版本: 1.1.0　2022 年 01 月

AnyCloud 平台 SDK 用户开发手册

声　　明

联　系　方　式

广州安凯微电子股份有限公司

地址：广州市黄埔区知识城博文路 107 号安凯微电子 H 大厦

电话: (86)-20-3221 9000

传真: (86)-20-3221 9258

邮编: 510555

销售热线:

(86)-20-3221 9499

电子邮箱:

sales@anyka.com

主页:

http: //www. anyka.com

目 录

1 前言

本文档为使用安凯 AnyCloud 平台的程序员而写，目的是提供 AnyCloud 平台 SDK 的使用方法。

说明：本文档适用于安凯 AnyCloud 平台所支持的所有系列芯片，并会随 AnyCloud 系统平台的发布而更新，请注意及时更新该文档。

2 版本变更说明

V1.1.0 版本修订说明

发布时间 2022 年 1 月，包括以下修改：

- 【新增】增加 AnyCloud39AV100 平台说明。

V1.0.4 版本修订说明

发布时间 2021 年 10 月，包括以下修改：

- 【更新】SDK 解压后的目录有更新。

V1.0.3 版本修订说明

发布时间 2021 年 8 月，包括以下修改：

- 【新增】针对 37E 芯片，更新视频配置相关的 dtb 说明。

V1.0.2 版本修订说明

发布时间 2021 年 5 月，包括以下修改：

- 【新增】针对 37E 芯片，更新新增的 dtb 说明。

V1.0.1 版本修订说明

发布时间 2020 年 12 月，包括以下修改：

- 【更新】平台用户开发手册把 U-Boot 编译的具体内容拆分成独立文档《Anycloud 平台 U-Boot 使用说明》。
- 【更新】工具链压缩包命名更新为 arm-anykav500-linux-uclibcgnueabi_V1.0.04.tar.bz2。

V1.0.0 版本修订说明

发布时间 2019 年 12 月，包括以下修改：

- 【更新】平台用户开发手册把中间件的接口说明拆分成独立的文档《平台中间件接口说明》。

3 平台简介

AnyCloud 平台是基于 AnyCloud 系列芯片的高清数字网络/楼宇对讲产品开发解决方案。AnyCloud 平台进行了模块化设计，用户可以通过调用各模块接口实现视频编解码、音频采集和播放、音频编解码等功能。

与本文档相对应的产品版本如下，支持的操作系统均为 Linux 操作系统。

芯片简称	系统平台	功能概要
AK39EV330L	AnyCloud39EV330	视频采集、视频编码、图像处理、音频采集和播放、音频编解码等
AK37D	AnyCloud37D	视频采集与播放、视频编解码、图像处理、音频采集与播放、音频编解码等
AK37E	AnyCloud37E	视频采集与播放、视频编解码、音频采集与播放、音频编解码等

芯片简称	系统平台	功能概要
AK3918AV100	AnyCloud39AV100	视频采集、视频编码、图像处理、音频采集与播放、音频编解码、智能视频处理等

4 环境准备

4.1 安装 SDK 包

平台 SDK 包位于 PDK_Vx.xx/SDK/目录下，文件名为 AKxxx_SDK_Vx.xx.tar.gz。

不同芯片对应的 SDK 压缩包名称如下：

芯片简称	系统平台	SDK压缩包名称
AK39EV330L	AnyCloud39EV330	AK39EV33X_SDK_Vx.xx.tar.gz
AK37D	AnyCloud37D	AK37D_SDK_Vx.xx.tar.gz
AK37E	AnyCloud37E	AK37E_SDK_Vx.xx.tar.gz
AK3918AV100	AnyCloud39AV100	AK3918AV100_SDK_Vx.xx.tar.gz

将 SDK 包安装到 Linux 服务器中的步骤如下：

- 1、拷贝。将 AKxxx_SDK_Vx.xx.tar.gz 拷贝到 Linux 服务器上。
- 2、执行解压命令：tar -zxf AKxxx_SDK_Vx.xx.tar.gz

- 3、等待命令执行完毕。解压出 AKxxx_SDK_Vx.xx 目录，结构如下所示。

```
├── auto_build.sh
├── ChangeLog.md
├── config.mk
├── image
├── make_image.sh
├── os
│   ├── build_kernel.sh
│   ├── build_uboot.sh
│   ├── config.mk
│   ├── driver
│   ├── driver_src.tar.gz
│   ├── linux.tar.gz
│   └── uboot.tar.gz
├── platform
│   ├── include
│   ├── lib
│   └── sample
├── QuickStartGuide.md
├── rootfs
│   ├── create_jffs2fs.sh
│   ├── create_squashfs.sh
│   ├── create_yaffs2fs.sh
│   ├── extract.sh
│   ├── Makefile
│   ├── mkfs.jffs2
│   ├── mksquashfs
│   ├── mkyaffs2image
│   ├── resource
│   ├── rootfs.lib.tar.gz
│   ├── rootfs.softlink.tar.gz
│   ├── rootfs.tar.gz -> rootfs.softlink.tar.gz
│   ├── scripts
│   ├── utils
│   └── wifi
├── rules.mk
└── tools
    ├── arm-any*-linux-uclibcgnueabi_V1.x.xx.tar.bz2
    ├── burntool
    ├── busybox-1.30.1.tar.bz2
    ├── envtool
    ├── gdbtool
    ├── mkuboot
    └── uart_burntool
```

4.2 编译服务器环境搭建

编译环境选择 Ubuntu16.04 64bit 版本，建议选择默认配置安装，需要额外安装以下软件工具，可以使用 apt-get install 命令安装或者选择源码包编译安装：

- lib32ncurses5
- lib32z1
- u-boot-tools
- libstdc++6
- libncurses5-dev
- liblzo2-dev:i386，
- mtd-utils
- liblzma-dev:i386
- bison
- flex

4.3 搭建开发环境

安凯提供编译好的工具链，用户直接解压即可。安装工具链，配置环境变量以及确认工具链具体步骤如下：

步骤一：安装工具链

将 arm-any*-linux-uclibcgnueabi_V1.x.xx.tar.bz2 拷贝至有读写权限的目录下，执行以下命令进行解压：

```
sudo tar jxvf arm-any*-linux-uclibcgnueabi_V1.x.xx.tar.bz2
```

解压后得到 arm-any*-linux-uclibcgnueabi，把 arm-any*-linux-uclibcgnueabi 这个工具链拷贝到 opt 目录下。

说明：此处的“opt”目录，用户可依据实际情况替换为自己的目标目录。

步骤二：配置主机 PATH 环境变量

AnyCloud37E 平台提供以下两种配置环境变量方法：

- 方法一

在 opt 目录下，添加以下命令到系统的 PATH 变量。

```
export PATH=$PATH:/opt/arm-any*-linux-uclibcgnueabi/bin
```

说明：此处的”opt” 目录，用户可依据实际情况替换为用户自己的目标目录。该方法只在当前会话有效，用户重新登录或者重新启动该方法即失效。

- 方法二

修改/etc/environment 文件， /etc/environment 的修改如下：

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/
usr/local/games:/opt/arm-any*-linux-uclibcgnueabi/bin"
```

说明：用户需要退出、重新登录该方法才能生效，并且永久有效。

步骤三：工具链确认

执行以下命令检测工具链是否安装成功：

```
arm-any*-linux-uclibcgnueabi-gcc  -v
```

如果系统能找到命令并显示 gcc 的版本信息，则工具安装成功。否则，安装失败。

注意：工具链基于 64 位的操作系统，不支持 32 位操作系统。

5 编译

5.1 一键编译

用户可运行 ./auto_build.sh 实现一键编译，使用一键编译需要用户手动配置好 AKxxx_SDK_Vx.xx 目录下的 config.mk 文件，用户需要根据自己目前使用的硬件进行配置。

运行 auto_build.sh 脚本即可一键完成内核、U-Boot 以及 Platform 编译，并将生成的镜像文件拷贝到 burntool 目录下。

AnyCloud39AV100 平台需要注意以下事项：

注意： config.mk 文件中的 CHIP_TYPE 要选对，否则会影响到 PHY 功能。

 # 芯片类型定义(AK3918AV100N, AK3918AV100P, AK3918AV100X, AK3918AV100SVT)

CHIP_TYPE = AK3918AV100N

AK3918AV100N 和 AK3918AV100P 是 64M 的，AK3918AV100X 是 128M 的。

 # 内存大小（64M，128M）

DRAM = 64M

一键编译后，会把所有的 dtb 文件都拷贝到烧录工具目录，用户需要根据当前的硬件配置，选择对应的 dtb 文件，然后手动将对应的 dtb 重命名成 cloudOS.dtb。

例如：使用的是 EVB_CBDR_AK3760E_V1.0.1.dtb 这个 dts，在烧录的时候需要把 burntool 下面的 EVB_CBDR_AK3760E_V1.0.1.dtb 重命名为 cloudOS.dtb，才能正确烧录，否则 burntool 会提示没有找到 cloudOS.dtb 的错误。

AnyCloud37E 的 dtb 文件说明

对于 SPI NOR Flash 或 SPI NOR Flash+SPI NAND Flash，对应 dtb 配置文件如表 5-1 所示。

表 5-1 AnyCloud37E SPI NOR Flash 配置文件对应信息

芯片型号	屏幕类型	是否有视频采集	DTS配置文件
AK3760E	RGB	否	EVB_CBDR_AK3760E_V1.0.1.dtb
AK3760E	RGB	是	EVB_CBDR_AK3760E_V1.0.1_camera.dtb
AK3760E	MIPI	否	EVB_CBDM_AK3760E_V1.0.1.dtb

芯片型号	屏幕类型	是否有视频采集	DTS配置文件
AK3760E	MIPI	是	EVB_CBDM_AK3760E_V1.0.1_camera.dtb
AK3760E	RGB	是	EVB_CBDR_AK3760E_V1.0.1_camera_TP9950.dtb
AK3760E	MIPI	是	EVB_CBDM_AK3760E_V1.0.1_camera_TP9950.dtb
AK3760E	MPU	否	EVB_CBDR_AK3760E_V1.0.1_mpu.dtb
AK3760E	MIPI	否	EVB_CBDM_AK3760E_V1.0.1_I2S.dtb

对于 SPI NAND Flash，生对应 dtb 配置文件如表 5-2 所示。

表 5-2 AnyCloud37E SPI NAND Flash 配置文件对应信息

芯片型号	屏幕类型	是否有视频采集	DTS配置文件
AK3760E	RGB	否	EVB_CBDR_AK3760E_V1.0.1_spinand.dtb
AK3760E	RGB	是	EVB_CBDR_AK3760E_V1.0.1_spinand_camera.dtb

芯片型号	屏幕类型	是否有视频采集	DTS配置文件
AK3760E	MIPI	否	EVB_CBDM_AK3760E_V1.0.1_spinand.dtb
AK3760E	MIPI	是	EVB_CBDR_AK3760E_V1.0.1_spinand_camera.dtb

对于 dts 的使用注意事项如下：

- 视频采集的硬件接口和 MAC1 管脚复用，如果有视频采集，就不能用 MAC1，有 camera 功能的 dtb 命名为 EVB_CBDM_AK3760E_Vx.x.x_camera.dtb。
- 带 camera_TP995 后缀的 dtb 文件是带有 sensor TP995 的配置。带 mpu 后缀的是 mpu 屏显示的 dtb 配置。
- 仅 MIPI 核心板支持 I2S，如 EVB_CBDM_AK3760E_V1.0.1_I2S.dtb。
- PDM 功能仅 MIPI 核心板支持，如 EVB_CBDM_AK3760E_V1.0.1.dtb，PDM 和 SD 卡有管脚复用。
- standby 功能 RGB 和 MIPI 核心板 dtb 配置都支持，如：EVB_CBDR_AK3760E_V1.0.1.dtb 和 EVB_CBDM_AK3760E_V1.0.1.dtb。
- MPU 屏编译的时候在 config.mk 配置里选择 RGB 屏，烧录时 dtb 选择 EVB_CBDR_AK3760E_V1.0.1_mpu.dtb。

AnyCloud37D 的 dtb 文件说明

对于 SPI NOR Flash，生成的常用 dtb 配置文件如表 5-3 所示。

表 5-3 AnyCloud37D SPI NOR Flash 配置文件对应信息

芯片型号	屏幕类型	DTS配置文件
AK3760D	MIPI	c500_cbd_ak3760d_dsi_v1.0.0.dtb
AK3760D	RGB	c500_cbd_ak3760d_rgb_v1.0.0.dtb
AK3760D	MPU	c500_cbd_ak3760d_mpu_v1.0.0.dtb
AK3761D	MIPI	c500_cbd_ak3761d_dsi_v1.0.0.dtb
AK3761D	MIPI	c500_cbd_ak3761d_dsi_v1.0.0_dual.dtb（双目）
AK3761D	RGB	c500_cbd_ak3761d_rgb_v1.0.0.dtb
AK3761D	RGB	c500_cbd_ak3761d_rgb_v1.0.0_dual.dtb（双目）
AK3761D	MPU	c500_cbd_ak3761d_mpu_v1.0.0.dtb

对于 SPI NAND Flash，生成的常用 dtb 配置文件如表 5-4 所示。

表 5-4 AnyCloud37D SPI NAND Flash 配置文件对应信息

芯片型号	屏幕类型	DTS配置文件
AK3760D	MIPI	c500_cbd_ak3760d_dsi_v1.0.0_spinand.dtb
AK3760D	RGB	c500_cbd_ak3760d_rgb_v1.0.0_spinand.dtb
AK3760D	MPU	c500_cbd_ak3760d_mpu_v1.0.0_spinand.dtb
AK3761D	MIPI	c500_cbd_ak3761d_dsi_v1.0.0_spinand.dtb
AK3761D	MIPI	c500_cbd_ak3761d_dsi_v1.0.0_spinand_dual.dtb（双目）
AK3761D	RGB	c500_cbd_ak3761d_rgb_v1.0.0_spinand.dtb
AK3761D	RGB	c500_cbd_ak3761d_rgb_v1.0.0_spinand_dual.dtb（双目）
AK3761D	MPU	c500_cbd_ak3761d_mpu_v1.0.0_spinand.dtb

AnyCloud39EV330 的 dtb 文件说明

对于 SPI NOR Flash，生成的常用 dtb 配置文件如表 5-5 所示。

表 5-5 AnyCloud39EV330 SPI NOR Flash 配置文件对应信息

芯片型号	DTS配置文件
AK3916EV330L	EVB_CBD_AK3916EV330_V1.0.0.dtb
AK3916EV331L	EVB_CBD_AK3916EV331_V1.0.0.dtb
AK3916EV331L	EVB_CBD_AK3916EV331_V1.0.0_dual.dtb（双目）
AK3918EV330L	EVB_CBD_AK3918EV330_V1.0.0.dtb
AK3918EV331L	EVB_CBD_AK3918EV331_V1.0.0_dual.dtb
AK3919EV330L	EVB_CBD_AK3919EV330_V1.0.0.dtb
AK3919EV331L	EVB_CBD_AK3919EV331_V1.0.0.dtb

对于 SPI NAND Flash，生成的常用 dtb 配置文件如表 5-6 所示。

表 5-6 AnyCloud39EV330 SPI NAND Flash 配置文件对应信息

芯片型号	DTS配置文件
AK3916EV330L	EVB_CBD_AK3916EV330_V1.0.0_spinand.dtb
AK3916EV331L	EVB_CBD_AK3916EV331_V1.0.0_spinand.dtb
AK3916EV331L	EVB_CBD_AK3916EV331_V1.0.0_sp

芯片型号	DTS配置文件
	inand_dual.dtb（双目）
AK3918EV330L	EVB_CBD_AK3918EV330_V1.0.0_spinand.dtb
AK3918EV331L	EVB_CBD_AK3918EV331_V1.0.0_spinand_dual.dtb
AK3919EV330L	EVB_CBD_AK3919EV330_V1.0.0_spinand.dtb
AK3919EV331L	EVB_CBD_AK3919EV331_V1.0.0_spinand_dual.dtb

AnyCloud39AV100 的 dtb 文件说明

对于 SPI NOR Flash，生成的常用 dtb 配置文件如表 5-7 所示。

表 5-7 AnyCloud39AV100 SPI NOR Flash 配置文件对应信息

芯片型号	DTS 配置文件
AK3918AV100N	EVB_CBDM_AK3918AV100N_V1.0.0.dtb
AK3918AV100P	EVB_CBDM_AK3918AV100P_V1.0.0.dtb
AK3918AV100X	EVB_CBDM_AK3918AV100X_V1.0.0.dtb

5.2 手动编译

5.2.1 内核编译

AnyCloud 平台提供以下两种手动内核编译方法：

5.2.1.1 手动编译方法一

- 进入 os 目录，执行以下命令编译 kernel：

```
./build_kernel.sh
```

注意：build_kernel.sh 需要依赖 PDK_Vx.xx/SDK/AKxxx_SDK_Vx.xx 目录下面的 config.mk，如果 config.mk 没有配置好会影响后续的编译、使用。

- 可以通过以下命令自动将编译好的 uImage、dtb、驱动拷贝到 config.mk 中指定的目录位置。

```
./build_kernel.sh -i
```

也可以手动将编译好的 uImage、dtb、驱动拷贝到 config.mk 中指定的目录位置。

5.2.1.2 手动编译方法二

步骤一：进入 os 目录下，解压 linux.tar.gz 内核源码到 kernel 目录。

步骤二：在 kernel 目录下执行以下命令，创建文件夹 build。

```
mkdir ../build
```

说明：如果用户需要制作 dts 文件，在上述步骤 2 后，执行以下操作：将新增的 dts 文件拷贝至内核的 arch/arm/boot/dts 目录下，并在 arch/arm/boot/dts/Makefile 中增加以下代码，不同的平台增加的代码不一样。

AnyCloud37E 平台增加以下代码：

```
dtb-$(CONFIG_MACH_AK37E) += board-name.dtb
```

AnyCloud37D 平台增加以下代码：

```
dtb-$(CONFIG_MACH_AK37D) += board-name.dtb
```

AnyCloud39EV330 平台增加以下代码：

```
dtb-$(CONFIG_MACH_AK39EV330) += board-name.dtb
```

AnyCloud39AV100 平台增加以下代码：

```
dtb-$(CONFIG_MACH_AK3918AV100) += board-name.dtb
```

步骤三：执行命令配置内核板级文件。

AnyCloud37E 平台核心板执行以下命令配置板级文件：

```
make O=../build anycloud_ak37e_mini_defconfig CROSS_COMPILE=arm-anykav500-linux
-uclibcgnueabi-
```

AnyCloud37D 平台核心板执行以下命令配置板级文件：

```
make O=../build anycloud_ak37d_mini_defconfig CROSS_COMPILE=arm-anykav500-linux
-uclibcgnueabi-
```

AnyCloud39EV330 平台核心板执行以下命令配置板级文件：

```
make O=../build anycloud_ak39ev33x_mini_defconfig CROSS_COMPILE=arm-anykav500-l
inux-uclibcgnueabi-
```

AnyCloud39AV100 平台核心板执行以下命令配置板级文件：

```
make O=../build anycloud_ak3918av100_mini_defconfig CROSS_COMPILE=arm-anykav500
-linux-uclibcgnueabi-
```

步骤四：执行以下命令编译 uImage、dtb 文件。

```
make O=../build dtbs modules uImage -j8 CROSS_COMPILE=arm-anykav500-linux-uclib
cgnueabi-
```

执行上述命令后，系统将生成以下两组文件：

1）在../build/arch/arm/boot 路径下生成 uImage 内核镜像文件。

2）在../build/arch/arm/boot/dts 路径下生成 dtb 配置文件。

用户需根据芯片型号和屏幕类型，选择相应的 dtb 配置，并将配置文件名重命名为 cloudOS.dtb，手动拷贝到 AKxxx_SDK_Vx.xx/tools/burntool 目录。

步骤五：拷贝驱动文件到 internal 目录下， 需要执行以下命令：

```
make modules_install O=../build INSTALL_MOD_PATH=../../rootfs/ko/internal
```

相对应的内核驱动会被安装到 AKxxx_SDK_Vx.xx/rootfs/ko/internel/目录中。

5.2.2 U-Boot 编译

U-Boot 编译 的具体内容请参见《Anycloud 平台 U-Boot 使用说明》。

AnyCloud39AV100 平台需要注意以下事项：

注意： U-Boot 编译时，要参考 build_uboot.sh 脚本，DEVICE_TREE 需要区分芯片型号 AK3918AV100N 和 AK3918AV100P、AK3918AV100X。

make all DEVICE_TREE = EVB_CBD_AK3918AV100N_V1.0.0 - j$num_of_cpu_cores CROSS_COMPILE = $CROSS_COMPILE

5.2.3 平台编译

平台编译步骤如下：

- 步骤一：把 driver 目录下的 externel 目录拷贝至 rootfs/ko 目录下。
- 步骤二：在 AKxxx_SDK_Vx.xx 目录下， 执行以下命令：

```
./make_image.sh
```

执行上述命令后，用户需将生成的一组镜像文件（如 表 5-8 与表 5-9 所示）拷贝到 burntool 目录下。

- 对于 SPI NOR Flash，生成的镜像文件如表 5-8 所示。

表 5-8 SPI NOR Flash 镜像文件

镜像名称	功能描述	访问权限
root.sqsh4	Linux标准的根文件系统。	只读
usr.sqsh4	挂载到/usr/目录，主要包含内容：平台相关的脚本，平台中间件各个模块的库，ISP相关配置，以模块形式加载的驱动程序*.ko。	只读
usr.jffs2	挂载到/etc/config/目录，主要包含：针对系统的配置密码及其他配置信息。	读写

- 对于 SPI NAND Flash， 生成的镜像文件如表 5-9 所示。

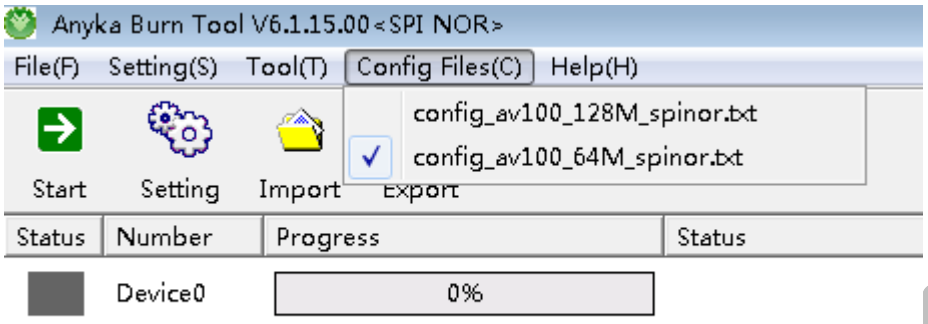
表 5-9 SPI NAND Flash 镜像文件

镜像名称	功能描述	访问权限
rootfs.yaffs2	Linux标准的根文件系统。	只读
usr.yaffs2	挂载到/usr/目录，主要包含内容：平台相关的脚本，平台中间件各个模块的库，ISP相关配置，以模块形式加载的驱动程序*.ko。	只读
config.yaffs2	挂载到/etc/config/目录，主要包含：针对系统的配置密码及其他配置信息。	读写

6 烧录

AnyCloud 平台提供两种烧录工具：BurnTool 和 Uart_BurnTool，具体使用方法请参见《AnyCloud 平台 Uart_BurnTool 使用说明》和《AnyCloud 平台 BurnTool 使用说明》。

对于 AnyCloud39AV100 平台，AK3918AV100N 和 AK3918AV100P 芯片是 64M 内存的，烧录时请选 64M 的 config 配置，如下图。



AK3918AV100X 芯片是 128M 内存的，烧录时请选 128M 的 config 配置。

7 中间件接口调用

AnyCloud 平台 SDK 包提供一套支持音视频采集、显示、播放及编解码等功能的中间件接口，存放于 SDK 包中的 AKxxx_SDK_Vx.xx/platform 目录下，供上层应用接口调用，以实现音视频应用的快速开发和集成。中间件接口的使用方法参见文档《平台中间件接口使用说明》。

8 系统启动和运行

8.1 系统启动流程

系统启动流程如图 8-1 所示：

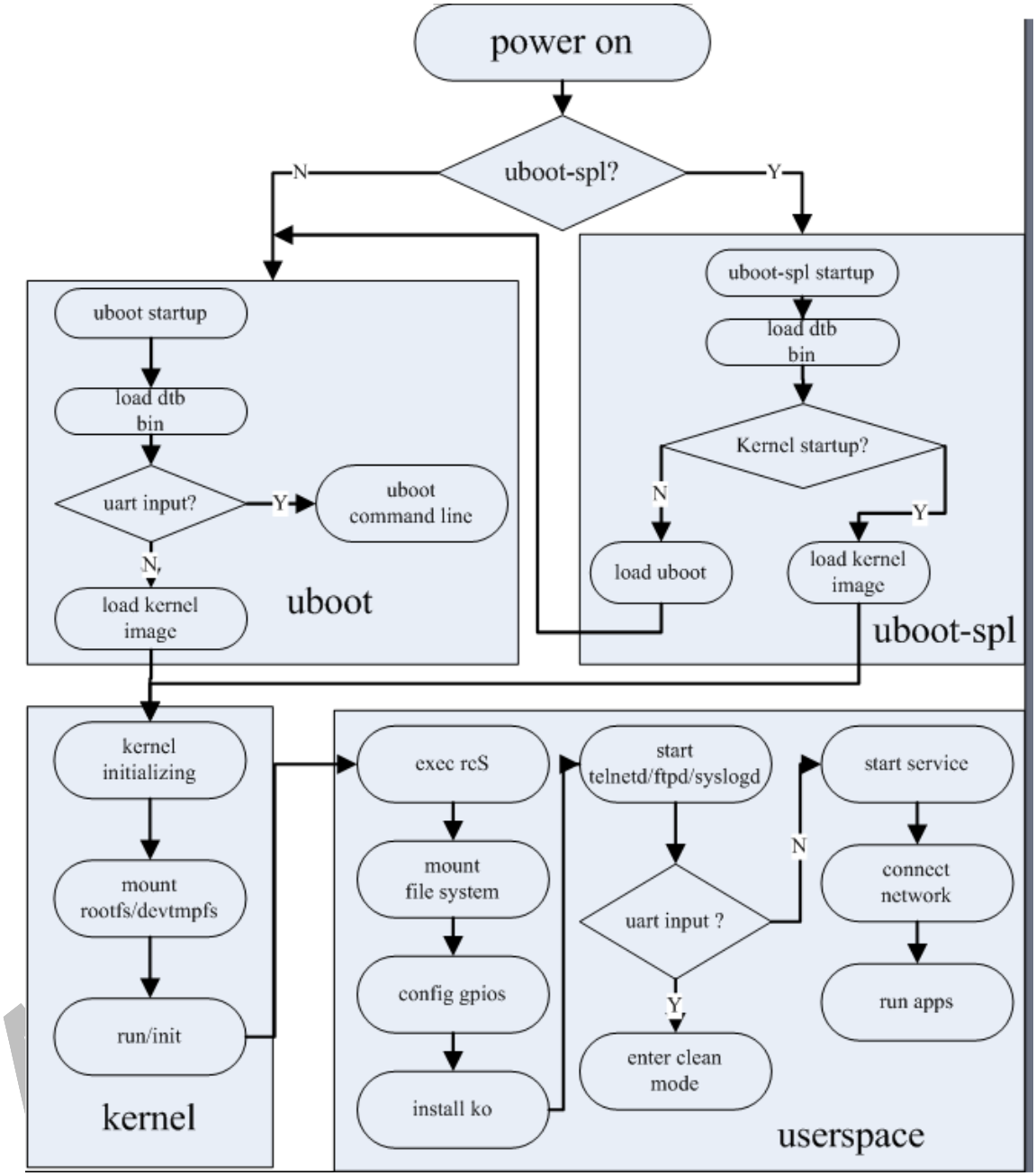


图 8-1 系统启动流程图

8.2 从串口信息看系统启动流程

系统上电启动后，会显示串口信息，现以 AnyCloud37E 平台举例说明系统启动步骤如下：

第一步：运行 U-Boot

系统上电后，shell 会话显示如下信息，表示 U-Boot 启动完成。

```
U-Boot 2013.10.0-V3.1.31 (Jul 29 2021 - 17:17:28)

uboot display lcd logo!
DRAM:  64 MiB
8 MiB
sd detect gpio mode:32!
mmc_sd: 0
asic pll is 500MHz!
In:    serial
Out:   serial
Err:   serial
Net:   eth-0

Hit any key to stop autoboot:  1
```

第二步：加载 Linux Kernel，解压并准备启动 Kernel

Shell 会话显示如下信息，表示 Kernel 启动完成。

```
KERNEL: size:0x00190000, offset:0x00049000

SF: 1638400 bytes @ 0x49000 Read: OK
## Booting kernel from Legacy Image at 80008000 ...
   Image Name:   Linux-4.4.192V2.4
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    1617208 Bytes = 1.5 MiB
   Load Address: 80008000
   Entry Point:  80008040
   Verifying Checksum ... OK
   XIP Kernel Image ... OK
   kernel loaded at 0x80008000, end = 0x80192d38
using: FDT
```

第三步：启动 Linux Kernel，初始化运行环境

Shell 会话显示如下信息，表示运行环境初始化完成。

```
Starting kernel ...

Uncompressing Linux... done, booting the kernel.
Booting Linux on physical CPU 0x0
Linux version 4.4.192V2.4 (songmengxing@anyka-PowerEdge-R440) (gcc version 4.9.
4 (Buildroot 2018.02.7_V1.0.03-g9ff3371) ) #1 Thu Jul 29 17:17:22 CST 2021
CPU: ARM926EJ-S [41069265] revision 5 (ARMv5TEJ), cr=0005317f
CPU: VIVT data cache, VIVT instruction cache
Machine model: EVB_CBDM_AK3760E_V1.0.1 board
Reserved memory: created CMA memory pool at 0x83000000, size 16 MiB
```

```
Reserved memory: initialized node cma_reserved@0x83000000, compatible id shared
-dma-pool
Memory policy: Data cache writeback
ANYKA CPU AK37XXE (ID 0x201902)
Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 16256
Kernel command line: console=ttySAK0,115200n8 root=/dev/mtdblock6 rootfstype=sq
uashfs init=/sbin/init mtdparts=spi0.0:220K@0x0(UBOOT),4K@0x37000(ENV),4K@0x380
00(ENVBK),64K@0x39000(DTB),1600K@0x49000(KERNEL),300K@0x1D9000(LOGO),2200K@0x22
4000(ROOTFS),300K@0x44a000(CONFIG),3500K@0x495000(APP) mem=64M memsize=64M
PID hash table entries: 256 (order: -2, 1024 bytes)
Dentry cache hash table entries: 8192 (order: 3, 32768 bytes)
Inode-cache hash table entries: 4096 (order: 2, 16384 bytes)
Memory: 23048K/65536K available (3170K kernel code, 126K rwdata, 1024K rodata,
140K init, 189K bss, 26104K reserved, 16384K cma-reserved)
Virtual kernel memory layout:
    vector  : 0xffff0000 - 0xffff1000   (   4 kB)
    fixmap  : 0xffc00000 - 0xfff00000   (3072 kB)
    vmalloc : 0xc4800000 - 0xff800000   ( 944 MB)
    lowmem  : 0xc0000000 - 0xc4000000   (  64 MB)
    modules : 0xbf000000 - 0xc0000000   (  16 MB)
      .text : 0xc0008000 - 0xc0420eb4   (4196 kB)
      .init : 0xc0421000 - 0xc0444000   ( 140 kB)
      .data : 0xc0444000 - 0xc0463a20   ( 127 kB)
       .bss : 0xc0463a20 - 0xc0492f78   ( 190 kB)
SLUB: HWalign=32, Order=0-3, MinObjects=0, CPUs=1, Nodes=1……
```

8.3 登录用户名和密码

系统完成上电启动后进入平台调试状态，登录用户名为 **root**，密码为空。

8.4 运行目录结构

设备正常运行后，用户将会看到如下所示的运行目录机构。

```
|-- bin
|-- data
|-- dev
|-- etc
|   |-- Version
|   |-- config
|   |-- init.d
|   |-- inittab
|   |-- profile
```

```
|   `|-- sysconfig
|-- lib
|   `-- modules
|-- linuxrc -> bin/busybox
|-- mnt
|-- proc
|-- sbin
|-- sys
|-- tmp
|-- usr
|   |-- bin
|   |-- lib
|   |-- modules
|   |-- sbin
|   `-- share
|       `-- udhcpc
`-- var
    |-- cache
    |-- log
    |-- run
    `-- spool
```

8.5 平台层次模块说明

8.5.1 脚本

平台相关的所有脚本大部分存放在/usr/sbin/目录下，从/etc/init.d/rc.local 开始，再启动文件系统、应用程序、网络等相关功能。相关脚本文件的用途如表 8-1 所示。

表 8-1 脚本文件功能描述

脚本名称	功能描述
rc.local	挂载usr文件系统、挂载jffs2文件系统、启动本地环网、启动main.sh脚本。
main.sh	设置内核打印等级、启动ftp服务器、启动telnet服务器、启动日志记录、加载必要的内核驱动、配置有线网络MAC地址、启动service.sh脚本。

脚本名称	功能描述
service.sh	用户程序入口。
update.sh	升级脚本，支持普通固件升级。
tf_card.sh	加载/卸载TF卡相关驱动。
nfs_start.sh	加载nfs文件系统相关。
wifi_driver.sh	WiFi启动停止脚本。
sar-adc.sh	SAR ADC加载/卸载脚本，若使用模拟光敏，需加载此脚本。
key.sh	按键加载/卸载脚本。

8.5.2 库文件

/usr/lib 库文件包括的 So 动态库如表 8-2 所示。

表 8-2 so 动态库文件

so库名称	说明	支持芯片
libakaudiofilter.so	音效处理库	AK37E、AK37D、AK39EV330L、AK3918AV100
libakaudiocodec.so	音频编解码库	AK37E、AK37D、AK39EV330L、AK3918AV100
libakv_encode.so	视频编码库	AK37E、AK37D、AK39EV330L

so库名称	说明	支持芯片
libakv_decode.so	视频解码库	AK37E、AK37D
libakmedia.so	多媒体库	AK37E、AK37D、AK39EV330L、AK3918AV100
libmpi_adec.so	音频解码库	AK37E、AK37D、AK39EV330L、AK3918AV100
libmpi_aenc.so	音频编码库	AK37E、AK37D、AK39EV330L、AK3918AV100
libmpi_mux.so	录像合成库	AK37E、AK37D、AK39EV330L、AK3918AV100
libmpi_demux.so	录像频解析库	AK37E、AK37D、AK39EV330L、AK3918AV100
libplat_ai.so	音频输入库	AK37E、AK37D、AK39EV330L、AK3918AV100
libplat_ao.so	音频输出库	AK37E、AK37D、AK39EV330L、AK3918AV100
libplat_common.so	公共服务库	AK37E、AK37D、AK39EV330L、

so库名称	说明	支持芯片
		AK3918AV100
libplat_dbg.so	模块信息输出管理库	AK37E、AK37D、AK39EV330L、AK3918AV100
libplat_log.so	打印信息管理库	AK37E、AK37D、AK39EV330L、AK3918AV100
libplat_mem.so	内存管理库	AK37E、AK37D、AK39EV330L、AK3918AV100
libplat_osal.so	操作系统适配模块库	AK37E、AK37D、AK39EV330L、AK3918AV100
libplat_thread.so	多线程操作库	AK37E、AK37D、AK39EV330L、AK3918AV100
libplat_vqe.so	声音质量增强模块库	AK37E、AK37D、AK39EV330L、AK3918AV100
libplat_vo.so	视频输出库	AK37E、AK37D
libplat_tde.so	图像二维处理库	AK37E、AK37D
libapp_ats.so	音频调试工具服务库	AK37E、AK37D、AK39EV330L、AK3918AV100

so库名称	说明	支持芯片
libplat_timer	音频调试工具timer适配库	AK37E、AK37D、AK39EV330L、AK3918AV100
libplat_vi.so	视频采集模块库	AK37E、AK37D、AK39EV330L、AK3918AV100
libmpi_venc.so	视频编码模块库	AK37E、AK37D、AK39EV330L、AK3918AV100
libmpi_vdec.so	视频解码模块库	AK37E、AK37D
libplat_uuid.so	获取芯片唯一ID模块	AK37E、AK3918AV100
libplat_vpss.so	视频处理子系统库	AK37D、AK39EV330L、AK3918AV100
libapp_its.so	ISP工具服务库	AK37D、AK39EV330L、AK3918AV100
libapp_md.so	移动侦测库	AK37D、AK39EV330L、AK3918AV100
libak_mrd.so	移动侦测算法库	AK37D、AK39EV330L、AK3918AV100
libakv_cnn.so	神经网络算法库	AK3918AV100
libmpi_svp.so	智能视觉处理库	AK3918AV100

8.5.3 内核驱动程序

在 Linux 4.4.192 内核版本上完成对 AnyCloud 系列芯片的移植，为应用层提供 AnyCloud 芯片的相关功能。/usr/modules 中包括如下表所示的 Ko 文件。

KO文件名称	说明	支持芯片
ak_adc_key.ko	使用数模转换实现多按键检测的驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
ak_gpio_keys.ko	GPIO按键驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
ak_efuse.ko	Efuse模块驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
ak_eth.ko	有线网卡的驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
ak_i2c.ko	TWI总线的驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
ak_ion.ko	内存管理器的驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
ak_leds.ko	LED灯的驱动	AK37E、AK37D、AK39EV330L、AK3918AV100

KO文件名称	说明	支持芯片
ak_mci.ko	基于MMC/SD/SDIO接口的设备的驱动，例如TF卡、SDIO WiFi等	AK37E、AK37D、AK39EV330L、AK3918AV100
ak_motor.ko	电机控制驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
ak_pcm.ko	音频数据录制和播放的驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
ak_pwm_char.ko	PWM模块的驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
ak_rtc.ko	RTC模块的驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
ak_saradc.ko	模数转换模块的驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
ak_uio.ko	用户空间的I/O驱动，视频编码和解码时使用	AK37E、AK37D、AK39EV330L、AK3918AV100
ak_hcd.ko	USB主机控制器驱动	AK37E、AK37D、AK39EV330L、AK3918AV100

KO文件名称	说明	支持芯片
ak_udc.ko	USB设备控制器驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
rtl8188ftv.ko	rtl8188ftv的WiFi驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
rtl8189ftv.ko	rtl8189ftv的WiFi驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
atbm6031.ko	atbm6031的WiFi驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
atbm6032.ko	atbm6032的WiFi驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
ak_ramdisk.ko	内存作为块设备驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
sensor_XXX.ko	Sensor驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
ak_image_capture.ko	视频采集驱动	AK37E
ts_icn85xx.ko	rgb触摸屏驱动	AK37E、AK37D

KO文件名称	说明	支持芯片
ts_icn85xx_mipi.ko	mipi触摸屏驱动	AK37E
ak_efuse.ko	efuse模块驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
ak_fb.ko	LCD屏的驱动	AK37E、AK37D
ak_gui.ko	2D加速模块的驱动	AK37E、AK37D
ak_cs42l52.ko	I2S的驱动	AK37E
exfat.ko	文件系统驱动	AK37E、AK37D、AK39EV330L、AK3918AV100
ak_venc_adapter.ko	视频编码适配驱动	AK3918AV100
ak_venc_bridge.ko	视频编码桥接驱动（加载时要先加载ak_venc_adapter.ko，再加载ak_venc_bridge.ko）	AK3918AV100