

PROJECT ABSTRACT

ThingsBoard IoT Dashboard using ESP32

An ESP32 sensor node integrated with ThingsBoard Community Edition over MQTT: widget dashboards, RPC downlink control, rule-chain alarms and CSV history export on a self-hosted open-source IoT platform. Delivered built-to-order with firmware, platform configuration guide, dashboard JSON, report, PPT and viva kit.

01. Title

ThingsBoard IoT Dashboard using ESP32

02. Introduction / Background

Students can make an ESP32 read a sensor in an afternoon, but the moment data must be seen, stored, alarmed and controlled from anywhere, projects stall on dashboard plumbing. ThingsBoard Community Edition is the real open-source answer used in industry pilots: an MQTT/HTTP/CoAP device platform with a widget dashboard builder, device attributes, server-side RPC and a rule-chain engine for alarms and processing. This project builds the complete loop on it: an ESP32 node publishing DHT22 telemetry, a dashboard of gauges, timeseries charts and control widgets, RPC downlinks switching a relay and LED, rule chains raising email and Telegram alarms, and CSV export of history — all on the student's own self-hosted instance, so the report documents a genuine platform deployment.

03. Problem Statement

Sensor-node projects usually end at serial-monitor readings because building the visualization, storage, alarming and control layers from scratch is disproportionate effort. The project addresses this by integrating the ESP32 with ThingsBoard CE — telemetry in, dashboards/alarms/control out — with a reproducible setup guide, an importable dashboard JSON and documented rule chains, demonstrating production-style IoT platform skills instead of another toy chart page.

04. Objectives

- Publish DHT22 telemetry from the ESP32 to ThingsBoard CE over MQTT with access tokens.
- Build a widget dashboard (gauges, timeseries charts, value cards, alarm widget) exported as JSON.
- Implement RPC downlink control of a relay and LED from dashboard switches.
- Design rule chains that raise threshold alarms with email and Telegram notification.

- Use device attributes for firmware version, location and configuration metadata.
- Export historical timeseries to CSV for report analysis.
- Document the multi-device provisioning pattern.
- Deliver firmware, configuration guides, report, PPT and viva Q&A.

05. Proposed Methodology

The work proceeds in four stages. (1) Node: ESP32 firmware reads the DHT22, publishes telemetry over MQTT with the device access token, and handles RPC requests to drive the relay and LED. (2) Platform: ThingsBoard CE is installed on the student's laptop, Raspberry Pi or VM; the device is provisioned with its token. (3) Presentation: the dashboard is composed from widgets and exported as JSON; the rule chain (filter → create alarm → email/Telegram nodes) is configured with clear conditions. (4) Evaluation: history is exported to CSV and the report analyses telemetry reliability, alarm latency and RPC round-trip behavior from the student's own deployment.

06. System Architecture / Working

The ESP32 publishes temperature and humidity as MQTT telemetry; ThingsBoard stores the timeseries and the dashboard widgets render gauges, charts and cards. Dashboard switches send RPC requests down to the ESP32, which drives the relay/LED and reports state back as telemetry. The rule chain evaluates each message against thresholds, creating alarms with the platform's active/acknowledged/cleared lifecycle and email/Telegram notifications. Attributes carry slow-changing metadata editable from the device page without firmware changes.

07. Hardware / Software Requirements

- ESP32 development board (Wi-Fi)
- DHT22 temperature-humidity sensor
- Relay module and LED for RPC control targets
- ThingsBoard Community Edition (self-hosted on laptop, Raspberry Pi or VM)
- MQTT device transport with per-device access tokens
- Email and Telegram credentials for alarm notification
- Arduino IDE with PubSubClient (C/C++ firmware)

08. Expected Outcomes

- A live ThingsBoard dashboard fed by the student's own ESP32 node.
- Working RPC control of relay and LED from dashboard widgets.
- Rule-chain alarms with email and Telegram delivery on threshold crossings.
- CSV history exports feeding the report's analysis.

- A complete report, PPT and viva Q&A on MQTT, ThingsBoard concepts and rule chains.

09. Applications

- Teaching production IoT platform skills: telemetry, dashboards, RPC, alarms.
- A reusable backend for any ESP32 sensor project needing visualization.
- Demonstrating self-hosted vs cloud IoT deployment tradeoffs.

10. Future Scope

- Additional ESP32 nodes with per-device tokens on the same dashboard.
- ThingsBoard Edge deployment study for offline-first sites.
- Custom widget development for project-specific visualizations.
- OTA firmware update flow via ThingsBoard.