



NT98530 IPC

SDK 快速入门指南

修订版本

版本	日期	修改
1.0	2021/7/01	初版

目录

NT98530 IPC SDK 快速入门指南	1
修订版本	2
目录	3
1 概述	6
2 搭建开发环境	7
2.1 概述	7
2.2 部署 Linux 开发环境	7
2.3 首次安装 SDK 开发包	8
2.3.1 展开 linux sdk 源码开发包	9
2.3.2 安装交叉编译工具链	9
2.4 Demo 单板	10
2.4.1 外部接口说明	10
2.4.2 启动源设定	12
2.4.3 UART 调试口	12
2.4.4 Demo 单板上电	13
2.4.5 Sensor 连接方式	13
3 编译 SDK	14

3.1	SDK 编译指令	14
3.2	编译 SDK 镜像文件	16
3.3	编译 Boot loader	19
3.3.1	展开 Boot loader 源码开发包	19
3.3.2	安装 Loader 交叉编译工具链	19
3.3.3	编译 Boot loader	20
4	固件烧录	21
4.1	裸板烧录	21
4.1.1	SD 卡	22
4.1.2	Ethernet	22
4.1.3	USB	23
4.1.4	Uart	23
4.2	非裸板烧录	23
4.2.1	SD 卡	23
4.2.2	Uboot shell	23
5	开机启动流程	24
5.1	启动流程概述	24
5.2	启动源配置	25
5.3	启动流程各部分	25
5.3.1	Rom code	25
5.3.2	Boot Loader	25
5.3.3	Uboot	25

5.3.4	Kernel	26
5.3.5	Rootfs	26
5.3.6	启动脚本	26
5.4	开机启动日志	26
6	运行 hdal 媒体业务	43
6.1	RTSP server	43
6.2	UVC CAM	46
7	常见问题解答	49
7.1	编译错误：程序中有游离的'#' -e #define NVT_UTS_RELEASE "4.19.148_"	49
7.2	!!! Stop booting because your power_config dtsi and bootstrap do not match !!!	50

1 概述

本文档主要介绍 NT98530 IPC SDK 的编译环境安装，SDK 编译，固件烧录，设备启动流程，以及执行示例代码，方便使用者能够快速搭建 NT98530 IPC SDK 开发环境，进行产品软件开发。

SDK 组件	版本
Toolchain	linux/u-boot: <ul style="list-style-type: none">● glibc: aarch64-ca53-linux-gnueabi-hf-8.4.01 loader: <ul style="list-style-type: none">● gcc-linaro-6.4.1-2018.05-x86_64_arm-eabi rtos: <ul style="list-style-type: none">● aarch64-ca53-elf-newlibc-8.5
Linux kernel	4.19.148
u-boot	2019.04
busybox	1.31.1

2 搭建开发环境

2.1 概述

NT98530 IPC SDK 主要使用 64bits Ubuntu 操作系统作为开发环境，可以是基于 Server 或 VirtualBox 上运行 64bits Ubuntu OS。另外，开发环境还需要安装一些必要的 Linux 开发套件，以及安装相应的 toolchain。

请参考 *NT98530_SDK_Pure_Linux_Programming_Guide_en.pdf* 文档说明。

2.2 部署 Linux 开发环境

Ubuntu OS 镜像文件 16.04/18.04/20.04 Desktop AMD64 version ISO 可以从 Ubuntu 官网 (<https://releases.ubuntu.com/>) 上下载。

安装好 Ubuntu 后，还需要安装一些必要的开发套件。如以下 Ubuntu 版本：

Ubuntu 16.04:

```
$ sudo apt-get install build-essential libc6-dev lib32ncurses5 libncurses5-dev libncurses5:i386  
libgl1-mesa-dev g++-multilib mingw-w64 tofrodos lib32z1 u-boot-tools zlib1g-dev bison  
libbison-dev flex mtd-utils vim squashfs-tools gawk cmake cmake-data liblz4-tool libmpc3  
libstdc++6 device-tree-compiler android-tools-fsutils texinfo libssl-dev lzop
```

Ubuntu 18.04:

```
$ sudo apt-get install build-essential libc6-dev lib32ncurses5 libncurses5-dev libncurses5:i386  
libgl1-mesa-dev g++-multilib mingw-w64 tofrodos lib32z1 u-boot-tools zlib1g-dev bison  
libbison-dev flex mtd-utils vim squashfs-tools gawk cmake cmake-data liblz4-tool libmpc3  
libstdc++6 device-tree-compiler android-tools-fsutils texinfo libssl-dev lzop
```

Ubuntu 20.04:

```
$ sudo apt-get install build-essential libc6-dev libncurses5-dev libncurses5:i386 libgl1-mesa-dev  
g++-multilib mingw-w64 tofrodos lib32z1 u-boot-tools zlib1g-dev bison libbison-dev flex mtd-utils  
vim squashfs-tools gawk cmake cmake-data liblz4-tool libmpc3 libstdc++6 device-tree-compiler  
android-sdk-libsparse-utils android-sdk-ext4-utils texinfo libssl-dev lzop
```

Ubuntu 默认使用的 shell 是 dash, SDK 开发环境需要使用 bash 作为默认的 shell，可通过如下命令修改：

```
$ sudo dpkg-reconfigure dash, 然后在弹出的窗口上选择“否”
```

可以通过 `ls -la /bin/sh` 命令来确认当前使用的 shell:

```
$ ls -la /bin/sh
```

```
/bin/sh -> bash
```

2.3 首次安装 SDK 开发包

在 Linux 环境里使用以下指令解压缩 SDK 开发包:

```
$ tar -jxvf nt98530_linux_sdk_{version}.tar.bz2
```

SDK 包解压缩后将生成如下目录结构树:

```
nt98530_linux_sdk_{version}
```

```
├── NT98530_Document_Tool
│   ├── hardware          硬件相关文档
│   │   ├── board
│   │   │   ├── document  Demo 单板 PCB/原理图及硬件应用指南等文档
│   │   │   └── chip
│   │   │       ├── document  芯片手册
│   │   └── only for reference  供开发参考的文档
│   ├── software          SDK 软件相关文档
│   │   ├── board        板端相关的 SDK 源码及文档
│   │   │   ├── document  OSDDRV, MEDIA, IQ, CVAI 等相关使用文档
│   │   └── pc            PC 端工具及文档说明
│   │       ├── document  固件烧录和 IQ 调试等的 PC 端工具使用说明文档
│   │       └── tool       固件烧录和 IQ 调试等的 PC 端工具
│   └── Disclaimer.pdf     SDK 免责声明文件
├── NT98530_SDK_Package    SDK 压缩包
│   ├── nt98530_linux_sdk_release_glibc_{version}.tar.bz2
│   └── nt98530_loader_release_{version}.tar.bz2
├── NT98530_Toolchain
│   ├── linux_glibc        linux/uboot glibc 编译工具链
│   │   ├── aarch64-ca53-linux-gnueabi-hf-8.4.01.tar.bz2
│   ├── loader            boot loader 编译工具链
│   │   ├── gcc-linaro-6.4.1-2018.05-x86_64_arm-eabi.tar.bz2
│   ├── rtos              rtos 编译工具链
│   │   └── aarch64-ca53-elf-newlib-8.5.tar.bz2
```


2.3.1 展开 linuxsdk 源码开发包

在 Linux 环境里使用以下指令展开 SDK 源码开发包：

```
$ tar -jxvf ${soc_name}_linux_sdk_{version}.tar.bz2
```

Linux sdk 源码展开后，目录树结构如下：

```
├─${SOC_NAME}_linux_sdk
│   ├──BSP                包含 linux, busybox, uboot 和 rootfs 的源码
│   ├──build              编译环境相关的脚本
│   ├──code               linux driver,hdal driver, lib,以及示例代码。
│   ├──configs            sdk model 设定相关设备树配置文件
│   ├──rtos               rtos sdk package (for fastboot : rtos boot linux)
│   ├──tools              linux 开源工具
│   └─Makefile            顶层 Makefile 文件
```

2.3.2 安装交叉编译工具链

可选择交叉编译工具链如下：

```
glibc: aarch64-ca53-linux-gnueabi-hf-8.4
```

请在 Linux 环境中用以下命令安装交叉编译工具链：

```
$ sudo tar -jxvf aarch64-ca53-linux-gnueabi-hf-8.4.01.tar.bz2 -C /opt/
```

如果要改变工具链默认安装路径，请将 build/envconfig.sh 里变量 CROSS_TOOLCHAIN_DIR 修改为自己对应的安装路径。

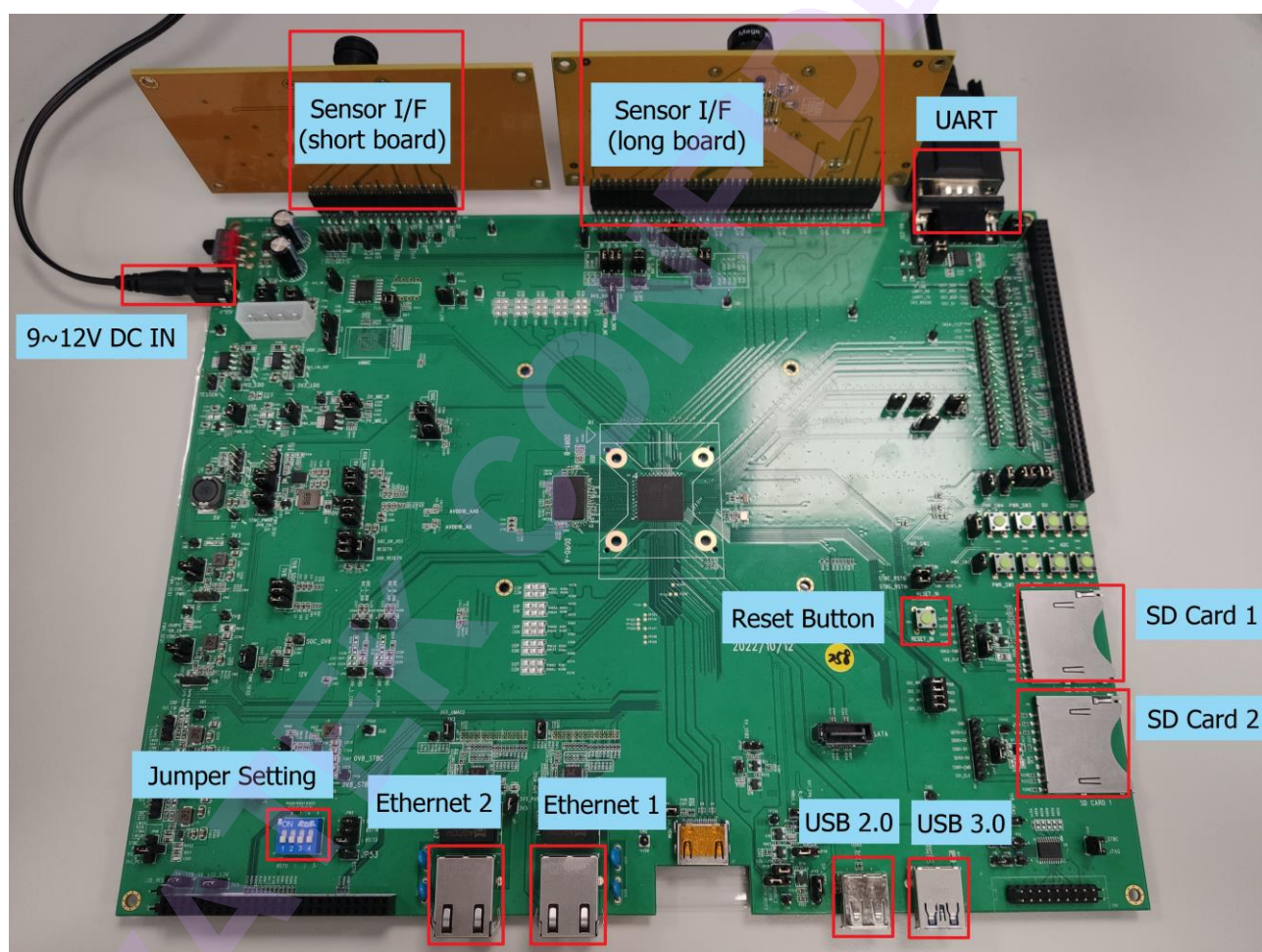
```
function setpath_toolchain_config()
{
    .....
    if [ -d /opt/ivot/$NVT_CROSS ]; then
        export CROSS_TOOLCHAIN_DIR=/opt/ivot
    else
        export CROSS_TOOLCHAIN_DIR=/opt/
    fi
    .....
}
```

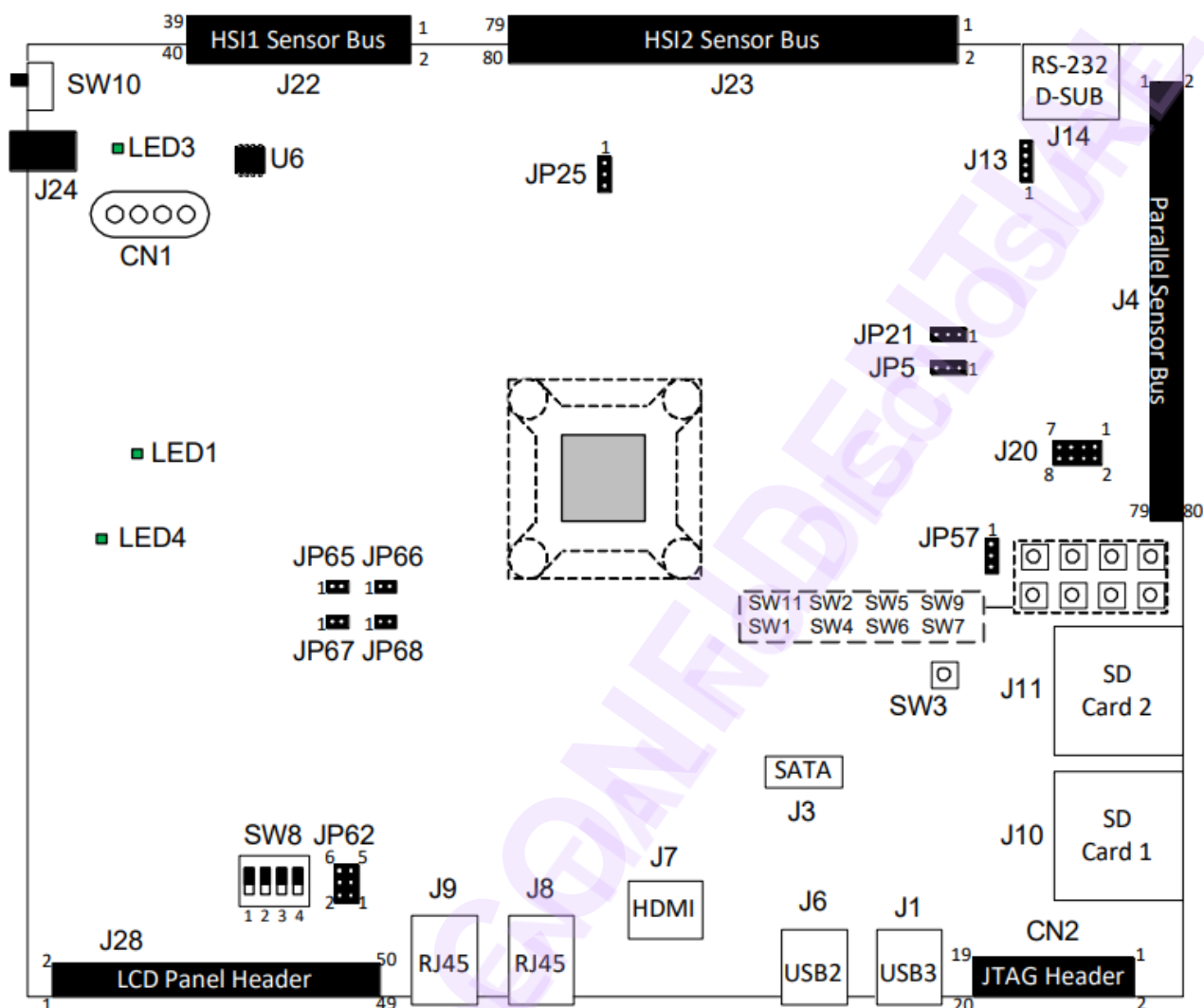
2.4 Demo 单板

NT98530 Demo 单板详细说明请参考 NT98530_EVB_User Guide.pdf。

- 默认配置 1GB+1GB 容量的 LPDDR4 SDRAM
- 默认配置 128MBytes SPI NAND Flash Memory
- 支持 HSI1, HSI2, 以及 Parallel Sensor 总线接口
- 预留 EMMC 接口
- 可配置启动源

2.4.1 外部接口说明






Part Reference	Descriptions
J24	DC-in power jack.
SW10	DC power switch.
SW8	DIP switch of boot-up device options.
SW1, SW2, SW4, SW11	System power-on tact switches.
SW3	Hardware reset switch.
J22	HSI1 sensor connector.
J23	HSI2 sensor connector
J14	RS-232 debug port for console.
J13	UART 4-pin header.

J4	Parallel sensor connector.
J10, J11	SD card slots
U6	SPI NAND flash.
J1	USB 3.1 gen1 connector.
J6	USB 2.0 connector
J7	HDMI 2.0 connector
J8, J9	Ethernet RJ45 connectors.
J28	LCD panel connector
J65, J66	MIC-in headers
J67, J68	Line-out audio headers.
SW5, SW6, SW7, SW9	ADC tact switches.

2.4.2 启动源设定

Demo 单板默认存储介质为 SPI Nand flash，不同介质启动模式配置参考如下：

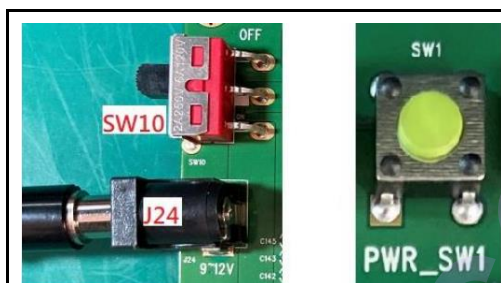
BST	[3]	[2]	[1]	[0]	Description BS[3:0]
	0	0	0	0	0x0: SPI_NOR
	0	0	0	1	0x1: SDIO1
	0	0	1	0	0x2: SPI Nand with On die ECC(2K page)
	1	0	0	0	0x8: EMMC 4 bits data bus
	1	0	0	1	0x9: EMMC 8 bits data bus
	0	1	0	0	0x4: Ethernet(ETH1)
	0	1	0	1	0x5: USB3 high speed
	1	1	0	0	0xC: Uart1
	-	-	-	-	Others

2.4.3 UART 调试口

NT98530 demo 单板保留 9pin 的 RS232 (J13) 和 4pin 的 uart (J14) 作为串口调试用。
串口调试波特率为 115200。



2.4.4 Demo 单板上电

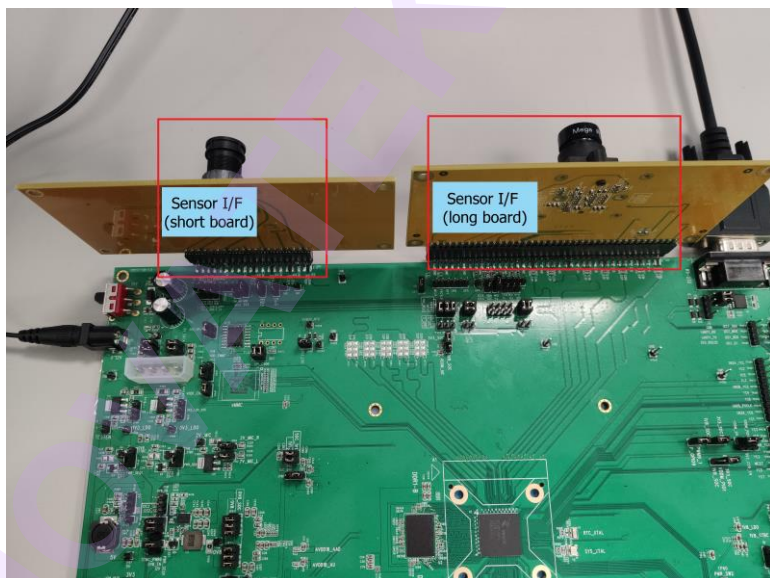


Demo 单板上电流程：

1. 插入 12V DC 电源到 J24。
 2. 拨动 SW10 开关，打开 DC 供电。
 3. 轻触 SW1，即可触发上电。
- 若有打开调试串口，即可看到上电启动信息。

2.4.5 Sensor 连接方式

将镜头朝向板子外侧后与板子连接。



3 编译 SDK

3.1 SDK 编译指令

NT98530 SDK 目录下的顶层 *Makefile*, 支援多个编译指令, 如“*make linux*” 可用于编译 linux 内核, “*make uboot*” 用于编译 u-boot...

为避免产生一些链接错误, 请通过顶层 *Makefile* 来编译 SDK。在使用这些命令前, 需要先初始化编译环境:

```
$ source build/envsetup.sh
```

编译指令	描述
<i>lunch</i>	选择或查看当前 SDK 的配置 model
<i>get_stuff_for_environment</i>	查看当前 SDK 编译环境
<i>make evb</i>	使用 NT98530 Demo board 需对 CSI 作 P/N swap 操作
<i>make clean</i>	清除当前 SDK 编译信息
<i>make all</i>	编译 sdk all-in-one 固件
<i>make help</i>	获取顶层 Makefile 可支持的编译指令

可使用“*make help*”指令获取其他可支持的指令:

```
$ make help
```

```
make help          -> show make command info
make all           -> build all
make linux         -> build linux-kernel
make linuxram      -> build linux-kernel with ramdisk support
make modules       -> build built-in kernel modules
make driver        -> build NVT linux driver modules
make atf           -> build ARM trusted firmware
make uboot         -> build loader(uboot)
make optee_os      -> build OPTEE kernel
make optee_client  -> build OPTEE client
make library       -> build library
make busybox       -> build busybox
make rootfs        -> build rootfs
```

make app -> build applications
make tools -> build tools
make sample -> build sample code
make post -> run postprocessing script
make pack -> Generate nvtpack image and preburn images

=====

make linux_config -> config linux-kernel
make linux_config_gcov -> modify kernel config for code coverage tool
make uboot_config -> config uboot
make busybox_config -> config busybox
make linux_header -> generate linux-kernel out of tree headers

=====

make clean -> clean all
make linux_clean -> clean linux-kernel & built-in kernel modules
make driver_clean -> clean NVT linux driver modules
make atf_clean -> clean ARM trusted firmware
make uboot_clean -> clean loader(uboot)
make optee_os_clean -> clean optee kernel
make optee_client_clean -> clean optee client application
make library_clean -> clean library
make busybox_clean -> clean busybox
make rootfs_clean -> clean rootfs
make app_clean -> clean applications
make tools_clean -> clean tools
make sample_clean -> clean sample code
make post_clean -> run postprocessing clean script
make pack_clean -> Remove nvtpack image

=====

3.2 编译 SDK 镜像文件

1. 进入 SDK 顶层目录

```
$ cd na51102_linux_sdk
```

2. 初始化编译环境

```
$ source build/envsetup.sh
```

3. model 选择（首次编译 sdk 或执行 make clean 指令后）：

```
$ lunch
```

Step 1: 选择 Project config

```
Loading default...
You're building on Linux
Lunch menu... select your Project config:
    1. Linux
    2. rtos
Which would you like? [Linux]
```

默认选择 Linux，可直接回车或输入 1 再回车。

Step 2: 选择 Model config

```
Lunch menu... select your Model config:
    1. cfg_530_IPCAM1_EMMC_EVB
    2. cfg_530_IPCAM1_EMMC_RAMDISK_EVB
    3. cfg_530_IPCAM1_EVB
    4. cfg_530_IPCAM1_NAND_RAMDISK_EVB
    5. cfg_530_IPCAM1_NAND_SQUASHFS_EVB
    6. cfg_530_IPCAM1_NOR_EVB
    7. cfg_530_IPCAM1_SECBOOT_EVB
    8. cfg_530_IPCAM1_TEE_EVB
    9. cfg_530_IPCAM1_TEE_SECBOOT_EVB
Which would you like? [cfg_530_IPCAM1_EVB]
```

根据单板配置及应用场景，选择相应的 Model 配置。

说明：NT98530 Demo 单板，存储介质默认是 SPI NAND，可选择 cfg_530_IPCAM1_EVB 或

cfg_530_IPCAM1_NAND_RAMDISK_EVB。

Model Config	描述
cfg_530_IPCAM1_EMMC_EVB	芯片:NT98530 存储介质: EMMC rootfs: EXT4
cfg_530_IPCAM1_EMMC_RAMDISK_EVB	芯片:NT98530 存储介质: EMMC rootfs: RAMDISK
cfg_530_IPCAM1_EVB	芯片:NT98530 存储介质: EMMC rootfs: UBI
cfg_530_IPCAM1_NAND_RAMDISK_EVB	芯片:NT98530 存储介质: SPI NAND rootfs: RAMDISK
cfg_530_IPCAM1_NAND_SQUASHFS_EVB	芯片:NT98530 存储介质: SPI NAND rootfs: SQUASH
cfg_530_IPCAM1_NOR_EVB	芯片:NT98530 存储介质: SPI NAND rootfs: SQUASH
cfg_530_IPCAM1_SECBOOT_EVB	芯片:NT98530 存储介质: SPI NAND rootfs: UBI 应用: SECBOOT
cfg_530_IPCAM1_TEE_EVB	芯片:NT98530 存储介质: SPI NAND rootfs: UBI 应用: TEE
cfg_530_IPCAM1_TEE_SECBOOT_EVB	芯片:NT98530 存储介质: SPI NAND rootfs: UBI 应用: TEE+ SECBOOT

Step 3: 选择 toolchain

Lunch menu... select your toolchain:

1. aarch64-ca53-linux-gnueabi-hf-8.4.01

```
Which toolchain would you like? [aarch64-ca53-linux-gnueabi-hf-8.4.01]
```

Toolchain 默认只有 glibc，可直接回车或选择 1。

4. 针对 NT98530 Demo 板使用，非 NT98530 Demo 板可跳过这一步骤

\$ make evb

说明：如果使用的是 530 demo 板，由于硬件设计 J22 CSI0 1C4D circuit 有做 PN SWAP，需要执行 “make evb”，会自动在 BSP/root-fs/rootfs/etc_Model/etc_xxx/init.d/S99_Sysctl 打开 csi phy p/n swap 命令：echo "phy_pn_swap" > /proc/nvt_ctl_sie/cmd

5. 编译整个系统：

\$ make all

执行“make all”编译完 SDK 后，将会在“output”目录下产生如下镜像文件：

```
├─ na51102_linux_sdk/..... 存放解压缩后的源代码和镜像文件
  │
  │   └─ logs/..... 编译过程产生的日志文件
  │   └─ output/..... 编译输出的镜像文件目录
  │       │
  │       │   └─ packed/
  │       │       │
  │       │       │   └─ FW98530A.bin..... ntpack 工具产生的 All-in-one 镜像文件
  │       │       │
  │       │       └─ raw/ ..... Raw 格式的输出镜像文件
  │       │
  │       └─ Image.bin..... Linux 内核镜像文件
  │       └─ u-boot.bin..... uboot 镜像文件(nvt checksum)
  │       └─ u-boot.lz.bin..... uboot 镜像文件(LZ 压缩格式)
  │       └─ u-boot.lzma.bin..... uboot 镜像文件(LZMA 压缩格式)
  │       └─ rootfs.ubifs.bin..... ubifs 镜像文件 (rootfs)
  │       └─ rootfs_1.rw.ubifs.bin..... 可读写文件系统
  │       └─ appfs.ubifs.nand.bin..... app ubifs 镜像文件
  │       └─ nvt-evb.bin..... linux 内核设备树 bin 档
  │       └─ atf.bin..... arm 可信任固件
  │       └─ FW98530A.ini..... the layout to pack to FW98530A.bin
```

3.3 编译 Boot loader

boot loader 配置编译请参考文档说明：

NT98530_Document_Tool\software\board\document\OSDRV\NT98530_Boot_Loader_User_Guide_en.pdf

3.3.1 展开 Bootloader 源码开发包

NT98530 Boot loader 源码包路径：

nt98530_linux_sdk_{version}\NT98530_SDK_Package\ nt98530_loader_release_{version}.tar.bz2

在 Linux 环境里使用以下指令展开 Boot loader 源码开发包：

```
$ tar -jxvf ${soc_name}_loader_release_{version}.tar.bz2
```

Boot loader 源码展开后，目录树结构如下：

```
├─${soc_name}_loader_release_{version}
│   ├──ARC                Library files
│   ├──Include            Loader library/driver header file
│   ├──LibExt             Control flow files
│   ├──MakeCommon        Makefiles for building loader
│   ├──Project            Source code to build project dependent loader
│   └──Tools              Executable files/ddr param ini required for building loader
```

3.3.2 安装 Loader 交叉编译工具链

Boot loader 交叉编译工具链包: gcc-linaro-6.4.1-2018.05-x86_64_arm-eabi.tar.bz2

修改交叉编译工具链默认安装路径

loader\MakeCommon\MakeOption.txt

```
CROSS_COMPILE := /opt/arm/gcc-linaro-6.4.1-2018.05-x86_64_arm-eabi/bin/arm-eabi-
```

请在 Linux 环境中用以下命令安装交叉编译工具链：

```
$ sudo tar -jxvf gcc-linaro-6.4.1-2018.05-x86_64_arm-eabi.tar.bz2 -C /opt/arm/
```

3.3.3 编译 Bootloader

- 进入编译目录

```
$ cd MakeCommon/
```

- 清除编译信息

```
$ make clean
```

- 编译 loader

```
$ make release
```

编译成果物存放在：

```
loader\Project\Model\Loader530_Data\Release\LD98530A.bin
```

说明：

当 STORAGEEXT=Eth 时，编译成果物为 loader.bin

4 固件烧录

NT98530 SDK 支持多种固件烧录方式。

烧录模式	烧录阶段	烧录方式	应用场景
裸板烧录	ROM code	SD 卡	<ul style="list-style-type: none"> ● 可配置 boot from sd ● 支持 sd/tf 卡
		Ethernet (MAC0)	<ul style="list-style-type: none"> ● 可配置 boot from ethernet(mac0) ● 支持 ethernet 接口
		USB (USB3.0 Port)	<ul style="list-style-type: none"> ● 可配置 boot from usb(mac0) ● 支持 USB3.0 接口
		UART(UART0)	<ul style="list-style-type: none"> ● 可配置 boot from uart ● 支持 uart0 调试串口
非裸板烧录	Loader	SD 卡	<ul style="list-style-type: none"> ● Loader 正常启动, 且能读 sd 卡 ● 支持 sd/tf 卡
	Uboot shell	nand/flash/mmc cmd	<ul style="list-style-type: none"> ● Uboot 正常启动
		nvt_update_all cmd	<ul style="list-style-type: none"> ● 支持 ethernet(tftp), u 盘或 sd 卡
		DFU	<ul style="list-style-type: none"> ● Uboot 正常启动 ● 支持 USB 接口
	Linux shell	Linux shell cmd	<ul style="list-style-type: none"> ● Linux shell 正常启动

当单板还未烧录固件, 或无法正常启动到 boot loader 阶段时, 需要选择在 rom code 阶段烧录固件。

rom code 阶段加载 boot loader 异常时会自动切换启动源: (SPI-NOR/SPI-NAND/eMMC) → SD card → Ethernet → USB → UART

4.1 裸板烧录

4.1.1 SD 卡

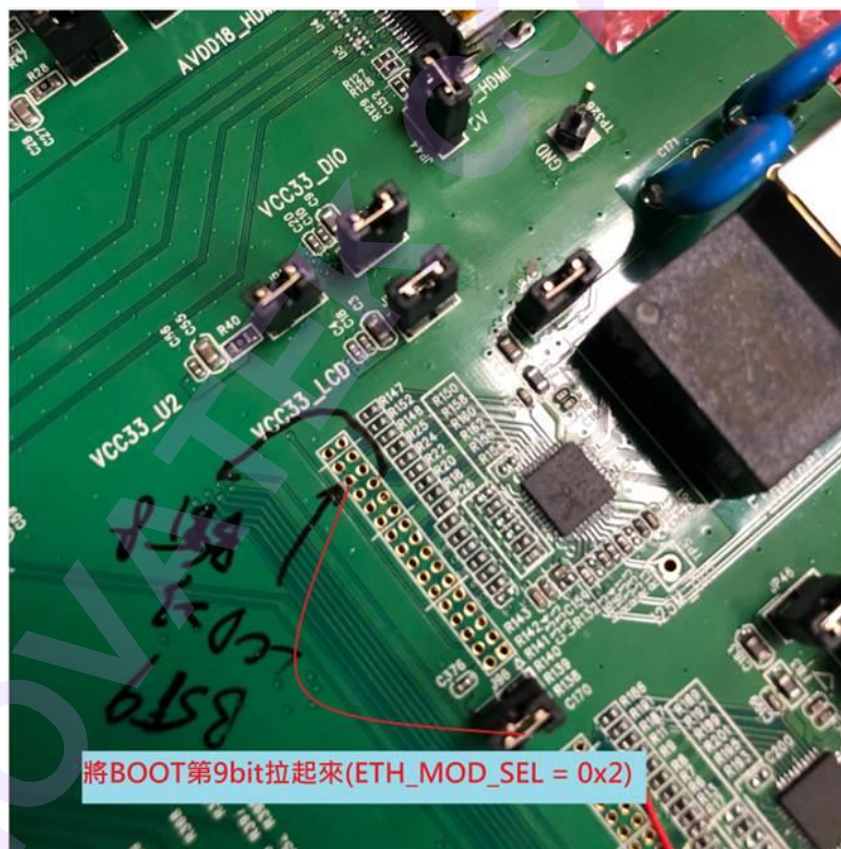
- 预先准备 bin 档
 - Boot loader: 如 LD98530A.bin
 - All-in-one FW: 如 FW98530A.bin
- 裸板使用 SD 卡烧写固件步骤：
 - 配置为 boot from sdio 或是可以通过自动切换启动源方式启动到 sdio
 - 格式化 SD 卡为 FAT32 文件格式
 - 先放入 boot loader bin, 如 LD98530A.bin (确保 loader bin 在 sd 卡第一扇区)
 - 再放入 all-in-one FW, 如 FW98530A.bin
 - 插卡上电开机, 即自动烧写

4.1.2 Ethernet

请参考 NT98530_Ethernet_Download_User_Guide_en.pdf

说明：NT98530 demo board 配置 boot from ethernet 方法：

- Bootstrap 要切成 0x4
- 跳線將 boot 第 9bit 設成 high (如下圖紅色接法)



- 其他步骤请按照 NT98530_Ethernet_Download_User_Guide_en.pdf 设定
- 上电开机，看到串口输出“NHG”，说明已配置 boot from Ethernet(RGMII)
- 再移除跳线，即可进行固件下载烧录。

4.1.3 USB

请参考 NT98530_Ethernet_Download_User_Guide_en.pdf

4.1.4 Uart

请参考 NT98530_Ethernet_Download_User_Guide_en.pdf

4.2 非裸板烧录

4.2.1 SD 卡

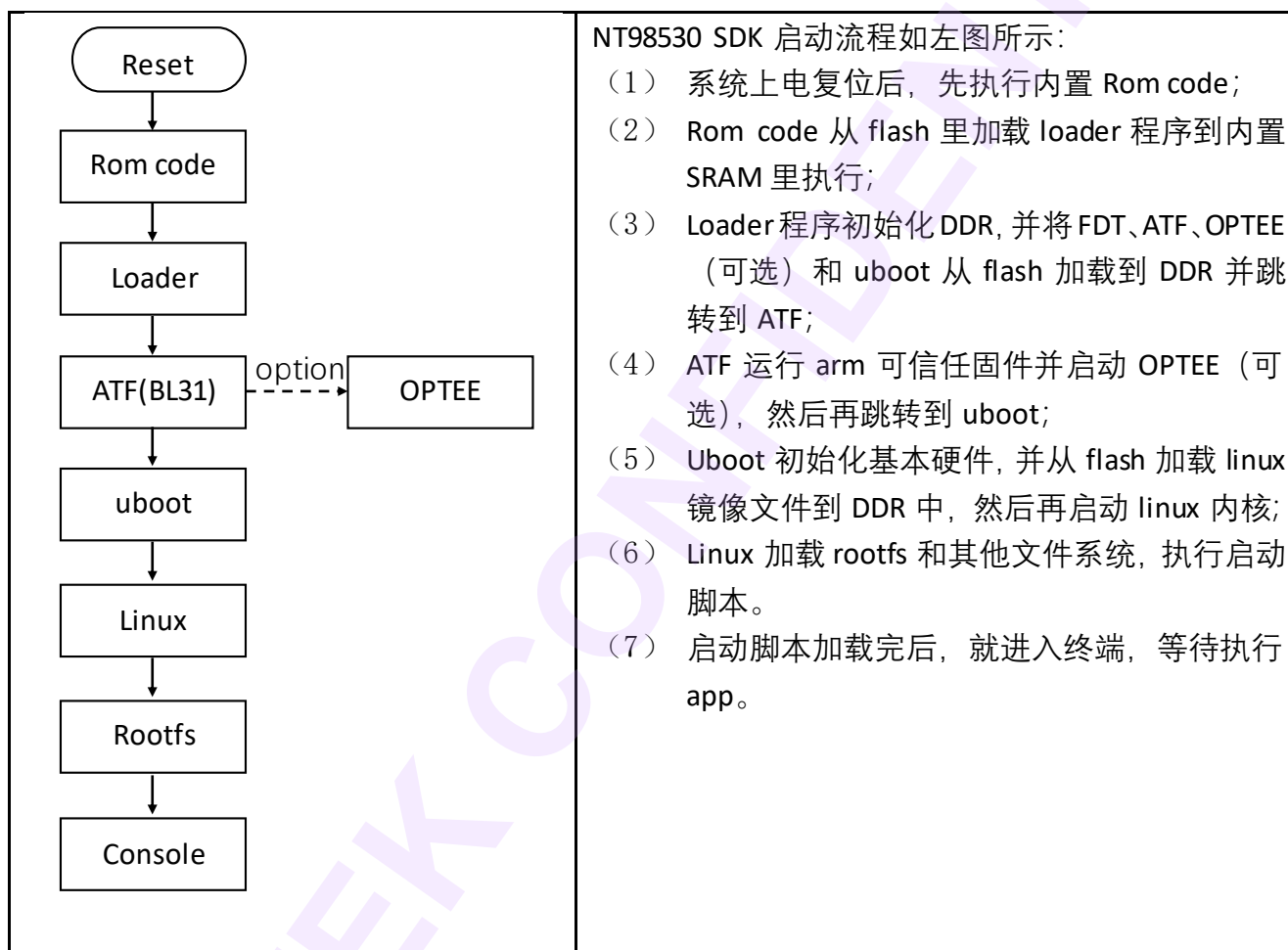
- 应用场景
 - Boot loader 阶段可以读取 SD 卡
- 使用 SD 卡烧写固件步骤：
 - SD 卡为 FAT32 文件格式
 - 放入 boot loader bin，如 LD98530A.bin（有需求更新 boot loader）
 - 放入 all-in-one FW，如 FW98530A.bin（有需求更新 all-in-on FW）
 - 插卡上电开机，即自动烧写

4.2.2 Ubootshell

- 应用场景
 - 可正常启动到 uboot 命令行窗口
- 使用 tftp 烧写 all-in-one 固件步骤：
 - 单板上电后，敲任意键进入 uboot 命令行
 - 配置网络环境：
 - ◆ setenv serverip xx.xx.xx.xx, 如 setenv serverip 192.168.0.8
 - ◆ setenv ipaddr xx.xx.xx.xx, 如 setenv ipaddr 192.168.0.3
 - ◆ ping [serverip], 如 ping 192.168.0.8
 - tftp 下载 all-in-one 固件到 DRAM
 - ◆ tftpboot 0x6000000 FW98530A.bin
 - 更新 DRAM 里的固件到存储介质
 - ◆ nvt_update_all 0x6000000 0

5 开机启动流程

5.1 启动流程概述



5.2 启动源配置

可参考 Datasheet 或 NT98530_Boot_Rom_code_User_Guide_en.pdf 文档说明。

Boot Source	BST[9]	BST[8]	BST[3]	BST[2]	BST[1]	BST[0]	Description	Message
SPI NOR	-	-	0	0	0	0	-	NP
SPI NAND	-	-	0	0	1	0	On die ECC (2K page)	ND
	-	-	0	0	1	1	BCH (2K page)	NB
	-	-	0	1	1	0	On die ECC(4K page)	ND
	-	-	1	0	1	0	BCH (4K page)	NB
eMMC	-	-	1	0	0	0	4 bits data bus	NE4
	-	-	1	0	0	1	8 bits data bus	NE8
SDIO (SDIO1)	-	-	0	0	0	1	-	NS1
Ethernet (MAC0)	0	0	0	1	0	0	RMII (refclk output)	NHO
	0	1	0	1	0	0	RMII (refclk input)	NHI
	1	0	0	1	0	0	RGMII	NHG
USB(U3)	-	-	0	1	0	1	High speed	NU
	-	-	1	0	1	1	Full speed	NU3
UART (UART1)	-	-	1	1	0	0	-	N

5.3 启动流程各部分

5.3.1 Rom code

参考 NT98530_Boot_Rom_code_User_Guide_en.pdf

5.3.2 Boot Loader

参考 NT98530_Boot_Loader_User_Guide_en.pdf

5.3.3 Uboot

参考 NT98530_SDK_UBoot_Programming_Guide_en.pdf

5.3.4 Kernel

参考 NT98530_SDK_Pure_Linux_Programming_Guide_en.pdf

5.3.5 Rootfs

参考 Filesystem_User_Guide_en.pdf

5.3.6 启动脚本

- 启动脚本存放在 BSP\root-fs\rootfs\etc_Model\etc_xxx\init.d\
- 启动脚本加载顺序：

```
rcS >> S00_PreReady >> S05_FS_Overlay >> S07_SysInit >> S10_SysInit2 >>  
S15_NvtApplInit >> S25_Net >> S99_Sysctl
```

说明：

S10_SysInit2：加载 hda1 相关 ko

5.4 开机启动日志

```
NDD>TLZMA
```

```
UNZOK!ini ver[0x40220613]
```

```
zq_trim[10]
```

```
ddr_speed 3733
```

```
CPU1400 DONE
```

```
>RPA
```

```
ddr init done
```

```
UNZOK!
```

```
Loader Start ...
```

```
LD_VER 03.03.02
```

```
530_LPDDR4_1866_8192Mb_8192Mb 07/07/2022 17:43:22
```

```
SD identify fail
```

```
card open fail
Pad driving increased: lv6

0xF0400008: 0x00103004
0xF040000C: 0x0500A000
0xF0030100: 0x22226666
0xF0030104: 0x22223666SPI NAND MID=000000C2 DEV=00000012
tmp_addr 0x08000000
LdCtrl2 0x00000000
l_bl31
bl31_addr 0x01F00000
bl31_size 0x00007120
l_uboot
uboot_addr 0x7E000000
uboot_size 0x0008FE10
core_cnt=0x00000004
smp(bl31)
fdt 0x00100000
shm 0x00A00000

    core2_jump 0xF0E00800
    code2JumpCodelen 0x00000010
    core2_entry2 0x00010000
    core2_entry 0xF0E00810
    code2EntryCodelen 0x00000188
    core2_entry2 0x00010000
bl_core2_reset
64core2_jump_program 0x0100605D
64code2JumpCodelen 0x0000018C
warm reset core0
jump_entry 0x01F00000
###NOTICE: BL31: v2.2(release):
NOTICE: BL31: Built : 11:11:13, Jul  8 2022
NOTICE: BL31: Found DTB
NOTICE: TZASC config:
NOTICE: 0x00100000@0x01f00000 (RW/N) Ree
NOTICE: arb_enable_wp level 0
```

NOTICE: arb_enable_wp level 2
INFO: ARM GICv2 driver initialized
INFO: BL31: Platform setup done
INFO: BL31: Initializing runtime services
INFO: BL31: Preparing for EL3 exit to normal world
INFO: Entry point address = 0x7e000000
INFO: SPSR = 0x3c9

U-Boot 2019.04 (Jul 08 2022 - 11:11:56 +0800)

CPU: Novatek NT @ 1399 MHz

DRAM: 2 GiB

Relocation to 0x7ff45000, Offset is 0x01f45000 sp at 7fc20e80

nvt_shminfo_init: The fdt buffer addr: 0x7fc43a20

otp_init[1.00.001]

120MHz

otp_timing_reg=0xff6050

CONFIG_MEM_SIZE	=	0x80000000
CONFIG_NVT_UIIMAGE_SIZE	=	0x01900000
CONFIG_NVT_ALL_IN_ONE_IMG_SIZE	=	0x07800000
CONFIG_UBOOT_SDRAM_BASE	=	0x7e000000
CONFIG_UBOOT_SDRAM_SIZE	=	0x02000000
CONFIG_LINUX_SDRAM_BASE	=	0x02000000
CONFIG_LINUX_SDRAM_SIZE	=	0x7c000000
CONFIG_LINUX_SDRAM_START	=	0x7c700000

NAND: NAND version: 1.0.31

drv_nand_reset: spi flash pinmux 0x0

ecc-mode = <0>

nand chip_sel = <0>

id = 0xc2 0x12 0xc2 0x12

use flash on-die ecc

nvt spinand 4-bit mode @ 24000000 Hz

128 MiB

MMC: sdio pinmux(0x1)

sdio pinmux(0x1)

NVT_MMC0: 0, NVT_MMC1: 1

```
misc_init_r: Firmware name: FW98530A.bin FW98530T.bin FW98530A.fdt.bin
misc_init_r: boot time: 860695(us)
nvt_detect_fw_tbin: Boot from flash or emmc
CMD8 timeout, status 44
CMD55 timeout, status 44
CMD1 timeout, status 44
Card did not respond to voltage select!
CMD8 timeout, status 44
CMD55 timeout, status 44
CMD1 timeout, status 44
Card did not respond to voltage select!
MMC interface configure failed
Open SD fail: No SD device?

NAND read: device 0 offset 0x40000, size 0x20000
131072 bytes read: OK

NAND read: device 0 offset 0x60000, size 0xf49
3913 bytes read: OK
nvt_getfdt_emb_part_info: Skip this partition node: partition_fdt.restore
mtdparts=mtdparts=spi_nand.0:0x40000@0x40000(fdt),0x40000@0xc0000(atf),0x1c0000@0x100000(u
boot),0x40000@0x2c0000(uenv),0x500000@0x300000(linux),0x3160000@0x800000(rootfs0),0x250000
0@0x3960000(rootfs1),0x21a0000@0x5e60000(app)
misc_init_r: boot time: 1121626(us)
DTS find cpu freq clock 1400MHz
Set CPU clk 1400MHz
misc_init_r: boot time: 1200790(us)
Net: na51102_eth_initialize 1.0.0.5
na51102_eth_initialize: dtb node /eth0@2,f02b0000 found
eqos_initialize 0x2f02b0000
eth_parse_phy_intf: get IO MEM 0x2f02b0000
DTS/eth0@2,f02b0000 found
eth_parse_phy_intf: ref-clk-out 0
eth_parse_phy_intf: pinmux detect RGMII 0x5
na51102_eth_initialize: dtb node /eth1@2,f0430000 found
eqos_initialize 0x2f0430000
```

```
eth_parse_phy_intf: get IO MEM 0x2f0430000
DTS/eth1@2,f0430000 found
eth_parse_phy_intf: ref-clk-out 0
eth_parse_phy_intf: pinmux detect emb phy 0x5
eth_parse_phy_intf: don't have sup embd. return
eth0
Hit any key to stop autoboot:  0
  do_nvt_boot_cmd: boot time: 1427530(us)
NVT firmware boot.....
nvt_detect_fw_tbin: Boot from flash or emmc

NAND read: device 0 offset 0x300000, size 0x40
  64 bytes read: OK

NAND read: device 0 offset 0x300000, size 0x253ee0
  2440928 bytes read: OK
nvt_ker_img_ungzip_linux: not gzip linux
nvt_boot_linux_bin_auto: linux_addr:0x00000000
nvt_boot_linux_bin_auto: linux_size:0x00253ee0
  do_nvt_boot_cmd: boot time: 1767868(us)
do_nvt_boot_cmd: bootargs:earlycon=nvt_serial,0x2f0290000 rootwait console=ttyS0,115200
debug_boot_weak_hash  root=ubi0:rootfs rootfstype=ubifs ubi.fm_autoconvert=1 init=/linuxrc
ubi.mtd=7 ro  bootts=757645,1772287 resume_addr=0x00a00088 print-fatal-signals=1
  Image Name:   Linux-4.19.148
  Image Type:   AArch64 Linux Kernel Image (uncompressed)
  Data Size:    2440864 Bytes = 2.3 MiB
  Load Address: 00080000
  Entry Point:  00080000
Linux Image is at 7c700000, uboot fdt image is at 7fc43a20, loader tmp fdt address is at 100000
booti 7c700040 - 7fc43a20
do_nvt_boot_cmd: boot time: 1823346(us)
do_nvt_boot_cmd: Uboot boot time:
  start:  757645 us
  ending: 1772287 us
kernel image compression type 3 size = 0x00253ee0 address = 0x00000000
Uncompressing Kernel Image
```

Boot Kernel from 0x7c700040 to 0x0

Copy Kernel to 0x0

Flattened Device Tree blob at 7fc43a20

Booting using the fdt blob at 0x7fc43a20

Loading Device Tree to 0000000017fdc000, end 0000000017ffff48 ... OK

Starting kernel ...

[0.000000] Booting Linux on physical CPU 0x0000000000 [0x410fd034]

[0.000000] Linux version 4.19.148 (root@cnhz-152355-01) (gcc version 8.4.0 (Buildroot 2020.02.9-23-g81e333f)) #1 SMP PREEMPT Thu Jul 7 16:39:39 CST 2022

[0.000000] Machine model: Novatek NA51102

[0.000000] earlycon: nvt_serial0 at MMIO 0x00000002f0290000 (options "")

[0.000000] bootconsole [nvt_serial0] enabled

[0.000000] debug_boot_weak_hash enabled

[0.000000] Reserved memory: created CMA memory pool at 0x0000000002800000, size 12 MiB

[0.000000] OF: reserved mem: initialized node dsp_cma@0x02800000, compatible id

shared-dma-pool

[0.000000] Reserved memory: created CMA memory pool at 0x0000000005000000, size 8 MiB

[0.000000] OF: reserved mem: initialized node cma0@0x05000000, compatible id shared-dma-pool

[0.000000] psci: probing for conduit method from DT.

[0.000000] psci: PSCIv1.1 detected in firmware.

[0.000000] psci: Using standard PSCI v0.2 function IDs

[0.000000] psci: MIGRATE_INFO_TYPE not supported.

[0.000000] psci: SMC Calling Convention v1.1

[0.000000] random: get_random_bytes called from start_kernel+0x90/0x498 with crng_init=0

[0.000000] percpu: Embedded 20 pages/cpu s44120 r8192 d29608 u81920

[0.000000] Detected VIPT I-cache on CPU0

[0.000000] CPU features: enabling workaround for ARM erratum 845719

[0.000000] Speculative Store Bypass Disable mitigation not required

[0.000000] CPU features: kernel page table isolation disabled by kernel configuration

[0.000000] Built 1 zonelists, mobility grouping on. Total pages: 96512

[0.000000] Kernel command line: earlycon=nvt_serial,0x2f0290000 rootwait console=ttyS0,115200

debug_boot_weak_hash root=ubi0:rootfs rootfstype=ubifs ubi.fm_autoconvert=1 init=/linuxrc

ubi.mtd=7 ro bootts=757645,1772287 resume_addr=0x00a00088 print-fatal-signals=1

[0.000000] Dentry cache hash table entries: 65536 (order: 7, 524288 bytes)

```
[ 0.000000] Inode-cache hash table entries: 32768 (order: 6, 262144 bytes)
[ 0.000000] Memory: 356444K/392192K available (4798K kernel code, 312K rwddata, 1124K rodata,
320K init, 291K bss, 15268K reserved, 20480K cma-reserved)
[ 0.000000] Virtual kernel memory layout:
[ 0.000000]   modules : 0xfffff80000000000 - 0xfffff80080000000   ( 128 MB)
[ 0.000000]   vmalloc : 0xfffff80080000000 - 0xfffff80080000000   ( 250 GB)
[ 0.000000]   .text : 0xfffff80080800000 - 0xfffff80080853000   ( 4800 KB)
[ 0.000000]   .rodata : 0xfffff80080853000 - 0xfffff80080865000   ( 1152 KB)
[ 0.000000]   .init : 0xfffff80080865000 - 0xfffff8008086a000   ( 320 KB)
[ 0.000000]   .data : 0xfffff8008086a000 - 0xfffff8008086ee008   ( 313 KB)
[ 0.000000]   .bss : 0xfffff8008086ee008 - 0xfffff800808736db8   ( 292 KB)
[ 0.000000]   fixed   : 0xfffff800808736db8 - 0xfffff800808736db8   ( 4108 KB)
[ 0.000000]   PCI I/O : 0xfffff800808736db8 - 0xfffff800808736db8   ( 16 MB)
[ 0.000000]   memory   : 0xfffff800808736db8 - 0xfffff800808736db8   ( 384 MB)
[ 0.000000] SLUB: HWalign=64, Order=0-3, MinObjects=0, CPUs=4, Nodes=1
[ 0.000000] rcu: Preemptible hierarchical RCU implementation.
[ 0.000000]   Tasks RCU enabled.
[ 0.000000] NR_IRQS: 64, nr_irqs: 64, preallocated irq: 0
[ 0.000000] GIC: Using split EOI/Deactivate mode
[ 0.000000] clk_nvt_init: CG clock init done
[ 0.000000] arch_timer: cp15 timer(s) running at 12.00MHz (phys).
[ 0.000000] clocksource: arch_sys_counter: mask: 0xffffffffffffff max_cycles: 0x2c47f4ee7,
max_idle_ns: 440795202497 ns
[ 0.000003] sched_clock: 56 bits at 12MHz, resolution 83ns, wraps every 4398046511096ns
[ 0.008470] Console: colour dummy device 80x25
[ 0.012747] Calibrating delay loop (skipped), value calculated using timer frequency.. 24.00
BogoMIPS (lpj=120000)
[ 0.023047] pid_max: default: 32768 minimum: 301
[ 0.027698] Mount-cache hash table entries: 1024 (order: 1, 8192 bytes)
[ 0.034230] Mountpoint-cache hash table entries: 1024 (order: 1, 8192 bytes)
[ 0.041882] ASID allocator initialised with 65536 entries
[ 0.047149] rcu: Hierarchical SRCU implementation.
[ 0.052363] NVTBOOTTS: nvt_bootts_init initial success
[ 0.057323] NVTBOOTTS: nvt_bootts_proc_init initial success
[ 0.063035] smp: Bringing up secondary CPUs ...
[ 0.067616] Detected VIPT I-cache on CPU1
```



```
[ 0.067651] CPU1: Booted secondary processor 0x0000000001 [0x410fd034]
[ 0.067910] Detected VIPT I-cache on CPU2
[ 0.067927] CPU2: Booted secondary processor 0x0000000002 [0x410fd034]
[ 0.068159] Detected VIPT I-cache on CPU3
[ 0.068176] CPU3: Booted secondary processor 0x0000000003 [0x410fd034]
[ 0.068213] smp: Brought up 1 node, 4 CPUs
[ 0.103592] SMP: Total of 4 processors activated.
[ 0.108265] CPU features: detected: 32-bit EL0 Support
[ 0.113424] CPU: All CPU(s) started at EL2
[ 0.117462] alternatives: patching kernel code
[ 0.122231] devtmpfs: initialized
[ 0.142983] clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns:
19112604462750000 ns
[ 0.152647] futex hash table entries: 1024 (order: 4, 65536 bytes)
[ 0.159323] pinctrl core: initialized pinctrl subsystem
[ 0.164650] nvtmem_dram_mapping_init: ddr_id: 0x00000000 chipid: 0x00000000 addr:
0x00000000 size: 0x80000000 active: 0x00000001
[ 0.164749] nvtmem_load_hdal_base: Parsing HDAL mem: ID: 0x00000000 Addr: 0x18000000 Size:
0x68000000
[ 0.185220] nvtmem: register char device successfully!
[ 0.190339] NVTMEM: 0.0.7
[ 0.192937] sramctl ver 1.0.0
[ 0.196081] NET: Registered protocol family 16
[ 0.200850] hw-breakpoint: found 6 breakpoint and 4 watchpoint registers.
[ 0.207562] DMA: preallocated 256 KiB pool for atomic allocations
[ 0.213539] *** nvt_poweroff_setup
[ 0.216981] nvt_otp_module_init
[ 0.226441] ext_clkout_src_set no extclk_src node found in dtsi
[ 0.232208] ext_clkout_src_set no extclk_src_ep0 node found in dtsi
[ 0.241605] pad_set_power: Not Existed PAD_POWER for P1
[ 0.246733] power_init: pad_power_id(20) set pad_power(0) failed!
[ 0.259252] nvt_cma nvt_cma: assigned reserved memory node cma0@0x05000000
[ 0.265976] nvt_cma nvt_cma: fmem: cma area id: 0, cma base: 0x5000000, size: 0x800000 added
[ 0.274174] nvt_cma nvt_cma: fmem/Version: 1.0.0
[ 0.278780] nvt_cma nvt_cma: Probe successfully
[ 0.283335] nvt_cma nvt_dsp_cma: assigned reserved memory node dsp_cma@0x02800000
```

```
[ 0.290740] nvt_cma nvt_dsp_cma: fmem: cma area id: 1, cma base:0x2800000, size:0xc00000
added
[ 0.299307] nvt_cma nvt_dsp_cma: fmem/Version: 1.0.0
[ 0.304252] nvt_cma nvt_dsp_cma: Probe successfully
[ 0.309158] nvt_os_mem nvt_os_mem: fmem/Version: 1.0.0
[ 0.314214] nvt_os_mem nvt_os_mem: Probe successfully
[ 0.321350] cryptd: max_cpu_qlen set to 1000
[ 0.328797] nvt_gpio 2f0070000.gpio: Register nvt_gpio_probe successfully
[ 0.335993] set mask to 32bit
[ 0.338868] nvt_i2c 2f0220000.i2c1: Register nvtim_i2c_probe ver=1.00.030 successfully
[ 0.346870] set mask to 32bit
[ 0.349787] nvt_i2c 2f0350000.i2c2: Register nvtim_i2c_probe ver=1.00.030 successfully
[ 0.357775] set mask to 32bit
[ 0.360707] nvt_i2c 2f03a0000.i2c3: Register nvtim_i2c_probe ver=1.00.030 successfully
[ 0.368733] set mask to 32bit
[ 0.371626] nvt_i2c 2f03b0000.i2c4: Register nvtim_i2c_probe ver=1.00.030 successfully
[ 0.379614] set mask to 32bit
[ 0.382547] nvt_i2c 2f03c0000.i2c5: Register nvtim_i2c_probe ver=1.00.030 successfully
[ 0.390534] set mask to 32bit
[ 0.393467] nvt_i2c 2f0200000.i2c6: Register nvtim_i2c_probe ver=1.00.030 successfully
[ 0.401454] set mask to 32bit
[ 0.404387] nvt_i2c 2f02c0000.i2c7: Register nvtim_i2c_probe ver=1.00.030 successfully
[ 0.412371] set mask to 32bit
[ 0.415307] nvt_i2c 2f02d0000.i2c8: Register nvtim_i2c_probe ver=1.00.030 successfully
[ 0.423297] set mask to 32bit
[ 0.426227] nvt_i2c 2f02e0000.i2c9: Register nvtim_i2c_probe ver=1.00.030 successfully
[ 0.434212] set mask to 32bit
[ 0.437147] nvt_i2c 2f02f0000.i2c10: Register nvtim_i2c_probe ver=1.00.030 successfully
[ 0.445224] set mask to 32bit
[ 0.448154] nvt_i2c 2f0410000.i2c11: Register nvtim_i2c_probe ver=1.00.030 successfully
[ 0.456675] clocksource: Switched to clocksource arch_sys_counter
[ 0.462640] VFS: Disk quotas dquot_6.6.0
[ 0.466521] VFS: Dquot-cache hash table entries: 512 (order 0, 4096 bytes)
[ 0.476143] NET: Registered protocol family 2
[ 0.480390] IP idents hash table entries: 8192 (order: 4, 65536 bytes)
[ 0.487179] tcp_listen_portaddr_hash hash table entries: 256 (order: 0, 4096 bytes)
```

```
[ 0.494651] TCP established hash table entries: 4096 (order: 3, 32768 bytes)
[ 0.501700] TCP bind hash table entries: 4096 (order: 4, 65536 bytes)
[ 0.508133] TCP: Hash tables configured (established 4096 bind 4096)
[ 0.514451] UDP hash table entries: 256 (order: 1, 8192 bytes)
[ 0.520230] UDP-Lite hash table entries: 256 (order: 1, 8192 bytes)
[ 0.526517] NET: Registered protocol family 1
[ 0.531309] hw perfevents: enabled with armv8_pmu3 PMU driver, 7 counters available
[ 0.540972] Initialise system trusted keyrings
[ 0.545290] workingset: timestamp_bits=62 max_order=17 bucket_order=0
[ 0.554570] squashfs: version 4.0 (2009/01/31) Phillip Lougher
[ 0.562169] Key type asymmetric registered
[ 0.566081] Asymmetric key parser 'x509' registered
[ 0.570969] Block layer SCSI generic (bsg) driver version 0.4 loaded (major 250)
[ 0.578308] io scheduler noop registered
[ 0.582200] io scheduler deadline registered (default)
[ 0.587403] io scheduler cfq registered
[ 0.591127] io scheduler mq-deadline registered (default)
[ 0.596500] io scheduler kyber registered
[ 0.601438] 2f0290000.uart: ttyS0 at MMIO 0x2f0290000 (irq = 10, base_baud = 3000000) is a NVT
[ 0.609868] nvt_serial: console setup on port #0
[ 0.614475] console [ttyS0] enabled
[ 0.614475] console [ttyS0] enabled
[ 0.621391] bootconsole [nvt_serial0] disabled
[ 0.621391] bootconsole [nvt_serial0] disabled
[ 0.630391] 2f0300000.uart: ttyS1 at MMIO 0x2f0300000 (irq = 11, base_baud = 3000000) is a NVT
[ 0.639100] 2f0310000.uart: ttyS2 at MMIO 0x2f0310000 (irq = 12, base_baud = 3000000) is a NVT
[ 0.647803] 2f0380000.uart: ttyS3 at MMIO 0x2f0380000 (irq = 13, base_baud = 3000000) is a NVT
[ 0.656499] 2f03e0000.uart: ttyS4 at MMIO 0x2f03e0000 (irq = 14, base_baud = 3000000) is a NVT
[ 0.665201] 2f03f0000.uart: ttyS5 at MMIO 0x2f03f0000 (irq = 15, base_baud = 3000000) is a NVT
[ 0.673911] 2f0470000.uart: ttyS6 at MMIO 0x2f0470000 (irq = 16, base_baud = 3000000) is a NVT
[ 0.682615] 2f0480000.uart: ttyS7 at MMIO 0x2f0480000 (irq = 17, base_baud = 3000000) is a NVT
[ 0.691321] 2f0490000.uart: ttyS8 at MMIO 0x2f0490000 (irq = 18, base_baud = 3000000) is a NVT
[ 0.700146] nvt_serial: driver initialized, nr_uarts = 9
[ 0.709393] loop: module loaded
[ 0.712667] spi_nand 2f0400000.nand: nand version: 1.04.206
[ 0.718393] spi_nand 2f0400000.nand: dma buffer va: 0000000080d31558, pa: 0x16640000
```

```
[ 0.726116] spi_nand 2f0400000.nand: chip_select 0
[ 0.730983] spi_nand 2f0400000.nand: ecc-mode: 0
[ 0.736867] spi_nand 2f0400000.nand: use flash on-die ecc
[ 0.742261] nand: device found, Manufacturer ID: 0xc2, Chip ID: 0x12
[ 0.748601] nand: Macronix MX35LF1GE4AB 1GiB 3.3V
[ 0.753288] nand: 128 MiB, SLC, erase size: 128 KiB, page size: 2048, OOB size: 64
[ 0.760846] Scanning device for bad blocks
[ 0.784833] random: fast init done
[ 0.929374] spi_nand 2f0400000.nand: 4-bit mode @ 24000000 Hz
[ 0.935344] 11 fixed-partitions partitions found on MTD device spi_nand.0
[ 0.942118] Creating 11 MTD partitions on "spi_nand.0":
[ 0.947336] 0x000000000000-0x000000040000 : "loader"
[ 0.952873] 0x000000040000-0x000000080000 : "fdt"
[ 0.958048] 0x000000080000-0x0000000c0000 : "fdt.restore"
[ 0.963917] 0x0000000c0000-0x000000100000 : "atf"
[ 0.969093] 0x000000100000-0x0000002c0000 : "uboot"
[ 0.974448] 0x0000002c0000-0x000000300000 : "uenv"
[ 0.979713] 0x000000300000-0x000000800000 : "kernel"
[ 0.985154] 0x000000800000-0x0000003960000 : "rootfs"
[ 0.990694] 0x0000003960000-0x0000005e60000 : "rootfs1"
[ 0.996285] 0x0000005e60000-0x0000008000000 : "app"
[ 1.001533] 0x000000000000-0x0000008000000 : "all"
[ 1.007329] mousedev: PS/2 mouse device common for all mice
[ 1.026762] nvt_rtc 2f0060000.rtc: rtc core: registered nvt_rtc as rtc0
[ 1.033521] i2c /dev entries driver
[ 1.037851] NET: Registered protocol family 10
[ 1.042768] Segment Routing with IPv6
[ 1.046449] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
[ 1.052651] NET: Registered protocol family 17
[ 1.057366] Loading compiled-in X.509 certificates
[ 1.073034] ubi0: attaching mtd7
[ 1.295435] ubi0: scanning is finished
[ 1.305964] ubi0: attached mtd7 (name "rootfs", size 49 MiB)
[ 1.311614] ubi0: PEB size: 131072 bytes (128 KiB), LEB size: 126976 bytes
[ 1.318474] ubi0: min./max. I/O unit sizes: 2048/2048, sub-page size 2048
[ 1.325242] ubi0: VID header offset: 2048 (aligned 2048), data offset: 4096
```

```
[ 1.332188] ubi0: good PEBs: 395, bad PEBs: 0, corrupted PEBs: 0
[ 1.338179] ubi0: user volume: 1, internal volumes: 1, max. volumes count: 128
[ 1.345380] ubi0: max/mean erase counter: 1/0, WL threshold: 4096, image sequence number:
1235683750
[ 1.354493] ubi0: available PEBs: 12, total reserved PEBs: 383, PEBs reserved for bad PEB handling:
20
[ 1.363787] ubi0: background thread "ubi_bgt0d" started, PID 109
[ 1.363807] nvt_rtc 2f0060000.rtc: setting system clock to 1970-01-01 09:14:29 UTC (33269)
[ 1.448929] UBIFS (ubi0:0): UBIFS: mounted UBI device 0, volume 0, name "rootfs", R/O mode
[ 1.457185] UBIFS (ubi0:0): LEB size: 126976 bytes (124 KiB), min./max. I/O unit sizes: 2048
bytes/2048 bytes
[ 1.467080] UBIFS (ubi0:0): FS size: 44314624 bytes (42 MiB, 349 LEBs), journal size 5967872 bytes (5
MiB, 47 LEBs)
[ 1.477493] UBIFS (ubi0:0): reserved for root: 0 bytes (0 KiB)
[ 1.483308] UBIFS (ubi0:0): media format: w4/r0 (latest is w5/r0), UUID
532DD238-925F-4899-9074-8860D2581486, small LPT model
[ 1.496038] VFS: Mounted root (ubifs filesystem) readonly on device 0:13.
[ 1.505081] devtmpfs: mounted
[ 1.508141] Freeing unused kernel memory: 320K
[ 1.512602] Run /linuxrc as init process
[Start]/etc/init.d/S00_PreReady
[Start]/etc/init.d/S05_FS_Overlay
[fs overlay] ubi mount rootfs1
[ 1.821577] ubi1: attaching mtd8
[ 1.988970] ubi1: scanning is finished
[ 1.999528] ubi1: attached mtd8 (name "rootfs1", size 37 MiB)
[ 2.005263] ubi1: PEB size: 131072 bytes (128 KiB), LEB size: 126976 bytes
[ 2.012133] ubi1: min./max. I/O unit sizes: 2048/2048, sub-page size 2048
[ 2.018907] ubi1: VID header offset: 2048 (aligned 2048), data offset: 4096
[ 2.025848] ubi1: good PEBs: 296, bad PEBs: 0, corrupted PEBs: 0
[ 2.031839] ubi1: user volume: 1, internal volumes: 1, max. volumes count: 128
[ 2.039045] ubi1: max/mean erase counter: 2/0, WL threshold: 4096, image sequence number:
1054876922
[ 2.048157] ubi1: available PEBs: 12, total reserved PEBs: 284, PEBs reserved for bad PEB handling:
20
[ 2.057452] ubi1: background thread "ubi_bgt1d" started, PID 129
```

```
[ 2.062283] UBIFS(ubi1:0): background thread "ubifs_bgt1_0" started, PID 132
[ 2.097125] UBIFS(ubi1:0): recovery needed
[ 2.195454] UBIFS(ubi1:0): recovery completed
[ 2.199956] UBIFS(ubi1:0): UBIFS: mounted UBI device 1, volume 0, name "rootfs"
[ 2.207341] UBIFS(ubi1:0): LEB size: 126976 bytes (124 KiB), min./max. I/O unit sizes: 2048
bytes/2048 bytes
[ 2.217241] UBIFS(ubi1:0): FS size: 31744000 bytes (30 MiB, 250 LEBs), journal size 4444160 bytes (4
MiB, 35 LEBs)
[ 2.227657] UBIFS(ubi1:0): reserved for root: 0 bytes (0 KiB)
[ 2.233474] UBIFS(ubi1:0): media format: w4/r0 (latest is w5/r0), UUID
E79F6E36-23AE-4D23-A603-F586B852A4FC, small LPT model
[ 2.257856] S05_FS_Overlay(116): drop_caches: 1
[Start]/etc/init.d/S07_APP_Overlay
[app overlay] ubi mount app
[ 2.296159] ubi2: attaching mtd9
[ 2.457318] ubi2: scanning is finished
[ 2.467848] ubi2: attached mtd9 (name "app", size 33 MiB)
[ 2.473252] ubi2: PEB size: 131072 bytes (128 KiB), LEB size: 126976 bytes
[ 2.480126] ubi2: min./max. I/O unit sizes: 2048/2048, sub-page size 2048
[ 2.486907] ubi2: VID header offset: 2048 (aligned 2048), data offset: 4096
[ 2.493852] ubi2: good PEBs: 269, bad PEBs: 0, corrupted PEBs: 0
[ 2.499848] ubi2: user volume: 1, internal volumes: 1, max. volumes count: 128
[ 2.507059] ubi2: max/mean erase counter: 1/0, WL threshold: 4096, image sequence number:
1930386159
[ 2.516172] ubi2: available PEBs: 12, total reserved PEBs: 257, PEBs reserved for bad PEB handling:
20
[ 2.525470] ubi2: background thread "ubi_bgt2d" started, PID 148
[ 2.531257] UBIFS(ubi2:0): background thread "ubifs_bgt2_0" started, PID 151
[ 2.566041] UBIFS(ubi2:0): recovery needed
[ 2.609263] UBIFS(ubi2:0): recovery completed
[ 2.613741] UBIFS(ubi2:0): UBIFS: mounted UBI device 2, volume 0, name "app"
[ 2.620865] UBIFS(ubi2:0): LEB size: 126976 bytes (124 KiB), min./max. I/O unit sizes: 2048
bytes/2048 bytes
[ 2.630762] UBIFS(ubi2:0): FS size: 28315648 bytes (27 MiB, 223 LEBs), journal size 4063232 bytes (3
MiB, 32 LEBs)
[ 2.641176] UBIFS(ubi2:0): reserved for root: 0 bytes (0 KiB)
```



```
[ 2.646999] UBIFS (ubi2:0): media format: w4/r0 (latest is w5/r0), UUID
535C7C3B-4524-4CFB-A71F-917BC9BB5AFF, small LPT model
[Start]/etc/init.d/S07_SysInit
[ 2.738936] nvt_mmc 2f0420000.mmc: cd_gpio is invalid
[ 2.776699] nvt_mmc 2f0420000.mmc: Using DMA, 4-bit mode sampling Positive edge, mmc0,
pinmux(0x1)
[ 2.846698] nvt_mmc 2f0500000.mmc: Using DMA, 4-bit mode sampling Positive edge, mmc1,
pinmux(0x1)
[Start]/etc/init.d/S10_SysInit2
[ 3.963721] kwrap: loading out-of-tree module taints kernel.
[ 3.974778] nvt_vos: 1.00.020.00 (Jul 7 2022 16:41:36)
[ 4.164611] kdrv_builtin_init init module =====
[ 4.170743] parse sie power_saving failed
[ 4.175789] parse vie power_saving failed
[ 4.181419] h26x_open:(0) HW_VERSION = 0x12020569, CLK = 379999969
[ 4.187703] vtrc_open:VTRC_INT_ENABLE = 1
[ 4.194035]
[ 4.194035] ===== nvt_jpg version: 1.0.16 =====
[ 4.205894] kdrv_builtin_init init module =====
[ 4.273361]
[ 4.276847] 2f00d0000.nvt_arb
[ 4.279808] 0. resource:0x2f00d0000 size:0x6000
[ 4.284325] 1. resource:0x2f00a0000 size:0x9000
[ 4.288937] 2. resource:0x2f0120000 size:0x9000
[ 4.293451] 3. resource:0x2f0130000 size:0x9000
[ 4.297970] 4. resource:0x2f0011100 size:0x100
[ 4.302442] IRQ0. ID108
[ 4.305041] DevID Major:245 minor:16
[ 4.308883] dma_set_monitor_period() 933000000
[ 4.313401] ddr_arb_platform_create_resource: exit
[ 4.319906] log.ko v2.12: Jul 7 2022 16:41:55 (mmap 0xfffffc015b80000 size 0x21800 vmalloc 0)
[ 4.328617]
[ 4.328617] LOG base fffffc0169d1400(DDR0) size 1K (start pointer 0xfffffc0169d1400)
[ 4.337994] PAGE_OFFSET(0x0) VMALLOCSTART(0x8000000) HZ(100)
[ 4.344006] nvt_hwcopy_module_init:
[ 4.347709] nvt_hwcopy_probe:2f0610000.hwcp
```

```
[ 4.352112] nvt_hwcopy_probe:DevID Major:244 minor:0
[ 4.357249] nvt_hwcopy_drv_init: done
[ 4.361212] dma_util_init, VER: 1.1.7
[ 4.399912]
[ 4.522483] not fast boot
[ 4.527592] Start Senphy EOC scan
[ 4.586685] OFSCAL_DONE
[ 4.766438] media memory: 0 18000000 68000000 0 0 0 0 0
[ 4.772018] page 0x18000 0x7fff
[ 4.775433] nvtmpp_init_mmz: ddr1 pa = 0x18000000 va = 0xfffff80400000000 , size = 0x68000000
[ 5.149729] nvt_mi_module_init:
[ 5.163525] nvt_dsi_module_init:
[ 5.166973] nvt_dsi_probe:2f0840000.dsi
[ 5.170799] nvt_dsi_probe:0. resource:0x2f0840000 size:0x1000
[ 5.176581] nvt_dsi_probe:IRQ0. ID76
[ 5.180418] nvt_dsi_probe:DevID Major:245 minor:45
[ 5.196293] nvt_csi_tx_module_init:
[ 5.200010] nvt_csi_tx_probe:2f0850000.csitx
[ 5.204269] nvt_csi_tx_probe:0. resource:0x2f0850000 size:0x1000
[ 5.210322] nvt_csi_tx_probe:IRQ0. ID77
[ 5.214410] nvt_csi_tx_probe:DevID Major:245 minor:46
[ 5.261424] nvt_hdmitx_module_init: version = [1.00.02]
[ 5.267035] nvt_composite_mux_clk_prepare: 2f0820000.hdmi reset
[ 5.273076] hdmi hook i2c[11] driver done.
[ 5.302651] nvt_dispdev_module_init:
[ 5.313005] panel_init:can not find logo node!
[ 5.317568] panel_init:cfg file no exist
[ 5.321650] Hello, panel: PW35P00
[ 5.357119] kdrv_vpe Version: 1.00.016.00 Jul 7 2022 16:41:55
[ 5.362943] Ssdrv_vpe Version: 1.02.00 Jun 30 2022 01:34:13
[ 5.392030] h26x kdrv version: 0.1.6, nvt codec version 0.0.1.19, built @ Jul 7 2022 16:41:41
[ 5.416727] JPEG kdrv version: 0.1.3, nvt version 1.0.16, built @ Jul 7 2022 16:41:45
[ 5.442871] JPEG decode ver1.1.4
[ 5.507910] [DE] hw_timeout_delay:3750000 ticks, plck:150000000
[ 5.513977] H26X Decoder IRQmode PLT:98530, version 0.1.0.0 built @ Jul 7 2022 16:41:43
[ 5.743297] nvt_sde_module_init!
```



```
isp_id_list: 0x3F, ae_id_list: 0x3F, awb_id_list: 0x3F, iq_id_list 0x3F
[ 5.804551] nvt_ae: module license 'Proprietary' taints kernel.
[ 6.082371]
[ 6.083859] nvt_dsp driver version: 1.1.3
[ 6.088118] 2f0de0000.dsp
[ 6.090727] nvt_dsp_probe... Turn on power
[ 6.094934] IRQ0. ID100
[ 6.097464] DevID Major:239 minor:0
[ 6.101063] nvt_dsp_drv_init
[ 6.103970] io_addr = 0xffffffff80099f5000, 0xffffffff800b000000
[ 6.104237] dsp_open... De-Reset CEVASENSPRO_RSTN
[ 6.117848] dsp_open...
[ 6.152797] nvt_md_probe:of_property_read_u32_array failed.
[ 6.355067] nvt_drv_tse init, ver: 1.0.00 Jul 7 2022 16:41:38
[ 6.370382] nvt_gpenc_module_init:
[ 6.373951] nvt_gpenc_probe:2f0860000.gpe
[ 6.378326] nvt_gpenc_probe:0. resource:0x2f0860000 size:0x100
[ 6.384152] nvt_gpenc_probe:1. resource:0x2f0870000 size:0x100
[ 6.390239] nvt_gpenc_probe:IRQ0. ID79
[ 6.394081] nvt_gpenc_probe:IRQ1. ID80
[ 6.398045] nvt_gpenc_probe:DevID Major:245 minor:196
[Start]/etc/init.d/S15_NvtApplnit
run isp_demon
crond[760]: crond (busybox 1.31.1) started, log level 8

[ISPD] v0.9.5.616
[ISPD] open ISP succeed
[Start]/etc/init.d/S25_Net
[ISPD] bind 8888
[ISPD] listen ...
[ 6.486964] libphy: Fixed MDIO Bus: probed
[ 6.514729] nvt_eth_env_probe: IO MEM res start 0x2f02b0000
[ 6.520532] nvt_eth_env_probe: get IO MEM 0xffffffff800c010000
[ 6.526273] nvt_eth_env_probe: get pinmux 0x5
[ 6.530711] nvt_eth_env_probe: pinmux detect RGMII 0x5
[ 6.535920] nvt_eth_env_probe: find sp-clk 0
```

```
[ 6.540271] nvt_eth_env_probe: TBD: pinctrl still not implement split phy clk
[ 6.617511] DWC_ETH_QOS 2f02b0000.eth0: Assigning random MAC address.
[ 6.718144] DWC_ETH_QOS: Phy detected at ID/ADDR 1
[ 6.733300] libphy: dwc_phy: probed
[ 6.737031] netif_napi_add() called with weight 128 on device eth%d
[ 6.743281] Supports TSO, SG and TX COE
[ 6.747119] Supports RX COE and GRO
[ 6.751029] set mask to 32bit
[ 6.758207] phydev exist: resume phy
[ 6.763378] DWC_ETH_QOS_yinit: assume APB 120 MHz
[ 6.768184] DWC_ETH_QOS_yinit: apb_clk 120000000 Hz
[ 6.774136] Queue0 Tx fifo size 16384, Rx fifo size 16384
[ 6.779540] Disabled JUMBO pkt
[ 6.782583] Enabled Rx watchdog timer
[ 6.786233] Enabled TSO
[ 6.788677] Disabled Rx Split header mode
[ 6.792672] phydev exist: start phy
[ 6.796534] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
```

Configuring loopback interface

[Start]/etc/init.d/S99_Sysctl

login[795]: root login on 'UNKNOWN'

NVTEVM Linux shell...

root@NVTEVM:~\$

6 运行 hdal 媒体业务

在 SDK 包中提供各种 sample code 以便熟悉开发环境，如 code/hdal/samples/ 目录下提供 HDAL 媒体业务使用的范例，code/sample/ 以及 code/application/source/ 目录中提供了其他 user space 的范例。

单板更新 SDK 固件后，在 /usr/bin 目录下已集成部分媒体示例应用。
可参考 code\hdal\samples\readme.txt 了解这些示例应用的使用。

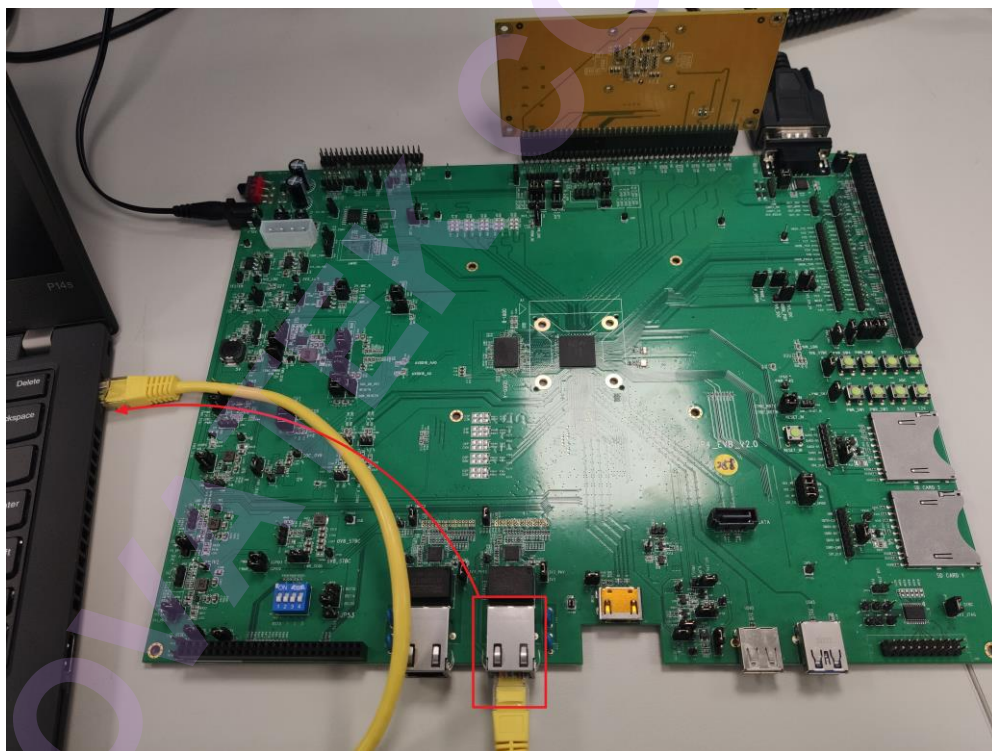
以下仅列举部分示例应用作说明。

6.1 RTSP server

RTSP server 的 sample code 位于 code/sample/pq_video_rtsp/，以及他所需要连结的 RTSP library 是 code/lib/source/nvtlive555。

预设 sdk 编译出来的 all-in-one 固件会包含这个 sample，可直接执行。

[Demo 单板网线连接]：如下图，优先使用 Ethernet 1。



[数据流]：sensor -> vcap -> vprc -> venc -> rtsp 流

[使用方式]:

通过执行 pq_video_rtsp 指令获取使用说明，然后根据单板的 sensor 及编码配置输入相应参数。

```
root@NVTEVM:~$ pq_video_rtsp
```

Usage: <sen_sel_1> <sen_shdr_1> <sen_sel_2> <sen_shdr_2> <enc_type> <enc_bitrate>.

<sen_sel_1 > : refer below sensor list
 <sen_shdr_1 > : 0(disable), 1(2-frame), 2(3-frame)
 <sen_sel_2 > : refer below sensor list
 <sen_shdr_2 > : 0(disable), 1(2-frame), 2(3-frame)
 <enc_type > : 0(H265), 1(H264), 2(MJPEG)
 <enc_bitrate > : Mbps for H265/H264, or Quality for MJPG

sensor list:

0(NULL), 1(imx290), 2(os02k10), 3(ar0237ir),
 4(ov2718), 5(os05a10), 6(imx317), 7(imx335),
 8(f37), 9(ps5268), 10(sc4210), 11(imx307_slave),
 12(f35), 13(imx415), 14(sc500ai), 15(sc401ai),
 16(os04a10), 17(gc4653), 18(imx415_HS), 19(imx678_HS),
 20(os02k10_VPE), 21(os04a10_VPE), 22(os05a10_DUAL), 99(PATTERN_GEN)

说明:

NT98530 demo 板默认搭配 IMX415 HS board，可执行如下指令：

```
pq_video_rtsp 18 => Demo IMX415 4KP60 rtsp
```

```
pq_video_rtsp 18 1 => Demo IMX415 sHDR 4KP60 rtsp
```

连上网络，打开 RTSP client 的 App，输入 rtsp 网络 URL，即可播放 rtsp 流。



板子与主机连接后，也需要修改主机端的 ip(相同网域才能通信)

網際網路通訊協定第 4 版 (TCP/IPv4) - 內容

一般

如果您的網路支援這項功能，您可以取得自動指派的 IP 設定。否則，您必須詢問網路系統管理員正確的 IP 設定。

☐ 自動取得 IP 位址(O)

☒ 使用下列的 IP 位址(S):

IP 位址(I): 192 . 168 . 0 . 1

子網路遮罩(U): 255 . 255 . 255 . 0

預設閘道(D): . . .

☐ 自動取得 DNS 伺服器位址(B)

☒ 使用下列的 DNS 伺服器位址(E):

慣用 DNS 伺服器(P): . . .

其他 DNS 伺服器(A): . . .

☐ 結束時確認設定(L)

進階(V)...

確定 取消

6.2 UVC CAM

UVC CAM 的 sample code 位于如下 SDK 路径：

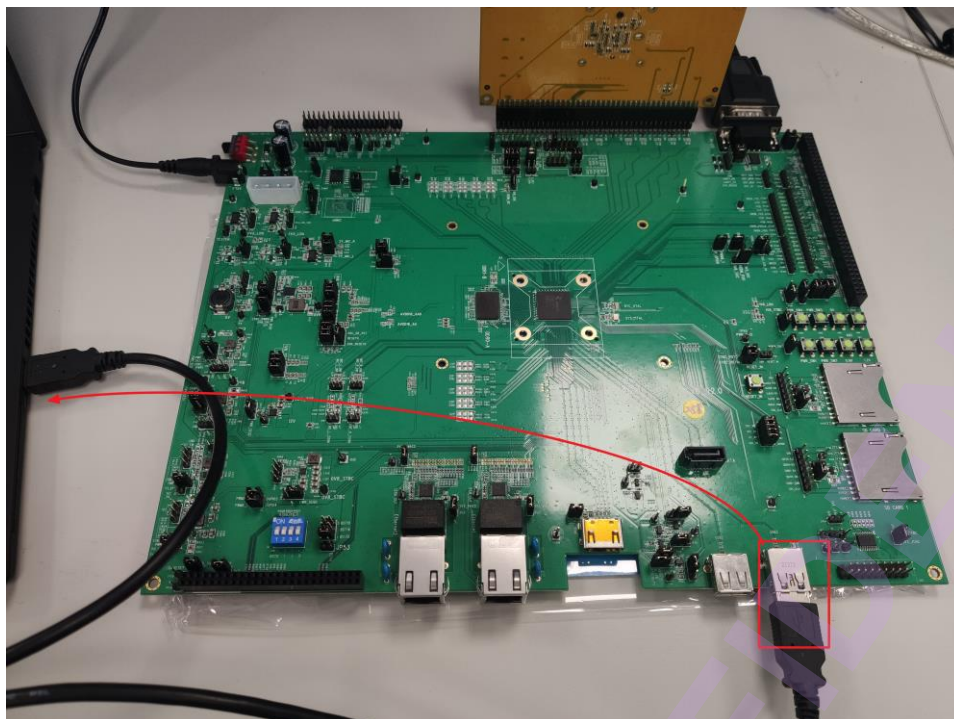
code\application\source\uvc_cam\

默认没有包在 all-in-one 固件里，可自行进到相应目录作编译：

na1102_linux_sdk/code/application/source/uvc_cam\$ make

编译成果物放在 na1102_linux_sdk/code\application\source\uvc_cam\output\hd_uvc_cam

[Demo 单板 USB 线连接]：如下图



使用 UVC CAM 前需要先加载以下 driver: **nvt_usb2dev**, **nvt_usb3dev** 和 **nvt_uvc**

```
root@NVTEVM:~$ modprobe nvt_usb2dev
```

```
[ 31.259388] nvt_usb2dev_module_init:
```

```
[ 31.263176] nvt_usb2dev_probe:2f05a0000.usb20
```

```
[ 31.267556] nvt_usb2dev_probe:0. resource:0x2F05A0000 size:0x00001000
```

```
[ 31.274134] nvt_usb2dev_probe:DevID Major:245 minor:198
```

```
[ 31.279591] usb2dev_power_on_init:[530] data = NULL
```

```
root@NVTEVM:~$ modprobe nvt_usb3dev
```

```
root@NVTEVM:~$ modprobe nvt_uvc
```

```
hd_uvc_cam [sensor_sel] [en_u3_controller].....
```

选项:

sensor_sel: 0 表示 SEN_SEL_IMX290

1 表示 SEN_SEL_OS05A10

2 表示 SEN_SEL_GC4653

3 表示 SEN_SEL_IMX415

4 表示 SEN_SEL_OS02K10

en_u3_controller: 表示是否支持 USB3.0

如 530 demo board 接 IMX415 HS board, 执行 `hd_uvc_cam 3 1`

USB 线连接 demo board 的 USB3.0 port 和 PC 端连接, 可通过 amcap 和 potplayer 等 UVC 播放器接收 UVC 串流。

7 常见问题解答

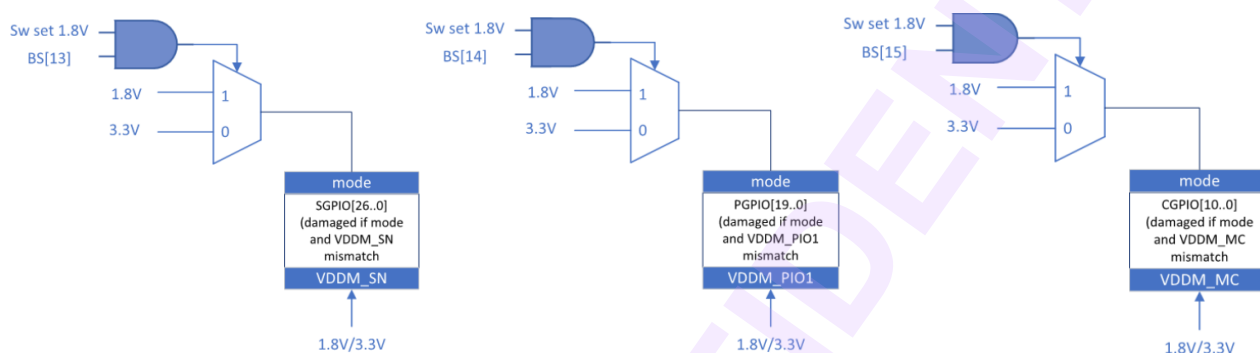
7.1 编译错误：程序中有游离的'#' -e #define NVT_UTS_RELEASE

"4.19.148_"

案例	说明
问题描述	<p>In file included from drivers/soc/nvt/basic/nvt-info.c:22: ./include/generated/nvtversion.h:1:1: 错误： expected identifier or '(' before '-' token -e #define NVT_UTS_RELEASE "4.19.148_" ^ ./include/generated/nvtversion.h:1:4: 错误： 程序中有游离的'#' -e #define NVT_UTS_RELEASE "4.19.148_" ^</p> <p>drivers/soc/nvt/basic/nvt-info.c: 在函数'nvt_version_show'中: drivers/soc/nvt/basic/nvt-info.c:571:34: 错误： 'NVT_UTS_RELEASE' undeclared (first use in this function); did you mean 'NETLINK_URELEASE'? seq_printf(m, "Version: %s \n", NVT_UTS_RELEASE); drivers/soc/nvt/basic/nvt-info.c:571:34: 附注 每个未声明的标识符在其出现的函数内只报告一次</p> <p>make-4.1[5]: *** [drivers/soc/nvt/basic/nvt-info.o] Error 1 make-4.1[4]: *** [drivers/soc/nvt/basic] Error 2 make-4.1[3]: *** [drivers/soc/nvt] Error 2 make-4.1[2]: *** [drivers/soc] Error 2 make[1]: *** [Makefile:1074: drivers] 错误 2 make: *** [Makefile:245: linux] 错误 2</p>
问题分析	<p>SDK 编译环境需要使用 bash 作为默认的 SHELL，但当前编译环境的 SHELL 可能还是 DASH。可使用 ls -la /bin/sh 来确认。</p> <p>\$ ls -la /bin/sh /bin/sh -> dash</p> <p>执行 sudo dpkg-reconfigure dash，在弹出窗口选择 NO，可修改默认 SHELL 为 BASH。</p>

7.2 !!! Stop booting because your power_config dtsi and bootstrap do not match !!!

NT98530 SGPIO[26..0], PGPIO[19..0], CGPIO[10..0] pad power 结构图如下, 只要 mode 跟 VDDM_xxx 不匹配, 就有可能损坏 IC pad, 所以需要 s/w 配合做一定程度的防呆:



说明: BS[14] 在板子接上某些 ethernet phy 时, 会被 phy 那端的 pad pull high, 导致 SOC 端无法 latch BS[14] = low, 因此 PGPIO[19..0] sw 只会做设定, 无法做防呆

需要配合设定 s/w phase 有

- loader: 要修改 Project/Model/ModelConfig_EMU_EVB.txt 裡面的 PAD_POWERID_P1 再重 build loader

```
# [PAD_POWERID_P1]: Select PGPIO[19..0] voltage level
# (Please refer to 530 SDO document hardware\board\document\NT98530_Application_Note_en_V0.1.pdf
section 1.9)
# (No default value is suggested, please match your PCB design!!)
# PAD_1P8V
# PAD_3P3V
PAD_POWERID_P1 = PAD_3P3V
```

- uboot/linux: 要修改 configs/cfg_gen/nvt-peri-dev.dtsi, 把相关的 pad group 设定成跟板子一致

```
&top {
    sgpio_0_26{power_config=<PAD_POWERID_SN PAD_1P8V>;};
    pgpio_0_19{power_config=<PAD_POWERID_P1 PAD_3P3V>;};
    cgpio_0_10{power_config=<PAD_POWERID_SD3 PAD_3P3V>;};
};
```

案例 1	说明
问题描述	WARNING: sgpio_0_26 is <1P8V>, but bootstrap[13] is <3P3V> !!! Stop booting because your power_config dtsi and bootstrap do not match !!!
问题分析	表示 nvt-peri.dtsi 设定 SGPIO[26..0]为 1.8V, 但单板 BS[13]被设定成 3.3V, 请找 project hw 讨论正确设定方式。
案例 2	说明
问题描述	WARNING: sgpio_0_26 is <3P3V>, but bootstrap[13] is <1P8V> !!! Stop booting because your power_config dtsi and bootstrap do not match !!!
问题分析	表示 nvt-peri.dtsi 设定 SGPIO[26..0]为 3.3V, 但单板 BS[13]被设定成 1.8V, 请找 project hw 讨论正确设定方式。
案例 3	说明
问题描述	WARNING: cgpio_0_10 is <1P8V>, but bootstrap[15] is <3P3V> !!! Stop booting because your power_config dtsi and bootstrap do not match !!!
问题分析	表示 nvt-peri.dtsi 设定 CGPIO[11..0]为 1.8V, 但单板 BS[15]被设定成 3.3V, 请找 project hw 讨论正确设定方式。
案例 4	说明
问题描述	WARNING: cgpio_0_10 is <3P3V>, but bootstrap[15] is <1P8V> !!! Stop booting because your power_config dtsi and bootstrap do not match !!!
问题分析	表示 nvt-peri.dtsi 设定 CGPIO[11..0]为 3.3V, 但单板 BS[15]被设定成 1.8V, 请找 project hw 讨论正确设定方式。