PROJECT ABSTRACT

IoT Nurse Call Button System using ESP32

A wireless nurse call system with ESP32 bedside call buttons, a nurse-station console with LCD and buzzer, bed-number identification and response-time logging on a local Wi-Fi dashboard — an academic IoT prototype delivered built-to-order with hardware, firmware, report, PPT and viva kit.

01. Title

IoT Nurse Call Button System using ESP32

02. Introduction / Background

In hospital wards, patients need a fast, reliable way to reach nurses, and nurses need to know exactly which bed is calling. Wired call-bell systems are expensive to retrofit into older buildings and keep no record of how long a call waited — yet delayed response to a patient call is a genuine patient-safety concern. A wireless IoT call system answers both problems: every bed gets a simple button, the central console identifies the calling bed instantly, and call-to-response times are logged for shift review. Built on ESP32 nodes communicating over local Wi-Fi (MQTT or ESP-NOW), this project demonstrates wireless alerting, central monitoring, an acknowledgement workflow and logging at demonstration scale. It is an academic prototype for learning IoT alerting and is not certified medical equipment.

03. Problem Statement

A ward needs patient-to-nurse alerting that is wireless, identifies the calling bed, confirms acknowledgement and records response times — without cloud accounts or building rewiring. The project builds this with ESP32 bedside units, a central console with LCD/buzzer/beacon, per-bed IDs, an acknowledge/cancel workflow, timestamp logging and a local web dashboard, all running on the ward's own Wi-Fi network.

04. Objectives

- Build 2–4 wireless bedside call units (ESP32, illuminated call button, cancel button, status LED, USB/battery power), each with a configurable bed ID.
- Build a central nurse-station console (ESP32, 16×2 LCD, buzzer, beacon lamp) that queues active calls in order.
- Implement the message link over local Wi-Fi using MQTT (local broker) or ESP-NOW, with a design target latency under 2 s.

- Implement the acknowledgement workflow: nurse acknowledges at the console or dashboard, the bedside lamp confirms, and call/cancel clears the entry.
- Log call-raised and acknowledged timestamps per bed and serve a single-page live dashboard from the central unit with no cloud account.
- Add a configurable escalation timer that re-alerts unacknowledged calls with a distinct tone.

05. Proposed Methodology

Each bedside unit sleeps until its call button is pressed, then wakes, transmits its bed ID and returns to low-power listening for the acknowledgement state. The central console receives the message, appends the bed to its active-call queue on the LCD, sounds the buzzer and lights the beacon. The nurse presses the acknowledge button (or the dashboard button); the console broadcasts the acknowledgement, the bedside LED changes state, and both timestamps are written to the log. Cancel at the bedside clears the call. The escalation timer re-sounds unacknowledged calls after the configured interval. Firmware is written in the Arduino IDE; the dashboard is a single HTML/CSS/JS page served directly by the central ESP32.

06. System Architecture / Working

The system has three parts. (1) Bedside nodes: ESP32, call/cancel buttons, status LED, USB 5 V with battery option. (2) Central console: ESP32, 16×2 I2C LCD, buzzer, beacon lamp, acknowledge button, timestamp log in flash. (3) Dashboard: a local web page showing live bed status and the response log. All traffic stays on the local Wi-Fi network; no internet or cloud service is required. Range is indoor-Wi-Fi dependent (approx 10–30 m typical).

07. Hardware / Software Requirements

- ESP32-WROOM-32 modules (bedside units + central console); ESP32-C3 variants optional
- 16×2 LCD with I2C backpack, buzzer, beacon lamp, push buttons, LEDs
- Arduino IDE firmware; MQTT local broker or ESP-NOW link
- Single-file HTML/CSS/JS dashboard served by the central ESP32
- USB 5 V power; battery option for bedside units
- Local Wi-Fi network (no internet required)

08. Expected Outcomes

A working wireless call system: pressing a bedside button raises a bed-identified alert at the console within the design-target latency, acknowledgement clears it with logged timestamps, the dashboard shows live status, and escalation re-alerts unanswered calls. The report documents the protocol, the measured behaviour of the student's

own build and the prototype's limits — including the mandatory statement that it is an academic prototype, not certified medical equipment and not for life-critical use.

09. Applications

- Teaching aid for IoT coursework: wireless nodes, MQTT/ESP-NOW messaging, central monitoring
- Template for other alerting systems (lab assistance calls, workshop help points)
- Demonstration of acknowledgement workflows and response-time logging
- Base for extensions such as cloud push alerts or ward-map displays

10. Future Scope

- Cloud push/SMS escalation for off-station nurses
- Larger bed counts with a ward-map dashboard
- Battery-life optimisation with deeper sleep scheduling
- Two-way voice or canned-message responses
- Fall-detection or vitals-threshold auto-alerts (with appropriate medical disclaimers)