

Name Key

CM _____ Section _____

ME430 - Mechatronics

Examination I

Sept 22nd, 2016

Problem	Score	
1	/ 27	
2	/ 19	
3	/ 16	
4	/ 14	
5	code	check off
	/ 22	/2
Total	/100	

You may use only:

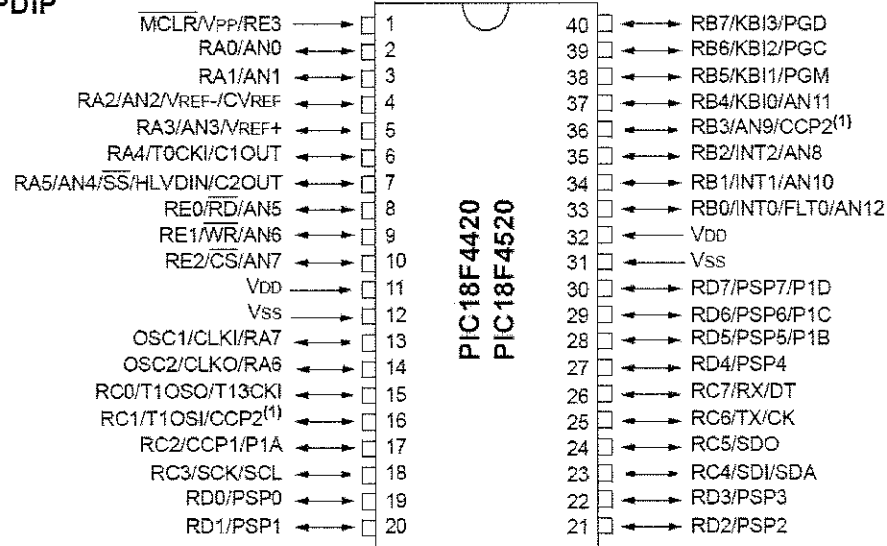
- Your computer
- Any paper notes (such as notes on the videos) you brought to the exam, so long as those notes were written by you or your lab partner.
- Any electronic notes or code residing on your local (C:) hard drive, so long as those notes/code were written by you or your lab partner.
- The course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

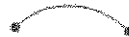
For Labs 1 and 2 you used a programmed PIC. Assume that the PIC on the next page is programmed the same way.

Draw the wiring connections necessary so that a button press will cause the BJT to drive the motor just to its right, cause the MOSFET to turn on the LED, and cause the Darlington to drive the motor just to its right.

Additionally, show the wiring connections for the PIC and basic switch. Hint: Since the same switch is being used to control 3 different output circuits, you should really branch the switch's input to the PIC to go to three different inputs pins, so that you can use 3 separate output pins (helps keep each output isolated).

You should show when a resistor, *required* snubber diode, or capacitor is necessary by adding its (schematic) symbol to the diagram. **However, you do not need to size any resistors or capacitors for this problem.**

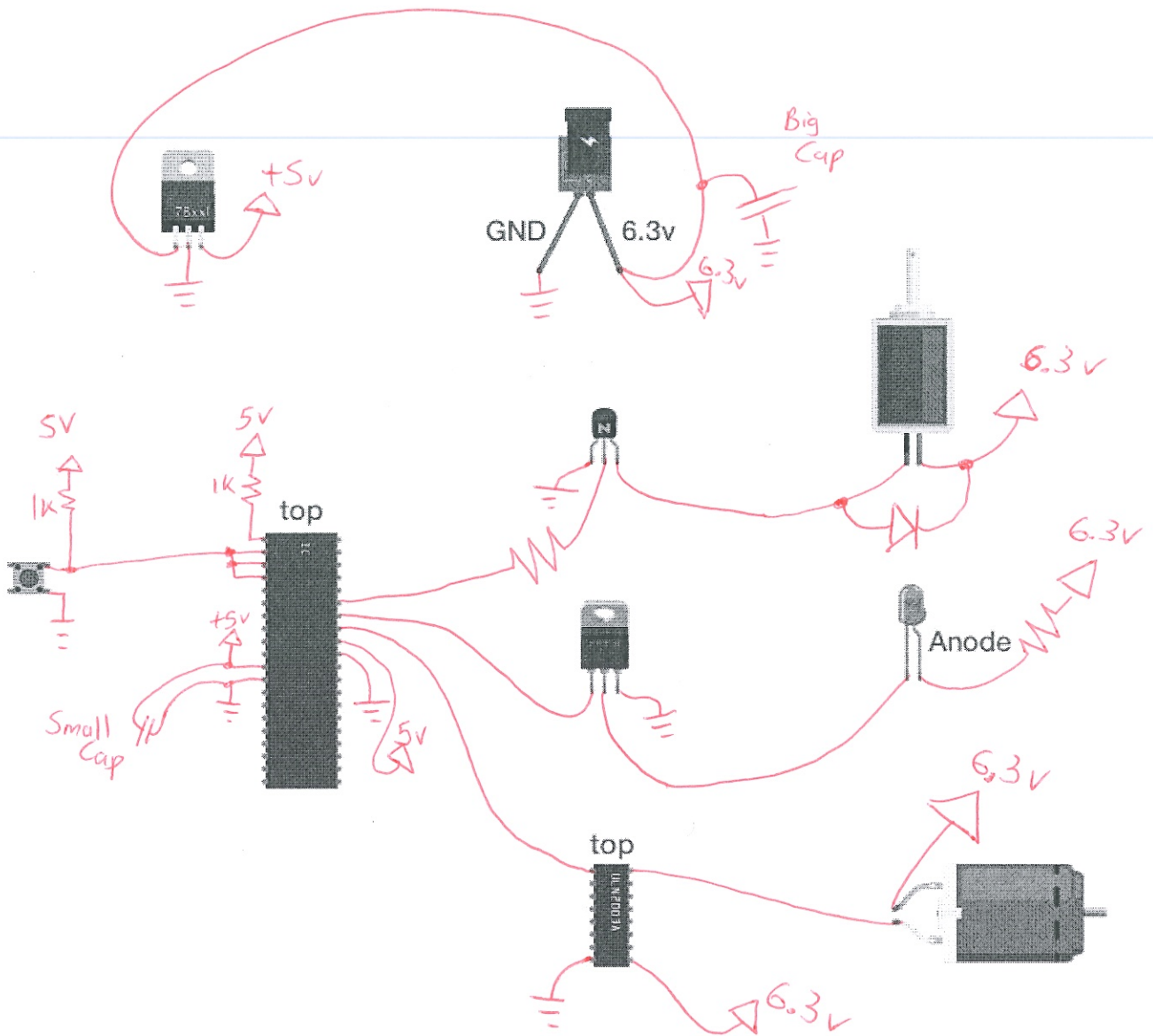
For parts already shown on the diagram, make connections using lines to the parts.



To keep your drawing neater feel free to use ground symbols, and voltage supply symbols. For the voltage symbols always label the value OR say regulated / unregulated.



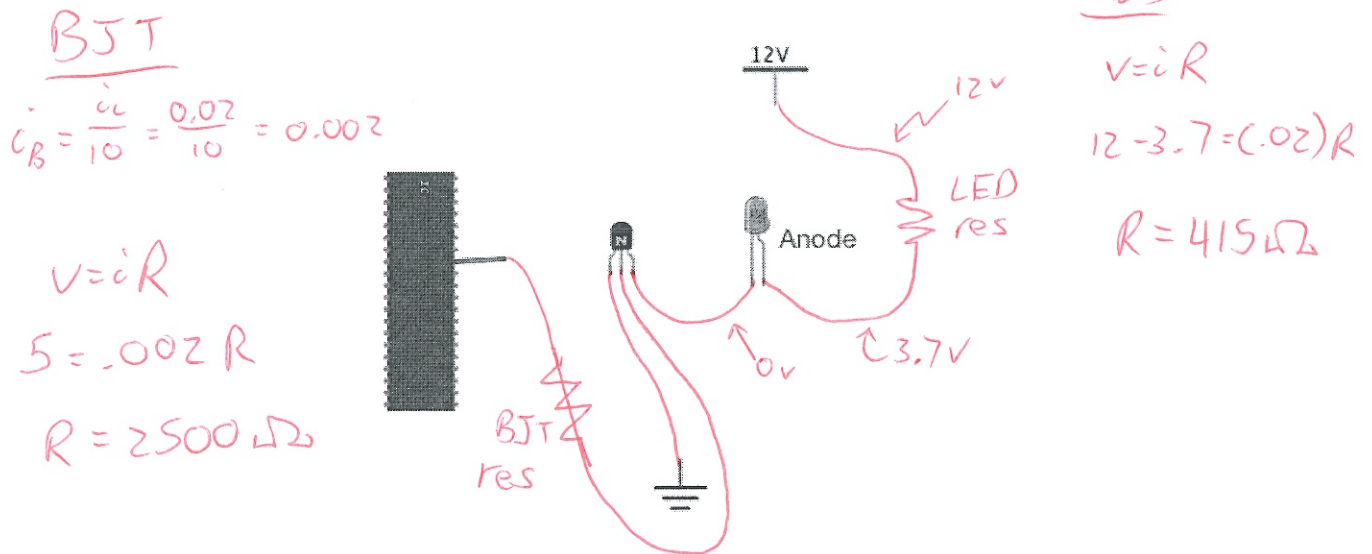
Additionally the top of the next page shows a voltage regulator and a power supply. Wire those up correctly and label those with ground symbols and labelled voltage supply symbols as well.



Problem 2 – Resistor sizing

For this problem you have an LED running on a 12 volt power supply, as shown below. The LED has a forward voltage drop of 3.7 volts and a **maximum** continuous current rating of 40 mA. We will control this LED with a BJT, and the BJT is controlled by the PIC. What **exact** size should the resistors for the BJT and the LED be, if we want to run the LED at the **HALF** of it's maximum possible current? What size should the resistors be if we must choose a **real resistor** from the E12 series of resistors? (Hint: Day 2, in class.)

Draw the connections below, adding resistor symbols when needed (don't worry about PIC connections other than the wire shown), put answers in the boxes, and show all work for your calculations.



Exact BJT resistor value calculated

2500 Ω

Exact LED resistor value calculated

415 Ω

Real BJT resistor value

2.7 K

Real LED resistor value

390 Ω

Problem 3 – Number systems

Suppose you had a 12 bit signed integer variable type, let's call it a jumbo. What binary bits are stored in memory for the a, b, and c **jumbo** variables?

jumbo a = 43;

0	0	0	0	0	0	1	0	1	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---

jumbo b = -12;

1	1	1	1	1	1	1	1	0	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---

jumbo c = 50000;

0	0	1	1	0	1	0	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

If c was printed, what value would print?

Value of c:

848

What are the min and max sizes for a **jumbo**?

signed jumbo min value

-2048

signed jumbo max value

2047

Problem 4 – C debugging

Consider the code scrap shown below.

```
#include <stdio.h>

void main(void) {
    printf("Exam 1\n");
    int x = 5;
    int y = 20;
    for (i = 0; i < 5; i = i + 2) {
        x += 5;
        if (x > 15) {
            y++;
        }
    }

    print("x = %d\n", x);
    print("y = %d\n", y);

    while (1) {
    }
}
```

Handwritten annotations:

- Move AFTER var declaration* (with an arrow pointing to the `for` loop header)
- declare i* (with an arrow pointing to the `i` in the `for` loop header)
- f* (circled around the `print` function name)
- ;* (circled around the semicolon at the end of the `print` statements)

- a. This code contains syntax errors and won't compile. Circle each mistake and indicate how it should be fixed. Modify only issues that will cause the code not to run-- do not change the functionality. (If you wish, you may download the full version of this code from <http://tinyurl.com/exam1-broken-code-c> and work with it on your computer. However, that is optional, there is no code turn-in on this problem.)
- b. If the errors were fixed what would the code print?

Output:

Handwritten output:

x = 20

y = 21

Problem 5 – 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 determines how many times values in an array increase as you move through the array from beginning to end. For example if you had the array:

```
#define NUM_VALUES 10
int values[NUM_VALUES] = {0, 1, 20, 20, -5, -6, 20, 19, 18, 0};
```

There would be 3 times where the values increased:

1. 0 → 1
2. 1 → 20
3. -6 → 20.

Additionally determine how many times the values decreased. In the example above there were 5 decreases.

1. 20 → -5
2. -5 → -6
3. 20 → 19
4. 19 → 18
5. 18 → 0

Print out the number of increases and decreases to the UART1 window as shown below.

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>)

For the array shown, your program should output:

There were 3 increases and 5 decreases.

Your printf statement should use this template to generate the output shown above.
There were %d increases and %d decreases.\n

You are required to use a for loop to solve this problem and will receive no credit for hardcoded solutions. Additionally your program must work for any array size (2 or larger) and any int values. You can assume that NUM_VALUES will match the size of the array, but you should make no assumptions about the contents of the array.

To help save you typing in the array visit: <http://tinyurl.com/exam1-starting-code-c>

In order to check off this problem make an array like the one shown, but assume that we will later test your code with other arrays.

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).

```
#include <stdio.h>

#define NUM_VALUES 10
int values[NUM_VALUES] = {0, 1, 20, 20, -5, -6, 20, 19, 18, 0};

void main(void) {
    int i, increaseCount = 0, decreaseCount = 0;

    printf("\n\nExam 1 - C Programming problem\n");

    for (i = 0; i < NUM_VALUES - 1; i++) {
        if (values[i] < values[i + 1]) {
            increaseCount++;
        } else if (values[i] > values[i + 1]) {
            decreaseCount++;
        }
    }

    printf("There were %d increases and %d decreases.\n", increaseCount, decreaseCount);

    while(1) {

    }
}
```