USAF ACADEMY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

ECE 382 GRADED REVIEW #2
FALL 2016

Name: _____ **Section:** _____

Academic Security	This examination is not released from academic security until **1630** on **11 November 2016**. Until this time, you may not discuss the examination contents or the course material with anyone other than your instructor.
Integrity	Your honor is extremely important. This academic security policy is designed to help you succeed in meeting academic requirements while practicing the honorable behavior our country rightfully demands of its military. Do not compromise your integrity by violating academic security or by taking unfair advantage of your classmates.
Authorized Resources	1. MSP430 blue reference book 2. Any calculator without ECE 382 equations or concepts
Instructions	▪ Show all work for full credit ▪ Box or circle your final answer. ▪ For all numerical answers, use engineering notation and include units. ▪ Completely label all your diagrams, drawings, graphs, etc. for full credit. ▪ Assume the use of the MSP430 and Code Composer Studio. ▪ You have **53 minutes** to complete this exam.

Problem	Value	Earned
1	20	
2	20	
3	24	
4	16	
5	20	
Total	100	

This page intentionally left blank

Problem 1 (20 points) *General Knowledge*

a. (4 points) List one advantage of using C and one advantage of using assembly when programming a microprocessor.

C –

Assembly –

b. (2 points each) Circle the correct answer.

T F If multiple interrupts occur, the lowest priority interrupt is handled first.

T F A header file should contain typedef declarations and function prototypes.

T F Setting BIT3 in the Status Register enables maskable interrupts.

T F When an interrupt is triggered on the MSP430, all general-purpose registers are preserved while the ISR executes.

c. (2 points) You want to take your newfound knowledge of C and revisit serial SPI communication. Which line of code will clear the transmit interrupt flag in the B serial module?

i. UCB0TXBUF |= UCB0TXIFG;
ii. UCB0TXBUF |= ~UCB0TXIFG;
iii. UCB0TXBUF &= UCB0TXIFG;
iv. UCB0TXBUF &= ~UCB0TXIFG;

d. (2 points) When one desires to modify the **actual** variable (not a copy of the variable) in a subroutine, one should use the pass by _____ technique to give the subroutine the information it needs to perform its modification.

e. (2 points each) Part of this class involves learning how to glean useful information from the technical documentation provided on a particular microcontroller. Look at the ADC10 section of your blue book. The analog-to-digital converter module allows you to choose the voltage reference level for this subsystem by changing the REF2_5 bit.

i. What are the two voltage levels the REF2_5 bit chooses between?

ii. What is the name of the register in which this REF2_5 bit is located?

Problem 2 (20 points) *Pointers, Dereferencing*

a. (4 points) Fill in the second row of the memory map below for the four lines of the initialized data below. For any spot that is not modified by the below code, insert 0x00. **The third row is not graded but may help with partial credit.**

```
char A = 9;          //at address 0x03EC
int B = 0x3456;      //at address 0x03EE
char *ptrA = &A;     //at next available word aligned spot
int *ptrB = &B;      //at next available word aligned spot
```

Address	0x03EC	0x03ED	0x03EE	0x03EF	0x03F0	0x03F1	0x03F2	0x03F3
Value								
Variable								

b. (2 points) Fill in the second row of the memory map below for the four lines of code below. For any spot that is not modified by the below code, insert 0x00. **The third row is not graded but may help with partial credit.**

```
char Y[3]= {0x23, 0x45, 0x86}; //at 0x03F4
char *ptrY = &Y; //next available word aligned spot!
```

Address	0x03F4	0x03F5	0x03F6	0x03F7	0x03F8	0x03F9
Value						
Variable						

]

Problem 2 cont'd *Pointers, Dereferencing*

(14 points) You are given the following memory map and code snippets. Use this information to answer the subsequent questions.

Address	Value	Variable
0x02EE	0x0035	
0x02F0	0xFA22	
0x02F2	0x2007	
0x02F4	0xBB08	
0x02F6	0x1234	
0x02F8	0x02EE	xPtr
0x02FA	0x02F0	yPtr
0x02FC	0x02F2	zPtr
0x02FE	0x99A5	

```
// xPtr is located at 0x02F8 and points to an char X
// yPtr is located at 0x02FA and points to an int Y
// zPtr is located at 0x02FC and points to array char Z[5]
```

c. (4 points) What is the value of `x` and `xPtr` after the following statement?

```
*xPtr = 0x40;
```

X: _____

xPtr: _____

d. (5 points) What is the value of `y` and `yPtr` after the following statement?

```
y = yPtr + *yPtr;
```

Y: _____

yPtr: _____

e. (5 points) What is the value of `Z[0]` and `zPtr` after the following statements? **Remember, the MSP430 stores values as little-endian.**

```
zPtr++;
Z[0] = *zPtr + 5;
```

Z[0]: _____

zPtr: _____

Problem 3 (24 points) *Interrupts, Timer_A*

a. You want to use Timer_A to measure a delay of 100 ms. You are using a 1 MHz SMCLK with an input divider of 4, and your timer is configured to use the Up/Down mode.

   i. (6 points) What value will TAR need to be compared with in order to achieve this delay?

   ii. (4 points) Write the lines of code necessary to configure Timer_A for this delay. If you did not get an answer for part A, use 2006. Hint: you can use just two lines. (You are not using interrupts.)

b. (8 points) Now that you have mastered the Timer_A portion of your code, you want to use a GPIO interrupt to toggle the red LED whenever the button (P1.3) is pushed. Write the four lines of code that will clear the interrupt flag for P1.3, trigger the interrupt on the falling edge of the button signal, and enable both local and global interrupts.

c. (4 points) You add your code developed in parts a) and b) above into a program that uses another button connected to P2.4 that will toggle the green LED when pressed (also in an ISR). Your friend Marie wonders what would happen if both buttons were pressed exactly at the same time (while both LEDs were off). Select the best answer to give to Marie.

   i. Both lights will turn on at the exact same time
   ii. Only the red will illuminate
   iii. Only the green will illuminate
   iv. The red will turn on first, followed quickly by the green
   v. The green will turn on first, followed quickly by the red

Briefly explain why you chose the answer you did.

d. (2 points) Your friend Kimo wants to learn more about the two lines of code **inside** the interrupt. One line turns the LED on or off. What should the other line of code do?

Problem 4 (16 points) *Converting between C and Assembly*

a. (12 points) Convert the following MSP430G2553 assembly code to C using a **for** loop. You are able to use the Delay function in C **as long as you expose the interface**. Assume the LED is already configured.

```
; P1.0 connects to an LED
; Delay is defined elsewhere

      bis.b  #BIT0, &P1DIR
      mov    #0xA, r12
      call   #mystery

forever:
      jmp    forever


mystery:
      mov    r12, r6
doAgain:
      cmp    #0, r6
      jz     done
      bis.b  #BIT0, &P1OUT
      call   #Delay
      call   #Delay
      call   #Delay
      bic.b  #BIT0, &P1OUT
      call   #Delay
      dec    r6
      jmp    doAgain
done:
      ret
```

Your C code goes here:

```
/* Remember, use a **for** loop for this answer
   Don't forget to make your assembly Delay function
available to C  */
#include <msp430.h>
// Function Prototypes




void main(void) {
    WDTCTL = WDTPW | WDTHOLD;    // Stop watchdog
    P1DIR |= BIT0;




} // end main
void mystery(                    ) {









} // end mystery
```

b. (4 points) Clearly explain to my grandma what the code in part a) does. Be specific.

Problem 5 (20 points) *C*

a. (6 points) Assuming the use of CCS for the MSP430, and given the below function myFunct, answer the following questions:

```
f = myFunct(a, b, c, d, e)
```

In which register is 'c' passed? _____

How is 'e' passed? _____

How is 'f' returned? _____

b. (3 points) What does the following line of code do? Choose the BEST answer.

P2DIR = BIT6;

i) does not affect P2.7

ii) sets P2.7 as an Input

iii) sets P2.7 as an Output

iv) does not affect P2.6

v) sets BIT6 equal to P2DIR

vi) only sets P2.6 as an Output

c. (3 points) Given the code segment below, select the correct final value of solution.

```
int solution = 0;
int a = 0x03;
int b = 0x05;
if (a & b){
      solution = a * b;
}
else{
      solution = a + b;
   }
```

i. solution = 0x00

ii. solution = 0x1

iii. solution = 0x15

iv. solution = 0x7

v. solution = 0x8

vi. solution = 0xF

d. (4 points) Explain the difference between "&" and "&&" and give an example of an instance where each would be used.

e. (4 points) Write a sentence clearly and specifically explaining what the following line of code does:

```
while(IR_DECODER_PIN!=0);
```

Bonus

a. (3 bonus points) Describe the three changes to the Timer the below command does?

```
TA0CTL |= TACLR;
```