

# Project Proposal

## CSC385H1 – Microprocessor Systems

The Listeners

March 28, 2024

<i>Group members</i>	Bob Shoaun Ng, Dian Rong, Max Wang, Rohan Tinna
<i>Project Name</i>	(JuLIAH) Just leave it at home
<i>Description</i>	A device that allows you to record loud noises caused by your pet, and also interact with your pet through an LED, while you are away from home.

## Technical description

### Tasks

The discovery board's user button toggles whether the board is monitoring or not. The user button is configured using interrupts that will update the program's state. On startup, the board is not monitoring. The current microphone state is indicated by an second LED.

While the board is monitoring, it ...

1. Repeatedly records noise data from the board's microphone(s) into a buffer that holds 5 seconds of data.
  - Use STM's FP-AI-SENSING1 function pack to continuously capture 5-second audio recordings. The recording function will convert the continuous audio signal into values (approximately 20,000 samples per second). These values can be stored as raw noise data in an array that can then be analyzed and processed.
    - Note: We may need external storage or additional RAM to have a large enough buffer to store our recorded audio.
  - Use the SimpleWaveRecorderPlayer repository to simultaneously capture audio data, which gets written to a wav file that can be playable.
2. Sends the most recently recorded audio as a wav file if the most recently recorded noise in the buffer contains any samples louder than 50 dB.

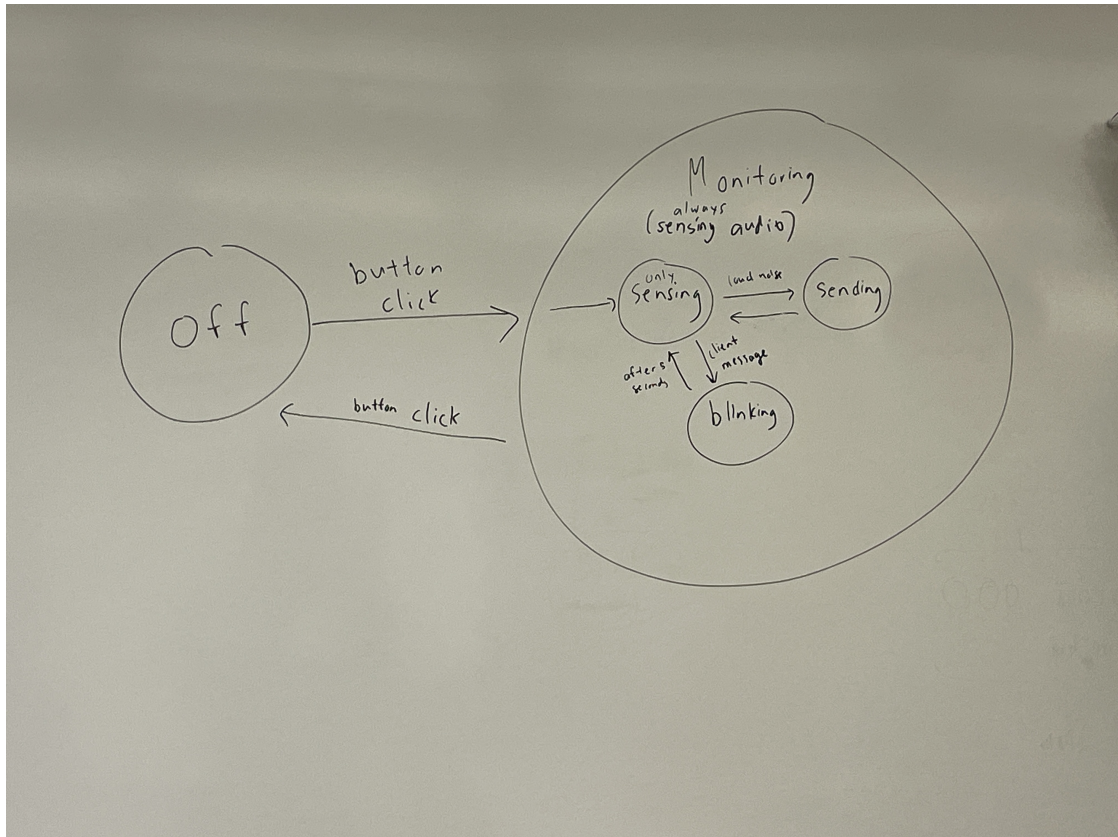


Figure 1: State Transition Diagram

- Check the volume of values in the buffer using `BSP_AUDIO_IN_GetVolume`
3. Transmits a recorded audio file over wifi.
    - API: `WifiInterface`, `NetworkInterface`, `TCPSocket`
    - Use `WifiInterface` to connect to a Wi-fi network.
    - Use `NetworkInterface` and `TCPSocket` to make TCP connections and make HTTP requests.
    - Use MQTT protocol because going to be sending lots of small data.
  4. Sending recorded noise to a MQTT Broker
    - Mosquitto can be used as the MQTT broker, to relay recordings from the publisher (discovery board) to subscribers (phone, laptop)
    - (Optional) A web server can be created to subscribe to the MQTT broker and receive all recordings, in addition it will also store the recordings in a database. This allows a history of recordings to be stored.
  5. Listen to recorded audio on your device through the website. Interact with device through the website.
    - A simple website can act as the MQTT client to receive recordings from the broker, and also publish blink LED commands to the broker. Written using HTML + CSS + Javascript and the `MQTT.js` package.
    - The website's user interface serves several functions:
      - Playback recordings (audio controls)
      - Send push notifications to device when new recording is received
      - Expose a "Blink LED" button, that when pressed, activates the LED on the discovery board (via MQTT broker).

## User story

User is leaving a pet at home:

1. User presses the button to start monitoring.
2. User leaves the device in their home, connected to a power source
3. User receives notification on their phone, stating there is a loud noise heard at home.
4. User accesses the website to listen to the audio recording on the loud noise. Realises the pet is restless.
5. User presses the "blink LED" button, which blinks the LED on the device at home to distract the pet.
6. User comes back home, and presses the button again to disable the microphone.

## Scope and challenge

The main challenges in this project are communicating with the board using wifi, recording audio from using the microphone, and setting up the MQTT broker.

### 0.1 Wi-fi Communication

We will use WiFiInterface to connect to wifi, and TCPSocket for communication between the board and client.

The biggest difficulty will be sending the (relatively large) .wav file from the board to the client, and figuring out how to trigger an LED from a client message.

Also we will need to figure out how to configure MQTT protocol for messages to be sent out via Wi-fi to the broker.

### 0.2 Recording Audio

We will use the FP-AI-SENSING1 function pack to work with audio from microphone, in a similar format to <https://github.com/janjongboom/b-1475e-iot01a-audio-mbed/blob/master/source/main.cpp>.

The difficulty lies in only saving data when the noise recorded is loud, saving an appropriate amount (i.e. large enough to have meaningful data, small enough for our board to handle), and saving the data in a transferable manner.

In order to convert recorded data into a playable wav file, we will use the SimpleWaveRecorderPlayer repository. The code's rec() function will allow us to create a wav file with a desired length.

### 0.3 Setting up MQTT broker

- Use Mosquitto as a MQTT broker between the board and web client.
- We will need to familiarize ourselves with the functions of Mosquitto, how to host it online, what is the maximum queue size.
- Check if its possible to intercept messages of a certain topic and store it in a database, if possible it removes the need of a separate web server to store the recordings. Redis can be used as a simple database to store recordings.
- Use MQTT.js as a MQTT client written in Javascript.

## Plan

### Week 1 (March 7)

- Rohan: Button to toggle between states of monitoring and not monitoring
  - Set up LED to indicate whether microphone is monitoring or not
  - Set up another LED blinking for 5 seconds
- Max: Set-up microphone and make sure audio API works
- Dian: Allow Discovery Board to connect to a Wi-fi network
- Bob: Setup Mosquitto broker (and topics)

### Week 2 (March 14 - Milestone 1)

- Max: Record and store 5 second audio clips
- Rohan: Query volume from Microphone API
- Dian: Setup TCP socket connection to MQTT broker
- Bob: Set up MQTT Client and frontend (Website)

### Week 3 (March 21 - Milestone 2)

- Rohan: Filter recordings so only those with a loud noise are kept
- Max: Convert recording to a .wav file
- Dian: Send audio recordings to Mosquitto
- Bob: Website frontend: playback recordings, button to blink LED

### Week 4 (March 28 - Milestone 3)

- Dian: Send blink LED command from MQTT client to MQTT broker
- Max: Receive and display audio recordings on MQTT clients
- Rohan: Receive data from web server to blink LED
- Bob: Hosting MQTT broker and website (MQTT client) online

### Presentation (April 1)