*Department of Electrical and Computing Engineering*

UNIVERSITY OF CONNECTICUT

**ECE 3411 Microprocessor Application Lab: Fall 2015**

# Quiz V

There are 3 questions in this quiz. There are 8 pages in this quiz booklet. Answer each question according to the instructions given.

You have **45 minutes** to answer the questions.

Some questions are harder than others and some questions earn more points than others—you may want to skim all questions before starting.

If you find a question ambiguous, be sure to write down any assumptions you make.
**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 A CLOSED BOOK, CLOSED NOTES QUIZ.**
**PLEASE TURN YOUR NETWORK DEVICES OFF.**

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*

| 1 (x/40) | 2 (x/40) | 3 (x/20) | Total (xx/100) |
|---|---|---|---|
| | | | |

**Name:**

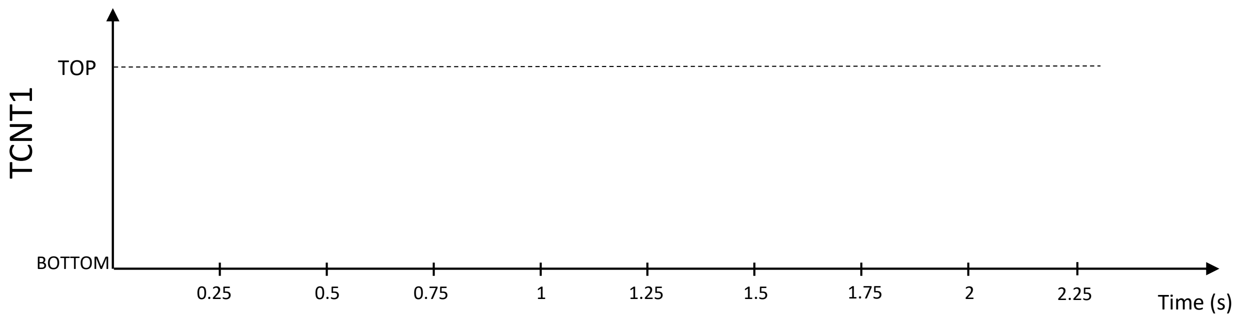
**Student ID:**

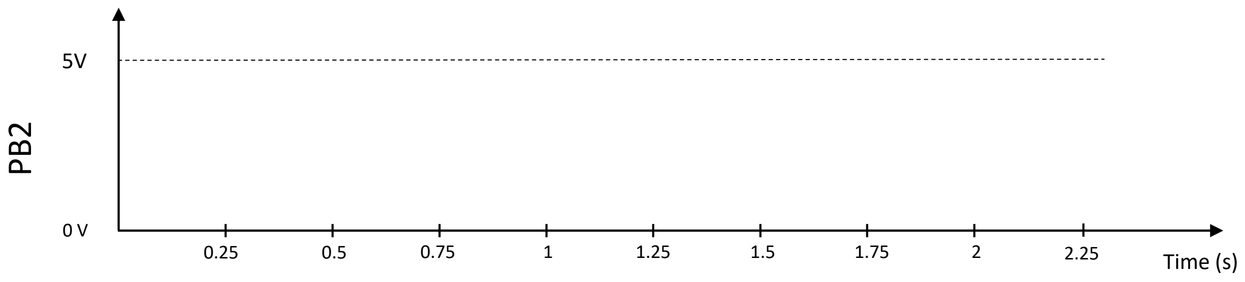
**1. [40 points]:** Assume a clock frequency of $f_{clk} = 16\text{MHz}$. Read the following initialization and ISRs:

```
#define MIN_TICKS 15624
#define MAX_TICKS 62499

// PWM variables
volatile uint16_t duty_cycle;
volatile uint16_t time_period;
volatile uint8_t toggle_flag;
int percentage_duty_cycle;

void initialization ()
{

    DDRB |= (1<<DDB2);
    time_period = MAX_TICKS;
    duty_cycle = time_period/4;
    toggle_flag = 0;

    // Setup Timer1
    OCR1A = time_period;
    OCR1B = duty_cycle;
    TCCR1A |= (1<<WGM11) | (1<<WGM10);
    TCCR1B |= (1<<WGM13) | (1<<WGM12);
    TCCR1A |= (1<<COM1B1);
    TIMSK1 |= (1<<OCIE1A);
    TCCR1B |= (1<<CS12);
}
// Timer 1 Compare Match A ISR (TCNT1 = OCR1A)
ISR (TIMER1_COMPA_vect)
{
    if(toggle_flag)
    {
        if( time_period > MIN_TICKS)
        {
            time_period = time_period/2;
            duty_cycle = time_period/4;
        }
        else
        {
            toggle_flag ^= 1;
        }
    }

    else
    {
        if( time_period < MAX_TICKS)
```

```
            {
                time_period = (time_period * 2) +1;
                duty_cycle = time_period/4;
            }
            else
            {
                toggle_flag ^= 1;
            }
        }

        OCR1A = time_period;
        OCR1B = duty_cycle;
    }
```

Starting from the moment of `Timer1`'s initialization, draw the waveforms of the `TCNT1` register value and the pin PB2 value w.r.t. time. Please draw the waveform strictly according to the timing scale shown on X-axis, otherwise no credit will be given.





**Initials:**

**2. [40 points]:** Given that the clock frequency ($clk_{I/O}$) of ATmega328P is 16MHz, you want to implement two pins (A and B) that output PWM signals.

**a.** The frequency of each PWM signal is 1MHz, and the initial duty cycle of these two PWM signals is 50%. Please write the initialization function for timer0 (signal A) and timer1 (signal B).

```
void initialization ()
{









}
```

**b.** The duty cycle of A and B (duty_cycle_A and duty_cycle_B) should be updated in the ISRs according to the following rules:

(a) For A: If duty_cycle_B $> 90\%$, then duty_cycle_A = (1+duty_cycle_A)/2. If duty_cycle_B $< 10\%$, then duty_cycle_A = duty_cycle_A /2. In other cases, duty_cycle_A does not change.

(b) For B: If duty_cycle_A $< 50\%$, then duty_cycle_B = (1+duty_cycle_B)/2. If duty_cycle_A $\geq 50\%$, then duty_cycle_B = duty_cycle_B /2.

Please write the ISRs for these two timers.

```
ISR (TIMER0_COMPA_vect)
{
}
```

```
ISR (TIMER1_COMPA_vect)
{
}
```

**Initials:**

**3. [20 points]:** Given that clock frequency ($clk_{I/O}$) of ATmega328P is 8MHz.
Assume the following about the code snippet given below:

- Each one of `instruction_1`, `instruction_2`, . . . , `instruction_52` takes 4 CPU cycles.
- Evaluating `while(1)` statement takes zero CPU cycles.
- Evaluating `if( !(ADCSRA & (1<<ADSC)) )` statement and executing its body take zero CPU cycles.

```
/*********** ECE3411 Quiz 5, Task 3 ************/
#define F_CPU 8000000UL
#include <avr/io.h>

/* Main Function */
int main(void)
{
    /* Configuring ADC Control and Status Register A */
    ADCSRA = 0x86;

    while(1)
    {
        instruction_1;
        instruction_2;
        instruction_3;
        ...
        ...
        instruction_52;
        if( !(ADCSRA & (1<<ADSC)) )
        {
            ADCSRA |= (1<<ADSC);    // Start A to D Conversion
        }

    } /* End of while(1) Loop */

} /* End of main() */
```

Answer the following questions about the code snippet.

**Initials:**

**a.** Given that it takes 13 ADC cycles, how much time (in microseconds) does it take to complete one ADC conversion?

**b.** What prevents the condition "`if( !(ADCSRA & (1<<ADSC)) )`" from being satisfied?

**c.** How much time (in microseconds) does it take to complete one iteration of "`while(1)`" loop?

**d.** What is the percentage of `while(1)` loop iterations for which the body of "`if( !(ADCSRA & (1<<ADSC)) )`" condition is executed?

**Initials:**

## End of Quiz

Please double check that you wrote your name on the front of the quiz.

**Initials:**