Fall 2023 CS120 Project 0. Warm Up

Due Date: Oct. 6, 2023

(3 points)

Please read the following instructions carefully:

- This project is to be completed by each group **individually**.
- Submit your code through Blackboard. The submission is performed by one of the group members.
- Each group needs to submit the code **once and only once**.
- This project will be checked with Project1.

Overview

This project serves as a prerequisite self-assessment. It is due before the course add/drop deadline.

Part 0. (0 point) Self- assessment

The following questions are summarized from students who have taken the course.

- Do you prefer a hands-on coding experience?

This course is project-oriented. The project accounts for a considerable amount of the total credit. You will have to write a lot of code throughout the semester. If your goal is to understand computer network protocols conceptually without getting your hands dirty, then please consult the instructor for further advice.

- Do you feel comfortable with learning on your own?

Project specification does not mandate the use of programming languages or libraries. This means that you are on your own to figure out which language or library to use. The course teaching team does not help with implementation issues.

- Do you feel comfortable with multithread programming?

You will need to send and receive packets at the same time, and thus at least 2 threads are needed. Multithreaded code can be difficult to debug due to distributed execution.

- Do you feel comfortable with hardware interfaces?

Unlike other CS courses where testcases are nicely-formatted and well-defined, this course does not have any testcases. Instead, you will have to interact with the real world in this project, which is cool and exciting. However, you might have to tune your code to fit hardware devices that usually do not guarantee a “perfect” outcome. You will have to face various hardware issues.

Part 1. (3 points) Understanding Your Tools

Different operating systems have different architectures for media I/O, but their usages are quite similar. In order to convey bits through acoustic signals, you have to correctly send and receive acoustic signals through the media interface.

Playing sound with a computer is about transmitting sound samples through the speaker. Recording is about sampling the sound signals through the microphone. You can start by referring to the Audio Stream Input/Output (ASIO) protocol [6]. ASIO is supported by most sound card drivers. Compared with the default/or common system audio interfaces, such as WindowsAudio or DirectSound, ASIO has a significant improvement in latency. Latency measures the elapse of time when a sample is "written" into the buffer of the sound interface and the time when the audio of that sample is actually played by the audio hardware. Latency matters a lot in our next project.

Latency is related to the buffer size of the sound interface. Choosing a smaller buffer size reduces latency, but might result in discontinuous sound in a loaded system. ASIO reduces the latency at its best while maintaining sound quality. ASIO has several driver implementations. ASIO4ALL [10] working in Windows is verified by the teaching team. For programming in JAVA, you can choose a JAVA wrapper [11] to access the ASIO driver. For programming in other languages, you can either use the ASIO native APIs directly or find other wrappers on your own. For Linux and MAC OS users, you can either try WineASIO or other low-latency audio interfaces, but there is no guarantee on that (the teaching team has not tested). Here is a utility to test the latency of the chosen sound interface [12]. The latency should be within 20 ms or better (15 ms) in order to finish Project 2.

In this part, your task is to use the chosen audio interface to correctly control your sound card to play digital samples through the speaker and record analog sound signals from the microphone. Playing and recording should be able to work simultaneously. You can use two separate threads for playing and recording.

Checkpoints:
The group provides one device: NODE1.
CK1(1.5 points). NODE1 is able to record the voice from TA (10 seconds) and then replay the recorded signals.
CK2(1.5 points). NODE1 is able to play a predefined sound (any) and record the playing sound at the same time. TA may say something during the recording. After 10 seconds, stop playing and recording. Then play the recorded sound for verification.

Tips:
a. Be careful about stereo settings. We only need one track
b. Be careful about the sampling rate of playback and recording, match them unless you know what you are doing.

Reference and Useful Links

[1] Ringing Effect https://en.wikipedia.org/wiki/Ringing_(signal)
[2] Equalization https://en.wikipedia.org/wiki/Equalization_(communications)
[3] OFDM https://en.wikipedia.org/wiki/Orthogonal_frequency-division_multiplexing
[4] Acoustic NFC http://dl.acm.org/citation.cfm?id=2534169.2486037
[5] FEC https://en.wikipedia.org/wiki/Forward_error_correction
[6] ASIO Audio Interface https://manual.audacityteam.org/man/asio_audio_interface.html
[7] ALSA http://www.alsa-project.org/main/index.php/Main_Page
[8] Digital Modulation https://web.stanford.edu/class/ee102b/contents/DigitalModulation.pdf
[9] Digital Modulation https://en.wikipedia.org/wiki/Modulation#Digital_modulation_methods
[10] ASIO4ALL www.asio4all.org
[11] JASIOhost https://github.com/mhroth/jasiohost
[12] RTL UTILITY https://oblique-audio.com/rtl-utility.php