

# msleep\_interruptible() in ethtool ioctl and keyboard input

**Source:** <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-04/7353.html>

---

**From:** Venkatesan, Ganesh ([ganesh.venkatesan\\_at\\_intel.com](mailto:ganesh.venkatesan_at_intel.com))

**Date:** 04/29/05

Date: Fri, 29 Apr 2005 08:54:23 -0700

To: <[linux-kernel@vger.kernel.org](mailto:linux-kernel@vger.kernel.org)>

In response of ethtool -p eth? [time], e1000 calls msleep\_interruptible() to sleep for <time> seconds or forever if <time> is not specified. In the meantime the NIC LEDs are blinked. This works fine in all cases.

With 2.4.x kernels this continues to work fine, even if the device generates an interrupt (remove the cable, for instance) while blinking is ON.

With 2.6.0+ kernels when the cable is removed, the NIC LEDs blink fine but the keyboard stops responding (only KDB and CTL-ALT-DEL work, all other keystrokes are buffered but not echoed back/acted on) till msleep\_interruptible completes. Mouse works fine.

I went backwards through the 2.5.x kernels and found that this change in behavior happened between 2.5.50 (where this works fine) and 2.5.51 (where this is broken). A slew of console driver changes went into 2.5.51 but I have not been able to locate which change is the cause for this behavior.

Please send me comments/ideas for further tests/questions for more data.

Following is the logic that implements the blinking:

```
static void
e1000_led_blink_callback(unsigned long data)
{
    struct e1000_adapter *adapter = (struct e1000_adapter *) data;

    if(test_and_change_bit(E1000_LED_ON, &adapter->led_status))
        e1000_led_off(&adapter->hw);
    else
        e1000_led_on(&adapter->hw);

    mod_timer(&adapter->blink_timer, jiffies + E1000_ID_INTERVAL);
}
```

## Linux–Kernel: msleep\_interruptible() in ethtool ioctl and keyboard input

```
}  
  
static int  
e1000_phys_id(struct net_device *netdev, uint32_t data)  
{  
    struct e1000_adapter *adapter = netdev->priv;  
  
    if(!data || data > (uint32_t)(MAX_SCHEDULE_TIMEOUT / HZ))  
        data = (uint32_t)(MAX_SCHEDULE_TIMEOUT / HZ);  
  
    if(!adapter->blink_timer.function) {  
        init_timer(&adapter->blink_timer);  
        adapter->blink_timer.function =  
e1000_led_blink_callback;  
        adapter->blink_timer.data = (unsigned long) adapter;  
    }  
  
    e1000_setup_led(&adapter->hw);  
    mod_timer(&adapter->blink_timer, jiffies);  
  
    msleep_interruptible(data * 1000);  
    del_timer_sync(&adapter->blink_timer);  
    e1000_led_off(&adapter->hw);  
    clear_bit(E1000_LED_ON, &adapter->led_status);  
    e1000_cleanup_led(&adapter->hw);  
  
    return 0;  
}
```

–  
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/>