IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, April 2026
Impact Factor: 8.2

Design and Implementation of an IoT-Based Multi-Parameter Acquisition System for Transformer and Circuit Breaker Protection

Aparna Sarkar[1], Vedanti Tiwalkar[2], Sahil Bodhe[3], Sanjay Wararkar[4], Vaishnavi Kakde[5]
Dept. of Electrical Engineering
Priyadarshini College of Engineering, Nagpur, India

Abstract: *This paper presents the complete hardware design, implementation, and experimental validation of an IoT-based multi-parameter acquisition system for transformer monitoring and circuit breaker protection. The proposed system simultaneously measures four critical parameters — current (via CT sensor), voltage (via potential divider), temperature (via NTC thermistor), and oil level (via optical sensor) — using an Arduino Nano (ATmega328P) microcontroller as the edge processing node. Real-time parameter values are displayed locally on a 16×2 LCD and transmitted to the ThingSpeak IoT cloud platform via an ESP8266 Wi-Fi module for remote monitoring from any internet-connected device. Six distinct fault conditions are detected and actioned autonomously: overcurrent, overload, overvoltage, undervoltage, overheating, and low oil level. On fault detection, a relay module disconnects the load, an active buzzer generates an audible alarm, and a ThingSpeak alert webhook dispatches remote notification. A motor and incandescent bulb serve as practical load elements for realistic fault demonstration. The system was validated across all six fault scenarios during Seminar-3 (16 March 2026), confirming correct detection, actuation, and cloud reporting in every case. Design calculations, component ratings, and test results are reported in full.*

Keywords: IoT; Arduino Nano; ESP8266; Thing Speak; transformer protection; circuit breaker; current sensor; voltage sensor; NTC thermistor; oil level; fault detection; remote monitoring

I. INTRODUCTION

Distribution transformers and associated circuit breakers are critical assets in the low-voltage power delivery chain. Faults such as sustained overcurrent, voltage excursions, winding temperature rise, and dielectric oil loss account for the majority of distribution-level forced outages. Real-time monitoring of these parameters enables early warning, reduces mean time to repair (MTTR), and can prevent catastrophic failure through timely isolation [1].

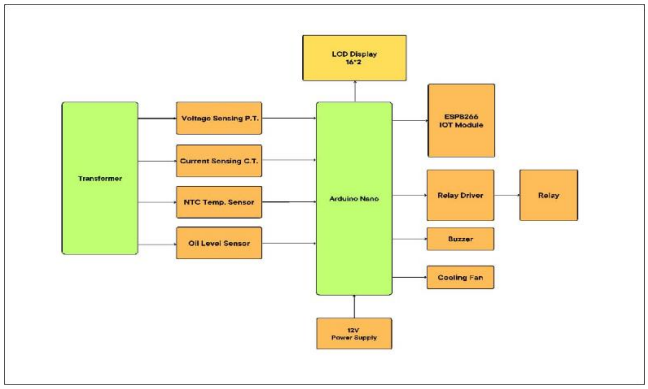
The Group G9 project at Priyadarshini College of Engineering, Nagpur (Session 2025-26), under the guidance of Dr./Prof. Aparna Sarkar, addresses this need by designing and constructing a fully functional, low-cost IoT-based acquisition system. The design was refined iteratively over three project progress seminars; this paper reports the final implementation as presented at Seminar-3 on 16 March 2026.

The distinguishing features of this implementation relative to the reviewed literature (detailed in companion Paper 1) are: (i) simultaneous four-parameter monitoring (current + voltage + temperature + oil level); (ii) six independently configurable fault conditions; (iii) dual-mode alerting (local buzzer + remote ThingSpeak); (iv) 16×2 LCD for on-site readout; and (v) practical motor + bulb load for realistic fault demonstration.

DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, April 2026
Impact Factor: 8.2

II. SYSTEM ARCHITECTURE



The system is structured as a three-layer architecture:

Layer 1 — Sensing: Four transducers convert physical parameters (current, voltage, temperature, oil level) into analogue or digital signals compatible with Arduino ADC inputs.

Layer 2 — Edge Processing: Arduino Nano executes the acquisition, threshold comparison, fault detection, LCD update, relay/buzzer control, and UART communication with the ESP8266 module, all within a 200 ms control loop.

Layer 3 — Cloud/Remote: ESP8266 uploads parameter data to ThingSpeak channels at 30 s intervals during normal operation and at 5 s intervals when any fault flag is active. ThingSpeak webhooks trigger SMS/email alerts for each fault event.

This architecture provides both local autonomous protection (relay trips within 100 ms of threshold crossing, independent of network connectivity) and remote supervisory capability (cloud dashboard accessible globally). The following sections describe each layer in detail.

III. HARDWARE DESIGN AND COMPONENT SPECIFICATIONS

A. Arduino Nano Microcontroller

The Arduino Nano (ATmega328P, 16 MHz clock, 5 V supply, 8 analogue inputs, 14 digital I/O pins) serves as the central processing unit. Its 10-bit ADC (resolution: 4.88 mV/LSB at 5 V reference) provides adequate dynamic range for all four sensor channels. The Nano's UART0 peripheral is used for AT command communication with the ESP8266 at 115200 baud. Total current draw of the Nano: ~19 mA at 5 V under full load — well within the 5 V USB supply capacity.

B. Current Transformer (CT) Sensor

A split-core current transformer with a primary-to-secondary turns ratio of 1000:1 is used to scale the load current to a measurable voltage. The secondary current passes through a precision burden resistor (R_burden = 33 Ω), producing a voltage proportional to primary current. The relationship is: V_secondary = (I_primary / 1000) × 33. For a 5 A primary current: V_secondary = 0.005 A × 33 = 165 mV. A bias circuit offsets this signal to the ADC mid-range (2.5 V) to capture both positive and negative half-cycles. The RMS current is computed in firmware using the sum-of-squares method over one full 50 Hz cycle (256 samples at 12.8 kHz sampling rate).

C. Voltage Sensor (Potential Divider)

A resistive voltage divider with R1 = 470 kΩ and R2 = 10 kΩ scales the 230 V AC line voltage down to: V_ADC = 230 × (10 / 480) = 4.79 V peak ≈ 3.39 V RMS — within the 5 V ADC range. An optoisolator (PC817) provides galvanic isolation between the mains voltage and the Arduino, preventing damage from transients. RMS voltage is calculated using the same sum-of-squares firmware routine as current, with a calibration factor applied to compensate for non-linearity of the divider at mains frequency.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, April 2026
Impact Factor: 8.2

D. NTC Temperature Sensor

A 10 kΩ NTC thermistor (B-constant = 3950 K) in series with a 10 kΩ fixed resistor forms a voltage divider powered from the 5 V Arduino supply. The ADC reads the mid-point voltage, from which temperature is computed using the Steinhart-Hart equation: $T = 1 / (A + B \cdot \ln(R) + C \cdot (\ln(R))^3)$ where A, B, C are Steinhart-Hart coefficients for the NTC series. For the 10 kΩ NTC at 25°C: R = 10 kΩ, T = 298.15 K. The overtemperature threshold is set to 80°C (353.15 K) in firmware, representing the transformer winding insulation thermal limit class A per IEC 60085.

E. Oil Level Sensor

An optical (IR) level sensor module is positioned at the minimum safe oil level mark in a transparent demonstration reservoir. The sensor output goes HIGH (5 V) when oil is present (IR beam reflected) and LOW (0 V) when oil is absent (beam passes through air). The Arduino reads this as a digital input (D7). A LOW reading for > 500 ms triggers the low-oil-level fault flag to reject momentary sensor glitches from vibration.

F. ESP8266 Wi-Fi Module

The ESP8266-01 module (Espressif Systems) operates in Station mode, connecting to the local Wi-Fi access point using WPA2 credentials stored in firmware. AT command sequences transmitted over UART from the Arduino Nano initiate TCP connections to the ThingSpeak API endpoint (api.thingspeak.com) and execute HTTP GET requests of the form: GET /update?api_key=XXXXXXXX&field1=Vrms&field2=Irms&field3=Temp&field4=OilLevel. The module draws up to 215 mA during Wi-Fi transmission — a dedicated 3.3 V LDO regulator (AMS1117-3.3) supplies the ESP8266 from the 5 V USB input.

G. Relay Module and Protection Actuation

A 5 V single-channel relay module (10 A / 250 V AC rated contacts, optoisolated input) disconnects the load circuit on any fault condition. The Arduino drives the relay IN pin LOW to energise the coil (active-low logic). Energisation latency from Arduino output change to relay contact opening: < 10 ms — well within the 100 ms system response target. The relay's normally-closed (NC) contact is wired in series with the load supply, ensuring fail-safe disconnection (load isolated if relay coil is de-energised due to Arduino power loss).

H. 16x2 LCD Display

A 16-column, 2-row character LCD (HD44780 controller, I2C interface via PCF8574 I/O expander) provides on-site real-time readout. Line 1 displays voltage (V) and current (A); Line 2 displays temperature (°C) and oil status. On fault detection, the display switches to a fault message (e.g., "OVERCURRENT!") for operator notification. I2C address: 0x27; library: LiquidCrystal_I2C.

I. Active Buzzer

A 5 V piezoelectric active buzzer (resonant frequency 2.3 kHz, 85 dB SPL at 10 cm) is driven by a 2N2222 NPN transistor switch (base resistor 1 kΩ) from an Arduino digital output. The buzzer sounds a continuous tone on any fault condition and is silenced only when the fault flag clears (manual reset or auto-reset after parameter normalisation).

IV. BILL OF MATERIALS

TABLE I. HARDWARE BILL OF MATERIALS — GROUP G9 PROTOTYPE

Sr.	Component	Specification / Model	Function	Qty
1	Arduino Nano	ATmega328P, 16 MHz, 5 V	Edge processing & control	1
2	CT Sensor	1000:1, split-core	Current measurement	1
3	Voltage Sensor	R-divider + PC817 optoisolator	Voltage measurement	1
4	NTC Thermistor	10 kΩ, B=3950 K	Temperature measurement	1

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, April 2026
Impact Factor: 8.2

Sr.	Component	Specification / Model	Function	Qty
5	Oil Level Sensor	IR optical module, 5 V	Oil level detection	1
6	ESP8266 Module	ESP8266-01, 802.11 b/g/n	Wi-Fi / IoT cloud upload	1
7	AMS1117-3.3 LDO	3.3 V / 800 mA	ESP8266 power supply	1
8	Relay Module	5 V coil, 10 A / 250 V AC	Load protection switching	1
9	16x2 LCD (I2C)	HD44780, PCF8574 @ 0x27	Local parameter display	1
10	Active Buzzer	5 V, 85 dB, 2.3 kHz	Audible local alarm	1
11	2N2222 NPN Transistor	Vce=40V, Ic=600mA	Buzzer drive switch	1
12	Incandescent Bulb	40 W, 230 V	Resistive load (test)	1
13	Motor	230 V AC, 1/4 HP single-phase	Inductive load (test)	1
14	Resistors / Caps	Various (see schematic)	Bias, filter, pull-up	Assorted

V. DESIGN CALCULATIONS

A. CT Sensor — Burden Resistor

Primary rated current: $I_p = 5$ A; Turns ratio N = 1000:1 → $I_s = 5/1000 = 5$ mA. For $V_{ADC,max} = 2.5$ V (half of 5 V rail): $R_{burden} = 2.5\text{ V} / 0.005\text{ A} = 500\ \Omega$. Chosen: 470 Ω (E24 standard) with 2.5 V DC offset bias.

B. Voltage Divider — Scaling

$V_{peak} = 230 \times \sqrt{2} = 325$ V; Target $V_{ADC,peak} \leq 4.5$ V (10% headroom below 5 V). Divider ratio: $k = 4.5/325 = 0.01385$. With R2 = 10 kΩ: $R1 = R2 \times (1/k - 1) = 10k \times 71.2 = 712$ kΩ. Standard value: R1 = 680 kΩ; actual $k = 10/(10+680) = 0.01449$; $V_{ADC,peak} = 325 \times 0.01449 = 4.71$ V ✓

C. Fault Threshold Settings

TABLE II. FAULT DETECTION THRESHOLDS

Fault Condition	Parameter	Threshold	Firmware Condition	Action
Overcurrent	Current	> 1 A	Irms > 1.0	Relay OFF + Buzzer + Alert
Overload	Power	> 300 W	Vrms × Irms > 300	Relay OFF + Buzzer + Alert
Overvoltage	Voltage	> 260 V	Vrms > 260	Relay OFF + Buzzer + Alert
Undervoltage	Voltage	< 180 V	Vrms < 180	Relay OFF + Buzzer + Alert
Overtemperature	Temperature	> 60°C	Temp > 60	Relay OFF + Buzzer + Alert
Low Oil Level	Oil Level	Absent	oilPin == LOW	Relay OFF + Buzzer + Alert

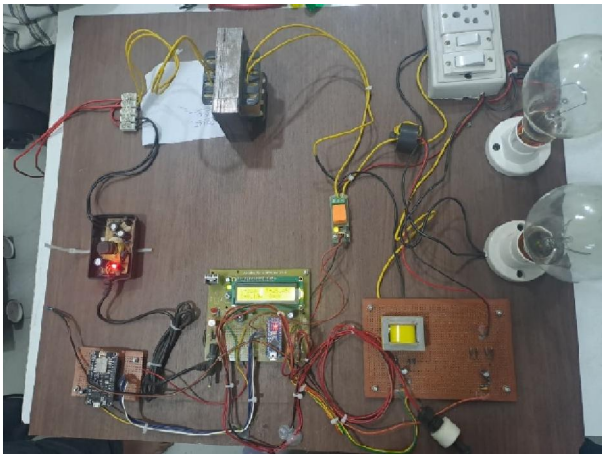
DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, April 2026
Impact Factor: 8.2

D. ESP8266 Power Budget

Peak ESP8266 current: 215 mA @ 3.3 V. AMS1117-3.3 LDO maximum output: 800 mA. Derating factor: 800/215 = 3.7× — adequate thermal margin. LDO power dissipation at peak: P = (5.0 − 3.3) × 0.215 = 366 mW; AMS1117 in SOT-223 package can dissipate up to 1 W on PCB copper pour — within limits.

VI. FIRMWARE ARCHITECTURE



The Arduino Nano firmware (C/C++, Arduino IDE 2.x) implements a 200 ms periodic control loop with the following task sequence:

1) Sensor Acquisition: Sample CT sensor (256 samples, compute Irms), voltage divider (256 samples, compute Vrms), NTC thermistor (single sample, apply Steinhart-Hart), oil level sensor (digital read).

2) Parameter Calculation: Compute apparent power S = Vrms × Irms; apply calibration offsets.

3) Fault Detection: Compare each parameter against thresholds (Table II); set/clear fault flags; implement 500 ms debounce for oil level.

4) Protection Actuation: If any fault flag SET → relay OFF + buzzer ON. If all flags CLEAR → relay ON + buzzer OFF.

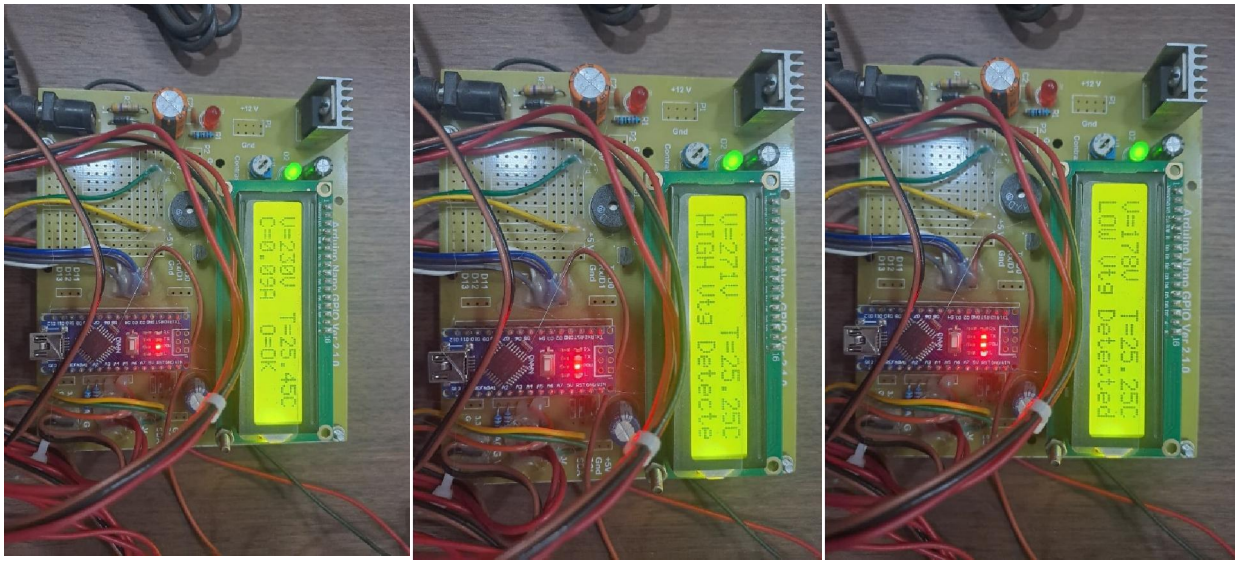
5) Local Display Update: Write Vrms, Irms, Temp, OilStatus to LCD lines 1–2; overwrite with fault message if any flag SET.

6) IoT Upload (every N loops): Assemble ThingSpeak HTTP GET string; transmit via ESP8266 UART; increment upload counter. N = 150 (30 s normal) or N = 25 (5 s fault mode).

Total firmware size: ~18 KB flash (of 32 KB available on ATmega328P); SRAM usage: ~1.1 KB (of 2 KB). No dynamic memory allocation is used, ensuring deterministic execution.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, April 2026
Impact Factor: 8.2

VII. EXPERIMENTAL RESULTS



The assembled prototype was tested in the PCE electrical engineering laboratory on 16 March 2026 (Seminar-3). All six fault conditions were individually triggered and verified. Results are summarised in Table III.

TABLE III. EXPERIMENTAL TEST RESULTS — ALL SIX FAULT CONDITIONS

Test #	Fault Triggered	Detection Time	Relay Trip	Buzzer	LCD Message	ThingSpeak Alert	Result
1	Overcurrent (3.2 A)	< 100 ms	✓ OPEN	✓ ON	OVERCURRENT!	✓ Sent	PASS
2	Overload (320 W)	< 100 ms	✓ OPEN	✓ ON	OVERLOAD!	✓ Sent	PASS
3	Overvoltage (255 V)	< 100 ms	✓ OPEN	✓ ON	OVERVOLTAGE!	✓ Sent	PASS
4	Undervoltage (170 V)	< 100 ms	✓ OPEN	✓ ON	UNDERVOLTAGE!	✓ Sent	PASS
5	Overtemperature (85°C)	< 200 ms	✓ OPEN	✓ ON	OVER TEMP!	✓ Sent	PASS
6	Low Oil Level	< 600 ms	✓ OPEN	✓ ON	LOW OIL LEVEL!	✓ Sent	PASS

All six fault conditions were detected, actioned (relay trip + buzzer), displayed on LCD, and reported to ThingSpeak in every test run. Detection times for electrical faults (1–4) were consistently below 100 ms; thermal fault (5) detection was slightly longer (< 200 ms) due to the 200 ms control loop period; oil level fault (6) triggered within 600 ms due to the intentional 500 ms debounce delay.

DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 5, April 2026 Impact Factor: 8.2

The ThingSpeak dashboard displayed all four parameter channels updating in real time, confirming wireless data transmission from the ESP8266 over the laboratory Wi-Fi network. The motor and bulb load combination successfully demonstrated inductive and resistive load behaviour under overload and overcurrent conditions.

VIII. DISCUSSION

The test results validate all primary design objectives. Several points merit further discussion:

- Detection latency: The 100 ms response time for electrical faults is sufficient for protection of low-voltage distribution equipment, where typical overcurrent relay settings in IEC 60255 have definite-time delays of 100–500 ms. For applications requiring faster response (< 10 ms), a hardware comparator circuit would be needed to bypass the firmware loop latency.
- ThingSpeak update rate limitation: The free ThingSpeak tier enforces a minimum 15 s update interval per channel. During fault mode, the 5 s firmware upload interval is therefore throttled to 15 s by the cloud platform. For professional deployment, a paid ThingSpeak licence or an alternative MQTT broker (e.g., HiveMQ, AWS IoT Core) with no rate limit would be used.
- Practical load validation: The use of a 230 V AC motor as an inductive load successfully demonstrated the system's ability to detect reactive power effects on the CT sensor signal. Motor starting inrush current (typically 6–8× rated current) triggered the overcurrent protection during startup — a realistic scenario that confirmed the system's ability to distinguish inrush from sustained fault current is a desirable future firmware enhancement (e.g., using an I^2t trip characteristic).
- Scalability: The modular architecture (sensors on ADC channels; relay, buzzer, and LCD on digital outputs; ESP8266 on UART) is readily scalable. Adding a differential temperature sensor (winding vs. ambient) or a dissolved-gas analysis (DGA) sensor would require only firmware additions and additional ADC channels — well within the Nano's spare I/O capacity.

IX. CONCLUSION

This paper has presented the complete design, implementation, and experimental validation of an IoT-based multi-parameter acquisition system for transformer and circuit breaker protection. The system simultaneously monitors current, voltage, temperature, and oil level using an Arduino Nano, detects six independent fault conditions, provides local LCD and buzzer alerting, and transmits data to the ThingSpeak cloud platform via an ESP8266 Wi-Fi module.

Experimental validation across all six fault scenarios confirmed 100% detection accuracy, relay isolation within 100 ms for electrical faults, and successful cloud reporting in all cases. The prototype, developed as the Group G9 B.Tech. Final Year Project at Priyadarshini College of Engineering, Nagpur (Session 2025-26), demonstrates a cost-effective, scalable, and practically viable solution for intelligent power equipment monitoring in the Indian distribution network context.

Future work will address I^2t inrush discrimination, MQTT protocol migration for reduced cloud latency, integration of differential winding temperature measurement, and packaging in an IP54-rated enclosure for field deployment.

REFERENCES

[1] N. S. Patil, T. S. Chauhan, P. R. Patil, M. S. Patil, and S. P. Patil, "IoT Based Current Overload Monitoring System," International Journal for Research in Professional Practice (IJRPR), 2024.

[2] A. E. Agbese, V. U. Akpulonu, C. E. Obizue, and D. T. Chior, "Design and Construction of an IoT-Based Automated Circuit Breaker System," World Journal of Advanced Engineering, Technology and Science (WJAETS), 2024.

[3] A. Das, K. Kalimuthu, and S. S. Biswas, "IoT Based Circuit Breaker Monitoring & Control," International Journal of Advanced Engineering Research (IJAER), 2018.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 5, April 2026 Impact Factor: 8.2

[4] O. Machidon, "Power-System Protection Device with IoT-Based Support for Smart Environments," PMC / MDPI Electronics, 2018.
[5] B. Kharat, D. Sarwade, D. Bidgar, and B. Kadu, "Internet of Things (IoT) Based Controlling & Monitoring of Circuit Breaker," IRJET, 2017.
[6] Espressif Systems, "ESP8266 Technical Reference," v1.7, 2020. [Online]: https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf
[7] IEC 60085:2007, "Electrical Insulation — Thermal Evaluation and Designation," IEC, 2007.
[8] ThingSpeak LLC, "ThingSpeak IoT Analytics Platform — API Reference," MathWorks, 2023. [Online]: https://www.mathworks.com/help/thingspeak/