CEG 4330/6330 – Microprocessor-based Embedded Systems

Lab 3 – Timer, Interrupt, and Serial I/O

Learning Objectives
The purpose of this lab is for you to practice the usage of timer output compare feature for waveform generation, low power mode with interrupt, serial I/O, and analog inputs.

Overview
In this lab you are going to create different projects using peripherals for the Arduino, including push buttons, speakers, LEDs, and keypads.

Keep all of your source code. They are needed later on for system integration.

Complete the following milestones for this lab. To earn points, demonstrate the project to the TA. Do not discard your code for any of the steps, it will be needed for future labs. You are not allowed to use the delay() or delayMicroseconds() functions for any of these milestones (except for Low Power Mode with Interrupt and Serial I/O steps):

PWM with Timers

1. Read two positive integers from the keyboard using the serial monitor, a pin and a value. Then call the Arduino function analogWrite(pin, value) to generate a square wave using the pin and value. Use an oscilloscope to measure the frequency and duty cycle of the square wave. Save this sketch separately. ***Demonstrate this sketch to the TA.

   Note: Digital pins 5 and 6 are connected to the Timer0, pins 9 and 10 are connected to Timer1, and pins 3 and 11 are connected to Timer2.

2. Using the Timer1 Fast PWM mode, generate a square wave on digital pin 10 by manipulating the Timer1 registers, TCCR1A, TCCR1B, etc. Modify the registers inside the Arduino's setup function, and leave the loop function empty. Use an oscilloscope to measure the frequency and duty cycle of the square wave. Do not use an interrupt service routine. Save this sketch separately. ***Demonstrate this sketch to the TA.

   Note: Digital pin 10 is connected to the Timer1 channel B output compare pin

3. Read two positive integers from the keyboard using the serial monitor, frequency and duty cycle. Using Timer1 in Fast PWM mode, generate a square wave on digital pin 10 with the frequency and duty cycle. Use an oscilloscope to measure the frequency and duty cycle of the square wave. Save this sketch separately. ***Demonstrate this sketch to the TA.

   Note: Some frequencies and duty cycles are not achievable, in these cases rounding to the nearest frequency and duty cycle is acceptable. Also, writing to the timer registers every loop iteration may cause some unpredictable behavior.

Low Power Mode with Interrupt and Serial I/O
Use new sketches for the following experiments:

1. Copy the code from:
   http://playground.arduino.cc/Learning/ArduinoSleepCode
   Study the code and learn how it works.

2. Modify the previous sketch by removing all the code related to the "go to sleep after ten seconds" feature. Make sure the serial I/O (i.e., serial monitor) part still works.

3. Use a pushbutton on pin 8 without an external pull-up resistor. Modify the previous sketch so that the UNO stays sleeping unless a serial monitor command is received or the push-button is pressed. Record the amount of button presses and print them to the IDE serial monitor with each button press. Debouncing may be needed. ***Demonstrate this sketch to the TA.

   Hint: This requires learning a new type of interrupt: pin change interrupt.

Optical Metronome with a Flashing LED
Use new sketches for the following experiments:

1. Using a function generator, generate a 500 to 2KHz, 0V to 5V square wave. Verify the signal using an oscilloscope. Demonstrate this step before connecting to Arduino to avoid damaging the Arduino. ***Demonstrate this step to the TA.

2. Connect the function generator output to digital pin 7 of the Arduino. Write a sketch to measure the frequency and duty cycle of the square wave input and then print out the measured values to the serial monitor. Only print when the frequency and duty cycle are different from the previous values.

   Hint: You may use the Arduino function pulseIn( ).

3. Connect an IR-proximity sensor (Sparkfun SEN-12728) to an analog input pin and connect an LED and a resistor to digital pin 10. Use a function generator to produce a 500 to 2KHz square wave on digital pin 7. Write a sketch that generates a PWM signal on pin 10 without the analogWrite( ) function. This PWM signal must have a frequency of at least 1KHz and the duty cycle must be proportional to the ADC input voltage from the IR proximity sensor.

   Next the LED should slowly blink at a frequency that is equal to Fin/2,000, where Fin is the frequency of the square wave connected to digital pin 7. For example, if Fin is 2KHz, then the LED should blink at 1 Hz (on for 0.5 seconds, off for 0.5 seconds), if Fin is 500Hz then the LED should blink at 0.25Hz. While the LED is on, the brightness must still be proportional to the IR sensor analog voltage. ***Demonstrate this sketch to the TA.

   Note: using the map() function may make the analog input easier to work with for changing the brightness.

How to Submit

Demonstrate the various milestones mentioned above to the TA in person. Submit all programs to Pilot.

Grading

This lab is worth 7.5 points, distributed as follows:

Task	Points
PWM with Timers (Part 1)	0.25
PWM with Timers (Part 2)	0.25
PWM with Timers (Part 3)	1.50
Low Power Mode with Interrupt and Serial I/O (Part 3)	2.00
Optical Metronome (Part 1)	1.50
Optical Metronome (Part 3)	2.00
Total	7.50