a). (10 pts) Determine exactly how many clock cycles it takes to subtract two numbers from each other for types int8\_t, int16\_t, and int32\_t (give me three different answers). When you are doing these subtractions, make sure that the numbers are large enough so that the compiler doesn’t optimize the results and you may also need to use the results in other parts of your code so the compiler doesn’t optimize it out. Include the relevant lines of C code you used and the assembly the compiler generated for each type (make sure to also include loads and stores). Hint: you should lookup how many clock cycles each assembly operation takes. I would also recommend indicating in the assembly code how long each instruction takes.

8 bit

One lds

1 sub

1 sts

16 bit

2 lds

2 sub

2 sts

32 bit

4 lds

4 sub

4 sts

b). (5 pts) How many microseconds does each subtraction take for each of the three types (uint8\_t, uint16\_t, and uint32\_t)?

8 bit time

0

16 bit time

4

32 bit time

4

c)

8 bit time

8

result8

0

16 bit time

16

result16

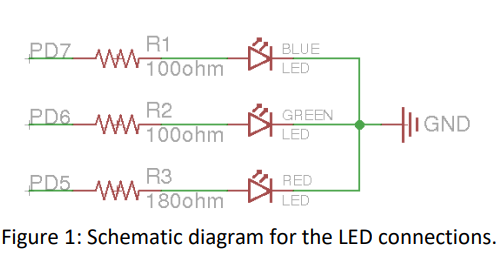
3

32 bit time

40

result32

0

a). (5 pts) In Lab 1, Figure 1, the circuit for connecting the LEDs is shown. The LEDs “drop” a fixed voltage and the resistors serve to limit the amount of current that flows through the LEDs. Assume the drops for each of these are RED 2.2V, GREEN 3.0V, and BLUE 3.0V1 . How much current flows through each of the LEDs? Note that the RED LED will have a different current than the other two. 

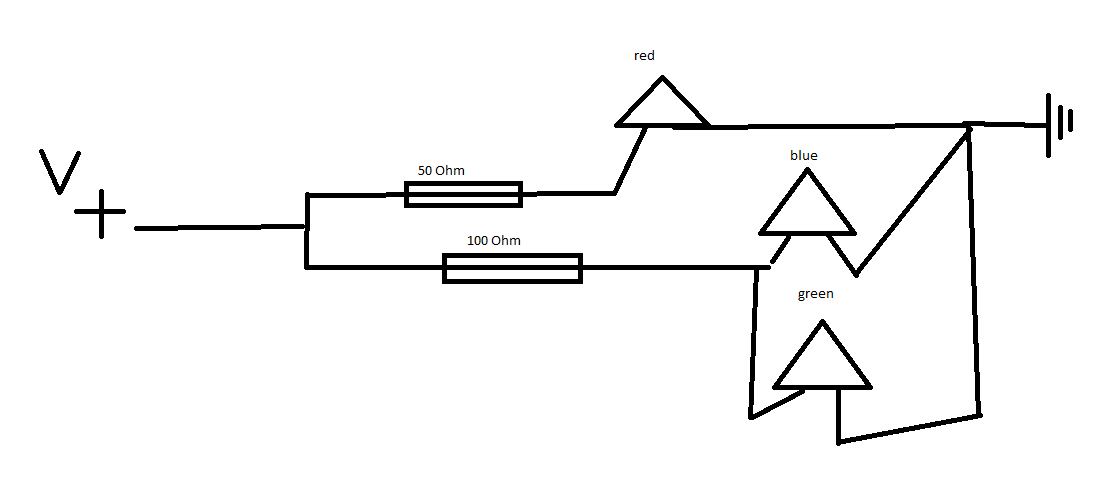
red: 15.55 mA

green/ blue : 20 mA

b). (5 pts) How many LEDs could be connected and controlled by a single pin on the Arduino (hint: look at the Atmel datasheet).

2

c). (5 pts) Draw a schematic showing how you would the three LEDs (RGB) to a single output pin while keeping the brightness of each individual LED the same as in the original case (assuming in this case that the pin can supply sufficient current).



Problem 4. Input a). (10 pts) In Lab 1, you configured the button by using a pullup resistor. However, as you learned in class, the input pins also have the ability to activate an internal pullup resistor. Write the C code (setting registers DDRx, etc) below to activate the internal pullup resistor of pin PC2. Also give the “Arduino” version using functions we discussed in class to set pins. Instructor sign off required: You should also implement this and show the instructor the functionality of using your button in Problem 5 without the resistor

Problem 5. For this problem you should complete the sections in morse.c where STUDENT CODE is indicated. To do this problem, you will need to include morse.c and morse.h in your sketch (use the menu Sketch->Add File. To call functions from morse.c, you will need to put #include "morse.h" in your main sketch file (note that it should be in quotes, not in < >).

Complete the morse.c code so you will be able to send Morse code blinks. You should make sure that you are able to specify any of LED\_RED, LED\_GREEN, or LED\_BLUE as the LED to output blinks or any combination of them (e.g. LED\_BLUE | LED\_RED). I would recommend creating helper functions that turn on or off LEDs so if you switch the pin that controls the LEDs, you only have to change code in one or two places. All of the code described here should be in a single program that runs at the same time.

You must also turn in your code on Canvas. Failing to electronically turn in your code will result in a 10 point penalty on this assignment. Points may also be deducted for no header, coding errors, poor style, or poor commenting. Note, also need to turn in a printout of your code for this assignment.

a). (5 pts) In morse.c, the Morse blink pattern (dots and dashes) for each character are stored in a single byte. Read the code and describe how this is done and what the meaning of each bit is.

b). (10 pts) Now write the C code to turn on and off the LEDs by setting the registers (e.g. DDRx, etc.). What pins did you connect the LEDs to? How did you configure these pins as output? How do you turn the LED on and off? Make sure to include the relevant code here.