

实验一 字符设备控制

一、实验目的：

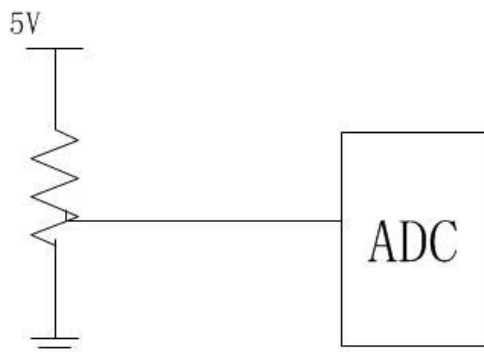
- 1 通过本实验的学习，了解 Linux 操作系统中的字符设备驱动程序结构
- 2 能编写简单的字符设备的驱动程序以及对所编写的设备驱动程序进行测试
- 3 了解 Linux 操作系统如何管理字符设备

二、实验器材：

ltop 4412 实验开发板一块
配套交叉串口线一根
微机一台 (Windows 操作系统)
U 盘一个

三、实验原理：

开发板上自带一个 LED，一个 12 位的 ADC 和一个阻值为 10K 的滑动变阻器。



ADC 模拟输入端和滑动变阻器滑动端相连，测得滑动端的电压值，即可换算出比值。

字符设备驱动程序主要包括初始化字符设备、字符设备的 I/O 调用和中断服务程序，主要包含以下几个函数：

open()：打开字符设备

memset()：填充内存

read()：字符设备读取

write()：字符设备写入

atoi()：ASCII2Int，字符串转整型

[int ioctl\(int fd, unsigned long request, ...\)](#)：字符设备 IO 控制，返回 0 表示成功

sleep(i) 线程休眠等待 i 秒

usleep(i) 线程休眠等待 i 毫秒

四、实验内容

编写一个简单的字符设备驱动程序

要求：

0.完成 helloworld 程序。

1.采集总阻值为 10K 的滑动变阻器的电阻值并通过控制台（printf 函数）显示

2. 控制两个 LED 灯以 1Hz 频率**循环**点亮（灯 1 闪一次，灯 2 闪一次，循环）

3.常态下，若增大滑动变阻器阻值，LED 闪烁频率随之明显减小；若减小滑动变阻器阻值，LED 闪烁频率随之明显增大（**负相关**）

4.若滑动变阻器阻值大于 9K 或低于 1K 视为超限，蜂鸣器以 **1Hz 频率**警报且 LED 暂停闪烁。超限解除后恢复常态（蜂鸣器停止，LED 灯随阻值变化）。

5.（**提高**）通常情况下，我们不会设定一个绝对的限值进行比较，而是采用类似斯密特触发器的方式处理超限警报，来降低边界附近由于抖动或测量误差造成的状态跳动。例如，原始程序是判断当前阻值低于 1k 时，进行下限警报，那么可以改为低于 950Ω进行下限警报，到达下限后高于 1050Ω恢复常态。例子中数字只为说明原理，实际数值可选取为便于展示效果的数值。

注：可参照提供的部分源码实验一/ex1.c，在此基础上实现。

五、实验步骤

1.编写并且编译源代码，生成可执行文件，拷贝到 U 盘

2.启动开发板，在超级终端中挂载 U 盘，执行可执行文件查看效果

3.具体挂载执行步骤可见附件"连接开发板并执行 helloworld.docx"