

Voice-Controlled Home Automation Hub using ESP32

An ESP32 relay hub switching four AC loads by voice (phone-assistant webhooks), web panel and physical buttons, with scheduling and overload sensing.

1. Introduction / Background

Voice is the interface most people actually use for smart homes, yet student builds often stop at a Bluetooth button app.

This hub switches four real AC loads through an opto-isolated relay board, commanded by voice phrases via the phone assistant, a web panel, or physical buttons — all sharing one device-state model in firmware.

2. Problem Statement

Typical student automation projects desynchronize when controlled from multiple interfaces, lack scheduling, and ignore AC switching safety. This project addresses all three: one state table for voice/panel/buttons, a local scheduling engine, and documented relay safety practice.

3. Objectives

- Switch four AC loads via voice commands, web panel and manual buttons without state conflicts.
- Run time schedules locally on the ESP32, independent of phone or cloud.
- Add overload sensing and a master kill-switch for safe demonstration.

4. Proposed Methodology

Spoken phrases are recognized by the phone assistant and converted by cloud routines into webhook calls; the hub receives them over MQTT and the ESP32 updates its single device-state table before driving the relays. Schedules stored in flash fire locally each minute. Optional ACS712-class sensors flag per-channel overloads.

5. System Architecture / Working

- Control: ESP32 + 4-channel opto-isolated relay board; per-channel LED and manual button.
- Voice path: phone assistant → cloud webhook → hub endpoint → MQTT → ESP32.
- Logic: one shared state table; flash-stored schedules; overload flags; master kill-switch.

6. Hardware / Software Requirements

- ESP32 dev board, 4-channel relay board, phone assistant + cloud webhook routines, web control panel, MQTT broker, optional current sensors, push-buttons.
- Arduino IDE (C/C++); 5 V DC logic; demo AC loads only.

7. Expected Outcomes

- A working 4-load hub responding to voice, panel and buttons in sync.
- Demonstrated schedules, kill-switch and overload flagging.
- A complete report, PPT and viva Q&A; on relay interfacing, MQTT and AC safety.

8. Applications

- Home/room automation demonstration.
- Teaching voice-webhook integration and state synchronization.
- Assistive switching for limited-mobility users (demo scale).

9. Future Scope

- Energy metering per channel.
- Presence-based automation with PIR.
- Local offline voice keyword spotting.

10. Conclusion

The hub shows that the hard part of voice automation is not recognition but coherent state, scheduling and safety — and solves those in firmware, honestly scoped to demo loads.