# 一、Linux内 核中断处理简介

## 1.1、裸机中断

## 1.2 linux中断

 1、先知道你要使用的中断对应的中断号。

 2、先申请request\_irq，此函数会激活中断。

 3、如果不用中断了，那就释放掉，使用free\_irq。

 4、中断处理函数irqreturn\_t (\*irq\_handler\_t) (int, void \*)。

 5、使能和禁止中断，

## 1.3 上半部和下半部

 中断一定要处理的越快越好，

 1、软中断

static struct softirq\_action softirq\_vec[NR\_SOFTIRQS] 10个

 要使用软中断，要先注册，使用函数open\_softir。注册以后使用raise\_softirq触发。

 软中断我们不要去用！！

 软中断我们不要去用！！

 2、tasklet

 也需要用到上半部，只是上半部的中断处理函数重点是调用tasklet\_schedule。

 1、定义一个tasklet函数。

 2、初始化、重点是设置对应的处理函数

 3、工作队列

## 1.4 设备树中断节点信息

 1、#interrupt-cells指定interrupt的cells数量，也就是属性interrupts。

 intc: interrupt-controller@00a01000 {

 compatible = "arm,cortex-a7-gic";

 #interrupt-cells = <3>;

 interrupt-controller;

 reg = <0x00a01000 0x1000>,

 <0x00a02000 0x100>;

 };

 gpio5: gpio@020ac000 {

 compatible = "fsl,imx6ul-gpio", "fsl,imx35-gpio";

 reg = <0x020ac000 0x4000>;

 interrupts = <GIC\_SPI 74 IRQ\_TYPE\_LEVEL\_HIGH>,

 <GIC\_SPI 75 IRQ\_TYPE\_LEVEL\_HIGH>;

 gpio-controller;

 #gpio-cells = <2>;

 interrupt-controller;

 #interrupt-cells = <2>;

 };

 fxls8471@1e {

 compatible = "fsl,fxls8471";

 reg = <0x1e>;

 position = <0>;

 interrupt-parent = <&gpio5>;

 interrupts = <0 8>;

 };

 interrupt-parent指定父中断。interrupts第一个cells就是gpio编号，因此上面就是用的gpio5\_io00。

 通过函数 irq\_of\_parse\_and\_map从inttertupes属性获取中断号。

# 二、编写试验驱动

 1、修改设备树

 2、编写中断驱动

 3、按键消抖

# 三、测试