Aye-Aye Sleep Monitoring

Nathan Hui
Engineers for Exploration
UC San Diego
nthui@ucsd.edu

September 14, 2023

1 Introduction

In 2020, the San Diego Zoo Wildlife Alliance[1] and Engineers for Exploration[2] set on a path to develop a system for monitoring animals in the care of the San Diego Zoo, both at the Zoo in Balboa Park, the Safari Park in Escondido, and the Biodiversity Reserve behind the Safari Park. The aim of this system was to provide 24/7 automated monitoring of animals with intelligent identification of abnormalities in behavior, which could indicate to animal care specialists the need to pay attention to certain things. This system would also provide a corpus of data for future machine learning/artificial intelligence tools.

The first phase of this system was aimed at monitoring the aye-aye lemur (*Daubentonia madagascariensis*) housed at the San Diego Zoo. This species of lemur is nocturnal, which presents challenges to animal care specialists in understanding the effect being on exhibit during the day impacts the aye-aye's ability to sleep. At a higher level, this also presents an opportunity to demonstrate a system capable of monitoring animals.

2 System Requirements

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMEND”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in RFC 2119.

1. The system MUST record video of the entire aye-aye enclosure.
2. The system MUST record video of the entire inside of the aye-aye nesting box.
3. The system SHOULD record audio from the inside of the aye-aye nesting box.
4. The system MAY record additional data from the aye-aye nesting box that indicate sleep behavior for the aye-aye.
5. The system MUST timestamp all data and store data in a central location.
6. The system MUST provide a programmatic interface to access the data.
7. The system MUST provide extensibility for additional sensor modalities, sensing, and automatic data processing.
8. The system SHOULD automatically recover from faults with minimal loss of data.
9. The system SHOULD provide for ensuring secure transfer of data.

[1] https://science.sandiegozoo.org/

[2] https://e4e.ucsd.edu

3 System Architecture

To solve this problem, we will use a central server as the end repository for all data, as well as for providing command and control. We abstract all sensors as sensor nodes, which will represent the different sensor modalities. For this phase of the design, we will connect the server and sensor nodes over Ethernet/WiFi. This is diagrammed in Figure 1.

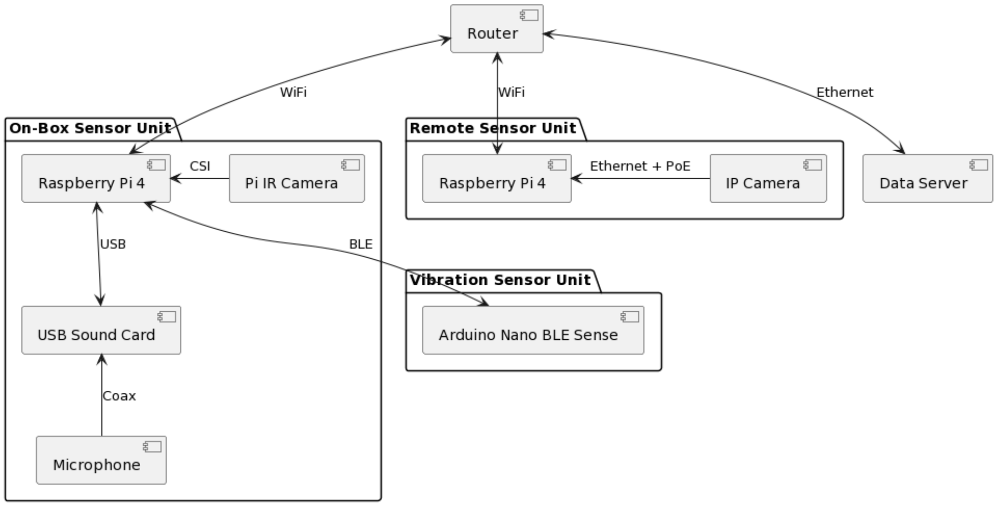

Figure 1: System Diagram

For this to work, each major component is specified below.

3.1 Data Server

1. The Data Server MUST aggregate all data from connected sensor nodes.
2. The Data Server MUST provide an interface to access the aggregated data.
3. The Data Server MUST add the following metadata to each data stream:
   (a) Originating sensor node
   (b) Timestamp
4. The Data Server MUST support nultiple sensor nodes that are dynamically added.
5. The Data Server SHOULD support both encrypted and unencrypted sensor data.
6. The Data Server SHOULD be capable of operating for 12 months without data offload or maintenance.
7. The Data Server SHOULD be capable of automatically resuming operations after power loss.
8. The Data Server MUST be able to monitor status of each Status Node.

3.2 On-Box Sensor Unit

1. The On-Box Sensor Unit MUST provide IR capable video data from the inside of the nesting box.
2. The On-Box Sensor Unit MUST provide IR illumination when necessary.

3. The On-Box Sensor Unit MUST provide audio data from the nesting box.
4. The On-Box Sensor Unit SHOULD be capable of automatically resuming operations after power loss.
5. The On-Box Sensor Unit MUST be powered from 60 Hz 120 VAC.
6. The On-Box Sensor Unit MUST utilize WiFi connectivity.

3.3 Remote Sensor Unit

1. The Remote Sensor Unit MUST provide IR capable video data from its field of view over the enclosure.
2. The Remote Sensor Unit SHOULD be capable of automatically resuming operations after power loss.
3. The Remote Sensor Unit MUST be powered from 60 Hz 120 VAC.
4. The Remote Sensor Unit MUST utilize WiFi connectivity.

3.4 Communications

1. Each Sensor Node MUST regularly indicate to the Data Server that it is alive.
2. Each Sensor Node type MUST conform to a preset set of data types/formats.
3. Sensor Nodes MUST regularly transmit data to the Data Server.

4 System and Component Design

4.1 Data Server

The code for the Data Server is located at `https://github.com/UCSD-E4E/ASM-data-server`. This is primarily based on an asyncio socket server, which creates individual handlers to manage each sensor node. These handlers manage the communications between the server and node, and also handle spinning up and down relevant services in order to handle incoming data.

4.2 Sensor Node

The code for all sensor nodes is located at `https://github.com/UCSD-E4E/ASM-rpi-node`. This is a common codebase for all sensor node types, since all of them do very similar things. The base class provides communications abstractions, and handles all of the asynchronous receive/send. Concrete subclasses handle the specific sensor bringup, configuration, and data management.