CSE P567 - Winter 2010

Name_________________

Lab 7 – Timers and Interrupts

You should complete the first 4 parts in Lab (by 9:30PM), and the rest as a homework assignment, which you will demo in the next Lab. There will be a small (10%) penalty for late turnin.

Overview

In this lab you will use the timers on the Atmel processor to generate the PWM signals more accurately than when you used a simple program driven model.

Part 1 – Simple Timer-Driven Interrupts

Set up a timer that generates an interrupt exactly 8 times per second. Write an interrupt routine, triggered by the timer interrupt that flips the light each time it is called. This will produce a light that blinks 4 times/sec. (Use timer 2 (or 1) so that you don't get in the way of the library code, which uses timer 0.)

Simple Timer-Driven Interrupt Demo ______________________________TA

Part 2 – Fast Timer-Driven Interrupts (2-person teams)

Change your first program so that the timer now generates interrupts 100,000 times per second. You won't be able to see the light switching of course, but using the oscilloscope, measure the signal frequency and pulse width to determine how accurately the microcontroller can perform this task. If you have not used an oscilloscope before, wait for us to show you how to use one.

Fast Timer-Driven Interrupt Demo_________________________TA

Part 3 – Accelerometer Interface (2-person teams)

Study the datasheet for the 2-axis accelerometer (see the Lab page). This accelerometer outputs the acceleration using a PWM signal, which allows a single wire interface. This accelerometer board has a 5-pin connector which you should use to connect it to your Arduino board. Here are the connections (the colors may not match – use the letters, which appear on the board pins, as the guide. For now, you only need to connect the power and GND to the Arduino board. How you connect the X and Y acceleration pins will depend on your program.

A (Red): +5 V
B: Self test (not used)
C (Yellow): Y acceleration
D (White): X acceleration
E (Black): Ground

Before proceeding, have your TA check your connections.

Accelerometer Interface Checkoff__________________________TA

Part 4 – Accelerometer Experiments (2-person teams)

Use the oscilloscope to look at the two X and Y PWM signals produced by the accelerometer while you tilt the accelerometer in all directions. Do these signals match what you expect to see from the data sheet?

What is the frequency of the signals?

What is the range in the duty cycle?

Accelerometer Ouput Scope Demo__________________________TA

Part 5 – Accelerometer Polling Interface

Write a program that uses a "polling style" interface to determine the PWM values of the 2-axis accelerometer and display these values in the Serial Monitor. This program should continuously monitor the inputs of the accelerometer, looking for edges. It can then use the micros() function to determine the duty cycle of the signal.

Accelerometer Polling Interface Demo__________________________TA

Part 6 – Accelerometer Interrupt Interface

Now write a program that uses external interrupts to do the same thing. You should program this directly using the external interrupt registers and the ISR() interrupt vector definition macro.

Accelerometer Interrupt-Driven Input Demo__________________________TA

Part 7 – Accelerometer Control of Tri-color LEDs

Now add this accelerometer interface to the program you wrote for controlling the tri-color LEDs. Use one axis to control the output color on a continuum between red and green and the second axis to control the LED brightness from off to bright. You may use the analogWrite to control the LED brightness. You will have to experiment with your accelerometer to find the min and max values for each axis.

Accelerometer Control of LEDs Demo__________________________TA