

# RK3399 USB DTS配置说明

发布版本：1.0

作者邮箱：[wulf@rock-chips.com](mailto: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(Type-C 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。

## 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	

## RK3399 USB DTS配置说明

- 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 DTS配置
    - 1.2.1 Type-C0 /C1 USB3.0 PHY DTS配置
    - 1.2.2 Type-C0 /C1 USB2.0 PHY DTS配置
  - 1.3 Type-C1 USB OTG DTS配置
- 2 Type-A USB3.0 Host DTS配置
- 3 Micro USB3.0 OTG DTS配置
- 4 Micro USB2.0 OTG DTS配置
- 5 USB2.0 Host DTS配置
  - 5.1 USB2.0 Host 控制器 DTS配置
  - 5.2 USB2.0 Host PHY DTS配置
- 6 参考文档

## 1 Type-C USB DTS配置 (default)

Type-C 的接口类型如下图1-1所示。

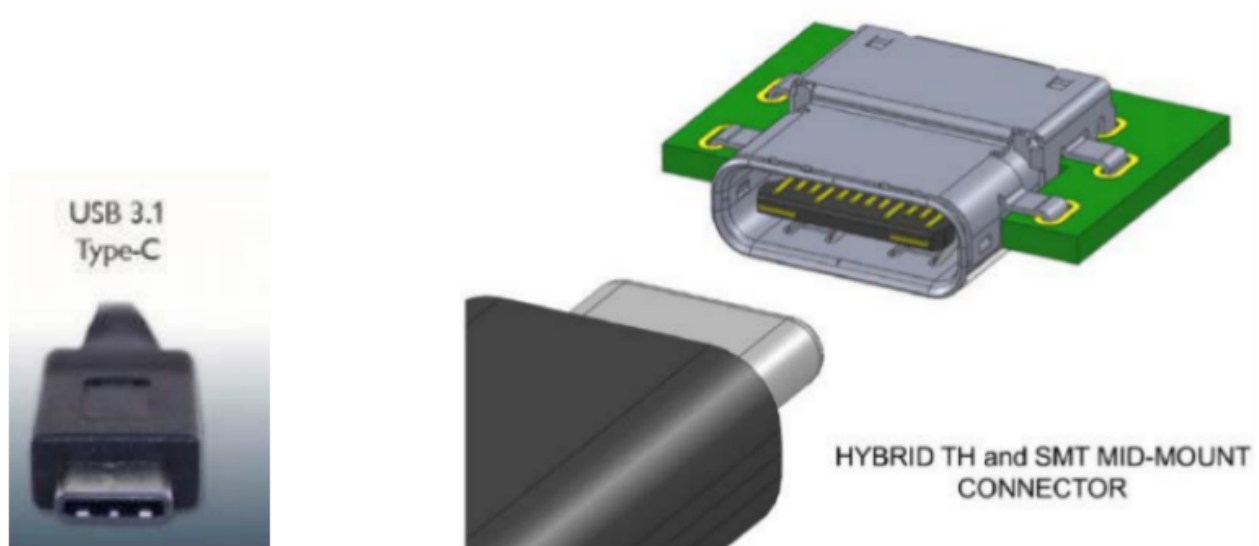


图1-1 Type-C 接口类型示意图

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

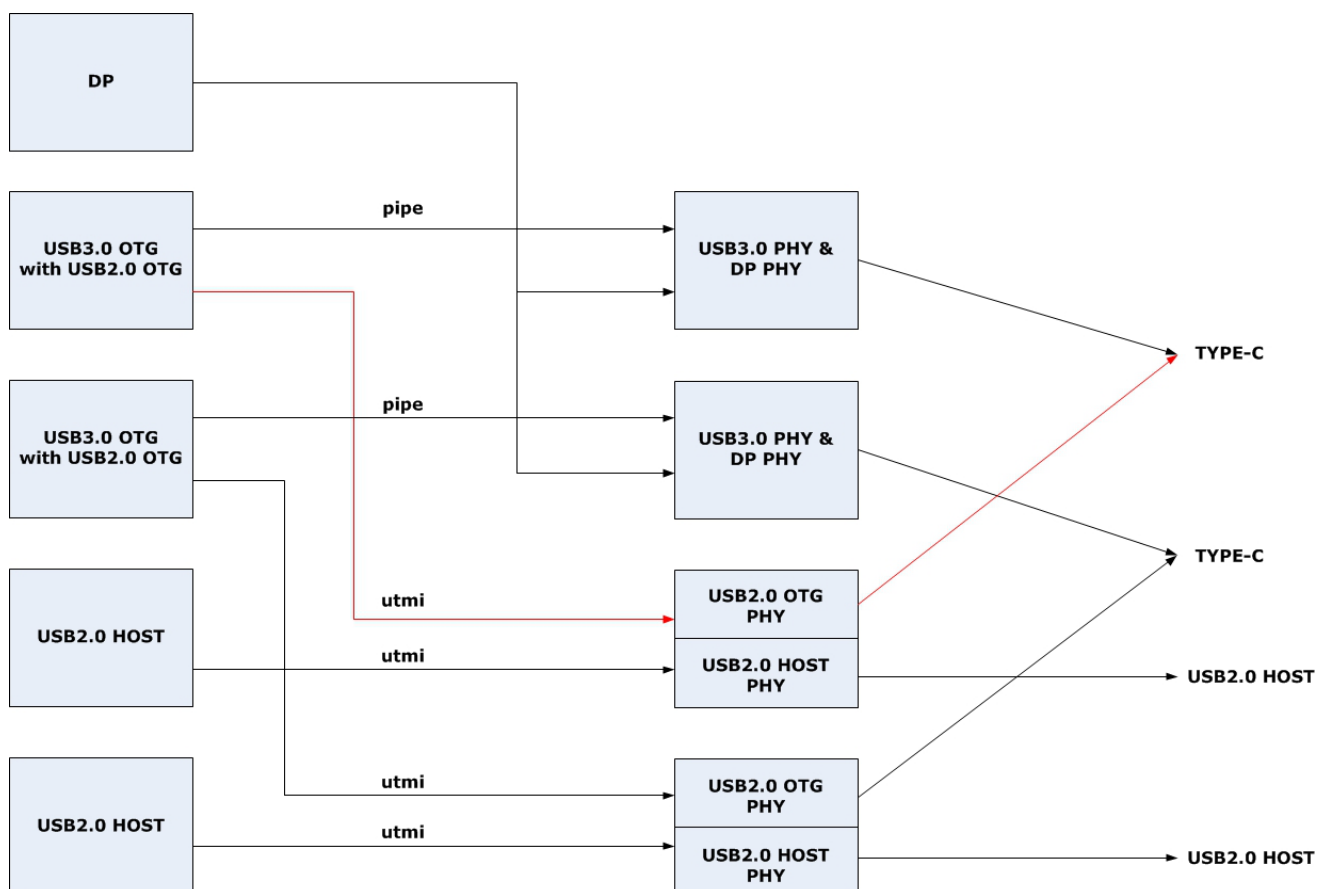


图1-2 RK3399 USB控制器&PHY连接示意图

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
```

```
1  usbd3_0: usb@fe800000 { /* Type-C0 USB3.0 控制器DTS配置*/
2      compatible = "rockchip,rk3399-dwc3";
3      clocks = <&cru SCLK_USB30TG0_REF>, <&cru SCLK_USB30TG0_SUSPEND>,
4              <&cru ACLK_USB30TG0>, <&cru ACLK_USB3_GRF>;
5      clock-names = "ref_clk", "suspend_clk",
6                  "bus_clk", "grf_clk";
7      power-domains = <&power RK3399_PD_USB3>;
8      resets = <&cru SRST_A_USB3_OTG0>;
9      reset-names = "usb3-otg";
10     #address-cells = <2>;
11     #size-cells = <2>;
12     ranges;
13     status = "disabled";
14     usbd3_dwc3_0: dwc3@fe800000 {
15         compatible = "snps,dwc3";
16         reg = <0x0 0xfe800000 0x0 0x100000>;
17         interrupts = <GIC_SPI 105 IRQ_TYPE_LEVEL_HIGH 0>;
18         dr_mode = "otg"; /* 支持OTG mode */
19         phys = <&u2phy0_otg>, <&tcphy0_usb3>; /* usb2 phy和usb3 phy属性 */
20         phy-names = "usb2-phy", "usb3-phy";
21         phy_type = "utmi_wide";
22         snps,dis_enblslpm_quirk;
23         snps,dis-u2-freeclk-exists-quirk;
24         snps,dis_u2_susphy_quirk;
25         snps,dis-del-phy-power-chg-quirk;
26         snps,tx-ipgap-linecheck-dis-quirk;
27         snps,xhci-slow-suspend-quirk;
28         snps,usb3-warm-reset-on-resume-quirk;
29         status = "disabled";
30     };
31 };
32
33 usbd3_1: usb@fe900000 { /* Type-C1 USB3.0 控制器DTS配置*/
34     compatible = "rockchip,rk3399-dwc3";
35     clocks = <&cru SCLK_USB30TG1_REF>, <&cru SCLK_USB30TG1_SUSPEND>,
36             <&cru ACLK_USB30TG1>, <&cru ACLK_USB3_GRF>;
37     clock-names = "ref_clk", "suspend_clk",
38                 "bus_clk", "grf_clk";
39     power-domains = <&power RK3399_PD_USB3>;
```

```

40         resets = <&cru SRST_A_USB3_OTG1>;
41         reset-names = "usb3-otg";
42         #address-cells = <2>;
43         #size-cells = <2>;
44         ranges;
45         status = "disabled";
46         usbdrd_dwc3_1: dwc3@fe900000 {
47             compatible = "snps,dwc3";
48             reg = <0x0 0xfe900000 0x0 0x100000>;
49             interrupts = <GIC_SPI 110 IRQ_TYPE_LEVEL_HIGH 0>;
50             dr_mode = "host"; /* 只支持Host mode */
51             phys = <&u2phy1_otg>, <&tcphy1_usb3>; /* usb2 phy和usb3 phy属性 */
52             phy-names = "usb2-phy", "usb3-phy";
53             phy_type = "utmi_wide";
54             snps,dis_enblslpm_quirk;
55             snps,dis-u2-freeclk-exists-quirk;
56             snps,dis_u2_susphy_quirk;
57             snps,dis-del-phy-power-chg-quirk;
58             snps,tx-ipgap-linecheck-dis-quirk;
59             snps,xhci-slow-suspend-quirk;
60             snps,usb3-warm-reset-on-resume-quirk;
61             status = "disabled";
62         };
63     };

```

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

```

1  &usbdrd3_0 {
2      extcon = <&fusb0>; /* extcon属性 */
3      status = "okay";
4  };
5
6  &usbdrd_dwc3_0 {
7      status = "okay";
8  };
9
10 &usbdrd3_1 {
11     extcon = <&fusb1>; /* extcon属性 */
12     status = "okay";
13 };
14
15 &usbdrd_dwc3_1 {
16     status = "okay";
17 };

```

## 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

```
1  tcphy0: phy@ff7c0000 {
2      compatible = "rockchip,rk3399-typec-phy";
3      reg = <0x0 0xff7c0000 0x0 0x40000>;
4      rockchip,grf = <&grf>;
5      #phy-cells = <1>;
6      clocks = <&cru SCLK_UPHY0_TCPDCORE>,
7              <&cru SCLK_UPHY0_TCPDPHY_REF>;
8      clock-names = "tcpdcore", "tcpdphy-ref";
9      assigned-clocks = <&cru SCLK_UPHY0_TCPDCORE>;
10     assigned-clock-rates = <50000000>;
11     power-domains = <&power RK3399_PD_TCPD0>;
12     resets = <&cru SRST_UPHY0>,
13             <&cru SRST_UPHY0_PIPE_L00>,
14             <&cru SRST_P_UPHY0_TCPHY>;
15     reset-names = "uphy", "uphy-pipe", "uphy-tcphy";
16     rockchip,typec-conn-dir = <0xe580 0 16>;
17     rockchip,usb3tousb2-en = <0xe580 3 19>;
18     rockchip,usb3-host-disable = <0x2434 0 16>;
19     rockchip,usb3-host-port = <0x2434 12 28>;
20     rockchip,external-psm = <0xe588 14 30>;
21     rockchip,pipe-status = <0xe5c0 0 0>;
22     rockchip,uphy-dp-sel = <0x6268 19 19>;
23     status = "disabled";
24
25     tcphy0_dp: dp-port {
26         #phy-cells = <0>;
27     };
28
29     tcphy0_usb3: usb3-port { /* Type-C0 USB3.0 port */
30         #phy-cells = <0>;
31     };
32 };
33
34 tcphy1: phy@ff800000 {
35     compatible = "rockchip,rk3399-typec-phy";
36     reg = <0x0 0xff800000 0x0 0x40000>;
37     rockchip,grf = <&grf>;
38     #phy-cells = <1>;
39     clocks = <&cru SCLK_UPHY1_TCPDCORE>,
40             <&cru SCLK_UPHY1_TCPDPHY_REF>;
41     clock-names = "tcpdcore", "tcpdphy-ref";
42     assigned-clocks = <&cru SCLK_UPHY1_TCPDCORE>;
43     assigned-clock-rates = <50000000>;
44     power-domains = <&power RK3399_PD_TCPD1>;
45     resets = <&cru SRST_UPHY1>,
46             <&cru SRST_UPHY1_PIPE_L00>,
47             <&cru SRST_P_UPHY1_TCPHY>;
48     reset-names = "uphy", "uphy-pipe", "uphy-tcphy";
49
50     rockchip,typec-conn-dir = <0xe58c 0 16>;
```

```

50         rockchip,usb3tousb2-en = <0xe58c 3 19>;
51         rockchip,usb3-host-disable = <0x2444 0 16>;
52         rockchip,usb3-host-port = <0x2444 12 28>;
53         rockchip,external-psm = <0xe594 14 30>;
54         rockchip,pipe-status = <0xe5c0 16 16>;
55         rockchip,uphy-dp-sel = <0x6268 3 19>;
56         status = "disabled";
57
58         tcphy1_dp: dp-port {
59             #phy-cells = <0>;
60         };
61
62         tcphy1_usb3: usb3-port { /* Type-C1 USB3.0 port */
63             #phy-cells = <0>;
64         };
65     };

```

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

```

1  &tcphy0 {
2      extcon = <&fusb0>;
3      status = "okay";
4  };
5
6  &tcphy1 {
7      extcon = <&fusb1>;
8      status = "okay";
9  };
10
11 &pinctrl {
12     .....
13     fusb30x {
14         fusb0_int: fusb0-int { /* TypeC0 fusb302 中斷 */
15             rockchip,pins = <1 2 RK_FUNC_GPIO &pcfg_pull_up>;
16         };
17
18         fusb1_int: fusb1-int { /* Type-C1 fusb302 中斷 */
19             rockchip,pins = <1 24 RK_FUNC_GPIO &pcfg_pull_up>;
20         };
21     };
22 };

```

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

```

1  &i2c0 {
2      fusb1: fusb30x@22 {
3          compatible = "fairchild,fusb302";
4          reg = <0x22>;
5          pinctrl-names = "default";
6          pinctrl-0 = <&fusb1_int>;
7          vbus-5v-gpios = <&gpio1 4 GPIO_ACTIVE_LOW>;
8          int-n-gpios = <&gpio1 24 GPIO_ACTIVE_HIGH>;

```

```

9         status = "okay";
10     };
11     .....
12 };
13
14 &i2c6 {
15     status = "okay";
16     fusb0: fusb30x@22 {
17         compatible = "fairchild,fusb302";
18         reg = <0x22>;
19         pinctrl-names = "default";
20         pinctrl-0 = <&fusb0_int>;
21         vbus-5v-gpios = <&gpio1 3 GPIO_ACTIVE_LOW>;
22         int-n-gpios = <&gpio1 2 GPIO_ACTIVE_HIGH>;
23         status = "okay";
24     };
25     .....
26 };

```

## 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
```

```

1  grf: syscon@ff770000 {
2      compatible = "rockchip,rk3399-grf", "syscon", "simple-mfd";
3      .....
4      u2phy0: usb2-phy@e450 {
5          compatible = "rockchip,rk3399-usb2phy";
6          reg = <0xe450 0x10>;
7          clocks = <&cru SCLK_USB2PHY0_REF>;
8          clock-names = "phyclk";
9          #clock-cells = <0>;
10         clock-output-names = "clk_usbphy0_480m";
11         status = "disabled";
12
13         u2phy0_otg: otg-port { /* Type-C0 USB2.0 PHY port */
14             #phy-cells = <0>;
15             interrupts = <GIC_SPI 103 IRQ_TYPE_LEVEL_HIGH 0>,
16                         <GIC_SPI 104 IRQ_TYPE_LEVEL_HIGH 0>,
17                         <GIC_SPI 106 IRQ_TYPE_LEVEL_HIGH 0>;
18             interrupt-names = "otg-bvalid", "otg-id",
19                             "linestate";
20             status = "disabled";
21         };
22
23         .....
24     };
25
26     u2phy1: usb2-phy@e460 {
27         compatible = "rockchip,rk3399-usb2phy";
28         reg = <0xe460 0x10>;

```

```

29         clocks = <&cru SCLK_USB2PHY1_REF>;
30         clock-names = "phyclk";
31         #clock-cells = <0>;
32         clock-output-names = "clk_usbphy1_480m";
33         status = "disabled";
34
35         u2phy1_otg: otg-port { /* Type-C1 USB2.0 PHY port*/
36             #phy-cells = <0>;
37             interrupts = <GIC_SPI 108 IRQ_TYPE_LEVEL_HIGH 0>,
38                         <GIC_SPI 109 IRQ_TYPE_LEVEL_HIGH 0>,
39                         <GIC_SPI 111 IRQ_TYPE_LEVEL_HIGH 0>;
40             interrupt-names = "otg-bvalid", "otg-id",
41                             "linestate";
42             status = "disabled";
43         };
44
45         .....
46     };

```

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

```

1  &u2phy0 {
2      status = "okay";
3      extcon = <&fusb0>; /* extcon 属性*/
4
5      u2phy0_otg: otg-port {
6          status = "okay";
7      };
8      .....
9  };
10
11 &u2phy1 {
12     status = "okay";
13     extcon = <&fusb1>; /* extcon 属性 */
14
15     u2phy1_otg: otg-port {
16         status = "okay";
17     };
18     .....
19 };
20

```

## 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



```

1 &usbdrd_dwc3_1 {
2     status = "okay";
3     dr_mode = "otg"; /* 配置Type-C1 USB控制器为OTG mode */
4 };

```

- 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。

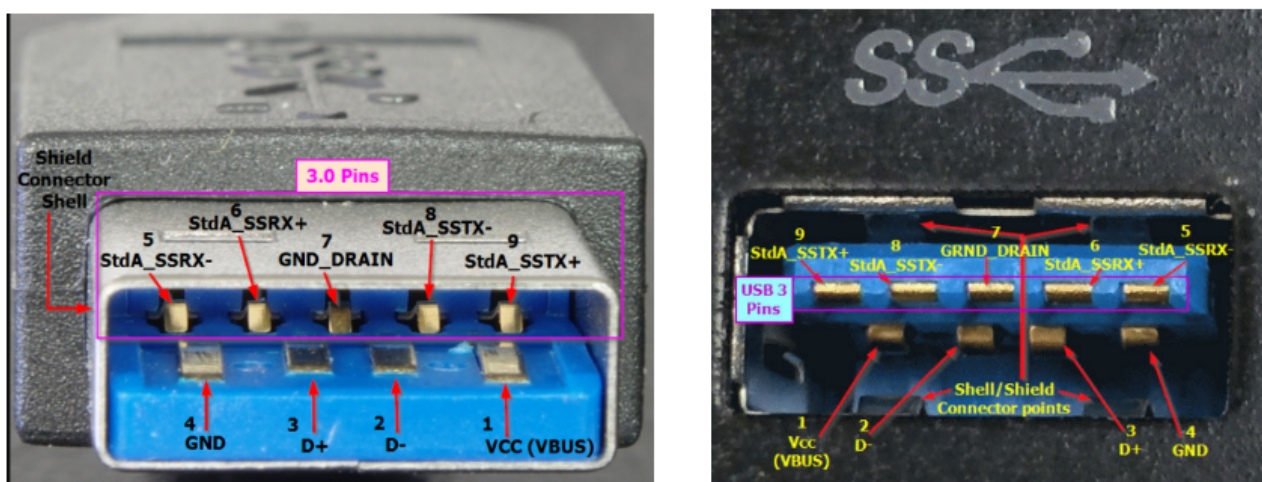


图2-1 Type-A USB3.0接口类型示意图

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
```

```

1  &tcphy0 {
2      extcon = <&fusb0>; /* Type-C0 USB3 PHY extcon属性 */
3      status = "okay";
4  };
5
6  &tcphy1 { /* Type-A USB3 PHY 删除了extcon属性 */
7      status = "okay";
8  };
9
10 &u2phy0 {
11     status = "okay";
12     extcon = <&fusb0>; /* Type-C0 USB2 PHY extcon属性 */
13
14     u2phy0_otg: otg-port {
15         status = "okay";
16     };
17     ....
18 };
19
20 &u2phy1 {
21     status = "okay"; /*Type-A USB2 PHY 删除了extcon属性*/
22
23     u2phy1_otg: otg-port {
24         status = "okay";
25     };
26     .....
27 };
28
29 &usbdrd3_0 {
30     extcon = <&fusb0>;
31     status = "okay";
32 };
33
34 &usbdrd_dwc3_0 {
35     dr_mode = "otg";
36     status = "okay";
37 };
38
39 &usbdrd3_1 {
40     status = "okay";
41 };
42
43 &usbdrd_dwc3_1 { /* Type-C1 USB控制器删除extcon属性，同时配置dr_mode为host */
44     dr_mode = "host";
45     status = "okay";
46 };
47

```

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

```

1  &pinctrl {
2      .....

```

```

3      fusb30x {
4          fusb0_int: fusb0-int {
5              rockchip,pins =
6                  <1 2 RK_FUNC_GPIO &pcfg_pull_up>;
7          };
8      };
9      .....
10 };
11
12 &i2c4 {
13     status = "okay";
14     fusb0: fusb30x@22 { /* Type-C0 对应的fusb302芯片的节点, Type-C1不需要fusb302 */
15         compatible = "fairchild,fusb302";
16         reg = <0x22>;
17         pinctrl-names = "default";
18         pinctrl-0 = <&fusb0_int>;
19         vbus-5v-gpios = <&gpio1 3 GPIO_ACTIVE_LOW>;
20         int-n-gpios = <&gpio1 2 GPIO_ACTIVE_HIGH>;
21         status = "okay";
22     };
23 };

```

### 3 Micro USB3.0 OTG DTS配置

Micro USB3.0 OTG的接口类型如下图3-1所示。

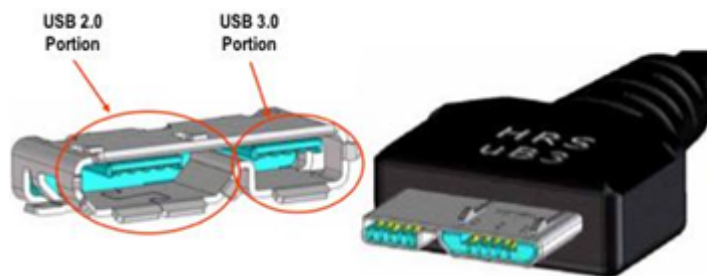


图3-1 Micro USB3.0 OTG接口类型示意图

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：

```

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

```

```

3 };
4
5 &u2phy0 {
6     status = "okay"; /*Micro USB2 PHY 删除了extcon属性*/
7     otg-vbus-gpios = <&gpio3 RK_PC6 GPIO_ACTIVE_HIGH>; /* Vbus GPIO配置, 见Note1 */
8
9     u2phy1_otg: otg-port {
10         status = "okay";
11     };
12     .....
13 };
14
15 &usbdrd3_0 {
16     extcon = <&u2phy0>; /* Micro USB3控制器的extcon属性引用u2phy0 */
17     status = "okay";
18 };
19
20 &usbdrd_dwc3_0 {
21     dr_mode = "otg"; /* Micro USB3控制器的dr_mode配置为otg */
22     status = "okay";
23 };

```

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:

```

1 vcc_otg_vbus: otg-vbus-regulator {
2     compatible = "regulator-fixed";
3     gpio = <&gpio3 RK_PC6 GPIO_ACTIVE_HIGH>;
4     pinctrl-names = "default";
5     pinctrl-0 = <&otg_vbus_drv>;
6     regulator-name = "vcc_otg_vbus";
7     regulator-min-microvolt = <5000000>;
8     regulator-max-microvolt = <5000000>;
9     enable-active-high;
10 };
11
12 &pinctrl {
13     .....
14     usb {
15         otg_vbus_drv: otg-vbus-drv {
16             rockchip,pins = <3 RK_PC6 RK_FUNC_GPIO &pcfg_pull_none>;
17         };
18     };
19 };
20
21 &u2phy0 {
22     status = "okay";
23 };

```

```

24     u2phy0_otg: otg-port {
25         vbus-supply = <vcc_otg_vbus>; /*配置Vbus regulator属性 */
26         status = "okay";
27     };
28     .....
29 };

```

## 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：

```

1  &tcphy0 {
2      status = "disabled";
3  };
4
5  &u2phy0 {
6      status = "okay"; /*Micro USB2 PHY 删除了extcon属性*/
7      otg-vbus-gpios = <&gpio3 RK_PC6 GPIO_ACTIVE_HIGH>; /* Vbus GPIO配置，见Note1 */
8
9      u2phy1_otg: otg-port {
10         status = "okay";
11     };
12     .....
13 };
14
15 &usbdrd3_0 {
16     extcon = <&u2phy0>; /* Micro USB3控制器的extcon属性引用u2phy0 */

```

```

17     status = "okay";
18 };
19
20 &usbdrd_dwc3_0 {
21     dr_mode = "otg"; /* Micro USB3控制器的dr_mode配置为otg */
22     maximum-speed = "high-speed"; /* maximum-speed 属性配置为high-speed */
23     phys = <&u2phy0_otg>; /* phys 属性只引用USB2 PHY节点 */
24     phy-names = "usb2-phy";
25     status = "okay";
26 };

```

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 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
```

```

1  usb_host0_ehci: usb@fe380000 {
2      compatible = "generic-ehci";
3      reg = <0x0 0xfe380000 0x0 0x20000>;
4      interrupts = <GIC_SPI 26 IRQ_TYPE_LEVEL_HIGH 0>;
5      clocks = <&cru HCLK_HOST0>, <&cru HCLK_HOST0_ARB>,
6              <&cru SCLK_USBPHY0_480M_SRC>;
7      clock-names = "hclk_host0", "hclk_host0_arb", "usbphy0_480m";
8      phys = <&u2phy0_host>;
9      phy-names = "usb";
10     power-domains = <&power RK3399_PD_PERIHP>;
11     status = "disabled";
12 };
13
14 usb_host0_ohci: usb@fe3a0000 {
15     compatible = "generic-ohci";
16     reg = <0x0 0xfe3a0000 0x0 0x20000>;
17     interrupts = <GIC_SPI 28 IRQ_TYPE_LEVEL_HIGH 0>;
18     clocks = <&cru HCLK_HOST0>, <&cru HCLK_HOST0_ARB>,
19             <&cru SCLK_USBPHY0_480M_SRC>;
20     clock-names = "hclk_host0", "hclk_host0_arb", "usbphy0_480m";
21     phys = <&u2phy0_host>;

```

```

22         phy-names = "usb";
23         power-domains = <&power RK3399_PD_PERIHP>;
24         status = "disabled";
25     };
26
27     usb_host1_ehci: usb@fe3c0000 {
28         compatible = "generic-ehci";
29         reg = <0x0 0xfe3c0000 0x0 0x20000>;
30         interrupts = <GIC_SPI 30 IRQ_TYPE_LEVEL_HIGH 0>;
31         clocks = <&cru HCLK_HOST1>, <&cru HCLK_HOST1_ARB>,
32             <&cru SCLK_USBPHY1_480M_SRC>;
33         clock-names = "hclk_host1", "hclk_host1_arb", "usbphy1_480m";
34         phys = <&u2phy1_host>;
35         phy-names = "usb";
36         power-domains = <&power RK3399_PD_PERIHP>;
37         status = "disabled";
38     };
39
40     usb_host1_ohci: usb@fe3e0000 {
41         compatible = "generic-ohci";
42         reg = <0x0 0xfe3e0000 0x0 0x20000>;
43         interrupts = <GIC_SPI 32 IRQ_TYPE_LEVEL_HIGH 0>;
44         clocks = <&cru HCLK_HOST1>, <&cru HCLK_HOST1_ARB>,
45             <&cru SCLK_USBPHY1_480M_SRC>;
46         clock-names = "hclk_host1", "hclk_host1_arb", "usbphy1_480m";
47         phys = <&u2phy1_host>;
48         phy-names = "usb";
49         power-domains = <&power RK3399_PD_PERIHP>;
50         status = "disabled";
51     };
52

```

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

```

1  &usb_host0_ehci {
2      status = "okay";
3  };
4
5  &usb_host0_ohci {
6      status = "okay";
7  };
8
9  &usb_host1_ehci {
10     status = "okay";
11 };
12
13 &usb_host1_ohci {
14     status = "okay";
15 };

```

## 5.2 USB2.0 Host PHY DTS配置

以RK3399 EVB USB2.0 Host PHY DTS配置为例:

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

```
1  grf: syscon@fff770000 {
2      compatible = "rockchip,rk3399-grf", "syscon", "simple-mfd";
3      reg = <0x0 0xff770000 0x0 0x10000>;
4      #address-cells = <1>;
5      #size-cells = <1>;
6      .....
7      u2phy0: usb2-phy@e450 {
8          compatible = "rockchip,rk3399-usb2phy";
9          reg = <0xe450 0x10>;
10         clocks = <&cru SCLK_USB2PHY0_REF>;
11         clock-names = "phyclk";
12         #clock-cells = <0>;
13         clock-output-names = "clk_usbphy0_480m";
14         status = "disabled";
15         .....
16         u2phy0_host: host-port { /* 配置USB2.0 Host0 USB2 PHY节点 */
17             #phy-cells = <0>;
18             interrupts = <GIC_SPI 27 IRQ_TYPE_LEVEL_HIGH 0>;
19             interrupt-names = "linestate";
20             status = "disabled";
21         };
22     };
23
24     u2phy1: usb2-phy@e460 {
25         compatible = "rockchip,rk3399-usb2phy";
26         reg = <0xe460 0x10>;
27         clocks = <&cru SCLK_USB2PHY1_REF>;
28         clock-names = "phyclk";
29         #clock-cells = <0>;
30         clock-output-names = "clk_usbphy1_480m";
31         status = "disabled";
32         .....
33         u2phy1_host: host-port { /* 配置USB2.0 Host1 USB2 PHY节点 */
34             #phy-cells = <0>;
35             interrupts = <GIC_SPI 31 IRQ_TYPE_LEVEL_HIGH 0>;
36             interrupt-names = "linestate";
37             status = "disabled";
38         };
39     };
```

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

```
1  vcc5v0_host: vcc5v0-host-regulator {
2      compatible = "regulator-fixed";
3      enable-active-high;
4      gpio = <&gpio4 25 GPIO_ACTIVE_HIGH>; /* 配置USB2.0 Host Vbus GPIO */
5      pinctrl-names = "default";
6      pinctrl-0 = <&host_vbus_drv>;
7      regulator-name = "vcc5v0_host";
```



```

8         regulator-always-on;
9     };
10
11 &pinctrl {
12     .....
13     usb2 {
14         host_vbus_drv: host-vbus-drv {
15             rockchip,pins =
16                 <4 25 RK_FUNC_GPIO &pcfg_pull_none>;
17         };
18     };
19     .....
20 };
21
22 &u2phy0 {
23     status = "okay";
24     ...
25     u2phy0_host: host-port {
26         phy-supply = <&vcc5v0_host>; /* 配置USB2.0 Host0 Vbus regulator属性 */
27         status = "okay";
28     };
29 };
30
31 &u2phy1 {
32     status = "okay";
33     ...
34     u2phy1_host: host-port {
35         phy-supply = <&vcc5v0_host>; /* 配置USB2.0 Host1 Vbus regulator属性 */
36         status = "okay";
37     };
38 };

```

## 6 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