CSCE 236 Embedded Systems, Fall 2017
Homework 1

Started: Tues, Aug 22nd
Checkoff Due Before Class: Aug 29th
Due in Class: Tues, Sept 5th

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. *(4 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 (*Note you only need the Arduino for this problem, you will not need a breadboard or any other components until the next part.*)*

Instructor sign off: *Before the start of class on the due date, you must show the* **instructor** *the functional code running on your Arduino for this part of the assignment. See the course website for office hour times and schedules.* **Plan ahead** *and email the instructor to make alternative arrangements if none of these times work, but do not leave this until the last moment. Failing to get signed off before the due date will result in a zero for this question.*

Note that part of the reason for this checkoff is for me to get to know you in addition to making sure you are ready for this course.

a). *(5 pts) For this problem, you should configure the Arduino programming software on your computer. Then program your Arduino with the sample Blink program (*`file->examples->basics->blink`*). Once you have verified that you can compile and program your board with this sample program, modify it so that it blinks a long on, long off, long on, short off, short on, short off, short on, long off pattern of blinks. Repeat this 4 times at startup (and then do nothing). Use a loop, do not just copy and paste code. Before doing this, make sure to read through and complete the rest of the questions in this section.*

Include a printout of the code with your assignment. **You must also turn in your code by visiting** `http://cse.unl.edu/~handin/`. *Failing to electronically turn in your code will result in up to a 20 point penalty on this assignment. Points may also be deducted for coding errors, poor style, or poor commenting.*

b). *(2 pts) How did you specify the version of the Arduino you are using in the Arduino programming environment?*

c). *(2 pts) How did you specify the serial port the Arduino is connected to in the software? What is the port name on your computer?*

Problem 3. *Resistors*

a). *(4 pts) In your packet you should have different types of resistors. Resistors are color coded to indicate their values. What value resistors do you have and what color pattern indicates that? (Hint: This is one problem that is easily solved by googling.)*

Problem 4. *Button Input*

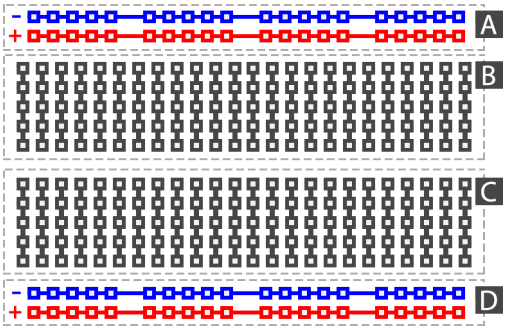

Figure 1: Breadboard connections.

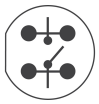

Figure 2: Button connection (when not pressed).

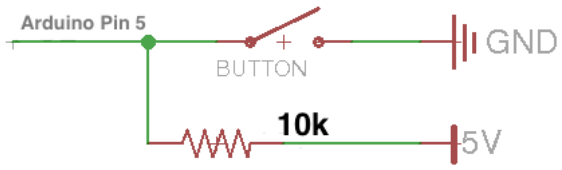

Figure 3: Schematic diagram to connect the button.

Remember, you should only connect wires and components when the Arduino is not powered or connected to your computer! Read this section and examine the sample code before doing any of the connections.

For this part of the homework you will need to use the breadboard, wires, a resistor, and button. Figure 1 shows how the wires in the breadboard are connected to each other. Areas A and D are connected horizontally and are usually used as power buses (use the standard convention of connecting positive voltage to red and ground to black/blue). Sections B and C of the breadboard are connected vertically within the block. **If you are unsure about how to use or connect to a breadboard please ask the instructor or a TA before connecting.**

Figure 3 shows a schematic for connecting the button to the Arduino and Figure 2 gives details on the button itself. When the button is pressed, all of the terminals are connected together. When it is not pressed, the top and bottom terminals are not connected, but the bottom two pins are still connected to each other as are the top two pins. Note that you can determine the orientation by looking at the slight "D" shape of the button and drawing. Following the schematic in Figure 3, connect the button to the Arduino, if you are unsure how to properly wire this connection, please ask the instructor. Note that you should connect the button to the pin labeled 5 on your arduino.

In order to read the button press, look at the example code for the button (`file->examples->digital->button`*). You will need to look at the code and modify it slightly to switch the button input to pin 5.*

a). *(4 pts) Include a printout of a picture of your final configuration that shows the breadboard and Arduino. Make sure the wires are organized in such a way that you can tell where everything is connected.*

b). *(4 pts) Modify the code so that the LED blinks slow when the button is not pressed and blinks fast when it is pressed. Include a copy of the relevant code (mainly the code in* `loop()`*) below or as an attachment.*

Problem 5. *(30 pts) Programming assignment: On the course website there is a C source code file that you will need to complete. The comments in the file indicate the portions of the code you must complete. You can compile and test this code by sshing to* `cse.unl.edu` *or using most any other standard C compiler. On the cse server, compile it using the command:* `gcc hw1.c -Wall -o hw1` *and then test it by running* `./hw1`*. See the instructor or TA if you have questions about this process.*

Include a printout of the code and its output with your solutions. **In addition you must also turn in your code by visiting** `http://cse.unl.edu/~handin/`**.**

Problem 6. *Dynamic memory allocation*

a). *(4 pts) What section of memory does* `malloc` *use? Give at least 2 reasons why it is a bad idea to use* `malloc` *on an embedded system?*

b). *(6 pts) For the following code, indicate where the memory for the variables are allocated (heap, stack, or global).*

```
volatile uint16_t var = 0x23;
uint8_t *ptr;
uint8_t data[] = {0,1,2,3,4,5,6,7};

int8_t runFunction(void){
  uint16_t var2 = 0x1FF;
  uint8_t *ptr1 = data + 2;
  static uint32_t executionCount = 0;
  //Don't ever use malloc!
  ptr = malloc(sizeof(uint8_t) * 10);
  executionCount++;
}
```

Problem 7. *Bit operations and data types*

a). *(4 pts) What is the value of* `((236>>1) & (0x5<<2)) | 9` *in* **hex***? Show your work for each step for full credit.*

b). *(4 pts) What is the value of* `(((0x1A<<2) + 0x18) & (1<<4))` *in* **hex***? Show your work for each step for full credit.*

c). *(4 pts) What are the maximum and minimum values of int8_t and uint8_t in* **decimal***?*

d). *(4 pts) What is the maximum and minimum values of int16_t in* **hex***?*

e). *(4 pts) Write the* `C` *code to set the 8-bit memory location 0xCD to value 0x42.*

f). *(5 pts) What is the value of* `output` *in* **hex** *after the following code has been executed?*

```
uint8_t output = 8;
uint8_t counter = 10;
uint16_t offset = 0x236;
while(counter > 0){
  counter = counter >> 1;
  output++;
}
output += offset;
```

Problem 8. *Pointers and memory, refer to the following code for these problems:*

```
uint8_t var = 10;
uint8_t data[] = {0,1,2,3,4,5,6,7};
uint8_t *p1 = &var;
uint8_t *p2 = data+3;

\\Draw picture here

data[1] = *p1;
p2[2] = 0;
```

a). *(5 pts) Draw a picture showing where each of the variables are stored and for the pointers, where the data they point to is stored right after all the types are defined (at the point where there is a comment in the code).*

b). *(5 pts) What are the values of* `var` *and* `data` *after running this code?*

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