

# VideoSDK - IOT Assignment

---

## Objective

Create an ESP32-based system that detects loud sounds using a microphone, records a short audio snippet, and sends an MQTT alert containing event metadata.

## Estimated Time

2 days

## Deliverables

### 1. ESP32 Firmware

- Continuously monitor analog microphone input via ADC (I2C) using ESP-IDF.
- Detect loud sound events based on a configurable threshold.
- On detection:
  - Record 3 seconds of audio (via ADC sampling).
  - Save as `.wav` or raw `.bin` file using SPIFFS.
  - Publish an MQTT message containing metadata.

### 2. MQTT Integration

- Use a public MQTT broker (e.g., `test.mosquitto.org`).
- Topic: `esp32/audio\_alerts/{device\_id}`.
- Payload example:

```
{  
  "event": "sound_detected",  
  "device": "esp32-audio-01",  
  "timestamp": "2025-06-28T13:10:00Z",  
  "amplitude": 0.91,  
  "audio_filename": "recording_168.wav"  
}
```

---

### 3. Serial Logging

- Log:
  - Threshold crossings
  - Recording events
  - File path
  - MQTT publishing result (success/failure)

### 4. README.md

- Setup Instructions: Mic wiring, SPIFFS formatting, MQTT config
- Usage: Test using Serial Monitor and MQTT dashboard
- Known Issues

### 5. Demo Recording

- A brief (1–2 min) screen capture showing:
  - Serial log during recording
  - MQTT alert being published
  - SPIFFS file listing after recording

### Technical Constraints

- Must use ADC (I2C) for audio capture (no external audio ICs).
- Only SPIFFS or internal RAM for storage.
- MQTT messages must be JSON formatted with minimal payload.

### ★ Bonus Points (Optional)

- Implement HTTPS Dile upload to a webhook (e.g., Mailgun or Discord)
  - Attach the `.wav` or `.bin` file
  - Use secure HTTPS POST with necessary headers
  - Include metadata (timestamp, filename) in the body or subject
- Add a second MQTT message (`audio/upload\_status`) indicating success/failure of the upload.