

# [PATCH 2/11] LED: Add LED Class

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-01/msg11419.html>

---

- *From:* Richard Purdie <[rpurdie@xxxxxxxx](mailto:rpurdie@xxxxxxxx)>
  - *Date:* Tue, 31 Jan 2006 13:41:28 +0000
- 

Add the foundations of a new LEDs subsystem. This patch adds a class which presents LED devices within sysfs and allows their brightness to be controlled.

Signed-off-by: Richard Purdie <[rpurdie@xxxxxxxx](mailto:rpurdie@xxxxxxxx)>

Index: linux-2.6.15/drivers/leds/Kconfig

```
=====  
--- /dev/null 1970-01-01 00:00:00.000000000 +0000  
+++ linux-2.6.15/drivers/leds/Kconfig 2006-01-29 16:02:38.000000000 +0000  
@@ -0,0 +1,18 @@  
+  
+menu "LED devices"  
+  
+config NEW_LEDS  
+ bool "LED Support"  
+ help  
+ Say Y to enable Linux LED support. This is not related to standard  
+ keyboard LEDs which are controlled via the input system.  
+  
+config LEDS_CLASS  
+ tristate "LED Class Support"  
+ depends NEW_LEDS  
+ help  
+ This option enables the led sysfs class in /sys/class/leds. You'll  
+ need this to do anything useful with LEDs. If unsure, say N.  
+  
+endmenu
```

Index: linux-2.6.15/drivers/leds/Makefile

```
=====  
--- /dev/null 1970-01-01 00:00:00.000000000 +0000  
+++ linux-2.6.15/drivers/leds/Makefile 2006-01-29 16:02:38.000000000 +0000  
@@ -0,0 +1,4 @@  
+  
+# LED Core  
+obj-$(CONFIG_NEW_LEDS) += led-core.o  
+obj-$(CONFIG_LEDS_CLASS) += led-class.o  
Index: linux-2.6.15/include/linux/leds.h
```

[PATCH 2/11] LED: Add LED Class

```
----- /dev/null 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.15/include/linux/leds.h 2006-01-29 16:03:21.000000000 +0000
@@ -0,0 +1,48 @@
+/*
+ * Driver model for leds and led triggers
+ *
+ * Copyright (C) 2005 John Lenz <lenz@xxxxxxxxxxxx>
+ * Copyright (C) 2005 Richard Purdie <rpurdie@xxxxxxxxxxxx>
+ *
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License version 2 as
+ * published by the Free Software Foundation.
+ *
+ */
+
+struct device;
+struct class_device;
+/*
+ * LED Core
+ */
+
+enum led_brightness {
+ LED_OFF = 0,
+ LED_HALF = 127,
+ LED_FULL = 255,
+};
+
+struct led_device {
+ const char *name;
+ int brightness;
+ int flags;
+#define LED_SUSPENDED (1 << 0)
+
+ /* A function to set the brightness of the led */
+ void (*brightness_set)(struct led_device *led_dev, enum led_brightness brightness);
+
+ struct class_device *class_dev;
+ /* LED Device linked list */
+ struct list_head node;
+
+ /* Trigger data */
+ char *default_trigger;
+
+ /* This protects the data in this structure */
+ rwlock_t lock;
+};
+
+extern int led_device_register(struct device *dev, struct led_device *led_dev);
+extern void led_device_unregister(struct led_device *led_dev);
+extern void led_device_suspend(struct led_device *led_dev);
```

[PATCH 2/11] LED: Add LED Class

```
+extern void led_device_resume(struct led_device *led_dev);
```

```
Index: linux-2.6.15/arch/arm/Kconfig
```

```
----- linux-2.6.15.orig/arch/arm/Kconfig 2006-01-29 14:37:31.000000000 +0000  
+++ linux-2.6.15/arch/arm/Kconfig 2006-01-29 16:02:15.000000000 +0000  
@@ -774,6 +774,8 @@
```

```
source "drivers/mfd/Kconfig"
```

```
+source "drivers/leds/Kconfig"
```

```
+
```

```
source "drivers/media/Kconfig"
```

```
source "drivers/video/Kconfig"
```

```
Index: linux-2.6.15/drivers/Makefile
```

```
----- linux-2.6.15.orig/drivers/Makefile 2006-01-29 14:42:52.000000000 +0000  
+++ linux-2.6.15/drivers/Makefile 2006-01-29 14:43:11.000000000 +0000  
@@ -68,6 +68,7 @@
```

```
obj-$(CONFIG_EISA) += eisa/
```

```
obj-$(CONFIG_CPU_FREQ) += cpufreq/
```

```
obj-$(CONFIG_MMC) += mmc/
```

```
+obj-$(CONFIG_NEW_LEDS) += leds/
```

```
obj-$(CONFIG_INFINIBAND) += infiniband/
```

```
obj-$(CONFIG_SGI_IOC4) += sn/
```

```
obj-y += firmware/
```

```
Index: linux-2.6.15/drivers/Kconfig
```

```
----- linux-2.6.15.orig/drivers/Kconfig 2006-01-29 14:42:51.000000000 +0000  
+++ linux-2.6.15/drivers/Kconfig 2006-01-29 14:43:11.000000000 +0000  
@@ -64,6 +64,8 @@
```

```
source "drivers/mmc/Kconfig"
```

```
+source "drivers/leds/Kconfig"
```

```
+
```

```
source "drivers/infiniband/Kconfig"
```

```
source "drivers/sn/Kconfig"
```

```
Index: linux-2.6.15/drivers/leds/leds.h
```

```
----- /dev/null 1970-01-01 00:00:00.000000000 +0000  
+++ linux-2.6.15/drivers/leds/leds.h 2006-01-29 16:02:38.000000000 +0000  
@@ -0,0 +1,26 @@
```

```
+/*
```

```
+ * LED Core
```

```
+ *
```

```
+ * Copyright 2005 Openedhand Ltd.
```

```
+ *
```

```
+ * Author: Richard Purdie <rpurdie@xxxxxxxxxxxxxxxx>
```

```
+ *
```

## [PATCH 2/11] LED: Add LED Class

```
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License version 2 as
+ * published by the Free Software Foundation.
+ *
+ */
+
+/* led_dev->lock must be held as write */
+static inline void led_set_brightness(struct led_device *led_dev, enum led_brightness value)
+{
+ if (value > LED_FULL)
+ value = LED_FULL;
+ led_dev->brightness = value;
+ if (!(led_dev->flags & LED_SUSPENDED))
+ led_dev->brightness_set(led_dev, value);
+}
+
+extern rwlock_t leds_list_lock;
+extern struct list_head leds_list;
+
Index: linux-2.6.15/drivers/leds/led-class.c
=====
--- /dev/null 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.15/drivers/leds/led-class.c 2006-01-29 16:02:38.000000000 +0000
@@ -0,0 +1,150 @@
+/*
+ * LED Class Core
+ *
+ * Copyright (C) 2005 John Lenz <lenz@xxxxxxxxxxxx>
+ * Copyright (C) 2005-2006 Richard Purdie <rpurdie@xxxxxxxxxxxxxxxx>
+ *
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License version 2 as
+ * published by the Free Software Foundation.
+ */
+
+#include <linux/config.h>
+#include <linux/module.h>
+#include <linux/kernel.h>
+#include <linux/init.h>
+#include <linux/list.h>
+#include <linux/spinlock.h>
+#include <linux/device.h>
+#include <linux/sysdev.h>
+#include <linux/timer.h>
+#include <linux/err.h>
+#include <linux/leds.h>
+#include "leds.h"
+
+static struct class *leds_class;
+
+static ssize_t led_brightness_show(struct class_device *dev, char *buf)
```

## [PATCH 2/11] LED: Add LED Class

```
+{
+ struct led_device *led_dev = dev->class_data;
+ ssize_t ret = 0;
+
+ /* no lock needed for this */
+ sprintf(buf, "%u\n", led_dev->brightness);
+ ret = strlen(buf) + 1;
+
+ return ret;
+}
+
+static ssize_t led_brightness_store(struct class_device *dev, const char *buf, size_t size)
+{
+ struct led_device *led_dev = dev->class_data;
+ ssize_t ret = -EINVAL;
+ char *after;
+
+ unsigned long state = simple_strtoul(buf, &after, 10);
+ if (after - buf > 0) {
+ ret = after - buf;
+ write_lock(&led_dev->lock);
+ led_set_brightness(led_dev, state);
+ write_unlock(&led_dev->lock);
+ }
+
+ return ret;
+}
+
+static CLASS_DEVICE_ATTR(brightness, 0644, led_brightness_show, led_brightness_store);
+
+
+/**
+ * led_device_suspend - suspend an led_device.
+ * @led_dev: the led_device to suspend.
+ */
+void led_device_suspend(struct led_device *led_dev)
+{
+ write_lock(&led_dev->lock);
+ led_dev->flags |= LED_SUSPENDED;
+ led_dev->brightness_set(led_dev, 0);
+ write_unlock(&led_dev->lock);
+}
+
+/**
+ * led_device_resume - resume an led_device.
+ * @led_dev: the led_device to resume.
+ */
+void led_device_resume(struct led_device *led_dev)
+{
+ write_lock(&led_dev->lock);
+ led_dev->flags &= ~LED_SUSPENDED;
```

## [PATCH 2/11] LED: Add LED Class

```
+ led_dev->brightness_set(led_dev, led_dev->brightness);
+ write_unlock(&led_dev->lock);
+ }
+
+ /**
+  * led_device_register - register a new object of led_device class.
+  * @dev: The device to register.
+  * @led_dev: the led_device structure for this device.
+  */
+int led_device_register(struct device *dev, struct led_device *led_dev)
+{
+ led_dev->class_dev = class_device_create(leds_class, NULL, 0, dev, "%s", led_dev->name);
+ if (unlikely(IS_ERR(led_dev->class_dev)))
+ return PTR_ERR(led_dev->class_dev);
+
+ rwlock_init(&led_dev->lock);
+ led_dev->class_dev->class_data = led_dev;
+
+ /* register the attributes */
+ class_device_create_file(led_dev->class_dev, &class_device_attr_brightness);
+
+ /* add to the list of leds */
+ write_lock(&leds_list_lock);
+ list_add_tail(&led_dev->node, &leds_list);
+ write_unlock(&leds_list_lock);
+
+ printk(KERN_INFO "Registered led device: %s\n", led_dev->class_dev->class_id);
+
+ return 0;
+}
+
+ /**
+  * led_device_unregister - unregisters a object of led_properties class.
+  * @led_dev: the led device to unregister
+  *
+  * Unregisters a previously registered via led_device_register object.
+  */
+void led_device_unregister(struct led_device *led_dev)
+{
+ class_device_remove_file(led_dev->class_dev, &class_device_attr_brightness);
+
+ class_device_unregister(led_dev->class_dev);
+
+ write_lock(&leds_list_lock);
+ list_del(&led_dev->node);
+ write_unlock(&leds_list_lock);
+}
+
+EXPORT_SYMBOL_GPL(led_device_suspend);
+EXPORT_SYMBOL_GPL(led_device_resume);
+EXPORT_SYMBOL_GPL(led_device_register);
```

[PATCH 2/11] LED: Add LED Class

```
+EXPORT_SYMBOL_GPL(led_device_unregister);
+
+static int __init leds_init(void)
+{
+ leds_class = class_create(THIS_MODULE, "leds");
+ if (IS_ERR(leds_class))
+ return PTR_ERR(leds_class);
+ return 0;
+}
+
+static void __exit leds_exit(void)
+{
+ class_destroy(leds_class);
+}
+
+subsys_initcall(leds_init);
+module_exit(leds_exit);
+
+MODULE_AUTHOR("John Lenz, Richard Purdie");
+MODULE_LICENSE("GPL");
+MODULE_DESCRIPTION("LED Class Interface");
Index: linux-2.6.15/drivers/leds/led-core.c
=====
--- /dev/null 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.15/drivers/leds/led-core.c 2006-01-29 14:43:11.000000000 +0000
@@ -0,0 +1,24 @@
+/*
+ * LED Class Core
+ *
+ * Copyright 2005-2006 Openedhand Ltd.
+ *
+ * Author: Richard Purdie <rpurdie@xxxxxxxxxxxxxxxx>
+ *
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License version 2 as
+ * published by the Free Software Foundation.
+ *
+ */
+
+#include <linux/kernel.h>
+#include <linux/list.h>
+#include <linux/module.h>
+#include <linux/spinlock.h>
+#include <linux/leds.h>
+#include "leds.h"
+
+rwlock_t leds_list_lock = RW_LOCK_UNLOCKED;
+LIST_HEAD(leds_list);
+
+EXPORT_SYMBOL_GPL(leds_list);
```

—  
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/>

---

- *Follow-Ups:*
  - ◆ **Re: [PATCH 2/11] LED: Add LED Class**
    - ◇ *From:* Greg KH
  - ◆ **Re: [PATCH 2/11] LED: Add LED Class**
    - ◇ *From:* Jan-Benedict Glaw
- Prev by Date: **[PATCH 6/11] LED: Add LED device support for the zaurus corgi and spitz models**
- Next by Date: **Re: iommu alloc failure and panic**
- Previous by thread: **[PATCH 6/11] LED: Add LED device support for the zaurus corgi and spitz models**
- Next by thread: **Re: [PATCH 2/11] LED: Add LED Class**
- Index(es):
  - ◆ **Date**
  - ◆ **Thread**