

[PATCH] RealTek(R) RTL-8169 PCI Gigabit Ethernet Card

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2003-08/8138.html>

From: Fernando Alencar Maróstica (*famarost_at_unimep.br*)

Date: 08/30/03

To: Marcelo Tosatti <marcelo@conectiva.com.br>

Date: 30 Aug 2003 16:53:49 -0300

Hi all,

Here's a patch for the new RealTek(R) RTL-8169 PCI Gigabit driver (drivers/net/r8169.c) in the 2.4.22-pre5 kernel. This patch fixes several functions, add new PCI suspend and resume stuff and cleanup code.

The patch applies against 2.4.22-pre5 and 2.5.75. I've tested the compilation with 2.4.22-pre5.

Cheers!

--

Fernando Alencar Maróstica
Graduate Student, Computer Science
Linux Register User Id #281457
University Methodist of Piracicaba
Department of Computer Science
email: famarost@unimep.br
homepage: <http://www.unimep.br/~famarost>

--- r8169.c.orig Sun Jul 13 13:32:55 2003

+++ r8169.c Sun Jul 13 13:55:02 2003

@@ -32,7 +32,15 @@

- Use ether_crc in stock kernel (linux/crc32.h)
- Copy mc_filter setup code from 8139cp
(includes an optimization, and avoids set_bit use)

-

+

+ <2003/07/13>

+ Fernando Alencar Marostica <famarost@unimep.br>

+ - Fix rtl8169_init_one()

+ - Fix rtl8169_get_stats()

+ - Fix rtl8169_remove_one()

+ - Add new rtl8169_resume() support

Linux-Kernel: [PATCH] RealTek(R) RTL-8169 PCI Gigabit Ethernet Card

+ - Add new rtl8169_suspend() support

+ - Several fixes and cleanup code

*/

```
#include <linux/module.h>
```

```
@@ -46,9 +54,9 @@
```

```
#include <asm/io.h>
```

```
#define RTL8169_VERSION "1.2"
```

```
−#define MODULENAME "r8169"
```

```
−#define RTL8169_DRIVER_NAME MODULENAME " Gigabit Ethernet driver " RTL8169_VERSION
```

```
−#define PFX MODULENAME ": "
```

```
+#define MODULE_NAME "r8169"
```

```
+#define RTL8169_DRIVER_NAME MODULE_NAME " Realtek(R) RTL-8169 PCI Gigabit Ethernet Card  
" RTL8169_VERSION
```

```
+#define PFX MODULE_NAME ": "
```

```
#ifndef RTL8169_DEBUG
```

```
#define assert(expr) \
```

```
@@ -101,11 +109,17 @@
```

```
#define RTL_R16(reg) readw (ioaddr + (reg))
```

```
#define RTL_R32(reg) ((unsigned long) readl (ioaddr + (reg)))
```

```
−static struct {
```

```
+
```

```
+struct rtl8169_card_info {
```

```
    const char *name;
```

```
−} board_info[] __devinitdata = {
```

```
− {
```

```
−"RealTek RTL8169 Gigabit Ethernet"},};
```

```
+ u8 mac_version; /* depend on RTL8169 docs */
```

```
+ u32 RxConfigMask; /* should clear the bits supported by this chip */
```

```
+};
```

```
+
```

```
+struct rtl8169_card_info rtl8169_card_info_tbl[] __devinitdata = {
```

```
+ { "RealTek RTL8169 Gigabit Ethernet", 0x02, 0xff7e1880 },
```

```
+ { "RealTek RTL8169s/8110s Gigabit Ethernet", 0x03, 0xff7e1880 },
```

```
+};
```

```
static struct pci_device_id rtl8169_pci_tbl[] __devinitdata = {
```

```
    {0x10ec, 0x8169, PCI_ANY_ID, PCI_ANY_ID, 0, 0, 0},
```

```
@@ -287,16 +301,19 @@
```

```
    unsigned char *RxBufferRings; /* Index of Rx Buffer */
```

```
    unsigned char *RxBufferRing[NUM_RX_DESC]; /* Index of Rx Buffer array */
```

```
    struct sk_buff *Tx_skbuff[NUM_TX_DESC]; /* Index of Transmit data buffer */
```

```
+#ifdef CONFIG_PM
```

```
+ u32 power_state[16];
```

```
+#endif
```

```
};
```

```
MODULE_AUTHOR("Realtek");
```

Linux-Kernel: [PATCH] RealTek(R) RTL-8169 PCI Gigabit Ethernet Card

```
-MODULE_DESCRIPTION("RealTek RTL-8169 Gigabit Ethernet driver");
+MODULE_DESCRIPTION("RealTek(R) RTL-8169 PCI Gigabit Ethernet Card");
+MODULE_LICENSE("GPL");
MODULE_PARM(media, "1-" __MODULE_STRING(MAX_UNITS) "i");

static int rtl8169_open(struct net_device *dev);
-static int rtl8169_start_xmit(struct sk_buff *skb, struct net_device *dev);
-static void rtl8169_interrupt(int irq, void *dev_instance,
- struct pt_regs *regs);
+static int rtl8169_start_xmit(struct sk_buff *skb, struct net_device *dev);
+static void rtl8169_interrupt(int irq, void *dev_instance, struct pt_regs *regs);
static void rtl8169_init_ring(struct net_device *dev);
static void rtl8169_hw_start(struct net_device *dev);
static int rtl8169_close(struct net_device *dev);
@@ -358,6 +375,7 @@
    int rc, i;
    unsigned long mmio_start, mmio_end, mmio_flags, mmio_len;
    u32 tmp;
+ int acpi_idle_state = 0, pm_cap;

    assert(pdev != NULL);
    assert(ioaddr_out != NULL);
@@ -365,20 +383,34 @@
    *ioaddr_out = NULL;
    *dev_out = NULL;

- // dev zeroed in alloc_etherdev
+ /* dev zeroed in alloc_etherdev */
    dev = alloc_etherdev(sizeof (*tp));
- if (dev == NULL) {
- printk(KERN_ERR PFX "unable to alloc new ethernet\n");
+ if (!dev) {
+ printk(KERN_ERR PFX "%s: Could not allocate new ethernet device.\n", pdev->slot_name);
    return -ENOMEM;
    }

    SET_MODULE_OWNER(dev);
    tp = dev->priv;

- // enable device (incl. PCI PM wakeup and hotplug setup)
+ /* enable device (incl. PCI PM wakeup and hotplug setup) */
    rc = pci_enable_device(pdev);
- if (rc)
+ if(rc) {
+ printk(KERN_ERR PFX "%s: unable to enable device\n", pdev->slot_name);
    goto err_out;
+ }
+
+ /* save power state before pci_enable_device overwrites it */
+ pm_cap = pci_find_capability(pdev, PCI_CAP_ID_PM);
+ if (pm_cap) {
```

Linux-Kernel: [PATCH] RealTek(R) RTL-8169 PCI Gigabit Ethernet Card

```
+ u16 pwr_command;
+ pci_read_config_word(pdev, pm_cap + PCI_PM_CTRL, &pwr_command);
+ acpi_idle_state = pwr_command & PCI_PM_CTRL_STATE_MASK;
+ }
+ else {
+ printk(KERN_ERR PFX "Cannot find PowerManagement capability, aborting.\n");
+ goto err_out_free_res;
+ }

        mmio_start = pci_resource_start(pdev, 1);
        mmio_end = pci_resource_end(pdev, 1);
@@ -400,10 +432,12 @@
    }

        rc = pci_request_regions(pdev, dev->name);
- if (rc)
+ if(rc) {
+ printk(KERN_ERR PFX "%s: Could not request regions.\n", pdev->slot_name);
        goto err_out_disable;
-
- // enable PCI bus-mastering
+ }
+
+ /* enable PCI bus-mastering */
        pci_set_master(pdev);

        // ioremap MMIO region
@@ -464,7 +498,6 @@
        struct rtl8169_private *tp = NULL;
        void *ioaddr = NULL;
        static int board_idx = -1;
- static int printed_version = 0;
        int i, rc;
        int option = -1, Cap10_100 = 0, Cap1000 = 0;

@@ -473,10 +506,15 @@

        board_idx++;

- if (!printed_version) {
- printk(KERN_INFO RTL8169_DRIVER_NAME " loaded\n");
- printed_version = 1;
+ #ifndef MODULE
+ {
+ /* when built-in, we only print version if device is found */
+ static int did_version;
+
+ if (!did_version++)
+ printk(KERN_INFO RTL8169_DRIVER_NAME " loaded\n");
        }
+ #endif
```

Linux-Kernel: [PATCH] RealTek(R) RTL-8169 PCI Gigabit Ethernet Card

```

    rc = rtl8169_init_board(pdev, &dev, &ioaddr);
    if (rc)
@@ -500,7 +538,7 @@
    dev->watchdog_timeo = TX_TIMEOUT;
    dev->irq = pdev->irq;
    dev->base_addr = (unsigned long) ioaddr;
-// dev->do_ioctl = mii_ioctl;
+//dev->do_ioctl = mii_ioctl;

    tp = dev->priv; // private data //
    tp->pci_dev = pdev;
@@ -526,7 +564,7 @@
    "%2.2x:%2.2x:%2.2x:%2.2x:%2.2x:%2.2x, "
    "IRQ %d\n",
    dev->name,
- board_info[ent->driver_data].name,
+ rtl8169_card_info_tbl[ent->driver_data].name,
    dev->base_addr,
    dev->dev_addr[0], dev->dev_addr[1],
    dev->dev_addr[2], dev->dev_addr[3],
@@ -627,28 +665,104 @@
    return 0;
}

-static void __devexit
-rtl8169_remove_one(struct pci_dev *pdev)
+/**
+ * rtl8169_get_stats: - Get rtl8169 read/write statistics
+ * @ethernet_device: The Ethernet Device to get statistics for
+ *
+ * Get TX/RX statistics for rtl8169
+ */
+struct net_device_stats *rtl8169_get_stats (struct net_device *ethernet_device)
{
- struct net_device *dev = pci_get_drvdata(pdev);
- struct rtl8169_private *tp = dev->priv;
+ struct rtl8169_private *tp = ethernet_device->priv;
+ void *ioaddr = tp->mmio_addr;
+ unsigned long flags;

- assert(dev != NULL);
- assert(tp != NULL);
+ if (netif_running(ethernet_device)) {
+ spin_lock_irqsave (&tp->lock, flags);
+ tp->stats.rx_missed_errors += ((unsigned long) readl (ioaddr + (RxMissed)));
+ writel ((0), ioaddr + (RxMissed));
+ spin_unlock_irqrestore (&tp->lock, flags);
+ }

- unregister_netdev(dev);

```

Linux-Kernel: [PATCH] RealTek(R) RTL-8169 PCI Gigabit Ethernet Card

```
- iounmap(tp->mmio_addr);
- pci_release_regions(pdev);
+ return &tp->stats;
+}

- // poison memory before freeing
- memset(dev, 0xBC,
- sizeof (struct net_device) + sizeof (struct rtl8169_private));
+/**
+ * rtl8169_remove_one: - Remove rtl8169 device
+ * @pci_device: The PCI Device to be removed
+ *
+ * Remove and release rtl8169 PCI Ethernet Device
+ */
+static void __devexit rtl8169_remove_one (struct pci_dev *pci_device)
+{
+ struct net_device *ethernet_device = pci_get_drvdata (pci_device);
+ struct rtl8169_private *tp = (struct rtl8169_private *) (ethernet_device->priv);

- pci_disable_device(pdev);
- kfree(dev);
- pci_set_drvdata(pdev, NULL);
+ if (!ethernet_device)
+ BUG();
+
+ unregister_netdev(ethernet_device);
+ iounmap((char *) (tp->mmio_addr));
+
+ #ifdef CONFIG_PM
+ pci_set_power_state (pci_device, 0);
+ pci_restore_state (pci_device, tp->power_state);
+ #endif
+
+ pci_release_regions(pci_device);
+ pci_disable_device(pci_device);
+ pci_set_drvdata(pci_device, NULL);
+
+ /* poison memory before freeing */
+ memset(ethernet_device, 0xBC, sizeof(struct net_device) + sizeof(struct rtl8169_private));
+ kfree(ethernet_device);
+}
+
+ #ifdef CONFIG_PM
+
+static int rtl8169_suspend (struct pci_dev *pci_device, u32 state)
+{
+ struct net_device *ethernet_device = pci_get_drvdata (pci_device);
+ struct rtl8169_private *tp = (struct rtl8169_private *) (ethernet_device->priv);
+ void *ioaddr = tp->mmio_addr;
+ unsigned long flags;
```

Linux-Kernel: [PATCH] RealTek(R) RTL-8169 PCI Gigabit Ethernet Card

```
+
+ if (!netif_running (ethernet_device))
+ return 0;
+
+ netif_device_detach (ethernet_device);
+ netif_stop_queue (ethernet_device);
+ spin_lock_irqsave (&tp->lock, flags);
+
+ /* Disable interrupts, stop Rx and Tx */
+ writew ((0), ioaddr + (IntrMask));
+ writeb ((0), ioaddr + (ChipCmd));
+
+ /* Update the error counts. */
+ tp->stats.rx_missed_errors += RTL_R32 (RxMissed);
+ writel ((0), ioaddr + (RxMissed));
+ spin_unlock_irqrestore (&tp->lock, flags);
+
+ return 0;
+}
+
+static int rtl8169_resume (struct pci_dev *pci_device)
+{
+ struct net_device *ethernet_device = pci_get_drvdata (pci_device);
+
+ if (!netif_running (ethernet_device))
+ return 0;
+
+ netif_device_attach (ethernet_device);
+ rtl8169_hw_start (ethernet_device);
+ netif_start_queue (ethernet_device);
+
+ return 0;
+}
+
+#endif /* CONFIG_PM */
+
+static int
+rtl8169_open(struct net_device *dev)
+{
+@@ -696,6 +810,7 @@
+
+     rtl8169_init_ring(dev);
+     rtl8169_hw_start(dev);
+ netif_start_queue(dev);
+
+     return 0;
+
+@@ -752,9 +867,6 @@
+
+     /* Enable all known interrupts by setting the interrupt mask. */
+     RTL_W16(IntrMask, rtl8169_intr_mask);
```

Linux-Kernel: [PATCH] RealTek(R) RTL-8169 PCI Gigabit Ethernet Card

```
-
- netif_start_queue(dev);
-
}

static void
@@ -1108,21 +1220,17 @@
    spin_unlock_irqrestore(&tp->lock, flags);
}

-struct net_device_stats *
-rtl8169_get_stats(struct net_device *dev)
- {
- struct rtl8169_private *tp = dev->priv;

- return &tp->stats;
- }

static struct pci_driver rtl8169_pci_driver = {
- .name = MODULENAME,
+ .name = MODULE_NAME,
    .id_table = rtl8169_pci_tbl,
    .probe = rtl8169_init_one,
    .remove = __devexit_p(rtl8169_remove_one),
- .suspend = NULL,
- .resume = NULL,
+ #ifdef CONFIG_PM
+ .suspend = rtl8169_suspend,
+ .resume = rtl8169_resume,
+ #endif
};

static int __init
```

-
To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@vger.kernel.org
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>