

Saia PG5 2.1

Manuel utilisateur

Les avantages des outils de programmation Saia PG5®

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

Les points forts des outils de programmation Saia 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, SBC 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 SBC S-Bus, PROFIBUS DP, LONWORKS®.
- **De puissantes bibliothèques complémentaires** pour accroître la richesse fonctionnelle du PG5.

Table des matières par chapitre

- 1 PCD-Mise en service rapide
- 2 Gestionnaire de projets
- 3 Device Configurator
- 4 PCD – Types de données
- 5 Editeur de symboles
- 6 Programmer avec Fupla
- 7 Structuration du programme
- 8 Programmer en Graftec
- 9 Programmer en IL (Instruction List)
- 10 Outils complémentaires
- 11 Saia PCD Networks
- 12 Profi-S-Bus
- 13 Ether-S-Bus
- 15 Profi-S-IO

Avant Propos

Ce document n'est pas un manuel de mise en service détaillé des automates SaiaPCD®, 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 2.0
- CD de documentation 26/803
- 1 automate PCD2.M5540 ¹
- 1 module de 8 entrées digitales PCD2.E110
- 1 module de 8 sorties digitales PCD2.A400
- 1 câble USB

Toutes les instructions nécessaires à l'installation du PG5 2.1 sur votre ordinateur sont fournies avec le CD PG5 version 2.0 (Voir sous : CD:\PG5_InstallationGuide_F.pdf)

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 AG

¹ un autre PCD peut aussi convenir

Table des matières

1	PCD – MISE EN SERVICE RAPIDE	3
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.M5540	4
1.2.3	Assemblage du PCD2.M5540	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	7
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.M5540 avec la programmation, les tests et l'outil de programmation Saia 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 Installation du matériel

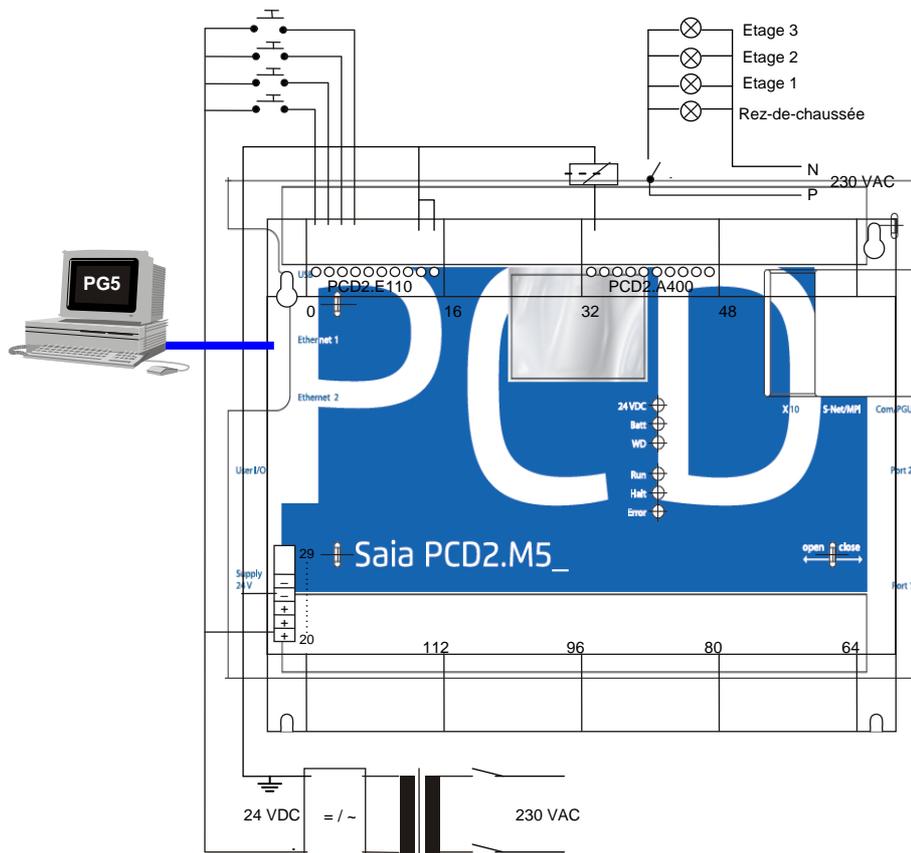
1.2.1 Exemple: éclairage des escaliers

Il est temps de réaliser un premier exemple d'application. Nous proposons d'automatiser l'éclairage des escaliers d'une maison d'habitation de 3 étages. A chaque étage se trouve un interrupteur de lumière de type impulsion. Une courte impulsion sur l'un des interrupteurs permet d'allumer la lumière pour une période de 5 minutes.

Les interrupteurs sont câblés sur quatre entrées du PCD. Nous choisissons les entrées 0, 1, 2, 3. (I 0, I 1, I 2, I 3)

La lumière de chacun des étages est allumée par un relais commandé à partir d'une seule sortie du PCD. Nous choisissons de câbler ce relais sur la sortie 32 (O 32).

1.2.2 Schéma d'implantation du PCD2.M5540

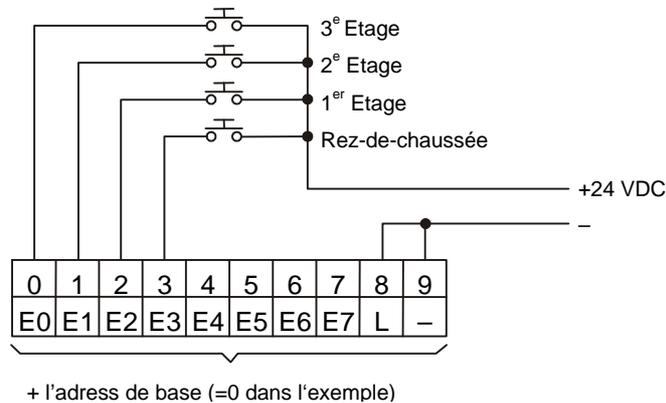


1.2.3 Assemblage du PCD2.M5540

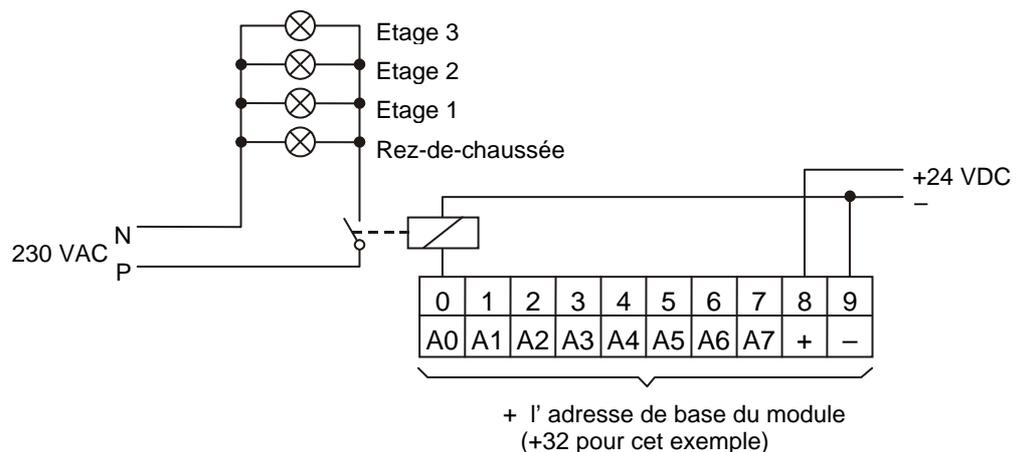
1. Retirer le couvercle et insérer la batterie au lithium de 3.0 V
2. Insérer un module PCD2.E110 à l'adresse de base 0. Cela permet de disposer de 8 entrées digitales 24 VDC aux adresses I0 à I7.
3. Insérer un module PCD2.A400 à l'adresse de base 32. Cela permet de disposer de 8 sorties digitales 24 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 USB de l'ordinateur sur le PCD.

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

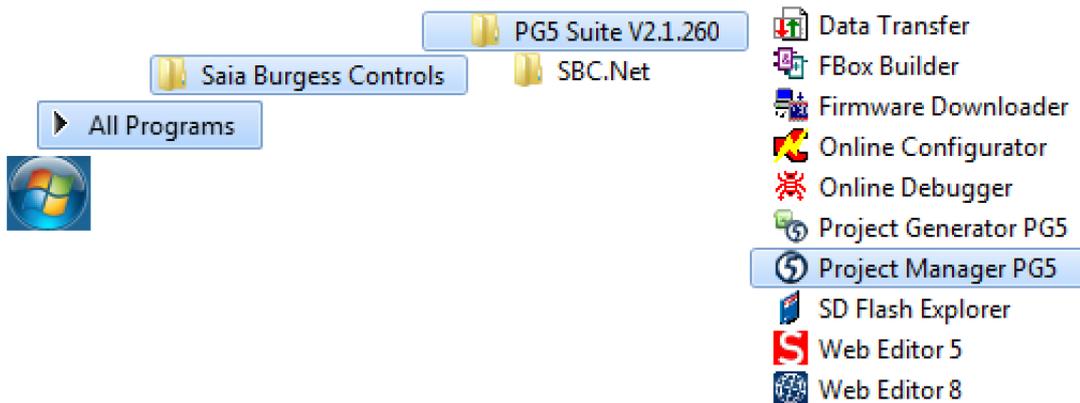
1.3.1 Installation du software

Installer le software Saia PG5® sur l'ordinateur selon les directives livrées avec le CD de distribution. (CD:\ PG5_InstallationGuide_F.pdf)

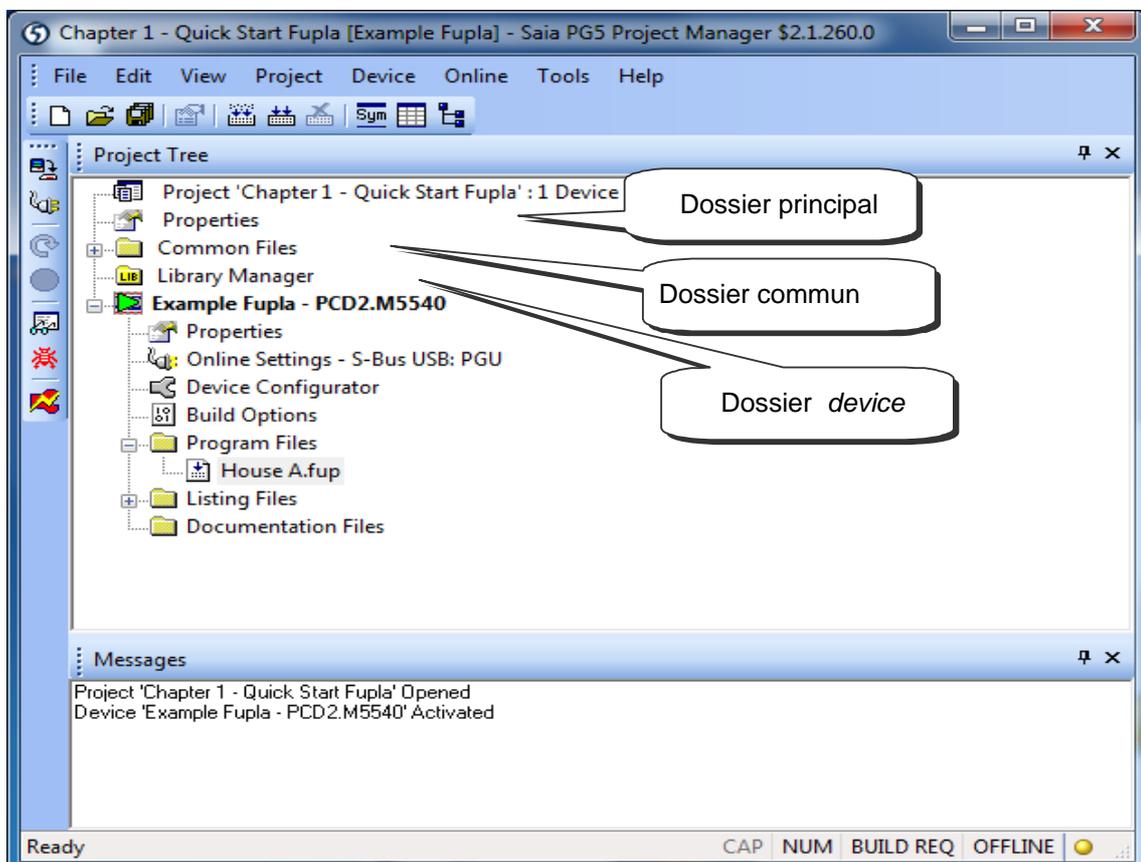
1.3.2 Démarrer PG5

Démarrer le gestionnaire de projet du PG5 avec le menu:

Start → Programs → Saia-Burgess → PG5 Suite 2.0 → Project Manager PG5



La fenêtre de gestion du projet est alors affichée. La sous fenêtre *Project Tree*, 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 *devices* qu'il contient.
- Le dossier commun, dans lequel, il est possible de déposer les programmes communs à tous les *devices*.
- Puis le ou les dossiers *devices*.

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

- Un dossier *Device Configurator* 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...*

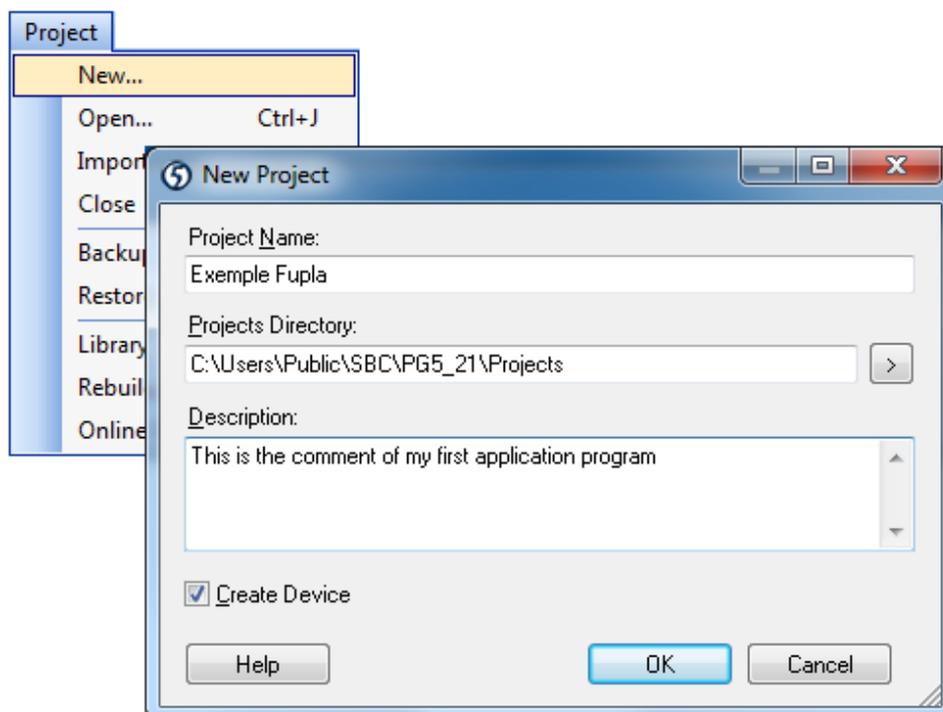
Il recherche tous les fichiers projet (.5pj) présents dans le répertoire des projets et les affiche dans une liste. Sélectionner l'un des projets de la liste ou marquer un projet de la liste et presser le bouton *Open*. Une autre alternative, presser le bouton *Browse* et chercher directement le fichier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

1.3.3 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 Device* et valider avec le bouton *OK*.

Créer un nouveau projet

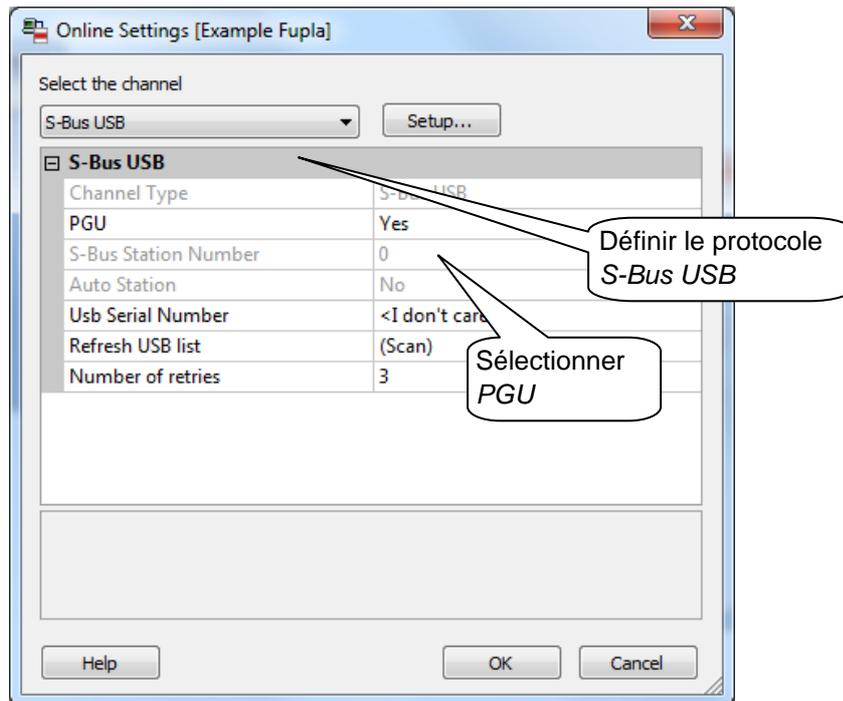


1.3.4 Configurations

Avant de travailler sur un des *devices* 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 *Online 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.

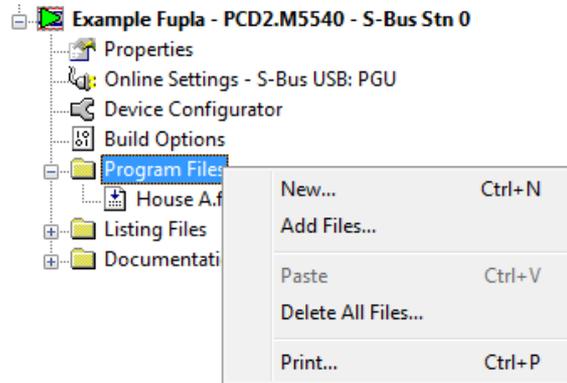
Channel S-Bus USB



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:

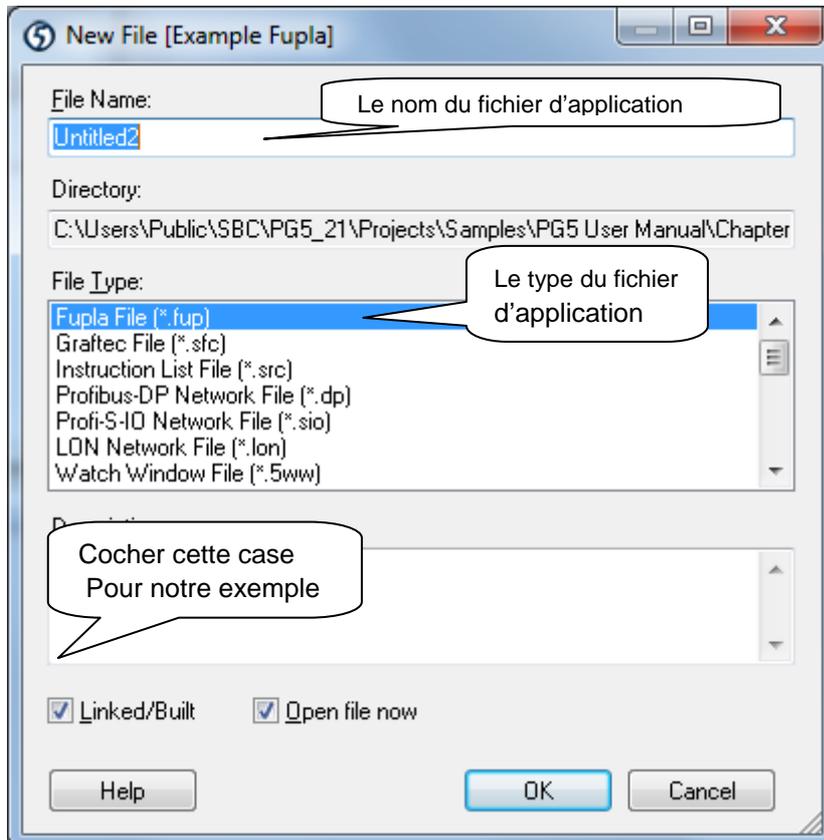
Marquer le dossier *Program File* de la fenêtre *Project Tree*, cliquer sur le bouton droite de la souris pour faire apparaître le menu de contexte et sélectionner le menu *New...*



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.

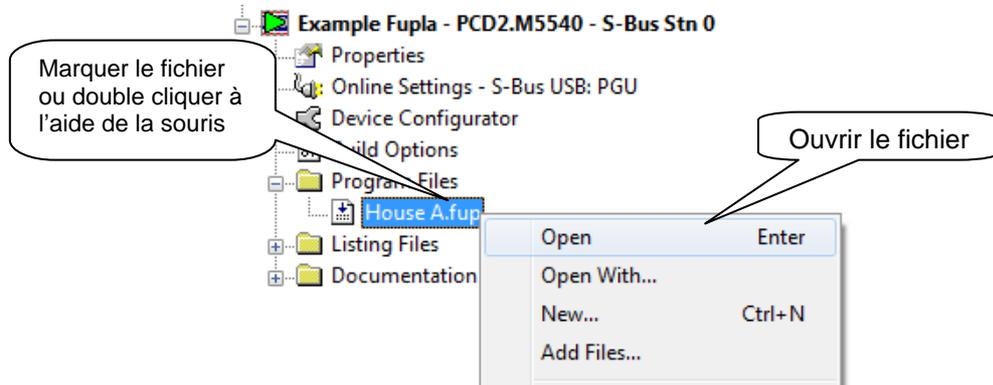
De nombreux éditeurs sont à disposition pour écrire les applications PCD. Le choix de l'éditeur dépend des préférences de l'utilisateur et du type d'application à réaliser. Pour notre première application nous choisissons l'éditeur le plus couramment employé et le plus universel: *Fupla File (*.fup)*



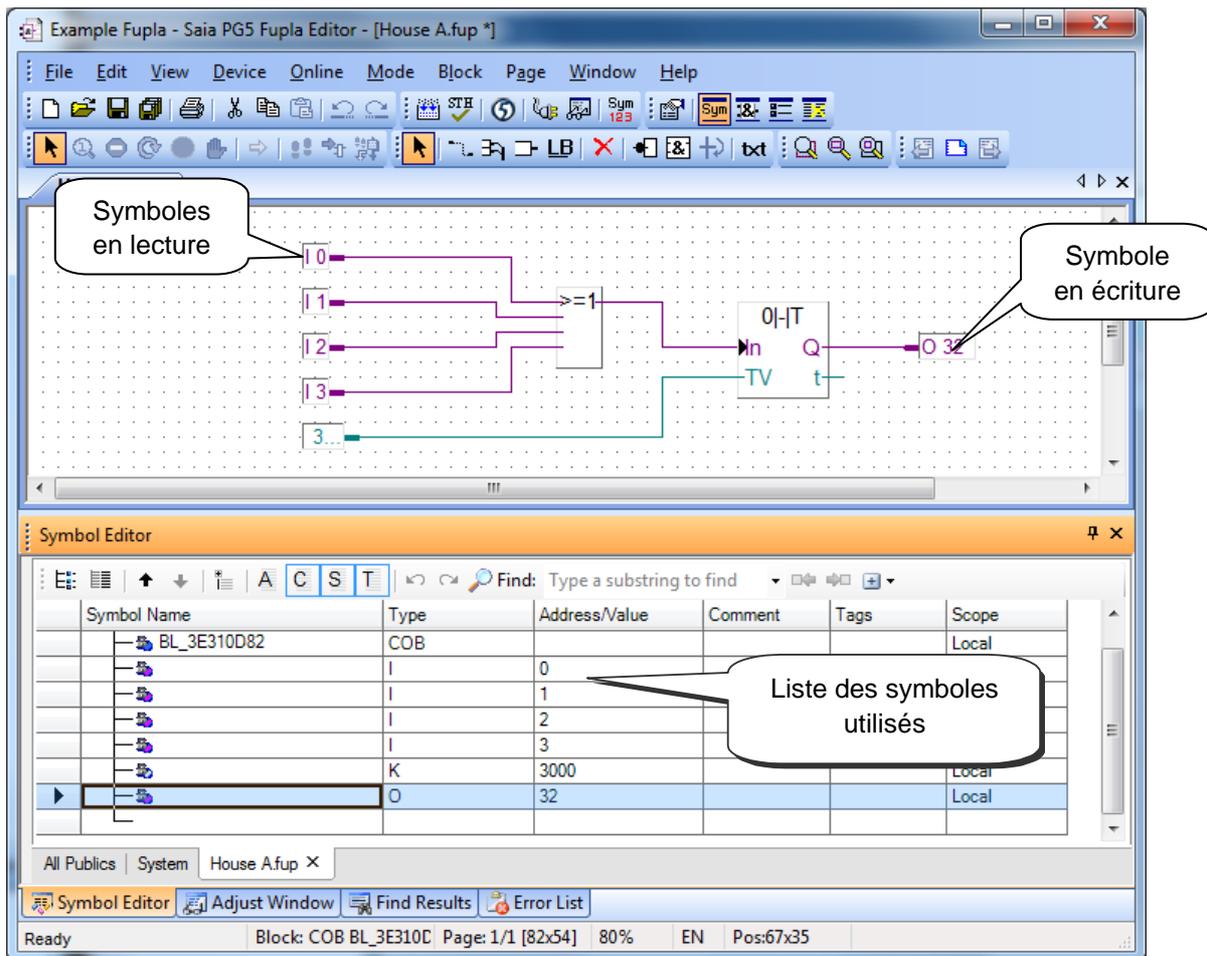
1.3.6 Ouvrir un Fichier

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 Tree*, cliquer sur le bouton droite 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).



Add
Connectors

Pour ajouter un connecteur et son symbole sur une page Fupla, sélectionner le bouton *Add 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
Symbols Editor

Nous remarquons que la liste des symboles introduits 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..., Workspace, 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 Unused connectors*.

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

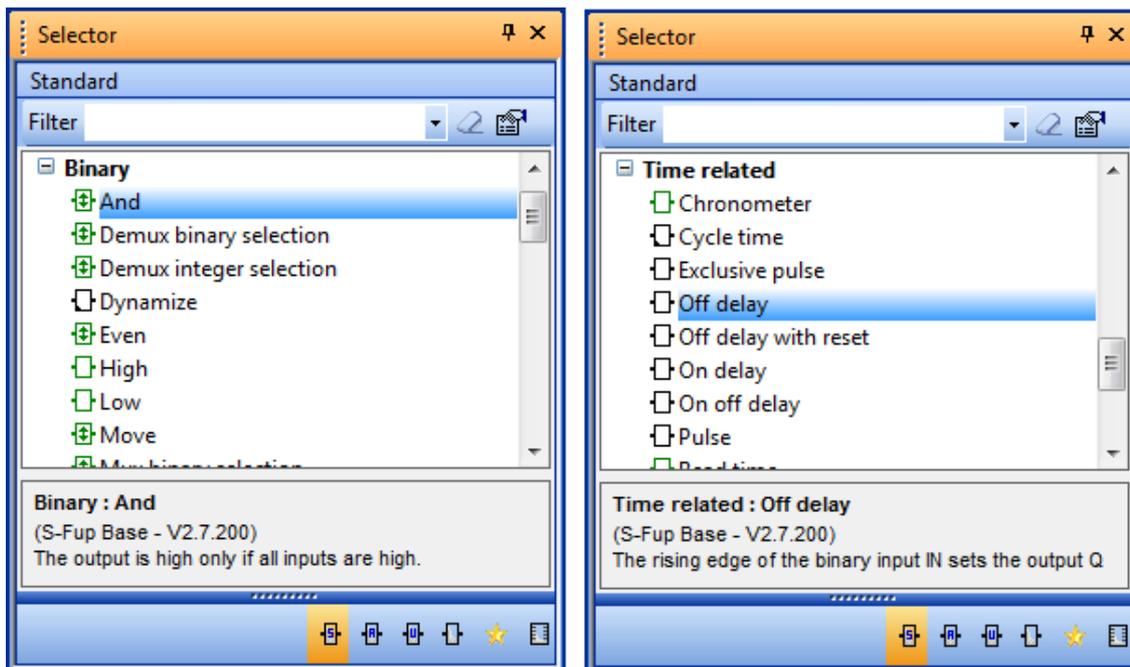
Edition des fonctions du programme

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 *Fboxes Selector...* que nous ouvrons avec le bouton *Show/Hide Selector Window*.



Show/Hide Selector



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 *Binary*.

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 *Time related*.

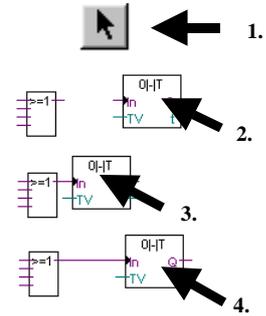
De plus amples informations concernant le groupe de fonction ou la fonction sélectionnée sont disponibles avec la touche F1 du clavier. Après avoir sélectionné une des fonctions graphiques de la fenêtre *Fboxes Selector...*, placer la souris sur la page d'édition du programme et cliquer sur le bouton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

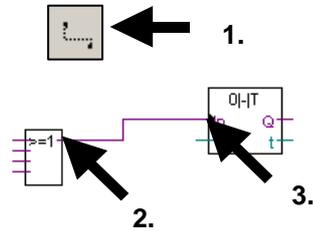
Première possibilité pour les liaisons horizontales

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 *Line 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 Object* 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*)

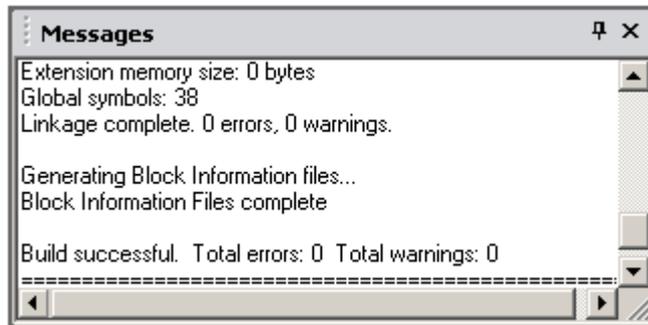


*Rebuild
All Files*

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 *Device, Build Changed Files* ou le bouton *Device, 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 *Online Settings* et *Device Configurator* 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.

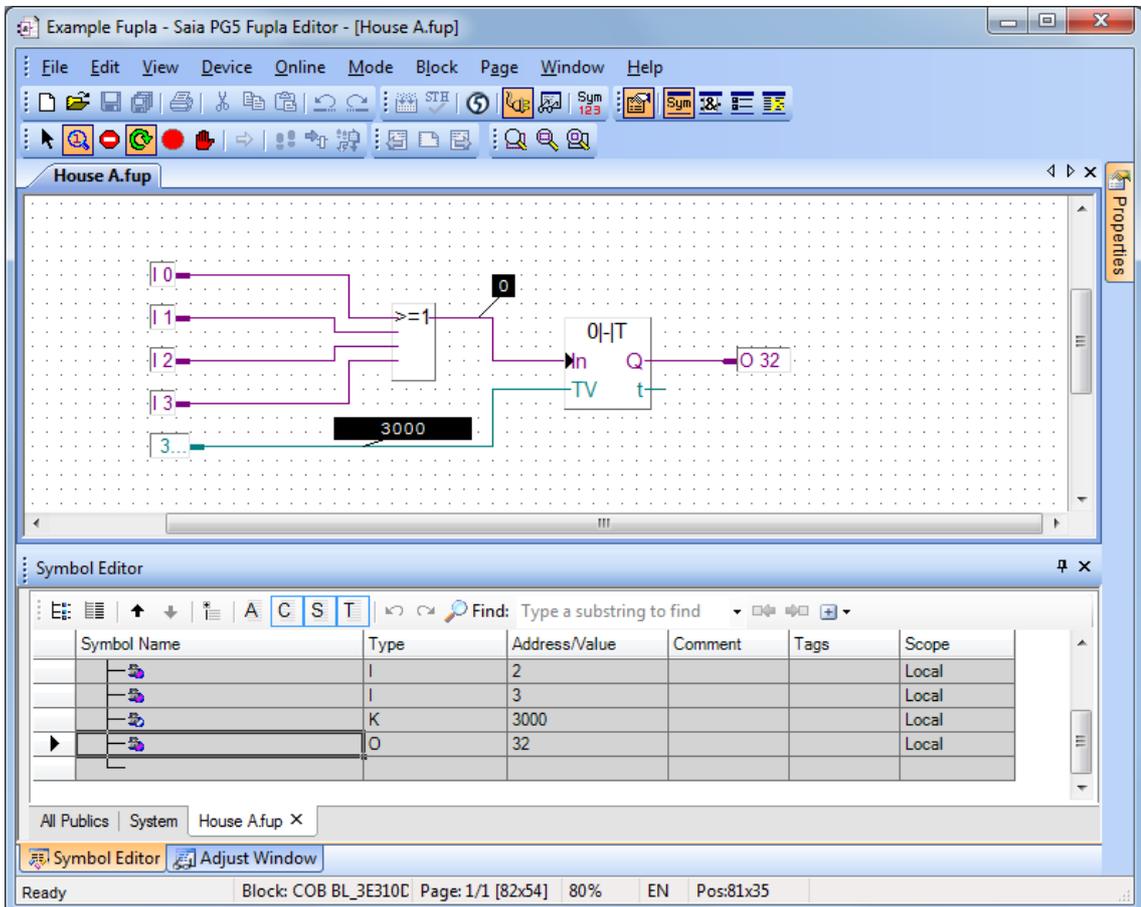


A la sélection du bouton *Stop*, la lampe *RUN* est éteinte, le PCD stop l'exécution du programme utilisateur.

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 *Add 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

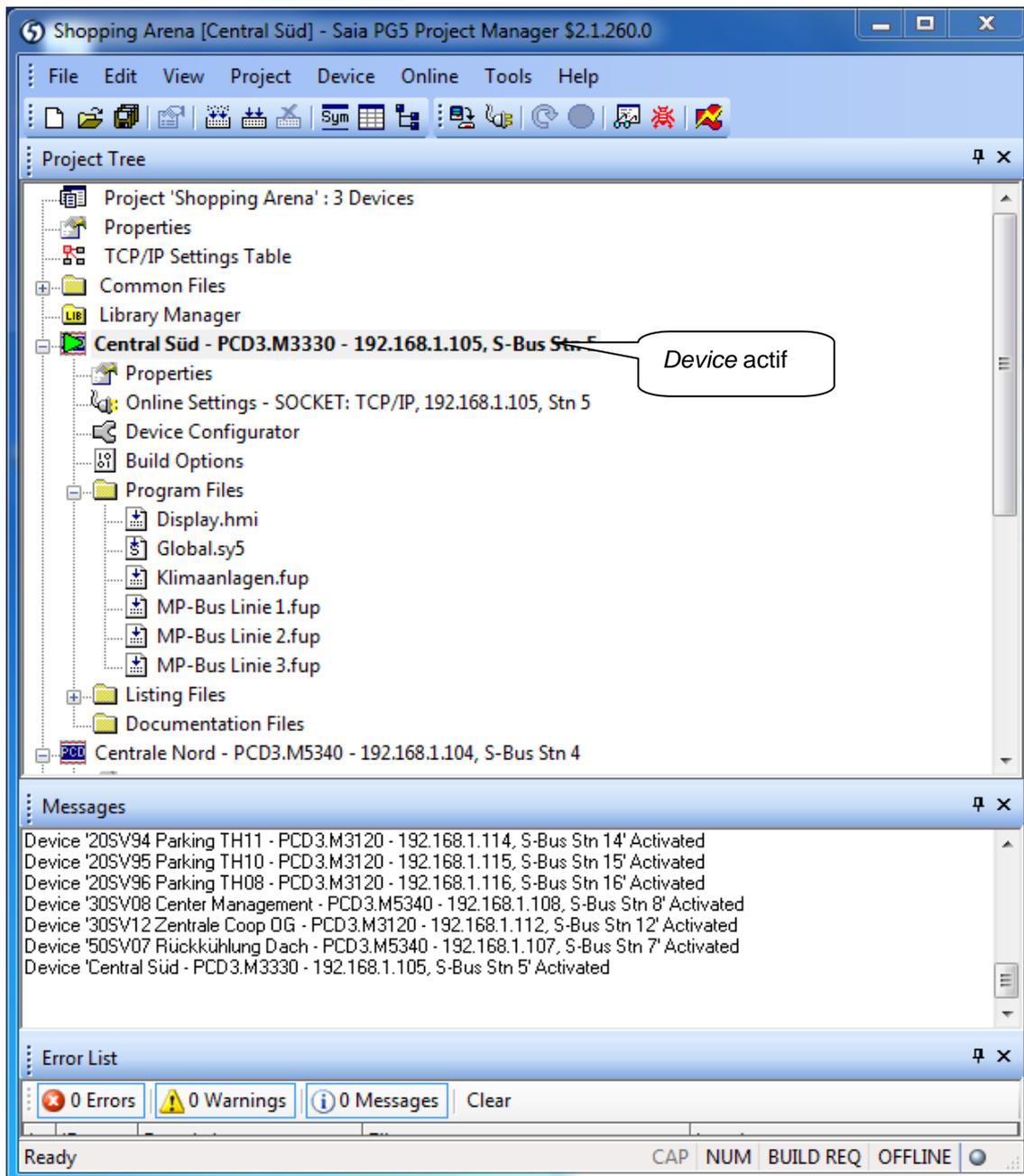
TABLE DES MATIÈRES	1
2 GESTIONNAIRE DE PROJETS	3
2.1 Organisation d'un projet	3
2.1.1 Ouvrir un projet	4
2.1.2 Créer un nouveau projet	4
2.1.3 Sauvegarde du projet sur le PC.....	5
2.1.4 Compresser, décompresser un projet ou <i>device</i>	6
2.2 La fenêtre <i>Project</i>.....	7
2.2.1 Dossier projet.....	7
2.2.2 Dossier commun	8
2.2.3 Dossier librairie	8
2.2.4 Dossier <i>device</i>	9
2.2.5 Settings Online	9
2.2.6 Raccorder le PC à l'automate PCD	10
2.2.7 <i>Device Configurator</i>	11
2.2.8 Build Options.....	11
2.2.9 Dossier Programmes.....	13
2.2.10 Types de fichiers.....	15
2.3 Construction du programme. (Build)	17
2.3.1 <i>Construction des fichier modifiés, Rebuild all, Build, Rebuild All Programs</i>	17
2.3.2 Options du <i>Build</i>	18
2.4 Fenêtre des messages.....	19
2.5 Charger le programme dans l'automate	20
2.5.1 <i>Download Program</i>	20
2.5.2 Options du <i>Download</i>	21
2.6 Commandes pour plusieurs <i>devices</i>	22
2.7 Self downloader	23
2.7.1 Préparer un fichier *.sd5.....	23
2.7.2 Download d'un fichier *.sd5	25
2.8 Mémoire Flash Backup	26
2.8.1 Sauvegarde du programme exécutable	26
2.8.2 Sauvegarde du projet source.....	27
Backup <i>Source</i> du projet	27
FTP downloader	28
Restauration du projet.....	30
2.8.3 Sauvegarde de données dans un fichier	30
2.9 Les fenêtres <i>View</i>	31
2.9.1 Structure des blocs d'organisation.....	31
2.9.2 Liste des symboles.....	32

2.9.3	Référence croisée.....	32
2.10	L’outil <i>Online Configurator</i>	33
2.10.1	<i>Configuration</i>	33
2.10.2	L’ historique PCD.....	34
2.10.3	Ajuster l’horloge de l’automate	34
2.10.4	Sauvegarde du programme en mémoire RAM.	35
2.10.5	<i>Create Diagnostic file</i>	36
2.11	Mise à jour du firmware. (<i>Firmware Downloader</i>)	36

2 Gestionnaire de projets

2.1 Organisation d'un projet

Les processus d'automatisation modernes sont composés de nombreux automates raccordés en réseaux. Par exemple, un processus de gestion technique de bâtiment peut compter un automate pour chaque fonction partielle de l'ensemble du projet: gestion de l'éclairage, contrôle du chauffage, contrôle de la ventilation, de la porte automatique du parking souterrain, ...



Le Saia PG5 Project Manager offre une vue globale de toutes les informations relatives aux automates qui composent le processus dans un projet unique. De nombreuses manipulations sur les automates qui constituent le projet peuvent être réalisées.

Par exemple, ajouter au projet de nouveaux fichiers programmes, ouvrir les fichiers pour rédiger les programmes, configurer les *devices*, construire et charger les programmes dans les automates, sauvegarder et restaurer les fichiers sources du projet dans un fichier *backup*, signaler les erreurs et alarmes lors de la constructions des programmes,...

La fenêtre *Project*, représente les automates du processus dans lesquels sont rassemblés les fichiers de configurations et les programmes qui les constituent. Pour afficher cette fenêtre sélectionner le menu *View, Project Tree*.

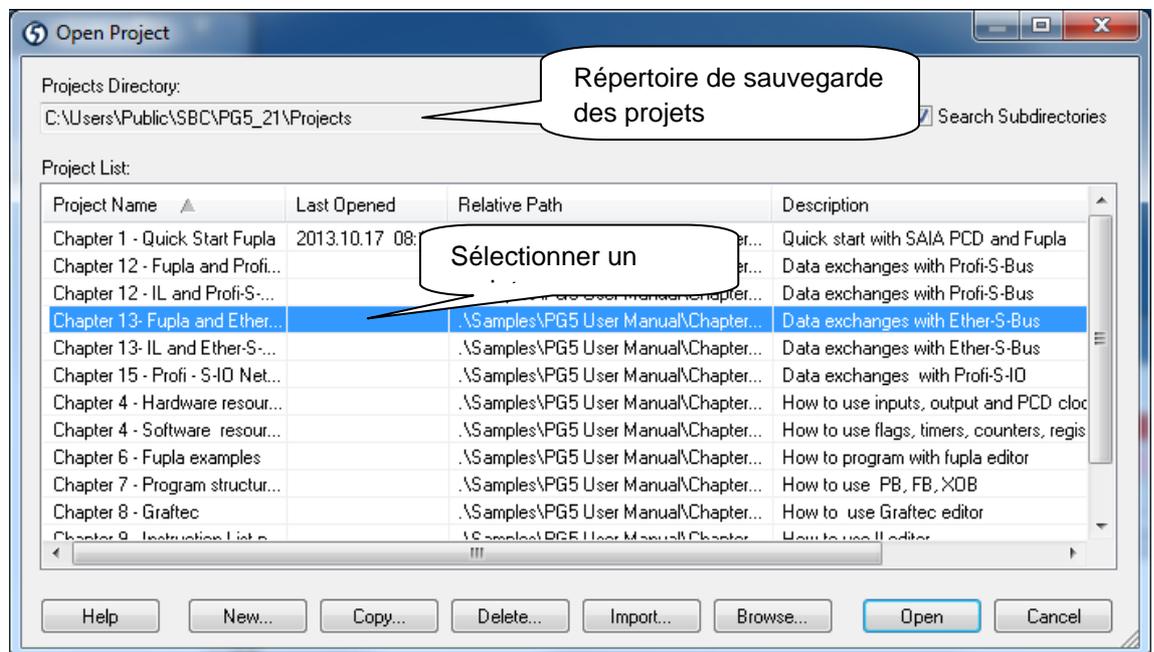
Un *device* actif est toujours marqué par un triangle vert. Tous les menus et boutons de la *Tool Bar* utilisent les configurations et fichiers programmes du *device* actif. Pour changer de *device* actif, sélectionner un autre *device* dans la structure du projet.

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*.

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.

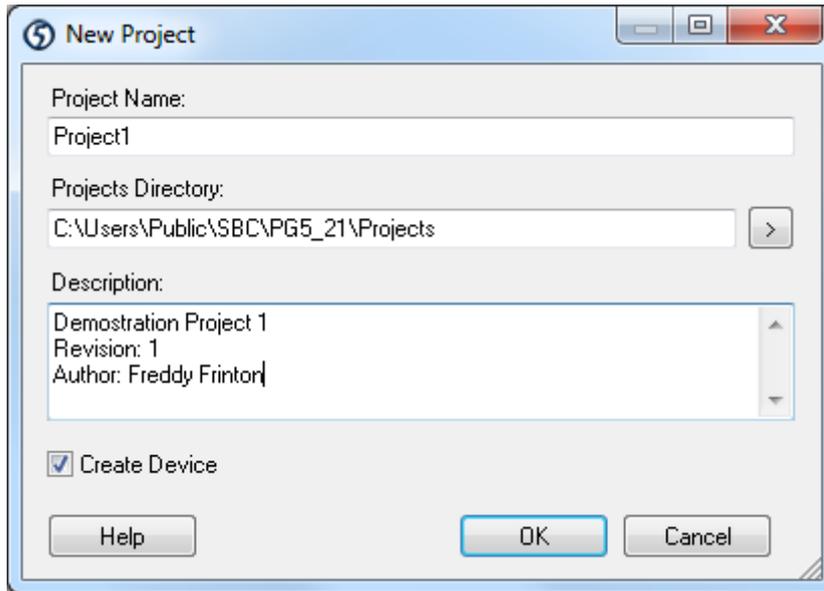
2.1.1 Ouvrir un projet

Le PG5 est livré avec tous les exemples de ce manuel. Le menu *Project, Open...* recherche tous les fichiers projet (.saia5pj) 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 *Device* (.saia5pc).



2.1.2 Créer un nouveau projet

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



- Project Name:** nom du nouveau projet
- Project Directory:** chemin au répertoire du projet
- Description:** courte description du projet
- Create Device:** crée un nouveau projet avec ou sans *devices*

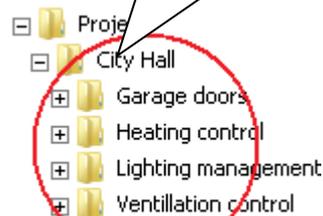
2.1.3 Sauvegarde du projet sur le PC

Par défaut, les projets sont sauvegardés dans le répertoire *Windows* suivants :
 XP: C:\Documents and Settings\All Users\Documents\Saia-Burgess\PG5 2_0\Projects
 Vista: C:\Users\Public\Documents\Saia-Burgess\PG5 2_0\Projects

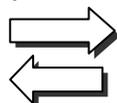
Si nécessaire le chemin de sauvegarde des projets ainsi que des bibliothèques sont ajustés à l'aide du menu *Tools, Options*:



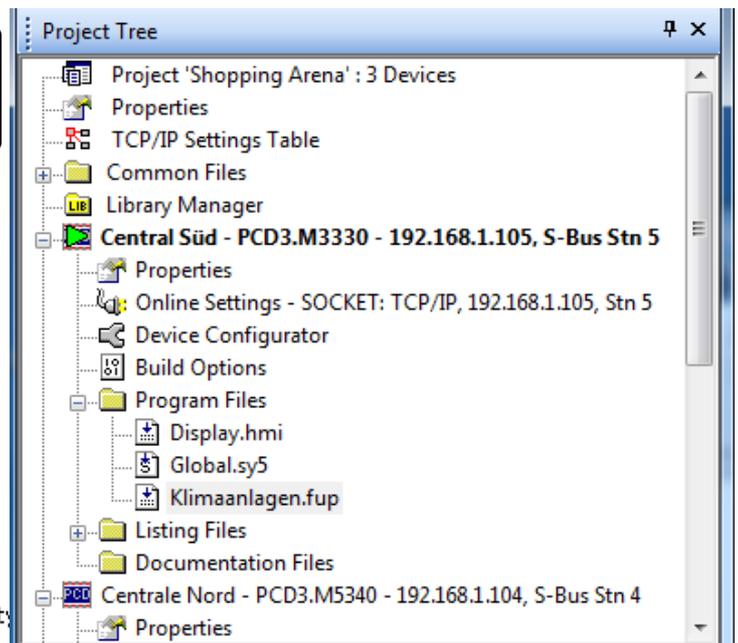
Le projet est sauvegardé dans un répertoire qui porte le nom du projet. Il est composé d'un sous répertoire par *Device*.



Project, Backup ...



Project, Restore ...



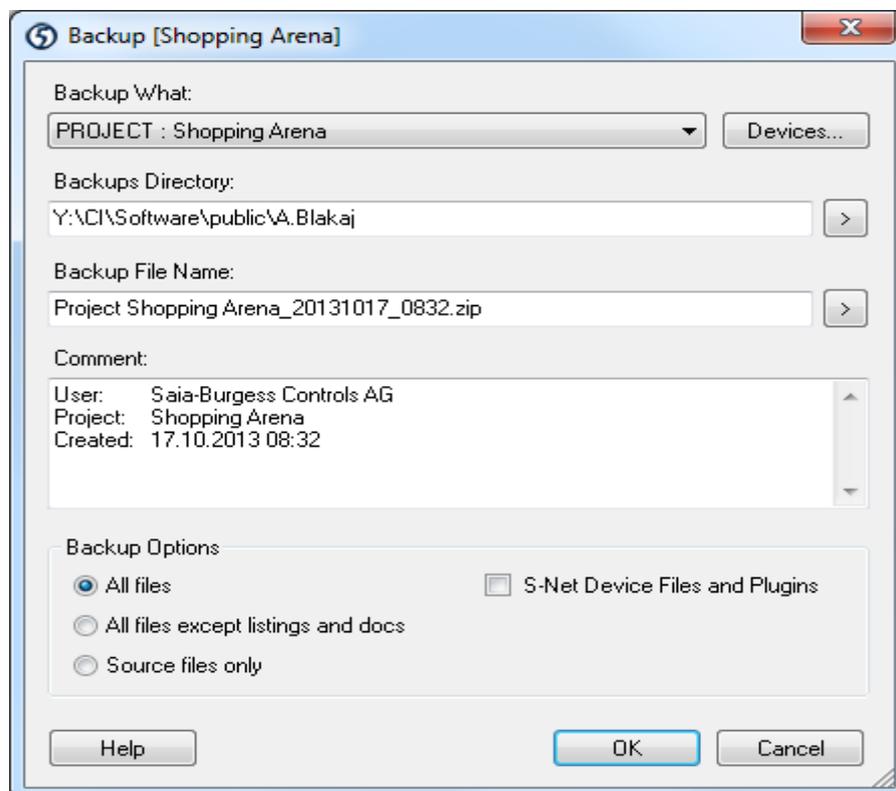
Pour sauvegarder un projet, il est nécessaire de conserver toute la structure des répertoires et des fichiers qui constituent le projet.

Des outils permettent de compresser toutes les informations relatives à un projet ou *device* vers un fichier au format *.zip et inversement.

2.1.4 Compresser, décompresser un projet ou *device*

Le menu *Project, Backup* compresses tout le projet ou uniquement un des *devices* du répertoire des projets dans un fichier au format *.zip

Le menu *Project, Restore* restaure le projet/*device* sauvegardé dans un fichier *.ZIP vers le répertoire des projets.



Backup Project or single Device :

Cette option sauvegarde l'ensemble du projet ou un des *devices*. Par défaut c'est le projet qui est proposé.

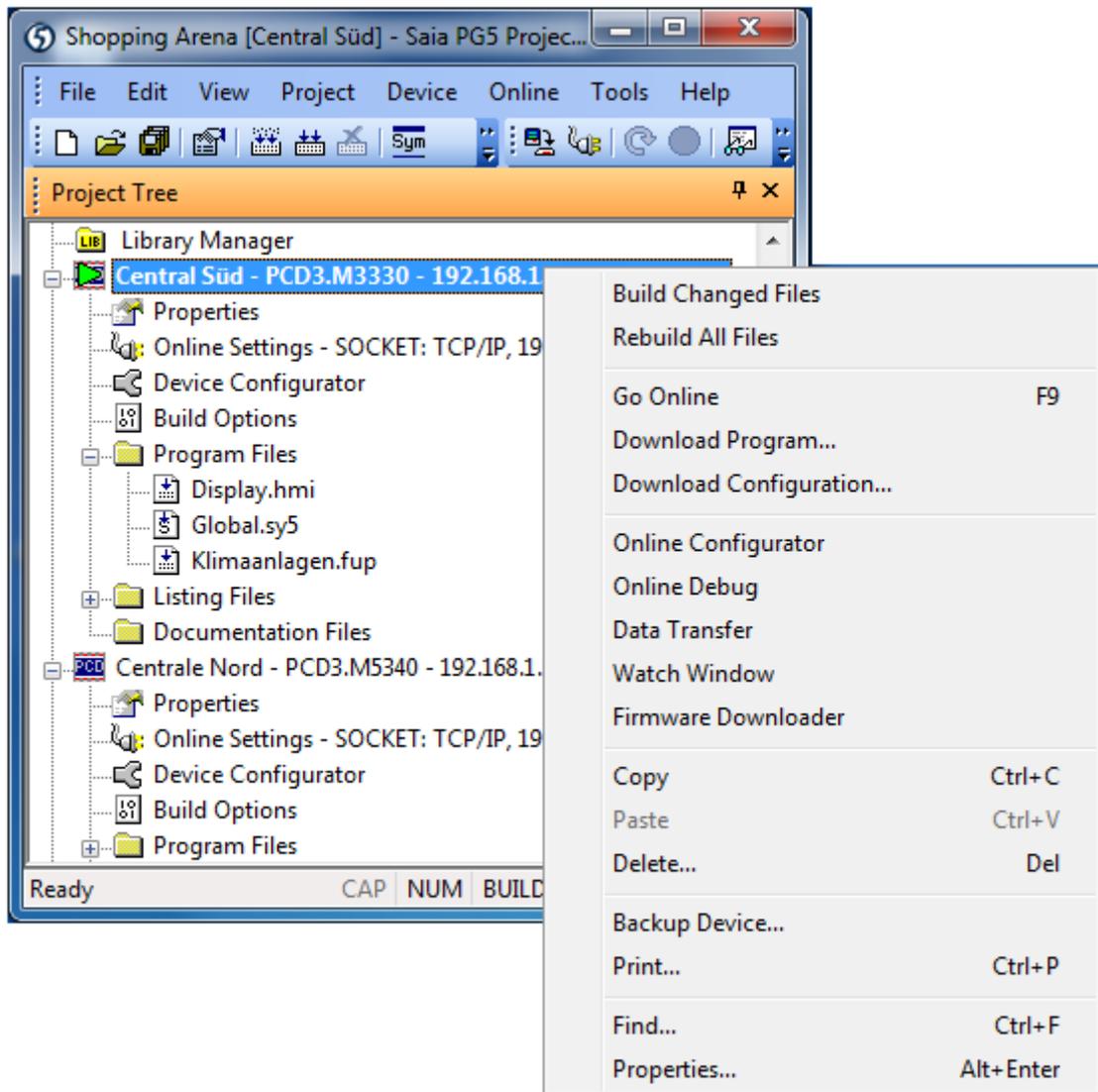
To compressed file : nom du fichier de sauvegarde .zip

Comment : commentaires libres

Backup what :

Définit le contenu de la sauvegarde, il n'est pas nécessaire de conserver tous les fichiers. Seul les fichiers sources sont importants.

2.2 La fenêtre *Project*



2.2.1 Dossier projet



Le dossier principal représente le projet avec son nom et le nombre de *devices* qui le constitue. La sélection du dossier projet permet de réaliser les actions suivantes avec le menu principal ou le menu de contexte.

Menu *Device, New*

Ajoute un nouveau *device* au projet.

Menu *Device, Import ...*

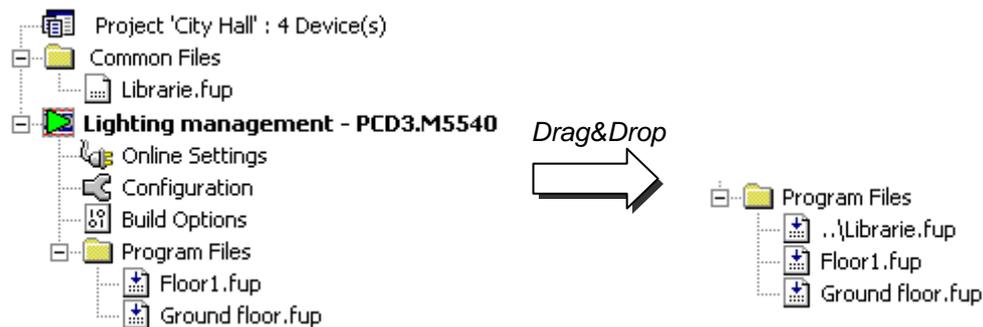
Importer les *devices* présents dans d'autres projets que celui ouvert.

Si le *device* est importé d'un ancien projet PG4 ou PG5, il est converti avec le nouveau format.

Menu *File, Properties*

Modifie les informations relatives au dossier projet: nom, description, protection contre l'écriture.

2.2.2 Dossier commun



Ce dossier est prévu pour recevoir les fichiers partagés avec plusieurs *devices*. Ces fichiers peuvent être copiés, collés ou simplement glissés dans le dossier programme des *devices* qui les utilisent. 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 *device*. Dans les deux cas, l'utilisateur modifie le même fichier et les corrections sont valables pour tous les *devices* qui y sont liés.

La sélection du dossier commun permet de réaliser les actions suivantes avec le menu principal ou le menu de contexte.

Menu **File, New**

Ajoute un nouveau fichier au dossier.

Contexte menu **Add Files**

Importe 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.

2.2.3 Dossier librairie



Ce dossier donne une vue des librairies disponibles dans le répertoire des librairies PG5 et du projet.

La liste *Installed libraries*, représente les librairies installées avec PG5 et disponibles pour tous les projets. Le chemin d'installation de ces librairies est défini par le menu *Tools, Options, Directories, Library Directory*.

La liste *Libraries Copied to Project*, représente les librairies sauvegardées avec le projet ouvert. Seul le projet ouvert peut faire usage de ces librairies, leurs chemins d'installation correspondent au répertoire du projet défini par le menu *Tools, Options, Directories, Projects Directory*.

Les librairies PG5 sont sauvegardées vers le projet par *Drag&Drop* de la librairie vers la liste des libraires correspondant au projet ou par la sélection de la librairie et du bouton *copy to project*.

Les librairies utilisateurs qui ne sont pas distribuées en standard avec le PG5, devraient être systématiquement sauvegardées avec le projet pour toujours garantir que les sauvegardes de projets sont complètes et permettent si nécessaire de réaliser un build.

Au cas où la version de la librairie présente dans PG5 et le projet sont différentes, le *build* utilise la librairie sélectionnées *Use*.

2.2.4 Dossier *device*



Chaque dossier *device* rassemble les configurations et les programmes d'un automate du projet. La sélection d'un dossier *device* permet de réaliser les actions suivantes avec le menu principal ou le menu de contexte.

Menu *Device, Set Active*



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

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

Menu *File, Properties*

Permet de modifier les informations relatives au dossier *device*: nom, description, protection contre l'écriture des fichiers présents dans le *device*.

Menu *Edit, Copy, Past, Delete*

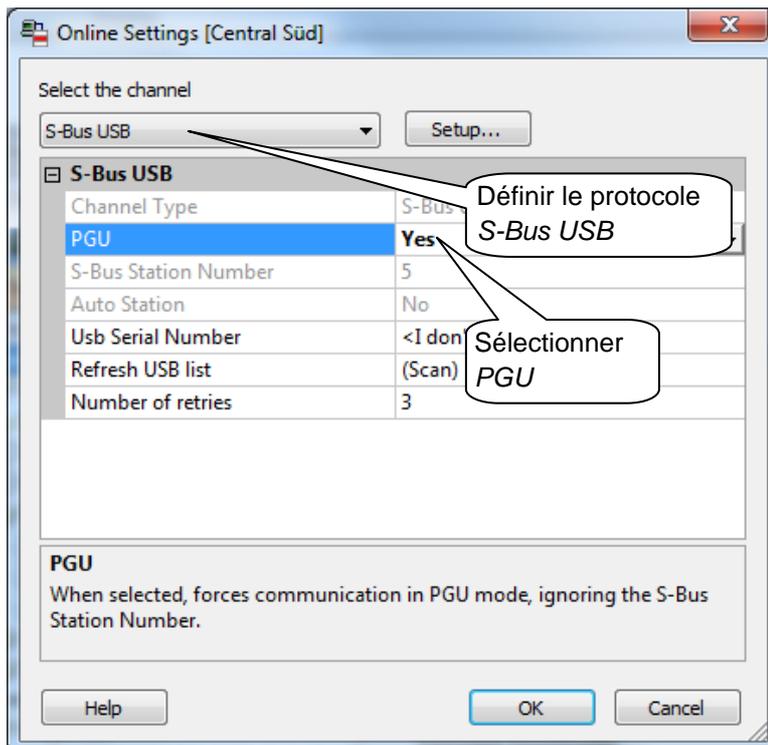
Copier/coller permet de dupliquer un *device* dans le projet. *Delete* permet de le supprimer un *device* du projet, de le placer dans la corbeille.

2.2.5 Settings Online



Le dossier *Settings Online* définit les paramètres de communication de 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 le *Device Configurator*.

Channel S-Bus USB



Le bouton *Add* permet de créer ses propres *channels* de communications avec le type et les paramètres de communications de votre choix ensuite disponibles dans la fenêtre *Online settings*.

2.2.6 Raccorder le PC à l'automate PCD

Channel S-Bus USB

L'interface USB n'est disponible que pour les nouveaux *devices* PCD2 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.



2.2.7 Device Configurator



Le *Device* configurateur permet de configurer les paramètres matériel de l'automate: le type de *Device*, la mémoire, les canaux de communications, les modules d'entrée/sorties mais vérifie aussi la consommation des modules d'entrées/sortie sur l'alimentation interne du PCD et imprime les étiquettes pour coller sur les modules E/S .

Pour mettre en service un automate PCD il est nécessaire de définir au moins le type de PCD et sa configuration mémoire. Les autres configurations pourront être ensuite complétée en fonction des services nécessaires tel que des réseaux de communication ou la gestion des E/S.



La méthode la plus simple de réaliser la configuration consiste à relier l'ordinateur au PCD avec un USB et de lire la configuration déjà présente dans la mémoire du PCD avec le menu *Online, Upload Configuration...* ou le bouton correspondant de la *tool bar*. Si la mémoire ne devait contenir aucune configuration, le firmware PCD se charge de retourner les informations qui conviennent. Puis vérifier les configurations correspondant à votre application.

Pour de plus amples renseignements, se reporter à la description du Device Configurateur.

2.2.8 Build Options

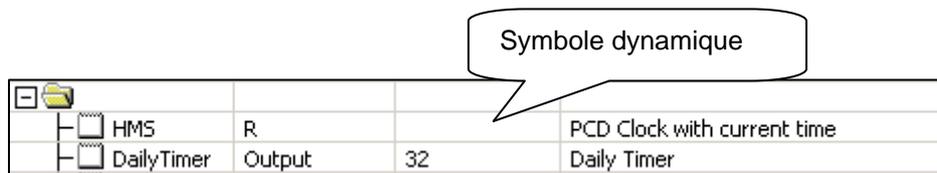


Media Allocation	
Last Timer	31
Timer Timebase in milliseconds (10..10000)	100
Has Volatile Flags	True
Last Volatile Flag	2999
Dynamic Registers	2000; 4095
Dynamic Texts	3000; 3499
Dynamic Data Blocks	3500; 3999
Dynamic RAM Texts	2000; 2499
Dynamic RAM Data Blocks	2500; 2999
Dynamic Timers	5; 31
Dynamic Counters	1400; 1599
Dynamic Volatile Flags	2500; 2999
First	2500
Last	2999
Dynamic Nonvolatile Flags	7500; 8191

Dynamic Registers, Texts, ...

Cette fenêtre supporte la réservation d'une plage d'adresses pour les registres, textes, blocs de données, temporisateurs, compteurs 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:



	HMS	R		PCD Clock with current time
	DailyTimer	Output	32	Daily Timer

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 2368: 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 principale de type EPROM ou Flash, 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.

Last Timer

Les temporisateurs et compteurs partagent la même plage d'adresse. L'adresse du dernier temporisateur (**last Timer**) détermine la répartition entre les temporisateurs et compteurs. (Génère une instruction DEFTC). Les adresses des temporisateurs/compteurs dynamiques doivent être définies en dessous/en dessus de cette limite. Par exemple, si le dernier temporisateur est 31, les temporisateurs sont disponibles avec les adresses 0...31 et les compteurs 32...1599.

Timer Time base in milliseconds

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.

Dynamic Nonvolatile Flags

Par défaut, tous les indicateurs sont non volatils. Si nécessaire les paramètres *Has Volatile Flags* et *Last Volatile Flag* permettent de définir une plage volatile. (Cet exemple définit des indicateurs volatils avec une plage d'adresses F 0 à F 2999)

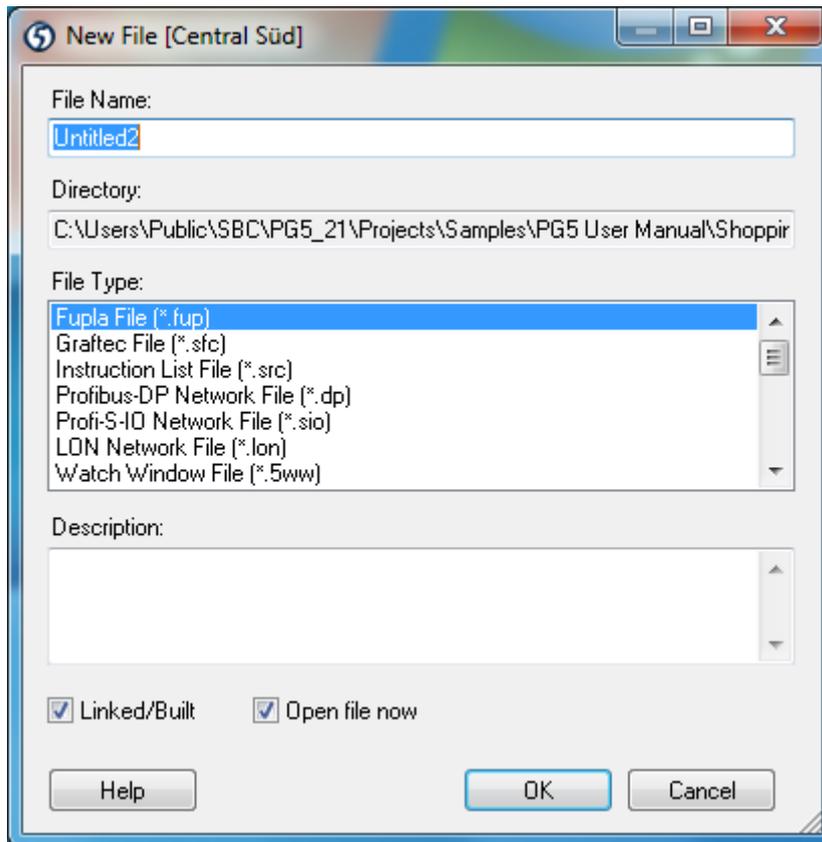
2.2.9 Dossier Programmes



Ce dossier est prévu pour recevoir tous les fichiers qui constituent le programme d'un PCD. La sélection d'un fichier permet de réaliser les actions suivantes avec le menu principal ou le menu de contexte.

Menu **File, New**

Ajoute un nouveau fichier au dossier, définir le nom et le type du fichier.



File Name : nom du fichier.

Directory: chemin jusqu'au fichier.

File Type: définit l'éditeur

Description : texte libre pour une description, un historique, informations de version,...

Linked/Build: option pour lier ou ignorer le fichier au *build*.

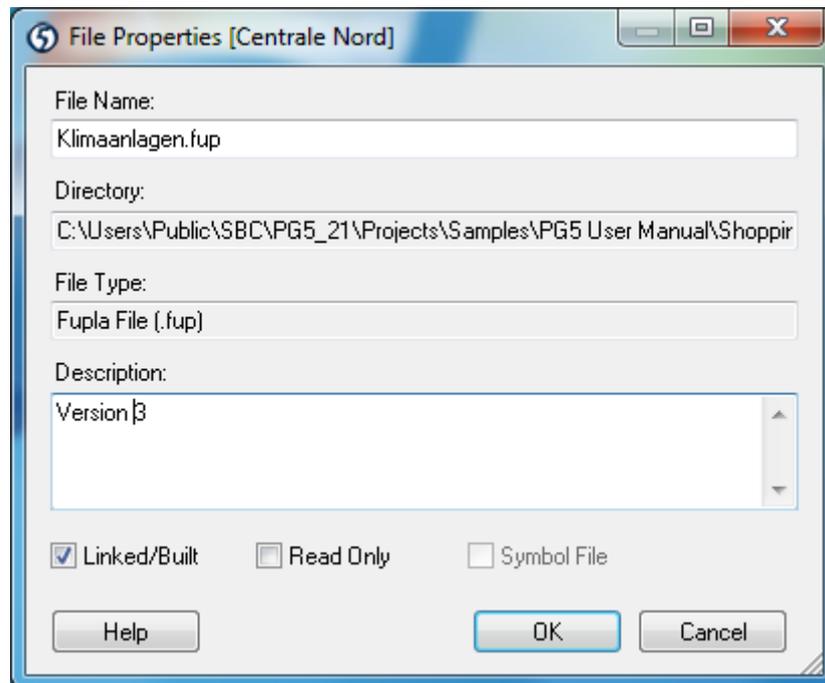
Menu **Add Files**

Importe les fichiers présents dans d'autres projets que celui ouvert.

Des fichiers peuvent être copiés dans le répertoire du *device* ou lié par un chemin, selon l'option sélectionnée dans le dialog *Add Files* : *Copy files into device directory*

Menu **File, Properties**

Modifier les informations relatives au fichier: nom, description, lie le fichier au *build*, protection contre l'écriture. (*Read Only*).



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.

Menu **Edit, Copy, Paste, Delete**

Un fichier présent dans un *device* du projet peut aussi être dupliqué par *Drag&Drop* ou Copié&Collé vers le répertoire d'un *device* voisin. Delete permet de le supprimer le *device*.

2.2.10 Types de fichiers

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

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

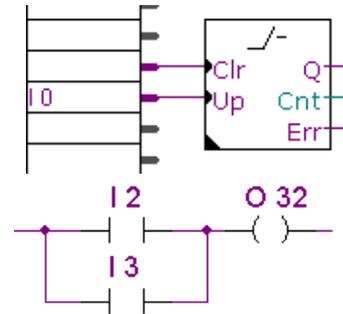
Permet de programmer sous la forme d'un texte avec un jeu de 127 instructions. Convient pour toutes les applications. Le code est rapide et efficace mais demande une certaine expérience de programmation.

```

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

L'éditeur Fupla (*.fup)

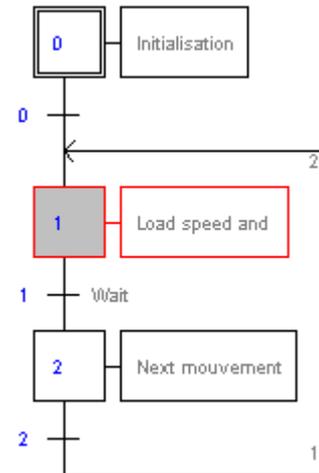
Permet dessiner les programmes sous la forme de plans de fonctions et de schémas à contacts. Ne nécessite aucune expérience de programmation. De nombreuses bibliothèques sont à disposition pour mettre rapidement en service les applications CVC et les réseaux de communication modem, Lon, Belimo, EIB,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



L'éditeur Graftec (*.sfc)

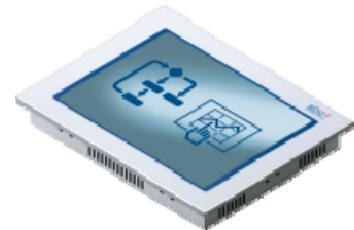
C'est un outil de structuration pour les programmes en liste d'instructions et Fupla. Convient particulièrement bien aux applications séquentielles avec des attentes sur des événements internes ou externes au PCD.

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



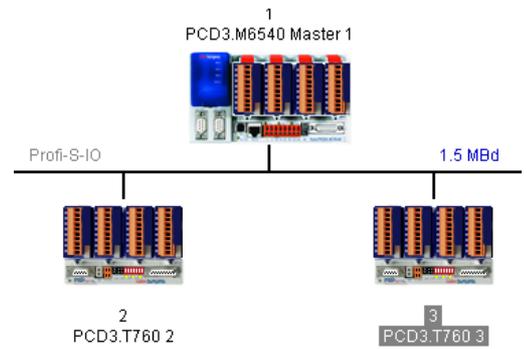
Le Web Editor Version 8 (*.sln)

Editeur de pages web pour surveiller le processus et modifier ses paramètres de fonctionnement. Ces pages sont sauvegardées sur le serveur de l'automate, des web panels (PCD7.D4xx) ou le disque dur de l'ordinateur. Les Web panels ainsi que l'explorateur Internet de l'ordinateur permettent d'afficher ces pages web au travers de tous les réseaux de communication de l'automate.



Configurateur Snet (*.DP, *.LON, *.SRIO)

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

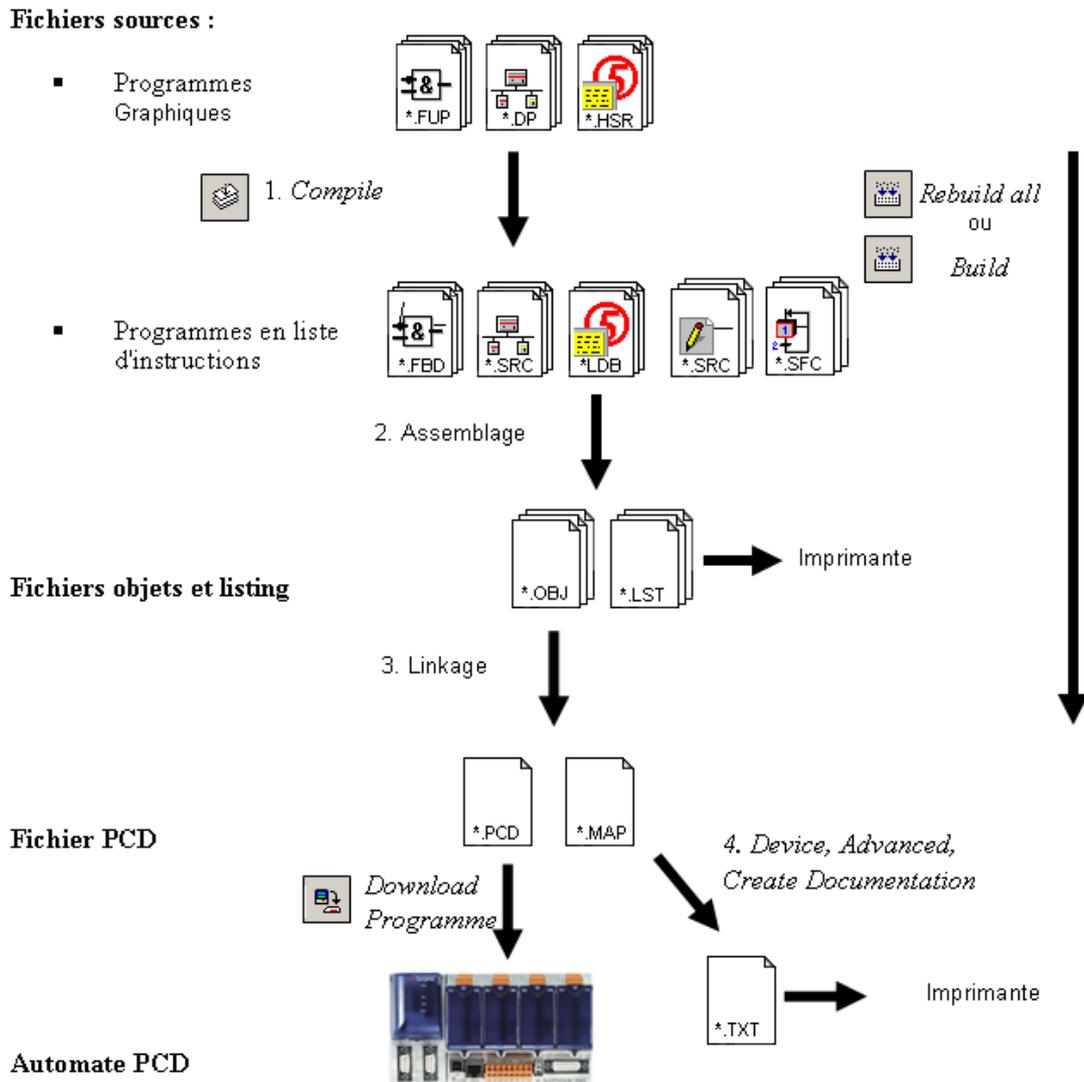
**L'éditeur HMI (*.HMI)**

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



2.3 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:



- Le linkage, chaîne les fichiers (*.OBJ) pour former un seul fichier (*.PCD) destiné à être chargé dans le l'automate.
- Une documentation peut être générée avec le menu *Device, Advance* du gestionnaire de projet, le résultat est disponible dans le dossier *Documentations Files*
-

2.3.1 Construction des fichier modifiés, Rebuild all, Build, Rebuild All Programs



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



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

Le menu *Project*, *Rebuild All Programs ...* réalise l'un après l'autre le *build* de tous les *devices* présents dans le projet. Une fois terminé, la fenêtre des messages indique le nombre de *devices* construit et la liste des éventuels *devices* qui n'auraient pas passé le *build* sans erreur. La sélection du message relatif aux *devices* en erreur affiche le détail du *build*.

2.3.2 Options du *Build*

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

Build	
Ask before saving changed files	True
Stop build on first error	False
Download after successful build	False
Download without confirmation	False
Clear message window on build	False
Create Listing files (.lst)	False
No titles or page breaks	True
Disable \$NOLIST	False
Hide Graftec parameters	False
Do not expand Macros	False
No cross-reference list	True
Create Map file (.map)	False
Create Documentation files (.txt)	False
Lines per page for listing and documentation files	60

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

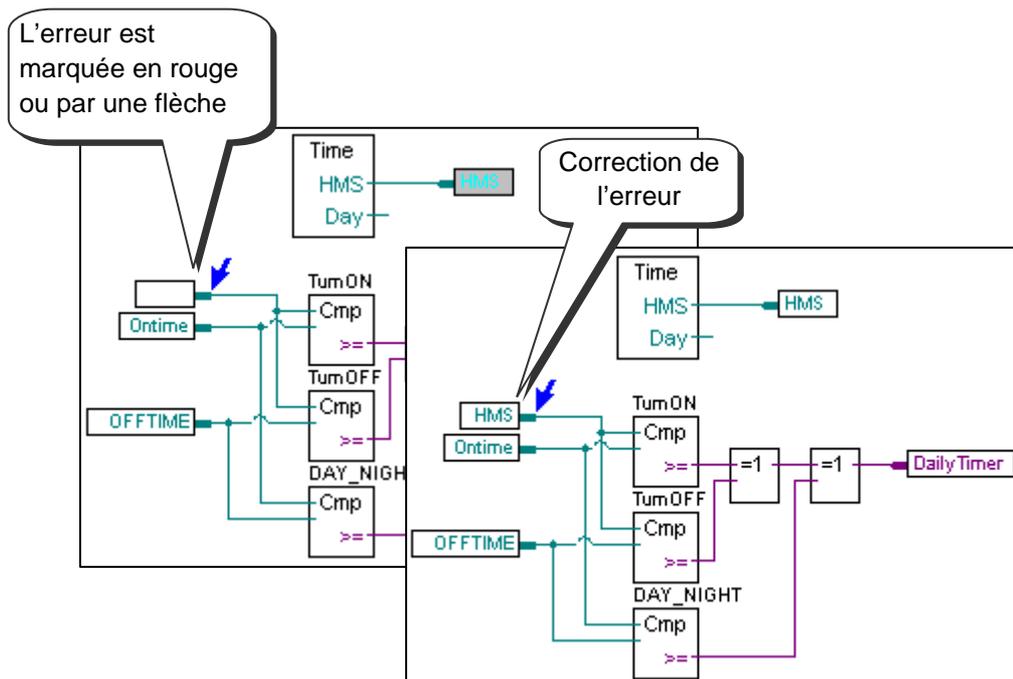
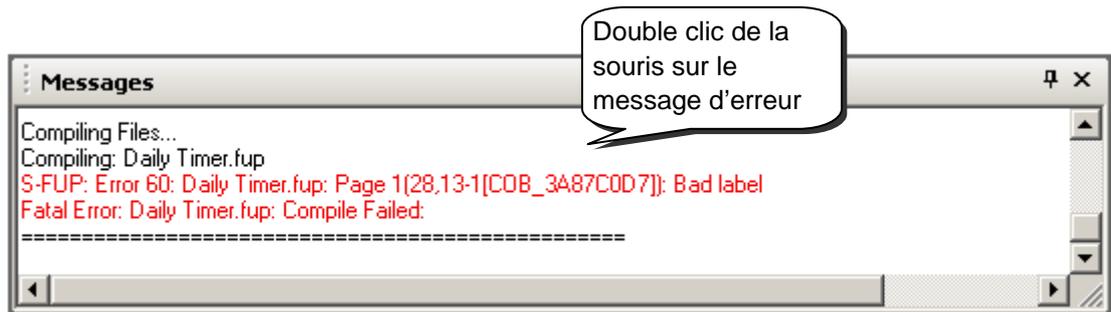
Crée un fichier avec l'espace mémoire nécessaire à l'application.

2.4 Fenêtre des messages

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 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. Alors que la sélection de la ligne du message d'erreur suivie de la sélection de la touche F1 affiche généralement l'aide relative à l'erreur.



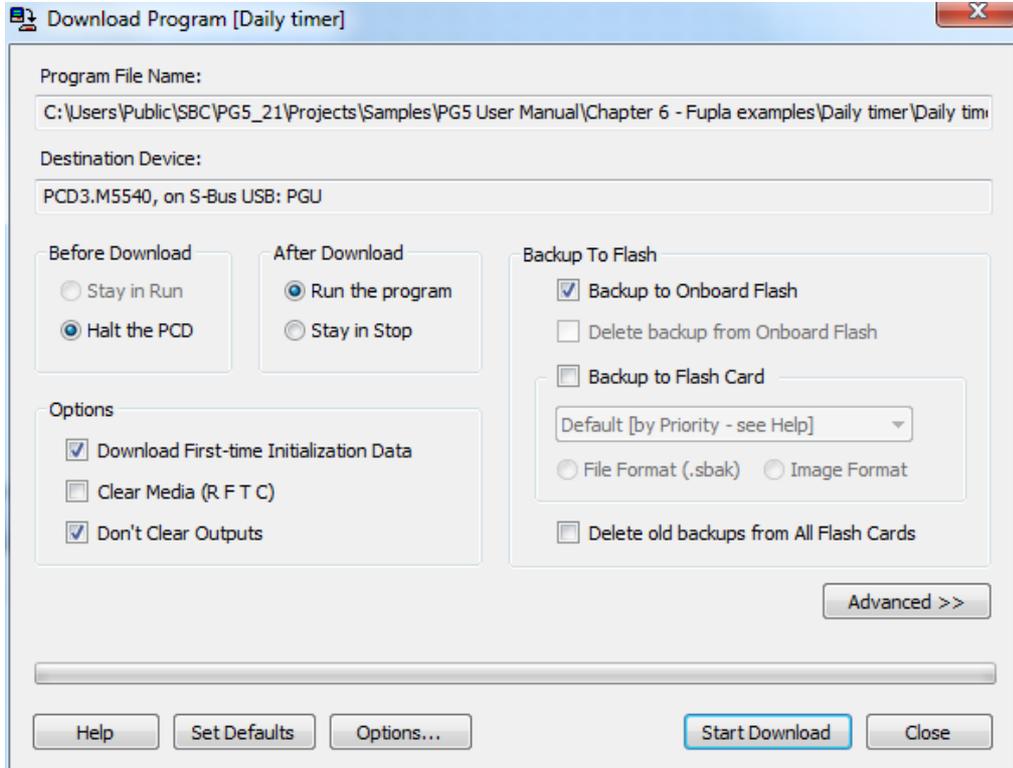
2.5 Charger le programme dans l'automate

2.5.1 Download Program



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 exécutable dans la mémoire RAM de l'automate.



Program File Name

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

All

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

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, *Downloadable files* = ...

First-time Initialisation Data Only

Ne charge que les données décrites ci-dessous.

First-time Initialisation Data

Initialisation certaines données lors du chargement du programme. Ces données sont initialisées comme suit :

`symbol EQU type [address] := initialisation_value`

Untitled1.fup	ROOT		
—◆ symbole0	R	10 := 314	First time initialisation value = 324
—◆ symbole1	R	11	

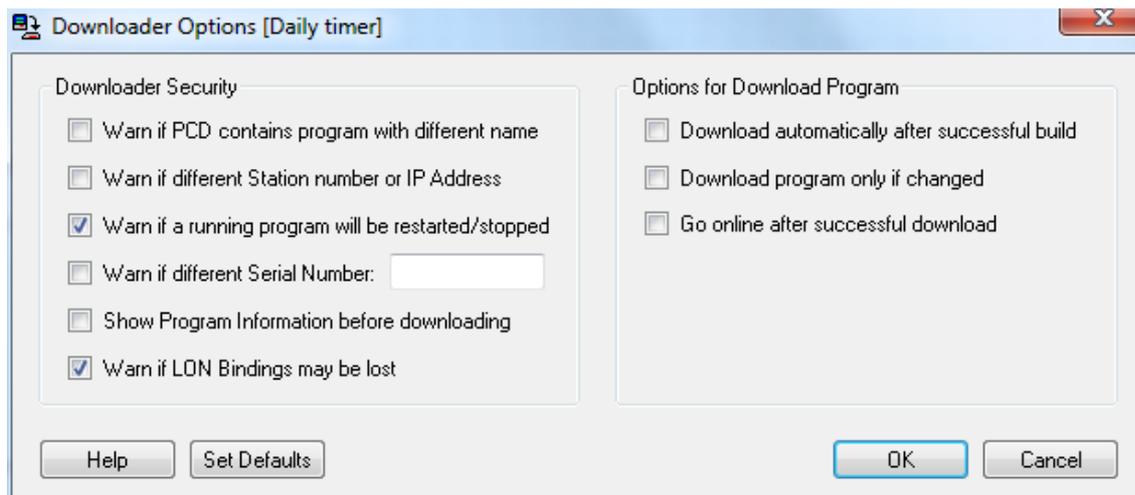
Les données qui ne sont pas initialisées au chargement du programme peuvent être initialisées à chaque démarrage à froid par un bloc XOB16.

Copy User Program to Flash

Une fois le chargement du programme terminé, effectue une copie du programme exécutable présent en mémoire RAM vers la mémoire Flash.

2.5.2 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. A utiliser que si un problème avec la mémoire de l'automate est suspecté. Problème très peu fréquent !

Run the program after successful download

Met automatiquement le PCD 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 PCD en ligne après le chargement.

Backup user program to Flash after download

Copie automatiquement le programme sur la mémoire backup Flash¹.

1) PCD2.M170, PCD2.M480, PCD2.M5, PCD4.M170 et PCD3

Si cette option n'est pas activée, il est toujours possible de la 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 PCD.

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 le stopper.

Do not clear Outputs on download or restart

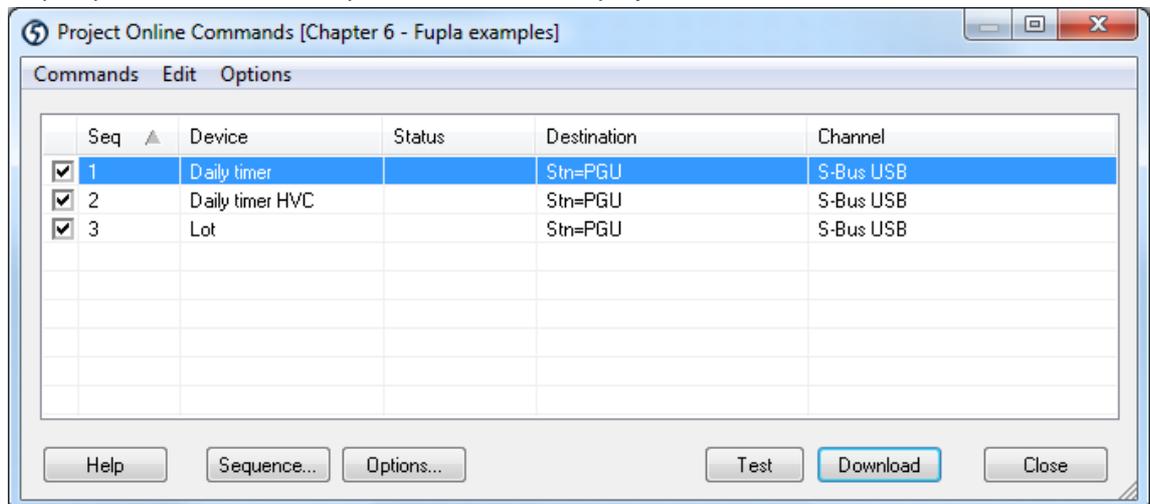
Cette option peut être utile avec les applications CVC. Cela évite de stopper une ventilation, d'éteindre la lumière durant le chargement d'un programme. Ne devrait pas être sélectionnés avec les autres applications.

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.

2.6 Commandes pour plusieurs devices

Si l'outil de programmation PG5 et tous les automates du projet sont raccordés sur un réseau de communication, le menu *Project, Online Commands ...* permet de réaliser quelques commandes sur plusieurs *devices* du projet.



Une table affiche la liste des *devices* présents dans le projet. Elle permet de sélectionner les *devices* concernés et d'activer une commande avec le menu.

Options, Device Sequence

Par défaut, les commandes multiples sont traitées l'une après l'autre selon la position des *devices* dans la fenêtre *Projects*. Cette option permet de modifier l'ordre des *devices*.

Commands, test

Vérifie le bon fonctionnement de la communication avec les *devices*. Si le test est négatif, vérifier les *Online Settings* du *device* concerné.

Commands, download programs.

Charge le programme dans les *devices* sélectionnés, les options de chargement sont configurables avec le menu *Options*. Attention, une des options peut être importante, elle permet de définir à quel instant les *devices* sont mis en *Stop* puis en *Run*.

Set Clock

Copie la date et l'heure de l'ordinateur dans chacun des *devices* sélectionné.

Run/Stop

Force chacun des *devices* sélectionné dans le mode choisi.

2.7 Self downloader

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

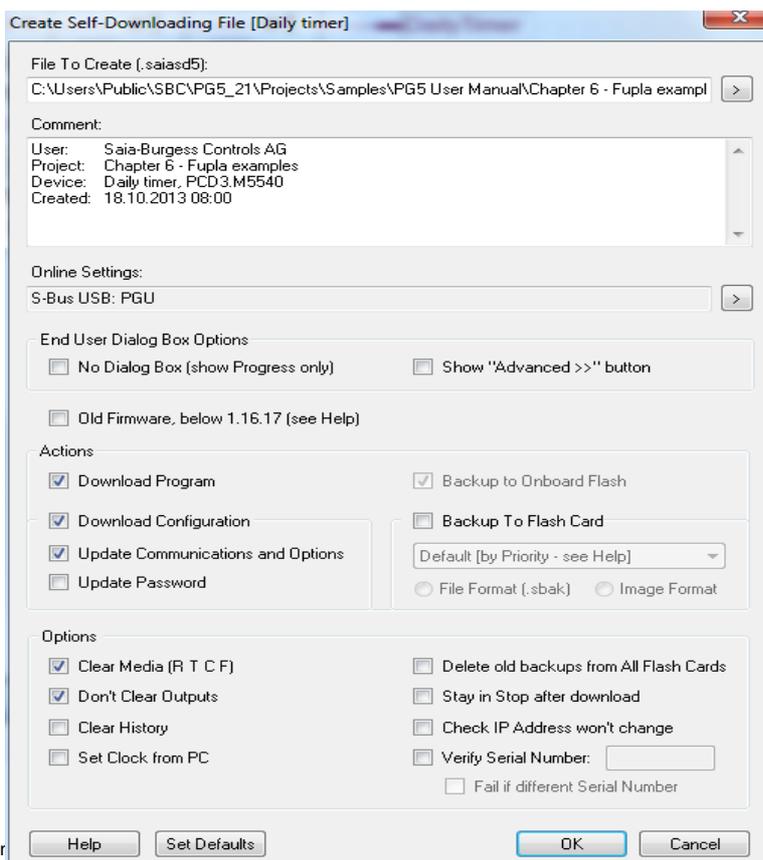
Cet outil prépare un fichier **.saia5d5* contenant toutes les informations nécessaires à la mise à jour des programmes et configurations d'un automate.

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

L'ouverture d'un fichier **.saia5d5* affiche le dialogue de chargement des données avec quelques paramètres et options correspondant à celles prédéfinies dans le projet. La personne présente sur le site d'exploitation peut éventuellement modifier ces options ou les garder tel quel puis les charger dans l'automate.

Ainsi, le chargement du programme, de la configuration ne nécessite pas de connaissances particulières de l'utilisation de PG5. Cet outil autorise le chargement des programmes et la configuration sans installer PG5 ni licence d'exploitation mais nécessite tout de même l'installation du programme *Online Tools* sur le site d'exploitation.

2.7.1 Préparer un fichier *.5d5



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

Le menu *Device, Advanced, 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, Configuration, Download* et *Online, Download Program ...*
Seul quelques nouveaux paramètres ont été ajoutés:

Create Files (*.sd5)

Défini le chemin et le nom du fichier .sd5

Show "Advanced >>" button

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

No Dialog box (Progress only)

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

Verify Serial Number

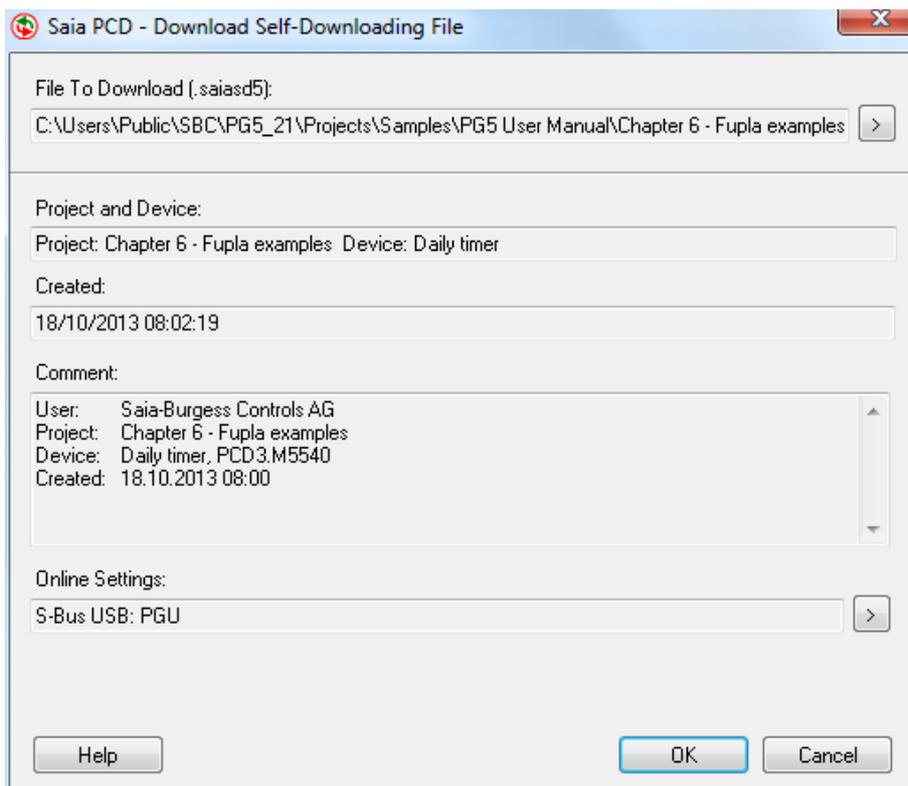
Le *self-downloading* vérifie si le numéro de série du PCD correspond à celui inscrit dans le champ *Serial Number*.

Le numéro de série est unique à chaque automate, 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 PCD. Il peut être lu directement sur le PCD avec le configurateur en ligne, menu *Online, Informations*.

2.7.2 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.

2.8 Mémoire Flash Backup



No d'article :	PCD7.R5xx	PCD3.R5xx	PCD.R600
Slots :	M1/M2	I/O 0,...,3	I/O slots 0,...,3
Systèmes :	PCD1.Mxxx0 PCD2.M5, PCD3	PCD3	PCD3

Les systèmes PCD offrent une petite mémoire Flash interne à l'automate mais aussi externe et amovible et de plus grande capacité sur des emplacements réservés à cet effet ou sur les slots I/O. Contrairement à la mémoire RAM de l'automate, les mémoires Flash ont l'avantage de ne pas être altérées lors d'une coupure d'alimentation. Nous proposons de les utiliser pour sauvegarder une copie du programme exécutable, une copie des fichiers sources qui constituent le projet ou pour sauvegarder des données dans un fichier accessible en lecture et écriture par le programme de l'automate.

2.8.1 Sauvegarde du programme exécutable

Le programme de l'automate est exécuté à partir d'une mémoire RAM, en cas de coupure d'alimentation associée à une batterie défectueuse, le contenu de la mémoire peut être altéré et l'automate ne fonctionne plus lorsque l'alimentation est à nouveau présente.

La mémoire de sauvegarde permet de conserver une copie du programme exécutable en RAM (*Code/Text/Extension*) sur une mémoire Flash dont le contenu n'est pas altéré lors de coupures de courant.

Le menu *Online, Flash Memory, Copy Program to Flash...* copie le programme exécutable en mémoire RAM vers la carte Flash et inversement avec la commande *Copy Program From Flash....*

Cette copie est supportée automatiquement en sélectionnant une option avec le menu *Tools, Options, Backup user program to Flash after download.*

Si le programme d'application en RAM (*Code/Text/Extension*) est altéré, le démarrage à froid du PCD copie automatiquement le programme exécutable disponible en mémoire Flash sur la mémoire RAM.

Attention, malgré la sauvegarde en mémoire Flash, les fichiers sources du projet doivent toujours être conservés car seul le fichier exécutable est chargé dans l'automate. Les fichiers sources restent sur le disque dur de l'ordinateur.

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

2.8.2 Sauvegarde du projet source

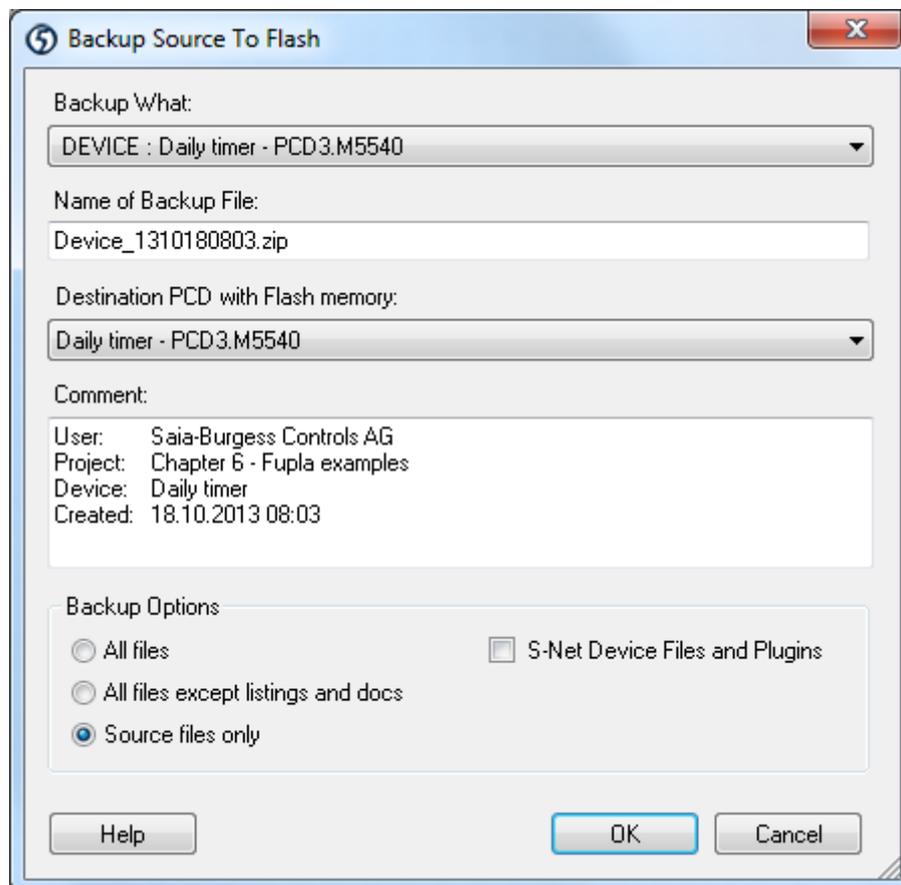
Les fichiers sources du programme de l'automate sont sauvegardés sur le disque dur de l'ordinateur. Seul le code exécutable créé par le *build* est chargé dans la mémoire RAM de l'automate. Sans les fichiers sources il n'est plus possible de modifier les programmes pour en assurer l'entretien. Il est donc important de bien les conserver et de toujours tenir la dernière version à disposition des techniciens d'entretien.

Nous offrons deux services différents pour assurer la sauvegarde des fichiers sources.

- Le menu *Projet, Backup...* sauvegarde sur le disque dur de l'ordinateur toute la structure de répertoires et les fichiers qui constituent le projet dans un fichier au format .zip. Un seul fichier est bien plus facile à transmettre qu'une structure de répertoires et de fichiers. La restauration du projet sur le disque dur à partir du fichier .zip est supportée avec le menu *Projet, Restore...*
- Le menu *Online, Flash Memory, Backup Source to Flash...* sauvegarde sur la mémoire flash de l'automate toute la structure de répertoires et les fichiers qui constituent le projet dans un fichier au format .zip. Le chargement de ce fichier est supporté par FTP au travers d'un canal de communication Ethernet uniquement. La restauration du projet sur le disque dur à partir du fichier .zip est réalisée avec le menu *Online, Flash Memory, Restore Source from Flash...*

Backup Source du projet

La commande *Online, Flash Memory, Backup Source to Flash...* commence par compresser toute la structure de répertoires et fichiers qui constitue le projet dans un fichier au format .zip



Backup Project or single Device :

Cette option sauvegarde l'ensemble du projet ou un des *devices*. Par défaut c'est le *device* actif qui est proposé.

Name of Backup file : nom du fichier de sauvegarde .zip

Destination PCD with Flash Memory :

Comment : commentaires libres

Texte libre, par défaut un texte est toujours présent avec le nom de son auteur, du projet et du *device* et la date. Il est toujours possible d'ajouter un numéro de version ou d'autres informations.

Backup what : définit le contenu de la sauvegarde, il n'est pas nécessaire de conserver tous les fichiers. Seul les fichiers sources sont importants.

FTP downloader

En suite, le *FTP downloader* est appelé, le dialogue suivant est affiché pour définir les informations relatives au chargement du fichier de sauvegarde sur la mémoire Flash.

Complete FTP connection parameters and target directory then press the Backup but...

FTP Connection	
User Name	root
Password	rootpasswd
Save Password	True
Host Name or IP Address	0.0.0.0
Local Computer	
PC File Name	C:\Users\hu2doga0\AppData\Local\Temp\Sa...
Remote Device	
Device Directory Name	

User Name
User name to login the FTP server
The default name for the Device is "root"

Help Backup Cancel

Root :

Nom d'identification de l'utilisateur pour accéder au serveur FTP. Si aucun utilisateur n'est défini, utiliser l'utilisateur par défaut: *root*.

Password:

L'accès au serveur FTP de l'automate est protégé par un mot de passe. Si aucun mot de passe n'est défini, utiliser le mot de passe par défaut: *rootpasswd*. Attention les mots de passe sont différents pour l'accès au serveur FTP et au programme de l'automate. Le mot de passe pour accéder au serveur FTP est défini à

partir d'un fichier de configuration présent sur la mémoire Flash. Le nom du fichier est *FTPConfig.txt*. Le mot de passe pour accéder au programme de l'automate est défini dans les *Device Configurator*, voir sous *Device Slot, Password*.

IP address:

Adresse IP de l'automate sur le réseau Ethernet. Actuellement, seul le canal Ethernet des *Online Settings* supporte la communication avec le serveur FTP.

Remember password:

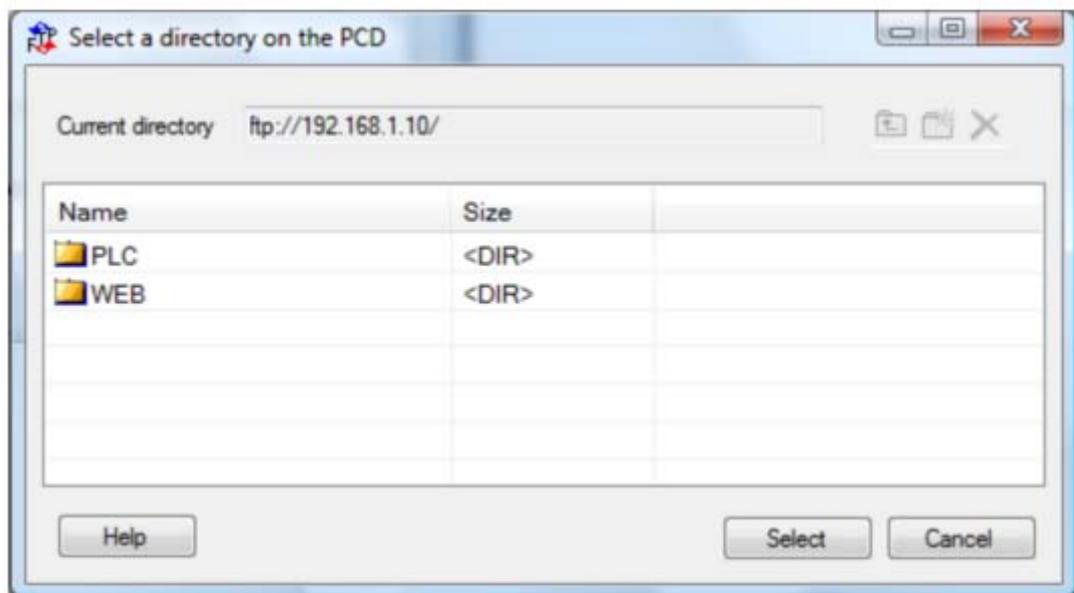
Si sélectionné avec *True*, ce dialogue conserve le mot de passe pour le prochain usage de cette fonction.

PCD file name:

Chemin d'accès et nom du fichier à charger sur la mémoire Flash.

PCD directory name:

A la sélection de cette ligne, le dialogue affiche un bouton sur sa droite. La sélection de ce bouton permet d'établir la communication avec le FTP serveur pour sélectionner la mémoire Flash et le répertoire de destination.



Cette commande peut prendre un peu de temps pour charger les informations ci-dessus et les afficher. Si plusieurs mémoires Flash sont accessibles, les différentes mémoires sont représentées avec leurs structures de répertoires.

Définir la mémoire et le chemin jusqu'au répertoire de mémorisation du fichier. Si nécessaire créer ou supprimer des répertoires avec les boutons réservés à cet effet.

M1_FLASH et M2_FLASH :

Mémoires Flash montées sur les slots M1 et M2 de l'automate.

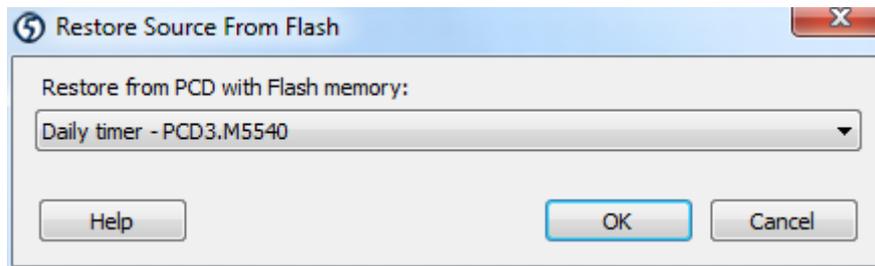
SL0FLASH, SL1FLASH, ..., SL1FLASH :

Mémoires Flash montées sur les slots d'entrées/sorties 0,1,...3 de l'automate.

Attention, cette fonctionnalité n'est supportée que par les automates récents équipés d'un mémoire flash avec un *file system*.

Restauration du projet

La commande *Online, Flash Memory, Restore File from Flash...* commence par charger la sauvegarde du projet avec le format .zip présent sur la mémoire Flash vers le disque dur de l'ordinateur puis le décompresse pour le restaurer.



Un dialogue permet de sélectionner le répertoire de l'un des *devices* du projet ouvert. Le *device* actif est proposé par défaut. C'est dans le sous répertoire BACKUP de ce *device* qu'est chargé le fichier .zip puis décompressé.

Le FTP uploader affiche un dialogue avec les mêmes paramètres que décrit précédemment pour l'opération de *download*.

La sélection du bouton droit correspondant du paramètre *Device File name* permet d'établir la communication avec le FTP serveur et de sélectionner la mémoire Flash, le répertoire source du fichier à restaurer.

Il ne reste alors plus qu'à confirmer le chargement et la restauration pour que ces tâches soient effectuées.

2.8.3 Sauvegarde de données dans un fichier.

La librairie Fboxes *File System* supporte les accès standards aux fichiers de données de la mémoire Flash. Ces Fichiers sont accessibles en lecture et écriture et peuvent être téléchargés au travers du serveur FTP de l'automate.



Pour de plus amples informations concernant les nombreux types de mémoires Flash disponibles, la configuration du serveur FTP ainsi que les possibilités de créer un fichier de données sur la mémoire Flash à l'aide du programme Fupla ou IL. Veuillez vous référer à la documentation détaillée disponible avec les manuels:

***SBC FTP Server and SBC Flash File System
Hardware manual for the PCD3
Hardware manual for the PCD2.M5
Hardware manual for the PCD1.Mxxx0***

2.9 Les fenêtres *View*

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

2.9.1 Structure des blocs d'organisation

Le langage de programmation PCD est structuré avec différents blocs d'organisation dans lesquels l'utilisateur insert 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).

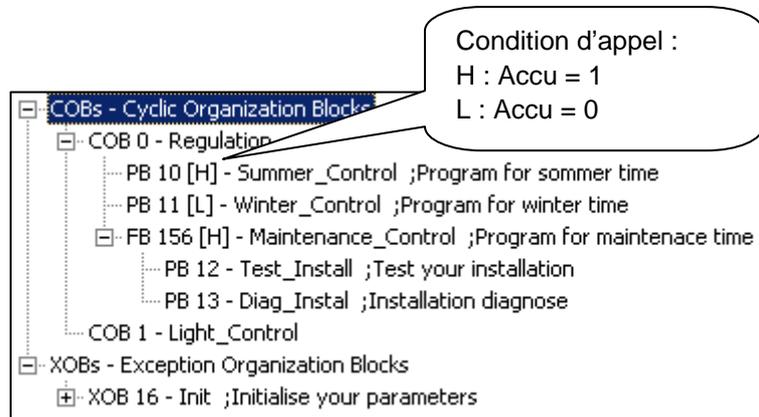


Block Call Structure

Après la construction du programme, le bouton *Block Call Structure* ou le menu *Block Call 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.



2.9.2 Liste des symboles

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



- *All Publics Symbols*, affiche les symboles partagés par tous les fichiers du *device* actif. Cette liste ne peut pas être éditée, pour éditer les symboles globaux, il faut ouvrir le fichier qui les définit.
- *Data List*, affiche tous les symboles du *device* actif. Les symboles globaux, locaux et les adresses absolues sont représentés. Les symboles qui ne sont pas utilisés ne sont pas affichés dans cette fenêtre.



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

2.9.3 Référence croisée

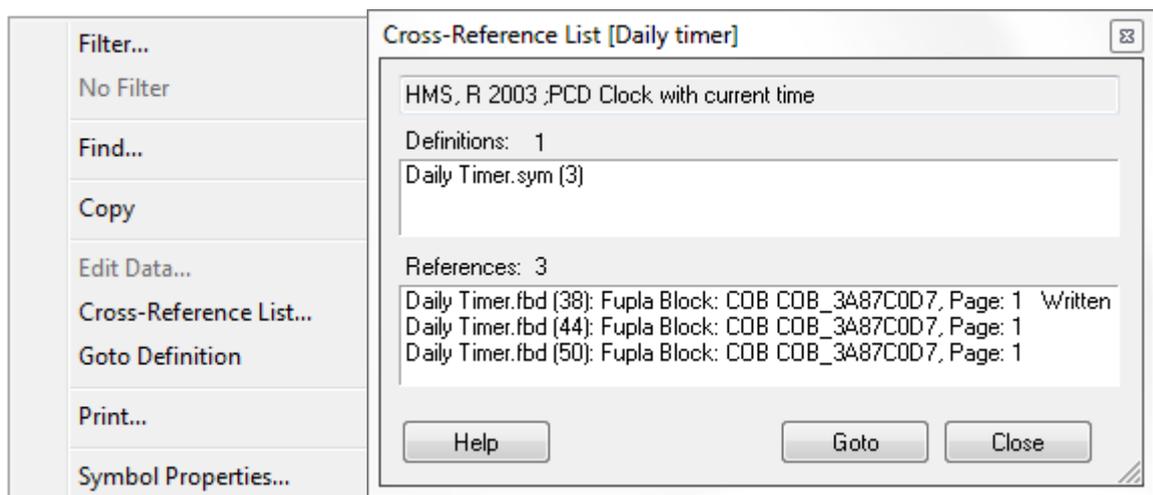
Les listes *All Publics Symbols* et *Data List* offrent la possibilité de sélectionner un symbole et de retourner une référence croisée: liste des pages du programme faisant usage du symbole sélectionné.

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*.



2.10 L'outil *Online Configurator*



Online Settings	Paramètres de communication
PCD type	Numéro d'article du PCD
Version	Version du firmware PCD
Program Name	Nom du programme, device dans le projet
Date	Date de l'horloge de l'automate (Si pas d'horloge : 1/1/92)
Time	Heure de l'horloge de l'automate
Day	Jour de la semaine : 1 = lundi, ... 7 = dimanche
Week	N° de la semaine de l'année (1, ..., 52)
Status	Mode d'exploitation : Run, Stop, Halt, Conditional Run

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 l'automate et le *Online Configurator*.

Veillez alors vérifier:

- Si l'ordinateur est correctement raccordé avec le câble PCD8.K111/USB ou le réseau.
- Si les paramètres de communications sont correctement sélectionnés avec le bouton *Online Settings...*

2.10.1 Configuration

Configuration...

Affiche les settings présent dans la *Configuration* du gestionnaire de projet. Voir description du Device Configurateur.

2.10.2 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 PCD est allumée.

Descriptif de l'erreur	Compteur d'erreurs	Ligne de programme	Date et heure
>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
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

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 (Coupure d'alimentation) n'apparaît dans *l'history* que si le XOB 0 est programmé.

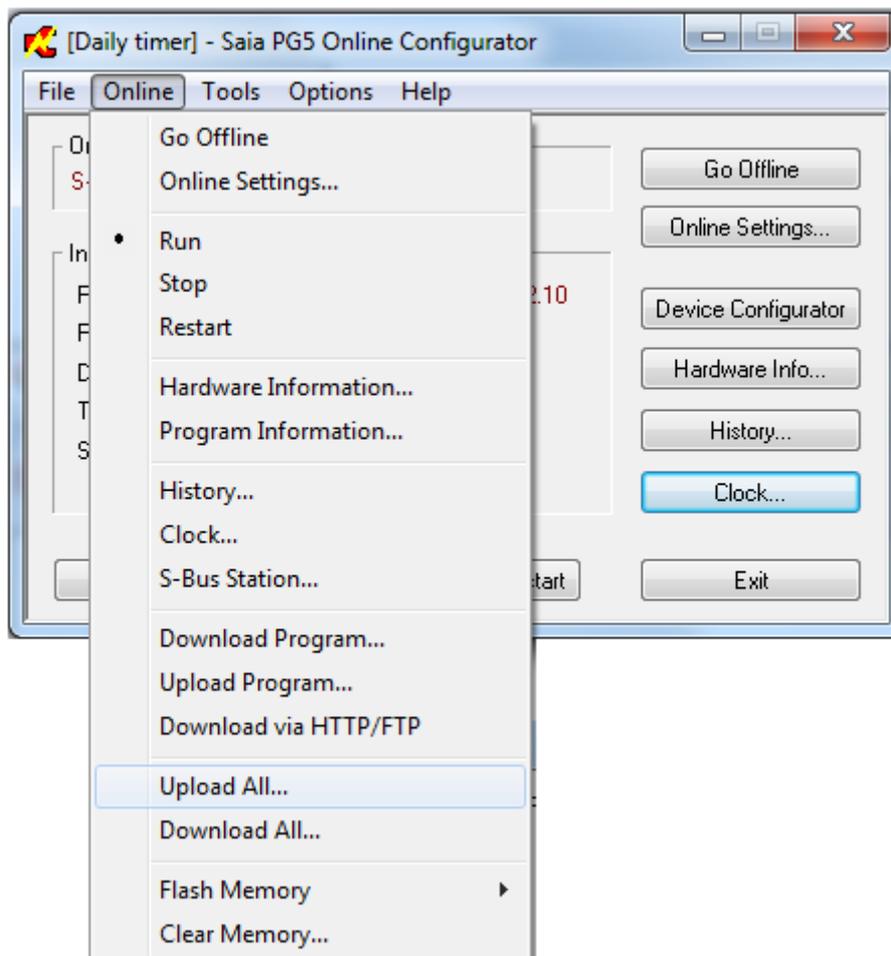
2.10.3 Ajuster l'horloge de l'automate

Clock...

Les automates disposent presque tous d'une horloge délivrant la date et l'heure. Le bouton *Clock* affiche le dialogue pour ajuster la date et heure du jour courant.

La sélection du bouton *Copy to PCD >>>* copie immédiatement l'heure de l'ordinateur sur l'automate. Il est aussi possible d'éditer directement la date et l'heure de l'automate dans les champs *SAIA PCD Clock*.

2.10.4 Sauvegarde du programme en mémoire RAM.



Voici une commande intéressante, elle permet de sauvegarder/restaurer le programme, la configuration ainsi que les valeurs des registres, indicateurs, compteurs, DB, textes, ... présents dans la mémoire RAM de l'automate.

C'est très pratique pour copier le programme vers un autre automate, soit pour dupliquer l'installation, soit pour échanger le PCD ou tout simplement pour être en mesure de restaurer l'état de la machine avant votre intervention.

Tools, Upload All ...

Sauvegarde l'intégralité des données en mémoire RAM de l'automate dans un fichier. au format .im5. Ce fichier contient les informations suivantes : (Programme, configuration, valeurs des registres, indicateurs, compteurs, DB, textes, ...)

Tools, Download All ...

Restore le fichier .im5 sur la mémoire RAM de l'automate.

2.10.5 Create Diagnostic file

C'est aussi une commande fort utile de ce *Online configurateur*. Elle permet de créer un fichier avec toutes les données nécessaires pour faire une demande de renseignement au près de notre support technique.

Ce fichier contient des informations tel que le type d'automate, sa version de firmware,...

Pour créer ce diagnostic, utiliser le menu **Tools, Create Diagnostic File ...**

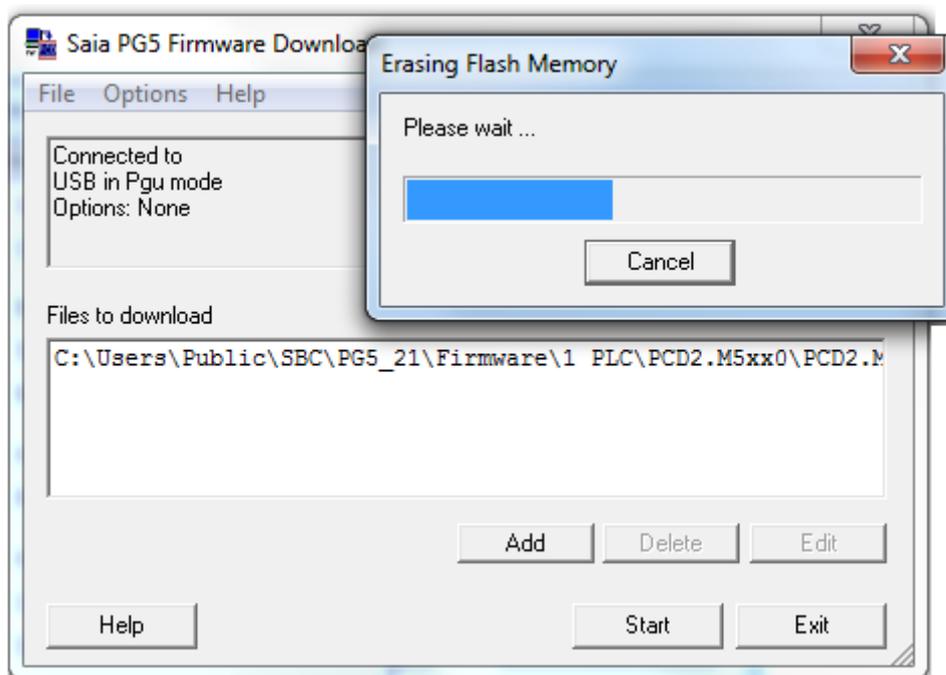
2.11 Mise à jour du firmware. (*Firmware Downloader*)

Pour que les automates plus anciens puissent aussi faire usage des fonctionnalités les plus récentes de l'outil de programmation, il est parfois nécessaire de mettre à jour le programme *firmware* à bord de l'automate.

Cette mise à jour est rapide et facile à mettre en oeuvre pour tous les automates dont le *firmware* est sauvegardé sur une mémoire Flash.

La version du *firmware* en service sur l'automate est affichée avec la fenêtre *Online Configurator*.

Nous effectuons le chargement du *firmware* avec un utilitaire disponible sous le menu *Tools, Firmware Downloader command, Tool, PCD Firmware Downloader* du gestionnaire de projet.



Le bouton **ADD** permet d'ajouter un nouveau fichier firmware (*.blk) dans la liste: *Files to download*. Les fichiers *firmwares* les plus récents sont installés avec le PG5 et sont automatiquement proposés.

Attention pour certaines applications bien particulières, cette fenêtre peut afficher plusieurs fichiers. Veuillez prendre garde à ce que seul le fichier *firmware* correspondant à l'automate raccordé sur l'ordinateur soit affiché.

Le menu *File, Settings* permet d'ajuster les paramètres de communication avec le mode *S-bus USB* et l'option *PGU*

Raccorder le câble USB l'automate.

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* de l'automate se mettent à clignoter. L'automate réorganise certaines informations en mémoire. Veuillez attendre encore une minute environ avant de le mettre hors tension ou de continuer le travail.

Table des matières

TABLE DES MATIÈRES	1
3 DEVICE CONFIGURATEUR.....	2
3.1 Lecture des paramètres de l'automate vers le configurateur	2
3.2 La vue principale du configurateur	3
3.3 Chargement des paramètres du configurateur vers l'automate.....	4
3.4 <i>Device properties</i>	5
3.4.1 <i>Memory</i>	5
3.4.2 <i>Password</i>	6
3.4.3 <i>S-Bus</i>	7
3.4.4 <i>Power Supply</i>	7
3.5 Serial S-Bus communication properties	7
3.5.1 <i>Full protocol (PGU) Serial-S-Bus</i> (esclave PGU pour ligne série)	7
3.5.2 <i>Public Line S-Bus Modem</i> (esclave PGU pour un ligne modem)	8
3.5.3 <i>Serial S-Bus Master Gateway</i> . (gateway maître).....	8
3.5.4 <i>S-Bus Mode and Timings</i>	8
3.6 Profi S-Bus communication properties	10
3.6.1 <i>Profi-S-Bus</i> (esclave).....	10
3.6.2 <i>Profi S-Bus Master Gateway</i> . (gateway maître)	11
3.6.3 <i>Bus Parameters: user defined</i>	12
3.7 Ether-S-Bus communication properties	13
3.7.1 <i>Ether-S-Bus</i> (esclave).....	13
3.7.2 <i>Profi S-Bus Master Gateway</i> . (gateway maître)	14
3.8 On board slots properties, configuration de la mémoire image	15
3.8.1 <i>Device properties</i> , configurations nécessaires.....	15
3.8.2 <i>Onboard slots</i> , configurations des modules E/S.....	15
3.8.3 Propriétés des E/S binaires.	16
3.8.4 Propriétés des E/S analogiques.....	17
3.9 Édition des étiquettes pour les modules I/O	18
3.10 Extension du device configurateur avec des nouveaux <i>devices</i> et modules E/S	19

3 Device configurateur



Le *Saia PG5 Device Confugurator* permet de configurer les paramètres matériel de l'automate: le type de *Device*, la mémoire, les canaux de communications, les modules d'entrée/sorties mais vérifie aussi la consommation des modules d'entrées/sortie sur l'alimentation interne du PCD et imprime les étiquettes pour coller sur les modules E/S.

Pour mettre en service un automate PCD il est nécessaire de définir au moins le type de PCD et sa configuration mémoire. Les autres configurations pourront être ensuite complétées en fonctions des services nécessaires tels que des réseaux de communication ou la gestion des E/S.

Pour les adeptes des versions de PG5 1.4 et plus anciennes, le *Device Configurator* est un programme entièrement nouveau, il présente les paramètres de configuration avec une interface homme machine nouvelle, organisée de manière très différente. Bien que des possibilités de configurations nouvelles soient disponibles. Les automates ainsi que leurs paramètres de configurations restent toujours les mêmes que précédemment.

3.1 Lecture des paramètres de l'automate vers le configurateur



La méthode la plus simple de réaliser la configuration consiste à relier l'ordinateur au PCD avec un câble USB et de lire la configuration déjà présente dans la mémoire du PCD avec le menu *Online, Upload Configuration...* ou le bouton correspondant de la *tool bar*. Si la mémoire ne devait contenir aucune configuration, le firmware PCD se charge de retourner les informations qui conviennent. Puis vérifier les configurations correspondant à votre application.

3.2 La vue principale du configurateur

Device		
Type	Description	
PCD3.M5540	CPU with 256/512/1024 KBytes RAM, 4 I/O slots (expandable), USB, Profi-S-Net, RS-232.	

Memory Slots		
Slot	Type	Description
M1		
M2		

Onboard Communications	
Type	Description
RS-485/S-Net	RS-485 port for Profi-S-Bus or general-purpose communications (D-Sub #2).
USB	Universal Serial Bus port, PGU or general-purpose.
RS-232/PGU	RS-232, PGU or general-purpose serial port (D-Sub #1).
RS-485	RS-485 port for general-purpose communications (Terminal block).
Ethernet	Ethernet port. IP Settings, DHCP.

Ethernet Protocols	
Section	Description
IP Transfer Protocols	FTP, HTTP Direct Protocols, ODM.
IP Protocols	DNS, SNTP, SNMP protocols.
HTTP Portal	HTTP Portal Communication For PCD Over Private Network.

Onboard I/O Slots		
Slot	Type	Description
Slot 0		
Slot 1		
Slot 2		
Slot 3		
+		

Indique le type de *Device* courant. S'il est sélectionné, affiche les propriétés de l'automate tel la taille mémoire RAM/EPROM/Flash interne, le mot de passe, l'usage du S-Bus, de la gestion des entrées/sorties, des options et de la consommation de courant sur le bus des E/S.

Pour modifier le type du *device*, placer la souris sur le *Device* courant et sélectionner le menu de context *Change Device Type ...*. Le menu de context *Properties* permet d'afficher la fenêtre des propriétés relatives.

Memory slots

Slots mémoires disponibles pour recevoir des extensions mémoires Flash correspondant au type de *Device* sélectionné. Pour configurer les slots mémoires, sélectionner et glisser les modules mémoires correspondant depuis la fenêtre *E/S Selector* vers leurs slots. Pour ouvrir, la fenêtre *E/S Selector* sélectionner le menu *View, Selector Window*.

Onboard Communication slots

Les slots de communication interne représentent les interfaces de communications disponibles. Selon les types de PCD, le nombre des interfaces de communication

sont différentes, elles ne sont configurées que si elles sont utilisées par le programme du PCD.

Certains slots sont prédéfinis, d'autres librement configurables et nécessitent de définir les modules de communication correspondant en les sélectionnant dans la fenêtre *E/S Selector* et de les glisser vers un slot.

La sélection d'un slot de communication affiche les paramètres de l'interface de communication dans la fenêtre des propriétés et permet de les définir.

Onboard slots

Représente les slots d'entrées/sorties disponibles de l'automate. Les modules E/S présents sont configurés en les sélectionnant dans la fenêtre *E/S Selector* pour les glisser vers un slot E/S. Comme pour les autres slots, la sélection de l'un des slots E/S affiche les paramètres de configuration relatifs.

Expansion slots

Les slots d'extension désignés avec un '+' peuvent recevoir un module d'extension de bus pour ajouter des modules E/S supplémentaires. La configuration d'un module d'extension se réalise en sélectionnant le module correspondant dans la fenêtre *E/S Selector* pour le glisser vers le slot d'extension noté avec '+'.

3.3 Chargement des paramètres du configurateur vers l'automate

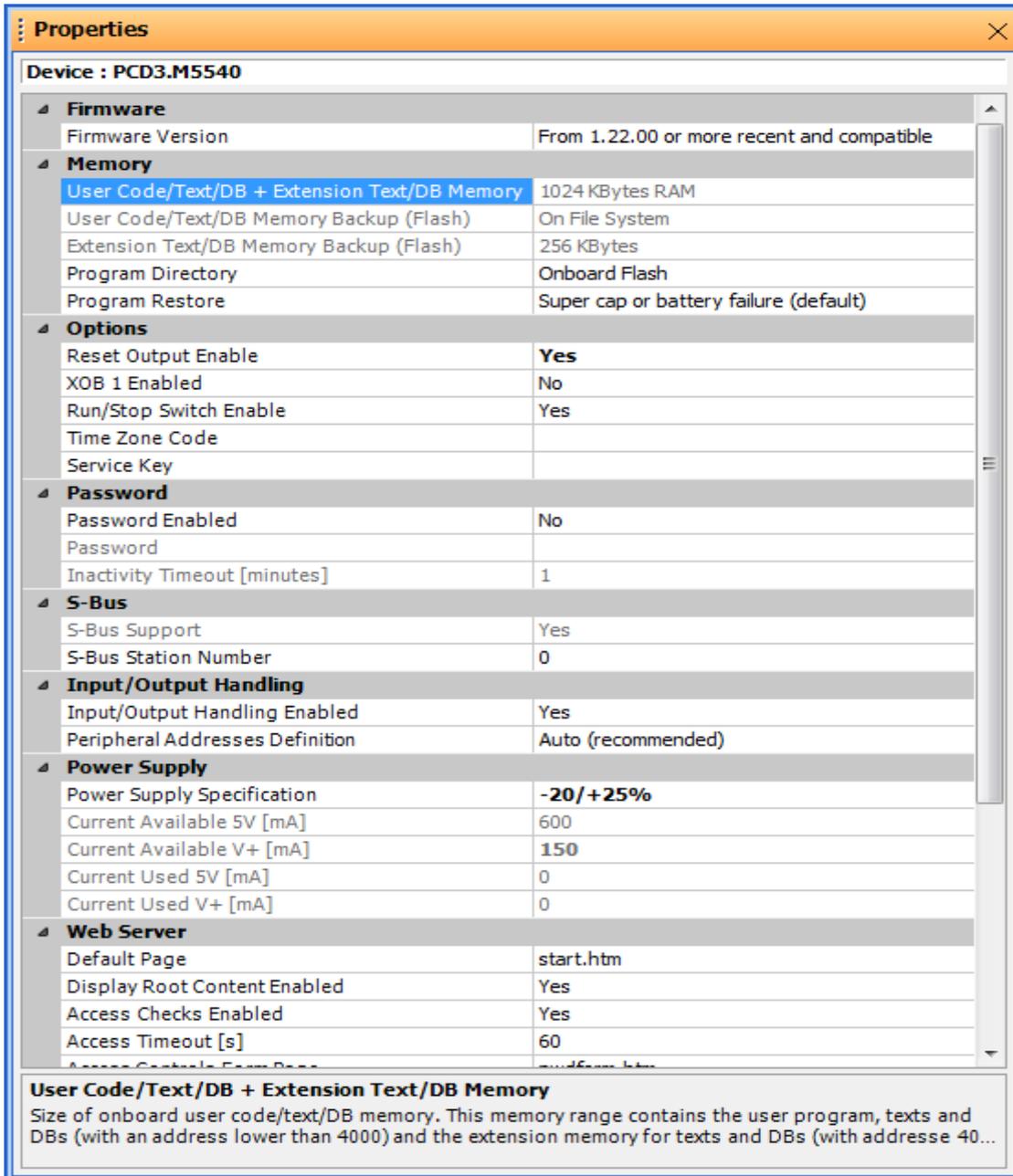
La configuration matérielle de l'automate est nécessaire lors de sa première utilisation mais aussi lors de toutes modifications telles que l'extension de la mémoire, le montage d'un module de communication, ...



Ces paramètres ne sont pas seulement configurés dans le Device Configurator mais aussi chargés dans la mémoire de l'automate avec le menu *Download Configuration* ou le bouton correspondant.

Attention, le chargement du programme depuis le *Project Manager* ne charge pas les configurations dans l'automate. Les configurations doivent être chargées à partir du Device Configurator.

3.4 Device properties



3.4.1 Memory

Code/Text/Extension memory

Représente la mémoire RAM disponible sur l'automate pour recevoir le programme ainsi que les textes et blocs de données compris entre les adresses 0 ...3999, mais aussi les textes et blocs de données d'adresse supérieur à 3999.

Sur certains anciens PCD cette mémoire peut être organisée de manière différente. La mémoire d'extension est séparée de la mémoire *Code/Text*. Nous avons alors deux paramètres:

Code/Text

Représente la mémoire RAM ou EPROM disponible sur l'automate pour recevoir le programme ainsi que les textes et blocs de données compris entre les adresses 0 ...3999

Sur certains anciens PCD cette mémoire peut être aussi de type EPROM. Alors que les nouveaux proposent de faire un backup du programme sur une mémoire flash.

Extension Memory

Représente la mémoire RAM disponible pour les blocs de données d'adresses supérieur à 3999.

User Program Memory Backup size (Flash)

Indique la taille de la mémoire Flash interne de l'automate, si elle est insuffisante, certains types de *devices* supportent une extension de mémoire Flash sur un ou plusieurs slots externes. Voir configurations sous *Memory Slots*.

La mémoire Flash est utilisée pour réaliser le backup du programme après son chargement sur l'automate. Le backup doit être activé dans le menu *Tools, Options* du *Project Manager* ou réalisé sur demande par le *Project Manager* avec la sélection du menu *Online, Flash Backup/Restore*.

Si une interruption de courant provoque une perte du programme en mémoire RAM, à la mise sous tension, le Firmware de l'automate restaure automatiquement le programme présent sur la mémoire Backup.

La mémoire Flash supporte aussi d'autres applications tel que la sauvegarde de données ou les fichiers sources du projet.

3.4.2 Password

Les automates disposent d'un mécanisme de protection par un mot de passe. S'il est protégé par un mot de passe, seul le protocole de communication réduit peut être utilisé. Il autorise seulement l'accès aux registres, temporisateurs, compteurs, indicateurs, entrées, sorties, blocs de données et l'horloge. Les autres données, tel le programme utilisateur, la configuration S-Bus et la table historique ne peuvent pas être accédées. Les commandes Run, Stop, Step,... sont mises hors fonction.

L'établissement de la communication sur un automate protégé par un mot de passe affiche un dialogue pour introduire le mot de passe avant de permettre l'usage du protocole complet. La sélection du bouton *Cancel* établit tout de même la communication mais avec les restrictions du protocole réduit.

En cas d'oubli du mot de passe, il faut effacer et reprogrammer la mémoire du PCD pour le configurer sans mot de passe ou avec un nouveau mot de passe.

Effacer la mémoire

Pour les mémoires RAM, c'est très simple. Il faut juste couper la tension de l'automate et de retirer la batterie pour un certain temps.

Pour les EPROM, la puce mémoire doit être retirée de l'automate et effacée avec la lampe UV, puis reprogrammée avec un programmeur d'EPROM avec le programme utilisateur et un mot de passe connu.

Pour les mémoires Flash, la puce mémoire doit être retirée de l'automate et effacée avec un programmeur d'EPROM qui supporte les puces mémoire Flash.

A la lecture des *hardwares settings*, le mot de passe du PCD est toujours absent.

Si non, cela permettrait de lire les mots de passe inconnus pour les charger ensuite dans un autre PCD qui deviendrait inaccessible jusqu'à ce que la mémoire soit retirée et effacée.

3.4.3 S-Bus

Si l'une des interfaces de communication définie sous *Onboard Communication* ou *Onboard Slots* sont utilisées, commencer par activer le support S-Bus et définir le numéro de station S-Bus.

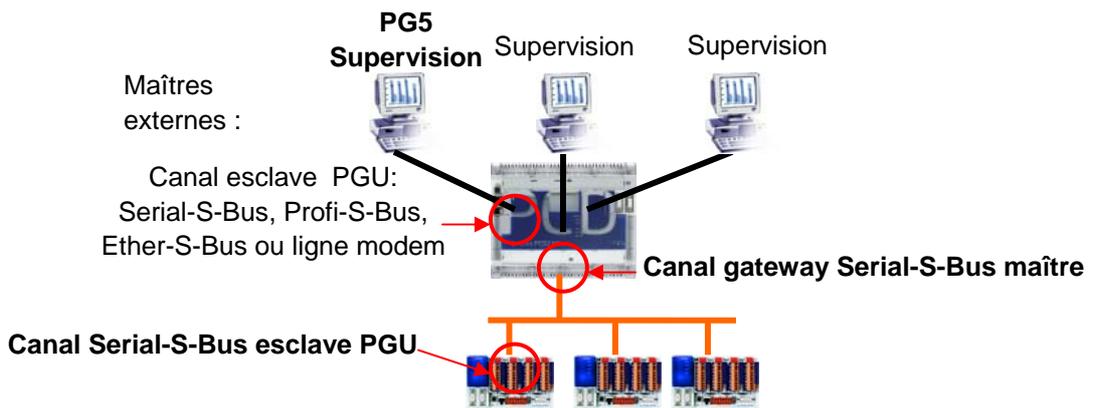
Le numéro de station S-Bus est commun à tous les canaux de communication de l'automate.

3.4.4 Power Supply.

L'alimentation 5 et 24 Volts nécessaire au fonctionnement des modules E/S est assurée pas le bus de l'automate. Il convient de vérifier si la configuration des E/S définie sous *Onborad Slots* du configurateur n'est pas supérieure aux valeurs de courant maximum disponible.

3.5 Serial S-Bus communication properties

Serial S-Bus est un réseau maîtres/esclave permettant de mettre les automates en réseau sur une ligne série RS 485/232 pour échanger les données entre PCD, superviser le processus. Il supporte aussi toutes les fonctionnalités de mise en service avec l'outil de programmation Saia PG5 (PGU) et une ligne modem analogique ou ISDN.



Selon le slots de communication série sélectionné, la fenêtre des propriétés permet de configurer le canal Serial-S-Bus comme esclave PGU pour ligne série, esclave PGU pour un ligne modem ou gateway maître. Commencer par activer la configuration souhaitée puis compléter les paramètres actifs.

General	
Port Number	0

Port Number

Numéro du canal de communication, ce numéro peut être utilisé par les instructions de programmes pour désigner le canal d'assignation (SASI) ou pour transmettre des télégrammes d'échange de données.

3.5.1 Full protocol (PGU) Serial-S-Bus (esclave PGU pour ligne série)

Serial S-Bus Port	
Enabled	Yes
Full Protocol (PGU)	Yes

Des slots de communications *Serial-S-Bus* esclaves supplémentaires peuvent être configurés avec un protocole réduit (sans la fonction PGU). Seule le slots PGU est configuré depuis le Device Configurator, la fonction esclave, sans la fonction PGU est configuré à partir du programme Fupla/IL à l'aide de l'instruction SASI.

Remarque, l'adresse S-Bus est définie dans les propriétés du device.

3.5.2 **Public Line S-Bus Modem** (esclave PGU pour un ligne modem)

Cette configuration est disponible que pour les slots de communications série qui offrent toutes les lignes de contrôles nécessaires pour se raccorder sur un modem. Offre les mêmes services que *Full protocol (PGU) Serial-S-Bus* mais au travers d'une ligne téléphonique et d'un modem analogique ou ISDN. Pour bénéficier d'une mise en service simplifiée, nous recommandons d'employer les modems fournis par Saia Burgess Controls.

Public Line S-Bus Modem	
Use Serial S-Bus For Modem	Yes
Full Protocol (PGU) on Modem Port.	Yes
Modem Name	
Modem Init	
Modem Reset	

Modem Name

Permet de sélectionner le type de modem utilisé. Il est recommandé d'employer les modems Saia-Burgess, ainsi ils sont déjà présents dans cette liste, les paramètres *Modem Init* et *Modem Reset* sont déjà configurés et testés.

Remarque, l'adresse S-Bus est définie dans les propriétés du device.

3.5.3 **Serial S-Bus Master Gateway.** (gateway maître)

La fonction *Gateway* est couramment utilisée pour relier deux réseaux de communication différents, adapter un outil de programmation Saia PG5, une supervision VisiPlus, une ligne modem sur le réseau Serial-S-Bus ou de créer un réseau multi-maîtres.

Tous les télégrammes reçus depuis les maîtres externes et qui ne sont pas destinés à la station *gateway* sont automatiquement redirigés vers le canal *gateway* maître.

Serial S-Bus Master Gateway	
Use For Gateway	Yes
First S-Bus Station	0
Last S-Bus Station	253

First/Last S-Bus Station

Permet de filtrer les télégrammes à transmettre sur le *Gateway* en fonction des numéros de station destinataires.

3.5.4 **S-Bus Mode and Timings**

S-Bus Mode and Timing	
Mode	Data Mode
Baud Rate	9600 Baud
Response Timeout [ms]	0
Training Sequence Delay [ms]	0
Turnaround Delay [ms]	0

Mode

Choix de la méthode appliquée par S-Bus pour indiquer le début et la fin des messages, doit être identique sur toutes les stations qui constituent le réseau. De préférence choisir le mode data, Cette méthode convient pour toutes les applications, en particulier, pour l'usage de modems ou équipements de réseaux normalisés.

Baudrate

Vitesse de communication, doit être identique sur toutes les stations du réseau.

Response Timeout

Temps d'attente en millisecondes de la station maître pour recevoir la réponse à un télégramme. Ce délai est particulièrement important lorsque l'esclave ne répond pas. S'il est trop long cela ralentit sévèrement la communication. S'il est trop court les réponses arrivent trop tard et provoquent l'échec du télégramme envoyé par le maître.

Par défaut la valeur des timings est de zéro, cela signifie que les timings par défaut sont appliqués. Dans la pratique nous les modifions uniquement lors de l'usage de *Gateways* ou de modems: généralement nous nous limitons à allonger le *Timeout* sur les maîtres externes.

Training Sequence Delay (TS)

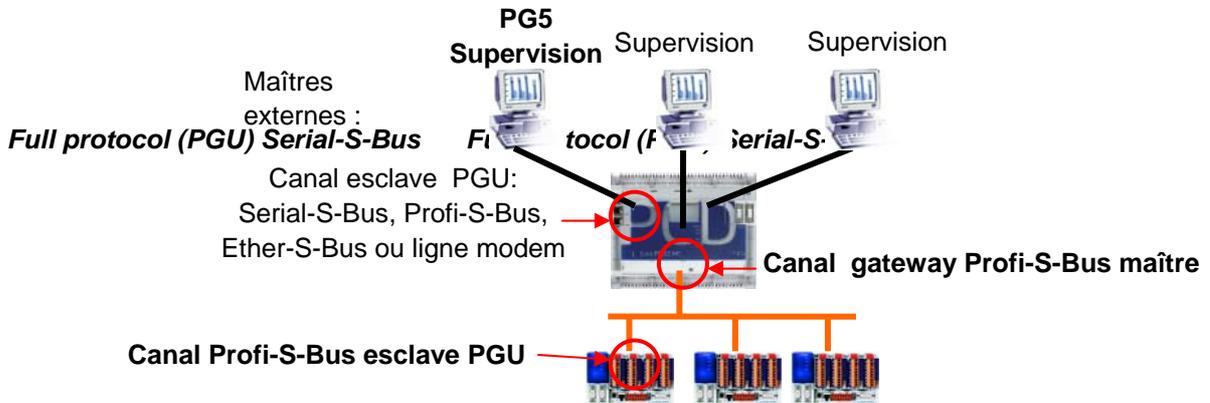
Délai en milliseconde entre l'activation du signal RTS (*Request To Send*) et la transmission du message. Avec S-Bus, l'activation de RTS, permet de sélectionner le driver RS-485 ou RS-422.

Turnaround Delay (TN)

Temps minimum en millisecondes, entre la fin d'une réponse et transmission du prochain télégramme. Cela donne le temps au poste éloigné pour commuter la ligne du mode émission en réception après avoir envoyé un message. Le délai TN est particulièrement important lors de l'utilisation de répéteurs PCD7.T100 ou modems pour lignes privée.

3.6 Profi S-Bus communication properties

Profi-S-Bus est un réseau de terrain multi-maître basé sur les standards Profibus FDL et le protocole SBC S-Bus. Il permet de mettre les automates en réseau pour échanger les données entre PCD, superviser le processus. Il supporte aussi toutes les fonctionnalités de mise en service avec l'outil de programmation Saia PG5 (PGU).



La fenêtre des propriétés correspondant au slot de communication S-Net, permet de configurer un canal Profi-S-Bus esclave ou gateway maître. Commencer par activer la configuration souhaitée puis compléter les paramètres actifs.

3.6.1 Profi-S-Bus (esclave)

Définit le canal Profi-S-Bus comme esclave ou esclave PGU. Cette définition peut être complétée avec la fonction maître en ajoutant une instruction SASI master dans le programme Fupla ou IL.

Profi-S-Bus	
Enabled Profi-S-Bus	Yes
Channel	10
Full Protocol (PGU)	Yes
Slave	Yes
Address	0
Baud Rate Profi-S-Bus	1.5 MBd
Bus Profile	S-Net

Esclave PGU (Yes)

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 (No)

Supporte uniquement l'échange de donnée avec les stations maîtres, systèmes de supervision et terminaux.

Address

Adresse de la station sur le réseau Profibus FDL. Pour désigner complètement l'automate avec lequel échanger des télégrammes, deux adresses sont nécessaires, l'adresse Profibus FDL et l'adresse S-Bus définie dans les propriétés du device.

Baudrate Profi-S-Bus

Vitesse de communication, doit être identique sur toutes les stations du réseau.

Bus Profile

Les timings pour la transmission sont regroupés dans trois profils :

- *User-defined* : les paramètres sont définis par l'utilisateur
- *S-Net* : utilise les valeurs les mieux adaptées au réseau Saia S-Net.
- *DP* : utilise les valeurs les mieux adaptées au réseau Profibus DP.

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 le réseaux.

3.6.2 Profi S-Bus Master Gateway. (gateway maître)

La fonction *Gateway* est couramment utilisée pour relier deux réseaux de communication différents, adapter un outil de programmation Saia PG5, une supervision VisiPlus, une ligne modem sur le réseau Profi-S-Bus.

Tous les télégrammes reçus depuis les maîtres externes et qui ne sont pas destinés à la station *gateway* sont automatiquement redirigés vers le canal *gateway* maître.

☐ Profi-S-Bus Master Gateway	
Use Profi-S-Bus For Gateway	Yes
First S-Bus Station Profi-S-Bus	0
Last S-Bus Station Profi-S-Bus	253
Response Timeout	0

First/Last S-Bus Station

Permet de filtrer les télégrammes à transmettre sur le *Gateway* en fonction des numéros de station destinataires.

Response Timeout

Par défaut la valeur du *Timeout* est de zéro, cela signifie que le timing par défaut est appliqué. Dans la pratique nous le modifions uniquement lors de l'usage de *Gateways* ou de modems: généralement nous nous limitons à allonger le *Timeout* sur les maîtres externes et jamais sur canal *gateway*.

3.6.3 Bus Parameters: user defined

Bus Profile	User defined
<input type="checkbox"/> Bus parameters	
Slot time	300
Min. Tsdr	11
Max. Tsdr	150
Quiet time	0
Setup time	1
Gap update factor	10
Highest station address	126
Max. retry limit	1

Les propriétés du bus contiennent tous les paramètres de communications Profi-S-bus. Ces valeurs ne peuvent être éditées que si le profil de bus *User-defined* est sélectionné. Les paramètres doivent être les mêmes pour toutes les stations sur le réseau.

Tous les délais sont donnés en bit (nombre de bits).

Slot Time

Temps maximum pour lequel l'émetteur de données ou du jeton attend la réponse du destinataire.

Min. Tsdr

Temps d'attente minimum d'un esclave après avoir reçu un télégramme jusqu'à l'émission de la réponse au maître.

Max. Tsdr

Temps d'attente maximum d'un esclave après avoir reçu un télégramme jusqu'à l'émission de la réponse au maître.

Quiet Time

Temps d'attente de l'émetteur après la fin d'une trame avant d'activer sa réception.

Setup Time

Temps écoulé entre un événement et la réaction correspondante

Gap Update Factor

Nombre de rotations du jeton entre deux entretiens GAP (mise à jour) cycles.

Highest Station Address

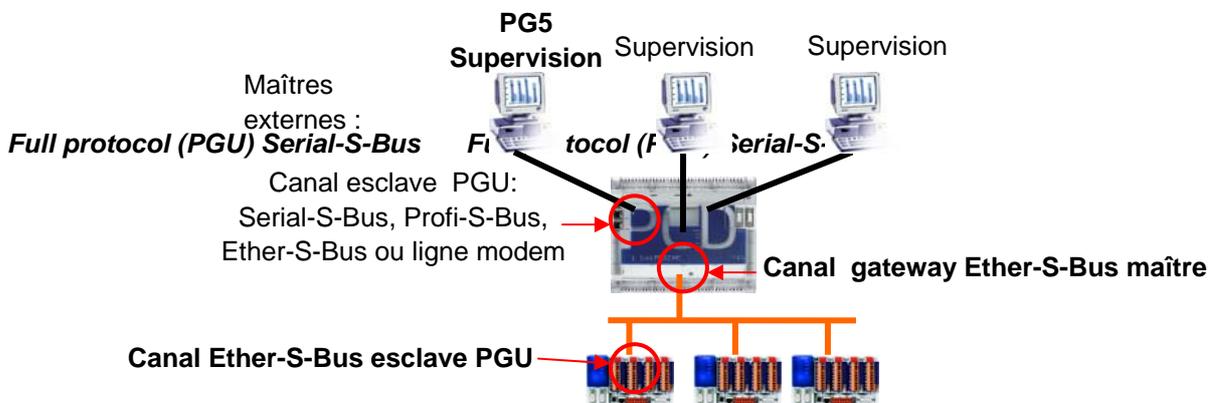
La plus haute adresses (HSA) sur le réseau

Max. Retry Limit

Nombre de répétitions sans accusé de réception avant d'envoyer un accusé de réception négatif.

3.7 Ether-S-Bus communication properties

Ether-S-Bus est un réseau multi-maître basé sur les standards Ethernet et le protocole SBC S-Bus. Il permet de mettre les automates en réseau pour échanger les données entre PCD, superviser le processus. Il supporte aussi toutes les fonctionnalités de mise en service avec l'outil de programmation Saia PG5 (PGU).



La fenêtre des propriétés correspondant au slot de communication S-Net, permet de configurer un canal Ether-S-Bus esclave ou gateway maître. Commencer par activer la configuration souhaitée puis compléter les paramètres actifs.

3.7.1 Ether-S-Bus (esclave)

Défini le canal Ether-S-Bus comme esclave ou esclave PGU. Cette définition peut être complétée avec la fonction maître en ajoutant une instruction SASI master dans le programme Fupla ou IL.

TCP/IP	
Channel Number	9
TCP/IP Enabled	Yes
Ethernet RIO Network	None
IP Address	0.0.0.0
Subnet Mask	255.255.255.0
Default Gateway	0.0.0.0
+ Access Control List	Hide

Ether-S-Bus	
Channel Number	9
Ether-S-Bus Enabled	Yes
IP Node	0
PGU Port	Yes
Slave	Yes
Network Groups	(Default)

IP Address

Adresse de la station sur le réseau Ethernet. Pour designer complètement l'automate avec lequel échanger des télégrammes, deux adresses sont nécessaires, l'adresse Ethernet et l'adresse S-Bus définie dans les propriétés du device.

Subnet mask

Défini la part de l'adresse IP qui appartient à l'identification sur le réseau et de l'hôte.

Tous les bits correspondant au réseau sont à 1 alors que les bits correspondant à l'hôte sont à zéro. L'identificateur de réseau et le résultat d'une combinaison logique ET entre l'adresse IP et le *subnet mask*. Chaque hôte TCP/IP exige un *subnet mask*, même sur un segment de réseau. Le *subnet mask* par défaut est donc 255.255.255.0 (ClassC)

Default Gateway

Adresse du routeur par défaut.

IP Node

Numéro du nœud TCP/IP. Le nœud est utilisé dans le programme d'application de l'automate pour référencer l'adresse IP de la station esclave avec laquelle échanger des données.

PGU port

Esclave PGU (Yes)

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 Saia PG5.

Esclave (No)

Supporte uniquement l'échange de donnée avec les stations maîtres, systèmes de supervision et terminaux.

Network Groups

Affiche la fenêtre de dialogue pour configurer les automates comme client ou serveur sur le groupe de réseau.

3.7.2 Profi S-Bus Master Gateway. (gateway maître)

La fonction *Gateway* est couramment utilisée pour relier deux réseaux de communication différents, adapter un outil de programmation Saia PG5, une supervision VisiPlus, une ligne modem sur le réseau Ether-S-Bus.

Tous les télégrammes reçus depuis les maîtres externes et qui ne sont pas destinés à la station *gateway* sont automatiquement redirigés vers le canal *gateway* maître

TCP/IP S-Bus Master Gateway	
Use TCP/IP For Gateway	Yes
First S-Bus Station	0
Last S-Bus Station	253
Response Timeout	0

First/Last S-Bus Station

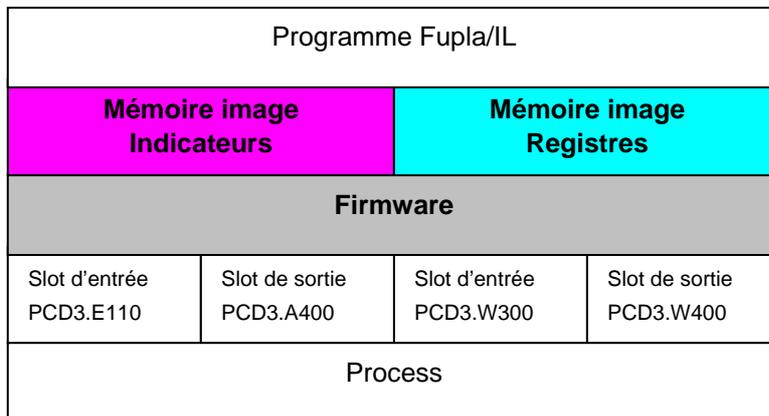
Permet de filtrer les télégrammes à transmettre sur le *Gateway* en fonction des numéros de station destinataires.

Response Timeout

Par défaut la valeur du *Timeout* est de zéro, cela signifie que le timing par défaut est appliqué. Dans la pratique nous le modifions uniquement lors de l'usage de *Gateways* ou de modems: généralement nous nous limitons à allonger le *Timeout* sur les maîtres externes et jamais sur canal *gateway*.

3.8 On board slots properties, configuration de la mémoire image.

La gestion des entrées/sorties de l'automate peuvent être gérées à l'aide d'une mémoire image constituée d'indicateurs/registres tenus à jour par le firmware de l'automate. Le programme Fupla ou IL n'accèdent plus directement sur les entrées/sorties pour les opérations de lecture/écriture des entrées/sorties mais travail avec la mémoire image.



La mémoire image est configurable avec les automates PCD1.Mxxx0, PCD2.Mxxx0 et PCD3.

Mais il est toujours possible d'accéder directement aux entrées/sorties sans configurer la mémoire image. Cela permet de garantir la compatibilité avec les anciens projets des précédentes version de PG5 et les automates qui ne la supporte pas encore: PCD1.M1xx, PCD2.M1xx, PCD2.M480.

3.8.1 Device properties, configurations nécessaires.

Si le firmware de l'automate supporte la mémoire image, la gestion des E/S peut être activée ou désactivée avec les paramètres ci-dessous.



Input/Output Handling Enabled

Yes, tous les paramètres d'E/S du module sont disponibles pour supporter la configuration de la mémoire image.

No, tous les paramètres définis dans *Onboard slots* sont sans influence sur le programme utilisateur.

Peripheral Address Definition

Définition automatique ou manuelle des plages d'adresses pour la mémoire image de chaque module.

3.8.2 Onboard slots, configurations des modules E/S.

Permet de configurer les modules E/S présents dans l'automate en les sélectionnant dans la fenêtre *E/S Selector* pour les glisser vers un slots E/S.

La sélection de l'un des slots E/S affiche les paramètres de configuration relatifs dans une fenêtre des propriétés.

La configuration des slots E/S n'est pas nécessaire si la mémoire image n'est pas supportée par le PCD ou pas activée dans les propriétés du *Device*.

3.8.3 Propriétés des E/S binaires.

Permet de définir les configurations nécessaires à l'élaboration de la mémoire image. Si la section *Media mapping* n'est pas affichée, cela signifie que le firmware de l'automate sélectionné sous *Device* ne supporte pas cette fonctionnalité.

Slot 0 : PCD2.E110, 8 Digital Inputs, 24VDC	
General	
Base Address	0
Power Consumption	
Power Consumption 5V [mA]	24
Media Mapping	
Media Mapping Enabled	No
Media Type	Flag
Number Of Media	8

Base Address

Adresse de base du module: 0, 16,32,...

Enabled Media Mapping

Actualisation cyclique de la mémoire image (registre ou indicateurs) avec les valeurs d'entrée, sorties présents sur les slots de l'automate.

Media Type

Type de media utilisé pour sauvegarder les valeurs des entrées ou sorties . Pour les E/S analogiques, le type est toujours registre. Pour les E/S binaires, c'est par défaut le type indicateur mais peut aussi être registre.

Number of Media

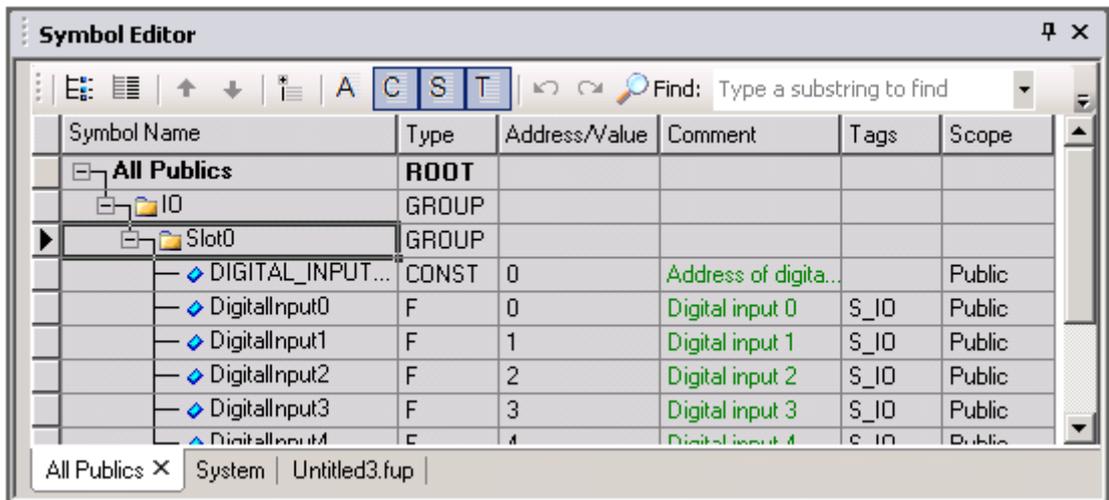
Nombre de medias nécessaires pour sauvegarder les valeurs. Par exemple, un module d'entrée numérique PCD3.E110 exige 8 indicateurs.

Symbol definition

Sélectionner ce paramètre pour afficher le bouton qui permettra d'afficher l'éditeur de symboles et de modifier les noms, commentaires et adresses des symboles correspondants aux E/S.

Media Mapping						
Slots / Symbols	Type	Address	Comments	Scope	Tags	
PCD2.M5540, CPU with 1 MBytes RAM, 8 I/O slots (expandable), 3 communication slots, USB, Profi-S-Net, RS-232, Ethernet.						
Slot 0, PCD2.E110, 8 digital inputs, 15..30VDC, 8ms, current draw 12mA at 5V.						
— S.IO.Slot0.DigitalInput	F [8]			Public	S_IO	
— IO.Slot0.DigitalInput0	F	S.IO.Slot0.DigitalInput + 0	Digital input 0	Public	S_IO	
— IO.Slot0.DigitalInput1	F	S.IO.Slot0.DigitalInput + 1	Digital input 1	Public	S_IO	
— IO.Slot0.DigitalInput2	F	S.IO.Slot0.DigitalInput + 2	Digital input 2	Public	S_IO	
— IO.Slot0.DigitalInput3	F	S.IO.Slot0.DigitalInput + 3	Digital input 3	Public	S_IO	
— IO.Slot0.DigitalInput4	F	S.IO.Slot0.DigitalInput + 4	Digital input 4	Public	S_IO	
— IO.Slot0.DigitalInput5	F	S.IO.Slot0.DigitalInput + 5	Digital input 5	Public	S_IO	
— IO.Slot0.DigitalInput6	F	S.IO.Slot0.DigitalInput + 6	Digital input 6	Public	S_IO	
— IO.Slot0.DigitalInput7	F	S.IO.Slot0.DigitalInput + 7	Digital input 7	Public	S_IO	

Après le build du programme, ces symboles sont disponibles pour élaborer les programmes Fupla et IL, nous les trouvons parmi les symboles All Publics sous: S.IO.Slot0, ...



3.8.4 Propriétés des E/S analogiques.

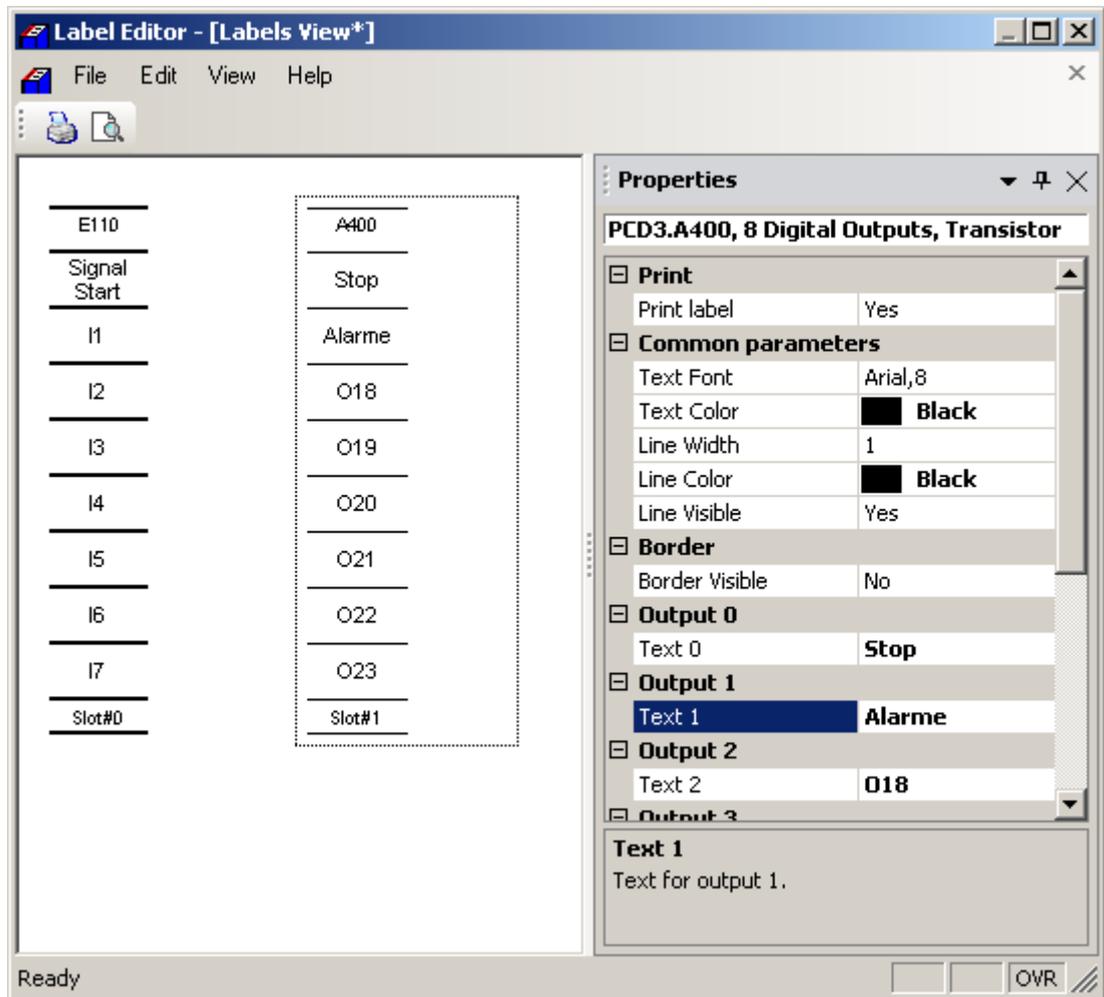
Se configure comme pour les modules d'E/S binaires, nous retrouvons les mêmes paramètres pour définir le *Media mapping* du module analogique sur des registres.

Analogue Output 1	
Output 1 Range	0..10V in mV or % resolution
Minimal Value Output 1	0
Maximal Value Output 1	10000
Reset Value Output 1	0

Par contre une nouvelle section offre les paramètres nécessaires à la configuration de la grandeur d'E/S de chaque canal.

3.9 Edition des étiquettes pour les modules I/O.

Le menu *Tools, Label* permet d'afficher la fenêtre ci-dessous pour préparer les étiquettes à placer sur les modules d'E/S PCD3 et le couvercle des PCD1.Mxxx0 et PCD2.Mxxx0. Sélectionner l'étiquette et éditer la fenêtre des propriétés.



Border Visible

Trace le pourtour de l'étiquette pour faciliter son découpage. Remarque, des feuilles prédécoupées sont disponibles pour imprimer les étiquettes.

La dimension des étiquettes imprimées peut paraître deux fois trop large car il est prévu de les plier en deux sur la longueur. Elles sont ainsi plus stable dans leurs support.

3.10 Extension du device configurateur avec des nouveaux *devices* et modules E/S

Au cas où de nouveaux modules ou automates sont disponibles, il n'est plus nécessaire d'installer une nouvelle version du device configurateur ou de PG5 pour les supporter. Il suffit d'installer le fichier XML qui décrit le nouveau hardware dans un sous répertoire d'installation de PG5 et de redémarrer le software.

C:\Program Files\SBC\PG5_21\DeviceTemplates

Table des matières

4 PCD – TYPES DE DONNEES 2

4.1 Au sommaire de ce chapitre2

4.2 Données matérielles (Hardware).....3

 4.2.1 Entrées et sorties digitales3

 4.2.2 Horloge.....4

 4.2.3 Entrées interruptives5

4.3 Données internes (softwares)6

 4.3.1 Indicateurs (Flags).....6

 4.3.2 Registres7

 4.3.3 Constantes.....8

 4.3.4 Temporisateurs et compteurs.....9

 4.3.5 Textes et blocs de données12

 4.3.1 Tableau récapitulatif14

4 PCD – Types de données

4.1 Au sommaire de ce chapitre

Ce chapitre donne un aperçu de tous les données utilisable dans les programmes PCD (entrées, sorties, indicateurs, registres, compteurs, timer,...), ainsi que leur usage et leur plage d'adresses.

4.2 Données matérielles (Hardware)

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

4.2.1 Entrées et sorties digitales

Les interfaces d'entrées et sorties assurent l'échange des données binaires (1 bit) avec l'environnement extérieur de l'automate. Ces interfaces adaptent les signaux extérieurs à l'environnement interne de l'automate et assure une bonne protection contre les perturbations d'un environnement extérieur hostile.

Les entrées rapportent l'état du processus : fins de course, boutons poussoirs, détecteurs de proximité, capteurs...

Les sorties agissent sur le processus : la commande des vannes, des lampes, des moteurs , ...

Les entrées sont uniquement accessibles en lecture, alors que les sorties sont accessibles en lecture et écriture.

Les modules d'entrées et sorties sont insérés à l'intérieur des slots de l'automate. Le choix du slot définit l'adresse de base des données.

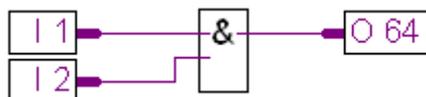
Exemple :

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

4.2.2 Horloge

Les automates PCD intègrent une horloge temps réel, souvent abrégée *RTC* (de l'anglais *Real Time Clock*). Une instruction permet de charger la date et l'heure dans un registre.

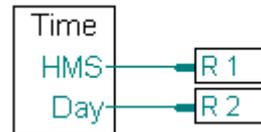
Exemple :

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 l'heure et la date de l'horloge PCD et les copie sur deux registres R 1 et R 2. 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)
```

4.2.3 Entrées interruptives

Certains automates 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.

Exemple :

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

Programme IL:

```

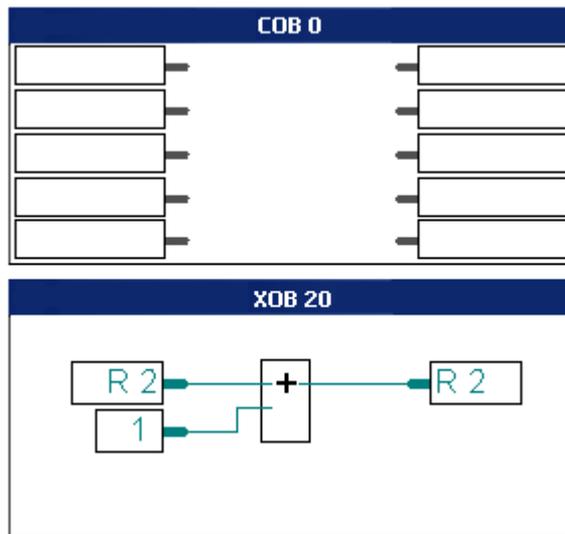
COB 0 ; Programme
      0 ; principal

ECOB

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

EXOB
    
```

Programme Fupla:



Fbox : Entier, Addition Add

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 rebonds des contacts mécaniques), une entrée digitale normale ne peut pas compter des impulsions dont la fréquence est 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 PCD2/3.H1xx dont la fréquence de comptage maximum est de 10 à 160 kHz selon les modules.

4.3 Données internes (softwares)

Les données internes n'ont aucune interaction directe avec l'environnement extérieur de l'automate, elles sont qualifiées de software. Ce sont des adresses mémoires pour l'usage interne des programmes.

4.3.1 Indicateurs (Flags)

Les indicateurs sont des données binaires (1 bit). 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. (*Build Options*)

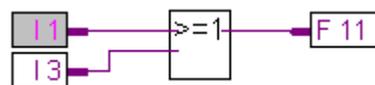
Exemple :

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

Programme IL:

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

Programme Fupla:



Fbox : Binaire, OU

Configuration des indicateurs non volatiles

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



Double clic de la souris

Media Allocation	
Last Timer	31
Timer Timebase in milliseconds (10..10000)	100
Has Volatile Flags	Yes
Last Volatile Flag	2999
Dynamic Volatile Flags	
First	2500
Last	2999
Dynamic Nonvolatile Flags	
First	7500
Last	8191

4.3.2 Registres

Les registres représentent des nombres entiers ou flottants de 32 bits.

Tous les registres sont non volatiles, leurs valeurs ne sont pas perdues lorsque l'automate est mis hors tension.

Les registres sont utiles pour effectuer des opérations arithmétiques sur les nombres entiers et flottants ainsi que pour les tâches de mesure et de régulation sur des valeurs analogiques.

Avec l'éditeur Fupla, les liens aux registres sont représentés par différentes couleurs: bleu pour un nombre entier et jaune pour un nombre flottant. Les registres représentés par des couleurs différentes ne peuvent pas être liés sur des Fboxes dont les couleurs de liens ne correspondent pas. Il faut d'abord convertir le format des registres ou employer une Fbox adéquate.

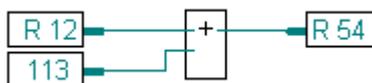
Exemple :

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

Le type 'K' indique que '113' est une constante et non pas un registre. Les valeurs de constantes 'K' sont non signées et comprises entre 0 et 16383. Pour utiliser une constante négative ou hors de ces limites, charger la constante dans un registre avec l'instruction LD puis additionner le registre en lieu et place de la constante.

Configuration des registres dynamiques

Le PG5 dispose d'un puissant outil d'allocation dynamique qui évite de préciser l'adresse des données software du programme. Le principe consiste à attribuer un nom de symbole à la donnée sans préciser d'adresse. Il peut être nécessaire de modifier ces paramètres si les programmes utilisent beaucoup de registres dynamiques et qu'une erreur d'assemblage le signale avec le message *Auto allocation overflow for type: R*. Il convient alors d'étendre la plage des adresses dynamiques pour les registres.

Dynamic Registers	
	2000; 4095
First	2000
Last	4095

4.3.3 Constantes

Les constantes représentent des nombres entiers ou flottants de 32 bits directement utilisés par le programme ou chargés dans des registres, temporisateurs, compteurs.

Exemple de constante : coefficients fixes comme π (PI) = 3,1415.

Exemple :

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 supportées par l'opération 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.

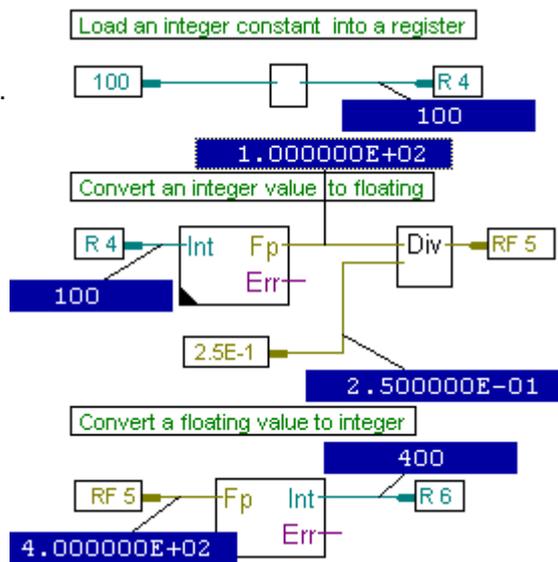
Utilisation des constantes dans les programmes

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

4.3.4 Temporisateurs et compteurs

Les temporisateurs et compteurs sont des nombres entiers positifs de 32 bits utilisés par le programme pour réaliser le décomptage d'unités de temps et le comptage d'informations concernant le processus.

Temporisateurs 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 1599 aux compteurs.

Par défaut, la base de temps des temporisateurs est de 100 ms. Toutes les 100 ms, le firmware de l'automate diminue chaque temporisateur d'une unité jusqu'à ce qu'il atteigne la valeur zéro. (Les temporisateurs sont uniquement décrémentés)

Les compteurs sont incrémentés et décrémentés par le programme avec les instructions INC et DEC ou les entrées des Fboxes correspondantes selon les informations provenant du processus.

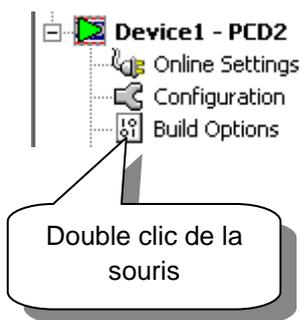
La valeur des compteurs et temporisateur peut être initialisées par une instruction LD ou l'entrée correspondante des Fboxes.

Temporisateurs et compteurs sont aussi exploitables avec des instructions binaires. Si le temporisateur ou le 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).

Ces informations sont souvent utiles pour le contrôle du processus. Déterminer si une temporisation est terminée, répétition de tâches supervisée par un compteur.

Configuration des temporisateurs et compteurs

La répartition des adresses entre les temporisateurs et compteurs, la modification de la base de temps des temporisateurs et l'allocation dynamiques des adresses sont supportés par la fenêtre *Build Options* ci-dessous:



Last Timer	31
Timer Timebase in milliseconds (10..10000)	100
<input type="checkbox"/> Dynamic Timers	5; 31
First	5
Last	31
<input type="checkbox"/> Dynamic Counters	1400; 1599
First	1400
Last	1599

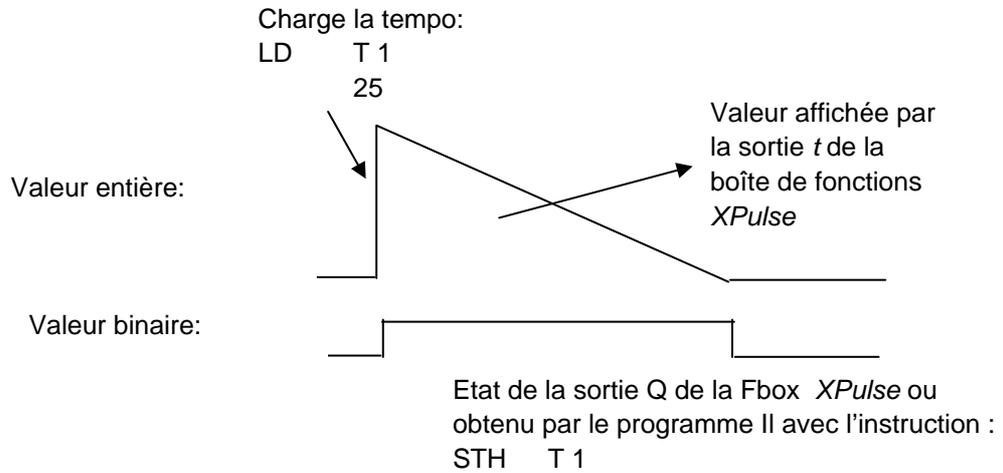


Informations techniques

La configuration d'un grand nombre de temporisateurs et la diminution de la base de temps augmentent la charge du PCD. Exemple: 100 temporisateurs mobilisent environ 2 % de la capacité PCD. Soyez donc vigilant lorsque vous envisagez ces modifications!

Exemple de temporisation

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

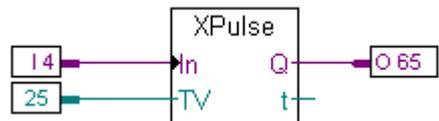


Programme IL:

```

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 *Build Options* 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 *Build Options*.

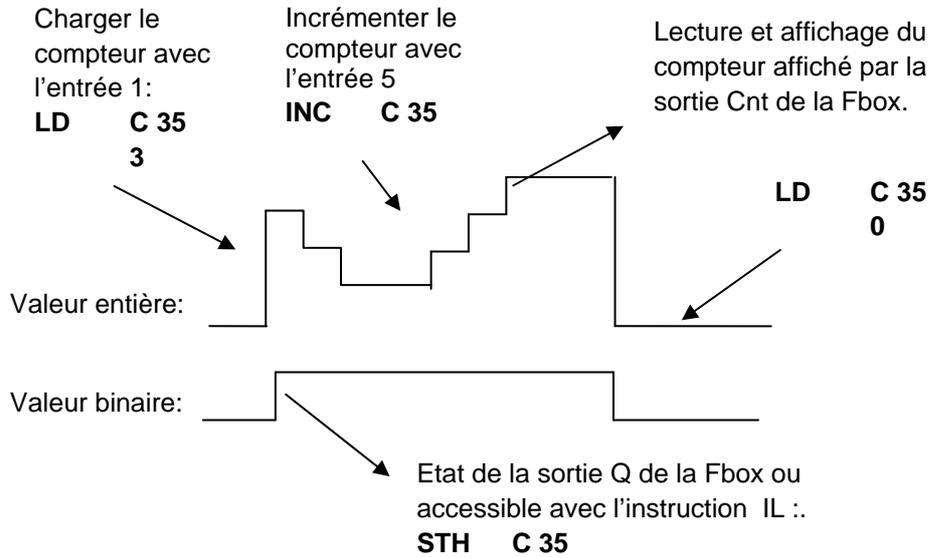
Format: T#nnnS|MS

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

Exemple de 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 décrémente le compteur.

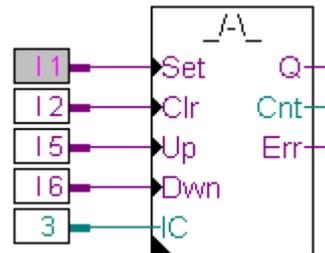


Programme IL:

```

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

4.3.5 Textes et blocs de données

Les textes et blocs de données sont des données non volatiles. Les textes représentent des chaînes de caractères, les blocs de données sont des tables de valeurs de 32 bits.

Les textes sont utilisés pour afficher des alarmes, paramètres sur des terminaux, envoyer des SMS, E-mail, ...

Les DB sont utilisés pour l'acquisition de données, la réalisation de tables de valeurs, ...



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 mémoire RAM, indépendante de la mémoire principale.

Les blocs de données et les textes avec les adresses 0 à 3999 sont conservés en mémoire principale avec le programme utilisateur. Si cette mémoire est de type Flash EPROM ou EPROM, veuillez tenir compte qu'en mode de fonctionnement *RUN*, celle-ci est accessible en lecture uniquement! Le programme ne peut pas modifier le contenu des blocs de données et textes.

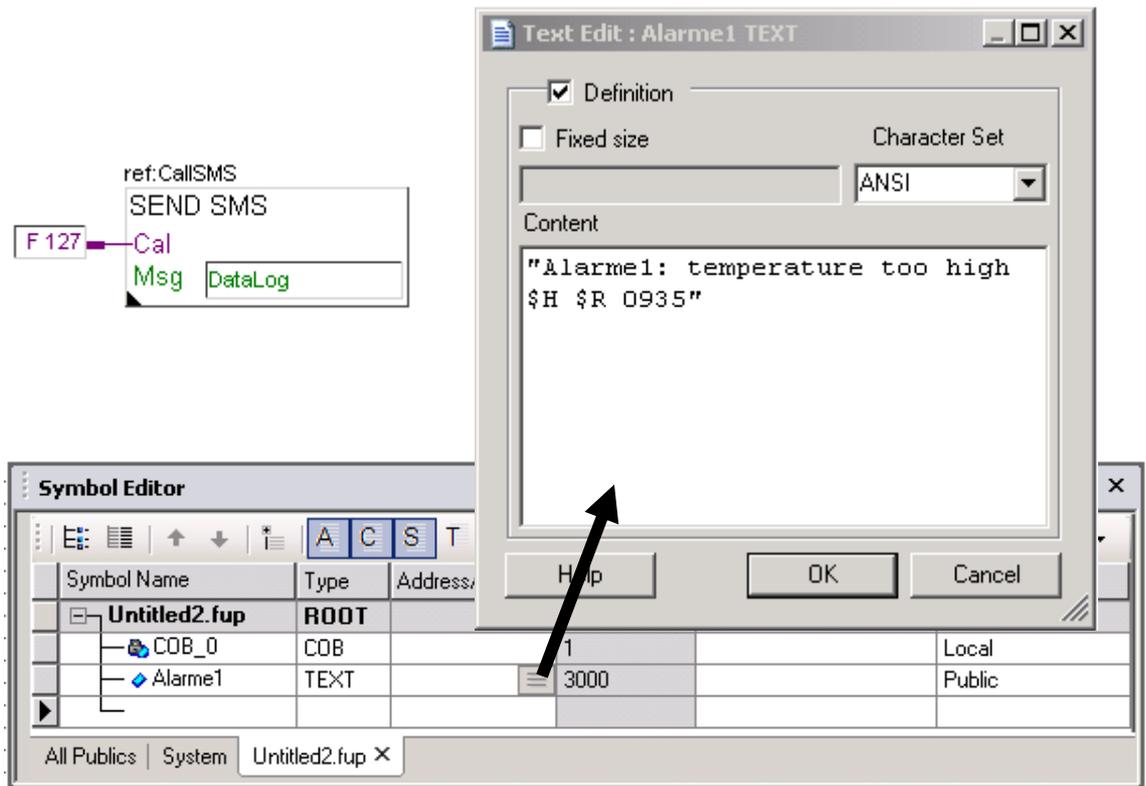
Si la lecture et l'écriture sur les blocs de données et textes doit toujours être garantie pour le programme, il convient d'utiliser les adresses de textes et blocs de données sur la mémoire d'extension avec les adresse 4000 et supérieures. Cette mémoire est en effet toujours une RAM, donc accessible en lecture et en écriture.

Avec les mémoires Flash Backup, ce souci n'existe pas puisque le programme est sauvegardé en mémoire RAM et que seul la copie du programme original est en flash.

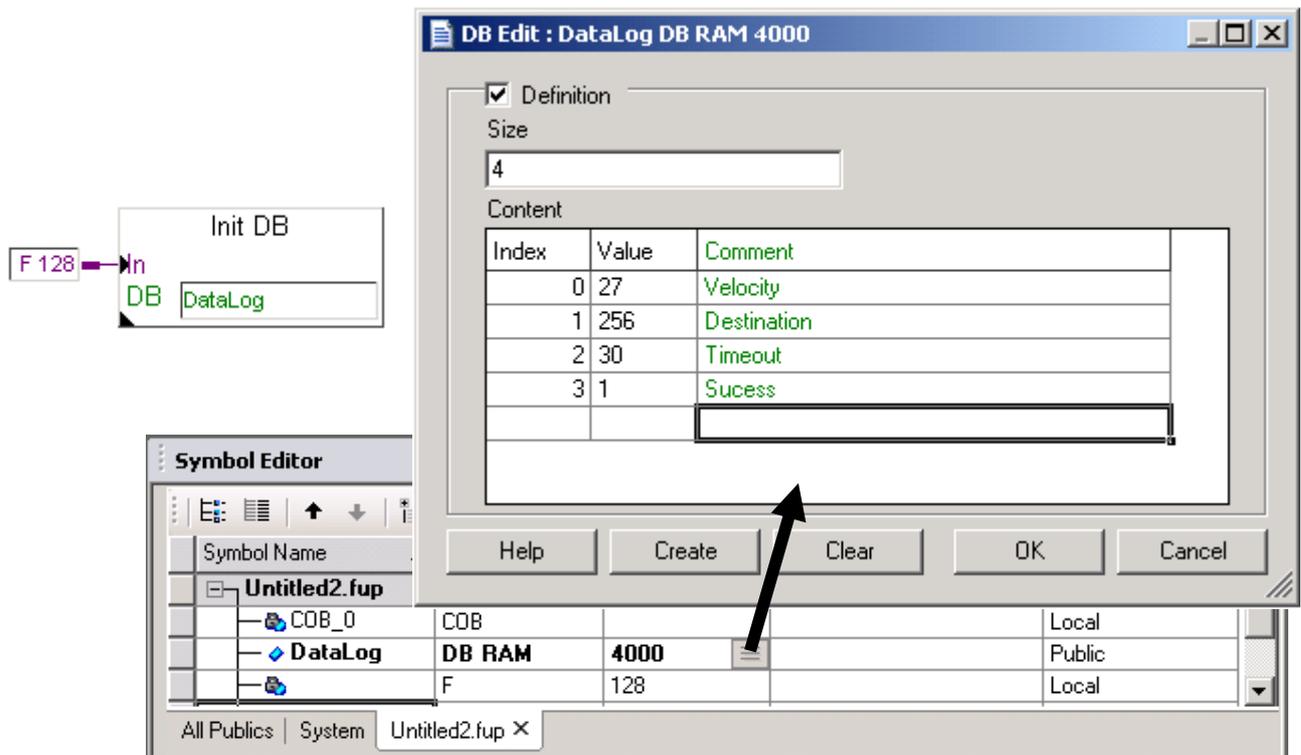
Déclaration de textes et blocs de données avec le code IL

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.

Déclaration d'un texte avec l'éditeur de symboles



Déclaration d'un bloc de donnée avec l'éditeur de symboles



4.3.1 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..16383	0,1		2) Non
Registres	R	0..16383		-2 147 483 648...+2 147 483 647 -9.22337E+18...+9.22337E+18	Non
Constantes	K			-2 147 483 648 à +2 147 483 647 -9.22337E+18 à +9.22337E+18	
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
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, mais configurable à partir des *Build Options*.
- 3) sauvegardé sur la même mémoire que les programmes (RAM / EPROM / FLASH)
- 4) sauvegardé sur la mémoire d'extension (RAM)

Contenu

5	L'EDITEUR DE SYMBOLES	2
5.1	Introduction	2
5.2	Aperçu	2
5.2.1	Composants de l'Editeur de Symboles	2
5.2.2	Les éléments d'un symbole	5
5.2.3	Regroupement des symboles	7
5.2.4	Champs d'applications des symboles - scope	7
5.2.5	Affichage des filtres.....	9
5.2.6	Définitions de symboles dans un fichier 'sy5'	10
5.2.7	Définition des symboles dans des fichiers 'xls', 'txt', 'rxp'	11
5.2.8	Définir des symboles pour les réseaux de communications	12
5.2.9	Définition des symboles dans un fichiers commun	13
5.3	Travailler avec les symboles	14
5.3.1	Ajouter des symboles à la liste des symboles	14
5.3.2	Ajout de plusieurs symboles	15
5.3.3	Symboles référencés	16
5.3.4	Ajout d'un symbole en travaillant avec l'éditeur IL	16
5.3.5	Ajout d'un symbole en travaillant avec l'éditeur Fupla.....	18
5.3.6	Modes d'adressage de symbole	19
5.3.7	Utilisation des symboles dans un programme	20
5.3.8	Recherche d'un symbole	24
5.3.9	Allocation automatique des adresses	25
5.3.10	Saisie de texte	26
5.3.11	Saisie de DB	27
5.3.12	Références croisées des symboles	27
5.3.13	Édition dans la grille de l'éditeur de symboles	28
5.3.14	Fonctionnalité d'ordonnancement dans la liste des symboles	30
5.3.15	Importation des déclaration des symboles 'EQU'	31
5.3.16	Importation/fusion de symboles	31
5.3.17	Exportation de symboles	33
5.3.18	'Tags' des symboles	34
5.3.19	Adresse actuelle.....	34
5.3.20	Initialisation de la valeur d'un symbole.....	35
5.3.21	Les mots réservés	36
5.3.22	Messages d'erreurs et avertissements	36

5 L'Editeur de Symboles

5.1 Introduction

Ce chapitre donne un aperçu de l'éditeur de symboles et de l'utilisation des symboles dans les programmes.

5.2 Aperçu

Un symbole est un nom qui représente l'adresse d'une entrée, d'une sortie, d'un indicateur, d'un registre, etc. Il est conseillé d'utiliser des noms de symboles lors de l'édition d'un programme, plutôt qu'une adresse d'un indicateur, d'un registre, d'un texte ou d'un DB.

Donner des noms significatifs rendra le programme plus facile à lire et à comprendre. Par exemple, vous pouvez assigner le symbole 'Oil_Pump' à O 32 et ensuite utiliser le symbol 'Oil_Pump' pour représenter la sortie O 32 dans votre programme.

De plus, l'utilisation des symboles permet la correction d'une adresse ou du type de données à l'aide de l'éditeur de symboles. Au lieu de faire la correction dans tous les endroits où le symbole est utilisé dans les programmes, il suffit de faire cette correction dans l'éditeur de symboles. Les modifications apportées à la définition du symbole sont automatiquement propagées à tous les endroits du programme où ce symbole est utilisé. Cette façon de faire limite considérablement le risque d'erreurs, qui dans ce genre de cas sont souvent difficiles à trouver.

Avant de commencer à écrire un programme, tous les symboles nécessaires peuvent être définis et répertoriés dans un outil appelé éditeur de symboles - 'Symbol Editor'. Une fois déclarés dans l'éditeur de symboles, ceux-ci sont connus dans tous les éditeurs de programme du PG5. L'éditeur de symboles est très utile pour trouver des éléments à l'intérieur des fichiers du programme, dans les rapports d'erreurs de programmation suite à la compilation, ou pendant le processus de débogage.

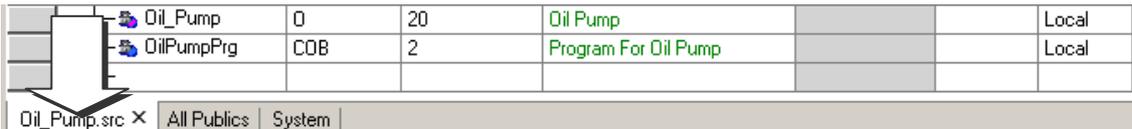
5.2.1 Composants de l'Editeur de Symboles

Ouverture de l'Editeur de Symboles :

La définition des symboles est sauvegardée dans les fichiers de programme (IL / Fupla etc). Dès qu'un fichier du programme est ouvert dans un éditeur, l'éditeur de symboles affiche la liste des symboles en rapport avec la partie du programme éditée.

Exemple:

L'ouverture du fichier de programme nommé 'Oil_Pump.src' ouvre automatiquement l'éditeur de symboles avec le même nom.



Oil_Pump	O	20	Oil Pump			Local
OilPumpPrg	COB	2	Program For Oil Pump			Local

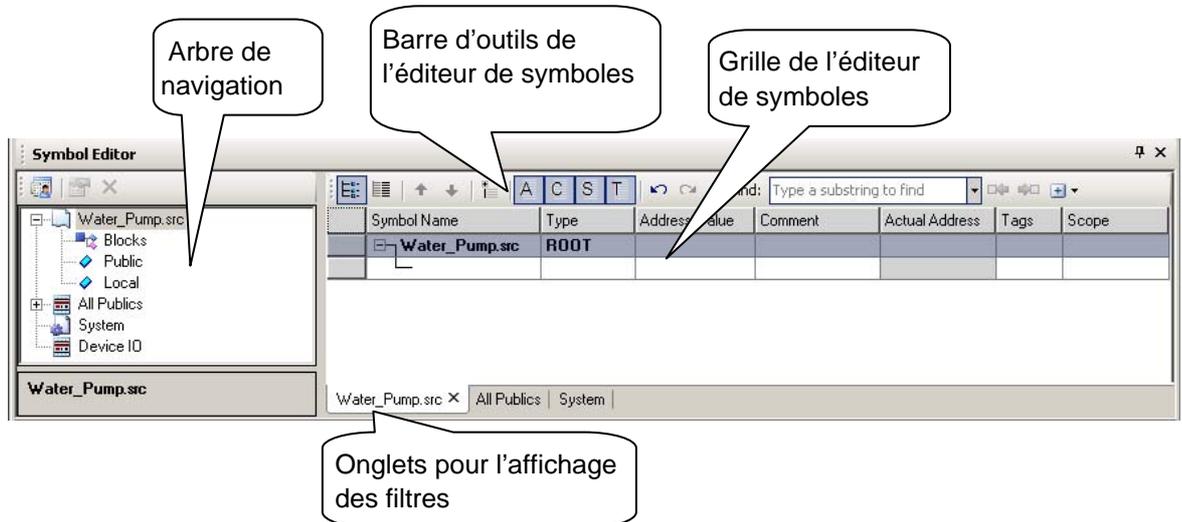
Oil_Pump.src × All Publics | System |



Show Hide
Symbols Editor

La fenêtre de l'éditeur des symboles peut être affichée avec le bouton 'Show / Hide Symbol Editor', ou via la commande de menu 'View / Symbol Editor'.

Composants de l'Editeur de Symboles:



Grille de l'éditeur de symboles:

La grille de l'éditeur de symboles est de type tableur. Une ligne vide est toujours présente à la fin de la liste des symboles, afin de permettre la définition de nouveaux symboles d'une façon aisée. Différentes actions peuvent être réalisées à partir du menu contextuel - menu affiché en cliquant sur la touche droite de la souris.

Barre d'outils de l'éditeur de symboles

Un ensemble de boutons est défini dans la partie supérieure de l'éditeur de symboles afin de pouvoir effectuer rapidement des actions sur les symboles. Il s'agit notamment de 'Show/Hide Navigator Tree' – afficher/cacher l'arbre de navigation, 'List view/Group view' – affichage des symboles sous forme de liste ou de groupe, 'Move up/down symbols' – déplacer un symbole vers le haut/bas, 'Expand/collapse' – étendre/réduire les groupes de symboles, 'Show/hide columns in grid' – afficher/cacher des colonnes de la grille, 'Undo/Redo' – annuler/répéter, 'Find symbols' – recherche de symboles etc.

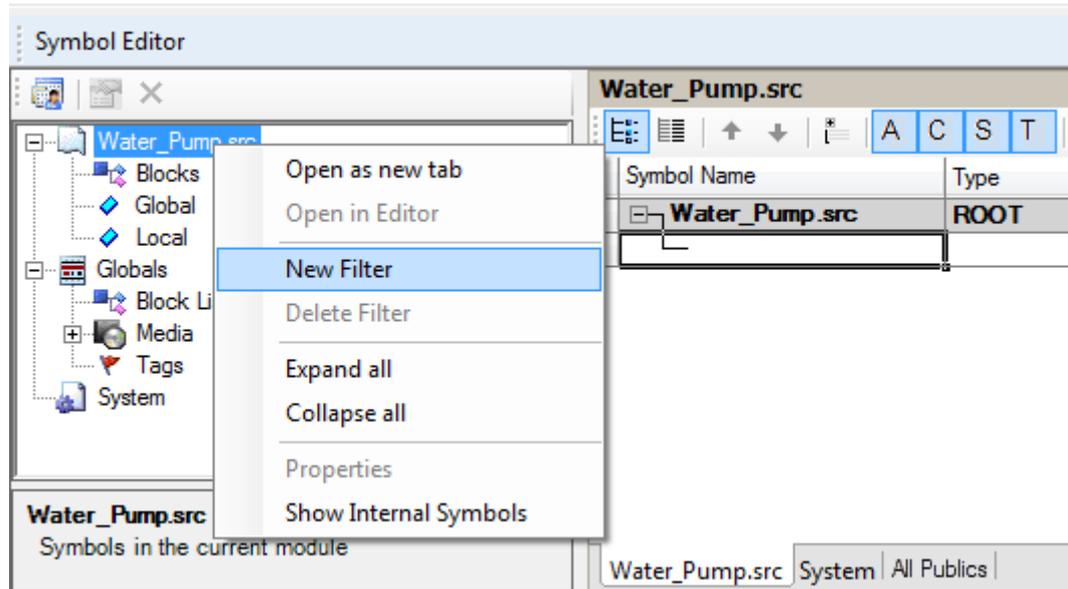
Onglets pour l'affichage des filtres

La liste des symboles est organisée en différentes vues, dépendant du filtre défini. Les différentes vues sont accessibles au moyen d'onglets, définis au bas de la grille des symboles. Pour afficher la grille des symboles correspondants au filtre, il suffit de cliquer sur l'onglet.

L'arbre de navigation:

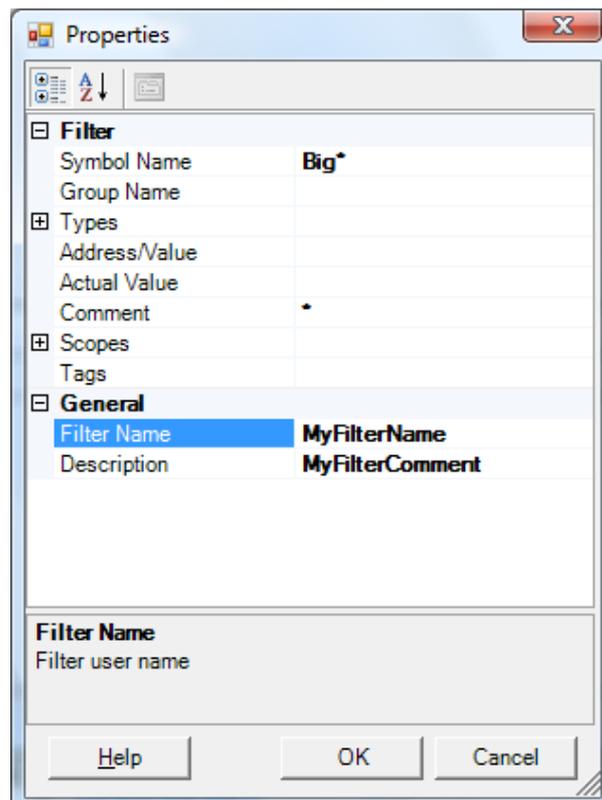
L'arbre de navigation contient la structure des filtres prédéfinis pour afficher les symboles. En sélectionnant un filtre, une nouvelle grille de symboles sera affichée, correspondant aux caractéristiques du filtre.

Il est possible de définir un nouveau filtre au moyen du menu contextuel, en sélectionnant la commande 'New Filter'. Une fenêtre de dialogue sera affichée afin de définir les caractéristiques du filtre. Une fois défini, le filtre sera ajouté dans l'arbre de navigation.



Propriétés des filtres:

Les symboles peuvent être filtrés en fonction de toutes les entrées présentes dans la grille des symboles. Chaque nouveau filtre est caractérisé par son nom et ses propriétés et sera automatiquement sauvegardé dans l'arbre de navigation. Les propriétés d'un filtre existant peuvent être modifiées à partir du menu contextuel, avec la commande 'Properties'.



5.2.2 Les éléments d'un symbole

Les symboles sont définis dans les fichiers de programme. Lors de l'ouverture d'un fichier programme, les symboles seront automatiquement affichés dans la grille des symboles. La grille des symboles associée au fichier du programme peut être consultée en sélectionnant l'onglet comportant le nom du fichier programme. Les symboles peuvent être édités à partir de cette grille.

Nom du symbole, d'une longueur maximale de 80 caractères

Type du symbole: spécifie le type de ressource, par exemple registre, indicateur, entrée, sortie, ,

Commentaire: Permet de définir des informations supplémentaires associées au symbole.

Symbol Name	Type	Address/Value	Comment	Actual Value	Tags	Scope
Oil_Pump.src	ROOT					
Normal_Run	I	3	Machine is in normal run	0		Public
Cond_Run	I	4	Machine is in conditiona...	0		Public
OilHigh	I	5	Oil Level is too high	0		Local
Emergency	I	7	Emergency Stop	0		Public
Intermediate_Flag	F			0		Local
Oil_Pump	O	20	Oil Pump	0		Local
OilPumpPrg	COB	2	Program For Oil Pump	0		Local

Nom du fichier auquel appartient les symboles ou nom du filtre.

Adresse/Valeur: permet de spécifier l'adresse de la ressource. La spécification d'une adresse est obligatoire pour les entrées/sorties et est facultative pour les autres types (registres, indicateurs, ...).

Scope ou contexte du symbole: les symboles locaux - 'Local' - peuvent être utilisés uniquement dans le fichier dans lequel ils sont définis, les symboles publics - 'Public' - peuvent être utilisés dans tous les fichiers du CPU actuelle. Les symboles externes - 'External' - sont des symboles publics, définis dans d'autres fichiers, mais utilisés dans le fichier courant.

Nom de symbole

Spécifie le nom de la ressource. Le premier caractère du nom de symbole est toujours une lettre, suivie par d'autres lettres, chiffres ou d'un ou de plusieurs caractères 'souligné' - '_'. Les symboles ne sont pas 'case sensitive' à moins qu'ils ne contiennent des caractères accentués. 'MotorOn' est identique à 'MOTORON', mais 'GRÜN' est différent de 'Grün'.

Les mots réservés ne peuvent pas être utilisés comme nom de symbole. La liste des mots réservés est définie en fin de ce chapitre.

Type

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

Adresse

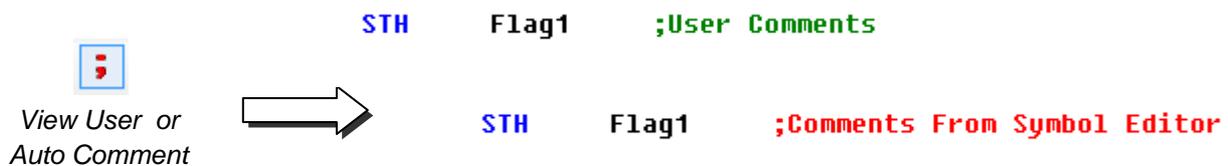
Définit l'adresse absolue de la ressource dans le programme du PCD. Chaque type de ressources – indicateur, registre, entrée, sortie, etc. comporte une plage d'adresse spécifique. Si l'adresse de la ressource n'est pas spécifiée, elle sera attribuée automatiquement lors de la construction du programme - le 'build'. C'est ce qu'on appelle l'allocation automatique. Il est obligatoire d'entrer l'adresse pour les ressources du type entrée ou sortie, car elles sont directement couplées au hardware. Chaque type de ressources compte de nombreuses adresses disponibles:

Entrées et sorties: selon les modules d'entrée/sortie insérés dans l'automate
Indicateurs: F 0, ..., F 16383
Registres: R 0, ..., R 16383
Temporisateurs, Compteurs: T/C 0, ..., T/C 1599

Commentaire

Le commentaire – 'comment' - est lié au symbole et permet de définir une information supplémentaire associée au symbole. Dans l'éditeur IL, les commentaires associés aux symboles peuvent être affichés en lieu et place du commentaire lié à la ligne de code du programme.

Basculer avec le bouton 'View User' ou 'Auto Comment'.



Adresse actuelle

La colonne des adresses actuelles – 'Actual Address' indique la valeur de l'adresse dynamique attribuée par le 'build' dans le cas où l'adresse de la ressource n'a pas été spécifiée par l'utilisateur. Cette colonne est en lecture seule.

Tags

Les tags peuvent être utilisés pour associer des informations complémentaires aux symboles. Ces informations peuvent être utilisées pour marquer des symboles appartenant au même réseau de communication, pour marquer des symboles qui seront utilisés dans un projet web ou HMI, pour marquer des symboles utilisés pour la supervision, etc. A l'aide de la fonction de filtrage de l'éditeur de symboles, les symboles marqués pourront être affichés dans une même vue.

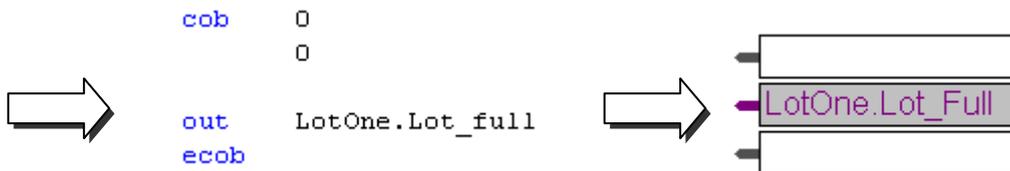
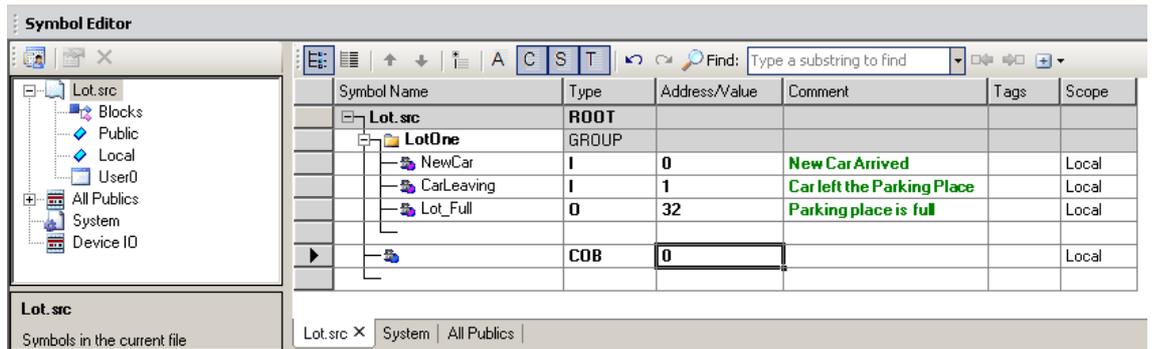
Scope

Le scope spécifie le champ d'application du symbole, dans les différents éléments du programme. Les scopes 'Local', 'Public' ou 'External' peuvent être spécifiés dans la liste déroulante. La signification des différents scopes sera donnée plus loin dans ce chapitre.

5.2.3 Regroupement des symboles

Les symboles peuvent être groupés afin de mettre ensemble les symboles correspondants à une même fonctionnalité. Cela rendra le programme plus facile à lire et à comprendre. Il suffit d'utiliser la commande 'New Group' dans le menu contextuel ou les touches "CTRL + G" pour ajouter un nouveau groupe dans l'éditeur de symboles, puis de définir ou glisser-déposer ('drag and drop') les symboles appartenants à ce groupe. Les noms de groupe peuvent être imbriqués jusqu'à un maximum de 10 niveaux, par exemple : 'Group1.Group2.Group3.Symbol0'.

Exemple: Le Group nommé 'LotOne' contient plusieurs symboles:

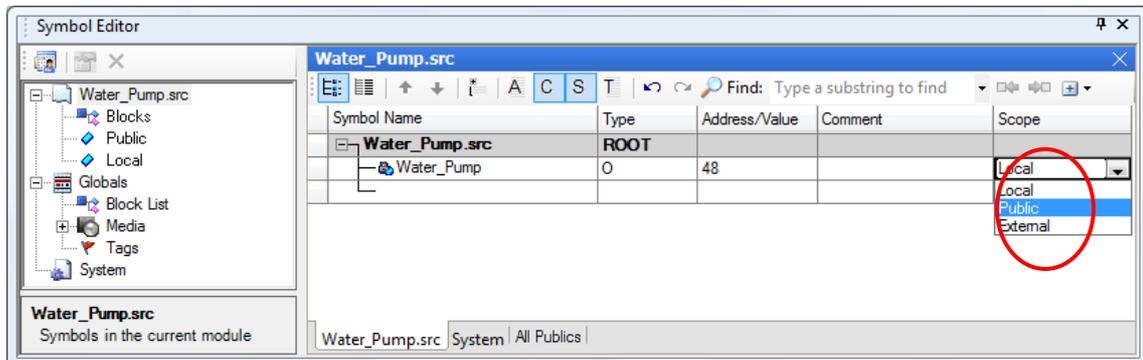


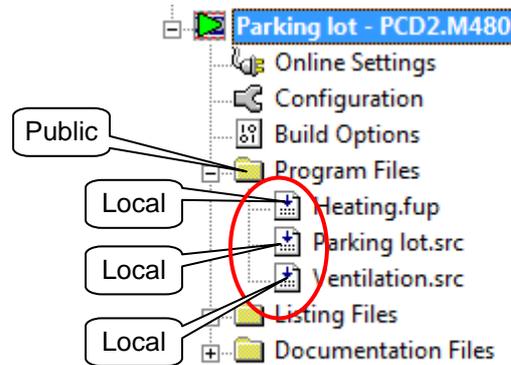
Dans le programme le nom du groupe 'LotOne' précède le nom du symbole 'Lot_full' et les deux sont séparés par un point.

Dans certains programmes, vous pouvez avoir de nombreux symboles avec des noms similaires, en particulier si vous avez à implémenter la même fonctionnalité plusieurs fois. Au lieu de différencier les symboles par leur nom, il est plus judicieux d'utiliser les mêmes noms de symboles, mais en les mettant dans des groupes différents. Par exemple : 'MotorA.Symbol1', 'MotorB.Symbol1' au lieu de 'Motor_A_Symbol1', 'Motor_B_Symbol1'.

5.2.4 Champs d'applications des symboles - scope

Le champ d'application d'un symbole peut être spécifié dans la colonne 'Scope'. Il existe trois types de scopes possibles: 'Local', 'Public' et 'External'.





Symboles locaux

Les symboles locaux - 'Local' - peuvent être utilisés uniquement dans le fichier dans lequel ils ont été définis. Par exemple, un symbole avec un champ d'application local, définie dans le fichier 'Parking_Lot.src' ne peut être utilisé que dans ce fichier. Ce symbole ne sera pas connu dans les autres fichiers sources.

Symboles publics

Les symboles publics – 'Public' - peuvent être utilisées dans n'importe quel fichier source d'un même contrôleur. Par exemple, les symboles publics définis dans le fichier 'Parking_Lot.src' pourront être utilisés dans le fichier 'Heating.fup' ou 'Ventilation.src' puisque tous ces fichiers sont présents dans le même contrôleur 'Parking lot'. Les symboles publics sont aussi appelés symboles globaux, ce qui veut dire que leur champ d'application est global, au niveau du contrôleur.

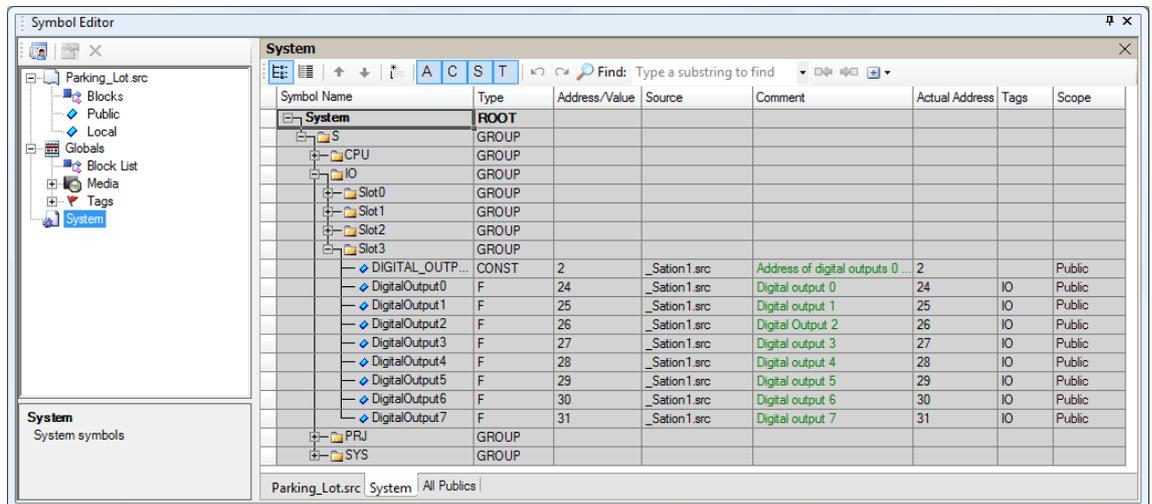
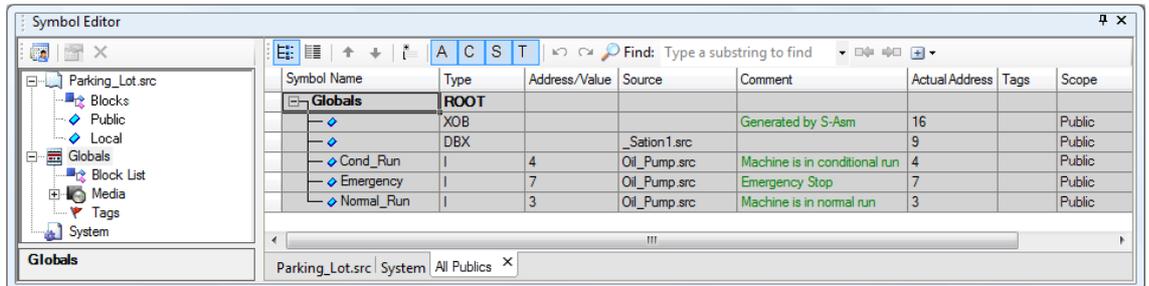
Symboles externes

Les symboles externes – 'External' – désignent une référence à un symbole public défini dans un autre fichier de programme. Cette définition indique que le symbole utilisé est valide dans le fichier source courant, mais que sa définition se trouve dans un autre fichier source. Lorsque qu'un symbole public est sélectionné dans la grille 'All Publics' de l'éditeur de symboles et ensuite utilisé, avec un 'drag and drop' vers le programme courant, la définition du symbole externe sera automatiquement ajoutée au fichier actuel. (Vous trouverez plus d'information sur l'utilisation des symboles publics dans la section 'Travailler avec les symboles').

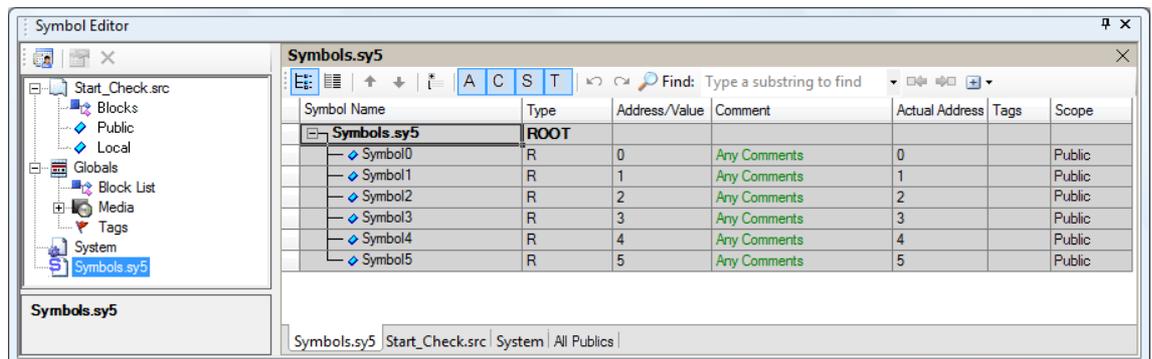
5.2.5 Affichage des filtres

Il est possible de naviguer à travers les filtres définis dans l'arbre de navigation et d'ouvrir le résultat de chaque filtre dans un nouvel onglet. Les copies d'écran suivantes sont des exemples de quelques filtres prédéfinis. Une grille avec un fond grisé indique que les symboles sont en lecture seule. La définition des symboles peut être changée uniquement dans le fichier source dans lequel le symbole est défini.

Le filtre prédéfini 'All Publics' affiche tous les symboles publics présents dans le contrôleur.



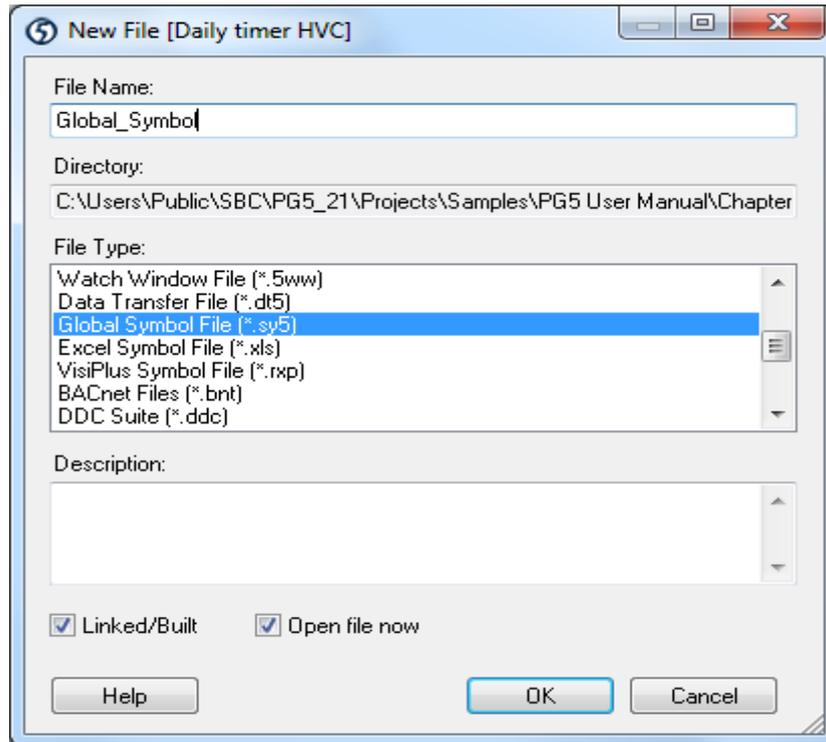
Si le contrôleur contient un fichier de symboles avec l'extension 'sy5', le filtre avec le nom de fichier apparaîtra dans l'arborescence de navigation, et tous les symboles définis dans le fichier 'sy5' seront affichés dans la grille.



5.2.6 Définitions de symboles dans un fichier 'sy5'

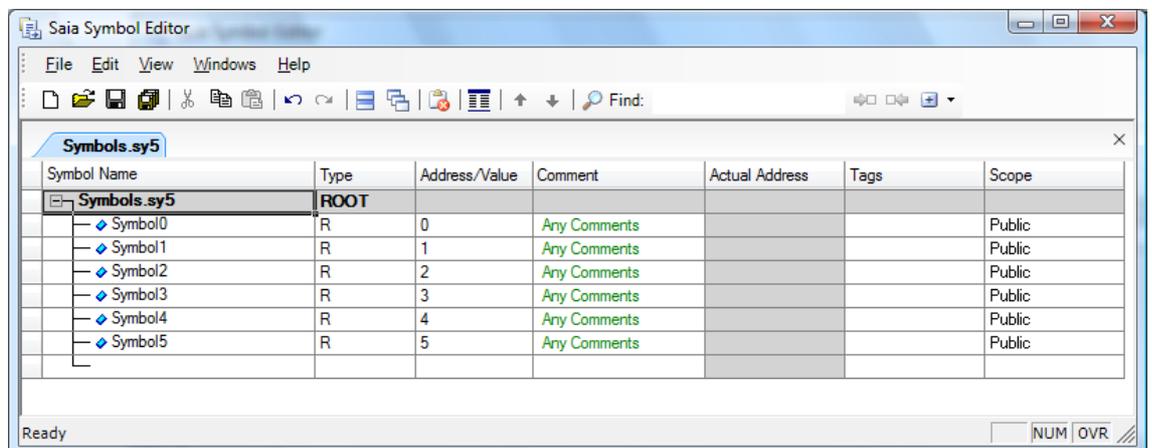
Comme nous l'avons vu dans les sections précédentes, les symboles sont directement définis dans les fichiers de programme (IL, Fupla, etc.).

Lorsque les symboles publics sont définies dans les fichiers du programme (IL, Fupla, etc.), ils ne sont pas perdus lorsque les fichiers du programme sont copiés d'un contrôleur à un autre. Il est donc plus avantageux de définir les symboles dans les fichiers de programme respectifs au lieu de les définir dans un seul fichier de symboles 'sy5'. Il est tout de même possible de définir un nouveau fichier de symboles 'sy5' et de l'ajouter au projet en sélectionnant le type de fichier 'sy5' dans le gestionnaire de projet 'Saia PG5 Program Manager'. Ce fichier de symboles sera ouvert et pourra être modifié au moyen de l'éditeur de symboles.



Fichier 'sy5' ouvert avec l'éditeur de symboles

Ce fichier de définition des symboles publics 'sy5' est ouvert dans l'éditeur de symboles, ce qui permet la modification des symboles existants et la création de nouveaux symboles.



Symboles globaux définis dans un projet PG5 Version 1.4 et plus ancien

Lorsqu'un projet PG5 1.4 est restauré dans la version PG5 2.0, la définition des symboles globaux du PG5 1.4 est sauvegardée dans le fichier 'Globals.sy5' et les symboles pourront être visualisés dans l'éditeur de symboles, comme symboles publics.

5.2.7 Définition des symboles dans des fichiers 'xls', 'txt', 'rxp'

La définition des symboles peut être faite dans des fichiers de programme (Fupla, IL), dans des fichiers de symboles 'sy5' mais aussi dans des fichiers d'autres applications comme Excel - 'xls', texte - 'txt', etc. Ces fichiers peuvent être ajoutés aux fichiers programme du contrôleur. Après un 'build', les symboles définis dans ces fichiers seront visible dans la grille 'All Publics' et pourront être utilisés dans tous les fichiers programme du contrôleur. De cette façon, il n'est plus nécessaire d'importer et/ou de fusionner des fichiers externes de symboles avec les symboles du programme. Un import des symboles provenant d'autres types de fichiers n'est plus nécessaire.

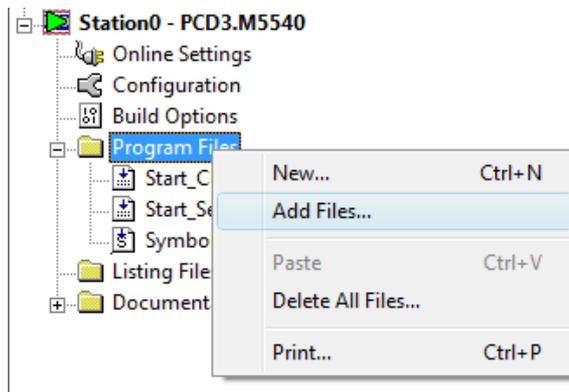
Toutefois, pour importer des symboles provenant d'autres fichiers, il est toujours possible de copier des symboles à partir des fichiers originaux tels que 'xls', 'txt', 'rxp', 'sy5' et de les coller dans la grille des symboles (plus d'information à la section 5.3.16).

Lorsqu'un fichier de symboles est défini, il peut être ajouté au projet à l'aide du menu contextuel du gestionnaire de projet, dans le dossier 'Program Files' en sélectionnant le menu 'Add Files...'. Il est également possible de créer un nouveau fichier de symboles ('xls', 'txt', etc.) avec le menu de contexte "New..." sur le dossier 'Program Files' et en sélectionnant le type de fichier approprié dans la boîte de dialogue 'New file'.

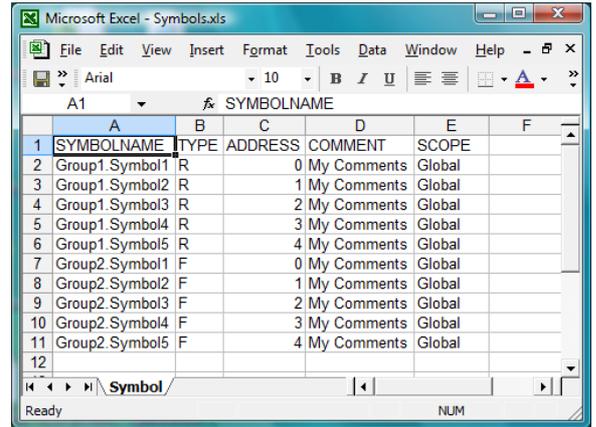
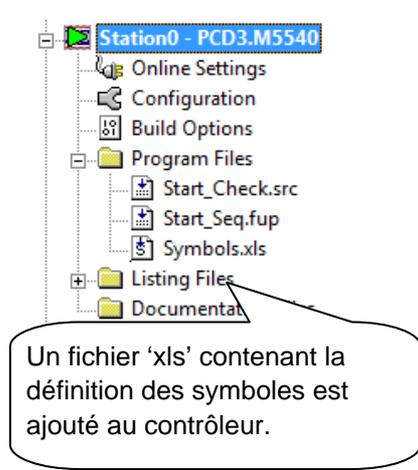
Note:

Les fichiers de symboles publics avec une extension 'sy5', 'xls' et 'rxp' sont édités avec le programme correspondant à leur extension et sauvegardées dans le format original.

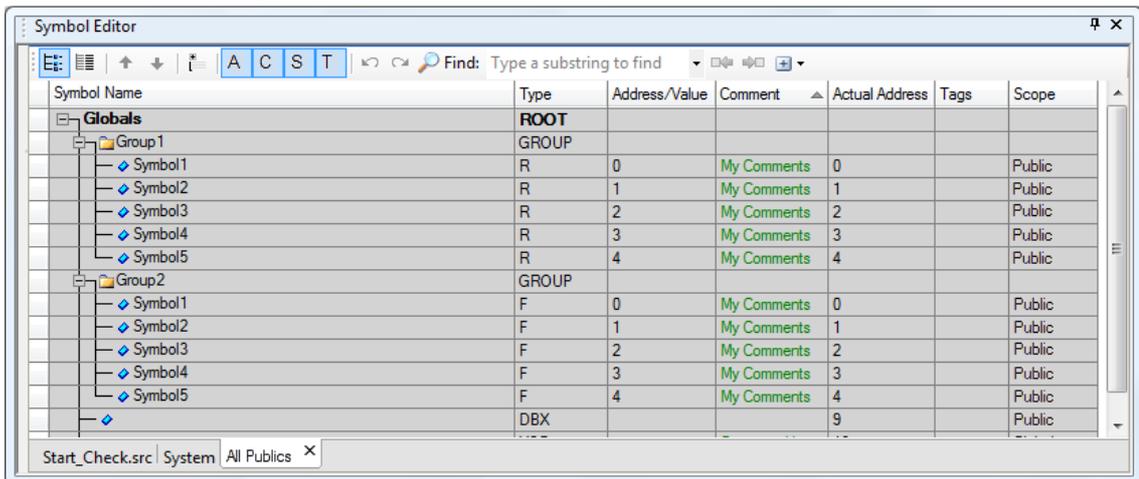
Ajouter un fichier de définition de symboles



Ajouter un fichier de définitions Excel (*.xls').



Après le 'build' du programme, les symboles publics définis dans le fichier Excel sont disponibles dans le projet sous la rubrique 'All Publics' et peuvent être utilisés dans les fichiers programme.



5.2.8 Définir des symboles pour les réseaux de communications

Le partage de données entre deux différents contrôleurs n'est pas aussi simple que le partage d'informations entre les fichiers programme d'un même contrôleur. Une connexion réseau est nécessaire entre les deux contrôleurs. Cette connexion réseau peut être configurée dans notre éditeur de réseau qui supporte SBus, Profibus DP et les réseaux LON. Les symboles Réseau peuvent être utilisés dans les programmes de transfert de données d'un contrôleur à l'autre. Ces symboles apparaissent comme des symboles publics.

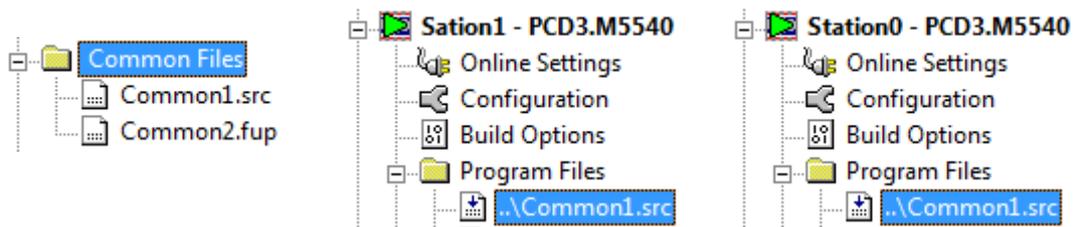
5.2.9 Définition des symboles dans un fichiers commun

Lorsqu'un même fichier est utilisé dans plusieurs contrôleurs, il est ajouté dans le dossier '*Common Files*' du gestionnaire de projet. Ces fichiers peuvent être des fichiers de programme comme '*src*', '*fup*', '*sfc*' ou des fichiers des symboles publics, comme '*sy5*', '*xls*', '*txt*', '*rxp*' etc.

Les symboles publics définis dans ces fichiers communs peuvent être partagés par plusieurs contrôleurs. Une fois le fichier commun référencé dans les fichiers programme du contrôleur courant, tous les symboles publics définis dans le fichier commun sont accessibles dans le contrôleur courant.

Exemple:

'*Common1.src*' est référencé dans les deux contrôleurs '*Station0*' et '*Station1*'. Tous les symboles publics définis dans '*Common1.src*' seront disponibles dans '*Station0*' et '*Station1*'.



5.3 Travailler avec les symboles

5.3.1 Ajouter des symboles à la liste des symboles

Méthode Simple

Ouvrez le fichier programme à modifier. Comme l'éditeur de symboles fait partie de chaque éditeur de programme (Fupla, IL, Graftec), il sera directement disponible. L'éditeur de symboles a toujours ligne vide en fin de grille, afin de faciliter l'entrée de nouveaux symboles. Pour ajouter un nouveau symbole, commencer par entrer le nom du symbole dans la colonne 'Symbol Name', puis à l'aide du tabulateur, se déplacer sur la colonne 'Type' pour sélectionner le type de symbole, entrez ensuite l'adresse et/ou la valeur initiale – 'Address/Value' - le commentaire – 'Comment' - et le 'Scope' du nouveau symbole. Une fois la définition du symbole complète, la valider avec la touche 'Enter'. Une nouvelle ligne est ajoutée automatiquement, permettant ainsi d'entrer le symbole suivant.

Méthode Rapide

Il est également possible d'entrer toute la définition du symbole sous la colonne 'Symbol Name'. Voir l'exemple ci-dessous.

The screenshot shows the Symbol Editor window with a table of symbols. The table has columns: Symbol Name, Type, Address/Value, Comment, Actual Address, Tags, and Scope. The following table represents the data shown in the screenshot:

Symbol Name	Type	Address/Value	Comment	Actual Address	Tags	Scope
Oil_Pump.src	ROOT					
Normal_Run	I	3	Machine is in normal run	3		Public
Cond_Run	I	4	Machine is in conditional run	4		Public
Oil_High	I	5	Oil Level is too high			Local
Emergency	I	7	Emergency Stop	7		Public
Intermediate_Flag	F					Local
Oil_Pump	O	20	Oil Pump			Local
OilPumpPrg	COB	2	Program For Oil Pump			Local
Waste_Pump 0 32;Waste Pump						Local

A yellow arrow points to the 'Symbol Name' column of the last row, where the text 'Waste_Pump 0 32;Waste Pump' is entered. Below the main table, a smaller table shows a zoomed-in view of the 'Waste_Pump' entry, highlighting the 'Local' scope dropdown menu.

La syntaxe pour entrer la définition d'un symbole est la suivante :

'Nom_de_Symbole type adresse ; commentaire'

Après avoir introduit la définition du symbole selon la syntaxe donnée ci-dessus, appuyer sur la touche 'Enter' du clavier; les informations seront placées automatiquement dans les champs appropriés.

Saisie partielle de symboles

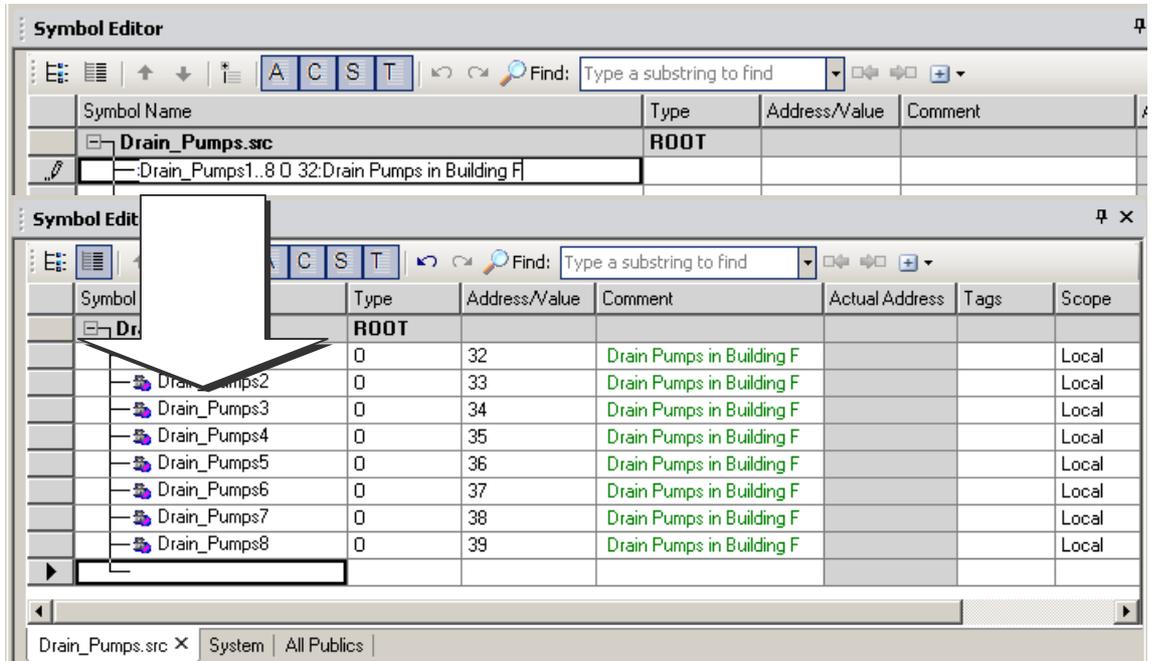
Il est possible de sauvegarder une définition partielle de symbole, ce qui signifie par exemple, que vous pouvez entrer le nom et le type, cesser de travailler sur l'éditeur de symboles et entrer l'adresse, le commentaire et les tags par la suite. Cela permet en tout temps d'interrompre votre travail sur l'éditeur de symboles et d'enregistrer le statut intermédiaire.

5.3.2 Ajout de plusieurs symboles

Il est possible d'entrer directement une liste de symboles à partir de la cellule 'Symbol Name'. Il suffit d'entrer le nom du symbole avec le premier et dernière numéro de l'élément, comme ci-dessous:

'Drain_Pumps1..8 O 32; pompes dans le bâtiment F.

'8' est le nombre de symboles à définir, 'O' est le type, et '32' est l'adresse de début de la plage des adresses. Appuyez sur 'Enter' et l'éditeur de symboles complétera la liste automatiquement. Jusqu'à 100 symboles peuvent être ajoutés en une fois.



5.3.3 Symboles référencés

Il est possible de définir un symbole avec, comme adresse, la référence à un autre symbole. Cette fonctionnalité est particulièrement utile lors de la définition d'un tableau de symboles – 'array' - où les adresses des symboles seront référencées à l'adresse de base du tableau, ce qui veut dire que l'adresse du symbole est définie par un autre symbole de la grille. Un tableau de symbole est défini par sans grandeur, placée entre les caractères '[' et ']'.

Par exemple :

Un tableau de trois registres, à partir d'une adresse dynamique est défini par 'Mes3Registres R [3] ; mon tableau de 3 registres'.

Un tableau de 4 registres, à partir d'une adresse fixe de 100 est défini par 'Mes4Registres R 100[4] ; mon tableau de 4 registres'.

L'utilisation de référence est utile dans le cas d'un déplacement d'une carte d'entrée/sortie d'un slot à un autre. Cela nécessitera uniquement le changement de l'adresse de base du tableau. Tous les autres symboles utilisant la référence seront actualisés automatiquement. Dans l'exemple suivant les symboles de référence 'Drain_Pumps2' à 'Drain_Pumps8' ont l'adresse de référence de premier symbole 'Drain_Pumps1'.

Symbol Name	Type	Address/Value	Comment	Actual Address	Tags	Scope
Drain_Pumps1	0	32[8]	Drain Pumps in buildnig F			Local
Drain_Pumps2	0	Drain_Pumps1+1	Drain Pumps in buildnig F			Local
Drain_Pumps3	0	Drain_Pumps1+2	Drain Pumps in buildnig F			Local
Drain_Pumps4	0	Drain_Pumps1+3	Drain Pumps in buildnig F			Local
Drain_Pumps5	0	Drain_Pumps1+4	Drain Pumps in buildnig F			Local
Drain_Pumps6	0	Drain_Pumps1+5	Drain Pumps in buildnig F			Local
Drain_Pumps7	0	Drain_Pumps1+6	Drain Pumps in buildnig F			Local
Drain_Pumps8	0	Drain_Pumps1+7	Drain Pumps in buildnig F			Local

5.3.4 Ajout de symboles en travaillant avec l'éditeur IL

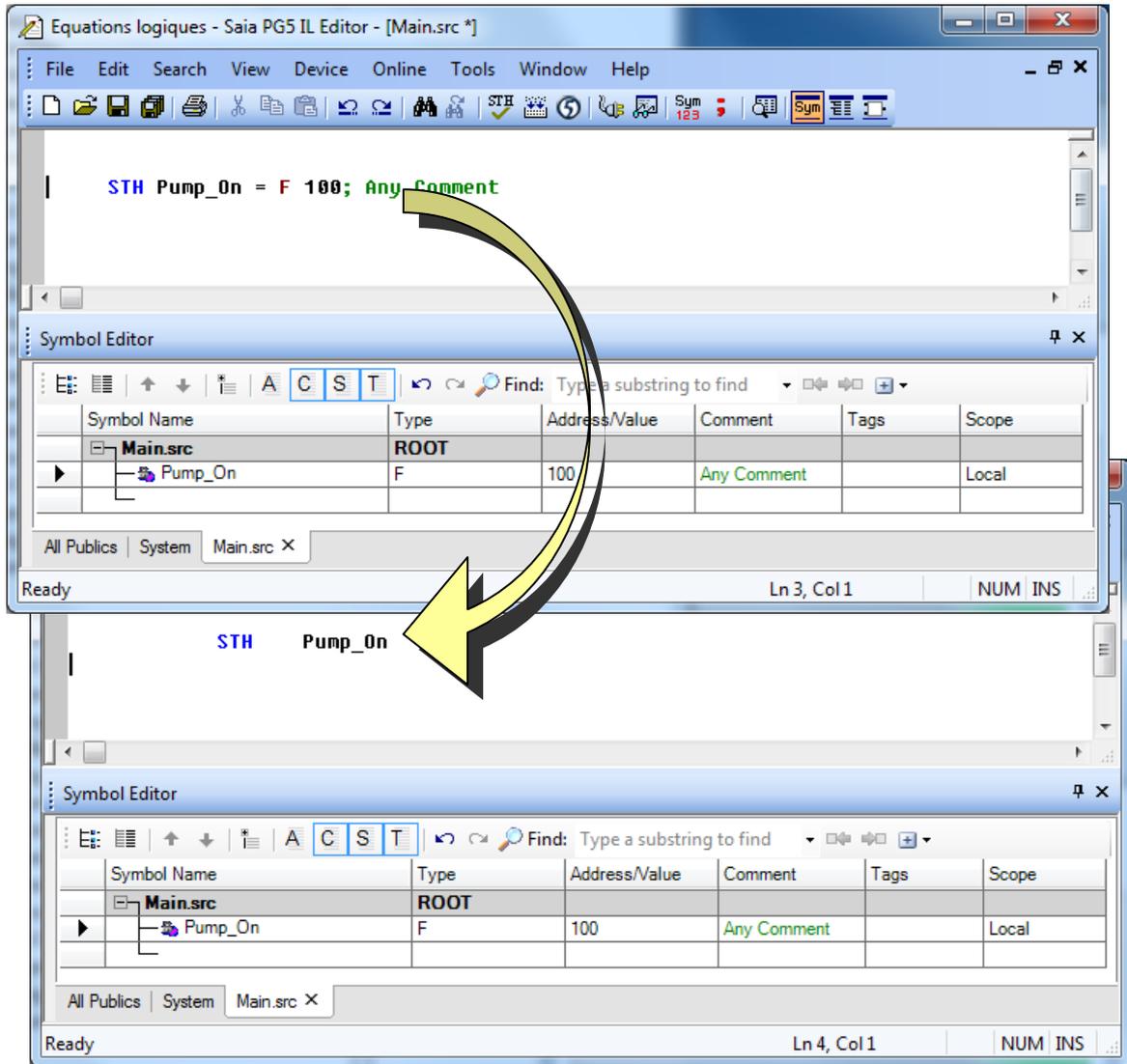
De nouveaux symboles peuvent être ajoutés lors de l'édition du programme en IL, en introduisant une ligne de code du programme avec le mnémonique et son opérande. Pour ce faire, il suffit d'écrire l'opérande, le nom du symbole suivi de sa définition avec la syntaxe suivante : 'Opérande Nom_de_Symbole= type adresse ; commentaire'.

En appuyant sur la touche 'Enter' du clavier, les informations vont se placer automatiquement dans les champs appropriés de l'éditeur de symboles, à condition que la définition soit correcte et que l'option 'Ajouter automatiquement les entrées de type ou de la valeur à l'option de la table des symboles' – 'Automatically add types/values to Symbol Editor while typing' - soit sélectionnée (commande 'Options' dans le menu 'Tools' de l'éditeur IL).

Note:

Tout nouveau symbole défini directement à partir de l'éditeur IL sera ajouté en tant que symbole local. Si nécessaire, le scope pourra être changée par la suite dans l'éditeur de symboles.

Exemple:



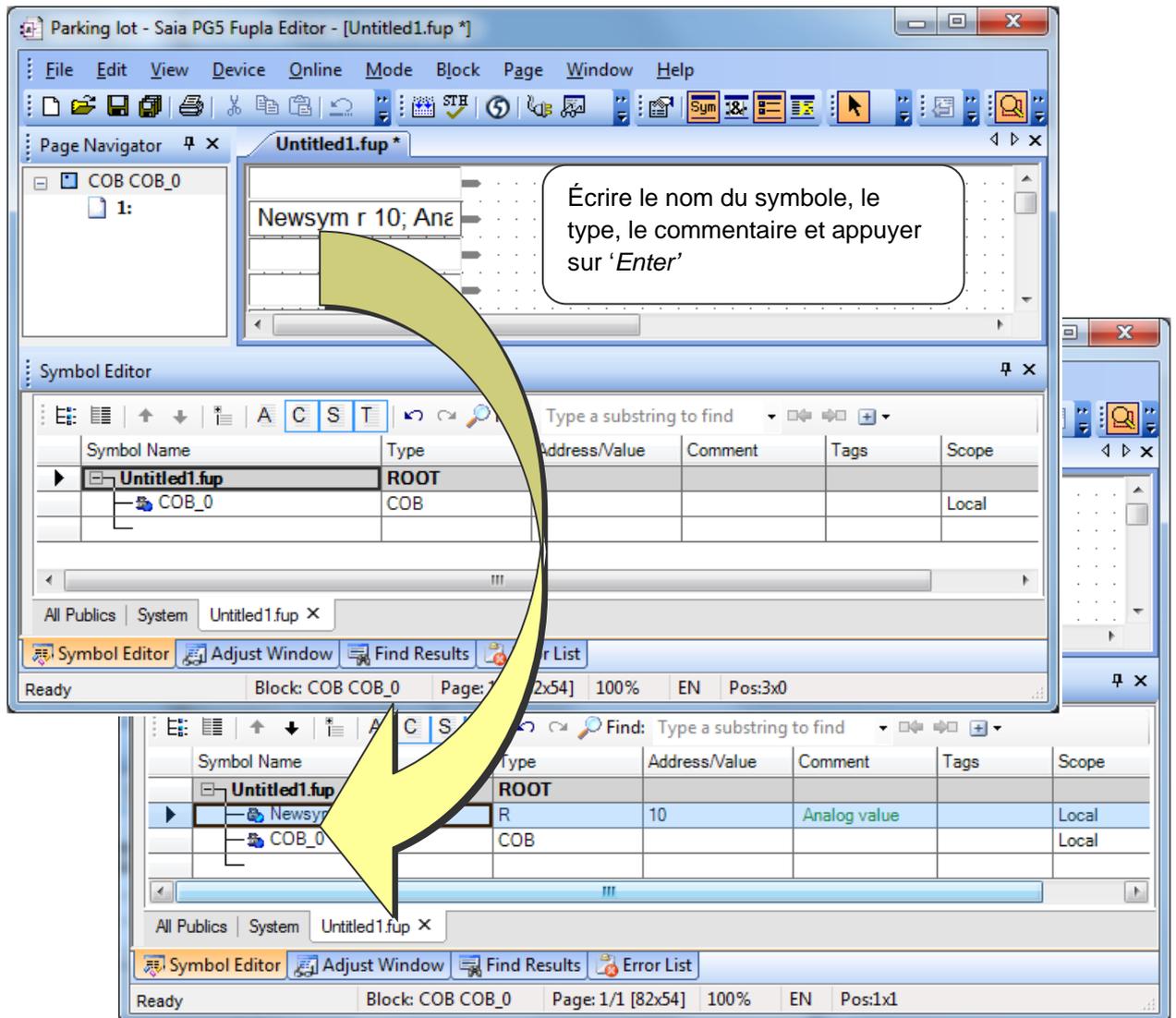
5.3.5 Ajout de symboles en travaillant avec l'éditeur Fupla

L'éditeur Fupla fonctionne exactement de la même façon que l'éditeur IL. Vous pouvez entrer des nouveaux symboles directement dans les champs d'entrée/sortie – connecteurs' - d'une page Fupla, avec la syntaxe suivante:

'Nom_de_Symbole type adresse ; commentaire'

Note:

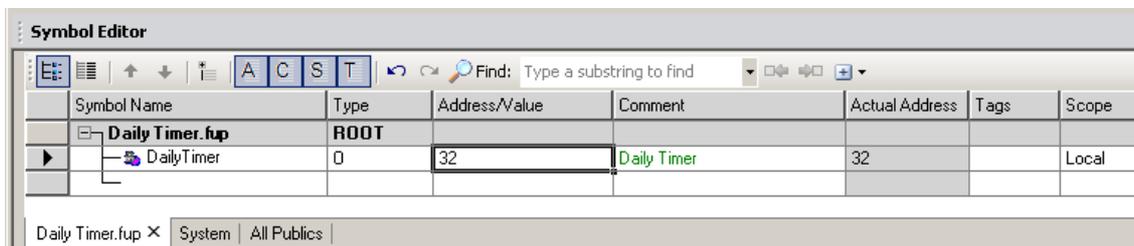
Tout nouveau symbole défini directement à partir de l'éditeur Fupla sera ajouté en tant que symbole local. Si nécessaire, le scope pourra être changé par la suite dans l'éditeur de symboles.



5.3.6 Modes d'adressage des symboles

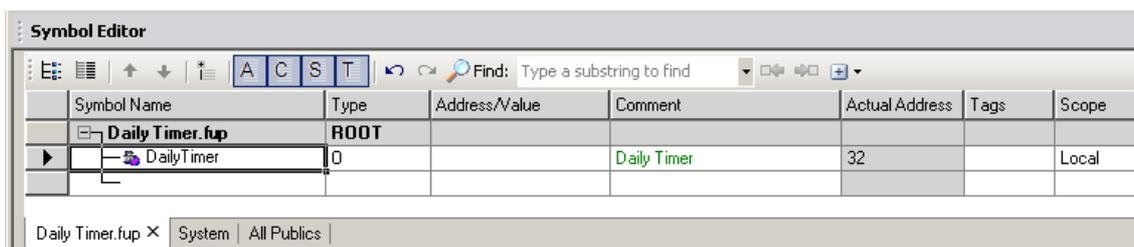
La définition d'un symbole ne comprend pas nécessairement toutes les informations présentées dans la grille. Nous pouvons distinguer trois types d'adressage:

Adressage complet avec noms de symbole



Une ressource est définie avec un nom de symbole, son type, son adresse et un commentaire. Le commentaire est optionnel mais utile pour faciliter la compréhension du programme. Une correction du nom de symbole, du type ou de l'adresse sera prise en charge dans l'éditeur de symboles et chaque connecteur dans un programme Fupla et chaque ligne dans un programme IL comportant le symbole modifié seront automatiquement mis à jour.

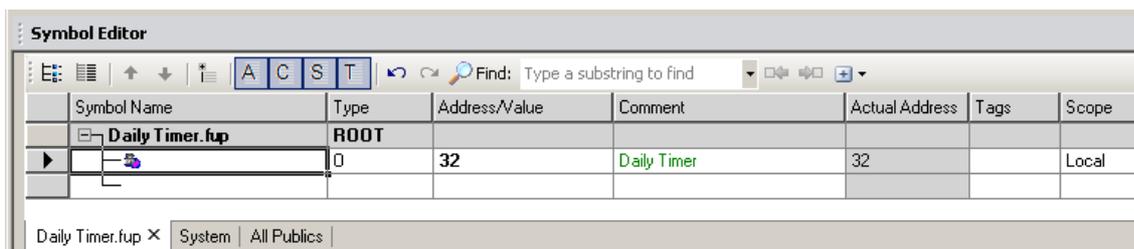
Adressage dynamique



Il s'agit d'une forme d'adressage symbolique dans lequel l'adresse n'est pas spécifiée par l'utilisateur. L'adresse est alors attribuée automatiquement au cours de la construction de programme - le 'build'. L'adresse attribuée est choisie dans une plage d'adresses définie dans les options de compilation (voir gestionnaire de projet). Après la construction – le build, l'adresse attribuée sera affichée dans la colonne 'Actual Address'.

L'adressage dynamique est disponible pour les indicateurs, les registres, les compteurs, les temporisateurs, les textes, les DB, les COB, les PB, les FB et les SB. Par contre, pour les entrées, pour les sorties et pour les XOB, l'adresse absolue doit toujours être spécifiée.

L'adressage absolu



Une ressource est définie uniquement avec son type et son adresse et un commentaire (optionnel). L'utilisation d'un adressage absolu directement dans le programme est un

inconvenient lorsque vous changez le type ou l'adresse. Le programme utilisateur ne sera pas mis à jour suite à une modification faite dans l'éditeur de symboles. Les changements doivent être faits manuellement dans les connecteurs de Fupla et/ou les lignes du programme IL. Il est donc préférable d'utiliser les noms de symboles, avec en option l'adressage dynamique.

5.3.7 Utilisation des symboles dans un programme

Au cours de l'édition d'un programme, les symboles déjà définis dans l'éditeur de symboles peuvent être utilisés de différentes manières. Toutes les méthodes décrites ci-dessous peuvent être utilisées dans l'éditeur IL et l'éditeur FUPLA.

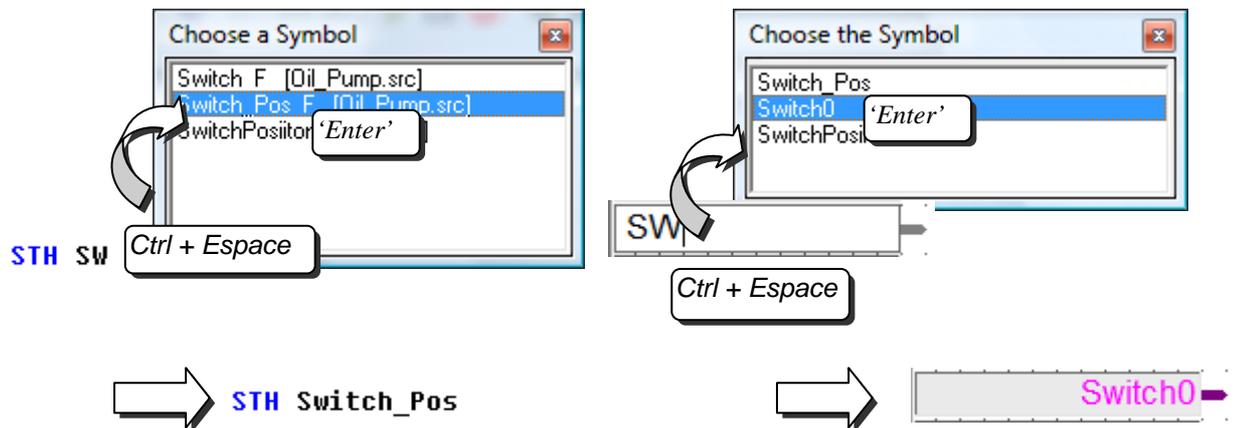
Entrer un symbole à partir du clavier

Le nom du symbole est inscrit dans son intégralité à partir du clavier pour chaque instruction qui l'utilise. Cette méthode peut engendrer des erreurs de frappe qui auront pour conséquences de générer des erreurs dans le programme.

Entrer un symbole à partir de la recherche sélective

Si vous utilisez de longs noms de symboles, votre programme sera plus facile à lire. Toutefois, il est fastidieux d'entrer le nom du symbole à chaque fois qu'il est utilisé. Cette action peut être évitée simplement en entrant les premiers caractères du symbole et en appuyant sur les touches 'Ctrl' + 'Espace' pour rechercher tous les symboles commençant par ces caractères. Le symbole recherché peut alors être sélectionné avec la souris ou avec les touches 'flèches' du clavier (↓, ↑) et confirmé en appuyant sur 'Enter'.

Exemple:



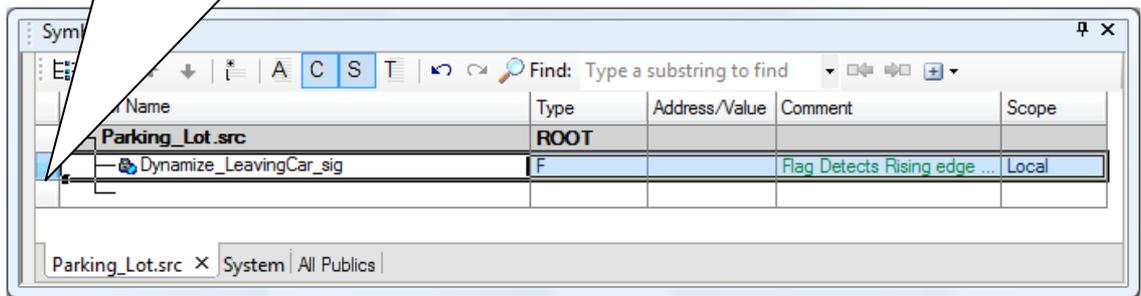
Glisser/déposer un symbole local

Une fois un symbole défini dans l'éditeur de symboles, il est possible de le glisser dans l'éditeur de programme – 'drag & drop' - et de l'utiliser sans avoir à entrer à nouveau son nom. De cette façon, toute erreur de frappe est évitée.

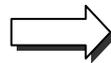
Pour effectuer cette manipulation, il faut positionner le curseur de la souris sur la première cellule de la ligne de définition du symbole, appuyer sur le bouton gauche de la souris et le maintenir, puis faire glisser le curseur de la souris dans l'éditeur de programme et lâcher le bouton de la souris. Le symbole est alors automatiquement ajouté à l'endroit indiqué par le curseur de la souris.

Exemple 1

Positionner le curseur de la souris sur la première cellule, appuyer sur le bouton gauche de la souris et le maintenir.



sth



sth DynamiseCar_Leaving

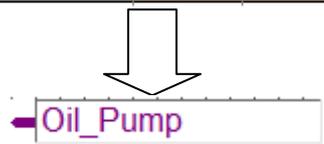
Glisser le curseur avec la souris dans l'éditeur IL

Lâcher le bouton de la souris

Exemple 2

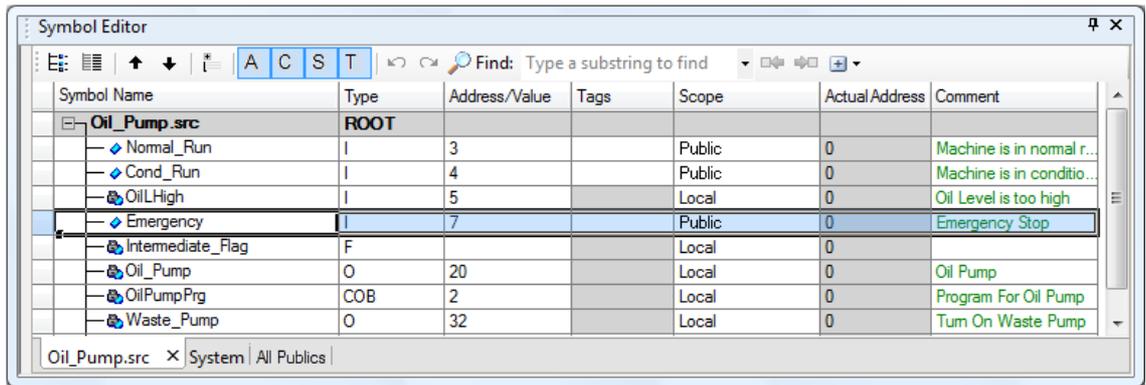
Sans relâcher le bouton de souris, glisser le symbole dans l'éditeur Fupla

Symbol Name	Type	Address/Value	Comment	Actual Address	Tags	Scope
est1.fup	ROOT					
COB_0	COB					Local
Oil_Pump	O	32	Oil pump			Local



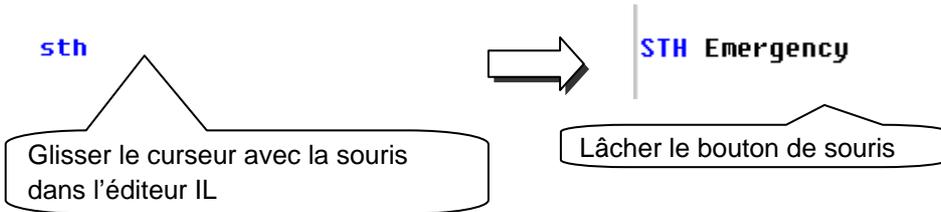
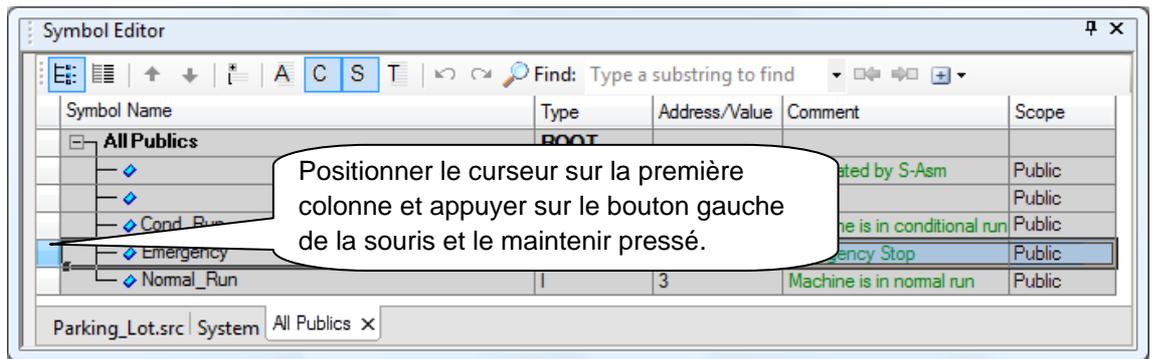
Glisser/déposer un symbole public

Le symbole public 'Emergency' est défini dans le fichier 'Oil_Pump.src'.

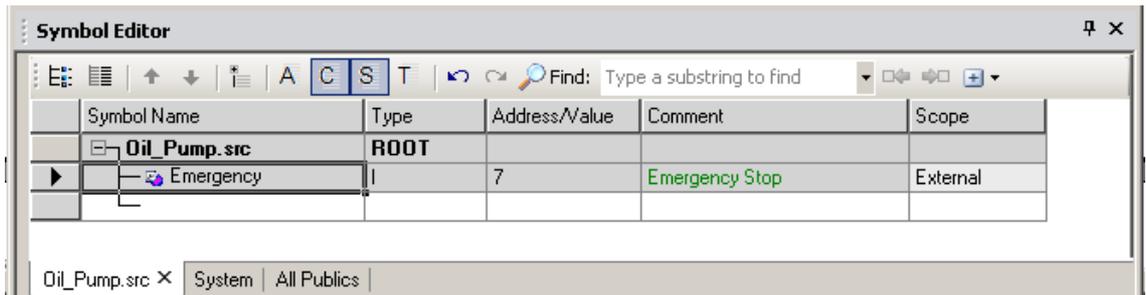


Les symboles publics définis dans le fichier 'Oil_Pump.src' apparaîtront dans la grille des symboles, sous l'onglet du filtre 'All Publics' lorsqu'un fichier source du contrôleur courant sera édité. Pour utiliser un symbole public, il suffit de le glisser et de le déposer de la grille des symboles publics, sous l'onglet 'All Publics' vers l'éditeur de programme. Ce symbole sera ensuite automatiquement ajouté dans la grille des symboles du programme actuellement édité, comme symbole externe – 'External'. Par exemple, lorsque le symbole public 'Emergency' est glissé vers l'éditeur de programme, sa référence – 'External' est automatiquement ajoutée au fichier 'Parking_Lot.src'.

Afficher les symboles publics dans 'Parking_Lot.src' :



Un symbole externe – 'External' - est automatiquement ajouté dans la grille des symboles du fichier 'Parking_Lot.src'.



Glisser/déposer plusieurs symboles

Il est possible de sélectionner plusieurs symboles et de les glisser/déposer en une fois dans l'éditeur de programme Fupla ou IL. Ceci réduit le nombre de manipulations et ainsi le nombre d'erreurs potentielles.

Ci-dessous, un exemple montrant la sélection de plusieurs symboles.

Sélectionner le premier symbole avec souris en cliquant sur la cellule de la première colonne.

Appuyer sur touche 'Shift' et sélectionner le dernier symbole.

Appuyer sur touche 'CTRL' et sélectionner un symbole particulier

Symb	name	Type	Address/Value	Comment	Actual Address	Tags	Scope
	Test.fup	ROOT					
	COB_0	COB					Local
	DrainPump1	0	32	drain pump in building F			Local
	DrainPump2	0	33	drain pump in building F			Local
	DrainPump3	0	34	drain pump in building F			Local
	DrainPump4	0					Local
	DrainPump5	0					Local
	DrainPump6	0					Local
	DrainPump7	0					Local
	DrainPump8	0					Local

Ci-dessous un exemple montrant le glissement des symboles dans l'éditeur Fupla.

Garder 'Ctrl' / 'Shift' enfoncé et pointer la souris sur la première colonne. Appuyer sur le bouton gauche de la souris et le maintenir pressé.

Sans relâcher le bouton de la souris, glisser dans l'éditeur et relâcher le bouton gauche.

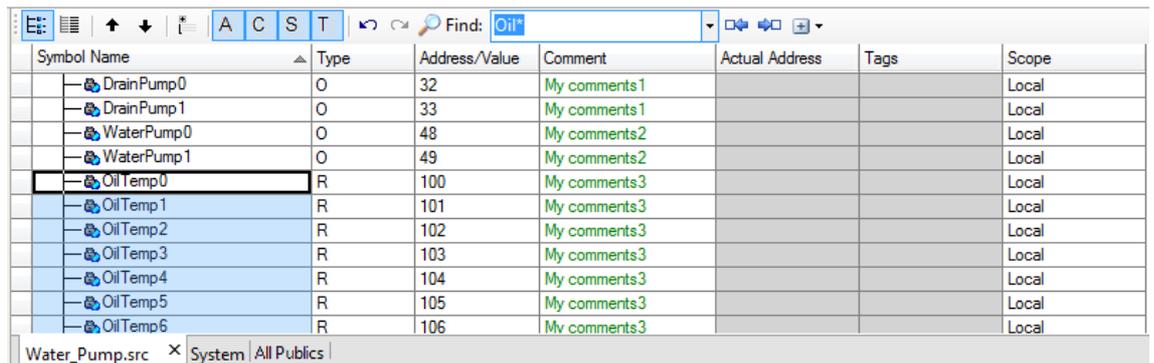
	DrainPump3	0
	DrainPump4	0
	DrainPump5	0
	DrainPump6	0
	DrainPump7	0

DrainPump1
DrainPump2
DrainPump3
DrainPump4
DrainPump7

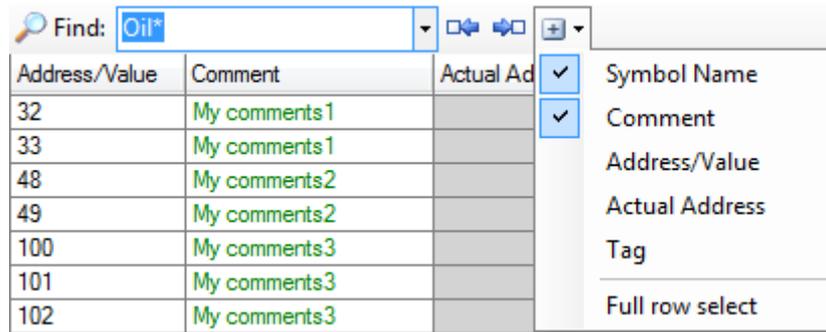
5.3.8 Recherche de symboles

Lorsqu' un grand nombre de symboles est défini dans la grille, l'outil de recherche 'Find' peut être utilisé afin de localiser un symbole particulier dans la grille. Une fois le symbole trouvé, il peut être alors glissé vers l'éditeur de programme.

La barre d'outils 'Find' permet de trouver le ou les symbole(s). Les boutons prochain - 'Next' - et précédant - 'Previous' - sont disponibles pour rechercher les symboles correspondants. Dans la boîte d'édition de la recherche, il est possible d'utiliser des caractères joker comme '*' – un ou plusieurs caractères - ou '?' – un caractère.



Dans les options avancées, il est possible de sélectionner les colonnes dans lesquelles la recherche sera effectuée.



5.3.9 Allocation automatique des adresses

Jusqu'à présent, les symboles ont été déclarés avec les éléments comme ci-dessous:

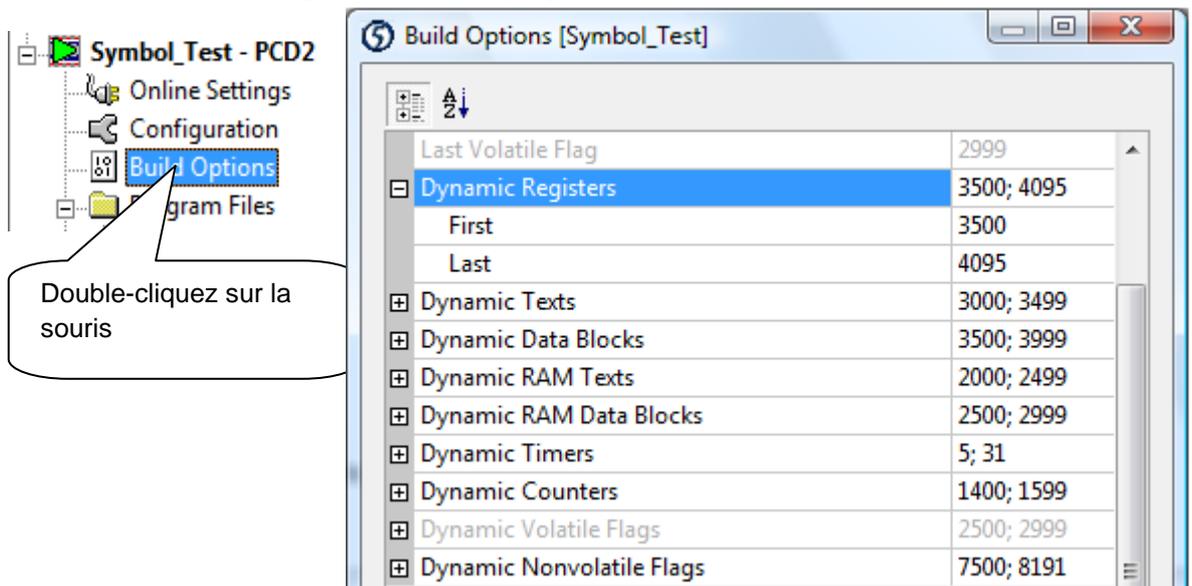
<u>Symbol name</u>	<u>Type</u>	<u>Address</u>	<u>Comment</u>
--------------------	-------------	----------------	----------------

Exemple: Pumpspeed R 2000 ;Speed in l/min

Pour des symboles autres que des entrées ou des sorties, il n'est pas nécessaire d'entrer une adresse. C'est l'application PG5 qui va attribuer une adresse à votre élément au moment de la construction – du 'build'. Cette fonctionnalité est appelée allocation automatique (ou dynamique). Lors du 'build', PG5 va chercher dans la plage d'adresses configurée dans les 'Build Options' une adresse disponible et l'attribuer à l'élément spécifié.

Exemple: Pumpspeed R ;Speed in l/min

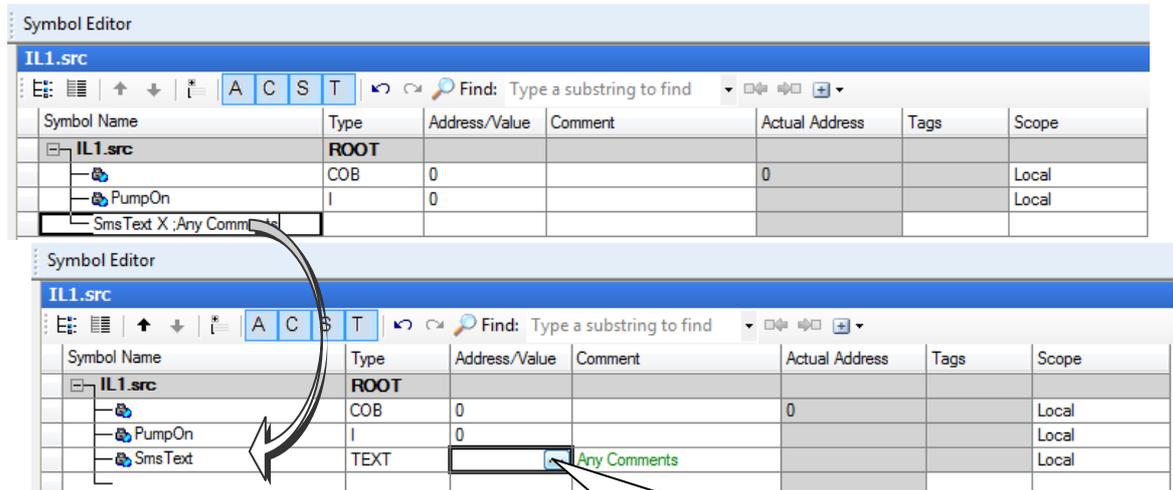
Si vous déclarez un registre dans votre programme sans lui donner une adresse:



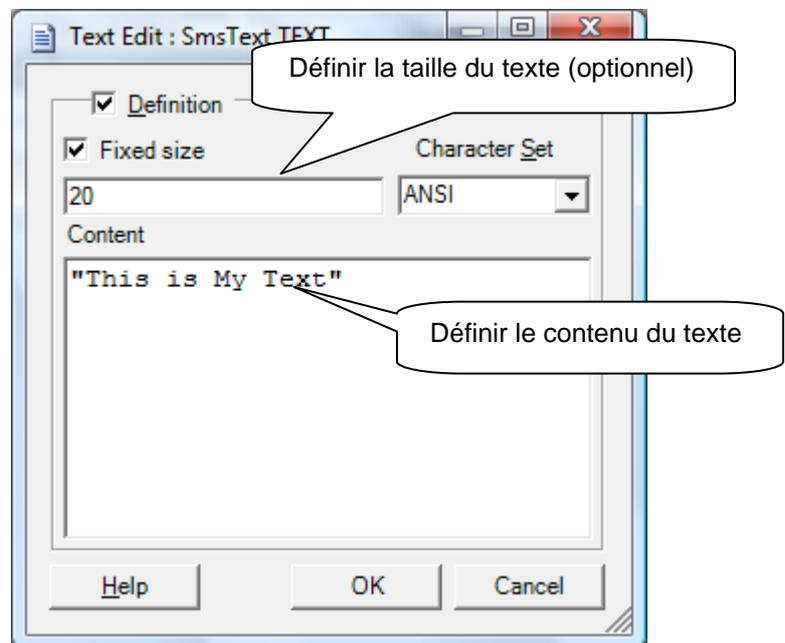
L'adresse du registre sera définie dans la plage située entre 3500 et 4095 pendant le processus de construction – le 'build'. Cette plage dynamique est définie entre 3500 et 4095 pour les registres et peut être ajustée dans les 'Build Options'. Lors du 'build', une adresse sera allouée et pourra être visible dans la colonne 'Actual Address' de l'éditeur de symboles.

5.3.10 Saisie de texte

Afin d'ajouter un texte dans le programme du PCD, celui-ci doit être déclaré au préalable. La déclaration peut être faite dans l'éditeur de symboles en sélectionnant le type 'TEXT', puis en ouvrant la fenêtre d'édition du texte ou moyen du bouton défini dans la colonne 'Address/Value' – voir l'exemple ci-dessous. Un texte peut aussi être déclaré intégralement à partir de la première cellule de la grille, en spécifiant le caractère 'X' après le nom de symbole, comme dans l'exemple suivant:



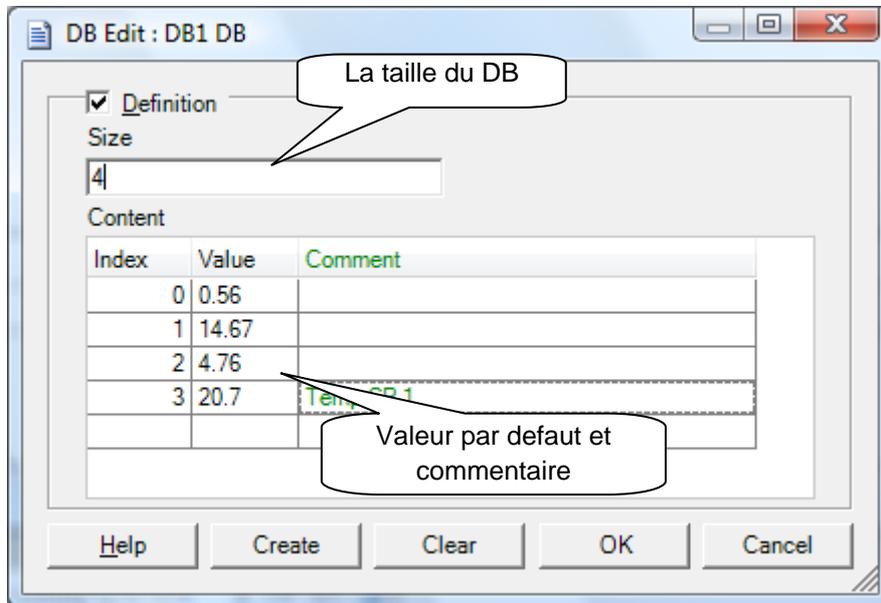
Cliquer sur le bouton pour définir le texte



Ne pas oublier d'utiliser les caractères " au début et à la fin du texte, car, en cas d'oubli, le texte ne sera pas valide.

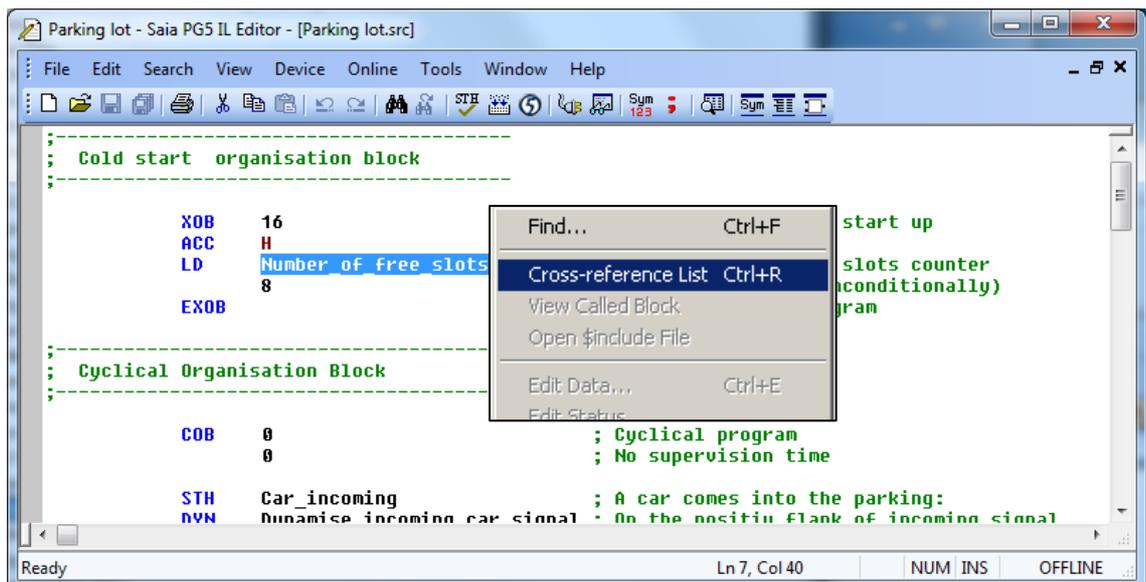
5.3.11 Saisie de DB

Pour éditer les DB, un éditeur spécifique est disponible. Il faut définir un symbole de type 'DB' et cliquer sur le bouton dans la colonne 'Adresse/Value' pour afficher une boîte de dialogue (voir ci-dessous). Entrez la taille de la DB – le nombre d'éléments - et cliquez sur le bouton 'Create'. Il est ensuite possible de définir des valeurs par défaut et des commentaires.



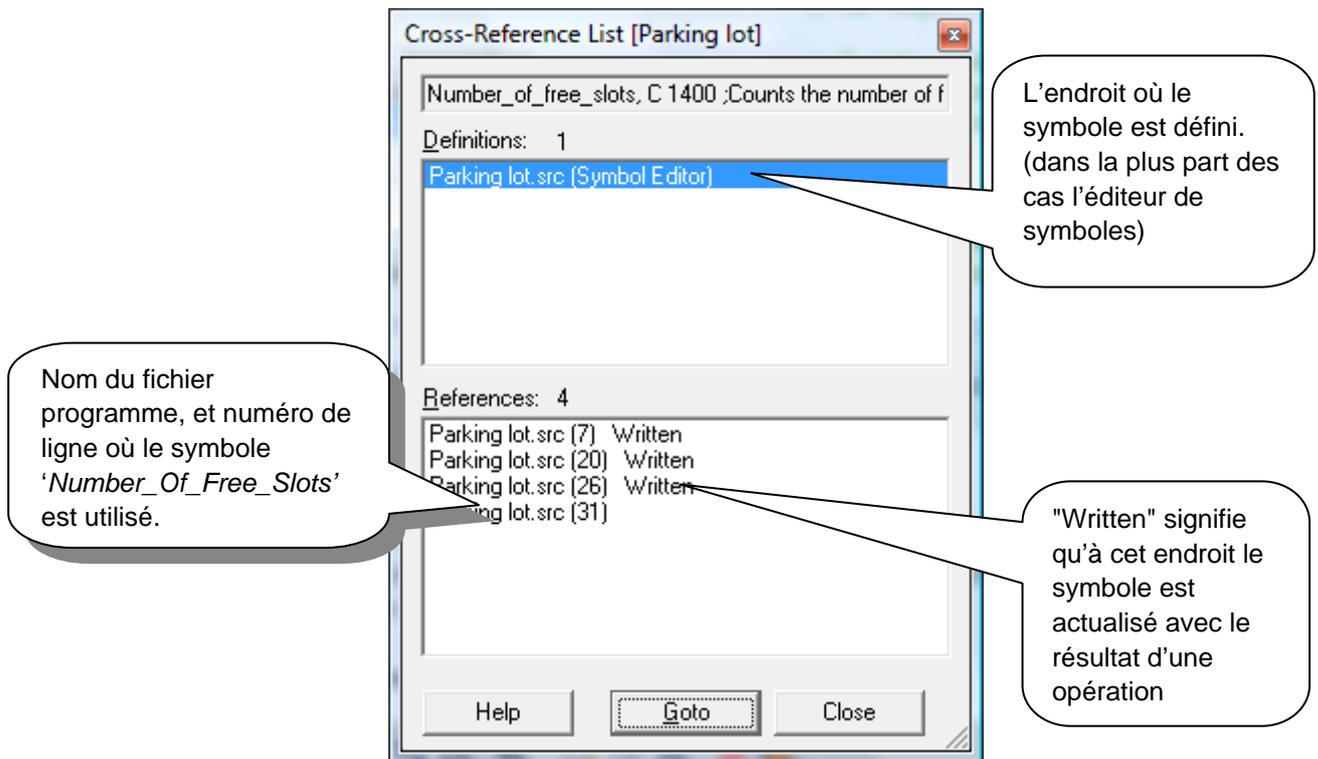
5.3.12 Références croisées des symboles

Un symbole est souvent utilisé à plusieurs reprises dans un ou plusieurs fichiers de programme. Une fois les fichiers sauvegardés, en cliquant sur un symbole avec la touche de droite dans un fichier programme et en sélectionnant la commande 'Cross-reference List' dans le menu contextuel, il est possible d'obtenir la liste des emplacements où le symbole est défini ou utilisé. Cette commande est aussi accessible au moyen du menu contextuel dans l'éditeur de symbole.

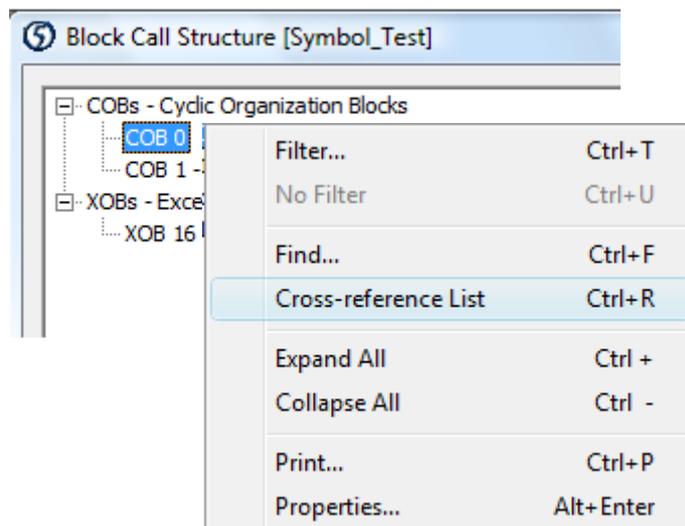


La fonction références croisées - 'cross-reference' – appelle une fenêtre de dialogue dans laquelle sont affichés le nom du fichier, le numéro de ligne et combien de fois le symbole a été utilisé. En double-cliquant sur n'importe quel élément de la liste de

références, le fichier programme sera automatiquement ouvert à l'endroit où le symbole spécifié est défini ou utilisé.



L'outil 'Cross-reference' n'est pas uniquement disponible depuis l'éditeur Fupla ou IL, il est aussi accessible depuis différents vues disponibles dans le gestionnaire de projet, par exemple depuis la vue 'Block Structure' (voir ci-dessous).



5.3.13 Édition dans la grille de l'éditeur de symboles

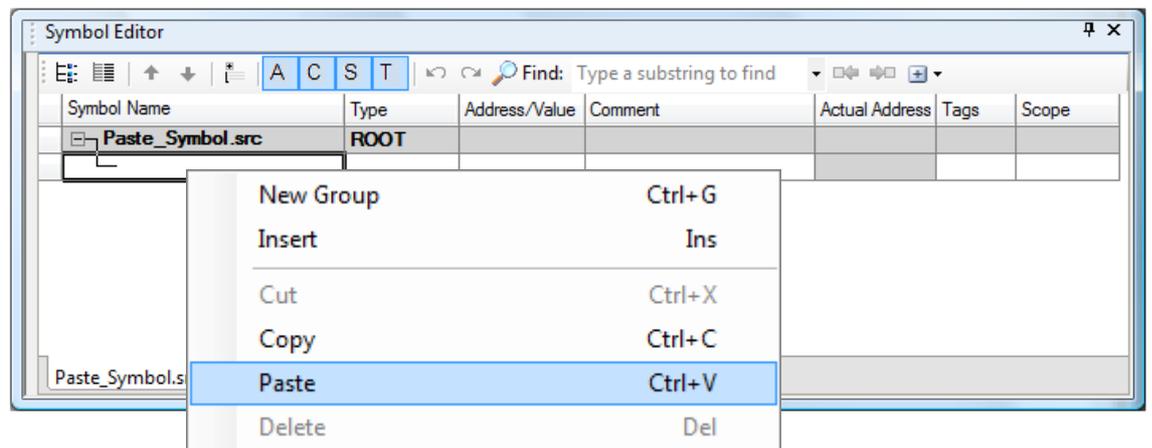
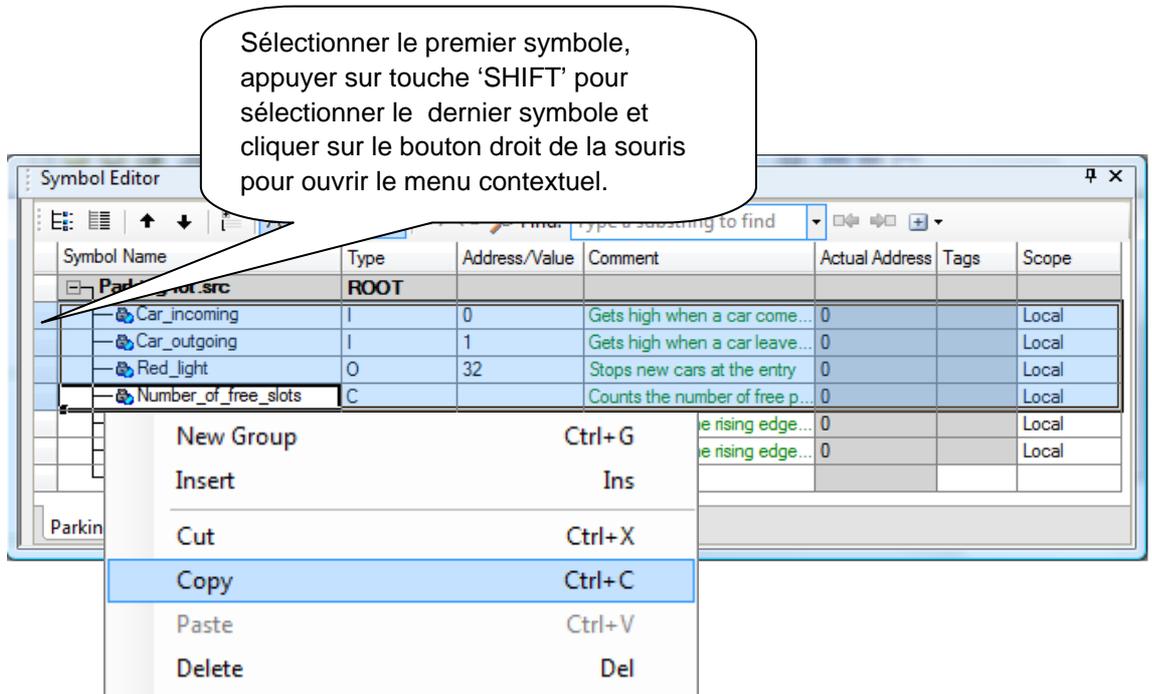
Dans la grille des symboles, il est possible de sélectionner des zones spécifiques et d'effectuer des actions comme copier/coller, couper/coller, effacer ... Ces actions peuvent être effectuées au sein de la même grille ou d'une grille à une autre. Ces actions sont utiles pour transférer ou dédoubler rapidement des définitions de symboles d'un fichier à un autre.

Les zones qui peuvent être sélectionnées et dans lesquelles les actions mentionnées ci-dessus peuvent être effectuées sont les suivantes:

- la définition complète d'un symbole - ligne entière
- une partie de la définition des symboles – uniquement les noms de symboles
- plusieurs définitions de symboles – plusieurs lignes entières
- une partie de la grille - seulement le nom des symboles et les types
- les groupes des symboles
- etc.

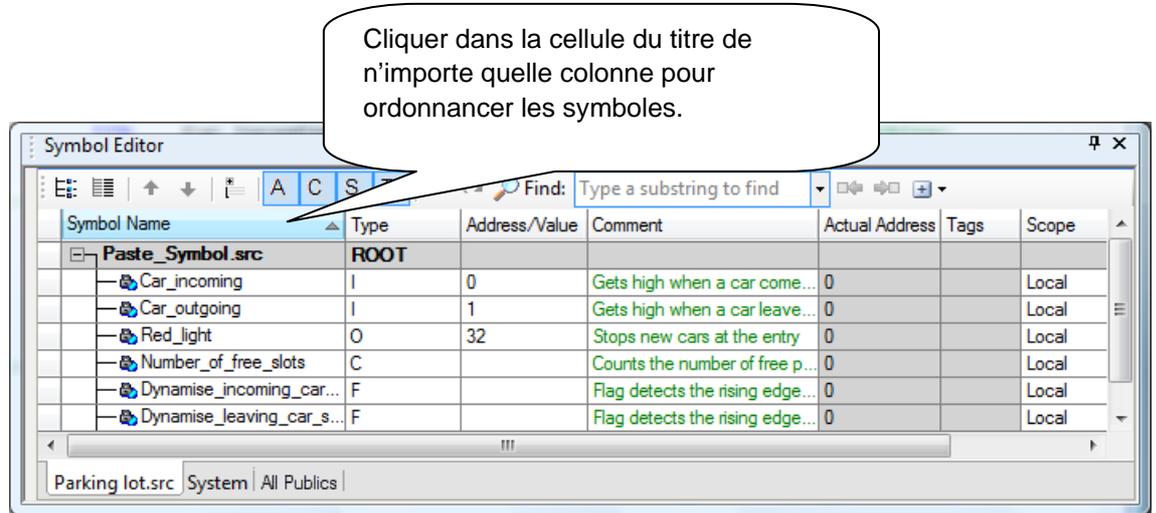
Pour éditer des symboles, il faut sélectionner la zone de symboles et cliquer le bouton droit de la souris et effectuer l'action voulue depuis le menu contextuel. Il faut ensuite aller à l'emplacement souhaité dans la même grille ou un fichier différent et cliquer à nouveau le bouton droit de la souris pour exécuter l'action requise depuis le menu contextuel.

Exemple : copier/coller des définitions de symboles:



5.3.14 Fonctionnalité d'ordonnancement dans la liste des symboles

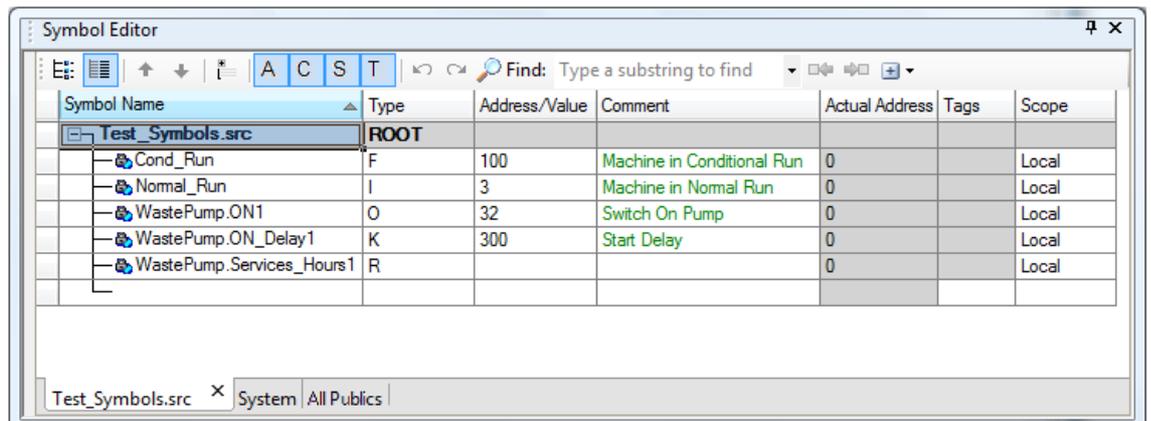
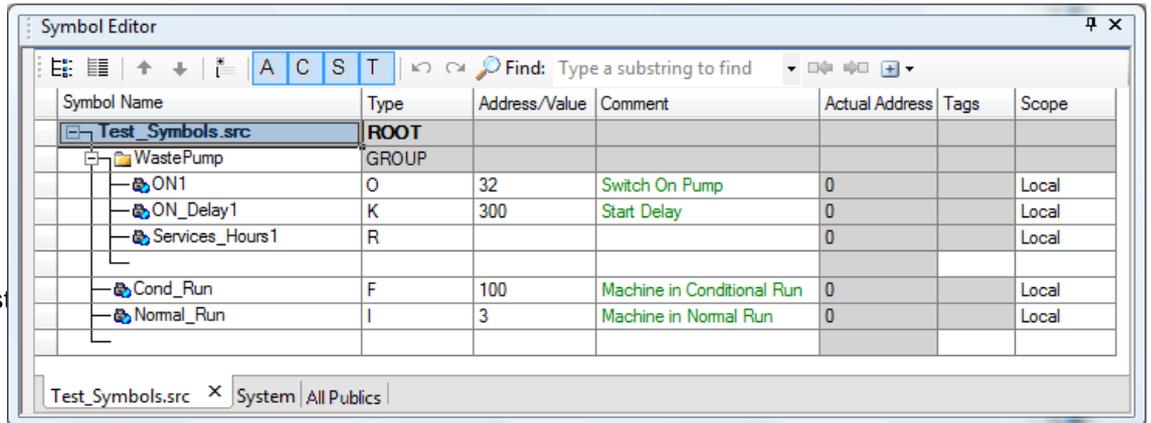
Dans la grille, les symboles peuvent être ordonnancés de différentes façons, en cliquant sur le nom de colonne. Cela signifie que les symboles peuvent être ordonnancés selon leur nom, leur type, leur adresse, leur commentaire, leur adresse réelle, leur tags ou leur scope.



Vous pouvez également ordonnancer vos symboles dans la 'List view'. A l'aide de la commande dans la barre d'outils, on peut commuter de la vue en groupe – les symboles sont affichés selon leur nom de groupe – 'Group View' – vers la vue en liste – les symboles complets, avec le nom de groupe - 'List view'.

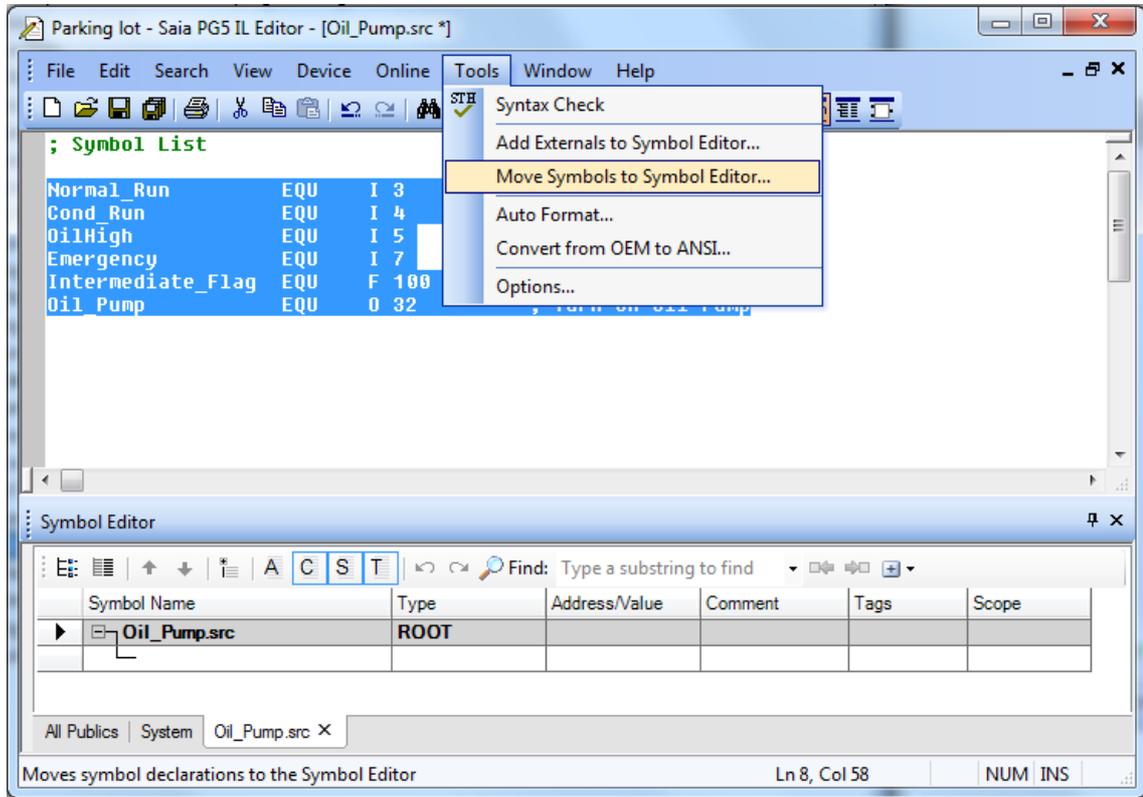


Select Group/List View



5.3.15 Importation de déclarations 'EQU'

Si vous avez de vieux fichiers PG3/4 contenant des déclarations 'EQU' ou 'DOC', il est possible d'importer ces déclarations vers l'éditeur de symboles en sélectionnant la zone contenant les déclarations, puis, sous le menu 'Tools', sélectionner la commande 'Move Symbols to Symbol Editor'.

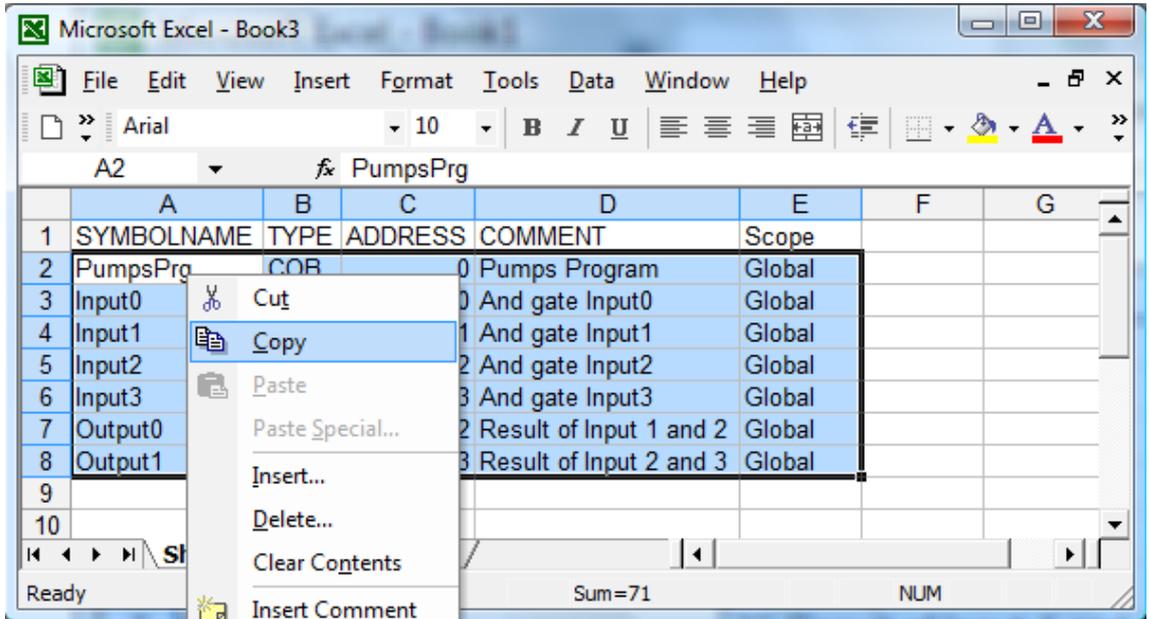


5.3.16 Importation/fusion de symboles

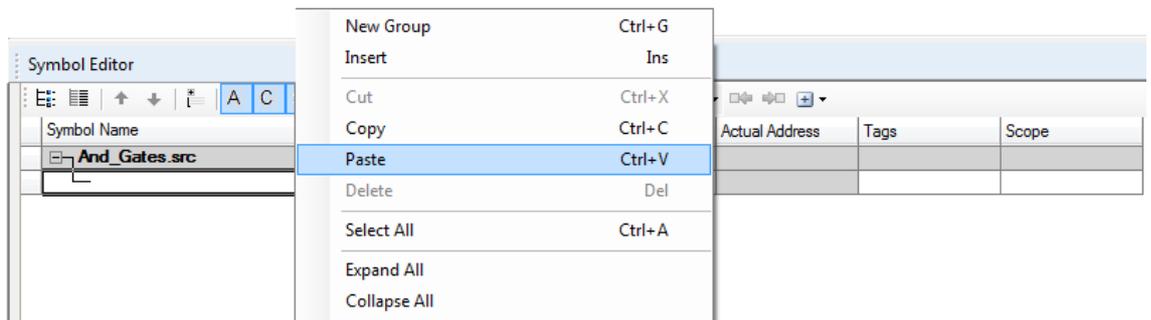
Les symboles peuvent être importés d'un programme externe (Electro CAD, Visi-Plus, ...) vers PG5 et être utilisés à l'intérieur d'un programme. Il en résulte une documentation cohérente tout au long du projet, depuis vos schémas électriques jusqu'au programme du contrôleur dans PG5. Il suffit d'utiliser la fonction d'exportation dans votre CAD pour exporter les symboles dans un fichier texte et de les copier dans l'éditeur de symboles.

On peut facilement copier des symboles à partir d'autres éditeurs tels Word, Excel, etc. vers l'éditeur de symboles. Pour ce faire, une fois la liste de symboles générée dans Excel, Winword ou n'importe quel éditeur de texte, il faut les copier/coller de l'éditeur externe vers l'éditeur de symboles.

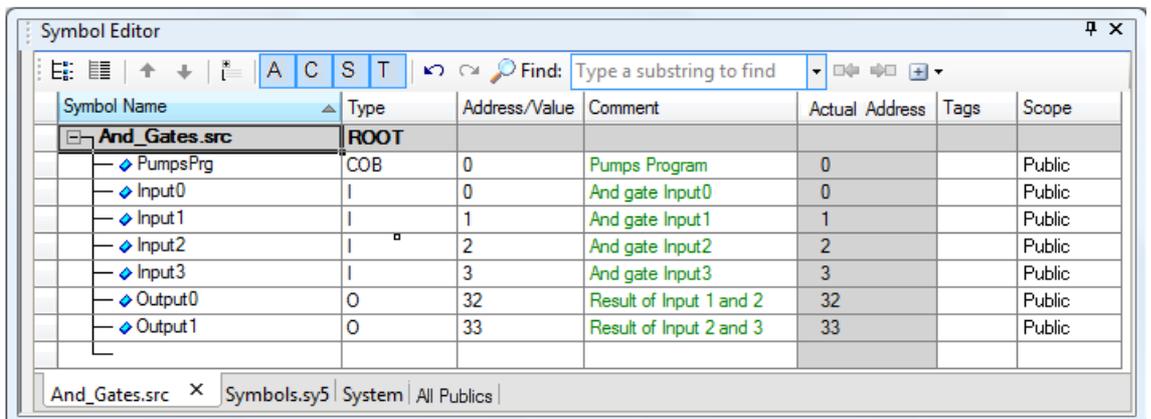
Exemple : Copier/Coller des symboles à partir d'une feuille Excel vers l'éditeur de symboles



Cliquer la touche droite de la souris pour faire apparaître le menu contextuel et sélectionner la fonction coller – 'Paste' - pour retrouver les symboles correspondants dans la grille de l'éditeur de symboles.



Liste des symboles après l'opération copier/coller.

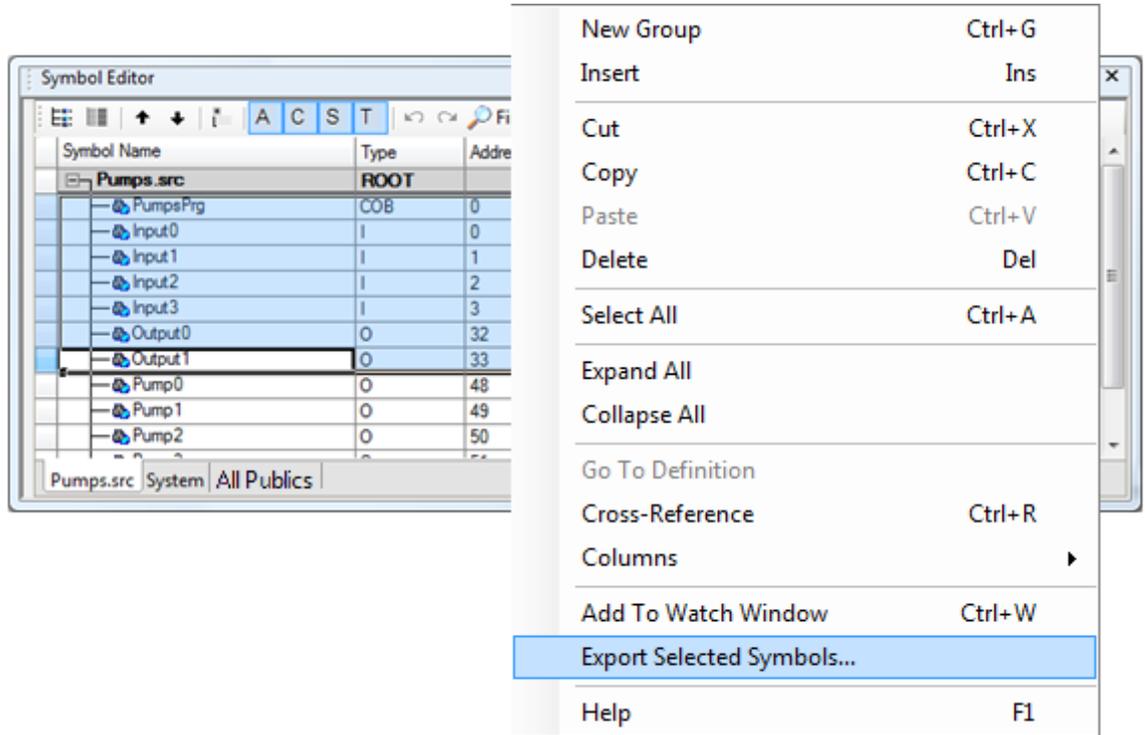


5.3.17 Exportation de symboles

Les symboles d'un programme peuvent être exportés vers d'autres applications comme Excel, Word ou VisiPlus afin de, par exemple, rédiger le rapport de mise en service. Il est également possible de copier/coller des symboles de l'éditeur de symboles vers d'autres applications comme Excel, Word, ou éditeur de texte.

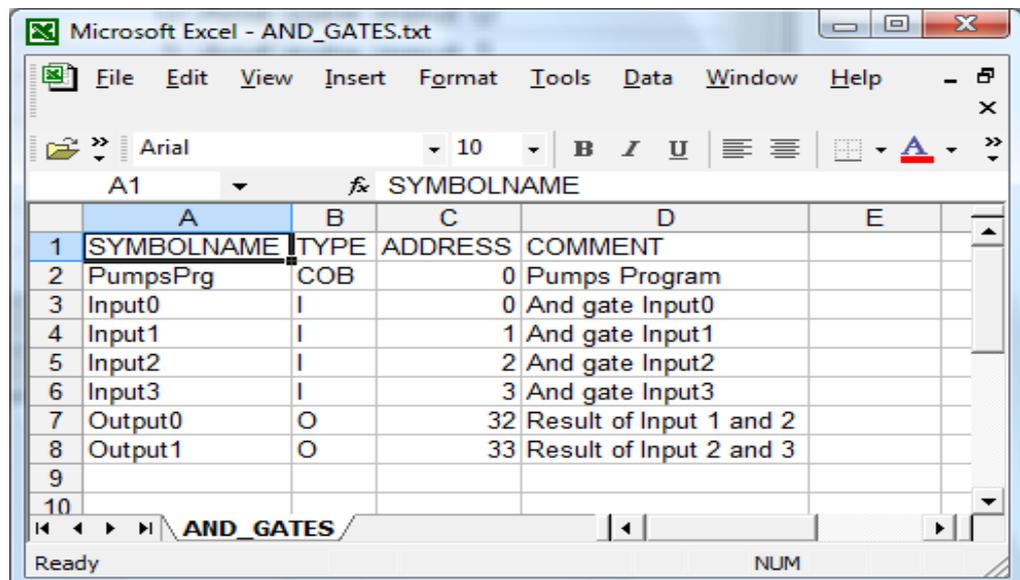
L'exemple ci-dessous montre l'exportation de symbole vers Excel:

Sélectionner les symboles à exporter et exécuter la commande 'Export Selected Symbols...' dans le menu contextuel.



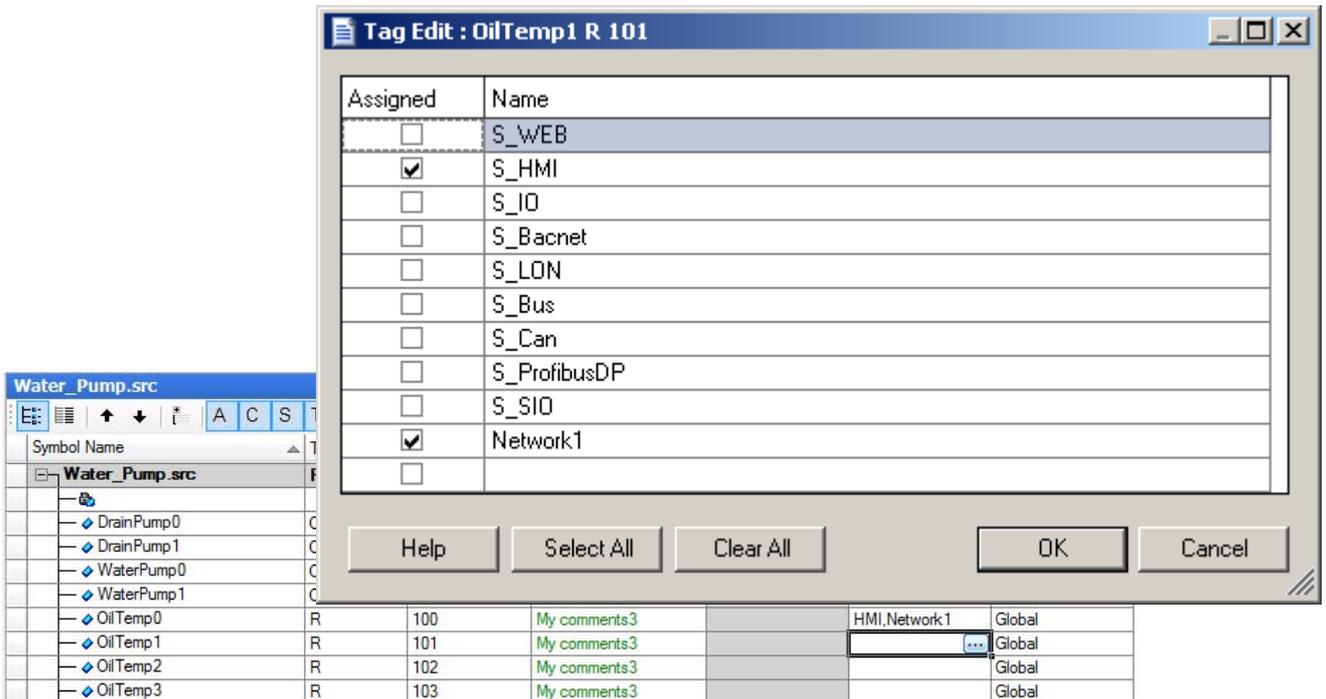
Lors de l'exportation d'une liste de symboles vers Excel, il est fortement conseillé d'utiliser le format de fichier texte, avec les valeurs séparées par des tabulations - 'Tab separated Text file' (*.txt). Vous obtiendrez de meilleurs résultats que si vous utilisez l'option format de fichier Excel (*.xls).

Démarrer Excel et ouvrir le fichier texte qui contient les symboles exportés.



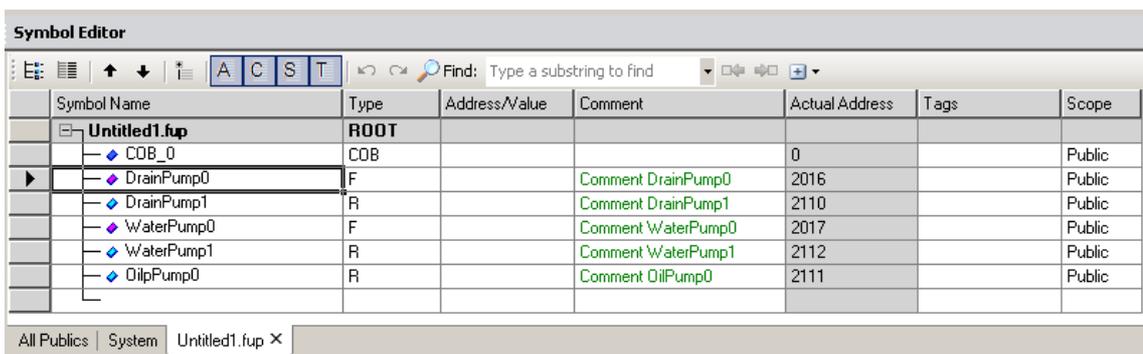
5.3.18 Tags des symboles

Les tags sont utilisées pour associer des informations supplémentaires aux symboles publics. Par exemple, pour marquer les symboles utilisés par une même fonctionnalité comme les symboles réseau, les symboles utilisés dans les éditeurs HMI ou Web, ou dans un programme de supervision. Pour chaque symbole, les tags peuvent être définis au moyen d'une boîte de dialogue comportant la liste des tags déjà définis, en cliquant sur le bouton se trouvant sur la droite de la cellule. Les tags peuvent aussi être directement entrés dans la cellule correspondante de la grille des symboles, séparés par une virgule. Les tags pourront être utilisés pour filtrer les symboles correspondants à un ou plusieurs tags spécifiques et pour faciliter une exportation ciblée des symboles dans des fichiers Excel ou 'Rxp'. Les 'tags' n'ont aucune influence sur le programme utilisateur.



5.3.19 Adresse actuelle

Après la construction du programme – le 'build', la colonne 'Actual Address' affiche les adresses qui ont été attribuées aux symboles dynamiques. L'espace d'adressage dynamique pour chaque type de données peut être modifié dans les options de compilation - 'Build Options' - du gestionnaire de projet.



5.3.20 Initialisation de la valeur d'un symbole

Il y a deux façons d'initialiser la valeur d'un symbole utilisé dans le contrôleur:

- Initialisation de la valeur du symbole au démarrage à froid (power-up)
- initialisation de la valeur du symbole lorsque le programme est chargé dans le contrôleur.

Initialisation lors du démarrage à froid

L'initialisation de la valeur d'un symbole lors d'un démarrage à froid se fait dans le bloc de fonction XOB 16. Ce bloc de fonction est traité une seule fois, durant le démarrage à froid du contrôleur. L'utilisateur écrit le code IL pour initialiser la valeur des symboles dans le bloc XOB 16.

Exemple: initialisation d'un indicateur et d'un registre lors d'un démarrage a froid du contrôleur.

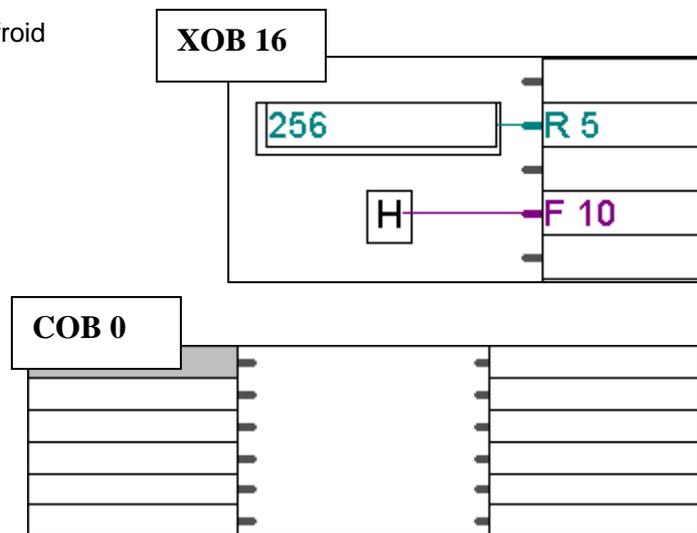
Programme en IL

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

EXOB

COB 0 ;Bloc cyclique
    0
...
;programme
...
ECOB
```

Programme en Fupla



Pour de plus amples renseignements au sujet des blocs COB et XOB, veuillez consulter le chapitre 7.

Initialisation lors du chargement du programme dans le contrôleur

Pour initialiser un symbole lors du chargement du programme dans le contrôleur, l'adresse de symbole doit être suivi par ':=' et la valeur d'initialisation.

Exemple:

SymbolA	R	5 := 256	
SymbolB	F	10 := 1	



Attention

Ne pas oublier de cocher l'option 'First-time Initialization Data' lorsque vous chargez le programme dans le contrôleur:


 First-time Initialisation Data

5.3.21 Les mots réservés

Les mots suivants sont réservés, et ne peuvent pas être utilisés comme noms de symbole:

- instructions de l'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,
- instructions spéciales: MOV : N, Q, B, W, L, D,
- codes conditionnelles: H, L, P, N, Z, E,
- toutes les instructions mnémoniques
- symboles prédéfinis
- symboles internes réservés à l'allocation automatique des ressources, commençant par un caractère souligné. Exemple: TEXT, F
- symboles internes __CSTART__, utilisés avec \$\$.

5.3.22 Messages d'erreurs et d'avertissements

Les messages d'erreurs et d'avertissement sont affichés lorsque les entrées faites dans l'éditeur de symboles ne sont pas valables. Voici quelques exemples :

Nom de symbole trop court:

Dans le cas où un seul caractère est entré comme nom de symbole, l'éditeur affiche une erreur 'Symbol name too short (min. 2 Characters)'.

Les mêmes noms des symboles définis plusieurs fois:

Lorsqu' un nouveau symbole public est ajouté avec un nom qui est déjà utilisé par un autre symbole public, le symbole éditeur affichera une erreur et les deux symboles seront marqués en rouge.

Lorsqu'un nouveau symbole local est ajouté avec un nom qui est déjà utilisé par un autre symbole local dans un même module ou fichier, le symbole éditeur affichera une erreur et les deux symboles seront marqués en rouge.

Lorsqu'un nouveau symbole local est ajouté avec un nom qui est déjà utilisé par un symbole public, l'éditeur de symboles affichera alors un message d'avertissement

Il est nécessaire de modifier le nom du symbole afin de supprimer ces erreurs.

Table des matières

TABLE DES MATIERES	1
6 PROGRAMMER AVEC FUPLA.....	3
6.1 Créer un nouveau projet.....	4
6.2 Organisation d'une fenêtre Fupla	5
6.3 Edition des symboles	6
6.3.1 Ajouter un nouveau symbole dans la liste <i>Symbols</i>	7
6.3.2 Modes d'adressage des symboles	8
6.3.3 Utiliser un symbole de la liste dans le programme Fupla	9
6.3.4 Symboles Local, Public et External.....	10
6.4 Edition des connecteurs.....	11
6.4.1 Placer un connecteur.....	11
6.4.2 Editer le symbole à l'intérieur du connecteur	11
6.4.3 Méthode rapide pour placer un symbole et son connecteur.....	11
6.4.4 Glisser, Copier/Coller, effacer un symbole	11
6.4.5 Copier/Coller, effacer un connecteur.....	12
6.4.6 Connecteur extensible (stretchable).....	12
6.4.7 Déplacer un connecteur verticalement.....	12
6.5 Edition des FBoxes Fupla.....	13
6.5.1 FBox selector.....	13
6.5.2 Edition d'une FBox	14
6.5.3 Edition d'une FBox extensible (stretchable).....	14
6.5.4 Edition d'une inversion logique.....	14
6.5.5 Dynamisation.....	15
6.5.6 Commentaires.....	15
6.5.7 FBox Help	15
6.6 Liaisons entre les FBox et connecteurs	16
6.6.1 Liaison par déplacement de la FBox.....	16
6.6.2 Liaison avec routage automatique	16
6.6.3 Liaison multiple avec routage automatique	16
6.6.4 Liaison de toutes les entrées, sorties d'une FBox à des connecteurs	16
6.6.5 Suppression de lignes, FBox, connecteurs ou symboles.....	17
6.6.6 Déplacer une FBox, un connecteur verticalement sans défaire les liens	17
6.6.7 Insérer une FBox sans défaire le lien.....	17
6.6.8 Règles à respecter	17
6.7 Edition de pages Fupla.....	18
6.7.1 Ajouter une page.....	18
6.7.2 Supprimer une page	18
6.7.3 Navigation entre pages	18
6.7.4 Page properties	19
6.7.5 Traitement du programme par le device	19

6.8	Copier, coller.....	20
6.8.1	Copier, coller une partie de programme	20
6.8.2	Copier, coller des symboles.....	20
6.9	Templates	21
6.9.1	Créer un template	21
6.9.2	Importation de templates	22
6.10	Édition d'un premier programme Fupla.....	24
6.10.1	Objectif.....	24
6.10.2	Méthode.....	24
6.10.3	Programmation	26
6.11	Construction du programme (Build)	27
6.12	Charger le programme dans l'automate	28
6.13	Débugger un programme (Debug)	28
6.13.1	Go On/Offline, Run, Stop, Step-by-step.....	28
6.13.2	29
6.13.3	Point d'arrêt.....	29
6.13.4	Visualiser les symboles ou leurs adresses.....	30
6.13.5	Visualiser l'état des symboles avec Fupla	30
6.13.6	Éditer les symboles en ligne	30
6.13.7	Visualiser l'état des symboles avec le <i>Watch window</i>	31
6.13.8	Mise à l'heure de l'horloge de l'automate	32
6.14	Paramètres d'ajustage.....	33
6.14.1	Initialisation des FBox CVC.....	34
6.14.2	FBox CVC avec paramètres d'ajustage	35
6.14.3	Mini application CVC	35
6.14.4	Ajustage des paramètres en ligne.	36
6.14.5	Rétablir les paramètres du fichier Fupla.	36
6.14.6	Sauvegarde des paramètres en ligne dans le fichier Fupla.....	37
6.14.7	Définir les symboles systèmes des paramètres d'ajustage.....	38
6.14.8	Définir les adresses systèmes des paramètres d'ajustage.....	39
6.15	Mise en service d'un module analogique	40
6.15.1	Saisie d'une mesure analogique.....	40
6.15.2	Exemple pour les entrées analogiques PCD2.W340.....	41
6.15.3	Exemple pour les sorties analogiques PCD2.W610.	42

6 Programmer avec Fupla

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 PLa*n » 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 FBoxes 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, Ventilation 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.

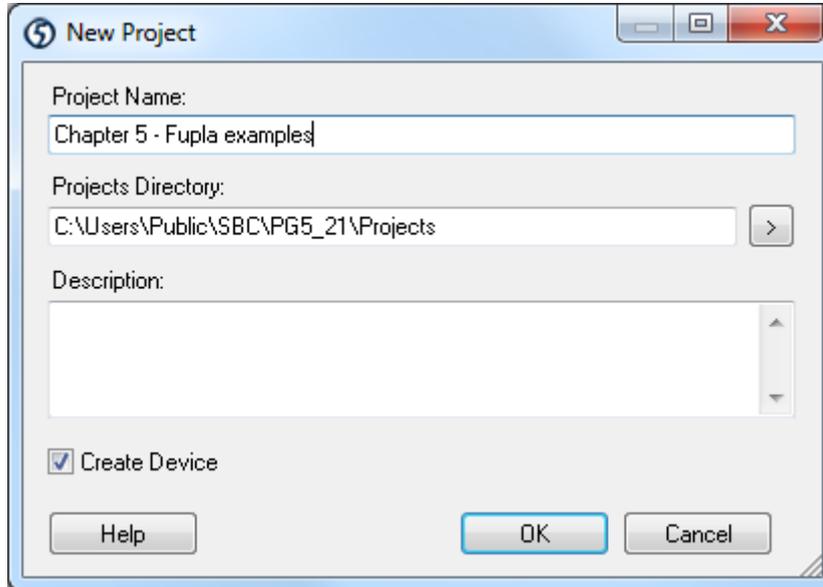
Bien d'autres bibliothèques comme pour les réseaux de communication LON, EIB ou les produits Belimo ... sont aussi disponibles.

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

6.1 Créer un nouveau projet

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

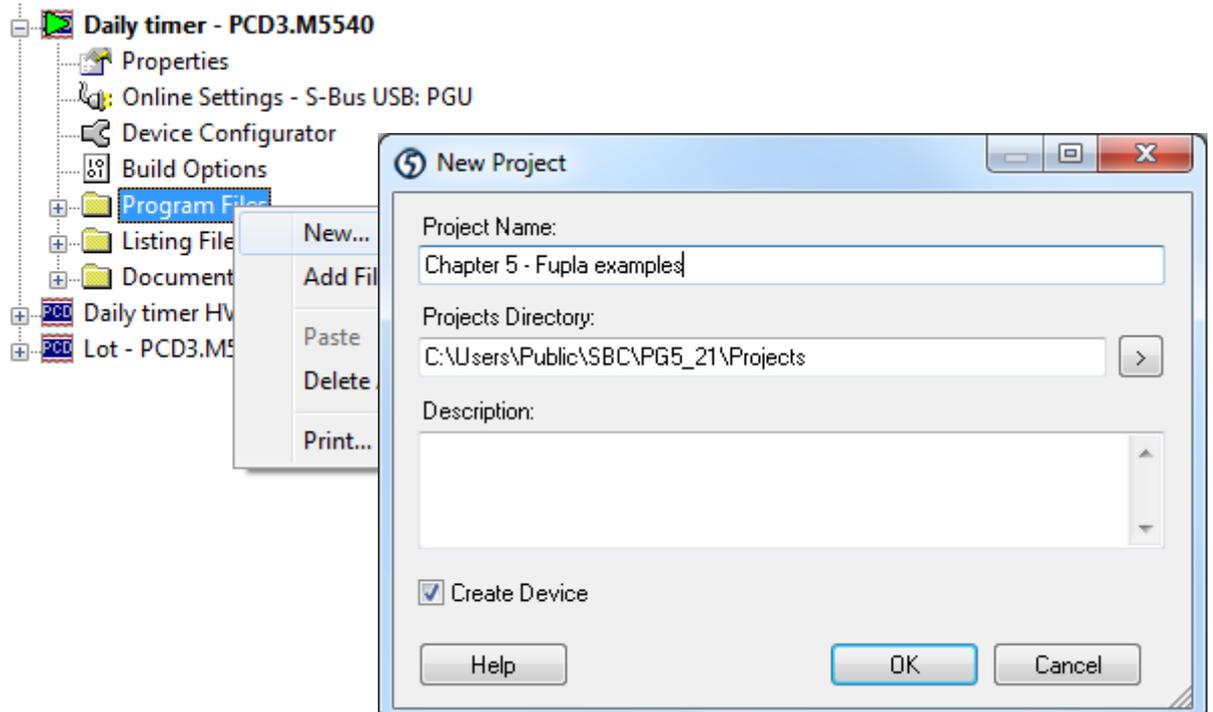
Dans la fenêtre *SAIA Project Manager*, sélectionner le menu *Project, New...* et créer le nouveau projet.



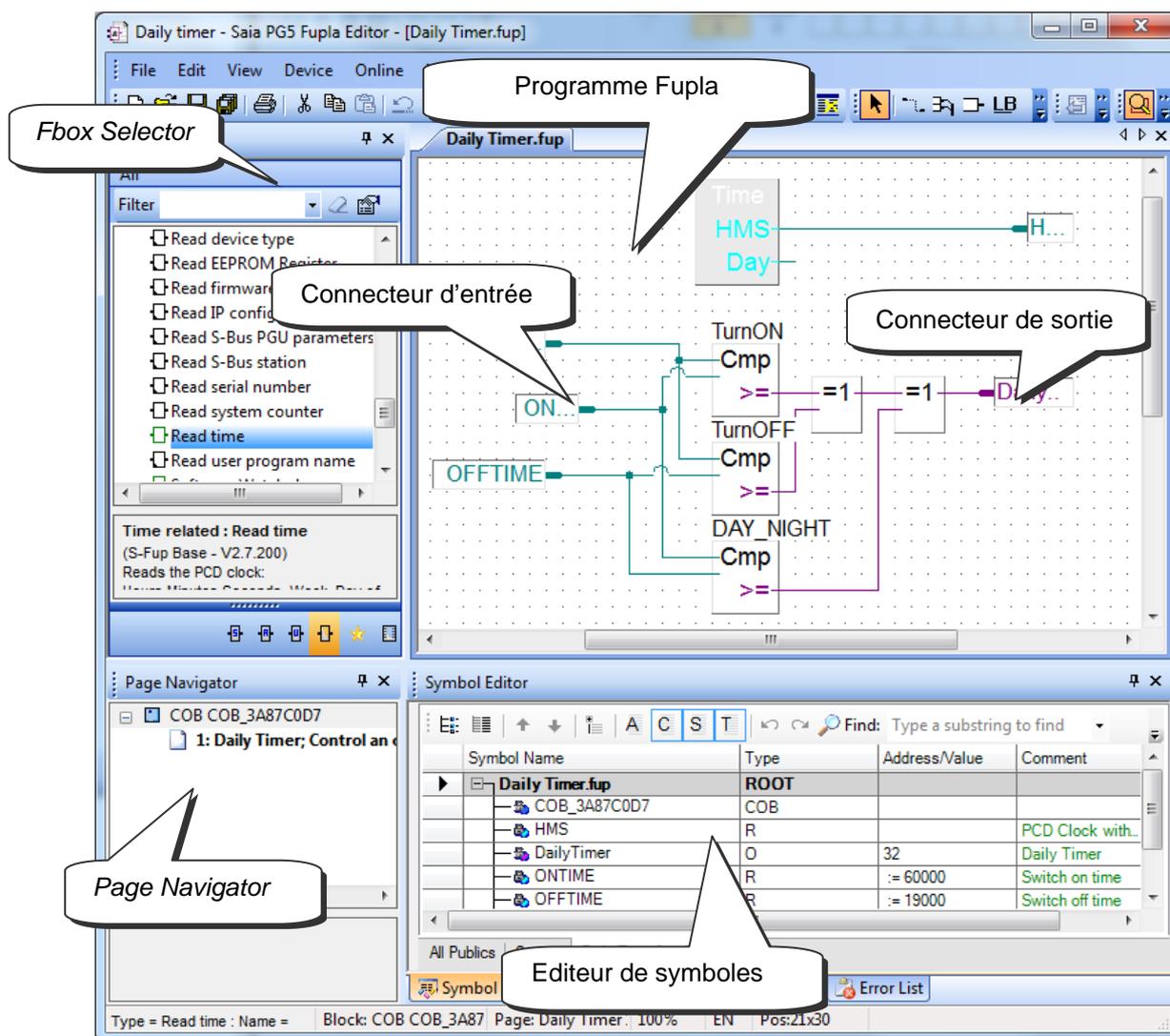
Pour ajouter un nouveau fichier programme dans le projet, marquer le répertoire *Program Files*, presser le bouton droit de la souris et sélectionner le menu *New...* ou sélectionner le bouton *New File* de la barre d'outils:



New File



6.2 Organisation d'une fenêtre Fupla



L'automate 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 *Symbol Editor*. 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 nous utilisons que dans les connecteurs d'entrée.

Au centre de la page nous avons le programme constitué de différentes FBox graphiques prélevées dans la fenêtre *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 FBoxes *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.

6.3 Edition des symboles



Show Hide
Symbols Editor

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

La fenêtre *Symbol Editor* rassemble les symboles du programme dans une liste affichée avec le bouton *Show or Hide the Symbols Editor* ou le menu *View, Symbols*. 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 *Symbol Editor*. Il n'est alors pas nécessaire de reporter la correction dans les connecteurs du programme Fupla mais uniquement dans la fenêtre *Symbol Editor*. 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 l'automate

Indicateurs: F 0...16383

Registres: R 0...16383

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

Scope

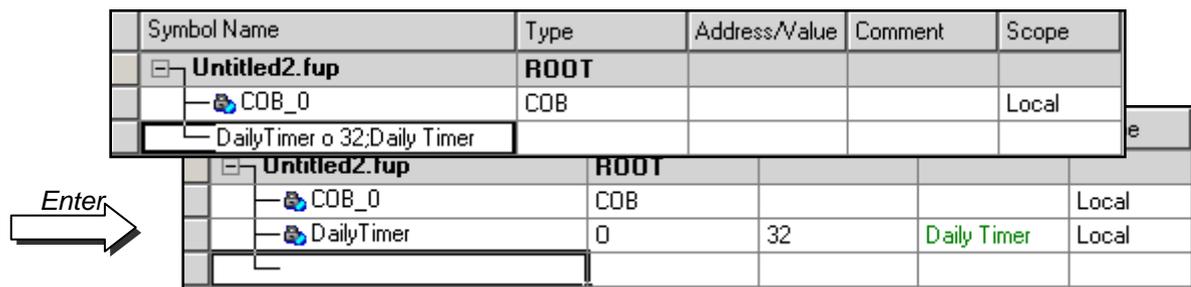
Le scope définit l'accessibilité des symboles: les symboles locaux sont uniquement accessibles par le fichier programme qui les définit. Les symboles *Public* sont accessibles par tous les fichiers présents dans le *device*. Les symboles *External* sont définis à l'extérieur du fichier ouvert.

6.3.1 Ajouter un nouveau symbole dans la liste *Symbols*

Méthode simple

Pour ajouter un symbole dans la liste, ouvrir la fenêtre *Symbol Editor*, une ligne libre de toute définition est toujours disponible, sélectionnez l'une après l'autre ses cellules pour compléter les champs *Symbol Name*, *Type*, *Address/Value*, *Comment* et *Scope*.

Méthode rapide 1



Symbol Name	Type	Address/Value	Comment	Scope
Untitled2.fup	ROOT			
COB_0	COB			Local
DailyTimer o 32;Daily Timer	COB			Local
DailyTimer o 32;Daily Timer	0	32	Daily Timer	Local

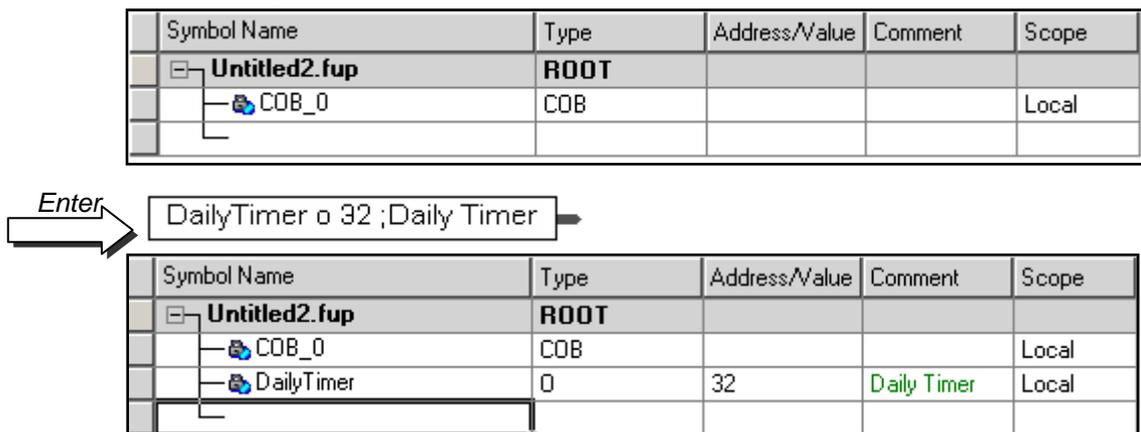
Il est aussi possible d'introduire les variables des divers champs d'informations depuis le champs **Symbol Name**. C'est plus pratique et plus rapide. Voir l'exemple ci-dessus.

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 cellules correspondantes.

Méthode rapide 2



Symbol Name	Type	Address/Value	Comment	Scope
Untitled2.fup	ROOT			
COB_0	COB			Local
DailyTimer	0	32	Daily Timer	Local

Les nouveaux symboles peuvent être aussi ajoutés à partir des connecteurs Fupla, selon la même syntaxe que la méthode précédente

* [] représente un caractère avec un espace.

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.

6.3.2 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

Symbol Name	Type	Address/Value	Comment	Scope
Untitled2.fup	ROOT			
 DailyTimer	0	32	Daily Timer	Local

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

Symbol Name	Type	Address/Value	Comment	Scope
Untitled2.fup	ROOT			
 DailyTimer	0	32	Daily Timer	Local

Le symbole est défini avec un nom, un type, une adresse est optionnellement un commentaire. Les corrections du nom, 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 nom de symbole.

Adressage dynamique

Symbol Name	Type	Address/Value	Comment	Scope
Daily Timer.fup	ROOT			
 HMS	R		PCD Clock w...	Local

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 *Build Options*. (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.

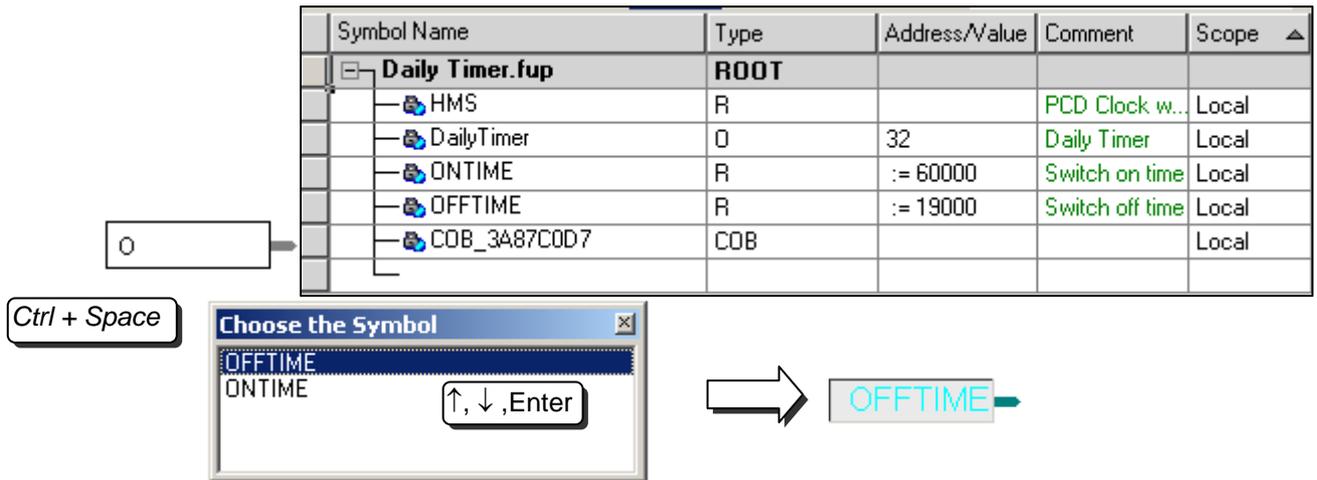
6.3.3 Utiliser un symbole de la liste dans le programme Fupla

A l'édition du programme, les symboles déjà définis dans la fenêtre *Symbol Editor* 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.



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

Symbol Name	Type	Address/Value	Comment	Scope
Daily Timer.fup	ROOT			
ONTIME	R	:= 60000	Switch on time	Local

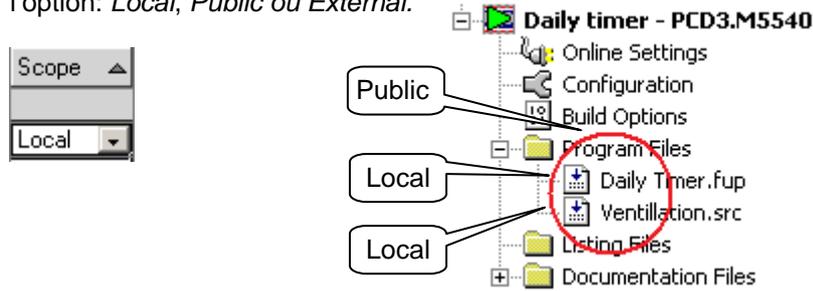


Cette manière de faire usage d'un symbole ne laisse aucune possibilité de faire une erreur de frappe. Dans la fenêtre *Symbol Editor*, placer la souris sur le bouton gauche de la souris et le maintenir pressé. Glisser la souris au dessus d'un connecteur vide et lâcher le bouton de la souris. Le symbole 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.

6.3.4 Symboles Local, Public et External

La fenêtre de définition des symboles comporte une cellule *Scope* pour définir l'option: *Local, Public ou External*.



Le scope définis l'accessibilité des symboles:

Les symboles locaux sont uniquement accessibles par le fichier programme qui les définis. Par exemple, les fichiers *Parking lot.fup* **ou** *Ventillation.src*.

Les symboles Public sont accessibles par tous les fichiers présents dans le *device*. Par exemple, les fichiers *Parking lot.fup* **et** *Ventillation.src*.



L'éditeur de symboles affiche plusieurs vues: une vue avec le nom du fichier ouvert, ainsi que les vues *All Publics* et *System*.

La vue avec le nom du fichier ouvert supporte la définition de tous les symboles publics et locaux nécessaires au programme du fichier ouvert.

Les nouveaux symboles ajoutés dans l'éditeur de symboles ou Fupla sont par défaut local ou public selon l'option définie sous le menu *View, Option, Symbols, Add symbols with Public scope*.

La vue *ALL Publics* rassemble les symboles publics définis dans le *device*. La vue *System* affiche la liste de tous les symboles systèmes. Les symboles des vues *All Publics* et *System* sont mis à jour lorsque le fichier est sauvé ou lors du *Build*.

La vue *All Publics* utilise le résultat du *Buid* pour rassembler les symboles publics définis dans les divers fichiers programmes du *device* sous une même est unique vue.

Pour placer un symbole public ou system dans le programme, sélectionner le symbole dans la vue *All Publics* ou *System* et le glisser dans la page Fupla. La référence au symbole public est dupliquée dans la grille correspondant au fichier ouvert avec un scope *External*. Cela indique que ce symbole est défini à l'extérieur du fichier.

La vue *All Publics* n'est pas éditable, si un symbole doit être complété ou corrigé, la grille indique le nom du fichier qui contient la définition et qu'il convient d'ouvrir pour travailler sur le symbole. (Menu de contexte *Columns, Source*)

Mais il est beaucoup plus intéressant de sélectionner le symbole dans la vue *All Publics* puis le menu *Goto Definition* pour ouvrir le fichier qui contient la définition du symbole.

6.4 Edition des connecteurs

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

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-mêmes selon votre convenance, veuillez désactiver l'option suivante avec le menu : *View, Options..., Workspace, 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 Unused Connectors.*

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

6.4.1 Placer un connecteur



Pour ajouter un connecteur et son symbole sur une page Fupla, sélectionner le bouton *Add Connector* 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.

6.4.2 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 *Symbol Editor*.

6.4.3 Méthode rapide pour placer un symbole et son connecteur

Les symboles déjà présents dans la fenêtre *Symbol Editor*, peuvent être glissé 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.

6.4.4 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 droite de la souris affiche le menu de contexte avec les opérations disponibles.

6.4.5 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.

6.4.6 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*.

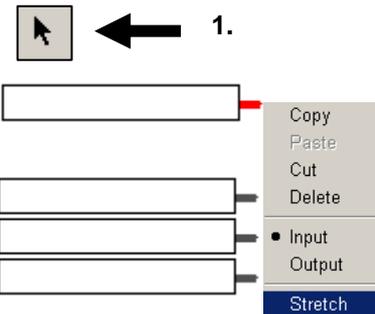
Sélectionner le connecteur sur la surface représentée en rouge.

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.



6.4.7 Déplacer un connecteur verticalement

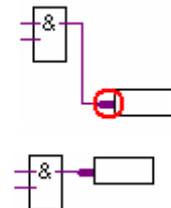
Pour déplacer le connecteur placer la souris dans le cercle rouge .

Presser la touche Shift du clavier et la maintenir.

Presser le bouton gauche de la souris et le maintenir.

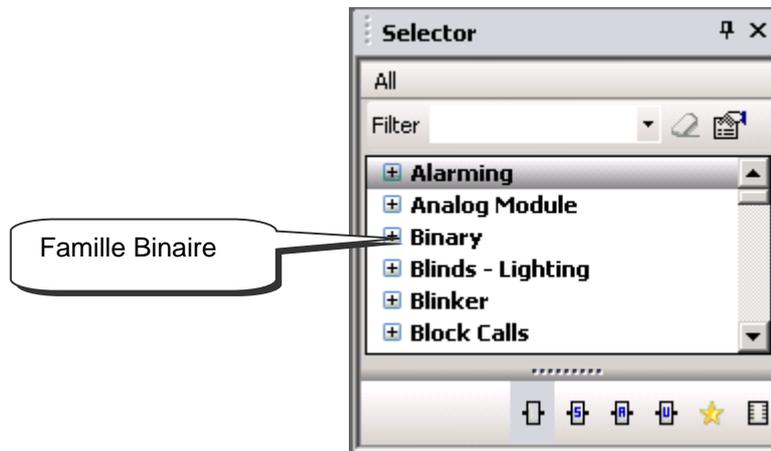
Glisser la souris verticalement sur une surface libre de la page.

Libérer le bouton de la souris et du clavier.



6.5 Edition des FBoxes Fupla

6.5.1 FBox Selector



Show/Hide
Selector
Window

La fenêtre *FBox Selector* rassemble toutes les FBoxes nécessaires à l'élaboration des programmes. Elles sont organisées en familles que nous pouvons considérer comme des bibliothèques indépendantes les unes des autres couvrant un domaine d'application particulier. Nous avons par exemples les familles suivantes:

<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>Compteurs</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, ...

...

Nos bibliothèques répondent à toutes les exigences d'automatisation d'un *device*, ce qui implique un vaste choix de familles et de FBoxes parmi lesquels il est parfois difficile de trouver la FBox qui convient. Ainsi nous offrons différentes facilités de recherches.

Si toutes les familles de FBoxes sont refermées (menu de contexte *Collapse All*) la sélection d'une famille suivie d'un caractère sur le clavier permet de faire défiler les familles dont le nom commence par la lettre sélectionnée. La recherche est aussi appliquée à toutes les FBoxes dont la famille est dépliée.

La *tool bar* dispose d'une cellule *Find* dont la saisie d'une chaîne de caractère filtre les FBoxes. Par exemple entrer *ADD* et valider l'entrée avec la touche *Enter* du clavier. Alors le *Selector* ne présente que les FBoxes supportant les additions arithmétiques avec les nombres entiers et flottants. Pour afficher la totalité des FBoxes disponibles, effacer l'entrée *Find* et presser la touche *Enter* du clavier.



Libraries

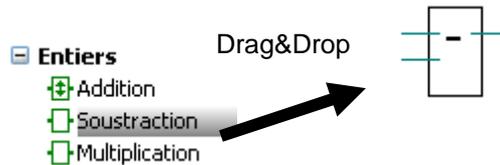
Le répertoire *Libraries* du *Project Manager Tree* affiche toutes les bibliothèques disponibles à partir du répertoire d'installation PG5 et du projet ouvert. Les bibliothèques dont nous ne souhaitons pas faire usage peuvent être désélectionnées, ce qui permet de réduire le nombre des familles de FBoxes affichées dans la fenêtre *Selector* de Fupla.



Favorites

Les FBoxes dont nous faisons un usage fréquent peuvent être collectionnées dans les favoris. Sélectionner la FBox puis le contexte menu *Add to Favorites* et afficher les favoris avec le bouton *Favorites*.

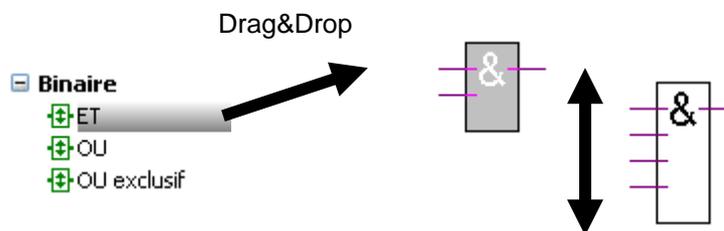
6.5.2 Edition d'une FBox



Les FBox nécessaires à décrire le programme sont saisies dans la fenêtre *Selector* puis glissée sur la page Fupla.

6.5.3 Edition d'une FBox extensible (stretchable)

Certaines FBoxes disposent d'un icône avec une flèche vers le haut et le bas. Cela signifie que la fonction est extensible, le nombre de liaisons est défini par un mouvement vertical de la souris.



Sélectionner une FBox extensible.

Drag&Drop sur la page Fupla.

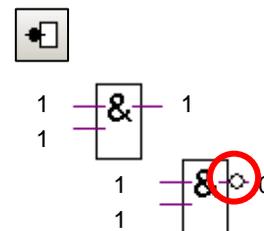
Déplacer la souris verticalement pour définir le nombre d'entrées.

Presser le bouton gauche de la souris.

6.5.4 Edition d'une inversion logique

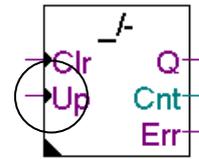
Sélectionner le bouton *Invert Connector*.

Placer la souris sur le lien d'entrée ou de sortie d'une FBox logique et presser le bouton gauche de la souris.



6.5.5 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.



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.

Fbox: *Compteurs*,
Haut + RAZ

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*



6.5.6 Commentaires

Des commentaires peuvent être insérés avec le programme:

Sélectionner le bouton *Place comment*.

Placez le commentaire sur la page Fupla, puis presser le bouton gauche de la souris.

Ecrire le commentaire.

Presser la touche *ENTER* du clavier.



Daily Timer

6.5.7 FBox Help

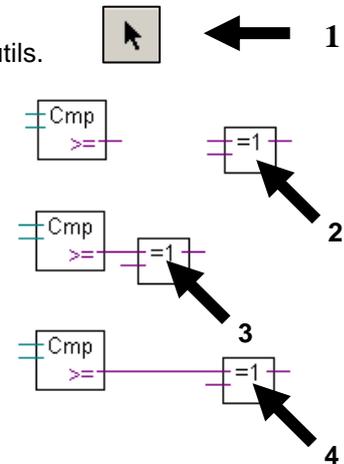
Pour obtenir une description complète d'une FBox, sélectionner la FBox dans le *Selector* ou la page Fupla puis presser la touche F1.

Pour identifier rapidement une FBox inconnue mais présente dans une page Fupla. Faire apparaître la fenêtre *Selector*, sélectionner la FBox inconnue à l'aide de la souris, la fenêtre *Selector* présente la famille et la FBox sélectionnée.

6.6 Liaisons entre les FBox et connecteurs

6.6.1 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.



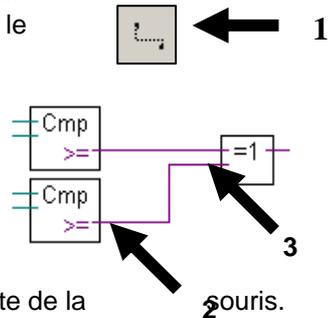
6.6.2 Liaison avec routage automatique

- 1 Sélectionner le bouton *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 :

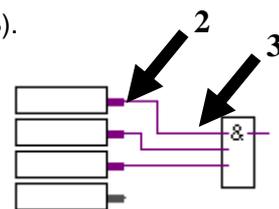
Il est aussi possible de sélectionner des points de passages intermédiaires.

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



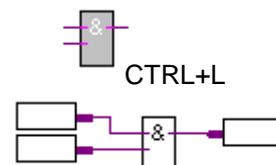
6.6.3 Liaison multiple avec routage automatique

- 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.



6.6.4 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 to Side Connectors*.



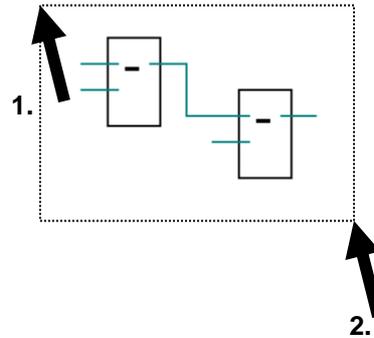
6.6.5 Suppression de lignes, FBox, connecteurs ou symboles

Sélectionner le bouton *Delete Object* de la barre d'outils puis sélectionner les liens, FBox, connecteurs ou symboles à effacer.



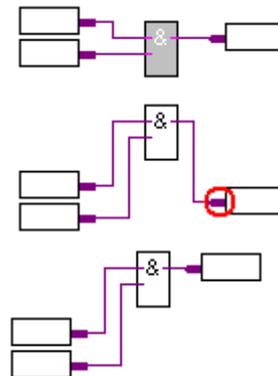
Une autre solution, plus rapide consiste à marquer un espace et de l'effacer.

- 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*.



6.6.6 Déplacer une FBox, un connecteur verticalement sans défaire les liens

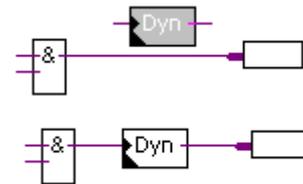
Placer la souris au dessus de la FBox.
 Presser la touche Shift du clavier et la maintenir.
 Presser le bouton gauche de la souris et le maintenir.
 Glisser la souris verticalement sur une surface libre de la page.
 Libérer le bouton de la souris et du clavier.



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

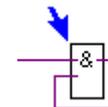
6.6.7 Insérer une FBox sans défaire le lien.

Sélectionner la FBox à insérer dans la fenêtre *Selector*.
 La placer au dessus du lien.



6.6.8 Règles à respecter

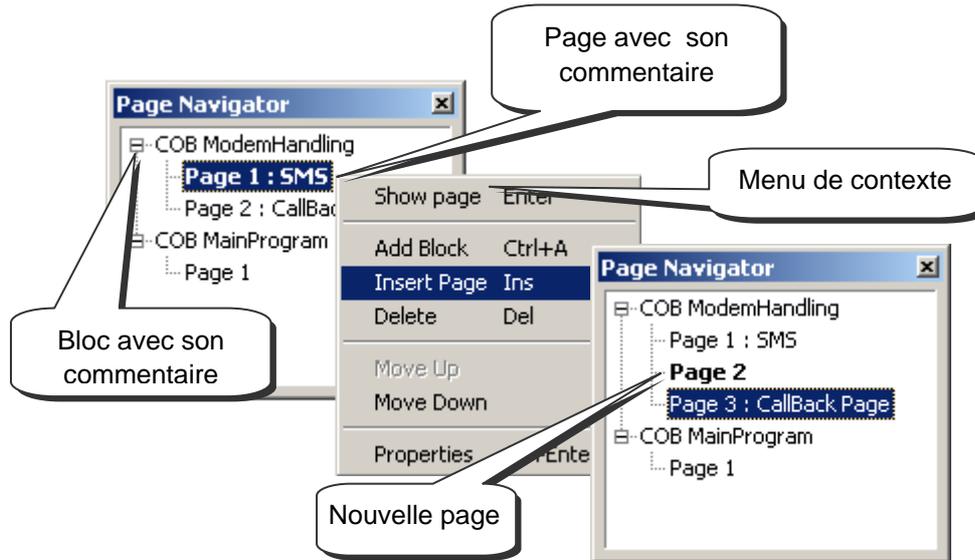
Il n'est pas permis de réaliser des boucles. Il en résulte un message d'erreur : Page 1: Error 55: Loop back detected



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*



6.7 Edition de pages Fupla



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 « Structuration du programme » de ce document.



Show/Hide
Page Navigator

6.7.1 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 Page* ou le menu *Page Insert After (Page Insert Before)*

6.7.2 Supprimer une page

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

6.7.3 Navigation entre pages



Goto Next
Page

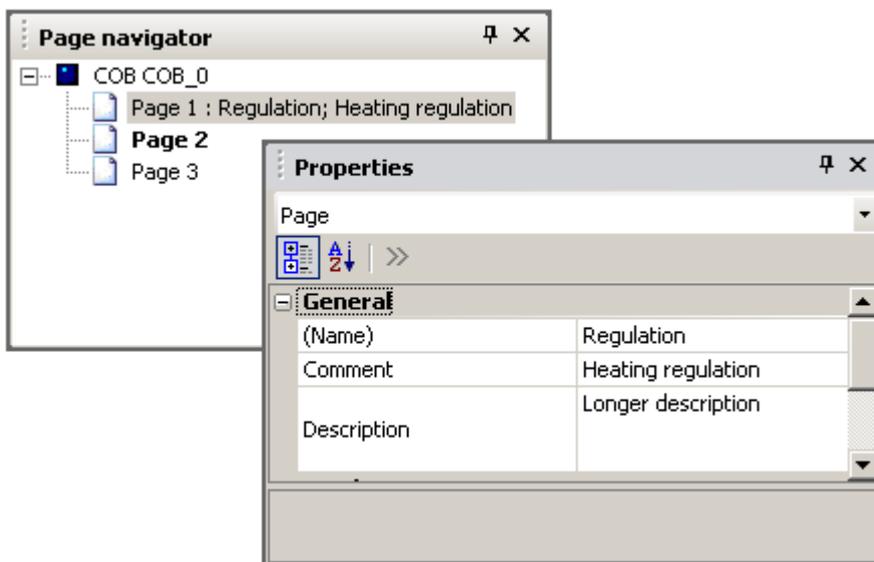
Ouvrir la fenêtre *Page Navigator*, placer la souris sur la page et sélectionner le bouton gauche de la souris avec un double clic.

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.

6.7.4 Page Properties

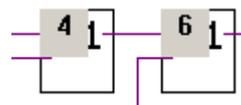
Permet de définir un nom, un commentaire est une description pour chaque page Fupla visibles dans la fenêtre *Page Navigator*. C'est très pratique pour naviguer dans les pages Fupla et faciliter l'entretien du programme.

Pour afficher la page *Properties*, sélectionner la page souhaitée dans le *Page Navigator* puis le menu de contexte *Properties*.



6.7.5 Traitement du programme par l'automate

L'automate traite le programme du coin supérieur gauche de la première page au coin inférieur droite de la dernière page. Pour connaître avec plus de précision l'ordre dans lequel les FBox sont traitées par l'automate, sélectionner le menu *Page, Show FBox priorities*

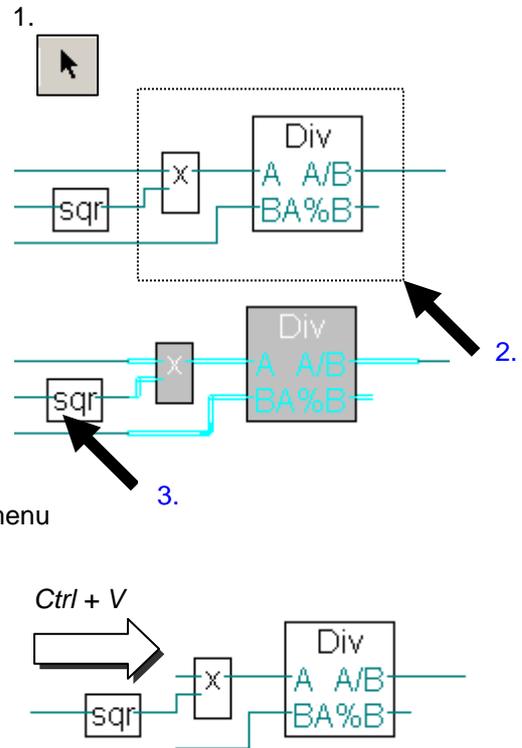


6.8 Copier, coller

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.

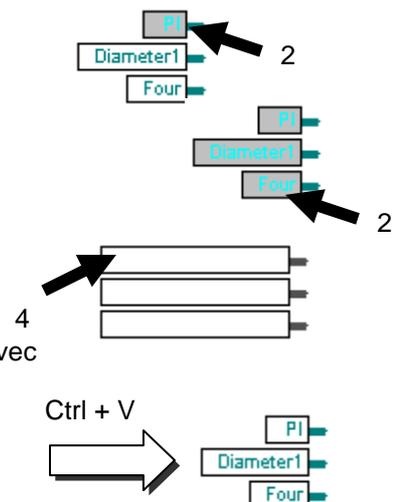
6.8.1 Copier, coller une partie de programme

- Sélectionner le bouton *Select Mode*.
- 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.
- 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.
- Copier la sélection avec le menu *Edit Copy* ou avec les touches *Ctrl + C*.
- Coller une copie de la sélection avec le menu *Edit Past* ou les touches *Ctrl + V*
- Positionner la copie dans la page
Fupla :
 - Placer la souris au centre de la copie.
 - Presser le bouton gauche de la souris.
 - Sans lâcher le bouton, glisser la souris.



6.8.2 Copier, coller des symboles

- Sélectionner le bouton *Select Mode*.
- 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.
- Copier la sélection avec le menu *Edit Copy* ou avec les touches *Ctrl + C*.
- Placer la souris sur un élément libre de la marge
- Coller une copie de la sélection avec le menu *Edit Past* ou les touches *Ctrl + V*.

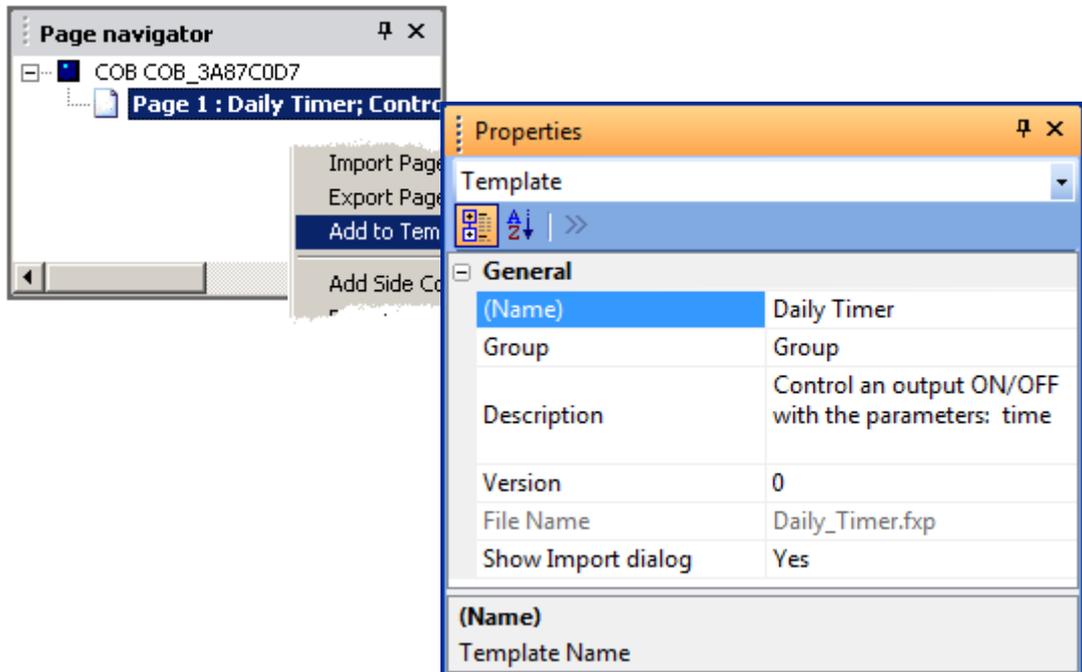


*) La touche *Ctrl* permet de sélectionner des symboles non consécutifs

6.9 Templates

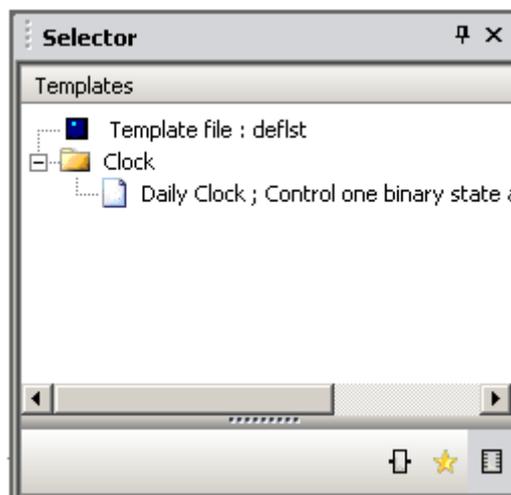
Les séquences de pages Fupla souvent présentes dans les divers programmes sont collectionnées sous la forme de *templates* que nous utilisons comme une librairie de pages dont nous ne modifions que les noms et les adresses des symboles.

6.9.1 Créer un template



La création d'un template est très simple à réaliser. Nous nous servons de la fenêtre *Page Navigator* pour sélectionner une séquence de pages, puis le menu de contexte *Add to templates*. Un dialog nous invite à définir le groupe, le nom et le commentaire du template.

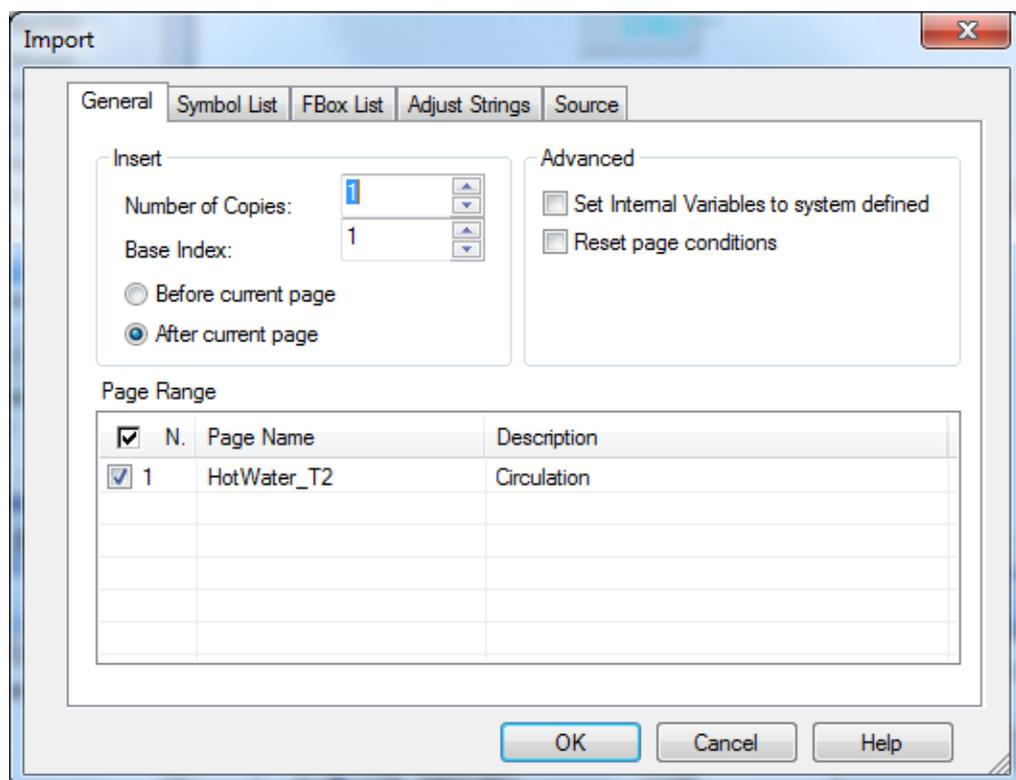
Les templates sont organisées en groupes comparables à des familles de Fboxes. Les groupes rassemblent les templates selon les critères de classification définis par son auteur. Le groupe et le nom définissent la structure d'organisation de la page Template du *Fbox Selector*.



6.9.2 Importation de templates

Les templates sont utilisables depuis tous les projets. Il suffit de sélectionner le template dans la fenêtre *Fbox Selector* et le glisser sur une page Fupla pour insérer la séquence de pages avec les FBoxes, les liens, les symboles, paramètres d'ajustages,... dans le fichier.

Un dialogue est automatiquement affiché, il nous permet d'ajuster le nom et les adresses des symboles importés par le template et les divers autres paramètres. Cette fonctionnalité peut être comparée à une macro ou une fonction avec des paramètres que nous pouvons utiliser comme une librairie.



Copie Number, Base Index

Si plusieurs copies du même template sont nécessaires, définir le nombre de copies et la valeur de base d'un index à placer dans les noms des symboles ou des groupes.

Before/After current page

Importe la séquence de page avant ou après la page affichée par le Fupla.

Set Internal variable to system defined.

Certaines FBoxes disposent de paramètres d'ajustages dont le nom et l'adresse sont définis par l'utilisateur ou par le système. (*Static symbols*)

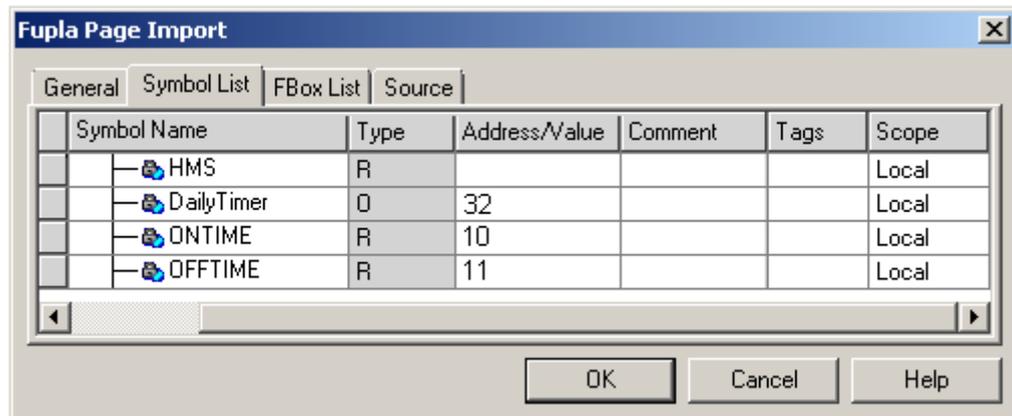
Ce paramètre permet de conserver les adresses et symboles statiques définis par l'auteur ou de restaurer l'adressage dynamique et les symboles par défaut.

Reset pages Conditions

Les propriétés des pages Fupla autorisent la définition de conditions d'exécutions de la pages. Ce paramètre conserver ou supprimer ces conditions à l'importation.

Page Range

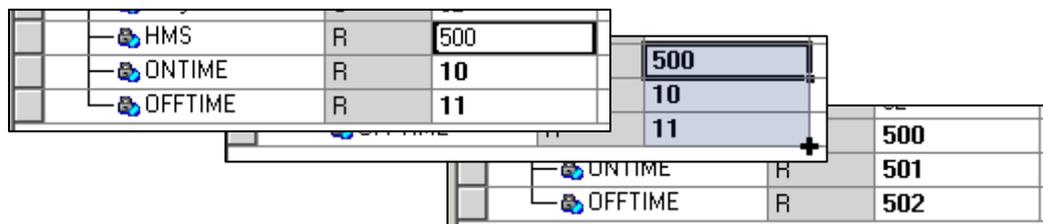
Permet de sélectionner l'importation individuelle des pages ou de toutes la séquence.



Le répertoire *Symbols List* représente tous les symboles importés avec le template. Il permet de redéfinir les symboles avec de nouveaux noms, adresses, commentaires et scopes.

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.



Pour corriger les adresses des symboles, nous les trions par type en sélectionnant l'en-tête de colonne *Type*. Nous éditons l'adresse de la première cellule, puis nous sélectionnons son angle du bas droit pour l'étirer vers le bas.

Lors de l'importation de plusieurs copies du même template, voir paramètre de la page *General*, il est utile d'insérer un index dans les noms des symboles ou des groupes en utilisant le caractère # . Ce caractère est automatiquement remplacé par l'index de base augmenté d'une unité pour chaque copie du template. Il est aussi possible de sélectionner les symboles puis le menu de contexte *Indexing*.

Le répertoire *FBox List* représente la liste de toutes les Fboxes dont les propriétés sont définies avec un nom. Nous pouvons ainsi les modifier ou les indexer avec le caractère #.

6.10 Edition d'un premier programme Fupla

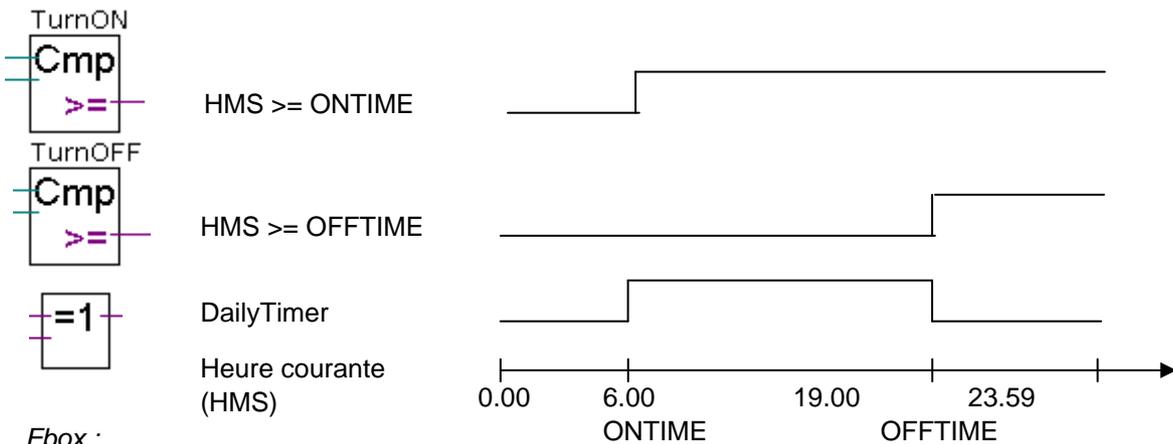
6.10.1 Objectif

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 FBoxes standards.

6.10.2 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 FBoxes les plus élémentaires possibles.

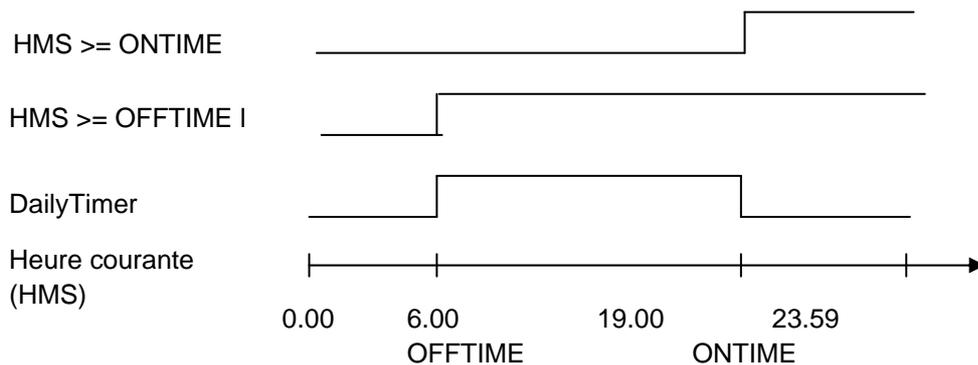
Pour l'exemple de cette horloge, nous proposons de faire deux comparaisons. La première détermine si l'heure courante *HMS*, l'heure que nous lisons à notre montre ou celle de l'automate, est plus grande ou égale à l'heure d'enclenchement *ONTIME* et la seconde détermine si l'heure courante est plus 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 :

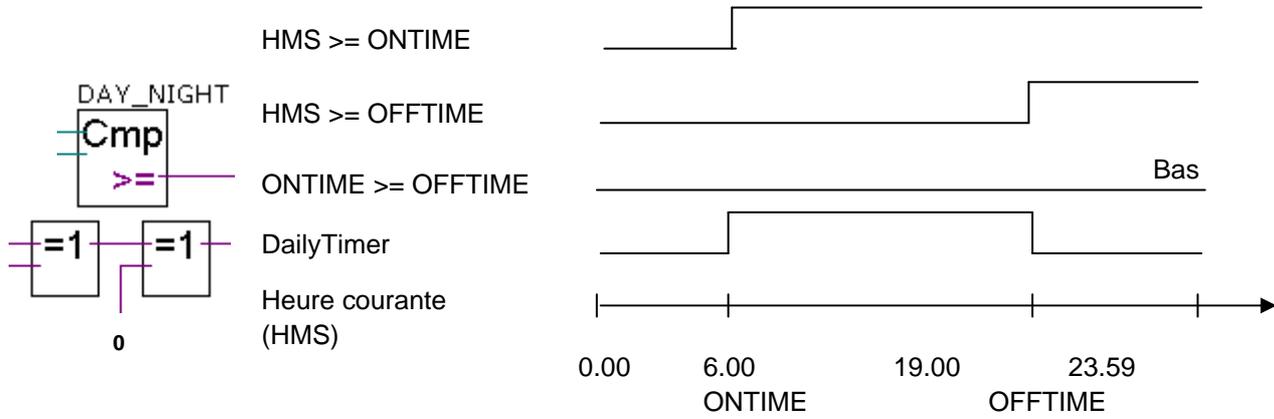
- Entiers, Plus grand ou égal
- Binaire OU exclusif

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 de l'automate est dans l'état inverse de celui souhaité.

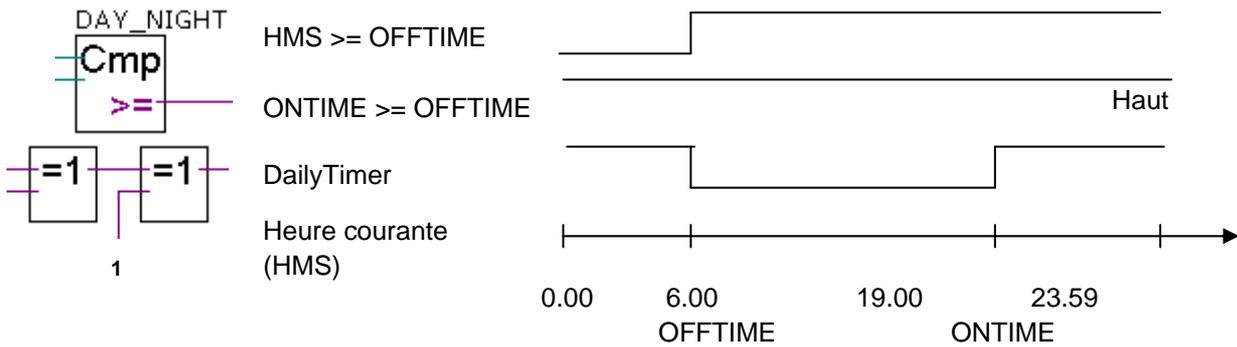


Il est donc nécessaire de compléter notre algorithme en ajoutant une troisième comparaison pour déterminer si l'heure d'enclenchement est plus grande ou égale à l'heure de déclenchement. La solution définitive est alors la suivante.

Sorties active le jour:

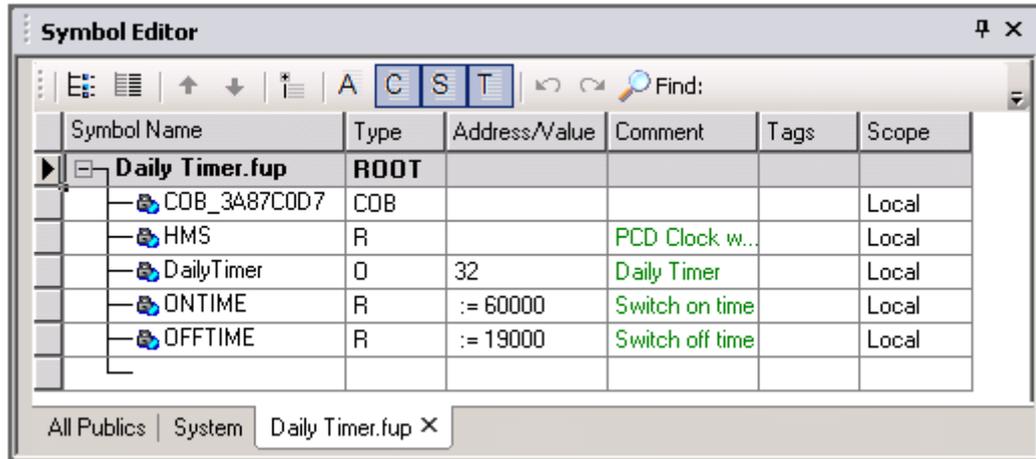


Sorties active la nuit:



6.10.3 Programmation

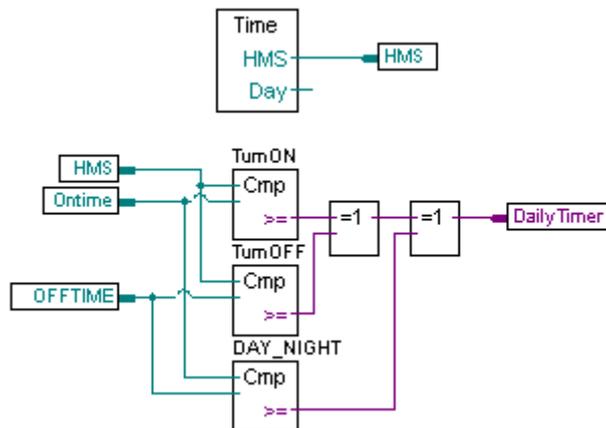
Au début de ce chapitre, nous avons déjà préparé un projet avec un fichier intitulé *DailyTimer.fup*. C'est dans ce fichier que nous allons utiliser pour écrire cet exemple de programme.



Symbol Name	Type	Address/Value	Comment	Tags	Scope
Daily Timer.fup	ROOT				
COB_3A87C0D7	COB				Local
HMS	R		PCD Clock w...		Local
DailyTimer	O	32	Daily Timer		Local
ONTIME	R	:= 60000	Switch on time		Local
OFFTIME	R	:= 19000	Switch off time		Local

Commençons par préparer la liste des symboles. Remarquons que l'heure courante de l'automate 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 l'automate (:=60000 signifie 6 heures 00 minutes 00 secondes.)

Attention, le démarrage à froid de l'automate ne va pas réinitialiser ces registres. Seul le chargement du programme les initialisera à nouveau !



Nous trouvons toutes les FBoxes nécessaires dans la fenêtre *Selector* :

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

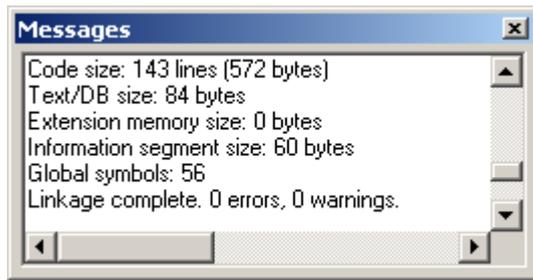
6.11 Construction du programme (Build)



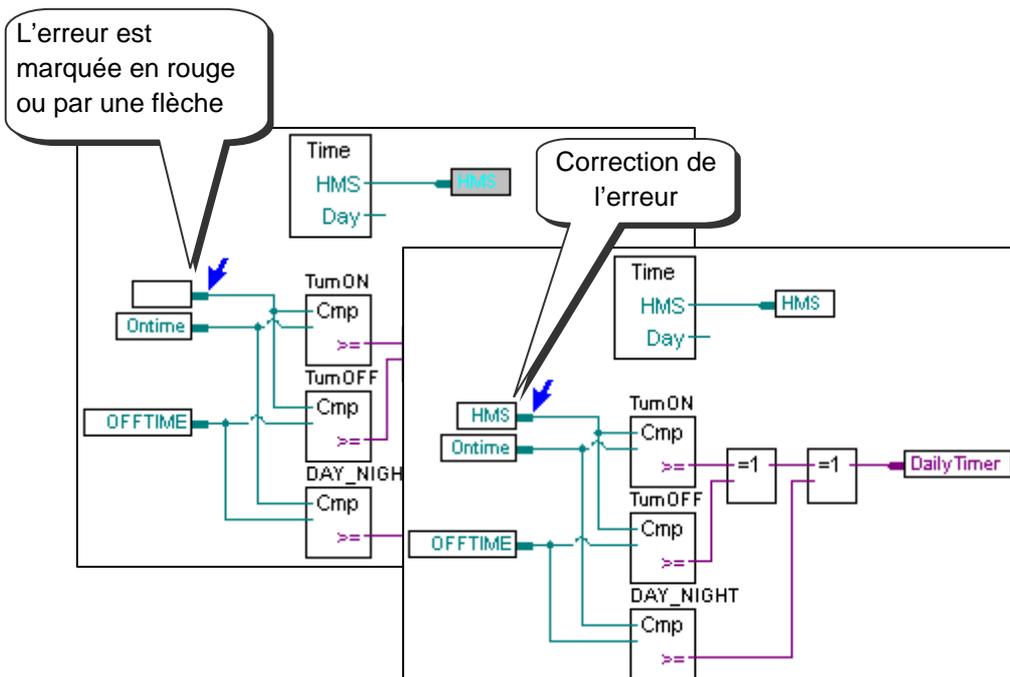
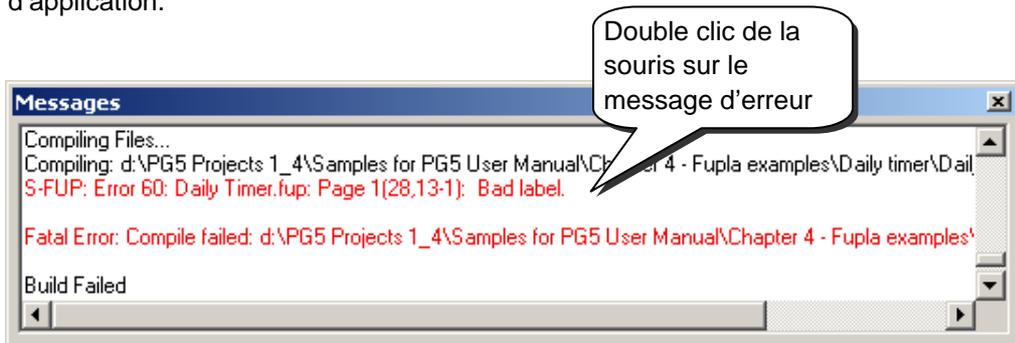
Build All

Le programme utilisateur est complètement décrit, mais pas encore utilisable par l'automate. 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/bouton *Device, Rebuild all* du gestionnaire de projet ou de l'éditeur Fupla.

La fenêtre des *Messages* nous renseigne sur le déroulement de la construction du programme. Nous remarquons 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.



6.12 Charger le programme dans l'automate



Download Program

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

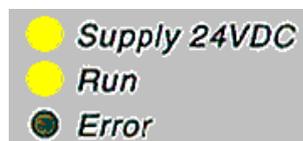
En cas de problème de communication, veuillez à nouveau contrôler les configurations *Settings Online* et *Device Configurator* ainsi que le câble de communication entre l'ordinateur et l'automate. (PCD8.K111, USB)

6.13 Débugger un programme (Debug)

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

6.13.1 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 de l'automate. A la sélection du bouton *Run*, la lampe *RUN* est allumée, l'automate exécute le programme utilisateur.

3. A la sélection du bouton *Stop*, la lampe *RUN* est éteinte, l'automate stop l'exécution du programme.
4. L'automate 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

6.13.2

6.13.3 Point d'arrêt

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

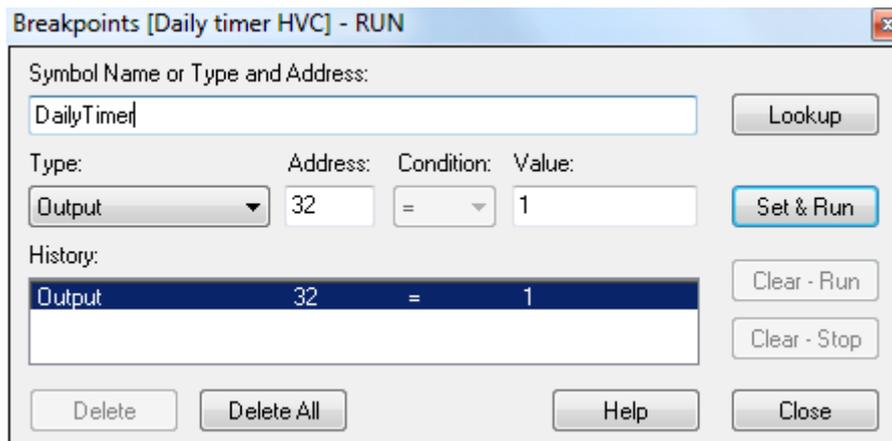
- Etat bas ou haut d'une entrée, sortie, indicateur, indicateur de statut
- Valeur présente dans un registre ou compteur



Set or Clear Breakpoints

Point d'arrêt sur un symbole

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



A l'aide de la fenêtre ci-dessus, définir le type et l'adresse 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 l'automate en mode Run conditionnel. La LED *Run* clignote et le bouton *Run* alterne les couleurs vert et rouge.

L'automate 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 l'automate 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 l'automate en mode RUN. La LED *Run* de l'automate est allumée et le bouton *Run* de couleur verte.

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

Si plusieurs points 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 ou bouton *Online, Run to, FBox* permet de stopper le programme sur la FBox sélectionnée pour en suite continuer en mode pas-à-pas.

6.13.4 Visualiser les symboles ou leurs adresses

Le bouton *Change Symbol/Resource view* 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.



6.13.5 Visualiser l'état des symboles avec Fupla

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

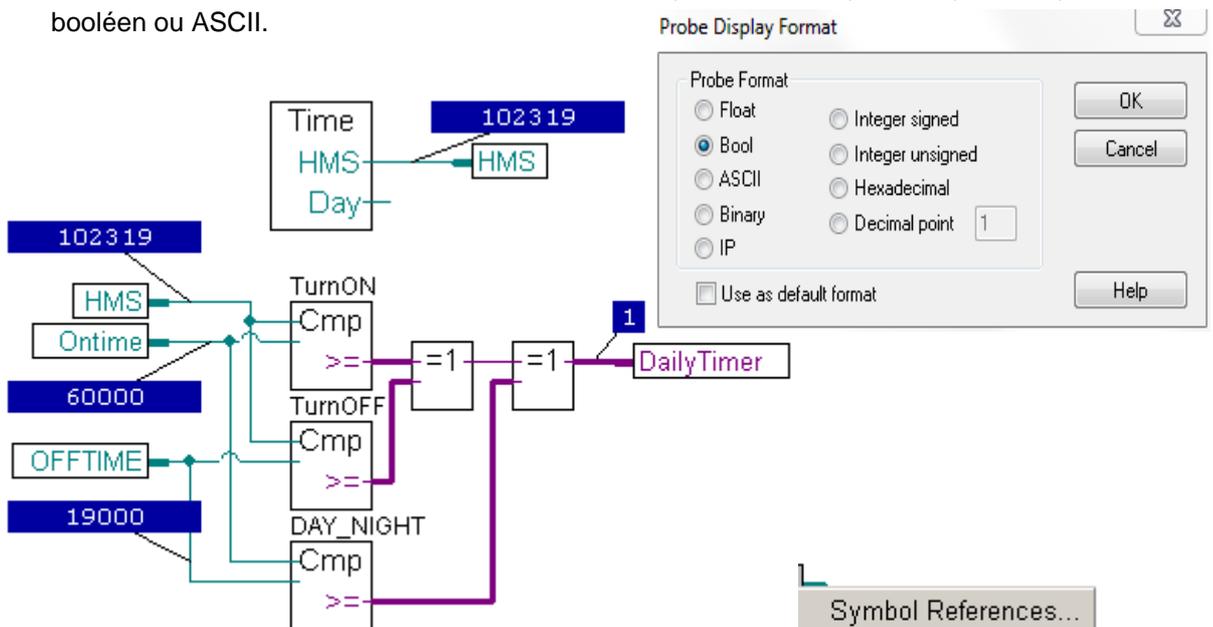
Les états logiques des informations binaires sont représentés par des liens larges ou fins. (large = 1 et fin = 0).



Add Probe

Toutes les autres informations peuvent aussi être représentées par une sonde en sélectionnant le bouton *Add Probe* puis les différents liens 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.



6.13.

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 ou la table des symboles.

Sélectionner un connecteur d'entrée à l'aide de la souris et presser le bouton droit pour afficher le menu de contexte. Le menu de contexte *Edit Data* permet de modifier l'état/valeur du symbole présent dans le connecteur.

6.13.7 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 glissez les symboles de l'éditeur de symboles vers la fenêtre *Watch window*

Placer la souris sur le bouton au début de la ligne, presser le bouton gauche de la souris

Glisser la souris sur le dans la fenêtre Watch window

Symboles avec leurs commentaires et leurs états/valeurs.

Watch Window

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

Daily timer.5sw * [Daily timer]

4. Start/Stop Monitoring

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

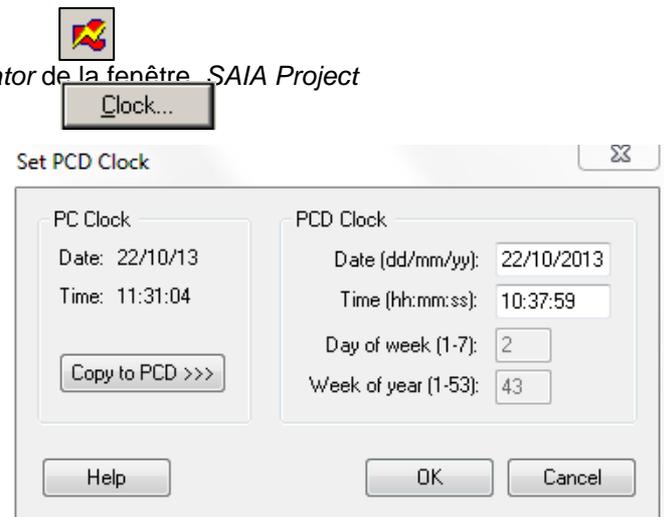
Symbol	Address	Value	Modify Value	Symbol Comment
HMS	R 2113	104852		PCD Clock with current time
DailyTimer	O 32	1		Daily Timer
Ontime	R 2115	60000	43000	Switch on time
OFFTIME	R 2114	19000		Switch off time

Daily timer.5sw * [Daily timer]

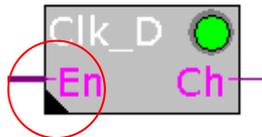
6.13.8 Mise à l'heure de l'horloge de l'automate

A la mise en service, l'horloge interne de l'automate n'est pas toujours à l'heure. Pour l'ajuster 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*.



6.14 Paramètres d'ajustage.



Fbox:CVC Horloges, Horloge journalière

Certaines FBoxes, marquées par un angle noir, supportent une fenêtre de propriétés pour éditer les paramètres d'ajustage.

Ces paramètres sont très utilisés avec la librairie CVC, mais aussi avec toutes les autres librairies de FBoxes.

Ils 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 des propriétés et ses paramètres d'ajustage, placer la souris sur une FBox marquée par un angle noir et cliquer deux fois sur le bouton gauche de la souris ou utiliser le menu de contexte *Properties*.

Edition des paramètres hors ligne.

Adjust Parameters	
Objet pour éditeur HMI	Non
Enclenchement	12:00
Déclenchement	12:00

L'édition hors ligne des paramètres est supportée par la fenêtre des propriétés. Les valeurs des paramètres sélectionnés sont conservées dans le fichier Fupla. Un *build* est *download* du programme sont nécessaires pour que les nouveaux paramètres soient en service dans l'automate.

Ajustage des paramètres en ligne.

Adjust Window				
On Edit Data				
Description	Online Value			Modify Value
<ul style="list-style-type: none"> ▶ HVC-Clocks: Horloge journalière <ul style="list-style-type: none"> ▶ Enclenchement ▶ Déclenchement ▶ Etat du canal 			← →	
			← →	

L'ajustage en ligne des paramètres est supporté par la fenêtre *View, Adjust window* qui ne présente que les paramètres ajustables en ligne et les paramètres de vérification. Les valeurs de paramètres modifiés sont directement ajustés dans la mémoire de l'automate et n'affectent pas le contenu du fichier Fupla.

La fenêtre *Adjust Window* est automatiquement affichée en lieu est place de la fenêtre des propriétés lorsque nous sommes en ligne.

6.14.1 Initialisation des FBox CVC.

Lors que nous utilisons certaines bibliothèques de FBoxes tel que 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 FBoxes après le chargement du programme ou après un démarrage à froid de l'automate (mise sous tension de l'automate)



Fbox: CVC Init,
Initialisation CVC 7

Après un chargement de programme et un démarrage à froid de l'automate, 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.

Reset	
Automatic Reset ...	Activated
Evaluate Reset Input	At startup

Chargement du programme et le paramètre Reset automatique:

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

Avec l'option Désactivé, tous les paramètres déjà présents dans l'automate 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 FBoxes CVC sont initialisés avec les valeurs définies à la programmation.

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

LED vert/rouge

Certaines FBoxes 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 FBoxes en erreur.

Remarque:

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

6.14.2 FBox CVC avec paramètres d'ajustage.

La FBox *Clk_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.

La sortie *Ch* 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.



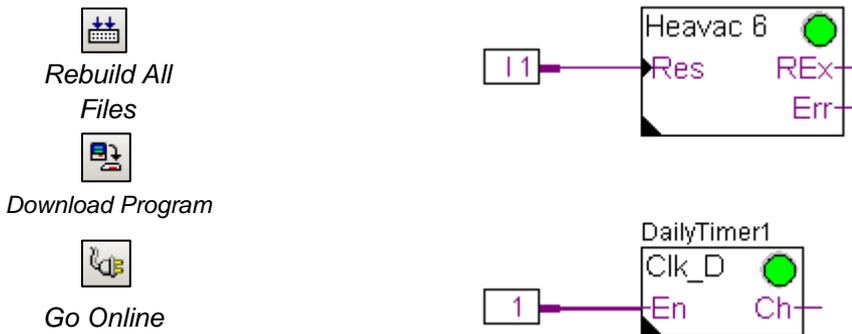
Fbox: CVC Horloges, Horloge journalière

Adjust Parameters	
Objet pour éditeur HMI	Non
Enclenchement	12:00
Déclenchement	12:00

6.14.3 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.

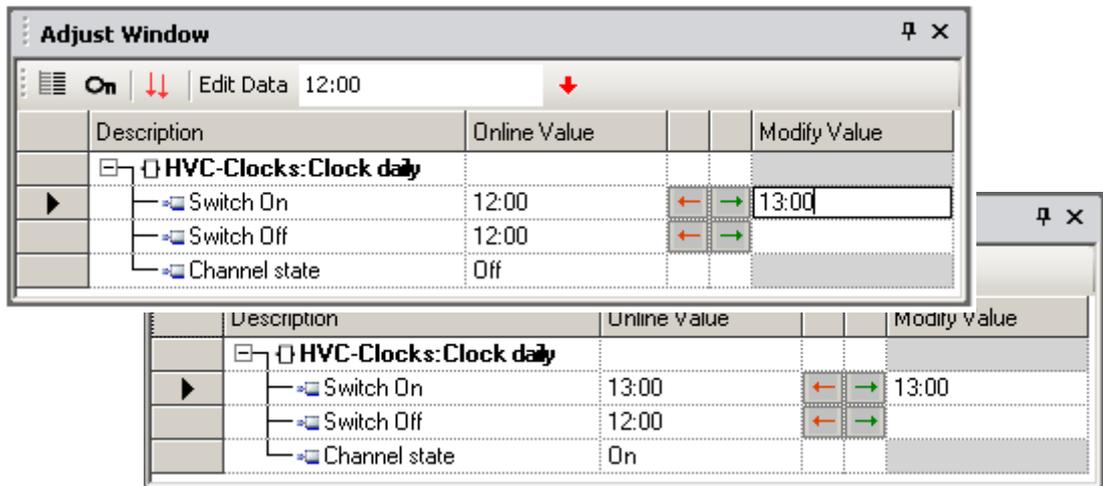
Seul les deux FBoxes présentées précédemment sont nécessaires. Réalisez le programme comme représenté ci-dessous puis réalisez le *Build All*, le *Download Program* et *Go Online*.



Remarque:

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

6.14.4 Ajustage des paramètres en ligne.



Lorsque nous sommes en ligne, l'ajustage des paramètres est supporté par la fenêtre *View, Adjust window*. Elle s'affiche en sélectionnant la Fbox dans la page Fupla et fonctionne comme une feuille *Watch Window* :

La colonne *Description* décrit les paramètres d'ajustage.

La colonne *Online Value* affiche les paramètres en service dans la mémoire de l'automate.

La colonne *Modify Value* permet d'éditer de nouvelles valeurs de paramètres et de les charger individuellement ou simultanément dans l'automate.



Chargement individuel du paramètre.

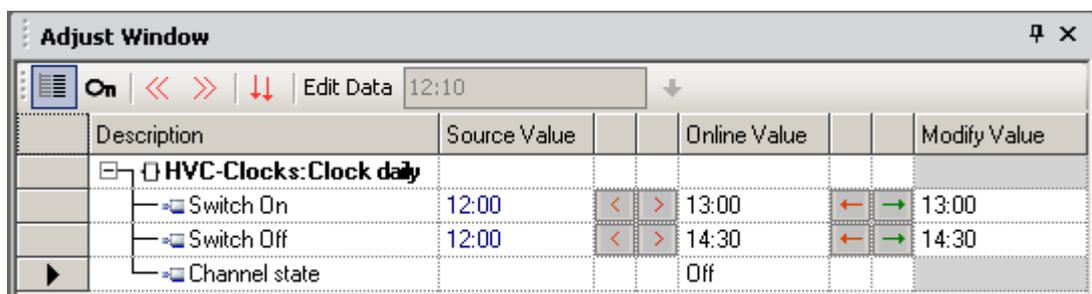


Chargement simultané de tous les paramètres.

Il est aussi possible de sélectionner le paramètre à modifier et de l'éditer dans la cellule *Edit Data* de la tool bar.

Les valeurs de paramètres modifiés sont directement ajustés dans la mémoire de l'automate et n'affectent pas le contenu du fichier Fupla.

6.14.5 Rétablir les paramètres du fichier Fupla.



Après de multiples modifications des paramètres d'ajustage, il est toujours possible de rétablir leur valeur initiales encore présente dans le fichier Fupla.

Le bouton *Show Source Value* affiche la colonne *Source Value* correspondant aux valeurs initiales des paramètres d'ajustages présentes dans le fichier Fupla. Voir la fenêtre des propriétés des Fboxes.



Rétablit individuellement les paramètres.



Rétablit tous les paramètres de la Fbox.

Le menu *Online Write Fboxes Adjust Parameters* rétablit les paramètres de toutes les Fboxes du fichier Fupla..

6.14.6 Sauvegarde des paramètres en ligne dans le fichier Fupla.

Si nous sommes satisfaits des paramètres d'ajustages présents dans la mémoire de l'automate, il est toujours possible de les mettre à jour dans le fichier Fupla.



Mise à jour individuelle des paramètres.



Mise à jour de tous les paramètres de la Fbox.

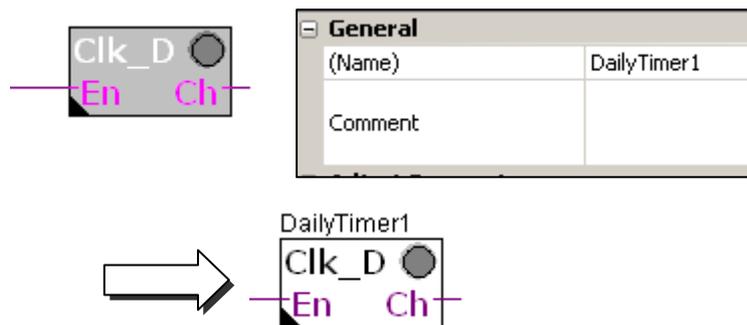
Menu *Online Read Fboxes Adjust Parameters* met à jour le fichier Fupla avec les paramètres de toutes les Fboxes.

6.14.7 Définir les symboles systèmes 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 correspondants aux paramètres affichés dans la fenêtre d’ajustage de la FBox.

Pour définir ces symboles, afficher la fenêtre des propriétés de la Fbox et compléter la section *General* avec un symbole.

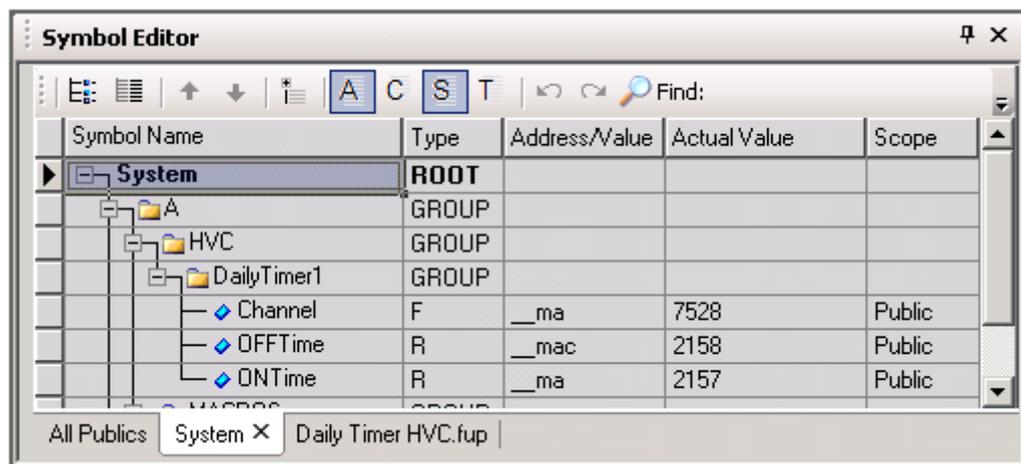


 **Rebuild All Files**

Construire le programme et ouvrir l’éditeur de symboles. Le répertoire System contient la liste des symboles systèmes de l’automate.

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).

 **Show or Hide Symbol Editor**

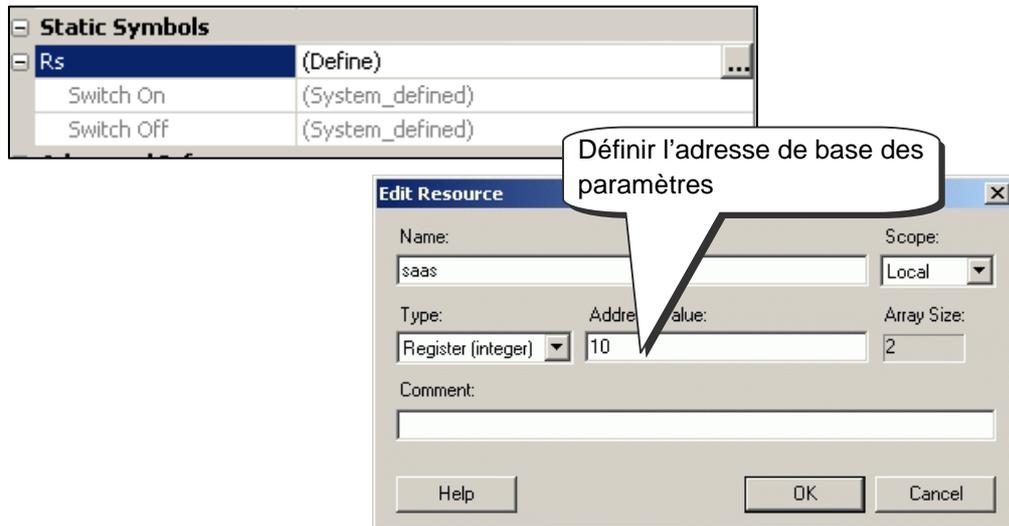


Il ne reste qu’ à utiliser ces nouveaux symboles dans le programme Fupla.



6.14.8 Définir les adresses systèmes des paramètres d'ajustage.

Définir le symbole système des paramètres d'ajustage comme présenté précédemment et ajouter l'adresse à partir de la des propriétés de la Fboxe. Sélectionner la ligne (Define), puis le bouton affiché à la fin de la ligne.



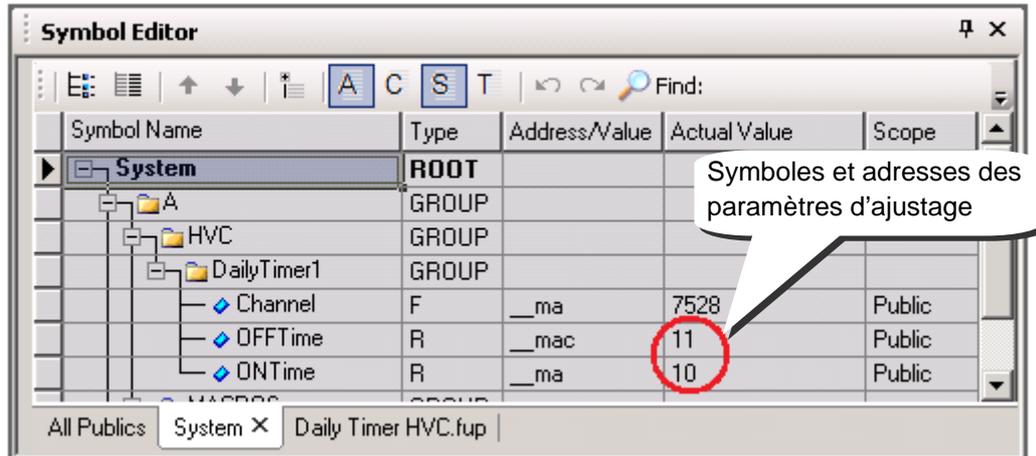
Construire le programme et ouvrir l'éditeur de symboles. Les symboles systèmes sont complétés avec les adresses des registres définis ci-dessus.



Rebuild All Files



Show or Hide Symbol Editor



6.15 Mise en service d'un module analogique

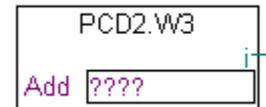
La lecture/écriture d'une grandeur analogique nécessite un petit programme propre à chaque module analogique présent dans l'automate. Il assure le multiplexage des canaux ainsi que la conversion AD/DA. Ce programme est supporté soit à l'aide de Fboxes, soit par le *Device Configurator* et la fonction *media mapping*. Voir description du *Device Configurator*.

6.15.1 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.



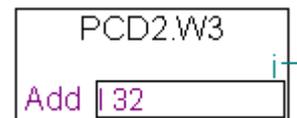
Pour les modules d'entrées et sorties analogiques, il est nécessaire de faire usage d'une FBox d'acquisition de la grandeur analogique. Ces FBoxes sont disponibles avec la librairie *Modules analogiques* ou *CVC-Analogue*.



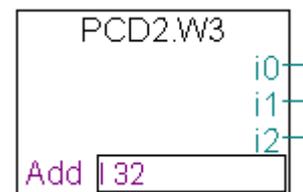
Ces librairies proposent une grande variété de FBoxes, chacune correspond à un module analogique dont le nom présenté dans le *FBox Selector* correspond au numéro d'article du module.

Les FBoxes 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 champ *Add* permet de définir l'adresse de base du module analogique. Cette adresse indique l'emplacement où est inséré le module dans l'automate : 0, 16, 32, ...



Les mesures analogiques sont directement disponibles sur les sorties I 0 à I 7 de la FBox. Il est possible de les lier directement à d'autres FBoxes 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 ou par les étapes et transition du Graftec.



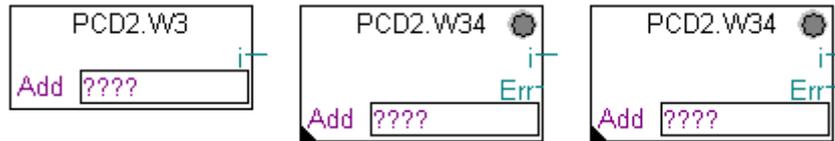
Attention :

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 (255). Si non, la valeur fournie par le module pourrait être erronée.



6.15.2 Exemple pour les entrées analogiques PCD2.W340

Si l'autonmate est équipé d'un module de 8 entrées analogiques universelles PCD2.W340, l'utilisateur peut utiliser l'une des FBoxes 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.

Adjust Parameters	
Configuration channel 0 t	
Ch 0 / Mode or sensor type	mV
Ch 1 / Mode or sensor type	Ni 1000
Ch 2 / Mode or sensor type	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.

Adjust Parameters	
Output when in error	Last value

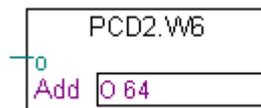
6.15.3 Exemple pour les sorties analogiques PCD2.W610.

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 définit l'adresse de base du module.

Contrairement aux FBoxes pour les entrées, les valeurs de consigne pour les sorties analogiques sont représentées sur le côté gauche de la FBox.

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

Si l'automate 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.

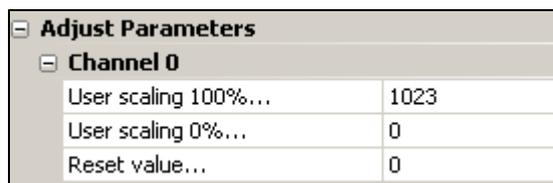


FBox : PCD2.W6

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.

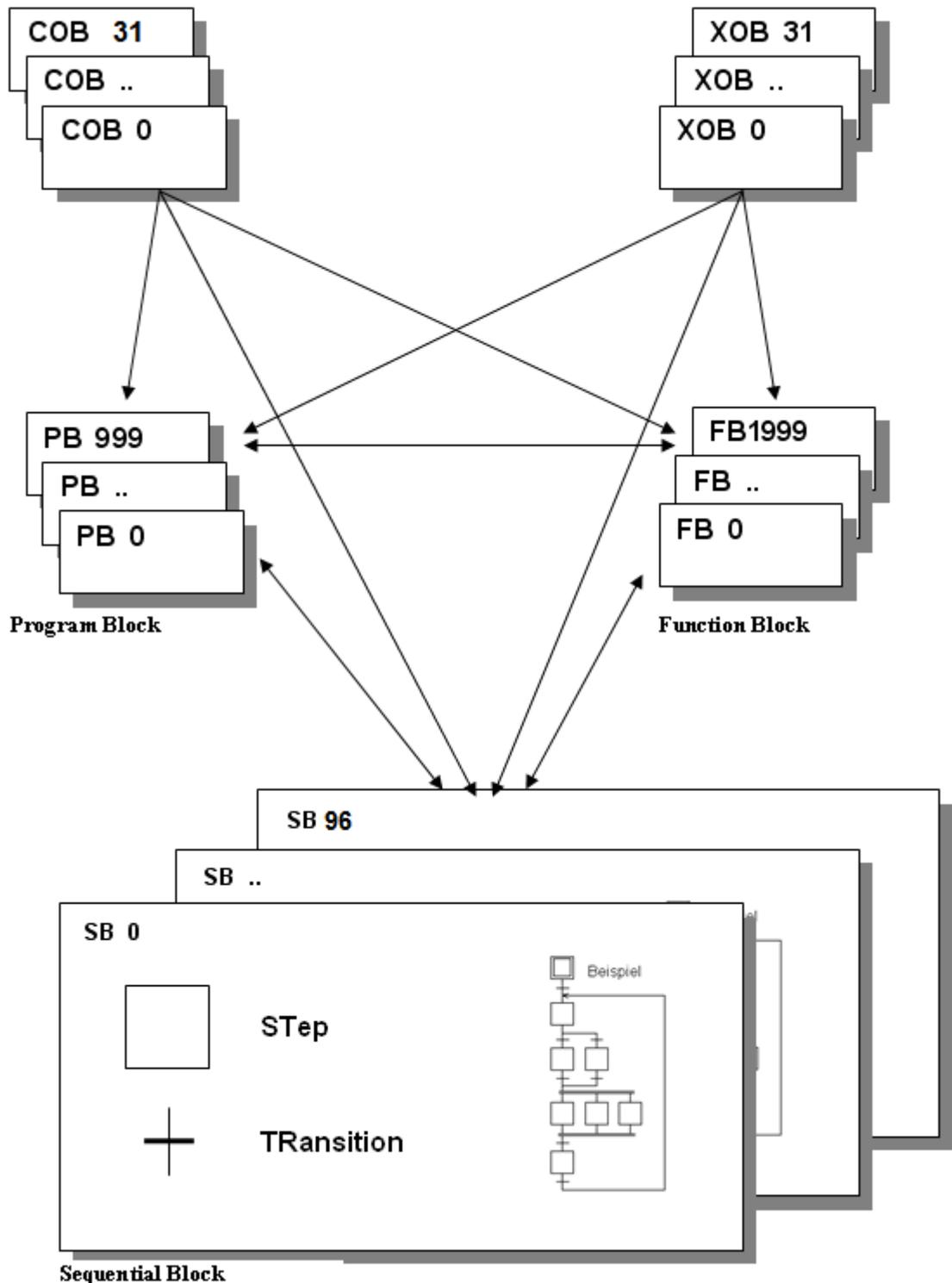
Le paramètre *Reset value*, correspond à la valeur appliquée sur le canal lors de la mise sous tension de l'automate.

Table des matières

7	STRUCTURATION DU PROGRAMME	3
7.1	Blocs d'organisation cyclique (COB 0 à 15, 31)	4
7.1.1	Définition.....	4
7.1.2	Ajouter un bloc dans le programme Fupla.....	5
7.1.3	Exemple.....	5
7.2	Bloc de programmes (PB 0 à 299,999) de fonctions (FB 0 à 999,1999)	6
7.2.1	Définition.....	6
7.2.2	Bloc programme avec appel conditionnel	7
7.2.3	Bloc de fonction avec des paramètres.....	9
7.3	Affichage de la structure des blocs.....	11
7.4	Blocs d'exceptions (XOB 0 à 31).....	12
7.4.1	Définition.....	12
7.4.2	Aperçu des XOB de la famille PCD	13
7.4.3	Application des XOB.....	14
7.4.4	Historique	17
7.4.5	Description des XOBs	18
7.5	Blocs séquentiels (SB 0 à 31, 96).....	22
7.6	Tableau récapitulatif.	22

7 Structuration du programme

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)



7.1 Blocs d'organisation cyclique (COB 0 à 31)

7.1.1 Définition

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

Au démarrage, l'automate commence par exécuter le COB 0, puis l'un après l'autre, les COBs 1 à 31, 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 programme doit compter 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*.

Si le programme est écrit en liste d'instructions, le bloc est délimité par l'instruction COB, suivie du numéro de bloc et se termine par l'instruction ECOB (End of COB). Le programme est inséré entre ces deux limites. Au début du bloc, l'ACCU est toujours *High*.

L'instruction COB possède deux opérandes. La première est le numéro du COB, la seconde le temps de supervision du *watchdog software*. Un temps égal à 0 signifie que le chien de garde software est inactif. S'il est activé avec 10 unités, 100 millisecondes sont accordées à l'automate pour traiter ce bloc. Si ce temps est dépassé, le XOB 11 est appelé.

Si 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 a été dépassé et redémarré avec un nouveau délai de supervision là où il a été interrompu.

Si 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 a été dépassé et redémarré avec un nouveau délai de supervision là où il a été interrompu. La lampe erreur n'est pas allumée, car le défaut a été prévu et pris en charge par le programme utilisateur.

Pour les programmes Fupla, le *watchdog software* est configurés à partir du menu *Block, Properties*.

Bon à savoir, chaque COB possède également son propre registre d'index.

7.1.2 Ajouter un bloc dans le programme Fupla

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

General	
(Name)	COB_3A87C0D7
Type	COB
Comment	
Number	
Scope	File
COB Supervision Time	0

Le menu *Block, Propriétés* affiche la fenêtre ci-contre:

Name:	Nom de symbole du bloc
Type:	Défini le type de bloc : COB, PB, FB, XOB
Comment:	Commentaire libre
Number:	Numéro de bloc. Par exemple COB 0 à 31, PB 0 à 999, ... Par défaut ce numéro du bloc est dynamique excepté pour les XOB. Dynamique signifie défini automatiquement lors du build. Si nécessaire nous pouvons le définir nous même.
Scope :	Scope du symbole de bloc (Local ou Global). Global si le symbole est employé à l'extérieur du fichier.
COB Supervision Time:	<i>Watchdog Software</i> en 1/100 secondes

7.1.3 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 31.

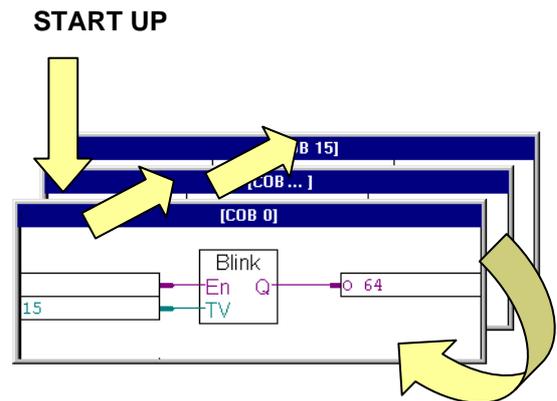
Programme IL

```

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
    
```

Programme Fupla



Fbox: *Clignoteurs, Clignoteur 1 temps*

7.2 Bloc de programmes (PB 0 à 999) de fonctions (FB 0 à 1999)

7.2.1 Définition

Le langage de programmation permet également de travailler avec des blocs de programmes (PB) et de fonctions (FB). Chaque automate dispose des PB 0 à 999 et FB 0 à 1999. Ils offrent un bon moyen d'organiser la hiérarchie des programmes.

La seule différence entre les PB et FB 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.

Précisons que les PB et FB ne sont traités que sur l'appel d'un COB, PB, FB ou d'un bloc séquentiel SB.

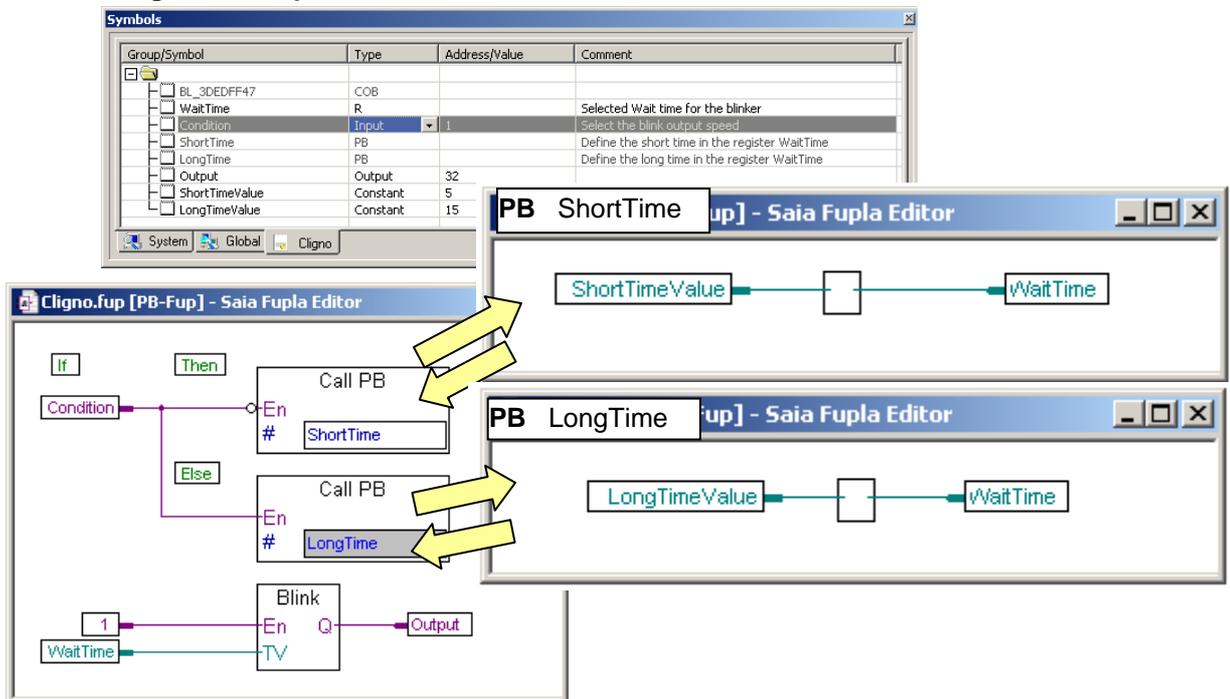
Nous distinguons deux types d'appels: conditionnel et inconditionnel. Le premier dépend du résultat d'une opération logique, le second ne dépend d'aucune condition. Il est possible d'appeler un même PB/FB plusieurs fois dans le programme.

Un PB/FB peut aussi appeler un deuxième PB/FB, qui lui appelle un troisième et ainsi de suite jusqu'à 31 niveaux d'imbrication maximum. Au-delà de 31 niveaux d'imbrication, le PCD fait appel à un XOB 10 (Procédure d'erreur).

7.2.2 Bloc programme avec appel conditionnel

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.

Si les programmes sont rédigés avec l'éditeur Fupla, celui-ci crée un nouveau bloc avec le menu *Block*, *New*, puis éventuellement compléter les paramètres de la vue *Block*, *Properties*.

Les Fboxes pour appeler les blocs sont disponibles dans le menu Fupla *View*, *Selector Windows*, famille *Block Call*

Programme IL:

```

;Two-speed Blinker
LongTime      EQU    PB 1
ShortTime     EQU    PB 2
ShortTimeValue EQU    K 5      ;0,5s
LongTimeValue EQU    K 15     ;1,5s
Condition     EQU    I 1
Output        EQU    O 32
WaitTime      EQU    T

          COB    0
          0

          STH    Condition      ;IF Condition = High)
          CPB    L ShortTime    ; THEN Call PB ShortTime
          CPB    H LongTime     ; ELSE Call PB LongTime
          ECOB

          PB     ShortTime
          STL    WaitTime      ;IF WaitTime = Low
          LD     WaitTime      ; load it with a short value
          ShortTimeValue
          COM    Output        ; Invert the output
          EPB

          PB     LongTime
          STL    WaitTime      ;IF WaitTime = Low
          LD     WaitTime      ; load it with a long value
          LongTimeValue
          COM    Output        ; Invert the output
          EPB

```

Si le programme est écrit en liste d'instructions, le début d'un bloc est marqué par l'instruction PB ou FB et le numéro du bloc comme opérande. La fin du bloc est définie par l'instruction EPB ou EFB. Le programme est inséré entre ces deux limites.

Au début du bloc, l'*ACCU* est toujours *High*. Lorsqu'un PB/FB est appelé, le contenu de l'*ACCU* est sauvé automatiquement, puis repositionné dans son état initial lorsque le bloc est exécuté. Cette méthode permet la programmation de structures «IF...THEN...ELSE...» où les blocs sont appelés conditionnellement.

Important, les flags arithmétiques (Erreur, Négatif, Positif et Zéro) ne sont pas sauvés et restaurés quand un PB ou FB est appelé, seul l'*ACCU* est sauvé. Si les flags arithmétiques doivent être sauvegardés, ils doivent être copiés sur des flags. Comme pour les COBs, la programmation de boucles d'attente n'est pas autorisée dans un PB ou FB. Les sauts (jumps) vers un autre bloc (COB, PB, FB, sont interdits. En règle générale, aucun saut ne doit être utilisé.

7.2.3 Bloc de fonction avec des paramètres

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

STL     = tempo    ;Si temporisation est à l'état bas
LDL     = tempo    ;initialise la tempo avec le
        = delay    ; paramètre 3
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

```

La seule différence entre les PB et FB réside dans la possibilité d'appeler les FB avec des paramètres.

L'appel du bloc FB est suivi par la liste des paramètres. Le premier paramètre dans la liste est le paramètre numéro 1, le suivant est le numéro 2 etc. La liste peut comporter un maximum de 255 paramètres.

A l'intérieur du bloc les numéros des paramètres sont optionnellement définis par un symbole de portée limitée au bloc.

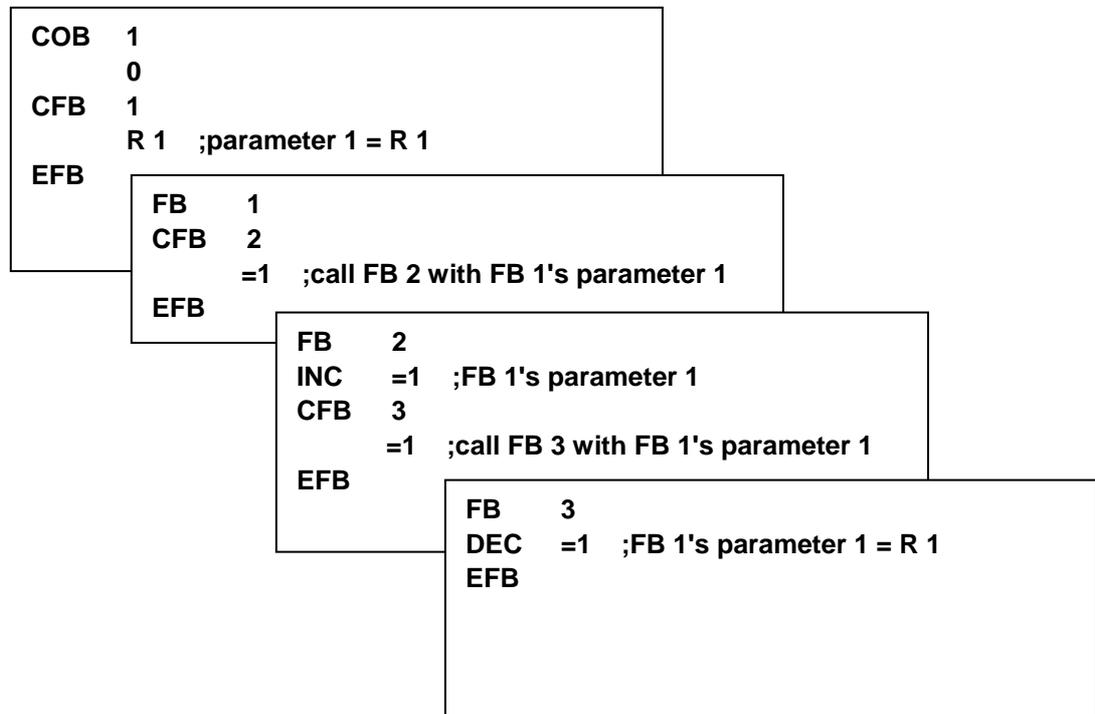
Les instructions font usage des symboles correspondant aux paramètres ou du numéro de paramètre précédé du caractère '='.

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.

Remarque, les FB avec paramètres ne peuvent être utilisés qu'à partir d'un programme en liste d'instructions.

Toutes les instructions n'autorisent pas l'utilisation de paramètres comme opérandes. Une attention toute particulière doit être portée sur l'instruction LD (Load). Une constante de 32-bits sera transférée en utilisant les 2 instructions LDL et LDH (Transferts de 16-bits). Pour plus d'informations, se référer au guide des instructions ou l'aide de l'éditeur IL.

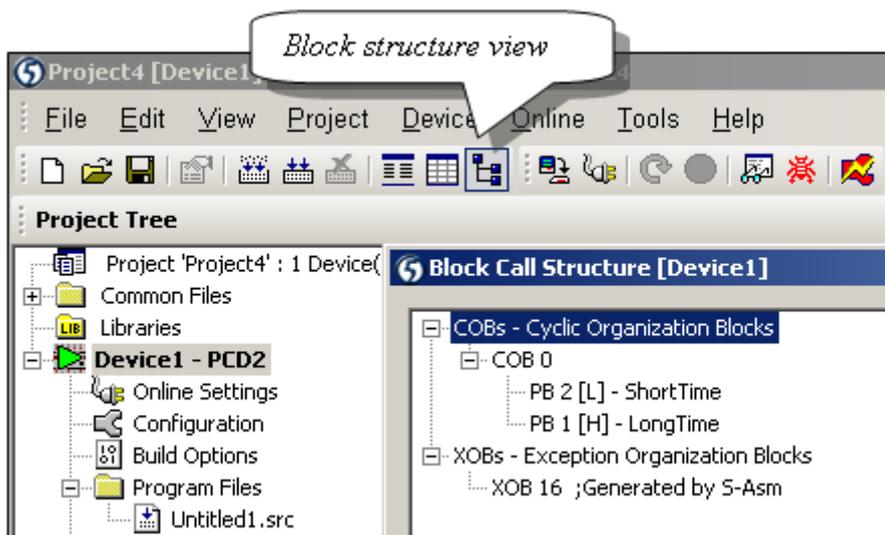
Lors d'appel de bloc FB imbriqués les un dans les autres, il est possible de transférer les paramètres du niveau supérieur vers le niveau inférieur comme suit:



7.3 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.



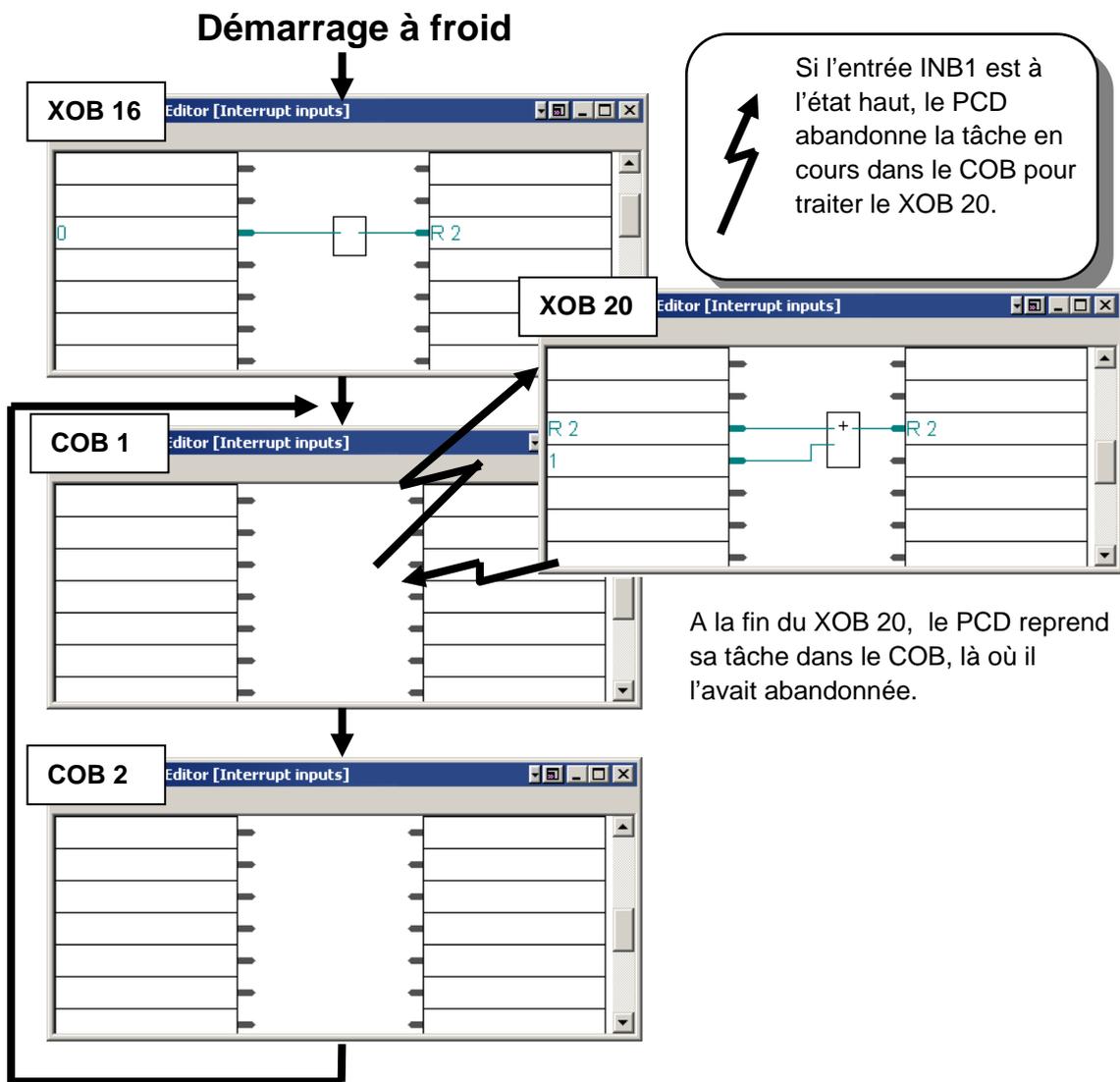
7.4 Blocs d'exceptions (XOB 0 à 31)

7.4.1 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.

7.4.2 Aperçu des XOB de la famille PCD

XOB	Description	Priorité
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
3	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
21	Entrée interruptive	3
22	Entrée interruptive	3
23	Entrée interruptive	3
25	Entrée interruptive « INB2 »	3
26	Interruption cyclique	2
27	Interruption cyclique	2
28	Interruption cyclique	2
29	Interruption cyclique	2
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 appelé. Après l'exécution de la routine d'exception, le programme retourne à l'endroit d'où elle a été 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.

Seul les XOB 0 et 8 peuvent interrompre l'exécution d'un autre XOB. Si une exception de niveau 2 survient pendant l'exécution d'un XOB de priorité moindre, celle-ci sera traitée directement après l'exécution du XOB courant. Les instructions pour utiliser les XOBs dans le diagnostic d'erreur, ainsi que les techniques de programmation des XOBs et leur incorporation dans le programme utilisateur sont traitées dans les paragraphes suivants. Il n'est pas possible d'appeler un XOB à partir du programme utilisateur.

7.4.3 Application des XOB

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

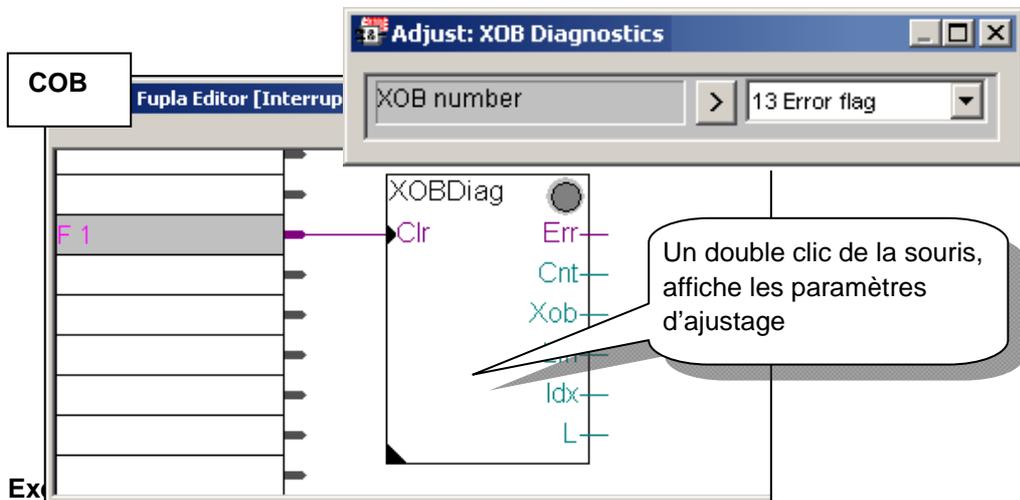
- erreur d'adressage des cartes I/O
- plus de 7 niveaux de programme
- plus de 32 transitions Grafcet actives
- boucle de programme
- erreur mathématique
- erreur de communication

Exemple Fupla:

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

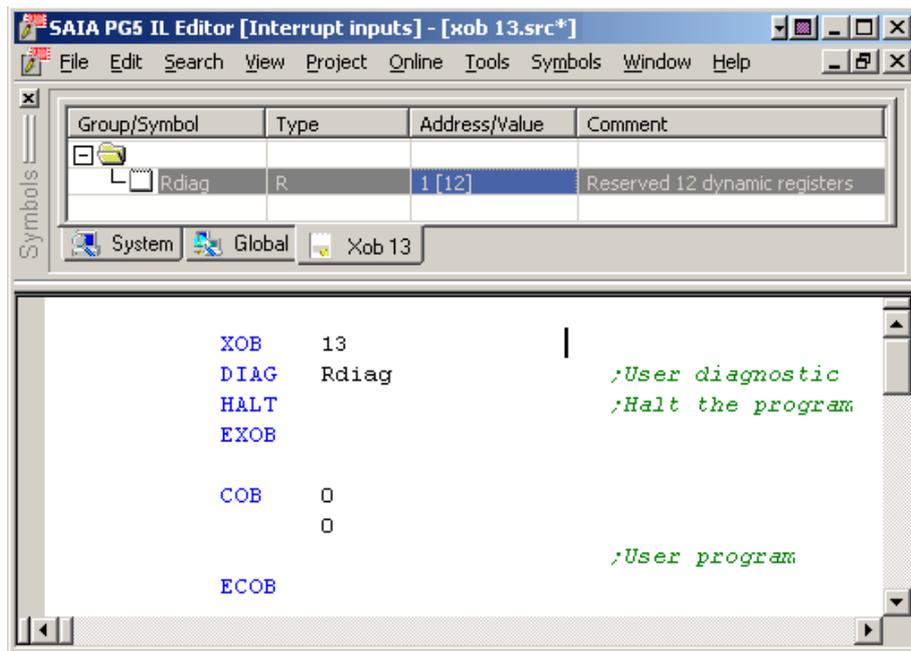
Avec Fupla, il n'est même pas nécessaire de créer les blocs XOB, il sont automatiquement ajoutés par la Fbox : *Spécial, Diagnostique XOB*

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



Ex

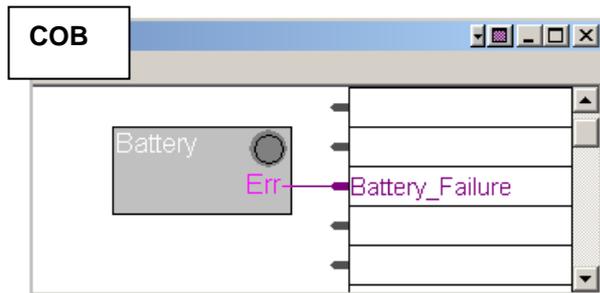
Avec le programme IL, le diagnostic fourni les mêmes informations que ci-dessus dans les registres Rdiag + 0 ... +12



- Surveillance des piles (à remplacer tous les 3 à 4 ans)

Exemple Fupla:

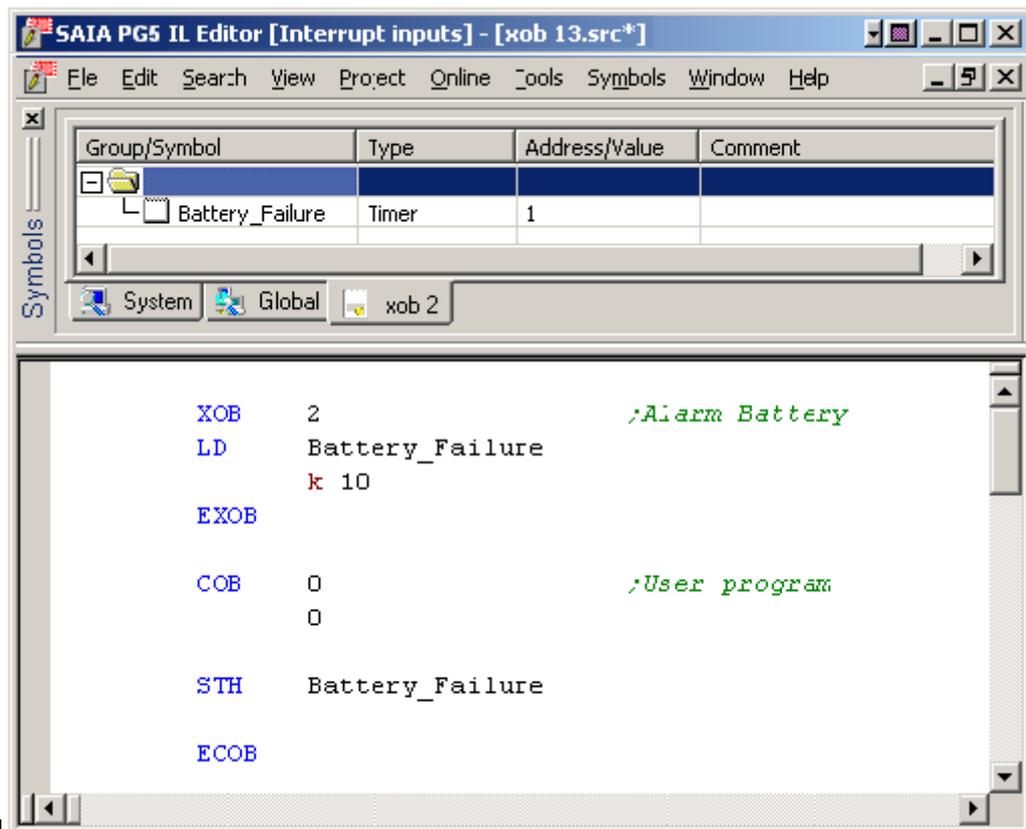
Avec Fupla, il n'est même pas nécessaire de créer le bloc XOB 2, il est automatiquement ajouté par la Fbox: *Spécial, Batterie*
 La sortie *Battery_Failure* est à l'état haut en cas de problème de batterie.



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, environs 1 seconde après avoir effectué le remplacement de la batterie.

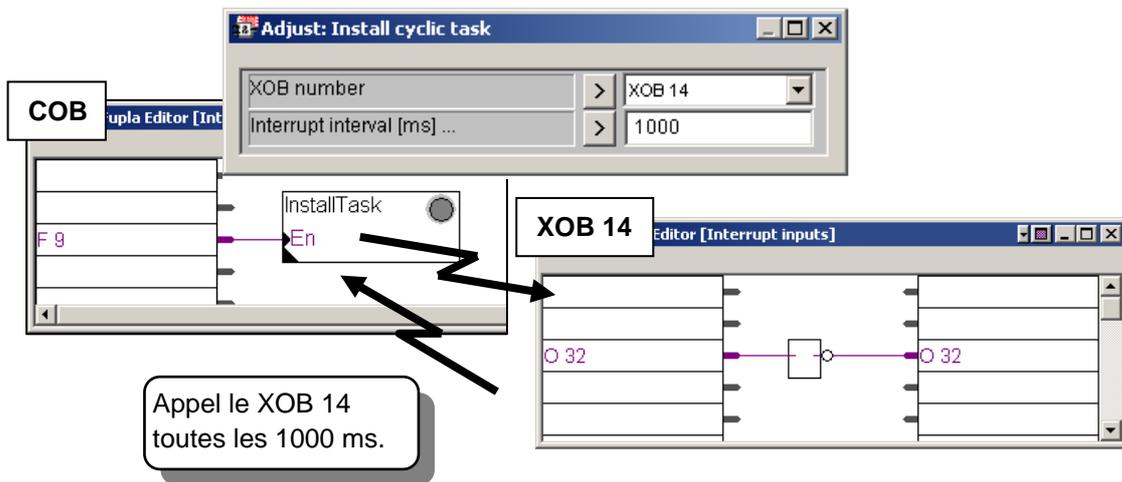


Si événements externes:

- 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

```

7.4.4 Historique

La fenêtre *History* protocoie 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

7.4.5 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. Toutes les sorties sont déclenchées, 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

(Power failure in extension rack)

Le régulateur de tension du module d'alimentation d'un rack d'extension 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 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 de l'automate, 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

(Parity error on address bus)

Le XOB 4 est seulement invoqué par un PCD 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

(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. Ce mécanisme n'est pas implémenté sur les PCD1/2/3.

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 inférieur 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 diagnostique 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. à 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. Le XOB 16 possède son propre registre d'index, qui n'influence pas le registre d'index des autres COBs.

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, 21, 22, 23, 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 Saia PCD respectifs).

XOB 25, 26, 27, 28, 29 : XOBs d'interruption cyclique

Les XOB 25,...29 sont appelés périodiquement avec une fréquence de 10 ms. à 1.000 s. Cette fréquence peut être définie par une instruction SYSWR.

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.

7.5 Blocs séquentiels (SB 0 à 96)

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.

7.6 Tableau récapitulatif.

Service	Média	Opérande	Remarque
Blocs d'Organisation Cycliques	COB	0...31	Minimum 1 COB par programme
Bloc Programme	PB	0...999	Sous programmes appelé par un COB, PB,FB,SB ou XOB
Bloc de Fonction	FB	0...1999	Fonction paramétrisable appelée par un COB, PB,FB,SB ou XOB
Bloc Séquentiel	SB	0...95	Sous programmes séquentiels appelé par un COB, PB ou FB (SB, XOB)
Step	ST	0...5999	
Transition	TR	0...5999	

Table des matières

8	PROGRAMMER EN GRAFTEC	3
8.1	Blocs séquentiels (SB 0 à 95).....	4
8.2	Structure des blocs séquentiels SB	4
8.2	Structure des blocs séquentiels SB	5
8.2.1	Règles d'évolution étapes, transitions.....	5
8.2.2	Transitions (TR 0 à 5999).....	6
8.2.3	Étapes (ST 0 à 5999)	7
8.2.4	Propriétés d'une étape ou transition.....	8
8.2.1	Séquences Graftec typiques.....	9
8.3	Edition d'un projet Graftec	10
8.3.1	Créer un nouveau projet	10
8.3.2	Ajouter un fichier Fupla ou IL.....	10
8.3.3	Appel du SB à partir d'un COB.....	11
8.3.4	Ajouter un fichier Graftec.....	11
8.3.5	Page Navigator, ajouter un bloc SB.....	12
8.4	Editer une structure séquentielle	13
8.4.1	Editer une séquence simple	13
8.4.2	Editer une liaison.....	13
8.4.3	Option smart cursor	13
8.4.4	Editer une divergence alternative (OU).....	14
8.4.5	Editer les liens de convergence.....	14
8.4.6	Editer une divergence simultanée. (ET)	14
8.4.7	Editer les liens de convergence.....	14
8.4.8	Ajouter un commentaire	14
8.4.9	Insérer une séquence.....	15
8.4.10	Effacer une séquence	15
8.4.11	Copier-coller de séquences	16
8.5	Ecriture d'un premier bloc séquentiel.....	17
8.5.1	Réalisation de la structure de base.....	17
8.5.2	Choix de l'éditeur IL ou Fupla	18
8.5.3	Edition des symboles	19
8.5.4	Programme d'une étape, charger un compteur	19
8.5.5	Programme d'une transition, attendre un signal de départ.....	19
8.5.6	Activer une sortie et démarrer une temporisation.....	20
8.5.7	Attendre la fin de temporisation.	20
8.5.8	Désactiver une sortie et attendre la fin de temporisation	20
8.5.9	Décrémentation d'un compteur	21
8.5.10	Divergence alternative.....	21
8.6	Construction et débogage du programme (Build)	22
8.6.1	Fenêtre des messages.....	22
8.6.2	Outils en ligne.....	22
8.7	Structuration du Graftec en pages.....	23
8.7.1	Règles pour éditer des pages.....	23

8.7.2	Edition d'une nouvelle page	24
8.7.3	Ouverture de pages	24
8.7.4	Suppression des pages	24
8.7.5	Block Navigator.....	25
8.8	Graftec templates.....	26
8.8.1	Créer un template	26
8.8.2	Importation de templates	27

8 Programmer en Graftec

Le Graftec est dérivé du standard Grafcet décrit dans la norme française NF C-03-190 et dans IEC 848. Mais ne suit pas exactement le standard, il contient certaines restrictions et améliorations et nous l'appelons Graftec.

Le Grafcet est indépendant de la technologie utilisée pour son implémentation. Il standardise la représentation du fonctionnement des processus séquentiels à l'aide d'un nombre limité de symboles graphiques régit par quelques règles simples.

Le Grafcet est constitué d'étapes qui définissent les actions sur le processus et de transitions qui contrôlent son évolution.

Une séquence consiste en une succession alternée d'étapes et de transitions. Une étape ne peut être évaluée que si la transition précédente le permet.

L'éditeur Graftec crée toutes les instructions nécessaires pour décrire le diagramme fonctionnel dans un Bloc Séquentiel SB selon les directives Grafcet.

Une application Graftec est programmée en deux étapes:

- La création du diagramme décrivant le déroulement séquentiel du programme.
- La programmation des étapes et transitions avec l'éditeur Fupla ou IL.

Après le *build* et le *download* du programme dans l'automate, il est possible d'observer le fonctionnement du programme lorsqu'il s'exécute. Ceci offre une aide considérable lors de sa mise en service.

Graftec offre la possibilité de diviser une grande structure en pages plus petites. L'enchaînement des pages offre alors un effet de zoom permettant de représenter son contenu et sa progression selon le niveau de détails désirés.

L'exécution d'un programme Graftec est purement séquentielle et suit les règles du Grafcet. Le résultat est une vitesse d'exécution optimale avec un temps de réaction extrêmement court. Même dans un programme avec de nombreuses étapes et transitions, seul les transitions actives sont exécutées. Le temps d'exécution des autres parties cycliques du programme n'est pas influencés, même par un grand programme Graftec.

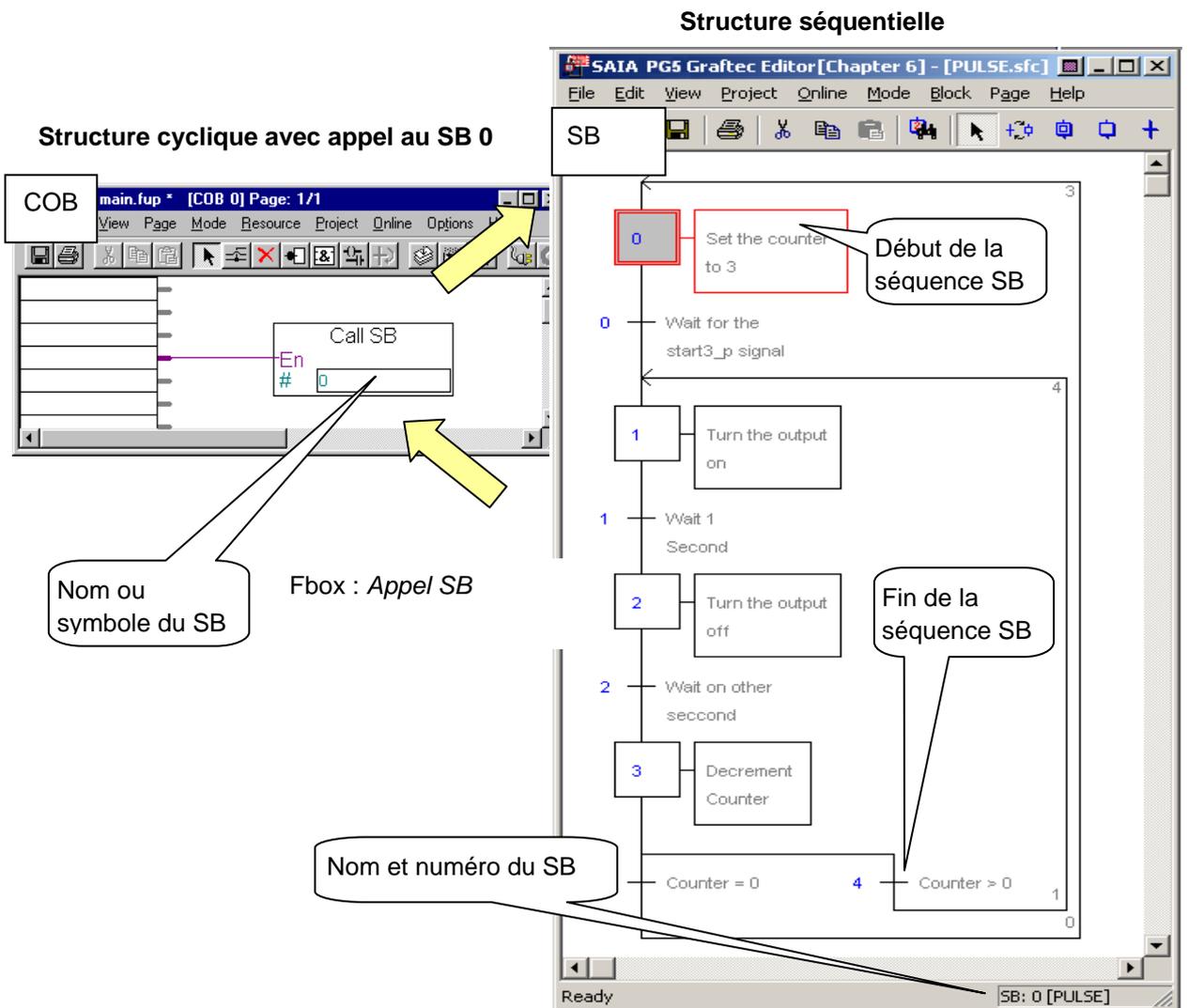
8.1 Blocs séquentiels (SB 0 à 95)

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.

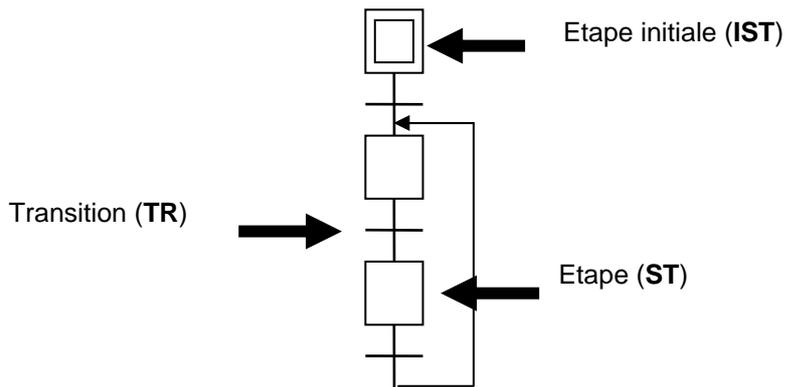
Une attente sur un événement séquentiel ne doit en aucun cas bloquer l'exécution permanente des programmes cycliques. Pour satisfaire à cette condition, les programmes séquentiels sont placés à l'intérieur de l'une des structures SB disponibles que l'on appelle à chaque cycle de programme.

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.

8.2 Structure cyclique avec appel au SB 0



Structure des blocs séquentiels SB



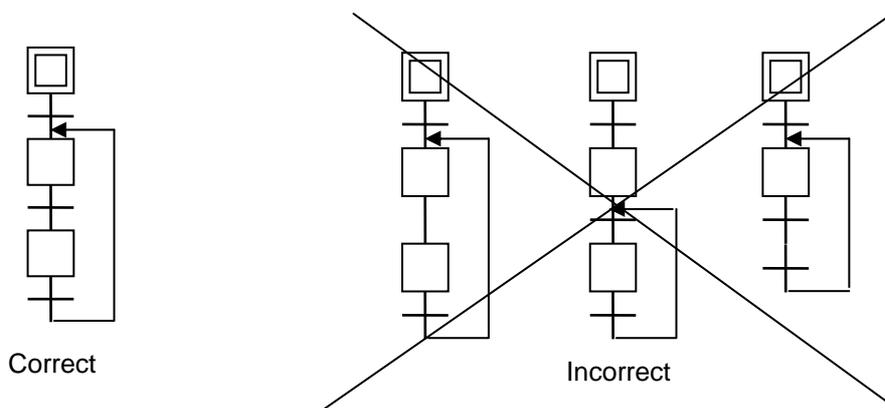
L'éditeur Graftec, permet de structurer les blocs SB à l'aide des étapes et transitions dans lesquelles l'utilisateur dépose les programmes sous la forme de fonctions graphiques ou de listes d'instructions.

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 étape que le programme va démarrer lors du premier appel au bloc SB. (Démarrage à froid)

La structure doit toujours être bouclée sur elle-même.

8.2.1 Règles d'évolution étapes, transitions

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.



8.2.2 Transitions (TR 0 à 5999)

:1 + Was the Info button touch?

L'évolution du processus Graftec est définie à l'aide du programme placé à l'intérieur des transitions, exécutée périodiquement jusqu'à l'apparition d'un événement détecté par le changement d'état d'une entrée, sortie, indicateur, évaluation de combinaisons logiques, ...

Si le programme est écrit en IL, la transition traite l'étape qui suit uniquement si l'*ACCU* est haut lorsque nous atteignons la fin de la transition.

Si le programme est écrit en Fupla, la transition traite l'étape qui suit uniquement si l'entrée de la Fbox ETR est haut.

Si non la transition reste active et sera évaluée à chaque cycle de programme jusqu'à ce que l'événement attendu survienne.

Exemple: Détecter le flanc positif d'une impulsion sur entrées

Programme IL

```
STH    I 8
DYN    F 80
```

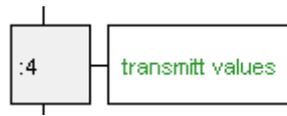
Programme Fupla



Remarque :

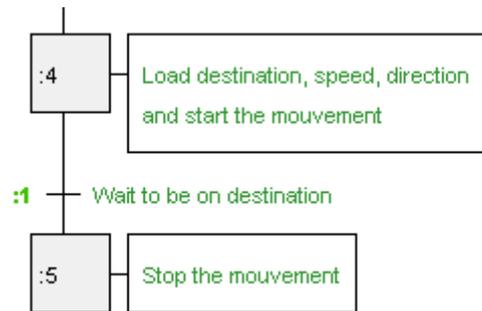
Pour les transitions en liste d'instructions, l'accumulateur est toujours à l'état haut au début d'une transition ou d'une étape. Ainsi toutes les instructions, y compris celle dépendantes de l'*ACCU*, sont toujours exécutées. Les transitions sans programme évoluent toujours vers l'étape qui suit.

8.2.3 Etapes (ST 0 à 5999)



Les étapes contiennent les programmes destinés à agir sur le processus d'automatisation: enclenchement/déclenchement des sorties, flags, calculs, charger une valeur dans un compteur ...

Exemple: Supposons que nous automatisons la translation d'un axe de machine à l'aide d'un moteur.



Nous définissons la position de destination, la vitesse et la direction du déplacement, puis nous démarrons le mouvement. Ces tâches sont exécutées qu'une seule fois, elles sont placées dans une étape.

Ensuite nous surveillons le déplacement, nous attendons d'être arrivé sur la position de destination. La position actuelle est évaluée et comparée avec la position de destination. Cette tâche est traitée périodiquement, elle est placée dans une transition. Arrivé sur la position de destination, la transition se termine avec un *Accu* ou une *Fbox ETR* à l'état haut, nous abandonnons la transition pour traiter l'étape qui suit.

L'étape suivante stop le moteur sur sa position de destination. Cette tâche est exécutée qu'une seule fois, elle est donc placée dans une étape.

Remarques :

Une étape sans programme passe directement à la transition suivante. Une étape n'est traitée qu'une seule fois, elle n'est pas traitée périodiquement comme les transitions!

Toutes structure Grafterc compte une étape initiale symbolisée avec un double cadre. Elle représente le point de départ du programme, lors du premier appel du bloc SB, après un démarrage à froid de l'automate ou une coupure de tension.



8.2.4 Propriétés d'une étape ou transition

La sélection d'une étapes ou transitions de la structure Graftec avec la souris affiche les informations suivantes dans la fenêtre disponible avec le menu *Block, Properties*.

General	
(Name)	SB_0.ST_5
Number	
Comment	Stop the mouvement
Type	Step
Scope	Local
Editor	No code editor

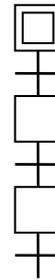
Name:	Nom du symbole de l'étape ou transition.
Number:	Numéro d'étape ou transition, Par défaut ce numéro du bloc est dynamique. Défini automatiquement lors du <i>build</i> . Si nécessaire nous pouvons le définir nous même. Selon les automates 2000 ou 6000 étapes et transitions sont à disposition
Comment:	Commentaire libre affiché sur le côté droite de l'étape ou transition
Type:	Etape ou transition
Scope:	Scope du symbole de l'étape ou transition. (Local ou Public). Public si le symbole est employé à l'extérieur du fichier.
Editor:	Editeur pour le programme

Ces informations sont aussi accessibles avec la fenêtre *View, Block Symbols*

8.2.1 Séquences Graftec typiques

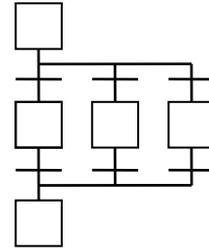
Séquence simple

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.



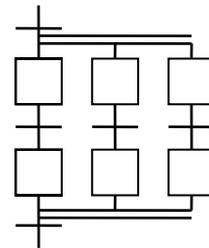
Divergence alternative (OU)

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.



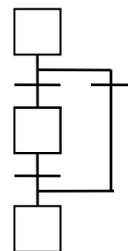
Divergence simultanée (ET)

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.



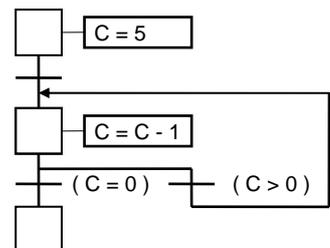
Saut de séquence

Le saut de séquence est réalisé à partir d'une divergence alternative et permet de traiter une séquence de manière conditionnelle.



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 exécute une séquence simple de longueur quelconque, décrémente le compteur. Si le compteur est non nul, la boucle est répétée.



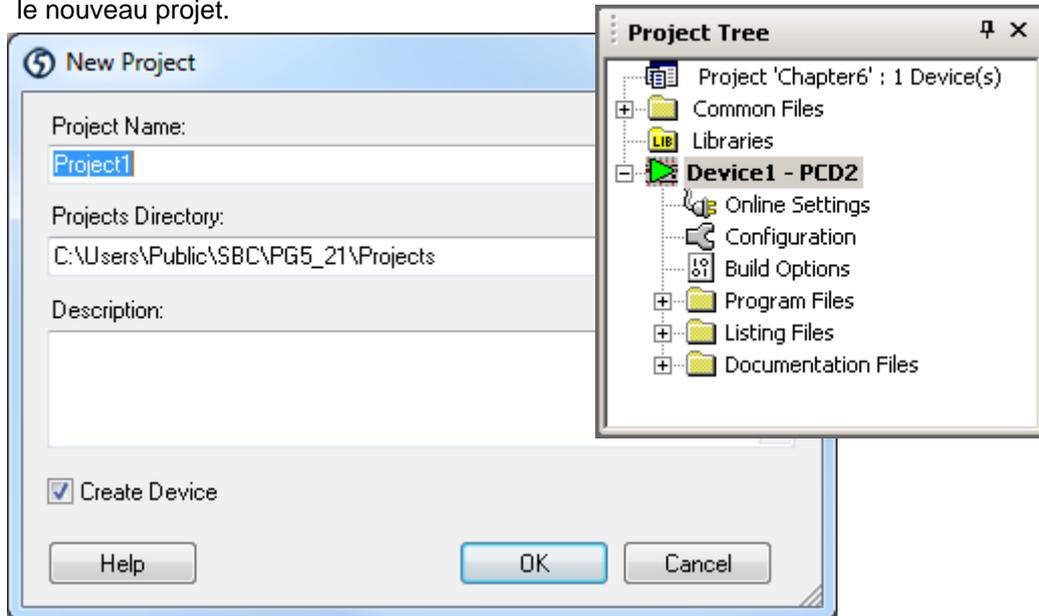
8.3 Edition d'un projet Graftec

Pour réaliser un exemple, nous conseillons de préparer un nouveau projet avec les fichiers nécessaires pour éditer les programmes Graftec.

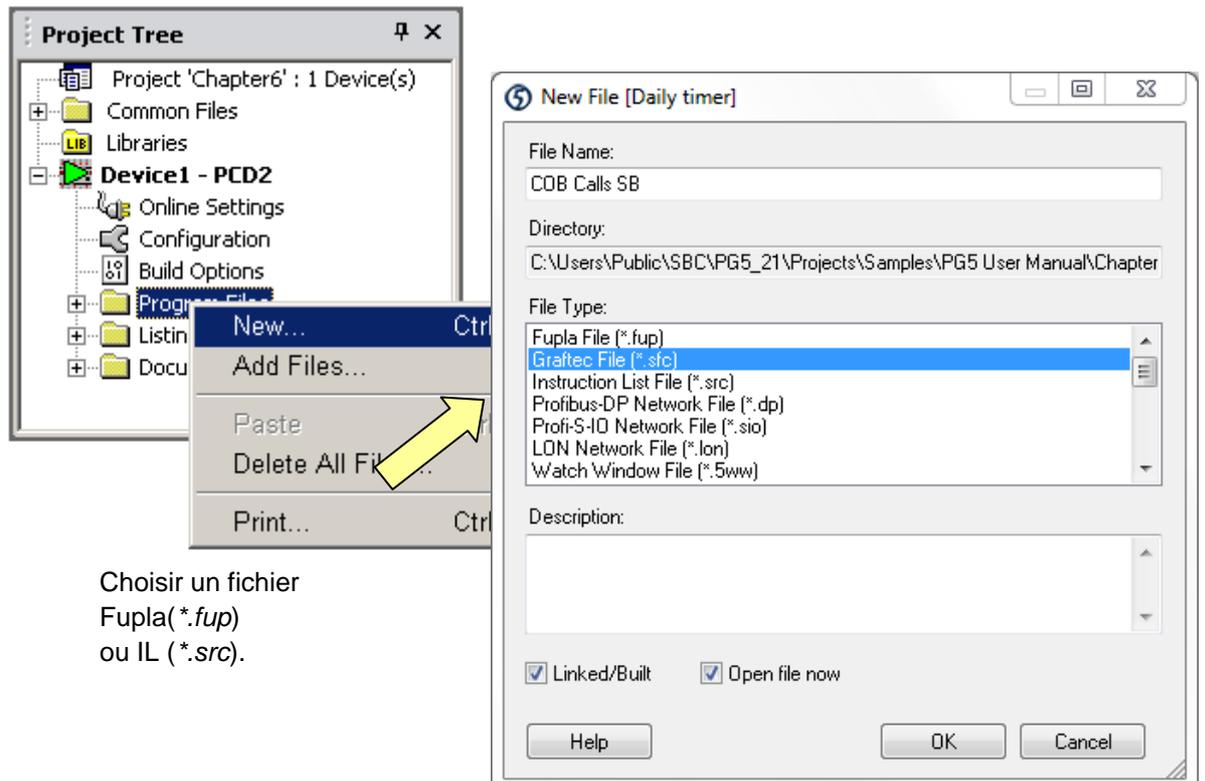
- Pour une programmation graphique, préparer un fichier Fupla et Graftec
- Pour une programmation en liste d'instruction, préparer un fichier SRC et Graftec

8.3.1 Créer un nouveau projet

Dans la fenêtre *SAIA PG5 Project Manager*, sélectionner le menu *Project, New...* et créer le nouveau projet.



8.3.2 Ajouter un fichier Fupla ou IL



Choisir un fichier Fupla (*.fup) ou IL (*.src).

8.3.3 Appel du SB à partir d'un COB

Selon le choix de programmation (SRC ou Fupla), appeler le SB à l'aide d'une instruction CSB ou d'une Fbox Call SB. Ouvrir le nouveau fichier et écrire le programme comme indiqué ci-dessous:

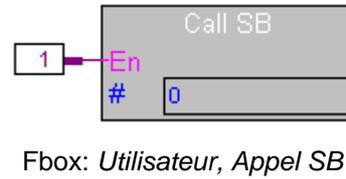
Programme IL:

```
COB 1 ; Début du COB
    0

CSB 0 ; Appel de SB 0

ECOB ; Fin du COB
```

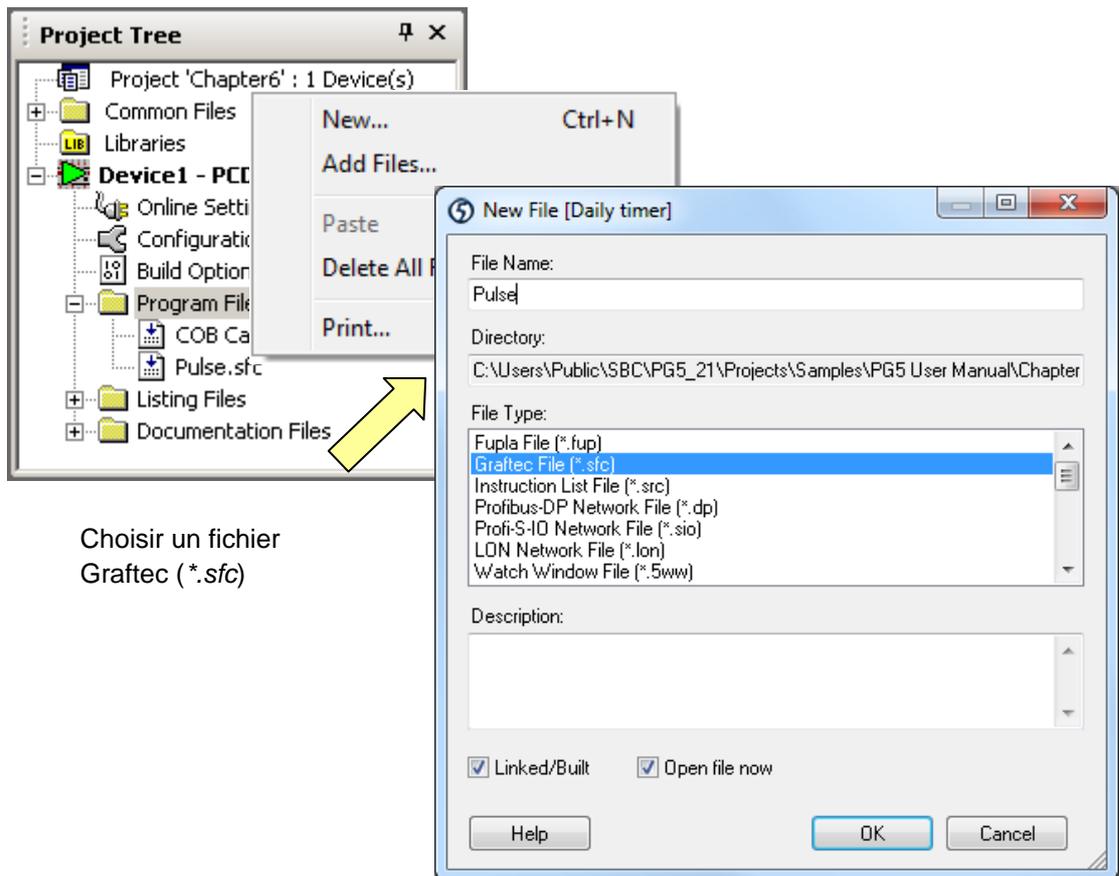
Programme Fupla:



L'appel du bloc SB peut être comme ci-dessus explicitement défini par une instruction CSB ou la Fbox correspondante. Mais elle peut être aussi implicite, PG5 se charge de créer un COB avec l'appel aux blocs SB définis dans les fichiers Graftec.

Une option permet d'activer ou désactiver ce service. Voir sous *Build Options*, puis *Advanced Options, Generate FB Calls*. (Cette option est par défaut active.)

8.3.4 Ajouter un fichier Graftec



Choisir un fichier Graftec (*.sfc)

8.3.5 Page Navigator, ajouter un bloc SB

A la création d'un fichier Graftec, l'éditeur crée automatiquement un bloc séquentiel SB et une étape initiale.

Plusieurs SB peuvent être édités dans un seul fichier Graftec. La sélection du menu View, *Other Windows, Page Navigator* affiche la liste de tous les SB et des pages qui constituent le fichier.

Si nécessaire nous ajoutons un bloc SB supplémentaire au fichier avec le menu *Block, New* et ajustons ses propriétés dans la fenêtrés affichées avec le menu *Block, Properties*

General	
(Name)	SB_2.SB_2
Number	
Comment	
Type	SB
Scope	Global

Name: Nom du symbole du bloc SB. Nous conseillons de nommer chaque SB avec un symbole, cela permet de faciliter la navigation dans les blocs

Comment: Commentaire libre

Number: Numéro de bloc, par défaut le numéro du bloc est dynamique. Défini automatiquement lors du build. Si nécessaire nous pouvons le définir nous même.
Ne pas confondre le numéro du SB avec les numéros des éléments étapes et transitions. Ce numéro comprend toute la chaîne d'étapes et transitions du SB représenté par le symbole dans le champ *Name*.

Scope : Scope du symbole du bloc SB (Local ou Public).
Public si le symbole du bloc est employé à l'extérieur du fichier.

Pour afficher la structure Graftec de l'un des SB présent dans le *Page Navigator*, positionner la souris au dessus du bloc et sélectionner le menu de contexte *Open Block*

Maintenant nous pouvons structurer le SB avec les étapes et transitions.

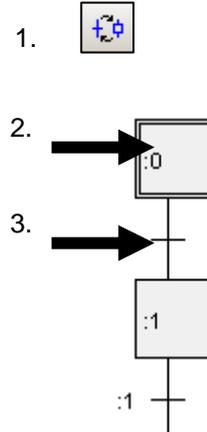
8.4 Editer une structure séquentielle

Un nouveau fichier Graftec affiche toujours l'étape initiale. L'exécution d'une séquence commence toujours ainsi. Nous pouvons ajouter de nouveaux éléments, soit au clavier, soit à l'aide de la barre d'outils.



8.4.1 Editer une séquence simple

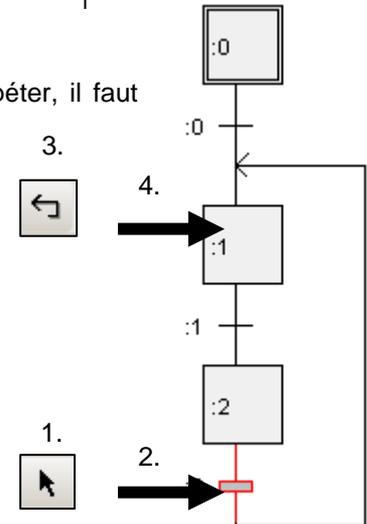
1. Sélectionner le bouton *Transition Mode*.
2. Placer la souris sur l'étape initiale et cliquer le bouton gauche de la souris.
3. Sélectionner le bouton *Step Mode*.
4. Déplacer la souris sur la nouvelle transition et cliquer.
5. Poursuivre de la même manière.



8.4.2 Editer une liaison

La séquence terminée, le programme l'est aussi. Pour la répéter, 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. Sélectionner le bouton *Select mode*.
2. Pointer la transition de départ.
3. Sélectionner sur le bouton *Link mode*.
4. Pointer l'étape vers laquelle lier la transition.

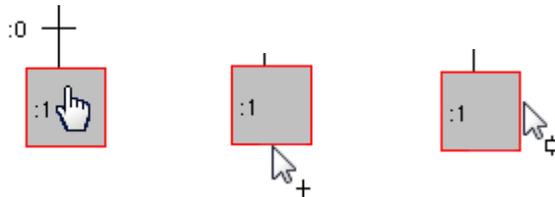


8.4.3 Option smart cursor



Smart Mode

L'édition de la séquence d'étapes et transitions peut fonctionner différemment que décrit en faisant usage du bouton *Smart Mode*. Cette option adapte automatiquement le mode et le curseur de la souris en fonction du contexte.



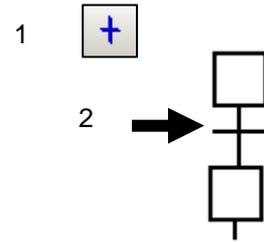
Si la souris est placée au centre de l'étape ou transition, le mode *Select mode* est appliqué. Le curseur est représenté par une main. Une double sélection de la souris ouvre l'élément pour éditer le programme.

Le déplacement de la souris vers le bas modifie le curseur qui représente une transition ou une étape selon ce qui sera inséré au dessous de l'élément lors de la sélection de la souris. C'est le mode insérer un transition qui est appliqué

Le déplacement de la souris sur le côté droite représente l'élément qui sera inséré sur le côté lors de la sélection de la souris.

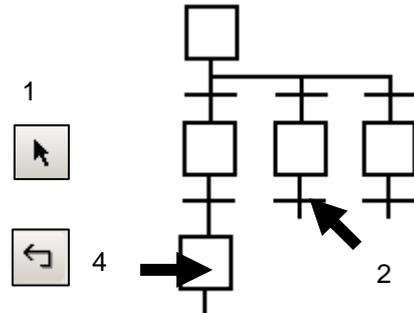
8.4.4 Editer une divergence alternative (OU)

1. Sélectionner le bouton *Transition mode*.
2. Pointer sur une transition déjà suivie d'une étape.
3. A chaque sélection de la souris, une nouvelle transition s'ajoute sur la droite.



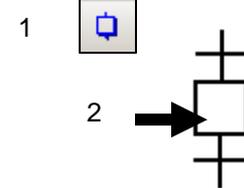
8.4.5 Editer les liens de convergence.

1. Sélectionner le bouton *Select mode*.
2. Pointer la transition à fermer.
3. Sélectionner le bouton *Link mode*.
4. Pointer l'étape à raccorder.



8.4.6 Editer une divergence simultanée. (ET)

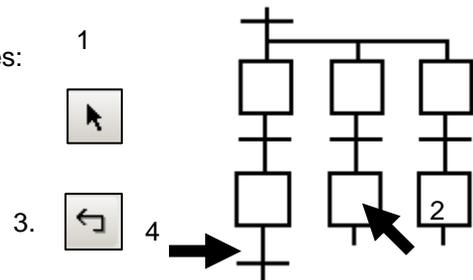
1. Sélectionner le bouton *Step mode*.
2. Pointer une étape déjà suivie d'une transition.
3. A chaque sélection de la souris, une nouvelle étape s'ajoute sur la droite.



8.4.7 Editer les liens de convergence.

Pour synchroniser les tâches simultanées:

1. Sélectionner le bouton *Select mode*.
2. Pointer l'étape à fermer.
3. Sélectionner le bouton *Link mode*.
4. Pointer la transition à raccorder.



8.4.8 Ajouter un commentaire

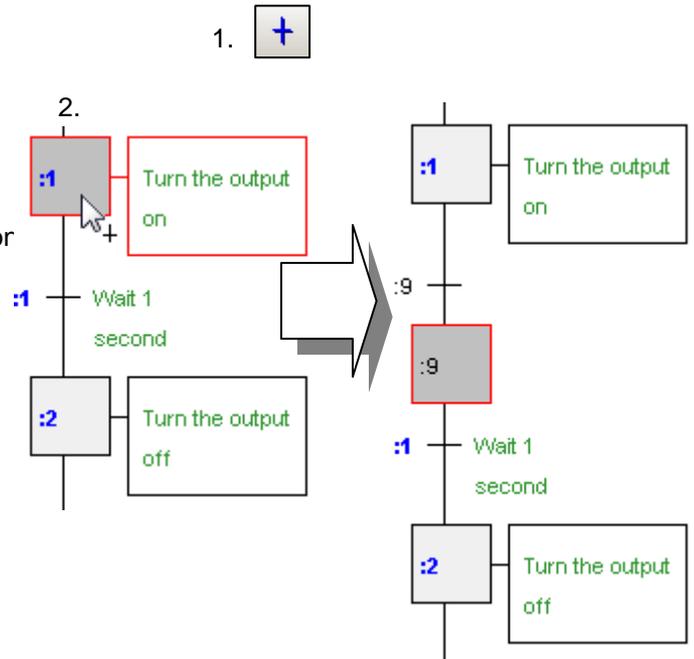
1. Sélectionner le bouton *Select mode*.
2. Pointer l'élément avec le bouton droit de la souris, choisir l'option *Properties...*
3. Introduire le commentaire dans le champ *Comment*.

Remarque : pour éditer le commentaire sur deux lignes, insérer la commande \n .

General	
(Name)	SB_0.ST_1
Number	
Comment	Turn input ON
Type	Step
Scope	Local
Editor	No code editor

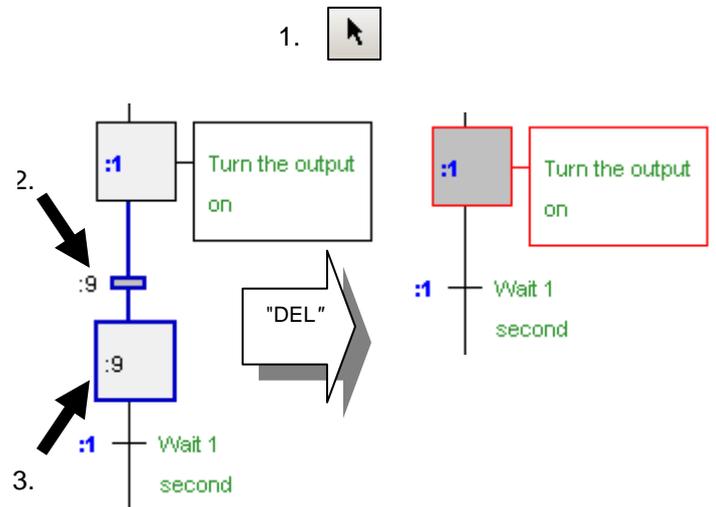
8.4.9 Insérer une séquence

1. Sélectionner le bouton *Transition Mode*.
2. Pointer une étape déjà suivie d'une transition.
3. L'éditeur ajoute une nouvelle étape et transitor



8.4.10 Effacer une séquence

1. Sélectionner le bouton *select mode*.
2. Pointer la première étape de la séquence.
3. Pointer la dernière transition de la séquence à supprimer en maintenant pressé la touche *Shift*.
4. Appuyez sur la touche *Del*.



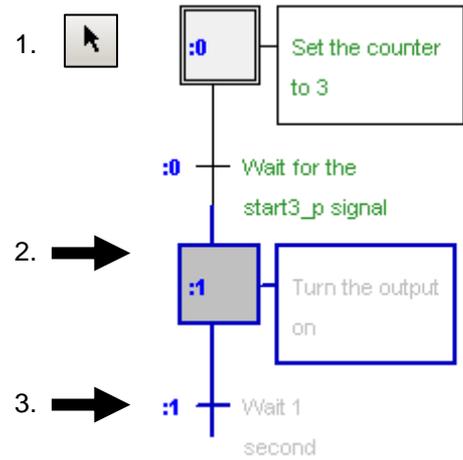
8.4.11 Copier-coller de séquences

Copier une séquence:

1. Sélectionner le bouton *Select mode*.
2. Pointer l'élément de début de la séquence.
3. Pointer le dernier élément de la séquence en maintenant pressé la touche *Shift*.
4. Menu *Edit, Copy* ou *CTRL+C*

Coller une séquence:

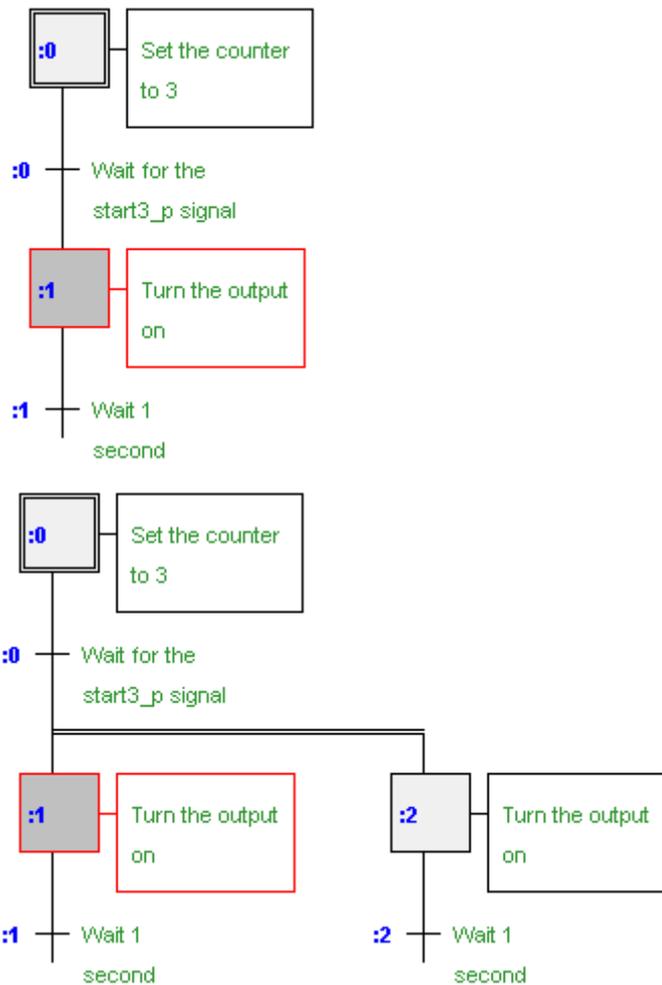
5. Sélectionner à nouveau le bouton *Select mode*
6. Pointer l'élément désignant la position pour insérer la séquence.
7. Menu *Edit, Paste* ou *CTRL+V*



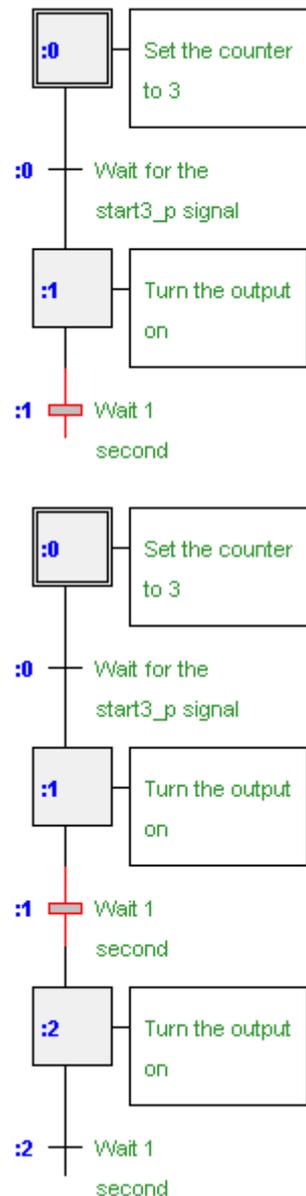
Remarque:

Selon la position de l'élément (transition ou étape) à insérer, la séquence peut être ajoutée sous ou à côté de l'élément sélectionné.

Insertion sur une étape



Insertion sur une transition

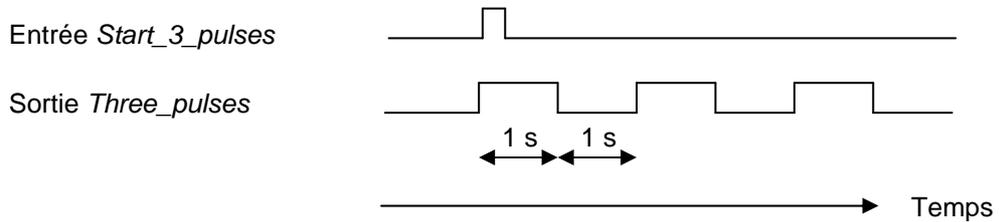


8.5 Ecriture d'un premier bloc séquentiel

Objectif:

Réaliser un programme destiné à faire clignoter la sortie O 33 (*Three_pulses*) trois fois de suite sur chaque activation de l'entrée I 2 (*Start_3_pulses*).

Diagramme temporel



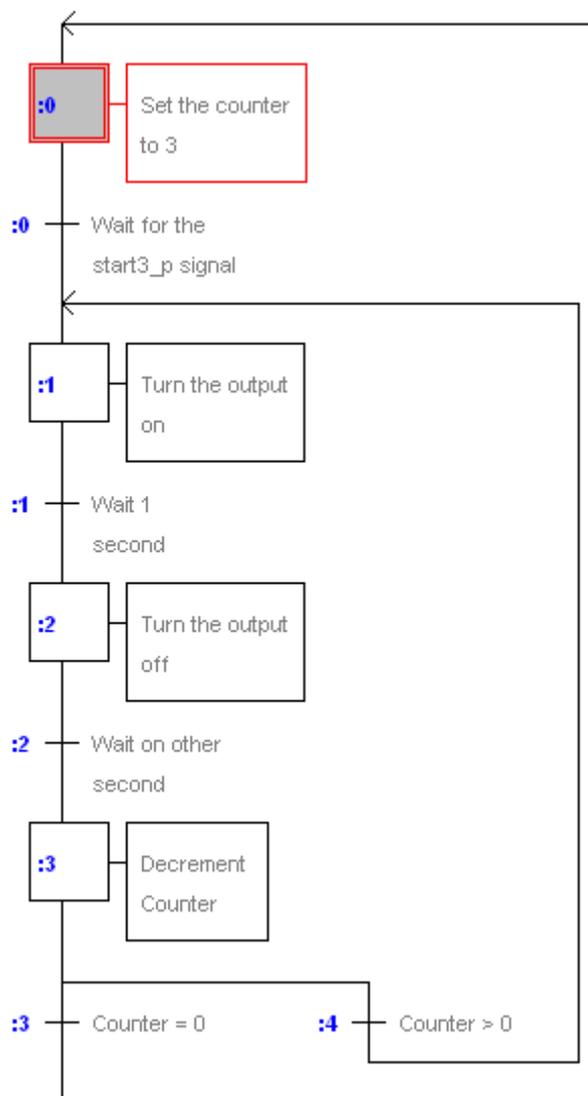
8.5.1 Réalisation de la structure de base

Nous commençons toujours par l'étape initiale qui sert de point de départ lors du démarrage à froid. Si elle n'est pas utile nous pouvons la laisser sans y définir de programme ou commentaire. Après quoi, nous attendons le signal de départ *Start_3_pulses*. Nous éditons une transition comme ci-dessous et ajoutons un commentaire dans la fenêtre des propriétés. Sélectionner l'étape ou transition et menu View, Properties Window.

The image shows a portion of a PLC ladder logic program. On the left, there is a counter block labeled ':0' connected to a block labeled 'Set the counter to 3'. Below it, a transition block labeled ':0' has a comment 'Wait for the start3_p signal'. On the right, the 'Properties' window is open, showing the configuration for the selected transition block.

Properties	
block :PULSE_,TR_0	
General	
(Name)	PULSE_,TR_0
Number	
Comment	Wait for the \nstart3_p signal
Type	Transition
Scope	Local
Editor	Function Block Diagram

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. Répéter l'opération trois fois, puis relancez la séquence.

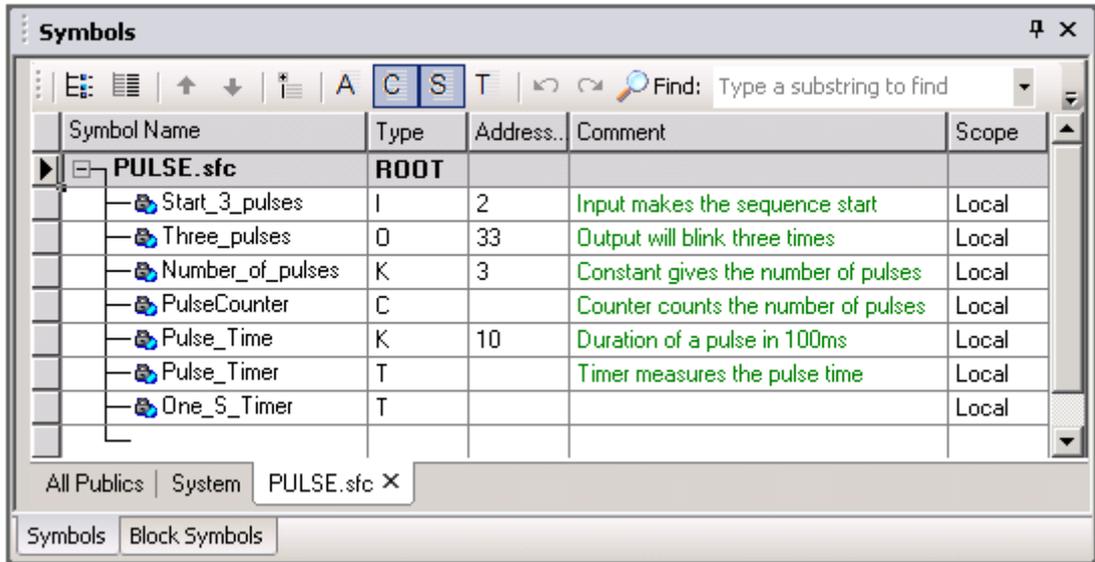


8.5.2 Choix de l'éditeur IL ou Fupla

La 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*). Le choix de l'éditeur est défini à l'aide d'une option du menu *View, Options..., Code Editor, default editor* ou la fenêtre des propriétés de la transition ou de l'étape sélectionnée.

8.5.3 Edition des symboles

Commençons par dresser, dans l'éditeur de symboles, la liste de tous les symboles utilisés, comme illustré dans la fenêtre ci-dessous.



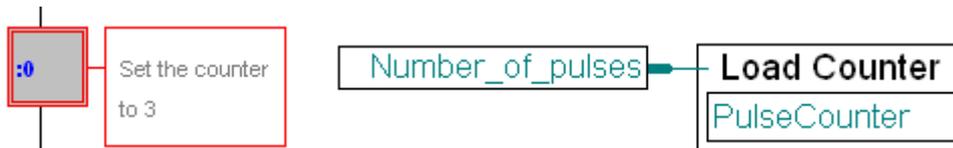
8.5.4 Programme d'une étape, charger un compteur

Nous ouvrons l'étape initiale pour y ajouter le programme permettant de charger le compteur *PulseCounter* avec la constante *Number_of_pulses* de valeur 3. Nous éditons le programme des étapes mais aussi les transitions en pointant l'élément dans la structure Graftec puis le menu *Edit, Code*.

Programme Fupla:

Utiliser la Fbox: Graftec, Charge compteur.

Attention: ne pas utiliser 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
```

8.5.5 Programme d'une transition, attendre un signal de départ

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 attendons que l'entrée *Start_3_pulses* passe à l'état haut.

Programme Fupla:



Ajouter la FBox: Graftec, Fin TR

Programme IL:

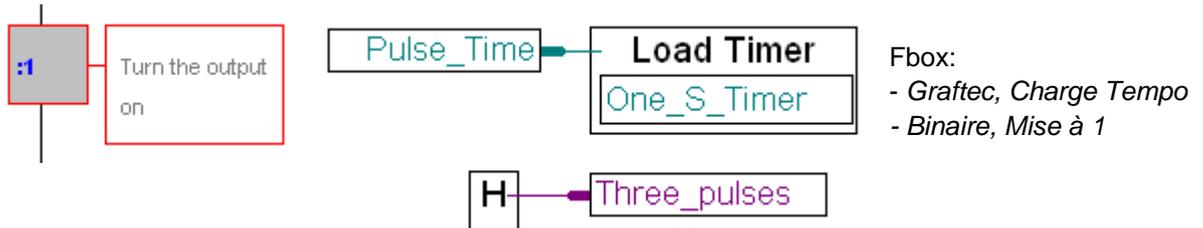
```
STH Start_3_pulses ; Copy the information Start_3_pulses into the accu
```

8.5.6 Activer une sortie et démarrer une temporisation

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 nous voulons les intégrer à un SB, nous nous limitons 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.



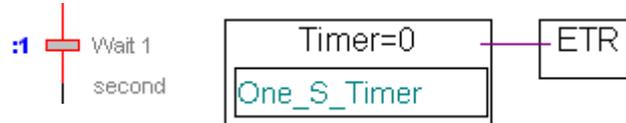
Fbox:
 - Graftec, Charge Tempo
 - Binaire, Mise à 1

Programme IL:

```
SET      Three_pulses      ;Active la sortie
LD       One_S_Timer       ;Charge la temporisation
                Pulse_Time
```

8.5.7 Attendre la fin de temporisation.

Programme Fupla:



Fbox:
 - Graftec, Tempo écoulé
 - Graftec, Fin TR

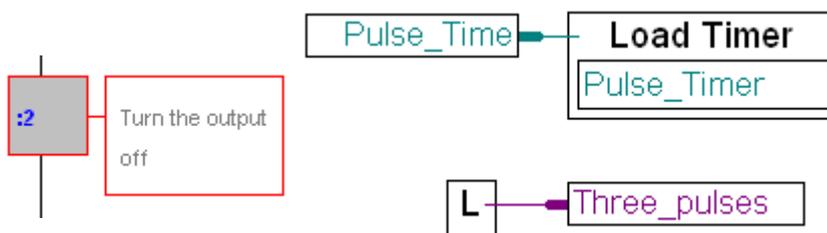
Programme IL:

```
STL      One_S_Timer       ;Active l'accumulateur lorsque la temporisation
                                est terminée
```

8.5.8 Désactiver une sortie et attendre la fin de temporisation

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
```

Programme Fupla:



Programme IL:

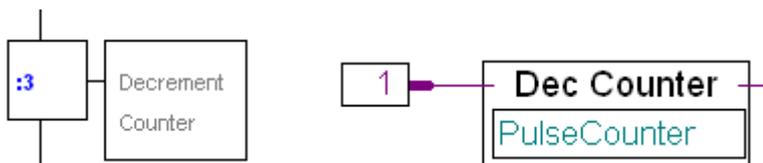
```
STL Pulse_Timer ;Active l'accumulateur lorsque la
temporisation
;est terminée
```

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 !

8.5.9 Décrémentation 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).



Programme IL:

Fbox: Graftec, Compteur -1

```
DEC PulseCounter ;Si l'accumulateur est à 1, décrémente le compteur
```

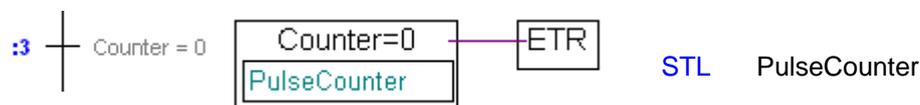
(Remarque:l'accu est toujours à 1 au début d'un ST/TR)

8.5.10 Divergence alternative

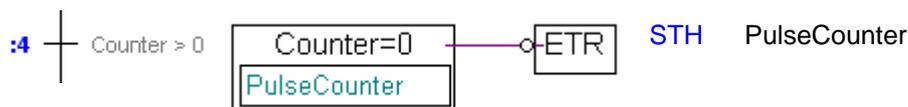
Les deux dernières transitions sont présentes pour effectuer un choix:

Programme Fupla:

Programme IL:



Fbox: Graftec, Compteur = 0



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*.

8.6 Construction et débbugage du programme (Build)



Build All

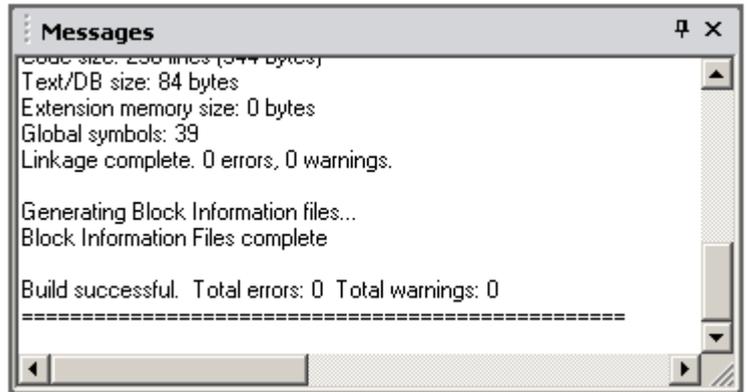
Une fois le programme terminé, sélectionner le bouton *Build* de la barre d'outils pour compiler la totalité du programme.

8.6.1 Fenêtre des messages

La fenêtre des *messages* nous 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.

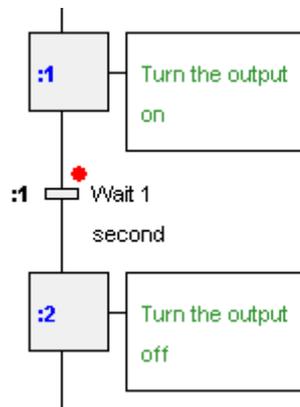
8.6.2 Outils en ligne



Download Program

Il reste à télécharger le programme et à passer en mode en ligne. 

Le bloc séquentiel peut être dynamiquement visualisé, un point rouge signale la transition ou l'étape active. Permet d'observer l'évolution du programme dans la structure Graftec.



Nous pouvons stopper le programme à tout moment avec le bouton *Stop* et poursuivre son exécution en pas-à-pas.



Chaque sélection du bouton *Step by Step* exécute une étape ou transition sans afficher son contenu



Chaque sélection du bouton *Step In* exécute une étape ou transition et affiche son contenu avec l'éditeur Fupla ou IL.

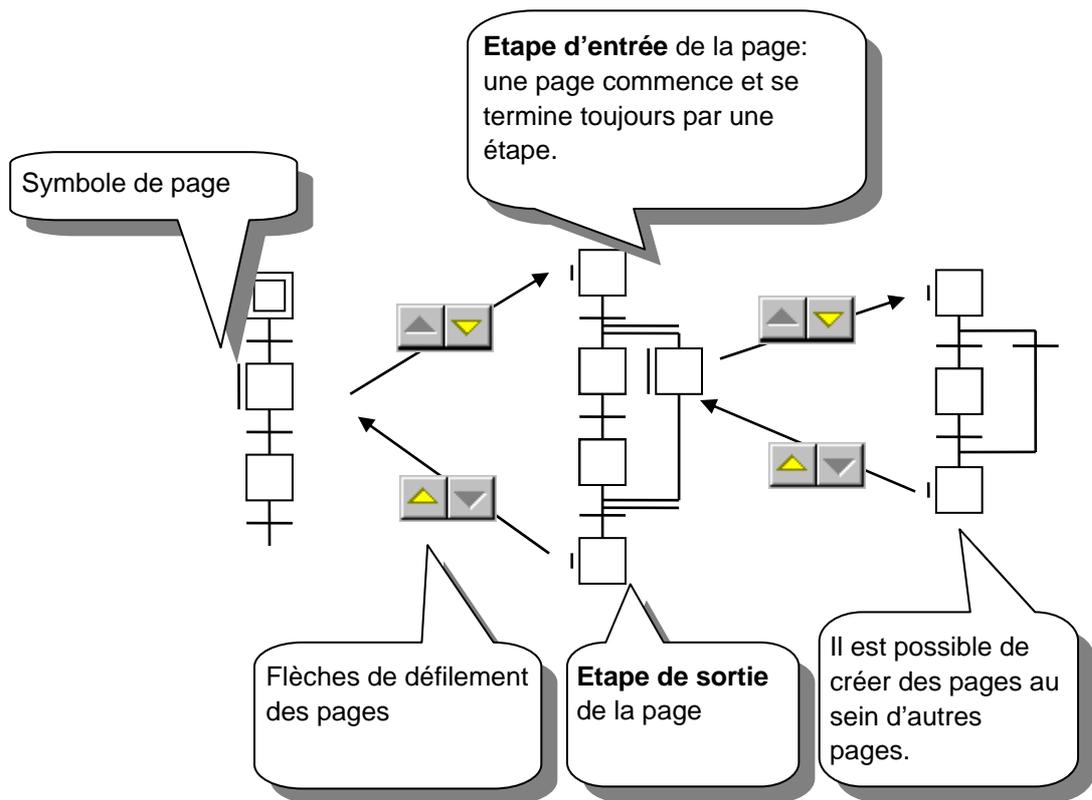


Le bouton *Run to Element* traite tous les éléments Graftec compris entre la transition active marquée avec un stop et l'élément Graftec sélectionné. Si en mode *Run*, stop le programme au passage de l'élément sélectionné.

8.7 Structuration du Graftec en pages

Graftec permet de grouper une séquence d'étapes et transitions dans un nouvel élément appelé page. Cet élément a l'apparence d'une étape, il peut aussi recevoir un commentaire et se distingue par un trait vertical sur le côté gauche. Ainsi la structure Graftec devrait être décrite à son plus haut niveau des fonctionnalités globales représentées par des pages que nous pouvons ouvrir pour afficher une séquence plus détaillée composée de transitions, étapes et pages. Les pages peuvent contenir d'autres pages sans limitation de niveau d'imbrication.

L'enchaînement des pages offre alors un effet de zoom permettant de représenter le contenu et la progression du programme selon le niveau de détails désiré.



8.7.1 Règles pour éditer des pages

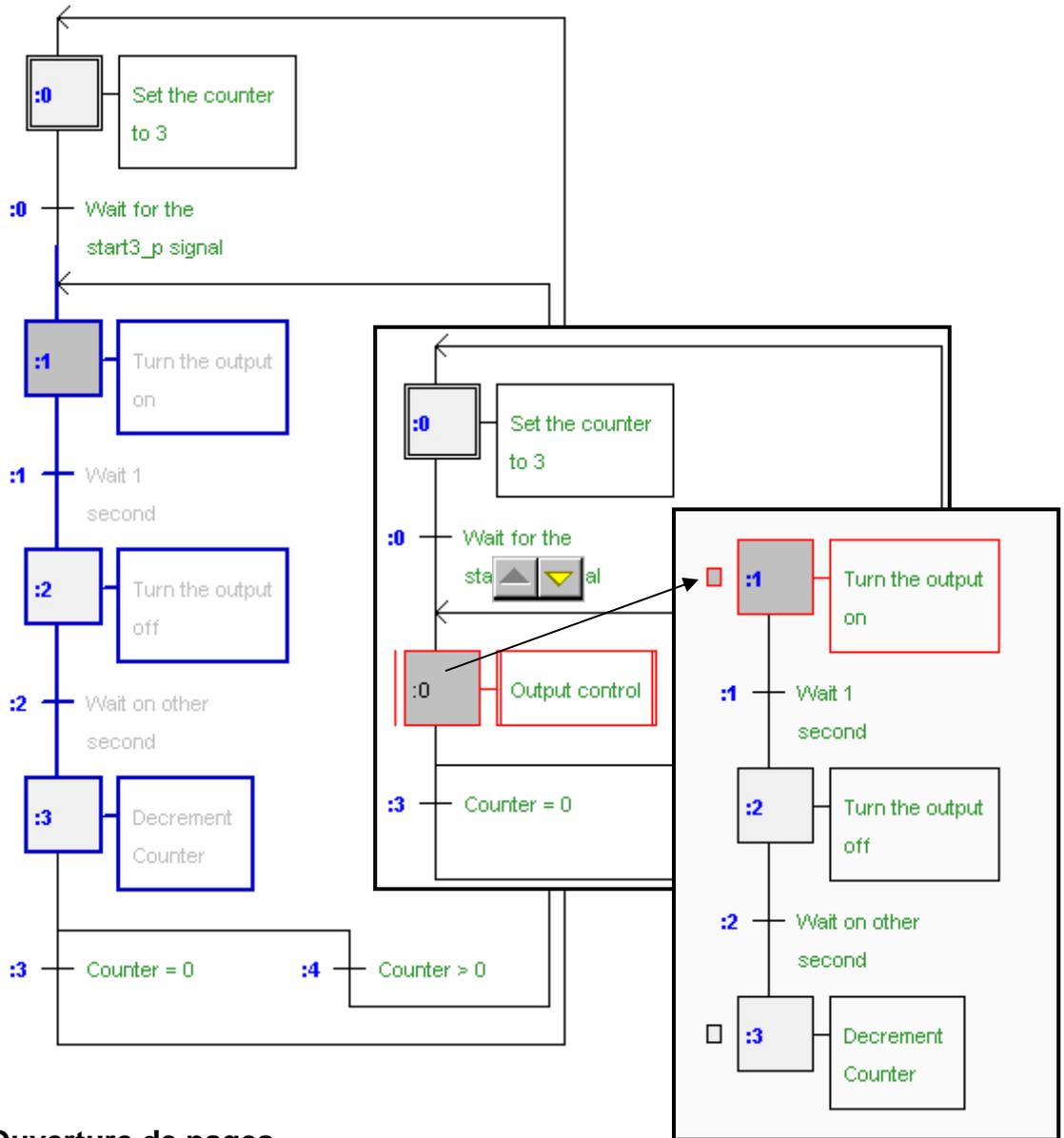
Les séquences Graftec appelées à devenir des pages doivent respecter quelques règles:

- La séquence représentée par une page commence et se termine toujours par une étape.
- La séquence ne comprend qu'une seule étape d'entrée et sortie.
- Il est impossible de déplacer ou de supprimer une étape d'entrée ou de sortie.

8.7.2 Edition d'une nouvelle page

Pour créer une page nouvelle, procéder comme suit :

- Sélectionner le bouton *Select mode* de la barre d'outils.
- Sélectionner la première étape de la séquence qui va constituer la page.
- Presser la touche *Shift* et sélectionner sur la dernière étape de la séquence.
- Sélectionner le menu *Page, Create* ou le bouton qui lui correspond.



8.7.3 Ouverture de pages



Pour afficher le contenu d'une page, pointer la page dans la structure Graftec puis le menu *Page, Subpage* ou le bouton correspondant.



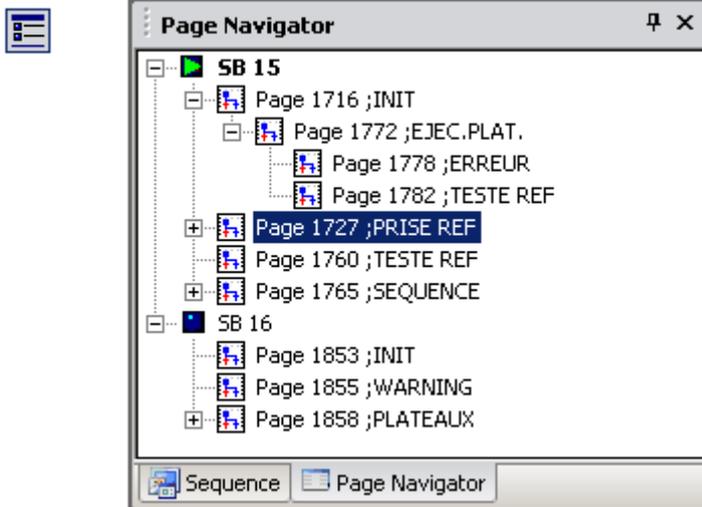
Le menu *Page, Calling* affiche la page de niveau supérieur. La page du premier niveau est affichée avec le menu *Page, Go To Main*



8.7.4 Suppression des pages

Remplace le symbole de la page par toute la séquence. Pointer la page dans la structure Graftec puis le menu *Page, Expand*

8.7.5 Block Navigator



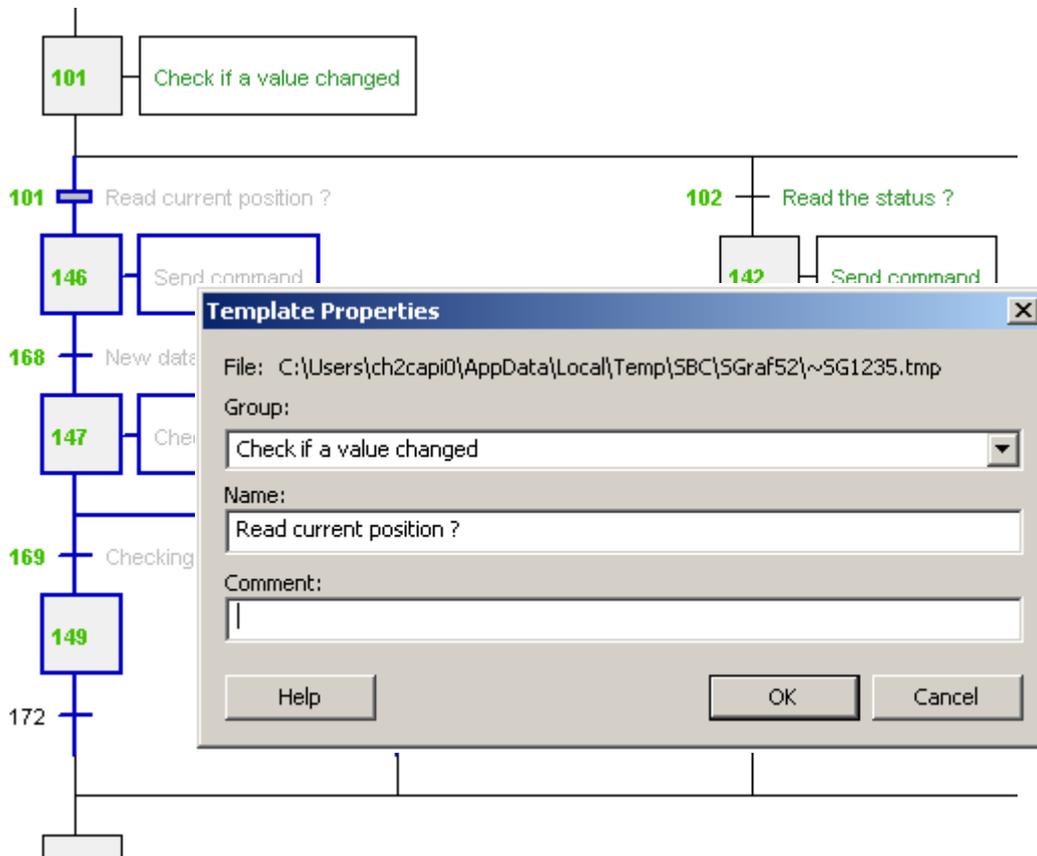
Il est vivement recommandé de faire usage des pages avec les grandes structures Graftec. Cela améliore la lecture du programme ainsi que la navigation entre les différentes fonctionnalités importantes représentées par les pages.

Le *Block Navigator* donne une vue globale de tous les blocks SB et des pages qui constitue le fichier. La sélection d'une page ou d'un bloc de cette vue affiche directement le bloc ou la page correspondante sans avoir à la rechercher dans la structure Graftec.

8.8 Graftec templates

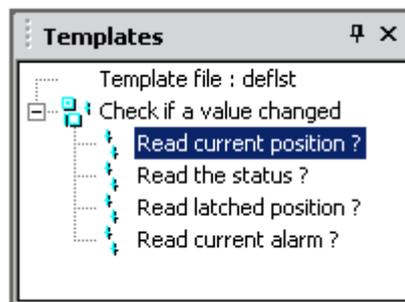
Les séquences d'étapes et transitions souvent présentes dans les divers programmes Graftec sont collectionnées sous la forme de *templates* que nous utilisons comme une librairie de séquences dont nous ne modifions que les noms et les adresses des symboles.

8.8.1 Créer un template



La création d'un template Graftec est très simple à réaliser. Nous sélectionnons une séquence d'étapes et transition quelconque, puis le menu de contexte *Add to templates*. Un dialogue nous invite à définir le groupe, le nom et le commentaire du template.

Les templates sont organisées en groupes comparables à des familles de fonctions. Les groupes rassemblent les templates selon les critères de classification définis par son auteur. Le groupe et le nom définissent la structure d'organisation de la fenêtre *View Templates*.



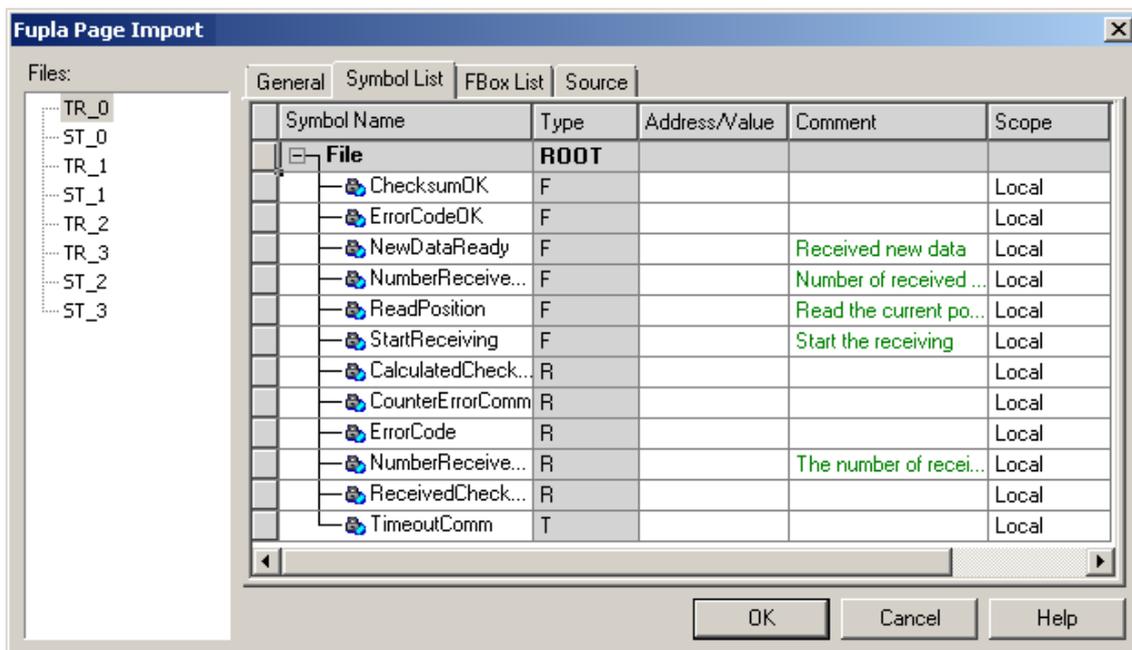
Les icônes des templates indiquent comment commence et se termine la séquence Graftec. Toutes sortes de séquences sont autorisées pour créer des templates : pages, divergences,...

8.8.2 Importation de templates

Les templates sont disponibles depuis tous les projets. Il suffit de sélectionner le template dans la fenêtre *View Templates* et le glisser sur la structure Grafterc pour insérer la séquence d'étapes et transitions ainsi que les commentaires et programmes Fupla ou IL qui les constituent.

Un dialogue est automatiquement affiché, il permet d'ajuster le nom et les adresses des symboles importés par le template et les divers autres paramètres.

Cette fonctionnalité peut être comparée à une macro ou une fonction avec des paramètres que nous utilisons comme une librairie.

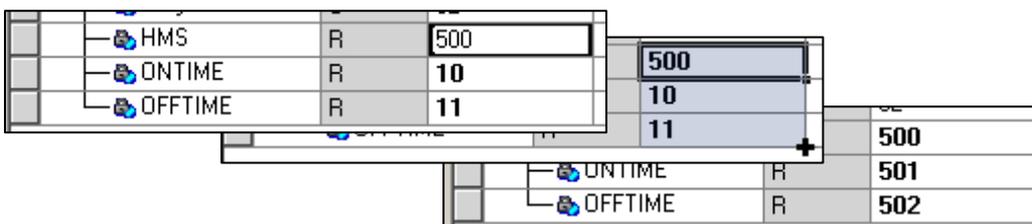


Files, représente la liste des étapes et transitions présentes dans le template.

Le répertoire *Symbols List* représente tous les symboles importés avec le template. Il permet de redéfinir les symboles avec de nouveaux noms, adresses, commentaires est scopes.

Sélectionner 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 notre choix.



Pour corriger les adresses des symboles, nous les trions par type en sélectionnant l'entête de colonne *Type*. Nous éditons l'adresse de la première cellule, puis nous sélectionnons son angle du bas droit pour l'étirer vers le bas.

Lors de l'importation de plusieurs copies du même template, voir paramètre de la page *General*, il est utile d'insérer un index dans les noms des symboles ou des groupes en utilisant le caractère # . Ce caractère est automatiquement remplacé par l'index de base augmenté d'une unité pour chaque copie du template. Il est aussi possible de sélectionner les symboles puis le menu de contexte *Indexing*.

Table des matières

9	PROGRAMMER EN IL (INSTRUCTION LIST)	2
9.1	Préparer un projet IL	3
9.1.1	Créer un nouveau projet	3
9.1.2	Créer un nouveau fichier IL	3
9.2	Organisation d'une fenêtre d'édition IL	4
9.2.1	Editer une ligne d'instruction	5
9.2.2	Mise en page des lignes d'instruction	6
9.2.3	Editer un bloc d'organisation	6
9.2.4	Ordre de traitement des instructions et des blocs	6
9.2.5	Règles à respecter pour éditer les blocs	7
9.3	Fenêtre <i>Symbol Editor</i>	8
9.3.1	Ajouter un nouveau symbole dans la liste <i>Symbol Editor</i>	9
9.3.2	Modes d'adressage des opérands	10
9.3.3	Utiliser la liste <i>Symbol Editor</i> dans le programme IL	11
9.3.4	Symboles Local, Public et External	12
9.4	Introduction au jeux d'instructions PCD	13
9.4.1	L'accumulateur	13
9.4.2	Les instructions binaires	14
9.4.3	Dynamisation	18
9.4.4	Les indicateurs de statut	19
9.4.5	Les instructions mots pour les temporisateurs	20
9.4.6	Les instructions mots pour les compteurs	22
9.4.7	Les instructions dépendantes de l'accumulateur	23
9.4.8	Les instructions mots pour l'arithmétique sur les nombres entiers	24
9.4.9	Les instructions mots pour l'arithmétique flottante	25
9.4.10	Conversion des registres entiers et flottants	25
9.4.11	Registre d'index	26
9.4.12	Sauts de programme	27
9.5	Edition d'un premier programme d'application	29
9.6	Construction du programme (Build)	31
9.7	Charger le programme dans l'automate	32
9.8	Débugger un programme (Debug)	32
9.8.1	Go On/Offline, Run et Stop	33
9.8.2	Mode pas-à-pas	34
9.8.3	Le point d'arrêt	35
9.8.4	Modification de programme en ligne	36
9.8.5	Visualiser et modifier l'état des symboles avec le <i>Watch window</i>	37
9.9	Mise en service d'un module analogique	38
9.9.1	Exemple pour les entrées analogiques PCD2.W340	38
9.9.2	Exemple pour les sorties analogiques PCD2.W610	39

9 Programmer en IL (Instruction List)

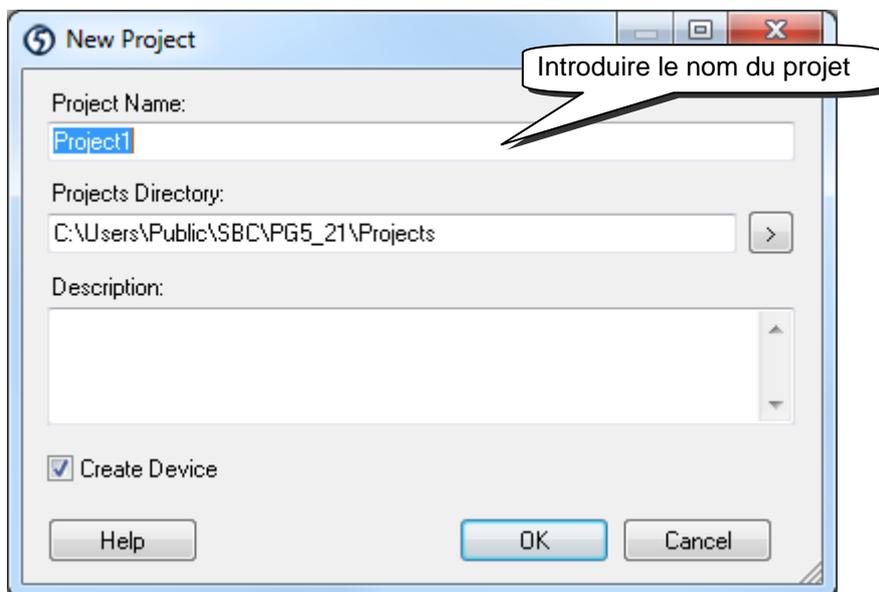
L'éditeur IL, est le moyen le plus flexible et le plus performant pour réaliser la programmation des automates Saia PCD. IL est un terme anglais "*Instruction List*" qui désigne un environnement de programmation non graphique où l'utilisateur écrit les programmes à l'aide d'un puissant jeu d'instruction 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.

9.1 Préparer un projet IL

Pour réaliser un exemple, nous conseillons de préparer un nouveau projet et de préparer un fichier pour éditer le programme IL.

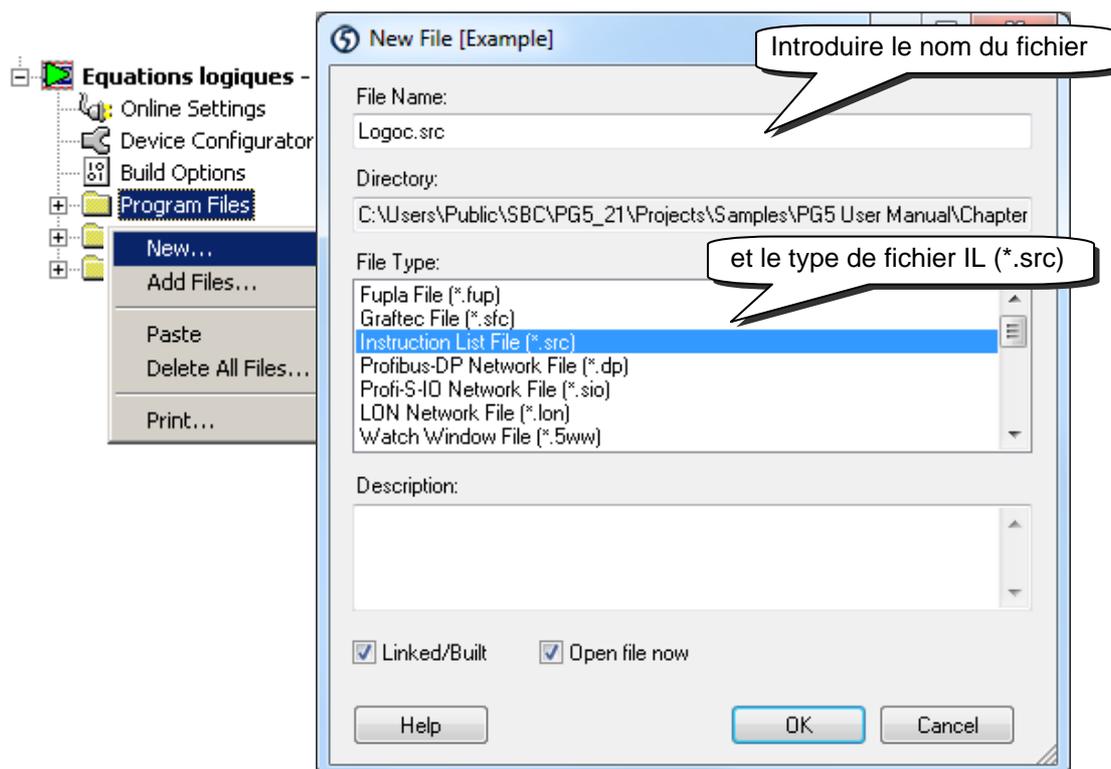
9.1.1 Créer un nouveau projet

Dans la fenêtre *Saia PG5 Project Manager*, sélectionner le menu *Project, New...* et créer le nouveau projet.



9.1.2 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 de contexte *New...* ou sélectionner le bouton *New File* de la barre d'outils.



9.2.1 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)

La mnémonique

Représenté en bleu, la mnémonique ou l'instruction programme définit le travail à effectuer sur une information tel qu'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.



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.

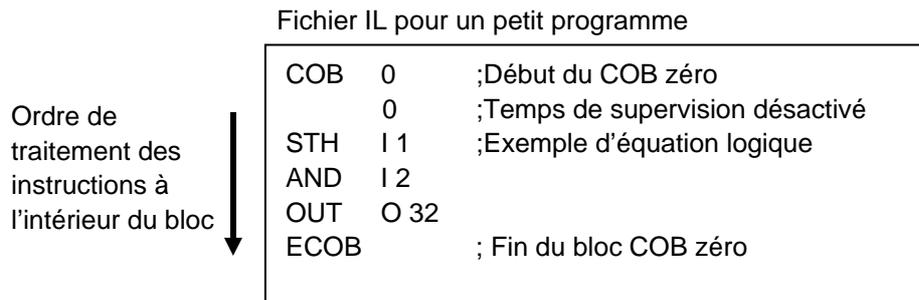


9.2.2 Mise en page des lignes d'instruction

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*.

9.2.3 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.

9.2.4 Ordre de traitement des instructions et des blocs

A l'intérieur d'un bloc, l'automate 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, 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.

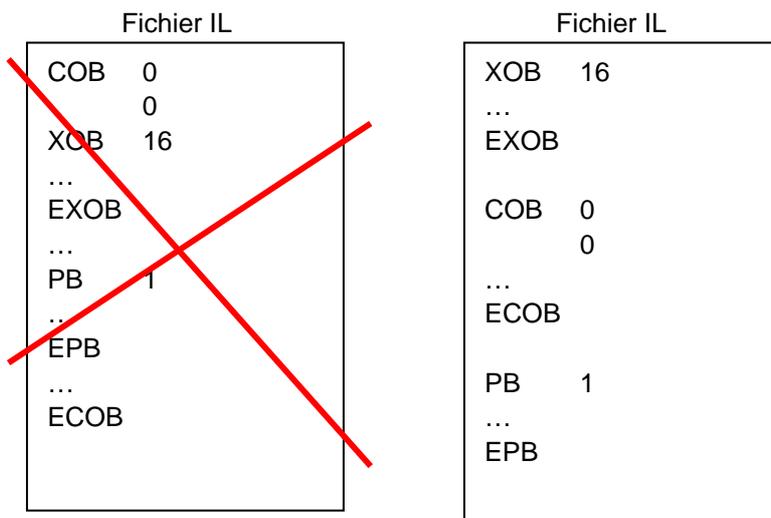
Tous les blocs pour les programmes séquentiels (SB), sous programmes (PB), fonctions paramétrables (FB) sont appelés par le programme utilisateur avec les

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.

9.2.5 Règles à respecter pour éditer les blocs

Même si les blocs peuvent être écrits dans un ordre quelconque, les règles suivantes sont à respecter :



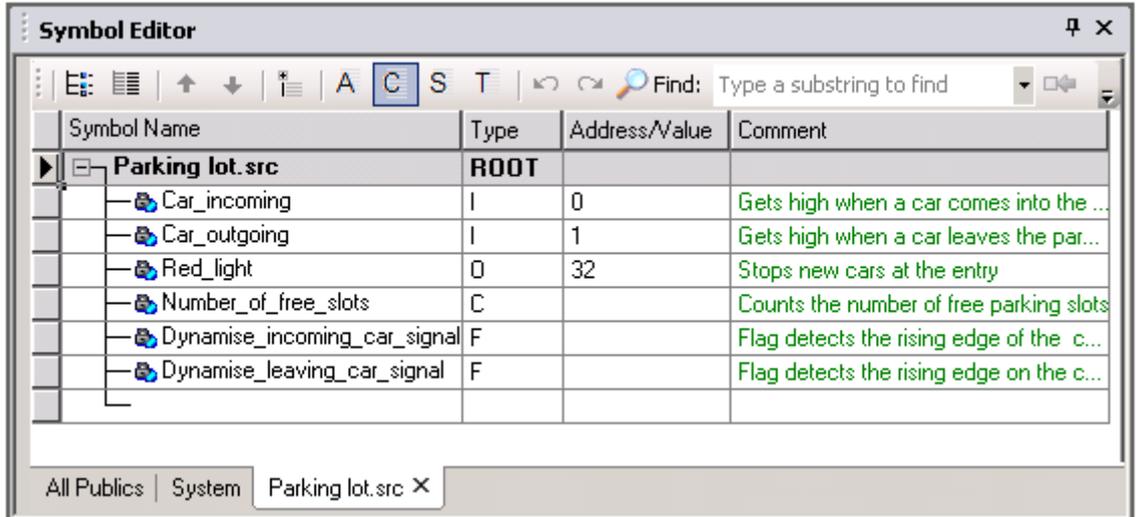
Les blocs ne peuvent pas être écrits les uns dans les autres mais à la suite les uns des autres.

Aucune instruction de programme ne peut être définie à l'extérieur d'un bloc, excepté les définitions des symboles, textes et blocs de données.

9.3 Fenêtre *Symbol Editor*



Show Hide Symbols Editor



La fenêtre *Symbol Editor* rassemble les opérandes 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 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 supporté depuis la fenêtre *Symbol Editor*. Il n'est alors pas nécessaire de reporter la correction à chaque ligne du programme mais uniquement dans la fenêtre *Symbol Editor*. 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 l'automate.

Indicateurs: F 0, ..., F 16383

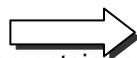
Registres: R 0, ..., R 16383

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

...

Le commentaire.

STH Flag ; Copie l'état du flag dans l'accumulateur



STH Flag ;Active incrémentation

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

9.3.1 Ajouter un nouveau symbole dans la liste *Symbol Editor*

Méthode simple

Pour ajouter un symbole dans la liste, ouvrir la fenêtre *Symbols Editor*, 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 *Symbol Name*, *Type*, *Address/Value*, *Comment* et *scope*.

Méthode rapide 1

Symbol Name	Type	Address/V...	Comment	Scope
[-] Parking lot.src	ROOT			
[-] Red_light o 32;stop new cars				
[-] Red_light	0	32	stop new cars	Local

Enter →

Il est aussi possible d'introduire les variables des divers champs d'informations depuis le champs *Symbol Name*. 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

Symbol Name	Type	Address/V...	Comment	Scope
[-] Parking lot.src	ROOT			

`sth Red_light= 0 32; stop new cars`

Enter →

Symbol Name	Type	Address/V...	Comment	Scope
[-] Parking lot.src	ROOT			
[-] Red_light	0	32	stop new cars	Local

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 même syntaxe que la méthode précédente. La sélection de la touche *Enter* du clavier permet de placer automatiquement le nouveau symbole dans la liste *Symbol Editor*.

9.3.2 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

Symbol Name	Type	Address...	Comment	Scope ▲
[-] Parking lot.src	ROOT			
[-] [Icon] [Empty]	F	50	Flag detects the ri...	Local
[Empty]				

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

Symbol Name	Type	Address...	Comment	Scope ▲
[-] Parking lot.src	ROOT			
[-] [Icon] Dynamise_incoming_car_signal	F	50	Flag detects the ri...	Local
[Empty]				

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

Symbol Name	Type	Address...	Comment	Scope ▲
[-] Parking lot.src	ROOT			
[-] [Icon] Dynamise_incoming_car_signal	F		Flag detects the ri...	Local
[Empty]				

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 *Build Options*. (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.

9.3.3 Utiliser la liste *Symbol Editor* dans le programme IL

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

Introduction du symbole au clavier.

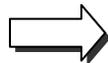
Le nom du symbole est complètement introduit au clavier pour chaque instruction qui en fait usage. Cette manière de faire laisse la possibilité d'éditer le nom du symbole avec une erreur de frappe qui se manifestera seulement à la construction du programme.

Introduction par recherche sélective.

Symbol Name	Type ▲	Address..	Comment	Scope
Parking lot.src	ROOT			
Number_of_free_slots	C		Counts the numbe...	Local
Dynamise_incoming_car_signal	F		Flag detects the ri...	Local
Dynamise_leaving_car_signal	F		Flag detects the ri...	Local
Car_incoming	I	0	Gets high when a ...	Local
Car_outgoing	I	1	Gets high when a ...	Local
Red_light			s at ...	Local

↑, ↓, Enter

sth Dyn



sth Dynamise_incoming_car_signal

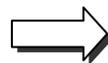
Seul les premiers caractères du nom du symbole sont introduits au clavier. La sélection simultanée des touches *Ctrl+espace* affiche une fenêtre avec la liste des symboles qui commencent avec les mêmes lettres que celles saisies précédemment. Il ne reste qu' à sélectionner le symbole désiré avec la souris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

Introduction par sélection et glissement

Symbol Name	Type ▲	Address..	Comment	Scope
Parking lot.src	ROOT			
Dynamise_incoming_car_signal	F		Flag detects the ri...	Local

Placer la souris sur le bouton au début de la ligne, presser le bouton gauche de la souris.

sth



sth Dynamise_incoming_car_signal

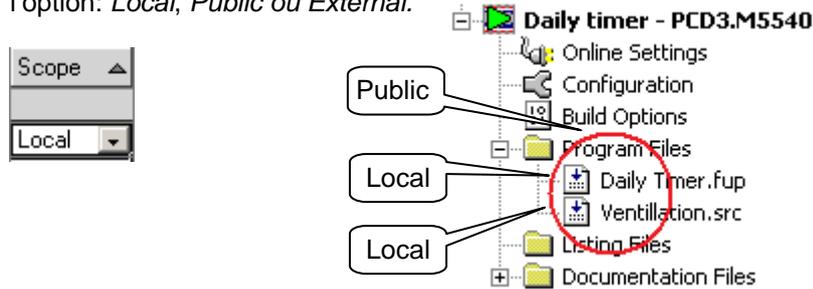
Glisser la souris dans l'éditeur IL

Lâcher le bouton de la souris

Cette manière de faire usage d'un symbole ne laisse aucune possibilité de faire une erreur de frappe. Dans la fenêtre *Symbols Editor* placer la souris sur le bouton au début de 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.

9.3.4 Symboles Local, Public et External

La fenêtre de définition des symboles comporte une cellule *Scope* pour définir l'option: *Local*, *Public* ou *External*.



Le scope définis l'accessibilité des symboles:

Les symboles locaux sont uniquement accessibles par le fichier programme qui les définis. Par exemple, les fichiers *Parking lot.fup* **ou** *Ventillation.src*.

Les symboles Public sont accessibles par tous les fichiers présents dans le *device*. Par exemple, les fichiers *Parking lot.fup* **et** *Ventillation.src*.



L'éditeur de symboles affiche plusieurs vues: une vue avec le nom du fichier ouvert, ainsi que les vues *All Publics* et *System*.

La vue avec le nom du fichier ouvert supporte la définition de tous les symboles publics et locaux nécessaires au programme du fichier ouvert.

Les nouveaux symboles ajoutés dans l'éditeur de symboles ou Fupla sont par défaut local ou public selon l'option définie sous le menu *View, Option, Symbols, Add symbols with Public scope*.

La vue *ALL Publics* rassemble les symboles publics définis dans le device. La vue *System* affiche la liste de tous les symboles systèmes. Les symboles des vues *All Publics* et *System* sont mis à jour lorsque le fichier est sauvé ou lors du *Build*.

La vue *All Publics* utilise le résultat du *Buid* pour rassembler les symboles publics définis dans les divers fichiers programmes du *device* sous une même est unique vue.

Pour placer un symbole public ou system dans le programme, sélectionner le symbole dans la vue *All Publics* ou *System* et le glisser dans la page Fupla. La référence au symbole public est dupliquée dans la grille correspondant au fichier ouvert avec un scope *External*. Cela indique que ce symbole est défini à l'extérieur du fichier.

La vue *All Publics* n'est pas éditable, si un symbole doit être complété ou corrigé, la grille indique le nom du fichier qui contient la définition et qu'il convient d'ouvrir pour travailler sur le symbole. (Menu de contexte *Columns, Source*)

Mais il est beaucoup plus intéressant de sélectionner le symbole dans la vue *All Publics* puis le menu *Goto Definition* pour ouvrir le fichier qui contient la définition du symbole.

9.4 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*.

9.4.1 L'accumulateur

L'accumulateur est une mémoire binaire que l'automate utilise activement avec les instructions binaires et quelques instructions mots. L'automate 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.

9.4.2 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 de l'automate.

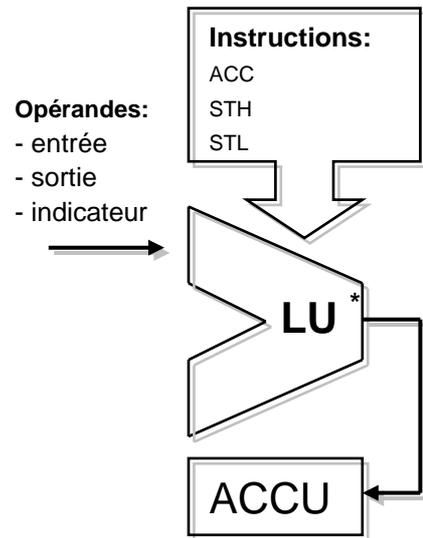
Les instructions binaires font toujours intervenir l'accumulateur. Certaines instructions binaires utilisent l'accumulateur pour y écrire des informations:

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.



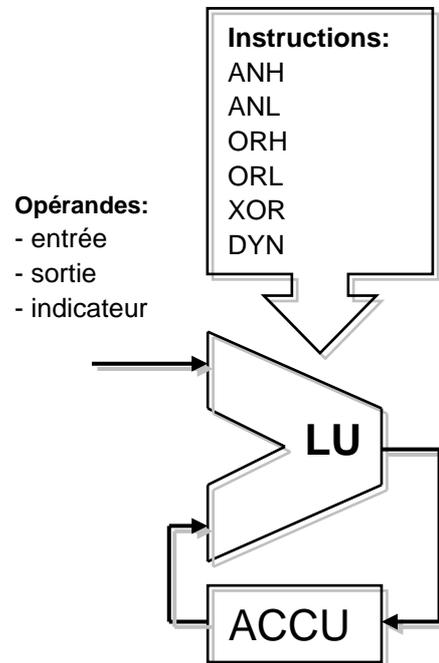
D'autres instructions lisent l'état de l'accumulateur pour effectuer une fonction binaire et placer le résultat dans le même accumulateur:

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.

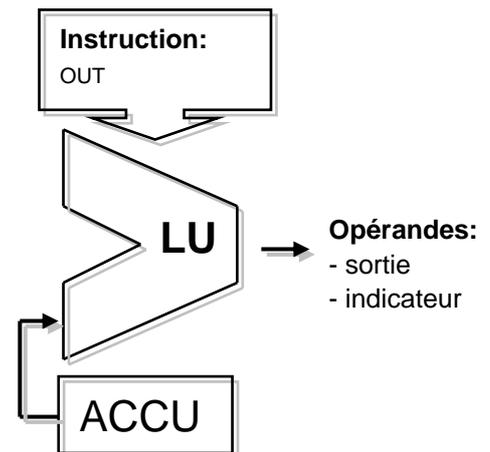
ORH F 100
Lit l'état de l'accumulateur, effectue une fonction logique OU avec l'état de l'indicateur 100. L'accumulateur prend l'état du résultat.

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.



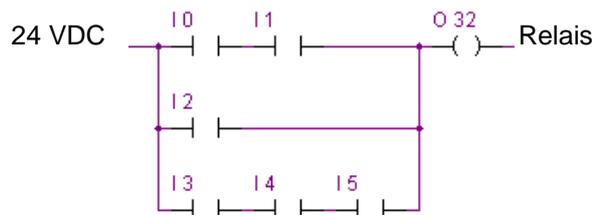
Le résultat d'une équation binaire est toujours sauvegardé dans l'accumulateur. Une instruction *OUT* permet de copier le contenu de l'accumulateur sur une sortie ou un indicateur:

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 :



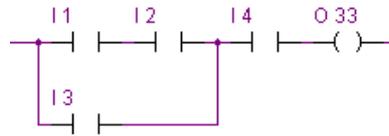
Une équation binaire commence toujours avec une instruction *STH* ou *STL*, puis se succèdent les fonctions *ANH* (*), *ORH* (+), *XOR* nécessaires.

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

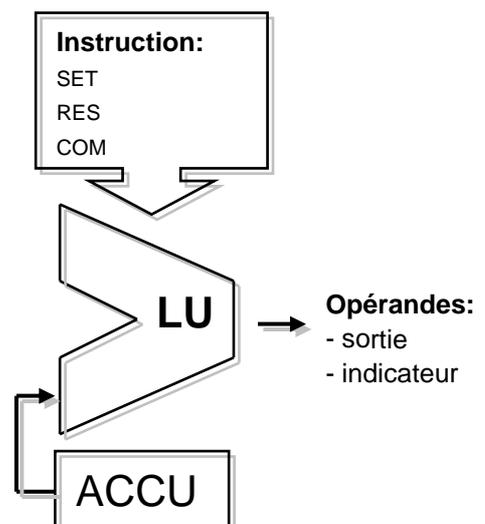
Si l'accumulateur est à l'état haut, la sortie 32 est forcée à l'état haut. Si non la sortie reste dans l'état actuel.

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

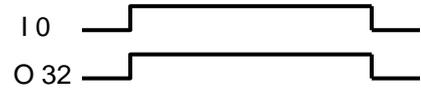
Si l'accumulateur est à l'état haut, l'état de la sortie 33 est inversé. Si non la sortie reste dans l'état actuel.



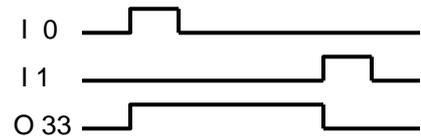
Exemple:

Cet exemple met en évidence les différences entre les instructions OUT, SET, RES, et COM

```
COB 0
    0
    STH I 0
    OUT O 32 ;Copie I 0 sur O 32
```

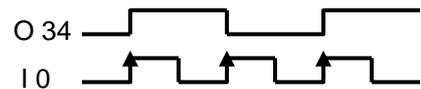


```
    STH I 0
    SET O 33 ;Mémorise un état haut sur
              ; la sortie 33
```



```
    STH I 1
    RES O 33 ;Mémorise un état bas sur
              ; la sortie 33
```

```
    STH I 0 ;Sur le flanc montant de I 0
    DYN F 1
    COM O 34 ;Inverse de l'état de la
              ; sortie 34
```



ECOB

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.

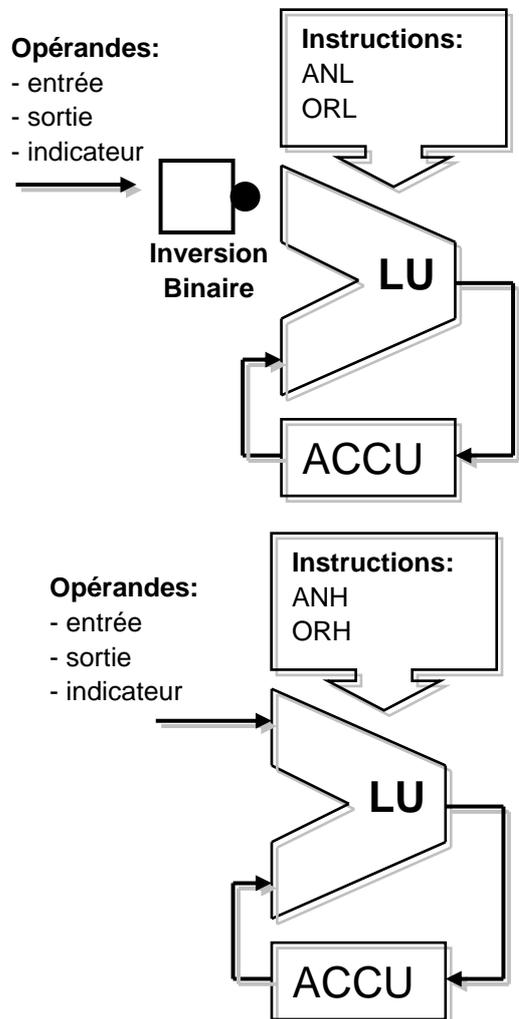
Exemples:

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.



9.4.3 Dynamisation

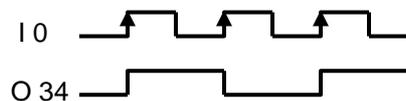
Les instructions binaires utilisent généralement l'état binaire bas ou haut pour effectuer une fonction binaire ou modifier l'état d'une sortie, indicateur.

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.

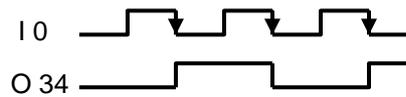
Exemple: détection d'un flanc montant

```
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.

9.4.4 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 de l'automate sont modifiés par les instructions mots et nous informent du résultat obtenu.

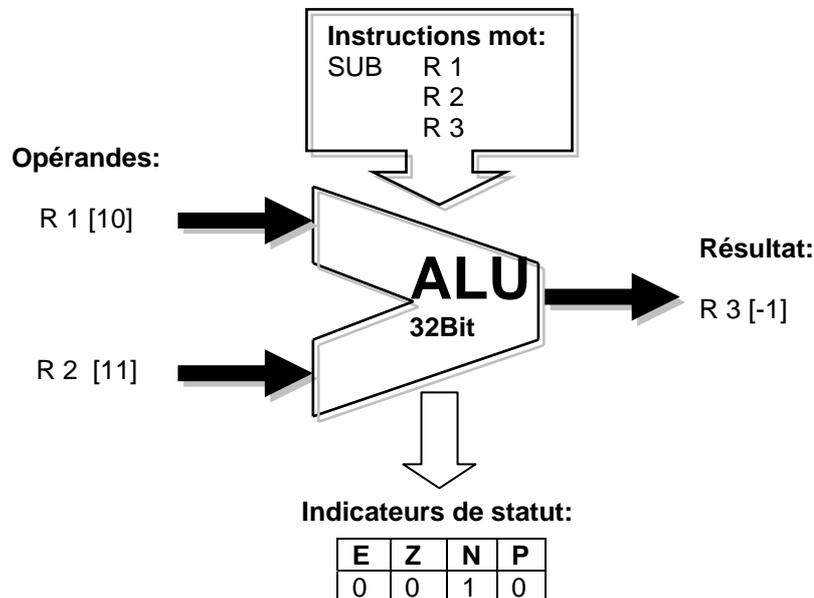
Indicateur positif	P	Est positionné si le résultat est positif.
Indicateur négatif	N	Est positionné si le résultat est négatif.
Indicateur zéro	Z	Est positionné si le résultat est zéro
Indicateur erreur	E	Est positionnée en cas d'erreur

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

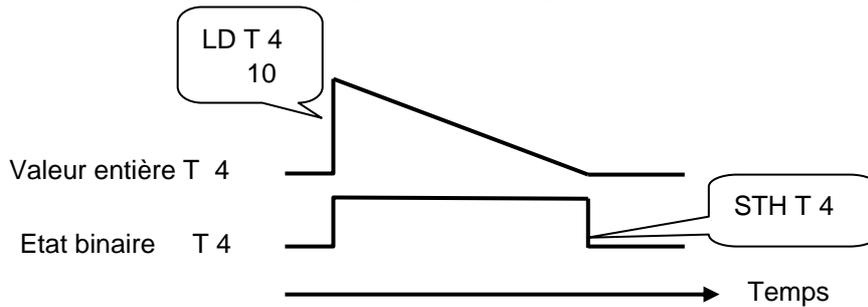
Les indicateurs de statut sont positionnés selon le résultat de la soustraction ($R_3 = R_1 - R_2$). Les valeurs des registres sont présentés entre crochet []. Le résultat de la soustraction est négatif, seul l'indicateur N est positionné.



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

9.4.5 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 ¹. 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.

Charger une temporisation

LD T 4

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

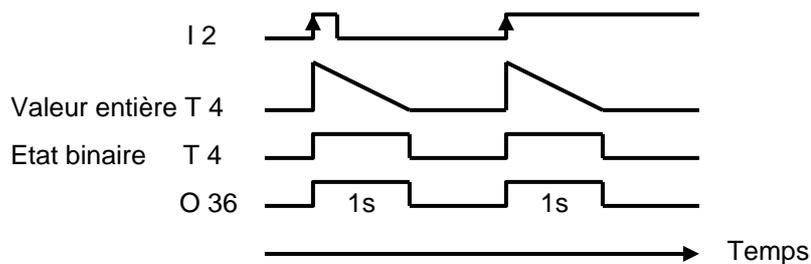
Lecture de l'état du temporisateur.

Utiliser une instruction binaire tel que: STH T 4 , ANH T 4, ORH T 4, ...

Exemple:

Sortir une impulsion d'une seconde sur la sortie 36 pour chaque flanc montant de l'entrée 2

Diagramme des états:



Programme correspondant:

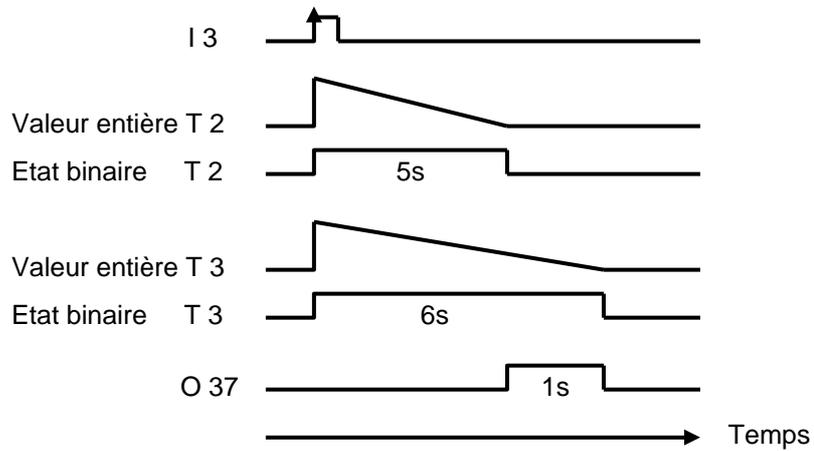
```
COB 0
0
STH I 2 ;La détection du flanc montant de l'entrée 2 ...
DYN F 2 ;...positionne l'accu à l'état haut
LD T 4 ;Si l'accu est haut, charge la temporisation
; pour 10 unités de temps
10
STH T 4 ;Copie l'état logique de la temporisation
; sur la sortie 36
OUT O 36
ECOB
```

¹ La base de temps peut être autre que 1/10^e de seconde (valeur par défaut). Elle peut être modifiée à partir des *Build Options*.

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:

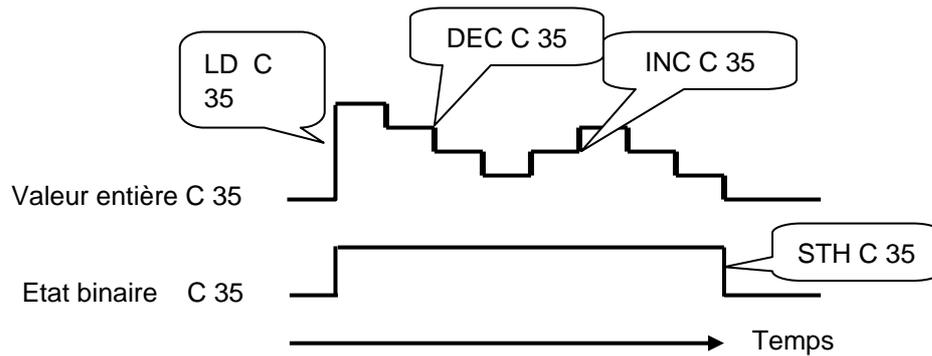


Programme correspondant:

```

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
    
```

9.4.6 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 décrémenteés à partir du programme de l'automate. 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.

Charger un compteur

```
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.

Lecture de l'état du compteur.

Utiliser une instruction binaire tel que:
STH C 35, ANH C 35, ORH C 35, ...

Incrémenter un compteur

```
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.

Décrémenter un compteur

```
DEC C 35
```

Si l'accumulateur est à l'état haut, le compteur 35 est décrémenteé d'une unité. Si non le compteur conserve la valeur actuelle.

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.

9.4.7 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
```

9.4.8 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.

Addition	Soustraction	Racine carrée
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
Multiplication	Division	Comparaison
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
Incrémenter	Décrémenter	Initialiser un registre
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 O 33 ; R 0 = R 1
ACC N
OUT O 34 ; R 0 < R 1
    
```

9.4.9 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

9.4.10 Conversion des registres entiers et flottants

L'automate 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

9.4.11 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:

SEI K 10	SEt Index register	Charge le registre d'index avec une constante 10
INI K 99	IN crement Index register	Incrémente le registre d'index et positionne l'accu à l'état haut tant que: Registre d'index <= K 99
DEI K 5	DE crement Index register	Décrémente le registre d'index et positionne l'accu à l'état haut tant que: Registre d'index >= K 5
STI R 0	ST ore Index register	Copie le registre d'index dans un registre 0
RSI R 0	Re Store 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
```

9.4.12 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		
JR	Saut relatif	Saute quelques lignes en avant ou en arrière par rapport à la ligne où se trouve l'instruction JR
JPD	Saut direct	Saute à un numéro de ligne par rapport au début du bloc (COB,PB,...)
JPI	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
JR		Le saut est toujours effectué à la ligne correspondant au label
JPD		
JPI		

Syntaxe d'une instruction pour un saut conditionnel			
Mnémonique	Condition	Label	Description
JR	H		Si l'accu est haut (High)
JPD	L		Si l'accu est bas (Low)
JPI	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.

```

COB    0
      0
STH    F 1      ;Copie l'état du flag sur l'accumulateur
DYN    F 2      ;Force l'accu à l'état haut sur un flanc
                ; positif du flag F1
JR     L Next   ;Si l'accu est à l'état bas,
                ; sauter au label Next
INC    R 10
Next:  NOP
      ECOB

```

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

```

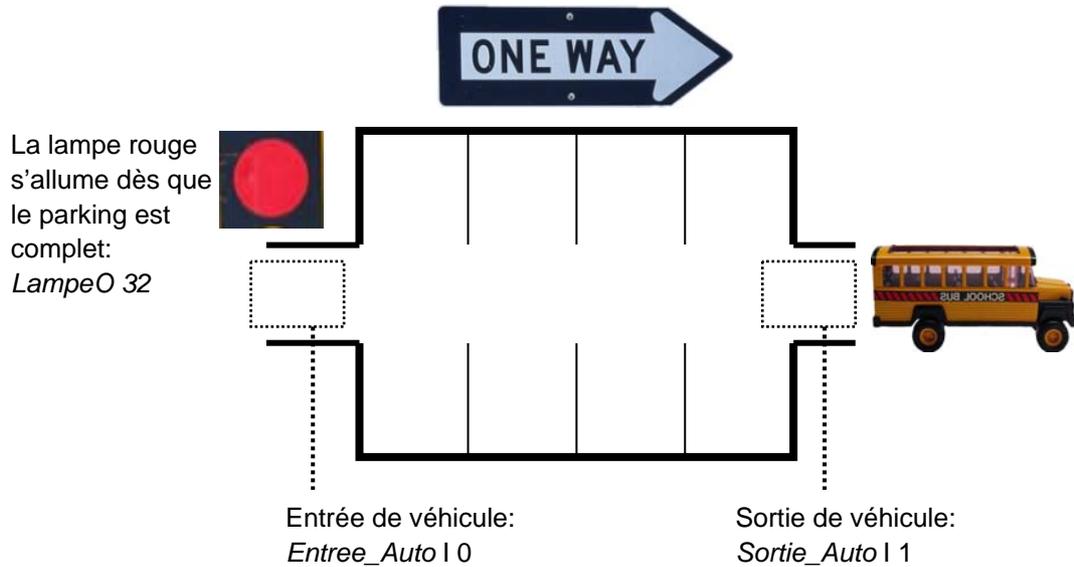
COB    0
      0
LD     R 2      ; Charge le numéro de ligne dans
                ; le registre
      Next
STH    F 1      ; Copie l'état du flag
                ; sur l'accumulateur
DYN    F 2      ; Force l'accu à l'état haut sur un
                ; flanc positif du flag F1
JPI    L 2      ; Si l'accu est bas, sauter au numéro
                ; de ligne défini par le registre 2
INC    R 10
Next:  NOP
      ECOB

```

Le saut indirect offre une grande flexibilité, le programme peut modifier lui même le numéro de ligne pour effectuer le saut.

9.5 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 de l'automate, 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 de l'automate. 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.

Symbol Name	Type	Address..	Comment	Scope
Parking lot.src	ROOT			
Car_incoming	I	0	Gets high when a car comes into the par...	Local
Car_outgoing	I	1	Gets high when a car leaves the parking	Local
Red_light	O	32	Stops new cars at the entry	Local
Number_of_free_slots	C		Counts the number of free parking slots	Local
Dynamise_incoming_car_signal	F		Flag detects the rising edge of the car in...	Local
Dynamise_leaving_car_signal	F		Flag detects the rising edge on the car le...	Local

```

; Cold start organisation block
;-----

XOB 16 ;Program executed at start up
ACC H
LD Number_of_free_slots;Initialize the free slots counter
8 ; with the value 8 (unconditionally)
EXOB ; End of start-up program

; Cyclical Organisation Block
;-----

COB 0 ; Cyclical program
0 ; No supervision time

STH Car_incoming ; A car comes into the parking:
DYN Dynamise_incoming_car_signal ;On the positiv flank of
; incoming signal
DEC Number_of_free_slots ;Decrement the number of
; free parking slots

;-----

STH Car_outgoing ; A car leaves into the parking:
DYN Dynamise_leaving_car_signal ;On the positiv flank of
; outgoing signal
INC Number_of_free_slots ;Increment the number of
; free parking slots

;-----

STL Number_of_free_slots ;if no more free parking
: slots(counter state= Low)
OUT Red_light ; Set the red light

ECOB ; End of Cyclical program

```

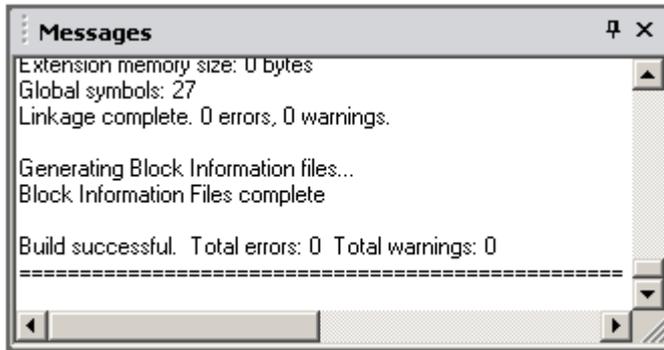
9.6 Construction du programme (Build)



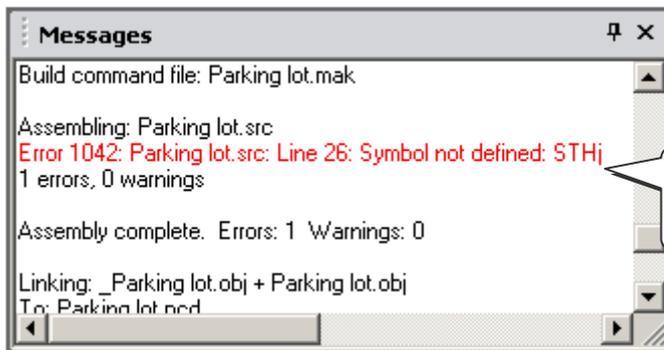
Build All Files

Le programme utilisateur est complètement décrit, mais pas encore utilisable par l'automate. 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 ou le bouton *Rebuild All Files* 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	Car inc	
DYN	Dynamise	incoming_car_signal
DEC	Number	_free_slots
STH	Car_incoming	
DYN	Dynamise_incoming_car_signal	
DEC	Number_of_free_slots	

Correction de l'erreur

9.7 Charger le programme dans l'automate



Download
Program

Le programme d'application est prêt. Il reste à le transférer de l'ordinateur vers l'automate avec le menu *Online, Download Program* ou le bouton *Download Program* de la fenêtre *SAIA PG5 Project Manager*.

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

9.8 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 la même interface que celle employé à l'édition.

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 opérandes et les numéros de lignes du programme.

```

;-----
; Cyclical Organisation Block
;-----
          COB      0                ; Cyclical program
          COB      0                ; No supervision time
000007   COB      0
000008   COB      0
000010   NOP

          STH      Car_incoming     ; A car comes into the park
000011   STH      I|0 0             [0]
          DYN      Dynamise_incoming_car_signal ; On the positiv flank of i
000012   DYN      F 7502           [0]
          DEC      Number_of_free_slots ; Decrement the number of
000013   DEC      C 1400           [8]

```

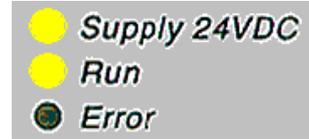
9.8.1 Go On/Offline, Run et Stop

Le mode *Online* permet d'établir la communication avec l'automate 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 de l'automate. A la sélection du bouton *Run*, la lampe RUN est allumée, l'automate exécute le programme utilisateur.

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



```

      STH   Car_incoming           ; A car comes into the par
000011 STH   I|0 0                 [0]
      DYN   Dynamise_incoming_car_signal ; On the positiv flank of
000012 DYN   F 7502                [0]
      DEC   Number_of_free_slots     ; Decrement the number o
000013 DEC   C 1400                [8]      A0 20 N0 P1 E0 IX0000
    
```

9.8.2 Mode pas-à-pas



Run to Cursor

Si l'automate est en mode run, marquer la première ligne à observer en mode pas-à-pas est sélectionner le bouton *Run to Cursor*. L'automate 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:



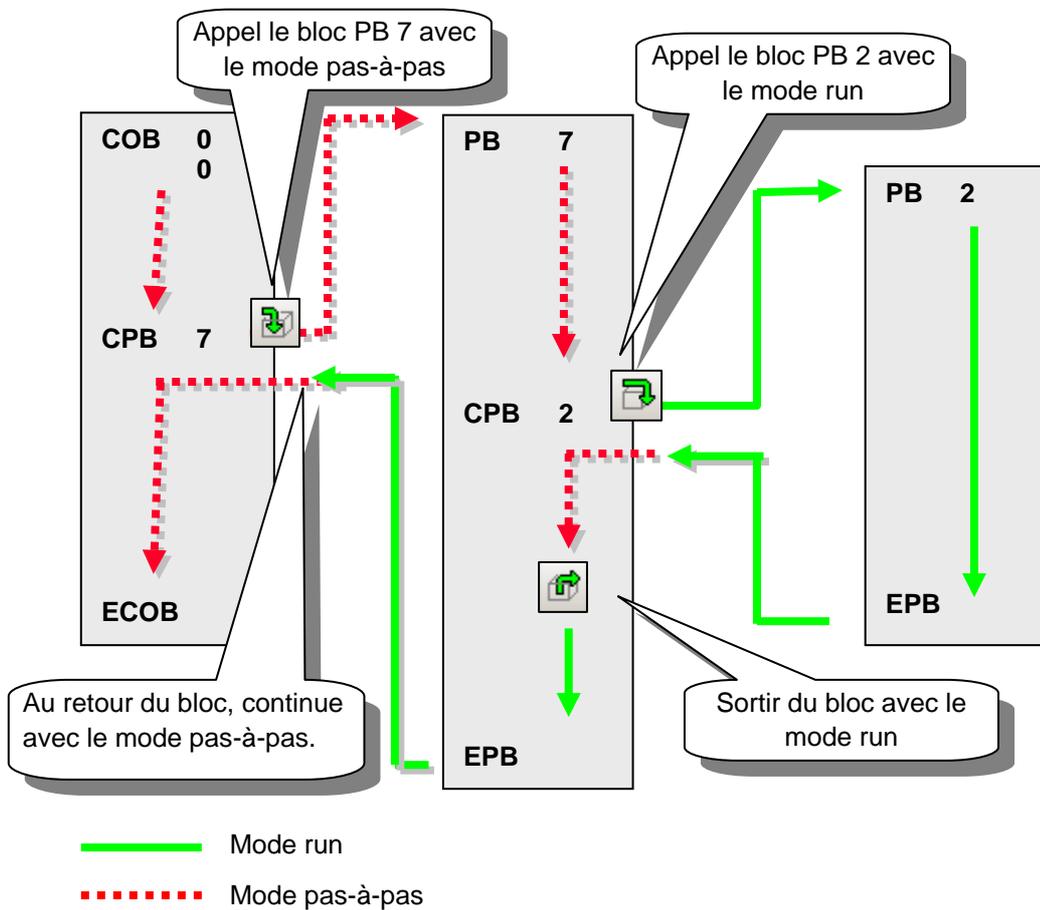
Entrer dans la structure pour l'analyser en pas-à-pas



Traiter la structure appelée avec le mode run et continuer en mode pas-à-pas juste après le retour au bloc qui a 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 a effectué l'appel.



```

000014  STH  Car_outgoing           ; A car leaves into the p
000015  DYN  Dynamise_leaving_car_sig ; On the positiv flank of
          F 7503                   [ 0 ]
    
```

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.

9.8.3 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

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*.

*Set/Clear
Breakpoints*

Type	Address	Condition	Value
Counter	1400	=	0
Counter	1400	>	4
Output	32	=	0

A l'aide de la fenêtre ci-dessus, définir le type et l'adresse 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 l'automate en mode Run conditionnel. La LED *Run* de l'automate clignote et le bouton *Run* alterne les couleurs vert et rouge.

L'automate 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 l'automate 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 l'automate en mode RUN. La LED *Run* de l'automate est allumée et le bouton *Run* de couleur verte.
- Le bouton *Clear-Stop*, force l'automate en mode Stop. La LED *Run* de l'automate est éteinte et le bouton *Run* de couleur rouge.

Si plusieurs points 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



Run to Cursor

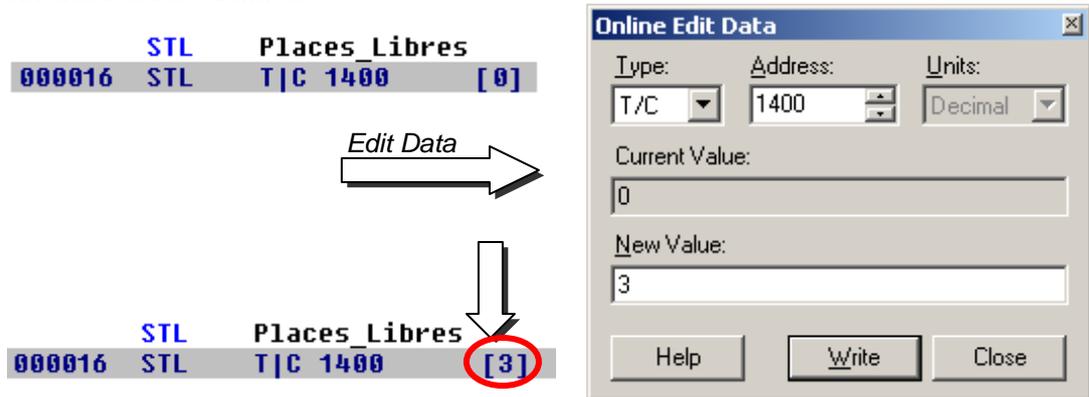
La sélection d'une ligne du programme puis du menu ou bouton *Online, Run to, Cursor* permet de stopper le programme sur la ligne sélectionnée pour en suite continuer en mode pas-à-pas.

9.8.4 Modification de programme en ligne

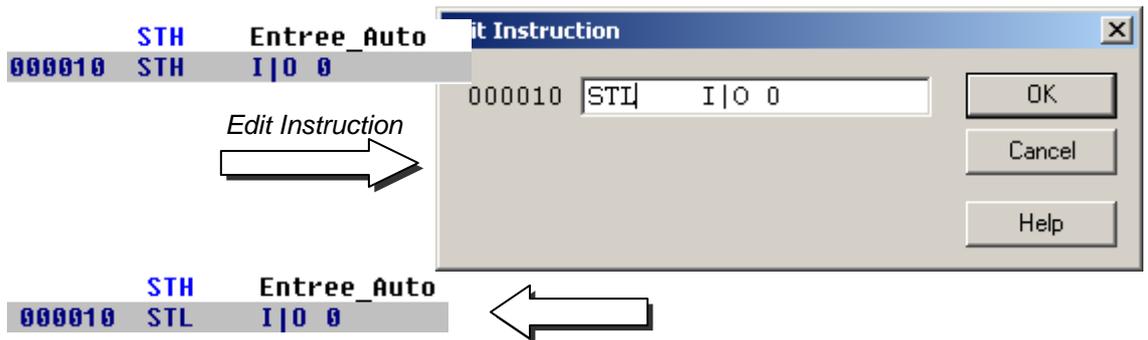
Lors d'un test pas-à-pas de programme, il est utile de modifier les états/valeurs de certaines opérandes/symboles et de vérifier le comportement du programme sous certaines conditions.

Sélectionner une des lignes actives (grise) à l'aide de la souris et presser le bouton droite pour afficher le menu de contexte.

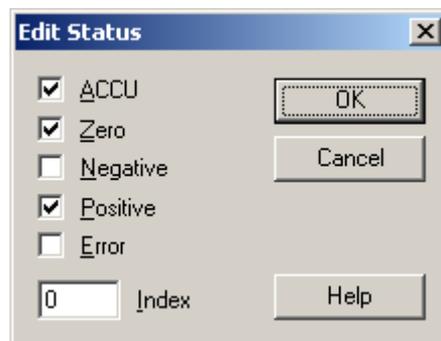
Le menu de contexte *Edit Data* permet de modifier l'état/valeur de l'opérande de l'instruction sélectionnée.



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*.



9.8.5 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*

Placer la souris sur le bouton au début de la ligne, presser le bouton gauche de la

Glisser la souris sur le dans la fenêtre Watch window

Symboles avec leurs commentaires et leurs états/valeurs.

Start/Stop Monitoring

Symbol Name	Address	Value	Modify Value	Chart	Module	Symbol Comment
Car_incoming	I 0	0			Parking lot.src	Gets high when a car comes
Car_outgoing	I 1	0			Parking lot.src	Gets high when a car leaves
Red_light	O 32	0			Parking lot.src	Stops new cars at the entry
Number_of_free_slots	C 1400	8			Parking lot.src	Counts the number of free
Dynamise_incoming_...	F 7502	0			Parking lot.src	Flag detects the rising edge
Dynamise_leaving_c...	F 7503	0			Parking lot.src	Flag detects the rising edge

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	Chart	Module	Symbol Comment
Car_incoming	I 0	0			Parking lot.src	Gets high when a car comes
Car_outgoing	I 1	0			Parking lot.src	Gets high when a car leaves
Red_light	O 32	0			Parking lot.src	Stops new cars at the entry
Number_of_free_slots	C 1400	8			Parking lot.src	Counts the number of free
Dynamise_incoming_...	F 7502	0			Parking lot.src	Flag detects the rising edge
Dynamise_leaving_c...	F 7503	0			Parking lot.src	Flag detects the rising edge

9.9 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

Avec les entrées et sorties analogiques, il est nécessaire de faire usage de routines spécifiques à chaque type de module analogique, elles assurent le multiplexage des canaux ainsi que la conversion AD/DA. Ces programmes sont supportés soit à l'aide du programme IL, soit par le *Device Configurator* et la fonction *media mapping*. Voir description du *Device Configurator*.

9.9.1 Exemple pour les entrées analogiques PCD2.W340

Si l'automate 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 l'automate
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.

9.9.2 Exemple pour les sorties analogiques PCD2.W610

Les sorties fonctionnent de manière similaire aux entrées.

Si l'automate 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 l'automate
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.com>

Table des matières

10	OUTILS COMPLÉMENTAIRES	3
10.1	Tranfert de données (<i>Data Transfer Utility</i>)	4
10.1.1	Utilisation du transfert de données	4
10.1.2	Démarrer <i>Data Transfert</i>	4
10.1.3	Sauvegarde rapide des données	4
10.1.4	Restaurer les données	5
10.1.5	Sauvegarde à l'aide d'un fichier script	5
10.1.6	Restaurer les données à l'aide d'un fichier script	6
10.1.7	Options de sauvegarde	6
10.1.8	Sauvegarde de données avec le mode ligne de commande	7
10.2	Fenêtre de surveillance (<i>Watch Window</i>)	8
10.2.1	Ouvrir une fenêtre <i>Watch Window</i>	8
10.2.2	Ajouter des données à la fenêtre <i>WatchWindow</i>	9
10.2.3	Start /Stop Monitoring	10
10.2.4	Modification en ligne des données	10
10.2.5	Format d'affichage	10
10.2.6	Watch Window et applications avec plusieurs devices	11
10.3	Le configurateur en ligne (<i>Online Configurator</i>)	15
10.3.1	Ajuster l'horloge de l'automate	15
10.3.2	L' historique PCD	16
10.4	Mise à jour du firmware. (<i>Firmware Downloader</i>)	17
10.5	Menus utilisateurs	18

10 Outils complémentaires

Le PG5 met à votre disposition plusieurs utilitaires complémentaires pour assurer divers petits services.

10.1 Transfert de données (*Data Transfer Utility*)

10.1.1 Utilisation du transfert de données

Cet outil est utilisé pour sauvegarder les états/valeurs des données de l'automate dans un fichier ASCII (*.dt5) ou de les restituer du fichier dans la mémoire de l'automate.

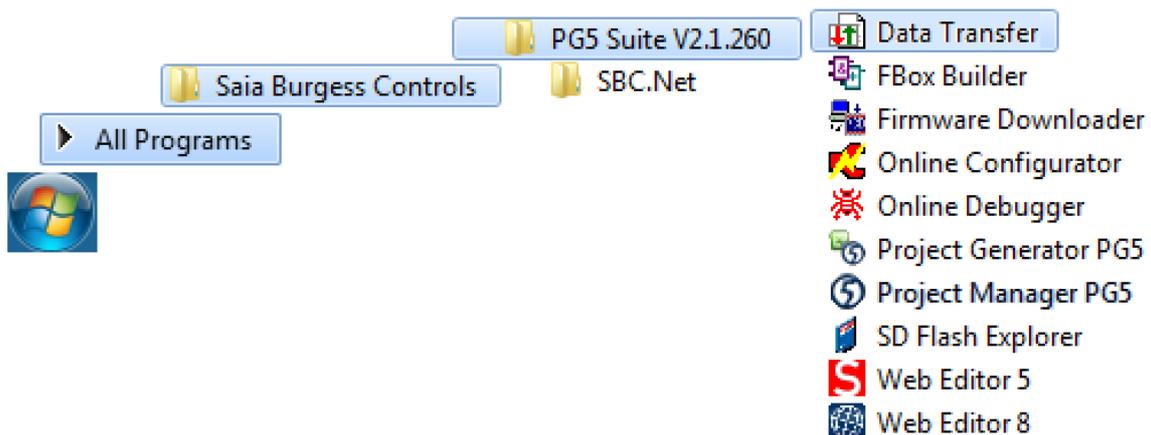
Les données transférées par cet outil sont les suivantes: entrées, sorties, indicateurs, temporisateurs, compteurs, registres, blocs de données et textes.

Attention, le programme et les configurations matériels de l'automate ne sont pas sauvegardés 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.

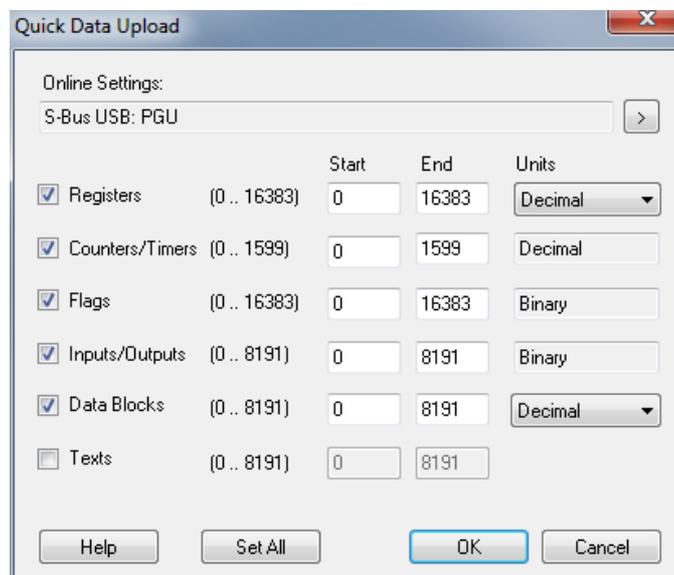
10.1.2 Démarrer *Data Transfer*

Démarrer le programme avec le menu:

Start → Programs → Saia Burgess Controls → PG5 Suite 2.1 → Data Transfer



10.1.3 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.

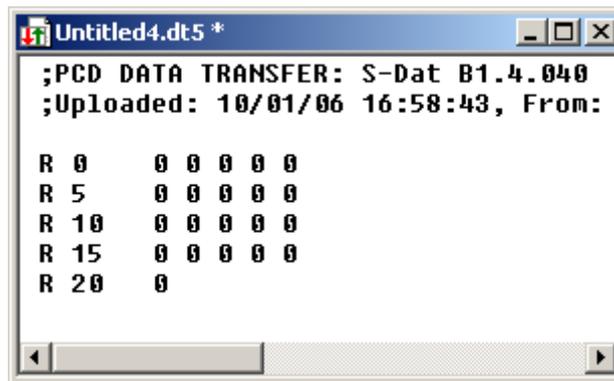
Sélectionner les types de données à sauvegarder, les plages d'adresses, éventuellement le format d'affichage pour les registres.

Sélectionner le bouton OK pour charger les données.

Si un message semblable à celui-ci est affiché, veuillez vérifier les paramètres de communications à l'aide du menu *Online, Online Settings* 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:



Le fichier de données peut être édité avec de nouvelles valeurs puis sauvegardés avec le menu *File, Save* ou le bouton *Save*.

10.1.4 Restaurer les données



Les fichiers déjà sauvegardés peuvent être à nouveau affichés avec le menu *File, open* ou le bouton *Open*

Open

Si nécessaire, les valeurs du fichier peuvent être éditées par l'utilisateur.

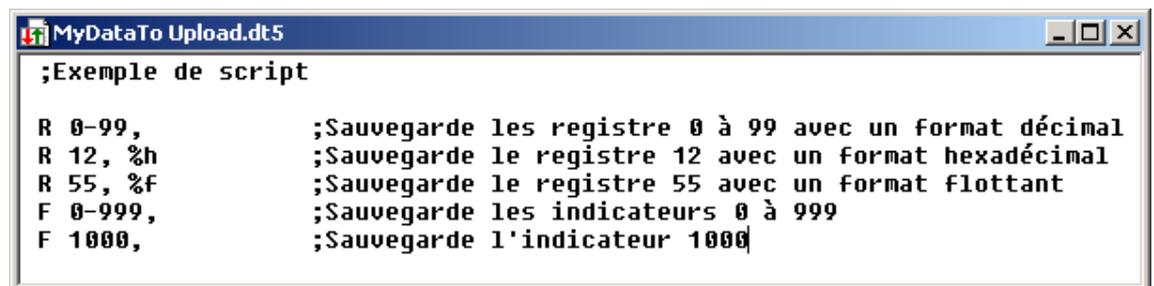


*Download
To PCD*

Les données sont restaurées dans la mémoire de l'automate avec le menu *Online, Download Data To PCD* ou le bouton *Download to PCD*.

10.1.5 Sauvegarde à l'aide d'un fichier script

Si nécessaire, la liste des données à sauvegarder peut être éditée dans un fichier script. Exemple:





Upload
From PCD

Sélectionner le menu *Online, Upload Data from PCD ...* ou le bouton *Upload from PCD* pour charger les données de l'automate 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*.

10.1.6 Restaurer les données à l'aide d'un fichier script

Un script offre aussi la possibilité d'éditer les données à restaurer. Exemple:



Download
To PCD

```
MyDataToDownload.dt5 *
;Exemple de script

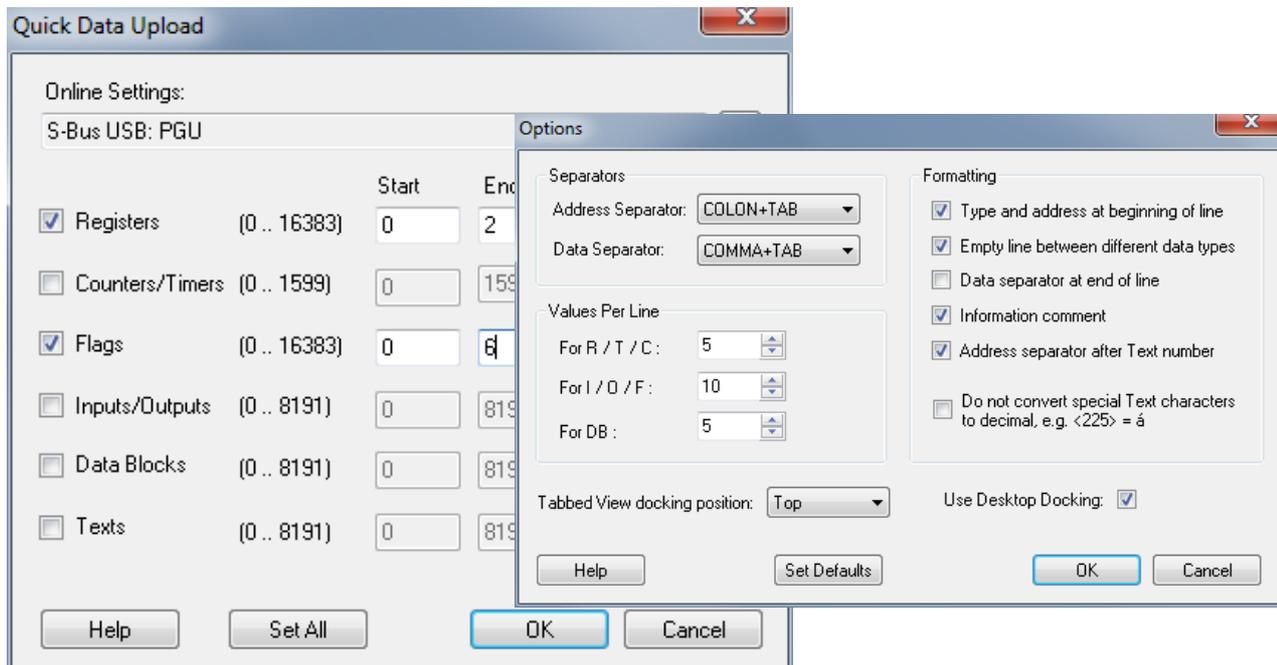
R 0-99, 0 ;Charge les registre 0 à 99 avec zéro
R 12, 32h ;Charge le registre 12 avec 32 hexadécimal
R 55, 64.3 ;Charge le registre 55 avec 64.3 flottant
F 0-999, 0 ;Charge les indicateurs 0 à l'état bas
F 1000, 1 ;Charge l'indicateur 1000 à l'état haut
```

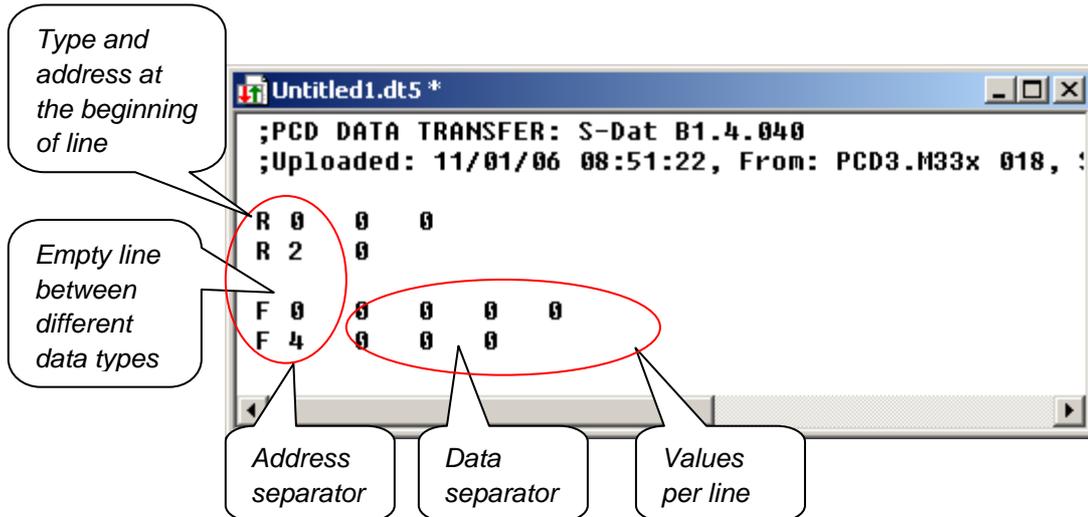
Sélectionner le menu *Online, Download Data To PCD ...* ou le bouton *Download To PCD* pour charger les données du script dans PCD.

10.1.7 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*.





10.1.8 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][/I=nnn][/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
	<i>Format : <type><début>[-<fin>][unités]</i>
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

10.2 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.

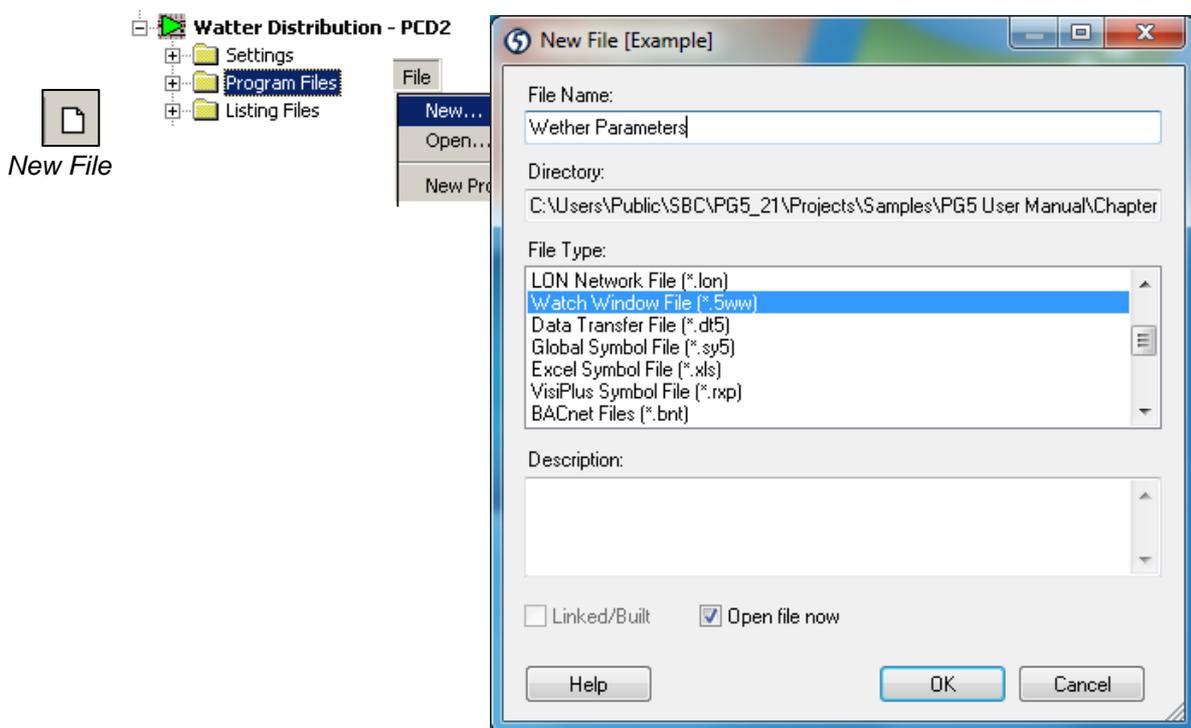
10.2.1 Ouvrir une fenêtre *Watch Window*



Watch Window

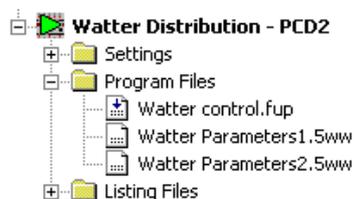
La fenêtre *Watch Window* est affichée à la sélection du menu *Online, Watch Window* ou du bouton *Watch Window*.

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*.



10.2.2 Ajouter des données à la fenêtre *WatchWindow*

Glisser les symboles du programme ou de l'éditeur de symboles vers la fenêtre *Watch Window*.

Placer la souris sur le bouton au début de la ligne, presser le bouton gauche de la souris

Glisser la souris sur le dans la fenêtre Watch window

Symboles avec leurs commentaires et leurs états/valeurs.

Start/Stop Monitoring

Symbol Name	Address	Value	Modify Value	Chart	Module	Symbol Comment
Car_incoming	I 0	0			Parking lot.src	Gets high when a car comes
Car_outgoing	I 1	0			Parking lot.src	Gets high when a car leaves
Red_light	O 32	0			Parking lot.src	Stops new cars at the entry
Number_of_free_slots	C 1400	8			Parking lot.src	Counts the number of free
Dynamise_incoming...	F 7502	0			Parking lot.src	Flag detects the rising edge
Dynamise_leaving_c...	F 7503	0			Parking lot.src	Flag detects the rising edge

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

Symbol	Address	Value	Modify Value	Symbol Comment
DailyTimer	O 32	1		Daily Timer
ONTIME	R 2005	60000		Switch on time

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

10.2.3 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*

10.2.4 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*.

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	Modify Value
HMS	R 2003	102816	
DailyTimer	O 32	1	
ONTIME	R 2005	83000	63000
OFFTIME	R 2004	180000	180000

10.2.5 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	▶

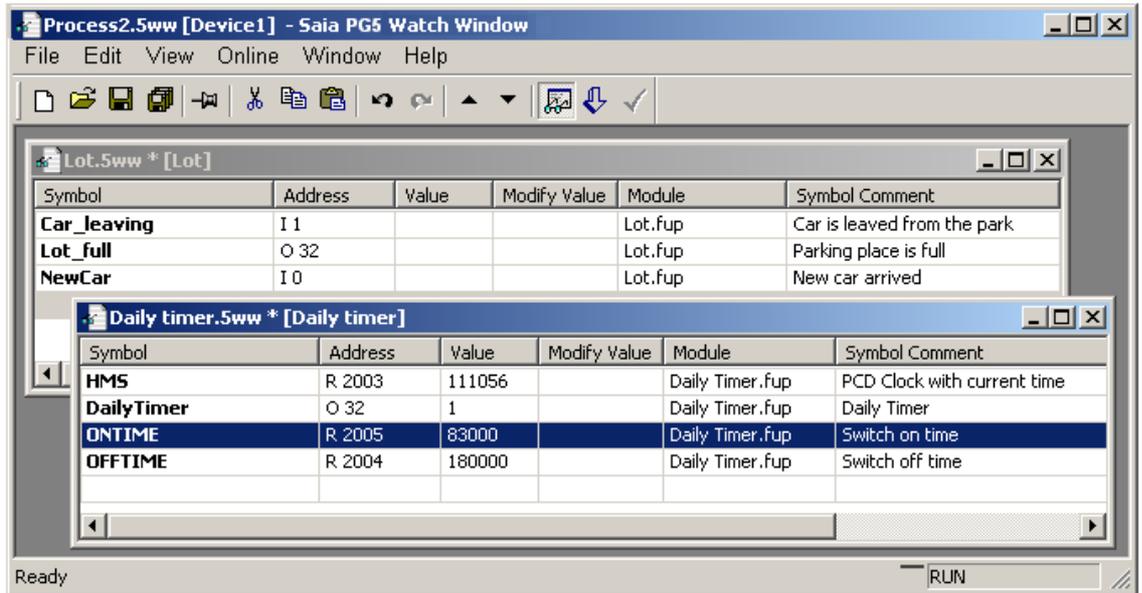
Float	Ctrl+Shift+F
ASCII	Ctrl+Shift+I
Binary	Ctrl+Shift+B
• Hexadecimal	Ctrl+Shift+H
Decimal	
IP	
IEEE Float	
Double IEEE Float	

Address	Value
O 32	0
R 2003	113245
R 2004	0002C6F0H
R 2005	53000

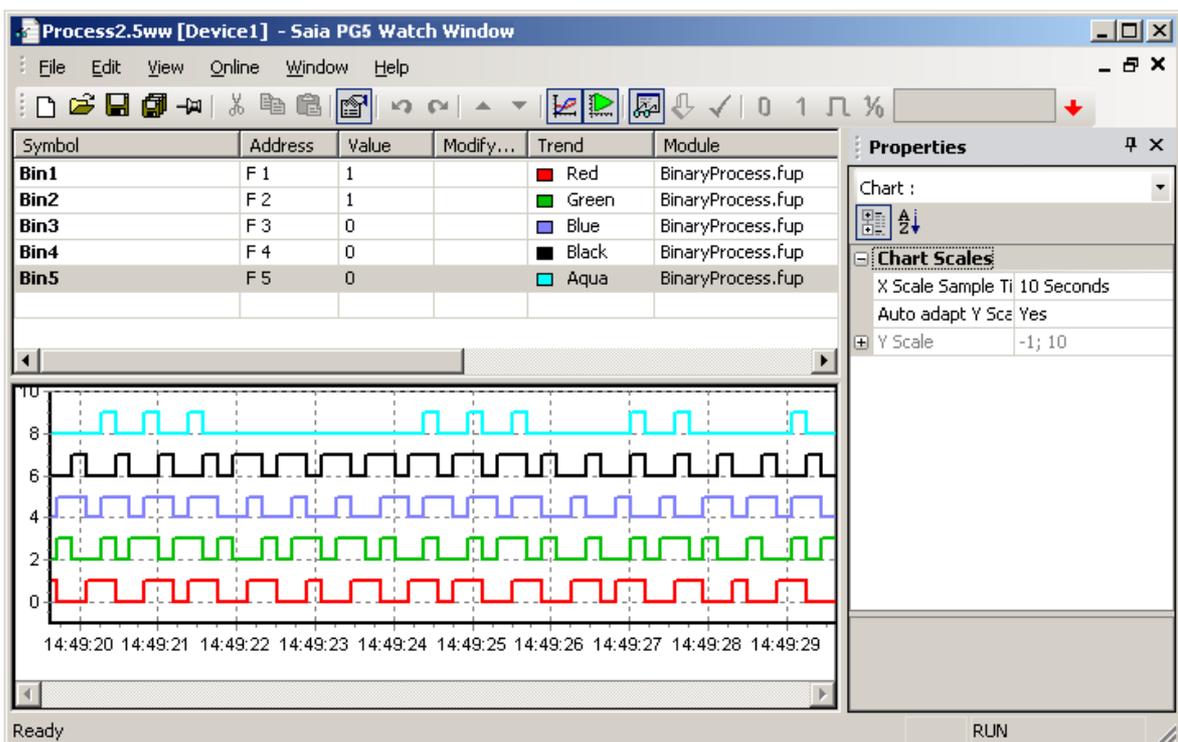
10.2.6 Watch Window et applications avec plusieurs devices

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 Watch Window ouvert fait par défaut usage des *Online settings* du device au quel il appartient. Il est ainsi possible de visualiser sur le réseau de communication, les données des différents PCD disponibles dans le projet.



10.2.7 Fonctionnalité 'Trend' – affichage des valeurs on-line





Show/Hide
Trend

La 'Watch Window' peut afficher un maximum de 8 valeurs: registres, flags etc... Pour afficher plus de 8 valeurs, il suffit d'ouvrir un nouveau fichier 'Watch Window'.

Pour afficher les valeurs d'un symbole dans un graphe, sélectionner le bouton 'Show/Hide Trend', puis dans la grille des symboles, assigner une couleur aux symboles à afficher sous la colonne 'Trend'. Activer l'affichage des courbes et leur actualisation à l'aide du bouton correspondant dans la barre des outils.



Start/Pause
Trend Update

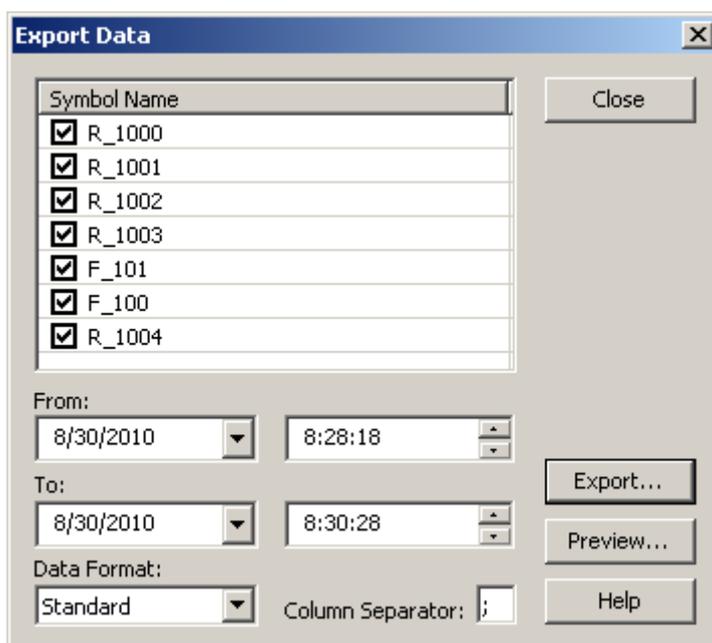
En cliquant sur le graphe, la période d'échantillonnage et les valeurs des échelles horizontales et verticales sont affichées dans la fenêtre des propriétés et pourront être ajustées aisément

10.2.8 Fonctionnalité 'Log' – enregistrement des valeurs

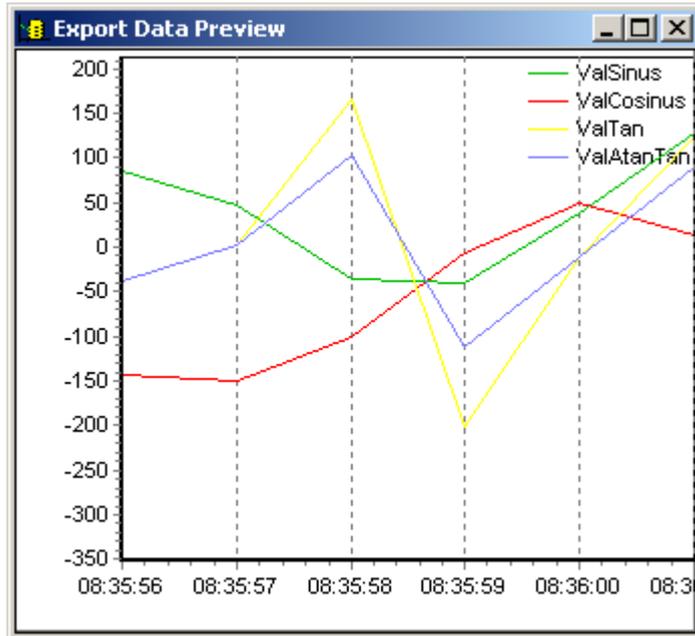
La fonctionnalité *Log* - enregistrement des valeurs affichées dans la 'Watch Window' - peut être activée en ouvrant la fenêtre des propriétés, depuis le menu 'View->Properties Window', puis sélectionner le ou les symboles dans la grille des symboles et ajuster la propriété 'Logging Enable' à 'Yes' pour enregistrer les valeurs. Les valeurs enregistrées pourront être exportées

Logging	
Enable	No
Sample Rate	1 Second

Le menu "Online -> Export Data" permet de sélectionner les valeurs de symboles à exporter et de déterminer la période d'enregistrement.



Le bouton 'Preview' permet d'afficher les courbes des valeurs enregistrées.



Le bouton 'Export' permet de sauvegarder les valeurs enregistrées dans un fichier. Le nom du fichier ainsi que l'emplacement sont à définir dans la fenêtre de dialogue affichée.

10.2.9 Afficher des symboles avec des amplitudes différentes dans le même graphe

Si les valeurs des symboles affichées dans le graphique sont d'amplitudes différentes, le symbole ayant une grande d'amplitude utilisera toute l'échelle verticale contrairement aux symboles ayant une petite d'amplitude qui n'utiliseront qu'une petite partie de l'échelle verticale.

Pour améliorer la visibilité des valeurs ayant une faible amplitude, deux possibilités sont offertes :

1. Un facteur d'échelle - 'Trending scale factor' peut être défini dans la fenêtre des propriétés, associé à chaque symbole. Ce facteur permet d'amplifier ou de réduire l'amplitude des valeurs affichées dans le graphe. Lors de la lecture de la valeur affichée dans le graphique, il faudra alors tenir compte de ce facteur afin d'obtenir la bonne valeur.
2. Une deuxième échelle verticale - 'Second Y axis' – peut être ajoutée sur la droite du graphe. Pour ce faire, sélectionner le symbole dans la grille et ajuster le paramètre 'Axis' dans la catégorie 'Trending'.

Note:

Le graphique est en général plus lisible si l'échelle verticale est ajustée automatiquement à l'amplitude des valeurs des symboles. Pour ajuster l'échelle de façon automatique, afficher les paramètres du graphe dans la fenêtre des propriétés en cliquant sur le graphe, puis ajuster le paramètre 'Auto adapt Y Scale' à 'Yes', ceci pour les axes placés à gauche ou à droite du graphe.

10.2.10 'Trend' contenant plusieurs symboles binaires

Si le graphe – 'Trend' - contient des symboles ayant la même amplitude sur l'échelle verticale, il est possible de définir un offset qui, ajouté à la valeur, améliorera la lisibilité du graphe. Cet offset est défini automatiquement pour les symboles binaires - flags. Après avoir sélectionné le symbole correspondant dans la grille ou dans le graphe, l'offset peut être ajusté dans la grille des propriétés à l'aide du paramètre '*Offset*', défini dans la catégorie '*Trending*'.

10.3 Le configurateur en ligne (*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 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

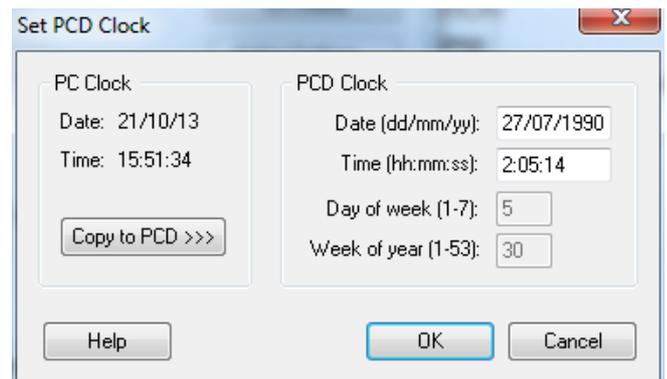
Si les paramètres de communications sont correctement sélectionnés avec le bouton *Settings*

10.3.1 Ajuster l'horloge de l'automate



Sélectionnez le bouton *Online Configurator* de la fenêtre *SAIA PG5 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*



10.3.2 L' historique PCD

Hjstory...

L'historique protocole 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.

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
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

Buttons: Help, Clear History, Close

Labels with arrows:

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

Remarques:

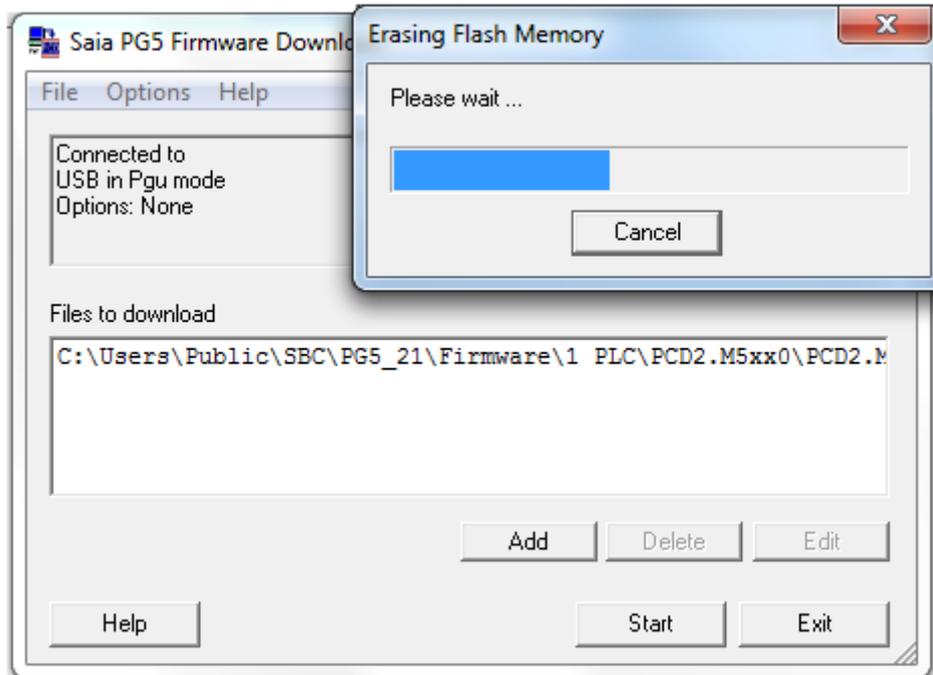
- Si l'erreur est imputable à une ligne de programme, celle-ci est précisée. Si non, elle est affichée en hexadécimal.
- Le XOB 0 n'apparaît que s'il est programmé.

10.4 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 ¹⁾, le firmware peut être mis à jour par un échange d'EPROM.

Seul les firmwares des automates les plus récents peuvent être rechargés en mémoire flash avec l'aide d'un petit utilitaire disponible avec le menu *Tool, Firmware downloader* du gestionnaire de projet.



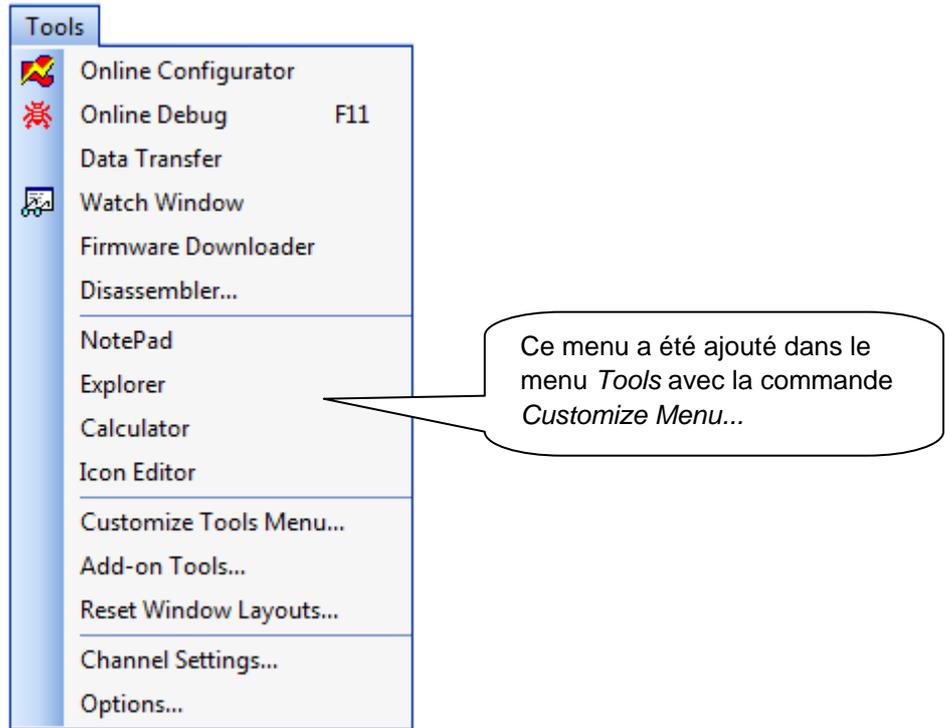
Marche à suivre:

- 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.

1) Les automates PCD1, PCD2.M110/120/150

10.5 Menus utilisateurs

Le menu *Tools* de la fenêtre *SAIA PG5 Project Manager* peut être étendu avec des raccourcis vers vos programmes favoris.



Pour ajouter un raccourci, procéder comme suit:

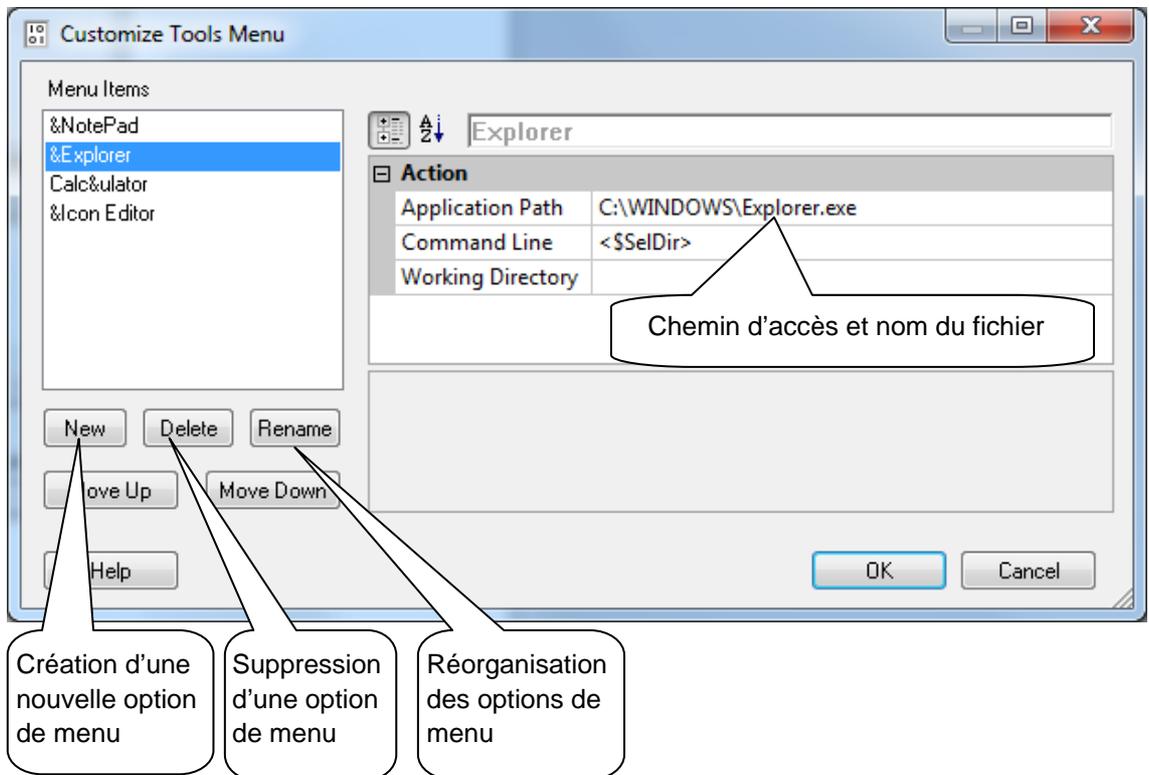


Table des matières

TABLE DES MATIERES	1
11 SAIA PCD NETWORKS (S-NET)	2
11.1 Au sommaire des prochains chapitres	2
11.2 Choix du réseau	3
11.2.1 Services supportés	3
11.2.2 Caractéristiques techniques	3

11 Saia PCD Networks (S-Net)

11.1 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 PG5 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.

11.2 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 :

11.2.1 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.

11.2.2 Caractéristiques techniques

11.2.2.1 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.

11.2.2.2 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 !

11.2.2.3 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.

11.2.2.4 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.

11.2.2.5 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
Caractéristiques					
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 PCD	Saia PCD	Saia PCD	Saia PCD	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
Domaine d'application	Industrie, bâtiment				

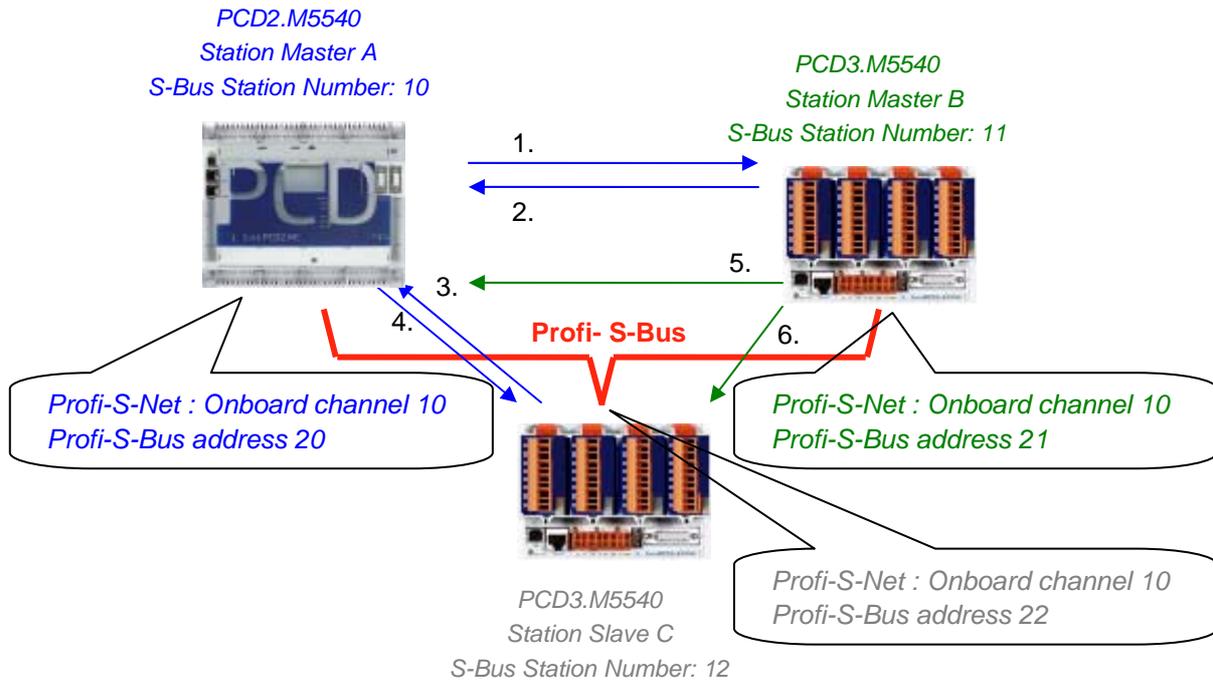
Table des matières

TABLE DES MATIERES	1
12 PROFI-S-BUS	2
12.1 Exemple de réseau Profi-S-Bus.	2
12.2 Exemples d'échanges de données Profi-S-Bus.	2
12.3 Le projet PG5.....	3
12.4 Paramètres du <i>Device Configurator</i>	3
12.4.1 Définir le type d'automate	3
12.4.2 Définir le numéro de station S-Bus sur le réseau.....	3
12.4.3 Définir le canal de communication Profi-S-Bus.....	4
12.4.4 Charger les paramètres du <i>Device Configurator</i> sur l'automate.....	5
12.5 Programme Fupla.....	5
12.5.1 Assigner le canal à l'aide d'une Fbox SASI.....	5
12.5.2 Assigner le canal maître	6
12.5.3 Assigner le canal esclave.....	6
12.5.4 Principe d'échanges des données sur un réseau multi-maîtres.	6
12.5.5 Echange de données entre les stations maîtres et esclaves	7
12.5.6 Diagnostiques	8
12.6 Programme II	11
12.6.1 Assigner le canal maître à l'aide d'une instruction SASI	11
12.6.2 Assigner le canal esclave.....	11
12.6.3 Principe d'échanges des données sur un réseau multi-maîtres.	11
12.6.4 Echange de données entre les stations maîtres et esclaves	12
12.6.5 Diagnostiques	13
12.7 Fonction Gateway.....	15
12.7.1 Application	15
12.7.2 Configuration de la fonction Gateway PGU.....	16
12.7.3 Configuration d'un Gateway Slave port esclave supplémentaire	18
12.7.4 Timings de communication	19
12.8 Autres références	20

12 Profi-S-Bus

Cet exemple présente comment échanger quelques données comme des registres et indicateurs entre les automates raccordés sur le réseau Profi-S-Bus.

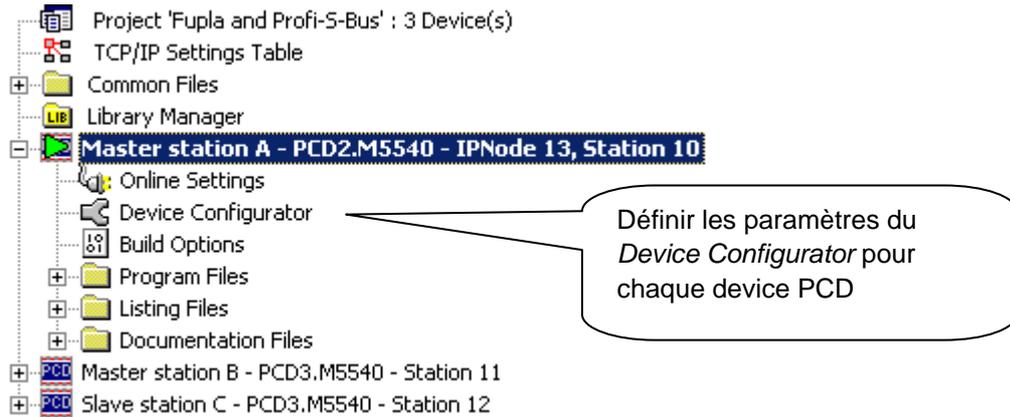
12.1 Exemple de réseau Profi-S-Bus.



12.2 Exemples d'échanges de données Profi-S-Bus.

	Master with data exchanges	Data on the network	Passive master or slave
	Master station A		Master station 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
			Slave station C
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
			Master station B
5	Temperature1 .. 4 Dynamic registers	Write the temperature measures to the slave C	Master_B.Temperature1 .. 4 R 100 .. 104
			Slave station C
6	Temperature1 .. 4 Dynamic registers	Write the temperature measures to the master A	Master_B.Temperature1 .. 4 R 100 .. 104

12.3 Le projet PG5



Le *Saia PG5 Project Manager* donne une vue de l'ensemble des automates d'un projet d'application ainsi que des paramètres pour les réseaux de communication. Nous commencerons par ajouter un device dans le projet pour chaque station disponible sur le réseau.

12.4 Paramètres du *Device Configurator*

Les paramètres du *Device Configurator* sont identiques pour une station maître et esclaves.

12.4.1 Définir le type d'automate

Device	
Type	Description
PCD2.M5540	CPU with 1M Bytes RAM, 8 I/O slots, 2 communication slot.

Device Type

Définir le type d'automate

12.4.2 Définir le numéro de station S-Bus sur le réseau

S-Bus	
S-Bus Support	Yes
Station Number	10

Device properties:

Station Number

Le numéro de station S-Bus est commun à tous les canaux de communication de l'automate.

12.4.3 Définir le canal de communication Profi-S-Bus

Onboard Communications	
Location	Type
Onboard	RS-232/RS-485
Onboard	RS-485/S-Net
Onboard	USB
Onboard	Ethernet

Profi-S-Bus	
Port Number Profi-S-Bus	10
Enabled Profi-S-Bus	Yes
Channel	10
Full Protocol (PGU) Profi-S-Bus	Yes
Slave	Yes
Address	20
Use S-Net Configuration	No
S-Net File Name	
Baud Rate Profi-S-Bus	1.5 MBd
Bus Profile	S-Net

Onboard Communication, properties:

Full Protocol (PGU) Profi-S-Bus

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.

Slave + PGU

Supporte l'échange de données avec les stations maîtres, systèmes de supervision et terminaux. Mais supporte aussi l'outil de programmation et de mise en service PG5..

Slave

Supporte uniquement l'échange de données avec les stations maîtres, systèmes de supervision et terminaux.

Address

Numéro de station Profi-S-Bus lié au canal

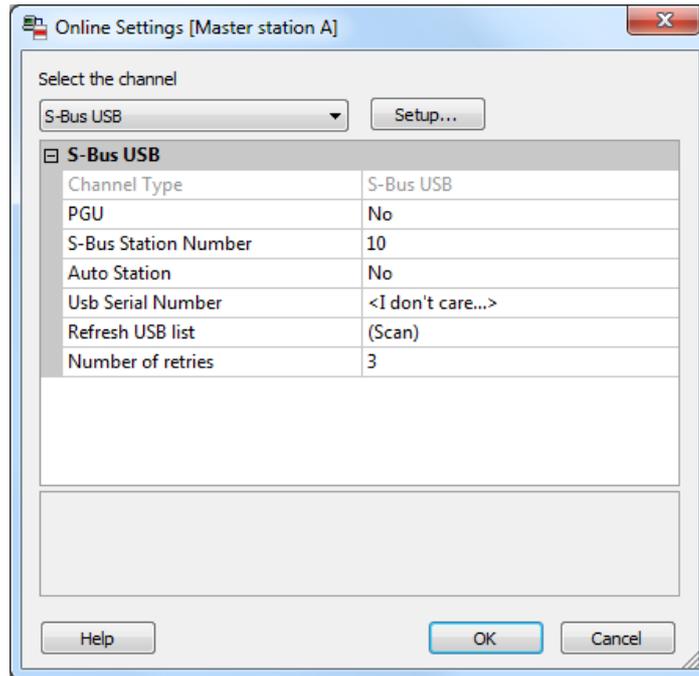
Baud Rate Profi-S-Bus

Vitesse de communication, doit être identique sur toutes les stations du réseau.

Bus Profile

Les timings pour la transmission sont regroupés dans trois profils : S-Net, DP ou utilisateur. Avec le profil utilisateur le bouton *Bus Parameter* permet de définir ses propres *timings*. Le Profil doit être identique sur toutes les stations du réseau. Le profil S-Net est nécessaire lors de l'usage de RIO PCD3.T76x sur les réseaux.

12.4.4 Charger les paramètres du *Device Configurator* sur l'automate



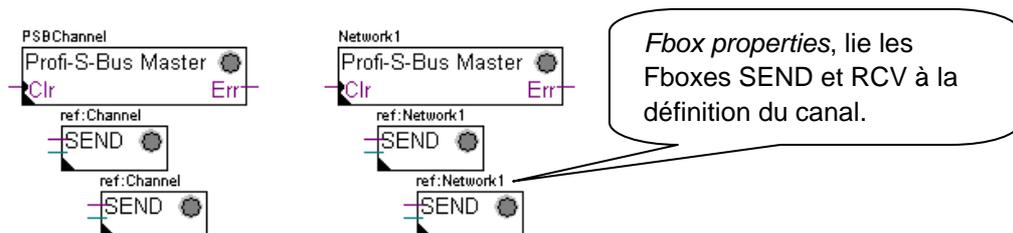
Avec les nouveaux systèmes PCD, les paramètres du *Device Configurator* peuvent être chargés sur l'automate au travers de l'interface USB. Il faut juste vérifier que les *Online Settings* soient définis avec un canal *S-Bus USB* + PGU



Charger les paramètres dans l'automate à l'aide du bouton *Download Configuration* présent dans la fenêtre *Device Configurator*.

12.5 Programme Fupla

12.5.1 Assigner le canal à l'aide d'une Fbox SASI



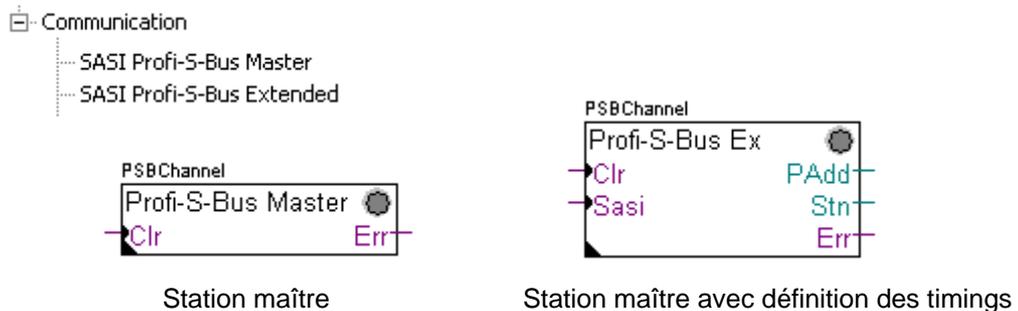
L'assignation du canal est assurée à l'aide d'une FBox SASI placée au début du fichier Fupla. Chaque réseau de communication dispose de sa propre Fbox SASI, car les paramètres d'assignation sont différents d'un réseau à l'autre, de même pour une station esclave ou maître.

Si l'automate 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 Fboxes 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 le *Device Configurator*.

Mais le numéro du canal de communication est toujours défini à l'aide des paramètres d'ajustage de la FBox SASI. Le numéro du canal dépend du Hardware de l'automate et du hardware de communication utilisé : slot B1, B2, interface série PCD7.F, ...

12.5.2 Assigner le canal maître



L'assignation du canal maître est assurée en complétant les paramètres du *Device Configurator* avec l'une des Fboxes 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 le *Device Configurator*.

Paramètres d'ajustage:

Channel

Définit le canal correspondant à l'interface sérielle raccordée au réseau. Dépend de l'automate 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)

12.5.3 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 le *Device Configurator*.

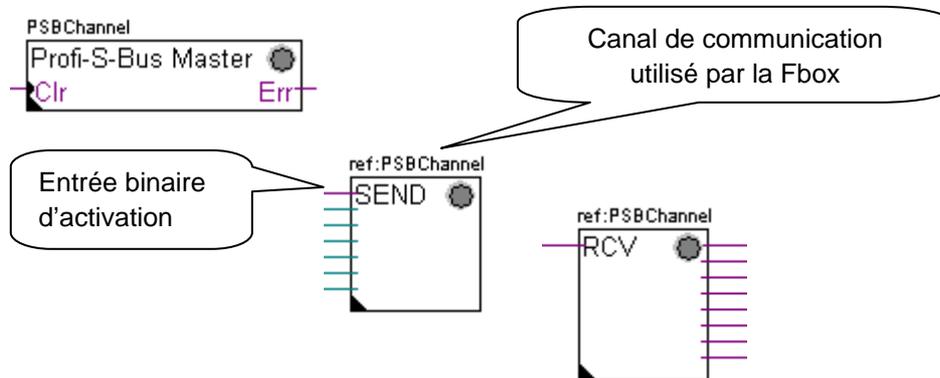
12.5.4 Principe d'échanges des données sur un réseau multi-maîtres.

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.

12.5.5 Echange de données entre les stations maîtres et esclaves



L'utilisateur contrôle l'échange des données entre les stations à l'aide de diverses Fboxes *Fupla* placées dans les pages *Fupla* et disponibles dans le *Fbox Selector*. Nous y trouvons des Fboxes 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*

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 de donnée sont répétés aussi rapidement que possible. Si une courte impulsion est appliquée sur l'entrée, l'échange des données est effectué au moins une fois mais il est toujours possible de le forcer avec le bouton *Execute* ou au démarrage à froid de l'automate avec l'option *Initialization* des paramètres d'ajustage.

Les données de la station maître présente 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 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 Fboxes 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 d'ajustage.

Profi-S-Bus Address

Défini le numéro de station Profi-S-Bus Bus esclave

Source, destination station (S-Bus)

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*

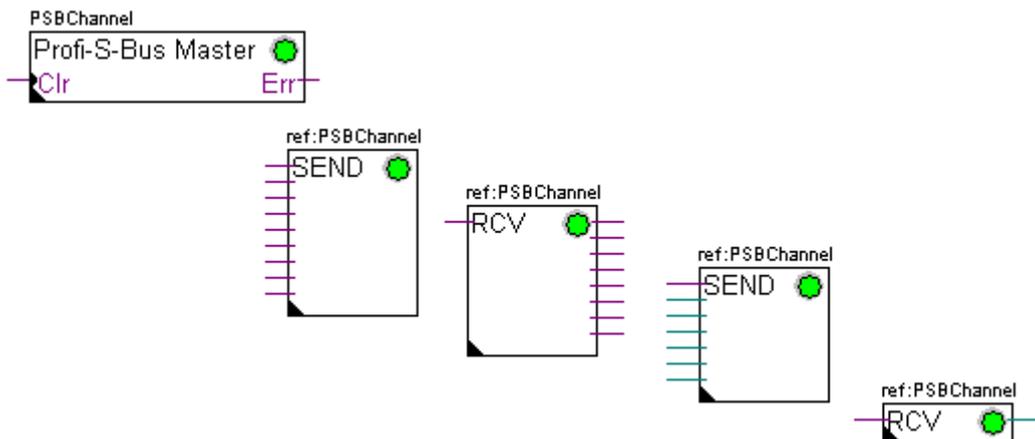
12.5.6 Diagnostiques



Si le programme est en ligne, une LED de couleur verte ou rouge est affiché en haut à droite des FBoxes *SASI*, *SEND* et *RCV*. Vert indique que la transmission des données de est OK, rouge indique une erreur.

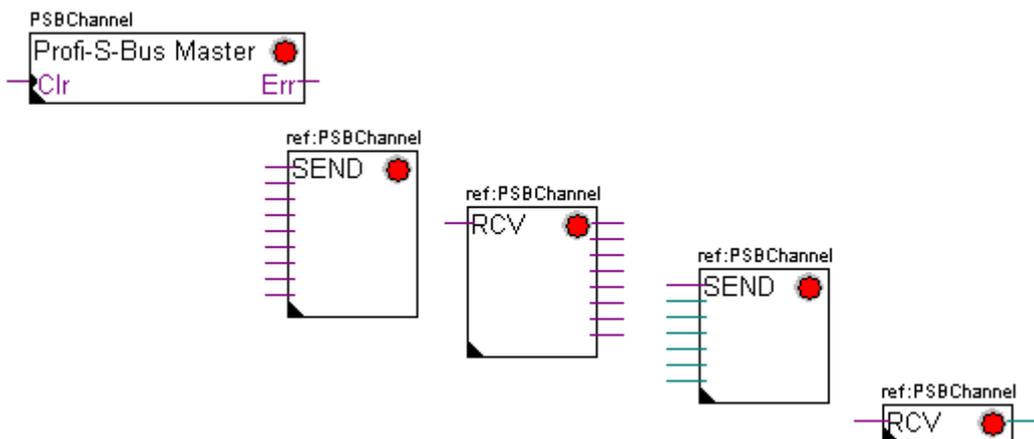
Fonctionnement correct

Toutes les FBoxes 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.

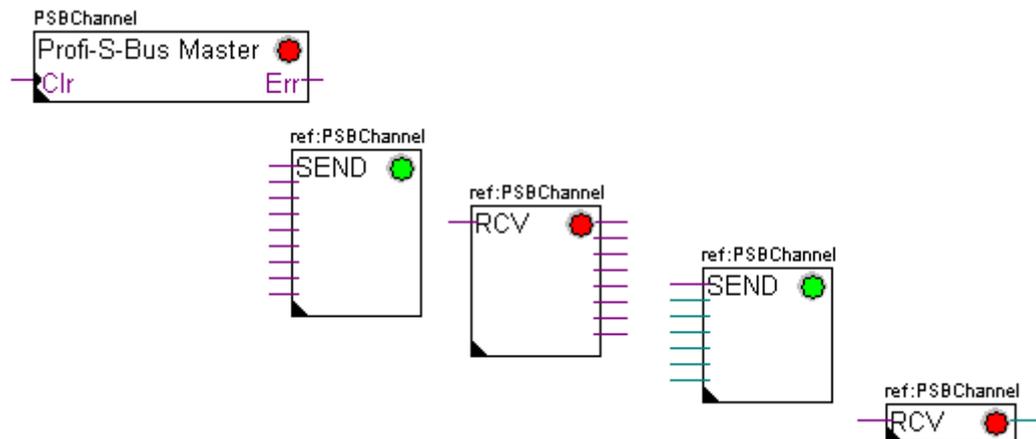


Actions correctives possibles sur la station maître ou esclave :

- Vérifier les paramètres du *Device Configurator*
- Vérifier le chargement des paramètres du *Device Configurator* dans l'automate.
- 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 le *Device Configurator* et la fonction *SAS/* sont identiques (même numéro de canal)
- Vérifier si l'automate 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 Fboxes vertes échangent correctement les données



Actions correctives possibles sur la station maître

Vérifier les paramètres de la fenêtre d'ajustage des FBoxes *SEND* et *RCV* dont le diagnostic est rouge. Vérifier que l'adresse de l'esclave soit bien présente dans le réseau.

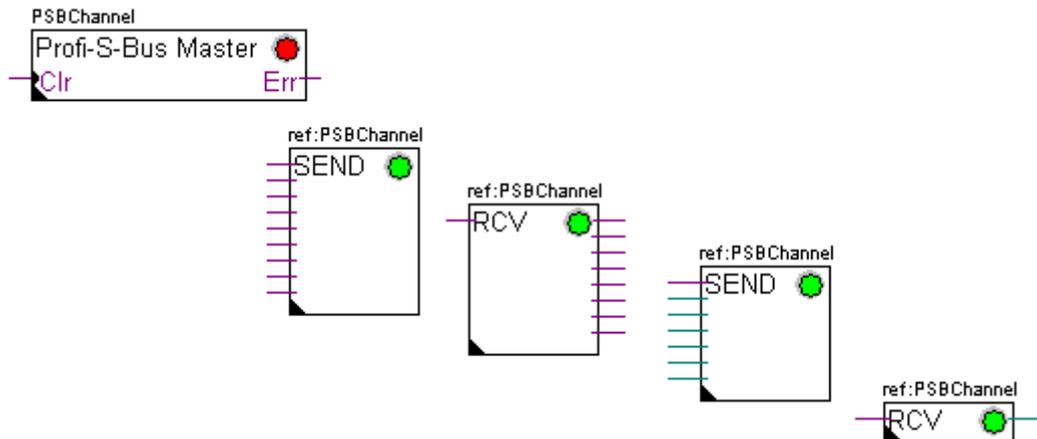
Actions correctives possibles sur la station esclave

Pour chaque Fbox *SEND* et *RCV* en erreur, relever le numéro de station esclave et vérifier les stations concernées.

- Vérifier si les paramètres du *Device Configurator* sont correctement définis
- Vérifier si l'automate 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

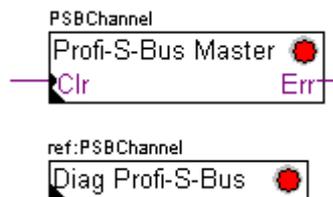
Seule la Fbox SASI est rouge

Ouvrir le fenêtre d'ajustage de la Fbox SASI et quitter la dernière erreur avec le bouton *Clear*.



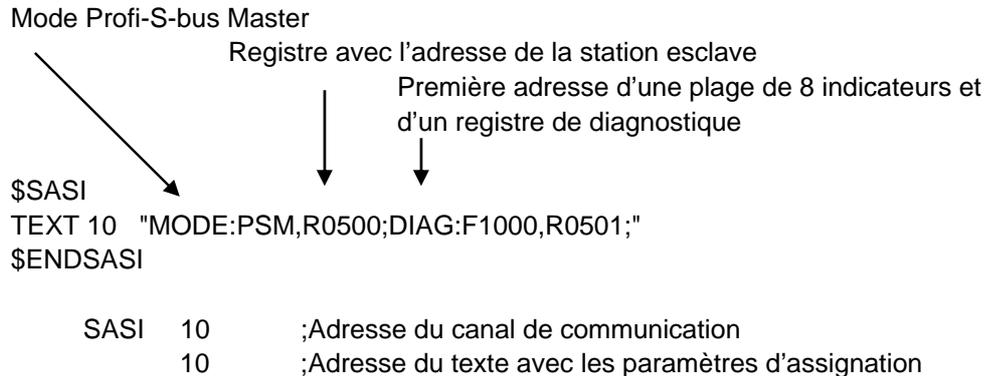
Fbox de diagnostic

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



12.6 Programme IL

12.6.1 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 l'automate 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 paramètres du *Device Configurator* .

12.6.2 Assigner le canal esclave

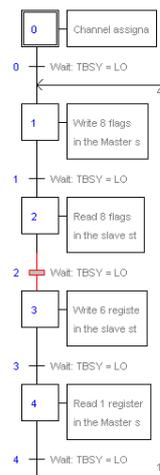
Aucune instruction SASI n'est nécessaire pour la station esclave d'un réseau Profi-S-Bus. Toutes les définitions nécessaires sont déjà présentes dans le *Device Configurator* .

12.6.3 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.

12.6.4 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 de l'automate.

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.

12.6.5 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 paramètres du *Device Configurator*
- Vérifier le chargement des paramètres du *Device Configurator* dans l'automate
- 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 le *Device Configurator* et l'instruction *SASI* sont identiques (même numéro de canal)
- Vérifier si l'automate 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 quelles 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 paramètres du *Device Configurator* sont correctement définis
- Vérifier si l'automate 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

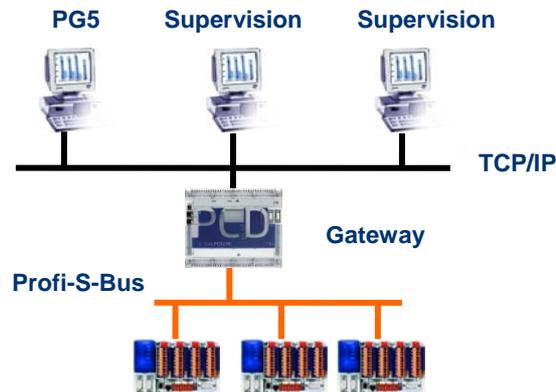
Registre de diagnostic

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.

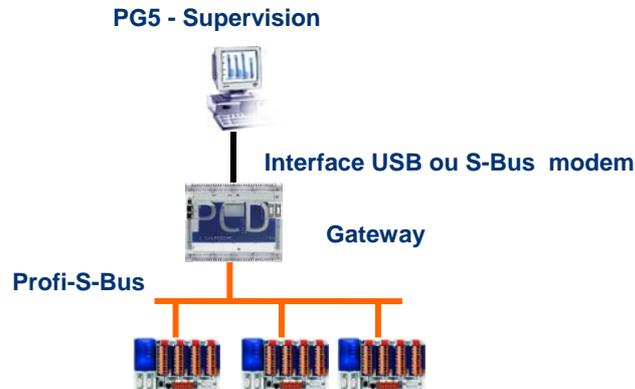
12.7 Fonction Gateway

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.

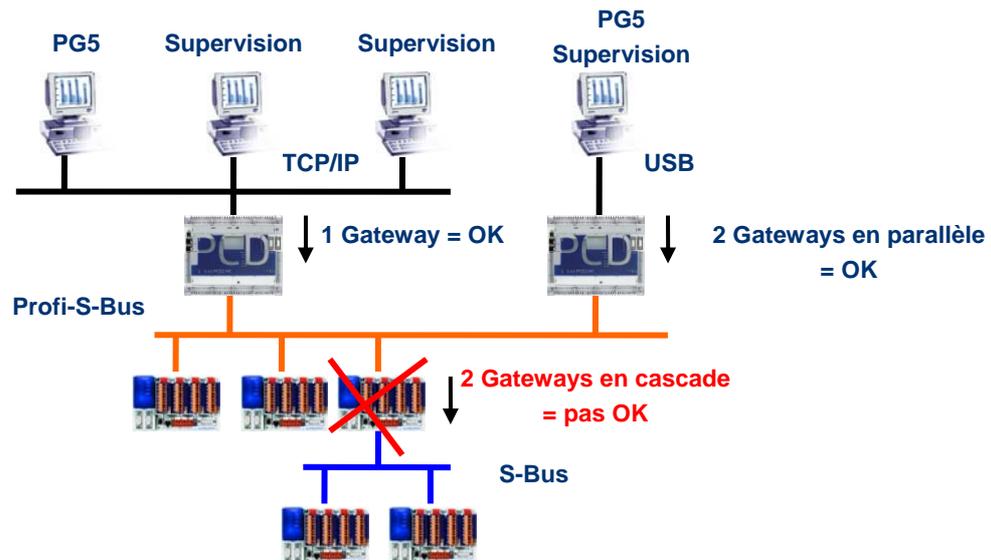
12.7.1 Application



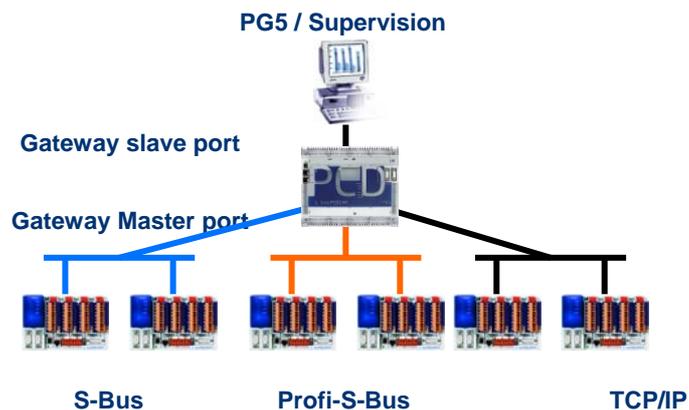
La fonction *Gateway* permet d'effectuer un pont entre deux réseaux. Par exemple lier un réseau Ethernet avec le réseau Profi-S-Bus. Ainsi les systèmes PCD échangent des données sur un bus de terrain propre au domaine de l'automatisation et séparé du réseau informatique d'entreprise. Mais les ordinateurs équipés d'un logiciel PG5 ou d'un système de supervision Visi+ peuvent échanger des données avec les différents automates.



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.

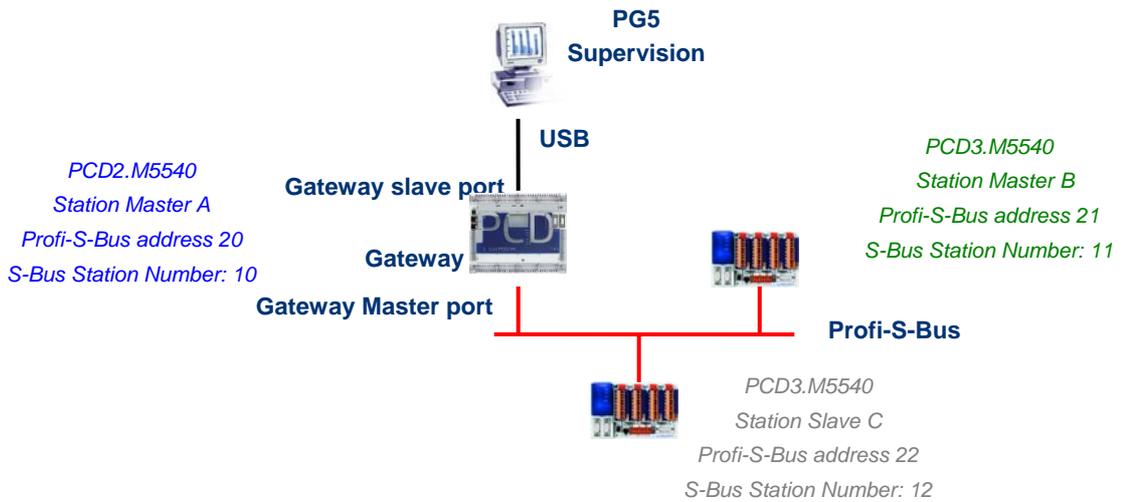
12.7.2 Configuration de la fonction *Gateway* PGU

La fonction *Gateway* est très facile à mettre en service, elle ne nécessite aucun programme, mais seulement quelques paramètres dans le *Device Configurator*.

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

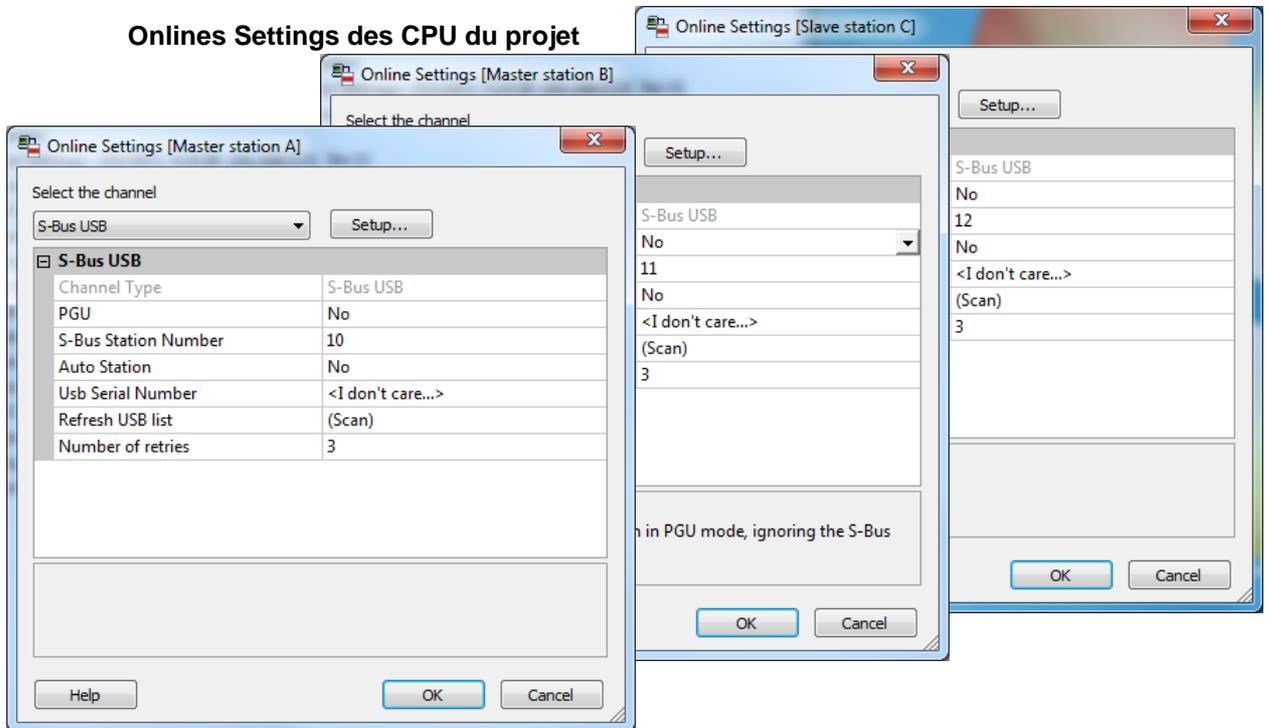


Onboard Communication, propriétés de la station Master A

Profi-S-Bus Master Gateway	
Use Profi-S-Bus For Gateway	Yes
First S-Bus Station Profi-S-Bus	0
Last S-Bus Station Profi-S-Bus	253
Response Timeout	0

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 automates, il faut encore ajuster les *Onlines Settings* de chaque device 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 devices, *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 device actif et de le tester en restant toujours lié avec le câble USB sur la station *Master A*

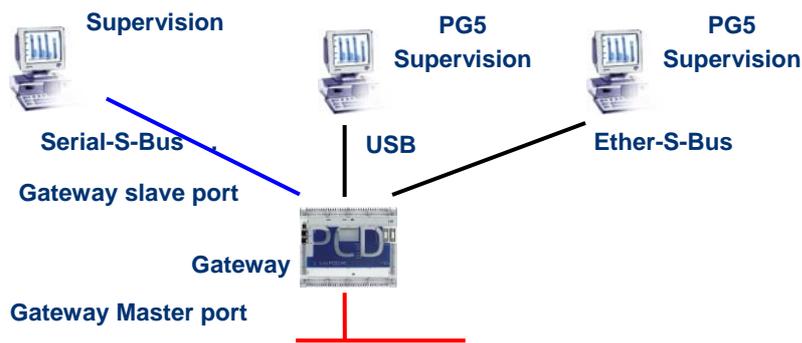
Master station B - PCD3.M5540 - Station 11

Pour communiquer avec une autre station du réseau, activer le device 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)

12.7.3 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.

Paramètres du Device Configurator

En général, l'automate ne supporte qu'un seul canal esclave PGU. Mais les nouveaux automates peuvent en supporter plusieurs sur le même automate. La configuration du second Gateway Slave PGU est complètement supportée par le *Device Configurator*.

Exemple: ajouter un second Gateway Ether-S-Bus, Profi-S-Bus

Onboard Communications		TCP/IP	
Location	Type	Channel Number	
Onboard	RS-232/RS-485	TCP/IP Enabled	Yes
Onboard	RS-485/S-Net	Ethernet RIO Network	None
Onboard	USB	IP Address	192.168.12.130
Onboard	Ethernet	Subnet Mask	255.255.255.0
		Default Gateway	0.0.0.0
		+ Access Control List	Hide

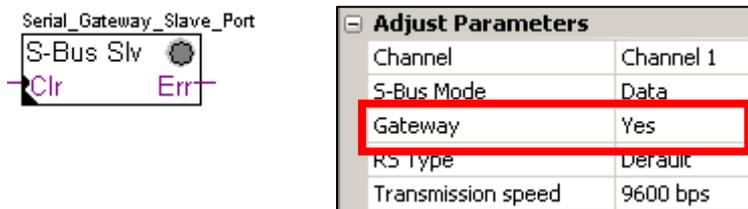
Le second *Gateway Slave port PGU* est ajouté en configurant le *Device Configurator* avec le nœud et l'adresse TCP/IP.

Programme Fupla ou IL

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



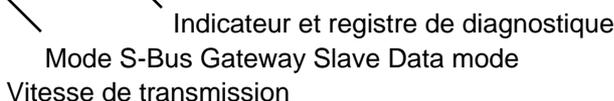
Le paramètre d'ajustage *Gateway* doit alors être défini avec l'option Yes. Selon le type de canal, les autres paramètres d'ajustage doivent être aussi correctement définis.

Exemple IL : ajouter un troisième Serial-S-Bus, Profi-S-Bus

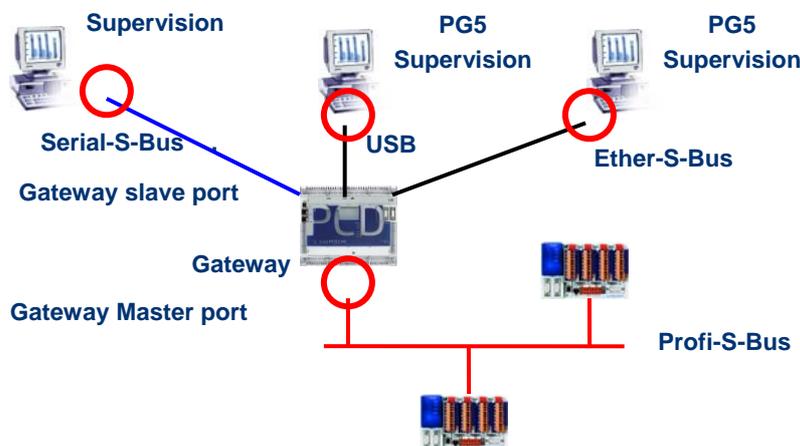
Utiliser le texte suivant pour assigner le canal :

```

$SASI
TEXT 11 "UART:9600;MODE:GS2;DIAG:F1110,R0501;"
$ENDSASI
    
```

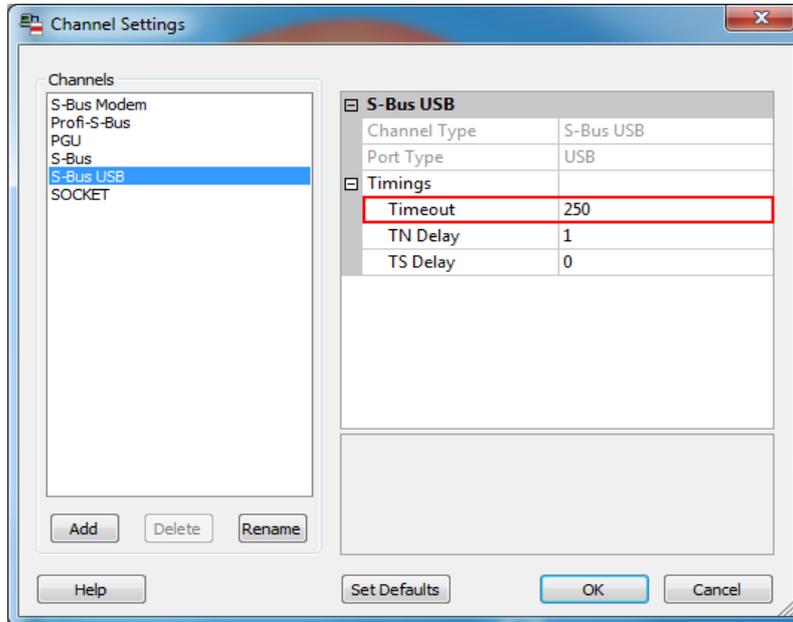


12.7.4 Timings de communication

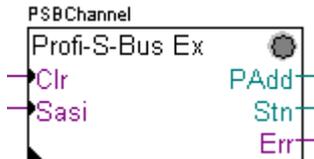


En général les *timings* de communication sont toujours définis avec les valeurs par défaut et cela fonctionne correctement. Mais l'usage de la fonction *Gateway* augmente les temps de réactions nécessaire à l'échange de données. Il est alors parfois nécessaire d'ajuster le timeout des stations maîtres 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*



Adjust Parameters	
Channel	Channel 10
Gateway	No
Timeout	0

12.8 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

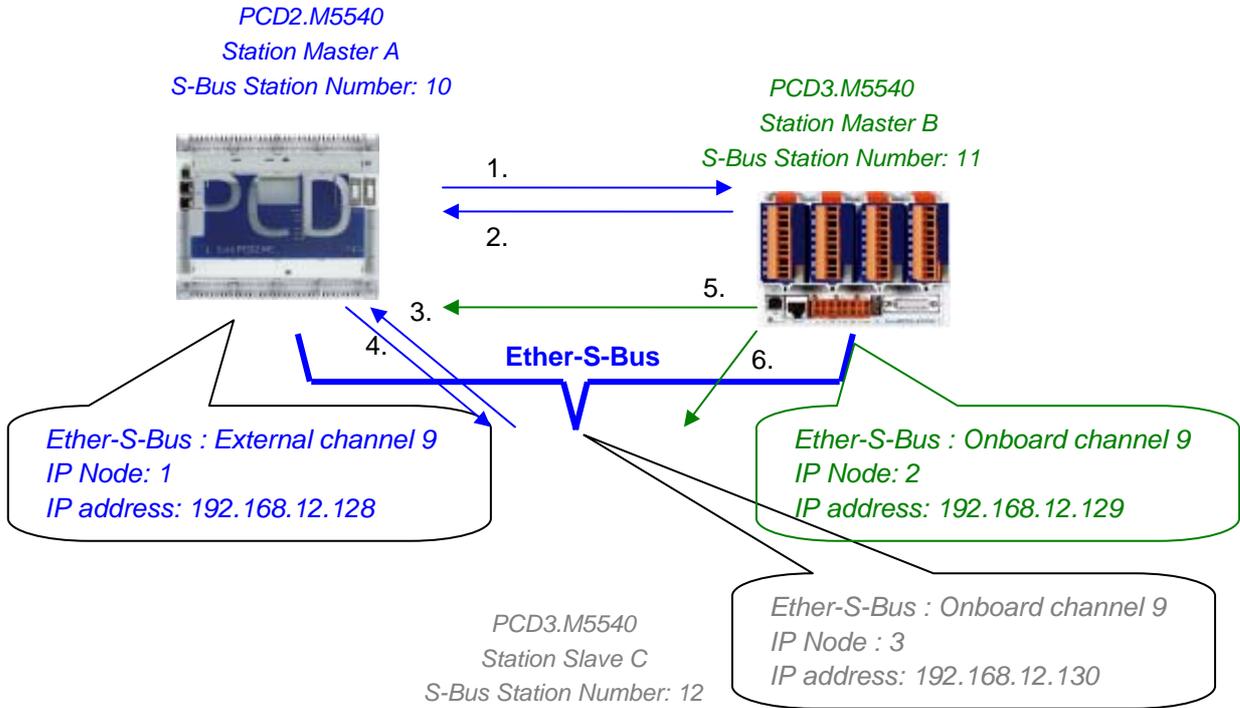
Table des matières

TABLE DES MATIERES	1
13 ETHER-S-BUS	2
13.1 Exemple de réseau Ether-S-Bus.	2
13.2 Exemples d'échanges de données Ether-S-Bus.	2
13.3 Le Projet PG5	3
13.4 Paramètres du Device Configurator	3
13.4.1 Définir le type d'automate	3
13.4.2 Définir le numéro de station S-Bus sur le réseau.....	3
13.4.3 Définir le canal de communication Ether-S-Bus	4
13.4.4 Charger les paramètres du <i>Device Configurator</i> sur l'automate.....	5
13.5 Programme Fupla.....	5
13.5.1 Assigner le canal à l'aide d'une Fbox SASI	5
13.5.2 Assigner le canal maître	6
13.5.3 Assigner le canal esclave	6
13.5.4 Principe d'échanges des données sur un réseau multi-maîtres.	6
13.5.5 Echange de données entre les stations maîtres et esclaves	7
13.5.6 Diagnostiques	8
13.6 Programme IL	11
13.6.1 Assigner le canal maître à l'aide d'une instruction SASI	11
13.6.2 Assigner le canal esclave	11
13.6.3 Principe d'échanges des données sur un réseau multi-maîtres.	11
13.6.4 Echange de données entre les stations maîtres et esclaves	12
13.6.5 Diagnostiques	13
13.7 Fonction Gateway	15
13.7.1 Application	15
13.7.2 Configuration de la fonction Gateway PGU	16
13.7.3 Configuration d'un Gateway Slave port esclave supplémentaire	18
13.7.4 Timings de communication	20
13.8 Autres références	21

13 Ether-S-Bus

Cet exemple présente comment échanger quelques données comme des registres et indicateurs entre les automates raccordés sur le réseau Ether-S-Bus.

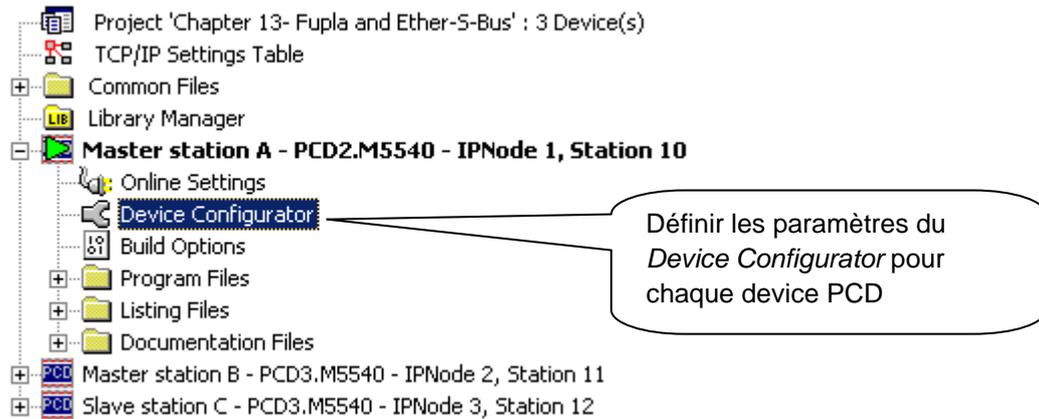
13.1 Exemple de réseau Ether-S-Bus.



13.2 Exemples d'échanges de données Ether-S-Bus.

	Master with data exchanges	Data on the network	Passive master or slave
	Master station A		Master station 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
			Slave station C
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
	Master station B		Master station A
5	Temperature1 .. 4 Dynamic registers	Write the temperature measures to the slave C	Master_B.Temperature1 .. 4 R 100 .. 104
			Slave station C
6	Temperature1 .. 4 Dynamic registers	Write the temperature measures to the master A	Master_B.Temperature1 .. 4 R 100 .. 104

13.3 Le Projet PG5



Le *Saia PG5 Project Manager* donne une vue de l'ensemble des automates d'un projet d'application ainsi que des paramètres pour les réseaux de communication. Nous commencerons par ajouter un device dans le projet pour chaque station disponible sur le réseau.

13.4 Paramètres du Device Configurator

Les paramètres du *Device Configurator* sont identiques pour station maître et esclaves sont semblables.

13.4.1 Définir le type d'automate

Device	
Type	Description
PCD2.M5540	CPU with 1M Bytes RAM, 8 I/O slots, 2 communication slot.

Device Type

Définir le type d'automate

13.4.2 Définir le numéro de station S-Bus sur le réseau

S-Bus	
S-Bus Support	Yes
Station Number	10

Device properties:

Station Number

Le numéro de station S-Bus est commun à tous les canaux de communication de l'automate.

13.4.3 Définir le canal de communication Ether-S-Bus

Onboard Communications	
Location	Type
Onboard	RS-232/RS-485
Onboard	RS-485/S-Net
Onboard	USB
Onboard	Ethernet

TCP/IP	
TCP/IP Enabled	Yes
IP Node	1
IP Address	192.168.12.128
Subnet Mask	255.255.255.0
Default Router	0.0.0.0
PGU port	Yes
Slave	Yes
Network groups	(Default)

Onboard Communication, properties:

IP Node

Numéro du nœud TCP/IP. Le nœud est utilisé dans les FBoxes SEND et RCV pour définir une station esclave avec laquelle échanger des données.

IP Address

Numéro de station Ether-S-Bus lié au canal

PGU Port

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.

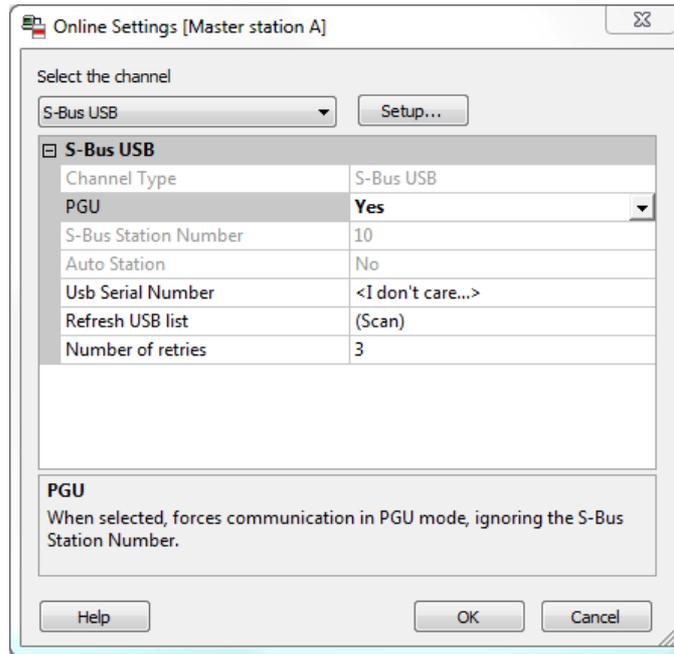
Esclave + PGU

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.

13.4.4 Charger les paramètres du *Device Configurator* sur l'automate



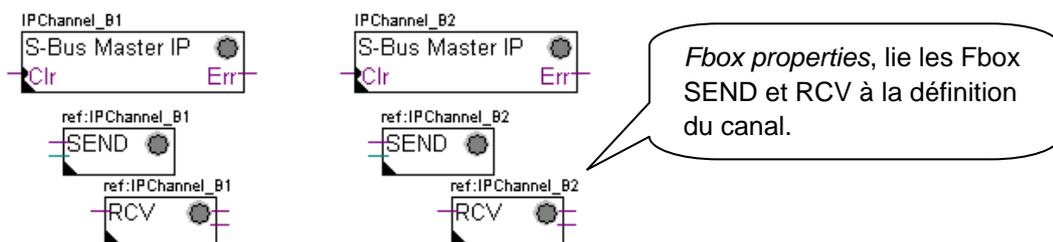
Avec les nouveaux systèmes PCD, les paramètres du *Device Configurator* peuvent être chargés sur l'automate au travers de l'interface USB. Il faut juste vérifier que les *Online Settings* soient définis avec un canal *S-Bus USB* + *PGU*.



Charger les paramètres dans l'automate à l'aide du bouton *Download* présent dans la fenêtre *Device Configurator*.

13.5 Programme Fupla

13.5.1 Assigner le canal à l'aide d'une Fbox SASI



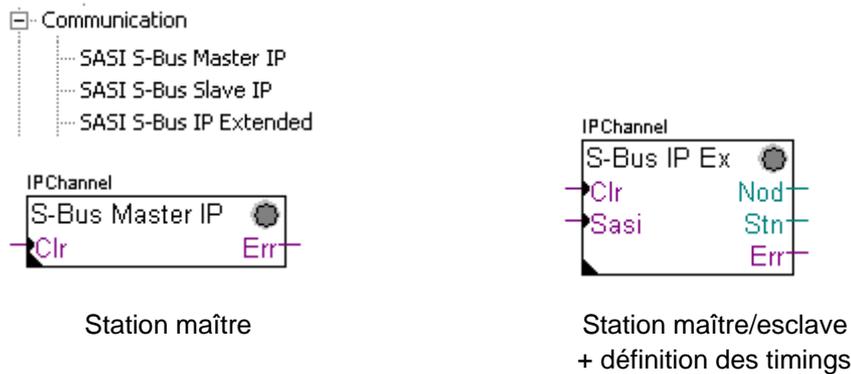
L'assignation du canal est assurée à l'aide d'une FBox SASI placée au début du fichier Fupla. Chaque réseau de communication dispose de sa propre Fbox SASI, car les paramètres d'assignation sont différents d'un réseau à l'autre, de même pour une station esclave ou maître.

Si l'automate 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 Fboxes 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 *Device Configurator*.

Mais le numéro du canal de communication est toujours défini à l'aide des paramètres 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, ...

13.5.2 Assigner le canal maître



L'assignation du canal maître est assurée en complétant les paramètres du *Device Configurator* esclave avec l'une des FBoxes ci-dessus

Paramètres d'ajustage:

Channel

Définit le numéro du canal raccordé au réseau. Dépend de l'automate 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)

13.5.3 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 *Device Configurator*.

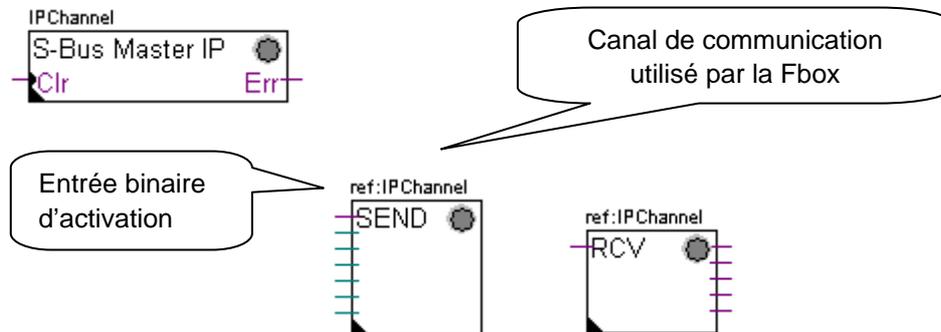
13.5.4 Principe d'échanges des données sur un réseau multi-maîtres.

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.

13.5.5 Echange de données entre les stations maîtres et esclaves



L'utilisateur contrôle l'échange des données entre les stations à l'aide de diverses Fboxes Fupla placées dans les pages Fupla et disponibles dans le Fbox *Selector*. Nous y trouvons des FBoxes 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*.

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 de données sont répétés aussi rapidement que possible. Si une courte impulsion est appliquée sur l'entrée, l'échange des données est effectué au moins une fois mais il est toujours possible de le forcer avec le bouton *Execute* ou au démarrage à froid de l'automate avec l'option *Initialization* des paramètres 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 d'ajustage : adresse de la station esclave, élément source et adresse de base.

Seul les stations maîtres sont programmées avec des FBoxes *SEND* et *RCV* ! Les stations esclaves doivent seulement être assignées avec le canal de communication.

Selon les Fboxes 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 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*

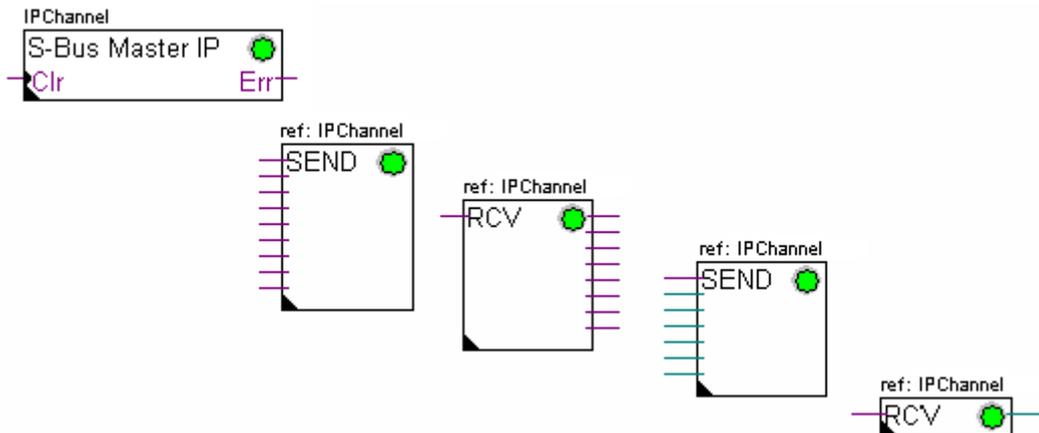
13.5.6 Diagnostiques



Si le programme est en ligne, une LED de couleur verte ou rouge est affiché en haut à droite des FBoxes *SASI*, *SEND* et *RCV*. Vert indique que la transmission des données de est OK, rouge indique une erreur.

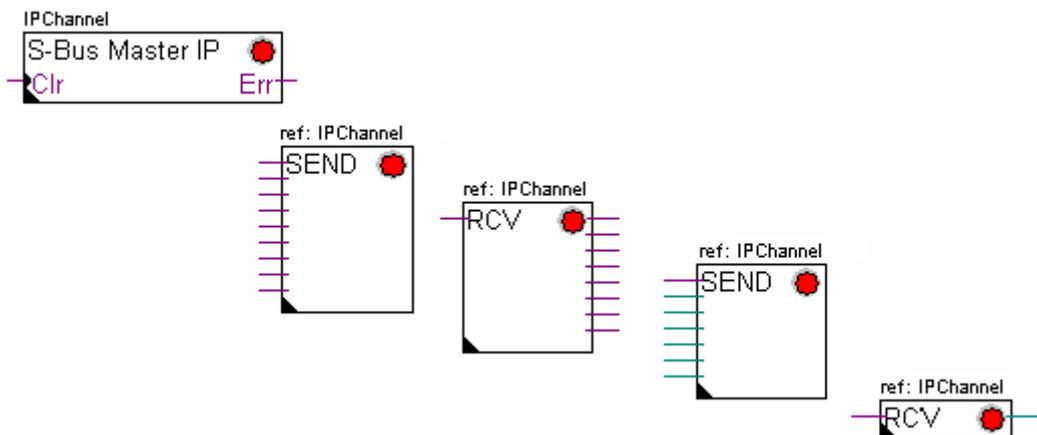
Fonctionnement correct

Toutes les FBoxes 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.

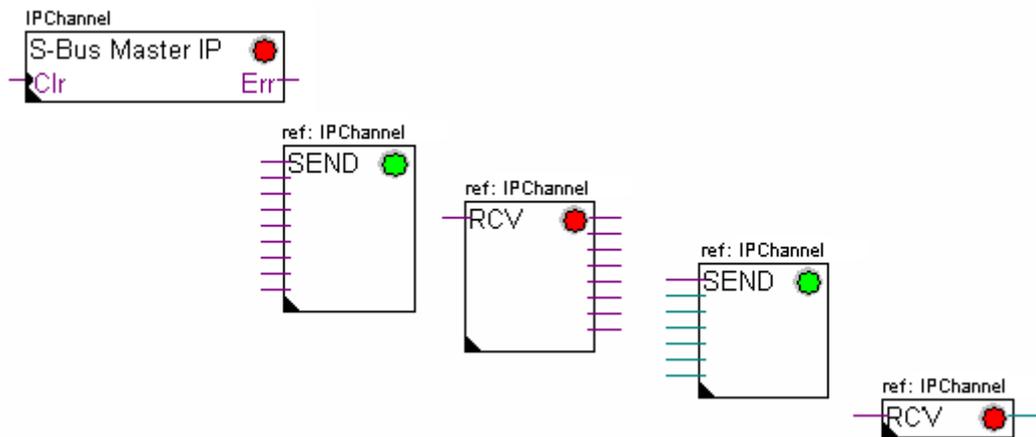


Actions correctives possibles sur la station maître ou esclave :

- Vérifier les paramètres du *Device Configurator*
- Vérifier le chargement des paramètres du *Device Configurator* dans l'automate
- Vérifier si le canal de communication défini avec le *Device Configurator* et la fonction *SAS/* sont identiques (même numéro de canal)
- Vérifier si l'automate 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 *SAS/* ainsi que quelques FBoxes *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 FBoxes *SEND* et *RCV* dont le diagnostic est rouge. Vérifier que l'adresse de l'esclave soit bien présente dans le réseau.

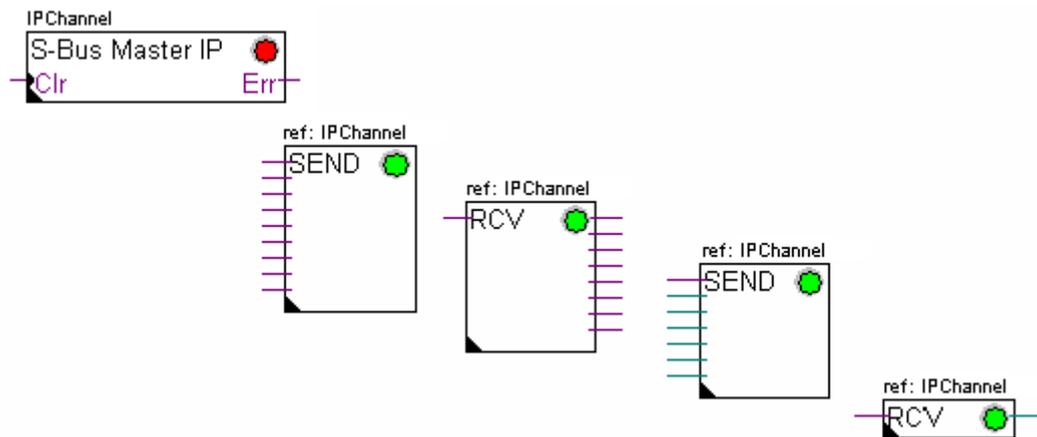
Actions correctives possibles sur la station esclave

Pour chaque Fbox *SEND* et *RCV* en erreur, relever le numéro de station esclave et vérifier les stations concernées.

- Vérifier si les paramètres du *Device Configurator* sont correctement définis
- Vérifier si l'automate 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*.



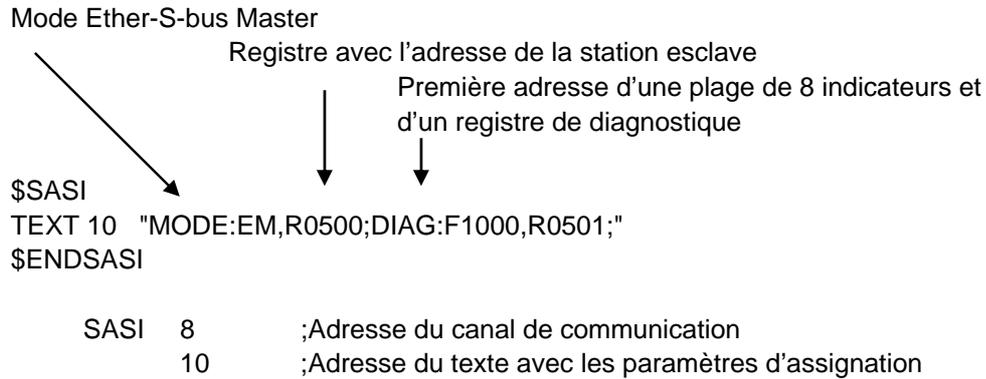
Fbox de diagnostic

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*



13.6 Programme IL

13.6.1 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 l'automate 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 les paramètres du *Device Configurator* .

13.6.2 Assigner le canal esclave

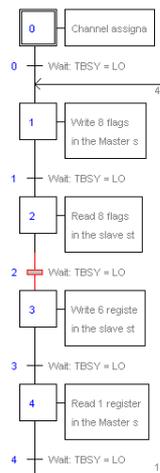
Aucune instruction SASI n'est nécessaire pour la station esclave d'un réseau Ether-S-Bus. Toutes les définitions nécessaires sont déjà présentes dans le *Device Configurator* .

13.6.3 Principe d'échanges des données sur un réseau multi-mâtres.

Le réseau de communication multi-mâtres 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-mâtres, 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.

13.6.4 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 de l'automate.

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.

13.6.5 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 paramètres du *Device Configurator*
- Vérifier le chargement des paramètres du *Device Configurator* dans l'automate
- 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 le *Device Configurator* et l'instruction *SASI* sont identiques (même numéro de canal)
- Vérifier si l'automate 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 paramètres du *Device Configurator* sont correctement définis
- Vérifier si l'automate 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

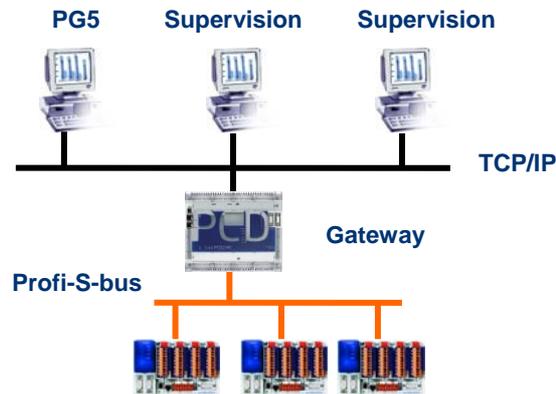
Registre de diagnostic

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.

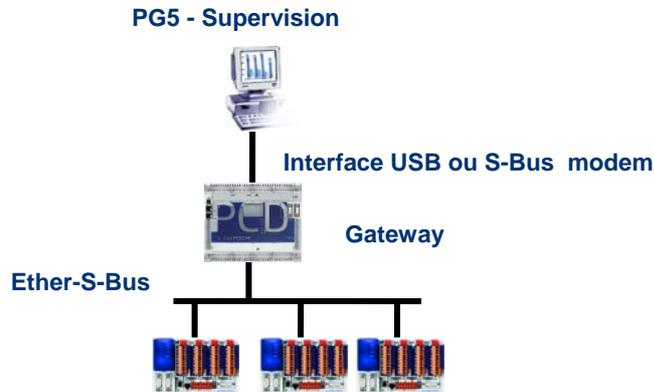
13.7 Fonction Gateway

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.

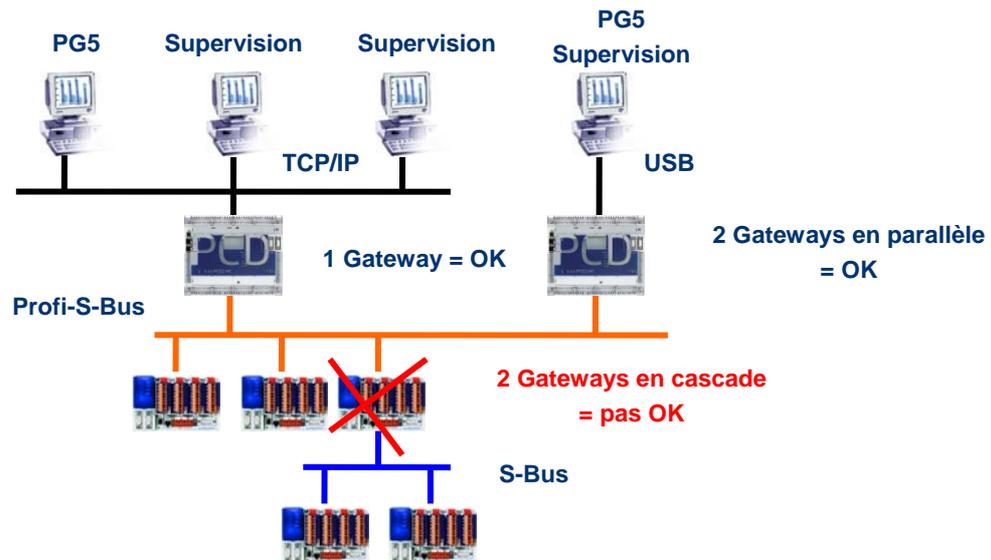
13.7.1 Application



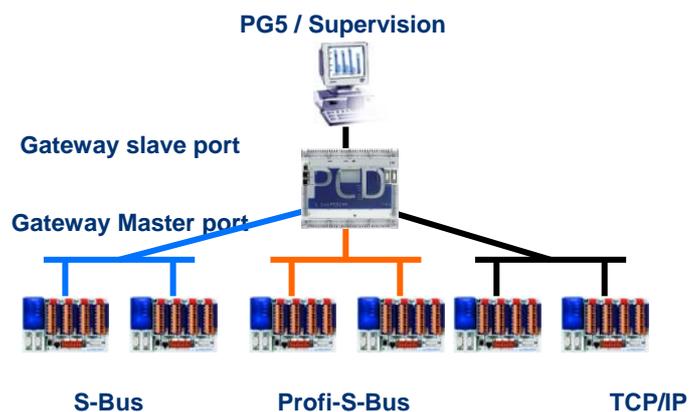
La fonction *Gateway* permet d'effectuer un pont entre deux réseaux. Par exemple lier un réseau Ethernet avec le réseau Profi-S-Bus. Ainsi les systèmes PCD échangent des données sur un bus de terrain propre au domaine de l'automatisation et séparé du réseau informatique d'entreprise. Mais les ordinateurs équipés d'un logiciel PG5 ou d'un système de supervision Visi+ peuvent échanger des données avec les différents automates.



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.

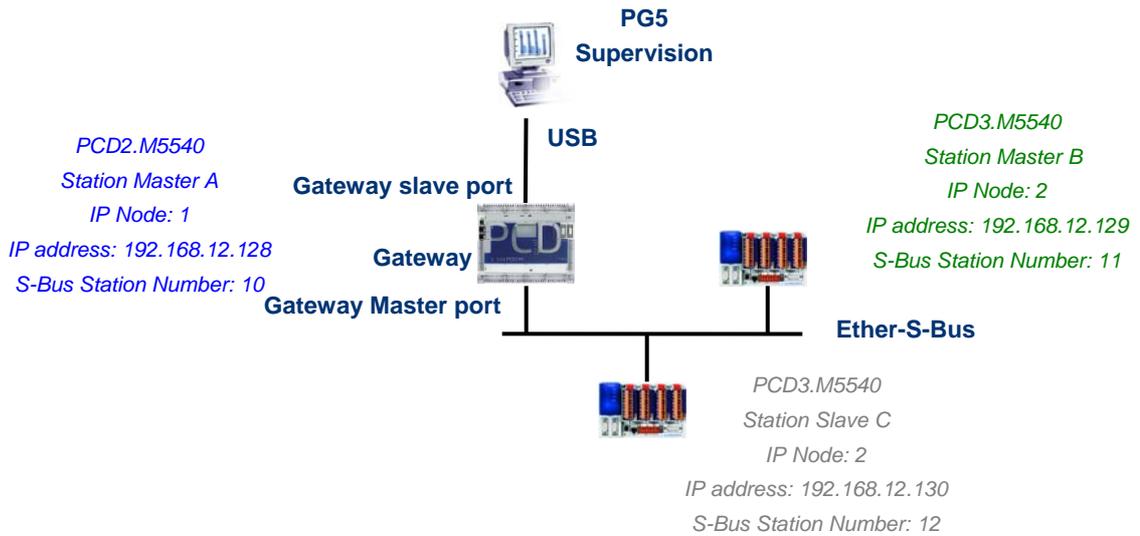
13.7.2 Configuration de la fonction Gateway PGU

La fonction *Gateway* est très facile à mettre en service, elle ne nécessite aucun programme, mais seulement quelques paramètres dans le *Device Configurator.de* de l'automate.

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

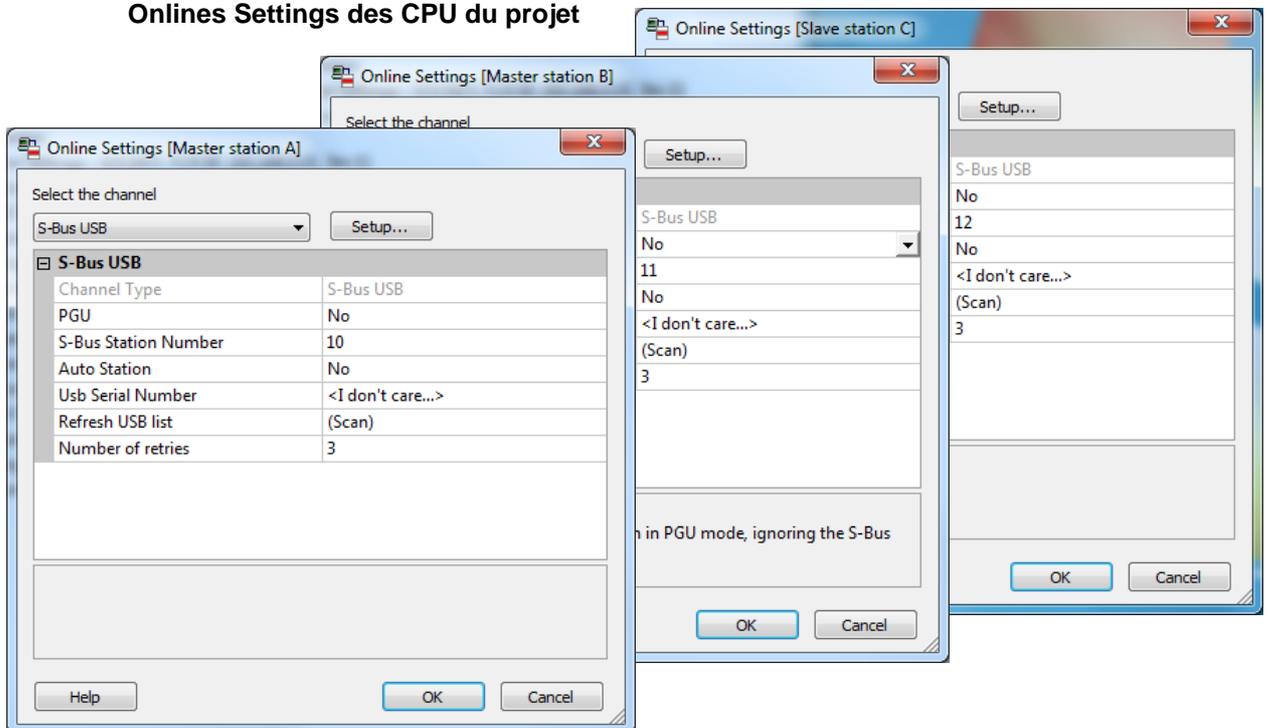


Onboard Communication, propriétés de la station Master A

TCP/IP S-Bus Master Gateway	
Use TCP/IP For Gateway	Yes
First S-Bus Station	0
Last S-Bus Station	253
Response Timeout	0

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 automates, il faut encore ajuster les *Onlines Settings* de chaque device 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 devices, *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 device actif et de le tester en restant toujours lié avec le câble USB sur la station *Master A*

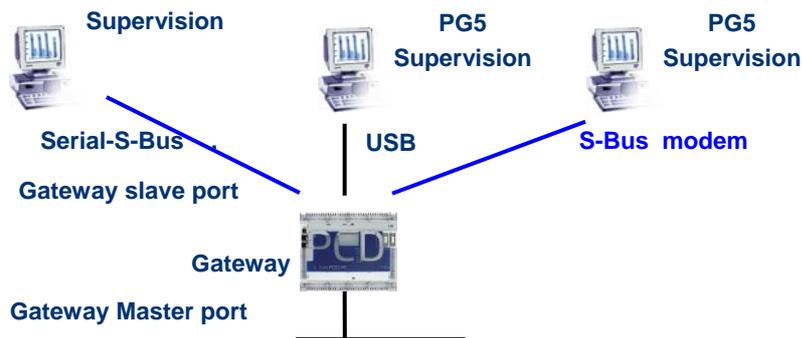
 Master station B - PCD3.M5540 - IPNode 2, Station 11

Pour communiquer avec une autre station du réseau, activer le device 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)

13.7.3 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.

Paramètres du *Device Configurator*

En général, l'automate ne supporte qu'un seul canal esclave PGU. Mais les nouveaux automates peuvent en supporter plusieurs sur le même automate. La configuration du second Gateway Slave PGU est complètement supportée par le *Device Configurator*.

Exemple: ajouter un second Gateway S-Bus modem, Ether-S-Bus

Onboard Communications	
Type	Description
RS-485/S-Net	RS-485 port for P...
USB	Universal Serial E...
RS-232/PGU	RS-232, PGU or c...
RS-485	RS-485 port for ge...
Ethernet	Ethernet port. IP S...

Public Line S-Bus Modem	
Port Number Modem	0
Use Serial S-Bus For Modem	Yes
Full Protocol (PGU) on Modem	Yes
Modem Name	T813/T814
Modem Init	AT&F1%CO&D
Modem Reset	ATZ\r

S-Bus Mode And Timing	
S-Bus Mode	Data Mode
Baud Rate	19200 Baud
Response Timeout [ms]	0
Training Sequence Delay [ms]	0
Turnaround Delay [ms]	0

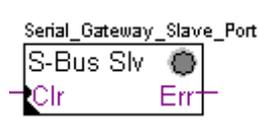
Le second *Gateway Slave port PGU* est ajouté en configurant le *Device Configurator* avec les paramètres pour le modem.

Programme Fupla ou IL

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 de l'automate ne sont supportées: registres, indicateurs, ...

Exemple Fupla : ajouter un troisième Serial-S-Bus, Ether-S-Bus



Adjust Parameters	
Channel	Channel 1
S-Bus Mode	Data
Gateway	Yes
RS Type	Default
Transmission speed	9600 bps

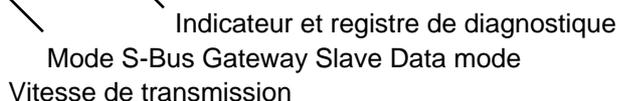
Le paramètre d'ajustage *Gateway* doit alors être défini avec l'option Yes. Selon le type de canal, les autres paramètres d'ajustage doivent être aussi correctement définis.

Exemple IL : ajouter un troisième Serial-S-Bus, Ether-S-Bus

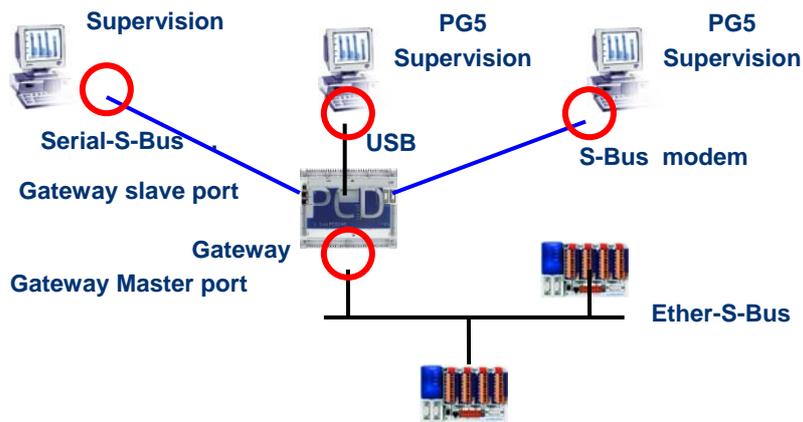
Utiliser le texte suivant pour assigner le canal :

```

$SASI
TEXT 11 "UART:9600;MODE:GS2;DIAG:F1110,R0501;"
$ENDSASI
    
```

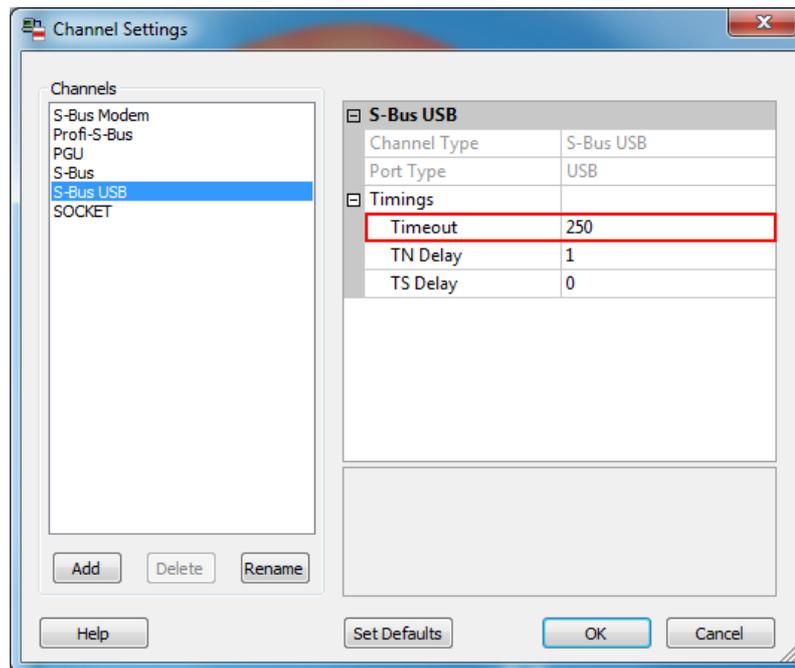


13.7.4 Timings de communication

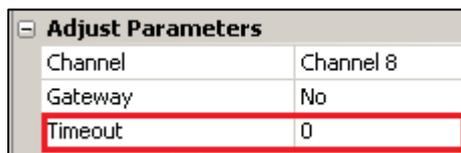


En général les *timings* de communication sont toujours définis avec les valeurs par défaut et cela fonctionne correctement. Mais l'usage de la fonction *Gateway* augmente les temps de réactions nécessaire à l'échange de données. Il est alors parfois nécessaire d'ajuster le timeout des stations maîtres 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*



13.8 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

TABLE DES MATIERES 1

15 PROFI-S-IO 2

15.1 Exemple de réseau Profi-S-IO2

15.2 Fonctionnement général.....2

15.3 Le Projet PG53

15.4 Définir les stations sur le réseau3

15.5 Configurer la station maître4

15.6 Configurer les stations esclaves4

 15.6.1 Configurer les modules d’entrées, sorties.....4

 15.6.2 Configurer les symboles des données déportées.....5

 15.6.3 Configurer les paramètres d’ajustages5

15.7 Configurer les paramètres du réseau.....6

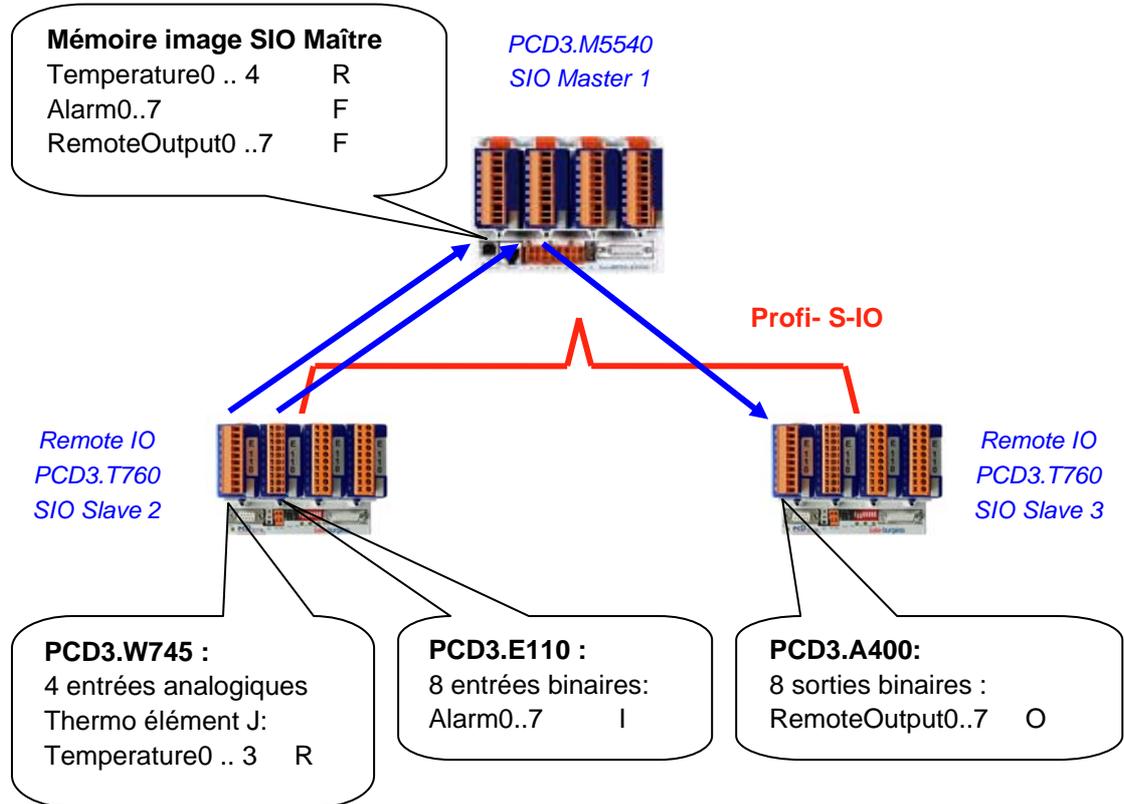
15.8 Exploiter les symboles réseau dans un programme Fupla ou IL.6

15.9 Autres références7

15 Profi-S-IO

Cet exemple présente comment utiliser les entrées, sorties binaires ou analogiques déportées sur des borniers RIO de type PCD3.T7xx.

15.1 Exemple de réseau Profi-S-IO.



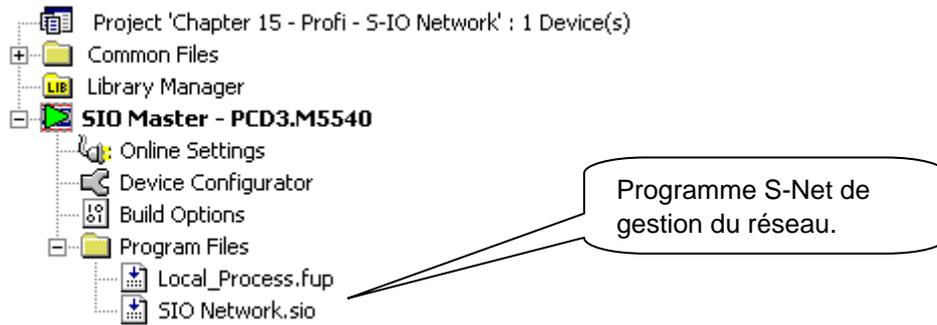
15.2 Fonctionnement général

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 du *Device Configurator* 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.

15.3 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).

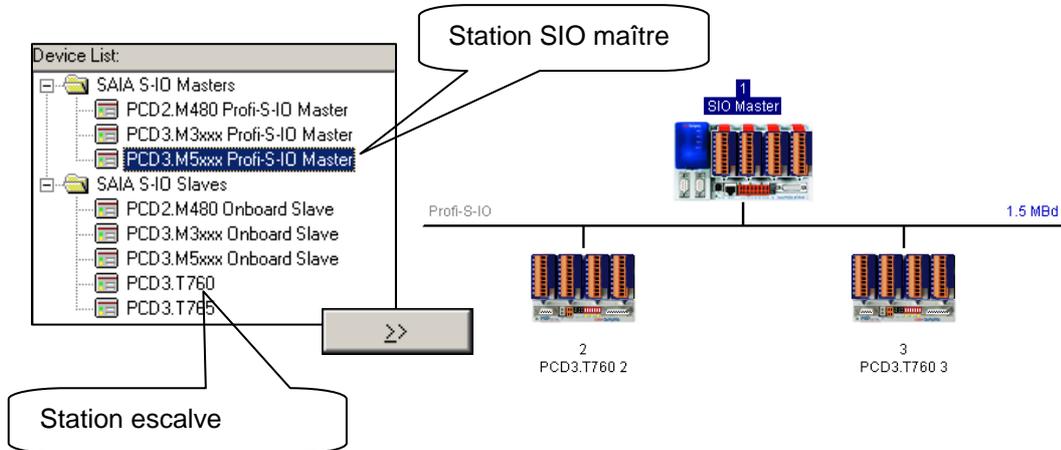
En fait l'usage de configurateur S-Net est identique pour l'échange de données sur un réseau Profi-S-IO et Profibus DP, les seules différences sont :

- L'extension du fichier de configuration: .SIO, .DP
- Les devices supportés sur le réseau: SIO = Saia devices, DP = devices Saia + d'autres fournisseurs.
- Les timings du réseau: Profile Snet, DP

Remarque :

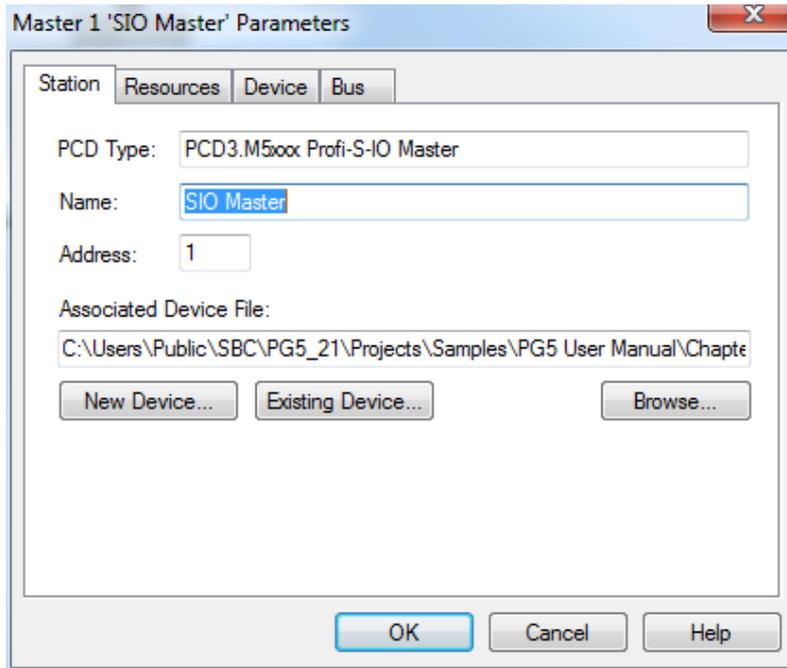
Si des PCD7.T7xx sont présents sur le réseau, toujours choisir le profile S-Net.

15.4 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.

15.5 Configurer la station maître

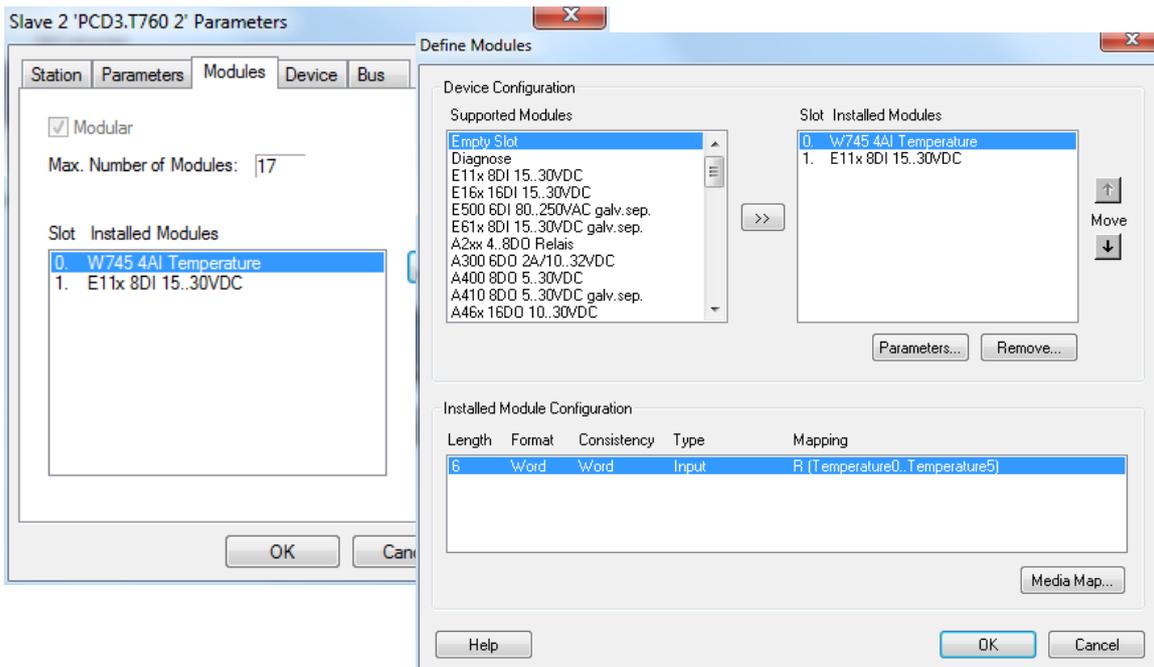


Une seule information doit être définie sur la station maître. C'est le chemin d'accès au device maître dans le quel S-Net doit sauvegarder le programme nécessaire à l'élaboration de la mémoire image pour la station maître.

Optionnellement, ce dialogue supporte aussi la modification du nom et de l'adresse du device CPU.

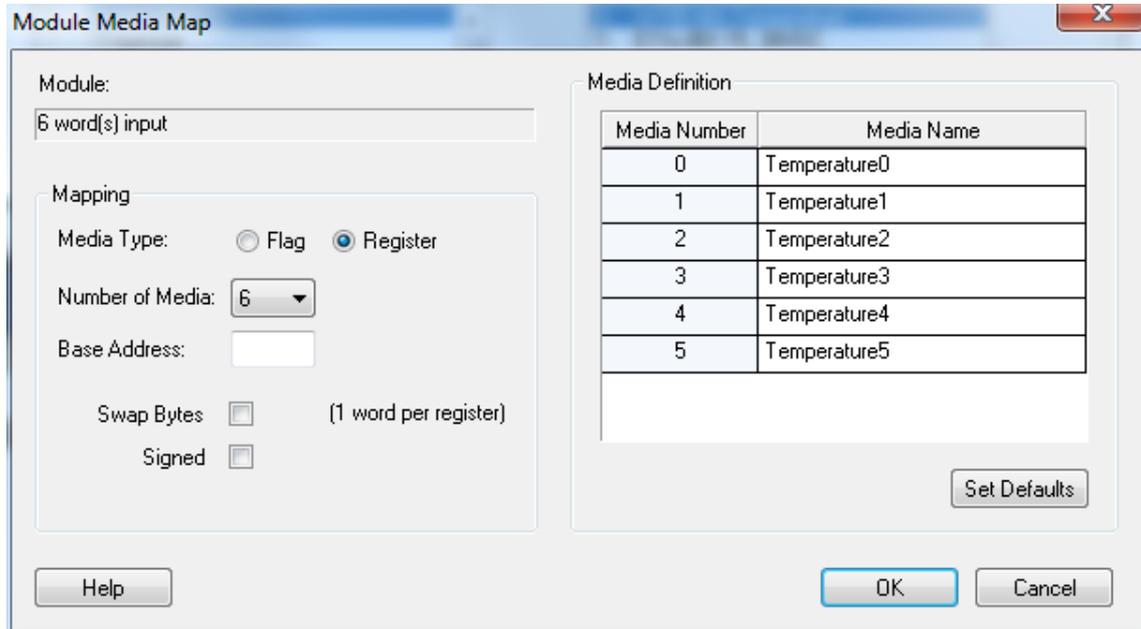
15.6 Configurer les stations esclaves

15.6.1 Configurer les modules d'entrées, sorties.



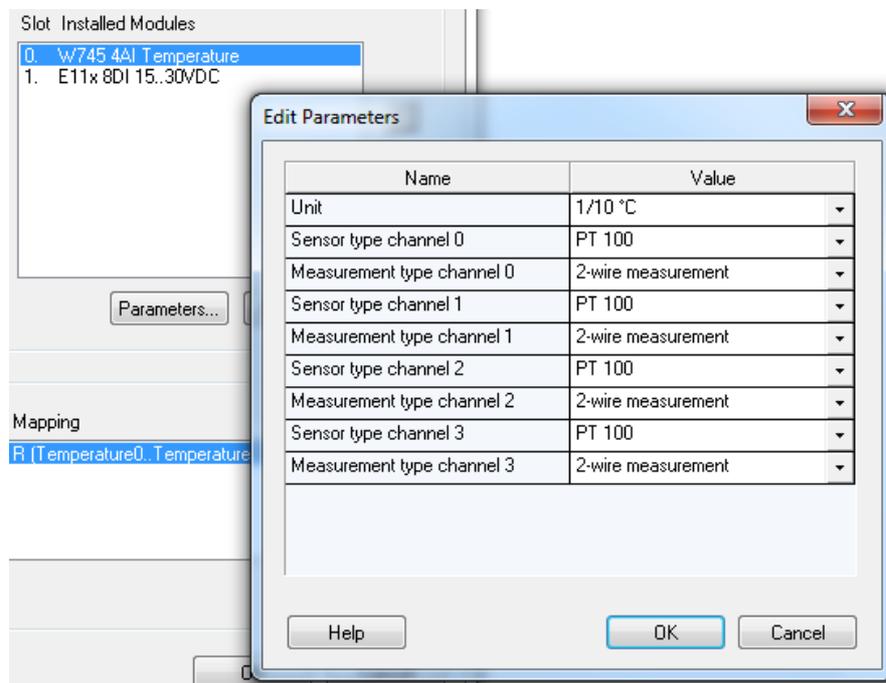
Pour chaque module d'entrée, sortie inséré sur la station esclave, sélectionner le type de module dans la liste *Supported modules* et l'ajouter dans la fenêtre *Slot Installed Modules* avec le bouton prévu à cet effet. Attention, le numéro de *Slot* et le type de module inséré dans la station esclave doivent toujours correspondre avec le numéro de *Slot* et le type de module présent dans la *fenêtre Slot Installed Modules* !

15.6.2 Configurer les symboles des données déportées



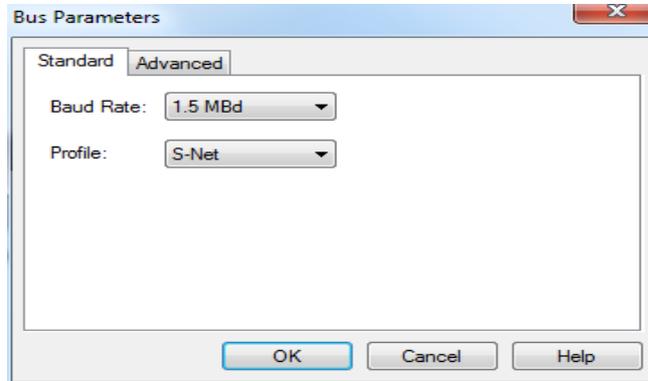
Pour chaque module présent dans la fenêtre *Slot Installed Modules*, sélectionner le module puis le bouton *Media Map* et définir les symboles de chaque donnée délivré par le module. Si nécessaire il est possible de définir l'adresse du premier flag ou registre à utiliser dans la station maître pour élaborer la mémoire image. Le plus simple est de ne rien définir, ainsi les adresses sont dynamiques.

15.6.3 Configurer les paramètres d'ajustages



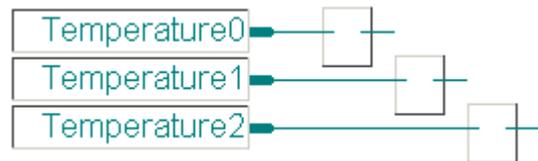
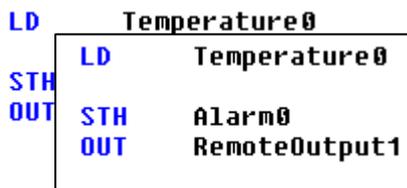
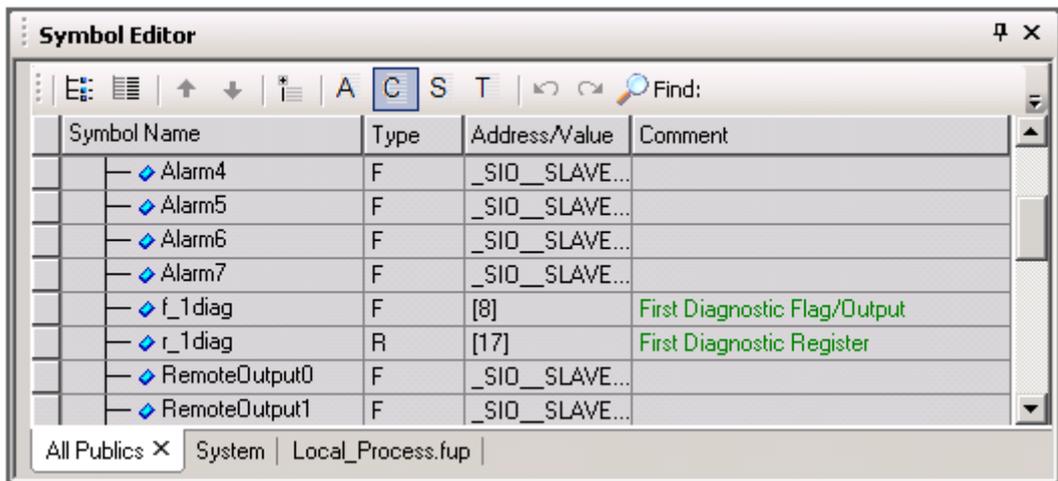
Avec certains modules comme les mesures analogiques, il peut être nécessaire de définir quelques paramètres supplémentaires pour la conversion de la mesure: unités de mesures, type de sondes de température, ...
 Ces informations peuvent être ajustées à l'aide de la fenêtre d'ajustage affichée en sélectionnant le module puis le bouton *Parameters*

15.7 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.

15.8 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.

15.9 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

Caractéristiques techniques

Configuration logicielle minimale (système d'exploitation)	PG5 2.1 fonctionne sous Windows 8 et 8.1 (64 bit) Windows 7 (32 et 64 bits) et Windows XP (32 bit). Microsoft .Net 4.0 Client Profile et Microsoft .Net 4.0 Extended. L'installeur .Net est disponible sur le disque d'installation : <DVD> :\Windows\dotNetFx40_Full_x86_x64.exe
Configuration matérielle minimale	Pour de meilleures performances, nous recommandons d'installer PG5 2.1 sur un ordinateur avec un processeur multi-cœur cadencé à au moins 2 GHz et une mémoire RAM d'au minimum 2GB (4GB ou supérieur est recommandé). L'installation du software nécessite un espace disque dure de 600 MB environ.
Jeu d'instructions PCD	Gestion de la totalité des 150 instructions du PCD
Boîtes de fonctions PG5 - FBoxes	Offre standard de plus de 250 boîtes de fonctions
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, PG4 et PG5 1.x
Communication	Transmissions sur TCP/IP, SAIA S-Bus®, PROFIBUS DP, BACnet et LONWORKS® intégrées au PG5



Saia-Burgess Controls AG
Bahnhofstrasse 18
CH-3280 Murten / Switzerland

Telephone ++41 26 672 71 11
Telefax ++41 26 670 44 43

E-mail: info@saia-pcd.com
Homepage: www.saia-pcd.com
Support: www.sbc-support.com

Your local contact: