# Android 10

# USB 模块说明书

2.4

2020.03.05

# 文档履历

| 版本号 | 日期 | 制/修订人 | 内容描述 |
|---|---|---|---|
| 1.0 | 2016.12.03 | | |
| 2.0 | 2018.12.21 | | 修正一些参数，增加只支持 device 模式的配置说明 |
| 2.1 | 2019.01.29 | | 增加一些 USB 说明 |
| 2.2 | 2019.03.01 | | 增加 menuconfig 配置说明 |
| 2.3 | 2019.09.12 | | 修改‘USB 系统概念简介’章节的标点符号使用 |
| 2.4 | 2020.03.05 | | 针对 A100/A133 做修改 |

# 目录

1. 概述 . . . . . 1
1.1 编写目的 . . . . . 1
1.2 适用范围 . . . . . 1
1.3 相关人员 . . . . . 1
2. 模块介绍 . . . . . 2
2.1 模块功能介绍 . . . . . 2
2.2 相关术语介绍 . . . . . 3
2.2.1 硬件术语 . . . . . 3
2.2.2 软件术语 . . . . . 4
2.3 USB 驱动程序框架 . . . . . 4
2.4 源码结构介绍 . . . . . 5
2.5 模块配置介绍 . . . . . 6
2.5.1 menuconfig 配置 . . . . . 6
2.5.2 Device Tree 配置 . . . . . 12
2.5.3 board.dts 配置 . . . . . 13
3. FAQ . . . . . 15
3.1 调试节点 . . . . . 15
3.2 调试方法 . . . . . 15
3.2.1 USB Host . . . . . 15
3.2.2 USB device . . . . . 16

3.3 常见问题 . . . . . 16
3.3.1 无法枚举 u 盘 . . . . . 16
3.3.1.1 现象 . . . . . 16
3.3.1.2 复现方法 . . . . . 16
3.3.1.3 排查过程 . . . . . 17
3.3.1.4 解决方法 . . . . . 17
3.3.1.5 结论 . . . . . 17
3.4 android 开发者选项中将 usb 切换为 ``file transfer" 模式，系统死机重启 . . . . . 17
3.4.1 现象 . . . . . 17
3.4.2 复现方法 . . . . . 19
3.4.3 排查过程 . . . . . 19
3.4.4 解决方法 . . . . . 19
3.4.5 结论 . . . . . 19
4. Declaration . . . . . 20

# 1. 概述

## 1.1 编写目的

介绍 USB 模块配置方法，以及使用注意事项。

## 1.2 适用范围

本模块设计适用于 Linux-4.9 内核 AW 平台。

## 1.3 相关人员

sdk 维护人员、USB 驱动开发者、系统配置人。

# 2. 模块介绍

## 2.1 模块功能介绍

由于 USB 系统由两部组成，所以 usb 驱动也由相应两部分组成。

主机侧驱动主要由应用层驱动，用于管理 hub 和 hcd 的 usb core 层驱动，主机控制器驱动这三部分组成，应用层驱动主要是通过 usb 的基本数据通信结构 urb 对 usb 设备进行操作，这类驱动是针对某一个 usb 接口写的，所以也叫作 usb 接口驱动，通常在主机端驱动人员要写的就是这类驱动。usb core 驱动里包含一个守护进程，通常进程是休眠的，当 usb port 上产生变化，usb root hub 监控到其端口上有设备插入或拔出，它就会去唤醒守护进程，然后运行 usb 设备的枚举，进而运行后面操作。主机控制器驱动里包含两部分，一部分是和 usb 接口标准相关的驱动，一部分是和控制器本身相关的，其中和接口标准相关的驱动占主要部分。

在 usb 设备侧，驱动由三层组成：Upper Layers，gadget drivers, device controller drivers。upper layers 属于应用层，通过 gadget drivers 来使用和控制 device controllers；gadget drivers 使用 gadget API，实现与具体硬件无关；device controller drivers 提供 gadget API，实现 gadget ops 和 endpoint ops。

不管是在主机侧还是设备侧，一个 usb device 它主要可由配置、接口、端口三部分表示。在主机侧它们分别由 usb_host_config、usb_inferface、usb_host_endpoint 表示。而设备侧由 usb_configuration、usb_function 表示。像我们现在的智能手机它可以用来拍照也可以用来当 U 盘，这两种就属于不同的配置。一个 usb 设备它可能包含多个配置，在使用某些功能前必需选择好相应的配置。一个配置由多个接口组成，一个接口表示一个功能，一个接口里可以包含多个接口设置。每个接口设置里包含与实现某一功能相关的端口，配置、接口和端口结构示意如下图所示。


图 1: 接口和端口结构示意图

当一个 usb 设备插入到 host 端时，主机端的 root hub 通过轮训监控或中断方式来触发 usb 枚举。当 usb 设备通过枚举后，通过 usb 总线找到与其配置驱动，然后 usb 设备才能正常工作，这个枚举过程由主机和设备共同完成。对于没有运行操作系统的 usb 设备，生产产家已经把设备侧枚举所需过程固化在设备里，驱动工程师可不用去关心设备枚举，他们只需要完成 usb 接口驱动程序即可。但对于有运行操作系统的 usb 设备就完全不一样了，驱动工程师不仅要实现 usb 设备功能驱动，还得实现与主机枚举过程相对应的驱动。如主机在对 usb 设备进行复位操作后，主机会发送 usb 请求去设置 usb 设备的地址，相应的 usb 设备侧必须实现设备地址机制，并返回 0 长度 usb 请求，做为收到数据 ACK。当主机要获取 usb 设备各种描述符或设置各种配置时，usb 设备侧也要实现相应操作。

## 2.2 相关术语介绍

### 2.2.1 硬件术语

| 术语 | 解释说明 |
| --- | --- |
| HCD | 主机控制器驱动 |
| OHCI | Open Host Controller Interface，USB1.1 协议控制器 |
| EHCI | Enhanced Host Controller Interface，USB2.0 协议控制器 |
| UDC | Usb Device Controller |
| PHY | AW 平台 USB 控制器硬件参数模块 |
| OTG | On-The-Go 的缩写，USB2.0 兼容协议，用于嵌入式主机连接设备 |

### 2.2.2 软件术语

| 术语 | 解释说明 |
| --- | --- |
| URB | USB Request Block 用于组织每一次的 USB 设备驱动的数据传输请求 |
| EP | Endpoint 端点 |
| Mass Storage | 大容量存储器，如 u 盘，移动硬盘等 |
| HID | Human Interface Device，如鼠标键盘等输入设备 |
| ADB | Android Debug Bridge，Android 调试桥接器 |
| MTP | Media Transfer Protocol，媒体传输协议 |
| PTP | picture transfer protoco，图片传输协议 |

## 2.3 USB 驱动程序框架

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。

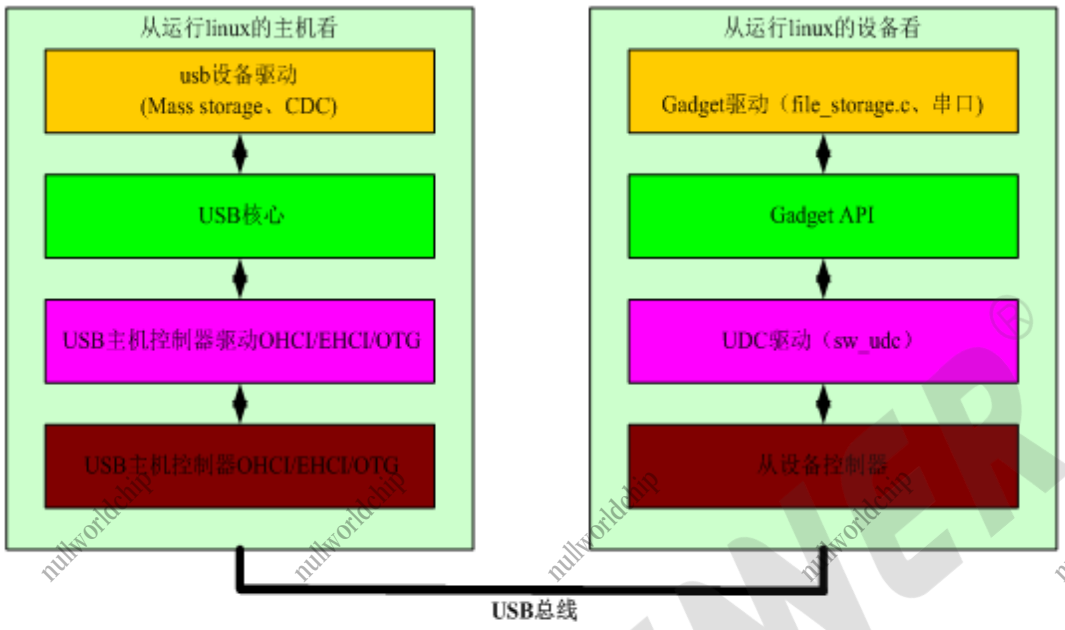


图 2: usb 驱动总体结构

USB 子系统主要任务包括：

a. 注册和管理设备驱动；

b. USB 设备寻找驱动，并初始化和配置设备；

c. 内核中表现设备的树形结构；

d. 与设备交互。

## 2.4 源码结构介绍

linux-4.9/driver/usb

```
├── core //usb核心代码
├── gadget //usb device功能配置代码，如adb、mtp或hid等
├── host
│   ├── ehci_sunxi.c //AW平台ehci控制器代码
│   ├── ohci_sunxi.c //AW平台ohci控制器代码
│   └── xhci_sunxi.c //AW平台xhci控制器代码
```

```
└── sunxi_usb
    ├── manager //AW平台otg、host与device功能自动切换管理代码
    └── udc //AW平台device功能配置代码
```

# 2.5 模块配置介绍

## 2.5.1 menuconfig 配置

在命令行中进入内核根目录，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：

首先，选择 Device Drivers 选项进入下一级配置，如下图所示：

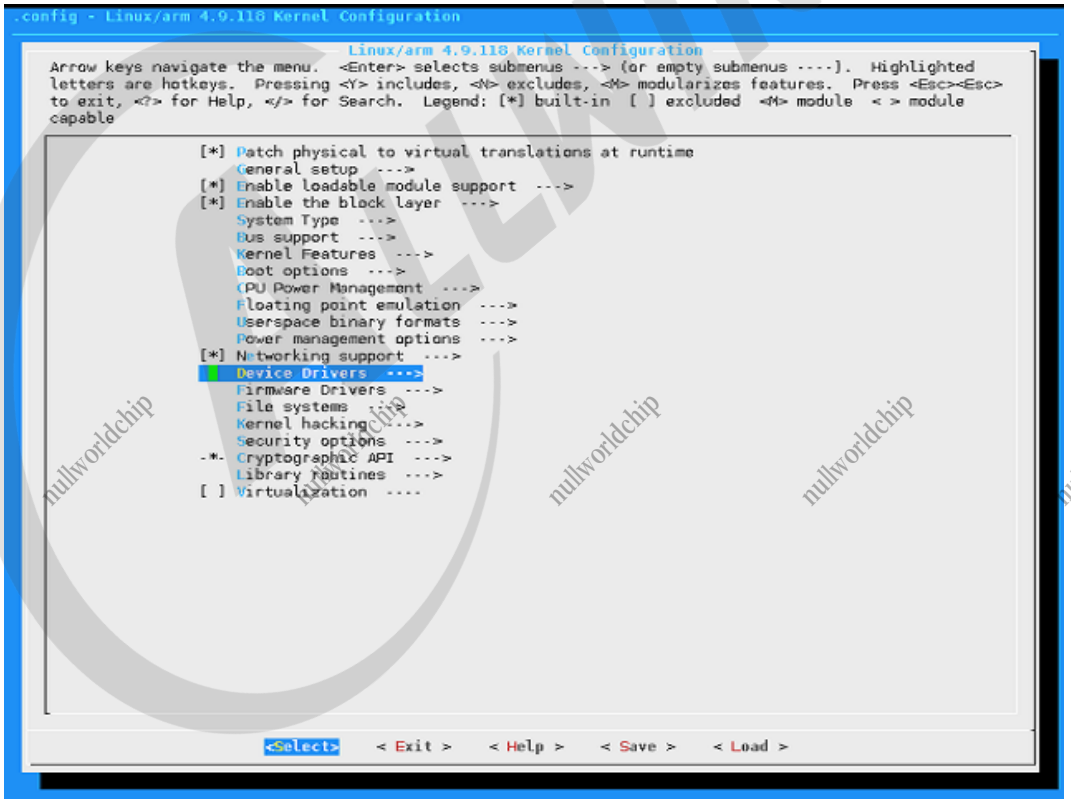


图 3: Device Drivers 选项配置

然后，选择 USB support 选项，进入下一级配置，如下图所示：

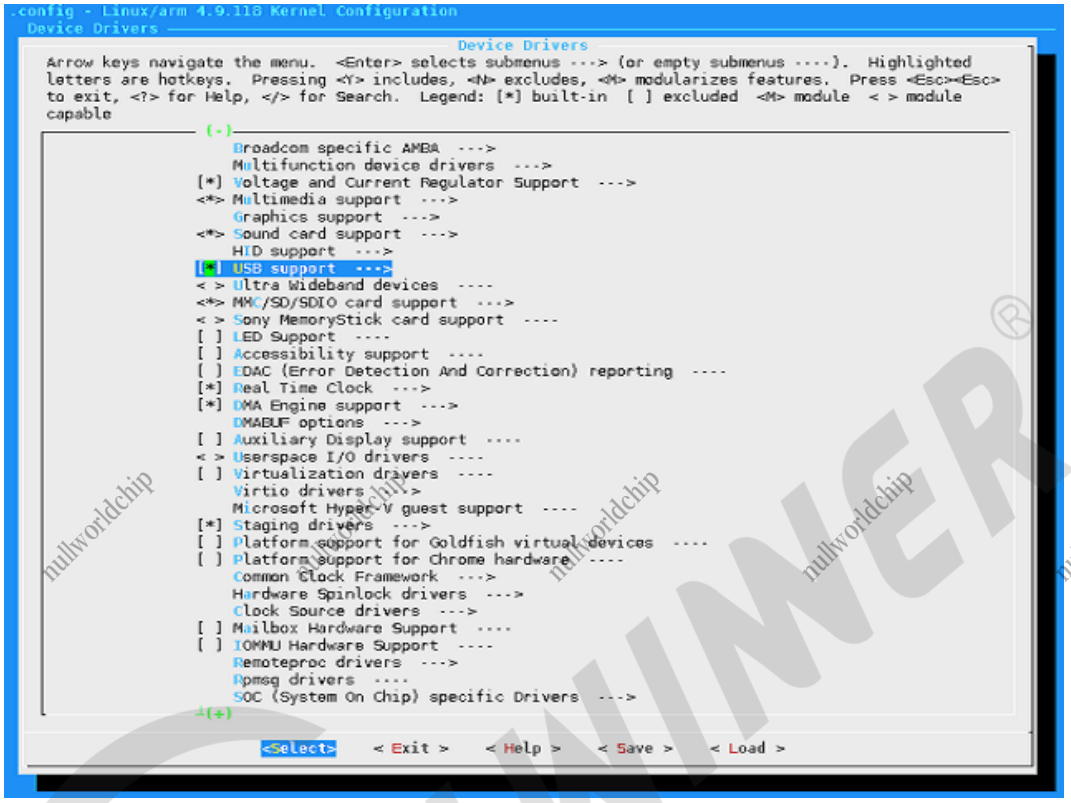


图 4: USB support 选项配置

打开如下两图的选项，如下图所示：

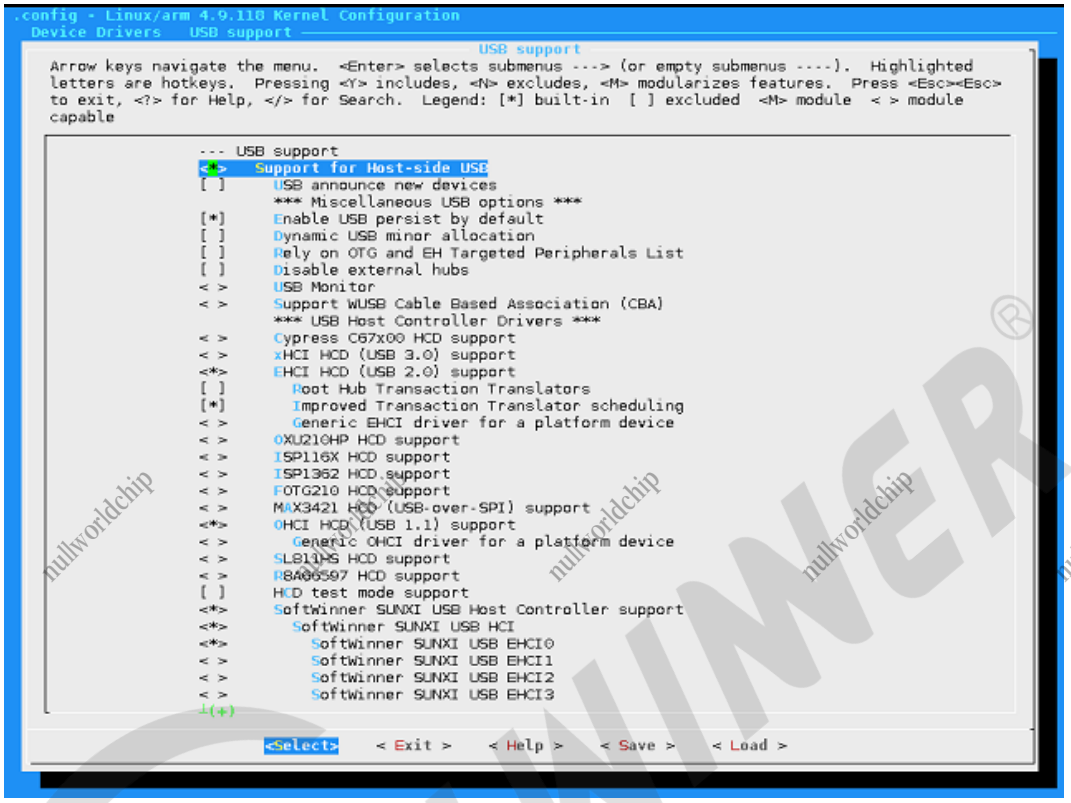


图 5: USB support 详细配置 1

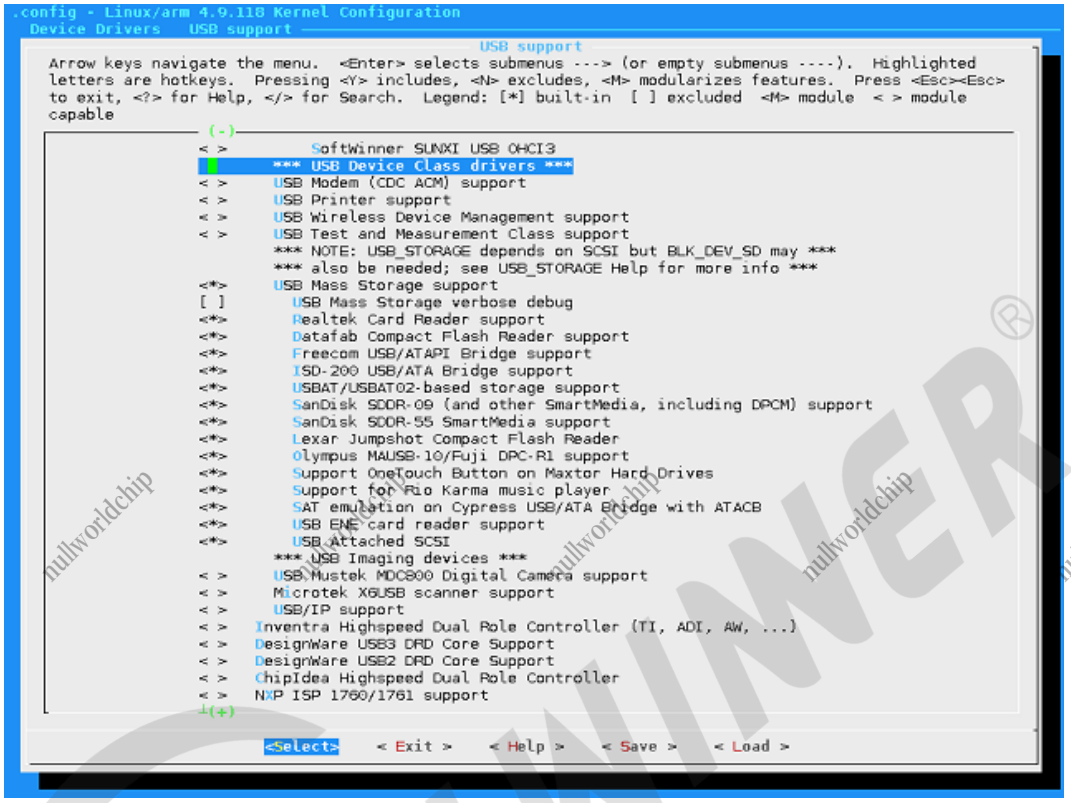


图 6: USB support 详细配置 2

然后，选择 USB Gadget Support，进入下一级配置，如下图所示:

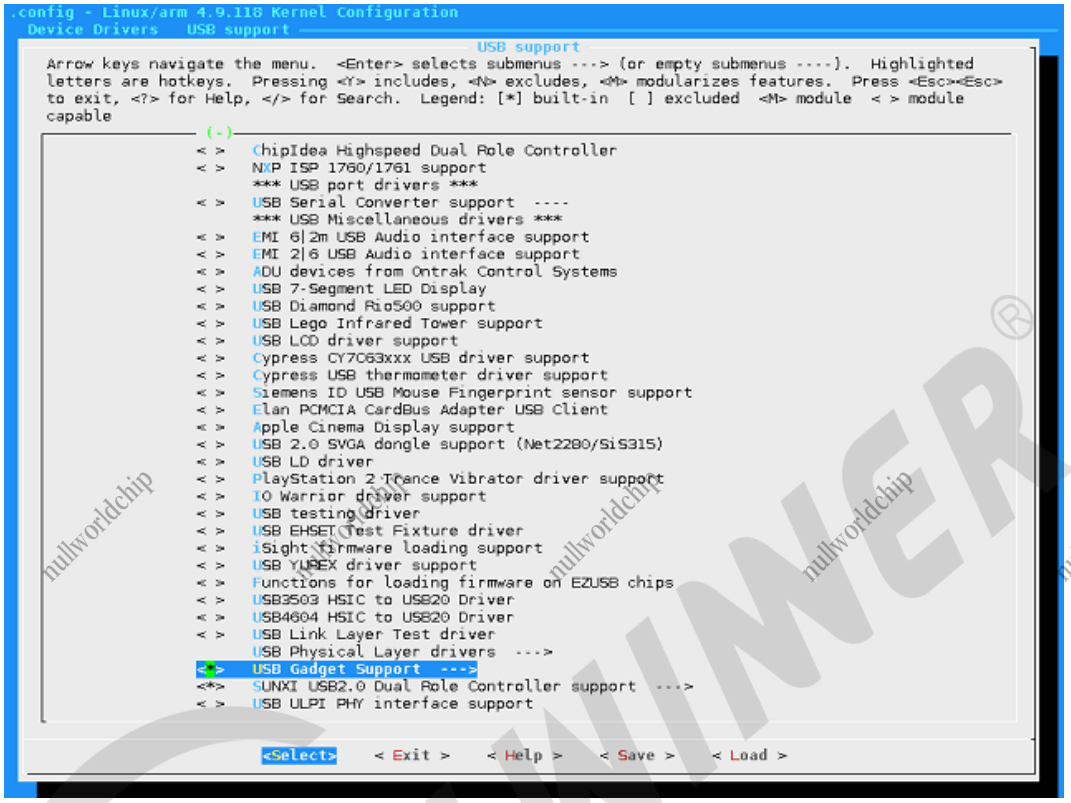


图 7: USB Gadget Suppor 选项配置

打开下图的选项，如下图所示：

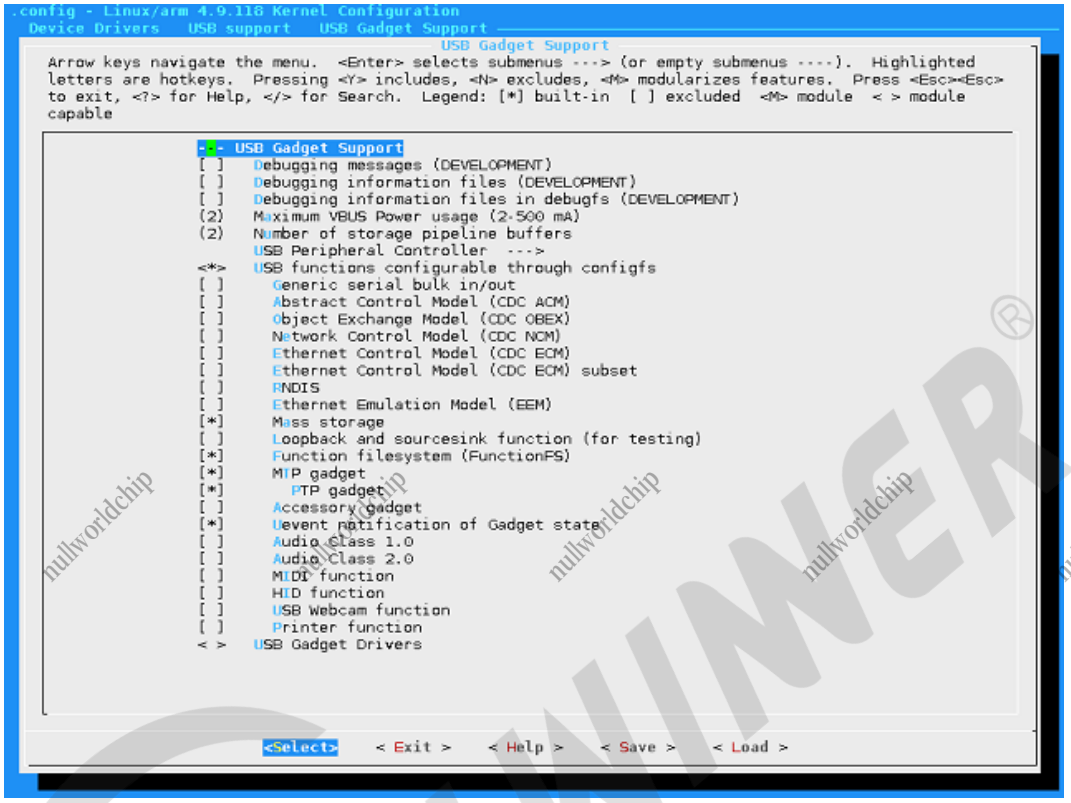


图 8: USB Gadget Suppor 详细配置

然后，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：

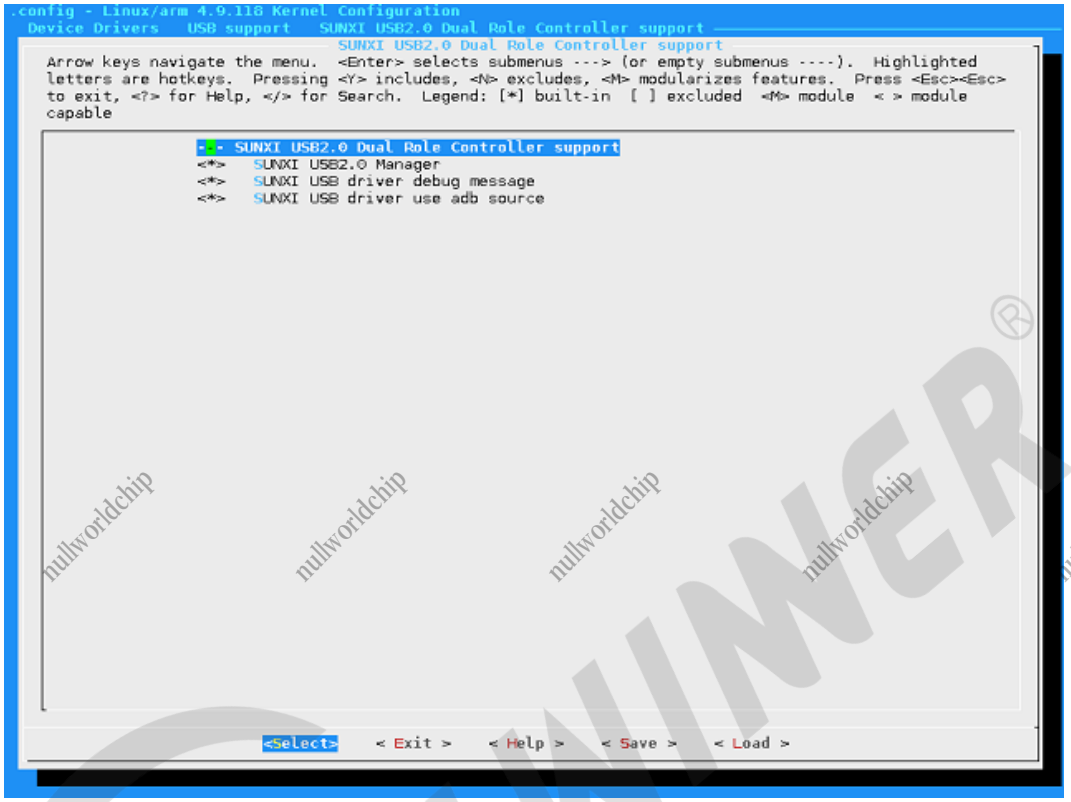


图 9: SUNXI USB2.0 Dual Role controller support 详细配置

## 2.5.2 Device Tree 配置

```
usbc0:usbc0@0 {
    device_type = "usbc0";
    compatible = "allwinner,sunxi-otg-manager";
    usb_port_type = <2>;
    usb_detect_type = <1>;
    usb_id_gpio;
    usb_det_vbus_gpio;
    usb_regulator_io = "nocare";
    usb_wakeup_suspend = <0>;
    usb_luns = <3>;
    usb_serial_unique = <0>;
    usb_serial_number = "20080411";
    rndis_wceis = <1>;
    status = "okay";
};
```

```
udc:udc-controller@0x05100000 {
  compatible = "allwinner,sunxi-udc";
  reg = <0x0 0x05100000 0x0 0x1000>, /*udc base*/
        <0x0 0x00000000 0x0 0x100>; /*sram base*/
  interrupts = <GIC_SPI 32 IRQ_TYPE_LEVEL_HIGH>;
  clocks = <&clk_usbphy0>, <&clk_usbotg>;
  status = "okay";
};

ehci0:ehci0-controller@0x05101000 {
  compatible = "allwinner,sunxi-ehci0";
  reg = <0x0 0x05101000 0x0 0xFFF>, /*hci0 base*/
        <0x0 0x00000000 0x0 0x100>, /*sram base*/
        <0x0 0x05100000 0x0 0x1000>; /*otg base*/
  interrupts = <GIC_SPI 30 IRQ_TYPE_LEVEL_HIGH>;
  clocks = <&clk_usbphy0>, <&clk_usbehci0>;
  hci_ctrl_no = <0>;
  status = "okay";
};

ohci0:ohci0-controller@0x05101400 {
  compatible = "allwinner,sunxi-ohci0";
  reg = <0x0 0x05101000 0x0 0xFFF>, /*hci0 base*/
        <0x0 0x00000000 0x0 0x100>, /*sram base*/
        <0x0 0x05100000 0x0 0x1000>; /*otg base*/
  interrupts = <GIC_SPI 31 IRQ_TYPE_LEVEL_HIGH>;
  clocks = <&clk_usbphy0>, <&clk_usbohci0>, <&clk_usbohci0_12m>, <&clk_osc48md4>, <&clk_hosc>, <&clk_losc>;
  hci_ctrl_no = <0>;
  status = "okay";
};
```

## 2.5.3 board.dts 配置

因为 usb0 otg device 控制器与标准 hci 复用为 usb0 的接口，所以 usbc0 为 hci 和 device 的总体。

```
/* --------------------------------
 *[usbc0]：控制器0的配置;
 *usb_used：USB使能标志。置1，表示系统中USB模块可用,置0,则表示系统USB禁用;
 *usb_port_type：USB端口的使用情况。 0：device only;1：host only;2：otg;
 *usb_detect_type：USB端口的检查方式。0：不做检测;1：vbus/id检查;2：id/dpdm检查;
 *usb_id_gpio：USB ID pin脚配置。具体请参考gpio配置说明;
 *usb_det_vbus_gpio: "axp_ctrl",表示axp提供;
 *usb_wakeup_suspend：0-SUPER_STANDBY, 1-USB_STANDBY.
 *--------------------------------
```

```
  *-------------------------------
  *---     USB0控制标志
  *-------------------------------
  */
 usbc0:usbc0@0 {
  device_type       = "usbc0";
  usb_port_type     = <0x2>
  usb_detect_type   = <0x1>
  usb_detect_mode   = <0x0>
  usb_id_gpio = <&pio PH 8 0 0 0xffffffff 0xffffffff>;
  usb_det_vbus_gpio  = "axp_ctrl";
  usb_regulator_io   = "nocare"
  usb_wakeup_suspend = <0x0>
  usb_serial_unique  = <0x0>
  usb_serial_number  = "20080411"
  rndis_wceis       = <1>;
  status            = "okay";
 }

 ehci0:ehci0-controller@0x05101000 {
    drvvbus-supply = <&reg_usb_vbus>;
 };

 ohci0:ohci0-controller@0x05101400 {
    drvvbus-supply = <&reg_usb_vbus>;
 };
```

1. usbc_used USB 使能标志。置 1，表示系统中 USB 模块可用, 置 0，则表示系统 USB 禁用
2. usb_port_type USB 端口的使用类型；0：device only；1：host only；2：OTG。
3. usb_detect_type USB 端口的检查方式。0：不做检测；1：vbus/id 检查；2：通过 id/dpdm 检测。
4. usb_id_gpio USB ID pin 脚配置。
5. usb_det_vbus_gpio USB DET_VBUS pin 脚配置。
6. usb_drv_vbus_gpio USB DRY_VBUS pin 脚配置。
7. usb_regualtor_io usb 控制器的供电域。
8. usb_wakeup_suspend 表示是否支持 usb 远程唤醒，1 为 usb standby，支持远程唤醒，0 为 super standby，不支持远程唤醒。A100 b3 场景配为 super standby，如果需要使用 usb standby，需配此处为 1，并且在 dts 的 usbc0/1 下加上 wakeup-source 节点，而由于 b3 版型上的 VCC-USB 与 VDD-USB 接的是 DCDC1 与 VDD-SYS，这 2 路电在 suspend 的时候会掉电，无法供控制器远程唤醒时工作，所以如果需要远程唤醒功能，还需改 USB 电路，将上述 2 路电换到 FCDO1 与 ALDO3。

# 3. FAQ

## 3.1 调试节点

不管 usb0 的 otg 配置怎么样，用户都可以通过操作 usb 提供的节点，进行动态切换角色，便于调试。切换角色：

```
device：cat sys/devices//platform/soc/usbc/usb_device
host： cat sys/devices/platform/soc/usbc/usb_host
```

另外可以通过 otg_role 查看当前的状态。

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

## 3.2 调试方法

USB 出现问题，一般可以按照以下步骤进行排查。

### 3.2.1 USB Host

1. 检查 VBUS 是否有电；
2. 无法连接设备的话，尝试更换同类型设备插入；
3. 保证硬件型信号完整性（USB 眼图良好）；
4. 对比正常情况与异常情况寄存器差异；
5. 根据出错 log 跟踪源码排查。

### 3.2.2 USB device

1. 确认当前模式是是 device 模式，按照上一节 ``调试方法" 的方法确认；

2. 确认相关功能的配置是否合理设置完成（adb、mtp 或 hid 等功能）。

## 3.3 常见问题

### 3.3.1 无法枚举 u 盘

#### 3.3.1.1 现象

必现问题。插入 u 盘，内核没有枚举到 u 盘。关键 log：

```
usb 2-1: new full-speed USB device number 2 using sunxi-ohci
usb 2-1: device descriptor read/64, error -62
usb 2-1: device descriptor read/64, error -62
usb 2-1: new full-speed USB device number 3 using sunxi-ohci
usb 2-1: device descriptor read/64, error -62
usb 2-1: device descriptor read/64, error -62
usb usb2-port1: attempt power cycle
usb 2-1: new full-speed USB device number 4 using sunxi-ohci
usb 2-1: device not accepting address 4, error -62
usb 2-1: new full-speed USB device number 5 using sunxi-ohci
usb 2-1: device not accepting address 5, error -62
usb usb2-port1: unable to enumerate USB device
```

#### 3.3.1.2 复现方法

见 ``现象" 小节。

#### 3.3.1.3 排查过程

1). 查看 drvvbus 电；

2). 测试 usb 眼图；

3). 与硬件同事沟通问题，得知 H616 的所有 USB 控制器共用 USB2 的 common 电路，但 USB2 未使能，其他 USB 就用不了。

#### 3.3.1.4 解决方法

打开 usb2。

#### 3.3.1.5 结论

硬件逻辑问题。

## 3.4 android 开发者选项中将 usb 切换为 ``file transfer" 模式，系统死机重启

### 3.4.1 现象

必现问题。android 开发者选项中将 usb 切换为 ``file transfer" 模式，系统死机重启。开机默认 adb +mtp 模式的话，不会出现此问题。关键 log：

```
Unable to handle kernel NULL pointer dereference at virtual address 00000000
pgd = 8525b926
[00000000] *pgd=00000000
Internal error: Oops: 5 [#1] PREEMPT SMP ARM
Modules linked in: xr829 gslX680new xradio_btfdi xradio_btlpm vin_v4l2 gc0310_mipi gc2355_mipi gc030a_mipi gc2385_mipi vin_io
    videobuf2_v4l2 videobuf2_dma_contig videobuf2_memops videobuf2_core mali(O)
```

```
CPU: 0 PID: 19305 Comm: MtpServer Tainted: G          O    4.9.170 #1
Hardware name: sun8iw15
task: a8fb7dcf task.stack: 952b9048
PC is at mtp_read+0xb8/0x2c0
LR is at preempt_count_add+0x10c/0x13c
pc : [<c06899fc>]    lr : [<c01552fc>]    psr: a0010093
sp : ea809e78  ip : ea809e48  fp : ea809ec4
r10: 00000003  r9 : 00000000  r8 : 00000000
r7 : 00000200  r6 : ec387070  r5 : c11226ff  r4 : ec387000
r3 : c1122cdc  r2 : 00000009  r1 : 00000000  r0 : 00000000
Flags: NzCv  IRQs off  FIQs on  Mode SVC_32  ISA ARM  Segment none
Control: 10c5387d  Table: 576ac06a  DAC: 00000051
Process MtpServer (pid: 19305, stack limit = 0x7ccec2e6)
Stack: (0xea809e78 to 0xea80a000)
9e60:                                     c100918c 945df800
9e80: 00000000 edb98000 c0179664 ea809e8c ea809e8c 000409aa 00000001 c0689944
9ea0: ed9e49c0 c100918c ea809f70 00000200 00000200 00000003 ea809f3c ea809ec8
9ec0: c02937c4 c0689950 00000200 00000003 ea809f04 ea809ee0 c042a1dc c02d8eac
9ee0: 00000000 00000000 ed9e49c0 00000000 00000200 00000000 ea809f3c ea809f08
9f00: c02944a4 c042a12c ea809f34 ea809f18 c02b37ec 000409aa 00000200 00000200
9f20: ed9e49c0 945df800 ea809f70 945df800 ea809f6c ea809f40 c02945c0 c0293788
9f40: c02b38e4 c02b3840 ea809f6c ed9e49c0 c100918c ed9e49c1 c100918c 945df800
9f60: ea809fa4 ea809f70 c0295690 c029452c 00000000 00000000 c010a5e8 000409aa
9f80: aae5d5a4 a59aaa06 00000001 00000003 c0107f44 ea808000 00000000 ea809fa8
9fa0: c0107f18 c029563c aae5d5a4 a59aaa06 0000002b 945df800 00000200 aaefc350
9fc0: aae5d5a4 a59aaa06 00000001 00000003 aae5d5d4 aae5d5bc 8425cac8 93aa3a80
9fe0: aae5d5a4 8425c998 93aad39d a83e111c 000b0010 0000002b 00000000 00000000
[<c06899fc>] (mtp_read) from [<c02937c4>] (__vfs_read+0x48/0x13c)
[<c02937c4>] (__vfs_read) from [<c02945c0>] (vfs_read+0xa0/0x154)
[<c02945c0>] (vfs_read) from [<c0295690>] (SyS_read+0x60/0xb0)
[<c0295690>] (SyS_read) from [<c0107f18>] (__sys_trace_return+0x0/0x10)
Code: ebebbe1b eaffffe4 ba00006b eaffffe2 (e5993000)
---[ end trace 87fde6c2b1eb4750 ]---
Kernel panic - not syncing: Fatal exception
CPU1: stopping
CPU: 1 PID: 0 Comm: swapper/1 Tainted: G      D    O    4.9.170 #1
Hardware name: sun8iw15
[<c0110a68>] (unwind_backtrace) from [<c010c5ac>] (show_stack+0x20/0x24)
[<c010c5ac>] (show_stack) from [<c04982b0>] (dump_stack+0x78/0x94)
[<c04982b0>] (dump_stack) from [<c010f1f4>] (handle_IPI+0x178/0x358)
[<c010f1f4>] (handle_IPI) from [<c0101590>] (gic_handle_irq+0x7c/0x84)
[<c0101590>] (gic_handle_irq) from [<c010d08c>] (__irq_svc+0x6c/0xa8)
Exception stack(0xef14bf18 to 0xef14bf60)
bf00:                                     00000000 c116bf78
bf20: 000409aa 000409aa ee4eb600 00000001 9129d73b 0000006d 00000001 ee6ba850
bf40: 00000050 ef14bfa4 ef14bf00 ef14bf68 c0a09fc4 c071f5ec 60030013 ffffffff
[<c010d08c>] (__irq_svc) from [<c071f5ec>] (cpuidle_enter_state+0x244/0x3a4)
[<c071f5ec>] (cpuidle_enter_state) from [<c071f79c>] (cpuidle_enter+0x24/0x28)
[<c071f79c>] (cpuidle_enter) from [<c0179cbc>] (call_cpuidle+0x44/0x48)
[<c0179cbc>] (call_cpuidle) from [<c0179fb8>] (cpu_startup_entry+0x164/0x1e0)
[<c0179fb8>] (cpu_startup_entry) from [<c010ee10>] (secondary_start_kernel+0x11c/0x13c)
[<c010ee10>] (secondary_start_kernel) from [<401019ec>] (0x401019ec)
```

```
Rebooting in 5 seconds..
```

### 3.4.2 复现方法

见``现象"小节。

### 3.4.3 排查过程

1).KASAN 工具排查内存溢出问题，发现问题的时候并没有内存溢出；

2). 排查 dailybuild 固件 change-list，看是否最近引入问题，此问题一直有；

3). 死啃代码，看为什么指针为出错是为空。

### 3.4.4 解决方法

修改代码逻辑，添加判断空指针跑其他分支。

### 3.4.5 结论

空指针引用。

# 4. Declaration

This document is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner. The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This document neither states nor implies warranty of any kind, including fitness for any particular application.tates nor implies warranty of any kind, including fitness for any particular application.