

Name _____

Class hours _____

ME430 - Mechatronics

Examination I

Sept 24th, 2019

Problem	Score	
1	/ 24	
2	/ 10	
3	/ 16	
4	/ 10	
5	/ 14	
6a	code	check off
	/ 18	/2
6b	code	check off
	/ 4	/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.

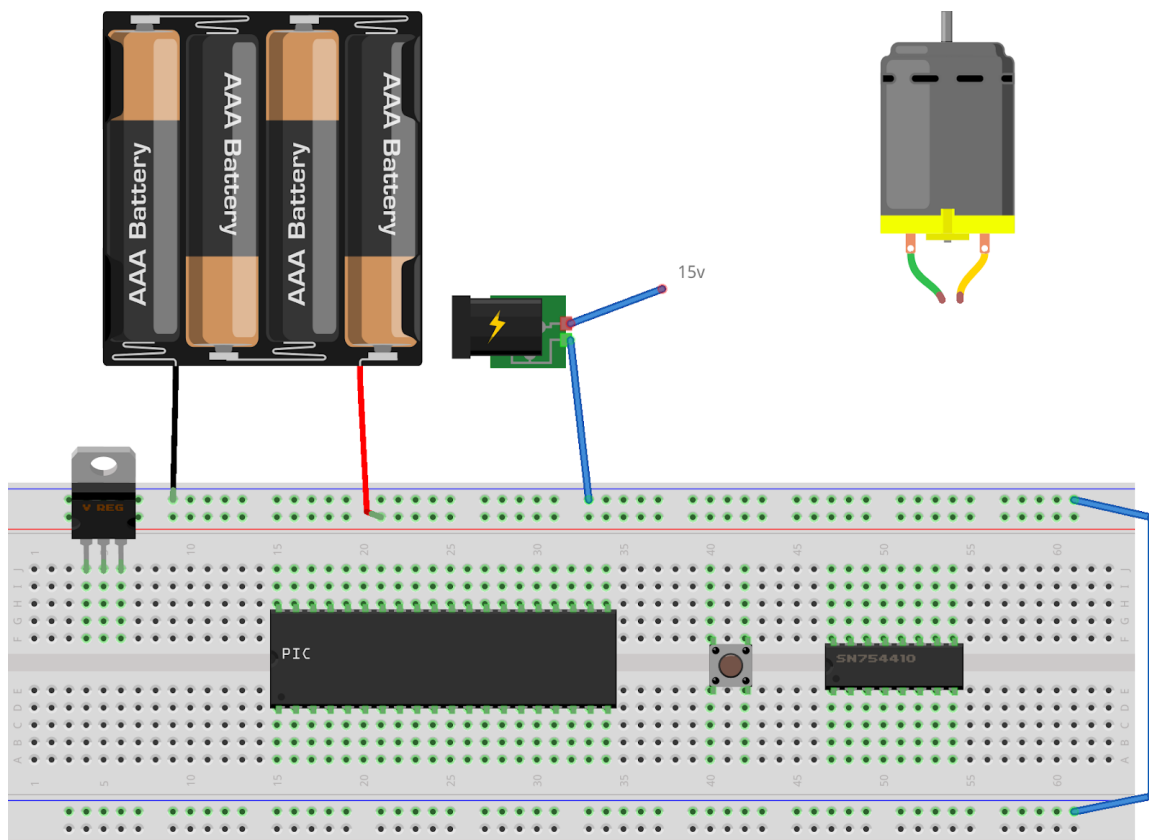
Anything not specifically allowed is prohibited. In particular, you may not use notes or code written by someone outside your lab group.

In this exam MPLAB may only be used for the final problem only!

Problem 1 - Bi-Directional DC Motor

Use the components shown below to connect a bi-directional DC motor circuit.

- Assume you want to drive the DC motor at 15v and you have a 15v power supply available (see image below).
- You also have a battery pack (roughly 6 volts) that needs to be regulated to 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 motor to perform various actions.
- Connect the PIC pins as needed to allow it to run (assume it is already programmed). Pin 1 is in the lower left corner for both chips below.
- Connect the PIC to the H-bridge as follows:
 - RD2 and RD3 are a pair and need to drive one side of the H-bridge
 - RD1 is a signal that can enable/disable the H-bridge
- Connect other H-Bridge pins as needed to drive the motor at +15v
- Add snubber diodes, decoupling capacitors, and any necessary resistors for the circuit (size all resistors and label values, sizing capacitors is not required).
- Draw lines for wires directly onto the parts below. Use a solid circle at the ends of your lines to show the connection points. Don't draw new parts (other than resistors, capacitors, and snubber diodes as needed).



Problem 2 – C debugging

Consider the code scrap shown below. Note, you may NOT use MPLAB for this problem.

```
#include <stdio.h>

5  #define MAX 5
6
7  int main() {
8      int i
9      for(0; i <= MAX; i = i + 2) {
10         total = i;
11         printf("At i = %d, the total is %d.\n", i, total);
12     }
13     printf("Average = %d.\n", total / MAX);
14 }
```

- 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 print:

```
At i = 0, the total is 0.
At i = 1, the total is 1.
At i = 2, the total is 3.
At i = 3, the total is 6.
At i = 4, the total is 10.
Average = 2.
```

However 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.

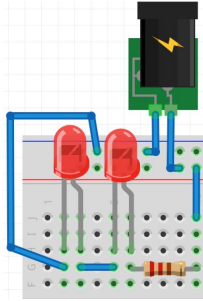
Problem 3 – Number systems and operators

Determine the output of each expression. Write the value stored as both a binary number and a 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 (but no MPLAB). Note that a char is a signed char by default.

	Value stored in binary	Decimal value if printed
char x = -16;		
int x = -16;		
char x = 110 + 110;		
char x = -100 - 100;		
char x = 50 % 5;		
char x = 7 / 4;		
int x = 8 / 10 * 5;		
char x = 8 * 5 / 10;		

Problem 4 – Resistor sizing

You have 2 LEDs with different specifications and you have decided to run them in a series circuit using a 12 volt power supply.



What size resistor do you need if the LEDs have the specifications:

	Forward Voltage Drop	Maximum current
LED 1	1.9 volts	25 mA
LED 2	2.3 volts	30 mA

You'd like to run the circuit safely and choose to target half the smaller max current to play it safe. Calculate the exact target value and then choose the nearest real resistor size from the E12 series. Then determine the color bands that you would expect to find on that resistor

Exact resistor value calculated

Real resistor value

Color bands for that real resistor (for example Yellow-Blue-Yellow):

Problem 5 – C programming

Write a small C program below to continually flash between two LED states

- State 1 - Only LEDs RC0 and RC2 are on for 1.5 seconds
- State 2 - Only LEDs RC1 and RC3 are on for 3 seconds

Use the default oscillator speed as done in Lab 3.

You can use MPLAB to help you write code if you like, but you may not run it on a green board.

```
#include <p18f4520.h>
#include <delays.h>

// Assume all the #pragma stuff is done correctly just not shown
void main (void) {
    ADCON1 = 0b00001111;    // Sets all the pins to digital
    // You write the code to do everything else that is needed...
```

```
}
```

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”.

- a) In that file, write a program that will determine the prime factors (13 and smaller) of the integers from 1 to 24. You should define an array of prime numbers using the following code:

```
int primes[] = {2, 3, 5, 7, 11, 13};
```

Your program should print to the UART1 when it finds a value in primes that divides the current number with zero remainder. For your demo your window should look exactly as shown below (with additional lines in place of the ellipsis):

```
The prime factors of 1 are:
The prime factors of 2 are: 2
The prime factors of 3 are: 3
The prime factors of 4 are: 2
The prime factors of 5 are: 5
The prime factors of 6 are: 2 3
...
The prime factors of 24 are: 2 3
```

- b) Modify your code so that it finds multiple instances of prime factors--in other words the product of the primes listed would be the tested number. For this demo, a few of the output lines become:

```
The prime factors of 4 are: 2 2
The prime factors of 5 are: 5
The prime factors of 6 are: 2 3
...
The prime factors of 24 are: 2 2 2 3
```

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.

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).