Lab 3 Report

Allen Pan & Paris Kaman

9/28

Requirements document

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- The allocation, within a system, of available electrical power, among the various functions that need to be performed

   Device driver- device driver is a computer program that operates or controls a particular type of device that is attached to a computer

   Critical section- a part of code where two (or more) pieces of your code might access the same variable at the same time, resulting in wrong values

   Latency- amount of time between when a signal happens and when it is recognized

Time jitter- difference between the shortest and longest execution time of a interrupt service routine

Modular programming- keeping parts of the code partitioned into different files for different types of actions, different devices, etc.

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?

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.

3) It will allow the operator to set the alarm time.

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.

6) An ADC will be used on a knob potentiometer to allow quick time and clock setting in the menu interface.

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 does not have critical sections. The interrupt service routine used to maintain time completes in as short a time as possible. All LCD I/O occurs in the main program.

The while loop in main is the critical section. It updates the global variable time, alarm, curStage, stages array, etc… And it draws on the LCD, based on these global variables. All these operations are supposed to be atomic, therefore we disable the interrupt at the beginning of the while loop, and enable it upon exit. The two ISRs, systick and button press, will not be allowed to occur until it exits the critical section. But we made this critical section short enough so it doesn't affect the interrupt functionality.

Since we are only setting a flag in systick handler, then update the time in main, it does not affect the usage of time in the button ISR. And for the alarm flag, which is cleared in the button ISR and set in the systick ISR, we made it atomic when clear the alarm flag, so systick won't interrupt when we are at the processing of turning the alarm flag off. The situation here is, if systick tries to enable the alarm flag when we are just pressing the button to disable it. This happens when a user presses the alarm disabling button when the alarm isn't even on, and at that instant the alarm is required to sound. So that button press wouldn't disable the alarm. This is really rare.

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

There will be two switch inputs. In the main menu, the switches can be used to activate

1) set time;
2) set alarm;
3) turn 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.

The knob is in use in the scenes "Set Time" and "Set Alarm" when you select a section of the time. You can turn the knob to change the number and then hit the switch again to select the value.

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?

    The system uses 2 input switches, and 1 input ADC knob in order to navigate the menus. The menus allow the user to set the time and set an alarm time using said inputs. The system uses the systick timer to provide interrupts to keep track of the time on the clock. The time is then output to the ST7735 in a numerical display and in an analog format using trigonometric functions to gather an angle for the time. When the alarm time is reached, a flag is set that lets the systick interrupt reach a section of the handler that toggles a voltage on Port D that is hooked up to a speaker. This results in a square wave that ends up sounding the speaker. This is turned off by one of the switches in the main scene.

    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.

    The preparation includes having each individual module of the project built, including time management, the alarm, the ST7735 display, and the input devices (buttons and ADC knob)

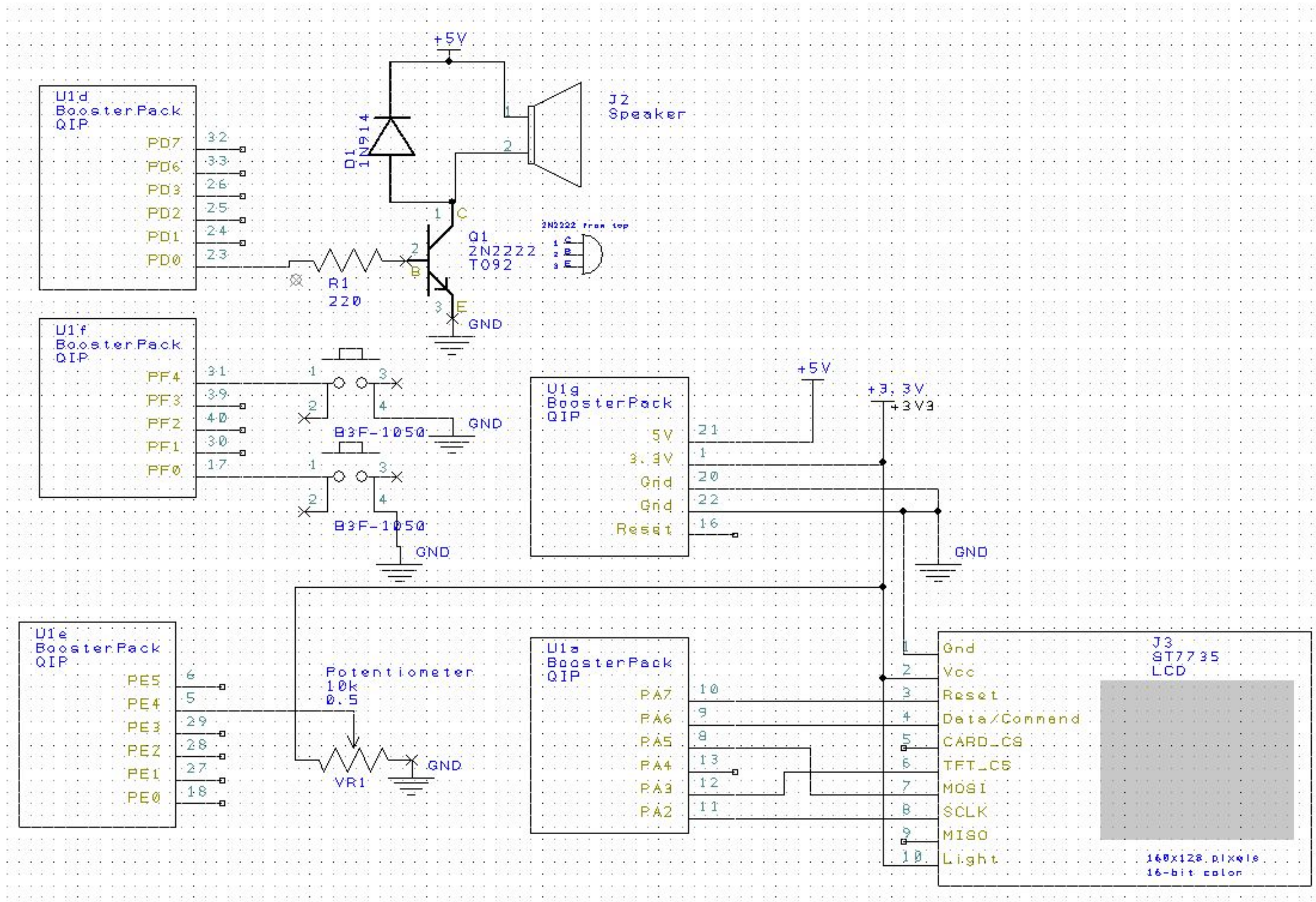
    The application of these modules includes putting them together into an interface and providing any additional features to make the system coherent. For our system, this includes menus with navigation tied to the buttons and ADC knob.

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

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

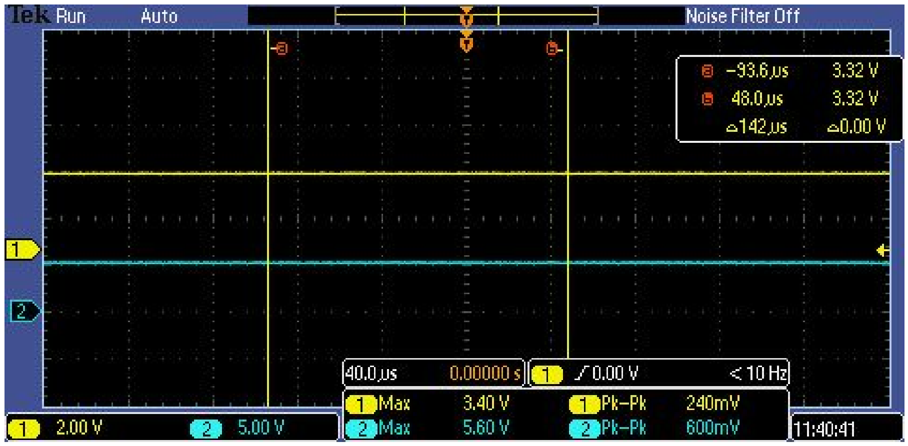
    The preparation was submitted on canvas before the lab session, and we demonstrated our project features to our TA during the lab. The report will be finalized and submitted for review on Friday, 9/29

Hardware Design

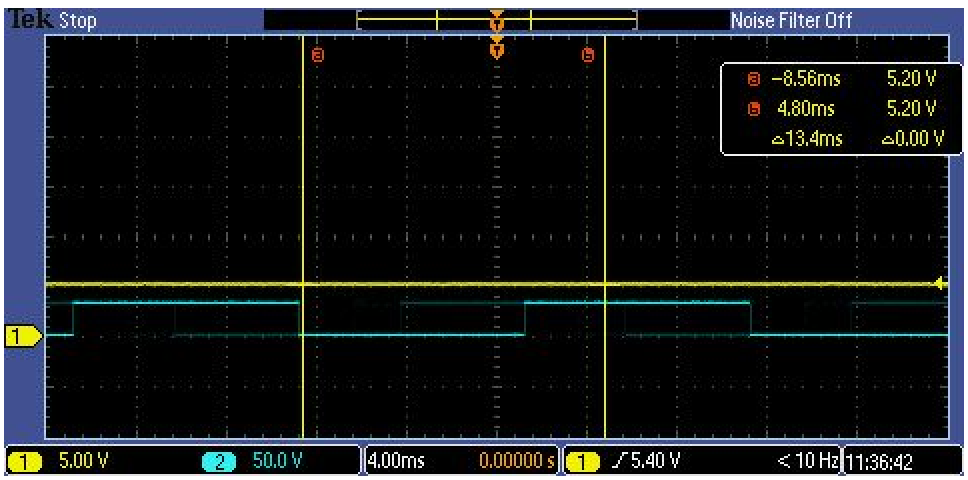


Measurement Data

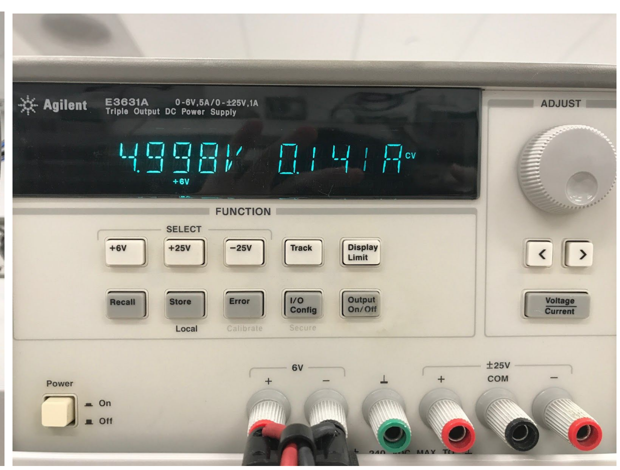
1. The 3.3 V voltage supply on board is measured to be around 3.40V and the 5.0V supply is measured to be 5.60V. The voltage ripple at 3.40V voltage supply is around 240mV peak-to-peak amplitude, which is equivalent to 84.85 mV RMS. The voltage ripple at 5.60V voltage supply is around 600mV peak-to-peak is equivalent to 212.13 mV RMS.



2. Plot the speaker voltage (or output voltage) versus time during an alarm sound



3. When powered with external 5V DC, the current required to run the alarm clock system is recorded. When the alarm is off, the current required is 103 mA. When the alarm is on, the current required is 141 mA.



Analysis and Discussion

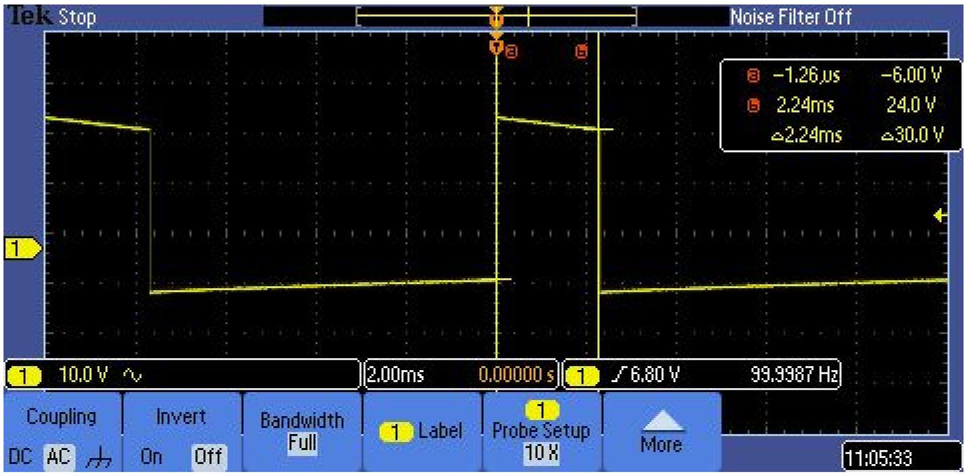
1) Give two ways to remove a critical section.

Make the operation atomic: disable interrupt right before the critical section and re-enable after it.

Remove sharing of variable: do not use global variables

2) How long does it take to update the LCD with a new time?

It takes 2.24ms to update the LCD with new time and draw the new set of hands



3) What would be the disadvantage of updating the LCD in the background ISR?

Updating the LCD inside an ISR increases the latency and will cause the system to be not real-time. We wish the ISR to execute quickly, because inside that ISR, it may not allow other interrupts to occur. Say the systick ISR lasts longer than the the systick interrupt interval, then it messes the time updating process.

4) Did you redraw the entire clock for each output? If so, how could you have redesigned the LCD update to run much faster, and create a lot less flicker?

We only draw the new hands and erase the old hands (by drawing it with background color). We do not clear the entire screen, which will make the output a lot slower.

5) Assuming the system were battery powered, list three ways you could have saved power.

1. Turns off the screen after a certain time of inactivity.
2. Update the LCD less frequently (which we already did).
3. The alarm turns off automatically after a certain duration, if not manually turned off.