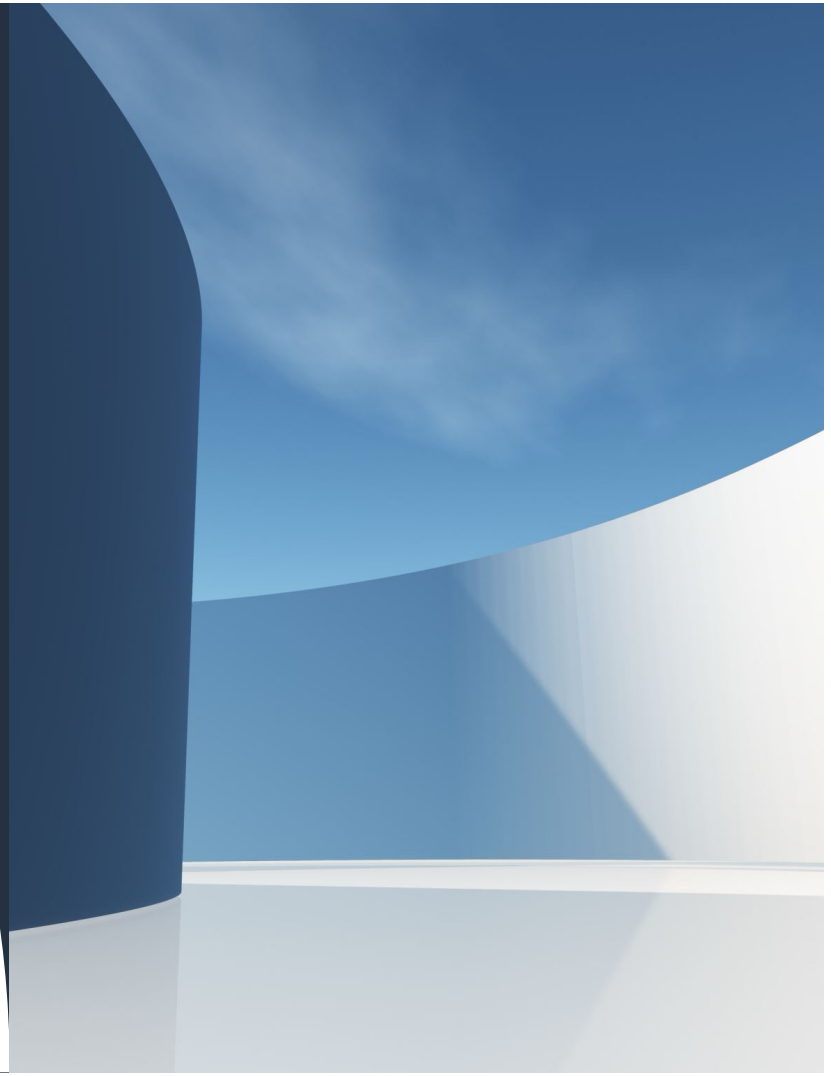
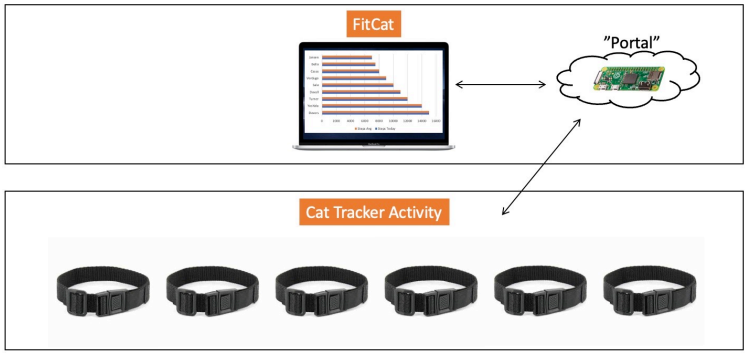
Quest 3: Fit Cat

By Joshua Arrevillaga, Michael Barany, Samuel Kraft



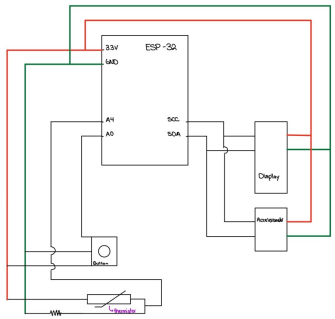
Project Description

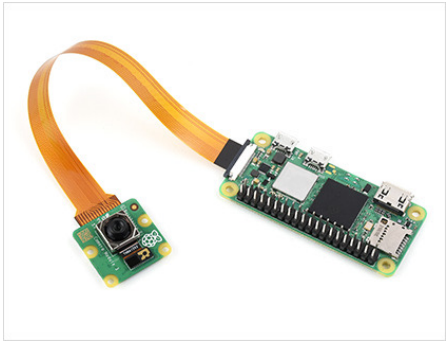
- **Wireless Cat Activity Trackers**: Develop multiple Cat Tracker devices using ESP32 modules connected via Wi-Fi to monitor cats' activities. These devices collect data on various activity states–resting, active, and highly active–and measure the cumulative time spent in each state.
- **Centralized Data Aggregation and Web Portal**: Implement a central Node.js server that aggregates data from all Cat Trackers. The server presents this information through a web portal accessible over the open Internet, displaying real-time leaderboards and charts for each cat's activity levels.
- **Bidirectional Communication and Leader Notifications**: Enable the central server to communicate back to each Cat Tracker. When there's a change in the leader (the cat with the most active time over the past 10 minutes), the server sends notifications to the devices, triggering updates on their alphanumeric displays and activating buzzers to alert the cats.
- **Integrated Webcam Streaming**: Incorporate a webcam streaming feature using a Pi Cam to provide live video feeds of the cats in the kennel. This stream is accessible from the client machine, allowing cat owners to watch their pets in real-time alongside their activity data.



Circuit

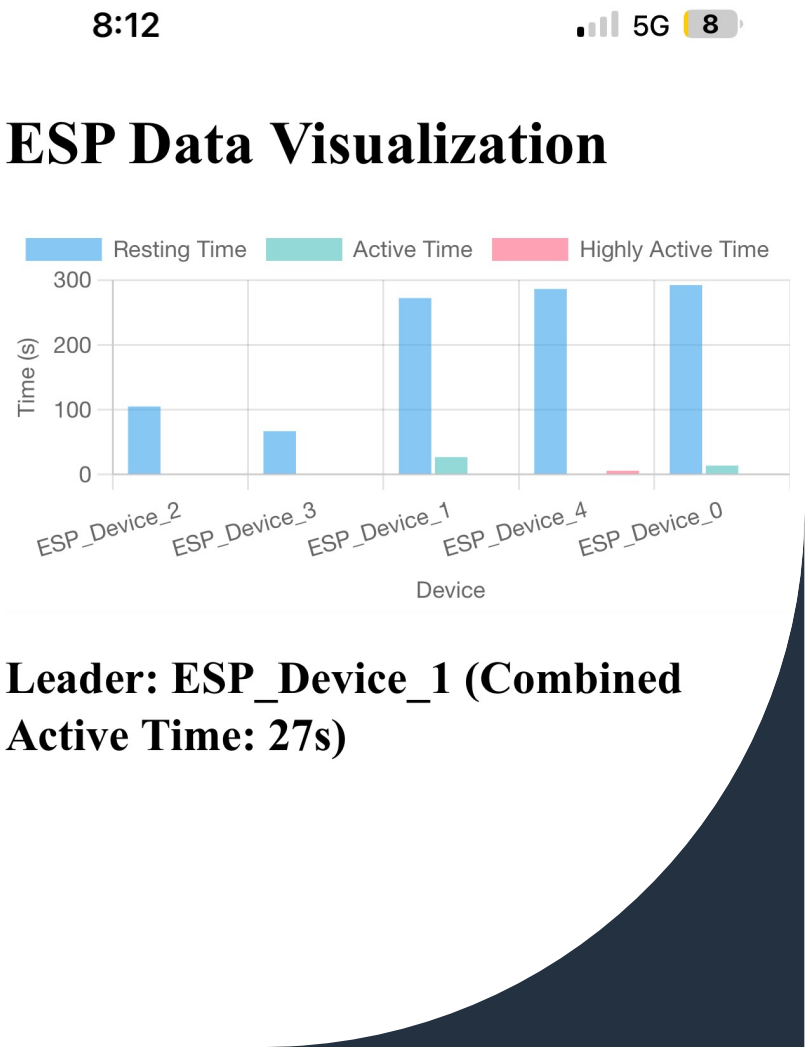
We utilized the same circuit from quest 2 and a raspberry pi





UI (index.html)

- Real-Time Data Visualization: The HTML page uses Chart.js to create a dynamic bar chart that displays real-time data from ESP devices, specifically their Resting Time, Active Time, and Highly Active Time. It utilizes Socket.IO to listen for data updates from the server and refreshes the chart accordingly to reflect the latest information.
- Activity Leaderboard Display: The page tracks and displays the leading devices in each activity category by updating the "Resting Leader," "Active Leader," and "Highly Active Leader" sections. This leaderboard shows which device has accumulated the most time in each state, providing a quick overview of the top performers.



Code Logic (CatTrack.c)

- Sensor Data Collection and Activity Recognition: The code reads accelerometer and temperature data to determine the cat's movement and orientation, classifying its activity into states like resting, moving, or upside down.
- Wireless Communication via Wi-Fi and UDP: It connects to a Wi-Fi network and sends the collected sensor data and activity states to a remote server using UDP. It also listens for incoming UDP commands to adjust device settings like the LED blink rate.
- User Interface and Control: It manages an alphanumeric display to show messages (like the cat's name and activity) when a button is pressed and controls an LED whose blink rate can be remotely updated, providing both local and remote interactivity.

Code Logic (server.js & script.js)

server.js

1. UDP Server for Data Aggregation and Device Control: The server.js script creates a UDP server that listens for incoming JSON-formatted data from multiple ESP32 devices. It aggregates this data—such as resting time, active time, and highly active time—for each device and stores it in the espData object. The server also identifies leaders in each activity category by comparing the activity times across all devices.
2. WebSocket Communication and Command Broadcasting: The server uses Socket.IO to emit real-time updates to connected web clients, providing them with the latest aggregated data and leader information for visualization. Additionally, when a leader change is detected, the server sends UDP commands to all ESP32 devices to adjust their LED blink rates, serving as a notification mechanism.

script.js

1.Real-Time Data Visualization with Chart.js: The script.js file establishes a WebSocket connection to receive live updates from the server. It utilizes Chart.js to create a dynamic bar chart that visualizes the resting time, active time, and highly active time for each ESP32 device. The chart updates in real-time as new data arrives, providing an immediate visual representation of device activities.

2.Leaderboard Calculation and Display: The script processes the incoming data to determine which device has the highest combined active and highly active time. It then updates the leaderboard display on the webpage to reflect the current leader, offering users an up-to-date view of the top-performing device in terms of activity levels.

Reflections

- Plan our timeline better with respects to other class workloads
- Use our resources better, asking professor and TAs for help when needed and properly go over documentation
- Finish skills on time
- Work more efficient with our teammates