

Microprocessor-Based Systems (E155)

Lab 5: Digital Audio

Learning Objectives

By the end of this lab you will have...

- Built a circuit to enable an I/O pin from your MCU to drive a speaker
- Implemented the timer functionality available on the ATSAM4S4B by reading the datasheet and writing your own library in C from scratch

Requirements

Build a system to play music on a speaker. Use your μ Mudd MkV, a LM386 audio amplifier, and an 8-ohm speaker. The μ Mudd should read a list of notes specifying the pitch (in Hz) and duration (in ms) of each note. It should generate a corresponding sequence of square waves. A period of 0 indicates a rest (silence for the given duration). A duration of 0 indicates the end of the song. Test your system on the score of Für Elise, which is provided.

Discussion

You can find lab5_starter.c on the class web page with the Für Elise score and some provided code.

Create a new project in Keil and select ATSAM4S4B in the ‘Select Device for Target’ window. In the ‘Manage Run-Time Environment’ window, expand ‘**Device**’ and select the checkbox in the ‘**Startup**’ row. Then click ‘**Resolve**’, then ‘OK’. Set ‘Options for Target 1’ and add your code to ‘Source Group 1’ in the same manner as in Lab 4.

When deploying your code using Keil, you’ll need to fix one issue in the default startup code before you can deploy your code to the board. In the ‘Project’ pane on the left, expand ‘**Device**’ and open the ‘**startup_SAM4S.c**’ file. Scroll down to line 210 and change the ‘1’ to a ‘0’ as shown below. This turns off built-in code that sets up the wrong clock source. Instead, you’ll initialize the SAM with the code in lab5_starter.c.

```
209  L*/
210  #define PMC_SETUP          0           // Reset values:
211  #define PMC_SCER_Val      0x00000100  // WO register (0x00000001)
.....
```

Also note that after deploying your code to the ATSAM4S4B, you’ll have to hit the “SAM_RESET” button on your board before the code will run. (Or you can run it in the debugger.)

A goal of this lab is for you to learn to interpret a datasheet and figure out how the timer works. Write your own code from scratch to use the system timer. Do not use the SAM4S4B libraries provided for future labs.

The GPIO pins don't generate enough output current to play satisfactory music directly on the speaker, so use an LM386 audio amplifier between the μ Mudd MkV and the speaker. Do not connect the μ Mudd directly to the speaker, as the current draw could damage it. The datasheet shows AC coupling from the amplifier to the speaker, but you can leave out the capacitors and resistors and produce an acceptable square wave. Volume control is optional. There are only a limited number of speakers available in the lab so *please leave the speakers in the supply cabinet when you leave lab*. Do not leave them attached to your breadboard when you are done working. If you kill a speaker, throw it away rather than putting it back in the cabinet for your unfortunate classmates.

Extra Credit

Up to one point of extra credit can be earned if you compose and play a different tune. The following information may help as you compose your music.

The duration depends on an arbitrary choice of tempo (speed at which the piece is played). If a whole note is chosen to be $\frac{1}{2}$ second long, other notes follow accordingly:

Duration	Seconds
Sixteenth	0.03125
Eighth	0.0625
Quarter	0.125
Half	0.25
Whole	0.5

Recall that the A above middle C (called A4) is 440 Hz and that an octave spans a factor of 2 in frequency. There are twelve notes in an octave spaced evenly on a geometric scale, so each is separated in frequency by a factor of $2^{(1/12)}$.

Note	Frequency (Hz)
A3	220
A sharp / B flat	233.1
B3	246.9
C3 (middle C)	261.6
C sharp / D flat	277.2
D3	293.7
D sharp / E flat	311.1
E3	329.6
F3	349.2
F sharp / G flat	370.0
G3	392.0

G sharp / A flat	415.3
A4	440
A sharp / B flat	466.2
B4	493.9
C4	523.3
C sharp / D flat	554.4
D4	587.3
D sharp / E flat	622.2
E4	659.2
F4	698.4
F sharp / G flat	740.9
G4	784.0
G sharp / A flat	830.6
A5	880

Credits

This lab was original developed in 2015 by Alex Alves '16 and redesigned for the μ Mudd Mark 5.1 in 2019 by Caleb Norfleet '21.