CECS 447 Spring 2021 Project 4

# ST7735R Color LCD

By

Jesus Franco

4/19/2021

This project will show an animated Elmo that is shooting lazers from its mouth on the ST7735 LCD display.

**Introduction:**
Project 4's sole purpose is to display a user's imagination of a quick animation on the ST7735 LCD display. For this project, an Elmo that is shooting lazers from its mouth was the determined imagined animated thought that became reality and was portrayed on the LCD screen.

**Port Table:**

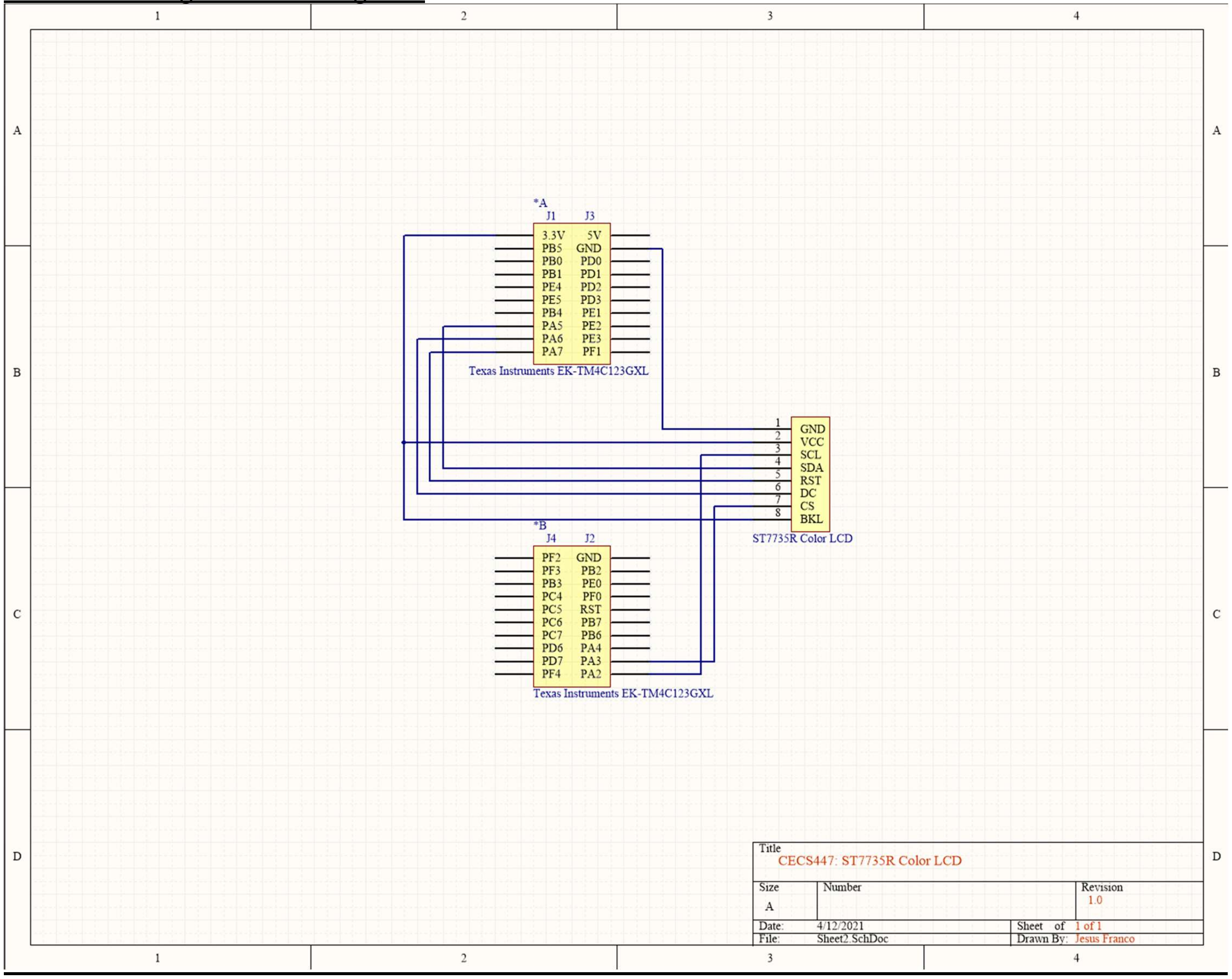
| Name | I/0 | Port | Description |
|---|---|---|---|
| ST7735 LCD SCK | Output | PA2 | A 50% duty cycle clock generated by the master. |
| ST7735 LCD SDA | Output | PA5 | A I2C bus serial data line. |
| ST7735 LCD A0 | Output | PA6 | Data/Command. |
| ST7735 LCD RESET | Output | PA7 | Reset. |
| ST7735 LCD CS | Output | PA3 | Chip enable. |

**Operation:**

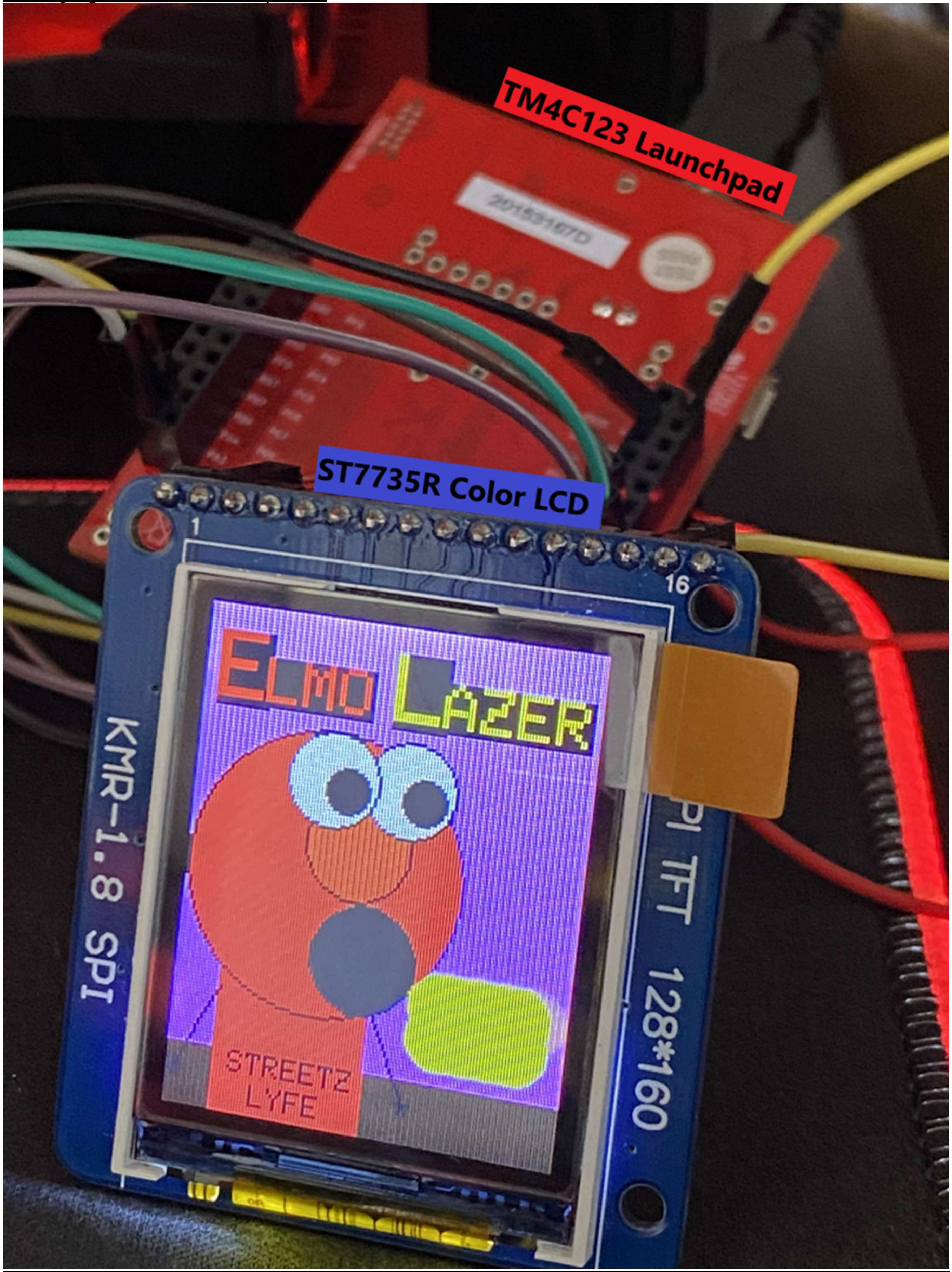
The process of creating this project is to create a figure with at least one animated object. A stationary Elmo is placed with its mouth open. The reason why its mouth is open is because Elmo will be shooting lazers from its mouth. The animated object will be the lazer that will constantly be shooting out of Elmo. Firstly, to create Elmo, a bunch of circles had to be placed to create Elmo's face. A rectangle was used for Elmo's body and multiple lines such as horizontal, vertical, and diagonal lines were used to make Elmo's arms and fingers. The background was filled to be a nice shade of pink and the ground was filled to be a brown rectangle, so it seemed like Elmo was in a room with pink walls and brown flooring. The lazers that are shooting from Elmo's mouth are 2 circles that are being fired together and seem parallel. As soon as the lazers reach the end of the LCD screen, 2 other circles that are the same color as the background will fill the beginning circles, so it seems like Elmo has stopped firing lazers for a bit. This will give an animated display of a cute Elmo figure shooting lazers continuously.

**Theory:**
In theory, this project was simply just calling functions. The hardest part about this project was limiting my imagination to think of a small animation that consists of just circles, rectangles, and lines. After an idea has been set, the rest was simply just positioning to find the best preferable location to place all the shapes and lines to get the right image. The debugger was mainly used for debugging since the TExaS sitronix peripheral did all the heavy lifting while repositioning objects to the correct coordinates. After the perfected image has been set up, the circuit building was the final concluding factor; however, the circuit was fairly simple since it only required a few wires to be connected directly to the Launchpad.

## Hardware Design/ Circuit Diagram:



**Photographs of Hardware System:**



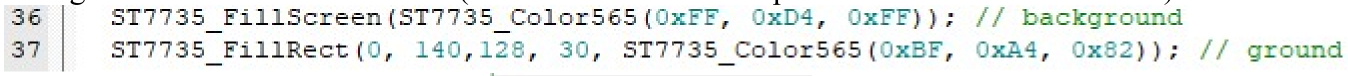
**Software Design:**

```
// Name: Jesus Franco
// Student ID: 014046368
// Course Number: CECS 447
// Assignment: Project 4: ST7735 Color LCD
// Description: This code will allow a colored image to be portrayed on the ST7735 LCD
display.
//                                          This code will have Elmo shooting out lazers
out of its mouth using the functions
//                                          from the ST7735.h where Elmo will be stationary
and the lazers will be moving in
//                                          one direction.

// Backlight (pin 10) connected to +3.3 V
// MISO (pin 9) unconnected
// SCK (pin 8) connected to PA2 (SSI0Clk)
// MOSI (pin 7) connected to PA5 (SSI0Tx)
// TFT_CS (pin 6) connected to PA3 (SSI0Fss)
// CARD_CS (pin 5) unconnected
// Data/Command (pin 4) connected to PA6 (GPIO)
// RESET (pin 3) connected to PA7 (GPIO)
// VCC (pin 2) connected to +3.3 V
// Gnd (pin 1) connected to ground

#include <stdio.h>
#include <stdint.h>
#include "string.h"
#include "ST7735.h"
#include "PLL.h"
#include "tm4c123gh6pm.h"

void SysTick_Init(void); // SysTick initialization
void SysTick_Wait10ms(unsigned long delay); // SysTick 10ms delay
void ElmoLazer(void); // Elmo lazer animation

// This function will plot Elmo onto the LCD screen.
void ST7735_XYPlotElmo()
{
      // DRAW THE GROUND AND BACKGROUND
      ST7735_FillScreen(ST7735_Color565(0xFF, 0xD4, 0xFF)); // background
      ST7735_FillRect(0, 140,128, 30, ST7735_Color565(0xBF, 0xA4, 0x82)); // ground
      // DRAW TEXT "ELMO"
      ST7735_DrawCharS(4, 8, 'E', ST7735_RED, 0, 3);   // E
      ST7735_DrawCharS(20, 15, 'L', ST7735_RED, 0, 2); // L
      ST7735_DrawCharS(32, 15, 'M', ST7735_RED, 0, 2); // M
      ST7735_DrawCharS(44, 15, 'O', ST7735_RED, 0, 2); // O
      // DRAW TEXT "LAZER"
      ST7735_DrawCharS(62, 8, 'L', ST7735_YELLOW, 0, 3);     // L
      ST7735_DrawCharS(78, 15, 'A', ST7735_YELLOW, 0, 2);  // A
      ST7735_DrawCharS(90, 15, 'Z', ST7735_YELLOW, 0, 2);  // Z
      ST7735_DrawCharS(102, 15, 'E', ST7735_YELLOW, 0, 2); // E
      ST7735_DrawCharS(114, 15, 'R', ST7735_YELLOW, 0, 2); // R
      // DRAW ELMO BODY
      ST7735_FillRect(15, 80, 50, 160,ST7735_RED); // body
      // DRAW ELMO LEFT ARM
      ST7735_DrawLine(15, 120, 1, 150, ST7735_BLACK); // left arm
      ST7735_DrawLine(5, 147, 0, 147, ST7735_BLACK);  // horizontal left finger
      ST7735_DrawLine(3, 145, 3, 150, ST7735_BLACK);  // vertical left finger
      // DRAW ELMO RIGHT ARM
      ST7735_DrawLine(65, 120, 79, 150, ST7735_BLACK); // right arm
      ST7735_DrawLine(75, 147, 80, 147, ST7735_BLACK); // horizontal right finger
      ST7735_DrawLine(77, 145, 77, 150, ST7735_BLACK); // vertical right finger
      // DRAW TEXT "STREETZ"
```

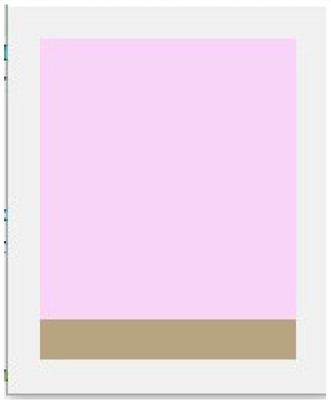
```c
      ST7735_DrawCharS(20, 140, 'S', ST7735_BLACK, 0, 1); // S
      ST7735_DrawCharS(26, 140, 'T', ST7735_BLACK, 0, 1); // T
      ST7735_DrawCharS(32, 140, 'R', ST7735_BLACK, 0, 1); // R
      ST7735_DrawCharS(38, 140, 'E', ST7735_BLACK, 0, 1); // E
      ST7735_DrawCharS(44, 140, 'E', ST7735_BLACK, 0, 1); // E
      ST7735_DrawCharS(50, 140, 'T', ST7735_BLACK, 0, 1); // T
      ST7735_DrawCharS(56, 140, 'Z', ST7735_BLACK, 0, 1); // Z
      // DRAW TEXT "LYFE"
      ST7735_DrawCharS(28, 150, 'L', ST7735_BLACK, 0, 1); // L
      ST7735_DrawCharS(34, 150, 'Y', ST7735_BLACK, 0, 1); // Y
      ST7735_DrawCharS(40, 150, 'F', ST7735_BLACK, 0, 1); // F
      ST7735_DrawCharS(46, 150, 'E', ST7735_BLACK, 0, 1); // E
      // DRAW ELMO FACE
      ST7735_FillCircle(45, 80, 45,ST7735_RED);       // head
      ST7735_DrawCircle(45, 80, 45,ST7735_BLACK);

      ST7735_FillCircle(55, 70, 17,ST7735_ORANGE); // nose
      ST7735_DrawCircle(55, 70, 17,ST7735_BLACK);

      ST7735_FillCircle(70, 50, 15,ST7735_WHITE); // right eye
      ST7735_DrawCircle(70, 50, 15,ST7735_BLACK);
      ST7735_FillCircle(75, 53, 7,ST7735_BLACK); // right pupil

      ST7735_FillCircle(45, 50, 15,ST7735_WHITE); // left eye
      ST7735_DrawCircle(45, 50, 15,ST7735_BLACK);
      ST7735_FillCircle(50, 53, 7,ST7735_BLACK); // left pupil

      ST7735_FillCircle(60, 105, 17,ST7735_BLACK); // mouth
}

unsigned long erase_flag = 0; // flag to determine when to erase lazer
unsigned long laser_flag = 1; // flag to determine when to do lazer
uint32_t x_blast = 87; // initial value of lazer
uint32_t x_erase = 87; // initial value of erasing lazer

int main(void){
  PLL_Init(12);   // Initialize PLL to 30.769 MHz
  ST7735_InitR(INITR_REDTAB); // Initialize ST7735 LCD screen
      SysTick_Init(); // Initialize SysTick
      ST7735_XYPlotElmo(); // Plot Elmo
  while(1){
            ElmoLazer(); // Begin lazer animation
            SysTick_Wait10ms(1); // wait 10ms (assumes 30.769 MHz clock)
      }
}

// This function will do the animation of the lazer blasting.
void ElmoLazer(void){
      // reset to initial values
      if( x_blast >= 117 && x_erase >= 117){
            x_blast = 87;
            x_erase = 87;
      }
      // laser blasting
      if (laser_flag){
            ST7735_FillCircle(x_blast, 115, 10,ST7735_YELLOW); // yellow blast
            ST7735_FillCircle(x_blast, 125, 10,ST7735_YELLOW); // yellow blast
            x_blast = x_blast + 5;
            if(x_blast >= 122){
            laser_flag = 0;
            erase_flag = 1;
            }
      }
      // lazer erasing
      if (erase_flag){
            ST7735_FillCircle(x_erase, 115, 10,ST7735_Color565(0xFF, 0xD4, 0xFF)); //
erase blast
            ST7735_FillCircle(x_erase, 125, 10,ST7735_Color565(0xFF, 0xD4, 0xFF)); //
erase blast
            x_erase = x_erase + 5;
            if(x_erase >= 122){
                  laser_flag = 1;
                  erase_flag = 0;
            }
      }

}

// Initialize SysTick with busy wait running at bus clock.
void SysTick_Init(void){
  NVIC_ST_CTRL_R = 0;                   // disable SysTick during setup
  NVIC_ST_RELOAD_R = NVIC_ST_RELOAD_M;  // maximum reload value
  NVIC_ST_CURRENT_R = 0;                // any write to current clears it
                                        // enable SysTick with core clock
  NVIC_ST_CTRL_R = NVIC_ST_CTRL_ENABLE+NVIC_ST_CTRL_CLK_SRC;
}

// Time delay using busy wait.
// The delay parameter is in units of the core clock.
void SysTick_Wait(unsigned long delay){
  volatile unsigned long elapsedTime;
  unsigned long startTime = NVIC_ST_CURRENT_R;
  do{
    elapsedTime = (startTime-NVIC_ST_CURRENT_R)&0x00FFFFFF;
  }
  while(elapsedTime <= delay);
}

// Time delay using busy wait.
// This assumes 30.769 MHz system clock.
void SysTick_Wait10ms(unsigned long delay){
  unsigned long i;
  for(i=0; i<delay; i++){
    SysTick_Wait(307690);  // wait 10ms (assumes 30.769 MHz clock)
  }
}
```

**Requirements:**

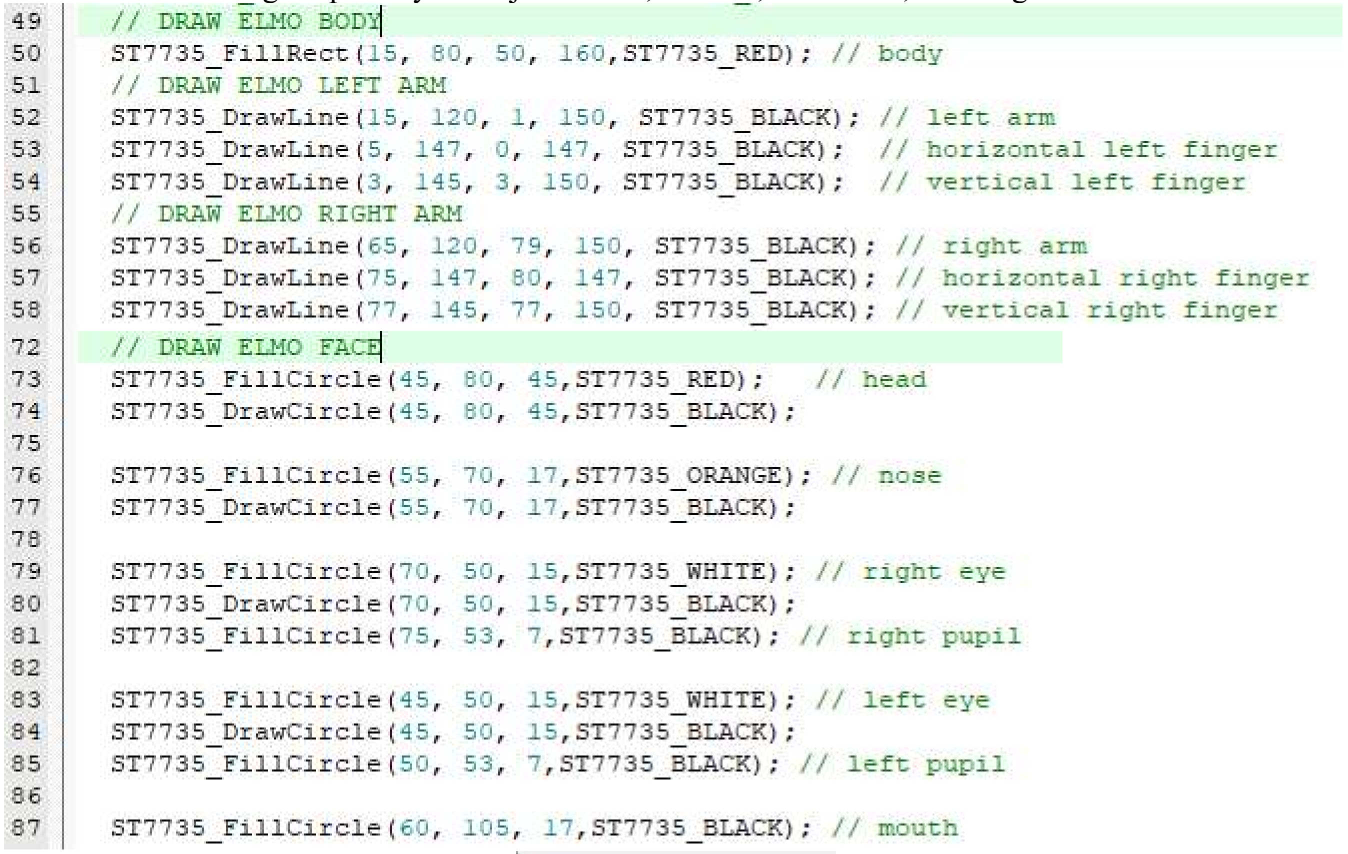
1. Background has at least two colors (different from the example code in both color and size).



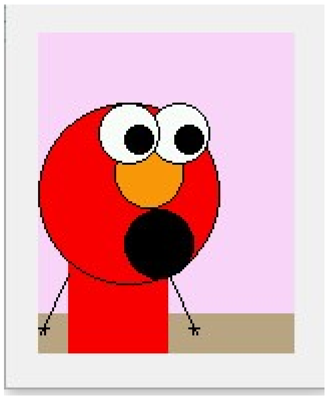
```
ST7735_FillScreen(ST7735_Color565(0xFF, 0xD4, 0xFF)); // background
ST7735_FillRect(0, 140,128, 30, ST7735_Color565(0xBF, 0xA4, 0x82)); // ground
```



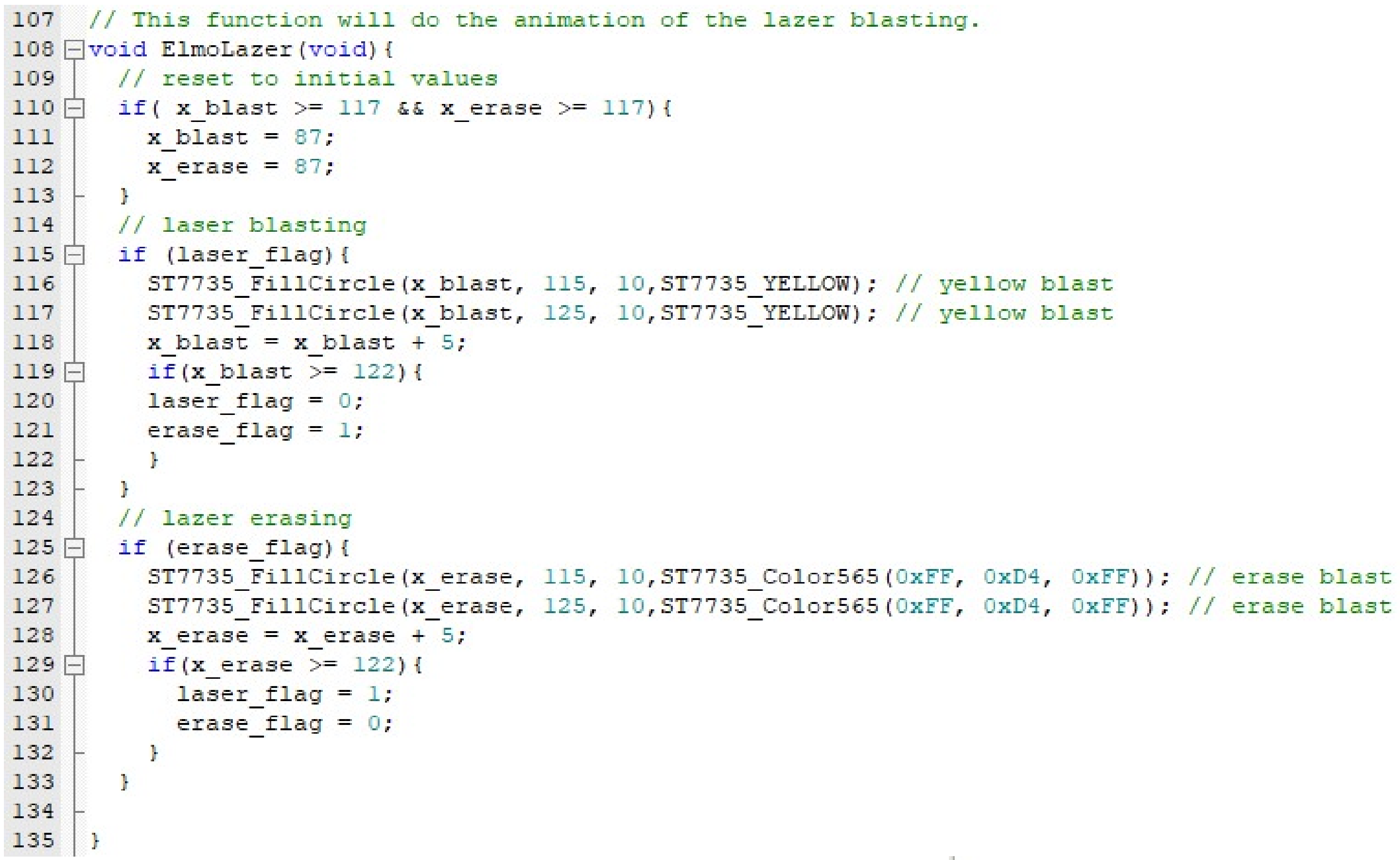
2. Include the following shapes in your object: circle, vertical, horizontal, and diagonal lines.



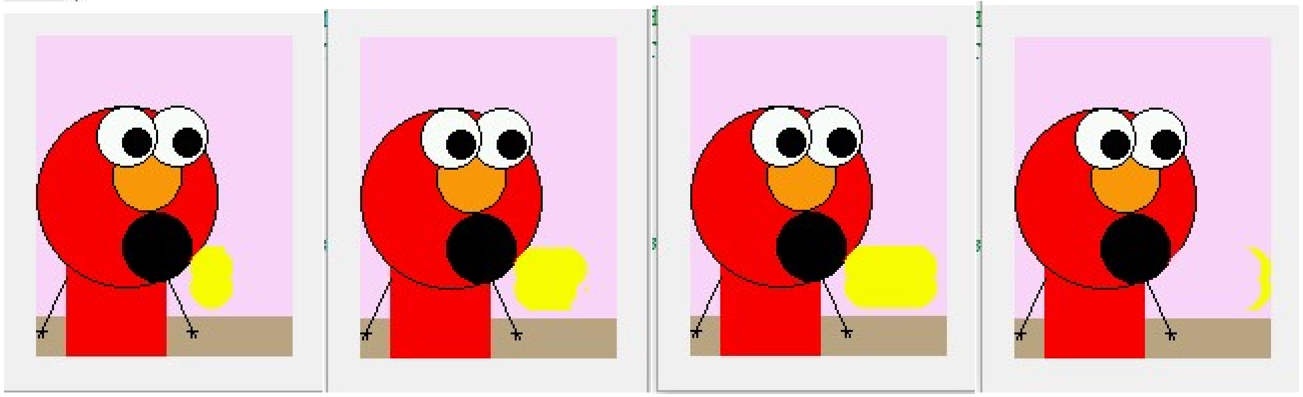
```
// DRAW ELMO BODY
ST7735_FillRect(15, 80, 50, 160,ST7735_RED); // body
// DRAW ELMO LEFT ARM
ST7735_DrawLine(15, 120, 1, 150, ST7735_BLACK); // left arm
ST7735_DrawLine(5, 147, 0, 147, ST7735_BLACK);  // horizontal left finger
ST7735_DrawLine(3, 145, 3, 150, ST7735_BLACK);  // vertical left finger
// DRAW ELMO RIGHT ARM
ST7735_DrawLine(65, 120, 79, 150, ST7735_BLACK); // right arm
ST7735_DrawLine(75, 147, 80, 147, ST7735_BLACK); // horizontal right finger
ST7735_DrawLine(77, 145, 77, 150, ST7735_BLACK); // vertical right finger
// DRAW ELMO FACE
ST7735_FillCircle(45, 80, 45,ST7735_RED);   // head
ST7735_DrawCircle(45, 80, 45,ST7735_BLACK);

ST7735_FillCircle(55, 70, 17,ST7735_ORANGE); // nose
ST7735_DrawCircle(55, 70, 17,ST7735_BLACK);

ST7735_FillCircle(70, 50, 15,ST7735_WHITE); // right eye
ST7735_DrawCircle(70, 50, 15,ST7735_BLACK);
ST7735_FillCircle(75, 53, 7,ST7735_BLACK); // right pupil

ST7735_FillCircle(45, 50, 15,ST7735_WHITE); // left eye
ST7735_DrawCircle(45, 50, 15,ST7735_BLACK);
ST7735_FillCircle(50, 53, 7,ST7735_BLACK); // left pupil

ST7735_FillCircle(60, 105, 17,ST7735_BLACK); // mouth
```



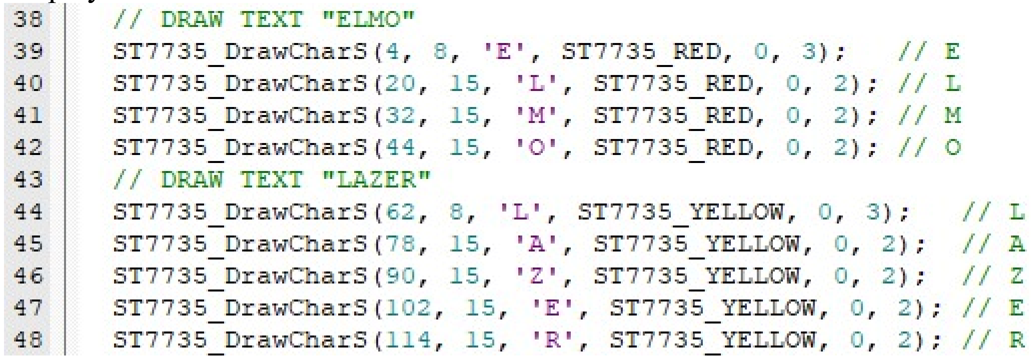
3. Has at least one moving object.



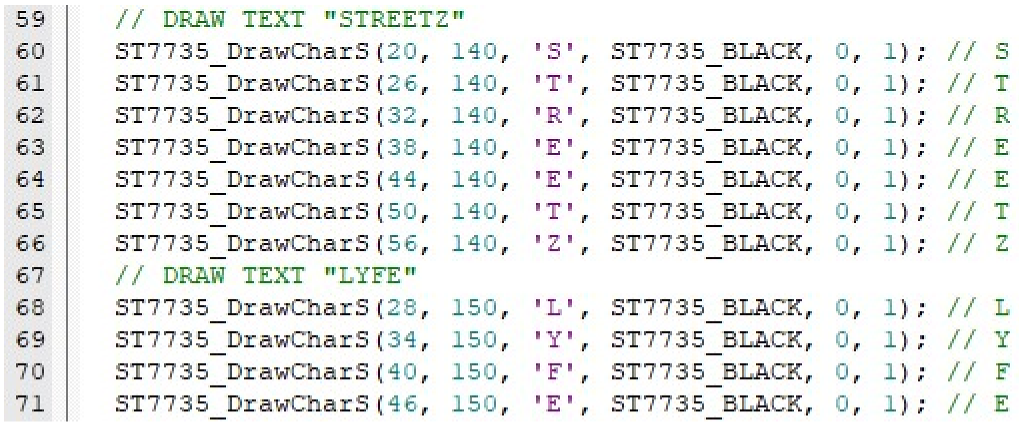
```
// This function will do the animation of the lazer blasting.
void ElmoLazer(void){
  // reset to initial values
  if( x_blast >= 117 && x_erase >= 117){
    x_blast = 87;
    x_erase = 87;
  }
  // laser blasting
  if (laser_flag){
    ST7735_FillCircle(x_blast, 115, 10,ST7735_YELLOW); // yellow blast
    ST7735_FillCircle(x_blast, 125, 10,ST7735_YELLOW); // yellow blast
    x_blast = x_blast + 5;
    if(x_blast >= 122){
    laser_flag = 0;
    erase_flag = 1;
    }
  }
  // lazer erasing
  if (erase_flag){
    ST7735_FillCircle(x_erase, 115, 10,ST7735_Color565(0xFF, 0xD4, 0xFF)); // erase blast
    ST7735_FillCircle(x_erase, 125, 10,ST7735_Color565(0xFF, 0xD4, 0xFF)); // erase blast
    x_erase = x_erase + 5;
    if(x_erase >= 122){
      laser_flag = 1;
      erase_flag = 0;
    }
  }

}
```



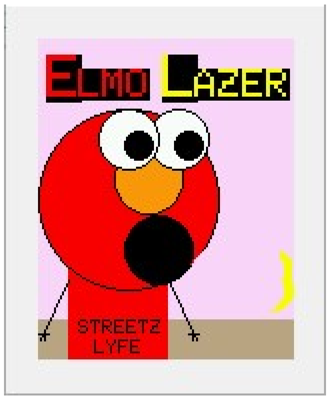
4. Display at least one line of color text with at least three different size and three different colors.



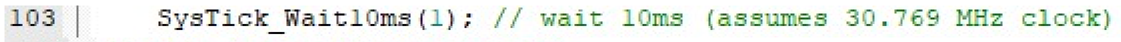
```
  // DRAW TEXT "ELMO"
  ST7735_DrawCharS(4, 8, 'E', ST7735_RED, 0, 3);   // E
  ST7735_DrawCharS(20, 15, 'L', ST7735_RED, 0, 2); // L
  ST7735_DrawCharS(32, 15, 'M', ST7735_RED, 0, 2); // M
  ST7735_DrawCharS(44, 15, 'O', ST7735_RED, 0, 2); // O
  // DRAW TEXT "LAZER"
  ST7735_DrawCharS(62, 8, 'L', ST7735_YELLOW, 0, 3);   // L
  ST7735_DrawCharS(78, 15, 'A', ST7735_YELLOW, 0, 2);  // A
  ST7735_DrawCharS(90, 15, 'Z', ST7735_YELLOW, 0, 2);  // Z
  ST7735_DrawCharS(102, 15, 'E', ST7735_YELLOW, 0, 2); // E
  ST7735_DrawCharS(114, 15, 'R', ST7735_YELLOW, 0, 2); // R
```



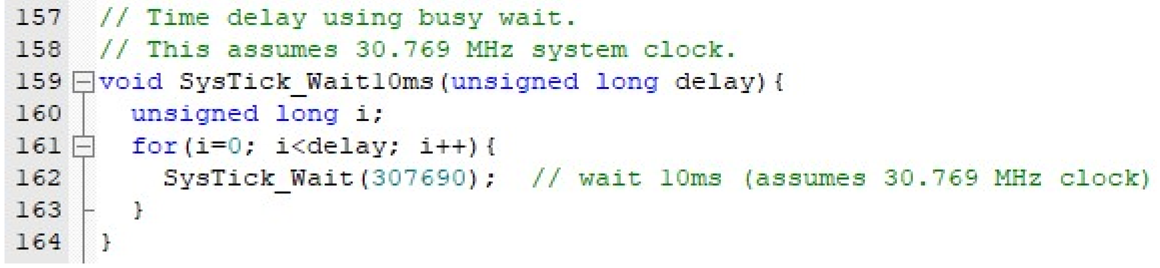
```
// DRAW TEXT "STREETZ"
ST7735_DrawCharS(20, 140, 'S', ST7735_BLACK, 0, 1); // S
ST7735_DrawCharS(26, 140, 'T', ST7735_BLACK, 0, 1); // T
ST7735_DrawCharS(32, 140, 'R', ST7735_BLACK, 0, 1); // R
ST7735_DrawCharS(38, 140, 'E', ST7735_BLACK, 0, 1); // E
ST7735_DrawCharS(44, 140, 'E', ST7735_BLACK, 0, 1); // E
ST7735_DrawCharS(50, 140, 'T', ST7735_BLACK, 0, 1); // T
ST7735_DrawCharS(56, 140, 'Z', ST7735_BLACK, 0, 1); // Z
// DRAW TEXT "LYFE"
ST7735_DrawCharS(28, 150, 'L', ST7735_BLACK, 0, 1); // L
ST7735_DrawCharS(34, 150, 'Y', ST7735_BLACK, 0, 1); // Y
ST7735_DrawCharS(40, 150, 'F', ST7735_BLACK, 0, 1); // F
ST7735_DrawCharS(46, 150, 'E', ST7735_BLACK, 0, 1); // E
```



5. Your are required to use SysTick timer to control time delay: replace DelayWait10ms() with SysTick timer.



```
    SysTick_Wait10ms(1); // wait 10ms (assumes 30.769 MHz clock)
```



```
// Time delay using busy wait.
// This assumes 30.769 MHz system clock.
void SysTick_Wait10ms(unsigned long delay){
  unsigned long i;
  for(i=0; i<delay; i++){
    SysTick_Wait(307690);  // wait 10ms (assumes 30.769 MHz clock)
  }
}
```

**Conclusion:**

For this project, it only required a limited imagination to create short animation for the color LCD display. This project was fun since it was fairly simple and allowed us to use our imagination to create a short and fun animation. The ST7735R LCD is very similar to the Nokia besides that it uses a lot more memory since it is colored and the Launchpad is limited to 32k bytes; thus, the reason why the extra credit requires a SD card to function.