



# Les outils de programmation des automates SAIA®PCD

Programmation, développement et configuration des solutions d'automatismes sur API

Controls saia-burgess

## Manuel utilisateur

### Les avantages décisifs des outils de programmation PG5

- **Portabilité des programmes:** les programmes PG5 sont exploitables sur toutes les plates-formes SAIA®PCD.
- **Organisation des programmes en fichiers** (contenant plusieurs blocs de programme): cette structure simplifie le partage des fichiers programmes entre plusieurs SAIA®PCD.
- **Mise en oeuvre des programmes PG3 et PG4 existants.**
- **Environnements de programmation et de mise au point intégrés** à chaque éditeur de programme.
- **Simplicité de programmation des terminaux de dialogue** avec l'éditeur de pupitre opérateur (HMI Editor).
- **Puissant jeu d'instructions** supporté par des macros et des directives assembleur.

### Les points forts du PG5

- **Un Gestionnaire de symboles** administrant tous les symboles de type local, global et réseau ou les groupes de symboles. L'adressage automatique vous épargne l'adressage fixe.
- **Un Gestionnaire de projet** pour piloter des installations complexes de PCD en réseau, sans oublier les afficheurs et la documentation.
- **Des fonctions de mise en service et de détection d'erreurs en ligne**, sur Ethernet-TCP/IP, SAIA®S-Bus, par modem...
- **Plusieurs environnements de programmation:**
  - FUPLA (blocs de fonctions, FBD)
  - S-Edit (liste d'instructions, IL)
  - GRAFTEC (diagramme fonctionnel en séquence, SFC)
- **Des éditeurs de réseaux intégrés** pour SAIA®S-Bus, PROFIBUS DP et FMS, LONWORKS®.
- **De puissantes bibliothèques complémentaires** pour accroître la richesse fonctionnelle du PG5.

## Table des matières

---

	Avant Propos	II
	Mise à jour	III
1	PCD – Mise en service rapide	1-3
2	Gestionnaire de projets	2-3
3	PCD - Ressources	3-3
4	Programmer avec Fupla	4-3
5	Structuration du programme	5-3
6	Programmer en Graftec	6-3
7	Programmer en IL (Instruction List)	7-3
8	Outils complémentaires	8-3
9	Saia Networks	9-2
10	Profi-S-Bus	10-2
11	Ether-S-Bus	11-2
12	Profi-S-IO	12-2

## Avant Propos

---

Ce document n'est pas un manuel de mise en service détaillé des automates SAIA-PCD, mais un document pour introduire le sujet et fournir un moyen d'apprentissage rapide qui ne présente que l'essentiel. Pour obtenir de plus amples informations veuillez alors vous référer aux aides fournies par l'outil de programmation ou aux manuels détaillés du CD de documentation.

Pour que votre formation se déroule dans de bonnes conditions, nous vous conseillons de prévoir les programmes, documents et matériels suivants:

- ♣ CD PG5 version 1.4
- ♣ CD de documentation 26/803
- ♣ 1 automate PCD2.M480 <sup>1</sup>
- ♣ 1 module de 8 entrées digitales PCD2.E110
- ♣ 1 module de 8 sorties digitales PCD2.A400
- ♣ 1 câble de programmation PCD8.K111

Toutes les instructions nécessaires à l'installation du PG5 1.4 sur votre ordinateur sont fournies avec le CD PG5 version 1.4 (Voir sous : CD:\PG5\InstallationGuide\_F.htm)

Une dernière précision, tous les noms anglais des menus, commandes, options, boutons, présents dans le programme PG5, sont rapportés dans ce manuel avec des caractères *italiques*.

Nous vous souhaitons plein succès pour votre formation et vos projets avec les produits PCD.

Votre partenaire Saia-Burgess Controls

---

<sup>1</sup> un autre PCD peut aussi convenir

## Table des matières

---

<b>1</b>	<b>PCD – MISE EN SERVICE RAPIDE</b>	<b>3</b>
1.1	Au sommaire de ce chapitre	3
1.2	Installation du matériel	4
1.2.1	Exemple: éclairage des escaliers	4
1.2.2	Schéma d'implantation du PCD2.M480	4
1.2.3	Assemblage du PCD2.M480	5
1.2.4	Câblage	5
1.3	Edition d'un programme PG5	6
1.3.1	Installation du software	6
1.3.2	Démarrer PG5	6
1.3.3	Créer un nouveau projet	6
1.3.4	Configurations	8
1.3.5	Ajouter un fichier programme	10
1.3.6	Ouvrir un Fichier	11
1.3.7	Editer le programme	11
1.4	Mise en service et test du programme	15
1.4.1	Construction d'un programme ( <i>Build</i> )	15
1.4.2	Charger le programme dans le PCD ( <i>Download</i> )	15
1.4.3	Recherche des erreurs et vérification ( <i>Debugging</i> )	16
1.5	Correction de programme	17



# 1 PCD – Mise en service rapide

---

## 1.1 Au sommaire de ce chapitre

Comme première prise de contact avec le matériel PCD, nous vous proposons d'entrer directement en matière avec la réalisation d'une petite application concrète.

Cela ne représente aucune difficulté si vous n'avez pas d'expérience des produits SAIA, tout est présenté en détail. Cet exemple présente la mise en service d'un PCD2.M480 avec la programmation, les tests et l'outil de programmation PG5.

Les chapitres suivants de ce document reprendront plus en détails, le contenu de la mise en service rapide. Avec bien d'autres informations supplémentaires.

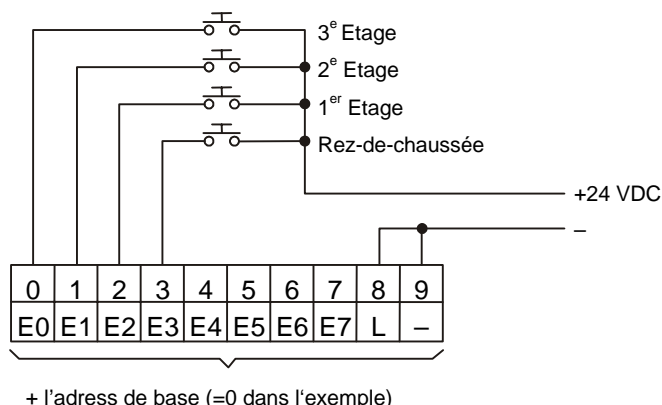


### 1.2.3 Assemblage du PCD2.M480

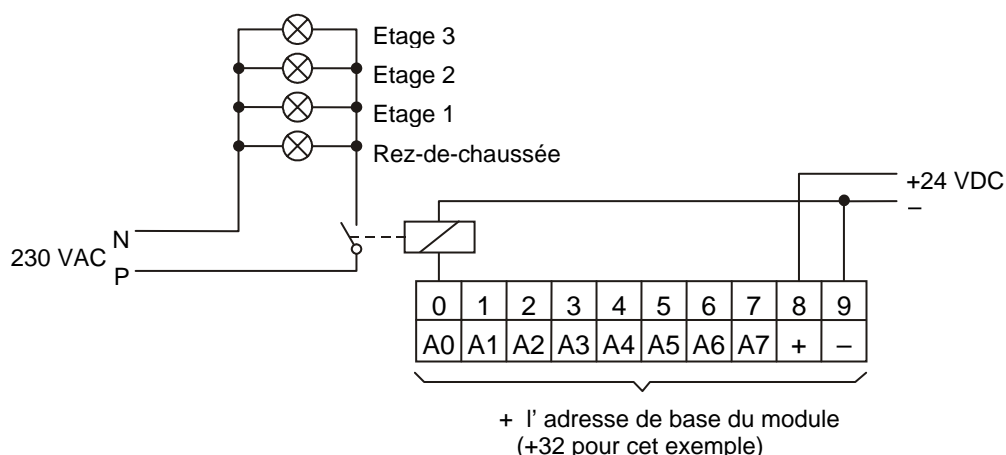
1. Retirer le couvercle et insérer la batterie au lithium de 3.0 V
2. Insérer un module PCD2.E110 à l'adresse de base 0. Cela permet de disposer de 8 entrées digitales 24 VDC aux adresses I0 à I7.
3. Insérer un module PCD2.A400 à l'adresse de base 32. Cela permet de disposer de 8 sorties digitales 24 VDC / 0,5 A aux adresses O 32 à O 38.
4. Faire coulisser chaque module d'entrées/sorties dans son logement et le pousser à fond jusqu'à ce qu'il se verrouille en position.

### 1.2.4 Câblage

1. Raccorder l'alimentation 24 VDC aux bornes 20 (+) et 23 (-). Il est possible d'employer une alimentation régulée de 24 VDC  $\pm$  20 % ou une simple alimentation 18 VDC  $\pm$ 15 % sans régulation (19 VAC redressée par un pont de diodes)
2. Raccorder les quatre entrées utiles selon la description hardware du module PCD2.E110 : raccorder les 4 interrupteurs d'impulsions sur les bornes 0 à 3, les bornes 8 et 9 sont raccordées au moins de l'alimentation



3. Raccorder la borne 0 sur la bobine du relais, la borne 8 sur le plus de l'alimentation 24 VDC et la borne 9 sur le moins de l'alimentation.



4. Raccorder l'interface RS 232 de l'ordinateur sur la prise PGU du PCD, utiliser un câble PCD8.K111.

Pour de plus amples renseignements concernant le montage hardware et le câblage, veuillez vous référer au manuel hardware de votre PCD.



## 1.3 Edition d'un programme PG5

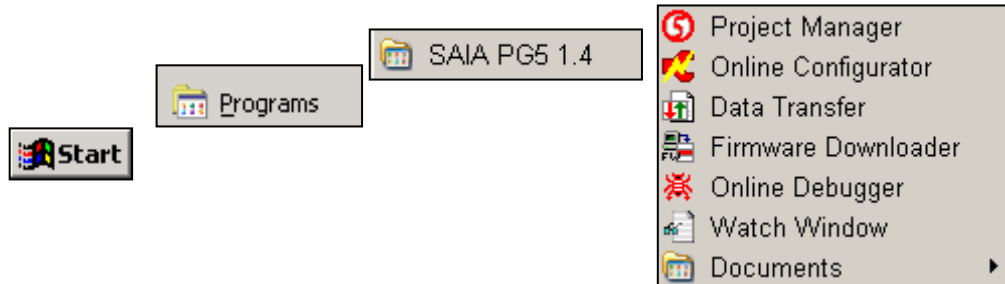
### 1.3.1 Installation du software

Installer le software SAIA PG5 sur l'ordinateur selon les directives livrées avec le CD de distribution. (CD:\PG5\InstallationGuide\_F.htm)

### 1.3.2 Démarrer PG5

Démarrer le gestionnaire de projet du PG5 avec le menu:

**Start --> Programs --> SAIA PG5 V 1.4 --> Project Manager**

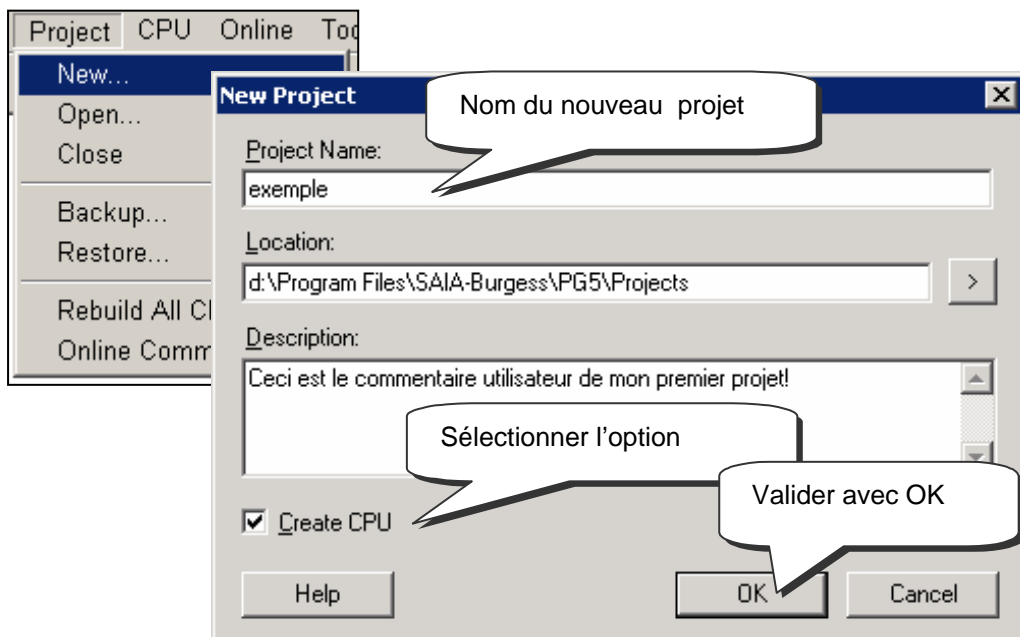


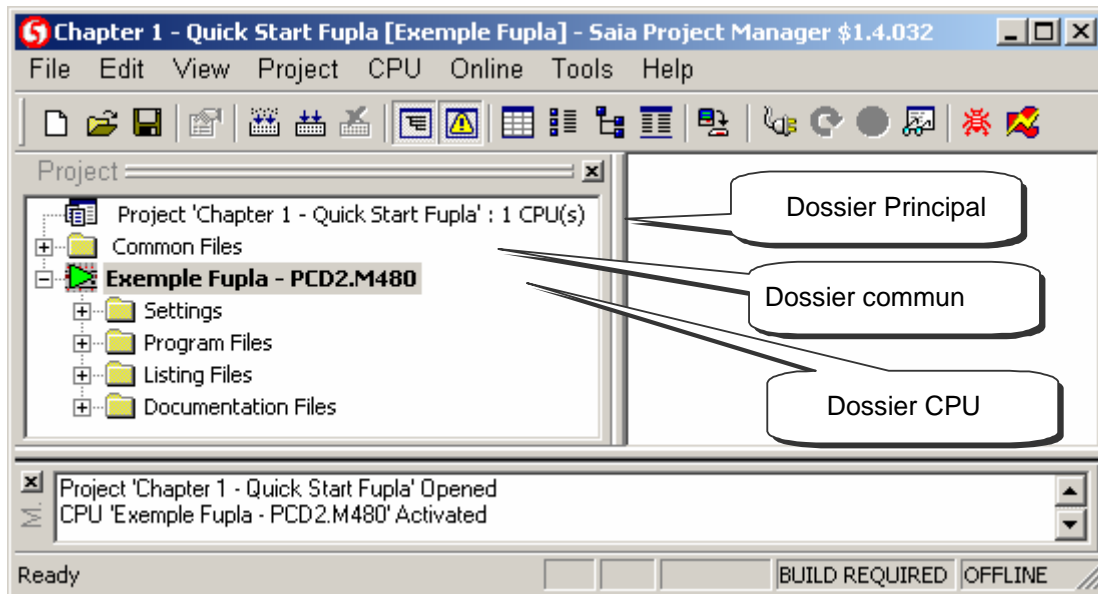
### 1.3.3 Créer un nouveau projet

Avant de se lancer dans l'écriture d'un premier programme, il faut d'abord créer ou ouvrir un nouveau projet qui contient les définitions de quelques paramètres de configuration ainsi que les fichiers constituant le programme d'application.

Si le projet n'est pas encore créé, sélectionner le menu *Project, New...*, définir le nom du nouveau projet dans le champ *Project Name*, sélectionner l'option *Create CPU* et valider avec le bouton *OK*.

#### Créer un nouveau projet





La fenêtre de gestion du projet est alors affichée. La sous fenêtre *Project*, représente la structure de votre projet. Au cas où cette fenêtre n'est pas représentée, il est possible de la faire apparaître en sélectionnant le menu *View, Project Tree*.

Les dossiers de la fenêtre *Project*, rassemblent les informations du projet selon certains critères d'organisation:

- Le dossier principal représente le projet avec son nom et le nombre de CPU qu'il contient.
- Le dossier commun, dans lequel, il est possible de déposer les programmes communs à tous les CPU.
- Puis le ou les dossiers CPU.

Chaque CPU correspond à un PCD. A l'intérieur du dossier CPU, nous trouvons:

- Un dossier *settings* avec les paramètres de configurations nécessaires à l'outil de programmation et à l'automate PCD.
- Un dossier *Program Files* avec les fichiers programmes de l'application PCD.
- Le dossier *Listing Files* est moins important, il contient des informations générées lors de la construction (*Build*) du programme d'application.

### Ouvrir un projet

Un projet déjà présent peut être ouvert avec le menu *Project, Open...*

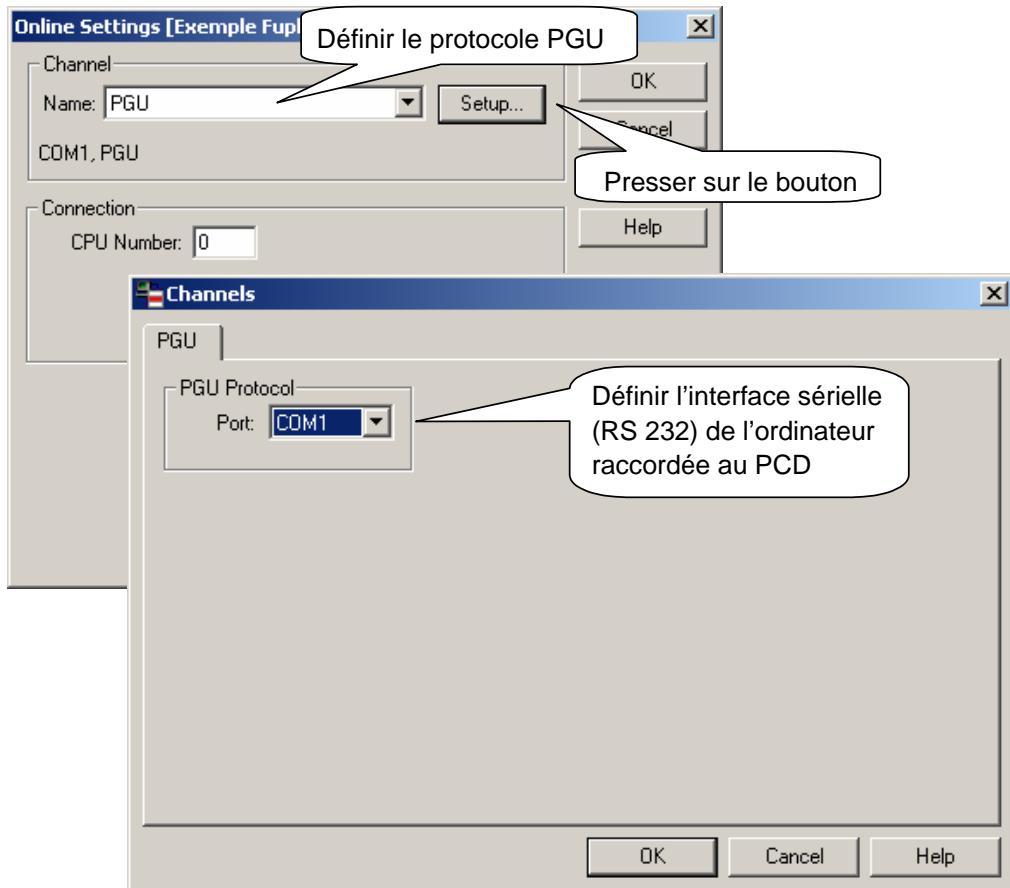
Il recherche tous les fichiers projet (.5pj) présents dans le répertoire des projets et les affiche dans une liste. Sélectionner l'un des projets de la liste ou marquer un projet de la liste et presser le bouton *Open*. Une autre alternative, presser le bouton *Browse* et chercher directement le fichier du projet ou CPU.

### 1.3.4 Configurations

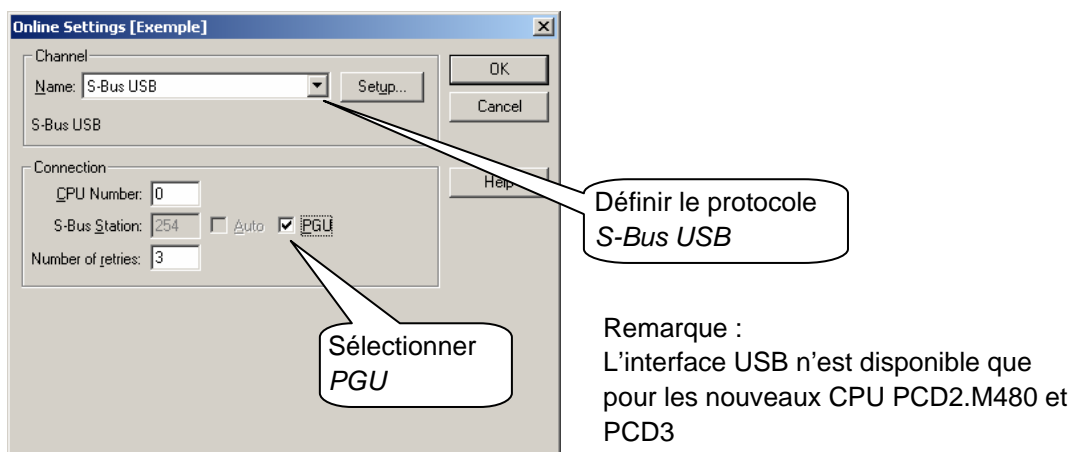
Avant de travailler sur un des CPU du projet, il est nécessaire de vérifier quelques paramètres de configurations pour s'assurer que l'outil de programmation et le PCD supportent convenablement le programme que nous allons réaliser.

Les *Onlines settings* permettent de définir les paramètres de communication entre l'outil de programmation et le PCD. Diverses fonctionnalités sont possibles, mais pour l'instant nous nous contentons de sélectionner le protocole par défaut et vérifier les paramètres de l'interface sérielle pour l'ordinateur.

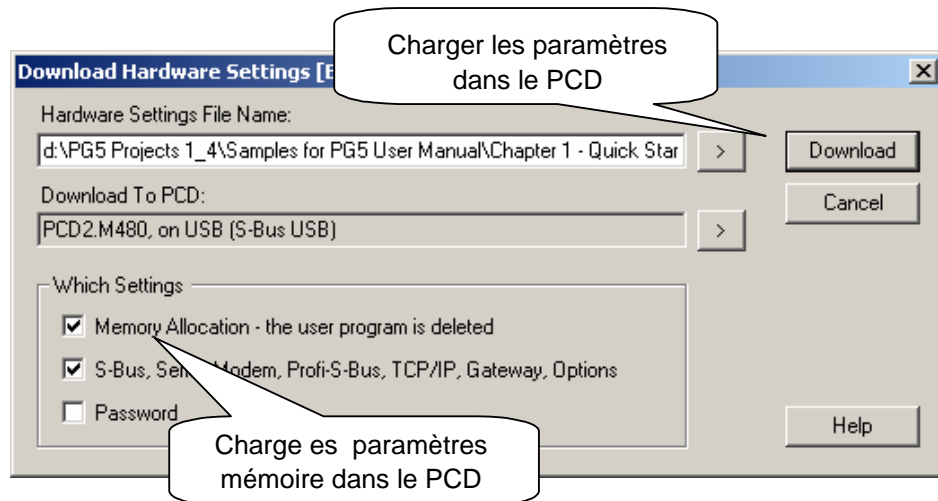
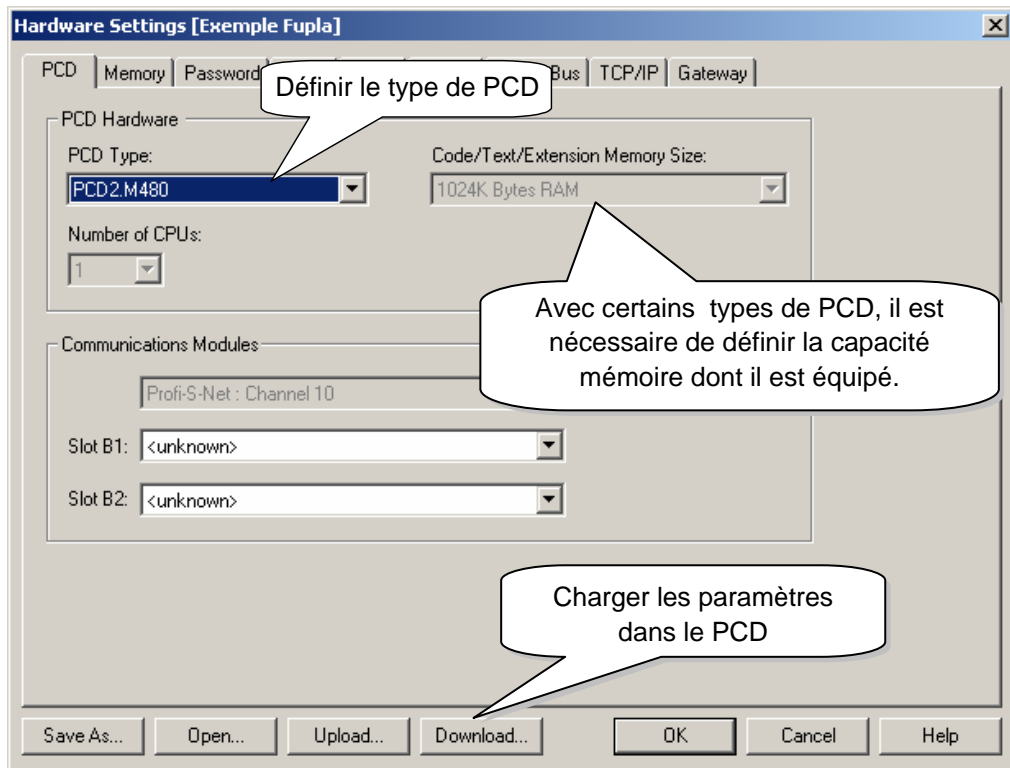
#### Channel PGU (RS 232)



#### Channel S-Bus USB



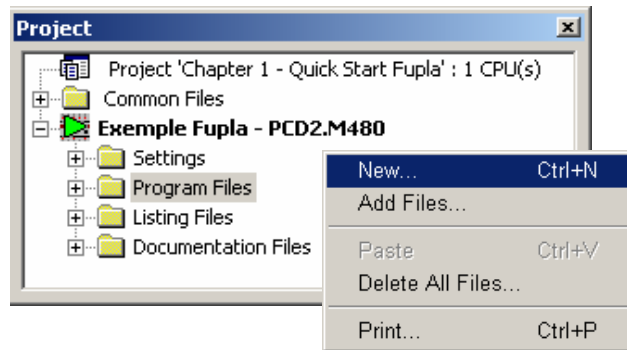
Les *Hardware settings* permettent de définir les paramètres du PCD. Ainsi que d'autres fonctions plus avancées tel que *Memory*, *S-bus*, *Gateway*, *Modem* et *Password* que nous n'utilisons pas pour l'instant. Ce qui est important de définir, c'est le type de PCD et la mémoire disponible pour l'utilisateur. Avec le PCD2.M480 'est très simple. Il est toujours livré avec 1024 KBytes RAM.



### 1.3.5 Ajouter un fichier programme

Les programmes d'application du PCD sont écrits dans un fichier que nous ajoutons au projet avec l'une des manipulations suivantes:

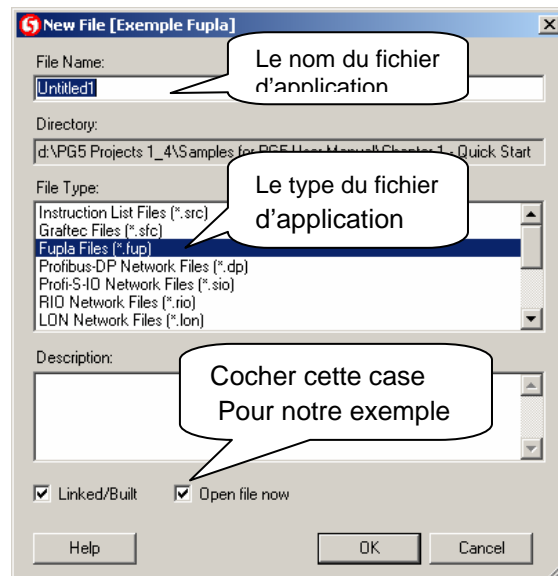
Marquer le dossier *Program File* de la fenêtre *Project*, cliquer sur le bouton droite de la souris pour faire apparaître le menu de contexte et sélectionner le menu *New...*



Une autre manière consiste à marquer le dossier *Program File* de la fenêtre *Project* et cliquer sur le bouton *New File* ou utiliser le menu *File, New...*

Une nouvelle fenêtre *New File* apparaît pour définir deux informations importantes: le nom du fichier d'application et le type de fichiers.

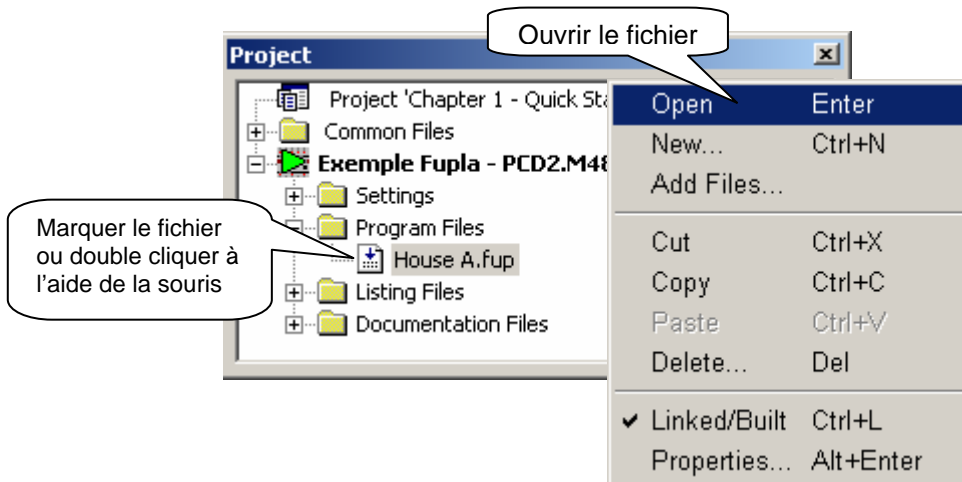
De nombreux éditeurs sont à disposition pour écrire les applications PCD. Le choix de l'éditeur dépend des préférences de l'utilisateur et du type d'application à réaliser. Pour notre première application nous choisirons l'éditeur le plus couramment employé et le plus universel: *Fupla File (\*.fup)*



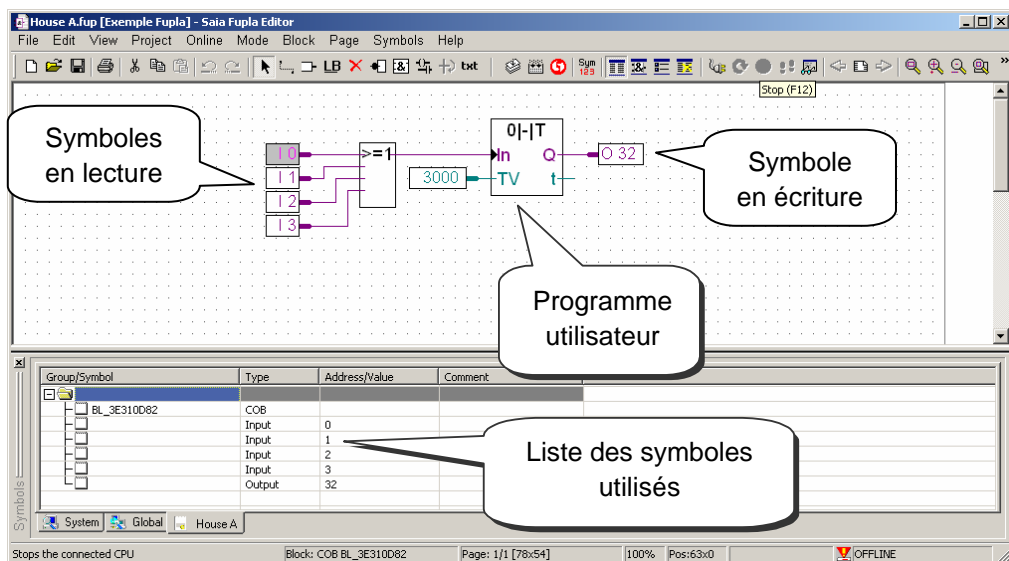
### 1.3.6 Ouvrir un Fichier

Si le fichier programme est déjà présent dans le dossier *Program Files*, ouvrez le de la manière suivante :

Marquer le fichier programme de la fenêtre *Project*, cliquer sur le bouton droit de la souris pour faire apparaître le menu de contexte et sélectionner le menu *Open...*



### 1.3.7 Editer le programme



## Edition des symboles

Les symboles représentent les informations à gérer avec le programme du PCD, soit l'éclairage des escaliers d'un bâtiment. Nous éditons les symboles dans les connecteurs de la page de Fupla. A gauche les symboles en lecture, à droite les symboles en écriture.

Ainsi, nous plaçons les informations des interrupteurs de lumières (I 0, I 1, I 2, I 3) dans les connecteurs de gauche et l'information pour la gestion du relais de lumière (O 32) dans le connecteur de droite. Les 5 minutes de temporisation de l'éclairage sont définies à gauche par une constante dont la valeur est exprimée en dixièmes de secondes. Soit 5 minutes de 60 secondes multiplié par 10 ( = 3000 \* 1/10e de seconde).



*Place  
Connectors*

Pour ajouter un connecteur et son symbole sur une page Fupla, sélectionner le bouton *Place Connectors* présent dans la barre des outils et positionner la souris sur la page Fupla. La sélection du bouton gauche de la souris permet d'ajouter un connecteur en lecture. Pour ajouter un connecteur en écriture, veuillez presser la touche *Shift* du clavier au moment de sélectionner le bouton de la souris. Le connecteur que vous venez d'ajouter est prêt pour recevoir un symbole, un curseur est représenté à l'intérieur du connecteur. Si vous ne souhaitez pas éditer immédiatement le symbole à l'intérieur du connecteur, sélectionnez la touche *ESC* du clavier et placer le connecteur suivant.

Pour éditer, modifier un symbole d'un connecteur déjà présent sur la page Fupla, sélectionner le connecteur avec un double clic rapide, un curseur est représenté à l'intérieur du connecteur, il est alors possible d'introduire l'adresse I 0 à I 3. Idem pour la sortie O 32 et la constante. Prenez garde à toujours laisser un espace entre la lettre I et l'adresse de l'entrée. De même avec la sortie.



*Show/Hide  
Symbol Editor*

Nous remarquons que la liste des symboles introduite dans les connecteurs se retrouve dans la fenêtre *Symbols*. Si l'éditeur de symbole est masqué, il est toujours possible de l'afficher avec le menu *View, Symbol Editor* ou avec le bouton *Show/Hide Symbol Editor*.

Remarque :

Il est possible que chaque nouvelle page offre par défaut une marge avec des connecteurs sur la gauche et la droite. Si vous préférez que les nouvelles pages soient présentées sans ces connecteurs et les placer vous-même selon votre convenance, veuillez désactiver l'option suivante avec le menu : *View, Options..., Connectors, Add Empty Side Connectors*.

Pour supprimer les connecteurs qui seraient éventuellement présents à gauche et droite de la page, sélectionner le menu : *Page, Remove Empty connectors*.

*Pour placer à nouveau les connecteurs sur une page blanche, sélectionner le menu : Page, Add Empty Side Connectors.*

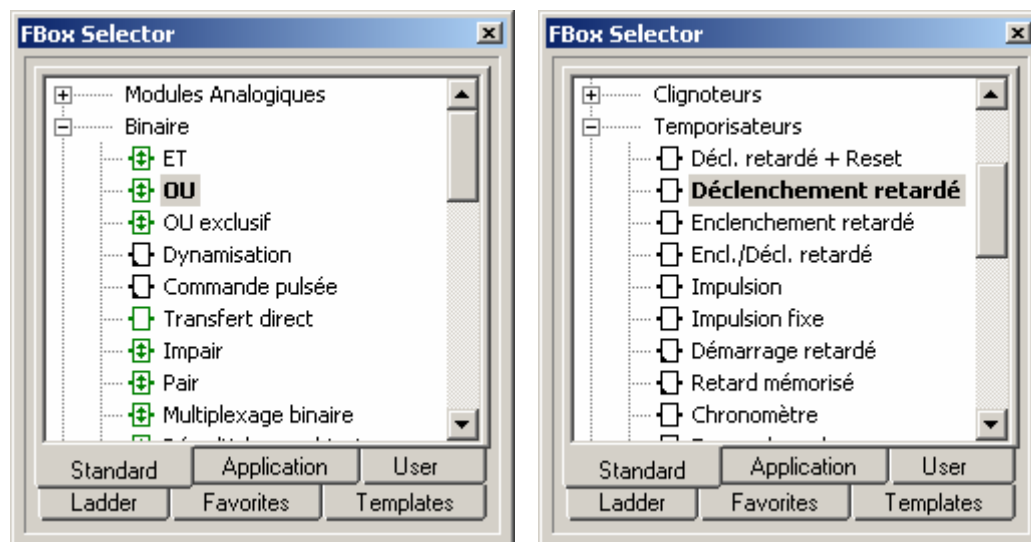
## Edition des fonctions du programme

Le programme est décrit entre les connecteurs en lecture et écriture. Nous y plaçons les fonctions graphiques qui constituent le programme d'application.

Ces fonctions graphiques sont disponibles à partir de la fenêtre *Fbox Selector...* que nous ouvrons avec le bouton *ADD FBox*.



Add Fbox



La première fonction nécessaire à notre exemple d'application permet de démarrer la minuterie pour allumer la lumière des escaliers avec une courte impulsion de l'un des interrupteurs de lumière. C'est une fonction *OU* logique que nous trouvons dans la famille de fonction *Binaire* de la librairie *standard*.

La seconde fonction permet de temporiser la lumière des escaliers pour un délai de 5 minutes. C'est une fonction de temporisation que nous trouvons dans la famille de fonction *Temporisateurs* de la librairie *standard*.

De plus amples informations concernant le groupe de fonction ou la fonction sélectionnée sont disponibles avec la touche F1 du clavier. Après avoir sélectionné une des fonctions graphiques de la fenêtre *Fbox Selector...*, placer la souris sur la page d'édition du programme et cliquer sur le bouton gauche la souris.

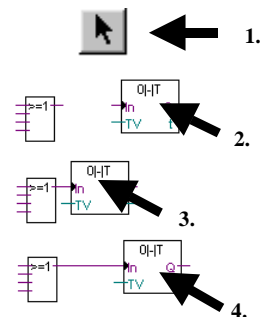
Avec certaines fonctions ajustables, tel le *OU* logique, il est possible de définir le nombre d'entrées que doit représenter la fonction. Ce choix s'effectue par un mouvement vertical de la souris, puis confirmer le choix en cliquant sur le bouton gauche de la souris.



## Lier les fonctions

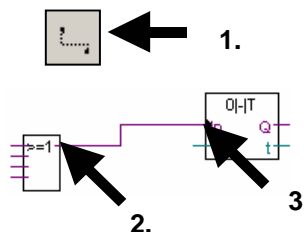
Première possibilité pour les liaisons horizontales

1. Sélectionner le bouton *Select Mode*
2. Positionner la souris sur une fonction graphique, presser sur le bouton gauche de la souris.
3. Glisser la fonction contre sa voisine en maintenant le bouton de la souris pressé.
4. Glisser la fonction vers sa position d'origine et lâcher le bouton de la souris.



Deuxième possibilité pour les liaisons avec changement de direction

1. Sélectionner le bouton *Auto Lines Mode*
2. Placer la souris sur le point de départ du lien, puis cliquer sur le bouton gauche la souris.
3. Placer la souris sur le point de destination du lien, puis cliquer sur le bouton gauche la souris.
4. En cas de besoin, il est possible de forcer le passage par des positions intermédiaires et d'interrompre l'édition de lien avec le bouton droite de la souris.



## Effacer un lien, une fonction, symbole ou un connecteur.

Sélectionner le bouton *Delete Mode* et sélectionner le lien, la fonction, le symbole ou le connecteur avec la souris.



## 1.4 Mise en service et test du programme

### 1.4.1 Construction d'un programme (*Build*)

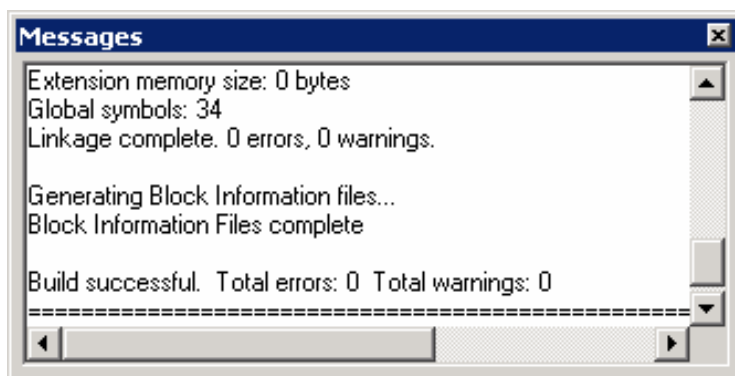


*Rebuild  
All Files*

Le programme utilisateur est complètement décrit, mais pas encore utilisable par le PCD. Il ne comprend pas le langage graphique de l'exemple décrit précédemment. Il est donc nécessaire de le traduire. C'est ce que fait l'outil de programmation lorsque l'utilisateur actionne le menu *CPU Rebuild All Files* ou le bouton *Rebuild All Files* du gestionnaire de projet ou de l'éditeur graphique.

La fenêtre des *Messages* nous renseigne sur le déroulement de la construction du programme. Nous remarquerons plusieurs étapes de construction: la compilation, l'assemblage et le linkage. Si le programme est correctement édité, la construction se termine par un message *Build successful. Total errors 0 Total warnings: 0*

Les éventuelles erreurs sont signalées par un message rouge. Un double clic de la souris sur ces messages permet de localiser l'erreur dans le programme d'application.



### 1.4.2 Charger le programme dans le PCD (*Download*)



*Download  
Program*

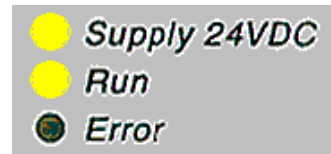
Le programme d'application est prêt. Il reste à le transférer de l'ordinateur vers le PCD avec le menu ou le bouton *Online Download Program* de la fenêtre *SAIA Project Manager*.

En cas de problème de communication, veuillez à nouveau contrôler vos configurations *Settings Online* et *Settings Hardware* ainsi que votre câble de communication entre l'ordinateur et le PCD. (PCD8.K111 ou USB)

### 1.4.3 Recherche des erreurs et vérification (*Debugging*)

Les programmes ne sont pas toujours parfaits dès la première version. Il est utile de les tester soigneusement. Le test d'un programme est supporté avec le même éditeur que celui employé lors de l'édition:

1. Se mettre en ligne avec le bouton *Go On /Offline*
2. Forcer l'automate en mode run avec le bouton *Run*



Parallèlement, observer la lampe *RUN* placée sur la face avant du PCD..



A la sélection du bouton *Run*, la lampe *RUN* est allumée, le PCD exécute le programme utilisateur.

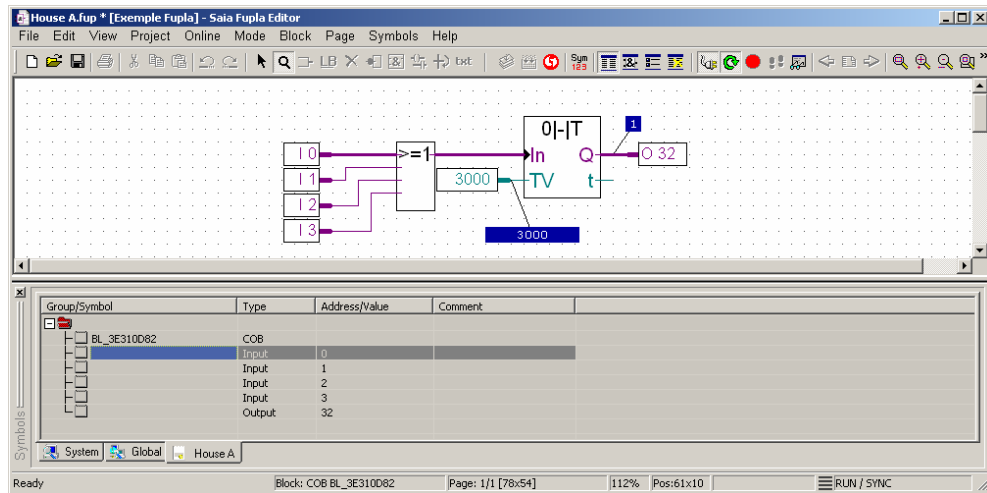


A la sélection du bouton *Stop*, la lampe *RUN* est éteinte, le PCD stop l'exécution du programme utilisateur.

Si l'éditeur est en ligne et le PCD en mode *RUN*, alors il est possible de représenter l'état de chacun des symboles employés par le programme:



- ♣ Les états logiques des informations binaires sont représentés par des liens larges ou fins. (large = 1 et fin = 0)
- ♣ Toutes les autres informations peuvent aussi être représentées par une sonde: sélectionner le bouton *Place Probe* puis le lien avec la souris.



## 1.5 Correction de programme

En cas de modification du programme veuillez procéder comme suit:

1. Quitter le mode en ligne (bouton *Go On /Offline* )
2. Editer la modification
3. Construire le projet (bouton *Build* )
4. Charger le programme dans le PCD (bouton *Download Program* )





## Table des matières

---

<b>TABLE DES MATIERES</b>	<b>1</b>
<b>2 GESTIONNAIRE DE PROJETS</b>	<b>3</b>
2.1 Au sommaire de ce chapitre	3
2.2 Organisation d'un projet	4
2.2.1 Exemple de projet d'application	4
2.2.2 Sauvegarde du projet sur le PC	6
2.2.3 Compresser un projet ou CPU	6
2.2.4 Ouvrir un projet	7
2.2.5 Créer un nouveau projet	7
2.3 La fenêtre <i>Project</i>	8
2.3.1 Dossier projet	8
2.3.2 Dossier commun	9
2.3.3 Dossier CPU	9
2.3.4 Settings Online	10
2.3.5 Raccorder le PC au PCD	11
2.3.6 Settings Hardware	12
2.3.7 Settings Software	17
2.3.8 Dossier Programme	19
2.3.9 Types de fichiers	20
2.3.10 Fichiers liés au projet	21
2.3.11 Fichiers Communs	21
2.4 Construction du programme. (Build)	22
2.4.1 <i>Rebuild all</i> et <i>Build</i>	23
2.4.2 Options du <i>Build</i>	23
2.5 Fenêtre des messages	24
2.6 Charger le programme dans le PCD (Download)	25
2.6.1 Options du <i>Download</i>	26
2.6.2 Charger le programme sur la mémoire backup (Flash)	27
2.6.3 Mémoire backup & transfert du programme d'application	27
2.7 Fenêtre View	27
2.7.1 Structure des blocs d'organisation	27
2.7.2 Liste des blocs d'organisation	28
2.7.3 Liste des symboles	28
2.7.4 Référence croisée	28
2.8 Backup du programme.	29
2.9 Self downloader	30
2.9.1 Préparer un fichier *.sd5	30
2.9.2 <i>Download</i> d'un fichier *.sd5	31



## **2      Gestionnaire de projets**

---

### **2.1    Au sommaire de ce chapitre**

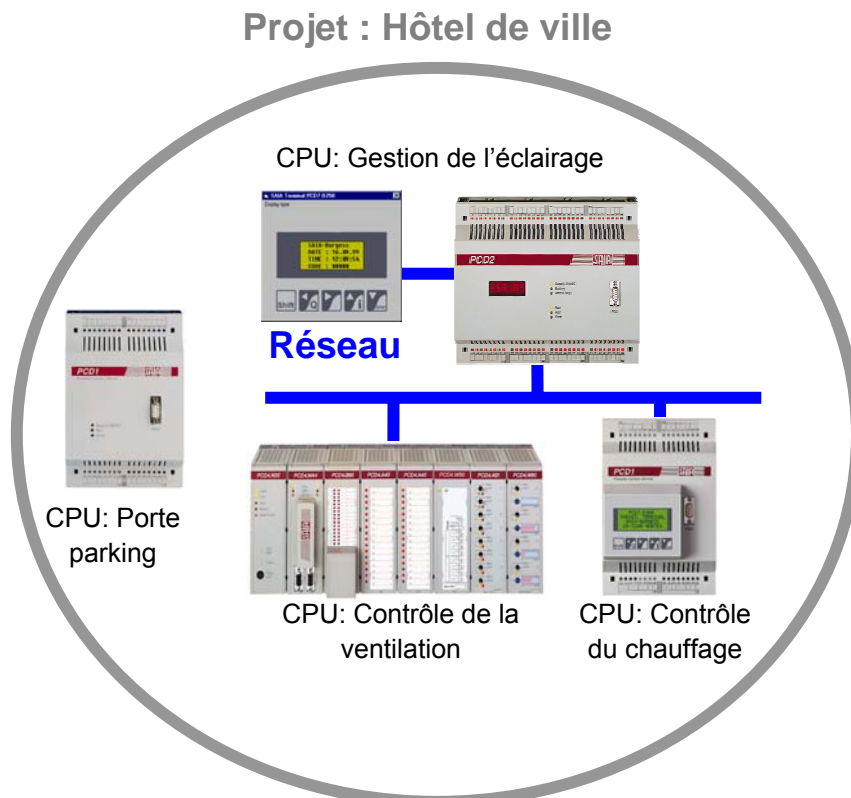
Les applications d'automatisation modernes sont souvent composées d'un grand nombre d'automates raccordés en réseaux. C'est pourquoi PG5 rassemble les programmes et configurations de tous les automates PCD d'une application dans un projet unique. Le gestionnaire de projet PG5 offre à l'utilisateur une vue globale de toutes les informations relatives à un projet.



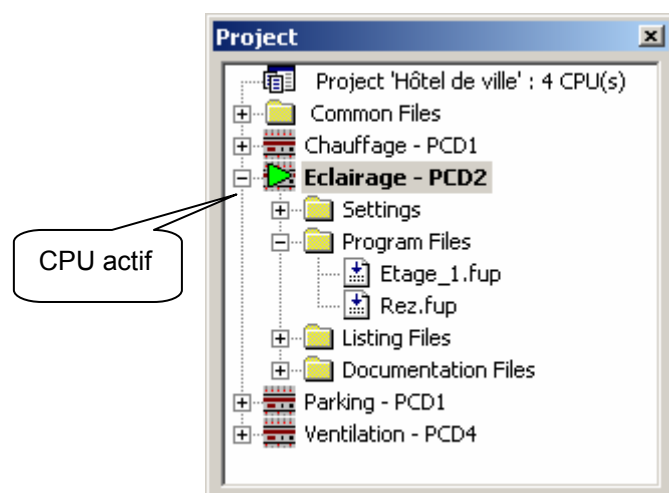
## 2.2 Organisation d'un projet

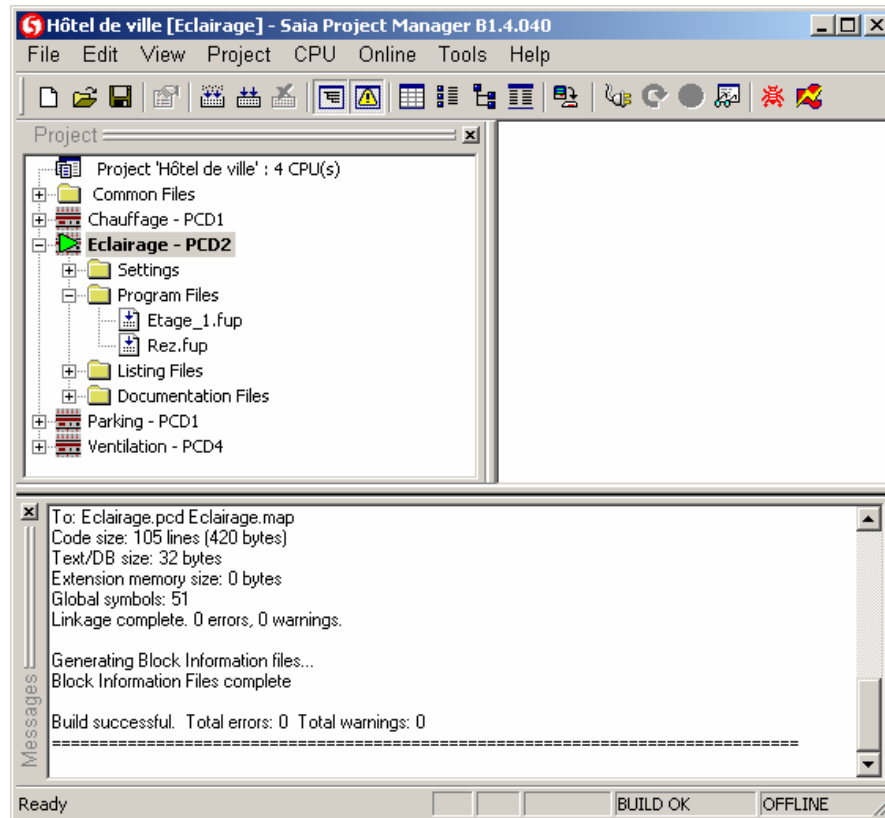
### 2.2.1 Exemple de projet d'application

Dans la pratique les installations automatisées sont presque toujours composées de plusieurs automates PCD décentralisés et raccordés par un réseau de communication. Chaque PCD supporte le contrôle d'une fonction particulière de l'ensemble du projet: gestion de l'éclairage, contrôle du chauffage, contrôle de la ventilation, la porte automatique du parking souterrain, ...



Le CPU actif est marqué par un triangle vert. Toutes les commandes pour construire, charger et tester le programme utilisent les configurations et fichiers programmes du CPU actif





Avec l'outil de programmation PG5, tous les CPU PCD appartenant à une même et unique application sont aussi réunis dans un projet PG5 unique.

Le gestionnaire de projet est composé de trois fenêtres :



*Project Tree*

La fenêtre *Project*, représente la structure du projet avec les CPU PCD qui le compose. Pour afficher cette fenêtre sélectionner le menu *View, Project Tree* ou le bouton *Project Tree*



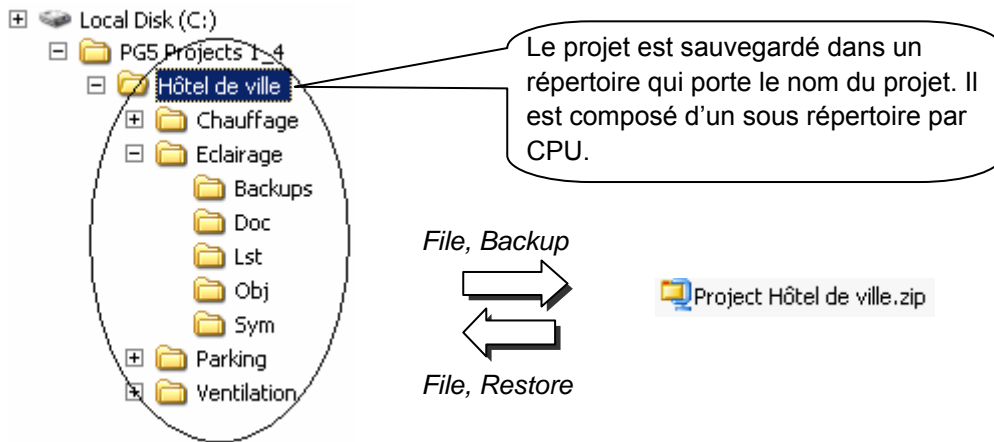
*Message Window*

La fenêtre *Message*, représente les messages d'alarme et d'erreur générés lors de la construction du programme. Pour afficher cette fenêtre sélectionner le menu *View, Message Window* ou le bouton *Message Window*.

La fenêtre *View* représente la liste des symboles et la structure de l'ensemble du programme. Elle permet aussi de réaliser une référence croisée des symboles.

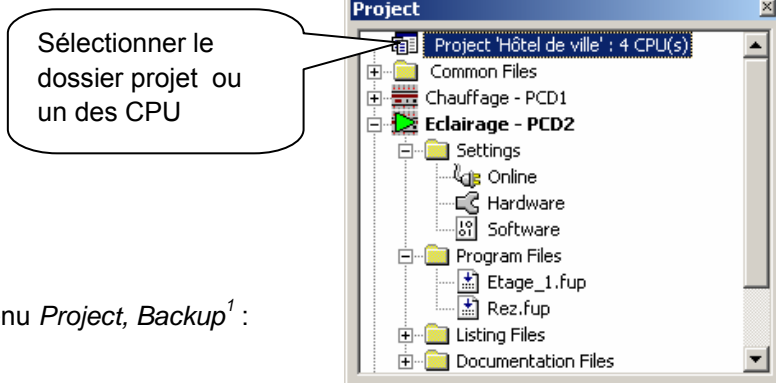
### 2.2.2 Sauvegarde du projet sur le PC

Par défaut, les projets sont sauvegardés dans le répertoire **C:\PG5 Projects 1\_4**

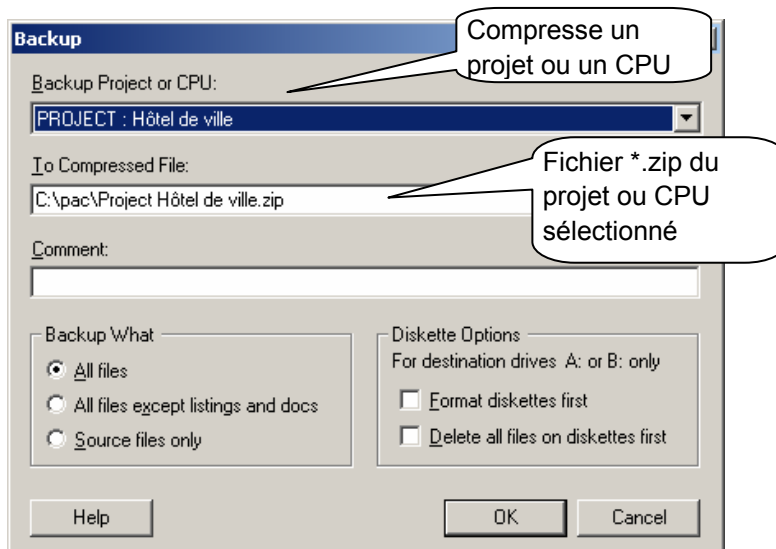


### 2.2.3 Compresser un projet ou CPU

Pour sauvegarder le projet, il est nécessaire de conserver toute la structure des répertoires avec les fichiers qu'ils contiennent. Le plus simple est de compresser toute la structure dans un fichier \*.ZIP



Sélectionner le menu *Project, Backup*<sup>1</sup> :

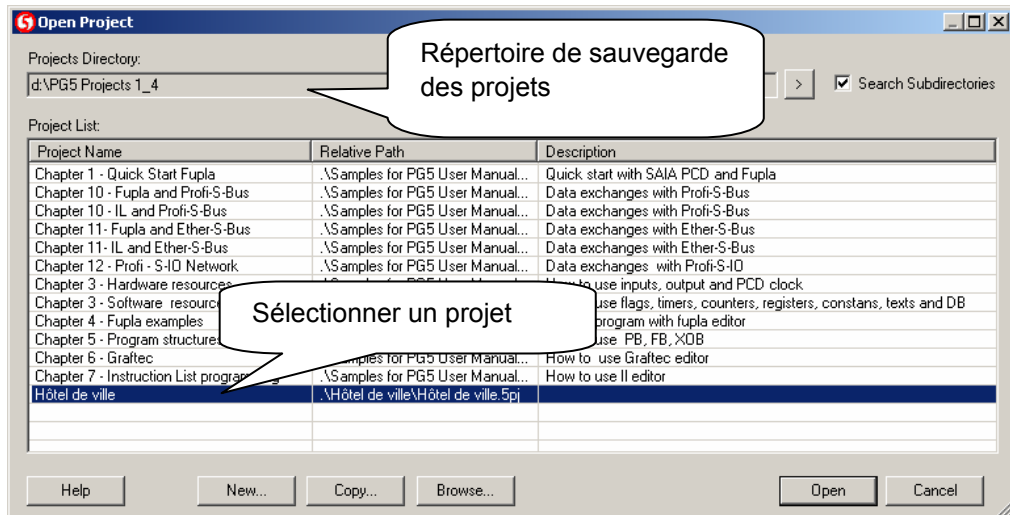


<sup>1</sup> Le menu *Project, Restore* permet de restaurer le projet/CPU sauvegardé dans un fichier \*.ZIP

## 2.2.4 Ouvrir un projet

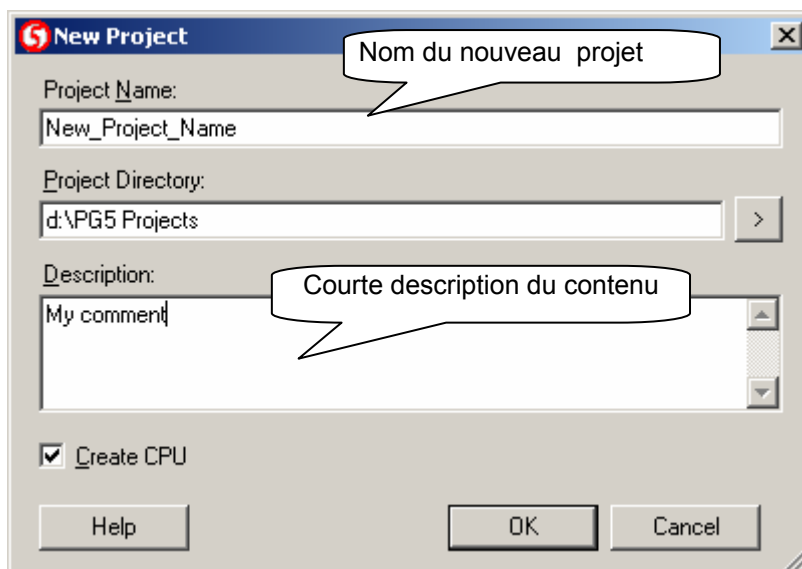
Le PG5 est livré avec tous les exemples de ce manuel. Le menu *Project, Open...* vous permet de les ouvrir et de les tester.

Un projet déjà présent peut être ouvert avec le menu *Project, Open...*, Il recherche tous les fichiers projet (.5pj) du répertoire des projets et les affiche dans une liste. Sélectionner l'un des projets de la liste d'un double clic de la souris ou marquer un projet de la liste et presser le bouton *Open*. Une autre alternative, presser le bouton *Browse* et chercher directement le fichier du projet ou CPU.

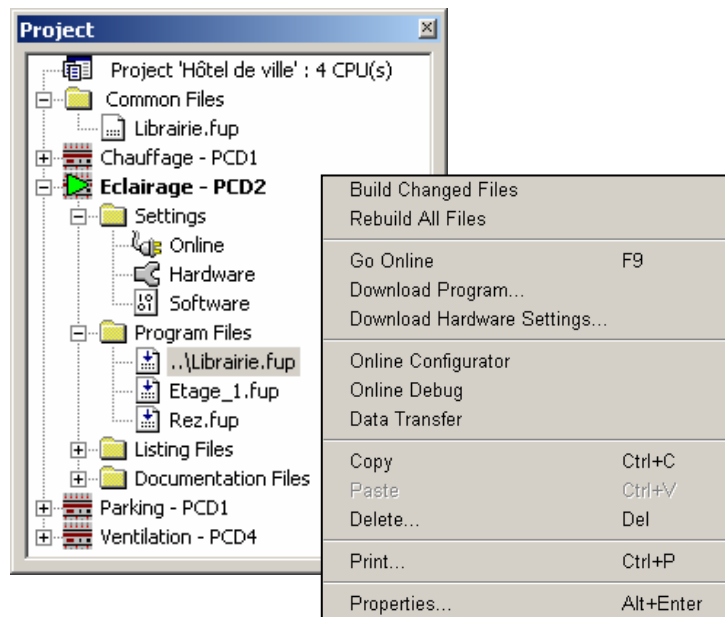


## 2.2.5 Créer un nouveau projet

Pour créer un nouveau projet, sélectionner le menu *Project, New...*, définir le nom du nouveau projet dans le champ *Project Name*, sélectionner l'option *Create CPU* et valider avec le bouton *OK*.



## 2.3 La fenêtre *Project*



Les dossiers de la fenêtre *Project*, rassemblent les informations du projet selon certains critères d'organisation:

### 2.3.1 Dossier projet



Le dossier principal représente le projet avec son nom et le nombre de CPU qui le constitue. Pour modifier les informations de ce dossier, marquer le dossier avec la souris et sélectionner le menu de contexte avec le bouton droit de la souris.

<i>New CPU...</i>	Permet d'ajouter un nouveau CPU au projet
<i>Import CPU...</i>	Permet d'importer les CPUs d'un autre projet PG5 ou d'un ancien projet PG4.
<i>Paste CPU</i>	Coller des CPUs
<i>Copy Project...</i>	
<i>Backup...</i>	Sauvegarde le projet vers un fichier *.zip
<i>Restore...</i>	Restaure le projet à partir d'un fichier *.zip
<i>Rebuild All CPUs</i>	Commandes pour effectuer le <i>Build</i> et le <i>Download</i> de tout le projet.
<i>Online Commands</i>	
<i>Print...</i>	
<i>Find...</i>	
<i>Properties</i>	Permet de modifier les informations relatives au dossier projet: nom du projet, description, ...

### 2.3.2 Dossier commun



Ce dossier est prévu pour recevoir les programmes communs à plusieurs CPU du projet.

Pour ajouter un fichier programme, marquer le dossier et sélectionner le menu de contexte *New File*.

Le menu de contexte *Add Files* permet d'importer tous les types de fichiers programmes PG5, mais aussi les documents de mise en service et d'entretien de l'application (Word, Excel, ...). Ces fichiers sont conservés avec le projet PG5.

Chaque CPU faisant usage d'un fichier commun utilise les symboles locaux définis par le fichier commun, par contre il utilise les symboles globaux de son propre CPU. Ainsi les symboles globaux utilisés dans le dossier commun peuvent être différents pour chaque CPU.

### 2.3.3 Dossier CPU



Chaque dossier CPU rassemble les configurations et les programmes d'un automate du projet. Pour modifier les informations d'un dossier CPU, marquer le dossier et sélectionner le menu de contexte.

*Set Active*



Commande pour rendre le CPU sélectionné dans la fenêtre *Project* actif. Le CPU actif est marqué par un triangle vert, toutes les commandes des menus et des boutons portent sur le CPU actif.

*Rebuild Changed files*

*Rebuild All Files*

*Go online*

*Download Program*

*Download hardware settings*

*Online Configurator*

*Online Debug*

*Data Transfert*

*Copy, Past*

*Delete*

*Print...*

Permet de copier/coller ou effacer le CPU sélectionné dans la fenêtre *Project*.

*Properties*

Permet de modifier les informations relatives au dossier CPU: nom du CPU, description, ...

Remarque :

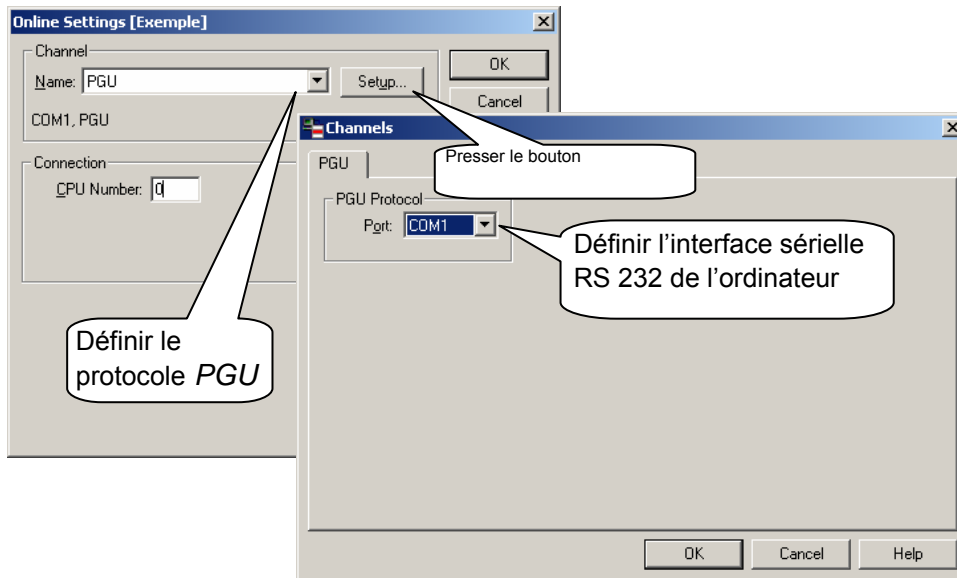
Si l'option suivante est sélectionnée: *Tools, Options, Activate CPU according to project tree location*. Le *SAIA Project Manager* active automatiquement le CPU selon le contexte d'utilisation. La commande *Set Active* n'est alors pas utile et pas représentée dans le menu de contexte.

### 2.3.4 Settings Online

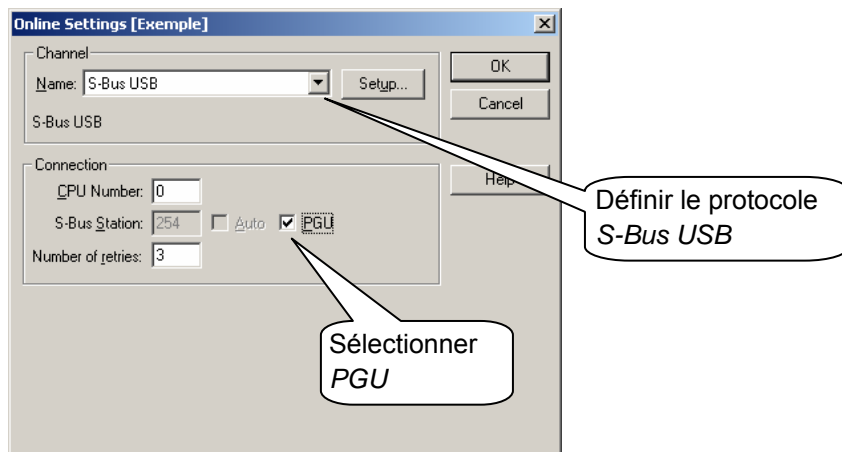


Le dossier *Settings Online* permet de définir les paramètres de communication pour l'outil de programmation PG5. Plusieurs protocoles de communications sont supportés: PGU, S-Bus, Ethernet,... Mais seul les protocoles *PGU* et *S-Bus USB* permettent de communiquer avec le PCD sans ajuster les *Settings Hardware* du PCD.

#### Channel PGU (RS 232)



#### Channel S-Bus USB



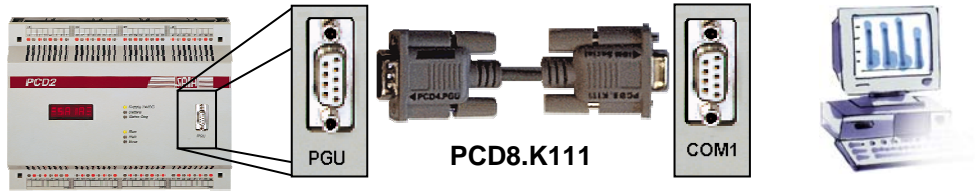
Remarque:

L'interface USB n'est disponible que sur les nouveaux CPU PCD2.M480 et PCD3.

### 2.3.5 Raccorder le PC au PCD

#### Channel PGU (RS 232)

Le câble PCD8.K111 supporte la liaison RS 232 entre le PC et le PCD. Pour de plus amples informations concernant ce câble, veuillez vous référer à votre manuel hardware PCD.



#### Channel S-Bus USB



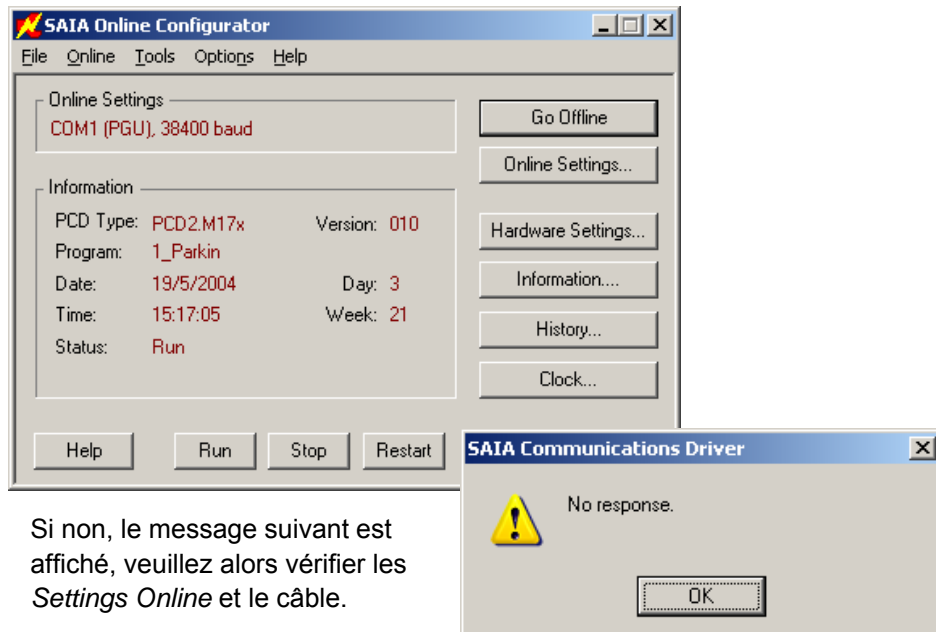
L'interface USB n'est disponible que pour les nouveaux CPU PCD2.M480 et PCD3



Online Configurator

#### Vérification du raccordement

Le bouton *Online Configurator* permet d'afficher les informations relatives au PCD. Si les informations représentées en rouge sont présentes dans la fenêtre, alors la communication fonctionne parfaitement bien.



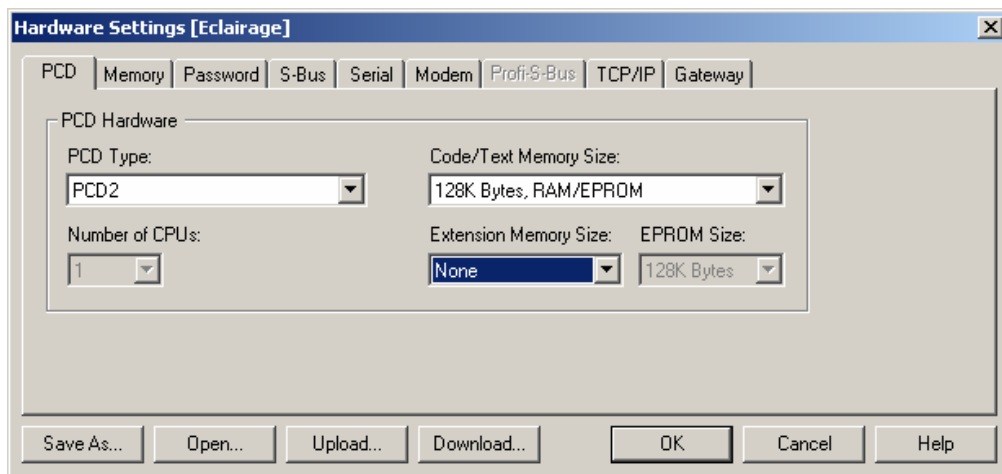
Si non, le message suivant est affiché, veuillez alors vérifier les *Settings Online* et le câble.



### 2.3.6 Settings Hardware



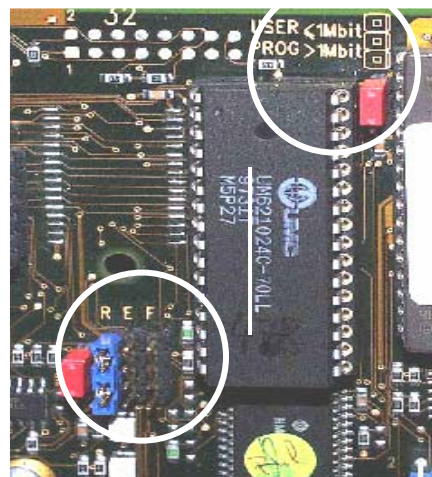
Le dossier *Settings Hardware* permet de définir les paramètres mémoire de communication pour l'automate PCD.



La configuration de la mémoire est nécessaire lors de la première utilisation de l'automate ou après une extension de la capacité mémoire du PCD. Deux solutions permettent de sélectionner les paramètres de la fenêtre ci-dessus:

- La première consiste à sélectionner le bouton *Upload* pour lire directement les données dans l'automate.
- La seconde consiste à définir les informations de la fenêtre avec l'aide des deux tableaux qui suivent. L'exemple ci-dessus correspond aux lignes marquées en gras des tableaux ci-dessous

Le tableau ci-dessous donne aussi des informations concernant le positionnement des cavaliers mémoires. Ces cavaliers sont à ajuster sur le CPU PCD. Pour de plus amples informations, veuillez vous référer à votre manuel Hardware.



Cavaliers mémoire d'un PCD2.M120 équipé d'une mémoire LM621024

PCD Type	Mémoire	N° de commande	Code/Text Memory Size	Extension Memory Size (RAM)	Cavaliers sur le PCD
PCS1.C8	Flash 2 MBit		240 kByte	128 kByte	Aucuns
	Flash 4 MBit		<sup>2)</sup> 1008 kByte	896 kByte	Aucuns
PCD1. M110 /120/150	Emplacement vide		17 kByte	Aucuns	R
	1 RAM 256 kBit	4 502 5414 0	32 kByte	13 kByte	R
	1 RAM 1MBit	4 502 7013 0	128 kByte	13 kByte	R
	1 Flash 1MBit	4 502 7141 0	112 kByte	13 kByte	E
	1 EPROM 512kBit	4 502 3958 0	64 kByte	13 kByte	E
	1 EPROM 1Mbit	4 502 7126 0	128 kByte	13 kByte	E
PCD2.M110 /120/150	Emplacement vide		32/128 kByte <sup>1)</sup>	Aucuns	R , <=1Mbit
	1 RAM 256 kBit	4 502 5414 0	32 kByte	24/128 kByte <sup>1)</sup>	R , <=1Mbit
	<b>1 RAM 1 Mbit</b>	<b>4 502 7013 0</b>	<b>128 kByte</b>	<b>24/128 kByte <sup>1)</sup></b>	<b>R , &lt;=1Mbit</b>
	1 RAM 4 MBits	4 502 7175 0	512 kByte	24/128 kByte <sup>1)</sup>	R , >1Mbit
	1 Flash 1 MBit	4 502 7175 0	112 kByte	24/128 kByte <sup>1)</sup>	F , <=1Mbit
	1 Flash 4 MBit	4 502 7141 0	448 kByte	24/128 kByte <sup>1)</sup>	F , >1Mbit
	1 EPROM 512 kBit	4 502 7224 0	64 kByte	24/128 kByte <sup>1)</sup>	E , <=1Mbit
	1 EPROM 1 MBit	4 502 3958 0		24/128 kByte <sup>1)</sup>	
	1 EPROM 4 MBit	<b>4 502 7126 0</b>	<b>128 kByte</b>	24/128 kByte <sup>1)</sup>	<b>E , &lt;=1Mbit</b>
		4 502 7223 0	512 kByte		E , >1Mbit
PCD4.M	2 RAM 62256	4 502 5414 0	64 kByte		RAM
	2 RAM 1 Mbit	4 502 7013 0	256 kByte	172 kByte,	RAM
	2 EPROM 256 kBit	4 502 5327 0	64 kByte	pour la	E256
	2 EPROM 512 kBit	4 502 3958 0	128 kByte	mémoire	E512
	2 EPROM 1 MBit	4 502 7126 0	256 kByte	PCD7.R310	E1M

<sup>1)</sup> 128 kByte pour tous les PCD2.M110/120 de version hardware plus grande ou égale à J, 128 kByte pour tous les PCD2.M150

<sup>2)</sup> 1008 kByte pour tous les PCS1de version hardware plus grande ou égale à E

Cavaliers: R= RAM, E = EPROM, F = Flash

Si la mémoire montée sur le PCD vous est inconnue, relevez la référence inscrite sur la mémoire et déterminez le numéro de commande ou la taille mémoire avec la table ci-dessous:

Taille mémoire	Numéro de commande	Référence
RAM 256 kBit	4 502 5414 0	SRM 2B256SLCX70 HY62256ALP-70 GM76C256CLL-70 M5M5256DP-70LL TC55257DPL-70L
<b>RAM 1 MBit</b>	<b>4 502 7013 0</b>	BS62LV1025 PC-70 <b>LP621024D-70LL</b> SRM20100LLC70 HY628100ALP-70 GM76C8128CLL-70 M5M51008BP-70L TC551001BLP-70L
RAM 4 MBit	4 502 7175 0	BS62LV4006PC P55 BS62LV4007PC P55 HM628512LP-5 KM684000ALP-5L KM684000BLP-5L
Flash 1 MBit	4 502 7141 0	AM29F010-70PC
Flash 4 MBit	4 502 7224 0	AM29F040 auf Sockel
EPROM 256 kBit	4 502 5327 0	UPD27C256AD-10 M27C256B-10F1 TMS27C256-10JL
EPROM 512 kBit	4 502 3958 0	AM27C512-15XF1 AMC27C512-15XF1 AM27C512-90DC UPD27C512D-10 M27C512-10XF1 M27C512-10F1
EPROM 1 MBit	4 502 7126 0	AM27C010-90DC NM27C010Q-90 M27C1001-10F1
EPROM 4 MBit	4 502 7223 0	AM27C040-100DC M27C4001-10F1

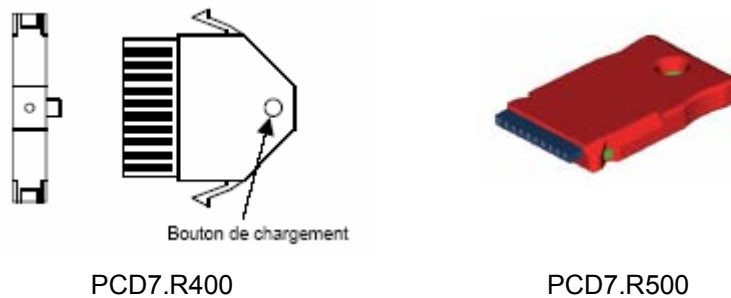
PCD Type	Code/Text/ Extension Memory Size (RAM)	Mémoire backup interne (Flash )	Mémoire backup externe (Flash )
PCD4.M170 PCD2.M170 PCD2.M480	1024 kByte	Aucune	PCD7.R400 - 1024 kByte
PCD3.M3020P CD3.M3120	128 kByte	128 kByte	PCD3.R500 - 128 kByte
PCD3.M3230 PCD3.M3330	256 kByte	256 kByte	PCD3.R500 - 256 kByte
PCD3.M5440 PCD3.M5540 PCD3.M6340 PCD3.M6540	512 kByte	256 kByte	PCD7.R500 - 512 kByte

Les nouveaux systèmes PCD supportent une mémoire backup interne ou externe PCD7.R400/R500 (facultative).

La mémoire backup permet de sauvegarder une copie du programme d'application en RAM (*Code/Text/Extension*) sur une mémoire Flash dont le contenu n'est pas perdu lors de coupure de courant ou de batterie défectueuse.

Nous vous recommandons de faire usage de la mémoire backup de vos PCD pour vous protéger de toute perte de données indésirable.

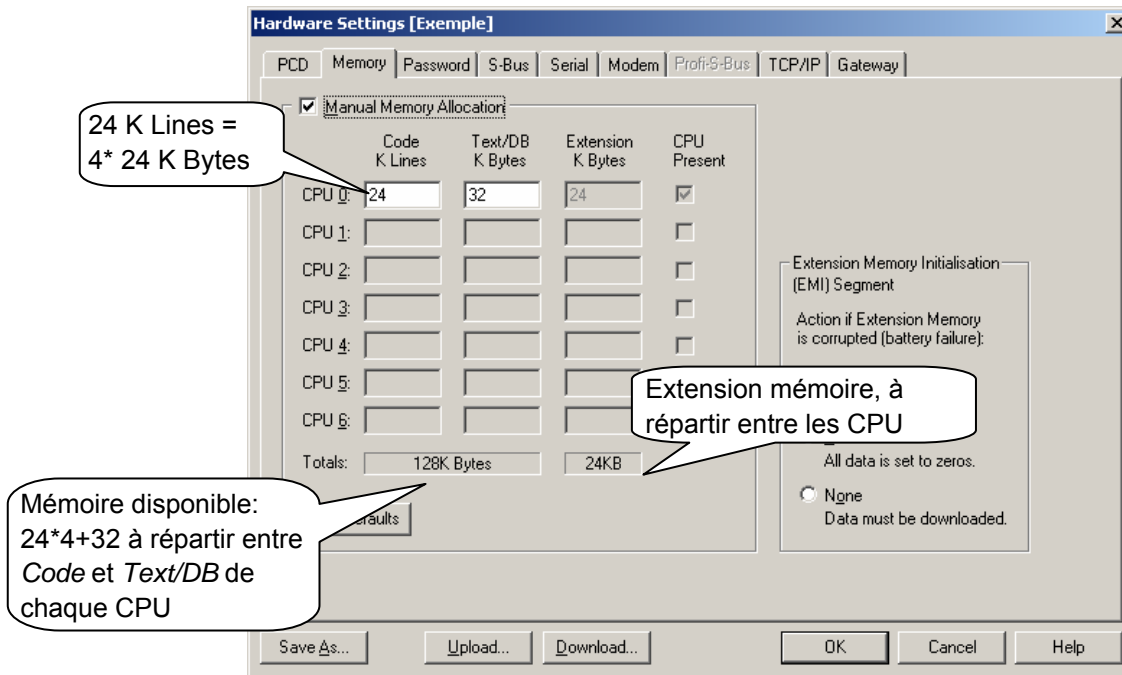
Si le programme d'application en RAM (*Code/Text/Extension*) est altéré, le démarrage à froid du PCD restore automatiquement le programme en mémoire backup.



La mémoire backup externe vous permet aussi de transférer des applications d'un automate à un autre et de créer une sauvegarde des textes et DBs RAM dans la mémoire d'extension pendant l'exécution de l'automate (adresse ≥ 4000).

Remarque :

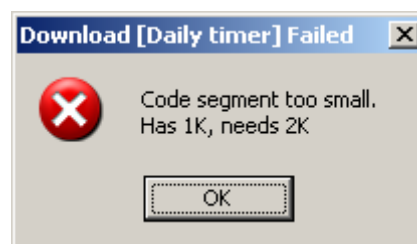
Malgré la sauvegarde en mémoire backup, les fichiers sources du projet doivent toujours être conservés car l'application n'est sauvegardée dans le PCD que sous forme de code machine.



La mémoire disponible définie sous le répertoire PCD est répartie entre le code programme et les textes de chaque CPU. Certains PCD comptent plusieurs CPU: PCD4.M44x et PCD6.Mxxx.

Pour les automates avec un seul CPU, la répartition mémoire est automatiquement adaptée au programme de l'utilisateur. *Manual Memory Allocation* peut être désactivé.

Les paramètres par défaut sont satisfaisants pour la majorité des applications. Excepté si le chargement du programme dans le PCD affiche une erreur du genre:



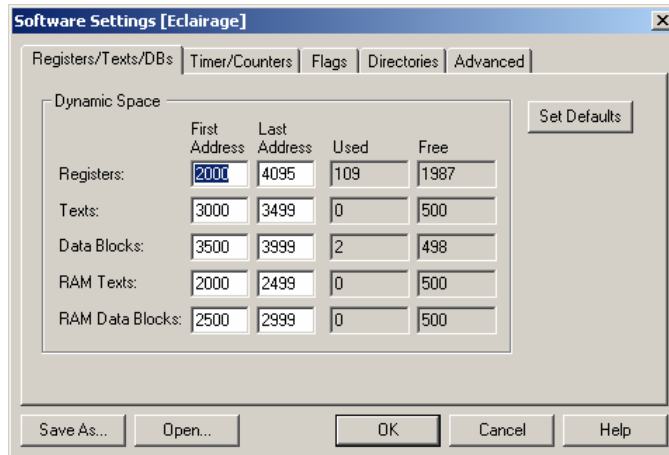
Trois solutions permettent d'éviter cette erreur :

- Désactiver l'option *Manual Memory Allocation*. Si la mémoire est suffisante, PG5 supporte automatiquement la répartition mémoire.
- Adapter la répartition de la mémoire selon le texte affiché.
- Augmenter la capacité mémoire du PCD.



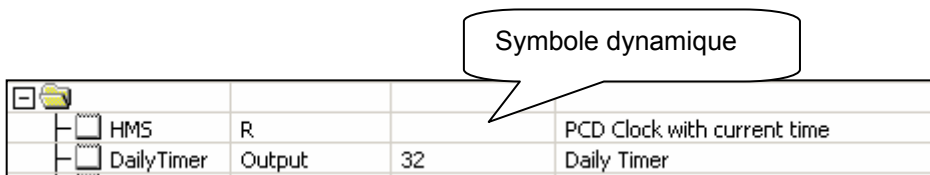
Après avoir sélectionné les *settings hardware*, ne pas oublier de les charger dans la mémoire du PCD avec le bouton *Download*.

### 2.3.7 Settings Software



Cette fenêtre supporte la réservation d'une plage d'adresses pour les registres, compteurs, temporisateurs et indicateurs dynamiques. A la construction du programme, ces adresses sont automatiquement attribuées aux symboles dynamiques définis par le programme utilisateur et les Fbox Fupla.

Un symbole est dynamique lorsque l'adresse du symbole n'est pas définie:



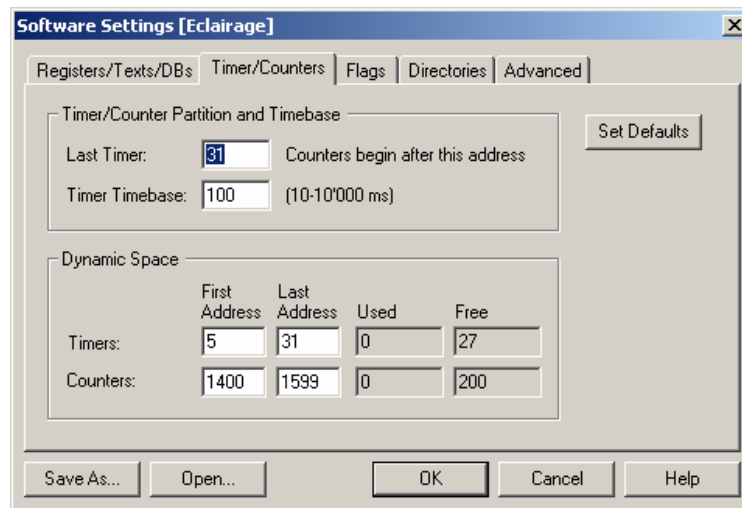
La configuration des adresses dynamiques n'est pas indispensable. Les paramètres par défaut sont satisfaisants pour la majorité des applications.

Excepté si la construction d'un long programme affiche une erreur du genre:

**Fatal Error 368: Auto-allocation/dynamic space overflow for type: R**

Alors il est nécessaire d'étendre la plage des symboles dynamiques mentionnés par le message.

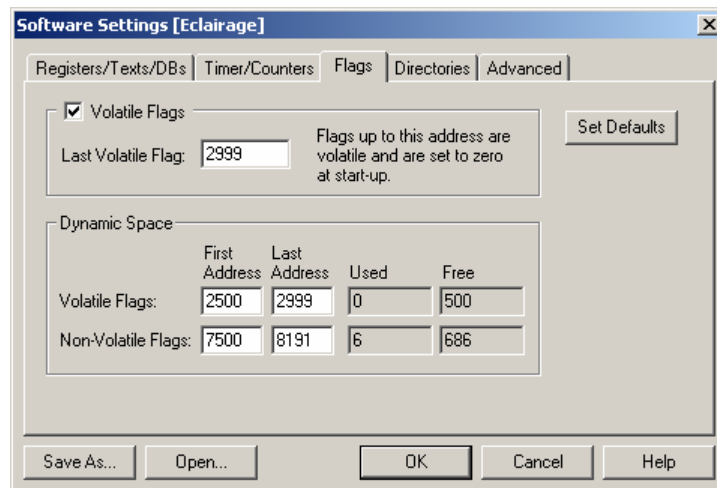
Si l'automate est équipé d'une mémoire EPROM ou Flasch, les plages dynamiques *RAM Texts* et *RAM Data Block* doivent aussi être ajustées avec des adresses supérieures ou égales à 4000 pour que ces informations se trouvent dans une plage de mémoire RAM.



Les PCD sont configurés avec 31 temporisateurs auxquels sont automatiquement retranchés une plage pour les symboles dynamiques. Avec certains programmes, il peut être nécessaire d'augmenter le nombre de temporisateurs.

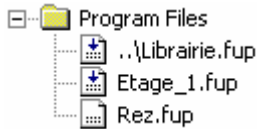
La base de temps pour la décrémentation des temporisateurs est de 0.1 secondes. Si nécessaire, il est possible l'ajuster avec une autre valeur. Attention la base de temps est sans influence sur les programmes Fupla. Seul les programmes IL sont affectés par ce paramètre.

Nous recommandons de ne pas définir un nombre de temporisateur inutilement grand ou une base de temps inutilement petite. Cela contribue à accélérer le temps de cycle des programmes.



Par défaut, tous les indicateurs sont non volatils. Si nécessaire le paramètre *Last Volatile Flag* permet de définir une plage volatile. (Cet exemple définit des indicateurs volatils pour les adresses F 0 à F 2999 volatils)

### 2.3.8 Dossier Programme



Ce dossier est prévu pour recevoir tous les fichiers qui constituent le programme d'un PCD. Pour modifier les fichiers d'un dossier CPU, marquer le dossier ou le fichier et sélectionner le menu de contexte.

Open	Enter	
New...	Ctrl+N	
Add Files...		
Cut	Ctrl+X	Permet de réaliser un copier, coller d'un fichier dans un autre CPU du projet
Copy	Ctrl+C	
Paste	Ctrl+V	
Delete...	Del	
Linked	Ctrl+L	Lie le fichier au projet
Properties...	Alt+Enter	

Ajoute un nouveau fichier programme

Importe un fichier depuis un autre projet ou CPU

Modifie les informations relatives au fichier (Nom, description)

Si un nouveau fichier est ajouté au dossier programme, définir le nom et le type du fichier.

File Name: Untitled1

Directory: d:\PG5 Projects 1\_4\Hôtel de ville\Eclairage

File Type: Fupla Files (\*.fup)

Description:

Linked/Built  Open file now

Buttons: Help, OK, Cancel

Nom du fichier

Définir le type de l'éditeur selon la liste ci-dessous

Coute description du contenu



### 2.3.9 Types de fichiers

Un CPU peut supporter plusieurs fichiers programmes de types différents. A chaque type de fichier correspond un éditeur spécifique à un domaine d'application.

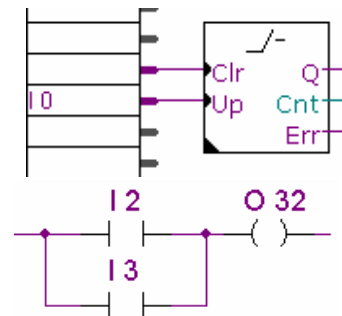
#### L'éditeur de listes d'instructions (\*.src)

Permet de programmer sous la forme d'un texte avec un jeu de 127 instructions. Convient pour toutes les applications, mais demande une certaine expérience de programmation.

```
COB 0
      0
STH  I 0
DYN  F 9
INC  C 53
ECOB
```

#### L'éditeur Fupla (\*.fup)

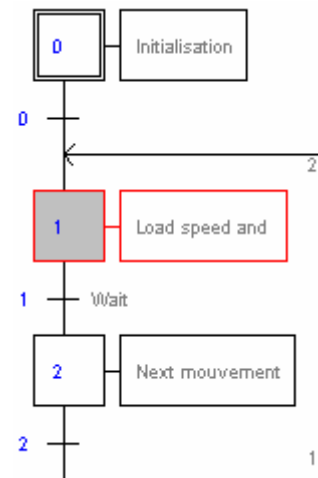
Permet dessiner les programmes sous la forme de plans de fonctions et de schémas à contacts. Ne nécessite aucune expérience de programmation. De nombreuses bibliothèques sont à disposition pour mettre rapidement en service les applications CVC et les réseaux de communication modem, Lon, Belimo, EIB, ...



#### L'éditeur Graftec (\*.sfc)

C'est un outil de structuration pour les programmes en liste d'instructions et Fupla. Convient particulièrement bien aux applications séquentielles avec des attentes sur des événements internes ou externes au PCD.

C'est l'outil idéal pour programmer des machines avec des commandes de moteurs, vérins, ...



#### L'éditeur HMI

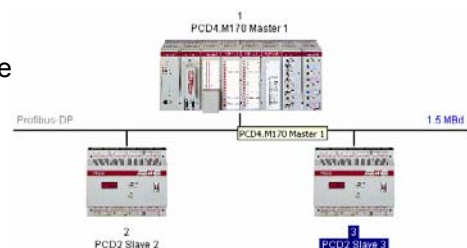
Permet de configurer le dialogue avec les terminaux PCD7.D1xx et PCD7.D2xx

( A installer en complément du PG5)

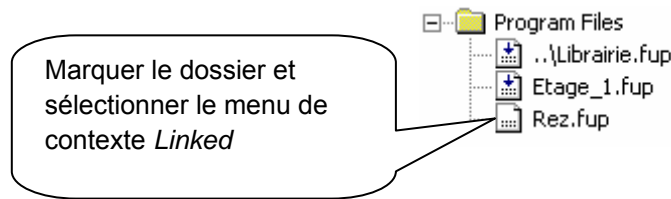


#### L'éditeur Snet (\*.DP, \*.LON, \*.SRIO)

Supporte la configuration des réseaux de communication Profibus DP, LON et SRIO.



### 2.3.10 Fichiers liés au projet

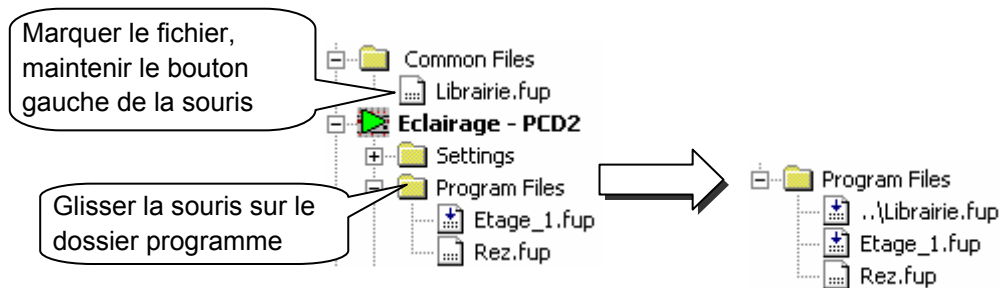


Les fichiers représentés par cet icône avec une flèche sont liés au projet. Ces fichiers font partie intégrante du programme et sont chargés dans la mémoire du PCD.



Les fichiers représentés par cet icône sans flèche ne sont pas liés au projet. Ces fichiers sont ignorés et ne sont pas chargés dans la mémoire du PCD. Cela peut être utile pour des programmes liés lors des tests de mise en service et qui ne sont pas présents dans le programme final.

### 2.3.11 Fichiers Communs



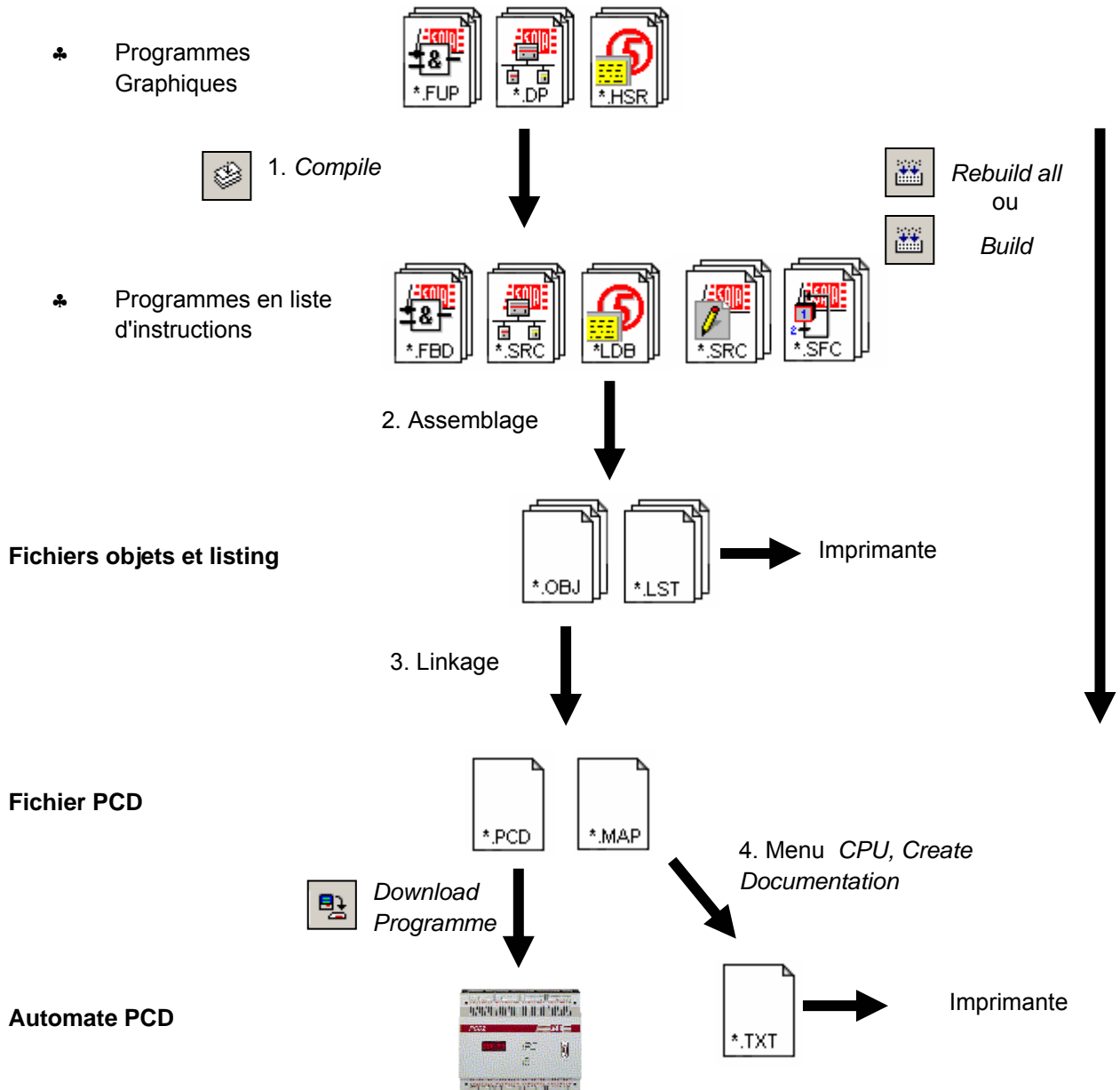
Les fichiers présents dans le dossier commun peuvent être copiés, collés ou simplement glissés dans le dossier programme des CPU qui les utilisent. Remarquez le nom du fichier copié ou glissé commence par deux points. Cela signifie que les informations sont disponibles depuis un dossier de niveau supérieur.

Le fichier peut être édité à partir du dossier commun ou du dossier programme d'un CPU. Dans les deux cas, l'utilisateur modifie le même fichier et les corrections sont valables pour tous les CPU qui y sont liés.

## 2.4 Construction du programme. (Build)

Le PCD ne peut pas directement recevoir les programmes préparés avec les éditeurs Fupla, IL, Graftec, Snet ou HMI. Ces fichiers doivent être préparés selon les différentes étapes présentées par ce graphique:

### Fichiers sources :



1. La compilation convertit les fichiers graphiques en liste d'instruction (\*.FBD, \*.SRC, \*.LDB)
2. L'assemblage produit des fichiers binaires de type (\*.OBJ) et un rapport d'assemblage (\*.LST) pour l'impression ou pour rechercher la raison de certaines erreurs d'assemblage.
3. Le linkage, chaîne les fichier (\*.OBJ) pour former un seul fichier (\*.PCD) destiné à être chargé dans le l'automate.
4. Une documentation peut être générée avec le menu *Create Documentation* du gestionnaire de projet, le résultat est disponible dans le dossier *Documentations Files*

### 2.4.1 **Rebuild all et Build**



Rebuild  
all

Le bouton, menu *CPU*, *Rebuild all Files* supportent la compilation, l'assemblage et le linkage de tous les fichiers liés au CPU actif.

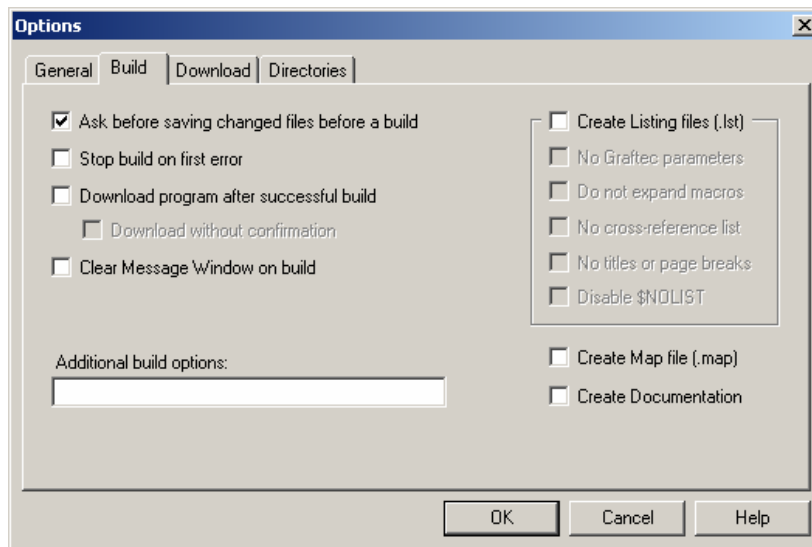


Build

Le bouton, menu *CPU*, *Build changed Files* effectue la même fonction mais uniquement sur les fichiers modifiés depuis le dernier *Build changed Files* ou *Rebuild all Files*. Cela permet de gagner un peu de temps avec la construction de gros programmes.

### 2.4.2 **Options du Build**

Il est encore possible de faire d'avantage en adaptant les paramètres de construction. Différentes options sont définissables avec le menu *Tools, Option*:



#### **Ask before saving changed files before a build**

Si ce paramètre est sélectionné, PG5 demande l'autorisation de sauvegarder les fichiers sources dont le contenu n'a pas encore été mémorisé avant de construire le programme. Si non les fichiers sont automatiquement sauvegardés.

#### **Stop build on first error**

La sélection de cette option met fin à la construction après la première erreur présente dans la fenêtre Messages.

#### **Download program after successful build**

La sélection de cette option charge automatiquement le programme dans le PCD. Mais seulement si la construction se termine sans erreur.

#### **Download without confirmation**

Normalement le chargement du programme dans la mémoire du PCD commence avec un dialogue de mise en garde que l'utilisateur quitte avec le bouton OK. La sélection de cette option charge directement le programme sans afficher ce dialogue.

#### **Clear Message Window on build**

Le contenu de la fenêtre Messages sera effacé au début de chaque construction.

#### **Create listing file**

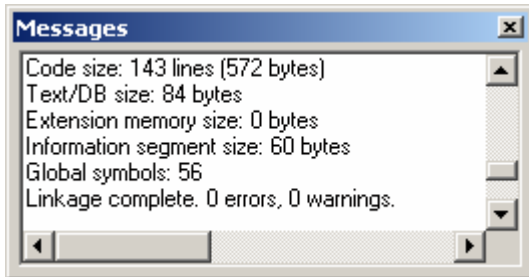
Crée un rapport d'assemblage (\*.LST)

#### **Create map file**

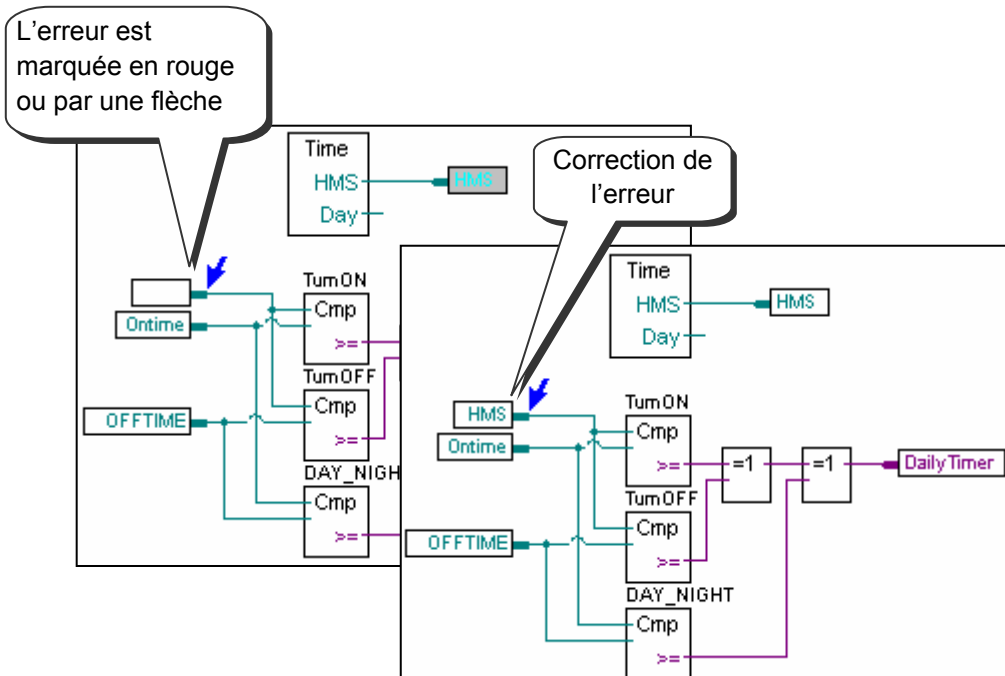
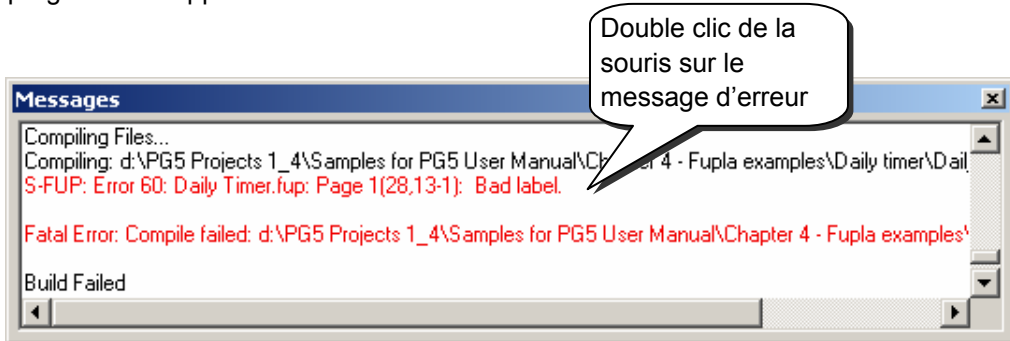
Crée un fichier avec l'espace mémoire nécessaire à l'application.

### 2.5 Fenêtre des messages

La fenêtre des Messages nous renseigne sur le déroulement de la construction du programme. Nous remarquerons plusieurs étapes de construction: la compilations, l'assemblage et le linkage. Si le programme est correctement édité, la construction se termine par un message *Build sucessful. Total errors 0 Total warnings: 0*



Les éventuelles erreurs sont signalées par un message rouge. Un double clic de la souris sur ces messages permet généralement de localiser l'erreur dans le programme d'application.

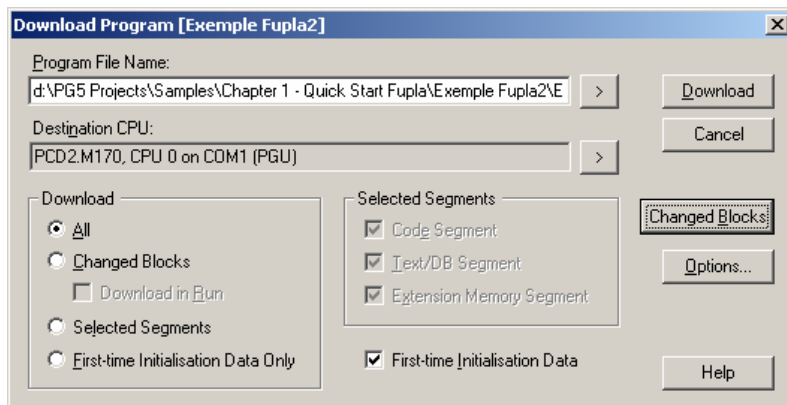


## 2.6 Charger le programme dans le PCD (Download)



Download Program

Si la construction se termine sans message d'erreur, le bouton *Download Program* ou le menu *Online, Download Program* permet de placer le programme dans la mémoire de l'automate PCD.



### Program File Name

Indique par défaut, le nom du fichier programme correspondant au CPU actif.

### All

Charge tout le programme (*Code Segment, Text/DB Segment, Extension Memory Segment*)

### Changed Blocks

Charge seulement les blocs (COB, PB, FB, SB, ST, TR, XOB) modifiés depuis le dernier Download. Cette option n'est utilisée que pour gagner un peu de temps avec les petites corrections de programme. Le bouton *Changed Blocks* permet d'afficher la liste des blocs modifiés.

### Download in Run

Permet de charger les blocs de programme modifiés sans stopper l'exécution du programme. Le bon fonctionnement de cette option peu être dépendante des corrections apportées au programme.

### Selected Segments

Ne charge que les segments définis sous *Selected Segments* :

*Code Segment* = Programme, *Text/DB Segment* = Textes et DB 0...3999, *Extension Memory Segment* = Textes et DB >3999.

### First-time Initialisation Data Only

Ne charge que les données décrites ci-dessous.

### First-time Initialisation Data

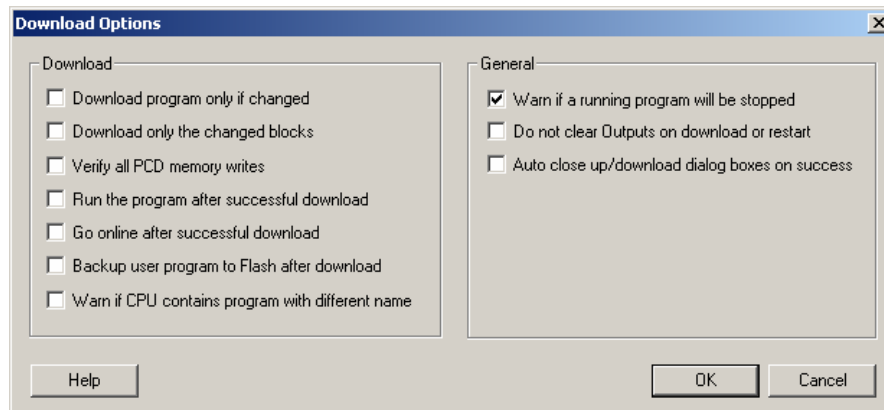
Cette option autorise l'initialisation de certaines données lors du chargement du programme. Les données initialisées par le chargement du programme sont définies comme suit : [Symbole] [Type] [Adresse]= [Valeur d'initialisation]

Group/Symbol	Type	Address/Value	Comment
Symbol0	R	10:= 314	First time initialisation value = 314
Symbol1	R	11	

Les données qui ne sont pas initialisées au chargement du programme peuvent être initialisées à chaque démarrage à froid par un bloc XOB16.

## 2.6.1 Options du *Download*

Différents paramètres sont définissables avec le menu *Tools*, *Option* ou le bouton *Option*. Ils permettent de personnaliser la procédure de chargement du programme.



### ***Download program only if changed***

Ne charge le programme que s'il est différent de celui déjà présent dans la mémoire du PCD.

### ***Download only the changed Blocks et Download in Run***

Voir page précédente

### ***Verify all PCD memory writes***

Toutes les données écrites dans le PCD sont lues en retour pour les comparer. Cette option ne devrait pas être sélectionnée, car elle contribue à allonger le temps de chargement du programme.

### ***Run the program after successful download***

Met automatiquement le CPU en *Run* après le chargement du programme. Attention, cette option ne doit être sélectionnée que si le programme fonctionne correctement et que cela ne met pas en danger des personnes ou installations.

### ***Go online after successful download***

Met automatiquement le CPU en ligne après le chargement.

### ***Backup user program to Flash after download***

Copie automatiquement le programme sur la mémoire backup Flash<sup>1</sup>.

Si cette option n'est pas activée, il est toujours possible de le réaliser après le download en sélectionnant le menu *Online*, *Flash Backup/Restore*.

### ***Warn if CPU contains program with different name***

Compare le nom du programme encore présent dans le PCD avec celui du programme en cours de chargement. Si les noms de ces programmes sont différents, un message est affiché pour prévenir le *Download* dans un mauvais CPU.

### ***Warn if a running program will be stopped***

Le chargement d'un programme peut stopper le PCD. La sélection de cette option permet d'afficher un message d'alarme avant de stopper le PCD.

### ***Do not clear Outputs on download or restart***

Cette option peut être utile avec les applications CVC. Cela évite de stopper une ventilation, d'éteindre la lumière durant le chargement d'un programme. Ne devrait pas être sélectionnés avec les autres applications.

### ***Auto close Up/Download dialog boxes on success***

Si cette option est sélectionnée, les dialogues pour le *Up/Download* ne sont affichés qu'en cas d'erreur.

1) PCD2.M170, PCD2.M480, PCD4.M170 et PCD3

## 2.6.2 Charger le programme sur la mémoire backup (Flash)

Si votre PCD est équipé d'une mémoire backup Flash<sup>1</sup>, le menu *Online, Flash Backup/Restore* permet de copier le programme chargé dans la mémoire RAM du PCD vers la carte Flash et inversement. Cette copie peut être supportée automatiquement en sélectionnant l'option *download* adéquate.

## 2.6.3 Mémoire backup & transfert du programme d'application

La mémoire backup permet de transférer le programme d'application d'un PCD vers un autre de même modèle:

- Charger le programme sur la mémoire backup.
- Couper l'alimentation du PCD pour détacher la mémoire backup.
- Mettre le PCD hors tension avant d'enficher la mémoire backup.
- Retirer la pile ou presser 3 secondes sur le bouton de la mémoire backup (PCD7.R400 seulement)
- Mettre le PCD sous tension. Les voyants clignotent pendant que le programme d'application est restauré à partir de la mémoire backup.
- Insérer le module de pile pour éviter le message d'erreur *battery fail*

## 2.7 Fenêtre View

Les informations affichées par cette fenêtre ne sont disponibles que si la construction du programme se termine sans erreurs.

### 2.7.1 Structure des blocs d'organisation

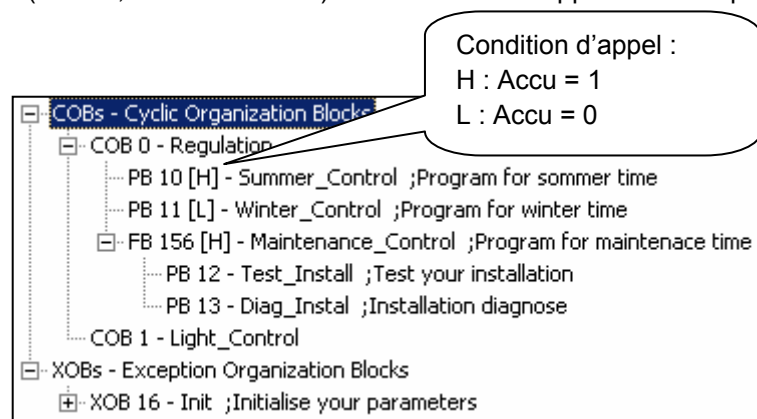
Le langage de programmation SAIA PCD est structuré avec différents blocs d'organisation dans lesquels l'utilisateur dépose les programmes de son application.

Chaque bloc offre un service particulier: programme cyclique (COB), programme séquentiel (SB), sous programme (PB), fonction paramétrable (FB), routine d'exception (XOB).

Après la construction du programme, le bouton *Block Structure view* ou le menu *View block Structure* représente la structure de l'ensemble des blocs d'organisation qui constituent programme.

L'exemple ci-dessous présente un programme constitué des blocs d'organisation COB 0, COB 1, XOB16 PB 10, PB11 et FB 156.

Remarquons que la structure COB 0 fait un appel conditionnel à trois sous structures (PB 10, 11 et FB 156). La condition d'appel est marquée entre les crochets.





## 2.7.2 Liste des blocs d'organisation



Block  
List  
view

Le bouton *Block List view* ou le menu *block List* affiche la liste de tous les blocs d'organisation qui constituent le programme.

Regulation	COB	0		
Summer_Control	PB	10		Program for sommer time
Winter_Control	PB	11		Program for winter time
Maintenance_Control	FB	156		Program for mainten...
Light_Control	COB	1		
Init	XOB	16		Initialise your parameters
Test_Install	PB	12		Test your installation
Diag_Instal	PB	13		Installation diagnose

## 2.7.3 Liste des symboles

Les menus *View*, *Global Symbols* et *View Data List* affichent la liste des symboles de l'ensemble du programme:

- *Global Symbols*, affiche les symboles partagés par tous les fichiers du CPU actif. Ces symboles peuvent être édités dans cette fenêtre.
- *Data List view*, affiche tous les autres symboles du CPU actif. Cette liste ne peut pas être éditée. Les symboles qui ne sont pas utilisés ne sont pas affichés dans cette fenêtre.



Global  
Symbols



View  
List view

Symbol ▲	Type	Address/...	Scope	Module	Comment
DailyTimer	O	32		Daily Timer.fbd	Daily Timer
HMS	R	2003	AUTO	Daily Timer.fbd	PCD Clock with current time
OFFTIME	R	2004	AUTO	Daily Timer.fbd	Switch off time
ONTIME	R	2005	AUTO	Daily Timer.fbd	Switch on time

## 2.7.4 Référence croisée

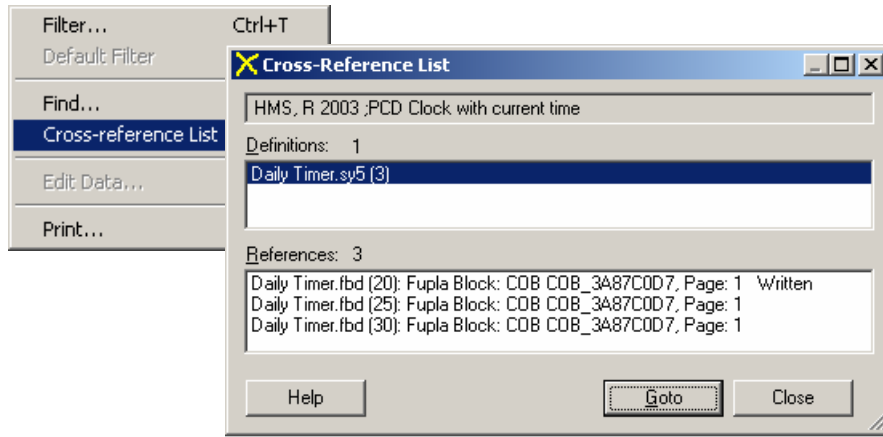
La liste *Global Symbols* et *Data List view* offre la possibilité de sélectionner un symbole et de retourner une référence croisée: liste des pages du programme faisant usage du symbole sélectionné.

Chaque ligne présente le nom du fichier et du bloc dans lequel le symbole est utilisé, ainsi qu'un numéro de ligne ou de page du programme. L'indication *Written* indique si le symbole est modifié à cet endroit du programme.

La liste *Definitions* montre où est défini le symbole. Par exemple la ligne de la définition EQU pour le code IL. La liste *References* présente où est utilisé le symbole dans le programme.

Pour les blocs, '>>' indique où le bloc peut être trouvé.

Pour visualiser les pages de programme faisant usage du symbole sélectionné lors de la référence croisée, il est juste nécessaire de marquer une des références et de sélectionner le bouton *Goto*.



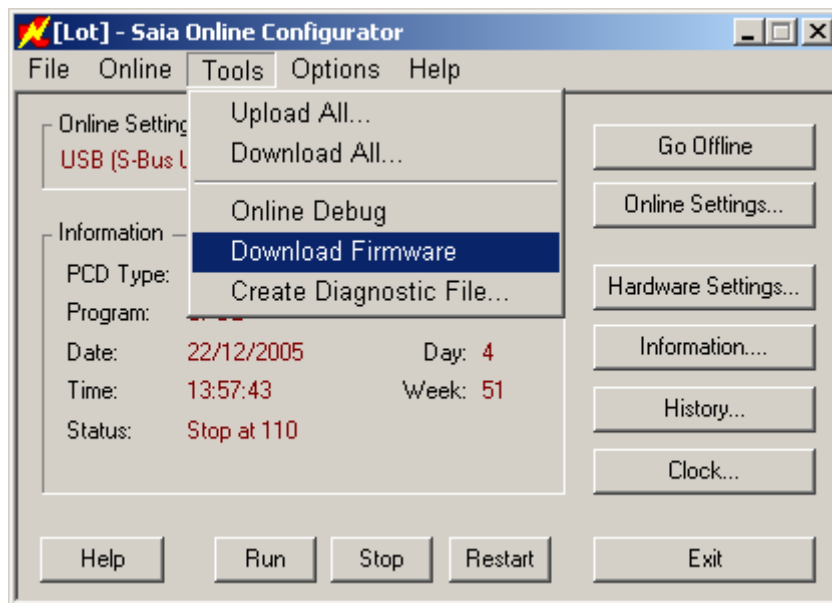
## 2.8 Backup du programme.

Le résultat d'une modification de programme PCD est parfois incertain. Par exemple, vous n'êtes pas sûr que les fichiers sources à disposition sont de la dernière version, l'installation ne vous est pas familière, ...

Que cela ne vous cause aucun souci, tout le contenu de la mémoire PCD peut être sauvegardé et restitué en cas de problème.

La manipulation qui suit permet de sauvegarder l'intégralité des données mémoire du PCD dans un fichier. (Programme, *Settings Hardwares*, valeurs des registres, indicateurs, compteurs, DB, textes, ...)

Pour restituer le programme dans la mémoire du PCD utiliser le menu *Tools, Download All*



Remarque :

Le backup peut aussi être réalisé avec une mémoire backup PCD7.R400/R500

## 2.9 Self downloader

Le self downloader simplifie beaucoup le chargement des programmes et *hardwares settings* sur le site d'exploitation des PCD.

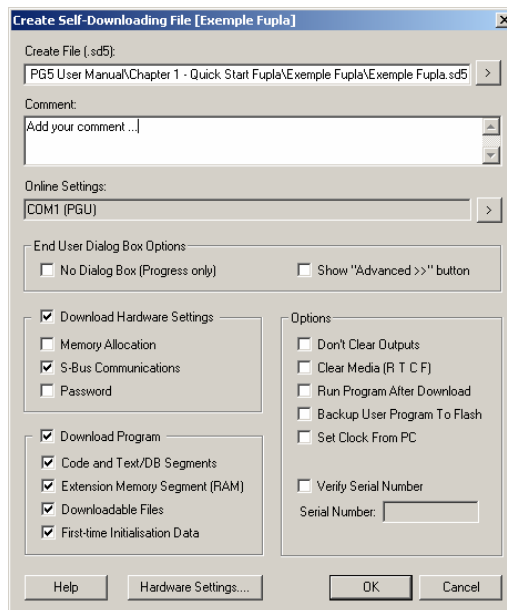
Cet outil PG5 permet de préparer un fichier \*.sd5 contenant toutes les informations nécessaires à la mise à jour des programmes et configurations d'un PCD.

Le programmeur PG5 peut facilement transmettre ce fichier par e-mail à la personne responsable du site d'exploitation des PCD.

L'ouverture d'un fichier \*.sd5 affiche le dialogue de chargement des données avec quelques paramètres et options correspondant à celle prédéfinies dans le projet PG5. La personne présente sur le site d'exploitation peut éventuellement modifier ces options ou les garder tel quel puis les charger dans le PCD.

Ainsi, le chargement du programme, des *hardwares settings* PCD ne nécessite pas de connaissance particulière de l'utilisation de PG5. Cet outil autorise le charment des programmes et *hardwares settings* sans installer PG5 ni licence d'exploitation mais nécessite tout de même l'installation du programme *Online Tools* sur le site d'exploitation.

### 2.9.1 Préparer un fichier \*.sd5



Le fichier est préparé à partir des informations présentes dans le CPU actif représenté par la fenêtre *Project Tree*. Il convient de vérifier si les configurations *Online settings* et *Hardware settings* sont bien correctes et de réaliser le *build* du CPU avant de préparer le fichier \*.sd5

Le menu CPU, Create Self-Downloading File, permet de configurer les paramètres et options pour le *self downloading* sur le site d'exploitation.

Ces paramètres et options sont identiques à ceux que nous connaissons déjà avec les menus: *Online*, *Hardware Settings*, *Download* et *Online*, *Download Program* ... Seul quelques nouveaux paramètres ont été ajoutés:

**Create Files (\*.sd5)**

Permet de définir le chemin et le nom du fichier .sd5

**Show "Advanced >>" button**

Permet de cacher toutes les présélections de ce dialogue lors du *self-downloading*

**No Dialog box (Progress only)**

Download le fichier .sd5 sans afficher de dialog (mode silence)

**Verify Serial Number**

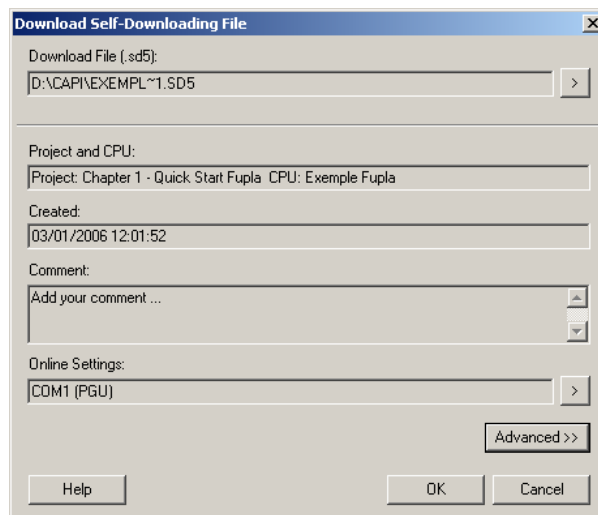
Le *self-downloading* vérifie si le numéro de série du PCD correspond à celui inscrit dans le champ *Serial Number*.

Le numéro de série est unique à chaque PCD, il permet de vérifier si le chargement est effectué sur le PCD pour lequel il est destiné.

**Remarque:**

le numéro de série n'est supporté que par les nouveaux systèmes PCD3. Il peut être lu directement sur le PCD avec le configurateur en ligne, menu *Online, Informations*.

## 2.9.2 Download d'un fichier \*.sd5



Pour charger le contenu d'un fichier .sd5, vérifier si le PG5 ou l'outil *Online Tools* sont bien installés sur l'ordinateur. Pour de plus amples renseignements, se référer au guide d'installation PG5.

Ouvrir le fichier .sd5 à partir du *windows explorer* avec une double sélection de la souris. Un dialogue comme celui représenté ci-dessus devrait être affiché.

Le chargement de son contenu peut être reconfiguré avec le bouton *Advanced*. Lancer le *download* avec le bouton OK.



## Table des matières

---

<b>3</b>	<b>PCD - RESSOURCES</b>	<b>2</b>
<b>3.1</b>	<b>Au sommaire de ce chapitre</b>	<b>2</b>
<b>3.2</b>	<b>Ressources matérielles (Hardware)</b>	<b>3</b>
3.2.1	Entrées et sorties digitales	3
3.2.2	Horloge	4
3.2.3	Entrées interruptives	5
<b>3.3</b>	<b>Ressources internes (softwares)</b>	<b>6</b>
3.3.1	Indicateurs (Flags)	6
3.3.2	Registres	7
3.3.3	Constantes	8
3.3.4	Temporisateurs et compteurs	9
3.3.5	Textes et blocs de données	12
3.3.1	Tableau récapitulatif.	14
<b>3.4</b>	<b>L'éditeur de symboles</b>	<b>15</b>
3.4.1	Désignation symbolique d'une ressource	15
3.4.2	Groupement de symboles	16
3.4.3	Portée des symboles	16
3.4.4	Symbole local	17
3.4.5	Symbole global	17
3.4.6	Définition d'un symbole global	18
3.4.7	Définition d'un symbole pour les réseaux de communication	18
<b>3.5</b>	<b>Exploitation des symboles</b>	<b>19</b>
3.5.1	Editer une liste	19
3.5.2	Ajouter plusieurs symboles à l'éditeur	20
3.5.3	Symbole avec référence	21
3.5.4	Importer les symboles d'une liste EQU	22
3.5.5	Importation de symboles avec d'autres applications	22
3.5.6	Ajouter des symboles durant l'édition du programme IL	22
3.5.7	Ajouter des symboles durant l'édition du programme Fupla	23
3.5.8	Transfert de symboles	24
3.5.9	Reconnaissance automatique des symboles	25
3.5.10	Allocation automatique d'adresse	25
3.5.11	Saisie de textes	26
3.5.12	Saisie de blocs de données	27
3.5.13	Recherche de symbole	27
3.5.14	Organisation des symboles	28
3.5.15	Réorganisation de la vue Liste	29
3.5.16	Exportation de symboles	30
3.5.17	Importation de symboles	32
3.5.18	Initialisation des symboles	34
3.5.19	Noms de symbole	35
3.5.20	Mots réservés	35

## **3 PCD - Ressources**

---

### **3.1 Au sommaire de ce chapitre**

Ce chapitre donne un aperçu de tous les types de données utiles dans une application.

La première partie de ce chapitre passe en revue ces ressources (entrées, sorties, indicateurs...), ainsi que leur usage et leur plage d'adresses.

La seconde partie explique comment les utiliser dans l'éditeur de symboles.

### 3.2 Ressources matérielles (Hardware)

Chaque programme est constitué de fonctions qui permettent de lire, d'écrire et de manipuler différents types de ressources. Les ressources servant à interagir avec l'environnement extérieur sont qualifiées de matérielles.

#### 3.2.1 Entrées et sorties digitales

I/O informations binaires (0/1)

Nombre maximum d'entrées et sorties	
PCD1	32 ( 64 <sup>3)</sup> )
PCD2.M120/M150	64/96/128 (255 <sup>3)</sup> ) <sup>1)</sup>
PCD2.M170 + PCD3.C100	510 <sup>2)</sup> <sup>3)</sup>
PCD2.M480 + PCD3.C100	1023 <sup>1)</sup> <sup>3)</sup>
PCD3.Mxxxx	1023 <sup>1)</sup> <sup>3)</sup>
PCD4	510 <sup>2)</sup>
PCD6	5100 <sup>2)</sup>

- 1) Les adresses 255 (et 511 du PCD2.M170/480) sont réservées pour le watchdog
- 2) Les adresses 255, 511, 767, 1023 etc.. jusqu'à 5119 sont réservées pour le watchdog
- 3) avec les modules d'entrées PCD2/3.E165 et/ou de sortie PCD2/3.A465

Les entrées et les sorties représentent respectivement des signaux à destination ou en provenance du PCD. Les entrées donnent l'état des fins de course, boutons poussoirs, détecteurs de proximité, capteurs... Les sorties permettent de commander des vannes, des lampes, des moteurs ,...

Vous pouvez lire et écrire des sorties. Vous ne pouvez que lire des entrées.

Entrées et sorties sont ajoutées au PCD en insérant des cartes d'E/S dans les emplacements désignés de l'automate. L'adresse de début d'un emplacement est définie soit par sa position (PCD1/2/3 et 4), soit par des commutateurs (PCD6).

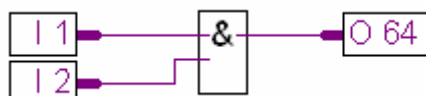
L'exemple qui suit active la sortie O64 si les entrées I1 et I2 sont toutes deux à l'état haut. Il est aussi possible de représenter cette fonction sous la forme d'une équation booléenne:  $O\ 64 = I\ 1 * I\ 2$ .

Programme IL:

```

COB  0
      0
STH  I 1
ANH  I 2
OUT  O 64
ECOB
    
```

Programme en Fupla :



Fbox : Binaire, ET



### 3.2.2 Horloge

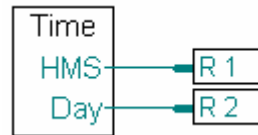
La plupart des PCD1 (PCD1.M120/130) et tous les PCD2/3/4/6 intègrent une horloge temps réel, souvent abrégée *RTC* (de l'anglais *Real Time Clock*). Une instruction spéciale vous permet de charger la date et l'heure dans un registre.

L'exemple suivant illustre la lecture de l'heure et de la date dans votre programme.

#### Programme IL:

```
COB 0
    0
RTIME R 1
ECOB
```

#### Programme Fupla:



Fbox: *Temporisateurs, Lecture horloge*

Ce programme lit le contenu de l'horloge PCD et la copie sur deux registres, dont seul le premier (R1) est spécifié. L'heure et la date sont représentées comme suit :

```
R 1 = 093510    9 heures, 35 minutes, 10 secondes
R 2 = 073030210  Semaine 07, 3ème jour de la semaine (mercredi),
                  le 10 février 03 (2003)
```

### 3.2.3 Entrées interruptives

Certains PCD<sup>1)</sup> sont équipés de deux entrées interruptives, INB1 et INB2. A chaque front montant de l'une de ces entrées, le programme PCD interrompt son déroulement normal pour exécuter un bloc d'exception XOB20 (INB1) ou XOB25 (INB2). Ces entrées peuvent s'activer jusqu'à la fréquence maximum de 1 000 fois par seconde.

L'exemple ci-dessous montre comment compter des impulsions avec l'entrée INB1

**Programme IL:**

```

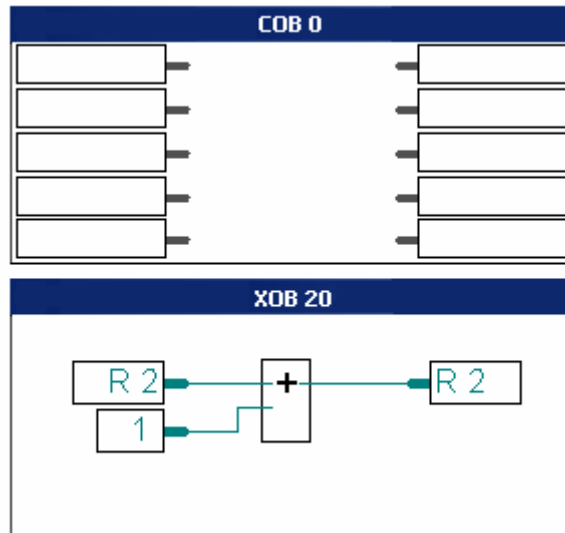
COB 0 ; Programme
      0 ; principal

ECOB

XOB 20 ; interruption INB1
INC R 2 ; incrémentation
        ; du registre R2

EXOB
    
```

**Programme Fupla:**



- 1) PCD1.M120/130, PCD2.M120/150, PCD2/4.M170, PCD2.M480 (4 entrées interruptives IN0 ... IN3), PCD3.M et PCD6.M3
- 2) Pour de plus amples informations, veuillez consulter le manuel hardware du PCD



En raison des limitations imposées par son filtre d'entrée (protection des entrées contre les perturbations électriques et les surtensions, les contacts « sauto-nettoyants »), une entrée digitale normale ne peut pas compter des impulsions de fréquence supérieure à 50 Hz. Les entrées interruptives représentent alors une alternative intéressante pour résoudre ce genre d'application. Elles permettent d'éviter l'usage des cartes de comptage PCD2.H1 ou PCD4.H1 dont la fréquence de comptage maximum est de 10 à 160 kHz selon les modules.

### 3.3 Ressources internes (softwares)

#### 3.3.1 Indicateurs (Flags)

Informations binaires (0/1)



Un indicateur sauvegarde une information à 1 Bit. Les adresses des indicateurs sont comprises entre F 0 et F 8191. Par défaut, les indicateurs ne sont pas volatiles: les états logiques sont conservés par la batterie lors d'une coupure d'alimentation. Si nécessaire, les indicateurs peuvent être rendus volatiles par une configuration software représentée au bas de cette page. (*Software settings*)

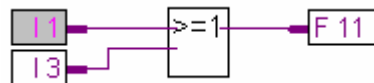
L'exemple ci-dessous force l'indicateur F 11 à l'état haut (H) lorsque les entrées 1 ou 3 sont H. Il est aussi possible de représenter cette fonction sous la forme d'une équation booléenne:  $F\ 11 = I\ 1 + I\ 3$

#### Utilisation des indicateurs dans un programme

Programme IL:

```
COB 0
    0
STH I 1
ORH I 3
OUT F 11
ECOB
```

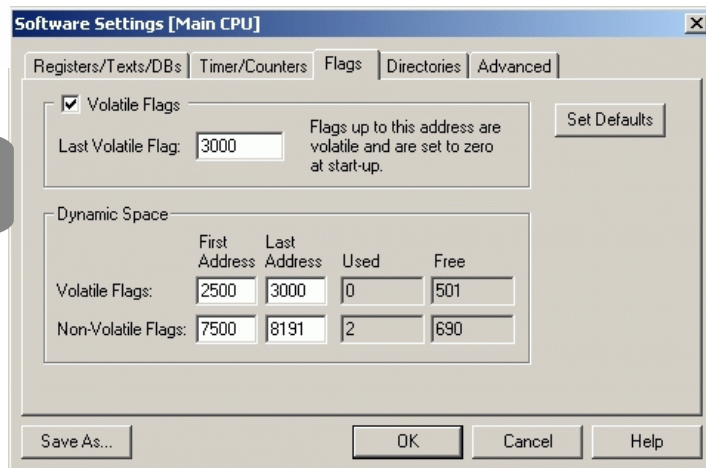
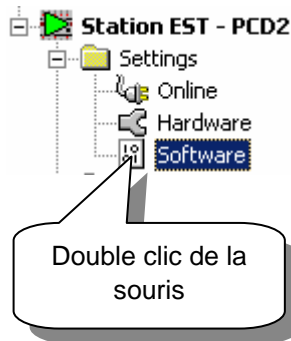
Programme Fupla:



Fbox : Binaire, OU

Par défaut, les indicateurs ne sont pas volatils. Pour les rendre volatils, sélectionner l'option *Volatile Flags* de la fenêtre *Software Settings* ci-dessous et définir une plage d'adresses.

#### Paramétrage des indicateurs

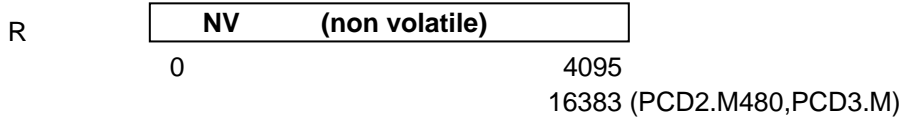


### 3.3.2 Registres

Informations 32 bits

Entiers: -2 147 483 648 à +2 147 483 647

Flottant: -9.22337E+18 à +9.22337E+18



Un registre peut contenir une valeur de format entier ou flottant. Les registres sont utiles pour effectuer des opérations arithmétiques ou les tâches de mesure et de régulation sur des valeurs analogiques. Avec Fupla, les lignes reliées à un registre sont représentées par différentes couleurs: bleu pour un entier et jaune pour une valeur flottante. Les liens bleus et jaunes ne peuvent pas être combinés par une Fbox ou reliés ensemble. Il est par exemple impossible de les additionner. Il faut d'abord convertir le format de l'un des registres.

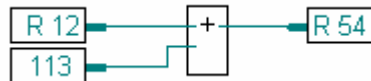
#### Utilisation des registres dans un programme

L'exemple suivant ajoute la constante 113 au contenu du registre R 12 et place le résultat dans le registre R 54:  $R\ 54 = R\ 12 + 113$ .

Programme IL:

```
COB 0
0
ADD R 12
K 113
R 54
```

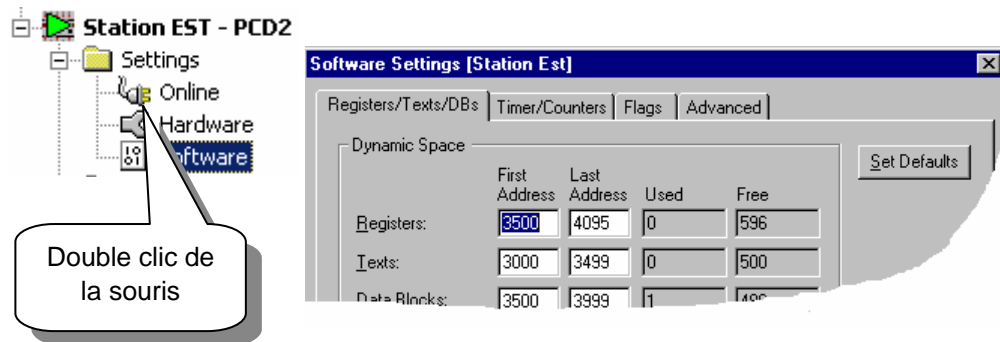
Programme Fupla:



Fbox: Entier, Addition

#### Paramétrage des registres

Le PG5 dispose d'un puissant outil d'allocation dynamique des ressources qui évite de préciser l'adresse des ressources software du programme. Le principe consiste à attribuer un nom de symbole à la ressource sans préciser d'adresse. Il est inutile de modifier ces paramètres si les programmes utilisent peu de registres dynamiques.



En cas d'erreur d'assemblage signalée par le message *Auto allocation overflow for type: R*, il convient d'étendre la plage des adresses dynamique pour les registres.

### 3.3.3 Constantes

Informations 32 bits

Entier: -2 147 483 648 à +2 147 483 647  
 Flottant: -9.22337E+18 à +9.22337E+18

Les constantes sont des valeurs fixes qui conservent la même valeur pendant l'exécution du programme. Elles sont écrites dans un registre.

**Exemple:** coefficients fixes comme  $\pi$  (PI) = 3,1415.

L'exemple suivant charge la constante 100 dans le registre R 4 et la divise par 0,25. Mais R 4 contient une valeur de format entier (100) à diviser par une valeur de format flottant (0,25), les informations sont de format différents et ne peuvent pas être supportée par l'opérations arithmétique, le contenu du registre 4 doit être converti en format flottant. Alors R 4 est copié dans le registre R 35 puis converti en format flottant et divisé par 0,25. Le résultat de la division est placé dans R 5 puis copié dans R 6 et converti en un format entier.

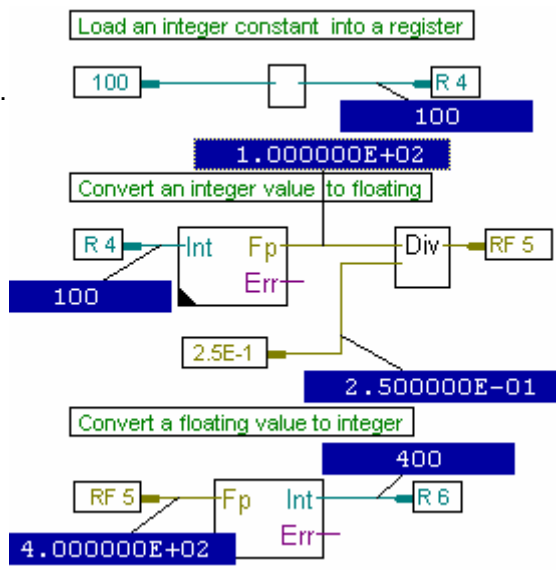
#### Utilisation des constantes dans les programmes

##### Programme IL:

```

COB  0      ;Bloc d'organisation
      0      ;cyclique
LD   R 4    ;Charge 100 dans R4.
      100
COPY R 4    ;Copie R4
      R 35   ;dans R 35
IFP  R 35   ;Converti R 35
      0      ;en format flottant.
LD   R 36   ;Charge la constante
      2.5e-1 ;0,25 dans R 36.
FDIV R 35   ;Divise la valeur
      R 36   ;par 0,25
      R 5    ;et place le résultat
           ;dans R 5.
COPY R 5    ;Copie R 5
      R 6    ;dans R 6
FPI  R 6    ;Converti R 6
      0      ;en format entier
ECOB
    
```

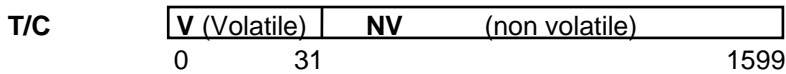
##### Programme Fupla:



- Fbox:
- Entier, Transfert direct
  - Convertisseurs, Entier flottant
  - Virgule flottante, Division
  - Convertisseurs, Flottant entier

### 3.3.4 Temporisateurs et compteurs

Informations 32 bits ( 0 ... 2 147 483 648)



Temporisateurs et compteurs supportent des valeurs comprise entre 0 et 2 147 483 648 (31 bits). Temporisateur et compteurs partagent la même plage d'adresses (0 à 1 599). Par défaut, les adresses 0 à 31 sont attribuées aux temporisateurs et les adresses 32 à 1 599 aux compteurs, la base de temps des temporisateurs est de 100 ms. Le système diminue chaque temporisateur d'une unité, toutes les 100 ms. La base de temps ainsi que la répartition des adresses entre temporisateurs et compteurs sont ajustables par la fenêtre *Software Settings*.

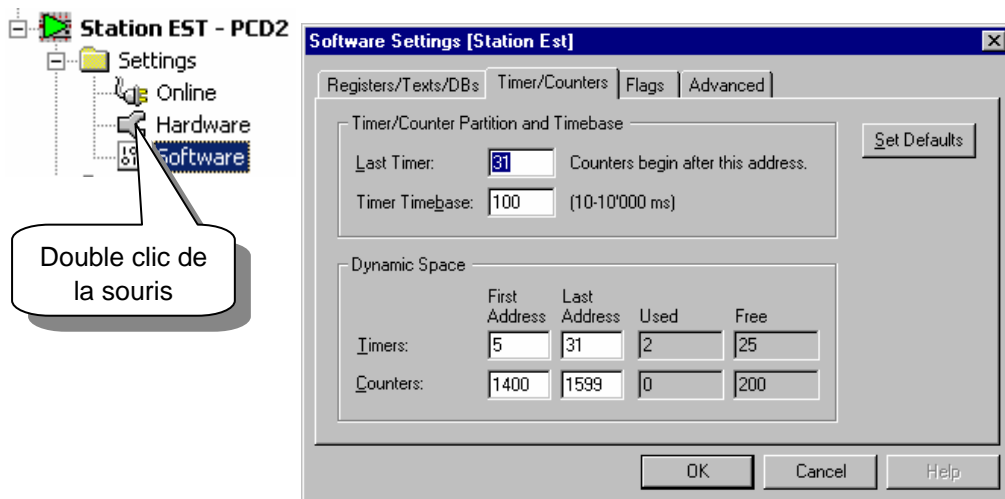
Les temporisateurs et compteurs ne supportent que les valeurs entières et positives. La valeur peut être initialisées par une instruction LD. Les temporisateurs ne peuvent être uniquement décrémentés. Les compteurs sont incrémentés et décrémentés par le programme utilisateur avec les instruction INC et DEC. (INC: ↑, DEC: ↓).

Temporisateurs et compteurs sont aussi exploitables avec des instructions binaires.

Si le temporisateur ou un compteur contient une valeur non nulle, son état logique est haut (1). Inversément, une valeur nulle le fait passer à l'état logique bas (0).

### Paramétrage des temporisateurs et compteurs

La répartition des plages d'adresse entre temporisateurs et compteurs tout comme la modification de la base de temps de 100 ms, s'effectuent dans le répertoire *Timer/Counters* de la fenêtre *Software Settings* ci-dessous:



### Informations techniques

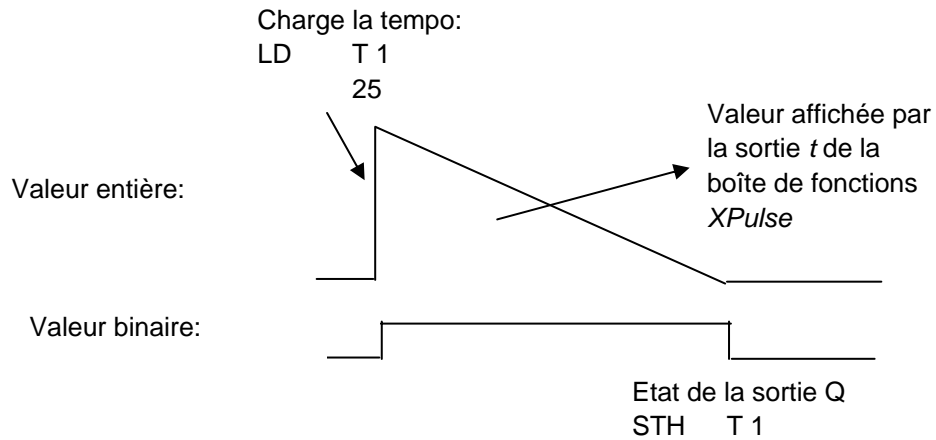
La configuration d'un grand nombre de temporisateurs et la diminution de la base de temps augmentent la charge du CPU.

Exemple: 100 temporisateurs mobilisent environ 2 % de la capacité CPU.

Soyez donc vigilant lorsque vous envisagez ces modifications!

**Exemple: temporisation**

Si le signal de l'entrée 4 passe de l'état bas (L) à haut (H), le flanc montant du signal d'entrée 4 force la sortie 0 65 à l'état H pour une durée de 2,5 secondes.



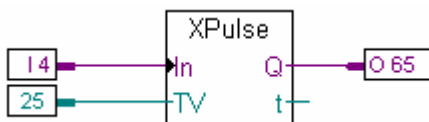
**Solution:**

**Programme IL:**

```

COB 0 ;Début du COB
      0 ;
STH I 4 ;Sur détection du flanc
DYN F 12 ;montant de l'entrée I 4,
LD T 1 ;charger le tempo T 1
      25 ;avec 2,5 secondes.
STH T 1 ;Copier l'état du tempo
OUT O 65 ;sur la sortie O 65.
ECOB ;Fin du COB
    
```

**Programme Fupla:**



Fbox : Temporisateurs, Impulsion fixe



**Information technique**

Les temporisateurs sont décrémentés à des intervalles de temps définis dans les *Software Settings*, sous *Timer*, *Timebase*. Si la base de temps est modifiée, toutes les constantes nécessaires à initialiser les temporisations doivent être adaptées pour conserver la même durée. Pour éviter cette correction, le format de donnée ci-dessous permet d'adapter automatiquement les constantes de temporisation à la base de temps sélectionnée dans les *Settings Softwares*.

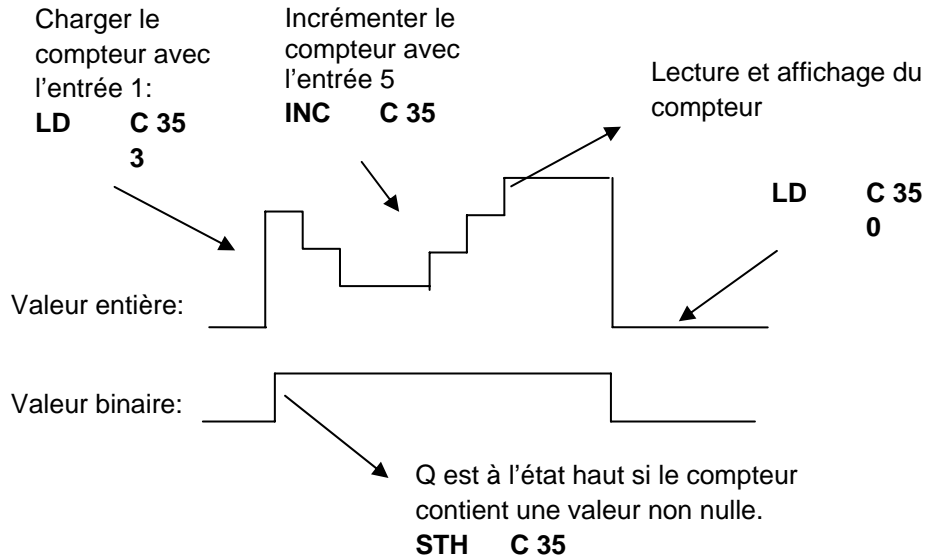
**Format: T#nnnS|MS**

BL_3DE393BA	COB		
DelayTime	K Constant	T#100MS	100 millisecondes
OneDay	K Constant	T#3600S	3600 secondes

### Exemple: Compteurs

Compteur, décompteur avec présélection et reset:

Un état logique haut (H) sur l'entrée 1 charge le compteur avec une valeur 3. Alors qu'un état logique haut (H) sur l'entrée 2 charge le compteur avec une valeur 0. Si les entrées 1 et 2 sont à l'état bas (L), les changements d'état bas/haut (L/H) des entrées 5 et 6 permettent d'incrémenter et décrémenter le compteur.



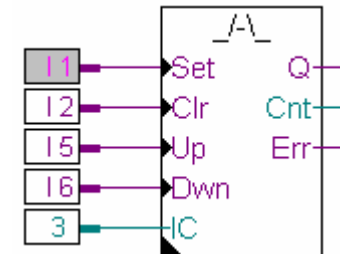
### Solution:

#### Programme IL:

```

COB 0 ;Début du COB
0 ;
STH I 1 ;Si l'entrée I 1 est à l'état haut,
LD C 35 ;charger le compteur C 35
3 ;avec 3.
STH I 2 ;Si l'entrée I 2 est à l'état haut,
LD C 35 ;charger le compteur C 35
0 ;avec 0.
STH I 5 ;Sur détection du flanc
DYN F 13 ;montant de l'entrée I 5,
INC C 35 ;incrémenter le compteur C 35.
STH I 6 ;Sur détection du flanc
DYN F 14 ;montant de l'entrée I 6,
DEC C 35 ;décrémenter C 35.
ECOB ;Fin du COB
    
```

#### Programme Fupla:



Fbox :  
Compteur, Haut/bas + PRS+ RAZ



### 3.3.5 Textes et blocs de données

TEXT/DB	<b>Mémoire principale</b>		<b>NV</b>
	0		3999
	<b>Mémoire d'extension</b>		<b>NV</b>
	4000		(PCD4/6) 7999
			(PCD2) 5999
			(PCD1) 4999
			(PCD2.M480, PCD3.M) 8191

Textes et blocs de données (*DB*) sont des ressources non volatiles. Les premiers, constitués de chaînes de caractères, sont des messages à afficher ou à envoyer à un boîtier de radiomessagerie, des séquences d'initialisation de modem...

Les DB, quant à eux, servent à l'acquisition de données, aux tables...



#### Information technique

##### Où sont sauvegardés les textes et DB ?

Les registres, indicateurs et temporisateurs/compteurs sont gérés par le système et stockés dans une petite RAM, indépendante de la mémoire principale.

Par contre, les blocs de données et les textes sont conservés en mémoire principale avec le programme utilisateur. Si vous faites d'une Flash EPROM ou d'une EPROM standard votre mémoire principale, sachez qu'en mode de fonctionnement *RUN*, celle-ci est accessible en lecture, mais pas en écriture ! Il est donc impossible d'y modifier le contenu de vos DB (pour l'acquisition de données, par exemple). Dans la plupart des cas, vous n'aurez pas ce souci, mais s'il vous faut effectuer des opérations de lecture/écriture sur ces DB, vérifiez qu'ils sont bien stockés en mémoire d'extension et adressés à partir de 4 000. Cette mémoire est en effet toujours une RAM, donc accessible en lecture et en écriture.

#### Exemple: déclaration de textes et blocs de données en langage IL

TEXT 10 "Bonjour!"	;Le texte n° 10 contient la chaîne « Bonjour! ».
TEXT 11 [7]"Hello"	;Le texte n° 11 contient 7 caractères, soit la chaîne ;« Hello » (5 caractères) suivi de 2 espaces blancs.
DB 12 45,46,78,999,0	;Le DB n° 12 contient 5 valeurs entières : ;45, 46, 78, 999, 0
DB 13 [10]	;Le DB n° 13 contient 10 valeurs, égales à 0 au démarrage.
DB 14 [4] 2,3	;Le DB n° 14 contient 4 valeurs, les 2 premières sont 2 ;et 3, les 2 dernières sont à 0.

### Exemple: acquisition de données avec Fupla

Voici comment enregistrer facilement les valeurs d'une carte analogique dans le DB n° 4010. Chaque activation du signal *Store* entraîne la lecture de la valeur analogique et son écriture dans le DB 4010.

The screenshot shows the SAIA Fupla Editor interface. A ladder logic diagram includes a 'Store' signal connected to a 'DB Logger' block, which is configured to write to 'DB 4010'. A 'PCD2.W2' block is also shown with an 'Add' value of 64. The 'Edit Data Block - DB 4010' dialog box is open, showing the following table:

Item	Value	Comment
0	0	Default values
1	0	
2	0	
3	0	

Annotations in the image include: 'Dimension du DB' pointing to the 'Size: 4' field; 'Double clic de la souris' pointing to the 'DB 4010' symbol in the ladder logic; and 'Valeurs par défaut' pointing to the table.

**Fbox:**

- Module analogique, PCD2.W2
- Blocs de données, Mémorisation

### Exemple: d'envoi SMS avec Fupla

Voici comment envoyer un message SMS à partir de l'état binaire d'une entrée digitale ou un d'un indicateur. Le message est défini dans le texte 10. Remarquez que les fonctions graphiques de cet exemple ont l'angle inférieur marqué en noir. Cela signifie que ces fonctions disposent d'une fenêtre avec des paramètres pour le numéro du natel destinataire, le modem, ... Les fenêtres d'ajustage peuvent être visibles avec un double clic de la souris sur le centre de la fonction.

The screenshot shows the SAIA Fupla Editor interface. A ladder logic diagram includes a 'Modem 14' block, a 'Call SMS' block, and a 'SEND SMS' block. The 'SEND SMS' block is configured with 'Msg' set to 'Text 10'. The 'Edit Text - Text 10' dialog box is open, showing the following text:

Text: "Alarm: battery failure"

Annotations in the image include: 'Un double clic de la souris, affiche les paramètres d'ajustage' pointing to the 'SEND SMS' block; and 'Double clic de la souris' pointing to the 'Text 10' symbol in the ladder logic.

**Fbox:**

- Modem, Modem Driver 14
- Modem SMS, Call SMS
- Modem SMS, Send SMS

### 3.3.1 Tableau récapitulatif.

Description	Média	Opérande	Binaire	Numérique	Volatile
Entrées	I	1) 0...8191	0,1		
Sorties	O	1) 0...8191	0,1		
Indicateurs	F	0...8191	0,1		2) Non
Temporisateurs	T	2) 0...31	0,1	0 ... 2 147 483 648	Oui
Compteurs	C	2) 32...1599	0,1	0 ... 2 147 483 648	Non
Registres	R	0... 4095 5) 0...16383		-2 147 483 648...+2 147 483 647 - 9.22337E+18...+9.22337E+18	Non
Texte	X	3) 0...3999 4) 4000 ...		Chaîne max. 3072 caractères	Non
Blocs de données	DB	3) 0...3999 4) 4000 ...		Max. 382 valeurs (accès lent) Max.16 383 valeurs (accès rapide)	Non

- 1) selon l'automate et sa configuration en entrées, sorties
- 2) par défaut, configurable à partir des *Settings Softwares*
- 3) sauvegardé sur la même mémoire que les programmes (RAM / EPROM / FLASH)
- 4) sauvegardé sur la mémoire d'extension (RAM)
- 5) PCD2.M480, PCD3.M

### 3.4 L'éditeur de symboles

Avant de commencer à programmer, il faut préparer la liste des ressources que vous envisagez d'utiliser: entrées sorties, temporisateurs,...

Ceux-ci doivent être connus du PG5 de façon qu'il puisse aisément les retrouver dans les fichiers programmes, signaler des erreurs de programmation ou aider au débogage du projet.

Tous ces éléments sont donc regroupés dans un outil centralisé, baptisé *Symbol Editor*.

Chaque ressource ou élément du PCD est ainsi désigné par un nom ou un symbole qui facilite la lecture du programme.

#### 3.4.1 Désignation symbolique d'une ressource

The screenshot shows the 'Symbols' window with a table of resources. Callouts provide detailed explanations for each column:

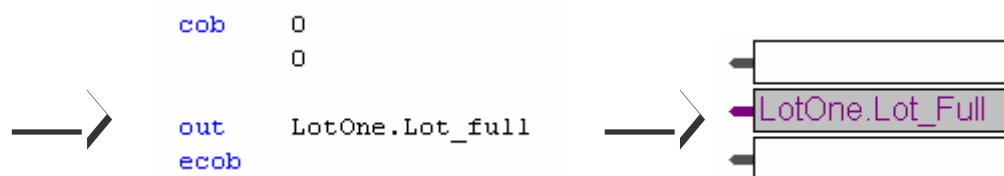
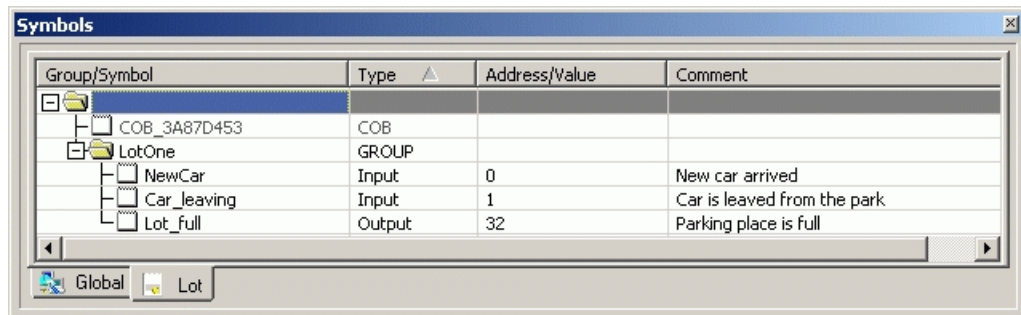
- Nom de la ressource (80 caractères maxi)**: Points to the 'Group/Symbol' column.
- Type : entrée, registre...**: Points to the 'Type' column.
- Commentaire : détail de chaque ressource, permet d'améliorer la lisibilité du programme.**: Points to the 'Comment' column.
- Nom du fichier auquel appartiennent les symboles. Ils ne sont connus que de ce fichier.**: Points to the 'Global' and 'Lot' tabs at the bottom.
- Adresse ou valeur :** Indiquez l'entrée ou la sortie correspondant au symbole. S'il s'agit d'une ressource interne, il est inutile d'en préciser l'adresse (Cf. chapitre 3.2), si non le système la choisira automatiquement une adresse (principe de l'allocation dynamique des ressources).
- Global, les symboles de cette liste sont connus de tous les fichiers du CPU**: Points to the 'Global' tab.

Group/Symbol	Type	Address/Value	Comment
COB_3A87D453	COB		
NewCar	Input	0	New car arrived
Car_leaving	Input	1	Car is leaved from the park
Lot_full	Output	32	Parking place is full

### 3.4.2 Groupement de symboles

Pour faciliter la lecture des programmes, vous pouvez grouper les symboles. Vous ajoutez un nouveau groupe à l'éditeur en sélectionnant le bouton droit de la souris et effectuez un glisser-déplacer des symboles de votre choix dans ce dossier, comme illustré ci-dessous.

Exemple : le groupe intitulé *LotOne* contient toute une liste de symboles.



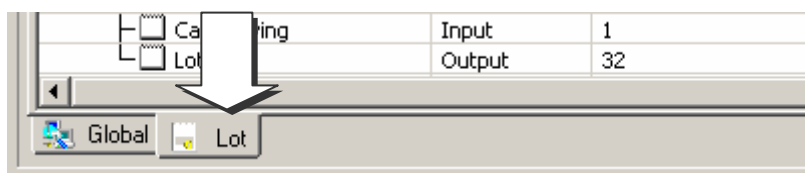
Dans le programme, le nom du groupe *LotOne* précède le nom du symbole *Lot\_full* et sont séparés par un point.

### 3.4.3 Portée des symboles

Les symboles ne sont normalement connus que d'un seul fichier. On les qualifie alors de « local ». Dès que vous ouvrez un fichier programme dans un éditeur, l'éditeur de symboles et sa liste de symboles s'ouvrent également.

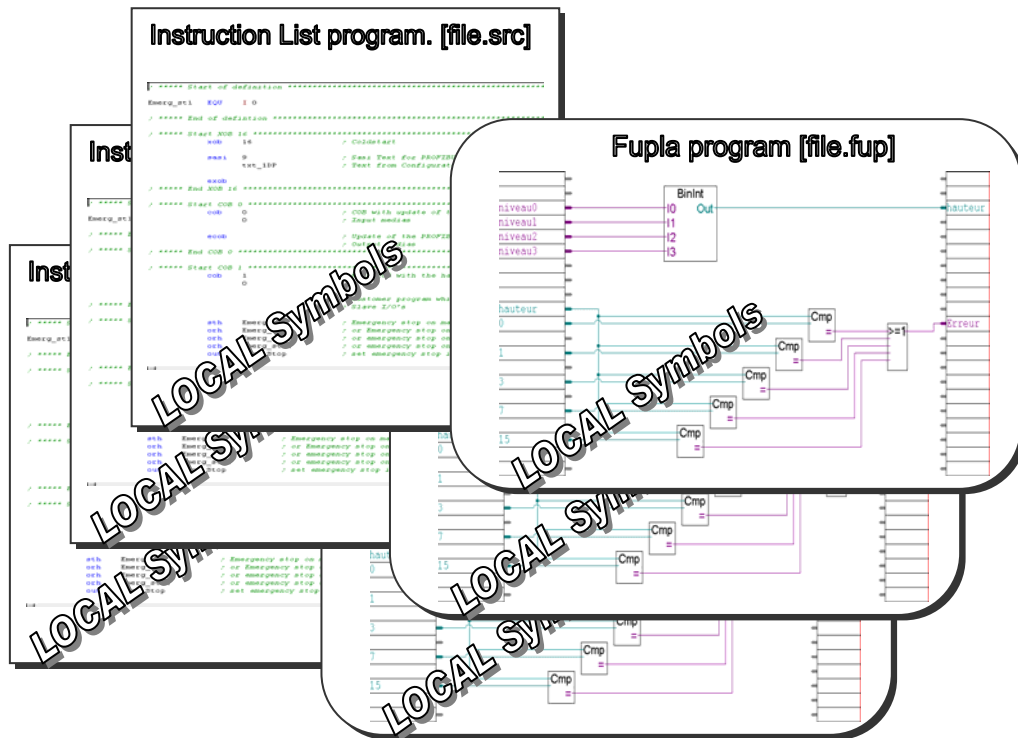
#### Exemple:

L'ouverture du fichier programme *Lot.src* ouvre automatiquement l'éditeur de symboles du même nom.



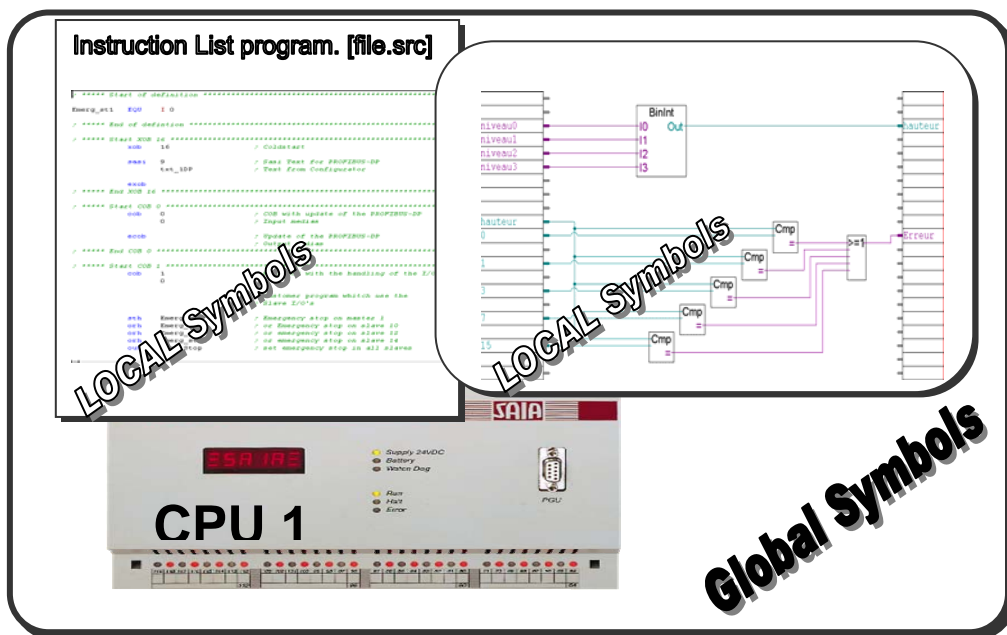
### 3.4.4 Symbole local

Un symbole local n'est connu que du fichier auquel il appartient.

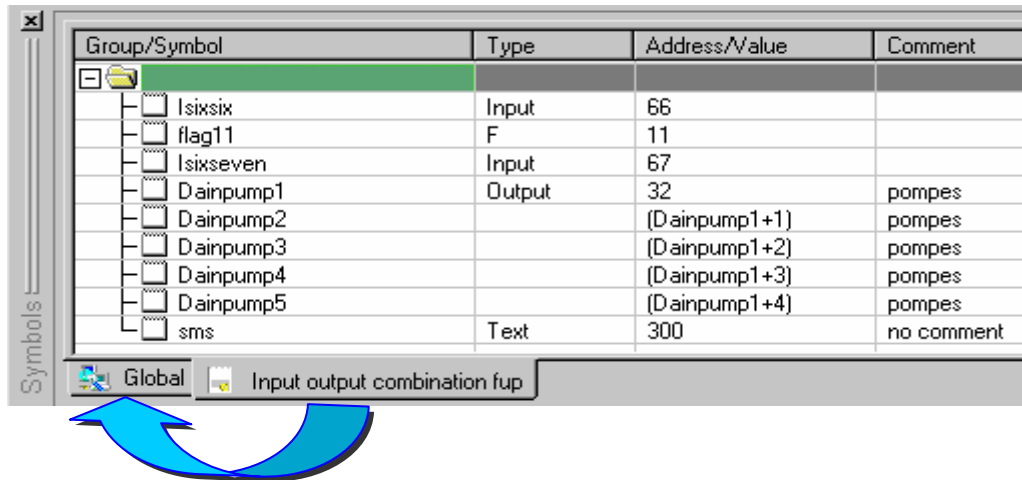


### 3.4.5 Symbole global

Un symbole global est connu de tous les fichiers de l'unité centrale.

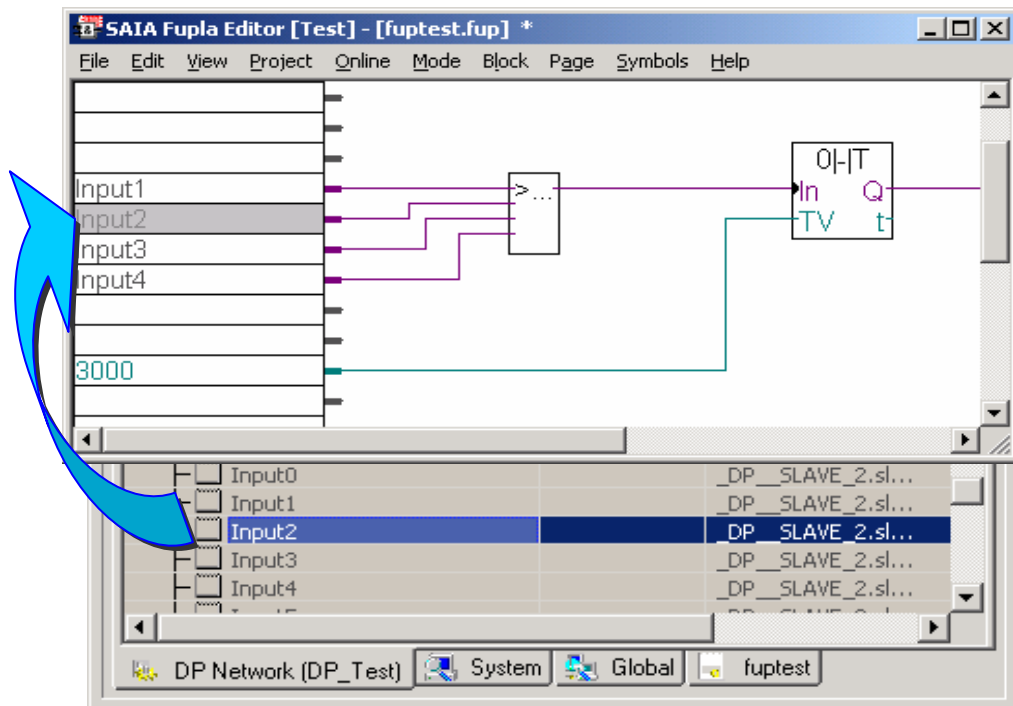


### 3.4.6 Définition d'un symbole global



Si les symboles sont utilisés par plusieurs fichiers, il faut alors les transférer de la liste locale à la liste globale. Sélectionner les symboles, pressez le bouton droit de la souris et choisir le menu *Advanced, Make Global*. Dès qu'un symbole devient global, il est visible depuis chaque fichier du projet.

### 3.4.7 Définition d'un symbole pour les réseaux de communication



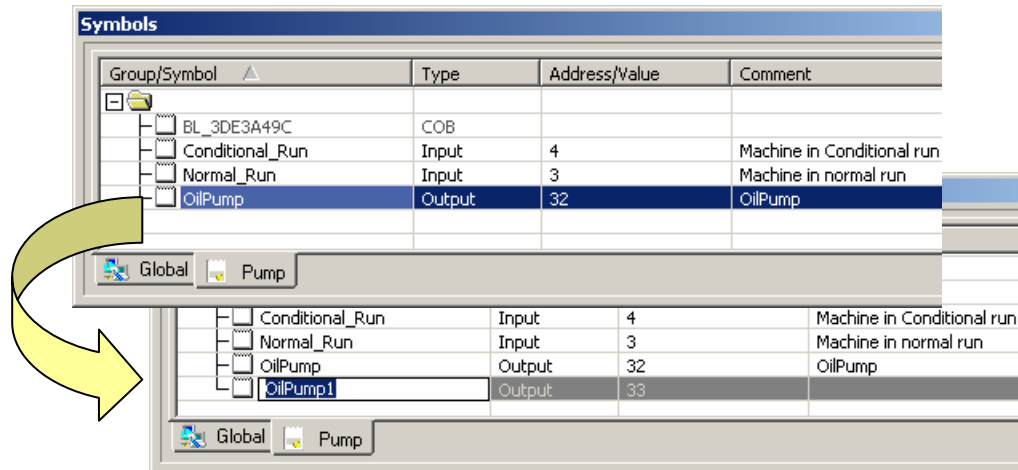
Les interfaces sérieelles et réseaux de communication supportent l'échange des données entre les CPU-PCD. L'éditeur de réseau permet de configurer les données à échanger avec les réseaux Profibus DP, Profibus FMS et Lon . Après la construction du programme (*Build*), toutes les données échangées avec le réseau de communication sont disponibles dans l'éditeur de symbole de plusieurs CPU. (Voir exemple pour Profibus DP)

## 3.5 Exploitation des symboles

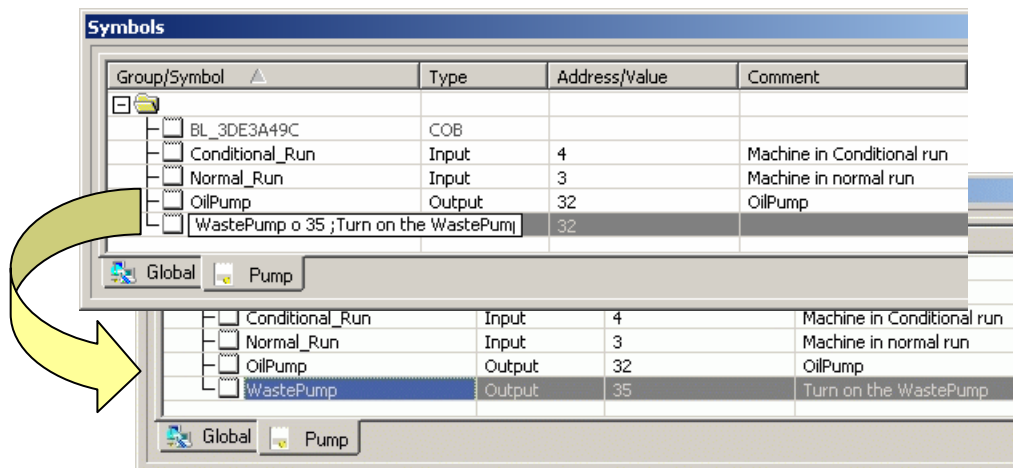
### 3.5.1 Editer une liste

Ouvrez le fichier sur lequel vous allez travailler, l'éditeur de symboles s'affiche automatiquement.

Cliquez dans la fenêtre *Group/Symbol* et enfoncez la touche *Insert*. Un nouveau champ de symbole s'ajoute instantanément à la liste. Saisissez le nom du symbole, son type, son adresse/valeur, ainsi qu'un commentaire. Appuyez sur *Entrée* pour valider.



Pour le symbole suivant, il suffit presser la touche *Insert* pour que l'éditeur affiche automatiquement le nom du symbole précédent ainsi que son adresse, majorée d'une unité (Cf. écran ci-dessous). Vous pouvez alors en modifier le commentaire tout en conservant le nom, le type, l'adresse/valeur ou entrer un nouveau nom, type, adresse/valeur et commentaire venant remplacer la précédente saisie.



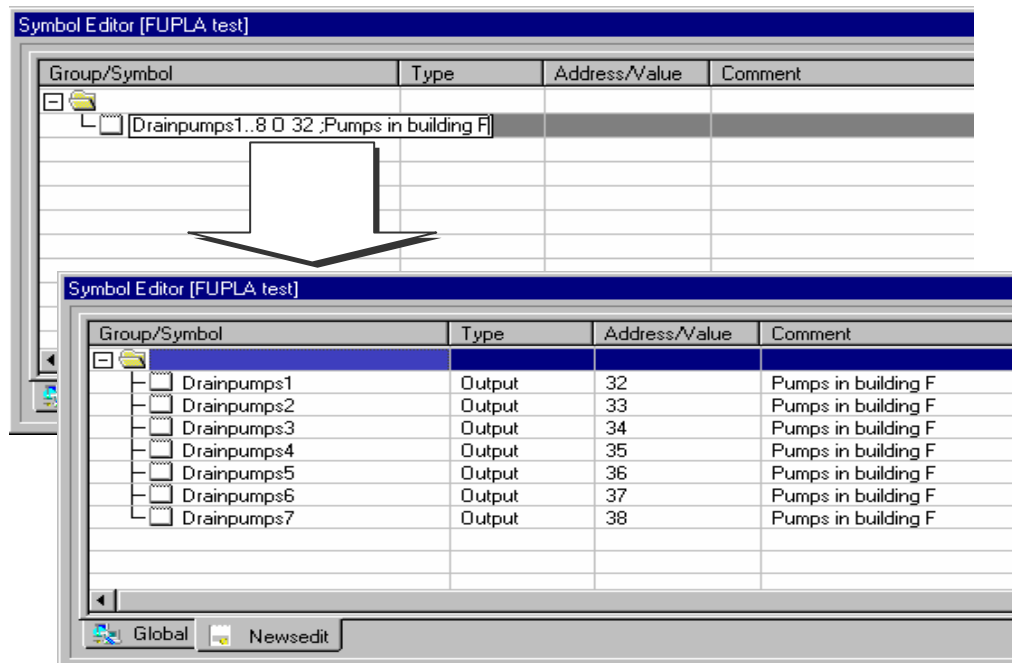
Si vous disposez déjà d'une liste que vous souhaitez simplement compléter, cliquez sur le dernier symbole et enfoncez *Insert* pour obtenir un nouveau champ de symbole.



### 3.5.2 Ajouter plusieurs symboles à l'éditeur

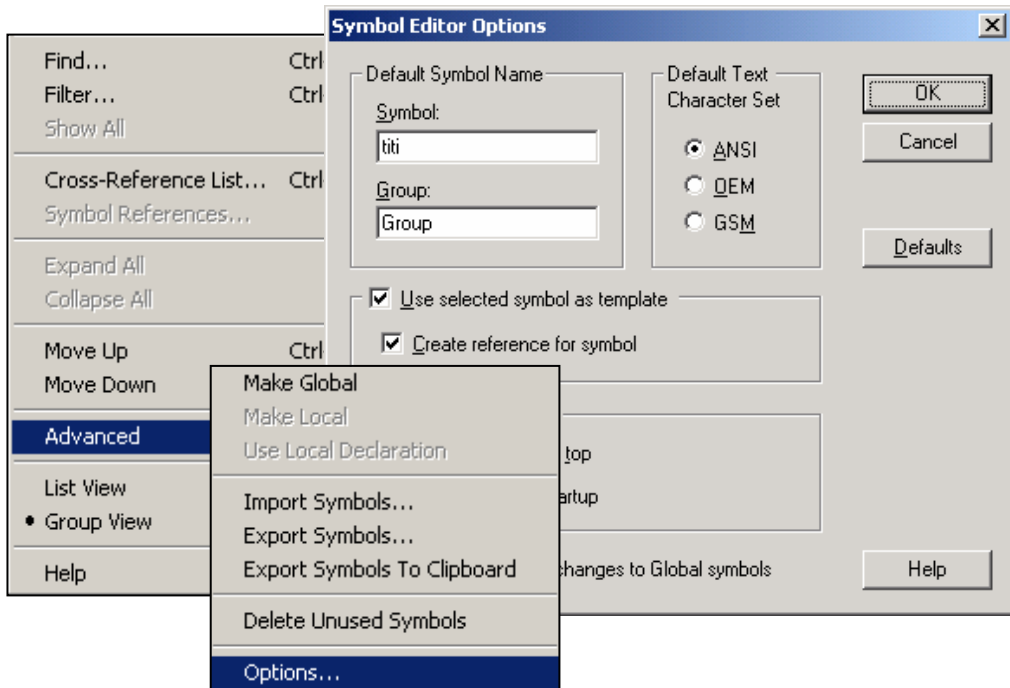
Il est possible d'ajouter toute une plage de symboles à votre liste. Entrez simplement le nom du symbole en précisant le numéro du premier et du dernier élément, comme dans l'exemple ci-dessous *Drainpumps1..8 O 32 ;Pumps in building F*.

Le chiffre 8 représente le nombre de symboles, O signale une sortie et 32 est l'adresse de début de votre plage. Validez par la touche *Entrée* : l'éditeur de symboles complète lui-même la liste.

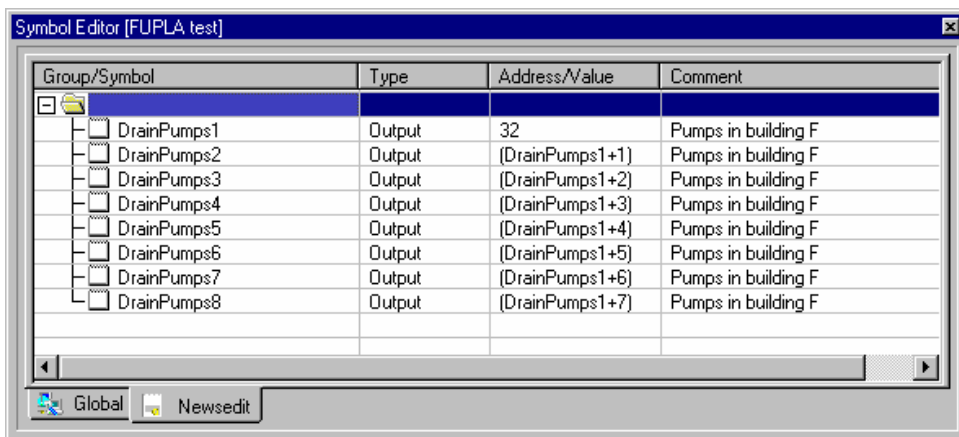


### 3.5.3 Symbole avec référence

Une liste de symbole peut aussi se référer à un symbole particulier. Sélectionner le menu *Symbols, Advanced, Options...* pour ouvrir la fenêtre *Symbol Editor Options*, entrer le symbole et cochez la case *Create reference for symbol*. Confirmer par *OK*.

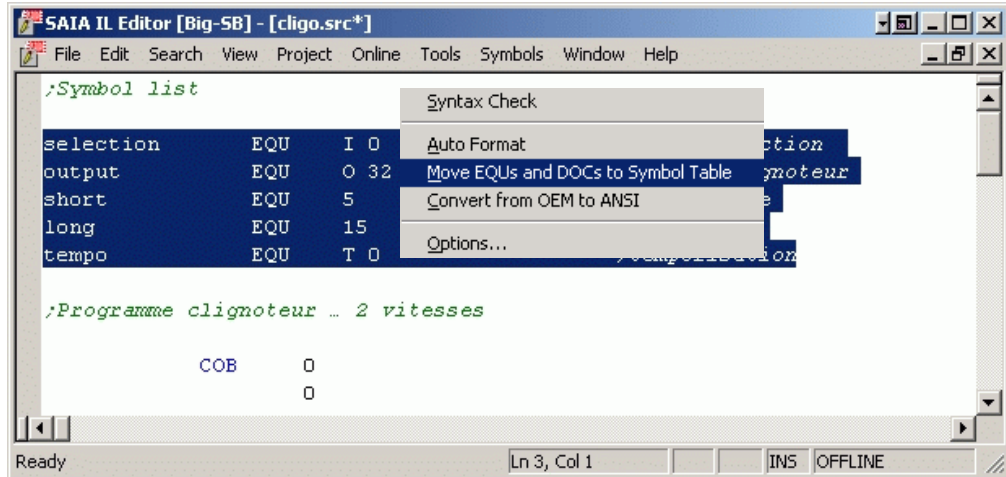


Marquer le symbole et presser la touche *Insert* du clavier. Le symbole apparaît incrémenté d'une unité. Cette fonction peut être utile si toute une suite de symboles est identique mais avec des adresses différentes. Seule les informations nécessaires doivent être adaptées.



### 3.5.4 Importer les symboles d'une liste EQU

Les listes EQU des anciens fichiers IL PG4 et PG3 peuvent être importées dans la fenêtre *Symbols*. Marquez la liste et importez les symboles avec le menu *Tools, Move EQUs and Docs to Symbol Table*. Les symboles sont automatiquement transférés dans la liste des symboles.

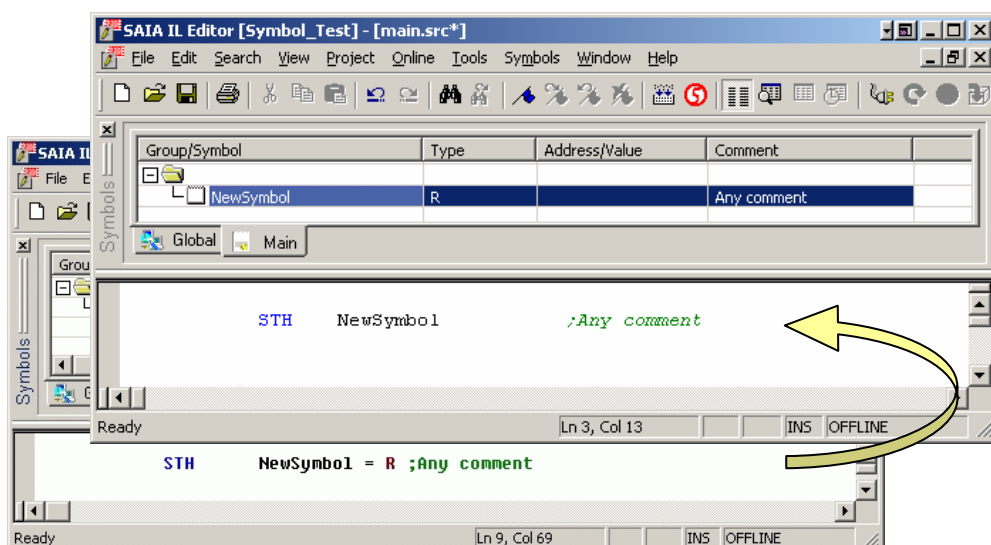


### 3.5.5 Importation de symboles avec d'autres applications

Les symboles appartenant à un autre programme (CAO électrique, supervision Visi+,...) peuvent également être importés dans un projet. Cette possibilité garantit la cohérence des documentations de l'ensemble du projet, l'usage des mêmes symboles sur les schémas électriques et dans le code programme. Utilisez la fonction *Export* de l'application CAO pour transférer les symboles dans un fichier texte et les importer dans l'éditeur de symboles.

### 3.5.6 Ajouter des symboles durant l'édition du programme IL

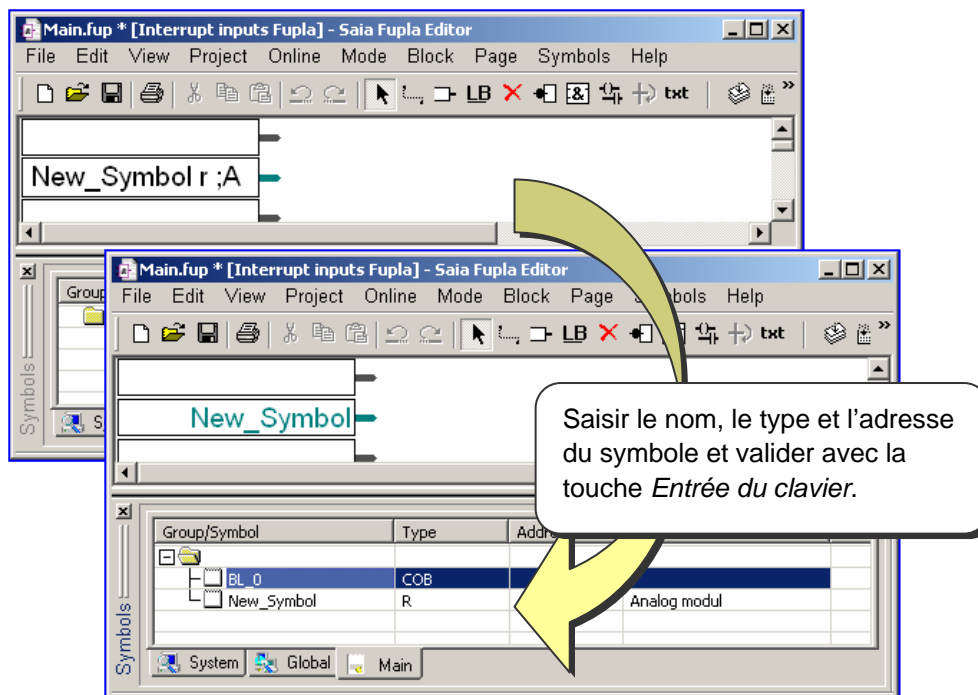
Il est possible d'ajouter chaque nouveau symbole directement depuis l'éditeur. Introduire l'instruction et son symbole puis ajouter « = », le type, l'adresse et le commentaire sur la même ligne. En appuyant sur la touche *Entrée* du clavier, ce symbole s'ajoute automatiquement à la liste, comme dans l'exemple ci-dessous.



### 3.5.7 Ajouter des symboles durant l'édition du programme Fupla

Depuis le champs d'entrées ou sorties, l'éditeur Fupla supporte aussi l'introduction d'un nouveau symbole pour l'éditeur de symboles. Veuillez respecter la syntaxe suivante pour y parvenir.

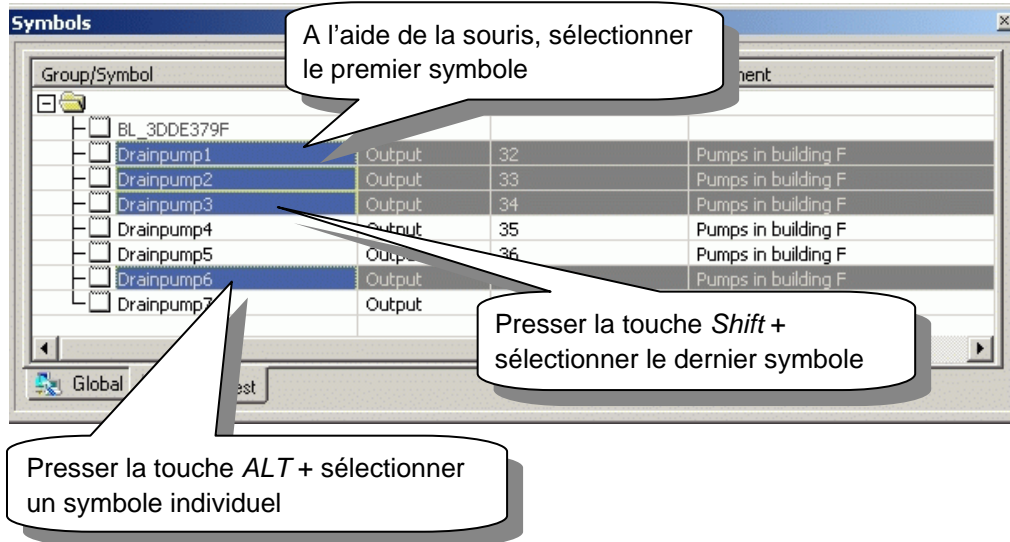
Syntaxe: *NomDuSymbole* Type [Adresse] [;Commentaire]



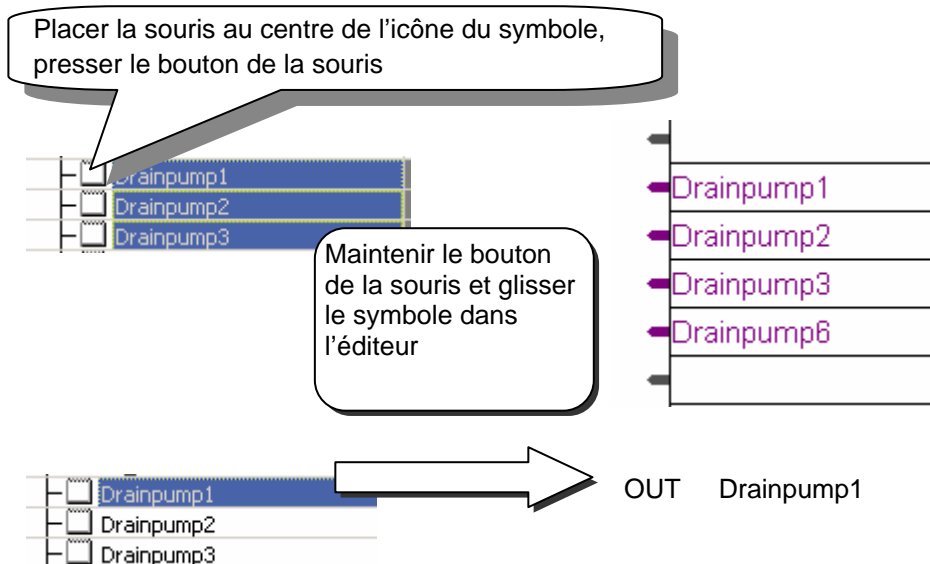
### 3.5.8 Transfert de symboles

Pour éviter d'introduire plusieurs fois les noms des symboles dans un programme et de prendre le risque de faire une erreur de frappe, il est possible de sélectionner un ou plusieurs symboles dans l'éditeur de symbole pour le glisser dans le programme Fupla ou IL.

**Exemple:** pour sélectionner plusieurs symboles



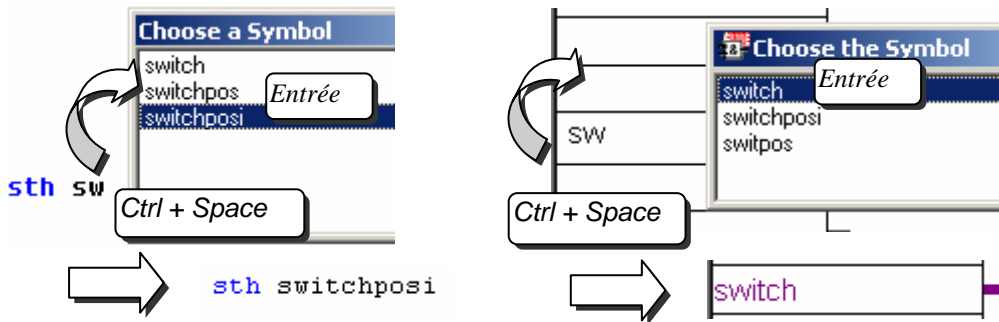
**Exemple:** pour glisser le symbole dans l'éditeur Fupla ou IL



### 3.5.9 Reconnaissance automatique des symboles

L'emploi de noms de symboles longs facilite la lecture des programmes. Par contre, cela peut être long et monotone de les éditer à chaque fois qu'ils interviennent dans le programme. Aussi est-il préférable d'écrire les premières lettres du symbole et de chercher tous les symboles commençant par ces mêmes lettres, à l'aide des touches *Ctrl* et *Barre d'espace*:

**Exemple:**



### 3.5.10 Allocation automatique d'adresse

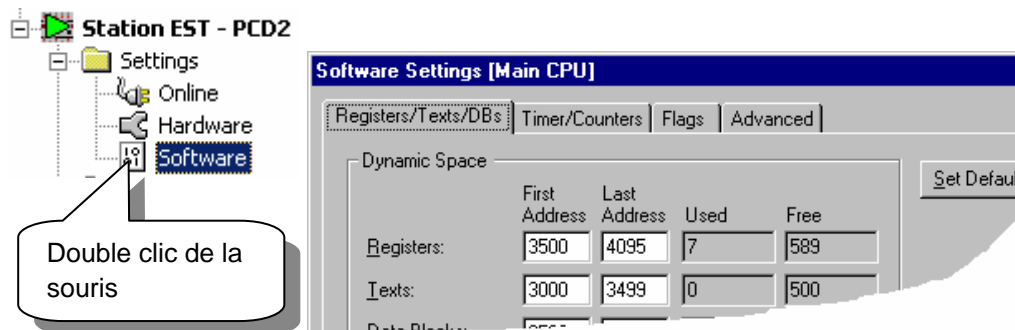
Jusqu'à présent, nous avons toujours déclaré les symboles comme suit:

Nom du symbole    Type    Adresse    Commentaire

**Exemple :**    Pumpspeed                    R                    2000                    ;Débit en l/min

Rappelons que les symboles, autre que les entrées ou sorties, ne nécessitent aucune adresse. Le PG5 se charge d'attribuer lui-même les adresses des symboles lors de la construction du programme (*Build*), selon le principe de l'allocation dynamique des ressources. Plus précisément, le PG5 recherche dans ses paramètres logiciels la plage d'adresses réservée à ce type de symbole et lui attribue automatiquement une adresse.

**Exemple:**    Pumpspeed                    R                    ;Débit en l/min

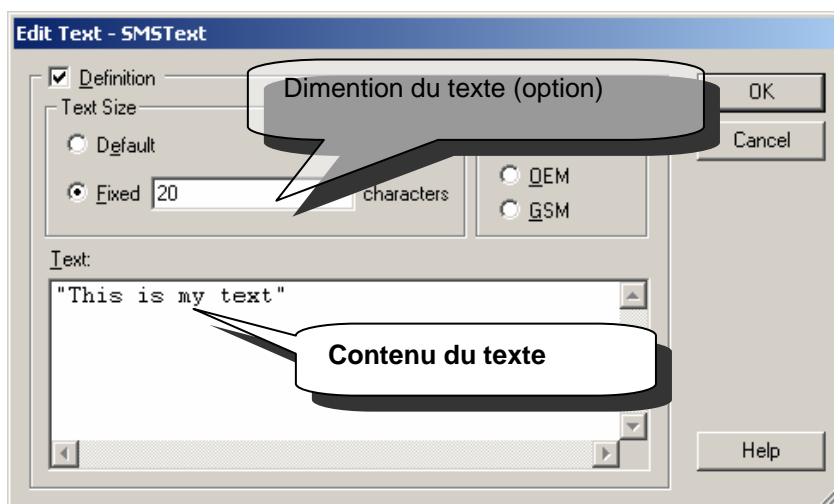
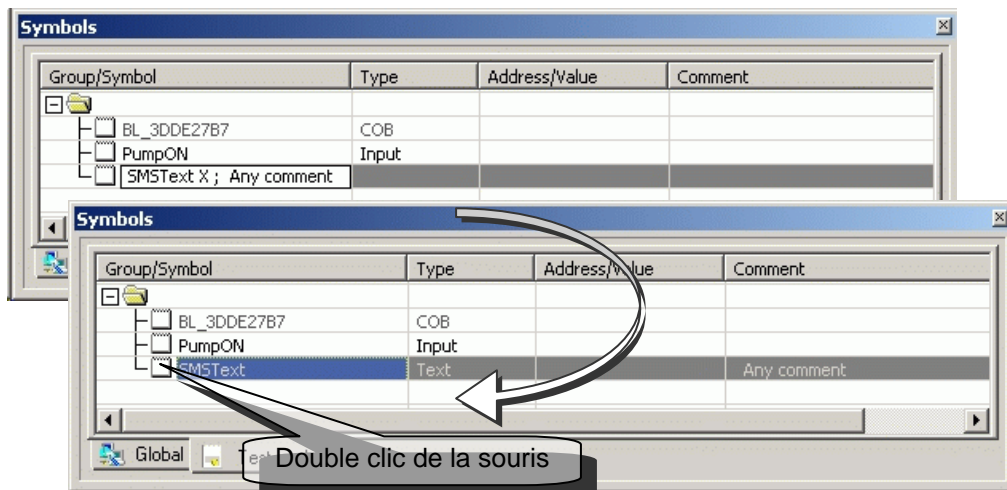


A la construction du programme, le registre sera numéroté avec une adresse comprise entre 3 500 à 4 095. Cette plage d'adresses est réservée à l'aide de la fenêtre *Software Settings*.

### 3.5.11 Saisie de textes

Avant de définir le contenu d'un texte, il faut d'abord le définir en ajoutant un symbole avec de type Text ou X après le nom de symbole.

Exemple:

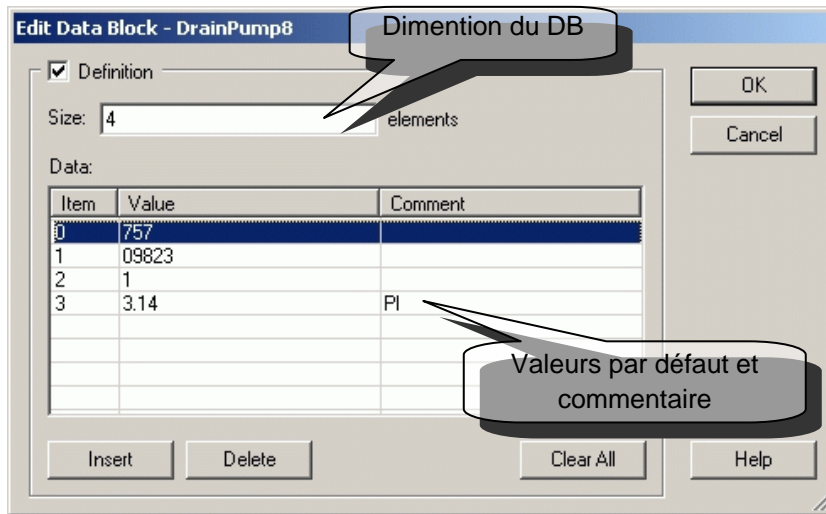


**Remarque :**

Ne pas oublier les guillemets! Si non le texte est refusé par le PG5.

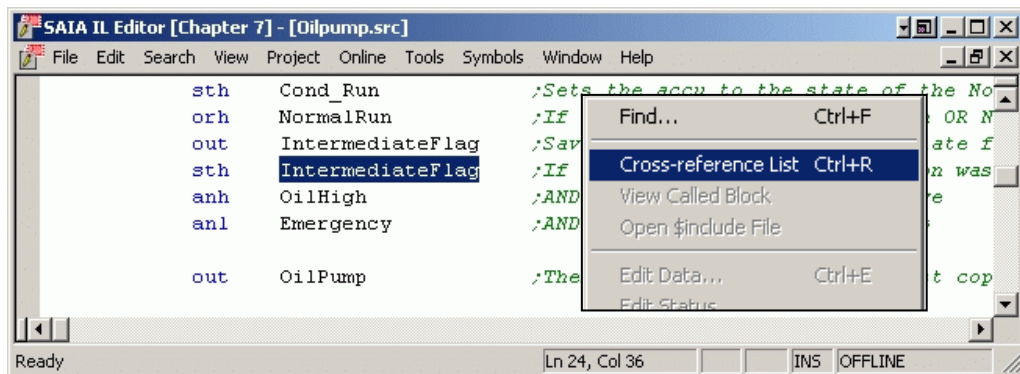
### 3.5.12 Saisie de blocs de données

Les blocs de données ont leur propre éditeur. Consultez l'aide pour en savoir davantage.



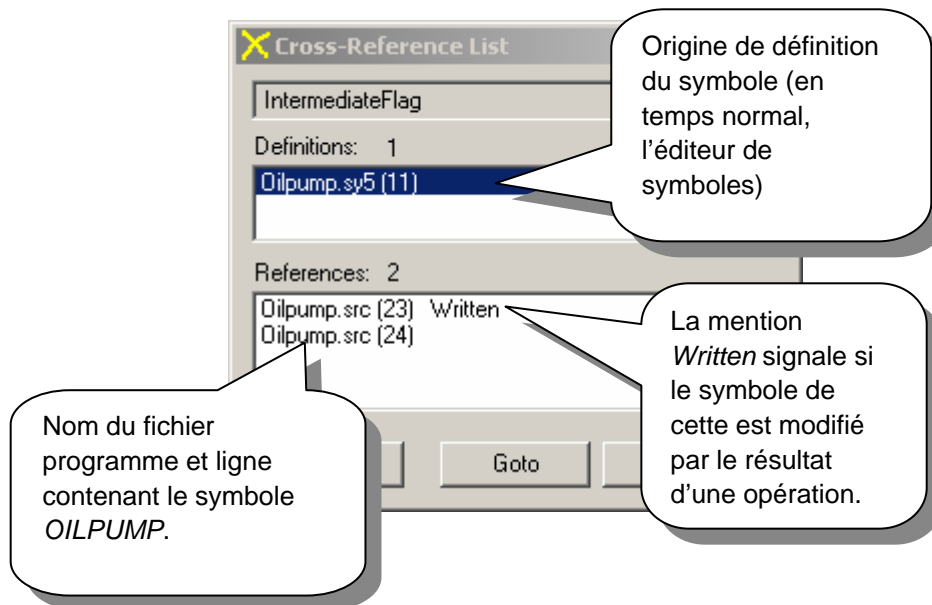
### 3.5.13 Recherche de symbole

Il est fréquent d'utiliser un symbole à plusieurs reprises dans le fichier programme, voir avec des fichiers distincts. Après la construction du programme, marquez un symbole et pressez le bouton droit de la souris pour ouvrir le menu de contexte et sélectionnez *Cross-Reference List...* pour démarrer la fonction de recherche croisée.



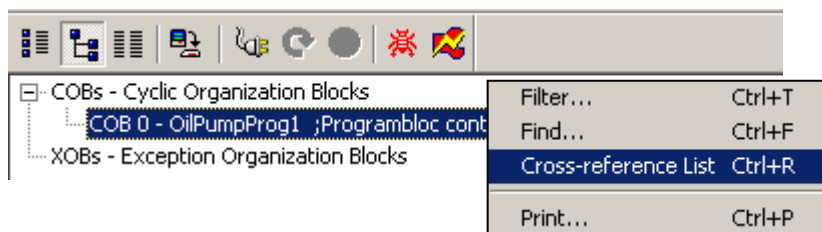
La fenêtre ci-dessous affiche les noms des fichiers, numéros de ligne (entre parenthèses) et occurrences d'un symbole donné. Un double clic sur l'une de ces informations a pour effet d'ouvrir le fichier programme et de placer le curseur sur le symbole.





Cet outil ne fonctionne pas seulement avec l'éditeur IL et Fupla, mais aussi avec les différentes vues du gestionnaire de projet.

**Exemple:** *Block Structure View*

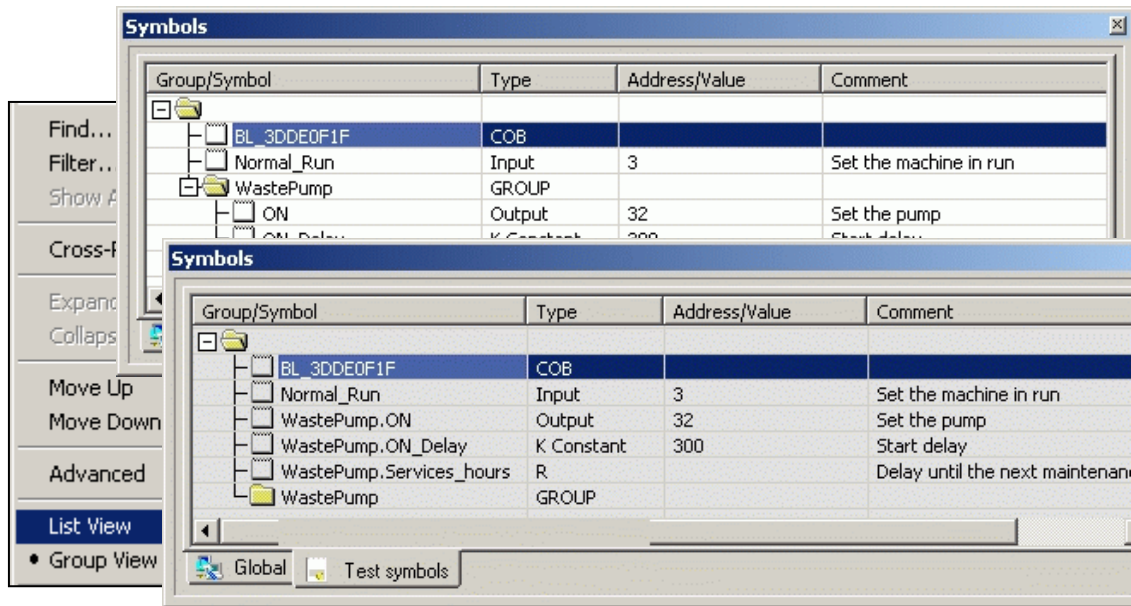


### 3.5.14 Organisation des symboles

Les symboles sont organisés selon leur ordre de saisie. Les symboles saisis au même moment restent groupés, même si de nouveaux symboles sont ajoutés par la suite.

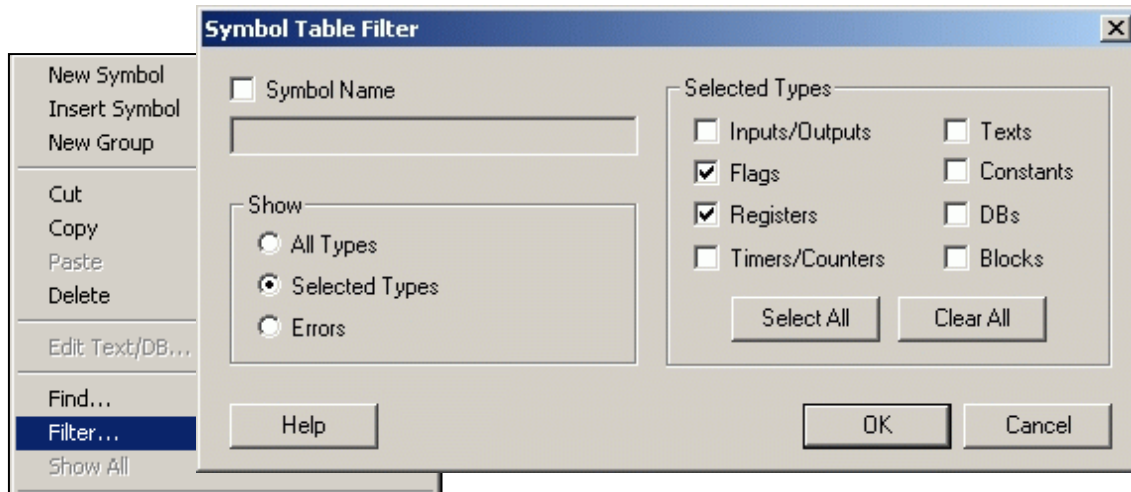
### 3.5.15 Réorganisation de la vue Liste

Il est possible de présenter les symboles avec une vue en groupe (*Group View*) ou en Liste (*List View*). La sélection de l'un des onglets de la fenêtre ci-dessous permet de trier les symboles par nom, type, adresse ou commentaire.



### Filterer les symboles

L'ordre original est rétabli dès à la sélection d'une vue en groupe. Si la liste comporte de nombreux symboles, il est parfois commode de n'afficher que certains types ou nom. C'est le rôle de la fonction de filtre *Filter*.



La fonction ci-dessus sélectionne les symboles à filter. Dès que le filtre est activé, les symboles changent d'icône.

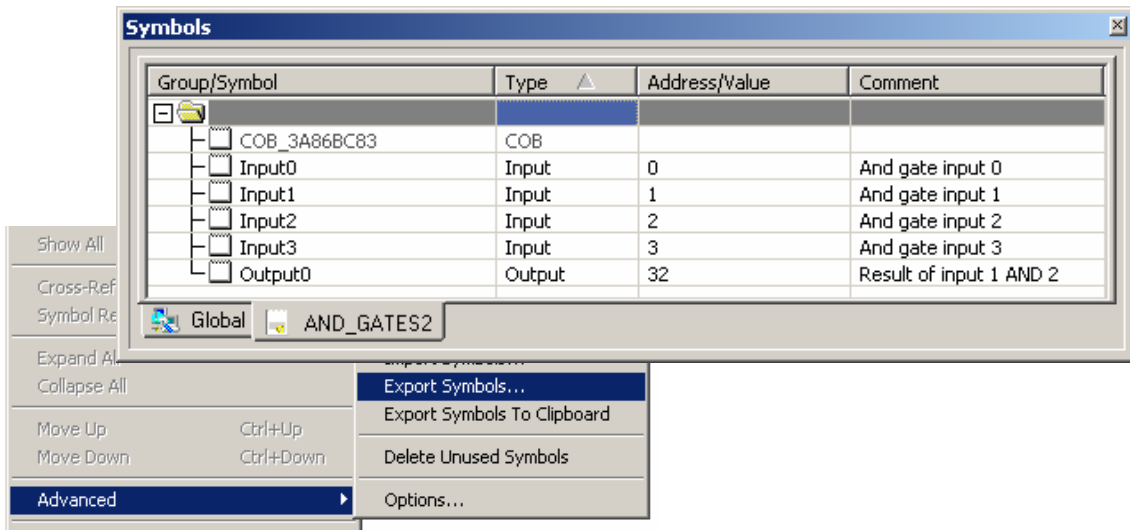
**Exemple :** Affichage complet des symboles par l'éditeur => [Icon]  
 Affichage partiel (filtré) des symboles => [Icon]

### 3.5.16 Exportation de symboles

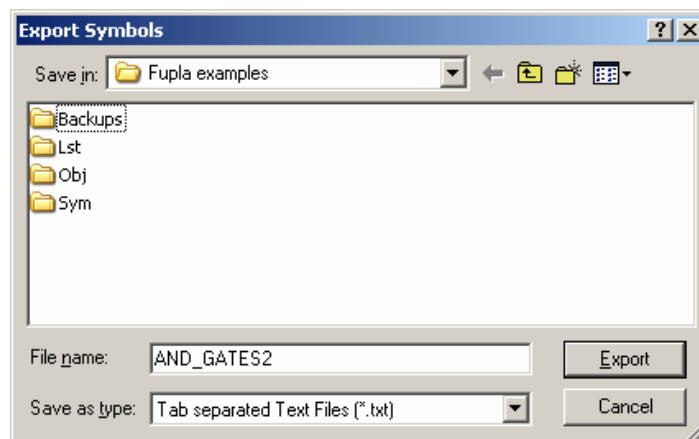
La liste des symboles d'un programme peut être exportée vers d'autres applications tel que Excel, Visiplus ou Winword pour élaborer votre rapport de mise en service par exemple.

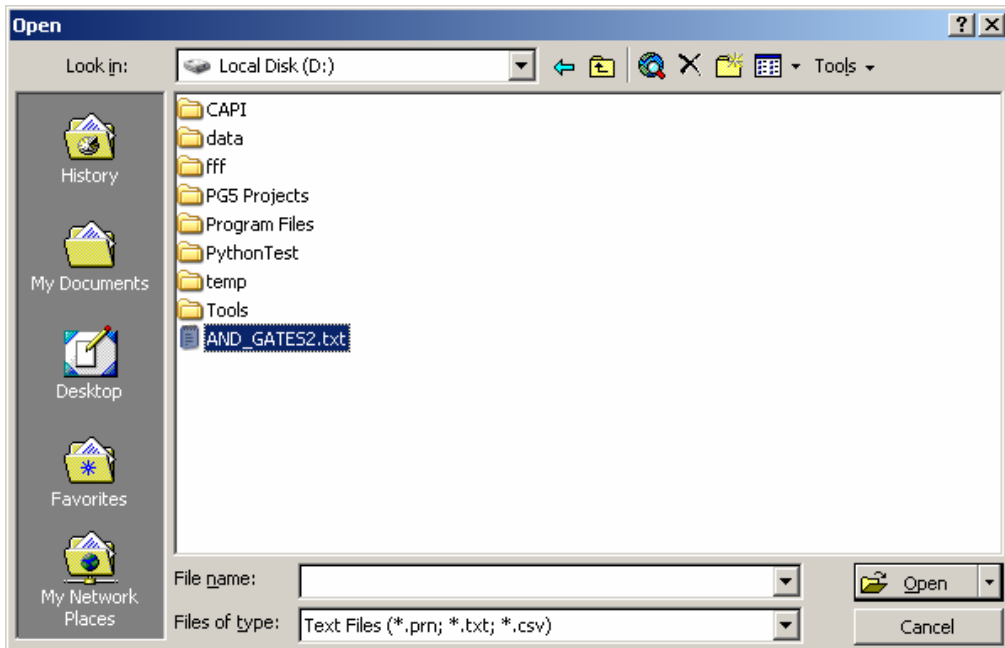
**Exemple:** exportation de symboles vers Excel

Sélectionner le menu de contexte *Export Symbols* de l'éditeur de symboles

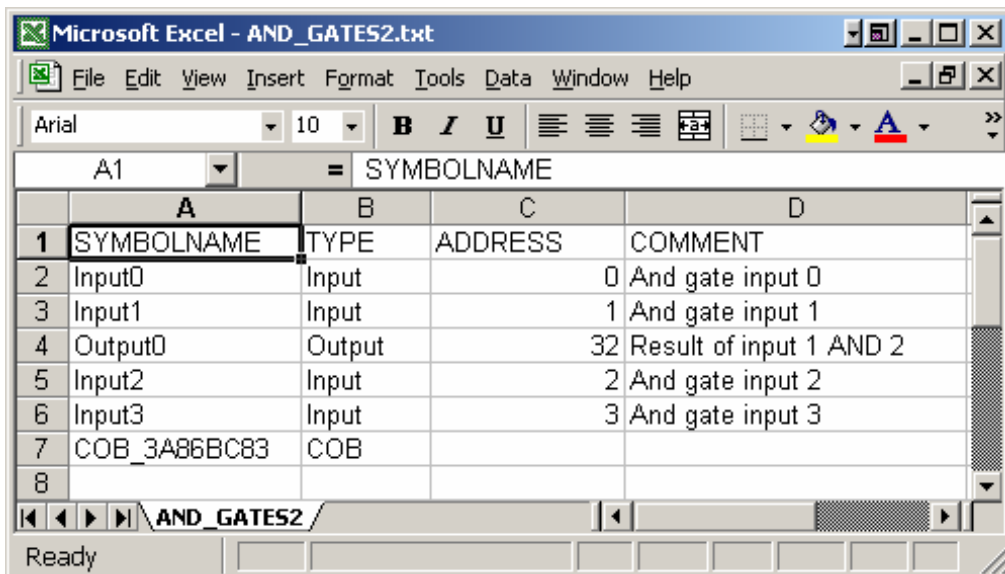


Pour exporter une liste de symboles vers Excel nous recommandons vivement d'employer le format *Tab separated Text Files (\*.txt)*. vous obtiendrez de meilleurs résultat qu'avec l'usage de l'option *Excel files (\*.xls)*.





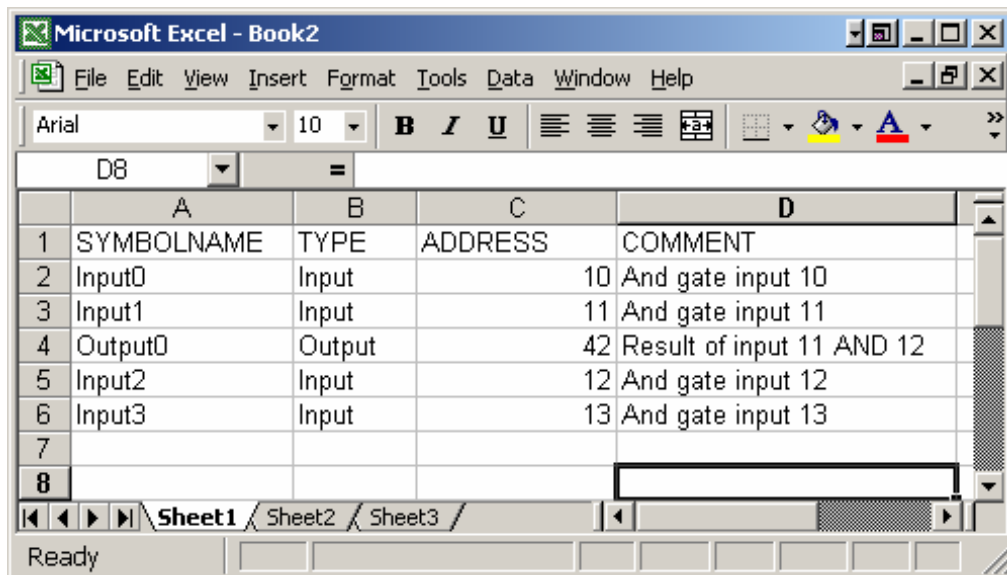
Démarrer Excel et ouvrir le fichier texte avec les symboles exportés



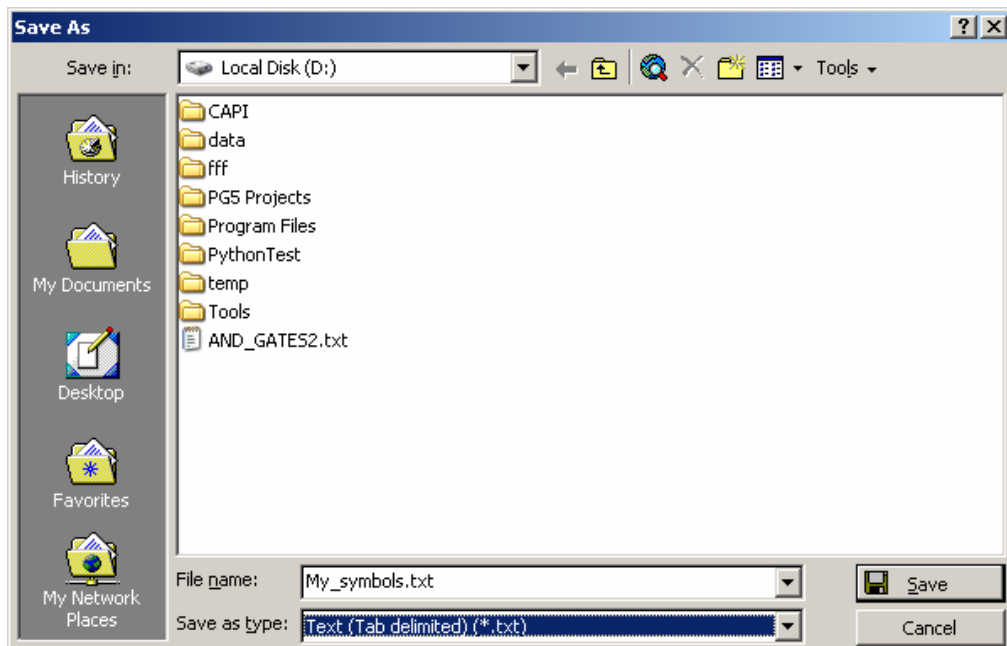
### 3.5.17 Importation de symboles

Il est aussi possible de préparer une liste de symboles dans l'éditeur Excel et de les importer dans le projet PG5. Pour y parvenir, préparer un fichiers de symboles comme l'exemple ci-dessous et le sauvegarder avec le format:

*Text(Tab delimited) (\*.txt)*

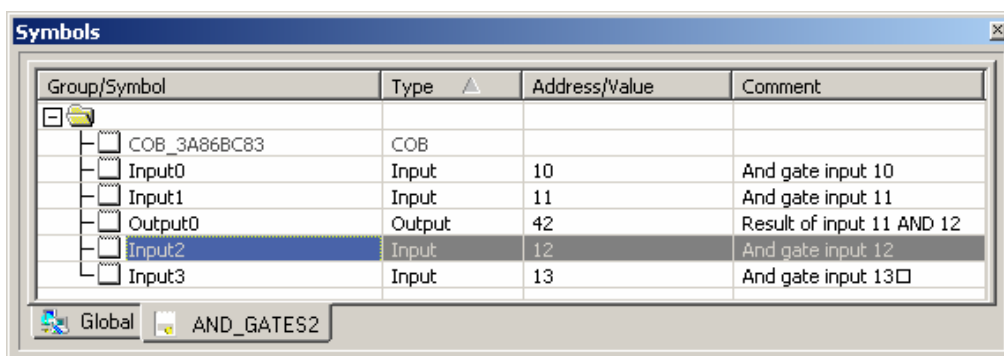
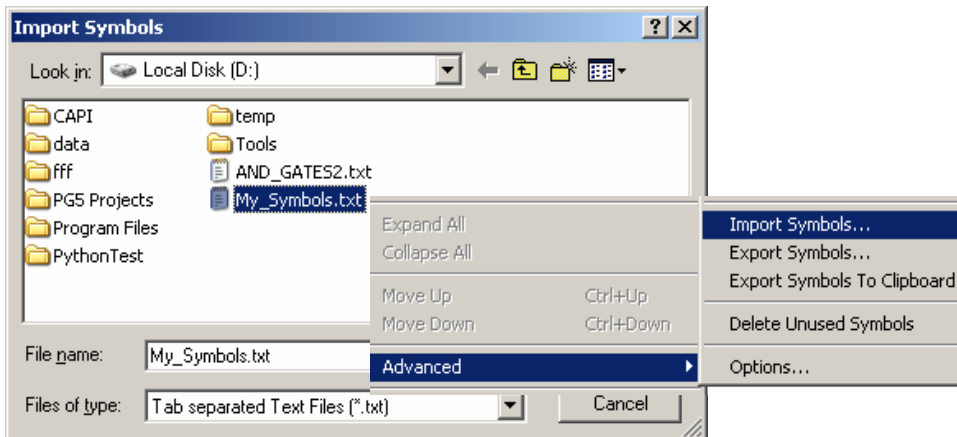


	A	B	C	D
1	SYMBOLNAME	TYPE	ADDRESS	COMMENT
2	Input0	Input	10	And gate input 10
3	Input1	Input	11	And gate input 11
4	Output0	Output	42	Result of input 11 AND 12
5	Input2	Input	12	And gate input 12
6	Input3	Input	13	And gate input 13
7				
8				



Dans l'éditeur de symboles PG5, sélectionner le menu de contexte *Advanced*, *Import Symbols*, sélectionner le fichier et importer.

**En cas de difficulté veuillez vous assurer que votre fichier Excel et bien refermé.**



### 3.5.18 Initialisation des symboles

Il y a deux possibilités pour initialiser les symboles utilisés par le PCD:

- ♣ initialisation lors du démarrage à froid de l'automate (mise sous tension)
- ♣ initialisation lors du chargement du programme dans l'automate

#### Au démarrage à froid

L'initialisation de symboles lors du démarrage à froid de l'automate est supportée par le bloc XOB 16. Ce bloc est traité seulement lors du démarrage à froid de l'automate. L'utilisateur écrit un programme pour initialiser les symboles dans le bloc XOB 16.

**Exemple:** initialiser un flag et un registre lors du démarrage à froid du PCD

#### Programme IL:

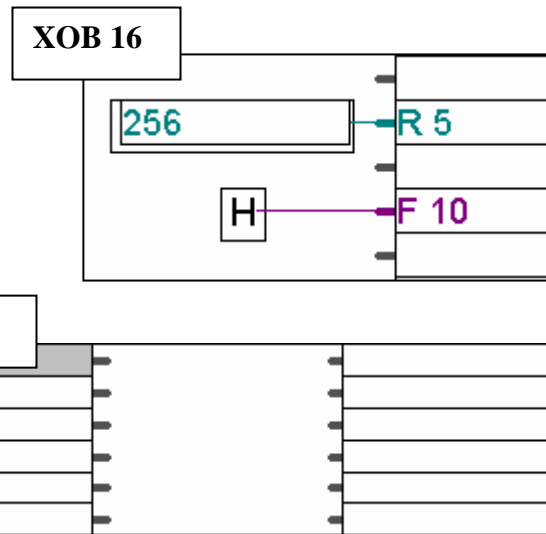
```
XOB 16 ;Bloc démarrage à froid
LD R 5 ; R 5 = 256
    256
SET F 10 ;F 10 = 1

EXOB

COB 0 ;Bloc cyclique
    0
...
;Votre programme
...

ECOB
```

#### Programme Fupla:



Pour de plus amples informations concernant les blocs COB et XOB, veuillez vous reporter au chapitre 5 de ce document.

#### Au chargement du programme

Pour initialiser un symbole lors du chargement du programme dans le PCD, faire suivre l'adresse du symbole par := puis la valeur d'initialisation.

#### Exemple:

Group/Symbol	Type	Address/Value	Com
BL_3E315C55	COB	0	
SymbolA	R	5:= 256	
SymbolB	F	10:=1	



#### Attention

Ne pas oublier d'activer l'option suivante lors du chargement du programme:



First-time Initialisation Data

### 3.5.19 Noms de symbole

Rappelons qu'il s'agit des noms attribués aux ressources du PCD (entrées, sorties, indicateurs, registres, blocs d'organisation cyclique...). Ils ne peuvent dépasser 80 caractères et ne tiennent pas compte de la casse. Ainsi, le PG5 ne fait pas de différence entre un symbole *MotorOn* et *MOTORON*.

Un symbole doit obligatoirement commencer par une lettre (a-z, A-Z), jamais un chiffre. Dans le corps du symbole, vous pouvez librement combiner nombres, lettres et soulignés « \_ », à l'exclusion du caractère espace.

**Vous ne pouvez pas utiliser de mots réservés en guise de noms de symbole.**

### 3.5.20 Mots réservés

Les mots suivants sont réservés et ne peuvent pas être utilisés :

- ♣ Déclarations en Assembleur : PUBL, EXTN, EQU, DEF, LEQU, LDEF, MACRO, ENDM, EXITM...
- ♣ Codes de commande et notations abrégées des différents types de données du PCD : I, O, F, R, C, T, K, M, COB, FB, TEXT, X, SEMA, DB,
- ♣ Codes spéciaux d'instruction MOV : N, Q, B, W, L, D,
- ♣ Codes conditionnels : H, L, P, N, Z, E,
- ♣ Tous les mnémoniques d'instruction,
- ♣ Symboles prédéfinis,
- ♣ Symboles internes réservés à l'allocation automatique des ressources, commençant par un souligné. Exemple :  
\_\_\_\_\_TEXT, \_\_\_\_\_F
- ♣ Symbole interne \_\_CSTART\_\_, utilisé pour les affectations \$\$.



## Table des matières

---

<b>TABLE DES MATIERES</b>	<b>1</b>
<b>4 PROGRAMMER AVEC FUPLA</b>	<b>3</b>
4.1 Au sommaire de ce chapitre	3
4.2 Préparer un projet Fupla	4
4.3 Créer un nouveau projet	4
4.4 Organisation d'une fenêtre Fupla	5
4.5 Edition des symboles	6
4.5.1 Ajouter un nouveau symbole dans la liste <i>Symbols</i>	7
4.5.2 Modes d'adressage des symboles	8
4.5.3 Utiliser un symbole de la liste dans le programme Fupla	9
4.5.4 Symboles local et global	10
4.6 Edition des connecteurs	11
4.6.1 Placer un connecteur	11
4.6.2 Editer le symbole à l'intérieur du connecteur	11
4.6.3 Méthode rapide pour placer un symbole et son connecteur	11
4.6.4 Glisser, Copier/Coller, effacer un symbole	11
4.6.5 Copier/Coller, effacer un connecteur	12
4.6.6 Connecteur extensible (stretchable)	12
4.6.7 Déplacer un connecteur verticalement	12
4.7 Edition des Fbox Fupla	13
4.7.1 FBox selector	13
4.7.2 Edition d'une Fbox	14
4.7.3 Edition d'une Fbox extensible (stretchable)	14
4.7.4 Edition d'une inversion logique	14
4.7.5 Dynamisation	15
4.7.6 Commentaires	15
4.7.7 Fbox Help	15
4.8 Liaisons entre les FBox et connecteurs	16
4.8.1 Liaison par déplacement de la Fbox	16
4.8.2 Liaison avec routage automatique	16
4.8.3 Liaison multiple avec routage automatique	16
4.8.4 Liaison de toutes les entrées, sorties d'une Fbox à des connecteurs	16
4.5.3 Suppression de lignes, Fbox, connecteurs ou symboles	17
4.8.5 Déplacer une Fbox, un connecteur verticalement sans défaire les liens	17
4.8.6 Insérer une Fbox sans défaire le lien.	17
4.8.4 Règles à respecter	17
4.9 Edition de pages Fupla	18
4.9.1 Ajouter une page	18
4.9.2 Supprimer une page	18

4.9.3	Navigation entre pages	18
4.9.4	Documentation de pages	19
4.9.5	Traitement du programme par le PCD	19
<b>4.10</b>	<b>Copier, coller</b>	<b>20</b>
4.10.1	Copier, coller une partie de programme	20
4.10.2	Copier, coller des symboles	20
<b>4.11</b>	<b>Exporter, importer des pages</b>	<b>21</b>
4.11.1	Exportation de pages	21
4.11.2	Importation de pages	22
<b>4.12</b>	<b>Édition d'un premier programme Fupla</b>	<b>24</b>
4.12.1	Objectif	24
4.12.2	Méthode	24
4.12.3	Programmation	26
<b>4.13</b>	<b>Construction du programme (Build)</b>	<b>27</b>
<b>4.14</b>	<b>Charger le programme dans le PCD</b>	<b>28</b>
<b>4.15</b>	<b>Débugger un programme (Debug)</b>	<b>28</b>
4.15.1	Go On/Offline, Run, Stop, Step-by-step	28
4.15.2	Point d'arrêt	29
4.15.3	Visualiser les symboles ou leurs adresses	30
4.15.4	Visualiser l'état des symboles avec Fupla	30
4.15.5	Editer les symboles en ligne	30
4.15.6	Visualiser l'état des symboles avec le <i>Watch window</i>	31
4.15.7	Mise à l'heure de l'horloge PCD	32
<b>4.16</b>	<b>Fenêtres d'ajustage</b>	<b>33</b>
4.16.1	Types de paramètres d'ajustage	34
4.16.2	Initialisation des Fbox CVC	35
4.16.3	Fbox CVC avec paramètres d'ajustage	36
4.16.4	Mini application CVC	36
4.16.5	Paramètres après le chargement du programme	37
4.16.6	Ecriture de paramètres en ligne.	37
4.16.7	Lecture des paramètres en ligne.	38
4.16.8	Rétablir les paramètres standards	38
4.16.9	Définir les symboles des paramètres d'ajustage	39
4.16.10	Définir les adresses des paramètres d'ajustage	40
<b>4.17</b>	<b>Mise en service d'un module analogique</b>	<b>41</b>
4.17.1	Saisie d'une mesure analogique	41
4.17.2	Exemple pour les entrées analogiques PCD2.W340	42
4.17.3	Exemple pour les sorties analogiques PCD2.W610.	43

## 4 Programmer avec Fupla

---

### 4.1 Au sommaire de ce chapitre

L'éditeur Fupla, est le moyen le plus simple et le plus rapide de s'initier à la programmation des automates PCD. Fupla est un terme anglais « *FUnction PLAN* » qui désigne un environnement de programmation graphique où l'utilisateur dessine les programmes à l'aide de plusieurs centaines d'éléments graphiques prédéfinis (FBox, Function Box). Ces Fbox sont organisées sous la forme de bibliothèques couvrant les applications de base mais aussi des fonctions plus spécialisées à certains domaines de métier. Par exemple la bibliothèque CVC pour le Chauffage, Ventillation et Climatisation , la bibliothèque modem pour réaliser un réseau d'automates supportant l'échange de données sur une ligne téléphonique (analogique, ISDN, GSM, GPRS), les messages SMS, Pager et DTMF. Bien d'autres bibliothèques comme pour les réseaux de communication LON, EIB ou les produits Belimo ... sont aussi disponibles.

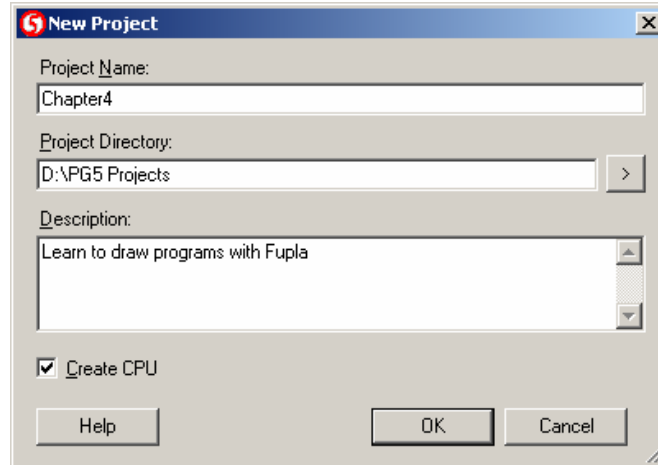
Le grand avantage du Fupla réside dans le fait que l'utilisateur peut mettre en service un PCD sans avoir à écrire une seule ligne de programme et sans avoir de connaissance particulière de programmation.

## 4.2 Préparer un projet Fupla

Pour réaliser un exemple, nous conseillons de préparer un nouveau projet dans lequel nous préparons les fichiers pour éditer le programme Fupla.

## 4.3 Créer un nouveau projet

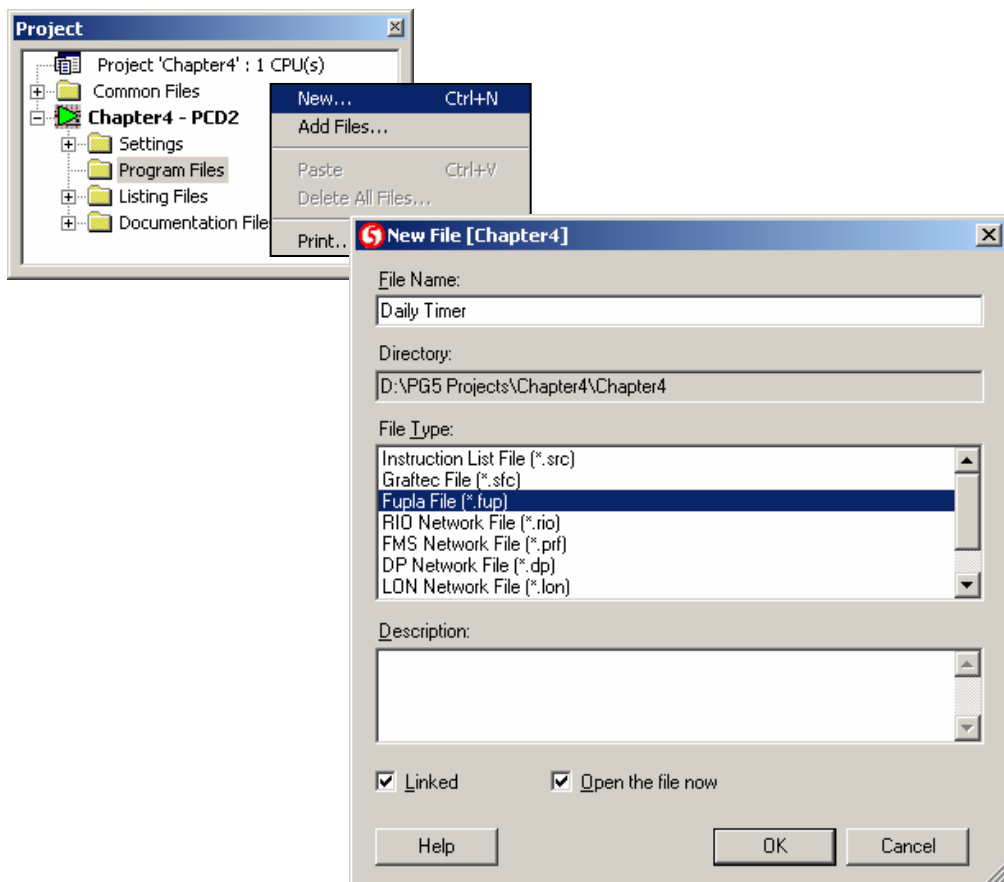
Dans la fenêtre *SAIA Project Manager*, sélectionner le menu *Project, New...* et créer le nouveau projet.



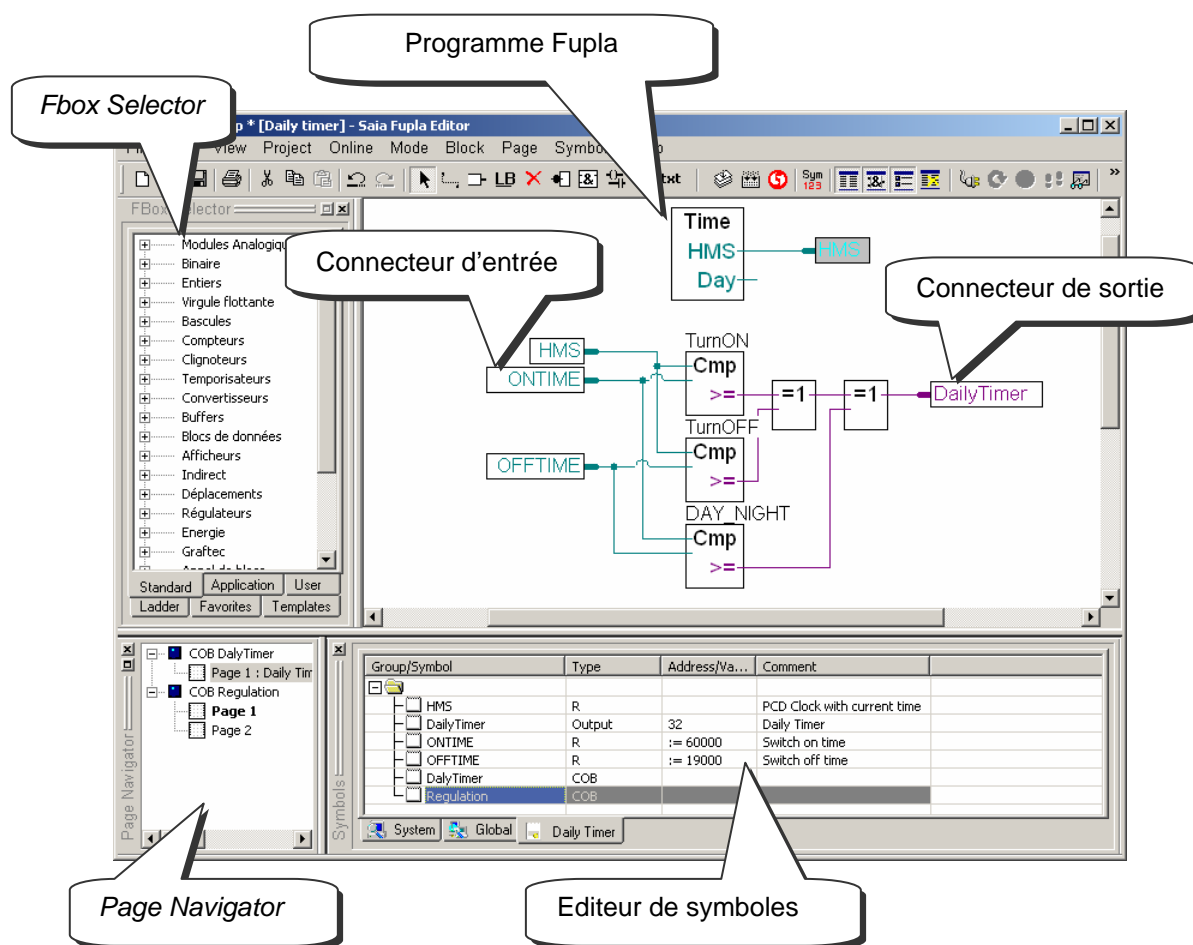
Pour ajouter un nouveau fichier programme dans le projet, marquer le répertoire *Program Files*, presser le bouton droit de la souris et sélectionner le menu *New...* ou sélectionner le bouton *New File* de la barre d'outils:



New File



### 4.4 Organisation d'une fenêtre Fupla



Le PCD lit les informations représentées par les connecteurs d'entrée, les évalue selon le programme et écrit le résultat dans les connecteurs de sortie. Les symboles utilisés par le programme sont tous rassemblés dans la fenêtre *Symbols*. De manière générale tous les symboles sont autorisés dans les connecteurs d'entrées et de sorties, excepté les symboles de type *input* et *constante*. Les entrées digitales et constantes sont des informations uniquement disponibles en lecture, que l'on ne peut utiliser que dans les connecteurs d'entrée.

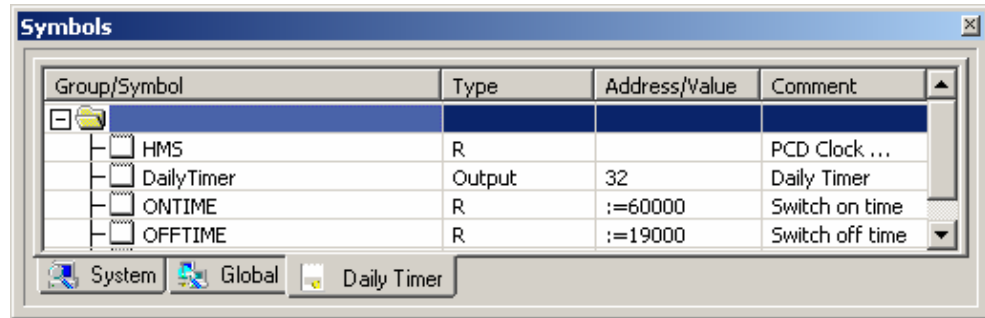
Au centre de la page nous avons le programme constitué de différentes Fbox graphiques prélevées dans la fenêtre *Fbox selector*. Les liens représentent les échanges d'informations entre les différentes Fbox. La couleur des liens définit le type des informations : violet pour les informations logiques, bleu pour les nombres entiers, jaune pour les nombres flottants. Les informations de types différents ou de couleurs différentes ne peuvent pas être liées ensemble sans les convertir dans un type commun. (Groupe de FBox *Standard, Convertisseurs*)

Si le programme est composé de plusieurs pages, la fenêtre *Page Navigator* permet d'ajouter, supprimer des pages ou de se déplacer rapidement dans la structure du programme.

## 4.5 Edition des symboles



Show Hide  
Symbols Editor



La fenêtre *Symbols* rassemble les symboles du programme dans une liste qui peut être affichée avec le bouton *Show Hide Symbols Editor* ou le menu *View Symbol Editor*. Chaque ligne définit toutes les informations relatives à une entrée, sortie, registre et constitue un symbole:

### Le symbole.

Le symbole est un nom qui désigne l'adresse d'une entrée, sortie, indicateur, registre, ... Il est conseillé d'utiliser ce nom ou symbole pour éditer le programme et non pas directement l'adresse de l'indicateur ou du registre. Ainsi toute correction portant sur l'adresse ou le type de cette information peut être supportée depuis la fenêtre *Symbols*. Il n'est alors pas nécessaire de reporter la correction dans les connecteurs du programme Fupla mais uniquement dans la fenêtre *Symbols*. Cela évite d'oublier de corriger le contenu d'un connecteur du programme et de créer une erreur difficile à trouver.

### Syntaxe pour le nom d'un symbole.

Le premier caractère est toujours une lettre suivie par d'autres lettres, chiffres ou du caractère souligné. Il est conseillé d'éviter les caractères tel que ö,è,ç,... Les majuscules ou minuscules sont sans influence significative, Motor\_On est MOTOR\_ON sont deux symboles identiques.

### Le type.

Définit le type de symbole: entrées (I), sortie (O), registre (R), compteur (C), temporisateur (T), texte (X), DB, ...

### L'adresse.

Chaque type de symbole compte de nombreuses adresses disponibles:

Entrées et sorties: selon les modules E/S insérés dans le PCD

Indicateurs: F 0, ..., F 8191

Registres: R 0, ..., R 4095<sup>1</sup>, 16383<sup>2</sup>

Temporisateurs, Compteurs: T/C 0, ..., T/C 1599

...

### Le commentaire.

Ce commentaire est lié à son symbole et peut être affiché dans la page Fupla. Placer la souris sur le connecteur, alors la définition complète du symbole est affichée dans une bulle.

<sup>1</sup> Tous les PCD

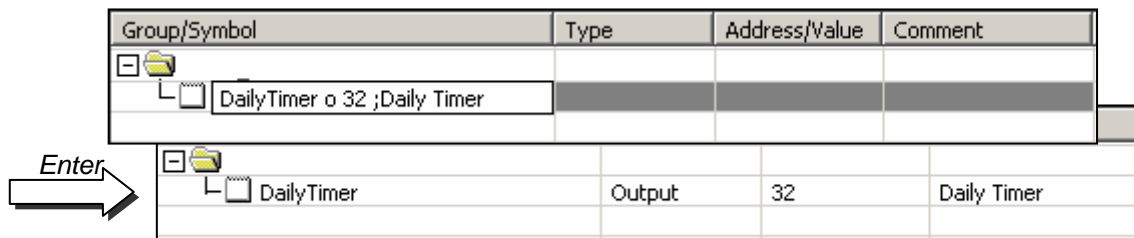
<sup>2</sup> PCD2.M480,PCD3

### 4.5.1 Ajouter un nouveau symbole dans la liste *Symbols*

#### Méthode simple

Pour ajouter un symbole dans la liste, ouvrir la fenêtre *Symbols*, placer la souris au centre de la fenêtre et sélectionner le menu de contexte *Insert Symbol* avec le bouton droit de la souris. Puis compléter les champs *Group/Symbol*, *Type*, *Address/Value* et *Comment*.

#### Méthode rapide 1



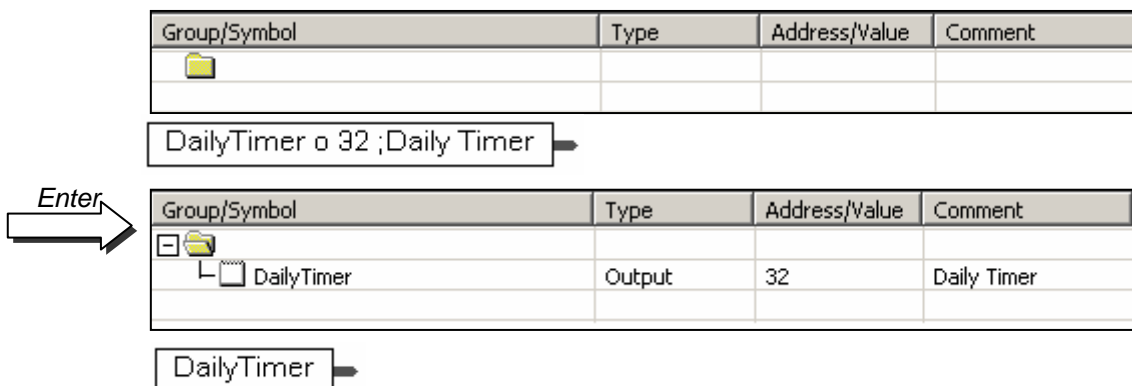
Il est aussi possible d'introduire les variables des divers champs d'informations depuis le champs *Group/Symbol*. C'est plus pratique et plus rapide. Voir l'exemple ci-dessus.

Syntaxe à respecter :

[nom\_du\_symbole] + [ ]<sup>\*</sup> + [Type] + [ ]<sup>\*</sup> + [Adresse] + [ ; ] + [Commentaire]

Si le nouveau symbole est défini selon la syntaxe ci-dessus, la sélection de la touche *enter* du clavier permet de placer automatiquement les informations dans les champs correspondants.

#### Méthode rapide 2



Les nouveaux symboles peuvent être aussi ajoutés depuis les connecteurs Fupla, selon la syntaxe suivante:

[nom\_du\_symbole] + [ ]<sup>\*</sup> + [ Type] + [ ]<sup>\*</sup> + [Adresse] + [ ; ] + [Commentaire]





La sélection de la touche *enter* du clavier permet de placer automatiquement le nouveau symbole dans la liste *Symbols*. Seulement si la définition du symbole est correcte.

<sup>\*</sup> [ ] représente un caractère avec un espace.

## 4.5.2 Modes d'adressage des symboles





La définition d'un symbole ne comprend pas obligatoirement toutes les informations présentées ci-dessus. Nous distinguons trois cas d'utilisation:

### Adressage absolu

Group/Symbol	Type	Address/Value	Comment
 			
  DailyTimer	Output	32	Daily Timer





Le symbole est défini avec un type, une adresse est optionnellement un commentaire. L'usage des adresses absolues représente un inconvénient lors de la correction du type ou de l'adresse. La correction ne peut pas être supportée à partir de la liste des symboles. La correction doit être reportée manuellement pour chaque connecteur du programme. Il est donc préférable d'utiliser un adressage symbolique ou dynamique.

### Adressage symbolique

Group/Symbol	Type	Address/Value	Comment
 			
  DailyTimer	Output	32	Daily Timer

Le symbole est défini avec un nom, un type, une adresse est optionnellement un commentaire. Les corrections du symbole, du type ou de l'adresse sont supportés depuis la liste des symboles et reportées automatiquement pour chaque connecteur du programme faisant usage de ce symbole.

### Adressage dynamique

Group/Symbol	Type	Address/Value	Comment
 			
  HMS	R		PCD Clock with

C'est un adressage symbolique ou l'adresse n'est pas définie. L'outil de programmation attribue automatiquement une adresse correspondant au type défini lors de la construction du programme. L'adresse est prélevée dans une plage d'adresses réservée par les *Settings Software*. (Voir dans le gestionnaire de projet.) Remarque, l'adressage dynamique est disponible avec les indicateurs, compteurs, temporisateurs, registres, textes, DB, COB, PB,FB,SB.

Par contre les adresses doivent toujours être définies pour les entrées, sorties et XOB.



### 4.5.3 Utiliser un symbole de la liste dans le programme Fupla

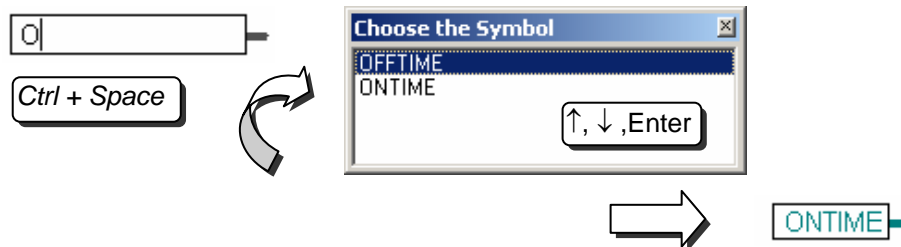
A l'édition du programme, les symboles déjà définis dans la fenêtre *Symbols* peuvent être utilisés de différentes manières:

#### Introduction du symbole au clavier.

Le nom du symbole est complètement introduit au clavier pour chaque instruction qui en fait usage. Cette manière de faire laisse la possibilité d'éditer le nom du symbole avec une erreur de frappe qui se manifestera seulement à la construction du programme.

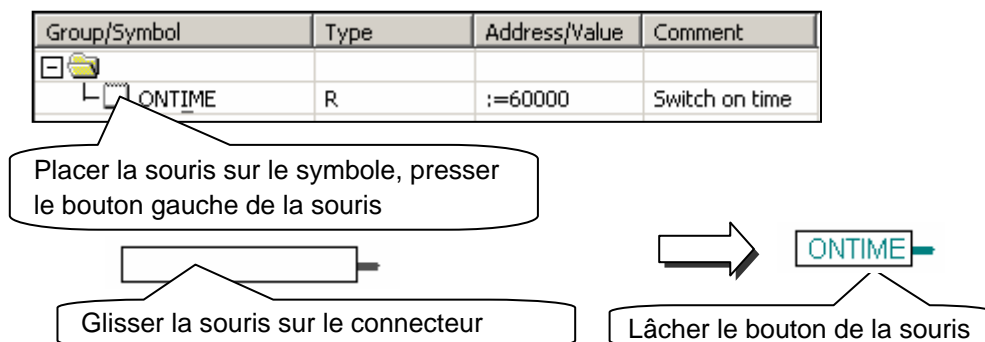
#### Introduction par recherche sélective.

Group/Symbol	Type	Address/Value	Comment
HMS	R		PCD Clock with current time
DailyTimer	Output	32	Daily Timer
ONTIME	R	:=60000	Switch on time
OFFTIME	R	:=19000	Switch off time



Seul les premiers caractères du nom du symbole sont introduits au clavier. La sélection simultanée des touches *Ctrl+espace* affiche une fenêtre avec la liste des symboles qui commencent avec les mêmes lettres que celles saisies précédemment. Il ne reste qu'à sélectionner le symbole désiré avec la souris ou les touches *↑, ↓* du clavier et valider la sélection avec la touche *Enter*.

#### Introduction par sélection et glissement



Cette manière de faire usage d'un symbole ne laisse aucune possibilité de faire une erreur de frappe. Dans la fenêtre *Symbols*, placer la souris sur la ligne de définition d'un symbole, presser le bouton gauche de la souris et le maintenir pressé. Glisser la souris au dessus d'un connecteur vide et lâcher le bouton de la souris. Le symbole sélectionné est automatiquement ajouté au connecteur pointé par la souris.

Il est aussi possible de glisser le symbole sur un espace libre de la page Fupla, cela permet d'ajouter automatiquement le connecteur et le symbole avec une seule manipulation.

### 4.5.4 Symboles local et global

La fenêtre de définition des symboles comporte deux répertoires : *Global* et *Local*



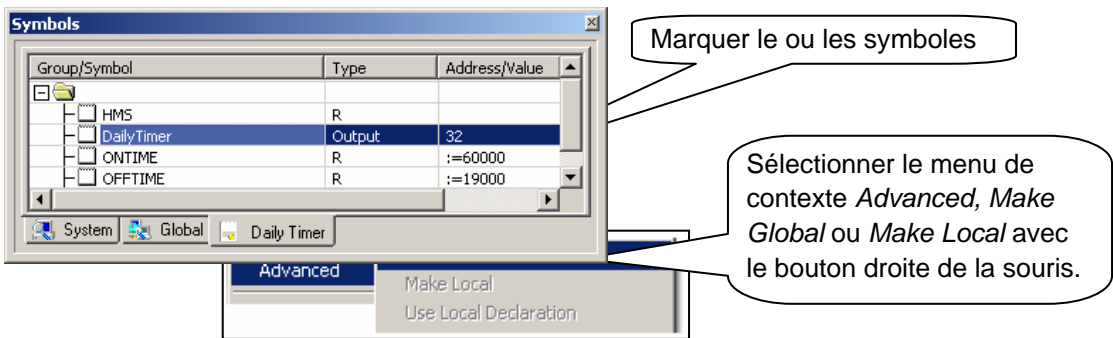
#### Définition

Les symboles locaux sont représentés dans un répertoire qui porte le nom du fichier qui les utilise. Ces symboles sont uniquement utilisables à l'intérieur ce fichier. (*Parking lot.src*)

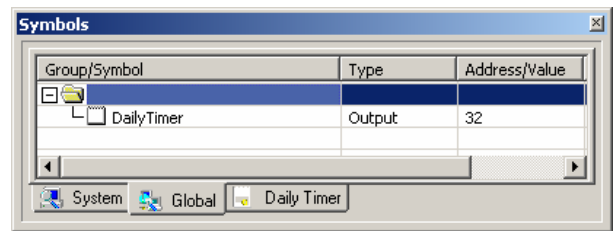
Les symboles globaux représentés dans le répertoire *Global* sont utilisables par tous les fichiers du CPU. (*Parking lot.src* et *Ventilation.src*)

#### Make Local/Global

Si nécessaire, les symboles de la fenêtre *Symbols* peuvent être déplacés du répertoire local vers le répertoire global ou inversement.



Le symbole est déplacé dans le répertoire *Global* ou *Local*



#### Remarque

Tout nouveau symbole défini directement depuis l'éditeur Fupla est ajouté dans un des répertoires *global* ou *local*, selon les présélections de l'option *Add symbols to Global table*. Voir menu de contexte *Advanced, Options* de la fenêtre *Symbols*

## 4.6 Edition des connecteurs

Des connecteurs d'entrée et sorties sont librement placés sur les pages Fupla, ils permettent de recevoir les symboles nécessaires au fonctionnement du programme décrit par les Fbox.

Il est possible que chaque nouvelle page offre encore par défaut une marge avec des connecteurs sur la gauche et la droite. Si vous préférez que les nouvelles pages soient présentées sans ces connecteurs et les placer vous-même selon votre convenance, veuillez désactiver l'option suivante avec le menu : *View, Options..., Layout, New pages with side connectors.*

Pour supprimer les connecteurs qui seraient éventuellement présents à gauche et droite de la page, sélectionner le menu : *Page, Remove Empty connectors.*

Pour placer à nouveau les connecteurs sur une page blanche, sélectionner le menu : *Page, Add Empty Side Connectors.*

### 4.6.1 Placer un connecteur



Place  
Connectors

Pour ajouter un connecteur et son symbole sur une page Fupla, sélectionner le bouton *Place Connectors* présent dans la barre des outils et positionner la souris sur la page Fupla. La sélection du bouton gauche de la souris permet d'ajouter un connecteur en lecture. Pour ajouter un connecteur en écriture, veuillez presser la touche *Shift* du clavier au moment de sélectionner le bouton de la souris. Le connecteur que vous venez d'ajouter est prêt pour recevoir un symbole, un curseur est représenté à l'intérieur du connecteur. Si vous ne souhaitez pas éditer immédiatement le symbole à l'intérieur du connecteur, sélectionnez la touche *ESC* du clavier et placer le connecteur suivant.

### 4.6.2 Editer le symbole à l'intérieur du connecteur

Pour éditer, modifier le symbole à l'intérieur d'un connecteur déjà présent sur la page Fupla, sélectionner le connecteur avec un double clic rapide de la souris, un curseur est représenté à l'intérieur du connecteur. Il est alors possible d'introduire le symbole avec sa définition complète.

Nous remarquons que les nouveaux symboles introduits dans les connecteurs sont automatiquement ajoutés à la liste des symboles présents dans la fenêtre *Symbols*.

### 4.6.3 Méthode rapide pour placer un symbole et son connecteur

Les symboles déjà présents dans la fenêtre *Symbols*, peuvent être glissés sur une surface libre de la page Fupla. Il en résulte un nouveau connecteur avec un symbole. Si le symbole est glissé sur une entrée ou sortie d'une Fbox, un connecteur d'entrée ou sortie est directement lié à la Fbox.

### 4.6.4 Glisser, Copier/Coller, effacer un symbole



La sélection de la surface représentée en rouge affecte uniquement le symbole. Il est possible de sélectionner le symbole à l'aide de la souris pour le glisser le copier coller vers un autre connecteur ou l'effacer. La sélection du bouton droit de la souris affiche le menu de contexte avec les opérations disponibles.

#### 4.6.5 Copier/Coller, effacer un connecteur



La sélection de la surface représentée en rouge affecte le connecteur et le symbole qu'il contient. La sélection du bouton droite de la souris affiche le menu de contexte avec les opérations disponibles.

#### 4.6.6 Connecteur extensible (stretchable)

Les connecteurs sont extensibles, cela signifie que le nombre de connecteurs peut être défini par un mouvement vertical de la souris.

Sélectionner le bouton Select Mode

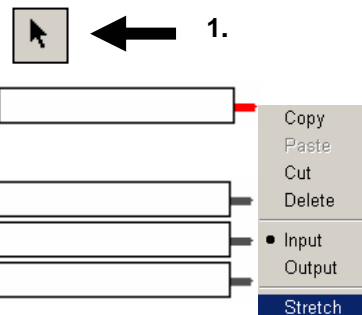
Sélectionner le connecteur sur la surface représentée en rouge

Afficher le menu de contexte avec le bouton droite de la souris

Sélectionner le menu Stretch

Déplacer la souris verticalement pour définir le nombre de connecteurs

Presser le bouton gauche de la souris.



#### 4.6.7 Déplacer un connecteur verticalement

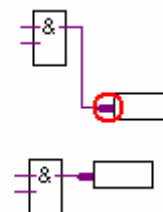
Pour déplacer le connecteur placer la souris dans le cercle rouge

Presser la touche Shift du clavier et la maintenir

Presser le bouton gauche de la souris et le maintenir

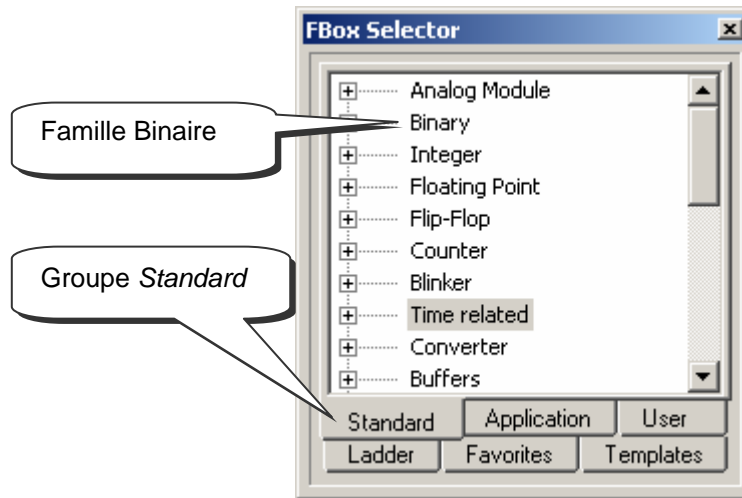
Glisser la souris verticalement sur une surface libre de la page

Libérer le bouton de la souris et du clavier



## 4.7 Edition des Fbox Fupla

### 4.7.1 FBox selector



Add Fbox

La fenêtre Fbox selector rassemble toutes les Fbox graphiques nécessaires à l'élaboration des programmes. Elles sont organisées en groupes d'application.

Le groupe *Standard* rassemble les Fbox de base nécessaire à tous les utilisateurs. Le groupe *Application* rassemble les Fbox spécifiques à certains domaines de métier spécialisés. Le groupe *Ladder* rassemble toutes les Fbox nécessaires à la programmation sous la forme de schémas de contacts.

En suite, chaque groupe est subdivisé en famille dans lesquelles sont réunis les Fbox couvrant un domaine d'application particulier. Avec le groupe *Standard* nous avons par exemples les familles de Fbox:

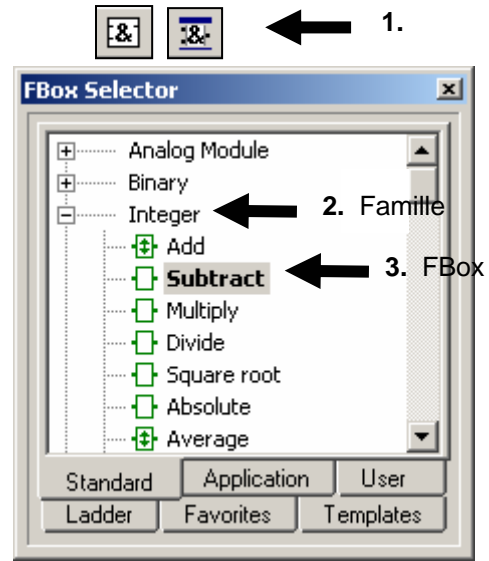
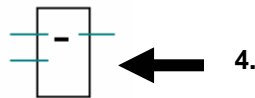
<i>Binaire</i>	FBox pour réaliser des équations logiques
<i>Entier</i>	FBox pour l'arithmétique en nombre entier
<i>Virgule flottante</i>	FBox pour l'arithmétique en nombre flottant
<i>Compteur</i>	FBox pour les tâches de comptage
<i>Temporisateur</i>	Fbox pour les tâches de temporisations
<i>Module Analogue</i>	Fbox pour le contrôle des cartes analogiques
<i>Communication</i>	Fbox pour l'échange de registres, flags, ... sur le réseau S-Bus ou Ethernet
<i>Convertisseurs</i>	Fbox pour convertir un mot binaire en entier, un entier en flottant, ...

...

### 4.7.2 Edition d'une Fbox

Les Fbox nécessaires à décrire le programme sont saisies dans le *Fbox Selector* puis glissée dans l'éditeur de programmes Fupla.

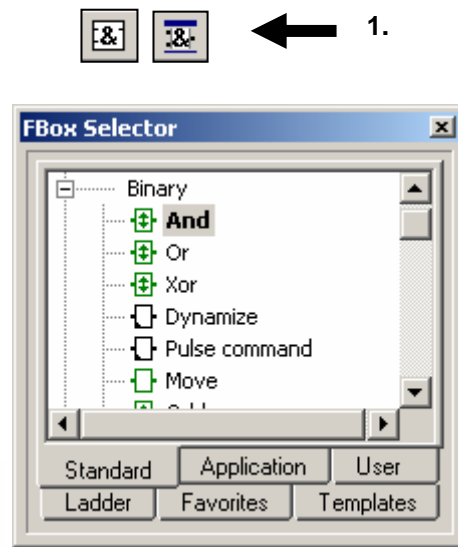
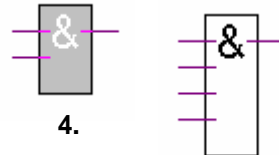
1. Sélectionner le bouton *Add Fbox* ou *Show/Hide FBox Selector*
2. Ouvrir une famille de Fbox
3. Sélectionner une Fbox
4. Placez la Fbox sur la page d'édition, puis presser le bouton gauche de la souris.



### 4.7.3 Edition d'une Fbox extensible (stretchable)

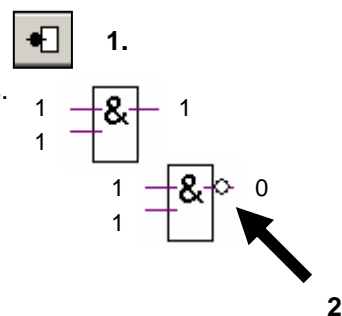
Certaines FBox sont extensibles, cela signifie que le nombre de liaisons peut être défini par un mouvement vertical de la souris.

- 1 Sélectionner le bouton *Add Fbox* ou *Show/Hide FBox Selector*.
- 2 Ouvrir une famille de Fbox.
- 3 Sélectionner une Fbox.
- 4 Placez la Fbox sur la page d'édition, puis presser le bouton gauche de la souris
- 5 Déplacer la souris verticalement pour définir le nombre d'entrées
- 6 Presser le bouton gauche de la souris.



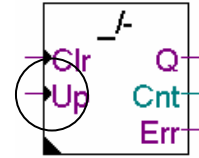
### 4.7.4 Edition d'une inversion logique

- 1 Sélectionner le bouton *Invert Binary Connector*.
- 2 Placer la souris sur le lien d'entrée ou de sortie d'une Fbox logique et presser le bouton gauche de la souris.



### 4.7.5 Dynamisation

Les entrées de certaines Fbox binaires sont dynamisées. Elles prennent uniquement en compte le flanc positif d'un signal logique. Nous les reconnaissons à leur petit triangle noir.



Fbox: *Compteurs*,  
Haut + RAZ

Par exemple un compteur d'impulsion ne peut pas être incrémenté lorsque son entrée *UP* est à un. Si non que se passerait-il si le signal *UP* reste à un pour un certain temps, le compteur pourrait s'incrémenter continuellement tant que le signal *UP* reste à un. C'est pour ce genre d'applications que certaines entrées digitales sont dynamisées. Ainsi seul le flanc positif du signal *UP* incrémente le compteur.

Parfois il est nécessaire d'ajouter une dynamisation sur l'entrée ou la sortie d'une Fbox quelconque. Nous utilisons alors la Fbox : *Binaire, Dynamisation*



### 4.7.6 Commentaires

Des commentaires peuvent être insérés avec le programme:

- 1 Sélectionner le bouton *Place comment*.
- 2 Placez le commentaire sur la page de programme, puis presser le bouton gauche de la souris.
- 3 Ecrire le commentaire
- 4 Presser la touche *ENTER*



Daily Timer

### 4.7.7 Fbox Help

Pour obtenir une description complète d'une Fbox, sélectionner la Fbox dans le *Fbox Selector* puis presser la touche F1.

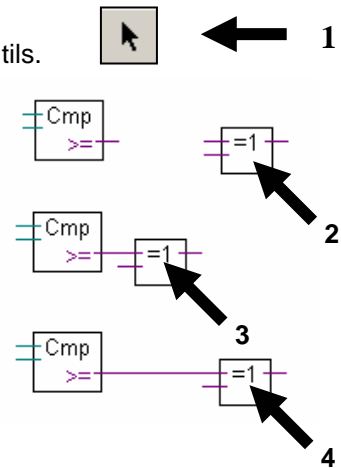
Une autre solution, placer la souris sur une Fbox présente dans le programme et sélectionner le bouton gauche de la souris avec un double clic.

Pour identifier rapidement une Fbox inconnue mais présente dans le programme. Faire apparaître la fenêtre *Fbox Selector*, placer la souris sur la Fbox inconnue, sélectionner le boutons gauche de la souris avec un simple clic. Alors la fenêtre *Fbox Selector* présente la Fbox sélectionnée dans le programme.

## 4.8 Liaisons entre les FBox et connecteurs

### 4.8.1 Liaison par déplacement de la Fbox

- 1 Sélectionner le bouton *Select Mode* de la barre d'outils.
- 2 Placer la souris sur la Fbox, presser le bouton gauche de la souris.
- 3 Sans lâcher le bouton de la souris, faire glisser la Fbox vers la Fbox voisine.
- 4 Les Fbox sont raccordées dès que les deux liaisons se touchent



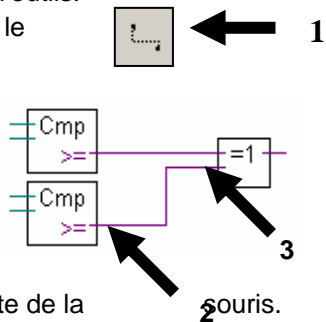
### 4.8.2 Liaison avec routage automatique

- 1 Sélectionner le bouton *Auto Line Mode* de la barre d'outils.
- 2 Placez la souris sur le point de départ et cliquez sur le bouton gauche de la souris.
- 3 Placez la souris sur le point de destination et cliquez sur le bouton gauche de la souris.

Remarque :

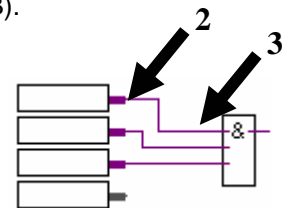
Il est aussi possible de sélectionner des points de passages intermédiaires.

Pour interrompre l'édition du lien, presser le bouton droite de la



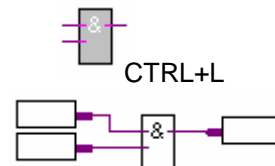
### 4.8.3 Liaison multiple avec routage automatique

- 1 Sélectionner le menu *Mode, Connect Bus* ou (CTRL+B).
- 2 Sélectionner le point de départ avec la souris
- 3 Puis le point de destination




### 4.8.4 Liaison de toutes les entrées, sorties d'une Fbox à des connecteurs

Placer la souris au dessus d'une Fbox, sélectionner le bouton droite de la souris pour sélectionner le menu de contexte *Connections, Connect All*.



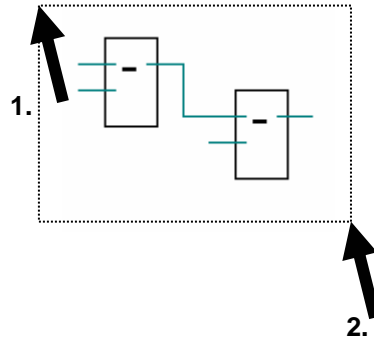


### 4.5.3 Suppression de lignes, Fbox, connecteurs ou symboles

Sélectionner le bouton *Delete Mode* de la barre d'outils puis sélectionnez les liens, Fbox, connecteurs ou symboles à effacer. 

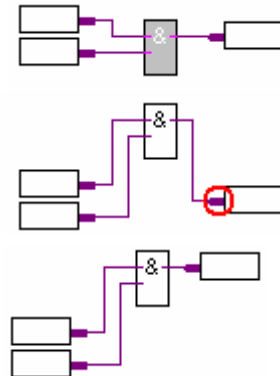
Une autre solution, plus rapide consiste à marquer un espace et de l'effacer.

- 1 Presser le bouton de la souris
- 2 Sans lâcher le bouton, glisser la souris
- 3 Lâcher le bouton de la souris
- 4 Sélectionner le menu *Edit, Delete*



### 4.8.5 Déplacer une Fbox, un connecteur verticalement sans défaire les liens

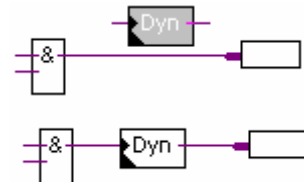
Placer la souris au dessus de la Fbox  
 Presser la touche Shift du clavier et la maintenir  
 Presser le bouton gauche de la souris et le maintenir  
 Glisser la souris verticalement sur une surface libre de la page  
 Libérer le bouton de la souris et du clavier



Pour déplacer le connecteur placer la souris dans le cercle rouge et répéter la manipulation.

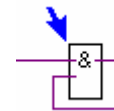
### 4.8.6 Insérer une Fbox sans défaire le lien.

Sélectionner la Fbox à insérer dans le *Fbox Selector*.  
 La placer au dessus du lien.



### 4.8.4 Règles à respecter

Il n'est pas permis de réaliser des boucles. Il en résulte un message d'erreur : Page 1: Error 55: Loop back detected



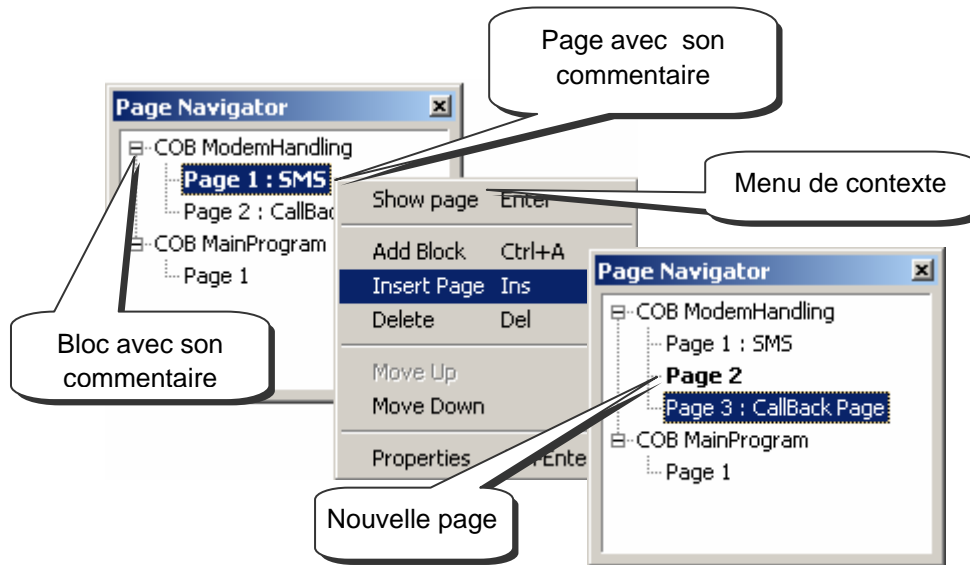
Il n'est pas permis de lier directement un connecteur d'entrée sur un connecteur de sortie. Il faut utiliser une Fbox *Binaire, Transfert direct* ou *Entier, Transfert direct*



Les symboles pour les connecteurs de sortie doivent toujours être liés à la sortie d'une Fbox. Si non, il en résulte un message d'erreur: Page 1: Error 53: Incomplet network



## 4.9 Edition de pages Fupla



Show/Hide  
Page Navigator

La fenêtre *Page Navigator* organise les programmes en blocs et en pages. Chaque fichier Fupla peut supporter jusqu'à 200 pages rassemblées dans des blocs COB, PB, FB, SB, .... Par défaut, les pages sont placées dans un bloc de type COB. Pour de plus amples renseignements concernant les blocs et leur usage, veuillez vous référer au chapitre 5 de ce document.

### 4.9.1 Ajouter une page



Insert Page

Ouvrir la fenêtre *Page Navigator*, marquer la page de référence et sélectionner le menu de contexte *Insert Page*.

Il est aussi possible d'ajouter une page après la page courante avec le bouton *Insert* ou le menu *Page Insert After (Page Insert Before)*

### 4.9.2 Supprimer une page

Ouvrir la fenêtre *Page Navigator*, marquer la page à supprimer et sélectionner le menu de contexte *Delete*

### 4.9.3 Navigation entre pages



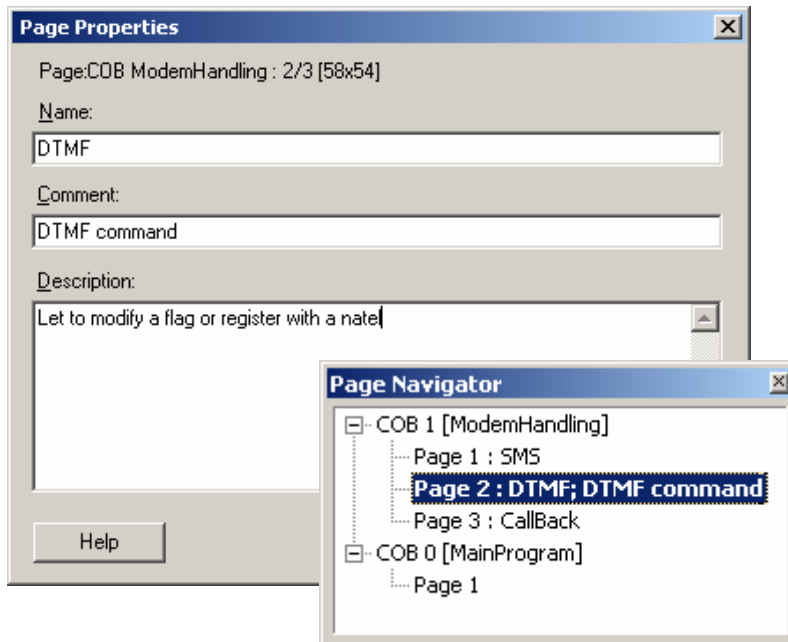
Goto Next  
Page

Ouvrir la fenêtre *Page Navigator*, placer la souris sur la page et sélectionner le bouton gauche de la souris avec un double clic.

Il est aussi possible de naviguer avec les boutons *Goto Previous Page* et *Goto Next Page* qui permettent de se déplacer page par page dans le bloc Fupla. Si l'un des boutons est gris, cela signifie que vous êtes sur la première ou la dernière page du bloc.

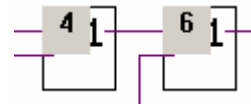
#### 4.9.4 Documentation de pages

Nous vous conseillons vivement de documenter chacune de vos pages Fupla. C'est très pratique pour naviguer dans les pages du programme. Le nom et le commentaire des pages sont ainsi visibles dans la fenêtre *Page navigator*. La description permet de laisser quelques informations utiles ou une description pour faciliter l'entretien du programme.



#### 4.9.5 Traitement du programme par le PCD

Le PCD traite le programme du coin supérieur gauche de la première page au coin inférieur droit de la dernière page. Pour connaître avec plus de précision l'ordre dans lequel les Fbox sont traitées par le PCD, sélectionner le menu *Page, Fbox priorities*

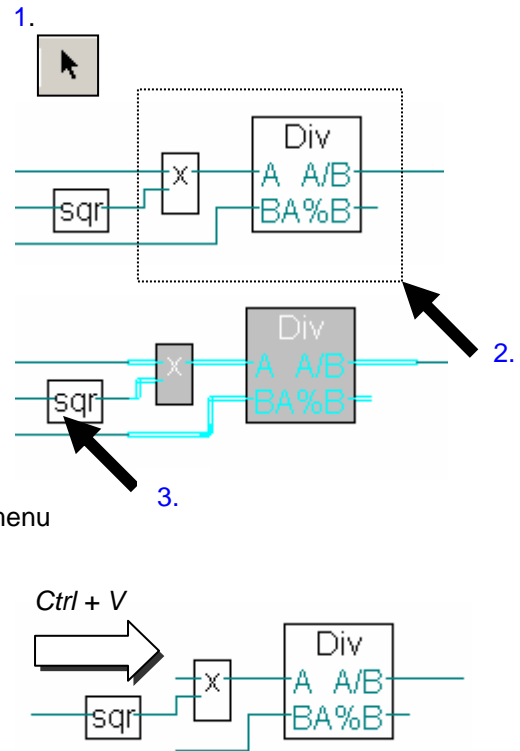


## 4.10 Copier, coller

Parfois certaines parties de programme sont répétitives, il n'est pas nécessaire de les éditer intégralement, c'est beaucoup plus rapide de les dupliquer avec un copié, collé puis de les adapter selon les besoins.

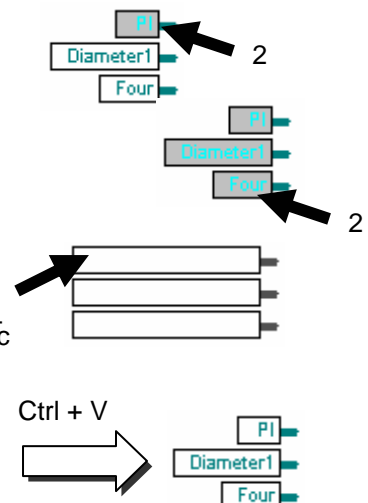
### 4.10.1 Copier, coller une partie de programme

1. Sélectionner le bouton *Select Mode*.
2. Marquer la surface à copier:
  - ♣ Presser le bouton gauche de la souris.
  - ♣ Sans lâcher le bouton, glisser la souris.
  - ♣ Lâcher le bouton gauche de la souris.
3. Ajouter une Fbox ou un lien à la sélection:
  - ♣ Presser la touche *Ctrl*
  - ♣ Sans lâcher la touche *Ctrl*, sélectionner les liens, connecteurs et Fbox à ajouter
4. Copier la sélection avec le menu *Edit Copy* ou avec les touches *Ctrl + C*
5. Coller une copie de la sélection avec le menu *Edit Past* ou les touches *Ctrl + V*
6. Positionner la copie dans la page Fupla :
  - ♣ Pacer la souris au centre de la copie.
  - ♣ Presser le bouton gauche de la souris.
  - ♣ Sans lâcher le bouton, glisser la souris.



### 4.10.2 Copier, coller des symboles

- 1 Sélectionner le bouton *Select Mode*.
- 2 Marquer une liste de symboles:
  - ♣ Placer la souris sur le premier symbole
  - ♣ Sélectionner le bouton gauche de la souris.
  - ♣ Placer la souris sur le dernier symbole.
  - ♣ Presser la touche *Shift* \*)
  - ♣ Sans lâcher la touche *Shift*, presser le bouton gauche de la souris.
- 3 Copier la sélection avec le menu *Edit Copy* ou avec les touches *Ctrl + C*
- 4 Placer la souris sur un élément libre de la marge
- 5 Coller une copie de la sélection avec le menu *Edit Past* ou les touches *Ctrl + V*.



\*) La touche *Ctrl* permet de sélectionner des symboles non consécutifs

## 4.11 Exporter, importer des pages

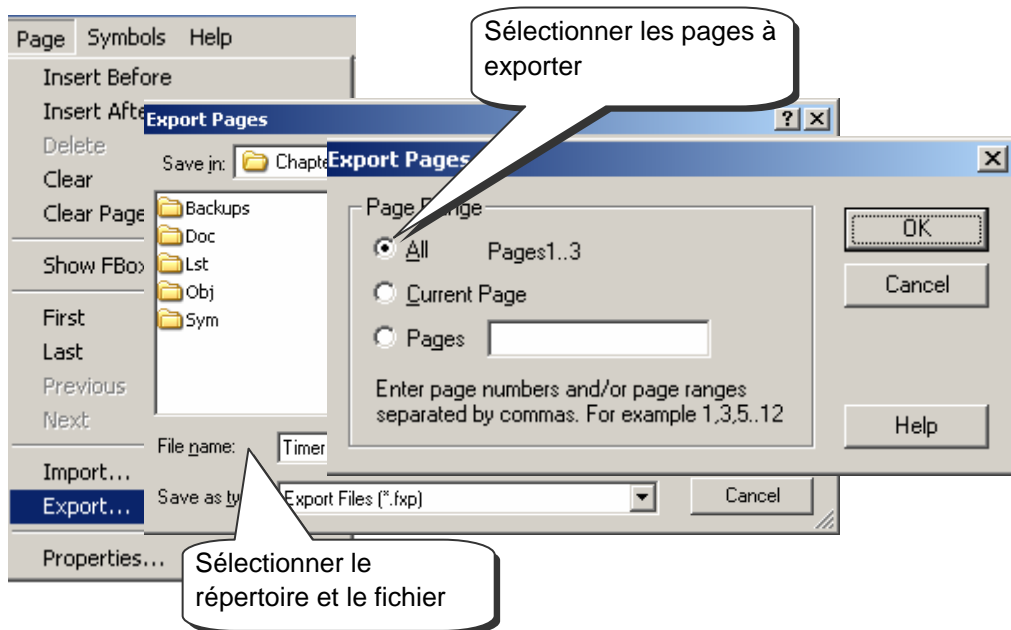
Exporter et importer des pages est un outil complémentaire à la fonction copier, coller mais beaucoup plus performant. Il offre les avantages suivants :

- Copier, coller de plusieurs pages
- Modifications des symboles et adresses importées
- Possibilité de préparer une bibliothèque de composants

### 4.11.1 Exportation de pages

L'exportation de pages permet de sélectionner une ou plusieurs pages du programme Fupla avec les Fbox, les liens, les commentaires, les symboles pour les sauvegarder dans un fichier. Cela permet de rassembler des parties de programmes couramment employées dans un ou plusieurs fichiers (\*.fxp) et d'élaborer une bibliothèque de composants pour construire rapidement vos applications futures.

Par exemple, vous utilisez souvent des horloges journalières, hebdomadaires et mensuelles pour enclencher et déclencher une sortie digitale à un moment définis de la journée, de la semaine ou du mois. Alors vous créer un programme de plusieurs pages, chaque page correspond à une des horloges couramment utilisées et vous exportez le tout dans fichier *Timer.fxp*.

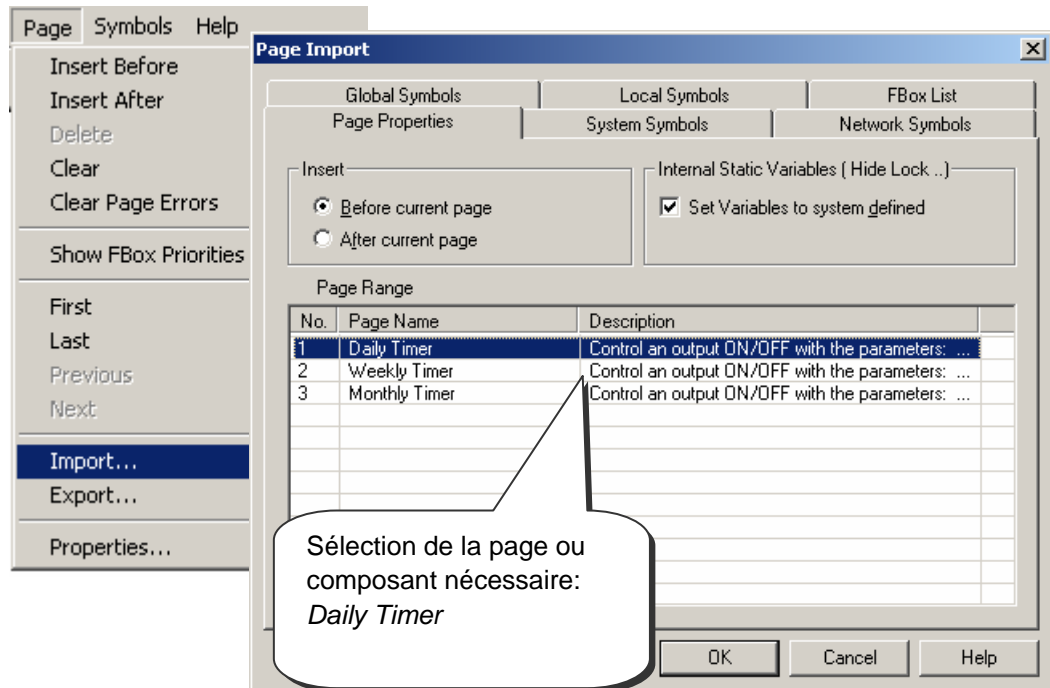


### 4.11.2 Importation de pages

L'importation de pages supporte la sélection d'une ou plusieurs pages d'un fichier (\*.fxp) ainsi que l'édition des symboles importés vers le programme Fupla.

Le répertoire *Page Range* permet de sélectionner la ou les pages à importer.

En ce qui concerne notre exemple, chaque fois que vous avez besoin d'une horloge, vous importez le fichier *Timer.fxp* et vous sélectionnez la ou les pages qui correspondent au composant nécessaire à votre application.



Les répertoires *Global Symbols* et *Local Symbols* représentent la liste des symboles correspondant aux pages sélectionnées. Il est même possible de modifier les noms et les adresses de chaque symbole:

Marquer les symboles pour les placer dans un groupe est le moyen le plus rapide de modifier les noms de tous les symboles du programme importé.

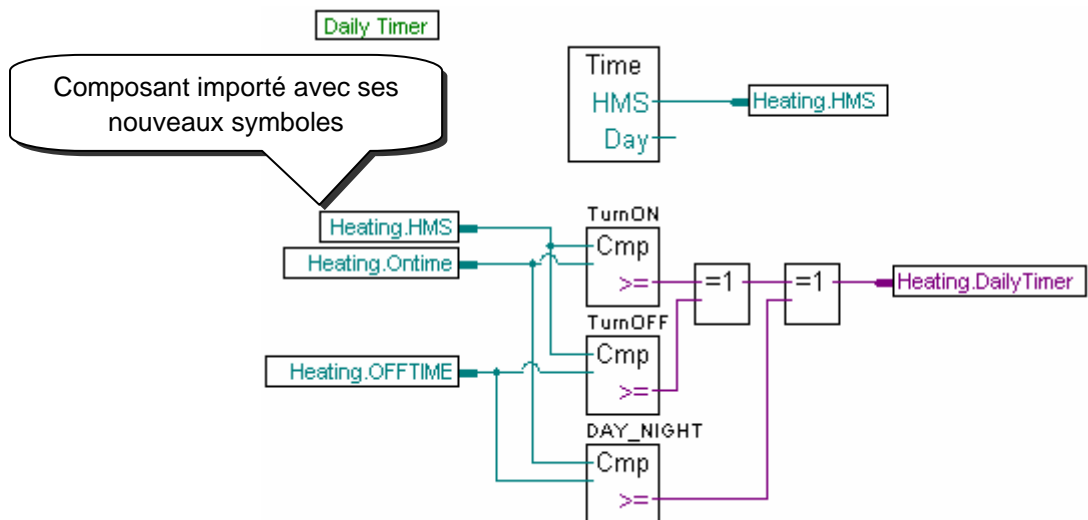
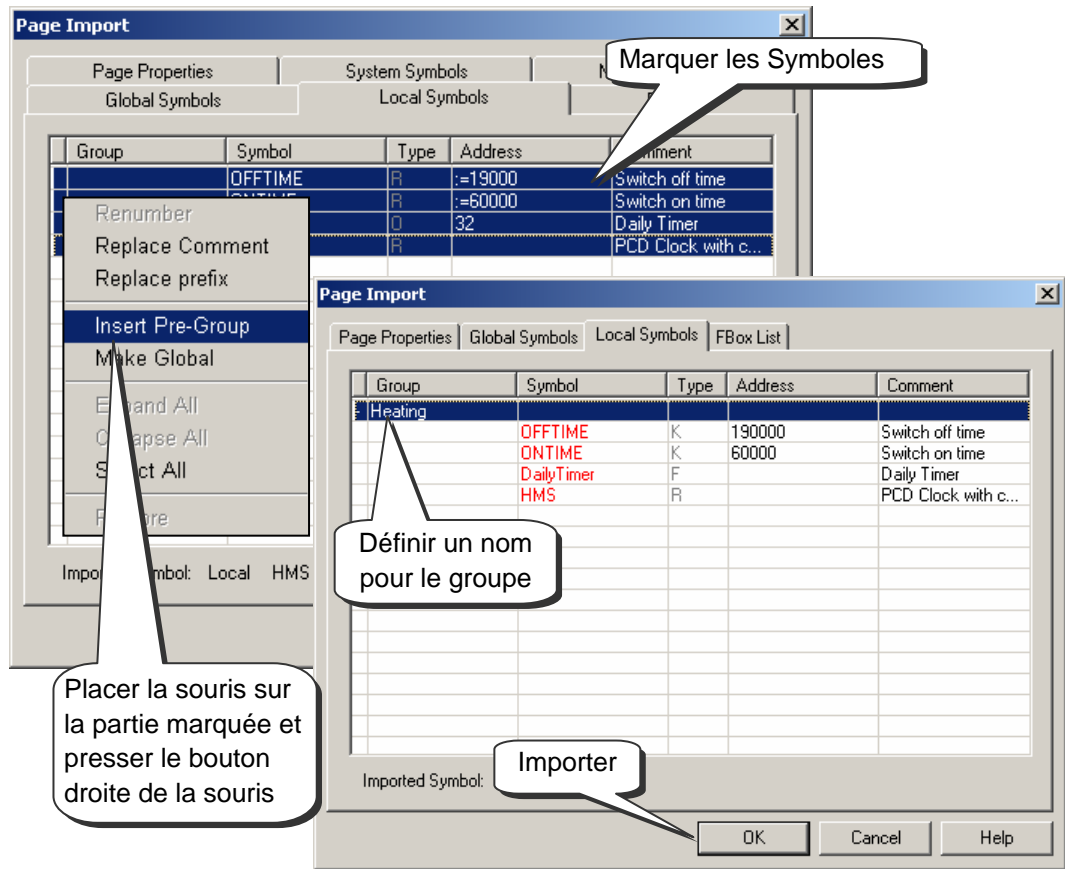
Le menu de contexte *Insert Pre-group* permet de placer les symboles sélectionnés dans un groupe avec un nom de votre choix.

Marquer tous les symboles pour les renuméroter est aussi un moyen rapide de modifier les adresses des symboles des pages importées.

Le menu de contexte *Renumber* permet de modifier les adresses des symboles sélectionnés avec un offset des adresses ou à partir d'une adresse de base.

Le répertoire *Fbox List* représente la liste des symboles liés à un groupe de Fbox. Ces symboles donnent un accès aux informations internes des Fbox tel que les adresses des paramètres des fenêtres d'ajustage. Ce répertoire permet de modifier les noms attachés aux Fbox des pages importées.

Avec notre petit exemple nous proposons d'importer les symboles du composant horloge journalière dans un groupe *Heating*. Pour chaque nouvelle horloge journalière, définir un nom de groupe différent. Ainsi les symboles des composants employés plusieurs fois appartiennent à des groupes différents et ne causent aucun problème d'adressage lors de la construction du programme.



## 4.12 Edition d'un premier programme Fupla

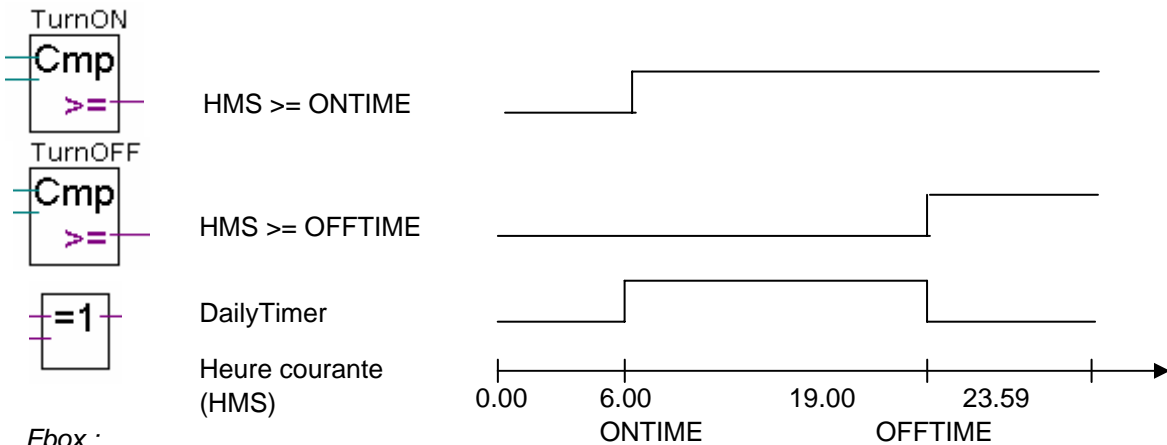
### 4.12.1 Objectif

Maintenant que l'environnement de travail est connu, il est possible de passer à l'élaboration d'un programme plus important que les équations logiques présentées jusque là. Nous proposons de réaliser une horloge journalière pour gérer une sortie digitale O 32 qui s'enclenche à 6.00 h du matin et qui se déclenche à 19.00 h du soir. Remarquons que cette fonction est disponible avec la librairie CVC et que nous allons tout de même la réaliser nous même à partir des Fbox standards.

### 4.12.2 Méthode

Avant de programmer, il est nécessaire de trouver une méthode qui se comporte selon notre cahier des charges et que nous pouvons réaliser avec les Fbox les plus élémentaires possibles.

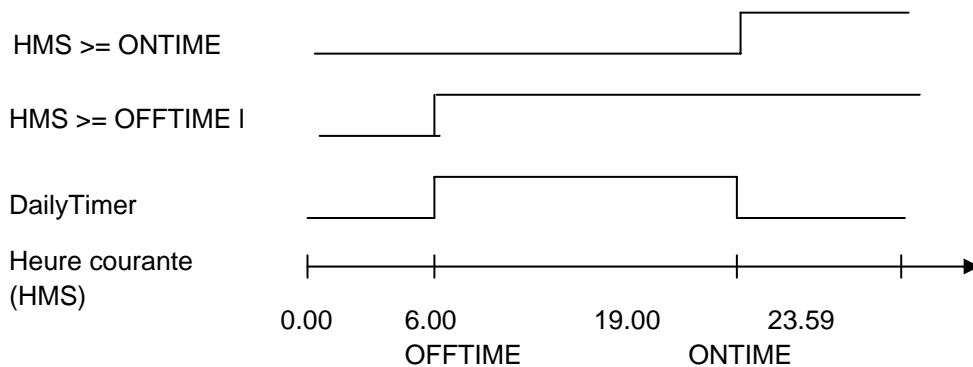
Pour l'exemple de cette horloge, nous proposons de faire deux comparaisons. La première détermine si l'heure courante *HMS*, l'heure que nous lisons à notre montre ou celle du PCD, est plus grande ou égale à l'heure d'enclenchement *ONTIME* et la seconde détermine si l'heure courante est plus petite ou égale à l'heure de déclenchement *OFFTIME*. Si les deux comparaisons sont vérifiées par une fonction logique OU exclusif, alors la sortie digitale *DailyTimer* de l'horloge doit être enclenchée.



Fbox :

- Entiers, Plus grand ou égal
- Binaire OU exclusif

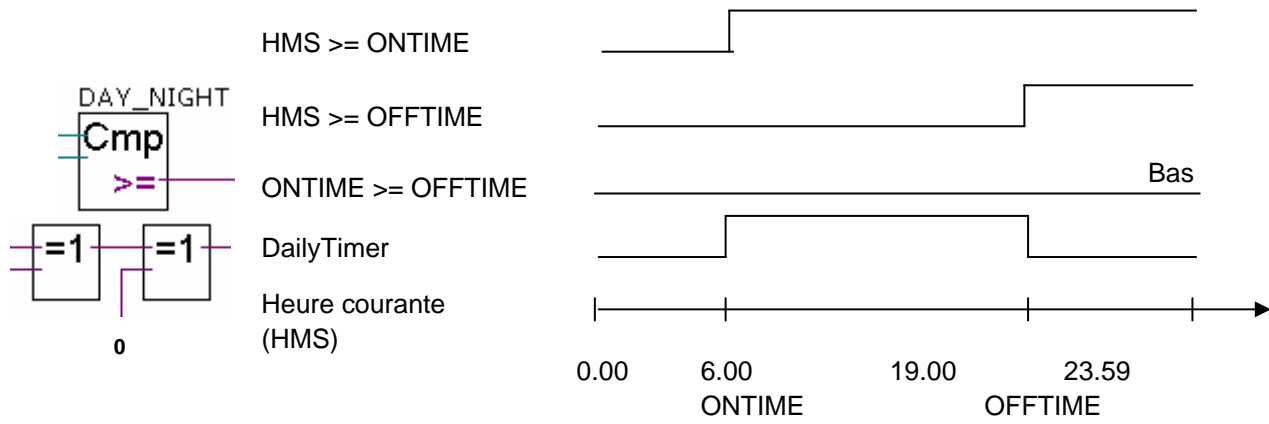
Cet algorithme propose une solution qui peut présenter quelques lacunes. Que se passe-t-il si on croise les consignes de l'heure d'enclenchement est de déclenchement? Le croquis ci-dessous démontre que la sortie du PCD sera dans l'état inverse de celui souhaité.



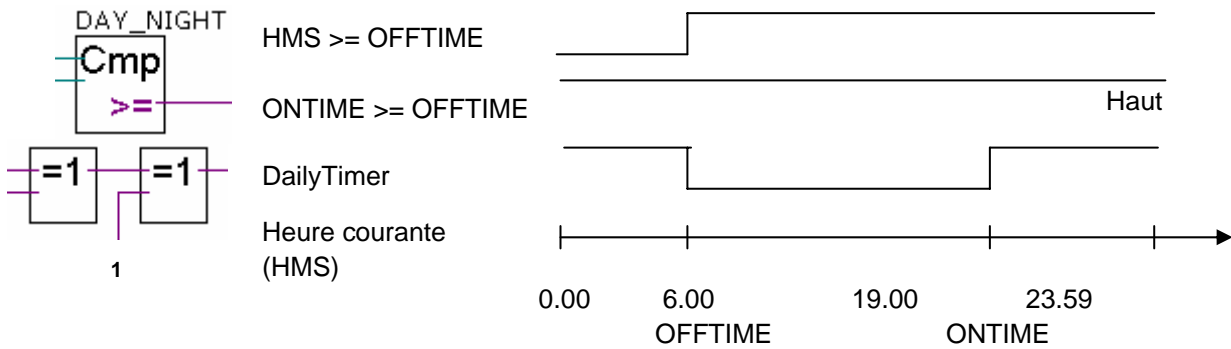


Il est donc nécessaire de compléter notre algorithme en ajoutant une troisième comparaison pour déterminer si l'heure d'enclenchement est plus grand ou égale à l'heure de déclenchement. La solution définitive est alors la suivante.

Sorties active le jour:



Sorties active la nuit:



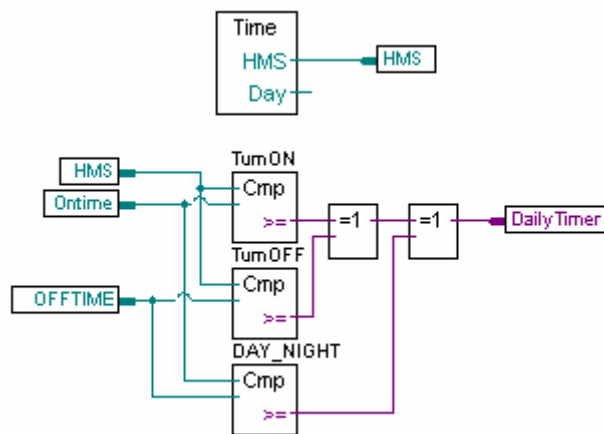
### 4.12.3 Programmation

Maintenant, il est temps de passer à la programmation. Au début de ce chapitre, nous avons déjà préparé un projet avec un fichier intitulé *DailyTimer.fup*. C'est dans ce fichier que vous allez écrire cet exemple de programme.

Group/Symbol	Type	Address/Value	Comment
HMS	R		PCD Clock with current time
DailyTimer	Output	32	Daily Timer
ONTIME	R	:=60000	Switch on time
OFFTIME	R	:=19000	Switch off time
COB_3A87C0D7	COB		

Commencez par préparer la liste des symboles. Remarquez que L'heure courante du PCD est sauvegardée dans un registre HMS dynamique, l'adresse du registre n'est pas définie, c'est le PG5 qui en attribuera une automatiquement lors de la construction du programme. Même remarque pour les heures d'enclenchement et déclenchement *ONTIME*, *OFFTIME* sauf que « :=60000 » n'est pas l'adresse d'un registre mais la valeur avec lequel il sera initialisé lors du chargement du programme dans le PCD ( :=60000 signifie 6 heures 00 minutes 00 secondes.)

Attention le démarrage à froid du PCD ne va pas réinitialiser ces registres. Seul le chargement du programme les initialiserons à nouveau !



Vous trouverez toutes les FBox nécessaires dans le groupe *Standard* de la fenêtre *Fbox Selector* :

- Temporisateurs, Lecture horloge
- Entiers, Plus grand ou égal
- Binaire OU exclusif

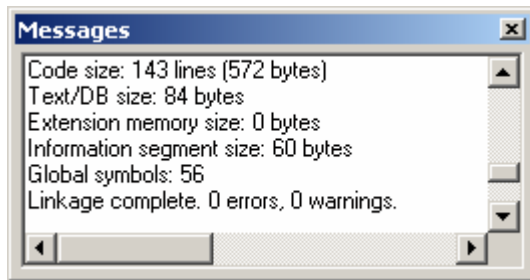
### 4.13 Construction du programme (Build)



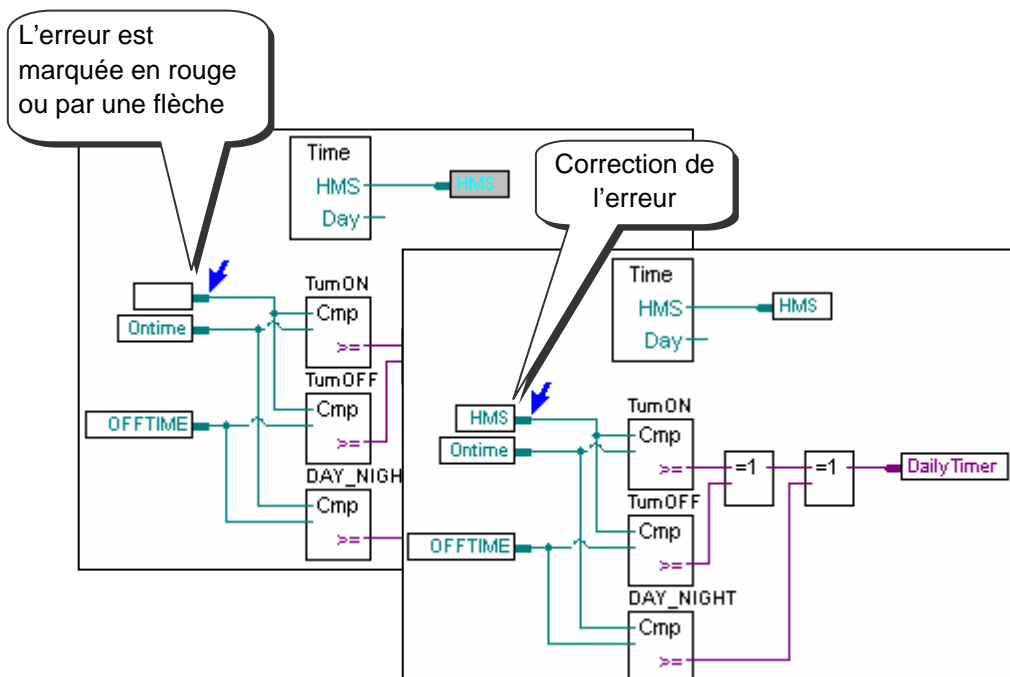
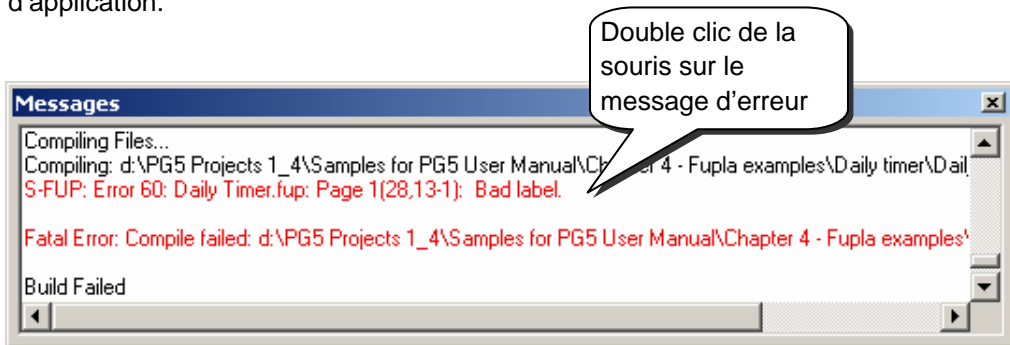
Build All

Le programme utilisateur est complètement décrit, mais pas encore utilisable par le PCD. Il ne comprend pas le langage graphique de l'exemple décrit précédemment. Il est donc nécessaire de le traduire. C'est ce que fait l'outil de programmation lorsque l'utilisateur actionne le menu *CPU Build* ou le bouton *Build* du gestionnaire de projet ou de l'éditeur graphique.

La fenêtre des *Messages* nous renseigne sur le déroulement de la construction du programme. Nous remarquerons plusieurs étapes de construction: la compilations, l'assemblage et le linkage. Si le programme est correctement édité, la construction se termine par un message Build successful. Total errors 0 Total warnings: 0



Les éventuelles erreurs sont signalées par un message rouge. Un double clic de la souris sur ces messages permet de localiser l'erreur dans le programme d'application.



#### 4.14 Charger le programme dans le PCD



*Download Program*

Le programme d'application est prêt. Il reste à le transférer de l'ordinateur vers le PCD avec le menu *Online, Download Program* ou le bouton *Download Program* de la fenêtre SAIA Project Manager.

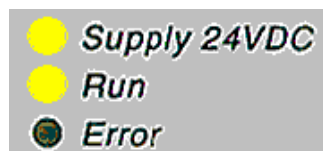
En cas de problème de communication, veuillez à nouveau contrôler vos configurations *Settings Online* et *Settings Hardware* ainsi que votre câble de communication entre l'ordinateur et le PCD. (PCD8.K111, USB)

#### 4.15 Débugger un programme (Debug)

Les programmes ne sont pas toujours parfaits dès la première version. Il est utile de les tester soigneusement. Le test d'un programme est supporté avec le même éditeur que celui employé lors de l'édition:

##### 4.15.1 Go On/Offline, Run, Stop, Step-by-step

1. Se mettre en ligne avec le bouton *Go On/Offline*
2. Forcer l'automate en mode run avec le bouton *Run*



Parallèlement, observer la lampe *RUN* placée sur la face avant du PCD. A la sélection du bouton *Run*, la lampe *RUN* est allumée, le PCD exécute le programme utilisateur.

3. A la sélection du bouton *Stop*, la lampe *RUN* est éteinte, le PCD stop l'exécution du programme utilisateur.
4. Le PCD exécute une Fbox à chaque sélection du bouton *Step by step* ou de la touche *F11*



Observez le pointeur *Stop*, il indique l'évolution pas-à-pas du programme

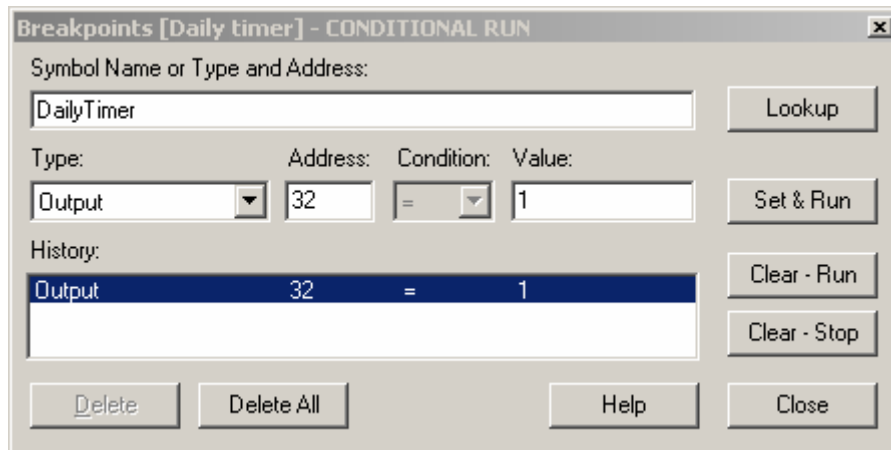
### 4.15.2 Point d'arrêt

Permet de stopper le programme sur un événement lié à une Fbox du programme ou un symbole:

Etat bas ou haut d'une entrée, sortie, indicateur, indicateur de statut  
Valeur présente dans un registre ou compteur

Point d'arrêt sur un symbole

La condition d'arrêt peut être déterminée à l'aide du menu *Online Breakpoints*.



A l'aide de la fenêtre ci-dessus, définir le type et l'adresse/numéro du symbole ou simplement glisser un symbole depuis l'éditeur de symbole dans le champs *Symbol Name*, puis déterminer la condition et l'état/valeur d'arrêt.

La sélection du bouton *Set&Run* force le PCD en mode Run conditionnel. La LED *Run* du PCD clignotte et le bouton *Run* du PCD alterne les couleurs vert et rouge.

Le PCD se met automatiquement en mode stop lorsque la condition d'arrêt est réalisée. Par exemple lorsqu'une instruction modifie la valeur de la sortie 32 est à l'état haut. La dernière Fbox traitée par le PCD est marquée en rouge. Il est possible de continuer le traitement du programme en mode pas-à-pas ou avec une autre condition d'arrêt.

Si nécessaire il est possible d'interrompre le mode run conditionnel :

Le bouton *Clear-Run*, force le PCD en mode RUN. La LED *Run* du PCD est allumée et le bouton *Run* du PCD de couleur verte.

Le bouton *Clear-Stop*, force le PCD en mode Stop. La LED *Run* du PCD est éteinte et le bouton *Run* du PCD de couleur rouge.

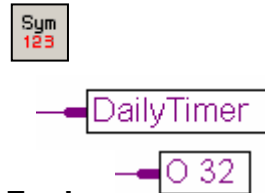
Si plusieurs point d'arrêts conditionnels ont été saisis, ils sont tous conservés dans le champs *History*. Il est alors possible d'en sélectionner un à l'aide de la souris et de l'activer avec le bouton *Set&Run*.

Point d'arrêt sur une Fbox du programme

La sélection d'une Fbox du programme puis du menu *Online, Run to, Fbox* permet de stopper le programme sur la Fbox sélectionnée pour en suite continuer en mode pas-à-pas.

### 4.15.3 Visualiser les symboles ou leurs adresses

Le bouton *Show Operand as symbol or value* permet d'afficher les informations des connecteurs avec leurs symboles ou leurs adresses. Si malgré tout le symbole n'est pas remplacé par l'adresse correspondante, cela signifie que ce symbole est dynamique.



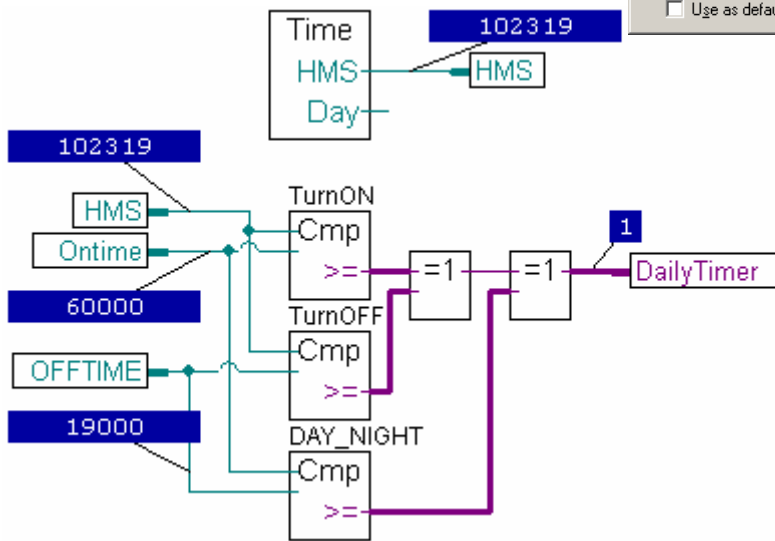
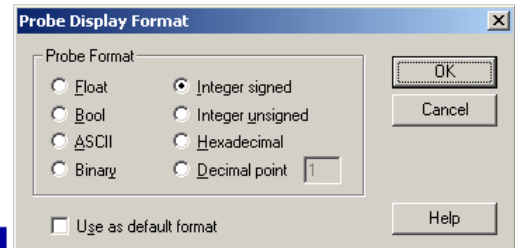
### 4.15.4 Visualiser l'état des symboles avec Fupla



Place Probe

Si l'éditeur est en ligne et le PCD est en mode *RUN*, alors il est possible de représenter l'état de chacun des symboles employés par le programme: les états logiques des informations binaires sont représentés par des liens larges ou fins. (large = 1 et fin = 0). Toutes les autres informations peuvent aussi être représentées par une sonde que l'on obtient en cliquant sur le lien avec la souris.

Un double clic sur une sonde ouvre la fenêtre *Probe Display Format* qui permet de choisir le format des valeurs à consulter: entier, hexadécimal, binaire, flottant, booléen ou ASCII.

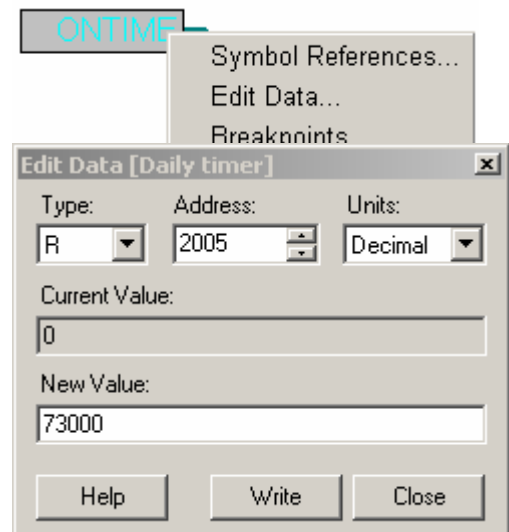


### 4.15.5 Editer les symboles en ligne

Pour vérifier le comportement du programme sous certaines conditions d'utilisation, il est parfois utile de modifier les états/valeurs des symboles présents dans les connecteurs d'entrées.

Sélectionner un connecteur d'entrée à l'aide de la souris et presser le bouton droite pour afficher le menu de contexte.

Le menu de contexte *Edit Data* permet de modifier l'état/valeur du symbole présent dans le connecteur.



### 4.15.6 Visualiser l'état des symboles avec le *Watch window*

Un autre moyen utile pour tester et visualiser l'état des symboles de notre exemple est de faire usage de la fenêtre *Watch window*. Sélectionnez le bouton *Watch window*. Puis glisser les symboles de l'éditeur de symboles vers la fenêtre *Watch window*

1. Placer la souris au centre de l'icône du symbole, presser le bouton gauche

2. Maintenir le bouton de la souris et glisser le symbole dans la fenêtre *Watch window*

3. Symboles avec leurs commentaires et leurs états/valeurs.

4. Start/Stop Monitoring

Symbol	Address	Value	Modify Value	Symbol Comment
HMS	R 2113	103034		PCD Clock with current time
DailyTimer	O 32	1		Daily Timer
Ontime	R 2115	60000		Switch on time
OFFTIME	R 2114	19000		Switch off time

Pour tester le bon fonctionnement de notre exemple d'horloge journalière. Nous allons modifier les consignes d'enclenchement *ONTIME* et de déclenchement *OFFTIME* et observer l'état de la sortie *DailyTimer*. Pour éditer une consigne procéder comme suit:

1. Start/Stop Monitoring

2. Placer la souris sur la valeur à éditer, presser 2 fois le bouton gauche de la souris et éditer la nouvelle valeur.

3. Download Values

Symbol	Address	Value	Modify Value	Symbol Comment
HMS	R 2113	104852		PCD Clock with current time
DailyTimer	O 32	1		Daily Timer
Ontime	R 2115	60000	43000	Switch on time
OFFTIME	R 2114	19000		Switch off time

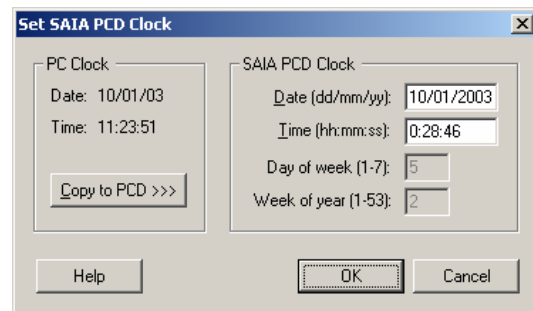
### 4.15.7 Mise à l'heure de l'horloge PCD

A la mise en service, l'horloge interne de l'automate n'est pas toujours à l'heure. Pour l'ajuster veuillez procéder comme suit:

1. Sélectionnez le bouton *Online configurator* de la fenêtre *SAIA Project Manager*. Puis le bouton *Clock*



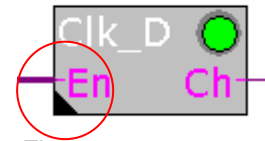
2. Copier l'heure du PC dans l'automate avec le bouton *Copy to PCD >>>* ou ajuster l'horloge dans les champs *SAIA PCD Clock*





### 4.16 Fenêtres d'ajustage

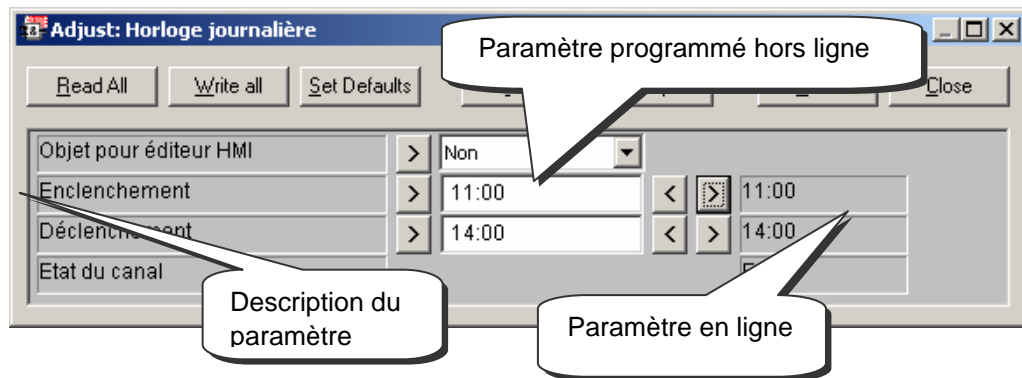
Certaines Fbox, marquées par un angle noir, supportent une fenêtre de dialogue avec des paramètres d'ajustage. Les paramètres d'ajustage sont beaucoup utilisés avec la librairie CVC, mais aussi avec les autres librairies de Fbox.



Fbox:  
CVC Horloges,  
Horloge journalière

Les paramètres d'ajustage servent à définir des variables comme pour les liens d'entrées. Mais avec des avantages supplémentaires: description de chaque paramètre, correction en ligne, simplicité d'utilisation, ...

Pour afficher la fenêtre d'ajustage, placer la souris sur une Fbox marquée par un angle noir et cliquer deux fois sur le bouton gauche de la souris.



La fenêtre d'ajustage dispose de trois colonnes :

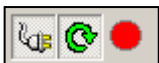
Une colonne avec une description du paramètre d'ajustage, permet d'orienter l'utilisateur sur l'usage d'un paramètre. Lorsque la description se termine par trois points, cela signifie qu'une information supplémentaire est disponible. Placer la souris sur le texte et cliquer deux fois sur le bouton gauche de la souris.



Mode hors ligne

Une colonne avec la valeur du paramètre hors ligne. C'est la valeur définie lors de la programmation de l'application. Elle est sauvegardée dans le fichier Fupla. Le paramètre peut être généralement défini sous la forme d'un entier, d'une sélection à choix multiples ou parfois par un bouton.

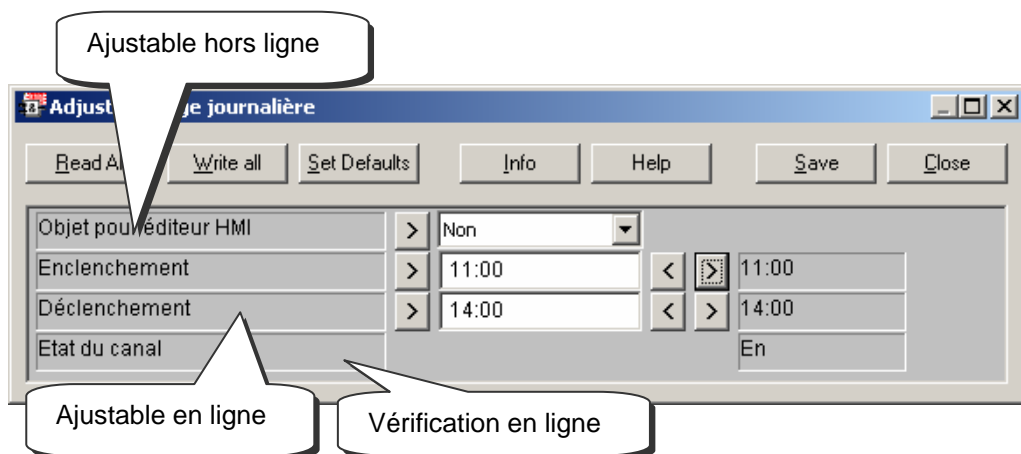
Avec certains paramètres, cette valeur peut aussi être éditée lorsque le PCD est en ligne.



Mode en ligne

Une colonne avec la valeur en ligne, cette colonne représente l'information sauvegardée dans la mémoire du PCD (registre ou flag). C'est l'information utilisée par le programme PCD lorsqu'il est en mode *Run*.

### 4.16.1 Types de paramètres d'ajustage



Les paramètres d'ajustage peuvent être classés en trois grands groupes.

Les paramètres ajustables hors ligne. (Off line)

Chaque modification d'un paramètre hors ligne nécessite un *Build All* et un *Download Program* pour être pris en compte par le programme du PCD.

Les paramètres ajustables en ligne. (On line)

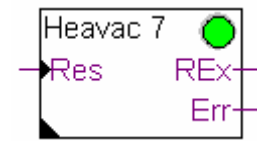
La modification d'un paramètre en ligne ne nécessite aucun *Build All* et aucun *Download Program* pour être pris en compte par le programme du PCD. Le paramètre est directement ajusté dans la mémoire du PCD

Les paramètres de vérification en ligne. (On line)

Ces paramètres ne sont pas prévus pour être modifiés, ils sont juste représentés pour information et vérification du bon fonctionnement du programme ou de la Fbox.

### 4.16.2 Initialisation des Fbox CVC

Lors que vous utilisez certaines bibliothèques de Fbox tel que les applications CVC, il est nécessaire de toujours placer une Fbox d'initialisation au début du fichier Fupla. Elle permet de gérer certaines tâches communes à la bibliothèque, tel que l'initialisation des Fbox après le chargement du programme ou après un démarrage à froid du PCD (mise sous tension du PCD)



Fbox: CVC Init,  
Initialisation CVC 7

Après un chargement de programme et un démarrage à froid du PCD, l'entrée *Res* de cette Fbox ainsi que les paramètres d'ajustage représentés ci-dessous ont une importante influence sur l'initialisation des paramètres d'ajustage de toutes les autres Fbox CVC du programme.

Les paramètres prennent leurs valeurs par défaut		
Reset manuel ...	<input type="button" value="Pre- / Reset"/>	<input type="text" value=""/>
Reset automatique ...	<input type="button" value="&gt;"/> Activé	<input type="text" value=""/>
Raison du reset automatique		<input type="text" value=""/>
Evaluer l'entrée Reset	<input type="button" value="&gt;"/> Au démarrage	

Chargement du programme et le paramètre *Reset automatique*:

Avec l'option *Activé*, les paramètres d'ajustage de toutes les Fbox CVC seront initialisés avec les valeurs définies par le programme.

Avec l'option *Désactivé*, tous les paramètres déjà présents dans le PCD sont conservés

L'entrée *Res* et le paramètre *Evaluer l'entrée Reset*:

Si l'entrée *reset* est à l'état haut, les paramètres d'ajustage de toutes les Fbox CVC seront initialisés avec les valeurs définies à la programmation.

Selon l'option du paramètre *Evaluer l'entrée Reset*, l'entrée *Res* ne sera jamais pris en compte, seulement au démarrage à froid du PCD ou en runtime (toujours)

LED vert/rouge

Certaines Fbox disposent d'une LED qui peut prendre trois couleurs distinctes.

Gris lors que l'automate est hors ligne, vert ou rouge lorsque l'automate est en ligne. Vert signifie que tout fonctionne parfaitement bien, rouge signifie une erreur généralement causée par les informations en entrées de la Fbox ou les paramètres d'ajustage sélectionnés qui ne conviennent pas. Pour de plus amples précisions, veuillez consulter les aides relatives aux Fbox en erreur.

Remarque:

Dans la bibliothèque CVC vous trouverez différentes versions de Fbox d'initialisation (Initialisation CVC 4, ...7). La version 7 est la plus récente. Nous vous recommandons d'utiliser la fonction 7 pour les nouvelles applications.

### 4.16.3 Fbox CVC avec paramètres d'ajustage

La Fbox *CIK\_D* permet de créer une horloge journalière identique à celle réalisée précédemment dans ce chapitre, mais avec une seule Fbox disponible dans la librairie CVC.

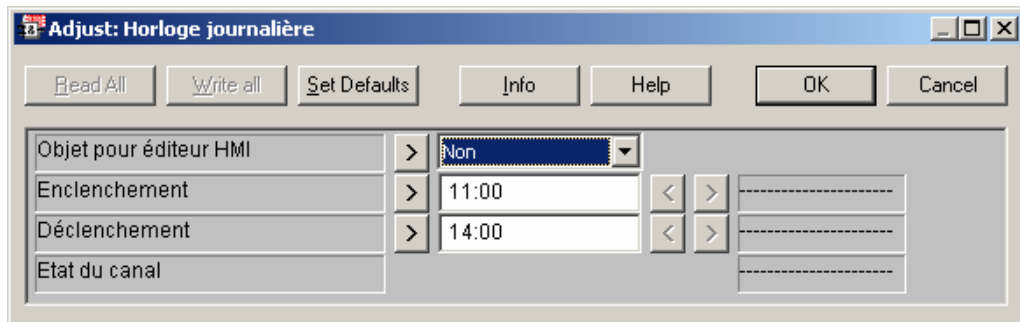


Fbox: CVC Horloges, Horloge journalière

La sortie de la Fbox peut être enclenchée ou déclenchée selon les heures définies dans la fenêtre d'ajustage.

Le paramètre *Objet pour éditeur HMI* n'est utile que pour l'usage de terminaux HMI. Si cette option n'est pas utilisée, garder le paramètre standard proposé.

L'entrée *En* permet de désactiver le fonctionnement de l'horloge. Si *En* est à l'état bas, la sortie *Ch* reste toujours désactivée.



### 4.16.4 Mini application CVC

Pour tester le fonctionnement des paramètres d'ajustages, nous proposons de réaliser à nouveau le programme d'horloge journalière présenté au début de ce chapitre. Mais cette fois nous allons le réaliser à l'aide de la librairie CVC.

Seul les deux Fbox présentées précédemment sont nécessaires. Réalisez le programme comme représenté ci-dessous puis réalisez le *Build All*, le *Download Program* et *Go Online*.



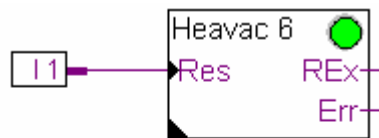
Rebuild All



Download Program



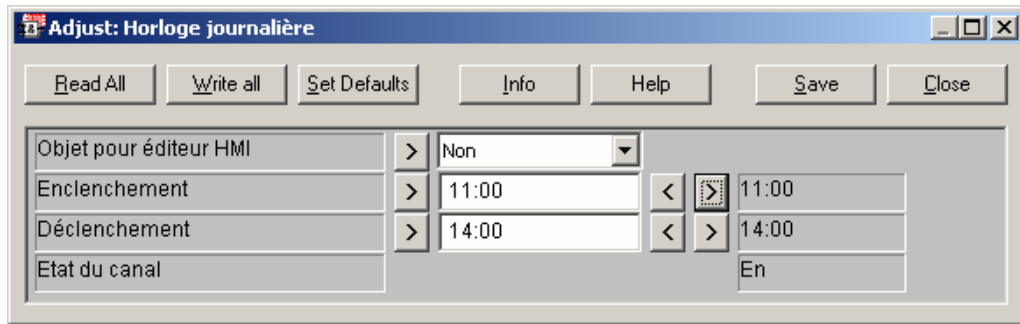
Go Online



Remarque:

Si le programme est étendu avec plusieurs autres Fbox CVC, la Fbox *Initialisation CVC 7* doit être placée qu'une seule fois sur la première page Fupla.

### 4.16.5 Paramètres après le chargement du programme

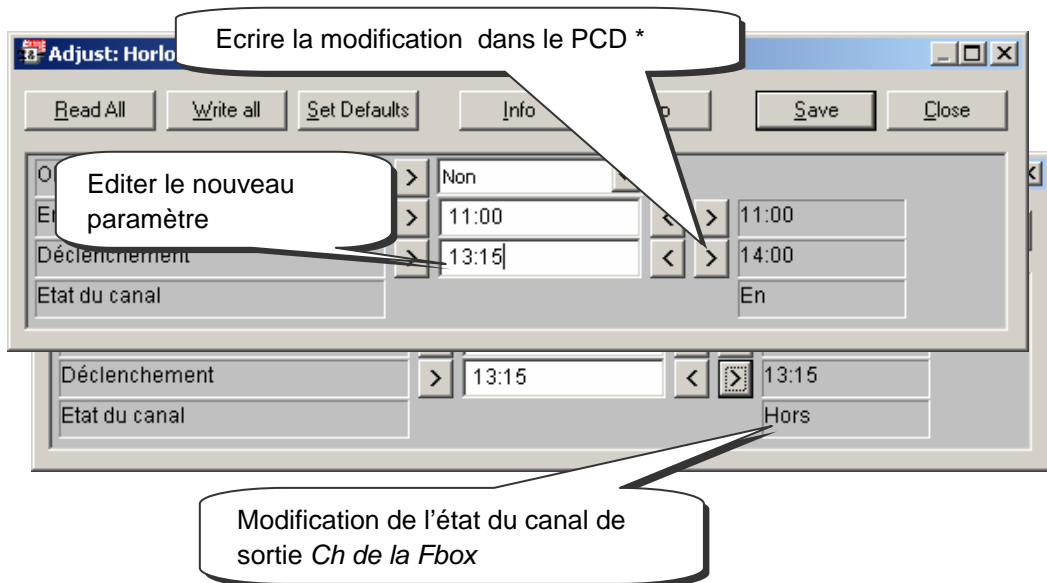


La colonne du milieu représente les paramètres d'enclenchement et de déclenchement de l'horloge journalière définis lors de la programmation.

De manière générale<sup>1</sup>, ce sont les mêmes paramètres en service dans le PCD et représentés dans la colonne de droite. (Paramètres d'enclenchement, déclenchement et l'état du canal de sortie *Ch*)

### 4.16.6 Ecriture de paramètres en ligne.

Lors des tests en ligne il est possible d'éditer une nouvelle valeur des paramètres d'enclenchement et de déclenchement du canal de sortie de la Fbox:



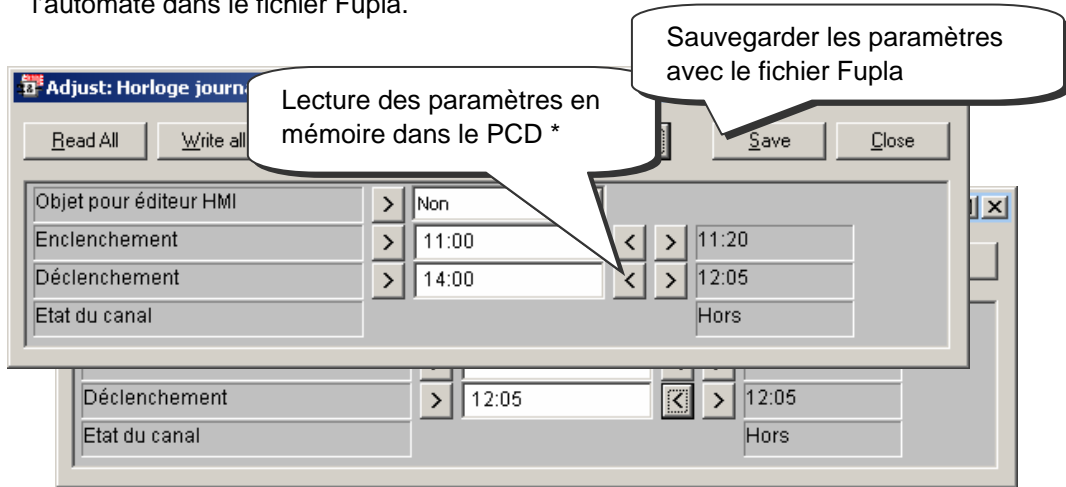
\* Il est aussi possible d'écrire tous les paramètres la fenêtre d'ajustage en sélectionnant le bouton *Write all*

Si les corrections de paramètres sont à conserver dans le programme fupla pour le prochain *Build All*, fermer la fenêtre d'ajustage avec le bouton *Save*, si non utiliser le bouton *Close*

<sup>1</sup> Cela peut être différent selon des options définies dans la fenêtre d'ajustage *Heavac 7* et l'état de l'entrée *Res* lors du démarrage à froid du PCD.

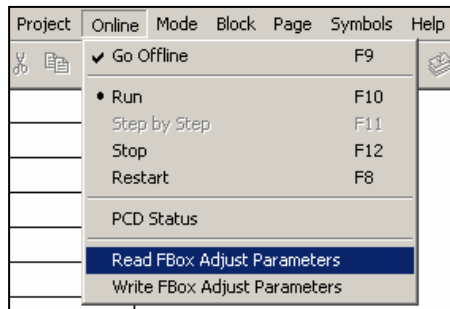
### 4.16.7 Lecture des paramètres en ligne.

Parfois l'utilisateur souhaite lire et sauvegarder les paramètres d'ajustage déjà présent dans la mémoire de l'automate pour le prochain *Build All*. Avec la manipulation suivante, les paramètres sont transférés de la mémoire de l'automate dans le fichier Fupla.



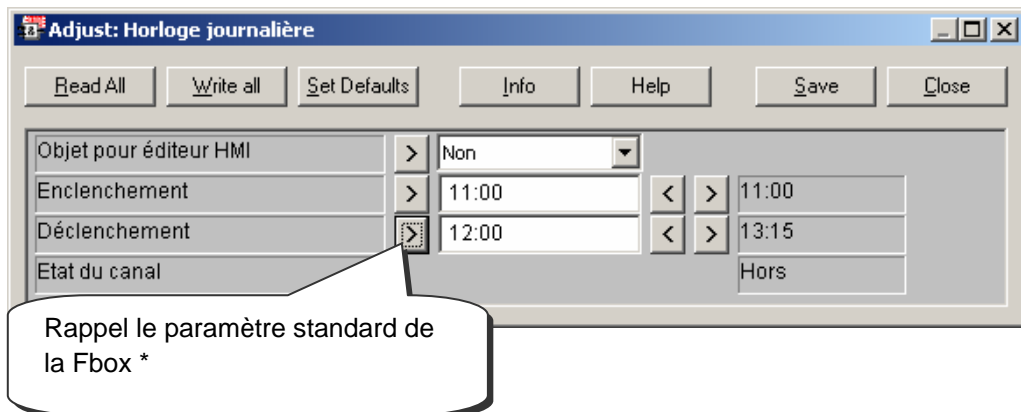
\* Il est aussi possible de lire tous les paramètres la fenêtre d'ajustage en sélectionnant le bouton *Read all*

\* Il est aussi possible de lire tous les paramètres d'ajustage de toutes les Fbox du fichier fupla dans la mémoire du PCD avec la menu *Online, Read Fbox Adjust parameter*



### 4.16.8 Rétablir les paramètres standards

Après de multiples modifications des paramètres, il est toujours possible de rétablir les paramètres standards. Ce sont les paramètres définis lorsque la Fbox est placée pour la première fois dans la page fupla.



\* Il est aussi possible de rétablir tous les paramètres standards de la fenêtre d'ajustage en sélectionnant le bouton *Set Defaults*

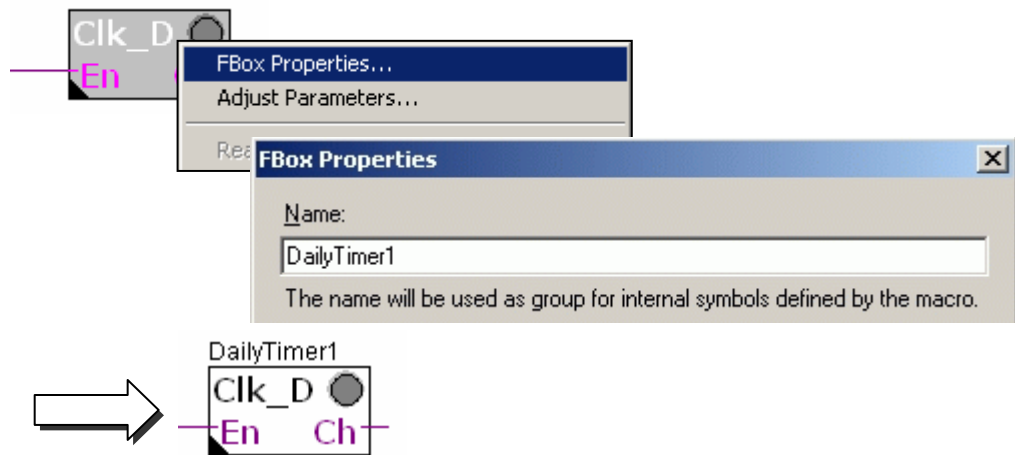
### 4.16.9 Définir les symboles des paramètres d'ajustage

Parfois, il est nécessaire de lire ou écrire les paramètres des fenêtres d'ajustages depuis le programme Fupla, le réseau de communication ou le système de supervision.

Ceci est possible, si un symbole est défini pour désigner les paramètres des flags ou des registres correspondant aux paramètres affichés dans la fenêtre d'ajustage de la Fbox.

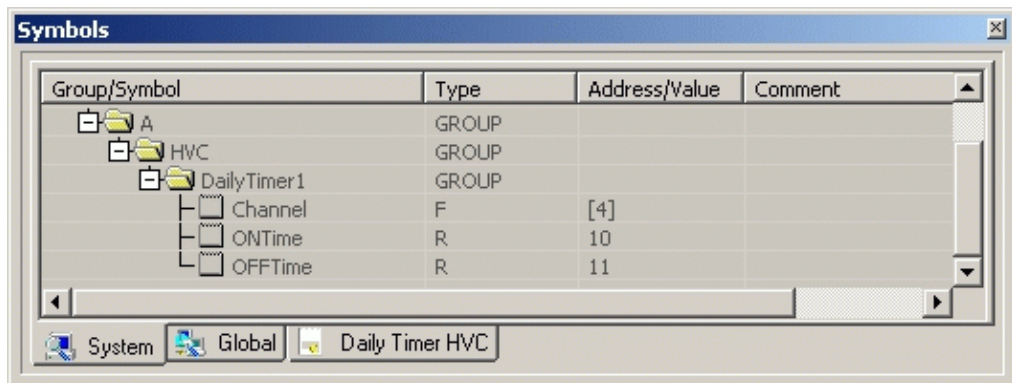
Pour définir ces symboles, placer la souris sur le centre de la Fbox, sélectionner le bouton droit pour faire afficher le menu de contexte.

Sélectionner le menu *Fbox Properties...* Définir un symbole pour désigner un groupe de paramètres liés à la Fbox sélectionnée.



Construire le programme et ouvrir l'éditeur de symboles. Un nouveau répertoire *System* est maintenant visible. Il contient la liste des symboles systèmes du PCD.

Avec la librairie CVC, tous les symboles systèmes correspondants aux paramètres des fenêtres d'ajustage sont groupés dans *A.HVC.name*. (name est le nom de la Fbox ).

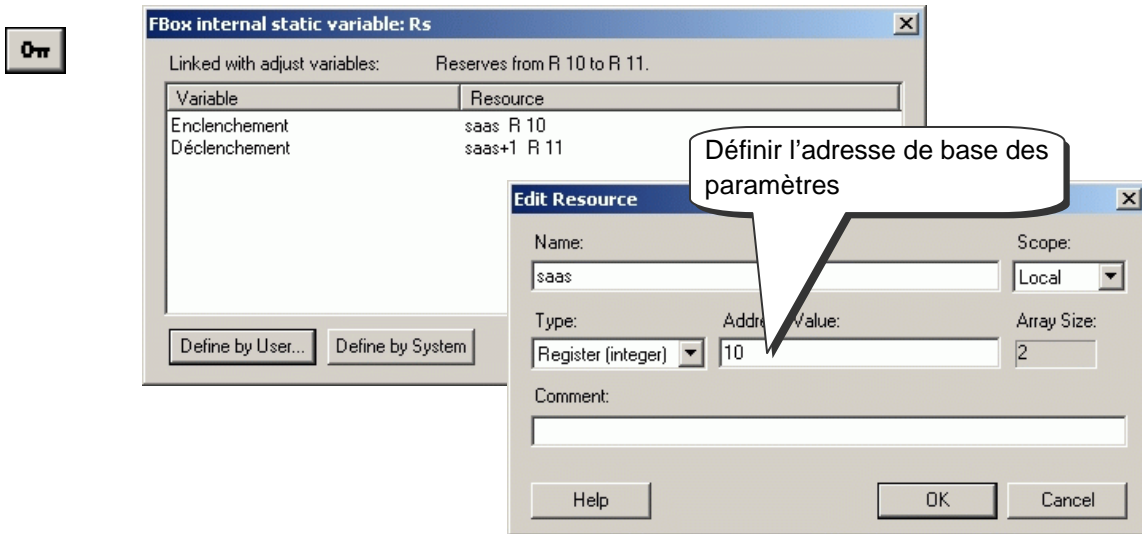
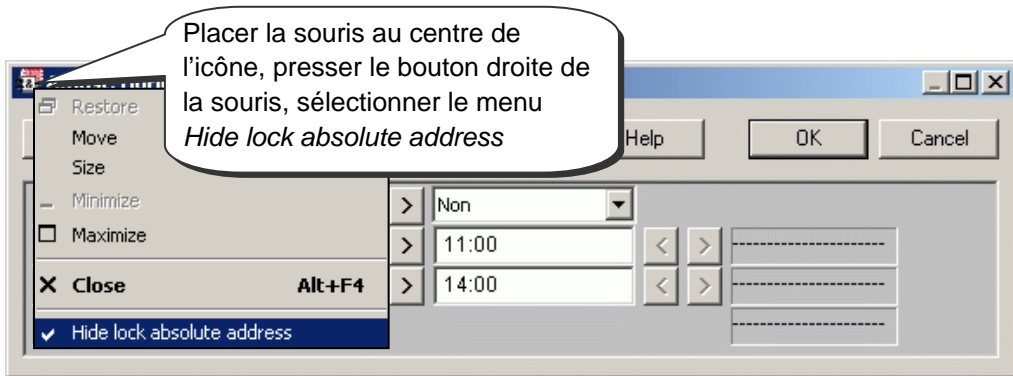


Il ne reste qu' à utiliser ces nouveaux symboles dans le programme Fupla.

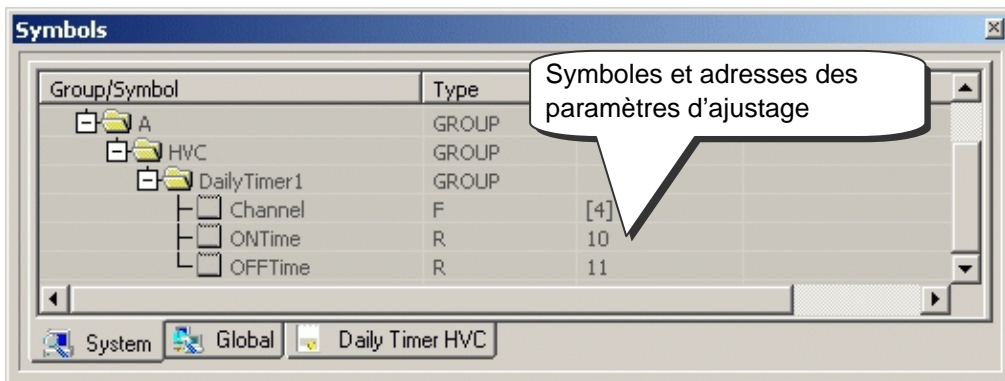


### 4.16.10 Définir les adresses des paramètres d'ajustage

Définir le symbole des paramètres d'ajustage comme présenté précédemment et ajouter l'adresse comme suit:



Construire le programme et ouvrir l'éditeur de symboles. Les symboles systèmes sont complétés avec les adresses des registres définis ci-dessus.





## 4.17 Mise en service d'un module analogique

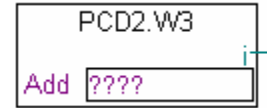
### 4.17.1 Saisie d'une mesure analogique

Les exemples de programme présentés jusque ici exploitent les entrées et sorties digitales en plaçant leurs adresses ou symboles



dans la marge de l'éditeur Fupla.

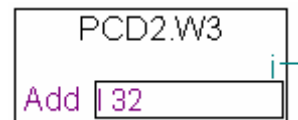
Pour les modules d'entrées et sorties analogiques, il est nécessaire de faire usage d'une Fbox d'acquisition de la grandeur analogique. Ces Fbox sont disponibles avec la librairie *Standard*, *Modules analogiques* ou *Application*, *CVC-Analogue*.



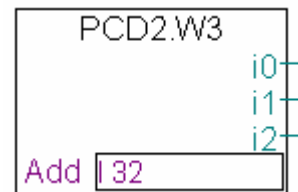
Ces librairies proposent une grande variété de Fbox, chacune correspond à un module analogique dont le nom présenté dans le *Fbox Selector* correspond au numéro d'article du module.

Les fbox analogiques sont étirables, l'utilisateur peut définir le nombre de canaux de mesure nécessaires à son application. Si certains canaux de mesure ne sont pas utilisés, l'addition d'un canal supplémentaire nécessite de redimensionner de la Fbox avec le menu de contexte *Resize Fbox*. La Fbox peut aussi être définie avec le maximum de canaux, mêmes s'ils ne sont pas tous utilisés.

Le champs *Add* permet de définir l'adresse de base du module analogique. Cette adresse indique l'emplacement ou est inséré le module dans le PCD : 0, 16, 32, ...

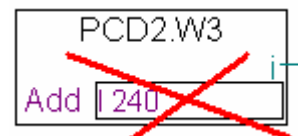


Les mesures analogiques sont directement disponibles sur les sorties I 0 à I 7 de la Fbox. Il est possible des les lier directement à d'autres Fbox ou de sauvegarder la mesure dans un registre. Sauvegarder la mesure dans un registre est une bonne solution, en particulier si la mesure est utilisée sur plusieurs pages différentes du programme.



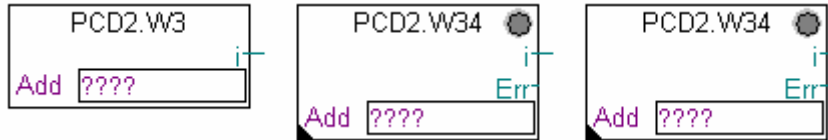
**Attention :**

Prenez garde à ne définir qu'une seule Fbox par module analogique et à ne pas insérer le module analogique à l'adresse du watchdog PCD (255). Si non, la valeur fournie par le module pourrait être erronée.



### 4.17.2 Exemple pour les entrées analogiques PCD2.W340

Si le PCD est équipé d'un module de 8 entrées analogiques universelles PCD2.W340, l'utilisateur peut utiliser l'une des Fbox Fupla ci-dessous et la définir avec le nombre de canaux de mesure nécessaires.



Fbox : PCD2.W3, PCD2.W34, PCD2.W34 avec erreur

Les unités de la mesure sont dépendantes du choix du module, de la Fbox et des paramètres d'ajustage.

Le module PCD2.W340 est universel, il supporte les mesures 0..10V, 0..2.5V, 0..20 mA et les sondes de température Pt/Ni 1000. Un pont doit être sélectionné sur le module pour définir la plage de mesure. La résolution de mesure est de 12 bits, soit 4095 états de mesures distincts. (Pour de plus amples renseignements concernant les modules, veuillez vous référer à votre manuel hardware PCD)

La Fbox *PCD2.W3* fournit une mesure brute. Pour ce module d'une résolution de 12 bits, cela correspond à une grandeur de mesure comprise entre 0 et 4095. C'est à l'utilisateur de convertir la mesure dans une unité physique standard.

La Fbox *PCD2.W34* est plus élaborée, une fenêtre d'ajustage permet de définir les unités de mesures pour chacun des canaux. La LED de la Fbox devient rouge si l'une des mesures sort de la plage de mesure valide : rupture ou court-circuit sur les câbles de la sonde. L'erreur peut être quittancée avec le bouton *Quittance* disponible dans la fenêtre d'ajustage.

Erreur...	Quittance	
---[Configuration des canaux 0 à 7]---		
Ch 0 / Mode ou type de sonde	>	mV
Ch 1 / Mode ou type de sonde	>	Ni 1000
Ch 2 / Mode ou type de sonde	>	uA

La Fbox *PCD2.W34 avec erreur* offre les mêmes services pour la conversion des unités mais dispose d'une sortie erreur pour indiquer quel est le canal en erreur et un paramètre d'ajustage supplémentaire pour définir une valeur de mesure par défaut lors d'une erreur.

Sortie en cas d'erreur	>	Ancienne valeur
------------------------	---	-----------------

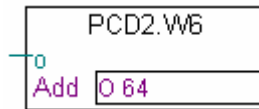
### 4.17.3 Exemple pour les sorties analogiques PCD2.W610.

Le principe est le même que pour les entrées, l'utilisateur place une Fbox correspondant au module analogique de sortie dans la page Fupla, l'étend pour sélectionner le nombre de canaux de sorties et défini l'adresse de base du module.

Contrairement aux Fbox pour les entrées, les valeurs de consigne pour les sorties analogiques sont représentées sur le coté gauche de la Fbox.

Il est possible de lier ces entrées directement à d'autres Fbox ou à des registres définis dans la marge gauche de la page Fupla.

Si le PCD est équipé d'un module de 4 sorties analogiques universelles PCD2.W610, l'utilisateur utilise la Fbox ci-dessous pour sortir un courant 0...20 mA ou 0...10 V.



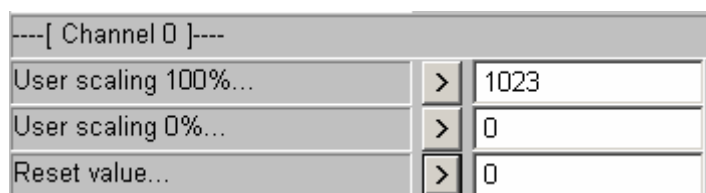
Fbox : PCD2.W6

Un pont doit être sélectionné sur le module pour définir la plage de sortie. La résolution de ce module est de 12 bits, soit 4095 états de consignes distincts.

La valeur entière appliquée sur l'entrée de la Fbox détermine la tension ou le courant de sortie du canal :

Valeur d'entrée sur la Fbox	Tension de sortie [V]	Courant de sortie [mA]
0	0	0
2047	5	10
4095	10	20

D'autres Fbox comme pour le module PCD2.W605 ( 6 sorties 0...10 V avec séparation galvanique) disposent d'une fenêtre d'ajustage pour adapter la plage des valeurs de consignes appliquées sur l'entrée de la Fbox :



Les paramètres *User scaling 0 et 100%* permettent de définir la valeur à appliquer sur l'entrée de la Fbox pour la tension minimum et maximum du canal.

Le paramètre *Reset value*, correspond à la valeur appliquée sur le canal lors de la mise sous tension du PCD.



## Table des matières

---

<b>5</b>	<b>STRUCTURATION DU PROGRAMME</b>	<b>3</b>
<b>5.1</b>	<b>Au sommaire de ce chapitre</b>	<b>3</b>
<b>5.2</b>	<b>Blocs d'organisation cyclique (COB 0 à 15)</b>	<b>4</b>
5.2.1	Définition	4
5.2.2	Exemple	5
5.2.3	Ajouter une structure	5
5.2.4	Délai de surveillance	6
<b>5.3</b>	<b>Bloc de programmes (PB 0 à 299)</b>	<b>7</b>
5.3.1	Définition	7
5.3.2	Exemple	7
<b>5.4</b>	<b>Blocs de fonctions (FB 0 à 999)</b>	<b>9</b>
5.4.1	Définition	9
5.4.2	Exemple d'appel à un bloc de fonction	9
<b>5.5</b>	<b>Affichage de la structure des blocs</b>	<b>10</b>
<b>5.6</b>	<b>Blocs d'exceptions (XOB 0 à 31)</b>	<b>11</b>
5.6.1	Définition	11
5.6.2	Aperçu des XOB de la famille PCD	12
5.6.3	Application des XOB	13
5.6.4	Historique	16
5.6.5	Description des XOBs	17
<b>5.7</b>	<b>Blocs séquentiels (SB 0 à 31, 96 )</b>	<b>21</b>
<b>5.8</b>	<b>Tableau récapitulatif.</b>	<b>21</b>



## 5 Structuration du programme

---

### 5.1 Au sommaire de ce chapitre

Le succès d'un bon programme est lié à sa structure. Elle rend le programme facile, rapide à entretenir et à développer. Le langage de programmation SAIA PCD est un langage structuré. Il propose différents blocs d'organisation dans lesquels l'utilisateur dépose les instructions de son application. Chaque bloc offre un service particulier à l'utilisateur. Les blocs d'organisation disponibles sont les suivants: blocs d'organisation cyclique (COB), blocs de fonctions (FB), blocs de programme (PB), blocs d'exception (XOB) et les blocs séquentiels (SB)

## 5.2 Blocs d'organisation cyclique (COB 0 à 15)

### 5.2.1 Définition

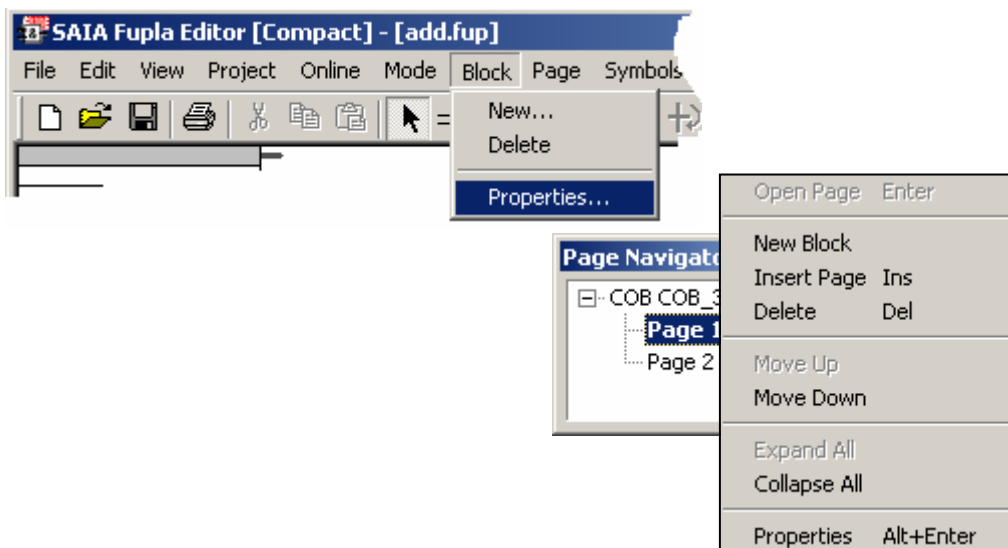
Les blocs d'organisation cyclique (COB) s'appliquent à des parties de programmes exécutées sans aucune boucles de programmes, ni attentes sur des événements internes ou externes au PCD.

Au démarrage du PCD, le programme commence par exécuter le COB 0, puis l'un après l'autre, les COBs 1 à 15, s'ils sont présents dans le programme. Tous sont automatiquement appelés successivement, dans une boucle sans fin.

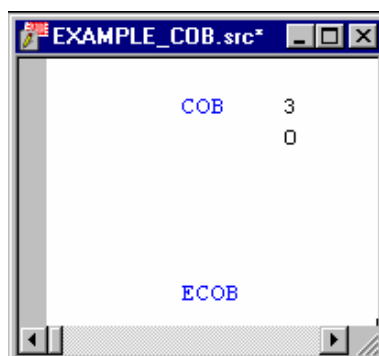
Tous les signaux nécessitant un traitement cyclique (fins de course pour commandes d'axe, signaux de coupure d'alimentation externe ou d'arrêt d'urgence, protections opérateur...) doivent être pris en compte dans un COB.

Le PCD doit avoir au moins un COB !

Il est important de bien comprendre la notion des blocs d'organisation cycliques. L'absence de boucles d'attente ne constitue pas un défaut de programmation, mais une sécurité. C'est en effet le seul moyen de garantir que les signaux importants de l'application sont contrôlés à intervalles de temps réguliers.



Si vous rédigez les programmes avec l'éditeur Fupla, celui-ci ouvre automatiquement le nouveau fichier avec un COB. Libre à vous de modifier le type de bloc ou le commentaire à l'aide du menu *Block, Properties*. Dans les programmes en liste d'instructions, le bloc est défini par les instructions dans le code programme.





### 5.2.2 Exemple

Voici un exemple de programme (en langage IL et en Fupla) assurant le clignotement de la sortie 64 toutes les 1,5 secondes. Ce programme est écrit dans le COB 0, il est suivi d'autres COB 1 à COB 15.

#### Programme IL

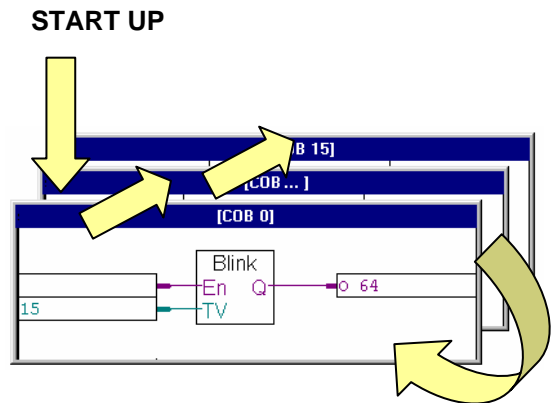
```

COB 0      ;Début de COB 0
0          ;Délai de surveillance
STL T 1    ;Si temporisateur T1 = 0,
LD T 1     ;charger 1,5 s.
          15
COM O 64   ;et inverser la sortie O64
ECOB      ;Fin de COB 0

COB 1      ;Bloc suivant
0
NOP
ECOB

;COBs suivants
    
```

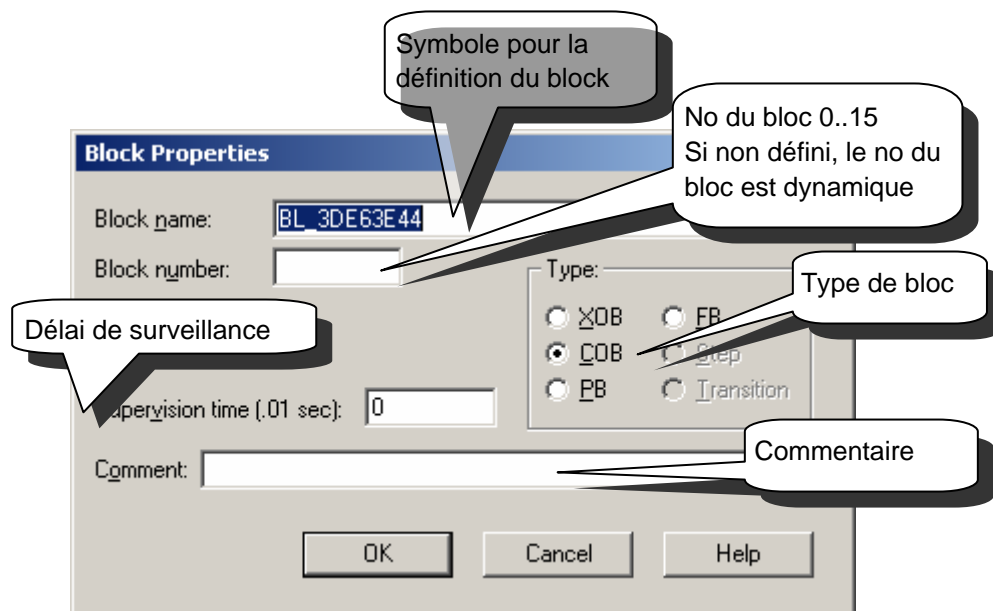
#### Programme Fupla



Fbox: Clignoteurs, Clignoteur 1 temps

### 5.2.3 Ajouter une structure

Le fichier Fupla peut recevoir plusieurs blocs de programme que l'on ajoute, efface ou édite à l'aide du menu *Block*.



#### 5.2.4 Délai de surveillance

Le Délai de surveillance permet de définir le temps maximum pour traiter le COB du début jusqu'à la fin. Passé ce délai, deux cas peuvent se présenter :

Le XOB 11 n'est pas programmé, alors le COB est abandonné pour traiter les COBs suivants jusqu'au dernier, la lampe erreur est allumée. Au cycle de programme suivant, le COB dont le temps de supervision à été dépassé et redémarré avec un nouveau délai de supervision là où il à été interrompu.

Le XOB 11 est programmé, alors le COB est abandonné pour traiter le XOB 11. A la fin du XOB 11, le COB dont le temps de supervision à été dépassé et redémarré avec un nouveau délai de supervision là ou il à été interrompu. La lampe erreur n'est pas allumée, car le défaut à été prévu et pris en charge par le programme utilisateur.

Un délai de surveillance de zéro signifie que le délai de surveillance est désactivé.

### 5.3 Bloc de programmes (PB 0 à 299)

#### 5.3.1 Définition

Le PG5 permet de travailler également avec des blocs de programmes (PB). Ils offrent un bon moyen d'organiser la hiérarchie des programmes. Précisons que les PB ne sont traités que sur l'appel d'un COB, PB, FB ou d'un bloc séquentiel SB.

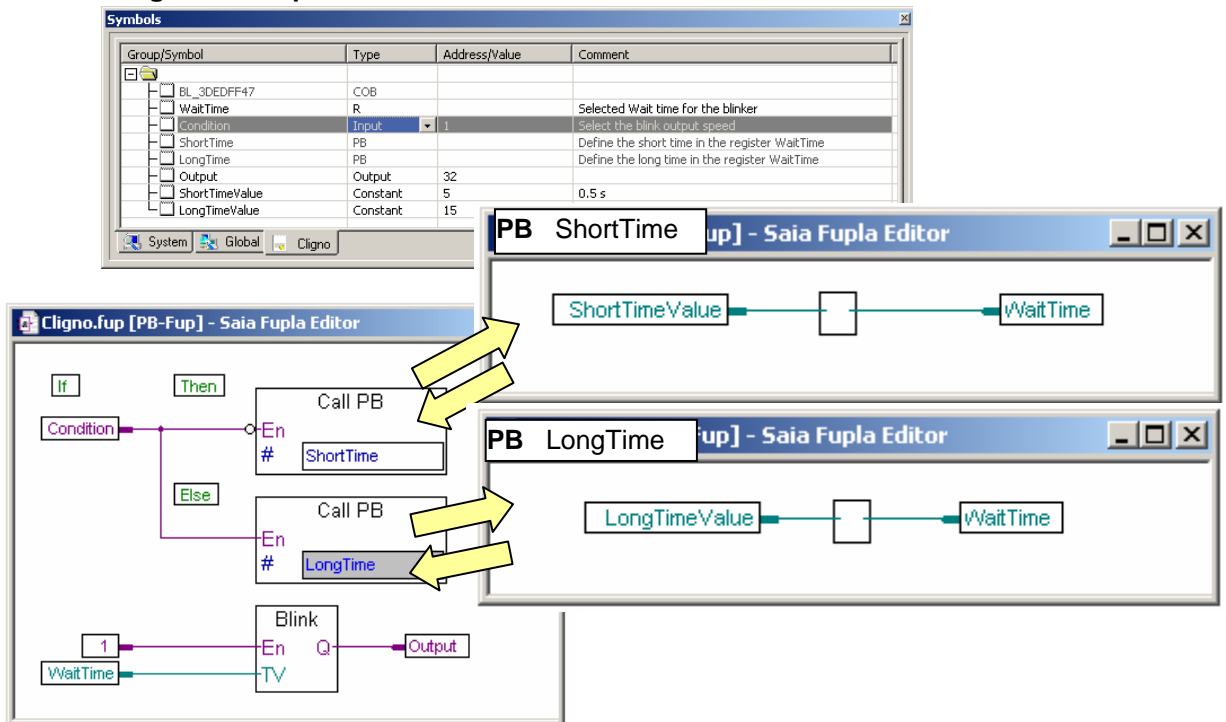
Nous distinguons deux types d'appels: conditionnel et inconditionnel. Le premier dépend du résultat d'une opération logique, le second ne dépend d'aucune condition. Il est possible d'appeler un même PB plusieurs fois dans le programme.

Un PB peut aussi appeler un deuxième PB, qui lui appel un troisième et ainsi de suite jusqu'à sept niveaux d'imbrication. Au-delà de 7 niveaux d'imbrication, le PCD fait appel à un XOB 10.

#### 5.3.2 Exemple

L'exemple qui suit est un clignoteur à deux vitesses dépendant de l'état logique de l'entrée *Condition*.

Programme Fupla:



Si l'entrée digitale *Condition* est à l'état logique bas, le PCD appelle le PB *ShortTime* et transfère une constante *ShortTimeValue* de 5 dans le registre *WaitTime*, si non le PCD appelle le PB *LongTime* et transfère une constante *LongTimeValue* de 15 dans le registre *WaitTime*. Le registre *WaitTime*, définit le temps de pause entre deux changements d'état du clignoteur *Blink*. Pour assurer l'initialisation du registre *WaitTime*, lors du démarrage à froid, le clignoteur doit être placé après l'appel des deux PB.

Programme IL:

The screenshot shows the SAIA IL Editor interface. At the top, there is a menu bar with options: File, Edit, Search, View, Project, Online, Tools, Symbols, Window, Help. Below the menu is a symbol table with the following data:

Group/Symbol	Type	Address/Value	Comment
ShortTimeValue	Constant	5	0.5 s
LongTimeValue	Constant	15	1.5 s
Tempo	Timer		

Below the symbol table, there are tabs for 'System', 'Global', and 'cligno'. The main editing area contains the following IL code:

```

COB      0
          0
STH      Condition           ;IF (Condition is high)
CPB      L ShortTime         ; THEN Call PB ShortTime
CPB      H LongTime          ; ELSE Call PB LongTime
ECOB

PB       ShortTime
STL     Tempo                ;IF tempo is low
LD      Tempo                ; load it with a short value
        ShortTimeValue
COM     Output               ; invert the output
EPB

PB       LongTime
STL     Tempo                ;IF tempo is low
LD      Tempo                ; load it with a long value
        LongTimeValue
COM     Output               ; invert the output
EPB
  
```

## 5.4 Blocs de fonctions (FB 0 à 999)

### 5.4.1 Définition

Les blocs de fonctions FB s'apparentent en bien des points aux PB. Comme eux, ils contiennent des parties de programme pouvant être appelées par d'autres blocs, de façon conditionnelle ou inconditionnelle. La seule différence réside dans la possibilité d'appeler les FB avec des paramètres, ce qui n'est pas le cas des PB.

Les FB offrent une solution idéale pour développer des bibliothèques de programmes utilisables par différents projets. Ce qui contribue à diminuer le temps de mise en service. Notons que les FB avec paramètres ne peuvent être utilisés qu'à partir d'un programme en liste d'instructions.

Les appels de blocs de fonction peuvent s'imbriquer les uns dans les autres jusqu'à maximum 7 niveaux. Au-delà de 7 niveaux d'imbrication, le PCD fait appel à un XOB 10.

### 5.4.2 Exemple d'appel à un bloc de fonction

L'exemple suivant illustre un FB faisant clignoter une sortie. Le bloc de fonctions est appelé deux fois : il fait clignoter d'abord la sortie 64 toutes les 1,5 secondes, puis la sortie 65, toutes les 3 secondes.

```

      FB      1          ;Début du FB

tempo DEF  = 1          ;[T]  Adresse d'un temporisateur
delay  DEF  = 2          ;[W]  Pause entre deux inversions du clignoteur
blinker DEF  = 3          ;[O,F] Adresse du clignoteur

      STL    =tempo      ;Si la temporisation est à l'état bas
      LDL    =tempo      ; initialise la tempo avec le paramètre =2
           =delay

      COM    =blinker    ; inverse le paramètre =3
      EFB

      COB    0
           0

      CFB    1          ;Appel le FB pour la première fois
           T 1
      15
           O 64

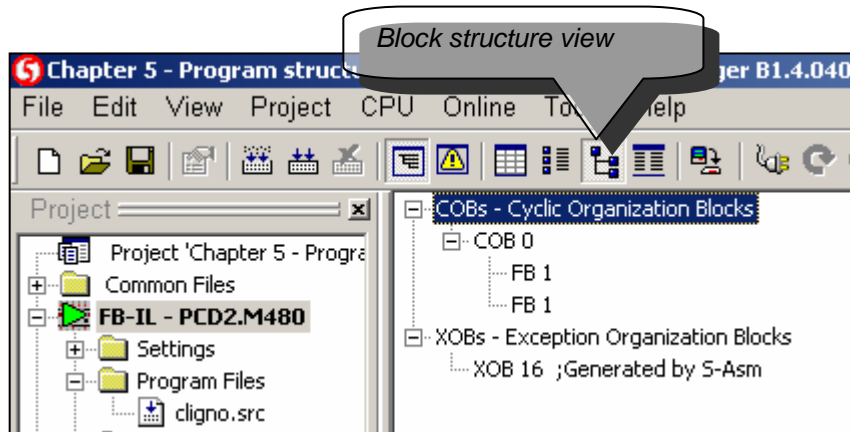
      CFB    1          ;Appel le FB pour la seconde fois
           T 2
           30
           O 65
      ECOB

```

## 5.5 Affichage de la structure des blocs

Après la construction du programme, le bouton *Block structure view* du gestionnaire de projets permet de visualiser la structure des blocs d'organisation.

La figure ci-dessous représente l'exemple de la page précédente. Nous remarquons qu'un FB1 est appelé à deux reprises par le COB 0.



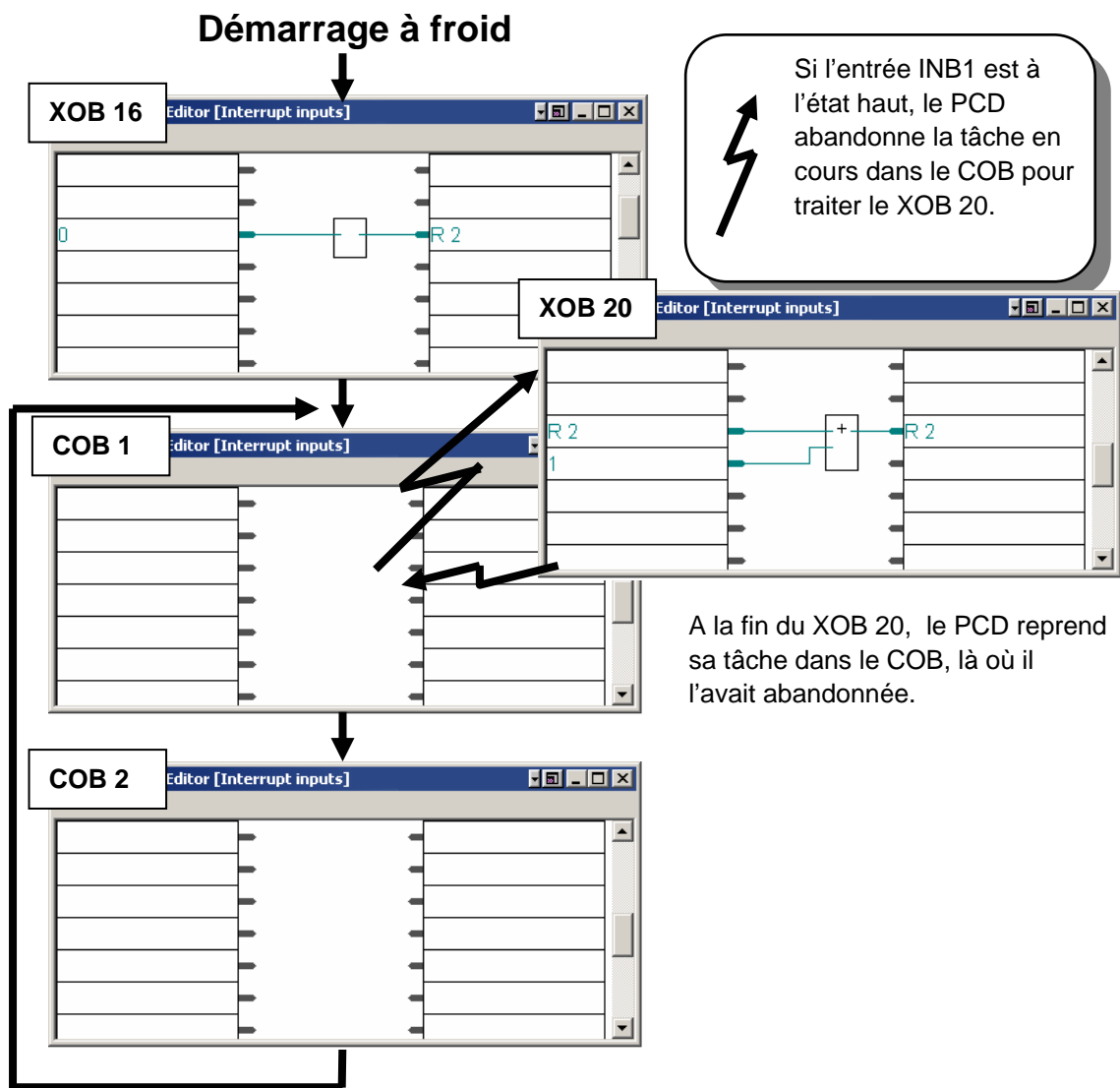
## 5.6 Blocs d'exceptions (XOB 0 à 31)

### 5.6.1 Définition

Les blocs d'exceptions sont des structures programmes qui sont appelés automatiquement dès qu'il se produit un événement particulier. Chaque événement hardware ou software est lié à un XOB. Ces événements ne peuvent pas être modifiés par l'utilisateur. Par contre, l'utilisateur est libre de programmer les actions à entreprendre à l'intérieur de chacun des XOB.

**Exemple:**

A la mise sous tension, l'automate initialise le registre 2 avec la valeur zéro. Ce registre sert à compter les impulsions de l'entrée interruptive INB1 (Fréquence maximum de 1 kHz).



**Exemple:**

Mettez le PCD sous tension et retirez la pile: la lampe *Error* s'allume. Si le programme contient un XOB 2 la lampe ne s'allume plus. Le XOB 2 est automatiquement exécuté.

**5.6.2 Aperçu des XOB de la famille PCD**

<b>XOB</b>	<b>Description</b>	<b>Priorité</b>
0	Problème d'alimentation du rack principal	4
1	Problème d'alimentation dans un rack d'extension (PCD6)	2
2	Problème de batteries ou batterie déchargée	2
4	Erreur de parité sur le bus d'adresse (PCD6)	1
5	Pas de réponse d'un module I/O (PCD4/6)	1
7	Surcharge du système (événements multiples)	3
8	Instruction non-valide	4
9	Trop de branches Graftec actives	1
10	Plus de 7 niveaux d'appel PB/FB	1
11	Temps de supervision du COB dépassé	3
12	Dépassement de la capacité du registre d'index	1
13	Flag erreur positionné	1
14	Interruption cyclique	3
15	Interruption cyclique	3
16	Démarrage à froid du PCD	4
17	Demande d'interruption d'un XOB via S-Bus	3
18	Demande d'interruption d'un XOB via S-Bus	3
19	Demande d'interruption d'un XOB via S-Bus	3
20	Entrée interruptive « INB1 »	3
25	Entrée interruptive « INB2 »	3
30	Perte de liaison maître esclave avec les RIO	1

Si une erreur se produit et que les XOB correspondants ne sont pas programmés, la lampe erreur sur la face du PCD est allumée et le programme utilisateur continue son travail.

Si une erreur se produit et que les XOB correspondants sont programmés, la lampe erreur sur la face du PCD reste éteinte, le bloc d'exception XOB est appelée.

Un mécanisme de priorité des XOB garantit le traitement des XOB les plus importants. La priorité de niveau 4 est la priorité la plus haute.



### 5.6.3 Application des XOB

Les erreurs ci-dessous se laissent facilement diagnostiquer par les XOB correspondants :

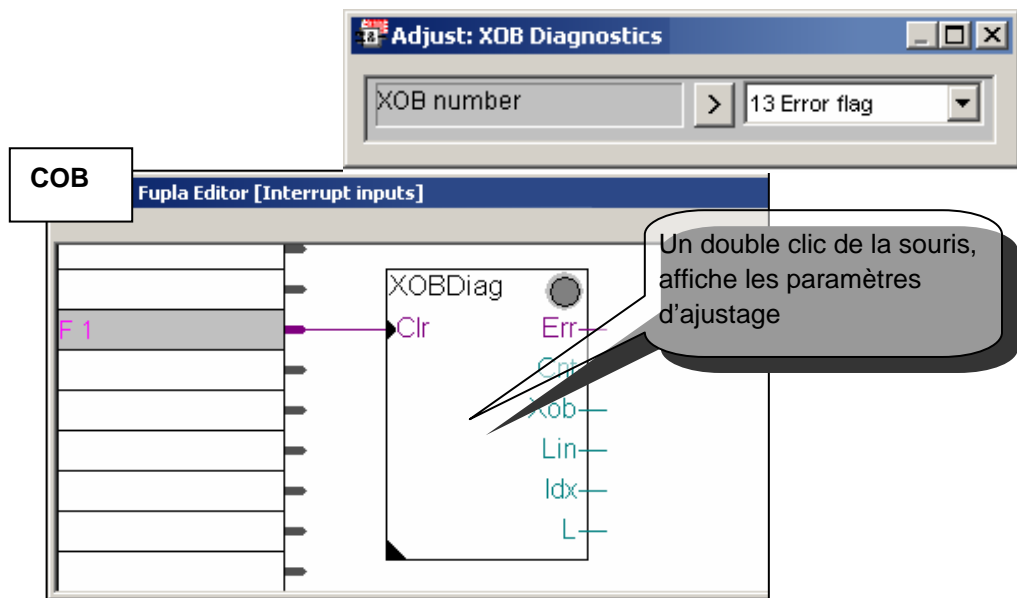
- erreur d'adressage des cartes I/O
- plus de 7 niveaux de programme
- plus de 32 transitions Graftec actives
- boucle de programme
- erreur mathématique
- erreur de communication

**Exemple Fupla:**

Localiser systématiquement une erreur de programme utilisateur en employant tous les outils disponibles à cet effet.

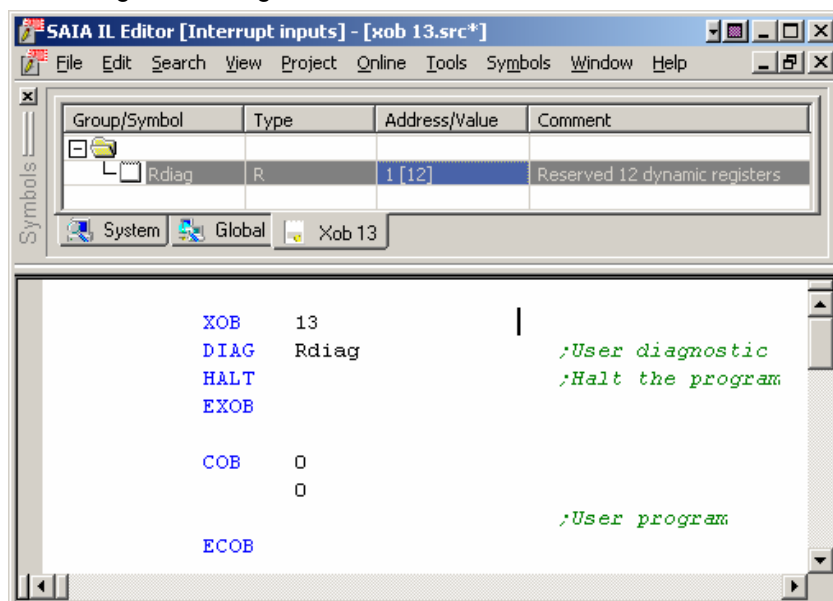
Avec Fupla, il n'est même pas nécessaire de créer les blocs XOB, ils sont automatiquement ajoutés par la Fbox : *Spécial, Diagnostique XOB*

Les informations relatives au diagnostic sont disponibles sur les sorties de la fonction, compteur d'erreur, numéro du XOB , no de ligne programme, ...



**Exemple IL:**

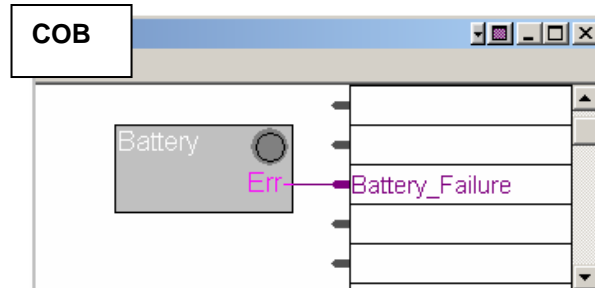
Avec le programme IL, le diagnostic fourni les mêmes informations que ci-dessus dans les registres Rdiag + 0 ... +12



Entretien du PCD :  
Surveillance des piles (à remplacer tous les 3 à 4 ans)

### Exemple Fupla:

Avec Fupla, il n'est même pas nécessaire de créer le bloc XOB 2, il est automatiquement ajouté par la Fbox: *Spécial, Batterie*  
La sortie *Battery\_Failure* est à l'état haut en cas de problème de batterie.



### Exemple IL:

Lorsque la batterie du PCD est en défaut, la lampe batterie sur la face frontale s'allume, le bloc d'exception XOB 2 est alors automatiquement appelé à intervalles de temps régulier.

Avec l'exemple ci-dessous, le XOB 2 charge une temporisation avec un délai de 1 seconde. Comme le bloc d'exception est appelé régulièrement, la temporisation est fréquemment initialisée et n'a pas la possibilité de décrémente jusqu'à zéro. L'état binaire de cette temporisation est alors haut en cas de défaut de batterie et tombe à l'état bas, environ 1 seconde après avoir effectué le remplacement de la batterie.

The screenshot shows the SAIA IL Editor interface. The 'Symbols' table is visible, showing the configuration for 'Battery\_Failure' as a 'Timer' with an address of '1'. Below the table, the ladder logic code for XOB 2 is displayed:

```

XOB 2 ;Alarm Battery
LD Battery_Failure
k 10
EXOB

COB 0 ;User program
0

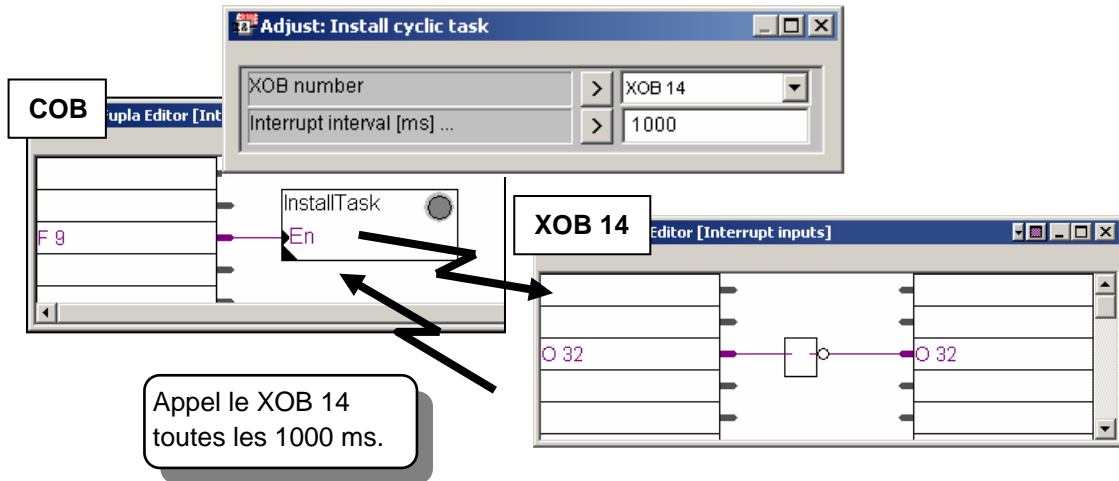
STH Battery_Failure
ECOB

```

Surveillance d'événements particuliers ou réponses rapide à des événements externes:  
 entrées interruptives  
 interruption occasionnelle du programme  
 interruption du programme sur réception d'un télégramme S-Bus  
 démarrage à froid : rétablissement des valeurs initiales

**Exemple Fupla:**

Sortir des impulsions sur une sortie digitale. Utiliser les Fbox *Spécial*, *Installe tâches cycliques* et *Binaire*, *Transfert direct*.



**Exemple IL:**

```

XOB      16
SYSWR   4014      ;Initialise le XOB 14
1000    ;avec une interruption de 1000 ms
EXOB

COB      0
        0
                                ;Programme utilisateur

ECOB

XOB      14      ;Interruption cyclique
COM     O 32    ;avec inversion de la sortie O 32
EXOB
    
```

### 5.6.4 Historique

L'historique de fenêtre *PCD History Table* protocal toutes les erreurs matérielles ou logicielles survenues sur votre PCD. Cette table est mise à jour en permanence, même si l'on ne programme pas les XOB.



L'historique se consulte à l'aide du bouton *Online configurator* ou du menu Tool, Online Configurator

*Online Configurator*

Remarques:

Chaque unité centrale a son historique.

L'erreur *BATT FAIL* n'existe que sur l'unité centrale 0.

Si l'erreur est imputable à une ligne de programme, celle-ci est précisée. Sinon, elle est affichée en hexadécimal.

Le XOB 0 n'apparaît que s'il est programmé.

Reason	Address	Time	Date
>7 CALL LEVELS	30	14:09:43	06/01/2003
>7 CALL LEVELS	30	14:09:43	06/01/2003
>7 CALL LEVELS	30	14:09:43	06/01/2003
>7 CALL LEVELS	30	14:09:43	06/01/2003
>7 CALL LEVELS	30	14:09:44	06/01/2003
>> >7 CALL LEVELS	30	14:09:44	06/01/2003
>7 CALL LEVELS	30	14:09:44	06/01/2003
>7 CALL LEVELS	30	14:09:44	06/01/2003
>7 CALL LEVELS	30	14:09:44	06/01/2003
>7 CALL LEVELS	30	14:09:44	06/01/2003
>7 CALL LEVELS	30	14:09:44	06/01/2003
>7 CALL LEVELS	30	14:09:44	06/01/2003
>7 CALL LEVELS	30	14:09:44	06/01/2003
>7 CALL LEVELS	30	14:09:44	06/01/2003
BATT FAIL	816 0	14:09:43	06/01/2003
IR OVERFLOW	0 0	12:00:00	06/01/2003
ERROR FLAG	772 6	14:09:44	06/01/2003

Help

Clear History

Close

Date et heure  
Ligne de programme  
Compteur d'erreurs  
Descriptif de l'erreur  
Erreur la plus récente

## 5.6.5 Description des XOBs

### **XOB 0: Problème d'alimentation du rack principal**

(Power failure in main rack)

Le régulateur de tension du module d'alimentation du rack principal a détecté une chute de tension importante. Dans un tel cas, toutes les sorties sont déclenchées comme suit:

pour le PCD1/PCD2/PCD4: immédiatement

pour le PCD6: après 1.5 ms.

Le XOB 0 est appelé et tous les CPUs sont mis en HALT. A partir du moment où le XOB 0 est appelé, il s'écoule approximativement 5 ms avant le CPU HALT. Pendant ce temps, le XOB 0 continue son exécution et des données peuvent être sauvées.

### **XOB 1: Problème d'alimentation dans un rack d'extension (PCD6)**

(Power failure in extension rack)

Le régulateur de tension du module d'alimentation d'un rack d'extension (PCD2 ou PCD6) a détecté une chute de tension excessive. Dans ce cas, toutes les sorties du rack d'extension sont déclenchées dans les 2 ms et le XOB 1 est appelé. Si les sorties de ce rack d'extension «mort» continuent d'être adressées par le programme utilisateur, le XOB 4 et/ou le XOB 5 seront appelés.

### **XOB 2: Problème de batteries ou batterie déchargée**

(Battery failure or low battery)

La batterie est déchargée ou est absente. Les informations des indicateurs non-volatiles, des registres ou du programme utilisateur en RAM ainsi que l'horloge peuvent être altérées. Après une longue période d'inutilisation du PCD (plus de 2 mois) un problème de batterie peut être indiqué sans que cela entraîne la perte de données. Un PCD neuf peut également présenter ces symptômes.

### **XOB 4: Erreur de parité sur le bus d'adresse (PCD6)**

(Parity error on address bus)

Le XOB 4 est seulement invoqué par un PCD6 comportant des racks d'extension. Le circuit surveillant le bus d'adresse a détecté une erreur de parité. Cette erreur peut être produite par un câble d'extension ou un rack d'extension défectueux ou par ce que le rack adressé n'existe pas. S'il y a une erreur, le mauvais élément ne peut être adressé.

### **XOB 5: Pas de réponse d'un module I/O (PCD4/6)**

(No response from I/O module)

Les modules d'entrées/sorties du PCD retournent un signal au CPU qui les a adressés. Si ce signal n'est pas renvoyé, le XOB 5 est appelé. Généralement, cet appel est dû à l'absence du module adressé, mais il peut également s'agir d'un mauvais décodage d'adresse de la part du module. Pour les modules PCD4 comportant seulement 8 éléments, le XOB 5 n'est pas appelé si un des éléments absent est adressé (l'adresse est malgré tout décodée et le signal de retour est envoyé). Pour le PCD6, seuls les nouveaux modules PCD6 retournent ce signal. Les modules PCA2 ne provoquent pas l'appel du XOB 5, même en cas d'absence.

**XOB 7: Système surchargé**

Le mécanisme d'attente pour les XOBs de niveau 2 et 3 est surchargé.

Si un XOB de niveau 2 ou 3 est traité au même instant qu'un XOB de priorité plus élevée (niveau 4), alors le XOB de priorité inférieure est mis en attente jusqu'à ce que le XOB prioritaire soit terminé. Le XOB 7 est appelé lorsque la file d'attente est pleine

**XOB 8: Instruction non-valide**

(Invalid opcode)

Le CPU a détecté un code d'instruction non-valide. Si un programme utilisateur est assemblé, chaîné et chargé dans le PCD: il n'est pas possible qu'une instruction non-valide se présente. En effet, la syntaxe du programme est vérifiée par l'éditeur IL et par l'assembleur. Néanmoins, si le programme utilisateur est modifié par le Debugger ou par la console à main, n'importe quelle erreur peut être introduite et causer l'appel du XOB 8. Les erreurs qui sont produites de cette manière sont: l'appel d'un bloc inexistant, une fin de bloc manquante, un saut sur la seconde ligne d'une instruction multi-lignes, un saut d'un bloc dans un autre, ...

**XOB 9: Trop de branches Graftec actives**

(Too many active Graftec branches)

Plus de 32 branches Graftec ont été activées simultanément dans un bloc séquentiel (SB). Plus de 32 branches parallèles peuvent être programmées dans un même SB, mais maximum 32 peuvent être activées simultanément.

**XOB 10: Plus de 7 niveaux d'appel PB/FB**

(More than 7 nested PB/FB calls)

Les PBs et les FBs peuvent s'imbriquer jusqu'à 7 niveaux. Un appel supplémentaire (appel du 8ème niveau) provoque l'exécution du XOB 10. L'appel du 8ème niveau n'est pas exécuté.

**XOB 11: Temps de supervision du COB dépassé**

(COB monitoring time exceeded)

Si la seconde ligne de l'instruction COB comporte un temps de supervision (en 1/100 de secondes) et que le temps d'exécution du COB dépasse le temps de supervision: le XOB 11 est appelé. Le temps d'exécution du COB est le temps qui s'écoule entre les instructions COB et ECOB. Le but original de ce temps de supervision est la recherche et l'éradication des blocages ou des délais pouvant survenir dans le programme utilisateur tels que boucles d'attente, boucle de comptage, ... On peut considérer le temps de supervision comme un «chien de garde» (watch-dog) software. Comme mentionné précédemment, les boucles d'attente et de comptage (utilisation de sauts) doivent être évitées; ceci minimise la possibilité de bloquer le programme utilisateur. Néanmoins, dans un programme correctement structuré, un ou plusieurs COBs peuvent comporter de très longs calculs mathématiques nécessitant un temps d'exécution assez long. Dès lors, les autres COBs comportant seulement des fonctions de contrôle ou de surveillance sont retardés. Si le temps de supervision défini pour le COB comportant de longs calculs est écoulé, le COB est abandonné et l'exécution continuera au départ du COB suivant.

L'endroit où le COB a été abandonné, ainsi que l'état de l'ACCU sont mémorisés automatiquement. Quand le COB original est invoqué à nouveau, il continuera à l'adresse qui suit immédiatement celle où il a été abandonné. Lorsque cette technique est utilisée, le XOB 11 ne doit pas être programmé afin de ne pas perdre de temps alors qu'il ne s'agit pas vraiment d'une erreur.

**XOB 12: Dépassement de la capacité du registre d'index**

(Index Register overflow)

La taille du registre d'index est de 13 bits (0 à 8191), cette taille est suffisante pour référencer tous les éléments. Si un programme contient un élément indexé qui tombe en dehors de sa plage d'adresse, le XOB 12 est appelé. Par exemple, si le Flag indexé 8000 est adressé avec le registre d'index contenant 500: le flag 8500 est référencé, ce qui tombe en dehors de la plage d'adresse valide des Flags (0 à 8191).

**XOB 13: Flag erreur positionné**

(ERROR flag set)

Plusieurs instructions du PCD peuvent enclencher le flag Erreur (consultez le Guide des instructions). Si une erreur se produit: le flag Erreur est positionné et le XOB 13 est appelé pour que celle-ci puisse être traitée (alarme, message d'erreur sur une imprimante, ...). Le XOB 13 est toujours appelé lorsque le flag Error est mis, quelle qu'en soit la cause: calcul, transfert de données ou erreur de communication. Si le diagnostic doit être adapté en fonction du type de l'erreur, un PB (ou FB) doit être appelé conditionnellement après chaque instruction pouvant positionner le flag Erreur.

Exemple: ....

```

DIV  R 500 ; valeur 1
R 520      ; valeur 2
R 550      ; resultat
R 551      ; reste
CPB  E 73  ;Si erreur alors appel du PB 73

```

....

```

PB  73 ; Division par zéro
SET  O 99
INC  C 1591
EPB

```

Le PB 73 est appelé après une division par zéro et enclenche la sortie 99, indiquant une division par zéro. Le compteur 1501 compte le nombre de fois où cette erreur se produit. Un dépassement de capacité lors d'une multiplication, pourrait par exemple activer la sortie 98 et le compteur 1590 peut compter cet événement. Le XOB 13 peut également être programmé, mais laissé vide. S'il n'est pas programmé, la LED Error sur la face avant du CPU sera allumée lorsque le flag Erreur est positionné.

**IMPORTANT:**

Le flag Erreur, ainsi que les autres flags arithmétiques (Positif, Négatif, Zéro) sont positionnés par certaines instructions ou événements; s'ils sont d'un quelconque intérêt pour votre programme, ils doivent être traités immédiatement car ils réfèrent toujours à la dernière instruction pouvant affecter leur état. Par exemple, si une addition correcte suit une division par zéro, le flag Erreur sera remis à zéro.

**XOB 14, 15: XOBs d'interruption cyclique**

Les XOB 14 et 15 sont appelés périodiquement avec une fréquence de 10 ms. a 1,000 s. Cette fréquence peut être définie par une instruction SYSWR.

**XOB 16: Démarrage à froid**

Le XOB 16 est un bloc de démarrage à froid. Il est traité à la mise sous tension du PCD ou lors d'une commande de démarrage à froid effectuée depuis l'outil de programmation. Le XOB 16 utilisé pour initialiser toutes sortes d'informations avant de traiter le programme. Une fois le XOB 16 terminé, le programme traite les COB dans l'ordre croissant de leurs numéros sans jamais revenir dans le XOB 16.

Le XOB 16 ne peut pas être redémarré par le programme utilisateur. Si une action particulière doit pouvoir être exécutée par un COB ainsi que pendant l'initialisation, cette action doit être écrite dans un PB ou FB qui sera appelé aussi bien du XOB 16 que d'un COB.

**XOB 17, 18, 19: Demande d'interruption d'un XOB via S-Bus**

Ces trois XOBs peuvent être utilisés comme routine d'interruption. Leur traitement peut être démarré via le bus de communication S-Bus. Il est possible de démarrer leur traitement avec l'instruction SYSWR ou la Fbox Fupla *Spécial, execute XOB*

**XOB 20 et XOB 25: Détection de changement d'une entrée d'interruption**

Le XOB 20 (resp. 25) est appelé quand l'entrée d'interruption INB1 (resp. INB2) du PCD a détecté un flanc montant (voir manuel Hardware des PCD respectifs).

**XOB 30: Perte de liaison maître esclave avec les RIO**

La liaison est testée après chaque message envoyé par la station maître vers la station esclave. Si le test est négatif, le CPU maître appelle le XOB 30. C'est par exemple lorsqu'une station en ligne est déconnectée du réseau ou mise hors tension.



## 5.7 Blocs séquentiels (SB 0 à 31, 96 <sup>1</sup>)

Les blocs séquentiels SB sont constitués par une suite d'étapes et de transitions. Dans une étape, vous exécutez une partie du programme. Dans une transition, vous attendez la condition permettant de passer à l'étape suivante.

La création des SB fait appel à un éditeur particulier : l'éditeur Graftec. C'est un excellent outil de programmation réservé aux traitements séquentiels. Les fichiers correspondants se terminent par l'extension « \*.sfc ».

Les SB peuvent être appelés depuis n'importe quel autre type de bloc.

## 5.8 Tableau récapitulatif.

Service	Média	Opérande	Remarque
Blocs d'Organisation Cycliques	COB	0...15	Minimum 1 COB par programme
Bloc Programme	PB	0...299	Sous programmes appelé par un COB, PB,FB,SB ou XOB
Bloc de Fonction	FB	0...999	Fonction paramétrisable appelée par un COB, PB,FB,SB ou XOB
Bloc Séquentiel	SB	0...32 0...96 <sup>1</sup>	Sous programmes séquentiels appelé par un COB, PB ou FB ( SB, XOB)
Step	ST	0...1999 0...5999 <sup>1</sup>	
Transition	TR	0...1999 0...5999 <sup>1</sup>	

<sup>1</sup> PCD2.M170/480, PCD4. M170 et PCD3



## Table des matières

---

<b>6</b>	<b>PROGRAMMER EN GRAFTEC</b>	<b>3</b>
<b>6.1</b>	<b>Au sommaire de ce chapitre</b>	<b>3</b>
<b>6.2</b>	<b>Blocs séquentiels (SB 0 à 31 )</b>	<b>4</b>
<b>6.3</b>	<b>Blocs d'organisation cyclique</b>	<b>5</b>
6.3.1	Déroulement des programmes cycliques	5
6.3.2	Temps de cycle	5
<b>6.4</b>	<b>Préparer un projet Graftec</b>	<b>6</b>
6.4.1	Créer un nouveau projet	6
6.4.2	Créer un nouveau fichier Fupla ou IL	6
6.4.3	Appel du SB à partir d'un COB	7
6.4.4	Créer un nouveau fichier Graftec	7
<b>6.5</b>	<b>Gestion des SB</b>	<b>8</b>
6.5.1	<i>Block Navigator</i>	8
6.5.2	Structure générale d'un SB	9
6.5.3	Règles d'évolution	9
6.5.4	Transitions (TR 0 à 1999 )	10
6.5.5	Etapes (ST 0 à 1999 )	11
<b>6.6</b>	<b>Structure des blocs séquentiels</b>	<b>12</b>
6.6.1	Séquence simple	12
6.6.2	Divergence alternative (OU)	12
6.6.3	Divergence simultanée (ET)	12
6.6.4	Saut de séquence	12
6.6.5	Répétition de séquence	12
<b>6.7</b>	<b>Edition d'une structure séquentielle</b>	<b>13</b>
6.7.6	Edition d'une séquence simple	13
6.7.7	Editer une liaison	13
6.7.8	Edition d'une divergence alternative (OU)	14
6.7.9	Editer les liens de convergence.	14
6.7.10	Edition d'une divergence simultanée. (ET)	14
6.7.11	Editer les liens de convergence.	14
6.7.12	Ajouter un commentaire	14
6.7.13	Insérer une séquence	15
6.7.14	Effacer une séquence	15
6.7.15	Copier-coller de séquence	16
<b>6.8</b>	<b>Ecriture d'un premier bloc séquentiel</b>	<b>17</b>
6.8.1	Ouverture du fichier	17
6.8.2	Réalisation de la structure de base	17
6.8.3	Choix de l'éditeur IL ou Fupla	18
6.8.4	Edition des symboles	19
6.8.5	Rédaction du programme	19
6.8.6	Programmation d'une transition	19

6.8.7	Utilisation de temporisateurs dans un SB	20
6.8.8	Attendre la fin de temporisation.	20
6.8.9	Répétition de la séquence pour la pause du clignoteur	21
6.8.10	Décrémentation d'un compteur	21
6.8.11	Divergence alternative	22
<b>6.9</b>	<b>Construction et débogage du programme (Build)</b>	<b>23</b>
6.9.1	Fenêtre des messages	23
6.9.2	Outils en ligne	23
<b>6.10</b>	<b>Structuration du graftec en pages</b>	<b>24</b>
6.10.1	Définition d'une page	24
6.10.2	Edition de pages	25

## **6 Programmer en Graftec**

---

### **6.1 Au sommaire de ce chapitre**

L'éditeur Graftec, est un éditeur prévu pour réaliser des programmes séquentiels à l'aide des langages Fupla ou IL. Ce chapitre introduit l'usage du Graftec pour réaliser ces programmes à l'aide de Blocs Séquentiel SB, d'étapes et transitions.

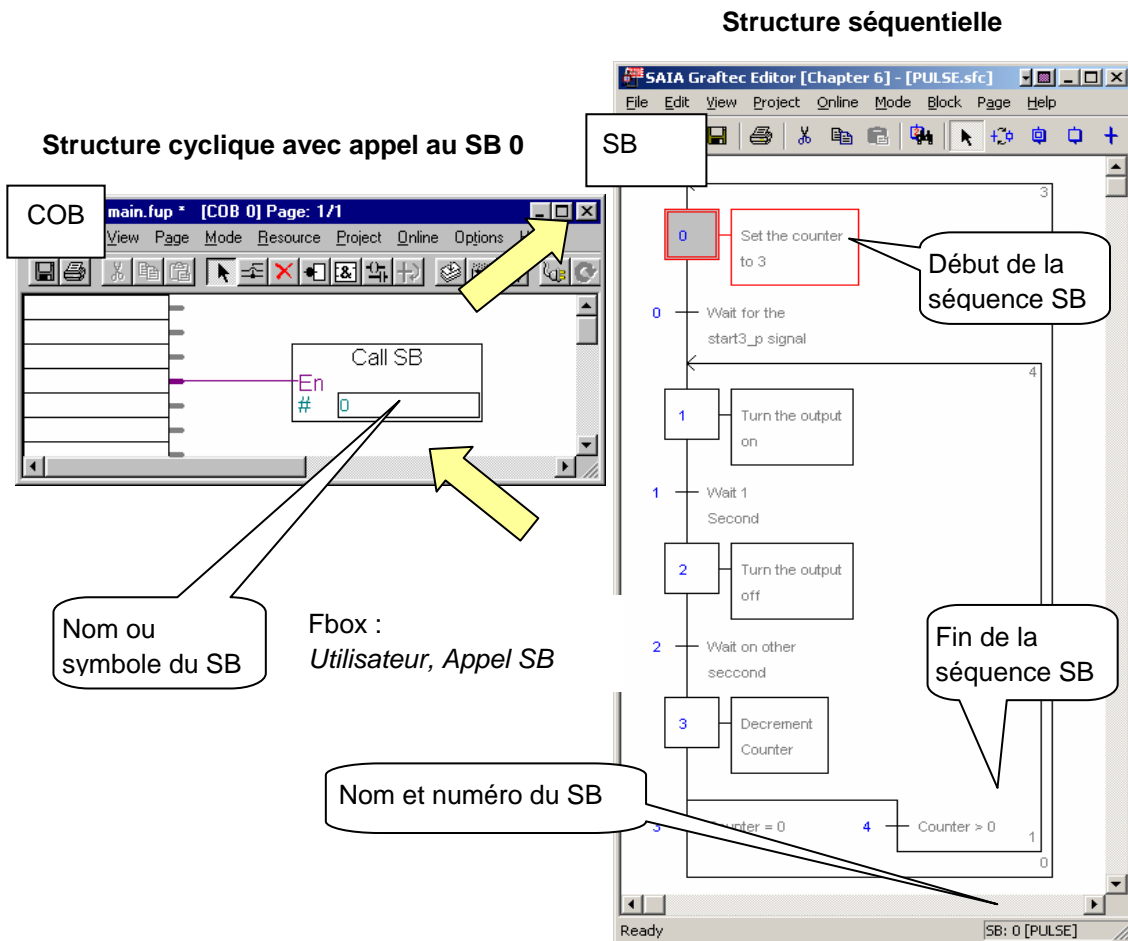
## 6.2 Blocs séquentiels (SB 0 à 31 <sup>1</sup>)

Dans les pages qui suivent nous proposons une technique de structuration différente et particulièrement efficace pour élaborer les programmes séquentiels où nous avons des attentes sur des événements programmés ou externes à l'automate.

En raison des attentes dont les durées sont indéterminées, nous ne pouvons pas estimer le temps de cycle des programmes séquentiels. Il est donc indispensable de bien séparer les programmes cycliques des programmes séquentiels.

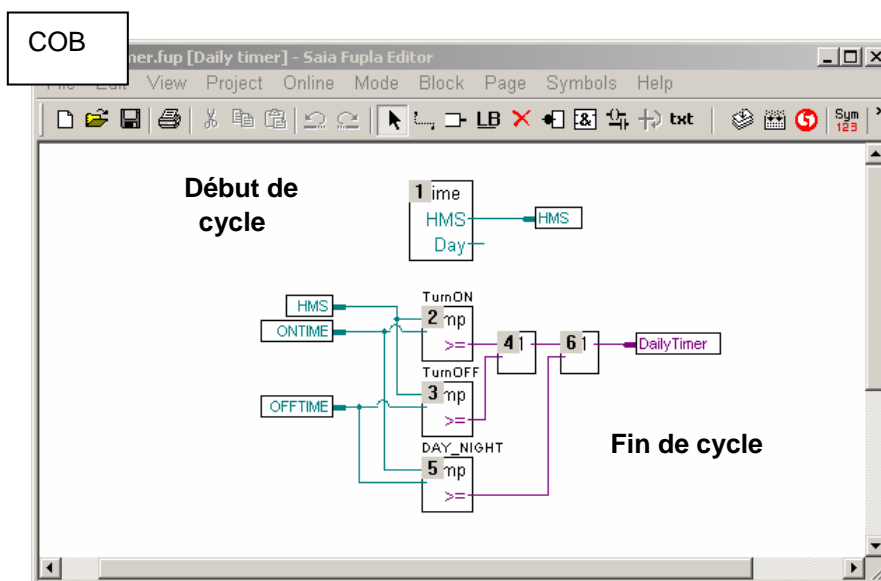
Une attente sur un événement séquentiel ne doit en aucun cas bloquer l'exécution permanente des programmes cycliques. Pour satisfaire à cette condition, les programmes séquentiels sont placés à l'intérieur de l'une des structures SB disponibles que l'on appelle à chaque cycle de programme.

Les SB ont une particularité, lorsque le programme séquentiel placé dans le SB est en attente sur un événement, le PCD met de côté le SB pour traiter les programmes cycliques. La suite du SB sera alors traitée au prochain cycle de programme.



<sup>1</sup> Les nouveaux PCD2.M170/480, PCD3 et PCD4.M170 supportent jusqu'à 96 SB

### 6.3 Blocs d'organisation cyclique



#### 6.3.1 Déroulement des programmes cycliques

Jusque là les programmes présentés étaient entièrement cycliques. Ils étaient réalisés à partir d'une liste de fonctions graphiques ou d'instructions que l'automate traite les une après les autres et aussi rapidement que possible depuis le début jusqu'à la fin du programme puis il reprend le travail au début pour un nouveau cycle.

#### 6.3.2 Temps de cycle

Le temps nécessaire pour traiter un cycle de programme est déterminé et correspond à la somme des temps d'exécution de chaque instruction ou fonction graphique. C'est ce que nous appelons le temps de cycle du programme.

Le temps de cycle est généralement de quelques millisecondes. Si une information sur une entrée digitale change d'état, l'adaptation des sorties digitales dépendantes de cette entrée sont mises à jour avec un délai presque instantané, le retard est en fait équivalent au temps de cycle.

Les programmes cycliques appartiennent à des structures de type COB, PB, FB ou XOB.

## 6.4 Préparer un projet Graftec

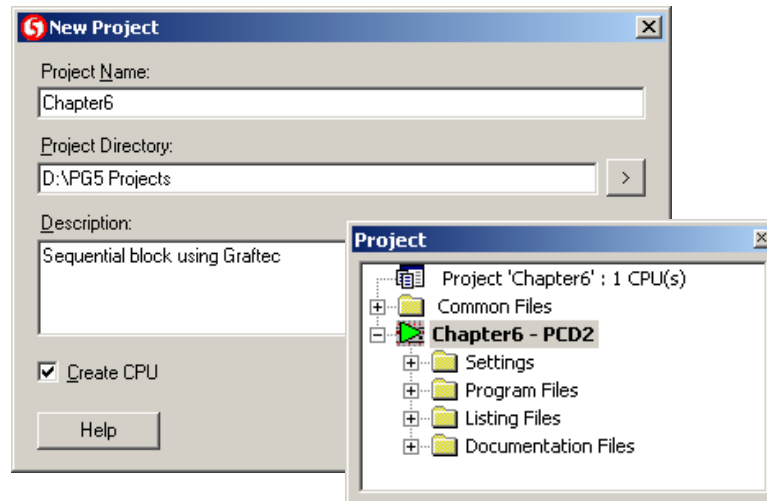
Pour réaliser un exemple, nous conseillons de préparer un nouveau projet dans lequel nous préparons les fichiers pour éditer les programmes Graftec:

Pour une programmation graphique, préparer un fichier Fupla et un fichier Graftec

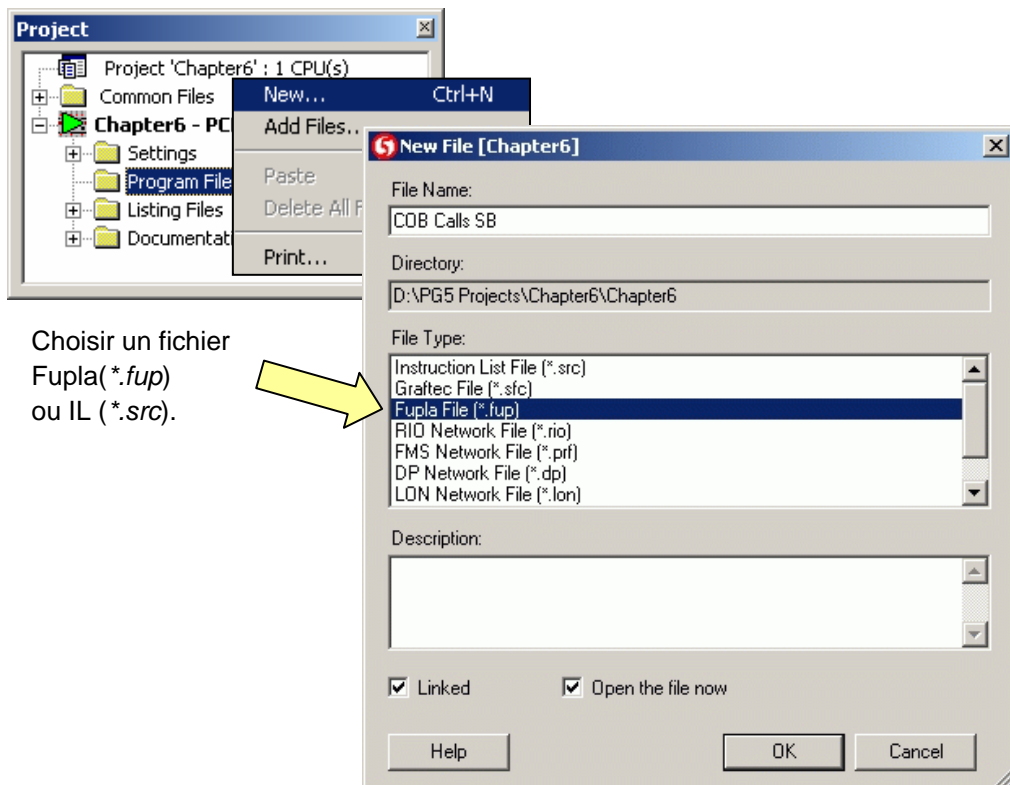
Pour une programmation en liste d'instruction, préparer un fichier SRC et un fichier Graftec

### 6.4.1 Créer un nouveau projet

Dans la fenêtre *SAIA Project Manager*, sélectionner le menu *Project, New...* et créer le nouveau projet.



### 6.4.2 Créer un nouveau fichier Fupla ou IL





### 6.4.3 Appel du SB à partir d'un COB

Selon votre choix de programmation (SRC ou Fupla), vous pouvez appeler le SB à l'aide d'une instruction CSB ou d'une Fbox Call SB. Le résultat est le même. Ouvrez votre nouveau fichier et écrivez votre programme comme indiqué ci-dessous:

**Programme IL:**

```
COB 1 ; Début du COB
0
CSB 0 ; Appel de SB 0
ECOB ; Fin du COB
```

**Programme Fupla:**



Fbox: *Utilisateur, Appel SB*

### 6.4.4 Créer un nouveau fichier Graftec

Choisir un fichier Graftec (\*.sfc)

## 6.5 Gestion des SB

### 6.5.1 Block Navigator

A la création d'un fichier Graftec, l'éditeur crée un bloc séquentiel SB 0 et une étape initiale. La sélection du bouton *Block Navigator* affiche la liste de tous les SB et des pages qui constituent le fichier.

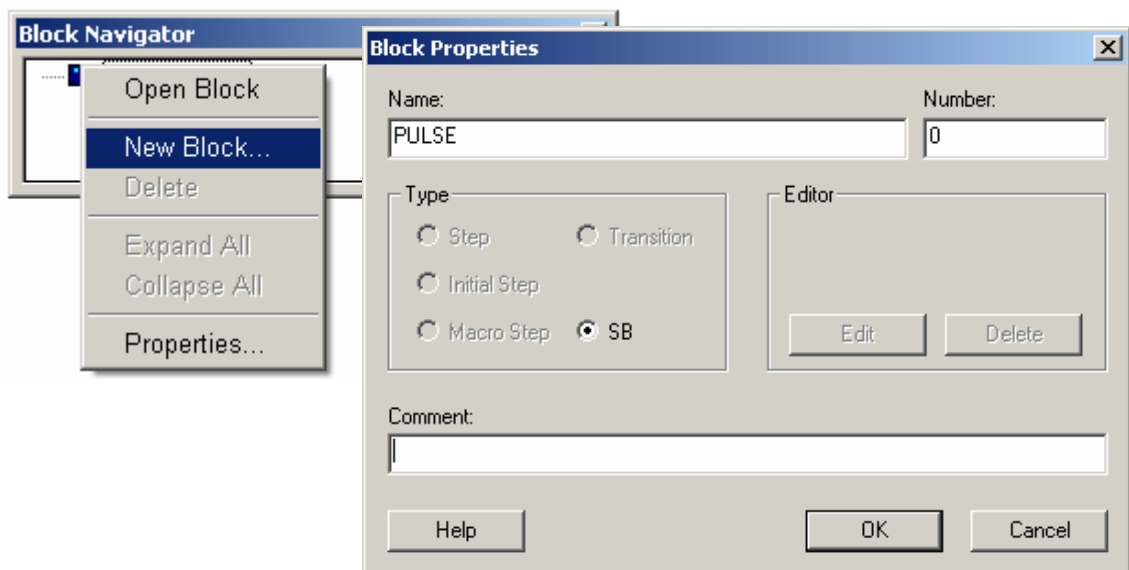
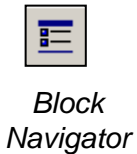
Pour ajouter un nouveau bloc SB dans le fichier, positionner la souris sur le *Block Navigator*, presser le bouton droite de la souris et sélectionner le menu de contexte *New Block*.

Le champ *Number* représente le numéro du SB. (Ne pas confondre le numéro du SB avec les numéros des éléments étapes et transitions. Ce numéro comprend toute la chaîne d'étapes et transition du SB représenté par le symbole dans le champ *Name*. Nous vous conseillons de nommer chaque SB avec un symbole, cela permet de faciliter la navigation dans les blocs

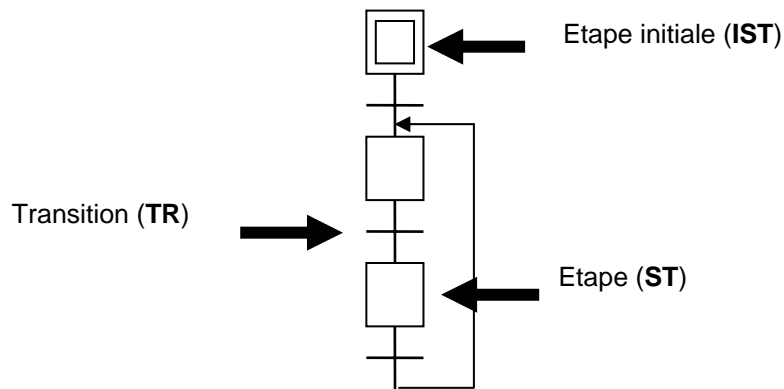
Pour afficher la structure Graftec de l'un des SB présent dans le *Bloc Navigator*, positionner la souris au dessus du bloc et sélectionner le menu de contexte *Open Block*

Le menu de contexte *Properties* permet de modifier le nom et le numéro du bloc sélectionné dans le *Block Navigator*.

Maintenant nous pouvons structurer le SB avec les étapes et transitions.



### 6.5.2 Structure générale d'un SB



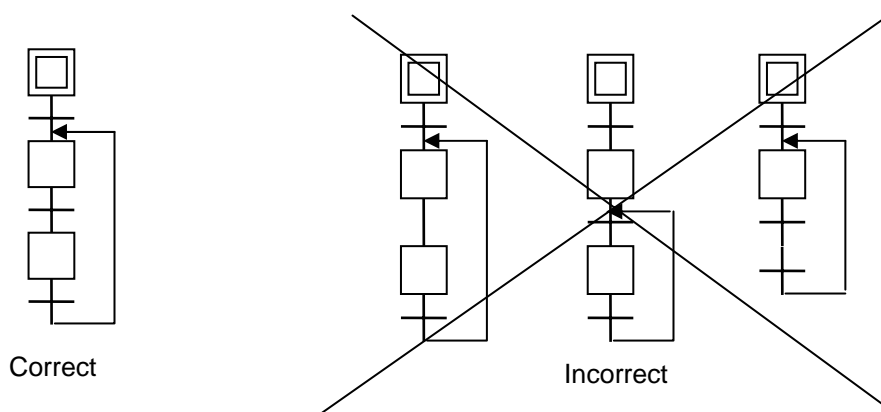
L'éditeur Graftec, permet de préparer la structure des blocs séquentiels à l'aide des étapes et transitions dans lesquelles l'utilisateur dépose les programmes sous la forme de fonctions graphiques ou de listes d'instructions.

Un bloc séquentiel SB commence généralement par une étape initiale symbolisée par un double carré. Elle représente le début du programme. C'est de cet endroit que le programme va démarrer lors du premier appel au bloc SB. (démarrage à froid)

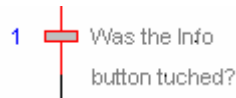
### 6.5.3 Règles d'évolution

L'édition d'un SB répond à une syntaxe stricte.

Un bloc commence toujours par une étape initiale, puis il alterne systématiquement transitions et étapes. Ainsi, deux étapes ou deux transitions ne doivent jamais être reliées directement.



## 6.5.4 Transitions (TR 0 à 1999 <sup>1</sup>)



Une transition contient les parties de programme devant être traitées jusqu'à l'apparition d'un événement donné:

attendre la réception d'un caractère sur le port série

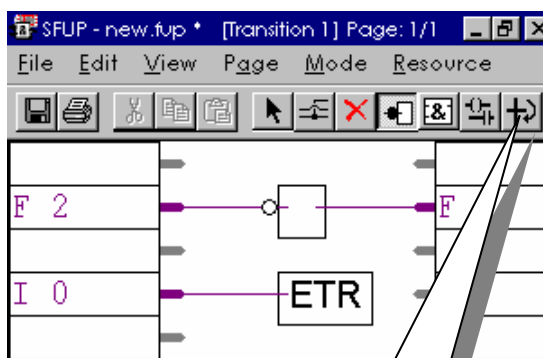
attendre la fin d'une temporisation

attendre sur un fin de course

La transition finit toujours par une instruction *ETR*, elle se répète à l'infini tant qu'*ETR* est inactive.

**Exemple:** inverser l'indicateur F2 à chaque cycle de programme jusqu'à ce que l'entrée 0 passe à l'état haut.

Programme Fupla



Programme IL

```

; Transition toggles Flag2 until
; Input 0 comes to high

COM F 2
STH I 0

```

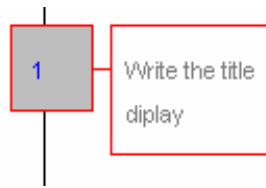
Il n'est pas nécessaire de définir un programme pour chaque transition. Une transition sans programme est toujours vraie (*true*) et ignorée.

Pour les transitions en liste d'instructions, l'accumulateur est toujours à l'état haut au début d'une transition ou d'une étape.

Vous pouvez éditer un maximum de 32 <sup>1</sup> blocs séquentiels, à l'aide de 2 000 <sup>1</sup> étapes et transitions.

<sup>1</sup> Les nouveaux PCD2.M170/480, PCD3 et PCD4.M170 supportent jusqu'à 6000 ST/TR et 96 SB

### 6.5.5 Etapes (ST 0 à 1999 <sup>1</sup>)

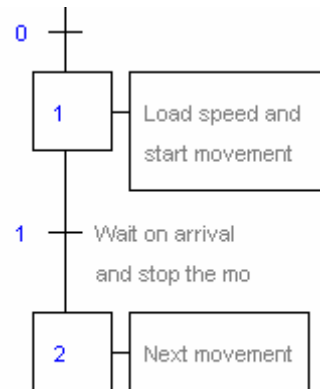


Les étapes contiennent normalement des parties de programme destinées à agir sur l'application à automatiser

Supposons que vous voulez effectuer un déplacement de A en B.

D'ordinaire, vous commencez par régler la vitesse et le sens de déplacement, puis vous lancez le mouvement. Ces deux tâches, non répétitives, peuvent figurer dans une étape, elles sont exécutées qu'une seule fois.

Vous devez ensuite surveiller le déplacement et arrêter le moteur dès qu'il atteint la destination B. Ce contrôle (qui consiste, par ex. à lire le fin de course) doit se reproduire inlassablement jusqu'à ce que le moteur parvienne à destination: voilà une situation parfaitement traitée dans une transition qui est, par définition, cyclique.



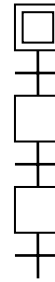
Une étape sans programme passe directement à la transition suivante. Une étape n'est traitée qu'une seule fois, elle n'est pas cyclique !

<sup>1</sup> Les nouveaux PCD2.M170/480, PCD3 et PCD4.M170 supportent jusqu'à 6000 ST/TR et 96 SB

## 6.6 Structure des blocs séquentiels

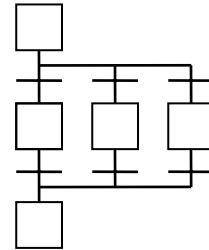
### 6.6.1 Séquence simple

Se compose d'une alternance d'étapes et transitions.  
Rappelons qu'il ne peut y avoir deux étapes ni deux transitions à la suite l'une de l'autre.



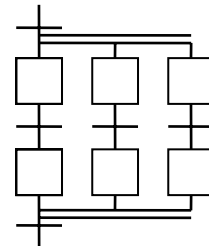
### 6.6.2 Divergence alternative (OU)

Une divergence alternative et un choix d'une séquence parmi plusieurs possibilités, le test des transitions s'effectue toujours de gauche à droite, la première transition dont la condition est vraie détermine la séquence à traiter. Une divergence alternative commence toujours par une étape qui va vers plusieurs transitions et se termine par la structure inverse, plusieurs transitions qui vont vers une seule étape. L'éditeur Graftec supporte maximum 32 divergences. Au-delà de 32 divergences, le PCD fait appel au XOB 9.



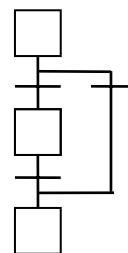
### 6.6.3 Divergence simultanée (ET)

Une divergence simultanée est constituée de plusieurs séquences qui seront traitées simultanément. Une divergence simultanée commence toujours par une transition qui va vers plusieurs étapes et se termine par la structure inverse, plusieurs étapes qui vont vers une seule transition de synchronisation. L'éditeur Graftec supporte maximum 32 divergences. Au-delà de 32 divergences, le PCD fait appel au XOB 9.



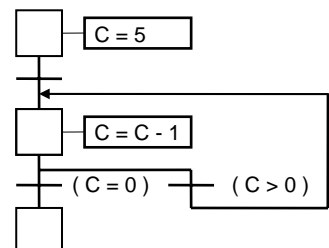
### 6.6.4 Saut de séquence

Le saut de séquence est réalisé à partir d'une divergence alternative et permet de traiter une séquence de manière conditionnelle.



### 6.6.5 Répétition de séquence

La répétition de séquence est aussi réalisée à partir d'une divergence alternative. Un compteur est initialisé avec le nombre de boucles de programmes, puis l'on entre dans une séquence simple de longueur quelconque, l'on décrémente le compteur et si le compteur est non nul alors on répète la boucle.

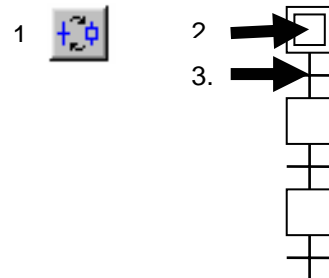


## 6.7 Edition d'une structure séquentielle

L'ouverture d'un nouveau fichier Graftec affiche l'étape initiale. L'exécution d'une étape commence toujours ainsi. Vous pouvez ajouter de nouveaux éléments au schéma, soit au clavier, soit à l'aide de la barre d'outils.

### 6.7.6 Edition d'une séquence simple

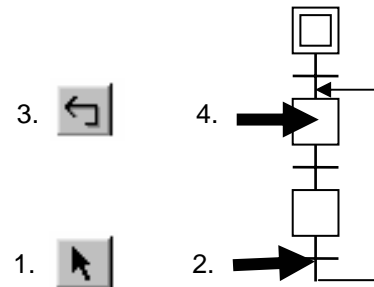
1. Sélectionnez le bouton *Mode mixte* de la barre d'outils.
2. Placez la souris sur l'étape initiale et cliquez.
3. Déplacez la souris sur la nouvelle transition et cliquez.
4. Poursuivez de la même manière



### 6.7.7 Editer une liaison

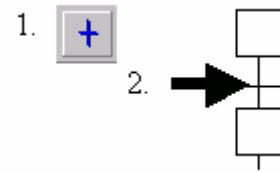
La séquence terminée, le programme l'est aussi. Pour le relancer, il faut ajouter une boucle. Rappel: il est impossible de relier deux étapes ou deux transitions, une boucle commence toujours sur une transition pour aller vers une étape.

1. Cliquez sur le bouton *Sélect mode* de la barre d'outils.
2. Pointez la transition de départ.
3. Cliquez sur le bouton *Link mode* de la barre d'outils.
4. Cliquez sur l'étape vers laquelle vous voulez lier la transition.



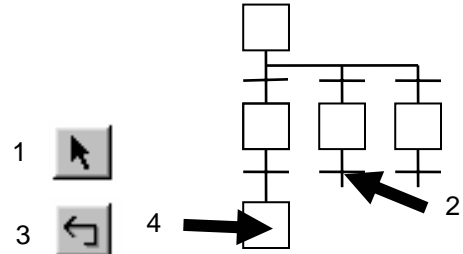
### 6.7.8 Edition d'une divergence alternative (OU)

1. Cliquez sur le bouton *Transition mode* de la barre d'outils.
2. Cliquez sur une transition déjà suivie d'une étape.
3. A chaque clic de souris, une autre transition s'ajoute sur la droite.



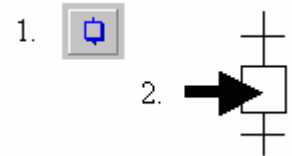
### 6.7.9 Editer les liens de convergence.

1. Cliquez sur le bouton *Select mode* de la barre d'outils.
2. Pointez la transition à fermer.
3. Cliquez sur le bouton *Link mode* de la barre d'outils.
4. Cliquez sur l'étape à raccorder.



### 6.7.10 Edition d'une divergence simultanée. (ET)

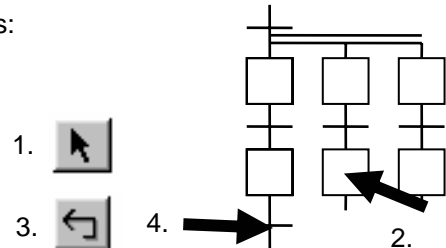
1. Cliquez sur le bouton *Step mode* de la barre d'outils.
2. Cliquez sur une étape déjà suivie d'une transition.
3. A chaque clic de souris, une autre étape s'ajoute sur la droite.



### 6.7.11 Editer les liens de convergence.

Pour synchroniser vos tâches simultanées:

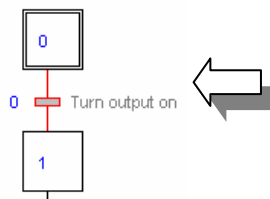
1. Cliquez sur le bouton *Select mode* de la barre d'outils.
2. Pointez l'étape à fermer.
3. Cliquez sur le bouton *Link mode* de la barre d'outils.
4. Cliquez sur la transition à raccorder.



### 6.7.12 Ajouter un commentaire

1. Cliquez sur le bouton *Select mode* de la barre d'outils.
2. Cliquez sur un élément avec le bouton droit de la souris, choisissez l'option *Properties...*
3. Introduire votre commentaire dans *le champ Comment*.

Remarque : pour éditer le commentaire sur deux lignes, insérer la commande \n .



Edit Code	Enter
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Delete	Del
Properties...	Alt+Enter

**Element Properties**

Name: PULSE\_ST\_1      Number: [ ]

Type:  Step     Transition  
 Initial Step  
 Macro Step     SB

Editor: Function Block Diagram  
   

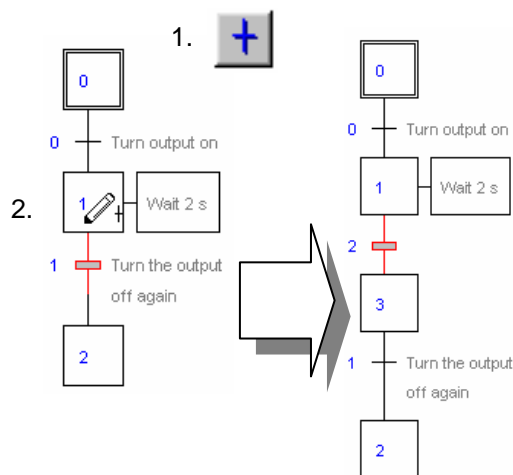
Comment:  
 Turn the output\non



### 6.7.13 Insérer une séquence

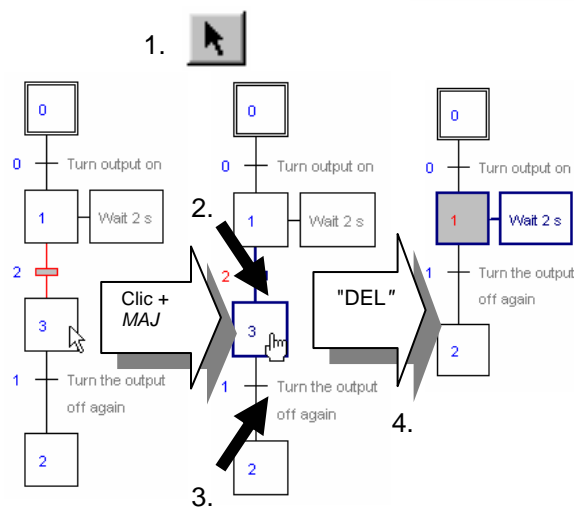
1. Cliquez sur le bouton *Transition mode* de la barre d'outils.
2. Cliquez sur une étape déjà suivie d'une transition.

L'éditeur ajoute une nouvelle étape et transition.



### 6.7.14 Effacer une séquence

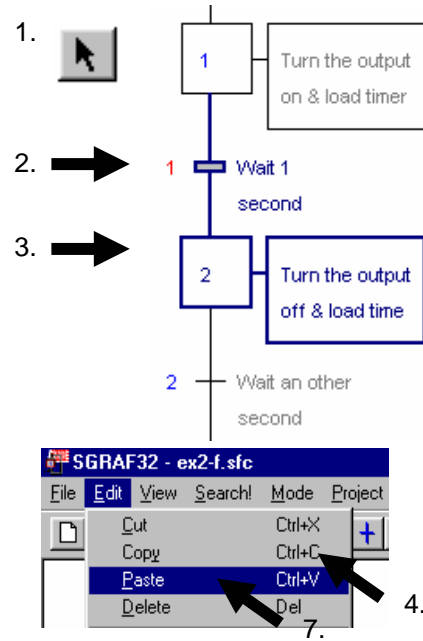
1. Cliquez sur le bouton *select mode* de la barre d'outils.
2. Cliquez sur la première transition de la séquence.
3. Cliquez sur la dernière étape de la séquence que vous voulez supprimer tout en maintenant enfoncée la touche *Shift*
4. Appuyez sur la touche *Del*.



### 6.7.15 Copier-coller de séquence

#### Copier une séquence:

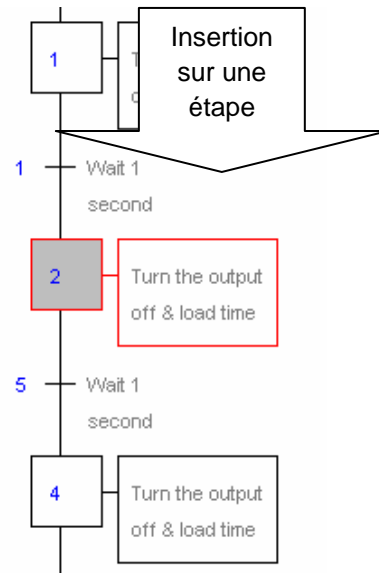
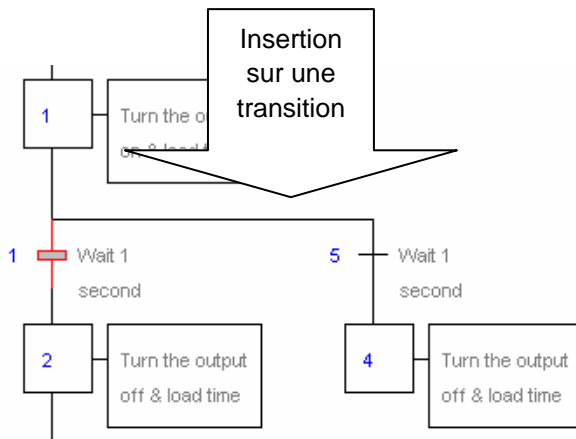
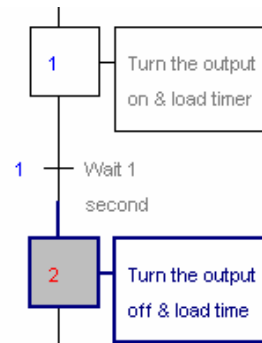
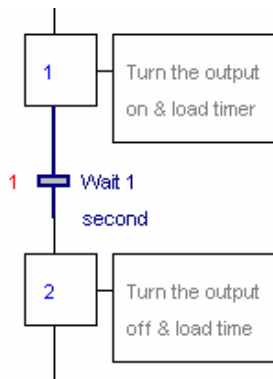
1. Cliquez sur le bouton *Select mode* de la barre d'outils.
2. Pointez le début de la séquence.
3. Cliquez sur la dernière étape de la séquence tout en maintenant enfoncée la touche *Shift*.
4. Déroulez le menu *Edit* et choisissez *Copy*.
5. Cliquez de nouveau sur le bouton *Select mode*
6. Cliquez sur l'endroit où vous voulez insérer la séquence.
7. Déroulez le menu *Edit* et choisissez l'option *Paste*.



#### Coller une séquence:

#### Remarque:

Selon la position de l'élément (transition ou étape) à insérer, la séquence peut être ajoutée sous ou à côté de l'élément sélectionné.



## 6.8 Ecriture d'un premier bloc séquentiel

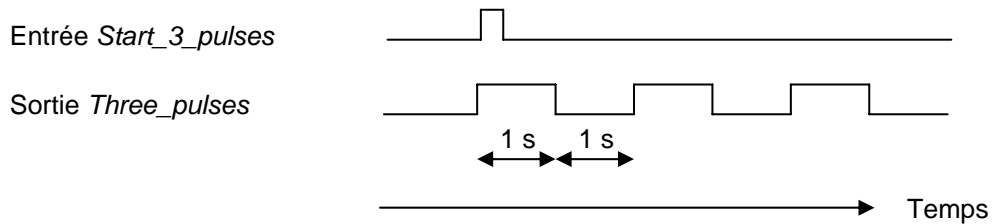
### 6.8.1 Ouverture du fichier

Ouvrez le fichier *PULSE.sfc*. Passez dans la liste des SB et chargez le SB intitulé *PULSE*.

Objectif:

Nous réaliserons un programme destiné à faire clignoter la sortie O 33 (*Three\_pulses*) trois fois d'affilée, sur chaque activation de l'entrée I 2 (*Start\_3\_pulses*).

Diagramme temporel



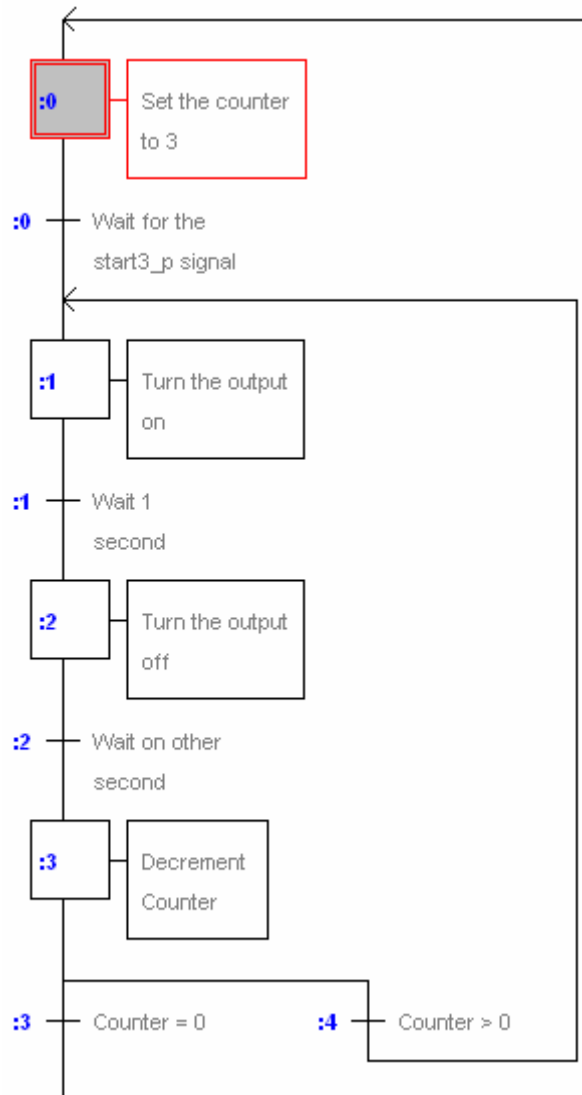
### 6.8.2 Réalisation de la structure de base

Nous commençons toujours par l'étape initiale qui sert de point de départ lors du démarrage à froid. Après quoi, nous pouvons attendre le signal de départ *Start\_3\_pulses*. Editez vos éléments comme ci-dessous et ajoutez-leur un commentaire:

The screenshot shows a portion of a ladder logic diagram on the left and an 'Element Properties' dialog box on the right. In the diagram, a transition element (a red bar with a white circle) is connected to a step element (a white box with ':0' inside). The step element is labeled 'Set the counter to 3'. The transition element is labeled 'Wait for the start3\_p signal'. The dialog box has the following fields and options:

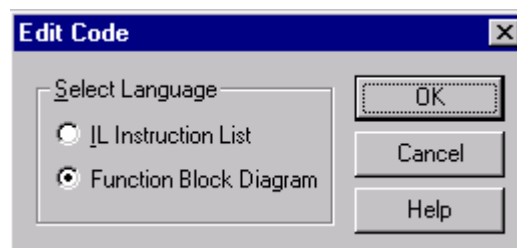
- Name: PULSE\_TR\_0
- Number: (empty)
- Type:
  - Step (radio button)
  - Transition (radio button, selected)
  - Initial Step (radio button)
  - Macro Step (radio button)
  - SB (radio button)
- Editor: Function Block Diagram
- Buttons: Edit, Delete
- Comment: Wait for the \nstart3\_p signal
- Buttons: Help, OK, Cancel

Au démarrage de la séquence, forcer la sortie *Three\_pulses* à l'état haut pendant 1 seconde, puis forcez la à l'état bas pour 1 seconde. répétez l'opération trois fois, puis relancez la séquence.



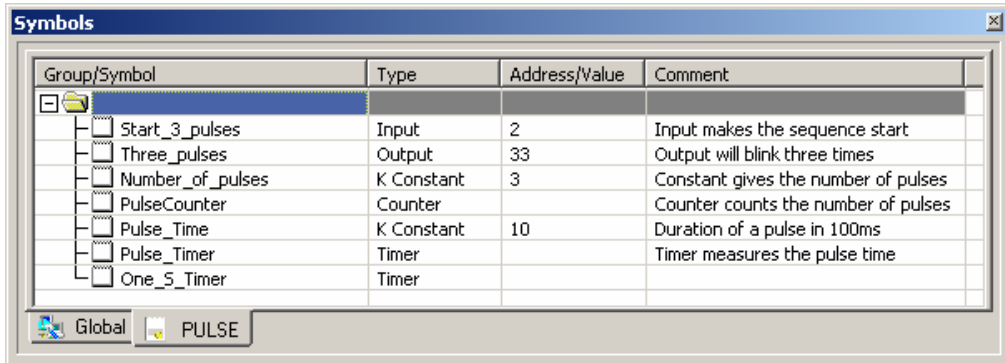
### 6.8.3 Choix de l'éditeur IL ou Fupla

Notre structure est achevée. Il reste à rédiger le programme de chaque étape et transition avec l'éditeur liste d'instructions (*IL Instruction List*) ou *Fupla (Function Block Diagram)*, à votre convenance. Pour notre exemple, nous programmerons l'étape initiale en Fupla. Double-cliquez sur l'étape initiale et choisissez l'éditeur Fupla:



### 6.8.4 Edition des symboles

Commençons par dresser, dans l'éditeur de symboles, la liste de tous les éléments utilisés, comme illustré dans la fenêtre ci-dessous.



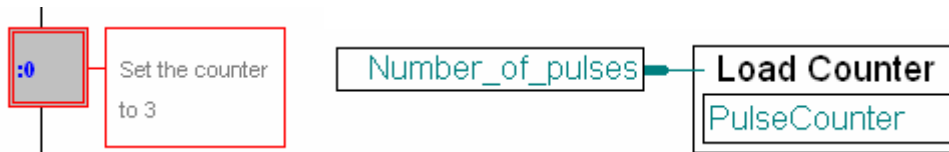
### 6.8.5 Rédaction du programme

Chargez le compteur *PulseCounter* avec la constante *Number\_of\_pulses* de valeur 3.

Programme Fupla:

Utilisez la Fbox: Graftec, Charge compteur

Attention: n'utilisez pas les temporisateurs ou compteurs d'autres familles de fonctions, qui sont réservés à des programmes cycliques.



Programme IL:

```
LD PulseCounter ;Counter initialisation
Number_of_pulses
```

### 6.8.6 Programmation d'une transition

Une transition boucle indéfiniment sur elle-même jusqu'à l'activation de la fin de transition *ETR* (programme Fupla) ou de l'accumulateur (programme IL). Dans la transition n° 0, nous attendrons que l'entrée *Start\_3\_pulses* passe à l'état haut. Ouvrez cette transition pour ajouter le programme suivant:

Programme Fupla:



Ajouter la FBox: Graftec, Fin TR

Programme IL:

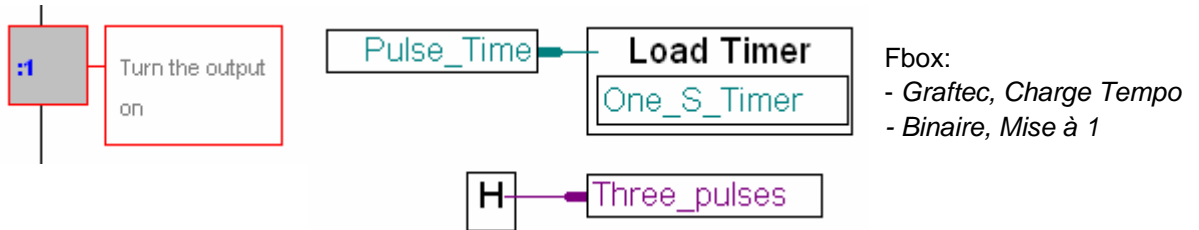
```
STH Start_3_pulses ; Copy the information Start_3_pulses into the accu
```

### 6.8.7 Utilisation de temporisateurs dans un SB

Procédez comme suit: activez la sortie et chargez le temporisateur dans l'étape, puis passez à la transition d'attente et d'interrogation du temporisateur jusqu'à échéance de la temporisation (temporisateur = 0).

**Programme Fupla:**

Les temporisateurs et compteurs de la bibliothèque de base du Fupla ne sont pas destinés aux SB. Ils concernent essentiellement les COB qui sont exécutés de façon cyclique et répétitive. Néanmoins, si vous voulez les intégrer à un SB, limitez-vous aux temporisateurs/compteurs de la famille de fonctions Graftec spécialement conçus à cet effet. Ils peuvent être chargés dans une étape, et leur état interrogé ultérieurement dans une autre étape ou transition.



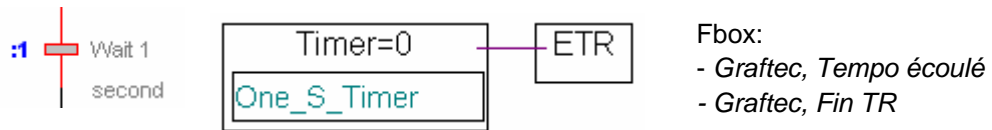
**Programme IL:**

```

SET      Three_pulses      ;Active la sortie
LD       One_S_Timer       ;Charge la temporisation
                Pulse_Time
    
```

### 6.8.8 Attendre la fin de temporisation.

**Programme Fupla:**



**Programme IL:**

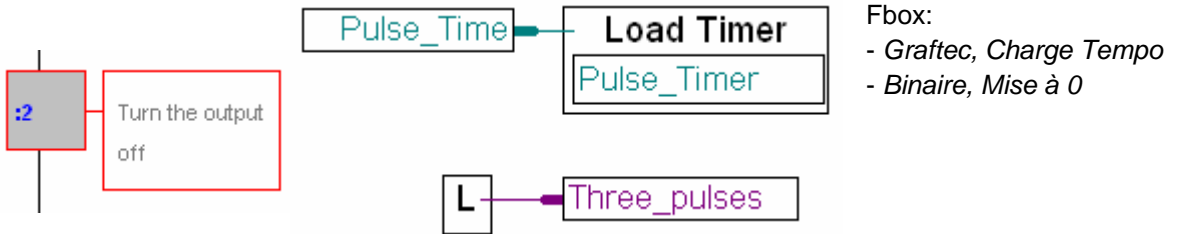
```

STL      One_S_Timer       ;Active l'accumulateur lorsque la temporisation
                                ;est terminée
    
```

### 6.8.9 Répétition de la séquence pour la pause du clignoteur

L'étape et la transition n° 2 sont en tous points identiques à l'étape et à la transition n° 1 sauf que la sortie *Three\_pulses* est forcée à l'état bas et qu'un autre temporisateur est utilisé.

**Programme Fupla:**



Fbox:  
 - *Graftec, Charge Tempo*  
 - *Binaire, Mise à 0*

**Programme IL:**

```
RES Three_pulses ;Active la sortie
LD Pulse_Timer ;Charge la temporisation
Pulse_Time
```

**TR:**

**Programme Fupla:**



**Programme IL:**

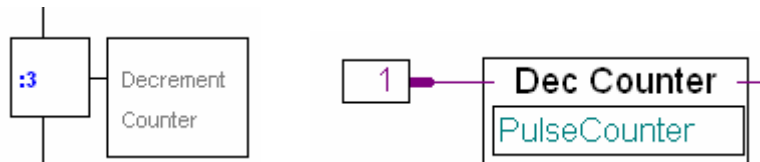
```
STL Pulse_Timer ;Active l'accumulateur lorsque la temporisation
;est terminée
```

**Remarque:** pour le ST/TR 1 et 2 nous avons fait usage de deux temporisateurs différents (*One\_S\_Timer* et *Pulse\_Timer*), par souci d'économie des adresses de temporisateur à disposition, nous aurions très bien pu utiliser deux fois le même temporisateur (*One\_S\_Timer* ou *Pulse\_Timer*) car ils ne sont pas utilisés simultanément mais l'un après l'autre !

### 6.8.10 Décrémentement d'un compteur

**Programme Fupla:**

Le compteur est décrémenté à chaque passage du programme, mais seulement si l'entrée binaire est à l'état haut (Voir l'aide en ligne sur la Fbox).



Fbox: *Graftec, Compteur -1*

**Programme IL:**

```
DEC PulseCounter ;Si l'accumulateur est à 1, décrémente le compteur
```

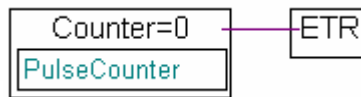
(Remarque:l'accu est toujours à 1 au début d'un ST/TR)

### 6.8.11 Divergence alternative

Les deux dernières transitions sont présentes pour effectuer un choix:

**Programme Fupla:**

:3 | Counter = 0

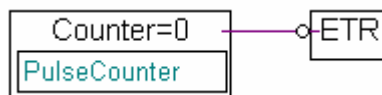


Fbox: Graftec, Compteur = 0

**Programme IL:**

STL PulseCounter

:4 | Counter > 0



STH PulseCounter

Transition 3: l'entrée ETR est active si le compteur est à zéro

Transition 4: l'entrée ETR est active si le compteur n'est pas à zéro



*Invert Binary  
connector*

Placer une inversion à l'entrée de la fonction *ETR* avec le bouton *invert binary connector*.



## 6.9 Construction et débbugage du programme (Build)



Une fois le programme terminé, sélectionner le bouton *Build* de la barre d'outils pour compiler la totalité du programme.

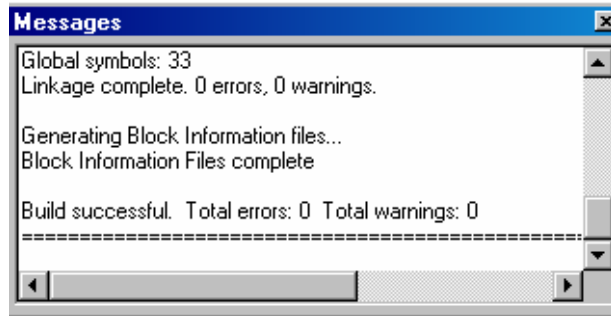
*Build All*

### 6.9.1 Fenêtre des messages

La fenêtre des *messages* vous renseigne sur le déroulement de la compilation.

Si le programme est correctement saisi, la dernière ligne de la fenêtre indique:

Build successful. Total errors: 0  
Total warnings: 0



Les erreurs éventuelles sont signalées en rouge. Double-cliquez sur le message correspondant pour accéder à la ligne de programme concernée.

### 6.9.2 Outils en ligne

Il reste à télécharger le programme et à passer en mode en ligne.



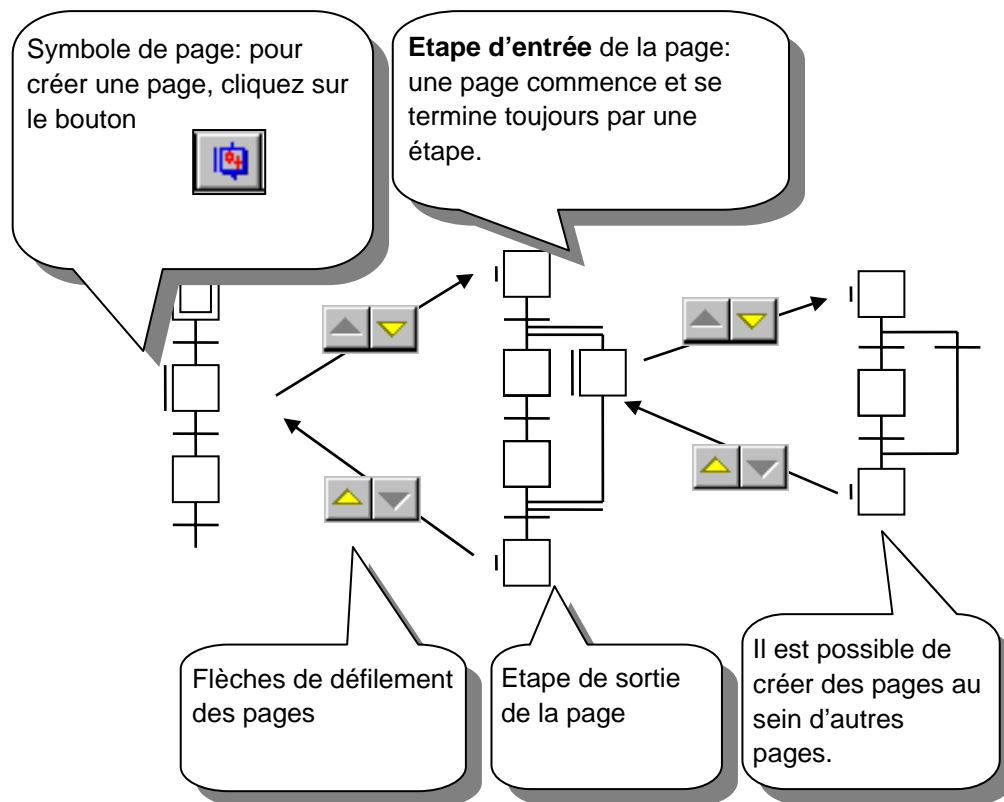
*Download Program*

Le bloc séquentiel peut être dynamiquement visualisé, un point rouge signale la transition ou l'étape active.

## 6.10 Structuration du graftec en pages

### 6.10.1 Définition d'une page

Dans le PG5, vous pouvez structurer vos gros programmes en de multiples pages s'enchaînant les unes aux autres.



#### Règles à respecter :

Une page commence se termine toujours par une étape.

Une page ne peut avoir qu'une seule étape d'entrée et qu'une seule étape de sortie.

Vous pouvez créer plusieurs niveaux d'imbrication des pages

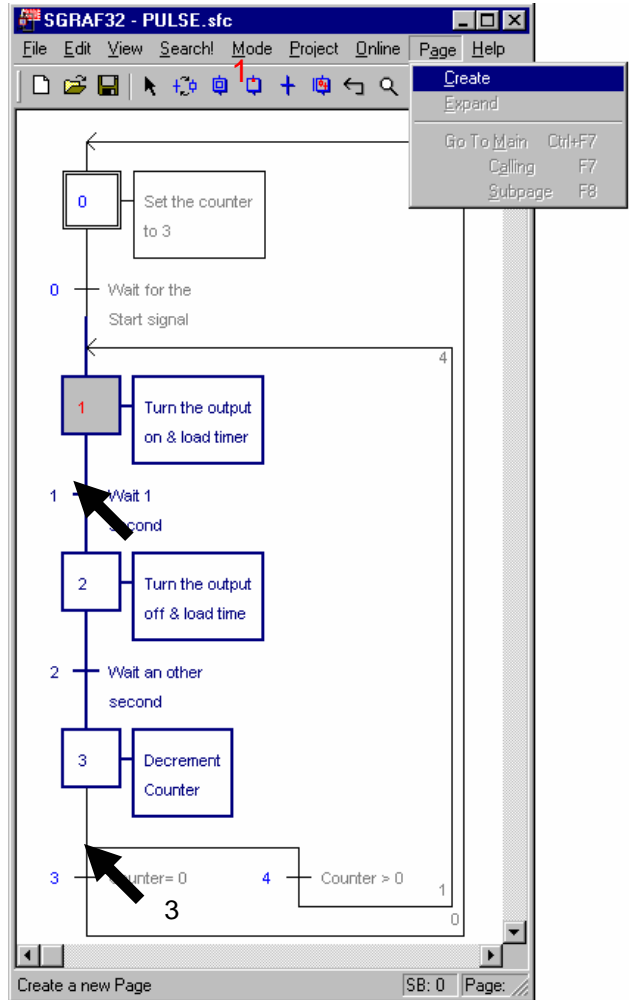
Il est impossible de déplacer ou de supprimer une étape d'entrée ou de sortie.

### 6.10.2 Edition de pages

Créer une page:

Permet de créer une page à partir d'une séquence de ST/TR.

1. Cliquez sur le bouton *Sélect mode* de la barre d'outils.
2. Cliquez sur la première étape de la séquence.
3. Enfoncez la touche *Shift* et cliquez sur la dernière étape de la séquence.
4. Sélectionner le menu *Page, Create*.



#### Ouverture de page:

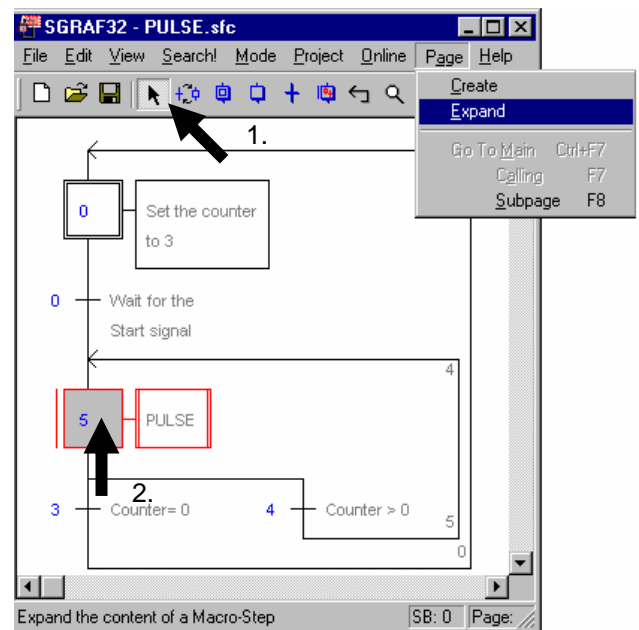
Affiche le contenu représenté par le symbole de page

1. Cliquez sur le bouton *Sélect mode* de la barre d'outils.
2. Cliquez sur la page de votre choix.
3. Déroulez le menu *Page* et choisissez *Calling* ou *Subpage*

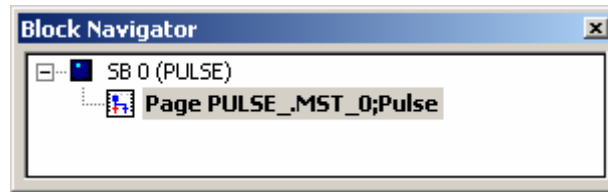
#### Suppression des pages:

Remplace le symbole de la page par toute la séquence.

1. Cliquez sur le bouton *Sélection de* de la barre d'outils.
2. Cliquez sur la page de votre choix.
3. Déroulez le menu *Page, Expand*.



Remarque: le *Block Navigator* facilite la navigation entre les pages qui composent le SB



## Table des matières

---

<b>7</b>	<b>PROGRAMMER EN IL (INSTRUCTION LIST)</b>	<b>3</b>
7.1	Au sommaire de ce chapitre	3
7.2	Préparer un projet IL	4
7.2.1	Créer un nouveau projet	4
7.2.2	Créer un nouveau fichier IL	4
7.3	Organisation d'une fenêtre d'édition IL	5
7.3.1	Editer une ligne d'instruction	6
7.3.2	Mise en page des lignes d'instruction	7
7.3.3	Editer un bloc d'organisation	7
7.3.4	Ordre de traitement des instructions et des blocs	7
7.3.5	Règles à respecter pour éditer les blocs	8
7.4	Fenêtre <i>Symbols</i>	9
7.4.1	Ajouter un nouveau symbole dans la liste <i>Symbols</i>	10
7.4.2	Modes d'adressage des opérandes	11
7.4.3	Utiliser un symbole de la liste des <i>Symbols</i> dans le programme IL	12
7.4.4	Symboles local et global	13
7.5	Introduction au jeux d'instructions PCD	14
7.5.1	L'accumulateur	14
7.5.2	Les instructions binaires.	15
7.5.3	Dynamisation	19
7.5.4	Les indicateurs de statut	20
7.5.5	Les instructions mots pour les temporisateurs	21
7.5.6	Les instructions mots pour les compteurs	23
7.5.7	Les instructions dépendantes de l'accumulateur	24
7.5.8	Les instructions mots pour l'arithmétique sur les nombres entiers	25
7.5.9	Les instructions mots pour l'arithmétique flottante	26
7.5.10	Conversion des registres entiers et flottants	26
7.5.11	Registre d'index	27
7.5.12	Sauts de programme.	28
7.6	Edition d'un premier programme d'application	30
7.7	Construction du programme (Build)	32
7.8	Charger le programme dans le PCD	33
7.9	Débugger un programme (Debug)	33
7.9.1	Visualiser le code compilé.	33
7.9.1	Go On/Offline, Run et Stop	34
7.9.2	Mode pas-à-pas	35
7.9.3	Le point d'arrêt	36
7.9.4	Modification de programme en ligne	37
7.9.5	Visualiser et modifier l'état des symboles avec le <i>Watch window</i>	38
7.9.6	Exemple pour les entrées analogiques PCD2.W340	39
7.9.7	Exemple pour les sorties analogiques PCD2.W610	40



## 7 Programmer en IL (Instruction List)

---

### 7.1 Au sommaire de ce chapitre

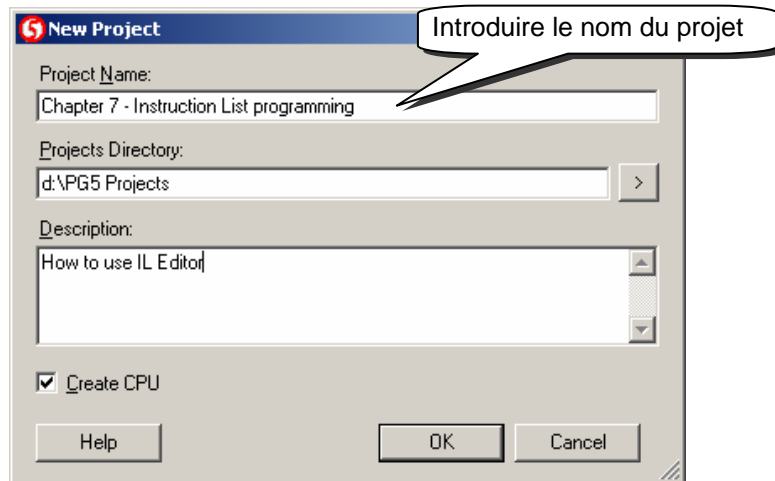
L'éditeur IL, est le moyen le plus flexible et le plus performant pour réaliser la programmation des automates PCD. IL est un terme anglais "*Instruction List*" qui désigne un environnement de programmation non graphique où l'utilisateur écrit les programmes à l'aide d'un puissant jeu d'instruction PCD. Tous les automates PCD utilisent ce jeu d'instruction. Ce qui garantit la portabilité des programmes d'un PCD à l'autre. L'éditeur IL n'est pas seulement une aide précieuse à l'édition des programmes mais aussi un outil de diagnostic et de test en ligne.

## 7.2 Préparer un projet IL

Pour réaliser un exemple, nous vous conseillons de préparer un nouveau projet et de préparer un fichier pour éditer le programme IL.

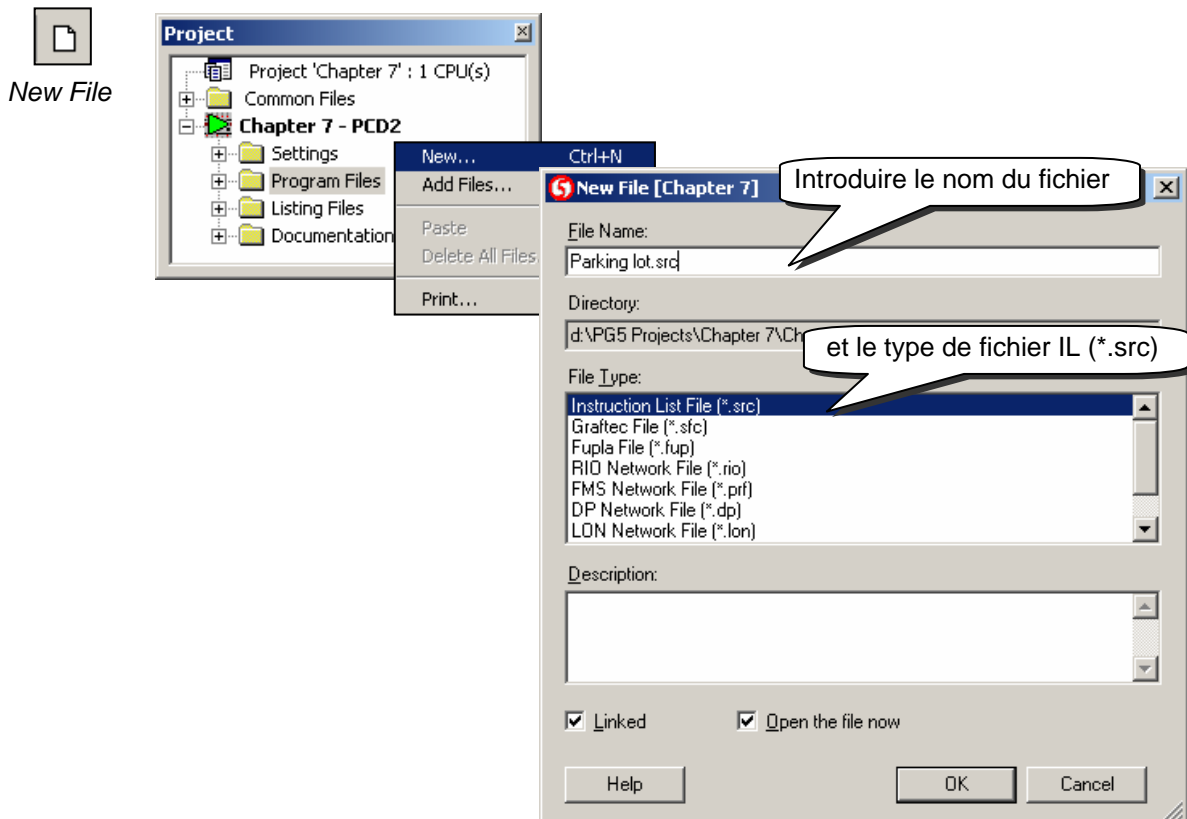
### 7.2.1 Créer un nouveau projet

Dans la fenêtre *SAIA Project Manager*, sélectionner le menu *Project, New...* et créer le nouveau projet.



### 7.2.2 Créer un nouveau fichier IL

Pour ajouter un nouveau fichier programme dans le projet, marquer le répertoire *Program Files*, presser le bouton droite de la souris et sélectionner le menu *New...* ou sélectionner le bouton *New File* de la barre d'outils:





### 7.3 Organisation d'une fenêtre d'édition IL

**Mnémoniques**

Labels	Opérandes	Commentaires
; Main programcycle		
COB	0 0	; Main Program cycle ; no supervisiontime
STH	NewCar_Signal	; Car outers the lot
DYN	DynamiseNewCar	
DEC	NumberOfFreeSlots	;Counts the number of free parking slots
;-----		
STH	Car_leaving_Signal	; Car leaves the lot
DYN	DynamiseCar_Leaving	
INC	NumberOfFreeSlots	
;-----		
STH	NumberOfFreeSlots	; turn in redlight as long as
OUT	red_light	; there are no free parkingslots
ECOB		; End of parking lot program

Début du bloc COB

Fin du bloc COB

Ordre de traitement des instructions à l'intérieur du bloc

Group/Symbol	Type	Address/Value	Comment
NewCar_Signal	Input	0	Input 0 gets a signal with each new car
Car_leaving_Signal	Input	1	Input 1 gets high if a car leave
red_light	Output	32	Stops new cars at the entry
NumberOfFreeSlots	Counter		Counts the number of free parking slots
DynamiseNewCar	F		Flag detects the rising edge of the new car inputs
DynamiseCar_Leaving	F		Flag detects the rising edge on the Car_leaving inputs

L'éditeur IL est comparable à tout autre éditeur de texte du commerce. Les fonctions de base comme marquer un texte, *Copy/Paste* ou *Find/Replace* sont bien présents. Mais l'éditeur IL offre plus qu'un traitement de texte conventionnel:

- Une mise en page propre à l'élaboration des programmes PCD.
- Des couleurs permettent l'identification de chaque type d'information
- Les symboles du programme sont rassemblés dans la fenêtre *Symbols*.
- Permet de visualiser le programme en ligne et de le tester pas-à-pas.

### 7.3.1 Editer une ligne d'instruction

Label	Mnémo.	Opérande	Commentaire
	;incrémenter le registre:		
	STH	Flag	;Copie l'état du flag dans l'accumulateur
	DYN	DFlag	;Force l'accu à l'état haut sur un flanc positif du symbole Flag
	JR	L Next	;Si l'accu est à l'état bas, sauter à next
	INC	Register	;Si non incrémenter le registre
Next:	NOP		

Chaque ligne de programme IL est structurée avec 4 colonnes:

Le label

Représenté en rouge, le label permet de lier une ligne de programme à un symbole. C'est utile pour réaliser les sauts de programme. (JR L Next)

Le mnémonique

Représenté en bleu, le mnémonique ou l'instruction programme définit le travail à effectuer sur une information tel que entrée, sortie, indicateur, registre, ...

L'opérande

Représentée en noir, l'opérande définit une information de type entrée, sortie, indicateur, registre,...



View Symbols  
or Values

Le bouton *View Symbol or Values* permet d'afficher alternativement l'adresse ou le symbole de l'opérande.

STH	Flag		
		STH	F 0

Le commentaire

Les commentaires de l'utilisateur sont représentés en vert et commencent par un point virgule. Ils sont présents à droite de la mnémonique et de l'opérande, mais peuvent aussi occuper toute une ligne.

```

$SKIP
*****
* AUTEUR:      Dupond Fred
* DATE:       18.06.2003
* FICHER:     logic.SRC
*****
$ENDSKIP

```

Un commentaire long de plusieurs lignes ne doit pas nécessairement commencer par un point virgule au début de chaque ligne, il est aussi possible de l'écrire entre deux consignes d'assemblage \$skip et \$endskip, l'ensemble du texte compris entre ces deux instructions ne sera pas pris en compte par l'assembleur.



View User or  
Auto Comment

Le bouton *View User or Auto Comment* permet d'afficher alternativement le commentaire utilisateur lié à chaque ligne de programme ou le commentaire automatique lié à chaque symbole utilisé comme opérande.

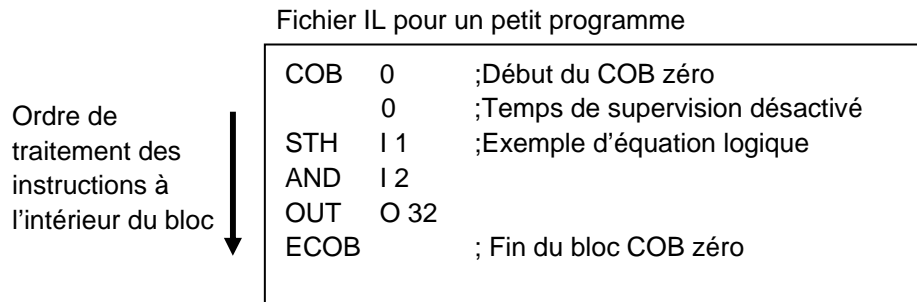
STH	Flag		
			; Copie l'état du flag dans l'accumulateur
		STH	Flag ;Active incrémentation

### 7.3.2 Mise en page des lignes d'instruction

Chaque ligne de programme peut être automatiquement mise en page lors de la pression de la touche *Enter* du clavier, en fin de ligne. Mais seulement si l'option *Auto Format while Typing* est sélectionnée. Voir menu *Tools, Options* de l'éditeur IL. Ce menu permet d'ajuster aussi la largeur des colonnes ...

Si la mise en page n'est pas convenable, il est possible de marquer quelques lignes ou toutes les lignes du fichier avec la souris pour les reformater avec le menu *Tools, Auto Format*.

### 7.3.3 Editer un bloc d'organisation



Le langage de programmation SAIA PCD est structuré avec différents blocs d'organisation dans lesquels l'utilisateur dépose les programmes de l'application. Chaque bloc offre un service particulier: programme cyclique (COB), programme séquentiel (SB,) sous programme (PB), fonction paramétrable (FB), routine d'exception (XOB).

Chaque bloc est délimité par une instruction de début et de fin de bloc. L'instruction COB marque, par exemple, le début d'un bloc qui se termine par une instruction identique mais précédée de la lettre E comme End of COB. Toutes les instructions de programme appartenant à ce bloc sont à placer entre les instructions COB et ECOB et jamais à l'extérieur du bloc.

Même le plus petit des programmes PCD comprend toujours un bloc COB et peut être par la suite étendu avec d'autres blocs selon les besoins.

### 7.3.4 Ordre de traitement des instructions et des blocs

A l'intérieur d'un bloc, le PCD traite les instructions du programme ligne par ligne depuis l'instruction qui délimite le début du bloc jusqu'à l'instruction de fin du bloc.

Si l'ordre des lignes d'instructions à l'intérieur d'un bloc d'organisation est important, l'ordre dans lequel sont écrits les blocs d'organisations est sans aucune importance. Les règles qui définissent l'ordre de traitement des blocs sont différentes:

Au démarrage à froid du PCD, l'automate recherche si le bloc de démarrage à froid XOB 16 est présent. S'il est présent, peut import qu'il le soit au début ou à la fin du fichier, le bloc de démarrage à froid est toujours traité en premier.

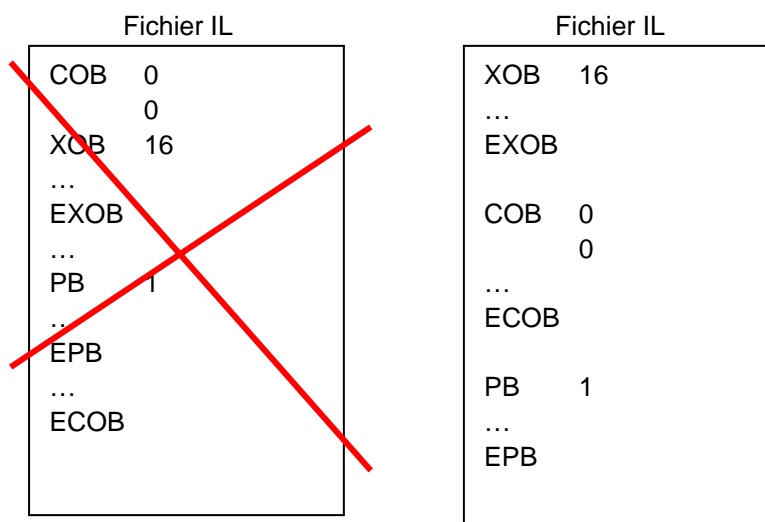
En suite, l'automate cherche les blocs COB présents dans le programme et les traite selon leur numéro d'ordre: COB 0, COB 1, ... COB 15, sans tenir compte de l'ordre dans lequel ils sont présents dans le fichier. Après le dernier COB, le programme reprend au début avec le COB 0.

Tous les blocs pour les programmes séquentiels (SB), sous programmes (PB), fonctions paramétrables (FB) sont appelés par le programme utilisateur avec les instructions CSB (Call SB), CPB (Call PB) et CFB (Call FB). C'est le programme de l'utilisateur qui détermine l'ordre et le moment pour traiter les blocs SB, PB et FB.

Tous les blocs d'exceptions sont automatiquement appelés dès qu'il se produit un événement particulier. Ces événements ne sont pas prévisibles et peuvent se produire à tout moment. Leur ordre de traitement n'est pas définissable. Chaque événement hardware ou software est lié à un XOB distinct. Ces événements ne peuvent pas être modifiés par l'utilisateur. Par contre, l'utilisateur est libre de programmer les actions à entreprendre à l'intérieur de chacun des XOB.

### 7.3.5 Règles à respecter pour éditer les blocs

Même si les blocs peuvent être écrits dans un ordre quelconque, les règles suivantes sont à respecter :



Les blocs ne peuvent pas être écrits les uns dans les autres mais à la suite les uns des autres.

Aucune instruction de programme ne peut être définie à l'extérieur d'un bloc, excepté les définitions des symboles, textes et blocs de données.

## 7.4 Fenêtre Symbols



Show Hide  
Symbols Editor

Group/Symbol	Type	Address/Value	Comment
NewCar_Signal	Input	0	Input 0 gets a signal with each new car
Car_leaving_Signal	Input	1	Input 1 gets high if a car leave
red_light	Output	32	Stops new cars at the entry
NumberOfFreeSlots	Counter		Counts the number of free parking slots
DynamiseNewCar	F		Flag detects the rising edge of the new car inputs
DynamiseCar_Leaving	F		Flag detects the rising edge on the Car_leaving inputs

La fenêtre *Symbols* rassemble les opérands du programme dans une liste qui peut être affichée avec le bouton *Show Hide Symbols Editor* ou le menu *View Symbol Editor*. Chaque ligne définit toutes les informations relatives à un opérande et constitue un symbole:

Le symbole.

Le symbole est un nom qui désigne l'adresse d'une entrée, sortie, indicateur, registre,... Il est conseillé d'utiliser ce nom ou symbole pour éditer le programme et non pas directement l'adresse de l'indicateur ou registre. Ainsi toute correction portant sur l'adresse ou le type de cette information peut être supportée depuis la fenêtre *Symbols*. Il n'est alors pas nécessaire de reporter la correction à chaque ligne du programme mais uniquement dans la fenêtre *Symbols*. Cela évite d'oublier de corriger une ligne du programme et de créer une erreur difficile à trouver.

Syntaxe pour le nom d'un symbole.

Le premier caractère est toujours une lettre suivie par d'autres lettres, chiffres ou du caractère souligné. Il est conseillé d'éviter les caractères tel que ö,è,ç,...

Les majuscules ou minuscules sont sans influence significative, Motor\_On est MOTOR\_ON sont deux symboles identiques.

Le type.

Définit le type de l'opérande: entrées (I), sortie (O), registre (R), compteur (C), temporisateur (T), texte (X), DB, ...

L'adresse.

Chaque type d'opérande compte de nombreuses adresses disponibles:

Entrées et sorties: selon les modules E/S insérés dans le PCD

Indicateurs: F 0, ..., F 8191

Registres: R 0, ..., R 4095<sup>1</sup>, 16383<sup>2</sup>

Temporisateurs, Compteurs: T/C 0, ..., T/C 1599

...

Le commentaire.

Ce commentaire est lié à son opérande et peut être affiché alternativement en lieu et place du commentaire utilisateur lié à chaque ligne de programme.

Utilisez le bouton View User or Auto Comment.



View User or  
Auto Comment

STH Flag ; Copie l'état du flag dans l'accumulateur  
STH Flag ;Active incrémentation

<sup>1</sup> Tous les PCD

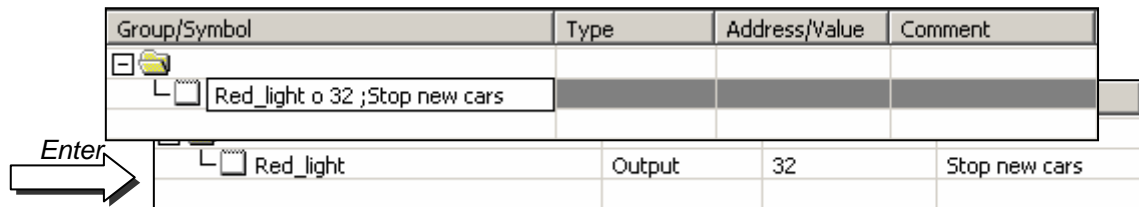
<sup>2</sup> PCD2.M480,PCD3

### 7.4.1 Ajouter un nouveau symbole dans la liste *Symbols*

Méthode simple

Pour ajouter un symbole dans la liste, ouvrir la fenêtre *Symbols*, placer la souris au centre de la fenêtre et sélectionner le menu de contexte *Insert Symbol* avec le bouton droit de la souris. Puis compléter les champs *Group/Symbol*, *Type*, *Address/Value* et *Comment*.

Méthode rapide 1



Group/Symbol	Type	Address/Value	Comment
Red_light o 32 ;Stop new cars			
Red_light	Output	32	Stop new cars

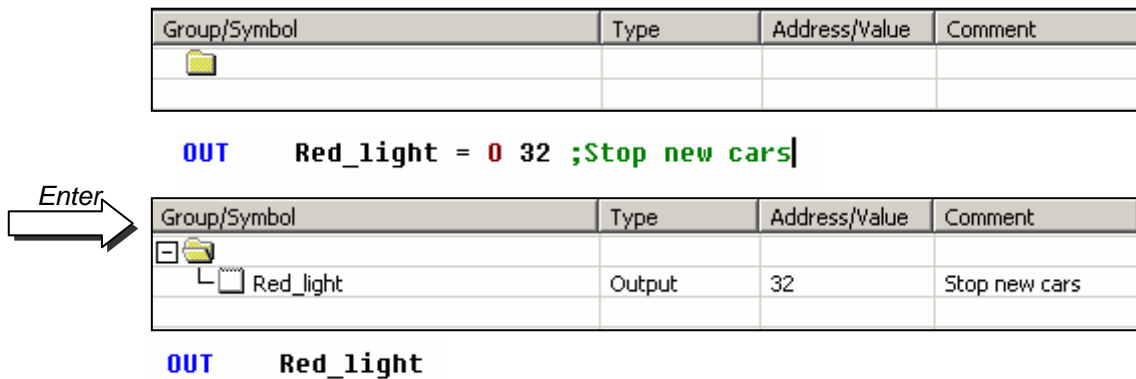
Il est aussi possible d'introduire les variables des divers champs d'informations depuis le champs *Group/Symbol*. C'est plus pratique et plus rapide. Voir l'exemple ci-dessus.

Syntaxe à respecter :

[nom\_du\_symbole] + [ ] + [Type] + [ ] + [Adresse] + [ ; ] + [Commentaire]

Si le nouveau symbole est défini selon la syntaxe ci-dessus, la sélection de la touche *enter* du clavier permet de placer automatiquement les informations dans les champs correspondants.

Méthode rapide 2



Group/Symbol	Type	Address/Value	Comment
OUT Red_light = 0 32 ;Stop new cars			
Red_light	Output	32	Stop new cars

Les nouveaux symboles peuvent être aussi ajoutés à l'édition du programme, pour le réaliser, éditer une ligne de programme avec le mnémonique et son opérande. Pour l'opérande définir le nom du nouveau symbole et sa définition selon la syntaxe suivante:



[nom\_du\_symbole] + [=] + [ Type] + [ ] + [Adresse] + [ ; ] + [Commentaire]

La sélection de la touche *enter* du clavier permet de placer automatiquement le nouveau symbole dans la liste *Symbols*. Seulement si la définition du symbole est correcte, seulement si l'option *Automatically add entered type/value to the Symbol Table* est sélectionnées. (Menu *Tools, Options* de l'éditeur IL)

## 7.4.2 Modes d'adressage des opérandes



La définition d'un symbole ne comprend pas obligatoirement toutes les informations présentées ci-dessus. Nous distinguons trois cas d'utilisation:

### Adressage absolu

Group/Symbol	Type	Address/Value	Comment
 	Output	32	Stops new cars



Le symbole est défini avec un type, une adresse est optionnellement un commentaire. L'usage des adresses absolues représente un inconvénient lors de la correction du type ou de l'adresse. La correction ne peut pas être supportée à partir de la liste des symboles. La correction doit être reportée manuellement pour chaque ligne de programme. Il est donc préférable d'utiliser un adressage symbolique ou dynamique.

### Adressage symbolique

Group/Symbol	Type	Address/Value	Comment
  red_light	Output	32	Stops new cars

Le symbole est défini avec un nom, un type, une adresse est optionnellement un commentaire. Les corrections du symbole, du type ou de l'adresse sont supportés depuis la liste des symboles et reportées automatiquement pour chaque ligne de programme faisant usage de ce symbole.

### Adressage dynamique

Group/Symbol	Type	Address/Value	Comment
  red_light	F		Stops new cars

C'est un adressage symbolique ou l'adresse n'est pas définie. L'outil de programmation attribue automatiquement une adresse correspondant au type défini lors de la construction du programme. L'adresse est prélevée dans une plage d'adresses réservée par les *Settings Software*. (Voir dans le gestionnaire de projet.) Remarque, l'adressage dynamique est disponible avec les indicateurs, compteurs, temporisateurs, registres, textes, DB, COB, PB,FB,SB.

Par contre les adresses doivent toujours être définies pour les entrées, sorties et XOB.

### 7.4.3 Utiliser un symbole de la liste des *Symbols* dans le programme IL

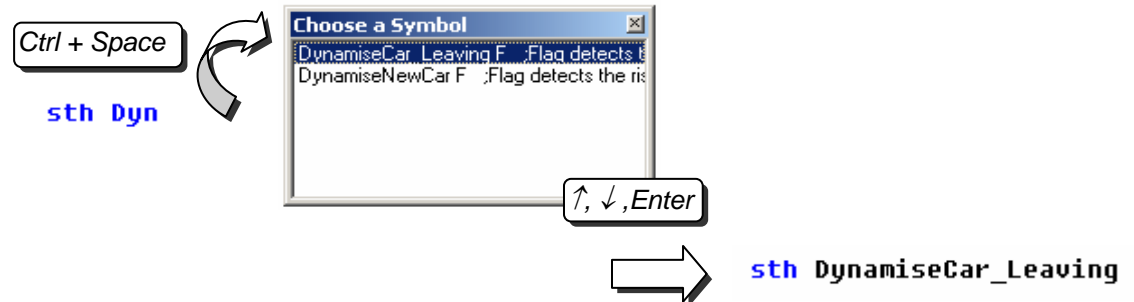
A l'édition du programme, les symboles déjà définis dans la fenêtre *Symbols* peuvent être utilisés de différentes manières:

Introduction du symbole au clavier.

Le nom du symbole est complètement introduit au clavier pour chaque instruction qui en fait usage. Cette manière de faire laisse la possibilité d'éditer le nom du symbole avec une erreur de frappe qui se manifestera seulement à la construction du programme.

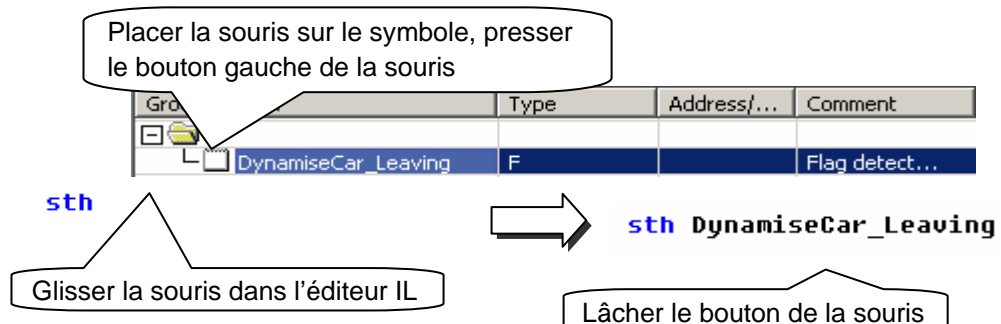
Introduction par recherche sélective.

Group/Symbol	Type	Address/...	Comment
[-] [Folder Icon]			
[-] NewCar_Signal	Input	0	Input 0 get...
[-] Car_leaving_Signal	Input	1	Input 1 get...
[-] red_light	Output	32	Stops new ...
[-] NumberOfFreeSlots	Counter		Counts the...
[-] DynamiseNewCar	F		Flag detect...
[-] DynamiseCar_Leaving	F		Flag detect...



Seul les premiers caractères du nom du symbole sont introduits au clavier. La sélection simultanée des touches *Ctrl+espace* affiche une fenêtre avec la liste des symboles qui commencent avec les mêmes lettres que celles saisies précédemment. Il ne reste qu'à sélectionner le symbole désiré avec la souris ou les touches *↑, ↓* du clavier et valider la sélection avec la touche *Enter*.

Introduction par sélection et glissement



Cette manière de faire usage d'un symbole ne laisse aucune possibilité de faire une erreur de frappe. Dans la fenêtre *Symbols*, placer la souris sur la ligne de définition d'un symbole, presser le bouton gauche de la souris et le maintenir pressé. Glisser la souris dans l'éditeur IL et lâcher le bouton de la souris. Le symbole sélectionné est automatiquement ajouté à l'emplacement pointé par la souris.



### 7.4.4 Symboles local et global

La fenêtre de définition des symboles comporte deux répertoires : *Global* et *Local*



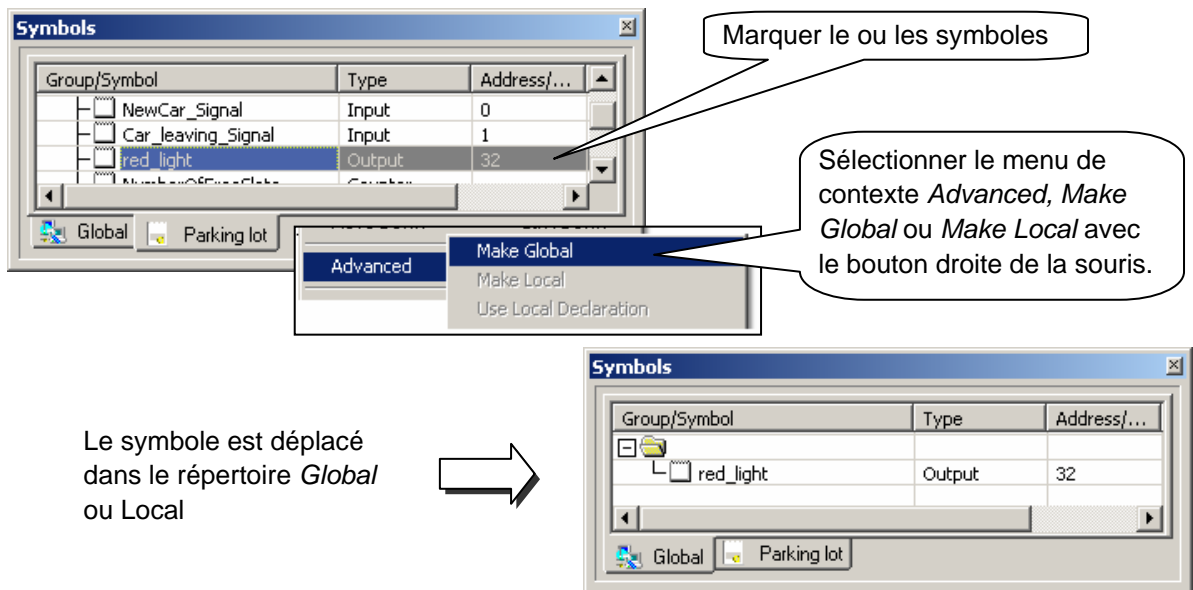
#### Définition

Les symboles locaux sont représentés dans un répertoire qui porte le nom du fichier qui les utilise. Ces symboles sont uniquement utilisables à l'intérieur ce fichier. (*Parking lot.src*)

Les symboles globaux représentés dans le répertoire *Global* sont utilisables par tous les fichiers du CPU. (*Parking lot.src* et *Ventilation.src*)

#### Make Local/Global

Si nécessaire, les symboles de la fenêtre *Symbols* peuvent être déplacés du répertoire local vers le répertoire global ou inversement.



#### Remarque

Tout nouveau symbole défini directement depuis l'éditeur IL est ajouté dans un des répertoires global ou local, selon les présélections de l'option *Add symbols to Global table*. Voir menu de contexte *Advanced*, *Options* de la fenêtre *Symbols*

## 7.5 Introduction au jeux d'instructions PCD

Ce paragraphe présente les caractéristiques générales du jeu d'instructions PCD. Pour de plus amples informations, la description détaillée de chaque instruction est disponible avec le manuel *Guide des instructions 26/733* ou avec les aides PG5. Pour obtenir l'aide spécifique à une instruction depuis l'éditeur IL. Ecrire l'instruction, la marquer avec la souris et sélectionner la touche *F1* du clavier. Une aide générale est aussi disponible avec le menu *Help, Instruction List Help*.

### 7.5.1 L'accumulateur

L'accumulateur est une mémoire binaire que l'automate utilise activement avec les instructions binaires et quelques instructions mots. Le PCD ne dispose que d'un seul accumulateur que nous pouvons considérer comme un indicateur particulier. L'état de l'accumulateur peut être forcé avec les instructions *ACC*. Ces instructions permettent de forcer aussi l'accu avec l'état des indicateurs de statut. (Voir description des indicateurs de statut.)

**Exemples:**

**ACC H**

Force l'accumulateur à l'état haut (Hight)

**ACC L**

Force l'accumulateur à l'état bas (Low)

**ACC C**

Inverse l'état de l'accumulateur.

### 7.5.2 Les instructions binaires.

Les instructions binaires utilisent des opérandes qui ne prennent que deux états distincts: 0 et 1 (bas et haut). Ces instructions sont utilisées pour élaborer des équations binaires avec les états des entrées, sorties, indicateurs, compteurs et temporisateurs du PCD.

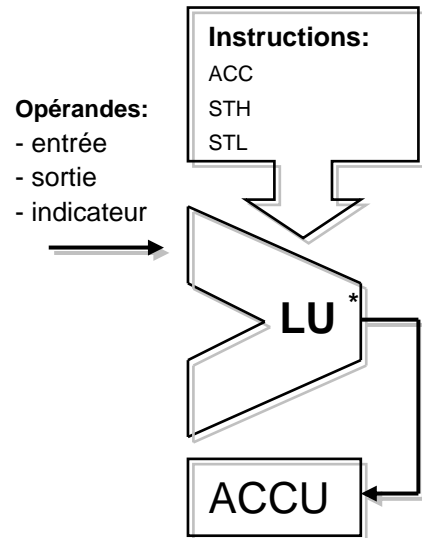
Les instructions binaires font toujours intervenir l'accumulateur. Certaines instructions binaires utilisent l'accumulateur pour y écrire des informations:

**Exemples:**

**ACC H**  
Force l'accumulateur à l'état haut (Hight)

**ACC L**  
Force l'accumulateur à l'état bas (Low)

**STH I 4**  
Copie l'état de l'entrée 4 sur l'accumulateur. L'accumulateur est à l'état haut si une tension de 24 volts est appliquée sur l'entrée 4. L'accumulateur est à l'état bas si une tension de zéro volts est appliquée sur l'entrée 4.



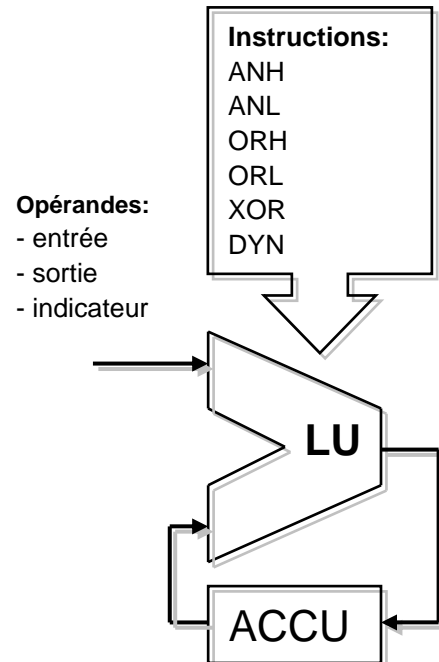
D'autres instructions lisent l'état de l'accumulateur pour effectuer une fonction binaire et placer le résultat dans le même accumulateur:

**Exemples:**

**ANH I 5**  
Lit l'état de l'accumulateur, effectue une fonction logique ET avec l'état de l'entrée 5. L'accumulateur prend l'état du résultat.

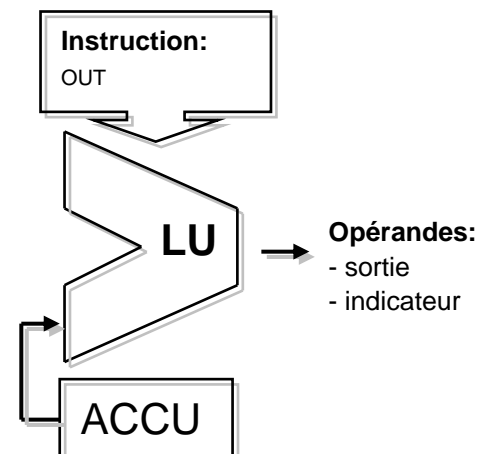
**ORH F 100**  
Lit l'état de l'accumulateur, effectue une fonction logique OU avec l'état de l'indicateur 100. L'accumulateur prend l'état du résultat.

**XOR T 3**  
Lit l'état de l'accumulateur, effectue une fonction logique OU exclusif avec l'état du temporisateur 3. L'accumulateur prend l'état du résultat.



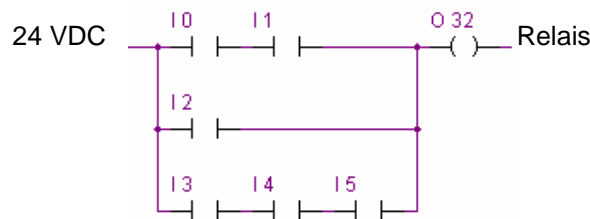
Le résultat d'une équation binaire est toujours sauvegardé dans l'accumulateur. Une instruction *OUT* permet de copier le contenu de l'accumulateur sur une sortie ou un indicateur:

**Exemple:**  
**OUT O 32**  
 Copie l'état de l'accumulateur sur la sortie 32.  
 Si l'accumulateur est à l'état haut, une tension de 24 Volts est appliquée sur la sortie 32.  
 Si l'accumulateur est à l'état bas, une tension de zéro Volts est appliquée sur la sortie 32.



**Exemple:** programmer une équation binaire simple

Cet exemple de programme réalise l'équation binaire:  $O32 = I0 * I1 + I2 + I3 * I4 * I5$  que nous pouvons aussi représenter par le schéma suivant :



Une équation binaire commence toujours avec une instruction *STH* ou *STL*, puis se succèdent les fonctions *ANH* (\*), *ORH* (+), *XOR* nécessaires.

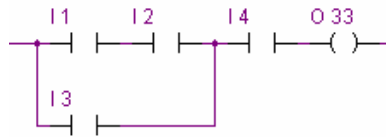
Remarquons que l'instruction *ORH* est prioritaire sur *ANH*. Chaque instruction *ORH* marque le début d'une nouvelle ligne de contacts du schéma ci-dessus. Le résultat partiel ou final de l'équation binaire est toujours placé dans l'accumulateur. L'instruction *OUT* permet d'exploiter le résultat de l'accumulateur pour modifier l'état d'une sortie ou d'un indicateur.

```

COB 0      ;Début du programme cyclique
0
STH I 0    ;Copie l'état de l'entrée I 0 dans l'accumulateur: Accu = I0
ANH I 1    ;Fonction ET entre l'état de l'accumulateur et l'entrée 1:Accu = I0*I1
ORH I 2    ;Fonction OU entre l'état de l'accumulateur et l'entrée 2:Accu= I0*I1+I2
ORH I 3    ; Accu = I0*I1+I2+I3
ANH I 4    ; Accu = I0*I1+I2+I3*I4
ANH I 5    ; Accu = I0*I1+I2+I3*I4*I5
OUT O 32   ;Copie le résultat de l'équation présent dans l'accumulateur sur la sortie
ECOB      ;Fin du programme cyclique
  
```

**Exemple:** programmer une équation binaire avec forçage de l'ordre d'évaluation

Cet exemple de programme réalise l'équation binaire :  $O33 = (I1 * I2 + I4) * I3$  que nous pouvons aussi représenter par le schéma suivant :



Parfois il est nécessaire de modifier l'ordre des priorités des fonctions binaires. Nous le faisons généralement en introduisant des parenthèses dans les équations. Mais le jeu d'instruction PCD ne dispose pas de parenthèses. Alors il faut diviser l'équation en deux sous équations plus petites. Une première équation qui évalue le résultat de la parenthèse et le mémorise temporairement sur un indicateur et une seconde équation qui reprend le résultat intermédiaire mémorisé sur l'indicateur pour évaluer l'équation finale.

```

COB 0
    0
    STH I 1      ;Première équation
    ANH I 2
    ORH I 4
    OUT F 0      ;Résultat de la fonction entre parenthèse: F0 =(I1*I2+I4)

    STH F 0      ;Seconde équation
    ANH I 3
    OUT O 33     ;Résultat final : O 33 = F0*I3
    ECOB
    
```

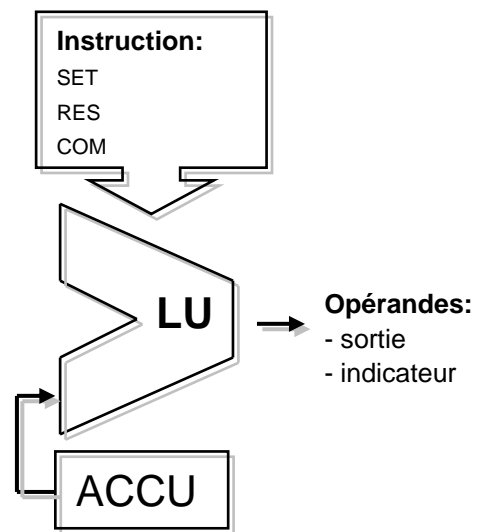
D'autres instructions binaires permettent d'exploiter aussi l'accumulateur pour modifier l'état d'une sortie ou d'un indicateur. Chaque instruction supporte une fonctionnalité différente.

**Exemple:**

**SET O 32**  
Si l'accumulateur est à l'état haut, la sortie 32 est forcée à l'état haut. Si non la sortie reste dans l'état actuel.

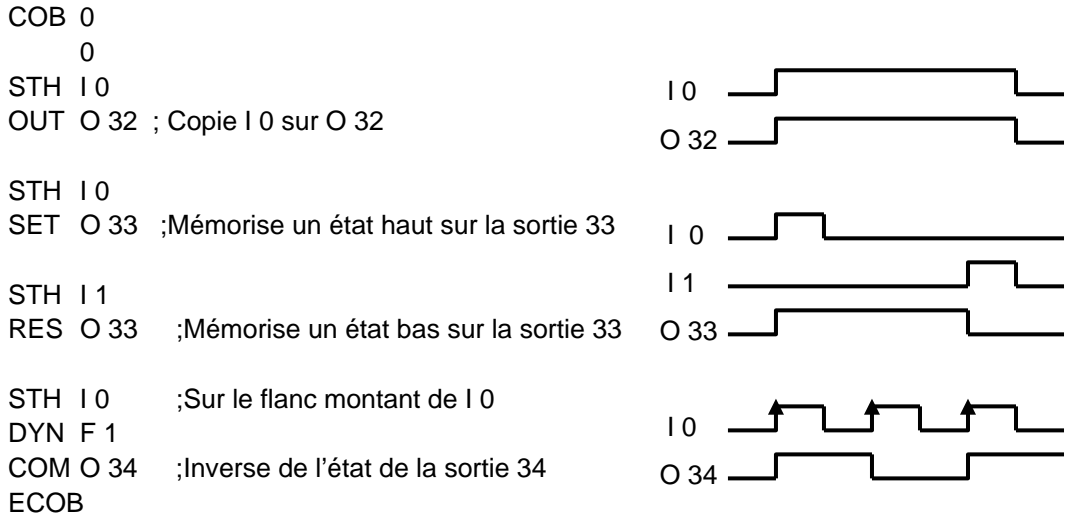
**RES O 32**  
Si l'accumulateur est à l'état haut, la sortie 32 est forcée à l'état bas. Si non la sortie reste dans l'état actuel.

**COM O 33**  
Si l'accumulateur est à l'état haut, l'état de la sortie 33 est inversé. Si non la sortie reste dans l'état actuel.



Exemple:

Cet exemple met en évidence les différences entre les instructions OUT, SET, RES, et COM



Certaines instructions binaires se terminent avec la lettre H ou L. Les instructions qui se terminent par un L inversent l'état de l'information avant de réaliser leur fonction.

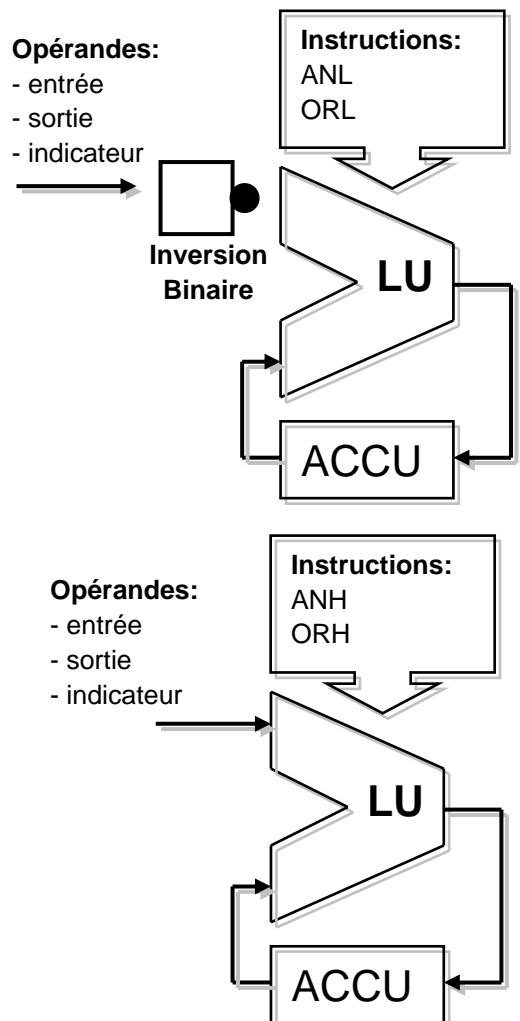
**Exemples:**

**STH I 4**  
Copie l'état de l'entrée 4 sur l'accumulateur. L'accumulateur est à l'état haut si une tension de 24 volts est appliquée sur l'entrée 4.

**STL I 4**  
Copie l'état inverse de l'entrée 4 sur l'accumulateur. L'accumulateur est à l'état bas si une tension de 24 volts est appliquée sur l'entrée 4.

**ANH I 5**  
Effectue une fonction logique ET entre l'état de l'accumulateur et l'état de l'entrée 5.

**ANL I 5**  
Effectue une fonction logique ET entre l'état de l'accumulateur et l'état inverse de l'entrée 5.



### 7.5.3 Dynamisation

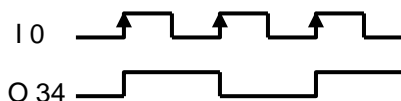
Les instructions binaires utilisent généralement l'état binaire bas ou haut pour effectuer une fonction binaire ou modifier l'état d'une sortie, indicateur.

Parfois ce n'est pas l'état binaire bas ou haut qui nous intéresse mais le passage de l'état bas à l'état haut pour incrémenter un compteur, par exemple.

Pour détecter un flanc montant, procéder comme suit: placer le résultat d'une équation binaire dans l'accumulateur et utiliser l'instruction *DYN* pour rechercher le flanc positif. Après l'instruction *DYN*, l'état de l'accumulateur est haut en cas de détection d'un flanc positif et bas dans le cas contraire. L'indicateur utilisé par l'instruction *DYN* ne doit être utilisé que par une seule instruction de dynamisation. Car il est utilisé pour conserver l'état de l'équation binaire pour le prochain cycle de programme.

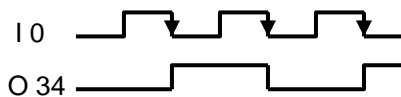
**Exemple:** détection d'un flanc montant

```
STH I 0
DYN F 3
COM O 34
```



**Exemple:** détection d'un flanc descendant

```
STL I 0
DYN F 3
COM O 34
```



Pour bien comprendre l'influence de l'instruction *DYN* sur le programme présenté ci-dessus. Nous vous proposons de supprimer l'instruction *DYN* et d'observer le comportement du programme.

### 7.5.4 Les indicateurs de statut

Contrairement aux instructions binaires, les instructions mots n'utilisent pas souvent l'accumulateur. Mais modifient presque toujours les indicateurs de statut.

Les 4 indicateurs de statut du PCD sont modifiés par les instructions mots et nous informent du résultat obtenu.

Indicateur positif	P	Est positionné si le résultat est positif.
Indicateur négatif	N	Est positionné si le résultat est négatif.
Indicateur zéro	Z	Est positionné si le résultat est zéro
Indicateur erreur	E	Est positionnée en cas d'erreur

Le flag erreur peut être positionné pour de nombreuses raisons et provoquer l'appel du bloc d'exception XOB 13 :

Dépassement de capacité après une instruction de multiplication de deux grands nombres

Division par zéro

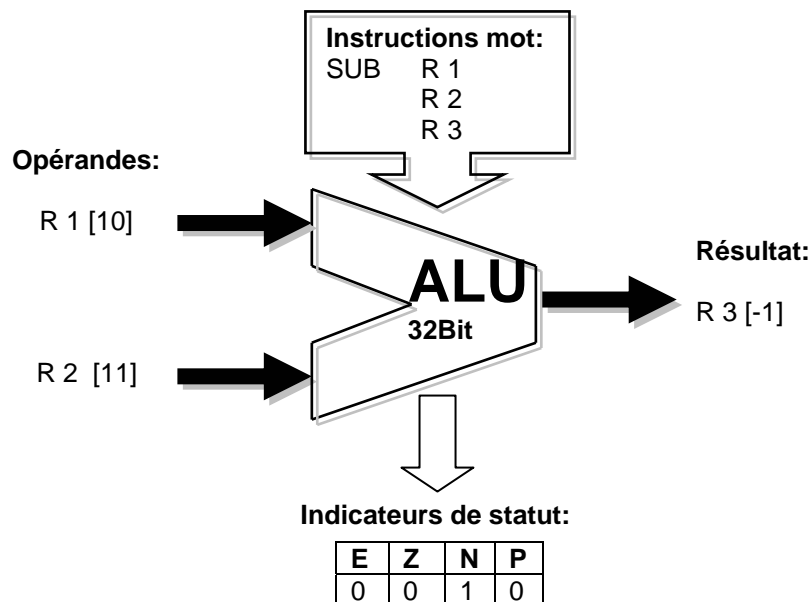
Racine carrée d'un nombre négatif

Erreur à l'assignation d'une interface de communication (Instruction SASI)

...

**Exemple:** état des indicateurs de statut après une soustraction arithmétique

Les indicateurs de statut sont positionnés selon le résultat de la soustraction ( $R3 = R1 - R2$ ). Les valeurs des registres sont présentés entre crochet []. Le résultat de la soustraction est négatif, seul l'indicateur N est positionné.

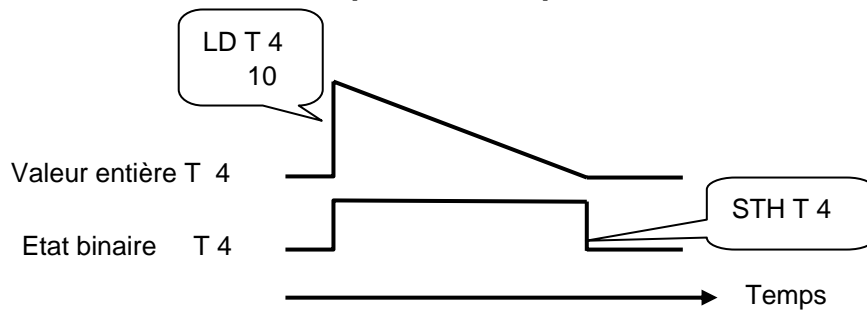


Si nécessaires les indicateurs de statut peuvent être copiés dans l'accumulateur pour les exploiter avec les instructions binaires, les instructions de saut de programmes ou lors d'appel à des blocs PB, FB et SB:

ACC P	Copie l'indicateur de statut P dans l'accumulateur
ACC N	Copie l'indicateur de statut N dans l'accumulateur
ACC Z	Copie l'indicateur de statut Z dans l'accumulateur
ACC E	Copie l'indicateur de statut E dans l'accumulateur



### 7.5.5 Les instructions mots pour les temporisateurs



Les temporisateurs disposent de deux informations: la valeur entière de temporisation et l'état binaire du temporisateur.

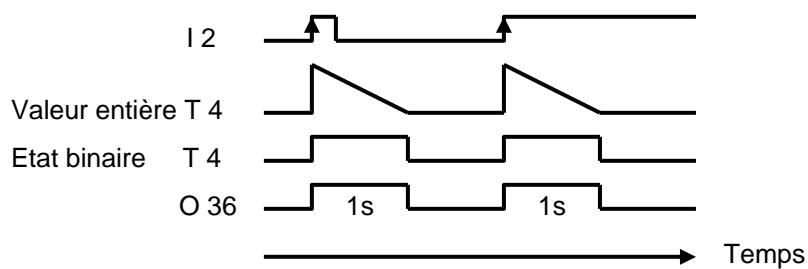
Pour réaliser une temporisation, charger la valeur de temporisation avec une valeur entière et positive qui détermine la durée de temporisation en dixième de seconde <sup>1</sup>. L'automate décrémente automatiquement la valeur de temporisation jusqu' à ce qu'elle atteigne la valeur zéro. L'état binaire du temporisateur est haut durant toute la décrémentation de la temporisation et tombe à l'état bas lorsque la valeur de temporisation atteint zéro.

<p><b>Charger une temporisation</b> LD T 4</p> <p>Si l'accumulateur est à l'état haut, le temporisateur T 4 est chargé avec une constante de 10. Si non le temporisateur conserve la valeur</p>	<p><b>Lecture de l'état du temporisateur.</b> Utiliser une instruction binaire tel que: STH T 4 , ANH T 4 , ORH T 4 , ...</p>
---	---

Exemple:

Sortir une impulsion d'une seconde sur la sortie 36 pour chaque flanc montant de l'entrée 2

Diagramme des états:



Programme correspondant:

```

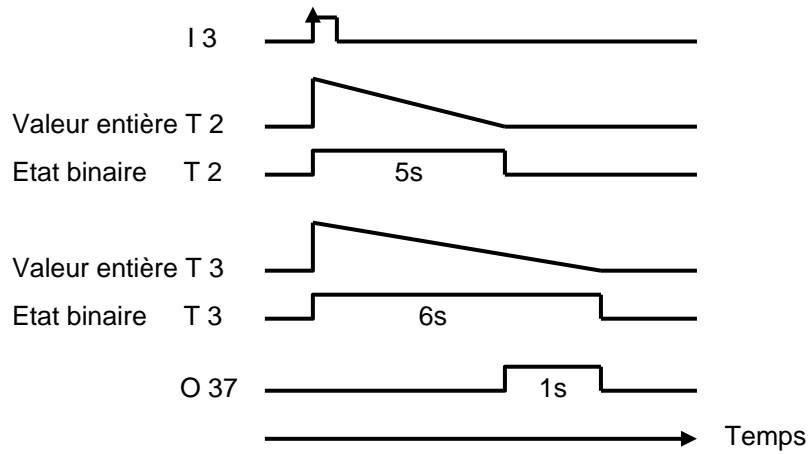
COB 0
0
STH I 2 ;La détection du flanc montant de l'entrée 2 ...
DYN F 2 ;...positionne l'accu à l'état haut
LD T 4 ;Si l'accu est haut, charge la temporisation pour 10 unités de temps
10
STH T 4 ;Copie l'état logique de la temporisation sur la sortie 36
OUT O 36
ECOB
    
```

<sup>1</sup> La base de temps peut être autre que 1/10<sup>e</sup> de seconde (valeur par défaut). Elle peut être modifiée à partir des *settings softwares*

Exemple:

Sortir une impulsion d'une seconde sur la sortie 37 avec un retard de 5 secondes pour chaque flanc montant de l'entrée 3

Diagramme des états:



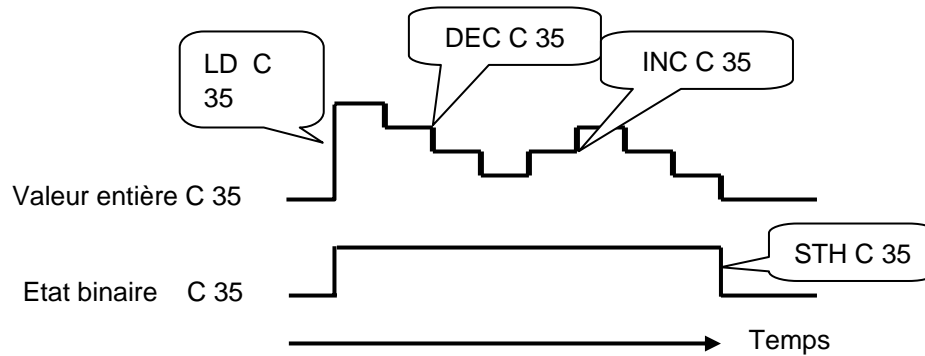
Programme correspondant:

```

COB 0
  0
STH I 3
DYN F 3
LD T 2
      50
LD T 3
      60
STH T 2
XOR T 3
OUT O 37
ECOB

```

### 7.5.6 Les instructions mots pour les compteurs



Comme pour les temporisateurs, les compteurs disposent de deux informations: la valeur entière de comptage et l'état binaire du compteur.

Pour réaliser un comptage, charger la valeur du compteur avec une valeur entière et positive.

Contrairement aux temporisateurs, les compteurs sont incrémentés ou décrémentés à partir du programme PCD. L'état binaire du compteur est haut lorsque la valeur de comptage est plus grande que zéro et tombe à l'état bas lorsque la valeur de comptage atteint zéro.

<p><b>Charger un compteur</b>                  LD C 35                  10                  Si l'accumulateur est à l'état haut, le compteur 35 est chargé avec une constante de 10. Si non le compteur conserve la valeur actuelle.</p>	<p><b>Lecture de l'état du compteur.</b>                  Utiliser une instruction binaire tel que:                  STH C 35, ANH C 35, ORH C 35, ...</p>
<p><b>Incrémenter un compteur</b>                  INC C 35                  Si l'accumulateur est à l'état haut, le compteur 35 est incrémenté d'une unité. Si non le compteur conserve la valeur actuelle.</p>	<p><b>Décrémenter un compteur</b>                  DEC C 35                  Si l'accumulateur est à l'état haut, le compteur 35 est décrémenté d'une unité. Si non le compteur conserve la valeur actuelle.</p>

**Indicateurs de statut**

Les instructions INC et DEC compteur modifient les indicateurs de statut en fonction du résultat de l'opération. (**Positif, Négatif, Zéro, Erreur**).

**Exemple :** Compter les impulsions d'une entrée binaire avec un compteur.

```

COB 0
    0
    STH I 2          ; Copie l'état de l'entrée sur l'accumulateur
    DYN F 3         ; Force l'accu à l'état haut sur un flanc positif de I 2
    INC C 35        ; Si l'accu est à l'état haut, incrémente le compteur
    ECOB
    
```

Les instructions *STH* et *DYN* permettent de lire l'information de l'entrée 2 et de positionner l'accu à l'état haut en cas de flanc positif et à l'état bas en cas d'absence de flanc. Selon l'état de l'accu, L'instruction *INC* incrémente le compteur 35.

### 7.5.7 Les instructions dépendantes de l'accumulateur

Nous avons vu que les instructions binaires utilisent intensément l'accumulateur, même certaines instructions mots utilisent aussi l'accumulateur.

Mais pas toutes les instructions utilisent l'accumulateur de la même manière. Il existe 7 instructions qui l'utilisent d'une manière particulière. Ce sont les instructions dépendantes de l'accumulateur. Elles ne sont traitées que si l'accumulateur à été préalablement mis à l'état haut. L'état de l'accumulateur est donc une condition déterminante.

Ces 7 instructions dépendantes de l'accumulateur sont les suivantes :

SET	
RES	
COM	
LD	Uniquement pour les temporisateurs et compteurs
LDL	Uniquement pour les temporisateurs et compteurs
INC	Uniquement pour les temporisateurs et compteurs
DEC	Uniquement pour les temporisateurs et compteurs

Exemple:

Créer une base de temps qui inverse une sortie à chaque seconde.

Cet exemple utilise trois instructions. La première *STL*, utilise l'accumulateur pour y déposer l'état inverse du temporisateur. Les deux suivantes, *LD* et *COM*, sont dépendantes de l'accumulateur, elles ne chargent la temporisation et inversent la sortie que si l'accumulateur à été préalablement mis à l'état haut par l'instruction *STL*

```
COB 0
    0
STL T 1 ;Si la temporisation est à l'état bas, l'accumulateur est dans l'état haut
LD T 1 ; charge la temporisation avec 10 unités de temps
    10
COM O 38 ; inverse l'état de la sortie
ECOB
```

### 7.5.8 Les instructions mots pour l'arithmétique sur les nombres entiers

Ces instructions sont utilisées pour élaborer des équations arithmétiques avec des registres et des constantes de type entier. Chaque instruction arithmétique compte plusieurs lignes et utilise des opérandes sur des registres ou des constantes, mais le résultat est toujours placé dans un registre.

<b>Addition</b>	<b>Soustraction</b>	<b>Racine carrée</b>
ADD R 0 R 1 R 3 ;R3=R0+R1	SUB R 0 K 18 R 3 ;R3=R0-18	SQR R 100 R 101
<b>Multiplication</b>	<b>Division</b>	<b>Comparaison</b>
MUL K 5 R 1 R 3 ;R3=5*R1	DIV R 0 R 1 R 3 ;R3=R0/R1 R 4 ;Reste	CMP R 0 R 1
<b>Incrémenter</b>	<b>Décrémenter</b>	<b>Initialiser un registre</b>
INC R 0 ;R0= R0+1	DEC R 0 ;R0= R0-1	LD R 0 K 19 ; R 0 = 19

**Indicateurs de statut**

Toutes les instructions arithmétiques ci-dessus modifient les indicateurs de statut en fonction du résultat de l'opération. (**P**ositif, **N**égatif, **Z**éro, **E**rrer), excepté l'instruction pour charger un registre avec une constante (LD)

**Différences entre registres et temporisateurs/compteurs**

Contrairement aux compteurs les instructions pour charger une constante dans un registre, incrémenter et décrémenter les registres ne sont pas dépendants de l'état de l'accumulateur.

La valeur du registre à incrémenter et décrémenter peut être un nombre entier positif mais aussi négatif.

**Exemple:**

Comparer le contenu de deux registres et enclencher trois sorties selon les conditions suivantes :

Registres	O 32	O 33	O 34
R 0 > R 1	Haut	Bas	Bas
R 0 = R 1	Bas	Haut	Bas
R 0 < R 1	Bas	Bas	Haut

L'instruction de comparaison effectue la soustraction R 0 – R 1 et positionne les indicateurs de statut selon le résultat:

Registres	P	N	Z	E
R 0 > R 1	1	0	0	0
R 0 = R 1	1	0	1	0
R 0 < R 1	0	1	0	0

CMP R 0 ;Effectue la soustraction R 0 – R 1, les indicateurs de statut sont  
R 1 ; modifiés selon le résultat de la soustraction

ACC P

OUT O 32 ; R 0 > R 1

ACC Z  
 OUT 0 33 ; R 0 = R 1  
 ACC N  
 OUT 0 34 ; R 0 < R 1

### 7.5.9 Les instructions mots pour l'arithmétique flottante

Ces instructions sont utilisées pour élaborer des équations arithmétiques avec des registres et des constantes de type flottant. Chaque instruction arithmétique commence par la lettre F pour signaler que c'est une instruction sur les nombres flottants. Les opérandes de ces instructions sont toujours des registres et jamais des constantes. Si une constante doit être utilisée. Il faut charger une constante dans un registre puis utiliser le registre avec l'instruction d'arithmétique flottante.

Addition	Soustraction	Racine carrée
FADD R 0 R 1 R 3 ;R3=R0+R1	FSUB R 0 R 1 R 3 ;R3=R0-R1	FSQR R 100 R 101 ;résultat
Multiplication	Division	Comparaison
FMUL R 0 R 1 R 3 ;R3=R0*R1	FDIV R 0 R 1 R 3 ;R3=R0/R1	FCMP R 0 R 1
Sinus	Cosinus	Arc tangente
FSIN R 10 R 11 ;résultat	FCOS R 10 R 11 ;résultat	FATAN R 10 R 11 ;résultat
Exponentielle	Logarithme népérien	Valeur absolue
FEXP R 20 R 21 ;résultat	FLN R 20 R 21 ;résultat	FABS R 30 R 31 ;résultat

Indicateurs de statut

Toutes les instructions ci-dessus modifient les indicateurs de statut excepté l'instruction LD pour charger une constante de type flottant.

Initialiser un registre
LD R 0 3.1415E0 ; R 0 = PI

### 7.5.10 Conversion des registres entiers et flottants

Le PCD dispose d'instructions distinctes pour les opérations arithmétiques sur les entiers et les flottants. Si un programme d'application doit additionner ou multiplier deux registres dont un contient un entier et le second un flottant, il serait nécessaire de convertir les registres dans un format entier ou flottant avant d'effectuer l'opération arithmétique.

Conversion entier-flottant	Conversion flottant-entier
IFP R 0 ; entier -> flottant 0 ; exposant	FPI R 0 ;flottant -> entier 0 ; exposant

### 7.5.11 Registre d'index

Chaque COB dispose d'un registre un peu particulier, le registre d'index. Le contenu du registre d'index peut être contrôlé avec les instructions suivantes:

<b>SEI K 10</b>	<b>SEt</b> Index register	Charge le registre d'index avec une constante 10
<b>INI K 99</b>	<b>IN</b> crement Index register	Incrémente le registre d'index et positionne l'accu à l'état haut tant que: Registre d'index <= K 99
<b>DEI K 5</b>	<b>DE</b> crement Index register	Décrémente le registre d'index et positionne l'accu à l'état haut tant que: Registre d'index >= K 5
<b>STI R 0</b>	<b>ST</b> ore Index register	Copie le registre d'index dans un registre 0
<b>RSI R 0</b>	<b>ReS</b> tore Index register	Copie le registre 0 dans le registre d'index

Beaucoup d'instructions PCD supportent l'usage du registre d'index. Ce registre permet de réaliser l'adressage indirect des registres, indicateurs, entrées, sorties, temporisateurs, ..., utilisés par les instructions du programme. Ces instructions sont les mêmes que celles utilisées habituellement mais complétées par une lettre X.

Exemple:

Les registres sont des mémoires non volatiles. C'est à dire qu'ils conservent leurs informations lors d'une coupure d'alimentation ou d'un démarrage à froid.

Si nous voulons rendre une plage de 100 registres volatiles, nous devons initialiser ces 100 registres avec une valeur de zéro lors du démarrage à froid.

Pour initialiser un registre à zéro nous pouvons faire usage de l'instruction suivante:

```
LD R 10
  K 0
```

Si nous avons 100 registres (R 10 à 109) à initialiser cela nous oblige à écrire 100 fois cette instruction en ne changeant que l'adresse du registre. Cela serait un peu long à réaliser.

Une autre solution consisterait à initialiser le registre d'index avec un index de zéro et réaliser une boucle de programme pour charger le premier registre à zéro, incrémenter l'index. Ainsi pour chaque boucle nous chargeons zéro dans un registre différent (R 10, R 11,.... R 109). A la 100e boucle le compteur d'index atteint la valeur d'index maximum (K 99), force l'accumulateur à l'état bas et permet de sortir de la boucle pour traiter la suite du programme.

```
XOB 16 ;Bloc de démarrage à froid
SEI K 0 ;Index = 0
LOOP:LDX R 10 ;Charge le registre dont l'adresse = 10 + index
      0 ;avec zéro
INI K 99 ;Incrémente l'index et modifie l'état de l'accu
JR H LOOP ;Si l'accu est haut, saut de programme au label LOOP
EXOB
COB 0 ;Bloc d'organisation cyclique
  0
...
ECOB
```

### 7.5.12 Sauts de programme.

Le jeu d'instructions IL dispose de trois instructions de sauts de programmes. Elles permettent de traiter une séquence d'instruction selon une condition binaire ou d'effectuer des boucles de programmes portant sur des tâches répétitives (indexation).


#### Les instructions de saut

<b>JR</b>	Saut relatif	Saute quelques lignes en avant ou en arrière par rapport à la ligne où se trouve l'instruction JR
<b>JPD</b>	Saut direct	Saute à un numéro de ligne par rapport au début du bloc (COB, PB,...)
<b>JPI</b>	Saut indirect	Comme pour JPD, mais le numéro de ligne est placé dans un registre

La destination du saut est généralement indiquée par un label définissant une ligne du programme. Mais il est aussi possible de définir le saut relatif avec le nombre de lignes à sauter en avant ou en arrière.


Saut faisant usage d'un label de ligne:

```
JR   L Next
      INC   R 10
Next: NOP
```



Saut faisant usage du nombre de lignes:

```
JR   L +1
      INC   R 10
      NOP
```



Le saut doit toujours être effectué à l'intérieur du bloc courant (COB, PB,...) et jamais à l'extérieur.

Si nécessaire le saut peut être toujours effectué ou seulement sous une condition binaire déterminée, tel l'état de l'accumulateur ou de l'un des indicateurs de statut.

#### Syntaxe d'une instruction pour un saut inconditionnel

Mnémonique	Label	Description
<b>JR</b>		Le saut est toujours effectué à la ligne correspondant au label
<b>JPD</b>		
<b>JPI</b>		

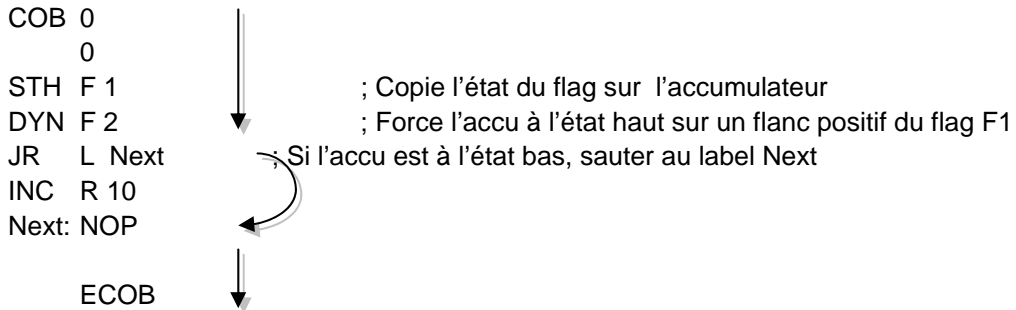
#### Syntaxe d'une instruction pour un saut conditionnel

Mnémonique	Condition	Label	Description
<b>JR</b>	H		Si l'accum est haut (High)
<b>JPD</b>	L		Si l'accum est bas (Low)
<b>JPI</b>	Z		Si l'indicateur de statut Z est haut
	P		Si l'indicateur de statut P est haut
	N		Si l'indicateur de statut N est haut
	E		Si l'indicateur de statut E est haut



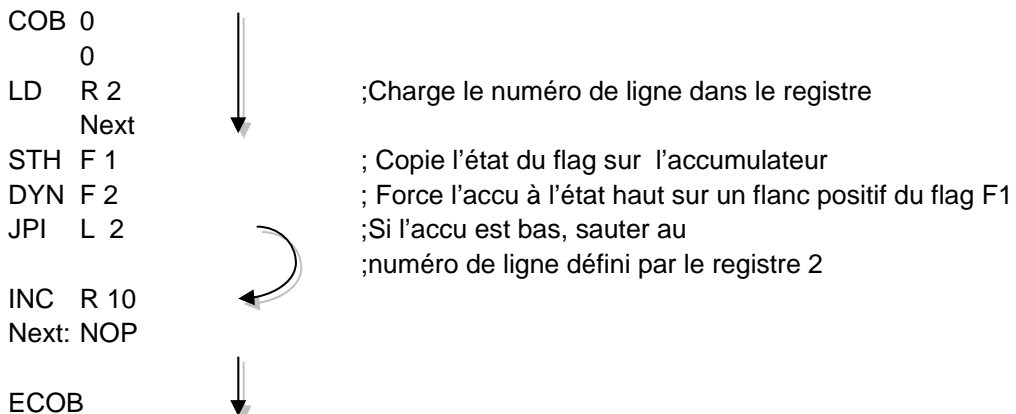
**Exemple :** Compter les impulsions d'une entrée binaire avec un registre. (saut relatif)

Contrairement aux compteurs, l'instruction d'incrémentation d'un registre n'est pas dépendante de l'état de l'accumulateur. L'usage d'une instruction de saut est alors utile pour incrémenter le registre lorsque cela est nécessaire seulement.



Les instructions *STH* et *DYN* permettent de lire l'information de l'indicateur F 1 et de positionner l'accu à l'état haut en cas de flanc positif et à l'état bas en cas d'absence de flanc. Selon l'état de l'accu, l'instruction *JR* saute à la ligne correspondant au label *Next*: ou d'incrémenter le registre avec l'instruction *INC*. La lettre *L* indique la condition pour effectuer le saut, pour cet exemple le saut n'est effectué que si l'accumulateur est à l'état bas (Low).

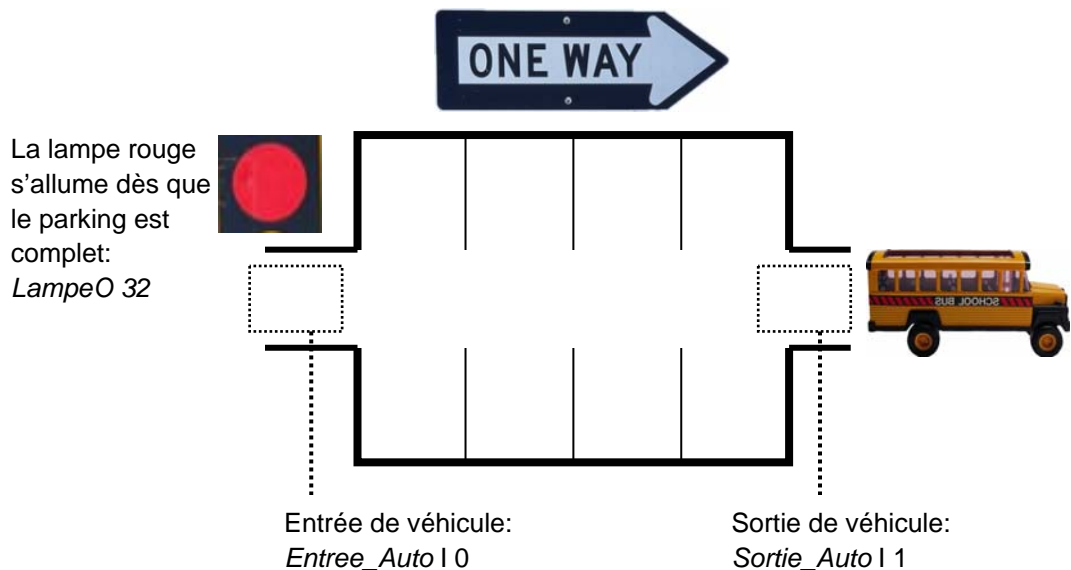
**Exemple :** Solution avec un saut indirect



Le saut indirect offre une grande flexibilité, le programme peut modifier lui même le numéro de ligne pour effectuer le saut.

## 7.6 Edition d'un premier programme d'application

Compter le nombre de places encore disponibles d'un parking de 8 places et allumer une lampe rouge lorsqu'il est complet.

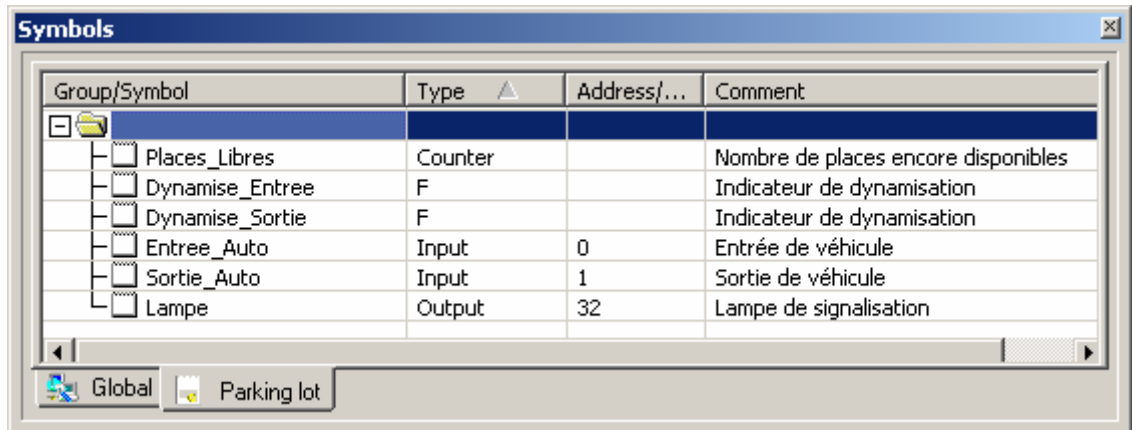


Au démarrage à froid du PCD, nous supposons que toutes les places de parc sont disponibles. Il faut alors commencer par initialiser le compteur de places libres avec la valeur 8. Cette initialisation est effectuée uniquement au démarrage du PCD. Nous la programmerons donc dans le bloc de démarrage à froid XOB 16. Les autres fonctions de ce programme seront assurées par un bloc d'organisation cyclique. (COB)

A l'entrée, le capteur *Entree\_Auto* délivre une impulsion à chaque entrée d'un nouveau véhicule. Il convient alors de détecter le flanc montant de ce signal pour décrémenter le compteur de places libres.

A la sortie, un second capteur *Sortie\_Auto* délivre une impulsion à chaque sortie de véhicule. Il convient alors de détecter le flanc montant de ce signal pour incrémenter le compteur de places libres.

Si le parking est complet la valeur entière du compteur indique zéro places disponibles. L'état logique du compteur nous informe de cette situation lorsqu'il est bas. Il convient alors d'enclencher la lampe rouge à l'entrée du parking.



; Bloc de démarrage à froid

-----

```
XOB 16
LD  Places_Libres ;Initialiser le compteur de places libres
    8
EXOB
```

; Bloc d'organisation cyclique

-----

```
COB 0
    0

STH Entree_Auto ;Sur le flanc montant du signal
DYN Dynamise_Entree
DEC  Places_Libres ;décrémenter le compteur de places libres
```

-----

```
STH Sortie_Auto ;Sur le flanc montant du signal
DYN Dynamise_Sortie
INC  Places_Libres ;incrémenter le compteur de places libres
```

-----

```
STL Places_Libres ;Contrôle la lampe rouge à l'entrée du parking
OUT Lampe

ECOB
```

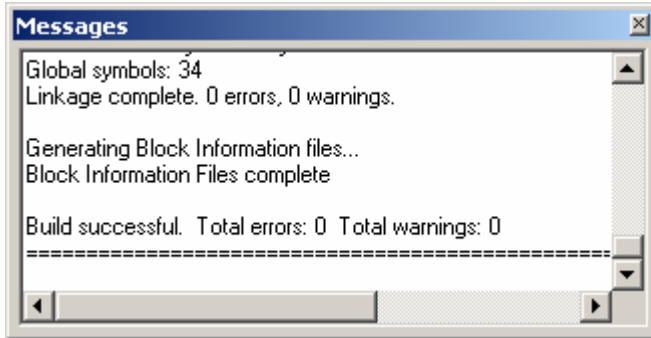
### 7.7 Construction du programme (Build)



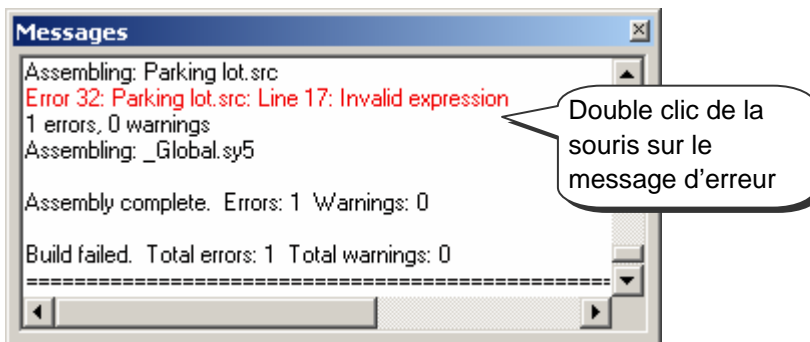
Build All

Le programme utilisateur est complètement décrit, mais pas encore utilisable par le PCD. Il est nécessaire de le traduire en un fichier binaire. C'est ce que fait l'outil de programmation lorsque l'utilisateur actionne le menu *CPU Build* ou le bouton *Build* du gestionnaire de projet ou de l'éditeur IL.

La fenêtre des Messages nous renseigne sur le déroulement de la construction du programme. Nous remarquerons plusieurs étapes de construction: l'assemblage et le linkage. Si le programme est correctement édité, la construction se termine par un message *Build successful. Total errors 0 Total warnings: 0*



Les éventuelles erreurs sont signalées par un message rouge. Un double clic de la souris sur ces messages permet de localiser l'erreur dans le programme d'application.



L'erreur est marquée en rouge

```
STHj Entree Auto
DYN Dynamise_Entree
DEC Places_Libres
```

Correction de l'erreur

```
STH Entree_Auto
DYN Dynamise_Entree
DEC Places_Libres
```

## 7.8 Charger le programme dans le PCD



Le programme d'application est prêt. Il reste à le transférer de l'ordinateur vers le PCD avec le menu *Online, Download Program* ou le bouton *Download Program* de la fenêtre SAIA Project Manager.

*Download Program*

En cas de problème de communication, veuillez à nouveau contrôler vos configurations *Settings Online* et *Settings Hardware* ainsi que votre câble de communication entre l'ordinateur et le PCD. ([PCD8.K111](#), [USB](#))

## 7.9 Débugger un programme (Debug)

Les programmes ne sont pas toujours parfaits dès la première version. Il est utile de les tester soigneusement. Le test d'un programme est supporté avec le même éditeur que celui employé lors de l'édition.

### 7.9.1 Visualiser le code compilé.



*Show Hide Code*

Le menu *View Code* ou le bouton *Show/Hide Code* permettent de représenter le code sources et le code obtenu après la construction sur une même page IL.

Les lignes blanches représentent le code source originale avec les symboles et commentaires.

Les lignes grises représentent le code qui résulte de la construction (*Build*) avec les adresses des opérands et les numéros de lignes du programme.

```

Parking lot.src
; Bloc d'organisation cyclique
;-----
          COB      0
          COB      0
000006   COB      0
000007   COB      0
000009   NOP

          STH      Entree_Auto      ;Sur le flanc montant du signal
000010   STH      I|0 0
          DYN      Dynamise_Entree
000011   DYN      F 7502
          DEC      Places_Libres    ;décrémenter le compteur de plac
000012   DEC      C 1400

;-----

          STH      Sortie_Auto      ;Sur le flanc montant du signal
000013   STH      I|0 1
          DYN      Dynamise_Sortie
000014   DYN      F 7503
          INC      Places_Libres    ;incrémenter le compteur de plac
000015   INC      C 1400
  
```

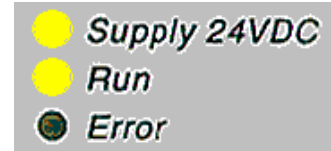
### 7.9.1 Go On/Offline, Run et Stop

Le mode Online permet d'établir la communication avec le PCD pour contrôler le mode de fonctionnement (Run, Stop, Pas-à-pas) et afficher toutes les informations nécessaires au test du programme.

Se mettre en ligne avec le bouton *Go On /Offline*



Forcer l'automate en mode run avec le bouton *Run*



Parallèlement, observer la lampe RUN placée sur la face avant du PCD. A la sélection du bouton *Run*, la lampe RUN est allumée, le PCD exécute le programme utilisateur.

A la sélection du bouton *Stop*, la lampe *RUN* est éteinte, le PCD stop l'exécution du programme utilisateur.



Après le *Stop*, observez la ligne affichée en rouge, elle indique l'instruction sur laquelle le programme s'est arrêté. Le chiffre entre crochet représente la valeur entière du compteur 1400 puis plus à droite, sont affichés les états de l'accumulateur, des indicateurs de statut et du registre d'index.

	DYN	Dynamise_Entree		
000011	DYN	F 7502		
	DEC	Places_Libres	;décrémenter le compteur de places libres	
000012	DEC	C 1400	[8]	A0 Z0 N0 P1 E0 IX0000

### 7.9.2 Mode pas-à-pas



Run to Cursor

Si le PCD est en mode run, marquer la première ligne à observer en mode pas-à-pas est sélectionner le bouton *Run to Cursor*.

Le PCD stop au passage sur la ligne marquée et autorise d'exécuter le programme pas-à-pas en pressant sur la touche F11 ou l'un des boutons présentés ci-dessous.

Si le programme fait appel à des structures PB, FB et SB, il n'est pas toujours nécessaire de les analyser en détail avec le mode pas-à-pas. Il est possible de choisir entre trois options suivantes:



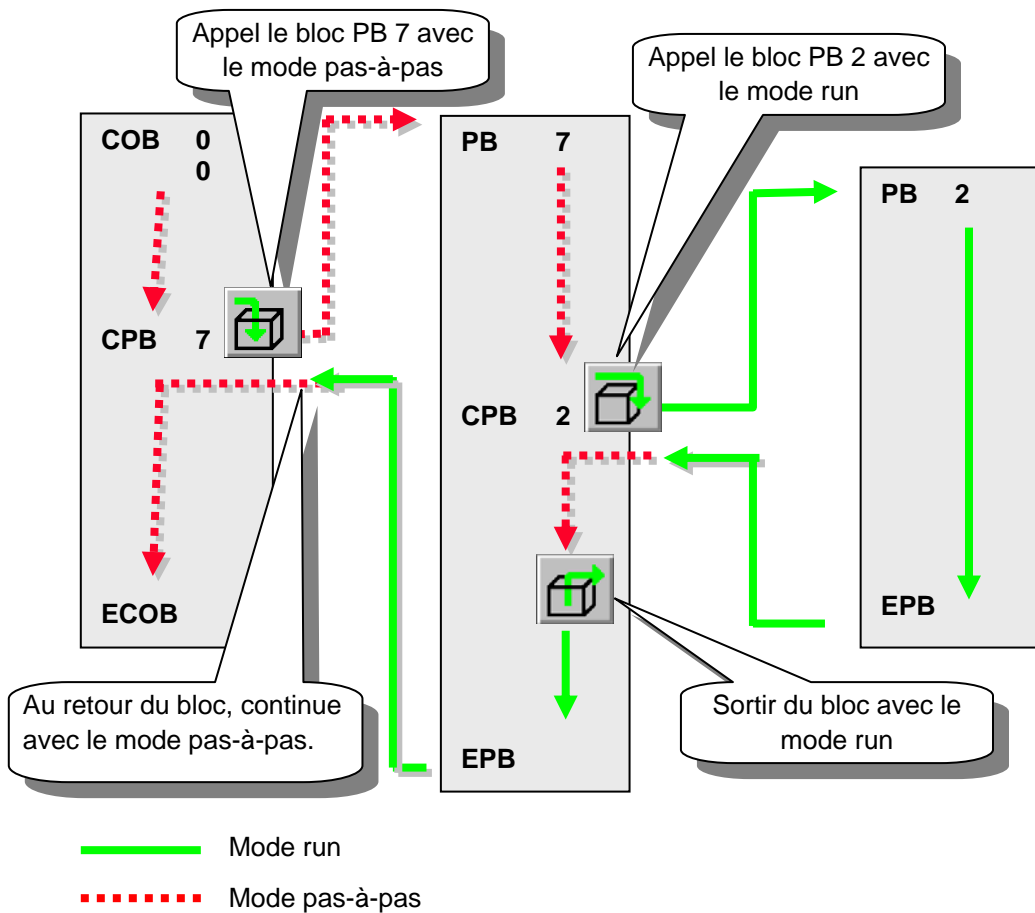
Entrer dans la structure pour l'analyser en pas-à-pas



Traiter la structure appelée avec le mode run et continuer en mode pas-à-pas juste après le retour au bloc qui à effectué l'appel.



Si le programme est entré à l'intérieur d'une structure dont le contenu est sans intérêt. Il est possible d'en sortir rapidement avec le mode run et de continuer en mode pas-à-pas après le retour au bloc qui à effectué l'appel.



```

    STH  Sortie_Auto      ;Sur le flanc montant du signal
000013 STH  I|0 1        [0]
    DYN  Dynamise_Sortie
    
```

Pour chaque pas de programme, observez la ligne affichée en rouge, elle se déplace à la ligne d'instruction suivante, le chiffre entre crochet représente l'état logique de l'entrée I 1, puis plus à droite, sont affichés les états de l'accumulateur, des indicateurs de statut et du registre d'index.

### 7.9.3 Le point d'arrêt

Permet de stopper le programme sur un événement lié à une ligne de programme ou un symbole:

- Etat bas ou haut d'une entrée, sortie, indicateur, indicateur de statut
- Valeur présente dans un registre ou compteur



Set/Clear  
Breakpoint

#### Point d'arrêt sur un symbole

La condition d'arrêt peut être déterminée à l'aide du menu *Online Breakpoints* ou du bouton *Set/Clear Breakpoint*.

Type	Address	Condition	Value
Counter	1400	>	4
Output	32	=	0

A l'aide de la fenêtre ci-dessus, définir le type et l'adresse/numéro du symbole ou simplement glisser un symbole depuis l'éditeur de symbole dans le champs *Symbol Name*, puis déterminer la condition et l'état/valeur d'arrêt.

La sélection du bouton *Set&Run* force le PCD en mode Run conditionnel. La LED *Run* du PCD clignote et le bouton *Run* du PCD alterne les couleurs vert et rouge.

Le PCD se met automatiquement en mode stop lorsque la condition d'arrêt est réalisée. Par exemple lorsqu'une instruction modifie la valeur du compteur 1400 avec une valeur plus grande que 4. La ligne qui suit la dernière instruction traitée par le PCD est marquée en rouge. Il est possible de continuer le traitement du programme en mode pas-à-pas ou avec une autre condition d'arrêt.

Si nécessaire il est possible d'interrompre le mode run conditionnel :

Le bouton *Clear-Run*, force le PCD en mode RUN. La LED *Run* du PCD est allumée et le bouton *Run* du PCD de couleur verte.

Le bouton *Clear-Stop*, force le PCD en mode Stop. La LED *Run* du PCD est éteinte et le bouton *Run* du PCD de couleur rouge.

Si plusieurs point d'arrêts conditionnels ont été saisis, ils sont tous conservés dans le champs *History*. Il est alors possible d'en sélectionner un à l'aide de la souris et de l'activer avec le bouton *Set&Run*.

#### Point d'arrêt sur une ligne programme

La sélection d'une ligne du programme puis du menu *Online, Run to, Cursor* permet de stopper le programme sur la ligne sélectionnée pour en suite continuer en mode pas-à-pas.

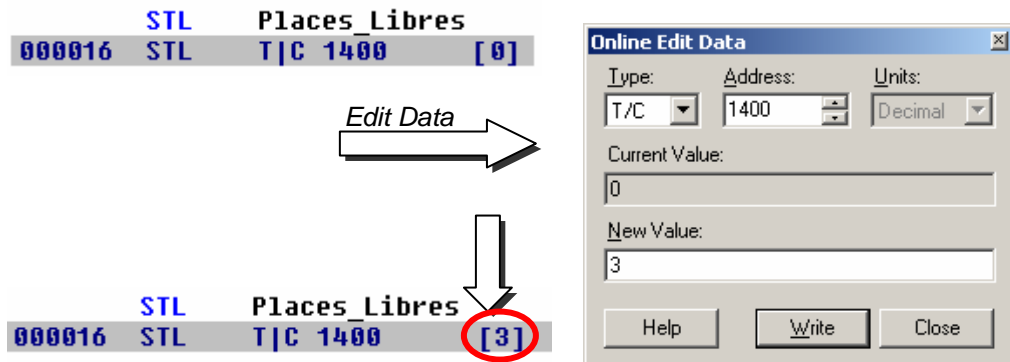


### 7.9.4 Modification de programme en ligne

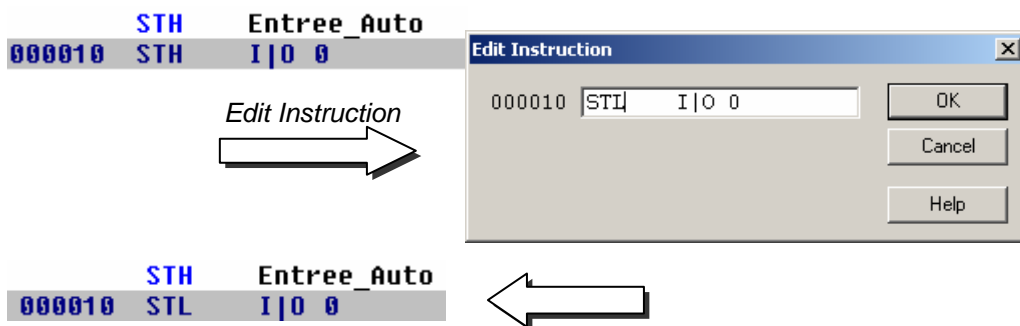
Lors d'un test pas-à-pas de programme, il est utile de modifier les états/valeurs de certaines opérandes/symboles et de vérifier le comportement du programme sous certaines conditions.

Sélectionner une des lignes actives (grise) à l'aide de la souris et presser le bouton droite pour afficher le menu de contexte.

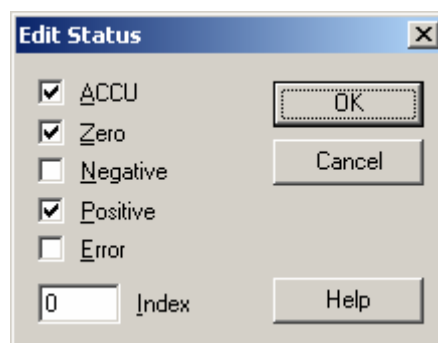
Le menu de contexte *Edit Data* permet de modifier l'état/valeur de l'opérande de l'instruction sélectionnée.



Le menu de contexte *Edit Instruction* permet de modifier le mnémotique et l'adresse de l'opérande correspondant à la ligne d'instruction sélectionnée.



Les indicateurs de statut peuvent aussi être modifiés à l'aide du menu de contexte *Edit Status*.



### 7.9.5 Visualiser et modifier l'état des symboles avec le *Watch window*

Un autre moyen utile pour tester et visualiser l'état des symboles de notre exemple est de faire usage de la fenêtre *Watch window*. Sélectionnez le bouton *Watch window*. Puis glissez les symboles de l'éditeur de symboles vers la fenêtre *Watch window*

1. Placer la souris au centre de l'icône du symbole, presser le bouton gauche

2. Maintenir le bouton de la souris et glisser le symbole dans la fenêtre *Watch window*

3. Symboles avec leurs commentaires et leurs états/valeurs.

Symbol	Address	Value	Modify Value	Symbol Comment
Car_incoming	I 0	0		Gets high when a car comes i...
Car_outgoing	I 1	0		Gets high when a car leaves t...
Red_light	O 32	0		Stops new cars at the entry
Number_of_free_slots	C 1400	8		Counts the number of free p...

#### 4. Start/Stop Monitoring

Pour modifier l'état/valeur d'un des symboles présent dans la fenêtre, procéder comme suit:

1. Start/Stop Monitoring

2. Placer la souris sur la valeur à éditer, presser 2 fois le bouton gauche de la souris et éditer la nouvelle valeur.

3. Download Values

Symbol	Address	Value	Modify Value	Symbol Comment
Car_incoming	I 0	0		Gets high when a car comes i...
Car_outgoing	I 1	0		Gets high when a car leaves t...
Red_light	O 32	0		Stops new cars at the entry
Number_of_free_slots	C 1400	8	5	Counts the number of free p...

Mise en service d'un module analogique

Les instructions de programme présentées jusqu'ici exploitent toutes les entrées et sorties digitales en plaçant leurs adresses ou symboles devant leur mnémonique  
Ex : ANH I 45

Mais avec les entrées et sorties analogiques, il est nécessaire de faire usage d'une routine d'acquisition de la grandeur analogique. Ces routines sont différentes pour chaque type de module analogique et sont décrites dans le manuel hardware de votre PCD.

### 7.9.6 Exemple pour les entrées analogiques PCD2.W340

Si le PCD est équipé d'un module de 8 canaux d'entrées analogiques universelles PCD2.W340, l'utilisateur peut se servir de la routine suivante :

```

BA EQU O 96 ; Adresse de base du module dans le PCD
ACC H ; L'ACCU doit être Haut
LD R 100 ; Défini le canal de mesure ( 0...7)
2

MUL R 100
K 32 ; Calcul de
R 100 ; l'octet de commande OC
ADD R 100 ; inclu le bit
K 264 ; de lancement.
R 100

SET BA+15 ; Lancement conversion A/N

BITO 9 ; Envoi l'octet de commande OC
R 100 ; inclu le bit de lancement
BA+0 ; à W3xx

BITIR 12 ; Lecture des 12 bits de la mesure (0...4095) dans R 77
BA+0
R 77
RES BA+15 ; Arrêt de la conversion A/N

```

Le module PCD2.W340 est universel, il supporte les mesures 0..10V, 0..2.5V, 0..20 mA et les sondes de température Pt/Ni 1000. Un pont doit être sélectionné sur le module pour définir la plage de mesure. La résolution de mesure est de 12 bits, soit 4095 états de mesures distincts.

La routine présentée ci-dessus saisi le canal défini dans le registre 100 et fourni une mesure brute dans le registre 77. Pour ce module d'une résolution de 12 bits, cela correspond à une grandeur de mesure comprise entre 0 et 4095.

C'est à l'utilisateur de convertir la mesure dans une unité physique standard.

### 7.9.7 Exemple pour les sorties analogiques PCD2.W610

Les sorties fonctionnent de manière similaire aux entrées.

Si le PCD est équipé d'un module de 4 canaux de sorties analogiques universelles PCD2.W610, l'utilisateur peut se servir de la routine suivante :

```

BA EQU O 96 ; Adresse de base du module dans le PCD
ACC H ; L'ACCU doit être Haut
LD R 100 ; Défini le canal de sortie ( 0...6)
2
BITOR 2 ; Transfert le canal vers le W6x0
R 100
BA+0
BITOR 2 ; Ecriture de 2 bits de remplissage
R 100
BA+0
LD R 277 ; Défini la valeur numérique de la sortie ( 0...4095)
3879
BITO R 12 ; Transfert les 12 bits de la valeur de sortie vers le W6x0
R 277
BA+0
SET BA+12 ; Lancement de la conversion N/A

```

Un pont doit être sélectionné sur le module pour définir la plage de sortie 0...20 mA ou 0...10 V. La résolution du module est de 12 bits, soit 4095 états de consignes distincts.

La valeur entière appliquée au registre 12 détermine la tension ou le courant de sortie du canal défini dans le registre 100:

Valeur d'entrée du registre 12	Tension de sortie [V]	Courant de sortie [mA]
0	0	0
2047	5	10
4095	10	20



Pour de plus amples informations et le chargement des exemples de programme IL concernant les modules analogiques, veuillez vous référer à votre manuel hardware ou à l'adresse internet : <http://www.sbc-support.ch>



## Table des matières

---

<b>8</b>	<b>OUTILS COMPLÉMENTAIRES</b>	<b>3</b>
<b>8.1</b>	<b>Au sommaire de ce chapitre</b>	<b>3</b>
<b>8.2</b>	<b>Tranfert de données (<i>Data Transfer Utility</i>)</b>	<b>4</b>
8.2.1	Utilisation du transfert de données	4
8.2.2	Démarrer <i>Data Transfert</i>	4
8.2.3	Sauvegarde rapide des données	4
8.2.4	Restaurer les données	5
8.2.5	Sauvegarde à l'aide d'un fichier script	5
8.2.6	Restaurer les données à l'aide d'un fichier script	6
8.2.7	Options de sauvegarde.	6
8.2.8	Sauvegarde de données avec le mode ligne de commande.	7
<b>8.3</b>	<b>Fenêtre de surveillance (<i>Watch Window</i>)</b>	<b>8</b>
8.3.1	Ouvrir une fenêtre <i>Watch Window</i>	8
8.3.2	Ajouter des données à la fenêtre <i>WatchWindow</i>	9
8.3.3	Start /Stop Monitoring	10
8.3.4	Modification en ligne des données	10
8.3.5	Format d'affichage	10
8.3.6	Watch Window et applications avec plusieurs CPU	11
<b>8.4</b>	<b>Le configurateur en ligne (<i>Online Configurator</i>)</b>	<b>12</b>
8.4.1	Le configurateur hors ligne	12
8.4.2	Le configurateur en ligne	12
8.4.3	La fenêtre Online Configurator	12
8.4.4	Ajuster l'horloge PCD	13
8.4.5	L' historique PCD	13
<b>8.5</b>	<b>Programmation d'EPROM</b>	<b>14</b>
<b>8.6</b>	<b>Mise à jour du firmware. (<i>Firmware Downloader</i>)</b>	<b>15</b>
<b>8.7</b>	<b>Menus utilisateurs</b>	<b>16</b>



## **8 Outils complémentaires**

---

### **8.1 Au sommaire de ce chapitre**

Le PG5 met à votre disposition plusieurs utilitaires complémentaires pour assurer divers petits services.



## 8.2 Transfert de données (*Data Transfer Utility*)

### 8.2.1 Utilisation du transfert de données

Cet outil est utilisé pour sauvegarder les états/valeurs des données du PCD dans un fichier ASCII (\*.dt5) ou de les restituer du fichier dans la mémoire PCD.

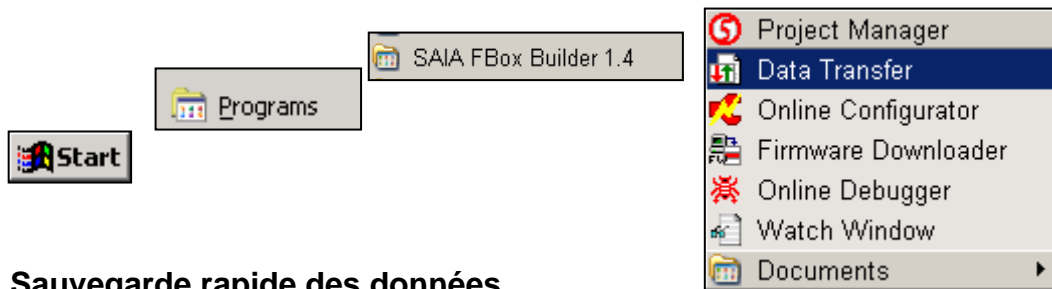
Les données transférées par cet outil sont les suivantes:  
entrées, sorties, indicateurs, temporisateurs, compteurs, registres, blocs de données et textes.

Attention, le programme et les configurations matériels du PCD ne sont pas sauvegardées par l'utilitaire *Data Transfer*. Pour sauvegarder le programme les configurations matériel et les données, il est préférable de faire un backup du programme. Voir description du gestionnaire de projets.

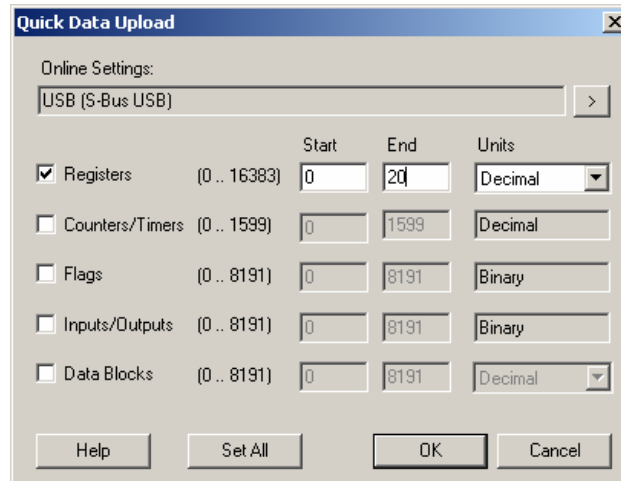
### 8.2.2 Démarrer *Data Transfer*

Démarrer le programme avec le menu:

Start --> Programs --> SAIA FBox Builder 1.4 --> Data Transfer



### 8.2.3 Sauvegarde rapide des données



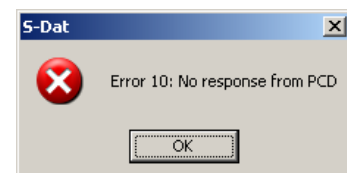
Quick Data Upload

Sélectionner le menu *Online, Quick Data Upload ...* ou le bouton *Quick Data Upload* pour afficher la fenêtre ci-dessus.

Sélectionner les types de données à sauvegarder, les plages d'adresses, éventuellement le format d'affichage pour les registres.

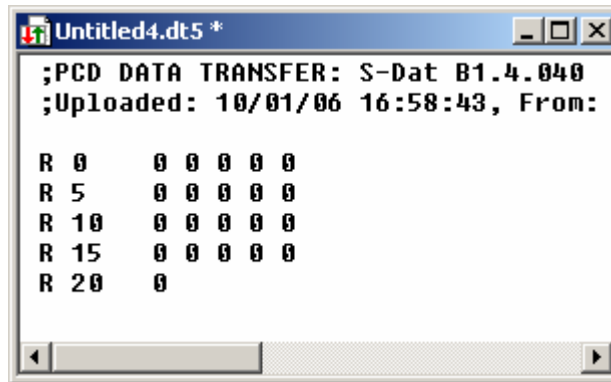
Sélectionner le bouton OK pour charger les données.

Si un message semblable à celui qui suit est affiché, veuillez vérifier les paramètres de communications à



l'aide du menu *Online, Settings Online* et vérifier si le câble PCD8.K111/USB relie correctement le PC au PCD.

Le chargement des données prend quelques instants avant d'être affichés:



```

;PCD DATA TRANSFER: S-Dat B1.4.040
;Uploaded: 10/01/06 16:58:43, From:

R 0      0 0 0 0 0
R 5      0 0 0 0 0
R 10     0 0 0 0 0
R 15     0 0 0 0 0
R 20     0
  
```

Le fichier de données peut être édité avec de nouvelles valeurs puis sauvegardés avec le menu *File, Save* ou le bouton *Save*.

### 8.2.4 Restaurer les données



*Open*

Les fichiers déjà sauvegardés peuvent être à nouveau affichés avec le menu *File, open* ou le bouton *Open*

Si nécessaire, les valeurs du fichier peuvent être éditées par l'utilisateur.

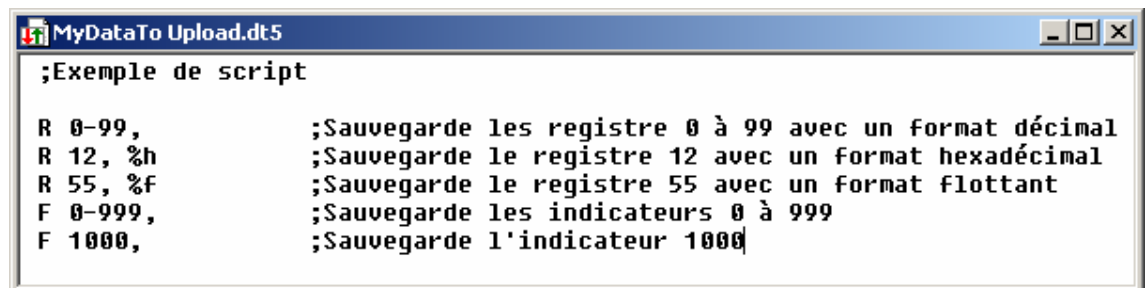


*Download  
To PCD*

Les données sont restaurées dans la mémoire du PCD avec le menu *Online, Download Data To PCD* ou le bouton *Download to PCD*.

### 8.2.5 Sauvegarde à l'aide d'un fichier script

Si nécessaire, la liste des données à sauvegarder peut être éditée dans un fichier script. Exemple:



```

;Exemple de script

R 0-99,           ;Sauvegarde les registre 0 à 99 avec un format décimal
R 12, %h         ;Sauvegarde le registre 12 avec un format hexadécimal
R 55, %F         ;Sauvegarde le registre 55 avec un format flottant
F 0-999,         ;Sauvegarde les indicateurs 0 à 999
F 1000,          ;Sauvegarde l'indicateur 1000
  
```



*Upload  
From PCD*

Sélectionner le menu *Online, Upload Data from PCD ...* ou le bouton *Upload from PCD* pour charger les données du PCD dans une seconde fenêtre distincte de la fenêtre des commandes.

Pour de plus amples informations concernant les commandes disponibles pour le script, veuillez vous référer à l'aide du programme. Voir menu *Help, Help Topics F1, General*.

### 8.2.6 Restaurer les données à l'aide d'un fichier script

Un script offre aussi la possibilité d'éditer les données à restaurer. Exemple:



```
MyDataToDownload.dt5 *
;Exemple de script

R 0-99, 0 ;Charge les registre 0 à 99 avec zéro
R 12, 32h ;Charge le registre 12 avec 32 hexadécimal
R 55, 64.3 ;Charge le registre 55 avec 64.3 flottant
F 0-999, 0 ;Charge les indicateurs 0 à l'état bas
F 1000, 1 ;Charge l'indicateur 1000 à l'état haut
```

Sélectionner le menu *Online, Download Data To PCD ...* ou le bouton *Download To PCD* pour charger les données du script dans PCD.

### 8.2.7 Options de sauvegarde.

La fenêtre affichée avec le menu *Tools, Options* permet d'ajuster le format des données à sauvegarder dans le fichier '\*.dt5'.

Avec les options ci-dessous, le fichier de données peut être facilement importé dans un éditeur *Microsoft Excel*.

The image shows two dialog boxes: "Quick Data Upload" and "Options".

**Quick Data Upload:**

- Registers: (0 .. 4095) Start: 0, End: 2, Units: Decimal
- Flags: (0 .. 8191) Start: 0, End: 6

**Options:**

- Separators: Address separator: TAB, Data separator: TAB
- Values per line: For R / T / C / DB: 2, For I / O / F: 4
- Formatting:
  - Empty line between different data types
  - Type and address at beginning of line
  - Data separator at end of line

The resulting data in "Untitled1.dt5" is:

```
;PCD DATA TRANS
;Uploaded: 11/01/06 08:51:22, From: PCD3.M33x 018, :
R 0 0 0
R 2 0
F 0 0 0 0 0
F 4 0 0 0
```

Callouts explain the format:

- "Type and address at the beginning of line" points to the first column of each line.
- "Empty line between different data types" points to the blank line between the register and flag sections.
- "Address separator" points to the space between the type and address.
- "Data separator" points to the space between data values.
- "Values per line" points to the four data values in the flag lines.

### 8.2.8 Sauvegarde de données avec le mode ligne de commande.

L'outil *Data Transfert* peut aussi être contrôlé à l'aide de lignes de commandes Dos. Cela permet de réaliser des fichiers batches pour sauvegarder périodiquement et automatiquement les données du PCD. Ces données peuvent être, en suite, utilisées par un programme Microsoft Excel ou une banque de donnée, ...

#### Syntaxe d'une ligne de commande:

*SDAT [ Nom\_du\_fichier[.dt5][données...]][/R=nnn][/IOnnn][/A=nnn][/D=nnn]*

Nom\_du\_fichier Nom du fichier à sauvegarder/restaurer

Données... Définition des données à sauvegarder. Si aucune donnée n'est définie, le fichier est restauré dans le PCD

*Format : <type><début>[-<fin>][unités]*

type R,C,O,F,DB

(C= compteurs/temporisateurs, O = entrées/sorties)

début Première adresse

fin Dernière adresse

unités D,H,F (Décimal, hexadécimal, Flottant) pour R,C,DB

/R=nnn nnn = valeur par ligne pour R,T,C,DB ( 1..256, défaut = 5)

/I =nnn nnn = valeur par ligne pour I,O,F ( 1..256, défaut = 10)

/A=nnn nnn = séparateur d'adresses ( TAB,SPACE,COMMA,COLON , défaut= TAB)

/D=nnn nnn = séparateur de données ( TAB,SPACE,COMMA,COLON , défaut= TAB)

Exemple:

sdat MyDatas.dt5 R0-99 R12H R55F F0-999 F1000 /R005 /I010

### 8.3 Fenêtre de surveillance (*Watch Window*)

Le *Watch Window* est un excellent outil de vérification des programmes et des installations. Il permet de visualiser et modifier en ligne toutes les données de l'application.

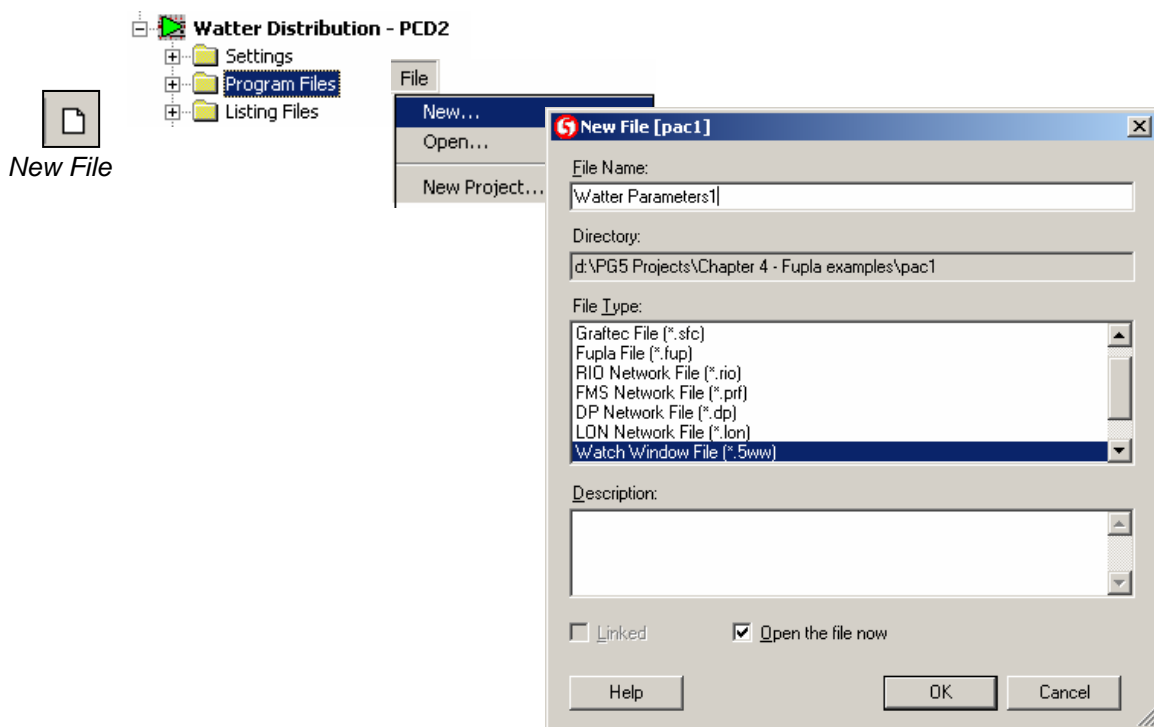
#### 8.3.1 Ouvrir une fenêtre *Watch Window*



*Watch Window*

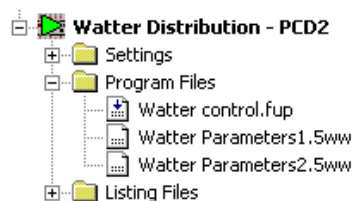
La fenêtre *Watch Window* est affichée à la sélection du menu *Online, Watch Window* ou du bouton *Watch Window*.

Il est aussi possible de préparer plusieurs fenêtres *Watch Window* différentes sous le répertoire *Programme File* du gestionnaire de projet. Ajouter un nouveau fichier de type *Watch Window File (\*.5ww)* avec le menu *File New* ou le bouton *New File*.



Remarque, les fichiers \*.5ww ne sont jamais liés au projet (pas de flèche à l'intérieur de l'icône du fichier). Ces informations n'ont aucune influence sur la construction du programme (*Build*).

Pour ouvrir le fichier \*.5ww, sélectionner le fichier par un double-clic de la souris ou marquer le fichier et sélectionner le menu *File Open*.



### 8.3.2 Ajouter des données à la fenêtre *WatchWindow*

Glisser les symboles du programme ou de l'éditeur de symboles vers la fenêtre *Watch Window*.

1. Placer la souris au centre de l'icône du symbole, presser le bouton gauche

2. Maintenir le bouton de la souris et glisser le symbole dans la fenêtre *Watch window*

3. Symboles avec leurs commentaires et leurs états/valeurs.

4. Start/Stop Monitoring

Symbol	Address	Value	Modify Value	Symbol Comment
HMS	R 2113	103034		PCD Clock with current time
DailyTimer	O 32	1		Daily Timer
Ontime	R 2115	60000		Switch on time
OFFTIME	R 2114	19000		Switch off time

Il est aussi possible de saisir les symboles directement dans la fenêtre:

Editer la nouvelle adresse

Symbol	Address	Value	Modify Value	Symbol Comment
DailyTimer	O 32	1		Daily Timer
ONTIME				

Symbol	Address	Value	Modify Value	Symbol Comment
DailyTimer	O 32	1		Daily Timer
ONTIME	R 2005	60000		Switch on time
	R 2004			

Symbol	Address	Value	Modify Value	Symbol Comment
DailyTimer	O 32	1		Daily Timer
ONTIME	R 2005	60000		Switch on time
OFFTIME	R 2004	19000		Switch off time

### 8.3.3 Start /Stop Monitoring



Start/Stop Monitoring

Le bouton *Start/Stop Monitoring* permet de visualiser les valeurs présentes sur le PCD pour chacun des symboles présent dans le watch Window.

Vérifier que la barre des statuts du Watch Window indique le mode *RUN*. Si nécessaire, forcer le PCD en *RUN* ou *STOP* avec le menu *Online*

### 8.3.4 Modification en ligne des données

La colonne *Modify Value* permet de définir de nouvelles valeurs pour plusieurs symboles et de les charger dans le PCD par la sélection du bouton *Download Values*.

1. Editer de nouvelles valeurs

2. Download Values

3. Les valeurs sont dans le PCD

Symbol	Address	Value	Modify Value
HMS	R 2003	102433	
DailyTimer	O 32	1	
ONTIME	R 2005	60000	83000
OFFTIME	R 2004	19000	180000

Symbol	Address	Value
HMS	R 2003	102816
DailyTimer	O 32	1
ONTIME	R 2005	83000
OFFTIME	R 2004	180000

### 8.3.5 Format d'affichage

Le format d'affichage des valeurs peut être ajusté selon les besoins à l'aide du

**Exemple:** Affichez le registre R 2004 en hexadécimal

Address	Value
O 32	0
R 2003	113245
R 2004	182000
R 2005	53000

- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Insert Line Ins
- Delete Line Del
- Move Up Ctrl+Up
- Move Down Ctrl+Down
- Units
- Show

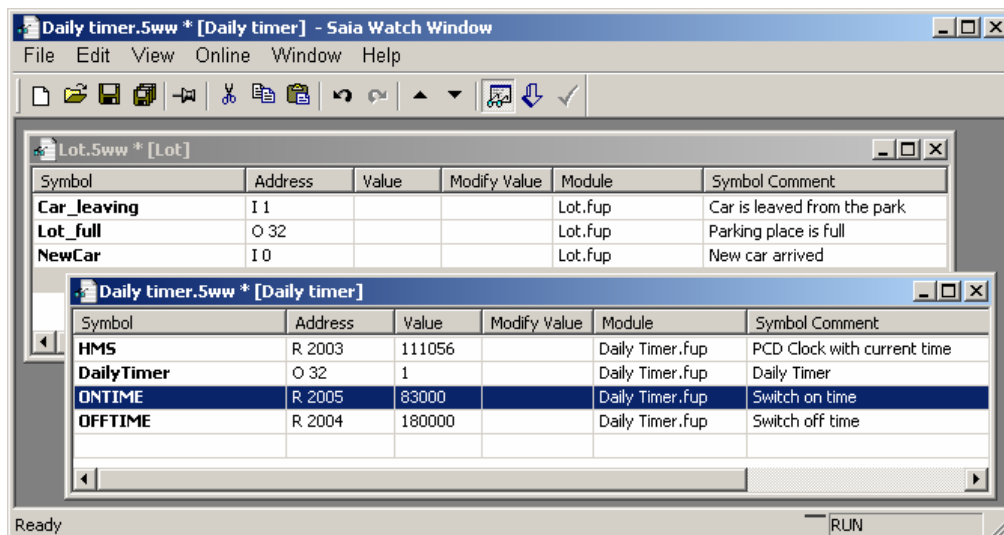
- Decimal Ctrl+D
- Floating Point Ctrl+F
- Hex Ctrl+H
- Binary
- ASCII

Address	Value
O 32	0
R 2003	113245
R 2004	0002C6F0H
R 2005	53000

### 8.3.6 Watch Window et applications avec plusieurs CPU

Le Watch Window permet d'ouvrir plusieurs documents simultanément. Le menu, la barre d'outil et le statut se rapportent toujours à la fenêtre active. Fenêtre avec la barre de titre marquée en bleu.

Chaque document Watchwindow ouvert fait par défaut usage des *Online settings* du CPU au quel il appartient. Il est ainsi possible de visualiser sur le réseau de communication, les données des différents PCD disponibles dans le projet.



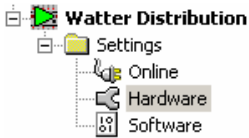


## 8.4 Le configurateur en ligne (*Online Configurator*)

Le PG5 dispose de deux outils de configuration:

Le configurateur hors ligne disponible depuis les *Settings Hardware* du gestionnaire de projet.

Le configurateur en ligne disponible avec le menu Tools, *Online Configurator* ou le bouton *Online Configurator*



### Le configurateur hors ligne

Permet de configurer la mémoire, les paramètres de communication et le mot de passe du PCD. Ces informations sont sauveées dans un fichier du projet PG5. L'utilisateur doit forcer le chargement des configurations dans la mémoire de l'automate avec un bouton *Download*.

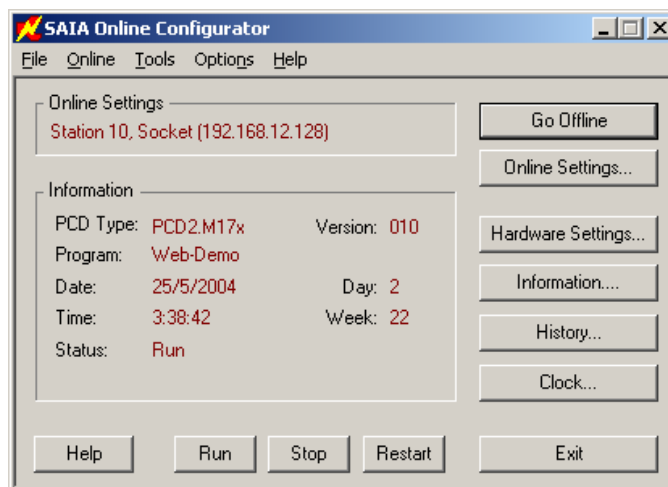
### 8.4.2 Le configurateur en ligne



*Online Configurator*

Permet de configurer la mémoire, les paramètres de communication et le mot de passe du PCD. Mais ces informations sont directement inscrites dans la mémoire de l'automate. Aucune trace de ces informations n'est conservée dans le projet PG5. Sans automate, il n'est pas possible de prendre connaissances de ces informations. Il est donc préférable d'utiliser le configurateur en ligne pour vérifier les données du PCD et de les configurer avec les *Settings Hardware*.

### 8.4.3 La fenêtre Online Configurator



<i>PCD type</i>	Numéro d'article du PCD
<i>Version</i>	Version du firmware PCD
<i>Program Name</i>	Nom du programme utilisateur
<i>Date</i>	Date de l'horloge PCD (Si pas d'horloge : 1/1/92)
<i>Time</i>	Heure de l'horloge PCD
<i>Day</i>	Jour de la semaine : 1 = lundi, ... 7 = dimanche
<i>Week</i>	N° de la semaine de l'année
<i>Status</i>	Mode d'exploitation : Run, Stop, Halt, Conditional
<i>Run</i>	
<i>Onlinesettings</i>	Connexion directe PGU ou S-BUS

Si les informations représentées en rouge ne sont pas affichées ou qu'un message dialogue *No response* est affiché, cela signifie que la communication ne peut pas être établie entre le PCD et le *Online Configurator*.

Veuillez alors vérifier:

Si l'ordinateur est correctement raccordé au PCD avec le câble PCD8.K111/USB

Si les paramètres de communications sont correctement sélectionnés avec le bouton *Settings*

Hardware Settings...

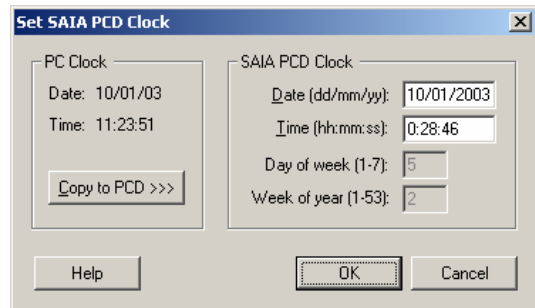
Configuration de la mémoire PCD et des paramètres de communication. Ce sont les mêmes paramètres que ceux déjà présents dans les *Hardware Settings* du gestionnaire de projet.

### 8.4.4 Ajuster l'horloge PCD

Clock...

Sélectionnez le bouton *Online configurator* de la fenêtre *SAIA Project Manager*. Puis le bouton *Clock*

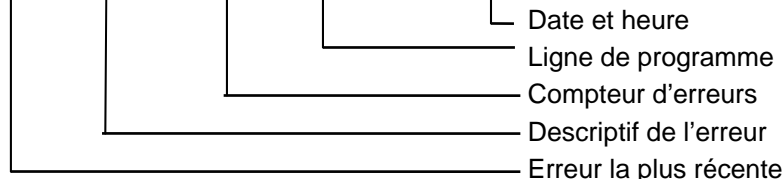
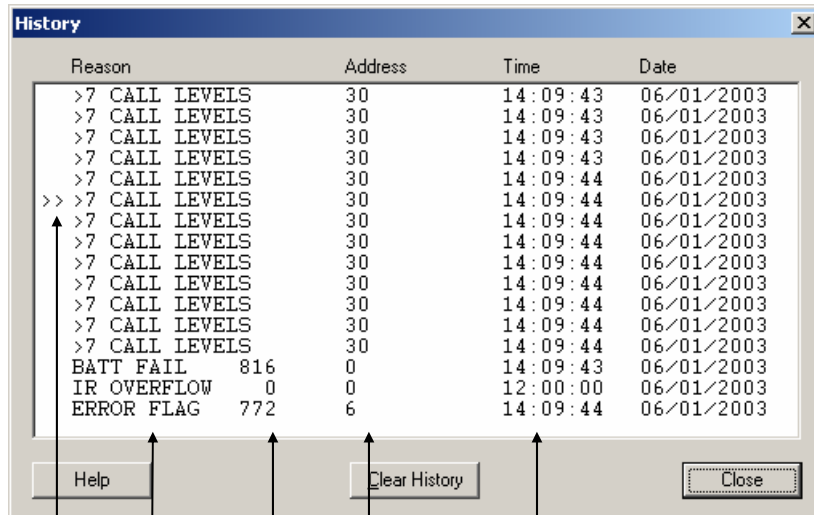
1. Copier l'heure du PC dans l'automate avec le bouton *Copy to PCD >>>* ou ajuster l'horloge dans les champs *SAIA PCD Clock*



### 8.4.5 L' historique PCD

Hjstory...

L'historique protcole toutes les erreurs matérielles ou logicielles survenues à l'exploitation du PCD. Cette table est mise à jour en permanence, même si les XOB ne sont pas programmés. A consulter lorsque la lampe *Error* du CPU est allumée.



Remarques:

Chaque unité centrale a son historique.

L'erreur *BATT FAIL* n'existe que sur l'unité centrale 0.

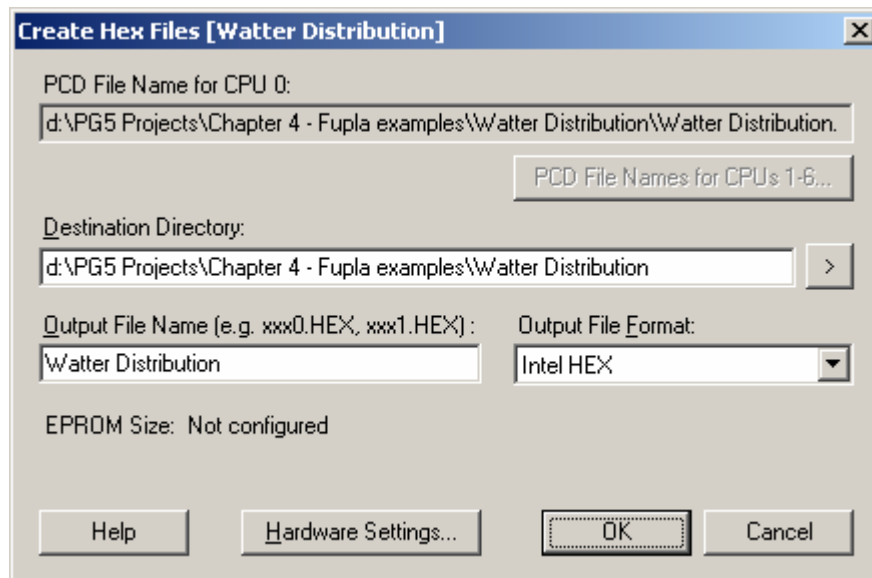
Si l'erreur est imputable à une ligne de programme, celle-ci est précisée. Si non, elle est affichée en hexadécimal.

Le XOB 0 n'apparaît que s'il est programmé.

## 8.5 Programmation d'EPROM

PG5 supporte la création de fichiers binaires ou hexadécimal pour tous les types de programmeur d'EPROM standards rencontrés sur le marché.

Dans le gestionnaire de projets, sélectionner le menu *CPU, Create Hex Files ...*



Pour créer un fichier EPROM:

- Vérifier les *Hardware Settings*
- Construire le programme. (*Build*)
- Choisir le format du fichier de sortie *Output File Format*
- Définir le répertoire et le nom du fichier de sortie sous *Destination Directory* et *Output File Name*
- Bouton *OK*

### Précisions techniques

L'EPROM ne contient pas seulement le programme PCD, mais aussi les *Hardware Settings*. A la mise sous tension, les *Hardware Settings* sont automatiquement copiés de L'EPROM sur le PCD. Mais seulement si le PCD a perdu ces informations (Suite à une coupure d'alimentation cumulée à la défectuosité/absence de batterie).

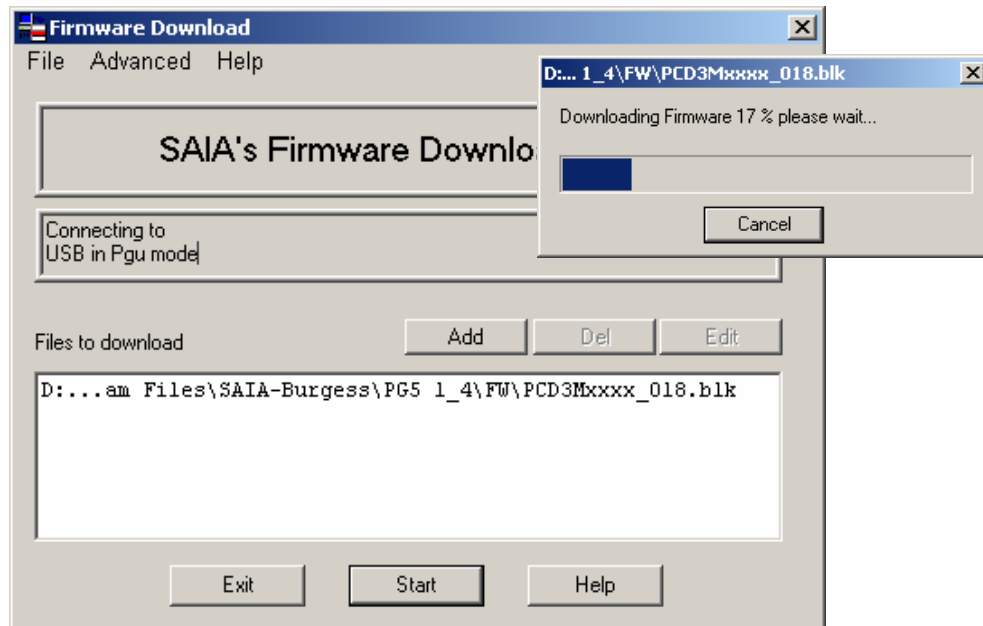
Au montage de l'EPROM, ne pas oublier de positionner correctement les ponts mémoire du PCD.

## 8.6 Mise à jour du firmware. (*Firmware Downloader*)

Parfois le programme firmware doit être mis à jour pour bénéficier des dernières nouveautés du produit PCD.

Pour la majorité des automates, le firmware peut être mis à jour par un échange d'EPROM.

Seul les firmwares des PCD les plus récents<sup>1)</sup> peuvent être rechargés en mémoire flash avec l'aide d'un petit utilitaire disponible avec le menu *Tool, Firmware downloader* du gestionnaire de projet.



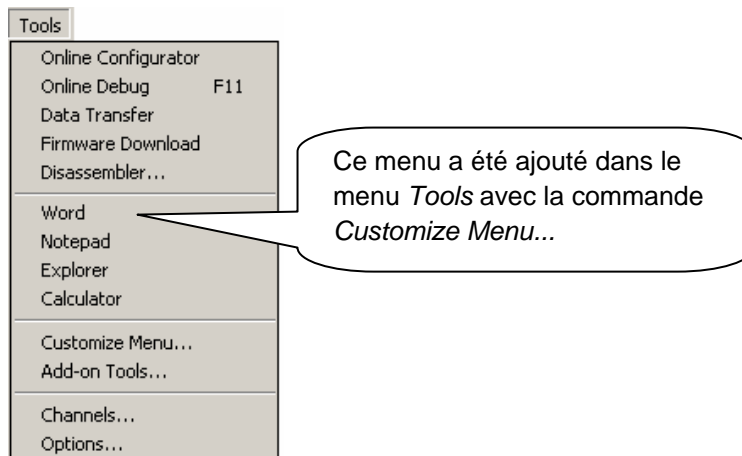
Marche à suivre:

- Le bouton *ADD* permet d'ajouter un nouveau fichier firmware (\*.blk) dans la liste: *Files*.
- Les fichiers firmwares les plus récents sont disponibles dans répertoire *FW* du CD de distribution PG5.
- Utiliser le menu *File, Settings* pour ajuster les paramètres de communications avec le mode PGU. (seul mode actuellement supporté)
- Sélectionner le firmware à charger dans le PCD
- Raccorder le câble PCD8.K111 à l'interface PGU du PCD
- Mettre le PCD hors tension -> sous tension
- Avec les PCD2.M480, manipuler deux fois le bouton *Run/Halt* alors que la led *Run* clignote encore.
- Charger le firmware avec le bouton *Start*, une fenêtre de dialogue indique la progression du transfert de données.
- A la fin du transfert de données, les leds *Run*, *Halt* et *Error* du PCD se mettent à clignoter. Le PCD réorganise certaines informations en mémoire. Veuillez attendre encore une minute avant de mettre l'automate hors tension ou de continuer votre travail.

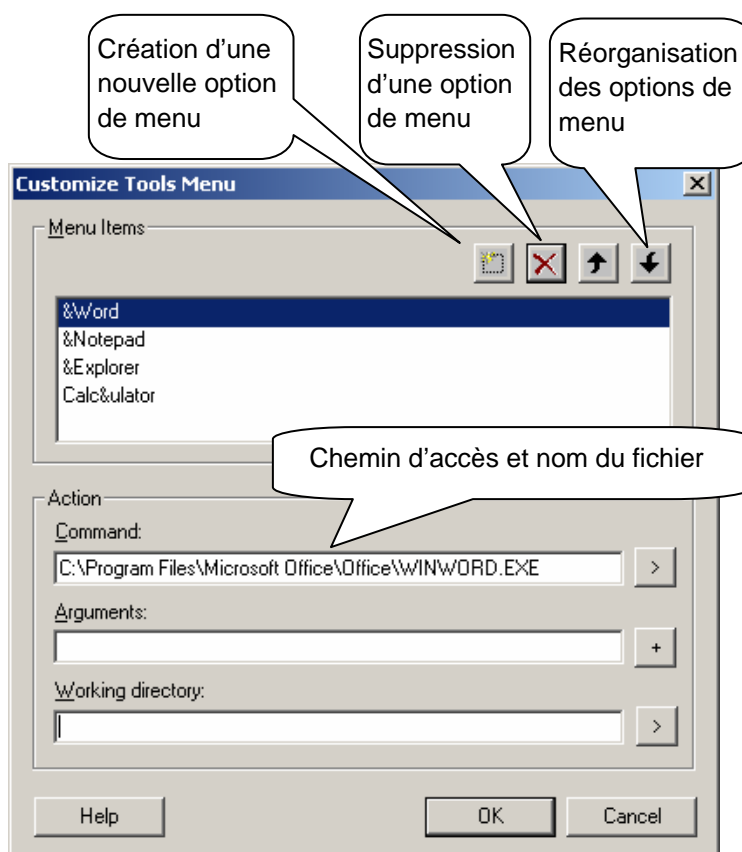
<sup>1)</sup> PCD2.M170, PCD4.M170, PCD2.M480

## 8.7 Menus utilisateurs

Le menu *Tools* de la fenêtre *SAIA project Manager* peut être étendu avec des raccourcis vers vos programmes favoris.



Pour ajouter un raccourci, procéder comme suit:



**Table des matières**

---

<b>TABLE DES MATIERES</b>	<b>1</b>
<b>9 SAIA NETWORKS (S-NET)</b>	<b>2</b>
9.1 Au sommaire des prochains chapitres	2
9.2 Choix du réseau	3
9.2.1 Services supportés	3
9.2.2 Caractéristiques techniques	3

## 9 Saia Networks (S-Net)

### 9.1 Au sommaire des prochains chapitres

Les concepts d'automatisations sont souvent composés de plusieurs automates PCD, terminaux et ordinateurs de supervision décentralisés et raccordés par un réseau de communication. Chaque station sur le réseau contrôle une partie de l'ensemble du processus automatisé et échange les informations nécessaires avec les autres stations disponibles sur le réseau.

Pour garantir la flexibilité nécessaire d'un tel concept, les systèmes PCD supportent l'accès à de nombreux réseaux de communications, chaque réseau à ses particularités et l'utilisateur porte son choix sur le réseau qui convient le mieux à son application.

PG5 est un outil de mise en service de concepts de communications très efficace :

- Le *Saia Project Manager* donne une vue de l'ensemble des PCD d'un projet d'application et de leurs paramètres de configurations dont les paramètres des réseaux de communication.
- L'éditeur Fupla ou IL permettent de définir les échanges de données entre les stations PCD sur le réseau.

Les exemples de programme décrit dans les chapitres qui suivent sont tous installés avec PG5 et vous servent de base pour tester et comprendre le fonctionnement des échanges de données entre les différents réseaux PCD. Vous remarquerez que certains d'entre eux sont très similaires à mettre en services.

## 9.2 Choix du réseau

Les systèmes PCD supportent de nombreux types de réseaux de communications, chaque réseau à ses particularités et l'utilisateur porte son choix sur le réseau qui convient le mieux à son application.

Voici un aperçu des différents réseaux S-Net disponibles:

- Profi-S-Bus : réseau de terrain basé sur le standard Profibus FDL
- Ether-S-Bus : réseau informatique basé sur le standard Ethernet
- Serial S-Bus : réseau basé sur une interface série RS 485/232
- S-Bus Modem : réseau basé une ligne téléphonique analogique ou numérique.
- Profi-S-IO : réseau de terrain basé sur le standard Profibus DP
- Profibus DP: réseau de terrain basé sur le standard Profibus DP

Ces différents réseaux se distinguent par leur services, caractéristiques techniques et domaines d'application :

### 9.2.1 Services supportés

Bien que tous les réseaux de communications sont en mesure d'assurer le transport des données PCD tel que les entrées, sorties, indicateurs, registres, ...

Certains d'entre eux offrent d'avantage. Ils supportent encore PG5 pour la programmation et la mise en services des systèmes PCD au travers du réseau de communication.

### 9.2.2 Caractéristiques techniques

#### 9.2.2.1 Vitesse de communication

La vitesse de communication donne une information indicative du temps de réaction nécessaire pour transférer des données d'un PCD à l'autre. Si la quantité de données à échanger est importante ou le temps de réaction doit être court, alors la vitesse de communication devrait être élevée. Remarque, si la vitesse de communication d'un réseau est ajustable, elle doit être identique pour toutes les stations du réseau.

#### 9.2.2.2 Distance maximum franchissable

La distance maximum peut être une limitation pour les installations PCD très éloignées les unes des autres. Pour chaque réseau, la distance maximum ne peut être franchie sans un système d'amplification des signaux électriques tel un répéteur ou un switch / Hub. En règle générale, la distance maximum est aussi dépendante de la vitesse de communication. Plus elle est élevée, plus la distance maximum est courte. Diminuer la vitesse de communication peut être une solution pour franchir de plus grandes distances !

#### 9.2.2.3 Protocole de communication

Le protocole c'est la séquence d'informations utilisée pour échanger des données entre deux stations du réseau de communication. Nous pouvons comparer le protocole à la langue utilisée par deux personnes qui parlent ensemble. Si les deux personnes parlent le français, elles se comprennent parfaitement bien et c'est aussi valable pour les protocoles de communication. Deux stations du réseau ne peuvent échanger des données que si elles utilisent le même protocole.

Les protocoles de certains réseaux de communication sont normalisés. C'est un grand avantage lorsqu'il est nécessaire de faire communiquer ensemble les



équipements provenant de divers fournisseurs. Les bus de terrains ou de capteurs sont souvent basés sur un protocole normalisé : Profibus DP

Sur certains réseaux de communication comme Ethernet ou Profibus FDL il est possible de supporter sur le même réseau des échanges de données basés sur des protocoles différents. Mais dans tous les cas les deux stations concernées par l'échange de données utilisent le même protocole.

#### **9.2.2.4 Mode d'échange des données maître-esclaves ou multi-maîtres.**

Un réseau de communication maître-esclaves est composé d'une seule station maître et de plusieurs stations esclaves. Le maître est la seule station autorisée à échanger des données avec les stations esclaves.

Un réseau de communication multi-maîtres est composé de plusieurs stations maîtres et esclaves. Chaque station maître est autorisée à échanger des données avec les autres stations maîtres, esclaves.

L'échange de données entre les esclaves n'est jamais autorisé.

#### **9.2.2.5 Domaine d'application**

Certains réseaux sont uniquement prévus pour un domaine d'application distinct. Par exemple, Profibus DP est un réseau orienté vers le domaine des machines. Le protocole de ces réseaux étant parfaitement bien normalisé, de nombreux équipements provenant de divers fournisseurs du marché permettent d'assurer des fonctions déportées sur ce bus tel que des commandes de moteurs, ...

Le réseau Ether-S-Bus est peut-être plus indiqué pour faire communiquer les équipements PCD avec des ordinateurs équipés d'un système de supervision, un serveur OPC ou tout simplement d'un outil de programmation et de mise en service PG5.

Le serial S-Bus est aussi très intéressant pour mettre en réseau le matériel PCD. C'est un réseau très économique qui autorise les mêmes services que Ether-S-bus sur les réseaux RS 485 mais aussi au travers de lignes téléphoniques analogique et ISDN (S-Bus Modem)

Le nouveau réseau Profi-S-Bus réunit tous les avantages d'un réseau multi-maîtres et d'une vitesse de communication élevée sur un réseau de terrain destinés aux applications d'automatisation industrielles.

## Réseaux de communication S-Net

Services :	Ether-S-Bus	Profi-S-Bus	Serial S-Bus	S-Bus Modem	Profi-S-IO Profibus DP
Programmation de PCD	oui	oui	oui	oui	non
Echange de données	oui	oui	oui	oui	oui
<b>Caractéristiques</b>					
Vitesse de transmission max	10 et 100 Mbd	12 Mbd	38.4 /115.2 Kbd	38.4 /115.2 Kbd	12 Mbd
Distance maximum sans répéteurs ou switsch/Hub	100 m	100 m	1200 m		100 m
Type de câble	4 paires torsadés	1 paire torsadé	1 paire torsadé		1 paire torsadé
Protocole	Saia	Saia	Saia	Saia	Normalisé ISO
Mode d'échange	Multi-Maîtres	Multi-Maîtres	Maître-esclaves	Multi-Maîtres	Maître-esclaves
Nombre maximum de stations	Non limité	126	256	Non limité	126
<b>Domaine d'application</b>	Industrie, bâtiment	Industrie, bâtiment	Industrie, bâtiment	Industrie, bâtiment	Industrie, bâtiment



## Table des matières

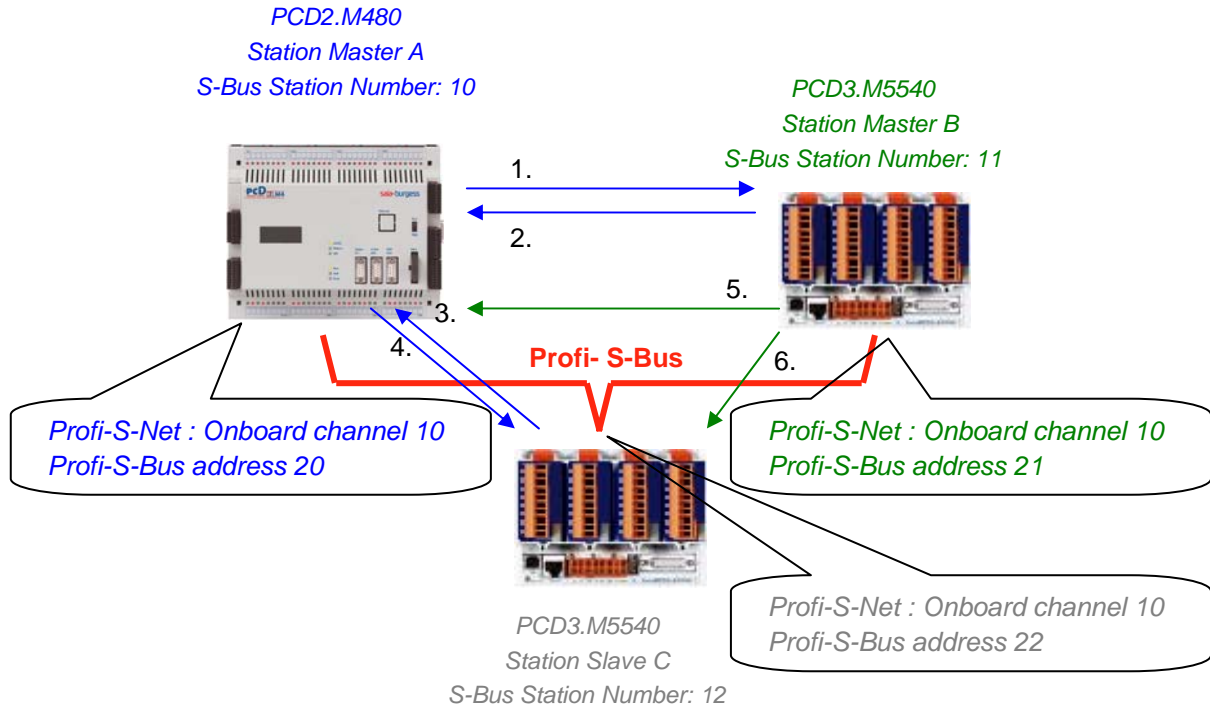
---

<b>TABLE DES MATIERES</b>	<b>1</b>
<b>10 PROFI-S-BUS</b>	<b>2</b>
10.1 Exemple de réseau Profi-S-Bus.	2
10.2 Exemples d'échanges de données Profi-S-Bus.	2
10.3 Le Projet PG5	3
10.4 Hardwares Settings maîtres, esclaves	3
10.4.1 Définir les paramètres PCD	3
10.4.2 Définir le numéro de station S-Bus sur le réseau	4
10.4.3 Définir le canal de communication Profi-S-Bus	4
10.4.4 Charger les Hardwares Settings sur le CPU	5
10.5 Programme Fupla	5
10.5.1 Assigner le canal à l'aide d'une Fbox SASI	5
10.5.2 Assigner le canal maître	6
10.5.3 Assigner le canal esclave	6
10.5.4 Principe d'échanges des données sur un réseau multi-maîtres.	6
10.5.5 Echange de données entre les stations maîtres et esclaves	7
10.5.6 Diagnostiques	8
10.6 Programme II	11
10.6.1 Assigner le canal maître à l'aide d'une instruction SASI	11
10.6.2 Assigner le canal esclave	11
10.6.3 Principe d'échanges des données sur un réseau multi-maîtres.	11
10.6.4 Echange de données entre les stations maîtres et esclaves	12
10.6.5 Diagnostiques	13
10.7 Fonction Gateway	15
10.7.1 Application	15
10.7.2 Configuration de la fonction Gateway PGU	16
10.7.3 Configuration d'un Gateway Slave port esclave supplémentaire	18
10.7.4 Timings de communication	19
10.8 Autres références	20

# 10 Profi-S-Bus

Cet exemple présente comment échanger quelques données comme des registres et indicateurs entre les PCD raccordés sur le réseau Profi-S-Bus.

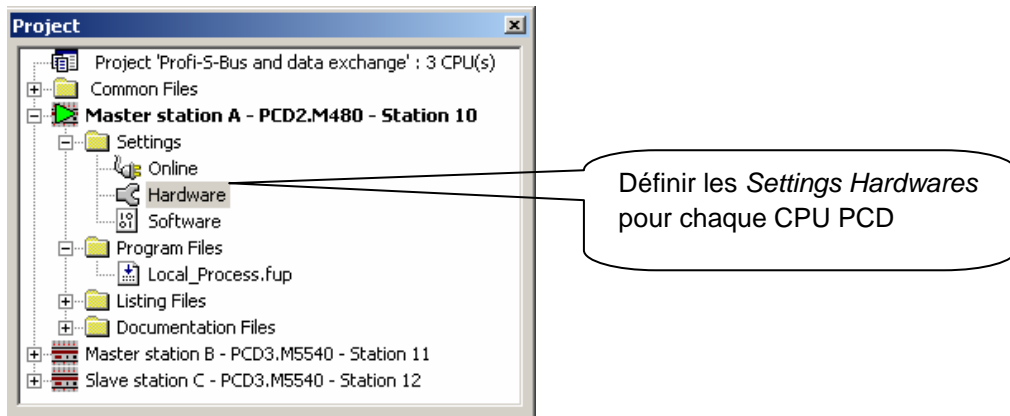
## 10.1 Exemple de réseau Profi-S-Bus.



## 10.2 Exemples d'échanges de données Profi-S-Bus.

	<b>Master with data exchanges</b>	<b>Data on the network</b>	<b>Passive master or slave</b>
	<b>Master station A</b>		<b>Master station B</b>
1	Blinker0 .. 7 F 0 .. 7	Write 8 flags in the Master station B	Station_A.Blinker0 .. 7 F 100 .. 107
2	Master_B.Value100 R 125	Read 1 register in the Master station B	Value100 R 25
			<b>Slave station C</b>
3	Slave_C.Binary0 .. 7 F 100 .. 107	Read 8 flags in the slave station C	Binary0 .. 7 F 20 .. 27
4	Value0 .. 5 R 0 .. 5	Write 6 registers in the slave station C	Master_A.Value0 .. 5 R 20 .. 25
	<b>Master station B</b>		<b>Master station A</b>
5	Temperature1 .. 4 Dynamic registers	Write the temperature measures to the slave C	Master_B.Temperature1 .. 4 R 100 .. 104
			<b>Slave station C</b>
6	Temperature1 .. 4 Dynamic registers	Write the temperature measures to the master A	Master_B.Temperature1 .. 4 R 100 .. 104

### 10.3 Le Projet PG5

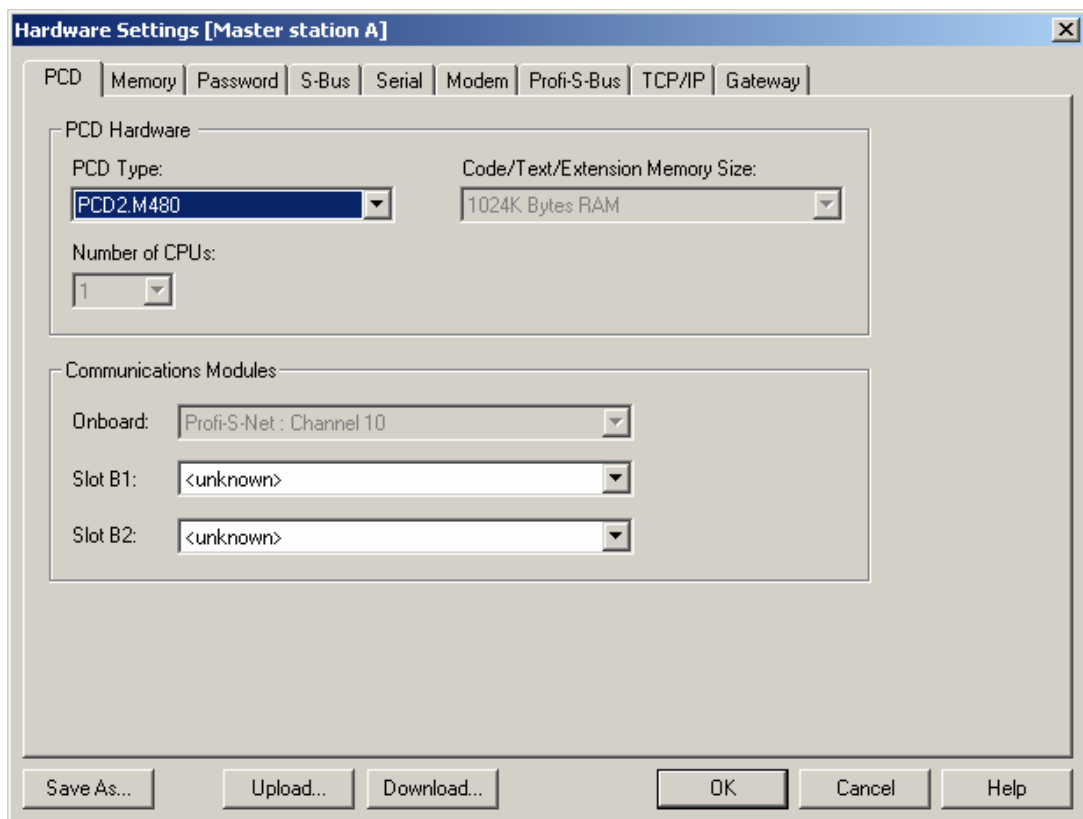


Le *Saia Project Manager* donne une vue de l'ensemble des PCD d'un projet d'application ainsi que des paramètres pour les réseaux de communication. Nous commencerons par ajouter un CPU dans le projet pour chaque station disponible sur le réseau.

### 10.4 Hardwares Settings maîtres, esclaves

Les configurations des *Hardwares Settings* d'une station maître et esclaves sont semblables.

#### 10.4.1 Définir les paramètres PCD

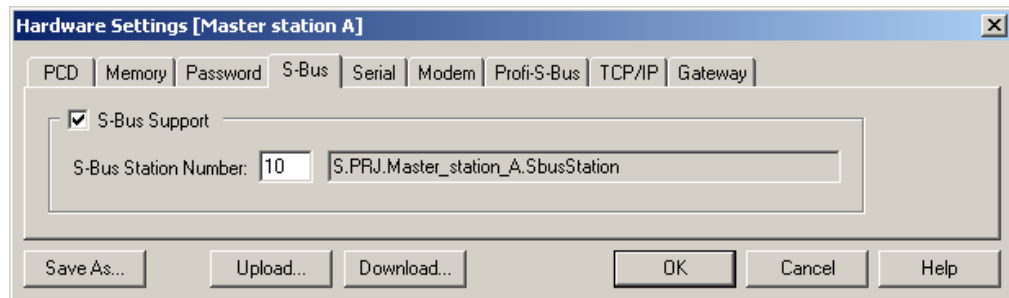


**PCD Type**

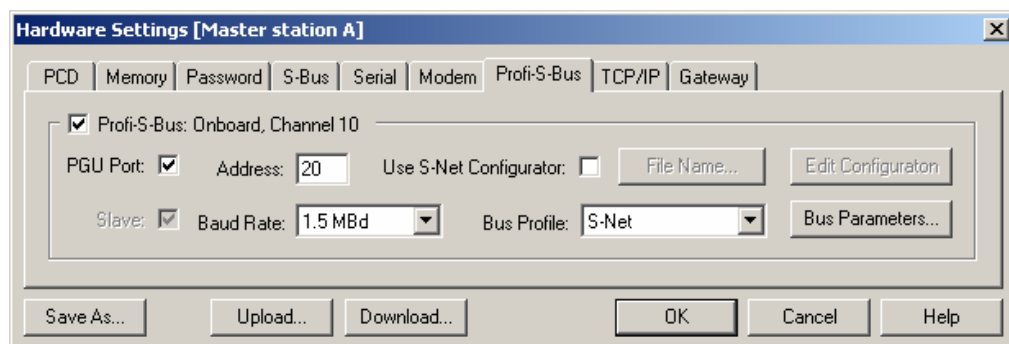
Définir le type de CPU

**Communication Modules**

Si nécessaire indiquer le type des modules de communications insérés dans les slot B1 et B2 du PCD2.M480.

**10.4.2 Définir le numéro de station S-Bus sur le réseau****S-Bus Station Number**

Le numéro de station S-Bus est commun à tous les canaux de communication du PCD

**10.4.3 Définir le canal de communication Profi-S-Bus****Address**

Numéro de station Profi-S-Bus lié au canal

**PGU Port ou Slave**

Définit le canal comme slave ou PGU. Cette définition peut être cumulée avec la fonction master en ajoutant une Fbox SASI master dans le programme Fupla.

**Esclave PGU**

Supporte l'échange de donnée avec les stations maîtres, systèmes de supervision et terminaux. Mais supporte aussi l'outil de programmation et de mise en service PG5..

**Esclave**

Supporte uniquement l'échange de donnée avec les stations maîtres, systèmes de supervision et terminaux.

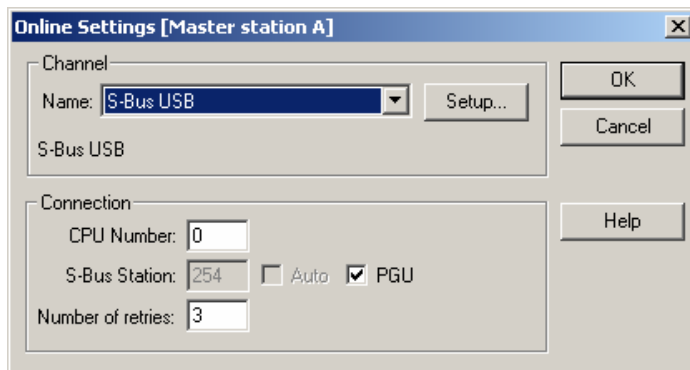
**Baud Rate**

Vitesse de communication, doit être identique sur toutes les stations du réseau.

**S-Bus Profile**

Les timings pour la transmission sont regroupés dans trois profils : S-Net, DP ou utilisateur. Avec le profile utilisateur le bouton *Bus Parameter* permet de définir ses propres *timings*. Le Profile doit être identique sur toutes les stations du réseau. Le profile S-Net est nécessaire lors de l'usage de RIO PCD3.T76x sur les réseaux.

**10.4.4 Charger les Hardwares Settings sur le CPU**

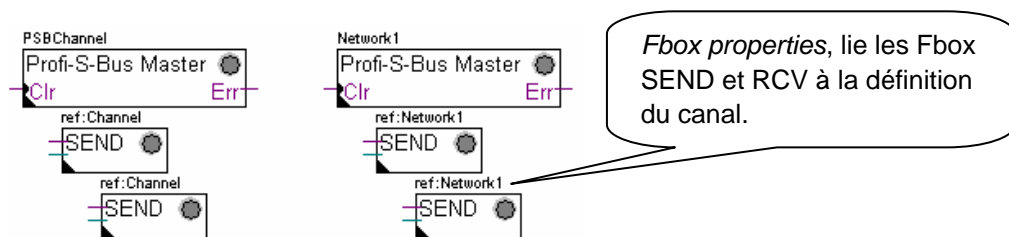


Avec les nouveaux systèmes PCD2.M480 et PCD3, les *Hardware Settings* peuvent être chargés sur les PCD2.M480 et PCD3 au travers de l'interface USB. Il faut juste vérifier que les *Online Settings* soient définis avec un canal *Profi-S-Bus PGU*.

Charger les paramètres dans le PCD à l'aide du bouton *Download* présent dans la fenêtre *Hardware Settings*.

**10.5 Programme Fupla**

**10.5.1 Assigner le canal à l'aide d'une Fbox SASI**



L'assignation du canal est assurée à l'aide d'une FBox SASI placée au début du fichier Fupla. Chaque réseau de communication dispose de sa propre Fbox SASI, car les paramètres d'assignation sont différents d'un réseau à l'autre, de même pour une station esclave ou maître.

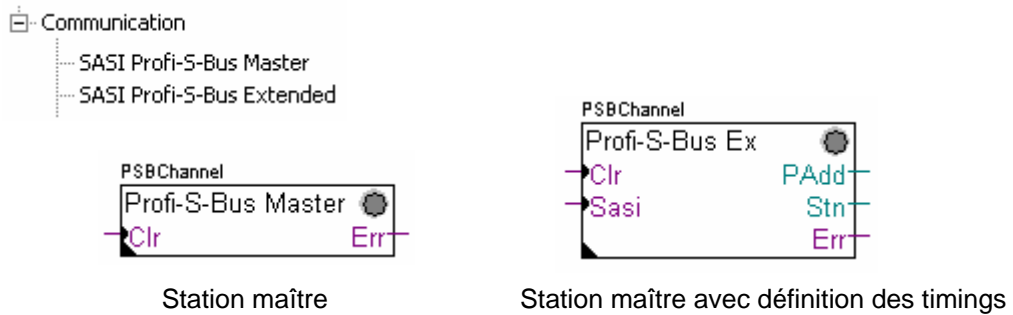
Si le PCD exploite plusieurs canaux de communications, définir chacun des canaux à l'aide de la Fbox SASI appropriée. Puis placer la souris sur la Fbox SASI et sélectionner le menu de contexte *Fbox properties*, définir un *Name* différent pour chaque Fbox selon le canal utilisé. Ce nom permet de lier diverses Fbox d'échanges SEND et RCV avec la Fbox SASI correspondant au canal utilisé.



Selon les réseaux, les paramètres du canal de communication peuvent être partiellement définis à partir de la fenêtre d'ajustage de la Fbox SASI et complétés dans les *Hardwares Settings*.

Mais le numéro du canal de communication est toujours défini à l'aide de la fenêtre d'ajustage de la FBox SASI. Le numéro du canal dépend du Hardware PCD et du hardware de communication utilisé : slot B1, B2, interface série PCD7.F, ...

### 10.5.2 Assigner le canal maître



L'assignation du canal maître est assurée en complétant les *Hardwares Settings* avec l'une des Fbox ci-dessus.

Seul le canal de communication et les timings du canal maître peuvent être ajustés par la Fbox. Les autres paramètres sont tous définis dans les *Hardwares Settings*.

#### Paramètres de la fenêtre d'ajustage:

##### **Channel**

Définit le canal correspondant à l'interface sérielle raccordée au réseau. Dépend du PCD et de son hardware.

##### **Timing**

Le Timeout est en général défini avec la valeur par défaut (0) et n'est ajusté que pour les applications particulières (Gateway)

### 10.5.3 Assigner le canal esclave

Aucune FBox SASI n'est nécessaire pour la station esclave d'un réseau Profi-S-Bus. Toutes les définitions nécessaires sont déjà présentes dans les *Hardwares Settings*.

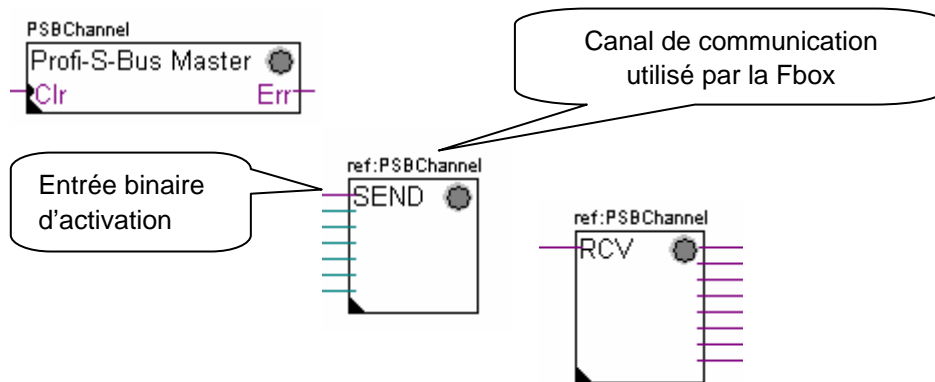
### 10.5.4 Principe d'échanges des données sur un réseau multi-maîtres.

Le réseau de communication multi-maîtres est composé de plusieurs stations maîtres et esclaves. Les stations maîtres sont les seules stations autorisées à lire ou écrire des données des autres stations maîtres mais aussi esclaves. L'échange de données entre les esclaves n'est pas autorisé.

Avec un mode de communication multi-maîtres, les échanges de données effectués sur le réseau sont répartis entre plusieurs maîtres. Mais à un instant donné, un seul maître dispose du jeton qui l'autorise à échanger des données avec l'ensemble des stations maître ou esclaves du réseau. Si le maître a terminé de transmettre ses données sur le réseau, le jeton est attribué à un autre maître qui à son tour aura la

liberté d'échanger des données avec l'ensemble des stations maître ou esclaves. Le jeton circule automatiquement entre les stations maîtres, les esclaves n'ont jamais le jeton et ne peuvent par conséquent pas lire ou écrire des données d'autres stations du réseau.

### 10.5.5 Echange de données entre les stations maîtres et esclaves



L'utilisateur contrôle l'échange des données entre les stations à l'aide de diverses Fbox *Fupla* placées dans les pages *Fupla* et disponibles dans le *Fbox Selector*. Nous y trouvons des Fbox pour écrire (*SEND*) ou lire (*RCV*) un paquet de données, mais aussi pour supporter différents formats de donnée : binaire, entiers, flottante, data blocs, ...

La Fbox *SEND* ou *RCV* peut être étirée avec plus ou moins d'entrées et sorties, ce qui permet de définir la taille du paquet de données à échanger avec une autre station.

L'adresse du canal de communication utilisé par la Fbox de transmission des données est défini par le symbole représenté en haut à gauche de la Fbox et la lie à la Fbox *SASI* portant le même nom dans laquelle est défini l'adresse du canal. Ce symbole peut être édité en plaçant la souris sur la Fbox et en sélectionnant le menu de contexte *Fbox properties Name*

Chaque Fbox *SEND* et *RCV* est équipé d'une entrée binaire d'activation des échanges de données. Si cette entrée est en permanence à l'état haut, les échanges donnée sont répétés aussi rapidement que possible. Si une courte impulsion est appliquée sur l'entrée, l'échange des données sera effectué au moins une fois mais il est toujours possible de le forcer avec le bouton *Execute* ou au démarrage à froid du PCD avec l'option *Initialization* de la de la fenêtre d'ajustage.

Les données de la station maître présentes sur les entrées d'une Fbox *SEND* sont envoyées vers la station esclave définie par la fenêtre d'ajustage. Alors que les données présentes sur les sorties d'une Fbox *RCV* sont proviennent de la station esclave selon les paramètres la fenêtre d'ajustage : adresse de la station esclave, élément source et adresse de base.

Seul les stations maîtres sont programmées avec des Fbox *SEND* et *RCV* ! Les stations esclaves doivent seulement être assignées avec le canal de communication.

Selon les Fbox utilisées, la fenêtre d'ajustage permet de définir vers quelles stations esclaves sont envoyées les données de la station maître (*SEND*), ou dans quelle stations esclave le maître lit les données. (*RCV*)

**Les paramètres de la fenêtre d'ajustage.**

**Profi-S-Bus Address**

Défini le numéro de station Profi-S-Bus Bus esclave

**Source, destination station**

Défini le numéro de station S-Bus Bus esclave

**Source, destination element**

Défini le type de données à écrire ou lire dans l'esclave

**Source, destination address**

Défini l'adresse de la première donnée à écrire ou lire dans l'esclave

Le nombre de données échangées est dépendant du nombre d'entrées ou sorties de la Fbox *SEND*, *RCV*

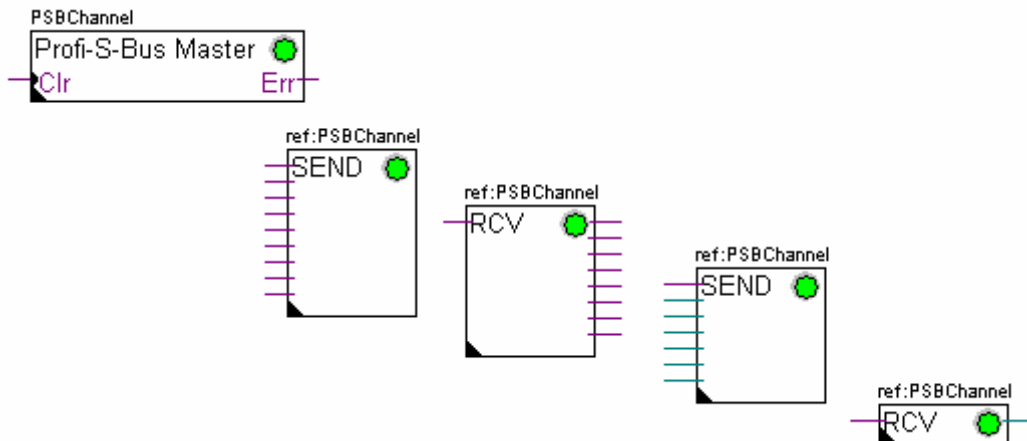
**10.5.6 Diagnostiques**



Si le programme est en ligne, une LED de couleur verte ou rouge est affiché en haut à droite des Fbox *SASI*, *SEND* et *RCV*. Vert indique que la transmission des données de est OK, rouge indique une erreur.

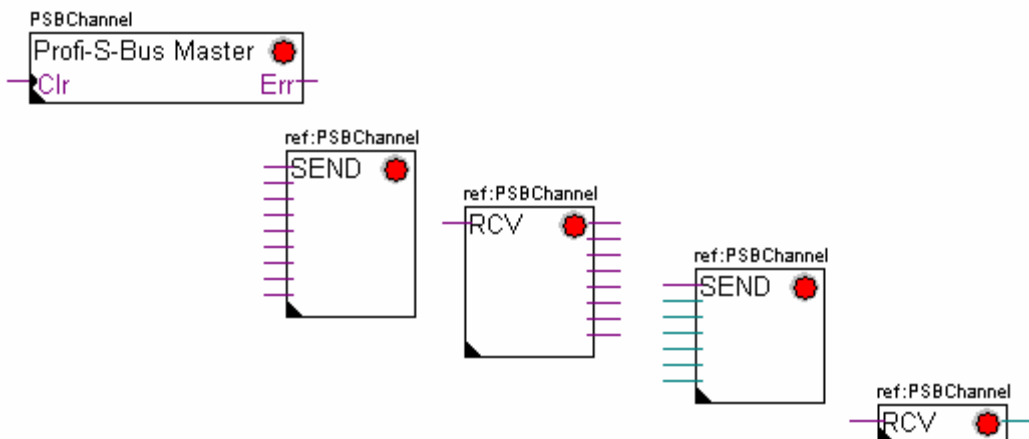
**Fonctionnement correct**

Toutes les Fbox sont vertes, les échanges de données fonctionnent correctement.



**Aucune donnée ne peut être échangée sur le réseau**

La Fbox *SASI*, *SEND* et *RCV* sont rouges, aucune donnée ne peut être échangée sur le réseau.

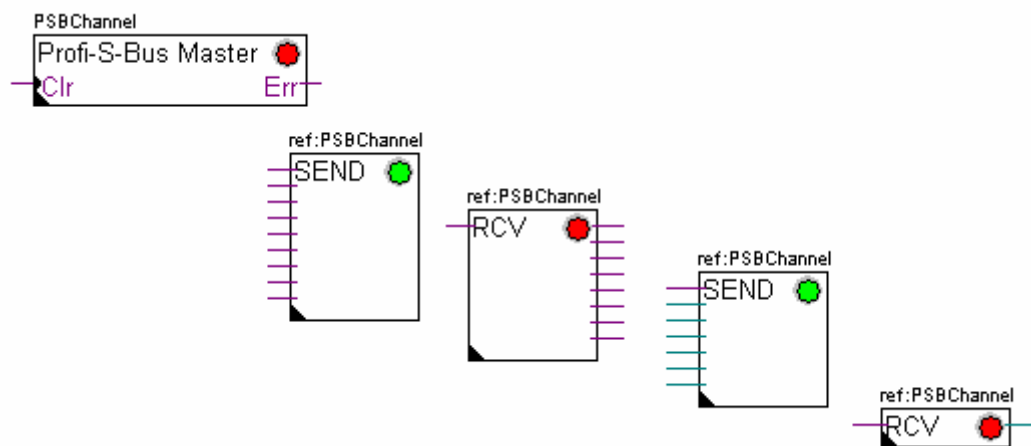


### Actions correctives possibles sur la station maître ou esclave :

- Vérifier les *Hardware Settings*
- Vérifier le chargement des *Hardware Settings* dans le PCD
- Vérifier si toutes les stations utilisent le même profile: S-Net, DP
- Vérifier si toutes les stations communiquent à la même vitesse
- Vérifier si le canal de communication défini *avec les Hardware Settings* et la fonction *SAS/* sont identiques (même numéro de canal)
- Vérifier si le PCD est équipé du hardware de communication nécessaire
- Vérifier si les stations sont raccordées au réseau et sous tension
- Vérifier le câblage du réseau
- Vérifier si la version du firmware supporte Profi-S-Bus

### Seul quelques données ne sont pas être échangées sur le réseau

La Fbox *SAS/* ainsi que quelques Fbox *SEND* et *RCV* sont rouges. Les Fbox vertes échantent correctement les données



### Actions correctives possibles sur la station maître

Vérifier les paramètres de la fenêtre d'ajustage des Fbox *SEND* et *RCV* dont le diagnostic est rouge. Vérifier que l'adresse de l'esclave soit bien présente dans le réseau.

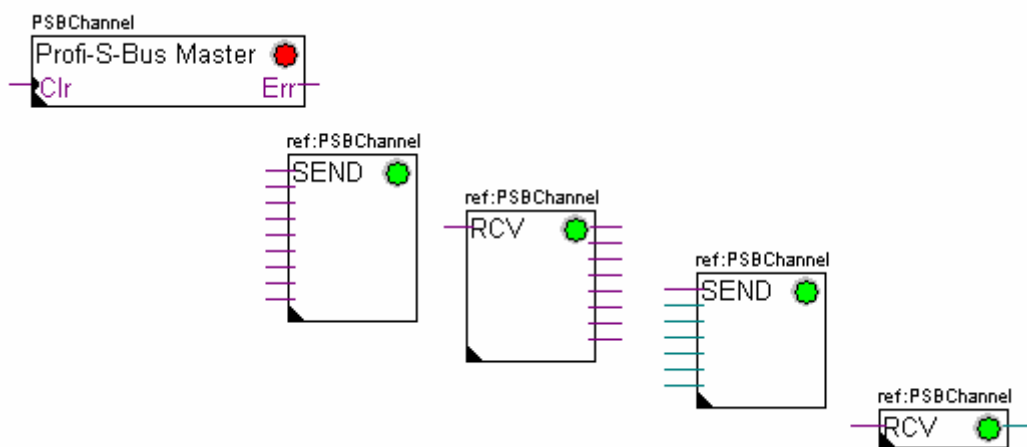
### Actions correctives possibles sur la station esclave

Pour chaque Fbox *SEND* et *RCV* en erreur, relever le numéro de station esclave et vérifier les stations concernées.

- Vérifier si les *Hardware Settings* sont correctement définis
- Vérifier si le PCD est équipé du hardware de communication nécessaire
- Vérifier si la station est raccordée au réseau et sous tension
- Vérifier le câblage au réseau
- Vérifier si la version du firmware supporte Profi-S-Bus

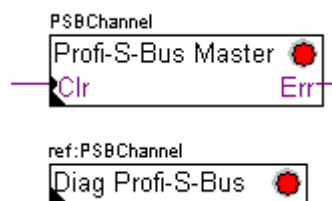
### Seule la Fbox SASI est rouge

Ouvrir le fenêtre d'ajustage de la Fbox SASI et quitter la dernière erreur avec le bouton *Clear*.



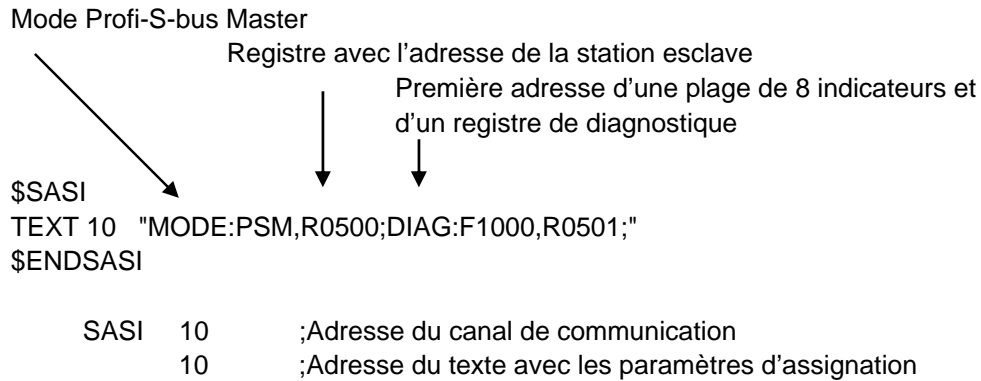
### Fbox de diagnostic

Si la lampe SASI est rouge, il est toujours possible d'obtenir un diagnostic en consultant la fenêtre d'ajustage de la fonction *SASI Diagnostic*. Cette fonction est à placer juste après la fonction SASI



## 10.6 Programme IL

### 10.6.1 Assigner le canal maître à l'aide d'une instruction SASI



L'assignation du canal est assurée à l'aide d'une instruction SASI placée au début du programme: séquence d'initialisation Graftec ou bloc d'initialisation XOB 16.

L'instruction SASI compte deux paramètres: l'adresse du canal de communication et l'adresse d'un texte avec tous les paramètres nécessaires au canal.

Les paramètres du texte d'assignation sont différents d'un réseau de communication à l'autre, ainsi que pour une station esclave ou maître.

Si le PCD exploite plusieurs canaux de communications, définir chaque canal à l'aide d'une instruction SASI et d'un texte d'assignation.

Selon les réseaux, les paramètres du canal de peuvent être complétés avec les *Hardware Settings*.

### 10.6.2 Assigner le canal esclave

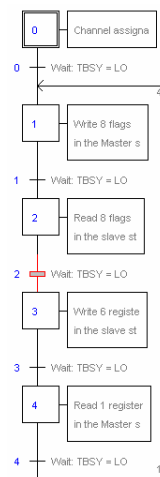
Aucune instruction SASI n'est nécessaire pour la station esclave d'un réseau Profi-S-Bus. Toutes les définitions nécessaires sont déjà présentes dans les *Hardware Settings*.

### 10.6.3 Principe d'échanges des données sur un réseau multi-maîtres.

Le réseau de communication multi-maîtres est composé de plusieurs stations maîtres et esclaves. Les stations maîtres sont les seules stations autorisées à lire ou écrire des données des autres stations maîtres mais aussi esclaves. L'échange de données entre les esclaves n'est pas autorisé.

Avec un mode de communication multi-maîtres, les échanges de données effectués sur le réseau sont répartis entre plusieurs maîtres. Mais à un instant donné, un seul maître dispose du jeton qui l'autorise à échanger des données avec l'ensemble des stations maître ou esclaves du réseau. Si le maître a terminé de transmettre ses données sur le réseau, le jeton est attribué à un autre maître qui à son tour aura la liberté d'échanger des données avec l'ensemble des stations maître ou esclaves. Le jeton circule automatiquement entre les stations maîtres, les esclaves n'ont jamais le jeton et ne peuvent par conséquent pas lire ou écrire des données d'autres stations du réseau.

### 10.6.4 Echange de données entre les stations maîtres et esclaves



Etape initiale: assignation du canal

Etape: échange de données

Transition : attend la fin de l'échange

L'échange des données entre les stations est un programme séquentiel: l'assignation du canal de communication n'est traitée qu'une seule fois, les échanges de données sur le réseau n'ont lieu que si l'échange de données qui précède est terminé. C'est pourquoi nous proposons de traiter les échanges de données IL avec l'éditeur Graftec.

L'étape initiale permet d'assigner le canal de communication au démarrage à froid du PCD.

Les autres étapes sont traitées en boucle et supportent chacune l'échange d'un paquet de données.

Chaque étape est séparée par une transition qui teste l'indicateur de diagnostic TBSY et définit si l'échange de données est terminé. Nous sommes autorisés à échanger les données définies par l'étape qui suit seulement si TBSY est à l'état bas.

#### Echanger des données à l'aide d'une étape

Avant d'échanger les données, nous devons définir l'adresse de la station esclave dans le registre déclaré à cet effet par le texte d'assignation:

#### **Définir l'adresse de la station slave**

```
LDL    R 500 ;Adresse du registre avec l'adresse de la station esclave
        11    ;Adresse S-Bus
```

```
LDH    R 500 ;Adresse du registre avec l'adresse de la station esclave
        21    ;Adresse Profi-S-Bus
```

L'échange des données entre les stations est supporté à l'aide de deux instructions: STXM pour écrire les données dans une station esclave (*SEND*) SRXM pour lire les données d'une station esclave (*RCV*)

Chacune de ces instructions compte quatre paramètres : l'adresse du canal, le nombre de données à échanger et l'adresse de la première donnée source, destination

**Ecriture de 8 indicateurs (F 0,...,F 7) dans une station esclave (F 200,...,F 207)**

STXM 10 ;Adresse du canal  
 8 ;Nombre de données à échanger  
 F 0 ;Adresse de la première donnée source (Station locale)  
 F 200 ;Adresse de la première donnée destination (Station esclave)

**Lecture d'un registre(R 25) d'une station esclave (R 125)**

SRXM 10 ;Adresse du canal  
 1 ;Nombre de données à échanger  
 R 25 ;Adresse de la première donnée source (Station esclave)  
 R 125 ;Adresse de la première donnée destination (Station locale)

Remarque :

Seul les stations maîtres sont programmées avec STXM et SRXM ! Les stations esclaves doivent seulement être assignées avec le canal de communication.

**Attendre la fin de transmission à l'aide d'une transition**

STL F 1003 ; Vérifie que TBSY soit à l'état bas

Le texte d'assignation définit une plage de 8 indicateurs de diagnostics pour la communication, le troisième est mis à l'état haut lors de la transmission des données et tombe à l'état bas lorsque l'échange est terminé.

**10.6.5 Diagnostiques****Assinations du canal**

En cas problème de communication, vérifier si l'assignation du canal est correctement effectuée. Traiter le programme en mode pas à pas et vérifier que l'instruction SASI ne positionne pas le flag erreur. Si non l'assignation du canal n'a pas pu être correctement effectuée et la communication ne peut pas fonctionner.

**Actions correctives possibles sur la station maître ou esclave :**

- Vérifier les *Hardware Settings*
- Vérifier le chargement des *Hardware Settings* dans le PCD
- Vérifier si toutes les stations utilisent le même profil: S-Net, DP
- Vérifier si toutes les stations communiquent à la même vitesse
- Vérifier si le canal de communication défini avec les *Hardware Settings* et l'instruction *SASI* sont identiques (même numéro de canal)
- Vérifier si le PCD est équipé du hardware de communication nécessaire
- Vérifier si les stations sont raccordées au réseau et sous tension
- Vérifier le câblage du réseau
- Vérifier si la version du firmware supporte Profi-S-Bus



### **Les données ne sont pas échangées sur le réseau**

Le texte d'assignation définit une plage de 8 indicateurs de diagnostics pour la communication, le cinquième (*TDIA : Transmitter diagnostic*) est mis à l'état haut lors d'une erreur de transmission des données. Le test pas à pas du programme de communication vous permet de déterminer quel sont les instructions STXM et SRXM en erreur.

Attention si une erreur de communication se produit le flag de diagnostic TDIA reste à l'état haut tant que le registre de diagnostic n'est pas repositionné à zéro.

### **Actions correctives possibles sur la station maître**

Vérifier les paramètres de des instructions STXM et SRXM en erreur. Vérifier que l'adresse de l'esclave soit bien présente dans le réseau.

### **Actions correctives possibles sur la station esclave**

Pour chaque instruction STXM et SRXM en erreur, relever le numéro de station esclave et vérifier les stations concernées.

- Vérifier si les *Hardware Settings* sont correctement définis
- Vérifier si le PCD est équipé du hardware de communication nécessaire  
Vérifier si la station est raccordée au réseau et sous tension
- Vérifier le câblage au réseau
- Vérifier si la version du firmware supporte Profi-S-Bus

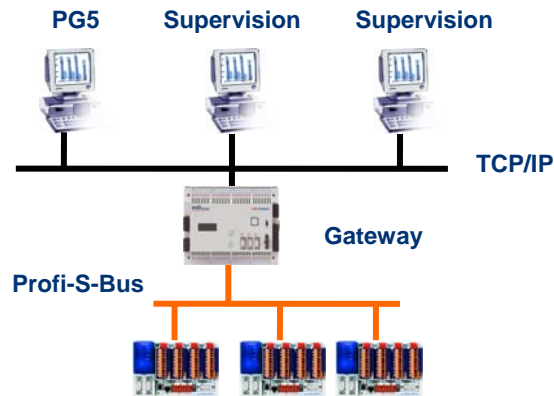
### **Registre de diagnostic**

Le registre de diagnostic peut donner plus d'informations sur la nature de l'erreur de communication, affichez le contenu binaire du registre et comparez le avec les descriptions du manuel des instructions PCD ou du réseau de communication.

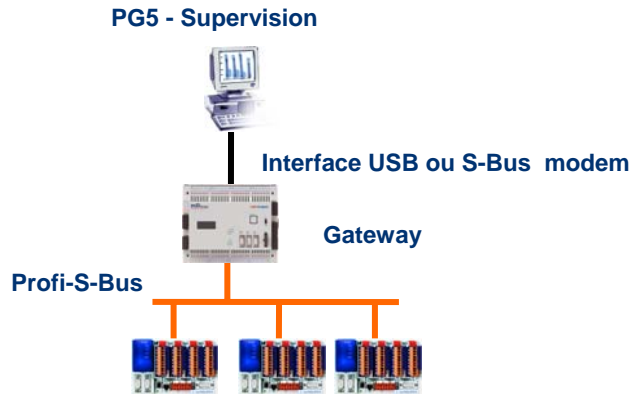
## 10.7 Fonction Gateway

La fonction *Gateway* est couramment utilisée pour faire communiquer ensemble deux réseaux de communication différents ou adapter un outil de programmation PG5, une supervision Visi+ sur un réseau différent que celui directement supporté par ces équipements.

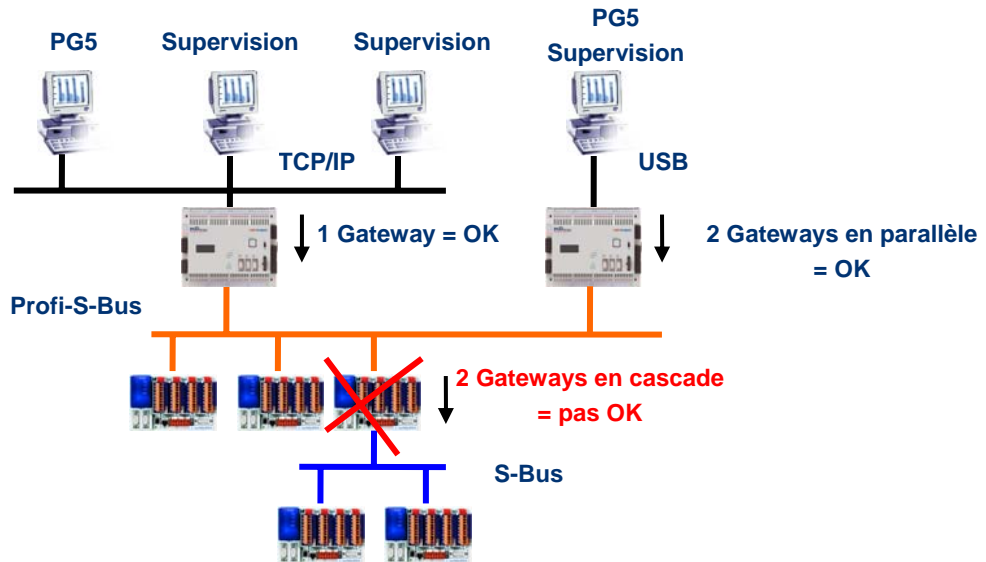
### 10.7.1 Application



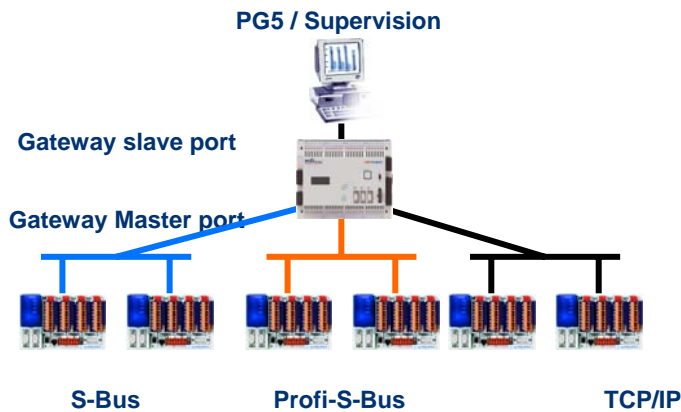
La fonction *Gateway* permet d'effectuer un pont entre deux réseaux. Par exemple lier un réseau Ethernet avec le réseau Profi-S-Bus. Ainsi les systèmes PCD échangent des données sur un bus de terrain propre au domaine de l'automatisation et séparé du réseau informatique d'entreprise. Mais les ordinateurs équipés d'un logiciel PG5 ou d'un système de supervision Visi+ peuvent échanger des données avec les différents PCD.



La fonction *Gateway* peut servir d'interface entre le réseau de communication et le monde extérieur. Par exemple, établir une communication via modem, ou une interface de communication USB.

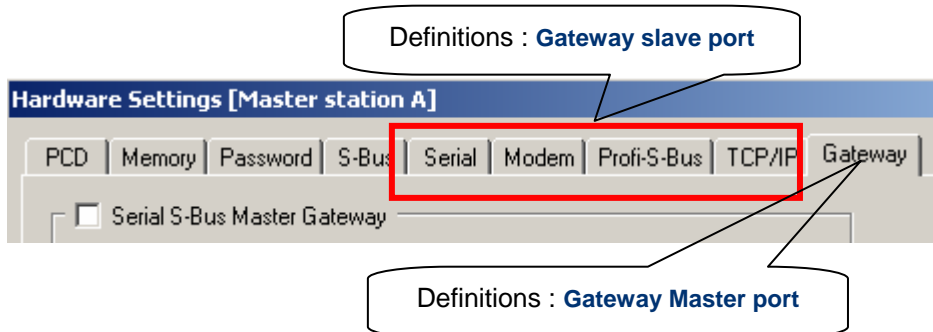


Pour des raisons de respect des timings de communication, nous ne sommes pas autorisés à utiliser plusieurs Gateways en cascade. Mais il est possible de les définir en parallèle sur le même réseau.



Si nécessaire, la fonction Gateway peut établir un pont vers plusieurs sous réseaux de communication.

### 10.7.2 Configuration de la fonction Gateway PGU

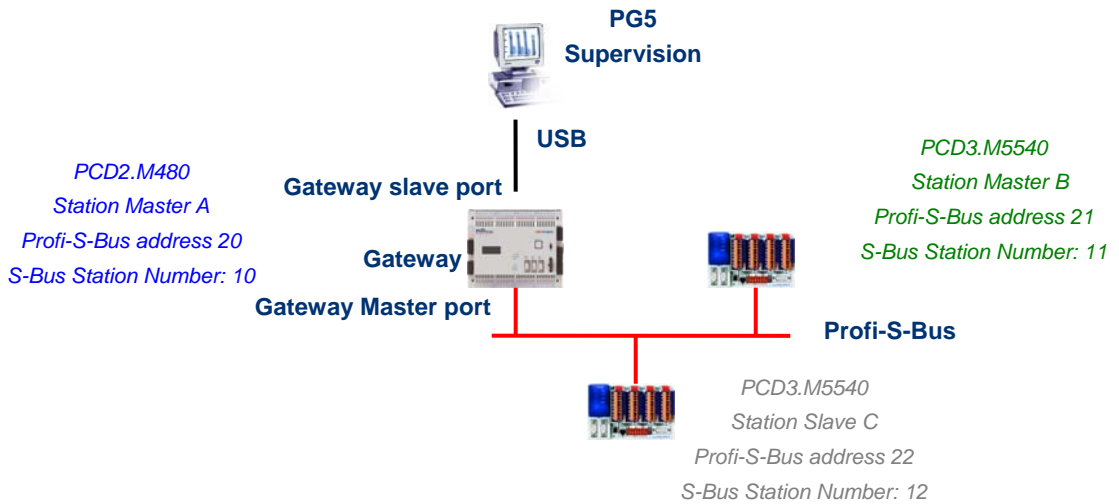


La fonction Gateway est très facile à mettre en service, elle ne nécessite aucun programme, mais seulement quelques paramètres dans les *Hardware Settings* du PCD.

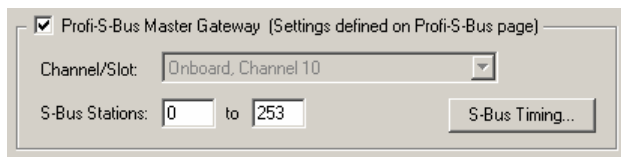
En règle générale, il faut seulement définir un *Gateway Slave Port* et un *Gateway Master Port*, en suite tout est supporté automatiquement par la fonction *Gateway*.

Si les données se présentent sur le *Gateway Slave Port* et qu'elles ne concernent pas la station locale (*Gateway*), alors les données sont transmises à l'un des sous réseaux raccordés sur les *Gateway Master Port*, selon les plages d'adresses valides définies pour chaque sous réseau.

**Exemple : Gateway USB, Profi-S-Bus**

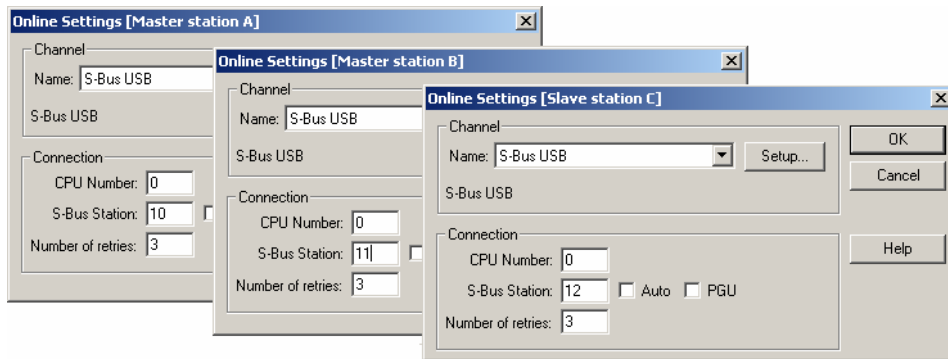


**Hardwares Settings de la station Master A**



Le Gateway USB est une exception, il ne demande aucun paramètre pour le *Gateway Slave port*, seul le *Gateway Master port* doit être paramétré. ( Ne pas oublier de charger la nouvelle configuration dans le Master A !)

**Onlines Settings des CPU du projet**



Pour établir la communication USB avec chacun des PCD, il faut encore ajuster les *Onlines Settings* de chaque CPU du projet avec le canal USB et le numéro de station S-Bus.

**Tester le bon fonctionnement de la fonction Gateway**

**Slave station C - PCD3.M5540 - Station 12**

Activer un des CPU, *Master B* ou *Slave C*, du projet et le mettre en ligne pour tester la communication avec la station.



Si nécessaire le *Online Configurator* permet de vérifier le numéro de la station en ligne. Il est ainsi possible de charger le programme dans le CPU actif et de le tester en restant toujours lié avec le câble USB sur la station *Master A*

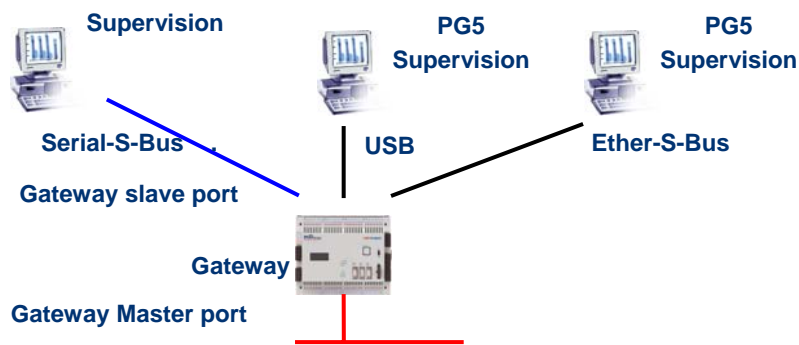
**Master station B - PCD3.M5540 - Station 11**

Pour communiquer avec une autre station du réseau, activer le CPU et se mettre en ligne.

Remarque:

Avec la fonction *Gateway*, seul le numéro de station S-Bus esclave est défini, le numéro de station Profi-S-Bus n'est pas pris en compte car les télégrammes sont adressés à toutes les stations Profi-S-Bus (Broadcast)

**10.7.3 Configuration d'un Gateway Slave port esclave supplémentaire**

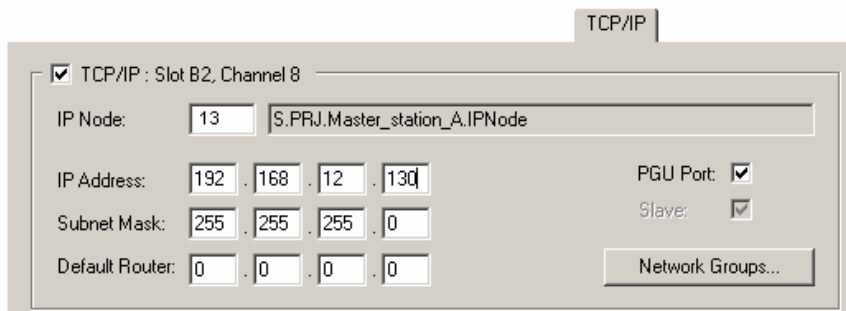


Le Gateway Slave port est un moyen d'accéder au réseau depuis l'extérieur. Si nécessaire, un second ou un troisième *Gateway Slave port* peuvent être définis.

**Hardwares Settings**

En général, le PCD ne supporte qu'un seul canal esclave PGU. Mais les nouveaux automates PCD2M480 et PCD3Mxxxx peuvent en supporter plusieurs sur le même PCD. La configuration du second Gateway Slave PGU est complètement supportée par les *Hardwares Settings*.

**Exemple: ajouter un second Gateway Ether-S-Bus, Profi-S-Bus**



Le second *Gateway Slave port PGU* est ajouté en configurant les *Hardware Settings* avec le nœud et l'adresse TCP/IP. Si l'automate est un PCD2.M480, il faut aussi définir le module de communication PCD sur le Slot B2 avec le PCD7.F65x (module Ethernet).

**Programme Fupla ou IL**

Avec les anciens PCD mais aussi les nouveaux PCD2M480 et PCD3Mxxxx, il est aussi possible d'utiliser une Fbox /instruction SASI supplémentaire pour ajouter un second *Gateway Slave port*.

Ce *Gateway slave port*, sans fonctionnalité PGU, ne supporte pas l'outil de programmation PG5 mais uniquement un terminal ou un système de supervision. Seul la lecture et l'écriture des données PCD ne sont supportées: registres, indicateurs, ...

**Exemple Fupla : ajouter un troisième Serial-S-Bus, Profi-S-Bus**

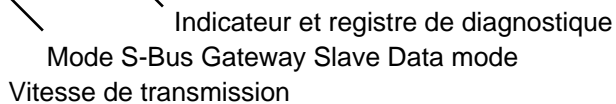


Le paramètre d'ajustage *Gateway* doit alors être défini avec l'option Yes. Selon le type de canal, les autres paramètres de la fenêtre d'ajustage doivent être aussi correctement définis.

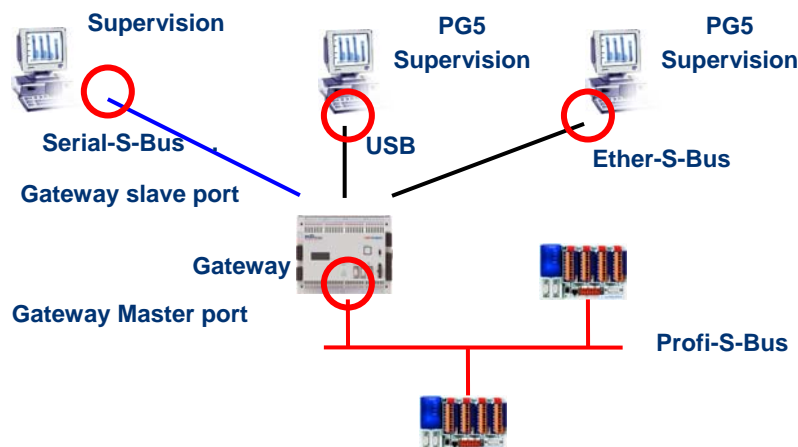
**Exemple IL : ajouter un troisième Serial-S-Bus, Profi-S-Bus**

Utiliser le texte suivant pour assigner le canal :

```
$SASI
TEXT 11 "UART:9600;MODE:GS2;DIAG:F1110,R0501;"
$ENDSASI
```



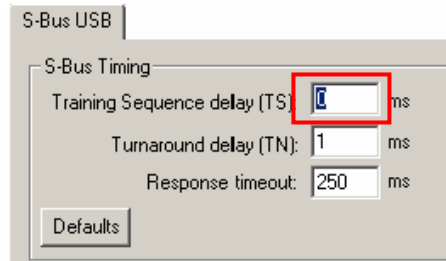
**10.7.4 Timings de communication**



En général les *timings* de communication sont toujours définis avec les valeurs par défaut et cela fonctionne correctement. Mais l'usage de la fonction *Gateway*

augmente les temps de réactions nécessaire à l'échange de données. Il est alors parfois nécessaire d'ajuster le timeout des stations maîtres faisant usage du *Gateway*. La figure ci-dessus indique quels sont les canaux maîtres dont le timeout devrait être ajusté.

Pour ajuster le *Timeout* du PG5, utiliser les *Online Settings* de la station *Master Station A* :



Pour ajuster le *Timeout* du programme d'échange de données sur le PCD *Gateway* utiliser la Fbox : *SASI Profi-S-Bus Extended*



## 10.8 Autres références

Pour de plus amples informations, vous pouvez aussi vous référer aux manuels suivants :

- Guide des instructions 26/133
- Profi-S-Bus (en préparation)
- Exemple de projet Profi-S-Bus installé avec votre PG5

## Table des matières

---

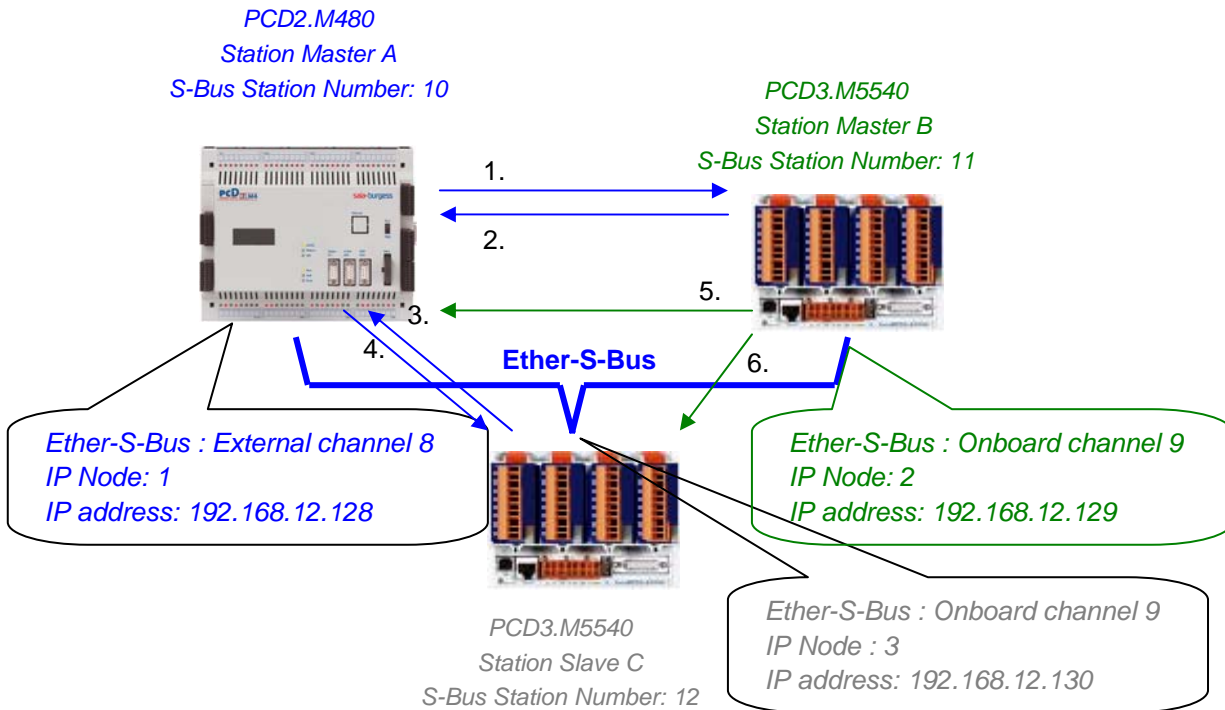
<b>TABLE DES MATIERES</b>	<b>1</b>
<b>11 ETHER-S-BUS</b>	<b>2</b>
11.1 Exemple de réseau Ether-S-Bus.	2
11.2 Exemples d'échanges de données Ether-S-Bus.	2
11.3 Le Projet PG5	3
11.4 Hardwares Settings maîtres, esclaves	3
11.4.1 Définir les paramètres PCD	3
11.4.2 Définir le numéro de station S-Bus sur le réseau	4
11.4.3 Définir le canal de communication Ether-S-Bus	4
11.4.4 Charger les Hardwares Settings sur le CPU	5
11.5 Programme Fupla	5
11.5.1 Assigner le canal à l'aide d'une Fbox SASI	5
11.5.2 Assigner le canal maître	6
11.5.3 Assigner le canal esclave	6
11.5.4 Principe d'échanges des données sur un réseau multi-maîtres.	6
11.5.5 Echange de données entre les stations maîtres et esclaves	7
11.5.6 Diagnostiques	8
11.6 Programme IL	11
11.6.1 Assigner le canal maître à l'aide d'une instruction SASI	11
11.6.2 Assigner le canal esclave	11
11.6.3 Principe d'échanges des données sur un réseau multi-maîtres.	11
11.6.4 Echange de données entre les stations maîtres et esclaves	12
11.6.5 Diagnostiques	13
11.7 Fonction Gateway	15
11.7.1 Application	15
11.7.2 Configuration de la fonction Gateway PGU	16
11.7.3 Configuration d'un Gateway Slave port esclave supplémentaire	18
11.7.4 Timings de communication	20
11.8 Autres références	20



# 11 Ether-S-Bus

Cet exemple présente comment échanger quelques données comme des registres et indicateurs entre les PCD raccordés sur le réseau Ether-S-Bus.

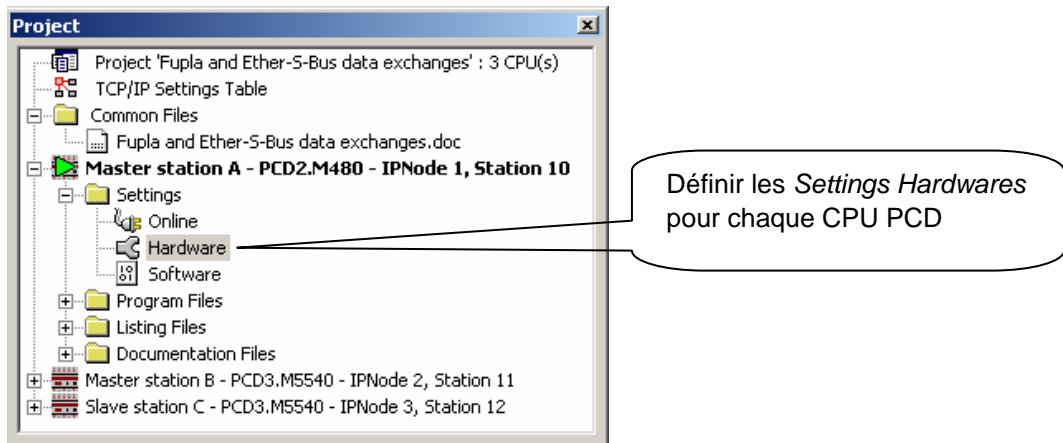
## 11.1 Exemple de réseau Ether-S-Bus.



## 11.2 Exemples d'échanges de données Ether-S-Bus.

	<b>Master with data exchanges</b>	<b>Data on the network</b>	<b>Passive master or slave</b>
	<b>Master station A</b>		<b>Master station B</b>
1	Blinker0 .. 7 F 0 .. 7	Write 8 flags in the Master station B	Station_A.Blinker0 .. 7 F 100 .. 107
2	Master_B.Value100 R 125	Read 1 register in the Master station B	Value100 R 25
			<b>Slave station C</b>
3	Slave_C.Binary0 .. 7 F 100 .. 107	Read 8 flags in the slave station C	Binary0 .. 7 F 20 .. 27
4	Value0 .. 5 R 0 .. 5	Write 6 registers in the slave station C	Master_A.Value0 .. 5 R 20 .. 25
	<b>Master station B</b>		<b>Master station A</b>
5	Temperature1 .. 4 Dynamic registers	Write the temperature measures to the slave C	Master_B.Temperature1 .. 4 R 100 .. 104
			<b>Slave station C</b>
6	Temperature1 .. 4 Dynamic registers	Write the temperature measures to the master A	Master_B.Temperature1 .. 4 R 100 .. 104

### 11.3 Le Projet PG5

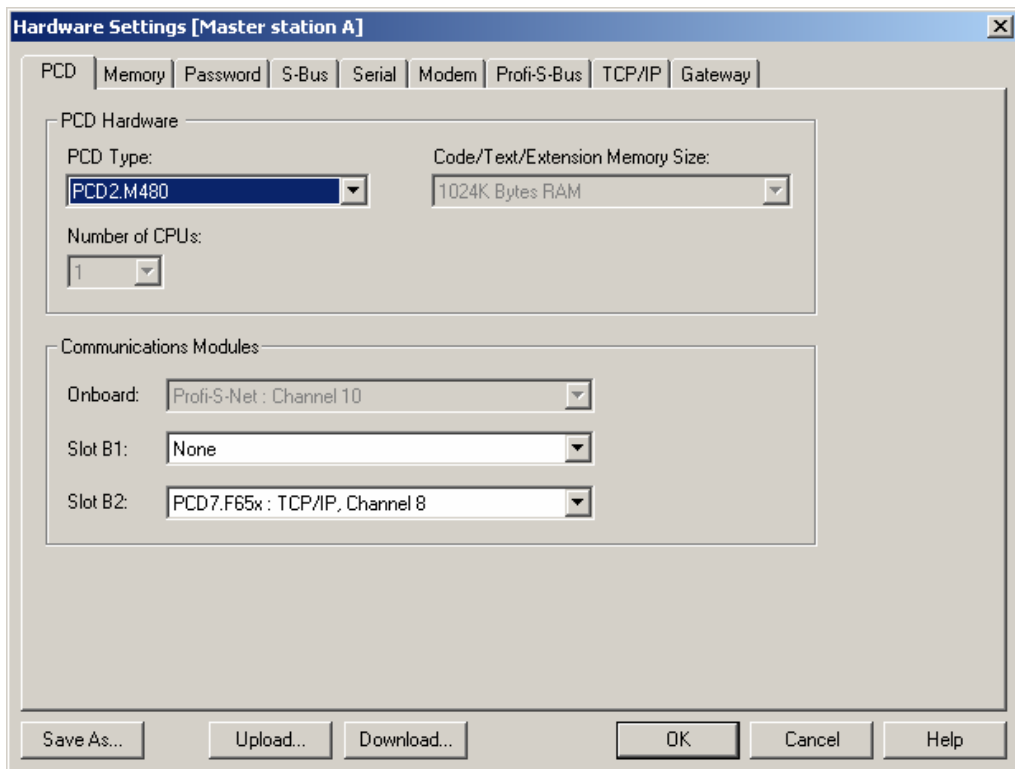


Le *Saia Project Manager* donne une vue de l'ensemble des PCD d'un projet d'application ainsi que des paramètres pour les réseaux de communication. Nous commencerons par ajouter un CPU dans le projet pour chaque station disponible sur le réseau.

### 11.4 Hardwares Settings maîtres, esclaves

Les configurations des *Hardwares Settings* d'une station maître et esclaves sont semblables.

#### 11.4.1 Définir les paramètres PCD

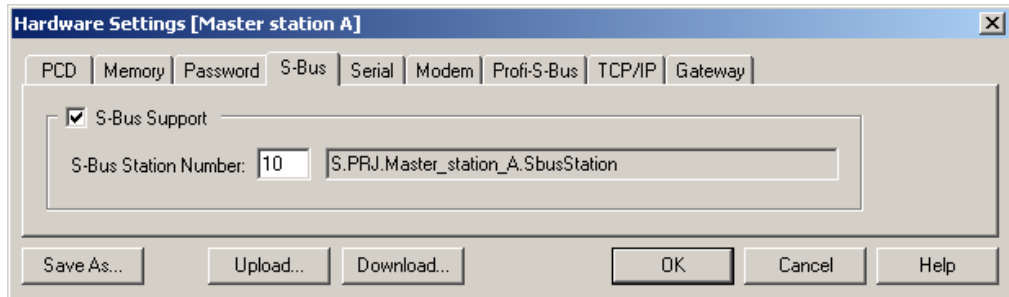


**PCD Type**

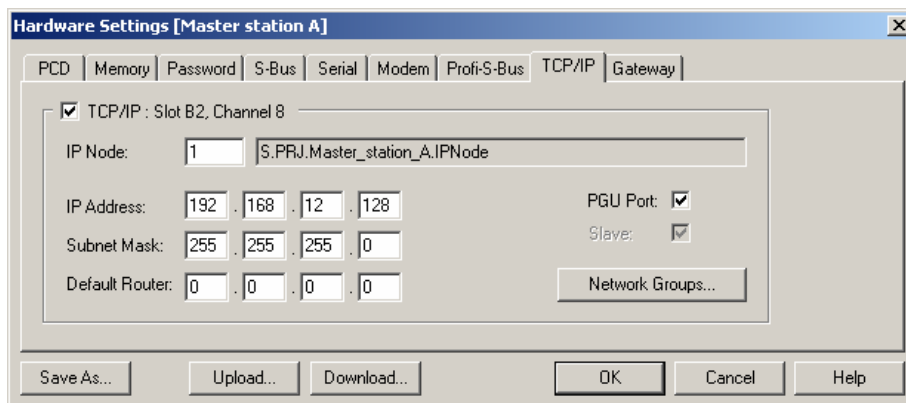
Définir le type de CPU

**Communication Modules**

Si nécessaire indiquer le type des modules de communications insérés dans les slot B1 et B2 du PCD2.M480.

**11.4.2 Définir le numéro de station S-Bus sur le réseau****S-Bus Station Number**

Le numéro de station S-Bus est commun à tous les canaux de communication du PCD

**11.4.3 Définir le canal de communication Ether-S-Bus****IP Address**

Numéro de station Ether-S-Bus lié au canal

**IP Node**

Numéro du nœud TCP/IP. Le nœud est utilisé dans les Fbox SEND et RCV pour définir une station esclave avec laquelle échanger des données.

**PGU Port ou Slave**

Définit le canal comme slave ou PGU. Cette définition peut être cumulée avec la fonction master en ajoutant une Fbox SASI master dans le programme Fupla.

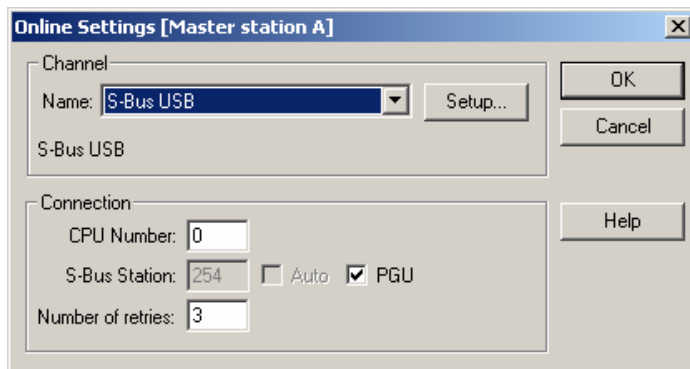
**Esclave PGU**

Supporte l'échange de donnée avec les stations maîtres, systèmes de supervision et terminaux. Mais supporte aussi l'outil de programmation et de mise en service PG5..

**Esclave**

Supporte uniquement l'échange de donnée avec les stations maîtres, systèmes de supervision et terminaux.

### 11.4.4 Charger les Hardwares Settings sur le CPU

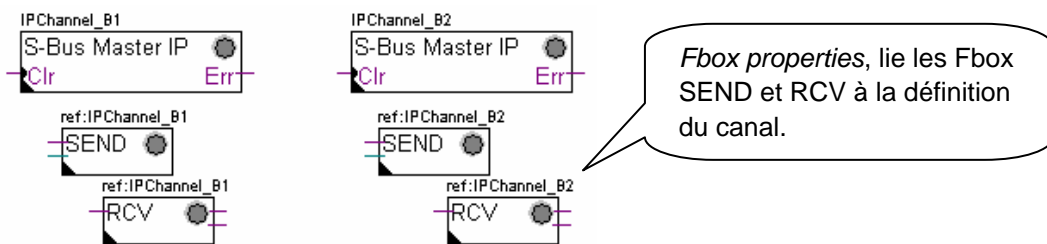


Avec les nouveaux systèmes PCD2.M480 et PCD3, les *Hardwares Settings* peuvent être chargés sur les PCD2.M480 et PCD3 au travers de l'interface USB. Il faut juste vérifier que les *Online Settings* soient définis avec un canal *Ether-S-Bus PGU*.

Charger les paramètres dans le PCD à l'aide du bouton *Download* présent dans la fenêtre *Hardware Settings*.

## 11.5 Programme Fupla

### 11.5.1 Assigner le canal à l'aide d'une Fbox SASI



L'assignation du canal est assurée à l'aide d'une FBox SASI placée au début du fichier Fupla. Chaque réseau de communication dispose de sa propre Fbox SASI, car les paramètres d'assignation sont différents d'un réseau à l'autre, de même pour une station esclave ou maître.

Si le PCD exploite plusieurs canaux de communications, définir chacun des canaux à l'aide de la Fbox SASI appropriée. Puis placer la souris sur la Fbox SASI et sélectionner le menu de contexte *Fbox properties*, définir un *Name* différent pour chaque Fbox selon le canal utilisé. Ce nom permet de lier diverses Fbox d'échanges *SEND* et *RCV* avec la Fbox SASI correspondant au canal utilisé.

Selon les réseaux, les paramètres du canal de communication peuvent être partiellement définis à partir de la fenêtre d'ajustage de la Fbox SASI et complétés dans les *Hardwares Settings*.

Mais le numéro du canal de communication est toujours défini à l'aide de la fenêtre d'ajustage de la FBox SASI. Le numéro du canal dépend du Hardware PCD et du hardware de communication utilisé : slot B1, B2, interface série PCD7.F, ...

### 11.5.2 Assigner le canal maître



L'assignation du canal maître est assurée en complétant les hardware Settings esclave avec l'une des Fbox ci-dessus

Paramètres de la fenêtre d'ajustage:

#### *Channel*

Défini le numéro du canal raccordé au réseau. Dépend du PCD et de sont hardware.

#### *Timing*

Le Timeout est en général défini avec la valeur par défaut (0) et n'est ajusté que pour les applications particulières (Gateway )

### 11.5.3 Assigner le canal esclave

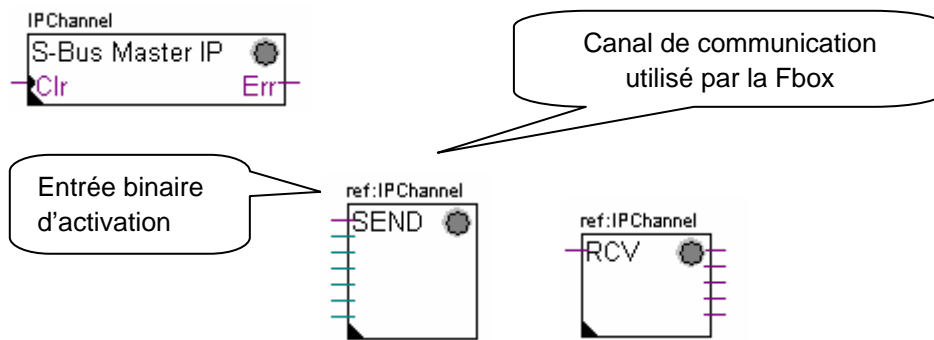
Aucune FBox SASI n'est nécessaire pour la station esclave d'un réseau Ether-S-Bus. Toutes les définitions nécessaires sont déjà présentes dans les *Hardware Settings*.

### 11.5.4 Principe d'échanges des données sur un réseau multi-maîtres.

Le réseau de communication multi-maîtres est composé de plusieurs stations maîtres et esclaves. Les stations maîtres sont les seules stations autorisées à lire ou écrire des données des autres stations maîtres mais aussi esclaves. L'échange de données entre les esclaves n'est pas autorisé.

Avec un mode de communication multi-maîtres, les échanges de données effectués sur le réseau sont répartis entre plusieurs maîtres. Mais à un instant donné, un seul maître dispose du jeton qui l'autorise à échanger des données avec l'ensemble des stations maître ou esclaves du réseau. Si le maître a terminé de transmettre ses données sur le réseau, le jeton est attribué à un autre maître qui à son tour aura la liberté d'échanger des données avec l'ensemble des stations maître ou esclaves. Le jeton circule automatiquement entre les stations maîtres, les esclaves n'ont jamais le jeton et ne peuvent par conséquent pas lire ou écrire des données d'autres stations du réseau.

### 11.5.5 Echange de données entre les stations maîtres et esclaves



L'utilisateur contrôle l'échange des données entre les stations à l'aide de diverses Fbox Fupla placées dans les pages Fupla et disponibles dans le *Fbox Selector*. Nous y trouvons des Fbox pour écrire (*SEND*) ou lire (*RCV*) un paquet de données, mais aussi pour supporter différents formats de donnée : binaire, entiers, flottante, data blocs, ...

La Fbox *SEND* ou *RCV* peut être étirée avec plus ou moins d'entrées et sorties, ce qui permet de définir la taille du paquet de données à échanger avec une autre station.

L'adresse du canal de communication utilisé par la Fbox de transmission des données est défini par le symbole représenté en haut à gauche de la Fbox et la lie à la Fbox SASI portant le même nom dans laquelle est défini l'adresse du canal. Ce symbole peut être édité en plaçant la souris sur la Fbox et en sélectionnant le menu de context *Fbox properties Name*

Chaque Fbox *SEND* et *RCV* est équipé d'une entrée binaire d'activation des échanges de données. Si cette entrée est en permanence à l'état haut, les échanges donnée sont répétés aussi rapidement que possible. Si une courte impulsion est appliquée sur l'entrée, l'échange des données sera effectué au moins une fois mais il est toujours possible de le forcer avec le bouton *Execute* ou au démarrage à froid du PCD avec l'option *Initialization* de la de la fenêtre d'ajustage.

Les données de la station maître présentes sur les entrées d'une Fbox *SEND* sont envoyées vers la station esclave définie par la fenêtre d'ajustage. Alors que les données présentes sur les sorties d'une Fbox *RCV* sont proviennent de la station esclave selon les paramètres la fenêtre d'ajustage : adresse de la station esclave, élément source et adresse de base.

Seul les stations maîtres sont programmées avec des Fbox *SEND* et *RCV* ! Les stations esclaves doivent seulement être assignées avec le canal de communication.

Selon les Fbox utilisées, la fenêtre d'ajustage permet de définir vers quelles stations esclaves sont envoyées les données de la station maître (*SEND*), ou dans quelle stations esclave le maître lit les données. (*RCV*)

**Les paramètres de la fenêtre d'ajustage.*****IP Node***

Défini le numéro de noeud Ether-S-Bus Bus esclave

***Source, destination station***

Défini le numéro de station S-Bus Bus esclave

***Source, destination element***

Défini le type de données à écrire ou lire dans l'esclave

***Source, destination address***

Défini l'adresse de la première donnée à écrire ou lire dans l'esclave

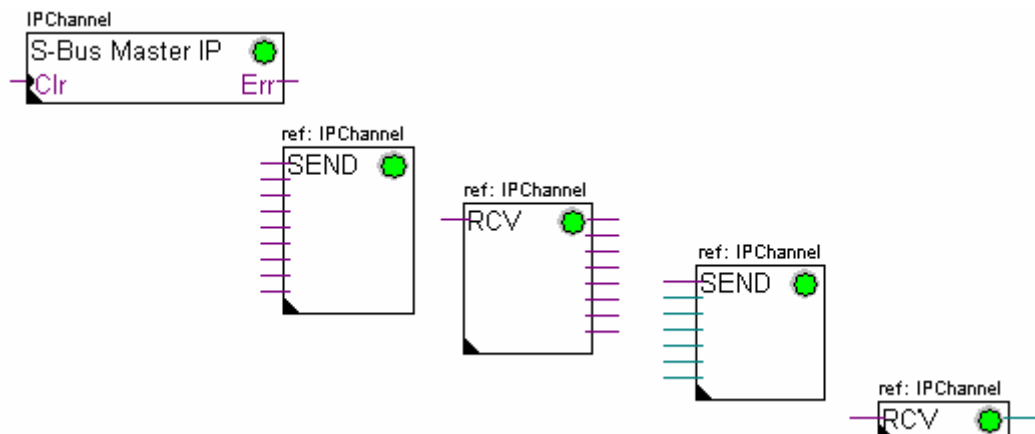
Le nombre de données échangées est dépendant du nombre d'entrées ou sorties de la Fbox *SEND*, *RCV*

**11.5.6 Diagnostiques**

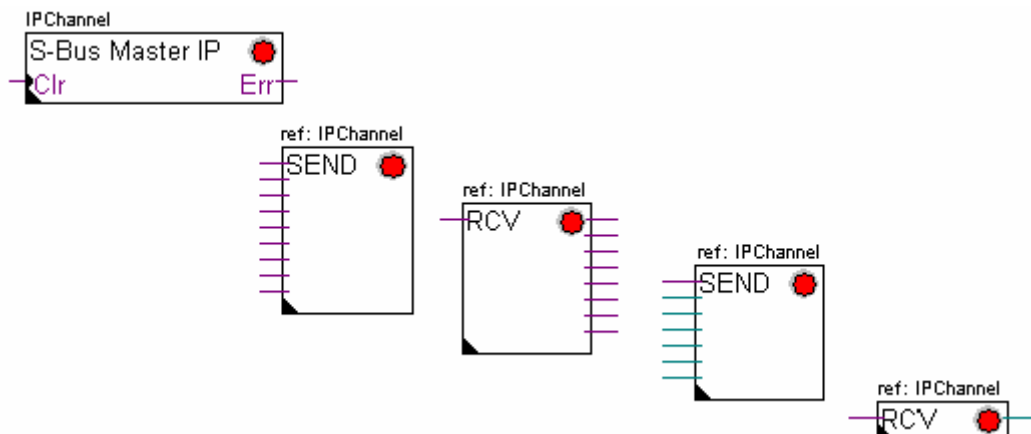
Si le programme est en ligne, une LED de couleur verte ou rouge est affiché en haut à droite des Fbox *SASI*, *SEND* et *RCV*. Vert indique que la transmission des données de est OK, rouge indique une erreur.

**Fonctionnement correct**

Toutes les Fbox sont vertes, les échanges de données fonctionnent correctement.

**Aucune donnée ne peut être échangée sur le réseau**

La Fbox *SASI*, *SEND* et *RCV* sont rouges, aucune donnée ne peut être échangée sur le réseau.

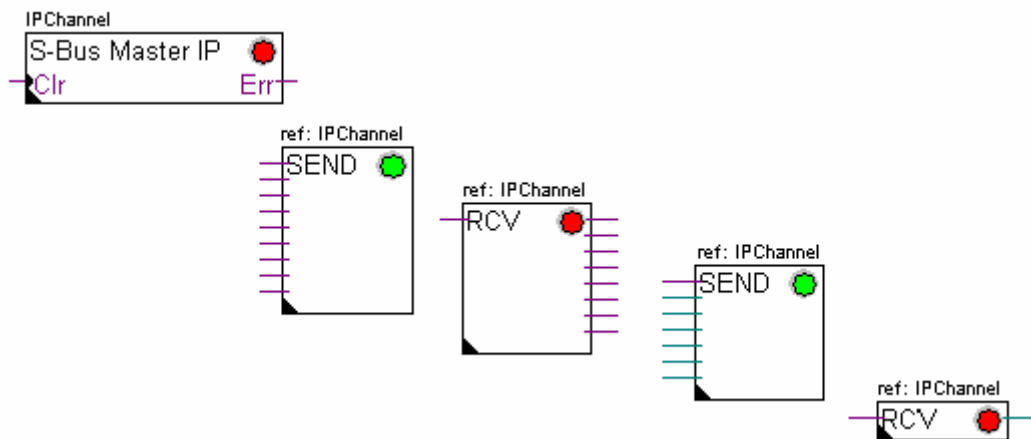


### Actions correctives possibles sur la station maître ou esclave :

- Vérifier les *Hardware Settings*
- Vérifier le chargement des *Hardware Settings* dans le PCD
- Vérifier si le canal de communication défini avec les *Hardware Settings* et la fonction *SASI* sont identiques (même numéro de canal)
- Vérifier si le PCD est équipé du hardware de communication nécessaire
- Vérifier si les stations sont raccordées au réseau et sous tension
- Vérifier le câblage du réseau
- Vérifier si la version du firmware supporte Ether-S-Bus

### Seul quelques données ne sont pas être échangées sur le réseau

La Fbox *SASI* ainsi que quelques Fbox *SEND* et *RCV* sont rouges. Les Fbox vertes échangent correctement les données



### Actions correctives possibles sur la station maître

Vérifier les paramètres de la fenêtre d'ajustage des Fbox *SEND* et *RCV* dont le diagnostic est rouge. Vérifier que l'adresse de l'esclave soit bien présente dans le réseau.

### Actions correctives possibles sur la station esclave

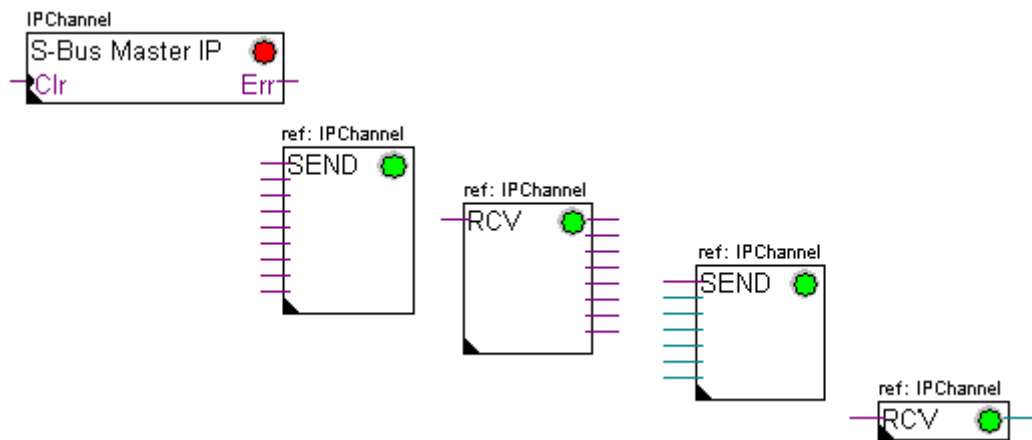
Pour chaque Fbox *SEND* et *RCV* en erreur, relever le numéro de station esclave et vérifier les stations concernées.

- Vérifier si les *Hardware Settings* sont correctement définis
- Vérifier si le PCD est équipé du hardware de communication nécessaire
- Vérifier si la station est raccordée au réseau et sous tension
- Vérifier le câblage au réseau
- Vérifier si la version du firmware supporte Ether-S-Bus



### Seule la Fbox SASI est rouge

Ouvrir le fenêtre d'ajustage de la Fbox *SASI* et quitter la dernière erreur avec le bouton *Clear*.



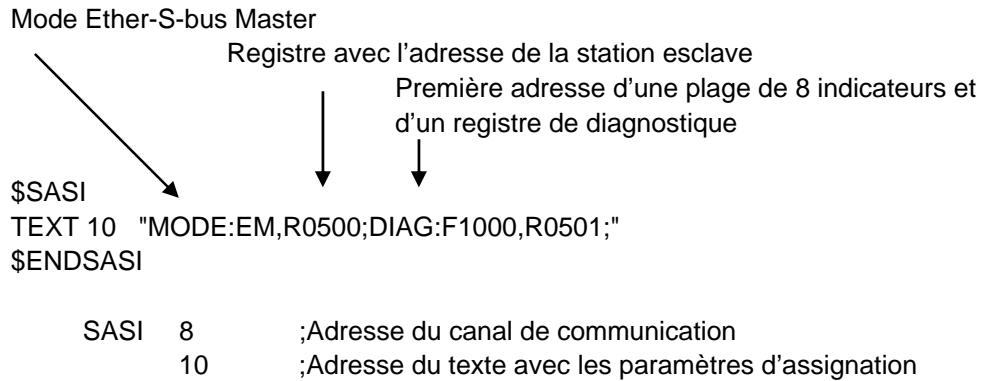
### Fbox de diagnostic

Si la lampe *SASI* est rouge, il est toujours possible d'obtenir un diagnostic en consultant la fenêtre d'ajustage de la fonction *SASI Diagnostic*. Cette fonction est à placer juste après la fonction *SASI*



## 11.6 Programme IL

### 11.6.1 Assigner le canal maître à l'aide d'une instruction SASI



L'assignation du canal est assurée à l'aide d'une instruction SASI placée au début du programme: séquence d'initialisation Graftec ou bloc d'initialisation XOB 16.

L'instruction SASI compte deux paramètres: l'adresse du canal de communication et l'adresse d'un texte avec tous les paramètres nécessaires au canal.

Les paramètres du texte d'assignation sont différents d'un réseau de communication à l'autre, ainsi que pour une station esclave ou maître.

Si le PCD exploite plusieurs canaux de communications, définir chaque canal à l'aide d'une instruction SASI et d'un texte d'assignation.

Selon les réseaux, les paramètres du canal de peuvent être complétés avec les *Hardware Settings*.

### 11.6.2 Assigner le canal esclave

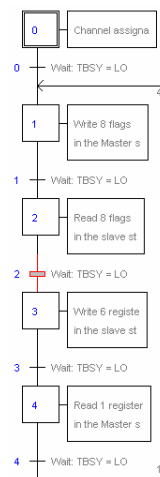
Aucune instruction SASI n'est nécessaire pour la station esclave d'un réseau Ether-S-Bus. Toutes les définitions nécessaires sont déjà présentes dans les *Hardware Settings*.

### 11.6.3 Principe d'échanges des données sur un réseau multi-maîtres.

Le réseau de communication multi-maîtres est composé de plusieurs stations maîtres et esclaves. Les stations maîtres sont les seules stations autorisées à lire ou écrire des données des autres stations maîtres mais aussi esclaves. L'échange de données entre les esclaves n'est pas autorisé.

Avec un mode de communication multi-maîtres, les échanges de données effectués sur le réseau sont répartis entre plusieurs maîtres. Mais à un instant donné, un seul maître dispose du jeton qui l'autorise à échanger des données avec l'ensemble des stations maître ou esclaves du réseau. Si le maître a terminé de transmettre ses données sur le réseau, le jeton est attribué à un autre maître qui à son tour aura la liberté d'échanger des données avec l'ensemble des stations maître ou esclaves. Le jeton circule automatiquement entre les stations maîtres, les esclaves n'ont jamais le jeton et ne peuvent par conséquent pas lire ou écrire des données d'autres stations du réseau.

### 11.6.4 Echange de données entre les stations maîtres et esclaves



Etape initiale: assignation du canal

Etape: échange de données

Transition : attend la fin de l'échange

L'échange des données entre les stations est un programme séquentiel: l'assignation du canal de communication n'est traitée qu'une seule fois, les échanges de données sur le réseau n'ont lieu que si l'échange de données qui précède est terminé. C'est pourquoi nous proposons de traiter les échanges de données IL avec l'éditeur Graftec.

L'étape initiale permet d'assigner le canal de communication au démarrage à froid du PCD.

Les autres étapes sont traitées en boucle et supportent chacune l'échange d'un paquet de données.

Chaque étape est séparée par une transition qui teste l'indicateur de diagnostic TBSY et définit si l'échange de données est terminé. Nous sommes autorisés à échanger les données définies par l'étape qui suit seulement si TBSY est à l'état bas.

#### **Echanger des données à l'aide d'une étape**

Avant d'échanger les données, nous devons définir l'adresse de la station esclave dans le registre déclaré à cet effet par le texte d'assignation:

#### **Définir l'adresse de la station slave**

```
LDL    R 500 ;Adresse du registre avec l'adresse de la station esclave
        11   ;Adresse S-Bus
```

```
LDH    R 500 ;Adresse du registre avec l'adresse de la station esclave
        2    ;Nœud IP
```

L'échange des données entre les stations est supporté à l'aide de deux instructions: STXM pour écrire les données dans une station esclave (*SEND*) SRXM pour lire les données d'une station esclave (*RCV*)

Chacune de ces instructions compte quatre paramètres : l'adresse du canal, le nombre de données à échanger et l'adresse de la première donnée source, destination

**Ecriture de 8 indicateurs (F 0,...,F 7) dans une station esclave (F 200,...,F 207)**

STXM 8 ;Adresse du canal  
 8 ;Nombre de données à échanger  
 F 0 ;Adresse de la première donnée source (Station locale)  
 F 200 ;Adresse de la première donnée destination (Station esclave)

**Lecture d'un registre(R 25) d'une station esclave (R 125)**

SRXM 8 ;Adresse du canal  
 1 ;Nombre de données à échanger  
 R 25 ;Adresse de la première donnée source (Station esclave)  
 R 125 ;Adresse de la première donnée destination (Station locale)

Remarque :

Seul les stations maîtres sont programmées avec STXM et SRXM ! Les stations esclaves doivent seulement être assignées avec le canal de communication.

**Attendre la fin de transmission à l'aide d'une transition**

STL F 1003 ; Vérifie que TBSY soit à l'état bas

Le texte d'assignation définit une plage de 8 indicateurs de diagnostics pour la communication, le troisième est mis à l'état haut lors de la transmission des données et tombe à l'état bas lorsque l'échange est terminé.

**11.6.5 Diagnostiques****Assinations du canal**

En cas problème de communication, vérifier si l'assignation du canal est correctement effectuée. Traiter le programme en mode pas à pas et vérifier que l'instruction SASI ne positionne pas le flag erreur. Si non l'assignation du canal n'a pas pu être correctement effectuée et la communication ne peut pas fonctionner.

**Actions correctives possibles sur la station maître ou esclave :**

- Vérifier les *Hardware Settings*
- Vérifier le chargement des *Hardware Settings* dans le PCD
- Vérifier si toutes les stations utilisent le même profil: S-Net, DP
- Vérifier si toutes les stations communiquent à la même vitesse
- Vérifier si le canal de communication défini *avec les Hardware Settings* et l'instruction *SASI* sont identiques (même numéro de canal)
- Vérifier si le PCD est équipé du hardware de communication nécessaire
- Vérifier si les stations sont raccordées au réseau et sous tension
- Vérifier le câblage du réseau
- Vérifier si la version du firmware supporte Ether-S-Bus

### **Les données ne sont pas échangées sur le réseau**

Le texte d'assignation définit une plage de 8 indicateurs de diagnostics pour la communication, le cinquième (*TDIA : Transmitter diagnostic*) est mis à l'état haut lors d'une erreur de transmission des données. Le test pas à pas du programme de communication vous permet de déterminer les instructions STXM et SRXM en erreur.

Attention, si une erreur de communication se produit le flag de diagnostic TDIA reste à l'état haut tant que le registre de diagnostic n'est pas repositionné à zéro.

#### **Actions correctives possibles sur la station maître**

Vérifier les paramètres des instructions STXM et SRXM en erreur. Vérifier que l'adresse de l'esclave soit bien présente dans le réseau.

#### **Actions correctives possibles sur la station esclave**

Pour chaque instruction STXM et SRXM en erreur, relever le numéro de station esclave et vérifier les stations concernées.

- Vérifier si les *Hardware Settings* sont correctement définis
- Vérifier si le PCD est équipé du hardware de communication nécessaire  
Vérifier si la station est raccordée au réseau et sous tension
- Vérifier le câblage au réseau
- Vérifier si la version du firmware supporte Ether-S-Bus

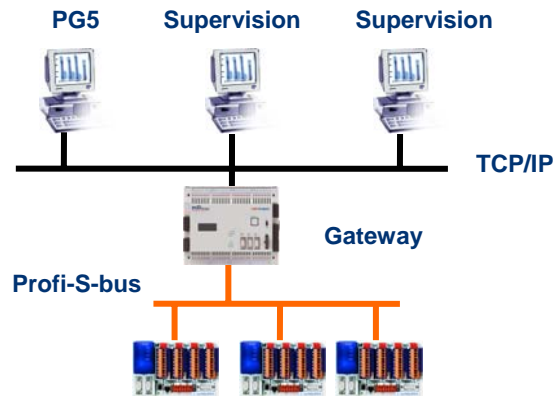
### **Registre de diagnostic**

Le registre de diagnostic peut donner plus d'informations sur la nature de l'erreur de communication, affichez le contenu binaire du registre et comparez le avec les descriptions du manuel des instructions PCD ou du réseau de communication.

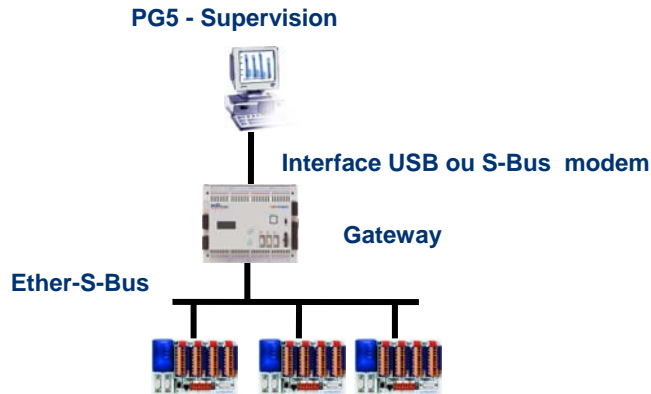
## 11.7 Fonction Gateway

La fonction *Gateway* est couramment utilisée pour faire communiquer ensemble deux réseaux de communication différents et adapter un outil de programmation, une supervision sur un réseau différent que celui directement supporté par ces équipements.

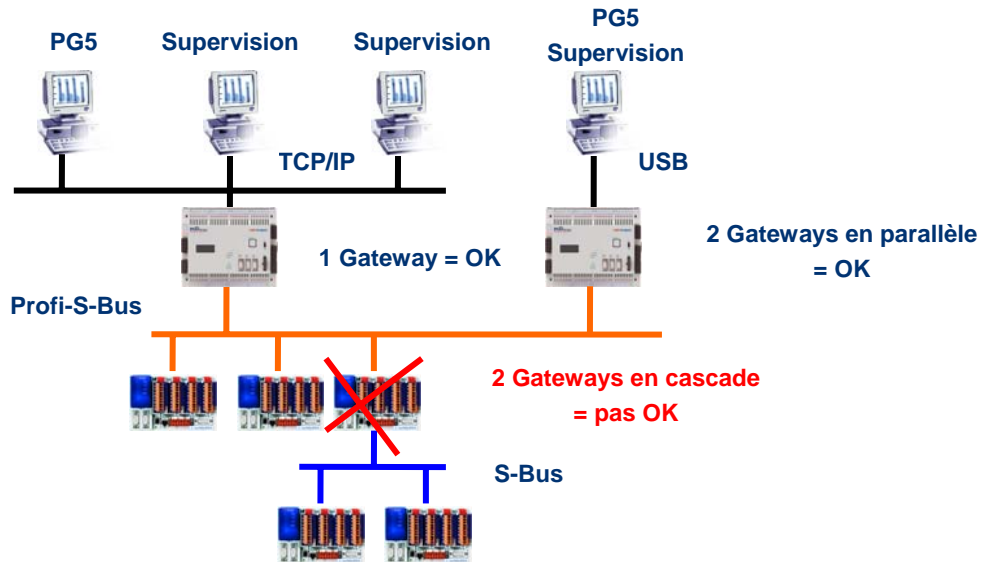
### 11.7.1 Application



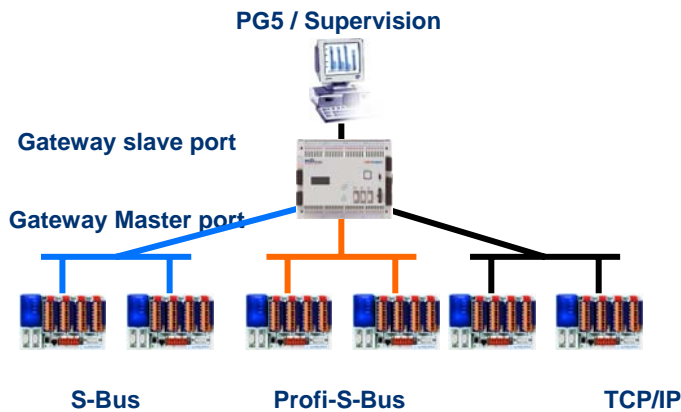
La fonction *Gateway* permet d'effectuer un pont entre deux réseaux. Par exemple lier un réseau Ethernet avec le réseau Profi-S-Bus. Ainsi les systèmes PCD échangent des données sur un bus de terrain propre au domaine de l'automatisation et séparé du réseau informatique d'entreprise. Mais les ordinateurs équipés d'un logiciel PG5 ou d'un système de supervision Visi+ peuvent échanger des données avec les différents PCD.



La fonction *Gateway* peut servir d'interface entre le réseau de communication et le monde extérieur. Par exemple, établir une communication via modem, ou une interface de communication USB.

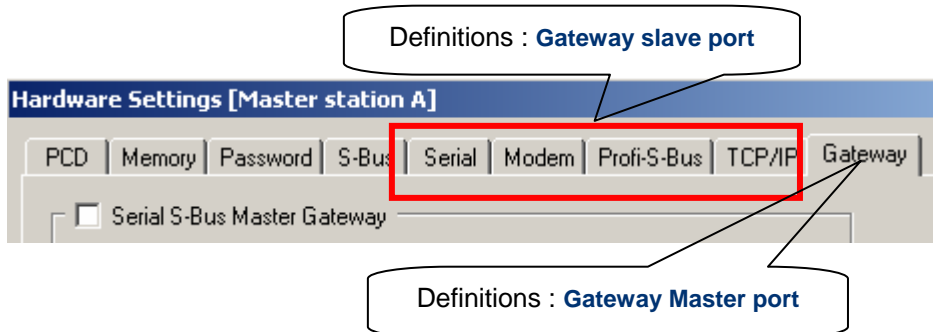


Pour des raisons de respect des timings de communication, nous ne sommes pas autorisés à utiliser plusieurs Gateways en cascade. Mais il est possible de les définir en parallèle sur le même réseau.



Si nécessaire, la fonction Gateway peut établir un pont vers plusieurs sous réseaux de communication.

### 11.7.2 Configuration de la fonction Gateway PGU

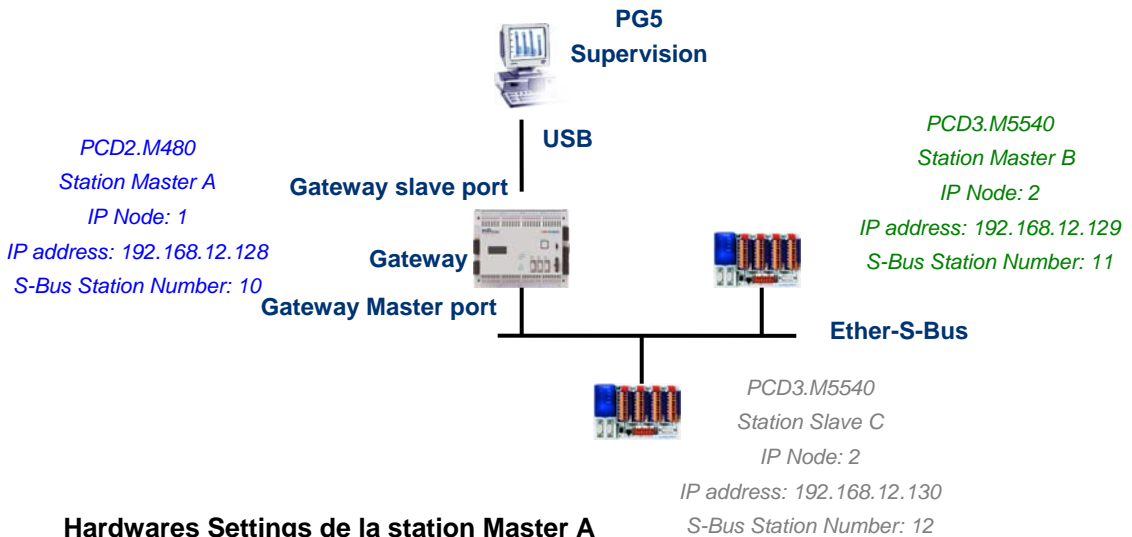


La fonction Gateway est très facile à mettre en service, elle ne nécessite aucun programme, mais seulement quelques paramètres dans les *Hardware Settings* du PCD.

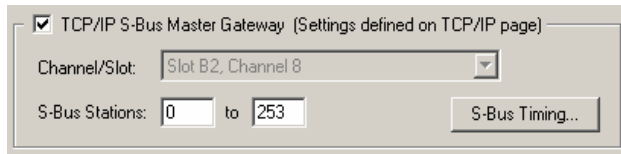
En règle générale, il faut seulement définir un *Gateway Slave Port* et un *Gateway Master Port*, en suite tout est supporté automatiquement par la fonction *Gateway*.

Si les données se présentent sur le *Gateway Slave Port* et qu'elles ne concernent pas la station locale (*Gateway*), alors les données sont transmises à l'un des sous réseaux raccordés sur les *Gateway Master Port*, selon les plages d'adresses valides définies pour chaque sous réseau.

**Exemple : Gateway USB, Ether-S-Bus**

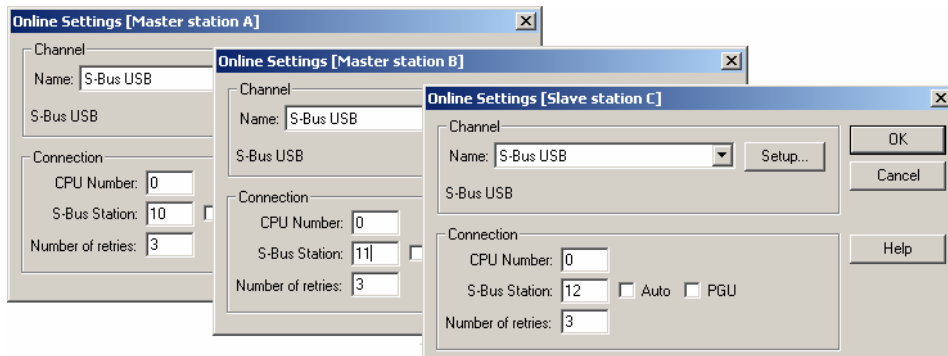


**Hardwares Settings de la station Master A**



Le Gateway USB est une exception, il ne demande aucun paramètre pour le *Gateway Slave port*, seul le *Gateway Master port* doit être paramétré. ( Ne pas oublier de charger la nouvelle configuration dans le Master A ! )


**Onlines Settings des CPU du projet**



Pour établir la communication USB avec chacun des PCD, il faut encore ajuster les *Onlines Settings* de chaque CPU du projet avec le canal USB et le numéro de station S-Bus.




### Tester le bon fonctionnement de la fonction Gateway

 Slave station C - PCD3.M5540 - IPNode 3, Station 12

Activer un des CPU, *Master B* ou *Slave C*, du projet et le mettre en ligne pour tester la communication avec la station.



Si nécessaire le *Online Configurator* permet de vérifier le numéro de la station en ligne. Il est ainsi possible de charger le programme dans le CPU actif et de le tester en restant toujours lié avec le câble USB sur la station *Master A*

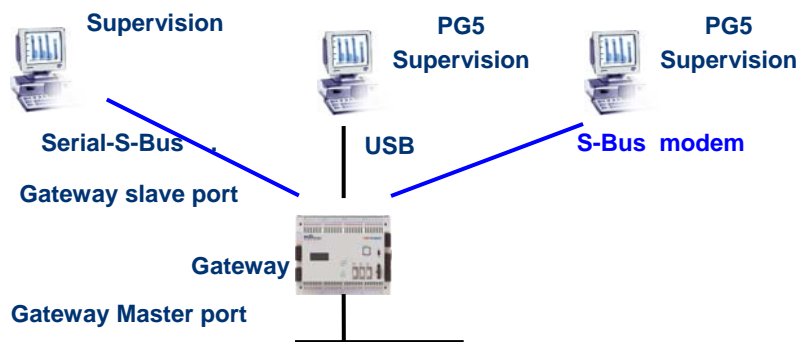
 Master station B - PCD3.M5540 - IPNode 2, Station 11

Pour communiquer avec une autre station du réseau, activer le CPU et se mettre en ligne.

Remarque:

Avec la fonction *Gateway*, seul le numéro de station S-Bus esclave est défini, le numéro de station *Ether-S-Bus* n'est pas pris en compte car les télégrammes sont adressés à toutes les stations *Ether-S-Bus* (Broadcast)

### 11.7.3 Configuration d'un Gateway Slave port esclave supplémentaire

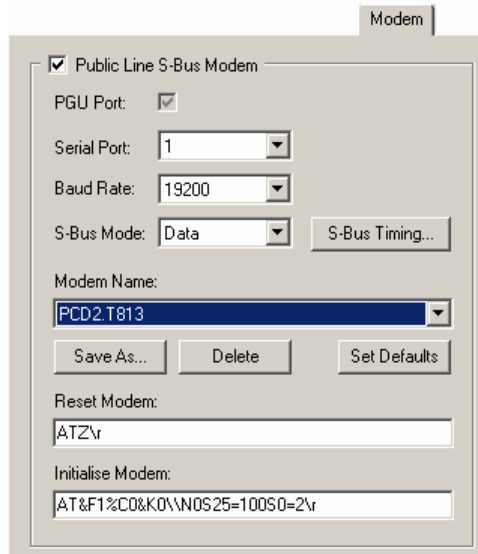


Le Gateway Slave port est un moyen d'accéder au réseau depuis l'extérieur. Si nécessaire, un second ou un troisième *Gateway Slave port* peuvent être définis.

#### Hardware Settings

En général, le PCD ne supporte qu'un seul canal esclave PGU. Mais les nouveaux automates PCD2M480 et PCD3Mxxxx peuvent en supporter plusieurs sur le même PCD. La configuration du second Gateway Slave PGU est complètement supportée par les *Hardware Settings*.

**Exemple: ajouter un second Gateway S-Bus modem, Ether-S-Bus**



Le second *Gateway Slave port PGU* est ajouté en configurant les *Hardware Settings* avec les paramètres pour le modem.

**Programme Fupla ou IL**

Avec les anciens PCD mais aussi les nouveaux PCD2M480 et PCD3Mxxxx, il est aussi possible d'utiliser une Fbox /instruction SASI supplémentaire pour ajouter un second *Gateway Slave port*.

Ce *Gateway slave port*, sans fonctionnalité PGU, ne supporte pas l'outil de programmation PG5 mais uniquement un terminal ou un système de supervision. Seul la lecture et l'écriture des données PCD ne sont supportées: registres, indicateurs, ...

**Exemple Fupla : ajouter un troisième Serial-S-Bus, Ether-S-Bus**



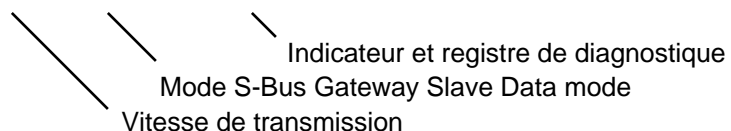
Le paramètre d'ajustage *Gateway* doit alors être défini avec l'option *Yes*. Selon le type de canal, les autres paramètres de la fenêtre d'ajustage doivent être aussi correctement définis.

**Exemple IL : ajouter un troisième Serial-S-Bus, Ether-S-Bus**

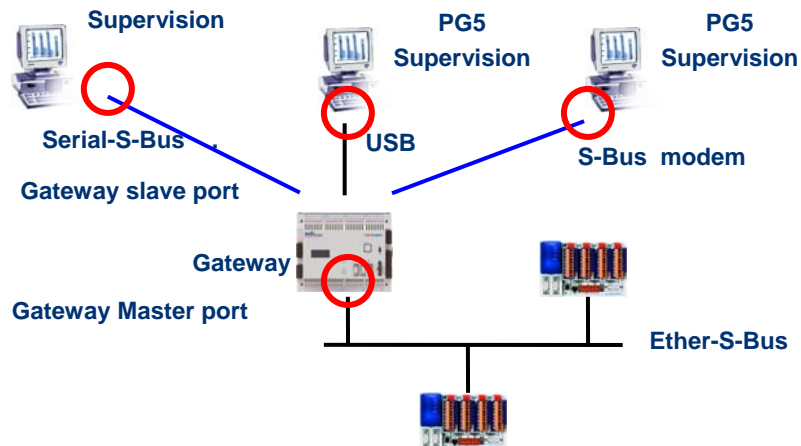
Utiliser le texte suivant pour assigner le canal :

```

$SASI
TEXT 11 "UART:9600;MODE:GS2;DIAG:F1110,R0501;"
$ENDSASI
    
```

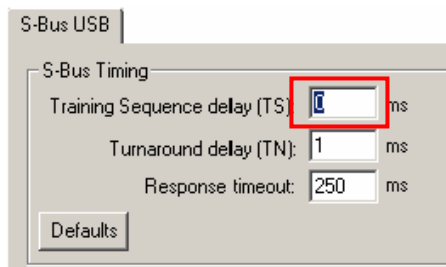


### 11.7.4 Timings de communication



En général les *timings* de communication sont toujours définis avec les valeurs par défaut et cela fonctionne correctement. Mais l'usage de la fonction *Gateway* augmente les temps de réactions nécessaire à l'échange de données. Il est alors parfois nécessaire d'ajuster le timeout des stations maîtres faisant usage du *Gateway*. La figure ci-dessus indique quels sont les canaux maîtres dont le timeout devrait être ajusté.

Pour ajuster le *Timeout* du PG5, utiliser les *Online Settings* de la station *Master Station A* :



Pour ajuster le *Timeout* du programme d'échange de données sur le PCD *Gateway* utiliser la Fbox : *SASI S-Bus IP Extended*



### 11.8 Autres références

Pour de plus amples informations, vous pouvez aussi vous référer aux manuels suivants :

- Guide des instructions 26/133
- Ethernet TCP/IP 27/776
- Exemple de projet Ether-S-Bus installé avec votre PG5

## **Table des matières**

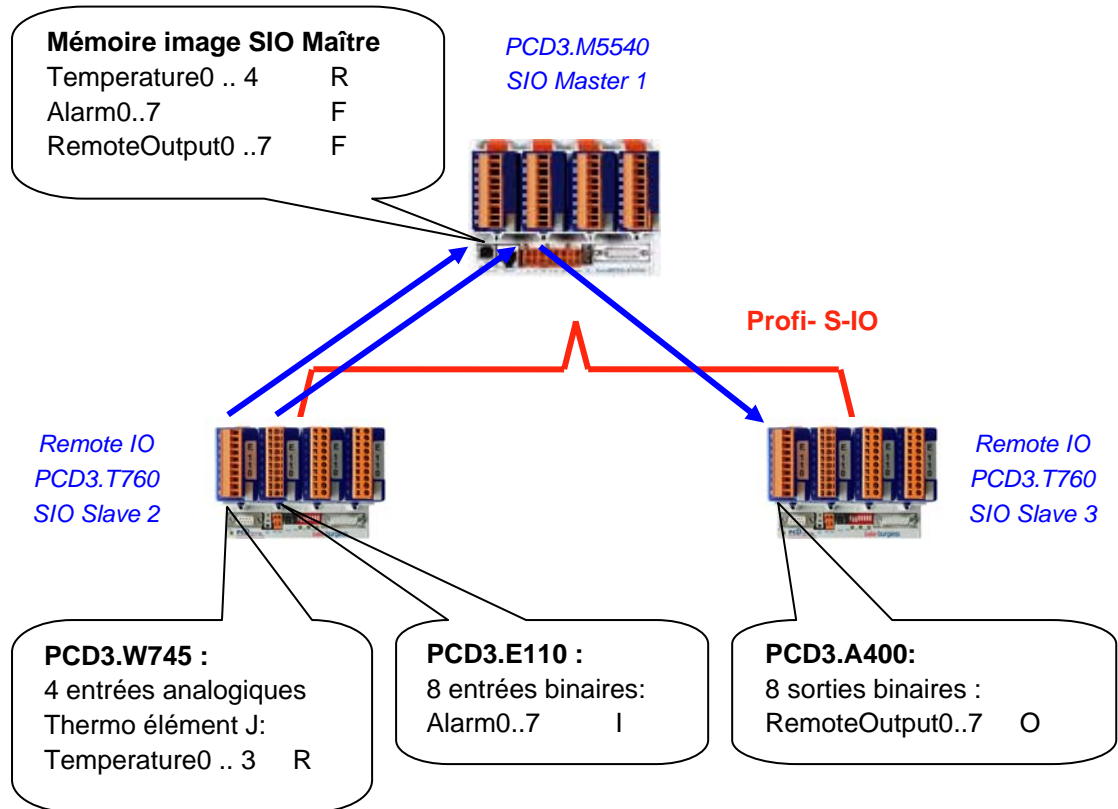
---

<b>TABLE DES MATIERES</b>	<b>1</b>
<b>12 PROFI-S-IO</b>	<b>2</b>
12.1 Exemple de réseau Profi-S-IO.	2
12.2 Fonctionnement général	2
12.3 Le Projet PG5	3
12.4 Définir les stations sur le réseau	3
12.5 Configurer la station maître	4
12.6 Configurer les stations esclaves	4
12.6.1 Configurer les modules d'entrées, sorties.	4
12.6.2 Configurer les symboles des données déportées	5
12.6.3 Configurer les paramètres d'ajustages	5
12.7 Configurer les paramètres du réseau.	6
12.8 Exploiter les symboles réseau dans un programme Fupla ou IL.	6
12.9 Autres références	7

## 12 Profi-S-IO

Cet exemple présente comment utiliser les entrées, sorties binaires ou analogiques déportées sur des borniers RIO de type PCD3.T7xx.

### 12.1 Exemple de réseau Profi-S-IO.



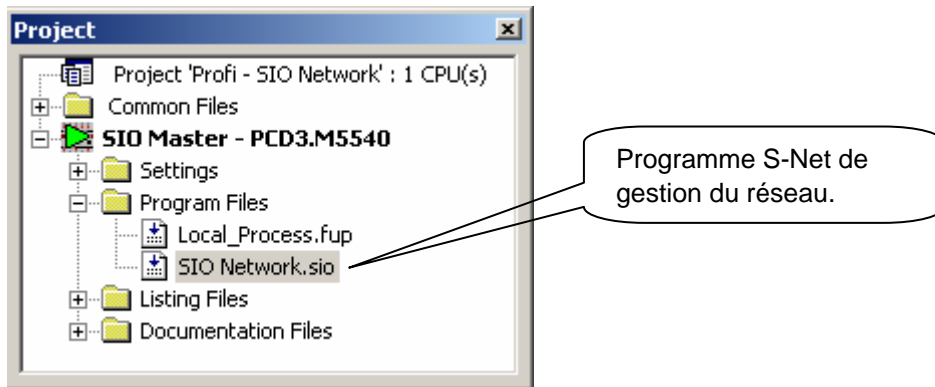
### 12.2 Fonctionnement général

Avec les réseaux Profibus DP et Profi-S-IO, les échanges de données sur le réseau ne sont pas définis avec l'aide de l'éditeur FuPla ou IL, ni configurés à l'aide des *Hardware Settings PG5*, mais uniquement à l'aide du configurateur S-Net. Ce configurateur supporte la définition des stations présentes sur le réseau ainsi que des modules d'entrées et sorties qui y sont insérés. Pour chaque module, les informations en entrée ou sortie sont définies avec un symbole qui sera exploité depuis le programme de la station maître.

Au *build* du projet PG5, S-Net crée automatiquement une mémoire image des données disponibles sur le réseau. Elles seront exploitées depuis le programme FuPla ou IL présent sur la station maître.

Les symboles qui désignent les informations sur les modules d'entrées et sorties déportés sont les mêmes que ceux disponibles sur la mémoire image de la station maître. Ainsi, le programme FuPla ou IL utilise les symboles de la mémoire image comme les autres symboles locaux à la station maître, la gestion des échanges de données sur le réseau sont clairement séparé du contrôle de processus.

### 12.3 Le Projet PG5



Le fichier S-Net s'ajoute dans la station maître de la même manière qu'un fichier Fupla ou IL. Mais choisir une extension de fichier .SIO (Profi-S-IO) ou .DP (Profibus DP).

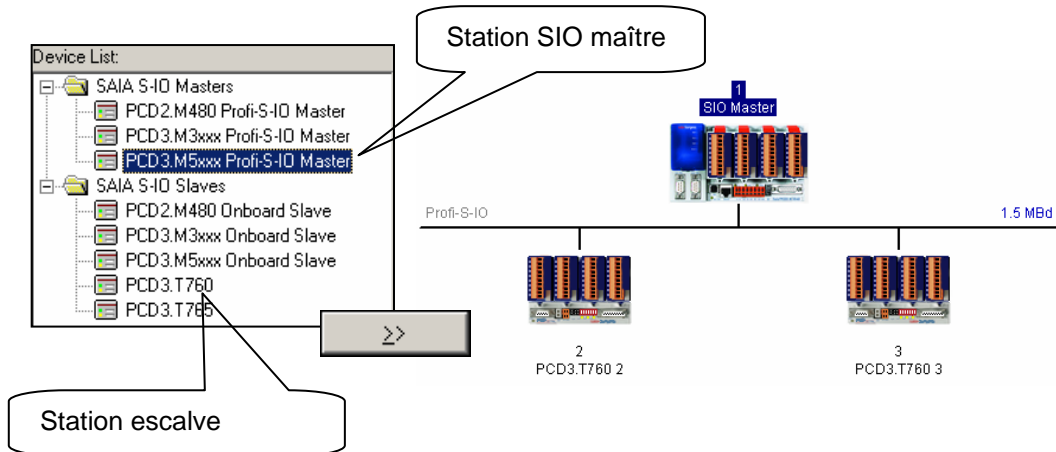
En fait l'usage de configurateur S-Net est identique pour l'échange de données sur un réseau Profi-S-IO et Profibus DP, les seules différences sont :

- L'extension du fichier de configuration: .SIO, .DP
- Les devices supportés sur le réseau: SIO = devices Saia, DP = devices Saia + d'autres fournisseurs.
- Les timings du réseau: Profile Snet, DP

Remarque :

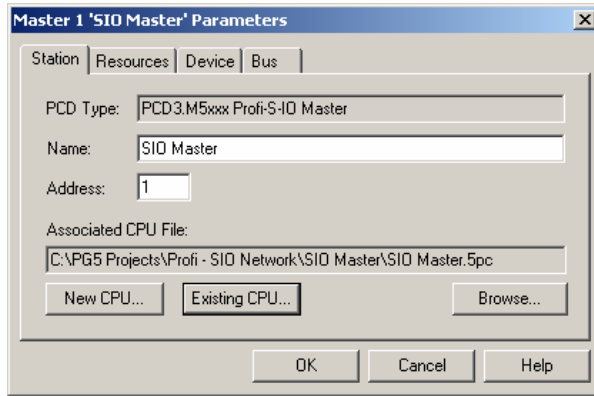
Si des PCD7.T7xx sont présents sur le réseau, toujours choisir le profile S-Net.

### 12.4 Définir les stations sur le réseau



Pour chacune des stations présentes dans le réseau, sélectionner le type de station dans la liste des *devices* et l'ajouter au réseau avec le bouton prévu à cet effet.

## 12.5 Configurer la station maître

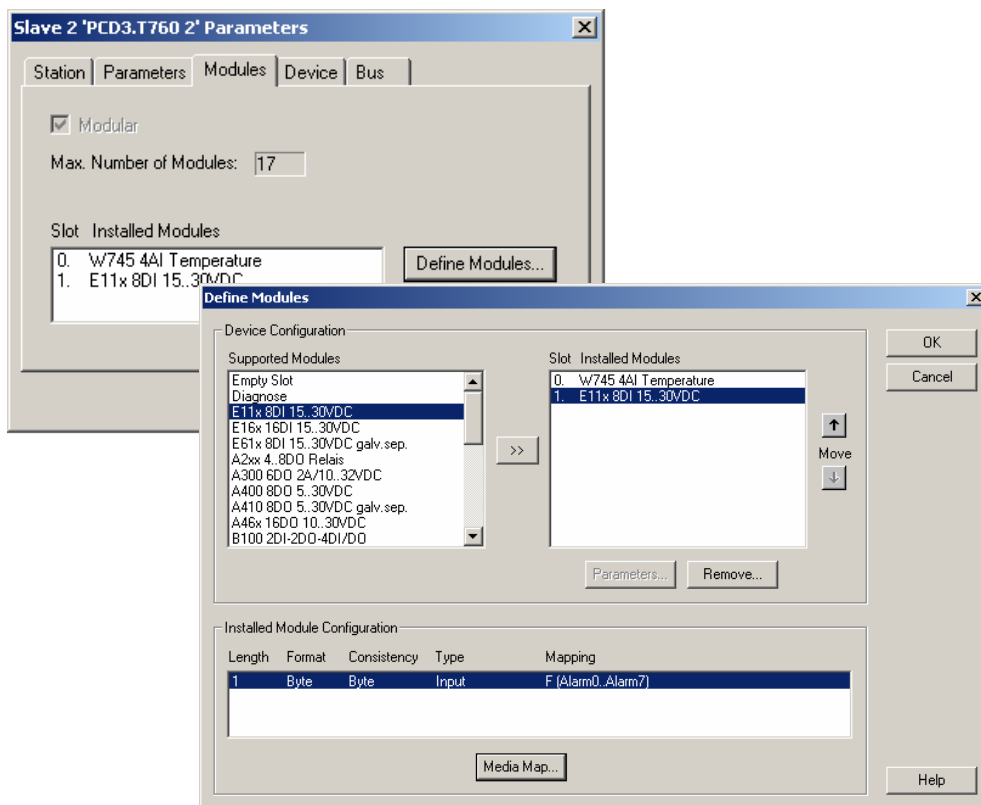


Une seule information doit être définie sur la station maître. C'est le chemin d'accès au CPU maître dans le quel S-Net doit sauvegarder le programme nécessaire à l'élaboration de la mémoire image pour la station maître.

Optionnellement, ce dialogue supporte aussi la modification du nom et de l'adresse du CPU.

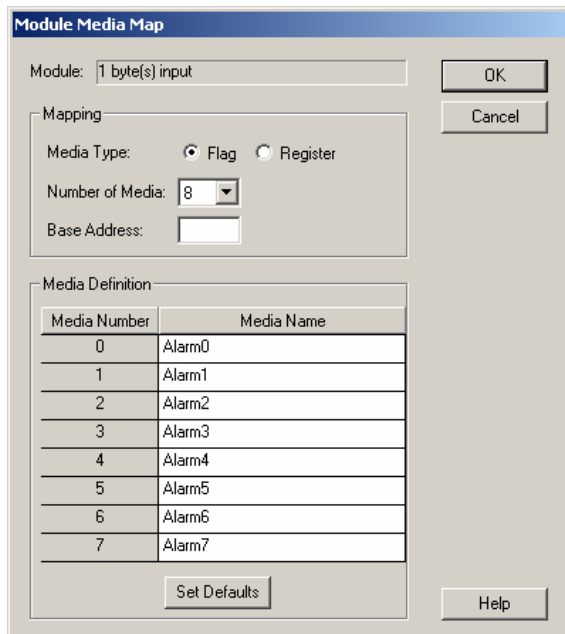
## 12.6 Configurer les stations esclaves

### 12.6.1 Configurer les modules d'entrées, sorties.



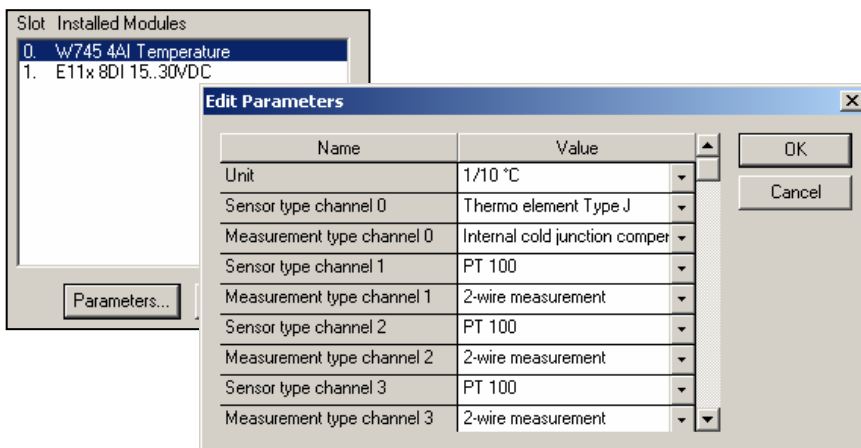
Pour chaque module d'entrée, sortie inséré sur la station esclave, sélectionner le type de module dans la liste *Supported modules* et l'ajouter dans la fenêtre *Slot Installed Modules* avec le bouton prévu à cet effet. Attention, le numéro de *Slot* et le type de module inséré dans la station esclave doivent toujours correspondre avec le numéro de *Slot* et le type de module présent dans la *fenêtre Slot Installed Modules* !

### 12.6.2 Configurer les symboles des données déportées



Pour chaque module présent dans la fenêtre *Slot Installed Modules*, sélectionner le module puis le bouton *Media Map* et définir les symboles de chaque donnée délivré par le module. Si nécessaire il est possible de définir l'adresse du premier flag ou registre à utiliser dans la station maître pour élaborer la mémoire image. Le plus simple est de ne rien définir, ainsi les adresses sont dynamiques.

### 12.6.3 Configurer les paramètres d'ajustages

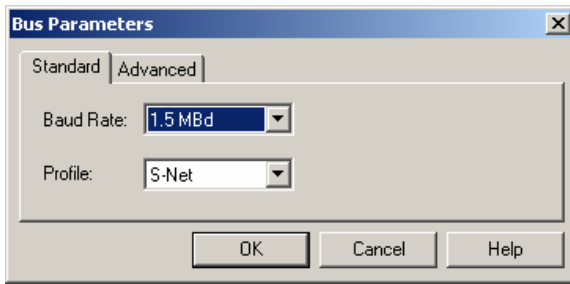


Avec certains modules comme les mesures analogiques, il peut être nécessaire de définir quelques paramètres supplémentaires pour la conversion de la mesure: unités de mesures, type de sondes de température, ...

Ces informations peuvent être ajustées à l'aide de la fenêtre d'ajustage affichée en sélectionnant le module puis le bouton *Parameters*

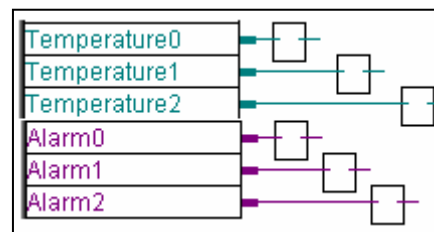
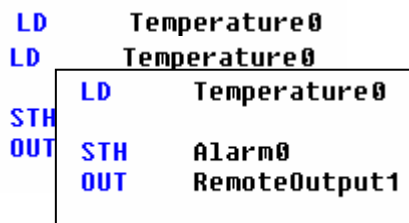
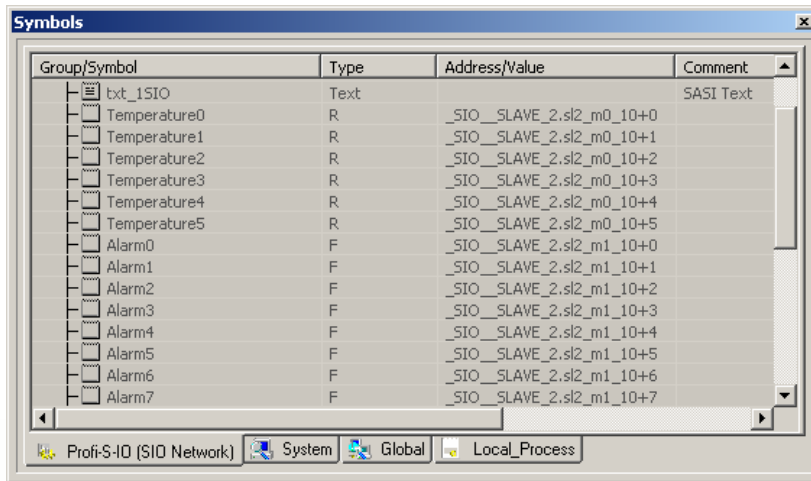


### 12.7 Configurer les paramètres du réseau.



Avec le menu *Edit Bus Parameter*, il est possible de définir la vitesse et le profile du réseau de communication.

### 12.8 Exploiter les symboles réseau dans un programme Fupla ou IL.



A la compilation du fichier S-Net (Menu Project,Compile) l'éditeur de symboles affiche une nouvelle feuille avec tous les symboles disponibles sur le réseau, Ces symboles peuvent être directement intégrés dans les fichiers Fupla et IL.

## 12.9 Autres références

Pour de plus amples informations, vous pouvez aussi vous référer aux manuels suivants :

- Profibus DP 26/ 765
- Profi-S-IO (en préparation)
- Exemple de projet Profi-S-IO installé avec votre PG5

## Caractéristiques techniques

### Références de commande

#### Caractéristiques techniques

Configuration logicielle minimale (système d'exploitation)	Windows 95 B Windows 98 seconde édition Windows NT 4.0 SP5 Windows 2000 Windows XP Préalables indispensables: installation de TCP/IP et de TAPI 2.0
Configuration matérielle minimale (compatible IBM PC)	Processeur Pentium 150 MHz 32 Mo de mémoire vive 30 Mo disponibles sur le disque dur Lecteur de CD-ROM
Jeu d'instructions PCD	Gestion de la totalité des 150 instructions du PCD
Boîtes de fonctions PG5	Offre standard de plus de 250 boîtes de fonctions
Modem	Modem de base et transmissions par modem implémentés dans le PG5. Des bibliothèques de fonctions modem étendues (mini-messages SMS et envoi vers boîtiers de radiomessagerie) sont également disponibles.
Langages de programmation	Liste d'instructions (IL), FUPLA (FBD) et GRAFTEC (SFC)
Unités centrales exploitables	Tous les modèles de SAIA®PCD (hormis la Série xx7)
Compatibilité PG5	Avec les programmes PG3 et PG4
Communication	Transmissions sur TCP/IP, SAIA®S-Bus, PROFIBUS DP, PROFIBUS FMS et LONWORKS® intégrées au PG5

#### Références de commande

Désignation	Description
PCD8.P59 000 M9	<b>Version complète du PG5</b> Cette fourniture comprend une licence d'exploitation sur disquette, la documentation et le programme sur cédérom.
PCD8.P59 000 M1	<b>Version de démonstration du PG5</b> Cette fourniture comprend la version complète du PG5, soumise à deux restrictions: désactivation de l'impression des fichiers programmes et traitement limité aux programmes ne dépassant pas 2000 lignes de code.

Saia-Burgess Controls SA  
Rue de la Gare 18  
CH-3280 Morat / Suisse

Téléphone 026 / 672 72 72  
Télécopieur 026 / 672 74 99

E-mail: [pcd@saia-burgess.com](mailto:pcd@saia-burgess.com)  
Homepage: [www.saia-burgess.com](http://www.saia-burgess.com)  
Support: [www.sbc-support.ch](http://www.sbc-support.ch)

Saia-Burgess Paris  
10, Bld. Louise Michel  
F-92230 Gennevilliers

Téléphone 01 / 46 88 07 70  
Télécopieur 01 / 46 88 07 99

E-mail: [gerard.fauvel@saia-burgess.com](mailto:gerard.fauvel@saia-burgess.com)  
Homepage: [www.saia-burgess.com](http://www.saia-burgess.com)

Branch Office:  
Saia-Burgess Benelux  
Mechelsesteenweg 277  
B-1800 Vilvoorde

Téléphone 02 / 456 06 20  
Télécopieur 02 / 460 50 44

E-mail: [office@saia-burgess.be](mailto:office@saia-burgess.be)  
Homepage: [www.saia-burgess.com](http://www.saia-burgess.com)