



Link training and firmware update

Louis Chauvet

louis.chauvet@bootlin.com

Display Next Hackfest 2026

© Copyright 2004-2026, Bootlin.

Creative Commons BY-SA 3.0 license.

Corrections, suggestions, contributions and translations are welcome!





Issue description



Display port in brief

Display port can:

- ▶ transport data
- ▶ side channel communication



Display port in brief

Display port can:

- ▶ transport data
- ▶ side channel communication

But to achieve fast speed, a link training is required:

- ▶ different kind of cables
- ▶ different quality of connectors
- ▶ different hardware
- ▶ ...



Issue

Link training uses side channel to work



Issue

Link training uses side channel to work

fwupd can use side channel to upgrade a firmware



Link training uses side channel to work

fwupd can use side channel to upgrade a firmware

Problem

- ▶ fwupd tries to update during a link training
- ▶ link training happen during fwupd



Proposed solution



Core modifications

```
// include/drm/display/drm_dp_helper.h
struct drm_dp_aux {
    /**
     * @lock: mutex for protecting AUX device operations
     */
    struct mutex lock;
}
// drivers/gpu/drm/display/drm_dp_aux_dev.c
static int auxdev_open(struct inode *inode, struct file *file) {
    mutex_lock(&aux_dev->aux->lock);
}
// drivers/gpu/drm/display/drm_dp_aux_dev.c
static int auxdev_release(struct inode *inode, struct file *file) {
    mutex_unlock(&aux_dev->aux->lock);
}
```



Per-driver modification

```
// In amd/.../link_dp_training.c
bool perform_link_training_with_retries(...) {
    scoped_guard(mutex, &dsc_aux->lock) {
        <link training code>
    }
}

// In i915/.../intel_dp_link_training.c
void intel_dp_start_link_train(...) {
    scoped_guard(mutex, &intel_dp->aux.lock) {
        <link training code>
    }
}c
```



Limitation

Userpace can DoS the link training: open `drm_dp_aux` and never close it

Other possible solution: mutex around sysfs read/write, but no guarantee for fwupd to not be interrupted by link training / link training may happen with a bad hardware state.

Let's discuss it !

Louis Chauvet

Slides under CC-BY-SA 3.0

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