

# [RFC PATCH 1/8] LED: Add LED Class

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-12/msg01201.html>

---

- *From:* Richard Purdie <rpurdie@xxxxxxxx>
  - *Date:* Mon, 05 Dec 2005 13:09:26 +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>  
Acked-by: Pavel Machek <pavel@xxxxxxx>

Index: linux-2.6.15-rc2/drivers/leds/Kconfig

```
=====  
--- /dev/null 1970-01-01 00:00:00.000000000 +0000  
+++ linux-2.6.15-rc2/drivers/leds/Kconfig 2005-12-05 11:29:19.000000000 +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-rc2/drivers/leds/Makefile

```
=====  
--- /dev/null 1970-01-01 00:00:00.000000000 +0000  
+++ linux-2.6.15-rc2/drivers/leds/Makefile 2005-12-05 11:29:19.000000000 +0000  
@@ -0,0 +1,4 @@  
+  
+# LED Core  
+obj-$(CONFIG_NEW_LEDS) += led_core.o  
+obj-$(CONFIG_LEDS_CLASS) += led_class.o
```

## [RFC PATCH 1/8] LED: Add LED Class

Index: linux-2.6.15-rc2/include/linux/leds.h

```
=====
--- /dev/null 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.15-rc2/include/linux/leds.h 2005-12-05 11:29:19.000000000 +0000
@@ -0,0 +1,43 @@
+/*
+ * Driver model for leds and led triggers
+ *
+ * Copyright (C) 2005 John Lenz <lenz@xxxxxxxxxxxx>
+ * Copyright (C) 2005 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.
+ *
+ */
+
+struct device;
+struct class_device;
+/*
+ * LED Core
+ */
+struct led_device {
+ char *name;
+ int brightness;
+ int flags;
+#define LED_SUSPENDED (1 << 0)
+
+ /* A function to set the brightness of the led.
+ * Values are between 0-100 */
+ void (*brightness_set)(struct led_device *led_dev, int value);
+
+ 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;
+};
+
+void leds_set_brightness(struct led_device *led_dev, int value);
+int leds_device_register(struct device *dev, struct led_device *led_dev);
+void leds_device_unregister(struct led_device *led_dev);
+void leds_device_suspend(struct led_device *led_dev);
+void leds_device_resume(struct led_device *led_dev);
Index: linux-2.6.15-rc2/arch/arm/Kconfig
```

```
=====
--- linux-2.6.15-rc2.orig/arch/arm/Kconfig 2005-11-20 03:25:03.000000000 +0000
```

[RFC PATCH 1/8] LED: Add LED Class

```
+++ linux-2.6.15-rc2/arch/arm/Kconfig 2005-12-05 11:28:38.000000000 +0000
@@ -738,6 +738,8 @@
```

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

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

```
+
```

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

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

```
Index: linux-2.6.15-rc2/drivers/Makefile
```

```
----- linux-2.6.15-rc2.orig/drivers/Makefile 2005-11-20 03:25:03.000000000 +0000
```

```
+++ linux-2.6.15-rc2/drivers/Makefile 2005-12-05 10:44:30.000000000 +0000
```

```
@@ -64,6 +64,7 @@
```

```
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-rc2/drivers/Kconfig
```

```
----- linux-2.6.15-rc2.orig/drivers/Kconfig 2005-11-20 03:25:03.000000000 +0000
```

```
+++ linux-2.6.15-rc2/drivers/Kconfig 2005-12-05 10:44:30.000000000 +0000
```

```
@@ -62,6 +62,8 @@
```

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

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

```
+
```

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

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

```
Index: linux-2.6.15-rc2/drivers/leds/leds.h
```

```
----- /dev/null 1970-01-01 00:00:00.000000000 +0000
```

```
+++ linux-2.6.15-rc2/drivers/leds/leds.h 2005-12-05 11:29:19.000000000 +0000
```

```
@@ -0,0 +1,15 @@
```

```
+/*
```

```
+ * LED Core
```

```
+ *
```

```
+ * Copyright 2005 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.
```

```
+ *
```

## [RFC PATCH 1/8] LED: Add LED Class

```
+ */
+
+extern rwlock_t leds_list_lock;
+extern struct list_head leds_list;
Index: linux-2.6.15-rc2/drivers/leds/led_class.c
=====
--- /dev/null 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.15-rc2/drivers/leds/led_class.c 2005-12-05 11:30:09.000000000 +0000
@@ -0,0 +1,171 @@
+/*
+ * LED Class Core
+ *
+ * Copyright (C) 2005 John Lenz <lenz@xxxxxxxxxxxx>
+ * Copyright (C) 2005 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/leds.h>
+#include "leds.h"
+
+static void leds_class_release(struct class_device *dev)
+{
+ kfree(dev);
+}
+
+static struct class leds_class = {
+ .name = "leds",
+ .release = leds_class_release,
+};
+
+static ssize_t leds_show_brightness(struct class_device *dev, char *buf)
+{
+ 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;
+}
+

```

## [RFC PATCH 1/8] LED: Add LED Class

```
+ return ret;
+}
+
+static ssize_t leds_store_brightness(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);
+ leds_set_brightness(led_dev, state);
+ write_unlock(&led_dev->lock);
+ }
+
+ return ret;
+}
+
+static CLASS_DEVICE_ATTR(brightness, 0644, leds_show_brightness, leds_store_brightness);
+
+/* led_dev->lock must be held as write */
+void leds_set_brightness(struct led_device *led_dev, int value)
+{
+ if (value > 100)
+ value = 100;
+ led_dev->brightness = value;
+ if (!(led_dev->flags & LED_SUSPENDED))
+ led_dev->brightness_set(led_dev, value);
+}
+
+void leds_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);
+}
+
+void leds_device_resume(struct led_device *led_dev)
+{
+ write_lock(&led_dev->lock);
+ led_dev->flags &= ~LED_SUSPENDED;
+ led_dev->brightness_set(led_dev, led_dev->brightness);
+ write_unlock(&led_dev->lock);
+}
+
+/**
+ * leds_device_register - register a new object of led_device class.
```

## [RFC PATCH 1/8] LED: Add LED Class

```
+ * @dev: The device to register.
+ * @led_dev: the led_device structure for this device.
+ */
+int leds_device_register(struct device *dev, struct led_device *led_dev)
+{
+ int rc;
+
+ led_dev->class_dev = kzalloc (sizeof (struct class_device), GFP_KERNEL);
+ if (unlikely (!led_dev->class_dev))
+ return -ENOMEM;
+
+ rwlock_init(&led_dev->lock);
+
+ led_dev->class_dev->class = &leds_class;
+ led_dev->class_dev->dev = dev;
+ led_dev->class_dev->class_data = led_dev;
+
+ /* assign this led its name */
+ strncpy(led_dev->class_dev->class_id, led_dev->name, sizeof(led_dev->class_dev->class_id));
+
+ rc = class_device_register (led_dev->class_dev);
+ if (unlikely (rc)) {
+ kfree (led_dev->class_dev);
+ return rc;
+ }
+
+ /* 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;
+}
+
+/**
+ * leds_device_unregister – unregisters a object of led_properties class.
+ * @led_dev: the led device to unregiester
+ *
+ * Unregisters a previously registered via leds_device_register object.
+ */
+void leds_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);
+
+}
```

[RFC PATCH 1/8] LED: Add LED Class

```
+ write_lock(&leds_list_lock);
+ list_del(&led_dev->node);
+ write_unlock(&leds_list_lock);
+}
+
+EXPORT_SYMBOL(leds_device_suspend);
+EXPORT_SYMBOL(leds_device_resume);
+EXPORT_SYMBOL(leds_device_register);
+EXPORT_SYMBOL(leds_device_unregister);
+
+static int __init leds_init(void)
+{
+ return class_register(&leds_class);
+}
+
+static void __exit leds_exit(void)
+{
+ class_unregister(&leds_class);
+}
+
+subsys_initcall(leds_init);
+module_exit(leds_exit);
+
+MODULE_AUTHOR("John Lenz");
+MODULE_LICENSE("GPL");
+MODULE_DESCRIPTION("LED Class Interface");
+
Index: linux-2.6.15-rc2/drivers/leds/led_core.c
=====
--- /dev/null 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.15-rc2/drivers/leds/led_core.c 2005-12-05 10:44:30.000000000 +0000
@@ -0,0 +1,24 @@
+/*
+ * LED Class Core
+ *
+ * Copyright 2005 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"
```

[RFC PATCH 1/8] LED: Add LED Class

```
+  
+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@xxxxxxxxxxxxxxxxxxx  
More majordomo info at <http://vger.kernel.org/majordomo-info.html>  
Please read the FAQ at <http://www.tux.org/lkml/>

---

• *Follow-Ups:*

- ◆ **Re: [RFC PATCH 1/8] LED: Add LED Class**  
    ◇ From: Greg KH
- ◆ **Re: [RFC PATCH 1/8] LED: Add LED Class**  
    ◇ From: David Vrabel

- Prev by Date: **[RFC PATCH 7/8] LED: Add IDE disk activity LED trigger**
- Next by Date: **Kernel BUG at page\_alloc.c:117!**
- Previous by thread: **[RFC PATCH 7/8] LED: Add IDE disk activity LED trigger**
- Next by thread: **Re: [RFC PATCH 1/8] LED: Add LED Class**
- Index(es):
  - ◆ **Date**
  - ◆ **Thread**