Subject: [PATCH 4/5] fixing errors handling during pci_driver resume stage [misc]
Posted by Dmitriy Monakhov on Tue, 09 Jan 2007 09:01:47 GMT
View Forum Message <> Reply to Message

pci drivers have to return correct error code during resume stage in
case of errors.
Signed-off-by: Dmitriy Monakhov <dmonakhov@openvz.org>

```
diff --git a/drivers/ide/pci/sc1200.c b/drivers/ide/pci/sc1200.c
index ff80937..a426905 100644
--- a/drivers/ide/pci/sc1200.c
+++ b/drivers/ide/pci/sc1200.c
@@ -394,10 +394,16 @@ static int sc1200_suspend (struct pci_de
 static int sc1200_resume (struct pci_dev *dev)
 {
  ide_hwif_t *hwif = NULL;
+ int err;

  pci_set_power_state(dev, PCI_D0); // bring chip back from sleep state
  dev->current_state = PM_EVENT_ON;
- pci_enable_device(dev);
+ err = pci_enable_device(dev);
+ if (err) {
+  dev_err(&dev->dev, "Cannot enable PCI device, aborting.\n");
+  return err;
+ }
+
  //
  // loop over all interfaces that are part of this pci device:
  //
diff --git a/drivers/misc/tifm_7xx1.c b/drivers/misc/tifm_7xx1.c
index 2ab7add..d659ad9 100644
--- a/drivers/misc/tifm_7xx1.c
+++ b/drivers/misc/tifm_7xx1.c
@@ -274,10 +274,15 @@ static int tifm_7xx1_suspend(struct pci_
 static int tifm_7xx1_resume(struct pci_dev *dev)
 {
  struct tifm_adapter *fm = pci_get_drvdata(dev);
- unsigned long flags;
+ unsigned long flags, err;

  pci_restore_state(dev);
-        pci_enable_device(dev);
+ err = pci_enable_device(dev);
+ if (err) {
+  dev_err(&dev->dev, "Cannot enable PCI device, aborting.\n");
+  return err;
```

```
+ }
+
        pci_set_power_state(dev, PCI_D0);
        pci_set_master(dev);

diff --git a/drivers/mmc/sdhci.c b/drivers/mmc/sdhci.c
index c2d13d7..736f74c 100644
--- a/drivers/mmc/sdhci.c
+++ b/drivers/mmc/sdhci.c
@@ -1109,7 +1109,11 @@ static int sdhci_resume (struct pci_dev

  pci_set_power_state(pdev, PCI_D0);
  pci_restore_state(pdev);
- pci_enable_device(pdev);
+ ret = pci_enable_device(pdev);
+ if (ret) {
+  dev_err(&pdev->dev, "Cannot enable PCI device, aborting.\n");
+  return ret;
+ }

  for (i = 0;i < chip->num_slots;i++) {
   if (!chip->hosts[i])
diff --git a/drivers/parisc/superio.c b/drivers/parisc/superio.c
index 1fd97f7..4428ffa 100644
--- a/drivers/parisc/superio.c
+++ b/drivers/parisc/superio.c
@@ -199,7 +199,8 @@ superio_init(struct pci_dev *pcidev)
  pci_write_config_word (pdev, PCI_COMMAND, word);

  pci_set_master (pdev);
- pci_enable_device(pdev);
+ if (pci_enable_device(pdev))
+  return;

  /*
   * Next project is programming the onboard interrupt controllers.
@@ -275,6 +276,7 @@ superio_init(struct pci_dev *pcidev)
    SUPERIO, (void *)sio)) {

   printk(KERN_ERR PFX "could not get irq\n");
+  pci_disable_device(pdev);
   BUG();
   return;
  }
```