Pinctrl和GPIO子系统的使用

1、设置pin的复用和电气属性

2、配置gpio的属性

- pinctrl子系统：借用pinctrl子系统来设置一个pin的复用和电气属性
    1. 获取设备树pin的信息
    2. 根据获取到的pin信息来设置pin的复用功能
    3. 根据获取到的pin信息来设置pin的电气特性，比如上、下拉、速度、驱动能力等。

    对于使用者来说，只需要在设备树里面设置好某个pin的相关属性即可，其他初始化工作均由pinctrl子系统来完成，pinctrl子系统目录为drivers/pinctrl
- **在设备树中如何添加一个pin的信息**

不同平台的格式不同。例如：rockship（瑞芯微）：

```
rockchip,pins =
    <4 17 RK_FUNC_GPIO &pcfg_pull_none>,// gpio4c0 4 17 pcfg_pull_none 不做上拉或者下拉
    <4 16 RK_FUNC_GPIO &pcfg_pull_none>;//gpio4c1 4 16
```

pinctrl驱动：

1. 如何找到系统对应的驱动：设备树里面的设备节点是通过compatiable属性列表，（驱动里面有一个描述驱动兼容性的），当设备树节点和驱动里面的兼容性匹配时候就拜师设备和驱动匹配了。我们只需要全局搜索comptible的字符串，找到含有这个字符串的.c文件
2. 当驱动和设备匹配之后，会执行probe函数

- gpio子系统：如果将gpio用作一个普通的io，就会用到gpio子系统

1. gpio在设备树中的表示方法：

```
pinctrl-names = "default", "rts_gpio";
        pinctrl-0 = <&uart0_rts>;//这个里面是rockchip,pins= <>信息
        pinctrl-1 = <&uart0_gpios>;
        BT,reset_gpio    = <&gpio2 RK_PD4 GPIO_ACTIVE_HIGH>;//因为 gpio-cells=<2>，1 是gp
        BT,wake_gpio     = <&gpio2 RK_PD2 GPIO_ACTIVE_HIGH>;
        BT,wake_host_irq = <&gpio0 RK_PA5 GPIO_ACTIVE_HIGH>;
        status = "okay";
```

可以参考设备树的绑定文档：kernel\Documentation\devicetree\bindings

例如：

```
/ {
    compatible = "rockchip,rk3399pro-toybrick-prod-linux","rockchip,rk3399pro";
    extbrd: extbrd {
        compatible = "prod-extboard";
```

```
        io-channels = <&saradc 0>, <&saradc 1>;
    };

    test_gpio{
        compatible = "alise,gpio_led";
        pinctrl-names = "default";
        pinctrl-0 = <&pingpio_led>;
        enable-gpios = <&gpio4 RK_PD2 GPIO_ACTIVE_HIGH>;
        status = "okay";
    };
};
&pinctrl{
    test{
        pingpio_led: pinctrl_gpio_test{
            rockchip,pins =
                <4 28 RK_FUNC_3 &pcfg_pull_none>;

        };
    };
};
```

2.驱动从设备树中获取gpio信息：of函数（参考与gpio有关的of函数）

①首先从设备树中获取gpio所在的设备节点：of_find_node_by_path()

②获取gpio：of_get_named_gpio函数，返回值就是GPIO编号。

③ 申请此gpio的管脚：int gpio_request();

④设置gpio驶入或者输出：int gpio_direction_input(unsigned gpio)；int gpio_direction_output(unsigned gpio, int value)；

⑤如果设置了gpio为输入，则使用这个函数读gpio_get_value，如果是输出：通过gpio_set_value设置输出值

⑥使用结束后释放此gpio

3.在driver/gpio目录下，gpio-xxx.c文件为具体芯片的驱动文件