

课程设计任务书
(指导教师填写)

课程设计名称 专业综合课程设计 学生姓名 甄保珠 专业班级 电信 103

设计题目 基于 ARM9-S3C2440的数字相框的设计

一、课程设计目的

1. 掌握 ARM的 S3C2440微处理器的工作原理。
2. 熟悉常用显示器件的工作原理。
3. 掌握 ARM编程和调试方法。

二、设计内容、技术条件和要求

1. 设计内容：基于 ARM9-S3C2440平台在 TFTLCD屏上显示汉字及图片
2. 技术条件：
 - 1) XYD2440开发平台
 - 2) J-LINK 仿真器
 - 3) USB转 RS232转换器
 - 4) GSM&GPRS模块
 - 5) 配套配件
3. 要求：
 - 1) 以 S3C2440芯片为核心，结合 TFTLCD原理，完成一个简单的接口电路设计。给出系统的组成方案，利用 PROTE完成原理图的设计。
 - 2) 完成工作流程和软件的设计。
 - 1) 显示自己的名字，名字分别用红黄蓝三种颜色、并满屏滚动。
 - 2) 选择 5 幅图片,使其轮流显示；
 - 3) 分别实现图像的横屏及竖屏显示.

三、时间进度安排.

- 1 嵌入式 ARM基础及模块实训阶段 1 天
 - 1) KEIL新建工程、工具、开发平台使用；
 - 2) GPIO实验讲解和五种流水灯程序实验；
2. 资料查阅与学习、讨论 2 天
 - 1) CPU S3C2440用户手册
 - 2) Norflash_SST39VF1601 用户手册
 - 3) LCD_W35_ZQ3506_LQ035Q1D006手册
3. 电路设计 2 天
 - 1) 最小系统设计 1 天
 - 2) 接口电路设计 1 天

3 天

- | | |
|-------------|-----|
| 5. 调试, 成果验收 | 2 天 |
|-------------|-----|

1、S3C2440用户手册.

- 指导教师签字：_____年 月 日

基于 ARM9-S3C2440的数字相框的设计

摘要

本设计使用 ARM9的 S3C2440芯片，在对该芯片的工作原理及其外围电路有一定了解的基础上，实现在 TFT LCD屏上显示自己的汉字名字并使其满屏滚动的功能；和制作数字相框，实现自动播放 5 张 320*240 图片功能，查看上一张、下一张照片的功能，以及横屏竖屏转换看图功能。用到的知识点有 LCD控制器、模数转换器及触摸屏接口、中断控制器等内容。数码相框是展示数码照片而非纸质照片的相框。众多的数码摄影产生的相片保存起来后，查看过程繁琐不方便，大量打印出来又非常浪费，如果用数码相框直接插上相机的存储卡展示照片，就非常酷了。数码摄影必然推动数码相框的发展。根据目前国内外对嵌入式的研究和开发，结合实际的实验条件，本项目使用硬件平台 ARM9的上 S3C2440A开发板作为目标机，使用安装 Windows XP 的 PC机作为宿主机，并在宿主机安装 Keil+MDK补丁，及 J-Flash 软件作为开发环境。主要实现工作包括：在宿主机上安装交叉编译工具，建立交叉编译环境，建立嵌入式软件平台。进行一个可以浏览与管理图片功能，且能进行幻灯片播放的课程设计。针对项目需求实现对驱动程序的完善和移植，制作适合此开发板的根文件系统，最后将设计的应用程序程植到开发板上，实现个人图片的掌上管理。

关键字：ARM9 S3C2440 LCD 电阻触摸屏 ADC 数模转换 TFT 驱动

一 硬件部分介绍

一. ARM9的 S3C2440微处理器简介

Micro2440 开发板由核心板 Micro2440 和底板 Micro2440SDK组成，简称为 Micro2440 开发板。Micro2440 核心板其实是一个最小系统板，它具有最基本的系统配置：

◆ CPU - 三星 S3C2440, 运行于 400Mhz

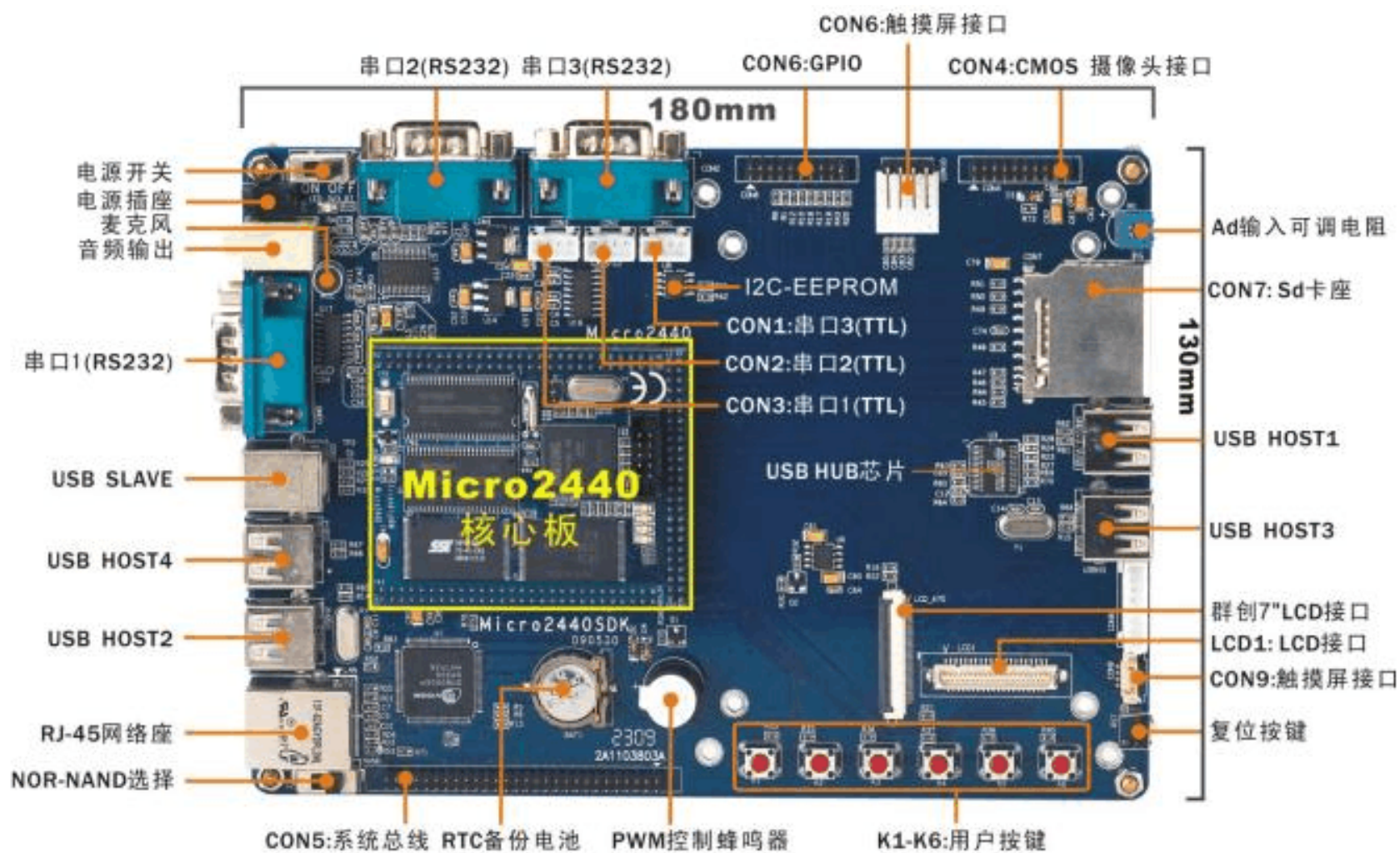


图 1 Micro2440 开发板

◆ NOR FLASH - 2M, 很多公司为了节省成本并不提供 NOR FLASH这对开发和产量是很不利的;

◆ NAND FLASH - 256M(可根据用户需求更改为 64M-1G)

◆ SDRAM - 64M, 由 2 片 16-bit 宽度的 32M SDRAM 组成;

◆ 1 个电源指示等和 4 个用户指示灯;

◆ 专业复位芯片;

◆ 在板 JTAG

◆ 专业电压调节芯片;

电源系统:

本开发板的电源系统比较简单, 直接使用外接的 5V 电源, 通过降压芯片产生整个系统所需要的三种电压: 3.3V、1.25V。如图 2 所示。

复位系统: Micro2440 核心板自带复位电路, 采用专业的复位芯片 MAX811 实现 CPU 所需要的低电平复位, 见下图:

本设计使用的核心芯片为三星公司的 16/32 位精简指令集 (RISC) 微处理器 S3C2440A 三星公司的 S3C2440A 为手持设备和普通应用提供了低功耗和高性能的小型芯片微控制器的解决方案。为了降低整体系统成本, S3C2440A 还提供了以下丰富的内部设备。

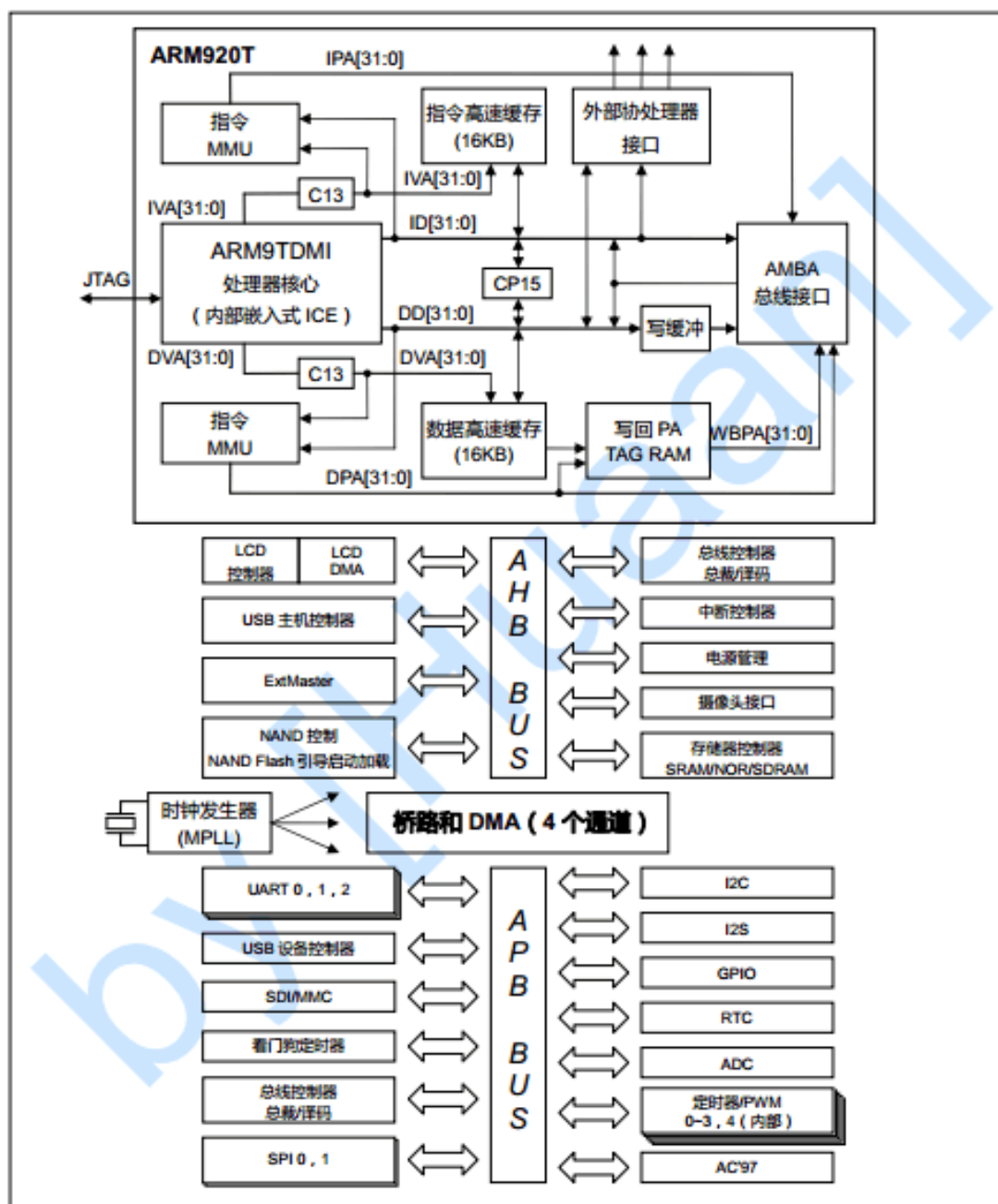
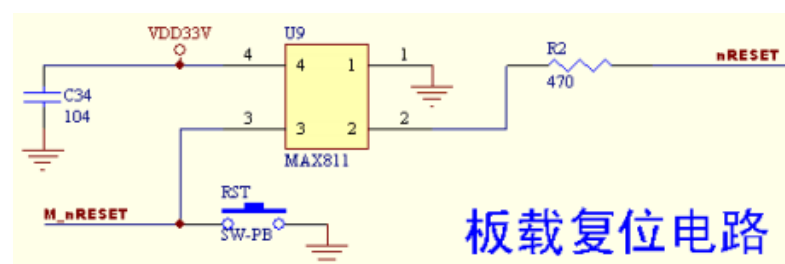
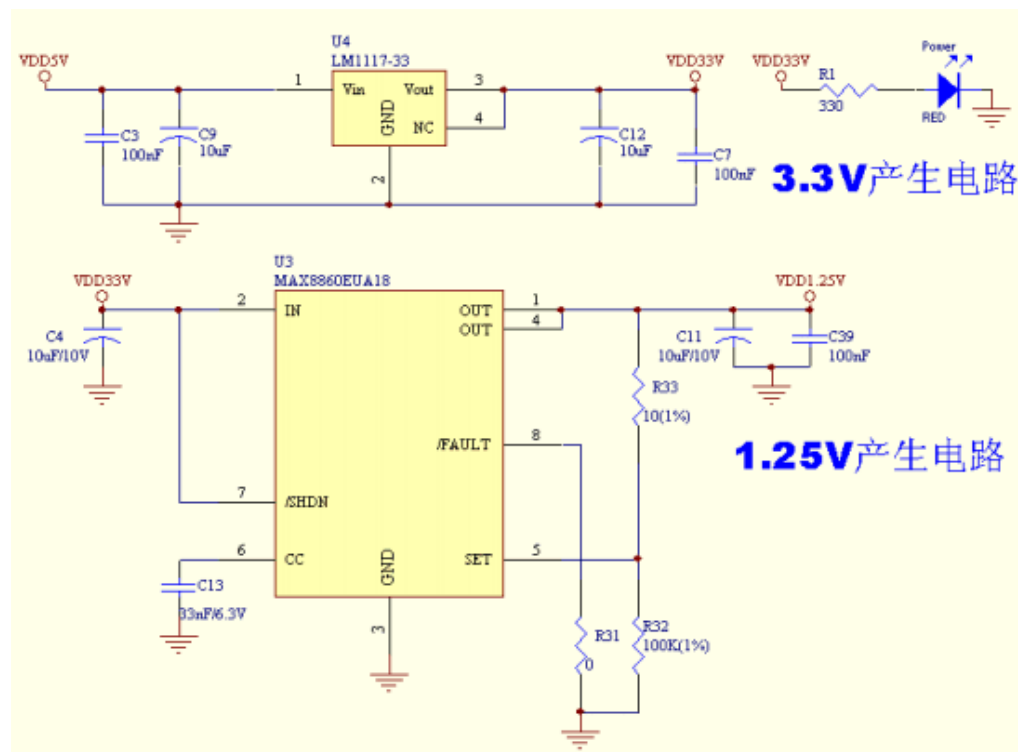
用户 LED:

	LED1	LED2	LED3	LED4
GPIO	GPB5	GPB6	GPB7	GPB8
可复用为	nXBACK	nXREQ	nXDACK1	nDREQ1
在原理图中的网络名	nLED_1	nLED_2	nLED_3	nLED_4

LED 是开发中最常用的状态指示设备, Micro2440 具有 4 个用户可编程 LED 它们直接与 CPU 的 GPIO 相连接, 低电平有效 (点亮), 详细的资源占用如上表:

S3C2440A 基于 ARM920T 核心, 0.13 μ m 的 CMOS 标准宏单元和存储器单元。低功耗, 简单, 精致, 且全静态设计特别适合于对成本和功率敏感型的应用。它采用了新的总线架构如先进微控制总线构架 (AMBA

S3C2440A 的突出特点是其处理器核心, 是一个由 Advanced RISC Machines (ARM 公司设计的 16/32 位 ARM920T 的 RISC 处理器。ARM920T 实现了 MMU、AMBA 总线和哈佛结构高速缓冲体系结构。这一结构具有独立的



16KB指令高速缓存和 16KB数据高速缓存。每个都是由具有 8 字长的行(line) 组成。

通过提供一套完整的通用系统外设，S3C2440A 减少整体系统成本和无需配置额外
图 2 S3C2440A 方框图

外的组件。S3C2440A集成了以下片上功能：

● 1.2V 内核供电, 1.8V/2.5V/3.3V 存储器供电, 3.3V 外部 I/O 供电，具备 16KB的指令缓存和 16KB的数据缓存和 MMU微处理器

- 外部存储控制器（SDRAM控制和片选逻辑）
- LCD控制器（最大支持 4K色 STN和 256K色 TFT）提供 1 通道 LCD专用 DMA
- 4 通道 DMA并有外部请求引脚
- 3 通道 UART(IrDA1.0, 64 字节发送 FIFO和 64 字节接收 FIFO)
- 2 通道 SPI
- 1 通道 IIC 总线接口（支持多主机）
- 1 通道 IIS 总线音频编码器接口
- AC' 97 编解码器接口
- 兼容 SD主接口协议 1.0 版和 MMC协议 2.11 兼容版
- 2 通道 USB主机/1 通道 USB设备（1.1 版）
- 4 通道 PWM定时器和 1 通道内部定时器/看门狗定时器
- 8 通道 10 位 ADC和触摸屏接口
- 具有日历功能的 RTC
- 摄像头接口（最大支持 4096×4096 像素输入；2048×2048 像素输入支持缩放）
- 130 个通用 I/O 口和 24 通道外部中断源
- 具有普通，慢速，空闲和掉电模式
- 具有 PLL片上时钟发生器

二. LCD屏基本知识

LCD屏的像素位：黑白屏：1bit 像素位；灰度屏：2,4bit 像素位；假(伪)彩：8bit 像素位，借助调色板显示。黑白屏->灰度屏->假彩->真彩。

真彩：16bit 像素位，65K色（65536 色），RGB(5:6:5)，0xF800-全红，0x07e0- 全绿，0x001f-全蓝，0x0000- 黑色，0xffff- 白色；

24bit 像素位，RGB(8:8:8)，0xff0000- 全红，0x00ff00- 全绿，0x0000ff- 全蓝；

32bit 像素位，RGB(8:8:8+阿尔法通道)。

显存：用一块内存存储当前屏幕的像素色彩。

三. TFT屏构造

开发板屏从上向下： 4 线电阻触摸屏、LCD屏+驱动 IC、PCB接口。

四. TFT屏驱动接口信号

1. 视频数据线 VD0-VD23 直接从内存提取数据；
2. 像素时钟线 VCLK 作用：切换像素与像素的时钟线，换下一个像素；
3. 行同步时钟 HSYNC 下一行扫描开始；
4. 场同步时钟 VSYNC 下一个画面开始；
5. 数据使能 VDEN

五. 2440TFT驱动器输出时序图分析及相关参数

- 1.VSPW 场同步时钟宽度，是行同步的
- 2.VBPD 场后无效行个数
- 3.VBFD 场前无效行个数
- 4.HSPW 行同步始终宽度--是像素时钟 VCLK的倍数

- 5.HBPD 行后无效像素个数
- 6.HFPD 行前无效像素个数
- 7.HOZVAL 一行点数，W35是（320-1）
- 8.LINEAL 有多少行，W35是（240-1）

以上八个参数都需要查找规格书填到相应寄存器。

六. 相关寄存器的设置

1. 控制寄存器 LCDCON1

[17_8]: 像素时钟分频值 CLKVAL VCLK看规格书: 6.4MHZ HCLK CPU内部时钟, 启动代码设为 100MHZ

[7]: =0

[6_5]: 11

[4_1]: 1101

[0]: 0-->1

2. 控制寄存器 2 LCDCON2

设: 场后 CBPD 行数 LINEVAL 场前 VFPD 场宽 VSPW

//以上也是从规格书中看, 填到相应寄存器中。

3. 控制寄存器 3 LCDCON3 设置: HBPD、HOZVALHFPD

4. 控制寄存器 4 LCDCON4 [7_0]:HSPW

5. 控制寄存器 5 LCDCON5 设置时钟极性, 字节交换; 某些信号线是否使能 (背光控制)

6. 帧缓冲器开始地址寄存器 1 (告诉显存地址在哪里)

[29_21]: 显存地址的[30_22]

[20_0]: 显存地址的[21_1]

申请显存, 定义一个二维数组

U16 LCDBUFFER[行][列]

7. 帧缓冲器地址寄存器 2 (告诉显存的结束地址)

[20_0]: 显存结束地址的[21_1]

8. 帧缓冲器地址寄存器 3

[21_11]:OFFSIZE, 当显存大于 LCD屏时, 每行无效的字节数/2

[10_0]: 有效宽度的字节数/2

七. 汉字取模软件使用

打开软件“字模提取 V2.1 CopyLeft By Horse2000”, 在文字输入区输入要取模的汉字, 按下“Ctrl+Enter”。参数设置→其他选项: 勾选横向取模, 去掉字节倒序。取模方式选择 C51 格式。模拟动画→放大格点。

汉字取模宽度*高度: 16*16 (默认), 14*14, 12*12。

ASCII 字符: A—Z, 0—9, 8*16 点阵。

例如:

/*-- 文字: 甄 --*/

/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --*/

0x00,0x00,0x08,0x00,0x00,0x20,0xFE,0xFC,0x0B,0xF8,0x01,0x20,0x28,0x40,0x0A,0x08,
0xFD,0x20,0x28,0x40,0x12,0x08,0x11,0xFC,0xFE,0x40,0x12,0x08,0x11,0x20,0xAA,0x78,

/*-- 文字: 保 --*/

/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --*/

0x33,0xF8,0x12,0x20,0xAA,0x48,0x30,0x40,0x7C,0x20,0xFE,0x48,0x50,0x40,0x13,0xFE,
0x10,0xA8,0x97,0xFC,0x10,0x70,0x10,0x98,0x10,0xE0,0x10,0xA8,0xFE,0x88,0x11,0x50,

```

/*-- 文字： 珠 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/
0x10,0xA8,0x10,0x88,0x12,0x48,0x1D,0x24,0x10,0xAA,0x14,0x44,0xE1,0x24,0x1E,0xCA,
0x18,0x42,0x42,0x22,0xF0,0x8A,0x10,0x40,0x00,0x20,0x40,0x06,0x10,0x40,0x00,0x20,

```

八. 字库的使用原理

ZI.C 里汉字模数组排列顺序按 GB2312

do- 汉字的区码， c5- 汉字的位码。

Pos = 94*(chs-161)+((chs+1)-161);// 汉字在字库中的首个字母

九. TFT屏显示图片

图片取模软件的使用：使用 “Image2Lcd v3.2 ” 软件。

- 1、打开要取模的图片，按如下设置。
- 2、输出数据类型： C 语言数组 (*.c)；
- 3、扫描方式： 水平扫描；
- 4、输出灰度： 16 位真彩色；
- 5、最大宽度和高度： 图片实际大小；
- 6、去掉 “包含图像头数据”， 勾选高位在前。
- 7、输出图像调整选择 16 位彩色， 颜色位数选择 R: 5bits,G:6bits,B:5bits 。
- 8、保存在 “ARM工程模板” 下的 “Source ” 文件夹下， 以 “*.c ” 命名。

检查字节数是否符合方法：（以 320*240 图片为例）若结果符合 320（weight）*240（height）*2（字节），则说明图片取模正确。图片有失真情况，一般都是模取错了。应该先改参数，在选择图片。

二 建立裸机工程

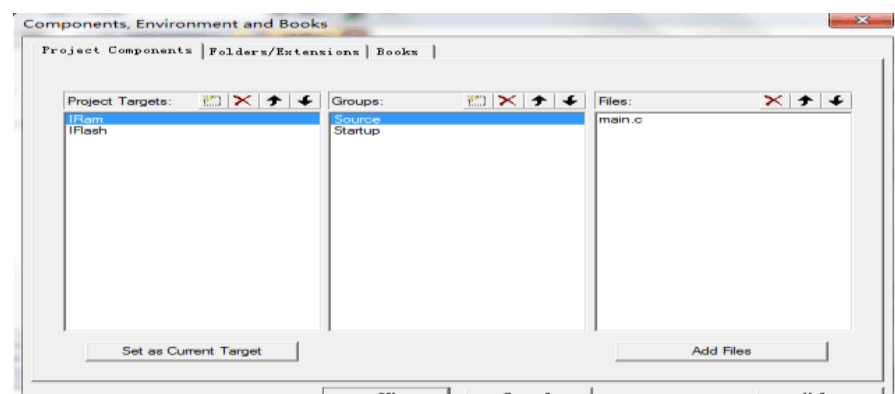
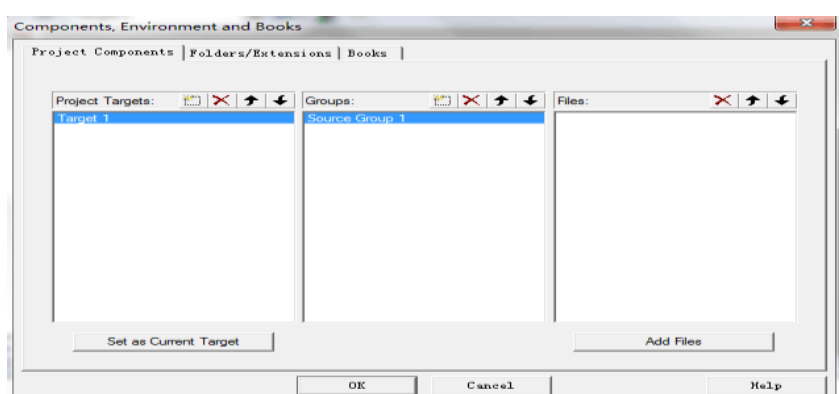
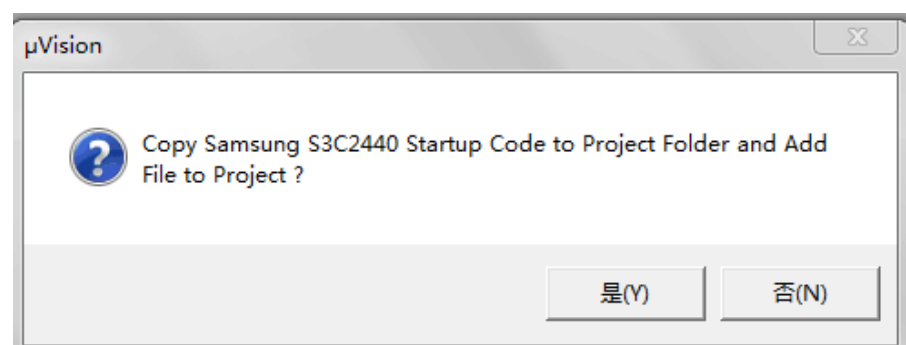
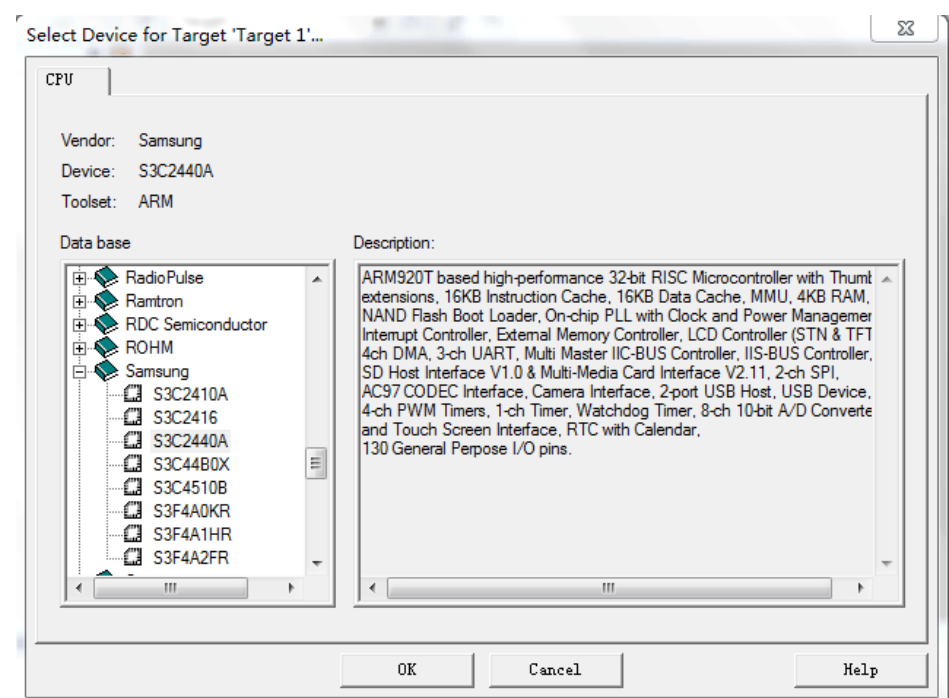
（1）、新建文件夹，命名为 “ARM工程模板”，打开裸机例程，复制其中的 Debug 和 include 文件到 ARM工程模板。Debug 里是在内存调试时的初始化脚本；include 里放的是头文件，可以是自己创建的和系统提供的。

（2）新建 List 和 Output 文件夹。List 存编译时输出*.list 中间文件；Output 中存放烧录的*.hex 文件和调试文件 axf 。

（3）新建 Source 文件夹，存放源代码 “*.c ”、“*.cpp ” 文件。

（4）直接复制 Startup 文件夹，里面放的是汇编启动代码。打开 Keil μ Vision4 软件，点击 Project →Newμ Vision Project ☐, 找到目标路径，文件名写为 “TFT” 并保存。在弹出的对话框中选择 Sumsung 下的 S3C2440A芯片；点击 “OK”。在弹出的如右图对话框所示，选择 “否” 即可；

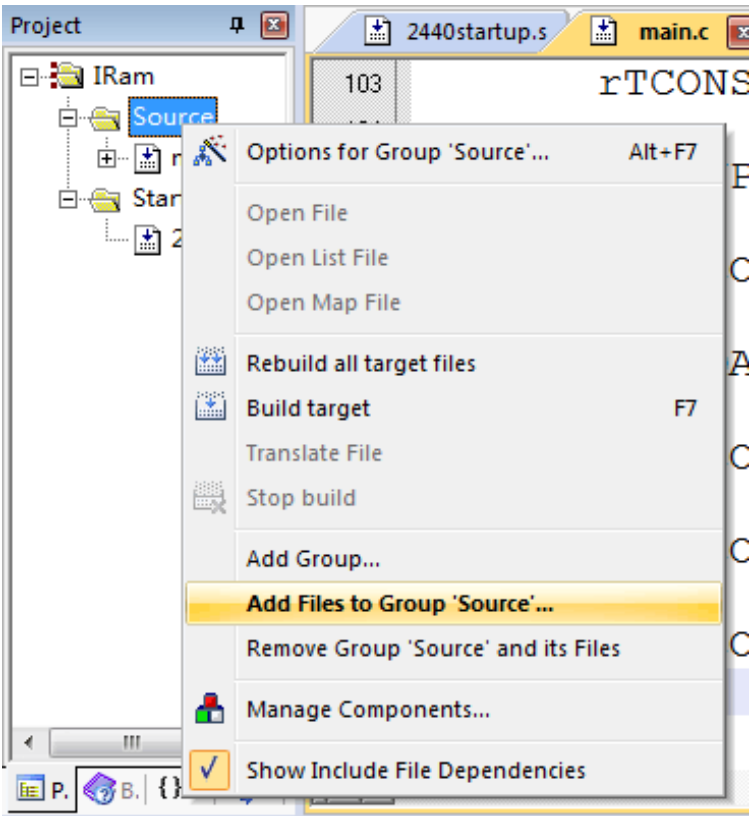
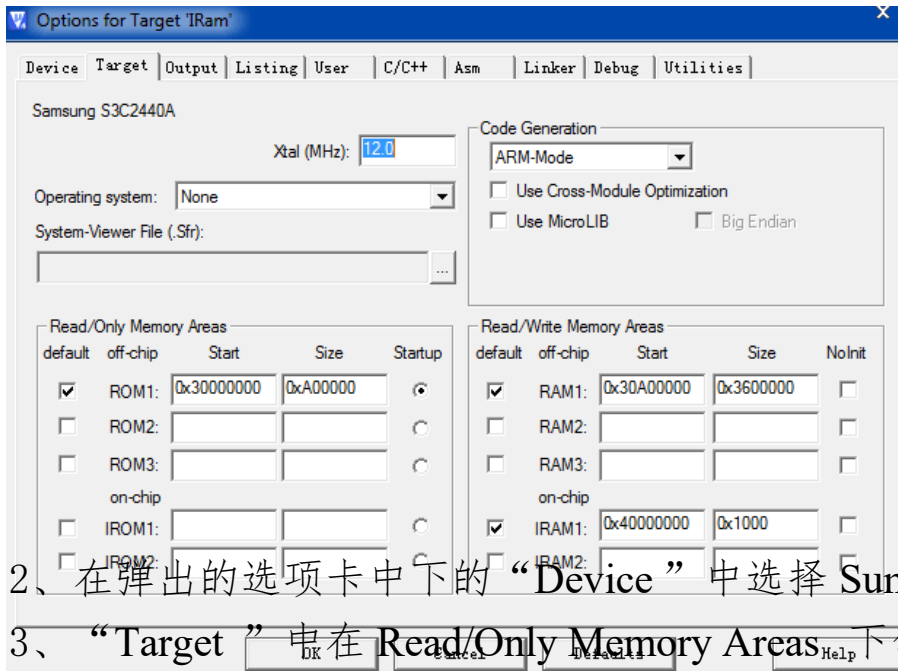
（5）代码分组，设置 IFlash 和 IRam的菜单选项。点击  按钮，将弹出如左图所示对话框。按右图设置该对话框。



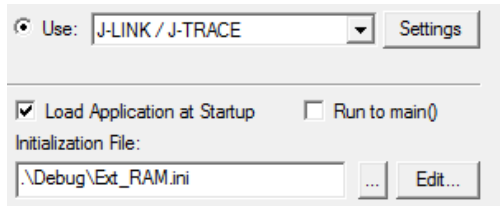
(6) 书写源码并添加到工程中。方法如右图所示。

(7) IRam下的工程设置。

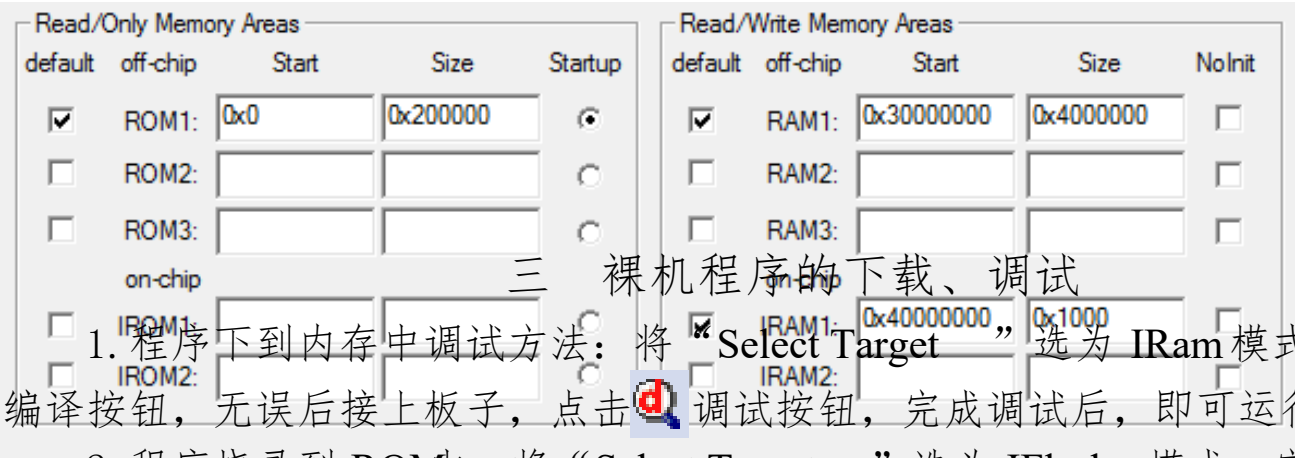
1、将“Select Target”选为 IRam模式，点击按钮，弹出如下对话框：



- 2、在弹出的选项卡中下的“Device”中选择 Samsung 下的 S3C2440A
- 3、“Target”中在 Read/Only Memory Areas 下勾选 ROM 的 default 项，填写 Start 和 Size，选上 Startup；在 Read/Write Memory Areas 下勾选 RAM 的 default 项，填写 Start 和 Size；
- 4、“Output”中点击“Select Folder for Objects...”将文件定位到 Output 文件夹下；“Name of Executable:”后填写 TFT_IRam并勾选“Create HEX File”；
- 5、“Listing”中点击“Select Folder for Listings...”将文件定位到 List 文件夹下；
- 6、“C/C++”下将“Include Paths”设定为“”；
- 7、“Linker”下保证勾选左上角的“Use MemoryLayout from Target Dialog”；
- 8、“Debug”下按照右图设置并添加 Debug 文件夹下的*.ini 文件；
- 9、“Utilities”下去掉“Update Target before Debugging”前面的对勾，点选“Use Target Driver for Flash Programming”，选择“J-LINK/J-TRACE”，点击“Settings”按钮，添加“AM29F160DB Flash”并点击“ok”即可。



(8) IFlash 下的工程设置。与 IRam大致相同，将“Select Target”选为 IFlash 模式，“Target”中在 Read/Only Memory Areas 下勾选 ROM 的 default 项，按照下图重新填写 Start 和 Size，选上 Startup；在 Read/Write MemoryAreas 下勾选 RAM 的 default 项，按照下图重新填写 Start 和 Size；“Name of Executable:”后填写 TFT_IFlash；“Debug”下“Initialization File”什么也不选，其它内容的设置同 IRam 下的工程设置。



三 裸机程序的下载、调试

1. 程序下到内存中调试方法：将“Select Target”选为 IRam模式，完成设置后，点击组建、编译按钮，无误后接上板子，点击调试按钮，完成调试后，即可运行程序，查看结果。

2. 程序烧录到 ROM中：将“Select Target”选为 IFlash 模式，完成设置后，这时注意将板子上面的 Nor flash/Nand flash 按钮搬到 Nor flash 处，然后点击组建、编译按钮，无误后接上板子，组建、编译，不要调试，点击下载按钮，程序即被下载到板子上。拔下 J-LINK 下载线，重启板子，即可看到刚下载到板子上的程序的结果。

四 课程设计完成过程及情况

1. 显示自己的名字，名字分别用红黄蓝三种颜色、并满屏滚动。

实现方法：（以名字向左滚动为例进行说明，如图9）。首先在 LCD 屏上绘制汉字名字，用 for 循环实现汉字在屏上的移动，接下来实现名字移动后的背景像素的拖尾擦除。用两层 for 循环将从名字后面位置开始的宽度为 319-48，高度大于汉字高度的区域的像素填充为背景色白色，并嵌套在汉字移动的 for 循环内让该区域跟随汉字一起移动。汉字满屏滚动原理同上所述。

程序流程图如图 10 所示：

完成情况结果如下图所示：

```
//向左滚动
for (j=0;j<319;j++)
{
    //绘制汉字字符，红字黑背景
    Draw_Text16(200-j,100,0xf800,0x0000,xu);
    Draw_Text16(216-j,100,0xf800,0x0000,shuang);
    Draw_Text16(232-j,100,0xf800,0x0000,shuang);
    delay(10000);
    //绘制名字后的白色背景
    for( y = 90 ; y < 120 ; y++ )
    {
        for(k=248-j;k<319+200-48-j;k++)
        {
            LCD_BUFFER[y][k] = 0xFFFF ;
        }
    }
}
```

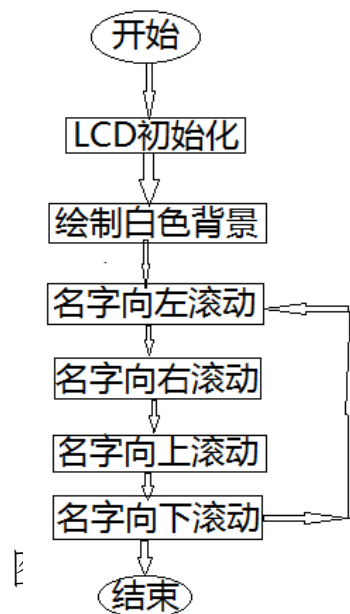
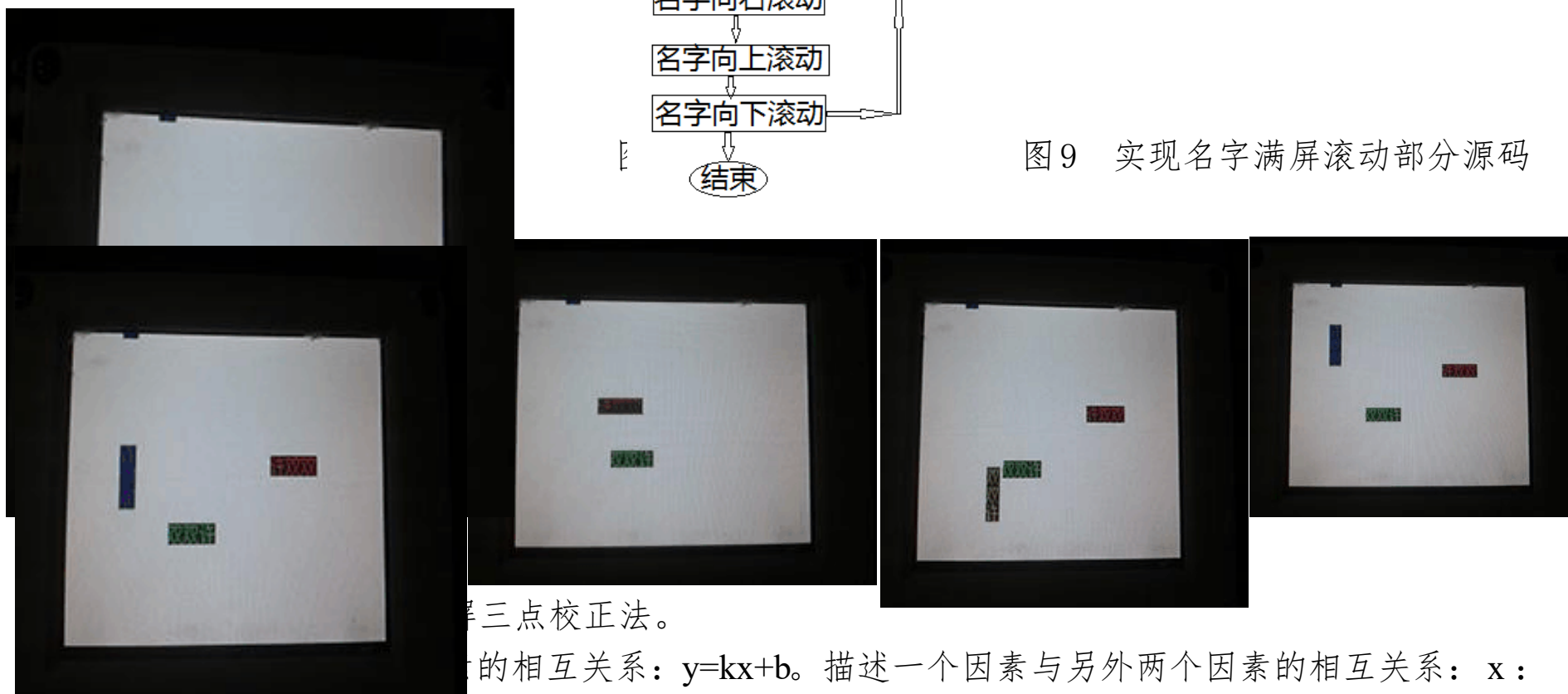


图9 实现名字满屏滚动部分源码



三点校正法。

的相互关系： $y=kx+b$ 。描述一个因素与另外两个因素的相互关系： x ：设为 x 轴实际坐标点，校正时打的十字叉； D,E,F ：分别为三个系数。

$x=D*x_{\text{轴电压}}+E*y_{\text{轴电压}}+F$ ；（三元一次方程，要采集三组样本）

$y=A*x_{\text{轴电压}}+B*y_{\text{轴电压}}+C$ ；

三组样本的位置设定：小.中.大(左上、右中、下中)

程序中出现的符号变量：

$yd[0_1_2]$: 存 y 轴三组十字形坐标样本

$xd[0_1_2]$: 存 x 轴三组十字形坐标样本

$xt[0_1_2]$: 存 x 轴点击后读取的 AD 值

$yt[0_1_2]$: 存 y 轴点击后读取的 AD 值

公式变换，以采集 x 轴为例

$x=D*x_{\text{轴电压}}+E*y_{\text{轴电压}}+F$

(24,32), (289,120), (160,210)

$24=D*86+E*70+F$

$xd[0]=D*xt[0]+E*yt[0]+F$

$290=D*1000+E*500+F$

$xd[1]=D*xt[1]+E*yt[1]+F$

$160=D*600+E*900+F$

$xd[2]=D*xt[2]+E*yt[2]+F$

4 线电阻触摸屏的结构：1、上(x) 下(y)2 层透明分布均匀电阻膜；2、x 电阻层左右引线(X+,X-) y 电阻层上下引线(Y+,Y-) 4 线。



方向为例，

- 1、+ 接 V+，X- 接地，Y+ 接 ADC 输入通道；Y- 断开，高阻态
- 2、用户点击后，X 层的分压由 Y 层正极导出进入 ADC

测量 Y 方向电压，由 X 正极导出，原理类同

电路连接：YP->AD 通道 5 XM->AD 通道 6 XP->AD 通道 7

相关寄存器：

1、ADC 控制寄存器 ADCCON

2、触摸屏控制寄存器 ADCTSC

[8]: 笔尖起落中断选择

[7]: YM 开关控制

0: 断开 1: 接通

[6]: YP 开关控制

0: 接通 1: 断开

[5]: XM 开关控制

0: 断开 1: 接通

[4]: XP 开关控制

0: 接通 1: 断开

[3]: XP 线上拉电阻使能

0: 使能 1: 不使能

[2]: 自动 x, y 测量控制

[1_0]: 测量模式

11

3、数据寄存器 0 ADCDAT0 [9_0]: x 方向结果

4、数据寄存器 1 ADCDAT1 [9_0]: y 方向结果

5、启动延时寄存器 ADCDLY

6、起落中断检测寄存器 ADCUPDN

[1]: 是否出现抬起动作

[0]: 是否出现落下动作

功能实现描述：首先进行屏幕校准，校准完成后开始绘制图片。在屏幕上方绘制时间，当前图片序号/图片

总数。在屏幕下方绘制四个按钮，分别为上一张、自动播放、竖屏、下一张按钮，当点击这四个按钮时，按钮变为红色背景黄色字，自动播放按钮按下后，5 张图片将轮流播放，按钮变为“STOP”，按下“STOP”后，自动播放停止。如果在自动播放模式下按下上一张或下一张按钮，则退出自动播放，并转到上一张或下一张图片。

程序分析：显示指定大小图片的功能函数如下图所示：

完成结果如图所示：



```
***** 函数名: Paint_Bmp()
***** 功 能: 显示指定大小图片
***** 参 数: x0 开始位置, y0 开始位置, l 高度, h 宽度, bmp 图片数据数组
***** 返回值: 无
*****
//l:320 h:240
void Paint_Bmp(int x0,int y0,int h,int l,const unsigned char bmp[])
{
    //0, 8, 320, 240
    int x,y;
    U32 c;
    int p = 0;

    for( y = y0 ; y < 220 ; y++ )
    {
        if(y0+y >= LCD_HEIGHT-22) break;

        for( x = 0 ; x < h ; x++ )
        {
            c = (bmp[p]<<8) | (bmp[p+1]);

            if ( ( (x0+x) < LCD_WIDTH) && ( (y0+y) < LCD_HEIGHT) )
                LCD_BUFFER[y0+y][x0+x] = c ;

            p = p + 2 ;
        }
    }
}
```




3

显示.

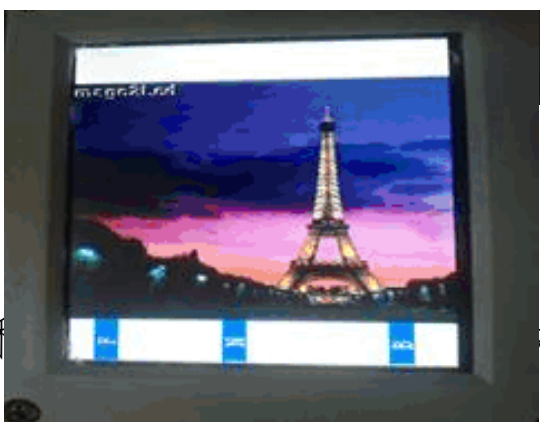
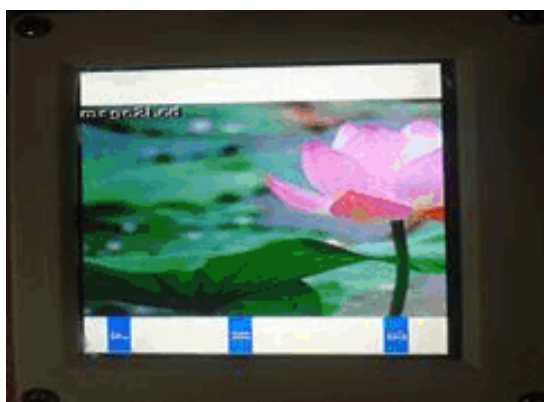
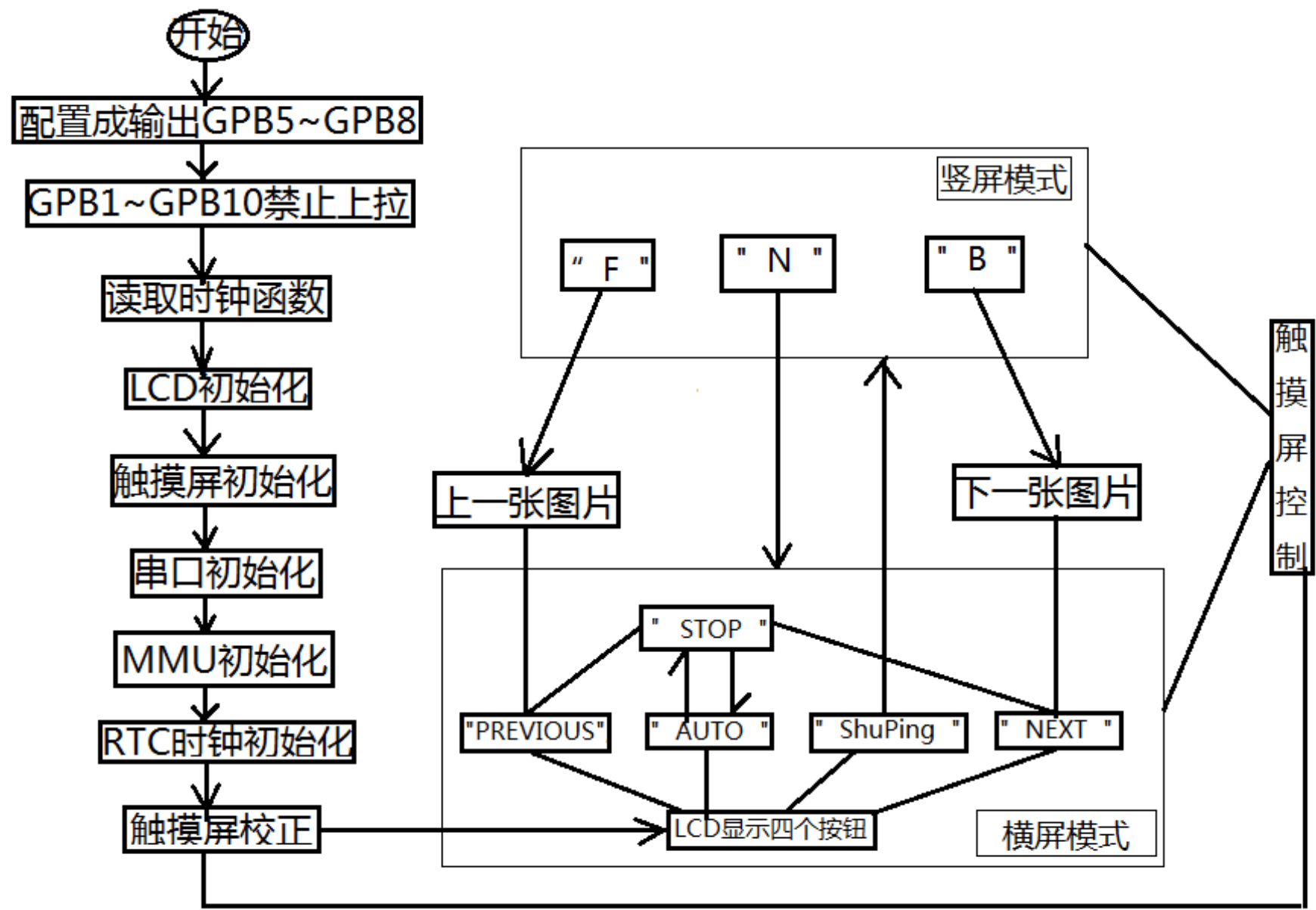
功能实现描述：点击横屏模式屏幕下方的“N”按钮，图片将转变“竖屏”按键，将分别跳转到上一张、下一张图片。

一张图片。

竖屏显示图片功能函数如右图所示。

```
//绘制竖屏图片
void Paint_BmpShuPing(int x0,int y0,int h,int l,const unsigned char bmp[])
{
    //40,0,240,320
    int x,y;
    U32 c;
    int p = 0;
    for( x = 320 ; x >= 40 ; x-- )
    {
        for( y = 0 ; y < 320 ; y++ )
        {
            if(x>=x0&&x<x0+h&&y>=y0&&y<y0+l)
            {
                c = (bmp[p]<<8)|bmp[p+1]; //取出一个点色彩
                LCD_BUFFER[y][x] = c ;
                p = p + 2 ;
            }
            else{
                LCD_BUFFER[y][x] = 0xFFFFFF ; //图片未占据的区域填充为白色背景
            }
        }
    }
}
```

程序流程图如下图所示：



果和 决办法

图片取模后产生的.c 文件名需与 main.c 中定义的图片名一致。

2. 当出现目标没有创建的错误时，一般是由于程序中有错误没有改正。
3. 出现溢出异常时，可以改大 ROM 和 RAM 的 Size 。
4. 可通过单步调试法查看程序出错原因。
5. 避免调用的函数嵌套太深，以致程序执行到该处时出现死机情况。

扩展功能：当图片尺寸小于 320*240 时，使图片居中显示。

```
//绘制白背景的小图片
void Paint_BmpL(int x0,int y0,int h,int l,const unsigned char bmp[])
{
    int x,y;
    U32 c;
    int p = 0;
    for( y = 0 ; y < LCD_HEIGHT ; y++ )
    {
        for( x = 0 ; x < LCD_WIDTH ; x++ )
        {
            if(x>=x0&&x<x0+h&&y>=y0&&y<y0+l){
                c = (bmp[p]<<8)|bmp[p+1];    //取出一个点色彩
                LCD_BUFFER[y][x] = c ;
                p = p + 2 ;
            }
            else{
                LCD_BUFFER[y][x] = 0xFFFFFFFF ;
            }
        }
    }
}
```

通过本次课程设计，我学会了如何建立自己的裸机工程，并对 Keil 软件，J-Link 下载，汉字取模软件，图片取模软件有了一定的了解，懂得了屏幕坐标系与其它坐标系的区别。在老师的耐心细致的讲解下，我学会了阅读芯片手册，看 tft 时序图，并通过完成作业，进一步熟悉了使用 Protel 软件绘制原理图，对 S3C2440A 芯片的最小系统及其工作原理有了一定程度的理解。

这次课程设计给了我很多益处。虽然比起平时只是坐在教室里听老师讲课来说，这个课程设计让我着实感觉劳累，但是它能让我们学到知识，让我感到充实。

1. 核心板认识（S3C2440最小系统）。最小系统：一个处理器能够工作最小硬件条件。

CPU s3c2440A，内部 RAM 4K字节，内置 ROM 0字节，工作频率最高 553MHZ

SDRAM32M字节*2片=64M 程序调试

总线地址空间：0x3000 000 _ （0x3400 00000-1）

1KB=1024字节

1MB=1024K

ROM(Norflash)：2M字节，下载到这里（我们都是 Norflash 下跑的。。。)

总线地址空间：0x0 _ (0x200000-1)

ROM(Nandflash)：K9F2G08,128字节；K9F2G08,256字节。并没有映射到总线空间，近似认为是个外接 U 盘。在 nandflash 启动模式下，上电后只有前 4K 代码会自动运行（一般远远大于 4K），所以，其余部分要用户写程序复制到内存运行。移植 Linux 操作系统后，就选择这种模式。。。

SDRAM 连接：地址线并联，数据线串联，2 片芯片的地址线是并联后与 CPU 连接。

2 种 ROM 有开关选择

- 2、应用扩展板

用户可以根据项目需要删减：

- 1)、TFT LCD接口：3.5、7 寸屏
- 2)、触摸屏：4 线制电阻触摸屏
- 3)、USB HOST USB主机设备接口，读写 U 盘，USB鼠标识别
- 4)、SD卡接口：读写 SD卡
- 5)、ADC接口：模数转换，测电压的接口，充电电流，电压接口
- 6)、CMOS摄像头接口：
- 7)、GPIO 手电筒控制、继电器控制，高低电平 0 和 1（0V和 3.3V）
- 8)、RS232 和电脑通讯，GPS 短信模块，RS485通信
- 9)、双声道立体声输出：MP3 MP4播放
- 10)、录音
- 11)、USB SLAVE USB从机，模拟 U 盘，模拟 USB鼠标，模拟 USB触摸板
- 12)、10/100M自适应网络接口：国际互联网
- 13)、RTC 百年历 2000~2099 年

名字滚屏滚动源程序：

```

/*****
****模块名：main.c
****功 能：LCD显示字符测试程序
****说 明： 本程序是用的屏是 X35 （索尼 3.5 寸屏） 其他屏可能要修改个别参数。
*****/
包含头文件,它在 工程目录/Include/2440addr.h
/*****/
#define U8 unsigned char
#define U16 unsigned short
#define U32 unsigned int
#define M5D(n) ((n) & 0x1ffff) // 用于设置显示缓存区时，取低 21 位地址
#define LCD_WIDTH 320 // 屏幕的宽
#define LCD_HEIGHT 240 // 屏幕的高
//垂直同步信号的脉宽、后肩和前肩
#define VSPW (0) //垂直同步信号的脉宽(X35 屏只能是该值)
#define VBPD (11) //垂直同步信号的后肩
#define VFPD (3) // 垂直同步信号的前肩
//水平同步信号的脉宽、后肩和前肩
#define HSPW (0) // 水平同步信号的脉宽
#define HBPD (68) //水平同步信号的后肩
#define HFPD (1) // 水平同步信号的前肩
//显示尺寸
#define LINEVAL (LCD_HEIGHT-1)
#define HOZVAL (LCD_WIDTH-1)
//for
#define CLKVAL_TFT 7 // 设置时钟信号
#define MVAL_USED 0 // 当屏是 TFT时设置为 0，STN屏时设置为 1
#define PNRMODE_TFT 3 // 选择显示模式 TFT 型 LCD
#define BPPMODE_TFT 12 //16 位 TFT型 LCD
//for LCDCON5
#define BPP24BL 0 //32 位数据表示 24 位颜色值时，低位数据有效，高 8 位无效
#define INVCLK 1 // 像素值在 VCLK下降沿有效
#define INVLINE 1 // 翻转 HSYNC信号
#define INVFRAME 1 // 翻转 VSYNC信号
#define INVVD 1 // 翻转 VD信号极性
#define INVVDEN 1 // 翻转 VDEN信号极性
#define PWREN 1 // 使能 PWREN信号
#define BSWP 0 // 颜色数据字节不交换
#define HSWP 1 // 颜色数据半字不交换(24BPP 时不用交换)
//定义显示缓存区 320 240

```

```

                U16类型，如果想用 24BPP则要用 U32
volatile U16 LCD_BUFFER[LCD_HEIGHT][LCD_WIDTH];
// 汉字表：
//（取模方式，横向取模，字节正序）
unsigned char const zhen[] = {
0x00,0x00,0x08,0x00,0x00,0x20,0xFE,0xFC,0x0B,0xF8,0x01,0x20,0x28,0x40,0x0A,0x08,
0xFD,0x20,0x28,0x40,0x12,0x08,0x11,0xFC,0xFE,0x40,0x12,0x08,0x11,0x20,0xAA,0x78,
}
unsigned char const bao[] = {
0x33,0xF8,0x12,0x20,0xAA,0x48,0x30,0x40,0x7C,0x20,0xFE,0x48,0x50,0x40,0x13,0xFE,
0x10,0xA8,0x97,0xFC,0x10,0x70,0x10,0x98,0x10,0xE0,0x10,0xA8,0xFE,0x88,0x11,0x50,
}
unsigned char const zhu[] = {
0x10,0xA8,0x10,0x88,0x12,0x48,0x1D,0x24,0x10,0xAA,0x14,0x44,0xE1,0x24,0x1E,0xCA,
0x18,0x42,0x42,0x22,0xF0,0x8A,0x10,0x40,0x00,0x20,0x40,0x06,0x10,0x40,0x00,0x20,
}
/***** 函数名： LCD_Init(void )
***** 功能： LCD初始化
***** 参数： 无
***** 返回值： 无
*****/
void LCD_Init(void)
{
    rGPCUP = 0x00000000;
    rGPCCON = 0xaaaa02a9;
    rGPDUP = 0x00000000;
    rGPDCON=0xaaaaaaaa; //Initialize VD[15:8]
    rLCDCON1=(CLKVAL_TFT<<8)|(MVAL_USED<<7)|(PNRMODE_TFT<<5)|(BPPMODE_TFT<<1)|0;
    rLCDCON2=(VBPD<<24)|(LINEVAL<<14)|(VFPD<<6)|(VSPW);
    rLCDCON3=(HBPD<<19)|(HOZVAL<<8)|(HFPD);
    rLCDCON4=(HSPW);
    rLCDCON5 = (1<<11) |(INVCLK<<10) | (INVLINE<<9) | (INVFRAME<<8) | (0<<7) |
(INVVDEN<<6) | (PWREN<<3) | (BSWP<<1) | (HWSWP);
    rLCDSADDR1=((U32)LCD_BUFFER>>22)<<21|M5D((U32)LCD_BUFFER>>1);
    //如果是 24BPP时是 M5D( ((U32)LCD_BUFFER+(LCD_WIDTH*LCD_HEIGHT*4))>>1);
24BPP一个字占 4 字节
    rLCDSADDR2=M5D( ((U32)LCD_BUFFER+(LCD_WIDTH*LCD_HEIGHT*2))>>1 );
    rLCDSADDR3=LCD_WIDTH*2/2; //如果是 24BPP时是 LCD_WIDTH*4/2 因为 24BPP一个字
占 4 字节
    rLCDINTMSK|=3; // MASK LCD Sub Interrupt LCD 里应用一般用不到中断，关掉
    rTCONSEL = 0; // Disable LPC3480

    rGPGUP=rGPGUP&~(1<<4)|(1<<4); // Pull-up disable
    rGPGCON=rGPGCON&~(3<<8)|(3<<8); //GPG4=LCD_PWREN (设置 GPG4为 LCD电源管脚)
    rGPGDAT = rGPGDAT | (1<<4); //10000
    rLCDCON5=rLCDCON5&~(1<<3)|(1<<3); // PWREN( 开电源使能)
    rLCDCON5=rLCDCON5&~(1<<5)|(0<<5); // INVPWREN( 电源使能信号不反转)
    rLCDCON1|=1; // 开启 LCD显示这句不能少
}
/***** 函数名： PutPixel(void )
***** 功能： 绘制像素点
***** 参数： x : 横坐标 y : 纵坐标 c 颜色
***** 返回值： 无
*****/

```

```

void PutPixel(U32 x,U32 y, U32 c )
{
    LCD_BUFFER[y][x] = c;
}
/*****
**** Draw_Text16(void )
**** 功能:    绘制大小为 16×16 的中文字符
**** 参数:    x : 横坐标 y : 纵坐标 color 文字颜色
               backColor 文字背景颜色 ch : 字模数组
**** 返回值:  无
*****/
void Draw_Text16(U32 x,U32 y,U32 color,U32 backColor,const unsigned char ch[])
{
    unsigned short int i,j;
    unsigned char mask,buffer;
    for(i=0;i<16;i++)
    {
        mask=0x80;           //          掩码 1000 0000
        buffer=ch[i*2];       //          提取一行的第一个字节
        for(j=0;j<8;j++)      //左边 8 点
        {
            if(buffer&mask)
            {
                PutPixel(x+j,y+i,color); //          为笔画上色
            }
            else
            {
                PutPixel(x+j,y+i,backColor); //          填充背景色
            }
            mask=mask>>1;
        }
        mask=0x80;           //          掩码
        buffer=ch[i*2+1];     //          提取一行的第二个字节
        for(j=0;j<8;j++)     //          右边 8 点
        {
            if(buffer&mask)
            {
                PutPixel(x+j+8,y+i,color); //          为笔画上色
            }
            else
            {
                PutPixel(x+j+8,y+i,backColor); //          填充背景色
            }
            mask=mask>>1;
        }
    }
}
/*****
**** 函数名:  delay()
**** 功 能:  延时
**** 参 数:  a : 延时参数
**** 返回值:  无
*****/
void delay(int a)
{
    int k;
    for(k=0;k<a;k++) ;
}

```



```

}
/*****
**** 函数名: Brush_Background( )
**** 功能: 绘制屏幕背景颜色, 颜色为 c
**** 参数: c 颜色
**** 返回值: 无
*****/
void Brush_Background( U32 c)
{
    int x,y ;
    for( y = 0 ; y < LCD_HEIGHT ; y++ )
    {
        for( x = 0 ; x < LCD_WIDTH ; x++ )
        {
            LCD_BUFFER[y][x] = c ;          //0b1111 1000 0000
        }
    }
}
/*****
**** 函数名: main()
**** 功能: LCD 写汉字, 字符测试程序
**** 参数: 无
**** 返回值:
*****/
int main(void)
{
    int i,j,k,x,y;
    LCD_Init();          //LCD 初始化
    //绘制白色背景
    Brush_Background(0xFFFFFFFF);
    while(1)
    {
        //向左滚动
        for (j=0;j<319;j++)
        {
            //绘制汉字字符, 红字黑背景
            Draw_Text16(200-j,100,0xf800,0x0000,zhen);
            Draw_Text16(216-j,100,0xf800,0x0000,bao);
            Draw_Text16(232-j,100,0xf800,0x0000,zhu);
            delay(10000);
            //绘制名字后的白色背景
            for( y = 90 ; y < 120 ; y++ )
            {
                for(k=248-j;k<319+200-48-j;k++)
                {
                    LCD_BUFFER[y][k] = 0xFFFF ;
                }
            }
        }
        //向右滚动
        for (j=0;j<319;j++)
        {
            //绘制汉字字符, 绿字黑背景
            Draw_Text16(100+j,150,0x07e0,0x0000,zhu);
            Draw_Text16(116+j,150,0x07e0,0x0000,bao);
            Draw_Text16(132+j,150,0x07e0,0x0000,zhen);
            delay(10000);
            //绘制名字后的白色背景

```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/006030153240010214>