PY 371 Electronic Lab Final Project Proposal

Boston University - Jingjie (Jack) Zheng

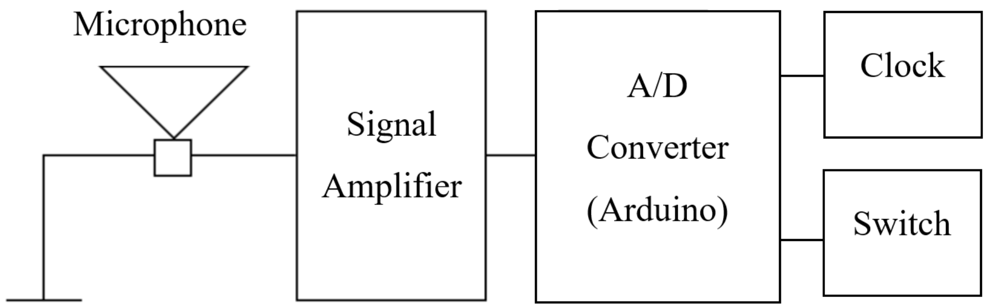
Abstract

I propose to apply concepts and methods in analog and digital circuits to assemble a system for recording and playing back sounds with a bandwidth of 4-kHz. The system will have two modes, Recording and Playing Back, and will be controlled though push buttons.

Overview of the Two Modes

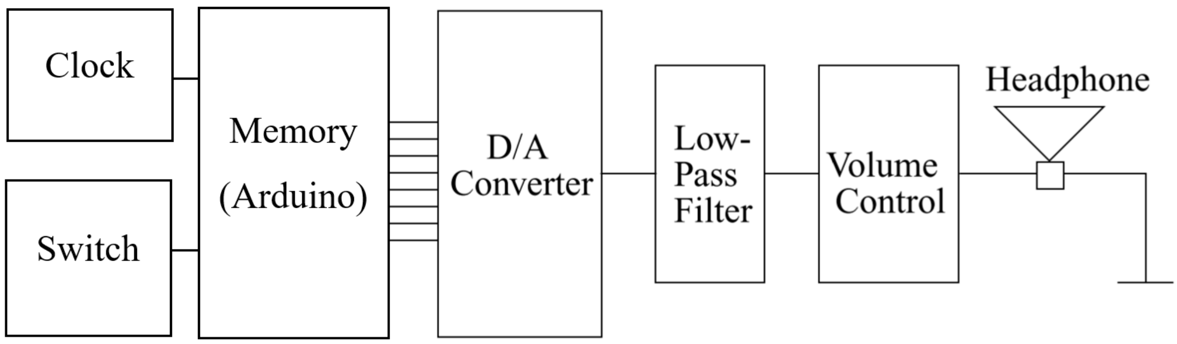
The 1st Mode -- Recording Mode will have three stages, as shown in the following graph.

1. The first stage is a microphone, which transduces mechanical sound signals into electrical analog signals.
2. The second stage is a signal amplifier, implemented using an operational amplifier.
3. The third stage is an 8-bit Analog-to-Digital Converter, which will sample the analog signal at 8-kHz rate and convert each sample into one of the 256 digital values between 0 and 5 Volts. This stage will be implemented using an Arduino Uno microcontroller, a system clock, and a push-button switch. The Arduino serves as the A/D Converter and will take three inputs: one from the signal amplifier, one from the system clock, and one from the switch. The switch tells the Arduino whether to turn this "Recording Mode" ON and start storing digital samples in sequence into the microcontroller's memory. The system clock is an oscillator that feeds a square wave of 8-kHz with peak-to-peak amplitude of 5 volts. This determines the sampling frequency of the A/D Converter.



The 2nd Mode – Playing Back Mode will have multiple stages, as shown in the following graph.

1. The first stage is reading and sending out 8-bit digital signals from the memory. These digital signals correspond to the signals previously sampled and stored in the memory of Arduino in the Recording Mode. The Arduino will take two inputs in this stage: one from the same system clock we used in the Recording Mode, and one from another push-button switch. The switch turns the Playing Back Mode ON/OFF and is not the same one as the switch in the Recording Mode. The Clock, which produces an 8-kHz square wave, provides the frequency based on which the Arduino will output the 8-bit digital signals.
2. The second stage is a Digital-to-Analog Converter, which converts the 8-bit digital signal to an analog signal.
3. The third stage is a low-pass filter, which has a cutoff frequency at 4-kHz. This stage is introduced to manage the aliasing problem in the signal processing. This also reflects that this whole system has a bandwidth of 4-kHz and cannot record and play back sounds have a frequency higher than 4-kHz.
4. The last stage is a volume control stage. As the name suggests, this stage controls the volume and sends either amplified or attenuated signals to a headphone, which plays out the sound.

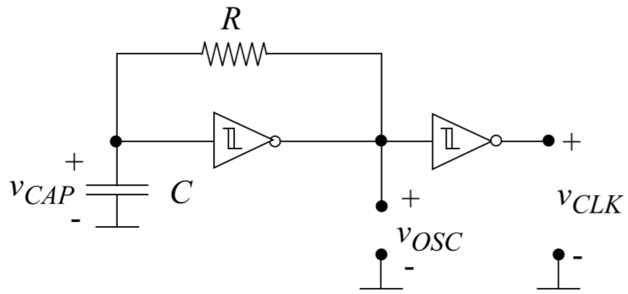


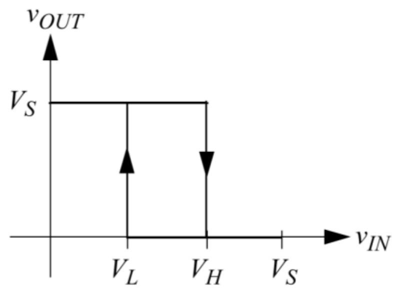
Note that some other features may be introduced to improve the usability of this system, such as using LEDs to indicate which mode is turned ON. These features are easy to implement, and they are not the central function of this system, so they will not be discussed here.

Details of Some Stages and Circuit Schematics

1. **The Clock**

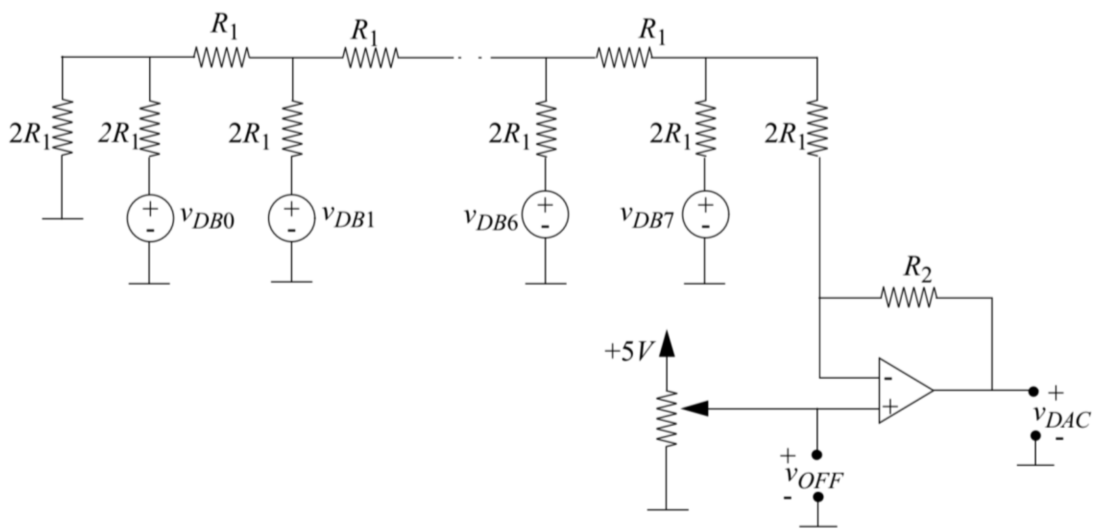
The circuit schematic is shown below. It is consisted of a square-wave oscillator followed by a CMOS inverter; the inverter functions only as a buffer. The oscillator is constructed from another CMOS inverter, a resistor and a capacitor. Both inverters are powered between the positive supply voltage V_S and ground, and both exhibit the hysteretic input-output characteristic defined in the figure. [1]





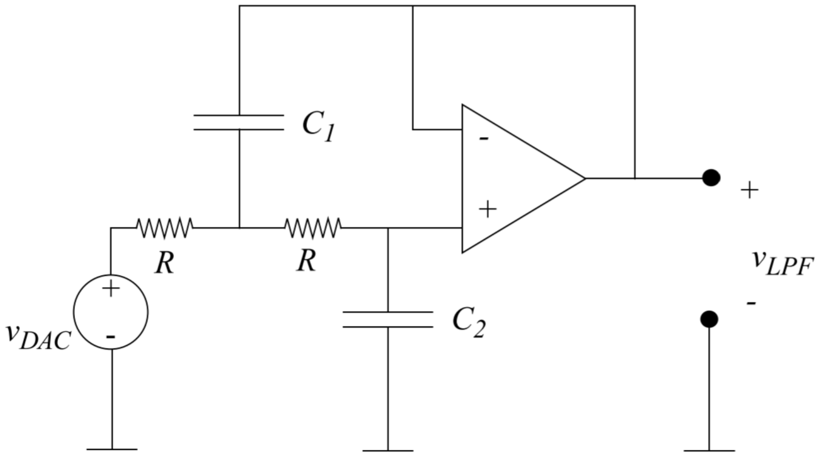
2. **The Digital-to-Analog Converter (DAC)**

This can be implemented in two ways. The first way is to use a SPI (Serial Peripheral Interface) DAC. The second way is to build our own, as shown in the circuit schematic below. The voltage sources V_{DB0} through V_{DB7} represent the voltages supplied by the eight data bits output by the Arduino. These voltages will be approximately 5 V when the corresponding data bit is a logical high, and approximately 0 V when the corresponding data bit is a logical low. The voltage V_{OFF}, which is set by a potentiometer, is an offset voltage that is used to bias the output of the converter. [1]



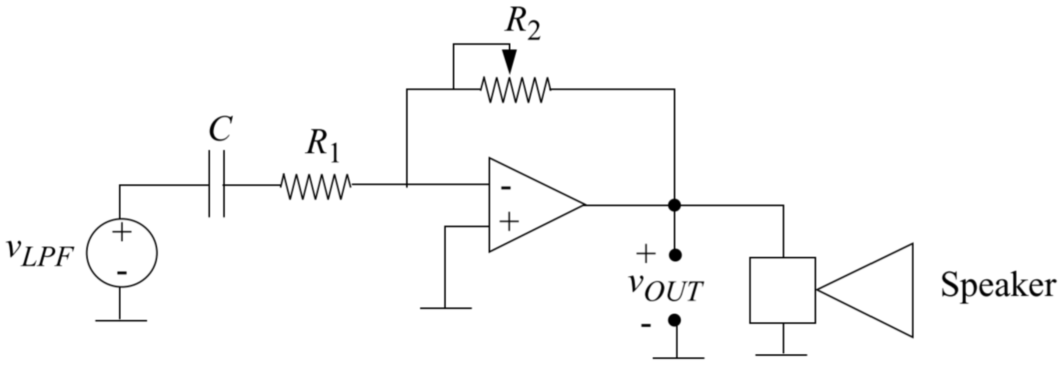
3. The Low-Pass Filter

The schematic is shown below. It is a second-order filter and is driven by the output of the digital-to-analog converter. [1]



4. The Volume Control

The schematic is shown below. The input is driven by the output of the low-pass filter. The output of this stage drives the headphone. A potentiometer is used for R_2 so that the gain of the circuit can be easily adjusted. (Because there exists a coupling capacitor at its input, the volume control stage behaves like a high-pass filter. In this way, the volume control stage is designed to prevent a possibly damaging DC voltage from being applied to the headphone. Such a voltage component could be present in V_{LPF} if, for example, V_{OFF} in the analog-to-digital converter is not properly adjusted to balance the output of the converter.)



Electronic Parts Needed

Part Category	Part Number	Quantity
SPI DAC	MCP4921	1
CMOS Inverters	74HC14	1
Op Amp	741 & 356	4
Speaker		1
Microphone		1
Arduino	UNO	1
Push-Button Switch		3
Red LED		1
Green LED		1
Resistors		n
Capacitors		n
Wires		n
Battery		1
Breadboard		1

Primary Reference Material

Courtesy of MIT Open Course Ware, Course 6.002, Circuits and Electronics

[1] https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-002-circuits-and-electronics-spring-2007/assignments/hw11.pdf