Embedded Systems Coursework Part 2

Supplementary Information

Stan Manilov

February 27, 2017

Here are some hints to help you get started with the coursework.

1 Work-flow

The debugging facilities are now more limited than for the first coursework. It is essential that you implement and test each part of your solution separately, before moving onto the next part. For this reason, it is useful to come up with a plan of how to break down the task. Figure 1 is an example dependence chart of the tasks you might decide to complete.

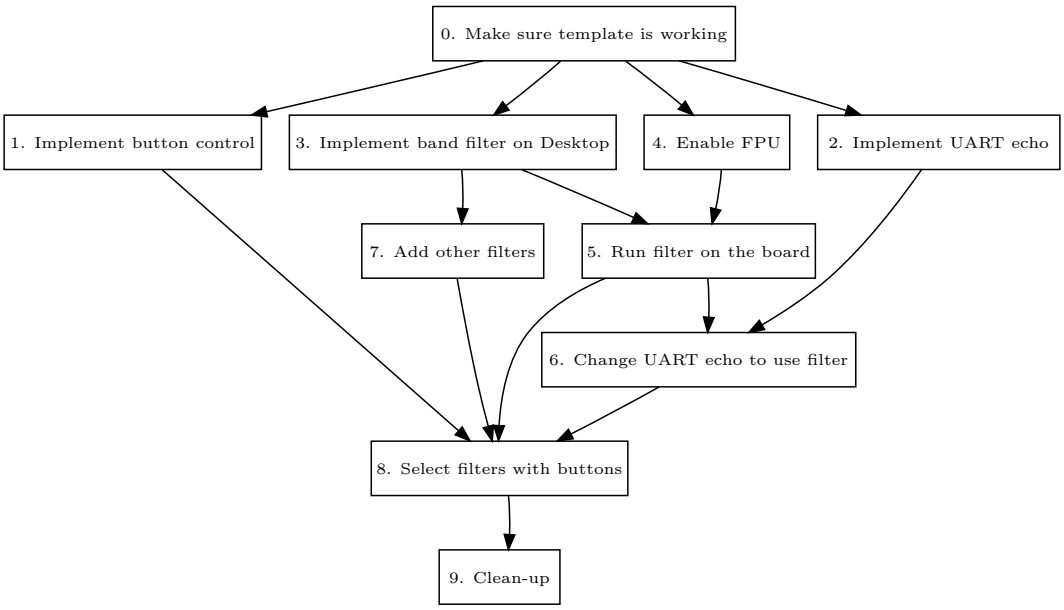

Figure 1: Example work-flow

You don't need to follow that work-flow, but again, make sure you define the separate steps you would need to take in order to complete the coursework.

2 Getting Started

You might be tempted to read everything you need to know to solve the coursework before you start writing any code. I would advice against that. My recommendation is to start by compiling and running the template code. Modify small parts of it and make sure you know which part of the code does what. Here are some questions that might help you get a deeper understanding of what's going on inside our embedded system:

- How can you blink the other LEDs?
- How are the LED on/off/toggle functions implemented?
- What are the `GPIOA_PSOR`, `GPIOA_PTOR`, `GPIOA_PCOR` registers?
- Why is it that data is always written to them? Why not read the data, when we want to toggle, e.g.?
- How is the clock for the LEDs enabled?
- What is initialized `in __init_hardware`?
- What is the purpose of `bareboard_flash.lcf`?
- How is it being used?
- How does the `Makefile` work?
- How about `vector.h` and `vector.c`?

These are just some questions to start you thinking. You don't need to know the answers to all of them in order to complete the coursework, and some are harder than others. Just run things and see what happens. Change things and see what breaks.

3 Push-buttons

I recommend you start coding by implementing functions to let you init and poll the push-buttons. Use `led.c` for inspiration. In particular, you would need to look at the documentation (K70 Ref Manual) for the registers assigned in `led.c` (`PORTA_PCR11`, `SIM_SCCG5`, etc.). The push-buttons are input-devices, while the LEDs are output devices. Also, looking at the K70 Tower Ref Manual will help you find which pins are the buttons connected to.

4 Closing note

1. Remember to come to the labs now: you are likely to have questions.
2. Ask me questions.

3. Good luck!

; 0x5374616e