

# Remote GStreamer Debugger

Marcin Kolny

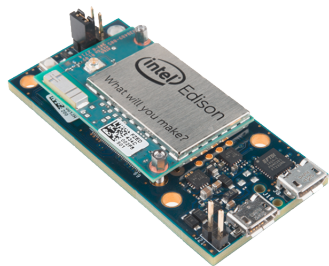
*marcin.kolny@gmail.com*

October 8, 2015

# Functions

- Display pipeline graph
- Watch data (events, queries, buffers, messages)
- Get/Set element properties

Should work both with local and remote pipelines.



# Implementation - trace plugin

## GST\_TRACE plugin

```
$ GST_TRACE=debugserver totem
```

```
$ GST_TRACE=debugserver gst-launch-1.0
```

```
$ # any GStreamer-based application
```

# Implementation - GTK+ Application

The screenshot displays the GStreamer Debugger interface. The main window title is "GStreamer Debugger". The menu bar includes "File", "Edit", "Server", and "Help". The toolbar contains "Log messages", "Queries", "Bus messages", "Buffers", and "Events".

The "Buffers" tab is active, showing a list of buffers:

- Buffer of size 33
- Buffer of size 7
- Buffer of size 33
- Buffer of size 23
- Buffer of size 67

The "Selected element" is "playsink" and the "Current bin" is "/playsink/". A "Refresh graph" button is visible.

The pipeline graph shows the following elements and connections:

- streamssynchronizer0 (blue box) connects to sink\_1 (blue box), sink\_0 (blue box), and sink\_2 (blue box).
- src\_0 (red box) connects to sink\_0 (blue box).
- vdbin (white box) contains sink (blue box) and src (red box).
- abin (white box) contains src (red box).
- text (white box) contains src (red box).

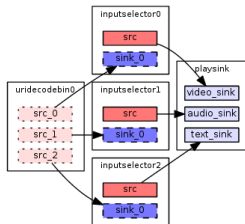
A data table is displayed with the following columns: Name, Value, and a hex dump. The hex dump shows data in columns of 12 bytes.

Name	Value	Hex Dump
Pts	12318970	20 20 20 20 20 20 20 20 20 20 20 20
Dts	18446744	57 65 6c 6c 2c 20 49 2c 20 75 68 2c
Duration	26350000	20 68 61 76 65 6e 27 74 0d 0a 20 20
Offset	18446744	20 20 20 20 20 20 20 20 20 20 20 20
Offset End	0	20 20 20 20 20 20 20 20 20 20 20 20
Data Size	67	
Object path	/inputselector2:src	
Data		20 20 20 20 20 20 20 20 20 20 20 20 2c 20 49 2c 20 75 68 2c 20 20 0d 0a 20 20 20 20 20 20 20 20 20 20 20 20 20 61 73 6b 65 6e 20 20

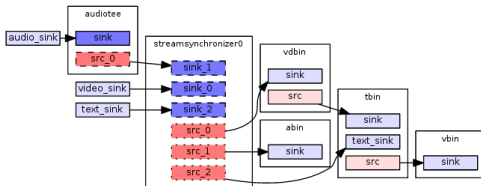
A dialog box titled "lt-gst-debugger-1.0" is open, showing the "Display format" set to "HEX" and "Columns in row" set to "12".

On the right side, there is a video player showing a scene from "Mr Robot (Season 1 Episode 1)". The subtitle reads: "Sorry. Credit card system".

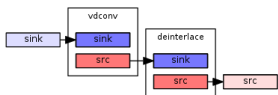
# Pipeline topology



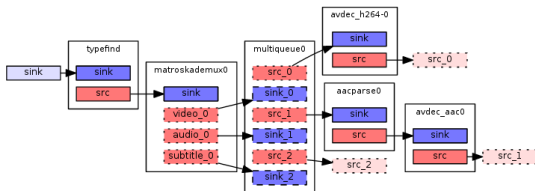
PATH: /



PATH: /playsink/



/playsink/vdbin/



PATH: /uridecodebin0/decodebin0/

# Buffers

The screenshot shows the `gst-debugger-1.0` interface. On the left, a list of buffers is displayed, with the selected one showing details:

Name	Value
Pts	4300000000
Dts	4300000000
Duration	3333333
Offset	1291
Offset End	0
Data Size	115200
Object path	/typefindelement0:src

Below the details, a hex dump of the buffer data is shown, consisting of multiple lines of hexadecimal values. The first few lines contain 'eb eb eb eb eb eb eb eb eb eb eb eb eb eb eb eb', followed by '10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10', and then '235 235 235 235 235 235 235 235'. The final line shows '016 016 016 016 016 016 016 016 016 016 016 016 016 016 016 016'.

On the right, a context menu is open, showing options like 'Add hook', 'Send data', and 'Pad'. The 'Send data' option is checked. Below the context menu, a hex dump window is open, displaying the same hex dump data as seen in the main interface. The window has a 'Close' button and a 'SET' button.

# Messages

Messages

Message of type: state-changed

Message of type: state-changed

Name	Value
message type	state-changed
message timestamp	99:99:99.999999999
message sequence number	403
old-state	playing
new-state	paused
pending-state	void-pending
message source	queue0

Type: GST\_MESSAGE\_EXTENDED

Add hook

Type

state-changed

Clear

Remove hook

# Queries

Queries

- Query of type: duration
- Query of type: duration
- Query of type: duration
- Query of type: duration
- Query of type: duration
- Query of type: duration

Type: GST\_QUERY\_DURATION

Pad: /typefindelement0:src

Add hook

Type	Pad
duration	

Name	Value
query type	duration
sent from pad	/decodebin0/typefind:src
format	3
duration	-1

Clear

Remove hook



# Events

Events

Event of type: navigation

Event of type: navigation

Event of type: qos

Event of type: qos

Event of type: ...

Name	Value
event type	navigation
event timestamp	99:99:99.999999999
event sequence number	463
event	mouse-move
button	0
pointer_x	296,000000
pointer_y	150,000000

Clear

Type: GST\_EVENT\_QOS

Pad: /capsfilter0:sink

Add hook

Type	Pad
navigation	
gap	
qos	/videotestsrc0:src
qos	/capsfilter0:sink

Remove hook

# Logs

logs

- gst\_bus\_timed\_pop\_filtered
- gst\_bus\_post
- gst\_bus\_source\_dispatch
- poll\_func
- gst\_bus\_source\_dispatch**

Name	Value
Level	5
Category name	GST_BUS
File	gstbus.c
Function	gst_bus_source_dispatch
Line	794
Object path	todo
Message	source 0x2705620 handler returns 1

Clear

Log level: GST\_LEVEL\_DEBUG

Debug category: GST\_ERROR\_SYSTEM

Add hook

Level	Category
LOG	GST_BUS
DEBUG	GST_ERROR_SYSTEM

Remove hook

Log threshold:

GST\_BUFFER:6,5

Overwrite current threshold

Set threshold

# Properties

name	capsfilter0	Refresh
parent	(GstObject:name) pipeline0	Refresh
qos	<input type="checkbox"/>	Refresh
caps	video/x-raw, framerate=(fraction)30/1	Refresh
caps-change-mode	immediate	Refresh
	delayed	

Plugins independent!

# Pads

Property Name	Property Value
Name	src
▼ Template	src
Presence	ALWAYS
Direction	SRC
▶ Caps	
Presence	ALWAYS
Direction	SRC
Peer pad	/capsfilter0:sink
▶ Current caps	
▼ Allowed caps	
▼ video/x-raw	
format	{ I420, YV12, YUY2, UYVY, AYUV, RGBx, BGRx, xRGB, xBGR, RGBA, BGRA, AF

- Bugzilla:  
[https://bugzilla.gnome.org/enter\\_bug.cgi?product=gst-debugger](https://bugzilla.gnome.org/enter_bug.cgi?product=gst-debugger)
- Releases:  
<ftp://ftp.gnome.org/pub/GNOME/sources/gst-debugger>
- GIT:  
<https://git.gnome.org/gst-debugger>