



# **RTOS TVD 开发指南**

**版本号: 1.0  
发布日期: 2021.4.15**

## 版本历史

| 版本号 | 日期        | 制/修订人   | 内容描述    |
|-----|-----------|---------|---------|
| 1.0 | 2021.4.15 | AWA1221 | 1. 添加初版 |



# 目 录

|                     |          |
|---------------------|----------|
| <b>1 前言</b>         | <b>1</b> |
| 1.1 文档简介            | 1        |
| 1.2 目标读者            | 1        |
| 1.3 适用范围            | 1        |
| <b>2 模块介绍</b>       | <b>2</b> |
| 2.1 模块功能介绍          | 2        |
| 2.2 相关术语介绍          | 2        |
| 2.2.1 硬件术语          | 2        |
| 2.2.2 软件术语          | 2        |
| 2.3 模块配置介绍          | 3        |
| 2.3.1 menuconfig 配置 | 3        |
| 2.3.2 SOC 平台配置说明    | 4        |
| <b>3 模块接口说明</b>     | <b>5</b> |
| 3.1 测试程序            | 5        |
| 3.2 亮度对比度和饱和度       | 6        |
| 3.3 判断信号是否锁住        | 7        |
| 3.4 选择输入通道          | 7        |
| <b>4 FAQ</b>        | <b>8</b> |
| 4.1 信号闪烁            | 8        |
| 4.2 图像质量            | 8        |

# 1 前言

## 1.1 文档简介

介绍 Sunxi 平台 RTOS 上 TVD 驱动的一般使用方法及调试接口，为开发与调试提供参考。

## 1.2 目标读者

- TVD 驱动开发人员/维护人员
- TVD 模块的应用层使用者

## 1.3 适用范围

表 1-1: 适用产品列表

| 产品名称 | 内核版本  | 驱动文件  |
|------|-------|---|
| D1s  | Melis | source/ekernel/subsys/<br>avframework/v4l2/drivers/<br>media/platform/sunxi-tvd |

## 2 模块介绍

### 2.1 模块功能介绍

TVD(Television Decoder), 也叫做 CVBS(Composite Video Broadcast Signal) IN 接口, 是一个视频输入接口用于传输复合视频信号。

特性:

1. 支持两个输入。同一时刻只能选择其中一个。
2. CVBS 输入, 支持 NTSC 和 PAL。
3. 支持采集转换为 YUV422 和 YUV420 格式。
4. 支持一个 3D comb 滤波。
5. 信号检测。
6. 支持亮度, 对比度和饱和度调节。

### 2.2 相关术语介绍

#### 2.2.1 硬件术语

表 2-1: 硬件术语

| 术语   | 解释  |
|------|---|
| TVD  | Television Decoder 电视解码                         |
| CVBS | Composite Video Broadcast Signal 复合视频广播信号       |
| NTSC | National Television Standards Committee , 一种分辨率 |
| PAL  | Phase Alternating Line , 一种分辨率                  |

#### 2.2.2 软件术语

表 2-2: 软件术语

| 术语   | 解释说明                                 |
|------|--------------------------------------|
| HAL  | Hardware Abstraction Layer, 硬件抽象层    |
| RTOS | Real Time Operating System, 实时操作系统   |
| GPIO | General Purpose Input/Output, 通用输入输出 |

## 2.3 模块配置介绍

### 2.3.1 menuconfig 配置

如下图红框所示路径，红框圈住的都需要选中。

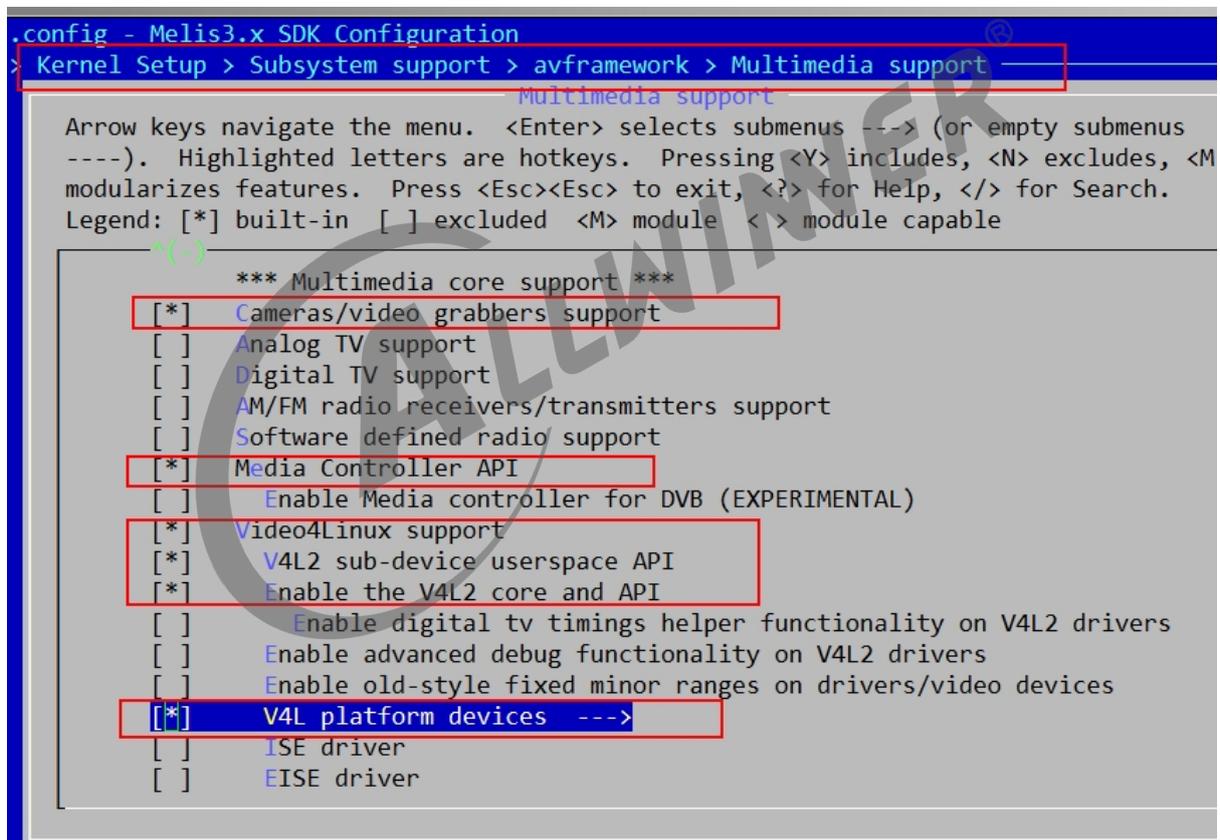


图 2-1: menuconfig

如下图红框所示路径，其中 SUNXI TVD(CVBS IN) driver test 是测试命令，非必要。

```
.config - Melis3.x SDK Configuration
[...] 1 Setup > Subsystem support > avframework > Multimedia support > V4L platform de
V4L platform devices
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus
----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M>
modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search.
Legend: [*] built-in [ ] excluded <M> module < > module capable

-- V4L platform devices
[*] SUNXI TVD(CVBS IN) driver
[*] SUNXI TVD(CVBS IN) driver test
[ ] eve
[ ] sunxi video input (camera csi/mipi isp vipp)driver
[ ] sensor power in front of vin
```

图 2-2: menuconfig2

驱动加载成功将生成/dev/video4 节点，而测试命令的名字叫做 video\_view。

### 2.3.2 SOC 平台配置说明

新平台，如果硬件没有变化，只需要修改下面这个文件，定义若干宏即可。

source/ekernel/subsys/avframework/v4l2/drivers/media/platform/sunxi-tvd/soc.h

根据平台宏，定义TVDO\_IRQ\_NO与TVDO\_REG\_BASE，分别表示中断号和模块基地址。

```
#if defined (CONFIG_ARCH_SUN20IW1)
#define TVD0_IRQ_NO 123
#define TVD0_REG_BASE 0x05c01000
#else
#error "NULL ARCH!!"
#endif

//1, auto agc, 0, manual mode
#define AGC_AUTO_ENABLE 1
//agc value when AGC_AUTO_ENABLE == 0
#define AGC_MANUAL_VALUE 64
//enable cagc
#define CAGC_ENABLE 1
// enable 3D filter
#define FLITER_USED 1
```

## 3 模块接口说明

### 3.1 测试程序

tvd 模块基于 v4l2 框架，测试用例根据 `menuconfig`，选中 SUNXI TVD(CVBS IN) driver test，系统启动后可以找到一个名字叫做 `video_view` 的程序。

对应的测试源文件是：`source/ekernel/subsys/avframework/v4l2/drivers/media/platform/sunxi-tvd/video_view.c`

主程序：`video_view`

选项说明：

1. `-ch channel`: 该选项后面接一个整数，代表你要打开的 `video` 节点，比如如果你要打开 `/dev/video4` 的话，那么这个数就是 4。该选项是必要选项不能没有。4~7 都表示单独的一路 `cvbs in`，而 8 表示多路合并。
2. `-fmt pixelfmt`: 该选项后面接一个整数，代表 `tvd` 模块采集的像素格式，总共有三种：0 表示 `nv12(yuv420sp uv combine)`，2 表示 `nv16(yuv422sp uv combine)`。该选项是非必要选项。
3. `-full enable`: 该选项后面接受一个整数，0 或者 1，1 表示使能自动根据显示分辨率全屏显示，0 则表示 `disable`。这个选项只有在选项有意义的情况下才行。该选项是非必要选项。
4. `-w width`: 该选项后面接受一个整数。该整数表示显示时的宽，只有在后面的数字为 0 时才有意义。该选项是非必要选项。
5. `-h height`: 该选项后面接受一个整数。该整数表示显示时的高，只有在后面的数字为 0 时才有意义。该选项是非必要选项。
6. `-c number`: 该选项后面接受一个整数，用来表示采集的帧数。如果是 -1 的话，那么一直采集。
7. `-screen_id disp_index`: 该选项后面接受一个整数，用来表示要在哪个显示上显示，范围是 -1~4。大于 0 的情况需要显示硬件支持。如果是 -1 则表示不显示。
8. `-di enable`: 该选项后面接受一个整数，0 或者 1，1 表示使能 `de-interlace`，也就是解隔行功能。该选项是非必须选项。
9. `-i input`: 该选项后面接受一个整数，范围 0~1，表示切换 `tvd` 的输入源，这个功能只有特定的 SOC 才支持。
10. `-b brightness`: 亮度值，范围是 0 到 255。
11. `-t contrast`: 对比度值，范围是 0 到 255。
12. `-s saturation`: 饱和度值，范围是 0 到 255。
13. `-help`: 打印命令行选项说明。或者直接运行 `video_view` 不带任何参数。

## 举例

```
#打开/dev/video4 采集格式是nv12, 在screen 0上显示。根据显示分辨率全屏显示（必要时会从采集分辨率缩放到屏分辨率）
video_view -ch 4 -fmt 0 -full 1 -screen_id 0

#同上, 除了采集格式换成nv16
video_view -ch 4 -fmt 2 -full 1 -screen_id 0

#非自动全屏显示, 而是指定显示的宽高为800x480, 注意要正常显示的前提是指定的宽高不超过屏幕分辨率。
video_view -ch 4 -fmt 0 -full 0 -screen_id 0 -w 800 -h 480

#打开/dev/video4 采集格式是nv12, 不显示, 采集两帧并各自保存在/mnt/F目录下, 保存文件名字的格式
video4_frame0_720x480_fmt0.yuv
video_view -ch 4 -fmt 0 -full 1 -screen_id -1 -c 2

#打开/dev/video4 采集格式是nv12, 显示在screen0上面, 亮度是50, 对比度是100, 饱和度是100
video_view -ch 4 -full 1 -b 50 -t 100 -s 100

#同上, 选择输入源1, 采集并显示在screen 0上面。
video_view -ch 4 -full 1 -i 1
```

**注意:** 由于 melis 命令接口的限制, 命令的参数个数有限制, 超出限制的个数将会报错。所以这里建议, 以上参数, 如果其后面的值是 0, 那么可以不用传该参数, 只传非 0 的参数。

## 3.2 亮度对比度和饱和度

调用 v4l2 ioctl VIDIOC\_S\_CTRL来实现对亮度, 对比度和饱和度的调节。

该 ioctl 核心的结构体:

```
1 /*
2  * C O N T R O L S
3  */
4 struct v4l2_control {
5     __u32      id;
6     __s32      value;
7 };
```

其中 id 可以是下面三个取值, 它们在linux/v4l2-controls.h中定义:

1. V4L2\_CID\_BRIGHTNESS
2. V4L2\_CID\_CONTRAST
3. V4L2\_CID\_SATURATION

上面三个 id 对应的value的范围都是 0~255。

```
1 struct v4l2_control ctl;
2 ctl.id = V4L2_CID_BRIGHTNESS;
3 ctl.value = dev.brightness;
```

```
4 ret = ioctl(fd, VIDIOC_S_CTRL, &ctl);
5 if (ret)
6     printf("Set brightness :%u fail:ret:%d\n", dev.brightness, ret);
```

### 3.3 判断信号是否锁住

参考 video\_view.c, 通过循环调用 VIDIOC\_G\_FMT 来判断视频信号是否锁住。这个步骤是必须的, 不等视频信号锁住就开始采集将造成采集的数据异常。

```
1 fmt.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
2 while (ioctl(fd, VIDIOC_G_FMT, &fmt)) {
3     printf("get signal failed.\n");
4     usleep(1000);
5 }
```

同时从返回的 fmt 中可以得知视频信号的宽高。

### 3.4 选择输入通道

通过 v4l2 的 ioctl 命令:VIDIOC\_S\_INPUT, 我们可以切换 tvd 的输入通道。

```
1 int input = 1;
2 ioctl(fd, VIDIOC_S_INPUT, &input);
```

## 4 FAQ

### 4.1 信号闪烁

可能原因有：

1. 接口接触问题，摄像头本身供电问题
2. 应用层没有等信号锁住就开始采集。参考[判断信号是否锁住](#)。

### 4.2 图像质量

可以采取下面两个措施改善图像质量

1. 采用 nv16 格式。但这样相应的带宽就会增加，然后要求通路上的所有模块都得支持 nv16。
2. 参考[亮度对比度和饱和度](#)。

## 著作权声明

版权所有 © 2021 珠海全志科技股份有限公司。保留一切权利。

本档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本档内容的部分或全部，且不得以任何形式传播。

## 商标声明

、 **全志科技** （不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

## 免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本档作为使用指导仅供参考。由于产品版本升级或其他原因，本档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本档中提供准确的信息，但并不确保内容完全没有错误，因使用本档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。