



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 : support@seliatec.com

PF 01

Carte multifonctions pour bus PC/104
Entrées et sorties industrielles
analogiques, numériques, comptage,
commande d'axes, communication

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 REPONDENT 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

1. <u>Introduction</u>	5
1.1 Contenu de l'emballage	5
1.2 Domaines d'applications.....	5
1.3 Caractéristiques d'utilisation	6
1.4 Caractéristiques physiques	7
2. <u>Démarrage rapide</u>	9
3. <u>Installation</u>	9
4. <u>Configuration</u>	10
4.1. Adresse de base.....	10
4.2. Interruptions.....	12
4.3. Fonction Chien de garde (Watchdog).....	14
4.4. Définition des adresses	15
5. <u>Installation de la DLL</u>	16
5.1. Introduction	16
5.1.1 Installation du driver de la carte PF 01	16
5.1.2 Descriptif des fichiers installés.....	16
5.1.3 Vérification de l'installation	17
5.1.4 Désinstallation.....	18
5.1.5.Problème à l'installation.....	18
5.1.6 Problème à l'appel de programme :.....	18
5.1.7 Problèmes à l'appel d'un exécutable :	19
5.1.8 Problème à la compilation d'une application :	19
5.2. Généralités sur l'utilisation de la librairie de fonctions.....	19
6. <u>Définitions des différentes fonctions relatives à la carte PF01</u>	22
7. <u>Fonctions relatives aux modules</u>	29
8. <u>Fonctions de bas niveau</u>	29
8.1 Fonctions I/Oselia : accès aux ports d'entrées/sorties	30

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 dans son enveloppe antistatique.
- Le manuel d'utilisation.
- La disquette contenant les programmes d'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 PF 01 permet d'équiper votre carte CPU pour bus PC/104 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 PF 01 est un matériel compatible PC, bus PC/104.

Elle est peut être adressée librement par des micro-interrupteurs (0 à 03FFh).

Pour les fonctions d'entrées/sorties, la carte occupe 16 octets à partir de l'adresse de base. Le décodage de l'adresse s'effectue dans l'espace I/O de 64 Ko du PC.

L'adresse de base est également utilisée avec les adresses supérieures A14 et A15, soit 4000 à 43FFh, 8000 à 83FFh et C000 à C3FFh.

L'espace 8000 à 83FFh sert à la lecture des registres d'interruptions

L'espace 4000 à 43FFh sert à la lecture des types de modules enfichés.

L'espace C000 à C3FFh sert également à la lecture des types de modules.

Les interruptions disponibles :

- ◆ XT : IRQ3, IRQ 5, IRQ6, IRQ7
- ◆ AT : IRQ10, IRQ11, IRQ12, IRQ15

La carte PF 01 permet de répondre, avec les modules suivants, à de nombreuses applications industrielles. Le choix et la combinaison des différents types de modules est totalement libre.

La carte PF 01 peut être équipée de 2 modules, sachant que les modules RM 01 et RM 02 occupent deux emplacements.

- 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 04** : 4 sorties analogiques 12 bit, $\pm 10V$
- Réf. **RA 05** : 1, 2 ou 4 sorties analogiques 12 bit, 0-10V, $\pm 10V$ ou $\pm 5V$
- 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. **RM 01** : commande d'axes pour 1, 2 ou 4 moteurs servo
- Réf. **RM 02** : commande d'axes pour 1, 2 ou 4 moteurs pas-à-pas
- 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 : 90 x 96 mm, au standard PC/104

Température de fonctionnement : 0 à +50°C

Température de stockage : -25 à +85°C

Humidité relative : 90% maximum, sans condensation

Watchdog : temps de déclenchement 5s ou débrayable

Connecteurs : - version PF 01-A: 2connecteurs à borniers débrochables, 10 points
- version PF 01-B: 2connecteurs à de type HE10, 10 points, pour câbles en nappe

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.

Signaux du bus PC/104

Rangée B		Rangée A	
GND	B1	A1	/ IO-CH CK
RESET DRV	B2	A2	D7
+ 5V	B3	A3	D6
IRQ2	B4	A4	D5
- 5V	B5	A5	D4
DRQ2	B6	A6	D3
- 12V	B7	A7	D2
RESERVE	B8	A8	D1
+ 12V	B9	A9	D0
GND	B10	A10	IO-CH RDY
/ MEMW	B11	A11	AEN
/ MEMR	B12	A12	A19
/ IOWR	B13	A13	A18
/ IORD	B14	A14	A17
/ DACK3	B15	A15	A16
DRQ3	B16	A16	A15
/ DACK1	B17	A17	A14
DRQ1	B18	A18	A13
/ DACK0	B19	A19	A12
CLK	B20	A20	A11
IRQ7	B21	A21	A10
IRQ6	B22	A22	A9
IRQ5	B23	A23	A8
IRQ4	B24	A24	A7
IRQ3	B25	A25	A6
/ DACK2	B26	A26	A5
T/C	B27	A27	A4
ALE	B28	A28	A3
+ 5V	B29	A29	A2
OSC	B30	A30	A1
GND	B31	A31	A0
GND	B32	A32	GND
Rangée D		Rangée C	
GND	D0	C0	GND
/ MEM-CS16	D1	C1	/ SBHE
/ IO-CS16	D2	C2	A23
IRQ10	D3	C3	A22
IRQ11	D4	C4	A21
IRQ12	D5	C5	A20
IRQ15	D6	C6	A19
IRQ14	D7	C7	A18
/ DACK0	D8	C8	A17
DRQ0	D9	C9	/ MEMR
/ DACK5	D10	C10	/ MEMW
DRQ5	D11	C11	D8
/ DACK6	D12	C12	D9
DRQ6	D13	C13	D10
/ DACK7	D14	C14	D11
DRQ7	D15	C15	D12
+ 5V	D16	C16	D13
/ MASTER	D17	C17	D14
GND	D18	C18	D15
GND	D19	C19	GND

2. Démarrage rapide

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

 ⇒ 4. Configuration
⇒ Manuels d'utilisation pour les modules

3. Installation

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

IMPORTANT



- Éteignez l'alimentation du système PC/104.
- Débranchez le cordon d'alimentation secteur.

- ♦ Placez le système sur une surface plane, stable et non encombrée.

Remarque : La carte est prévue pour fonctionner sur un slot PC/104 AT (16 bit). Un slot XT (8-bit) convient également. Dans ce cas les interruptions étendues IRQ10, IRQ11, IRQ12, IRQ15 de l'AT ne sont plus utilisables.

 **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.
- ♦ Configurez la carte (Voir chapitre 4.Configuration) .

4. Configuration

La configuration d'une carte consiste à choisir l'adresse de base de celle-ci et de l'équiper avec les modules nécessaires à l'application.

L'adressage est expliqué dans le chapitre suivant.

Les cartes de base et les modules répondent au standard Industrial Modul Bus ® de SELIAtec. Ceci garantit une parfaite compatibilité pour la connectique, les niveaux électriques et les dimensions de tous les produits de la famille.

4.1. Adresse de base

La carte PF 01 occupe 2 fois 8 octets dans l'espace I/O de 64 Ko du PC.

Pour les fonctions d'entrées/sorties, la carte occupe 16 octets à partir de l'adresse de base.

Elle est peut être adressée librement par des micro-interrupteurs dans l'espace 0 à 03FFh.

Le bloc de micro-interrupteurs servant à la configuration de l'adresse est repéré par la dénomination SW1.

L'adresse de base est également utilisée avec les adresses supérieures A14 et A15, par duplication automatique, sans que l'utilisateur ait de configuration supplémentaire à effectuer. Les adresses sont donc miroitées dans les espaces 4000 à 43FFh, 8000 à 83FFh et C000 à C3FFh avec le même offset que l'adresse de base sélectionnée grâce aux micro-interrupteurs.

L'espace 8000 à 83FFh sert à la lecture des registres d'interruptions. L'utilisation est expliquée dans les chapitres respectifs des modules qui offrent les fonctions d'interruption.

Les espaces 4000 à 43FFh et C000 à C3FFh servent à la lecture des types de modules enfichés.

L'utilisation de ces adresses permet de détecter le type de module enfiché à chaque emplacement sur la carte PF 01, avec toutes les caractéristiques qui lui correspondent. Les logiciels fournis avec la carte permettent d'utiliser ce dispositif et de créer un fichier avec les drivers pour les modules utilisés.

Ce concept d'autofiguration est nommé BOOT&WORK ®.

Il est intégré à toutes les cartes de base et les modules de la famille de produits au standard Industrial Modul Bus ® de SELIAtec.

Pour plus d'informations, consultez les Annexes Programmation et les Manuels d'utilisation des modules

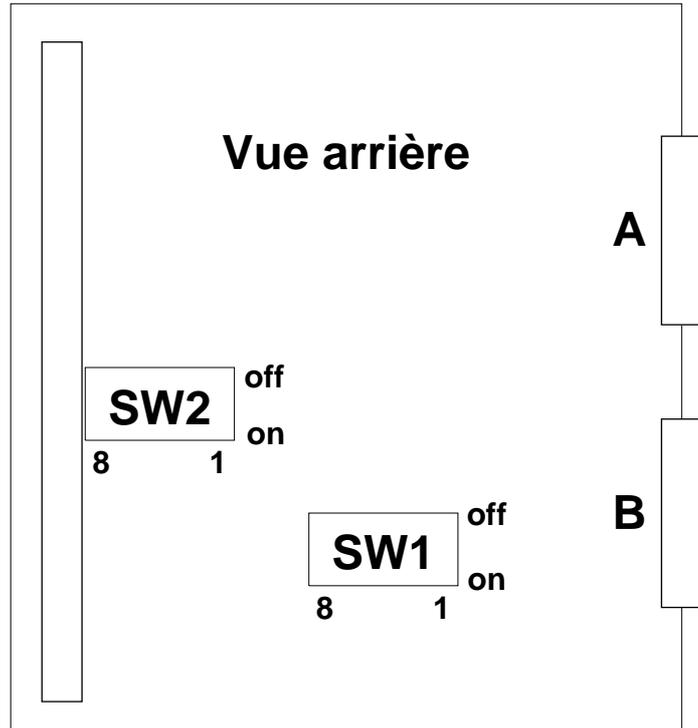


Schéma d'implantation de la carte PF 01

A la livraison, l'adresse de base est sélectionnée à 0100 HEX.

Tableau du décodage des bit d'adresse :

Positionnement des micro-interrupteurs du bloc SW1

Bit d'adresse	A9	A8	A7	A6	A5	A4
Micro-interrupteur	S6	S5	S4	S3	S2	S1
Etat logique	0	1	0	0	0	0

Dans l'exemple ci-dessus, l'adresse de base 100_{HEX} est décodée, conformément à la configuration lors de la livraison.

*Un état logique 0 correspond à la position ON des interrupteurs.
Un état logique 1 correspond à la position OFF des interrupteurs.*

NOTE

Le micro-switch S7 doit toujours rester sur OFF
Le micro-switch S8 du bloc SW1 n'est pas pris en compte pour la configuration de l'adresse de base. Il sert à la sélection de la fonction Watchdog.

Exemples d'adressage :

Bit d'adresse	A9	A8	A7	A6	A5	A4
Micro-interrupteur	S6	S5	S4	S3	S2	S1
Etat logique	1	1	1	0	0	1

Dans l'exemple ci-dessus, l'adresse de base 390_{HEX} est décodée.

Bit d'adresse	A9	A8	A7	A6	A5	A4
Micro-interrupteur	S6	S5	S4	S3	S2	S1
Etat logique	1	0	1	1	0	0

Dans l'exemple ci-dessus, l'adresse de base 2C0_{HEX} est décodée.

Bit d'adresse	A9	A8	A7	A6	A5	A4
Micro-interrupteur	S6	S5	S4	S3	S2	S1
Etat logique	1	1	0	0	0	0

Dans l'exemple ci-dessus, l'adresse de base 300_{HEX} est décodée.

4.2. Interruptions

Certains modules montés sur la carte PF 01 émettent des interruptions pour un statut déterminé. Ces interruptions sont regroupées sur une ligne commune.

En fonction de l'interruption PC souhaitée, positionnez le micro-interrupteur correspondant du bloc SW2 sur la position ON.



ATTENTION : un seul interrupteur doit être sur ON, tous les autres doivent être sur OFF. En cas de non-respect, la carte, les modules ou la carte CPU peuvent subir des dommages importants.

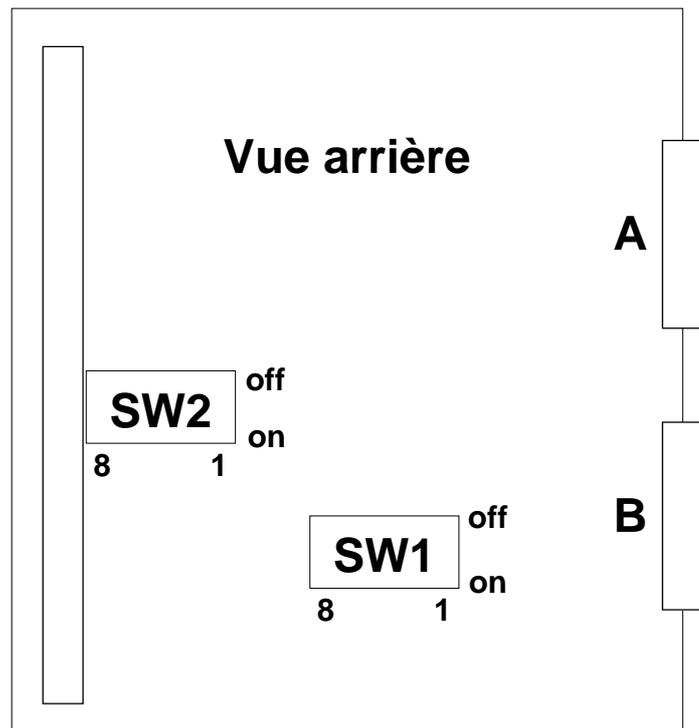


Schéma d'implantation de la carte PF 01

Correspondance des micro-interrupteurs do bloc SW2 :

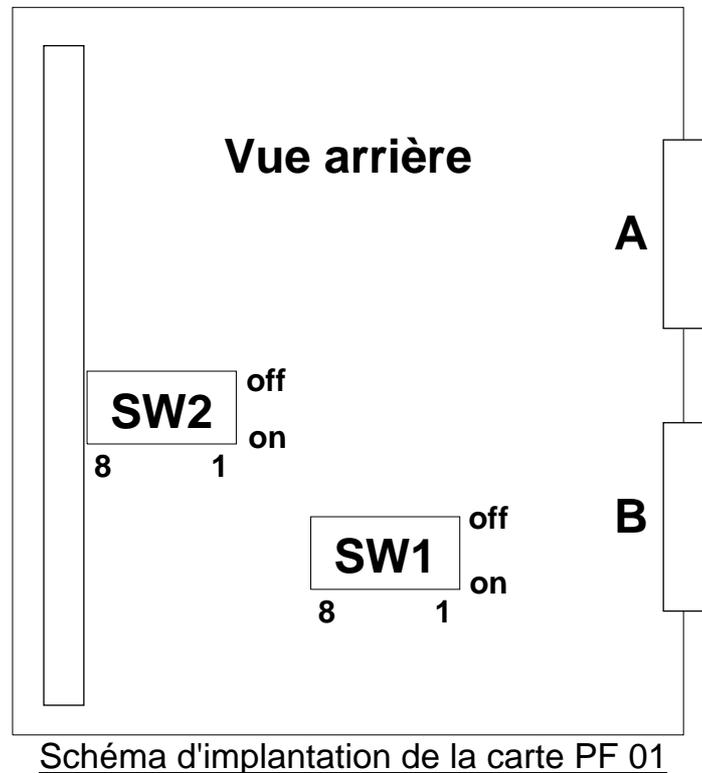
- S1 = IRQ 3
- S2 = IRQ 5
- S3 = IRQ 6
- S4 = IRQ 7
- S5 = IRQ 10
- S6 = IRQ 11
- S7 = IRQ 12
- S8 = IRQ 15

La carte PF 01 est également 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).

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

A la livraison, aucune interruption n'est sélectionnée.

4.3. Fonction Chien de garde (Watchdog)



La carte PF 01 est équipée d'une fonction chien de garde (Watchdog). Le watchdog est constitué d'une bascule monostable avec une constante de temps.

Cette bascule monostable est réarmée chaque fois que la carte PF 01 est adressée en lecture (RD) ou en écriture (WR). Si la fonction watchdog est configurée et si la carte PF 01 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 système, c-à-d que les sorties digitales ou analogiques seront désactivées.

La fonction Watchdog est sélectionnée à l'aide du micro-interrupteur S8 du bloc SW1.

Correspondance du micro-interrupteur S8 du bloc SW1:

S8 : ON = watchdog en fonction : temps de déclenchement 5 secondes
OFF = watchdog inactif

4.4. Définition des adresses

Fonctions I/O

Adresses	Lecture (RD)	Ecriture (WR)
Base + 0 à Base + 7	Fonctions I/O du MODULE A et désactivation de l'interruption A	
Base + 8 à Base + Fh	Fonctions I/O du MODULE B et désactivation de l'interruption B	

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

Lecture des registres d'interruptions

Adresses	Lecture (RD)
Base + 8000	Interruption MODULE A
Base + 8008	Interruption MODULE B

Fonctions BOOT&WORK ®

Adresses	Lecture (RD)	Ecriture (WR)
Base + 4000 Base + C000	Autoconfiguration du MODULE A	
Base + 4008 Base + C008	Autoconfiguration du MODULE B	

5. Installation de la DLL

5.1. Introduction

5.1.1 Installation du driver de la carte PF 01

Arrêtez votre PC et installer la carte dans un Slot PC/104

Redémarrer le PC et installer le Driver de communication en procédant comme décrit ci-dessous :

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

Attention : Par défaut, l'installation se fait pour Windows 95 et 98. Choisissez l'option Sélection manuelle puis Windows NT si tel est votre système.

5.1.2 Descriptif des fichiers installés

L'installation a créé un répertoire sous *C :SELIAtec\PF01* (à moins que vous n'ayez modifié le chemin à l'installation) . Sous ce répertoire, vous trouverez :

- Un fichier PF01.txt nécessaire à l'exportation sous Visual Basic
- Un exécutable *Config.exe* qui vous permet de vérifier que la carte est reconnue et que les modules détectés correspondent à ceux désirés.
- Un répertoire **Tests** comprenant les programmes source (en langage c) et les exécutables des modules
 - RA01.
 - RA02.
 - RA03.
 - RA04.
 - RA05.
 - RC01.
 - RC02.
 - RC03.
 - RD01.
 - RD02.
 - RD03.
 - RD04.
 - RMXX.

- Un répertoire **Headers** comprenant tous les fichiers.h nécessaires au développement de vos applications.
 - PF01.h : nécessaire à tous les modules.
 - Models.h : nécessaire au fonctionnement du *BOOT&WORK*.
 - I2C.h : nécessaire au module de communication.
 - Profile.h et Complex.h : nécessaire à l'appel des fonctions *PF01_RMXX_linéar*, *PF01_RMXX_circular* et *PF01_RMXX_execute* des modules RM01 et RM02.
 - Rmxx.h : nécessaire à l'appel des fonctions bas niveau des modules RM01 et RM02.

- Un répertoire **Librairies** comprenant les fichiers contenant les fonctions développées par SELIAtec. Celui-ci contient 3 sous répertoires :
 - Debug** : contient les fichiers PF01.lib et PF01.dll en Debug (version complète et dense de la compilation). Ces fichiers ne sont nécessaires que si vous rencontrez des problèmes de compilation avec la version Release.
 - Release** : contient les fichiers PF01.lib et PF01.dll en Release (version ne comprenant que le strict nécessaire à l'exécution). Ces fichiers sont suffisants pour la quasi-totalité des applications.

5.1.3 Vérification de l'installation

Afin de vérifier que le driver fonctionne correctement et pour vous faciliter vos tests, l'installation va placer des exécutables directement sous le menu :
Démarrer\Programmes\SELIAtec_PF01.

Lancer le programme *Config*. Un écran doit apparaître, indiquez l'adresse de base de la carte. Appuyez ensuite sur détection et vérifiez que les modules détectés sont identiques à ceux présents sur la carte.

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 lancer le ou les programmes de test correspondant à votre ou vos modules :

(Démarrer\Programmes\SELIAtec_PF01).

Ces programmes de test dont les sources vous sont fournies dans le répertoire *SELIAtec\Tests\Sources* sont très simples et vérifient les fonctions de base des modules. Effectuez les tests en choisissant les options fournies par les différents programmes. En cas de problème à l'appel de ces programmes, reportez-vous à la section problème à l'appel de programme.

5.1.4 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.

Lors de la désinstallation (menu *Démarrer\Programmes\SELIAtec_PF01\Désinstaller PF01*), le programme détruit les différents répertoires où ont été installés des fichiers. Ainsi, si vous avez créé un projet sous le répertoire *C:\SELIAtec*, celui-ci sera détruit lors de la désinstallation.

- Vérifiez que vous n'avez pas de fichiers qui vous sont propres sous *C:\SELIAtec*.

5.1.5.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 :

Aller dans le menu *Démarrer\Rechercher\Fichiers* ou *Dossiers* et taper *IOSelia.dll*. Il doit se situer *sous C:\Windows\System*.

Faites de même pour vérifier que le fichier *IOSelia.vxd* se trouve sous *C:\Windows\System*.

Lancer le programme *config.exe*. Redémarrer votre ordinateur, si la carte n'est toujours pas reconnue, le problème n'est pas dû au driver mais au matériel.

Windows NT/2000:

Aller dans le menu *Démarrer\Rechercher\Fichiers* ou *Dossiers* et taper *IOSelia.dll*. Il doit se situer sous *C:\WinNT\System*.

Faites de même pour vérifier que le fichier *IOSelia.sys* se trouve sous *C:\WinNT\System32\DRIVERS*.

Lancer le programme *config.exe*. Redémarrer votre ordinateur, si la carte n'est toujours pas reconnue, le problème n'est pas dû au driver mais au matériel.

5.1.6 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*.

5.1.7 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 PF01.dll sous *C:\Windows\System* pour Windows 95 et 98 ou *C:\WinNT\System* pour NT. Le fichier installé automatiquement est la version Release.
- Si vous avez ce message mais que la Dll est présente, remplacez celle-ci par la version Debug fournie dans *C:\SELiAtec\PF01\Librairies\Debug*.
- 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* -> *ressources matérielles*. Faites une liste des ressources allouées à tous vos périphériques et vérifiez que ces ressources ne se chevauchent pas.

5.1.8 Problème à la compilation d'une application :

Vérifiez que vous avez bien sous le répertoire courant les fichiers nécessaires à votre application. Vous devez obligatoirement avoir, quel que soit votre module, les fichiers PF01.lib et PF01.h (vous pouvez aussi placer le fichier PF01.dll mais il est plus propre de l'avoir sous *C:\Windows\System*). Le fichier PF01.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.

5.2. 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 PF01 dans les environnements Windows 95/98, et Windows NT.

Avant d'utiliser une carte PF01, il est nécessaire d'appeler la fonction :

- "PF01_define_card"

pour identifier automatiquement les différents modules présents grâce au concept *BOOT&WORK*®. Pour cela il faut indiquer l'adresse de base de la carte PF 01 configurée à l'aide des micro-interrupteurs (voir documentation technique pour plus d'information)

Chacun des modules pouvant être inséré sur une carte PF01 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 « PF01_define_card » réalise l'identification successive des modules insérés sur une carte PF01 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 PF01).

Après son appel, les fonctions "PF01_*_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.

A la fin de l'utilisation d'une carte PF01, il est indispensable d'appeler la fonction :

"PF01_undefine_card"

afin de libérer les ressources réservées et de permettre à un autre logiciel de réserver la carte.

En fonction des modules présents sur votre carte PF01, exécutez les fonctions

- "PF01_*_entrees" ou "PF01_*_sorties" pour lire ou écrire des valeurs analogiques
- "PF01_*_raw_entrees" ou "PF01_*_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.

- "PF01_raw_outportb"
- "PF01_raw_inportb"

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

En cas d'erreur la fonction "PF01_error" indique le code de la dernière erreur détectée.

La librairie "PF01" 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 5.0, 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 PF01.

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

6. Définitions des différentes fonctions relatives à la carte PF01

enum PF01_Error_Codes

Liste des codes erreurs des différentes fonctions

aucune erreur

PF01_ERROR_NONE = IOSELIA_ERROR_NONE

le numéro de carte est invalide

PF01_ERROR_BAD_CARD_NUMBER

l'emplacement ne contient pas le module en question

PF01_ERROR_BAD_MODULE

le numéro d'emplacement est invalide

PF01_ERROR_BAD_SLOT_NUMBER

l'offset demandé est trop important

PF01_ERROR_BAD_PORT_OFFSET

la valeur n'est pas dans la plage autorisée

PF01_ERROR_OUT_OF_BOUNDS

la valeur lue est incorrecte

PF01_ERROR_BAD_VALUE

le contenu de l'eeprom du module est invalide

PF01_ERROR_BAD_EEPROM,

l'accès à l'I2C est impossible

PF01_ERROR_BAD_EEPROM_I2C,

tous les descripteurs de cartes ont été alloués

PF01_ERROR_CARD_FULL

Dernière erreur

PF01_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 PF01_error() renvoie le code de cette erreur.

enum PF01_Types

Liste des différents types de modules

PF01_TYPE_NONE

aucun module

PF01_TYPE_RA01

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

PF01_TYPE_RA02

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

PF01_TYPE_RA03

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

PF01_TYPE_RA04

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

PF01_TYPE_RA05

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

PF01_TYPE_RC01

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

PF01_TYPE_RC03

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

PF01_TYPE_RD01

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

PF01_TYPE_RD02

8 sorties relais statiques AC, contacts ouverts au repos

PF01_TYPE_RD03

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

PF01_TYPE_RD04

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

PF01_TYPE_RM01

commande d'axes pour 1,2 ou 4 moteurs servo avec retours codeur

PF01_TYPE_RM02

commande d'axes pour 1,2 ou 4 moteurs pas à pas

PF01_TYPE_RT01

1 voie de communication série RS 232

ULONG PF01_error (void)

Cette fonction renvoie le code de la dernière erreur survenue. Elle doit être appelée lorsqu'une des fonctions PF01 a échoué afin de déterminer la cause de l'échec.

Retour: valeur de la dernière erreur (Voir aussi PF01_ErrorCodes)

```
BOOL PF01_define_card(IN ULONG base, IN ULONG interruptno, IN void *info, OUT ULONG * card)
```

Cette fonction doit être appelée afin d'utiliser une carte PF01 et plus particulièrement les modules installés dans les 4 emplacements. Cette fonction réserve les ports d'entrées/sorties d'accès aux différents modules ainsi que l'interruption éventuelle.

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 "PF01_*_is".

Utilisez la fonction "PF01_undefine_card" pour libérer la carte.

Paramètre: **base** - adresse de base de la carte (en fonction de la position des micro-interrupteurs), l'adresse par défaut est 0x100h

interruptno - numéro de l'interruption, si vous voulez utiliser la carte sous interruption.

info- argument variable.

 Sous Win95 : pointeur sur une fonction à exécuter lors de l'interruption,

 Sous WinNT : nom de l'événement déclenché lors de l'interruption

card - 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 PF01_undefine_card(vous pouvez considérer que le numéro de carte est l'adresse de base de votre carte). Si la définition a réussi, ce paramètre contient le descripteur nécessaire aux accès ultérieurs.

Retour: TRUE en cas de succès de l'opération

```
BOOL PF01_undefine_card (IN ULONG carte)
```

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: **carte** - le descripteur de carte renvoyé par la fonction "PF01_define_card"

Retour: TRUE en cas de succès de l'opération
Voir aussi PF01_ErrorCodes et PF01_define_card

BOOL PF01_type_is(*IN* ULONG , *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 "PF01_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 PF01_Types

BOOL PF01_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 "PF01_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 PF01_Models

BOOL PF01_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 "PF01_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 PF01_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 "PF01_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 PF01_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 "PF01_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 PF01_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 "PF01_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 PF01_ports_resources (IN ULONG carte, OUT ULONG * debut, OUT ULONG * fin)

Cette fonction renvoie les ports d'entrées/sorties utilisés par une carte PF01.

Paramètre: **carte** - le descripteur de carte renvoyé par la fonction PF01_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 PF01_interrupt_resources (*IN* ULONG carte, *OUT* ULONG * interruptno)

Cette fonction renvoie le numéro de l'interruption utilisée par une carte PF01.

Paramètre: **carte** - le descripteur de carte renvoyé par la fonction "PF01_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 PF01_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 PF01 accessible à l'adresse 'base'+0x8000+slot_offset(0-3).

Paramètre: **carte** - le descripteur de carte renvoyé par la fonction "PF01_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 PF01_raw_outportb (*IN* ULONG carte, *IN* LONG slot, *IN* ULONG offset, *IN* ULONG 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 "PF01_define_card"
 slot - numéro du slot sur lequel se trouve le module :
 0 pour A, 1 pour B, 2 pour C, 3 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 PF01_raw_intportb (*IN* ULONG carte, *IN* ULONG slot, *IN* ULONG offset, *IN* ULONG *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 "PF01_define_card"
 slot - numéro du slot sur lequel se trouve le module :
 0 pour A, 1 pour B, 2 pour C, 3 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

8.1 Fonctions I/Oselia : accès aux ports d'entrées/sorties

La librairie I/Oselia permet l'accès aux ports d'entrées/sorties ainsi que la gestion des interruptions dans les environnements Windows 95/98 et Win NT.

Pour un système d'exploitation multitâche sans compromis, l'utilisation de la librairie est conseillée dans la mesure où l'accès exclusif aux ressources est garanti par un mécanisme de verrouillage. De plus, certains logiciels ne disposent pas d'instructions d'entrées/sorties dans leur langage de programmation. Dans ce cas, il est impératif d'utiliser les fonctions fournies par cette librairie dynamique .

Les différentes fonctions présentes dans la librairie dynamique sont:

- **IOSelia_error** : Indication de la dernière erreur commise
- **IOSelia_lock** : Verrouillages de ressources
- **IOSelia_unlock** : Déverrouillages des ressources
- **IOSelia_inportb** : Lecture d'un port d'entrée/sortie
- **IOSelia_outportb** : Ecriture sur un port d'entrée/sortie

Les fonctions "IOSelia_inportb" et "IOSelia_outportb" représentent les fonctions utiles de la librairie et elles correspondent aux instructions machines d'accès aux ports d'entrées/sorties, à savoir "in al,dx" et "out dx,al".

Toutes les fonctions renvoient une valeur booléenne indiquant le succès de l'opération demandée. En cas d'erreur la fonction "IOSelia_error" indique le code de la dernière erreur commise.

La librairie "IOSelia" 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 5.0, les différentes fonctions doivent être déclarées (voir l'instruction DECLARE dans votre manuel de programmation de Visual Basic).

enum IOSELIA_ErrorCodes

Liste des codes erreurs des différentes fonctions. Toutes les fonctions de la librairie renvoient une valeur booléenne indiquant le succès de l'opération. Si une erreur s'est produite la fonction IOSELIA_Error () renvoie le code de cette erreur.

IOSELIA_ERROR_NONE

aucune erreur

IOSELIA_ERROR_INVALID_PARAMETERS

Paramètres sont incorrects

IOSELIA_ERROR_USED_IOPORTS

La plage de ports d'entrées/sorties n'est pas libre

IOSELIA_ERROR_USED_INTERRUPT

L'interruption demandée n'est pas libre

IOSELIA_ERROR_NOT_LOCKED

La ressource n'est pas verrouillée

IOSELIA_ERROR_BAD_IOPORTS

Les ports d'entrées/sorties demandés sont invalides

IOSELIA_BAD_INTERRUPT

L'interruption demandée est invalide

IOSELIA_ERROR_TOO_MANY_LOCKS

Le nombre de verrous maximum a été atteint

IOSELIA_ERROR_INVALID_HANDLE

Le descripteur est invalide

IOSELIA_ERROR_CANNOT_RELEASE_INTERRUPT

L'interruption ne peut être déverrouillée

IOSELIA_ERROR_CANNOT_CREATE_EVENT

L'événement associé à l'interruption ne peut être créé

IOSELIA_ERROR_DEVICE_IO_CONTROL

Erreur lors de la procédure DEVICE_IO_CONTROL

IOSELIA_ERROR_USED_RESSOURCES

Les ressources demandées sont déjà utilisées

IOSELIA_ERROR_UNSUPPORTED

Cette fonctionnalité n'est pas supportée dans cette version

Liste des fonctions

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.

- ULONG IOSelia_error** (void)
Indication de la dernière erreur commise
- BOOL IOSelia_Lock**
(IN ULONG base, IN ULONG interruptno, IN VOID interruptno
IN VOID * info, OUT ULONG * handle)
Verrouillage des ressources
- BOOL IOSelia_unlock**
(IN ULONG handle)
Déverrouillage des ressources
- BOOL IOSelia_outportb**
(IN ULONG handle, IN ULONG offset, OUT ULONG * value)
Ecriture sur un port d'entrée/sortie
- BOOL IOSelia_inportb**
(IN ULONG handle, IN ULONG offset, OUT ULONG * value)
OUT ULONG * value
Lecture sur un port d'entrée/sortie

ULONG IOSelia_error (void)

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

Retour code de la dernière erreur
Voir aussi IOSelia_ErrorCodes

```
BOOL IOSelia_lock (IN ULONG iostart, IN ULONG ioend, IN ULONG interruptno, IN VOID *info, OUT ULONG *handle)
```

Cette fonction doit être appelée afin de réserver l'accès à une ressource.

Une ressource est soit :

- soit une plage d'adresse d'entrées/sorties
 - soit un numéro d'interruption avec une fonction utilisateur associée éventuelle
- soit les deux à la fois

Paramètre **iostart**- adresse de base de la plage d'entrée/sortie
 ioend- adresse de fin de la plage d'entrée/sortie
 interruptno- numéro de l'interruption déclenchée
 info- argument variable.
 Sous Win95 : pointeur sur une fonction à exécuter lors de l'interruption,
 Sous WinNT : nom de l'événement déclenché lors de l'interruption

Retour TRUE en cas de succès de l'opération

Voir aussi IOSelia_ErrorCodes

Exemple:

```
BOOL success ;
ULONG handle, error ;
                  // réservation des ports d' e/s de 0x300 à 0x308
success=IOSelia_lock (0x300, 0x308, 0,NULL, &handle) ;
If ( ! Success ) error= IOSelia_error ( ) ;
                  // réservation d'une interruption et définition d'une fonction
                  // utilisateur sous Windows 95
                  // le prototype de la fonction est "void (* func) ( )"
success=IOSelia_lock (0,0,7,isrfunction,&handle) ;
If ( ! succès ) error=IOSelia_error ( ) ;
                  // réservation d'une interruption et définition d'un événement
                  // déclenché sous Windows NT
                  // l'utilisateur devra se mettre en attente de l'événement par
                  // la fonction "WaitForSingleObjet" dans une tâche séparée
                  // pour éviter une attente active.
Success= IOSelia_lock (0,0,7,"EventInt 7",&handle);
If (! Success) error=IOSelia_error ( ) ;
```

Le descripteur "handle" renvoyé par la fonction permet l'accès exclusif aux ports d'entrées/sorties réservés. Dans le cas d'une réservation d'interruption, celle-ci est verrouillée afin d'éviter tout conflit avec un autre périphérique. Le déclenchement de l'interruption provoque soit l'exécution d'une fonction utilisateur dans le cas de Windows 95, soit le déclenchement d'un événement sous Windows NT.

Utilisez la fonction "IOSelia_unlock" pour libérer les ressources.

BOOL IOSelia_unlock (IN ULONG handle)

Cette fonction restitue les ressources réservées et identifiées par le prescripteur "handle".

Paramètre **handle-** est le descripteur de carte renvoyé lors de l'allocation de la ressource par IOSelia_lock

Retour TRUE en cas de succès de l'opération
Voir aussi IOSelia_ErrorCodes et IOSelia_lock

Exemple:

```
          // Libération des ressources identifiées par le descripteur handle  
BOOL success= IOSelia_unlock (handle) ;  
If (! Success) error=IOSelia_error( )
```

**BOOL IOSelia_outportb (IN ULONG handle, IN UNLOG offset OUT
ULONG value)**

Ecriture d'un caractère sur un port d'entrée/sortie. Celui-ci est référencé par le "handle" retourné lors de la réservation de la plage des ports d'entrée/sortie plus un décalage "offset ajouter à l'adresse de base référencée par le "handle".

Paramètre **handle-** est le descripteur renvoyé lors de l'allocation de la ressource par "IOSelia_lock
Offset- correspond au décalage du port auquel on souhaite accéder, par rapport à l'adresse de base.
Value- représente l'octet à émettre sur le port d'entrées/sorties

Retour TRUE en cas de succès de l'opération
Voir aussi IOSelia_ErrorCodes et IOSelia_lock

Exemple:

```
ULONG value_8bits=0xE8 ;  
BOOL success=IOSelia_inportb (handle,0x5,value_8bits);  
If (! Success ) error=IOSelia_error ( );  
    // Ecriture de la valeur 1xE8 sur le port d' e/s 0x305
```

<pre>BOOL IOSelia_inportb (IN ULONG handle, IN ULONG offset, OUT ULONG * value)</pre>

Lecture d'un caractère sur un port d'entrée/sortie. Celui-ci est référencé par le "handle" retourné lors de la réservation de la plage des ports d'entrées/sorties plus un décalage "offset" à ajouter à l'adresse de base référencé par le "handle".

Paramètre **handle-** est le descripteur renvoyé lors de l'allocation de la ressource par IOSelia_lock.
 Offset- correspond au décalage du port auquel on souhaite accéder, par rapport à l'adresse de base.

Retour TRUE en cas de succès de l'opération
Voir aussi IOSelia_ErrorCodes et IOSelia_lock

Exemple:

```
ULONG value_8bits;  
BOOL success= IOSelia_inportb (handle, 0x5)
```