



Unisoc Confidential For hiar

UDS710_UDX710 USB 客制化指导手册

文档版本
发布日期

V1.1
2020-10-10

版权所有 © 紫光展锐（上海）科技有限公司。保留一切权利。

本文件所含数据和信息都属于紫光展锐（上海）科技有限公司（以下简称紫光展锐）所有的机密信息，紫光展锐保留所有相关权利。本文件仅为信息参考之目的提供，不包含任何明示或默示的知识产权许可，也不表示有任何明示或默示的保证，包括但不限于满足任何特殊目的、不侵权或性能。当您接受这份文件时，即表示您同意本文件中内容和信息属于紫光展锐机密信息，且同意在未获得紫光展锐书面同意前，不使用或复制本文件的整体或部分，也不向任何其他方披露本文件内容。紫光展锐有权在未经事先通知的情况下，在任何时候对本文件做任何修改。紫光展锐对本文件所含数据和信息不做任何保证，在任何情况下，紫光展锐均不负任何与本文件相关的直接或间接的、任何伤害或损失。

请参照交付物中说明文档对紫光展锐交付物进行使用，任何人对紫光展锐交付物的修改、定制化或违反说明文档的指引对紫光展锐交付物进行使用造成的任何损失由其自行承担。紫光展锐交付物中的性能指标、测试结果和参数等，均为在紫光展锐内部研发和测试系统中获得的，仅供参考，若任何人需要对交付物进行商用或量产，需要结合自身的软硬件测试环境进行全面的测试和调试。

Unisoc Confidential For hiar

紫光展锐（上海）科技有限公司



前言

概述

本文简要介绍了 USB 的软硬件架构和关键部件功能，列举了常见的 USB 客户问题和解决方案，并对客制化遇到的问题给出说明和修改方案。

读者对象


USB 开发及测试人员。

缩略语

缩略语	英文全名	中文解释
USB	Universal Serial Bus	通用串行总线
MTP	Media Transfer Protocol	媒体传输协议
DTS	Device Tree Source	设备树源码
ADB	Android Debug Bridge	安卓调试桥

符号约定

在本文中可能出现下列标志，它所代表的含义如下。

符号	说明
 说明	用于突出重要/关键信息、补充信息和小窍门等。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害。

变更信息

文档版本	发布日期	修改说明
V1.0	2020-9-9	第一次正式发布。

文档版本	发布日期	修改说明
V1.1	2020-10-10	更新模板、优化表达

关键字

USB、Type-C、Micro-B。

Unisoc Confidential For hiar

目 录

1 USB 介绍.....	1
1.1 USB 硬件介绍	1
1.2 USB 软件介绍	1
2 USB 客制化.....	4
2.1 客制化 USB Interface	4
2.1.1 修改 USB Device Tree.....	4
2.1.2 修改 pinmap	4
2.2 客制化 USB vendor/product	5
2.2.1 客制化方法	5
2.2.2 举例.....	5
3 常见问题.....	7
3.1 VtsHalUsbl_0 Target 模块测试失败.....	7
3.2 MTP 枚举失败.....	8
3.3 BBAT OTG 测试.....	8
3.4 手动切换 USB function	9

Unisoc Confidential For hiar

1 USB 介绍

1.1 USB 硬件介绍

从硬件角度来看，USB 包括三个部分：USB controller、USB PHY 和 USB interface。

- USB controller

USB controller 是 ASIC 逻辑实现的 USB 协议逻辑电路，固化在芯片内部，实现了 USB 协议处理。展锐平台目前支持三种不同的 USB Controller：DWC2、DWC3、MUSB。

说明

MUSB 不支持 USB HUB，DWC2、DWC3 支持 USB HUB。

- USB PHY

USB PHY 完成 USB 信号的数模转换。展锐平台支持三个 USB PHY：

- Synopsys USB3.0 PHY
- SPRD USB2.0 PHY
- INTEL USB2.0 PHY

- USB interface

这部分主要指 USB 对外的接口，分为 Type-C 和 Micro-B 接口。

- Type-C 接口与芯片相关，在芯片内由 Type-C Controller 实现 Type-C 协议。
- Micro-B 接口不同于 Type-C，在芯片内部没有可依赖的逻辑电路，而是依赖 EIC 检测电平的变化来达到检测 USB Cable 插拔状态的目的。

对于一款指定的芯片，USB controller 和 USB PHY 在芯片设计阶段就已固化，是不可以客制化的。但 USB interface 的实现依赖于产品 PCB 的设计，是可以客制化的。

1.2 USB 软件介绍

USB Driver

USB Driver 包含三个不同模块：

- UDC driver

这部分与 HW 的 USB controller 相关，直接驱动 USB controller。其包含 3 种不同的 UDC driver，分别是 DWC2、DWC3、MUSB，源代码对应的路径如下：

- DWC2: bsp/kernel/kernel4.14/drivers/usb/dwc2
- DWC3: bsp/kernel/kernel4.14/drivers/usb/dwc3
- MUSB: bsp/kernel/kernel4.14/drivers/usb/musb

- USB PHY driver

统一放在 bsp/kernel/kernel4.14/drivers/usb/phy 目录下。

- USB interface driver
 - Type-C driver 统一放在 bsp/kernel/kernel4.14/drivers/usb/typec 目录下。
 - Micro-B 由于 HW 不存在独立的逻辑，没有单独路径，在每个 UDC driver 中都包含了 Micro-B source code。

USB device tree

除了与 HW component 相关的 driver 外，USB driver 的可变参数统一放在 device tree 进行维护。在 Kernel4.14 中，需添加 extcon 来触发 USB 机制。以下就是 USB DT node:

- bsp/kernel/kernel4.14/arch/arm64/boot/dts/sprd/roc1.dtsi

USB DTS 配置:

```
usb3: usb3@20f00000 {  
  
    compatible = "sprd,roc1-dwc3";  
    reg = <0 0x20f00000 0 0x100000>;  
    status = "disabled";  
    #address-cells = <2>;  
    #size-cells = <2>;  
    ranges;  
    clocks = <&ipa_gate CLK_IPA_USB_EB>,  
            <&ipa_gate CLK_IPA_USB_SUSPEND_EB>,  
            <&ipa_gate CLK_IPA_USB_REF_EB>,  
            <&ipa_clk CLK_USB_REF>,  
            <&g12_pll CLK_TWPLL_24M>,  
            <&ext_32k>;  
    clock-names = "core_clk", "ref_clk", "susp_clk",  
                "ipa_usb_ref", "ipa_usb_ref_source",  
                "ipa_usb_ref_default";  
    usb-phy = <&ssphy>, <&ssphy>;  
    power-domains = <&pd_ipa_sys>;  
    dwc3@20f00000 {  
        compatible = "snps,dwc3";  
        reg = <0 0x20f00000 0 0x100000>;  
        interrupts = <GIC_SPI 135 IRQ_TYPE_LEVEL_HIGH>;  
        interrupt-names = "irq";  
        usb-phy = <&ssphy>, <&ssphy>;  
        usb-pam = <&pamu3>;  
        phy_type = "utmi_wide";  
        snps,usb3-u1u2-disable;
```

```

snps,dis_u3_susphy_quirk;
snps,dis_u2_susphy_quirk;
snps,overlap_check_quirk;
snps,ep_out_aligned_size_quirk;
snps,disconnect_event_quirk;
snps,usb3_slow_suspend;
snps,host_suspend_capable;
snps,dis-del-phy-power-chg-quirk;

};
};

```

- bsp/kernel/kernel4.14/arch/arm64/boot/dts/sprd/ud710-2h10.dts

- Extcon 配置:

```

extcon_gpio: extcon-gpio {
compatible = "linux,extcon-usb-gpio";
vbus-gpio = <&pmic_eic 0 GPIO_ACTIVE_HIGH>;
};

```

- USB DTS 配置:

```

&ssphy {
vdd-supply = <&vddusb33>;
extcon = <&extcon_gpio>;
status = "okay";
};

&usb3 {
extcon = <&extcon_gpio>, <&pmic_typec>;
vbus-supply = <&vddvbus>;
status = "okay";
};

```

以上是 Type-C 接口的配置方法，如果客户使用的是 Micro-B 接口，需要修改两处：

- ssphy 的 extcon 字段为 Micro-B 接口的 vbus-gpio
- usb3 的 extcon 字段为 Micro-B 接口的 vbus-gpio

2 USB 客制化

2.1 客制化 USB Interface

对于一款指定的芯片，只有 USB Interface 可以客制化。USB Controller 和 USB PHY 由 ASIC 设计时固化在芯片内部。展锐的参考方案中默认是 Type-C 接口。当客户选择 Micro-B 接口，就需要客制化 USB Interface。

客制化 USB interface driver 时，需要先熟悉硬件设计上使用哪个 pin 作为 USB-ID 功能，然后修改 USB DT node 和 USB-ID pin 的 pinmap setting 即可。

下文是 Micro-B 配置方法和示例。

2.1.1 修改 USB Device Tree

在 device tree 中添加 extcon-gpio 节点，并将 extcon-gpio 节点中的 vbus-gpio 和 id-gpio 字段修改为 Micro-B 接口对应的 gpio。以下是一个修改 device tree 的示例：

bsp/kernel/kernel4.14/arch/arm64/boot/dts/sprd/ud710-2h10.dts

```
extcon_gpio: extcon-gpio {
    compatible = "linux,extcon-usb-gpio";
    vbus-gpio = <&pmic_eic 0 GPIO_ACTIVE_HIGH>;
+   id-gpio = <&ap_eic 4 GPIO_ACTIVE_HIGH>;
};

&usb3 {
-   extcon = <&extcon_gpio>, <&pmic_typec>;
+   extcon = <&extcon_gpio>;
    vbus-supply = <&vddvbus>;
    status = "okay";
};
```

2.1.2 修改 pinmap

由于对应的 pin 作为 EIC 功能使用，所以需要对应 board 的 pinmap 设置。

文件路径在 bsp/bootloader/u-boot15/board/spreadtrum/ud710_2h10/pinmap-ud710.c

```
diff --git a/board/spreadtrum/ud710_2h10/pinmap-ud710.c b/board/spreadtrum/ud710_2h10/pinmap-ud710.c
```

```
index 8fa55034a0..a85ad2b5f1 100755
```

```
--- a/board/spreadtrum/ud710_2h10/pinmap-ud710.c
+++ b/board/spreadtrum/ud710_2h10/pinmap-ud710.c
```

@@ -29,6 +29,7 @@

```
static pinmap_t pinmap[]={
+{ REG_MISC_PIN_KEYIN2,
BITS_PIN_DS(1)|BIT_PIN_NULL|BIT_PIN_WPU|BIT_PIN_SLP_AP|BIT_PIN_SLP_WPU|BIT_PIN_SLP_IE },
```

说明

作为 USB ID 的 pin 必须选择支持 EIC 功能的 pin。因为 EIC 支持防抖功能，可以避免由于抖动导致的 USB event 误报，从而保证 USB cable 检测的稳定性。

2.2 客制化 USB vendor/product

根据 USB 规范的规定，所有的 USB 设备都有供应商 ID（VID）和产品识别码（PID）。主机通过不同的 VID 和 PID 来区别不同的设备。VID 和 PID 长度均为两个字节。

- 供应商 ID（VID）：由供应商向 USB 执行论坛申请，每个供应商的 VID 是唯一的。
- 产品识别码（PID）：PID 由供应商自行决定。

PID/VID 唯一标识一个设备。除了使用 VID/PID 编码来表示 vendor 和 product，系统还提供了明文字符串描述 vendor 和 product，也是可以客制化的。

2.2.1 客制化方法

USB driver 提供给 USER space configuration node 来设置 USB vendor 和 product 的信息。其节点路径如下：

- VID node: /config/usb_gadget/g1/idProduct
- PID node: /config/usb_gadget/g1/idVendor
- Vendor string: /config/usb_gadget/g1/configs/b.1/strings/0x409/manufacturer
- Product string: /config/usb_gadget/g1/configs/b.1/strings/0x409/product

每一个节点都需要配置。节点配置文件：init.common.usb.rc。

2.2.2 举例

配置文件：device/sprd/rocl/common/rootdir/root/init.common.usb.rc。

配置信息如下：

```
property:sys.usb.ffs.ready=1 && property:ro.bootmode=iq && property:sys.usb.configfs=1
tprop sys.usb.mode iq
ite /config/usb_gadget/g1/idVendor 0x1782
ite /config/usb_gadget/g1/idProduct 0x4012
ite /config/usb_gadget/g1/bcdDevice 0xffff
ite /config/usb_gadget/g1/bDeviceClass 0
ite /config/usb_gadget/g1/configs/b.1/strings/0x409/configuration "ptp_adb_vser"
/config/usb_gadget/g1/configs/b.1/f1
/config/usb_gadget/g1/configs/b.1/f2
```

从设备端可以获取到相应的信息，如下所示：

```
ud710_2h10:/config/usb_gadget/g1 # ls
UDC          bDeviceProtocol bMaxPacketSize0 bcdUSB  functions idVendor strings
bDeviceClass bDeviceSubClass bcdDevice        configs idProduct os_desc
```

Unisoc Confidential For hiar

3 常见问题

3.1 VtsHalUsb1_0 Target 模块测试失败

原因

VtsHalUSB1.0 test case 需要硬件支持 Type-C 接口。如果客户的手机是 Micro-B 接口，则会测试失败。

解决办法

取消 VTS 关于 USB 的测试项。在 device/sprd/roc1/common/DeviceCommon.mk 下做如下修改：

步骤 1 关闭 VtsHalUSB1.0 的 HAL 层的 service。

```
diff --git a/common/DeviceCommon.mk b/common/DeviceCommon.mk
index 8ff3bb3..28b88d3 100755
--- a/common/DeviceCommon.mk
+++ b/common/DeviceCommon.mk
@@ -114,9 +114,6 @@ PRODUCT_PACKAGES += \
    PRODUCT_PACKAGES += \
        android.hardware.renderscript@1.0-impl

-#add USB vts service
-PRODUCT_PACKAGES += \
-    android.hardware.usb@1.1-service
```

步骤 2 如下所示，删除 device/sprd/\$(platform_name)/common/manifest_main.xml 中标红部分。

```
diff --git a/common/manifest_main.xml b/common/manifest_main.xml
index 062912b..48552d0 100644
--- a/common/manifest_main.xml
+++ b/common/manifest_main.xml
@@ -198,16 +198,6 @@
     </interface>
     <fqname>@2.0::ISoundTriggerHw/default</fqname>
 </hal>
-    <hal format="hidl">
-        <name>android.hardware.usb</name>
-        <transport>hwbinder</transport>
-        <version>1.1</version>
-        <interface>
-            <name>IUsb</name>
-            <instance>default</instance>
-        </interface>
-        <fqname>@1.1::IUsb/default</fqname>
-    </hal>
    <hal format="hidl">
        <name>android.hardware.vibrator</name>
        <transport>hwbinder</transport>
```

步骤 3 如下所示，删除 device/sprd/\$(platform_name)/common/sepolicy/file_contexts 中标红部分。

```
diff --git a/common/sepolicy/file_contexts b/common/sepolicy/file_contexts
index 9e6db3e..2d70f18 100755
--- a/common/sepolicy/file_contexts
+++ b/common/sepolicy/file_contexts
@@ -308,9 +308,6 @@
 #add for connmgr
 /((vendor|system/vendor)/bin/hw/vendor/.sprd/.hardware/.connmgr@1\..0-service u:object_r:hal_connmgr_default_exec:s0

-#for USB vts1.1
-/(vendor|system/vendor)/bin/hw/android/.hardware/.usb@1\..1-service u:object_r:hal_usb_default_exec:s0
-
+/(vendor|system/vendor)/bin/hw/android/.hardware/.bluetooth@1\..0-service\..uniso
+c u:object_r:hal_bluetooth_default_exec:s0
+/(vendor|system/vendor)/bin/hw/android/.hardware/.bluetooth@1\..0-service\..uniso
+c-lazy u:object_r:hal_bluetooth_default_exec:s0
```

---结束

3.2 MTP 枚举失败

MTP buffer 在枚举时动态分配，如果分配失败可以通过减小 buffer size 和 buffer count 的方法规避这个问题。文件路径：kernel/drivers/usb/gadget/function/f_mtp.c

下图中绿框分别是 buffer size 和 buffer count。

```
#define MTP_BULK_BUFFER_SIZE 65536
#define MTP_MINIMUM_BUFFER_SIZE 16384
#define INTR_BUFFER_SIZE 28
#define MAX_INST_NAME_LEN 40

/* String IDs */
#define INTERFACE_STRING_INDEX 0

/* values for mtp_dev.state */
#define STATE_OFFLINE 0 /* initial state, disconnected */
#define STATE_READY 1 /* ready for userspace calls */
#define STATE_BUSY 2 /* processing userspace calls */
#define STATE_CANCELED 3 /* transaction canceled by host */
#define STATE_ERROR 4 /* error from completion routine */

/* number of tx and rx requests to allocate */
#define TX_REQ_MAX 4
#define RX_REQ_MAX 2
```

3.3 BBAT OTG 测试

开启 OTG 测试功能，步骤如下：

步骤 1 确定以下文件中是否有 OTG 项。如果有且其值为 0，需要将其值改为 1。

@device/sprd/roc1/ud710_2h10/rootdir/prodnv/BBAT.conf

```
...
-- 0 OTG
++ 1 OTG
...
```

步骤 2 预编译动态库

@device/sprd/rocl/common/DeviceCommon.mk

```
...
#bbat test
PRODUCT_PACKAGES += autotestsensorinfo \
                    autotestgps \
                    autotestsim \
++                    autotestotg \
                    autotestcard
...
```

步骤 3 修正"host_enable"节点，平台默认代码多加了一组双引号，具体以客户项目实际 path 为准。

@vendor/sprd/proprieties-source/autotest/interface/otg/otg.cpp

```
static int get_dev_node_path(char *node_name, char *path)
{
    if (!node_name && !path) {
        LOGD("%s, invalid parameter\n", __func__);
    }

    -- sprintf(path, "/sys/devices/platform/soc/soc:ap-ahb/20200000.usb/%s", node_name);
    ++ sprintf(path, "/sys/devices/platform/soc/soc:ap-ahb/20200000.usb/%s", node_name); //此处改为项目实际path
    LOGD("%s, path: %s", __func__, path);
    return 0;
}
```

说明

- --表示删除当行代码，--不能出现在代码中。
- ++表示增加当行代码，++不能出现在代码中。

----结束

3.4 手动切换 USB function

```
adb root
adb remount
adb shell
setprop sys.usb.config "functions string"
```

function string 的常用组合如下：

```
Adb
mtp,adb
ptp,adb
```

cdrom.adb
rndis.adb
midi.adb

更多组合，请参考以下两个 USB 相关的 rc 文件：

- system/core/rootdir/init.usb.configfs.rc
- device/sprd/rocl/common/rootdir/root/init.common.usb.rc

Unisoc Confidential For hiar