Name Rubric CM _____ Section _____

ME430 - Mechatronics
Examination I
December 16th, 2014

Problem	Score	
1	/ 18	
2	/ 14	
3	/ 10	
4	/ 26	
5	/ 12	
6	code	check off
	/ 18	/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.

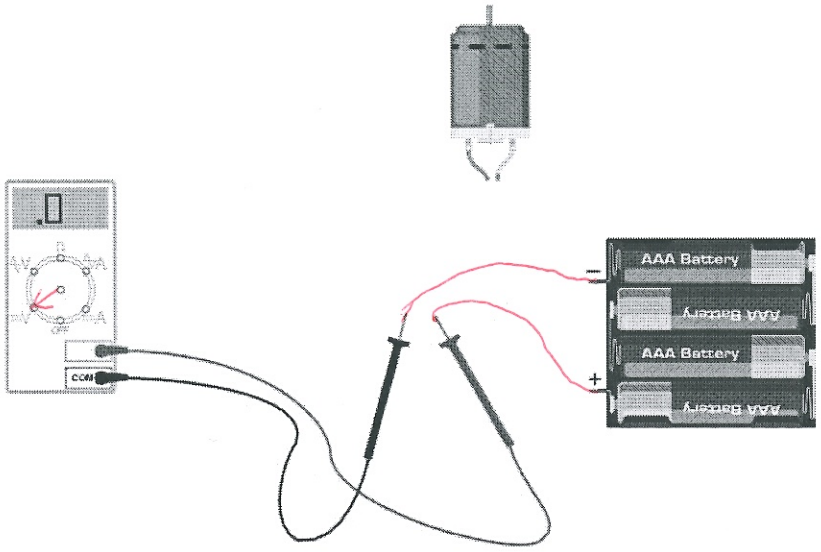
Problem 1 – Number systems

What binary value is stored for each variable?

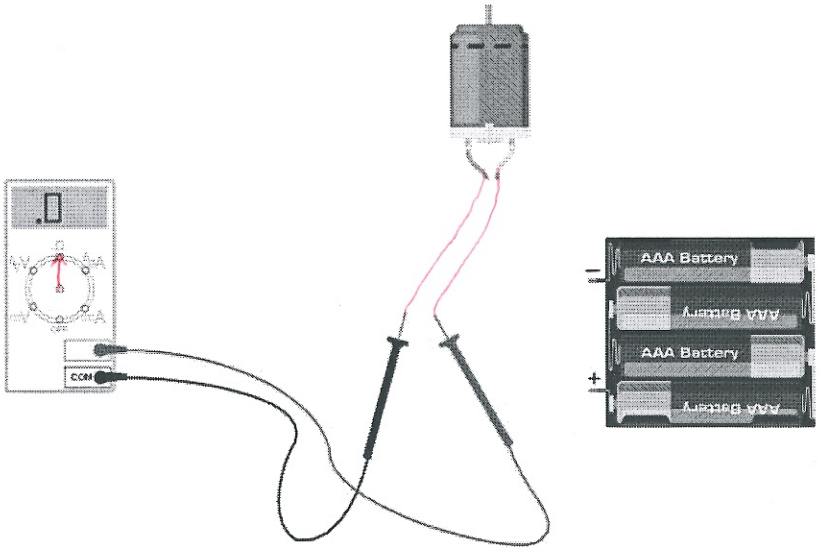
	Binary value
char **a** = 22;	0001 0110
char **b** = -101;	1001 1011
char **r2** = 0xD2;	1101 0010
char **c** = '3'; 51 → 0x33	0011 0011
char **d** = 11 ^ 2; 1011 ^ 0010 = 1001	0000 1001
char **e** = 11 / 2;	0000 0101

Problem 2 – DMM:

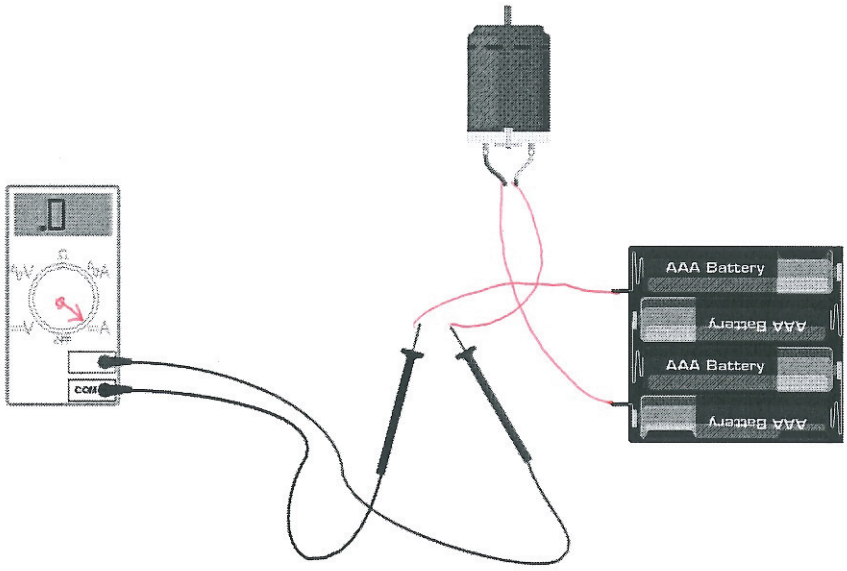
a.) On the diagram below set the knob on the DMM (draw a line from the middle of the circle to one of the outside options) and connect the probes as appropriate to measure the **voltage** of the battery pack. Make connects for a positive value to be displayed on the DMM.



b.) On the diagram below set the knob on the DMM (draw a line from the middle of the circle to one of the outside options) and connect the probes as appropriate to measure the **resistance** of the motor.

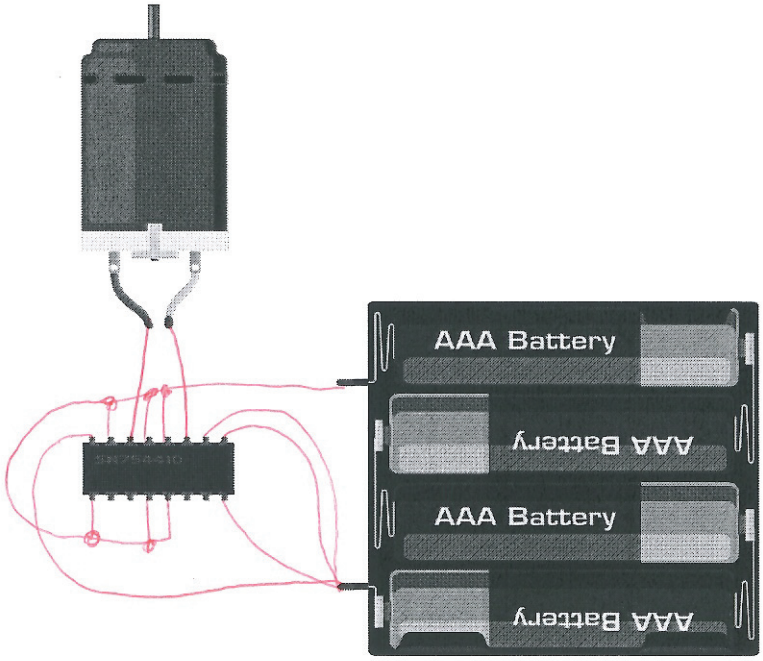


c.) On the diagram below set the knob on the DMM (draw a line from the middle of the circle to one of the outside options) and connect the probes as appropriate to measure the **current** of the motor.



Problem 3 – H-Bridge:

In an ideal world you would use regulated power on some pins of an H-Bridge and unregulated power on other pins. Additionally, in an ideal world, you would use snubbers, decoupling capacitors, and resistors on enable lines that go to power. However, for this problem you ONLY have a pack of wires/alligator clips and the three items shown below: a motor, H-Bridge, and battery pack. Connect the motor to the H-Bridge as appropriate (not just directly to the power supply), then draw all the wire connections needed make the motor "just go".

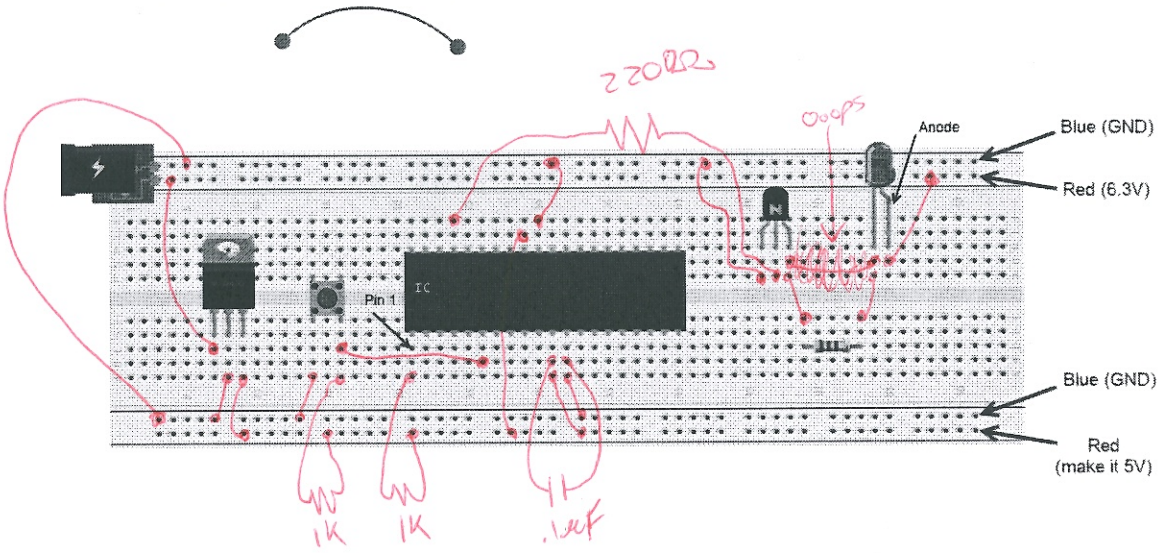


Problem 4 – BJT

Below you will find a 6.3 volt power source connected on the top rail, a voltage regulator, a pushbutton, an 18F4520 PIC, a NPN BJT, a 220 ohm resistor, and an LED.

- Connect the voltage regulator (making a 5 volt power and ground rail on bottom)
- Set up a basic switch circuit with the pushbutton, and connect the signal line to RA4.
- Make all of the connections necessary for the PIC (assume it is already programmed). Add a cap.
- Connect the LED and 220 ohm resistor such that the LED will be controlled by the BJT (note that you are required to use the 220 ohm here, instead of the normal 330 ohm we often use).
- Add 1 resistor of an appropriate, real-value size to connect RB4 to the BJT.
- Show your work and put the resistor size in the box below (use an E6 or E12 series resistor size).
- Add any additional wires to make the circuit perform.

You will need to draw a resistor and a cap, but don't redraw any other components: connect to the ones shown. For wire connections on the board draw solid circles at the connection points. For example:



$V = iR$

$6.3 - 1.7 = i(220)$

$i = \frac{4.6}{220}$

$i = .021\ A$

$\frac{i}{10} \Rightarrow 0.0021\ A$

$V = iR$

$5 = (.0021)R$

$R = 2391\Omega$

Resistor size used

2200Ω

2.2K

Problem 5 – C debugging

a. Consider the code scrap shown below. There are 3 lines that cause syntax errors that would not let this code compile. Mark the corrections on the code below.

```
/** Header Files ***************************************/
#include <p18f4520.h>
#include <stdio.h>
#include <adc.h>

/** Configuration Bits *********************************/
#pragma config OSC = INTIO67
#pragma config WDT = OFF
#pragma config LVP = OFF
#pragma config BOREN = OFF
#pragma config XINST = OFF

/** Define Constants Here ******************************/
#define PRESSED 0;
#define UNPRESSED 1;
/*********************************************************
 * Function:        void main(void)
 *********************************************************/
#pragma code
void main(void) {
    // Set the clock to 4 MHz
    OSCCONbits.IRCF2 = 1;
    OSCCONbits.IRCF1 = 1;
    OSCCONbits.IRCF0 = 0;

    // Pin IO Setup
    OpenADC(ADC_FOSC_8 & ADC_RIGHT_JUST & ADC_12_TAD,
            ADC_CH0 & ADC_INT_OFF & ADC_REF_VDD_VSS,
            0x0B); // Four analog pins
    TRISA = 0xFF; // All of PORTA input
    TRISB = 0xFF; // All of PORTB input
    TRISC = 0x00; // All of PORTC output
    TRISD = 0x00; // All of PORTD output
    PORTC = 0x00; // Turn off all 8 Port C outputs

    char value = 100;

    if (PORTBbits.RB0 == PRESSED) {
        value += 100;
    }
    if (PORTBbits.RB1 == PRESSED) {
        value += 100;
    } else if (PORTBbits.RB2 == PRESSED) {
        value += 100;
    } else if (PORTBbits.RB3 == PRESSED) {
        value += 100;
    }

    while (1) {
        // This area loops forever
    }
}
```

omit ;

omit ;

Variable declaration must be the first thing in the function.

100

200

too big for a char

300

WAY too big for a char

b. If the three lines were fixed and the code was run while buttons on RB0, RB2, and RB3 were being held down. What would be the decimal value of the char variable called **value**?

44

Problem 6 – Simple C

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 code to print out all of the multiples of 17 that occur between 1 and 200. Then print out the total number of multiples found. Setup your UART1 window to display the printed output.

Recall: You learned to enable the UART1 window in the day 4 lecture "Intro to MPLABx", "Hello World Program". (http://www.rosebotics.org/me430/unit?unit=4&lesson=2)

Your output should look like this:

```
17 is a multiple of 17.
34 is a multiple of 17.
51 is a multiple of 17.
68 is a multiple of 17.
85 is a multiple of 17.
102 is a multiple of 17.
119 is a multiple of 17.
136 is a multiple of 17.
153 is a multiple of 17.
170 is a multiple of 17.
187 is a multiple of 17.
There were 11 multiples of 17 found from 1 to 200
```

You are required to use a for loop to solve this problem and will receive no credit for hardcoded solutions. For example a number like 11 or 187 certainly may not appear in your code.

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 *.c file).

```c
#include <stdio.h>

void main(void) {
    int i;
    int totalMultiplesOf17 = 0;

    // Option 1
    for (i = 1; i < 200; ++i) {
        if (i % 17 == 0) {
            printf("%d is a multiple of 17.\n", i);
            totalMultiplesOf17++;
        }
    }

    // Option 2
//    for (i = 17; i < 200; i = i + 17) {
//        printf("%d is a multiple of 17.\n", i);
//        totalMultiplesOf17++;
//    }

    printf("There were %d multiples of 17 found from 1 to 200\n", totalMultiplesOf17);

    while (1) {
    }
}
```