Department of Electrical and Computing Engineering

UNIVERSITY OF CONNECTICUT

ECE 3411 Microprocessor Application Lab: Spring 2019

Lab Test 5

This two-page booklet contains one multi-part test problem. Answer the questions according to the instructions given.

You have **120 minutes (3:30 p.m. to 5:30 p.m.)** to program your AVR. You must upload the code by PDF/Word via Husky CT “Lab Test 1” by 5:30 p.m. The tasks in this test are also to be demonstrated to the instructor or TA from 5:00 p.m. – 6:00 p.m.

Answer questions sequentially to complete the tasks easily — you may want to skim all questions before starting. If you find a question ambiguous, be sure to write down any assumptions you make. For clarification, you may watch the TA demo video clip in HuskyCT.

Be neat and legible. If we can’t understand your answer, we can’t give you credit! Write your name in the space below. Write your initials at the bottom of each page.

THIS IS AN OPEN BOOK, OPEN NOTES TEST. YOU CAN USE YOUR LAPTOP.

Any form of communication with other students is considered **cheating** and will merit an F as final grade in the course.

Do not write in the boxes below

a(x/10)	b(x/30)	c(x/40)	d(x/20)	Total 100(xx/100)

Name:

Student ID:

Q1. [100 points] Write C code to program the ATmega328PB XPlained mini kit and demonstrate that it implements the following tasks:

a. [10 points] Configure a timer to generate a 4 kHz PWM signal with at least 1% duty resolution.
b. [30 points] Set up the ADC based temperature sensor (MCP9701) and photo sensor (PDV-P9001) to display the ambient temperature and light on the LCD.
   → The first row of the LCD will display the temperature in Celsius. The display format is "**T(C)=23.18**"
   → Second row will display the brightness level between 0 to 10. The format display format is "**Bright = 5**". Note: If photo-sensor gives 0V, then bright level =0; if it gives 0.5V, bright level =1; if 1V, bright level =2;…….. if 5V, bright level =10.
   *Read the values from the ADC. Use the full 10-bit ADC resolution.
   Average the values ADC readings and refresh the LCD every 500ms. At the end of the LCD refresh, toggle a LED connected to **PD1.
   ***You are not allowed to use the delay_ms or delay_us function.
c. [40 points] Use SPI to control the glow of LED.
   → Set up the clock frequency of SPI0 to 250kHz (period 4μs).
   → Use SPI based DAC (MCP4921). Connect a LED and a resistor to the output of the DAC to control the glow of LED.
   → Control the glow of LED in the range of level 0 to 10. If level is 0, the LED will be completely off or emit minimum light. If level is 10, then LED will glow with maximum brightness.
   → The level of the LED should be corresponding to the brightness level displayed in the second row of the LCD.

d. [20 points] Modulate the brightness level using the 4 kHz PWM signal that you configured in part (a).
   → Duty of the PWM will be 0 to 100% based on the brightness levels. i.e. if level is 5, then duty is 50%.
   → The described functions in (a), (b), and (c) will be fully functional with (d).