Lab #7: PWM Audio Signal Generation

Introduction

There are many devices that can be controlled periodically by pulsing their control signals. For example, stepper motors rotate some fixed amount for each pulse applied to their inputs. The intensity of a light or the speed of a DC motor can be controlled with pulse-width modulated (PWM) digital waveforms, with the light/motor effectively controlled by the average voltage of the signal. A PWM waveform is a periodic signal comprising pulses of variable duration (width). For a constant period, variations in the pulse width can create different desired effects.

The purpose of this lab is to generate a PWM signal waveform, with keypad/D-pad selectable frequencies. These will be approximate audio signals, which will be output through an 8-Ω speaker for our auditory pleasure. In later labs, these audio signals will be used to provide feedback to the user when playing a small game.

PWM Signal Characteristics

Figure 1 illustrates three PWM signals, and the two key parameters of a PWM signal. Each waveform is characterized by two parameters:

$$T = T_1 + T_2$$

and

$$duty\ cycle = \frac{T_1}{T}$$

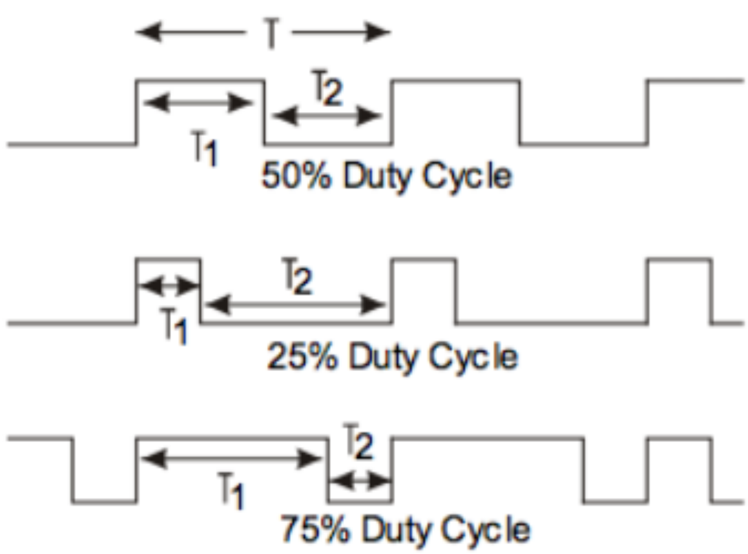

Figure 1: Pulse-Width Modulated (PWM) waveforms

Here, T is the *period* o the waveform, with the signal high for time T_1 and low for T_2. The *duty cycle* of the waveform is defined as the high time divided by the period: $T_1/(T_1 + T_2) = T_1.T$. The three waveforms in Figure 1 have duty cycles of 50%, 25%, and 75%, as indicated.

The term “pulse-width modulation” refers to the alteration (modulation) of the high time, T_1, while maintaining a constant period, T. However, in this lab the duty cycle will be kept at a constant 50% while varying the period T in order to create different frequencies. A constant duty cycle of 50% allows for a nice-sounding, clean audio signal, whereas a lower or higher duty cycle causes the sound to be sharp and have varying volume.

PWM Signal Generation

In this lab, PWM signals are to be generated by the programmable timer module utilized in the previous lab.

Overviews of the timer module are given in the Monday presentation slides for this lab and the previous lab, posted on the class web page, the STM32L1xx Reference Manual (chapter on general-purpose timers TM9/10/11), plus ELEC 2220 course slides on timers, also linked to the class web page.

Hardware and Software Design

You are to implement a variable frequency PWM waveform generator with keypad/D-pad selectable frequencies. Keypad keys 0:4 and all D-pad keys except for ‘Reset’ are to select one of 5 unique musical notes. The selected note should be displayed using the binary representation of the lower half of the corresponding ASCII character. Additionally, the ‘Reset’ or ‘F’ key should turn off the output signal. Table 1 shows the frequencies for common music notes, while Table 2 shows the ASCII character values for the common music notes and the desired display. (*HINT – Consider placing counter register values for ten non-zero duty cycles in an array, with the number corresponding to the pressed keypad button used as an index into that array.*)

Music Note	Frequency (Hz)
A_4	440.00
B_4	493.88
C_5	523.25
D_5	587.33
E_5	659.25
F_5	698.46
G_5	783.99

Table 1: Frequencies of Common Music Notes

Character	ASCII value	Desired Output
A	0x41	0001
B	0x42	0010
C	0x43	0011
D	0x44	0100
E	0x45	0101
F	0x46	0110
G	0x47	0111

Table 2: ASCII Character values

A C progam is to be written to produce the variable frequency PWM signal using programmanble timer module TIM10 and output pin PA6. For this lab, the initial PWM frequency period T is arbitrary, but the duty cycle must be initialized to 0% in order to have no sound playing upon start.

Pre-Lab Assignment

Prior to the laboratory, design the C program to meet the specifications described above in the section "Hardware and Software Design". In your laboratory notebook, record the following (prior to lab):

1. Flowcharts for setting up the programmable timer in such a way that a PWM signal is produced; i.e. setting up the appropriate flags and registers.
2. Draft a program (or directions to your content on the H: drive)
3. A plan for testing the PWM audio signal generator
4. The PSC and ARR values required for each unique musical note

Laboratory Experiments

1. Verify that your hardware is operating properly by using test programs from previous labs, or some other brief program that exercises the hardware (D-pad or Keypad)
2. Download and run your PWM waveform generation program, displaying the generated waveform on the oscilloscope. Demonstrate to the lab instructor that you can use the keypad or D-pad to select the frequency from 5 unique values and maintain a 50% duty cycle, as well as demonstrate the stop condition. Also show that the stop condition leaves the signal in the low state.
3. Once proper operation has been demonstrated, the instructor will give you an 8-Ω speaker to play your waveforms.
4. Record observations on the sound quality of each note. Change the duty cycles to 25% and 75%, then record more observations. Which one sounds the best?
5. Use watch variables and the logic analyzer to trace the flow of your program. For example, if the program does not appear to be responding to certain keypad entries,

capture I/O port information to determine the sequence of instructions executed, the values read from the keypad, etc.

Deliverables

Refer to the schedule on the course web page for this week's deliverables

Possible Information for Future Laboratory Reports

1. Briefly describe the circuit (but not "wire by wire") and the test programs (include a circuit diagram and C program source listings).
2. Discuss your results. For each program, include a table and graphical plot of measured duty cycles, and at least one image of the oscilloscope or logic analyzer screen that shows the generated PWM signal.
3. Discuss what you had to change to generate PWM signals with different frequencies. In the next lab, be prepared to use these notes to play a song.