

Name _____

CM _____ Section _____

ME430 - Mechatronics

Examination I

Jan. 7, 2016

Problem	Score	
1	/ 10	
2	/ 26	
3A	/ 10	
3B	/ 12	
3C	/ 6	
4	/ 12	
5	code	check off
	/ 22	/2
Total	/100	

You may use only:

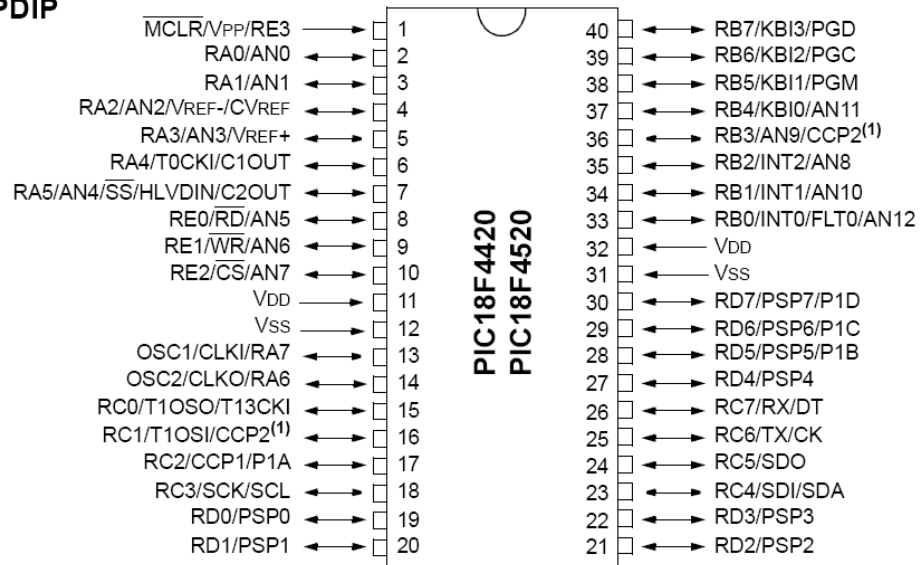
- Your computer
- Any paper notes (such as notes on the videos) you brought to the exam, so long as those notes were written by you or your lab partner.
- Any electronic notes or code residing on your local (C:) hard drive, so long as those notes/code were written by you or your lab partner.
- The course 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

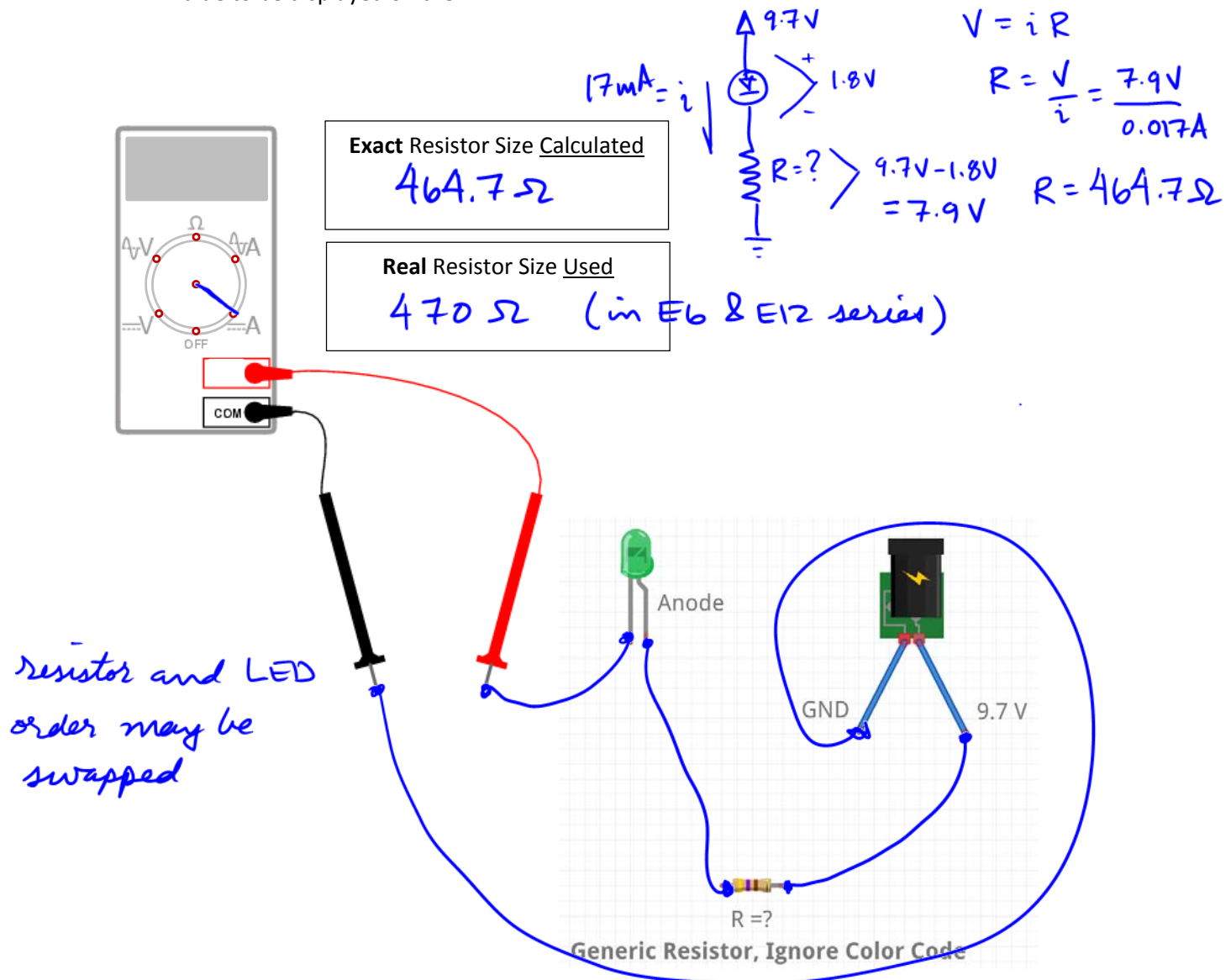
This diagram can also be found on the course website. It is copied here as a reference.

40-pin PDIP



Problem 1 – Resistor Sizing and DMM (10 pts)

- a.) Compute the resistor size to limit the current through the green LED in the figure below to 17 mA if the LED has a nominal forward voltage drop of 1.8 V and it is powered by the supply in the figure. Show your work and put the **exact** resistor size you calculated for the LED circuit **AND** your **real** resistor size used in the boxes below (use an E6 or E12 series resistor size).
- 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 and components as appropriate to measure the **current** through the LED. Make connections for a positive value to be displayed on the DMM.

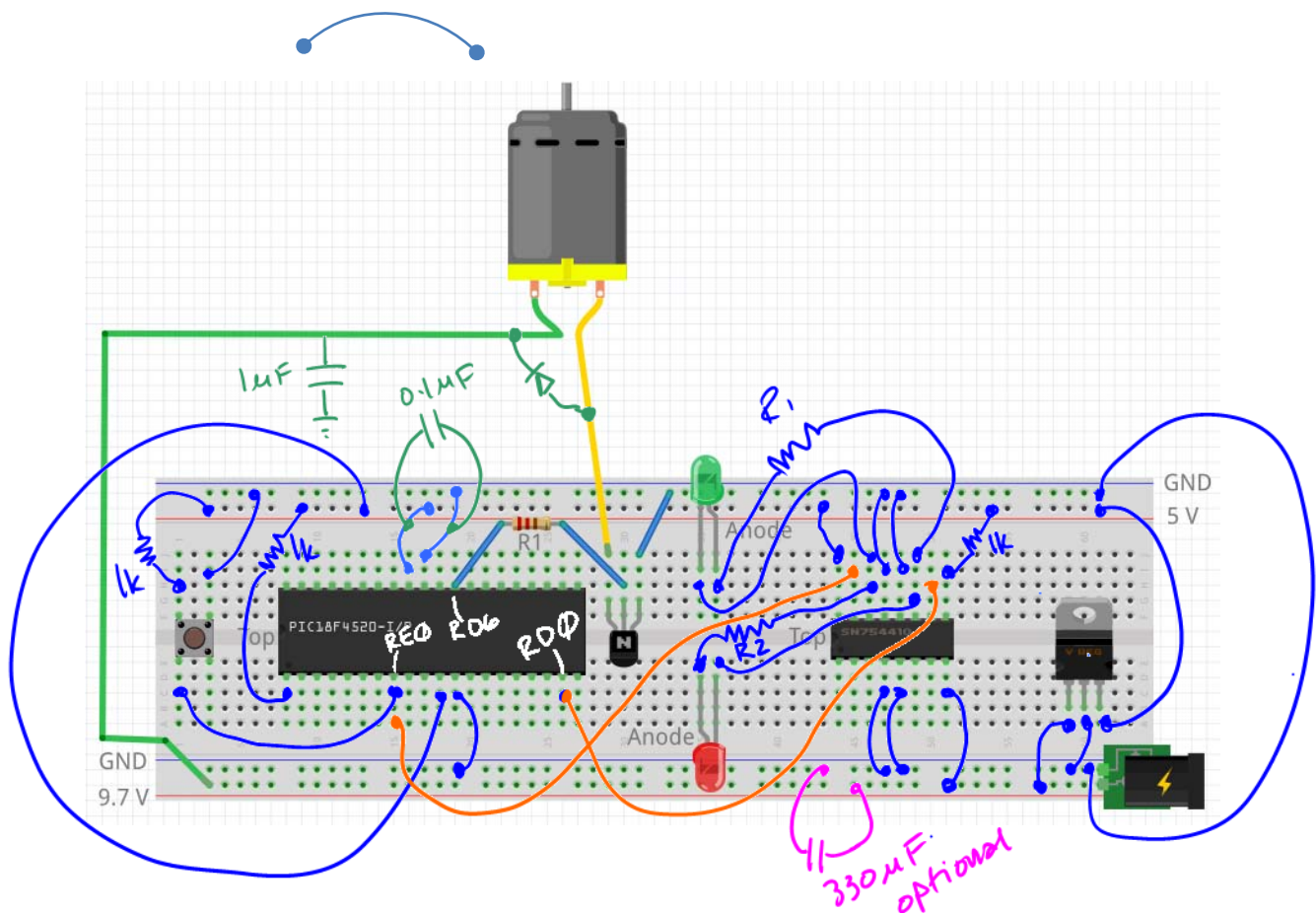


Problem 2 - BJT Motor and H-Bridge for LEDs (26 pts)

On the breadboard below you will find a 9.7 volt power source connected on the bottom rail, a voltage regulator, a pushbutton, an 18F4520 PIC, a NPN BJT, an H-bridge, and two LEDs (green and red). The brushed DC motor has been wired for unidirectional operation so that it is controlled by the BJT off of unregulated power. The BJT is controlled by pin RD6 on the PIC, which will be controlled by the pushbutton on RE0.

- Connect the voltage regulator (making a 5 volt power and ground rail on top)
- Set up a basic switch circuit with the pushbutton, and connect the signal line to RE0.
- Make all of the connections necessary for the PIC (assume it is already programmed).
- Connect the LEDs to one H-bridge circuit. These lights will be used to indicate whether the motor is on or off. (Red light lit means the motor is off, Green light lit means the motor is on.) Run the LEDs off unregulated power. The lights will be controlled by the pushbutton on RE0 and the inversion of the button signal output on RD0. Make all connections necessary for H-bridge.
- Add snubber diodes and decoupling capacitors as needed to have a well-protected circuit – you will need to draw these components in.
- Add any additional wires to make the circuit perform.

Do not redraw any components in the diagram below - connect the given components. Do draw in any required resistors, capacitors, or diodes. Specify resistor and capacitor sizes when known, otherwise label them with an appropriate variable, such as R_1 . Do not compute any resistor or capacitor sizes for this problem. For wire connections on the board draw solid circles at the connection points. For example:



Problem 3B – Operators I (12 pts)

For this question, assume that **a**, **b**, **c**, and **d** are all signed chars. If **a**=23 and **b**=2, what will be the result of the following operations? If there is an overflow, just state it is an overflow.
(Express your answers in decimal or note that the result is an overflow.)

$c = a \% 10;$ $c = 3$
 $c = a / 10;$ $c = 2$
 $c = a * 3 / 4;$ $c = (23 * 3) / 4 = 69 / 4 = 17$
 $c = a * 6 / 8;$ **overflow** ($c = -14$)
 $c = a \& b;$ $= 0b00010111 \& 0b00000010 = 0b00000010 = 2$
 $d = a \& \& b;$ $= 1$ (TRUE b/c 'a' and 'b' are both non-zero which is TRUE, so TRUE & TRUE = TRUE)

Problem 3C – ASCII (6 pts)

For this question, assume that **a**, **b**, **c**, **d**, and **e** are all signed chars.

Ascii
 $a = '9';$ 57
 $b = '1';$ 49
 $c = a - b;$ $c = 57 - 49 = 8$
 $d = b + 1;$
 $e = a + 1;$ $e = 57 + 1 = 58$

What will the value of **c** be in decimal?

$c = 8$

What will the value of **e** be in decimal and ASCII?

\downarrow
 58

Problem 4 – C debugging (12 pts)

- a. Download the code "broken_code.c" from the course website. You may create a project and test the code if you wish. Important lines from the code are reproduced below. There are two problems that cause errors that prevent the code from compiling. Fix those problems and mark the corrections on the code below. (Line numbers given match line numbers in "broken_code.c".)

```

12  #include <stdio.h>
23  #define b = 9

```

```

34  void main(){
35      int a = 3;
36      int c = 12;
37

```

```

38      if (c > b > a)
39      {
40          b = b + a;
41          a++;
42      }

```

```

43      printf("a = %4d\n", a);
44      printf("b = %4d\n", b);

```

```

45
46      for (m=20; m>16; m--)
47      {

```

```

48          a++;
49          printf("a = %4d\n", a);
50      }
51  }

```

- b. If those problems are fixed, what will print when we run the code?

a = 3

b = 9

a = 4

a = 5

a = 6

a = 7

m

a @ printf

20

4

19

5

18

6

17

7

16

x

doesn't reach printf

- c. Does line 40 execute and change the value of **b**, again assuming the compiling problems are fixed? If so, what change(s) need to be made if the programmer intended for line 40 to execute? Mark those changes on the code above. If we make those changes, what will print when we run the code?

No. Line 40 does not execute as written.

need if condition to be

if ((c > b) && (b > a))

output

a = 4

b = 12

a = 5

a = 6

a = 7

a = 8

Problem 5 – C Programming (22 pts + 2 pts = 24 pts)

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

Your main objective is to find all the combinations of integers A, B, which form the legs of a right triangle such that the hypotenuse is also an integer. You should loop through the integers $1 \leq a \leq 16$, $1 \leq b \leq 16$, and print to the UART1 window each time a successful combination is found.

Your output should look like this:

```
A= 3, B= 4, forms a right triangle.  
A= 4, B= 3, forms a right triangle.  
A= 5, B= 12, forms a right triangle.  
... (Several more pairs will be found)
```

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>)

Big .c programming hint: since we want to deal exclusively with integer math, you should test whether

$$\left(\sqrt{(a)^2 + (b)^2}\right)^2 \text{ is equal to } ((a)^2 + (b)^2)$$

The .c 'sqrt' function contained in the math.h library returns a double precision float. Assign the square root to another int variable before squaring the left hand side. The comparison above should detect whether any round-off occurred during type casting.

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

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

----- C-Code -----

```

/** Header Files *****/
#include <p18f4520.h>
#include <stdio.h>
#include <math.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 SAMPLE 100

/** Global Variables *****/
int a, b, c, test1, test2a, test2b, success;

/* Function: void main(void)
*****/
#pragma code

void main(void) {
    for (a = 1; a < 17; a++) {
        for (b = 1; b < 17; b++) {
            test1 = a * a + b*b;
            test2a = sqrt(test1);
            test2b = test2a*test2a;
            success = test2b == test1;
            if (success) {
                printf("A= %4d, B= %4d, forms a right triangle.\n", a, b);
            }
        }
    }
    while (1) {
        // This area loops forever
    }
}

```

----- UART I/O Window Output -----

```

A=    5, B=   12, forms a right triangle.
A=    6, B=    8, forms a right triangle.
A=    8, B=    6, forms a right triangle.
A=    8, B=   15, forms a right triangle.
A=    9, B=   12, forms a right triangle.
A=   12, B=    5, forms a right triangle.
A=   12, B=    9, forms a right triangle.
A=   12, B=   16, forms a right triangle.
A=   15, B=    8, forms a right triangle.
A=   16, B=   12, forms a right triangle.

```