

# Re: [UPDATED v3][PATCH 1/7] regulator: consumer interface

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-03/msg03574.html>

---

- *From:* Liam Girdwood <[lg@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:lg@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Sun, 09 Mar 2008 11:10:53 +0000
- 

On Fri, 2008-03-07 at 19:43 -0800, David Brownell wrote:

On Thursday 06 March 2008, Liam Girdwood wrote:

```
+static inline int uA_to_A(int uA) { return uA / 1000000; }
```

So: 999999 uA == 0A ... should DIV\_ROUND\_UP() or another rounding function be involved in some of these conversions?

Or maybe the dividing conversions should not be provided, and code should just be doing math in units that don't encourage such problems to appear. I don't think one rounding policy can fit all (including truncation, as above).

ok, this sounds like the best approach.

```
+struct regulator *__must_check regulator_get(struct device *dev,  
+ const char *id);
```

The semantics of "id" and "dev" are unspecified in this patch, so this isn't a good definition of the consumer interface!

'id' is really the regulator name and will be renamed in the next patch.

Plus, that works more like a "lookup" than a "get" ... the usual convention is that "get" and "put" update refcounts. But I think I see an assumption here that a regulator may have only one user...

Re: [UPDATED v3][PATCH 1/7] regulator: consumer interface

A regulator only has one user as it's used to store some device specific power data. However, a regulator\_dev has many users. I'll add a refcount on get/put.

```
+int regulator_set_voltage(struct regulator *regulator, int uV);
```

You described a mode where consumers could set ranges that might overlap (e.g. 1.6 to 1.9V, 1.8 to 2.0) and the result would be some compatible result. But I don't see how that could be achieved, since that's the only call to provide a consumer's constraints.

In this example the power domain constraint (not consumer constraint) would be 1.6 -> 2.0 V. i.e. this range is safe for all consumers on this domain as all can operate at 2.0V and some can operate as low as 1.6V.

The actual regulator output will be determined by consumer voltage requests. e.g. On power domain A, consumer x sets 1.8V and consumer y sets 1.9V, hence regulator output will be 1.9V. (as y needs 1.9 to operate, but x can operate at 1.8 - 2.0)

Presumably one configures a voltage then enables it? How does one turn a voltage supply on or off? I'm guessing that zero volts doesn't equate to "off"...

Some regulators cant go down as far as 0V ;)

We have regulator\_enable() and regulator\_disable() to turn on and off regulator output.

e.g. set voltage -> enable -> do some stuff -> disable.

Something that's lacking here is simple examples. Like: how do I get the power supply associated with an MMC/SD card socket, turn it on (to, say, 3V3), set it to supply a different voltage (maybe 1V8), then turn it off?

## Re: [UPDATED v3][PATCH 1/7] regulator: consumer interface

I have some examples in my git tree :-

backlight

<http://opensource.wolfsonmicro.com/cgi-bin/gitweb.cgi?p=linux-2.6-audioplus.git;a=blob;f=drivers/video/backlight/>

LED's

<http://opensource.wolfsonmicro.com/cgi-bin/gitweb.cgi?p=linux-2.6-audioplus.git;a=blob;f=drivers/leds/leds-wm8350/>

CPUfreq

<http://opensource.wolfsonmicro.com/cgi-bin/gitweb.cgi?p=linux-2.6-audioplus.git;a=blob;f=arch/arm/mach-mx3/cpufreq/>

Audio

<http://opensource.wolfsonmicro.com/cgi-bin/gitweb.cgi?p=linux-2.6-audioplus.git;a=blob;f=sound/soc/imx/imx32adsp/>

In most cases we are passing the power supply name to the consumer driver as platform data. e.g.

```
struct wm8350_led_platform_data wm8350_led_data = {
    .name = "wm8350:white",
    .default_trigger = "heartbeat",
    .isink = WM8350_ISINK_A,
    .dcdc = WM8350_DCDC_5,
    .voltage_ramp = WM8350_DC5_RMP_20V,
    .retries = 5,
    .half_value = 9863,
    .full_value = 27898,
};
```

How would I cope with that voltage supply being shared by two sockets, with cards that may support different voltage ranges and have different current requirements? (Configurations of interest include two cards that can coexist at 1V8, and two that can't ... one might not support 1V8, or it might demand too much power. Also, zero and one cards present.)

In this case the MMC/SD power domain would be 1V8 to 3V3 and it would be upto the MMC/SD driver to ensure it didn't over voltage a 1V8 card with 3V3 in this case. It would also be possible that the system designer would assign each slot a separate regulator to provide max flexibility in this case.

Re: [UPDATED v3][PATCH 1/7] regulator: consumer interface

Liam

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>