PROJECT ABSTRACT

IoT EV Charging Monitor with Billing Dashboard

A retrofit metering unit (PZEM-004T + contactor + ESP32) turns an ordinary AC charge point into a metered, billed station with a live billing dashboard. Educational metering prototype — not a certified revenue meter. Delivered built-to-order with hardware, firmware, dashboard app, report, PPT and viva kit.

01. Title

IoT EV Charging Monitor with Billing Dashboard — true energy metering, contactor control and per-session billing.

02. Introduction / Background

Societies and campuses installing EV charge points usually end up with dumb sockets: one electricity bill, no fair way to split charging costs. Commercial smart chargers solve this at prices small setups cannot justify. The missing piece is a retrofit metering-and-billing layer — real kWh measurement per session, a tariff engine, and receipts — which this project builds around the PZEM-004T energy meter and an ESP32.

A 40 A contactor under ESP32 control handles safe start/stop with e-stop and overload cut-off, and the dashboard shows live power, running bills, session receipts and daily revenue.

03. Problem Statement

Small charging setups need fair per-session EV billing — measured energy, configurable tariffs, receipts and revenue summaries — without the cost of commercial smart chargers, while keeping high-current switching safe and documented.

04. Objectives

- Meter V, I, P and kWh per session with the PZEM-004T over Modbus (design target).
- Apply configurable tariffs (■/kWh + session fee, peak/off-peak slots) in a live billing engine.
- Switch charging via a 40 A contactor with e-stop input and overload cut-off; show live figures on the unit OLED.
- Generate session receipts and daily revenue summaries on the dashboard; document mains-safety throughout.

05. Proposed Methodology

On session start the ESP32 closes the contactor; the PZEM-004T streams electrical parameters that publish over MQTT every 2 s and accumulate locally as backup. The billing engine multiplies session kWh by the active tariff slot plus fixed fee. Stop triggers — driver action, current-taper full-charge detection, or overload — open the contactor and finalize the receipt. A buyer-run calibration procedure verifies metering against a reference load.

06. System Architecture / Working

- Power path: supply → 40 A contactor → EVSE socket; PZEM-004T in series for metering.

- Control: ESP32 (Modbus polling, MQTT publish, contactor drive, e-stop/overload logic) + OLED driver display.
- Billing: dashboard service applies tariffs, renders live bill, archives receipts in SQLite (single-file demo included).
- Safety: overload cut-off, e-stop input, documented mains-safety build notes; supervised assembly required.

07. Hardware / Software Requirements

- Hardware: ESP32, PZEM-004T energy meter, 40 A contactor, 0.96" OLED, e-stop button, enclosure, EV socket/cable assembly.
- Software: Arduino/ESP-IDF firmware (Modbus, MQTT); Mosquitto; Python/Flask or Node.js billing backend; SQLite.
- Host: a local always-on PC or Raspberry Pi for broker + dashboard.

08. Expected Outcomes

A working metered charge point producing live bills, session receipts and revenue summaries on the dashboard. Metering accuracy is verified by the buyer-run calibration; the documentation plainly states educational-meter status — for internal cost-splitting, not legal metrology — and that payment collection is manual in the base build.

09. Applications

- Residential society EV cost-splitting.
- Campus and office charge points.
- Small fleet depot charging records.
- Electrical-measurement coursework.

10. Future Scope

- Payment-gateway integration (UPI).
- RFID/QR driver authentication.
- OCPP-compatible cloud uplink.
- Load-balancing across multiple charge points.