

UNIVERSITY OF BRITISH COLUMBIA
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

EECE 259: Introduction to Microcomputers
Assignment 4: Programming C, Interrupts and Project
Handed out March 21, 2011

Overview

In this homework, you will learn how to:

1. Write C and assembly language programs with subroutines,
 2. Perform more complex decision-making,
 3. Use interrupts or polling to solve problems with simultaneous time constraints, and
 4. Specify and solve a bigger project involving both hardware and software.
-

PART 1: Study Questions

Easy ones...

1. Write a C program *and* Nios II assembly language program to count the total number of switches (SW9 to SW0) that are set to '1'. Place the count on the green LEDs. Run the program continuously in a loop so it is always adding the switches and displaying the count.
2. Modify the C and assembly programs from the previous question to count the longest sequence of '1's. For example, if the switches are %0111010110, the longest sequence is 3.

Medium...

3. Write a C *and* assembly program to compute x^y and display the result on the red LEDs. The integers x and y should be input one at a time using the switches. For example, set the switches to the value for x then press and release KEY3. Next, set the switches to the value for y then press and release KEY3 again. The program should automatically compute the answer and wait for the next value of x . *Suggestion:* use one subroutine to input the data and another to compute x^y .

Hard...

4. Write a C subroutine and an assembly language subroutine to copy and modify a null-terminated string from one location in memory to another. Assume the *from* address is given in register r4, and the *to* address is given in register r5. While copying the string, you must clean up some messy typing by modifying it as follows:
 - When there are two or more spaces in a row separating two words in *from*, ensure only a single space is placed in *to*.
 - When you encounter a period in *from*, assume it ends a sentence. Ensure sentences ending with a period are always followed by exactly two spaces in *to*. (Be careful that you do not add any additional spaces that may appear right after the period in *from*.)

5. An **ABS brake controller** does two things at the same time. First, it monitors an input signal to indicate that the wheel is spinning. Second, if the wheel ever stops spinning, it must quickly release and engage the brakes (pulsing them) several times per second. By pulsing the brakes, the wheel is allowed to spin again and this restores tire traction. For this problem, you will write a Nios II assembly language program and use interrupts to do the following two things at the same time:
- **Detect wheel spin.** First, your program should continuously poll a switch and count the number of 0-to-1 transitions. For this, assume KEY3 represents an encoder attached to the wheel: if the wheel is spinning rapidly, you will get many 0-to-1 transitions per second. You can assume that at least 10ms will pass before the next 0-to-1 transition occurs. However, you don't know exactly when it will occur. Hence, the polling must be done frequently enough to not miss any transitions!
 - **If no spin, pulse the brakes.** Second, your program should apply the brakes by sending a '1' to LEDG0. Every 100ms (**approximately**), you should check that at least 5 transitions have been detected, suggesting the wheel is still spinning. If fewer than 5 transitions have occurred, release the brakes for the next 100ms period by sending a '0' to LEDG0. If at least 5 transitions have been detected, the wheel is still spinning so you should continue to apply the brake for the next 100ms.
 - *Hint: you should use interrupts for this!*

... continue to next page...

PART 2: Practical Assignment Project

For PA4, you will create your own interesting project. It should involve programming your DE1 board, and it may involve interfacing with other hardware. To get you started, a few suggestions are listed on the course web site.

Before you start working on your project, *it must be approved by a TA or the instructor*. The precise steps are as follows:

1. Think of a few cool project ideas. Talk to friends, TAs, or the instructor. Keep it simple enough you can actually do it on time, but make it interesting!
2. Write up a short 1-page proposal/specification and present it to your TA.

Your proposal should be less than ½ page of text. ***Use point form!***

Fully describe the inputs, outputs, and operation of the system.

You must be clear whether you are going to use **assembly language** or **C language**.

You are encouraged to use diagrams or figures to help explain the goals of your project.

The proposal should *clearly specify* what is being promised (i.e., the deliverable details). Think of it as a *contract* between you (vendor) and the TA (customer). The customer needs to know *exactly* what he is going to get for the money he gives you (grades). Note: the contract must be *complete enough* that if another group in 259 saw it, they could build the *exact same thing* and the customer could not tell the difference without “looking inside” at the source code etc.

3. The TA will give you some verbal feedback. You will probably have to rewrite your proposal, or you may have to choose a different topic. Get the form back to the TA for signing.
4. When the proposal is exactly perfect, the TA or instructor will sign your Proposal Form. **You must keep this signed form (the TA does not need a copy).**
5. A previously *Signed Proposal Form* is a requirement to get marked in PA4. It indicates the TA/instructor agreed the proposal is complete, unambiguous, and solvable in the given time.
6. The *Signed Proposal Form* is a formal contract. Do not lose it. You must present it to your TA again during the marking session for PA4. The TA will mark you based upon *how well you meet your prior commitments given in the contract*. If you completely meet the contract, and answer all the TA’s questions adequately, you will get 10/10.
7. Up to 3 bonus marks will be awarded to each project that *significantly exceeds* the contract requirements. (Note: minor improvements, like adding a blinking light, are not sufficient.)