Department of Electrical and Computing Engineering

UNIVERSITY OF CONNECTICUT

ECE 3411 Microprocessor Application Lab: Spring 2019

Lab Test 6

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/10)	c(x/10)	d(x/30)	e(x/20)	f(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 3 kHz PWM signal with at least 1% duty resolution.
b. [10 points] Configure SPI to control the glow of LED.
   → Use SPI based DAC (MCP4921). Connect a LED and a resistor to the output of the DAC to control the glow of LED.
c. [10 points] Set up the ADC based temperature sensor (MCP9701) to display the ambient temperature on the LCD.
   → The first row of the LCD will display the temperature in Celsius. The display format is "**T1(C)=23.18**"
   *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.
d. [30 points] Set up I2C based temperature sensor (TC740A) to sense the ambient temperature.
   → The second row of the LCD will display the temperature in Celsius. The display format is "**T2(C)=23.00**"
e. [20 points] Modulate PWM and control SPI based on the temperature readings.
   → The PWM duty will be ten times the last digit of the temperature reading. i.e. if "T1(C)=2**3**.18", duty ratio will be **3***10=30%.
   → The SPI output voltage will be half of the last digit of the temperature reading. i.e. if "T2(C)=2**3**.00", The output voltage will be **3***0.5=1.5V.
   → The input and corresponding output options for this case are in the following table. This is mode 1(default mode).

Input	Output
T1	PWM
T2	SPI

f. [20 points] Use 2 switches to control change the mode of output.
→ Initially the output will be in mode 1 as describe in (e). Switches are not pressed in this case.
→ If SW1 is pressed, then program will run in mode 2.
→ If SW2 is pressed, then program will run in mode 3.
→ If both SW1 and SW2 are pressed, then program will run in mode 4.
→ There would be 4 mode of operation. The modes are defined by following table:

Mode 1		Mode 2		Mode 3		Mode 4	
Input	Output	Input	Output	Input	Output	Input	Output
T1	PWM	T2	PWM	T2	PWM	T1	PWM
T2	SPI	T1	SPI	T2	SPI	T1	SPI