



**SELIAtec S.A.S.**  
**53, rue de Rountzenheim**  
**B.P. 34**  
**67620 SOUFFLENHEIM**  
**Tél. : 03 88 86 68 54**  
**Fax : 03 88 86 74 76**  
**Email : [infos@seliatec.com](mailto:infos@seliatec.com)**

## **PI 12**

Carte multifonctions pour bus PCI  
Entrées et sorties industrielles  
analogiques, numériques,  
comptage, communication série

## NOTIFICATION

Les informations contenues dans ce manuel sont susceptibles d'être modifiées sans préavis.

**SELIAtec S.A.S. NE PEUT ÊTRE TENU RESPONSABLE DES OMISSIONS TECHNIQUES OU RÉDACTIONNELLES, NI DES DOMMAGES CONSÉCUTIFS À LA FOURNITURE OU À L'UTILISATION DU PRODUIT ET DE SON MANUEL.**

**AVANT LA MISE EN SERVICE, IL INCOMBE A L'UTILISATEUR DE VERIFIER QUE LES CARACTERISTIQUES TECHNIQUES REPENDENT AUX SPECIFICATIONS DONNEES POUR LE PRODUIT.**

Le présent manuel contient des informations protégées par copyright. Aucune partie du présent document ne peut être photocopiée ou reproduite sous quelque forme que ce soit sans l'accord écrit préalable de SELIAtec S.A.S.

Les noms et marques cités dans ce manuel sont déposés par les fabricants respectifs.

## Conventions



Cette rubrique vous informe qu'une commande est à saisir au clavier.



Cette rubrique vous informe que la disquette fournie avec la carte est à utiliser pour la suite des commandes.

### NOTE

Cette rubrique permet de clarifier certaines informations.



**ATTENTION** Cette rubrique vous recommande de suivre scrupuleusement les instructions sous peine de provoquer une détérioration du matériel ou une perte de données.



**IMPORTANT** Soyez prudent. Cette rubrique vous avertit que la mauvaise utilisation du matériel ou un non-respect des instructions risquent d'entraîner des blessures corporelles.



### DIRECTIVE EUROPEENNE 89/336/CEE

Conformité des produits à la directive 89/336/CEE. Selon les normes européennes harmonisées relatives aux appareils de traitement de l'information :

#### - émission NF EN 55022, - immunité NF EN 50082-1.

Les environnements couverts sont les sites résidentiels, les locaux commerciaux et l'industrie légère intérieurs et extérieurs. Ex : ateliers, laboratoires, centres de service, bureaux, etc...

Les sites qui sont caractérisés comme étant alimentés directement en basse tension par le réseau public sont considérés comme résidentiels, commerciaux ou l'industrie légère. La conformité à des normes relatives à d'autres sites peut être vérifiée sur demande.

En utilisation normale les cartes SELIAtec sont implantées dans un PC et peuvent être considérées comme étant un composant faisant partie d'une installation complète.

Il est donc important que l'intégrateur des produits s'assure que le PC lui-même ainsi que tous les équipements et le câblage extérieur répondent dans leur ensemble aux Directives Européennes CEM.

Nous préconisons pour la connexion de tous les signaux d'entrées et de sorties d'utiliser du câble, des connecteurs et des capots de connecteurs blindés. Il est important de veiller à la continuité du blindage entre le connecteur de la carte, le capot et le câble.

Pour plus de renseignements, voir en annexe CEM



Bien respecter les règles concernant la protection de l'environnement lorsque vous mettez au rebut des déchets électroniques.

## TABLE DES MATIÈRES

<b>1. <u>Introduction</u></b>	<b>5</b>
1.1 Contenu de l'emballage .....	5
1.2 Domaines d'applications.....	5
1.3 Caractéristiques d'utilisation .....	6
1.4 Caractéristiques physiques .....	7
<b>2. <u>Démarrage rapide</u></b>	<b>7</b>
<b>3. <u>Installation</u></b>	<b>8</b>
3.1 Ouverture de l'ordinateur .....	8
<b>4. <u>Configuration</u></b>	<b>10</b>
4.1. Adresse de base.....	10
4.2. Fonction chien de garde (watchdog) .....	11
4.3. Définition des adresses .....	11
4.4. Interruptions.....	12
4.5. Concept Boot&Work .....	13
4.6. Connectique.....	13
<b>5. <u>Installation de la DLL</u></b>	<b>14</b>
5.1 Installation du pilote de la carte PI 12.....	14
5.2 Descriptif des fichiers installés .....	14
5.3 Vérification de l'installation .....	15
5.4 Désinstallation .....	15
5.5.Problème à l'installation.....	16
5.5.1 Problème à l'appel de programme :.....	17
5.5.2 Problèmes à l'appel d'un exécutable : .....	17
5.5.3 Problème à la compilation d'une application .....	17
5.6. Généralités sur l'utilisation de la librairie de fonctions.....	17
<b>6. Fonctions relatives à la carte PI 12</b>	<b>20</b>
<b>7. Fonctions relatives aux modules</b>	<b>27</b>
<b>8. Fonctions de bas niveau</b>	<b>27</b>
<b>9. Exemple en Visual Basic 6.0</b>	<b>29</b>

## 1. Introduction

### 1.1 Contenu de l'emballage

Déballez soigneusement votre carte d'extension. Vérifiez que vous avez bien reçu tous les éléments suivants:

- La carte d'extension PC dans son enveloppe antistatique.
- Le manuel d'utilisation.
- La disquette contenant les drivers, un logiciel de test et des exemples.

Si un de ces éléments manquait ou était endommagé, contactez-nous immédiatement au 03 88 86 68 54.

### 1.2 Domaines d'applications

La carte d'extension PI 12 permet d'équiper votre PC d'entrées et de sorties industrielles pour des applications d'automatisme, de régulation et de robotique.

- ◆ Mesure : ex. pesage, déplacement, contrôle dimensionnel
- ◆ Pilotage de machines : ex. industrie automobile, machines-outils, bois, textile
- ◆ Automatisation et régularisation : ex. suivi de production, gestion climatique
- ◆ Robotique : ex. manipulateur, robot de montage
- ◆ Contrôle qualité : ex. banc de test
- ◆ Scientifique, recherche
- ◆ Supervision : ex. surveillance, visualisation de processus

### 1.3 Caractéristiques d'utilisation

La carte d'extension PI 12 est un matériel compatible PC, pour bus PCI.

Pour les fonctions d'entrées/sorties, le décodage de l'adresse s'effectue dans l'espace I/O du PC. L'adresse de base et l'interruption sont directement déterminées par le système par la fonction Plug&Play.

La carte PI 12 permet de répondre, avec les modules suivants, à de nombreuses applications industrielles en ne nécessitant qu'un slot dans le PC. Le choix et la combinaison des différents types de modules sont totalement libres.

- Réf. **RA 01** : 8 entrées analogiques 12 bit, 0-5V,  $\pm 10V$ , 0-20 mA
- Réf. **RA 02** : 8 entrées analogiques 12 bit,  $\pm 5V$ , 0-2.5V
- Réf. **RA 03** : 2 sorties analogiques 12 bit, 0-10V,  $\pm 10V$
- Réf. **RA 05** : 1, 2 ou 4 sorties analogiques 12 bit, 0-10V,  $\pm 10V$ ,  $\pm 5V$
- Réf. **RD 01** : 8 entrées digitales optoisolées 24V AC ou DC, masse ou +24V commun
- Réf. **RD 02** : 8 sorties relais statiques 0-50V AC ou DC, 100 mA
- Réf. **RD 03** : 8 sorties relais statiques 0-60V DC, 100 mA, type NPN
- Réf. **RD 04** : 8 sorties relais statiques 0-60V DC, 100 mA, type PNP
- Réf. **RC 01** : 1 voie de comptage 32 bit pour codeur incrémental, entrée différentielle ou TTL
- Réf. **RC 02** : 1 voie de comptage pour codeur absolu, interface SSI
- Réf. **RC 03** : 1 voie de comptage 32 bit pour codeur incrémental, entrée différentielle ou TTL optoisolées
- Réf. **RT 01** : 1 port série RS 232 pour communication asynchrone

## 1.4 Caractéristiques physiques

**Alimentation** : 5V 150mA typique,  $\pm 12V$  pour l'utilisation avec certains modules

**Dimensions** : 250 x 99 mm.

**Température de fonctionnement** : 0 à +50°C

**Température de stockage** : -25 à +85°C

**Humidité relative** : 90% maximum, sans condensation

**Connecteurs** : connecteur MiniSub-D 50 points femelle, de type SCSI

**Protections** : Afin de palier aux perturbations électromagnétiques et de se conformer aux directives de la Communauté Européenne, toutes les lignes disponibles sur les connecteurs sont protégées par des filtres EMI. Ces filtres permettent d'agir directement sur les signaux d'entrées et de sorties contre les interférences électromagnétiques en protégeant ainsi efficacement la carte.

## 2. Démarrage rapide



**ATTENTION.** Pour la mise en service rapide de votre carte, reportez-vous impérativement aux chapitres suivants:

- ⇒ 4. Configuration
- ⇒ Manuels d'utilisation des modules

## 3. Installation

Pour installer la carte d'extension, procédez comme suit :

### 3.1 Ouverture de l'ordinateur



#### IMPORTANT

- Éteignez l'ordinateur.
- Débranchez le cordon d'alimentation secteur.

- ◆ Placez le PC sur une surface plane, stable et non encombrée.
- ◆ Enlevez le capot de l'ordinateur.

### 3.2 Mise en place de la carte

- ◆ Retirez la plaque de protection de l'un des emplacements libres du BUS PCI (Voir figure 1) .

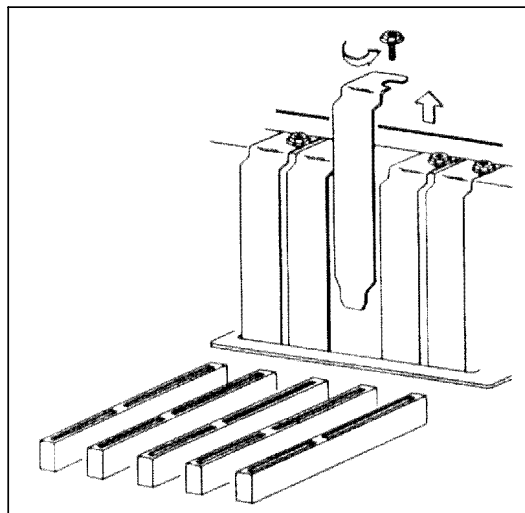


Figure 1





**ATTENTION** Les décharges d'électricité statique peuvent endommager les composants électroniques.

- ◆ Respectez les quelques règles de prudence suivantes:
  - Transportez et conservez la carte dans son emballage antistatique.
  - Ne touchez pas les composants électroniques.
  - Manipulez la carte par les bords et son équerre métallique de fixation.
- ◆ Insérez la carte dans l'emplacement choisi. Veillez à bien l'aligner sur le connecteur PCI avant de l'enfoncer (Voir figure 2) .

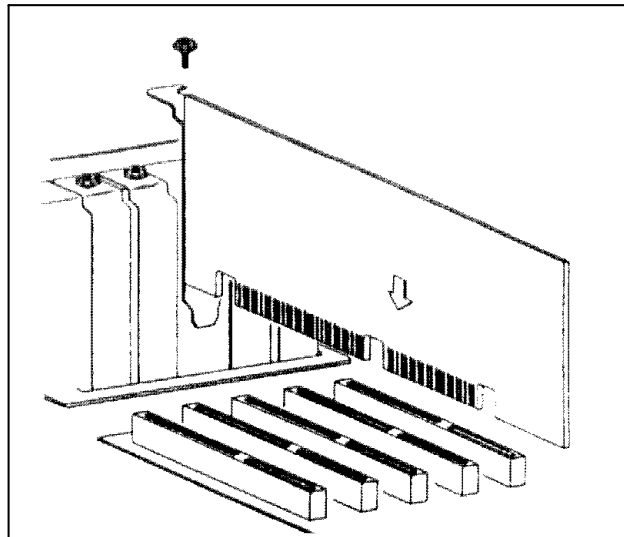


Figure 2

- ◆ Enfoncez la carte complètement en appuyant sans forcer.
- ◆ Remettez en place la vis de fixation.
- ◆ Remontez l'ordinateur. La carte est maintenant prête à l'emploi.

## 4. Configuration

Les cartes de base et les modules répondent au standard Industrial Modul Bus<sup>®</sup> de SELIAtec. Ceci garantit une parfaite compatibilité pour la connectique, les niveaux électriques et les dimensions de tous les produits de la famille : CF04 /CF06 /PI01 /PI12/ PI12 /PF01/ et des modules associés.

### 4.1. Adresse de base

L'adresse de base de la carte PI 12 est directement choisie par le système. Aucune configuration hardware n'est nécessaire.

L'échange de données pour chaque module, nécessite un certain nombre d'octets. Cet espace mémoire nécessaire est lié à l'adresse de base de la carte PI 12 et à l'emplacement choisi (A, B, C et D).

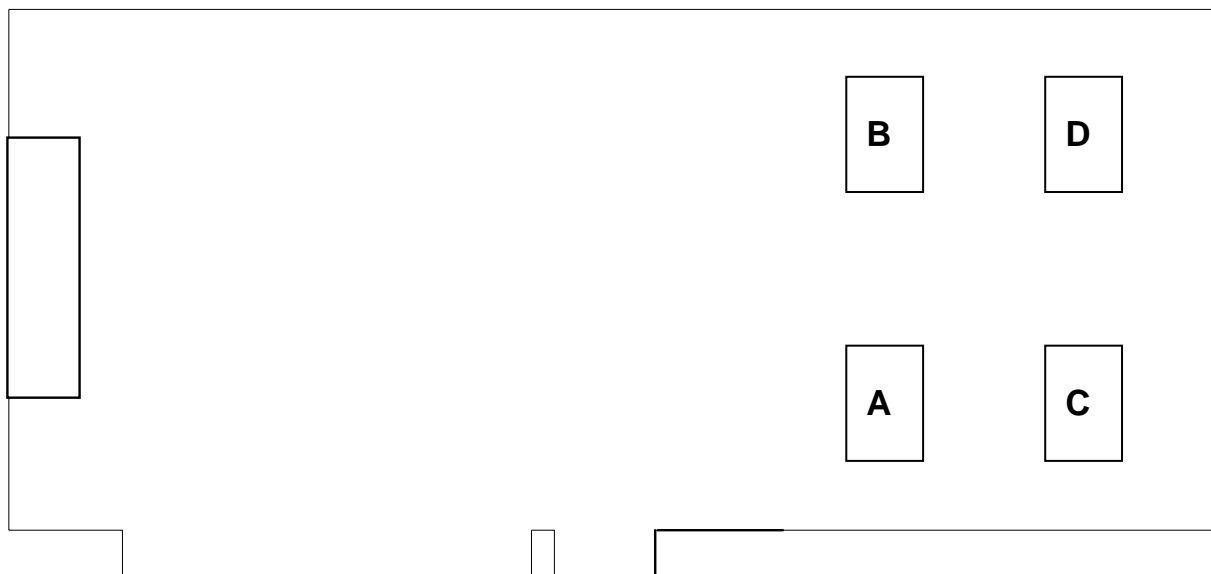


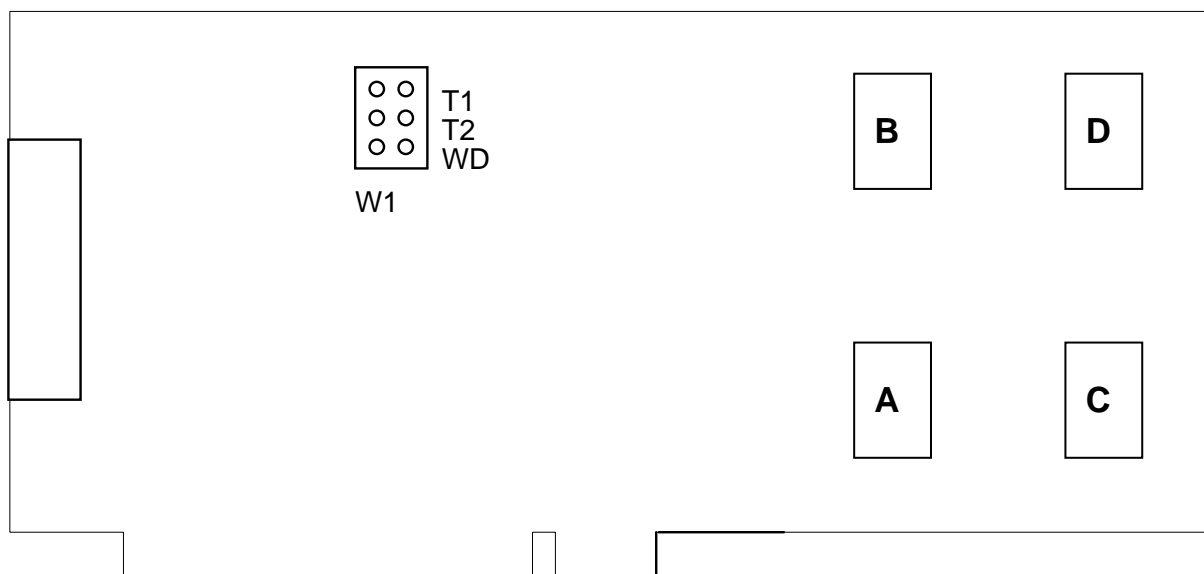
Schéma d'implantation de la carte PI 12

#### **NOTE**

#### **Risque d'incompatibilité**

Le système "Plug&Play" permet d'éviter les risques d'incompatibilité entre les différentes cartes présentes dans le PC.

## 4.2. Fonction chien de garde (watchdog)



La carte PI 12 est équipée d'une fonction chien de garde (Watchdog). Le watchdog est constitué d'une bascule monostable avec une constante de temps de base d'env. 5 ms ou bien définie par l'utilisateur.

Cette bascule monostable est réarmée chaque fois que la carte PI 12 est adressée en lecture (RD) ou en écriture (WR). Si la fonction watchdog est configurée et si la carte PI 12 n'est plus adressée pendant la durée de la constante de temps, le signal "watchdog" est transmis sur le ou les modules connectés. Ce signal aura le même effet sur la fonctionnalité des modules qu'un RESET à la mise sous tension du PC, c-à-d que les sorties digitales ou analogiques seront désactivées.

La fonction Watchdog et les constantes de temps sont sélectionnées en positionnant des cavaliers sur le bloc W1 indiqué ci-dessus.

### Correspondance des cavaliers du bloc SW1:

Cavalier positionné en WD, constante de temps = env. 5 ms

Cavaliers positionnés en WD et T1, constante de temps = env. 100 ms

Cavaliers positionnés en WD et T2, constante de temps = env. 5 s

## 4.3. Définition des adresses

Le système "Plug&Play" va déterminer l'adresse de base de la carte à partir de laquelle l'espace nécessaire aux différents modules sera réservé et défini comme ci-dessous.

## Fonctions I/O

Adresses	Lecture (RD)	Ecriture (WR)
Base + 0 à Base + 7	Fonctions I/O du MODULE A	
Base + 8 à Base + Fh	Fonctions I/O du MODULE B	
Base + 10 à Base + 17	Fonctions I/O du MODULE C	
Base + 18 à Base + 1Fh	Fonctions I/O du MODULE D	

De plus amples détails sont donnés dans les chapitres respectifs aux modules.

## 4.4. Interruptions

La sélection de l'interruption est déterminée automatiquement par le système "Plug&Play".

Certains modules montés sur la carte PI 12 émettent des interruptions pour un statut déterminé. Ces interruptions sont regroupées sur une ligne commune.

La carte PI 12 est équipée d'un système de mémorisation des interruptions provenant des différents modules. La ligne d'interruption de chacun des deux modules restera active tant que le module concerné n'aura pas été adressé en lecture (RD), voir le tableau ci-dessous.

Uniquement, à l'issue de cette action, une autre interruption pourra être traitée.

## Lecture des registres d'interruptions

Adresses	Lecture (RD)
Base + 0x40	Interruption MODULE A
Base + 0x48	Interruption MODULE B
Base + 0x50	Interruption MODULE C
Base + 0x58	Interruption MODULE D

## 4.5. Concept Boot&Work

Grâce au concept d'autoconfiguration, nommé BOOT&WORK<sup>®</sup>, les modules implantés sur la carte PI 12 sont directement identifiés.

L'utilisation des adresses décrites dans le tableau ci-dessous permet de détecter le type de module enfiché à chaque emplacement (A, B, C ou D) avec toutes les caractéristiques qui lui correspondent.

### Fonctions BOOT&WORK<sup>®</sup>

Adresses	Lecture (RD) / Ecriture (WR)
Base + 0x20 Base + 0x60	Activation ligne I2C Autoconfiguration du MODULE A
Base + 0x28 Base + 0x68	Activation ligne I2C Autoconfiguration du MODULE B
Base + 0x30 Base + 0x70	Activation ligne I2C Autoconfiguration du MODULE C
Base + 0x38 Base + 0x78	Activation ligne I2C Autoconfiguration du MODULE D

## 4.6. Connectique

La connectique de la carte PI 12 est donnée dans les manuels d'utilisation des différents modules à implanter :

**- Modules analogiques**

RA 01 – RA 02 – RA 03 – RA 05

**- Modules numériques**

RD 01 – RD 02 – RD 03 – RD 04

**- Modules de comptage**

RC 01 - RC02 - RC03

**- Module de communication série**

RT 01

Dans le cas contraire, se reporter au câblage de la carte PI 01. La carte PI 12 est compatible à 100% avec les câblages de la PI 01.

## 5. Installation de la DLL

### 5.1 Installation du pilote de la carte PI 12

Installer le pilote de communication en procédant comme décrit ci-dessous :

- Fermer toutes vos applications en cours.
- Lancer le programme Install.exe fourni sur le CD-ROM d'installation.
- Suivre les instructions données à l'écran.

Arrêter votre PC et installer la carte dans un Slot PCI.

- Redémarrer le PC lorsque le programme d'installation vous le suggère.

Si l'installation n'a rencontré aucun problème, lors du démarrage de l'ordinateur, votre système doit détecter un nouveau périphérique. Si l'ordinateur vous le demande, sélectionner le pilote PI12 de SELIAtec.

### 5.2 Descriptif des fichiers installés

L'installation a créé un dossier sous *C:\Program Files\SELIAtec\PI12* (à moins que vous n'ayez modifié le chemin à l'installation). Sous ce dossier, vous trouverez :

- Un exécutable *Config.exe* qui vous permet de vérifier que la carte est reconnue, que les modules détectés correspondent à ceux souhaités et de les tester.
- Un fichier *PI12.HLP* contenant de l'aide sur le programme de test *Config.exe*.
- Un fichier *VB.txt* comprenant des exemples d'appels des fonctions sous VB 6.0.
- Un exécutable *Wdreg.exe* permettant d'enregistrer *PI12.inf* dans la base de registre de votre PC. Cet exécutable est appelé à la fin de l'installation, avant que vous ne redémarriez votre PC. Vous ne devriez pas avoir à vous en soucier si vous ne rencontrez pas de problème d'installation.
- Un répertoire **Sources** comprenant des programmes source (en langage C)

RA01 (et 02).

RA03 (et 04).

RA05

RC01.

RC02.

RC03.

RD01.

RD02.

RD03.

RD04.

- Un répertoire **Headers** comprenant tous les fichiers.h nécessaires au développement de vos applications.  
PI12.h : nécessaire à tous les modules.  
PI12PI01.h : pour les anciens projets à migrer de PI01 vers PI12.  
Models.h : nécessaire au fonctionnement du *BOOT&WORK*.
- Un répertoire **Librairies** comprenant les librairies de fonctions développées par SELIAtec. Celui-ci contient :
  - les fichiers PI12.lib et PI12.dll en Release (version ne comprenant que le strict nécessaire à l'exécution). Ces fichiers sont suffisants pour la quasi-totalité des applications.

### 5.3 Vérification de l'installation

Afin de vérifier que le driver fonctionne correctement et pour vous faciliter vos tests, l'installation va placer un exécutable directement sous le menu :  
*Démarrer\Programmes\SELIAtec\PI12*.

Lancer le programme *Config.exe*. Un écran doit apparaître vous indiquant la carte détectée et les modules présents sur les différents Slots.  
En cas d'échec de ce programme (un écran noir apparaît et disparaît aussitôt) se reporter à la section Problème à l'installation.

Si la détection des modules s'est correctement déroulée, vous pouvez commencer à tester les modules. Pour cela, effectuer les branchements adéquats et tester au travers du programme *Config.exe*.  
(*Démarrer\Programmes\SELIAtec\PI12\*).

Ce programme de test est très simple et vérifie les fonctions de base des modules. Effectuez les tests en choisissant les options fournies par les différentes interfaces. En cas de problème à l'appel de ces programmes, reportez-vous à la section problème à l'appel de programme.

### 5.4 Désinstallation

**IMPORTANT** : AVANT DE DESINSTALLER, ASSUREZ-VOUS DES POINTS SUIVANTS : EN CAS DE NON RESPECT DE CES REGLES, VOUS POURRIEZ NE PLUS POUVOIR REDEMARRER VOTRE ORDINATEUR.

- Arrêter le PC et retirer la carte PI12.
- Vérifiez que vous n'avez pas de fichiers qui vous sont propres sous *C:\Program Files\SELIAtec\PI12*.
- Exécutez le programme de désinstallation des pilotes de gestion des cartes PI12.

## 5.5.Problème à l'installation

La première chose à faire est de vérifier que tous les fichiers nécessaires à la communication avec le matériel via votre système d'exploitation sont bien installés.

### **Windows 95/98 :**

Allez dans le menu *Démarrer\Rechercher\Fichiers ou Dossiers* et tapez PI12.inf. Il doit se situer sous *C:\Windows\Inf*. De la même manière, PI12.dll doit être sous *C:\Windows\System*.

Faites de même pour vérifier que le fichier windrvr.vxd se trouve sous *C:\Windows\System\MM32*. Si ces deux fichiers sont présents mais que la carte n'est toujours pas reconnue, allez dans le menu *Démarrer\Exécuter* et tapez la commande *C:\Program Files\SELIAtec\PI12\WDREG16 -vxd INSTALL* qui va permettre de charger les fichiers de reconnaissance de la carte dans le registre de démarrage.

Redémarrez votre ordinateur, si la carte n'est toujours pas reconnue, le problème n'est pas dû au driver mais au matériel.

### **Windows NT/2000 :**

Allez dans le menu *Démarrer\Rechercher\Fichiers ou Dossiers* et tapez PI12.inf. Il doit se situer sous *C:\WinNT\Inf*. De la même manière, PI12.dll doit être sous *C:\WinNT\System32*.

Faites de même pour vérifier que le fichier windrvr6.sys se trouve sous *C:\WinNT\System32*. Si ces 2 fichiers sont présents mais que la carte n'est toujours pas reconnue, allez dans le menu *Démarrer\Exécuter* et tapez la commande *C:\Program Files\SELIAtec\PI12\WDREG -inf C:\WinNT\Inf\PI12.inf INSTALL* qui va permettre de charger les fichiers de reconnaissance de la carte dans le registre de démarrage. Pour WinNT 4 tapez simplement *C:\Program Files\SELIAtec\PI12\WDREG INSTALL*.

Redémarrez votre ordinateur, si la carte n'est toujours pas reconnue, le problème n'est pas dû au driver mais au matériel.

### **Windows XP :**

Allez dans le menu *Démarrer\Rechercher\Fichiers ou Dossiers* et taper PI12.inf. Il doit se situer sous *C:\Windows\Inf*. De la même manière, PI12.dll doit être sous *C:\Windows\System32*.

Faites de même pour vérifier que le fichier windrvr6.sys se trouve sous *C:\Windows\System32*. Si ces 2 fichiers sont présents mais que la carte n'est toujours pas reconnue, allez dans le menu *Démarrer\Exécuter* et tapez la commande *C:\Program Files\SELIAtec\PI12\WDREG -inf C:\Windows\Inf\PI12.inf INSTALL* qui va permettre de charger les fichiers de reconnaissance de la carte dans le registre de démarrage.

Redémarrez votre ordinateur, si la carte n'est toujours pas reconnue, le problème n'est pas dû au driver mais au matériel.



### 5.5.1 Problème à l'appel de programme :

Tout d'abord, il faut identifier le type de problème. Si un message apparaît vous stipulant que vous avez un problème d'initialisation de la DLL, reportez-vous à la section Problème à l'installation.

### 5.5.2 Problèmes à l'appel d'un exécutable :

- Si vous avez un message stipulant qu'une DLL est manquante, vérifiez que vous avez bien le fichier P112.dll sous *C:\Windows\System* pour Windows 95 et 98 ou *C:\WinNT\System32* pour Windows NT/2000 et XP. Le fichier installé automatiquement est la version Release.
- Si le programme semble fonctionner mais que vous sortez tout de suite, c'est que vous avez un problème de réservation de ressources. Ceci peut-être provoqué par une mauvaise exécution d'un programme. Dans ce cas, redémarrez votre ordinateur et réessayez. Les ressources sont alors libérées et l'exécution doit s'effectuer sans problème. Dans le cas contraire, vérifiez les ressources allouées à votre carte dans le *panneau de configuration* -> *système* -> *ressources matérielles*. Faites une liste des ressources allouées à tous vos périphériques et vérifiez que ces ressources ne se chevauchent pas.

### 5.5.3 Problème à la compilation d'une application

Vérifiez que vous avez bien sous le répertoire courant (ou lié à votre projet) les fichiers nécessaires à votre application. Vous devez obligatoirement avoir, quel que soit votre module, les fichiers P112.lib et P112.h pour les compilateurs C (vous pouvez aussi placer le fichier P112.dll mais il est plus propre de l'avoir sous *C:\Windows\System* pour Win95/98, *C:\WinNT\System32* pour Win NT/2000 ou *C:\Windows\System32* pour Win XP).

Le fichier P112.lib doit être mis en Ressources et les différents fichiers.h en Header. Reportez-vous au descriptif des fichiers installés pour connaître les fichiers à lier à votre application.

## 5.6. Généralités sur l'utilisation de la librairie de fonctions

Cette librairie permet une utilisation simplifiée des différents modules pouvant être insérés sur une carte multi-fonctions PI12 dans les environnements Windows 95/98, et Windows NT4/2000/XP.

**Avant d'utiliser une carte PI12, il est nécessaire d'appeler les fonctions :**

- **"PI12\_get\_card\_pos (OUT ULONG \*BusNum, OUT ULONG \*SlotNum, OUT ULONG \*CardsNum)"**  
pour récupérer tous les paramètres propres à la carte PI12
- 
- **"PI12\_define\_card"**  
pour identifier automatiquement les différents modules présents grâce au concept *BOOT&WORK*®.

Chacun des modules pouvant être inséré sur une carte PI12 possède une mémoire permanente contenant des informations telles que le type, le modèle, la date de programmation, le numéro de version et un texte décrivant les fonctions du module. La fonction « PI12\_define\_card » réalise l'identification successive des modules insérés sur une carte PI12 et la réservation des ressources utilisées (plage d'adresse d'entrées/sorties et numéro d'interruption de la carte et commun à tous les modules de la carte PI12).

Après son appel, les fonctions "PI12\*\_is" permettent la récupération des diverses informations sur les modules présents ainsi que les ressources les concernant.

Une fois les différents modules reconnus, ils sont identifiés par le numéro de la carte et le numéro de l'emplacement, ainsi lorsqu'une fonction spécifique d'un module est appelée, celle-ci vérifie la présence effective d'un tel module à cet emplacement.

**À la fin de l'utilisation d'une carte PI12, il est indispensable d'appeler la fonction :**

**"PI12\_undefine\_card"**

afin de libérer les ressources réservées et de permettre à un autre logiciel de réserver la carte. Une libération est automatiquement faite dans le déchargement de la DLL mais le résultat n'est pas garanti.

En fonction des modules présents sur votre carte PI12, exécutez les fonctions

- "PI12\*\_entrees" ou "PI12\*\_sorties" pour lire ou écrire des valeurs analogiques.
- "PI12\*\_raw\_entrees" ou "PI12\*\_raw\_sorties" pour lire ou écrire les valeurs correspondantes en binaires.

Les fonctions de bas niveaux ci-dessous permettent l'accès direct aux ports d'entrées/sorties des différents modules.

- "PI12\_raw\_outportb"
- "PI12\_raw\_inportb"

Toutes les fonctions renvoient une valeur booléenne indiquant le succès de l'opération demandée.

En cas d'erreur la fonction "PI12\_error" indique le code de la dernière erreur détectée selon la structure PI12-Error-Codes définie dans le fichier PI12.h.

La librairie "PI12" est utilisable dans tout environnement de programmation supportant les librairies dynamiques. Toutefois, il peut s'avérer nécessaire de déclarer les fonctions de la librairie avant leur utilisation. Ainsi sous Visual Basic, les différentes fonctions doivent être déclarées (voir l'instruction DECLARE dans votre manuel de programmation de Visual Basic).

**Remarque :**

Pour l'utilisation d'un module RT 01, veuillez configurer le driver standard du port série de votre système. Lancez l'outil de définition d'un nouveau port série, indiquez son adresse de base en fonction de son emplacement sur la carte PI12.

Adresse du port série = adresse de base + (8 x le numéro de l'emplacement)

Le numéro de l'emplacement est 0 pour A, 1 pour B, 2 pour C et enfin 3 pour D.

## 6. Fonctions relatives à la carte PI 12

enum PI12\_Error\_Codes

### Liste des codes erreurs des différentes fonctions

aucune erreur

PI12\_ERROR\_NONE = 0UL

les paramètres sont incorrects

PI12\_ERROR\_INVALID\_PARAMETERS = 0xE0550000UL

la plage mémoire n'est pas libre

PI12\_ERROR\_USED\_MEMPORTS

l'interruption n'est pas libre

PI12\_ERROR\_USED\_INTERRUPT

la ressource n'est pas verrouillée

PI12\_ERROR\_NOT\_LOCKED

le nombre de verrous maximum a été atteint

PI12\_ERROR\_TOO\_MANY\_LOCKS

le descripteur est invalide

PI12\_ERROR\_INVALID\_HANDLE

pas de carte présente

PI12\_NO\_BOARD

les ressources demandées sont déjà utilisées

PI12\_ERROR\_USED\_RESOURCES

cette fonctionnalité n'est pas supporté dans cette version

PI12\_ERROR\_UNSUPPORTED

le numéro de carte est invalide

PI12\_ERROR\_BAD\_CARD\_NUMBER

l'emplacement ne contient pas le module en question

PI12\_ERROR\_BAD\_MODULE

le numéro d'emplacement est invalide

PI12\_ERROR\_BAD\_SLOT\_NUMBER

l'offset demandé est trop important

PI12\_ERROR\_BAD\_PORT\_OFFSET

la valeur n'est pas dans la plage autorisée

PI12\_ERROR\_OUT\_OF\_BOUNDS

la valeur lue est incorrecte

PI12\_ERROR\_BAD\_VALUE

le contenu de l'eprom du module est invalide

PI12\_ERROR\_BAD\_EEPROM,

l'accès à l'I2C est impossible

PI12\_ERROR\_BAD\_EEPROM\_I2C,

dernière erreur

PI12\_ERROR\_LAST

Toutes les fonctions de la librairie renvoient une valeur booléenne indiquant le succès de l'opération. Si une erreur s'est produite la fonction PI12\_error() renvoie le code de cette erreur.

**enum MODULE\_Types**

Liste des différents types de modules

MODULE\_TYPE\_NONE

aucun module

MODULE\_TYPE\_RA01

8 entrées analogiques 12 bit, tension +/-10V ou 0-5V, courant 0-20mA

MODULE\_TYPE\_RA02

8 entrées analogiques 12 bit, en tension +/-5V ou 0-2,5V

MODULE\_TYPE\_RA03

2 sorties analogiques 12 bit, en tension +/-10V ou 0-10V

MODULE\_TYPE\_RA04

4 sorties analogiques 12 bit, en tension +/-10V

MODULE\_TYPE\_RA05

1, 2 ou 4 sorties analogiques 12 bit, en tension 0-10V, +/-10V, +/-5V

MODULE\_TYPE\_RD01

8 entrées digitales optoisolées 24V AC ou DC, masse ou +24V commun

MODULE\_TYPE\_RD02

8 sorties relais statiques AC, contacts ouverts au repos

MODULE\_TYPE\_RD03

8 sorties relais statiques DC type PNP, contacts ouverts au repos

MODULE\_TYPE\_RD04

8 sorties relais statiques DC type NPN, contacts ouverts au repos

MODULE\_TYPE\_RC01

1 voie de comptage pour codeur incrémental, 32 bit, E. diff ou TTL

MODULE\_TYPE\_RC02

1 voie de comptage pour codeur absolu, SSI

MODULE\_TYPE\_RC03

1 voie de comptage pour codeur incrémental, 32 bit, entrées 5V ou 24V opto.

MODULE\_TYPE\_RT01

1 voie de communication série RS 232

**ULONG PI12\_errors (void)**

Cette fonction renvoie le code de la dernière erreur survenue. Elle doit être appelée lorsqu'une des fonctions PI12 a échoué afin de déterminer la cause de l'échec.

Retour: valeur de la dernière erreur (Voir aussi PI12\_ErrorCodes)

**BOOL PI12\_get\_card\_pos(OUT ULONG \*BusNum, OUT ULONG \*SlotNum, OUT ULONG \*CardsNum)**

Cette fonction retourne tous les paramètres propres à la carte.

Paramètre : **BusNum** - numéro du bus alloué à la carte.

**SlotNum** - numéro du slot alloué à la carte.

**CardsNum** - numéro de la carte.

Retour: TRUE en cas de succès de l'opération

**BOOL PI12\_define\_card(IN ULONG bus\_num, IN ULONG slot\_num, IN PI12\_INT\_HANDLER func, OUT ULONG \* carte)**

Cette fonction doit être appelée afin d'utiliser une carte PI12 et plus particulièrement les modules installés dans les 4 emplacements.

Les différents modules sont détectés automatiquement par le mécanisme du BOOT&WORK et des informations les concernant peuvent être récupérées par l'intermédiaire des fonctions "PI12\_\*\_is".

Utilisez la fonction "PI12\_undefine\_card" pour libérer la carte.

Paramètre: **bus\_num** - numéro du bus de la carte. Celui-ci peut être récupéré par la fonction adresse de base de la carte PI12\_get\_card\_pos.

**slot\_num** - numéro du slot de la carte. Celui-ci peut être récupéré par la fonction PI12\_get\_card\_pos.

**PI12\_INT\_HANDLER** - fonction d'interruption qui doit initialiser la structure PI12\_INT\_PARAM composée des arguments carte, value, intreg, slot. (Voir PI12.h pour la structure).

**carte** - Un numéro de carte qui est en fait le numéro de handle réservé dans les ressources de windows(ce qui correspond en fait à une « image » de vos sorties physiques en mémoire). Ce numéro de carte doit être repassé à la fonction PI12\_undefine\_card (vous pouvez considérer que le numéro de carte est l'adresse de base de votre carte).

Retour: TRUE en cas de succès de l'opération

**BOOL PI12\_undefine\_card (IN ULONG card)**

Cette fonction doit être appelée à la fin de l'utilisation de la carte. Cette fonction libère notamment les différentes ressources utilisées par la carte.

Paramètre:       **card** - le descripteur de carte renvoyé par la fonction "PI12\_define\_card"

Retour:       TRUE en cas de succès de l'opération  
Voir aussi   PI12\_ErrorCodes et PI12\_define\_card

**BOOL PI12\_type\_is (IN ULONG carte, IN ULONG slot, OUT ULONG \*type)**

Cette fonction indique le type de module présent sur une carte à un emplacement donné.

Paramètre:       **carte** - le descripteur de carte renvoyé par la fonction "PI12\_define\_card"  
                  **slot** - le numéro d'emplacement  
                  **type** - du module présent à l'emplacement donné

Retour:               TRUE en cas de succès de l'opération  
Voir aussi           La structure MODULE\_Types dans le fichier Models.h

**BOOL PI12\_model\_is (IN ULONG carte, IN ULONG slot, OUT ULONG \*modele)**

Cette fonction indique le modèle de module présent sur une carte à un emplacement donné.

Paramètre:       **carte** - le descripteur de carte renvoyé par la fonction "PI12\_define\_card"  
                  **slot** - le numéro d'emplacement  
                  **modele** - du module présent à l'emplacement donné

Retour:               TRUE en cas de succès de l'opération  
Voir aussi           La structure MODULE\_Types dans le fichier Models.h



**BOOL PI12\_version\_is** (*IN* ULONG carte, *IN* ULONG slot, *OUT* ULONG \* version)

Cette fonction indique le numéro de version du module présent sur une carte à un emplacement donné.

Paramètre:       **carte** - le descripteur de carte renvoyé par la fonction "PI12\_define\_card"  
                  **slot** - le numéro d'emplacement  
                  **version** - du module présent à l'emplacement donné

Retour:            TRUE en cas de succès de l'opération

**BOOL PI12\_name\_is** (*IN* ULONG carte, *IN* ULONG slot, *OUT* char \*nom, *IN* int len)

Cette fonction indique le nom du module présent sur une carte à un emplacement donné.

Paramètre:       **carte** - le descripteur de carte renvoyé par la fonction "PI12\_define\_card"  
                  **slot** - le numéro d'emplacement  
                  **nom** - du module présent à l'emplacement donné  
                  **len** - taille du buffer de caractères "nom"

Retour:            TRUE en cas de succès de l'opération

**BOOL PI12\_date\_is** (*IN* ULONG carte, *IN* ULONG slot, *OUT* char \*date, *IN* int len)

Cette fonction indique la date de programmation du module présent sur une carte à un emplacement donné.

Paramètre:       **carte** - le descripteur de carte renvoyé par la fonction "PI12\_define\_card"  
                  **slot** - le numéro d'emplacement  
                  **date** - de programmation du module présent à l'emplacement donné  
                  **len** - taille du buffer de caractères "date"

Retour:            TRUE en cas de succès de l'opération

**BOOL PI12\_description\_is** (*IN* ULONG *carte*, *IN* ULONG *slot*, *OUT* char *\*description*, *IN* int *len*)

Cette fonction renvoie une description des caractéristiques du module présent sur une carte à un emplacement donné.

Paramètre:       **carte** - le descripteur de carte renvoyé par la fonction "PI12\_define\_card"  
                  **slot** - le numéro d'emplacement  
                  **description** - du module présent à l'emplacement donné  
                  **len** - taille du buffer de caractères "description"

Retour:       TRUE en cas de succès de l'opération

**BOOL PI12\_ports\_resources** (*IN* ULONG *carte*, *OUT* ULONG *\* debut*, *OUT* ULONG *\* fin*)

Cette fonction renvoie les ports d'entrées/sorties utilisés par une carte PI12.

Paramètre:       **carte** - le descripteur de carte renvoyé par la fonction PI12\_define\_card"  
                  **debut** - adresse de base de la carte  
                  **fin** - adresse de fin de la plage de ports d'entrées/sorties

Retour:       TRUE en cas de succès de l'opération

**BOOL PI12\_interrupt\_resources** (*IN* ULONG *carte*, *OUT* ULONG *\* interruptno*)

Cette fonction renvoie le numéro de l'interruption utilisée par une carte PI12.

Paramètre:       **carte** - le descripteur de carte renvoyé par la fonction "PI12\_define\_card"  
                  **interruptno** - numéro d'interruption utilisé par la carte, correspondant aux IRQ standards du PC.

Retour:       TRUE en cas de succès de l'opération

**BOOL PI12\_reg\_interruption** (*IN* ULONG *carte*, *IN* ULONG *slot*, *OUT* ULONG \* *interruptreg*)

Cette fonction renvoie le contenu du registre d'interruption d'un module d'une carte PI12 accessible à l'adresse 'base'+0x20+slot\_offset(0-1).

Paramètre:       **carte** - le descripteur de carte renvoyé par la fonction "PI12\_define\_card"  
                  **slot** - le numéro d'emplacement  
                  **interruptreg** - la valeur du registre d'interruption du module

Retour:        TRUE en cas de succès de l'opération

## 7. Fonctions relatives aux modules

Vous trouverez dans les documentations techniques des modules, les fonctions relatives à chaque type.

Toutes les fonctions de la librairie renvoient une valeur booléenne indiquant le succès de l'opération demandée. Tous les paramètres des fonctions sont passés en arguments, les préfixes IN et OUT permettent d'identifier respectivement les données en entrées et les variables modifiées par la fonction.

## 8. Fonctions de bas niveau

**BOOL PI12\_raw\_outportb** (*IN* ULONG *carte*, *IN* LONG *slot*, *IN* ULONG *offset*, *IN* BYTE *val\_8bits*)

Cette fonction de bas niveau permet d'écrire de façon brute une valeur 8 bits à n'importe quelle adresse.

Paramètre:       **carte** - le descripteur de carte renvoyé par la fonction "PI12\_define\_card"  
                  **slot** - numéro du slot sur lequel se trouve le module :  
                  1 pour A, 2 pour B, 3 pour C et 4 pour D.  
                  **offset** - décalage désiré depuis l'adresse du module entre 0 et 7.  
                  **Val\_8 bits** - valeur à envoyer à l'adresse voulue.

Retour:        TRUE en cas de succès de l'opération

**BOOL PI12\_raw\_inportb (IN ULONG carte, IN ULONG slot, IN ULONG offset, OUT BYTE \*val\_8bits)**

Cette fonction de bas niveau permet de lire de façon brute une valeur 8 bits à n'importe quelle adresse.

Paramètre:        **carte** - le descripteur de carte renvoyé par la fonction "PI12\_define\_card"  
                      **slot** - numéro du slot sur lequel se trouve le module :  
                          1 pour A, 2 pour B, 3 pour C et 4 pour D.  
                      **offset** - décalage désiré depuis l'adresse du module entre 0 et 7.  
                      **Val\_8 bits** - valeur à envoyer à l'adresse voulue.

Retour:        TRUE en cas de succès de l'opération

La fonction d'interruption vous permet de récupérer une structure PI12\_INT\_PARAM comprenant les arguments :

**ULONG carte** = numéro de la carte ayant émis une interruption.

**ULONG slot** = numéro du slot ayant émis une interruption (1, 2, 3 ou 4).

**ULONG value** = valeur lue sur le module (32 bit pour les compteurs, 8 bit pour les modules digitaux, 12 bits pour les modules analogiques).

**ULONG intreg** = valeur du registre d'interruption à base + 0x40.

## 9. Exemple en Visual Basic 6.0

```
Private Type Tab1
tabbus(0 To 10) As Long
End Type
Private Type Tab2
slotbus(0 To 10) As Long
End Type

Private Declare Function PI12_define_card _
Lib "PI12.dll" (ByVal bus As Integer, ByVal slot As Integer, ByVal func As
Integer, handle As Integer) As Integer
Private Declare Function PI12_undefine_card _
Lib "PI12.dll" (ByVal handle As Long) As Integer
Private Declare Function PI12_get_card_pos _
Lib "PI12.dll" (bus As Tab1, slot As Tab2, cards As Long) As Integer
Private Declare Function PI12_errors _
Lib "PI12.dll" () As Long

Private Sub PI12error_Click()

Dim LongRetour As Integer
Dim handle As Integer
Dim bus As Tab1
Dim slot As Tab2
Dim cards As Long
Dim tabbus(0 To 10) As Long
Dim tabslot(0 To 10) As Long
Print
cards = 4

LongRetour = PI12_get_card_pos(bus, slot, cards)

Print "bus"; bus.tabbus(0)
Print "slot"; slot.slotbus(0)
Print "cards"; cards
Print "retour"; LongRetour
m_bus = bus.tabbus(0)
m_slot = slot.slotbus(0)

LongRetour = PI12_define_card(m_bus, m_slot, 0, handle)

Print "valeur de retour du handle :"; handle
LongRetour = PI12_undefine_card(handle)
Print "valeur retournee par undefine :"; LongRetour
```

End Sub