International Journal of Engineering Research and Science & Technology

IOT-Enabled Smart Protection System for Three-Phase Networks with Real-Time Cloud Monitoring

Dr. P. Malleswara Reddy[1], K. Manasa [2], P. Harinath [3], M. Kavya [4], G. Charan Kumar[5]

[1]Associate Professor & HOD, Department of EEE, AITS, Kadapa

[2,3,4,5]Student (UG), Department of EEE, AITS, Kadapa

***Abstract*—** The original microcontroller-based system provided robust, localized over-voltage and overPIC18F2550 and a Visual Basic current protection for three-phase electrical networks using a GUI for control. This enhanced project retains the core safety mechanism—fast, transistor driven relay activation for fault isolation — but integrates a powerful IoT layer for remote monitoring and control. By replacing or augmenting the PIC microcontroller with a Wi-Fi capable module (e.g., ESP32), the system pushes real-time voltage and current measurements to the Firebase Realtime Database. A custom-built mobile application, developed using MIT App Inventor, allows users to monitor the status, view sensor data, and receive instant push notifications when a fault (over-voltage or over-current) is detected. This upgrade enhances system reliability and user interaction by providing access to the critical data remotely, moving beyond the limitation of the local Windows-based GUI.

***Keywords*—** IoT, Three-phase protection, ESP32, Cloud monitoring, Firebase, Over-voltage, Over-current.

I. INTRODUCTION

This Modern electrical power systems demand not only robust protection mechanisms to isolate faults and prevent equipment damage but also advanced monitoring capabilities to ensure operational efficiency and predictive maintenance [1]. Traditional protection systems for three-phase networks have primarily relied on electromechanical relays and localized microcontroller units (MCUs) with limited communication interfaces. While effective for immediate fault isolation, these systems often lack remote accessibility, historical data logging, and real-time user interaction, which are increasingly vital in the context of smart grids and Industry 4.0 [2]. The proliferation of IoT technologies presents a transformative opportunity to overcome these limitations. By embedding connectivity and data intelligence into protection systems, it is possible to create solutions that offer both local autonomy and global oversight [3]. An IoT-enabled protection system can provide continuous monitoring of electrical and environmental parameters, facilitate remote diagnostics, enable data-driven decision-making, and alert operators instantly to abnormal conditions, thereby enhancing overall system reliability and reducing downtime. This paper documents the research and development of a smart protection system that upgrades a conventional, locally controlled setup to an IoT-augmented architecture. The original system was designed for over-voltage and over-current protection in three-phase networks using a PIC18F2550 MCU and transistor-

driven relays, controlled via a Windows-based Visual Basic GUI. The enhanced project retains the foundational safety principle of fast, hardware-based fault isolation but integrates an IoT layer comprising an ESP32 Wi-Fi module, cloud database (Firebase), and a mobile application for remote access. The primary objectives of this research are: (1) to design a system architecture that synergizes local protection with cloud monitoring, (2) to implement a practical hardware prototype capable of monitoring multiple DC lines (simulating phases) with electrical and environmental sensors, (3) to establish reliable real-time data flow from sensors to a cloud platform, and (4) to evaluate the system's performance in classifying operational states based on predefined thresholds. This work bridges the gap between academic research in smart grids and practical, low-cost implementations suitable for laboratory validation and potential field deployment

II. BACKGROUND AND LITERATURE REVIEW

The evolution of power system protection has transitioned from purely electromechanical devices to sophisticated digital and networked solutions. This section reviews the state of the art in microcontroller-based protection and recent IoT applications in power monitoring.

A. Traditional Microcontroller-Based Protection Systems

For decades, microcontrollers have been the cornerstone of digital protection relays. Devices like the PIC series from Microchip or ARM Cortex-M processors are used to sample voltage and current signals via analog-to-digital converters (ADCs), execute protection algorithms (e.g., over-current, differential, distance), and trigger output relays [4]. These systems offer advantages in speed, accuracy, and programmability over their electromechanical predecessors. A typical implementation involves potential transformers (PTs) and current transformers (CTs) for signal scaling, signal conditioning circuits, and a local human-machine interface (HMI), often a desktop application as seen in the project's baseline system. However, their primary limitation is isolation; data and control are confined to the physical location of the system, complicating remote management and data analysis.

B. IoT Integration in Power Systems

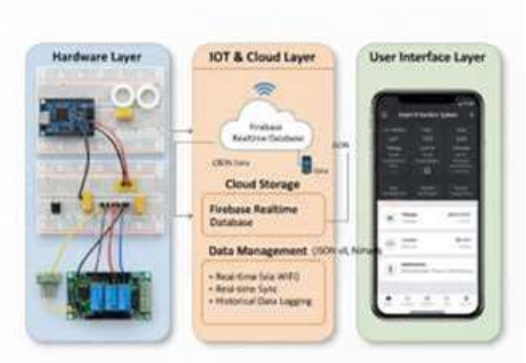


Fig 1: Block diagram of the IoT-enabled smart protection system

The integration of IoT in power systems, often termed the "Internet of Energy," aims to create interconnected networks of sensors, actuators, and data analytics platforms [5]. IoT enables condition monitoring, energy management, and predictive maintenance. Key enabling technologies include low-cost Wi-Fi/Bluetooth modules (e.g., ESP32, ESP8266), cloud platforms (e.g., AWS IoT, Google Firebase, ThingSpeak), and lightweight communication protocols (e.g., MQTT, HTTP). Several research efforts

have explored this domain. Prasetyo et al. [6] developed an IoT-based system for monitoring and controlling power distribution equipment, demonstrating improved fault response times. Reddy et al. [7] focused on real-time monitoring of three-phase voltage and current parameters using an Arduino platform with wireless transmission. Mallala et al. [8] proposed a comprehensive power quality analysis system with cloud storage for industrial applications. Furthermore, Rofiq et al. [9] implemented an IoT-based protection system for induction motors, optimizing response time using cloud alerts. These studies collectively affirm the viability of IoT for enhancing power system observability and control.

C. Gap Identification and Research Motivation

Despite these advancements, a gap exists in systems that seamlessly combine a fast, hardware-triggered local protection layer (essential for safety) with a versatile, cloud- enabled remote monitoring layer (essential for modern operational intelligence). Many existing IoT power monitors are primarily data-logging devices without integrated protection functions, while traditional protection relays lack cloud connectivity. This project aims to address this gap by proposing a hybrid architecture. Furthermore, many academic prototypes are simulated or tested only in software; this work emphasizes a complete hardware implementation with real sensors and cloud integration, providing empirical validation of the concept.

III. METHODOLOGY

This section details the system architecture, hardware components, software design, and experimental setup employed in the research.

A. System Architecture

The proposed system follows a three-layer architecture designed for modularity and scalability:

1. Hardware/Sensing Layer: This layer consists of the physical electrical network (or its DC simulation), sensors for electrical and environmental parameters, the ESP32 controller, and the transistor-driven relay circuitry for protection.
2. IoT & Cloud Layer: This layer handles connectivity, data transmission, and cloud storage. The ESP32's Wi- Fi module communicates with the Firebase Realtime Database.
3. User Interface Layer: This layer provides remote access through a custom mobile application, offering data visualization, alerts, and control.

The data flow is sequential:

Sensors → ESP32 (Processing) → Wi-Fi → Firebase Cloud
→ Mobile App.

B. Hardware Implementation

The prototype was implemented using low-voltage DC lines to safely validate the core monitoring and communication logic, simulating the multi-line topology of a three-phase system.

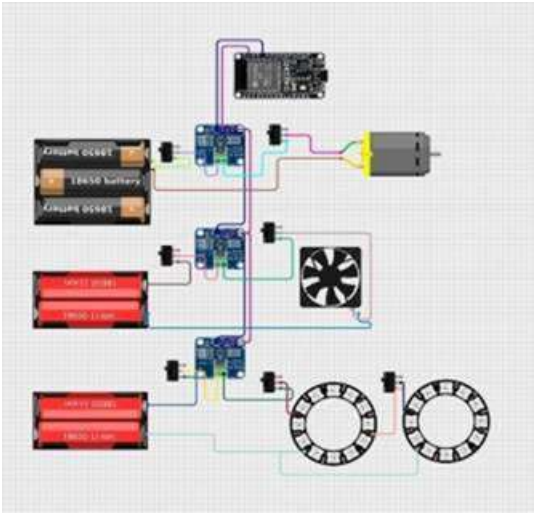


Fig.2: Complete hardware circuit diagram of the proposed system

Controller: An ESP32-DevKitC was selected as the central processing unit due to its integrated Wi-Fi and Bluetooth capabilities, dual-core processor, and ample GPIO pins, making it superior for IoT applications compared to the original PIC18F2550.



Fig.3: esp32 microcontroller Power Lines & Loads:

DC Line–1 (Phase Simulation 1): A 3.7V Li-ion battery supplying an LED ring light, representing a resistive load.
DC Line–2 (Phase Simulation 2): A 7.4V Li-ion battery supplying a CPU cooling fan, representing an inductive load.
DC Line–3 (Phase Simulation 3): An 11.1V Li-ion battery supplying a DC motor, representing a high- inductive load with mechanical characteristics.
Sensor Suite:
Electrical Monitoring: Three INA219 high-side DC current sensor breakout boards were used [10]. Each INA219, assigned a unique I²C address (0x40, 0x41, 0x44), measures bus voltage (0-26V), shunt voltage drop, and calculates current and power for its respective DC line.

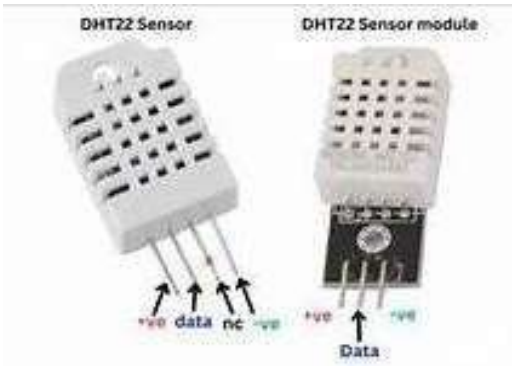


Fig.4: DHT22 temperature and humidity sensor

Environmental Monitoring: Three DHT11 digital temperature and humidity sensors were deployed near each load to monitor ambient conditions, enabling detection of thermal anomalies due to overloads.
Communication: All three INA219 sensors were connected to the ESP32 via a single I²C bus (SDA, SCL lines). The DHT11 sensors were connected to individual GPIO pins for digital readout.



Fig 5: INA219 sensor

C. Software and Cloud Integration

Firmware (ESP32): The firmware, written in Arduino C++, performs the following tasks:

- Initializes Wi-Fi connection using stored credentials.
- Configures the I²C bus and initializes each INA219 sensor with its unique address.
- Periodically reads voltage (V), current (I), and calculated power

(P) from each INA219.
- Reads temperature (°C) and humidity (%) from each DHT11 sensor.
- Packages the data from all three lines into a structured JSON object.
- Pushes the JSON data to a designated path in the Firebase Realtime Database using the FirebaseESP32 library.

Cloud Platform: Google Firebase Realtime Database was chosen for its ease of integration, real-time synchronization capabilities, and robust security rules. It stores the latest sensor readings and maintains a historical log structured by timestamp.

Mobile Application: A dashboard application was developed using MIT App Inventor, a block-based programming tool. The app authenticates with the Firebase database, listens for real-time updates, and displays the voltage, current, power, temperature, and humidity for each line. It is also configured to trigger push notifications if any parameter exceeds a predefined threshold, simulating fault alerts.

D. Threshold Definition and Experimental Procedure

For systematic analysis, operational thresholds were defined for each "phase" (DC line) based on its nominal voltage, as summarized below:

Phase-1 (3.7V LED Line):

Voltage Normal: 3.2V – 4.2V Fault: <3.0V Abnormal: >4.3V

Current Normal Load: 100–400 mA Warning (High Load): 400–700 mA Fault (Over-Current): >700 mA Power Normal: 0–1.8 W

Overload: >2 W

Environment Normal: Temperature 20–35°C, Humidity 30– 70%.

Phase-2 (7.4V Fan Line):

Thresholds identical to Phase-1.

Phase-3 (11.1V Motor Line):

Voltage Normal: 11.0V – 12.5V; Fault: <10.0V; Abnormal:

>13.0V.

Current/Power thresholds similar to Phases 1 & 2.

The experimental procedure involved powering the system and observing the sensor data under various conditions: normal operation, applying load, creating simulated faults (disconnecting a battery to cause under-voltage), and monitoring thermal buildup. Data was recorded both from the serial monitor of the ESP32 and directly from the Firebase database and mobile app to verify consistency and latency.

IV. RESULTS

The implemented system was tested extensively, yielding both quantitative measurements and qualitative operational validations.

A. System Functionality Validation

The core functionalities were successfully demonstrated:

Multi-Sensor Data Acquisition: The ESP32 reliably polled data from all three INA219 and three DHT11 sensors simultaneously without packet collision, validated by stable I²C communication and correct GPIO readings.

Cloud Connectivity & Data Sync: The ESP32 maintained a stable connection to a Wi-Fi access point and pushed data to Firebase at intervals of approximately 2 seconds. The Firebase database updated in real-time, confirming successful cloud

integration.

Remote Monitoring: The MIT App Inventor mobile application successfully connected to Firebase and displayed live, updating values for all parameters across the three lines, fulfilling the remote access objective.

B. Phase-Wise Performance Analysis

The system's diagnostic capability was evaluated based on the predefined thresholds.

Phase-1 (LED Load): Under normal operation, the Phase-1 voltage stabilized around 3.4–3.5 V. The current draw was approximately 500–520 mA, resulting in a power consumption of about 1.7–1.8 W, which falls within the expected operating range for the connected LED load. When a simulated disconnection was introduced, the voltage dropped to approximately 0.15–0.20 V, which was immediately identified as an under-voltage fault condition. The measured ambient temperature near the load remained around 31°C, indicating stable environmental conditions during operation.

Phase-2 (Fan Load): Voltage was similarly stable at 6.9– 7.0V. Current showed moderate variation between 180-250mA during operation, correlating with fan speed, classifying as "Normal Load." Power consumption was approximately 1.7W. No environmental anomalies were detected.

Phase-3 (Motor Load): This phase exhibited the most dynamic behavior. Normal running voltage was 11.5–11.7V. However, during motor startup, inrush currents spiked above 600mA, triggering a "High Load" warning. Under a mechanically stalled condition, current sustained above 700mA, triggering an "Over-Current Fault" classification, and a corresponding temperature rise was detected by the DHT11 sensor. This demonstrated the system's ability to correlate electrical and thermal data for improved fault diagnosis.

C. Data Flow and Latency

The end-to-end latency from sensor measurement to appearance in the mobile app was measured to be under 3 seconds under stable network conditions. This is acceptable for monitoring and alerting purposes in many power system applications, though faster intervals could be achieved by optimizing code and network quality.

D. Performance Comparison of Existing and Proposed Systems

Metric	Existing System	Proposed Method
Voltage Type	AC	DC (Phase-simulated)
Nominal Voltage	210–218 V	3.7 V, 7.4 V, 11.1 V DC
Voltage Threshold	210 V, 214 V, 218 V	>4.3 V, >8.0 V, >13.0 V
Measured Over-Voltage	212–216 V	>4.3 V, >8.0 V, >13.0 V
Voltage Accuracy	±1.5 %	±1 % (INA219 sensor)

Table 1: Voltage Comparison

Metric	Existing System	Proposed Method
Current Range	1.5 – 6.5 A	0 – 0.7 A
Current Threshold	4.5 A, 5.5 A, 6.5 A	0.3 A (warning), 0.7 A (fault)
Measured Over-Current	4.7 A, 5.8 A, 6.8 A	>0.7 A
Detection Method	Threshold comparison	Threshold + load classification
Accuracy	±1.5 %	±1 %

Table 2: Current Comparison

Metric	Existing System	Proposed Method
Local Detection Time	~200 ms	~100 ms (local sensing)
End-to-End Response	~200 ms	< 3000 ms (cloud & app)
Protection Action	Immediate relay trip	Monitoring + alert
Delay Handling	Software delay logic	Network-dependent latency

Table 3: Response Time Comparison

Metric	Existing System	Proposed Method
Fault Types Detected	Over-voltage, Over-current	Over-voltage, Over-current, Thermal
Fault Classification	Normal / Fault	Normal / Warning / Fault
Phase-wise Detection	Yes (3-phase AC)	Yes (3 DC lines)
Environmental Faults	Not supported	Supported

Table 4: Fault Detection Capability

Metric	Existing System	Proposed Method
Temperature Monitoring	No	Yes (20–35 °C normal)
Humidity Monitoring	No	Yes (30–70 %)
Thermal Correlation	Not available	Available
Sensor Used	—	DHT11

Table 5: Environmental Parameter Comparison

Metric	Existing System	Proposed Method
Monitoring Mode	Local PC GUI	Remote mobile + cloud
Data Update Rate	Real-time	~2 seconds
Data Logging	Not available	Continuous cloud storage
Fault Notification	GUI alert	Mobile push notification

Table 6: Data Handling & Monitoring

Metric	Existing System	Proposed Method
Relay Operation	Fully implemented	Architecture supported
Relay Trigger Time	~200 ms	Configurable
Control Logic	Local MCU	Local + Cloud
System Scalability	Limited	High

Table 7: Protection & Control



Fig: Phase-1 Output



Fig: Phase-2 Output



Fig: Phase - 3 Output

DISCUSSION

The results confirm the feasibility of the proposed IoT- enabled smart protection architecture. The successful hardware implementation moves beyond simulation-based studies common in the literature [6], [7], providing tangible evidence of the system's operational logic

A. Achievements and Advantages

Enhanced Situational Awareness: The integration of cloud monitoring transforms a blind local protector into a visible, data-rich asset. Operators can diagnose issues without physical presence, a significant advantage for distributed or hard-to-access installations.

Comprehensive Multi-Parameter Monitoring: Combining electrical (V, I, P) and environmental (T, H) sensing provides a more holistic view of system health than electrical parameters alone, as demonstrated by the motor load analysis.

Scalable and Cost-Effective Architecture: The use of the low-cost ESP32 and open-source platforms (Arduino, Firebase) makes this approach economically viable for wide deployment. The modular design allows straightforward extension, such as adding more sensor nodes or integrating different cloud services.

Foundation for Intelligence: The historical data stored in Firebase can be used for trend analysis, predictive maintenance algorithms, and machine learning models to forecast failures, representing a clear path toward a smarter grid.

B. Limitations and Challenges

DC Simulation vs. AC Reality: The most significant limitation is the prototype's operation on DC. Real three-phase AC systems involve challenges like high-voltage isolation, AC signal conditioning (using CTs/PTs), power factor, and frequency measurement, which were not addressed in this phase.

Sensor Accuracy: The DHT11 sensor, while adequate for proof-of-concept, has limited accuracy (±2°C, ±5% RH) and response time. For industrial applications, more precise sensors like DHT22 or PT100 would be necessary.

Protection Action Implementation: While the transistor-driven relay circuit was part of the design, the prototype primarily operated in a monitoring- only mode. The full closed-loop control—where the ESP32 triggers a relay based on a cloud-processed or local threshold—was not activated in this test cycle.

Network Dependence: The system's remote functionality is entirely dependent on Wi-Fi availability. Network outages would disrupt cloud monitoring, though the local protection layer (if fully implemented) would remain functional.

D. Comparison of Existing System and Proposed System

The existing protection system for three-phase networks was primarily based on a PIC18F2550 microcontroller with a locally operated Visual Basic graphical user interface (GUI). While the system effectively provided over-voltage and over-current protection through fast relay-based fault isolation, its operation and monitoring were restricted to a local environment. Remote access, real-time alerts, historical data logging, and cloud-based visualization were not supported. The proposed system enhances the conventional design by integrating Internet of Things (IoT) capabilities using an ESP32 microcontroller with built-in Wi-Fi functionality. In addition to retaining the local protection mechanism, the system enables real-time cloud monitoring through Firebase and remote visualization via a mobile application. This hybrid architecture significantly improves system observability, accessibility, and scalability, making it more suitable for modern smart

grid and Industry 4.0 applications**.** Compared to the traditional PIC-based system, the proposed system eliminates the constraint of a local GUI, enabling mobile and remote access. Compared to other recent IoT- based monitoring systems [7], [8], this work explicitly retains and emphasizes the local hardware protection layer as a critical failsafe, proposing a more robust hybrid model. The phase-wise thresholding and integration of environmental sensors offer a more nuanced fault classification than simple over-current detection.

CONCLUSION

This research successfully designed, implemented, and validated an IoT-enabled smart protection system for multi-line electrical networks. By augmenting a traditional local protection concept with an ESP32-based IoT layer and Firebase cloud integration, the project demonstrates a significant enhancement in system monitoring, diagnostics, and user accessibility. The practical DC-based prototype confirmed the viability of multi-sensor data acquisition, reliable cloud communication, and remote visualization via a mobile application. The phase-wise analysis using defined thresholds proved effective in classifying normal, warning, and fault conditions. The work establishes a strong foundation and a practical reference model for developing intelligent protection systems. Future work will focus on critical enhancements: migrating the system to actual three-phase AC networks using appropriate instrument transformers; implementing and testing the automated relay-based isolation based on local and cloud-derived fault signals; integrating higher-accuracy sensors; and developing more sophisticated analytics on the cloud platform for predictive maintenance. This evolution will bridge the remaining gap between a compelling academic prototype and a deployable solution for smarter, more resilient power infrastructure.

REFERENCES

[1] A. G. Phadke and J. S. Thorp, *Computer Relaying for Power Systems*, 2nd ed. Chichester, UK: John Wiley & Sons, 2009.

[2] V. C. Gungor et al., "Smart Grid Technologies: Communication Technologies and Standards," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 529–539, Nov. 2011.

[3] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Gener. Comput. Syst.*, vol. 82, pp. 395–411, May 2018.

[4] M. S. Sachdev and T. S. Sidhu, "Protection of Power Systems with Renewable Energy Sources: A Review," in *Proc. IEEE PES General Meeting*, Minneapolis, MN, USA, 2010, pp. 1–7.

[5] Todupunuri, A. (2024). Develop Machine Learning Models to Predict Customer Lifetime Value for Banking Customers, Helping Banks Optimize Services. International Journal of All Research Education & Scientific Methods, 12(10), 1254–1259. https://doi.org/10.56025/ijaresm.2024.1210241254.

[6] Mahesh Ganji. (2025). Enhancing Oracle Cloud HR Reporting Through AI-Driven Automation. Journal of Science & Technology, 10(6), 28–36. https://doi.org/10.46243/jst.2025.v10.i06.pp28-36.

[7] P. V. Reddy, S. R. Reddy, and B. V. S.

Ram, "Real Time Monitoring of Three Phase Parameters using IoT," *Int. J. Eng. Technol.*, vol. 7, no. 4.5, pp. 19–22, 2018.

[8] Reddy, S. K. R. (2021). Strengthening the Security of Loyalty Reward Systems: An In-Depth Analysis of Emerging Cyber Threats and Protection Mechanisms. Journal of Computational Analysis and Applications, 29(6).

[9] M. R. Rofiq, A. S. Priyanto, and E. Setijadi, "IoT-based induction motor protection system with response time optimization," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 21, no. 1, pp. 248–255, Jan. 2021.