

Hazard Analysis Audio360

Team #6, Six Sense
Omar Alam
Sathurshan Arulmohan
Nirmal Chaudhari
Kalp Shah
Jay Sharma

Table 1: Revision History

Date	Developer(s)	Change
Date1	Name(s)	Description of changes
Date2	Name(s)	Description of changes
...

Contents

1	Introduction	1
2	Scope and Purpose of Hazard Analysis	1
3	System Boundaries and Components	1
3.1	Inside the System Boundary	1
3.2	Outside the System Boundary	2
4	Critical Assumptions	3
5	Failure Mode and Effect Analysis	3
6	Safety and Security Requirements	3
7	Roadmap	3

[You are free to modify this template. —SS]

1 Introduction

A hazard is anything that prevents the Audio360 system from notifying users with important sounds near them with high precision and accuracy. For deaf and hard-of-hearing individuals who rely on the system for situational awareness, any failure to detect, classify, or display audio information could result in missed safety cues, social interactions, or environmental awareness.

This hazard analysis identifies potential failure modes in the Audio360 audio localization system and establishes safety requirements to ensure reliable operation in real-world scenarios.

2 Scope and Purpose of Hazard Analysis

The scope of this document is to identify possible hazards within the Audio360 system components, the effects and causes of failures, mitigation steps, and resulting safety and security requirements.

Potential losses that could be incurred due to system failures include physical injury from missed emergency vehicle warnings or approaching machinery, household accidents from undetected safety alerts, missed social interactions and communication opportunities, reduced independence and confidence in daily activities, and loss of user trust in the assistive technology system.

3 System Boundaries and Components

This section is broken down into two subsections: one for components within the system boundary, and one for components outside the system boundary.

3.1 Inside the System Boundary

The following components are within the system’s control and responsibility:

- **Embedded firmware:** Real-time operating system and all embedded software running on the processing unit.
- **Signal processing module:** Real-time digital signal processing algorithms including frequency domain transforms, filtering, and time-domain analysis.
- **Direction of arrival (DoA) estimation:** Algorithms for computing sound source direction on a 2D plane based on time difference of arrival and phase differences across microphones.

- **Audio classification engine:** Sound fingerprinting and classification logic to identify and categorize detected sounds (e.g., speech, vehicles, alarms).
- **Visualization Controller:** Component responsible for creating and sending visualization output to the glasses.
- **Configuration and calibration:** Microphone array calibration routines and system configuration parameters.

3.2 Outside the System Boundary

The following external entities interact with the system but are not under its direct control:

- **Users:** Individuals who are deaf or hard of hearing wearing the device. Their actions, responses to alerts, and interpretation of displayed information are outside the system's control.
- **Environmental sounds:** Acoustic signals in the physical environment, including speech, vehicle noises, alarms, and ambient sounds. The system detects and processes these but does not generate or control them.
- **Audio capture subsystem:** Synchronized sampling logic for the microphone array, including analog-to-digital conversion interfaces and buffer management. This component is included in the microcontroller and is not within our system's control.
- **Physical microphone hardware:** Microphone sensors that capture acoustic pressure waves. While the system controls their digital interface, the physical transduction mechanism is external.
- **Smart glasses hardware:** The physical display device, including its screen, optics, power management, and form factor. The system sends display commands but does not control the hardware's internal operation.
- **Microcontroller:** Component responsible for processing real time data of sensor inputs. This hardware component's performance and reliability are outside our system's control.
- **Power supply:** Battery or external power source providing electrical power to system components. Power management at the hardware level is external to the software system.
- **Physical environment:** Room acoustics, ambient noise levels, temperature, and other environmental factors that affect sound propagation and microphone performance.

4 Critical Assumptions

[These assumptions that are made about the software or system. You should minimize the number of assumptions that remove potential hazards. For instance, you could assume a part will never fail, but it is generally better to include this potential failure mode. —SS]

5 Failure Mode and Effect Analysis

[Include your FMEA table here. This is the most important part of this document. —SS] [The safety requirements in the table do not have to have the prefix SR. The most important thing is to show traceability to your SRS. You might trace to requirements you have already written, or you might need to add new requirements. —SS] [If no safety requirement can be devised, other mitigation strategies can be entered in the table, including strategies involving providing additional documentation, and/or test cases. —SS]

6 Safety and Security Requirements

[Newly discovered requirements. These should also be added to the SRS. (A rationale design process how and why to fake it.) —SS]

7 Roadmap

[Which safety requirements will be implemented as part of the capstone timeline? Which requirements will be implemented in the future? —SS]

Appendix — Reflection

1. What went well while writing this deliverable?

Sathurshan: The team had a strong understanding of the safety-critical nature of the system. Some members of the team also had experience in writing a Hazard Analysis document from extra-curricular activities. This made performing the hazard analysis more straightforward, as we were able to effectively identify and evaluate potential risks within the system.

2. What pain points did you experience during this deliverable, and how did you resolve them?

Sathurshan: Some sections of this deliverable were dependent on the completion of other sections within this document and the SRS. As a result, managing the timeline for these sections was challenging. To resolve this, we coordinated closely with the owners of the dependent sections, which improved collaboration and allowed us to exchange constructive feedback to ensure consistency across the document.

3. Which of your listed risks had your team thought of before this deliverable, and which did you think of while doing this deliverable? For the latter ones (ones you thought of while doing the Hazard Analysis), how did they come about?

Sathurshan: Before this deliverable, I had already identified the risk of microphone failures, as it serves as the system's primary input. During the hazard analysis, we also identified the risk of unauthorized access to system components, particularly data stored on the microcontroller and modification of the software. This risk emerged as we considered non-physical hazards, which the following questions allude to.

4. Other than the risk of physical harm (some projects may not have any appreciable risks of this form), list at least 2 other types of risk in software products. Why are they important to consider?

Sathurshan: Other forms of software-related risks include data privacy violations and security vulnerabilities that allow unauthorized access without user consent. These are critical to consider because they can use user data can be used against them, resulting in ethical consequences.