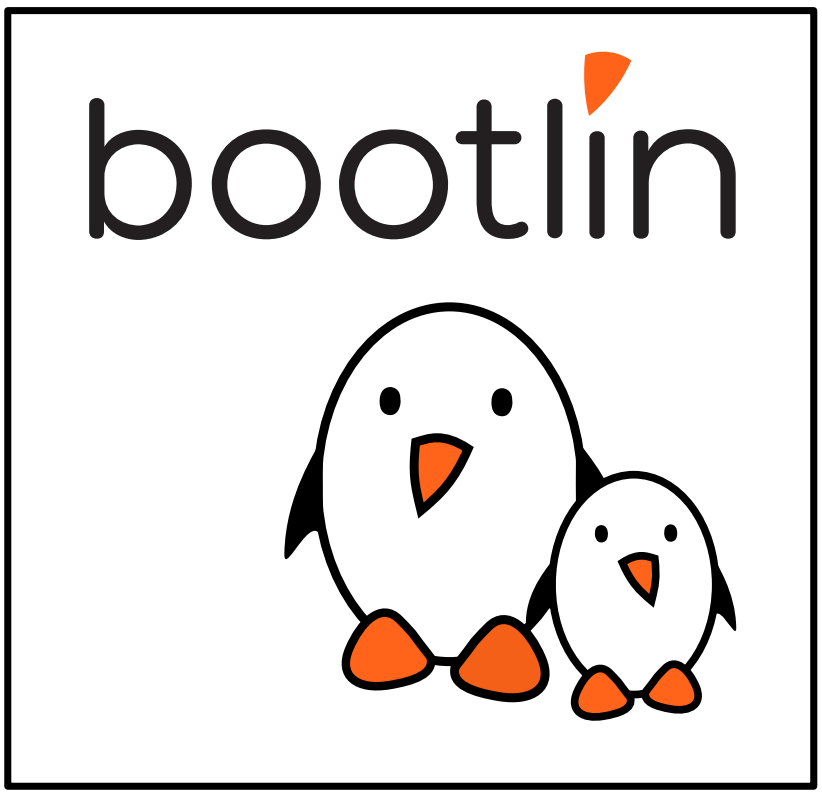
DRM DP link training properties

Köry Maincent
kory.maincent@bootlin.com

Display Next Hackfest 2026



Köry maincent

- Embedded Linux engineer at **Bootlin**
  - Development, consulting and training about **embedded Linux**
  - Open-source focus
- **Linux kernel** device driver developer
- Bootloaders, Buildroot and Yocto integration
- Open-source contributor
- Living in **Toulouse**, France
- New in the DRM subsystem and in DisplayPort world

DisplayPort link training

How it works

Iterative Process:

- Sink reads training patterns and reports status via AUX channel
- Source adjusts parameters (voltage, pre-emphasis) based on feedback
- Repeats until success or max attempts reached
- Falls back to lower link rate/lane count if training fails

=> The negociated link characterstics might not match the maximum capabilities between the sink and the source.

- Noise in the cable, length of the cable.
- Other disturbances

Linux side

Current status

Every drivers report the information on their own.

- Mainly debugfs
- The design that cross the developper mind at development time

i915

```
# ls /sys/kernel/debug/dri/1/eDP-1
...
i915_dp_force_lane_count
i915_dp_force_link_rate
i915_dp_force_link_retrain
i915_dp_force_link_training_failure
i915_dp_link_retrain_disabled
i915_dp_max_lane_count
i915_dp_max_link_rate
...
```

dp_link_settings_read

```
/* function description
 * get/ set DP configuration: lane_count, link_rate, spread_spectrum
...
 * cat /sys/kernel/debug/dri/0/DP-x/link_settings
 *
 * It will list current, verified, reported, preferred dp configuration.
 * current -- for current video mode
 * verified --- maximum configuration which pass link training
 * reported --- DP rx report caps (DPCD register offset 0, 1 2)
 * preferred --- user force settings
```

```
      snprintf(rd_buf_ptr, str_len, "Current:  %d  0x%x  %d  ",
                      link->cur_link_settings.lane_count,
                      link->cur_link_settings.link_rate,
                      link->cur_link_settings.link_spread);
```

Goal and current proposal

Make it generic

- Know the capabilities (lanes number, rates supported, DSC supported) of each DP connectors to point the best one to the user.
- Report the “power” of the DP link.
- In MST case, userspace can know negotiated link parameters and be smarter in choosing resolutions on each of the downstream sinks based on the total link bandwidth between source and the hub.
- Other usages? Tell me.

Development Support proposition

DRM connector properties

- Bitmask properties for lane number and link rates. Shows the controller capabilities when not plugged. Shows the negociated link parameters when a display is plugged.
- Bool property for DSC. Present only if supported. (Maybe should not be under the connectors.)
- Read only for now can be change to read write in the future if needed

Questions

- Should we think about reporting all display + link capabilities?
  Most (all?) controller driver currently does not scan every possibilities but stop when it reach a working link configuration.
- MST case. Should all connectors in-between have the DRM properties set to the identical value?

Current development

- [PATCH RFC 00/12] Add support for DisplayPort link training information report
- V2 nearly ready.

Thank you!

Questions?

Köry Maincent

kory.maincent@bootlin.com

https://bootlin.com/pub/conferences/