CSE 2240 LCD Mini Group Project

Team Members:
Arely Alcantara
Dominic Priolo
Madeline Hamilton
Megan Scott

Course Number:
CSE 2240, Spring 2017

Submitted Date:
May 7, 2017

1. General Project Description

LCDs, or liquid crystal displays, are devices that produce images utilizing a combination of a backlight and tiny filters controlled by an electric current. They are increasingly becoming the preferred method of image display for several reasons. LCDs can be made in almost any size or shape, do not use much power compared to competitors, emit very little heat, are not affected by magnetic fields, and are very compact, thin, and light. For these reasons, numerous companies, including ones that make watches, smartphones, televisions, or indoor and outdoor signage, have switched from using CRTs (cathode ray tubes) or plasma to using LCDs when building their products, and so it is useful to study the ways in which LCDs are programmed and implemented. For this report in particular, the inner workings and usage of the LCD mounted on the MCBSTM32C ARM evaluation board will be outlined. We will use the RTX_Traffic project, whose main components are the RTX_Traffic.c file and the GLCD.h file, to gain insight on how the LCD works in the MCBSTM32C board.

2. LCD Functionality Implementation concepts (i.e. as an example Bar Graph, Counter, RTX_Blinky, Traffic Light etc)

RTX_Traffic is a traffic light controller that uses a start and end time to control traffic and pedestrian flow.

- GLCD.h assigns colors to hexadecimal numbers and builds the interface for setting the colors, displaying text, and interacting with the LCD. For example, the file equates the

color black to be the hexadecimal number 0x0000. Additionally, the file includes interface-related methods, including methods for setting the window size, setting the text color, displaying chars and strings and implementing bargraphs.

- Traffic.c implements the interface outlined in GLCD.h. It starts by defining the colors for the traffic light/lamp output (red, yellow, and green), the stop and walk lamp output. A setLights method is used to determine which lights from the traffic light should be on or off. It takes in 2 arguments: one that represents the light and another that represents whether the light is on or off. The method compares the light to red, yellow, green, stop, and walk, and displays a char corresponding to the appropriate action. Back in main, the LEDs are set, and registers are used for push buttons. The entire simulation is divided into 8 tasks:
  1) Initialization : this portion of the code initializes the other 7 tasks.
     This step is essential, since it allows all of the other tasks on the lists to perform by initializing them. The program would not run without it.
  2) Command : this portion shows the menu to the user and prompts for a response.
     This task prints the command menu to the user and gets a user's input. The user's input is then put into a switch statement that decides which portion of the code to run. In case D, the start time and end time are printed. The program then looks for the user to hit escape. If not, then the clock time is displayed. In case T, the program sets the time command. It reads the time input and stores them in ctime. In case E, the program sets the end time variable, reading the time input and storing it into end time variable. In case S, the program sets the start time variable. It reads the input and stores the time in the variable start. In the default case, the program simply prints the menu. This task is on an endless loop.
  3) Clock : this portion consists of an endless loop that checks for valid time input.
     This task establishes a clock and continuously calculates the time. In the endless loop, the program checks to see if the number of seconds has become 60. If so, then it sets the variable second back to 0 and checks if minutes is equal to 60. If so, then minutes goes back to 0 and hours are checked against 24. If hours is equal to 24, then hours is reset to 0. All of this code is contingent on the first if statement, which checks to see if second is 60. The clock also checks if the time has changed. Lastly, the clock has a 100 millisecond time delay.
  4) Blinking : this task controls the yellow light that is on between the red "stop" light and the green "go" light. The yellow light turns on for 30 ticks, then turns off for 30 ticks. After blinking, the program goes to the main light controls, which control the stop and go lights.

5) Lights : In this task, the program turns the various lights on and off for specific times. The red light turns on for 50 ticks, starts blinking before the lights turn off, and waits 50 more ticks. The green light is then turn on for 50 ticks, then the yellow light will be turned on and the green lights off. Then, the red red light is turned on for 50 ticks, and the walk light for pedestrians is turned on. After 50 ticks, the walk light is turned off and the stop light for pedestrians is displayed.

6) Keyread : When a pedestrian presses a button, this portion of the code sends a message to the “Lights” task, which triggers the pedestrian walk light to turn on when the red light for cars is turned on.

7) Escape : When the ESC key is pressed, the program is terminated.

8) LCD : This sets the background color to blue the entire time the program is running and always shows the text "MCBSTM32C Demo", "RTX Traffic", and "www.keil.com". The text will be white for 400 ticks and then red for 400 ticks. The text represents the purpose of the program, to demo what the LCD board can do, the title of the program, and the Keil website.

3. Task Division

- Obtain files from computer lab and find files to analyze (files that use LCD) - Arely
- How can the LCD be implemented in real-life situations research/implementation -Madeline
- Analyze/research about LCD -all
- Use RTX_Traffic to see how LCD is implemented -all
- Look through GLCD.h and summarize what’s going on, outline Traffic.c for others to analyze -Arely
- Look through Traffic.c and explain/summarize/elaborate on how LCD is implemented -Dominic, Megan
- Proofread/review report -Madeline

4. Project Organization

This section of the project describes the external entities (Instructor/TA) to the project as well as internal structure of the organization and roles and relevant responsibilities
The roles and duties are listed as follows:

S.N	Name	Roles	% Participation	Responsibilities
1	Arely Alcantara	Project Leader	25%	- Take initiative in leading the term project - Delegate responsibilities and collaborate with others
2	Madeline Hamilton	Researcher/Writer	25%	- Research common applications of the LCD - Proofread/review report
3	Dominic Priolo	Software Reviewers	25%	- Look at RTX_Traffic and analyze how the LCD is implemented in MCBSTM32C board
4	Megan Scott		25%	- Go through traffic.c and research more about the registers used - Contribute to report

Table 1: Roles/Responsibilities