Programming Assignment

October 19, 2016

Your second programming assignment will duplicate the functionality of programming assignment 1 using drivers to properly abstract and encapsulate the hardware interfaces. You will create a driver for the push-buttons and LEDs, then use these drivers to implement a system that duplicates the functionality of the first assignment. Your code must be perfectly functional (a necessary condition to receive a grade). You grade will be based on how well your drivers and program match the programming practices outlined in class and in the course coding guidelines found in the wiki. Specifically, you will be graded on the following:

- The main program accesses hardware only through the drivers
- Hardware is correctly configured by the driver
- The drivers do not interface directly with one another (e.g. the push-button driver should not be calling functions to maniipulate the LEDS).
- Drivers manipulate only the hardware they are intended to work with (e.g. DDRD=0 would be inappropriate to clear both LEDS).
- The main program calls the initialization routines only at the appropriate point, and does not repeatedly reinitialize the hardware.
- Your program uses appropriate types for all variables and functions.
- Overall, you code can be understood by someone not familiar with this course or the actual purpose of the program. This includes using appropriate variable and function names and appropriate programing constructs (e.g. use the correct form of while statement).
- Every source file contains your name, and a brief (one or two line) description at the top (e.g. "Interface definition for pushbutton driver" or "Implementation of push-button driver")
- Your program does not use any third-party libraries or functions.
- Every source file (.h and .c) contains a #include for **all** necessary include files. For example, if your pushbutton.h uses uint8_t, it should include <stdint.h>.
- Every header file is protected against multiply inclusions
- You followed submission and file naming instructions

Your pushbutton driver must have, at a minimum, the following interface (debouncing is not necessary)

- void initialize_pushbutton_array()
- bool left_button_is_pressed() //returns true if left button is pressed
- bool right_button_is_pressed()
- bool middle_button_is_pressed()

Your LED driver must have, at a minimum, the following interface:

- initialize_left_led()
- initialize_right_led()
- turn_on_left_led()
- turn_off_left_led()
- turn_on_right_led()
- turn_off_right_led()
- toggle_left_led()
- toggle_right_led()