DoorView: Smart Video Doorbell System

Semester Project Documentation

Embeded IOT

Nexus Lab

Submitted by:

Azhar Faiz

Ali Ejaz

Ali Faisal

1. Introduction

Traditional doorbell systems provide only basic alert functionality without visual verification or remote interaction. Commercial smart doorbells depend heavily on cloud services, leading to latency, privacy risks, and recurring costs. This project proposes an offline, LAN-based Smart Video Doorbell using ESP32 microcontrollers.

2. Problem Statement

Existing solutions are either limited in functionality or expensive and cloud-dependent. There is a need for a low-cost, privacy-focused smart doorbell that operates entirely on a local network.

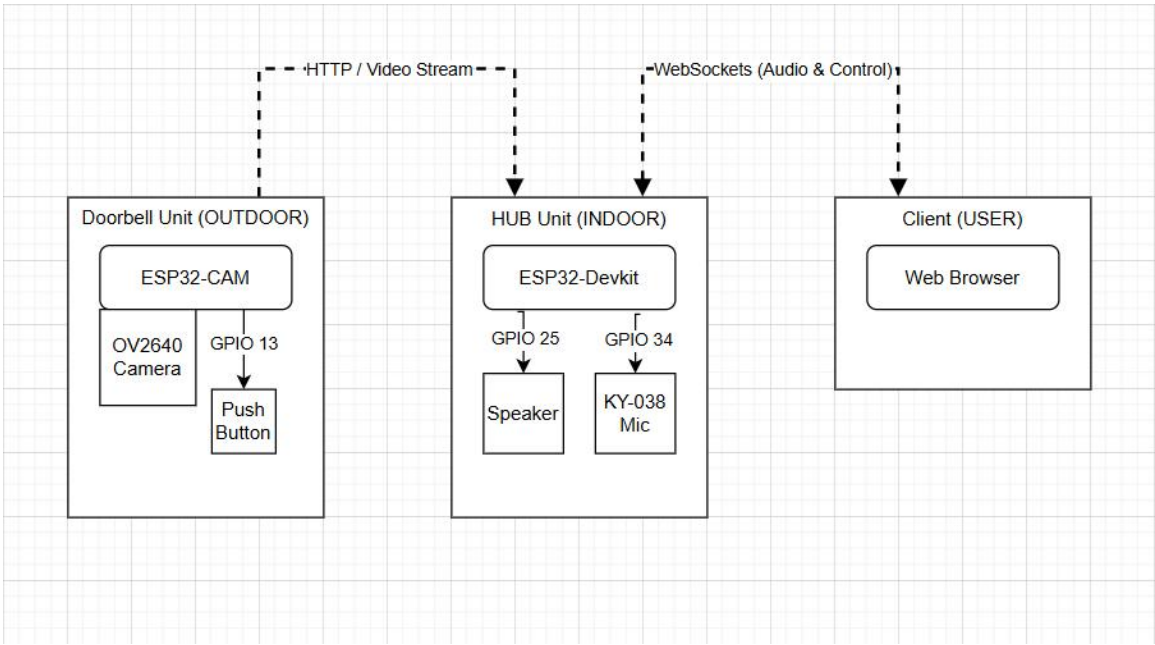
3. Objectives

- Design a LAN-based smart video doorbell
- Enable real-time video and two-way audio
- Eliminate cloud dependency
- Ensure low latency and data privacy

4. System Architecture

The system consists of three main components: Hub (ESP32 Dev Module), Doorbell Unit (ESP32-CAM), and Client (Web Browser). The Hub manages networking, audio, and the web interface, while the ESP32-CAM handles video streaming and button detection.

4.1 Block Diagram



5. Hardware Implementation

5.1 Hardware Components

- ESP32 Dev Module (Hub)
- ESP32-CAM (Doorbell)
- KY-038 Microphone Module
- Speaker
- Push Button

5.2 Software Stack

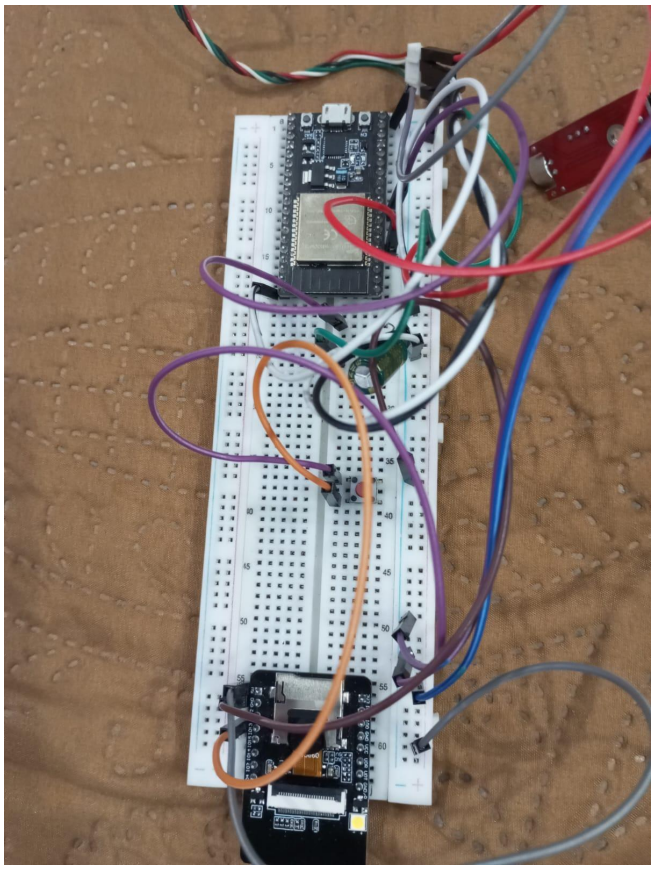
Platform: Arduino IDE (C++)
Network Mode: Wi-Fi SoftAP
SSID: Smart_Doorbell_Hub
Hub IP: 192.168.4.1 Hub Password: password123
Protocols: HTTP, WebSockets
Audio Sampling Rate: 8000 Hz
Chunk Size: 256 bytes

5.3 Hub Unit Wiring

Component	Pin	ESP32 Pin	Description
Speaker	black	GND	Ground
Speaker	white	GND	Ground
Speaker	red	5V	Power of Speaker
Speaker	green	GPIO 25	Audio Output
KY-038 Mic	AO	GPIO 34	ADC Input
KY-038 Mic	VCC	3.3V	Power
KY-038 Mic	GND	GND	Ground

5.2 Doorbell Unit Wiring

Component	Pin	ESP32-CAM Pin	Description
Push Button	Signal	GPIO 13	Ring Trigger
Push Button	GND	GND	Ground
Power	VCC	5V	Power Input
Power	GND	GND	Ground



6 Functional Workflow

The system operates through a defined event-driven architecture:

Event Trigger:

When the visitor presses the tactile button on the Doorbell Unit, the ESP32-CAM initiates a standard HTTP GET request to the Hub (/ring endpoint).

Hub Processing:

Upon receiving the signal, the Hub performs three parallel actions:

- Sets the internal isRinging state to true.
- Generates a synthetic sine wave (Ding-Dong chime) via the internal DAC (GPIO 25).
- Broadcasts a "RING" command via WebSocket (Port 81) to all connected clients.

Client Response:

The client's browser receives the command, immediately unhides the video element, and loads the MJPEG stream directly from the Camera Unit.

Audio Pipeline:

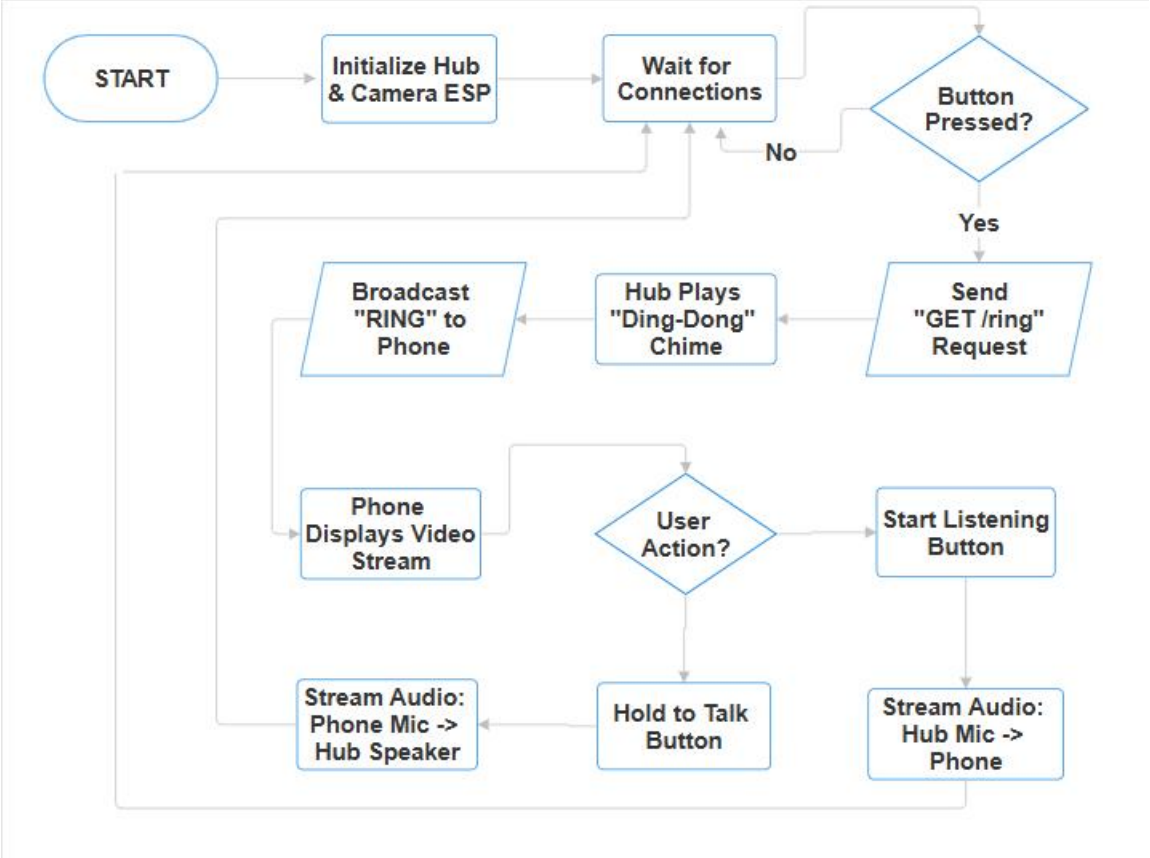
Listen Mode:

The Hub samples the KY-038 microphone at 8kHz, buffers the data into 256-byte packets, and streams it via WebSocket Port 82.

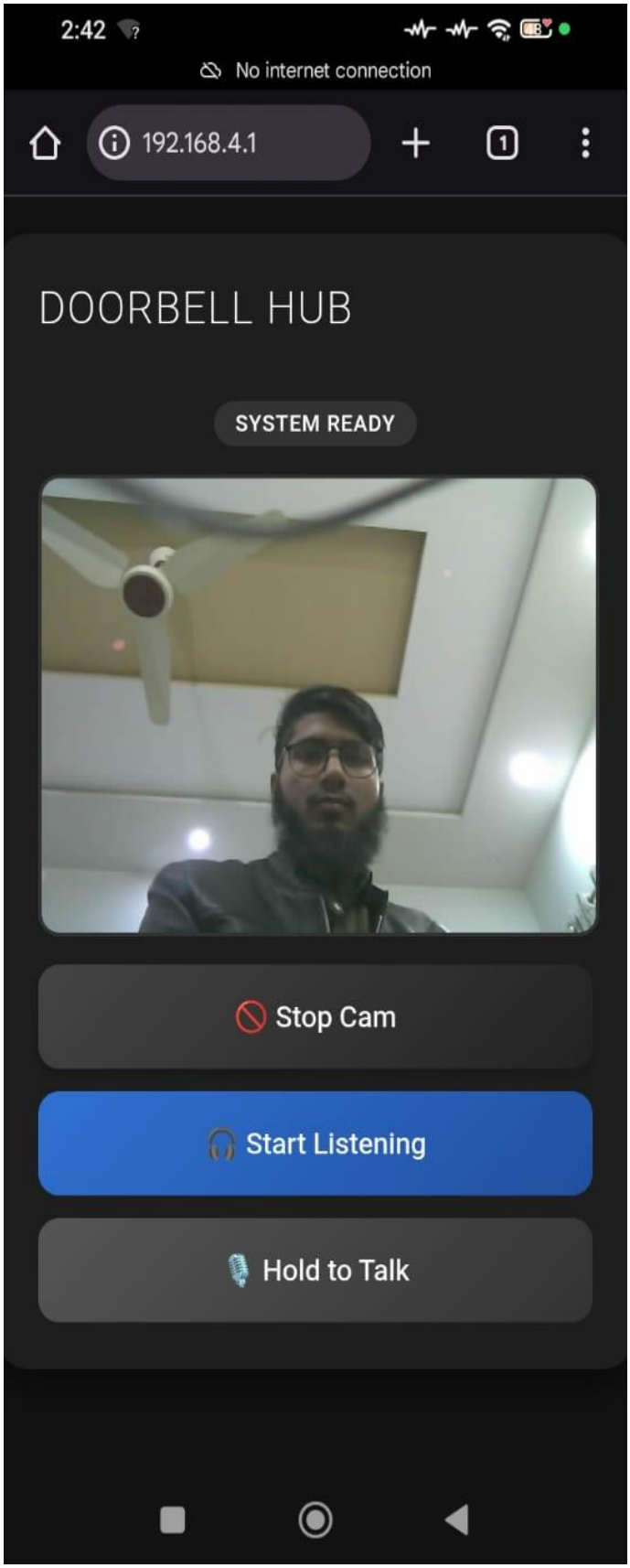
Talk Mode:

The browser captures audio, downsamples it to 16kHz to match the ESP32's capabilities, and streams raw binary data to WebSocket Port 81, which the Hub writes directly to the speaker.

7.1 Flow Chart




2:39
No internet connection
192.168.4.1
DOORBELL HUB
SYSTEM READY
No Signal
View Cam
Start Listening
Hold to Talk


2:42
No internet connection
192.168.4.1
DOORBELL HUB
SYSTEM READY
Stop Cam
Start Listening
Hold to Talk

8. Browser Configuration (Chrome Security Bypass)

Modern browsers enforce strict security policies that block microphone access (getUserMedia API) on non-secure (HTTP) origins. Since this local network operates without an SSL certificate, a manual override is required on client devices.

- **Flag:** chrome://flags/#unsafely-treat-insecure-origin-as-secure
- **Procedure**: Enable the flag and explicitly whitelist the Hub's IP address (http://192.168.4.1).
- **Result:** This forces the browser to treat the local Hub as a trusted origin, unlocking full two-way audio functionality.

9. Conclusion

The ESP32 Smart Video Doorbell System successfully meets the design criteria for a decentralized home security solution. By leveraging the dual-core architecture of the ESP32 and the dedicated camera interface of the ESP32-CAM, the system achieves zero-latency video streaming and real-time audio without reliance on external internet or cloud subscriptions. This project proves that high-performance IoT devices can be built cost-effectively while maintaining strict user privacy and data sovereignty.

10. Future Enhancements

- **Enclosure Design:**

  Development of a 3D-printed, weather-resistant case to protect the outdoor unit from moisture and dust.

- **Power Optimization:**

  Implementation of ESP32 "Deep Sleep" modes combined with "External Wake-up" triggers to allow the Doorbell unit to run for months on a single Li-ion battery.

- **Night Vision:**

Integration of an Infrared LEDs to provide clear visibility in low-light conditions.

- **Mobile Application:**

  Development of a dedicated Flutter or React Native app to replace the browser configuration step and provide native push notifications.