

# Tina

camera 自适应使用说明书 v3.0

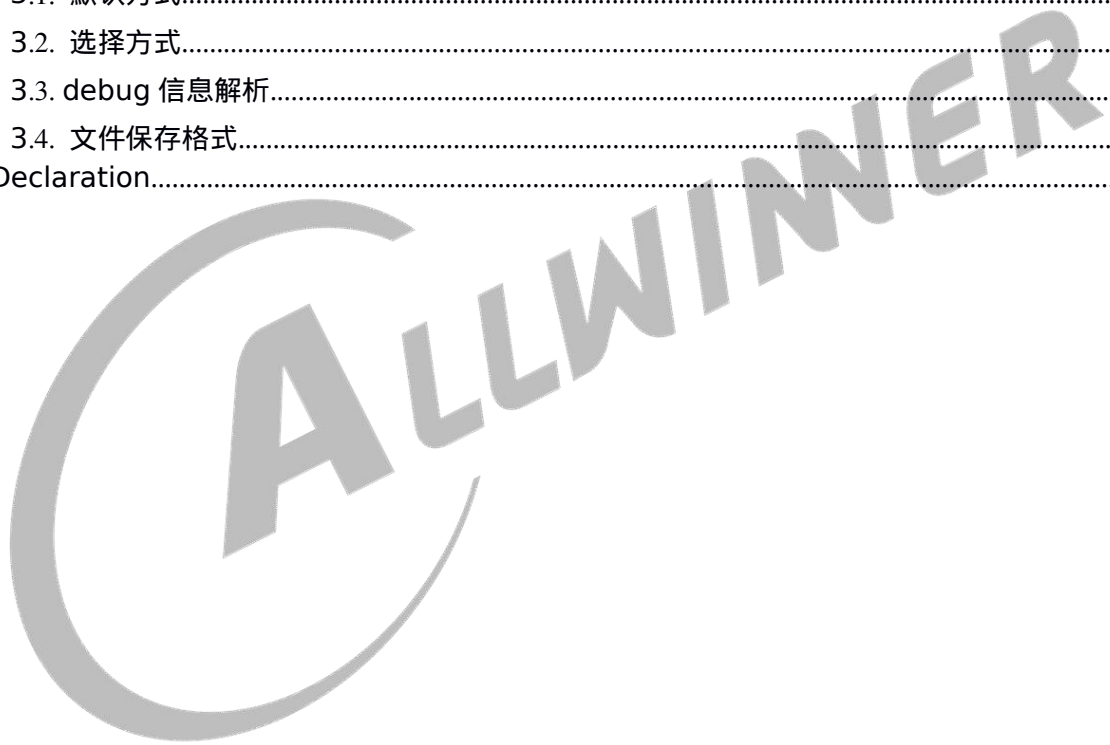
# 文档履历

版本号	日期	制/修订人	制/修订记录
V3.0			



# 目 录

1. 概述.....	4
1.1. 编写目的.....	4
1.2. 适用范围.....	4
1.3. 相关人员.....	4
2. 模块配置.....	5
2.1. menuconfig 配置说明.....	5
2.1.1. camera 内核模块配置.....	5
2.1.2. camera 测试 demo 配置.....	7
2.2. 源码结构介绍.....	8
3. camerademo 使用方法.....	9
3.1. 默认方式.....	11
3.2. 选择方式.....	12
3.3. debug 信息解析.....	15
3.4. 文件保存格式.....	17
4. Declaration.....	18



## 1. 概述

### 1.1. 编写目的

介绍 Tina 系统中 camerademo 的使用方法，为 camera sensor 的测试提供参考

### 1.2. 适用范围

适用于所有使用 Tina 系统存在 camera 的硬件平台

### 1.3. 相关人员

公司开发人员、客户



## 2. 模块配置

### 2.1.menuconfig 配置说明

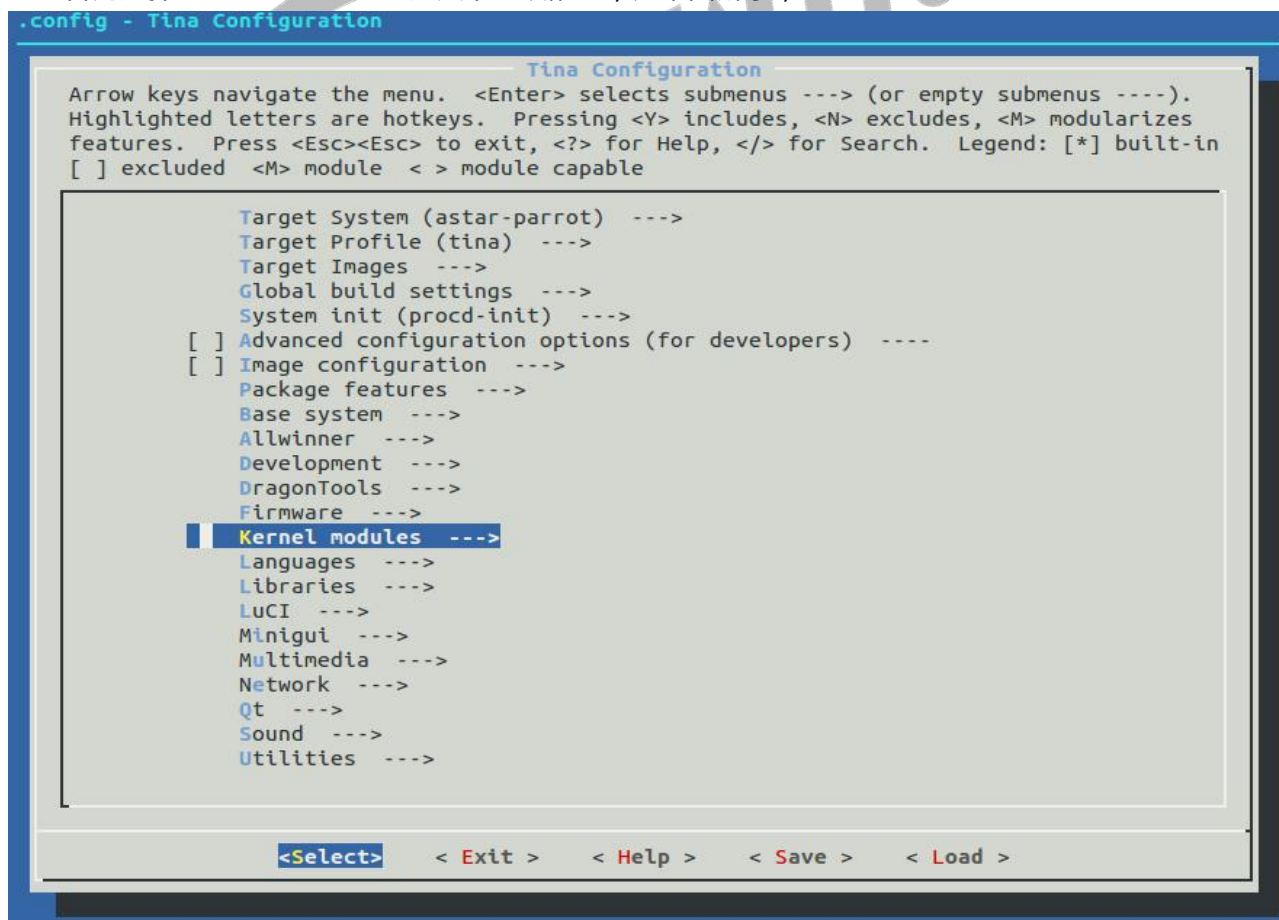
#### 2.1.1. camera 内核模块配置

在命令行中进入 Tina 根目录，执行 make menuconfig 进入配置主界面，并按以下配置路径操作选择编译 camera 相关模块：

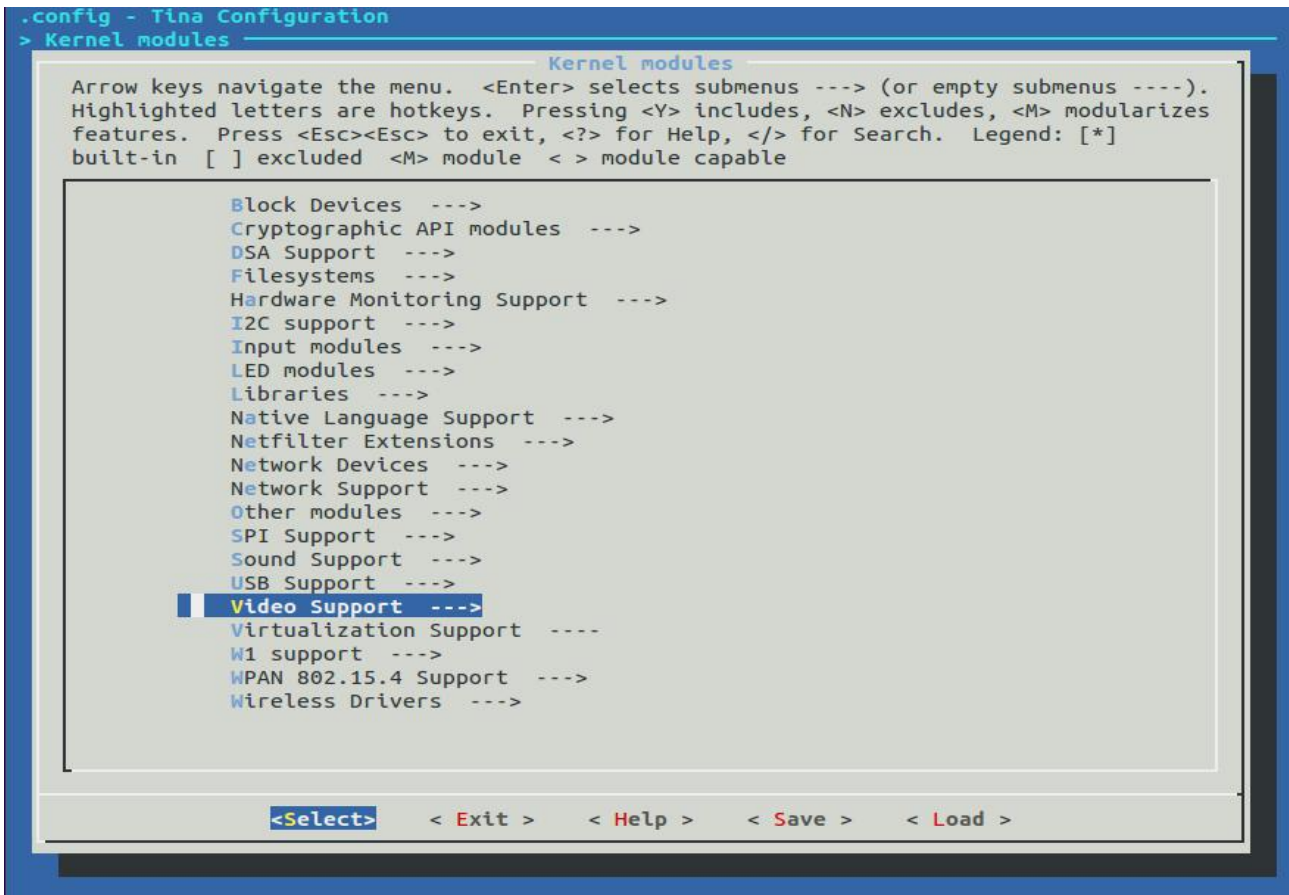
```
Kernel modules
├─>Video Support
│   ├──>kmod-sunxi-vfe(vfe 框架的 csi camera) (详见①)
│   ├──>kmod-sunxi-vin(vin 框架的 csi camera) (详见②)
│   └──>kmod-sunxi-uvic(ucv camera) (详见③)
```

- 详注：
- ①平台使用 vfe 框架的 csi camera 选择该驱动；
  - ②平台使用 vin 框架的 csi camera 选择该驱动；(该项与 vfe 框架，在同一个平台只会出现其中一个)
  - ③usb camera 选择该驱动；

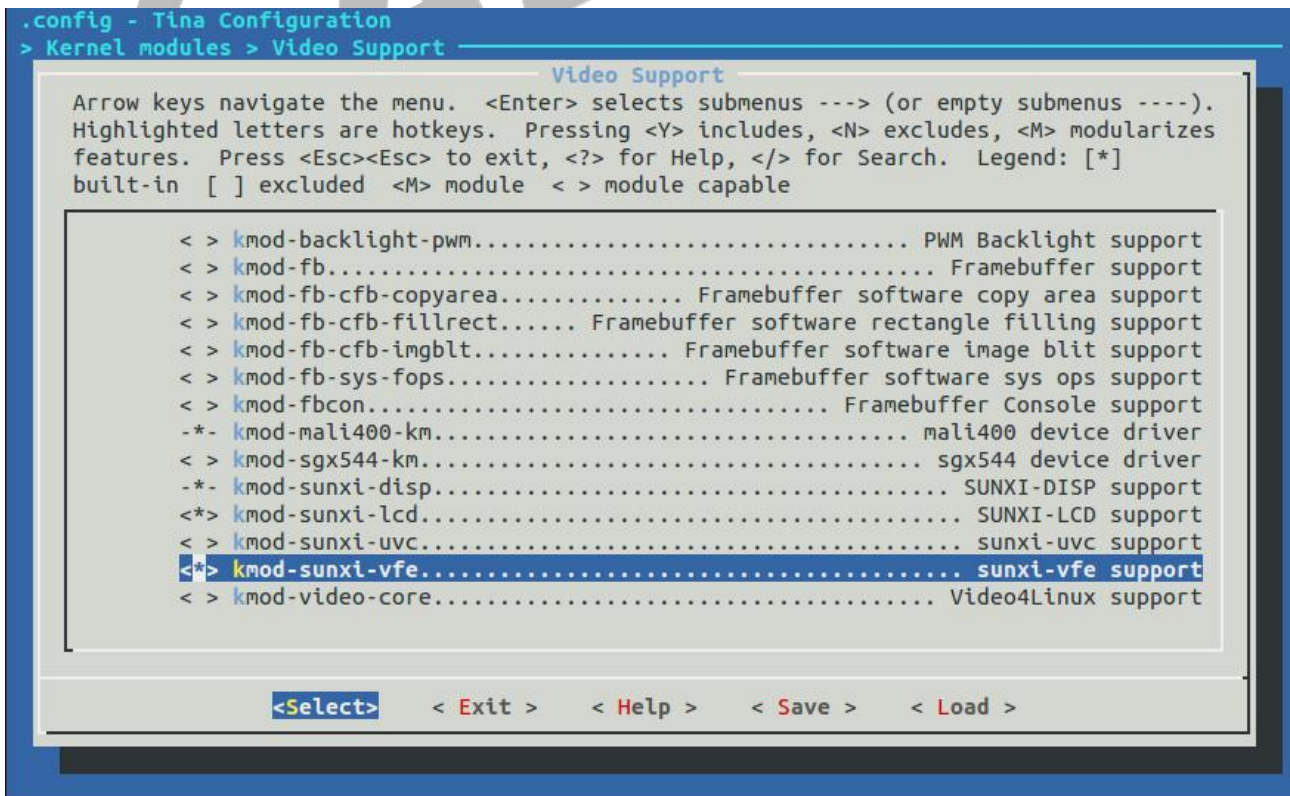
首先选择 Kernel modules 进入下一项配置，如下图所示，



然后选择 Video Support，如下图所示，



最后选上 kmod-sunxi-vfe 则会编译 csi camera 相关模块，选上 kmod-sunxi-uvic 则会编译 usb camera 相关模块。如下图所示，

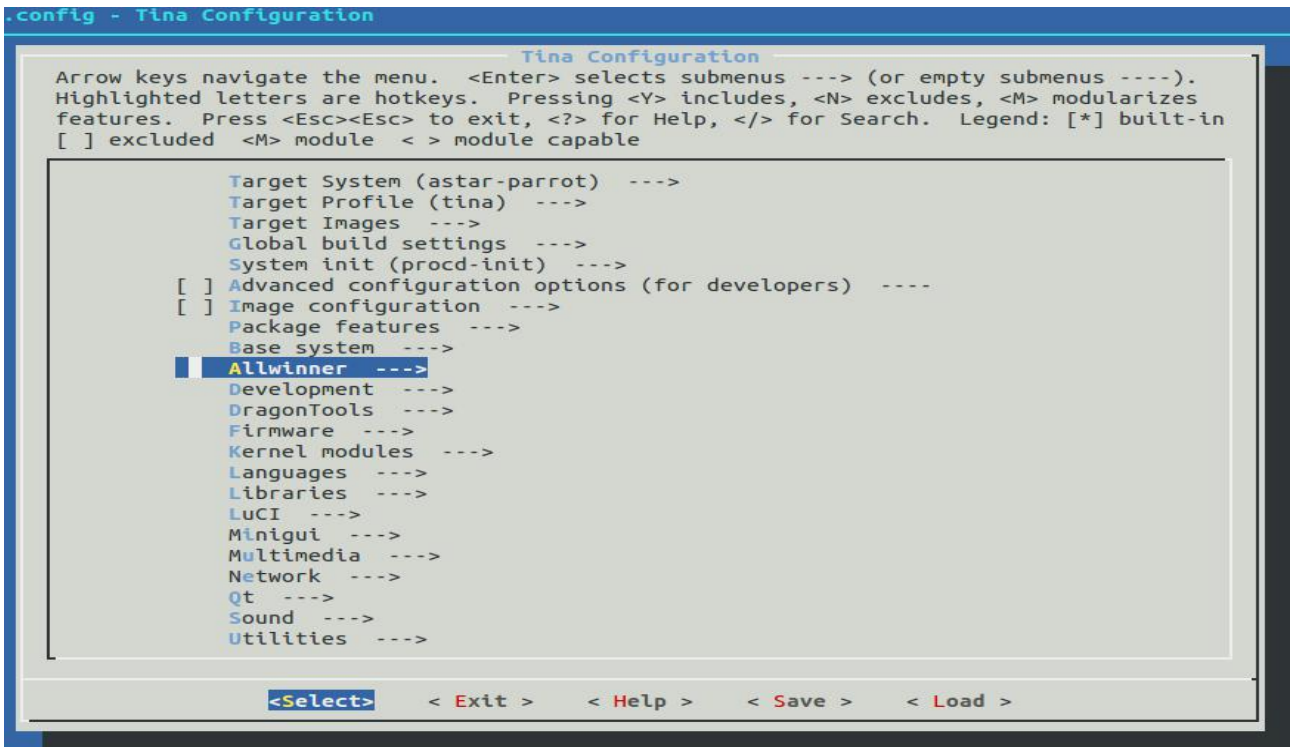


### 2.1.2. camera 测试 demo 配置

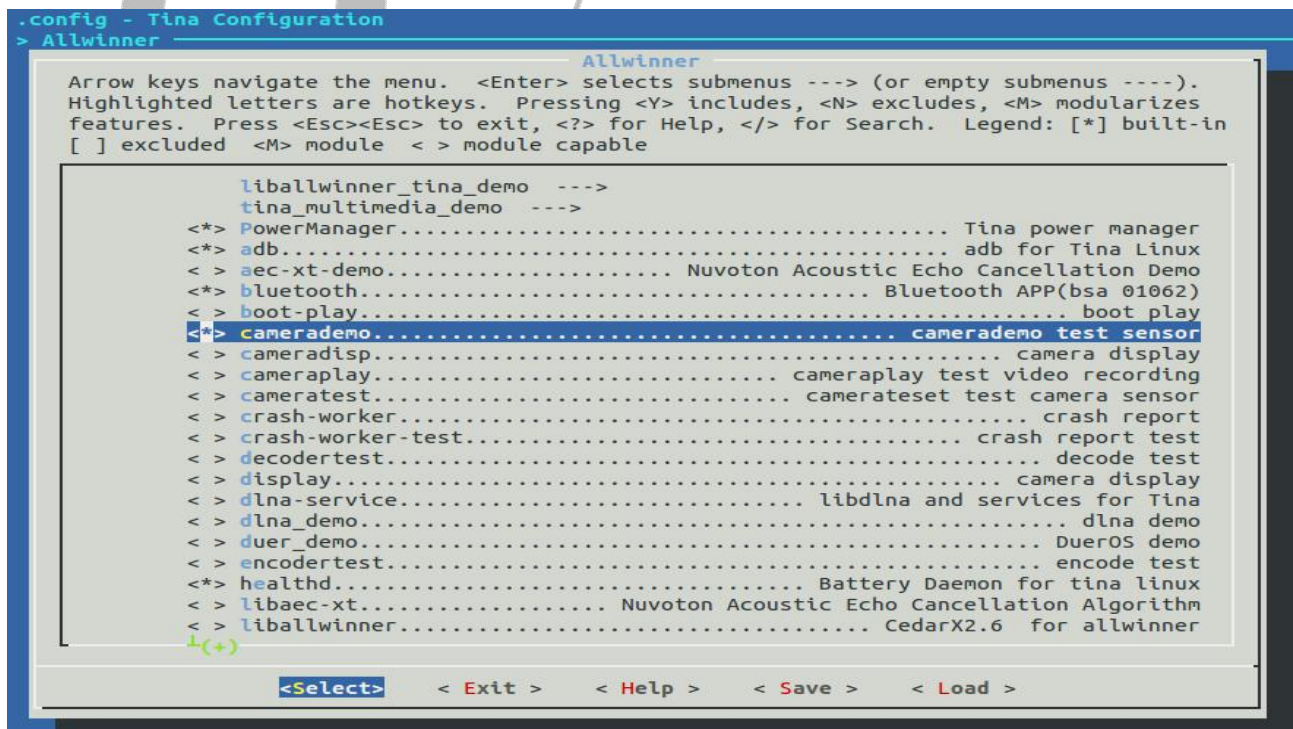
在命令行中进入 Tina 根目录，执行 make menuconfig 进入配置主界面，并按以下配置路径操作：

```
Allwinner
└─>camerademo
```

首先，选择 Allwinner 选项进入下一级配置，如下图所示：



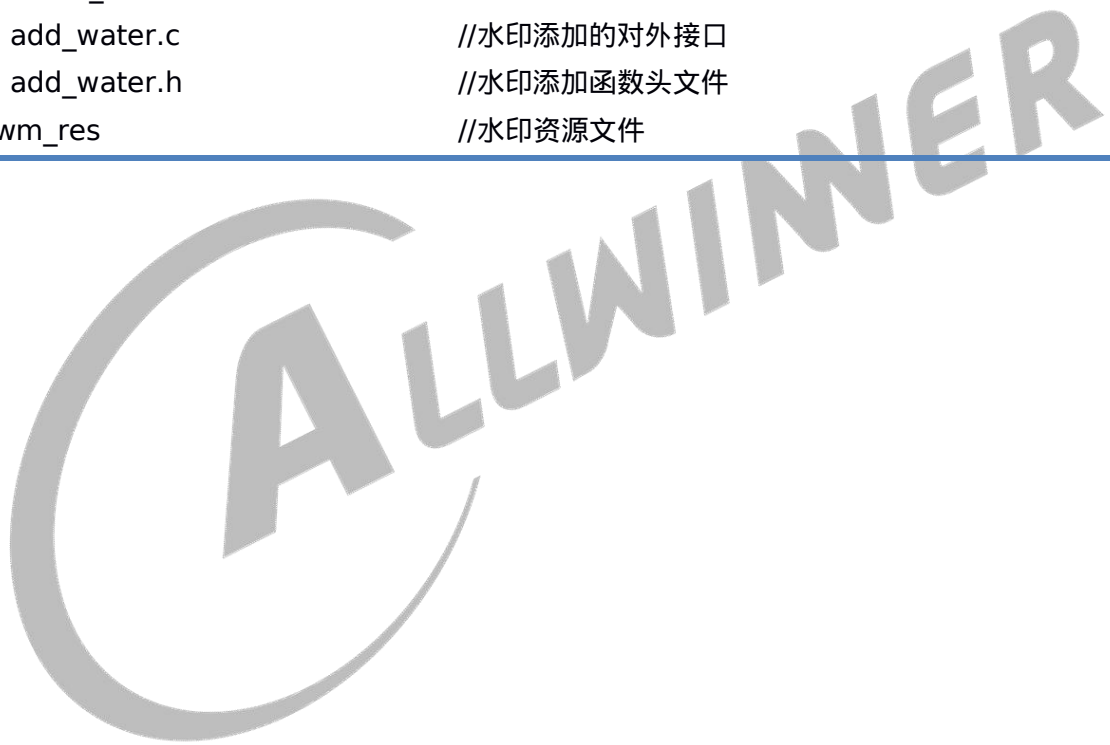
然后，选择 camerademo 选项，可选择 <\*> 表示直接编译进内核，也可以选择 <M> 表示编译成 ko 模块。



## 2.2.源码结构介绍

camerademo 的源代码位于 package/allwinner/camerademo/目录下：

```
|---src
|  camerademo.c           //camea 测试的主流程代码
|  camerademo.h           //camera demo 相关数据结构
|  common.c               //实现共用的函数，转换时间、保存文件、测试帧率等
|  common.h               //共用函数头文件
|  convert.c              //实现图像格式转换函数
|  convert.h              //图像格式转换函数头文件
|  water_mark.c           //添加水印的具体实现
|  water_mark.h           //添加水印的相关数据结构
|  add_water.c            //水印添加的对外接口
|  add_water.h            //水印添加函数头文件
|---wm_res                //水印资源文件
```





### 3. camerademo 使用方法

在小机端加载成功后输入 camerademo help，假如驱动产生的节点 video0（测试默认以/dev/video0 作为设备对象）可以打开则会出现下面提示：

通过提示我们可以得到一些提示信息，了解到该程序的运行方式、功能，可以查询 sensor 支持的分辨率、sensor 支持的格式以及设置获取照片的数量、数据保存的格式、路径、添加水印、测试数据输出的帧率、从 open 节点到数据流打通需要的时间等，help 打印信息如下图：

```

root@TinaLinux:/# camerademo help
[CAMERA]*****
[CAMERA]*
[CAMERA]*           this is camera test.
[CAMERA]*
[CAMERA]*
[CAMERA]*****
[CAMERA]***** camerademo help *****
[CAMERA] This program is a test camera.
[CAMERA] It will query the sensor to support the resolution, output format and test frame rate.
[CAMERA] At the same time you can modify the data to save the path and get the number of photos.
[CAMERA] When the last parameter is debug, the output will be more detailed information
[CAMERA] There are eight ways to run:
[CAMERA] 1.camerademo --- use the default parameters.
[CAMERA] 2.camerademo debug --- use the default parameters and output debug information.
[CAMERA] 3.camerademo setting --- can choose the resolution and data format.
[CAMERA] 4.camerademo setting debug --- setting and output debug information.
[CAMERA] 5.camerademo NV21 640 480 0 bmp /tmp 5 --- param input mode,can save bmp or yuv.
[CAMERA] 6.camerademo NV21 640 480 0 bmp /tmp 5 debug --- output debug information.
[CAMERA] 7.camerademo NV21 640 480 0 bmp /tmp 5 Num --- /dev/videoNum param input mode,can save bmp or yuv.
[CAMERA] 8.camerademo NV21 640 480 0 bmp /tmp 5 Num debug --- /dev/videoNum output debug information.
[CAMERA]*****
root@TinaLinux:/#

```

Camerademo 共有 4 种运行模式：

1) 默认方式：直接输入 camerademo 即可，在这种运行模式下，将设置摄像头为 640\*480 的 NV21 格式输出图像数据，并以 BMP 和 YUV 的格式保存在 /tmp 目录下，而当输入 camerademo debug 将会输出更详细的 debug 信息；

2) 探测设置 camerademo setting：将会在运行过程中根据具体 camera 要求输入设置参数，当输入 camerademo setting debug 的时候，将会输出详细的 debug 信息；

3) 快速设置：camerademo argv[1] argv[2] argv[3] argv[4] argv[5] argv[6] argv[7]，将会按照输入参数设置图像输出，同样，当输入 camerademo argv[1] argv[2] argv[3] argv[4] argv[5] argv[6] argv[7] debug 时将会输出更详细的 debug 信息。

4) 选择 camera 设置 : camerademo argv[1] argv[2] argv[3] argv[4] argv[5] argv[6] argv[7] argv[8] ,将会按照输入参数设置图像输出 ,同样 ,当输入 camerademo argv[1] argv[2] argv[3] argv[4] argv[5] argv[6] argv[7] argv[8] debug 时将会输出更详细的 debug 信息。



### 3.1.默认方式

当输入 camerademo 之后，使用默认的参数运行，则会打印一下信息，如下图：

```

root@TinaLinux:/# camerademo
[CAMERA]*****
[CAMERA]*
[CAMERA]*          this is camera test.
[CAMERA]*
[CAMERA]*****
[CAMERA]*****
[CAMERA] open /dev/video0!
[CAMERA]*****
[CAMERA]*****
[CAMERA] The path to data saving is /tmp.
[CAMERA] The number of captured photos is 5.
[CAMERA] save bmp and yuv format
[CAMERA] do not use watermarks
[CAMERA]*****
[CAMERA] Using format parameters NV21.
[CAMERA] camera pixelformat: NV21
[CAMERA] Resolution size : 640 * 480
[CAMERA] The photo save path is /tmp.
[CAMERA] The number of photos taken is 5.
[CAMERA] Camera capture framerate is 30/1
[CAMERA] VIDIOC_S_FMT succeed
[CAMERA] fmt.type = 1
[CAMERA] fmt.fmt.pix.width = 640
[CAMERA] fmt.fmt.pix.height = 480
[CAMERA] fmt.fmt.pix.pixelformat = NV21
[CAMERA] fmt.fmt.pix.field = 1
[CAMERA] stream on succeed
[CAMERA] capture num is [0]
[CAMERA_PROMPT] the time interval from the start to the first frame is 87 ms
[CAMERA] capture num is [1]
[CAMERA] capture num is [2]
[CAMERA] capture num is [3]
[CAMERA] capture num is [4]
[CAMERA] Capture thread finish
[CAMERA] close /dev/video0
root@TinaLinux:/#
    
```

首先可以清楚的看到成功 open video0 节点，并且知道照片数据的保存路径、捕获照片的数量以及当前设置：是否添加水印、输出格式、分辨率和从开启流传输到第一帧数据达到时间间隔等信息。如果需要了解更多的详细信息，可以在运行程序的时候输入参数 debug 即运行 camerademo debug 将会打开 demo 的 debug 模式 输出更详细的信息 包括 camera 的驱动类型，支持的输出格式以及对应的分辨率，申请 buf 的信息，实际输出帧率等。

### 3.2.选择方式

在选择模式下有两种运行方式，一种是逐步选择，在 camera 的探测过程，知道其支持的输出格式以及分辨率之后再设置 camera 的相关参数；另一种是直接在运行程序的时候带上相应参数 程序按照输入参数运行( 其中还可以选择 camera 索引 从而测试不同的 camera )，

1) 输入 camerademo setting，则按照程序的打印提示输入相应选择信息即可。

1. 输入保存路径、照片数量、保存的格式以及是否添加水印等

```
[CAMERA] *****
[CAMERA] Please enter the data save path:
/tmp
[CAMERA] Please enter the number of captured photos:
5
[CAMERA] Please enter the data save type:
[CAMERA] 0:save BMP and YUV formats
[CAMERA] 1:save BMP format
[CAMERA] 2:save YUV format
0
[CAMERA] Whether to add a watermark, enter 0 or 1, 0---NO, 1---YES:
1
```

2. 选择输出格式

```
[CAMERA] *****
[CAMERA] The sensor supports the following formats :
[CAMERA] index 0 : YUV422P
[CAMERA] index 1 : YUV420
[CAMERA] index 2 : YVU420
[CAMERA] index 3 : NV16
[CAMERA] index 4 : NV12
[CAMERA] index 5 : NV61
[CAMERA] index 6 : NV21
[CAMERA] index 7 : YUYV
[CAMERA] index 8 : YVYU
[CAMERA] index 9 : UYVY
[CAMERA] index 10 : VYUY
[CAMERA] Please enter the serial number you need for pixelformat:
4
[CAMERA] The input value is 4.
[CAMERA] camera pixelformat: NV12
```

3. 选择输出图像分辨率

```
[CAMERA] *****
[CAMERA] The NV12 supports the following resolutions:
[CAMERA] Index 0 : 2592 × 1936
[CAMERA] Index 1 : 2048 × 1536
[CAMERA] Index 2 : 1920 × 1080
[CAMERA] Index 3 : 1600 × 1200
[CAMERA] Index 4 : 1280 × 960
[CAMERA] Index 5 : 1280 × 720
[CAMERA] Index 6 : 1024 × 768
[CAMERA] Index 7 : 800 × 600
[CAMERA] Index 8 : 640 × 480
[CAMERA] Please enter the serial number you need for windows size:
0
[CAMERA] The input value is 0.
[CAMERA] Resolution size : 2592 × 1936
```

其它信息与默认设置一致，如需打印详细的信息，运行 camerademo setting debug 即可。

2) 第二种是设置参数：

1. 默认的 video 0 节点：camerademo argv[1] argv[2] argv[3] argv[4] argv[5] argv[6] argv[7]。

输入参数代表意义如下：

argv[1]：camera 输出格式---NV21 YUYV MJPEG 等；

argv[2]：camera 分辨率 width；

argv[3]：camera 分辨率 height；

argv[4]：是否添加水印：0---不添加，1---添加；

argv[5]：保存照片的格式：all---bmp 和 yuv 格式都保存、bmp---仅以 bmp 格式保存、yuv---仅以 yuv 格式保存；

argv[6]：捕获照片的保存路径；

argv[7]：捕获照片的数量；

例如：camerademo NV21 640 480 0 yuv /tmp 2，将会输出 640\*480 大小的 NV21 格式照片以 yuv 格式、不添加水印保存在/tmp 路径下，照片共 2 张。

其它信息与默认设置一致，如需打印详细的信息，运行 camerademo argv[1] argv[2] argv[3] argv[4] argv[5] argv[6] argv[7] debug 即可。

```

root@TinaLinux:/# camerademo NV21 640 480 0 yuv /tmp 2
[CAMERA]*****
[CAMERA]*
[CAMERA]*          this is camera test.          *
[CAMERA]*
[CAMERA]*****
[CAMERA]*****
[CAMERA] open /dev/video0!
[CAMERA]*****
[CAMERA]*****
[CAMERA] The path to data saving is /tmp.
[CAMERA] The number of captured photos is 2.
[CAMERA] save yuv format
[CAMERA] do not use watermarks
[CAMERA]*****
[CAMERA] Using format parameters NV21.
[CAMERA] camera pixelformat: NV21
[CAMERA] Resolution size : 640 * 480
[CAMERA] The photo save path is /tmp.
[CAMERA] The number of photos taken is 2.
[CAMERA] Camera capture framerate is 30/1
[CAMERA] VIDIOC_S_FMT succeed
[CAMERA] fmt.type = 1
[CAMERA] fmt.fmt.pix.width = 640
[CAMERA] fmt.fmt.pix.height = 480
[CAMERA] fmt.fmt.pix.pixelformat = NV21
[CAMERA] fmt.fmt.pix.field = 1
[CAMERA] stream on succeed
[CAMERA] capture num is [0]
[CAMERA_PROMPT] the time interval from the start to the first frame is 90 ms
[CAMERA] capture num is [1]
[CAMERA] Capture thread finish
[CAMERA] close /dev/video0
root@TinaLinux:/#
    
```

2.选择其他的 video 节点 : camerademo argv[1] argv[2] argv[3] argv[4] argv[5] argv[6] argv[7] argv[8]。

输入参数代表意义如下 :

- argv[1] : camera 输出格式---NV21 YUYV MJPEG 等 ;
- argv[2] : camera 分辨率 width ;
- argv[3] : camera 分辨率 height ;
- argv[4] : 是否添加水印 : 0---不添加 , 1---添加 ;
- argv[5] : 保存照片的格式 : all---bmp 和 yuv 格式都保存、bmp---仅以 bmp 格式保存、yuv---仅以 yuv 格式保存 ;
- argv[6] : 捕获照片的保存路径 ;
- argv[7] : 捕获照片的数量 ;
- argv[8] : video 节点索引 ;

例如 : camerademo YUYV 640 480 0 yuv /tmp 1 1 , 将会打开/dev/video1 节点并输出 640\*480 大小的以 yuv 格式、不添加水印保存在/tmp 路径下, 照片共 1 张。

其它信息与默认设置一致, 如需打印详细的信息, 运行 camerademo argv[1] argv[2] argv[3] argv[4] argv[5] argv[6] argv[7] argv[8] debug 即可。

```

root@TinaLinux:/# camerademo YUYV 640 480 0 yuv /tmp 1 1
[CAMERA]*****
[CAMERA]*
[CAMERA]*          this is camera test.
[CAMERA]*
[CAMERA]*****
[CAMERA]*****
[CAMERA] open /dev/video1!
[CAMERA]*****
[CAMERA]*****
[CAMERA] The path to data saving is /tmp.
[CAMERA] The number of captured photos is 1.
[CAMERA] save yuv format
[CAMERA] do not use watermarks
[CAMERA]*****
[CAMERA] Using format parameters YUYV.
[CAMERA] camera pixelformat: YUYV
[CAMERA] Resolution size : 640 * 480
[CAMERA] The photo save path is /tmp.
[CAMERA] The number of photos taken is 1.
[CAMERA] Camera capture framerate is 30/1
[CAMERA] VIDIOC_S_FMT succeed
[CAMERA] fmt.type = 1
[CAMERA] fmt.fmt.pix.width = 640
[CAMERA] fmt.fmt.pix.height = 480
[CAMERA] fmt.fmt.pix.pixelformat = YUYV
[CAMERA] fmt.fmt.pix.field = 1
[CAMERA] stream on succeed
[CAMERA] capture num is [0]
[CAMERA_PROMPT] the time interval from the start to the first frame is 65 ms
[CAMERA] Capture thread finish
[CAMERA] close /dev/video1
    
```

### 3.3.debug 信息解析

以下 debug 信息将说明 sensor 驱动的相关信息，拍摄到的照片保存位置、数量、保存的格式以及水印使用情况等：

```

root@TinaLinux:/# camerademo debug
[CAMERA]*****
[CAMERA]*
[CAMERA]*           this is camera test.
[CAMERA]*
[CAMERA]*****
[CAMERA]*****
[CAMERA] open /dev/video0!
[CAMERA]*****
[CAMERA_DEBUG] Query device capabilities succeed
[CAMERA_DEBUG] cap.driver=sunxi-vin
[CAMERA_DEBUG] cap.card=sunxi-vin
[CAMERA_DEBUG] cap.bus_info=
[CAMERA_DEBUG] cap.version=65536
[CAMERA_DEBUG] cap.capabilities=-2061496320
[CAMERA]*****
[CAMERA] The path to data saving is /tmp.
[CAMERA] The number of captured photos is 5.
[CAMERA] save bmp and yuv format
[CAMERA] do not use watermarks
    
```

以下 debug 信息将说明驱动框架支持的格式以及 sensor 支持的输出格式：

```

[CAMERA_DEBUG]*****
[CAMERA_DEBUG] enumerate image formats
[CAMERA_DEBUG] format index = 0, name = YUV422P
[CAMERA_DEBUG] format index = 1, name = NV16
[CAMERA_DEBUG] format index = 2, name = NV61
[CAMERA_DEBUG] format index = 3, name = YUV420
[CAMERA_DEBUG] format index = 4, name = YVU420
[CAMERA_DEBUG] format index = 5, name = NV12
[CAMERA_DEBUG] format index = 6, name = NV21
[CAMERA_DEBUG] format index = 7, name = YUYV
[CAMERA_DEBUG] format index = 8, name = UYVY
[CAMERA_DEBUG] format index = 9, name = VYUY
[CAMERA_DEBUG] format index = 10, name = YVYU
[CAMERA_DEBUG] format index = 11, name = YUYV
[CAMERA_DEBUG] format index = 12, name = UYVY
[CAMERA_DEBUG] format index = 13, name = VYUY
[CAMERA_DEBUG] format index = 14, name = YVYU
[CAMERA_DEBUG]*****
[CAMERA_DEBUG] The sensor supports the following formats :
[CAMERA_DEBUG] Index 0 : YUV422P.
[CAMERA_DEBUG] Index 1 : NV16.
[CAMERA_DEBUG] Index 2 : NV61.
[CAMERA_DEBUG] Index 3 : YUV420.
[CAMERA_DEBUG] Index 4 : YVU420.
[CAMERA_DEBUG] Index 5 : NV12.
[CAMERA_DEBUG] Index 6 : NV21.
[CAMERA_DEBUG] Index 7 : YUYV.
[CAMERA_DEBUG] Index 8 : UYVY.
[CAMERA_DEBUG] Index 9 : VYUY.
[CAMERA_DEBUG] Index 10 : YVYU.
[CAMERA_DEBUG] Index 11 : YUYV.
[CAMERA_DEBUG] Index 12 : UYVY.
[CAMERA_DEBUG] Index 13 : VYUY.
[CAMERA_DEBUG] Index 14 : YVYU.
    
```

类似以下的信息代表这相应格式支持的分辨率信息：

```
[CAMERA_DEBUG] *****
[CAMERA_DEBUG] The YUV422P supports the following resolutions:
[CAMERA_DEBUG] Index 0 : 2592 * 1936
[CAMERA_DEBUG] *****
[CAMERA_DEBUG] The NV16 supports the following resolutions:
[CAMERA_DEBUG] Index 0 : 2592 * 1936
[CAMERA_DEBUG] *****
[CAMERA_DEBUG] The NV61 supports the following resolutions:
[CAMERA_DEBUG] Index 0 : 2592 * 1936
[CAMERA_DEBUG] *****
```

以下信息将会提示将要设置到 sensor 的格式和分辨率等信息：

```
[CAMERA] camera pixelformat: NV21
[CAMERA] Resolution size : 2592 * 1936
[CAMERA] The photo save path is /tmp.
[CAMERA] The number of photos taken is 5.
```

以下信息将会提示设置格式的情况，buf 的相应信息等：

```
[CAMERA] Camera capture framerate is 1/1
[CAMERA] VIDIOC_S_FMT succeed
[CAMERA] fmt.type = 9
[CAMERA] fmt.fmt.pix.width = 2592
[CAMERA] fmt.fmt.pix.height = 1936
[CAMERA] fmt.fmt.pix.pixelformat = NV21
[CAMERA] fmt.fmt.pix.field = 1
[CAMERA_DEBUG] reqbuf number is 3
[CAMERA_DEBUG] map buffer index: 0, mem: 0xb679e000, len: 72db00, offset: 0
[CAMERA_DEBUG] map buffer index: 1, mem: 0xb6070000, len: 72db00, offset: 72e000
[CAMERA_DEBUG] map buffer index: 2, mem: 0xb5942000, len: 72db00, offset: e5c000
[CAMERA] stream on succeed
```

以下信息将提示当前拍照的照片索引以及从开启流传输到 dqbuf 成功的时间间隔：

```
[CAMERA] capture num is [0]
[CAMERA_DEBUG] *****DQBUF[0] FINISH*****
[CAMERA_PROMPT] the time interval from the start to the first frame is 196 ms
[CAMERA_DEBUG] the interval of two frames is 0 ms
[CAMERA_DEBUG] *****QBUB[0] FINISH*****
```

以下信息提示该 sensor 的实际测量帧率信息：

```
[CAMERA_DEBUG] *****
[CAMERA_DEBUG] Query the actual frame rate.
[CAMERA_DEBUG] camera fps = 22.
[CAMERA_DEBUG] *****
```

以下信息提示从 open 节点到可以得到第一帧数据的时间间隔，默认设置为测试拍照的相应设置：

```
[CAMERA_DEBUG] *****
[CAMERA_DEBUG] Performance Testing---format:NV21 size:2592 * 1936
[CAMERA_DEBUG] The interval from open to streaming is 345 ms.
[CAMERA_DEBUG] *****
```



### 3.4.文件保存格式

设置完毕之后，将会在所设路径（默认 /tmp）下面保存图像数据，数据分别有两种格式，一种是 YUV 格式，以 source\_格式.yuv 名称保存；一种是 BMP 格式，以 bmp\_格式.bmp 格式保存，如下图所示。

```
root@TinaLinux:/tmp# ls
TZ                booting_state    run              source_NV21_5.yuv
bmp_NV21_1.bmp    lib              shm              state
bmp_NV21_2.bmp    lock             source_NV21_1.yuv tmp
bmp_NV21_3.bmp    log              source_NV21_2.yuv
bmp_NV21_4.bmp    resolv.conf      source_NV21_3.yuv
bmp_NV21_5.bmp    resolv.conf.auto source_NV21_4.yuv
root@TinaLinux:/tmp#
```



## 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 acknowledgment 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 datasheet neither states nor implies warranty of any kind, including fitness for any particular application.

