PROJECT ABSTRACT

IoT Escalator Health Monitor with Vibration and Performance Dashboard

A wireless sensor node that tracks escalator vibration, motor current, step speed and gearbox temperature, streaming data to a dashboard with waveform/spectrum analysis, fault logging and maintenance suggestions. Delivered built-to-order with the working node, firmware, dashboard, documentation, report, PPT and viva kit.

01. Title

IoT Escalator Health Monitor with Vibration and Performance Dashboard

02. Background

Escalators in malls and metro stations run 12–18 hours a day, and their failures are disruptive and occasionally dangerous: step-chain elongation, bearing wear and drive faults develop silently until a breakdown or an emergency stop. Maintenance today is mostly schedule-based — technicians inspect at fixed intervals whether the machine needs it or not. Vibration is the earliest tell: a degrading bearing or loosening chain changes the machine's vibration signature weeks before failure. An IoT node that watches vibration RMS, motor current, step speed and gearbox temperature, and trends them on a dashboard, turns maintenance from calendar-based to condition-based. This project builds that node.

03. Problem Statement

Facility teams lack continuous, low-cost visibility into escalator mechanical health, so developing faults are found only at scheduled inspections or after a failure. The project addresses this with a wireless ESP32 sensor node (MEMS accelerometer plus current, speed and temperature sensing) that streams health data to a dashboard offering vibration waveform and spectrum views, a 7-day trend, an event/fault log and plain-language maintenance suggestions — advisory only; it never commands the escalator controller.

04. Objectives

- Measure tri-axial vibration with a MEMS accelerometer and compute RMS, crest factor and dominant frequency.
- Monitor motor current, step speed and gearbox temperature alongside vibration.
- Stream data wirelessly from the ESP32 node to a web dashboard.
- Render vibration waveform and frequency-spectrum views plus a 7-day RMS trend with alert threshold.
- Maintain an event/fault log and generate plain-language maintenance suggestions from trend rules.
- Deliver the full student kit: node, firmware, dashboard, documentation, report, PPT and viva Q&A.;

05. Proposed Methodology

The system is developed in four stages. (1) Sensing: a MEMS accelerometer is magnet-mounted at the drive-end bearing; a current sensor, step-speed pickup and temperature probe cover the remaining channels. (2) Edge processing: the ESP32 samples vibration at ~1.6 kHz, computes RMS/crest-factor/dominant-frequency features, and shows a live bar graph on the node's OLED. (3) Telemetry and backend: features and periodic raw windows are published over Wi-Fi to a web service. (4) Dashboard and rules: the UI renders gauges, waveform and spectrum views, the trend chart and the event log; threshold and trend rules raise alerts and maintenance suggestions.

06. System Architecture / Working

The sensor node mounts near the escalator drive: the accelerometer picks up bearing and chain vibration while the current sensor, speed pickup and temperature probe watch the drive. The ESP32 samples vibration at about 1.6 kHz, derives RMS, crest factor and dominant frequency each window, and publishes the feature set with motor current, step speed and temperature. The dashboard shows live gauges, a switchable time-waveform / frequency-spectrum view, a 7-day vibration-RMS trend against the alert line, and an event log. When RMS crosses the threshold or the trend slope indicates developing wear (e.g. chain elongation), the system logs a fault event and suggests the corresponding maintenance check. The node only observes — all control stays with the escalator's own controller.

07. Hardware / Software Requirements

- ESP32 development board
- MEMS accelerometer (ADXL345-class) with magnetic mount
- Current sensor (ACS712-class), step-speed pickup, temperature probe
- 0.96" OLED display (on-node live readout)
- Enclosure, cabling, mounting hardware
- Backend: Node.js/Python + database
- Web dashboard (HTML/CSS/JS)

08. Expected Outcomes

- A working node reporting vibration features, current, speed and temperature.
- A dashboard with gauges, waveform/spectrum views, 7-day trend and event log.
- Threshold and trend-rule alerts with maintenance suggestions.
- Complete documentation: report, PPT, installation notes and viva Q&A.;

09. Applications

- Mall and metro escalator condition monitoring
- Facility-management predictive-maintenance pilots
- Teaching vibration analysis and IoT telemetry
- A base for multi-escalator fleet dashboards

10. Future Scope

- On-device FFT with automatic fault-frequency (bearing defect) matching
- LoRa/cellular variant for sites without Wi-Fi
- Fleet dashboard covering many escalators with comparative analytics
- Integration with computerized maintenance-management (CMMS) ticketing

11. Declaration

This is a built-to-order IoT project. Vibration thresholds and sampling figures are design targets; fault-detection rules are heuristic advisories, not certified diagnostics. The system is strictly monitoring-only and must never be wired into escalator safety circuits.