数码相册

《嵌入式系统结构与设计》期末 PROJECT

2016-1-13

SS, SYSU

目录

小组计划与分工

小组计划

利用这学期学习的知识，在基于 PXA270 开发板上实现数码相册，功能包含图片定时自动轮播，手动切换。数码管时间显示。另外，加入文字显示以及日期显示，使得本项目具备了一定的互动性。本项目展示的图片为国旗，文字为国家名字，通过控制文字的显示与否，可以使用本项目进行“限时猜国旗”游戏，增加了互动性与趣味性。

小组分工

XXX	**XXX**
键盘中断	定时器中断
LCD 初始化，图片显示以及切换	文字显示以及数码管显示
协作完成设计报告，测试报告以及经验总结报告（完成中）	

Table 1 **小组分工明细表**

小组互评

打分人	被评人	得分
XXX	XXX	100
XXX	XXX	100

Table 2 **组内互评表**

我们小组组员都非常用心以及认真，最后几周都会抽空去实验室做这个期末项目。特别是最后的一周，几乎是熬夜看代码，花了很大的功夫仔细看了代码示例以及参考书籍，然后花了好几个整天的时间在实验室做实验。终于做出来了。不管效果怎么样，我们的组员相互之间都对对方感到很满意。就这个学习态度，我们都给对方满分。

第一章．项目概述

项目预定目标

我们小组的要达到的目标:基于 PXA270 开发板，利用本学期的学习内容完成一个数码相册，具体如下：

1) 用 PXA270 开发板上的 LCD 显示相片，有多张图片可进行切换。
2) 用系统定时器中断，可以定时播放照片。
3) 用小键盘中断方式设计各种按键功能，如：前一张，后一张，自动播放，停止，开始等等。
4) 利用数码管来显示当前时间。

最终完成功能

本小组按照预定计划，顺利地实现了所有的基础功能。另外，经过探讨与摸索之后，还实现

了如下的扩展功能

1） 利用 PXA270 开发板上的数码管来显示当前日期。

2） 在 PXA270 开发板上的 LCD 显示文字。

本项目展示的图片为国旗，文字为国家名字，通过控制文字的显示与否，可以使用本项目进行“限时猜国旗”游戏，增加了互动性与趣味性。

测试结果

每个基本功能都能正常实现，扩展的功能也都能够如我们所愿的实现。本项目使用直入键盘和矩阵键盘，具体地效果如下

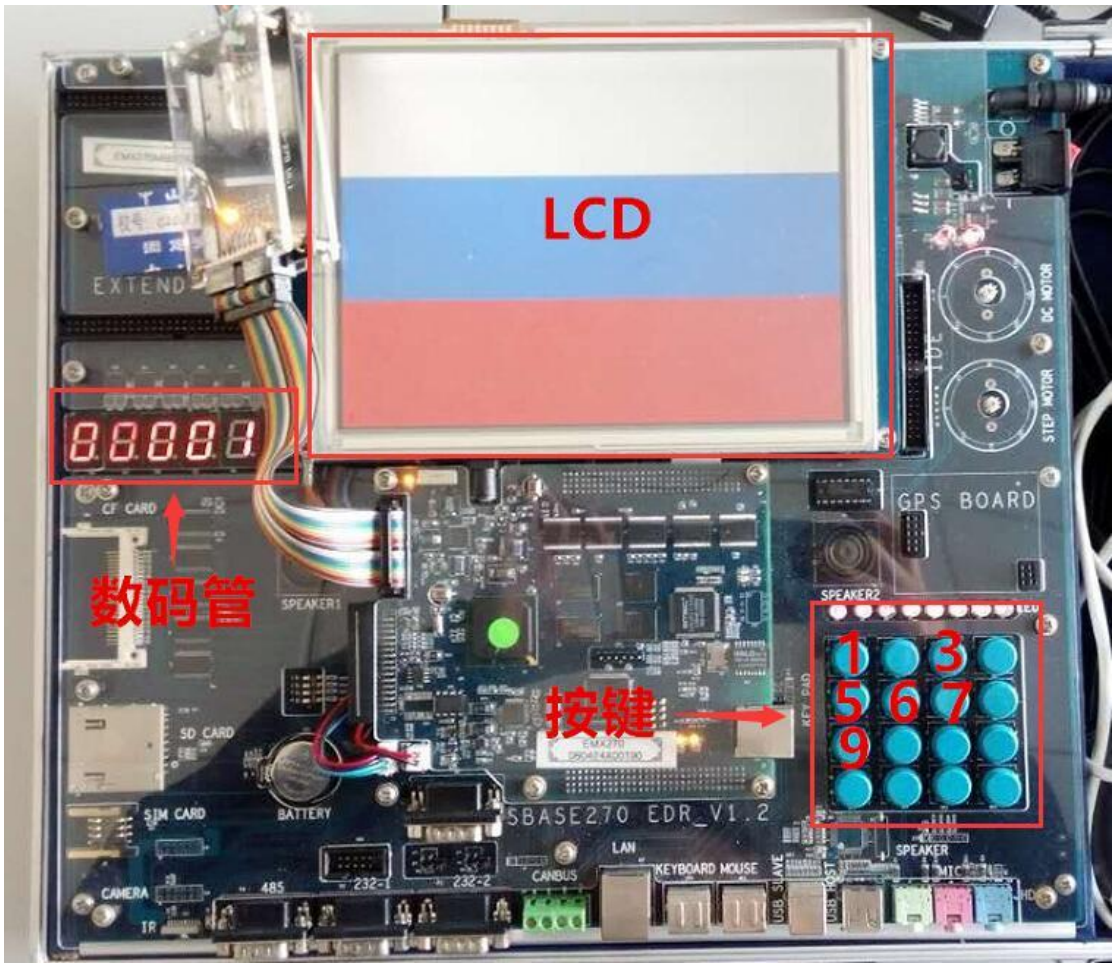

Figure 1 PXA270 板示例图

1） 开机时，数码管显示时间

2） 按下按键 1，成功显示当前日期。

3） 按下按键 3，成功显示文字，再次按下即关闭文字。

4） 按下按键 5，LCD 成功显示上一张图片。

5） 按下按键 6，LCD 成功显示下一张图片。

6） 按下按键 7，启动图片轮播，LCD 上的照片进行循环播放，再次按下即停止轮播。

7） 按下按键 9，重置时间，日期以及图片

第二章．项目人员组成及分工

XXX 负责实现

1） 键盘中断功能（用小键盘中断方式设计各种按键功能，如：前一张，后一张，自动播放，停止，开始等等。）
2） LCD 初始化，图片的显示以及图片切换

XXX 负责实现

1） PXA270 开发板上的数码管来显示当前时间以及当前日期。
2） 系统定时器的中断，可以定时播放照片。
3） 在 PXA270 开发板上的 LCD 上显示文字。

小组成员共同协作完成

1） 2015 年《嵌入式系统结构与设计》期末 project 设计报告，
2） 经验总结报告、
3） 测试报告
4） 答辩用的 PPT。

第三章．项目效果

最终效果图如下所示，由于只能放静态图，所以此处仅贴上所有可能出现的画面，无法体现轮播效果。

Figure 2 **数码管显示系统时间** 00:02:48

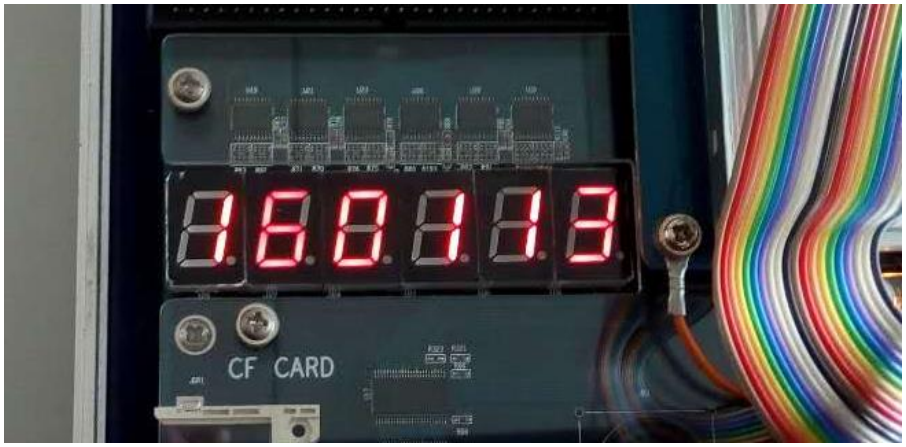

Figure 3 **数码管显示日期** 2016 年 1 月 13 日

Figure 4 显示图片一

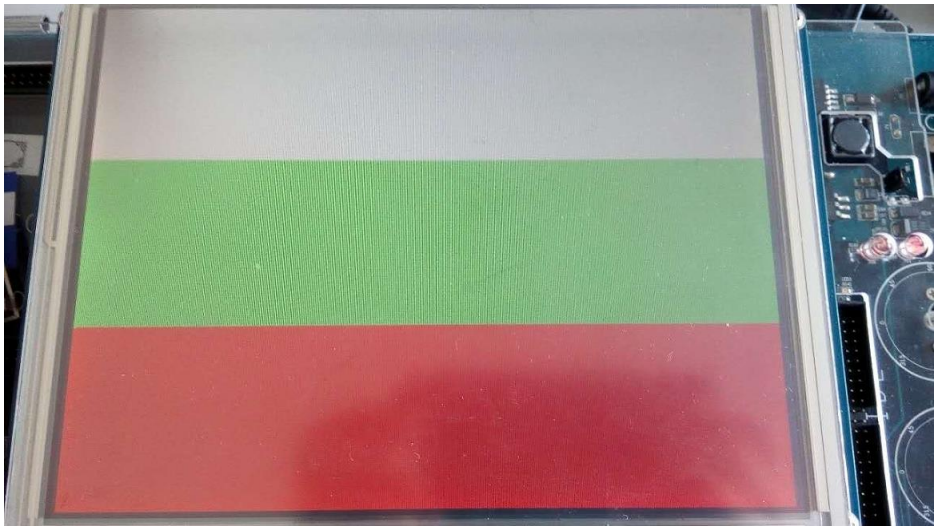

Figure 5 显示图片二

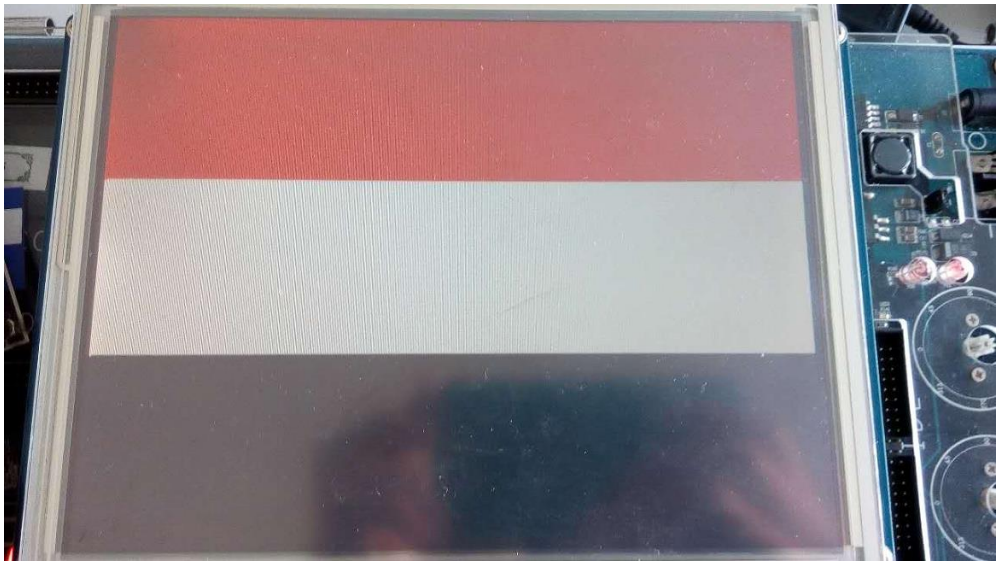

Figure 6 显示图片三

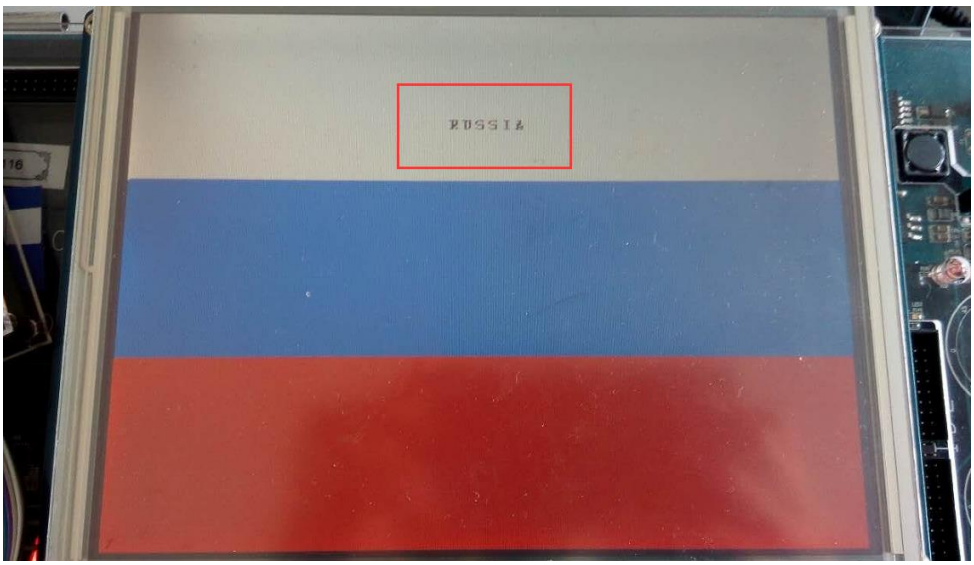

Figure 7 俄罗斯国旗，带名称 RUSSIA

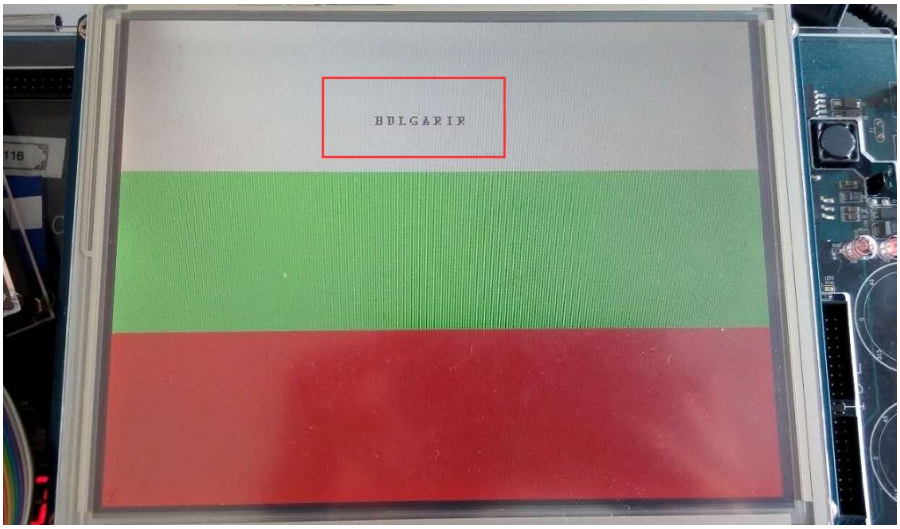

Figure 8 保加利亚国旗，带名称 BULGARIR

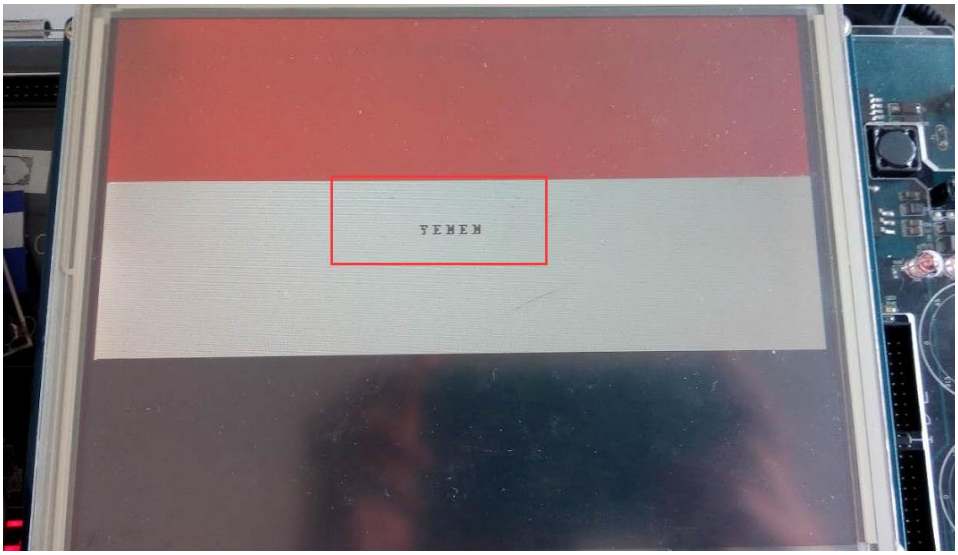

Figure 9 也门国旗，带名称 YEMEN

第四章．项目开发过程

数码管显示日期与时间

本过程可以细分为两部，一个是数码管显示，一个是实时时钟。

数码管显示原理

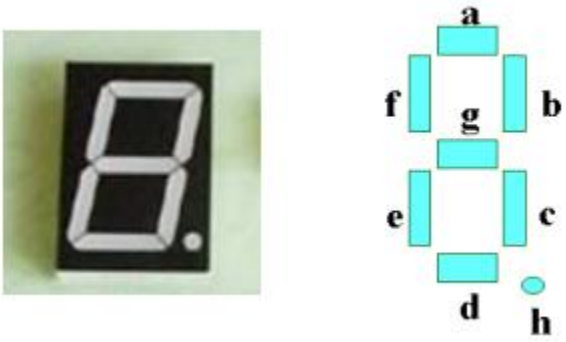

Figure 10 七段数码管结构图

本实验中使用的是共阳数码管，所以，当把相应的位置设置为 0 时，该段就会亮起。又由于最高位 dp 位必须输出低电平，对应的数码管才能够正常工作。所以，就是把最高位置 0，后面按照想要的显示形状，按 gfedcba 排好，再转化为 16 进制码，就可以显示全部数字了。

显示	共阳	显示	共阳
0	0x40	1	0x79
2	0x24	3	0x30
4	0x19	5	0x12
6	0x02	7	0x78
8	0x00	9	0x10

Table 3 数码管对应 16 进制码（基础使用文档中的 5 为 0x22 和 6 为 0x12 有误，应该是 5 为 0x12,6 为 0x02，谨此说明）

然后，我们把这些 16 进制码存储在一个 unsigned long 的数组 NUM_CODE 中，以备后续使用。

注意：如果需要将该数组存放在其他文件，然后再 extern 到主程序中的话，必须将该数组声明为 const，否则会出现主程序无法读取该数组的情况。后续的照片显示以及文字都是需要声明为 const 的，不再累述。

实时时钟

需要初始化 RTSR，RCNR 寄存器，然后，通过读取 RYCR 来获取日期，读取 RDCR 来获取时间。具体的读取方法以 RDCR 为例，如下

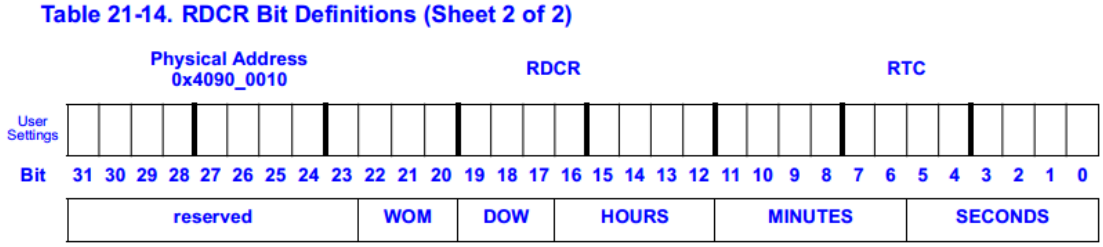

Figure 11 RDCR 寄存器结构

所以，可以通过移位的方法，来分离出其中的时，分，秒。具体方法如下：

```
(RDCR << 15) >> 27        //HOURS
(RDCR << 20) >> 26        //MINUTES
(RDCR << 26) >> 26        //SECONDS
```

而日期的分离方法与此相似，可参考 *Intel PXA27x Processor Family Developer's Manual* 的 *21.5.10* 中的 RYCR 寄存器来进行。

数码管与时钟结合

本项目使用一个 get_led_display 函数来实现两者的结合，具体的思路就是通过求 Mod 和除法将读入的时或分等，分解为十位和个位。然后，将个位对应的数码管显示码左移 8 位，再加上十位，就得到了一个完整的显示码，再将其赋值给 LED_CSn 即可。get_led_display 代码如下

```
unsigned long get_led_display(unsigned long num)
{
    int unit = (int)num % 10;
    int decade = ((int)num / 10) % 10;
    unsigned long answer = NUM_CODE[unit];
    answer <<= 8;
    answer += NUM_CODE[decade];
    return answer;
}
```

键盘模块

需要先在 boot 中开启键盘中断，主要有：

1） 在 library_regitster 中 EXPORT ICMR 以及 init_ICMR。

2） 在 library_variant 中 EXPORT PSSR(Power Manager Sleep Status Register)

3） 在 boot 中 IMPORT 以上三个并对其赋予初始值，开启键盘中断。

接下来，注释掉 boot 中原有的 IRQ 跳转哑函数，添加 handler_IRQ.s，并在主程序中添加 IRQ 处理函数。

在 IRQ 处理函数中，根据读取到的 KPDK_VALUE(直入键盘)以及 KPAS_VALUE(矩阵键盘)的值，来执行相应的动作。比如说，本项目中，当按下按键 1 时，显示日期，代码如下

```
switch (KPDK_VALUE)
{
    case 0x40:                          //key-press 1, show date
        led_display(2);
        Delay(500);
        break;
……
}
```

LCD 模块

本模块是本项目中，涉及的汇编代码较多的部分，需要初始化较多的寄存器。

LCD 初始化

1） 在 library_register.s 中对需要使用到的寄存器初始化并 EXPORT，大致包括相关 GPIO，LCD 控制器以及 Frame Descript。

2） 添加 post_initDescrpit.s 文件，其中主要是对 LCCR 等寄存器进行初始化。
3） 添加 print_method.c 文件，其中包含了许多屏幕打印函数，可修改后用于后续的照片显示。
4） 修改 post_initGPIO.s 文件，将其中的 GAFR2_L 寄存器的值修改为 0xaaaaaaaa，否则屏幕会花屏。（此步骤非常重要）

由于本项目的 LCD 初始化不需要进行什么特殊变化，所以，相关寄存器的值除了 GAFR2_L 之外，其他都不用修改，直接复制基础实验 ads 版中的实验 8LCD 实验即可。

图片显示

LCD 屏幕参数

PXA270 的 LCD 分辨率为 256x600，帧缓冲区输出到屏幕的顺序为从右往左，从上往下。

将图片转为数组

1） LCDC 内部调色板为 16 位/像素，RGB 采用 5:6:5 格式。
2） 将彩色图片 convert 为 RGB 格式，再将其 spilt 为 R，G，B 三个通道。
3） 正常情况下每个通道的每个像素为 0~255，需要 8 个位。而对于只有 5 个位的，可以使用 Quantizer 的方法压缩。
4） 最后再将每个像素的三个数值拼合，转化为 16 进制数保存。

可以使用 Python 的 PIL 结合 Numpy 实现。

实现思路

先将图片的每个像素转化为 16 进制数数组，然后将值写入帧缓冲区，逐个像素显示。

文字显示

方法与图片显示类似。实际上文字也是通过操作每个像素的显示，将文字“画”到 LCD 上的。

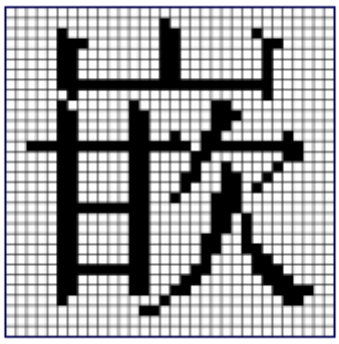

Figure 12 文字字模

计算机只需用 1 位来表示 1 个小格是否有填色，如果有填色，则用 1 表示，没有则用 0 表示，最后再转化为 16 进制数。

不过，本项目显示的是英文，所以只需创建 26 个字母的显示的数组即可。另外，*基础实验 ads 版*实验 8LCD 实验中的 char_8x13.c 文件中，给出了常见的字符的 16 进制数显示码，可以直接使用。

图片切换

1） 图片上下翻页。对于每张图片，使用一个 index 来进行标示，然后键盘中断时，响应事件为修改 index 的值，这样便可以实现上一页和下一页的效果。

```
switch (KPAS_VALUE)
{
    case 0x00:                          //key-press 5, previous page
        index--;
```

```
            lcd_display();
            break;
      case 0x01:                          //key-press 6, next page
            index++;
            lcd_display();
            break;
……
}
```

2） 图片自动播放。在主函数 dummyOS 中，设置一个循环来播放照片，该循环通过一个标志来启动。这样，在按下某一按键时，修改该标志，便可启动该循环，触发图片自动播放事件。图片的轮播之间，需要一定的时间间隔，这个使用定时器中断来实现，后续将会说明。

3） 停止播放。修改该循环标志为否，停止循环。

循环播放图片代码如下：

```
void dummyOs(void)
{
……
            while (autoplay == 1) {
                  lcd_display();
                  led_display(1);
……
```

定时器模块

需要先开启定时器中断。

1） 将 library_register.s 中的 init_ICMR 的值重设为 0x3c400010。

2） 在 handler_IRQ.s 中，启动定时器中断，注意，应该将 MOV R1,#0x3c400010 修改为 LDR R1, =init_ICMR，因为前者会造成地址越界而无法通过编译。

3） 然后在主程序中加入 OIER， OSCR，OSSR，OSMR0

4） 给定时器 0OSMR0 添加一个定时，通过判断 OSSR 的值，然后做出相应，即可实现定时器中断。

启动定时器的代码如下

```
OIER = 0x1;
pretimer = OSCR;
OSMR0 = pretimer + 0x800000;
```

第五章．项目总结

时间的连续显示

问题描述

一开始将时间显示写在中断处理函数中，目的是当按下按键时，才显示时间。后来发现这样是行不通的，当触发中断时，程序会跳转到中断位置继续运行，知道该处理函数执行完在跳出来。所以，如果写在中断中，每次按下时，然后显示某一时刻的时间，就停止了。

解决方法

将时间显示写在主函数 dummyOS 中，给定一个死循环，这样，就会一直显示。当需要跳转

到某些中断中进行，若遇到需要使用 Delay 的情况，可将 Delay 切分得小一点，在中间加入时间显示函数。也就是模拟 CPU 的时间分片的方法，造成一种数码管时间一直在显示的“假象”。

数码管日期初始化以及显示异常

问题描述

本项目通过按下按键 1，将数码管的显示内容切换为日期。但是，一开始按下按键没有任何效果，后来修正之后又发现日期与预期的初始化不一致，一直是 00.01.01.

解决方法

一开始没有效果，是因为没有在中断处理并显示日期的后面，追加 Delay。这样，处理完显示日期之后，程序马上就跳出中断进入主函数，由于主函数中会循环执行时间显示，这样就冲刷了日期显示，因此没有效果。追加 Delay 就可以解决。

而日期显示的异常，需要在上述的 get_led_display 函数中，将原来得到的 unsigned long 的值强制转换为 int。在初始化时，注意先初始化时间寄存器 RDCR，再初始化日期寄存器 RYCR。这点相当重要。

LCD 花屏

问题描述

初始化 LCD 屏，烧入开发版中，发现开机之后屏幕颜色越来越深，后面就全部黑了。执行任何操作都没有用。

解决方法

我们小组探索了好久，找不到元婴。后来询问其他小组的同学，需要修改 post_initGPIO.s 文件，将其中的 GAFR2_L 寄存器的值修改为 0xaaaaaaaa，这样就顺利解决问题了。

图片打印的巧妙思路

问题描述

将图片转化为数组的方法，会遇到一个问题，就是数组的大会非常巨大。比如一个 50x50 的小图片，像素就有 2500 个，这样必将导致生成的二进制文件非常大，烧入开发板的时间会特别长（1~2 小时），不利于调试。

解决方法

我们小组探讨之后，得到一个较为巧妙的解决方案。就是我们打印内容较为单一的图片，比如国旗。这样我们就只需要两三种颜色，然后通过控制指针的移动，实现在屏幕上手动打印简单的图案。这样子，就不再需要一个几千个元素的图像数组，改为一个几个元素的颜色数组即可，大大减小了生成的二进制文件。

另外，由于打印图案时，不用每个像素都更改一次颜色值，可以大量减少寄存器的赋值操作，提高图片的打印速度。

只能中断一次

问题描述

当修改 library_register.s 中的 init_ICMR 开启定时器中断之后，发现键盘中断以及定时器中断都只能触发一次。

解决方法

分析汇编代码之后发现，需要对应修改 handler_IRQ.s 中的初始化，将 MOV R1,#0x3c000000 修改为 LDR R1, #0x3c400010，这样便可正常使用中断了。

项目进一步完善的建议

1) 可以考虑加入图像的放大缩小等变化
2) 可以考虑使用 flash 来存放图片，这样就可以放置更加生动有趣的图片了。
3) 加入蓝牙模块，实现手机客户端与 PXA270 版的互动
4) 增加串口，实现电脑与 PXA270 板的信息传输

参考文献

1. 亿道电子科技基础实验文档
2. Intel PXA27x Processor Family Developer's Manual