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
2025-10-06	Nirmal, Sathurshan, Omar, Kalp, Jay	Initial Write-up
2025-10-26	Sathurshan	Update sampling rate requirement

Contents

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 degraded 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

1. Microphone
   - All microphones are able to capture frequencies between at least 250 Hz to 8kHz, which is the human hearing audio frequency range. [1].
   - Microphones operate correctly within normal operating temperatures of 0-35°C.
2. Microcontroller
   - The analog-to-digital converters (ADC) that comes with the hardware provides stable conversions in real-time.
   - Processor's clock frequency remains stable under normal operating temperatures 0-35°C.
3. Output Display
   - Visual indicators on the smart glasses remain visible in typical lighting conditions.

4. System Integration
    - Voltage and power is maintained by the controller firmware itself and we do not need to manage that in software.

5 Failure Mode and Effect Analysis

5.1 Severity Mapping Table

The severity, occurrence, and detection ratings used in the FMEA table are defined as follows:

Rating	Severity	Occurrence	Detection
1	Negligible	Rare	Always detected
2	Minor	Uncommon	Easy to detect
3	Major	Occasional	Moderately difficult to detect
4	Critical	Frequent	Difficult to detect
5	Catastrophic	Very frequent	Impossible to detect

5.2 Priority Mapping Table

The priority level is determined by the summation of the severity, occurrence, and detection ratings:

Product of severities	Priority Level
1 - 4	Low Priority
5 - 10	Medium Priority
10 - 15	High Priority

#	Component / Function	Potential Failure Mode	Effect on User / System	Likely Cause(s)	Severity	Occurrence Frequency	Detection Method	Detection Likelihood	Priority (S + O + D)	Recommended Action	Relevant Requirement(s)
1	Microphone	Unresponsive / Distorted Microphone.	Not able to effectively localize audio. Unable to provide warnings to user.	Microphone circuit failure. Microphone damage. Excessive ambient noise.	5	1	Multiple corrupted microphone data frames. Excessive white noise detection. Short circuit detection for microphones.	3	9	Detect failure using microphone audio (or lack thereof) and notify user.	FR1.4, FR2.3, FR3.5, FR7.2, NFR2.1
2	Visualization Controller	Disconnection from display.	Unable to provide visual notifications to user.	Depending on connection type: cable, wireless interference, dropped connection.	3	2	Loss of connection signal. Failure to send data to display.	1	6	There are no safety requirements that can mitigate this hazard.	NFR2.1
3	Embedded Firmware	System crashes and freezes.	The system remains unusable until it is restarted. Unable to provide notifications to user.	Software bugs. Insufficient Error Handling Insufficient Requirements.	5	3	System watchdog timer Halt Interrupt.	1	9	Implement a watchdog timer to reset the system in the event of a crash. Implement adequate logging to diagnose the cause of firmware failure during post-mortem. Robust testing and error handling.	FR1.3, FR4.1, FR4.4
4	Sound Detection (Audio360 Engine)	Failure to detect important sounds.	Failure to notify user of critical sounds.	Insufficient microphone quality. Poor classification algorithm. Excessive ambient noise.	4	3	This is difficult to detect without extensive testing. Impossible to detect in production.	5	12	Have a thorough library of sounds to detect and test against. Measure microphone performance on sound library.	FR3.5, FR4.4, FR5.1, FR6.1
5	Sound Classifier (Audio360 Engine)	Misclassification of sounds.	Notify the user of the wrong sounds.	Insufficient classification algorithm Insufficient microphone quality.	3	2	This is difficult to detect without extensive user testing. Impossible to detect in production.	5	10	Extensive testing in real-world environments and simulation. Verify microphone quality during operation.	FR4.4, FR5.1, FR5.4, NFR5.1, NFR5.2
6	Sound Localizer (Audio360 Engine)	Inaccurate direction determination for sounds.	Misinform the user of the direction of important sounds.	Incorrect localization algorithm. Insufficient microphone quality.	4	2	This is difficult to detect without extensive localization testing. Impossible to detect in production.	5	11	Extensive localization testing in real-world environments and simulation.	FR5.2, NFR5.3

5.3 Out of Scope Failure Modes

The project involves off the shelf hardware components with their own possible failure modes and possible mitigations. Since the hardware and electrical components are not being designed as part of the project, these failure modes are considered out of scope:

- Battery failure / damage.
- Physical display damage.
- Power supply issues.
- Physical damage to the glasses.

6 Safety and Security Requirements

- **SR1:** The microcontroller shall run the main Audio360 software in a closed environment. Only the approved microphones and output display shall interface to it. This ensures protection against unauthorized access.
- **SR2:** The system shall permanently discard microphone audio data immediately after completion of audio analysis. This requirement ensures data privacy and prevents unauthorized reuse of raw audio.
- **SR3:** The system shall detect and flag audio anomalies such as clipping, lost signal and silence. This enables identification of microphone or signal path faults.
- **SR4:** The system shall disable audio classification and directional analysis features until microphone faults addressed in SR3 are resolved. This prevents the generation of unreliable or unsafe outputs.
- **SR5:** The visualization controller shall present information in a non-intrusive manner, minimizing visual obstruction so users can safely perform external activities.
- **SR6:** The visualization controller component shall alert users when critical features such as sound classification or direction determination fail. This ensures that users are aware of degraded safety functions.
- **SR7:** The system shall have an audio processing success rate end to end of at least 90% over 60 seconds. This mitigates classification related hazards identified in the FMEA.
- **SR8:** The system shall notify users when a sound classification result has low confidence or is unrecognized to prevent misleading contextual feedback.
- **SR9:** The system shall estimate the direction of arrival of an audio source with a maximum error of 45 degrees. This ensures reliable directional awareness for the user.
- **SR10:** The system shall perform continuous diagnostics on all hardware components to monitor hardware errors in real-time. This ensures the system can react to failures as soon as they occur.

7 Roadmap

The following safety and security requirements are planned for implementation within the scope of the capstone project timeline:

- SR2
- SR3
- SR5
- SR7
- SR9

The following safety and security requirements are identified as stretch goals. Their implementation will depend on the available development bandwidth and project progress:

- SR4
- SR6
- SR8
- SR10

References

[1] T. Tedeschi, *Human hearing frequency and audible range*. Miracle-Ear, 2025. [Online]. Available: https://www.miracle-ear.com/blog-news/human-hearing-range

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.

   Kalp: I think having written the SRS document before this one made the hazard analysis more straightforward, as we had a clear understanding of the system and the risks associated with it. We had many discussions during the SRS document that really clarified the requirements and the expected software states of the system, really helping us quickly go through this document without much difficulty.

   Nirmal Chaudhari: For this deliverable, I focused on the critical assumptions section. What worked well while writing this deliverable was having a clear enough picture of constraints that exist with this project from the environment section of the SRS doc. From these constraints, it became clear what is out of our hands, and needs assumptions for our project to validate the requirements we have set.

   Omar: I think what went well when writing this document was the cross collaboration between myself and Sathurshan. We were able to have in depth discussions about the potential hazards and risks from different components since he was responsible for most of the technical requirements. This was instrumental in ensuring a detailed and accurate FMEA table.

   Jay: The FMEA structure provided a clear framework for thinking through potential failures systematically. Breaking down each component and considering what could go wrong helped us identify risks we might have otherwise missed. The team's collaborative approach also meant we could challenge each other's assumptions and ensure we weren't overlooking critical failure modes.

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.

   Kalp: I think the main pain point was that there was high dependence between the sections of the document. This lead to us assigning the tasks to each person in large chunks (to avoid the issues we ran into during the SRS document), but then that left us with people not having much to contribute to on the document. We ended up having to reshuffle the tasks around to help mitigate this issue.

   Nirmal Chaudhari: While working on this deliverable, it sometimes became unclear what should be listed as an assumption vs constraint. As we continued brainstorming however, we started to look at it from another angle where the assumptions are technically just listed as boundaries that our system has, and requires assumptions. Moreover, referencing between the sections other people are working on for consistency was a difficult task to do, especially within a small time frame.

Omar: One pain point I experienced was ensuring that the FMEA table was well formatted and readable. Adjusting the latex code to ensure that the table fit within the page margins while still being legible was a challenge. I resolved this by experimenting with different table types and packages, even adjusting page sizes till it worked.

Jay: Initially, it was difficult to assess the severity and likelihood of different failure modes objectively. We had to balance being thorough without being overly pessimistic about every possible edge case. I resolved this by focusing on realistic operating conditions and consulting with teammates who have more hardware experience to ground my estimates in practical constraints.

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.

   Kalp: A big risk that I think the team was was already considering before the deliverable was risk of hardware failure (microphone, smart glasses, microcontroller, etc.). This was mainly due to us just thinking mainly about the high level system, with the hardware being major components of that mental model. During the document is when I would say we started to think more about security and privacy risks, specifically with data access breach and unauthorized access to the system.

   Nirmal Chaudhari: One of the listed risks we thought of before the deliverable was not being able to find a microcontroller with 4 ADCs with the limited budget we had. This risk was highlighted to us by our supervisor from our initial meeting. One risk we though about while doing this deliverable was risks associated with the environment, which we have no control over. During our team brainstorming section, we realized that weather conditions like rain, humidity, wind are things our system will have to be robust enough to handle given the limited budget we have. To address these risks, we added in our critical assumption that for the scope of this project we will have perfect weather conditions.

   Omar: Prior to this deliverable, I had already considered the risk of system crashes and freezes due to my experience with embedded systems. During the hazard analysis, we had to delve further into the specific connection mechanism that the smart glasses provide. This led to the identification of the risk of disconnection from the display, which I had not previously considered.

   Jay: We had already considered the risk of classification errors and inaccurate direction estimation since those are core to the system's value proposition. What emerged during the hazard analysis was thinking about failures, where one component's malfunction could propagate through the system and cause misleading outputs rather than obvious failures. This came from systematically working through the FMEA table and considering how each failure mode affects downstream components.

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 user data can be used against them, resulting in ethical consequences.

Kalp: I think two forms of software-related risks that are important to consider are data privacy violations and reliability issues. With many software systems dealing with data from the user, or providing important data to the user, it's important to consider the risks of data being leaked / breached (make private information public or inform the user something wrong potentially misguiding their actions).

Nirmal Chaudhari: One other type of risk that exists with software projects is security and privacy risks. While working on the environment section of the SRS document, I realized there are a lot of legal constraints around collecting and analyzing sounds in a given environment. This is important to consider, since if its left unaddressed moving the system into a production environment would be impossible. Another type of risk that exists is environmental risks. If the battery we choose is inefficient or is harmful to the environment when being disposed of, this would negatively impact the environment around us.

Omar: Two other types of software-related risks include performance degradation over time. This is an important issues with emerging technologies that rely on machine learning models, as they may become less effective as the environment and sensors change overtime, even on the same hardware. Another risk is bad UI/UX design, which can lead to user frustration and abandonment of the product. This is especially important to consider for assistive technologies, as they need to be intuitive and easy to use for individuals with varying levels of ability.

Jay: Two important risks are loss of user confidence and poor code maintainability. If the system gives false alarms or inconsistent results, users will stop trusting it and won't rely on it when they actually need it. Maintainability is also critical because if the code is messy or poorly organized, fixing bugs later becomes risky and time-consuming, which is especially problematic for a safety device that needs to stay reliable over time.