

# Kernel driver for lp5521

- National Semiconductor LP5521 led driver chip
- Datasheet: <http://www.national.com/pdf/LP/LP5521.html>

Authors: Mathias Nyman, Yuri Zaporozhets, Samu Onkalo

Contact: Samu Onkalo (samu.p.onkalo-at-nokia.com)

## Description

LP5521 can drive up to 3 channels. Leds can be controlled directly via the led class control interface. Channels have generic names: lp5521:channelx, where x is 0 .. 2

All three channels can be also controlled using the engine micro programs. More details of the instructions can be found from the public data sheet.

LP5521 has the internal program memory for running various LED patterns. There are two ways to run LED patterns.

1. Legacy interface - `enginex_mode` and `enginex_load` Control interface for the engines:

x is 1 .. 3

`enginex_mode`:

disabled, load, run

`enginex_load`:

store program (visible only in engine load mode)

Example (start to blink the channel 2 led):

```
cd /sys/class/leds/lp5521:channel2/device
echo "load" > engine3_mode
echo "037f4d0003fff6000" > engine3_load
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

sysfs contains a selftest entry.

The test communicates with the chip and checks that the clock mode is automatically set to the requested one.

Each channel has its own led current settings.

- `/sys/class/leds/lp5521:channel0/led_current` - RW
- `/sys/class/leds/lp5521:channel0/max_current` - RO

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

example platform data:

```
static struct lp55xx_led_config lp5521_led_config[] = {
    {
        .name = "red",
        .chan_nr = 0,
        .led_current = 50,
        .max_current = 130,
    }, {
        .name = "green",
        .chan_nr = 1,
        .led_current = 0,
        .max_current = 130,
    }, {
        .name = "blue",
        .chan_nr = 2,
        .led_current = 0,
        .max_current = 130,
    }
};

static int lp5521_setup(void)
{
    /* setup HW resources */
}
```

```

static void lp5521_release(void)
{
    /* Release HW resources */
}

static void lp5521_enable(bool state)
{
    /* Control of chip enable signal */
}

static struct lp55xx_platform_data lp5521_platform_data = {
    .led_config      = lp5521_led_config,
    .num_channels    = ARRAY_SIZE(lp5521_led_config),
    .clock_mode      = LP55XX_CLOCK_EXT,
    .setup_resources = lp5521_setup,
    .release_resources = lp5521_release,
    .enable          = lp5521_enable,
};

```

**Note:**

chan\_nr can have values between 0 and 2. The name of each channel can be configurable. If the name field is not defined, the default name will be set to 'xxxx:channelN' (XXXX : pdata->label or i2c client name, N : channel number)

If the current is set to 0 in the platform data, that channel is disabled and it is not visible in the sysfs.