CSCE 236 Embedded Systems, Spring 2019
Test 2
Duration: Class Duration

Thursday, April 25th, 2019

Instructions: Write your name on every sheet. You will have the full class period to complete this test. **Make sure to show your work to ensure you receive partial credit if your final answer is incorrect.** This is a closed book quiz, no computers, textbooks, notes, etc. are allowed. This test is out of 100 points.

Unless otherwise specified, assume that questions refer to Arduino/Atmel processor we have been using in class.

Name:

1 Multiple Choice (Circle all answers that apply).

(i) (5 pts) Which of the following doe the **sei()** function do?

(a) Serial Port interrupt disable

(b) Timer 1 interrupt enable

(c) Global interrupt disable

(d) Global interrupt enable

(ii) (6 pts) Which of the following could lead to a DC motor overheating? Answer any or all that apply.

(a) Very small delays between change of directions

(b) Braking the motors between direction change

(c) Not braking the motors between direction change

(d) Locking the motor in one direction

(e) Applying PWM to the motor input

(f) Power supply exceeding the required rating

(g) Power supply not enough for powering up motors at full speed

(iii) (5 pts) The *I2C* protocol has which of the following model?

(a) Many Masters, many slaves

(b) 1 Master, many slaves

(c) Full Duplex

(d) No Masters

(iv) (5 pts) When is a timer overflow flag set when using Phase Correct PWM?

(a) When timer reaches starting value

(b) When timer reaches MAX value

(c) When timer reaches BOTTOM value

(d) When timer reaches OCRA value

(v) (4 pts) If a PWM signal was 300Hz in Phase Correct Mode, approximately what would the frequency be in Fast PWM mode?

(a) 100Hz

(b) 150Hz

(c) 300Hz

(d) 600Hz

2 Bit Masking

Hex and bit operations (all references to bit locations are zero referenced). For each bit operation sub problem write a single line of C code to achieve the desired result.

a) (5 pts) Set the lower 4 bits in the 8-bit variable var to 0x9.

b) (5 pts) Clear the top 7 bits in the 16-bit variable var.

c) (5 pts) What is the value (in hex) of the 8-bit variable var after this operation var = ((13«2) | (4«5))?

d) (5 pts) What is the 8-bit value of ((0x56»3) + 3) in hex?

e) (5 pts) Set the upper 5 bits in the 8-bit variable var to the lower 5 bits in the 16-bit variable config. (Remember to do this in a single line of code.)

3 Serial

(i) (2 pts) Expand the acronym UART.

(ii) (2 pts) Expand the acronym SPI.

(iii) (2 pts) Give the major difference between serial vs parallel communication.

(iv) (5 pts) Consider an I2C set up as follows. The arduino is connected to an actuator and a serial interface with another arduino using I2C. Serial fires an interrupt and calls the corresponding ISR when it receives the input data. Then the required action is performed based on the data received on the serial using ***another function*** in your code. For this data needs to be shared between the ISR and the function that performs the action. Explain conceptually how would you share data between the ISR and the function.

(v) (5 pts) Suppose your project has several modules. Each module performs a separate task and contains several C functions. Currently, you have written your code for all the modules in a single C file. Due to the requirements of the project, you would need to share the data between all the functions within each single module. You know that using GLOBAL variables is not a recommended practice. Explain briefly how would you achieve the problem of sharing data between the functions within each module.

4 PWM and Timer/Counter

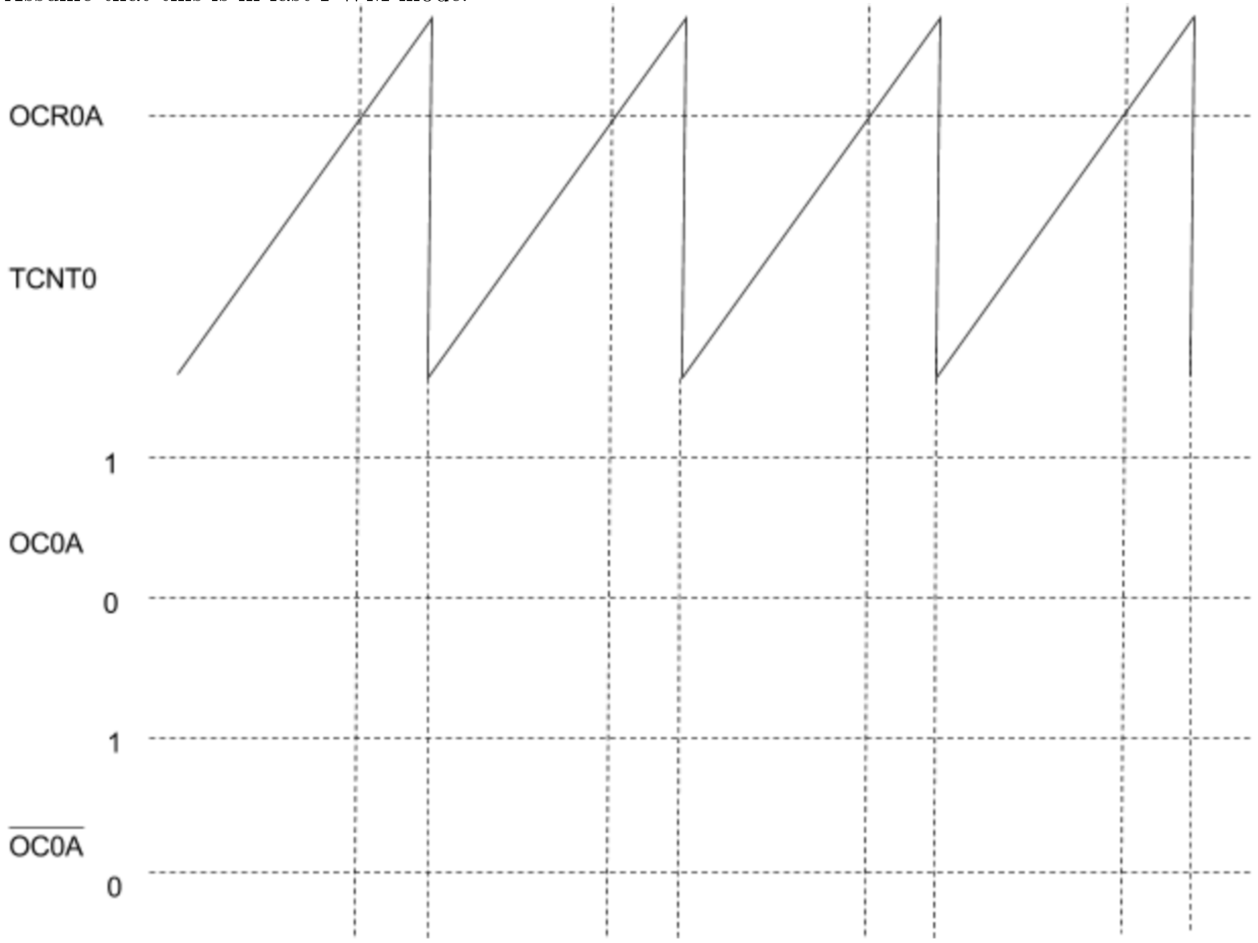
(i) (5 pts) Describe one benefit and one drawback of using Timer interrupts in your code.

(ii) (4 pts) Using Timer0 for timing operations is not recommended. Why do you think so?

(iii) (5 pts) Digital to Analog converters can be used to produce an analog signal, which in turn can be used to control the motors. However this is not preferable over PWM. Why do you think so?

(iv)(5 pts) Did you use PWM in your project? Why or why not?

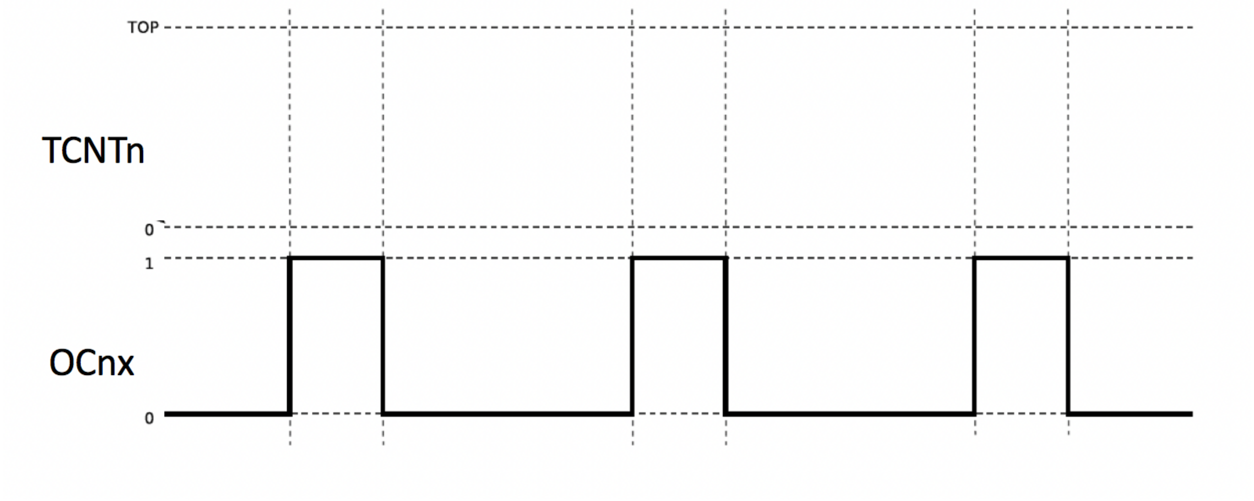
(v)(10 pts)In the following figure, draw the signal output on the pin OC0A and $\overline{OC0A}$ (indicates the inverted mode) given the clock value TCNT0 (zig-zag line) and the specified output compare register value, OCR0A. Assume that this is in fast PWM mode. [1]



[1]When configured to non-inverted mode, OC0A is cleared on Compare Match and set at BOTTOM. In inverted opposite happens

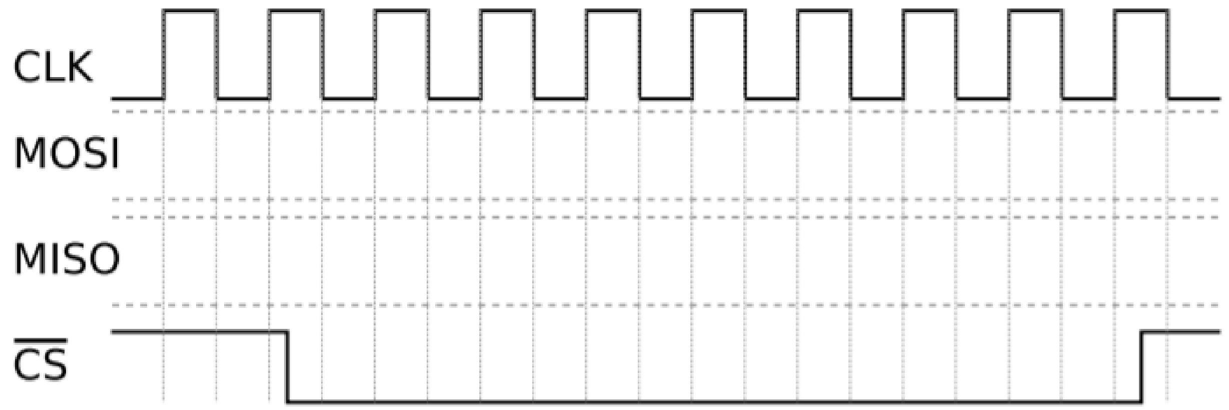
(vi) (5 pts) This is an open ended question. Clearly explain your reasoning to score full points.
In Fast PWM mode, PWM waveform can be generated on OCnx pins by setting the output when TCNTn (i.e. the timer) reaches the TOP and clearing when TCNTn matches with OCRnx (i.e. compare match).
In CTC mode, TCNTn is cleared to ZERO when its value matches the OCnx register (i.e. compare match). It then restarts counting. A waveform in CTC mode can be generated on OCn pin by toggling the output on each compare match.
Do you think CTC mode can be used to generate a PWM waveform? What are the drawbacks, if any, when using the CTC mode?

(vii) (5 pts) In the following figure, draw the clock, TCNTn, that will generate the lower square wave when it is configured in **Phase Correct PWM** mode. Also draw the correct line in the upper part of the diagram to indicate the location of OCRnx. **Note:** In non-inverting Compare Output mode, the Output Compare (OC1x) bit is cleared on compare match between TCNT1 and OCR1x while up-counting, and OC1x is set on the compare match while down-counting.



5 Communication

(i) (5 pts) In the following figure, draw the proper signal for sending the from the master to the slave the value 0xA5 on the rising edge and from the slave to the master the value 0x63 on the falling edge. Assume that the data is transmitted MSB first (i.e. human readable on the waveform).



(ii) (5 pts) How many **bytes** could be sent per second with a baud rate of 115200 when configured as (you can leave the answer as a fraction):?

```
//Use double speed here
UCSR0A = (1<<U2X0);
//Enable tx and rx
UCSR0B = (1<<RXEN0)|(1<<TXEN0);
//Set frame format to 7 data, 2 stop bits, even parity
UCSR0C = (3<<UCSZ01) | (1<<USBS0) | (1<<UPM01);
```

6 Interrupts

Interrupt Example Code. For this problem, refer to the following code. This code monitors two of the external interrupts connected to two different buttons. The goal of the code is to turn on one of the LEDs if the counter is less than zero, the other when the counter is greater than zero, and to have both off when counter is zero. Note that during the **SIGNALs** interrupts are disabled.

```
00: int32_t counter = 0;
01:
02: SIGNAL(INT0_vect){
03:   counter++
04: }
05:
06: SIGNAL(INT1_vect){
07:   counter--;
08: }
09:
10: void loop(){
11:   if(counter < 0){
12:      //Red on
13:      digitalWrite(LED_RED,HIGH);
14:   }else if(counter > 0){
15:      //Green on
16:      digitalWrite(LED_GREEN,HIGH);
17:   }else if(counter == 0){
18:      //Both off
19:      digitalWrite(LED_RED,LOW);
20:      digitalWrite(LED_GREEN,LOW);
21:   }
22: }
```

(i) (5 pts) The above code does not work properly. Describe how this code could end up with both LEDs on (at least until the buttons are pressed again). Refer to line numbers to help in your explanation

7 Documentation And Code Use

The following two questions cover the importance of documentation and hence are open ended questions.
(i) (5 pts) Why do you think using version control (like git) is important during the development process of an embedded system application (like your project)?

(ii) Bonus: (5 pts) Provide one specific situation where version control helped you during the development process of your project.