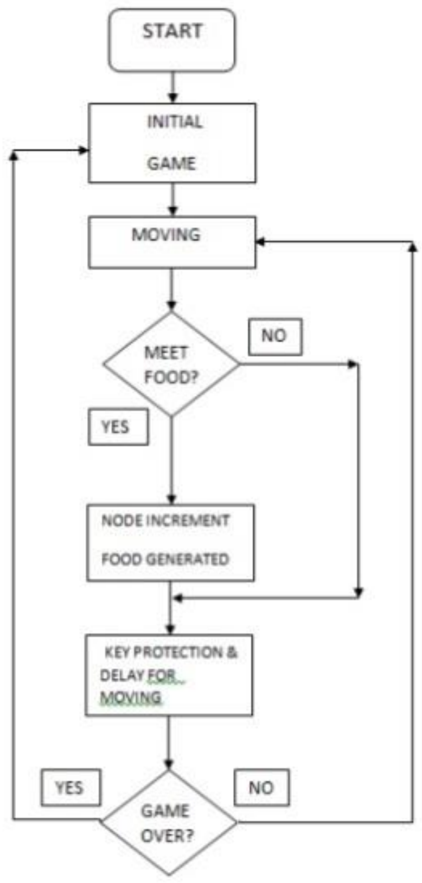
Hubei University of Technology

Experiment report

Grade	

Course name	Embedded SYSTEMS AND DESIGNS			
Experimental name	EXPERIMENT 4 – **Snake Game Player Design**			
Departments	COMPUTER SCIENCE	**Lecturer**	Dr. Liu Chun	
Name	**Rimon Mahmud**	Student id	**1811561124**	
Experimental purpose	The aim of this experiment is to design a shake game module			
Experimental preparation	**1.Experimental environment:** PROTEUS 8 PROFESSIONAL, WINDOWS 10 **2. Knowledge preparation：** LPC2124 processor, LPC2124 Architecture, C language.			
Experimental content	In this project we'll able to design a fully functional gaming module with CPD LPC2124			
Experimental analysis	By this project we've been familiar with LPC2124 CPU and its function. We also able to program a LCD display. we run an active snake game			

Experimental flowchart	 START INITIAL GAME MOVING MEET FOOD? NO YES NODE INCREMENT FOOD GENERATED KEY PROTECTION & DELAY FOR MOVING YES NO GAME OVER?

Code (main . c)

```
#include  "config.h"
#include  "lcddrv.h"
#include  <string.h>

/*********ASCII ****/
TCOLOR disp_color;
TCOLOR back_color;

int dir[9][2]={0,-1,0,1,-1,0,1,0,-1,1,-1,-1,1,1,1,-1,0,0};
int  snake_length, snake_dir = 3,food_x,food_y,gameover;
int keyvalue[16]={
                -1, 0,-1,-1,
                 2,-1, 3,-1,
                -1, 1,-1,-1,
                -1,-1,-1,-1};
///////////////
struct Snake
{
   int x,y;
}snake[1000];

void delayMs(int n)
{
   int i;
   for(i = 0;i < 1000;i++)
      for(;n>0;n--);
}

/*Akram MD Asim*/////Akram MD Asim/////ASCII color1//color2////
void GUI_SetColor(TCOLOR color1, TCOLOR color2)
 {
    GUI_CopyColor(&disp_color,color1);
    GUI_CopyColor(&back_color,color2);
 }

 //////////0<x<127,0<y<63
 void  Big_Point(uint8 x,uint8 y,TCOLOR  color)
 {
    int i;
    for (i=0;i < 9;i++)
    {
      GUI_Point(x+dir[i][0],y+dir[i][1],color);
    }
 }

 ////////////////////Akram MD Asim/////////////
 void snake_init(int x,int y)
 {
 int i ;
 snake_length = 0;
 for (i = 0;i < 10;i++)
     {
           Big_Point(x,y,LCD_DISP_COLOR);
           snake[snake_length].x = x;
           snake[snake_length].y = y;
           snake_length++;
           x-=3;
      }
}

//////////////////
void creat_food()
{
     int i,flag = 1;
      Big_Point(food_x,food_y,LCD_BACK_COLOR);
     do
     {
           //stand((unsigned)time(NULL));
       food_x = rand() % 127;
       food_y = rand() % 63;
       for( i= 0;i  < snake_length-1;i++)
       {
            if(snake[i].x+2 <= food_x || food_x <= snake[i].x-2 || snake[i].y+2
<= food_y || food_y <= snake[i].y-2)
                    flag = 0;
            }
      }while(flag);
      Big_Point(food_x,food_y,LCD_DISP_COLOR);
}

////////////////////////
void snake_move(int direction)
{
   int i;
   struct Snake last;
last = snake[snake_length-1];
   ///////////////////////////////////////////
for(i = snake_length - 1;i>0;i--)
{
   snake[i]=snake[i-1];
}
////////// diraction/////////////////
  snake[0].x += dir[direction][0] * 3;
  snake[0].y += dir[direction][1] * 3;
///////////////
  if(snake[0].x>= 127)snake[0].x = 1;
  else if(snake[0].x <= 0)snake[0].x = 126;
  else if(snake[0].y <= 0)snake[0].y = 62;
  else if(snake[0].y >= 63)snake[0].y = 1;
//////////////////
  for(i = 1; i< snake_length -1; i++)
  {
   if(snake[i].x + 2>= snake[0].x && snake[0].x>= snake[i].x-2 &&
snake[i].y+2 >= snake[0].y && snake [0].y >= snake[i].y-2)
   {
      gameover =1;
      return;
     }
   }
   ////////////
  if(snake[0].x +2 >= food_x && food_x >= snake[0].x-2
&&snake[0].y+2 >= food_y && food_y >= snake[0].y-2)
  {
   snake[snake_length]=last;
   creat_food();
   snake_length++;
   }
    else
   {
        Big_Point(last.x,last.y,LCD_BACK_COLOR);
}
//////////
Big_Point(snake[0].x,snake[0].y,LCD_DISP_COLOR);
```

```
}


/////////////
int GetDir()
{
  int temp,temp1,temp2;
  IO0DIR |=0x0000F;
  IO0SET |=0x0000F;
  temp1= IO0PIN& 0X000F0000;
  ////////////
switch(temp1)
{
  case 0X0010000:temp1 = 0;break;
  case 0X0020000:temp1 = 1;break;
  case 0X0040000:temp1 = 2;break;
  case 0X0080000:temp1 = 3;break;
  default:temp1 = 4;
}
  if(temp1 != 4)
   {
IO0DIR &= 0xFFFFFFF0;
IO0DIR |= 0x000F0000;
IO0CLR |= 0x0000000F;
IO0SET |= 0x000F0000;
temp2 = IO0PIN & 0x0000000F;
    /////////////////////////
switch(temp2)
{
case 0x001:temp = temp1*4 + 0;break;
case 0x002:temp = temp1*4 + 1;break;
case 0x004:temp = temp1*4 + 2;break;
case 0x008:temp = temp1*4 + 3;break;
default:temp = 0;
}
IO0CLR |= 0xF0000;
IO0DIR &= 0x0FFFF;
}
return temp;
}

int main (void)
{

  int dir = 3;
  GUI_Initialize();
  GUI_SetColor(LCD_DISP_COLOR,LCD_BACK_COLOR);
  snake_init(64,32);
  creat_food();
 while (!gameover)
 {
 dir = keyvalue[GetDir()];
 if(dir == -1)
dir = snake_dir;
 if(snake_dir + dir != 1 && snake_dir + dir !=5)
 {
 snake_dir = dir;
 }
 snake_move(snake_dir);
 delayMs(100000);
   }
}
```

Code (target . c)

```
#define IN_TARGET
#include "config.h"


void __irq IRQ_Exception(void)
   {
  while(1);
   }
  /*******************************************/
  void FIQ_Exception(void)
   {
    while(1);
    }
  /*********** Target  limit  ***********/
    void TargetInit(void)
    {

    }

    void TargetResetInit(void)
    {

          MAMCR=2;

          #if Fcclk < 20000000
          MAMTIM=1;
          #else
          #if Fcclk < 40000000
          MAMTIM=2;
          #else
          MAMTIM= 3;
          #endif
          #endif

          VICIntEnClr=0xffffffff;
          VICVectAddr=0;
          VICIntSelect=0;
```

```
}

#include "rt_sys.h"
#include "stdio.h"

#pragma import(__use_no_semihosting_swi)
#pragma import(__use_two_region_memory)

int__rt_div0(int a)
{
      a = a;
      return 0;
}

int fputc(int ch,FILE*f)
{
      ch = ch;
      f = f;
      return 0;
}

int fgetc(FILE*f)
{
      f = f;
      return 0;
}

int _sys_close(FILEHANDLE fh)
{
      fh = fh;
      return 0;
}

int _sys_write(FILEHANDLE fh,const unsigned char *buf,
unsigned len, int mode)
{

      fh = fh;
      buf = buf;
      len =len;
      mode = mode;
      return 0;
}

 int _sys_read(FILEHANDLE fh, unsigned char *buf,
unsigned len, int mode)
{
       fh = fh;
      buf = buf;
```

```
        len =len;
        mode = mode;
        return 0;
}
        void_ttywrch(int ch)
{
        ch=ch;
}
int _sys_istty(FILEHANDLE fh)
{
        fh = fh;
        return 0;
}

int _sys_seek(FILEHANDLE fh,long pos)
{
        fh = fh;
        return 0;
}
int _sys_ensure(FILEHANDLE fh)
{
        fh = fh;
        return 0;
}
long _sys_flen(FILEHANDLE fh)
{
        fh =fh;
        return 0;
}
int _sys_tmpnam(char * name, int sig, unsigned maxlen)
{
        name = name;
        sig = sig;
        maxlen=maxlen;
        return 0;
}
void _sys_exit(int returncode)
{
        returncode = returncode;
}
char* _sys_command_string(char * cmd, int len)
{
        cmd = cmd;
        len = len;
        return 0;
}
```

Code (lcddrv . c)

```
/****************************************Copyright
(c)**************************************************
**--------------File Info---------------------------------------------------------------
---------------
** File name: LCMDRV.C
** Last modified Date:
** Last Version: 1.0
** Descriptions: MG12864Í¼ĐÎÒº³¾§Ä£¿éÇý¶¯³ÌĐò¡£T6963C¿ØÖÆÆ÷
** Modified by:
** Modified date:
** Version:
** Descriptions:
*************************************************************
*******************************************/
#include "config.h"
#include "lcddrv.h"
/* ¶¨Òå×ÜÏßÆðÊ¼µÄGPIO£¬¼´D0¶ÔÓ¦µÄGPIOÖµ(P0.4) */
/* ¶¨ÒåÏÔÊ¾»º³åÇø */
#define  BUS_NO              4
/* Êä³ö×ÜÏßÊý¾Ýºê¶¨Òå */
#define OutData(dat)  IO0DIR  =  IO0DIR  |(0xff<<BUS_NO);  IO0CLR  =
0xff<<BUS_NO; IO0SET = (dat&0xff)<<BUS_NO
#define InData()        IO0DIR  =  IO0DIR  &~(0x000000ff<<BUS_NO);dat
= (uint8)((IO0PIN&(0xFFFFFFFF))>>BUS_NO)
/* ¶¨ÒåREAD¿ØÖÆ */
#define  LCM_RD              12
#define  LCM_UNREAD()              IO0SET = 1<<LCM_RD
#define  LCM_READ()                       IO0CLR = 1<<LCM_RD
/* ¶¨ÒåWRITE¿ØÖÆ */
#define  LCM_WR     13
#define  LCM_UNWRITE()             IO0SET = 1<<LCM_WR
#define  LCM_WRITE()               IO0CLR = 1<<LCM_WR
/* ¶¨ÒåC/D#¿ØÖÆ */
#define  LCM_CD              14
#define  LCM_COM()                        IO0SET = 1<<LCM_CD
#define  LCM_DATA()                       IO0CLR = 1<<LCM_CD
/* ¶¨ÒåC/D#¿ØÖÆ */
#define  LCM_CE              15
#define  LCM_DISABLE()                    IO0SET = 1<<LCM_CE
#define  LCM_ENABLE()                     IO0CLR = 1<<LCM_CE
/* ¶¨ÒåLCM²Ù×÷µÄÃüÁî×Ö */
// T6963C ÃüÁî¶¨Òå
#define LCM_CUR_POS 0x21 // ¹â±êÎ»ÖÃÉèÖÃ
#define LCM_CGR_POS 0x22 // CGRAM Æ«ÖÃµØÖ·ÉèÖÃ
#define LCM_ADD_POS 0x24 // µØÖ·Ö¸ÕëÎ»ÖÃ
#define LCM_TXT_STP 0x40 // ÎÄ±¾ÇøÊ×Ö·
#define LCM_TXT_WID 0x41 // ÎÄ±¾Çø¿í¶È
#define LCM_GRH_STP 0x42 // Í¼ĐÎÇøÊ×Ö·
#define LCM_GRH_WID 0x43 // Í¼ĐÎÇø¿í¶È
#define LCM_MOD_OR       0x80 // ÏÔÊ¾·½Ê½Âß¼»ò
```

```
#define LCM_MOD_XOR 0x81 // ÏÔÊ¾·½Ê½Âß¼-Òì»ò
#define LCM_MOD_AND 0x82 // ÏÔÊ¾·½Ê½Âß¼-Óë
#define LCM_MOD_TCH 0x83 // ÏÔÊ¾·½Ê½ÎÄ±¾ÌØÕ÷
#define LCM_DIS_SW         0x90                                   //
ÏÔÊ¾¿ª¹ØD0=1/0:¹â±êÉÁË¸ÆôÓÃ/½ûÓÃ
// D1=1/0:¹â±êÏÔÊ¾ÆôÓÃ/½ûÓÃ
// D2=1/0:ÎÄ±¾ÏÔÊ¾ÆôÓÃ/½ûÓÃ
// D3=1/0:Í¼ÐÎÏÔÊ¾ÆôÓÃ/½ûÓÃ
#define      LCM_CUR_SHP      0xA0      //      ¹â±êÐÎ×´Ñ¡Ôñ0xA0-
0xA7±íÊ¾¹â±êÕ¼µÄÐÐÊý
#define LCM_AUT_WR 0xB0 // ×Ô¶¯Ð´ÉèÖÃ
#define LCM_AUT_RD 0xB1 // ×Ô¶¯¶ÁÉèÖÃ
#define LCM_AUT_OVR 0xB2 // ×Ô¶¯¶Á/Ð´½áÊø
#define LCM_INC_WR 0xC0 // Êý¾ÝÒ»´ÎÐ´µØÖ·¼Ó1
#define LCM_INC_RD 0xC1 // Êý¾ÝÒ»´Î¶ÁµØÖ·¼Ó1
#define LCM_DEC_WR 0xC2 // Êý¾ÝÒ»´ÎÐ´µØÖ·¼õ1
#define LCM_DEC_RD 0xC3 // Êý¾ÝÒ»´Î¶ÁµØÖ·¼õ1
#define LCM_NOC_WR 0xC4 // Êý¾ÝÒ»´ÎÐ´µØÖ·²»±ä
#define LCM_NOC_RD 0xC5 // Êý¾ÝÒ»´Î¶ÁµØÖ·²»±ä
#define LCM_SCN_RD 0xE0 // ÆÁ¶Á
#define LCM_SCN_CP 0xE8 // ÆÁ¿½±´
#define LCM_BIT_OP 0xF0 // Î»²Ù×÷
uint8 const  turnf[8] = {7,6,5,4,3,2,1,0};
uint8 const  DEC_HEX_TAB1[8] = {0x80, 0x40, 0x20, 0x10, 0x08, 0x04,
0x02, 0x01};
uint8 const  DEC_HEX_TAB[8] = {0x01, 0x02, 0x04, 0x08, 0x10, 0x20,
0x40, 0x80};

/**********************************************************
*******************************************/

/**********************************************************
*******************************************
** º¯ÊýÃû³Æ: LCM_READSTATE
** ¹¦ÄÜÃèÊö: ¶ÁÈ¡LCMÄÚ²¿µÄ×´Ì¬
** Êä¡¡Èë: ÎÞ
** Êä¡¡³ö: LCMÄÚ²¿×´Ì¬Öµ
** È«¾Ö±äÁ¿:
** µ÷ÓÃÄ£¿é:
** Modified by:
** Modified date:
**-------------------------------------------------------------------------------
--------------
*********************************************************
******************************************/
uint8 LCM_READSTATE()
{
        uint8 dat;
        IO0DIR &= ~(0x000000ff<<BUS_NO);
        LCM_UNWRITE();
```

```
		LCM_COM();
		LCM_READ();
		LCM_ENABLE();
		//DELAY5();
		//DELAY5();
		//DELAY5();
		//InData();
		dat = (uint8)((IO0PIN)>>BUS_NO);
		//LCM_UNREAD();
		//LCM_UNWRITE();
		LCM_DISABLE();
		return dat;
}
/************************************************************
*********************************************
** º¯ÊýÃû³Æ: LCM_STA01
**                                                          ¹¦ÄÜÃèÊö:
×´Ì¬Î»STA1,STA0ÅÐ¶Ï¶ÁÐ´Ö¸Áî ºÍ¶ÁÐ´Êý¾Ý£¬ÔÚ¶ÁÐ´Êý¾Ý»òÕßÐ´
ÈëÃüÁîÇ°±ØÐë±£Ö¤¾ùÎª1
** Êä¡¡Èë: ÎÞ
** Êä¡¡³ö: ÎÞ
** È«¾Ö±äÁ¿: ÎÞ
** µ÷ÓÃÄ£¿é: LCM_READSTATE
** Modified by:
** Modified date:
**----------------------------------------------------------------------------------------
--------------
************************************************************
*******************************************/
uint8 LCM_STA01(void)
{
   uint8 i;
   for(i=10;i>0;i--)
   {
      if(( LCM_READSTATE() & 0x03) == 0x03) // ¶ÁÈ¡×´Ì¬
      {
         break;
      }
   }
   return(i); // Èô·µ»ØÁãËµÃ÷´íÎó
}
/************************************************************
*********************************************
** º¯ÊýÃû³Æ: LCM_STA3
** ¹¦ÄÜÃèÊö: ×´Ì¬Î»STA3
** Êä¡¡Èë: ÎÞ
** Êä¡¡³ö: ÎÞ
** È«¾Ö±äÁ¿: ÎÞ
** µ÷ÓÃÄ£¿é: LCM_READSTATE
** Modified by:
```

```
** Modified date:
**------------------------------------------------------------------------------------
--------------
************************************************************
*******************************************/
uint8 LCM_STA3(void)
{
   uint8 i;
   for(i=10;i>0;i--)
   {
      if(( LCM_READSTATE() & 0x08) == 0x08) // ¶ÁÈ¡×´Ì¬
      {
         break;
      }
   }
   return(i); // Èô·µ»ØÁãËµÃ÷´íÎó
}
/************************************************************
********************************************
** º¯ÊýÃû³Æ: LCM_WrCommand
** ¹¦ÄÜÃèÊö: Ð´ÃüÁî×Ó³ÌÐò
** Êä¡¡Èë: command  ÒªÐ´ÈëLCMµÄÃüÁî×Ö
** Êä¡¡³ö: ÎÞ
** È«¾Ö±äÁ¿: ÎÞ
** µ÷ÓÃÄ£¿é: ÎÞ
** Modified by:
** Modified date:
**------------------------------------------------------------------------------------
--------------
************************************************************
*******************************************/
void LCM_WrCommand(uint8 command)
{
         LCM_UNREAD();
         LCM_COM();
         LCM_WRITE();
         LCM_ENABLE();
         OutData(command);
         //LCM_UNWRITE();
         //LCM_READ();
         LCM_DISABLE();
}
/************************************************************
********************************************
** º¯ÊýÃû³Æ: LCM_WrData
** ¹¦ÄÜÃèÊö: Ð´Êý¾Ý×Ó³ÌÐò
** Êä¡¡Èë: wrdata      ÒªÐ´ÈëLCMµÄÊý¾Ý
** Êä¡¡³ö: ÎÞ
** È«¾Ö±äÁ¿: ÎÞ
** µ÷ÓÃÄ£¿é: ÎÞ
** Modified by:
** Modified date:
**---------------------------------------------------------------------------------------
--------------
************************************************************
*******************************************/
void LCM_WrData(uint8 wrdata)
{
        LCM_UNREAD();
        LCM_DATA();
        LCM_WRITE();
        LCM_ENABLE();
        OutData(wrdata);
        //LCM_UNWRITE();
        //LCM_READ();
        LCM_DISABLE();
}
/*************************************************************
*******************************************
** º¯ÊýÃû³Æ: LCM_WrParameter
** ¹¦ÄÜÃèÊö: ÏòLCMÐ´Èë²ÎÊý£¬´øË«²ÎÊý£¬Ò»¸ö²ÎÊý£¬»òÕß²»´ø²ÎÊý
** Êä¡¡Èë: cmd²ÎÊý£»para1²ÎÊý1£»para2²ÎÊý2£»num²ÎÊý¸öÊý
** Êä¡¡³ö: ·µ»Ø²Ù×÷½á¹û
** È«¾Ö±äÁ¿: ÎÞ
** µ÷ÓÃÄ£¿é: ÎÞ
** Modified by:
** Modified date:
**---------------------------------------------------------------------------------------
--------------
************************************************************
*******************************************/
uint8 LCM_WrParameter(uint8 cmd,uint8 para1,uint8 para2,uint8 num)
{
        switch (num)
        {
                case 0x00:
                /*
                        if(LCM_STA01() == 0)
                {
                return 1;
                }
        */
                        LCM_WrCommand(cmd);
                        break;
                case 0x01:
                        /*
                        if(LCM_STA01() == 0)
                {
                return 1;
                }
```

```
                    LCM_WrData(para1);
                    if(LCM_STA01() == 0)
                    {
                    return 2;
                    }
                    LCM_WrCommand(cmd);
                    */
                    LCM_WrData(para1);
                    LCM_WrCommand(cmd);
                            break;
                    case 0x02:
                    /*
                            if(LCM_STA01() == 0)
                    {
                    return 1;
                    }
                    LCM_WrData(para1);
                            if(LCM_STA01() == 0)
                    {
                    return 2;
                    }
                             LCM_WrData(para2);
                    if(LCM_STA01() == 0)
                    {
                    return 3;
                    }
                    LCM_WrCommand(cmd);
            */
            LCM_WrData(para1);
            LCM_WrData(para2);
            LCM_WrCommand(cmd);
                    break;
            }
            return 0;
}
/************************************************************
*******************************************
** º¯ÊýÃû³Æ: LCM_ReadByte
** ¹¦ÄÜÃèÊö: ¶ÁÈ¡Ö¸¶¨µãÉÏµÄ×Ö½ÚÊý¾Ý
** Êä¡¡Èë: x,y×ø±êÖµ
** Êä¡¡³ö: ·µ»Ø¸ÃµãÉÏµÄ×Ö½ÚÊý¾Ý
** È«¾Ö±äÁ¿: ÎÞ
** µ÷ÓÃÄ£¿é: ÎÞ
** Modified by:
** Modified date:
**---------------------------------------------------------------------------------
--------------
************************************************************
*******************************************/
uint8 LCM_ReadByte(uint8 x, uint8 y)
```

	```
	{
	        uint8 dat=0xff;
	        uint8 x1;
	        uint32 iPos;
	        x1 = x >> 3; // È¡Y·½Ïò·ÖÒ³µØÖ·
	        iPos = (uint32)y * 0x1e + x1;
	        LCM_WrParameter(LCM_ADD_POS,iPos&0xff,iPos/256,2);
	        LCM_WrParameter(LCM_NOC_RD,0,0,0);
	        /*
	        if(LCM_STA01() == 0)
	   {
	      return 1;
	   }
	   */
	   IO0DIR = IO0DIR &~(0x000000ff<<BUS_NO);
	        LCM_UNWRITE();
	        LCM_DATA();
	        LCM_READ();
	        LCM_ENABLE();
	        //InData();
	        dat = (uint8)((IO0PIN)>>BUS_NO);
	        LCM_DISABLE();
	        return dat;

	}
	/************************************************************
	*********************************************
	** º¯ÊýÃû³Æ: LCM_DispIni
	** ¹¦ÄÜÃèÊö: LCMÓ²¼þ³õÊ¼»¯
	** Êä¡¡Èë: ÎÞ
	** Êä¡¡³ö: ÎÞ
	** È«¾Ö±äÁ¿: ÎÞ
	** µ÷ÓÃÄ£¿é: ÎÞ
	** Modified by:
	** Modified date:
	**------------------------------------------------------------------------------------------
	--------------
	************************************************************
	********************************************/
	void LCM_DispIni(void)
	{
	 uint32  i;
	// ÉèÖÃÒý½ÅÁ¬½ÓÄ£¿é
	#if LCM_RD < 16
	   PINSEL0 &= ~(3 << (2 * LCM_RD));
	#else
	   PINSEL1 &= ~(3 << (2 * (LCM_RD - 16)));
	#endif

	#if LCM_WR < 16
	```

```
   PINSEL0 &= ~(3 << (2 * LCM_WR));
#else
   PINSEL1 &= ~(3 << (2 * (LCM_WR - 16)));
#endif

#if LCM_CD < 16
   PINSEL0 &= ~(3 << (2 * LCM_CD));
#else
   PINSEL1 &= ~(3 << (2 * (LCM_CD - 16)));
#endif

#if  BUS_NO<9
   for (i = BUS_NO; i < BUS_NO+8; i++)
   {
      PINSEL0 &= ~(3 << (2 * i));
   }
#else
   for (i = BUS_NO; i < 16; i++)
   {
      PINSEL0 &= ~(3 << (2 * i));
   }
   for (; i < (BUS_NO+8); i++)
   {
      PINSEL1 &= ~(3 << (2 * (i-16)));
   }
#endif
  // ÉèÖÃI/OÎªÊä³ö·½Ê½
  IO0DIR                                                                  =
IO0DIR|(1<<LCM_RD)|(1<<LCM_WR)|(1<<LCM_CD)|(1<<LCM_CE);
  IO0DIR = IO0DIR|(0xFF<<BUS_NO);

  LCM_WrParameter(LCM_TXT_STP,0x00,0x00,2);
  LCM_WrParameter(LCM_TXT_WID,0x1E,0x00,2);
  LCM_WrParameter(LCM_GRH_STP,0x00,0x00,2);
  LCM_WrParameter(LCM_GRH_WID,0x1E,0x00,2);
  LCM_WrParameter(LCM_CUR_SHP|0x01,0,0,0);
  LCM_WrParameter(LCM_MOD_OR,0,0,0);
  LCM_WrParameter(LCM_DIS_SW|0x08,0,0,0);
}
/************************************************************
********************************************
** º¯ÊýÃû³Æ: GUI_FillSCR()
**                                                      ¹¦ÄÜÃèÊö:
È«ÆÁÌî³ä¡£Ö±½ÓÊ¹ÓÃÊý¾ÝÌî³äÏÔÊ¾»º³åÇø¡£¸ù¾ÝLCMµÄÊµ¼ÊÇé¿ö
±àÐ´´Ëº¯Êý
** Êä¡¡Èë: dat          Ìî³äµÄÊý¾Ý
** Êä¡¡³ö: ÎÞ
** È«¾Ö±äÁ¿: ÎÞ
** µ÷ÓÃÄ£¿é: ÎÞ
** Modified by:
```

```
** Modified date:
**-------------------------------------------------------------------------------------
--------------
**************************************************************
********************************************/
void  GUI_FillSCR(TCOLOR dat)
{
        uint32  i;
        LCM_WrParameter(LCM_ADD_POS,0x00,0x00,2);
        LCM_WrParameter(LCM_AUT_WR,0x00,0x00,0);
        for(i=0;i<240*128/8;i++)
        {
                //LCM_STA3();
                LCM_WrData(dat);
        }
        LCM_WrParameter(LCM_AUT_OVR,0x00,0x00,0);
        LCM_WrParameter(LCM_ADD_POS,0x00,0x00,2);
}
/*************************************************************
*********************************************
** º¯ÊýÃû³Æ: GUI_Initialize
**                                                                  ¹¦ÄÜÃèÊö:
³õÊ¼»¯GUI£¬°üÀ¨³õÊ¼»¯ÏÔÊ¾»º³åÇø£¬³õÊ¼»¯LCM²¢ÇåÆÁ
** Êä¡¡Èë: ÎÞ
** Êä¡¡³ö: ÎÞ
** È«¾Ö±äÁ¿: ÎÞ
** µ÷ÓÃÄ£¿é: ÎÞ
** Modified by:
** Modified date:
**-------------------------------------------------------------------------------------
--------------
**************************************************************
********************************************/
void  GUI_Initialize(void)
{
        LCM_DispIni();                                   //
³õÊ¼»¯LCMÄ£¿é¹¤×÷Ä£Ê½£¬´¿Í¼ÐÎÄ£Ê½
        GUI_FillSCR(0x00);                          //
³õÊ¼»¯»º³åÇøÎª0x00£¬²¢Êä³öÆÁÄ»(ÇåÆÁ)
}
/*************************************************************
*********************************************
** º¯ÊýÃû³Æ: GUI_Point
** ¹¦ÄÜÃèÊö: ÔÚÖ¸¶¨Î»ÖÃÉÏ»µã
**                                                                       Êä¡¡Èë:
xÖ¸¶¨µãËùÔÚÁÐµÄÎ»ÖÃ£»yÖ¸¶¨µãËùÔÚÐÐµÄÎ»ÖÃ£»colorÏÔÊ¾ÑÕ
É«(¶ÔÓÚºÚ°×É«LCM£¬Îª0Ê±Ãð£¬Îª1Ê±ÏÔÊ¾)
** Êä¡¡³ö: ·µ»ØÖµÎª1Ê±±íÊ¾²Ù×÷³É¹¦£¬Îª0Ê±±íÊ¾²Ù×÷Ê§°Ü
** È«¾Ö±äÁ¿: ÎÞ
** µ÷ÓÃÄ£¿é: ÎÞ
** Modified by:
** Modified date:
**------------------------------------------------------------------------------------
--------------
***********************************************************
*******************************************/


uint8  GUI_Point(uint8 x, uint8 y, TCOLOR color)
{
        uint8 x1;
        uint32 iPos;
        x1          =          x          >>          3;          //
È¡Y·½Ïò·ÖÒ³µØÖ·,ÒòÎª×îÐ¡´æ´¢µ¥ÔªÎª8*8,°´8ÐÐÒ»¸öµ¥Ôª·ÃÎÊ
        iPos          =          (uint32)y          *          0x1e          +
x1;//¼ÆËãµØÖ·:0x1eÊÇÎÄ±¾µÄ¿í¶È,¸ß¶È
        LCM_WrParameter(LCM_ADD_POS,iPos&0xff,iPos/256,2);//·Ö±
ðÈ¡³öµÍµØÖ·£¬¸ßµØÖ·;Ð´ÈëLCD
        x1 = turnf[ x & 0x07 ];//¼ÆËã¾ßÌåµÄÐÐ
        //uint8 const  turnf[8] = {7,6,5,4,3,2,1,0};
        color = color <<3;
        x1          =          LCM_BIT_OP|x1|color;          //
×Ö½ÚÄÚÎ»ÖÃ¼ÆËã,LCM_BIT_OPÎªÎ»²Ù×÷Ö¸Áî
        /*Î»²Ù×÷£º
        1 1 1 1 N3 N2 N2 N0
        ÎÞ²ÎÊý
        ¸ÃÖ¸Áî¿É½«ÏÔÊ¾»º³åÇøÄ³µ¥ÔªµÄÄ³Ò»Î»ÇåÁã»òÖÃ1£¬¸Ãµ¥Ô
ªµØÖ·ÓÉµ±Ç°µØÖ·Ö¸ÕëÌá¹©¡£
        N3£½1ÖÃ1£¬N3£½0
ÇåÁã¡£N2£N0£º²Ù×÷Î»¶ÔÓ¦µ¥ÔªµÄD0£D7Î»¡£*/
        LCM_WrParameter(x1,0,0,0);;
        return  1;
}


/***********************************************************
*******************************************
** º¯ÊýÃû³Æ: GUI_ReadPoint
**                                                                    ¹¦ÄÜÃèÊö:
¶ÁÈ¡Ö¸¶¨µãµÄÑÕÉ«¡£¶ÔÓÚµ¥É«£¬ÉèÖÃretµÄd0Î»Îª1»ò0£¬4¼¶»Ò¶È
ÔòÎªd0¡¢d1ÓÐÐ§£¬8Î»RGBÔòd0--
d7ÓÐÐ§£¬RGB½á¹¹ÔòR¡¢G¡¢B±äÁ¿ÓÐÐ§
**          Êä¡¡Èë:          xÖ¸¶¨µãËùÔÚÁÐµÄÎ»ÖÃ£»
yÖ¸¶¨µãËùÔÚÐÐµÄÎ»ÖÃ£»ret±£´æÑÕÉ«ÖµµÄÖ¸Õë
** Êä¡¡³ö: ·µ»Ø0±íÊ¾Ö¸¶¨µØÖ·³¬³ö»º³åÇø·¶Î§
** È«¾Ö±äÁ¿: ÎÞ
** µ÷ÓÃÄ£¿é: ÎÞ
** Modified by:
** Modified date:
**------------------------------------------------------------------------------------
```

```
--------------
************************************************************
*******************************************/
uint8  GUI_ReadPoint(uint8 x, uint8 y, TCOLOR *ret)
{
        TCOLOR       bak;
        uint8   x1;
        bak = LCM_ReadByte(x,y);
        x1 = turnf[ x & 0x07 ];
        if( (bak & (DEC_HEX_TAB[x1&0x07]) ) ==0)
                *ret = 0x00;
        else
                *ret = 0x01;
        return  1;
}
/************************************************************
********************************************
** º¯ÊýÃû³Æ: GUI_HLine
** ¹¦ÄÜÃèÊö: »Ë®Æ½Ïß£¬²Ù×÷Ê§°ÜÔÒòÊÇÖ¸¶¨µØÖ·³¬³ö»º³åÇø·¶Î§
** Êä¡¡Èë: x0    Ë®Æ½ÏßÆðµãËùÔÚÁÐµÄÎ»ÖÃ
*          y0    Ë®Æ½ÏßÆðµãËùÔÚÐÐµÄÎ»ÖÃ
*          x1    Ë®Æ½ÏßÖÕµãËùÔÚÁÐµÄÎ»ÖÃ
*          color ÏÔÊ¾ÑÕÉ«(¶ÔÓÚºÚ°×É«LCM£¬Îª0Ê±Ãð£¬Îª1Ê±ÏÔÊ¾)
** Êä¡¡³ö: ÎÞ
** È«¾Ö±äÁ¿: ÎÞ
** µ÷ÓÃÄ£¿é: ÎÞ
** Modified by:
** Modified date:
**-------------------------------------------------------------------------------------
--------------
************************************************************
*******************************************/
void  GUI_HLine(uint8 x0, uint8 y0, uint8 x1, TCOLOR color)
{  uint8  bak;
  if(x0>x1)                                          //
¶Ôx0¡¢x1´óÐ¡½øÐÐÅÅÁÐ£¬ÒÔ±ã»Í¼
  {  bak = x1;
     x1 = x0;
     x0 = bak;
  }
  do
  {  GUI_Point(x0, y0, color);                    // ÖðµãÏÔÊ¾£¬Ãè³ö´¹Ö±Ïß
     x0++;
  }while(x1>=x0);
}

/************************************************************
********************************************
** º¯ÊýÃû³Æ: GUI_RLine
** ¹¦ÄÜÃèÊö: »ÊúÖ±Ïß¡£
```

	** Êä¡¡Èë: x0 Ë®Æ½ÏßÆðµãËùÔÚÁÐµÄÎ»ÖÃ
	* y0 Ë®Æ½ÏßÆðµãËùÔÚÐÐµÄÎ»ÖÃ
	* x1 Ë®Æ½ÏßÖÕµãËùÔÚÁÐµÄÎ»ÖÃ
	* color ÏÔÊ¾ÑÕÉ«(¶ÔÓÚºÚ°×É«LCM£¬Îª0Ê±Ãð£¬Îª1Ê±ÏÔÊ¾)
	** Êä¡¡³ö: ÎÞ
	** È«¾Ö±äÁ¿: ÎÞ
	** µ÷ÓÃÄ£¿é: ÎÞ
	** Modified by:
	** Modified date:
	**-- 	
	-------------- 	
	**
	**/
	void GUI_RLine(uint8 x0, uint8 y0, uint8 y1, TCOLOR color)
	{ uint8 bak;
	  if(y0>y1) //
	¶Ôx0¡¢x1´óÐ¡½øÐÐÅÅÁÐ£¬ÒÔ±ã»Í¼
	   { bak = y1;
	      y1 = y0;
	      y0 = bak;
	   }
	   do
	   { GUI_Point(x0, y0, color); // ÖðµãÏÔÊ¾£¬Ãè³ö´¹Ö±Ïß
	      y0++;
	   }while(y1>=y0);
	}
	
	/**
	***
	** End Of File
	**
	**/

Code (startup . c)	;** **/ ;define the stack size ;¶¨Òå¶ÑÕ»µÄ´óÐ¡ SVC_STACK_LEGTH EQU 0 FIQ_STACK_LEGTH EQU 0 IRQ_STACK_LEGTH EQU 512 ABT_STACK_LEGTH EQU 0 UND_STACK_LEGTH EQU 0 NoInt EQU 0x80 USR32Mode EQU 0x10 SVC32Mode EQU 0x13 SYS32Mode EQU 0x1f IRQ32Mode EQU 0x12

```
FIQ32Mode   EQU 0x11

   IMPORT __use_no_semihosting_swi

;The imported labels
;ÒýÈëµÄÍâ²¿±êºÅÔÚÕâÉùÃ÷
   IMPORT  FIQ_Exception                 ;Fast interrupt exceptions handler
¿ìËÙÖÐ¶ÏÒì³£´¦Àí³ÌÐò
   IMPORT  __main                        ;The entry point to the main function
CÓïÑÔÖ÷³ÌÐòÈë¿Ú
   IMPORT   TargetResetInit                          ;initialize the target board
Ä¿±ê°å»ù±¾³õÊ¼»¯

;The emported labels
;¸øÍâ²¿Ê¹ÓÃµÄ±êºÅÔÚÕâÉùÃ÷
   EXPORT  bottom_of_heap
   EXPORT  StackUsr

   EXPORT  Reset
   EXPORT __user_initial_stackheap

   CODE32
   PRESERVE8
   AREA    vectors,CODE,READONLY
      ENTRY

;interrupt vectors
;ÖÐ¶ÏÏòÁ¿±í
Reset
      LDR     PC, ResetAddr
      LDR     PC, UndefinedAddr
      LDR     PC, SWI_Addr
      LDR     PC, PrefetchAddr
      LDR     PC, DataAbortAddr
      DCD      0xb9205f80
      LDR     PC, [PC, #-0xff0]
      LDR     PC, FIQ_Addr

ResetAddr          DCD     ResetInit
UndefinedAddr       DCD     Undefined
SWI_Addr           DCD     SoftwareInterrupt
PrefetchAddr        DCD     PrefetchAbort
DataAbortAddr       DCD     DataAbort
Nouse              DCD     0
IRQ_Addr           DCD     0
FIQ_Addr           DCD     FIQ_Handler


;Î´¶¨ÒåÖ¸Áî
Undefined
```

```
      B      Undefined

;ÈíÖÐ¶Ï
SoftwareInterrupt
      B      SoftwareInterrupt

;È¡Ö¸ÁîÖÐÖ¹
PrefetchAbort
      B      PrefetchAbort

;È¡Êý¾ÝÖÐÖ¹
DataAbort
      B      DataAbort

;¿ìËÙÖÐ¶Ï
FIQ_Handler
      STMFD   SP!, {R0-R3,R12,LR}
      BL      FIQ_Exception
      LDMFD   SP!, {R0-R3,R12,LR}
      SUBS    PC,  LR,  #4


;**************************************************************
*********************************************/
InitStack
      MOV     R0, LR
;Build the SVC stack
;ÉèÖÃ¹ÜÀíÄ£Ê½¶ÑÕ»
      MSR     CPSR_c, #0xd3
      LDR     SP, StackSvc
;Build the IRQ stack
;ÉèÖÃÖÐ¶ÏÄ£Ê½¶ÑÕ»
      MSR     CPSR_c, #0xd2
      LDR     SP, StackIrq
;Build the FIQ stack
;ÉèÖÃ¿ìËÙÖÐ¶ÏÄ£Ê½¶ÑÕ»
      MSR     CPSR_c, #0xd1
      LDR     SP, StackFiq
;Build the DATAABORT stack
;ÉèÖÃÖÐÖ¹Ä£Ê½¶ÑÕ»
      MSR     CPSR_c, #0xd7
      LDR     SP, StackAbt
;Build the UDF stack
;ÉèÖÃÎ´¶¨ÒåÄ£Ê½¶ÑÕ»
      MSR     CPSR_c, #0xdb
      LDR     SP, StackUnd
;Build the SYS stack
;ÉèÖÃÏµÍ³Ä£Ê½¶ÑÕ»
      MSR     CPSR_c, #0x5f
      LDR     SP, =StackUsr
```

```
        MOV     PC, R0


;************************************************************
*******************************************/
ResetInit

        BL      InitStack              ;³õÊ¼»¯¶ÑÕ» Initialize the stack
        BL       TargetResetInit         ;Ä¿±ê°å»ù±¾³õÊ¼»¯ Initialize the target
board
                               ;Ìø×ªµ½cÓïÑÔÈë¿Ú Jump to the entry point of C
program
        B       __main

;************************************************************
*******************************************/
__user_initial_stackheap
    LDR   r0,=bottom_of_heap
;    LDR   r1,=StackUsr
    MOV   pc,lr


StackSvc          DCD     SvcStackSpace + (SVC_STACK_LEGTH - 1)* 4
StackIrq          DCD     IrqStackSpace + (IRQ_STACK_LEGTH - 1)* 4
StackFiq          DCD     FiqStackSpace + (FIQ_STACK_LEGTH - 1)* 4
StackAbt          DCD     AbtStackSpace + (ABT_STACK_LEGTH - 1)* 4
StackUnd          DCD     UndtStackSpace + (UND_STACK_LEGTH - 1)* 4



;/* ·ÖÅä¶ÑÕ»¿Õ¼ä */
        AREA    MyStacks, DATA, NOINIT, ALIGN=2
SvcStackSpace       SPACE   SVC_STACK_LEGTH * 4  ;Stack spaces for
Administration Mode ¹ÜÀíÄ£Ê½¶ÑÕ»¿Õ¼ä
IrqStackSpace       SPACE   IRQ_STACK_LEGTH * 4  ;Stack spaces for
Interrupt ReQuest Mode ÖÐ¶ÏÄ£Ê½¶ÑÕ»¿Õ¼ä
FiqStackSpace     SPACE   FIQ_STACK_LEGTH * 4  ;Stack spaces for Fast
Interrupt reQuest Mode ¿ìËÙÖÐ¶ÏÄ£Ê½¶ÑÕ»¿Õ¼ä
AbtStackSpace       SPACE   ABT_STACK_LEGTH * 4  ;Stack spaces for
Suspend Mode ÖÐÖ¹ÒåÄ£Ê½¶ÑÕ»¿Õ¼ä
UndtStackSpace      SPACE   UND_STACK_LEGTH * 4  ;Stack spaces for
Undefined Mode Î´¶¨ÒåÄ£Ê½¶ÑÕ»


        AREA    Heap, DATA, NOINIT
bottom_of_heap    SPACE    1

        AREA    Stacks, DATA, NOINIT
StackUsr
```

	END ;/** ** ;** End Of File ;** **/