

Tina

GStreamer v1.0

文档履历

版本号	日期	制/修订人	制/修订记录
V1.0	2018.4.11	xiebin	



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. Tina gstreamer configuration.....	5
Gstreamer 配置方法.....	6
3. Tina gstreamer utils 使用方法.....	7
4. Allwinner 支持的编解码解码及显示框架.....	8
5. 调试方法.....	9



1. 概述

1.1. 编写目的

使 Tina gstreamer 开发人员了解在 tina 上使用 gstreamer 的基本方法

1.2. 适用范围

Tina gstreamer 相关配置及使用

1.3. 相关人员

多媒体开发人员



2. Tina gstreamer configuration

Gstreamer 的 elements 是 gstreamer 的功能模块, 每一个功能模块有创建 pipeline 的标准 API 和基于 gobject 的各不不同的 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。

包名	作用
gstreamer	Gstreamer 核心库, 提供创建管道的统一接口
Gst-plugins-base	让 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	基于全志私有显示 API 的用于多媒体显示 elements

打开不同的 elements 需要不同的配置, 可以根据需要裁剪。以下是打开所有 elements 相关包的方法。以下配置需要进入详细配置去打开你需要的 plugin 来使你的应用程序工作

Multimedia---->gst*

```

Target System (tulip-d1) --->
Target Profile (tina) --->
Target Images --->
Global build settings --->
System init (procd-init) --->
[ ] Advanced configuration options (for developer:
[ ] Image configuration --->
Package features --->
Base system --->
Allwinner --->
Development --->
DragonTools --->
Firmware --->
Kernel modules --->
Languages --->
Libraries --->
LuCI --->
Minigui --->
Multimedia --->
Network --->
Qt --->

```

Gstreamer 配置方法

```
< > ffmpeg..... FFmpeg
< > ffprobe..... FFmpeg CLI media
< > ffserver..... stream
< > fswebcam.....
<*> gst-omx..... gst
<*> gst1-libav..... GStreamer Libav
    Select GStreamer libav configuration options --->
- *- gstreamer1-libs..... GStreamer core
    Select GStreamer libraries --->
<*> gstreamer1-plugins-aw..... GStreamer plugins collect
    Select GStreamer aw modules --->
<*> gstreamer1-plugins-bad..... GStreamer plugins collect
    Select GStreamer bad modules and libraries --->
- *- gstreamer1-plugins-base..... GStreamer plugins collect
    Select GStreamer base modules and libraries --->
<*> gstreamer1-plugins-good..... GStreamer plugins collect
    Select GStreamer good modules --->
<*> gstreamer1-plugins-ugly..... GStreamer plugins collect
    Select GStreamer ugly modules --->
```



3. Tina gstreamer utils 使用方法

Gstreamer 自带的常用工具有 `gst-launch-1.0` 和 `gst-inspect-1.0`。通过 `gst-inspect-1.0` 可以查询系统中有哪些 plugin 可以使用。

`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
  Class               Source/File
  Description         Read from arbitrary point in a file
  Author              Erik Walthinsen <omega@cse.ogi.edu>

Plugin Details:
  Name                corelements
  Description         GStreamer core elements
  Filename             /usr/lib/x86_64-linux-gnu/gstreamer-1.0/libg
  Version             1.8.0
  License             LGPL
  Source module       gstreamer
```

Rank: 具有同样功能 element 时选择 Rank 值更大的 element

Filename: element 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:
  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_sta

Element has no clocking capabilities.

URI handling capabilities:
  Element can act as source.
```

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,此时钟源在同步时可以使用。

```
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: -1
  num-buffers    : Number of buffers to output before sending EOS
                  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
```

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。

另外一个工具为 gst-launch-1.0 用来通过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

4. Allwinner 支持的编解码解码及显示框架

Allwinner 播放器使用的数据通路如下,对于播放可以直接使用 playbin 来播放文件。

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

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。

全志支持通过私有的 display 接口和 wayland 来显示数据。私有显示接口对应的 plugin 为 sunxifbsink,wayland 对应的 sink 为 waylandsink。

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

Gst-launch-1.0 playbin uri=file:///tmp/test.mkv video-sink=waylandsink 来播放视频。

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。

其他常用的方法 demo 如下:

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 从摄像头采集视频并硬编码为 h264 存储到 mkv 文件中

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

预览摄像头采集数据,并同时硬编码为 h264 存储到 mkv 文件中。

5. 调试方法

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

设置全局调试优先级

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

设置局部调试优先级

```
gst-launch-1.0 videotestsrc ! fakesink --gst-debug-help 列举调试优先级的种类
```

查看图形化的 pipeline 通路:

```
1.export GST_DEBUG_DUMP_DOT_DIR=.
```

```
2.执行 gst-launch-1.0 命令建立管道,停止管道后查看当前目录下的.dot 文件
```

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

```
4.eog test.png
```

