Kernel driver for lp5523

- National Semiconductor LP5523 led driver chip
- Datasheet: http://www.national.com/pf/LP/LP5523.html

Authors: Mathias Nyman, Yuri Zaporozhets, Samu Onkalo Contact: Samu Onkalo (samu.p.onkalo-at-nokia.com)

Description

LP5523 can drive up to 9 channels. Leds can be controlled directly via the led class control interface. The name of each channel is configurable in the platform data - name and label. There are three options to make the channel name.

a. Define the 'name' in the platform data

To make specific channel name, then use 'name' platform data.

- /sys/class/leds/R1 (name: 'R1')
- /sys/class/leds/B1 (name: 'B1')

b. Use the 'label' with no 'name' field

For one device name with channel number, then use 'label'. - /sys/class/leds/RGB:channelN (label: 'RGB', N: 0 ~ 8)

c. Default

If both fields are NULL, 'lp5523' is used by default. - /sys/class/leds/lp5523:channelN (N: 0 ~ 8)

LP5523 has the internal program memory for running various LED patterns. There are two ways to run LED patterns.

1. Legacy interface - enginex_mode, enginex_load and enginex_leds

   Control interface for the engines:

   x is 1 .. 3

   enginex_mode:
   disabled, load, run
   enginex_load:
   microcode load
   enginex_leds:
   led mux control

```
cd /sys/class/leds/lp5523:channel2/device
echo "load" > engine3_mode
echo "9d80400004ff05ff437f0000" > engine3_load
echo "111111111" > engine3_leds
echo "run" > engine3_mode
```

   To stop the engine:

```
echo "disabled" > engine3_mode
```

2. Firmware interface - LP55xx common interface

For the details, please refer to 'firmware' section in leds-lp55xx.txt

LP5523 has three master faders. If a channel is mapped to one of the master faders, its output is dimmed based on the value of the master fader.

For example:

```
echo "123000123" > master_fader_leds
```

creates the following channel-fader mappings:

```
channel 0,6 to master_fader1
channel 1,7 to master_fader2
channel 2,8 to master_fader3
```

Then, to have 25% of the original output on channel 0,6:

```
echo 64 > master_fader1
```

To have 0% of the original output (i.e. no output) channel 1,7:

```
echo 0 > master_fader2
```

To have 100% of the original output (i.e. no dimming) on channel 2,8:

```
echo 255 > master_fader3
```

To clear all master fader controls:

```
echo "000000000" > master_fader_leds
```

Selftest uses always the current from the platform data.

Each channel contains led current settings. - /sys/class/leds/lp5523:channel2/led_current - RW - /sys/class/leds/lp5523:channel2/max_current - RO

Format: 10x mA i.e 10 means 1.0 mA

Example platform data:

```
static struct lp55xx_led_config lp5523_led_config[] = {
        {
                .name           = "D1",
                .chan_nr        = 0,
                .led_current    = 50,
                .max_current    = 130,
        },
...
        {
                .chan_nr        = 8,
                .led_current    = 50,
                .max_current    = 130,
        }
};

static int lp5523_setup(void)
{
        /* Setup HW resources */
}

static void lp5523_release(void)
{
        /* Release HW resources */
}

static void lp5523_enable(bool state)
{
        /* Control chip enable signal */
}

static struct lp55xx_platform_data lp5523_platform_data = {
        .led_config     = lp5523_led_config,
        .num_channels   = ARRAY_SIZE(lp5523_led_config),
        .clock_mode     = LP55XX_CLOCK_EXT,
        .setup_resources   = lp5523_setup,
        .release_resources = lp5523_release,
        .enable            = lp5523_enable,
};
```

Note

chan_nr can have values between 0 and 8.