# ESP ChatBot: Offline-First Voice Interface on ESP32

Wake Word Detection, Audio Capture, and MQTT/HTTP Streaming

Abdul Baseer

AI — Embedded AI — ESP32 Developer

August 20, 2025

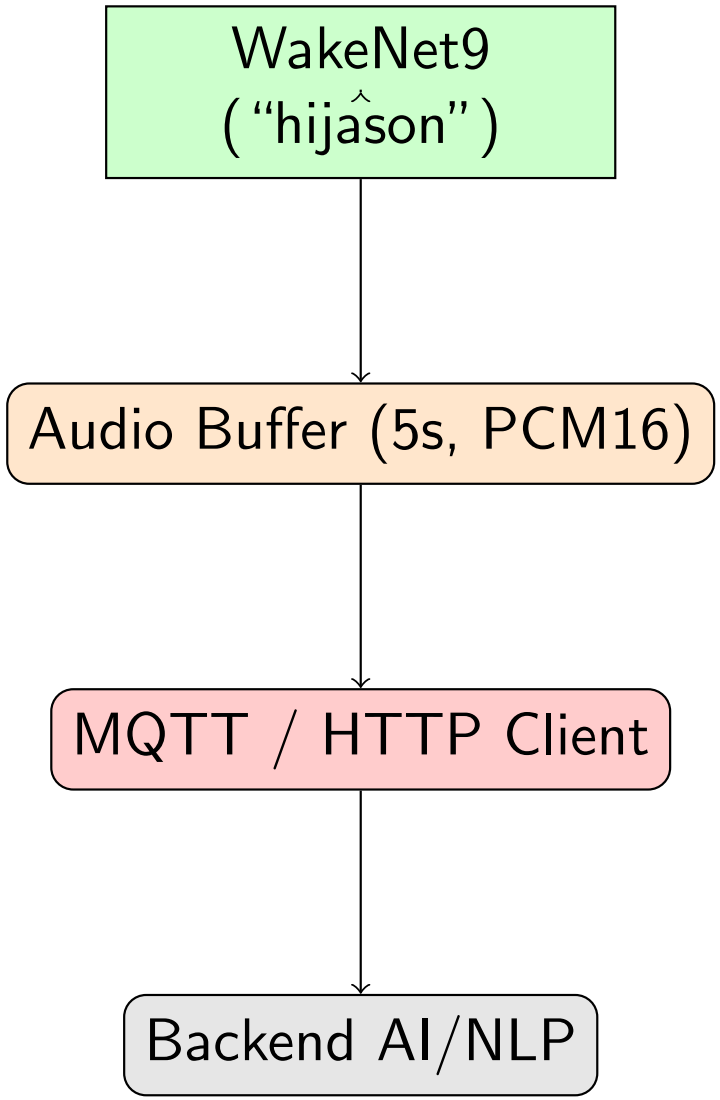
# Agenda

# What is ESP ChatBot?

- Lightweight, offline-first voice interface framework
- Runs on ESP32-S3 with ESP-IDF v5.x
- Built on Espressif's ESP-SR v2.0.5
- Detects custom wake word "**hijason**"
- Captures audio from I2S microphone
- Publishes PCM audio via MQTT or HTTP

# System Architecture

- Wake Word Detection: WakeNet9
- Audio Capture: I2S, 16kHz, PCM16
- Networking: MQTT / HTTP
- Modular components:
  - `custom_audio`
  - `custom_network`
  - `custom_system`



# Wake Word Detection

- Model: WakeNet9
- Custom keyword: **“hijason”**
- Model stored as binary array in flash (`hijason.h`)

**Integration Example:**

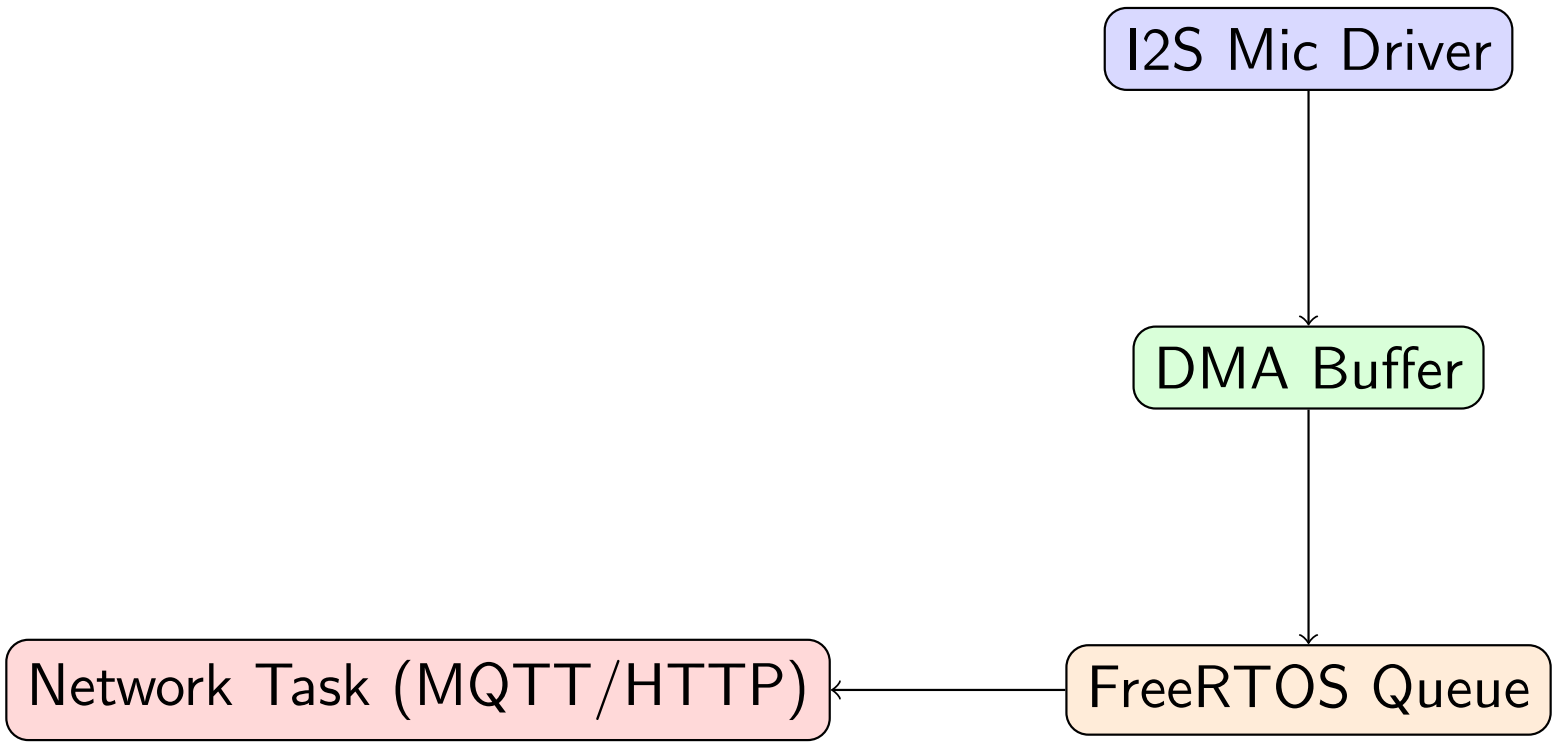
```
// Reference to WakeNet9 model
const esp_wn_iface_t *wn_iface = &ESP_WN9_MODEL;
model_iface_data_t *wn_model = (model_iface_data_t *)&
    hijason;

// Start Wake Word Detection
wn_iface->create(wn_model);
wn_iface->detect(wn_model, audio_buffer);
```

# Audio Pipeline

- Captures 5 seconds of audio after wake word detection
- Mono, 16-bit, 16kHz sampling
- Stored in ring buffer
- Processed asynchronously via FreeRTOS tasks

**Pipeline Flow:**



# Networking

- Sends audio buffer to backend server
- Protocols: MQTT or HTTP

**MQTT Example:**

```
// Publish PCM audio
esp_mqtt_client_publish(client,
    "esp/audio/hijason",
    (const char *)pcm_buffer,
    buffer_size, 0, 0);
```

**HTTP Example:**

```
POST /api/audio HTTP/1.1
Content-Type: audio/raw
Body: [binary PCM16LE data]
```

# Performance & Security

## Performance

- WakeNet9 on ESP32-S3:
- $\sim$28% average CPU load
- $\sim$39% peak CPU usage
- Optimized FreeRTOS task priorities

## Security

- MQTT not encrypted by default
- Recommend TLS + authentication
- Always validate audio buffer bounds

# Future Enhancements

- On-device Multinet for offline command recognition
- Lightweight TensorFlow Lite STT pipeline
- TTS playback (voice responses)
- Power management: light sleep between detections

# Conclusion

- ESP ChatBot enables low-power, offline-first voice interfaces
- WakeNet9 + ESP32-S3 = reliable wake word detection
- Modular design for easy integration with AI backends
- Flexible networking via MQTT and HTTP

**Questions?**