CSCE 336 Embedded Systems, Spring 2023
Homework 2

Due: Thursday, Feb 16th, 2023 (start of class)

Instructions: This homework is an individual assignment, collaboration is not allowed. Show your work and describe your reasoning to get partial credit if your solution is incorrect. Unless otherwise specified, assume problems refer to the Arduino board we are using. This assignment is out of 100 points, but is equally weighted with other homework assignments.

Name: ________________________________

Problem 1. *(5 pts) (To be completed at end of assignment) Approximately how much time did the total assignment take? Which problem took longest and how much time did it take? List any resources you used to complete this assignment (e.g. websites, datasheets, office hours, discussions with others, etc.). Be specific and if you did not use any other resources include the statement "I did not use outside resources for this assignment."*

Problem 2. Arduino Setup and Programming

Instructor sign off: *Before the start of class on the due date, you must show proof of the functional code running on your Arduino for this part of the assignment and upload it to Canvas. You can take a short video for the instructor or TAs.* ***Plan ahead*** *and email the instructor or TA if you have issues, but do not leave this until the last moment. Failing to complete this portion of the homework before the due date will result in a zero for the programming portion of this assignment.*

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 copy of your code on Canvas for this assignment.*

Problem 2a. *(10 pts) For this problem, you should write a program that will monitor the keyboard from the* ***Serial Monitor*** *(located in the tools menu) and will blink one time if you send it a '1', two if you send a '2', and so on up to '9'. Further, you should create a flag such that if you send an 's' then all blinks after will be slow blinks (perhaps 1 second on, 1 second off) and if you send a 'f' all blinks after will be fast (quarter second on/off or thereabouts). To get started with this, look at the example code (the communication section is a good place to start) that comes with the Arduino sketch environment and their online help. Do not make the code any more complex than it needs to be (e.g. do not use serialEvent()), but also remember that there should be appropriate comments in your code. Additional info can be found about the Serial Library on Arduino Website or the Serial.read() function. For the Instructor Sign off please upload a link to a short video to Canvas.*

Instructor sign off: ______________

Problem 2b. *(5 pts) What is the voltage the processor on the Arduino runs at? What is the power it is using if the current is 25mA?*

Problem 2c. *(5 pts) What is the maximum clock speed for the ATmega328 used on the Arduino if there is no external oscillator?*

Problem 3. Operation Timing

It is often useful to time how long a particular operation takes on an embedded system to ensure proper functionality. Most operations happen very quickly, but by executing the same operation many times (by putting it in a loop) you can determine how long an operation takes by simply using an LED and stopwatch. The problem is that the loop itself will incur some overhead. To further complicate this problem on an 8-bit processor, such as our Atmel, this overhead will be dependent on whether the loop counter is 8-, 16-, or 32-bits.

*In this section, you should determine how long each iteration of a for loop takes when using an 8-, 16-, and 32-bit unsigned integer as the loop counter (**uint8_t**, **uint16_t**, and **uint32_t**, respectively). To do this, turn on an LED before entering the for loop and then turn it off once the for loop is complete. The compiler will optimize out any code that does not do anything, so we need to add an instruction that prevents this. You can do this by adding an assembly command (which the compiler will not remove) that does nothing. The easiest is to use the "nop" (no operation) command. To do this in c do (for the 8-bit version):*

```
//Declare this globally to prevent compiler optimizations
//This is especially important for 32 bit types
uint8_t cntr8 = 0xff;
for(cntr8 = 0; cntr8 < 255; cntr8++){
      asm volatile("nop");
}
```

You should then experiment until you come up with values that make the LED stay on for somewhere between 5 and 15 seconds. This is sufficiently long that you can get a good estimate of how long each iteration takes by timing this overall time with a stopwatch. Note that for the 8-bit and 16-bit variables you may need to nest loops to delay for a sufficiently long period. In this case you can ignore any overhead that the nesting may cause. In addition to enabling you to time operation overhead, this will also give you an alternative, albeit less accurate, way to delay for a specific period of time as opposed to using the Arduino delay(...) function.

Note: The compiler likes to optimize variables. To avoid problems associated with this, declare your counters globally and assign them values that can only be represented with the specified type. For instance, for 32 bit counters, declare them as uint32_t cntr32 = 0xffffffff;. Also, if you get the same values for 16- and 32-bit counters it is likely a compiler optimization causing issues.

Problem 3a. (10 pts) Approximately how many clock cycles does an 8-bit, 16-bit, and 32-bit loop iteration as described above take? How did you determine this? Hint: you can do this good enough with just a stop watch and LED as described above. Otherwise you can use the millis() function.

Problem 3b. *(10 pts) Show a program that turns on the LED for 6 second using 32-bit loops, blinks once (off-on-off), then turns on the LED for 3 second using a 16-bit loop, blinks twice, and finally turns on the LED for 2 seconds using a 8-bit loop. Get this program signed off by showing the TA or instructor during office hours.*

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 copy of your code on Canvas for this assignment. For the Instructor Sign off please upload a link to a short video to Canvas.*

Instructor sign off: _______________

Problem 4. Digital I/O*. Refer to the Arduino Uno R3 schematic posted on the course website for this question. For these questions, make sure to give C code where appropriate.*

Problem 4a. *(5 pts) On the Arduino, how would you configure pin 3 on the output headers to be an output and how would determine if the pin is set* ***`HIGH`****? For this question, use the Arduino commands.*

Problem 4b. *(5 pts) Which pin on the Atmel does this pin correspond to (tell me the port and pin number)?*

Problem 4c. *(5 pts) Which registers (there are three) control this pin and how would you set this pin to an input and determine if it is set to high? By register, I mean those you find in the datasheet (e.g.* ***`PORTD`****).*

Problem 4d. *(5 pts)* How would you set this pin to an output and set it low using the registers?

Problem 5. Resistors

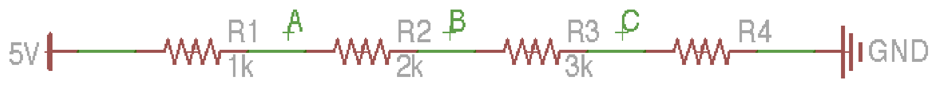

Figure 1: Schematic diagram for the LED connections.

Problem 5a. *(5 pts) If the current flowing through R3 is 0.50mA, what is the value of R4?*

Problem 5b. *(5 pts) In the schematic above (and the specified current), what is the voltage at points A, B, and C?*

Problem 5c. *(5 pts) What is the equivalent series resistance of all of these resistors (including R4)? What is the equivalent resistance if they were all placed in parallel instead?*

Problem 6. ATmega328 Datasheet

Problem 6a). *(5 pts) What memory address is the* ***`PORTC`*** *register located at (it is the higher value of those listed in the datasheet)?*

Problem 6b). *(5 pts) How would you set (give the C code) bits 3 and 4 in the* ***`DDRC`*** *register to bits 6 and 7 in the variable* ***`var`****? Make sure you set these bits at the same time and do so in a single line of code (e.g.* ***`DDRC=`****...).*

Problem 6c). *(5 pts) Which assembly instructions take the most clock cycles to complete? How many clock cycles do they take?*

Problem 6d). *(5 pts) What is the end memory address for the Internal SRAM on the Atmel on the Arduino? Where did you find this in the datasheet?*

Do not forget to fill in the amount of time you spent on this assignment and resources you used in Question 1.