



Tinalinux

Gstreamer 使用指南

1.0
2019.07.23

文档履历

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

目录

1. 概述	1
1.1 编写目的	1
1.2 适用范围	1
1.3 相关人员	1
2. Gstreamer Element	2
3. Gstreamer Configuration	3
3.1 在命令行中进入根目录，执行 <code>make menuconfig</code>	3
3.2 选择 Multimedia 进入配置界面	4
3.3 勾选 gstreamer 相关包	4
3.4 保存配置后退出	4
4. Gstreamer Utils	5
4.1 <code>gst-inspect-1.0</code>	5
4.1.1 通过 <code>gst-inspect-1.0</code> 可以查询系统中有哪些 plugin 可以使用	5
4.1.2 <code>gst-inspect-1.0 [element-name]</code> 可以查看对应 element 的属性。	5
4.2 <code>gst-launch-1.0</code>	7
4.2.1 通过 <code>gst-launch-1.0</code> 命令来创建数据通路。	7
5. Allwinner 支持的编解码解码及显示框架	8
5.1 播放视频可直接使用 <code>playbin</code>	8
5.2 全志支持硬解码方式为通过 <code>gst-omx</code> 来硬解码和硬编码	8
5.3 全志支持通过私有的 <code>display</code> 接口和 <code>wayland</code> 显示数据	8

5.4 其他常用的 demo	8
6. 调试方法	10
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
6.4.1 导入 DOT 环境变量	10
6.4.2 执行命令建立管道, 停止管道后查看当前目录下的 .dot 文件	11
6.4.3 将 dot 转成 png	11
6.4.4 查看 png	11
7. Declaration	12

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 多个并且还不断在增加，所以 gstreamer 官方对 elements 进行了分类，分类标准如下：

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：

3.1 在命令行中进入根目录，执行 make menuconfig

```
Target Profile (tina) --->
Target Images --->
Global build settings --->
System init (procd-init) --->
[*] Advanced configuration options (for developers) --->
[*] Image configuration --->
Package features --->
Base system --->
Administration --->
Allwinner --->
avs --->
Cortana --->
Development --->
Firmware --->
Kernel modules --->
Languages --->
Libraries --->
LuCI --->
Minigui --->
Multimedia --->
Network --->
```

图 1: menuconfig 配置显示界面

3.2 选择 Multimedia 进入配置界面

```

FFmpeg --->
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
<> gstreamer1-libs..... GStreamer core libraries
<> gstreamer1-plugins-aw..... GStreamer plugins collection (aw)
<> gstreamer1-plugins-bad..... GStreamer plugins collection (bad)
<> gstreamer1-plugins-base..... GStreamer plugins collection (base)
<> gstreamer1-plugins-good..... GStreamer plugins collection (good)
<> gstreamer1-plugins-ugly..... GStreamer plugins collection (ugly)
<> gstreamer1-utils..... GStreamer utilities
<> live..... LIVE555 Streaming Media
    
```

图 2: multimedia 配置显示界面

3.3 勾选 gstreamer 相关包

```

FFmpeg --->
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: gstreamer 配置显示界面

3.4 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`

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

4.1.2 `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: `gst-inspect` 显示界面 1

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

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

实现的接口:
  GstURISrc

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 5: 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 6: 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

4.2.1 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 支持的编解码解码及显示框架

5.1 播放视频可直接使用 playbin

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

5.2 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。

5.3 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。

5.4 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 ! OMXJPEGvideodec ! 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:

6.4.1 导入 DOT 环境变量

```
export GST_DEBUG_DUMP_DOT_DIR=.
```

6.4.2 执行命令建立管道, 停止管道后查看当前目录下的.dot 文件

6.4.3 将 dot 转成 png

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

6.4.4 查看 png

```
cog test.png
```

7. 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 acknowledgement 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. tates nor implies warranty of any kind, including fitness for any particular application.