Lab 5 – Simple C++ Terminal

Overview

In this two week lab, you will create a simple C++ based serial terminal that allows you to talk to your Wunderboard. Your C++ program must be robust and catch errors. You will need to setup the C++ serial classes for your operating systems and create a number to ASCII conversion system on the Wunderboard. You will be using the basic 'stream' functionality of C++.

Prelab

1. No prelab this week

Procedure

1. You need to make both a Wunderboard program and a C++ program in this lab. It is recommended you start with the Wunderboard program and then write the C++ program. No Wunderboard skeleton file is provided, so please use a previous lab as a starting point.

2. The Wunderboard needs to be able to write ASCII numbers to the serial port. For example, the decimal value (one byte) of 123 needs to be converted to an array of characters '1', '2', '3', and '\0'. You must write your own function to do this. You may not use itoa(), sprintf(), or similar function.

3. Once you have this functioning, you should write a simple program that outputs a count four times a second to the serial port at 9600bps. When the count reaches 50, it should count backwards to 0, then count up to 50, etc...

4. It is recommended that you test this using a standard terminal program (Termite, CuteCom, or screen as appropriate). You would connect to the Wunderboard and see the counting live.

5. Once you are sure the Wunderboard is working download the C++ sample. Based on your operating environment, the installation of the serial classes varies system to system. Please check the appendix for more information and if stuck, consult your TA.

6. The sample C++ file contains some of the startup code needed to properly setup the serial class for use. Read through the code and take note of comments that say you need to make changes. Start by figuring these out. This first version of the serial code is simplistic. We will be improving it over the up-coming weeks.

   Reading comments to understand code might seem frustrating at first, but this type of work is very common in programming when using libraries supplied by others.

7. Once you feel ready to update the code, start by setting up the C++ to take whatever is sent over the serial port and display it to the screen. You can exit a C++ program running on the command line by typing 'CTRL-c'.

8. Finally, add in the ability to transmit anything entered on the keyboard followed by a return over the serial port. Do not send the return, and if what is entered is 'EXIT' close the program.

Demonstrate and Submit code

Show your code and program to your TA. They will watch your system run to see that your Java program displays the count up and down. While counting, they will unplug the Wunderboard to ensure it is the source of the counting.

Lab Summary

Task	Completed?
Prelab	0pts.
Custom itoa written	6pts.
Wunderboard code works	6pts.
C++ code works	8pts.
Study Questions	0pts.