EECE 344 Digital Systems Design

Lab 5: Basic Signal Generator

1 Pre-Lab

1. Review the lecture notes about the concepts of interrupt handling.
2. Read the textbook Sections 9.1 through 9.5 about the interrupt handling on Cortex-M4.
3. Review the lecture notes about methods of DAC.
4. Read the textbook Section 10.2 about DAC.

2 Objective

The objective of this lab is to practice interrupt handling and DAC:

1. Structure your program to utilize two processor modes: SVC and IRQ.
2. Initialize GPIO Port F to allow switches to generate falling-edge interrupts.
3. Construct an interfacing circuit to convert digital signals from the microprocessor to analog signals.
4. Implement the microprocessor and the interfacing circuit to be a basic signal generator: saw-tooth signal and sine wave signal.

3 Requirements

Required parts: the Tiva LaunchPad for TM4C123, several resistors, jumpers, ADALM 2000 (as the +5V power source and oscilloscope), op-amp, and a breadboard.

Work in pairs for this lab assignment.

4 System Requirements

Signal generator is a fundamental instrument in engineering. Typical commercial signal generator is able to give square waves, triangle waves, sinusoidal waves with different frequencies, and more. Keep in mind that these generated signals are in the analog domain. In this lab, you will design a basic signal generator using the microprocessor and several peripherals. The microprocessor processes digital data, so we will need to construct the digital to analog converter (DAC) as the interface to the analog world.

Your basic signal generator is supposed to work as follows.

1. The push buttons on Port F are supposed to be the input selections.
2. If SW1 (interfaced to PF4) is pressed, then the signal to be generated is a saw-tooth wave with a period of 256 ms. The saw-tooth wave is shown in Figure 1. The numbers on the vertical axis, such as 0.5 and 1, indicate the half-scale and full-scale output. They do not mean specific voltage values. The horizontal line numbers represent the number of periods.
3. If SW2 (interfaced to PF0) is pressed, then the signal to be generated is a sine wave with the period of 60 ms. The shape of a sine wave is shown in Figure 2 below. We will shift the sine wave to be in 0 to the full-scale voltage.

4. If both switches are pressed, then the signal generator outputs nothing.

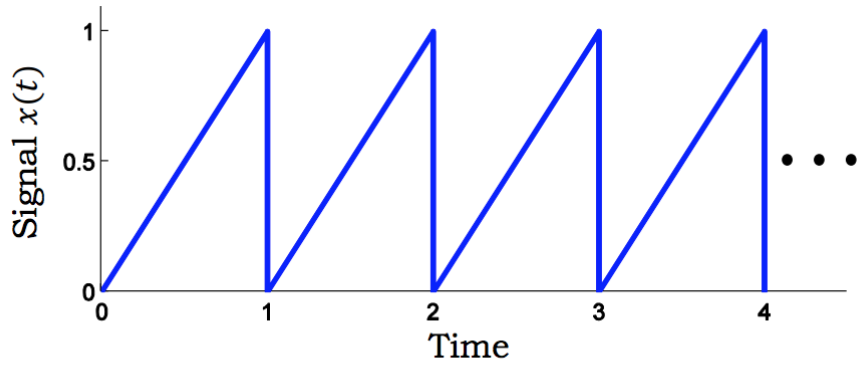

Figure 1: **Sawtooth wave**

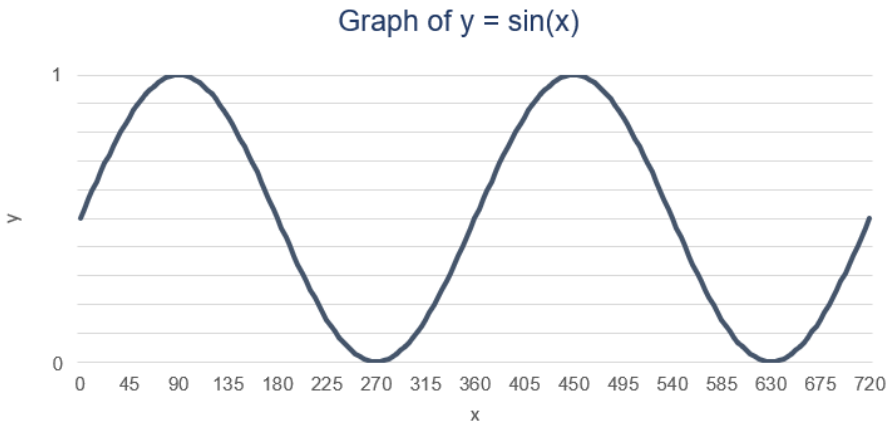

Figure 2: A sine wave shift to [0, 1].

4.1 Design and Implement the Embedded Software

We will write two functions for our system. One for the SVC mode and the other for the IRQ mode. The two functions are main() for SVC and GPIOF_Handler() for the IRQ. Carefully read the following system software requirements and draw two flowcharts, one for main() and the other for GPIOF_Handler(). In your deliverables, draw your flowcharts using a software editor.

1. Set the frequency to be 50 MHz by PLL.
2. Create the unit delay (such as SysTick_Wait1ms) by using the SysTick.
3. Configure PF4 and PF0 as inputs. Further configure PF to be able to generate interrupts. Configure NVIC to allow PF interrupts. Program PF to be at priority level 5. Enable the global interrupt.
4. Define PF4 and PF0 to give falling-edge interrupts.
5. Configure PB as DAC outputs.
6. Declare a global variable. Initialize it to 0.
7. In the main function, depending on the global variable's value, output different signals. If the variable is 0, output nothing. If the global variable is 1, output a saw tooth waveform with the period to be 256 ms. If the global variable is 2, output a sine waveform with the period to be 60 ms.

8. The GPIOF_Handler changes the value of the global variable. If SW1 is pressed, then change the global variable to be 1. If SW2 is pressed, change the global variable to be 2. If both switches are pressed, then the global variable is changed to 0.

4.2 Build the Interfacing DAC Circuit

Recall that there are two popular methods to construct DAC: the binary weighted DAC and the R/2R ladder. Popular DAC ICs use the R/2R method. Draw the circuit of your DAC. In the deliverables, you will need to use a software to draw your interfacing circuit.

1. Make PB7 the MSB of your DAC.
2. Make PB0 the LSB of your DAC.
3. Construct any bits in between PB7 and PB0 according to the DAC circuit explained in the lecture.
4. Use an op-amp, such as 741, to obtain the final analog output of your DAC.
5. Observe the output signal of your DAC using an oscilloscope.

5 Grading

The grading of your lab comprises the following components. Refer to the rubrics about the ranking of each component.

5.1 Performance of the Lab

Show the performance of your built system. Explain the interfacing circuit to your instructor.

5.2 Demonstration

When demonstrating the program, you are expected to explain each line of code if asked. Each student will be asked a different question to demonstrate his/her understanding of the project at hand.

5.3 Deliverables

Please indicate who your lab partner is in your deliverables. Submit your deliverables on Canvas to “Lab 5: Basic Signal Generator” under “Assignments” by the due time and date. Refer to the schedule shown on Canvas. For this lab, your deliverables are:

1. Provide the two flowcharts you designed for main() and GPIOF_Handler(). Use software to create the flowchart, as any hand-drawn flowchart will be classified as at best second ranking or lower according to the rubrics.
2. Your interfacing circuit drawn by a professional software, such as PCB Artist (available at https://www.4pcb.com/free-pcb-design-software.html). Any hand-drawn chart will be classified as at best second ranking or lower according to the rubrics.

5.4 Code

Submit your code on Canvas to “Lab 5: Basic Signal Generator” under “Assignments” by the due time and date. Your project folder should contain all C code, compressed into a zip file. Include comments and indentation in your code. Points will be deducted for sloppy code lacking documentation.

5.5 Lab Report

You (as an individual student, not a group) may choose to write a lab report for this lab if you have not fulfilled the requirement of writing three lab reports. If you choose to write a report for Lab 5, submit your report on Canvas “Assignments” by the due time and date. Follow the instructions in the section on "Lab Report Writing Guidelines" in “EECE 344 Digital Systems Design Lab Policy.” In the body of your report, make sure you include the following:

1. The two flowcharts of the program you designed. You may use the flowcharts you prepared for Deliverables (5.3 above). Use software to create the flowcharts, as any hand-drawn graph will be classified as at best second ranking or lower according to the rubrics.
2. Your interfacing circuit drawn by professional software. You may use the circuit you prepared for Deliverables (5.3 above). Any hand-drawn circuit will be classified as at best second ranking or lower according to the rubrics.
3. A photo of the circuit on the breadboard for the system.

Provide technical writing explaining your design and the graph of your code. Describe any challenges in your experiment and how you solved the problems. Draw a conclusion about designing and implementing FSM on the LaunchPad.