Linaro Connect
MADRID 2024 | MAY 14-17 2024

Pixel 6 and Tensor G1 SoC upstreaming status and plans

Peter Griffin (Linaro)
Will McVicker (Google)

Agenda

- Pixel 6 on android-mainline
- Pixel 6 support upstream
- Google Landing Team highlights
- What’s next?

What is android-mainline?

- `android-mainline` is the primary branch for new Android kernel development.
- Closely follows the upstream Linux kernel
- Android Common Kernel LTS branches are cut from `android-mainline`, e.g. `android15-6.6`

Android kernel team follows a strict ***Upstream First*** guideline.

Technical Debt includes:

- Android kernel features blocked from upstreaming due to no “in-tree user” and applicable to all future Android versions.
  - Scheduler vendor hooks
  - Ashmem technical debt
  - Overlayfs
  - etc.

Pixel 6 on android-mainline

Pixel 6 has been supported on android-mainline for almost 3 years!

All `android-mainline` changes are tested on Pixel 6 in presubmit and postsubmit.

- ~80k CTS and VTS tests
- Performance benchmarking tests

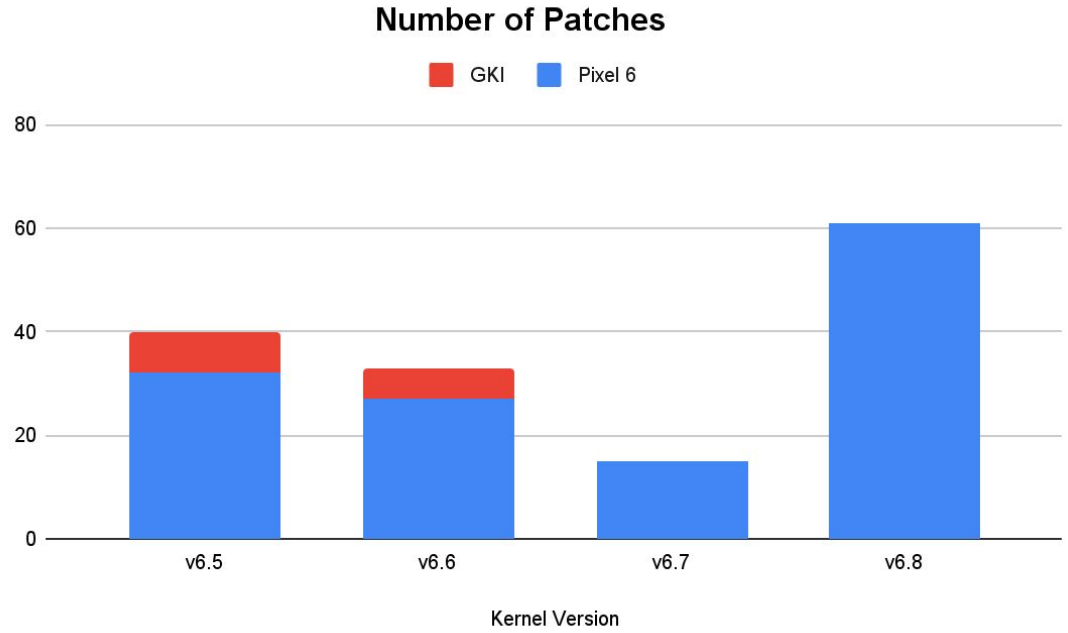
At least 16 ***upstream*** bugs found by testing Pixel 6 on `android-mainline` in CI!

What are the advantages?

- Increased GKI testing on real in-market hardware.
  - Catch regressions from upstream merges before they propagate to Android vendors!
  - Support on android-mainline automatically provides support on the next LTS branch.
  - Testing on form-factor hardware
    - Better real-world performance, thermal, and power testing than on a development board.
- Allows Android development on a kernel very close to upstream
  - Helps Android developers be more involved in upstream development and discussions.
- Easier in-market kernel upgrades
  - Easier to maintain functionality and performance incrementally for each Linux release, than to upgrade from one LTS release to the next.

Pixel 6 patches per Linux release

Graph gives an indication of the effort required to maintain out-of-tree drivers incrementally.



Give it a try!

The Pixel 6 android-mainline project has created various *public* resources

- Public documentation for building and flashing Pixel kernels.
  - https://source.android.com/docs/setup/build/building-pixel-kernels
- Periodically updating the AOSP android-mainline Pixel 6 kernel drivers.
  - https://android.googlesource.com/kernel/manifest/+/refs/heads/gs-android-gs-raviole-mainline
- Public resources provide benefits to the wider ecosystem!
  - Linaro KWG and LCG engineers can test upstream scheduler patches with the latest kernel on a fully functioning, form-factor device.
  - Developer boards often lack hardware found on real devices.
    - Less thermally constrained
    - Wall-power vs battery powered

Upstreaming of Tensor G1 and Pixel 6 (Oriole)

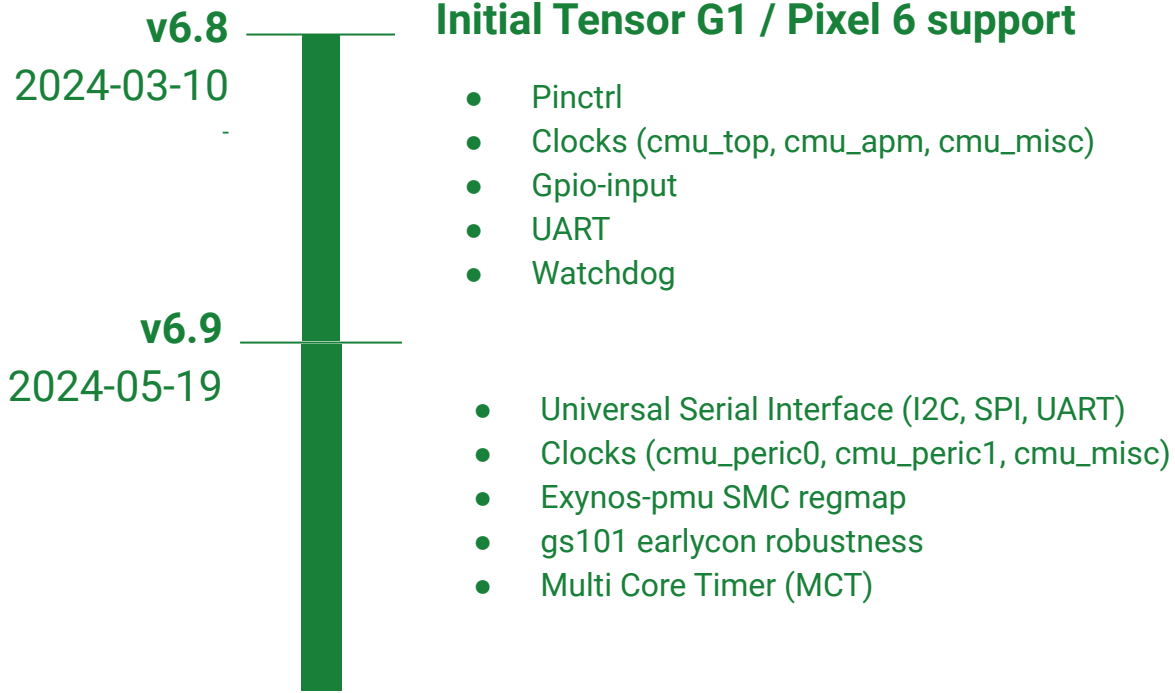
Why upstream Pixel 6?

- Allows kernel testing even earlier (during Linux -rc's, linux-next etc)
- Easier bisection of kernel regressions
- Helps reduce Android kernel technical debt
  - Some downstream Android features rely on specific hardware that isn't present upstream resulting in NACKs due to no "in-tree user".

Long Term Goal

Enable a **fully** upstream Android development platform on Pixel 6.

Upstreaming status



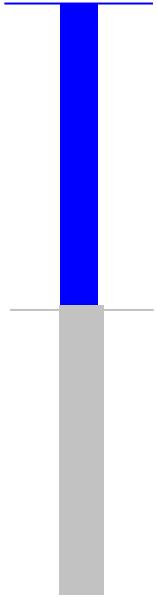
Initial Tensor G1 / Pixel 6 support

- Pinctrl
- Clocks (cmu_top, cmu_apm, cmu_misc)
- Gpio-input
- UART
- Watchdog

- Universal Serial Interface (I2C, SPI, UART)
- Clocks (cmu_peric0, cmu_peric1, cmu_misc)
- Exynos-pmu SMC regmap
- gs101 earlycon robustness
- Multi Core Timer (MCT)

Upstreaming status

v6.10
2024-07-28



Queued (as of 25 April)

- Define all USI nodes
- UFS phy and bindings

Under review (target v6.10)

- More clocks (cmu_hsi0, cmu_hsi2)
- UFS host driver and bindings
- USB phy driver and bindings
- DWC3 exynos glue and bindings

What’s coming next upstream?

Over the coming year we plan to upstream the following drivers:

- ACPM
  - Mailbox
  - PMIC
  - DVFS
  - Thermal
  - Power Domains
- More clocks :)
- IOMMU
- DRM Display
- Graphics
- Audio
- Touch
- PCIe & PHY

Upstream development tooling

- Various tooling developed to make upstream kernel development easier on Pixel 6.
  - Makefile based build system https://git.codelinaro.org/linaro/googlelt/pixelscripts

```
> make help
Build and optionally flash Pixel 6 images
The following make targets exist:
    clean              clean all generated files
    build              build all kernel artefacts
    flash              flash and run images (potentially rebuilding them first)
    boot               flash vendor_boot and boot boot.img (potentially rebuilding images first)
    dt_binding_check_vendor_exynos
    dt_binding_check_vendor_google
    dtbs_check_oriole validate Pixel6 DTs
    validate_dt        run checks of DT bindings and DTs related to Pixel6
    [..]
```

Upstream development tooling

- Yocto meta-bsp-google-gs layer
  - Machine layer with all Yocto configuration as appropriate for gs101
  - ./setup-environment script to get building quickly
- Allows generation of busybox initramfs
- Recipes for
  - Kernel
  - 3 different initramfs' (for different workflows)
  - Supporting tools like mkbootimg.py
- mkbootimg.py integration
  - recipe
  - class to use to allow Yocto to build flashable Pixel images (boot.img and vendor_boot.img)

Future Plans

- Enablement of Pixel 6 in kernel CI services
  - lkft.linaro.org
  - kernelci.org
  - Android Common Kernel presubmit and postsubmit
- Continue upstreaming Pixel 6 device drivers
- Replacement of out-of-tree drivers with upstreamed drivers

For instructions on running upstream kernel on your Pixel 6 device follow the **README.md** at https://git.codelinaro.org/linaro/googlelt/pixelscripts.

Come see an awesome Pixel 6 demo and chat with us on Demo Friday!

Thanks to

- All the kernel maintainers, reviewers, testers on the mailing list
  - Krzysztof Kozlowski
  - Rob Herring
  - Sam Protsenko
  - All the folks at Google

Linaro Connect
MADRID 2024 | MAY 12-17 2024

Thank you