



CV180X & CV181X MediaSDK 安装及 升级使用手册

Version: 1.0.0

Release date: 2024-05-11

©2022 北京晶视智能科技有限公司
本文件所含信息归北京晶视智能科技有限公司所有。
未经授权，严禁全部或部分复制或披露该等信息。

目录

1	声明	2
2	概述	3
2.1	阅读说明	3
2.2	Release 内容	3
3	开发环境配置	4
3.1	Linux 服务器	4
3.1.1	VirtualBoxVM 安装 Ubuntu	4
3.1.2	开发环境搭建	6
3.2	构建编译环境	6
4	编译和烧录	8
4.1	SDK 安装位置	8
4.2	编译准备	8
4.3	板端配置	9
4.4	编译	10
4.5	烧录	10
4.5.1	使用 SD 烧录	10
4.5.2	使用 USB 烧录	10
5	MediaSDK 目录结构介绍	11
6	FLASH 和文件系统介绍	13
6.1	FLASH 分区介绍	13
6.2	文件系统介绍	13
7	地址空间分配与使用介绍	15
7.1	DDR 内存分配	15
7.2	DDR 内存管理说明	15

修订记录

Revision	Date	Description
1.0.0	2024/05/11	初版

1 声明



法律声明

本数据手册包含北京晶视智能科技有限公司（下称“晶视智能”）的保密信息。未经授权，禁止使用或披露本数据手册中包含的信息。如您未经授权披露全部或部分保密信息，导致晶视智能遭受任何损失或损害，您应对因之产生的损失/损害承担责任。

本文件内信息如有更改，恕不另行通知。晶视智能不对使用或依赖本文件所含信息承担任何责任。本数据手册和本文件所含的所有信息均按“原样”提供，无任何明示、暗示、法定或其他形式的保证。晶视智能特别声明未做任何适销性、非侵权性和特定用途适用性的默示保证，亦对本数据手册所使用、包含或提供的任何第三方的软件不提供任何保证；用户同意仅向该第三方寻求与此相关的任何保证索赔。此外，晶视智能亦不对任何其根据用户规格或符合特定标准或公开讨论而制作的可交付成果承担责任。

联系我们

地址 北京市海淀区丰豪东路 9 号院中关村集成电路设计园（ICPARK）1 号楼

深圳市宝安区福海街道展城社区会展湾云岸广场 T10 栋

电话 +86-10-57590723 +86-10-57590724

邮编 100094（北京）518100（深圳）

官方网站 <https://www.sophgo.com/>

技术论坛 <https://developer.sophgo.com/forum/index.html>

2 概述

2.1 阅读说明

该文档为车载 Turnkey 产品的安装及升级使用说明，可以帮助用户能快速在 EVB 版上搭建 MediaSDK 的运行环境。

文档包含下述主要内容：

- SDK 环境搭建、配置、编译、烧录
- MediaSDK 目录结构介绍
- 烧录分区和文件系统介绍
- 地址空间分配与使用介绍

2.2 Release 内容

MediaSDK Release 内容包含如下组成部分：

Release 内容	说明
MediaSDK_cardv.tar.gz	SDK 压缩包
Host-tools.tar.gz	工具链压缩包

注解：

- RISC-V 工具链下载地址：<https://sophon-file.sophon.cn/sophon-prod-s3/drive/23/03/07/16/host-tools.tar.gz>
- ARM 工具链下载地址：<https://sophon-file.sophon.cn/sophon-prod-s3/drive/23/02/27/16/host-tools.tar.gz>

3 开发环境配置

3.1 Linux 服务器

用户可选择使用：（推荐安装 Ubuntu 20.04 LTS 以上版本）

- Ubuntu OS 计算机
- Windows OS 计算机 + Virtualbox VM (上面运行 Ubuntu)

注解： Virtualbox VM 下载网址：<https://www.virtualbox.org/wiki/Downloads>

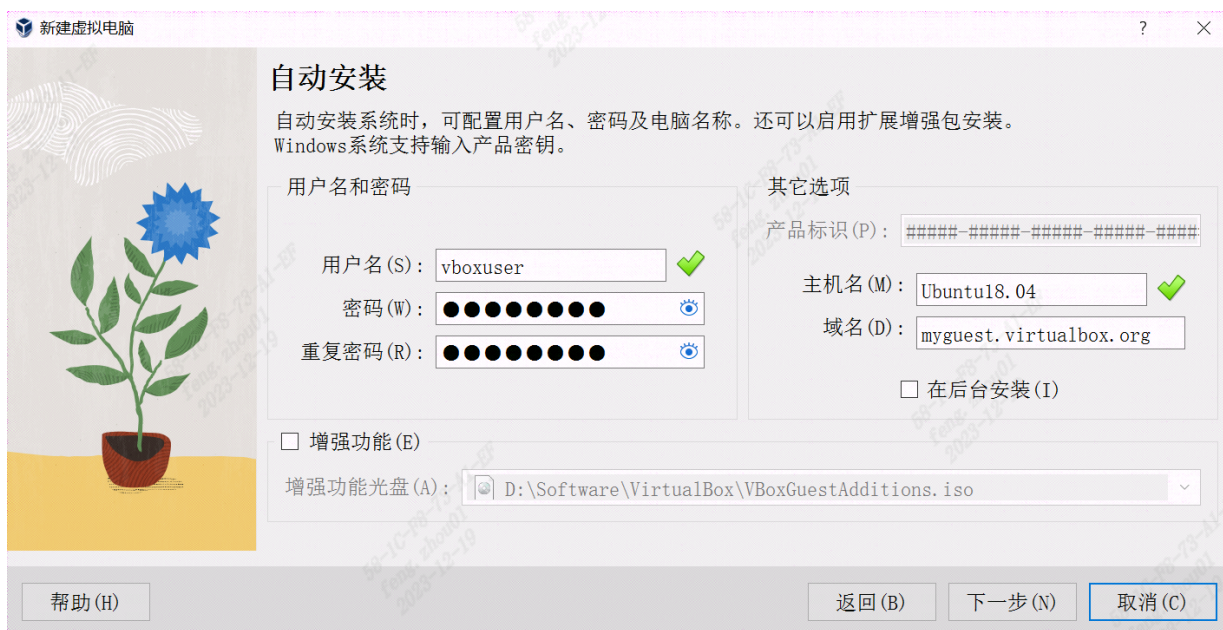
Ubuntu LTS 下载网址：<https://releases.ubuntu.com/>

3.1.1 VirtualBoxVM 安装 Ubuntu

- 建立新的 VM，并加以命名



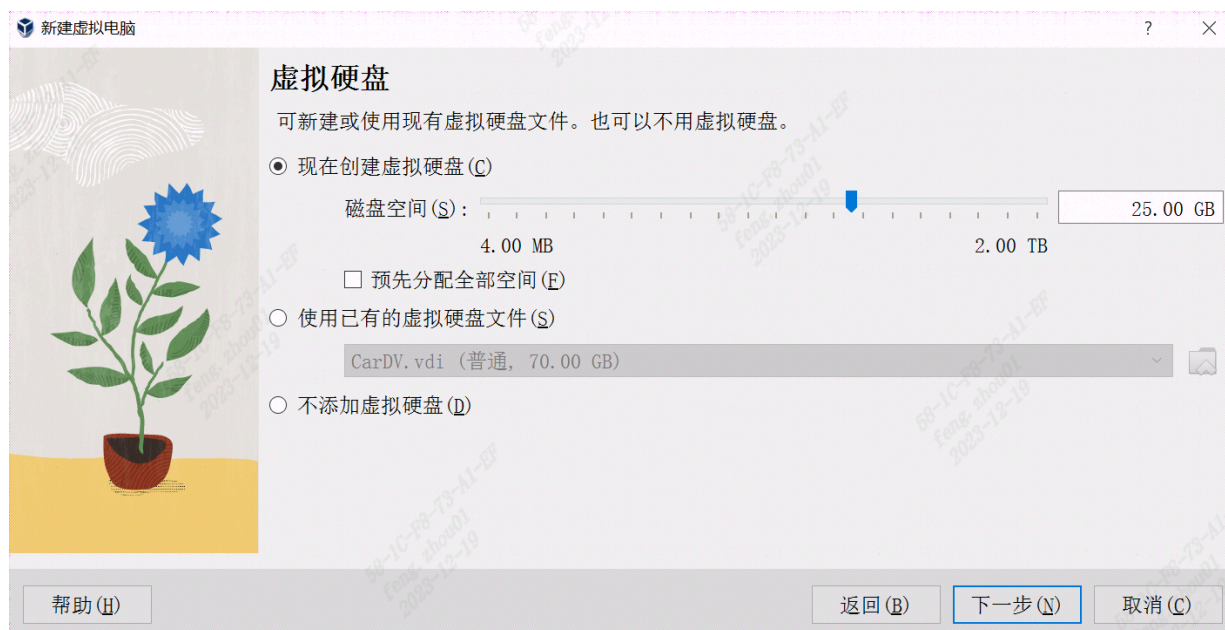
- 设置用户名和主机名



· 设置内存和处理器个数



· 预留硬盘空间，供后续存放 SDK 用



3.1.2 开发环境搭建

请用户在 Ubuntu OS 上自行配置网络，并安装 ssh, ftp, samba, nfs 等常用组件。

如果选择 Windows OS 计算机 + Virtualbox VM 的方式，推荐 Windows 端安装 MobaXterm, FileZila, Visual Studio Code 等软件提高开发效率。

3.2 构建编译环境

```
$ sudo apt-get install tar
$ sudo apt-get install vim
$ sudo apt-get install scons
$ sudo apt-get install cmake //版本要求至少3.13
$ sudo apt-get install ninja-build
$ sudo apt-get install python3-dev
$ sudo apt-get install python3-pip
$ sudo apt-get install python3-setuptools
$ sudo apt-get install git
$ sudo apt-get install bison
$ sudo apt-get install flex
$ sudo apt-get install libssl-dev
$ sudo apt-get install device-tree-compiler
$ sudo apt-get install libncurses5-dev libncursesw5-dev

$ pip3 install yoctools
$ pip3 install Pillow
$ pip3 install lz4
$ pip3 install jinja2
```

注解:

- 建议客户将虚拟环境的 Python 配置，切换为默认使用 Python 3.0 版本以上。
-

4 编译和烧录

4.1 SDK 安装位置

建议用户将 Release 的 SDK 安装包放置在 `/home/$(user)/` 目录下，方便后续设置工具链路径的环境变量。

4.2 编译准备

解压 `host-tools.tar.gz`

```
$ tar xzf host-tools.tar.gz
```

编辑环境变量

```
$ vim ~/.bashrc
```

环境变量添加工具链路径

```
# 添加riscv工具链

$ export PATH=$PATH:$PWD/host-tools/gcc/riscv64-linux-musl-x86_64/bin

$ export PATH=$PATH:$PWD/host-tools/gcc/riscv64-linux-x86_64/bin

$ export PATH=$PATH:$PWD/host-tools/gcc/riscv64-elf-x86_64/bin

# 添加arm工具链

$ export PATH=$PATH:$PWD/host-tools/gcc/gcc-linaro-6.3.1-2017.05-x86_64_arm-linux-gnueabihf/bin

$ export PATH=$PATH:$PWD/host-tools/gcc/gcc-linaro-6.3.1-2017.05-x86_64_aarch64-linux-gnu/bin

$ export PATH=$PATH:$PWD/host-tools/gcc/gcc-linaro-6.3.1-2017.05-x86_64_aarch64-elf/bin
```

更新当前环境变量

```
$ source ~/.bashrc
```

解压 MediaSDK_cardv.tar.gz

```
$ tar xzf MediaSDK_cardv.tar.gz
```

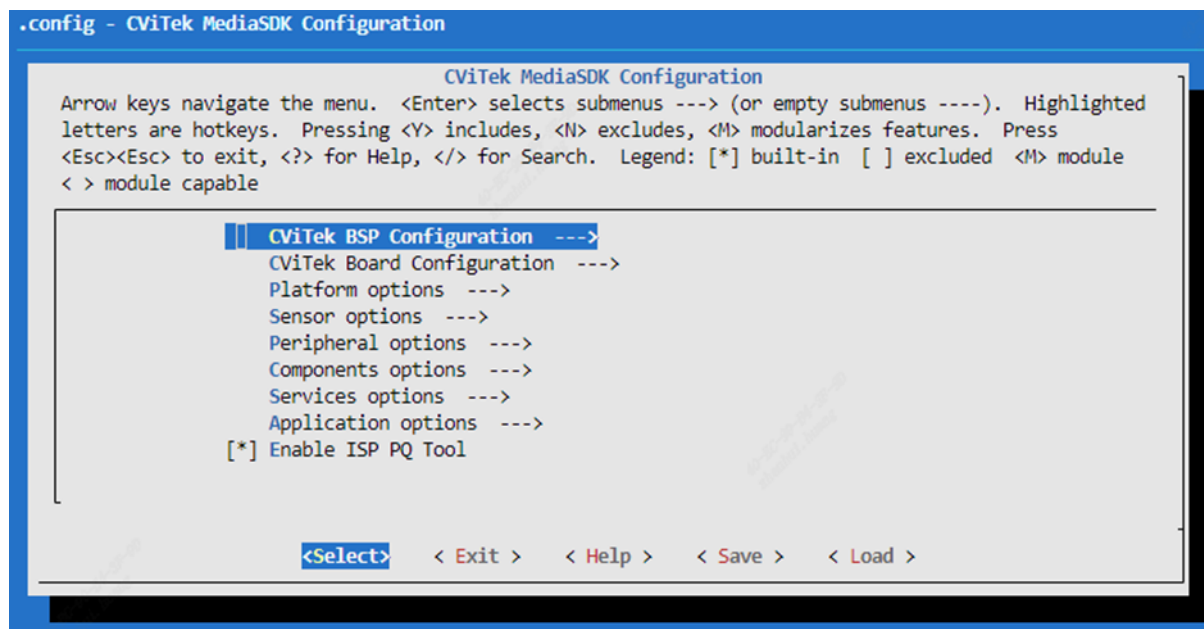
进入 MediaSDK 目录，后续操作都在 MediaSDK 目录执行

```
$ cd MediaSDK
```

4.3 板端配置

用户可以使用 make menuconfig 进行配置

```
$ make menuconfig
```



或者使用 defconfig (defconfig 文件位于 ./configs 目录下)

```
$ ls configs (to select a defconfig to start with)
```

```
$ make xxx_defconfig (eg. make cv1811h_spinor_gc4653_bt656_defconfig)
```

配置完成后，会在 MediaSDK 根目录下生成 .config 文件保存最终的配置

4.4 编译

编译是以 root 权限进行。

```
$ make clean  
$ make all  
$ make install
```

生成烧录文件（即（镜像文件）会被放至根目录/out/下，其中除文件夹，其余都是烧录的镜像文件）

```
$ make burn_images
```

生成升级包（生成 upgrade_0.1.8.bin）

```
$ make ota_images
```

4.5 烧录

4.5.1 使用 SD 烧录

将生成的镜像文件拷贝至 sd 卡根目录，插入 sd 卡至开发板，上电开机，烧录过程会自动开始。待烧录结束后，板端会自动执行 reboot 开机。

4.5.2 使用 USB 烧录

请用户参考《量产烧写使用指南》进行 usb 烧录。

5 MediaSDK 目录结构介绍

MediaSDK 目录结果如下：

```
|-- applications
|-- build
|-- cmake
|-- configs
|-- cpsl
|-- framework
|-- out
|-- platform
|-- scripts
|-- .config
|-- kbuild
|-- kconfig
|-- Makefile
|-- Makefile.kbuild
|-- Makefile.version
```

简要说明如下：

- application：包含应用层、客制化配置 ini 参数等。
- cpsl：跨平台支持层，提供系统底层库，包括 mpi, hal, osal, tpu 等接口的库文件和头文件。其中 hal 层以 source code 方式提供，并支持客户进行扩展。
- platform：包含 linux kernel, osdrv,u-boot,rootfs,fsbl,opensbi 相关目录，以 source code 方式提供。
- framework：媒体中间件层，包括 mapi 接口，流媒体相关组件，service 参考实现等。除部分组件外，以 source code 方式提供。

- configs: 存放 defconfig 文件。
- build & scripts: 编译所需脚本、Makefile 文件和配置。

其中板端配置主要在 build/boards 里面, 包含了以下配置 (以./build/cv181x/cv1811h_wevb_0007a_spinor 为例):

```
|-- alios          # alios引脚配置
|-- dts_riscv      # dts配置
|-- linux          # kernel config配置
|-- parititon      # Flash 分区配置
|-- rootfs         # rootfs 定制化
|-- rootfs_script  # clean rootfs脚本
|-- u-boot
| |-- cvi_board_init.c # uboot引脚配置
| |-- cvitek_cv1811h_wevb_0007a_spinor_defconfig # uboot config 配置
|-- memmap.py      #内存分配
```


6 FLASH 和文件系统介绍

6.1 FLASH 分区介绍

以 `cv1811h_wevb_0007a_spinor` 为例, MediaSDK 当前的 FLASH 分区配置可参考 `./build/boards/cv181x/cv1811h_wevb_0007a_spinor/partition` 下的 xml。

注解: 用户定制 rootfs 的时候, 可能会导致最终编译的 `rootfs.spinor` 固件大小大于 flash 分配的 rootfs 分区大小, 导致编译报错。请用户修改 `partition` 下的 xml 合理配置分区大小 (各分区大小需要 64 位对齐)。

6.2 文件系统介绍

默认情况下, 文件系统包含如下分区:

1. `boot`分区 (文件系统不可见)
2. `rootfs`分区 (`mount`至/)
3. `system`分区 (`mount`至/`mnt/system`)
4. `appcfg`分区 (存储应用参数不可见)
5. `appcfgdef`分区 (备份应用参数不可见)

注解: 当前 rootfs 中可读可写的分区仅有/`mnt/system` 和/`mnt/sd`。

启功脚本及初始化过程:

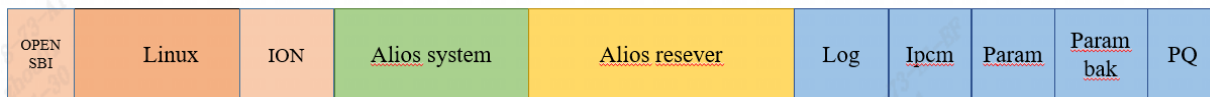
```
/etc/init.d  
  
xxx.sh  
  
-> yyy.sh  
  
-> zzz.sh
```

用户应用程序启动可以更改 `/mnt/system` 目录中的 `auto.sh` 添加相应启动项。

7 地址空间分配与使用介绍

7.1 DDR 内存分配

MediaSDK 的 DDR 主要按照下图所示进行划分：



7.2 DDR 内存管理说明

以 cv1811h_wevb_0007a_spinor 为例：

用户可以修改 build/boards/cv181x/cv1811h_wevb_0007a_spinor/memmap.py 文件，来更改 MediaSDK 当前的内存分配策略。