



Linux USB 开发指南

**版本号: 1.3
发布日期: 2021.03.10**

版本历史

版本号	日期	制/修订人	内容描述
1.0	2020.6.30	AWA1458	1. 添加初始版本。
1.1	2020.10.27	AWA1458	1. 添加 Gadget 配置；2. 添加端点修改；3. 更换清晰图片等。
1.2	2021.01.18	AWA1458	1. 添加 USB Standby 注意事项；
1.3	2021.03.10	XAA0190	1. 更换清晰图片；2. 添加使用范围；3. 添加功能配置说明等；



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 模块配置介绍	2
2.3.1 Device Tree 配置说明	2
2.3.2 board.dts 配置说明	4
2.3.3 kernel menuconfig 配置说明	6
2.4 源码结构介绍	9
2.5 驱动框架介绍	10
2.6 Gadget 配置	11
2.6.1 打开内核配置	12
2.6.2 linux-4.x USB Gadget 配置流程	12
2.7 端点配置	13
2.7.1 端点 fifo 大小	13
2.7.2 端点的属性	13
2.7.3 定义端点的方向	13
2.8 调试方法	14
2.8.1 调试节点	14
2.8.1.1 USB0 调试节点	14
2.8.1.2 USB1 调试节点	14
2.8.2 眼图测试	15
2.8.2.1 USB Device 眼图测试	15
2.8.2.2 USB Host 眼图测试	15
3 FAQ	16
3.1 常见问题	16
3.1.1 USB 基本功能异常排查	16
3.1.1.1 USB Host 基本功能异常排查步骤	16
3.1.1.2 USB Device 基本功能异常排查步骤	17
4 附录	18
4.1 Linux-4.x Gadget 配置示例	18
4.1.1 小机做 mass storage	18
4.1.2 小机做 cdrom	18
4.1.3 小机做 UAC1	19
4.1.4 小机做 UAC2	19
4.1.5 小机做 UVC	19

4.1.6 小机做 HID	20
4.1.7 小机做 rndis	21
4.1.8 小机做 acm	21
4.1.9 小机做 adb	21
4.1.10 小机做 mass storage+adb	22
4.1.11 小机做 uvc+uac1	22
4.1.12 小机做 hid+cdrom	23
4.1.13 小机做 rndis+adb	24



插 图

2-1 Device Drivers 选项配置	6
2-2 USB Support 选项配置	6
2-3 USB Support 详细配置 1	7
2-4 USB Support 详细配置 2	7
2-5 USB Gadget Support 选项配置	8
2-6 USB Gadget Support 详细配置	8
2-7 USB Peripheral Controller 详细配置	9
2-8 SUNXI USB2.0 Dual Role Controller Support 详细配置	9
2-9 USB 驱动总体结构	11



1 前言

1.1 文档简介

介绍 USB 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: 适用产品列表

产品名称	内核版本	驱动文件
T509	Linux-4.9	drivers/usb/*
T507	Linux-4.9	drivers/usb/*
MR813	Linux-4.9	drivers/usb/*
R818	Linux-4.9	drivers/usb/*
A133	Linux-4.9	drivers/usb/*
A133	Linux-5.4	drivers/usb/*
A53	Linux-4.9	drivers/usb/*
R528	Linux-5.4	drivers/usb/*
H616	Linux-4.9	drivers/usb/*
H616	Linux-5.4	drivers/usb/*

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。

2.2 相关术语介绍

表 2-1: 术语介绍

术语	说明
USB	Universal Serial Bus, 通用串行总线
OTG	On-The-Go
ADB	Android Debug Bridge, Android 调试桥
Gadget	小配件
HCD	Host Controller Driver, 主机控制器驱动
UDC	USB Device Controller, USB 设备控制器
HCI	Host Controller Interface, 主机控制器接口
EHCI	Enhanced Host Controller Interface, 增强型主机控制器接口
OHCI	Open Host Controller Interface, 开放式主机控制器接口

2.3 模块配置介绍

2.3.1 Device Tree 配置说明

设备树中存在的是该类芯片所有平台的模块配置，设备树文件的路径为：kernel/linux-4.9/arch/arm64（32 位平台为 arm）/boot/dts/sunxi/xxx.dtsi(xxx 为具体芯片型号，如 sun50iw10p1 等), 设备树配置如下所示：

- USB0 配置

```

1  usb0:usb0@0 {
2      device_type = "usb0";
3      compatible = "allwinner,sunxi-otg-manager";
4      usb_port_type = <2>;
5      usb_detect_type = <1>;
6      usb_id_gpio;
7      usb_det_vbus_gpio;
8      usb_regulator_io = "nocare";
9      usb_wakeup_suspend = <0>;
10     usb_luns = <3>;
11     usb_serial_unique = <0>;
12     usb_serial_number = "20080411";
13     rndis_wceis = <1>;
14     status = "okay";
15 };
16
17 udc:udc-controller@0x05100000 {
18     compatible = "allwinner,sunxi-udc";
19     reg = <0x0 0x05100000 0x0 0x1000>, /*udc base*/
20         <0x0 0x00000000 0x0 0x100>, /*sram base*/
21         <0x0 0x05200000 0x0 0x1000>; /*usb1 base, for common circuit*/
22     interrupts = <GIC_SPI 32 IRQ_TYPE_LEVEL_HIGH>; /*设备使用的中断*/
23     clocks = <&clk_usbphy0>, <&clk_usbotg>, <&clk_usbehci1>, <&clk_usbphy1>; /*设备使用的时钟*/
24     status = "okay"; /*是否使能该设备*/
25 };
26
27 ehci0:ehci0-controller@0x05101000 {
28     compatible = "allwinner,sunxi-ehci0";
29     reg = <0x0 0x05101000 0x0 0xFFF>, /*hci0 base*/
30         <0x0 0x00000000 0x0 0x100>,
31         <0x0 0x05100000 0x0 0x1000>,
32         <0x0 0x07010250 0x0 0x10>, /*prcm base, for usb standby*/
33         <0x0 0x05200000 0x0 0x1000>; /*usb1 base, for common circuit*/
34     interrupts = <GIC_SPI 30 IRQ_TYPE_LEVEL_HIGH>;
35     clocks = <&clk_usbphy0>, <&clk_usbehci0>, <&clk_usbehci1>, <&clk_usbphy1>;
36     hci_ctrl_no = <0>; /*主机控制器的序列*/
37     status = "okay";
38 };
39
40 ohci0:ohci0-controller@0x05101400 {
41     compatible = "allwinner,sunxi-ohci0";
42     reg = <0x0 0x05101000 0x0 0xFFF>, /*hci0 base*/
43         <0x0 0x00000000 0x0 0x100>,
44         <0x0 0x05100000 0x0 0x1000>,
45         <0x0 0x07010250 0x0 0x10>, /*prcm base, for usb standby*/
46         <0x0 0x05200000 0x0 0x1000>; /*usb1 base, for common circuit*/
47     interrupts = <GIC_SPI 31 IRQ_TYPE_LEVEL_HIGH>;
48     clocks = <&clk_usbphy0>, <&clk_usbohci0>, <&clk_usbohci1>, <&clk_usbphy1>;
49     hci_ctrl_no = <0>;
50     status = "okay";
51 };

```

• USB1 配置

```

1  usb1:usb1@0 {
2      device_type = "usb1";
3      usb_regulator_io = "nocare";
4      usb_wakeup_suspend = <0>;

```

```

5   status = "okay";
6 };
7
8 ehci1:ehci1-controller@0x05200000 {
9     compatible = "allwinner,sunxi-ehci1";
10    reg = <0x0 0x05200000 0x0 0xFFF>,
11          <0x0 0x00000000 0x0 0x100>,
12          <0x0 0x05100000 0x0 0x1000>,
13          <0x0 0x07010250 0x0 0x10>;
14    interrupts = <GIC_SPI 33 IRQ_TYPE_LEVEL_HIGH>;
15    clocks = <&clk_usbphy1>, <&clk_usbehci1>;
16    hci_ctrl_no = <1>;
17    status = "okay";
18 };
19
20 ohci1:ohci1-controller@0x05200400 {
21     compatible = "allwinner,sunxi-ohci1";
22     reg = <0x0 0x05200000 0x0 0xFFF>,
23           <0x0 0x00000000 0x0 0x100>,
24           <0x0 0x05100000 0x0 0x1000>,
25           <0x0 0x07010250 0x0 0x10>;
26     interrupts = <GIC_SPI 34 IRQ_TYPE_LEVEL_HIGH>;
27     clocks = <&clk_usbphy1>, <&clk_usbohci1>, <&clk_usbohci1_12m>, <&clk_osc48md4>, <&
28             clk_hosc>, <&clk_losc>;
29     hci_ctrl_no = <1>;
30     status = "okay";
31 };

```

2.3.2 board.dts 配置说明

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 默认配置信息。不同 soc、版型及内核版本对应的 board.dts 具体路径如下:device/config/chips/soc/configs/{board}/\${内核版本}/board.dts。

• USB0 配置

```

1   usb0:usb0@0 {
2       device_type = "usb0";
3       usb_port_type = <0x2>;
4       usb_detect_type = <0x1>;
5       usb_id_gpio = <&pio PH 8 0 0 0xffffffff 0xffffffff>;
6       usb_det_vbus_gpio = "axp_ctrl";
7       usb_regulator_io = "nocare";
8       det_vbus_supply = <&usb_power_supply>;
9       usb_wakeup_suspend = <0>;
10      usb_luns = <3>;
11      usb_serial_unique = <0>;
12      usb_serial_number = "20080411";
13      rndis_wceis = <1>;
14      status = "okay";
15  };
16  注： (1) usb_port_type: usb0口默认的模式。
17      置0: devcie模式;
18      置1: host模式;

```

```

19 置2: otg模式。
20  (2) usb_detect_type: usb0口otg检测模式。
21 置0: 不做检测;
22 置1: vbus/id检测;
23 置2: id/dpdm检测。
24  (3) usb_wakeup_suspend: standby模式。
25 置0: super standby模式;
26 置1: usb standby模式, 支持远程唤醒。
27
28 udc:udc-controller@0x51000000 {
29     det_vbus_supply = <&usb_power_supply>
30 }
31 ehci0:ehci0-controller@0x05101000 {
32     drvvbus-supply = <&reg_drivevbus>;
33 };
34
35 ohci0:ohci0-controller@0x05101400 {
36     drvvbus-supply = <&reg_drivevbus>;
37 };

```

📖 说明

若使用 **usb standby** 模式, 需注意如下:

- 1、IC 支持远程唤醒;
- 2、若条件 1 满足, 相关硬件部分需严格按照《硬件设计文档》设计;
- 3、若条件 1、2 满足, 额外添加属性 “wake-up-source;”, 启用 **usb standby** 功能。

• USB1 配置

```

1  usb1:usb1@0 {
2      device_type = "usb1";
3      usb_regulator_io = "nocare";
4      usb_wakeup_suspend = <0>;
5      status = "okay";
6  };
7
8  ehci1:ehci1-controller@0x05200000 {
9      drvvbus-supply = <&reg_usb1_vbus>;
10 };
11
12 ohci1:ohci1-controller@0x05200400 {
13     drvvbus-supply = <&reg_usb1_vbus>;
14 };

```

• Vbus 配置

```

1 reg_usb1_vbus: usb1-vbus {
2     compatible = "regulator-fixed";
3     gpio = <&pio PH 10 1 2 0 1>;
4     regulator-name = "usb1-vbus";
5     regulator-min-microvolt = <5000000>;
6     regulator-max-microvolt = <5000000>;
7     regulator-enable-ramp-delay = <1000>;
8     enable-active-high;
9 };

```

2.3.3 kernel menuconfig 配置说明

进入内核根目录，执行 make ARCH=arm menuconfig（64 位平台为 make ARCH=arm64 menuconfig）进入配置主界面，并按以下步骤操作：

选择 Device Drivers 选项进入下一级配置，如下图所示：

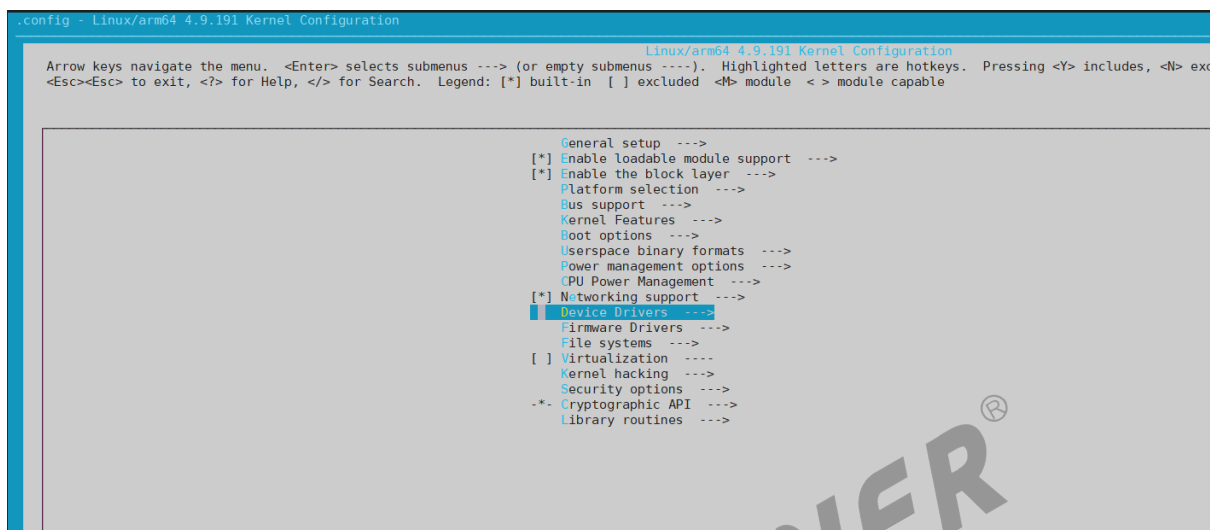


图 2-1: Device Drivers 选项配置

选择 USB support 选项，进入下一级配置，如下图所示：

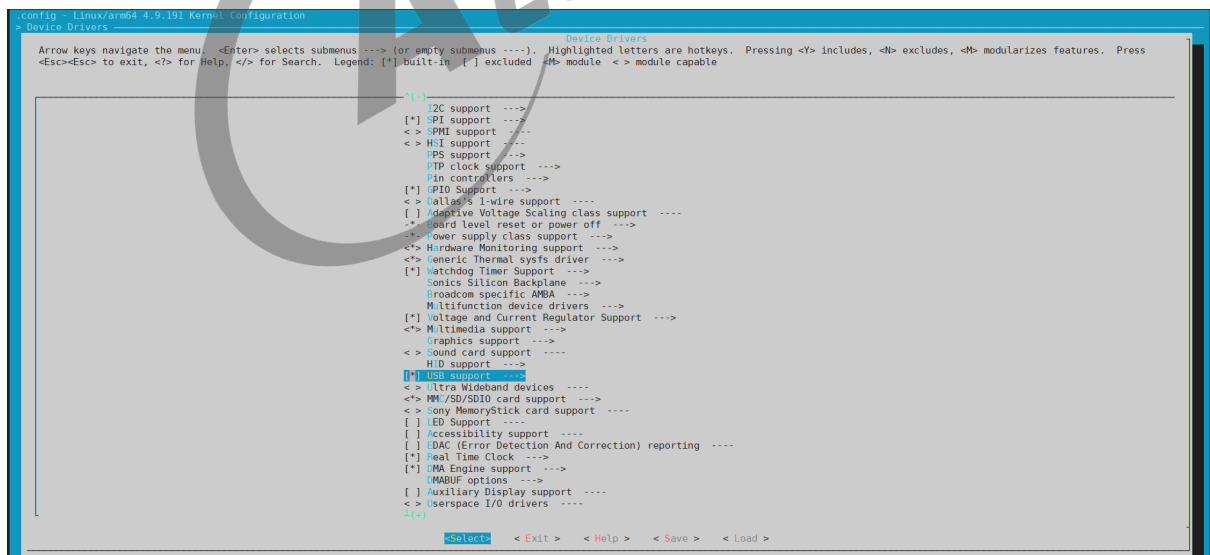


图 2-2: USB Support 选项配置

打开如下两图的选项，如下图所示：

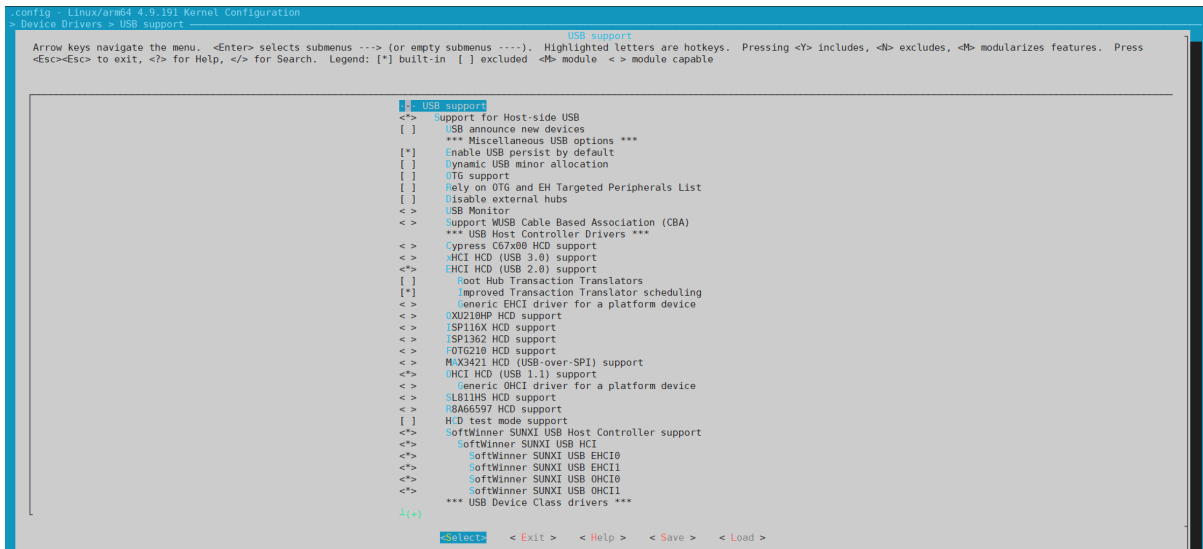


图 2-3: USB Support 详细配置 1

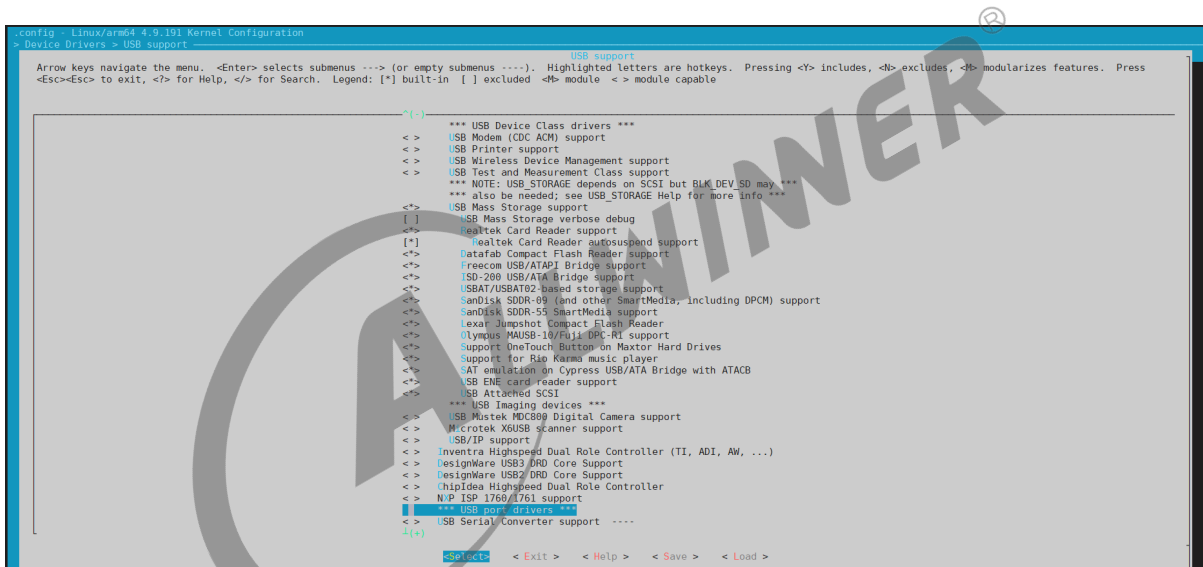


图 2-4: USB Support 详细配置 2

选择 USB Gadget Support，进入下一级配置，如下图所示：

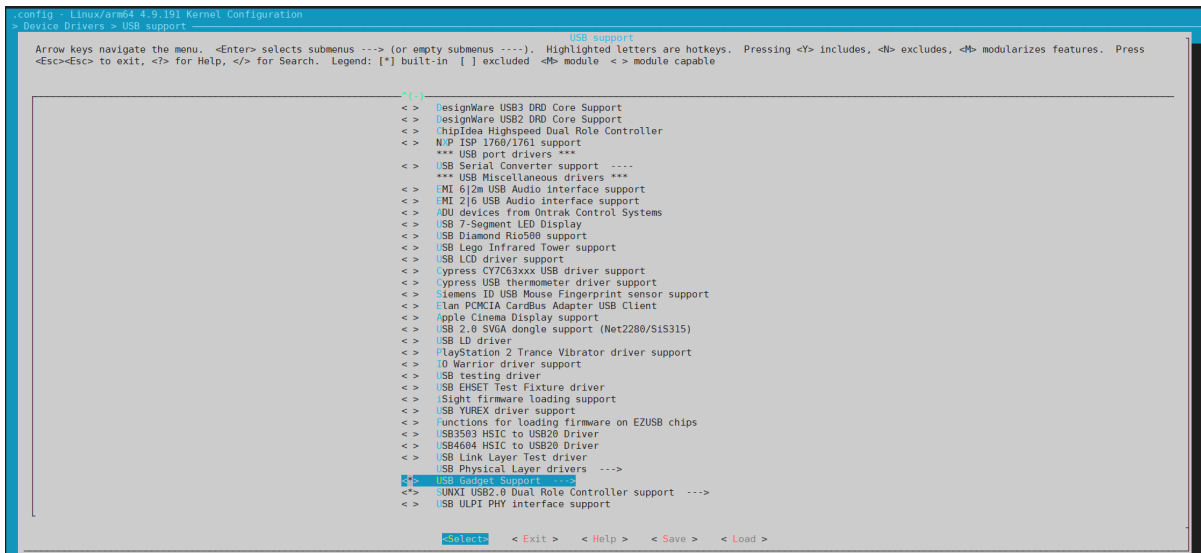


图 2-5: USB Gadget Support 选项配置

打开下图的选项，并在对应配置中打开所需的功能性配置，如：需要存储功能时，需打开下图中的“mass storage”配置，如下图所示：

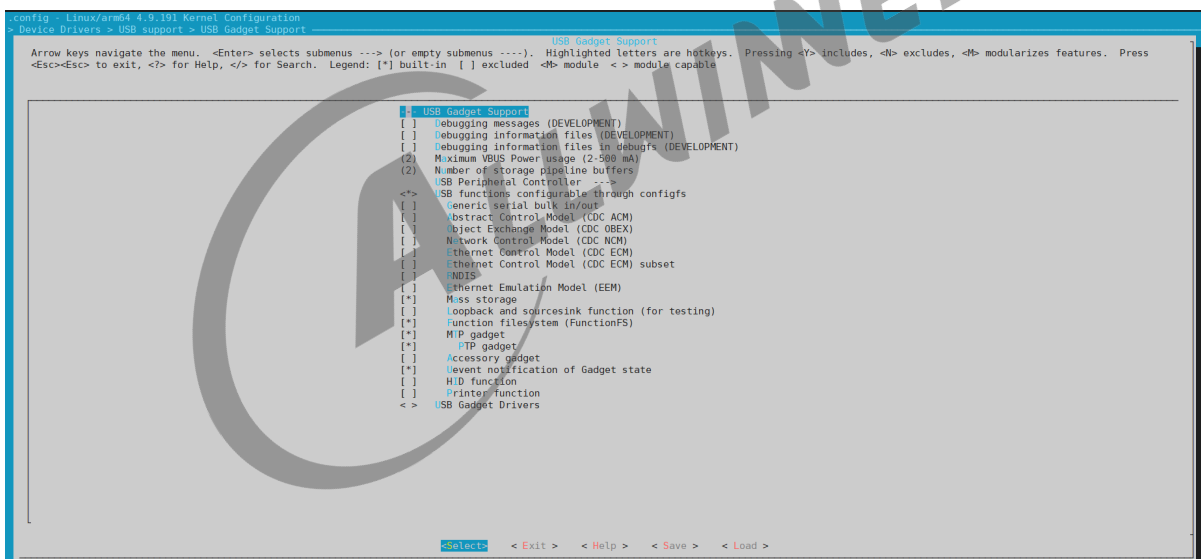


图 2-6: USB Gadget Support 详细配置

进入 USB Peripheral Controller，并打开下图选项：

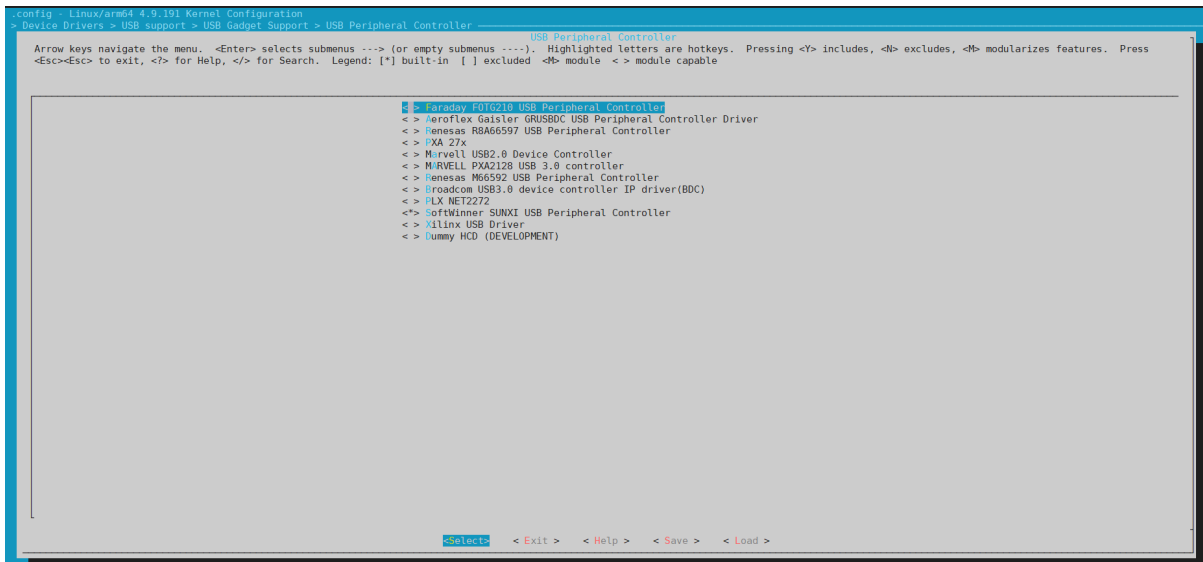


图 2-7: USB Peripheral Controller 详细配置

返回上一级，即 USB support，进入 SUNXI USB2.0 Dual Role controller support，并打开下图选项，如下图所示：

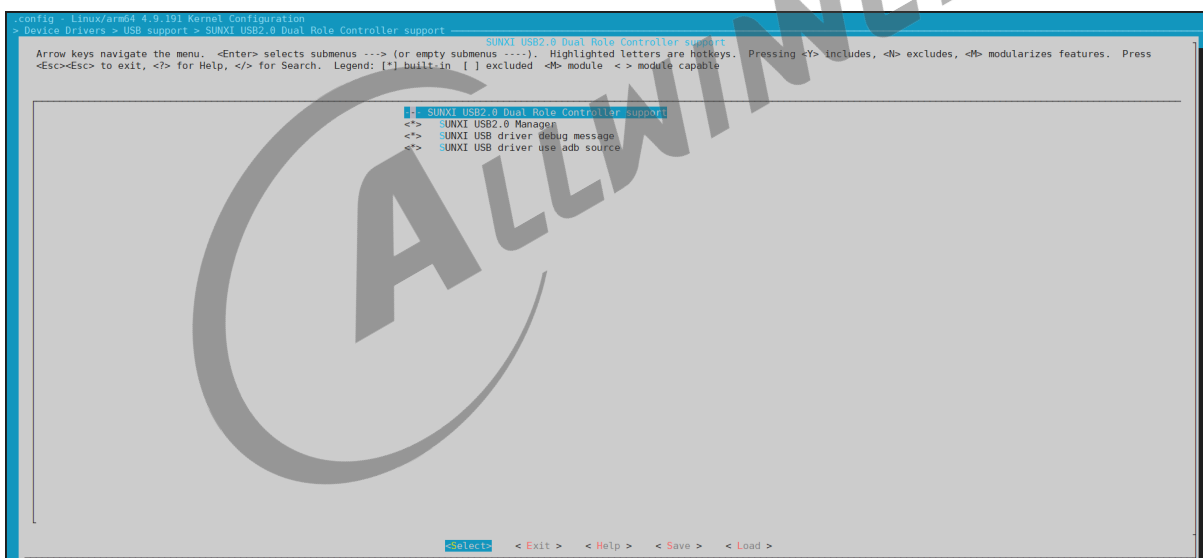


图 2-8: SUNXI USB2.0 Dual Role Controller Support 详细配置

2.4 源码结构介绍

USB 驱动的源代码位于内核 drivers/usb 目录下，如下是 sunxi 平台相关源码：

- Host

```
drivers/usb/host/  
├─ ehci_sunxi.c  
├─ ohci_sunxi.c  
├─ sunxi_hci.c  
└─ sunxi_hci.h
```

- UDC 和 Manager

```
drivers/usb/sunxi_usb/  
├─ include  
│   ├── sunxi_hcd.h  
│   ├── sunxi_sys_reg.h  
│   ├── sunxi_udc.h  
│   ├── sunxi_usb_board.h  
│   ├── sunxi_usb_bsp.h  
│   ├── sunxi_usb_config.h  
│   ├── sunxi_usb_debug.h  
│   └─ sunxi_usb_typedef.h  
├─ Kconfig  
├─ Makefile  
├─ manager  
│   ├── usbc0_platform.c  
│   ├── usbc_platform.h  
│   ├── usb_hcd_servers.c  
│   ├── usb_hcd_servers.h  
│   ├── usb_hw_scan.c  
│   ├── usb_hw_scan.h  
│   ├── usb_manager.c  
│   ├── usb_manager.h  
│   ├── usb_msg_center.c  
│   └─ usb_msg_center.h  
├─ misc  
│   └─ sunxi_usb_debug.c  
├─ udc  
│   ├── sunxi_udc_board.c  
│   ├── sunxi_udc_board.h  
│   ├── sunxi_udc.c  
│   ├── sunxi_udc_config.h  
│   ├── sunxi_udc_debug.c  
│   ├── sunxi_udc_debug.h  
│   ├── sunxi_udc_dma.c  
│   └─ sunxi_udc_dma.h  
└─ usbc  
    ├── usbc.c  
    ├── usbc_dev.c  
    ├── usbc_i.h  
    └─ usbc_phy.c
```

2.5 驱动框架介绍

Linux 内核提供了完整的 USB 驱动程序框架。USB 总线采用树形结构，在一条总线上只能有唯一的主机设备。Linux 内核从主机和设备两个角度观察 USB 总线结构。下图是 Linux 内核从主机和设备两个角度观察 USB 总线结构的示意图。

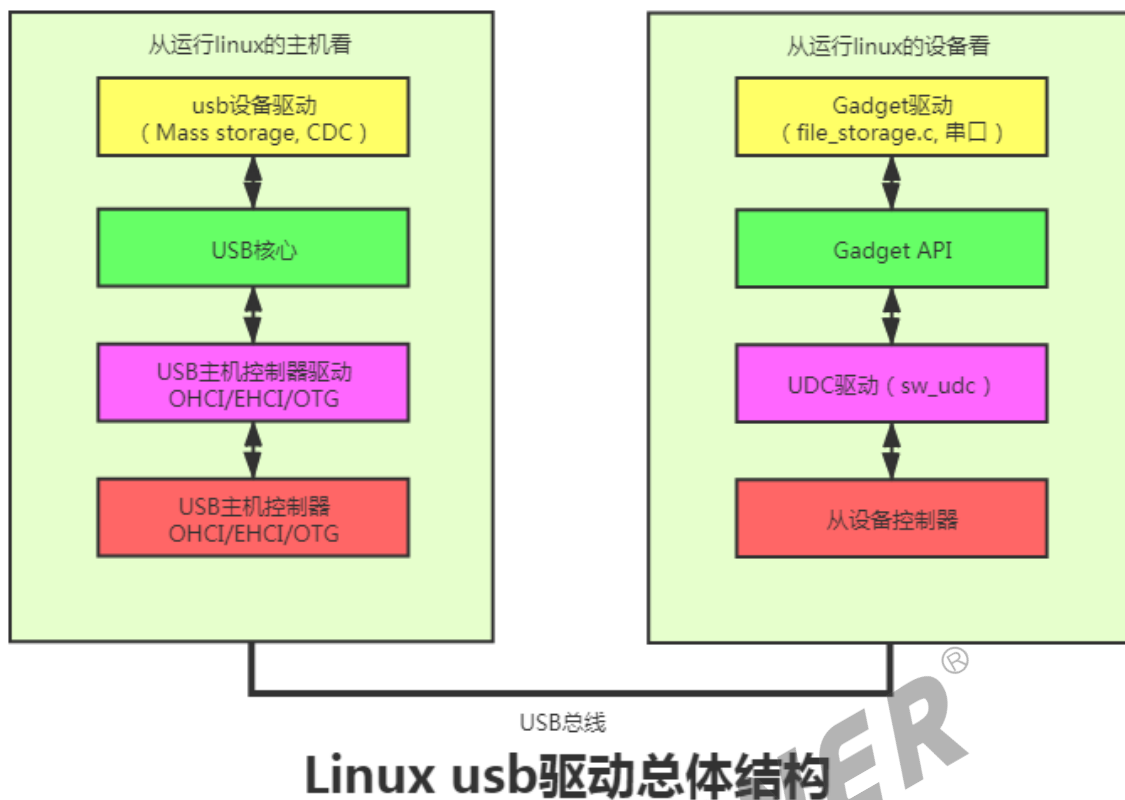


图 2-9: USB 驱动总体结构

USB 子系统主要任务包括：

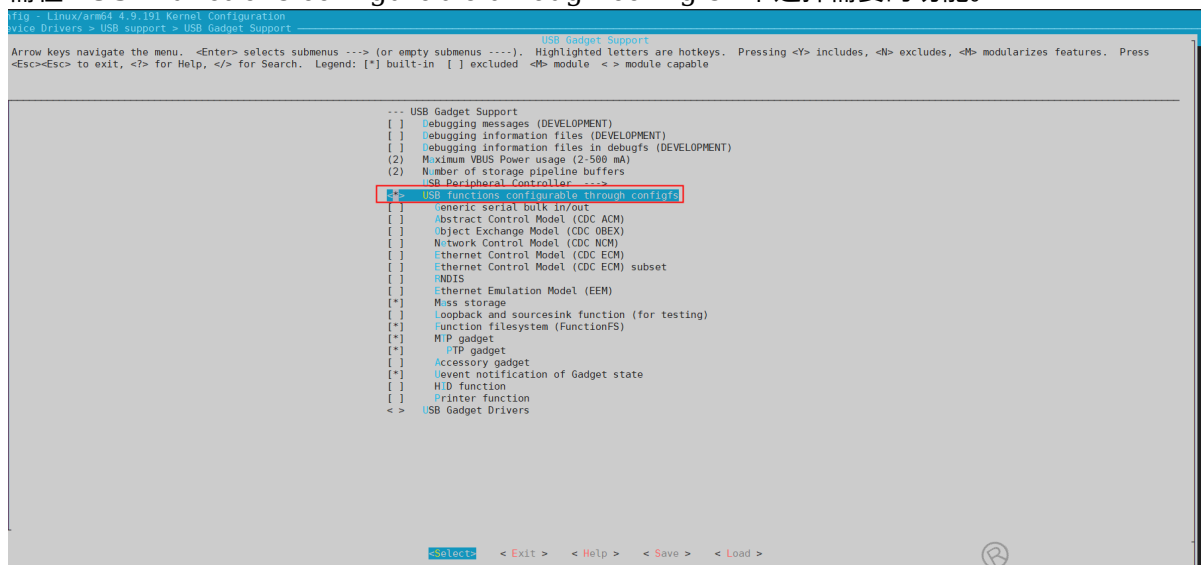
- 注册和管理设备驱动；
- USB 设备寻找驱动，并初始化和配置设备；
- 内核中表现设备的树形结构；
- 与设备交互。

2.6 Gadget 配置

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，通用的配置流程如下：

2.6.1 打开内核配置

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



2.6.2 linux-4.x USB Gadget 配置流程

Linux-4.x 使用 configs 框架实现 composite gadget 功能。具体流程如下：

```
mount -t configs none /sys/kernel/config
1、建立gadgets，并写入gadget的PID、VID、序列号等信息
mkdir /sys/kernel/config/usb_gadget/g1
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
2、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
3、建立功能functions
mkdir /sys/kernel/config/usb_gadget/g1/functions/<name>.<instance name>
说明: <name> function name
      <instance name>任意字符串
4、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
5、使能gadget
echo <udc name> > UDC
```

常见 Gadget 功能的配置方式见附录。

2.7 端点配置

在 Gadget 配置使用过程中，可能出现端点的默认配置方式无法满足需求的情况，故需对端点进行修改满足需求。可参考现有的端点进行修改。譬如将批量端点改成中断端点，参考现有的中断端点进行修改即可。改动内容包括端点 fifo 大小，端点属性，端点方向。

2.7.1 端点 fifo 大小

```
1 以4k平台为例：
2 static const struct sw_udc_fifo ep_fifo[] = {
3     {ep0name,      0,    512,  0}, /*name,  fifo_addr,  fifo_size, double_fifo*/
4     {ep1in_bulk_name, 512, 512,  0},
5     {ep1out_bulk_name, 1024, 512,  0},
6     {ep2in_bulk_name, 1536, 512,  0},
7     {ep2out_bulk_name, 2048, 512,  0},
8     {ep3_iso_name,    2560, 1024, 0},
9     {ep4_int_name,    3584, 512,  0},
10 };
```

2.7.2 端点的属性

```
1 .ep[2] = {
2     .num      = 1,
3     .ep = {
4         .name      = ep1out_bulk_name,
5         .ops        = &sunxi_udc_ep_ops,
6         .maxpacket  = SW_UDC_EP_FIFO_SIZE,
7         .maxpacket_limit = SW_UDC_EP_FIFO_SIZE,
8         .caps       = USB_EP_CAPS(USB_EP_CAPS_TYPE_BULK,
9                                 USB_EP_CAPS_DIR_OUT),
10    },
11    .dev        = &sunxi_udc,
12    .bEndpointAddress = (USB_DIR_OUT | 1),
13    .bmAttributes = USB_ENDPOINT_XFER_BULK,
14 },
15 };
```

2.7.3 定义端点的方向

```
1 /**
2  * ep_fifo_in[i] = {n} i: the physic ep index, n: ep_fifo's index for the ep
3  *
4  * eg: ep_fifo_in[2] = {3} ==> ep2_in is in ep_fifo[3]
5  *
6  * ep3_iso_name and ep4_int_name cannot be tx or rx simultaneously.
```

```
7  *
8  */
9  static const int ep_fifo_in[] = {0, 1, 3, 5, 6, 7};
10 static const int ep_fifo_out[] = {0, 2, 4, 5, 6, 8};
11 };
```

2.8 调试方法

2.8.1 调试节点

2.8.1.1 USB0 调试节点

查看 USB0 当前 Role

```
cat /sys/devices/platform/soc/usb0/otg_role
```

手动切换到 Host 模式

```
cat /sys/devices/platform/soc/usb0/usb_host
```

手动切换到 Device 模式

```
cat /sys/devices/platform/soc/usb0/usb_device
```

2.8.1.2 USB1 调试节点

卸载主机驱动

通过下述命令找到主机驱动节点及对应路径

```
find -name ehci_enable
find -name ohci_enable
```

然后根据上述结果，按如下命令卸载主机驱动 (以 t5 平台为例)

```
echo 0 > sys/devices/platform/soc/5200000.ehci1-controller/ehci_enable
echo 0 > sys/devices/platform/soc/5200000.ohci1-controller/ohci_enable
```

加载主机驱动

通过下述命令找到主机驱动节点及对应路径

```
find -name ehci_enable
find -name ohci_enable
```

然后根据上述结果，按如下命令加载主机驱动（以 t5 平台为例）

```
echo 1 > sys/devices/platform/soc/52000000.ehci1-controller/ehci_enable  
echo 1 > sys/devices/platform/soc/52000000.ohci1-controller/ohci_enable
```

2.8.2 眼图测试

2.8.2.1 USB Device 眼图测试

```
获取otg_ed_test的路径path  
find /sys/ -name otg_ed_test  
测试眼图命令  
echo test_pack > path/otg_ed_test
```

2.8.2.2 USB Host 眼图测试

```
获取ed_test的路径path  
find /sys/ -name ed_test  
测试眼图命令  
echo test_pack > path/ed_test
```

3 FAQ

3.1 常见问题

3.1.1 USB 基本功能异常排查

3.1.1.1 USB Host 基本功能异常排查步骤

- 多找几个 USB 设备试试，排除个别 USB 设备本身的问题。
- 多更换几根 USB 线缆试试，排除个别 USB 线缆的问题。
- 多找几个 PC 主机做相同的实验，作为参考对比。若在 PC 有相同现象，则认为正常。
- 若硬件有多个 USB 口，尝试同样条件下测试其他 USB 口的主机功能是否正常。
- 样机设备 USB 口外接独立供电的 USB-HUB 设备，再将 USB 设备连接到 USB-HUB 上，确认主机功能是否正常。
- 确认主机驱动是否加载成功。
 - (1) 若为 USB0 口，则可通过如下方式确认：

```
cat /sys/devices/platform/soc/usbc0/btg_role
```

- (2) 若为 USB1 口，可通过如下方式确认：

```
cat sys/devices/platform/soc/52000000.ehci1-controller/ehci_enable
cat sys/devices/platform/soc/52000000.ohci1-controller/ohci_enable
若为0，则没有加载Host驱动。
```

- 重新加载 Host 驱动，确认此时功能是否正常。
 - (1) 若为 USB0 口，则可通过如下方式：
方式 1：重新插拔 OTG 线。
方式 2：手动切换到 Host 模式。
 - (2) 若为 USB1 口，则可通过卸载驱动、再加载驱动。
- 对比 SDK 代码与最新发布的代码或者补丁，确认代码是否更新到最新。
- 同样条件下，分别打印出功能异常板子和功能正常板子的相关寄存器，并进行对比，确认是否有不同之处。
- 出现异常时，测试 USB 高速眼图是否正常。
- 若眼图测试未通过，可尝试调节眼图参数。

3.1.1.2 USB Device 基本功能异常排查步骤

- 多换几个 PC 主机做相同的测试，排除个别 PC 的问题。
- 多更换几根 USB 线缆做相同的测试，排除个别 USB 线缆的问题。
- 确认 Device 驱动是否加载成功，可通过如下方式：

(1) 通过 Log:

```
[ 104.732695] insmod_device_driver  
[ 104.732695]  
device_chose finished!
```

(2) 通过节点查看当前 Role。

- 重新加载 Device 驱动，确认此时功能是否恢复正常。
 - (1) 重新插拔 USB 线。
 - (2) 手动切换到 Device 模式。
- 对比 SDK 代码与最新发布的代码或者补丁，确认代码是否更新到最新。
- 同样条件下，分别打印出功能异常板子和功能正常板子的相关寄存器，并进行对比，确认是否有异常。
- 出现异常时，确认 USB 高速眼图是否正常。

4 附录

4.1 Linux-4.x Gadget 配置示例

4.1.1 小机做 mass storage

```
dd if=/dev/zero of=/dev/a.bin bs=1M count=100
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

4.1.2 小机做 cdrom

```
mount -t configfs 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。

4.1.3 小机做 UAC1

```
mount -t configfs 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
```

4.1.4 小机做 UAC2

```
mount -t configfs 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
```

4.1.5 小机做 UVC

```
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/uvic.usb0/streaming/mjpeg/m/720p/
dwFrameInterval
echo 333333 > /sys/kernel/config/usb_gadget/g1/functions/uvic.usb0/streaming/mjpeg/m/720p/
dwDefaultFrameInterval
echo 442368000 > /sys/kernel/config/usb_gadget/g1/functions/uvic.usb0/streaming/mjpeg/m/720p/
dwMinBitRate
echo 442368000 > /sys/kernel/config/usb_gadget/g1/functions/uvic.usb0/streaming/mjpeg/m/720p/
dwMaxBitRate
echo 1843200 > /sys/kernel/config/usb_gadget/g1/functions/uvic.usb0/streaming/mjpeg/m/720p/
dwMaxVideoFrameBufferSize
mkdir /sys/kernel/config/usb_gadget/g1/functions/uvic.usb0/streaming/header/h
ln -s /sys/kernel/config/usb_gadget/g1/functions/uvic.usb0/streaming/mjpeg/m/ /sys/kernel/
config/usb_gadget/g1/functions/uvic.usb0/streaming/header/h/
ln -s /sys/kernel/config/usb_gadget/g1/functions/uvic.usb0/streaming/header/h/ /sys/kernel/
config/usb_gadget/g1/functions/uvic.usb0/streaming/class/fs
ln -s /sys/kernel/config/usb_gadget/g1/functions/uvic.usb0/streaming/header/h/ /sys/kernel/
config/usb_gadget/g1/functions/uvic.usb0/streaming/class/hs
mkdir /sys/kernel/config/usb_gadget/g1/functions/uvic.usb0/control/header/h
ln -s /sys/kernel/config/usb_gadget/g1/functions/uvic.usb0/control/header/h/ /sys/kernel/
config/usb_gadget/g1/functions/uvic.usb0/control/class/fs/
ln -s /sys/kernel/config/usb_gadget/g1/functions/uvic.usb0/control/header/h/ /sys/kernel/
config/usb_gadget/g1/functions/uvic.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/uvic.usb0/ /sys/kernel/config/usb_gadget/g1
/configs/c.1/
ls /sys/class/udc/ | xargs echo > /sys/kernel/config/usb_gadget/g1/UDC

```

4.1.6 小机做 HID

```

mount -t configfs 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。

4.1.7 小机做 rndis

```
mount -t configfs 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
```

4.1.8 小机做 acm

```
mount -t configfs 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
```

4.1.9 小机做 adb

```
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/manufacture
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
```

4.1.10 小机做 mass storage+adb

```
mount -t configfs none /sys/kernel/config
mkdir /sys/kernel/config/usb_gadget/g1
echo "0x18d1" > /sys/kernel/config/usb_gadget/g1/idVendor
echo "0x0003" > /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/manufacture
mkdir /sys/kernel/config/usb_gadget/g1/functions/ffs.adb
mkdir /sys/kernel/config/usb_gadget/g1/functions/mass_storage.usb0
echo ${BLOCK_PATH} > /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/ffs.adb/ /sys/kernel/config/usb_gadget/g1/
configs/c.1/ffs.adb
ln -s /sys/kernel/config/usb_gadget/g1/functions/mass_storage.usb0/ /sys/kernel/config/
usb_gadget/g1/configs/c.1/mass_storage.usb0
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
```

4.1.11 小机做 uvc+uac1

```
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/uac1.usb0
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/
width
echo 720 > /sys/kernel/config/usb_gadget/g1/functions/uvc.usb0/streaming/mjpeg/m/720p/
height
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
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/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/
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
```

4.1.12 小机做 hid+cdrom

```
mount -t configfs 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/hid.usb0
mkdir /sys/kernel/config/usb_gadget/g1/functions/mass_storage.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
ln -s /sys/kernel/config/usb_gadget/g1/functions/hid.usb0/ /sys/kernel/config/usb_gadget/g1
/configs/c.1/hid.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
```

4.1.13 小机做 rndis+adb

```
mount -t configfs none /sys/kernel/config
mkdir /sys/kernel/config/usb_gadget/g1
echo "0x18d1" > /sys/kernel/config/usb_gadget/g1/idVendor
echo "0x0010" > /sys/kernel/config/usb_gadget/g1/idProduct
mkdir /sys/kernel/config/usb_gadget/g1/strings/0x409
mkdir /sys/kernel/config/usb_gadget/g1/functions/ffs.adb
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
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
```

著作权声明

版权所有 © 2021 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。