Final Project: Frogger Game on

LandTiger Board

University at Buffalo, The State University at New York

EE379 Spring 2019

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Written By** | Adarsh Govindarajan |  | **Revision History** | | | |
| **Engineer** | Shreya Puranik |  | **Rev** | **Date** | **Session** | **Approved/Grade** |
| **Engineer** | Prankur Sinha |  | Orig | 5/10/19 | Fri. 12PM |  |
| **Engineer** | Chitvan Chauhan |  |  |  |  |  |
| **Teaching Assistant** | Lalit Kishore Marepalli |  |  |  |  |  |
| **Student Assistant** | Paul Dehm |  |  |  |  |  |
|  |  |  |  |  |  | **EE379S19**  **Final Project** |

# Table of Contents

[Table of Contents 2](#_Toc505075449)

[1. Introduction 2](#_Toc505075451)

[1.1 Overview 2](#_Toc505075452)

[1.2 Document Scope 2](#_Toc505075453)

[1.3 Intended Audience 2](#_Toc505075454)

[2. Software Design 3](#_Toc505075456)

[2.1.1 Function main() 3](#_Toc505075459)

[2.1.2 Function void drawSquare(int cx, int cy, int width, unsigned short color) 4](#_Toc505075460)

[3. Testing and Verification 4](#_Toc505075461)

[3.1 Objective Verification 4](#_Toc505075462)

# Introduction

## Overview

This system implements an embedded system that displays the designed Frogger game on the LCD Display on the LandTiger LPC1768 development board and enables the user to play it. The software was developed in ARM Assembly language using the Keil uVision Integrated Development Environment. The objective of this project was to implement the frogger game on the LCD display of the board and play the game using the joystick control present on the board. The output of this system was tested and visualized on the development board, using the LCD Display and the Joystick.

## Document Scope

The scope of this document is to provide an accurate description of the Final Project Frogger game implementation design. This document discusses the API documentation of the software, and testing on the embedded system development board.

## Intended Audience

The audience for this document is the EE379 session TAs and professor, as well as the lab partners in the Friday 12PM session.

# Software Design

## Global Variables:

|  |  |
| --- | --- |
| Definition and Initialization | int carsX[3][3] = {{37,130,210}, {47,150, 230}, {60, 170, 220}};  int carsY[3][3] = {{23,116,196}, {33,136,216}, {46, 156, 206}};  int trucksX[2][2] = {{80,170}, {20, 150}};  int trucksY[2][2] = {{130,220}, {70, 200}};  int logsX[2][3] = {{50,160, 200}, {50, 120,180}};  int logsY[2][3] = {{90,200, 240}, {90, 160,240}};  int turtlesX1[3][4] = {{30,90,150,210}, {50, 130,210,240}, {70, 200,210,240}};  int turtlesX2[3][4] = {{38,98,158,218}, {58, 138,218,240}, {78, 208,218,240}};  int turtlesY[3][4] = {{22,82,142,202}, {42, 122,202,232}, {62, 192,202,232}};  int homes[1][5] = {{30, 75, 130, 180, 220}}; |
| Description | These Global Variables initialize the values of pixels for each of the element of the game being displayed on the LCD screen. These are two dimensional arrays holding the locations of these elements (or pixels) on the LCD display. |

## Function move\_carsX (int vx)

|  |  |
| --- | --- |
| Return type | void |
| Parameters | int vx |
| Description | This function adds the moving functionality to the white cars which are already displayed on the screen. There are two nested for loops which change the location of the pixels of the white colored cars with every iteration. It also checks that if the white cars reach the defined bound, the function will generate new cars from the other side. |

## Function move\_carsY (int vx)

|  |  |
| --- | --- |
| Return type | void |
| Parameters | int vx |
| Description | This function adds the moving functionality to the black cars which are already displayed on the screen. These black cars move behind the white cars, changing the color of the pixels behind to black (color of the background). There are two nested for loops which change the location of the pixels of the white colored cars with every iteration. It also checks that if the black cars reach the defined bound, the function will generate new cars from the other side. |

## Function move\_trucksX (int vx)

|  |  |
| --- | --- |
| Return type | void |
| Parameters | int vx |
| Description | Functionality similar to function move\_carsX (int vx). |

## Function move\_trucksY (int vx)

|  |  |
| --- | --- |
| Return type | void |
| Parameters | int vx |
| Description | Functionality similar to function move\_carsY (int vx). |

## Function move\_logsX (int vx)

|  |  |
| --- | --- |
| Return type | void |
| Parameters | int vx |
| Description | Functionality similar to function move\_carsX (int vx). |

## Function move\_logsY (int vx)

|  |  |
| --- | --- |
| Return type | void |
| Parameters | int vx |
| Description | Functionality similar to function move\_carsY (int vx). |

## Function move\_turtlesX1 (int vx)

|  |  |
| --- | --- |
| Return type | void |
| Parameters | int vx |
| Description | Functionality similar to function move\_carsX (int vx). The Turtles X1 defines the green portion of the turtle. |

## Function move\_turtlesX2 (int vx)

|  |  |
| --- | --- |
| Return type | void |
| Parameters | int vx |
| Description | Functionality similar to function move\_carsX (int vx). The Turtles X1 defines the brown portion of the turtle. |

## Function move\_turtlesY (int vx)

|  |  |
| --- | --- |
| Return type | void |
| Parameters | int vx |
| Description | Functionality similar to function move\_carsY (int vx). |

## Function move\_frog (int vx)

|  |  |
| --- | --- |
| Return type | void |
| Parameters | int vx |
| Description | This function adds the functionality of toggling the frog using the joystick with the use of pointers. The pointers are updating the x and y locations of the frog (variables following with cx and cy) as we toggle the joystick to move up, down, right or left. |

## Function draw\_frog (int cx, int cx, int cy, int width, int length, unsigned short color)

|  |  |
| --- | --- |
| Return type | void |
| Parameters | int cx, int cy, int width, int length, unsigned short color |
| Description | This function is drawing the frog on the LCD display with the given color in parameter and is basically a square. The function uses nested loops to draw the pixels on the screen. |

## Function draw\_car (int cx, int cx, int cy, int width, int length, unsigned short color)

|  |  |
| --- | --- |
| Return type | void |
| Parameters | int cx, int cy, int width, int length, unsigned short color |
| Description | This function is drawing the car on the LCD display with the given color in parameter and is basically a square. The function uses nested loops to draw the pixels on the screen. |

## Function draw\_safezone (int cx, int cx, int cy, int width, int length, unsigned short color)

|  |  |
| --- | --- |
| Return type | void |
| Parameters | int cx, int cy, int width, int length, unsigned short color |
| Description | This function is drawing the safe zone and the background on the LCD display with the given color in parameter and is basically a square. The function uses nested loops to draw the pixels on the screen. |

## Function draw\_truck (int cx, int cx, int cy, int width, int length, unsigned short color)

|  |  |
| --- | --- |
| Return type | void |
| Parameters | int cx, int cy, int width, int length, unsigned short color |
| Description | This function is drawing the truck on the LCD display with the given color in parameter and is basically a square. The function uses nested loops to draw the pixels on the screen. |

## Function draw\_logs (int cx, int cx, int cy, int width, int length, unsigned short color)

|  |  |
| --- | --- |
| Return type | void |
| Parameters | int cx, int cy, int width, int length, unsigned short color |
| Description | This function is drawing the frog on the LCD display with the given color in parameter and is basically a square. The function uses nested loops to draw the pixels on the screen. |

## Function draw\_turtle (int cx, int cx, int cy, int width, int length, unsigned short color)

|  |  |
| --- | --- |
| Return type | void |
| Parameters | int cx, int cy, int width, int length, unsigned short color |
| Description | This function is drawing the turtle on the LCD display with the given color in parameter and is basically a square. The function uses nested loops to draw the pixels on the screen. |

## Function draw\_homes (int cx, int cx, int cy, int width, int length, unsigned short color)

|  |  |
| --- | --- |
| Return type | void |
| Parameters | int cx, int cy, int width, int length, unsigned short color |
| Description | This function is drawing the home on the LCD display with the given color in parameter and is basically a square. The function uses nested loops to draw the pixels on the screen. |

## Function delay (int count)

|  |  |
| --- | --- |
| Return type | void |
| Parameters | int count |
| Description | This function is adding a delay. |

## 2.2.20 Function int main ()

|  |  |
| --- | --- |
| Return type | Void |
| Parameters | None |
| Description | This function is adding the main functionality to the game by calling the various functions defined above with the values as we need for implementing the design. |

# Testing and Verification

## Objective Verification

The functionality of the screen saver module was verified in deployment on the LandTiger LPC1768 development board. After programming the board using the Keil ulink device, the pixels in the LCD Display lighted up to form a square which moved from the center of the display. When the square reached the edge of the display, it bounced off and went in the opposite direction until it reached another edge. This process kept on repeating, as required.