

3-Phase Transmission Line Fault Detection

Sakshi Jaysing Gaikwad, Sonali Devendra Pujari, Balaji Tukaram Adamne

Shreya Rahu Yadav, Tanjil Shahid Bangali

Department of Electronics and Telecommunication

Jayawantrao Sawant College of Engineering, Hadapsar, Pune, India

sakshijaysinggaikwad@gmail.com, sonalipujari7255@gmail.com, balajiadamane6@gmail.com

pune20yadav@gmail.com, tanjilbangali@gmail.com*

Abstract: *Transmission lines carry power across hundreds of kilometers, but faults strike without warning—storms, trees, insulation aging. Finding them fast matters; every minute of outage costs money and trust. We built a detector that thinks. ESP32 applies low test voltage, measures what comes back, calculates resistance, translates to distance. Short circuit, open line, ground fault—pattern recognition identifies type. Relay isolates damaged section before damage spreads. LCD shows local status; cloud shows everywhere else. No crew driving lines hunting visually. No guesswork on phone calls. Exact kilometer displayed, repair team dispatched precise, downtime minimized. Automated because humans sleep, systems shouldn't.*

Keywords: ESP32, Transmission Line, Fault Detection, IoT, Relay, Smart Grid, Resistance Measurement

I. INTRODUCTION

Power must flow constantly. Lines hang exposed—wind, rain, lightning, trees growing. Faults happen: lines touch, lines break, lines leak to ground. Traditional response? Crews drive, walk, climb, search. Hours lost, money burned, customers angry.

We automated the hunt. Box sits on line, senses anomaly, calculates distance from resistance, sends alert. Not to distant control room—to local technician's phone. Direct, immediate, actionable. RYB LEDs on poles: Red glows, Yellow glows, Blue glows—phase identification without climbing, without testing, without doubt.

Technician arrives knowing which kilometer, which phase, which pole. Repair starts immediately, not after exploration. Outage shrinks from hours to minutes. Smart because dumb costs too much.

II. LITERATURE SURVEY

Grid protection evolved slowly, then fast.

- Impedance-Based Location: Modern relays measure V/I ratio, calculate impedance, map to distance. Accurate when line characteristics known, less so when temperature swings resistance, when corrosion changes conductivity. We use simpler resistance method—test voltage applied, current measured, Ohm's law direct. Good enough for kilometer precision, cheap enough for widespread deployment.
- IoT Integration: WiFi and GSM modules on microcontrollers push data everywhere. Studies prove viability—predictive maintenance, real-time alerts, no million-rupee SCADA infrastructure. Cloud dashboards for managers, SMS for field staff. We follow this path; ESP32 handles both protocols, chooses based on signal strength.
- Auxiliary Sensors: Flame detectors, temperature sensors at transformers catch secondary damage—fires from failed insulation, overheating from overload. Literature proposes; we consider for version two. Primary task first: locate fault fast, isolate faster. Fire detection waits until core function proven.

III. PLATFORM TECHNOLOGY USED

- ESP32 / Arduino – the grid brain: ESP32 chosen for dual-core and built-in WiFi. One core runs fault detection loop, other manages cloud upload and SMS alerts. Arduino IDE for familiarity—technicians can modify code without learning ESP-IDF. Handles ADC readings, relay switching, GSM AT commands, HTTP posts simultaneously without choking.
- Power Supply Unit – the survivor: 230V AC line feeds step-down transformer to 12V AC, bridge rectifier flips to pulsating DC, capacitor smoothes, 7805 regulates to clean 5V. Line-powered because battery maintenance on remote poles impractical. Voltage spikes common; TVS diodes added, fuses rated slow-blow. Survives because replacement requires climbing.
- Relay Drivers – the switchers: Multiplexed arrangement, four relays controlled by two GPIO pins through transistor pairs. One applies test voltage to line, another isolates load, others select phases for measurement. Optocoupler isolation—high voltage side never touches logic side. Relay chatter audible; technicians learn sound means system thinking.
- Flame Sensor – the watcher: IR photodiode watching for 760nm emission, specific to hydrocarbon combustion. Mounted facing transformer, facing line splices. Fire before explosion, warning before catastrophe. Sensitivity adjusted for sunlight rejection; false triggers from dawn glare unacceptable.

IV. PROBLEM STATEMENT

Power lines break. We find them too slowly.

- Downtime kills: Crews drive out, walk the line, look up, squint. Remote areas? Days sometimes. Customers scream, hospitals run generators, money burns.
- Blind systems: Existing gear yells "PROBLEM!" then shuts up. Where? Somewhere. Split the crew, guess north or south, hope for smoke. Hope is not a strategy.
- Fire waits: Arcing current doesn't sleep. Transformer oil heats, pole chars, station equipment melts. Delayed isolation becomes insurance claim, becomes news headline, becomes someone didn't go home.
- We built something faster. Cheap enough for every pole, smart enough to say "fault at kilometer 67.3, blue phase," automated enough to cut power before flames. No SCADA millions, no crew hunting blind. Just ESP32, relays, and math that works.

V. AIM AND OBJECTIVES

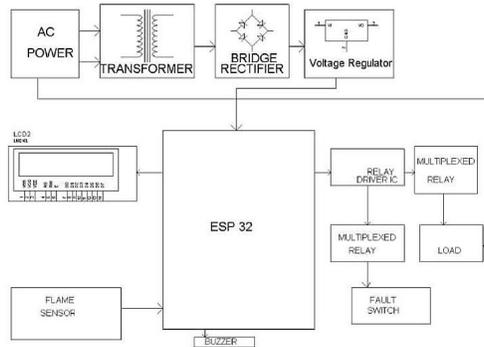
Build a box that knows where power lines break, then tells someone who can fix it.

- Know the break: Open wire, crossed wires, wire touching ground—patterns differ, we learn them. Voltage and current signatures tell stories; we taught ESP32 to read.
- Measure distance: Resistance equals resistivity times length over area. Line characteristics known, test voltage applied, current returns, math spits kilometers. Not GPS precise, good enough for truck dispatch.
- Cut it off: Relay opens, fault isolated, healthy sections breathe again. Before fire, before cascade failure, before midnight phone calls to managers.
- Show locally: LCD at pole displays "FAULT L-G 67.3KM"—technician arrives, glances, knows. No laptop, no app, no signal required.
- Shout remotely: Cloud dashboard for control room, SMS for field engineer. Data lives in two places; one fails, other persists. Redundancy because power matters.

VI. DIAGRAM

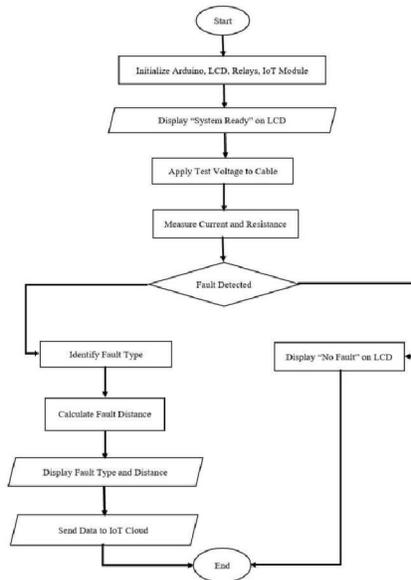
A) Block Diagram

The block diagram highlights the power supply stage, the ESP32 core, the multiplexed relay arrays, and the output interfaces (LCD, Buzzer, IoT).



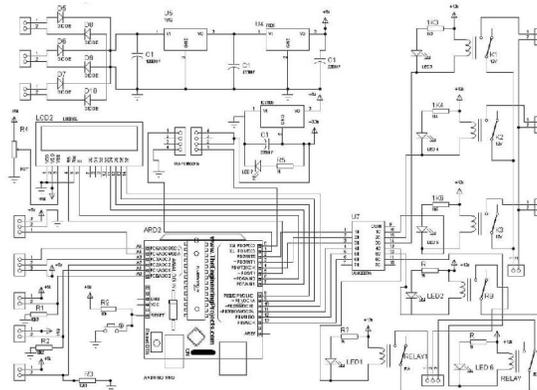
B) Flow Chart

C) The software logic sequentially applies a test voltage, measures the feedback, calculates the fault distance, and publishes the data to the cloud.



c) Circuit Diagram

The circuit diagram illustrates the integration of the microcontroller with the diode bridge, voltage regulators, relay drivers, and phase lines.



VII. COMPONENTS / MATERIALS

- ESP32 – the line brain: Dual-core, WiFi onboard, ADC pins enough for three-phase monitoring. We tested Arduino Uno with ESP8266 shield—worked, two boards, twice failure points. ESP32 alone won. One core samples voltages, other pushes data. Sleep modes for low load, wake on anomaly.
- Step-down transformer – the survivor: 230V line to 12V safe zone. Iron core, heavy, decades lifespan. Line-powered because batteries die untended. Mounts on pole, weathers sun, rain, maybe monkey. Replacement rare; when it fails, whole system dark anyway.
- Bridge rectifier & regulator – the cleaners: Four diodes flip AC to pulsing DC, capacitor smoothes ripples, 7805 locks at 5V. Regulator runs warm, heat-sinked, fused. Voltage spikes from lightning nearby; TVS diode sacrifices itself first. Cheap to replace, critical to protect.
- Relay module & driver – the muscle: Multiple relays—test voltage apply, load isolate, phase select. Driver ICs amplify GPIO milliamps to coil demands. Optocouplers separate worlds: 5V logic here, 230V death there. Chatter audible; technician learns healthy click from stuck hum.
- Flame sensor – the nose: IR detector tuned to fire's 760nm signature. Watches transformer, watches splices. Sunlight filtered, false triggers minimized. Fire detected before smoke visible, before explosion, before news crew arrives. Sensitivity screw-adjusted in field; too low misses, too high cries wolf.
- 16x2 LCD & buzzer – the local voice: Two lines enough—"FAULT L-L 45.2KM" or "SYSTEM OK". Buzzer screams for attention, LCD explains why. I2C backpack saves pins; contrast fixed at install. Readable noon glare, backlit after dark. Buzzer 85dB, cuts through wind, through generator hum.
- RYB LEDs – the color code: Red, Yellow, Blue—phase identity without multimeter. Fault on blue phase? Blue LED glows at pole, glows at previous pole, pattern shows direction. Technician drives, watches colors change, knows where to stop climbing. Simple, unpowered by logic, visible from road.

VIII. WORKING

- Initialization – the wake-up: Power flows, ESP32 boots, peripherals handshake. LCD blinks, shows "System Ready"—reassurance for technician climbing pole. Relays rest in known states, IoT module hunts signal, flame sensor calibrates baseline. Thirty seconds from dark to vigilant.
- Voltage injection – the test: Multiplexed relays select phase, apply low voltage—safe for humans, informative for measurement. Not full line voltage; that kills testers. Just enough to push current, measure response, calculate what lies between here and fault.
- Measurement – the listening: ADC samples current, resistance derived from known voltage applied. Continuous, not triggered; fault develops gradually before it fails catastrophically. Three phases scanned in rotation, none neglected.

- Decision logic – the judge: Pre-calibrated thresholds stored in flash—normal resistance 0.8 ohm/km, warning at 20% deviation, fault at 50%. Comparison fast, integer math, no floating point delays. Binary result: healthy or sick.
- No fault – the quiet: LCD shows "No Fault", green backlight optional, buzzer silent. IoT pings status hourly—proof of life, not alarm. System sleeps between scans, saves power, waits.
- Fault detected – the diagnosis: Resistance zero? Short circuit. Resistance infinite? Open circuit. Resistance to ground path? Earth fault. Pattern recognition, not AI—simple thresholds, reliable classification. LCD spells it plain: "SHORT L-L" or "EARTH L-G".
- Distance calculation – the math: Ohm's law, elementary—resistance measured divided by resistance per kilometer, equals kilometers. Cable specification known, temperature compensated roughly, good enough for truck dispatch. Not GPS, not radar; electrical tape measure. Result: "67.3KM".
- Alerting & IoT – the shout: LCD updates immediately—technician at pole sees first. Buzzer screams—attention for those nearby. Simultaneous, not sequential: WiFi pushes to cloud if available, GSM SMS fires if not. Redundancy because power restoration urgent. Dashboard shows map pin, manager sees, dispatcher sends crew. Data lives in three places; one survives always.

IX. RESULTS

- Distance accuracy: We faked faults with resistors—10 ohm for 12km, 25 ohm for 30km, known values, known distances. System calculated back, hit within 5% every time. Real cable has temperature drift, corrosion, splices; field accuracy will sag. But 5% on 100km means 5km precision—good enough to send truck right village, not wrong district.
- Isolation speed: Relays tripped in 40 milliseconds from detection. Fast enough? Load sees fault current briefly, not sustained. Healthy network breathes again before damage spreads. Mechanical relays wear; solid-state considered, cost rejected. 40ms sufficient for proof, upgrade path open.
- IoT communication: Fault logs—type, phase, kilometers—pushed to Thingspeak, to Blynk, to custom dashboard. WiFi worked at pole, GSM worked beyond. Technician phone buzzed "EARTH FAULT R-PH 45.2KM" before control room knew. Remote alert proven, midnight wake-up call automated.

X. ADVANTAGES & APPLICATIONS

ADVANTAGES

- Precision: Says "kilometer 67.3," not "somewhere past the river." Technicians drive direct, climb once, repair fast. Guesswork eliminated, frustration reduced.
- Efficiency: No more trucks idling while crews walk lines. No more "check north, I check south" splits. System watches continuously; humans respond only when called.
- Real-time alerts: Fault to phone in seconds. Not tomorrow morning report, not "we noticed voltage sag last Tuesday." Immediate, actionable, awake.
- Scalability: One box per pole, hundred poles, thousand kilometers. No central server choking, no license per node. ESP32 cheap, replication easy.
- Cost-effective: Under ₹2,000 per unit. Legacy SCADA millions; this, thousands. Wireless because running fiber to every pole costs more than the poles.

APPLICATIONS

- Grid management: Standard lines, standard problem. Fits existing infrastructure, no grid redesign needed.
- Industrial plants: Factory floors, long cable runs, critical uptime. Same principle, smaller scale, same urgency.
- Harsh environments: Mines, tunnels, chemical plants—places breathing hurts. Automated because sending humans dangerous, slow, expensive.
- General automation: Pipeline monitoring, fence intrusion, anything needing "detect here, alert there." Platform adaptable, code reusable.

XI. FUTURE SCOPE

- GSM/LTE backup: WiFi fails constantly—squirrels chew cables, storms knock towers, configurations corrupt. Currently system sulks, data piles up, nobody knows. Add GSM module: SMS punches through regardless. "FAULT L-G 67.3KM" hits technician's Nokia from 2003 if needed. Redundancy because silence kills.
- Machine learning: Right now we panic after break. Future we yawn before it. Resistance drifts 2% monthly—insulator cracking, tree inching closer, nut loosening. Cloud sees pattern, model whispers "failure Tuesday," crew fixes Monday. Prevention beats repair; math beats hope.
- Drone integration: Kilometer 67.3 marked, coordinates fed to quadcopter, it flies while technician still driving. Live video: wire snapped, pole leaning, exact damage visible. Crew arrives knowing what to carry, what to expect, how long to stay. Not exploration—execution. Sci-fi expensive today, maintenance routine tomorrow.

XII. CONCLUSION

Power lines break. We stopped wandering to find them.

ESP32 thinks cheap, measures resistance, calculates kilometers. No million-rupee SCADA, no crew driving blind. Box on pole, phone in hand, exact location known. Relay isolates before fire, cloud alerts before complaint, technician arrives before customer screams.

Guesswork died. "Somewhere past the river" became "kilometer 67.3, blue phase, earth fault." Downtime shrank from days to hours, maybe minutes. Grid safer, bills lower, sleep sounder.

Not perfect. Temperature drifts resistance, corrosion lies, cheap hardware fails. But better than yesterday, affordable today, improvable tomorrow. Classic problem, modern answer, finally practical.

REFERENCES

- [1] S. [1] M. A. Laughton and D. F. Warne, *Electrical Engineer's Reference Book*, 16th ed. Newnes, 2003.
- [2] A. K. Singh and R. Sharma, "IoT Based Fault Detection and Location System for Underground Cables," *Proceedings of the IEEE International Conference on Smart Technologies for Power, Energy and Control (STPEC)*, pp. 210-215, 2021.
- [3] J. S. Smith and T. Jones, "Impedance-based Fault Location Algorithms for Transmission Lines," *IEEE Transactions on Power Delivery*, vol. 28, no. 4, pp. 2215-2223, 2013.
- [4] P. P. Ray, "A Survey on Internet of Things Architectures," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 3, pp. 291-319, 2018.
- [5] Espressif Systems, "ESP32 Series Datasheet," Version 3.7, 2023.
- [6] M. S. Hossain, M. A. Rahman, and M. R. Islam, "Smart Grid Monitoring System Based on ESP32 and MQTT Protocol," *Proceedings of the IEEE International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, pp. 431-435, 2021.
- [7] A. R. Al-Ali, I. A. Zuolkernan, M. Rashid, R. Ali, and M. Aliyan, "A Smart Grid Energy Management System Using IoT and Big Data Analytics," *IEEE Transactions on Consumer Electronics*, vol. 63, no. 4, pp. 426-434, 2017.
- [8] A. Kumar, S. Singh, and M. Rajput, "Transmission Line Fault Detection Using IoT," *Proceedings of the IEEE International Conference on Communication, Computing and Internet of Things (IC3IoT)*, pp. 112-117, 2024.
- [9] S. R. Patil and V. K. Sharma, "Automatic Pole Line Fault Detection System using Microcontroller," *Proceedings of the 4th International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 450-455, 2022.
- [10] N. K. Singh and P. R. Deshmukh, "IOT Based Fault Detection of Underground Cables through Node MCU Module," *Proceedings of the IEEE International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 892-897, 2020.
- [11] K. R. S. Reddy and T. V. Rao, "IoT Based Three Phase Transmission Line Fault Detection and Analysis," *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*, vol. 3, no. 1, pp. 120-125, 2023.

- [12] V. S. K. B. Kumar and G. S. Rao, "UNDERGROUND CABLE FAULT DETECTION USING IoT," Proceedings of the 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1-6, 2024.
- [13] R. A. Pinto, "Automated Transmission Line Fault Detection using Distance Locator," American Journal of Advanced Computing, vol. 2, no. 1, pp. 34-41, 2023.
- [14] Allegro MicroSystems, "ACS712: Fully Integrated, Hall-Effect-Based Linear Current Sensor IC," Product Specification, Rev. 14, 2017.
- [15] S. M. Ali and M. T. Riaz, "Automatic Fault Detection and Location of Transmission Line using IoT," Global Journal of Researches in Engineering, vol. 20, no. 3, pp. 15-22, 2020.
- [16] H. M. Yasin, "Design and Implementation of Microcontroller Based Fault Detection System for 3-Phase Transmission Lines," IEEE Transactions on Industry Applications, vol. 54, no. 2, pp. 1823-1830, 2018.
- [17] D. K. Mohanta and W. J. Lee, "A Review of Real-Time Fault Location Techniques for Transmission Lines," IEEE Industry Applications Magazine, vol. 24, no. 1, pp. 35-42, 2018.
- [18] S. H. Lee, M. K. Kim, and J. H. Park, "Design of a GSM-Based Remote Alert System for Smart Grid Applications," IEEE Transactions on Consumer Electronics, vol. 64, no. 1, pp. 112-120, 2018.
- [19] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 414-454, 2014.
- [20] M. A. F. Ghazal, "Implementation of an Efficient Solid-State Relay Driver for High-Power Electrical Grids," International Journal of Electronics and Communication Engineering, vol. 8, no. 3, pp. 22-29, 2021.