RK3399 USB DTS 配置说明

发布版本：1.2

作者邮箱：wulf@rock-chips.com

日期：2019-06-25

文档密级：公开资料

概述

本文档提供RK3399 USB DTS的配置方法。RK3399支持两个Type-C USB 3.0(Type-C PHY is a combination of USB 3.0 SuperSpeed PHY and DisplayPort Transmit PHY)和两个USB 2.0 Host。其中，两个Type-C USB 3.0控制器硬件都可以支持OTG(USB Peripheral和USB Host)，并且向下兼容USB2.0/1.1/1.0。此外，Type-C USB 3.0可以根据实际的应用需求，将物理接口简化设计为Type-A USB 3.0/2.0，Micro USB 3.0/2.0等多种接口类型，内核USB驱动已经兼容这几种不同类型的USB接口，只需要根据实际的硬件设计修改对应的板级DTS配置，就可以使能相应的USB接口。

产品版本

芯片名称	内核版本
RK3399	Linux4.4

读者对象 本文档（本指南）主要适用于以下工程师： 软件工程师 技术支持工程师

修订记录

日期	版本	作者	修改说明
2018-03-01	V1.0	吴良峰	初始版本
2019-01-09	V1.1	吴良峰	使用markdownlint修订格式
2019-06-25	V1.2	吴良峰	1. 增加Type-C to Type-A USB 2.0说明 2. 增加VBUS供电说明 3. 更新文档目录名称 4. 参考示例由EVB改为Sapphire Excavator Board 5. 修订一些错误

1 Type-C0/1 USB 3.0 DTS

Type-C 的接口类型如下图1-1所示。

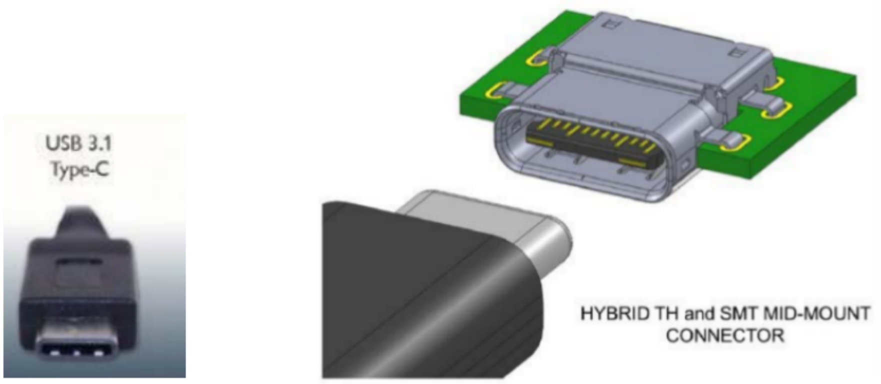

图1-1 Type-C 接口类型示意图

RK3399 SoC内部4个USB控制器与USB PHY的连接如下图1-2所示。

其中，DP是指Display Port控制器，DP与USB 3.0共用Type-C PHY。如图1-2所示，一个完整的Type-C功能，是由Type-C USB 3.0 PHY & DP PHY和USB 2.0 OTG PHY两部分组成的，这两部分PHY在芯片内部的硬件模块是独立的，供电也是独立的。

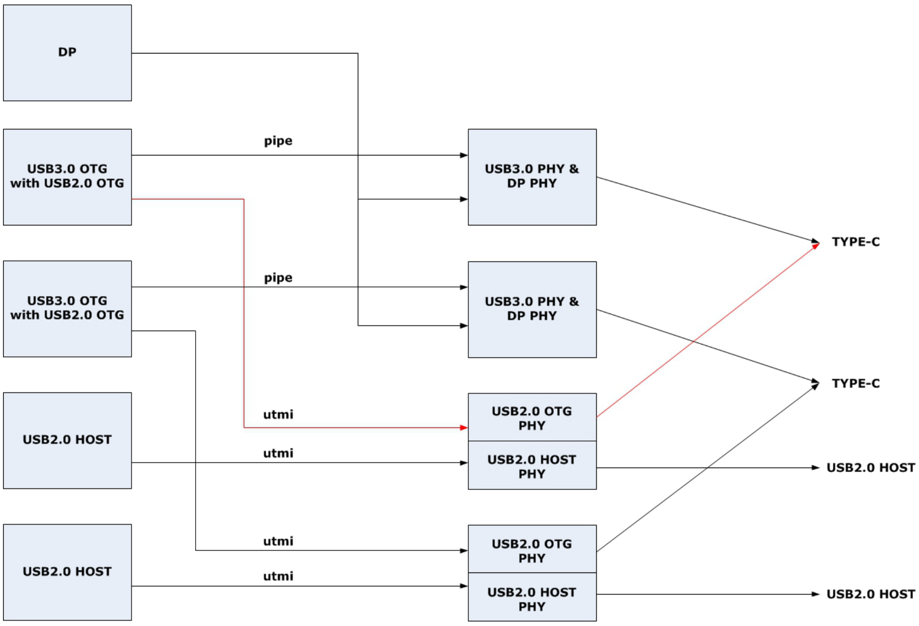

图1-2 RK3399 USB控制器&PHY连接示意图

RK3399 SDK DTS的默认配置，支持Type-C0 USB 3.0 OTG功能，Type-C1 USB 3.0 Host功能。DTS的配置主要包括DWC3控制器、Type-C USB 3.0 PHY以及USB 2.0 PHY。

1.1 Type-C0 /1 USB Controller DTS

Type-C0/1 USB控制器硬件都支持USB 3.0 OTG（USB Peripheral和USB Host）功能，并且向下兼容USB 2.0/1.1/1.0。但由于当前内核的USB 框架只支持一个USB 口作为Peripheral功能，所以SDK默认配置Type-C0支持OTG mode，而Type-C1仅支持Host mode。如果要配置Type-C1支持OTG mode，请参考：

1.3 Type-C1 USB OTG Mode DTS

以RK3399 Sapphire Excavator Board 的 Type-C0/C1 USB 3.0 控制器DTS配置为例：

`arch/arm64/boot/dts/rockchip/rk3399.dtsi`

```
usbdrd3_0: usb0 { /* Type-C0 USB3.0 控制器DTS配置*/
                compatible = "rockchip,rk3399-dwc3";
                clocks = <&cru SCLK_USB3OTG0_REF>, <&cru SCLK_USB3OTG0_SUSPEND>,
                         <&cru ACLK_USB3OTG0>, <&cru ACLK_USB3_GRF>;
                clock-names = "ref_clk", "suspend_clk",
                              "bus_clk", "grf_clk";
                power-domains = <&power RK3399_PD_USB3>;
                resets = <&cru SRST_A_USB3_OTG0>;
                reset-names = "usb3-otg"; /* USB0 控制器的 reset */
                #address-cells = <2>;
                #size-cells = <2>;
                ranges;
                status = "disabled";
                usbdrd_dwc3_0: dwc3@fe800000 {
                        compatible = "snps,dwc3";
                        reg = <0x0 0xfe800000 0x0 0x100000>;
                        interrupts = <GIC_SPI 105 IRQ_TYPE_LEVEL_HIGH 0>;
                        dr_mode = "otg"; /* 支持OTG mode */
                        phys = <&u2phy0_otg>, <&tcphy0_usb3>; /* usb3/2 phy属性 */
                        phy-names = "usb2-phy", "usb3-phy";
                        phy_type = "utmi_wide";
                        snps,dis_enblslpm_quirk;
                        snps,dis-u2-freeclk-exists-quirk;
                        snps,dis_u2_susphy_quirk;
                        snps,dis-del-phy-power-chg-quirk;
                        snps,tx-ipgap-linecheck-dis-quirk;
                        snps,xhci-slow-suspend-quirk;
                        snps,xhci-trb-ent-quirk;
                        snps,usb3-warm-reset-on-resume-quirk;
                        status = "disabled";
                };
        };

usbdrd3_1: usb1 { /* Type-C1 USB3.0 控制器DTS配置*/
                compatible = "rockchip,rk3399-dwc3";
                clocks = <&cru SCLK_USB3OTG1_REF>, <&cru SCLK_USB3OTG1_SUSPEND>,
                         <&cru ACLK_USB3OTG1>, <&cru ACLK_USB3_GRF>;
                clock-names = "ref_clk", "suspend_clk",
                              "bus_clk", "grf_clk";
                power-domains = <&power RK3399_PD_USB3>;
                resets = <&cru SRST_A_USB3_OTG1>;
                reset-names = "usb3-otg"; /* USB1 控制器的 reset */
                #address-cells = <2>;
                #size-cells = <2>;
                ranges;
                status = "disabled";
                usbdrd_dwc3_1: dwc3@fe900000 {
                        compatible = "snps,dwc3";
                        reg = <0x0 0xfe900000 0x0 0x100000>;
                        interrupts = <GIC_SPI 110 IRQ_TYPE_LEVEL_HIGH 0>;
                        dr_mode = "host"; /* 只支持Host mode */
                        phys = <&u2phy1_otg>, <&tcphy1_usb3>; /* usb3/2 phy属性 */
                        phy-names = "usb2-phy", "usb3-phy";
                        phy_type = "utmi_wide";
                        snps,dis_enblslpm_quirk;
                        snps,dis-u2-freeclk-exists-quirk;
                        snps,dis_u2_susphy_quirk;
                        snps,dis-del-phy-power-chg-quirk;
                        snps,tx-ipgap-linecheck-dis-quirk;
                        snps,xhci-slow-suspend-quirk;
                        snps,xhci-trb-ent-quirk;
                        snps,usb3-warm-reset-on-resume-quirk;
                        status = "disabled";
                };
        };
```

`arch/arm64/boot/dts/rockchip/rk3399-sapphire.dtsi`

```
&usbdrd3_0 {
        extcon = <&fusb0>; /* 配置extcon属性，用于接收fusb302驱动的 UFP/DFP notifier*/
        status = "okay";
};

&usbdrd3_1 {
        /* USB1 为Type-A接口，只支持USB Host，不用配置extcon属性 */
        status = "okay";
};

&usbdrd_dwc3_0 {
        status = "okay";
};

&usbdrd_dwc3_1 {
        dr_mode = "host"; /* 配置USB1 为Host only mode */
        status = "okay";
};
```

1.2 Type-C0 /1 USB PHY DTS

Type-C0/1 USB PHY的硬件由USB 3.0 PHY（只支持Super-speed）和USB 2.0 PHY（支持High-speed/Full-speed/Low-speed）两部分组成。所以，对应的USB PHY DTS也包括USB 3.0 PHY和USB 2.0 PHY两部分。

1.2.1 Type-C0 /1 USB 3.0 PHY DTS

以RK3399 Sapphire Excavator Board Type-C0 /1 USB 3.0 PHY DTS配置为例：

`arch/arm64/boot/dts/rockchip/rk3399.dtsi`

```
tcphy0: phy@ff7c0000 {
                compatible = "rockchip,rk3399-typec-phy";
                reg = <0x0 0xff7c0000 0x0 0x40000>;
                #phy-cells = <1>;
                clocks = <&cru SCLK_UPHY0_TCPDCORE>,
                         <&cru SCLK_UPHY0_TCPDPHY_REF>;
                clock-names = "tcpdcore", "tcpdphy-ref";
                assigned-clocks = <&cru SCLK_UPHY0_TCPDCORE>;
                assigned-clock-rates = <50000000>;
                power-domains = <&power RK3399_PD_TCPD0>;
                resets = <&cru SRST_UPHY0>,
                         <&cru SRST_UPHY0_PIPE_L00>,
                         <&cru SRST_P_UPHY0_TCPHY>;
                reset-names = "uphy", "uphy-pipe", "uphy-tcphy";
                rockchip,grf = <&grf>;
                rockchip,typec-conn-dir = <0xe580 0 16>;
                rockchip,usb3tousb2-en = <0xe580 3 19>;
                rockchip,usb3-host-disable = <0x2434 0 16>;
                rockchip,usb3-host-port = <0x2434 12 28>;
                rockchip,external-psm = <0xe588 14 30>;
                rockchip,pipe-status = <0xe5c0 0 0>;
                rockchip,uphy-dp-sel = <0x6268 19 19>;
                status = "disabled";

                tcphy0_dp: dp-port {
                        #phy-cells = <0>;
                };

                tcphy0_usb3: usb3-port { /* Type-C0 USB3.0 port */
                        #phy-cells = <0>;
                };
        };

tcphy1: phy@ff800000 {
                compatible = "rockchip,rk3399-typec-phy";
                reg = <0x0 0xff800000 0x0 0x40000>;
                #phy-cells = <1>;
                clocks = <&cru SCLK_UPHY1_TCPDCORE>,
                         <&cru SCLK_UPHY1_TCPDPHY_REF>;
                clock-names = "tcpdcore", "tcpdphy-ref";
                assigned-clocks = <&cru SCLK_UPHY1_TCPDCORE>;
                assigned-clock-rates = <50000000>;
                power-domains = <&power RK3399_PD_TCPD1>;
                resets = <&cru SRST_UPHY1>,
                         <&cru SRST_UPHY1_PIPE_L00>,
                         <&cru SRST_P_UPHY1_TCPHY>;
                reset-names = "uphy", "uphy-pipe", "uphy-tcphy";
                rockchip,grf = <&grf>;
                rockchip,typec-conn-dir = <0xe58c 0 16>;
                rockchip,usb3tousb2-en = <0xe58c 3 19>;
                rockchip,usb3-host-disable = <0x2444 0 16>;
                rockchip,usb3-host-port = <0x2444 12 28>;
                rockchip,external-psm = <0xe594 14 30>;
                rockchip,pipe-status = <0xe5c0 16 16>;
                rockchip,uphy-dp-sel = <0x6268 3 19>;
                status = "disabled";

                tcphy1_dp: dp-port {
                        #phy-cells = <0>;
                };

                tcphy1_usb3: usb3-port { /* Type-C1 USB3.0 port */
                        #phy-cells = <0>;
                };
        };
```

`arch/arm64/boot/dts/rockchip/rk3399-sapphire.dtsi`

```
&tcphy0 {
        extcon = <&fusb0>;
        status = "okay";
};

&tcphy1 {
        /* Type-C1 使用的是Type-A USB接口，不用配置extcon属性 */
        status = "okay";
};

&pinctrl {
        ......
        fusb30x {
                fusb0_int: fusb0-int { /* 配置TypeC0 fusb302 中断 */
                        rockchip,pins = <1 2 RK_FUNC_GPIO &pcfg_pull_up>;
                };
        };
};

&i2c4 { /* 配置fusb302芯片的i2c */
        status = "okay";
        i2c-scl-rising-time-ns = <475>;
        i2c-scl-falling-time-ns = <26>;

        fusb0: fusb30x@22 {
                compatible = "fairchild,fusb302";
                reg = <0x22>;
                pinctrl-names = "default";
                pinctrl-0 = <&fusb0_int>;
                int-n-gpios = <&gpio1 2 GPIO_ACTIVE_HIGH>;
                vbus-5v-gpios = <&gpio2 0 GPIO_ACTIVE_HIGH>;
                status = "okay";
        };
};
```

1.2.2 Type-C0 /1 USB 2.0 PHY DTS

RK3399 有两个 USB 2.0 combphy（一个PHY支持两个port，一个port连接OTG，连接port连接Host），本文档称之为USB 2.0 PHY0和PHY1（参考图1-2）。其中，PHY0的port0作为Type-C0 USB的USB 2.0 PHY，PHY1的port0作为Type-C1 USB的USB 2.0 PHY。

以RK3399 Sapphire Excavator Board Type-C0 /1 USB2.0 PHY DTS配置为例：

`arch/arm64/boot/dts/rockchip/rk3399.dtsi`

```
grf: syscon@ff770000 {
		compatible = "rockchip,rk3399-grf", "syscon", "simple-mfd";
		......
		u2phy0: usb2-phy@e450 {
			compatible = "rockchip,rk3399-usb2phy";
			reg = <0xe450 0x10>;
			clocks = <&cru SCLK_USB2PHY0_REF>;
			clock-names = "phyclk";
			#clock-cells = <0>;
			clock-output-names = "clk_usbphy0_480m";
			status = "disabled";

			u2phy0_otg: otg-port { /* 配置Type-C0 USB2.0 PHY0 port0 */
				#phy-cells = <0>;
				interrupts = <GIC_SPI 103 IRQ_TYPE_LEVEL_HIGH 0>,
					     <GIC_SPI 104 IRQ_TYPE_LEVEL_HIGH 0>,
					     <GIC_SPI 106 IRQ_TYPE_LEVEL_HIGH 0>;
				interrupt-names = "otg-bvalid", "otg-id",
						  "linestate";
				status = "disabled";
			};

			......
		};

		u2phy1: usb2-phy@e460 { /* 配置Type-C1 USB2.0 PHY1 port0 */
			compatible = "rockchip,rk3399-usb2phy";
			reg = <0xe460 0x10>;
			clocks = <&cru SCLK_USB2PHY1_REF>;
			clock-names = "phyclk";
			#clock-cells = <0>;
			clock-output-names = "clk_usbphy1_480m";
			status = "disabled";

			u2phy1_otg: otg-port { /* Type-C1 USB2.0 PHY1 port0*/
```

```
                                #phy-cells = <0>;
                                interrupts = <GIC_SPI 108 IRQ_TYPE_LEVEL_HIGH 0>,
                                             <GIC_SPI 109 IRQ_TYPE_LEVEL_HIGH 0>,
                                             <GIC_SPI 111 IRQ_TYPE_LEVEL_HIGH 0>;
                                interrupt-names = "otg-bvalid", "otg-id",
                                                  "linestate";
                                status = "disabled";
                        };

                        ......
                };
```

`arch/arm64/boot/dts/rockchip/rk3399-sapphire.dtsi`

```
&u2phy0 {
        status = "okay";
        extcon = <&fusb0>; /* extcon 属性*/

        u2phy0_otg: otg-port {
                status = "okay";
        };
        ......
};

&u2phy1 {
        status = "okay";
        /* u2phy1 只支持USB Host，不用配置 extcon 属性 */

        u2phy1_otg: otg-port {
                status = "okay";
        };
        ......
};

```

1.3 Type-C1 USB OTG Mode DTS

在1.1 Type-C0 /1 USB Controller DTS中已经提到，由于当前的内核USB框架只能支持一个USB 口作为Peripheral功能，所以RK3399 SDK默认配置Type-C0作为OTG mode 支持USB Peripheral功能，而Type-C1只支持Host mode。实际产品中，可以根据应用需求，配置Type-C1为OTG mode，支持USB Peripheral功能，需要修改的地方有两个：

- DTS的“dr_mode”属性

```
&usbdrd_dwc3_1 {
        status = "okay";
        dr_mode = "otg";  /* 配置Type-C1 USB控制器为OTG mode */
        extcon = <&fusb1>; /* 注意：extcon 属性要根据实际的硬件电路设计来配置 */
};
```

- init.rk30board.usb.rc 的USB控制器地址 （适用于Android平台）

  设置USB控制器的地址为Type-C1 USB控制器的基地址：

  `setprop sys.usb.controller "fe900000.dwc3"`

2 Type-C to Type-A USB 3.0 Host DTS

Type-A USB 3.0的接口类型如下图2-1所示。

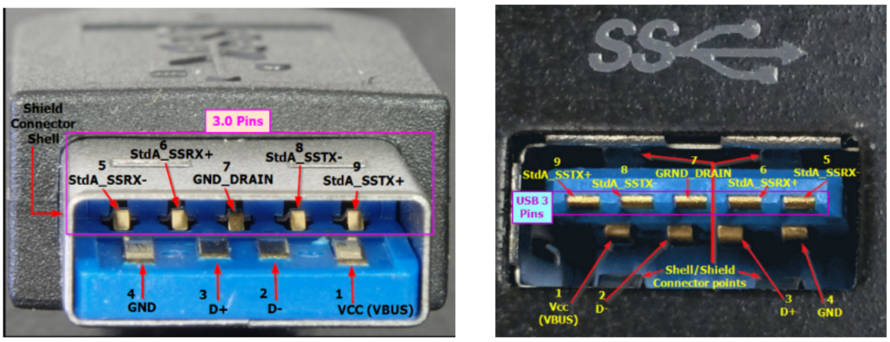

图2-1 Type-A USB3.0接口类型示意图

Type-C USB可以配置为Type-A USB使用。如RK3399 Sapphire Excavator Board平台的Type-C1 USB默认设计为Type-A USB 3.0 Host。这种设计，USB Vbus 5V一般为常供电，不需要单独的GPIO控制，也不需要fusb302芯片，但Type-C的三路供电需要正常开启，如下图2-2所示，才能支持USB 3.0 Super-speed。

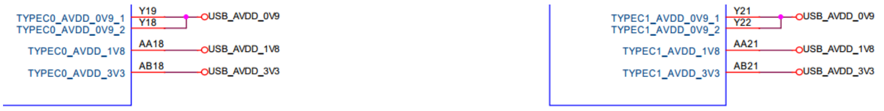

图2-2 Type-C 供电电路

Type-A USB3.0 Host DTS配置的注意点如下：

- 对应的fusb节点不要配置，因为Type-A USB3.0不需要fusb302芯片
- 对应的USB控制器父节点（usbdrd3）和PHY的节点（tcphy和u2phy）都要删除extcon属性
- 对应的USB控制器子节点（usbdrd_dwc3）的dr_mode属性要配置为"host"

以RK3399 Sapphire Excavator Board 平台为例（Type-C0 配置为Type-C接口，Type-C1配置为Type-A USB 3.0 接口），Type-A USB 3.0 Host DTS对应的配置方法如下：

`arch/arm64/boot/dts/rockchip/rk3399-sapphire.dtsi`

```
/* Enable Type-C1 USB 3.0 PHY */
&tcphy1 {
        /* Type-C1 使用的是Type-A USB接口，不用配置extcon属性 */
        status = "okay";
};

/* Enable Type-C1 USB 2.0 PHY */
&u2phy1 {
        status = "okay";
        /* u2phy1 只支持USB Host，不用配置 extcon 属性 */

        u2phy1_otg: otg-port {
                status = "okay";
        };
        ......
};

/* Configurate and Enable Type-C1 USB 3.0 Controller */
&usbdrd3_1 {
        status = "okay";
};

&usbdrd_dwc3_1 {
        /* 配置dr_mode为host，表示只支持Host only mode，并且不用配置 extcon 属性 */
        dr_mode = "host";
        status = "okay";
};
```

3 Type-C to Micro USB 3.0 OTG Mode DTS

Micro USB 3.0 OTG的接口类型如下图3-1所示。

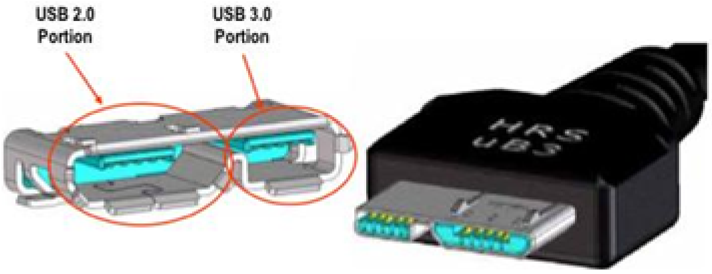

图3-1 Micro USB3.0 OTG接口类型示意图

为了节省硬件成本，Type-C USB可以配置为Micro USB 3.0 OTG使用。这种设计，硬件上不需要fusb302芯片，USB Vbus 5V一般由GPIO控制，Type-C的三路供电与2 Type-C to Type-A USB 3.0 Host DTS的硬件电路一样，需要正常开启。

Micro USB3.0 OTG DTS配置的注意点如下：

- 对应的fusb节点不要配置，因为Micro USB3.0不需要fusb302芯片
- 对应的USB PHY节点（tcphy和u2phy）都要删除extcon属性
- 对应的USB控制器父节点（usbdrd3）中，extcon属性引用为u2phy的节点
- 对应的USB控制器子节点（usbdrd_dwc3）的dr_mode属性要配置为"otg"
- 对应的USB2 PHY节点（u2phy）中，配置Vbus regulator
- Micro USB 3.0 OTG 是根据ID脚的电平变化（与Micro USB 2.0 OTG相同）来切换Peripheral mode和Host mode

以Type-C0 USB配置为Micro USB 3.0 OTG为例：

```
/* Enable Type-C0 USB 3.0 PHY */
&tcphy0 {
        /* Type-C0 使用的是Micro USB 3.0接口，不用配置extcon属性 */
        status = "okay";
};

/* Enable Type-C0 USB 2.0 PHY */
&u2phy0 {
        /* USB 2.0 PHY 不用配置extcon属性 */
        status = "okay";
        otg-vbus-gpios = <&gpio3 RK_PC6 GPIO_ACTIVE_HIGH>; /* Vbus GPIO配置，见Note1 */

        u2phy1_otg: otg-port {
                status = "okay";
        };
        ......
};

/* Configurate and Enable Type-C0 USB 3.0 Controller */
&usbdrd3_0 {
        /* USB控制器的extcon属性必须引用u2phy0，才能支持Peripheral mode和Host mode切换 */
        extcon = <&u2phy0>;
        status = "okay";
};

&usbdrd_dwc3_0 {
        /* USB控制器的dr_mode必须配置为otg */
        dr_mode = "otg";
        status = "okay";
};
```

Note1.

Kernel 4.4最新的代码，已经将OTG USB Vbus的控制改为regulator的方式，对应的提交信息如下：

commit a1ca1be8f6ed “phy: rockchip-inno-usb2: use fixed-regulator for vbus power”

参考文档：

Documentation/devicetree/bindings/phy/phy-rockchip-inno-usb2.txt

DTS中对USB Vbus的控制，应该改为：

```
vcc_otg_vbus: otg-vbus-regulator {
                compatible = "regulator-fixed";
                gpio = <&gpio3 RK_PC6 GPIO_ACTIVE_HIGH>;
                pinctrl-names = "default";
                pinctrl-0 = <&otg_vbus_drv>;
                regulator-name = "vcc_otg_vbus";
                regulator-min-microvolt = <5000000>;
                regulator-max-microvolt = <5000000>;
                enable-active-high;
        };

&pinctrl {
        ......
        usb {
                otg_vbus_drv: otg-vbus-drv {
                        rockchip,pins = <3 RK_PC6 RK_FUNC_GPIO &pcfg_pull_none>;
                };
        };
};

&u2phy0 {
        status = "okay";

        u2phy0_otg: otg-port {
                vbus-supply = <&vcc_otg_vbus>; /*配置Vbus regulator属性 */
                status = "okay";
        };
        ......
};
```

4 Type-C to Micro USB 2.0 OTG Mode DTS

Micro USB 2.0 OTG的接口类型如下图4-1所示。

图4-1 Micro USB2.0 OTG接口类型示意图

为了节省硬件成本，Type-C USB可以配置为Micro USB 2.0 OTG使用。这种设计，硬件上不需要fusb302芯片，USB Vbus 5V一般由GPIO控制，因为不需要支持USB3.0，所以对应的Type-C三路供电（USB_AVDD_0V9，USB_AVDD_1V8，USB_AVDD_3V3）可以关闭。

Micro USB2.0 OTG DTS配置的注意点如下:

- 对应的fusb节点不要配置，因为Micro USB2.0不需要fusb302芯片
- Disable对应的USB3 PHY节点（tcphy）
- 对应的USB2 PHY节点（u2phy）要删除extcon属性，并且配置Vbus regulator
- 对应的USB控制器父节点（usbdrd3）中，extcon属性引用为u2phy
- 对应的USB控制器子节点（usbdrd_dwc3）的dr_mode属性要配置为"otg"，maximum-speed 属性配置为high-speed，phys 属性只引用USB2 PHY节点

以Type-C0 USB配置为Micro USB2.0 OTG为例：

```
/* Disable Type-C0 USB 3.0 PHY */
&tcphy0 {
        status = "disabled";
};

/* Enable Type-C0 USB 2.0 PHY */
&u2phy0 {
        /* USB 2.0 PHY 不用配置extcon属性 */
        status = "okay";
        otg-vbus-gpios = <&gpio3 RK_PC6 GPIO_ACTIVE_HIGH>; /* Vbus GPIO配置，见Note1 */

        u2phy1_otg: otg-port {
                status = "okay";
        };
        ......
};

/* Configurate and Enable Type-C0 USB 3.0 Controller */
&usbdrd3_0 {
        /* USB控制器的extcon属性必须引用u2phy0，才能支持Peripheral mode和Host mode切换 */
        extcon = <&u2phy0>;
        status = "okay";
};

&usbdrd_dwc3_0 {
        dr_mode = "otg"; /* USB控制器的dr_mode配置为otg */
        maximum-speed = “high-speed”; /* maximum-speed 属性配置为high-speed */
        phys = <&u2phy0_otg>; /* phys 属性只引用USB2 PHY节点 */
        phy-names = "usb2-phy";
        status = "okay";
};
```

Note1.

Kernel 4.4最新的代码，已经将OTG USB Vbus的控制改为regulator的方式（commit a1ca1be8f6ed “phy: rockchip-inno-usb2: use fixed-regulator for vbus power”），参考文档：

Documentation/devicetree/bindings/phy/phy-rockchip-inno-usb2.txt

所以，DTS中对OTG USB Vbus的控制，请参考3 Type-C to Micro USB 3.0 OTG Mode DTS中Vbus regulator的配置方法。

5 Type-C to Type-A USB 2.0

Type-C to Type-A USB 2.0的硬件设计方案（ ID脚悬空），可以细化为三种不同的实现形式，分别是：

1. Type-C to Type-A USB 2.0 OTG mode DTS
2. Type-C to Type-A USB 2.0 Host only mode DTS
3. Type-C to Type-A USB 2.0 OTG mode DTS and Support DP 4 Lane

以下章节详细说明上述三种Type-C to Type-A USB 2.0方案的DTS配置。

5.1 Type-C to Type-A USB 2.0 OTG mode DTS

该方案的特点是，支持USB 2.0 OTG功能，Vbus为常供电或者通过GPIO/PMIC控制。系统启功后，需要应用层通过内核的接口设置USB控制器工作于Host mode。Type-C的三路供电（见图2-2）可以关闭。

以Type-C0 USB配置为Type-C to Type-A USB 2.0 OTG mode为例，其中，Vbus通过GPIO3_PC6控制

```
/* Disable Type-C0 USB 3.0 PHY */
&tcphy0 {
        status = "disabled";
};

/* 配置Vbus regulator属性 */
```

```
vcc_otg_vbus: otg-vbus-regulator {
                compatible = "regulator-fixed";
                gpio = <&gpio3 RK_PC6 GPIO_ACTIVE_HIGH>;
                pinctrl-names = "default";
                pinctrl-0 = <&otg_vbus_drv>;
                regulator-name = "vcc_otg_vbus";
                regulator-min-microvolt = <5000000>;
                regulator-max-microvolt = <5000000>;
                enable-active-high;
        };

&pinctrl {
        ......
        usb {
                otg_vbus_drv: otg-vbus-drv {
                        rockchip,pins = <3 RK_PC6 RK_FUNC_GPIO &pcfg_pull_none>;
                };
        };
};

/* Enable Type-C0 USB 2.0 PHY */
&u2phy0 {
        status = "okay";

        u2phy0_otg: otg-port {
                vbus-supply = <&vcc_otg_vbus>; /* 配置Vbus regulator属性，见Note1 */
                status = "okay";
        };
        ......
};

/* Configurate and Enable Type-C0 USB 3.0 Controller */
&usbdrd3_0 {
        /* USB控制器不用配置extcon属性，通过内核节点来切换Peripheral mode和Host mode，见Note2 */
        status = "okay";
};

&usbdrd_dwc3_0 {
        dr_mode = "otg"; /* USB控制器的dr_mode配置为otg */
        maximum-speed = “high-speed”; /* maximum-speed 属性配置为high-speed */
        phys = <&u2phy0_otg>; /* phys 属性只引用USB2 PHY节点 */
        phy-names = "usb2-phy";
        status = "okay";
};
```

Note1

假如Vbus为常供电（也即系统开机后，Vbus一直为高），则不需要配置“vbus-supply”属性，但需要增加如下的DTS属性，否则，会出现USB ADB无法正常连接的情况。

```
&u2phy0_otg {
        rockchip,vbus-always-on;
};
```

Note2

内核中切换USB控制器工作在Peripheral mode或Host mode的接口：

旧的接口使用方法：

```
1.Force host mode
  echo host > sys/kernel/debug/usb@fe800000/rk_usb_force_mode
2.Force peripheral mode
  echo peripheral > sys/kernel/debug/usb@fe800000/rk_usb_force_mode
```

新的接口使用方法：

```
1.Force host mode
  echo host > sys/kernel/debug/usb0/dwc3_mode
2.Force peripheral mode
  echo peripheral > sys/kernel/debug/usb0/dwc3_mode
```

5.2 Type-C to Type-A USB 2.0 Host only mode DTS

该方案的特点是，只支持Host 功能，Vbus为常供电，进系统后不需要Device功能，但可以支持固件烧录。Type-C的三路供电（见图2-2）可以关闭。

以Type-C0 USB配置为Type-C to Type-A USB 2.0 Host mode为例，其中，Vbus为常供电，不需要软件控制

```
/* Disable Type-C0 USB 3.0 PHY */
&tcphy0 {
        status = "disabled";
};

/* Enable Type-C0 USB 2.0 PHY */
&u2phy0 {
        status = "okay";

        u2phy0_otg: otg-port {
                /* 不需要配置vbus regulator属性 */
                status = "okay";
        };
        ......
};

/* Configurate and Enable Type-C0 USB 3.0 Controller */
&usbdrd3_0 {
        /* USB控制器不用配置extcon属性，通过内核节点来切换Peripheral mode和Host mode */
        status = "okay";
};

&usbdrd_dwc3_0 {
        dr_mode = "host"; /* USB控制器的dr_mode配置为host only mode */
        maximum-speed = “high-speed”; /* maximum-speed 属性配置为high-speed */
        phys = <&u2phy0_otg>; /* phys 属性只引用USB2 PHY节点 */
        phy-names = "usb2-phy";
        status = "okay";
};
```

5.3 Type-C to Type-A USB 2.0 OTG mode DTS and Support DP 4 Lane

该方案的特点是，支持USB 2.0 OTG功能，同时USB 3.0的Tx和Rx配置给DP使用，以支持DP 4 lanes的功能。Type-C的三路供电（见图2-2）需要正常开启。Rockchip SDK Kernel默认没有支持该方案，如果要支持该方案，需要正确配置DTS，同时，还要增加新的驱动`drivers/extcon/extcon-pd-virtual.c`，该驱动的作用是替代fusb302驱动，发通知给Type-C PHY驱动和DP驱动，以配置DP 4 lanes。如果有该功能需求，请提交Issue到Rockchip的Redmine平台，或者发邮件给本文档的作者wulf@rock-chips.com

DTS配置参考如下：

```
/* 配置VPD驱动，用于发送通知给Type-C PHY驱动和DP驱动，以配置DP 4 lanes */
vpd0:virtual-pd0{
        compatible = "linux,extcon-pd-virtual";
        pinctrl-names = "default";
        pinctrl-0 = <&vpd0_int>;
        dp-det-gpios = <&gpio4 25 GPIO_ACTIVE_LOW>;
        hdmi-5v-gpios = <&gpio4 29 GPIO_ACTIVE_LOW>;

        /* 0: positive, 1: negative*/
        vpd,init-flip = <0>;
        /* 0: u2, 1: u3*/
        vpd,init-ss = <0>;
        /* 0: dfp, 1: ufp, 2: dp 3: dp/ufp */
        vpd,init-mode = <2>;
        status = "okay";
};
```

```

&pinctrl {
        vpd {
                vpd0_int: vpd0-int {
                        rockchip,pins =
                                <4 25 RK_FUNC_GPIO &pcfg_pull_up>;
                };
        };
};

/* 配置DP */
&cdn_dp {
        status = "okay";
        extcon = <&vpd0>;
        dp_vop_sel = <1>;
};

/* 配置Type-C0 PHY */
&tcphy0 {
        extcon = <&vpd0>;
        status = "okay";
};

/* 配置USB2 PHY */
&u2phy0 {
        status = "okay";
        /* 这里不需要配置extcon属性 */

        u2phy0_otg: otg-port {
                status = "okay";
        };

        u2phy0_host: host-port {
                phy-supply = <&vcc5v0_host>;
                status = "okay";
        };
};

/* 配置USB控制器 */
&usbdrd3_0 {
        /* extcon不是必要的。如果用micro OTG接口，需要配置。如果用Type-A接口，不用配置，见Note1 */
        extcon = <&u2phy0>;
        status = "okay";
};

&usbdrd_dwc3_0 {
        dr_mode = "otg"; /* 配置为otg mode，支持peripheral和host切换 */
        maximum-speed = "high-speed"; /* 配置最高支持high-speed */
        phys = <&u2phy0_otg>, <&tcphy0_usb3>; /* 必须要配置U2和U3 PHY */
        phy-names = "usb2-phy", "usb3-phy";
        status = "okay";
};
```

Note1

如果用Type-A接口，系统启动后，需要应用层通过内核提供的OTG mode切换节点，配置USB控制器工作Peripheral mode或者Host mode。配置方法参考5.1 Type-C to Type-A USB 2.0 OTG mode DTS

6 USB 2.0 Host DTS

RK3399 支持两个USB2.0 Host接口，对应的USB控制器为EHCI&OHCI，相比Type-C接口的多种硬件设计方案，USB2.0 Host的接口一般只有一种设计方案，即Type-A USB2.0 Host接口，对应的DTS配置，包括控制器DTS配置和PHY DTS配置。

实际方案中，用户一般不用重新配置Host Controller DTS，只需要根据实际硬件电路的USB VBUS设计，修改Host PHY DTS的 “phy-supply ” 属性。

6.1 USB 2.0 Host Controller DTS

以RK3399 Sapphire Excavator Board USB2.0 Host 控制器 DTS配置为例:

`arch/arm64/boot/dts/rockchip/rk3399.dtsi`

```
usb_host0_ehci: usb@fe380000 {
                compatible = "generic-ehci";
                reg = <0x0 0xfe380000 0x0 0x20000>;
                interrupts = <GIC_SPI 26 IRQ_TYPE_LEVEL_HIGH 0>;
                clocks = <&cru HCLK_HOST0>, <&cru HCLK_HOST0_ARB>,
                         <&cru SCLK_USBPHY0_480M_SRC>;
                clock-names = "hclk_host0", "hclk_host0_arb", "usbphy0_480m";
                phys = <&u2phy0_host>;
                phy-names = "usb";
                power-domains = <&power RK3399_PD_PERIHP>;
                status = "disabled";
        };

usb_host0_ohci: usb@fe3a0000 {
                compatible = "generic-ohci";
                reg = <0x0 0xfe3a0000 0x0 0x20000>;
                interrupts = <GIC_SPI 28 IRQ_TYPE_LEVEL_HIGH 0>;
                clocks = <&cru HCLK_HOST0>, <&cru HCLK_HOST0_ARB>,
                         <&cru SCLK_USBPHY0_480M_SRC>;
                clock-names = "hclk_host0", "hclk_host0_arb", "usbphy0_480m";
                phys = <&u2phy0_host>;
                phy-names = "usb";
                power-domains = <&power RK3399_PD_PERIHP>;
                status = "disabled";
        };

usb_host1_ehci: usb@fe3c0000 {
                compatible = "generic-ehci";
                reg = <0x0 0xfe3c0000 0x0 0x20000>;
                interrupts = <GIC_SPI 30 IRQ_TYPE_LEVEL_HIGH 0>;
                clocks = <&cru HCLK_HOST1>, <&cru HCLK_HOST1_ARB>,
                         <&cru SCLK_USBPHY1_480M_SRC>;
                clock-names = "hclk_host1", "hclk_host1_arb", "usbphy1_480m";
                phys = <&u2phy1_host>;
                phy-names = "usb";
                power-domains = <&power RK3399_PD_PERIHP>;
                status = "disabled";
        };

usb_host1_ohci: usb@fe3e0000 {
                compatible = "generic-ohci";
                reg = <0x0 0xfe3e0000 0x0 0x20000>;
                interrupts = <GIC_SPI 32 IRQ_TYPE_LEVEL_HIGH 0>;
                clocks = <&cru HCLK_HOST1>, <&cru HCLK_HOST1_ARB>,
                         <&cru SCLK_USBPHY1_480M_SRC>;
                clock-names = "hclk_host1", "hclk_host1_arb", "usbphy1_480m";
                phys = <&u2phy1_host>;
                phy-names = "usb";
                power-domains = <&power RK3399_PD_PERIHP>;
                status = "disabled";
        };

```

`arch/arm64/boot/dts/rockchip/rk3399-sapphire.dtsi`

```
&usb_host0_ehci {
        status = "okay";
};

&usb_host0_ohci {
        status = "okay";
};

&usb_host1_ehci {
        status = "okay";
};

&usb_host1_ohci {
        status = "okay";
};
```

6.2 USB 2.0 Host PHY DTS

以RK3399 Sapphire Excavator Board USB2.0 Host PHY DTS配置为例:

`arch/arm64/boot/dts/rockchip/rk3399.dtsi`

```
grf: syscon@ff770000 {
                compatible = "rockchip,rk3399-grf", "syscon", "simple-mfd";
                reg = <0x0 0xff770000 0x0 0x10000>;
                #address-cells = <1>;
                #size-cells = <1>;
                ......
                u2phy0: usb2-phy@e450 {
                        compatible = "rockchip,rk3399-usb2phy";
                        reg = <0xe450 0x10>;
                        clocks = <&cru SCLK_USB2PHY0_REF>;
                        clock-names = "phyclk";
                        #clock-cells = <0>;
                        clock-output-names = "clk_usbphy0_480m";
                        status = "disabled";
                        ......
                        u2phy0_host: host-port { /* 配置USB2.0 Host0 PHY节点 */
                                #phy-cells = <0>;
                                interrupts = <GIC_SPI 27 IRQ_TYPE_LEVEL_HIGH 0>;
                                interrupt-names = "linestate";
                                status = "disabled";
                        };
                };

                u2phy1: usb2-phy@e460 {
                        compatible = "rockchip,rk3399-usb2phy";
                        reg = <0xe460 0x10>;
                        clocks = <&cru SCLK_USB2PHY1_REF>;
                        clock-names = "phyclk";
                        #clock-cells = <0>;
                        clock-output-names = "clk_usbphy1_480m";
                        status = "disabled";
                        ......
                        u2phy1_host: host-port { /* 配置USB2.0 Host1 PHY节点 */
                                #phy-cells = <0>;
                                interrupts = <GIC_SPI 31 IRQ_TYPE_LEVEL_HIGH 0>;
                                interrupt-names = "linestate";
                                status = "disabled";
                        };
                };
```

`arch/arm64/boot/dts/rockchip/rk3399-sapphire.dtsi`

```
/* 配置USB2.0 Host Vbus regulator */
vcc5v0_host: vcc5v0-host-regulator {
                compatible = "regulator-fixed";
                enable-active-high;
                gpio = <&gpio4 25 GPIO_ACTIVE_HIGH>;
                pinctrl-names = "default";
                pinctrl-0 = <&host_vbus_drv>;
                regulator-name = "vcc5v0_host";
                regulator-always-on;
        };

&pinctrl {
        ......
        usb2 {
                host_vbus_drv: host-vbus-drv {
                        rockchip,pins =
                                <4 25 RK_FUNC_GPIO &pcfg_pull_none>;
                };
        };
        ......
};

&u2phy0 {
        status = "okay";
        ...
        u2phy0_host: host-port {
                /* 配置USB2.0 Host0 phy-supply属性，用于控制Vbus */
                phy-supply = <&vcc5v0_host>;
                status = "okay";
        };
};

&u2phy1 {
        status = "okay";
        ...
        u2phy1_host: host-port {
                /* 配置USB2.0 Host1 phy-supply属性，用于控制Vbus */
                phy-supply = <&vcc5v0_host>;
                status = "okay";
        };
};
```

7 USB 3.0 force to USB 2.0

该功能是指在USB 3.0 Tx/Rx连接的情况下 ，要强制让USB运行在USB 2.0的速率。这种应用场景，一般用于硬件设计问题导致USB 3.0工作异常或者某些特殊的场景需求，需要去掉USB 3.0功能，只要支持USB 2.0。 由于这不是常规功能，所以SDK驱动默认没有支持该功能。Rockchip以独立的补丁形式，发布给有这类需求的客户。如果有该功能需求，请提交Issue到Rockchip的Redmine平台，或者发邮件给本文档的作者wulf@rock-chips.com

8 关于USB VBUS供电的说明

RK3399 平台的USB Vbus供电硬件电路设计，主要有三种方案：

1. 使用GPIO控制电源稳压芯片输出Vbus 5V供电电压；

   方案1是Rockchip平台常用的方案，可以用于Type-C USB 接口，Type-A USB接口，Micro USB接口等，不同接口，对应的DTS配置方案不同，具体如下：

   （1）Type-C USB接口的Vbus GPIO配置，参考1.2.1 Type-C0 /1 USB 3.0 PHY DTS中“vbus-5v-gpios”属性的配置；

   （2）Type-A USB接口的Vbus GPIO配置，参考6.2 USB 2.0 Host PHY DTS

   （3）Micro USB的Vbus GPIO配置，参考3 Type-C to Micro USB 3.0 OTG Mode DTS

2. 使用PMIC（如RK817/RK818）输出Vbus 5V供电电压；

   （1）如果PMIC使用的是RK8xx（RK809除外），DTS不用配置Vbus的属性，如果配置，反而可能会导致Vbus供电异常。这种方案，驱动会通过发送 EXTCON_USB_VBUS_EN 的通知给PMIC驱动，以控制Vbus的供电。

（2）如果PMIC使用的是其他Vendor的芯片，请参考驱动 drivers/power/rk818_charger.c实现接收EXTCON_USB_VBUS_EN 通知的逻辑。

（3）如果PMIC使用的是RK809，由于该PMIC只有Vbus输出5V的供电功能，没有充电功能，所以不适用于使用发送 EXTCON_USB_VBUS_EN 的通知的方法。可参考：

`arch/arm64/boot/dts/rockchip/rk3326-evb-ai-va-v11.dts`

```
rk809: pmic@20 {
        regulators {
                vcc5v0_host: SWITCH_REG1 {
                        regulator-name = "vcc5v0_host";
                };
        }；
}；
&u2phy {
        status = "okay";
        u2phy_host: host-port {
                status = "okay";
        };
        u2phy_otg: otg-port {
                vbus-supply = <&vcc5v0_host>;
                status = "okay";
        };
};
```

3. 开机后，硬件直接输出Vbus 5V供电电压，不需要软件控制，一般用于USB Host接口；

9 参考文档

1. Documentation/devicetree/bindings/usb/generic.txt
2. Documentation/devicetree/bindings/usb/dwc3.txt
3. Documentation/devicetree/bindings/usb/rockchip,dwc3.txt
4. Documentation/devicetree/bindings/usb/usb-ehci.txt
5. Documentation/devicetree/bindings/usb/usb-ohci.txt
6. Documentation/devicetree/bindings/phy/phy-rockchip-typec.txt
7. Documentation/devicetree/bindings/phy/phy-rockchip-inno-usb2.txt