HW#3: Mini Project

Roshan Baby Reji

Person #: 50205771

Introduction:

In this lab we will be designing a game in which there will be an UFO which can shoot grenades. Features of this UFO are restricted according to the criteria given. Only a maximum of three grenades will be displayed at any given time, irrespective of how many times the user might press the shoot button. We will also discuss the use of speaker on the board to generate sounds when grenades are released or they disappear.

Design Criteria and Restrictions:

We have taken each pixel in this diagram as 4X4 pixels box and represented the given shapes on the graphic display. The colors were given accurately according the assignment question.

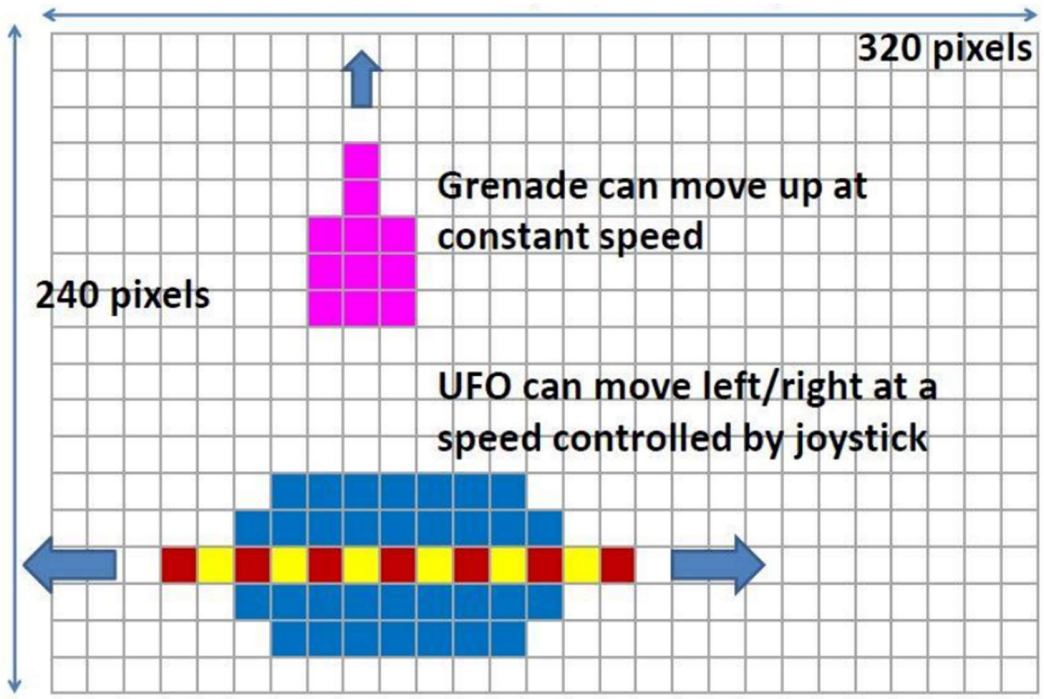


Figure: Design Criteria

We had to develop a simple application that uses the LCD display on the Land-Tiger board. The display must be is always on black background. On the lower edge on the display a UFO s according to the feature criteria must be designed. It should be able to move to the left or to the right according to the control from the Joystick on the board. When the INT0 push button is pressed the UFO should shoot a grenade, which moves up at constant speed until it hits the upper edge of the display and disappears. Only a maximum of three grenades should be displayed at any given time, irrespective of how many times the user might press the button. After three grenades have released, the program flow must wait until one or more grenades have

disappeared from the screen before launching the next grenade. Movements of all objects on canvas/display are updated with the help of Timer 0 interrupts. Use of speaker to shown to generate sounds when grenades are released or they disappear.

Program Output:

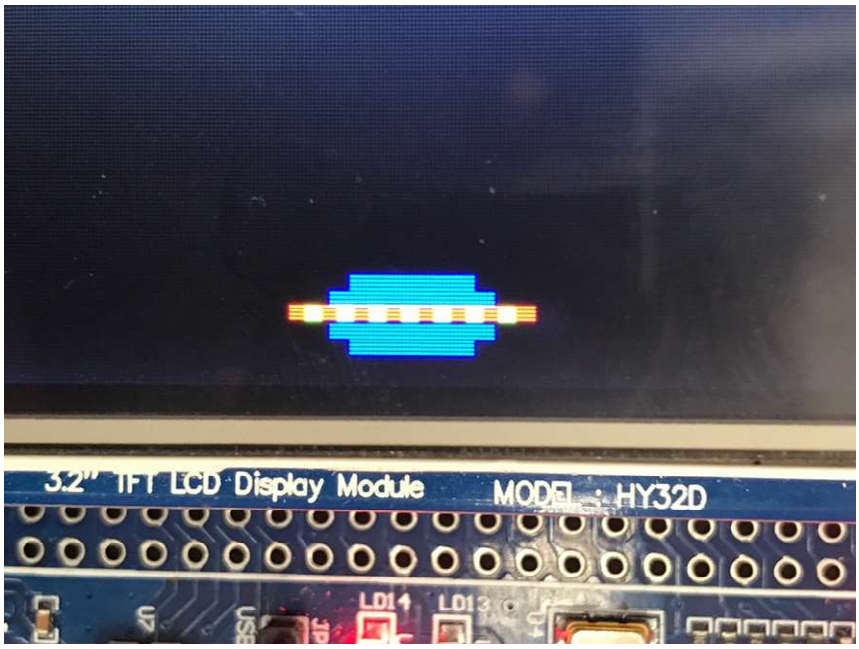


Figure: Image of the UFO

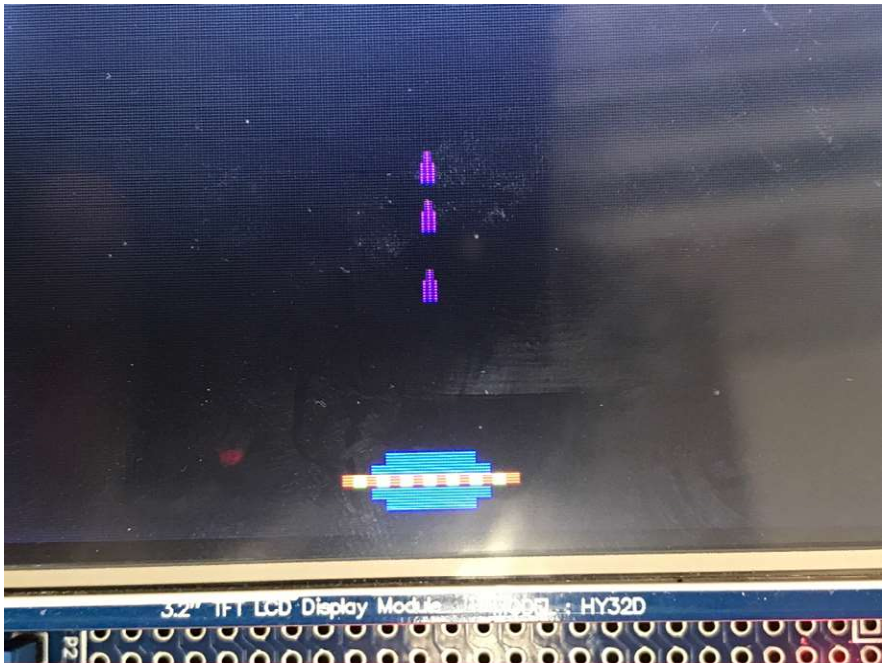
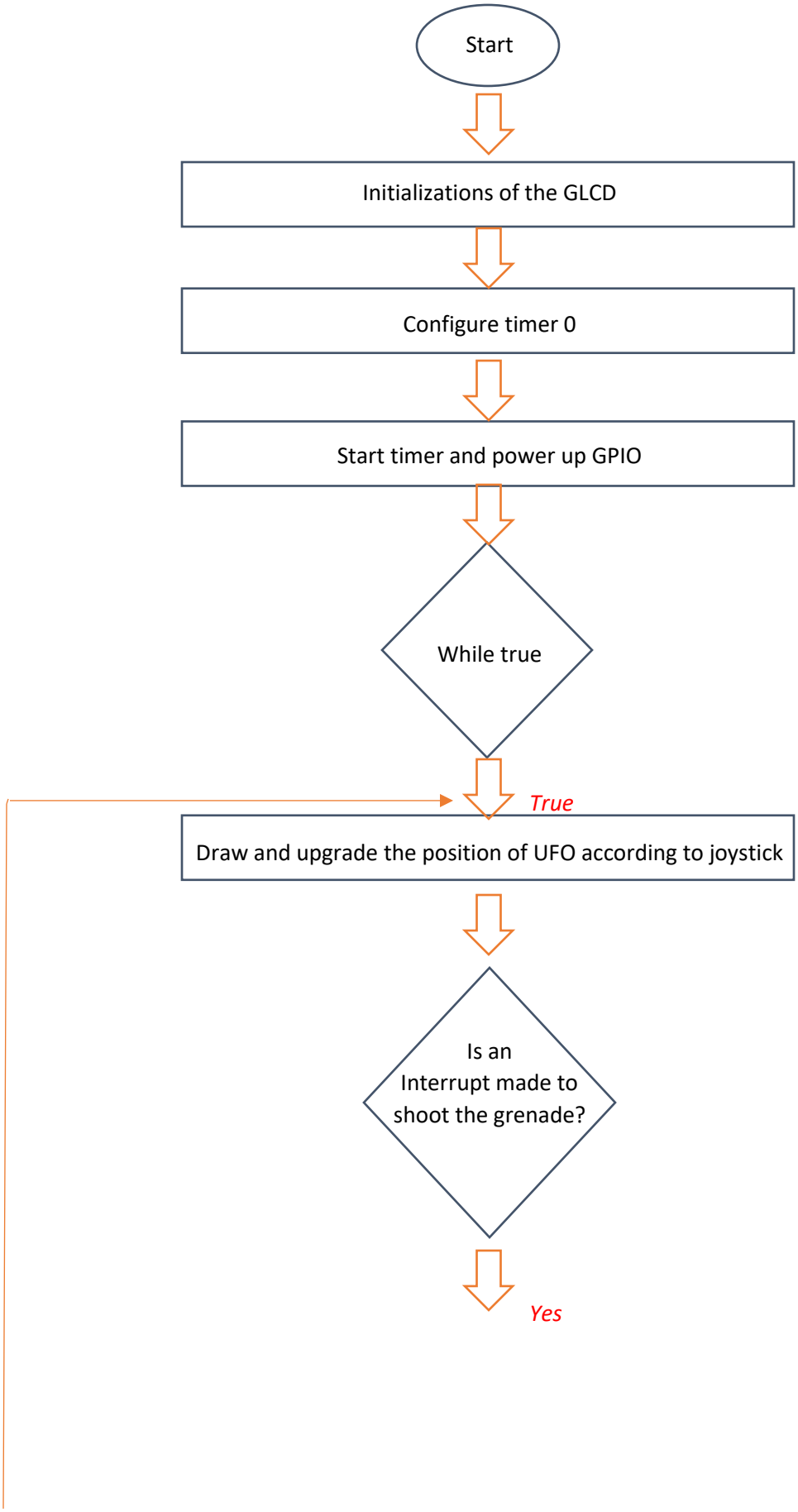
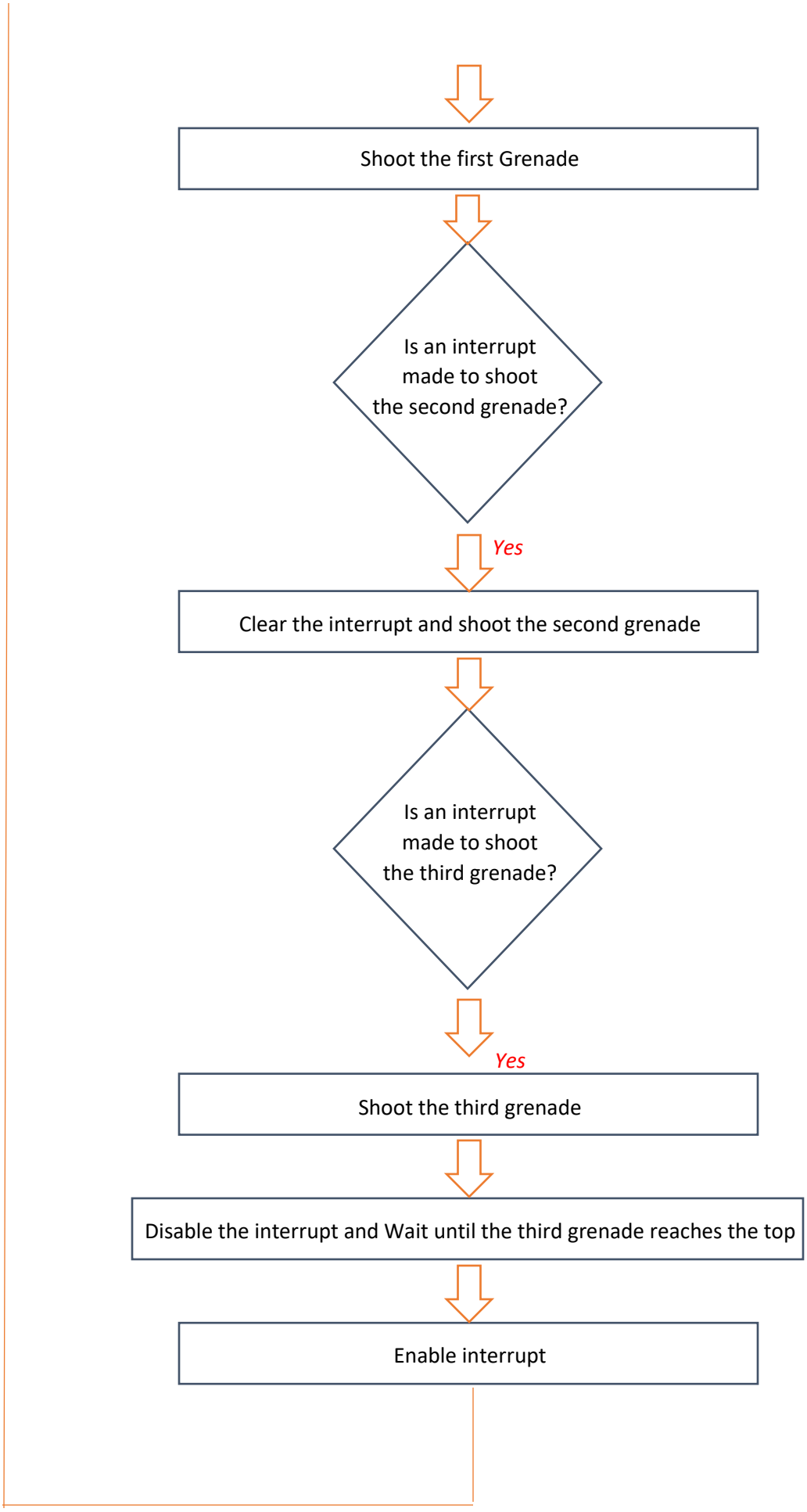


Figure: Image of the UFO with grenade

Flowchart of the Program:-




Shoot the first Grenade
Is an interrupt made to shoot the second grenade?
Yes
Clear the interrupt and shoot the second grenade
Is an interrupt made to shoot the third grenade?
Yes
Shoot the third grenade
Disable the interrupt and Wait until the third grenade reaches the top
Enable interrupt

Solution:

Design of the UFO and grenade:

First, design the UFO we have used draw_rectangle () function from the cris_utils header file. As mention we have used on box in the above design criteria to represent 4X4 matrix square. The colors were also filled according to the given parameters. The central part of the UFO had design strip had alternating colors between red and yellow which were also drawn using a for and efficient way to decrease the number of lines in the code. The grenade was drawn with purple color using draw_rectangle () function.

Interrupt for limiting the grenade to 3 onscreen:

This was the main task of the code, which was to restrict number of onscreen grenade to three and ignore the buttons pressed. Now to shoot the grenade as soon as the button was pressed we implemented interrupt method. We used INT0 for the same. In the initial part of the code we wrote code to power up the timer and to enable various other parameters which helped use to use the INT0 button and to call the interrupt request handler whenever the button was pressed. After that the program flow enters an infinite while loop constantly updating the position of the UFO and refreshing the screen. Whenever we hit the INT0 button IRQ is called which updates the variable shoot to 1 the program flow enters a conditional loop which resets the variable shoot back to zero and calls fire_1_grenade () function which displays the grenade.

If the button is pressed three or more than three times, then the program flow enters fire_3_grenade() function which displays and restricts the number of onscreen grenades to maximum of three. This restriction is done by disabling the interrupt 0 which disables the call of IRQ whenever the INT0 button is pressed and a delay of certain amount is given after which the interrupt is enabled again which brings back the program flow back to beginning.

Movement of UFO and grenade:

For the movement of UFO from left to right we have used the onboard joystick. In our program we have declared a function called ufo_position() for getting the data from the joystick and update the position of UFO. This function has two parts first is to get data from the joystick and second is to refresh the screen and display the UFO at updated location. As we know that joystick which is present on P1.25 to P1.29. We get the data and then decide if it was left or right and accordingly the conditional blocks given. When the joystick is pressed left or right first it checks if the variable cursor (which is the center position of the UFO) is weather at the extreme end of the screen or not. If it is not present in extremes of screen then it is shifted left or right by 9 pixels accordingly.

To refresh the screen the color of text is changed to background color, which is black in our case. The same UFO is drawn in black, which makes it disappear then UFO is redrawn at updated positon with appropriate color. For the grenade the refresh technique is same as UFO we draw the same grenade with black color and then update the y co-ordinate b 12 pixels then redraw the grenade by calling plot_grenade() function.

Generate sounds when grenade is released or when it disappear:

We can use onboard speaker to generate sounds, which will support the graphics for the game. Tones or sounds can played whenever the grenades are launched or when the hit the top the screen. As in the lab 5 we learnt how to use the data coming from the USB port to play audio from the speaker, we can do the same with stored audio data in some variable. We should note that we should select a short duration of a tone or audio effect as longer the audio the increased chance of lag may occur in the game. DAC output interface, speaker output driver (LM386) is present in LPC1768 board. We should note that in order to use the loud speaker we should enable the JP2 jumper. The speaker is connected on port 0 pin 26. Whenever the INT0 is called we can insert a small line of code which can play the tone or the audio effect.

Conclusion:

This was a fun assignment we revised the use of interrupt and learned to use other on-board components like joystick. Initially we had problems in refreshing the screen in order prevent multiple occurrence of the UFO or grenade but careful review of the program made us realise that to use. We improved the refresh rate by just refreshing the part where the UFO was instead of the whole screen which made the UFO looked like it moved smoothly and swiftly. With further improvement we can also add sound to the game. This assignment also encouraged us to develop more games.

Code Listing:

Main Code:

```
#include <stdio.h>

#include <stdlib.h>

#include "LPC17xx.H"                            /* LPC17xx definitions
*/

#include "GLCD.h"

#include "Serial.h"

#include "CRIS_UTILS.h"


#define __FI        1                           /* Font index 16x24
*/

#define crscr_msk 0x1F


volatile int exec = 1;

volatile uint32_t postn;

volatile int ufo_x_position=120, ufo_y_position=220, x_cord1, y_cord1, shoot
= 0;


void delay(int n)

{

      int i;

      for(i=0;i<n;i++)        //loop to count till the delay ends

    {

    }

}


int main (void)

{

    GLCD_Init();  // (0) Initializations of GLCD

      GLCD_Clear(Black); // Clear the LCD display


    // configuring timer 0
```

```
    LPC_SC->PCONP |= 1 << 1; // Powering up Timer 0
    LPC_SC->PCLKSEL0 |= 1 << 2; // Clock for timer = CCLK, i.e., CPU Clock (
    LPC_TIM0->MR0 = 1 << 18; // giving a value suitable for the LED blinking
                             // frequency based on the clock frequency
    LPC_TIM0->MCR |= 1 << 0; // Comparing Interrupt on Match 0
    LPC_TIM0->MCR |= 1 << 1; // Resetting timer on Match 0
    LPC_TIM0->TCR |= 1 << 1; // Manually Resetting Timer 0 (forced);
    LPC_TIM0->TCR &= ~(1 << 1); // Stop resetting the timer

    NVIC_EnableIRQ(TIMER0_IRQn);   // Enable timer interrupt;
            LPC_GPIOINT->IO2IntEnR |= 1 << 10;
            NVIC_EnableIRQ(EINT3_IRQn); //enabling external interrupt 3

    LPC_TIM0->TCR |= 1 << 0; // Start timer
    LPC_SC->PCONP |= ( 1 << 15 ); // Power up GPIO
    LPC_GPIO2->FIODIR |= 1 << 1; // Put P1.29 into output mode. LED is
connected to P1.29

    while(1)      //infinite loop for moving the spaceship and firing
     {
       ufo_x_position = ufo_position();    //drawing and upgrading the
position of UFO

       if(shoot == 1)         //is an interrupt made to shoot the grenade?
      {
             shoot = 0;                                  //clear interrupt
             fire_1_grenade(ufo_x_position,ufo_y_position);       //funtion
for drawing first grenade
      }
      }
}

int ufo_position()  //for checking the movement and returning the position of
the UFO
      {
```

```
      if ( exec == 1 )
            {
            ufo_plot(ufo_x_position,ufo_y_position);
            postn = (LPC_GPIO1->FIOPIN >> 25) & crscr_msk;
            LPC_GPIO2->FIOPIN = (~postn) & crscr_msk;
            postn=~postn;
            if ((postn & 0x10) && (ufo_x_position < 265))
            {
            GLCD_SetTextColor(Black); //Resetting the position of the UFO

      draw_rectangle((ufo_x_position),(ufo_y_position),(ufo_x_position+8),(uf
o_y_position),4);


      draw_rectangle((ufo_x_position+10),(ufo_y_position+4),(ufo_x_position+1
8),(ufo_y_position+4),4);
      draw_rectangle((ufo_x_position+10),(ufo_y_position-
4),(ufo_x_position+18),(ufo_y_position-4),4);


      draw_rectangle((ufo_x_position+10),(ufo_y_position+4),(ufo_x_position+1
8),(ufo_y_position+4),4);
      draw_rectangle((ufo_x_position+10),(ufo_y_position-
4),(ufo_x_position+18),(ufo_y_position-4),4);
      draw_rectangle((ufo_x_position+10),(ufo_y_position-
8),(ufo_x_position+28),(ufo_y_position-8),4);


      draw_rectangle((ufo_x_position+10),(ufo_y_position+8),(ufo_x_position+2
8),(ufo_y_position+8),4);



      GLCD_SetTextColor(Yellow);


      draw_rectangle((ufo_x_position+60),(ufo_y_position),(ufo_x_position+64)
,(ufo_y_position),4);
      GLCD_SetTextColor(Red);


      draw_rectangle((ufo_x_position+64),(ufo_y_position),(ufo_x_position+68)
,(ufo_y_position),4);



      GLCD_SetTextColor(Blue);
```

```
      draw_rectangle((ufo_x_position+50),(ufo_y_position+4),(ufo_x_position+5
8),(ufo_y_position+4),4);

      draw_rectangle((ufo_x_position+50),(ufo_y_position-
4),(ufo_x_position+58),(ufo_y_position-4),4);

      draw_rectangle((ufo_x_position+50),(ufo_y_position-
4),(ufo_x_position+58),(ufo_y_position-4),4);


      draw_rectangle((ufo_x_position+50),(ufo_y_position+4),(ufo_x_position+5
8),(ufo_y_position+4),4);


      draw_rectangle((ufo_x_position+45),(ufo_y_position+8),(ufo_x_position+4
8),(ufo_y_position+8),4);

      draw_rectangle((ufo_x_position+45),(ufo_y_position-
8),(ufo_x_position+48),(ufo_y_position-8),4);

      ufo_x_position = ufo_x_position + 9;

                  }


else

if((postn & (0x02)) && (ufo_x_position > 9))

{

      GLCD_SetTextColor(Black);


      draw_rectangle((ufo_x_position+52),(ufo_y_position),(ufo_x_position+60)
,(ufo_y_position),4);


      draw_rectangle((ufo_x_position+42),(ufo_y_position+4),(ufo_x_position+5
0),(ufo_y_position+4),4);

      draw_rectangle((ufo_x_position+42),(ufo_y_position-
4),(ufo_x_position+50),(ufo_y_position-4),4);


      draw_rectangle((ufo_x_position+42),(ufo_y_position+4),(ufo_x_position+5
2),(ufo_y_position+4),4);

      draw_rectangle((ufo_x_position+42),(ufo_y_position-
4),(ufo_x_position+52),(ufo_y_position-4),4);


      draw_rectangle((ufo_x_position+32),(ufo_y_position+8),(ufo_x_position+4
5),(ufo_y_position+8),4);

      draw_rectangle((ufo_x_position+32),(ufo_y_position-
8),(ufo_x_position+45),(ufo_y_position-8),4);



      GLCD_SetTextColor(Yellow);
```

```
      draw_rectangle((ufo_x_position-
4),(ufo_y_position),(ufo_x_position),(ufo_y_position),4);

      GLCD_SetTextColor(Red);

      draw_rectangle((ufo_x_position-8),(ufo_y_position),(ufo_x_position-
4),(ufo_y_position),4);



      GLCD_SetTextColor(Blue);


      draw_rectangle((ufo_x_position+2),(ufo_y_position+4),(ufo_x_position+10
),(ufo_y_position+4),4);


      draw_rectangle((ufo_x_position+2),(ufo_y_position+4),(ufo_x_position+10
),(ufo_y_position+4),4);

      draw_rectangle((ufo_x_position+2),(ufo_y_position-
4),(ufo_x_position+10),(ufo_y_position-4),4);

      draw_rectangle((ufo_x_position+2),(ufo_y_position-
4),(ufo_x_position+10),(ufo_y_position-4),4);


      draw_rectangle((ufo_x_position+12),(ufo_y_position+8),(ufo_x_position+1
5),(ufo_y_position+8),4);

      draw_rectangle((ufo_x_position+12),(ufo_y_position-
8),(ufo_x_position+15),(ufo_y_position-8),4);

      ufo_x_position = ufo_x_position - 9;

}


else

      ufo_x_position = ufo_x_position;

exec = 0;

}

      return(ufo_x_position);

      }


      //Function for drawing the first grenade

int fire_1_grenade(int x_cord1, int y_cord1)

      {

        int y1=y_cord1,x1=x_cord1;

            while(y1>12)

            {
```

```
            ufo_x_position = ufo_position();                //updating the
UFO's position according to the position of the Joystick
            GLCD_SetTextColor(Purple);
            if(y1>12)
                  plot_grenade(x1,y1);
            if(shoot == 1)          //Is an interrupt made to shoot the
grenade?
                  {
                  GLCD_Clear(Black);
                  shoot = 0;  //clear interrupt
                  fire_2_grenade(ufo_x_position,ufo_y_position,x1,y1);
      //making arrangements to shoot the second grenade if the interrupt is
pressed the second time
                  break;
                  }

            delay(3000000);
            GLCD_SetTextColor(Black);
            plot_grenade(x1,y1);
            y1=y1-12;
            }
      return(0);
      }

      //Function for drawing the second grenade
      int fire_2_grenade(int x_cord1, int y_cord1, int x1, int y1)
      {
       int y2=y_cord1,x2=x_cord1;
       shoot = 0;

      while(y2>12 || y1>12)         //Until the second grenade reaches the
Top
            {
            ufo_x_position = ufo_position();
            GLCD_SetTextColor(Purple);
            if(y2>12)
```

```
                  plot_grenade(x2,y2);
            plot_grenade(x1,y1);


            delay(3000000);
            GLCD_SetTextColor(Black);
            if(y2>12)
                  plot_grenade(x2,y2);
            plot_grenade(x1,y1);
            y2=y2-12;
            y1=y1-12;
            if(y1<8)
                  {
                  GLCD_Clear(Black);
                  NVIC_ClearPendingIRQ(EINT3_IRQn);
                  fire_1_grenade(x1,y1+8);
                  break;
                  }


            if(shoot == 1)          //Is An interrupt made to shoot the
grenade?
                  {
                  shoot = 0;        //clear interrupt
                  GLCD_Clear(Black);
                  NVIC_ClearPendingIRQ(EINT3_IRQn);
                  fire_3_grenade(ufo_x_position,ufo_y_position,x1,y1,x2,y2);
      //making arrangement to shoot the third grenade if the interrupt is
pressed the third time
                  break;
                  }
            }
            return(0);
      }


      //Function for drawing the third grenade
```

```
      int fire_3_grenade(int x_cord1, int y_cord1, int x1, int y1, int x2,
int y2)
      {
            int y3=y_cord1,x3=x_cord1;
            NVIC_ClearPendingIRQ(EINT3_IRQn);
      while(y3>12 || y2>12 || y1>12)            //Until the third grenade
reaches the Top
            {
            ufo_x_position = ufo_position();          //Updating the UFO's
position
            GLCD_SetTextColor(Purple);
            plot_grenade(x3,y3);
            plot_grenade(x2,y2);
            plot_grenade(x1,y1);
            NVIC_DisableIRQ(EINT0_IRQn);              //to disable the
interrupt so that no more grenades are fired after 3 in the screen

            shoot=0;
            delay(3000000);
            GLCD_SetTextColor(Black);

            plot_grenade(x3,y3);
            plot_grenade(x2,y2);
            plot_grenade(x1,y1);
            y3=y3-12;
            y2=y2-12;
            y1=y1-12;
            if(y1<8)                        //if 1st grenade reaches top, go to
grenade 2
                  {
                  GLCD_Clear(Black);
                  NVIC_ClearPendingIRQ(EINT3_IRQn);
                  NVIC_EnableIRQ(EINT3_IRQn);         //enable interrupt
                  fire_2_grenade(x2,y2-8,x3,y3+2);
                  break;
                  }
```

```
            else if (y2<8)
            {
              GLCD_Clear(Black);
                  NVIC_EnableIRQ(EINT3_IRQn);         //enable interrupt
                  fire_1_grenade(x3,y3-5);
                  break;
            }
            }
      return(0);
      }


void TIMER0_IRQHandler(void)   //interrupt handling on timer 0
      {
  if ( (LPC_TIM0->IR & 0x01) == 0x01 ) // if MR0 interrupt
    {
                  LPC_TIM0->IR |= 1 << 0; // Clear MR0 interrupt flag
                  LPC_GPIO1->FIOPIN ^= 1 << 1; // toggle the P0.29 LED;
                  exec = 1; //execution is allowed
    }
      }

void EINT3_IRQHandler(void) //handling the shoot interrupt
      {
  volatile int i = 0;
      LPC_GPIOINT->IO2IntClr |= 1 << 10;
      for(i=0;i<100;i++)
            {}
      fire = 1;
      }
```

Function for drawing the graphics (UFO and the grenades) on the LCD

```
int draw_rectangle(int x_ufo_1, int y_ufo_1, int x2, int y2, int n)
      {
      int i;
      for(i = 0; i<n; i++)          //draws a rectangle
            {
            draw_line(x_ufo_1,(y_ufo_1+i),x2,(y_ufo_1+i));
            }
      return(0);
      }
int ufo_plot(int x_ufo_0, int y_ufo_0)          //plotting ufo
      {
      volatile int i;
      for(i=0;i<15;i++)       //used to draw the centre rectangle of ufo with
alternating red and yellow color
                  {
                  last_color = Red;
                  last_color = lcd_colors[ i % 2 ];
                  GLCD_SetTextColor(last_color);

      draw_rectangle((x_ufo_0+(4*i)),y_ufo_0,(x_ufo_0+(4*i)+4),y_ufo_0,4);
                  }


            GLCD_SetTextColor(Blue);

      draw_rectangle((x_ufo_0+10),(y_ufo_0+4),(x_ufo_0+50),(y_ufo_0+4),4);

      draw_rectangle((x_ufo_0+10),(y_ufo_0+4),(x_ufo_0+50),(y_ufo_0+4),4);
            draw_rectangle((x_ufo_0+10),(y_ufo_0-4),(x_ufo_0+50),(y_ufo_0-
4),4);
            draw_rectangle((x_ufo_0+10),(y_ufo_0-4),(x_ufo_0+50),(y_ufo_0-
4),4);

      draw_rectangle((x_ufo_0+15),(y_ufo_0+8),(x_ufo_0+45),(y_ufo_0+8),4);
```

```
            draw_rectangle((x_ufo_0+15),(y_ufo_0-8),(x_ufo_0+45),(y_ufo_0-
8),4);
            return(0);
      }

int plot_grenade(int x_ufo_1, int y_ufo_1)            //plotting grenade
      {
      draw_rectangle((x_ufo_1+29),(y_ufo_1-16),(x_ufo_1+33),(y_ufo_1-16),4);
      draw_rectangle((x_ufo_1+29),(y_ufo_1-19),(x_ufo_1+33),(y_ufo_1-19),4);
      draw_rectangle((x_ufo_1+30),(y_ufo_1-23),(x_ufo_1+32),(y_ufo_1-23),4);

      return(0);
      }
```