

经验总结报告

《嵌入式系统结构与设计》期末 PROJECT 数码相册

目录

一.	项目思考与总结.....	1
1.	三种帮助.....	1
2.	两个麻烦.....	2
二.	遇到难题及解决方案.....	2
1.	代码无法烧入开发板.....	2
2.	时间的连续显示.....	2
3.	数码管日期初始化以及显示异常.....	2
4.	LCD 花屏.....	2
5.	图片打印的巧妙思路.....	3
6.	只能中断一次.....	3

一. 项目思考与总结

本次项目涉及的知识点相对较多，基本上涵盖了本学期课堂上所学习到的基本知识，包括键盘中断，定时器中断，数码管显示，实时时钟，LCD 显示器等等。虽然之前一直都很认真地完成了其他小实验，但是，当把各个部件组合起来使用的时候，还是遇到较多的问题。

1. 三种帮助

在完成本次实验中，以下三点对我们起了很大的帮助

- 1) 两份文档，分别是基础实验文档以及 Intel PXA27x Processor Family Developer's Manual。前者包含了我们需要用到的模块的具体使用方法的介绍，而后者则是所有寄存器结构的介绍，一般可以根据名称搜索，然后查看该寄存器的结构，分析之后可以用来做出相应的初始化
- 2) 一份代码，基础实验 ads 版。这份代码非常地具有参考价值，一般出问题就对比一下汇编代码与该实例代码中的不同，然后分析该代码的含义，或者找到对应寄存器的设置，进行适当的修改，一般就可以解决问题。
- 3) 与其他小组交流。大家遇到的问题大同小异，逻辑上的实现一般没有太大问题，主要在于某些特定寄存器值得设置。通常组合各种模块之后，进行初始化时需要变动的就是一两个关键的寄存器，自己琢磨需要耗费大量时间，如果多与其他小组交流，完全可以起

到事半功倍的效果。

2. 两个麻烦

下面再提两点此次大作业过程中的麻烦之处。

- 1) 没有 DEBUG 模式。由于使用的是交叉编译, 代码需要在板上才能跑。所以, 每次都必须写完代码, 生成二进制文件, 烧入板上才能看到效果。这样, 就没有办法加入断点来进行调试, 每次面对错误时都会显得十分无力与茫然。貌似实验使用的编辑器是有生成 DEBUG 之类的文件的, 但不知道该如何使用。
- 2) 没有 PXA270 实验箱的模拟开发环境。大作业必须在实验室完成, 时间和空间上都相对不自由。如果有模拟开发环境, 将大大提高编码的灵活性, 也可以提高编码的热情。

二. 遇到难题及解决方案

下面仅就本次项目开发过程中遇到的一些难题以及其解决方案做个总结, 以供交流与学习。

1. 代码无法烧入开发板

问题描述

PXA270 开发板有些老化严重, 保养也不大好, 有时候烧代码到主板时总是会出现各种错误。

解决方法

大概可以分为以下几种情况

- 1) Cable error, 这个是 jtag 线有问题, 更换一个一般就可以了。
- 2) Cannot find device ID, 这个有可能是系统没有安装 gpio 驱动, 安装一下驱动, 重启电脑, 再试一次; 也有可能是 PXA270 板的电源开关没有打开
- 3) Verification error, 多重启几次箱子, 多烧几次代码, 一旦成功烧进去一次, 之后出现烧录时一般就不会再出现该错误了。

2. 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”。

3. 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。

4. 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。

5. 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，就是数组的大小非常巨大。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。

6. 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。

2016 年 1 月 13 日