PROJECT ABSTRACT

IoT Bus Occupancy Display System

An ESP32-based bus occupancy system: door IR counters track passengers on board, and a P10 LED matrix at bus stops shows seats free, next stop and arrival time.

Project type: Hardware + Software (Hybrid)

Availability: Built to order

Suitable branches: IoT & Embedded, Electronics / E&TC

Technologies: ESP32, IR break-beam sensors, P10 LED matrix, GPS module, web dashboard

Price: Request a quotation

Prepared by Projectech · projectech.in — for academic project reference.

1. Introduction / Background

Bus passengers decide whether to board without knowing if seats are free; during peak hours this means crushed full buses while the next one runs half empty. Transport operators likewise lack real-time load data for scheduling.

This project builds a bus occupancy system: an ESP32 on each bus counts boardings and alightings with IR break-beam pairs at the doors, snaps counts to stops with a GPS geofence, and drives a P10 LED matrix display at the bus stop showing seats free, standees, next stop and expected arrival. A depot dashboard shows live occupancy across the fleet and per-stop load along each route.

2. Problem Statement

Passengers at bus stops have no visibility of crowding before the bus arrives, and depots plan schedules on tickets sold rather than actual loads. Existing AVL systems track location but not occupancy.

This project adds the missing occupancy layer: door sensors count passengers on and off, GPS geofences correct drift at every terminus reset, stop displays inform waiting passengers, and the fleet dashboard gives the depot live load per bus and per route segment.

3. Objectives

- Count boardings and alightings with IR break-beam pairs at bus doors.
- Correct count drift with GPS geofence stop detection and terminus resets.
- Drive a P10 LED matrix stop display showing seats free, standees, next stop and arrival estimate.
- Publish live occupancy per bus to a depot dashboard with fleet-wide status.
- Log per-stop boarding/alighting data for route load analysis.

4. Proposed Methodology

- Mount IR break-beam pairs at the demo bus door; wire to the ESP32 with the direction-detection state machine from the gate-counter design.
- Read GPS position; when the bus enters a stop geofence, snapshot the count against that stop; zero and re-sync counters at the depot terminus.
- Render the stop display frame on the P10 LED matrix: route number, seats free, standees, next stop and arrival minutes.
- Publish occupancy over Wi-Fi/cellular to the dashboard; draw the route load curve, fleet cards and event log.
- Validate with scripted boarding scenarios on the demo rig: board/alight known counts and check the display and dashboard agree.

5. System Architecture / Working

The edge layer has two parts: the bus unit (ESP32 + door IR pairs + GPS) that maintains the signed passenger count and publishes it, and the stop display unit (ESP32 + P10 LED matrix) that renders the latest frame for its stop. Counts are corrected at every terminus reset so sensor drift cannot accumulate across trips.

The application layer is the BusSeat Live depot dashboard: a preview of exactly what passengers see on the LED board, the occupancy-along-route chart against the 40-seat capacity line, fleet cards with per-bus load and next stop, and an event log of capacity events, resets and offline units.

6. Hardware / Software Requirements

- Hardware: 2x ESP32 development boards (bus unit + stop display), IR break-beam sensor pairs, NEO-6M GPS module, P10 LED matrix display panel, cardboard bus model for the demo rig, 5 V power supplies, breadboard and jumper wires.
- Software: Arduino IDE with TinyGPS++ and DMD2/P10 libraries, Wi-Fi network, web dashboard (single-file HTML/JS app included).

7. Expected Outcomes

- Live passenger counting with terminus-reset drift correction.

- A working P10 LED stop display showing seats free, next stop and arrival time.
- Fleet dashboard with per-bus occupancy and route load curves.
- Event logging for full-bus, offline-unit and reset events.
- A documented build: wiring diagram, firmware, dashboard and test scenarios.

8. Applications

- College and campus shuttle services.
- City bus feeder routes and BRTS corridors.
- Corporate employee transport fleets.
- Tourist hop-on/hop-off bus services.
- Any shared-vehicle service needing load visibility.

9. Future Scope

- Cellular (4G) uplink for real on-road deployment.
- Passenger mobile app mirroring the stop display.
- Demand-responsive scheduling suggestions from load history.
- Fare integration: occupancy-linked dynamic headways.
- CCTV-based cross-check of IR counts with edge vision.