**System Design Document**

**For**

**Audio Surveillance System**

Team members: Zachary Tauscher, Jacob Attia, Caleb Leeb, Jaclyn Welch

| Version/Author | Date |
| --- | --- |
| Version 1 | 28 September 2021 |
|  |  |
|  |  |
|  |  |

**TABLE OF CONTENT**

[1](#_heading=h.30j0zll) INTRODUCTION 3

[1.1](#_heading=h.1fob9te) Purpose and Scope 3

[1.2](#_heading=h.3znysh7) Project Executive Summary 3

[1.2.1](#_heading=h.2et92p0) System Overview 3

[1.2.2](#_heading=h.tyjcwt) Design Constraints 3

[1.2.3](#_heading=h.3dy6vkm) Future Contingencies 3

[1.3](#_heading=h.1t3h5sf) Document Organization 3

[1.4](#_heading=h.4d34og8) Project References 4

[1.5](#_heading=h.2s8eyo1) Glossary 4

[2](#_heading=h.17dp8vu) SYSTEM ARCHITECTURE 4

[2.1](#_heading=h.3rdcrjn) System Hardware Architecture 4

[2.2](#_heading=h.26in1rg) System Software Architecture 4

[2.3](#_heading=h.lnxbz9) Internal Communications Architecture 4

[3](#_heading=h.35nkun2) HUMAN-MACHINE INTERFACE 4

[3.1](#_heading=h.1ksv4uv) Inputs 5

[3.2](#_heading=h.44sinio) Outputs 5

[4](#_heading=h.2jxsxqh) DETAILED DESIGN 5

[4.1](#_heading=h.z337ya) Hardware Detailed Design 6

[4.2](#_heading=h.3j2qqm3) Software Detailed Design 6

[4.3](#_heading=h.1y810tw) Internal Communications Detailed Design 7

[5](#_heading=h.4i7ojhp) EXTERNAL INTERFACES 7

[5.1](#_heading=h.2xcytpi) Interface Architecture 7

[5.2](#_heading=h.1ci93xb) Interface Detailed Design 8

[6](#_heading=h.3whwml4) SYSTEM INTEGRITY CONTROLS 8

**INTRODUCTION**

## **Purpose and Scope**

This section provides a brief description of the Audio Surveillance System’s purpose and scope.

## **Project Executive Summary**

This section provides a description of the Audio Surveillance System from a management perspective and an overview of the framework within which the conceptual system design was prepared.

### System Overview

The goal of the Audio Surveillance System is to record sounds with at least three microphone arrays, each consisting of at least three microphones, and transfer the audio to a Raspberry Pi with bluetooth capabilities. The Raspberry Pi will transfer this information to a computer with the Audio Surveillance System’s application, which will visualize the data for the user to view. This view will display a basic map of the room under surveillance, indicate voices and footsteps in the room, and where these sounds are detected in the room. Two methods for the set up of the microphone arrays are currently being considered: omnidirectional microphone arrays using signal strength to localize sound, and directional microphone arrays that triangulate the location of the sound.

### Design Constraints

The restraints of this system are as follows: cost, disguise, hardware capability, and access to pure sound. The quality and amount of hardware we use will affect the cost and the appeal to the customer conversely. Hardware such as a data acquisition board, which would increase quality of data and assist in data processing, were removed from the system design due to cost constraints. The microphone arrays purchased for the system will be the majority of the budget. With regards to disguise, the purpose of the Audio Surveillance System is to replace traditional video surveillance with audio surveillance to make surveillance more discreet and cost effective. Using small hardware and being able to place it around a room in a discreet way will limit the hardware we can use and the arrangement of the microphone arrays. Hardware capability will determine how accurate the visualization of the sounds in a room can be. The microphone arrays’ ability to determine the strength and direction of sound accurately will affect the accuracy of the system. Lastly, to test the design and provide data for the machine learning software to differentiate between sounds, pure sounds will have to be provided to the system. The physical ability to access pure sound is a constraint that will limit the accuracy of our system.

### Future Contingencies

The current design of the system includes the integration of several microphone arrays with a Raspberry Pi to a central computer for processing using bluetooth. If the Raspberry Pi processor is not able to handle sending the data to the computer via bluetooth, we will adjust to a wired setup for the Raspberry Pi and computer.

## **Document Organization**

This section describes the organization of the Systems Design Document.

## **Project References**

This section provides a bibliography of key project references and deliverables that have been produced before this point.

## **Glossary**

Supply a glossary of all terms and abbreviations used in this document. If the glossary is several pages in length, it may be included as an appendix.

# **SYSTEM ARCHITECTURE**

In this section, describe the system and/or subsystem(s) architecture for the project. References to external entities should be minimal, as they will be described in detail in Section 6, External Interfaces.

## **System Hardware Architecture**

In this section, describe the overall system hardware and organization. Include a list of hardware components (with a brief description of each item) and diagrams showing the connectivity between the components. If appropriate, use subsections to address each subsystem.

## **System Software Architecture**

The Audio Surveillance System will

## **Internal Communications Architecture**

In this section, describe the overall communications within the system; for example, LANs, buses, etc. Include the communications architecture(s) being implemented, such as X.25*,* Token Ring, etc. Provide a diagram depicting the communications path(s) between the system and subsystem modules. If appropriate, use subsections to address each architecture being employed.

**Note:** The diagrams should map to the FRD context diagrams.

# **HUMAN-MACHINE INTERFACE**

This section provides the detailed design of the system and subsystem inputs and outputs relative to the user/operator. Any additional information may be added to this section and may be organized according to whatever structure best presents the operator input and output designs. Depending on the particular nature of the project, it may be appropriate to repeat these sections at both the subsystem and design module levels. Additional information may be added to the subsections if the suggested lists are inadequate to describe the project inputs and outputs.

## **Inputs**

The input to the system will be the sound in the room, which will be received by the microphone arrays, then uploaded to the Raspberry Pi processor by bluetooth. The Raspberry Pi will send this data to the processing computer as a serialized stream of values representing the strength and direction of the sounds, which will be processed in Anaconda.

## **Outputs**

The output of the system will be a display in a visual diagram on the screen of the computer of the sounds in the room for the user to view. This will include the type of sound a visual display of its location.

# **DETAILED DESIGN**

This section provides the information needed for a system development team to actually build and integrate the hardware components, code and integrate the software modules, and interconnect the hardware and software segments into a functional product. Additionally, this section addresses the detailed procedures for combining separate COTS packages into a single system. Every detailed requirement should map back to the FRD, and the mapping should be presented in an update to the RTM and include the RTM as an appendix to this design document.

## **Hardware Detailed Design**

A hardware component is the lowest level of design granularity in the system. Depending on the design requirements, there may be one or more components per system. This section should provide enough detailed information about individual component requirements to correctly build and/or procure all the hardware for the system (or integrate COTS items).

If there are many components or if the component documentation is extensive, place it in an appendix or reference a separate document. Add additional diagrams and information, if necessary, to describe each component and its functions, adequately. Industry-standard component specification practices should be followed. For COTS procurements, if a specific vendor has been identified, include appropriate item names. Include the following information in the detailed component designs (as applicable):

* Power input requirements for each component
* Signal impedances and logic states
* Connector specifications (serial/parallel, 11-pin, male/female, etc.)
* Memory and/or storage space requirements
* Processor requirements (speed and functionality)
* Graphical representation depicting the number of hardware items (for example, monitors, printers, servers, I/O devices), and the relative positioning of the components to each other
* Cable type(s) and length(s)
* User interfaces (buttons, toggle switches, etc.)
* Hard drive/floppy drive/CD-ROM requirements
* Monitor resolution

## **Software Detailed Design**

This section will detail the design of the software used for the Audio Surveillance System. This system will use Scikit-Learn as a library to enable machine learning in Python. This will utilize Anaconda for package management, and Spyder as the IDE while in testing and development. We will be setting up the Raspberry Pi with Raspberry Pi Imager to be able to intake the data and send it to the computer in a serialized stream.

# **EXTERNAL INTERFACES**

External systems are any systems that are not within the scope of the system under development, regardless whether the other systems are managed by the State or another agency. In this section, describe the electronic interface(s) between this system and each of the other systems and/or subsystem(s), emphasizing the point of view of the system being developed.

## **Interface Architecture**

In this section, describe the interface(s) between the system being developed and other systems; for example, batch transfers, queries, etc. Include the interface architecture(s) being implemented, such as wide area networks, gateways, etc. Provide a diagram depicting the communications path(s) between this system and each of the other systems, which should map to the context diagrams in Section 1.2.1. If appropriate, use subsections to address each interface being implemented.

## **Interface Detailed Design**

For each system that provides information exchange with the system under development, there is a requirement for rules governing the interface. This section should provide enough detailed information about the interface requirements to correctly format, transmit, and/or receive data across the interface. Include the following information in the detailed design for each interface (as appropriate):

* The data format requirements; if there is a need to reformat data before they are transmitted or after incoming data is received, tools and/or methods for the reformat process should be defined
* Specifications for hand-shaking protocols between the two systems; include the content and format of the information to be included in the hand-shake messages, the timing for exchanging these messages, and the steps to be taken when errors are identified
* Format(s) for error reports exchanged between the systems; should address the disposition of error reports; for example, retained in a file, sent to a printer, flag/alarm sent to the operator, etc.
* Graphical representation of the connectivity between systems, showing the direction of data flow
* Query and response descriptions

If a formal Interface Control Document (ICD) exists for a given interface, the information can be copied, or the ICD can be referenced in this section.

# **SYSTEM INTEGRITY CONTROLS**

Sensitive systems use information for which the loss, misuse, modification of, or unauthorized access to that information could affect the conduct of State programs, or the privacy to which individuals are entitled.

Developers of sensitive State systems are required to develop specifications for the following minimum levels of control:

* Internal security to restrict access of critical data items to only those access types required by users
* Audit procedures to meet control, reporting, and retention period requirements for operational and management reports
* Application audit trails to dynamically audit retrieval access to designated critical data
* Standard Tables to be used or requested for validating data fields
* Verification processes for additions, deletions, or updates of critical data

Ability to identify all audit information by user identification, network terminal identification, date, time, and data accessed or changed.