Name _____   Class hours _____

ME430 - Mechatronics
Examination I
Sept 19th, 2017

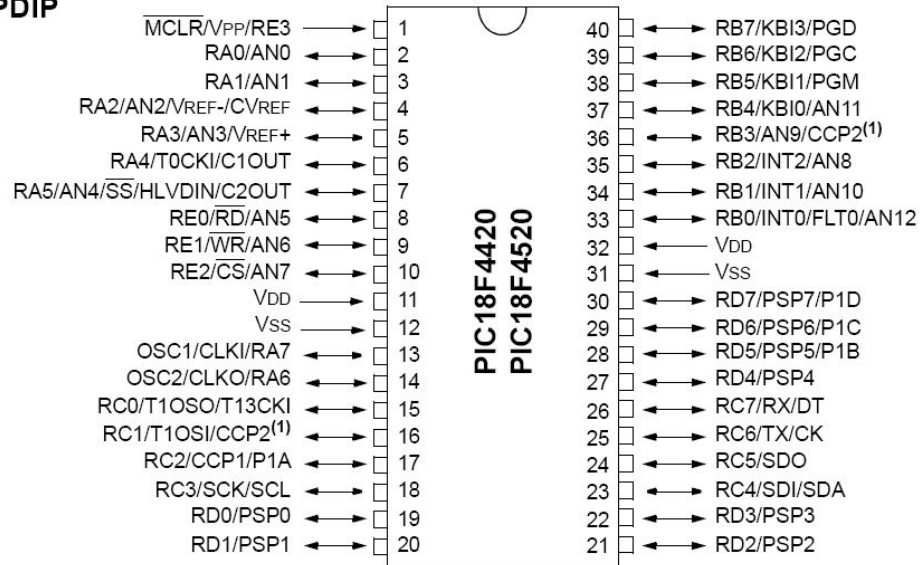| Problem	Score	
1	/ 16	
2	/ 6	
3	/ 20	
4	/ 20	
5	/ 14	
6	code	check off
	/ 22	/2
Total	/100	

You may use only:

- Your computer
- Any paper notes (such as notes on the videos) you brought to the exam, so long as those notes were written by you or your lab partner.
- Any electronic notes or code residing on your local (C:) hard drive, so long as those notes/code were written by you or your lab partner.
- The course website. (This is the only approved use of the internet for this exam.)
- A calculator (optional).
- Moodle for code submission.

Anything not specifically allowed is prohibited. In particular, you may not use notes or code written by someone outside your lab group.

Reference
This diagram can also be found on the course website. It is copied here as a reference.

40-pin PDIP

$\overline{\text{MCLR}}$/VPP/RE3	→	1	40	RB7/KBI3/PGD
RA0/AN0	↔	2	39	RB6/KBI2/PGC
RA1/AN1	↔	3	38	RB5/KBI1/PGM
RA2/AN2/VREF-/CVREF	↔	4	37	RB4/KBI0/AN11
RA3/AN3/VREF+	↔	5	36	RB3/AN9/CCP2[(1)]
RA4/T0CKI/C1OUT	↔	6	35	RB2/INT2/AN8
RA5/AN4/$\overline{\text{SS}}$/HLVDIN/C2OUT	↔	7	34	RB1/INT1/AN10
RE0/$\overline{\text{RD}}$/AN5	↔	8	33	RB0/INT0/FLT0/AN12
RE1/$\overline{\text{WR}}$/AN6	↔	9	32	VDD
RE2/$\overline{\text{CS}}$/AN7	↔	10	31	VSS
VDD	→	11	30	RD7/PSP7/P1D
VSS	→	12	29	RD6/PSP6/P1C
OSC1/CLKI/RA7	→	13	28	RD5/PSP5/P1B
OSC2/CLKO/RA6	↔	14	27	RD4/PSP4
RC0/T1OSO/T13CKI	↔	15	26	RC7/RX/DT
RC1/T1OSI/CCP2[(1)]	↔	16	25	RC6/TX/CK
RC2/CCP1/P1A	↔	17	24	RC5/SDO
RC3/SCK/SCL	↔	18	23	RC4/SDI/SDA
RD0/PSP0	↔	19	22	RD3/PSP3
RD1/PSP1	↔	20	21	RD2/PSP2

PIC18F4420
PIC18F4520

Problem 1 – Number systems and operators

Determine the output of each expression. Write the value stored as both an 8 bit binary number and as a signed char decimal number. YOU MAY NOT USE MPLAB or other programming tools to enter this code and print results. You may use a calculator or other similar calculator programs on your computer.

char x = 50 + 50;

Value stored in binary [ ][ ][ ][ ][ ][ ][ ][ ]   [   ] Decimal value if printed

char x = 100 + 100;

Value stored in binary [ ][ ][ ][ ][ ][ ][ ][ ]   [   ] Decimal value if printed

char x = -10;

Value stored in binary [ ][ ][ ][ ][ ][ ][ ][ ]   [   ] Decimal value if printed

char x = 44 / 5;

Value stored in binary [ ][ ][ ][ ][ ][ ][ ][ ]   [   ] Decimal value if printed

char x = 16 * 16;

Value stored in binary [ ][ ][ ][ ][ ][ ][ ][ ]   [   ] Decimal value if printed

char x = 30 % 7;

Value stored in binary [ ][ ][ ][ ][ ][ ][ ][ ]   [   ] Decimal value if printed

char x = 15 ^ 2;

Value stored in binary [ ][ ][ ][ ][ ][ ][ ][ ]   [   ] Decimal value if printed

Problem 2 - Binary
Assume that you have a variable called a **decabit** that is a signed 10 bit number. What is the largest and smallest value that you could hold in a signed **decabit**?

Smallest value	Largest value

Problem 3 - Stepper Motor

Use the components shown below to connect a stepper motor circuit.

- Assume you want to drive the stepper at 12v and you have an unregulated 12v power supply available (just write **+12v** next to any power symbols you add below that need 12 volts).
- Assume you also have a regulated 5 volt power supply (just write 5v next to any power symbols you add below that need 5 volts).
- Set up a basic switch circuit with the pushbutton and connect the signal line to RC0. Pretend like the PIC is already programmed and pushing the button will cause the stepper to spin 3 laps (or something like that).
- Connect the PIC pins as needed to allow it to run (assume it is already programmed).
- Connect the PIC to the H-bridge as follows:
    - RA0 and RA1 are a pair and need to drive one side of the H-bridge
    - RD0 and RD1 are a pair and need to drive the other side of the H-bridge
    - RB6 is a signal that can enable/disable the H-bridge
- Connect other H-Bridge pins as needed to drive the stepper at +12v
- Add snubber diodes, decoupling capacitors, and any necessary resistors for the circuit (size all resistors and label values, sizing capacitors is not required).

Problem 4 – BJT resistor sizing

Assume you have a 9 volt battery, voltage regulator, motor with a 96 ohm resistance, BJT, and PIC microcontroller. Draw the circuit connections below to use a PIC and drive the motor using the 9 volt battery. Do NOT draw power symbols this time, you must connect with wires all power lines. However you may draw ground symbols. Add resistors, snubber diodes, and capacitors as needed. Assume the PIC is programmed to control the motor using RB0. For the resistor to the BJT show your work for calculating the resistor size, put your exact resistor value in the left box, and real resistor size from the E12 series in the right box.

Exact BJT resistor value calculated	Real BJT resistor value

red 9v

black GND
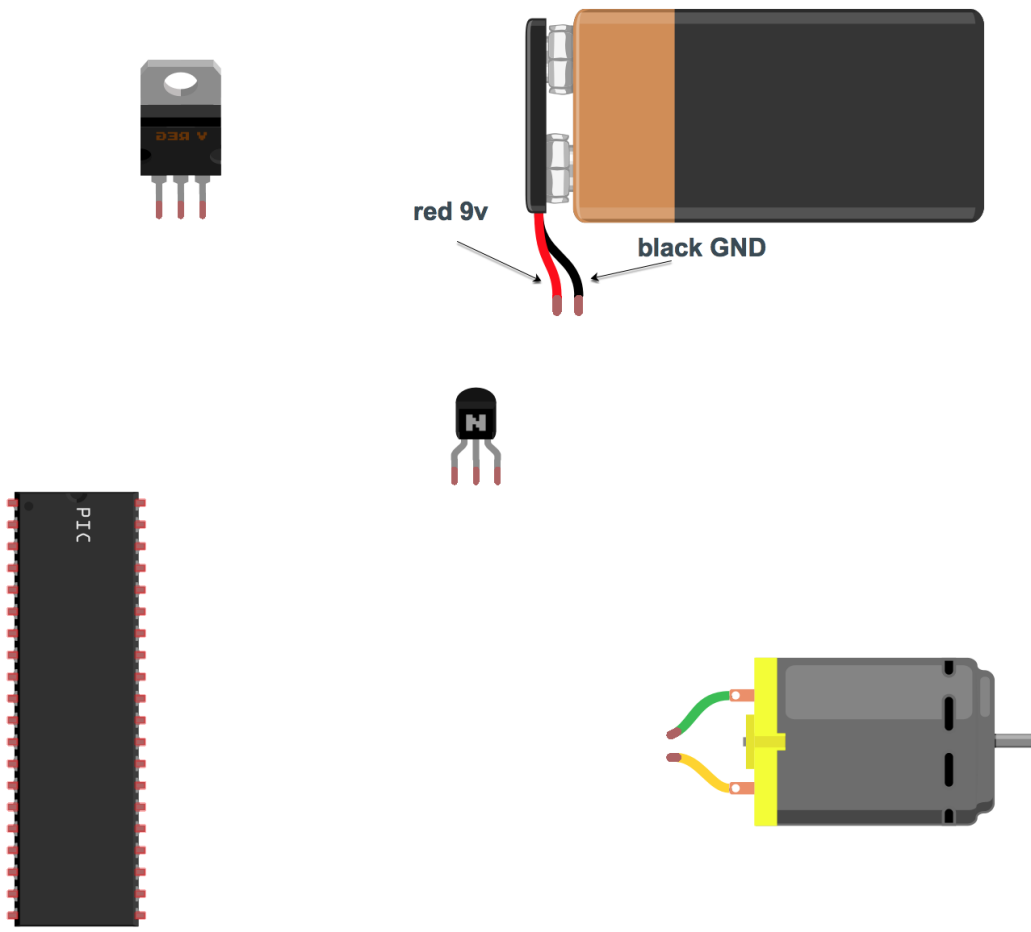
Problem 5 – C debugging
Consider the code scrap shown below.

```c
#include <stdio.h>

void main(void)
    int i;
    int x = 0;

    for (i = 0; i < 10; i = i + 2) {
        x = x + i
        if (x = 8) {
            fprintf("Getting big!\n");
        }
    }
    fprintf("x = %d\n", x);

    while (1) {
    }
}
```

a. This code contains syntax errors and won't compile. Mark each syntax error mistake with an arrow, label it as **SYNTAX**, and indicate how it should be fixed. Note: only fix things that *would cause the program to crash* in this step.

b. Additionally the code contains mistakes that make the code run incorrectly. They are not syntax errors that cause a crash, but they cause it not to run as intended. The intended goal of this program is to add up 0 + 2 + 4 + 6 + 8 (print "Getting big!" at 8) + 10 and then print x = 30, but that won't happen even after the syntax errors are fixed. Mark each semantic error with an arrow, label it as **SEMANTIC**, and indicate how it should be fixed.

Problem 6 – C Programming

Create a new project in MPLABx that uses the Simulator. Starting from template.c, create a file called "lastname_firstname.c".

In that file, write a program that will determine whether or not each value in an array is divisible by your favorite number. The program should also determine how many times your favorite number appears in that array. You should define the array of values and your favorite number using the following code:

```
#define FAVORITE_NUM 7
#define LIST_LENGTH 9


int number_list[LIST_LENGTH] = {19, 21, 7, 55, 8, 7, 22, 42, 13};
```

Your program should print to the UART1 when it finds a value in number_list that is divisible by FAVORITE_NUM. It should also print the total number of times FAVORITE_NUM appears in number_list. For your demo your window should look exactly as shown below:

```
21 is divisible by my favorite number
7 is divisible by my favorite number
7 is divisible by my favorite number
42 is divisible by my favorite number
The number list contains my favorite number 2 times
```

Recall that you learned how to **enable the UART1 window** in the day 4 lecture "Intro to MPLABx" in the "Hello World Program".

You are required to use a for loop to solve this problem and will receive no credit for hard-coded solutions. Assume that we will later test your code with a different favorite number and other array values, so make sure that your program works for other values of FAVORITE_NUM and values in number_list.

When you complete this problem and have it running, call your instructor over to check it off on the front sheet.

Regardless of whether you finish or not, submit your code into the Moodle dropbox (just the lastname_firstname.c file).