



SIM8500系列 USB应用文档

智能模组

芯讯通无线科技(上海)有限公司

上海市长宁区临虹路289号3号楼芯讯通总部大楼

电话: 86-21-31575100

技术支持邮箱: support@simcom.com

官网: www.simcom.com

名称:	SIM8500系列 USB应用文档
版本:	1.00
日期:	2022.03.08
状态:	已发布

版权声明

本手册包含芯讯通无线科技（上海）有限公司（简称：芯讯通）的技术信息。除非经芯讯通书面许可，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并不得以任何形式传播，违反者将被追究法律责任。对技术信息涉及的专利、实用新型或者外观设计等知识产权，芯讯通保留一切权利。芯讯通有权在不通知的情况下随时更新本手册的具体内容。

本手册版权属于芯讯通，任何人未经我公司书面同意进行复制、引用或者修改本手册都将承担法律责任。

芯讯通无线科技(上海)有限公司

上海市长宁区临虹路289号3号楼芯讯通总部大楼

电话：86-21-31575100

邮箱：simcom@simcom.com

官网：www.simcom.com

了解更多资料，请点击以下链接：

<https://cn.simcom.com/>

技术支持，请点击以下链接：

<http://cn.simcom.com/ask/index-cn.html> 或发送邮件至 support@simcom.com

版权所有 © 芯讯通无线科技(上海)有限公司 2021，保留一切权利。

关于文档

版本历史

版本	日期	作者	备注
1.00	2022.03.08	葛正阳	第一版

适用范围

本文档适用于 SIMCom SIM8500E 系列，SIM8500CE 系列。

目录

关于文档.....	3
版本历史.....	3
适用范围.....	3
目录.....	4
1 介绍.....	5
1.1 本文目的.....	5
1.2 参考文档.....	5
1.3 术语和缩写.....	5
2 USB 2.0 功能描述.....	6
2.1 特性.....	6
2.2 USB 2.0 OTG 功能流程.....	7
3 Micro USB 的软硬件设计.....	8
4 TYPE-C 的软硬件设计.....	9
5 USB 转以太网的软硬件设计.....	10
6 USB PHY Turning 参数导入.....	11

1 介绍

1.1 本文目的

基于 SIM8500 平台，本文讲述了 Micro 端口和 TYPE-C 端口的 USB 2.0 参考设计同时引入了 USB 转以太网的参数设计。

参考此应用文档，开发者可以很快理解并快速开发相关业务。

1.2 参考文档

[1] [D1T1-2 - USB Type-C System Overview.pdf](#)

[2] [USB 2.0 Specification](#)

1.3 术语和缩写

表 1：术语和缩写

英文缩写	解释与描述
CC	Configuration Channel, TYPE-C 识别检测引脚
USB ID	USB 2.0 ID 引脚，拉低后主机进入 Host 模式
VBUS	USB 总线电源正极
VBAT	电池电源正极
GPIO	通用输入输出端口

2 USB 2.0 功能描述

USB 2.0 是一种主从协议，也是日常生活中最常见的高速总线协议。在一个 USB 网络中，只能有一个主机(Host) 与从机(Device) 进行通讯，一个主机最多可以与 127 个从机进行通讯。当一个主机与多个从机进行连接时，就需要通过 USB HUB 将端口进行拆分。在 USB 2.0 规范中，引入了 OTG (On-The-Go) 的概念，支持 OTG 的设备可以在 主机 / 从机 两种角色之间进行切换，从而更加方便地连接外设与其他主机设备。

2.1 特性

➤ Micro USB 接口；

✧ 高性价比

Micro 接口价格便宜，完全支持 USB 2.0 规范，比较适合用作内部调试接口。

✧ 支持 OTG

可以通过拉低 ID PIN 引脚使能模块的主机模式 (Host 模式)，用于连接外设。

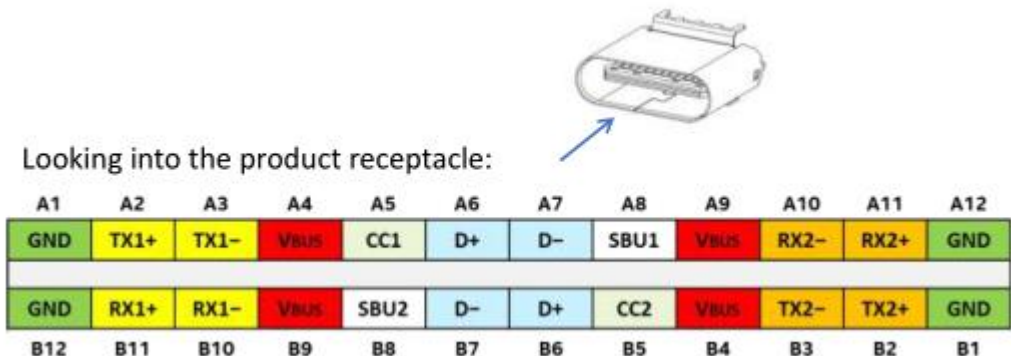
➤ USB TYPE-C 接口；

✧ 高可用性

USB TYPE-C 接口支持正反盲插，稳定性和使用寿命远高于 Micro USB，最大可支持 100W 电源传输，有着丰富拓展功能，同时兼容 USB 2.0，适合用作用户接口。

✧ 支持 DRD

TYPE-C 的规范更加复杂，命名也有些不同。支持 DRD (Dual-Role-Data) 的设备可以简单地理解为也支持 OTG 功能。与 Micro USB 不同，TYPE-C 使用 CC 引脚进行设备识别和通讯，比如识别设备的插入方向和使能模块进入主机模式 (Host 模式)。



2.2 USB 2.0 OTG 功能流程

注：左边是 SIMCOM EVB 开发所使用的 Micro USB 参考设计，右边是可选参考设计，虚线为组合选项，实线为固定选项。

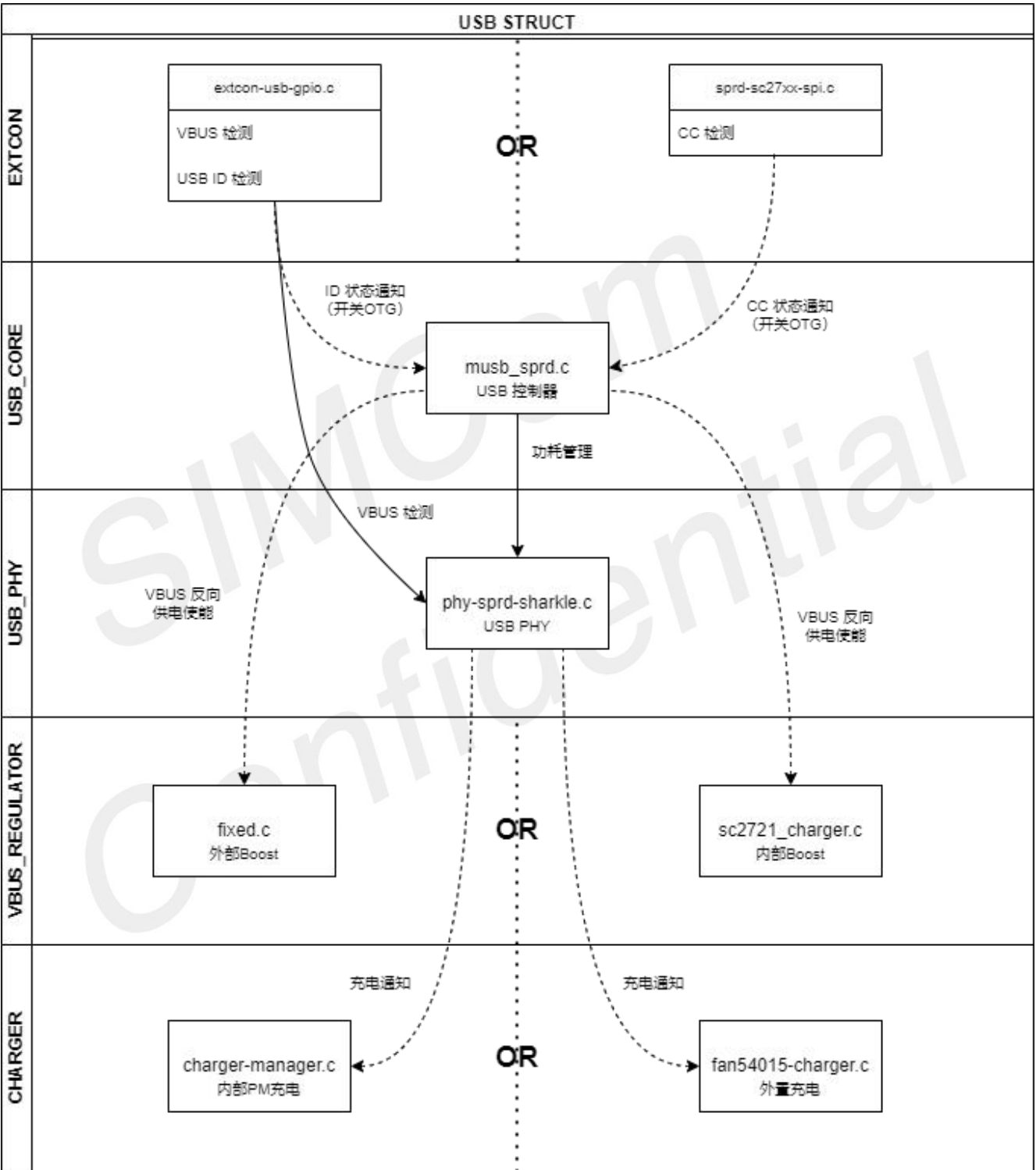


图 1: USB 架构框图

3 Micro USB 的软硬件设计

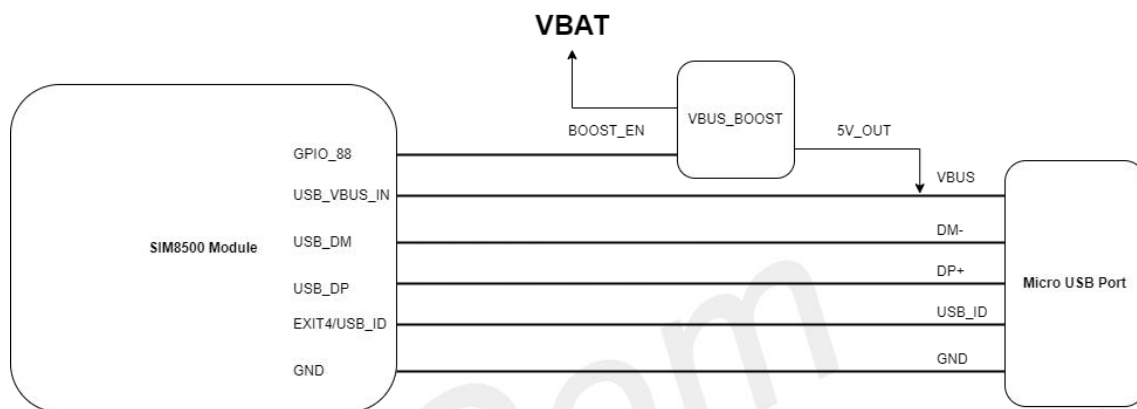


图 2: Micro USB 硬件架构框图

软件配置关键点:

```

1  extcon_gpio: extcon-gpio {
2      compatible = "linux,extcon-usb-gpio";
3      vbus-gpio = <&pmic_eic 0 GPIO_ACTIVE_HIGH>; /* VBUS 检测GPIO */
4      id-gpio = <&eic_debounce 4 GPIO_ACTIVE_HIGH>; /* USB ID 检测GPIO */
5  };
6
7  regulator@0 {
8      cm-regulator-name = "vddgen0";
9      cable@0 {
10         cm-cable-name = "USB";
11         extcon = <&extcon_gpio>;
12     };
13 };
14
15 vddvbus_ldo: vddvbus_ldo {
16     compatible = "regulator-fixed";
17     regulator-name = "vddvbus_ldo";
18     gpio = <&ap_gpio 88 0>;
19     enable-active-high;
20 };
21
22 &hsphy {
23     vdd-supply = <&vddusb33>;
24     extcon = <&extcon_gpio>; /* usb 通知器 */
25 };
26
27 &usb {
28     extcon = <&extcon_gpio>; /* usb 通知器 */
29     vbus-supply = <&vddvbus_ldo>; /* 外部VBUS Boost Regulator */
30 };
31
32 &pmic_typec {
33     status = "disabled"; /* 关闭 TYPE-C检测 */
34 };
35
36 &pmic_charger {
37     status = "okay";
38     phys = <&hsphy>; /* 由usb phy 驱动上报充电状态 */
39     monitored-battery = <&bat>;
40 };
41

```


4 TYPE-C 的软硬件设计

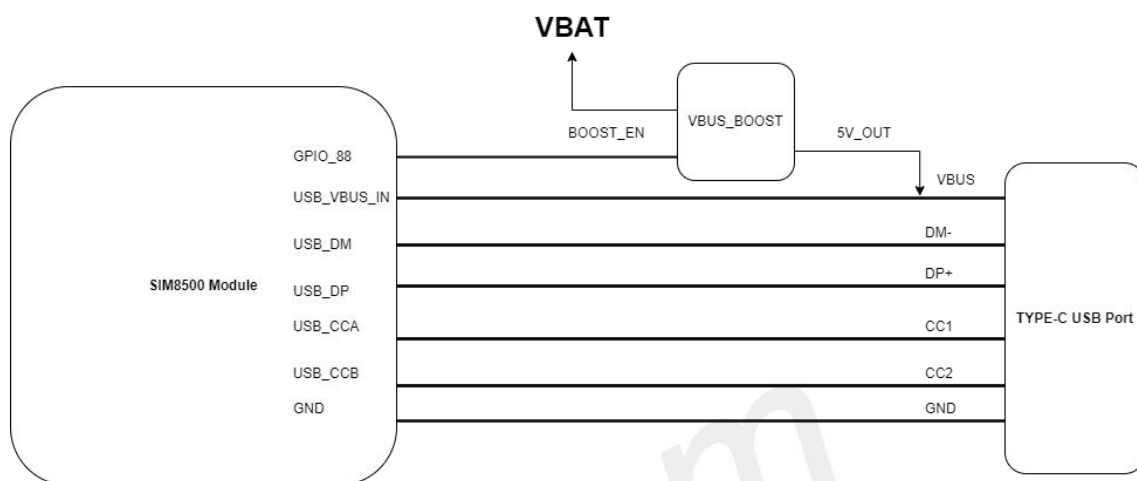


图 3: TYPE-C 硬件架构框图

软件配置关键点：

```

1  extcon_gpio: extcon-gpio {
2  |   compatible = "linux,extcon-usb-gpio";
3  |   vbus-gpio = <&pmic_eic 0 GPIO_ACTIVE_HIGH>; /* VBUS 检测GPIO */
4  | };
5
6  regulator@0 {
7  |   cm-regulator-name = "vddgen0";
8  |   cable@0 {
9  |   |   cm-cable-name = "USB";
10 |   |   extcon = <&pmic_typec>;
11 |   };
12 | };
13
14  vddvbus_ldo: vddvbus_ldo {
15  |   compatible = "regulator-fixed";
16  |   regulator-name = "vddvbus_ldo";
17  |   gpio = <&ap_gpio 88 0>;
18  |   enable-active-high;
19  | };
20
21  &hsphy {
22  |   vdd-supply = <&vddusb33>;
23  |   extcon = <&extcon_gpio>; /* usb 通知器 */
24  | };
25
26  &usb {
27  |   extcon = <&extcon_gpio>, <&pmic_typec>;
28  |   vbus-supply = <&vddvbus_ldo>;
29  | };
30
31  &pmic_typec {
32  |   status = "ok"; /* 使能 TYPE-C检测 */
33  | };
34
35  &pmic_charger {
36  |   status = "okay";
37  |   phys = <&hsphy>; /* 由usb phy 驱动上报充电状态 */
38  |   monitored-battery = <&bat>;
39  | };
40

```

5 USB 转以太网的软硬件设计

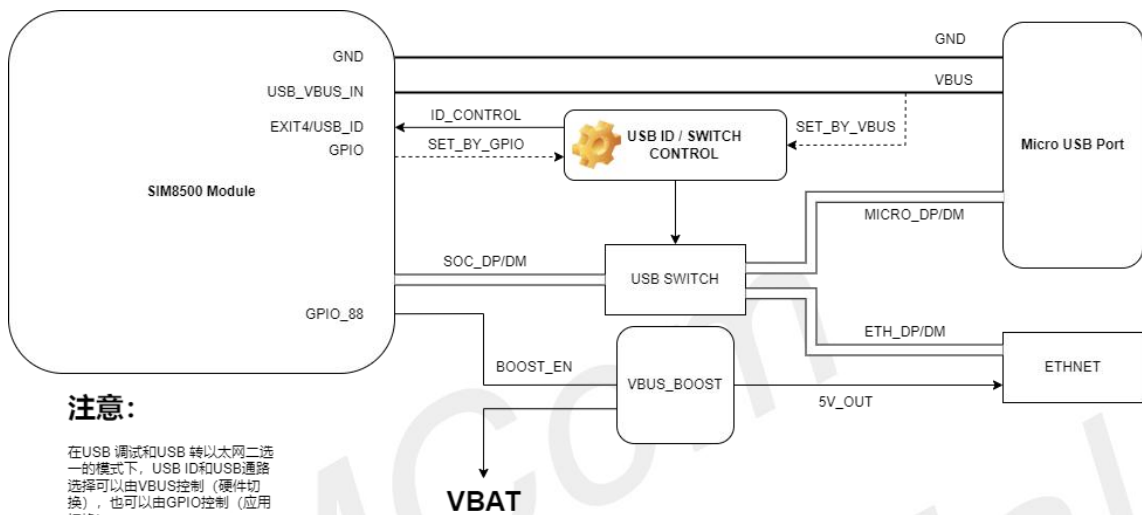


图 4: USB 转以太网 硬件架构框图

USB 转以太网软件配置与 Micro USB 相同。USB 开启以太网由硬件电路控制，控制端可以是 VBUS，也可以是 GPIO。当控制端为 VBUS 时，插入 USB 调试线，硬件控制电路将 USB ID 引脚释放，并切换 USB 通路到 Micro USB 端口，以开启 Device 模式方便与电脑连接。当控制端为 GPIO 时，可以由上层软件控制 GPIO 切换和打开以太网功能，方便用户管理。

其他注意事项：

SIM8500 系列使用了展锐 SOC，该 SOC 的主 USB 端口*（1）不支持 USB HUB 拓展端口，不支持戴尔鼠标。

（1）主 USB 端口：SIM8500E 有两个 USB 端口，主 USB 端口不支持 HUB，拓展 USB 端口支持 HUB 但只能处于 HOST 模式。

常用 USB 转以太网芯片表格：

网卡芯片	网络规格
AX88179	千兆以太网
AX88772	千兆以太网
RTL8153	千兆以太网
RTL8152	百兆以太网
DM9621	百兆以太网

6 USB PHY Turning 参数导入

当 USB 眼图测试出现问题或者 USB PHY 信号质量需要进行调优时，就需要使用到参数导入。首先参考文档《USB20_HS_Device_Mode_Test_SOP》，修改对应寄存器的值。

测试步骤



平台：
SC9832E/SC9820E/
SL8541E/SL8521E

- 1、修改寄存0x4043004C的值bit[28:27]，可以改变上升沿下降沿
- 2、修改寄存0x4043004C的值bit[26:21]，可以改变幅度

5.32.5.101 analog_usb20_USB20_TRIMMING

0x0000004C						analog_usb20_USB20_TRIMMING(0x03EF4301)										analog_usb20_USB20_TRIMMING					
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16					
Name	Reserved			analog_usb20_USB20_TUNEHSAMP			analog_usb20_USB20_TFREGRS					analog_usb20_USB20_TFHSRES									
Type	RO			RW			RW					RW									
Reset	0	0	0	0	0	0	1	1	1	1	1	0	1	1	1	1					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Name	analog_usb20_USB20_TUNERISE		analog_usb20_USB20_TUNEOTG			analog_usb20_USB20_TUNEDSC		analog_usb20_USB20_TUNESQ				analog_usb20_USB20_TUNEQ			analog_usb20_USB20_TUNEPLLS						
Type	RW		RW			RW		RW				RW			RW						
Reset	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1					

修改寄存器的值可以在 adb 中使用 lookat 工具，该工具由 SIM8500/device/sprd/sharkle/common/DeviceCommon.mk 中的 PRODUCT_PACKAGES_DEBUG 这一宏控制编译。将该宏改成 PRODUCT_PACKAGES 后即可将 lookat 编译进系统 bin 目录。

```

808 813 ..... v15310_parameter \
809 814 ..... Before
810 ..... PRODUCT_PACKAGES_DEBUG += \
815 ..... PRODUCT_PACKAGES += \
811 816 ..... iperf \
812 817 ..... vibffttest \

```

Lookat 工具的使用方法：

```
adb root
adb shell
```

读取 0x4043004C 地址的 1 个字节数据：

```
lookat -l 1 0x4043004C
```

写入数据到 0x4043004C

```
lookat -s xxx 0x4043004C
```

xxx 就是改为你想要的值

驱动增加导入参数参考如下：

```

1 diff --git a/arch/arm/boot/dts/sharkle.dtsi b/arch/arm/boot/dts/sharkle.dtsi
2 index f401a47ba5b2..1a57d3b4a3b2 100644
3 --- a/arch/arm/boot/dts/sharkle.dtsi
4 +++ b/arch/arm/boot/dts/sharkle.dtsi
5 @@ -491,6 +491,7 @@
6 ..... reg = <0x20e00000 0x3030>;
7 ..... reg-names = "phy_glb_regs";
8 ..... sprd,syscon-enable = <&aon_apb_regs>;
9 + ..... sprd,syscon-anag8 = <&anlg_phy_g8_regs>;
10 ..... sprd,tune-value = <0x0005af33>;
11 ..... sprd,vdd-voltage = <3300000>;
12 ..... #phy-cells = <0>;
13 diff --git a/drivers/usb/phy/phy-sprd-sharkle.c b/drivers/usb/phy/phy-sprd-sharkle.c
14 index 5ad5b2fc4584..34fcc661c292 100644
15 --- a/drivers/usb/phy/phy-sprd-sharkle.c
16 +++ b/drivers/usb/phy/phy-sprd-sharkle.c
17 @@ -36,6 +36,7 @@ struct sprd_hsphy {
18 ..... struct regulator ..... *vdd;
19 ..... struct regmap ..... *hsphy_glb;
20 ..... struct regmap ..... *pmic;
21 + ..... struct regmap ..... *ana_g8;
22 ..... u32 ..... vdd_vol;
23 ..... u32 ..... phy_tune;
24 ..... atomic_t ..... reset;
25 @@ -176,6 +177,8 @@ static int sprd_hsphy_init(struct usb_phy *x)
26 ..... reg1 |= MASK_AP_AHB_USB20_SAMPLER_SEL;
27 ..... writel_relaxed(reg1, phy->base + REG_AP_AHB_OTG_CTRL1);
28 .....
29 + ..... regmap_write(phy->ana_g8, 0x4c, 0x1EF4300); //0x1EF4300这个改成你调整之后的值即可
30 + .....
31 ..... if (!atomic_read(&phy->reset)) {
32 ..... /* USB PHY write register need to delay 2ms~3ms */
33 ..... usleep_range(2000, 3000);
34 @@ -420,6 +423,13 @@ static int sprd_hsphy_probe(struct platform_device *pdev)
35 ..... if (!otg)
36 ..... return -ENOMEM;
37 .....
38 + ..... phy->ana_g8 = syscon_regmap_lookup_by_phandle(dev->of_node,
39 + ..... "sprd,syscon-anag8");
40 + ..... if (IS_ERR(phy->ana_g8)) {
41 + ..... dev_err(&pdev->dev, "ap USB anag8 syscon failed!\n");
42 + ..... return PTR_ERR(phy->ana_g8);
43 + ..... }
44 + .....
45 ..... hsphy_glb = syscon_regmap_lookup_by_phandle(dev->of_node,
46 ..... "sprd,syscon-enable");
47 ..... if (IS_ERR(hsphy_glb)) {
48 .....

```

如何验证导入结果？使用 lookat 工具将 0x4043004C 的值读出并与驱动中所预设的值进行比对。