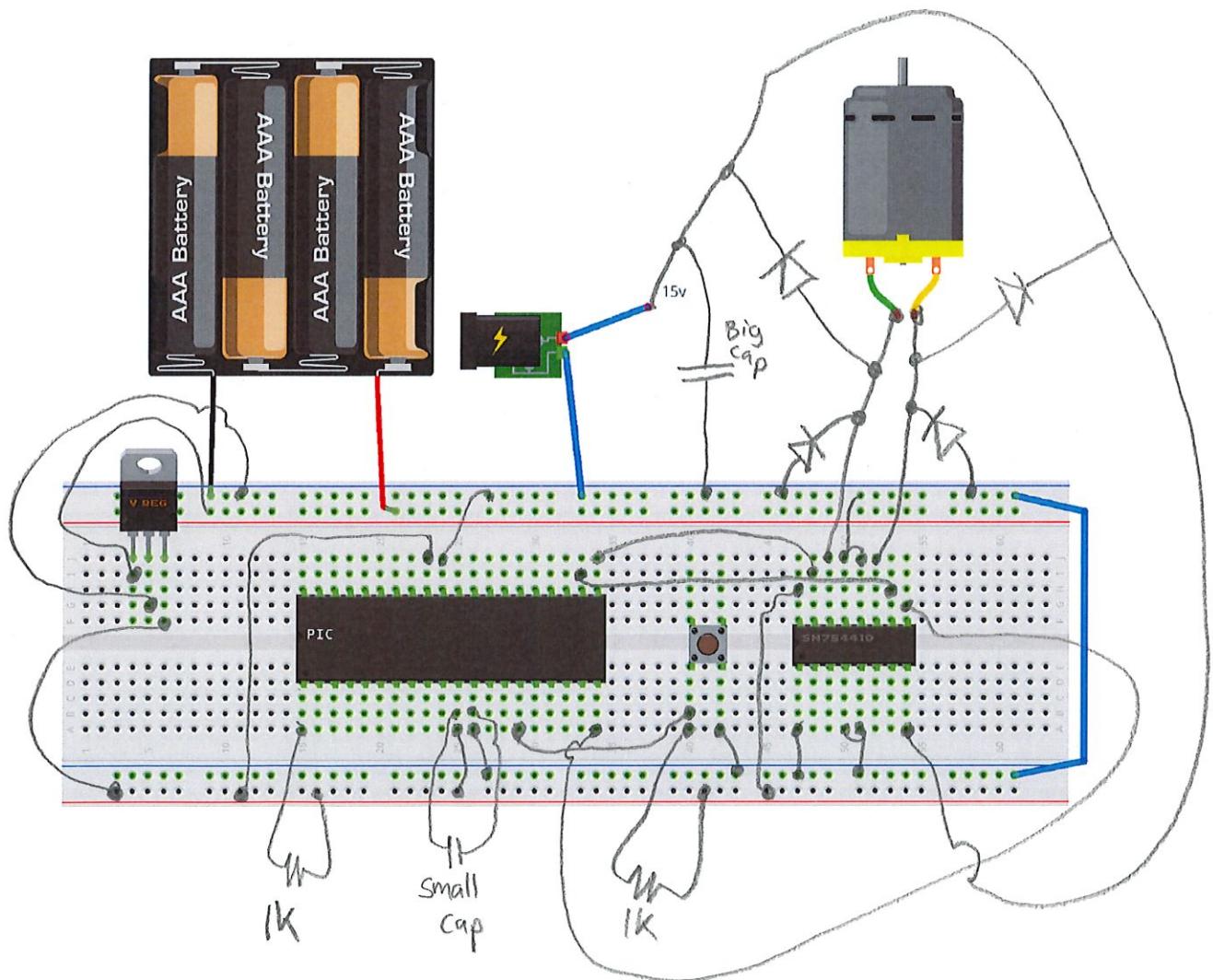


### Problem 1 - Bi-Directional DC Motor

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 an unregulated 15v power supply available (see image below).
- Assume you also have a battery pack 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 wire directly onto the parts below, don't draw new parts.



**Problem 2 – C debugging**

Consider the code scrap shown below.

```

SYNTAX
1  #include <stdio.h>
2  #define MAX 5
3
4  int main() {
5      int i
6      for(0; i <= MAX; i = i + 2) {
7          total = i;
8          printf("At i = %d, the total is %d.\n", i, total);
9      }
10     printf("Average = %d.\n", total / MAX);
11 }
12
13
14

```

Handwritten annotations:

- SYNTAX** (written vertically on the left)
- SYNTAX** (written above line 5, pointing to `total = 0;`)
- SYNTAX** (written above line 8, pointing to `int i`)
- SYNTAX** (written above line 9, pointing to `for(0; i <= MAX; i = i + 2)`)
- SYNTAX** (written above line 10, pointing to `total = i;`)
- SYNTAX** (written above line 11, pointing to `printf("At i = %d, the total is %d.\n", i, total);`)
- SYNTAX** (written above line 12, pointing to `}`)
- SYNTAX** (written above line 13, pointing to `printf("Average = %d.\n", total / MAX);`)
- SYNTAX** (written above line 14, pointing to `}`)
- SYNTAX** (written above line 15, pointing to `i = 0;`)
- SYNTAX** (written above line 16, pointing to `i = i + 2`)
- SYNTAX** (written above line 17, pointing to `total = total + i;`)
- SYNTAX** (written above line 18, pointing to `total += i;`)
- SYNTAX** (written above line 19, pointing to `total = total + i;`)
- SYNTAX** (written above line 20, pointing to `total += i;`)
- SYNTAX** (written above line 21, pointing to `total = total + i;`)
- SYNTAX** (written above line 22, pointing to `total += i;`)
- SYNTAX** (written above line 23, pointing to `total = total + i;`)
- SYNTAX** (written above line 24, pointing to `total += i;`)
- SYNTAX** (written above line 25, pointing to `total = total + i;`)
- SYNTAX** (written above line 26, pointing to `total += i;`)
- SYNTAX** (written above line 27, pointing to `total = total + i;`)
- SYNTAX** (written above line 28, pointing to `total += i;`)
- SYNTAX** (written above line 29, pointing to `total = total + i;`)
- SYNTAX** (written above line 30, pointing to `total += i;`)
- SYNTAX** (written above line 31, pointing to `total = total + i;`)
- SYNTAX** (written above line 32, pointing to `total += i;`)
- SYNTAX** (written above line 33, pointing to `total = total + i;`)
- SYNTAX** (written above line 34, pointing to `total += i;`)
- SYNTAX** (written above line 35, pointing to `total = total + i;`)
- SYNTAX** (written above line 36, pointing to `total += i;`)
- SYNTAX** (written above line 37, pointing to `total = total + i;`)
- SYNTAX** (written above line 38, pointing to `total += i;`)
- SYNTAX** (written above line 39, pointing to `total = total + i;`)
- SYNTAX** (written above line 40, pointing to `total += i;`)
- SYNTAX** (written above line 41, pointing to `total = total + i;`)
- SYNTAX** (written above line 42, pointing to `total += i;`)
- SYNTAX** (written above line 43, pointing to `total = total + i;`)
- SYNTAX** (written above line 44, pointing to `total += i;`)
- SYNTAX** (written above line 45, pointing to `total = total + i;`)
- SYNTAX** (written above line 46, pointing to `total += i;`)
- SYNTAX** (written above line 47, pointing to `total = total + i;`)
- SYNTAX** (written above line 48, pointing to `total += i;`)
- SYNTAX** (written above line 49, pointing to `total = total + i;`)
- SYNTAX** (written above line 50, pointing to `total += i;`)
- SYNTAX** (written above line 51, pointing to `total = total + i;`)
- SYNTAX** (written above line 52, pointing to `total += i;`)
- SYNTAX** (written above line 53, pointing to `total = total + i;`)
- SYNTAX** (written above line 54, pointing to `total += i;`)
- SYNTAX** (written above line 55, pointing to `total = total + i;`)
- SYNTAX** (written above line 56, pointing to `total += i;`)
- SYNTAX** (written above line 57, pointing to `total = total + i;`)
- SYNTAX** (written above line 58, pointing to `total += i;`)
- SYNTAX** (written above line 59, pointing to `total = total + i;`)
- SYNTAX** (written above line 60, pointing to `total += i;`)
- SYNTAX** (written above line 61, pointing to `total = total + i;`)
- SYNTAX** (written above line 62, pointing to `total += i;`)
- SYNTAX** (written above line 63, pointing to `total = total + i;`)
- SYNTAX** (written above line 64, pointing to `total += i;`)
- SYNTAX** (written above line 65, pointing to `total = total + i;`)
- SYNTAX** (written above line 66, pointing to `total += i;`)
- SYNTAX** (written above line 67, pointing to `total = total + i;`)
- SYNTAX** (written above line 68, pointing to `total += i;`)
- SYNTAX** (written above line 69, pointing to `total = total + i;`)
- SYNTAX** (written above line 70, pointing to `total += i;`)
- SYNTAX** (written above line 71, pointing to `total = total + i;`)
- SYNTAX** (written above line 72, pointing to `total += i;`)
- SYNTAX** (written above line 73, pointing to `total = total + i;`)
- SYNTAX** (written above line 74, pointing to `total += i;`)
- SYNTAX** (written above line 75, pointing to `total = total + i;`)
- SYNTAX** (written above line 76, pointing to `total += i;`)
- SYNTAX** (written above line 77, pointing to `total = total + i;`)
- SYNTAX** (written above line 78, pointing to `total += i;`)
- SYNTAX** (written above line 79, pointing to `total = total + i;`)
- SYNTAX** (written above line 80, pointing to `total += i;`)
- SYNTAX** (written above line 81, pointing to `total = total + i;`)
- SYNTAX** (written above line 82, pointing to `total += i;`)
- SYNTAX** (written above line 83, pointing to `total = total + i;`)
- SYNTAX** (written above line 84, pointing to `total += i;`)
- SYNTAX** (written above line 85, pointing to `total = total + i;`)
- SYNTAX** (written above line 86, pointing to `total += i;`)
- SYNTAX** (written above line 87, pointing to `total = total + i;`)
- SYNTAX** (written above line 88, pointing to `total += i;`)
- SYNTAX** (written above line 89, pointing to `total = total + i;`)
- SYNTAX** (written above line 90, pointing to `total += i;`)
- SYNTAX** (written above line 91, pointing to `total = total + i;`)
- SYNTAX** (written above line 92, pointing to `total += i;`)
- SYNTAX** (written above line 93, pointing to `total = total + i;`)
- SYNTAX** (written above line 94, pointing to `total += i;`)
- SYNTAX** (written above line 95, pointing to `total = total + i;`)
- SYNTAX** (written above line 96, pointing to `total += i;`)
- SYNTAX** (written above line 97, pointing to `total = total + i;`)
- SYNTAX** (written above line 98, pointing to `total += i;`)
- SYNTAX** (written above line 99, pointing to `total = total + i;`)
- SYNTAX** (written above line 100, pointing to `total += i;`)

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).

	Value stored in binary	Decimal value if printed
a char x = -16;	1111 0000	-16
b int x = -16;	1111 1111 1111 0000	-16
c char x = 110 + 110;	1101 1100	-36
d char x = -100 - 100;	0011 1000	56
e char x = 50 % 5;	0000 0000	0
f char x = 7 / 4;	0000 0001	1
g int x = 8 / 10 * 5;	0000 0000	0
h char x = 8 * 5 / 10;	0000 0100	4

0010 1110  
32 15 47

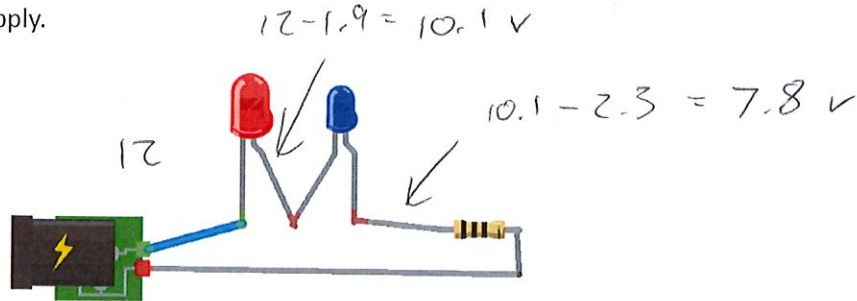
1101 1100

0010 0100

0001 1001 0000  
1110 0111 0000 ← -200

**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

$$V = IR$$

$$7.8 = \frac{0.025}{2} R$$

$$R = 624 \Omega$$

$$\begin{array}{l} 560 \Delta 64 \\ 624 \nearrow \\ 680 \Delta 56 \checkmark \end{array}$$

Exact resistor value calculated

$$624 \Omega$$

Real resistor value

$$680 \Omega$$

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

Blue - Gray - Brown