

EE 445L - Lab 3: Alarm Clock

Joshua Bryant
jmb6357

James Morris
jsm3288

October 13, 2014

1 Objective

1.1 Overview

1.1.1 Objectives

The objectives of this project are to design, build, and test a music player. Educationally, students are learning how to interface a DAC, how to design a speaker amplifier, how to store digital music in ROM, and how to perform DAC outputs in the background. Your goal is to play your favorite song.

1.1.2 Process

The project will be developed using the TM4C1294 board. There will be three switches that the operator will use to control the music player. The system will be built on a solderless breadboard and run on the usual USB power. The system will use off-board switches. A hardware/software interface will be designed that allows software to control the player. There will be at least three hardware/software modules: switch input, DAC output, and the music player. The process will be to design and test each module independently from the other modules. After each module is tested, the system will be built and tested.

1.1.3 Roles and Responsibilities

EE445L students are the engineers and the TA is the client. Students are expected to make minor modifications to this document in order to clarify exactly what they plan to build. Students are allowed to divide responsibilities of the project however they wish, but, at the time of demonstration, both students are expected to understand all aspects of the design.

1.1.4 Interactions with Existing Systems

The system will use the TM4C1294 board, a solderless breadboard, and the speaker as shown in figure 5.1 in the Lab05.doc. It will be powered using the USB cable.

1.1.5 Terminology

SSI A widely used serial interface standard for industrial applications between a master and a slave. SSI is based on RS-422 standards and has a high protocol efficiency.

Linearity A linear function is a function that satisfies two properties: additivity and homogeneity.

Frequency Response The frequency at which gain drops to 0.707 of the normal value. For a low pass system, the frequency response ranges from 0 to a maximum value. For a high pass system, the frequency response ranges from a minimum value to infinity. For a bandpass system, the frequency response ranges from a minimum to a maximum value.

Loudness The characteristic of a sound that is a subjective measure and is often confused with objective measures of sound strength such as sound pressure, sound pressure level, sound intensity or sound power.

Pitch A perceptual property that allows the ordering of sounds on a frequency-related scale.

Instrument An embedded system that collects information, same as data acquisition system.

Tempo Speed or pace of a given piece of music.

Envelope A function used for shaping the amplitude, frequency, or phase of a signal.

Melody A linear succession of musical tones that the listener perceives as a single entity.

Harmony The use of simultaneous pitches or chords.

1.1.6 Security

The system may include software from StellarisWare and from the book. NO software written for this project may be transmitted, viewed, or communicated with any other EE445L student past, present, or future (other than the lab partner of course). It is the responsibility of the team to keep its EE445L lab solutions secure.

1.2 Function Description

1.2.1 Functionality

If the operator presses the play/pause button the music will play or pause. If the operator presses the play/pause button once the music should pause. Hitting the play/pause again causes music to continue. The play/pause button does not

restart from the beginning, rather it continues from the position it was paused. If the rewind button is pressed, the music stops and the next play operation will start from the beginning. There is a mode switch that allows the operator to control some aspect of the player. Possibilities include instrument, envelope or tempo.

There must be a C data structure to hold the music. There must be a music driver that plays songs. The length of the song should be at least 30 seconds and comprise of at least 8 different sounds. Although you will be playing one song, the song data itself will be stored in a separate place and be easy to change. The player runs in the background using interrupts. The foreground (main) initializes the player, then executes for(;;) do nothing loop. If you wish to include OLED output, this output should occur in the foreground. The maximum time to execute one instance of the ISR will be measured in lab and will be included in this document once it has been tested. You will need public functions **Rewind**, **Play** and **Stop**, which perform operations like a cassette tape player. The **Play** function has an input parameter that defines the song to play. A background thread implemented with output compare will fetch data out of your music structure and send them to the DAC.

There must be a C data structure to store the sound waveform, or instrument. You are free to design your own format, as long as it uses a formal data structure (i.e., **struct**). The generated music must sound beautiful utilizing the SNR of the DAC. Although you only have to implement one instrument, it should be easy to change instruments.

1.2.2 Scope

Phase 1 is the preparation; phase 2 is the demonstration; and phase 3 is the lab report. Details can be found in the lab manual.

1.2.3 Prototypes

A prototype system running on the TM4C1294 board and solderless breadboard will be demonstrated. Progress will be judged by the preparation, demonstration, and lab report.

1.2.4 Performance

The system will be judged by three qualitative measures. First, the software modules must be easy to understand and well-organized. Second, the system must employ an abstract data structure to hold the sound and the music. There should be a clear and obvious translation from sheet music to the data structure. Backward jumps in the ISR are not allowed. Waiting for SSI output to complete is an acceptable backwards jump. Third, all software will be judged according to style guidelines. Software must follow the style described in Section 3.3 of the book. There are three quantitative measures. First, the SNR of the DAC output of a sine wave should be measured. Second, the maximum time to run

one instance of the ISR will be recorded. Third, you will measure power supply current to run the system. There is no particular need to optimize any of these quantitative measures in this system.

1.2.5 Usability

There will be three switch inputs. The DAC will be interfaced to a 32-ohm speaker.

1.2.6 Safety

If you are using headphones, please verify the sound is not too loud before placing the phones next to your ears.

1.3 Deliverables

1.3.1 Reports

A lab report described in Lab05.doc is due by the date listed in the syllabus. This report includes the final requirements document.

1.3.2 Audits

The preparation is due at the beginning of the lab period on the date listed in the syllabus.

1.3.3 Outcomes

There are three deliverables: preparation, demonstration, and report.

2 Hardware Design

