CE161 – AUTUMN 2012
ASSIGNMENT

Assignment Due at 11:59:59 am (right before noon) on Wednesday, Dec. 12.

On Plagiarism

The work you submit must be your own. Any material, code or ideas you make use of in this assignment, whether from textbooks, classmates, the web or any other source must be acknowledged in your report, and the extent of the reference clearly indicated. In doubt, please consult the University's guidelines on plagiarism:

http://www.essex.ac.uk/plagiarism

As a rule, any material, idea, code or text that was not created solely by you (and solely for this assignment) must be referenced and/or credited appropriately.

Electronic Submission URL: https://courses.essex.ac.uk/ocs/

Please also see your handbook for rules regarding the late submission of assignments

ASSESSMENT AND RULES

- The assignment should preferably be done in pairs, but you are allowed to do it individually if you prefer. Groups of 3 or more are NOT allowed.
- You must NOT discuss solutions to the assignment with anyone other than your partner in the submitted assignment.
- This assignment counts for 20% of the overall marks for CE161.
- Be sure to put your registration number(s) and name(s) on your submitted code.

STRUCTURE OF THE ASSIGMENT

You are to submit a single ***main.c*** file to be built and run in the AT91SAM71S256-EK board. The file will contain your solutions to three levels of problem. The levels are as follows:

- LEVEL 1, for up to 40% of the assignment (i.e., passing mark): a simple C program with some variations on the work done in the labs.
- LEVEL 2, for up 60% (requires completion of LEVEL A1): an intermediate task which significantly extends the work done in the labs.
- LEVEL 3, for up to 80% (requires completion of levels 1 and 2): a more advance task.

In addition to these levels, (up to) an additional, independent 20% will be given for the clarity, amount and usefulness of code comments.

The tasks for each level are described in the next two pages.

GENERAL RULES FOR THE main.c FILE

The `main.c` program file you submit must start with a few comment lines containing your name (and that of your partner in this assignment) and registration number. You are welcome to use parts of the programs we used in the CE161 labs, but be sure to acknowledge (in the form of program comments) the source and what parts of the code were borrowed.

The program will be marked in terms of functional completeness (does it do everything it is required to do?), clarity (can a third person easily understand the code based on the variable names and indentation?), and simplicity (i.e., good programming practice: try to find simple ways to accomplish the required functionality; don't get fancy for no reason).

Your program must not only execute the required functionality as described below, but it must also ensure that buttons and LEDs do not work out of sequence (for example, the program must not execute BUTTON5's function out of the required sequence or when another button is pressed , etc.).

You should be able to do all levels in this assignment without having to create any new external functions or header files. However, you may need to add more items to the list of #include files. **Your submission should consist of a single C file. Both you and your partner must submit the same file**.

NOTE: If the file you submit does not compile in the CSEE Lab 1 computers, you will only obtain partial marks.

Caution: If your built binary program for some reason turns out to be larger than 64kB, the board will not take it in its SRAM. In that case (and only in that case) you should use the board's Flash memory (which has 256kB), use the *flash.bin* file (instead of the *sram.bin* one), and use address 0x10000 (instead of 0x202000; answer 'no' if SAM-BA asks whether to unlock any addresses). Do this only if necessary and, if you do it, add a comment at the top of the *main.c* program, right under your name, letting me know about it.

LEVEL 1 – Basic user interaction functions

In order for the program to execute levels 2 and 3 described below, you program must first have a user interface stage. As follows:

Program Entry: Your program should start by running two red lights back and forth and in opposite directions as done in the lab activities in Week 9. While the board is in this state, it must wait for the user to press either button 2 or 3 to indicate which level she/he wants to test (pressing button 2 will execute Level 2, etc.).

Once the desired level has been tested, the user will press button 7 to return to the program entry stage. Pressing button 8 at any time will cause the entire program to exit.

Before exiting the program, the board will show the word 'OFF' for three seconds on the display.

LEVEL 2 – A countdown timer

You will program the AT91 board to work as a countdown timer. The board buttons will be used for controlling the board and entering the desired timer numbers. The LEDs will be used for indicating to the user when he/she can enter commands.

Your level 2 program must execute the following 6 steps (in the order shown):

1. Start the level 2 part by flashing all 8 LEDs with red for two seconds.
2. All LEDs are then turned OFF, except for LED1, which now remains green. When LED1 is green this means the board is now waiting for the user to enter the time at which the countdown will start.
3. The start time will be up to 60 seconds and 00 hundredths. So, this is the number that will be shown at this stage on the AT91 board display: '60.00'. To change this number, do the following:
    a. Pressing BUTTON1 reduces the seconds, one by one. BUTTON3 will increase the seconds. Holding one of these buttons must cause the changes to be faster. The maximum seconds must not exceed 60. The minimum is zero.
    b. Pressing BUTTON2 reduces the hundredths, one by one. BUTTON4 will increase the hundredths. Again, holding the button will cause the changes to be faster. The maximum hundredths must not exceed 99. The minimum is zero.

    NOTE: The display should change to the new start time after every change. So, for example, to start the countdown at 57.94 seconds, press BUTTON1 three times and BUTTON2 six times. '57.94' should then appear on the board display after this is done.

4. Once the desired start time has been set, press BUTTON5 to start the countdown. At this point, all LEDs should be OFF, but the display must show the countdown time remaining.

    NOTE: use a stop watch or the computer clock to check that the countdown runs at the correct speed (e.g., 10s should actually take 10s, etc.). The hundredths do not need to be perfect, but they need to decrement and reset properly (i.e., start at 99 again every time a second goes by). If the countdown timer is not very accurate, to the second, then you must change your program to correct this.

5. When the countdown reaches '00.00', the display will show '- - . - -' and all LEDs will be orange for 3 seconds.
6. After this, the user should either press BUTTON6 to return to step 1 above, or BUTTON7 to finish this part of the program . The display should show only zeroes before leaving from this stage.

LEVEL 3 – A number conversion calculator

This part of the program will allow the user to enter a (big-endian) 6 bit binary number and convert it to either decimal or hexadecimal, by executing the following steps:

1. When the program starts at this level, LED8 will blink 4 times in red colour and the display will show '--.--'. At this point, the user will press button 1 to choose conversion to decimal, or button 2 to do conversion to hexadecimal.
2. After the user has chosen the conversion mode, LED8 will blink 4 times in green colour. At this point the user will utilize buttons 1 to 6 to enter the binary number to be converted. Pressing a button will make that bit 1. Not pressing it, will leave it as 0. So, for example, if the user presses buttons 1 and then 3, the entered binary number will be 101000. Buttons 1-6 can be pressed in any order.
3. Once the user has entered the desired binary number, pressing button 7 will execute the desired conversion. The result will be shown in the board's display. The correct result must be shown.
4. At this point, the user should press button 6 to convert another number or button 7 to return to the main program entry stage.

The program needs to be able to convert positive integers only.