



# **Tina Linux Gstreamer 使用指南**

**版本号: 1.0**  
**发布日期: 2020-09-04**

## 版本历史

版本号	日期	制/修订人	内容描述
1.0	2019.07.23	AWA1020	创建



# 目 录

<b>1 概述</b>	<b>1</b>
1.1 编写目的 . . . . .	1
1.2 适用范围 . . . . .	1
1.3 相关人员 . . . . .	1
<b>2 Gstreamer Element</b>	<b>2</b>
<b>3 Gstreamer Configuration</b>	<b>3</b>
<b>4 Gstreamer Utils</b>	<b>6</b>
4.1 gst-inspect-1.0 . . . . .	6
4.2 gst-launch-1.0 . . . . .	8
<b>5 Allwinner 支持的编解码及显示框架</b>	<b>9</b>
<b>6 调试方法</b>	<b>10</b>
6.1 设置全局调试优先级 . . . . .	10
6.2 设置局部调试优先级 . . . . .	10
6.3 列举调试优先级的种类 . . . . .	10
6.4 查看图形化的 pipeline 通路: . . . . .	10

## 插 图

3-1 menuconfig 配置显示界面 . . . . .	3
3-2 multimedia 配置显示界面 . . . . .	3
3-3 gstreamer 配置显示界面 . . . . .	4
3-4 硬件编解码配置显示界面 1 . . . . .	4
3-5 硬件编解码配置显示界面 2 . . . . .	5
4-1 gst-inspect 显示界面 1 . . . . .	6
4-2 gst-inspect 显示界面 2 . . . . .	7
4-3 gst-inspect 显示界面 3 . . . . .	8



# 1 概述

---

## 1.1 编写目的

本文档主要介绍全志科技 Tina Linux SDK 的 Gstreamer 使用方法。

## 1.2 适用范围

Tina Linux SDK V3.0 之后的版本。

## 1.3 相关人员

Tina Linux 的开发人员、客户。

## 2 Gstreamer Element

Elements 是 gstreamer 的功能模块，每一个功能模块有创建 pipeline 的标准 API 和各不相同的 property; 由于 elements 多达 400 多个并且还不断增加，所以 gstreamer 官方对 elements 进行了分类，分类标准如下：

表 2-1: gstreamer elements 分类标准

包名	作用
Gstreamer	Gstreamer 核心库，提供创建管道的统一接口
Gst-plugins-base	Gstreamer 能够使用的基本 elements
Gst-plugins-good	基于 LGPL 的经过良好测试的高质量 elements
Gst-plugins-ugly	经过良好测试的 elements，但可能有版权问题
Gst-plugins-bad	未经过充分测试的库 elements
Gst-plugins-libav	基于 libav 的多媒体相关的 elements
Gst-omx	基于 openmax 用来支持硬解码的 elements
Gst-plugins-aw	基于全志私有接口用于多媒体的 elements

### 3 Gstreamer Configuration

不同的 elements 需要不同的配置，可以在功能开发完成后裁剪不需要的 elements 配置。打开 gstreamer elements 的方法如下：

#### (1) 在命令行中进入根目录，执行 **make menuconfig**

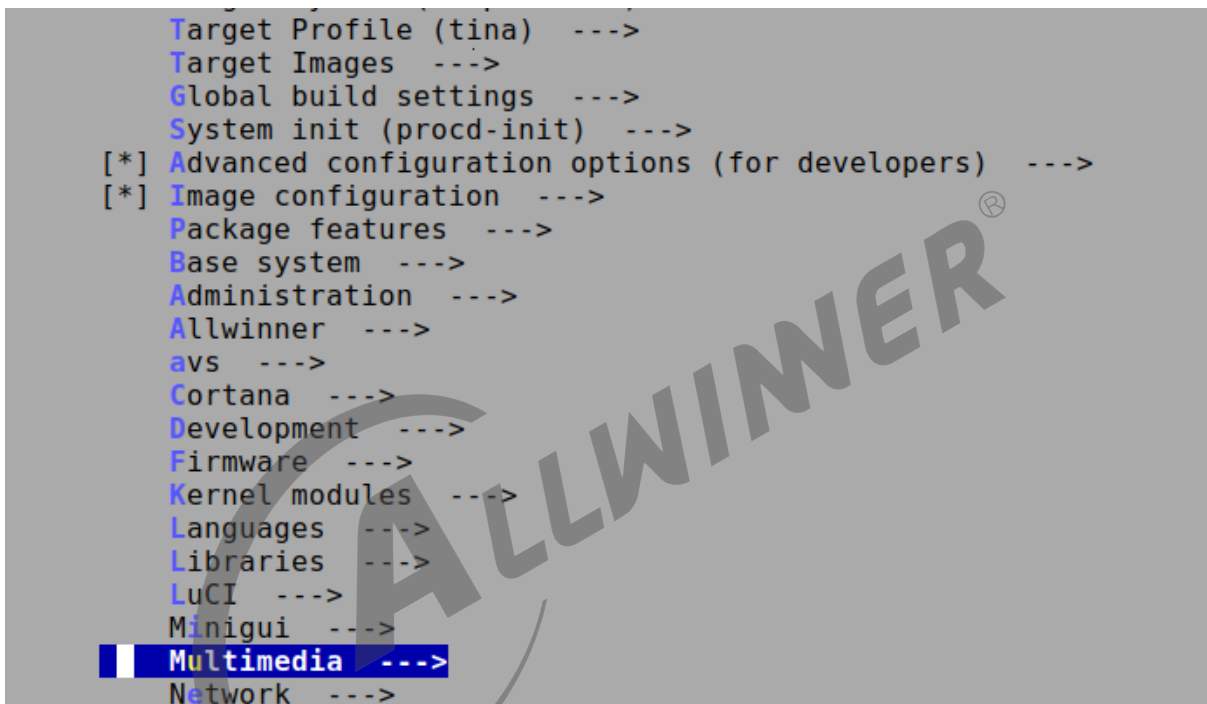


图 3-1: menuconfig 配置显示界面

#### (2) 选择 **Multimedia** 进入配置界面

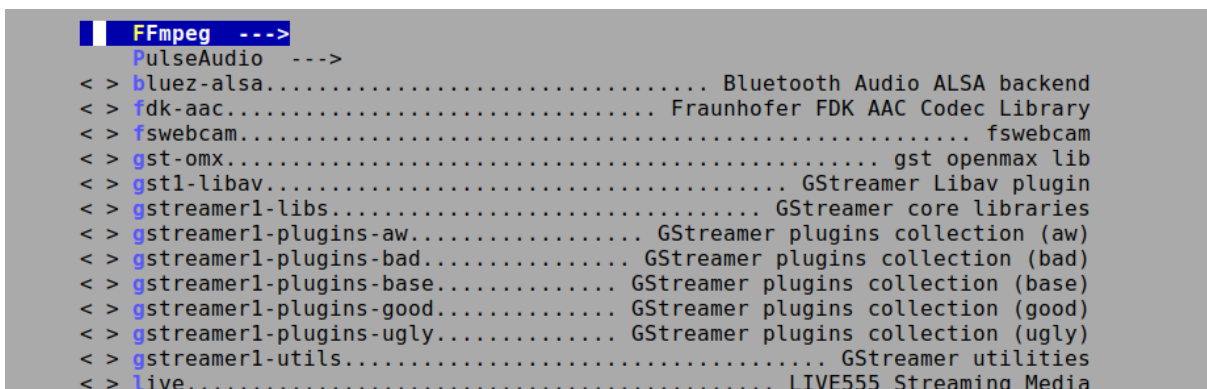


图 3-2: multimedia 配置显示界面

### (3) 勾选 gstreamer 相关包

```

Fmpeg --->
PulseAudio --->
< > bluez-alsa..... Bluetooth Audio ALSA backend
< > fdk-aac..... Fraunhofer FDK AAC Codec Library
< > fswebcam..... fswebcam
<*> gst-omx..... gst openmax lib
<*> gst1-libav..... GStreamer Libav plugin
    Select GStreamer libav configuration options --->
-.*- gstreamer1-libs..... GStreamer core libraries
    Select GStreamer libraries --->
<*> gstreamer1-plugins-aw..... GStreamer plugins collection (aw)
    Select GStreamer aw modules --->
<*> gstreamer1-plugins-bad..... GStreamer plugins collection (bad)
    Select GStreamer bad modules and libraries --->
-.*- gstreamer1-plugins-base..... GStreamer plugins collection (base)
    Select GStreamer base modules and libraries --->
<*> gstreamer1-plugins-good..... GStreamer plugins collection (good)
    Select GStreamer good modules --->
<*> gstreamer1-plugins-ugly..... GStreamer plugins collection (ugly)
    Select GStreamer ugly modules --->
<*> gstreamer1-utils..... GStreamer utilities
< > Live..... LIVE555 Streaming Media

```

图 3-3: gstreamer 配置显示界面

### (4) 配置硬件编解码

```

.config - Tina Configuration
> Allwinner

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

< > awsystrace..... awsystrace for Tina Linux
< > benchmarks..... Benchmark program
< > boot-play..... boot play
<*> camerademo..... camerademo test sensor ----
< > eyesee-minigui..... eyesee-minigui for Tina Linux
< > healthd..... Battery Daemon for tina linux ----
< > leakage_current..... A tool to check leakage current
-.*- libAWispApi..... camera VIN ISP api for Allwinner
< > libaec-xt..... Nuvoton Acoustic Echo Cancellation Algorithm ----
-.*- liballwinner-base..... base
< > liballwinner-ipc..... liballwinner ipc for tina parrot
< > libawadbauth..... adb auth lib for Tina Linux
< > libc-bench..... libc-bench from Android L
< > libcamapi..... camera api for Allwinner
-.*- libcedarx..... libcedarx for allwinner
    Select cedarx configuration options --->
        Select Cedarx memory options --->
        Choose whether to use VIN ISP (YES) --->
        Choose whether to use playready (NO) --->
< > libdisplay..... display testcase for allwinner ----
< > libminui..... tiny minui library ----
< > libresample..... libresample
<*> libsec_key..... secure key for allwinner ----
< > libspeex-lite..... tiny speex resample library
< > libsysutils..... libsysutils from Android L
-.*- libuapi..... unit api for Allwinner ----
< > libutils..... libutils from Android L
< > logwrapper..... logwrapper
< > mad-demo..... MAD demo
L(+)

<Select> < Exit > < Help > < Save > < Load >

```

图 3-4: 硬件编解码配置显示界面 1



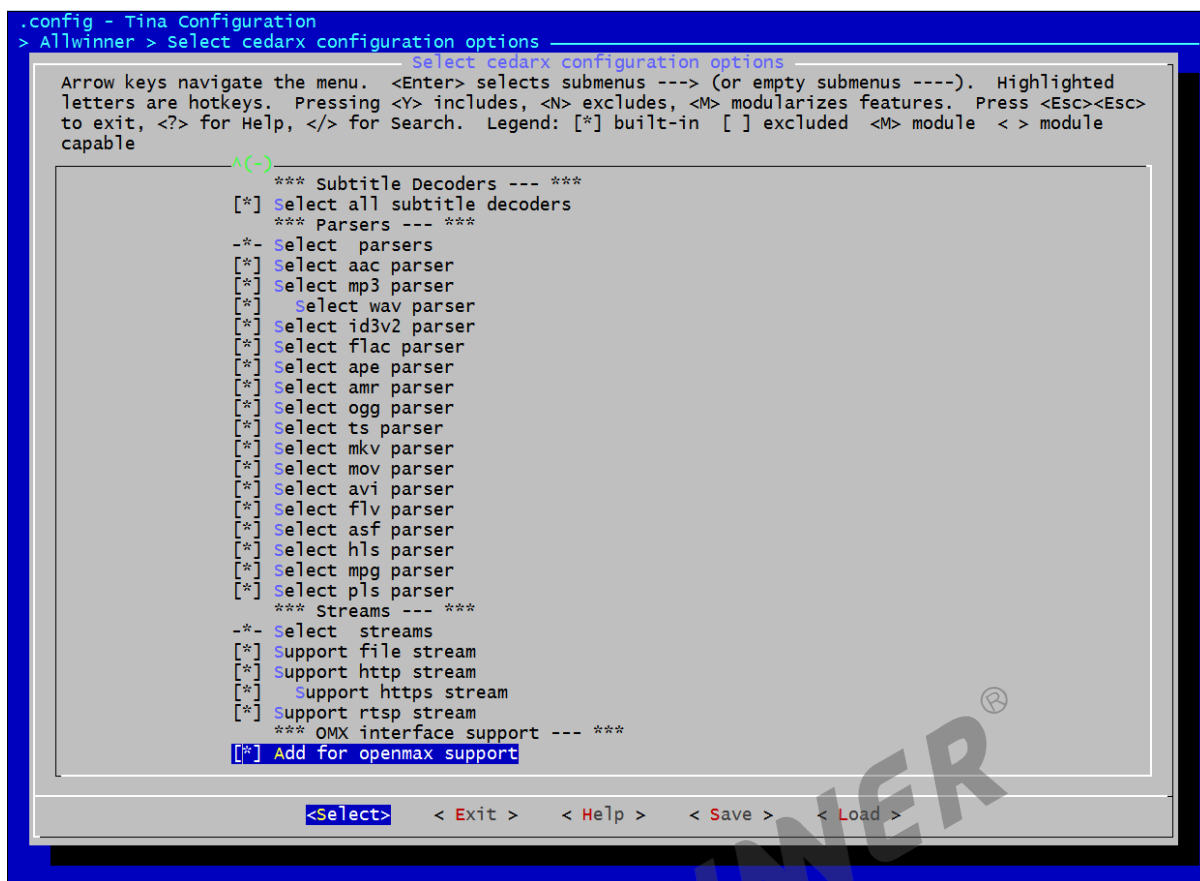


图 3-5: 硬件编解码配置显示界面 2

通过以上方式选中 openmax 即可使用全志的硬件进行视频编解码，如若不选中，则使用软件编解码。

#### (5) 保存配置后退出

开始编译相应方案并生成 gstreamer 相关文件

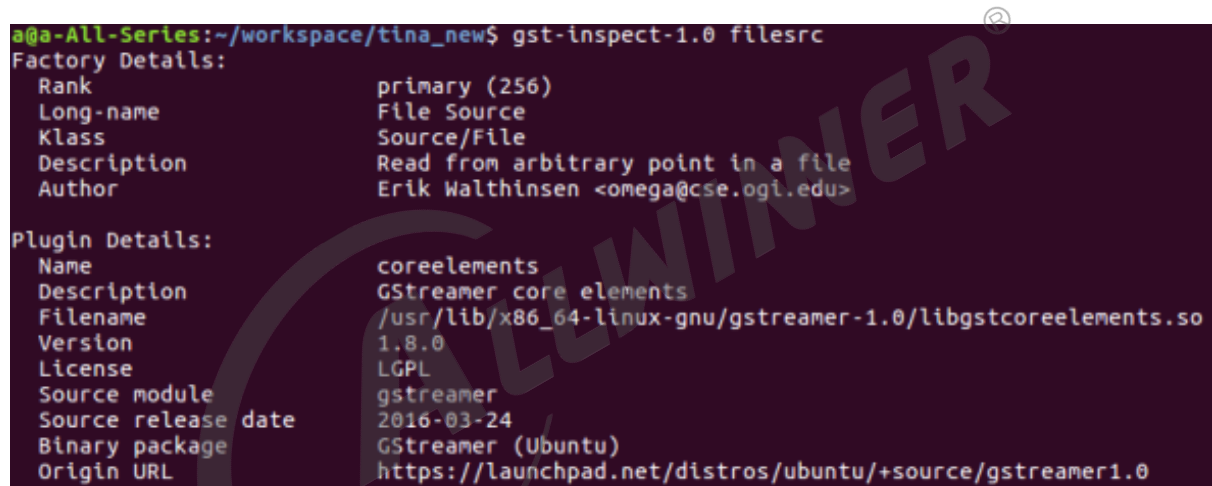
## 4 Gstreamer Utils

Gstreamer 自带的常用工具有 `gst-launch-1.0` 和 `gst-inspect-1.0`。

### 4.1 `gst-inspect-1.0`

(1) 通过 `gst-inspect-1.0` 可以查询系统中有哪些 `plugin` 可以使用

(2) `gst-inspect-1.0 [element-name]` 可以查看对应 `element` 的属性



```
a@a-All-Series:~/workspace/tina_new$ gst-inspect-1.0 filesrc
Factory Details:
  Rank:                primary (256)
  Long-name:           File Source
  Klass:               Source/File
  Description:         Read from arbitrary point in a file
  Author:              Erik Walthinsen <omega@cse.ogi.edu>

Plugin Details:
  Name:                coreelements
  Description:          GStreamer core elements
  Filename:             /usr/lib/x86_64-linux-gnu/gstreamer-1.0/libgstcoreelements.so
  Version:              1.8.0
  License:              LGPL
  Source module:        gstreamer
  Source release date:  2016-03-24
  Binary package:       GStreamer (Ubuntu)
  Origin URL:           https://launchpad.net/distros/ubuntu/+source/gstreamer1.0
```

图 4-1: `gst-inspect` 显示界面 1

表 4-1: `element` 部分属性解析

Rank	具有同样功能 <code>element</code> 时选择 Rank 值更大的 <code>element</code>
Filename	<code>elemnt</code> 对应的库所在的位置
Source module	<code>element</code> 对应的分类

```
GObject
+----GInitiallyUnowned
      +----GstObject
            +----GstElement
                  +----GstBaseSrc
                        +----GstFileSrc

实现的接口:
    GstURISHandler

Pad Templates:
    SRC template: 'src'
    Availability: Always
    Capabilities:
        ANY

Element Flags:
    no flags set

Element Implementation:
    Has change_state() function: gst_base_src_change_state

Element has no clocking capabilities.

URI handling capabilities:
    Element can act as source.
    Supported URI protocols:
        file

Pads:
    SRC: 'src'
    Pad Template: 'src'
```

图 4-2: gst-inspect 显示界面 2

GObject 部分为继承图，Pad Template 表示数据管道可以支持的输入输出数据格式；实现的接口表明他实现了基类哪些相关的功能；Element has no clocking capabilities. 表示此 element 不提供时钟源。

```
Element Properties:
  name      : The name of the object
              flags: 可读, 可写
              String. Default: "filesrc0"
  parent    : The parent of the object
              flags: 可读, 可写
              Object of type "GstObject"
  blocksize  : Size in bytes to read per buffer (-1 = default)
              flags: 可读, 可写
              Unsigned Integer. Range: 0 - 4294967295 Default: 4096
  num-buffers : Number of buffers to output before sending EOS (-1 = unlimited)
              flags: 可读, 可写
              Integer. Range: -1 - 2147483647 Default: -1
  typefind   : Run typefind before negotiating
              flags: 可读, 可写
              Boolean. Default: false
  do-timestamp : Apply current stream time to buffers
              flags: 可读, 可写
              Boolean. Default: false
  location   : Location of the file to read
              flags: 可读, 可写, 只能在NULL、或READY状态下改变
              String. Default: null
```

图 4-3: gst-inspect 显示界面 3

上述为 element 的 property, 可以改写 element 的属性来指定数据流向时的操作。

## 4.2 gst-launch-1.0

通过 gst-launch-1.0 命令来创建数据通路:

```
gst-launch-1.0 filesrc location=/tmp/test.avi ! fakesink
```

使用! 来链接各个 element, 使用空格来指定前面 element 对应的 property。

```
gst-launch-1.0 videotestsrc ! video/x-raw,format=YV12 ! videoconvert ! Xvimagesink
```

使用 capsfilter 来指定穿过两个 element 数据的格式。

## 5 Allwinner 支持的编解码及显示框架

### (1) 播放视频可直接使用 playbin

```
gst-launch-1.0 playbin uri=file:///tmp/test.mkv
```

### (2) 全志支持硬解码方式为通过 **gst-omx** 来硬解码和硬编码

通过在小机端 `gst-inspect-1.0 | grep omx` 命令来查看支持的硬解码格式。

### (3) 全志支持通过私有的 **display** 接口和 **wayland** 显示数据

私有显示接口对应的 plugin 为 `sunxifbsink`。

```
gst-launch-1.0 playbin uri=file:///tmp/test.mkv video-sink=sunxifbsink
```

wayland 对应的 plugin 为 `waylandsink`。

```
gst-launch-1.0 playbin uri=file:///tmp/test.mkv video-sink=waylandsink
```

Wayland 基于 DRM 显示框架，`sunxifbsink` 基于 `fbdev`，所以在编译阶段只能选择一个（可通过 `gststreamer` 配置显示界面进行选择）。

### (4) 其他常用的 demo

从摄像头采集视频，同时硬编码为 h264，封装为 mkv 文件并保存于 tmp 目录下。

```
gst-launch-1.0 v4l2src ! video/x-raw,format=YV12,framerate=30/1,width=640,height=480 !  
  autovideoconvert ! omxh264videoenc ! h264parse ! video/x-h264,stream-format=avc,width  
  =640,height=480 ! matroskamux ! filesink location=/tmp/test.mkv
```

预览摄像头采集数据，同时硬编码为 h264，封装为 mkv 文件并保存于 tmp 目录下。

```
gst-launch-1.0 v4l2src ! omxmjpegvideodec ! tee name=srctee ! queue ! sunxifbsink srctee. !  
  autovideoconvert ! omxh264videoenc ! h264parse ! video/x-h264,stream-format=avc,width  
  =1280,height=720 ! matroskamux ! filesink location=/tmp/test.mkv
```

## 6 调试方法

### 6.1 设置全局调试优先级

```
gst-launch-1.0 videotestsrc ! fakesink --gst-debug-level=[number]
```

### 6.2 设置局部调试优先级

```
gst-launch-1.0 videotestsrc ! fakesink --gst-debug=[category_name]:[number]
```

### 6.3 列举调试优先级的种类

```
gst-launch-1.0 videotestsrc ! fakesink --gst-debug-help
```

### 6.4 查看图形化的 pipeline 通路:

#### (1) 导入 DOT 环境变量

```
export GST_DEBUG_DUMP_DOT_DIR=./
```

#### (2) 执行命令建立管道, 停止管道后查看当前目录下的.dot 文件

#### (3) 将 dot 转成 png

```
dot -Tpng ./xxx.dot >test.png
```

#### (4) 查看 png

```
eog test.png
```






## 著作权声明

版权所有 © 2020 珠海全志科技股份有限公司。保留一切权利。

本文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。

本文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、发表或传播本文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。在本文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。本文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，并严格遵循本文档的使用说明。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。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（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。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文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。