CSCE 236 Embedded Systems, Spring 2014
Homework 1

Started: Weds, Jan 15th, 2014
Due: Friday, Jan 25th, 2014 (start of class)

Instructions: This homework is an individual assignment, collaboration is not allowed. If you discuss any problems with others, please note this on the assignment as described in the syllabus. Also note any materials outside of lecture notes, course textbooks, and datasheets that you used. Show your work and describe your reasoning to get partial credit if your solution is incorrect. This homework is due on the date listed above before the start of class.

Name:

Problem 1 (5pts). *(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?*

Problem 2 (40pts). *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. **You must also turn in your code by visiting** `http://cse.unl.edu/~cse236/handin/`**.** *If you have not used cse handin previously, you will need to register using the registration link on the left of the webpage. 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.*

Problem 3. *Dynamic memory allocation*

a) (5pts). *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) (5pts). *Assume a processor has 4.0k of RAM and there are two functions that require a 2k temporary buffer. These functions are called often, but never execute at the same time. Is it better to use a global shared buffer or a local variable within each function? Explain.*

c) (5pts). *For the following code, indicate where the memory for the variables are allocated (heap, stack, or global).*

```
uint8_t var = 0x10;
uint8_t *ptr;
uint8_t data[] = {0,1,2,3,4,5,6,7};

int8_t main(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 4. *Bit operations and data types*

a) (5pts.). *What is the value of* `((64>>2) ^ (13<<1))` *in* **hex***?*

b) (5pts.). *What is the value of* `(((16<<2) + 7) | (1<<6))` *in* **hex***?*

c) (5pts). *What are the maximum and minimum values of int16_t and uint16_t in* **decimal***?*

d) (5pts). *What is the maximum and minimum values of int8_t in* **hex***?*

e) (5pts). *Write the* `C` *code to set the 8-bit memory location 0x10 to 0x49.*

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

```
uint8_t output = 13;
uint16_t counter = 0x7FFF;
output += counter;
while(counter > 0){
  counter = counter >> 1;
  output++;
}
```

Problem 5. *Pointers and memory, refer to the following code for these problems:*

```
uint8_t num = 39;
uint8_t arr[] = {79,80,81,82,83,84,85,86};
uint8_t *p1 = arr+5;
uint8_t *p2 = &num;

//Draw memory layout at this point

*(p1+2) = *p2;
arr[3] = *p2+4;
p1[0] = *(p1+1)+5;
```

a) (5pts). *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) (5pts). *What are the values of* `num` *and* `arr` *after running this code?*

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