

中国地质大学（武汉）自动化学院

嵌入式系统技术实习报告

课 程： 嵌入式系统技术实习

学 号： 20201000128

班 级： 231202-01

姓 名： 刘瑾瑾

指导老师： 刘玮、张莉君、彭健
陆承达、张盼

二〇二二年十二月十二日

目录

第一部分：裸机系统设计及实现——智能手机系统	1
1.1 总体设计	1
1.2 功能介绍	2
1.2.1 密码	2
1.2.2 功能选择界面	4
1.2.3 电子相册	5
1.2.4 计算器	6
1.2.5 音乐播放器	7
1.2.6 钢琴	9
1.2.7 电子书	10
1.2.8 手电筒	11
1.2.9 闹钟	12
1.3 实习心得	12
1.3.1 遇到问题及解决	12
1.3.2 收获感想	14
第二部分：Linux 系统体验实习	16
2.1 对 linux 系统、移植、驱动开发的理解	16
2.1.1 对 linux 系统的理解	16
2.1.2 对 linux 系统移植的理解	16
2.1.3 对 linux 系统驱动开发的理解	17
2.2 基于 linux 系统的应用开发与裸机开发的区别	18
2.2.1 应用开发环境的硬件设备不同	18
2.2.2 程序下载方式不同	18
2.2.3 固件的存储位置不同	18
2.2.4 驱动不同	18
2.2.5 启动方式不同	19
2.3 成果展示	20
2.4 实习心得	21
2.4.1 遇到问题及解决	21
2.4.2 收获感想	22
实习意见建议	23
附录	24

第一部分：裸机系统设计及实现——智能手机系统

1.1 总体设计

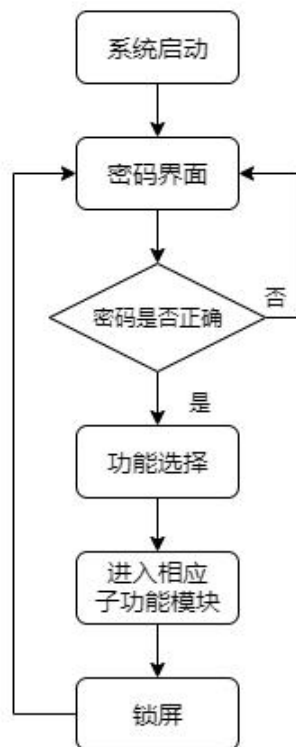
利用此次嵌入式实习的机会，我们计划设计一款功能比较丰富的多功能智能手机系统。尽可能使用学习到的 S3C2440 的所有相关功能和模块，如课堂上学习的中断和定时器功能、实习第一阶段的 LCD 显示屏、触摸屏、PWM 模块和字库的移植等。

我们的总体设计是首先输入密码，密码正确则进入功能选择界面，包括电子相册、计算器、音乐播放器、钢琴、电子书、手电筒和闹钟功能，也可直接退出到锁屏界面。

系统结构如下所示：



总体结构流程图如下所示：

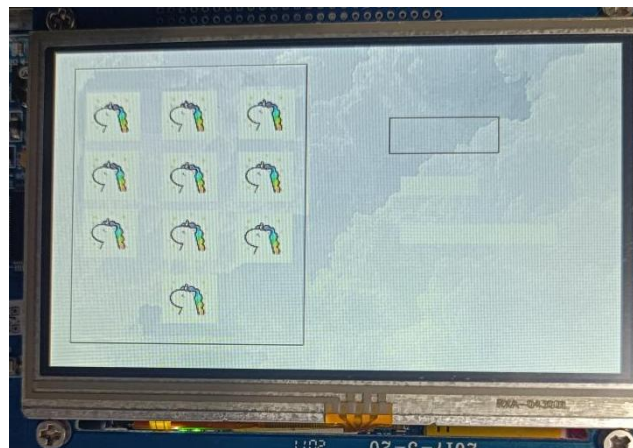


1.2 功能介绍

1.2.1 密码

(1) 简要介绍:

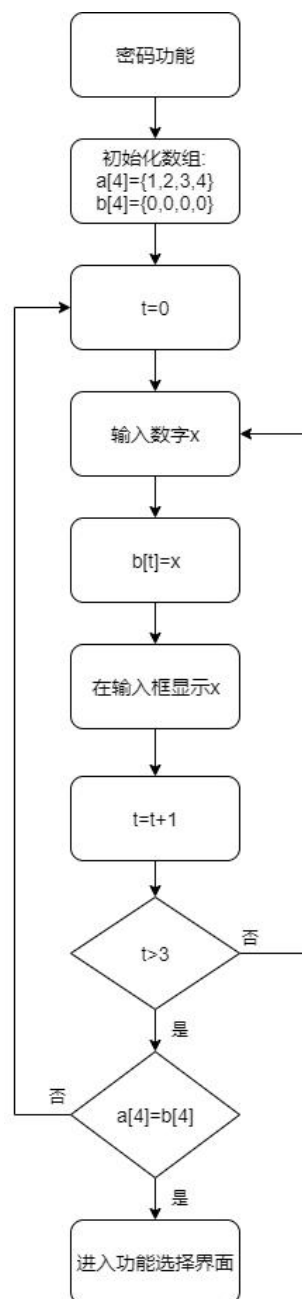
将程序下载到单片机。通过 NAND Flash 启动，首先进入密码锁屏界面，需输入密码。如果密码错误，会继续停留在密码界面，提示“密码错误，请重新输入密码”；如果密码正确，提示“密码正确，欢迎使用智能手机”，并进入功能选择界面。



(2) 设计思路:

设置两个大小为 4 的数组，数组一存储设定的正确密码，数组二存储用户输入的密码，用户输入一个数字，程序将该数字存储到数组中并将其显示在密码输入框。当四个数字全部输入完毕，在显示框显示“****”，将数组一和数组二进行比对，若完全相同，则显示“密码正确，欢迎使用智能手机”，结束密码程序的运行，进入主功能界面；若不同，则显示“密码错误，请重新输入密码”，将输入框清零，重新输入。

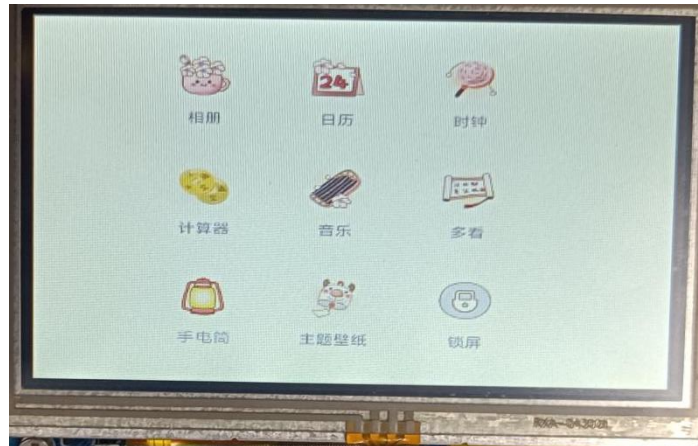
程序流程图如下所示:



1.2.2 功能选择界面

(1) 简要介绍:

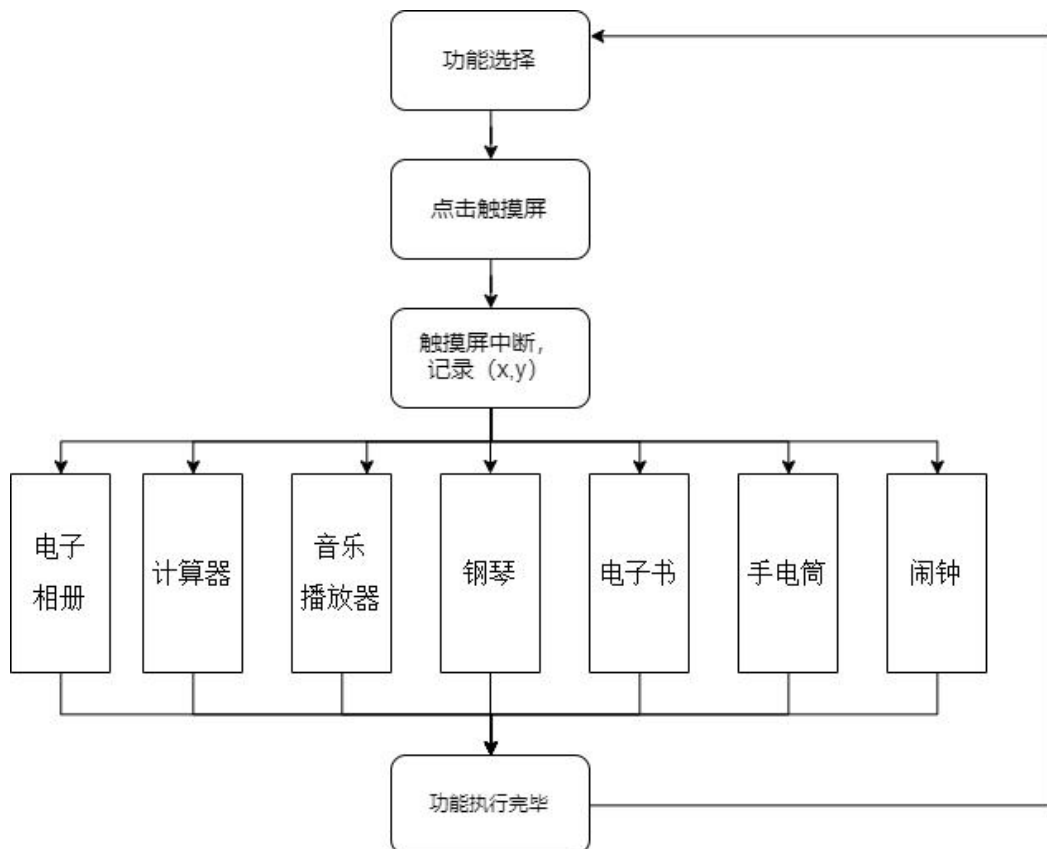
功能界面主要包括电子相册、计算器、音乐播放器、钢琴、电子书、手电筒和闹钟功能，点击相应功能，即可进入相应的功能界面，锁屏功能直接



回到密码界面。

(2) 设计思路:

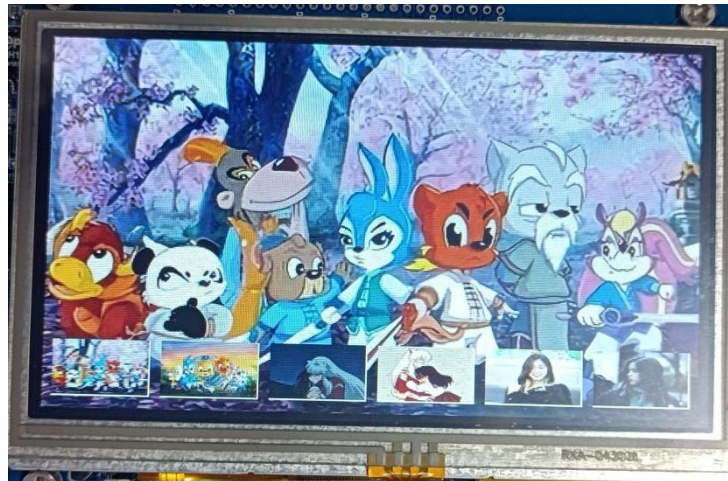
利用触摸屏中断，点击 LCD 显示屏，进入中断，坐标 (x, y) 发生改变，通过 x 和 y 的范围判断需执行的功能。程序流程图如下所示：



1.2.3 电子相册

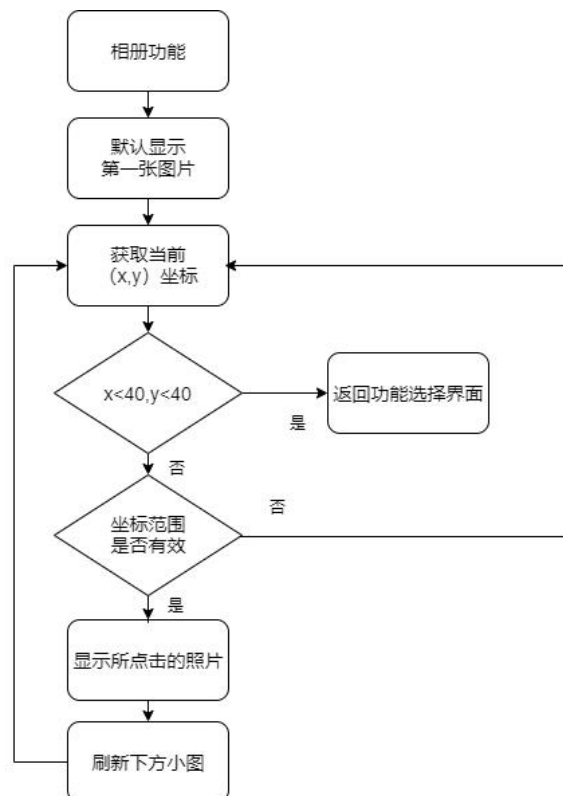
(1) 简要介绍:

进入电子相册功能后，LCD 显示屏会显示当前图片，并在下方罗列相册中的图片，点击相应图片，图片放大并显示。



(2) 设计思路:

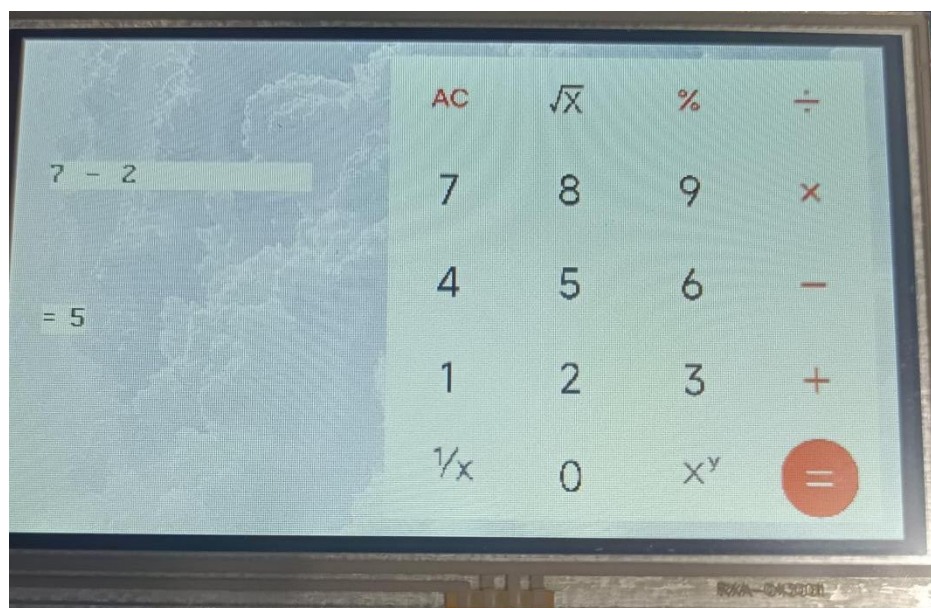
利用触摸屏中断，点击 LCD 显示屏，进入中断，坐标(x, y)发生改变，根据(x, y)的当前坐标判断是否需要更新图片，在显示的大图片更新后，在下方刷新显示小图片。程序流程图如下所示：



1.2.4 计算器

(1) 简要介绍:

进入计算器功能后，按相应按键输入对应的字符，并将字符显示在输入框内，可以计算相对简单的加减乘除运算以及开方、次方和求倒数运算，按等号后输出框会显示结算结果。按 AC 键可以清零重新计算，此时输入框和输出框会刷新为空。

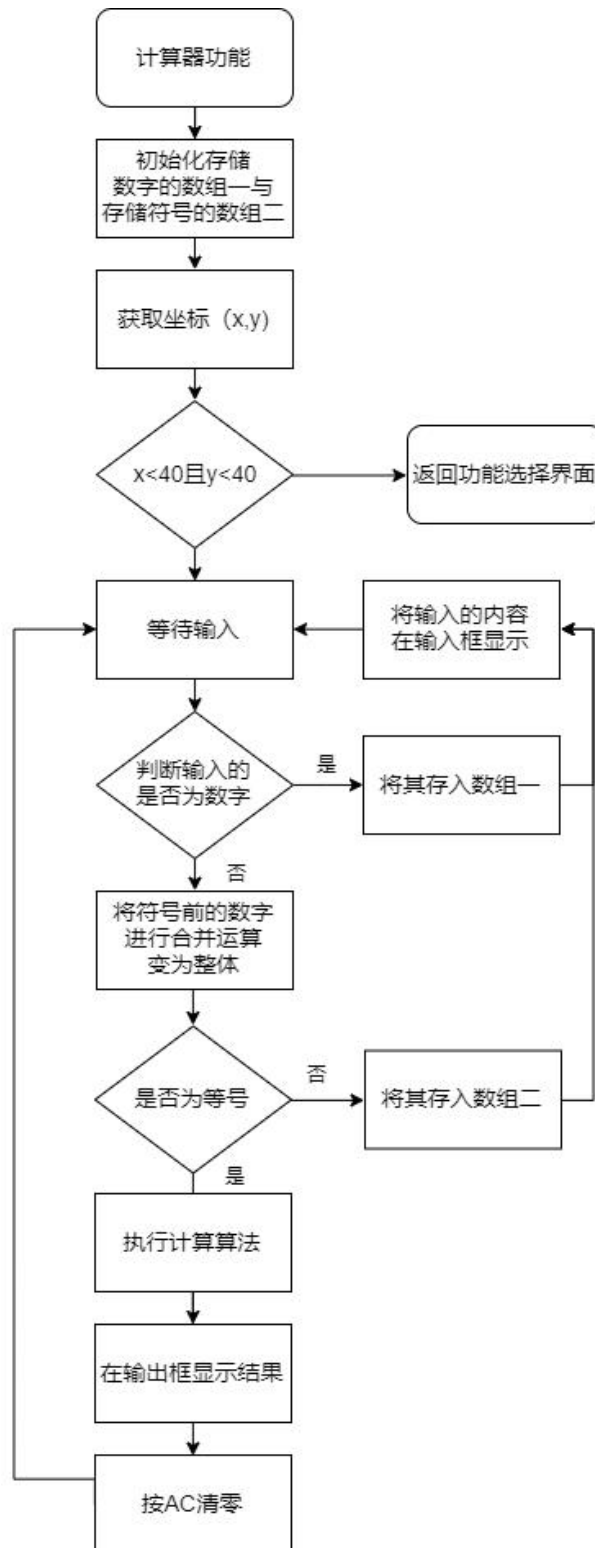


(2) 设计思路:

初始化两个数组，数组一存储输入的数字，数组二存储输入的符号。等待计算器输入：若输入的是数字，则将其存入数组一；若输入的是符号，需要将符号前的数字合并为一个整体，比如要输入的数字为 123，但是 1、2 和 3 是单独输入的，需要进行转换变为 123，并重新存入数组一，之后需判断是否为等号，若不是等号，将运算符存入数组二，若是等号，则执行计算算法。执行完毕后，结果会显示在输出框中，清零即可重新输入。按左上角即可返回功能选择界面。

计算算法的实现：需要判断优先级和符号的目数，比如：‘*’和‘/’比‘+’和‘-’的优先级更高，‘+’，‘-’是二目运算符，‘√’是单目运算符；另外，需要注意数组一和数组二的对应关系。例如：1+√2/3，需要区分出‘√’是单目运算符，判断优先级先计算‘√’，再计算‘/’，最后计算‘+’。

程序流程图如下所示：

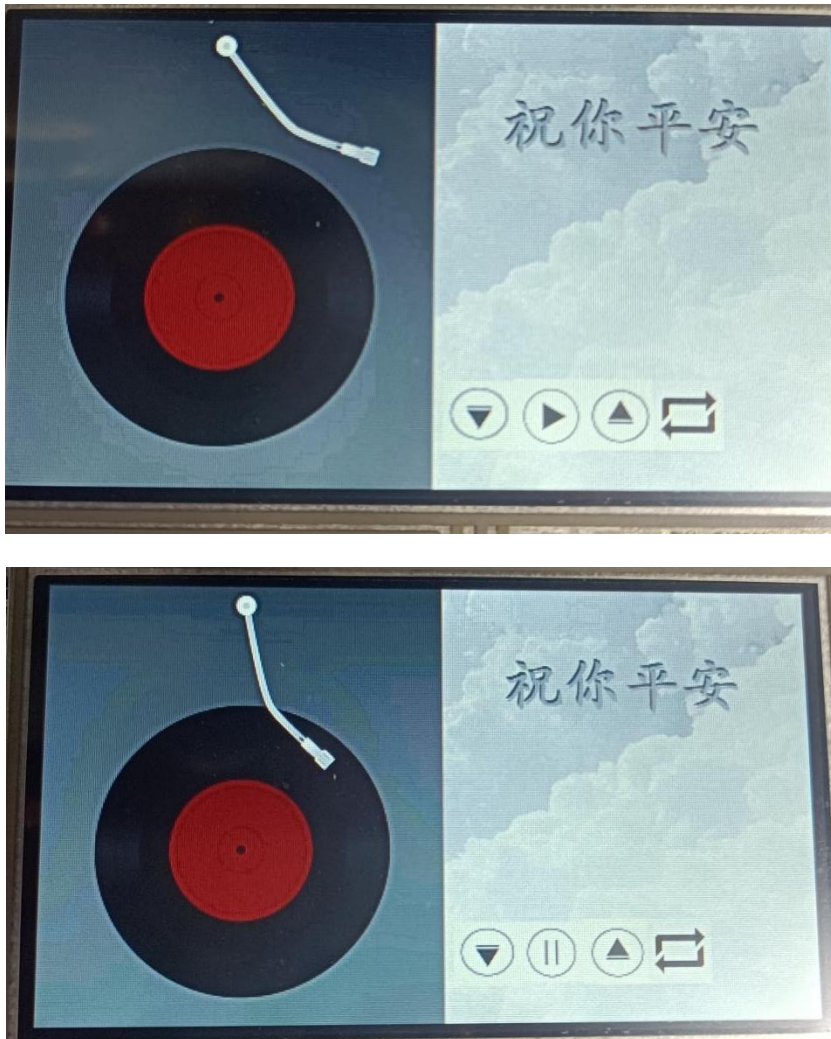


1.2.5 音乐播放器

(1) 简要介绍:

进入音乐播放器功能后，通过控制 PWM 的占空比和频率从而控制蜂鸣器的

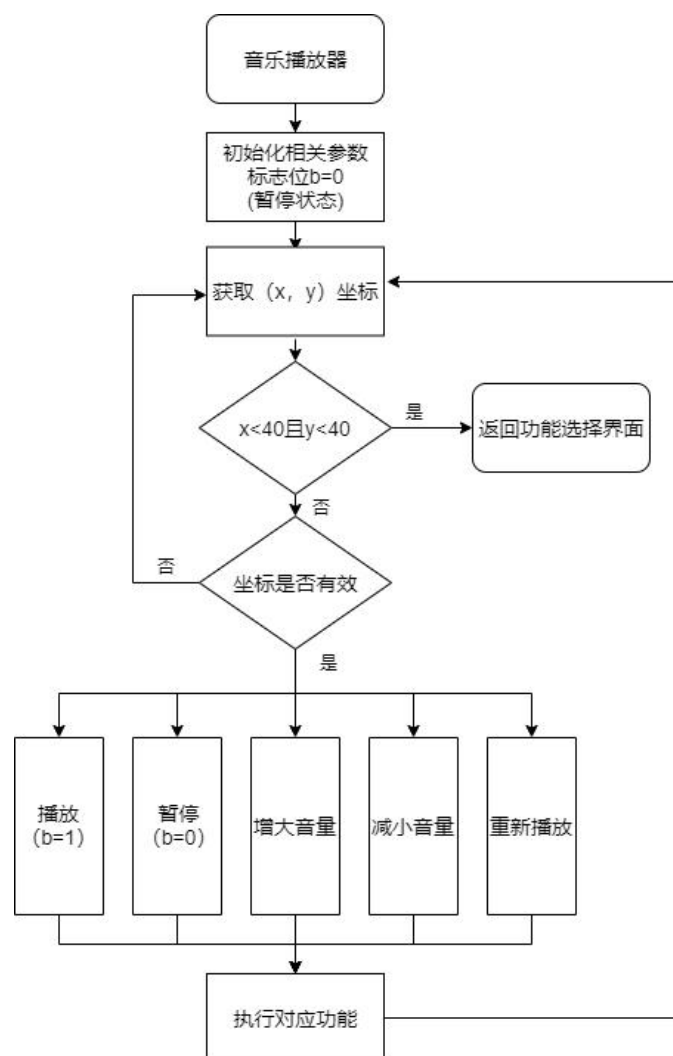
音量和音调，点击播放按钮播放音乐，旁边的两个按钮可调节音量大小，最右边的按钮可重新播放，点击左上角返回到功能选择界面，播放和暂停的界面如下所示：



(2) 设计思路：

利用触摸屏中断，点击 LCD 显示屏，进入中断，获取(x, y)坐标，执行暂停、播放、增大音量、减小音量和重新播放功能，设置标志位用来保存音乐播放器的当前状态（播放或者暂停），通过此标志位判断 PWM 波何时产生何时停止，每次增加音量或减小音量后对 x 和 y 进行清零，否则在 while 循环中占空比会一直增大或减小直至超出范围。

程序流程图如下所示：



1.2.6 钢琴

(1) 简要介绍:

进入钢琴功能界面，按下钢琴键，可发出对应的音调，点击左上角可返回功能选择界面。



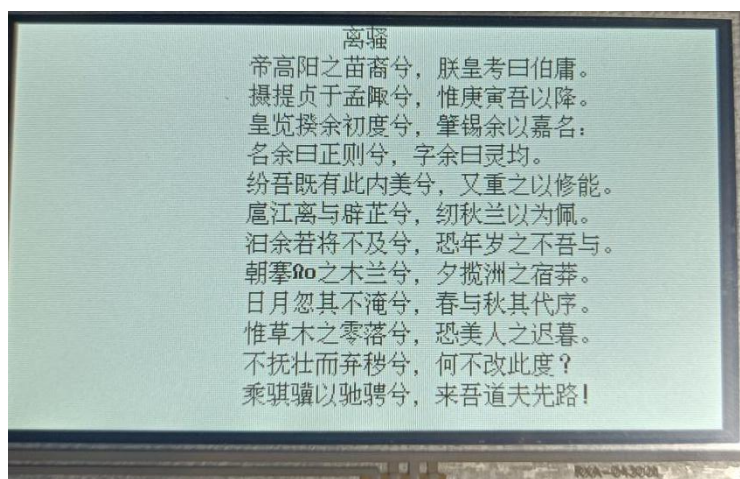
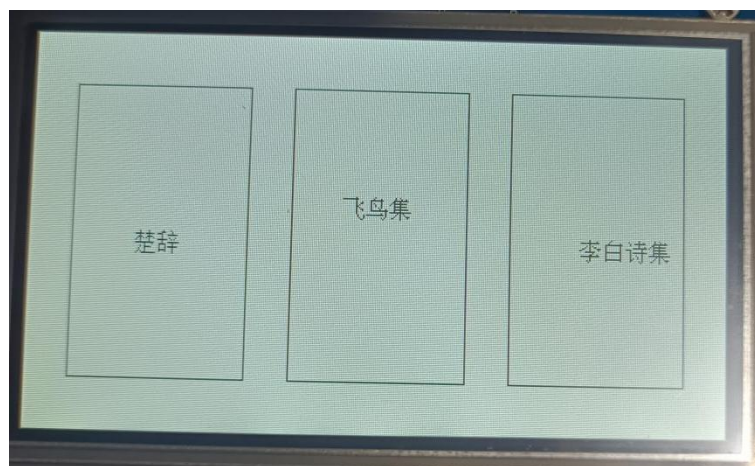
(2) 设计思路:

利用触摸屏中断，按下相应钢琴键，进入中断，获取相应的(x, y)坐标，蜂鸣器播放相应的音调并持续一定时间后关闭，每次对坐标(x, y)清零，防止其一直播放同一音调。

1.2.7 电子书

(1) 简要介绍:

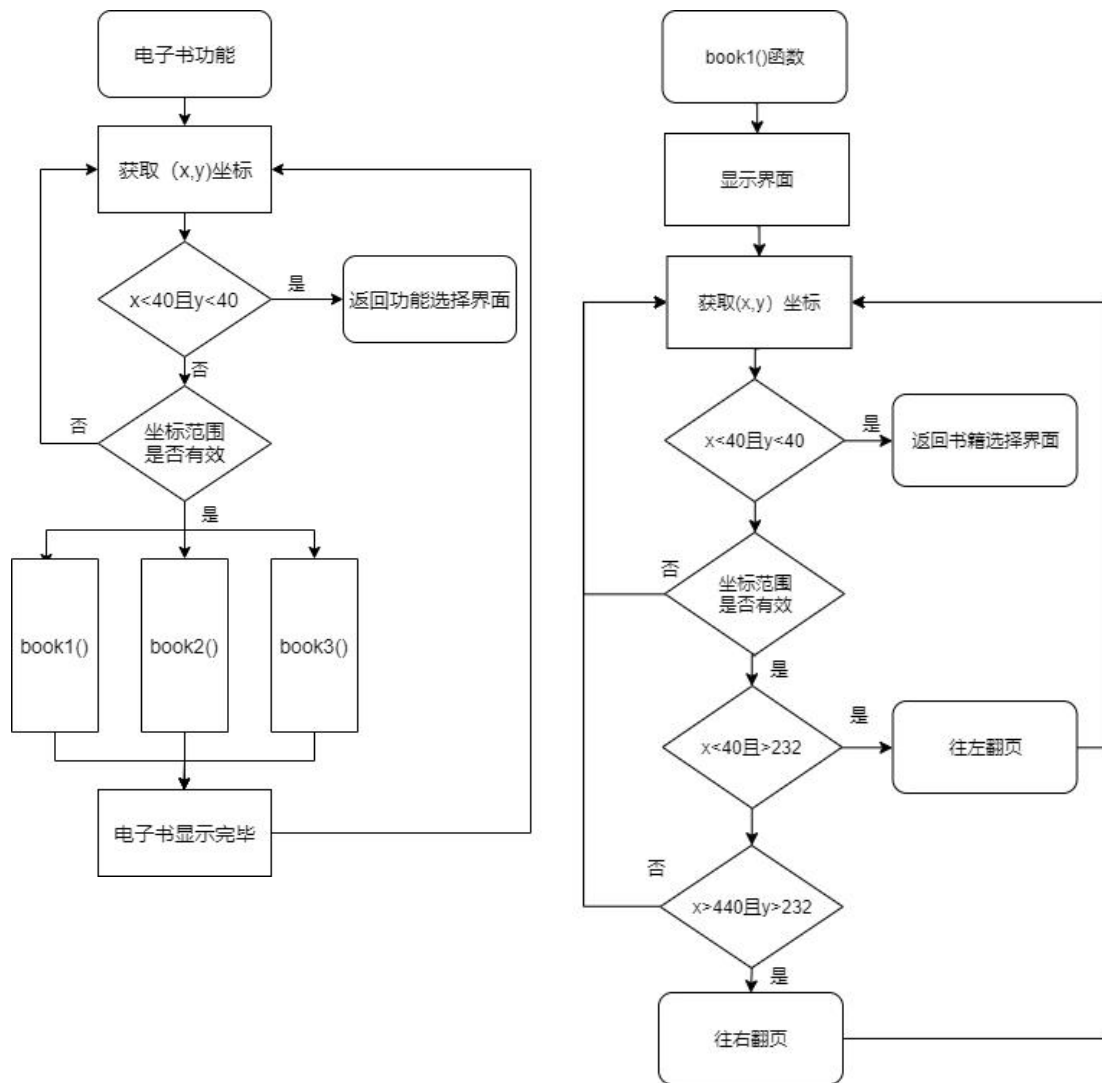
利用中文字库以及与 LCD 显示相关函数，实现了电子书功能，书架共有三本书，点击相应区域即可观看，进入后，可通过点击左下角和右下角切换界面，左上角返回书架，重新选择书籍。



(2) 设计思路:

主要利用字库功能显示对应文字，为了便于书籍内容的显示，将三本书的显示分别作为子函数，在电子书的功能函数中调用，使程序具有更好的结构化。

电子书的程序流程图如下：



1.2.8 手电筒

(1) 简要介绍：

进入手电筒功能，按下按键，对应的 LED 灯亮，当四个灯同时亮的时候，开始执行二进制流水灯程序，开发板上的灯依次点亮。



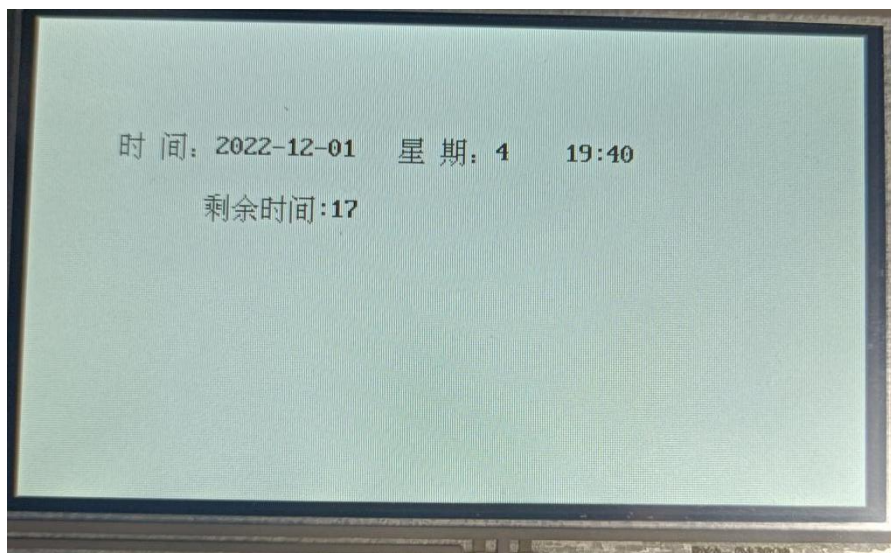
(2) 设计思路:

使用外部中断通过下降沿触发判断按键是否按下,若按键按下,点亮对应的LED灯,注意子中断寄存器 EINTPEND 也需要写一清零,且必须在 SRCPND 和 INTPND 之前清零。如果先清除 SRCPND 的话,然后在清除 EINTPEND 的过程中,SRCPND 会以为又有中断发生,又会置 1。当四个灯都亮,关闭中断,执行流水灯程序。流水灯的实现:利用开发板上的四个 LED 灯,初始化相关引脚,给定初值 0xffff,每次减去 0x20,即可实现二进制流水灯。

1.2.9 闹钟

(1) 简要介绍:

利用定时器的功能,可显示当前时间;也可手动定时,定时时间结束,蜂鸣器报警提醒。



(2) 设计思路:

利用 RTC 的相关寄存器设置相应的年、月、日、星期、时和分,再使用 Lcd_printf() 函数在 LCD 显示屏上显示。设置计时时间,如 1min,即 60s,参数剩余时间设置为 60,每次循环减 1,调用 Delay() 函数延时 1s,更新 LCD 显示屏上的数字,当剩余时间为 0 时,开启蜂鸣器,播放音乐。

1.3 实习心得

1.3.1 遇到问题及解决

(1) 问题一及解决方法

问题:不能正确的生成.bin 文件

解决方法：经过尝试发现当 DebugRel 已存在.bin 文件时,电脑有时会生成不了.bin 文件，将原.bin 文件删除，就可以正常生成.bin 文件。如果是通过 U 盘将一个电脑的文件移动到另一个文件，删除 DebugRel 文件重新编译即可。

(2) 问题二及解决方法

问题：不能显示多张图片，在第 n 张图片显示过后出现紫色等纯色图像

解决方法：不能显示多张照片是由于内存空间不够，在工程中 nand.c 文件中，找到 RdNF2SDRAM 将 U32 size=100000 修改为更大值。

(3) 问题三及解决方法

问题：电子相册大图一直刷新，小图一直显示乱码，图片不能正常显示。

解决方法：经过调试发现，在点击屏幕后，坐标 (x,y) 一直维持着一个值，在发生触摸屏中断（即再次点击）后坐标才会更新，导致一直调用 Paint_Bmp 刷新显示对应图片，每次调用 Paint_Bmp 函数后将 (x,y) 清零即可解决大图刷新问题。小图显示错乱原因和大图一致，设置一个标志位,当大图更新后，将标志位置位，通过判断此标志位是否置位更新小图，更新完成后将标志位置位，就可以使图片显示稳定。

(4) 问题四及解决方法

问题：改变 PWM 波的占空比，蜂鸣器一直不能正常工作，发出声音。

解决方法：经检查发现，调用 Buzzer_Freq_Set1（）函数时传递的参数为 u32 类型，占空比为小数，会发生类型转换，向下取整会变为 0。将传递的参数类型改为 float 类型即可正常工作。

(5) 问题五及解决方法

问题：在一个循环中，每次获取 (x, y) 坐标，如果让音量增大或减小所设定值，就需要将坐标清零，否则会超出占空比的最大范围，而连续播放就要求 (x, y) 坐标一直拥有一个设定范围的值，否则就无法连续播放。音乐播放和暂停的切换与音量增大减小功能不能同时运行，出现自动反复切换播放与暂停界面，音乐播放不连续现象。

解决方法：设置两个标志位，b 和 c，b 用来记录当前音乐播放器的状态（b=1 为播放状态，b=0 为暂停状态）方便播放和暂停的切换，在播放过程中，如果音量增大或减小则将 c 置为 1，通过 c=1 作为判断条件实现连续播放，即可解决该问题。

(6) 问题六及解决方法

问题：电子书矩形框显示错误，无法显示正常的矩形框。

解决方法：在使用 `Glib_Rectangle()` 函数时误认为参数与 `Paint_Bmp()` 函数一致，前者的前两个参数表示矩形框左上角坐标，接下来的两个参数代表矩形框右下角坐标；而后者的前两个参数表示图片左上角坐标，接下来的两个参数分别表示图片的长和宽。区分好对应参数的含义即可解决问题。

(7) 问题七及解决方法

问题：电子书显示时出现字符显示重叠，第二页的字符和第一页字符重叠。

解决方法：在显示第二页前使用 `Lcd_ClearScr()` 函数将 LCD 屏清屏，然后再使用 `Lcd_printf()` 显示文字。

(8) 问题八及解决方法

问题：计算器计算结果正确，但在 LCD 显示屏上显示错误

解决方法：`int` 类型转换为 `char` 类型时，自动转换数字对应的 ASCII 码，将其减去 48 即可，或者可以直接使用占位符，将 `int` 类型的数据直接显示。

1.3.2 收获感想

在本次嵌入式实习中，我学习到了 LCD 显示屏、触摸屏和 PWM 模块的使用以及字库移植，结合课堂上学习到的定时器和中断相关内容，和队友合作设计了一个属于自己的嵌入式系统，对 S3C2440 开发板相关模块的使用更加熟练。

在次实习中，我们制作的系统包含多个功能，我认为这对程序的结构化要求比较高。我们设计了一个功能选择函数 `function()`，在 `function()` 内调用相应的功能函数，使程序具有了更好的结构化，方便了对各个功能的函数进行修改和调试，不使用相关功能时，将调用的函数注释掉即可。

在本次设计中，我遇到了很多问题，但通过上网查资料和请教同学和老师成功解决了相关问题。刚开始编程时，我们对于开发板的相关模块的使用不够熟练，在实习过程中遇到了很多问题，但随着解决问题能力的提高以及对相关代码的熟悉，在后续过程中遇到的问题逐渐减少。我认为，嵌入式设计开发与单片机开发的编程思想都是通用的，在单片机开发中的编程思路完全可以在嵌入式开发中使用，但也要在使用过程中明确两者的区别，把握好两者的相同之处和不同之处，对于后续的学习和发展都有好处。

通过这次实习，提高了我的动手能力与实践能力，将课本上的相关知识和实践结合，加深了我对理论知识的理解，也让我学习到了很多编程知识。另外，我也发现了自身很多的不足之处，设计的系统功能也不是很完善，与别人相比还有

很大差距，编程能力还有待提高，以后会加强这方面的锻炼。

综上所述，在本次实习中，我受益匪浅。在裸机系统的设计和开发中，培养了我的设计和开发能力，初步完成了首个属于自己的系统，体验了嵌入式开发的过程，也让我认识到自己的不足。本次实习让我对后续自身的发展和学习方向有了更清楚的认识，为我学习和使用 **STM32** 等其他嵌入式设备打下了良好的基础。

第二部分：Linux 系统体验实习

2.1 对 linux 系统、移植、驱动开发的理解

2.1.1 对 linux 系统的理解

linux，全称 GNU/Linux，是一种免费使用和自由传播的类 UNIX 操作系统，它主要受到 Minix 和 Unix 思想的启发，是一个基于 POSIX 的多用户、多任务、支持多线程和多 CPU 的操作系统。它能运行主要的 Unix 工具软件、应用程序和网络协议。它支持 32 位和 64 位硬件。Linux 继承了 Unix 以网络为核心的设计思想，是一个性能稳定的多用户网络操作系统。

(1) Linux 的基本思想有两点：第一，一切都是文件；第二，每个文件都有确定的用途。其中第一条详细来讲就是系统中的所有都归结为一个文件，包括命令、硬件和软件设备、操作系统、进程等等对于操作系统内核而言，都被视为拥有各自特性或类型的文件。

(2) 完全免费：Linux 是一款免费的操作系统，用户可以通过网络或其他途径免费获得，并可以任意修改其源代码。

(3) 多用户、多任务：Linux 支持多用户，各个用户对于自己的文件设备有自己特殊的权利，保证了各用户之间互不影响。多任务则是现代电脑最主要的一个特点，Linux 可以使多个程序同时并独立地运行。

(4) 良好的界面：Linux 同时具有字符界面和图形界面。在字符界面用户可以通过键盘输入相应的指令来进行操作。它同时也提供了类似 Windows 图形界面的 X-Window 系统，用户可以使用鼠标对其进行操作。

(5) 支持多种平台：Linux 可以运行在多种硬件平台上，如具有 x86、680x0、SPARC、Alpha 等处理器的平台。同时 Linux 也支持多处理器技术。多个处理器同时工作，使系统性能大大提高。

2.1.2 对 linux 系统移植的理解

一个完整的 Linux 系统移植包含 3 个部分：(1) bootloader 的制作；(2) Linux 内核的移植；(3) 根文件系统的制作。本次实习由于时间限制，我们只进行了 Linux 内核的移植，分为以下几个步骤：(1) 获取 linux 系统源码；(2) 解压系统源码；(3) 在系统中添加对 ARM 的支持；(4) 修改平台输入时钟；

(5) 制作 TQ2440 的内核配置单；(6) 修改机器码；(7) 编译并加载镜像文件。

一般来讲 Linux 系统移植是把 Linux 从一个已经支持的硬件架构上移植到一个还未支持的硬件架构上。由于 Linux 最初是 Linus Trovalds 为 x86 架构写的，因此一般 Linux 系统移植是把 x86 版本移植到别的架构上，例如 arm。

我对 linux 系统移植的理解就是把程序代码从一种运行环境转移到另一种运行环境，在已有 linux 系统源代码的基础上修改配置和参数，使其成为可以在指定硬件系统上运行的 linux 系统，包括 bootloader 的制作、Linux 内核的移植和根文件系统的制作三部分。

2.1.3 对 linux 系统驱动开发的理解

驱动程序一般指的是设备驱动程序，是一种可以使计算机和设备进行相互通信的特殊程序。相当于硬件的接口，操作系统只有通过这个接口，才能控制硬件设备的工作，假如某设备的驱动程序未能正确安装，便不能正常工作。如果一个硬件设备没有驱动程序，只有操作系统是不能发挥特有功能的，也就是说驱动程序是介于操作系统与硬件之间的媒介，实现双向的传达，即将硬件设备本身具有的功能传达给操作系统，同时也将操作系统的标准指令传达给硬件设备，从而实现两者的无缝连接。

Linux 驱动具体包含三种，一般后两种，处理器厂家都会提供 BSP 支持包，开发中涉及最多的就是字符设备：

(1) 字符设备，准确的说应该叫“字节设备”，软件操作设备时是以字节为单位进行的。典型的如 LCD、串口、LED、蜂鸣器、触摸屏.....

(2) 块设备，块设备是相对于字符设备定义的，块设备被软件操作时是以块（多个字节构成的一个单位）为单位的。设备的块大小是设备本身设计时定义好的，软件是不能去更改的，不同设备的块大小可以不一样。常见的块设备都是存储类设备，如：硬盘、NandFlash、iNand、SD.....

(3) 网络设备，网络设备是专为网卡设计的驱动模型，linux 中网络设备驱动主要目的是为了支持 API 中 socket 相关的那些函数工作。

在实习过程中，我们进行了 LCD、触屏和 USB 的驱动移植以及 Nand Flash 的驱动移植，也尝试编写了自己的驱动程序——hello 和 Led 驱动程序，并通过 U 盘实现了在开发板上加载和卸载程序模块。我认为，驱动的实现是为了操作硬件，使硬件运行在某种工作状态，驱动是连接硬件与内核的桥梁，驱动程序向下兼容硬件，向上提供 linux 系统操作硬件的相关接口，对于顶层的应用软件，无法直接调动硬件，是通过操作系统调用驱动程序提供的接口间接进行的。

对于 linux 驱动开发，我的理解是大部分设备驱动已经包含在 linux 内核中或者已经由厂商提供，但是不同的厂商硬件设备不同，我们需要对内核进行一些参数和配置的修改，完成设备驱动移植，使 linux 系统能够正常驱动相关硬件工作；当然，我们也可以编写自己的驱动程序，通过 insmod 和 rmmod 命令来加载和卸

载驱动模块。

2.2 基于 linux 系统的应用开发与裸机开发的区别

对于 ARM 的应用开发主要有两种方式：一种是直接在 ARM 芯片上进行应用开发，不采用操作系统，也称为裸机编程，这种开发方式主要应用于一些低端的 ARM 芯片上，其开发过程非常类似单片机。

还有一种是在 ARM 芯片上运行操作系统，对于硬件的操作需要编写相应的驱动程序，应用开发则是基于操作系统的，这种方式的嵌入式应用开发与单片机开发差异较大。下面简要分析一下两种开发方式的不同之处：

2.2.1 应用开发环境的硬件设备不同

裸机开发：开发板，仿真器（调试器），USB 线；

Linux 开发：开发板，网线，串口线，SD 卡；

对于 Linux 开发，通常是没有硬件的调试器的，尤其是在应用开发的过程中，很少使用硬件的调试器，程序的调试主要是通过串口进行调试的；但是需要说明的是，对于 ARM 芯片也是有硬件仿真器的，但通常用于裸机开发。

2.2.2 程序下载方式不同

裸机开发：仿真器（调试器）下载，或者是串口下载；

Linux 开发：串口下载、tftp 网络下载、或者直接读写 SD、MMC 卡等存储设备，实现程序下载；

这个与开发环境的硬件设备是有直接关系的，由于没有硬件仿真器，故 Linux 开发时通常不采用仿真器下载；这样看似不方便，其实给 ARM-Linux 的应用开发提供了更多的下载方式。

2.2.3 固件的存储位置不同

裸机开发：通常具备 flash 存储器，固件程序通常存储在该区域，若固件较大则需要通过外部电路设计外部 flash 用于存储固件。

Linux 开发：由于其一般没有片内的 flash，并且需要运行操作系统，整个系统映像通常较大，故 ARM-Linux 开发的操作系统映像和应用通常存储在外部的 MMC、SD 卡上，或者采用 SATA 设备等。

2.2.4 驱动不同

裸机驱动一般针对没有操作系统支持的层面，不用考虑操作系统对它的调用。Linux 驱动是在裸机驱动基础上，按照一定的规范来实现，虽然实现的都是同一个东西，Linux 驱动掺杂了许多维护信息，Linux 设备驱动比裸机驱动多了一些框架。

所谓裸机主要是指系统软件平台没有用到操作系统。在基于 ARM 处理器平台的软件设计中，如果整个系统只需要完成一个相对简单而且独立的任务，那么可以不使用操作系统，只需要考虑在平台上如何正确地执行这个单任务程序。不

过，在这种方式下同样需要一个 **Boot Loader**，这个时候的 **Boot Loader** 一般是一个简单的启动代码加载程序。

在裸机方式下，**ARM** 的软件集成开发环境就显得极为重要，因为在这种方式下可以把所有代码都放在这个环境里面编写、编译和调试。在这种方式下测试驱动程序，首先要完成 **CPU** 的初始化，然后把需要测试的程序装载到系统的 **RAM** 区/或者 **SDRAM** 中。当然，如果需要处理一些复杂的中断处理的话，最好也把 **CPU** 的复位向量表放到 **RAM** 区中。把所有程序都调试好之后，再把最后的程序烧写到 **Flash** 里面去执行。

而在 **linux** 开发下，**Bootloader** 就是一小段程序，它在系统上电时开始执行，初始化硬件设备、准备好软件环境、最后调用操作系统内核。

Bootloader 的实现非常依赖具体的硬件，在嵌入式系统中 硬件配置千差万别，即使是相同的 **CPU**，它的外设（比如 **Flash**）也可能不同，所以不可能有一个 **Bootloader** 支持所有的 **CPU**，所有的电路板。即使是支持 **CPU** 架构比较多的 **U-Boot**，也不是一拿来就可以使用的，需要进行一些移植。

Boot Loader 对 **Nand Flash** 的操作是为读取（装载）存储在 **Nand** 上的内核，当内核已由 **Boot Loader** 读取到内存中去后，系统控制权交与内核控制，此时，一般是 **Boot Loader** 中的一些参数内核需要用到，其它基本上没有用了，原来用于存放 **Boot Loader** 的内存（有的开发板 **Nand Flash** 中有前面的 **4K** 存储空间与内存地址重叠，小点的 **Boot Loader** 可直接在这 **4K** 的空间中执行）基本上可以由内核回收利用，所以内核不会用到 **Boot Loader** 上的 **Nand** 的驱动，也就就是内核需要自己的 **Nand Flash** 驱动，不然文件系统无法加载。

2.2.5 启动方式不同

裸机开发：其结构简单，通常是芯片厂商在程序上电时加入固定的跳转指令，直接跳转到程序入口（通常在 **flash** 上）；开发的应用程序通过编译器编译，采用专用下载工具直接下载到相应的地址空间；所以系统上电后直接运行到相应的程序入口，实现系统的启动。

Linux 开发：由于采用 **ARM** 芯片，执行效率高，功能强大，外设相对丰富，是功能强大的计算机系统，并且需要运行操作系统，所以其启动方式和单片机有较大的差别，但是和家用计算机的启动方式基本相同。其启动一般包括 **BIOS**，**bootloader**，内核启动，应用启动等阶段：

(a)启动 **BIOS**: **BIOS** 是设备厂家（芯片或者是电路板厂家）设置的相应启动信息，在设备上电后，其将读取相应硬件设备信息，进行硬件设备的初始化工作，然后跳转到 **bootloader** 所在位置（该位置是一个固定的位置，由 **BIOS** 设置）。**BIOS** 的启动和单片机启动类似，需要采用相应的硬件调试器进行固件的写入，

存储在一定的 flash 空间，设备上电启动后读取 flash 空间的指令，从而启动 BIOS 程序。

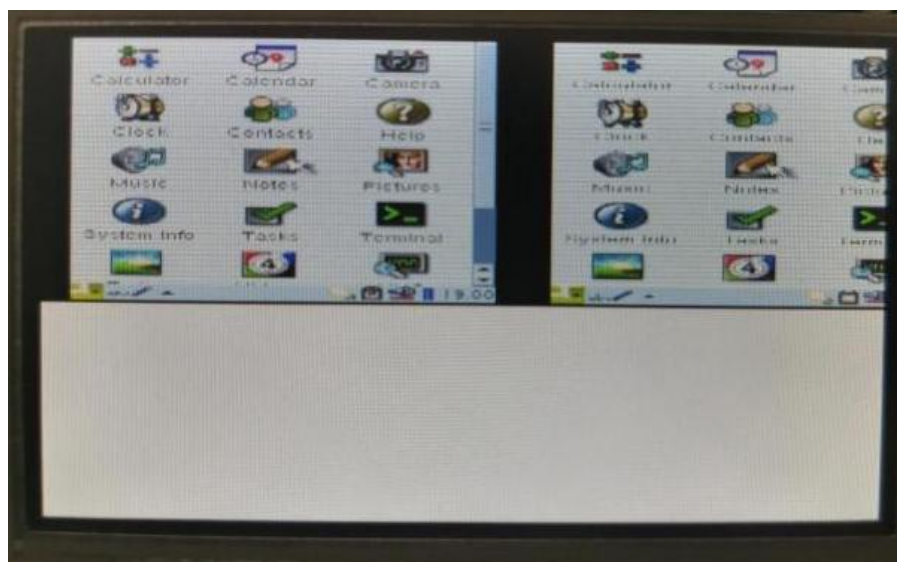
(b)启动 bootloader: 该部分已经属于嵌入式 Linux 软件开开的部分,可以通过代码修改定制相应的 bootloader 程序,bootloader 的下载通常是采用直接读写 SD 卡等方式。即编写定制相应的 bootloader,编译生成 bootloader 映象文件后,利用工具(专用或通用)下载到 SD 卡的 MBR 区域(通常是存储区的第一个扇区)。此时需要在 BIOS 中设置,或者通过电路板的硬件电路设置,选择 bootloader 的加载位置;若 BIOS 中设置从 SD 卡启动,则 BIOS 初始化结束后,将跳转到 SD 卡的位置去执行 bootloader,从而实现 bootloader 的启动。Bootloader 主要作用是初始化必要的硬件设备,创建内核需要的一些信息并将这些信息通过相关机制传递给内核,从而将系统的软硬件环境带到一个合适的状态,最终调用操作系统内核,真正起到引导和加载内核的作用。

(c)启动内核 : bootloader 启动完成初始化等相关工作之后,将调用内核启动程序。这就进入了实际的操作系统相关内容的启动了,包括相应的硬件配置,任务管理,资源管理等内核程序的启动。

(d)启动应用: 在操作系统内核启动之后,就可以开始启动需要的应用,去完成真正的业务操作了。

2.3 成果展示

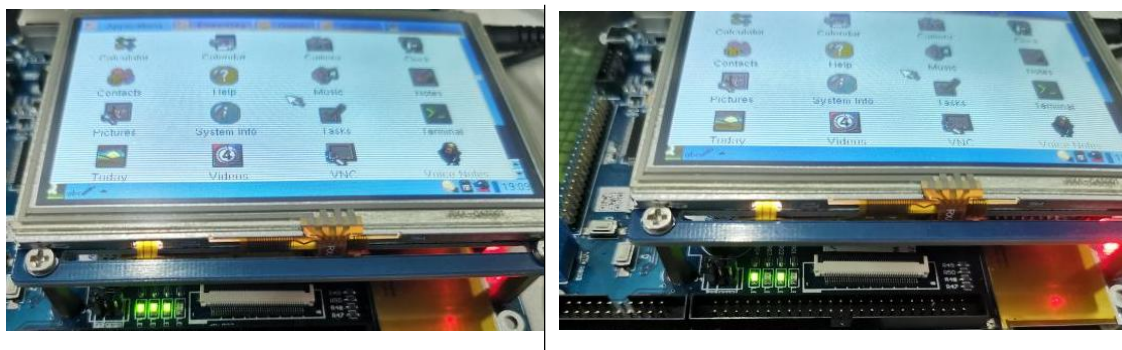
2.3.1 只导入内核系统和根文件系统后的嵌入式系统



2.3.2 导入内核系统、根文件系和驱动后的嵌入式系统



2.3.3 LDE 控制效果



2.4 实习心得

2.4.1 遇到问题及解决

(1) 问题一及解决方法

问题：在第二节实现“我的”内核的移植时，`make zImage` 没有生成 `zImage.bin` 文件。

解决方法：修改 `Makefile` 文件时，添加 “`@cp -f arch/arm/boot/zImage zImage.bin`” 时 “-f” 前缺少空格，将空格添加上。在解压缩后，我们对内核源码 `linux-2.6.30.4` 进行过移动，导致部分文件丢失，重新解压即可解决问题。

(2) 问题二及解决方法

问题：在第四节实现“我的”驱动移植时配置内核找不到“`Framebuffer Console support`”这个选项。

解决方法：仔细寻找配置清单，该选项的路径与指导书的路径不同，在其他

路径中可以找到，默认不需要修改。

(3) 问题三及解决方法

问题：在进行开机 LOGO 功能时，虽然完成了指导书中的步骤修改，但无法生成镜像文件。

解决方法：缺少对应的 LOGO 文件，将老师提供的 LOGO 文件放在对应的路径下即可生成镜像文件。

(4) 问题四及解决方法

问题：使用命令 “\$led 1 0” LED 灯没有反映

解决方法：仔细看指导书，\$只是代表此时为开发板的命令行，删掉“\$”即可。注意区分 PC 机 Linux 和开发板 Linux 的相关指令，PC 机端 Linux 的命令行用“#”，开发板上的 Linux 的命令行用“\$”，但它们并不是命令内容。

2.4.2 收获感想

在 Linux 系统体验实习过程中，我初步体验了 Linux 系统开发的流程，对 Linux 系统的开发有了大致了解。Linux 系统开发的体量很大，我们的实习只是截取了其中一小部分，在仔细参考指导书的基础上，我们还是遇到了很多问题。对于相关文件相关内容以及配置清单的相关配置的修改一定要注意细节，否则镜像文件无法正常生成，还需要花费大量时间寻找错误。可以想象在实际的 Linux 系统移植中遇到的问题只会更多，Linux 系统的开发需要学习大量内容。

在本次实习中，我对 Linux 系统下的命令行更加熟练，学习到了 Linux 内核的简单移植、根文件系统的移植和各种驱动模块的移植与开发，加深了我对 Linux 开发的兴趣，增强了我的实践能力，让我对 Linux 的专业知识有了更深的理解。

当然，本次实习只是让我们简单了解了 Linux 开发的流程，要想从事相关工作，还需要学习很多内容。就像老师在开篇说的“需要耐心，还需要百折不挠的精神”，在学习嵌入式开发过程中，这句话伴随始终。不管是在学习中还是生活中，个人的能力可以在不断地发现错误并解决问题的过程中迅速提升。我在以后学习专业知识的过程中，会牢记老师的教诲，努力寻找解决问题的方法。

实习意见建议

通过此次实习，我设计了一个裸机系统，初步体验了 linux 操作系统的移植，实际操作了 linux 开发的过程。结合我的自身经历和实习过程中遇到的问题，我想提出以下建议：

1. 实验室的电脑 USB 驱动存在问题，三台电脑中往往只有一台电脑能够使用，有的电脑 USB 驱动能够使用三个小时左右后会出现问题，导致开发板不能连接。在裸机开发中，保存好代码重启就可以，但是在 linux 操作系统开发中，重启后虚拟机中的配置内容全都会消失，所有工作都需要重新开始，如果可以保存虚拟机的内容可以提高效率，或者提供解决 USB 驱动失效问题的方法；
2. 实习报告的相关要求可以在实习之前发出来，同学们可以对自己设计的系统相关功能实现进行拍照保存，方便后续报告的书写；
3. 音乐转换为数组时在网上找到的转换软件都不是很理想，希望老师可以提供相关软件，或者在实习群里搜集本次实习效果比较好的软件，方便下一届的同学们进行操作；
4. 第二部分的实习时间比较短，开发板每个组只有一块，资源有限，代码写好后需要花费大量时间调试，很难设计出理想的系统；而第三部操作比较简单，遇到的问题也比较少，可以适当增加第二部分的实习时间，减少第三部分的实习时间；
5. linux 系统体验实习感觉一直在照着实验指导书修改文件内容和配置清单，但是不知道修改的部分的作用，可以提供适当的资料方便同学们进一步理解。

附录:

1.密码功能函数

```

void password(void){
int mima[]={1,2,3,4};
int mimal[4];
int flag1=1;

while(flag1){
    int flag=1;
    int t=0;
    Paint_Bmp(0, 0, 480,272, suoping);
    Lcd_printf(300,110,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"");
    Lcd_printf(300,150,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"");
    Glib_Rectangle(290,60,380,90,0x0000);
    while(flag){

        if(xdata>0 && xdata<217 && ydata>11 && ydata<227){
            if(xdata>25 && xdata<76 && ydata>25 && ydata <76){
                mimal[t]=1;
                Lcd_printf(300+20*t,70,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"1");
            }

            if(xdata>76 && xdata<152 && ydata>25 && ydata <76){
                mimal[t]=2;
                Lcd_printf(300+20*t,70,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"2");
            }

            if(xdata>152 && xdata<217 && ydata>25 && ydata <76){
                mimal[t]=3;
                Lcd_printf(300+20*t,70,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"3");
            }
        }
    }
}

```

```

if(xdata>25 && xdata<76 && ydata>76 && ydata <126){
    mimal[t]=4;
    Lcd_printf(300+20*t,70,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"4");
}

if(xdata>76 && xdata<152 && ydata>76 && ydata <126){
    mimal[t]=5;
    Lcd_printf(300+20*t,70,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"5");
}

if(xdata>152 && xdata<217 && ydata>76 && ydata <126){
    mimal[t]=6;
    Lcd_printf(300+20*t,70,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"6");
}

if(xdata>25 && xdata<76 && ydata>126 && ydata <176){
    mimal[t]=7;
    Lcd_printf(300+20*t,70,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"7");
}

if(xdata>76 && xdata<152 && ydata>126 && ydata <176){
    mimal[t]=8;
    Lcd_printf(300+20*t,70,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"8");
}

if(xdata>152 && xdata<217 && ydata>126 && ydata <176){
    mimal[t]=9;
    Lcd_printf(300+20*t,70,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"9");
}

if(xdata>76 && xdata<152 && ydata>176 && ydata <227){
    mimal[t]=0;
    Lcd_printf(300+20*t,70,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"0");
}

t++;

```

```

xdata=0;

ydata=0;

}

if(t==4){
    flag=0;
    Delay(800);
    Lcd_printf(300,70,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"*");
    Lcd_printf(320,70,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"*");
    Lcd_printf(340,70,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"*");
    Lcd_printf(360,70,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"*");
}

}

if(mima[0]==mima1[0] && mima[1]==mima1[1] && mima[2]==mima1[2] &&
mima[3]==mima1[3]){
    Lcd_printf(300,110,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"密码正确");
    Lcd_printf(300,150,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"欢迎使用智能手
机");

    Delay(800);
    Lcd_ClearScr(0);
    flag1=0;
}

else{
    Lcd_printf(300,110,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"密码错误");
    Lcd_printf(300,150,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"请重新输入密码
");

    Delay(800);
}

}

```

```
}
```

2.电子相册功能函数

```
void picture(void){
    int pix_x=480;
    int pix_y=272;
    int a=0;
    Lcd_ClearScr(0xffffffff);
    Paint_Bmp(0, 0, 480,272, bp1);
    //Delay(50);

    Glib_FilledRectangle(8,222,78,262,0xFFFFFFFF);
    Paint_Bmp(8, 222, 70, 40, p1);
    //Delay(50);

    Glib_FilledRectangle(87,222,157,262,0xFFFFFFFF);
    Paint_Bmp(87,222, 70, 40, p2);
    //Delay(50);

    Glib_FilledRectangle(166,222,236,262,0xFFFFFFFF);
    Paint_Bmp(166,222, 70, 40, p3);
    //Delay(50);

    Glib_FilledRectangle(245,222,315,262,0xFFFFFFFF);
    Paint_Bmp(245,222, 70,40, p4);
    //Delay(50);

    Glib_FilledRectangle(324,222,394,262,0xFFFFFFFF);
    Paint_Bmp(324,222, 70, 40, p5);
    //Delay(50);

    Glib_FilledRectangle(403,222,473,262,0xFFFFFFFF);
    Paint_Bmp(403, 222, 70, 40, p6);
    //Delay(50);

    while(1)
```



```

{
    if(ydata>2*pix_y/3)
    {if(xdata<pix_x/6)
        {
            Paint_Bmp(0, 0, 480,272, bp1);

            a=1;
        }
        else if(xdata>pix_x/6&& xdata<pix_x/3)
        {
            Paint_Bmp(0, 0, 480,272, bp2);

            a=1;
        }
        else if(xdata>pix_x/3&& xdata<pix_x/2)
        {
            Paint_Bmp(0, 0, 480,272, bp3);

            a=1;
        }
        else if(xdata>pix_x/2&& xdata<pix_x*2/3){
            Paint_Bmp(0, 0, 480,272, bp4);

            a=1;
        }
        else if(xdata>pix_x*2/3&& xdata<pix_x*5/6){
            Paint_Bmp(0, 0, 480,272, bp5);

            a=1;
        }
        else if(xdata>pix_x*5/6){
            Paint_Bmp(0, 0, 480,272, bp6);

            a=1;
        }
    }
}

```

```

        xdata=0;

        ydata=0;

    }

    if(a==1){

        //Delay(50);

        Glib_FilledRectangle(8,222,78,262,0xFFFFFFFF);

        Paint_Bmp(8, 222, 70, 40, p1);

        //Delay(50);

        Glib_FilledRectangle(87,222,157,262,0xFFFFFFFF);

        Paint_Bmp(87,222, 70, 40, p2);

        //Delay(50);

        Glib_FilledRectangle(166,222,236,262,0xFFFFFFFF);

        Paint_Bmp(166,222, 70, 40, p3);

        //Delay(50);

        Glib_FilledRectangle(245,222,315,262,0xFFFFFFFF);

        Paint_Bmp(245,222, 70,40, p4);

        //Delay(50);

        Glib_FilledRectangle(324,222,394,262,0xFFFFFFFF);

        Paint_Bmp(324,222, 70, 40, p5);

        //Delay(50);

        Glib_FilledRectangle(403,222,473,262,0xFFFFFFFF);

        Paint_Bmp(403, 222, 70, 40, p6);

        //Delay(50);

        a=0;

    }

    if(xdata>0&& xdata<40&&ydata>0&&ydata<40){

        Paint_Bmp(0, 0, 480,272, background);

        return;

    }

```

```

    }
}

int pow1(int x,int y){
int w=1;
while(y){
w*=x;
--y;
}
return w;
}

```

3.计算器功能函数

```

void calculator(){

int num_str[20];
char oper_arr[20];
int num_str_length=0;
int num_length=0;
int oper_length=0;
int num_adr=1;//从（20，70）开始横着显示吧
int j=0;
int i=0;
int m;
char num_arr[20];
char *result;
const char* result1;
int temp;
char* aaa;

Paint_Bmp(0,0,480,272,calculator_back);

Lcd_printf(20,70,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"0
");

```

```

while(1){

    if(xdata>225&&xdaxdata<285&&ydata>160&&ydata<210){

        num_str[num_str_length++]=1;

        Lcd_printf(20*num_adr,70,0x0000,(0x1f<<11) | (0x3f<<5) |
(0x1f),1,"1");

        num_adr++;

    }

    else if(xdata>285&&xdaxdata<348&&ydata>160&&ydata<210){

        num_str[num_str_length++]=2;

        Lcd_printf(20*num_adr,70,0x0000,(0x1f<<11) | (0x3f<<5) |
(0x1f),1,"2");

        num_adr++;

    }

    else if(xdata>348&&xdaxdata<410&&ydata>160&&ydata<210){

        num_str[num_str_length++]=3;

        Lcd_printf(20*num_adr,70,0x0000,(0x1f<<11) | (0x3f<<5) |
(0x1f),1,"3");

        num_adr++;

    }

    else if(xdata>225&&xdaxdata<285&&ydata>110&&ydata<160){

        num_str[num_str_length++]=4;

        Lcd_printf(20*num_adr,70,0x0000,(0x1f<<11) | (0x3f<<5) |
(0x1f),1,"4");

        num_adr++;

    }

    else if(xdata>285&&xdaxdata<348&&ydata>110&&ydata<160){

        num_str[num_str_length++]=5;

        Lcd_printf(20*num_adr,70,0x0000,(0x1f<<11) | (0x3f<<5) |

```

```

(0x1f),1,"5");

    num_adr++;
}
else if(xdata>348&&xdata<410&&ydata>110&&ydata<160){
    num_str[num_str_length++]=6;
    Lcd_printf(20*num_adr,70,0x0000,(0x1f<<11) | (0x3f<<5) |
(0x1f),1,"6");

    num_adr++;
}
else if(xdata>225&&xdata<285&&ydata>60&&ydata<110){
    num_str[num_str_length++]=7;
    Lcd_printf(20*num_adr,70,0x0000,(0x1f<<11) | (0x3f<<5) |
(0x1f),1,"7");

    num_adr++;
}
else if(xdata>285&&xdata<348&&ydata>60&&ydata<110){
    num_str[num_str_length++]=8;
    Lcd_printf(20*num_adr,70,0x0000,(0x1f<<11) | (0x3f<<5) |
(0x1f),1,"8");

    num_adr++;
}
else if(xdata>348&&xdata<410&&ydata>60&&ydata<110){
    num_str[num_str_length++]=9;
    Lcd_printf(20*num_adr,70,0x0000,(0x1f<<11) | (0x3f<<5) |
(0x1f),1,"9");

    num_adr++;
}
else if(xdata>285&&xdata<348&&ydata>210&&ydata<260){
    num_str[num_str_length++]=0;

```

```
Lcd_printf(20*num_adr,70,0x0000,(0x1f<<11) | (0x3f<<5) |  
(0x1f),1,"0");  
  
    num_adr++;  
}  
else if(xdata>420&&xdxdata<455&&ydata>170&&ydata<200){  
    if(num_str_length=1){  
        num_arr[num_length++]=num_str[0];  
    }  
    else{  
        for(j=0;j<num_str_length;j++){  
            num_arr[num_length]+=  
num_str[j]*pow1(10,num_str_length-j-1);  
        }  
        num_length++;    //num_arr[0]对应长度为 1  
    }  
  
    for(j=0;j<num_str_length;j++){  
        num_str[j]=0;  
    }  
    num_str_length=0;  
  
    oper_arr[oper_length++]='+';  
    Lcd_printf(20*num_adr,70,0x0000,(0x1f<<11) | (0x3f<<5) |  
(0x1f),1,"+");  
  
    num_adr++;  
}  
else if(xdata>420&&xdxdata<455&&ydata>120&&ydata<150){  
    if(num_str_length=1){  
num_arr[num_length++]=num_str[0];
```

```

    }

    else{

        for(j=0;j<num_str_length;j++){

            num_arr[num_length]+=

num_str[j]*pow1(10,num_str_length-j-1);

            }

            num_length++;    //num_arr[0]对应长度为 1

        }

        for(j=0;j<num_str_length;j++){

            num_str[j]=0;

        }

        num_str_length=0;

        oper_arr[oper_length++]='-';

        Lcd_printf(20*num_adr,70,0x0000,(0x1f<<11) | (0x3f<<5) |

(0x1f),1,"-");

        num_adr++;

    }

    else if(xdata>420&&xdta<455&&ydata>70&&ydata<100){

        if(num_str_length=1){

            num_arr[num_length++]=num_str[0];

        }

        else{

            for(j=0;j<num_str_length;j++){

                num_arr[num_length]+=

num_str[j]*pow1(10,num_str_length-j-1);

                }

                num_length++;    //num_arr[0]对应长度为 1

            }

        }
    }

```



```
for(j=0;j<num_str_length;j++){
    num_str[j]=0;
}
num_str_length=0;

oper_arr[oper_length++]='*';

Lcd_printf(20*num_adr,70,0x0000,(0x1f<<11) | (0x3f<<5) |
(0x1f),1,"*");

num_adr++;
}
else if(xdata>420&&xdxdata<455&&ydata>20&&ydata<50){
    if(num_str_length=1){
num_arr[num_length++]=num_str[0];
    }
    else{
        for(j=0;j<num_str_length;j++){
            num_arr[num_length]+=
num_str[j]*pow(10,num_str_length-j-1);
        }
        num_length++;    //num_arr[0]对应长度为 1
    }

    for(j=0;j<num_str_length;j++){
        num_str[j]=0;
    }
    num_str_length=0;

    oper_arr[oper_length++]='/';

    Lcd_printf(20*num_adr,70,0x0000,(0x1f<<11) | (0x3f<<5) |
```

```

(0x1f),1,"/");

num_adr++;

}

else if(xdata>348&& xdata<410&&ydata>20&&ydata<50){
for(j=0;j<num_str_length;j++){
num_arr[num_length]+=
num_str[j]*pow1(10.0,num_str_length-j-1);
}
num_length++;
for(j=0;j<num_str_length;j++){
num_str[j]=0;
}
num_str_length=0;

oper_arr[oper_length++]='%';
Lcd_printf(20*num_adr,70,0x0000,(0x1f<<11) | (0x3f<<5) |
(0x1f),1,"%");

num_adr++;

}

else if(xdata>285&& xdata<348&&ydata>20&&ydata<50){
for(j=0;j<num_str_length;j++){
num_arr[num_length]+=
num_str[j]*pow1(10.0,num_str_length-j-1);
}
num_length++;
for(j=0;j<num_str_length;j++){
num_str[j]=0;
}

```

```
    }

    num_str_length=0;

    oper_arr[oper_length++]='s';

    Lcd_printf(20*num_adr,70,0x0000,(0x1f<<11) | (0x3f<<5) |
(0x1f),1,"sqrt");

    num_adr=num_adr+4;

    }

    else if(xdata>230&& xdata<270&& ydata>210&& ydata<260){

        for(j=0;j<num_str_length;j++){

            num_arr[num_length]+=

num_str[j]*pow(10.0,num_str_length-j-1);

        }

        num_length++;

        for(j=0;j<num_str_length;j++){

            num_str[j]=0;

        }

        num_str_length=0;

        oper_arr[oper_length++]='d';

        Lcd_printf(20*num_adr,70,0x0000,(0x1f<<11) | (0x3f<<5) |
(0x1f),1,"rec");

        num_adr=num_adr+3;

    }

    else if(xdata>348&& xdata<410&& ydata>210&& ydata<260){

        if(num_str_length=1){

            num_arr[num_length++]=num_str[0];
```

```

    }

else{
    for(j=0;j<num_str_length;j++){
        num_arr[num_length]+=
num_str[j]*pow(10.0,num_str_length-j-1);
    }
    num_length++;    //num_arr[0]对应长度为 1
}

for(j=0;j<num_str_length;j++){
    num_str[j]=0;
}
num_str_length=0;

oper_arr[oper_length++]='^';
Lcd_printf(20*num_adr,70,0x0000,(0x1f<<11) | (0x3f<<5) |
(0x1f),1,"^");

num_adr++;
}

else if(xdata>230&&xdaxdata<270&&ydata>20&&ydata<50){
//按下清零后，数字数组清零、运算符数组清零、显示地址重开
for(j=0;j<num_length;j++){
    num_arr[j]=0;
}
num_length=0;
for(j=0;j<oper_length;j++){
    oper_arr[j]=0;
}
for(j=0;j<num_str_length;j++){
    num_str[j]=0;
}

```

```
    }

    num_length=0;

    num_adr=1;

    num_str_length=0;

    Lcd_printf(20,70,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"0

");

    Lcd_printf(20,200,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"

");

    Lcd_printf(20,150,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"

");

}

else if(xdata>420&& xdata<455&&ydata>210&&ydata<260){

if(num_str_length=1){

    num_arr[num_length++]=num_str[0];

}

else{

    for(j=0;j<num_str_length;j++){

        num_arr[num_length]+=

num_str[j]*pow(10.0,num_str_length-j-1);

    }

    num_length++;    //num_arr[0]对应长度为 1

}

}

for(j=0;j<num_str_length;j++){

    num_str[j]=0;

}

num_str_length=0;
```

```
if(oper_arr[0]=='*'){
    //num_arr[0]=num_arr[0]*num_arr[1];
    temp=num_arr[0]*num_arr[1];
}
else if(oper_arr[0]=='/'){
    //num_arr[0]=num_arr[0]/num_arr[1];
    temp=num_arr[0]/num_arr[1];
}
else if(oper_arr[0]=='+'){
    //num_arr[0]=num_arr[0]+num_arr[1];
    temp=num_arr[0]+num_arr[1];
}
else if(oper_arr[0]=='-'){
    //num_arr[0]=num_arr[0]-num_arr[1];
    temp=num_arr[0]-num_arr[1];
}
else if(oper_arr[0]=='%'){
    //num_arr[0]=num_arr[0]%num_arr[1];
    temp=num_arr[0]%num_arr[1];
}
else if(oper_arr[0]=='s'){
    //num_arr[0]=pow1(num_arr[0],0.5);
    temp=pow1(num_arr[0],0.5);
}
else if(oper_arr[0]=='d'){
    temp=1/num_arr[0];
}
else if(oper_arr[0]=='^'){
```

```

//num_arr[0]=pow1(num_arr[0],num_arr[1]);

                                temp=pow1(num_arr[0],num_arr[1]);

                                }

                                Lcd_printf(20,150,0x0000,(0x1f<<11) | (0x3f<<5) |
(0x1f),1,"= %d",temp);

                                for(j=0;j<num_length;j++){
                                num_arr[j]=0;
                                }

                                num_length=0;

                                }

                                if(xdata>0&&xddata<40&&ydata>0&&ydata<40){
                                Paint_Bmp(0, 0, 480,272, background);
                                return;
                                }
                                xdata=0;
                                ydata=0;
                                }

                                }

```

4.音乐播放器功能函数

```

void Buzzer_Freq_Set0(U32 freq,float freqk ){

    rGPBCON &= ~3;           //set GPB0 as tout0, pwm output

    rGPBCON |= 2;

```

```

rTCFG0 &= ~0xff;

rTCFG0 |= 15;          //prescaler = 15+1

rTCFG1 &= ~0xf;

rTCFG1 |= 2;          //mux = 1/8


rTCNTB0 = (PCLK>>7)/freq;

rTCMPB0 = rTCNTB0*freq;    //修改这一句，来调整占空比，同学们可以自己做实验
尝试

rTCON &= ~0x1f;

rTCON |= 0xb;          //disable    deadzone,    auto-reload,    inv-off,    update
TCNTB0&TCMPB0, start timer 0

rTCON &= ~2;          //clear manual update bit
}

//这个函数中 rTCMPB0 = rTCNTB0*freq;有可能会溢出，使用时要小心
void Buzzer_Freq_Set1(float freq){

    rGPBCON &= ~3;          //set GPB0 as tout0, pwm output

    rGPBCON |= 2;


    rTCFG0 &= ~0xff;

    rTCFG0 |= 15;          //prescaler = 15+1

    rTCFG1 &= ~0xf;

    rTCFG1 |= 2;          //mux = 1/8

    rTCNTB0 = (PCLK>>7)/200;

    rTCMPB0 = rTCNTB0*freq;    // 50%

    rTCON &= ~0x1f;

    rTCON |= 0xb;          //disable    deadzone,    auto-reload,    inv-off,    update

```


TCNTB0&TCMPB0, start timer 0

```
rTCON &= ~2;          //clear manual update bit
}
```

void music(){

unsigned char Temp1,Temp2;

U16 freq=200;// lci 1000

float freqk=0.03;

int x=0;

int y=0;

unsigned int Addr=0;

int temp=0.03;//保存暂停前的音量，即占空比

int b=0; //b=0 暂停状态，b=1 播放状态

int c=0;

Paint_Bmp(0,0,480,272,zanting);//切换背景图片

while(1){

if(xdata>335&&xdata<370&&ydata>215&&ydata<250){//增加音量

if(freqk<1)

freqk+=0.01;

//Buzzer_Freq_Set1(freqk);

c=1;

xdata=0;

ydata=0;

}

else if(xdata>255&&xdata<290&&ydata>215&&ydata<250){//减少音量

if(freqk>0.01)

```

freqk=0.01;

//Buzzer_Freq_Set1(freqk);

c=1;

xdata=0;

ydata=0;

}

else if((xdata>295&&xdata<330&&ydata>215&&ydata<250)||c==1){

    if(b==0){

        x=xdata;

        y=ydata;

        Paint_Bmp(0,0,480,272,bofang);//切换为暂停键，播放开始

        freq=200;

        //freqk=temp;

        Temp1=SONG[Addr++];

        Temp2=SONG[Addr++];

        freq=25000/Temp1;

        //b=1;

        Buzzer_Freq_Set0( freq,freqk);

        Delay(12*Temp2);

        if(Addr>215){

            Addr=0;

        }

    }

    if(xdata>295&&xdata<330&&ydata>215&&ydata<250){

        if(b==0&&(x!=xdata||y!=ydata)){

            b=1;

        }

    }

    else if(b==1){

```

```

//temp=freqk;

Buzzer_Stop();

Paint_Bmp(0,0,480,272,zanting);//切换为开始键，播放暂停

b=0;

c=0;

xdata=0;

ydata=0;

}

}

//xdata=0;

//ydata=0;

}

else if(xdata>375&& xdata<410&&ydata>215&&ydata<250){//重新开始播放

//temp=freqk;

Buzzer_Stop();

Paint_Bmp(0,0,480,272,zanting);//切换为开始键，播放暂停

if(Addr>0&&Addr<215){

    Addr=0;

}

/*Temp1=SONG[Addr++];

Temp2=SONG[Addr++];

freq=25000/Temp1;

Buzzer_Freq_Set0( freq );

Delay(12*Temp2);*/

xdata=0;

ydata=0;

}

if(xdata>0&&xdata<40&&ydata>0&&ydata<40){

```

```

        Buzzer_Stop();

        Paint_Bmp(214,214,48,48,background);//切换为主菜单

        return;

    }

}

}

```

5.钢琴功能函数

```

void play_piano(void){
    int pix_x=480;

    int pix_y=272;

    int a=0;

    U16 freq=200;// lci 1000

    float freqk=0.03;

    U32 piano1[10]={0x30,0x2b,0x26,0x24,0x20,0x1c,0x19,0x18,0x15,0x13};

    //U32 piano[10]={50,150,250,350,450,550,650,1250,1550,1950};

    Lcd_ClearScr(0xffffffff);

    if(xdata<pix_x&&ydata<pix_y){

        Paint_Bmp(0, 0, 480,272, piano);} //刷新为钢琴界面

        //Buzzer_Freq_Set1(freqk);

        while(1)

        {

            if(ydata>183)

            {
                if(xdata<48&&xdata>0)//中 1

                {

                    //Buzzer_Freq_Set0( piano[0] );

                    //Delay(50);

                    //Buzzer_Stop1();

```

```
Beep(piano1[0],50);

Uart_Printf( "Piano0:%d !\n\n",piano[0] );

}

else if(xdata>48&&data<96)//中 2
{
    //Buzzer_Freq_Set0( piano[1] );
    //Delay(50);
    //Buzzer_Stop1();
    Beep(piano1[1],50);
    Uart_Printf( "Piano1:%d !\n\n",piano[1] );
}

else if(xdata>96&&data<144)//中 2
{
    //Buzzer_Freq_Set0( piano[2] );
    //Delay(50);
    //Buzzer_Stop1();
    Beep(piano1[2],50);
    Uart_Printf( "Piano2:%d !\n\n",piano[2] );
}

else if(xdata>144&&data<192)//中 3
{
    //Buzzer_Freq_Set0( piano[3] );
    //Delay(50);
    //Buzzer_Stop1();
    Beep(piano1[3],50);
    Uart_Printf( "Piano3:%d !\n\n",piano[3] );
```

```

}

else if(xdata>192&&data<240)//中 4
{
    //Buzzer_Freq_Set0( piano[4] );
    //Delay(50);
    //Buzzer_Stop1();
    Beep(piano1[4],50);
    Uart_Printf( "Piano4:%d !\n\n",piano[4] );
}

else if(xdata>240&&data<288)//中 5
{
    //Buzzer_Freq_Set0( piano[5] );
    //Delay(50);
    //Buzzer_Stop1();
    Beep(piano1[5],50);
    Uart_Printf( "Piano5:  %d !\n\n",piano[5] );;
}

else if(xdata>288&&data<336)//中 6
{
    //Buzzer_Freq_Set0( piano[6] );
    //Delay(50);
    //Buzzer_Stop1();
    Beep(piano1[6],50);
    Uart_Printf( "Piano6:  %d !\n\n",piano[6] );
}

else if(xdata>336&&data<384)//中 7
{
    //Buzzer_Freq_Set0( piano[7] );
    //Delay(50);

```

```

        //Buzzer_Stop1();

        Beep(piano1[7],50);

        Uart_Printf( "Piano7:%d !\n\n",piano[7] );
    }
else if(xdata>384&&xddata<432)//高 1
{
    //Buzzer_Freq_Set0( piano[8] );

    //Delay(50);

    //Buzzer_Stop1();

    Beep(piano1[8],50);

    Uart_Printf( "Piano8:%d !\n\n",piano[8] );
}
else if(xdata>432&&xddata<480)//高 2
{
    //Buzzer_Freq_Set0( piano[9] );

    //Delay(50);

    //Buzzer_Stop1();

    Beep(piano1[9],50);

    Uart_Printf( "Piano9:%d !\n\n",piano[9] );
}

xddata=0;

ydata=0;
}

if(xdata>0&&xddata<40&&ydata>0&&ydata<40){

    Paint_Bmp(0, 0, 480,272, background);

    Buzzer_Stop();

    return;

}

}

```

```
}
```

6.电子书功能函数

```
void book1(void){
    int a=0;
    Lcd_ClearScr(0xffffffff);
    while(1){
        if(a==0){
            //Lcd_ClearScr(0xffffffff);
            Lcd_printf(223,0,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"橘颂");
            Lcd_printf(160,20,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"后皇嘉树，橘徕服兮。");
            Lcd_printf(160,40,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"受命不迁，生南国兮。");
            Lcd_printf(160,60,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"深固难徙，更壹志兮。");
            Lcd_printf(160,80,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"绿叶素荣，纷其可喜兮。");
            Lcd_printf(160,100,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"曾枝剌棘，圆果抟兮。");
            Lcd_printf(160,120,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"青黄杂糅，文章烂兮。");
            Lcd_printf(160,140,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"精色内白，类任道兮。");
            Lcd_printf(160,160,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"纷缊宜脩，嫋而不丑兮。");
            Lcd_printf(160,180,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"嗟尔幼志，有以异兮。");
            Lcd_printf(160,200,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"独立不迁，岂不可喜兮。");
```



```

Lcd_printf(160,220,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"深固难徙，廓其无求兮。
");

Lcd_printf(160,240,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"苏世独立，横而不流兮。
");

    while(xdata>440&&ydata>232){

        a=1;

    }

}

if(a==1){
//Lcd_ClearScr(0xfffff);

Lcd_printf(223,0,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"          ");
Lcd_printf(160,20,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"闭心自慎，终不失过兮。
");

Lcd_printf(160,40,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"秉德无私，参天地兮。
");

Lcd_printf(160,60,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"愿岁并谢，与长友兮。
");

Lcd_printf(160,80,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"淑离不淫，梗其有理兮。
");

Lcd_printf(160,100,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"年岁虽少，可师长兮。
");

Lcd_printf(160,120,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"行比伯夷，置以为像兮。
");

Lcd_printf(160,140,0x0000,(0x1f<<11)      |      (0x3f<<5)      |      (0x1f),1,"
");

Lcd_printf(160,160,0x0000,(0x1f<<11)      |      (0x3f<<5)      |      (0x1f),1,"
");

Lcd_printf(160,180,0x0000,(0x1f<<11)      |      (0x3f<<5)      |      (0x1f),1,"

```

```
");
    Lcd_printf(160,200,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"
");
    Lcd_printf(160,220,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"
");
    Lcd_printf(160,240,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"
");

while((xdata>440&&ydata>232)||xdata<40&&ydata>232)){
    if(xdata>440&&ydata>232){
        a=2;
    }
    if(xdata<40&&ydata>232){
        a=0;
    }
}

if(a==2){
    //Lcd_ClearScr(0xffffffff);
    Lcd_printf(223,0,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1," 离 骚
");
    Lcd_printf(160,20,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"帝高阳之苗裔兮，朕皇考
曰伯庸。");
    Lcd_printf(160,40,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"摄提贞于孟陬兮，惟庚寅
吾以降。");
    Lcd_printf(160,60,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"皇览揆余初度兮，肇锡余
以嘉名：");
    Lcd_printf(160,80,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"名余曰正则兮，字余曰灵
均。");
    Lcd_printf(160,100,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"纷吾既有此内美兮，又
```

重之以修能。");

Lcd_printf(160,120,0x0000,(0x1f<<11)|(0x3f<<5)|(0x1f),1,"扈江离与辟芷兮，纫秋兰以为佩。");

Lcd_printf(160,140,0x0000,(0x1f<<11)|(0x3f<<5)|(0x1f),1,"汨余若将不及兮，恐年岁之不吾与。");

Lcd_printf(160,160,0x0000,(0x1f<<11)|(0x3f<<5)|(0x1f),1,"朝搴阰之木兰兮，夕揽洲之宿莽。");

Lcd_printf(160,180,0x0000,(0x1f<<11)|(0x3f<<5)|(0x1f),1,"日月忽其不淹兮，春与秋其代序。");

Lcd_printf(160,200,0x0000,(0x1f<<11)|(0x3f<<5)|(0x1f),1,"惟草木之零落兮，恐美人之迟暮。");

Lcd_printf(160,220,0x0000,(0x1f<<11)|(0x3f<<5)|(0x1f),1,"不抚壮而弃秽兮，何不改此度？");

Lcd_printf(160,240,0x0000,(0x1f<<11)|(0x3f<<5)|(0x1f),1,"乘骐骥以驰骋兮，来吾道夫先路！");

```
while(xdata<40&&ydata>232){
    a=1;
}

}

if(xdata>0&&xdata<40&&ydata>0&&ydata<40){
    xdata=0;
    ydata=0;
    return;
}
xdata=0;
ydata=0;
}
}
```

```

void book2(void){
Lcd_ClearScr(0xfffff);
while(1){
Lcd_printf(223,0,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"飞鸟集");

    Lcd_printf(20,20,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"If you shed tears when you
miss the sun, you also miss the stars ");

    Lcd_printf(20,40,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"如果你因错过太阳而哭泣,
那么你也将错过群星了。");

    Lcd_printf(20,60,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"What you are you do not
see, what you see is your shadow ");

    Lcd_printf(20,80,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"你看不见自己, 你见到的
只是自己的影子。");

    Lcd_printf(20,100,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"I cannot choose the best.
The best chooses me ");

    Lcd_printf(20,120,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"我不能选择那最好的。是
那最好的选择了我。");

    Lcd_printf(20,140,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"That I exist is a perpetual
surprise which is life ");

    Lcd_printf(20,160,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"我的存在, 是一个永恒的
惊奇, 这就是人生。");

    Lcd_printf(20,180,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"God finds himself by
creating ");

    Lcd_printf(20,200,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"神从创造中找到了自己。");

    Lcd_printf(20,220,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"The stars are not afraid to
appear like fireflies ");

    Lcd_printf(20,240,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"群星毫不畏惧自己会看
似萤火虫。");
}
}

```

```

        if(xdata>0&& xdata<40&&ydata>0&&ydata<40){

            xdata=0;

            ydata=0;

            return;

        }

        xdata=0;

        ydata=0;

    }

}

void book3(void){

int a=0;

Lcd_ClearScr(0xffffffff);

while(1){

if(a==0){

Lcd_printf(223,0,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"将进酒");

        Lcd_printf(100,20,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"君不见黄河之水天上来，
        奔流到海不复回。          ");

        Lcd_printf(100,40,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"君不见高堂明镜悲白发，
        朝如青丝暮成雪。          ");

        Lcd_printf(100,60,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"人生得意须尽欢，莫使金
        樽空对月。          ");

        Lcd_printf(100,80,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"天生我材必有用，千金散
        尽还复来。          ");

        Lcd_printf(100,100,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"烹羊宰牛且为乐，会须
        一饮三百杯。          ");

        Lcd_printf(100,120,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"岑夫子，丹丘生，将进
        酒，杯莫停。          ");

        Lcd_printf(100,140,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"与君歌一曲，请君为我
        倾耳听。          ");

```

```
Lcd_printf(100,160,0x0000,(0x1f<<11)|(0x3f<<5)|(0x1f),1,"钟鼓饔玉不足贵，但愿  
长醉不愿醒。");
```

```
Lcd_printf(100,180,0x0000,(0x1f<<11)|(0x3f<<5)|(0x1f),1,"古来圣贤皆寂寞，惟有  
饮者留其名。");
```

```
Lcd_printf(100,200,0x0000,(0x1f<<11)|(0x3f<<5)|(0x1f),1,"陈王昔时宴平乐，斗酒  
十千恣欢谑。");
```

```
Lcd_printf(100,220,0x0000,(0x1f<<11)|(0x3f<<5)|(0x1f),1,"主人何为言少钱，径须  
沽取对君酌。");
```

```
Lcd_printf(100,240,0x0000,(0x1f<<11)|(0x3f<<5)|(0x1f),1,"五花马、千金裘，呼儿  
将出换美酒，与尔同销万古愁。");
```

```
while(xdata>440&&ydata>232){
```

```
    a=1;
```

```
}
```

```
}
```

```
if(a==1){
```

```
Lcd_printf(223,0,0x0000,(0x1f<<11)|(0x3f<<5)|(0x1f),1,"行路难");
```

```
Lcd_printf(100,20,0x0000,(0x1f<<11)|(0x3f<<5)|(0x1f),1,"金樽清酒斗十千，玉盘珍  
羞直万钱。");
```

```
Lcd_printf(100,40,0x0000,(0x1f<<11)|(0x3f<<5)|(0x1f),1,"停杯投箸不能食，拔剑四  
顾心茫然。");
```

```
Lcd_printf(100,60,0x0000,(0x1f<<11)|(0x3f<<5)|(0x1f),1,"欲渡黄河冰塞川，将登太  
行雪满山。");
```

```
Lcd_printf(100,80,0x0000,(0x1f<<11)|(0x3f<<5)|(0x1f),1,"闲来垂钓碧溪上，忽复乘  
舟梦日边。");
```

```
Lcd_printf(100,100,0x0000,(0x1f<<11)|(0x3f<<5)|(0x1f),1,"行路难，行路难，多歧  
路，今安在？");
```

```
Lcd_printf(100,120,0x0000,(0x1f<<11)|(0x3f<<5)|(0x1f),1,"长风破浪会有时，直挂  
云帆济沧海。");
```

```

        Lcd_printf(100,140,0x0000,(0x1f<<11)      |      (0x3f<<5)      |      (0x1f),1,"
    ");
        Lcd_printf(100,160,0x0000,(0x1f<<11)      |      (0x3f<<5)      |      (0x1f),1,"
    ");
        Lcd_printf(100,180,0x0000,(0x1f<<11)      |      (0x3f<<5)      |      (0x1f),1,"
    ");
        Lcd_printf(100,200,0x0000,(0x1f<<11)      |      (0x3f<<5)      |      (0x1f),1,"
    ");
        Lcd_printf(100,220,0x0000,(0x1f<<11)      |      (0x3f<<5)      |      (0x1f),1,"
    ");
        Lcd_printf(100,240,0x0000,(0x1f<<11)      |      (0x3f<<5)      |      (0x1f),1,"
    ");

        while(xdata<40&&ydata>232){
            a=0;
        }

    }

    if(xdata>0&&xdata<40&&ydata>0&&ydata<40){
        xdata=0;
        ydata=0;

        return;
    }

    xdata=0;
    ydata=0;
}

}

void dianzishu(void){
    Lcd_ClearScr(0xffffffff);
    while(1){

```

```

Glib_Rectangle(30,36,150,236,0);

Lcd_printf(73,136,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"楚辞");
Glib_Rectangle(180,36,300,236,0);
Lcd_printf(215,110,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"飞鸟集");
Glib_Rectangle(330,36,450,236,0);
Lcd_printf(378,136,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"李白诗集");


if(xdata>30&&xdxdata<150&&ydata>36&&ydata<236){

    book1();

    Lcd_ClearScr(0xffffffff);

}

if(xdata>180&&xdxdata<300&&ydata>36&&ydata<236){

    book2();

    Lcd_ClearScr(0xffffffff);

}

if(xdata>330&&xdxdata<450&&ydata>36&&ydata<236){

    book3();

    Lcd_ClearScr(0xffffffff);

}

if(xdata>0&&xdxdata<40&&ydata>0&&ydata<40){

    Paint_Bmp(0, 0, 480,272, background);

    return;

}

}

```



```
}
```

7.手电筒功能函数

```
U8 Key_Scan(void)
```

```
{
```

```
    Delay(80);
```

```
    if ((rGPFDAT & (1 << 0)) == 0)
```

```
    {
```

```
        rGPBDAT = rGPBDAT & ~(LED4);           //亮 LED4
```

```
        return 4;
```

```
    }
```

```
    else if ((rGPFDAT & (1 << 2)) == 0)
```

```
    {
```

```
        rGPBDAT = rGPBDAT & ~(LED3);           //亮 LED3
```

```
        return 3;
```

```
    }
```

```
    else if ((rGPFDAT & (1 << 4)) == 0)
```

```
    {
```

```
        rGPBDAT = rGPBDAT & ~(LED2);           //亮 LED2
```

```
        return 2;
```

```
    }
```

```
    else if ((rGPFDAT & (1 << 1)) == 0)
```

```
    {
```

```
        rGPBDAT = rGPBDAT & ~(LED1);           //亮 LED1
```

```
        return 1;
```

```
    }
```

```

else
{
    rGPBDAT = rGPBDAT & ~0x1e0 | 0x1e0;    //LED[8:5] => 1;

    return 0xff;
}

}

static void __irq Key_ISR(void)
{
    U8 key;
    U32 r;

    EnterCritical(&r);
    if(rINTPND==BIT_EINT1)
    {
        //Uart_Printf("eint1\n");
        ClearPending(BIT_EINT1);
    }
    if(rINTPND==BIT_EINT4_7)
    {
        ClearPending(BIT_EINT4_7);
        if(rEINTPEND&(1<<4))
        {
            //Uart_Printf("eint4\n");
            rEINTPEND |= 1<<4;
        }
    }
    if(rINTPND==BIT_EINT0)

```

```

{
    //Uart_Printf("eint0\n");
    ClearPending(BIT_EINT0);
}

if(rINTPND==BIT_EINT2)
{
    //Uart_Printf("eint2\n");
    ClearPending(BIT_EINT2);
}

key=Key_Scan();
if (key == 0xff) {
    ExitCritical(&r);
    lighth();
}
}

void light(void) {
    int temp = 0xffff;
    Lcd_ClearScr(0xffffffff);
    rGPBUP = 0xffff;          // LED [8:5] => PU En
    rGPBCON = 0x15400;        //LED[8:5] => OUTPUT;
    rGPBDAT = temp;
    while (1) {
        temp = temp - 0x20;
        rGPBDAT = temp;
        Delay(1000);
        if (xdata > 0 && xdata < 40 && ydata>0 && ydata < 40) {
            //DisableIrq(BIT_EINT0|BIT_EINT1|BIT_EINT2|BIT_EINT4_7);

```

```

        Paint_Bmp(0, 0, 480, 272, background);

        return;
    }
}

}

```

8. 闹钟功能函数

```

void clock(void){
    unsigned int Addr=0;

    unsigned char Temp1,Temp2;

    U16 year ;

    U8 month, day ,week;        // week

    U8 hour, minute, second ;

    U32 freq;

    float freqk=0.03;

    U16 a=20;

    rRTCCON = 1 ;        //RTC read and write enable


    rBCDYEAR = 22 ;        //年
    rBCDMON  = 12 ;        //月
    rBCDDATE = 1 ;        //日
    rBCDDAY  = 4 ;        //星期
    rBCDHOUR = 19 ;        //小时
    rBCDMIN  = 40 ;        //分
    //rBCDSEC  = 25 ;        //秒


    rRTCCON &= ~1 ;        //RTC read and write disable


    rRTCCON = 1 ;        //RTC read and write enable
}

```

```

year = 2000+rBCDYEAR ;    //年
month = rBCDMON ;    //月
day = rBCDDATE ;    //日
week = rBCDDAY ;    //星期
hour = rBCDHOUR ;    //小时
minute = rBCDMIN ;    //分
second = rBCDSEC ;    //秒

rRTCCON &= ~1 ;    //RTC read and write disable

Lcd_ClearScr(0xffffffff);

Lcd_printf( 50,65,0x0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"时 间： %4d-%02d-%02d   星
期： %d    %02d:%02d\n ", year, month, day,week,hour, minute);

while( 1)
{

    Lcd_printf( 100,100,0xff0000,(0x1f<<11) | (0x3f<<5) | (0x1f),1,"剩余时间:%d", a);

    if(a!=0){

        a-=1;

        Delay( 1000);

    }

    if(a==0){

        Temp1=SONG[Addr++];

        Temp2=SONG[Addr++];

        freq=25000/Temp1;

        //b=1;

        Buzzer_Freq_Set0( freq,freqk);
    }
}

```

```

        Delay(12*Temp2);

        if(Addr>215){

            Addr=0;

        }

    }

    if(xdata>0&&xdata<40&&ydata>0&&ydata<40){

        Paint_Bmp(0, 0, 480,272, background);

        Buzzer_Stop();

            return;

        }

    //经典返回语句

}

}

```

9. 功能选择函数

```

void function(void){

    int temp=1;

    while(1){

        Paint_Bmp(0, 0, 480,272, background);

        if(xdata>96&&xdata<156&&ydata>5&&ydata<76){

            picture();

        }

        if(xdata>96&&xdata<156&&ydata>106&&ydata<170)

        {

            calculator();

        }

        else if(xdata>200&&xdata<260&&ydata>106&&ydata<170)

        {

            music();

        }

    }

}

```

```
else if(xdata>200&&xdata<260&&ydata>5&&ydata<76)
{
    play_piano();
}
else if(xdata>295&&xdata<360&&ydata>106&&ydata<170)
{
    dianzishu();
}
else if(xdata>96&&xdata<156&&ydata>200&&ydata<264)
{
    light();
}
else if(xdata>295&&xdata<360&&ydata>5&&ydata<76){
    clock();
}
else if(xdata>295&&xdata<360&&ydata>200&&ydata<264){
    return;
}
}
```

10. 主函数循环部分

```
while(1){
    password();
    Paint_Bmp(0, 0, 480,272, background);
    function();
}
```