

RK3399 USB DTS配置说明

发布版本：1.0

作者邮箱：wulf@rock-chips.com

日期：2018.3.1

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 USB3.0(TypeC PHY is a combination of USB3.0 SuperSpeed PHY and DisplayPort Transmit PHY)，两个USB2.0 Host。其中，两个Type-C USB3.0控制器都可以支持OTG（USB Peripheral和USB Host），并且向下兼容USB2.0/1.1/1.0。Type-C USB3.0可以根据实际的应用需求，将物理接口设计为Type-A USB3.0 Host，Micro USB3.0 OTG，Micro USB2.0 OTG等类型，内核USB驱动已经兼容这几种不同类型的USB接口，只需要修改DTS配置，就可以使能相应的USB接口。

DTS参考文档：

Documentation/devicetree/bindings/usb/usb-ehci.txt

Documentation/devicetree/bindings/usb/usb-ohci.txt

Documentation/devicetree/bindings/usb/generic.txt

Documentation/devicetree/bindings/usb/dwc3.txt

Documentation/devicetree/bindings/usb/rockchip,dwc3.txt

Documentation/devicetree/bindings/phy/phy-rockchip-typec.txt

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

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
2018.3.1	V1.0	吴良峰	

- 1 Type-C USB DTS配置 (default)
 - 1.1 Type-C0 /C1 USB 控制器DTS配置
 - 1.2 Type-C0 /C1 USB PHY D TS配置
 - 1.2.1 Type-C0 /C1 USB3.0 PHY D TS配置
 - 1.2.2 Type-C0 /C1 USB2.0 PHY D TS配置
 - 1.3 Type-C1 USB OTG DTS配置
- 2 Type-A USB3.0 Host D TS配置
- 3 Micro USB3.0 OTG DTS配置
- 4 Micro USB2.0 OTG DTS配置
- 5 USB2.0 Host D TS配置
 - 5.1 USB2.0 Host 控制器 DTS配置
 - 5.2 USB2.0 Host PHY D TS配置

1 Type-C USB DTS配置 (default)

Type-C 的接口类型如下图1-1所示。

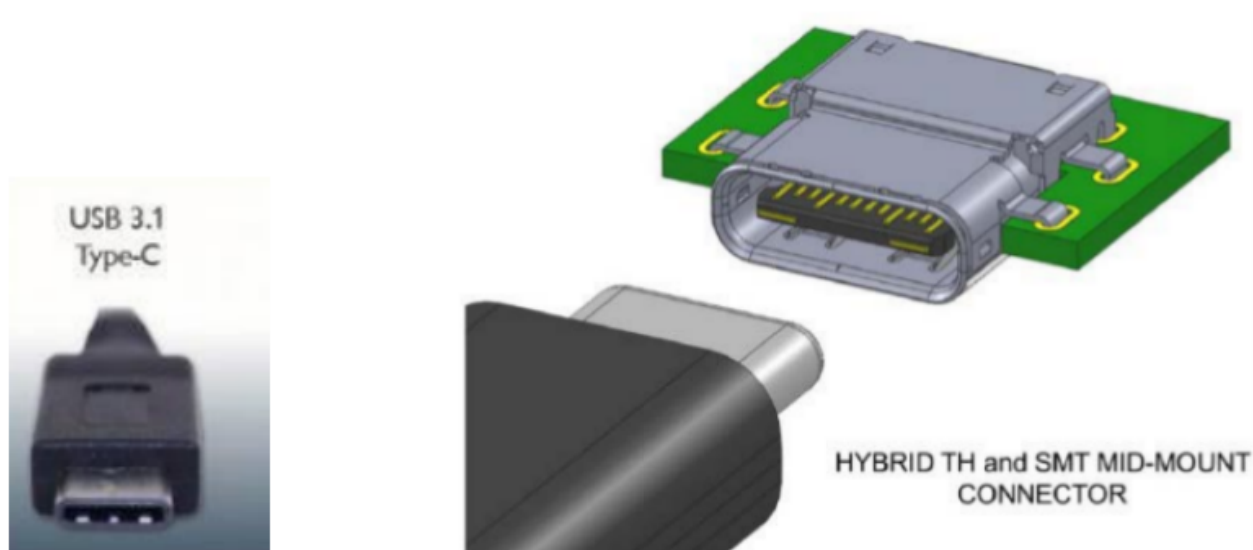


图1-1 Type-C 接口类型示意图

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

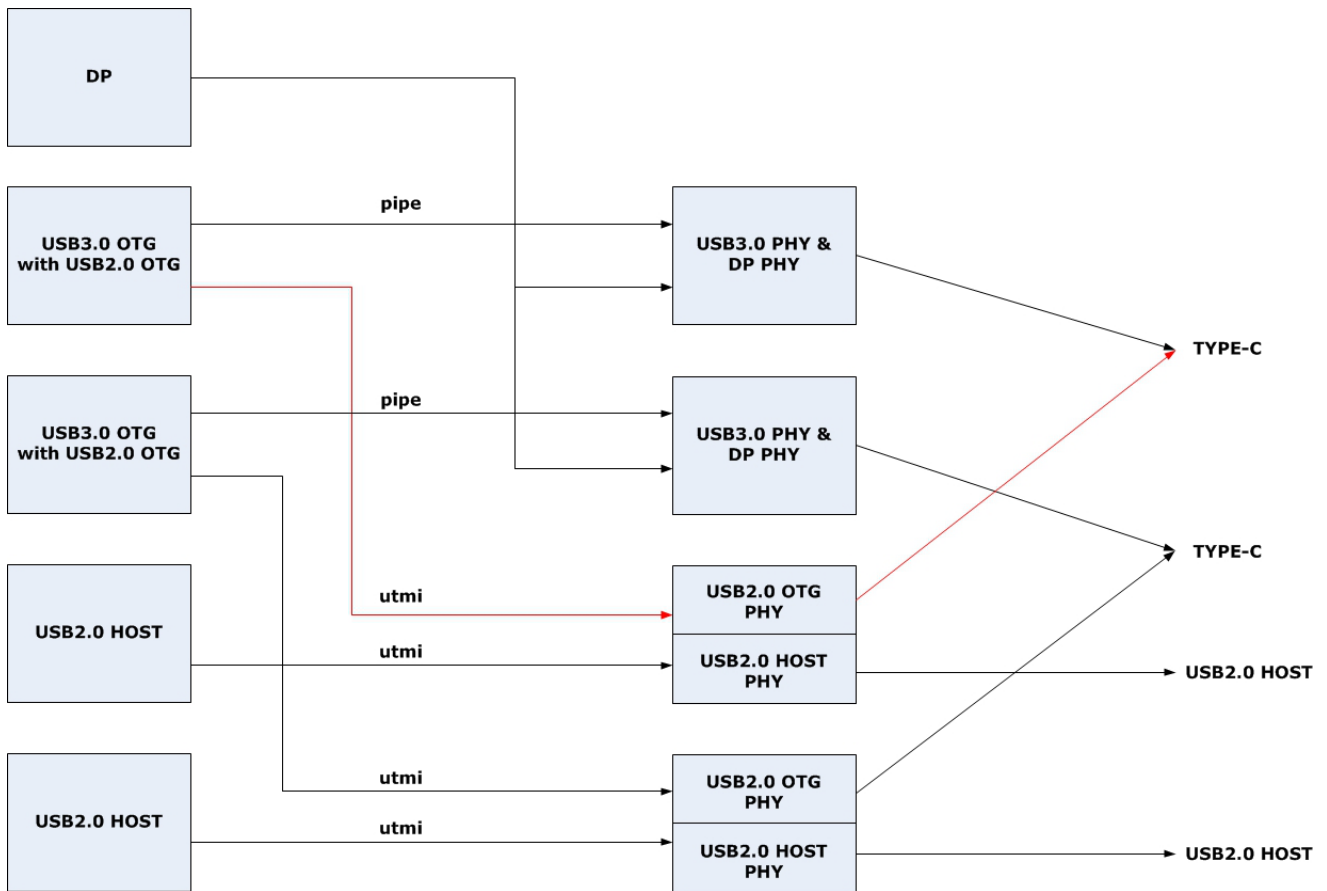


图1-2 RK3399 USB控制器&PHY连接示意图

RK3399 SDK DTS的默认配置，支持Type-C0 USB3.0 OTG功能，Type-C1 USB3.0 HOST功能。DTS的配置主要包括DWC3控制器、Type-C USB3.0 PHY、USB2.0 PHY。

1.1 Type-C0 /C1 USB 控制器DTS配置

Type-C0/C1 USB控制器支持USB3.0 OTG（USB Peripheral和USB Host）功能，并且向下兼容USB2.0/1.1/1.0。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，以RK3399 EVB Type-C0 /C1 USB3.0 控制器DTS配置为例：

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

```
usbdrd3_0: usb@fe800000 { /* Type-C0 USB3.0 控制器DTS配置*/
    compatible = "rockchip,rk3399-dwc3";
    clocks = <&cru SCLK_USB30TG0_REF>, <&cru SCLK_USB30TG0_SUSPEND>,
            <&cru ACLK_USB30TG0>, <&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";
    #address-cells = <2>;
    #size-cells = <2>;
    ranges;
    status = "disabled";
    usbdrd_dwc3_0: dwc3@fe800000 {
        compatible = "snps,dwc3";
```

```

        reg = <0x0 0xfe800000 0x0 0x1000000>;
        interrupts = <GIC_SPI 105 IRQ_TYPE_LEVEL_HIGH 0>;
        dr_mode = "otg"; /* 支持OTG mode */
        phys = <&u2phy0_otg>, <&tcphy0_usb3>; /* usb2 phy 和usb3 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,usb3-warm-reset-on-resume-quirk;
        status = "disabled";
    };

};

usbdrd3_1: usb@fe900000 { /* Type-C1 USB3.0 控制器DTS配置*/
    compatible = "rockchip,rk3399-dwc3";
    clocks = <&cru SCLK_USB30TG1_REF>, <&cru SCLK_USB30TG1_SUSPEND>,
        <&cru ACLK_USB30TG1>, <&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";
    #address-cells = <2>;
    #size-cells = <2>;
    ranges;
    status = "disabled";
    usbdrd_dwc3_1: dwc3@fe900000 {
        compatible = "snps,dwc3";
        reg = <0x0 0xfe900000 0x0 0x1000000>;
        interrupts = <GIC_SPI 110 IRQ_TYPE_LEVEL_HIGH 0>;
        dr_mode = "host"; /* 只支持Host mode */
        phys = <&u2phy1_otg>, <&tcphy1_usb3>; /* usb2 phy 和usb3 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,usb3-warm-reset-on-resume-quirk;
        status = "disabled";
    };

};

```

```

&usbdrd3_0 {
    extcon = <&fusb0>; /* extcon 属性 */
    status = "okay";
};

&usbdrd_dwc3_0 {
    status = "okay";
};

&usbdrd3_1 {
    extcon = <&fusb1>; /* extcon 属性 */
    status = "okay";
};

&usbdrd_dwc3_1 {
    status = "okay";
};

```

1.2 Type-C0 /C1 USB PHY DTS配置

Type-C0/C1 USB PHY的硬件由USB3.0 PHY（只支持Super-speed）和USB2.0 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USB3.0 PHY和USB2.0 PHY两部分。

1.2.1 Type-C0 /C1 USB3.0 PHY DTS配置

以RK3399 EVB3 Type-C0 /C1 USB3.0 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>;
    rockchip,grf = <&grf>;
    #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 = <500000000>;
    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,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>;
    rockchip,grf = <&grf>;
    #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 = <500000000>;
    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,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 -evb.dtsi
```

```

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

&tcphy1 {
    extcon = <&fusb1>;
    status = "okay";
};

```

```

};

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

        fusb1_int: fusb1-int { /* Type-C1 fusb302 中断 */
            rockchip,pins = <1 24 RK_FUNC_GPIO &pcfg_pull_up>;
        };
    };
};
};

```

arch/arm64/boot/dts/rockchip/rk3399-evb-rev3.dtsi

```

&i2c0 {
    fusb1: fusb30x@22 {
        compatible = "fairchild,fusb302";
        reg = <0x22>;
        pinctrl-names = "default";
        pinctrl-0 = <&fusb1_int>;
        vbus-5v-gpios = <&gpio1 4 GPIO_ACTIVE_LOW>;
        int-n-gpios = <&gpio1 24 GPIO_ACTIVE_HIGH>;
        status = "okay";
    };
    .....
};

&i2c6 {
    status = "okay";
    fusb0: fusb30x@22 {
        compatible = "fairchild,fusb302";
        reg = <0x22>;
        pinctrl-names = "default";
        pinctrl-0 = <&fusb0_int>;
        vbus-5v-gpios = <&gpio1 3 GPIO_ACTIVE_LOW>;
        int-n-gpios = <&gpio1 2 GPIO_ACTIVE_HIGH>;
        status = "okay";
    };
    .....
};

```

1.2.2 Type-C0 /C1 USB2.0 PHY DTS配置

以RK3399 EVB3 Type-C0 /C1 USB2.0 PHY DTS配置为例：

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

```

grf: syscon@fff770000 {
    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 { /* Type-C0 USB2.0 PHY port */
        #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 {
    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 PHY port*/
        #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 -evb.dtsi
```

```

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

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

```



```

};
.....
};

&u2phy1 {
    status = "okay";
    extcon = <&fusb1>; /* extcon 属性 */

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

```

1.3 Type-C1 USB OTG DTS配置

在[1.1 Type-C0 /C1 USB 控制器DTS配置](#)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，修改为Type-C1作为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 */
};

```

- 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-A USB3.0 Host DTS配置

Type-A USB3.0的接口类型如下图2-1所示。

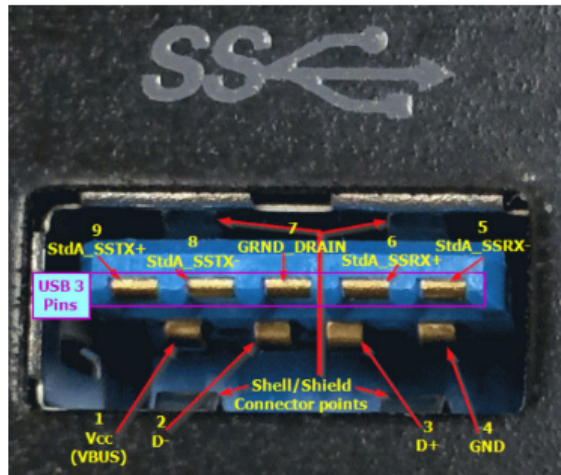
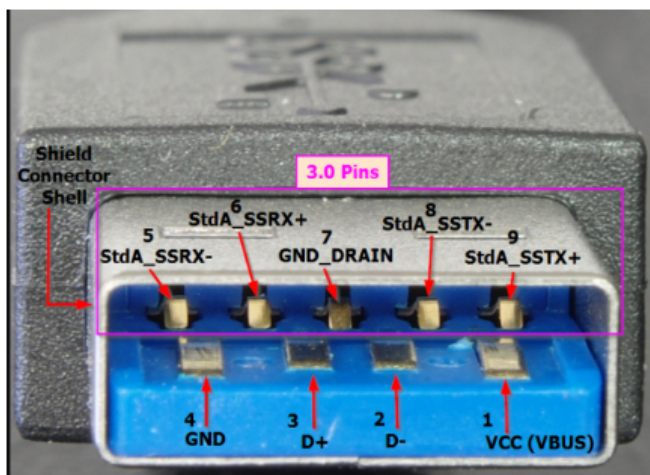


图2-1 Type-A USB3.0接口类型示意图

Type-C USB可以配置为Type-A USB使用。如RK3399 BOX SDK平台的Type-C1 USB默认设计为Type-A USB 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USB3.0 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 BOX平台为例（Type-C0 配置为Type-C接口，Type-C1配置为Type-A USB3 接口），介绍Type-A USB3.0 Host DTS配置的方法：

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

```
&tcphy0 {
    extcon = <&fusb0>; /* Type-C0 USB3 PHY extcon 属性 */
    status = "okay";
};

&tcphy1 { /* Type-A USB3 PHY 删除了extcon属性 */
    status = "okay";
};

&u2phy0 {
    status = "okay";
    extcon = <&fusb0>; /* Type-C0 USB2 PHY extcon 属性 */

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

```

        ....
};

&u2phy1 {
    status = "okay"; /*Type-A USB2 PHY 删除了extcon属性*/

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

&usbdrd3_0 {
    extcon = <&fusb0>;
    status = "okay";
};

&usbdrd_dwc3_0 {
    dr_mode = "otg";
    status = "okay";
};

&usbdrd3_1 {
    status = "okay";
};

&usbdrd_dwc3_1 { /* Type-C1 USB 控制器删除extcon属性,同时配置dr_mode为host */
    dr_mode = "host";
    status = "okay";
};

```

arch/arm64/boot/dts/rockchip/rk3399 -box-rev2.dts

```

&pinctrl {
    .....
    fusb30x {
        fusb0_int: fusb0-int {
            rockchip,pins =
                <1 2 RK_FUNC_GPIO &pcfg_pull_up>;
        };
    };
    .....
};

&i2c4 {
    status = "okay";
    fusb0: fusb30x@22 { /* Type-C0 对应的fusb302芯片的节点, Type-C1不需要fusb302 */
        compatible = "fairchild,fusb302";
        reg = <0x22>;
        pinctrl-names = "default";
        pinctrl-0 = <&fusb0_int>;
        vbus-5v-gpios = <&gpio1 3 GPIO_ACTIVE_LOW>;
    };
};

```

```

        int-n-gpios = <&gpio1 2 GPIO_ACTIVE_HIGH>;
        status = "okay";

    };

};

```

3 Micro USB3.0 OTG DTS配置

Micro USB3.0 OTG的接口类型如下图3-1所示。

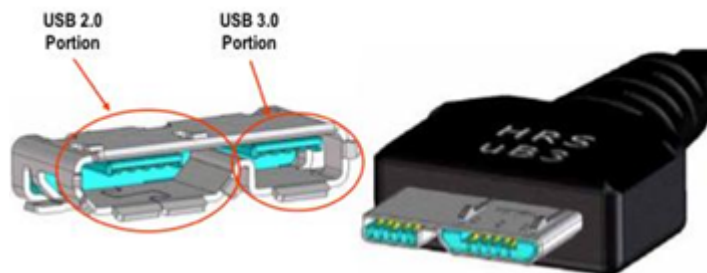


图3-1 Micro USB3.0 OTG接口类型示意图

Type-C USB可以配置为Micro USB3.0 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-A USB3.0 Host DTS配置](#)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
- 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

以Type-C0 USB配置为Micro USB3.0 OTG为例：

```

&tcphy0 { /* Micro USB3 PHY 删除了extcon属性 */
    status = "okay";
};

&u2phy0 {
    status = "okay"; /*Micro USB2 PHY 删除了extcon属性*/
    otg-vbus-gpios = <&gpio3 RK_PC6 GPIO_ACTIVE_HIGH>; /* Vbus GPIO 配置，见Note1 */

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

&usbdrd3_0 {
    extcon = <&u2phy0>; /* Micro USB3 控制器的extcon属性引用u2phy0 */
    status = "okay";
};

&usbdrd_dwc3_0 {

```

```

dr_mode = "otg"; /* Micro USB3 控制器的dr_mode配置为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（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，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 Micro USB2.0 OTG DTS配置

Micro USB2.0 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

Type-C USB可以配置为Micro USB2.0 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：

```
&tcphy0 {
    status = "disabled";
};

&u2phy0 {
    status = "okay"; /*Micro USB2 PHY 删除了extcon属性*/
    otg-vbus-gpios = <&gpio3 RK_PC6 GPIO_ACTIVE_HIGH>; /* Vbus GPIO 配置，见Note1 */

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

&usbdrd3_0 {
    extcon = <&u2phy0>; /* Micro USB3 控制器的extcon属性引用u2phy0 */
    status = "okay";
};

&usbdrd_dwc3_0 {
    dr_mode = "otg"; /* Micro USB3 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，应该参考[3 Micro USB3.0 OTG DTS配置](#)中Vbus regulator的配置方法。

5 USB2.0 Host DTS配置

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。

5.1 USB2.0 Host 控制器 DTS配置

以RK3399 EVB 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 -evb.dtsi
```

```

&usb_host0_ehci {
    status = "okay";
};

&usb_host0_ohci {
    status = "okay";
};

&usb_host1_ehci {
    status = "okay";
};

&usb_host1_ohci {
    status = "okay";
};

```

5.2 USB2.0 Host PHY DTS配置

```
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 USB2 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 USB2 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 -evb.dtsi

```

vcc5v0_host: vcc5v0-host-regulator {
    compatible = "regulator-fixed";
    enable-active-high;
    gpio = <&gpio4 25 GPIO_ACTIVE_HIGH>; /* 配置USB2.0 Host Vbus GPIO */
    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 {
        phy-supply = <&vcc5v0_host>; /* 配置USB2.0 Host0 Vbus regulator 属性 */
        status = "okay";
    };
};

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