

# Porting of drivers/phy/phy-tusb1210.c

May 25, 2017

## 1 Introduction

The driver `drivers/phy/phy-tusb1210.c` was introduced in 2013 in the commit `1c14905`. It is a TI TUSB1210 ULPI PHY driver. This driver was accompanied by changes in Kconfig and a make file. Gcc produces 2 errors that reduce to 1 error.

## 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,
...)
```

The first commit is `5e324eb` at 50%. The commit message talks about the new argument specifies the direction and initial value for output. The commit contains two examples. These both have the same value for the new argument, but it is not clear whether this value is always appropriate.

The next commit is `8a68771` at 50% that again has the same argument value.

The next commit is `35eed7a` at 50% that shows a different argument value. This one mentions “output” and the new argument value is `GPIO_IN`. We have two calls and they both call `gpiod_direction_output` on success. Thus it seems that we want the output variant.

The next commit is `a33c380` at 50% that does mention output, but the constant used is `GPIO_OUT_LOW`. Perhaps there are other output variants, so we look further.

The next commit is `06d3f2e` at 33%. This one also uses `GPIO_OUT_LOW`.

The seventh commit is `0a8ba6e` at 14%. This is the first that uses `GPIO_OUT_HIGH`. This one shows a `gpiod_direction_output` call with second argument 1 in the `GPIO_OUT_HIGH` case, and also contains an occurrence of `GPIO_OUT_LOW` with `gpiod_direction_output` having second argument 0. In our calls the second argument is 0, so we go with the `GPIO_OUT_LOW` value. We also observe that this commit removes the `gpiod_direction_output` calls and associated error handling code. The comments in the commit log messages suggest that we should do this too. We thus obtain:

```

@@
expression e,e1,e2,err;
statement S;
@@

e = devm_gpiod_get(e1,e2
+                ,GPIOD_OUT_LOW
                )
...
if (!IS_ERR(e)) {
- err = gpiod_direction_output(e, 0);
- if (err) S
... }

```

The change in the code has the same effect, although the developer somewhat reorganized it at the same time.

=== success 7/17, 0a8ba6e