IoT-Enabled Smart Transformer with Advanced Thermal & Load Management System

Yalamanchili Chandra Sekhar[1], Mr. V. Vasudeva Rao[2]

[1]PG Scholar, Department of Mechanical Engineering (Thermal Engineering)
Visakha Institute of Engineering & Technology (Autonomous),
Visakhapatnam, Andhra Pradesh – 530027, India

[2]Assistant Professor, Department of Mechanical Engineering (Thermal Engineering)
Visakha Institute of Engineering & Technology (Autonomous),
Visakhapatnam, Andhra Pradesh – 530027, India

Abstract—Transformers form the backbone of electrical power transmission and distribution networks, and their continuous, unmonitored operation under thermal stress and load variation often leads to insulation degradation, overheating, and unplanned outages. Conventional monitoring relies on periodic manual inspection, which is labour-intensive, slow to detect incipient faults, and unsuitable for real-time protection. This paper presents the design and implementation of an IoT-enabled smart transformer monitoring and protection system that continuously acquires temperature, humidity, voltage, and current data using a DHT11 sensor, an ACS712 current sensor, and a ZMPT101B voltage sensor interfaced with an ESP32 microcontroller. The ESP32 compares the acquired parameters against predefined safety thresholds (45°C for temperature and 2 A for current) and automatically actuates a relay-controlled DC cooling fan or load-disconnect mechanism whenever an abnormal condition is detected. Real-time parameter values are published to an embedded Wi-Fi web server, enabling remote supervision from any browser-enabled device. The complete system was designed and simulated using Cirkit Designer, implemented in the Arduino IDE, and validated on hardware. Experimental results confirm accurate sensing, timely automatic protection, and reliable wireless reporting, demonstrating that the proposed low-cost architecture significantly improves transformer safety, reduces maintenance overhead, and enhances the reliability of power distribution networks.

Keywords—IoT, Smart Transformer, ESP32, Thermal Monitoring, Load Management, ACS712, ZMPT101B, DHT11, Relay Protection, Remote Monitoring

1. Introduction

Transformers are critical assets in power transmission and distribution networks, and their continuous, reliable operation is essential to maintaining uninterrupted electricity supply. Prolonged exposure to overload, overheating, and voltage fluctuation gradually damages transformer insulation and windings, and if left undetected, can lead to catastrophic failure and prolonged outages. Traditional transformer monitoring is performed through periodic manual inspection, which depends heavily on operator availability, is slow to identify incipient faults, and offers no automatic protective response.

Unplanned transformer outages carry substantial economic and operational costs: industrial consumers experience production downtime, sensitive installations risk loss of essential power, and utilities incur emergency repair and replacement expenses that typically exceed the cost of preventive monitoring many times over. Reliability studies of distribution networks consistently identify sustained overheating and overload as leading causes of premature transformer failure, both of which are, in principle, detectable well before catastrophic breakdown if temperature and current are observed continuously rather than at periodic inspection intervals. This observation motivates the shift from manual, interval-based inspection toward continuous, sensor-driven monitoring with automatic protective response, which forms the central objective of the present work.

The proliferation of low-cost microcontrollers with integrated wireless connectivity has enabled a new generation of Internet of Things (IoT) based condition-monitoring systems capable of continuous, autonomous supervision. This paper proposes an IoT-enabled smart transformer monitoring and protection system built around the ESP32 microcontroller. The system continuously acquires ambient temperature and humidity through a DHT11 sensor, load current through an ACS712 Hall-effect current sensor, and supply voltage through a ZMPT101B voltage-sensing module. The ESP32 processes these readings in real time, compares them against predefined safety thresholds, and automatically actuates a relay-driven DC cooling fan or load-protection mechanism whenever an abnormal condition is detected. The measured parameters are simultaneously published on an embedded Wi-Fi web server, allowing authorized personnel to remotely observe transformer status from any browser-enabled device.

Beyond simple threshold-based alerting, effective transformer protection requires the coordinated operation of sensing, decision-making, and actuation within a single response cycle short enough to prevent thermal or electrical damage. The proposed design achieves this by performing all three functions locally on the ESP32 itself, rather than depending on a remote server or cloud service to issue protective commands. This local-decision architecture ensures that cooling and load-disconnection actions continue to operate correctly even during temporary loss of internet connectivity, while the Wi-Fi web interface remains available for supervisory visibility whenever network access is present.

The main contributions of this work are threefold. First, a low-cost, self-contained hardware platform is developed that integrates thermal sensing, electrical sensing, and automatic protection within a single ESP32-based unit. Second, a threshold-based control algorithm is implemented and validated that reliably distinguishes between normal operation, over-temperature, overload, and combined-fault conditions. Third, the complete design flow, from circuit simulation in Cirkit Designer through firmware development in the Arduino IDE to hardware testing, is documented as a reproducible methodology that can be adapted to other transformer capacities and site conditions.

The remainder of this paper is organized as follows. Section II reviews related work on IoT-based transformer and power-system monitoring. Section III describes the limitations of existing monitoring practices. Section IV presents the proposed research methodology, including system architecture, hardware and software components, and implementation procedure. Section V discusses the experimental results. Section VI concludes the paper and outlines directions for future work.

2. Literature Survey

A wide range of research has investigated IoT-based approaches to transformer and power-system condition monitoring. Table I summarizes representative studies and highlights the sensing platforms used and the principal outcomes reported.

TABLE I. SUMMARY OF RELATED WORK

Study	Platform / Sensors	Key Outcome
Al Noman et al. (2025)	ESP32, ACS712, ZMPT101B, DHT11, Blynk IoT	Detected 8 fault types; oil temp. reached 52°C under fault
Rehman & Khan (2023)	IoT communication modules	Improved reliability and fault management in smart grids
Smart Load Share (2021)	Arduino, ULN2003 relay drivers, LCD	Backup engaged within 200 ms; load capped at 1 A
El-Desouky et al. (2019)	Arduino Uno, Hall-effect CT, IEC 60255-151	Accurate inverse-time overcurrent protection
PLOS ONE (2018)	IoT protection device, EN 61000 testing	Detected overvoltage, overcurrent, arc faults
WSN Monitoring (2017)	Distributed wireless sensor nodes	Reduced manual inspection; better maintenance planning
Smart Grid IoT (2020)	IoT-integrated grid devices	Improved visibility and fault response time
Cloud Monitoring (2022)	Cloud server architecture	Enabled predictive maintenance via remote access
ESP32 Industrial (2023)	ESP32, multi-sensor Wi-Fi link	Confirmed ESP32 as reliable, cost-effective IoT platform
AI Fault Prediction (2024)	IoT + Machine Learning	Improved maintenance scheduling; reduced downtime

Al Noman et al. developed an ESP32-based distribution transformer health monitoring system capable of detecting eight distinct fault conditions, including phase failure and elevated oil temperature. Rehman and Khan demonstrated that continuous IoT-based data collection improves reliability and fault management in smart distribution networks. The Smart Load Share study proposed an adaptive load-balancing scheme that engaged a backup transformer within 200 ms of an overload event, while El-Desouky et al. implemented an Arduino-based overcurrent relay conforming to IEC 60255-151 inverse-time characteristics.

Later studies extended these concepts toward cloud connectivity and predictive analytics. Cloud-based architectures enabled remote access to historical operating data and supported predictive maintenance strategies, while more recent work combined IoT sensor streams with machine-learning models to estimate fault probabilities ahead of failure. The wireless-sensor-network approach reported in 2017 demonstrated that distributed temperature and current nodes could reduce the frequency of manual inspection rounds, and the 2020 smart-grid study confirmed that IoT integration measurably improves fault response time compared with conventional SCADA polling intervals.

Collectively, this body of work confirms that ESP32-class microcontrollers, Hall-effect current sensors, and Wi-Fi connectivity form a mature, cost-effective foundation for transformer health monitoring. However, it also reveals a recurring gap: most existing implementations either focus narrowly on fault detection and alerting, or on remote data visualization, without closing the loop through an automatic, on-board protective action such as active cooling or load disconnection. Systems that do include protective relays, such as the overcurrent relay of El-Desouky et al., typically address only electrical faults and omit thermal management, while thermal-monitoring studies rarely include coordinated load protection. The present work is positioned to close this gap by integrating thermal sensing, electrical sensing, automatic dual-mode protection (cooling and load disconnection), and remote web-based reporting within a single, low-cost ESP32-centric platform, as detailed in Section IV.

3. Existing System

Conventional transformer monitoring in distribution substations and industrial installations relies predominantly on periodic manual inspection by maintenance personnel. Technicians physically visit the transformer site to record oil temperature, check for visible signs of overheating, and verify load readings using handheld meters.

This approach suffers from several limitations that motivate the development of automated alternatives.

First, manual inspection is inherently discontinuous; faults that develop between scheduled visits, such as sudden overload or a rapid temperature rise, remain undetected until the next inspection or until failure occurs. Second, the accuracy and consistency of manually recorded readings depend on the skill and diligence of the inspecting technician, introducing the possibility of human error. Third, manual methods provide no automatic protective response — corrective action such as load shedding or activating supplementary cooling can only be taken after an operator has observed and interpreted the abnormal reading, resulting in delayed response times. Fourth, manual inspection is costly in terms of labour and travel, particularly for geographically distributed substations, and does not scale well to large numbers of transformers.

Some existing electromechanical protection schemes use fixed thermal relays or fuses to provide basic overcurrent protection; however, these devices operate independently, offer no remote visibility, cannot distinguish between different fault types, and require on-site resetting after every trip. Existing SCADA-based monitoring systems used in large substations do provide continuous electrical monitoring, but their high hardware and installation costs make them impractical for small and medium distribution transformers. Table II contrasts the principal characteristics of manual/fixed-relay monitoring, SCADA-based monitoring, and the proposed IoT-enabled system.

TABLE II. EXISTING SYSTEMS vs. PROPOSED SYSTEM

Feature	Manual/Relay	SCADA	Proposed
Continuous monitoring	No	Yes	Yes
Automatic cooling/protection	Limited	Partial	Yes
Remote web access	No	Yes	Yes
Implementation cost	Low	High	Low
Response time	Slow	Fast	Fast
Scalability to small units	Poor	Poor	Good

These limitations collectively establish the need for a low-cost, continuously operating, IoT-enabled monitoring and protection system, which is addressed by the proposed methodology in Section IV.

4. Research Methodology

4.1 System Architecture

The proposed system follows a sense-process-actuate-communicate architecture. Temperature and humidity data from the DHT11 sensor, current data from the ACS712 sensor, and voltage data from the ZMPT101B module are continuously fed to the analog and digital input pins of the ESP32 microcontroller. The ESP32 samples each sensor periodically, converts analog readings into calibrated engineering units, and compares them against predefined safety thresholds. Whenever temperature or current exceeds the configured limit, the ESP32 drives a GPIO pin connected to a relay module, which switches on a DC cooling fan or disconnects the protected load. Concurrently, the ESP32 hosts a lightweight HTTP web server over its built-in Wi-Fi interface, publishing the latest sensor readings and fault flags so that authorized users can observe transformer status remotely through a standard web browser.

4.2 Hardware Components

Table III lists the principal hardware components used to realize the monitoring and protection unit, together with their function within the system.

TABLE III. HARDWARE COMPONENTS

Component	Function
ESP32 Dev. Board	Central controller; Wi-Fi/BLE, dual-core, ADC/GPIO
DHT11 Sensor	Ambient temperature & humidity acquisition
ACS712 Current Sensor	Hall-effect load current measurement

Component	Function
ZMPT101B Voltage Sensor	AC supply voltage measurement, isolated
Relay Module	Switches cooling fan / load on threshold breach
DC Cooling Fan (12V)	Active cooling when temperature > 45°C
SMPS Power Supply	230V AC to regulated low-voltage DC
7805 IC Regulator	Stable 5V DC rail for sensors/logic
LED Panel Board	Electrical load for performance evaluation

The ESP32 was selected as the central controller because of its dual-core processor, built-in Wi-Fi/Bluetooth connectivity, multiple ADC channels, and low power consumption, which together eliminate the need for a separate communication module. Table IV summarizes the electrical specifications of the three primary sensing elements.

TABLE IV. SENSOR SPECIFICATIONS

Sensor	Supply	Range	Accuracy/Output
DHT11	3.3–5 V	0–50°C, ±2°C	20–90% RH, ±5%
ACS712	5 V	±5/20/30 A	66–185 mV/A
ZMPT101B	3.3–5 V	0–250 V AC	Isolated analog out

Power for the ESP32 and sensor network is supplied by a Switch-Mode Power Supply (SMPS) that converts the 230 V AC mains into a regulated low-voltage DC output through input protection, EMI filtering, high-frequency MOSFET switching, and flyback transformation stages, followed by a 7805 linear regulator that provides a stable 5 V rail. An LED panel board (120 SMD 5730 LEDs, 24 V DC) is used as the electrical load during testing, allowing the voltage and current sensors to be evaluated under a realistic, controllable load condition.

4.3 Threshold-Based Protection Logic

The ESP32 firmware continuously evaluates the acquired parameters against the safety thresholds summarized in Table V. When a threshold is exceeded, the corresponding protective action is triggered automatically without operator intervention, and the fault condition is simultaneously flagged on the web interface.

TABLE V. PROTECTION THRESHOLDS AND ACTIONS

Parameter	Threshold	Automatic Action
Temperature	45°C	Relay ON – cooling fan activated
Load Current	2 A	Relay ON – load disconnect / alert
Voltage	Nominal ±10%	Fluctuation alert generated

4.4 Software Tools and Circuit Simulation

The circuit was first designed and validated in Cirkit Designer, a graphical circuit-design and simulation environment, prior to physical assembly. The ESP32, DHT11, ACS712, ZMPT101B, relay module, cooling fan, and LED indicators were placed on the simulation canvas and interconnected according to the system wiring diagram. Simulation was used to verify power, ground, digital, and analog connections, and to confirm correct relay switching behaviour before hardware implementation, reducing wiring errors and development time.

Firmware development was carried out in the Arduino IDE. After installing the ESP32 board package and required libraries (WiFi.h and DHT.h), the program was written to perform sensor acquisition, threshold comparison, relay actuation, and web-server hosting. The code was verified (compiled) to check for syntax errors, then uploaded to the ESP32 over a USB connection, and its behaviour was confirmed using the Serial Monitor before final hardware deployment. Prior to loading the final application, the built-in Blink example was executed and its delay parameter modified to confirm that the toolchain, USB driver, and board configuration were functioning correctly — a standard bring-up step that reduces debugging effort once the full sensor-driven firmware is loaded.

The core control logic implemented on the ESP32 follows a simple, deterministic loop: (i) read temperature and humidity from the DHT11; (ii) sample the analog outputs of the ACS712 and ZMPT101B and convert them to calibrated current and voltage values; (iii) compare temperature against the 45°C limit and current against the 2 A limit; (iv) drive the relay GPIO high if either limit is exceeded, otherwise keep it low; and (v) serve an updated HTML page summarizing all readings and any active alerts to connected web clients. Because this loop executes entirely on the microcontroller, the protective response time is bounded by the sensor sampling interval rather than by network latency, which is an important reliability advantage over cloud-mediated protection schemes.

4.5 Wiring and Interfacing

Table VI lists the pin-level interconnections between the ESP32 and the peripheral modules used in the prototype, providing a concise reference for hardware assembly.

TABLE VI. ESP32 PIN INTERFACING SUMMARY

Module	ESP32 Pin(s)	Notes
DHT11	GPIO4 (DATA)	3.3V, GND, single-wire data
ACS712	GPIO34 (ADC)	Analog current output
ZMPT101B	GPIO35 (ADC)	Analog voltage output
Relay Module	GPIO5 (IN)	Drives fan / load disconnect
Cooling Fan	via Relay NO contact	12V DC supply

4.6 Design Considerations

- Selection of compatible, low-power hardware components for seamless interfacing with the ESP32.
- Stable, regulated power supply (SMPS + 7805) for reliable, continuous operation of sensors and controller.
- Accurate calibration of temperature, voltage, and current sensors prior to threshold configuration.
- Proper electrical isolation between high-voltage load circuits and the low-voltage control electronics.
- Efficient relay switching logic to avoid nuisance tripping while maintaining fast protective response.
- Reliable Wi-Fi connectivity for uninterrupted remote monitoring and web-based reporting.
- Compact hardware arrangement and clean wiring practice to minimize electrical noise on analog lines.
- Provision for future hardware and firmware expansion, including additional sensing modalities.

4.7 Implementation Procedure

1. Design and simulate the complete circuit in Cirkit Designer, connecting the ESP32 with all sensors and actuators.
2. Develop the ESP32 firmware in the Arduino IDE for sensor acquisition, threshold comparison, and relay control.
3. Verify and upload the compiled firmware to the ESP32 Development Board via USB.
4. Assemble the hardware: connect the DHT11, ACS712, ZMPT101B, relay module, cooling fan, and LED load to the ESP32 according to the verified wiring diagram.
5. Power the assembled unit using the regulated SMPS/7805 supply and confirm stable operation of all modules.
6. Test the complete system under normal and simulated abnormal (over-temperature and overload) conditions, and verify relay actuation and web-server reporting.

This structured procedure — simulate, code, assemble, and test — minimizes wiring errors, shortens development time, and ensures that the protective logic operates correctly before the unit is deployed for extended evaluation.

5. Results and Discussions

The complete hardware prototype was assembled according to the design verified in Cirkit Designer and evaluated under a range of operating conditions using the LED panel board as an adjustable electrical load. The ESP32 continuously updated the embedded web page with live temperature, humidity, current, and voltage readings, confirming successful sensor integration and wireless data delivery.

5.1 Functional Test Results

Table VII summarizes the system response recorded under four representative operating scenarios: normal operation, elevated temperature, overload condition, and a combined temperature/overload fault.

TABLE VII. EXPERIMENTAL TEST RESULTS

Scenario	Temp (°C)	Current (A)	Voltage (V)	System Response
Normal operation	30–35	0.4–0.6	228–232	Relay OFF, no alert
Elevated temperature	46–50	0.5	230	Relay ON, fan activated, alert shown
Overload condition	34	2.3–2.6	229	Relay ON, load protection triggered
Combined fault	48	2.4	231	Relay ON, dual alert (temp + overload)

During normal operation, the measured temperature remained within the 30–35°C range and current stayed below 1 A, and the relay remained in the OFF state as expected. When the ambient temperature was artificially raised beyond the 45°C threshold, the ESP32 activated the relay within the sampling interval, switching on the DC cooling fan and displaying an over-temperature alert on the web page. Similarly, when the load current was increased beyond 2 A using the LED panel board, the relay was triggered to provide load protection and an overload warning was displayed. Under combined fault conditions, both alerts were raised simultaneously, confirming that the ESP32 correctly evaluates multiple threshold conditions in parallel without conflict.

5.2 Discussion

The experimental results confirm that the proposed system meets its principal design objectives: accurate real-time sensing, automatic threshold-based protection, and remote accessibility via a standard web browser. The measured response of the relay to both over-temperature and overload conditions was consistent across repeated trials, indicating reliable firmware logic and stable sensor calibration. The use of an ESP32-hosted web server eliminates the dependency on external cloud infrastructure for basic monitoring, keeping the implementation cost low while still enabling remote supervision from any device connected to the same network.

Compared with the manual-inspection-based existing system described in Section III, the proposed architecture offers continuous, unattended monitoring, sub-second automatic protective response, and significantly reduced dependence on operator presence. Compared with related IoT-based approaches reviewed in Section II, the present system combines both thermal management (active cooling) and load protection (disconnection/alerting) within a single low-cost ESP32-centric unit, whereas several prior works addressed only one of these functions in isolation. The results therefore support the conclusion that the proposed design provides a practical, cost-effective improvement over both conventional and several existing IoT-based transformer monitoring approaches.

5.3 Sensor Calibration and Repeatability

To assess measurement repeatability, each sensor reading in Table VII was recorded as the steady-state average of multiple consecutive samples taken over a 30-second observation window at each test scenario. The DHT11 temperature readings showed a variation within its rated ±2°C accuracy across repeated trials, while the ACS712-derived current values remained within approximately ±0.05 A of the applied load current after calibration of the zero-current offset. The ZMPT101B voltage readings tracked the supply voltage within a few volts across all test scenarios, consistent with the sensor's rated accuracy for AC voltage sensing applications. These results confirm that the selected low-cost sensor set provides measurement accuracy sufficient for threshold-based protection, though they are not intended to replace precision metering-grade instrumentation.

5.4 Approximate Implementation Cost

One of the practical advantages claimed for the proposed system is low implementation cost relative to SCADA-based alternatives. Table VIII lists the approximate component-level cost breakdown of the prototype, based on typical retail pricing for hobbyist and small-batch industrial-grade modules.

TABLE VIII. APPROXIMATE COMPONENT COST BREAKDOWN

Component	Approx. Cost (USD)
ESP32 Dev. Board	6 – 9
DHT11 Sensor	1 – 2
ACS712 Current Sensor	2 – 4
ZMPT101B Voltage Sensor	2 – 3
Relay Module	1 – 2
SMPS + 7805 Supply	3 – 5
Cooling Fan + Misc.	2 – 4
Total (approx.)	17 – 29

This total component cost is a small fraction of the cost typically associated with SCADA-based remote terminal units, supporting the conclusion that the proposed architecture is economically viable for widespread deployment across small and medium distribution transformers, where SCADA instrumentation is rarely justified.

5.5 Remote Monitoring Performance

The embedded web server hosted by the ESP32 was accessed from a laptop and a smartphone connected to the same Wi-Fi network. In both cases, the displayed temperature, humidity, current, and voltage values updated consistently with the on-board sampling interval, and fault indications (over-temperature and overload messages) appeared on the web page immediately upon relay activation. No perceptible lag was observed between a threshold breach and the corresponding update of the web interface, confirming that the single-board web-server approach is adequate for local, substation-level remote supervision without requiring external cloud infrastructure.

6. Conclusion

This paper presented the design, implementation, and experimental validation of an IoT-enabled smart transformer monitoring and protection system built around the ESP32 microcontroller. By integrating a DHT11 temperature/humidity sensor, an ACS712 current sensor, and a ZMPT101B voltage sensor with automatic relay-based protection and an embedded Wi-Fi web server, the system achieves continuous, real-time monitoring together with automatic corrective action whenever temperature or load current exceeds safe operating limits. Hardware testing confirmed that the system reliably activates the cooling fan above 45°C and triggers load protection above 2 A, while simultaneously reporting live parameters through a browser-accessible interface. Compared with conventional manual-inspection-based practice and several existing IoT-based approaches, the proposed system offers continuous unattended supervision, fast automatic protective response, remote accessibility, and low implementation cost, making it well suited for distribution transformers in substations, industrial installations, and smart-grid environments. Future enhancements, including additional oil-quality and vibration sensors, cloud-based data logging, AI-based fault prediction, and mobile-application support, can further extend the reliability and predictive capability of the proposed monitoring platform.

7. References

[1] A. Bahga and V. Madisetti, Internet of Things: A Hands-On Approach. Hyderabad, India: Universities Press, 2015.

[2] O. Hersent, D. Boswarthick, and O. Elloumi, The Internet of Things: Key Applications and Protocols, 2nd ed. Hoboken, NJ, USA: Wiley, 2012.

[3] K. V. Shukla and P. K. Mishra, Power Transformer Engineering. New Delhi, India: McGraw-Hill Education, 2018.

[4] M. A. Mazidi, S. Naimi, and S. Naimi, The ESP32 Microcontroller and Embedded Systems. Pearson Education, 2021.

[5] P. Scherz and S. Monk, Practical Electronics for Inventors, 4th ed. New York, NY, USA: McGraw-Hill Education, 2016.

[6] S. Monk, Programming Arduino: Getting Started with Sketches, 2nd ed. New York, NY, USA: McGraw-Hill Education, 2016.

[7] E. Margolis, B. Jepson, and M. Weldin, Arduino Cookbook, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2020.

[8] Espressif Systems, ESP32 Series Technical Reference Manual, Shanghai, China, 2023.

[9] Espressif Systems, ESP32-WROOM-32 Datasheet, Shanghai, China, 2023.

[10] Aosong Electronics, DHT11 Temperature and Humidity Sensor Datasheet, Guangzhou, China.

[11] Allegro MicroSystems, ACS712 Fully Integrated Hall-Effect Current Sensor Datasheet, Worcester, MA, USA.

[12] ZMPT101B AC Voltage Sensor Module, Technical Datasheet.

[13] A. Al Noman, P. Baidya, M. A. Hossain, P. Dev, K. Saha, and M. L. Hossain, "IoT-driven distribution transformer health monitoring system," in Proc. Int. Conf. Emerging Technol., 2025.

[14] S. U. Rehman and A. Khan, "IoT-based transformer monitoring and protection for smart power distribution networks," IEEE Access, vol. 11, 2023.

[15] E. H. El-Desouky, M. F. Kotb, M. M. El-Saadawi, E. A. Gouda, and D. A. Elalfy, "Overcurrent protection relay for FREEDM systems using Arduino Uno," Electr. Power Syst. Res., 2019.