limits, and exposes APIs for the mobile application. The mobile interface provides a dashboard that displays current moisture readings, pump status, historical charts, and controls for manual override and setpoint adjustment. Administrative functions include device registration, secure credential management, and access control to ensure that only authorized users can change operational parameters.

Testing and validation procedures include sensor calibration against reference meters, pump flow characterization to translate run time into delivered volume, and end-to-end latency measurements to quantify the time between a sensor reading and its appearance in the cloud. These measurements inform control parameters and user expectations for responsiveness. Together, the hardware and software design choices produce a modular, extensible system that balances cost, reliability, and functionality while enabling future enhancements such as weather-aware scheduling, predictive irrigation algorithms, and integration with alternative communication technologies.

IV. RESULT AND ANALYSIS

Experiments were designed to validate three core aspects of the prototype: sensor calibration, actuator performance, and end-to-end data transfer to the cloud. Each test series was repeated multiple times and averaged to reduce random error and to provide representative performance metrics under the prototype's operating conditions. The following sections summarize the measured behaviour, interpret the results, and highlight practical implications for deployment and further development.

A. FC-28 Soil Moisture Sensor Calibration

Calibration of the FC-28 resistive probe was performed by collecting analog-to-digital converter (ADC) readings across a range of soil moisture conditions and comparing those readings to reference moisture levels. The calibration campaign comprised ten runs with ten samples per run; averaged values were used to construct a mapping between the FC-28 ADC output and nominal moisture meter values. The measured ADC values decreased as soil moisture increased, with the driest condition returning an average ADC of approximately 891 and the wettest condition returning an average ADC of approximately 312. Intermediate moisture levels produced non-uniform spacing in ADC values: for example, ADC values of roughly 858, 599, 546, 487, 457, 441, 414 and 354 corresponded to the successive moisture meter steps between the extremes. These results indicate a clear inverse relationship between ADC reading and moisture content, but they also reveal significant nonlinearity in the midrange. In practice this nonlinearity and the known susceptibility of resistive probes to corrosion and environmental drift imply that per-site calibration and periodic revalidation are necessary if FC-28 probes are used long term. For deployments that prioritize low maintenance and long-term stability, capacitive sensors are recommended because they provide a more linear response and are less affected by soil chemistry and electrode degradation. Nevertheless, the calibration table derived from these tests is sufficient to implement a threshold-based control loop for the prototype and to translate ADC readings into actionable moisture categories for irrigation control.

B. DC Water Pump Discharge Test

Pump performance was characterized by measuring the volume of water delivered over a fixed three-second interval at a range of drive levels.

Each measurement was repeated ten times and averaged to produce a reliable flow profile. At full drive the pump delivered approximately 167 milliliters in three seconds, which extrapolates to roughly 3,340 milliliters per minute under steady conditions. As the drive level was reduced, the delivered volume decreased in a roughly monotonic fashion: at 90 percent drive the average three-second volume was about 146 milliliters, at 80 percent it was about 126 milliliters, and at 70 percent it was about 110 milliliters. Below approximately 30 percent drive the pump's output fell to marginal or zero measurable flow in the three-second sampling window. These measurements allow the system designer to convert desired irrigation volumes into precise pump run times and to select appropriate PWM or drive settings for different irrigation tasks. The data also underscore the importance of characterizing the pump across the expected operating range: small changes in drive level produce measurable differences in delivered volume, and the pump exhibits a practical lower bound below which flow is negligible. For reliable volumetric control, the firmware should enforce minimum on-time durations and avoid operating the pump in the low-efficiency region where flow is inconsistent.

C. Data Transfer Performance

End-to-end data transfer performance was evaluated by measuring the latency between the timestamp when the ESP32 sent a telemetry packet and the timestamp when the same record appeared in the cloud database. Ten sequential samples were recorded and averaged to quantify typical responsiveness under the test network conditions. The measured response times varied between sub-second and several seconds, with an average round-trip latency of approximately 2.28 seconds. To assess the influence of network provider on latency, a broader test was conducted using three different internet service providers with twenty samples each. The provider-level averages observed in that comparison were approximately 3.4 seconds for Provider A (First Media), 0.6 seconds for Provider B (MyRepublic), and 1.6 seconds for Provider C (IM3). These results demonstrate that cloud update latency is primarily a function of local network performance and the chosen access provider; under typical conditions observed in the tests, telemetry arrival at the cloud remained below four seconds. From a system design perspective, these latencies are acceptable for non-time-critical irrigation control, but they reinforce the need for local decision making and buffering: the node must be capable of executing control actions autonomously and storing telemetry locally when connectivity is degraded, then synchronizing historical records once the network is restored.

D. Data Visualization and System Behaviour

Real-time telemetry from multiple sensor nodes was successfully displayed on the mobile dashboard and in the cloud console during the test campaigns. The dashboard presented calibrated moisture values, node identifiers, pump status, and timestamps, enabling remote monitoring and manual override. The system's logging capability produced time-stamped records of moisture trends and irrigation events that can be used to refine setpoints, estimate water budgets, and support agronomic analysis. During extended runs the combination of calibrated sensor mapping, pump flow characterization, and cloud-based setpoint management allowed the prototype to maintain soil moisture within target bands with minimal manual intervention. Observed operational issues were primarily related to sensor nonlinearity and occasional network interruptions; both were mitigated by applying the calibration mapping and by implementing local buffering and retry logic in the firmware.

The experimental results validate the feasibility of a low-cost, IoT-driven irrigation controller for small to medium agricultural applications. Key practical takeaways are that sensor selection and calibration materially affect control accuracy, pump characterization is essential for volumetric irrigation planning, and network variability necessitates robust local autonomy and data persistence. For production deployments, migrating from resistive probes to capacitive sensors will reduce maintenance overhead and improve long-term accuracy. Likewise, hardening the hardware with a PCB, weatherproof enclosure, and proper power management (including solar options for off-grid sites) will increase reliability. Finally, the measured cloud latencies support the chosen architecture in which the ESP32 performs primary control decisions locally while the cloud provides monitoring, historical analysis, and remote configuration.

V. CONCLUSION

Based on experimental data, it can be concluded that data transfer to the database did not experience problems. But there is an average delay of 2.28 seconds. Based on testing of 3 internet providers, the results of the speed of sending data to the firebase averaged no more than 4 seconds. The process of sending data is dependent on the speed of internet provider.

REFERENCES

[1] A. Parlina, H. Murfi and K. Ramli, "Smart City Research in Indonesia: A Bibliometric Analysis," 2019 16th International Conference on Quality in Research (QIR): International Symposium on Electrical and Computer Engineering, Padang, Indonesia, 2019, pp. 1-5, doi: 10.1109/QIR.2019.8898264.

[2] A. Firmansah, Aripriharta, I. M. Wirawan, H. W. Herwanto, I. Fadlika and Muladi, "Design and Experimental Validation of the Self-powered IoT for Indoor Temperature-Humidity Monitoring," 2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE), Denpasar, Bali, Indonesia, 2019, pp. 139-143, doi: 10.1109/ICEEIE47180.2019.8981426.

[3] M. Suryanegara, A. S. Arifin, M. Asvial, K. Ramli, M. I. Nashiruddin and N. Hayati, "What are the Indonesian Concerns About the Internet of Things (IoT)? Portraying the Profile of the Prospective Market, in IEEE Access, vol. 7, pp. 2957-2968, 2019, doi: 10.1109/ACCESS.2018.2885375.

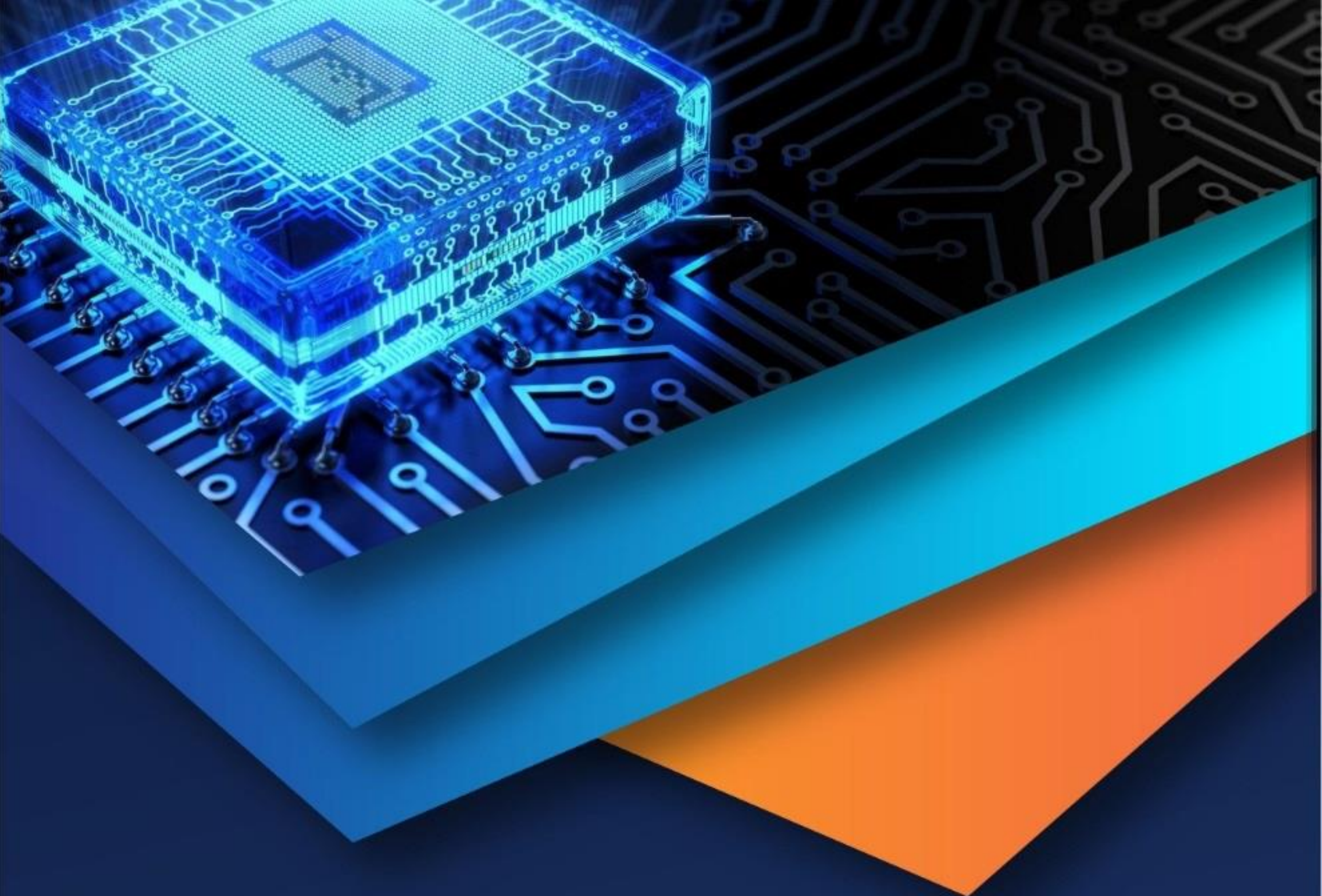
[4] A Wijaya, M Rivai. (2018). Control and Monitoring Irrigation System IoT-based Using Banana Pi.

[5] A. R. Altahtawi, E.Andika, W. N. Harjanto, “Initial Design Development of Wireless Node-based Automatic Irrigation Control Systems and The Internet of Things,” 2018 ISSN: 20852517.

[6] P. W. Ciptadi, R. H. Hardyanto. (2018). Application of IoT Technology on Hydroponic Plants Using Arduino and Blynk Android.

[7] M. S. Hadi, M. R. Maulana, M. A. Mizar, I. A. E. Zaeni, A. N. Afandi and M. Irvan, "Self Energy Management System for Battery Operated Data Logger Device Based on IoT," 2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE), Denpasar, Bali, Indonesia, 2019, pp. 133-138, doi: 10.1109/ICEEIE47180.2019.8981406.

[8] M. S. Hadi, M. A. A. Shidiqi, I. A. E. Zaeni, M. A. Mizar and M. Irvan, "Voice-Based Monitoring and Control System of Electronic Appliance Using Dialog Flow API Via Google Assistant," 2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE), Denpasar, Bali, Indonesia, 2019, pp. 106110, doi: 10.1109/ICEEIE47180.2019.8981415.



INDEX COPERNICUS

10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY