Hyder Shad
hs25796
EE 445L Lab 3
TTH 12:30 -2pm
9/28/2018

Lab 3 Report

Requirements Design:

1. Overview

1.1. Objectives: Why are we doing this project? What is the purpose?

The objectives of this project are to design, build and test an alarm clock. Educationally, students are learning how to design and test modular software and how to perform switch/keypad input in the background.

1.2. Process: How will the project be developed?

The project will be developed using the TM4C123 board. There will be switches or a keypad. The system will be built on a solderless breadboard and run on the usual USB power. The system may use the on board switches and/or the on board LEDs. Alternatively, the system may include external switches. The speaker will be external. There will be at least four hardware/software modules: switch/keypad input, time management, LCD graphics, and sound output. The process will be to design and test each module independently from the other modules. After each module is tested, the system will be built and tested.

1.3. Roles and Responsibilities: Who will do what? Who are the clients?

EE445L students are the engineers and the TA is the client. Students are expected to modify this document to clarify exactly what they plan to build. Students are allowed to divide responsibilities of the project however they wish, but, at the time of demonstration, both students are expected to understand all aspects of the design.

1.4. Interactions with Existing Systems: How will it fit in?

The system will use the TM4C123 board, a ST7735 color LCD, a solderless breadboard, and be powered using the USB cable.

1.5. Terminology: Define terms used in the document.

Power budget, device driver, critical section, latency, time jitter, and modular programming. See textbook for definitions.

Power Budget: how much power is available to power devices or components and within the designer/client constraints

Device Driver: software that provides the interface between programming code and the hardware structure

Critical Section: when parallel processes or interrupts attempt to write/read from the same location and values are overwritten or not updated/read correctly.

Latency: the delay or time it takes to complete a process, such as entering and exiting an interrupt or taking an ADC sample or the time it takes for a piece of hardware to trigger/activate.

Time Jitter: differences and variations between the time a process is scheduled to be started and when it is actually started.

Modular programming: having code a main code that uses subroutines and call functions which implement specific tasks.

1.6. Security: How will intellectual property be managed?

The system may include software from Tivaware and from the book. No software written for this project may be transmitted, viewed, or communicated with any other EE445L student past, present, or future (other than the lab partner of course). It is the responsibility of the team to keep its EE445L lab solutions secure.

2. Function Description

2.1. Functionality: What will the system do precisely?

The clock must be able to perform five functions. 1) It will display hours and minutes in both graphical and numeric forms on the LCD. The graphical output will include the 12 numbers around a circle, the hour hand, and the minute hand. The numerical output will be easy to read. 2) It will allow the operator to set the current time using switches or a keypad. 3) It will allow the operator to set the alarm time including enabling/disabling alarms. 4) It will make a sound at the alarm time. 5) It will allow the operator to stop the sound. An LED heartbeat will show when the system is running.

2.2. Scope: List the phases and what will be delivered in each phase.

Phase 1 is the preparation; phase 2 is the demonstration; and phase 3 is the lab report. Details can be found in the lab manual.

2.3. Prototypes: How will intermediate progress be demonstrated?

A prototype system running on the TM4C123 board, ST7735 color LCD, and solderless breadboard will be demonstrated. Progress will be judged by the preparation, demonstration and lab report.

2.4. Performance: Define the measures and describe how they will be determined.

The system will be judged by three qualitative measures. First, the software modules must be easy to understand and well-organized. Second, the clock display should be beautiful and effective in telling time. Third, the operation of setting the time and alarm should be simple and intuitive. The system should not have critical sections. All shared global variables must be identified with documentation that a critical section does not exist. Backward jumps in the ISR should be avoided if possible. The interrupt service routine used to maintain time must complete in as short a time as possible. This means all LCD I/O occurs in the main program. The average current on the +5V power will be measured with and without the alarm sounding.

2.5. Usability: Describe the interfaces. Be quantitative if possible.

There will be two to four switch inputs. In the main menu, the switches can be used to activate 1) set time; 2) set alarm; 3) turn on/off alarm; and 4) display mode. The user should be able to set the time (hours, minutes) and be able to set the alarm (hour, minute). Exactly how the user interface works is up to you. After some amount of inactivity the system reverts to the main menu. The user should be about to control some aspects of the display configuring the look and feel of the device. The switches MUST be debounced, so only one action occurs when the operator touches a switch once.

The LCD display shows the time using graphical display typical of a standard on the wall clock. The 12 numbers, the minute hand, and the hour hand are large and easy to see. The clock can also display the time in numeric mode using numbers.

The alarm sound can be a simple square wave. The sound amplitude will be just loud enough for the TA to hear when within 3 feet.

2.6. Safety: Explain any safety requirements and how they will be measured.

The alarm sound will be VERY quiet in order to respect other people in the room during testing. Connecting or disconnecting wires on the protoboard while power is applied may damage the board.

3. Deliverables

3.1. Reports: How will the system be described?

A lab report described below is due by the due date listed in the syllabus. This report includes the final requirements document.

3.2. Audits: How will the clients evaluate progress?

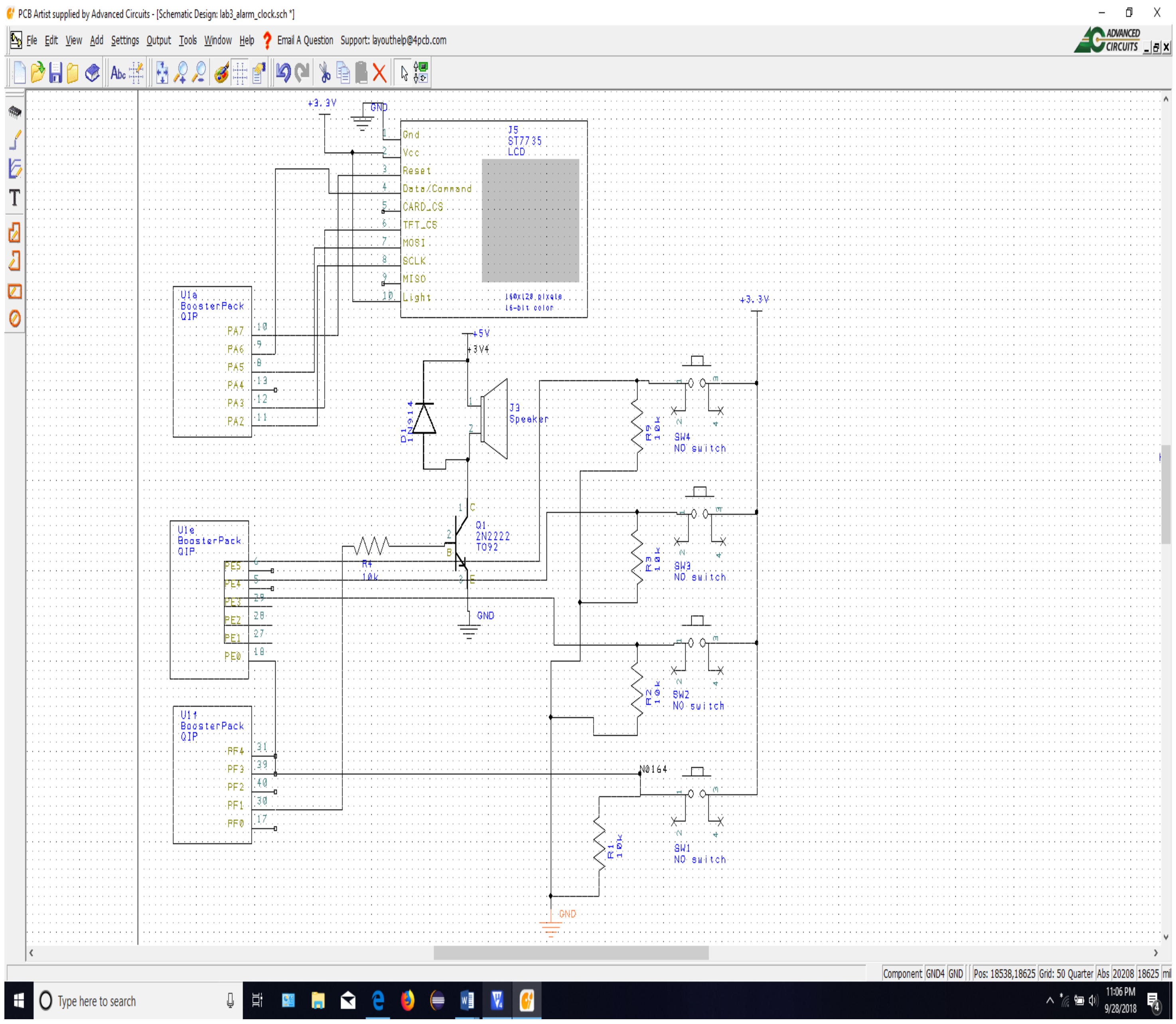
The preparation is due at the beginning of the lab period on the date listed in the syllabus.

3.3. Outcomes: What are the deliverables? How do we know when it is done?

There are three deliverables: preparation, demonstration, and report.

Hardware Design:

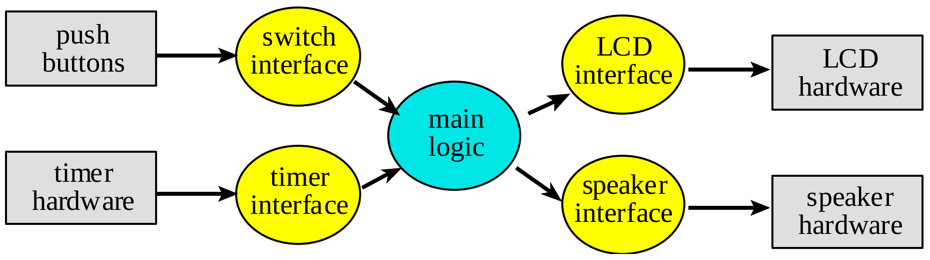
Below is the schematic for the hardware implementation of Lab 3.



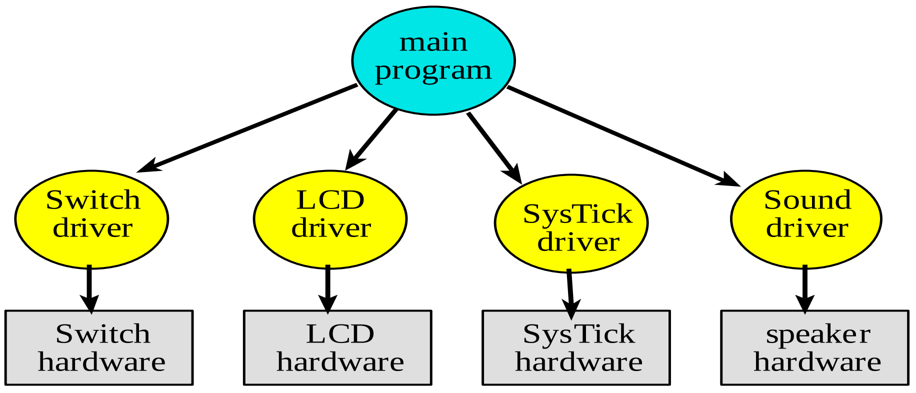
Software Design:

The approach was identical to the data flow and call graph diagrams in the lab documentation.

Data Flow:



Call Graph:



Measurement Data:

The DC voltages used by Arduino Metro Mini board to power the switches and the speaker were relatively stable. The 3.3 V voltage supply on board is measured to be around 3.31V and the 5.0V supply is measured to be 4.8V. The voltage ripple at 3.31V voltage supply is around 28.28 mVrms for noise at that level. The voltage ripple at 4.8V voltage supply is approximately 101.19 mVrms, which we expected to be higher given that a higher voltage rating normally results in a higher noise. These measurements were taken using a multimeter and oscilloscope.

For checkout, the board was hooked up to the Arduino for ease of use as a power source.
For analysis, the LaunchPad was powered using an Agilent 5V power supply. The recorded current reading for when the alarm was off was 82mA, and with the alarm ringing the current was 97mA. This jump in current is due to the alarm LED on the board as well as the small current used to activate the transistor for the speaker when the alarm GPIO pin, PF1, is turned on.

Analysis:

1. The critical sections can be removed by disabling interrupts, writing the value, and then re-enabling interrupts so no other processes can read or modify from the variable address. The variables could also be changed from global to local so other functions cannot access them simultaneously.
2. It takes approximately 5.1ms to update the LCD with a new time.
3. Updating the LCD in the ISR increases latency and results in the system not being real-time. An ISR must execute quickly so that the main program or other ISRs can handle events and data processing. If an ISR for updating the display in a car with sensor data takes too long, the main program or another ISR won't be able to process data from a sensor indicating and oncoming collision in real time and that information could be delayed to the driver and result in a crash.
4. The clock face was drawn only once at the beginning and then again if the user switched back and forth between alarm mode and current time displays. The hands were erased and redrawn because it was significantly faster. The clock face was drawn while interrupts were disabled due to the latency, so critical sections were avoided.
5. The system power requirements could be reduced by not using as many LEDs as my design did, or by switching to a non-backlit, monochrome LCD display. A buzzer instead of a speaker could be used, or a different type of smaller speaker. The board's clock speed could also be reduced to save battery life without significantly affecting the ability to record button inputs. Using the WaitForInterrupt() puts the board in low power mode until an interrupt is triggered, which also would greatly minimize power consumption