Counters and Pulse Width Modulation

**Objectives:** The purpose of this lab is to learn how to generate a pulse width modulated signal using the real time interrupt system and the output compare function used in the previous lab.

**Prelab Questions:**
1. What counter values are necessary (High, Low, Period) to generate a 20% duty cycle, 800 Hz waveform?
2. What is pulse width modulation?
3. Name some uses of pulse width modulation.

**References**

**Equipment for this lab:**
- 68HC11 trainer kit, to include 68HC11 EVBU and prototyping strips.
- IBM compatible PC to connect to the trainer kit via an RS-232 serial cable.
- Agilent 54621D oscilloscope.
- Floppy disk provided by student.

**Laboratory Exercise**

Notes:
- Make sure your development EVBU board is connected to power (green LED on board ON)
and the serial port of the EVBU is connected to the serial port of your development PC
containing the AXIDE software, configured to the correct port at the correct baud rate, etc.
- Be sure to bring your interface-students.RTF file from 68HC11 lab #2.
- The lab TA has a copy of the .s19 files for each part of the lab. The TA will
  demonstrate the waveforms to you at the beginning of the class.

**Part 1**
In this part, you will generate a PWM signal by changing the duty cycle of the waveform generated in the previous lab. This concept of PWM will later be used to control the speed of the DC Motor. Use the Real Time Interrupt generated in the previous lab to modify the duty cycle of the waveform from 5% to 95% in increments of 1%. That is, modify the values of “High” and “Low” inside the Real Time Interrupt service routine.
When the duty cycle reaches 95%, then reset it back to 5%. The Period of the waveform doesn’t change. Note that both interrupt service routines are accessing the same data (High, Low, Period) and appropriate precautions should be taken when modifying shared data. Make sure the duty cycle increases from 5% to 95% (not decreases from 95% to 5%).

The required subroutines are provided to you on the website (stud_PWM.RTF). Complete the code and check the output at pin PA5/OC3 on the board.

T. A. Initials:______________________

**Part 2**

In the last part of this lab, you will create a timer that will count the number of minutes and seconds that a program has been running (well, at least since the interrupts were enable, that is). You will use the timer for the program developed in lab2. The interrupt should be enabled after the first message is displayed. So, the timer should display the amount of time the program has been running, with every message displayed. The time should be displayed on the fourth line of the display. Your program should be able to display at least 10’s of minutes properly and should have two fixed places for the number of seconds (e.g. the time display should appear as “24:06” if the program has been running for 24 minutes and 6 seconds). Below is a suggested set of steps for developing the timer:

- Generate an interrupt at a known rate (e.g. 20 ms).
- Keep a counter that counts up to the number of interrupts equivalent to one second.
- Increment the number of seconds at that point and reset the counter.
- When the number of seconds reaches 60, increment the number of minutes and reset the seconds.
- Append a subroutine that can be called that displays the current minute and second values and call it from the appropriate portions of the main program loop.

Bring the program that you used in Lab2 (interface-students.RTF) and make required additions to the code to display the time.

Demonstrate the functional time-keeping part of the motor control interface.

T. A. Initials:______________________