Embedded Streaming Media with Gstreamer and BeagleBoard

ESC-228
Presented by Todd Fischer
todd.fischer (at) ridgerun.com
Changes by Mark A. Yoder

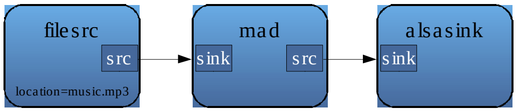
GStreamer

- Streaming media framework – audio and video
- Close to 200 (800?) plug-ins available
- Higher level than just input / filters / output
  - Networking, audio/video mixed streams, auto data handling
  - Various options utilizing hardware accelerators

GStreamer Overview

- Elements
  - Sources, filters, sinks
- Bins and Pipelines
  - Containers, pipeline is the overall bin
- Pads
  - Element source / sink connection points
- Caps
  - Capabilities organized by stream type with a set of properties

Simple MP3 Player



- Create dynamically using **`gst-launch`**
- Source element reads from a file
- Filter element converts MP3 to PWM
- Sink element passes to ALSA output

Hands On xM Exercise 0

- Start up Windowing environment with two terminals

```
beagle$ cd exercises/gstreamer
beagle$ git pull
```

- Play audio file

```
beagle$ ./a2
```

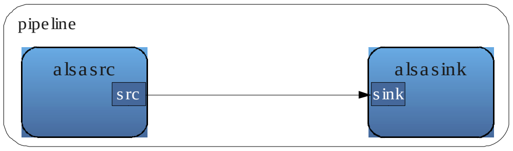
- Actual command

```
beagle$ cat a2
gst-launch filesrc location=bbb.flac ! flacdec ! alsasink
```

- There are lots of audio scripts to try
  - a1, a2, a3, a4, a5

Simple PA System

- Create dynamically using **`gst-launch`**
- Source element ALSA audio in
- No filters
- Sink element passes to ALSA output



Keeping Plug-ins Organized

- Each known plug-in is added to registry
- Most aspects of plug-in are tracked in the registry
- Registry support run-in pipeline creation and dynamic filter selection
- Use **`gst-inspect`** to list plug-ins

Hands On Exercise 1

- Using **`gst-inspect`**, list
  - All plug-ins
  - All video plug-ins
  - Element properties for **`filesrc`** plug-in

Hands On Exercise 2

- GStreamer video pipelines

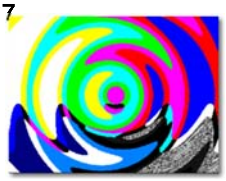
```
beagle$ ./v1
beagle$ cat v1
gst-launch videotestsrc ! ffmpegcolorspace ! fbdevsink
```

- Other video scripts

```
v1, v2, v3, v4, v5, v6, v7
```

- Idea is the same
  - source data, filter data, send data to sink
- Network demo

```
n1, n2, n3, n4
```



Hands On Exercise 2

- Network demo

```
n1, n2, n3, n4
gst-launch audiotestsrc freq=1000 !
mulawenc ! rtppcmupay ! udpsink
host=$HOST port=5555

gst-launch udpsrc port=5555
caps="application/x-rtp" ! queue !
rtppcmudepay !  mulawdec  ! alsasink
```

Performance - Data Passing

- Stream held in buffers with data, timestamp, other info
- When possible, buffer memory allocated by sink pad
- Use hardware when data copy is necessary

Performance - Data Transformation

- Cortex A8 compiler optimization
- NEON
  - Single Instruction Multiple Data
- C64
  - Video accelerator
- DMA and other data movers