CSE466 LAB 2 – ASSEMBLY AND PULSE-WIDTH MODULATION

AUTUMN 2011

OBJECTIVES

In this lab you will DO the following:

- Recreate Part 4 of Lab 1 using assembly instead of C.
- Control the brightness of an LED using a hardware PWM (Pulse-width modulation) module.

In this lab you will LEARN the following:

- The native (assembly) instruction set of the MSP430.
- How to configure a timer peripheral for PWM output.
- How to operate an oscilloscope.

DELIVERABLES

At the beginning of the next lab period you will demo your working system to the TA.

Prior to the beginning of your next lab period, you will turn in the following to the course dropbox:

- A PDF containing answers to all questions posed in this lab prompt
- Your fully commented and neatly presented code, in a zipped format

RESOURCES

These documents and web resources will be useful in completion of the lab and/or in answering the questions posed.

- MSP430F2013 Datasheet
- MSP430x2xx Family User's Guide
- MSP430 Code Examples
- MSP430 Software Coding Techniques
- eZ430-F2013 User's Guide
- Code Composer Wiki
- Code Composer v4.2 User's Guide for MSP430
- MSP430 Optimizing C/C++ Compiler User's Guide

PART 1: INTERRUPTS IN ASSEMBLY

GOAL:

- Implement Part 4 from Lab 1 using assembly code.

HINTS:

- Reference the following MSP430 assembly code examples from TI :
  - msp430x20x3_ta_02.asm
  - msp430x20x3_sd16A_01.asm
- Be sure to select "Treat as an assembly-only project" when creating the project in CCS.

Question 1.1: How many lines of Assembly did this program require? How many lines of C?

Question 1.2: What was the resulting binary size for assembly and C versions? How did you determine this?

PART 2: OSCILLOSCOPE TUTORIAL

We will now take a brief interlude to introduce (or hopefully re-introduce) you to the oscilloscope. This is an important tool for the embedded systems engineer, and will be useful for Part 3 of this lab.

For this section, please go through the Oscilloscope Tutorial on the course website and answer the questions below. Reference the oscilloscope manual if needed.

Question 2.1: Which parameters of circuit operation (current, voltage, resistance, frequency, etc) is your oscilloscope capable of directly measuring or calculating?

Question 2.2: TRUE or FALSE: It's OK to connect two oscilloscope probe ground clips to different nodes (justify your answer).

PART 3: PWM GENERATION WITH TIMER_A

A pulse-width modulated (PWM) waveform is a rectangular wave with a variable ON time. The ON time of the waveform relative to its period is known as its duty cycle. PWM signals are used when there is a need to vary the effective power of a signal, such as in a motor or lighting controller.

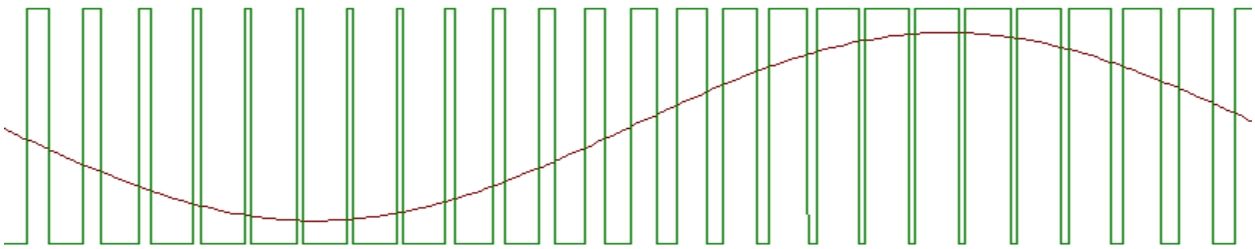


Figure 1. The green waveform is a PWM signal with varying duty cycle. Red represents the average signal level.

The TIMER_A peripheral of the MSP430 may be configured to produce a PWM waveform directly on an output pin. Like many hardware peripherals of the MSP430, the timer may be configured in a "set-and-forget" manner if desired, and will run continuously until reconfigured. The file "msp430x20x3_ta_16.c" configures the timer to produce a PWM signal, routes this PWM signal to a physical output port, and then switches off the CPU entirely. The PWM operation will run independent of the CPU and other subsystems of the microcontroller.

HINTS:

- Review the TIMER_A section in the MSP430F2013 datasheet.
- Only certain pins may be configured to produce the PWM signal.

PROCEDURE:

1. Create a new project and import “msp430x20x3_ta_16.c”.
2. Connect the LED to the PWM output port of the MSP.
3. Connect an oscilloscope probe between the PWM output port and ground.
4. Run the application and observe the LED and oscilloscope waveform.
5. Change the duty cycle of the PWM waveform and reprogram the target. Observe changes.
6. Change the period of the PWM waveform and reprogram the target. Observe changes.
7. Produce an automatic LED fade effect by adjusting the PWM duty cycle in an ISR. The LED should smoothly fade between low- and high-brightness states.

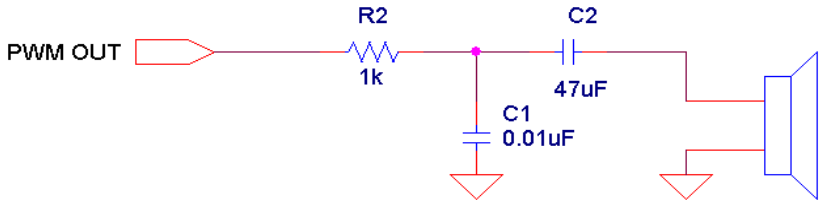
Question 3.1: What is the advantage of direct hardware control of the PWM output, rather than toggling the pin in an ISR?

Question 3.2: How did changing the PWM period affect operation of the device?

PART 4: OPTICAL THEREMIN

GOALS:

- Use the PWM module as a musical tone generator. Connect the PWM output to a piezoelectric speaker element using the filter shown.
- Select which musical tone (from the table below) to play back based on light intensity. You should be able to “play” the optical theremin by waving your hand over the light sensor.



HINTS:

- The PWM signal produced for each tone should have a 50% duty cycle to ensure maximum loudness and consistent sound quality. Make sure you are achieving a 50% duty cycle on every tone.

Table 1. Musical note frequency map. Bold notes make up the C Major scale.

Musical Note	**C**	C#/Db	**D**	D#/Eb	**E**	**F**	F#/Gb	**G**	G#/Ab	**A**	A#/Bb	**B**
Frequency (Hz)	**262**	277	**294**	311	**330**	**349**	370	**392**	415	**440**	466	**494**