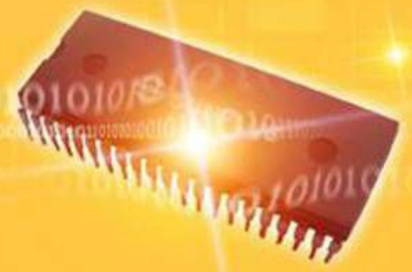
嵌入式系统工程师



Linux IO控制技术

大纲

- Linux 设备控制概述
- Linux IO设备控制

大纲

- Linux 设备控制概述
- Linux IO设备控制

Linux 设备控制概述

对于硬件设备，Linux采用了与裸机完全不同的机制进行管理。

Linux下的所有硬件(IO、键盘、鼠标等)均是以文件的形式进行统一管理的，每个设备在/dev/目录下都有一个设备文件与之对应。操作相应的文件即是操作相应的硬件，从而大大简化了设备的操作过程。

Linux系统对各式各样的硬件采用了类似的架构编写、管理设备驱动程序。加载驱动的时候会在/dev/目录下生成设备文件。

Linux 设备控制概述

- 终端下我们可以通过ls /dev/命令来查看当前系统下所有的设备文件。

```
[root@sunplusedu /]#ls /dev/
audio       fb2
console     fb3
dsp         s3c2410_serial0
event0      s3c2410_serial1
fb0         s3c2410_serial2
fb1         s3c2410_serial3
```

总之：在Linux的世界里，一切设备皆文件。

Linux 设备控制概述

- 操作设备文件的函数
  - 打开、关闭

```
int open(const char *pathname, int flags);
int close(int fd);
```

  - 读、写

```
ssize_t read(int fd, void *buffer,size_t count);
ssize_t write(int fd,void *buffer,size_t count);
```

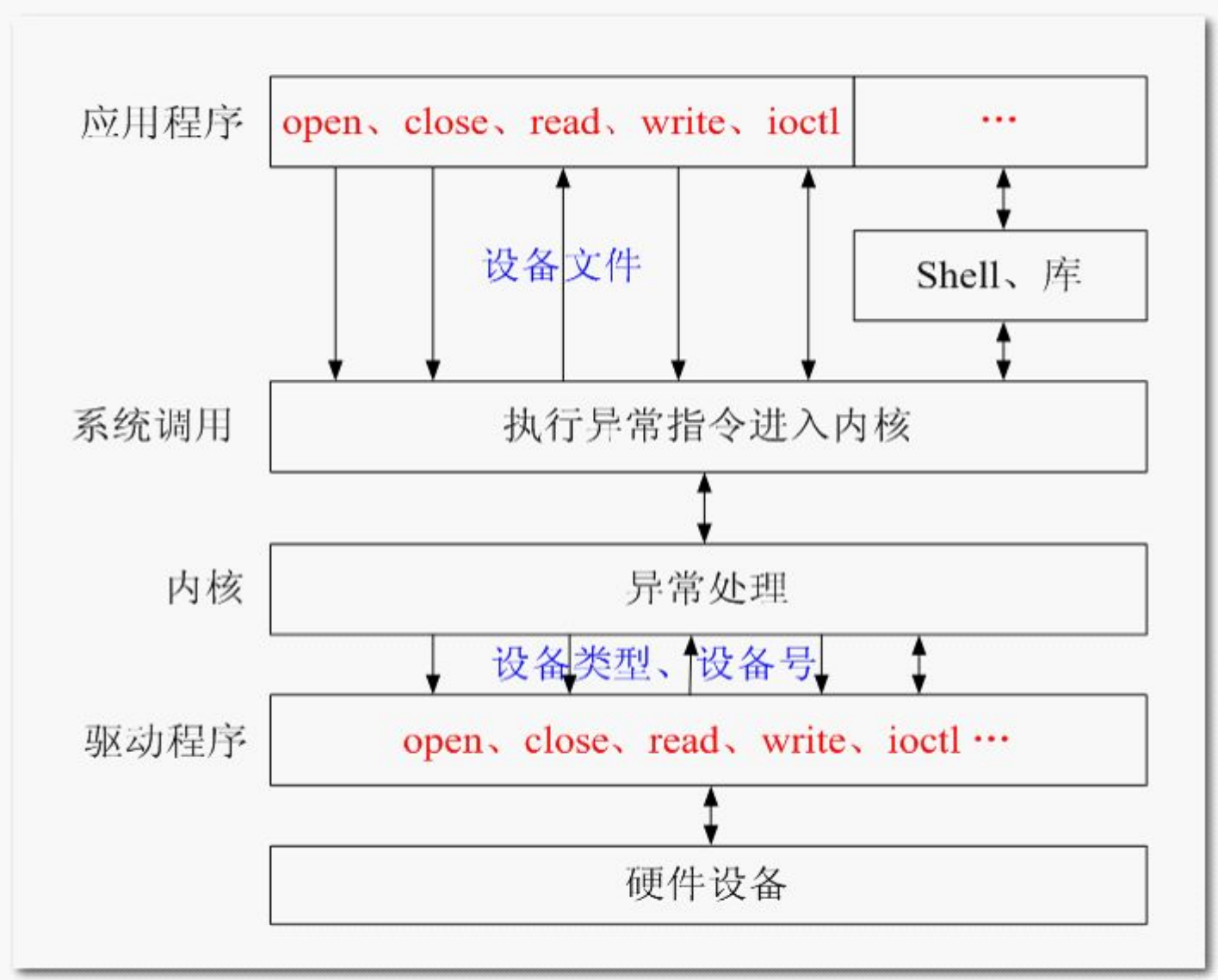
  - 属性控制

```
int ioctl(int fd, unsigned long int cmd, ...);
```

- 驱动文档

s5pv210-gpio.ko对应的文档

Linux 设备控制概述



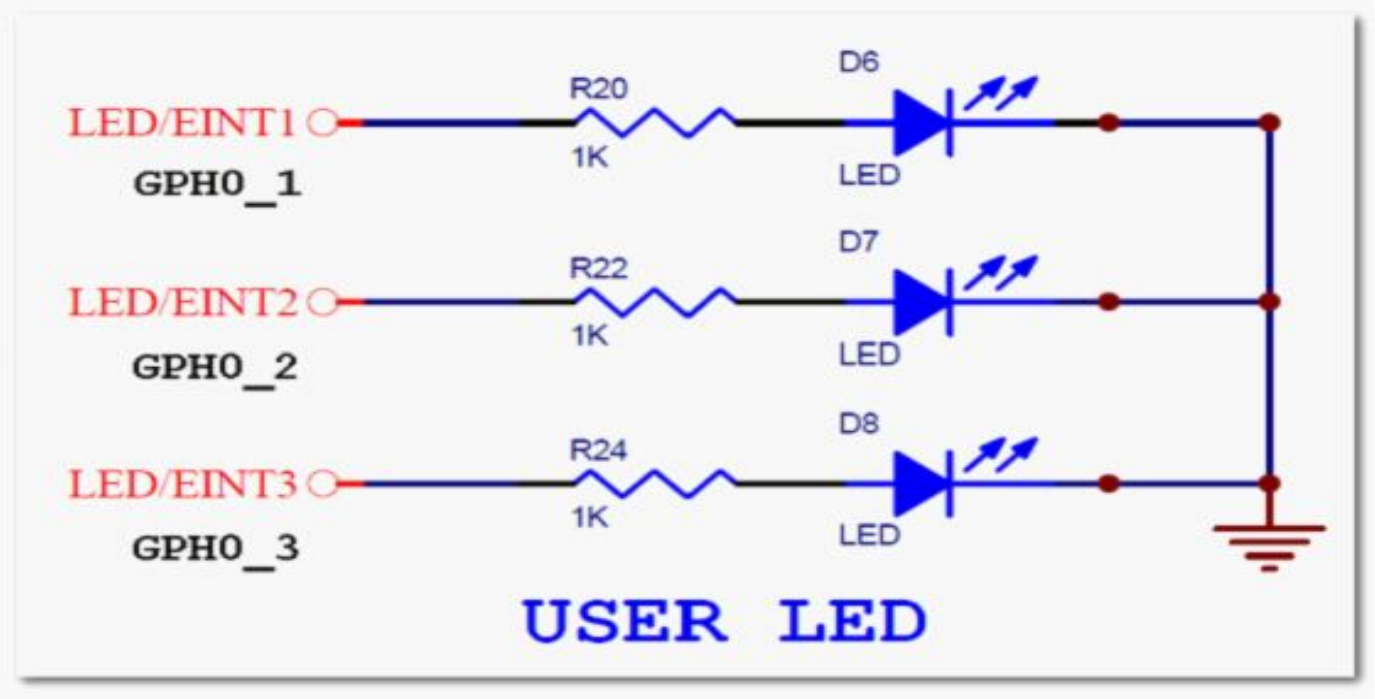
大纲

- Linux 设备控制概述
- Linux IO设备控制
  - IO设备输出（LED灯）
  - IO设备输入（键盘）
  - 监听文件描述符（select）

Linux IO设备控制

- IO设备输出（LED灯）

  电路如图：



GPH0_1、GPH0_2、GPH0_3输出高电平，即可点亮相应的LED灯。例：gpio_led.c

Linux IO设备控制

- GTK按钮控制开发板LED灯

  功能：

  用GTK绘制下图所示界面

  3个按钮分别控制相应的LED灯亮灭

例：

gtk_button_led

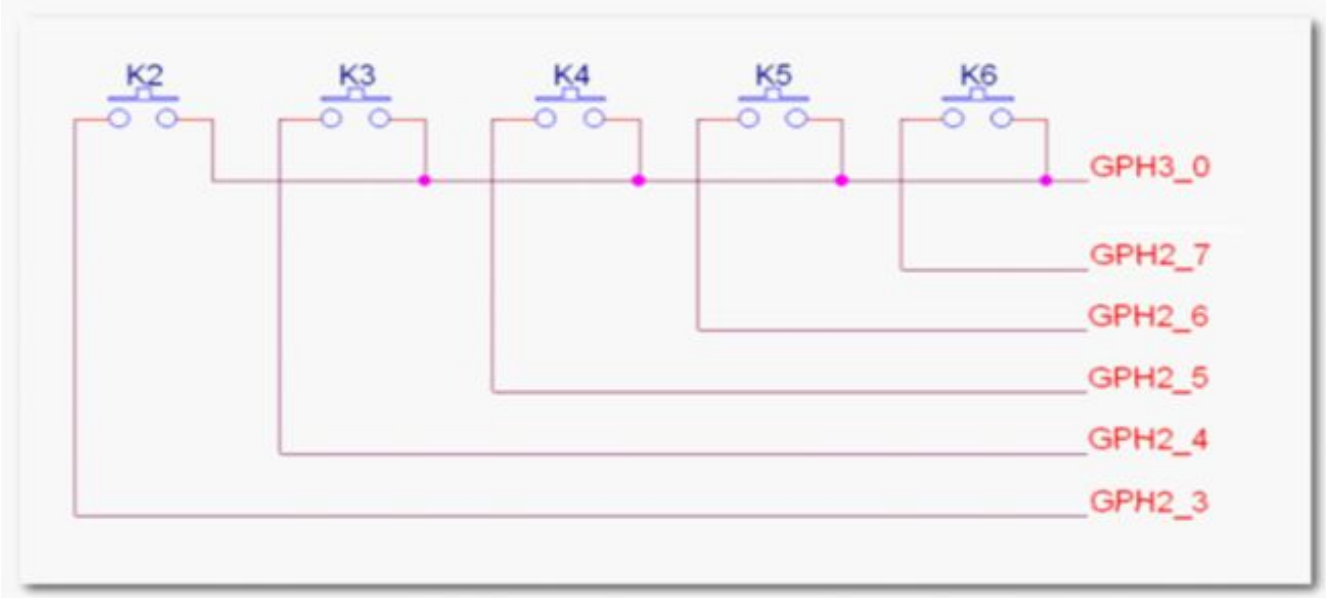
大纲

- Linux 设备控制概述
- Linux IO设备控制
  - IO设备输出（LED灯）
  - IO设备输入（键盘）
  - 监听文件描述符（select）

Linux IO设备控制

- IO设备输入（键盘）

  电路如图：



GPH3_0输出低电平，GPH2_3、GPH2_4、GPH2_5、GPH2_6、GPH2_7设置为输入，读取输入引脚的值即可判断哪个键被按下。例：gpio_key1_5.c

Linux IO设备控制

- 使用驱动工程师提供的键盘扫描程序

  在实际应用时，键盘扫描程序一般并不会由应用工程师去编写，键盘扫描程序会由驱动工程师来写。

  键盘扫描驱动及其使用程序：key4+5。

大纲

- Linux 设备控制概述
- Linux IO设备控制
  - IO设备输出（LED灯）
  - IO设备输入（键盘）
  - 监听文件描述符（select）

Linux IO设备控制

- 监听文件描述符（select）

```
#include <sys/select.h>
int select(int nfds,
           fd_set *readfds,
           fd_set *writefds,
           fd_set *exceptfds,
           struct timeval *timeout);
```

功能：

监听并等待多个文件描述符的属性变化（可读、可写或错误异常）。

Linux IO设备控制

参数：

nfds：要监听的文件描述符的范围。即监听的最大描述符再加1。也可将其设置为<sys/select.h>头文件中的一个常数：FD_SETSIZE（其值经常是1024）。

readfds：读描述符集，其存储了需要监视可读属性变化的文件描述符，select函数返回后，其存储了可读属性已经变化了的文件描述符。

Linux IO设备控制

writefds：写描述符集，其存储了需要监视可写属性变化的文件描述符，select函数返回后，其存储了可写属性已经变化了的文件描述符。

exceptfds：异常描述符集，其存储了需要监视错误异常属性变化的文件描述符，select函数返回后，其存储了错误异常属性已经变化了的文件描述符。

timeout：select函数的超时时间。

Linux IO设备控制

- 文件描述符集

```
void FD_ZERO(fd_set *fdset);
```

功能：清空描述符集。

```
void FD_SET(int fd, fd_set *fdset);
```

功能：将一个描述符添加到描述符集。

```
void FD_CLR(int fd, fd_set *fdset);
```

功能：将一个描述符从描述符集中删除。

```
int FD_ISSET(int fd, fd_set *fdset);
```

功能：测试指定的描述符是否在集合中。

Linux IO设备控制

- timeout：其存储了时间结构体变量的地址。

  时间结构体如下：

```
struct timeval{
        time_t tv_sec;          /* 秒 */
        suseconds_t tv_usec; /* 微秒 */
};
```

Linux IO设备控制

- timeout的取值情况：

1、永远等待：

若timeout = NULL，则select函数等待到其监听的文件描述符属性的变化。

2、等待固定时间

若timeout指向的时间结构体（struct timeval）变量的时间值不等于0秒0毫秒。则select函数在timeout指定的时间内等待其监听的文件描述符属性的变化，超时后select函数不等待返回0。

Linux IO设备控制

3、永远不等待（轮询）

若timeout指向的时间结构体（struct timeval）变量的时间值等于0秒0微秒。则select函数不管文件描述符是否有变化，都立刻返回。文件描述符属性无变化返回0，有变化返回准备好的描述符数量。

Linux IO设备控制

返回值：

情况一：出错，返回-1。

情况二：超时，返回0。

情况三：准备好的描述符数量，其值大于0。

例：select

凌阳教育官方微信：Sunplusedu

Tel: 400-705-9680 , BBS: www.51develop.net , QQ群：241275518