Lab 3 Report

Alarm Clock

# Requirements Document

# Hardware Design

This section will cover the hardware for the alarm clock system. There will be a PCBArtist circuit drawing of the system with descriptions of all the important components. The following figure shows the circuit diagram for the alarm clock system:

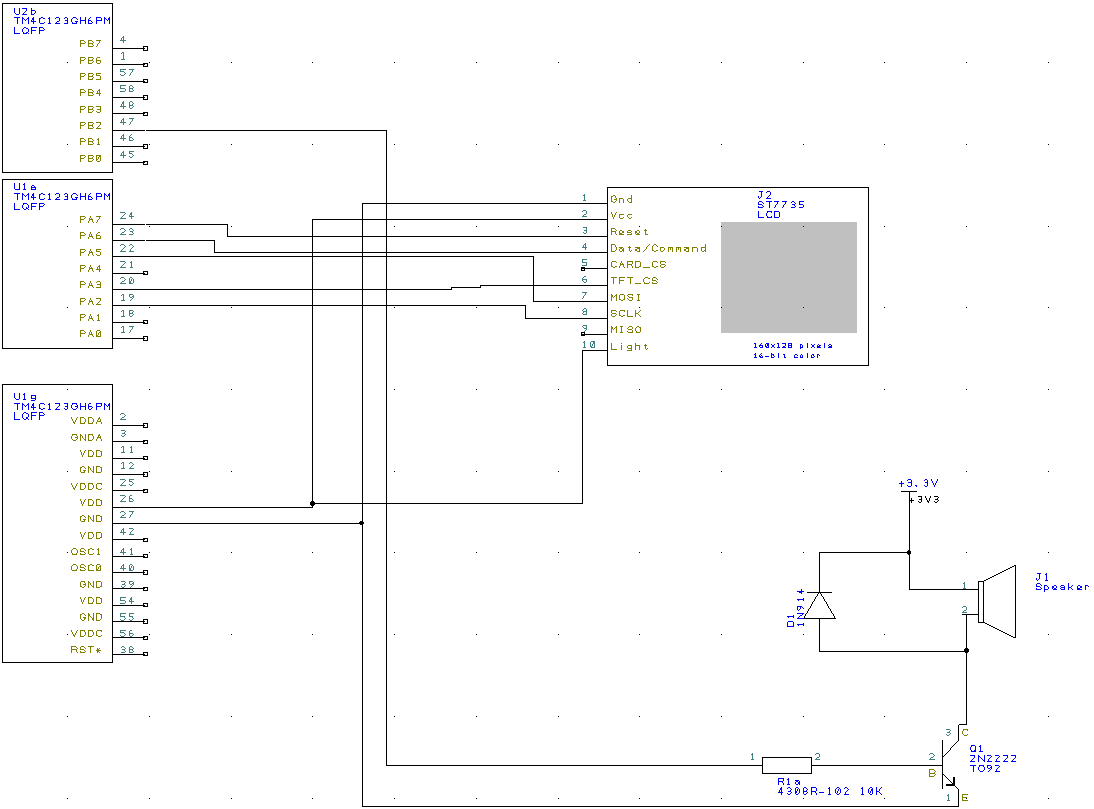


Figure 1: Circuit Diagram

The ST7735 display is used as a graphical output to the user for the system. The digital time and graphical clock will both be displayed using this screen while the system is running. The screen is updated in an ISR right after the time is updated. The connections from the screen to the board are the standard connections as described in ST7735.c and ST7735.h for the screen to work properly.

The speaker component is used to generate sound whenever the alarm goes off. The speaker is connected to VCC and the collector of a transistor. A diode is connected over the speaker to ensure that no backwards current goes through the speaker because this would create inductance and raise the voltage. The transistor is then connected to GPIO pin PB2. A square wave is generated by toggling PB2 at a certain frequency thus causing the speaker to generate a tone. The transistor acts like a switch, and sending current through the base of the transistor causes the switch to flip. Doing this at a certain frequency is what creates the square wave.

# Software Design

This section will cover the software for the alarm clock system by showing the software diagram and call graphs for the different modules.

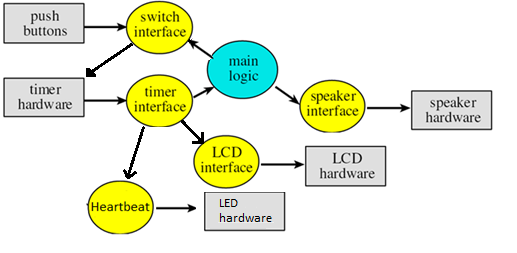


Figure 2: Software Diagram

Figure 2 shows a software diagram for the alarm clock system. The rectangles represent hardware components to the system while the ovals represent software modules. The on-board buttons are used for user input, and the main program will repeatedly check the user inputs by using the switch interface to check the button values. The switch interface uses the timer hardware to debounce the on-board switches. The value of the switch has to remain constant for a certain amount of time, and this time is measured using the timers.

The timer interface is the module that keeps track of the current time, and increments it every second. This is done using the on-board timer hardware. The timer interface also toggles the heartbeat for the system. After incrementing the time, the timer interface will then use the LCD interface to update the screen. This adds undesirable time to the ISR execution, but it solves our problem of a critical section. By having the same thread increment the time and update the LCD with the new time, there is never an opportunity for multiple threads to access the shared variable.

The speaker interface will toggle PB1 at a certain frequency to generate a sound on the speaker hardware. Different frequencies correspond to different sounds.