密级状态：绝密（　　）　秘密（　　）　内部（ ）　公开（√ ）

RK3288_WiFi&BT_开发配置参考说明

（技术部，MID 组）

文件状态： [ ] 正在修改 [√] 正式发布	当前版本：	V1.0
	作　　者：	高伟龙
	完成日期：	2014-06-17
	审　　核：	
	完成日期：	2014-06-18

福州瑞芯微电子有限公司

Fuzhou Rockchips Semiconductor Co . , Ltd

版 本 历 史

版本号	作者	修改日期	修改说明	备注
V1.0	2014.06.17	高伟龙	初始版本	
V1.1	2014.08.13	胡卫国	增加 rkwifi, rtl8188eu, esp8089 兼容说明	
V1.2	2014.08.20	胡卫国	增加 rtl8723bs 说明	
V1.3	2014.09.15	高伟龙	增加 rtl8723au rtl8723bu 说明，对内核 wifi 配置做了适当修改。	
V1.4	2015.02.27	胡卫国	Android 5.0 不同 wifi bt 芯片兼容修改后的配置说明	
V1.5	2015.03.10	胡卫国	精减成 DTS 部分	

目 录

1 WiFi & BT DTS 配置说明

3.10 内核在开发方面跟 3.0.X 存在较大的差异，其中一个就是关于 DTS 部分的使用和配置。下面将以 RK3288 SDK 的 DTS 文件"arch/arm/boot/dts/rk3288-tb.dts"为例，分别针对 WiFi 和 BT 的 DTS 配置进行详细的说明，客户在实际开发过程中根据说明进行相关的 GPIO 或者功能设置。

1.1 WiFi DTS 配置说明

以 RK3288 为例

```
    wireless-wlan {
        compatible = "wlan-platdata";

        wifi_chip_type = "bcmwifi";

        sdio_vref = <1800>; //1800mv or 3300mv

        //keep_wifi_power_on;

        //power_ctrl_by_pmu;
        power_pmu_regulator = "act_ldo3";
        power_pmu_enable_level = <1>; //1->HIGH, 0->LOW

        //vref_ctrl_enable;
        //vref_ctrl_gpio = <&gpio0 GPIO_A2 GPIO_ACTIVE_HIGH>;
        vref_pmu_regulator = "act_ldo3";
        vref_pmu_enable_level = <1>; //1->HIGH, 0->LOW

        WIFI,poweren_gpio = <&gpio4 GPIO_D4 GPIO_ACTIVE_HIGH>;
        WIFI,host_wake_irq = <&gpio4 GPIO_D6 GPIO_ACTIVE_HIGH>;
        //WIFI,reset_gpio = <&gpio0 GPIO_A2 GPIO_ACTIVE_LOW>;

        status = "okay";
};
```

上面部分内容是 WiFi 的 DTS 配置内容，主要包括电源控制、中断等功能脚的配置。下面将对各个配置项（一般客户只需要修改下面红色标出部分参数）的功能进行详细描述：

wifi_chip_type = "bcmwifi";

用来确认 wifi 芯片型号，由于 Android 5.0 后 wifi android 部分修改成动态识别不同的 chip，所以这个配置存在差异。

可通过 kernel .config 中是否有以下定义来确认是否有 wifi android 部分动态识别修改：

CONFIG_RTL8723BS=m // 如果有这个定义，则有 wifi android 部分动态识别修改，否则没有。

具体配置如下

	有 wifi android 部分动态识别修改	没有 wifi android 部分动态识别修改
rk901	wifi_chip_type = "rk901";	wifi_chip_type = "bcmwifi";
rk903	wifi_chip_type = "rk903";	wifi_chip_type = "bcmwifi";
ap6181	wifi_chip_type = "ap6181";	wifi_chip_type = "bcmwifi";
ap6210	wifi_chip_type = "ap6210";	wifi_chip_type = "bcmwifi";
ap6234	wifi_chip_type = "ap6234";	wifi_chip_type = "bcmwifi";
ap6330	wifi_chip_type = "ap6330";	wifi_chip_type = "bcmwifi";
ap6335	wifi_chip_type = "ap6335";	wifi_chip_type = "bcmwifi";
ap6441	wifi_chip_type = "ap6441";	wifi_chip_type = "bcmwifi";
ap6476	wifi_chip_type = "ap6476";	wifi_chip_type = "bcmwifi";
rtl8188eu	wifi_chip_type = "rtl8188eu";	wifi_chip_type = "rtkwifi";
rtl8723au	wifi_chip_type = "rtl8723au";	wifi_chip_type = "rtkwifi";
rtl8723bu	wifi_chip_type = "rtl8723bu";	wifi_chip_type = "rtkwifi";
rtl8723bs	wifi_chip_type = "rtl8723bs";	wifi_chip_type = "rtkwifi";
rtl8723bs_vq0	wifi_chip_type = "rtl8723bs_vq0";	wifi_chip_type = "rtkwifi";
rtl8189es	wifi_chip_type = "rtl8189es";	wifi_chip_type = "rtkwifi";
rtl8812au	wifi_chip_type = "rtl8812au";	wifi_chip_type = "rtkwifi";
esp8089	wifi_chip_type = "esp8089";	wifi_chip_type = "esp8089";

sdio_vref = <1800>; //1800mv or 3300mv

这个配置项配置 WiFi 模组的 IO 参考电压值(目前只针对 RK3288 平台，RK312x 平台此参数无作用)，根据实际硬件设计中提供给 WiFi 模组参考电压输入的电压值来进行设定，以 AP6210 为例，模组的 IO 参考电压输入脚是第 22 脚 VDDIO，在配置 WiFi DTS 时需要量取这个脚的电压。假设是 1.8V，需要填入这个配置项为 1800。参考电压设置错误会导致 WiFi 通信异常引起 WiFi 打不开或者工作不稳定。

```
//keep_wifi_power_on;
```

这个配置项打开之后 wifi 模组将在内核启动过程中上电，并且一直保持上电状态，后续开关 wifi 时不对模组电源进行控制，针对一些有此需求的 wifi 模组比如 rtl8723au 和 rtl8723bu 因为 wifi 和 bt 共用电源因此一般情况下在内核启动时直接对其上电并一直保持上电状态，以保证 wifi 和 bt 都能正常工作。此配置项默认情况下是关闭的。

```
//power_ctrl_by_pmu;
power_pmu_regulator = "act_ldo3";
power_pmu_enable_level = <1>; //1->HIGH, 0->LOW
```

这个部分的配置项是关于 WiFi 电源控制，通常情况下 WiFi 电源控制是由主控的 GPIO 来进行，但是也有部分产品设计实用 PMU 来作为 WiFi 电源或者模拟 GPIO 来控制 WiFi 模组供电，如果是这种情况那么需要将“power_ctrl_by_pmu”打开，并在“power_pmu_regulator”中填入对应的 PMU 输出口，以及有效电平“power_pmu_enable_level”。

```
//vref_ctrl_enable;
//vref_ctrl_gpio = <&gpio0 GPIO_A2 GPIO_ACTIVE_HIGH>;
vref_pmu_regulator = "act_ldo3";
vref_pmu_enable_level = <1>; //1->HIGH, 0->LOW
```

这个部分的配置是对一些可控制 IO 参考电压开关的项目提供参考电影开关的设置。默认情况下是关闭的，具体功能是实现在模组不工作（模组上的所有功能芯片都不工作）的情况下切断 IO 参考电压输出，以此来降低一定的系统功耗。这个是可选配置并非所有硬件都支持，也不是所有 WiFi 模组都支持。

```
WIFI,poweren_gpio = <&gpio4 GPIO_D4 GPIO_ACTIVE_HIGH>;
```

这个是最常用，或者一定会用到的配置项，也就是 WiFi 的电源控制 GPIO 口设定的位置，客户需要根据硬件原理图确认 WiFi 电源控制脚接到主控的哪个 GPIO 口，然后填入这个地方，以 3288 SDK 板为例，使用 AP6335，WiFi 电源控制是第 12 脚“WL_REG_ON”接到主控的 GPIO4D4，高电平有效，因此如上所述方式填入。

rfkill-wlan.c 驱动中对这个脚的控制逻辑为：在开机 probe 时会去 disable，在打开 wifi

时会去 enable，关闭 wifi 时会 disable。休眠唤醒时不做控制。

WIFI,host_wake_irq = <&gpio4 GPIO_D6 GPIO_ACTIVE_HIGH>;

这个配置项是 WiFi 中断脚的配置，某些 WiFi 模组没有这个脚可以不用配置直接将此配置项注释掉。使用 Broadcom 的 WiFi 比如 AP6xxx 以及 RK90x 等模组都需要正确配置这 GPIO。

Broadcom wifi AP6xxx 系统会使用此中断脚作为 wifi 数据中断脚，此中断脚有异常将会导致 WiFi 无法正常工作。其它 WiFi，例如 RTL8723BS，在机器进入休眠时，如果有 WiFi 数据到来时此中断用来唤醒机器。此中断脚有异常并不会造成 WiFi 无法正常工作。

//WIFI,reset_gpio = <&gpio0 GPIO_A2 GPIO_ACTIVE_LOW>;

这个配置项是预留的，某些 WiFi 模组具备 RESET 脚，需要配置。目前 Broadcom 的 WiFi 暂时都没有此功能脚，因此默认是注释掉的。

rfkill-wlan.c 驱动中对这个脚的控制逻辑为：在开机 probe 时并不控制，在打开 wifi 时会去 enable，关闭 wifi 时会 disable。休眠唤醒时不做控制。

1.2 BT DTS 配置说明

以 RK3288 为例

```
    wireless-bluetooth {
        compatible = "bluetooth-platdata";

        //wifi-bt-power-toggle;

        uart_rts_gpios = <&gpio4 GPIO_C3 GPIO_ACTIVE_LOW>;
        pinctrl-names = "default","rts_gpio";
        pinctrl-0 = <&uart0_rts>;
        pinctrl-1 = <&uart0_rts_gpio>;

        BT,power_gpio = <&gpio4 GPIO_D3 GPIO_ACTIVE_HIGH>;
        BT,reset_gpio = <&gpio4 GPIO_D5 GPIO_ACTIVE_HIGH>;
        BT,wake_gpio = <&gpio4 GPIO_D2 GPIO_ACTIVE_HIGH>;
        BT,wake_host_irq = <&gpio4 GPIO_D7 GPIO_ACTIVE_LOW>;

        status = "okay";
};
```

上面部分内容是 BT 的 DTS 配置内容，主要包括电源控制、中断等功能脚的配置。下面将对各个配置项的功能进行详细描述：

//wifi-bt-power-toggle;

这个配置项关于 WiFi 和 BT 共用一个电源控制的情况，例如 RealTek 的 RTL8723BU，WiFi 和 BT 电源控制是同一个，而 Broadcom 的模组 WiFi 和 BT 的电源控制是分开的，因此默认这个配置是注释掉的。客户根据实际使用的模组进行配置。

uart_rts_gpios = <&gpio4 GPIO_C3 GPIO_ACTIVE_LOW>;

pinctrl-names = "default","rts_gpio";

pinctrl-0 = <&uart0_rts>;

pinctrl-1 = <&uart0_rts_gpio>;

这个部分的配置是关于 BT 的 uart 口 rts 脚的配置，部分型号的 BT 在操作过程中需要对 uart 的 RTS 脚进行控制。例如 AP6210、AP6335 等。因此在使用这一类的 BT 模组时需要配置这个部分。

BT,power_gpio = <&gpio4 GPIO_D3 GPIO_ACTIVE_HIGH>;

这个配置项是关于 BT 的电源控制 GPIO 配置，以 3288 SDK 板为例，AP6335 BT 的电源控制口是第 34 脚 BT_REG_ON 连接到的主控 GPIO 口是 GPIO4D3，高电平有效。

rfkill-bt.c 驱动中对这个脚的控制逻辑为：在开机 probe 时会去 disable，在打开 bt 时会去 enable，关闭 bt 时会 disable。休眠唤醒时不做控制。

BT,reset_gpio = <&gpio4 GPIO_D5 GPIO_ACTIVE_HIGH>;

这个配置项是关于 BT 的 RESET 脚配置，这个脚不同的 BT 模组不一定都有，具体以实际原理图为准。

rfkill-bt.c 驱动中对这个脚的控制逻辑为：在开机 probe 时会去 disable，在打开 bt 时会去 enable，关闭 bt 时会 disable。休眠唤醒时不做控制。

BT,wake_gpio = <&gpio4 GPIO_D2 GPIO_ACTIVE_HIGH>;

这个配置项是关于 BT 的 WAKE 脚配置，以 3288 SDK 板为例，AP6335 BT 的 WAKE 脚是第 6 脚 BT_WAKE 连接到的主控 GPIO 口是 GPIO4D2，高电平有效。

rfkill-bt.c 驱动中对这个脚的控制逻辑为：在 bt 空闭时，通过 disable 它来让 bt 进入低功耗模式。

BT,wake_host_irq = <&gpio4 GPIO_D7 GPIO_ACTIVE_LOW>;

这个配置项是关于 BT 的中断脚配置，以 3288 SDK 板为例，AP6335 BT 的中断脚是第 6 脚 BT_WAKE 连接到的主控 GPIO 口是 GPIO4D2，高电平有效。

rfkill-bt.c 驱动中对这个脚的控制逻辑为：在机器进入休眠时，如果有 bt 数据到来时此中断用来唤醒机器。

2 SDIO 相关配置

WiFi 一般使用 SDIO 接口，所以需要注意 sdio 相关配置：

RK3288：(在板级 dts 中配置)

```
&sdio {
        clock-frequency = <50000000>;
        clock-freq-min-max = <200000 50000000>;
        supports-highspeed;
        supports-sdio;
        ignore-pm-notify;
        keep-power-in-suspend;
        //cap-sdio-irq;
        status = "okay";
};
```

RK312x: （在 rk312x-sdk.dtsi 中配置）

```
 &sdio {
        clock-frequency = <37500000>;
        clock-freq-min-max = <200000 37500000>;
        supports-highspeed;
        supports-sdio;
        ignore-pm-notify;
        keep-power-in-suspend;
        cap-sdio-irq;
        status = "okay";
};
```

3 UART 相关配置

BT 一般使用 UART 接口，需要在板级 dts 中作相应配置：

RK3288：

```
 &uart_bt {
        status = "okay";
        dma-names = "!tx", "!rx";
        pinctrl-0 = <&uart0_xfer &uart0_cts>;
};
```

RK3128 BOX: (默认使用 uart1)

```
&uart1{
        status = "okay";
        dma-names = "!tx", "!rx";
        pinctrl-0 = <&uart1_xfer &uart1_cts>;
};
```

RK3128 MID: (默认使用 uart0)

```
&uart0 {
        status = "okay";
        dma-names = "!tx", "!rx";
        pinctrl-0 = <&uart0_xfer &uart0_cts>;
};
```