Name P. CUNNINGHAM (KEY)   CM ________   Section _____

ME430 - Mechatronics

Examination I – Written Portion

September 27, 2011

Problem	Score
1	/ 15
2	/ 12
3	/ 15
4	/ 8
Total	/ 50

For the written portion of the exam, you may use only:

- 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.
- A pencil/pen.
- A calculator (optional).

For the computer portion of the exam, 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).
- ANGEL 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 – Variable types:

Convert the following hex numbers to Binary:
0x7C; 0x0A; 0x09; 0x7F; 0x36

$0x7C \rightarrow \boxed{0111\ 1100} \rightarrow 2^6+2^5+2^4+2^3+2^2$

$= 64+32+16+8+4 = \boxed{124}$

$0x0A \rightarrow \boxed{0000\ 1010} \rightarrow 2^3+2^1 = \boxed{10}$

$0x09 \rightarrow \boxed{0000\ 1001} \rightarrow \boxed{9}$

To Decimal:

$0x7F \rightarrow \boxed{0111\ 1111} \rightarrow 2^6+2^5+2^4+2^3+2^2+2^1+2^0$

$= 64+32+16+8+4+2+1 = \boxed{127}$

$0x36 \rightarrow \boxed{0011\ 0110} \rightarrow 2^5+2^4+2^2+2^1$

$= 32+16+4+2 = \boxed{54}$

What is the 'smallest' variable type that will store one of these numbers?

(unsigned) char → 8 bits

Write a line of c code that stores the five numbers in an array using the least amount of memory.

(unsigned) char myArray[] = {0x7c, 0x0A, 0x09, 0x7F, 0x36};

Problem 2 – Operators

The following global variables have been defined

```
char a, b, c, d;
```

→ signed char −128 to +127

Based on these definitions, report the results (in decimal form) of the following operations. Show your work for full credit.

```
a = -7 * 20;
```

= −140

overflow

−140 → '−' 1000 1100

↓ undo 2's comp.

0111 0100

$= 2^6 + 2^5 + 2^4 + 2^2$

$= 64 + 32 + 16 + 4 = \boxed{116}$

```
b = 32 % 9;
```

$= \boxed{5}$

= 4 R 5

```
c = 35 | 13;
```

(0010 0011) | (0000 1101)

= (0010 1111)

$= 2^5 + 2^3 + 2^2 + 2^1 + 2^0$

$= 32 + 8 + 4 + 2 + 1 = \boxed{47}$

```
d = 13 << 3;
```

← 3spaces

13 → 000(0 110 1000)

$= 2^6 + 2^5 + 2^3$

$= 64 + 32 + 8$

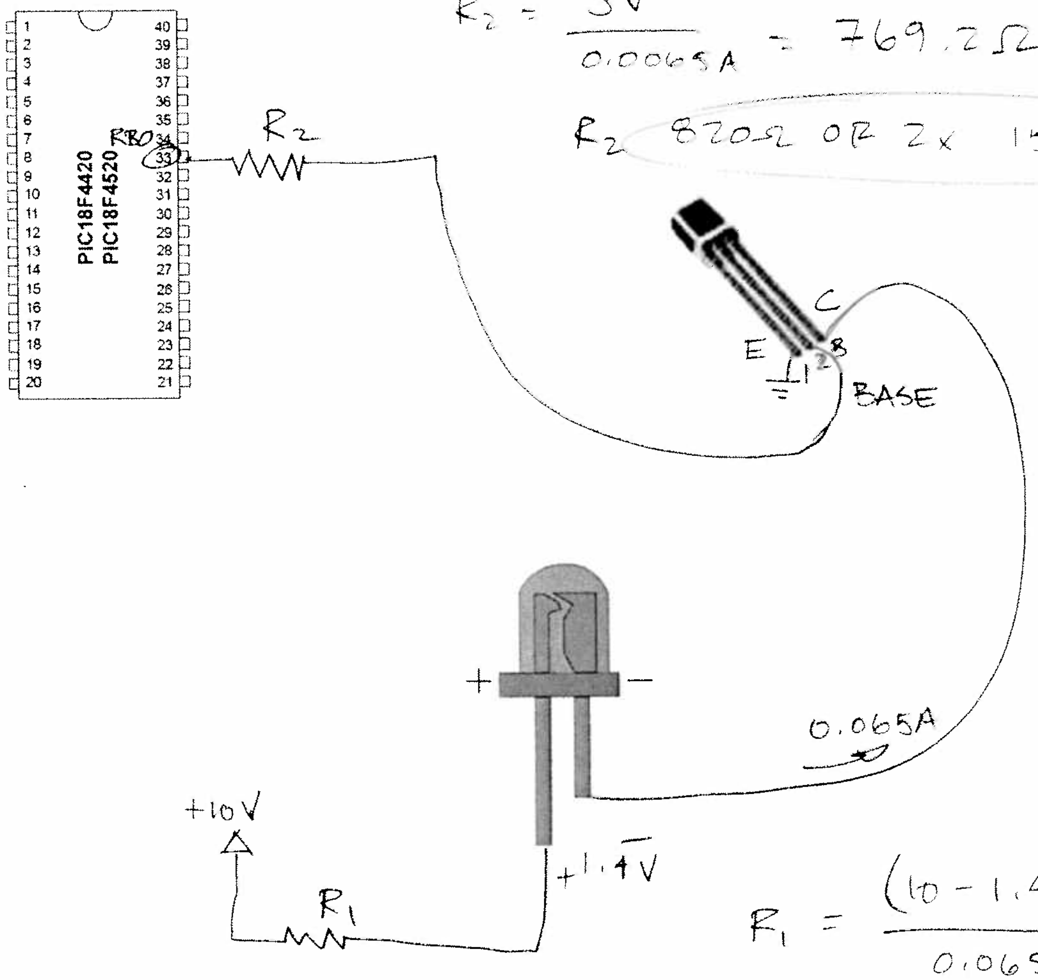
$= \boxed{104}$

Problem 3 – Resistor sizing BJT (NPN) & LED:

Assume you have a 10 volt supply and we want to use that power supply to run an infrared LED (emitter) at 65 mA. The LED has a 1.4 volt forward voltage drop. We would like to use a PIC's RBO pin to drive an (NPN) BJT to run the LED.

$$R_2 = \frac{5\,V}{0.0065\,A} = 769.2\,\Omega$$

R_2 820Ω OR 2x 1500Ω



$$R_1 = \frac{(10 - 1.4)\,V}{0.065\,A} = 132.3\,\Omega$$

R_1 120Ω OR 2x 270Ω in ||

On the diagram above:

- Draw in the circuit for the transistor running the LED using the 10 V power supply.
- Show the calculations to size any resistors that are needed for the circuit. Choose standard E12 series (10%) size resistors and label them on the diagram.

x2

- Connect RB0 on the PIC to the transistor circuit. You do NOT need to show other connections on the PIC (power, ground, etc.)
- Make connections to the physical devices shown, don't redraw a schematic symbol for a BJT or an LED.

Problem 4 – Clock Frequency and Instruction Cycles

Write the lines of code needed to set up the PIC internal oscillator to run at 125 kHz.

```
# pragma config OSC = INT067

OSCCONbits.IRCF2 = 0;
OSCCONbits.IRCF1 = 0;
OSCCONbits.IRCF0 = 1;
```

How long would it take to complete 78 instruction cycles using this clock?

clock freq = 125 kHz

$\downarrow \div 4$

instr. cycle freq = 31.25 kHz

$\downarrow \frac{1}{x}$

instr. cycle period = 32 μsec

$$\left(78 \text{ instr. cycles}\right)\left(\frac{32\ \mu\text{sec}}{\text{instr. cycle}}\right)$$

$$\boxed{\begin{gathered} = 2496\ \mu\text{sec} \\ \text{or} \\ 0.0025 \text{ sec} \end{gathered}}$$

Name P. CUNNINGHAM (KEY) CM ________ Section _____

ME430 - Mechatronics

Examination I – Computer Portion

September 27, 2011

Problem	Score
5	/ 10
6	/ 20
7	/ 25
Total	/ 55

For the computer portion of the exam, 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).
- ANGEL 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 5 -- Fix this code:

Suppose that we have downloaded our standard template.c file, included <stdio.h> properly, and created the main routine shown below. Assume that idx and result are global char variables appropriately defined before the code below appears.

```
void main (void)
{
        ADCON1 = 0x0F;
        TRISB = 0x~~0C~~;  → 0x00;   to set all as outputs
        PORTB = 0;

        idx = 1;
        result = 0;
        while(idx<=5)  {
            idx = idx + 1;
            result = result + idx;
            printf("result = %d \n", result);   3   omitted
            PORTB = result;
        }
        while(1) {
        }
  3  omitted
```

move idx increment line after result is computed.

When we run this code, we want to set the Port B output pins to the binary representation of result and to print:

```
result = 1
result = 3
result = 6
result = 10
result = 15
```

However, in this code there are two syntax errors (code that will cause the program not to build), one programming logic error (code that will cause you to not get the output you want), and one hardware setup issue (code that will cause the hardware to not behave as you expect). Change the code so that it prints the five lines and sets Port B correctly, and clearly label the four errors in the code.

Problem 6 -- For loops and printing:

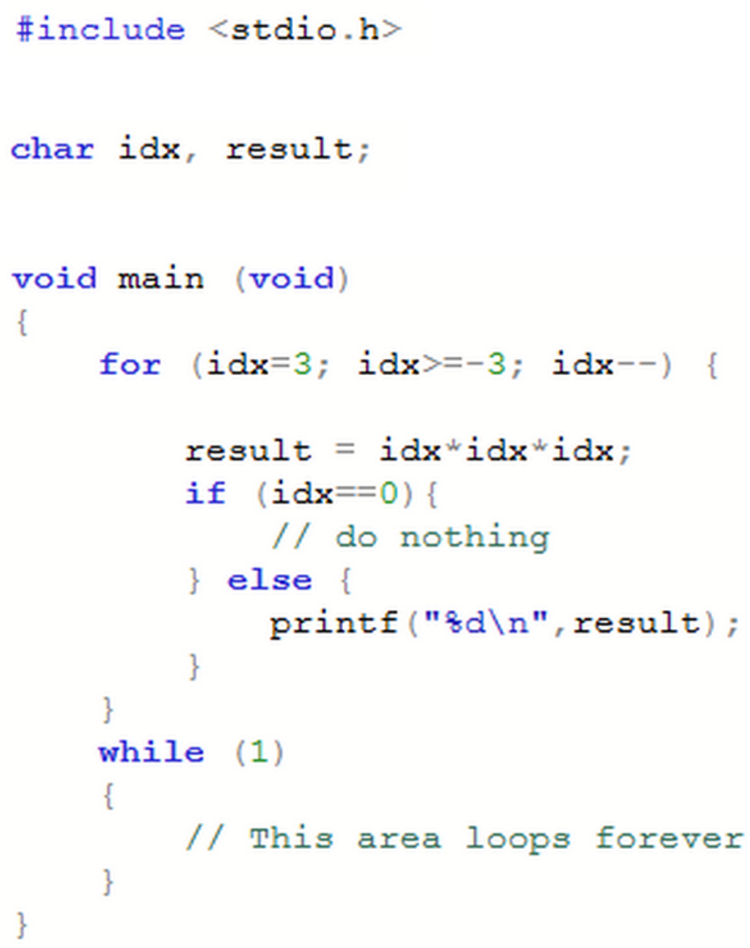
Starting from template.c, write a piece of code that *uses a for loop* to print out the numbers 27, 8, 1, -1, -8, and -27 using a formula (not hardcoding in those numbers). Print each value on a separate line. That is, the output of the code (to the Uart1 IO window) should be:

```
27
8
1
-1
-8
-27
```

For full credit on this problem, you must use the index of the for loop with an appropriate formula to accomplish this task (i.e. you can't just type in four different print statements).

Rename your template.c file to **problemXXX_yourName.c**

Put your finished code (just the *.c file) in the Angel dropbox.



```c
#include <stdio.h>


char idx, result;


void main (void)
{
    for (idx=3; idx>=-3; idx--) {

        result = idx*idx*idx;
        if (idx==0){
            // do nothing
        } else {
            printf("%d\n",result);
        }
    }
    while (1)
    {
        // This area loops forever
    }
}
```

Problem 7 – Wiring a stepper motor and PIC:

On the next page you will be drawing the connections necessary to drive a stepper motor with a PIC that someone else has programmed.

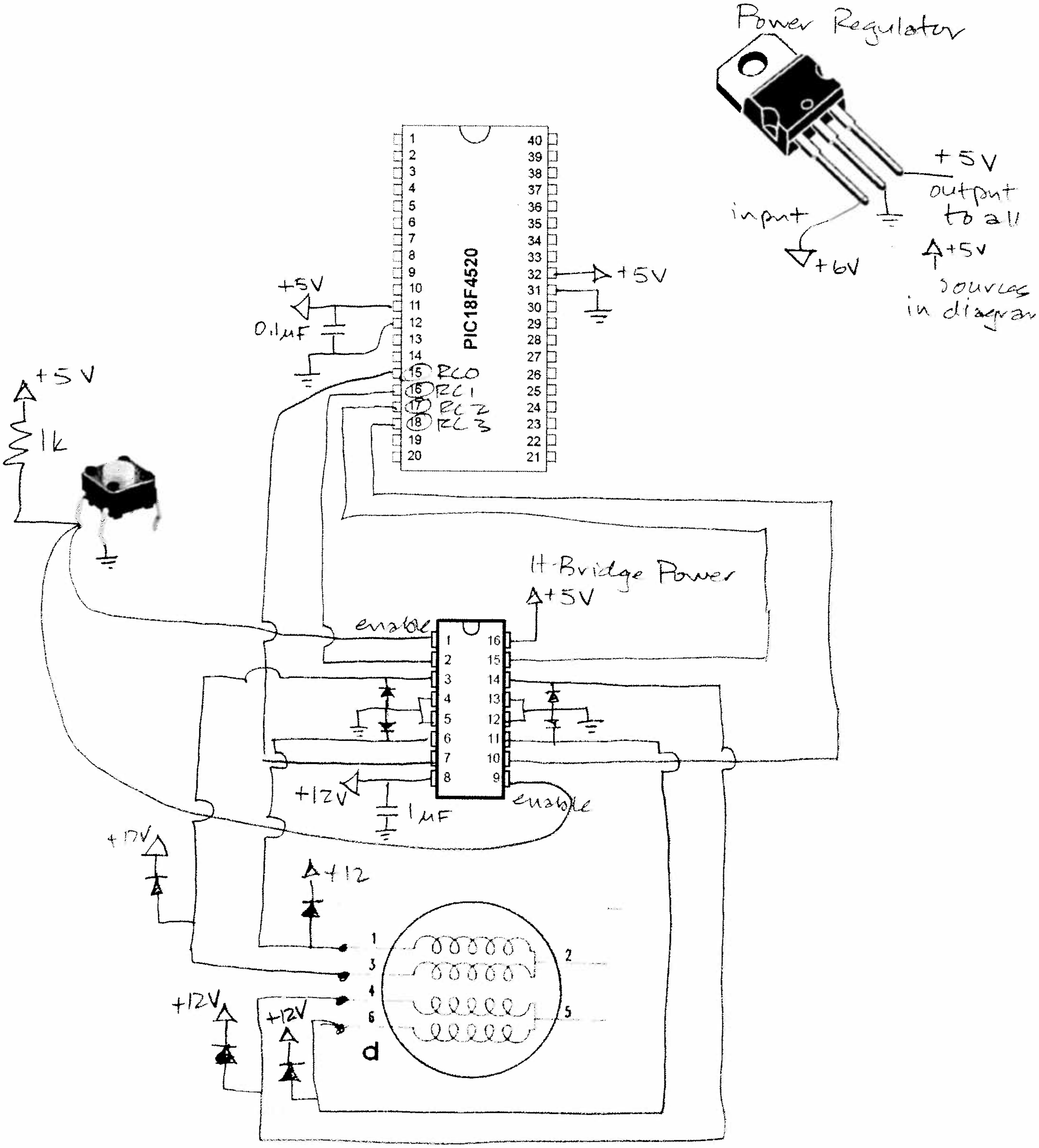
Assume that the PIC is programmed with the following code snippet:

```
void CounterClockWise() {
                                     RC3  RC2  RC1  RC0
//     BLACK = 1; BROWN = 0; ORANGE = 0; YELLOW = 1;
       PORTC = 0b00001001;
       DelayTime(HalfSecond);
//     BLACK = 0; BROWN = 1; ORANGE = 0; YELLOW = 1;
       PORTC = 0b00000101;
       DelayTime(HalfSecond);
//     BLACK = 0; BROWN = 1; ORANGE = 1; YELLOW = 0;
       PORTC = 0b00000110;
       DelayTime(HalfSecond);
//     BLACK = 1; BROWN = 0; ORANGE = 1; YELLOW = 0;
       PORTC = 0b00001010;
       DelayTime(HalfSecond);
}
```

Make sure to accomplish the following when you add your connections on the next page :

- Draw a 12 V power source and an unregulated 6 V power source. Hook up the voltage regulator chip to the 6V source to create a 5 V regulated line.
- Connect the H-bridge to the stepper motor as we did in lab.
- Connect the PIC and the H-bridge to power and ground. The motor should run at 12 V.
- Connect the PIC to H-bridge (a PIC pinout will be helpful).
- Add a pushbutton/basic switch circuit which is connected to the enable pin of the H-Bridge so that when the pushbutton is pressed the stepper motor stops.
- Add appropriate forms of inductive kick protection to the circuit, including snubber diodes and decoupling capacitors.
- Label all resistor and capacitor sizes. Use real resistor sizes from the E12 (10%) series.

You must connect to the physical devices already shown (i.e. don't ignore the pictures and just draw schematic symbols for them).


Power Regulator
+5V
output to all
input
+6V
+5v
sources in diagram
PIC18F4520
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
+5V
+5V
0.1μF
RC0
RC1
RC2
RC3
+5 V
1k
H-Bridge Power
+5V
enable
1
2
3
4
5
6
7
8
16
15
14
13
12
11
10
9
+12V
1μF
enable
+17V
+12
1
3
4
6
2
5
+12V
+12V
d