

Porting of drivers/leds/leds-kttd2692.c

May 25, 2017

1 Introduction

The driver `drivers/leds/leds-kttd2692.c` was introduced in 2013 in the commit `b7da8c5`. It DOES WHAT? This driver was accompanied by WHAT other files. Gcc produces HOW MANY errors/warnings that reduce to HOW MANY errors/warnings.

2 `devm_gpiod_get`

Gcc reports that the function `devm_gpiod_get` has too few arguments. We try the following patch query (`step1.cocci`):

```
@bad depends on before || after@
flexible expression list[n] es;
@@

devm_gpiod_get(es)

@depends on !bad && (before || after)@
expression e;
@@

devm_gpiod_get(...,
+ e,
...)
```

This is the same issue as considered in `drivers_phy_phy-tusb1210.c_1c14905`. Here, however, there are no calls to `aninput` or `output` function near the `devm_gpiod_get` call. There are, however, calls to `gpiod_direction_output` scattered throughout the code, with varying second arguments. We previously noted the possible `devm_gpiod_get` argument `GPIOD_ASIS`. The first result is one only fixing the current driver so we pretend that it is not there. The first result after that is `8a68771` at 50% that includes the `GPIOD_ASIS` value. Looking through the snapshot of the affected file at the time for the commit shows that it also calls `gpiod_direction_output` in various places with various second arguments. We thus follow that model and make the following change:

```
@@
expression e1,e2;
@@

devm_gpiod_get(e1,e2
+ , GPIOD_ASIS
)
```

=== success 1/16, 8a68771

3 struct led_classdev.brightness_set_sync

Gcc reports that the field `brightness_set_sync` of the structure `struct struct led_classdev` is not known. We try the following patch query (step2.cocci):

```
@bad1 depends on after@
struct led_classdev *e;
@@

    e->brightness_set_sync

@bad2 depends on after@
struct led_classdev e;
@@

    e.brightness_set_sync

@bad3 depends on after@
identifier i;
expression e;
@@

struct led_classdev i = {
    .brightness_set_sync
    = e,
};

@bad4 depends on after@
identifier i;
expression e;
@@

struct led_classdev i[...] = {...,{
    .brightness_set_sync
    = e,
},...};

@depends on !bad1 && !bad2 && !bad3 && !bad4@
struct led_classdev *e;
@@

- e->brightness_set_sync

@depends on !bad1 && !bad2 && !bad3 && !bad4@
struct led_classdev e;
@@

- e.brightness_set_sync

@depends on !bad1 && !bad2 && !bad3 && !bad4@
identifier i;
expression e;
@@

struct led_classdev i = {
- .brightness_set_sync
```

```

    = e,
};

@depends on !bad1 && !bad2 && !bad3 && !bad4@
identifier i;
expression e;
@@

struct led_classdev i[...] = {...,{
- .brightness_set_sync
    = e,
},...};

```

There is one result, 437a424 at 41%. This shows by many examples that the change should be:

```

@@
struct led_classdev *e;
@@

e->
- brightness_set_sync
- brightness_set_blocking

@@
struct led_classdev e;
@@

e.
- brightness_set_sync
- brightness_set_blocking

@@
identifier i;
expression e;
@@

struct led_classdev i = {
- .brightness_set_sync
+ .brightness_set_blocking
    = e,
};

@@
identifier i;
expression e;
@@

struct led_classdev i[...] = {...,{
- .brightness_set_sync
+ .brightness_set_blocking
    = e,
},...};

```

The actual driver, however, changed more substantially, to change the value stored in this field as well. This issue is not illustrated by the obtained commit.

=== failure for the approach