# IOT-ENABLED WEB-BASED MONITORING AND CONTROL SYSTEM FOR SOLAR-POWERED RICE FIELD IRRIGATION PUMPS: A SUSTAINABLE SMART AGRICULTURE DESIGN

Ni Made Sri Ariantini[1*], Riforson Wirawan Julianus Gagola[2], I Gede Para Atmaja[3]
[1,2,3] Politeknik Negeri Manado, Indonesia
*Corresponding Author: nimadesriariantini23@gmail.com

**ABSTRACT**
Efficient and sustainable irrigation requires an integrated approach to water management, energy supply, and remote supervision. This study aimed to design an Internet of Things (IoT)-based monitoring and control system for a paddy-field irrigation pump using a web platform and a solar photovoltaic power source. A Research and Development approach was applied through literature review, functional requirement analysis, hardware and software architecture design, control-logic development, interface design, and database modelling. The proposed system uses an ESP32 microcontroller, an HC-SR04 ultrasonic sensor for water-level measurement, a PZEM004T module for voltage, current, power, and energy monitoring, a relay-driven pump actuator, a local LCD, and a photovoltaic energy chain consisting of a solar panel, charge controller, battery, and inverter. The resulting design supports real-time monitoring, automatic operation based on minimum and maximum water-level thresholds, manual web-based control, historical data storage, and protection against critically low water levels and excessive pump power. The architecture also separates sensing, processing, communication, actuation, energy, application, and data-management layers, thereby improving maintainability and future scalability. The design provides a coherent basis for a more flexible and environmentally responsible irrigation system; however, field implementation and performance testing are still required to evaluate sensor accuracy, communication stability, energy efficiency, and long-term reliability.
**Keywords:** Automated Pump Control; Internet of Things; Rice-Field Irrigation; Solar Photovoltaic System; Web-Based Monitoring

## INTRODUCTION

The availability of timely water is one of the main prerequisites for maintaining production sustainability in rice fields. In many locations, irrigation still relies heavily on manual channel opening or pump operation based on the direct observation of farmers. The pattern can function on a small scale, but it becomes less effective when the water source is far enough, land conditions change rapidly, operator power is limited, or irrigation must be done at a given time. Operations that rely solely on manual inspections also make it difficult to record water conditions and energy consumption, so decisions are usually made based on estimates rather than traceable data. As a result, the pump could potentially run longer than necessary, stop late when the reservoir is full, or remain operational when water conditions are unsafe.

Irrigation problems are not only related to water distribution, but also to the availability and cost of energy. Pumps that use the power grid require stable access, while the use of fuel incurs recurring operational costs, logistical needs, and direct emissions. In agricultural areas far from the power grid, limited energy supply can hamper irrigation schedules and reduce service reliability. In this context, the irrigation system needs to be designed as a unit that includes water condition measurement, pump control, energy supply, device protection, data storage, and user interface. A piecemeal approach can result in tools that work locally, but they don't necessarily provide enough information for long-term monitoring, evaluation, and decision-making.

The Internet of Things (IoT) provides a framework for connecting sensors, control devices, communication networks, and monitoring applications in a single data stream. Da Xu, He, and Li (2014) explain that IoT in industrial environments allows physical objects to generate data, communicate with each other, and support automation and decision-making. Kumar, Tiwari, and Zymbler (2019) also emphasized the role of IoT as an approach that integrates devices, connectivity, and digital services. In irrigation management, the integration allows water levels, pump conditions, and electrical parameters to be monitored periodically without requiring operators to be on site. The same data can be used to automatically activate the pump, provide alerts, and compile an operation history.

The application of IoT in agriculture has evolved from just sensor readings to resource management support systems. Nandurkar, Thool, and Thool (2014) designed a precision farming system based on a wireless sensor network to obtain field data in a structured manner. Khan, Hossain, and Islam (2020) show that soil and weather monitoring can be combined with irrigation control to improve the system's response to environmental conditions. Although the characteristics of the sensor and the measurement object may differ, the relevant principle is that irrigation decisions should preferably be based on actual data, explicit control rules, and communication channels that allow the user to supervise. Thus, automation does not eliminate the role of the user, but provides a more consistent basis of information.

On the energy side, the utilization of solar power plants offers an opportunity to increase the independence of pump systems in locations that obtain adequate solar radiation. Saxena and Dutta (2019) place solar energy as an important part of the transition to renewable energy. In the application of water pumping, Chandel, Naik, and Chandel (2015) explained that the photovoltaic water pumping system can be used for irrigation and water supply needs, especially in areas that are difficult to reach by the power grid. The integration of solar power plants with irrigation pumps reduces dependence on conventional supplies, but the design must take into account the adequacy of the panels, battery charging characteristics, power conversion, initial needs of the motor, long operating time of the pump, and losses on the inverter.

Previous studies have generally highlighted one part of the system, such as monitoring farm conditions, irrigation control, sensor networks, or solar-powered pumping. The gap that still needs to be answered is how to structure all of these components as an integrated architecture that can be understood, developed, and tested in stages. A system that only monitors water levels does not explain the energy consumption of the pump; systems that only use solar power do not necessarily provide remote control; While web applications without databases and device protection risk resulting in unreliable services. Therefore, a design is needed that connects data acquisition, processing and control logic, IoT communications, actuators, energy sources, web dashboards, and data storage.

The novelty of this design lies in the unification of the functions of monitoring water levels and electrical parameters, automatic and manual controls, protection mechanisms, web interfaces, and solar energy sources in one architectural model. Ultrasonic sensors are used to obtain water levels without direct contact with liquids, while PZEM004T are used to monitor the voltage, current, power, and energy of the pump. The ESP32 acts as a control and communication center, the relay connects the digital logic with the pump actuator, and the web dashboard provides operation information as well as control options. The database is designed to store sensor data, pump status changes, user activity, and notification logs so that the state of the system is not only visible in real-time, but can also be reviewed historically.

This research aims to design an IoT-based monitoring and control system for rice field irrigation water pumps and a web platform with solar energy sources. These objectives are described as the preparation of hardware and software architecture, the design of system workflows, the development of automatic and manual control logic, the preparation of protection mechanisms, the design of dashboard monitoring, and database modeling. The results of the research are limited to technical and logical design as the basis for implementation. This study does not present field performance claims, as sensor accuracy, energy efficiency, communication stability, and long-term reliability still have to be tested through physical prototypes and operation under real conditions.

## METHOD

The research uses a Research and Development (R&D) approach that is oriented towards designing technological products. The product in question is not a commercial device that has been tested, but an integrated system design that includes architecture, workflow, control logic, interface, and database. This approach was chosen because the research problem demanded the synthesis of several technical components and not testing the relationships between variables in respondents. The object of the research is the design of the irrigation pump monitoring and control system, while the analysis unit includes the functions of sensing, processing, communication, actuation, energy supply, user interface, and data management. There are no human subjects or collection of personal data at this stage of design.

The research data sources consist of scientific literature that discusses IoT, precision agriculture, smart irrigation systems, solar energy, and photovoltaic pumps; technical specifications of components; as well as functional needs derived from irrigation operation scenarios. The literature is used to identify the general principles of the system, while component specifications are used to ensure functional compatibility. Functional requirements are arranged based on the conditions that the system must serve, namely water level reading, monitoring of electrical parameters, automatic operation, manual control, shutdown of the pump in unsafe conditions, transmission of data to the IoT platform, visualization via the web, and storage of operation history.

The design instruments are in the form of a needs analysis sheet, function-component matrix, block diagrams, flowcharts, interface designs, and database schemas. Needs analysis sheets are used to distinguish key needs, protection needs, and information needs. A matrix of functions mapped out each need for a responsible device. Block charts are used to see the relationships between layers, while flowcharts are used to examine the order of decisions in auto, manual, and protection modes. The interface design shows the information that should be available to the user, and the database schema ensures that real-time data can be stored along with the logging time and identity of the control activity.

The main components of the hardware are ESP32, HC-SR04 ultrasonic sensor, PZEM004T module, relay, LCD, and water pump. The ESP32 was chosen because it provides processing capabilities, digital interfaces, and wireless connectivity in a single control board. The HC-SR04 is designed to measure the distance between the sensor and the water surface; The distance value is then converted to a level based on the height of the reservoir. PZEM004T functions to read voltage, current, active power, and cumulative energy on the load side. The relay acts as a switch that receives commands from the ESP32 and connects or disconnects the pump, while the LCD provides a local display when web access is not available.

The solar power supply chain consists of solar panels, solar charge controllers, batteries, and DC-AC inverters. The panel converts solar radiation into electrical energy, the charge controller regulates the charging process and protects the battery from unsuitable charging conditions, the battery stores energy so that the system can operate when the intensity of the sun fluctuates, and the inverter supplies AC power to the pump. At the implementation stage, the capacity of each component should be calculated based on the pump power, motor initial current, daily operating time, conversion losses, allowable depth of battery discharge, and margin for weather conditions. This research places the energy chain as an integral part of the system, rather than as an external resource that is assumed to always be available.

The software is designed in three layers. The embedded layer on the ESP32 runs sensor reading, data conversion, boundary evaluation, relay control, mode management, and data transmission. The IoT service layer receives data from the device, forwards commands, and maintains state synchronization. The web application layer presents dashboards, graphs, notifications, and user controls. In order for the status displayed to be no different from the physical condition, a manual command must be followed by confirmation from the ESP32 after the relay has changed. The same principle applies to protection: when an unsafe condition is detected, the local device must be able to stop the pump without waiting for a server response.

Data collection at the design stage is carried out through document review and specification extraction. Each source is read to identify relevant functions, usage limits, communication needs, and potential failures. The information is recorded in a needs matrix and used to construct an operation scenario. Normal scenarios include sensor readings, pump start-up at minimum level, shutdown at maximum level, manual control, and data storage. Abnormal scenarios include critical water levels, power exceeding safe limits, connection loss, invalid sensor data, and commands that conflict with protection. In this way, the design not only describes the normal flow, but also the basic response to interference.

Data analysis uses descriptive-functional analysis. The identified needs are mapped to components and processes, then checked for consistency through block diagrams and flowcharts. Logical analysis is done by tracing each scenario from sensor inputs to actuator outputs and web views. The design is considered adequate if each function has a clear data source, decision process, output, and recording mechanism; there are no orders that go beyond protection; and the flow of information can be traced from the device to the database. Because the study did not include field prototypes, no statistical analysis, actual efficiency measurements, or device endurance tests were conducted.

**Research Stages**

The stages of the research are summarized in Figure 1. The process begins with a literature review to identify the concepts of IoT, irrigation, solar-powered pumping, and monitoring needs; followed by designing a system to establish the architecture and data flow; hardware design to select sensors, controllers, relays, LCDs, pumps, and solar PV components; software design to compile automatic, manual, and protective algorithms; and designing a monitoring interface to define real-time information, graphs, notifications, and user controls. This sequence is iterative. If a function cannot be fulfilled by the selected component or flow, the design is returned to the previous stage to be adjusted.

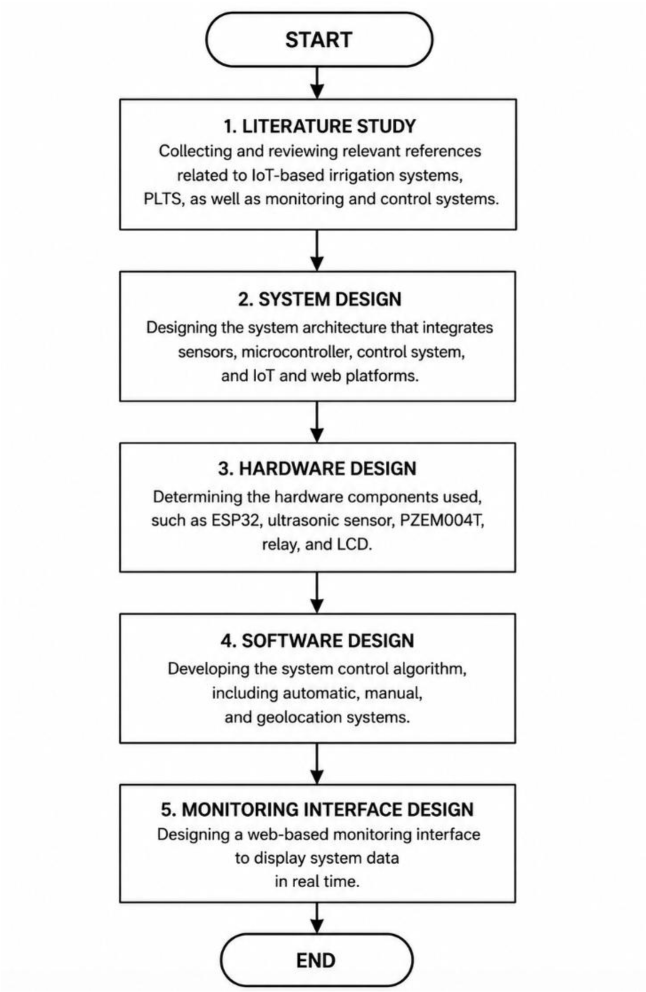


**Figure 1. Research Stage Flowchart**

Figure 1 shows that the final result of each stage is input for the next stage. Literature studies generate a need; needs translated into architecture; architecture defines components; components limit the choice of algorithms; and the algorithm determines the data that should be displayed on the interface. These relationships are important so that dashboards don't display variables that aren't actually measured, databases don't store data aimlessly, and web controls don't send commands that can't be executed securely by on-premises devices.

## RESULTS AND DISCUSSION

### System Architecture

The main results of the research are in the form of an integrated architecture that divides the system into six parts: data acquisition, processing and control, IoT communication, web monitoring and control, actuators, and solar energy sources. This division aims to separate the responsibilities of each section without cutting off the flow of information. Data acquisition produces a picture of physical conditions; processing and control translating data into decisions; communication connects devices with services; web applications present information and receive commands; the actuator executes the results; and solar power plants provide energy. The layered structure makes it easy to develop because changes to the dashboard, for example, don't have to change the sensor array as long as the data format remains consistent.

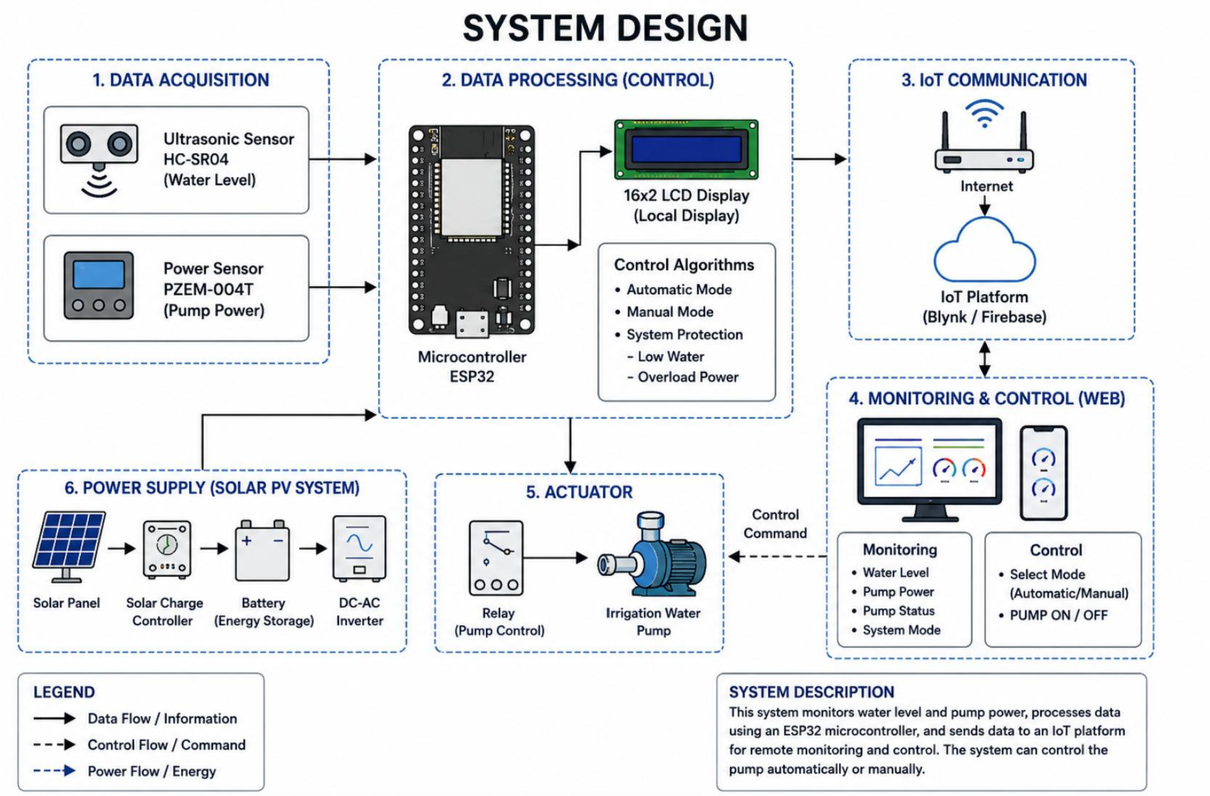


**Figure 2. Design of IoT-Based Water Pump Monitoring and Control System and Solar Power Plant**

Figure 2 shows that ultrasonic and PZEM004T sensors are the main data sources. Ultrasonic sensors provide water level data, while PZEM004T provide voltage, current, power, and energy data. The separation of the two types of data allows the system to answer two different questions: whether water needs to be pumped and whether the pump is operating under acceptable electrical conditions. If only the water level is measured, the pump can be switched on without information regarding the load; In contrast, electricity monitoring alone does not provide a basis for determining water replenishment needs. The integration of the two strengthens the control and early diagnosis functions.

The ESP32 receives sensor data, performs validation, executes control algorithms, updates LCDs, controls relays, and communicates with IoT platforms. This local role is important because safety decisions should not be entirely dependent on the internet. When the network is disconnected, the system must still be able to read the water level and turn off the pump at critical conditions. Such architecture is in line with the principle of edge control, which is that decisions that require quick responses are placed close to physical processes, while servers are used for monitoring, storage, and user interaction. This approach reduces the risk that a loss of connectivity causes the pump to remain active unattended.

On the energy side, solar panels, charge controllers, batteries, and inverters form a separate supply line from the data line, but their status affects operations. The current design monitors the pump load through the PZEM004T; Further development can add measurements of battery voltage, charging current, and panel energy to make the energy balance more complete. The use of inverters allows the AC pump to be used, but adds to the conversion losses and the need to handle the initial current of the motor. For new applications, the selection of a suitable DC pump can be considered to make the conversion path simpler. The final choice should be determined through load analysis and component availability in the field.

Compared to conventional systems that only use local switches, this architecture provides better traceability. Each change can be attributed to sensor data, time, mode of operation, and command source. However, traceability will only be valuable if device time is synchronized, data is not lost when the connection is lost, and the message format is consistent. Therefore, the implementation needs to provide a local queue or temporary storage mechanism, data quality markers, and a resubmission process. This has not been visualized in detail on the diagram, but is a technical consequence of the real-time and historical monitoring objectives.

**System Workflow**

The system workflow integrates condition reading, processing, control, communication, and display. When the system is activated, the ESP32 initializes the sensors, relays, LCDs, as well as connections to the IoT platform. After initialization, the device reads the water level and electrical parameters at regular intervals. The raw value should be checked before use. Ultrasonic readings that are outside the body's geometric range, too abrupt changes, or communication failures PZEM004T need to be marked as invalid data. Without validation, a single incorrect reading can trigger an unnecessary start or stop of the pump.

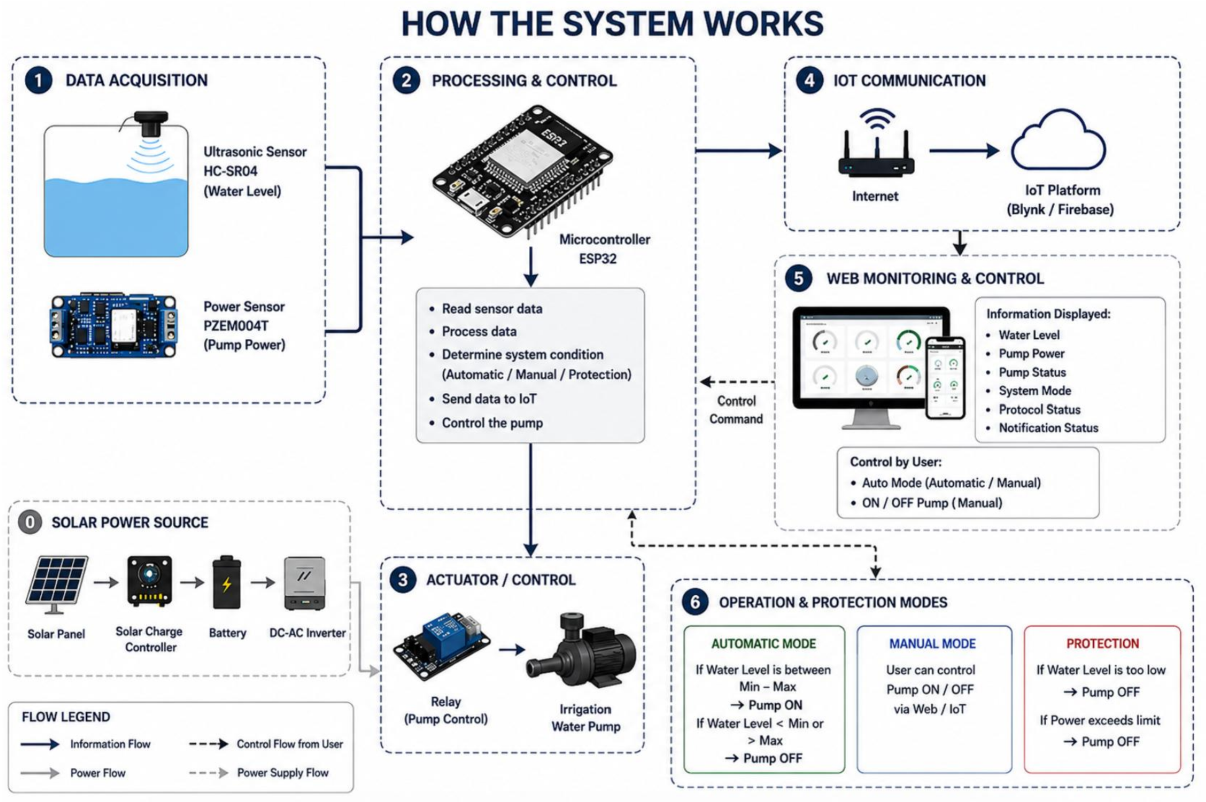


**Figure 3. How the System Works Flow**

Figure 3 shows the repetitive relationship between data acquisition and control processes. The sensor sends data to the ESP32, then the controller determines the system condition and passes the information to the IoT platform. At the same time, commands from the dashboard can be returned to the device. This two-way flow requires that each message has an identity, time, and execution state. An ON button on the web is not enough to be considered successful when clicked; The dashboard should receive confirmation that the relay is fully active and that Protection is not blocking commands. Such confirmation prevents a discrepancy between the interface status and the physical state of the pump.

The reading cycle needs to be adjusted to the dynamics of the system. The water level in the reservoir generally changes more slowly than the electrical signal, so the reading and delivery intervals can be distinguished. Electrical data can be read more frequently to detect spikes, while data to servers can be summarized to keep network usage and database capacity efficient. The separation between sampling, control evaluation, and logging helps the system respond quickly without having to store every raw value. At the implementation stage, the exact interval must be determined through testing so that important changes are not missed and the computational load remains low.

Workflows should also consider priorities. Protection has the highest priority, followed by automatic or manual control, then display needs. If the power exceeds the safe limit or the water level enters a critical condition, the pump must be stopped even if the user sends the ON command. Once the protection is active, the system sends a notification and records the cause of the shutdown. Reactivation should require that conditions have been normal for a certain period of time or require user acknowledgment. The mechanism prevents the pump from repeatedly turning on and off when the value is right around the limit.

**Pump Control Design**

The control design provides automatic mode and manual mode. Automatic mode is used to keep the water level within the set range. When the level drops below the minimum limit, the ESP32 activates the relay and the pump begins to fill or deliver water. When the level reaches the maximum limit, the relay is turned off. The use of two limits forms a simple hysteresis so that the pump does not change state whenever the value fluctuates slightly around one threshold. The limit range needs to be adjusted to the dimensions of the tub, pump discharge, irrigation needs, and sensor reading delay time.

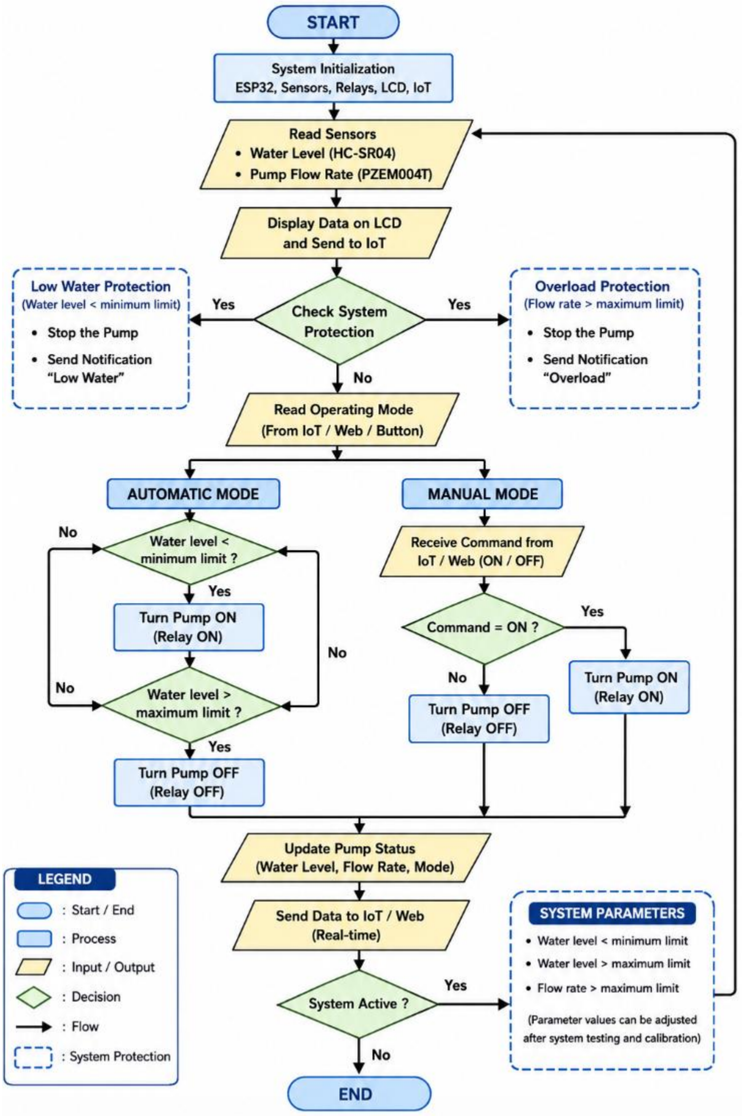


**Figure 4. Flowchart Pump Control System**

Figure 4 places a protection check before mode selection. This order is correct because the safety of the device must apply in both automatic and manual modes. Low water protection is intended to prevent the pump from running without an adequate water supply, while overpower protection prevents operation from being resumed when the load shows abnormal conditions. In implementation, the power limit should not be determined from the nominal power alone. Initial current, voltage variations, power factors, and sensor tolerances must be taken into account so that the system does not disconnect the pump under normal conditions or respond late to an outage.

Manual mode provides flexibility when operators need to perform tests, maintenance, or operations that are not on an automated schedule. Although it is called manual, the command still passes ESP32 and local protection. Thus, manual means that the user determines the state of operation, rather than deactivating the entire safety mechanism. The dashboard needs to clearly display the active mode and restrict control access to authorized users. Mode changes should also be noted as they affect the interpretation of the pump's status history.

Automatic controls that only use water levels are relatively simple and easy to understand, but do not include plant water needs, soil moisture, precipitation, or weather forecasts. Khan et al. (2020) show that soil and weather parameters can enrich irrigation decisions. Therefore, this design can be seen as a foundation that focuses on water availability and pump operations. Subsequent developments may add soil moisture sensors, rain gauges, or weather data, but the addition of variables must be accompanied by an evaluation of the benefits, costs, calibration needs, and complexity of the algorithm.

The reliability of the control is also affected by the quality of the installation. The ultrasonic sensor must be installed perpendicular to the surface of the water and protected from splashes, dew and obstructions. The relay must have a contact and isolation capacity that corresponds to the load of the motor, while the high-voltage line must be separated from the logic circuit. The system requires additional electrical protection such as a fuse or miniature circuit breaker, grounding, an environmentally appropriate enclosure, and a secure wiring. Logical design cannot replace such physical electrical protection.

**Web-Based Monitoring Design**

The web dashboard is designed as a supervision center that can be accessed from different locations. Key information is placed on a compact card that displays the water level, pump power, pump status, and operating mode. The presentation helps users get a quick overview before looking at the details. Voltage, current, power, and energy parameters are displayed separately so that changes in electrical conditions can be distinguished. Historical graphs are used to see water level and voltage patterns, while the notification panel displays conditions that require attention.

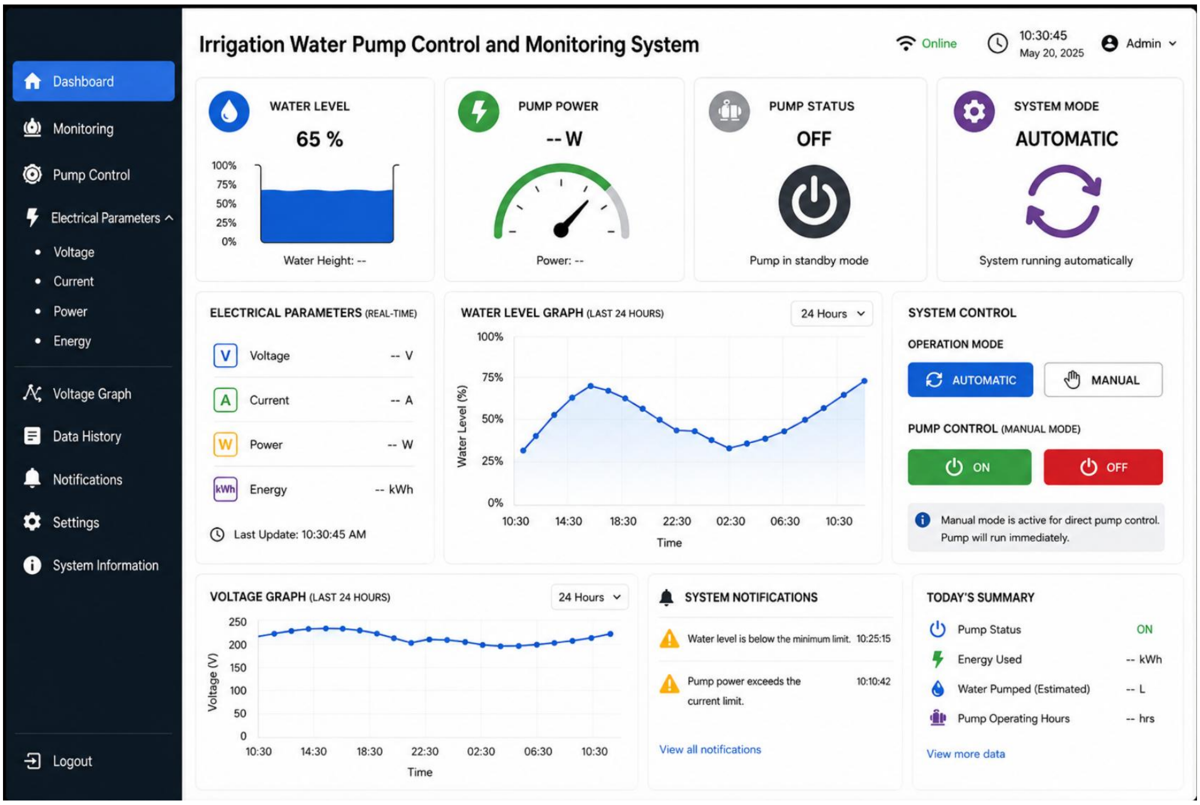


**Figure 5. Water Pump Monitoring and Control Dashboard Design**

Figure 5 shows that the monitoring and control functions are placed in a single interface. This integration shortens the user's workflow, but requires an interaction design that prevents errors. The ON and OFF buttons should indicate the last confirmed status of the device, not just the last command sent. Commands that impact the pump can be equipped with confirmation dialogs, process indicators, and rejection messages if protection is active. Color can help differentiate status, but important information also needs to be conveyed through text or icons to remain legible on a monochrome display and accessible to users with limited color perception.

Historical graphs provide context that is not available on real-time values. A pattern of water level drop can indicate the rate of discharge, while the duration of pump operation can be attributed to the energy used. When the data is stored consistently, users can compare operations between days, detect longer charging times, and identify load changes. However, the dashboard should not conclude damage from just one indicator. The analysis needs to consider a combination of levels, voltages, currents, power, operating time, and event records.

Notifications are designed for conditions where water levels are below the minimum limit, critical levels, power exceeds limits, communication failures, or invalid sensors. Each alert should have a priority level, time, trigger value, handling status, and links to related data. Too frequent alerts can cause alarm fatigue, so the system requires rules of event merge and recurrence intervals. Users should also be able to distinguish between informative notifications, warnings, and emergencies. Notification history is an important part of evaluating whether the boundaries used are too sensitive or less responsive.

Web access expands the surface of security risks. Authentication, session management, role restriction, communication encryption, and activity auditing should be implementation requirements. Passwords should not be stored in readable form, whereas communication between devices, servers, and browsers needs to use secure protocols. Accounts with control rights should be distinguished from those that only monitor. Database design provides role attributes, but security mechanisms must still be implemented at the application and service levels of IoT.

**Database Design**

The database is designed to maintain continuity between sensor data, pump status, control activity, and notifications. The structure consists of user, sensor_data, pump_control, and notification_log tables. Table separation prevents operational data from being mixed with account data and makes it easier to compile queries. Each record uses a time marker so that changes can be sorted. The relationship with the user is used in control activities and confirmed notifications, while sensor data can be stored without relying on the user's session.

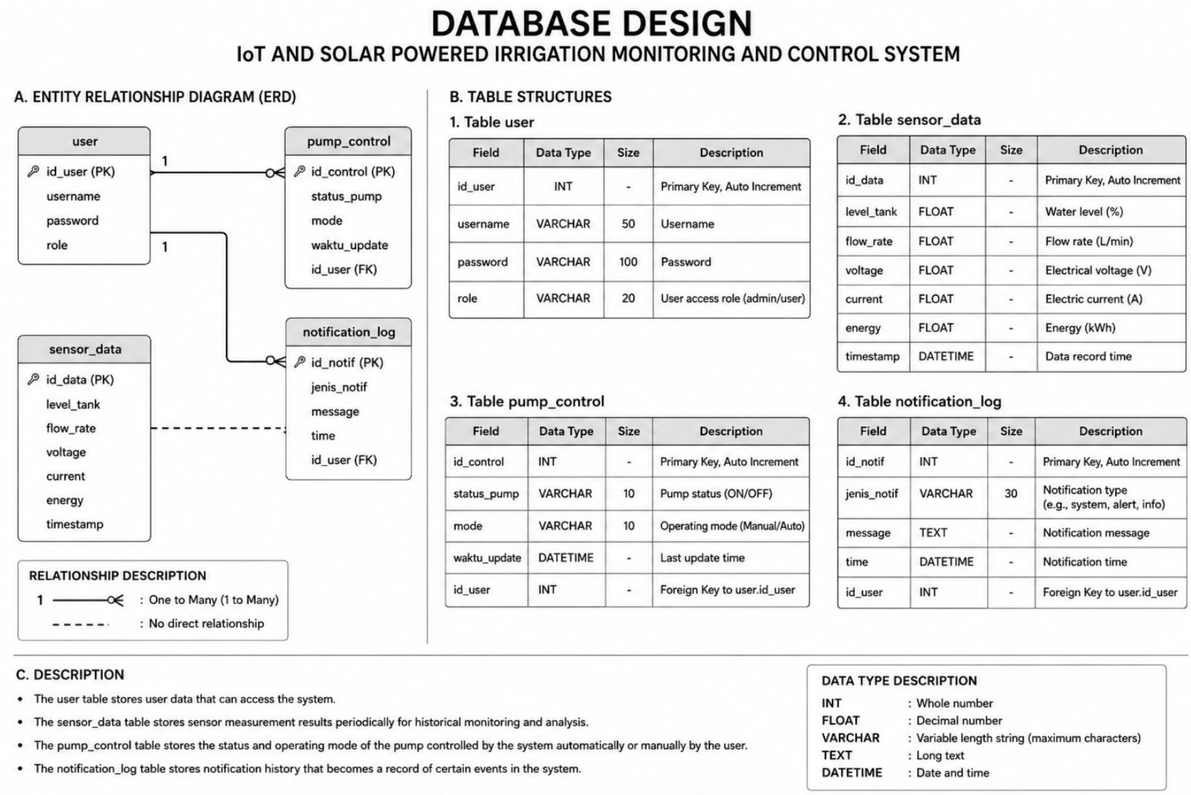


**Figure 6. Water Pump Monitoring and Control System Database Design**

Figure 6 shows that the user table stores account identities, usernames, passwords, and roles. On implementation, the password field should contain the corresponding algorithm-generated hash, not the original password. The table sensor_data store water levels, voltage, current, power, energy, and timestamps. This structure is simple and supports graphing, but the volume of data will increase with the logging frequency. Therefore, retention policies, indexes in the time column, and the possibility of aggregation of old data are needed. Invalid values should be stored with quality flags so that they can be distinguished from truly measurable zeros.

The pump_control table records the pump status, operating mode, change time, and the user who sent the command. This record allows for a reconstruction of the sequence of operations, especially when the status changes due to manual commands. To make it more complete, the table can be added to the source of changes, such as automatic, manual, protection, or recovery, as well as execution results. The notification_log table stores the notification type, message, time, and associated users. The addition of priority levels, trigger values, acknowledged status, and resolution times will support more systematic incident management.

Database relationships must maintain integrity without hindering automatic logging. If the account is deleted, the control history should be maintained with anonymized identities or references that were not deleted. Sensor data and logs also need to be protected from unauthorized alterations as they are used as operational proof. Periodic backups, access rights restrictions, and time synchronization are all part of data reliability. This design does not address specific replication or cloud services, so implementations can be tailored to the available infrastructure as long as the principles of integrity and traceability are maintained.

**Integrative Analysis and Limitations**

Integratively, the design addresses the basic needs of irrigation pump management: knowing water conditions, controlling the pump, monitoring the electrical load, providing alternative energy, and keeping records. IoT integration extends the functionality of devices from automatic switches to operational information systems. This is in line with the concept of industrial IoT connecting physical objects and digital services (Da Xu et al., 2014) as well as the utilization of sensors in precision agriculture (Nandurkar et al., 2014). Meanwhile, the use of solar power plants supports pumping applications in locations that require energy independence as discussed by Chandel et al. (2015).

The advantages of the design lie in the clear separation of functions, the existence of two modes of operation, priority protection, electrical monitoring, historical dashboard, and database. For users, the system has the potential to reduce the need for repetitive inspections and provide faster information. For managers, the history can be used to evaluate operations and plan maintenance. For developers, the layered architecture makes it easy to add sensors or features. However, these benefits are still potential because the design has not been tested on discharge, body size, network quality, weather, and real usage patterns.

The first limitation is that the sensor calibration results are not yet available. Ultrasonic readings can be affected by position, body shape, surface waves, temperature, and humidity. PZEM004T also has a measurement tolerance that must be compared to a reference tool. The second limitation is the lack of communication testing to measure latency, packet loss, and recovery after connection drops. The third limitation is that the calculation of solar PV capacity has not been carried out based on the actual load profile. Without such data, energy design cannot be declared sufficient for a given daily operation.

The next limitation has to do with environmental safety and resilience. Web and IoT systems require authentication testing, authorization, encryption, and resilience to repetitive commands. Field devices should be tested against heat, moisture, dust, water splashes, and electrical interference. In addition, the control rules need to be tested to ensure that no short cycling occurs and that the protection does not hinder normal operation. Field tests should be carried out in stages, starting from testing each sensor, testing relays without loads, testing pumps with controlled power sources, solar PV integration, communication testing, and finally operations on the ground.

Further research needs to measure water level accuracy, electrical parameter errors, control response time, dashboard stability, communication availability percentage, energy generated and used, pump operation duration, and protection success. Comparisons with conventional operation can include operator uptime, energy consumption, failure frequency, and water level consistency. The data allows for quantitative assessment and economic feasibility analysis. Thus, the design produced in this study serves as an initial specification that can be reduced to a scalable prototype and test protocol.

## CONCLUSION

The research resulted in an integrated design of an IoT-based irrigation pump monitoring and control system with solar energy sources. The design connects ultrasonic sensors for water level, PZEM004T for electrical parameters, ESP32 as a controller and communication device, relay as an actuator, LCD as a local display, IoT platform and web dashboard as a monitoring medium, as well as a database to store sensor data, pump status, control activities, and notifications. The system provides automatic modes based on minimum and maximum limits, manual modes over the web, and protection that has priority against operating commands. A layered structure that separates sensing, processing, communication, actuation, energy, application, and data management provides a clear foundation for implementation and development. The integration has the potential to increase the flexibility of irrigation management, traceability of operations, and the utilization of renewable energy. However, the results of the research are still in the design stage and have not proven field performance. Subsequent implementations should include sensor calibration, control and protection testing, communication latency measurement, application safety evaluation, solar power capacity calculation, as well as reliability tests under actual environmental conditions. The results of these tests are needed before the system can be considered effective, efficient, safe, and feasible to be widely applied to rice field irrigation management.

**REFERENCES**

Chandel, A., Naik, M. N., & Chandel, R. (2015). Review of solar photovoltaic water pumping system technology for irrigation and community drinking water supplies. Renewable and Sustainable Energy Reviews, 49, 1084–1099. https://doi.org/10.1016/j.rser.2015.04.083

Da Xu, L., He, W., & Li, S. (2014). Internet of Things in industries: A survey. IEEE Transactions on Industrial Informatics, 10(4), 2233–2243. https://doi.org/10.1109/TII.2014.2300753

Khan, M. A., Hossain, M. A., & Islam, M. R. (2020). IoT-based smart irrigation system using soil and weather monitoring. IEEE Access, 8, 115709–115720. https://doi.org/10.1109/ACCESS.2020.3003192

Kumar, S., Tiwari, A., & Zymbler, M. (2019). Internet of Things is a revolutionary approach for future technology enhancement: A review. Journal of Big Data, 6(1), 111. https://doi.org/10.1186/s40537-019-0268-2

Nandurkar, S. R., Thool, V. R., & Thool, R. C. (2014). Design and development of precision agriculture system using wireless sensor network. In Proceedings of the IEEE International Systems Conference (pp. 1–6). IEEE. https://doi.org/10.1109/SysCon.2014.6819262

Saxena, A. K., & Dutta, V. (2019). Solar energy and its future role in renewable energy. International Journal of Engineering Research & Technology, 8(5), 234–238.