

▶ GStreamer Conference 2026 · Lightning Talk

Embedded Reality

GStreamer Versioning Strategy for SoC-Based Products

Kye Yeon Ko
amy.ko@lge.com

The Reality : Upstream moves fast, embedded products follow at their own pace

SoC Vendor Reality

HW codec drivers tied to a specific GStreamer version

Upgrading GStreamer = re-validating full HW integration

Product Team Reality

Aligned with SoC vendor release

Shared across TV, mobile, and any SoC-based product

: Gap between upstream and deployed version widens

even version lags behind, How staying close to upstream?

Strategy 1 – Reference Version Management

We track Yocto LTS releases to select our GStreamer reference versions

Yocto Version	Codename	LTS Support	GStreamer	Our Status
4.0	Kirkstone	2022 – 2026	1.20.x	✓ Current product
5.0	Scarthgap	2024 – 2028	1.22.x	✓ Next SoC preparation
6.0	Wrynose	2026 – 2030	1.28.x	– Monitoring

reference
1.20.x

- Based on Yocto 4.0 LTS
- mono repository support

reference
1.24.x

- Based on Yocto 5.0 LTS
- stability improvements
- webOS-specific feature requirements



gst-omx

Deprecated upstream, but required for some SoC HW codec integration.
We maintain it – keeping it as close to SoC release form as possible.

Principle: Stay as close to SoC form as possible → easier patch application & diff comparison

Strategy 2 – Backporting from Upstream

We monitor upstream and apply security patches and critical bug fixes to our reference versions

Feature & Bug Fix Backport

20 Bug fixes
+

Example: v4l2videodec – Dynamic Resolution Change

- Pipeline stall from missing event handling
- Decoder hang during trick-play
- Infinite loop on cap negotiation failure

Also: HLS/DASH seeking, waylandsink, GStreamer core bug fixes backported

Security (CVE) Backport

40 Unpatched CVEs
+

15 High severity
+

Severity breakdown:

Critical / High

Use-after-free, Stack/Heap buffer overflow, OOB write

Medium

NULL pointer deref, OOB read, Integer overflow

We don't wait for upstream to knock

The version may lag but The quality doesn't have to.

Strategy 3 – Bugs Found, Fixed, Contributed Upstream!

Bugs found while maintaining our reference versions → fixed and contributed upstream

9

Contributors

54

Patches merged

Category	Count	Example
Resource Leak	14	fd / memory / resource leak fixes
Concurrency	8	Missing mutex unlock in multiple shutdown paths (gst-plugins-base, shapewipe)
Parsing / Security	7	Out-of-bounds access, bounds check & validation for rtcp / rtpbuffer / sdp / rtsp / qtdemux ESDS
Memory Safety	6	USE_AFTER_FREE, read from pointer after free
Feature Fix	2	h264/h265/h266parse HDR SEI caps update, HEVC mime codec bit shifting

Staying close to upstream reduces our own maintenance burden.

PROPOSAL : Please Consider GStreamer LTS Support

Like Yocto, we ask the GStreamer community to provide Long-Term Support versions

EU Cyber Resilience Act (CRA)

Adopted 2024 · Enforcement from 2027

Security updates for the entire lifetime

SoC-locked versions still need CVE patches



1.20.x	`22.02–`23.07	17 month
1.22.x	`23.01–`24.04	15 month
1.24.x	`24.03–`25.06	15 month
1.26.x	`25.03–`26.03	12 month

Yocto LTS

Yocto 4.0 (Kirkstone)	2022–2026
Yocto 5.0 (Scarthgap)	2024–2028
Yocto 6.0 (Wrynose)	2026–2030

What LTS support would mean for embedded teams:

No more self-backporting

CVE patches applied by upstream maintainers — accurate, timely

CRA compliance

Security update obligation met without extra engineering effort

Ecosystem trust

GStreamer becomes viable for safety-critical & regulated products

Embedded teams can't choose the version. But the community can help them stay secure.

Thank you