Lab 8 Software Drivers for an Embedded System

This laboratory assignment accompanies the book, Embedded Systems: Real-Time Interfacing to ARM Cortex M Microcontrollers, ISBN-13: 978-1463590154, by Jonathan W. Valvano, copyright © 2012.

Goals	• Collect and test all hardware components for the project,
	• Write low-level software for each I/O interface,
	• Validate that overall project is possible, and update the requirements document.
Review	• Data sheets for your microcontroller, and your hardware components
Starter files	None

Background
       Normally, one passes around the design cycle multiple times. However, in this class, Labs 7, 8 and 11 represent one pass around the design cycle. In this lab you will perform prototype hardware and software implementations. The goal is to verify your collection of chips, resistors, capacitors and connectors will operate as intended when you get to Lab 11. Furthermore, you will write and test low-level software drivers for all I/O interfaces. The more you develop and test during this lab, the higher the probability your product in Lab 11 will function as intended.

Preparation (do this before your lab period)
Part a) You must have all resistors, capacitors, chips, connectors, and components for the project. If your project uses the LM3S811, this part will be delivered to you with the PCB. Otherwise you should have all parts.

Part b) If necessary, show on your circuit diagram (SCH file) how this Lab 8 test prototype will be different from the final system. For example, your final product might run on a LM3S811, but you will test it on the LM3S1968.

Part c) Write all the header files (prototypes for public functions) for your entire system. The most efficient process is to have designed, implemented and typed all software as part of the preparation. In this way, you can build the circuits and debug while the TA is present.

Procedure (do this during your lab period)
Part a) Running on the prototype system, debug the hardware and software in a modular fashion. You should write and test all software that directly interacts with the hardware. You do not need to have the high level software written, but all low-level device driver software should be written and tested in Lab 8.
Part b) The requirements document is fluid, meaning it is expected to change during product development. You may find your system will have higher or lower performance than originally conceived. You may also need to modify what it will do. Rewrite your requirements document to reflect what you now think is realistic for your system.
Part c) Collect preliminary performance data as appropriate. This step is especially important if your device has transducers.

Deliverables (exact components of the lab report)
A) Objectives
        1-page requirements document (Procedure b)
B) Hardware Design
        Modified circuit diagram (SCH file), as tested in Lab 8
C) Software Design
        Include the low-level I/O drivers (Preparation c, Procedure a)
D) Measurement Data
        Give any performance data you collected (Procedure c)
E) Analysis and Discussion (none)

Checkout (show this to the TA)
        Show the hardware/software prototype of your system to your TA. Explain how the prototype is different from the final product. Discuss the testing features of your design. Discuss your top three worries about finishing Lab 11 on time.

Lab 8 grading (different from labintro.pdf)
Lab 8 is the second of three parts to your own project. The grading rubric for this lab will be different from the one mentioned in the labintro.pdf document."

****
*****

You will be graded on the quality of your prototype design i.e. how close to the final system your prototype is. You should be able to state where all your prototype design differs with the actual system and how you will account for this difference while doing your measurements.

Preparation (20) shown to TA before lab starts
    Show all parts to the TA, except the processor and surface mount accelerometer. (10)
    Most recent SCH file (5)
    Header files for software drivers (5)

Checkout (30)
    Prototype demonstration, quality of design (25)
    Identifying top 3 worries (5) - Not identifying a worry is a worry. The earlier you identify it, the better it is. So think hard on it.

Software Quality (30)
    Modularity and organization (10)
    Readability (10)
    Functionality (10)

Report (20)
    Refined requirements document (10)
    Measurement data (5)
    Modified Hardware Circuit SCH (5)