CSE3442 (Spring 2020)
Lab #4

In this lab, you will start coding the first steps of the project. In the class project, a terminal interface needs to be coded so that the device can be controlled. The following steps will guide you through this process:

1. Start with the serial.c file from class, renaming the file lab4_*your_name*.c, where *your_name* is replaced with your name as it appears in MyMav. The code in initHw(), getcUart0(), putcUart0(), and putsUart0() can be used without modification.
2. Add this code to the file towards the top of the file to indicate the maximum number of characters that can be accepted from the user and the structure for holding UI information:
   ```
   #define MAX_CHARS 80
   #define MAX_FIELDS 5
   typedef struct _USER_DATA
   {
   char buffer[MAX_CHARS+1];
   uint8_t fieldCount;
   uint8_t fieldPosition[MAX_FIELDS];
   char fieldType[MAX_FIELDS];
   } USER_DATA;
   ```
3. The code in main() can be rewritten as follows:
   ```
   USER_DATA data;
   initHw();
   getsUart0(USER_DATA* data);
   putsUart0(data.buffer);
   while (true);
   return 0;
   ```
4. Code the function getsUart0() function implementing this algorithm:
   a. Initialize a local variable, count, to zero.
   b. Get a character from UART0.
   c. If the character is a backspace (ASCII code 8 or 127) and count > 0, then decrement the count so that the last character will be overwritten.
   d. If the character is a carriage return (ASCII code 13) (this is sent when you type <*enter_key*> indicating the end of the string), then add a null terminator (ASCII code 0) to the end of the string and return from the function.
   e. If the character is a space (ASCII code 32) or another printable character (ASCII code >= 32), then store the character in data->buffer[count]. Increment the count. If count == maxChars (no more room in the str[], then add a null terminator to the end of the string and return from the function.

f. Loop back to (b) unless the function is exited in (d) or (e).

5. Test the function as follows:

a. Type the following string in the PC terminal application: test project<*enter_key*>

b. Verify that the output string is: test project

c. Type the following string: ABCe<*backspace_key*>defg<*enter_key*>

d. Verify that the output string is: ABCdefg

e. Type the following string without pressing a trailing <*enter_key*>:
1234567890123456789012345678901234567890123456789012345678901234567890123456789 01234567890

f. Verify that the output string is:
1234567890123456789012345678901234567890123456789012345678901234567890123456789 01234567890

g. Demonstrate your code and e-mail the file to the grader.