



Linux USB3 开发指南

版本号: 1.7
发布日期: 2025.4.1

版本历史

版本号	日期	制/修订人	内容描述
1.0	2023.03.13	AWA1881	初始版本。
1.1	2023.04.26	AWA1881	1. 更新部分配置项； 2. 支持 Type-C 接口的 OTG 切换。
1.2	2023.09.28	AWA1881	根据内部开发板支持完整 USB3.0 DRD 方案。
1.3	2024.03.14	AWA1881	更新适配范围、调试方法、测试方法、FAQ 等。
1.4	2024.08.15	AWA1927	更新适配范围、配置项、FAQ 等。
1.5	2024.11.26	AWA1881	1. 更新适用范围、相关术语及模块硬件介绍部分描述。 2. 更新 dts 配置示例及注意事项。 3. 更新 Gen2 日志及测试说明。 4. 更新性能测试示例。
1.6	2025.03.28	AWA1881	1. 更新配置示例。
1.7	2025.4.1	AWA1927	1. 更新适用范围。 2. 更新 FAQ。

目 录

1 前言	1
1.1 文档简介	1
1.2 目标读者	1
1.3 适用范围	1
2 模块介绍	2
2.1 模块功能介绍	2
2.2 相关术语介绍	2
2.3 模块硬件介绍	2
2.4 模块配置介绍	3
2.4.1 kernel menuconfig 配置说明	3
2.4.2 Device Tree 配置说明	7
2.4.2.1 路径定义	7
2.4.2.2 配置示例	7
2.4.2.3 注意事项	8
2.4.3 board.dts 配置说明	8
2.4.3.1 路径定义	8
2.4.3.2 配置示例	8
2.4.3.3 注意事项	14
2.4.3.4 扩展说明	14
2.5 源码结构说明	14
3 Gadget 配置与使用	16
3.1 Gadget 内核配置	16
3.2 Gadget 配置流程	16
3.3 Gadget 功能切换	18
4 日志分析	19
4.1 驱动加载日志分析	19
4.2 Device 日志分析	20
4.3 Host 日志分析	21
5 调试方法	25
5.1 常用内核调试接口	25
5.1.1 USB Host 总线调试接口	25
5.1.2 USB Host Debugfs 调试接口	26
5.1.3 USB Trace 类调试接口	27
5.1.4 USB Gadget 调试接口	27
5.2 USB3.0 调试方法	28

5.2.1	DWC3 调试节点	28
5.2.2	XHCI 调试节点	29
5.3	Type-C 调试方法	30
5.3.1	PS8743 配置说明	30
5.3.2	GMA340 配置说明	30
5.3.3	CC Logic 调试	30
6	测试方法	32
6.1	USB3.0 U3 信号测试步骤	32
6.2	USB3.0 U2 信号测试步骤	33
7	FAQ	34
7.1	USB 常见问题思路分析	34
7.1.1	ADB 无法识别为超速	34
7.1.2	mass_storage 无法识别为超速	34
7.1.3	device 配置成 3+ 复合功能时异常	35
7.1.4	dwc3_sunxi_plat 驱动加载失败	35
7.1.5	xhci 注册失败	35
7.1.6	MTP 从 PC 拷贝文件至样机进度条不动	35
7.1.7	USB 主机功能驱动异常	37
7.1.8	如何确定 USB 外设的枚举速率	37
7.1.9	USB 传输速率低	37
8	附录	38
8.1	Gadget 配置示例	38
8.1.1	小机做 mass storage	38
8.1.2	小机做 cdrom	38
8.1.3	小机做 UAC1	39
8.1.4	小机做 UAC2	40
8.1.5	小机做 UVC	40
8.1.6	小机做 HID	41
8.1.7	小机做 rndis	42
8.1.8	小机做 acm	43
8.1.9	小机做 adb	43
8.2	性能测试示例	44
8.2.1	挂载设备	44
8.2.2	写性能测试	45
8.2.3	读性能测试	45
8.2.4	测试节点	45

插图

图 2-1	内核 menuconfig 根菜单	3
图 2-2	内核 menuconfig XHCI 菜单	4
图 2-3	内核 menuconfig DWC3 驱动菜单	5
图 2-4	Allwinner USB DWC3 驱动菜单	6
图 2-5	Allwinner USB PHY 驱动菜单	6
图 2-6	Allwinner USB DWC3 驱动菜单	6
图 3-1	内核 menuconfig Gadget Support 菜单	16
图 4-1	USB Device Tree Viewer	21
图 7-1	mass_storage	34
图 7-2	USB EQ setting	36



1 前言

1.1 文档简介

介绍 USB3 模块配置和调试方法。

1.2 目标读者

USB 模块开发、维护人员。

1.3 适用范围

表 1-1: 产品、内核版本及驱动文件

产品名称	内核版本	驱动文件
A523	Linux-5.15	drivers/usb/*、bsp/drivers/usb
T527	Linux-5.10	drivers/usb/*、bsp/drivers/usb
T527	Linux-5.15	drivers/usb/*、bsp/drivers/usb
MR527	Linux-5.15	drivers/usb/*、bsp/drivers/usb
A527	Linux-5.15	drivers/usb/*、bsp/drivers/usb
AI985	Linux-5.15	drivers/usb/*、bsp/drivers/usb
H728	Linux-5.15	drivers/usb/*、bsp/drivers/usb
A733	Linux-5.15	drivers/usb/*、bsp/drivers/usb
T736	Linux-5.15	drivers/usb/*、bsp/drivers/usb
MR536	Linux-5.15-origin	drivers/usb/*、bsp/drivers/usb
T536	Linux-5.10-origin	drivers/usb/*、bsp/drivers/usb
A733	Linux-6.6	drivers/usb/*、bsp/drivers/usb
A333	Linux-6.6	drivers/usb/*、bsp/drivers/usb
A537	Linux-6.6	drivers/usb/*、bsp/drivers/usb

* 注：本文档描述仅适用于 Linux-5.10 及以上的内核版本。

2 模块介绍

2.1 模块功能介绍

USB 有主机功能和从设备功能。做主机时，能连接 U 盘、USB 鼠标等 USB 设备；做从设备时，具有 ADB 调试等从设备功能。

AW SOC 通常内置多个 USB 控制器，不同控制器互相独立，USB3.0 控制器包含 xHCI。关于每个平台的具体的 USB 套数、参数、特性、模块结构图等，请参考发布文档中《XXX_User_Manual_Vx.x.pdf》对应模块的描述。

2.2 相关术语介绍

表 2-1: USB3 模块相关术语介绍：

术语	解释说明
Sunxi	指 Allwinner 的一系列 SoC 硬件平台
USB	Universal Serial Bus, 通用串行总线
xHCI	eXtensible Host Controller Interface, 可扩展的主机控制器接口。
DWC3	DesignWare USB3.0 控制器及驱动

2.3 模块硬件介绍

USB 模块使用 USB0、USB1、USB2、USB3 的形式来标记 USB 接口。某个端口是否支持 DRD 需要结合 Spec 和硬件原理图进行确定，一般仅支持一套 DRD。特别地，对于支持 USB3.0 的平台，一般会支持 USB0 (USB2.0) 和 USB2 (USB3.0) 两套 DRD 控制器。

USB 模块的配置涉及到了信号引用、pin 脚配置、供电配置等，需要结合硬件原理图，具体请参考发布文档中《XXX 硬件设计指南.pdf》对应 USB 模块的描述。

2.4 模块配置介绍

2.4.1 kernel menuconfig 配置说明

在 SDK 根目录下，执行./build.sh menuconfig，进入配置主界面，并按以下步骤操作：

必须选上 USB Support 项后才能支持 USB 模块并进行进一步的配置。

如果需要在支持 USB Host，首先需要选上 <> **Support for Host-side USB** 项，然后会出现如下的 Host 相关的配置。其中，

- **OHCI HCD (USB 1.1) support** 选择 OHCI Driver 的配置；
- **EHCI HCD (USB 2.0) support** 选择 EHCI Driver 的配置；
- **xHCI HCD (USB 3.0) support** 选择 xHCI Driver 的配置。

步骤一，选择 Device Drivers 选项进入下一级配置，如下图所示：

开启 USB support，把 CONFIG_USB_SUPPORT 配置为 Y。

```
.config - Linux/arm64 5.15.41 Kernel Configuration
> Device Drivers
Device Drivers
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenu ----). Highlighted letters are hotkeys. Pressing
Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module <> module capable

[*] Device Tree and Open Firmware support --->
<> Parallel port support ----
[*] Block devices --->
  NVME Support --->
  Misc devices --->
  SCSI device support --->
  Serial ATA and Parallel ATA drivers (libata) ----
  [ ] Multiple devices driver support (RAID and LVM) ----
  <> Generic Target Core Mod (TCM) and ConfigFS Infrastructure ----
  [ ] Fusion MPT device support --->
  IEEE 1394 (FireWire) support --->
[*] Network device support --->
  Input device support --->
  Character devices --->
  I2C support --->
  I3C support ----
  *- SPI support ----
  <> SPMI support --->
  <> HSI support --->
  <> PPS support --->
  PTP clock support --->
  *- Pin controllers --->
  *- GPIO support --->
  <> Dallas' 1-wire support ----
  *- Board level reset or power off ----
  *- Power supply class support --->
  <> Hardware Monitoring support ----
  *- Thermal drivers --->
  *- Watchdog Timer Support --->
  <> Sonics Silicon Backplane support ----
  <> Broadcom specific AMBA ----
  Multifunction device drivers --->
  *- Voltage and Current Regulator Support --->
  *- Remote Controller support --->
  [ ] HDMI CEC drivers ----
  [*] Multimedia support --->
  Graphics support --->
  <*> Sound card support --->
  HID support --->
  [*] USB support --->
  <*> MM/SD/SDIO card support --->
  <> Sony MemoryStick card support ----
  *- LED Support --->
  [ ] Accessibility support ----
v(+)
<-select> <Exit> <?> Help <Save> <Load>
```

图 2-1: 内核 menuconfig 根菜单

步骤二，选择 USB support，进入下级配置，如下图所示：

开启 xHCI HCD (USB 3.0) support，把 CONFIG_USB_XHCI_HCD 配置为 Y。

```
.config - Linux/arm64 5.15.41 Kernel Configuration
> Device Drivers > USB support

Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing
Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module <> module capable

--- USB support
[ ] USB LED Triggers
<> USB ULPI PHY interface support
<> USB GPIO Based Connection Detection Driver
<*> Support for Host-side USB
[*] PCI based USB host interface
[*] USB announce new devices
*** Miscellaneous USB options ***
[*] Enable USB persist by default
[ ] Limit USB device initialization to only a few retries
[ ] Dynamic USB minor allocation
[*] OTG support
[ ] Rely on OTG and EH Targeted Peripherals List
[ ] Disable external hubs
<> USB 2.0 OTG FSM implementation
<> USB port LED trigger
(2) Default autosuspend delay
<> USB Monitor
*** USB Host Controller Drivers ***
<*> Cypress C67x00 HCD support
<*> xHCI HCD (USB 3.0) support
[ ] xHCI support for debug capability
<> Support for additional Renesas xHCI controller with firmware
<*> Generic xHCI driver for a platform device
EHCI HCD (USB 2.0) support
[*] Root Hub Transaction Translators
[*] Improved Transaction Translator scheduling
<> Support for Freescale on-chip EHCI USB controller
<> Generic EHCI driver for a platform device
<> OXU210HP HCD support
<> ISP116X HCD support
<> FOTG210 HCD support
<> MAX3421 HCD (USB-over-SPI) support
<*> OHCI HCD (USB 1.1) support
<*> OHCI support for PCI-bus USB controllers
<> Generic OHCI driver for a platform device
<> UHCI HCD (most Intel and VIA) support
<> SU811HS HCD support
<> R8A66597 HCD support
[ ] HCD test mode support
*** USB Device Class drivers ***
<*> USB Modem (CDC ACM) support
<> USB Printer support
<> USB Wireless Device Management support

V( )
<-select> < Exit > < Help > < Save > < Load >
```

图 2-2: 内核 menuconfig XHCI 菜单

步骤三，选择 DesignWare USB3 DRD Core Support，如下图所示：

开启 DesignWare USB3 DRD Core Support，把 CONFIG_USB_DWC3 配置为 Y。

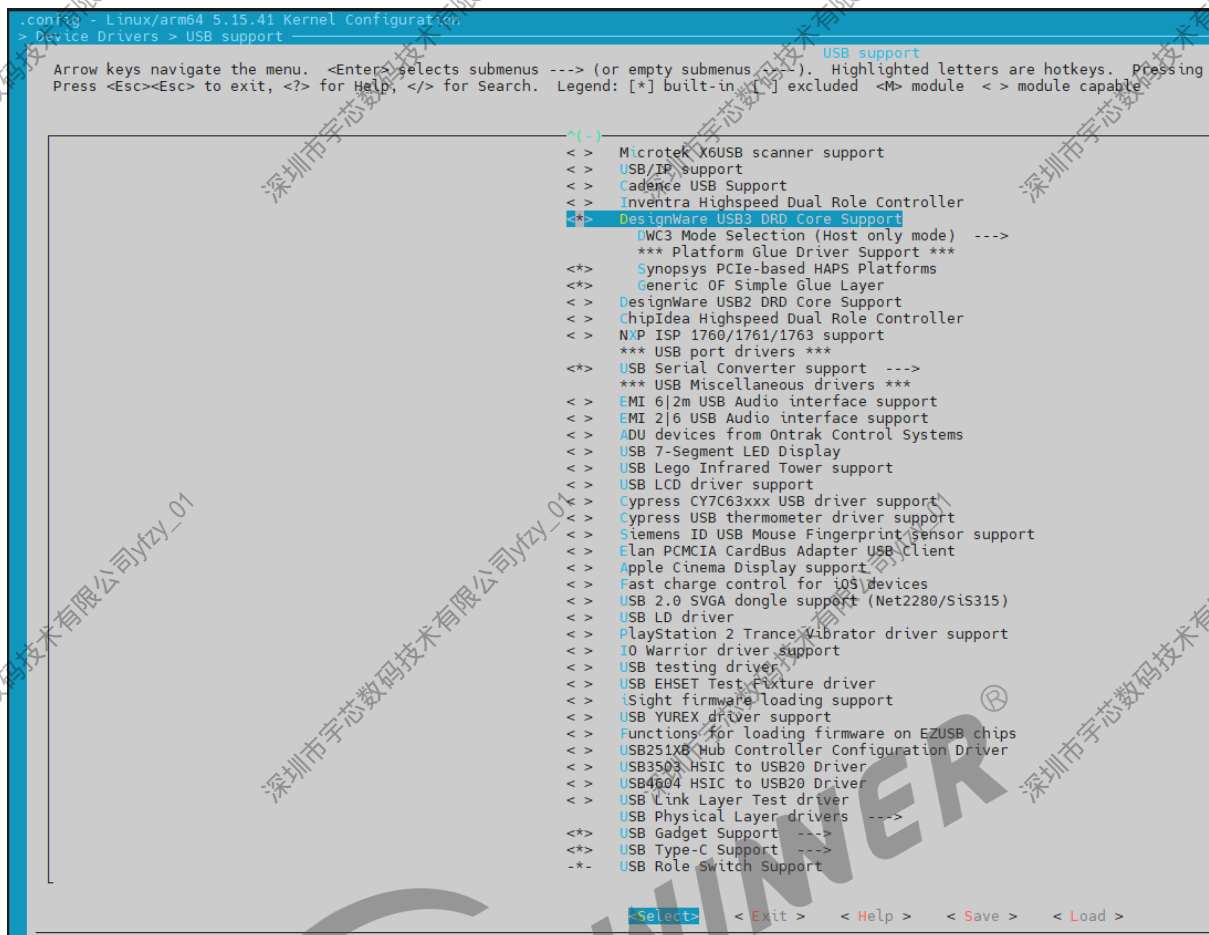


图 2-3: 内核 menuconfig DWC3 驱动菜单

步骤四，回到主界面，依次选择 Allwinner BSP > Device Drivers > USB Drivers > USB DWC3 Drivers，如下图所示：

<1> 开启 Allwinner USB3.0 Dual Role Controller support，把 CONFIG_USB_SUNXI_DWC3 配置为 Y；

<2> 开启 Allwinner SoC USB PHY driver，把 CONFIG_PHY_SUNXI_PLAT 配置为 Y；

<3> 某些自研 PHY 平台需要开启 Allwinner USB2.0/3.0 PHY support，把 CONFIG_USB_SUNXI_PHY 配置为 Y；

<4> 开启 Allwinner USB2.0 AW PHY Driver，把 CONFIG_USB_SUNXI_AWPHY 配置为 Y。

<5> 开启 Allwinner DWC3 Platforms，把 CONFIG_DWC3_SUNXI_PLAT 配置为 Y。

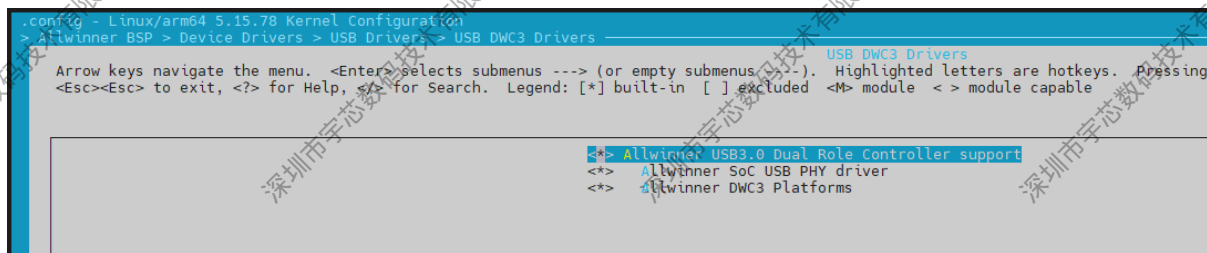


图 2-4: Allwinner USB DWC3 驱动菜单

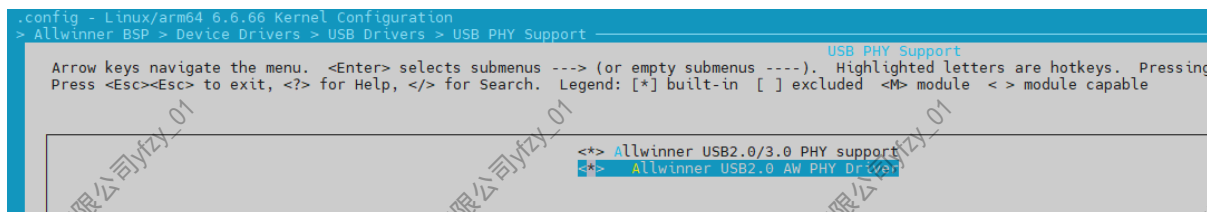


图 2-5: Allwinner USB PHY 驱动菜单

步骤五，如果支持 Type-C Port Controller，还需要按照以下步骤打开对应配置项（注：必须外挂 HUSB311 以及 PS8743 芯片才需要打开，若外挂其他 CC Logic 以及 Switch 芯片可参考对应驱动进行开发）。

回到主界面，依次选择 Allwinner BSP > Device Drivers > USB Drivers > USB Type-C Support，如下图所示：

<1> 开启 Allwinner USB Type-C support，把 CONFIG_USB_SUNXI_TYPEC 配置为 Y；

<2> 开启 Hynetek HUSB311 Type-C chip driver，把 CONFIG_TYPEC_HUSB311 配置为 Y；

<3> 进入 USB Type-C Multiplexer/DeMultiplexer Switch support 页，开启 Parade PS8743 Type-C cross switch driver，把 CONFIG_TYPEC_MUX_PS8743 配置为 Y。

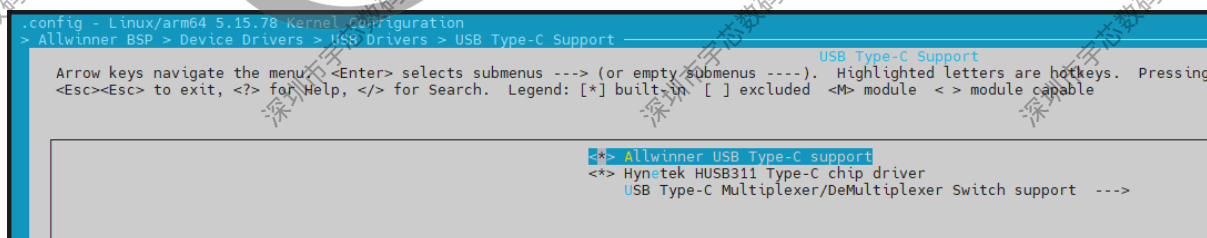


图 2-6: Allwinner USB Type-C 驱动菜单

其他功能驱动配置参考《Linux_USB2_开发指南.pdf》。

2.4.2 Device Tree 配置说明

2.4.2.1 路径定义

设备树中定义的是该类芯片对应于 IC 规格的所有配置，设备树文件路径如下：

```
32/64 位平台：bsp/configs/linux-x.x/{CHIP}.dtsi
```

2.4.2.2 配置示例

```
1  usbc2:usbc2@12 {
2     device_type = "usbc2";
3     compatible = "allwinner,sunxi-plat-dwc3";
4     reg = <0x0 0x12 0x0 0x1000>;
5     #address-cells = <2>;
6     #size-cells = <2>;
7     ranges;
8     aw,inv-sync-hdr-quirk;
9     aw,hcgen2-phygen1-quirk;
10    status = "disabled";
11
12    xhci2: xhci2-controller@4d00000 {
13        compatible = "snps,dwc3";
14        reg = <0x0 0x04d00000 0x0 0x100000>;
15        interrupts = <GIC_SPI 35 IRQ_TYPE_LEVEL_HIGH>;
16        dr_mode = "otg"; // dr_mode option: host, peripheral, otg
17        clocks = <&ccu CLK_USB2_MF>, <&ccu CLK_USB2_U2_REF>, <&ccu CLK_USB2_SUSPEND>;
18        clock-names = "bus_clk", "ref_clk", "suspend";
19        assigned-clocks = <&ccu CLK_USB2_SUSPEND>;
20        resets = <&ccu RST_USB_2>;
21        reset-names = "hci";
22        power-domains = <&pd SUN60IW2_PCK_USB2>;
23        maximum-speed = "super-speed-plus";
24        phy_type = "utmi";
25        snps,dis_enblslpm_quirk;
26        snps,dis-u1-entry-quirk;
27        snps,dis-u2-entry-quirk;
28        snps,dis_u3_susphy_quirk;
29        snps,dis_u2_susphy_quirk;
30        phys = <&u2phy>, <&combophy PHY_TYPE_USB3>;
31        phy-names = "usb2-phy", "usb3-phy";
32        status = "disabled";
33    };
34 };
35
36 u2phy: phy@4e00000 {
37     compatible = "allwinner,sunxi-plat-phy";
38     reg = <0x0 0x04e00000 0x0 0x800>; /* Application Registers */
39     #phy-cells = <0>;
40     status = "disabled";
41 };
42
43 combophy: phy@4f00000 {
```

```
44 compatible = "allwinner, inno-combphy";
45 reg = <0x0 0x04f00000 0x0 0x80000>; /* Sub-System Application Registers
46      <0x0 0x04f80000 0x0 0x80000>; /* Combo INNO PHY Registers */
47 reg-names = "phy-ctl", "phy-clk";
48 power-domains = <&pd1 A523_PCK_PCIE>;
49 phy_refclk_sel = <0>; /* 0: internal clk; 1: external clk */
50 resets = <&ccu RST_BUS_PCIE_USB3>;
51 #phy-cells = <1>;
52 status = "disabled";
53 };
```

2.4.2.3 注意事项

说明

- **clocks**: 目前内核支持 DWC3 控制器配置 `bus_early`, `ref`, `suspend` 三个时钟。
- **power-domains**: 用于配置 USB3 是否支持 Power Domain, 若未配置但系统支持 PPU 模块, 会导致 USB3 的 Power 电源域掉电;
- **aw,hcgen2-phygen1-quirk**: 可选配置, 若配置此选项, 表示 HC 主控支持 Gen2, 但 PHY 仅支持 Gen1, 需要强制 HC 到 Gen1, 否则对于 Gen2 的外设接入会导致主控进入 SSP 的链路训练, 在 SCD1 和 SCD2 阶段无法退出并把 PHY 识别为 10Gbps;
- **aw,inv-sync-hdr-quirk**: 可选配置, 若配置此选项, 表示使用第三方 PHY, 需要在 Gen2 链路训练时进行极性翻转。
- **dr_mode**: 表示将控制器配置的模式, `dr_mode` option: `host`, `peripheral`, `otg`;
- **maximum-speed**: 用于配置 USB3 作为 Device 时支持的最大速度;
- **phy_type**: 用于配置 USB3 控制器内部 U2 PHY 的类型;
- **snps,dis_enblslpm_quirk**: 详见 `Documentation/devicetree/bindings/usb/snps,dwc3.yaml`, 其他 `quirk` 类似;
- **phy_refclk_sel**: 0 表示使用内部参考时钟, 1 表示使用外部参考时钟, FPGA 阶段一般使用外部参考时钟。
- 某些平台支持类似 `combophy` 的其他 PHY, 如 `serdes`、`synopsys`, 此处不做过多描述, 详见对应平台的 `dtsti`。

2.4.3 board.dts 配置说明

2.4.3.1 路径定义

`board.dts` 用于保存每一个板级平台的设备信息 (如 `demo` 板, `perf1` 板等), 里面的配置信息会覆盖上面的 Device Tree 中 `dtsti` 默认配置信息。不同 IC、版型及内核版本对应的 `board.dts` 具体路径如下。

```
device/config/chips/IC/configs/BOARD//board.dts
```

2.4.3.2 配置示例

2.4.3.2.1 示例 1: HOST 模式

常用于作为 HOST 模式参考如下配置:

```

1 &usb2 {
2     device_type = "usb2";
3     drvbus-supply = <&reg_drivevbus>;
4     aw,vbus-shared-quirk; /* 若配置此项，则drvbus-supply不生效，说明与其他端口共用vbus。 */
5     status = "okay";
6 };
7
8 &xhci2 {
9     dr_mode = "host"; /* 控制器支持的主从模式，可选择的配置为： host, peripheral, otg。 */
10    phys = <&u2phy>, <&combo1_usb>; /* 某些平台支持USB3.0使用COMB1需覆写phys节点。 */
11    status = "okay";
12 };
13
14 &u2phy {
15     status = "okay";
16 };
17
18 &combophy {
19     phy_use_sel = <1>; /* 0:PCIE; 1:USB3; 2:PCIE_USB3_U2 */ /* combophy的使用者配置，0:PCIE; 1:USB3; 2:
20     PCIE_USB3_U2。 */
21     status = "okay";
22 };
23
24 &serdes {
25     status = "okay";
26 };
27
28 &synopsys {
29     status = "okay";
30 };

```

📖 说明

若仅使用 Host 功能接口，则需要配置相关选项以支持具体的功能：

- **dr_mode**：表示将控制器配置的模式，需要固定配置为 host。
- **phy_use_sel**：PCIE_USB3_U2（即配置为 2）为新增模式，用于 PCIE 和 USB3.0 同时工作的场景，其中，PCIE 使用 **combophy**，XHCI 控制器使用 U2 PHY，此时 USB3.0 只支持高速传输，使用此模式时需要确认 **combophy** 驱动中已支持该项功能。
- **serdes/synopsys**：仅适用于某些支持 serdes 或 synopsys 子系统的平台的 ComboPHY，主要用于替换 **combophy** 节点，三选一即可。

2.4.3.2.2 示例 2：Combo 模式

对于支持非标准 Type-C 接口（usb2.0drd+usb3.0host）的平板参考以下配置：

```

1 &usb2 {
2     device_type = "usb2";
3     usb_detect_type = <0x0>;
4     usb_detect_mode = <0x0>;
5     det_mode_supply = <&usb_power_supply>;
6     usb_gma340_sel_gpio = <&pio PH 7 GPIO_ACTIVE_HIGH>;
7     enable-active-high;
8     usb_gma340_oe_gpio = <&pio PH 8 GPIO_ACTIVE_HIGH>;
9     enable-active-high;
10    extcon = <&usb0>;
11    drvbus-supply = <&reg_drivevbus>;

```

```

12 aw,vbus-shared-quirk;
13 aw,u2drd-u3host-quirk;
14 status = "okay";
15 };
16
17 &xhci2 {
18     dr_mode = "host";
19     status = "okay";
20 };
21
22 &u2phy {
23     status = "okay";
24 };
25
26 &combophy {
27     resets = <&ccu RST_BUS_PCIE_USB3>;
28     phy_use_sel = <1>; /* 0:PCIE; 1:USB3; 2:PCIE_USB3_U2 */
29     status = "okay";
30 };
31
32 &serdes {
33     aw,phy-switcher-quirk;
34     status = "okay";
35 };
36
37 &synopsys {
38     status = "okay";
39 };

```

📖 说明

若使用 `typec` 接口，则需要配置相关选项以支持具体的功能：

- `usb_detect_type`: 0 表示 pmu 类型，1 表示 tcpm 类型，2 表示 gpio 类型，不配置则表示默认不支持；
- `usb_detect_mode`: 0 表示采用 Notify 方式，1 表示采用线程扫描方式，2 表示采用 EXTCON 方式，不配置则表示默认不支持；
- `det_mode_supply`: 若选择 pmu 类型，则通过此节点获取 pmu 的 TypeC 状态，包括正反插以及 device 还是 host；
- `usb_gma340_sel_gpio`: GMA340 外挂芯片的开关选择，0 表示 SW1，1 表示 SW2，驱动里会根据获取到的 Type-C 状态进行正反插切换；
- `usb_gma340_oe_gpio`: GMA340 外挂芯片的输出使能，0 表示开关打开，1 表示开关关断（默认为打开，若未支持 USB3 则需要将其配置为关断状态）；
- `extcon`: 通过内核原生 EXTCON 框架设置 USB3.0 的模式
- `drvbus-supply`: 可由 vbus 节点提供，也可由 PMU 模块提供，作为 host 时给外设供电；
- `aw,vbus-shared-quirk`: 可选配置，若配置此选项，表示与 USB2 控制器驱动共用 vbus，USB3 控制器驱动不会操作 vbus；
- `aw,u2drd-u3host-quirk`: 可选配置，若配置此选项，表示 USB2 控制器支持 DRD，USB3 控制器仅支持 HOST，在这种 Combo 端口下支持 USB3 的 Host 加载卸载、ComboPHY 的 Power On/Off；
- `dr_mode`: 表示将控制器配置的模式，`dr_mode option`: host, peripheral, otg。
- `role-switch-default-mode`: 表示在 otg 模式下配置控制器默认切换的模式，`option`: host, peripheral。
- `resets`: 在支持 PCIE_USB3_U2 模式时已把此配置移到 dtsti 的 `combophy` 节点，可以不配置。
- 根据平台选择配置 `combophy` 还是 `serdes` 以及 `synopsys` 节点，具体可咨询 FAE。
- `aw,phy-switcher-quirk`: 对于支持 `serdes` 的平台，需要配置此属性，否则在插拔时使用 `phy_switcher` 传入 `PHY_MODE_USB_OTG` 时会出现卡住问题。

2.4.3.2.3 示例 3：OTG 模式

搭配 SWITCH 芯片以及 CC Logic 芯片参考以下配置（此处以 PS8743 和 HUSB311 为例，某些支持内置 Type-C 接口的平台需要配置 **phy_switcher** 进行支持）：

```
1 phy_switcher: phy_switcher@10 {
2   compatible = "allwinner,sunxi-phy-switcher";
3   reg = <0x0 0x10 0x0 0x0>;
4   hotplug = <&pio PH 4 GPIO_ACTIVE_HIGH>; /* DP热插拔的pin脚配置。*/
5   aux_p = <&r_pio PL 10 GPIO_ACTIVE_HIGH>; /* DP的aux_p的pin脚配置。*/
6   aux_n = <&r_pio PL 11 GPIO_ACTIVE_HIGH>; /* DP的aux_n的pin脚配置。*/
7   phys = <&combo0_dp>, <&combo0_usb>; /* 引用的PHY配置，可通过phy_set_mode_ext传入正反插及模式参数。*/
8   phy-names = "dp-phy", "usb-phy";
9
10  status = "okay";
11
12  mode-switch; /* 用于Type-C内核框架获取模式的切换。*/
13  orientation-switch; /* 用于Type-C内核框架获取正反插方向的切换。*/
14  svid = <0xff01>; /* 用于识别配件模式altmode为DP。*/
15  port {
16    #address-cells = <1>;
17    #size-cells = <0>;
18    usbdp_orientation_switch: endpoint@0 {
19      reg = <0>;
20      remote-endpoint = <&usb2_orien_sw>; /* 使用port描述，phy_switcher与远程hub311设备关联，用于切换方
    向。*/
21    };
22    usbdp_dp_altmode_mux: endpoint@1 {
23      reg = <1>;
24      remote-endpoint = <&dp_altmode_mux>; /* 使用port描述，phy_switcher与远程hub311设备关联，用于切换
    MUX。*/
25    };
26  };
27 };
28
29 &twi5 {
30   ps8743: ps8743@11 {
31     compatible = "parade,ps8743";
32     reg = <0x11>;
33     hotplug = <&pio PH 19 GPIO_ACTIVE_HIGH>; /* DP支持热插拔的io配置，仅使用USB3.0可以不配置。*/
34     aw_rx_eq_emphasis_quirk; /* USB3.0 RX端EQ参数是否需要调整。*/
35     aw_rx_eq_emphasis = <0x10>; /* USB3.0 RX端EQ参数调整值，默认为2.4dB，结合Spec配置为0x10，表示需要调整
    为5dB。*/
36     status = "okay";
37
38     orientation-switch; /* 用于Type-C内核框架获取正反插方向的切换。*/
39     svid = <0xff01>; /* 用于识别配件模式altmode为DP。*/
40     port {
41       #address-cells = <1>;
42       #size-cells = <0>;
43       usbdp_orientation_switch: endpoint@0 {
44         reg = <0>;
45         remote-endpoint = <&usb2_orien_sw>; /* 使用port描述，ps8743与远程hub311设备关联，用于切换方
        向。*/
46       };
47       usbdp_dp_altmode_mux: endpoint@1 {
48         reg = <1>;
```

```
49     remote-endpoint = <&dp_altmode_mux>; /* 使用port描述, ps8743与远程hub311设备关联, 用于切换
      MUX。 */
50     };
51 };
52 };
53
54     hub311: hub311@4e {
55         compatible = "hynetek,hub311"; /* 设备匹配名, 对应Type-C PD控制器的型号。 */
56         reg = <0x4e>;
57         interrupt-parent = <&r_pio>;
58         hub311,intr_gpio = <&r_pio PL 13 GPIO_ACTIVE_LOW>; /* CC Logic的中断pin, 根据中断触发TCPM状态机。 */
59         /* interrupts = <PM 5 GPIO_ACTIVE_LOW>;可按照上述配置, 增加“wakeup-source;”支持HUSB311作为唤醒源 */
60         /* interrupt-names = "wakeup"; */
61         vbus-supply = <&reg_qc_drivevbus>; /* VBUS输出的开关, 根据CC状态通过regulator框架设置Vbus的开启和关闭。
      */
62         det_usb_supply = <&charger_power_supply>; /* 通过TCPC的ops在PD握手过程中进行电压和电流的设置, 实现快
      充等功能。 */
63         /* aw,vbus-debounce-quirk; /* 用于规避UGREEN的CD275适配器未接入电源时连接到HUSB311导致反复对外放电异
      常 */
64         /* aw,vbus-tryon-debounce = <1200>;Vbus开启与上一次关闭的时间差。 */
65         /* aw,vbus-check-debounce = <600>;Vbus关闭与上一次开启的时间差。 */
66         /* aw,port-reset-quirk;用于配置HUSB311唤醒SoC后是否进行复位。 */
67         wakeup-source;
68         status = "okay";
69
70         ports {
71             #address-cells = <1>;
72             #size-cells = <0>;
73
74             port@0 {
75                 reg = <0>;
76                 usb2_role_sw: endpoint@0 {
77                     remote-endpoint = <&dw3_0_role_switch>; /* 使用port描述, hub311与远程dw3控制器关联, 用于
      实现OTG角色切换。 */
78                 };
79             };
80         };
81
82         usb_con: connector {
83             compatible = "usb-c-connector"; /* 设备匹配名, USB连接器的型号, 一般有A、B、C。 */
84             label = "USB-C"; /* 标签, 标注当前USB口为C口。 */
85             data-role = "dual"; /* 数据传输方向: dual: 双向, host/device: 主机/从机。 */
86             power-role = "dual"; /* 充电/供电方向: dual: 双向, sink/source: 充电/放电。 */
87             try-power-role = "source"; /* 优先匹配充电/供电方向, sink/source: 充电/放电; 对于DP设备误识别, 可配置
      为“source”。 */
88             op-sink-microwatt = <1000000>; /* 最小充电功率。 */
89             // typec-power-opmode = "default";
90             // pd-disable;
91             slow-charger-loop; /* 匹配档位时, 是否从最低默认档位开始。 */
92             sink-pdos = < PDO_FIXED(5000, 500, (0x1<<29)|(0x1<<25))
93                 PDO_VAR(5000, 5000, 2000)
94                 PDO_VAR(9000, 9000, 2000)
95                 /* PDO_VAR(12000, 12000, 3000)
96                 PDO_VAR(15000, 15000, 3000)
97                 PDO_VAR(20000, 20000, 2250) */
98                 >; /* 配置充电档位, 例子中配置的档位如下: 默认档: 5V 0.5A, 档1: 5V 2A, 档2: 9V 2A。 */
99             source-pdos = < PDO_FIXED(5000, 500, (0x1<<29)|(0x1<<25))
100                 PDO_VAR(5000, 5000, 1200) >; /* 配置放电档位, 例子中配置的档位如下: 默认档: 5V 0.5A, 档
101                 1: 5V 1.2A。 */
102             /* fixme */
```

```
102 sink-vdos = <0xd1002e99 0x00 0x03110000>; /* 定义VDO消息, 使TCPM框架可以获取caps。 */
103 sink-vdos-v1 = <0xd1002e99 0x00 0x03110000>; /* 定义VDO v1消息。 */
104
105 altmodes {
106     #address-cells = <1>;
107     #size-cells = <0>;
108
109     altmode@0 {
110         reg = <0>;
111         svid = <0xff01>; /* 用于识别配件模式altmode为DP。 */
112         vdo = <0xffffffff >; /* 定义VDO消息, 用于displayport.c加载驱动。 */
113     };
114 };
115
116 ports {
117     #address-cells = <1>;
118     #size-cells = <0>;
119
120     port@0 {
121         reg = <0>;
122         usbc2_orien_sw: endpoint {
123             remote-endpoint = <&usbdp_orientation_switch>; /* 使用port描述, hub311与远程ps8743关
124 联, 用于切换方向。 */
125         };
126     };
127     port@1 {
128         reg = <1>;
129         dp_altmode_mux: endpoint {
130             remote-endpoint = <&usbdp_dp_altmode_mux>; /* 使用port描述, hub311与远程ps8743设备关
131 联, 用于切换MUX。 */
132         };
133     };
134 };
135 };
136
137 &xhci2 {
138     dr_mode = "otg";
139     usb-role-switch; /* 用于DWC3框架识别是否通过Type-C框架进行DRD切换。 */
140     usb-psy-name = "axp519-charger"; /* 用于支持DWC3通过Gadget框架设置限流 */
141     /* role-switch-default-mode = "host"; otg模式下可配置默认模式: option "host, peripheral" */
142     port {
143         #address-cells = <1>;
144         #size-cells = <0>;
145         dwc3_0_role_switch: endpoint@0 {
146             reg = <0>;
147             remote-endpoint = <&usbc2_role_sw>; /* 使用port描述, dwc3控制器与远程hub311关联, 用于实现OTG角色
148 切换。 */
149         };
150     };
151 };
```

2.4.3.3 注意事项

📖 说明

其他节点 `usb2`、`xhci2`、`u2phy`、`combophy` (或者 `serdes`、`synopsys`) 参考 HOST 模式章节进行配置即可。

2.4.3.4 扩展说明

很多设备的数据接口是通过子节点” port “来描述，一个 port 的配置取决于参与到数据传输的其他设备通过 ‘endpoint 字节节点’ 描述。

```
1 device {
2     ...
3     ports {
4         #address-cells = <1>;
5         #size-cells = <0>;
6
7         port@0 {
8             ...
9             endpoint@0 { ... };
10            endpoint@1 { ... };
11        };
12        port@1 { ... };
13    };
14};
```

- 如果一个 port 配置到和同一总线的多个远程设备工作，则必须为它们每一个提供一个” endpoint ”子节点，如果在一个设备节点上存在多个 port 或者在一个 port 上有多个 endpoint 或者 port 节点需要与一个选定的硬件接口联系，通常的方案是使用 #address-cells #size-cells 和 reg 属性。
- 所有的 port 可以分组到 ports 节点下，允许 #address-cells、#size-cells 属性标识 port、endpoint 节点和一个设备可能含有的所有子节点。
- 两个 endpoint 节点通过” remote-endpoint ”句柄彼此联系。设备的 endpoint 子节点包含对于这个设备和其他设备交换数据所有需要的属性。
- 允许一个 port 下的多个 endpoint 处于激活状态。

2.5 源码结构说明

```
1 drivers/usb/dwc3/
2 |-- Kconfig
3 |-- Makefile
4 |-- dwc3-debug.c /* 用于注册dwc3相关的调节点，支持CONFIG_DEBUG_FS有效。 */
5 |-- dwc3-sunxi-plat.c /* dwc3驱动平台相关源文件。 */
6 |-- dwc3-sunxi.h /* dwc3驱动平台相关头文件。 */
7 |-- phy-sunxi-plat.c /* USB2.0 PHY驱动，提供给core层接口。 */
```

```
8  \-- xhci-debug.c          /* 用于注册xhci相关的调试节点，支持CONFIG_DEBUG_FS有效。 */
9
10 drivers/phy/
11 |--- Kconfig
12 |--- Makefile
13 |-- sunxi-cadence-combophy.c /* Cadence Combo PHY驱动，提供给core层接口。 */
14 |-- sunxi-inno-combophy.c   /* INNO Combo PHY驱动，提供给core层接口。 */
15 |-- sunxi-synopsys-combophy.c /* Synopsys Combo PHY驱动，提供给core层接口。 */
16
17 drivers/usb/phy/
18 |-- Kconfig
19 |-- Makefile
20 |-- sunxi-awphy-plat.c      /* Allwinner 自研 USB2.0 PHY驱动，提供给core层接口。 */
21
22 drivers/usb/typec/
23 |-- Kconfig
24 |-- Makefile
25 |-- mux
26 | |--- Kconfig
27 | |--- Makefile
28 | |--- fsa4480.c           /* USB2.0和Audio Switch驱动，用于USB2.0和模拟音频进行切换。 */
29 | |--- gpio-gma340-mux.c   /* Type-C接口USB3.0 Switch驱动，用于USB3.0进行正反插切换。 */
30 | |--- ps8743.c           /* Type-C接口USB3.0和DP Switch驱动，用于进行模式配置和正反插切换。 */
31 | |-- sunxi-phy-switcher.c /* Type-C接口USB3.0和DP Switch驱动，用于TCPM框架通知模式配置和正反插切换。 */
32 | |*/
33 |-- tcpm
34 | |--- Kconfig
35 | |--- Makefile
36 | |-- tcpci_husb311.c     /* CC Logic和PD PHY驱动，初始化CC芯片，用于向tcpci框架注册回调实现Vendor操作。 */
```

3 Gadget 配置与使用

Gadget 是指具有 USB 设备控制器的 USB 设备，根据具体的功能配置，连接到 PC 后可以作为 mass storage、uac 等设备。Linux 有原生 gadget 框架，通用的配置流程可参考下文。

3.1 Gadget 内核配置

需在“USB Gadget functions configurable through configs”下选择需要的功能。

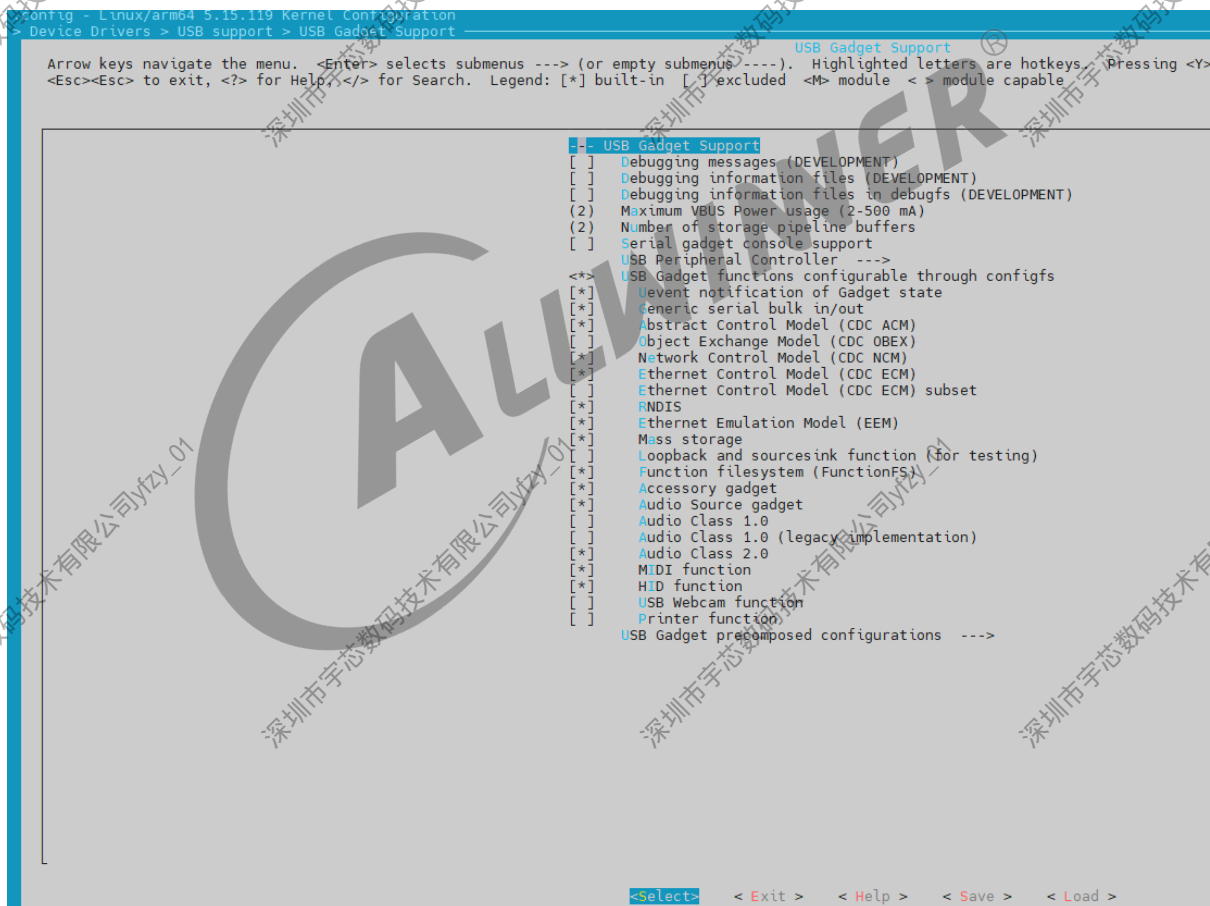


图 3-1: 内核 menuconfig Gadget Support 菜单

3.2 Gadget 配置流程

Linux 使用 configs 框架实现 composite gadget 功能。具体流程如下：

- 挂载 configs:

```
mount -t configs none /sys/kernel/config
```

挂载完成之后在 /sys/kernel/config 目录下就会生成 usb_gadget/ 目录。

- 建立 gadgets:

```
mkdir /sys/kernel/config/usb_gadget/g1
```

创建 g1/ 目录之后，该目录下会生成很多配置目录，这里的 g1 表示 gadget 1，一个 UDC 对应一个 gadget，如果当前的 SOC 上有多个 gadget，可以创建多个 gx 目录。

目前 AW 支持 USB3.0 的平台（除 H6 外）都会同时支持 USB2.0 DRD 和 USB3.0 DRD 控制器。因此，可以使用 g1 和 g2 配置两套 Device 控制器支持各自对应的 Gadget 功能驱动。

- 写入 gadget 的 PID、VID、序列号等信息:

```
echo "VID" > /sys/kernel/config/usb_gadget/g1/idVendor
echo "PID" > /sys/kernel/config/usb_gadget/g1/idProduct
mkdir /sys/kernel/config/usb_gadget/g1/strings/0x409
echo "manufacturer" > /sys/kernel/config/usb_gadget/g1/strings/0x409/manufacturer
echo "product" > /sys/kernel/config/usb_gadget/g1/strings/0x409/product
```

- 建立 gadget 相关配置 configurations

```
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1
echo 0xc0 > /sys/kernel/config/usb_gadget/g1/configs/c.1/bmAttributes
echo 500 > /sys/kernel/config/usb_gadget/g1/configs/c.1/MaxPower
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1/strings/0x409
```

- 建立功能 functions

```
mkdir /sys/kernel/config/usb_gadget/g1/functions/<name>.<instance name>
```

: function name : 任意字符串

- 建立功能和配置的链接

```
ln -s /sys/kernel/config/usb_gadget/g1/functions/<name>.<instance name> /sys/kernel/config/usb_gadget/g1/configs/c.1
```

- 使能 gadget

```
echo <udc name> > UDC
```

常见 Gadget 功能的配置方式见附录。

3.3 Gadget 功能切换

如果系统中已经配置了某个 gadget 功能，此时要配置另一个功能，需要先将上一功能清理干净。

AW 系统中默认启动执行 adb 配置脚本，因此这里以清理 adb 功能为例进行说明。

```
//停止守护进程
./etc/adb_conf.sh stop

//清理相关功能链接
rm -fr /sys/kernel/config/usb_gadget/g1/configs/c.1/ffs.adb

//取消挂载
umount /sys/kernel/config
```

执行以上操作，正常关闭 adb 后，根据需要的 gadget 功能，参考【附录】章节进行配置即可。

4 日志分析

本章节主要提供 kernel 阶段初始化时相关的日志，以供分析和参考。不同的平台和不同版本的 kernel 之间可能会有轻微差异，但是大致流程是不变的。

以下按照初始化的时间顺序来展示 usb3 相关的 log。

4.1 驱动加载日志分析

- Combo Phy 驱动加载后，初始化 PCIE 和 USB3 共用的 Sub System，常见打印示例如下。

```
[ 2.230944] inno-combphy 4f00000.phy: Sub System Version: 0x10000
```

```
[ 2.015468] sunxi-cadence-combophy 6c00000.serdes: Sub System Version v2.0.0
```

```
[ 0.500252] synopsys-combophy dc80000.synopsys: sunxi:sunxi_synopsys_combophySynopsys Combophy Version v2.1.0.0
```

- USB2.0 Phy 驱动加载后，用于初始化 phy，打印如下。

```
[ 2.727800] sunxi-plat-phy 4e00000.phy: phy provider register success
```

```
[ 0.519879] sunxi-plat-awphy c000000.phy: Allwinner USB2.0 PHY Version v1.0.0
```

- DWC3 平台驱动注册，当内核 core 层驱动还未注册并初始化控制器时将返回 EPROBE_DEFER，之后将通过框架重新 probe；

hub311 在通过 tcpci_register_port 注册 port 时，core 层还未注册 usb_role_switch，将返回错误码。

当依赖的资源加载后，内核框架会重新尝试 probe 对应的驱动。

```
[ 2.756814] sunxi-plat-dwc3 12.usbc2: failed to get drvdata dwc3
.....
[ 3.610576] i2c 5-004e: Fixing up cyclic dependency with 4d00000.xhci2-controller
[ 3.619109] I2C functionality : OK...
[ 3.624304] hub311 5-004e: Vbus Debounce: enabled, Time: 1200-600 (ms)
[ 3.632184] hub311 5-004e: fail to register tcpci port, -517
[ 3.638828] sunxi-twi 2503400.twi5: sunxi:twi_sunxi:v2.6.4 probe success
[ 3.674916] get usb_detect_type is fail, 22
[ 3.679620] get usb_detect_mode is fail, 22
[ 3.684308] get gma340-oe is fail, 22
[ 3.688414] get gma340-sel is fail, 22
```

```
[ 3.692777] sunxi-plat-dwc3 12.usbc2: DesignWare USB3 Allwinner Glue Layer Driver(v1.0.13 2023-07-10 12:00)
.....
[ 4.883910] ps8743 5-0011: setting rx_eq_emphasis: 0x10
[ 4.889807] ps8743 5-0011: Revision ID:0x0b00, Chip ID:0x8741, probe success
[ 4.897789] I2C functionality : OK...
[ 4.902673] hub311 5-004e: Vbus Debounce: enabled, Time: 1200-600 (ms)
[ 4.912949] hub311 5-004e: vbus is already Off
[ 4.919549] ps8743 5-0011: orientation:0, set sw conf 0x00 -> 0xa8
[ 4.927581] hub311 5-004e: battery exist: no
[ 4.927693] hub311 5-004e: vbus is already Off
[ 4.932765] hub311 5-004e: Vendor ID:0x2e99, Product ID:0x0311, probe success
[ 4.937992] hub311 5-004e: Setting voltage/current limit 0 mV 0 mA
[ 4.946100] psci_checker: PSCI checker started using 8 CPUs
[ 4.953274] ps8743 5-0011: orientation:0, set sw conf 0xb8 -> 0xa8
.....
```

注意：hub311 和 ps8743 仅对于支持 OTG 模式的 USB3.0 的 DRD 需要加载。

4.2 Device 日志分析

- 当通过双 Type-A 端口、双 Type-C 端口、Type-A 转 Type-C 的线缆连接至 PC 时，会根据实际情况识别为 SuperSpeed 设备或者 HighSpeed 设备。

```
[ 5.271990] ps8743 5-0011: orientation:1, set sw conf 0xb8 -> 0xb8
[ 5.279443] hub311 5-004e: get current limit 2400 mA
[ 5.285150] hub311 5-004e: Setting voltage/current limit 5000 mV 500 mA
[ 5.292933] hub311 5-004e: vbus is already Off
[ 5.616387] ps8743 5-0011: orientation:1, set sw conf 0xb8 -> 0xb8
[ 6.150485] android_work: sent uevent USB_STATE=CONNECTED
[ 6.183556] android_work: sent uevent USB_STATE=CONFIGURED
[ 6.274151] hub311 5-004e: get current limit 2400 mA
[ 6.279886] hub311 5-004e: Setting voltage/current limit 5000 mV 500 mA
[ 6.601105] ps8743 5-0011: orientation:1, set sw conf 0xb8 -> 0xb8
[ 7.258892] hub311 5-004e: get current limit 2400 mA
[ 7.264620] hub311 5-004e: Setting voltage/current limit 5000 mV 500 mA
[ 7.583205] hub311 5-004e: get current limit 2400 mA
[ 7.588924] hub311 5-004e: Setting voltage/current limit 5000 mV 2400 mA
```

在小机端可以使用以下指令查看：

```
cat /sys/devices/platform/soc@3000000/12.usbc2/4d00000.xhci2-controller/udc/4d00000.xhci2-controller/
current_speed
```

在 PC 端推荐使用 USB Device Tree Viewer 工具查看，此工具可以很方便的查看设备描述符。

```

===== USB Port18 =====
Connection Status      : 0x01 (Device is connected)
Port Chain             : 1-18
Properties              : 0x03
IsUserConnectable     : yes
PortIsDebugCapable    : yes
PortHasMultiCompanions : no
PortConnectorIsTypeC  : no
ConnectionIndex       : 0x12 (Port 18)
CompanionIndex        : 0
CompanionHubSymLnk    : USB#ROOT_HUB30#42c02a0e80#f18a0e88-c30c-11d0-8815-00a0c906bed8
CompanionPortNumber   : 0x01 (Port 1)
-> CompanionPortChain : 1-1

===== Summary =====
Vendor ID              : 0x18D1 (Google Inc.)
Product ID            : 0x0002
USB Version            : 3.2 Gen 1
Port maximum Speed    : SuperSpeed
Device maximum Speed  : SuperSpeed
Device Connection Speed : SuperSpeed
Self powered          : yes
Demanded Current      : 496 mA
Used Endpoints        : 3

===== USB Device =====
+++++ Device Information +++++
Friendly Name          : ConFig's ffs gadget
Device Description     : ADB #*
Device Path 1         : \\?\USB#VID_18D1&PID_0002#0147572d10818852251#{f72fe0d4-cbcb-407d-8814-9ed673d0dd6b} (ANDROID_USB_CLASS_ID)
Device Path 2         : \\?\USB#VID_18D1&PID_0002#0147572d10818852251#{dee824ef-729b-4a0e-9c14-b7117d33a817} (GUID_DEVINTERFACE_WIUSB)
Device Path 3         : \\?\USB#VID_18D1&PID_0002#0147572d10818852251#{a5dcbf10-6530-11d2-901f-00c04fb951ed} (GUID_DEVINTERFACE_USB_DEVICE)
Kernel Name           : Device\USBPDO-11
Device ID             : USB\VID_18D1&PID_0002#0147572d10818852251
Hardware IDs          : USB\VID_18D1&PID_0002&REV_0515 USB\VID_18D1&PID_0002
Driver KeyName        : {88bae032-5a81-49f0-bc3d-a4ff138216d6}\0462 (GUID_DEVCLASS_USBDEVICE)
Driver                : \SystemRoot\System32\drivers\winusb.sys (Version: 10.0.19041.1 Date: 2019-12-07)
Driver Inf            : C:\Windows\inf\winusb.inf
Legacy BusType        : PNPBUS
Class                 : USBDevice
Class GUID             : {88bae032-5a81-49f0-bc3d-a4ff138216d6} (GUID_DEVCLASS_USBDEVICE)
Service               : WInUSB
Enumerator            : USB
Location Info         : Port_#0018, Hub_#0001
Location IDs          : PCROOT(0)#PCI(1400)#USBROOT(0)#USB(18), *ACPI(USB_0)#ACPI(PC00)#ACPI(XHCI)#ACPI(RHUB)#ACPI(SS02)
Container ID          : {323e0f16-b57f-5c69-8a1a-40dff91f3574}
Manufacturer Info    : WinUSB #*
Capabilities          : 0x14 (Removable, UniqueID)
Status                : 0x0180600A (DN_DRIVER_LOADED, DN_STARTED, DN_DISABLEABLE, DN_REMOVABLE, DN_NT_ENUMERATOR, DN_NT_DRIVER)
Problem Code          : 0
HcdDisableSelectiveSuspend : 0
EnableSelectiveSuspend : 0
SelectiveSuspendEnabled : 0
EnhancedPowerMgmtEnabled : 0
IdleInWorkingState   : 0
WakeFromSleepState   : 0
Power State          : D0 (supported: D0, D3_wake from D0)

```

图 4-1: USB Device Tree Viewer

4.3 Host 日志分析

- 当使用 OTG 线（也可以是 Type-C 转 Type-C 的线缆连接外设）接入各种外设后，将会通过 Type-C 的 TCPM 框架加载 XHCI 主机控制器驱动。

连接不同的设备将会根据设备特性（SuperSpeed Plus、SuperSpeed、HighSpeed、Full-Speed、LowSpeed）匹配总线并进行枚举，从以下 Log 中不难看出，xhci 控制器在初始化过程中同时注册总线 3 和总线 4，其中总线 3 是高速总线，用于向下兼容 USB2.0 外设，总线 4 是超高速总线，用于识别 USB3.0 外设。

```

[ 40.675111] hub311 5-004e: set vbus On
[ 40.679482] dwc3 4d00000.xhci2-controller: request (____ptrval____) was not queued to ep0out
[ 40.689028] android_work: sent uevent USB_STATE=DISCONNECTED
[ 40.696361] xhci-hcd xhci-hcd.24.auto: xHCI Host Controller
[ 40.702673] xhci-hcd xhci-hcd.24.auto: new USB bus registered, assigned bus number 3
[ 40.711542] xhci-hcd xhci-hcd.24.auto: hcc params 0x0110ffc4 hci version 0x110 quirks 0x0000000000010010
[ 40.722204] xhci-hcd xhci-hcd.24.auto: irq 511, io mem 0x04d00000
[ 40.729274] xhci-hcd xhci-hcd.24.auto: xHCI Host Controller
[ 40.735562] xhci-hcd xhci-hcd.24.auto: new USB bus registered, assigned bus number 4
[ 40.744294] xhci-hcd xhci-hcd.24.auto: Host supports USB 3.1 Enhanced SuperSpeed
[ 40.752705] usb usb3: New USB device found, idVendor=1d6b, idProduct=0002, bcdDevice=5.15
[ 40.762015] usb usb3: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[ 40.770149] usb usb3: Product: xHCI Host Controller
[ 40.775632] usb usb3: Manufacturer: Linux 5.15.119 xhci-hcd
[ 40.781896] usb usb3: SerialNumber: xhci-hcd.24.auto
[ 40.787819] hub 3-0:1.0: USB hub found
[ 40.792042] hub 3-0:1.0: 1 port detected
[ 40.796683] usb usb4: We don't know the algorithms for LPM for this host, disabling LPM.

```

```
[ 40.805820] usb usb4: New USB device found, idVendor=1d6b, idProduct=0003, bcdDevice= 5.15
[ 40.815095] usb usb4: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[ 40.823202] usb usb4: Product: xHCI Host Controller
[ 40.828665] usb usb4: Manufacturer: Linux 5.15.119 xhci-hcd
[ 40.834911] usb usb4: SerialNumber: xhci-hcd.24.auto
[ 40.840765] hub 4-0:1.0: USB hub found
[ 40.844988] hub 4-0:1.0: 1 port detected
```

注意：需要说明的是，当配置为 Host 模式，在驱动加载过程中将会直接初始化 Host 控制器；配置为 Combo 模式，将会通过 AXP2202（或 AXP515）的 CC 状态进行 Host 的动态加载和卸载。

- 当接入 3.1 的 U 盘时，会被识别为 SuperSpeedPlus 设备，并挂到总线 4 上。

```
[ 85.953376] usb 4-1: new SuperSpeed Plus Gen 2x1 USB device number 2 using xhci-hcd
[ 85.982285] usb 4-1: New USB device found, idVendor=090c, idProduct=2320, bcdDevice=1.00
[ 85.992800] usb 4-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 86.003661] usb 4-1: Product: 512GB thinkplus
[ 86.009711] usb 4-1: Manufacturer: Lenovo
[ 86.015268] usb 4-1: SerialNumber: AA0000000000000000221
[ 86.032901] scsi host1: uas
[ 86.039349] scsi 1:0:0:0: Direct-Access  Lenovo 512GB thinkplus 1000 PQ: 0 ANSI: 6
[ 86.059094] sd 1:0:0:0: [sdb] 1000215216 512-byte logical blocks: (512 GB/477 GiB)
[ 86.072200] sd 1:0:0:0: [sdb] Write Protect is off
[ 86.079192] sd 1:0:0:0: [sdb] Mode Sense: 43 00 00 00
[ 86.091058] sd 1:0:0:0: [sdb] Write cache: enabled, read cache: enabled, doesn't support DPO or FUA
[ 86.107629] sdb: sdb1
[ 86.112338] sd 1:0:0:0: [sdb] Attached SCSI disk
```

- 当接入 3.0 的 U 盘时，会被识别为 SuperSpeed 设备，并挂到总线 4 上。

```
[ 41.183450] usb 4-1: new SuperSpeed USB device number 2 using xhci-hcd
[ 41.209077] usb 4-1: New USB device found, idVendor=152d, idProduct=0578, bcdDevice= 5.08
[ 41.218285] usb 4-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 41.226292] usb 4-1: Product: USB to ATA/ATAPI Bridge
[ 41.231958] usb 4-1: Manufacturer: JMicron
[ 41.236545] usb 4-1: SerialNumber: 0123456789ABCDEF
[ 41.245801] scsi host0: uas
[ 41.249801] scsi 0:0:0:0: Direct-Access  JMicron Generic 0508 PQ: 0 ANSI: 6
[ 42.855529] sd 0:0:0:0: [sda] 250069680 512-byte logical blocks: (128 GB/119 GiB)
[ 42.863956] sd 0:0:0:0: [sda] 4096-byte physical blocks
[ 42.870146] sd 0:0:0:0: [sda] Write Protect is off
[ 42.875563] sd 0:0:0:0: [sda] Mode Sense: 53 00 00 08
[ 42.881591] sd 0:0:0:0: [sda] Disabling FUA
[ 42.886280] sd 0:0:0:0: [sda] Write cache: enabled, read cache: enabled, doesn't support DPO or FUA
[ 42.896756] sd 0:0:0:0: [sda] Optimal transfer size 33553920 bytes not a multiple of physical block size (4096 bytes)
[ 43.254334] sda: sda1
[ 43.259027] sd 0:0:0:0: [sda] Attached SCSI disk
```

- 当接入 2.0 的 U 盘时，会被识别为 HighSpeed 设备，并挂到总线 3 上。

```
[ 52.411037] usb 3-1: new high-speed USB device number 2 using xhci-hcd
[ 52.563927] usb 3-1: New USB device found, idVendor=23a9, idProduct=ef18, bcdDevice= 1.00
[ 52.573140] usb 3-1: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 52.581162] usb 3-1: Product: Teclast CoolFlash
```

```
[ 52.586243] usb 3-1: Manufacturer: Generic
[ 52.591466] usb-storage 3-1:1.0: USB Mass Storage device detected
[ 52.598908] scsi host0: usb-storage 3-1:1.0
[ 53.631814] scsi 0:0:0:0: Direct-Access Teclast CoolFlash 1.00 PQ: 0 ANSI: 4
[ 53.641578] sd 0:0:0:0: [sda] 61472960 512-byte logical blocks: (31.5 GB/29.3 GiB)
[ 53.650391] sd 0:0:0:0: [sda] Write Protect is off
[ 53.655777] sd 0:0:0:0: [sda] Mode Sense: 03 00 00 00
[ 53.661578] sd 0:0:0:0: [sda] No Caching mode page found
[ 53.667551] sd 0:0:0:0: [sda] Assuming drive cache: write through
[ 53.697096] sda: sda1
[ 53.701188] sd 0:0:0:0: [sda] Attached SCSI removable disk
```

- 当接入全速的 U 盘时，会被识别为 FullSpeed 设备，并挂到总线 3 上。

```
[ 171.162478] usb 3-1: new full-speed USB device number 5 using xhci-hcd
[ 171.314932] usb 3-1: New USB device found, idVendor=0c76, idProduct=0003, bcdDevice=1.00
[ 171.324122] usb 3-1: New USB device strings: Mfr=0, Product=0, SerialNumber=0
[ 171.338763] usb-storage 3-1:1.0: USB Mass Storage device detected
[ 171.346040] scsi host0: usb-storage 3-1:1.0
[ 172.475360] scsi host0: scsi scan: INQUIRY result too short (5), using 36
[ 172.483005] scsi 0:0:0:0: Direct-Access USB FLASH DISK 1.00 PQ: 0 ANSI: 2
[ 172.494891] sd 0:0:0:0: [sda] 512000 512-byte logical blocks: (262 MB/250 MiB)
[ 172.503734] sd 0:0:0:0: [sda] Write Protect is off
[ 172.509117] sd 0:0:0:0: [sda] Mode Sense: 0b 00 00 08
[ 172.515687] sd 0:0:0:0: [sda] No Caching mode page found
[ 172.521673] sd 0:0:0:0: [sda] Assuming drive cache: write through
[ 172.699465] sda: sda1
[ 172.705241] sd 0:0:0:0: [sda] Attached SCSI removable disk
```

- 当接入低速鼠标或者键盘时，会被识别为 LowSpeed 设备，并挂到总线 3 上。

```
[ 254.907157] usb 3-1: new low-speed USB device number 4 using xhci-hcd
[ 255.066800] usb 3-1: New USB device found, idVendor=18f8, idProduct=1286, bcdDevice=1.00
[ 255.076019] usb 3-1: New USB device strings: Mfr=0, Product=1, SerialNumber=0
[ 255.084027] usb 3-1: Product: USB GAMING MOUSE
[ 255.100965] input: USB GAMING MOUSE as /devices/platform/soc@3000000/12.usbc2/4d00000.xhci2-controller/xhci-hcd.24.auto/usb3/3-1/3-1:1.0/0003:18F8:1286.0003/input/input5
[ 255.175173] hid-generic 0003:18F8:1286.0003: input: USB HID v1.10 Mouse [USB GAMING MOUSE] on usb-xhci-hcd.24.auto-1/input0
[ 255.191380] input: USB GAMING MOUSE Keyboard as /devices/platform/soc@3000000/12.usbc2/4d00000.xhci2-controller/xhci-hcd.24.auto/usb3/3-1/3-1:1.1/0003:18F8:1286.0004/input/input6
[ 255.267415] hid-generic 0003:18F8:1286.0004: input: USB HID v1.10 Keyboard [USB GAMING MOUSE] on usb-xhci-hcd.24.auto-1/input1
```

- 当断开 OTG 线后，将会通过 Type-C 的 TCPM 框架卸载 XHCI 主机控制器驱动，并释放已分配的总线号。

```
[ 444.969601] usb 3-1: USB disconnect, device number 4
[ 444.976461] hub311 5-004e: set vbus Off
[ 444.981545] hub311 5-004e: Setting voltage/current limit 0 mV 0 mA
[ 444.989186] ps8743 5-0011: orientation:0, set sw conf 0xb8 -> 0xa8
[ 445.010038] xhci-hcd xhci-hcd.24.auto: remove, state 4
[ 445.015830] usb usb4: USB disconnect, device number 1
[ 445.022216] xhci-hcd xhci-hcd.24.auto: USB bus 4 deregistered
```

```
[ 445.037688] xhci-hcd xhci-hcd.24.auto: remove, state 1  
[ 445.043493] usb usb3: USB disconnect, device number 1  
[ 445.115314] xhci-hcd xhci-hcd.24.auto: USB bus 3 deregistered
```



5 调试方法

5.1 常用内核调试接口

5.1.1 USB Host 总线调试接口

- 查看系统支持 USB Host 设备驱动

```
/sys/bus/usb/drivers # ls
cdc_acm   ums-cypress  ums-karma    ums-usbat   usbhid
hub       ums-datafab  ums-onetouch ums_eneub6250 uvcvideo
option    ums-freecom  ums-realtek  usb
uas       ums-isd200   ums-sddr09   usb-storage
ums-alauda ums-jumpshot ums-sddr55   usbfs
```

- 查看 USB Host 端口枚举的设备信息

以 002 BUS 为例，设备枚举信息如下：

```
[ 56.150899] usb 2-1: new SuperSpeed USB device number 2 using xhci-hcd
[ 56.188236] usb 2-1: New USB device found, idVendor=0951, idProduct=1666, bcdDevice= 0.01
[ 56.197440] usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 56.205451] usb 2-1: Product: DataTraveler 3.0
[ 56.210432] usb 2-1: Manufacturer: Kingston
[ 56.215108] usb 2-1: SerialNumber: E0D55EA574D9F481482D03BA
[ 56.222926] usb-storage 2-1:1.0: USB Mass Storage device detected
[ 56.230397] scsi host0: usb-storage 2-1:1.0
[ 57.251123] scsi 0:0:0:0: Direct-Access Kingston DataTraveler 3.0 PQ: 0 ANSI: 6
[ 57.261097] sd 0:0:0:0: [sda] 60437492 512-byte logical blocks: (30.9 GB/28.8 GiB)
[ 57.269976] sd 0:0:0:0: [sda] Write Protect is off
[ 57.275367] sd 0:0:0:0: [sda] Mode Sense: 4f 00 00 00
[ 57.281303] sd 0:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
[ 57.295608] sda: sda1
[ 57.299333] sd 0:0:0:0: [sda] Attached SCSI removable disk
```

设备枚举成功后，会在 `/sys/bus/usb/devices` 路径下注册很多的节点。

通过相关节点可获取 host 口枚举的设备 VID、PID 和 `bNumInterfaces`、`bInterfaceProtocol`、`bInterfaceSubClass` 等信息。

例如，`2-1` 路径下的 `idVendor` 和 `idProduct` 节点，能够读取到对应的 VID 和 PID，这与枚举时的日志信息一致。

- USB Host 控制器动态初始化接口

```
/sys/bus/usb/driver/usb/bind,unbind
```

使用方法：**/sys/bus/usb/driver/usb/**路径下的每个“usbX”节点，分别对应一个USB HOST 主机控制器。

若要重新初始化 USB Host 控制器 “usb2”，则执行如下命令：

```
echo usb2 > /sys/bus/usb/driver/usb/unbind  
echo usb2 > /sys/bus/usb/driver/usb/bind
```

执行上述命令后，就会重新初始化对应的“usb2”控制器，并重新枚举该控制器连接的 USB 外设，示例如下。

```
/sys/bus/usb/drivers/usb # ls  
2-1 bind uevent unbind usb1 usb2 usb3 usb4  
/sys/bus/usb/drivers/usb # echo usb2 > unbind  
[ 560.775722] usb 2-1: USB disconnect, device number 2  
/sys/bus/usb/drivers/usb # echo usb2 > bind  
[ 568.762068] hub 2-0:1.0: USB hub found  
[ 568.766330] hub 2-0:1.0: 1 port detected
```

5.1.2 USB Host Debugfs 调试接口

- 挂载 debugfs 文件系统

```
mount -t debugfs none /sys/kernel/debug
```

- 查看 USB 总线上的所有 USB 设备信息

```
cat /sys/kernel/debug/usb/devices
```

- usbmon 抓包工具的调试接口

```
/sys/kernel/debug/usb/usbmon
```

- 查看 UVC 外设的调试接口

```
/sys/kernel/debug/usb/uvcvideo
```

说明

- 内核配置需开启CONFIG_DEBUG_FS宏方可使用 debugfs；
- 内核配置需开启CONFIG_USB_MON宏方可使用 usbmon 调试接口；
- 内核配置需开启CONFIG_MEDIA_USB_SUPPORT和CONFIG_USB_VIDEO_CLASS宏才能识别 UVC 外设。

接入 UVC 设备至 HOST 口 uvcvideo 目录才有相关调试接口。

前提：内核配置打开 debugfs：即 CONFIG_DEBUG_FS = y。

- 查看 XHCI 控制器的调试接口

```
cd /sys/kernel/debug/usb/4d00000.xhci2-controller
cd /sys/kernel/debug/usb/xhci
```

说明

注：后续章节将会着重介绍 xhci 相关的调节点功能。

5.1.3 USB Trace 类调试接口

- Trace USB Gadget Driver 与 USB Device Controller Driver 交互数据的调试接口：

```
/sys/kernel/debug/tracing/events/gadget
```

- Trace DWC3 控制器传输流程的调试接口：

```
/sys/kernel/debug/tracing/events/dwc3
```

- Trace xHCI 控制器传输流程的调试接口：

```
/sys/kernel/debug/tracing/events/xhci-hcd
```

说明

注意：内核配置需开启 CONFIG_FUNCTION_TRACER 宏。

5.1.4 USB Gadget 调试接口

- USB Gadget 动态断开重连的调试接口：

```
断开连接：echo disconnect > /sys/class/udc/[usb controller name]/soft_connect
使能连接：echo connect > /sys/class/udc/[usb controller name]/soft_connect
```

- USB Gadget 查询当前的枚举速率

```
cat /sys/class/udc/[usb controller name]/current_speed
```

- USB Gadget 查询当前的枚举状态 (如: not attached/configured)

```
cat /sys/class/udc/[usb controller name]/state
```

 说明

命令中 '[usb controller name]' 需要修改为芯片对应的 USB Device 控制器的名称。如 A523 则修改为 4100000.udc-controller, 示例如下。

```
# echo disconnect > /sys/class/udc/4100000.udc-controller/soft_connect
[ 314.281272] android_work: sent uevent USB_STATE=DISCONNECTED
[ 314.287831] android_work: did not send uevent (0 0 0000000000000000)
# echo connect > /sys/class/udc/4100000.udc-controller/soft_connect
[ 324.802215] android_work: sent uevent USB_STATE=CONNECTED
[ 324.846555] android_work: sent uevent USB_STATE=CONFIGURED
```

5.2 USB3.0 调试方法

 说明

以下测试指令需要挂载 debugfs 才可使用：`mount -t debugfs none /sys/kernel/debug`。

5.2.1 DWC3 调试节点

- regdump: 可用于查看寄存器。

```
cat /sys/kernel/debug/usb/4d00000.xhci2-controller/regdump
```

- lsp_dump: LSP MUX 参数调试, 查看和调整 LSP, 参数根据模式确定, 见备注。

```
cat /sys/kernel/debug/usb/4d00000.xhci2-controller/lsp_dump
echo [0x-] > /sys/kernel/debug/usb/4d00000.xhci2-controller/lsp_dump
```

e.g:

```
echo 0xc170 > /sys/class/sunxi_dump/dump; cat /sys/class/sunxi_dump/dump [详见 Spec 0xC170 寄存器介绍]
```

- mode: 模式切换, 用于查看和切换主从模式 (otg 模式下才可切换)。

```
cat /sys/kernel/debug/usb/4d00000.xhci2-controller/mode
echo [str] > /sys/kernel/debug/usb/4d00000.xhci2-controller/mode
注: [str] 可传入 host/device/otg
```

- testmode: 查看和进行眼图测试, 指定 host 或者 device 才可进行眼图测试。

```
cat /sys/kernel/debug/usb/4d00000.xhci2-controller/testmode
echo [str] > /sys/kernel/debug/usb/4d00000.xhci2-controller/testmode
注：[str] 可传入 no test/test_j/test_k/test_se0_nak/test_packet/test_force_enable
```

- link_state：查看和切换链路状态，仅 device 支持。

```
cat /sys/kernel/debug/usb/4d00000.xhci2-controller/link_state
echo [str] > /sys/kernel/debug/usb/4d00000.xhci2-controller/link_state
注：[str] 可传入 SS.Disabled/Rx.Detect/SS.Inactive/Recovery/Compliance/Loopback
```

- compliance：用于 USB3.0 眼图信号测试，查看当前正在进行的测试 Pattern 或者切换到某个测试 Pattern。

测试 Pattern 的具体含义参考 **Universal Serial Bus 3.1 Specification, Revision 1.0, 6.4.4 Compliance Pattern**。

```
cat /sys/kernel/debug/usb/4d00000.xhci2-controller/compliance
echo CP[xx] > /sys/kernel/debug/usb/4d00000.xhci2-controller/compliance
注：CP[xx] 可传入 CP00~CP12，超过则会提示 “this is unknown pattern xx!”
```

- force_host：用于 **Combo 模式** 手动切换到 host 模式并加载 XHCI 控制器驱动。

```
cat /sys/kernel/debug/usb/4d00000.xhci2-controller/force_host
echo [str] > /sys/kernel/debug/usb/4d00000.xhci2-controller/force_host
注：[str] 可传入 enabled/disabled，分别用于加载和卸载 host 驱动。
```

5.2.2 XHCI 调试节点

- portsc：查看端口状态和进入 compliance mode，仅 host 支持。

```
cat /sys/kernel/debug/usb/xhci/xhci-hcd.24.auto/ports/port02/portsc
echo compliance > /sys/kernel/debug/usb/xhci/xhci-hcd.24.auto/ports/port02/portsc
```

注：不连接外设时测试，若连接外设则会打印 ash: write error: Operation not permitted。

- port：查看端口状态以及复位 XHCI 控制器，仅 host 支持。

```
cat /sys/kernel/debug/usb/xhci/xhci-hcd.24.auto/ports/port02/port
echo reset > /sys/kernel/debug/usb/xhci/xhci-hcd.24.auto/ports/port02/port
```

注：与 portsc 节点功能类似，支持传入 reset 指令的功能，用于复位控制器。

5.3 Type-C 调试方法

5.3.1 PS8743 配置说明

外挂 Type-C 的 Switch 芯片较为简单，仅需按照如下配置即可完成正反插和配件模式 AltMode 的切换。

1	4-Lane DP(正插)	0x00h <bit7:2>	0xe8	11101000
2	4-Lane DP(反插)	0x00h <bit7:2>	0xec	11101100
3	2-Lane DP+USB3.1(正插)	0x00h <bit7:2>	0xf8	11111000
4	2-Lane DP+USB3.1(反插)	0x00h <bit7:2>	0xfc	11111100
5	USB3.1(正插)	0x00h <bit7:2>	0xb8	10111000
6	USB3.1(反插)	0x00h <bit7:2>	0xbc	10111100

注：主要是 offset 0x00 的值，是设置 PS8743 工作模式的；先把模式设置好，如果有信号兼容性问题，再根据 Spec 调整 EQ。

调试时可以通过 i2c 调试工具直接访问和读写寄存器的值进行芯片功能验证。

```
i2cget -y -f 5 0x11
i2cset -y 5 0x11 0x00 0xb8
i2cset -y 5 0x11 0x00 0xbc
i2cdump -y -f 5 0x11
```

5.3.2 GMA340 配置说明

外挂 Type-C 的 Switch 芯片较为简单，仅需按照如下配置即可完成正反插。

OE 引脚：拉低使能，拉高失能。

SEL 引脚：差分线切换选择。拉低时 A Pin 切换到 B Pin；拉高时 A Pin 切换到 C Pin。

注：调试时可以通过裸写 io 寄存器进行芯片功能验证。

5.3.3 CC Logic 调试

目前内核原生支持 FUSB302、RT1711 等几款 CC 芯片，HUSB311 驱动也基于此进行开发，整个 TCPM 框架的状态机较为复杂，推荐使用以下方式进行功能开发和调试：

```
mount -t debugfs none /sys/kernel/debug  
cat /sys/kernel/debug/usb/tcpm-5-004e/log
```

若尚未开启 HUSB311 软件驱动，则可以通过 i2c 调试工具直接访问寄存器的值。

```
i2cdump -y -f 5 0x4e
```

若已开启软件驱动，则可以通过 regmap 节点对寄存器进行访问，也可以直接使用 i2c-tools 进行调试。

```
cat /sys/kernel/debug/regmap/${dev-name}/registers
```



6 测试方法

6.1 USB3.0 U3 信号测试步骤

Step1: 执行以下指令挂载 debugfs。

```
mount -t debugfs none /sys/kernel/debug
```

Step2: 确认 USB3.0 控制器是否处于 HOST 模式。

对于泛平板，如果支持非标准 Type-C 接口 **Combo 模式**，可以通过以下指令切换到 HOST 模式进行测试。

```
echo enabled > /sys/kernel/debug/usb/4d00000.xhci2-controller/force_host
```

Step3: 查看当前链路状态。

```
cat /sys/kernel/debug/usb/xhci/xhci-hcd.24.auto/ports/port02/portsc
```

Step4: 复位 XHCI 控制器。

```
echo reset > /sys/kernel/debug/usb/xhci/xhci-hcd.24.auto/ports/port02/port
```

Step5: 再次查看当前链路状态。

```
cat /sys/kernel/debug/usb/xhci/xhci-hcd.24.auto/ports/port02/portsc
```

Step6: 查看当前测试 Pattern。

```
cat /sys/kernel/debug/usb/4d00000.xhci2-controller/compliance
```

Step7: 测试 CP4，Gen2 速率使用 CP9~CP12 进行测试。

```
echo CP04 > /sys/kernel/debug/usb/4d00000.xhci2-controller/compliance
```

Step8: 确认链路状态。

```
cat /sys/kernel/debug/usb/xhci/xhci-hcd.24.auto/ports/port02/portsc
```

6.2 USB3.0 U2 信号测试步骤

- HOST 眼图测试

```
echo host > /sys/kernel/debug/usb/6a00000.xhci2-controller/mode echo test_pack > /sys/  
kernel/debug/usb/xhci/xhci-hcd.41.auto/ports/port01/ed_test
```

📖 说明

如果以往默认驱动不支持，需联系 FAE 获取相关补丁，仅接上有 D+/D-的差分线：
<http://gerrit.allwinnertech.com:8081/c/lichee/bsp/+271510>

- Device 眼图测试

```
echo device > /sys/kernel/debug/usb/6a00000.xhci2-controller/mode echo test_packet > /  
sys/kernel/debug/usb/6a00000.xhci2-controller/testmode
```

📖 说明

要在设备树里将 dr_mode 配成为 device 或 otg 模式，若是 host 则无法进行测试

7.1.3 device 配置成 3+ 复合功能时异常

部分平台 USB3.0 控制器在 device 模式下仅支持 5 对端点 (含 EP0)，所以当配置的复合功能所需 ep 端点数超过支持数时，功能异常。

7.1.4 dwc3_sunxi_plat 驱动加载失败

- 问题现象：

```
sunxi:dwc3_sunxi_plat-12.usbc2:[ERR]: failed to get drvdata dwc3
```

因 power-domain 未适配完善，导致影响 dwc3 驱动加载，当出现 failed to get drvdata dwc3 提示时，请排查 power-domain 是否正常支持。

7.1.5 xhci 注册失败

- 问题现象：

```
wrn: combophy1 wait pma ready timeout
xhci-hcd xhci-hcd.41.auto: xHCI Host Controller
xhci-hcd xhci-hcd.41.auto: new USB bus registered, assigned bus number 1
xhci-hcd xhci-hcd.41.auto: can't setup: -110
xhci-hcd xhci-hcd.41.auto: USB bus 1 deregistered
xhci-hcd: probe of xhci-hcd.41.auto failed with error -110
```

因 combophy 出现 timeout，导致影响 xhci，当出现 wrn: combophy1 wait pma ready timeout 提示时，请排查 combophy 的电源是否正常。

7.1.6 MTP 从 PC 拷贝文件至样机进度条不动

- 测试环境：A523 EVB 采用标准 Type-C，支持 USB3.0 DRD（外挂 PS8743 和 HUSB311）

- 问题现象：

<1> 对于 Android 平台，连接到 PC 的 USB2.0 端口时能够互相拷贝，连接到 PC 的 USB3.0 端口时从样机拷贝文件到 PC 正常，从 PC 拷贝文件至样机进度条始终为 0。

<2> 对于 Linux 平台，连接到 PC 的 USB2.0 端口时能够正常使用 ADB 的 push 和 pull 传输文件，连接到 PC 的 USB3.0 端口时，从样机 pull 文件正常，从 PC 端 push 文件出现如下报错，且进行 Mass Storage 测试出现 ATTO 提示“写入文件出错”。

```
adb: error: 65544-byte write failed: No error
```

- 分析思路：

<1> 前期 USB3.0 Device 调试过程中已使用 Ver 板型完成 12H 的 Device ATTO 压力测试，因此倾向于 EVB 板型的部分软件配置或者外围芯片影响。

<2> 对比其他 Type-A 接口板型 (T527 Demo)，将其配置为 Device 模式 (dts 配置为 “peripheral”，menuconfig 配置为 **CONFIG_USB_DWC3_GADGET** 或者选择 **CONFIG_USB_DWC3_DUAL_ROLE**)，使用 ADB 或者 Mass Storage 测试正常，进行 60min 压力测试，提示 “未找到错误”。

- 参数调试：

结合以上分析，基本确认是由于 PS8743 对 RX 信号造成干扰引起此问题，对照其 Spec 尝试修改 0x3b~0x3f 寄存器。

```
i2cdump -y -f 5 0x11
i2cset -y -f 5 0x11 0x3b 0x10
```

参考《PS8743_Register_Table_Ver1.0_20170307_for allwinner.pdf》第 3 小节 “USB related settings”，按照上述修改 RX 相关的寄存器后进行 ADB push 即可正常。

3Bh	0x00	R/W	<7:4> USB Type C connector facing receiver equalization setting 0000:Compensate for channel loss up to 2.4dB 0001:Compensate for channel loss up to 5dB 0010:Compensate for channel loss up to 6.5dB 0011:Compensate for channel loss up to 7.4dB 0100:Compensate for channel loss up to 8.7dB 0101:Compensate for channel loss up to 10.9dB 0110:Compensate for channel loss up to 12.8dB 0111:Compensate for channel loss up to 13.8dB 1000:Compensate for channel loss up to 14.8dB 1001:Compensate for channel loss up to 15.4dB 1010:Compensate for channel loss up to 16.0dB 1011:Compensate for channel loss up to 16.7dB 1100:Compensate for channel loss up to 18.8dB 1101:Compensate for channel loss up to 21.3dB 1110:Compensate for channel loss up to 22.2dB 1111:reserved <3:0> reserved	USB EQ setting (Type C connector ->PS8743, all loss referring to 2.5GHz)
-----	------	-----	--	--

图 7-2: USB EQ setting

- 解决方案：

在 ps8743 设备树节点新增 “aw,rx_eq_emphasis_quirk” 和 “aw,rx_eq_emphasis” 属性，用于配置 USB3.0 RX 端 EQ 参数是否需要调整以及调整多少 dB，此处把 “aw,rx_eq_emphasis” 配置为 0x10。（说明：此值与 PCB 和板级上 Type-C connector 与 PS8743 的分布相关，如果按照参考设计一般不需要进行调整。）

- 测试结果：

搭建 ATTO 压力测试 30min 能够正常完成压力测试；使用 Android 固件进行 MTP 拷贝或者 ADB 传输文件均正常。

7.1.7 USB 主机功能驱动异常

现象：串口有 Host 枚举 USB 设备的 log，但是没有出现 class 驱动的打印信息

方法：kernel 没有加载相应 class 驱动，需重新配置 kernel，开启对应 class 驱动支持即可。

7.1.8 如何确定 USB 外设的枚举速率

通过打印的 USB 枚举的关键 log 进行确认。

- “usb 4-1: new SuperSpeed Plus Gen.2x1 USB device number 2 using xhci-hcd”，该 log 出现 SuperSpeed Plus 表示超速+，即 10Gbps，用 xhci 控制器的总线 4。
- “usb 4-1: new SuperSpeed USB device number 2 using xhci-hcd”，该 log 出现 SuperSpeed 表示超速，即 5Gbps，用 xhci 控制器的总线 4。
- “usb 3-1: new high-speed USB device number 2 using xhci-hcd”，该 log 出现 high-speed 表示高速，用 xhci 控制器的总线 3。
- “usb 3-1: new full-speed USB device number 5 using xhci-hcd”，该 log 出现 full-speed 表示全速，用 xhci 控制器的总线 3。
- “usb 3-1: new low-speed USB device number 4 using xhci-hcd”，该 log 出现 low-speed 表示低速，用 xhci 控制器的总线 3。

7.1.9 USB 传输速率低

影响 USB 传输速率的主要因素：

- USB 信号质量
- USB 控制器总线频率 AHB CLK 和 DMA Burst Length
- CPU 运行频率
- 存储介质读写性能的影响
- 存储设备的文件系统格式

遇到相关性能问题时，具体分析方法请参考 USB 性能分析章节。

8 附录

8.1 Gadget 配置示例

8.1.1 小机做 mass storage

依赖配置项：

```
CONFIG_USB_CONFIGFS_MASS_STORAGE=y
Device Drivers --->
[*] USB support --->
<*> USB Gadget Support --->
<*> USB Gadget functions configurable through configs
[*] Mass storage
```

configs 配置方法：

```
dd if=/dev/zero of=/dev/a.bin bs=1M count=800
mount -t configfs none /sys/kernel/config
mkdir /sys/kernel/config/usb_gadget/g1
echo "0x18d1" > /sys/kernel/config/usb_gadget/g1/idVendor
echo "0x0001" > /sys/kernel/config/usb_gadget/g1/idProduct
mkdir /sys/kernel/config/usb_gadget/g1/strings/0x409
mkdir /sys/kernel/config/usb_gadget/g1/functions/mass_storage.usb0
echo /dev/a.bin > /sys/kernel/config/usb_gadget/g1/functions/mass_storage.usb0/lun.0/file
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1
echo 0xc0 > /sys/kernel/config/usb_gadget/g1/configs/c.1/bmAttributes
echo 500 > /sys/kernel/config/usb_gadget/g1/configs/c.1/MaxPower
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1/strings/0x409
ln -s /sys/kernel/config/usb_gadget/g1/functions/mass_storage.usb0/ /sys/kernel/config/usb_gadget/g1/configs/c.1/
ls /sys/class/udc/ | xargs echo > /sys/kernel/config/usb_gadget/g1/UDC
```

说明

如果需要增加 lun，在 functions/mass_storage.usb0 下：

```
mkdir lun.1
mkdir lun.2
```

8.1.2 小机做 cdrom

依赖配置项：

```
CONFIG_USB_CONFIGFS_MASS_STORAGE=y
Device Drivers --->
```

```

[*] USB support --->
<*> USB Gadget Support --->
<*> USB Gadget functions configurable through configs
[*] Mass storage

```

configs 配置方法:

```

mount -t configs none /sys/kernel/config
mkdir /sys/kernel/config/usb_gadget/g1
echo "0x1f3a" > /sys/kernel/config/usb_gadget/g1/idVendor
echo "0xa4ac" > /sys/kernel/config/usb_gadget/g1/idProduct
mkdir /sys/kernel/config/usb_gadget/g1/strings/0x409
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1
echo 0xc0 > /sys/kernel/config/usb_gadget/g1/configs/c.1/bmAttributes
echo 500 > /sys/kernel/config/usb_gadget/g1/configs/c.1/MaxPower
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1/strings/0x409
mkdir /sys/kernel/config/usb_gadget/g1/functions/mass_storage.usb0
echo 1 > /sys/kernel/config/usb_gadget/g1/functions/mass_storage.usb0/lun.0/cdrom
echo /tmp/phoenixcard.iso > /sys/kernel/config/usb_gadget/g1/functions/mass_storage.usb0/lun.0/file
ln -s /sys/kernel/config/usb_gadget/g1/functions/mass_storage.usb0/ /sys/kernel/config/usb_gadget/g1/configs/c.1/mass_storage.usb0
ls /sys/class/udc/ | xargs echo > /sys/kernel/config/usb_gadget/g1/UDC

```

说明

/tmp/phoenixcard.iso 根据实际情况更改（任意文件修改后缀即可）。

8.1.3 小机做 UAC1

依赖配置项:

```

CONFIG_SOUND=y
CONFIG_SND=y
CONFIG_USB_CONFIGFS_F_UAC1=y

Device Drivers --->
<*> Sound card support --->
<*> Advanced Linux Sound Architecture --->
[*] USB support --->
<*> USB Gadget Support --->
<*> USB Gadget functions configurable through configs
[*] Audio Class 1.0

```

configs 配置方法:

```

mount -t configs none /sys/kernel/config
mkdir /sys/kernel/config/usb_gadget/g1
echo "0x1d61" > /sys/kernel/config/usb_gadget/g1/idVendor
echo "0x0101" > /sys/kernel/config/usb_gadget/g1/idProduct
mkdir /sys/kernel/config/usb_gadget/g1/strings/0x409
mkdir /sys/kernel/config/usb_gadget/g1/functions/uac1.usb0
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1
echo 0xc0 > /sys/kernel/config/usb_gadget/g1/configs/c.1/bmAttributes
echo 500 > /sys/kernel/config/usb_gadget/g1/configs/c.1/MaxPower
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1/strings/0x409
ln -s /sys/kernel/config/usb_gadget/g1/functions/uac1.usb0/ /sys/kernel/config/usb_gadget/g1/configs/c.1/
ls /sys/class/udc/ | xargs echo > /sys/kernel/config/usb_gadget/g1/UDC

```

 说明

uac1 仅支持高速，由于不支持 SuperSpeed 描述符，在接入 PC 时会识别为超速设备会导致获取配置描述符失败。可以使用 USB2.0 的线缆或者 USB2.0 的主机端口，也可以将 dts 里的 maximum-speed 属性修改为 “high-speed” 进行规避。

8.1.4 小机做 UAC2

依赖配置项：

```
CONFIG_SOUND=y
CONFIG_SND=y
CONFIG_USB_CONFIGFS_F_UAC2=y

Device Drivers --->
<*> Sound card support --->
  <*> Advanced Linux Sound Architecture --->
    [*] USB support --->
      <*> USB Gadget Support --->
        <*> USB Gadget functions configurable through configs
          [*] Audio Class 2.0
```

configs 配置方法：

```
mount -t configs none /sys/kernel/config
mkdir /sys/kernel/config/usb_gadget/g1
echo "0x1d61" > /sys/kernel/config/usb_gadget/g1/idVendor
echo "0x0101" > /sys/kernel/config/usb_gadget/g1/idProduct
mkdir /sys/kernel/config/usb_gadget/g1/strings/0x409
mkdir /sys/kernel/config/usb_gadget/g1/functions/uac2.usb0
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1
echo 0xc0 > /sys/kernel/config/usb_gadget/g1/configs/c.1/bmAttributes
echo 500 > /sys/kernel/config/usb_gadget/g1/configs/c.1/MaxPower
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1/strings/0x409
ln -s /sys/kernel/config/usb_gadget/g1/functions/uac2.usb0/ /sys/kernel/config/usb_gadget/g1/configs/c.1/
ls /sys/class/udc/ | xargs echo > /sys/kernel/config/usb_gadget/g1/UDC
```

8.1.5 小机做 UVC

依赖配置项：

```
CONFIG_MEDIA_CAMERA_SUPPORT=y
CONFIG_USB_CONFIGFS_F_UVC=y

Device Drivers --->
<*> Multimedia support --->
  [*] Cameras/video grabbers support
  [*] USB support --->
    <*> USB Gadget Support --->
      <*> USB Gadget functions configurable through configs
        [*] USB Webcam function
```

configs 配置方法：

```

mount -t configfs none /sys/kernel/config
mkdir /sys/kernel/config/usb_gadget/g1
echo "0x1f3a" > /sys/kernel/config/usb_gadget/g1/idVendor
echo "0x100d" > /sys/kernel/config/usb_gadget/g1/idProduct
mkdir /sys/kernel/config/usb_gadget/g1/strings/0x409
mkdir /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0
mkdir -p /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0/streaming/mjpeg/m/720p
echo 1280 > /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0/streaming/mjpeg/m/720p/wWidth
echo 720 > /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0/streaming/mjpeg/m/720p/wHeight
echo 333333 > /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0/streaming/mjpeg/m/720p/dwFrameInterval
echo 333333 > /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0/streaming/mjpeg/m/720p/
dwDefaultFrameInterval
echo 442368000 > /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0/streaming/mjpeg/m/720p/dwMinBitRate
echo 442368000 > /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0/streaming/mjpeg/m/720p/dwMaxBitRate
echo 1843200 > /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0/streaming/mjpeg/m/720p/
dwMaxVideoFrameBufferSize
mkdir /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0/streaming/header/h
ln -s /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0/streaming/mjpeg/m/ /sys/kernel/config/usb_gadget/g1/
functions/uvc.usb0/streaming/header/h/
ln -s /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0/streaming/header/h/ /sys/kernel/config/usb_gadget/g1/
functions/uvc.usb0/streaming/class/fs
ln -s /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0/streaming/header/h/ /sys/kernel/config/usb_gadget/g1/
functions/uvc.usb0/streaming/class/hs
ln -s /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0/streaming/header/h/ /sys/kernel/config/usb_gadget/g1/
functions/uvc.usb0/streaming/class/ss
mkdir /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0/control/header/h
ln -s /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0/control/header/h/ /sys/kernel/config/usb_gadget/g1/
functions/uvc.usb0/control/class/fs/
ln -s /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0/control/header/h/ /sys/kernel/config/usb_gadget/g1/
functions/uvc.usb0/control/class/hs/
ln -s /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0/control/header/h/ /sys/kernel/config/usb_gadget/g1/
functions/uvc.usb0/control/class/ss/
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1
echo 0xc0 > /sys/kernel/config/usb_gadget/g1/configs/c.1/bmAttributes
echo 500 > /sys/kernel/config/usb_gadget/g1/configs/c.1/MaxPower
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1/strings/0x409
ln -s /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0/ /sys/kernel/config/usb_gadget/g1/configs/c.1/
ls /sys/class/udc/ | xargs echo > /sys/kernel/config/usb_gadget/g1/UDC

```

注意：测试过程中发现会有如下报错，检查上述配置是否存在 SuperSpeed 的描述符未配置。

```

[ 41.788046] configfs-gadget gadget: uvc: uvc_function_bind()
[ 41.794433] udc 4d00000.xhci2-controller: failed to start g1: -19
echo: write error: No such device

```

8.1.6 小机做 HID

依赖配置项：

```

CONFIG_USB_CONFIGFS_F_HID=y

Device Drivers -->
[*] USB support --->
<*> USB Gadget Support --->
<*> USB Gadget functions configurable through configfs
 HID function

```

configs 配置方法：

```
mount -t configs none /sys/kernel/config/
mkdir /sys/kernel/config/usb_gadget/g1
echo 0x0525 > /sys/kernel/config/usb_gadget/g1/idVendor
echo 0xa4ac > /sys/kernel/config/usb_gadget/g1/idProduct
mkdir /sys/kernel/config/usb_gadget/g1/strings/0x409
mkdir /sys/kernel/config/usb_gadget/g1/functions/hid.usb0
echo 512 > /sys/kernel/config/usb_gadget/g1/functions/hid.usb0/report_length
echo -ne <report_desc> > /sys/kernel/config/usb_gadget/g1/functions/hid.usb0/report_desc
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1
echo 0xc0 > /sys/kernel/config/usb_gadget/g1/configs/c.1/bmAttributes
echo 500 > /sys/kernel/config/usb_gadget/g1/configs/c.1/MaxPower
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1/strings/0x409
ln -s /sys/kernel/config/usb_gadget/g1/functions/hid.usb0/ /sys/kernel/config/usb_gadget/g1/configs/c.1/
ls /sys/class/udc/ | xargs echo > /sys/kernel/config/usb_gadget/g1/UDC
```

说明

report_desc 根据需求自定义，示例如下。

```
\\x05\\x01\\x09\\x06\\xa1\\x01\\x05\\x07\\x19\\xe0\\x29\\xe7\\x15\\x00\\x25\\x01\\x75\\x01\\x95\\x08\\x81\\x02\\x95\\x01\\x75\\x08\\x81\\x03\\x95\\x05\\x75\\x01\\x05\\x08\\x19\\x01\\x29\\x05\\x91\\x02\\x95\\x01\\x75\\x03\\x91\\x03\\x95\\x06\\x75\\x08\\x15\\x00\\x25\\x65\\x05\\x07\\x19\\x00\\x29\\x65\\x81\\x00\\xc0
```

8.1.7 小机做 rndis

依赖配置项：

```
CONFIG_USB_CONFIGFS_RNDIS=y
Device Drivers --->
[*] USB support --->
<*> USB Gadget Support --->
<*> USB Gadget functions configurable through configs
[*] RNDIS
```

configs 配置方法：

```
mount -t configs none /sys/kernel/config
mkdir /sys/kernel/config/usb_gadget/g1
echo "0x1f3a" > /sys/kernel/config/usb_gadget/g1/idVendor
echo "0x200a" > /sys/kernel/config/usb_gadget/g1/idProduct
mkdir /sys/kernel/config/usb_gadget/g1/strings/0x409
mkdir /sys/kernel/config/usb_gadget/g1/functions/rndis.usb0
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1
echo 0xc0 > /sys/kernel/config/usb_gadget/g1/configs/c.1/bmAttributes
echo 500 > /sys/kernel/config/usb_gadget/g1/configs/c.1/MaxPower
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1/strings/0x409
ln -s /sys/kernel/config/usb_gadget/g1/functions/rndis.usb0/ /sys/kernel/config/usb_gadget/g1/configs/c.1/rndis.usb0
ls /sys/class/udc/ | xargs echo > /sys/kernel/config/usb_gadget/g1/UDC
```

8.1.8 小机做 acm

依赖配置项：

```
CONFIG_USB_ACM=y
CONFIG_USB_CONFIGFS_ACM=y

Device Drivers --->
[*] USB support --->
<*> USB Modem (CDC ACM) support
<*> USB Gadget Support --->
<*> USB Gadget functions configurable through configs
[*] Abstract Control Model (CDC ACM)
```

configs 配置方法：

```
mount -t configs none /sys/kernel/config
mkdir /sys/kernel/config/usb_gadget/g1
echo "0x1f3a" > /sys/kernel/config/usb_gadget/g1/idVendor
echo "0x0007" > /sys/kernel/config/usb_gadget/g1/idProduct
mkdir /sys/kernel/config/usb_gadget/g1/strings/0x409
mkdir /sys/kernel/config/usb_gadget/g1/functions/acm.usb0
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1
echo 0xc0 > /sys/kernel/config/usb_gadget/g1/configs/c.1/bmAttributes
echo 500 > /sys/kernel/config/usb_gadget/g1/configs/c.1/MaxPower
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1/strings/0x409
ln -s /sys/kernel/config/usb_gadget/g1/functions/acm.usb0 /sys/kernel/config/usb_gadget/g1/configs/c.1/acm.usb0
ls /sys/class/udc/ | xargs echo > /sys/kernel/config/usb_gadget/g1/UDC
```

注意：出现以下报错时，检查 CONFIG_USB_ACM 配置是否打开。

```
[ 87.430781] Config c/1 of g1 needs at least one function.
[ 87.436844] ERR: sunxi_udc_dequeue: driver is null
[ 87.442269] configs-gadget 4100000.udc-controller: failed to start g1: -22
echo: write error: Invalid argument
```

8.1.9 小机做 adb

依赖配置项：

```
CONFIG_NET=y
CONFIG_UNIX=y
CONFIG_INET=y
CONFIG_USB_CONFIGFS_F_FS=y

[*] Networking support --->
Networking options --->
<*> Unix domain sockets
[*] TCP/IP networking
Device Drivers --->
[*] USB support --->
<*> USB Gadget Support --->
<*> USB Gadget functions configurable through configs
[*] Function filesystem (FunctionFS)
```

configs 配置方法：

```
mount -t configfs none /sys/kernel/config
mkdir /sys/kernel/config/usb_gadget/g1
echo "0x18d1" > /sys/kernel/config/usb_gadget/g1/idVendor
echo "0x0002" > /sys/kernel/config/usb_gadget/g1/idProduct
mkdir /sys/kernel/config/usb_gadget/g1/strings/0x409
echo "20080411" > /sys/kernel/config/usb_gadget/g1/strings/0x409/serialnumber
echo "Android" > /sys/kernel/config/usb_gadget/g1/strings/0x409/manufacturer
mkdir /sys/kernel/config/usb_gadget/g1/functions/ffs.adb
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1
echo 0xc0 > /sys/kernel/config/usb_gadget/g1/configs/c.1/bmAttributes
echo 500 > /sys/kernel/config/usb_gadget/g1/configs/c.1/MaxPower
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1/strings/0x409
ln -s /sys/kernel/config/usb_gadget/g1/functions/ffs.adb/ /sys/kernel/config/usb_gadget/g1/configs/c.1/ffs.adb
mkdir /dev/usb-ffs
mkdir /dev/usb-ffs/adb
mount -o uid=2000,gid=2000 -t functionfs adb /dev/usb-ffs/adb/
ls /sys/class/udc/ | xargs echo > /sys/kernel/config/usb_gadget/g1/UDC
```

注意：按照上述配置出现以下报错时需要执行adb & 开启 adb 守护进程。

```
[ 69.369687] ERR: sunxi_udc_dequeue: driver is null
[ 69.375097] configfs-gadget 4100000.udc-controller: failed to start g1: -19
echo: write error: No such device
```

8.2 性能测试示例

主机端性能一般用硬盘 SSD 进行衡量；设备端性能可以通过 ATTO 工具或者 MTP/Mass Storage 拷贝进行测试，详见“性能分析”章节。此外，也可以通过 USB 协议分析仪（推荐 LeCroy 的 USB Protocol Suite 系列可通过 Timing Calculator 界面自动计算性能和带宽占用率）计算 USB 总线上的数据传输速率。

8.2.1 挂载设备

通过 OTG 线缆等接入外设并能够正常识别，可以根据识别到的设备节点进行挂载和性能测试，路径根据实际挂载路径修改。

```
/# mkdir mnt/usb
/# mount /dev/sda1 mnt/usb/
mount: mounting /dev/sda1 on mnt/usb/ failed: Invalid argument
```

说明

- 在 Linux 上测试时，出现 `mount: mounting /dev/sda1 on /mnt/usb/ failed: Invalid argument`，只需要对 `/dev/sda1` 进行格式化即可；格式化 U 盘：`mkfs.ext4 /dev/sda1`。
- 在 Android 上测试时，外设会自动挂载到 `/storage` 上，经过 fuse 导致测试速率变慢，里面会有很多权限管理的操作；如果要测试单纯的硬件接口性能在 `/mnt/media_rw/` 里进行测试。
- 测试性能时应当注意以下几点：a. 测试峰值性能需要格式化硬盘；b. 测试 Android 性能时，需要开启 `antutu-benchmark` 软件主界面，确保系统进入性能模式，此时大小核以最高频运行；c. 测试 Android 应用拷贝文件性能时，建议使用 Settings 系统设置应用里 Storage 项下的 Files 应用程序，该应用为 Android Mainline 的官方应用，Files by Google 为 GMS 包的第三方应用。

8.2.2 写性能测试

```
1 echo 3 > /proc/sys/vm/drop_caches
2 time dd if=/dev/zero of=/mnt/usb/test.bin bs=4M count=1024 conv=sync
```

8.2.3 读性能测试

```
1 echo 3 > /proc/sys/vm/drop_caches
2 time dd if=/mnt/usb/test.bin of=/dev/null bs=4M count=1024 conv=sync
```

8.2.4 测试节点

`/dev/zero` 和 `/dev/urandom` 是 Linux 系统中的两个特殊设备文件，它们各自具有不同的用途和特性。

在性能测试中，选择使用 `/dev/zero` 还是 `/dev/urandom` 取决于你的测试目标和具体场景：

使用 `/dev/zero`

- 内存和 I/O 性能测试：如果你想测试系统的内存分配、文件 I/O 性能或速度，使用 `/dev/zero` 更为合适。因为它提供的只是零字节，读取速度通常较快且开销低。
- 简单的性能基准：在需要生成大文件以测试文件系统性能或磁盘速度时，使用 `/dev/zero` 会更有效率。

使用 `/dev/urandom`

- 随机性相关的性能测试：如果你需要评估随机数生成的性能或随机数据处理的速度，那么使用 `/dev/urandom` 是合适的选择。
- 安全相关测试：在测试与加密或安全相关的应用时，使用 `/dev/urandom` 可以更好地模拟真实情况。




著作权声明

版权所有 © 2025 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明

、、**全志科技**、（不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。