Here is a template for your reference.



An experiment report on

**“Snake Game Player”**

Created by

ID: 2011561103

Class: 20lc

Under The Guidance of

**Dr. Liu Chun**

**(**Assistant professor, Dept. of CST)

**Department of Computer Science & Technology**

**Hubei University of Technology**

**Wuhan, Hubei Province, P. R. China**

**Content**

**CHAPTER PAGE NO.**

**List of Figure i**

**List of Table ii**

**Acknowledgment iii**

**Abstract iv**

**1. INTRODUCTION 1-2**

**1.1 Introduction 1**

**1.2 Objectives 1**

**1.3 Project Description 1**

**2. HARDWARE DESIGN 2-10**

**2.1 Hardware Diagram 2**

2.1.1 The Hardware Diagram consists of 2

**2.2 Circuit Diagram 3**

2.2.1 Components Used in the Circuit 4

2.2.2 Components Description 4

**3. SOFTWARE DESIGN 11-23**

**3.1 Software Diagram 11**

**3.2 Flow Chart 11**

**3.3 Function 12-18**

3.3.1 Draw a robust snake 12

3.3.2 Double linked list 13

3.3.3 Back Insertion 14

3.3.4 Typical D-link list operation 15

3.3.5 Snake move 16

3.3.6 Cross the boundary 16

3.3.7 Food reproduce 16

3.3.8 snake\_map[] 17

**3.4 Another method to design software 19-23**

3.4.1 Big point 19

3.4.2 Snake 29

3.4.3 Food 20

3.4.4 Snake move 21

3.4.5 Initial values used in the I/O ports 22

**4. SYSTEM DESIGN 24-27**

**4.1 Complete System Diagram (Hardware) 24**

**4.2 Complete System Diagram (Software) 24**

4.2.1 Game Running 25

4.2.2 Game Over 26

4.2.3 New Game 26

**5. DISCUSSION & REFERENCE 27**

**5.1 Discussions 27**

**5.2 References 27**

**6. CODE 28-83**

**6.1 main.c 28**

**6.2 lcddrv.c 36**

**6.3 target.c 50**

**6.4 Startup.s 57**

**6.5 config.h 63**

**6.6 lcddrv.h 67**

**6.7 LPC2124.h 68**

**6.8 target.h 81**

**6.9 LPC2124.sct 83**

**List of Figure**

**FIGURE PAGE NO.**

**Figure 1: Hardware Diagram 2**

**Figure 2: Circuit Diagram 3**

**Figure 3: LPC2124 4**

**Figure 4: Pin Diagram of LPC 2124 5**

**Figure 5: LM3228 LCD 7**

**Figure 6: 4X4 KEYPAD 8**

**Figure 7: KEYPAD PINOUT 9**

**Figure 8: 10K Resistor 10**

**Figure 9: Software Diagram 11**

**Figure 10: Flowchart 11**

**Figure 11: Nine small cells 12**

**Figure 12: Doubly Linked List Node 13**

**Figure 13: Back insertion 14**

**Figure 14: Complete System Diagram (Hardware) 24**

**Figure 15: Complete System Diagram (Software) 24**

**Figure 16: Game running-1 25**

**Figure 17: Game running-2 25**

**Figure 18: Game over 26**

**Figure 19: New game 26**

**TABLE PAGE NO.**

**Table 1: Components Used in the Circuit 3**

**Table 2: PINSEL Table 6**

**Table 3: LM3228 LCD Pin Description 8**

**Table 4: Description of KEYPAD Pin 9**

**Acknowledgment**

We offer appreciation to project manager Asst. Prof. Dr. Liu Chun (Dept. Of Computer Science and Technology, HBUT) for directing us all through the task, the work and giving us addresses and making sense of the undertaking and aiding when we were stops in the center during the advancement stage. We could never have constructed such a difficult task without his assistance.

**Abstract**

This undertaking intends to bring the tomfoolery and effortlessness of snake game for certain new highlights. It will incorporate PC controlled canny rivals whose point will be to challenge the human players. It will likewise have the multiplayer highlight that will permit more than one players to play the game over an organization.

This undertaking investigates another aspect in the conventional snake game to make it really fascinating and testing. The effortlessness of this game creates it an ideal possibility for a minor undertaking as we can zero in on cutting edge subjects like multiplayer usefulness and execution of PC controlled shrewd rivals.

**Chapter 1**

**1.1 Introduction**

Project documentation is concerned with describing the delivered software product, in this case the Snake game project. Project documentation includes user documentation which tells users how to use the software product and system documentation which is principally intended for further development and understanding.

**1.2 Objectives**

Snake game is a mobile action game, whose goal is to control a snake to move and collect food in a map. In this paper we develop a controller based on movement rating functions considering smoothness, space, and food. Scores given by these functions are aggregated by linear weighted sum, and the snake takes the action that leads to the highest score. To find a set of good weight values, we apply an evolutionary algorithm. We examine several algorithm variants of different crossover and environmental selection operators. Experimental results show that our design method is able to generate smart controllers.

**1.3 Project Description**

Snake Game project is a mobile game application. This game is created for the recreation of people. To create this game we actually need some hardware device and software application. For hardware we mainly need micro-controller, LCD and Keypad. Now how this game may played.

Firstly Welcome screen,

This is the starting screen in your game, on pressing s should start the game. On this screen, you have to provide an overview of the scoring system and the controls.

Secondly Moving the Player

Currently, the snake doesn't move at all. Let's make it move. To Do:

• The snake should move up, down, left, and right when you press those arrows.

• The snake shouldn't have a sudden change in direction. It should not move left immediately if it is moving right or vice versa. Same is true for up and down movements i.e., if the snake is going up and the down arrow is pressed, nothing should happen. That is supposed to be true for every opposite direction The enumeration for UP, DOWN, LEFT and RIGHT are defined in (key.h). Snake should eat the food

If you get the snake moving, you will see that it just walk (slithers) over the food right now. To Do:

• Make it eat the food.

• It should grow by size if it eats a growing food and should reduce in size if it eats a reducing (junk) food. These food are indicated by 'X' and 'O' on the board.

• Once eaten, the food should disappear and another food (of random type) should be spawn at a random location on the game window.

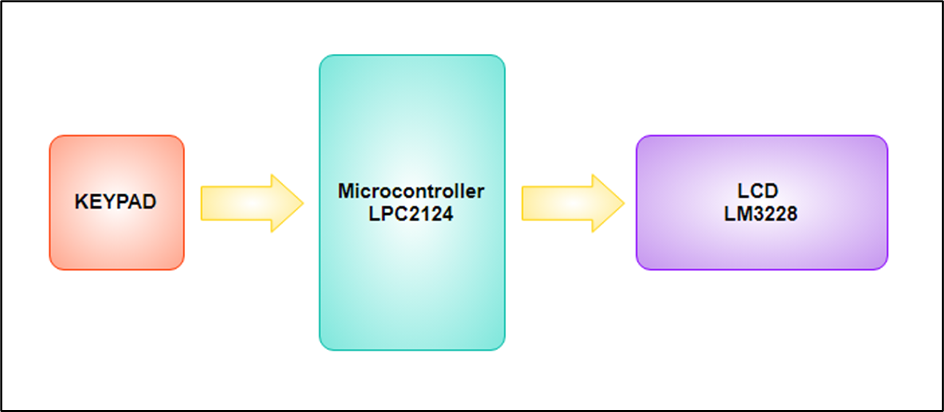
Show points

No points are printed right now. To Do:

• Calculate the points as per the food eaten\*

**2. HARDWARE DESIGN**

**2.1 Hardware Diagram**

****

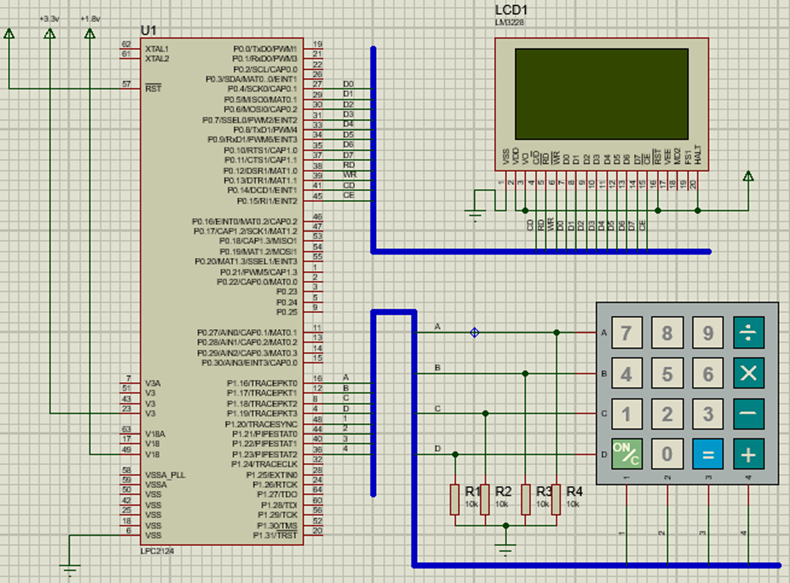
**Figure 1: Hardware Diagram**

**2.1.1 The Hardware Diagram consists of**

1. **Microcontroller LPC2124**
2. **LCD LM3228**
3. **KEYPAD**

**2.2 Circuit Diagram**

**2.2.1 Components Used in the Circuit**



**Figure 2: Circuit Diagram**

**2.2.2 Components Description**

|  |  |  |
| --- | --- | --- |
| **SL. No.** | **Components** | **Specification** |
| 1 | Microcontroller | LPC2124 |
| 2 | LCD | LM3228 |
| 3 | KEYPAD | SMALLCALC |
| 4 | Resistors | 10K |

**Microcontroller**

In this project we’re using the NXP (founded by Philips) LPC2124 Microcontroller. The NXP LPC2124 is an ARM7TDMI-S based high-performance 32-bit RISC Microcontroller with Thumb extensions 256KB on-chip Flash ROM with In-System Programming (ISP) and In-Application Programming (IAP) 16KB RAM, Vectored Interrupt Controller, Two UARTs, I2C serial interface, 2 SPI serial interfaces, Two timers (7 capture/compare channels), PWM unit with up to 6 PWM outputs, 4-channels 10bit ADC, Real Time Clock, Watchdog Timer, General purpose I/O pins. CPU clock up to 60 MHz, On-chip crystal oscillator and On-chip PLL.

****

**Figure 3: LPC2124**

**Basics of LPC2124**

When coming to ARM7 Programming there are 5 things you need to be get familiarize with. They are-

* PINSEL
* IODIR
* IOSET
* IOCLR
* IOPIN

****

**Figure 4: Pin Diagram of LPC 2124**

**PINSEL:**

A 32 bit register which is used to select the function of the pins in which the user needs it to operate. There are four functions for each pins of the controller, in which the first function one was GPIO (General Purpose Input Output). It means that the pin can either act as an Input or Output with no specific functions

.There are totally three PINSEL register in LPC2124 Controller in order to control the functions of the Pins in the respective ports. The classification is given below

PINSEL0 – Controls functions of Port0.0 – Port0.15

PINSEL1 – Controls functions of Port0.16-Port0.31

PINSEL2 – Controls functions of Port1.16-Port1.31

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Pin Name** | **Function when 00** | **PINSEL0** | **Function when 01** | **Function when 10** | **Function when 11** | **Reset Value** |
| P0.0 | GPIO Port 0.0 | 1:0 | TxD (UART0) | PWM1 | Reserved | 00 |
| P0.1 | GPIO Port 0.1 | 3:2 | RxD (UART0) | PWM3 | EINT0 | 00 |
| P0.2 | GPIO Port 0.2 | 5:4 | SCL (I2C) | Capture 0.0 (TIMER0) | Reserved | 00 |
| P0.3 | GPIO Port 0.3 | 7:6 | SDL (I2C) | Match 0.0 (TIMER0) | EINT1 | 00 |
| P0.4 | GPIO Port 0.4 | 9:8 | SCK (SPI0) | Capture 0.1 (TIMER0) | Reserved | 00 |
| P0.5 | GPIO Port 0.5 | 11:10 | MISO (SPI0) | Match 0.1 (TIMER0) | Reserved | 00 |
| P0.6 | GPIO Port 0.6 | 13:12 | MOSI (SPI0) | Capture 0.2 (TIMER0) | Reserved | 00 |
| P0.7 | GPIO Port 0.7 | 15:14 | SSEL (SPI0) | PWM2 | EINT2 | 00 |
| P0.8 | GPIO Port 0.8 | 17:16 | TxD (UART1) | PWM4 | Reserved | 00 |
| P0.9 | GPIO Port 0.9 | 19:18 | RxD (UART1) | PWM6 | EINT3 | 00 |
| P0.10 | GPIO Port 0.10 | 21:20 | RTS (UART1) | Capture 1.0 (TIMER1) | Reserved | 00 |
| P0.11 | GPIO Port 0.11 | 23:22 | CTS (UART1) | Capture 1.1 (TIMER1) | Reserved | 00 |
| P0.12 | GPIO Port 0.12 | 25:24 | DSR (UART1) | Match 1.0 (TIMER1) | Reserved | 00 |
| P0.13 | GPIO Port 0.13 | 27:26 | DTR (UART1) | Match 1.1 (TIMER1) | Reserved | 00 |
| P0.14 | GPIO Port 0.14 | 29:28 | CD (UART1) | EINT1 | Reserved | 00 |
| P0.15 | GPIO Port 0.15 | 31:30 | RI (UART1) | EINT2 | Reserved | 00 |

**Table 2: PINSEL Table**

**IODIR:**

Like DDR in AVR and TRIS in PIC, ARM uses IODIR register to specify the direction which in which we are going to use the pins. Two 32 bit registers IODIR0 for Port0 (P0.0 – P0.31) and IODIR1 for Port (P1.16- P1.31). Kindly note that loading values in IODIR, it will take effect only if the Pins are used as GPIO and the directions are controlled automatically if it was specified with any special functions

**IOSET:**

This Register is meant to set the pins in the Ports where writing 1 to it will set the respective pin while 0 will have no effect. There are two registers dedicated for both the ports IOSET0 –P0.0 – P0.31 and IOSET1 for P1.16 – P1.31

**IOCLR:**

This Register is meant to clear the pins in the Ports where writing 1 will clear the respective pin while 0 will have no effect in the Ports. There are two registers dedicated for both the ports IOCLR0 –P0.0 – P0.31 and IOCLR1 for P1.16 – P1.31

**IOPIN:**

This is used only when we assign certain pins as Input in the IODIR register. There are two registers dedicated for both the ports IOPIN0 –P0.0 – P0.31 and IOPIN1 for P1.16 – P1.

**2. LCD**

In this project we’re using LM3228 LCD (Liquid Crystal Display). It’s a 128 x 64 Dots Graphic LCD. It has 20 pin to control whole things.

**FEATURES**

• 128 x 64 dots + 4 Icons

• Built-in controller (KS0108)

• + 5V power supply

• 1/64 duty cycle

• EL backlight (built-in EL inverter)

• Built-in N.V



**Figure 5: LM3228 LCD**

|  |  |  |
| --- | --- | --- |
| **PIN NUMBER** | **SYMBOL** | **FUNCTION** |
| 1 | VSS | Ground (0V) |
| 2 | VDD | Logic Supply Voltage (+5V) |
| 3 | VO | LCD drive voltage for contrast adjustment |
| 4 | C/D | WR=“L”...C/D=“H” : Command write C/D=“L”: Data write  RD=“L”...C/D=“H” : Status read C/D=“L”: Data read |
| 5 | RD | Data read Active Low |
| 6 | WR | Data write Active Low |
| 7 | D0 | Data Bus Line 0 |
| 8 | D1 | Data Bus Line 1 |
| 9 | D2 | Data Bus Line 2 |
| 10 | D3 | Data Bus Line 3 |
| 11 | D4 | Data Bus Line 4 |
| 12 | D5 | Data Bus Line 5 |
| 13 | D6 | Data Bus Line 6 |
| 14 | D7 | Data Bus Line 7 |
| 15 | CE | Chip enable Active Low |
| 16 | RST | Chip reset Active Low |
| 17 | VEE | Negative voltage input for LC drive (Negative voltage output for  models with on-board negative voltage generator) |
| 18 | MD2 | Mode Selection |
| 19 | FS1 | Terminals for selection of font size |
| 20 | HALT | Halt Function (H = Normal, L = Stop Oscillation) |

**Table 3: LM3228 LCD Pin Description**

**3. KEYPAD**

We’re using **4X4 KEYPAD in this project**



**Figure 6: 4X4 KEYPAD**

**4X4 KEYPAD Pin Configuration**

4X4 KEYPAD MODULES are available in different sizes and shapes. But they all have same pin configuration. It is easy to make 4X4 KEYPAD by arranging 16 buttons in matrix formation by yourself.

****

**Figure 7: KEYPAD PINOUT**

|  |  |
| --- | --- |
| **Pin Number** | **Description** |
| 1 | PIN1 is taken out from 1st ROW |
| 2 | PIN1 is taken out from 2nd ROW |
| 3 | PIN1 is taken out from 3rd ROW |
| 4 | PIN1 is taken out from 4th ROW |
| 5 | PIN1 is taken out from 1st COLUMN |
| 6 | PIN1 is taken out from 2nd COLUMN |
| 7 | PIN1 is taken out from 3rd COLUMN |
| 8 | PIN1 is taken out from 4th COLUMN |

**Table 4: Description of KEYPAD Pin**

**10K Resistor**

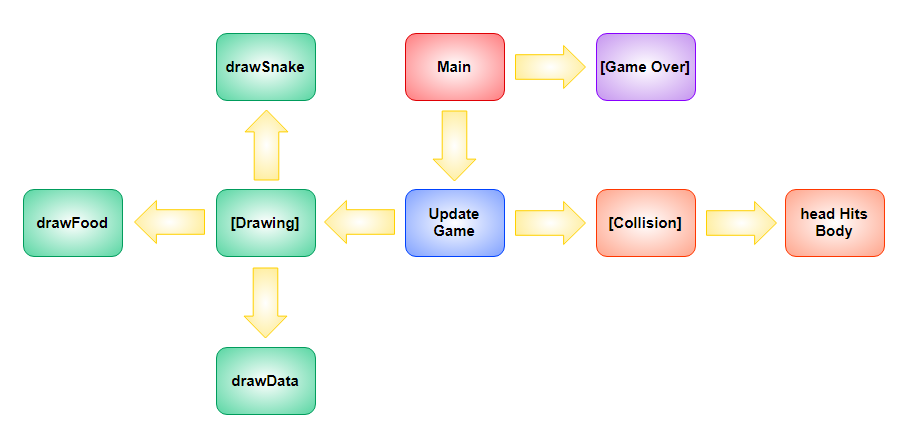
A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses. Here we using 10K resistor

****

**Figure 8: 10K Resistor**

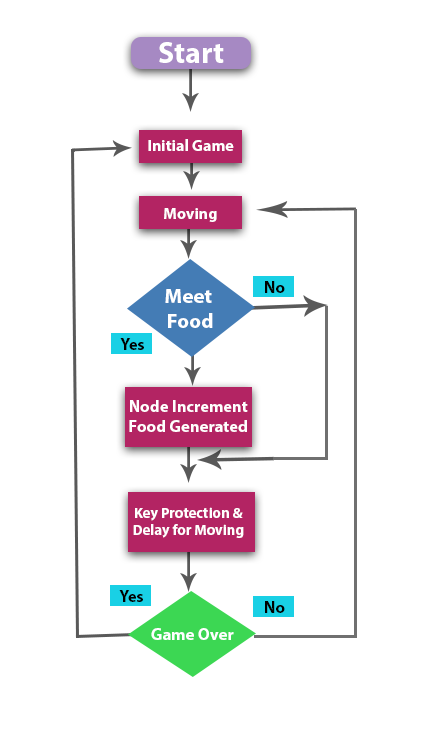
**3. SOFTWARE DESIGN**

**3.1 Software Diagram**

****

**Figure 9: Software Diagram**

**3.2 Flow Chart**

****

**Figure 10: Flowchart**

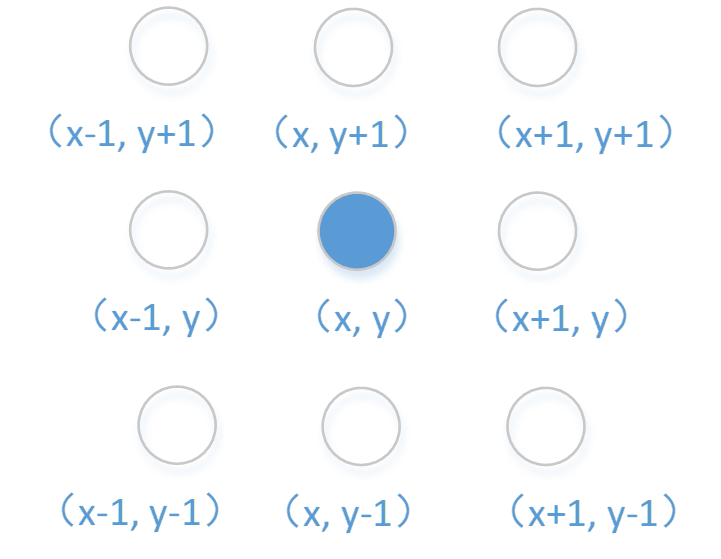
**3.3 Function**

**3.3.1 Draw a robust snake**

**. nine small cells**

**.We should draw a robust snake instead of just one point**

**void Display(uint8 x,uint8 y)**



**Figure 11: Nine small cells**

**{**

**GUI\_Point(x, y, LCD\_DISP\_COLOR);**

**GUI\_Point(x+1, y, LCD\_DISP\_COLOR);**

**GUI\_Point(x-1, y, LCD\_DISP\_COLOR);**

**GUI\_Point(x, y+1, LCD\_DISP\_COLOR);**

**GUI\_Point(x, y-1, LCD\_DISP\_COLOR);**

**GUI\_Point(x+1, y+1, LCD\_DISP\_COLOR);**

**GUI\_Point(x+1, y-1, LCD\_DISP\_COLOR);**

**GUI\_Point(x-1, y+1, LCD\_DISP\_COLOR);**

**GUI\_Point(x-1, y-1, LCD\_DISP\_COLOR);**

**}**

**void Clear(uint8 x,uint8 y)**

**{**

**GUI\_Point(x, y, LCD\_BACK\_COLOR);**

**GUI\_Point(x+1, y, LCD\_BACK\_COLOR);**

**GUI\_Point(x-1, y, LCD\_BACK\_COLOR);**

**GUI\_Point(x, y+1, LCD\_BACK\_COLOR);**

**GUI\_Point(x, y-1, LCD\_BACK\_COLOR);**

**GUI\_Point(x+1, y+1, LCD\_BACK\_COLOR);**

**GUI\_Point(x+1, y-1, LCD\_BACK\_COLOR);**

**GUI\_Point(x-1, y+1, LCD\_BACK\_COLOR);**

**GUI\_Point(x-1, y-1, LCD\_BACK\_COLOR);**

**}**

**3.3.2 Double linked list**

**struct part**

**{**

**uint8 x, y;**

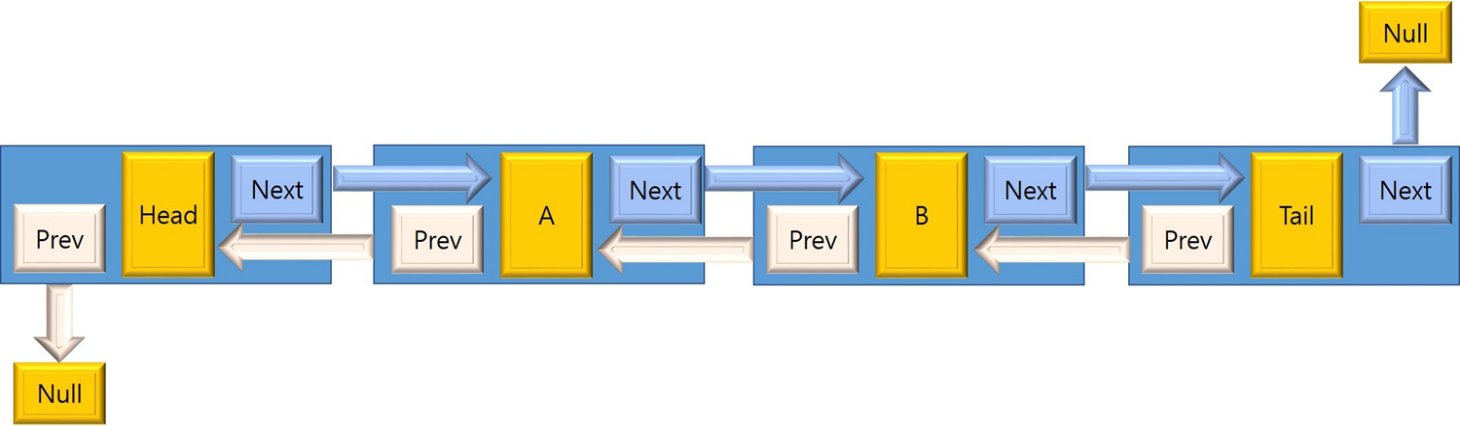
**struct part \*next,\*per;**

**};**

**head\_x=21;head\_y=10;**

**struct part \*head,\*tail;**

**uint8 snake\_map[42][21];**

****

**Figure 12: Doubly Linked List Node**

**3.3.3 The back insertion mirrors the front insertion.**

**The back insertion mirrors the front insertion.**

1. **Create new node (C).**
2. **Check if list is empty.**
   * **If head is empty**
     1. **set C as the new head. Existence of head node indicates that there are elements in the list (well, at least for this implementation).**
   * **Otherwise, check if tail is null.**
     1. **If tail is null**
        1. **Set next node pointer of current head (B) to point to new node (C).**
        2. **Set prev node pointer of new node (C) to point at current head (B).**
     2. **Otherwise,**
        1. **Set the prev node of C to point at current tail (B).**
        2. **Set the next node of B to point at new node (C).**
   * **Assign the newly created node (C) as the new tail.**
   * **Set next node** of **C** **to null.**

Diagram

Description automatically generated

**Figure 13: Back insertion**

**3.3.4 Typical D-link list operation**

**int dlink\_append\_last(void \*pval)**

**{**

**node \*pnode=create\_node(pval);**

**if (!pnode)**

**return -1;**

**pnode->next = phead; //insert before phead**

**pnode->prev = phead->prev;**

**phead->prev->next = pnode;**

**phead->prev = pnode;**

**count++;**

**return 0;**

**void Init\_snake() //control snake with dlink**

**{ struct part \*body;**

**head=(struct part\*)malloc(sizeof(struct part));**

**tail=(struct part\*)malloc(sizeof(struct part));**

**body=(struct part\*)malloc(sizeof(struct part));**

**body->x=21;**

**body->y=10;**

**body->next=NULL;**

**body->per=NULL;**

**snake\_map[21][10]=1;**

**head->next=body;**

**tail->per=body;**

**size=1;**

**Display(tail->per->x\*3+1,tail->per->y\*3+1);**

**//begin at the middle point 64\*32**

**}**

**3.3.5 Snake move**

**Move(dis){**

**if(dis==1) head\_x++; // 1:move to right side**

**if(dis==2) head\_x--; // 2: move to left side**

**if(dis==3) head\_y++; // 3: move down**

**if(dis==4) head\_y--; // 4: move up**

**}**

**3.3.6 Cross the boundary**

**if(head\_x==43) head\_x=1;//42\*3+1=127**

**if(head\_x==0) head\_x=42;**

**if(head\_y==20) head\_y=1;**

**if(head\_y==0) head\_y=20;**

**3.3.7 Food reproduce**

**if(head\_x==X&&head\_y==Y)**

**{**

**food();**

**size++;**

**}**

**void food()**

**{**

**while(1)**

**{**

**X=rand()%40;**

**Y=rand()%20;**

**if(X>0&&Y>0)**

**if(snake\_map[X][Y]==0)**

**{**

**Display(X\*3+1,Y\*3+1);**

**break;**

**}**

**}**

**}**

**3.3.8 snake\_map[]**

* **Snake map[] is used to record the location of the snake and decide whether to eat it.**
* **That is to say, each time we only need to determine whether the corresponding point of map is 1.If each snake traverses every point, the time complexity is high.**
* **To judge whether you eat yourself, you have to go through the whole snake, but if I store it in map according to the corresponding coordinates when drawing snakes, I will find the corresponding point directly when judging, so I don't need to go through the snake.**

**unsigned int display\_snake()**

**{**

**struct part \*body;**

**body=(struct part\*)malloc(sizeof(struct part));**

**body->x=head\_x;**

**body->y=head\_y;**

**head->next->per=body;**

**body->next=head->next;**

**head->next=body;**

**Display(head\_x\*3+1,head\_y\*3+1);**

**if(snake\_map[head\_x][head\_y]==1) return 0;**

**else snake\_map[head\_x][head\_y]=1;**

**if(head\_x==X&&head\_y==Y) {**

**food();**

**size++; }**

**else {**

**snake\_map[tail->per->x][tail->per->y]=0;**

**Clear(tail->per->x\*3+1,tail->per->y\*3+1);**

**tail->per=tail->per->per;**

**free(tail->per->next);**

**}**

**return 1;**

**}**

**3.4 Another method to design software**

3.4.1 Big point

//画一个大点，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);

}

}

**3.4.2 Snake**

//初始化一条蛇，长度为30，头部在显示屏中心

void snake\_init(int x,int y)

{

int i ;

snake\_length = 0; //初始长度为0

for(i = 0;i < 10;i++) //长度为10的蛇

{

Big\_Point(x,y,LCD\_DISP\_COLOR);

snake[snake\_length].x = x;

snake[snake\_length].y = y;

snake\_length++;

x -= 3;

}

}

**3.4.3 Food**

//随机产生一个食物

void creat\_food()

{

int i,flag = 1;

Big\_Point(food\_x,food\_y,LCD\_BACK\_COLOR); //消除旧点

do//先执行一次，产生一个新事物

{

//srand((unsigned)time(NULL));

food\_x = rand() % 127;

food\_y = rand() % 63;

for(i = 0;i < snake\_length-1;i++) //判断新产生的点是否在蛇身上

{

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); //产生新点

}

**3.4.4 Snake move**

void snake\_move(int direction){//蛇运动

int i;

struct Snake last; //保留最后一个点的坐标

last = snake[snake\_length-1];

//将后面的点向前移动，就是将前面的点坐标保存在后面一个数组节点上

for(i = snake\_length - 1;i > 0;i--) {

snake[i]= snake[i-1]; }

snake[0].x += dir[direction][0] \* 3; //将头向direction移动

snake[0].y += dir[direction][1] \* 3;

if(snake[0].x >= 127)snake[0].x = 1; //判断是否出界

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++) {//判断头是否撞到蛇身

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; //游戏结束

return ; } }

if(snake[0].x+2 >= food\_x && food\_x >= snake[0].x-2 //判断是否吃到食物

&& snake[0].y+2 >= food\_y && food\_y >= snake[0].y-2) {

snake[snake\_length] = last; //吃到食物将食物放到最后，就是之前保存的最后一个点

creat\_food(); //产生新食物

snake\_length++; //蛇身加长 }

else {

Big\_Point(last.x,last.y,LCD\_BACK\_COLOR); //消除蛇尾 }

//重新显示蛇

Big\_Point(snake[0].x,snake[0].y,LCD\_DISP\_COLOR); //将头显示出来}

**3.4.5 Initial values used in the I/O ports**

/获取运动方向

int GetDir()

{

int temp,temp1,temp2;

IO0DIR |= 0x0000F; //列输出

IO0SET |= 0x0000F; //列拉高

temp1 = IO0PIN & 0x000F0000; //判断行

//获取行

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) //有键按下

{

IO0DIR &= 0xFFFFFFF0; //列输入

IO0DIR |= 0x000F0000; //行输出

IO0CLR |= 0x0000000F; //列置0

IO0SET |= 0x000F0000; //行拉高

temp2 = IO0PIN & 0x0000000F;

//获取列

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; //行拉低

IO0DIR &= 0x0FFFF; //行输入

}

return temp;

}

**4. SYSTEM DESIGN**

**4.1 Complete System Diagram (Hardware)**

**Schematic

Description automatically generated**

**Figure 14: Complete System Diagram (Hardware)**

**4.2 Complete System Diagram (Software)**

Graphical user interface, text

Description automatically generated

**Figure 15: Complete System Diagram (Software)**

**4.2.1 Game Running**



Chart

Description automatically generated

**Figure 16: Game running-1**

A screenshot of a computer

Description automatically generated with low confidence

**Figure 17: Game running 2**

**4.2.2 Game Over**

Game playing time, when snake bite his body. The game is over.

Schematic

Description automatically generated

**Figure 18: Game over**

**4.2.3 New Game**

**When game is over then starting a new game.**

Diagram, schematic

Description automatically generated

**Figure 19: New game**

**5. DISCUSSION & REFERENCE**

**5.1 Discussions**

The coding of Snake was extremely difficult with many errors arising. Many systems had to be written numerous ways before a final working solution was found. For example, two different movement methods were used prior to final version; however, even the final version is flawed as vertical movement causes the snake to change scale. There were also issues with the food – snake collision detection. While the final version resulted in a snake that could eat food, the movement glitch caused the food to cause further size issues.

Despite the fact that the game could not truly be played due to the fact no score could be given, the game is still satisfying. With the exception of the size glitch when turning, the snake responds to user input and moves around the screen as directed. Given longer to work on this, the collision detection with the movement would be the first thing fixed. By fixing this, all other sections of code that are currently not working would run. The leaderboard would work as there would be correct scores input, and the snake would grow as the food would cause it to only increase by one and not varying numbers based on direction. In addition, fixing the movement would allow for the snake to die when colliding with itself. In the current state, the snake moves as a matrix so it can not kill itself as it would be impossible to move in any direction. This failure to establish a perfect movement system was the biggest disappointment of the game as all other problems stemmed from it.

For these reasons, it is recommended that anyone who wishes to recreate this game starts simply when writing the code. It is advisable that they first perfect the snake movement controls before messing with the food generation. By taking the code in small sections, it is easier to get individual features to work. Building off this, use functions to contain each aspect of the game. Using functions made it easier to determine where errors were occurring when debugging the code. It also kept the code more organized.

**5.2 References**

[**https://www.w3schools.com/**](https://www.w3schools.com/)

[**https://stackoverflow.com/**](https://stackoverflow.com/)

[**https://www.geeksforgeeks.org/**](https://www.geeksforgeeks.org/)

[**https://wenku.baidu.com/**](https://wenku.baidu.com/)

**6. CODE**

**6.1 main.c**

**/\* Main.c file generated by New Project wizard**

**\***

**\* Created: 周一 10月 3 2016**

**\* Processor: LPC2124**

**\* Compiler: Keil for ARM**

**\*/**

**#include <LPC2124.h>**

**#include "config.h"**

**#include "lcddrv.h"**

**TCOLOR disp\_color;**

**TCOLOR back\_color;**

**int food\_x;**

**int food\_y;**

**int life = 1;**

**int n = 10;**

**int direction = 2;**

**int x[300] = {18, 17, 16, 15, 14, 13, 12, 11, 10, 9};**

**int y[300] = {10, 10, 10, 10, 10, 10, 10, 10, 10, 10};**

**uint8 const pic[]={0xF0,0xF0,0xF0,0xF0};**

**uint8 const \*const map[]=**

**{**

**pic**

**};**

**void GUI\_SetColor(TCOLOR color1, TCOLOR color2)**

**{ GUI\_CopyColor(&disp\_color, color1);**

**GUI\_CopyColor(&back\_color, color2);**

**}**

**void delay()**

**{**

**int i,j;**

**for(i=0;i<256;i++)**

**for(j=1000;j>0;j--);**

**}**

**int code[] =**

**{**

**-1,0,-1,-1,**

**1,-1,2,-1,**

**-1,3,-1,-1,**

**-1,-1,-1,-1**

**};**

**void Show\_Point(int x,int y) //显示点 x取值0-63 y取值0-31**

**{**

**GUI\_Point(4 \* x, 4 \* y, LCD\_DISP\_COLOR);**

**GUI\_Point(4 \* x + 1, 4 \* y, LCD\_DISP\_COLOR);**

**GUI\_Point(4 \* x + 2, 4 \* y, LCD\_DISP\_COLOR);**

**GUI\_Point(4 \* x + 3, 4 \* y, LCD\_DISP\_COLOR);**

**GUI\_Point(4 \* x, 4 \* y + 1 , LCD\_DISP\_COLOR);**

**GUI\_Point(4 \* x + 1, 4 \* y + 1 , LCD\_DISP\_COLOR);**

**GUI\_Point(4 \* x + 2, 4 \* y + 1, LCD\_DISP\_COLOR);**

**GUI\_Point(4 \* x + 3, 4 \* y + 1, LCD\_DISP\_COLOR);**

**GUI\_Point(4 \* x, 4 \* y + 2 , LCD\_DISP\_COLOR);**

**GUI\_Point(4 \* x + 1, 4 \* y + 2 , LCD\_DISP\_COLOR);**

**GUI\_Point(4 \* x + 2, 4 \* y + 2, LCD\_DISP\_COLOR);**

**GUI\_Point(4 \* x + 3, 4 \* y + 2, LCD\_DISP\_COLOR);**

**GUI\_Point(4 \* x, 4 \* y + 3 , LCD\_DISP\_COLOR);**

**GUI\_Point(4 \* x + 1, 4 \* y + 3 , LCD\_DISP\_COLOR);**

**GUI\_Point(4 \* x + 2, 4 \* y + 3, LCD\_DISP\_COLOR);**

**GUI\_Point(4 \* x + 3, 4 \* y + 3, LCD\_DISP\_COLOR);**

**}**

**void Clean\_Point(int x,int y) //x取值0-31 y取值0-15**

**{**

**GUI\_Point(4 \* x, 4 \* y, LCD\_BACK\_COLOR);**

**GUI\_Point(4 \* x + 1, 4 \* y, LCD\_BACK\_COLOR);**

**GUI\_Point(4 \* x + 2, 4 \* y, LCD\_BACK\_COLOR);**

**GUI\_Point(4 \* x + 3, 4 \* y, LCD\_BACK\_COLOR);**

**GUI\_Point(4 \* x, 4 \* y + 1 , LCD\_BACK\_COLOR);**

**GUI\_Point(4 \* x + 1, 4 \* y + 1 , LCD\_BACK\_COLOR);**

**GUI\_Point(4 \* x + 2, 4 \* y + 1, LCD\_BACK\_COLOR);**

**GUI\_Point(4 \* x + 3, 4 \* y + 1, LCD\_BACK\_COLOR);**

**GUI\_Point(4 \* x, 4 \* y + 2 , LCD\_BACK\_COLOR);**

**GUI\_Point(4 \* x + 1, 4 \* y + 2 , LCD\_BACK\_COLOR);**

**GUI\_Point(4 \* x + 2, 4 \* y + 2, LCD\_BACK\_COLOR);**

**GUI\_Point(4 \* x + 3, 4 \* y + 2, LCD\_BACK\_COLOR);**

**GUI\_Point(4 \* x, 4 \* y + 3 , LCD\_BACK\_COLOR);**

**GUI\_Point(4 \* x + 1, 4 \* y + 3 , LCD\_BACK\_COLOR);**

**GUI\_Point(4 \* x + 2, 4 \* y + 3, LCD\_BACK\_COLOR);**

**GUI\_Point(4 \* x + 3, 4 \* y + 3, LCD\_BACK\_COLOR);**

**}**

**void Show\_Line(int x[],int y[],int n) //显示线**

**{**

**int i;**

**for(i = 0; i < n; i++)**

**{**

**Show\_Point(x[i],y[i]);**

**}**

**}**

**void move(int x[],int y[]) //移动**

**{**

**int i, k;**

**int tail\_x, x1;**

**int tail\_y, y1;**

**tail\_x = x[n - 1];**

**tail\_y = y[n - 1];**

**Show\_Line(x, y, n); //画长度为n的线条**

**for(i = n - 1; i > 0; i--)**

**{**

**x[i] = x[i - 1];**

**y[i] = y[i - 1];**

**x[i] &= 31; //x[i]高3位清零**

**y[i] &= 15; //y[i]高4位清零**

**}**

**switch(direction)**

**{**

**case 0: y[0] = y[0] - 1; y[0] &= 15; break;**

**case 1: x[0] = x[0] - 1; x[0] &= 31; break;**

**case 2: x[0] = x[0] + 1; x[0] &= 31; break;**

**case 3: y[0] = y[0] + 1; y[0] &= 15; break;**

**}**

**for(k = 1; k < n; k++)**

**{**

**if(x[0] == x[k] && y[0] == y[k])**

**life = 0;**

**}**

**if(x[0] == food\_x && y[0] == food\_y)**

**{**

**int flag = 1;**

**n++;**

**x[n - 1] = tail\_x;**

**y[n - 1] = tail\_y;**

**while(flag)**

**{**

**int j;**

**flag = 0;**

**x1 = rand()%31;**

**y1 = rand()%15;**

**for(j = 0;j < n;j++)**

**{**

**if(x1 == x[j] && y1 == y[j])**

**flag = 1;**

**}**

**}**

**food\_x = x1;**

**food\_y = y1;**

**}**

**else**

**Clean\_Point(tail\_x, tail\_y);**

**}**

**int GetDir() //获取键盘按键值**

**{**

**int temp = 0;**

**int temp1,temp2;**

**IO0DIR |= 0x000F0000; //列输出**

**IO0SET |= 0x000F0000; //列拉高**

**temp1 = IO0PIN & 0x00F00000; //判断行**

**//获取行**

**switch(temp1)**

**{**

**case 0x00100000:temp1 = 0;break;**

**case 0x00200000:temp1 = 1;break;**

**case 0x00400000:temp1 = 2;break;**

**case 0x00800000:temp1 = 3;break;**

**default:temp1 = 4;**

**}**

**if(temp1 != 4) //有键按下**

**{**

**IO0DIR &= 0xFFF0FFFF; //列输入**

**IO0DIR |= 0x00F00000; //行输出**

**IO0CLR |= 0x000F0000; //列置0**

**IO0SET |= 0x00F00000; //行拉高**

**temp2 = IO0PIN & 0x000F0000;**

**//获取列**

**switch(temp2)**

**{**

**case 0x0010000:temp = temp1\*4 + 0;break;**

**case 0x0020000:temp = temp1\*4 + 1;break;**

**case 0x0040000:temp = temp1\*4 + 2;break;**

**case 0x0080000:temp = temp1\*4 + 3;break;**

**default:temp = 0;**

**}**

**IO0CLR |= 0xF00000; //行拉低**

**IO0DIR &= 0x0FFFFF; //行输入**

**}**

**return temp;**

**}**

**int main (void)**

**{**

**GUI\_Initialize();// 初始化LCM**

**GUI\_SetColor(LCD\_DISP\_COLOR,LCD\_BACK\_COLOR); // 设置前景色和背景色**

**// Write your code here**

**food\_x = rand()%31;**

**food\_y = rand()%15;**

**while (life)**

**{**

**int a = 0;**

**Show\_Point(food\_x, food\_y);**

**a = code[GetDir()];**

**if(a != -1)**

**{**

**if((a + direction) != 3 ) //设置不能回头**

**direction = a;**

**}**

**move(x,y);**

**delay();**

**}**

**}**

**6.2 lcddrv.c**

**#include "config.h"**

**#include "lcddrv.h"**

**#define BUS\_NO 4**

**#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)**

**/\* 定义READ控制 \*/**

**#define LCM\_RD 12**

**#define LCM\_UNREAD() IO0SET = 1<<LCM\_RD**

**#define LCM\_READ() IO0CLR = 1<<LCM\_RD**

**/\* 定义WRITE控制 \*/**

**#define LCM\_WR 13**

**#define LCM\_UNWRITE() IO0SET = 1<<LCM\_WR**

**#define LCM\_WRITE() IO0CLR = 1<<LCM\_WR**

**/\* 定义C/D#控制 \*/**

**#define LCM\_CD 14**

**#define LCM\_COM() IO0SET = 1<<LCM\_CD**

**#define LCM\_DATA() IO0CLR = 1<<LCM\_CD**

**/\* 定义C/D#控制 \*/**

**#define LCM\_CE 15**

**#define LCM\_DISABLE() IO0SET = 1<<LCM\_CE**

**#define LCM\_ENABLE() IO0CLR = 1<<LCM\_CE**

**/\* 定义LCM操作的命令字 \*/**

**// T6963C 命令定义**

**#define LCM\_CUR\_POS 0x21 // 光标位置设置**

**#define LCM\_CGR\_POS 0x22 // CGRAM 偏置地址设置**

**#define LCM\_ADD\_POS 0x24 // 地址指针位置**

**#define LCM\_TXT\_STP 0x40 // 文本区首址**

**#define LCM\_TXT\_WID 0x41 // 文本区宽度**

**#define LCM\_GRH\_STP 0x42 // 图形区首址**

**#define LCM\_GRH\_WID 0x43 // 图形区宽度**

**#define LCM\_MOD\_OR 0x80 // 显示方式逻辑或**

**#define LCM\_MOD\_XOR 0x81 // 显示方式逻辑异或**

**#define LCM\_MOD\_AND 0x82 // 显示方式逻辑与**

**#define LCM\_MOD\_TCH 0x83 // 显示方式文本特征**

**#define LCM\_DIS\_SW 0x90 // 显示开关D0=1/0:光标闪烁启用/禁用**

**// D1=1/0:光标显示启用/禁用**

**// D2=1/0:文本显示启用/禁用**

**// D3=1/0:图形显示启用/禁用**

**#define LCM\_CUR\_SHP 0xA0 // 光标形状选择0xA0-0xA7表示光标占的行数**

**#define LCM\_AUT\_WR 0xB0 // 自动写设置**

**#define LCM\_AUT\_RD 0xB1 // 自动读设置**

**#define LCM\_AUT\_OVR 0xB2 // 自动读/写结束**

**#define LCM\_INC\_WR 0xC0 // 数据一次写地址加1**

**#define LCM\_INC\_RD 0xC1 // 数据一次读地址加1**

**#define LCM\_DEC\_WR 0xC2 // 数据一次写地址减1**

**#define LCM\_DEC\_RD 0xC3 // 数据一次读地址减1**

**#define LCM\_NOC\_WR 0xC4 // 数据一次写地址不变**

**#define LCM\_NOC\_RD 0xC5 // 数据一次读地址不变**

**#define LCM\_SCN\_RD 0xE0 // 屏读**

**#define LCM\_SCN\_CP 0xE8 // 屏拷贝**

**#define LCM\_BIT\_OP 0xF0 // 位操作**

**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};**

**uint8 LCM\_READSTATE()**

**{**

**uint8 dat;**

**IO0DIR = 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;**

**}**

**uint8 LCM\_STA01(void)**

**{**

**uint8 i;**

**for(i=10;i>0;i--)**

**{**

**if(( LCM\_READSTATE() & 0x03) == 0x03) // 读取状态**

**{**

**break;**

**}**

**}**

**return(i); // 若返回零说明错误**

**}**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\* 函数名称: LCM\_STA3**

**\*\* 功能描述: 状态位STA3**

**\*\* 输　入: 无**

**\*\* 输　出: 无**

**\*\* 全局变量: 无**

**\*\* 调用模块: LCM\_READSTATE**

**\*\* Modified by:**

**\*\* Modified date:**

**\*\*------------------------------------------------------------------------------------------------------**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**uint8 LCM\_STA3(void)**

**{**

**uint8 i;**

**for(i=10;i>0;i--)**

**{**

**if(( LCM\_READSTATE() & 0x08) == 0x08) // 读取状态**

**{**

**break;**

**}**

**}**

**return(i); // 若返回零说明错误**

**}**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\* 函数名称: LCM\_WrCommand**

**\*\* 功能描述: 写命令子程序**

**\*\* 输　入: command 要写入LCM的命令字**

**\*\* 输　出: 无**

**\*\* 全局变量: 无**

**\*\* 调用模块: 无**

**\*\* 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();**

**}**

**void LCM\_WrData(uint8 wrdata)**

**{**

**LCM\_UNREAD();**

**LCM\_DATA();**

**LCM\_WRITE();**

**LCM\_ENABLE();**

**OutData(wrdata);**

**//LCM\_UNWRITE();**

**//LCM\_READ();**

**LCM\_DISABLE();**

**}**

**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;**

**}**

**uint8 LCM\_ReadByte(uint8 x, uint8 y)**

**{**

**uint8 dat=0xff;**

**uint8 x1;**

**uint32 iPos;**

**x1 = x >> 3; // 取Y方向分页地址**

**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;**

**}**

**void LCM\_DispIni(void)**

**{**

**uint32 i;**

**// 设置引脚连接模块**

**#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**

**// 设置I/O为输出方式**

**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);**

**}**

**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);**

**}**

**void GUI\_Initialize(void)**

**{**

**LCM\_DispIni(); // 初始化LCM模块工作模式，纯图形模式**

**GUI\_FillSCR(0x00); // 初始化缓冲区为0x00，并输出屏幕(清屏)**

**}**

**uint8 GUI\_Point(uint8 x, uint8 y, TCOLOR color)**

**{**

**uint8 x1;**

**uint32 iPos;**

**x1 = x >> 3;**

**iPos = (uint32)y \* 0x1e + x1;**

**LCM\_WrParameter(LCM\_ADD\_POS,iPos&0xff,iPos/256,2);//分别取出低地址，高地址;写入LCD**

**x1 = turnf[ x & 0x07 ];//计算具体的行**

**//uint8 const turnf[8] = {7,6,5,4,3,2,1,0};**

**color = color <<3;**

**x1 = LCM\_BIT\_OP|x1|color; // 字节内位置计算,LCM\_BIT\_OP为位操作指令**

**/\*位操作：**

**1 1 1 1 N3 N2 N2 N0**

**无参数**

**该指令可将显示缓冲区某单元的某一位清零或置1，该单元地址由当前地址指针提供。**

**N3＝1置1，N3＝0 清零。N2－N0：操作位对应单元的D0－D7位。\*/**

**LCM\_WrParameter(x1,0,0,0);;**

**return 1;**

**}**

**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;**

**}**

**void GUI\_HLine(uint8 x0, uint8 y0, uint8 x1, TCOLOR color)**

**{ uint8 bak;**

**if(x0>x1) // 对x0、x1大小进行排列，以便画图**

**{ bak = x1;**

**x1 = x0;**

**x0 = bak;**

**}**

**do**

**{ GUI\_Point(x0, y0, color); // 逐点显示，描出垂直线**

**x0++;**

**}while(x1>=x0);**

**}**

**void GUI\_RLine(uint8 x0, uint8 y0, uint8 y1, TCOLOR color)**

**{ uint8 bak;**

**if(y0>y1) // 对x0、x1大小进行排列，以便画图**

**{ bak = y1;**

**y1 = y0;**

**y0 = bak;**

**}**

**do**

**{ GUI\_Point(x0, y0, color); // 逐点显示，描出垂直线**

**y0++;**

**}while(y1>=y0);**

**}**

**6.3 target.c**

**#define IN\_TARGET**

**#include "config.h"**

**void \_\_irq IRQ\_Exception(void)**

**{**

**while(1); // 这一句替换为自己的代码**

**}**

**void FIQ\_Exception(void)**

**{**

**while(1); // 这一句替换为自己的代码**

**}**

**void TargetInit(void)**

**{**

**/\* 添加自己的代码 \*/**

**}**

**void TargetResetInit(void)**

**{**

**/\* 设置系统各部分时钟 \*/**

**/\* PLLCON = 1;**

**#if ((Fcclk / 4) / Fpclk) == 1**

**VPBDIV = 0;**

**#endif**

**#if ((Fcclk / 4) / Fpclk) == 2**

**VPBDIV = 2;**

**#endif**

**#if ((Fcclk / 4) / Fpclk) == 4**

**VPBDIV = 1;**

**#endif**

**#if (Fcco / Fcclk) == 2**

**PLLCFG = ((Fcclk / Fosc) - 1) | (0 << 5);**

**#endif**

**#if (Fcco / Fcclk) == 4**

**PLLCFG = ((Fcclk / Fosc) - 1) | (1 << 5);**

**#endif**

**#if (Fcco / Fcclk) == 8**

**PLLCFG = ((Fcclk / Fosc) - 1) | (2 << 5);**

**#endif**

**#if (Fcco / Fcclk) == 16**

**PLLCFG = ((Fcclk / Fosc) - 1) | (3 << 5);**

**#endif**

**PLLFEED = 0xaa;**

**PLLFEED = 0x55;**

**while((PLLSTAT & (1 << 10)) == 0);**

**PLLCON = 3;**

**PLLFEED = 0xaa;**

**PLLFEED = 0x55;**

**\*/**

**/\* 设置存储器加速模块 \*/**

**MAMCR = 2; //MAMCR 8u2 00－MAM 功能被禁止、01－MAM功能部分使能、10－MAM功能完全使能**

**/\* MAMTIM 8u3**

**000＝0－保留**

**001＝1－一段时间内只有 1 个处理器时钟（cclk）用于 MAM 取指。**

**010＝2－一段时间内只有2个处理器时钟（cclk）用于MAM取指。**

**011＝3－一段时间内只有 3 个处理器时钟（cclk）用于 MAM 取指。**

**100＝4－一段时间内只有4个处理器时钟（cclk）用于 MAM 取指。**

**101＝5－一段时间内只有5 个处理器时钟（cclk）用于 MAM 取指。**

**110＝6－一段时间内只有 6 个处理器时钟（cclk）用于 MAM 取指。**

**111＝7－一段时间内只有7个处理器时钟（cclk）用于 MAM 取指。**

**\*/**

**#if Fcclk < 20000000**

**MAMTIM = 1;**

**#else**

**#if Fcclk < 40000000**

**MAMTIM = 2;**

**#else**

**MAMTIM = 3;**

**#endif**

**#endif**

**/\* 初始化VIC \*/**

**VICIntEnClr = 0xffffffff;//清所有中断使能，即关中断**

**VICVectAddr = 0;//默认向量地址寄存器**

**VICIntSelect = 0;//1：对应的中断请求分配为FIQ。0：对应的中断请求分配为IRQ。**

**/\* 添加自己的代码 \*/**

**}**

**#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;**

**}**

**6.4 Startup.s**

**;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**;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**

**;\*\*\***

**6.5 config.h**

**#ifndef \_\_CONFIG\_H**

**#define \_\_CONFIG\_H**

**//这一段无需改动**

**//This segment should not be modified**

**#ifndef TRUE**

**#define TRUE 1**

**#endif**

**#ifndef FALSE**

**#define FALSE 0**

**#endif**

**typedef unsigned char uint8; /\* defined for unsigned 8-bits integer variable 无符号8位整型变量 \*/**

**typedef signed char int8; /\* defined for signed 8-bits integer variable 有符号8位整型变量 \*/**

**typedef unsigned short uint16; /\* defined for unsigned 16-bits integer variable 无符号16位整型变量 \*/**

**typedef signed short int16; /\* defined for signed 16-bits integer variable 有符号16位整型变量 \*/**

**typedef unsigned int uint32; /\* defined for unsigned 32-bits integer variable 无符号32位整型变量 \*/**

**typedef signed int int32; /\* defined for signed 32-bits integer variable 有符号32位整型变量 \*/**

**typedef float fp32; /\* single precision floating point variable (32bits) 单精度浮点数（32位长度） \*/**

**typedef double fp64; /\* double precision floating point variable (64bits) 双精度浮点数（64位长度） \*/**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**/\* uC/OS-II specital code \*/**

**/\* uC/OS-II的特殊代码 \*/**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**#define USER\_USING\_MODE 0x10 /\* User mode ,ARM 32BITS CODE 用户模式,ARM代码 \*/**

**//**

**/\* Chosen one from 0x10,0x30,0x1f,0x3f.只能是0x10,0x30,0x1f,0x3f之一 \*/**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**/\* ARM的特殊代码 \*/**

**/\* ARM specital code \*/**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**//这一段无需改动**

**//This segment should not be modify**

**#include "LPC2124.h"**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**/\* 应用程序配置 \*/**

**/\*Application Program Configurations\*/**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**//以下根据需要改动**

**//This segment could be modified as needed.**

**#include <stdio.h>**

**#include <ctype.h>**

**#include <stdlib.h>**

**#include <setjmp.h>**

**#include <rt\_misc.h>**

**#include <math.h>**

**/\***

**#include "LCMDRV.h"**

**#include "LOADBIT.H"**

**#include "GUI\_StockC.h"**

**#include "GUI\_CONFIG.H"**

**#include "keyboard.h"**

**\*/**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**/\* 本例子的配置 \*/**

**/\*Configuration of the example \*/**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**/\* System configuration .Fosc、Fcclk、Fcco、Fpclk must be defined \*/**

**/\* 系统设置, Fosc、Fcclk、Fcco、Fpclk必须定义\*/**

**#define Fosc 11059200 //Crystal frequence,10MHz~25MHz，should be the same as actual status.**

**//应当与实际一至晶振频率,10MHz~25MHz，应当与实际一至**

**#define Fcclk (Fosc \* 4) //System frequence,should be (1~32)multiples of Fosc,and should be equal or less than 60MHz.**

**//系统频率，必须为Fosc的整数倍(1~32)，且<=60MHZ**

**#define Fcco (Fcclk \* 4) //CCO frequence,should be 2、4、8、16 multiples of Fcclk, ranged from 156MHz to 320MHz.**

**//CCO频率，必须为Fcclk的2、4、8、16倍，范围为156MHz~320MHz**

**#define Fpclk (Fcclk / 4) \* 1 //VPB clock frequence , must be 1、2、4 multiples of (Fcclk / 4).**

**//VPB时钟频率，只能为(Fcclk / 4)的1、2、4倍**

**#include "target.h" //This line may not be deleted 这一句不能删除**

**#endif**

**6.6 lcddrv.h**

**#define TCOLOR uint8**

**#define LCD\_DISP\_COLOR 1 //marco定义LCD的前景色和背景色**

**#define LCD\_BACK\_COLOR 0**

**#define GUI\_LCM\_XMAX 256 //240 /\* 定义液晶x轴的像素数 \*/**

**#define GUI\_LCM\_YMAX 64 //128**

**extern void GUI\_Initialize(void);**

**extern void GUI\_FillSCR(TCOLOR dat);**

**//extern void GUI\_ClearSCR(void);**

**#define GUI\_ClearSCR() GUI\_FillSCR(0x00)**

**extern uint8 GUI\_Point(uint8 x, uint8 y, TCOLOR color);**

**extern uint8 GUI\_ReadPoint(uint8 x, uint8 y, TCOLOR \*ret);**

**extern void GUI\_HLine(uint8 x0, uint8 y0, uint8 x1, TCOLOR color);**

**extern void GUI\_RLine(uint8 x0, uint8 y0, uint8 y1, TCOLOR color);**

**//extern int GUI\_CmpColor(TCOLOR color1, TCOLOR color2);**

**#define GUI\_CmpColor(color1, color2) ( (color1&0x01) == (color2&0x01) )**

**//extern void GUI\_CopyColor(TCOLOR \*color1, TCOLOR color2);**

**#define GUI\_CopyColor(color1, color2) \*color1 = color2&0x01**

**6.7 LPC2124.h**

**#ifndef \_\_LPC21xx\_H**

**#define \_\_LPC21xx\_H**

**/\* Vectored Interrupt Controller (VIC) \*/**

**#define VICIRQStatus (\*((volatile unsigned long \*) 0xFFFFF000))**

**#define VICFIQStatus (\*((volatile unsigned long \*) 0xFFFFF004))**

**#define VICRawIntr (\*((volatile unsigned long \*) 0xFFFFF008))**

**#define VICIntSelect (\*((volatile unsigned long \*) 0xFFFFF00C))**

**#define VICIntEnable (\*((volatile unsigned long \*) 0xFFFFF010))**

**#define VICIntEnClr (\*((volatile unsigned long \*) 0xFFFFF014))**

**#define VICSoftInt (\*((volatile unsigned long \*) 0xFFFFF018))**

**#define VICSoftIntClr (\*((volatile unsigned long \*) 0xFFFFF01C))**

**#define VICProtection (\*((volatile unsigned long \*) 0xFFFFF020))**

**#define VICVectAddr (\*((volatile unsigned long \*) 0xFFFFF030))**

**#define VICDefVectAddr (\*((volatile unsigned long \*) 0xFFFFF034))**

**#define VICVectAddr0 (\*((volatile unsigned long \*) 0xFFFFF100))**

**#define VICVectAddr1 (\*((volatile unsigned long \*) 0xFFFFF104))**

**#define VICVectAddr2 (\*((volatile unsigned long \*) 0xFFFFF108))**

**#define VICVectAddr3 (\*((volatile unsigned long \*) 0xFFFFF10C))**

**#define VICVectAddr4 (\*((volatile unsigned long \*) 0xFFFFF110))**

**#define VICVectAddr5 (\*((volatile unsigned long \*) 0xFFFFF114))**

**#define VICVectAddr6 (\*((volatile unsigned long \*) 0xFFFFF118))**

**#define VICVectAddr7 (\*((volatile unsigned long \*) 0xFFFFF11C))**

**#define VICVectAddr8 (\*((volatile unsigned long \*) 0xFFFFF120))**

**#define VICVectAddr9 (\*((volatile unsigned long \*) 0xFFFFF124))**

**#define VICVectAddr10 (\*((volatile unsigned long \*) 0xFFFFF128))**

**#define VICVectAddr11 (\*((volatile unsigned long \*) 0xFFFFF12C))**

**#define VICVectAddr12 (\*((volatile unsigned long \*) 0xFFFFF130))**

**#define VICVectAddr13 (\*((volatile unsigned long \*) 0xFFFFF134))**

**#define VICVectAddr14 (\*((volatile unsigned long \*) 0xFFFFF138))**

**#define VICVectAddr15 (\*((volatile unsigned long \*) 0xFFFFF13C))**

**#define VICVectCntl0 (\*((volatile unsigned long \*) 0xFFFFF200))**

**#define VICVectCntl1 (\*((volatile unsigned long \*) 0xFFFFF204))**

**#define VICVectCntl2 (\*((volatile unsigned long \*) 0xFFFFF208))**

**#define VICVectCntl3 (\*((volatile unsigned long \*) 0xFFFFF20C))**

**#define VICVectCntl4 (\*((volatile unsigned long \*) 0xFFFFF210))**

**#define VICVectCntl5 (\*((volatile unsigned long \*) 0xFFFFF214))**

**#define VICVectCntl6 (\*((volatile unsigned long \*) 0xFFFFF218))**

**#define VICVectCntl7 (\*((volatile unsigned long \*) 0xFFFFF21C))**

**#define VICVectCntl8 (\*((volatile unsigned long \*) 0xFFFFF220))**

**#define VICVectCntl9 (\*((volatile unsigned long \*) 0xFFFFF224))**

**#define VICVectCntl10 (\*((volatile unsigned long \*) 0xFFFFF228))**

**#define VICVectCntl11 (\*((volatile unsigned long \*) 0xFFFFF22C))**

**#define VICVectCntl12 (\*((volatile unsigned long \*) 0xFFFFF230))**

**#define VICVectCntl13 (\*((volatile unsigned long \*) 0xFFFFF234))**

**#define VICVectCntl14 (\*((volatile unsigned long \*) 0xFFFFF238))**

**#define VICVectCntl15 (\*((volatile unsigned long \*) 0xFFFFF23C))**

**/\* Pin Connect Block \*/**

**#define PINSEL0 (\*((volatile unsigned long \*) 0xE002C000))**

**#define PINSEL1 (\*((volatile unsigned long \*) 0xE002C004))**

**#define PINSEL2 (\*((volatile unsigned long \*) 0xE002C014))**

**/\* General Purpose Input/Output (GPIO) \*/**

**#define IOPIN0 (\*((volatile unsigned long \*) 0xE0028000))**

**#define IOSET0 (\*((volatile unsigned long \*) 0xE0028004))**

**#define IODIR0 (\*((volatile unsigned long \*) 0xE0028008))**

**#define IOCLR0 (\*((volatile unsigned long \*) 0xE002800C))**

**#define IOPIN1 (\*((volatile unsigned long \*) 0xE0028010))**

**#define IOSET1 (\*((volatile unsigned long \*) 0xE0028014))**

**#define IODIR1 (\*((volatile unsigned long \*) 0xE0028018))**

**#define IOCLR1 (\*((volatile unsigned long \*) 0xE002801C))**

**#define IO0PIN (\*((volatile unsigned long \*) 0xE0028000))**

**#define IO0SET (\*((volatile unsigned long \*) 0xE0028004))**

**#define IO0DIR (\*((volatile unsigned long \*) 0xE0028008))**

**#define IO0CLR (\*((volatile unsigned long \*) 0xE002800C))**

**#define IO1PIN (\*((volatile unsigned long \*) 0xE0028010))**

**#define IO1SET (\*((volatile unsigned long \*) 0xE0028014))**

**#define IO1DIR (\*((volatile unsigned long \*) 0xE0028018))**

**#define IO1CLR (\*((volatile unsigned long \*) 0xE002801C))**

**/\* Memory Accelerator Module (MAM) \*/**

**#define MAMCR (\*((volatile unsigned char \*) 0xE01FC000))**

**#define MAMTIM (\*((volatile unsigned char \*) 0xE01FC004))**

**#define MEMMAP (\*((volatile unsigned char \*) 0xE01FC040))**

**/\* Phase Locked Loop (PLL) \*/**

**#define PLLCON (\*((volatile unsigned char \*) 0xE01FC080))**

**#define PLLCFG (\*((volatile unsigned char \*) 0xE01FC084))**

**#define PLLSTAT (\*((volatile unsigned short\*) 0xE01FC088))**

**#define PLLFEED (\*((volatile unsigned char \*) 0xE01FC08C))**

**/\* VPB Divider \*/**

**#define VPBDIV (\*((volatile unsigned char \*) 0xE01FC100))**

**/\* Power Control \*/**

**#define PCON (\*((volatile unsigned char \*) 0xE01FC0C0))**

**#define PCONP (\*((volatile unsigned long \*) 0xE01FC0C4))**

**/\* External Interrupts \*/**

**#define EXTINT (\*((volatile unsigned char \*) 0xE01FC140))**

**#define EXTWAKE (\*((volatile unsigned char \*) 0xE01FC144))**

**#define EXTMODE (\*((volatile unsigned char \*) 0xE01FC148))**

**#define EXTPOLAR (\*((volatile unsigned char \*) 0xE01FC14C))**

**/\* Timer 0 \*/**

**#define T0IR (\*((volatile unsigned long \*) 0xE0004000))**

**#define T0TCR (\*((volatile unsigned long \*) 0xE0004004))**

**#define T0TC (\*((volatile unsigned long \*) 0xE0004008))**

**#define T0PR (\*((volatile unsigned long \*) 0xE000400C))**

**#define T0PC (\*((volatile unsigned long \*) 0xE0004010))**

**#define T0MCR (\*((volatile unsigned long \*) 0xE0004014))**

**#define T0MR0 (\*((volatile unsigned long \*) 0xE0004018))**

**#define T0MR1 (\*((volatile unsigned long \*) 0xE000401C))**

**#define T0MR2 (\*((volatile unsigned long \*) 0xE0004020))**

**#define T0MR3 (\*((volatile unsigned long \*) 0xE0004024))**

**#define T0CCR (\*((volatile unsigned long \*) 0xE0004028))**

**#define T0CR0 (\*((volatile unsigned long \*) 0xE000402C))**

**#define T0CR1 (\*((volatile unsigned long \*) 0xE0004030))**

**#define T0CR2 (\*((volatile unsigned long \*) 0xE0004034))**

**#define T0CR3 (\*((volatile unsigned long \*) 0xE0004038))**

**#define T0EMR (\*((volatile unsigned long \*) 0xE000403C))**

**/\* Timer 1 \*/**

**#define T1IR (\*((volatile unsigned long \*) 0xE0008000))**

**#define T1TCR (\*((volatile unsigned long \*) 0xE0008004))**

**#define T1TC (\*((volatile unsigned long \*) 0xE0008008))**

**#define T1PR (\*((volatile unsigned long \*) 0xE000800C))**

**#define T1PC (\*((volatile unsigned long \*) 0xE0008010))**

**#define T1MCR (\*((volatile unsigned long \*) 0xE0008014))**

**#define T1MR0 (\*((volatile unsigned long \*) 0xE0008018))**

**#define T1MR1 (\*((volatile unsigned long \*) 0xE000801C))**

**#define T1MR2 (\*((volatile unsigned long \*) 0xE0008020))**

**#define T1MR3 (\*((volatile unsigned long \*) 0xE0008024))**

**#define T1CCR (\*((volatile unsigned long \*) 0xE0008028))**

**#define T1CR0 (\*((volatile unsigned long \*) 0xE000802C))**

**#define T1CR1 (\*((volatile unsigned long \*) 0xE0008030))**

**#define T1CR2 (\*((volatile unsigned long \*) 0xE0008034))**

**#define T1CR3 (\*((volatile unsigned long \*) 0xE0008038))**

**#define T1EMR (\*((volatile unsigned long \*) 0xE000803C))**

**/\* Pulse Width Modulator (PWM) \*/**

**#define PWMIR (\*((volatile unsigned long \*) 0xE0014000))**

**#define PWMTCR (\*((volatile unsigned long \*) 0xE0014004))**

**#define PWMTC (\*((volatile unsigned long \*) 0xE0014008))**

**#define PWMPR (\*((volatile unsigned long \*) 0xE001400C))**

**#define PWMPC (\*((volatile unsigned long \*) 0xE0014010))**

**#define PWMMCR (\*((volatile unsigned long \*) 0xE0014014))**

**#define PWMMR0 (\*((volatile unsigned long \*) 0xE0014018))**

**#define PWMMR1 (\*((volatile unsigned long \*) 0xE001401C))**

**#define PWMMR2 (\*((volatile unsigned long \*) 0xE0014020))**

**#define PWMMR3 (\*((volatile unsigned long \*) 0xE0014024))**

**#define PWMMR4 (\*((volatile unsigned long \*) 0xE0014040))**

**#define PWMMR5 (\*((volatile unsigned long \*) 0xE0014044))**

**#define PWMMR6 (\*((volatile unsigned long \*) 0xE0014048))**

**#define PWMPCR (\*((volatile unsigned long \*) 0xE001404C))**

**#define PWMLER (\*((volatile unsigned long \*) 0xE0014050))**

**/\* Universal Asynchronous Receiver Transmitter 0 (UART0) \*/**

**#define U0RBR (\*((volatile unsigned char \*) 0xE000C000))**

**#define U0THR (\*((volatile unsigned char \*) 0xE000C000))**

**#define U0IER (\*((volatile unsigned char \*) 0xE000C004))**

**#define U0IIR (\*((volatile unsigned char \*) 0xE000C008))**

**#define U0FCR (\*((volatile unsigned char \*) 0xE000C008))**

**#define U0LCR (\*((volatile unsigned char \*) 0xE000C00C))**

**#define U0LSR (\*((volatile unsigned char \*) 0xE000C014))**

**#define U0SCR (\*((volatile unsigned char \*) 0xE000C01C))**

**#define U0FDR (\*((volatile unsigned char \*) 0xE000C028))**

**#define U0DLL (\*((volatile unsigned char \*) 0xE000C000))**

**#define U0DLM (\*((volatile unsigned char \*) 0xE000C004))**

**/\* Universal Asynchronous Receiver Transmitter 1 (UART1) \*/**

**#define U1RBR (\*((volatile unsigned char \*) 0xE0010000))**

**#define U1THR (\*((volatile unsigned char \*) 0xE0010000))**

**#define U1IER (\*((volatile unsigned char \*) 0xE0010004))**

**#define U1IIR (\*((volatile unsigned char \*) 0xE0010008))**

**#define U1FCR (\*((volatile unsigned char \*) 0xE0010008))**

**#define U1LCR (\*((volatile unsigned char \*) 0xE001000C))**

**#define U1MCR (\*((volatile unsigned char \*) 0xE0010010))**

**#define U1LSR (\*((volatile unsigned char \*) 0xE0010014))**

**#define U1MSR (\*((volatile unsigned char \*) 0xE0010018))**

**#define U1SCR (\*((volatile unsigned char \*) 0xE001001C))**

**#define U1FDR (\*((volatile unsigned char \*) 0xE0010028))**

**#define U1DLL (\*((volatile unsigned char \*) 0xE0010000))**

**#define U1DLM (\*((volatile unsigned char \*) 0xE0010004))**

**/\* I2C Interface \*/**

**#define I2CONSET (\*((volatile unsigned char \*) 0xE001C000))**

**#define I2STAT (\*((volatile unsigned char \*) 0xE001C004))**

**#define I2DAT (\*((volatile unsigned char \*) 0xE001C008))**

**#define I2ADR (\*((volatile unsigned char \*) 0xE001C00C))**

**#define I2SCLH (\*((volatile unsigned short\*) 0xE001C010))**

**#define I2SCLL (\*((volatile unsigned short\*) 0xE001C014))**

**#define I2CONCLR (\*((volatile unsigned char \*) 0xE001C018))**

**/\* SPI0 (Serial Peripheral Interface 0) \*/**

**#define S0SPCR (\*((volatile unsigned short\*) 0xE0020000))**

**#define S0SPSR (\*((volatile unsigned char \*) 0xE0020004))**

**#define S0SPDR (\*((volatile unsigned short\*) 0xE0020008))**

**#define S0SPCCR (\*((volatile unsigned char \*) 0xE002000C))**

**#define S0SPINT (\*((volatile unsigned char \*) 0xE002001C))**

**/\* SPI1 (Serial Peripheral Interface 1) \*/**

**#define S1SPCR (\*((volatile unsigned short\*) 0xE0030000))**

**#define S1SPSR (\*((volatile unsigned char \*) 0xE0030004))**

**#define S1SPDR (\*((volatile unsigned short\*) 0xE0030008))**

**#define S1SPCCR (\*((volatile unsigned char \*) 0xE003000C))**

**#define S1SPINT (\*((volatile unsigned char \*) 0xE003001C))**

**/\* Real Time Clock \*/**

**#define ILR (\*((volatile unsigned char \*) 0xE0024000))**

**#define CTC (\*((volatile unsigned short\*) 0xE0024004))**

**#define CCR (\*((volatile unsigned char \*) 0xE0024008))**

**#define CIIR (\*((volatile unsigned char \*) 0xE002400C))**

**#define AMR (\*((volatile unsigned char \*) 0xE0024010))**

**#define CTIME0 (\*((volatile unsigned long \*) 0xE0024014))**

**#define CTIME1 (\*((volatile unsigned long \*) 0xE0024018))**

**#define CTIME2 (\*((volatile unsigned long \*) 0xE002401C))**

**#define SEC (\*((volatile unsigned char \*) 0xE0024020))**

**#define MIN (\*((volatile unsigned char \*) 0xE0024024))**

**#define HOUR (\*((volatile unsigned char \*) 0xE0024028))**

**#define DOM (\*((volatile unsigned char \*) 0xE002402C))**

**#define DOW (\*((volatile unsigned char \*) 0xE0024030))**

**#define DOY (\*((volatile unsigned short\*) 0xE0024034))**

**#define MONTH (\*((volatile unsigned char \*) 0xE0024038))**

**#define YEAR (\*((volatile unsigned short\*) 0xE002403C))**

**#define ALSEC (\*((volatile unsigned char \*) 0xE0024060))**

**#define ALMIN (\*((volatile unsigned char \*) 0xE0024064))**

**#define ALHOUR (\*((volatile unsigned char \*) 0xE0024068))**

**#define ALDOM (\*((volatile unsigned char \*) 0xE002406C))**

**#define ALDOW (\*((volatile unsigned char \*) 0xE0024070))**

**#define ALDOY (\*((volatile unsigned short\*) 0xE0024074))**

**#define ALMON (\*((volatile unsigned char \*) 0xE0024078))**

**#define ALYEAR (\*((volatile unsigned short\*) 0xE002407C))**

**#define PREINT (\*((volatile unsigned short\*) 0xE0024080))**

**#define PREFRAC (\*((volatile unsigned short\*) 0xE0024084))**

**/\* A/D Converter \*/**

**#define ADCR (\*((volatile unsigned long \*) 0xE0034000))**

**#define ADDR (\*((volatile unsigned long \*) 0xE0034004))**

**/\* CAN Acceptance Filter RAM \*/**

**#define AFRAM (\*((volatile unsigned long \*) 0xE0038000))**

**/\* CAN Acceptance Filter \*/**

**#define AFMR (\*((volatile unsigned long \*) 0xE003C000))**

**#define SFF\_sa (\*((volatile unsigned long \*) 0xE003C004))**

**#define SFF\_GRP\_sa (\*((volatile unsigned long \*) 0xE003C008))**

**#define EFF\_sa (\*((volatile unsigned long \*) 0xE003C00C))**

**#define EFF\_GRP\_sa (\*((volatile unsigned long \*) 0xE003C010))**

**#define ENDofTable (\*((volatile unsigned long \*) 0xE003C014))**

**#define LUTerrAd (\*((volatile unsigned long \*) 0xE003C018))**

**#define LUTerr (\*((volatile unsigned long \*) 0xE003C01C))**

**/\* CAN Central Registers \*/**

**#define CANTxSR (\*((volatile unsigned long \*) 0xE0040000))**

**#define CANRxSR (\*((volatile unsigned long \*) 0xE0040004))**

**#define CANMSR (\*((volatile unsigned long \*) 0xE0040008))**

**/\* CAN Controller 1 (CAN1) \*/**

**#define C1MOD (\*((volatile unsigned long \*) 0xE0044000))**

**#define C1CMR (\*((volatile unsigned long \*) 0xE0044004))**

**#define C1GSR (\*((volatile unsigned long \*) 0xE0044008))**

**#define C1ICR (\*((volatile unsigned long \*) 0xE004400C))**

**#define C1IER (\*((volatile unsigned long \*) 0xE0044010))**

**#define C1BTR (\*((volatile unsigned long \*) 0xE0044014))**

**#define C1EWL (\*((volatile unsigned long \*) 0xE0044018))**

**#define C1SR (\*((volatile unsigned long \*) 0xE004401C))**

**#define C1RFS (\*((volatile unsigned long \*) 0xE0044020))**

**#define C1RID (\*((volatile unsigned long \*) 0xE0044024))**

**#define C1RDA (\*((volatile unsigned long \*) 0xE0044028))**

**#define C1RDB (\*((volatile unsigned long \*) 0xE004402C))**

**#define C1TFI1 (\*((volatile unsigned long \*) 0xE0044030))**

**#define C1TID1 (\*((volatile unsigned long \*) 0xE0044034))**

**#define C1TDA1 (\*((volatile unsigned long \*) 0xE0044038))**

**#define C1TDB1 (\*((volatile unsigned long \*) 0xE004403C))**

**#define C1TFI2 (\*((volatile unsigned long \*) 0xE0044040))**

**#define C1TID2 (\*((volatile unsigned long \*) 0xE0044044))**

**#define C1TDA2 (\*((volatile unsigned long \*) 0xE0044048))**

**#define C1TDB2 (\*((volatile unsigned long \*) 0xE004404C))**

**#define C1TFI3 (\*((volatile unsigned long \*) 0xE0044050))**

**#define C1TID3 (\*((volatile unsigned long \*) 0xE0044054))**

**#define C1TDA3 (\*((volatile unsigned long \*) 0xE0044058))**

**#define C1TDB3 (\*((volatile unsigned long \*) 0xE004405C))**

**/\* CAN Controller 2 (CAN2) \*/**

**#define C2MOD (\*((volatile unsigned long \*) 0xE0048000))**

**#define C2CMR (\*((volatile unsigned long \*) 0xE0048004))**

**#define C2GSR (\*((volatile unsigned long \*) 0xE0048008))**

**#define C2ICR (\*((volatile unsigned long \*) 0xE004800C))**

**#define C2IER (\*((volatile unsigned long \*) 0xE0048010))**

**#define C2BTR (\*((volatile unsigned long \*) 0xE0048014))**

**#define C2EWL (\*((volatile unsigned long \*) 0xE0048018))**

**#define C2SR (\*((volatile unsigned long \*) 0xE004801C))**

**#define C2RFS (\*((volatile unsigned long \*) 0xE0048020))**

**#define C2RID (\*((volatile unsigned long \*) 0xE0048024))**

**#define C2RDA (\*((volatile unsigned long \*) 0xE0048028))**

**#define C2RDB (\*((volatile unsigned long \*) 0xE004802C))**

**#define C2TFI1 (\*((volatile unsigned long \*) 0xE0048030))**

**#define C2TID1 (\*((volatile unsigned long \*) 0xE0048034))**

**#define C2TDA1 (\*((volatile unsigned long \*) 0xE0048038))**

**#define C2TDB1 (\*((volatile unsigned long \*) 0xE004803C))**

**#define C2TFI2 (\*((volatile unsigned long \*) 0xE0048040))**

**#define C2TID2 (\*((volatile unsigned long \*) 0xE0048044))**

**#define C2TDA2 (\*((volatile unsigned long \*) 0xE0048048))**

**#define C2TDB2 (\*((volatile unsigned long \*) 0xE004804C))**

**#define C2TFI3 (\*((volatile unsigned long \*) 0xE0048050))**

**#define C2TID3 (\*((volatile unsigned long \*) 0xE0048054))**

**#define C2TDA3 (\*((volatile unsigned long \*) 0xE0048058))**

**#define C2TDB3 (\*((volatile unsigned long \*) 0xE004805C))**

**/\* CAN Controller 3 (CAN3) \*/**

**#define C3MOD (\*((volatile unsigned long \*) 0xE004C000))**

**#define C3CMR (\*((volatile unsigned long \*) 0xE004C004))**

**#define C3GSR (\*((volatile unsigned long \*) 0xE004C008))**

**#define C3ICR (\*((volatile unsigned long \*) 0xE004C00C))**

**#define C3IER (\*((volatile unsigned long \*) 0xE004C010))**

**#define C3BTR (\*((volatile unsigned long \*) 0xE004C014))**

**#define C3EWL (\*((volatile unsigned long \*) 0xE004C018))**

**#define C3SR (\*((volatile unsigned long \*) 0xE004C01C))**

**#define C3RFS (\*((volatile unsigned long \*) 0xE004C020))**

**#define C3RID (\*((volatile unsigned long \*) 0xE004C024))**

**#define C3RDA (\*((volatile unsigned long \*) 0xE004C028))**

**#define C3RDB (\*((volatile unsigned long \*) 0xE004C02C))**

**#define C3TFI1 (\*((volatile unsigned long \*) 0xE004C030))**

**#define C3TID1 (\*((volatile unsigned long \*) 0xE004C034))**

**#define C3TDA1 (\*((volatile unsigned long \*) 0xE004C038))**

**#define C3TDB1 (\*((volatile unsigned long \*) 0xE004C03C))**

**#define C3TFI2 (\*((volatile unsigned long \*) 0xE004C040))**

**#define C3TID2 (\*((volatile unsigned long \*) 0xE004C044))**

**#define C3TDA2 (\*((volatile unsigned long \*) 0xE004C048))**

**#define C3TDB2 (\*((volatile unsigned long \*) 0xE004C04C))**

**#define C3TFI3 (\*((volatile unsigned long \*) 0xE004C050))**

**#define C3TID3 (\*((volatile unsigned long \*) 0xE004C054))**

**#define C3TDA3 (\*((volatile unsigned long \*) 0xE004C058))**

**#define C3TDB3 (\*((volatile unsigned long \*) 0xE004C05C))**

**/\* CAN Controller 4 (CAN4) \*/**

**#define C4MOD (\*((volatile unsigned long \*) 0xE0050000))**

**#define C4CMR (\*((volatile unsigned long \*) 0xE0050004))**

**#define C4GSR (\*((volatile unsigned long \*) 0xE0050008))**

**#define C4ICR (\*((volatile unsigned long \*) 0xE005000C))**

**#define C4IER (\*((volatile unsigned long \*) 0xE0050010))**

**#define C4BTR (\*((volatile unsigned long \*) 0xE0050014))**

**#define C4EWL (\*((volatile unsigned long \*) 0xE0050018))**

**#define C4SR (\*((volatile unsigned long \*) 0xE005001C))**

**#define C4RFS (\*((volatile unsigned long \*) 0xE0050020))**

**#define C4RID (\*((volatile unsigned long \*) 0xE0050024))**

**#define C4RDA (\*((volatile unsigned long \*) 0xE0050028))**

**#define C4RDB (\*((volatile unsigned long \*) 0xE005002C))**

**#define C4TFI1 (\*((volatile unsigned long \*) 0xE0050030))**

**#define C4TID1 (\*((volatile unsigned long \*) 0xE0050034))**

**#define C4TDA1 (\*((volatile unsigned long \*) 0xE0050038))**

**#define C4TDB1 (\*((volatile unsigned long \*) 0xE005003C))**

**#define C4TFI2 (\*((volatile unsigned long \*) 0xE0050040))**

**#define C4TID2 (\*((volatile unsigned long \*) 0xE0050044))**

**#define C4TDA2 (\*((volatile unsigned long \*) 0xE0050048))**

**#define C4TDB2 (\*((volatile unsigned long \*) 0xE005004C))**

**#define C4TFI3 (\*((volatile unsigned long \*) 0xE0050050))**

**#define C4TID3 (\*((volatile unsigned long \*) 0xE0050054))**

**#define C4TDA3 (\*((volatile unsigned long \*) 0xE0050058))**

**#define C4TDB3 (\*((volatile unsigned long \*) 0xE005005C))**

**/\* Watchdog \*/**

**#define WDMOD (\*((volatile unsigned char \*) 0xE0000000))**

**#define WDTC (\*((volatile unsigned long \*) 0xE0000004))**

**#define WDFEED (\*((volatile unsigned char \*) 0xE0000008))**

**#define WDTV (\*((volatile unsigned long \*) 0xE000000C))**

**#endif // \_\_LPC21xx\_H**

**6.8 target.h**

**#ifndef IN\_TARGET**

**extern void Reset(void);**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\* 函数名称: Reset**

**\*\* 功能描述: 目标板软复位**

**\*\* 输　入: 无**

**\*\***

**\*\* 输　出: 无**

**\*\***

**\*\* 全局变量: 无**

**\*\* 调用模块: 无**

**\*\***

**\*\* 作　者: 陈明计**

**\*\* 日　期: 2003年5月30日**

**\*\*-------------------------------------------------------------------------------------------------------**

**\*\* 修改人:**

**\*\* 日　期:**

**\*\*------------------------------------------------------------------------------------------------------**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**extern void TargetInit(void);**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\* 函数名称: TargetInit**

**\*\* 功能描述: 目标板初始化代码，在需要的地方调用，根据需要改变**

**\*\* 输　入: 无**

**\*\***

**\*\* 输　出: 无**

**\*\***

**\*\* 全局变量: 无**

**\*\* 调用模块: 无**

**\*\***

**\*\* 作　者: 陈明计**

**\*\* 日　期: 2003年5月30日**

**\*\*-------------------------------------------------------------------------------------------------------**

**\*\* 修改人:**

**\*\* 日　期:**

**\*\*------------------------------------------------------------------------------------------------------**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**#endif**

**6.9 LPC2124.sct**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; \*\*\* Scatter-Loading Description File for LPC2103 \*\*\*

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

LR\_ROM1 0x00000000

{ ; load region size\_region

ER\_ROM1 0x00000000 0x00020000

{ ; load address = execution address

Startup.o (vectors, +First)

\* (+RO)

}

RW\_IRAM1 0x40000000

{ ; RW data

Startup.o (MyStacks)

\* (+RW,+ZI)

}

}