Dexon
Tecnologias Digitais

Guia de Instalação do Leitor e Gravador de Cartões *e-Smart*

Sistema Operacional Linux

- Realiza operações de forma rápida e segura.
- É conectado diretamente em computadores e servidores através de um Cabo USB.
- Atende todos os requisitos da Certificação Digital.
- É ideal para entidades bancarias e serviços de empresas na internet

```
======================================================
=      Alcormicro Smart card reader                  =
=    Linux TARBALL Package Release Note              =
======================================================

If the version of  libusb is under 1.0.8 , it need to update to latest one.
Otherwise skip  step1.

==========================================

#Step1. Install  libusb

 # tar xjf libusb-1.0.8.tar.bz2
 cd libusb-1.0.8
 ./configure
 make
 make install         # ( Login as a root to install driver )

#===========================================

#Disable libudev(./configure will fail with can't find libudev if no disable
libudev)
#note:If ./configure display the note of can't find libusb.* then check the
PKG_CONFIG_PATH,Using the command:echo $PKG_CONFIG_PATH.

#==========================================

#Step2. Install  pcsc-lite

 PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig;export
PKG_CONFIG_PATH

 # tar xjf pcsc-lite-1.7.4.tar.bz2

 cd ../pcsc-lite-1.7.4
 ./configure --disable-libudev
 make
 make install          #(Login as a root to install driver)

#Step3. Install ccid driver

 # tar xjf ccid-1.4.5-alcor.tar.bz2
  cd ../ccid-1.4.5-alcor
 ./configure
 make
 make install         # (Login as a root to install driver)
 cd src
 cp 92_pcscd_ccid.rules /etc/udev/rules.d/

note: you can test it follow these steps
 $ cd pcsc-lite-1.7.4/doc/example
 $ ./pcsc_demo


http://lesswatts.org/projects/devices-power-management/usb.php
Device and Bus Power Management
```

USB selective suspend

This is a project in progress for the USB subsystem. USB selective suspend allows you to suspend a USB device on demand. If one device doesn't support selective suspend, then the entire bus must remain active. This not only consumes USB bus power unnecessarily, but also prevents the CPU from entering lower power states.

A white paper describing USB selective suspend can be found here.

Autosuspend on Linux

For a device to be autosuspended in Linux, it must have in-kernel driver support. Currently there are several types of USB devices that have autosuspend support:

- printers
- hubs
- some USB Ethernet devices
- USB LCDs

Although kernel drivers may support autosuspend, some USB devices may not properly implement autosuspend. These devices may behave in unexpected ways, or simply not work after the kernel attempts to suspend them. Often a physical disconnection from the bus will fix the problem, but only until the kernel attempts to suspend the device again.

If you see these types of problems, please send mail to linux-usb-users@lists.sourceforge.net. Include output from `lsusb -v`, dmesg output with CONFIG_USB_DEBUG turned on, and a description of the symptoms. We would also appreciate a note that your USB device actually works with autosuspend.

Enabling Autosuspend

To enable autosuspend, you must recompile your kernel with CONFIG_USB_SUSPEND. (As of 2.6.23-rc6, this feature is marked "experimental".) You may also want to enable CONFIG_USB_DEBUG so you can see suspend and resume messages via dmesg.

Autosuspending USB devices

To attempt to autosuspend your USB device, first use lsusb as root to find out the bus number and device number of your usb device:

```
 $ lsusb
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 009: ID 058f:9540 Alcor Micro Corp.
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Then find your device's directory in /sys/bus/usb/devices/. Look in directories that are named with two numbers separated with a dash:

```
 $ cd /sys/bus/usb/devices
 /sys/bus/usb/devices# ls
1-0:1.0  3-0:1.0  4-1      5-0:1.0  usb2  usb4
2-0:1.0  4-0:1.0  4-1:1.0  usb1     usb3  usb5
 /sys/bus/usb/devices# cat 4-1/busnum
4
 /sys/bus/usb/devices# cat 4-1/devnum
9
```

We know the USB to ethernet device's directory is 1-2 because the device and bus numbers match the lsusb output. Now we can tell the kernel that it should suspend this device automatically if it is not being used. First we set the idle timeout to 2 seconds:

```
/sys/bus/usb/devices# echo 2 > 4-1/power/autosuspend
```

The timeout can be set to any integer number of seconds. If set to -1, the device will not autosuspend. Then we make sure the kernel will automatically suspend the device, and resume the device if data needs to be transferred:

```
/sys/bus/usb/devices# echo auto > 4-1/power/level
```

Other options to echo to this file are "on" and "suspend":

- "on" will force the device to be on all the time.
- "suspend" will permanently suspend the device until the user echoes "on" or "auto" to this file.
  (Note that this is a simplification, since the value of the power/wakeup file may allow the device to signal a remote wakeup.)

For a more complete description of USB power management, see the file Documentation/usb/power-management.txt, which is in kernel sources 2.6.24-rc2 and later.