



**NOTES:**

# DRIVER-LINUX2.6- PCI307

## S O M M A I R E

<b>A.</b>	<b>INSTALLATION DU MATERIEL.....</b>	<b>4</b>
<b>B.</b>	<b>INSTALLATION LOGICIEL .....</b>	<b>5</b>
<b>D.</b>	<b>FONCTIONS LOGICIELLES .....</b>	<b>6</b>
	<i>1. PCI307_CTLDEBUG.....</i>	<i>9</i>
	<i>2. PCI307_CTLINIT.....</i>	<i>10</i>
	<i>3. PCI307_CTLVERS.....</i>	<i>11</i>
	<i>4. PCI307_CTLNICPT.....</i>	<i>12</i>
	<i>5. PCI307_CTLSTOPT.....</i>	<i>13</i>
	<i>6. PCI307_CTLSTART.....</i>	<i>14</i>
	<i>7. PCI307_CTLBT2REP.....</i>	<i>15</i>
	<i>8. PCI307_CTLBT2NONREP.....</i>	<i>16</i>
	<i>9. PCI307_CTLSTARTC.....</i>	<i>17</i>
	<i>10. PCI307_CTLSTOPC.....</i>	<i>18</i>
	<i>11. PCI307_CTLFILT.....</i>	<i>19</i>
	<i>12. PCI307_CTLLECALL.....</i>	<i>20</i>
	<i>13. PCI307_CTLLECTU.....</i>	<i>21</i>
<b>E.</b>	<b>APPLICATION .....</b>	<b>22</b>

## A. INSTALLATION DU MATERIEL



- ⇒ L'ordinateur doit être hors tension.
- ⇒ Déchargez-vous de votre potentiel électrostatique éventuel en touchant une surface métallique.
- ⇒ Installez les cartes en prenant soin de les configurer suivant les spécifications techniques et vos propres besoins.

Rappelons que nos cartes **PCI** sont « Plug and Play » et vous dispensent de configurer l'interface avec l'ordinateur. (voir documentation technique de chaque carte et le chapitre annexe).

- ⇒ Mettez l'ordinateur sous tension.



### Nota :

Par souci de simplification, et de bon déroulement de la procédure d'installation, nous vous conseillons de ne pas changer les autres cartes d'emplacement lors de l'ajout de la carte **PCI 307**.

Maintenant l'ordinateur a identifié le nouveau matériel et il est prêt à l'emploi.

Le driver tourne au niveau Kernel.

Chaque ressource de la carte peut être sollicitée au niveau « user » en faisant appel aux différents IOControl que nous allons expliquer dans le chapitre suivant.

Nous vous indiquons que les anciens logiciels applicatifs (mode user) sont compatibles avec ce nouveau driver.

Une simple re-compilation est nécessaire.

## B. INSTALLATION LOGICIEL

Le driver Linux 2.6 de la carte **PCI307** est fourni avec les fichiers sources :

```
pci307.c
pci307.h
makefile
pci307_install
example/pci307_example.c
example/makefile
```

La compilation s'effectue tout simplement en exécutant la commande « make ». Vous obtenez alors un fichier nommé pci307.ko près à être utiliser pour le driver.

Deux méthodes s'offrent à vous pour installer le driver : la méthode automatique ou la méthode manuelle.

La méthode automatique consiste à exécuter la commande « ./pci307\_install start ». La désinstallation du driver s'exécute avec la commande « ./pci307\_install stop ». la commande « ./pci307\_install status » donne l'état du driver et « ./pci307\_install restart » redémarre la carte **PCI307**.

La méthode manuelle est plus longue à mettre en œuvre. Pour installer le driver, il est nécessaire de lancer la commande « insmod ./pci307.ko ». Vous devez également assigner la carte. Pour cela, vous devez lancer la commande « mknod -m 666 /dev/pci307 c <major> <minor> ».

« Major » est unique et est toujours connu. Pour le connaître, vous pouvez l'identifier grâce à la commande « cat /proc/devices | grep pci307 ». Vous devez obtenir normalement un résultat du genre « 254 pci307 ».

« Minor » est le numéro de la carte. Votre première carte a le numéro 0, votre seconde le numéro 1, et ainsi de suite. Si vous avez une seule carte **PCI307** et votre « major » est 254, vous devez assigner votre carte avec la commande suivante : « mknod -m 666 /dev/pci307 c 254 0 ».

## D. FONCTIONS LOGICIELLES

### Rappels :

Linux exécute les applications dans le mode user.

Uniquement les drivers mode Kernel peuvent tourner en mode Privilégié.

C'est le mode qui permet d'accéder aux périphériques.

Dans ce contexte, pour gérer au mieux les périphériques, le programmeur utilise les IOCTLs pour communiquer directement avec le driver mode Kernel.

Les IOCTLs s'utilisent directement dans une application ou par l'intermédiaire d'une DLL.

Nous vous conseillons de suivre l'exemple d'application fourni sur la disquette.

Le fichier source est nommé pci307\_example.c

Ce programme reprend principalement les IOCTLs du driver. Les fichiers sources du driver sont fournis avec la disquette laissant ainsi à l'utilisateur le libre choix de modifier ce dernier, si cela s'avérerait nécessaire.

### Les fonctions init\_module

La fonction init\_module appelle une la fonction init\_PCI307() qui assure les tâches suivantes :

Identification de la carte PCI307 par son Vendor\_Id, Device\_Id (Cf norme Bus PCI).

Mapping des zones mémoires de la carte

Déclaration des structures internes au driver

Déclaration de la routine d'interruption

Initialisation de la carte

Initialisation des structures internes

### La fonction pci307\_open()

Cette fonction assure la connexion entre l'application et le driver. Elle retourne un pointeur sur un fichier spécial qui sera passé en paramètre lors de tous les appels aux autres fonctions du driver par l'application.

Cette fonction est appelée par la fonction C :

```
fdx = open(nom_fichier_spécial,mode, 0); .
```

### La fonction pci307\_release()

Cette fonction assure la déconnexion entre l'application et le driver. Elle est appelée par la fonction C : **close(fdx);** où fdx est le descripteur de fichier retourné par open().

## La fonction pci307\_ioctl()

La fonction **IOCTL** envoie directement au driver un code de contrôle. Nous l'utiliserons pour envoyer tous les codes de contrôle supportés par le driver de la carte.

```
Int (*ioctl)(
    struct file hDevice,      // handle to device of interest
    unsigned int Command,    // control code of operation to perform
    unsigned long *arg,      // pointer to data structure
);
```

### Structure pci307\_ioctl

```
typedef struct pci307_ioctl
{
    unsigned    adr;    /* Numero du compteur          */
    long        val;    /* Valeur du compteur          */
    float       freq;   /* fréquence pour rapport cyclique*/
    unsigned    typ;    /* Type de fonctionnement     */
    unsigned    cpl;    /* Info. complementaire        */
    int         err;    /* Code retour d'erreur        */
    int         mod;    /* Mode d'attente du decompneur */
    pid_t       pid;    /* Pid du process i            */
    int         sig;    /* Signal a emettre            */
} pci307_ioctl;
```

<b><i>IOCTL</i></b>	<b><i>Description</i></b>
<b>PCI307_CTLDEBUG</b>	Positionne le mode DEBUG
<b>PCI307_CTLINIT</b>	Cette commande initialise la carte PCI307
<b>PCI307_CTLVERS</b>	Retourne le n° de version du driver
<b>PCI307_CTLINICPT</b>	Assignment directe d'une tension à une sortie analogique
<b>PCI307_CTLSTOPT</b>	Cette commande arrête les acquisitions continues des compteurs
<b>PCI307_CTLSTART</b>	Cette commande lance les acquisitions continues des compteurs
<b>PCI307_CTLBT2REP</b>	Cette commande active la base de temps BT2 de la carte PCI307 en mode répétitif
<b>PCI307_CTLBT2NONREP</b>	Cette commande désactive la base de temps BT2 de la carte PCI307 en mode répétitif.
<b>PCI307_CTL_STARTC</b>	Lancement du comptage/décomptage d'un compteur
<b>PCI307_CTL_STOPC</b>	Arrêt du comptage/décomptage d'un compteur
<b>PCI307_CTLARTCPT</b>	Armement de la détection de débordement d'un décompteur (décompteur à 0)
<b>PCI307_CTLFILT</b>	Positionnement de la constante de filtrage
<b>PCI307_CTLLECALL</b>	Lecture des valeurs en cours des 4 compteurs
<b>PCI307_CTLLECTU</b>	Lecture de la valeur en cours d'un compteur

## 1. *PCI307\_CTLDEBUG*

### Description

Cette commande exécute le mode DEBUG pour la carte **PCI307**.

### Syntaxe

```
hDevice = FileHandle           // Handle to device of interest
Command = PCI307_CTLDEBUG      // Control Code
*arg = &param                   // parameter
```

### Paramètres de l'argument

Aucun paramètre à transmettre.

un entier :

Si param = 0 , pas de trace

Si param != 0, le driver génère des traces dans le fichier /var/log/messages.

Ces traces peuvent être lues par la commande Linux **dmesg**.

### Code de retour

La fonction ioctl() retourne CP\_CTL\_OK.

## 2. *PCI307\_CTLINIT*

### **Description**

Cette commande initialise la carte PCI307 et les structures internes du driver. Les canaux de comptage devront être configurés avant utilisation.

### **Syntaxe**

```
hDevice = FileHandle           // Handle to device of interest
Command = PCI307_CTLINIT      // Control Code
*arg = & param                 // structure de type pci307_ioctl
```

### **Paramètres de l'argument**

Un pointeur sur une structure de type pci307\_ioctl.

### **Code de retour**

```
CP_CTL_OK      Pas d'erreur.
CP_CTL_ERR     En cas d'erreur.
```

### 3. PCI307\_CTLVERS

#### Description

Cette commande retourne le n° de version du driver.

#### Syntaxe

```
hDevice = FileHandle           // Handle to device of interest
Command = PCI307_CTLVERS      // Control Code
*arg = &param                  // structure de type pci307_ioctl
```

#### Paramètres de l'argument

Un pointeur sur une structure de type pci307\_ioctl.

#### Paramètres retournés :

param.val : Cet élément de la structure contient le numéro de version du driver

#### Code de retour

```
CP_CTL_OK           Pas d'erreur.
CP_ERR_TIMEOUT      En cas d'échec.
```

## 4. PCI307\_CTLINICPT

### Description

Cette commande configure le compteur sélectionné dans le mode défini par le paramètre typ.

### Syntaxe

```
hDevice = FileHandle           // Handle to device of interest
Command = PCI307_CTLINICPT     // Control Code
*arg = &param                   // structure de type pci307_ioctl
```

### Paramètres de l'argument

Un pointeur sur une structure de type pci307\_ioctl.

param.adr : contient le numéro de compteur [0 ..3]

param.typ : contient la fonctionnalité du compteur :

CP\_C\_COMPTE : Compteur en mode comptage

CP\_C\_DECOMP : Compteur en mode décomptage

CP\_C\_FREPER : Mesure de fréquence/période

CP\_C\_RCYCL : Mesure de rapport cyclique

CP\_C\_NO : Déconfiguration d'un compteur

param.freq : Si param.typ = CP\_C\_RCYCL, param.freq contient la fréquence en Hz du signal à mesurer (param.freq est un nombre flottant).

param.val : Si param.typ = CP\_C\_DECOMP, param.val contient la valeur initiale du compteur (param.val est un nombre entier long).

### Paramètres retournés :

Param.err : Cet élément de la structure contient le code d'erreur de la commande

CP\_ERR\_CFG : Erreur sur la fonctionnalité

CP\_ERR\_NVOIE : Erreur sur le numéro de compteur

CP\_ERR\_INC : Commande incompatible avec le device

CP\_ERR\_VAL : Erreur valeur de fréquence : param.freq = 0

### Code de retour

CP\_CTL\_OK Pas d'erreur.

CP\_CTL\_ERR En cas d'erreur.

## 5. PCI307\_CTLSTOPT

### Description

Cette commande arrête les acquisitions continues des compteurs.

### Syntaxe

```
hDevice = FileHandle           // Handle to device of interest
Command = PCI307_CTLSTOPT     // Control Code
*arg = &data                   // structure de type pci307_ioctl
```

### Paramètres de l'argument

Un pointeur sur une structure de type pci307\_ioctl.

### Paramètres retournés :

Param.err : Cet élément de la structure contient le code d'erreur de la commande  
CP\_ERR\_DSTO : Les acquisitions continues ne sont pas actives  
CP\_CTL\_OK si la commande est correctement réalisée

### Code de retour

CP_CTL_OK	Pas d'erreur.
CP_CTL_ERR	En cas d'erreur.

## 6. *PCI307\_CTLSTART*

### Description

Cette commande lance les acquisitions continues des compteurs.

### Syntaxe

```
hDevice = FileHandle           // Handle to device of interest
Command = PCI307_CTLSTART      // Control Code
*arg = &param                  // structure de type pci307_ioctl
```

### Paramètres de l'argument

Un pointeur sur une structure de type pci307\_ioctl.

### Paramètres retournés :

param.err : Cet élément de la structure contient le code d'erreur de la commande  
CP\_ERR\_DSTA : Les acquisitions continues sont déjà actives  
CP\_CTL\_OK si la commande est correctement réalisée

### Code de retour

CP_CTL_OK	Pas d'erreur.
CP_CTL_ERR	En cas d'erreur.

## 7. PCI307\_CTLBT2REP

### Description

Cette commande active la base de temps BT2 de la carte PCI307 en mode répétitif. Il en résulte une interruption périodique toutes les 10 ms.

**Ce mode est obligatoire pour la lecture des rapports cycliques car c'est lors de cette interruption que la détection de niveau continu sur le signal d'entrée est réalisée.**

### Syntaxe

```
hDevice = FileHandle           // Handle to device of interest
Command = PCI307_CTLBT2REP     // Control Code
*arg = &param                  // structure de type pci307_ioctl
```

### Paramètres de l'argument

Un pointeur sur une structure de type pci307\_ioctl.

### Paramètres retournés :

Aucun

### Code de retour

CP\_CTL\_OK            Pas d'erreur.

## 8. *PCI307\_CTLBT2NONREP*

### **Description**

Cette commande désactive la base de temps BT2 de la carte PCI307 en mode répétitif.

L'interruption périodique existe toujours, mais la fréquence n'est pas constante et oscille entre 10 et 11 ms. Ceci est du au temps de synchronisation/réarmement de l'horloge de la carte.

**A l'initialisation, la carte est dans le mode BT2 non répétitive.**

### **Syntaxe**

```
hDevice = FileHandle           // Handle to device of interest
Command = PCI307_CTLBT2NONREP // Control Code
*arg = &param                  // structure de type pci307_ioctl
```

### **Paramètres de l'argument**

Un pointeur sur une structure de type pci307\_ioctl.

### **Paramètres retournés :**

Aucun

### **Code de retour**

CP\_CTL\_OK      Pas d'erreur.

## 9. PCI307\_CTLSTARTC

### Description

Cette commande permet le lancement du comptage/décomptage d'un compteur.

### Syntaxe

```
hDevice = FileHandle           // Handle to device of interest
Command = PCI307_CTLSTARTC    // Control Code
*arg = &param                  // structure de type pci307_ioctl
```

### Paramètres de l'argument

Un pointeur sur une structure de type pci307\_ioctl.  
param.adr : contient le numéro de compteur [0 ..3]

### Paramètres retournés :

Aucun

### Code de retour

CP_CTL_OK	Pas d'erreur.
CP_ERR_DSTA	Erreur le compteur est déjà en start
CP_ERR_NFREE	Erreur le compteur n'est pas configuré pour ce mode
CP_ERR_NVOIE	Erreur le numéro de compteur est incorrecte

## 10. PCI307\_CTLSTOPC

### Description

Cette commande permet l'arrêt du comptage/décomptage d'un compteur.

### Syntaxe

```
hDevice = FileHandle           // Handle to device of interest
Command = PCI307_CTLSTOPC     // Control Code
*arg = &param                  // structure de type pci307_ioctl
```

### Paramètres de l'argument

Un pointeur sur une structure de type pci307\_ioctl.  
param.adr : contient le numéro de compteur [0 ..3]

### Paramètres retournés :

Aucun

### Code de retour

CP_CTL_OK	Pas d'erreur.
CP_ERR_DSTO	Erreur le compteur est déjà en stop
CP_ERR_NFREE	Erreur le compteur n'est pas configuré pour ce mode
CP_ERR_NVOIE	Erreur le numéro de compteur est incorrecte

## 11. PCI307\_CTLFILT

### Description

Positionnement de la constante de filtrage (permet de filtrer les signaux d'entrés en cas de signaux bruités)

### Syntaxe

```
hDevice = FileHandle           // Handle to device of interest
Command = PCI307_CTLFILT      // Control Code
*arg = &param                  // Pointer to the board structure
```

### Paramètres de l'argument

Un pointeur sur une structure de type pci307\_ioctl.  
param.adr : contient le numéro de compteur [0 ..3]  
param.val : contient la valeur de filtrage

### Paramètres retournés :

Aucun

### Code de retour

CP_CTL_OK	Pas d'erreur.
P_CTL_ERR	Erreur de programmation

## 12. PCI307\_CTLLECALL

### Description

Cette commande permet de lire les valeurs en cours des 4 compteurs.

### Syntaxe

```
hDevice = FileHandle           // Handle to device of interest
Command = PCI307_CTLLECALL     // Control Code
*arg = &param                   // Pointer to the board structure
```

### Paramètres de l'argument

Un pointeur sur une structure de type pci307\_ioctl.

### Paramètres retournés :

param.valeurs[0-3] : Retourne la valeur des 4 compteurs

### Code de retour

CP\_CTL\_OK            Pas d'erreur.

### 13. PCI307\_CTLLECTU

#### Description

Cette commande permet de lire la valeur en cours d'un compteur.

#### Syntaxe

```
hDevice = FileHandle           // Handle to device of interest
Command = PCI307_CTLLECTU      // Control Code
*arg = &param                  // Pointer to the board structure
```

#### Paramètres de l'argument

Un pointeur sur une structure de type pci307\_ioctl.  
param.adr : contient le numéro de compteur [0 ..3]

#### Paramètres retournés :

param.val : Retourne la valeur du compteur

#### Code de retour

```
CP_CTL_OK      Pas d'erreur.
CP_ERR_NVOIE   Erreur le numéro de compteur est incorrecte
```

## E. APPLICATION

Ce programme est fourni dans le fichier zip.

Il reprend les codes IOCTL dans des exemples simples, et vous permettra avec son code source de démarrer votre projet rapidement.

Le programme source peut être directement compilé et modifié.

Pour le lancer, il suffit d'exécuter la commande « *./pci307\_example* ».

Cette application permet le comptage, le décomptage, le mode quadrature et le rapport cyclique



Attention : pour réaliser ce test il faut câbler **A\_ISO(+/-)** et **B\_ISO(+/-)**

**ANNEXE :**

**PC INTERRUPT LEVELS**

IRQ0	Timer Tick
IRQ1	Keyboard Controller
IRQ2	Cascade interrupt
IRQ3	COM2, COM4
IRQ4	COM1, COM3
IRQ5	Audio
IRQ6	Diskette
IRQ7	LPT1, LPT3
IRQ8	Real time clock
IRQ9	(PCI device) default video
IRQ10	(PCI device) default SCSI
IRQ11	(PCI device) default audio
IRQ12	Mouse interrupt
IRQ13	Math co-processor
IRQ14	IDE primary
IRQ15	IDE secondary



**Ressource éventuellement exploitable par une carte PCI**