

[PATCH 6/11] LED: Add LED device support for the zaurus corgi and spitz models

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-01/msg11412.html>

- *From:* Richard Purdie <rpurdie@xxxxxxxx>
 - *Date:* Tue, 31 Jan 2006 13:41:43 +0000
-

Adds LED drivers for LEDs found on the Sharp Zaurus c7x0 (corgi, shepherd, husky) and cxx00 (akita, spitz, borzoi) models.

Signed-off-by: Richard Purdie <rpurdie@xxxxxxxx>

Index: linux-2.6.15/arch/arm/mach-pxa/corgi.c

```
----- linux-2.6.15.orig/arch/arm/mach-pxa/corgi.c 2006-01-29 16:02:30.000000000 +0000
+++ linux-2.6.15/arch/arm/mach-pxa/corgi.c 2006-01-29 16:11:47.000000000 +0000
@@ -165,6 +165,15 @@
```

```
/*
+ * Corgi LEDs
+ */
+static struct platform_device corgiled_device = {
+ .name = "corgi-led",
+ .id = -1,
+ };
+
+
+/*
* Corgi Touch Screen Device
*/
static struct resource corgits_resources[] = {
@@ -298,6 +307,7 @@
&corgikbd_device,
&corgibl_device,
&corgits_device,
+ &corgiled_device,
};
```

```
static void __init corgi_init(void)
```

Index: linux-2.6.15/arch/arm/mach-pxa/spitz.c

```
----- linux-2.6.15.orig/arch/arm/mach-pxa/spitz.c 2006-01-29 16:02:30.000000000 +0000
+++ linux-2.6.15/arch/arm/mach-pxa/spitz.c 2006-01-29 16:11:48.000000000 +0000
@@ -243,6 +243,15 @@
```

[PATCH 6/11] LED: Add LED device support for the zaurus corgi and spitz models

```
/*
+ * Spitz LEDs
+ */
+static struct platform_device spitzled_device = {
+ .name = "spitz-led",
+ .id = -1,
+};
+
+
+/*
* Spitz Touch Screen Device
*/
static struct resource spitzts_resources[] = {
@@ -419,6 +428,7 @@
&spitzkbd_device,
&spitzts_device,
&spitzbl_device,
+ &spitzled_device,
};

static void __init common_init(void)
Index: linux-2.6.15/drivers/leds/Kconfig
=====
--- linux-2.6.15.orig/drivers/leds/Kconfig 2006-01-29 16:04:20.000000000 +0000
+++ linux-2.6.15/drivers/leds/Kconfig 2006-01-29 16:11:54.000000000 +0000
@@ -22,6 +22,20 @@
These triggers allow kernel events to drive the LEDs and can
be configured via sysfs. If unsure, say Y.

+config LEDS_CORGI
+ tristate "LED Support for the Sharp SL-C7x0 series"
+ depends LEDS_CLASS && PXA_SHARP_C7xx
+ help
+ This option enables support for the LEDs on Sharp Zaurus
+ SL-C7x0 series (C700, C750, C760, C860).
+
+config LEDS_SPITZ
+ tristate "LED Support for the Sharp SL-Cxx00 series"
+ depends LEDS_CLASS && PXA_SHARP_Cxx00
+ help
+ This option enables support for the LEDs on Sharp Zaurus
+ SL-Cxx00 series (C1000, C3000, C3100).
+
config LEDS_TRIGGER_TIMER
tristate "LED Timer Trigger"
depends LEDS_TRIGGERS
Index: linux-2.6.15/drivers/leds/Makefile
=====
--- linux-2.6.15.orig/drivers/leds/Makefile 2006-01-29 16:04:20.000000000 +0000
```

[PATCH 6/11] LED: Add LED device support for the zaurus corgi and spitz models

```
+++ linux-2.6.15/drivers/leds/Makefile 2006-01-29 16:11:54.000000000 +0000
@@ -4,5 +4,9 @@
obj-$(CONFIG_LEDS_CLASS) += led-class.o
obj-$(CONFIG_LEDS_TRIGGERS) += led-triggers.o

+# LED Platform Drivers
+obj-$(CONFIG_LEDS_CORGI) += leds-corgi.o
+obj-$(CONFIG_LEDS_SPITZ) += leds-spitz.o
+
# LED Triggers
obj-$(CONFIG_LEDS_TRIGGER_TIMER) += ledtrig-timer.o
Index: linux-2.6.15/drivers/leds/leds-corgi.c
=====
--- /dev/null 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.15/drivers/leds/leds-corgi.c 2006-01-29 16:08:42.000000000 +0000
@@ -0,0 +1,121 @@
+/*
+ * LED Triggers 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/config.h>
+#include <linux/kernel.h>
+#include <linux/init.h>
+#include <linux/platform_device.h>
+#include <linux/leds.h>
+#include <asm/mach-types.h>
+#include <asm/arch/corgi.h>
+#include <asm/arch/hardware.h>
+#include <asm/arch/pxa-regs.h>
+#include <asm/hardware/scoop.h>
+
+void corgiled_amber_set(struct led_device *led_dev, enum led_brightness value)
+{
+ if (value)
+ GPCR0 = GPIO_bit(CORGI_GPIO_LED_ORANGE);
+ else
+ GPCR0 = GPIO_bit(CORGI_GPIO_LED_ORANGE);
+}
+
+void corgiled_green_set(struct led_device *led_dev, enum led_brightness value)
+{
+ if (value)
```

[PATCH 6/11] LED: Add LED device support for the zaurus corgi and spitz models

```
+ set_scoop_gpio(&corgiscoop_device.dev, CORGI_SCP_LED_GREEN);
+ else
+ reset_scoop_gpio(&corgiscoop_device.dev, CORGI_SCP_LED_GREEN);
+}
+
+struct led_device corgi_amber_led = {
+ .name = "corgi:amber",
+ .default_trigger = "sharpsh-charge",
+ .brightness_set = corgiled_amber_set,
+};
+
+struct led_device corgi_green_led = {
+ .name = "corgi:green",
+ .default_trigger = "nand-disk",
+ .brightness_set = corgiled_green_set,
+};
+
+#ifdef CONFIG_PM
+static int corgiled_suspend(struct platform_device *dev, pm_message_t state)
+{
+#ifdef CONFIG_LEDS_TRIGGERS
+ if (corgi_amber_led.trigger && strcmp(corgi_amber_led.trigger->name, "sharpsh-charge"))
+#endif
+ led_device_suspend(&corgi_amber_led);
+ led_device_suspend(&corgi_green_led);
+ return 0;
+}
+
+static int corgiled_resume(struct platform_device *dev)
+{
+ led_device_resume(&corgi_amber_led);
+ led_device_resume(&corgi_green_led);
+ return 0;
+}
+#endif
+
+static int corgiled_probe(struct platform_device *pdev)
+{
+ int ret;
+
+ ret = led_device_register(&pdev->dev, &corgi_amber_led);
+ if (ret < 0)
+ return ret;
+
+ ret = led_device_register(&pdev->dev, &corgi_green_led);
+ if (ret < 0)
+ led_device_unregister(&corgi_amber_led);
+
+ return ret;
+}
+
```

[PATCH 6/11] LED: Add LED device support for the zaurus corgi and spitz models

```
+static int corgiled_remove(struct platform_device *pdev)
+{
+ led_device_unregister(&corgi_amber_led);
+ led_device_unregister(&corgi_green_led);
+ return 0;
+}
+
+static struct platform_driver corgiled_driver = {
+ .probe = corgiled_probe,
+ .remove = corgiled_remove,
+#ifdef CONFIG_PM
+ .suspend = corgiled_suspend,
+ .resume = corgiled_resume,
+#endif
+ .driver = {
+ .name = "corgi-led",
+ },
+ };
+
+static int __devinit corgiled_init(void)
+{
+ return platform_driver_register(&corgiled_driver);
+}
+
+static void corgiled_exit(void)
+{
+ platform_driver_unregister(&corgiled_driver);
+}
+
+module_init(corgiled_init);
+module_exit(corgiled_exit);
+
+MODULE_AUTHOR("Richard Purdie <rpurdie@xxxxxxxxxxxxxxxx>");
+MODULE_DESCRIPTION("Corgi LED driver");
+MODULE_LICENSE("GPL");
Index: linux-2.6.15/drivers/leds/leds-spitz.c
=====
--- /dev/null 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6.15/drivers/leds/leds-spitz.c 2006-01-29 16:08:09.000000000 +0000
@@ -0,0 +1,125 @@
+/*
+ * LED Triggers 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.
+ */
```

[PATCH 6/11] LED: Add LED device support for the zaurus corgi and spitz models

```
+ */
+
+#include <linux/config.h>
+#include <linux/kernel.h>
+#include <linux/init.h>
+#include <linux/platform_device.h>
+#include <linux/leds.h>
+#include <asm/hardware/scoop.h>
+#include <asm/mach-types.h>
+#include <asm/arch/hardware.h>
+#include <asm/arch/pxa-regs.h>
+#include <asm/arch/spitz.h>
+
+void spitzled_amber_set(struct led_device *led_dev, enum led_brightness value)
+{
+ if (value)
+ set_scoop_gpio(&spitzscoop_device.dev, SPITZ_SCP_LED_ORANGE);
+ else
+ reset_scoop_gpio(&spitzscoop_device.dev, SPITZ_SCP_LED_ORANGE);
+}
+
+void spitzled_green_set(struct led_device *led_dev, enum led_brightness value)
+{
+ if (value)
+ set_scoop_gpio(&spitzscoop_device.dev, SPITZ_SCP_LED_GREEN);
+ else
+ reset_scoop_gpio(&spitzscoop_device.dev, SPITZ_SCP_LED_GREEN);
+}
+
+static struct led_device spitz_amber_led = {
+ .name = "spitz:amber",
+ .default_trigger = "sharpsl-charge",
+ .brightness_set = spitzled_amber_set,
+};
+
+static struct led_device spitz_green_led = {
+ .name = "spitz:green",
+ .default_trigger = "ide-disk",
+ .brightness_set = spitzled_green_set,
+};
+
+#ifdef CONFIG_PM
+static int spitzled_suspend(struct platform_device *dev, pm_message_t state)
+{
+#ifdef CONFIG_LEDS_TRIGGERS
+ if (spitz_amber_led.trigger && strcmp(spitz_amber_led.trigger->name, "sharpsl-charge"))
+#endif
+ led_device_suspend(&spitz_amber_led);
+ led_device_suspend(&spitz_green_led);
+ return 0;
+}
```

[PATCH 6/11] LED: Add LED device support for the zaurus corgi and spitz models

```
+
+static int spitzled_resume(struct platform_device *dev)
+{
+ led_device_resume(&spitz_amber_led);
+ led_device_resume(&spitz_green_led);
+ return 0;
+}
+#endif
+
+static int spitzled_probe(struct platform_device *pdev)
+{
+ int ret;
+
+
+ if (machine_is_akita())
+ spitz_green_led.default_trigger = "nand-disk";
+
+
+ ret = led_device_register(&pdev->dev, &spitz_amber_led);
+ if (ret < 0)
+ return ret;
+
+
+ ret = led_device_register(&pdev->dev, &spitz_green_led);
+ if (ret < 0)
+ led_device_unregister(&spitz_amber_led);
+
+
+ return ret;
+}
+
+static int spitzled_remove(struct platform_device *pdev)
+{
+ led_device_unregister(&spitz_amber_led);
+ led_device_unregister(&spitz_green_led);
+
+
+ return 0;
+}
+
+static struct platform_driver spitzled_driver = {
+ .probe = spitzled_probe,
+ .remove = spitzled_remove,
+#ifdef CONFIG_PM
+ .suspend = spitzled_suspend,
+ .resume = spitzled_resume,
+#endif
+ .driver = {
+ .name = "spitz-led",
+ },
+};
+
+static int __devinit spitzled_init(void)
+{
+ return platform_driver_register(&spitzled_driver);
+}
```

[PATCH 6/11] LED: Add LED device support for the zaurus corgi and spitz models

```
+
+static void spitzled_exit(void)
+{
+ platform_driver_unregister(&spitzled_driver);
+}
+
+module_init(spitzled_init);
+module_exit(spitzled_exit);
+
+MODULE_AUTHOR("Richard Purdie <rpurdie@xxxxxxxxxxxxxxxx>");
+MODULE_DESCRIPTION("Spitz LED driver");
+MODULE_LICENSE("GPL");
```

—

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/>

-
- Prev by Date: [*\[PATCH 0/11\] LED Class, Triggers and Drivers*](#)
 - Next by Date: [*\[PATCH 2/11\] LED: Add LED Class*](#)
 - Previous by thread: [*\[PATCH 0/11\] LED Class, Triggers and Drivers*](#)
 - Next by thread: [*\[PATCH 2/11\] LED: Add LED Class*](#)
 - Index(es):
 - ◆ [*Date*](#)
 - ◆ [*Thread*](#)