一、什么是中断

一种硬件上的通知机制，用来通知CPU发生了某种需要立即处理的事件

分为：

1. 内部中断 CPU执行程序的过程中，发生的一些硬件出错、运算出错事件（如分母为0、溢出等等），不可屏蔽
2. 外部中断 外设发生某种情况，通过一个引脚的高、低电平变化来通知CPU （如外设产生了数据、某种处理完毕等等）

二、中断处理原理

任何一种中断产生，CPU都会暂停当前执行的程序，跳转到内存固定位置执行一段程序，该程序被称为总的中断服务程序，在该程序中区分中断源，然后进一步调用该中断源对应的处理函数。

中断源对应的处理函数被称为分中断处理程序，一般每一个分中断处理程序对应一个外设产生的中断

写驱动时，如果外设有中断，则需要编写一个函数（分中断处理程序）来处理这种中断

三、中断接口

3.1 中断申请

```
int request_irq(unsigned int irq, irq_handler_t handler, unsigned long flags,const
char *name, void *dev)
/*
参数:
    irq: 所申请的中断号
    handler: 该中断号对应的中断处理函数
    flags: 中断触发方式或处理方式
        触发方式: IRQF_TRIGGER_NONE      //无触发
                  IRQF_TRIGGER_RISING    //上升沿触发
                  IRQF_TRIGGER_FALLING   //下降沿触发
                IRQF_TRIGGER_HIGH    //高电平触发
                IRQF_TRIGGER_LOW         //低电平触发
        处理方式:
               IRQF_DISABLED         //用于快速中断，处理中屏蔽所有中断
                IRQF_SHARED        //共享中断
        name: 中断名 /proc/interrupts
        dev: 传递给中断例程的参数，共享中断时用于区分那个设备，一般为对应设备的结构体地
址，无共享中断时写NULL
返回值: 成功: 0 失败: 错误码
*/
```

3.2 中断释放

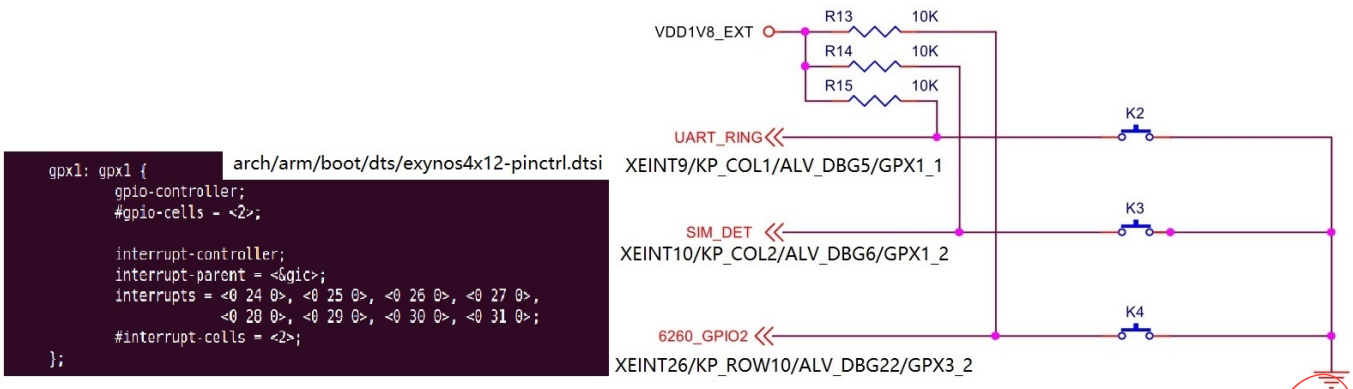
```
void free_irq(unsigned int irq, void *dev_id);
/*
功能：释放中断号
参数：
    irq：设备号
    dev_id：共享中断时用于区分那个设备一般强转成设备号，无共享中断时写NULL
*/
```

3.3 中断处理函数原型

```
typedef irqreturn_t (*irq_handler_t)(int, void *);
/*
参数：
    int：中断号
    void*:对应的申请中断时的dev_id
返回值：
    typedef enum irqreturn irqreturn_t; //中断返回值类型
    enum irqreturn {
        IRQ_NONE    = (0 << 0),
        IRQ_HANDLED = (1 << 0),
        IRQ_WAKE_THREAD = (1 << 1),
    };
    返回IRQ_HANDLED表示处理完了,返回IRQ_NONE在共享中断表示不处理
*/
```

四、按键驱动

按键原理图：



arch/arm/boot/dts/exynos4x12-pinctrl.dtsi

```
gpx1: gpx1 {
        gpio-controller;
        #gpio-cells = <2>;

        interrupt-controller;
        interrupt-parent = <&gic>;
        interrupts = <0 24 0>, <0 25 0>, <0 26 0>, <0 27 0>,
                     <0 28 0>, <0 29 0>, <0 30 0>, <0 31 0>;
        #interrupt-cells = <2>;
};
```

```
mykey2_node {
    compatible = "mykey2,key2";
    key2-gpio = <&gpx1 1 0>;
    interrupt-parent = <&gpx1>;
    interrupts = <1,3>;
};
```

32	64	–	EINT16_31	External Interrupt
31	63	–	EINT[15]	External Interrupt
30	62	–	EINT[14]	External Interrupt
29	61	–	EINT[13]	External Interrupt
28	60	–	EINT[12]	External Interrupt
27	59	–	EINT[11]	External Interrupt
26	58	–	EINT[10]	External Interrupt
25	57	–	EINT[9]	External Interrupt
24	56	–	EINT[8]	External Interrupt

exynos4412-fs4412.dts中增加节点

```
mykey2_node {
    compatible = "mykey2,key2";
    key2-gpio = <&gpx1 1 0>;
    interrupt-parent = <&gpx1>;
    interrupts = <1 3>;
};
```