PLUTO API de sécurité

Manuel de programmation

À propos de ce manuel
Ce manuel est divisé en deux parties : la première décrit comment se servir du logiciel de programmation Pluto Manager et la seconde les règles du langage de programmation.

La première partie débute par le chapitre « Écrire votre premier programme » qui vous guide à travers un exemple simple et constitue un bon début pour les tout nouveaux utilisateurs.

Le langage utilisé est lié à la norme de programmation IEC 61131-3. La programmation peut aussi s'effectuer sous forme de texte à l'aide d'un éditeur standard. Avant d'être chargé dans le système, le code doit être compilé au format hexadécimal. Le transfert d'un fichier hexadécimal dans un automate Pluto et la surveillance peuvent être effectués soit à l'aide de Pluto Manager, soit avec un programme standard comme Hyper Terminal.

Table des matières

Première partie 5

1 Consignes de sécurité 5
2 Cyber security disclaimer 5
3 Supported Windows versions 5
4 Installation 6
5 Écrire votre premier programme 7
5.1 Créer un nouveau projet 7
5.2 Nom et description 8
5.3 Inclure le fichier source 8
5.4 Enregistrer 9
5.5 Choix de la bibliothèque de blocs de fonctions 9
5.6 Configuration matérielle 10
5.6.1 Jeu d'instructions n°2/Jeu d'instructions n°3 11
5.7 Configuration des E/S 12
5.7.1 No Filt (Aucun filtre) 12
5.7.2 Désactivation des impulsions de test 13
5.8 Exemple de configuration des E/S 14
5.9 Appellation des variables 15
5.10 Programmation en logique Ladder 16
5.11 Ajouter des commentaires et finaliser un réseau 22
5.12 Réseau suivant 23
5.13 Connexion des composants 25
6 Projet Ouvrir, Enregistrer, Quitter …. 27
6.1 Protéger par mot de passe 28
6.1.1 Ouvrir un fichier protégé par mot de passe 29
7 Configuration du bus 30
7.1 Numéro de l'identifiant « IDFIX number » 31
7.1.1 Read IDFIX number from Pluto (Lire le numéro IDFIX sur Pluto) 31
7.2 « Advanced settings » (Paramètres avancés) 32
7.3 « External communication » (Communication externe) 32
8 Options des E/S 33
9 Fonctions du bus AS-i 34
9.1 Configuration initiale des fonctions AS-i 34
9.1.1 « New Pluto » (Nouveau Pluto), choix de la famille et du numéro de Pluto 34
9.1.2 Modes de fonctionnement du bus AS-i 35
9.1.2.1 Variantes du « monitor mode » : 36
9.1.3 Page des paramètres spécifiques à AS-i 37
9.1.4 Configuration manuelle des types d'esclaves (profils) 38
9.1.4.1 Undefined 38

9.1.4.2 Safe input 39
9.1.4.3 Nonsafe Standard slaves 41
9.1.4.4 Nonsafe A/B slaves 41
9.1.4.5 Combined Transaction A/B slaves 41
9.1.4.6 Analogue input slaves 41
9.1.4.7 Analogue output slaves (non-safe) 42
9.1.4.8 Safe Output 42
9.1.4.9 Pluto as Safe Input 43
9.1.5 Écrire un paramètre à l'esclave et recevoir des informations en retour 43
9.2 Configuration en ligne du bus AS-i 44
9.2.1 Lire les esclaves AS-i 44
9.2.1.1 Configuration en Monitor mode 45
9.2.2 Apprentissage des codes de sécurité 46
9.2.2.1 Activer la sortie d'un esclave 47
9.3 Autres outils en ligne 47
9.3.1 État du bus AS-i 48
9.3.2 « Show code table » (Afficher le tableau des codes) 50
9.3.3 « Teach code table » (Apprentissage du tableau des codes) 50
9.3.4 « Erase code table » (Effacer le tableau des codes) 50
9.3.5 Modifier l'adresse d'un esclave 51
10 Entrées analogiques Pluto D20 et D45 – blocs de fonction 52
10.1 Exemple d'application avec deux capteurs – mesure de la température 54
11 Entrées de compteur Pluto D45 55
11.1 Application avec deux codeurs – contrôle de vitesse 57
11.2 Application avec un codeur et une valeur analogique – contrôle de la vitesse 58
12 Variables 60
12.1.1 Nom symbolique 60
12.1.2 Description 60
12.2 Variables locales/globales 60
12.2.1 Exportation de variables 63
12.3 Variables rémanentes 65
12.3.1 Effacer les variables rémanentes 66
12.4 Exporter et importer les noms de variables 67
13 Programmation de la logique Ladder 68
13.1 Mode édition 69
13.2 Menu Outils 70
13.3 Update / Undo 72
13.4 Déploiement/Rétraction d'un réseau 72
13.5 Glisser-déposer 73
13.6 Options 75
13.7 Séquences 77
14 Paramètres du projet 78
14.1 Bibliothèque de fonctions 78
14.2 Fusion de projets 79
15 Compilation 80
16 Préférences générales 81
17 Fonctions en ligne 83
17.1 Communication 83
17.2 Menu « Tools » 83
17.2.1 Erase PLC Program (Effacer le programme de l'automate/Changer de mot de passe) 83
17.2.2 Online info 83
17.2.3 Copy online IDFIX to Clipboard 83
17.2.4 Terminal Window (fenêtre terminal) 84
17.2.5 Reset all Plutos 84
17.2.6 Write IDFIX 85
17.2.7 Upload Program from Pluto 85

17.2.8 Pluto System Software (Mise à jour des fichiers du système d'exploitation sur Pluto) 87
17.3 Transfert d'un programme 89
17.4 Insertion d'un nouveau Pluto dans un projet existant 90
17.5 Modification de la vitesse de transmission sur le bus (baud rate), erreur Er26 90
17.6 Online 91
17.7 Sceau 94

Deuxième partie 95

1 Instructions sur les bits 95
1.1 Adressage des opérandes sur les bits 95
1.2 Bits de registres (jeu d'instructions nº 3 seulement) 97
1.3 Instructions booléennes 98
1.4 Détection de front 100
1.4.1 Détection de front inversée (jeu d'instructions nº 3 seulement) 100
1.5 Fonctions SET et RESET 101
1.6 Fonction de basculement (Toggle) 102
1.7 Temporisations 103
2 Mémoires 105
2.1 Mémoires locales (M) 105
2.2 Mémoires globales (GM) 105
2.3 Mémoires du système (SM) 106
3 Séquences 107
3.1 Adressage 107
3.2 Saut 108
4 Opérations numériques 110
4.1 Registres 110
4.1.1 Adressage 110
4.1.1.1 Moitiés de registres doubles 110
4.1.2 Fonctions 111
4.1.3 Registres système 115
4.2 Utilisation des valeurs analogiques 117
5 Déclaration de programme en format texte 119
5.1 Identité, numéro de Pluto 119
5.2 Déclaration de code de programme 119
5.3 Déclaration des E/S 120
5.4 Noms symboliques 121
6 Syntaxe de programme en format texte : 122
7 Annexe A, compatibilité avec les Pluto antérieurs 123

Première partie
Pluto Manager

1 Consignes de sécurité

Noter que les erreurs de logique, un arrêt d'urgence ne commandant pas la bonne sortie par exemple, ne peuvent pas être détectées par le logiciel. Le programme doit donc être vérifié et les applications de sécurité testées minutieusement avant leur utilisation.

2 Cyber security disclaimer

It is your sole responsibility to provide and continuously ensure a secure connection between the product and your network or any other network (as the case may be). You shall establish and maintain any appropriate measures (such as but not limited to the installation of firewalls, application of authentication measures, encryption of data, installation of anti-virus programs, etc) to protect the product, the network, its system and the interface against any kind of security breaches, unauthorized access, interference, intrusion, leakage and/or theft of data or information. ABB Ltd and its affiliates are not liable for damages and/or losses related to such security breaches, any unauthorized access, interference, intrusion, leakage and/or theft of data or information.

For more information/contact regarding ABB cyber security see:
http://www.abb.com/cybersecurity

3 Supported Windows versions

Officially supported: Windows 7, Windows 10. It is highly recommended to use one of these.

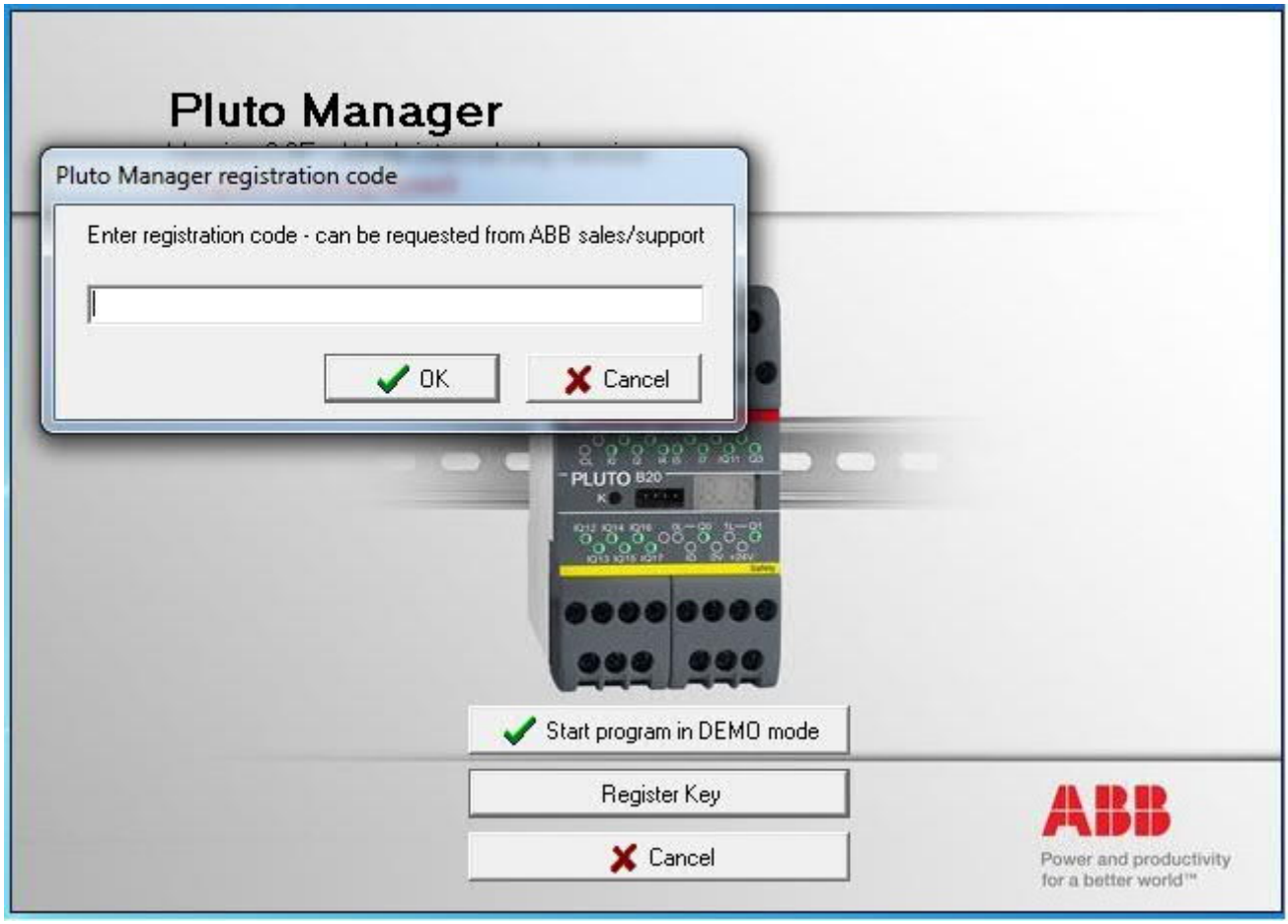
Other Windows versions:

Windows 8 and 8.1 should work, but are not fully tested.
Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016 and Windows Server 2019 probably work, but are not tested. Windows Vista may work, but is not recommended. Not all cyber security attack counter measures will be performed by Pluto Manager running Vista. Earlier Windows versions will not run.

4 Installation

Pour installer Pluto Manager, exécuter le fichier EXE (InstallPlutoManager…exe) sans aucun paramètre. L'utilisateur est guidé tout au long de l'installation et peut choisir un autre emplacement que celui proposé par défaut.

Un code de licence (registration code) vous est demandé. Sans code de licence, Pluto Manager peut être utilisé en mode DÉMO : il est alors possible de programmer mais impossible de compiler ou d'utiliser la fonction En ligne.



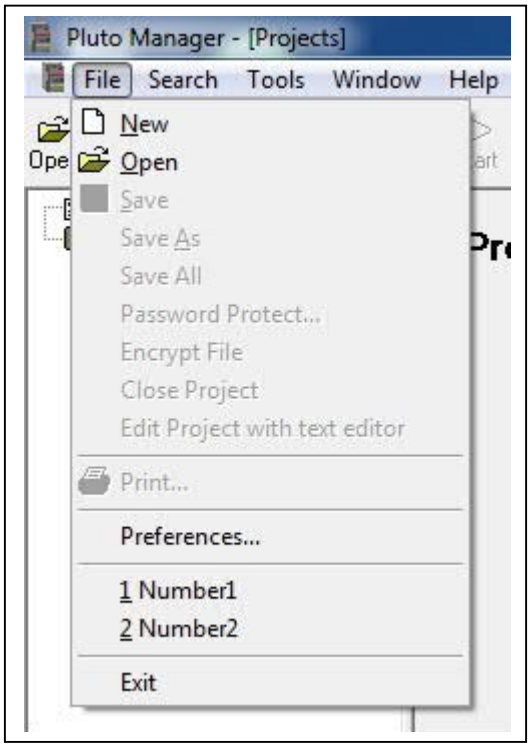
Un code de licence est nécessaire pour pouvoir utiliser la version complète de Pluto Manager. Ce code peut être saisi au démarrage du programme.

5 Écrire votre premier programme

La méthode la plus rapide pour s'initier à Pluto Manager est d'écrire un programme. L'exemple donné ici vous guide tout au long de la création d'un programme Pluto.

5.1 Créer un nouveau projet

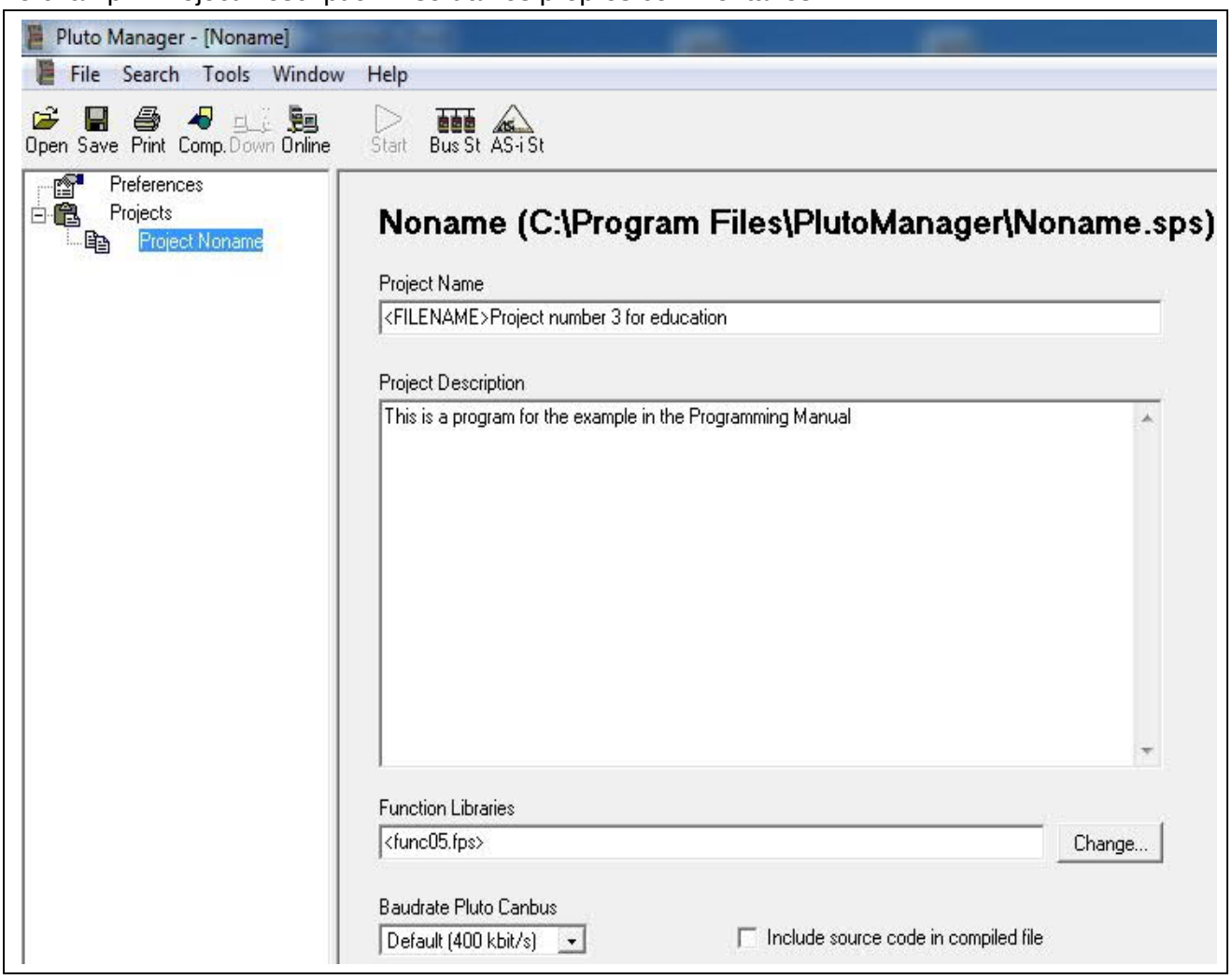
Pour créer un nouveau projet après avoir ouvert Pluto Manager, sélectionner « New » (Nouveau) dans le menu « File » (Fichier). L'option « Open » (Ouvrir) permet d'ouvrir un fichier déjà créé.



5.2 Nom et description

La première page qui s'affiche comporte les champs « Project Name » et « Project Description ».

« Project Name », le nom du projet, sera transféré dans les Pluto et contrôlé pour passer en ligne. <FILENAME> (nom du fichier) est le nom par défaut et sera remplacé par le nom donné au fichier du programme.
Le champ « Project Description » sert à vos propres commentaires.

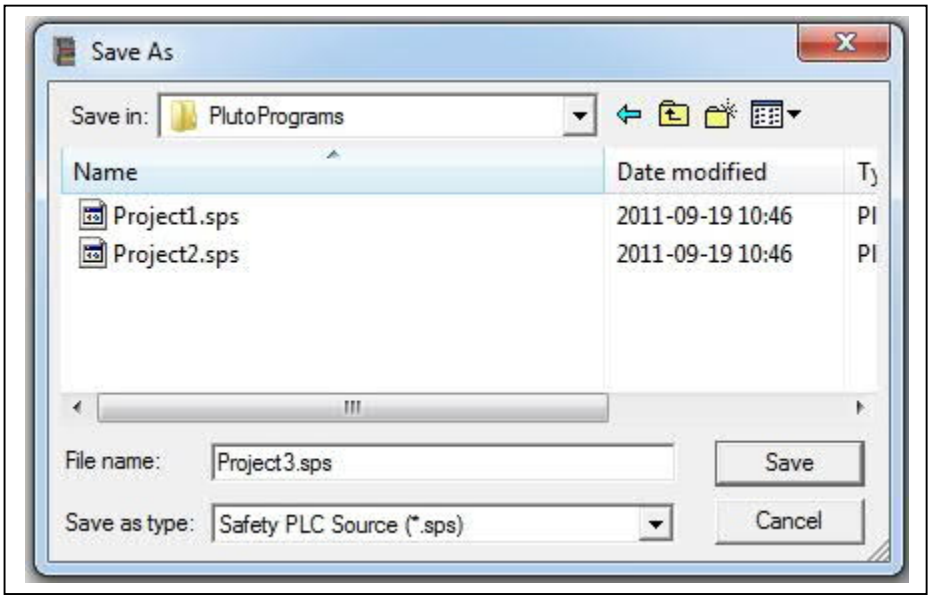


5.3 Inclure le fichier source

Si la case « Include source code in compiled file » est cochée (Inclure le fichier source dans le fichier compilé), le code source de l'automate est aussi transféré dans le Pluto et peut lui aussi être chargé dans un PC depuis le Pluto si nécessaire (upload). Toutefois, la taille du fichier est supérieure (ce qui peut poser problème si le programme est déjà gros) et toute personne disposant d'un PC et du mot de passe du Pluto peut récupérer le fichier source et le modifier.

5.4 Enregistrer

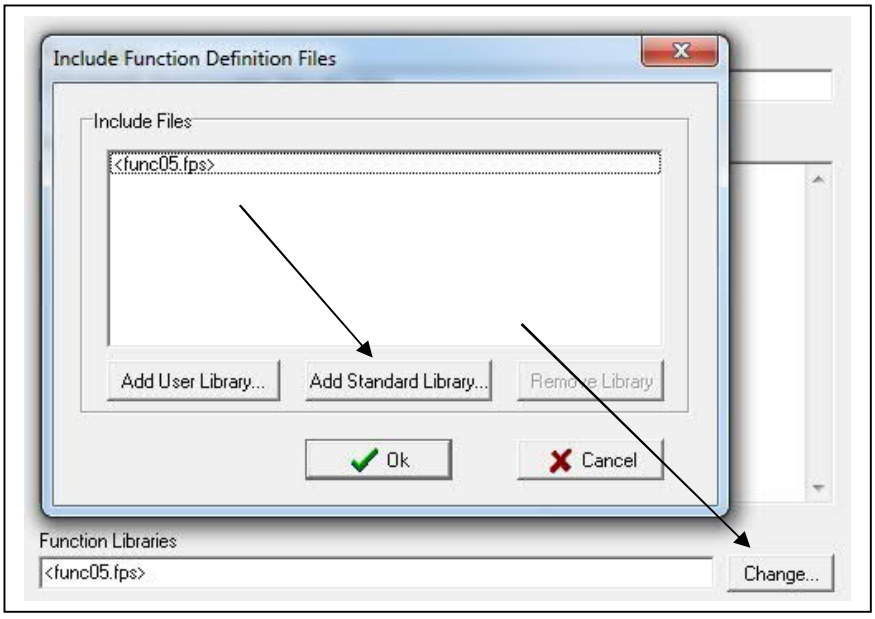
À ce stade, il conseillé d'effectuer une première sauvegarde.
La barre d'outils contient un bouton d'enregistrement rapide. Si le projet n'a jamais été enregistré auparavant, la fenêtre de dialogue « Save As » (Enregistrer sous) s'affiche.
Les options « Save » et « Save As » sont aussi accessibles dans le menu « File » (Fichier).
Sauf indication contraire, le fichier est automatiquement enregistré avec l'extension .sps.



5.5 Choix de la bibliothèque de blocs de fonctions

Le système Pluto propose un grand nombre de macros/blocs préprogrammés dédiés à des fonctions et des modules de sécurité divers. Ces blocs de fonctions sont enregistrés dans des fichiers de bibliothèques séparés. La bibliothèque standard est fournie par Jokab. L'utilisateur peut aussi créer ses propres bibliothèques.

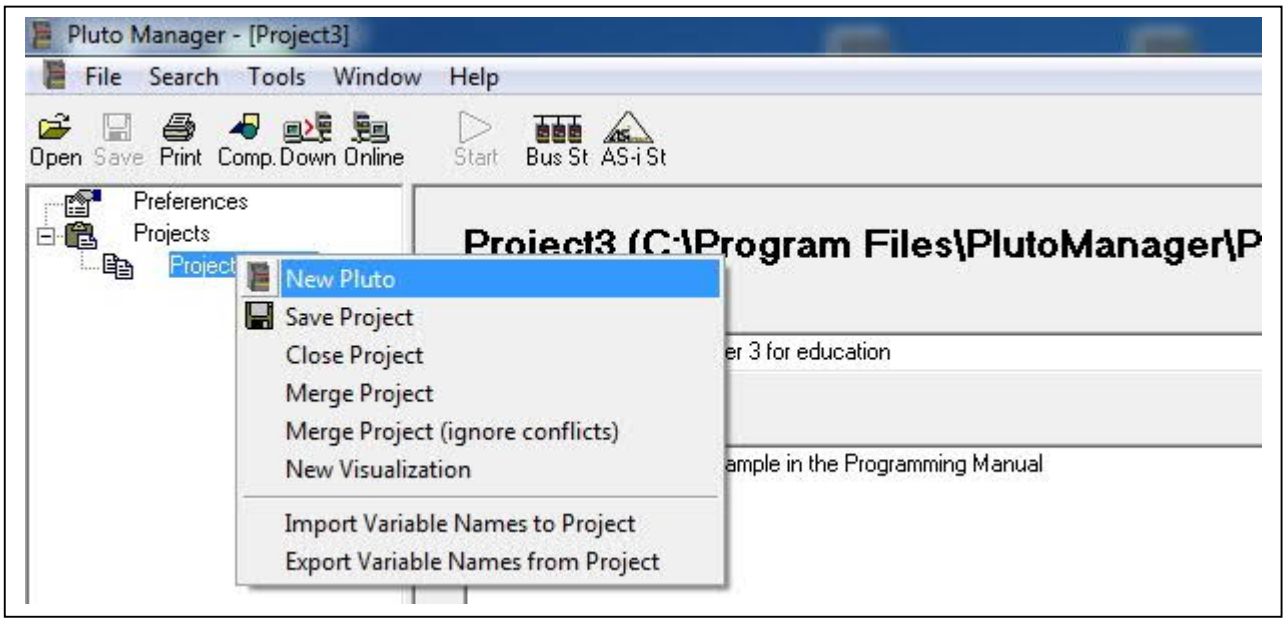
Sélectionner « Function library », « Change » puis « Add standard Library » (Bibliothèque de fonctions/Changer/Ajouter bibliothèque standard). Une liste des bibliothèques disponibles s'affiche alors.



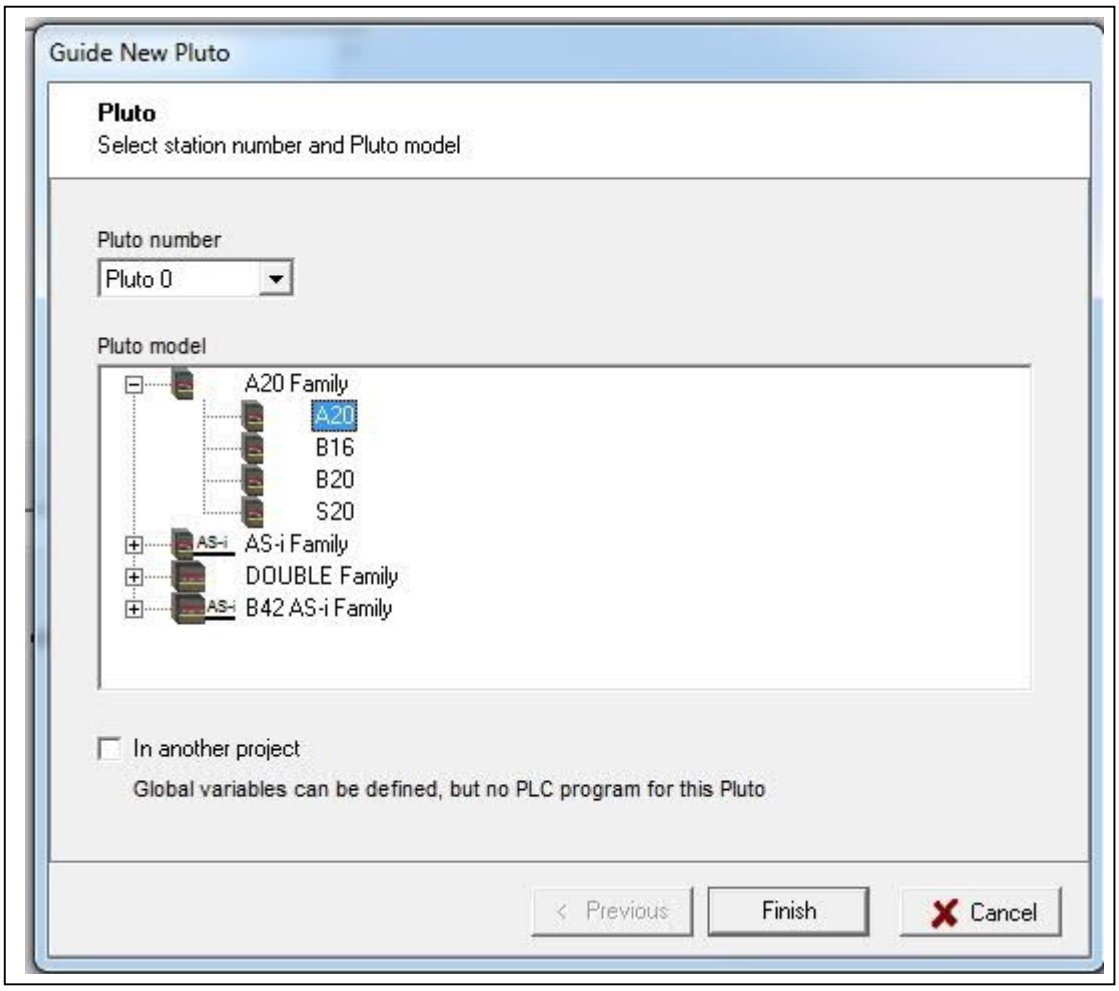
La fonction « Add standard Library » recherche dans le répertoire « ..\PlutoManager\Library » les fichiers normalement enregistrés à cet emplacement par le programme d'installation. Si « Add User Library » (Ajouter bibliothèque utilisateur) est sélectionné, Pluto Manager recherche les fichiers dans le dossier où sont enregistrés les fichiers de projets.

5.6 Configuration matérielle

L'étape suivante consiste à indiquer quel matériel est installé.
Aller dans le menu arborescent de gauche et cliquer du bouton droit de la souris sur le nom du projet. Sélectionner « New Pluto » (Nouveau Pluto) dans le menu qui s'affiche.

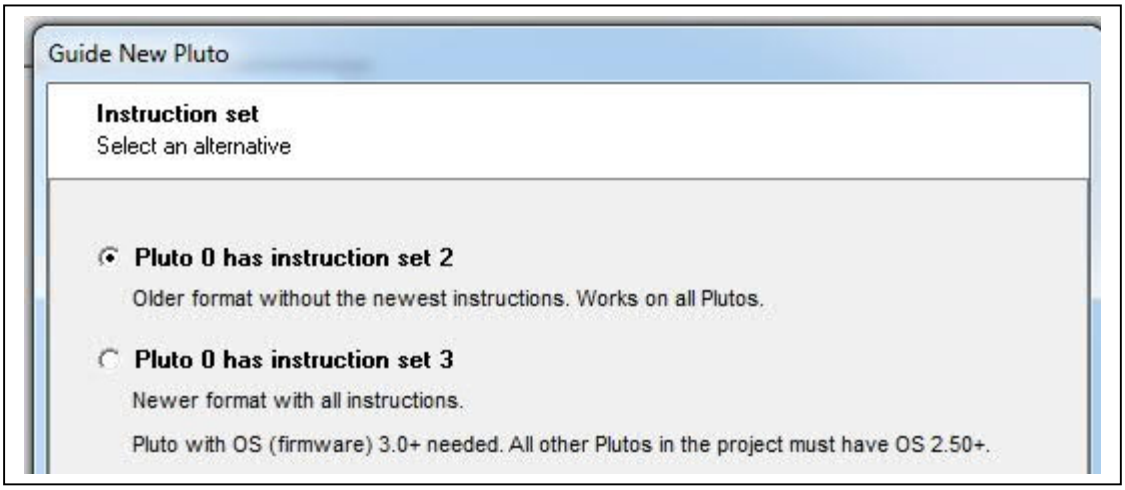


Une boîte de dialogue permettant d'entrer le numéro du Pluto s'affiche alors. Les numéros de Pluto sont compris entre 0 et 31.

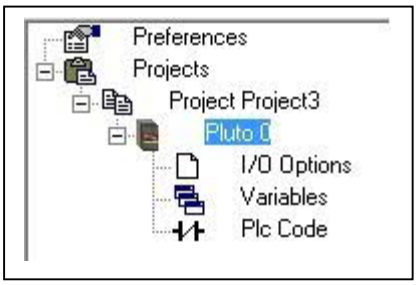


5.6.1 Jeu d'instructions n°2/Jeu d'instructions n°3

Lors du choix du type de Pluto, il est demandé de choisir le Jeu d'instructions à utiliser, nº 2 ou nº 3. Le jeu d'instructions n° 3 (Instruction set 3) est uniquement compatible avec les Pluto ayant un OS version 3.0 ou plus. Il contient de nouvelles instructions comme une temporisation au déclenchement (Off delayed timer), la multiplication et la division entre registres et constantes, des registres doubles (32 bits), la détection d'absence de front ("Not positive edge" et "Not negative edge"), la possibilité d'adresser des bits spécifiques de registres et une plage étendue d'adresses. Ces instructions sont décrites dans la deuxième partie de ce manuel.



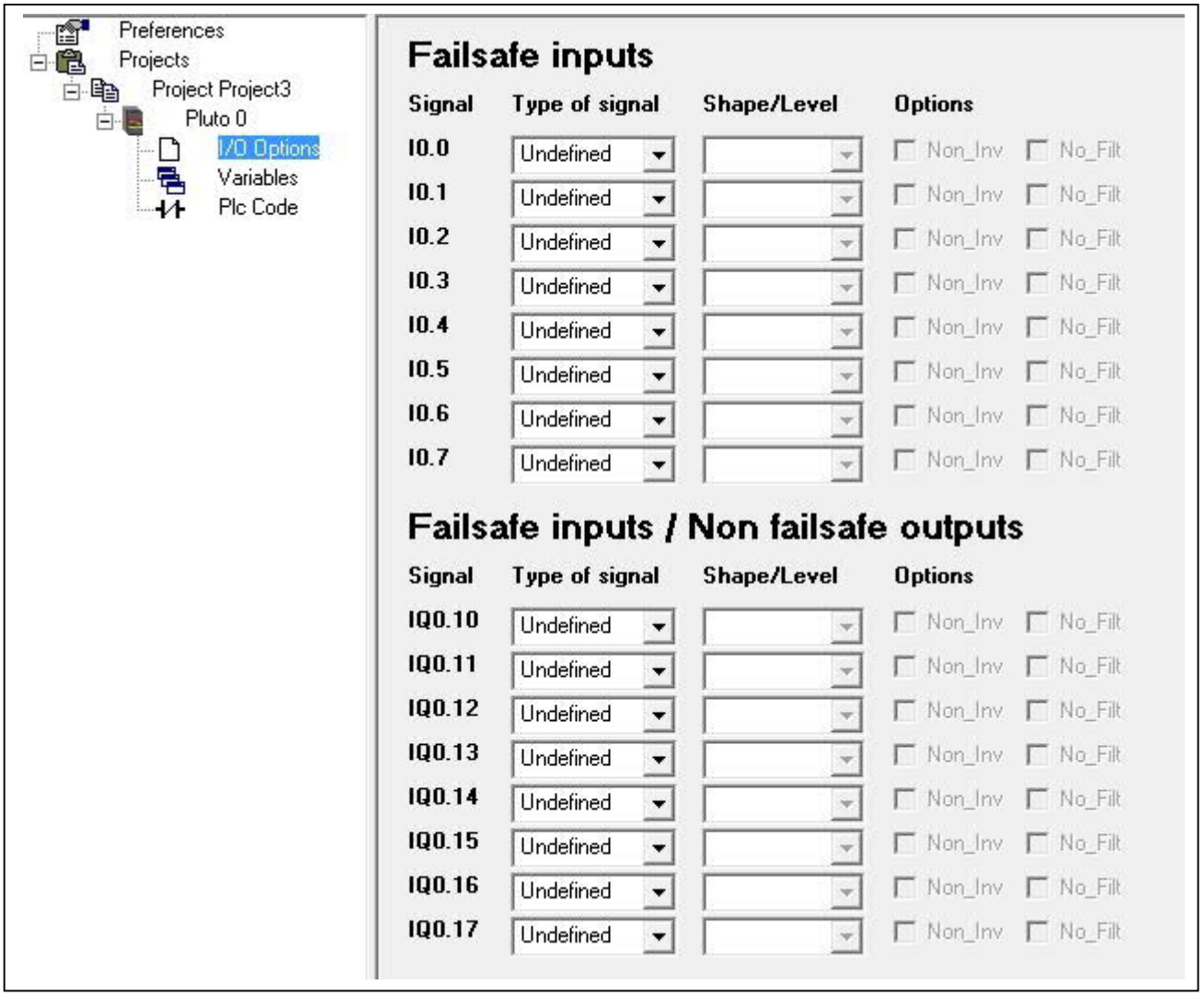
Une fois le numéro de Pluto, le type de Pluto et le jeu d'instructions choisis, le Pluto et ses pages « I/O options », « Variables » et « PLC code » sont ajoutés à l'arborescence.



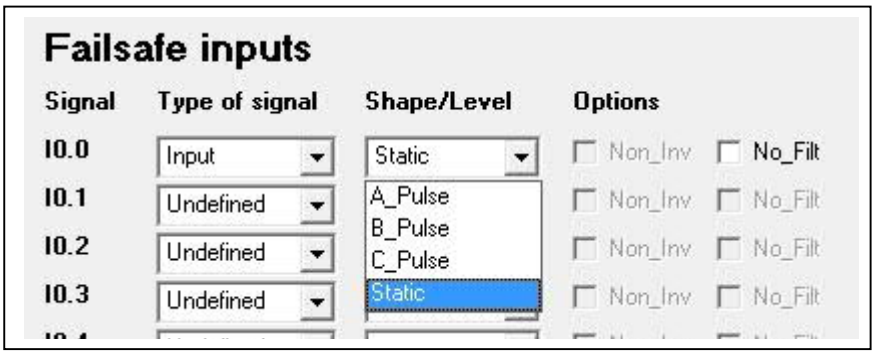
5.7 Configuration des E/S

Comme elles peuvent être utilisées de différentes manières, les E/S doivent être configurées. Leur configuration doit refléter le schéma électrique et le matériel utilisé.

La fenêtre « I/O Options » (Options des E/S) affiche les bornes de connexion I0…I7 et IQ10…IQ17. Les sorties de sécurité Q0…Q3 ne sont pas affichées puisqu'elles ne peuvent être utilisées que d'une seule manière.



Le paramétrage s'opère au moyen de menus déroulants.



5.7.1 No Filt (Aucun filtre)

Si la case « No_Filt » est cochée, le temps de réponse est réduit de 5ms mais la sensibilité aux perturbations augmente.

5.7.2 Désactivation des impulsions de test

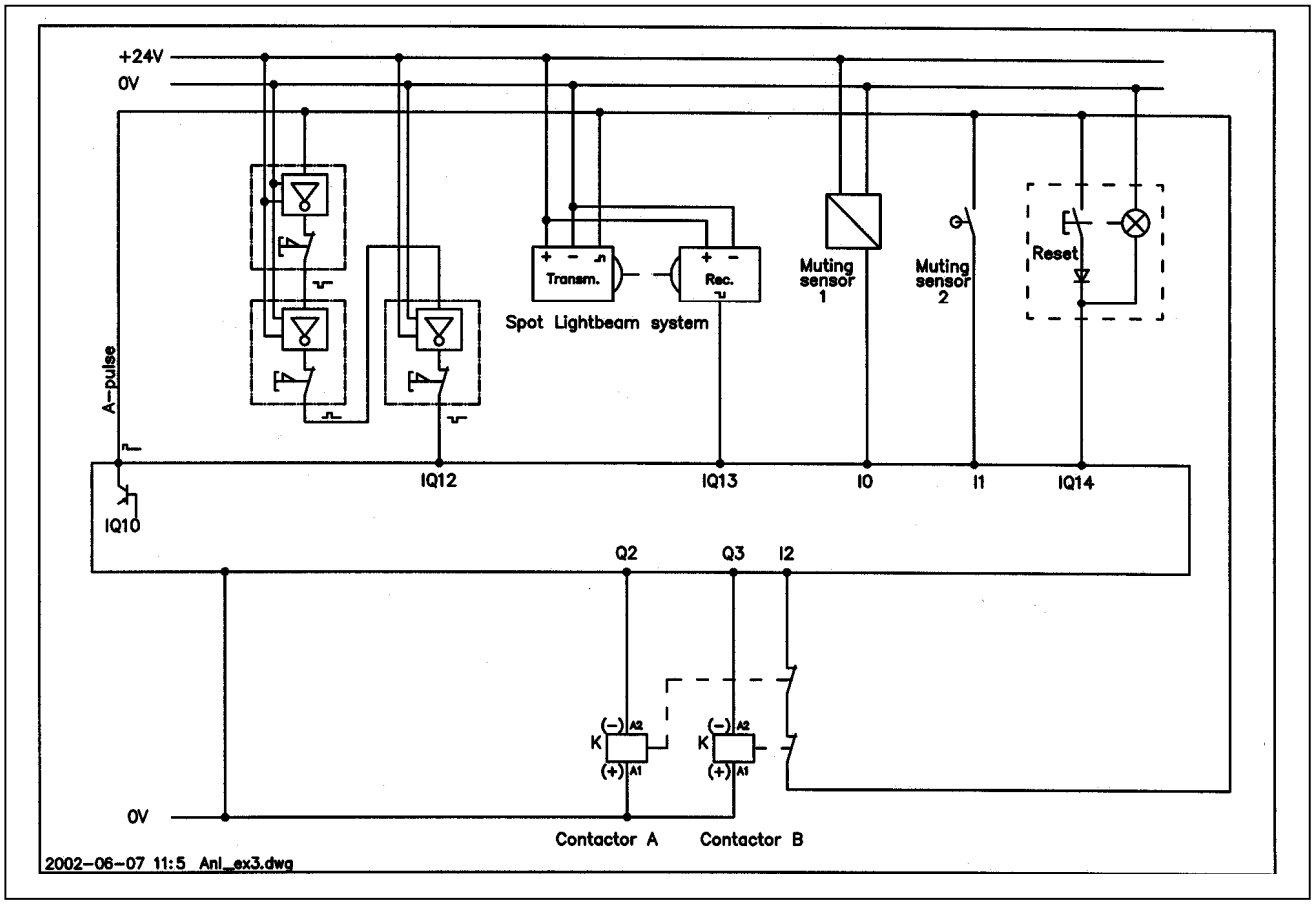
Les impulsions de test des sorties Q2 et Q3 (décrites dans le Manuel Matériel de Pluto) peuvent parfois poser problème selon les équipements connectés. Par exemple, des contacteurs modernes à capacitance élevée peuvent causer une Er40 dans le Pluto.

C'est pour cette raison qu'il est possible de désactiver ces impulsions de test dans les Pluto A20 v2, B20 v2, S20 v2 and Pluto D20. Toutefois, si elles sont désactivées, le Pluto ne peut plus détecter un court-circuit entre Q2 et Q3 ou entre Q2/Q3 et un autre Pluto.

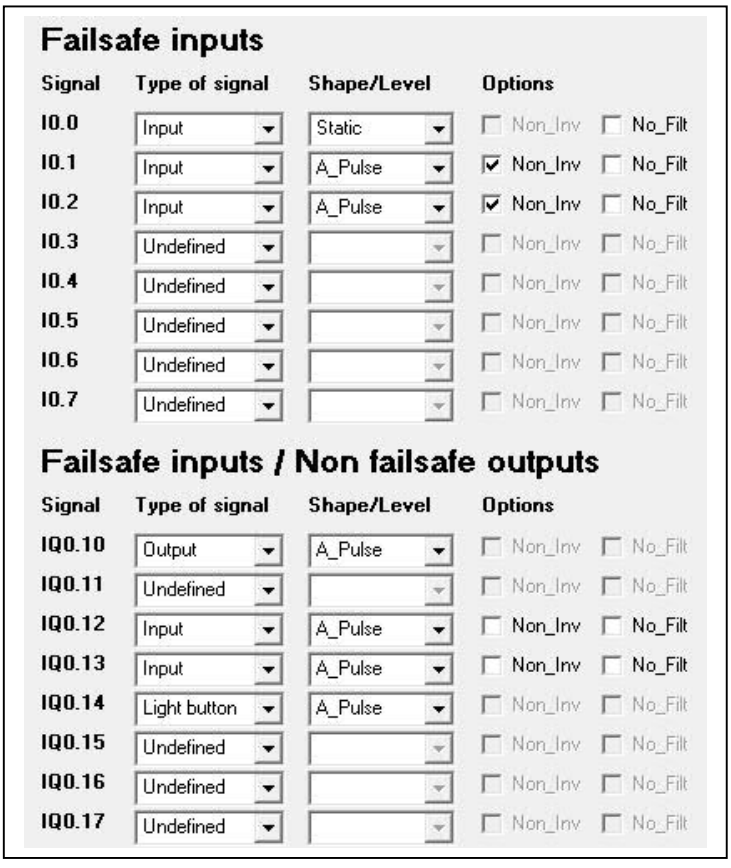
Dans Pluto Manager, les impulsions de test de Q2/Q3 peuvent être désactivées sur la page « I/O Options ».

5.8 Exemple de configuration des E/S

Les illustrations ci-dessous présentent d'abord un schéma de connexion puis la configuration correspondante dans la fenêtre « I/O-option » (Options des E/S).



REMARQUE : la configuration des E/S dépend de la conception matérielle. L'utilisation correcte des entrées, sorties, signaux dynamiques, etc., liés à la sécurité relève de la responsabilité du concepteur de l'installation.



Failsafe inputs

Signal	Type of signal	Shape/Level	Options
I0.0	Input	Static	☐ Non_Inv ☐ No_Filt
I0.1	Input	A_Pulse	☑ Non_Inv ☐ No_Filt
I0.2	Input	A_Pulse	☑ Non_Inv ☐ No_Filt
I0.3	Undefined		☐ Non_Inv ☐ No_Filt
I0.4	Undefined		☐ Non_Inv ☐ No_Filt
I0.5	Undefined		☐ Non_Inv ☐ No_Filt
I0.6	Undefined		☐ Non_Inv ☐ No_Filt
I0.7	Undefined		☐ Non_Inv ☐ No_Filt

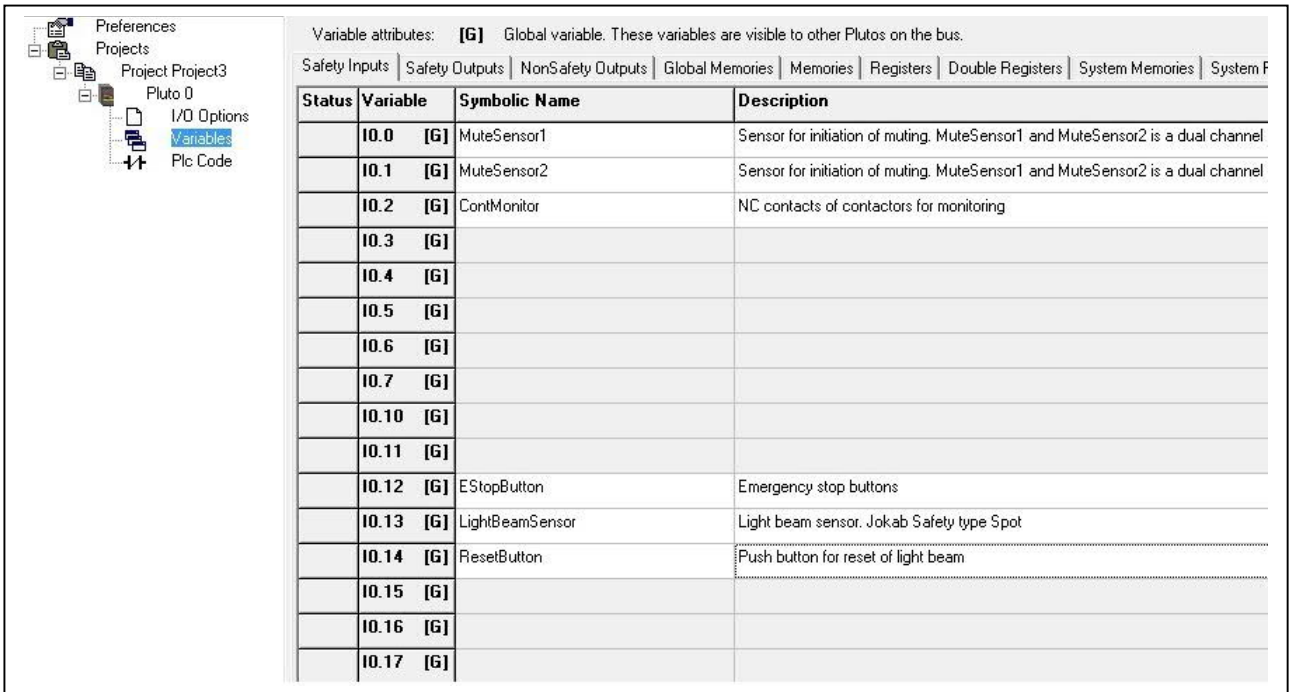
Failsafe inputs / Non failsafe outputs

Signal	Type of signal	Shape/Level	Options
IQ0.10	Output	A_Pulse	☐ Non_Inv ☐ No_Filt
IQ0.11	Undefined		☐ Non_Inv ☐ No_Filt
IQ0.12	Input	A_Pulse	☐ Non_Inv ☐ No_Filt
IQ0.13	Input	A_Pulse	☐ Non_Inv ☐ No_Filt
IQ0.14	Light button	A_Pulse	☐ Non_Inv ☐ No_Filt
IQ0.15	Undefined		☐ Non_Inv ☐ No_Filt
IQ0.16	Undefined		☐ Non_Inv ☐ No_Filt
IQ0.17	Undefined		☐ Non_Inv ☐ No_Filt

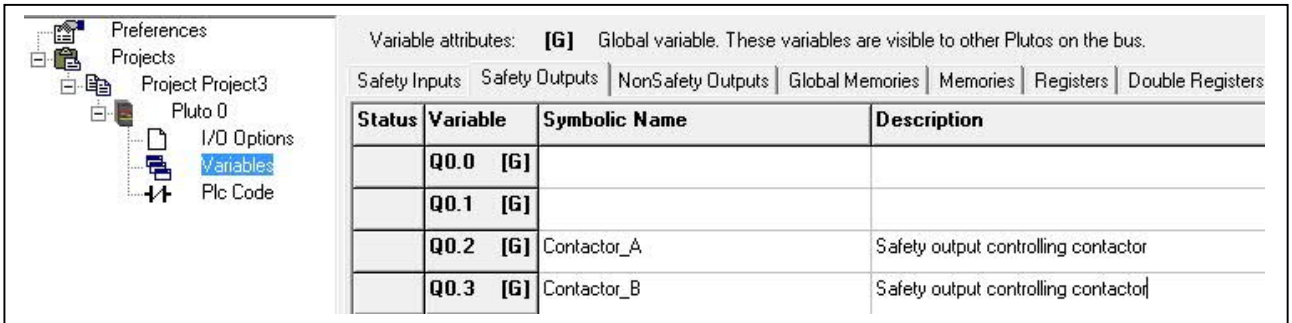
5.9 Appellation des variables

Ouvrir la fenêtre « Variables » en cliquant du bouton gauche sur le symbole correspondant dans le menu arborescent de gauche. Toutes les variables, entrées, sorties, mémoires, registres, etc., peuvent recevoir un nom symbolique qui sera utilisé dans le programme à la place du nom réel de la variable. Ce nom peut être attribué ultérieurement ou ne pas être attribué du tout.

La zone « Description » permet de rédiger un commentaire sur les variables.



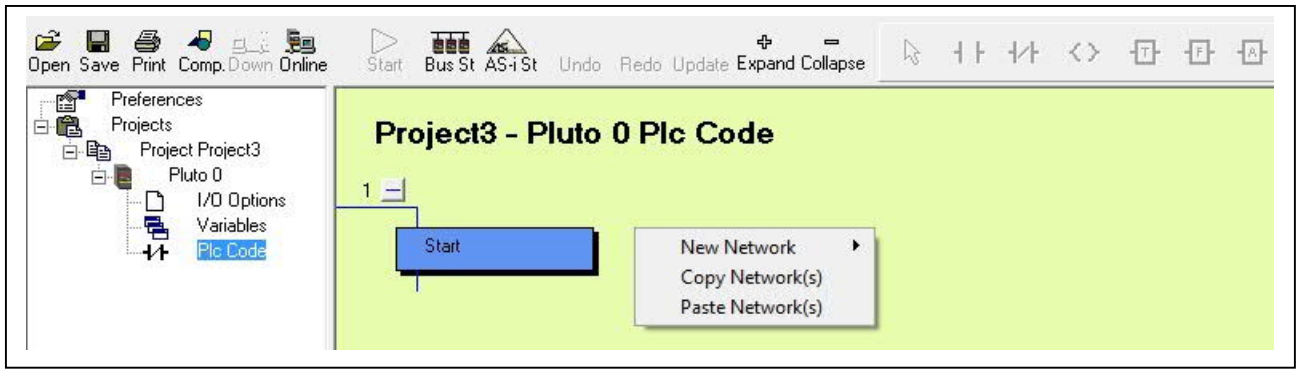
Noms et descriptions des entrées du Pluto 0.



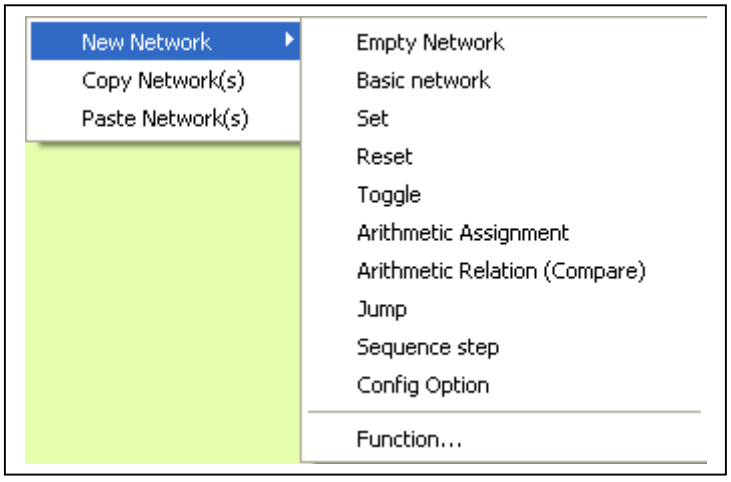
Noms et descriptions des sorties du Pluto 0.

5.10 Programmation en logique Ladder

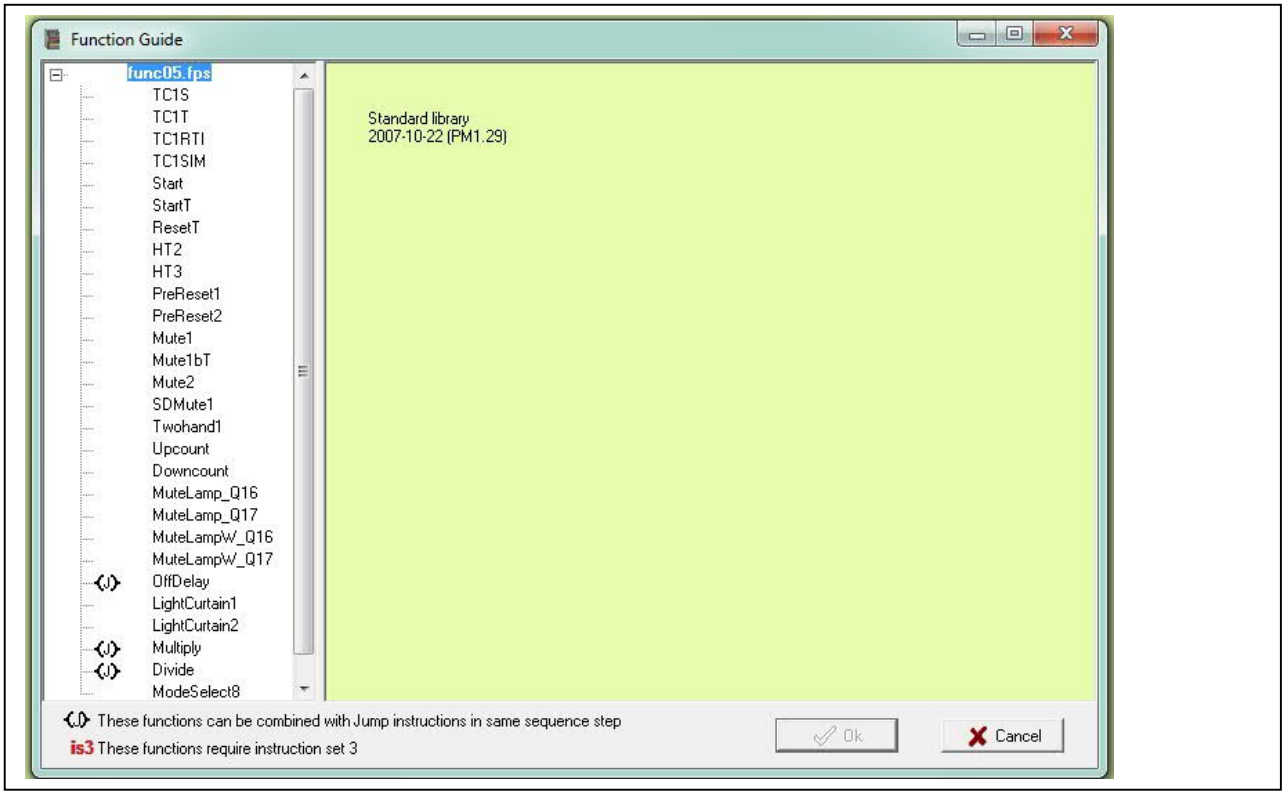
Ouvrir la fenêtre « PLC Code » (Code Automate) en cliquant du bouton gauche sur le symbole correspondant dans le menu arborescent de gauche. Placer le pointeur de la souris n'importe où dans la zone de droite puis cliquer du bouton droit pour créer un nouveau réseau qui viendra se placer à la suite de celui sur lequel pointe la souris. Un menu comportant quatre options, dont « New Network » (Nouveau réseau), s'affiche alors.



Pour créer un nouveau réseau, cliquer sur « New Network ». Un menu divisé par un séparateur s'affiche alors. La première partie présente des fonctions Ladder simples et la deuxième, sous le séparateur, les blocs de fonctions disponibles quand on clique sur « Function… ».

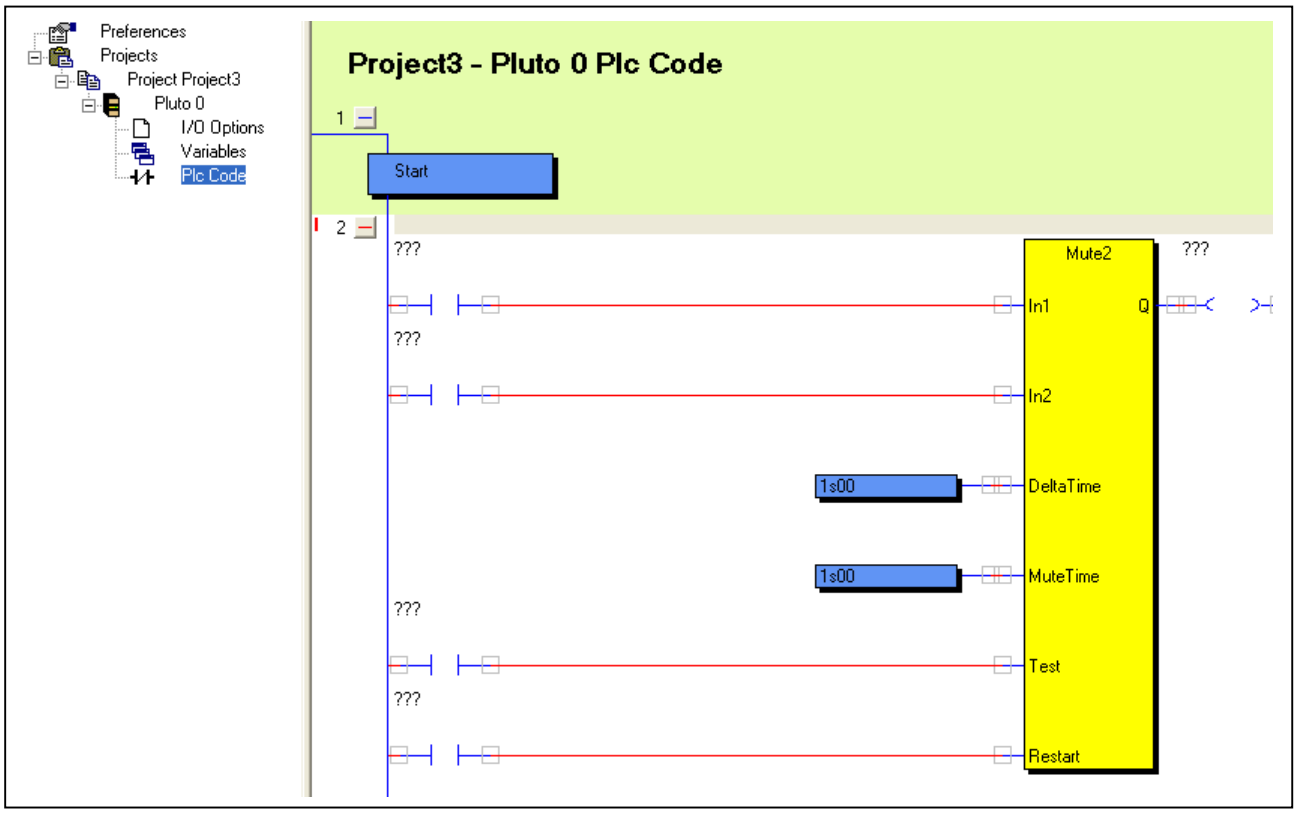


Un clic sur « Function… » ouvre la boîte de dialogue ci-dessous. Les blocs de fonctions disponibles peuvent être choisis dans la liste de gauche. Les blocs de fonctions sont décrits dans un document séparé.

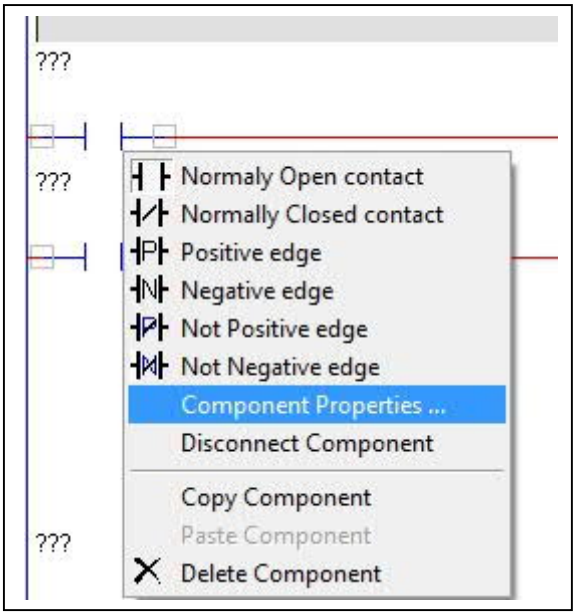


Dans cet exemple, nous avons besoin d'une fonction d'inhibition et avons constaté que le bloc « Mute2 » convenait. Un clic du bouton gauche de la souris sur « Mute2 » dans le menu crée un réseau Ladder contenant le bloc « Mute2 ».

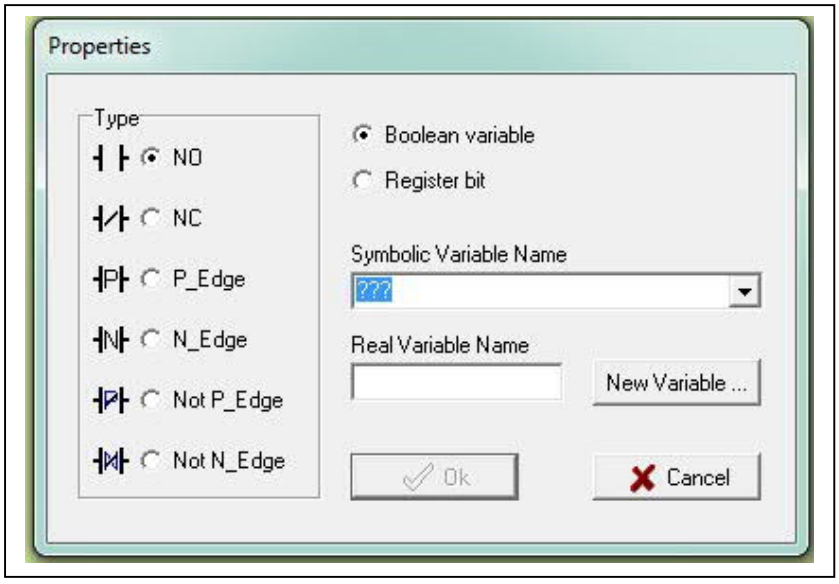
Le fond blanc du réseau correspond au mode d'édition. Chaque réseau doit être édité séparément.



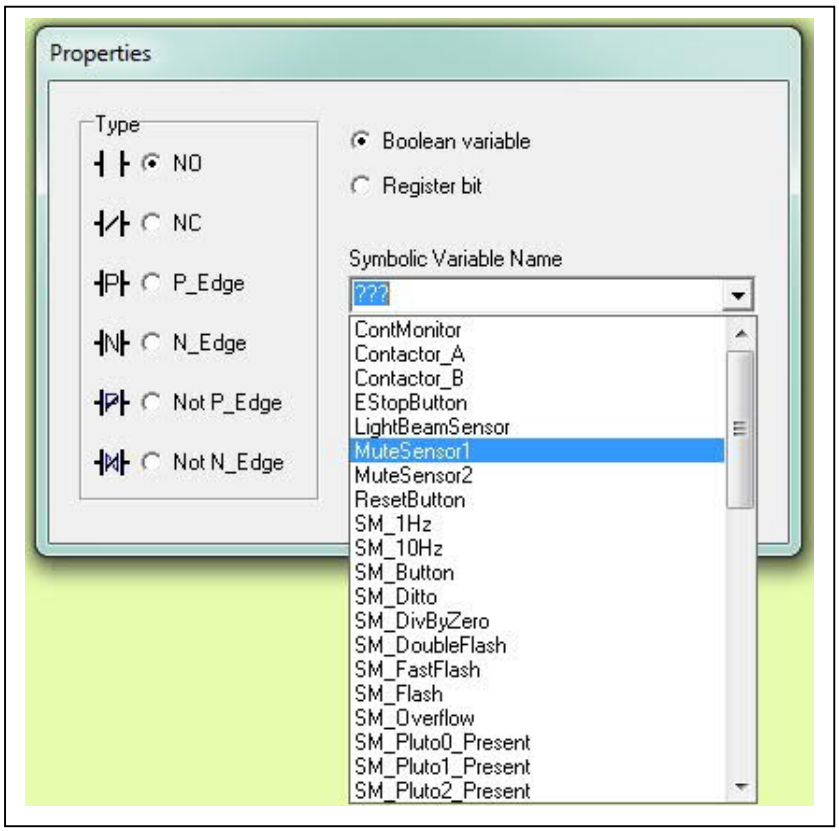
Les composants Ladder marqués de « ??? » doivent maintenant être définis ou dans certains cas supprimés. En cliquant du bouton droit de la souris sur un composant, trois options vous sont proposées. « Component Properties » ouvre une boîte de dialogue permettant de définir les propriétés du composant, « Disconnect Component » déconnecte le composant de la ligne de contact rouge et « Delete component » supprime le composant.
Double-cliquer du bouton gauche de la souris pour sélectionner directement « Component properties » (Propriétés du composant).



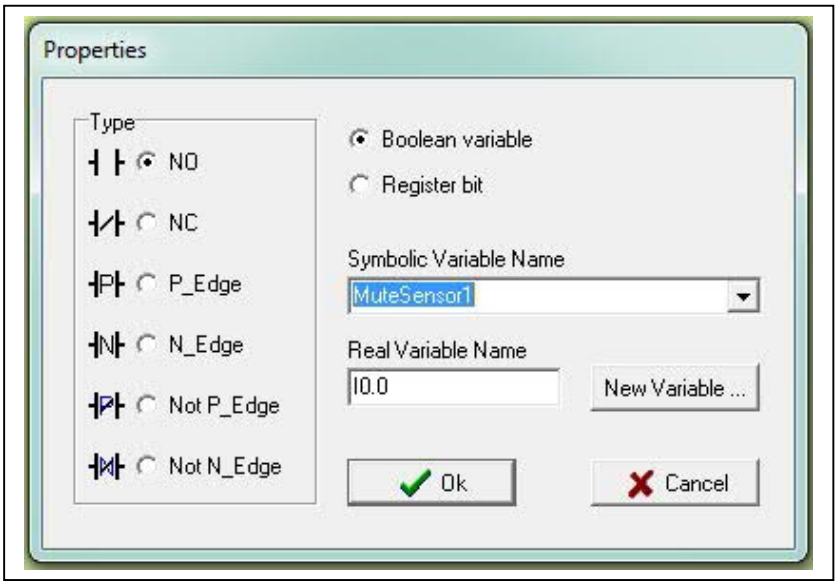
La boîte de dialogue « Properties » (Propriétés) contient les symboles de contacts correspondant aux options normalement ouvert, normalement fermé, front montant et front descendant.
Il existe deux façons de nommer une variable, soit en utilisant son vrai nom « Real variable name », p. ex. I0.0, I0.1, M0.3.., soit en ouvrant la liste « Symbolic Variable Name ».



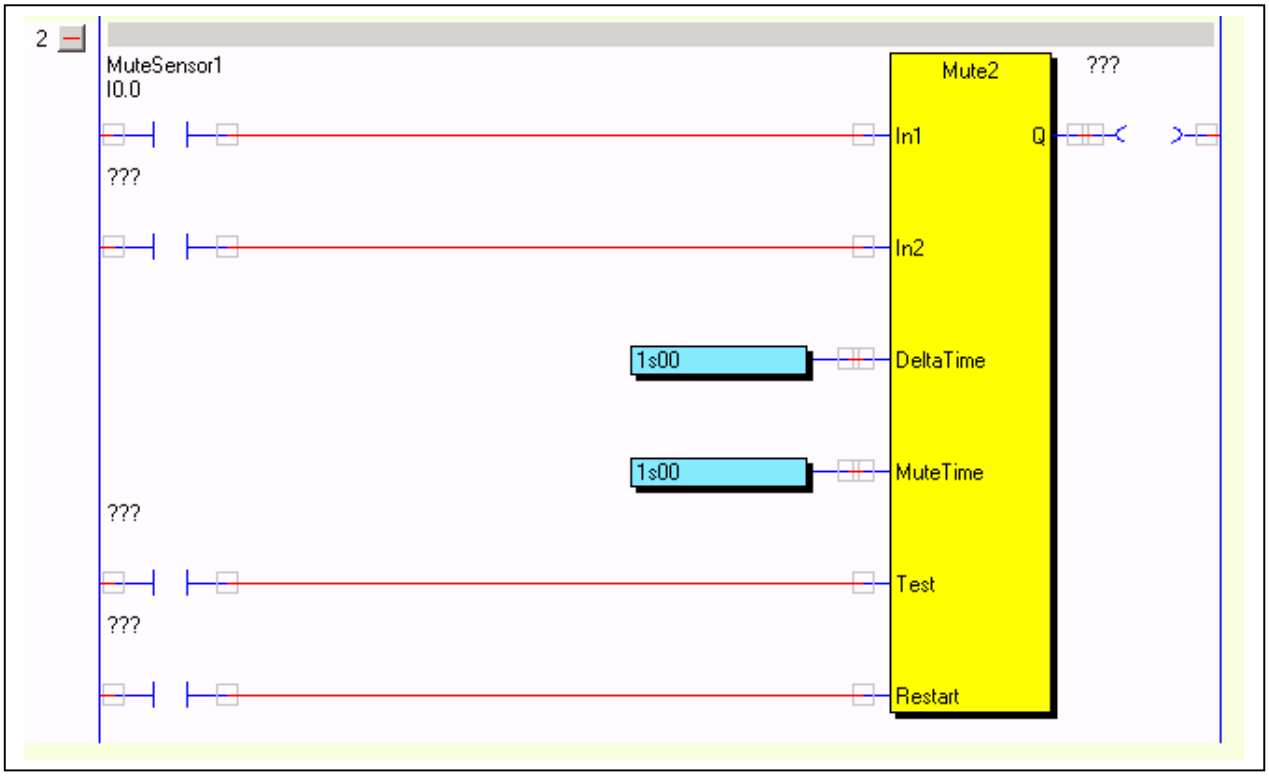
La liste sous « Symbolic Variable Name » contient tous les noms symboliques de variables déjà attribués.



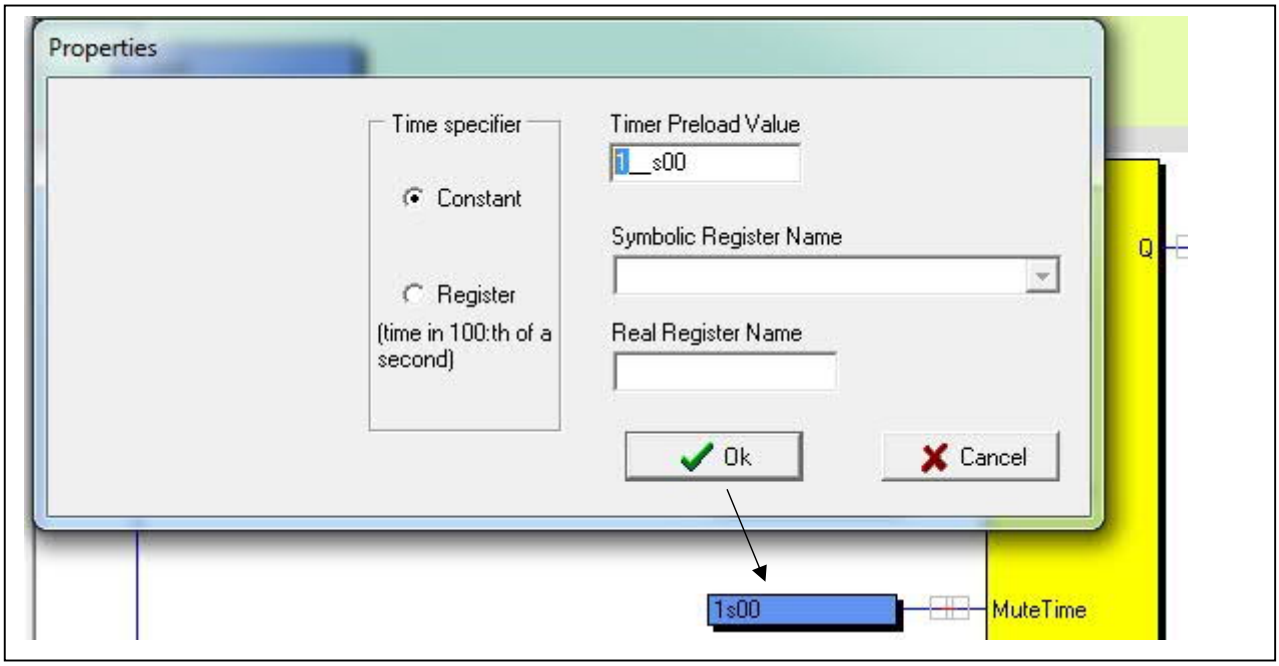
Confirmer en cliquant sur « OK ».



Une fois définis, le nom symbolique et le nom réel de la variable sont indiqués au-dessus du composant.

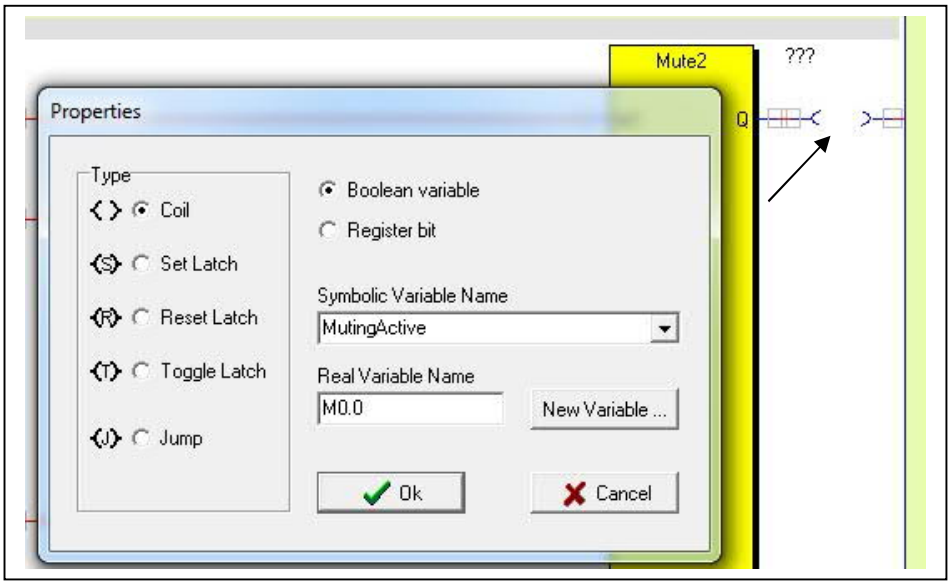


La valeur d'une temporisation est modifiée de façon similaire mais à partir d'une autre boîte de dialogue permettant d'indiquer la valeur de la tempo. « s » est utilisé comme point décimal/séparateur pour les secondes (1s00 = 1,00 seconde).

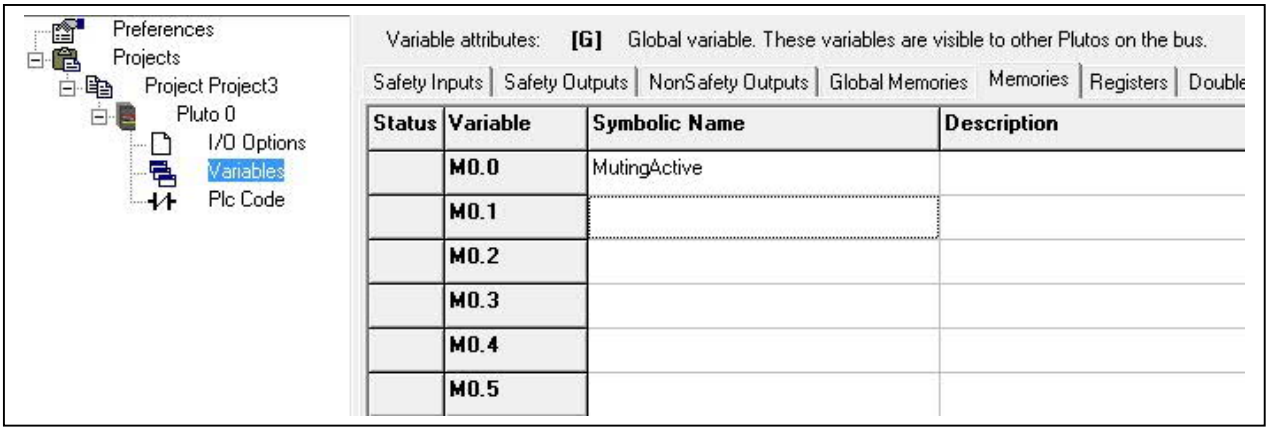


La sortie d'un bloc de fonction peut être associée directement à une sortie physique (Q), à une mémoire (M ou GM), comme M0.0 dans notre exemple, ou à l'entrée d'un autre bloc.

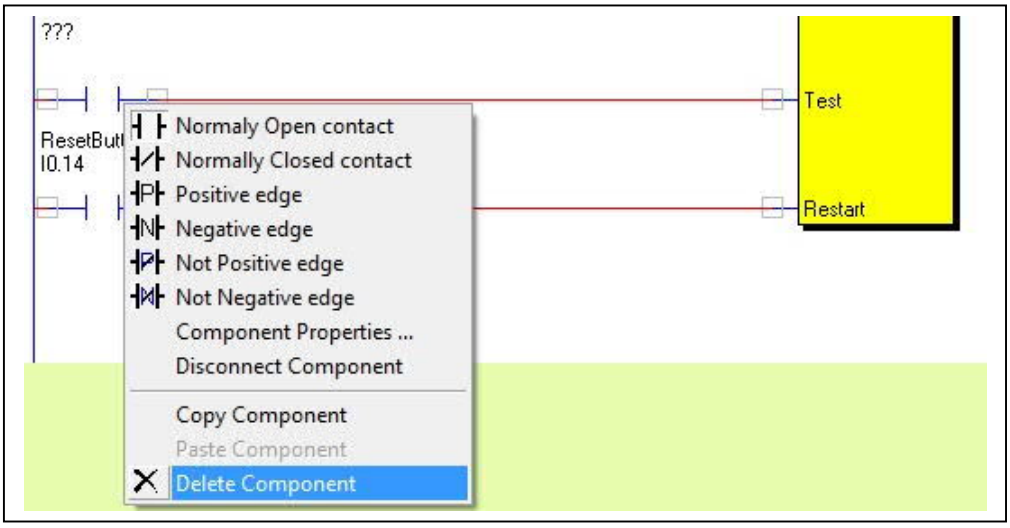
En double cliquant sur le composant Ladder, une boîte de dialogue s'affiche contenant diverses fonctions de sortie.



Pour éviter toute erreur, il est préférable de donner un nom aux mémoires dès leur toute première utilisation. Pour ce faire, ouvrir la fenêtre « Variables » en cours d'édition d'un réseau Ladder (sauf si une boîte de dialogue est affichée).



Dans notre exemple, l'entrée Test du bloc de fonction « Mute2 » ne doit présenter aucune condition d'entrée. Le composant est donc supprimé.



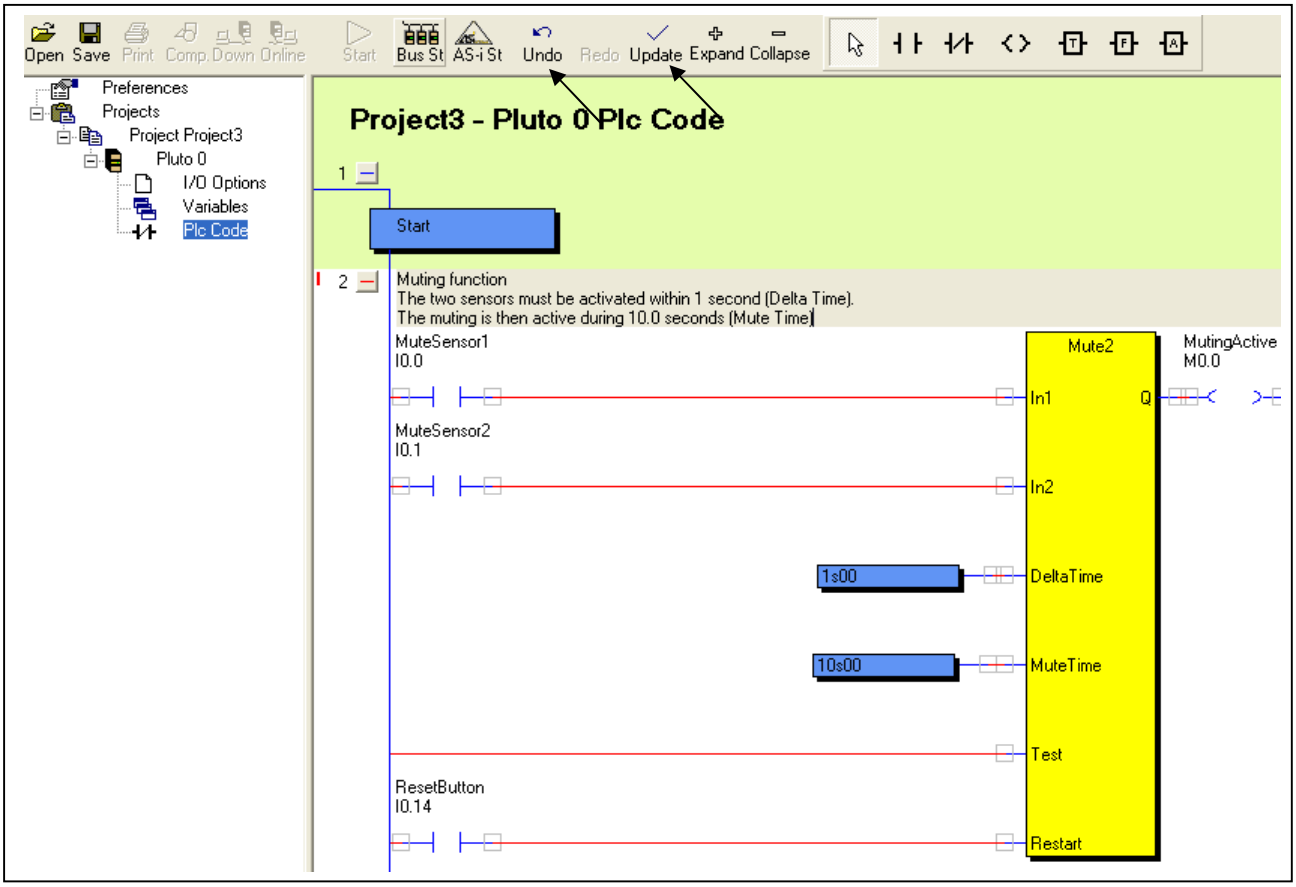
5.11 Ajouter des commentaires et finaliser un réseau

La zone grise en début de réseau permet de saisir des commentaires. Tout ce qui est écrit au clavier en mode édition est inséré dans ce champ.

Une fois l'édition du réseau terminée, quitter le mode édition
- En cliquant du bouton gauche sur « Update » (mettre à jour)

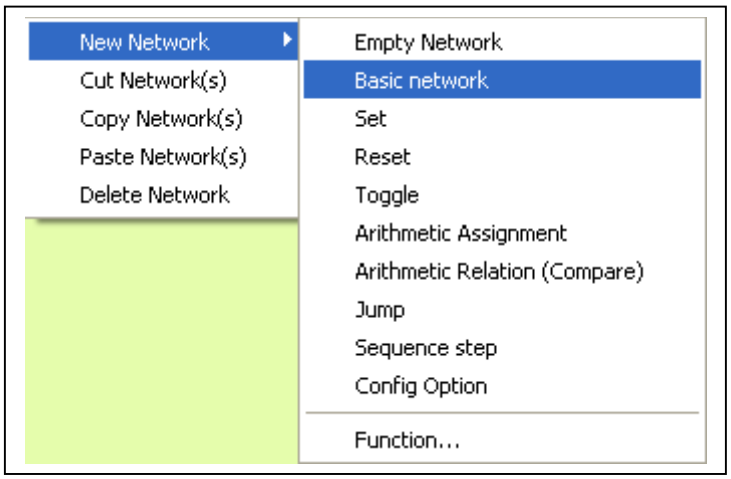
ou en :
- Appuyant sur la touche « F3 » ou en
- Appuyant sur « Esc » puis répondant « Yes » dans la boîte de dialogue.

En appuyant sur « Undo », toutes les modifications apportées au réseau sont effacées. La touche F2 peut être utilisée à la place de « Undo ».

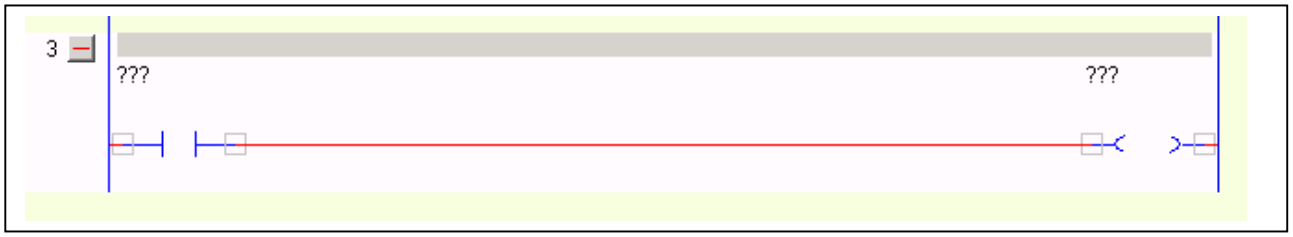


5.12 Réseau suivant

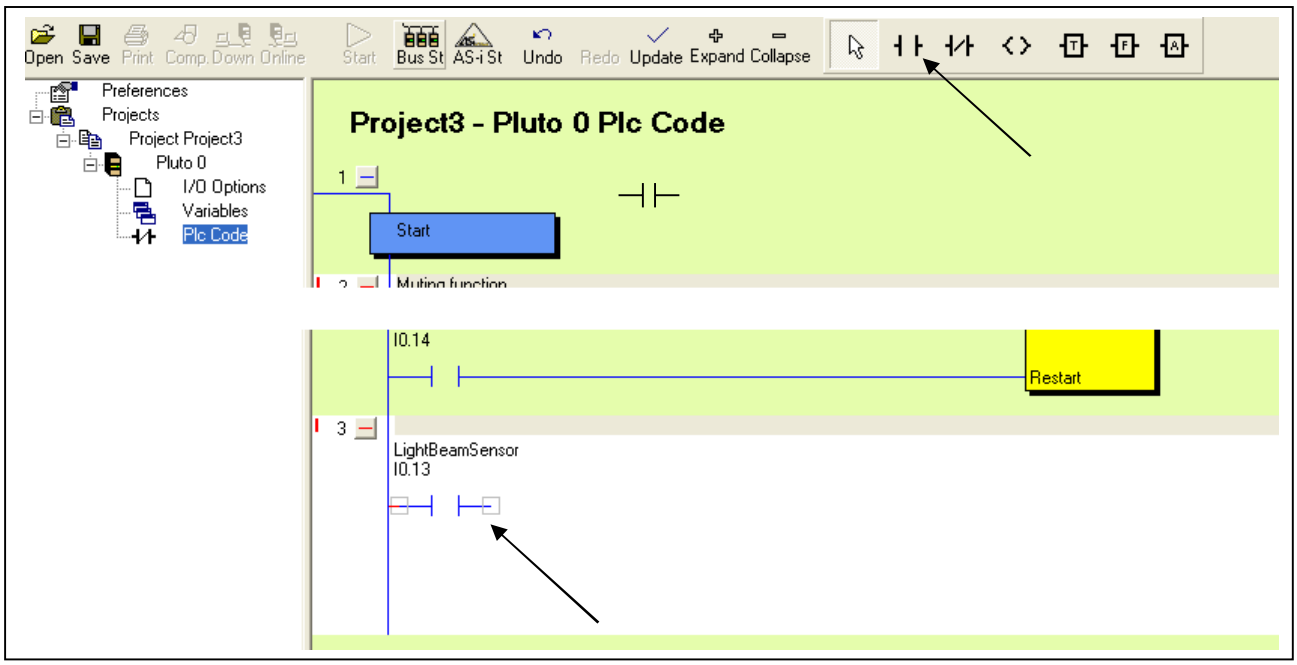
Dans le réseau suivant, nous allons configurer nos fonctions de sécurité et paramétrer une sortie de sécurité. À des fins d'exercice, nous choisissons cette fois « Basic network » (réseau de base) au lieu d'un bloc de fonction.
Cliquer du bouton droit quelque part dans le premier réseau. Sélectionner « New network » et « Basic network ».



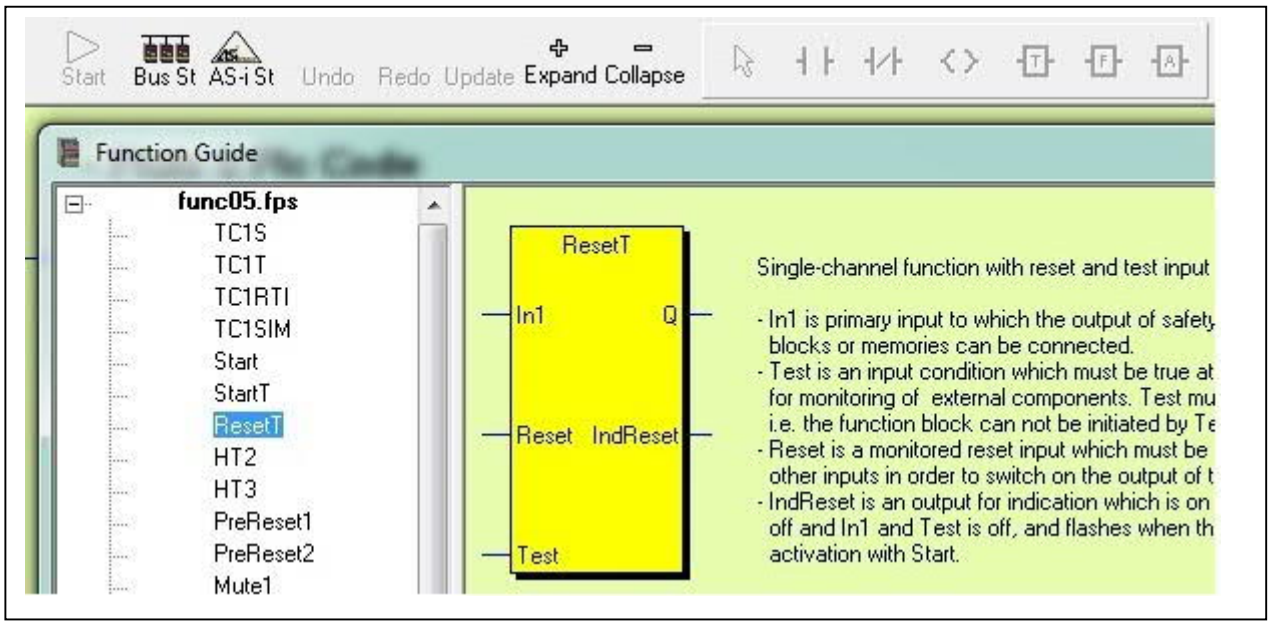
Nous obtenons un réseau avec un contact NO et une sortie.



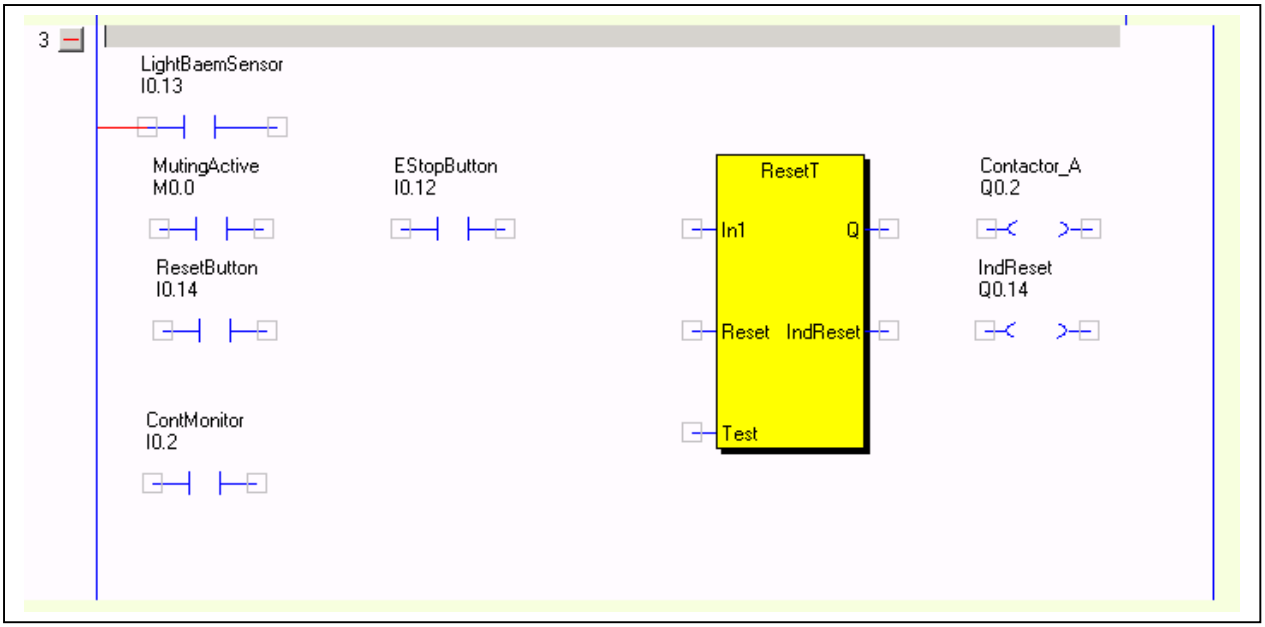
Supprimer la sortie et choisir « LightBeamSensor », dans la fenêtre des propriétés pour le premier contact Ladder. Nous devons maintenant ajouter un nouveau contact sélectionné dans le menu Outils. Cliquer du bouton gauche sur le symbole Contact NO. Le pointeur prend la forme d'un contact NO. Placer le contact où vous voulez dans le réseau, fixer son emplacement d'un clic gauche de la souris puis saisir ses propriétés.



Dans ce réseau, nous avons besoin d'un bloc de fonctions appelé « ResetT » qui est un bloc avec une entrée de sécurité pouvant effectuer la surveillance d'un bouton de réarmement lumineux. Cliquer sur le symbole F pour afficher la liste des blocs de fonctions disponibles, choisir « ResetT » et l'insérer dans le réseau.

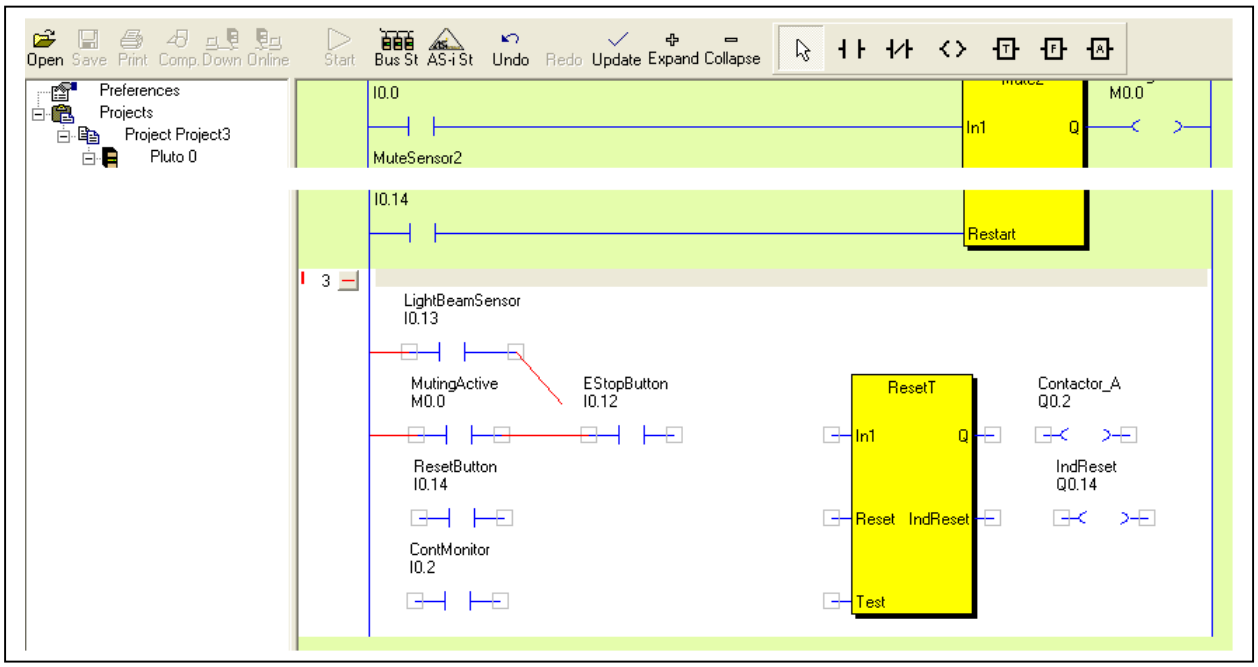


Sélectionner de la même manière le reste des composants nécessaires. Les blocs de fonctions se trouvent sous le symbole « F », les temporisations sous « T » et les fonctions arithmétiques sous « A ».

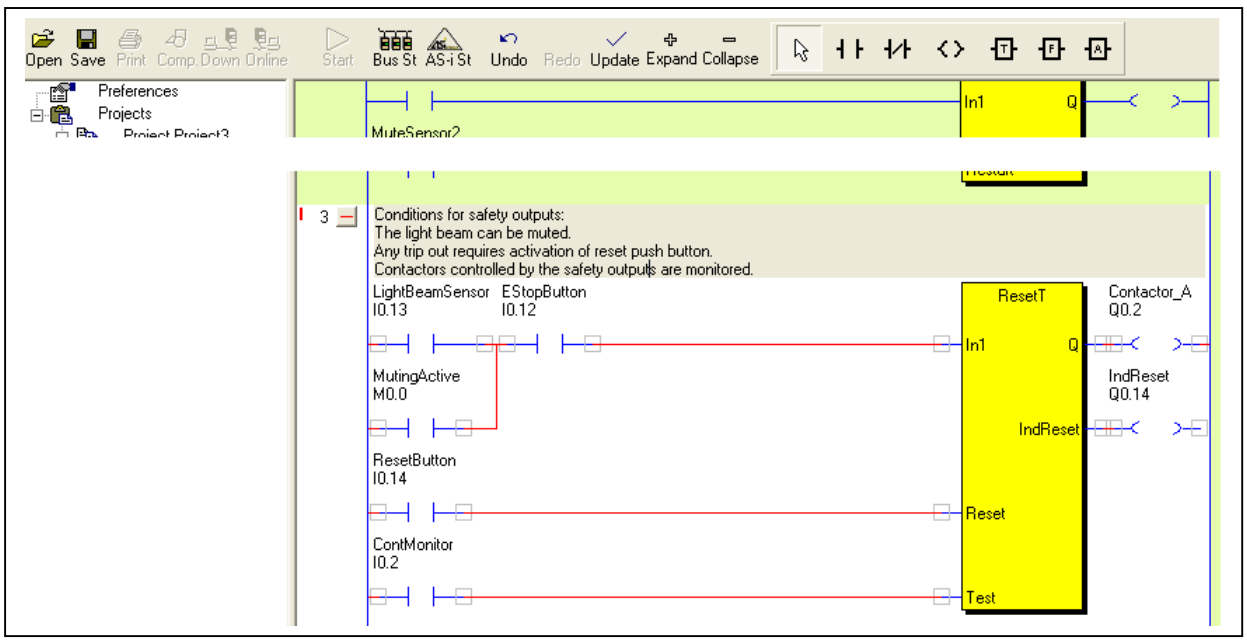


5.13 Connexion des composants

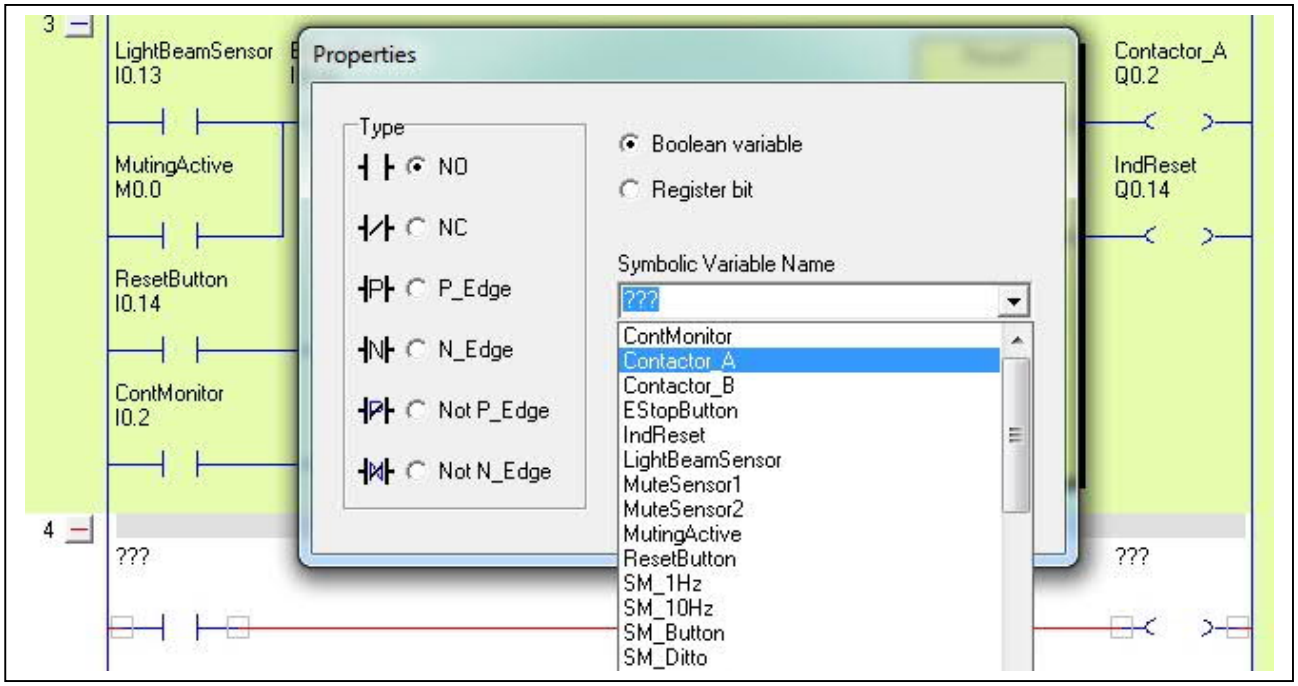
Quand la flèche du menu Outils est en surbrillance, il est possible de dessiner, supprimer et modifier les lignes entre les composants et de déplacer les composants. Les opérations « Dessiner une ligne », « Modifier une ligne », « Modifier les caractéristiques des composants », « Changer de composant » et « Déplacer les composants » sont décrites au chapitre 9,1 « Mode d'édition ».



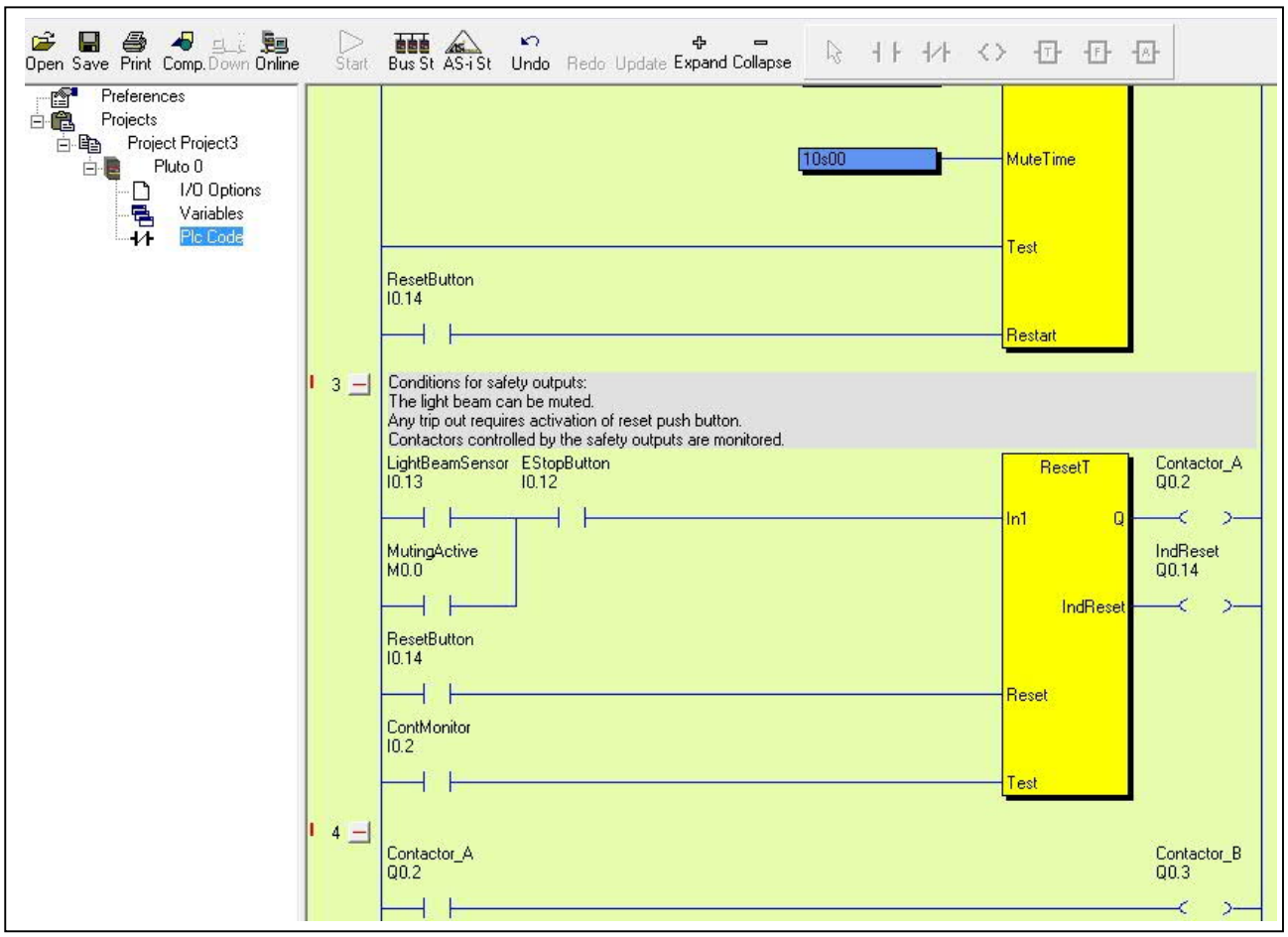
Une fois tous les composants placés et connectés, appuyer sur le bouton « Update » (mettre à jour) ou la touche F3.
Noter que la sortie du bloc de fonction « IndReset » (Indication réarmement) est une sortie secondaire pouvant être laissée déconnectée si vous n'en avez pas besoin. Si un composant (Q, M ou GM) est connecté à cette sortie, son côté droit sera connecté automatiquement à la ligne commune de droite lors de la validation du réseau.



Après la mise à jour, nous passons au dernier réseau de ce programme. La fonction de sécurité permet de commander les deux contacteurs A et B connectés chacun à sa sortie. Nous allons programmer le contacteur B pour qu'il fonctionne exactement comme le contacteur A. Au lieu de créer le même réseau que pour le contacteur A, on peut utiliser la variable « Contactor_A » (Q0.2) qui contient le résultat logique du réseau précédent. Pour cela, créer un réseau de base (basic network), ouvrir la boîte de dialogue « Properties » du premier contact et choisir « Contactor_A » dans la liste « Symbolic Variable Name ». Choisir « Contactor_B » pour la sortie.

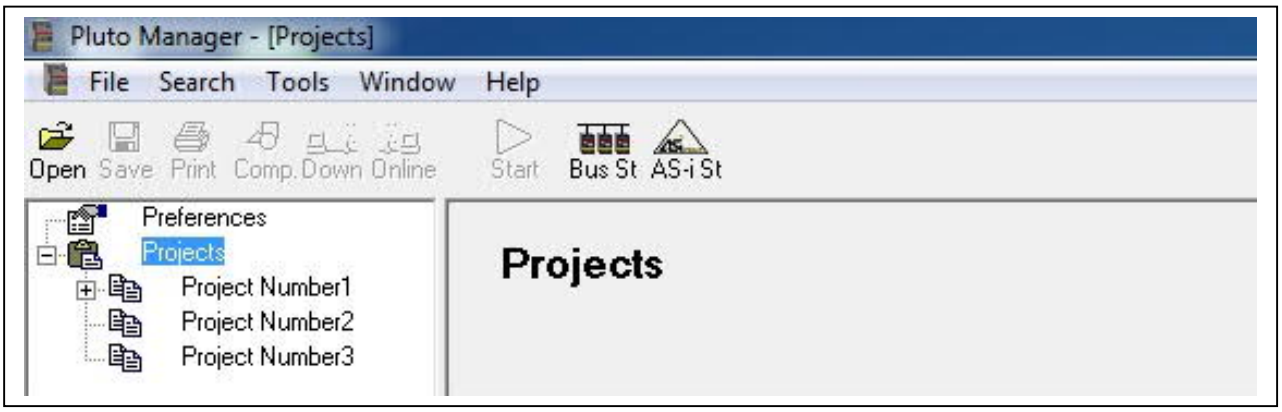


Terminé



6 Projet Ouvrir, Enregistrer, Quitter ….

La fenêtre de Pluto Manager est divisée en deux champs. Le champ de gauche contient un menu arborescent toujours visible qui sert à naviguer entre les différentes fenêtres affichées dans le champ de droite de l'écran. Plusieurs projets peuvent être ouverts en même temps.



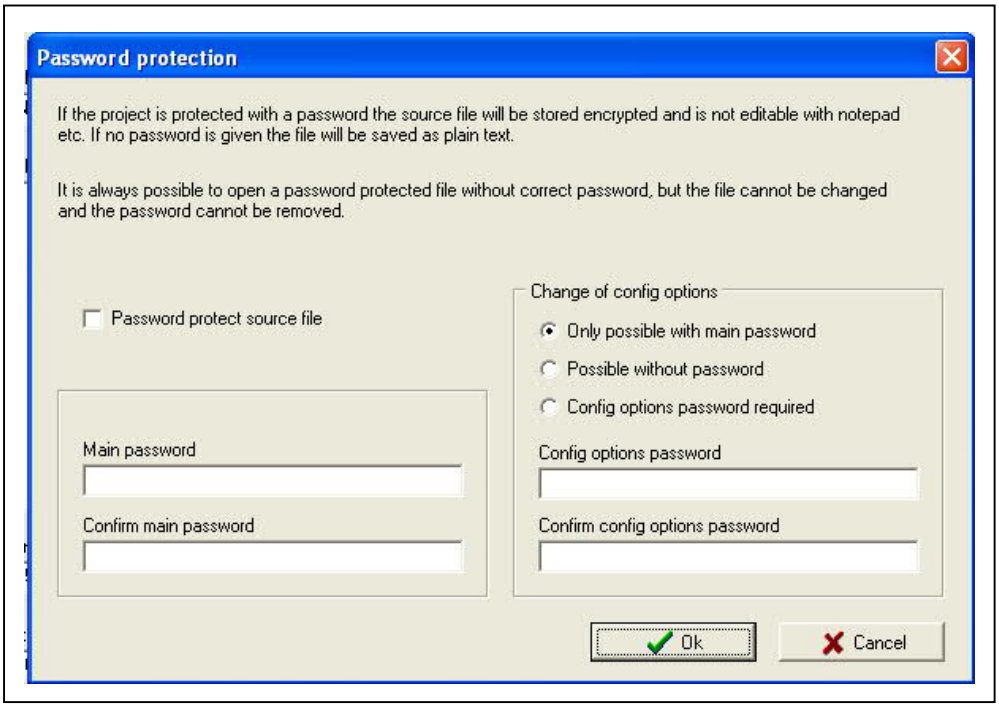
Commandes :

Ouvrir un nouveau projet :	- Cliquer du bouton droit de la souris sur « Projects » dans le menu arborescent et sélectionner « New Project » ou - Ouvrir le menu « File »→ « New »
Ouvrir un projet existant :	- Cliquer du bouton droit de la souris sur « Projects » dans le menu arborescent et sélectionner « Open Project » ou - Utiliser le bouton rapide « Open » dans la liste des outils, ou - Ouvrir le menu « File »→ « Open »
Fermer un projet :	- Cliquer du bouton droit de la souris sur le nom du projet dans le menu arborescent. Sélectionner « Close Project » ou - Marquer l'un des projets ouverts dans le menu arborescent. Ouvrir le menu « File »→ « Close Project».
Enregistrer :	- Cliquer du bouton droit de la souris sur le nom du projet dans le menu arborescent. Sélectionner « Save Project » ou - Marquer dans le menu arborescent l'un des projets ouverts. Utiliser le bouton rapide « Save » dans la liste des outils ou - Marquer dans le menu arborescent l'un des projets ouverts. Ouvrir le menu « File »→ « Save Project »
Tout enregistrer :	- Ouvrir le menu « File »→ « Save All ». Tous les projets ouverts sont sauvegardés.
Protéger par mot de passe :	- Ouvrir le menu « File »→ « Password protect » Voir la description détaillée ci-dessous.

6.1 Protéger par mot de passe

Il est possible de protéger le programme de l'automate avec un mot de passe. Le programme est alors protégé contre les modifications involontaires ou les modifications par une personne non habilitée. Il est toujours possible d'ouvrir un fichier protégé par mot de passe mais le fichier ne peut pas être modifié sans le mot de passe.

Choisir « File »/« Password protect » :



Pour protéger un fichier par mot de passe, cocher « Password protect source file » et choisir un mot de passe principal (Main password). Les différentes options sont indiquées à droite de l'illustration ci-dessus (« Change of config options ») (voir 9.5 Options). Cela signifie que si des options sont utilisées, elles peuvent avoir un autre niveau de protection par mot de passe que le reste du programme.

Only possible with main password:
Les options ont alors le même mot de passe que le reste du programme.

Possible without password:
Les options peuvent alors être modifiées sans mot de passe. Un mot de passe est toutefois nécessaire pour modifier le reste du programme.

Config options password required:
Les options peuvent être modifiées avec un mot de passe spécifique. Le mot de passe principal (Main password) permet toutefois de modifier le programme et les options.

6.1.1 Ouvrir un fichier protégé par mot de passe

La fenêtre suivante s'affiche quand on ouvre un fichier protégé par mot de passe :

Open with full permission:
Le mot de passe principal (Main password) doit être saisi. Il est alors possible de tout modifier dans le programme.

Open with permission to configure:
Un mot de passe spécifique à une option (« Config options password required ») doit être saisi ici. Seules les options peuvent alors être modifiées.
Si « Only possible with main password » a été choisi précédemment, le mot de passe principal doit être saisi (« Main password »). Observer que seules les options peuvent alors être modifiées.
Si l'on a choisi de pouvoir modifier les options sans mot de passe (« Change of config options possible without password »), il n'est pas nécessaire de saisir un mot de passe.

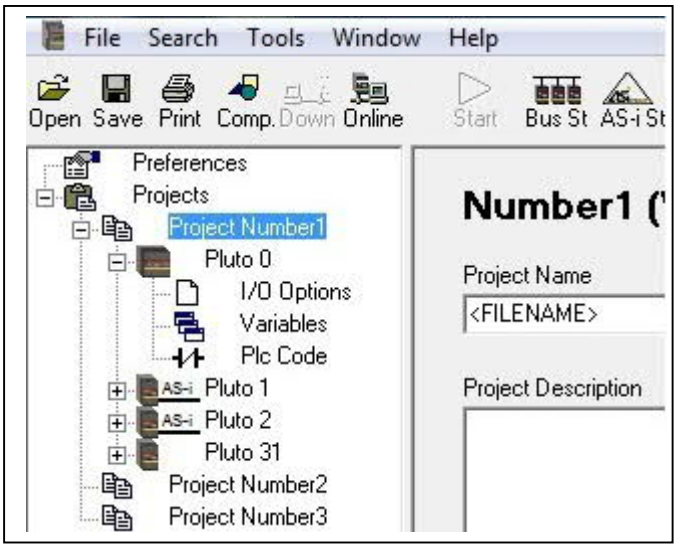
Open in read only mode:
Aucun mot de passe n'est nécessaire et aucune modification n'est possible.

Supprimer la protection par mot de passe
Pour supprimer la protection par mot de passe : ouvrir avec « Open with full permission », choisir « File »/ « Password protect » et décocher « Password protect source file ».

Cliquer sur OK.

7 Configuration du bus

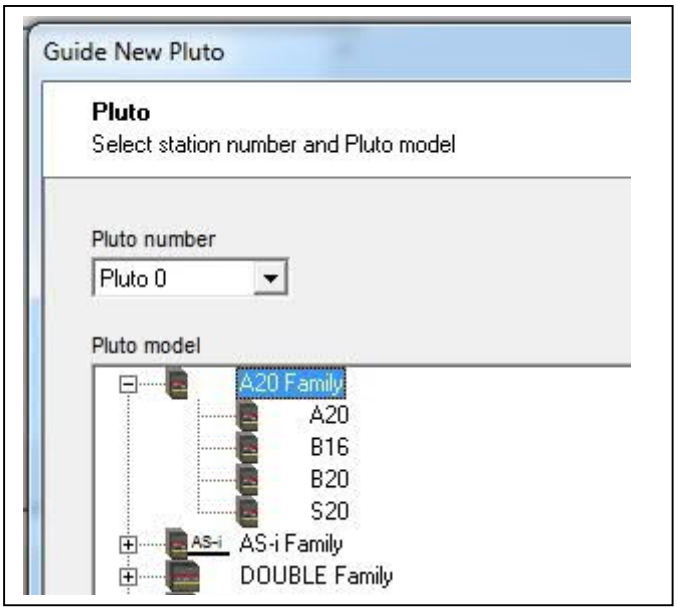
Les Pluto peuvent fonctionner comme des entités séparées ou être interconnectés via le bus Pluto. Un projet peut comporter de 1 à 32 Pluto. Les programmes de tous les Pluto sont enregistrés dans un fichier .sps commun chargé dans chaque automate.



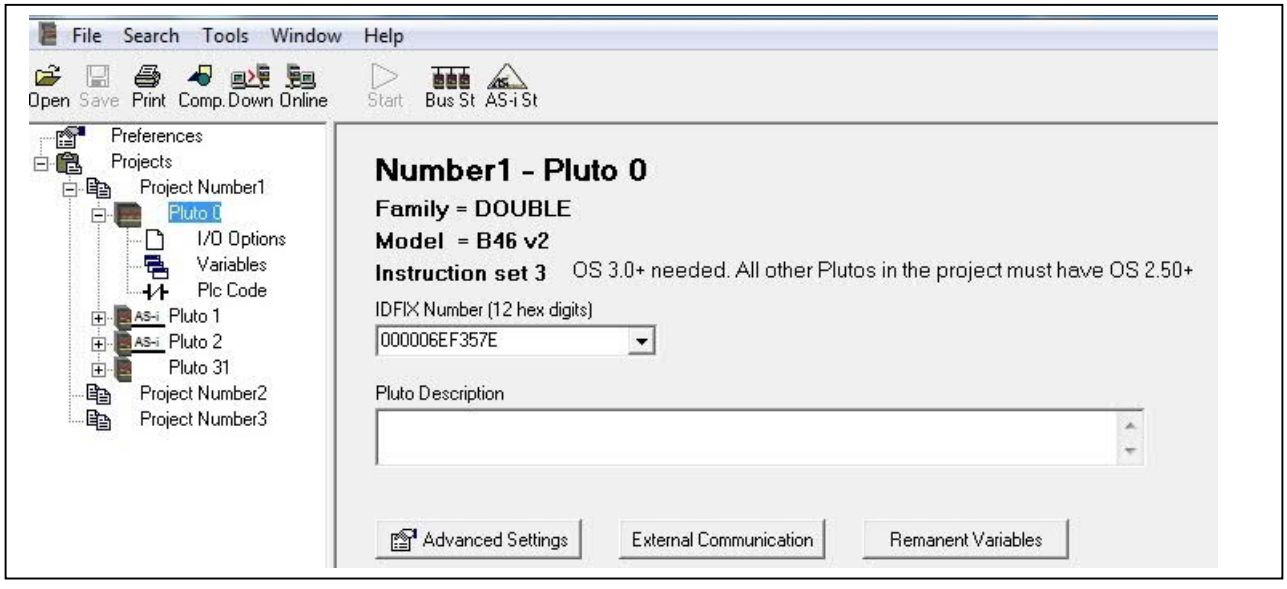
Commandes :
Cliquer du bouton droit de la souris sur « Project [*name*] » → « New Pluto » dans l'arborescence.

Choisir le type de Pluto (« Pluto Family »)
Entrer un numéro de Pluto entre 0 et 31.

Le numéro de Pluto est indiqué dans les adresses des E/S. Les entrées de Pluto 0 sont désignées comme : I0.0, I0.1, I0.2,… et celles deu Pluto 1 comme : I1.0, I1.1, I1.2,… etc.



Le menu suivant s'affiche quand on clique sur un des Pluto du projet, le Pluto 0 par exemple.

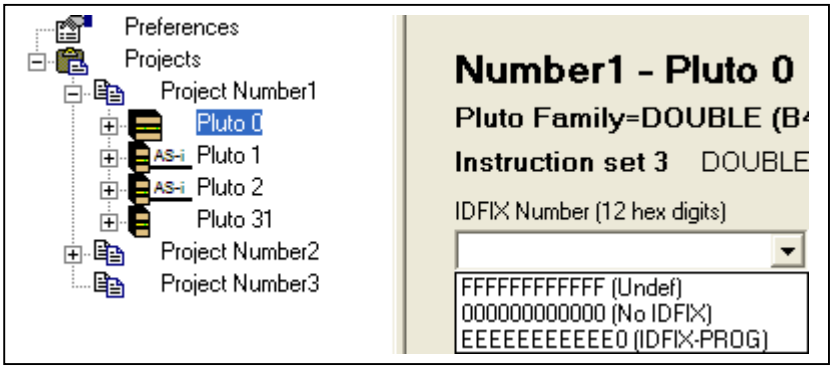


7.1 Numéro de l'identifiant « IDFIX number »

Lorsque plusieurs Pluto sont connectés au bus, chacun doit être doté d'un identifiant externe contenant un nombre hexadécimal unique. (Voir le Manuel d'utilisation Matériel.)

Le numéro d'identifiant doit être saisi dans le champ « IDFIX Number ». Comme le numéro est encore inconnu à ce stade, il peut être laissé de côté jusqu'au moment du transfert et du test du système.

Si le projet ne contient qu'un Pluto sans IDFIX, choisir « No IDFIX » dans la liste déroulante.

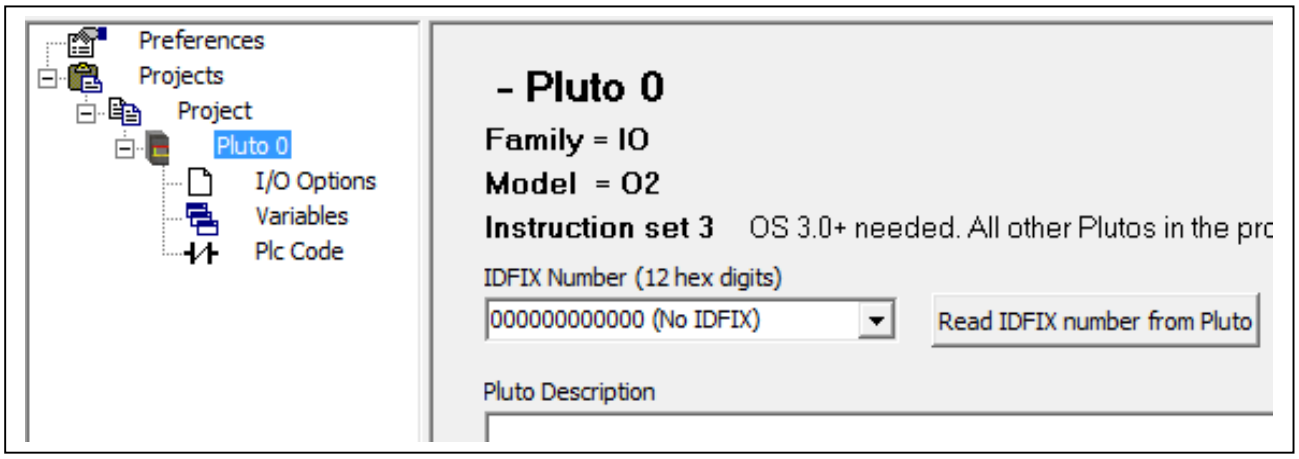


Si « IDFIX-PROG » est utilisé (voir le Manuel d'utilisation Matériel), choisir "IDFIX-PROG" dans la liste déroulante.

« Pluto description »
Ce champ ne sert qu'aux commentaires et aux descriptions. Il n'est pas chargé dans le Pluto.

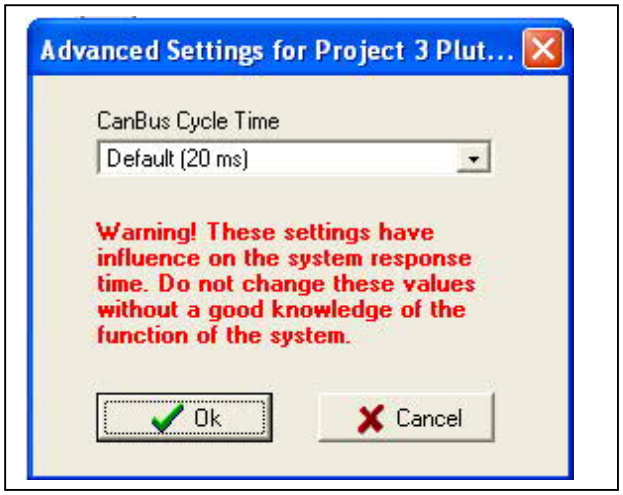
7.1.1 Read IDFIX number from Pluto (Lire le numéro IDFIX sur Pluto)

Avec Pluto Manager 2.20 ou plus récent et Pluto OS 3.4 ou plus récent, il est possible de lire le numéro IDFIX de Pluto en cliquant sur le bouton « Lire le numéro IDFIX sur Pluto » dans Pluto Manager.



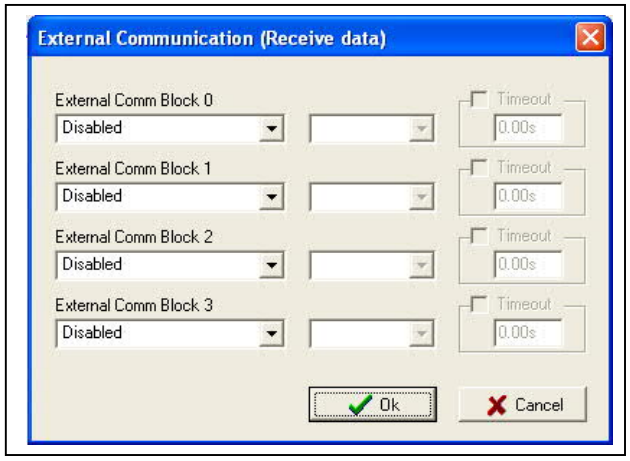
7.2 « Advanced settings » (Paramètres avancés)

Le bouton « Advanced settings » donne accès au temps de cycle du bus Pluto (« CanBus Cycle Time »). Ce réglage est décrit dans le Manuel d'utilisation Matériel. Il influe sur le temps de réponse du système et ne doit être modifié que par des personnes ayant une bonne connaissance du fonctionnement du système.



7.3 « External communication » (Communication externe)

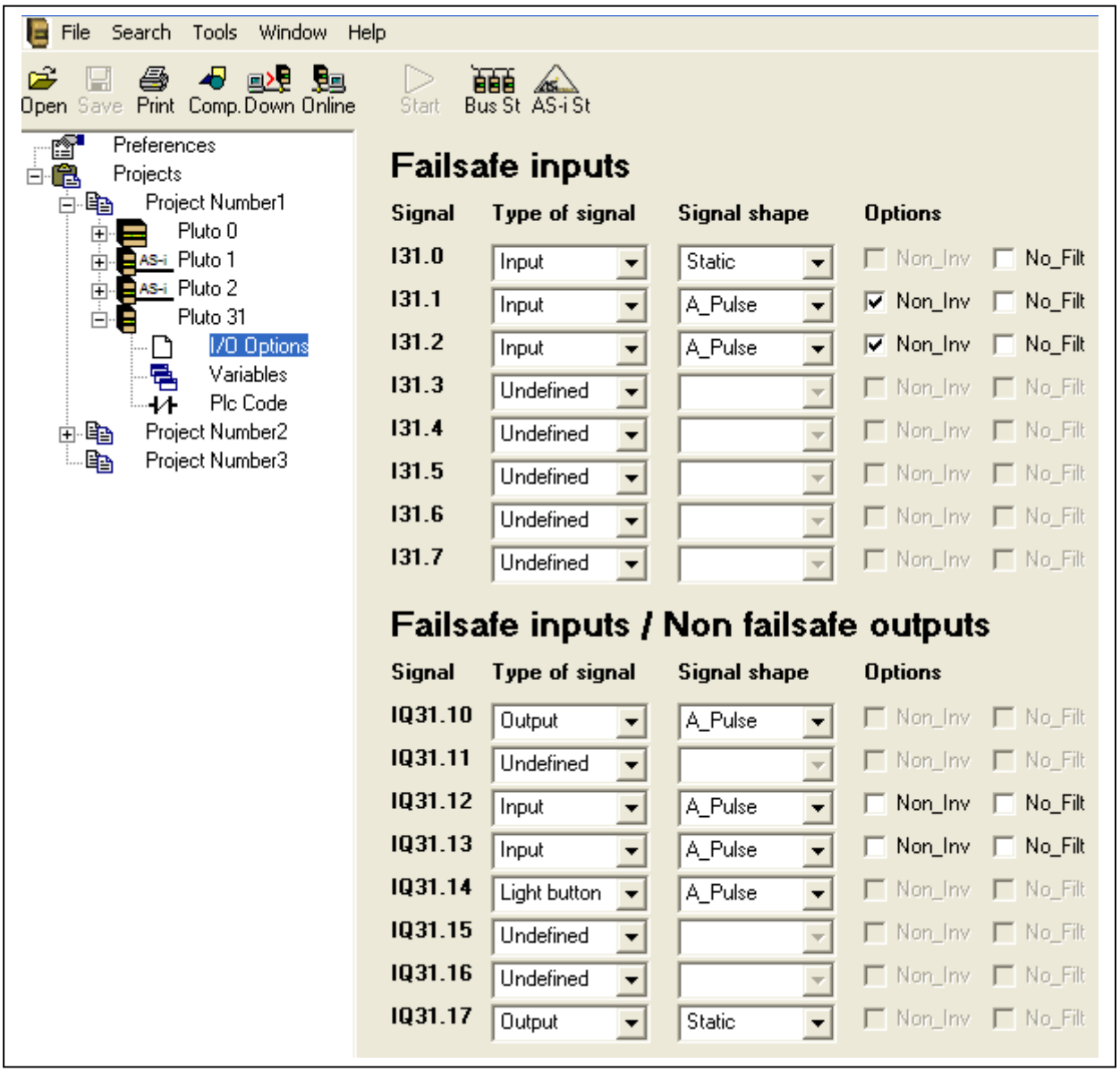
Le bouton « External Communication » ouvre la fenêtre suivante. Cette fonction est utilisée quand Pluto doit recevoir des données d'une passerelle connectée au bus Pluto. Elle est décrite dans le manuel des passerelles pour Pluto.



8 Options des E/S

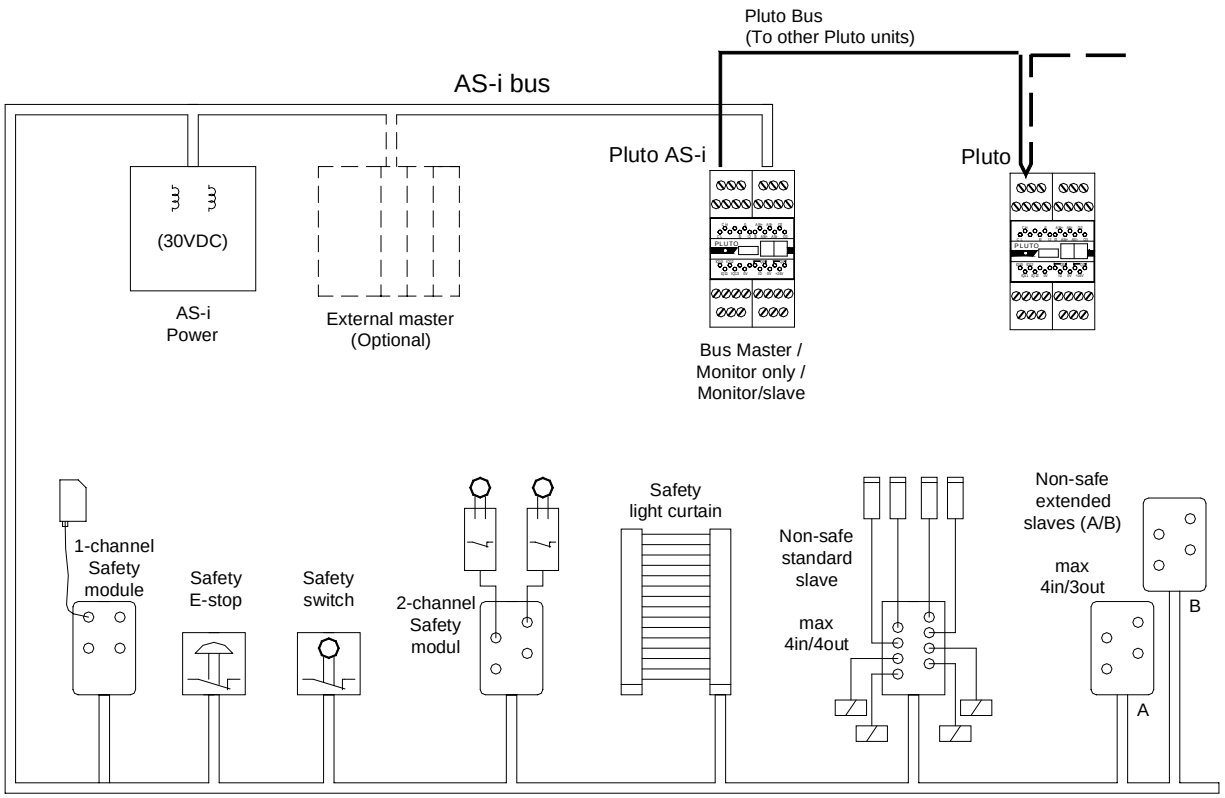
La page « I/O Options » s'affiche en cliquant sur l'icône correspondante dans le menu arborescent. Les paramètres sont sélectionnés à l'aide de menus déroulants et de cases à cocher. Les combinaisons interdites sont automatiquement bloquées.

La page « I/O Options » est la même pour les différents modèles de Pluto. Seul le nombre d'E/S diffère.



9 Fonctions du bus AS-i

(Pluto AS-i et B42 AS-i seulement. Voir aussi le Manuel d'utilisation Matériel de Pluto)



Pluto avec bus AS-i et exemples de types d'esclaves AS-i.

(Remarque : pour les Pluto AS-i version 1, les esclaves A/B ne peuvent être gérés qu'en mode Monitor.)

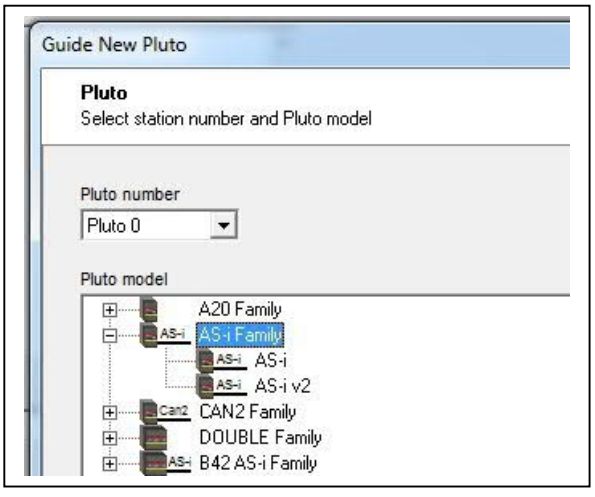
9.1 Configuration initiale des fonctions AS-i

La configuration d'un Pluto AS-i est décrite pas à pas ci-dessous.

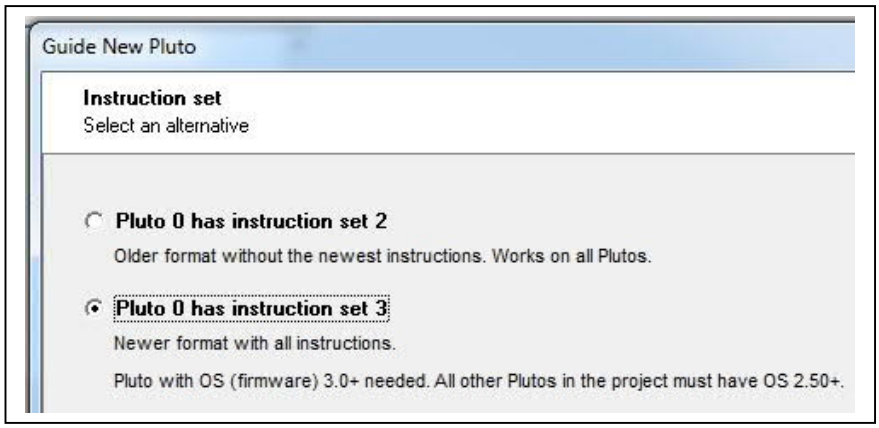
9.1.1 « New Pluto » (Nouveau Pluto), choix de la famille et du numéro de Pluto

Placer le pointeur sur le menu arborescent de gauche. Cliquer du bouton droit de la souris et choisir « New Pluto » (comme décrit au paragraphe 5).

Choisir « Pluto AS-i » ou « Pluto B42 AS-i » dans la liste et choisir le numéro du Pluto sur le bus Pluto.

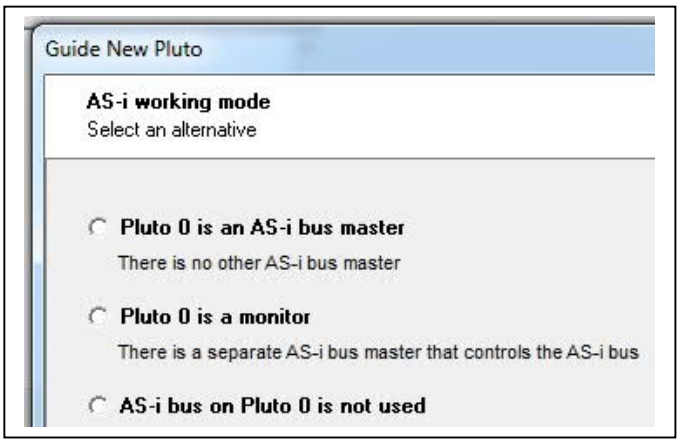


Dans le cas de AS-i v2 ou B42 AS-i, il vous est demandé de choisir le jeu d'instructions, « instruction set 2 » ou « instruction set 3 ». (Décrit au paragraphe 3.6.1 et dans la deuxième partie de ce manuel.)



9.1.2 Modes de fonctionnement du bus AS-i

Dès qu'un Pluto AS-i a été choisi, on vous demande quel est le mode de fonctionnement souhaité (AS-i Working mode).



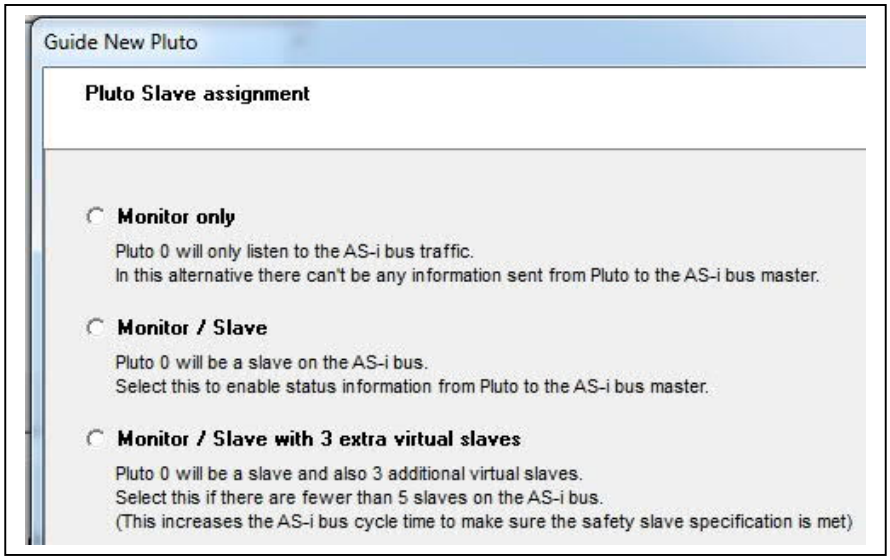
Pluto is an AS-i bus master (Master mode) doit être choisi si aucun autre maître n'est connecté au bus AS-i. Pluto possède alors le contrôle total du bus. Pour l'utilisateur, la différence principale est que Pluto peut commander les sorties des esclaves non sécurisés.

Pluto is a monitor (Monitor/slave mode) doit être choisi s'il existe un autre maître que le Pluto sur le bus AS-i. Généralement, cet autre maître est un PLC standard non sécurisé qui commande les dispositifs non sécurisés sur le bus AS-i. Le Pluto ne fait alors que lire les esclaves AS-i- Un Pluto utilisé comme moniteur peut non seulement lire toutes les E/S à destination et en provenance des esclaves de sécurité mais aussi les entrées et sorties des esclaves non sécurisés.

AS-i bus on Pluto is not used doit être choisi si la fonctionnalité AS-i/le bus AS-i n'est pas utilisé.

9.1.2.1 Variantes du « monitor mode » :

Si le « monitor mode » est choisi, trois possibilités sont proposées dans une nouvelle boîte de dialogue.



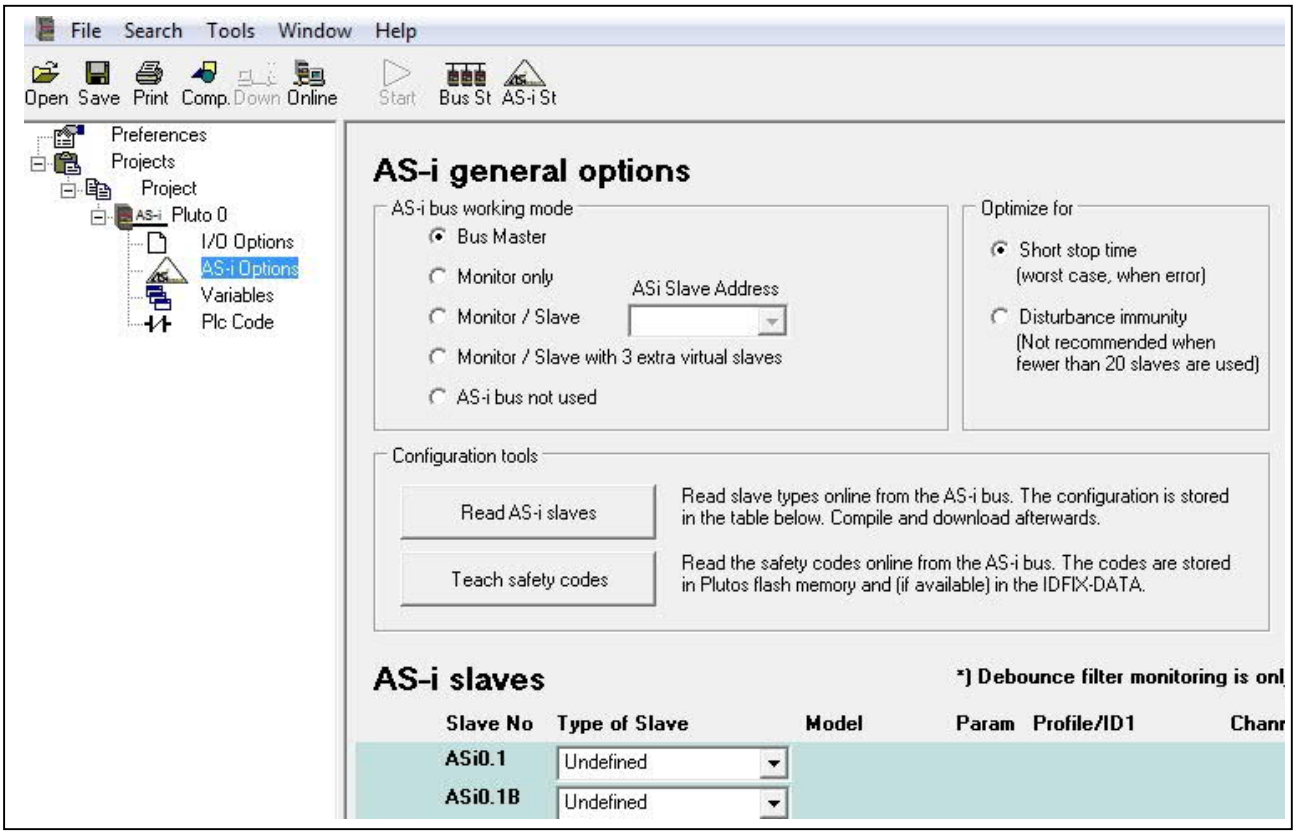
Monitor only : Un autre maître commande le bus. Pluto écoute la communication sur le bus et lit les E/S en provenance et à destination de tous les esclaves. (Aussi bien les entrées de sécurité que les entrées/sorties non sécurisées).

Monitor/Slave : Pluto fonctionne alors non seulement comme un moniteur (mode « Monitor only ») mais aussi comme un esclave non sécurisé du maître. Pluto et le maître peuvent communiquer l'un avec l'autre, 4 bits dans chaque direction. Dans ce mode de fonctionnement, les adresses des esclaves doivent aussi être choisies.

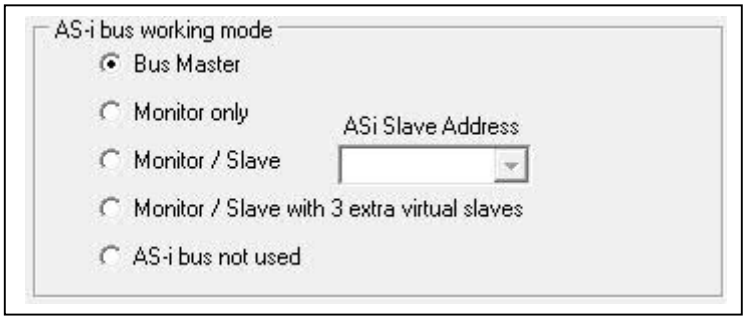
Monitor/Slave with 3 extra virtual slaves : Comme « Monitor/Slave » avec trois esclaves virtuels en plus. Ce mode de fonctionnement doit être choisi quand moins de 5 esclaves AS-i sont connectés au bus : le temps de cycle du bus AS-i est très court si peu d'esclaves AS-i sont connectés au bus et si le temps de cycle est trop court, les esclaves de sécurité n'ont pas le temps de mettre à jour leur code de sécurité.

9.1.3 Page des paramètres spécifiques à AS-i

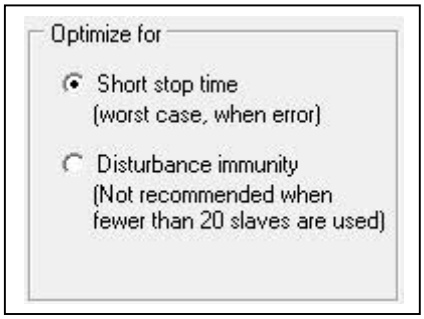
Cliquer sur « AS-i Options » dans le menu arborescent de gauche pour afficher la page des paramètres spécifiques à AS-i.



Mode de fonctionnement du bus AS-i
Bien que le mode de fonctionnement soit choisi immédiatement après le choix du Pluto AS-i, il peut être modifié ultérieurement. Comme l'illustration le montre, il existe trois possibilités pour le mode Moniteur (Monitor Mode).



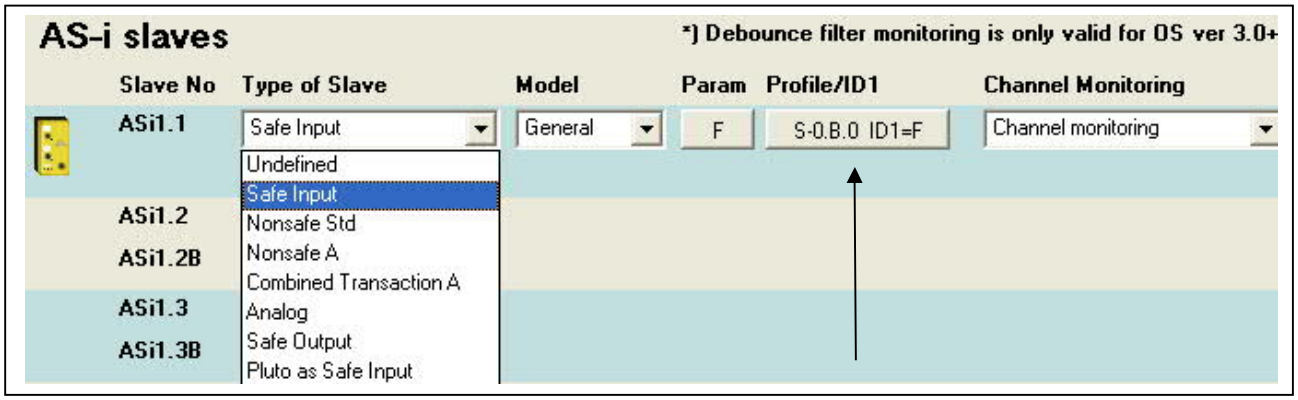
Optimisation pour un temps d'arrêt court ou une meilleure résistance aux perturbations
Comme représenté sur l'illustration, un temps d'arrêt court (Short stop time) doit être choisi quand moins de 20 esclaves sont connectés au bus.
Si l'on choisit la résistance aux perturbations (Disturbance immunity), le système est moins sensible aux perturbations sur le bus AS-i, mais le temps d'arrêt peut augmenter de jusqu'à 10 ms.



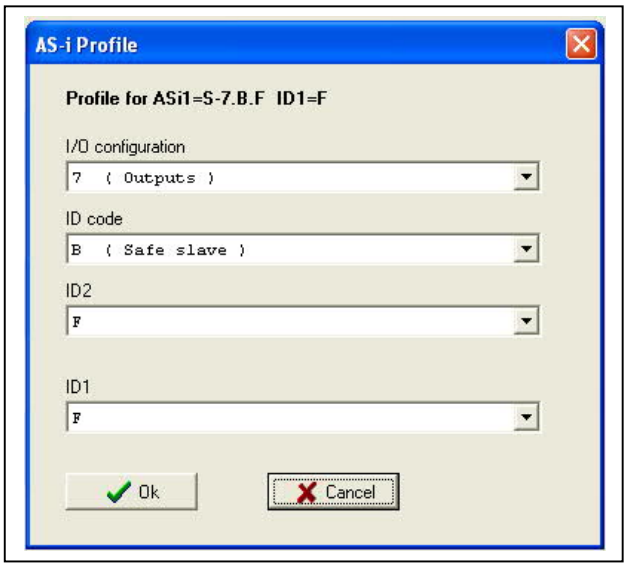
9.1.4 Configuration manuelle des types d'esclaves (profils)

La configuration semi-automatique décrite au chapitre 7.2 nécessite une communication En ligne avec le Pluto AS-i. Comme l'automate est souvent programmé sans être En ligne, la configuration peut aussi être effectuée manuellement. Le programmeur n'est pas obligé de remplir le tableau en cas programmation hors ligne. Un message est alors affiché lors de la compilation indiquant que les esclaves ne sont pas configurés.

Jusqu'à 31 esclaves (ou 62 esclaves A/B) peuvent être connectés au bus AS-i et ils peuvent être configurés manuellement dans Pluto Manager sous « AS-i Options/Type of Slave » pour chaque numéro d'esclave. Comme illustré, il existe 8 choix possibles.



Sauf dans le cas de « Undefined », un bouton se trouve sous « Profile/ID1 ». Cliquer sur ce bouton pour afficher une fenêtre permettant de saisir le profil de l'esclave. (Seulement si Pluto est utilisé comme maître (Bus Master). Si Pluto est utilisé comme moniteur, voir « 7.2.11 Configuration en mode Moniteur ».)



Vous trouverez ci-dessous une description des différents types d'esclaves et un tableau contenant les noms des variables d'entrée et de sortie pour chaque type d'esclave.

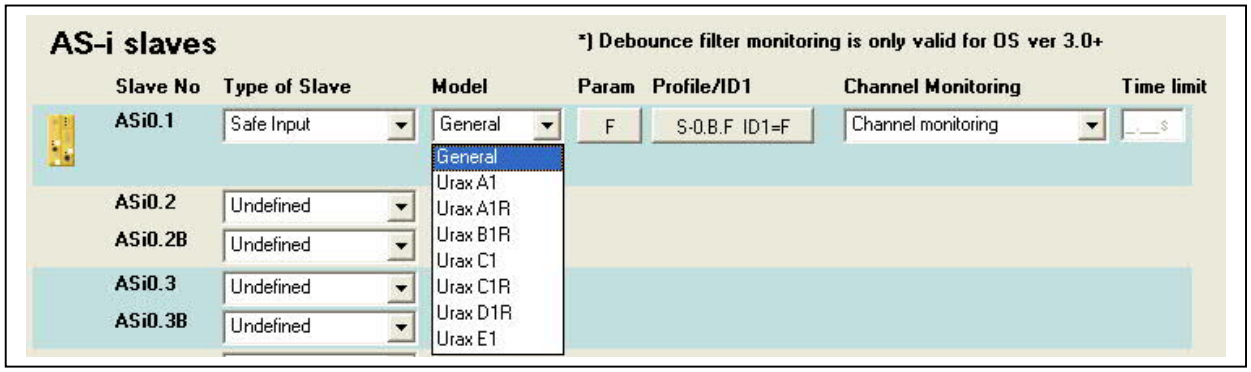
9.1.4.1 Undefined

Choisir Undefined/Indéfini quand aucun esclave ne doit être connecté à cette adresse.

9.1.4.2 Safe input

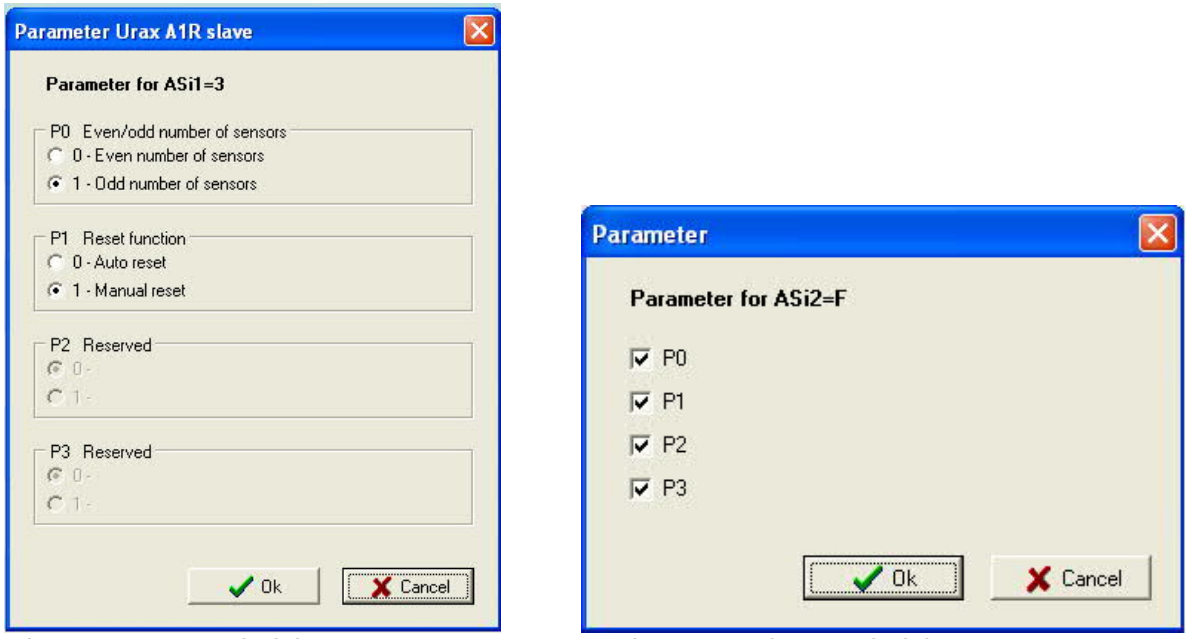
Un esclave entrée de sécurité dispose physiquement d'une entrée deux canaux. Il est toutefois configuré comme une seule entrée dans Pluto/Pluto Manager. L'esclave peut avoir jusqu'à 4 sorties non sécurisées. Pour les noms des variables, voir le tableau dans 7.1.4.4 Esclaves A/B non sécurisés.

Si « Safe Input » (entrée de sécurité) est choisi, la page suivante s'affiche :



Le type d'esclave d'entrée de sécurité est choisi dans la liste déroulante sous « Model ». Choisir « General » pour tous les types d'esclaves n'étant pas des Urax. Pour les Urax, choisir le modèle utilisé.

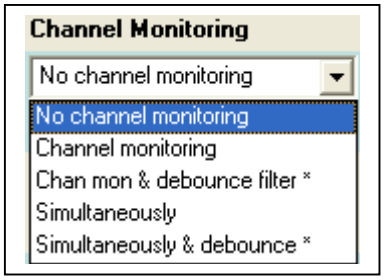
Cliquer sur « **Param** » pour définir le paramètre de l'esclave. Ce paramètre détermine le mode de fonctionnement de l'esclave.



Si Urax-A1R est choisi. *Si « General » est choisi.*

« Profile/ID1 » décrit l'esclave et indique son nombre d'entrées/sorties, s'il s'agit d'un esclave de sécurité, d'un esclave A/B, etc. Les codes des profils sont décrits plus en détail dans différents documents sur l'AS-i. Voici quelques exemples :
S-0.B… - Esclave de sécurité
S-7.B… - Esclave de sécurité avec sorties
S-7.0… - Esclave son sécurisé standard avec 4 entrées et 4 sorties.
Pour les Urax, le « Profile/ID1 » est choisi automatiquement quand le type d'Urax utilisé est choisi. Pour les autres types d'esclaves, voir leur manuel pour choisir le bon profil.

« Channel Monitoring »
Les nœuds de sécurité ont souvent des entrées deux canaux. Dans ce cas, l'utilisateur peut choisir entre différents types de contrôle des canaux.

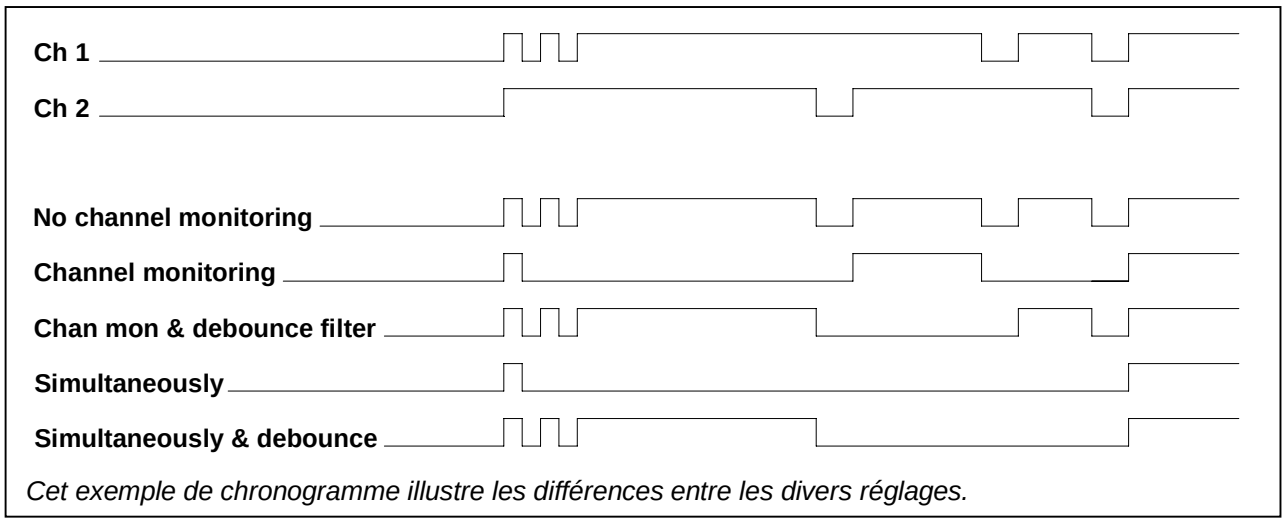


- Aucun contrôle (No channel monitoring) :
  Les deux canaux doivent être actifs, mais aucun contrôle n'est effectué entre les deux canaux. Configuration normale pour les esclaves à un seul canal.
- Contrôle canaux (Channel Monitoring)
  Paramètre par défaut. Si un canal est désactivé, l'autre doit aussi est désactivé puis les deux activés.
- Contrôle canaux & filtre rebonds (Chan mon & debounce filter)*:
  Comme pour le contrôle canaux + une fois les deux contacts activés, ils peuvent rebondir pendant un certain temps sans que le nœud déclenche. Le contrôle de discordance des deux canaux est désactivé durant le « temps de rebond » mais les rebonds ne sont pas filtrés. Ce mode convient aux portes dotées d'interrupteurs mécaniques par exemple.
- Simultanément (Simultaneously) :
  Comme pour le contrôle canaux avec un délai maximum entre l'activation de chaque canal.
- Simultanément & rebonds (Simultaneously & debounce)* :
  Comme simultanément, mais les contacts peuvent rebondir durant le temps indiqué. Le contrôle de discordance des deux canaux est désactivé durant le « temps de rebond » mais les rebonds ne sont pas filtrés.

*OS version 3.0 ou plus

Le Contrôle canaux n'est pas disponible pour les URAX (sauf URAX-C1), car le contrôle des deux canaux est effectué par l'URAX lui-même (sauf URAX-C1).

« Time limit »
Si « Simultaneously » est choisi, le délai max. souhaité en secondes est saisi là.



Cet exemple de chronogramme illustre les différences entre les divers réglages.

9.1.4.3 Nonsafe Standard slaves

Un esclave standard non sécurisé peut avoir jusqu'à 4 entrées non sécurisées locales et/ou jusqu'á 4 sorties non sécurisées locales. Pour les noms des variables, voir le tableau ci-dessous.

9.1.4.4 Nonsafe A/B slaves

Deux esclaves, un esclave A et un esclave B, partagent la même adresse. Cela signifie que jusqu'à 62 esclaves A/B peuvent être connectés à un réseau au lieu de 31 qui est le maximum pour les autres types d'esclaves. Un esclave A/B non sécurisé peut avoir jusqu'à 4 entrées et/ou 3 sorties. Dans Pluto, ces entrées et sorties sont locales. Pour les noms des variables, voir le tableau ci-dessous.

	« Type of Slave » dans le menu			
	« Safe input » (Esclaves 1-15)	**« Safe input » (Esclaves 16-31)**	**« Nonsafe Std »**	**« Nonsafe A » (« Nonsafe B »)**
Entrées de sécurité globales	ASi_.x	-	-	-
Entrées de sécurité locales	-	ASi_.x	-	-
Entrées non sécurisées locales	-	-	ASi_.x.1 ASi_.x.2 ASi_.x.3 ASi_.x.4	ASi_.x.1 ASi_.x.2 ASi_.x.3 ASi_.x.4 (ASi_.xB.1) (ASi_.xB.2) (ASi_.xB.3) (ASi_.xB.4)
Sorties non sécurisées locales	ASq_.x.1 ASq_.x.2 ASq_.x.3 ASq_.x.4	ASq_.x.1 ASq_.x.2 ASq_.x.3 ASq_.x.4	ASq_.x.1 ASq_.x.2 ASq_.x.3 ASq_.x.4	ASq_.x.1 ASq_.x.2 ASq_.x.3 (ASq_.xB.1) (ASq_.xB.2) (ASq_.xB.3)

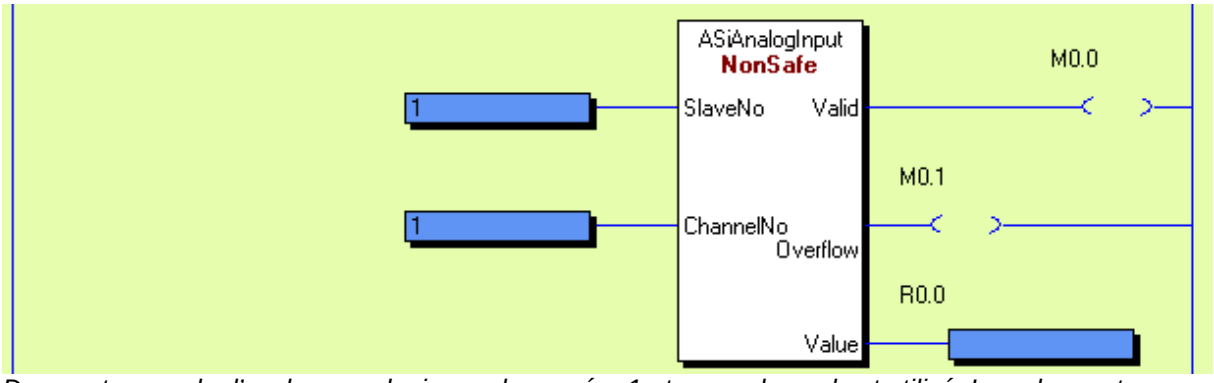
« _ » = n° de Pluto, « x » = n° d'esclave.

9.1.4.5 Combined Transaction A/B slaves

Pluto est compatible avec les esclaves de type « Combined Transaction » ayant 4 entrées et 4 sorties.
Profil AS-I : S-7.A.7

9.1.4.6 Analogue input slaves

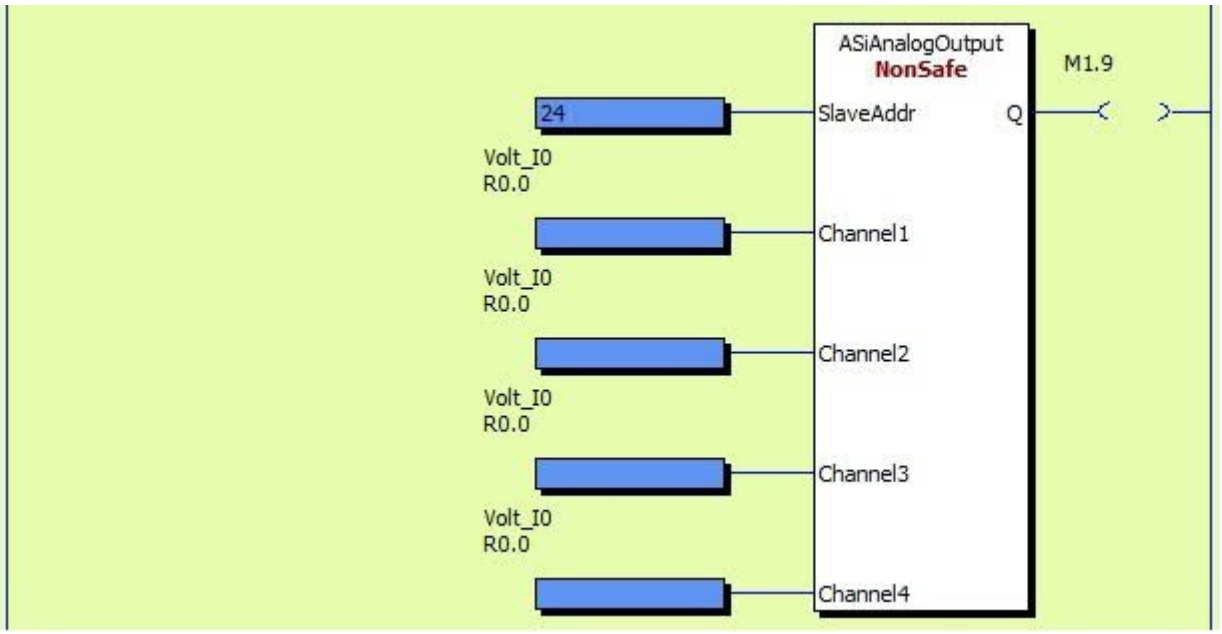
Il s'agit d'une entrée analogique non sécurisée qui lit une valeur d'entrée analogique par canal et envoie une représentation numérique de cette valeur sur le bus AS-i. L'esclave peut avoir jusqu'à 4 canaux d'entrée et nécessite un bloc de fonction spécial (ASiAnalogInput) par canal.



Dans cet exemple, l'esclave analogique a le numéro 1 et un seul canal est utilisé. La valeur est enregistrée dans le registre 0.

9.1.4.7 Analogue output slaves (non-safe)

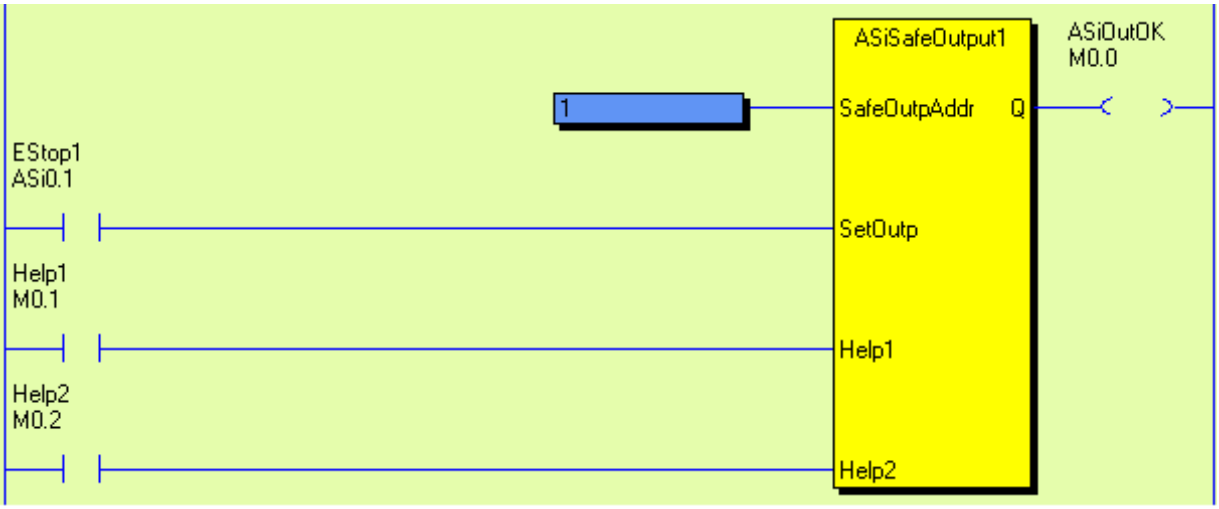
Esclave de type non de sécurité avec des sorties analogiques, normalement 4-20 mA ou 0-10 V. L'esclave peut avoir jusqu'à 4 canaux de sortie. Les sorties analogiques sont commandées avec le bloc de fonction « ASiAnalogOutput ». Un registre par canal est connecté au bloc pour la définition des valeurs de sortie.



*Le bloc de fonction **ASiAnalogOutput** doit être utilisé pour les esclaves de sortie analogique. Dans cet exemple avec une sortie de sortie analogique à l'adresse 24, les quatre sorties sont définies sur la valeur dans le registre R0.0.*

9.1.4.8 Safe Output

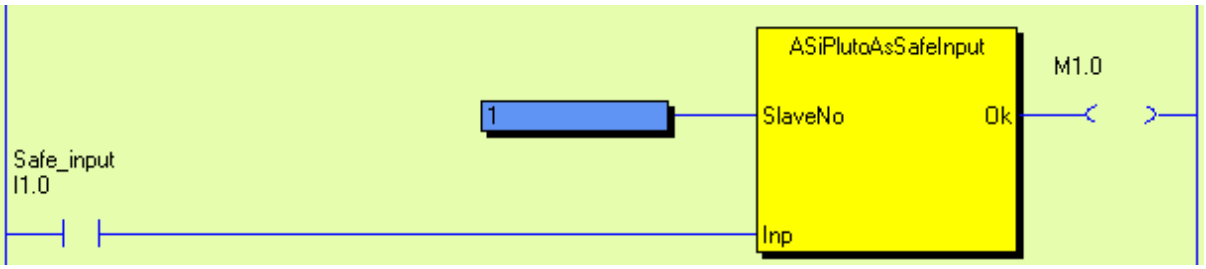
Esclave de sécurité à (à ce jour) une seule sortie de sécurité. Un bloc de fonction spécial « ASiSafeOutput1 » est nécessaire dans le programme de l'automate. Cet esclave est généralement associé à un esclave non de sécurité pour une information d'état. Même si cet esclave non de sécurité se trouve dans le même boîtier que l'esclave sortie de sécurité, ils ont des adresses différentes et Pluto les considère comme deux esclaves séparés. Pluto peut commander jusqu'à 16 esclaves « PlutoAsSafeInput » + « SafeOutput ».



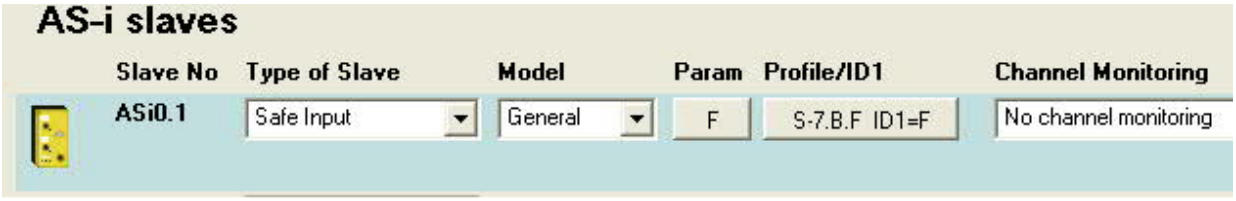
*Le bloc de fonction **ASiSafeOutput1** doit être utilisé pour les esclaves avec une sortie de sécurité. SafeOutpAddr est l'adresse de l'esclave, SetOutp définit la sortie de sécurité et Help1 et Help2 sont des signaux d'aide. Le mode d'utilisation des signaux d'aide varie en fonction du fabricant d'esclaves.*

9.1.4.9 Pluto as Safe Input

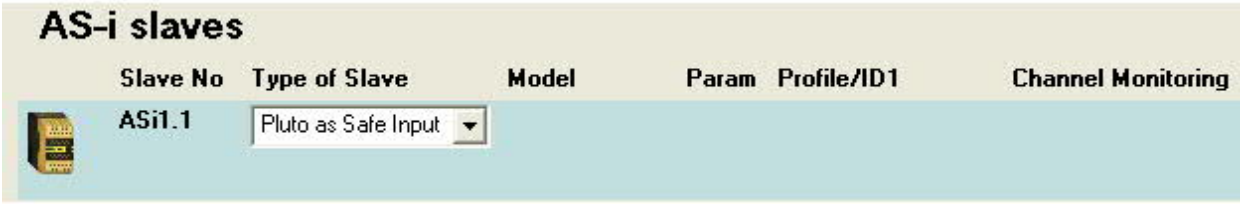
Pour un Pluto utilisé comme esclave d'entrée de sécurité. Le bloc de fonction « PlutoAsSafeInput » est utilisé dans le programme de l'automate. L'entrée de sécurité et les sorties non sécurisées sont configurées comme un esclave d'entrée de sécurité (esclave « Safe input ») dans le tableau ci-dessus. Pluto peut gérer jusqu'à 16 esclaves « PlutoAsSafeInput » + « SafeOutput ».



Exemple : le Pluto 1 est utilisé comme esclave d'entrée de sécurité (safe input slave). Le numéro d'esclave est 1 et I1.0 est utilisée comme entrée du bloc de fonction.



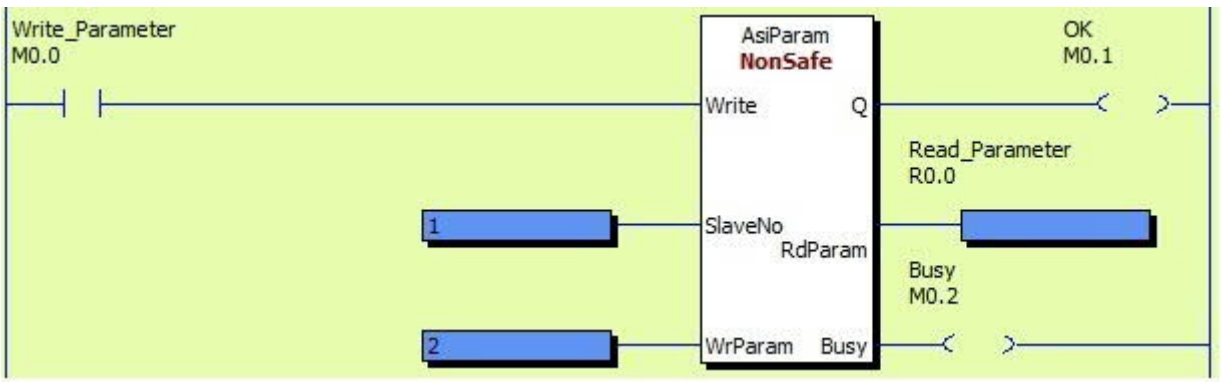
Configuration du Pluto 0 qui est le maître qui lit l'esclave n° 1.



Configuration du Pluto 1 qui fonctionne comme esclave d'entrée de sécurité (esclave « Safe input »).

9.1.5 Écrire un paramètre à l'esclave et recevoir des informations en retour

Pour certains esclaves AS-i du marché, il est possible d'envoyer un paramètre à l'esclave et de recevoir des informations/données en retour. Le bloc de fonction « ASiParam » est requis ici.

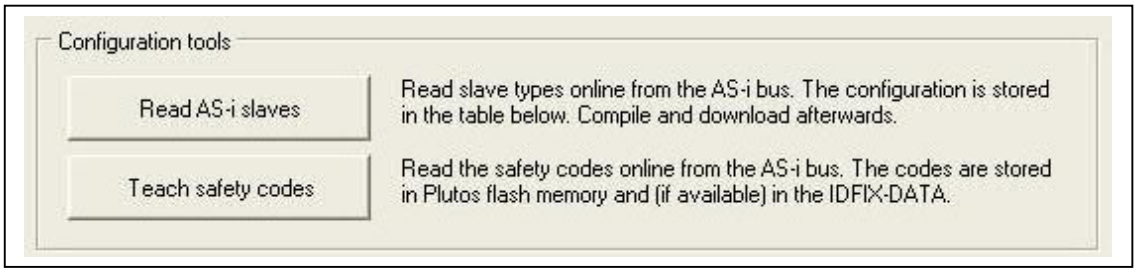


*Exemple avec **AsiParam** : le paramètre 2 est écrit à l'esclave 1. Les données reçues en retour de cet esclave seront saisies dans le registre R0.0.*

9.2 Configuration en ligne du bus AS-i

Pour pouvoir effectuer la configuration décrite ci-dessous, le programme doit être compilé et transféré dans le Pluto.

Les boutons « Read AS-i slaves » et « Teach safety codes » sont des fonctions semi-automatiques qui permettent de lire les types des esclaves connectés au bus AS-i et d'apprendre leurs codes de sécurité.

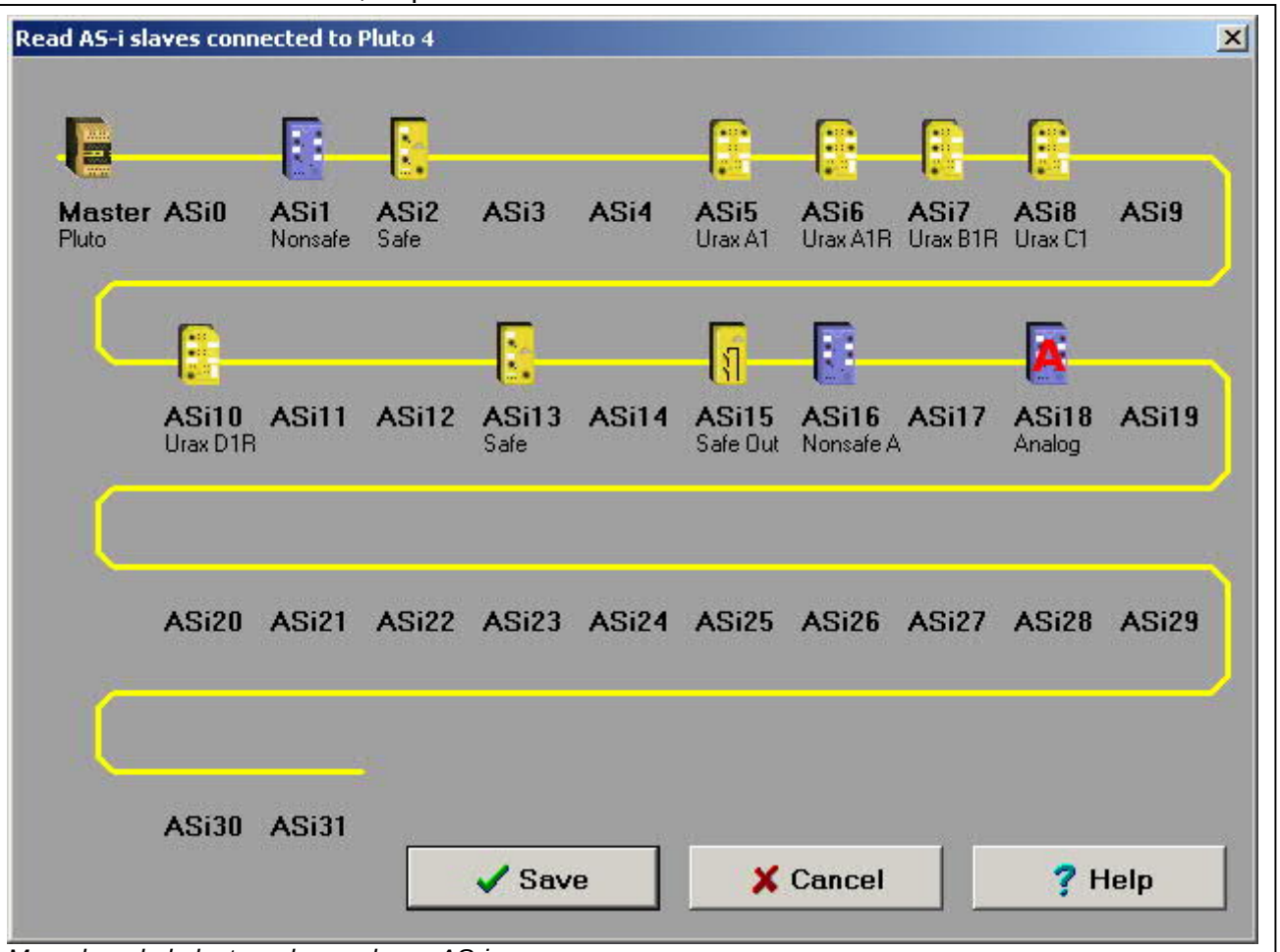


9.2.1 Lire les esclaves AS-i

- Cliquer sur « Read AS-i slaves ».

Pluto se met à balayer le bus AS-i pour connaître les types des esclaves connectés. L'illustration ci-dessous est affichée.

- Si tout semble correct, cliquer sur « Save ».



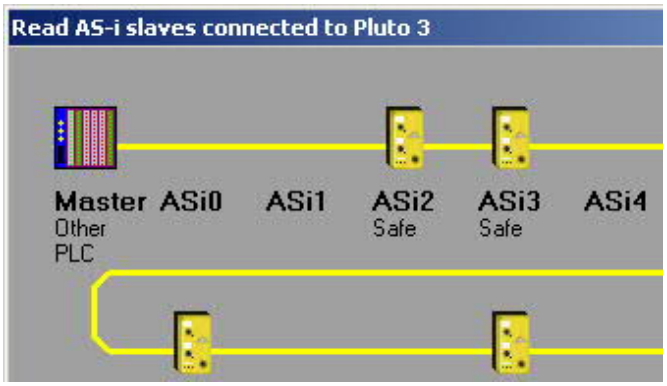
Menu lors de la lecture des esclaves AS-i.
Ici, Pluto est maître sur le bus. Les N° 1 et 16 sont des esclaves non sécurisés standards, 2 et 13 sont des esclaves entrées de sécurité, 5 à 8 et 10 sont des Urax (entrées de sécurité), 15 est un esclave sortie de sécurité et 18 et un esclave entrée analogique.

Save
Cliquer sur « Save » pour enregistrer les profils des esclaves (les types des esclaves) dans le tableau contenu dans le programme de l'automate. Remarque : le profil entier est lu et écrit dans le tableau en mode maître uniquement.

Enregistrer la configuration des esclaves.
La liste est enregistrée dans le programme de l'automate, ce qui signifie que la configuration doit être compilée et transférée dans le Pluto.

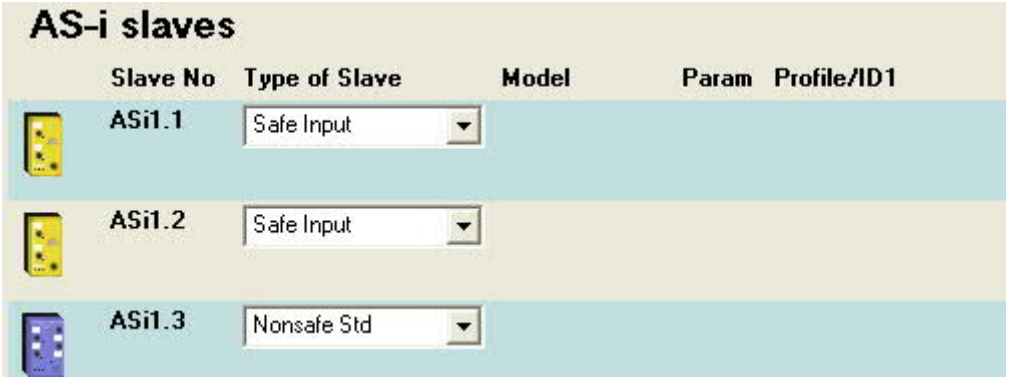
9.2.1.1 Configuration en Monitor mode

Si Pluto est configuré comme Moniteur, la procédure de configuration est pratiquement la même, à quelques différences près.



La fenêtre de l'état du bus indique la présence d'un maître externe.

La principale différence est que tous les paramètres des esclaves ne sont pas affichés en mode Monitor. Il est seulement affiché quels sont les esclaves de sécurité et les esclaves non sécurisés.



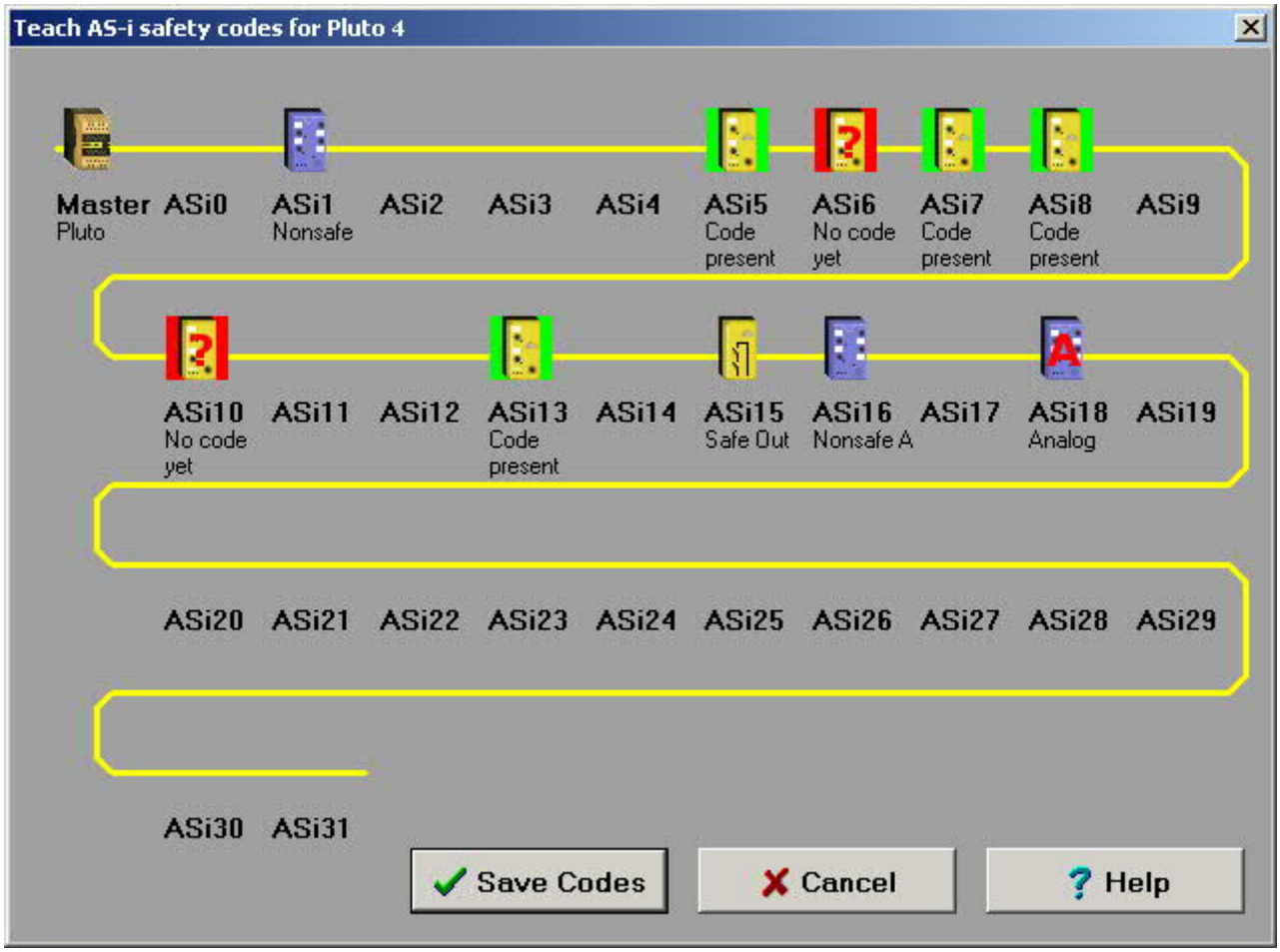
Liste des types des esclaves et des paramètres de sécurité des esclaves de sécurité en mode moniteur.

9.2.2 Apprentissage des codes de sécurité

L'apprentissage des codes de sécurité ressemble à la lecture des profils des esclaves. L'apprentissage est effectué au démarrage du système. Les codes de sécurité sont enregistrés dans le programme de l'automate et le programmeur n'en a pas besoin lors de la programmation.

- Cliquer sur le bouton

Une illustration du bus est affichée. Un capteur de sécurité doit être activé pour que « Code present » soit affiché. Il suffit que chaque capteur soit activé une fois au cours de l'apprentissage.



Menu au cours de l'apprentissage des codes de sécurité.
Les esclaves n° 6 et 10 n'ont présenté aucun code. Ils ne sont probablement pas activés.

- Une fois tous les codes de sécurité obtenus, cliquer sur « Save Codes ».

Quand Pluto enregistre les codes, le fonctionnement normal est interrompu. Un code d'erreur (souvent Er71) est alors affiché et après 5 secondes, Pluto redémarre automatiquement.

Les codes sont enregistrés dans la mémoire interne de Pluto et dans l'IDFIX-DATA / IDFIX-PROG s'il est connecté. (Au redémarrage ou en cas de conflit, ce sont les codes dans l'IDFIX qui sont utilisés. Ils sont alors écrits dans la mémoire du Pluto.)

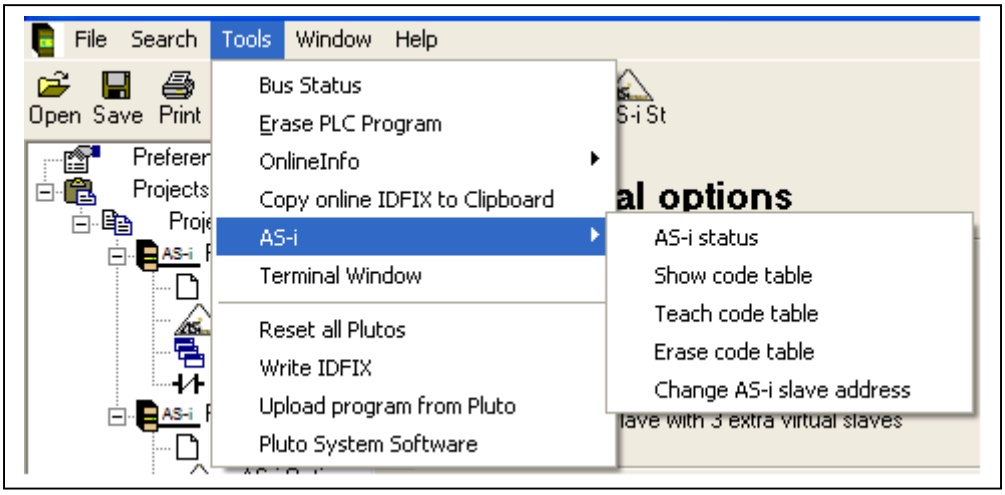
9.2.2.1 Activer la sortie d'un esclave

Il peut être nécessaire de configurer des paramètres pour que l'esclave transmette le code de sécurité. Cliquer sur « Teach safety codes » puis cliquer du bouton droit sur le symbole de l'esclave, cliquer du bouton gauche sur « Set param and Data » et sélectionner la sortie à activer. Quand « Code present » est affiché, cliquer sur « Save codes ».

Cliquer du bouton droit sur l'esclave, cliquer du gauche sur « Set param and Data » et sélectionner la sortie à activer.

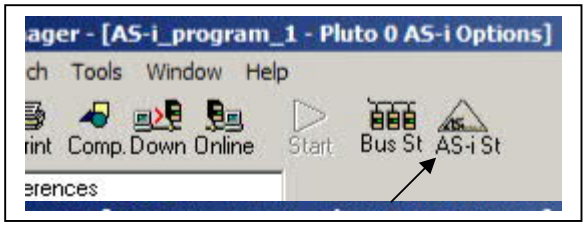
9.3 Autres outils en ligne

Vous trouverez d'autres outils En ligne sous Tools → AS-i.

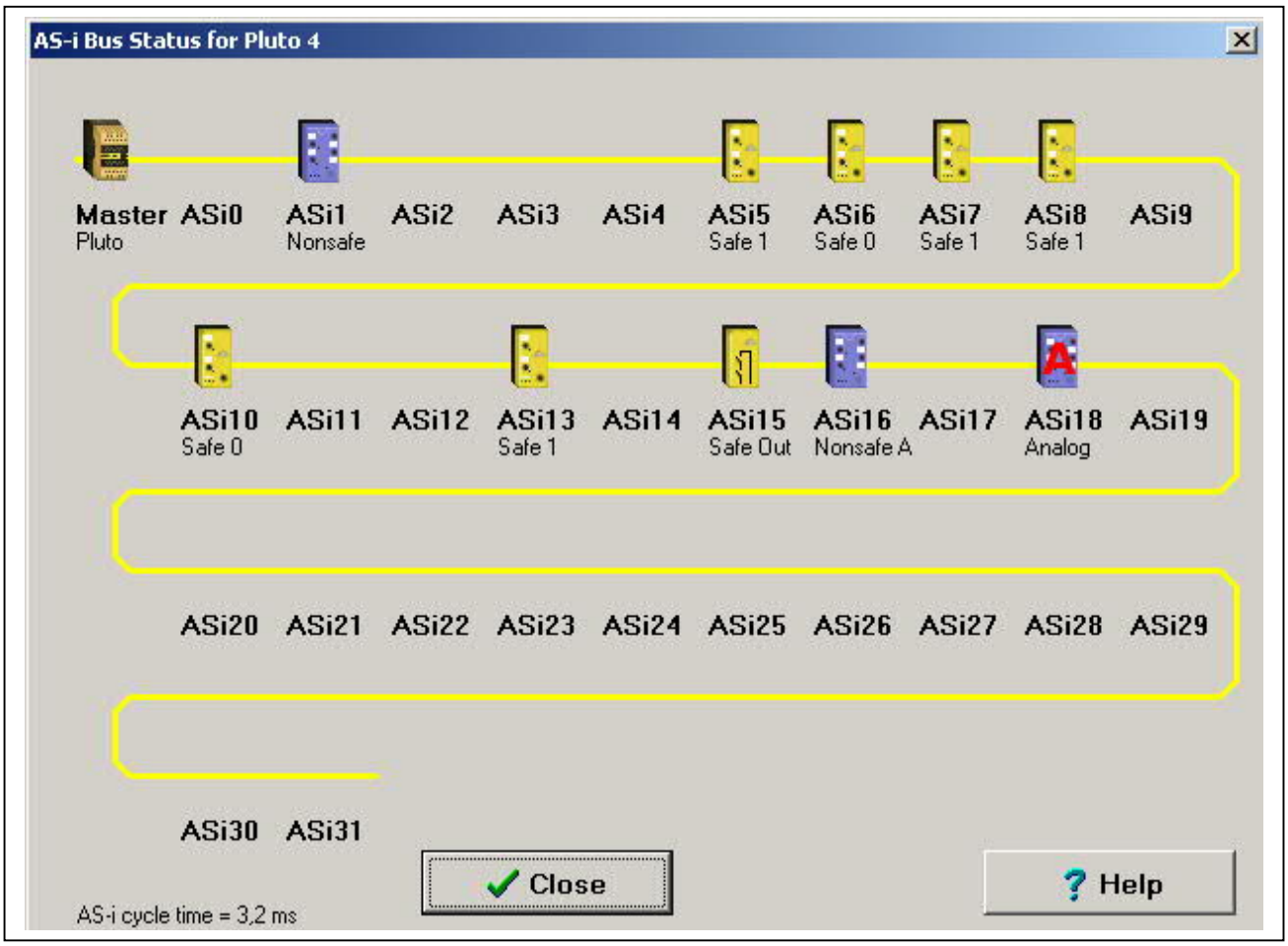


9.3.1 État du bus AS-i

Pour accéder à l'état du bus AS-i, choisir « AS-i/AS-i status » sous « Tool » ou cliquer directement sur le symbole « AS-i St » dans la barre des outils.



L'état contient des informations sur le bus AS-i comme les types d'esclaves, l'état des esclaves de sécurité, le temps de cycle du bus, etc.



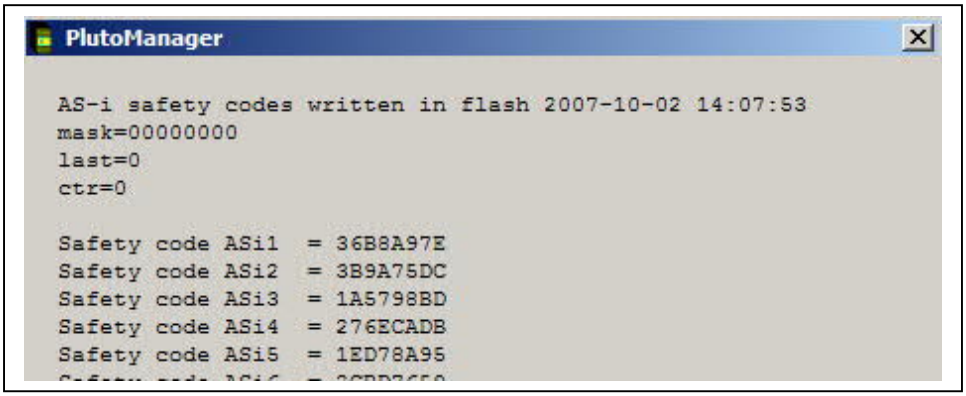
Vous trouverez des explications sous « Help ».

Explication des symboles dans la fenêtre d'état.

Pluto	Pluto (comme maître/entrée/entrée de sécurité)
Other PLC	Autre automate comme maître sur le bus
Nonsafe	Esclave non sécurisé
Analogue	Esclave entrée de sécurité
Unknown	Esclave inconnu. Peut se produire si le maître ne lit pas continuellement les ID des esclaves
Safe input	Esclave entrée de sécurité
Safe output	Esclave sortie de sécurité
Not in project	Présence d'un nœud ne faisant pas partie du projet courant
No Scan / Ch Err	Erreur de nœud ou de surveillance de canal. Double-cliquer sur le nœud pour une explication
Pluto missing	L'esclave Pluto n'est pas lu par le maître
Missing	Nœud faisant partie du projet mais absent du bus

9.3.2 « Show code table » (Afficher le tableau des codes)

Une liste de tous les codes de sécurité.



9.3.3 « Teach code table » (Apprentissage du tableau des codes)

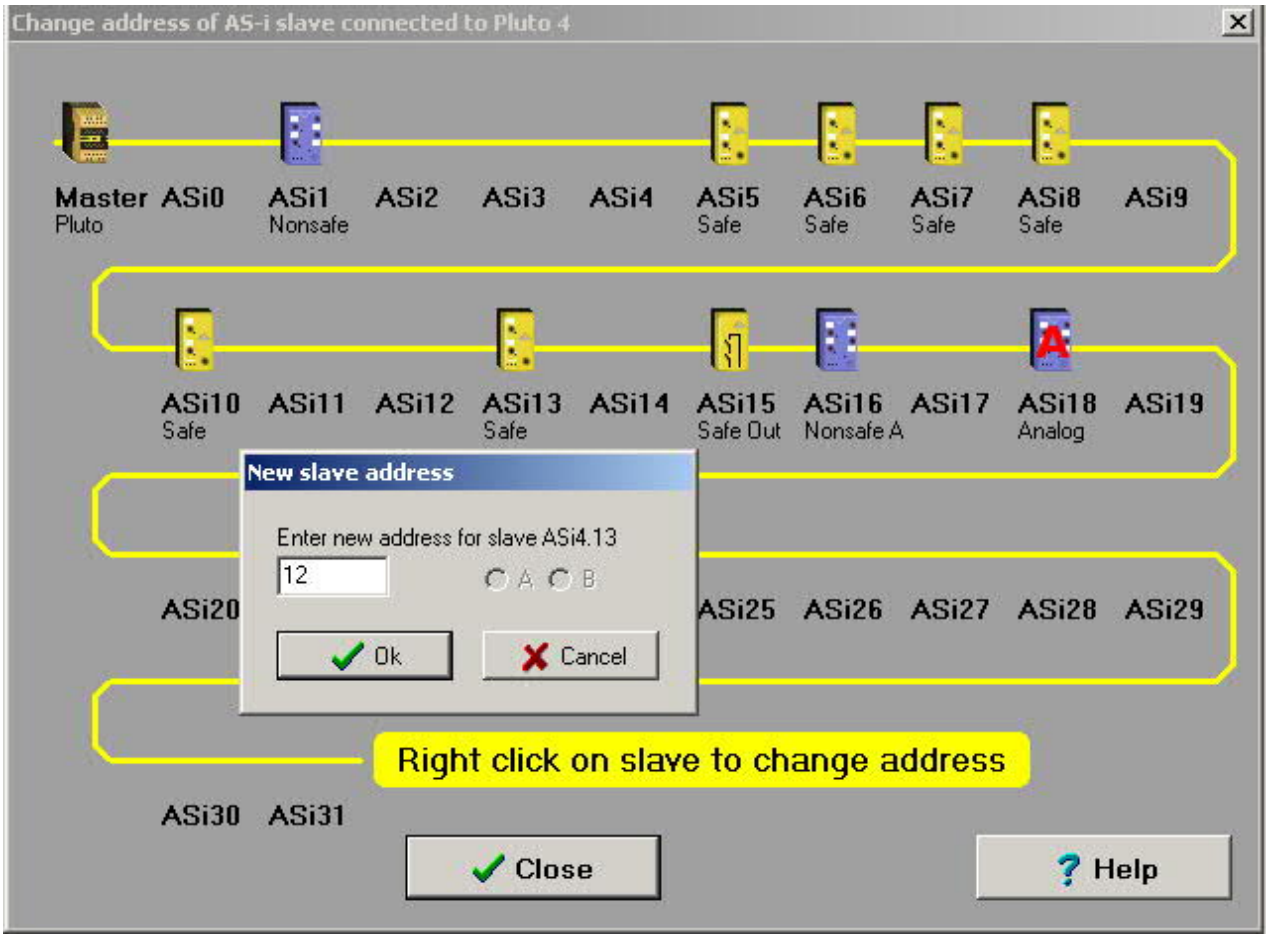
Même fonction que « Apprentissage des codes de sécurité » sur la page « AS-i options ». (Voir 7.2.2 ci-dessus)

9.3.4 « Erase code table » (Effacer le tableau des codes)

Il est possible d'effacer les codes de sécurité de la mémoire de Pluto et de IDFIX-DATA / IDFIX-PROG (si connecté).

Les codes de sécurité ne sont pas enregistrés dans le programme de l'automate et ne sont donc pas effacés quand le programme est effacé.

9.3.5 Modifier l’adresse d’un esclave

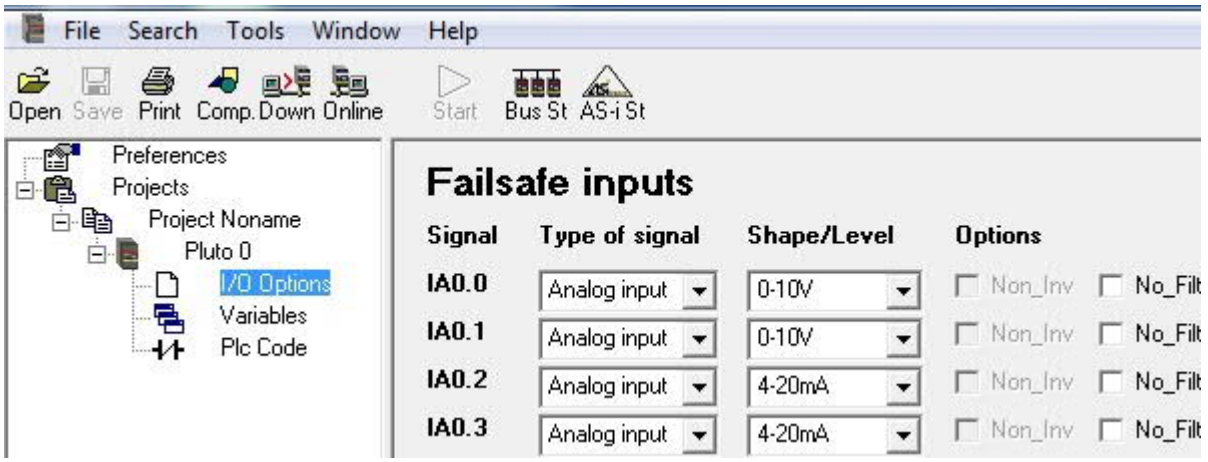


Exemples de modification d’adresse. L’esclave 13 devient 12.

Résultat après le changement d’adresse.

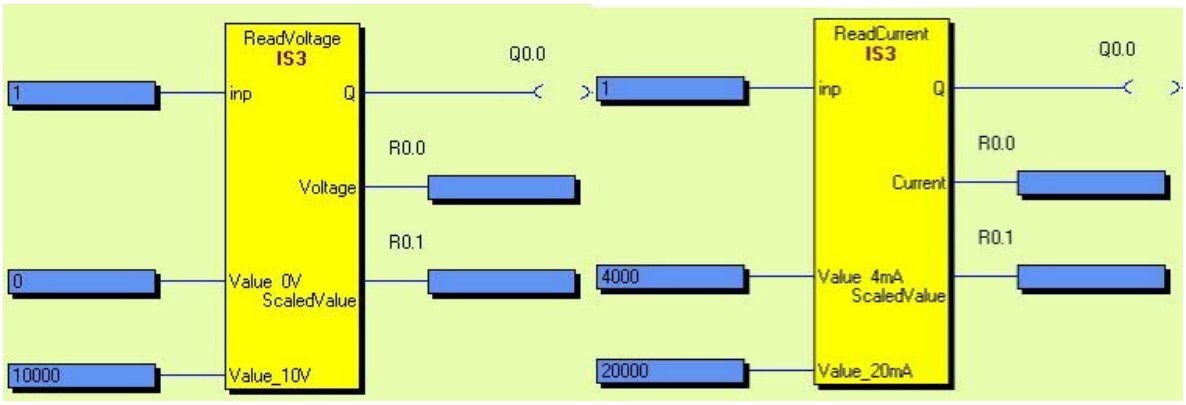
10 Entrées analogiques Pluto D20 et D45 – blocs de fonction

Pluto D20 et Pluto D45 sont équipés respectivement de 4 et 8 entrées analogiques de sécurité 4-20 mA/0-10 V. Ces entrées (D20 : IA0 – IA3, D45 : IA0 – IA7) peuvent être configurées dans Pluto Manager comme des entrées de sécurité « ordinaires », comme des entrées analogiques 0-10 V ou comme des entrées analogiques 4-20 mA.



IA0.0 et IA0.1 sont configurées comme des entrées analogiques 0-10 V, et IA0.2 et IA0.3 comme des entrées analogiques 4-20 mA.

Pour l'entrée analogique 0-10 V, le bloc de fonction « ReadVoltage » est nécessaire, et pour l'entrée analogique 4-20 mA, le bloc de fonction « ReadCurrent » est nécessaire. Ces deux blocs de fonction sont inclus dans la bibliothèque « Analog01.fps ». Sont également inclues des versions 32 bit des blocs de fonction (« ReadVoltage_32 » et « ReadCurrent_32 ») pour l'utilisation avec des doubles registres.



Blocs de fonction ReadVoltage et ReadCurrent.

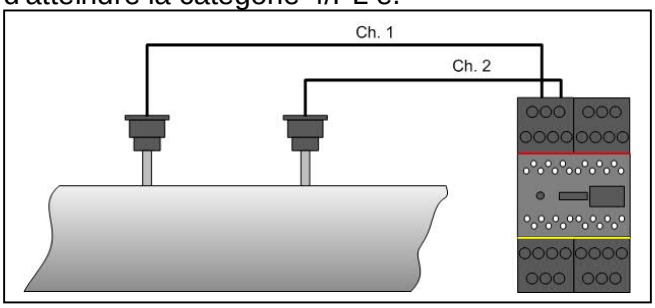
Bloc de fonction **ReadVoltage**. Description des entrées et des sorties :	
Input	Entrée connectée au bloc.
Value 0V	Valeur d'entrée pour mise à l'échelle. À 0 V, la sortie « Scaled value » affichera cette valeur.
Value 10V	Valeur d'entrée pour mise à l'échelle. À 10 V, la sortie « Scaled value » affichera cette valeur.
Q	Sortie OK. La valeur est dans les limites.
Tension	Sortie avec valeur absolue étalonnée en mV.
Scaled Value	Sortie avec valeur mise à l'échelle.

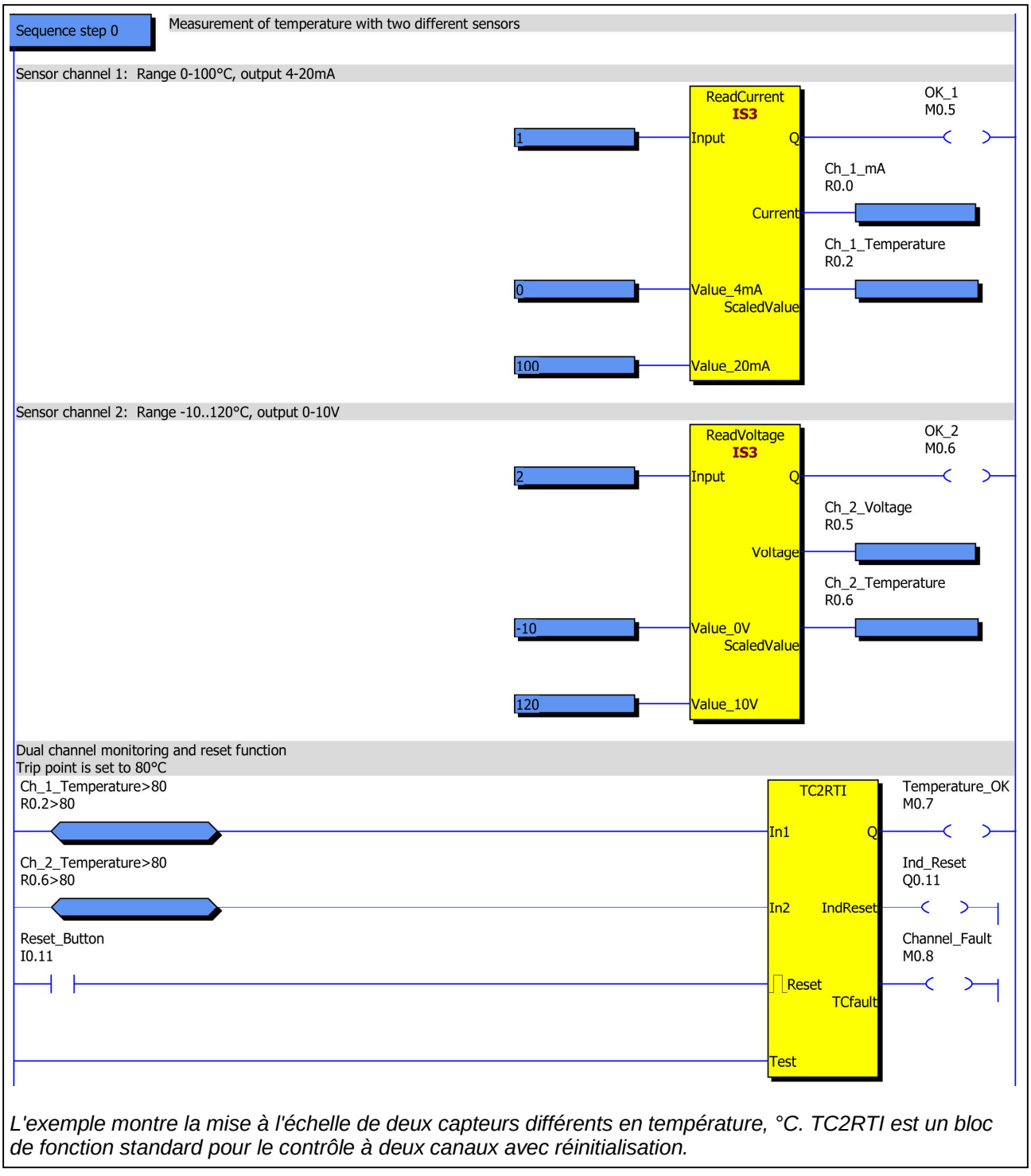
Bloc de fonction **ReadCurrent**. Description des entrées et des sorties :	
Input	Entrée connectée au bloc.
Value 4mA	Valeur d'entrée pour mise à l'échelle. À 4 mA, la sortie « Scaled value » affichera cette valeur.
Valeur 20mA	Valeur d'entrée pour mise à l'échelle. À 20mA, la sortie « Scaled value » affichera cette valeur.
Q	Sortie OK. La valeur est dans les limites.
Current	Sortie avec valeur absolue étalonnée en µA.
Scaled Value	Sortie avec valeur mise à l'échelle.

Remarque : pour qu'une application atteigne le niveau SIL 3/PL e, deux capteurs en parallèle, avec chacun une entrée analogique et un bloc de fonction, doivent être utilisés.

10.1 Exemple d'application avec deux capteurs – mesure de la température

Avec l'exemple d'application ci-dessous, l'utilisation de deux capteurs différents permet d'atteindre la catégorie 4/PL e.

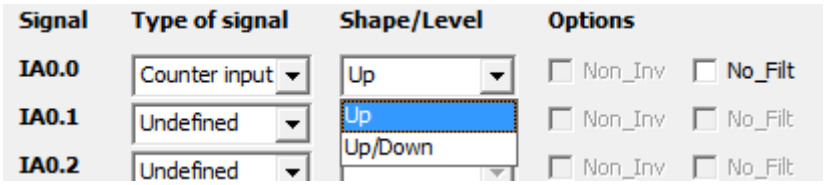




L'exemple montre la mise à l'échelle de deux capteurs différents en température, °C. TC2RTI est un bloc de fonction standard pour le contrôle à deux canaux avec réinitialisation.

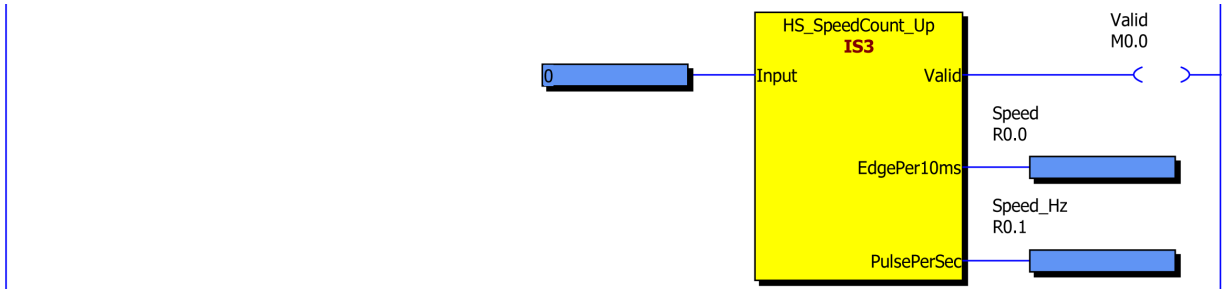
11 Entrées de compteur Pluto D45

Pour Pluto D45, les entrées IA0 – IA3 peuvent être configurées comme des entrées de compteur (comptage d'impulsions) qui fonctionnent pour des fréquences jusqu'à 14000 Hz. Les entrées de compteur IA0 – IA3 peuvent être utilisées selon deux modes, le comptage croissant et le comptage croissant/décroissant. Une description est donnée dans le manuel de matériel de Pluto. Les entrées doivent être configurées dans Pluto Manager.



Configuration de l'entrée de compteur.

Pour les entrées de compteur configurées pour le comptage « croissant », le bloc de fonction « HS_SpeedCount_Up » doit être utilisé.

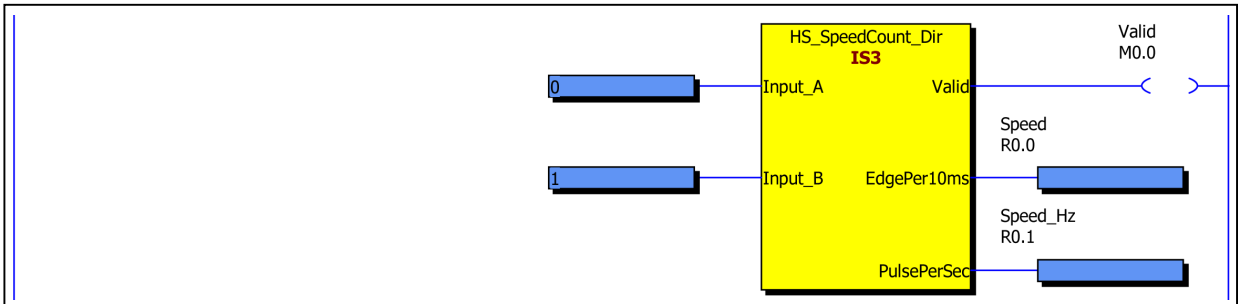


Exemple : l'entrée IA0 est configurée comme entrée HS_SpeedCount_Up.

Bloc de fonction **HS_SpeedCount_Up**. Description des entrées et des sorties :	
Input	Entrée connectée au bloc.
Valid	Sortie OK. La valeur est dans les limites.
EdgePer10ms*	Vitesse de sortie aux bords par 10 ms. À connecter à un registre (R).
PulsePerSec*	Vitesse de sortie en Hz. À connecter à un registre (R).

*Les deux sorties se réfèrent à la même vitesse, seule la mise à l'échelle diffère.

Pour les entrées de compteur configurées pour le comptage « croissant/décroissant », le bloc de fonction « HS_SpeedCount_Dir » doit être utilisé.



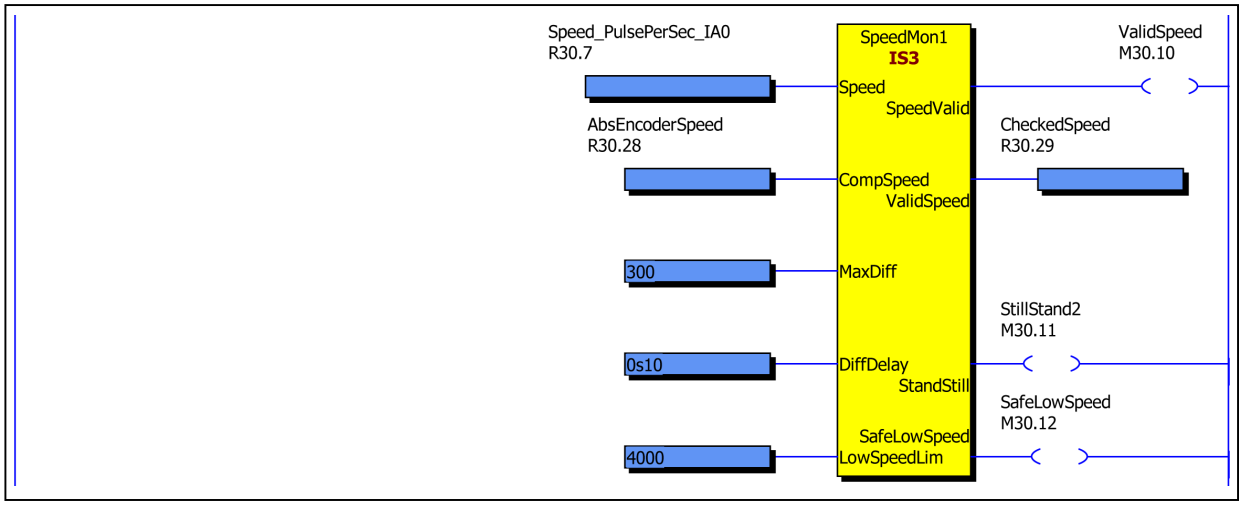
Exemple : les entrées IA0 et IA1 sont configurées comme entrée HS_SpeedCount_Dir.

Bloc de fonction **HS_SpeedCount_Dir**. Description des entrées et des sorties :	
Input_A	Entrée A connectée au bloc.
Input_B	Entrée B connectée au bloc.
Valid	Sortie OK. La valeur est dans les limites.
EdgePer10ms	Vitesse de sortie aux bords par 10 ms. À connecter à un registre (R).
PulsePerSec	Vitesse de sortie en Hz. À connecter à un registre (R).

*Les deux sorties se réfèrent à la même vitesse, seule la mise à l'échelle diffère.

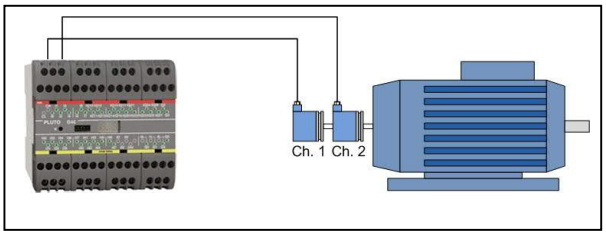
Pour le contrôle de la vitesse et de l'immobilisation, il est possible d'utiliser la fonction « SpeedMon1 ». Les entrées de vitesse peuvent obtenir les valeurs de sources différentes telles que les blocs de fonction pour les codeurs incrémentaux, les codeurs absolus, les entrées analogiques, etc.
Le bloc de fonction présente trois fonctions de sécurité :

- Comparaison d'un registre « Speed » à un second registre de vitesse « CompSpeed » et contrôle que la différence ne dépasse pas la valeur définie dans le registre d'entrée « MaxDiff ». Si la différence est dans la limite, la sortie « SpeedValid » est définie et la sortie « ValidSpeed » sera égale à « Speed » L'entrée de temporisation « DiffDelay » est temporisée pour la comparaison. Le bloc permet aux deux valeurs « Speed » et « CompSpeed » de différer d'une valeur supérieure à MaxDiff pendant ce temps.
- Contrôle de l'immobilisation de l'entrée « Speed » avec hystérésis. La sortie « StandStill » est définie lorsque la valeur de l'entrée « Speed » était égale à 0 pendant 0,7 s. Lorsqu'une augmentation/diminution de la valeur « Speed » est permise trois fois dans chaque sens.
- Vitesse de sécurité (SLS). La sortie SafeLowSpeed est définie lorsque la valeur d'entrée « Speed » est inférieure à la valeur d'entrée « LowSpeedLim ».

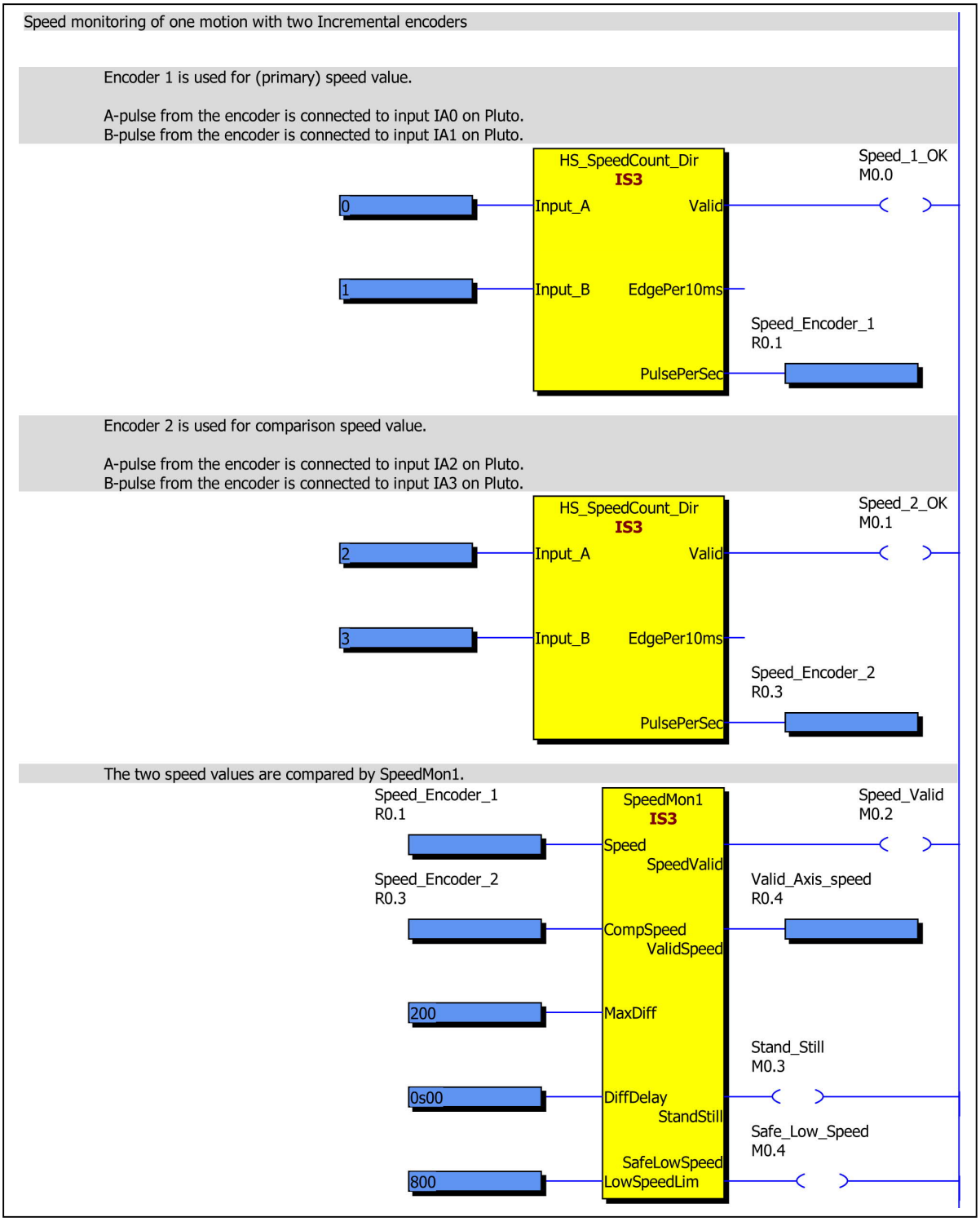


Bloc de fonction **SpeedMon1**. Description des entrées et des sorties :	
Vitesse	Registre d'entrée de la valeur de vitesse (entrée de vitesse principale)
CompSpeed	Registre d'entrée pour le contrôle de la valeur de l'entrée principale de vitesse.
MaxDiff	Entrée pour la différence maximale permise entre « Speed » et « CompSpeed ».
DiffDelay	Temporisation pour la comparaison. MaxDiff peut être dépassé pendant cet intervalle de temps.
LowSpeedLim	Valeur limite de la vitesse de sécurité.
Vitesse/valide	Sortie utilisée lorsque les deux valeurs de vitesse sont dans les limites « MaxDiff ».
Valide/vitesse	Normalement égal à l'entrée « Speed ». En cas de défaillance 32767.
StandStill	Sortie définie à l'immobilisation.
SafeLowSpeed	Sortie définie lorsque la vitesse est inférieure à l'entrée « LowSpeedLim ».

11.1 Application avec deux codeurs – contrôle de vitesse

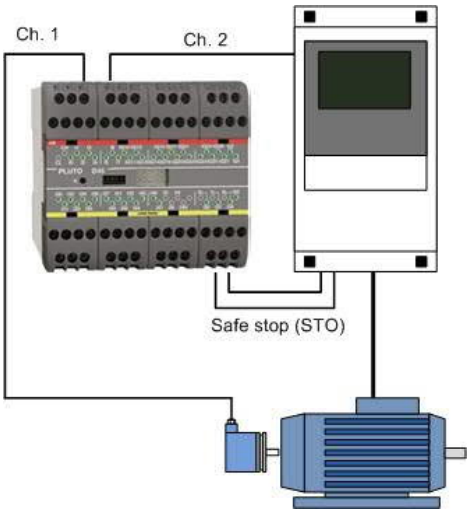


Avec l'exemple d'application ci-dessous, l'utilisation de deux codeurs incrémentaux permet d'obtenir la catégorie 4/PL e pour le contrôle de la vitesse et la fonction de vitesse de sécurité. Pour le contrôle de l'immobilisation, la catégorie 3/PL d peut être atteinte si le mouvement est détecté régulièrement.
Remarque : les défaillances telles que la rupture de câble ne sont pas détectées pendant l'immobilisation. Pour cette raison, chaque immobilisation ne doit pas durer plus de quelques heures.



11.2 Application avec un codeur et une valeur analogique – contrôle de la vitesse

Avec l'exemple d'application ci-dessous, l'utilisation d'un codeur et de la valeur analogique d'un convertisseur de fréquence, la catégorie 3/PL d peut être réalisée pour le contrôle de vitesse, la fonction de vitesse de sécurité et le contrôle de l'immobilisation. Pour le contrôle de l'immobilisation, une détection régulière du mouvement est nécessaire.



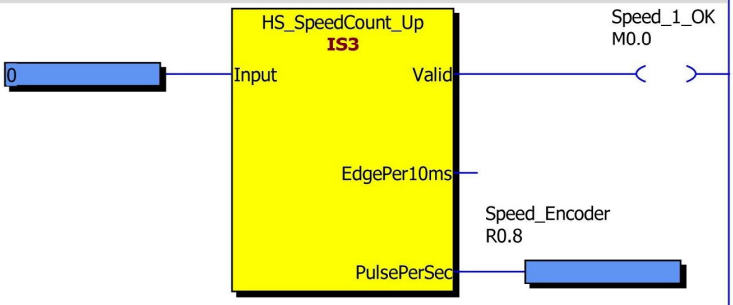
Exemple d'application avec un codeur et un second canal d'un convertisseur de fréquence

Remarque : les défaillances telles que la rupture du câble du codeur ne sont pas détectées pendant l'immobilisation. Pour cette raison, chaque immobilisation ne doit pas durer plus de quelques heures.
Toutefois, la rupture de câble dans le canal analogique est détectée car 4 mA est défini comme 0 vitesse. La rupture de câble entraîne 0 mA and Speed_Freq_Conv = -122. Le bloc SpeedMon1 détectera la défaillance.

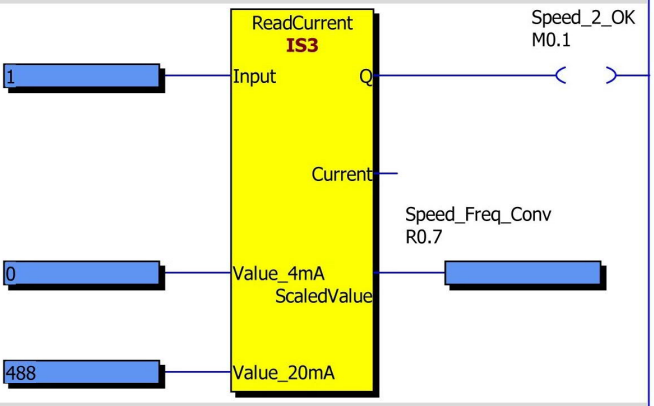
Example with speed monitoring of one motion with one photocell and
an anlogue signal from a frequency converter

The photocell gives 20 pulses/revolution
When the frequency converter gives maximum, the speed is 24.4 revolutions per second and
the analogue output gives 20mA.
At that speed the photocell frequency is 24.4 x 20 = 488 pulses/sec

The photcell is connected to input IA0 on Pluto.

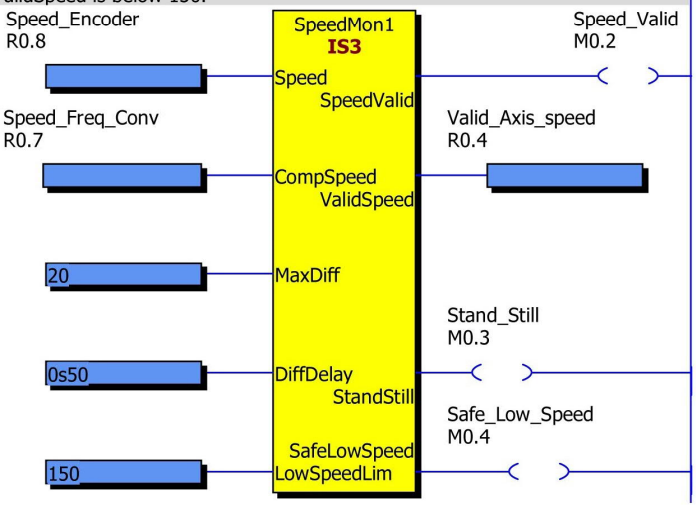


Analogue output from frequency converter is connected to input IA1 on Pluto.
The analogue input value from the frecuency converter is scaled to 488 at 20mA to match the encoder value.



The two speed values are compared by SpeedMon1.
The photocell is used for (primary) speed value.
Analogue value from the frequency converter is used for comparison of the photocell speed value.

The value from the frequency converter is allowed to differ 20 (MaxDiff) however during 0.5 sec (DiffDelay)
it is alowed to differ more than 20. After that SpeedValid switches off and ValidSpeed is set to 32767.
StandStill is initiated when Speed and ValidSpeed is 0. (CompSpeed does not need to be 0, but within MaxDiff)
The output SafeLowSpeed is on when the Speed and ValidSpeed is below 150.



Exemple de programme de contrôle de vitesse avec un codeur et un canal analogique.

12 Variables

Cliquer sur « Variables » dans le menu arborescent pour ouvrir les pages où vous pouvez attribuer un nom symbolique à chaque variable et saisir une description pour chacune.

12.1.1 Nom symbolique

Un nom peut être attribué à chaque variable. Ce nom, dit symbolique, peut être utilisé dans le programme à la place du vrai nom de la variable. Il n'est pas obligatoire d'attribuer un nom symbolique et il peut aussi être attribué ultérieurement.
Les caractères suivants sont autorisés dans les noms de variables :

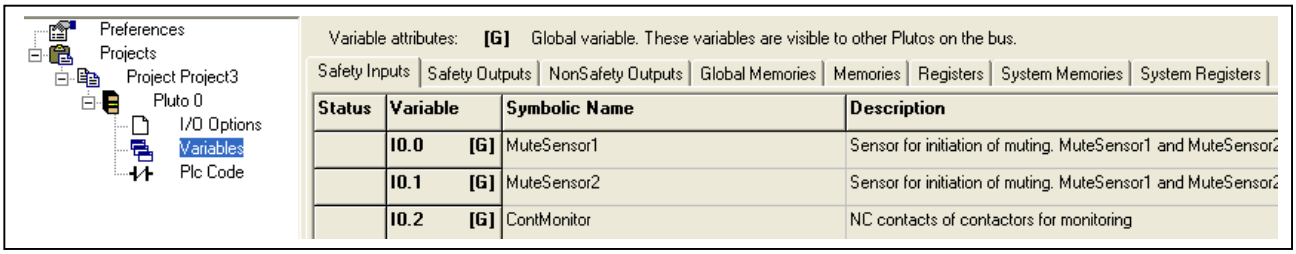
- A – Z, a – z et tous les autres caractères conformément à la norme Unicode (par exemple : caractères scandinaves, grecs, russes et chinois).
- 0 – 9, mais pas comme premier caractère.
- _ (Tiret bas/Underscore) est autorisé mais pas comme premier caractère.
- . (Point) est autorisé avec le jeu d'instructions 2 mais pas le 3.

12.1.2 Description

La description n'a aucune incidence sur les autres fonctions.

12.2 Variables locales/globales

En haut de la page, des onglets représentent tous les types de variables. Les variables peuvent être globales ou locales. Les variables globales peuvent être utilisées par tous les Pluto connectés au bus ; les variables locales ne peuvent utilisées que par « leur » Pluto. Les variables globales sont repérées par (G).



Chaque type/Famille	Famille A20 (sauf B22 et D20), O2
	Variables globales:
Entrées de sécurité	I_.0…7, 10…17
Sorties de sécurité	Q_.0 …Q_.3
Mémoires globales	GM_.0 … GM_.11
	Variables locales:
Entrées de sécurité	-
Entrées non sécurisées	-
Sorties de sécurité	-
Sorties non sécurisées	Q_.10 …Q_.17
Mémoires	M_.0 … M_.599
Registres	R_.0 ..149
*Registres doubles***	*DR_.0…DR_.148 (seulement nombres pairs)*
Mémoires du système	SM_.0 ..199
Registres du système	SR_.0 ..99

***Avec le jeu d'instructions nº 3 seulement. Un registre double est constitué de deux registres consécutifs. Voir la deuxième partie du manuel.*

Chaque type/Famille	Pluto B22
	Variables globales:
Entrées de sécurité	I_.0…7, 10…17
Sorties de sécurité	-
Mémoires globales	GM_.0 … GM_.11
	Variables locales:
Entrées de sécurité	I_.20…I_.25
Entrées non sécurisées	-
Sorties de sécurité	-
Sorties non sécurisées	Q_.10 …Q_.17
Mémoires	M_.0 … M_.599
Registres	R_.0 ..149
*Registres doubles***	*DR_.0…DR_.148 (seulement nombres pairs)*
Mémoires du système	SM_.0 ..199
Registres du système	SR_.0 ..99

***Avec le jeu d'instructions nº 3 seulement. Un registre double est constitué de deux registres consécutifs. Voir la deuxième partie du manuel.*

Chaque type/Famille	Pluto D20
	Variables globales:
Entrées de sécurité	IA_.0…IA_3, I_.4…I_.7, I_10… I_17
Sorties de sécurité	Q_.0 …Q_.3
Mémoires globales	GM_.0 … GM_.11
	Variables locales:
Entrées de sécurité	-
Entrées non sécurisées	-
Sorties de sécurité	-
Sorties non sécurisées	Q_.10 …Q_.17
Mémoires	M_.0 … M_.599
Registres	R_.0 ..149
*Registres doubles***	*DR_.0…DR_.148 (seulement nombres pairs)*
Mémoires du système	SM_.0 ..199
Registres du système	SR_.0 ..99

***Avec le jeu d'instructions nº 3 seulement. Un registre double est constitué de deux registres consécutifs. Voir la deuxième partie du manuel.*

Chaque type/Famille	Pluto B46, S46
	Variables globales:
Entrées de sécurité	I_.0…7, 10…17
Sorties de sécurité	Q_.0 …Q_.3
Mémoires globales	GM_.0 … GM_.11
	Variables locales:
Entrées de sécurité	I_.20..27, 30..37, 40..47
Entrées non sécurisées	-
Sorties de sécurité	Q_.4…Q_.5
Sorties non sécurisées	Q_.10 ...17, 20 ..27
Mémoires	M_.0 … M_.599
Registres	R_.0 ..149
*Registres doubles***	*DR_.0…DR_.148 (seulement nombres pairs)*
Mémoires du système	SM_.0 ..199
Registres du système	SR_.0 ..99

***Avec le jeu d'instructions nº 3 seulement. Un registre double est constitué de deux registres consécutifs. Voir la deuxième partie du manuel.*

Chaque type/Famille	Pluto D45
	Variables globales:
Entrées de sécurité	IA_.0…IA_7, I_10…I_17
Sorties de sécurité	Q_.0 …Q_.3
Mémoires globales	GM_.0 … GM_.11
	Variables locales:
Entrées de sécurité	I_.20..27, 30..37, 40..47
Entrées non sécurisées	-
Sorties de sécurité	Q_.4…Q_.5
Sorties non sécurisées	Q_.10 ...17, 20 ..27
Mémoires	M_.0 … M_.599
Registres	R_.0 ..149
*Registres doubles***	*DR_.0…DR_.148 (seulement nombres pairs)*
Mémoires du système	SM_.0 ..199
Registres du système	SR_.0 ..99

***Avec le jeu d'instructions nº 3 seulement. Un registre double est constitué de deux registres consécutifs. Voir la deuxième partie du manuel.*

Chaque type/Famille	Pluto AS-i
	Variables globales:
Entrées de sécurité	I_.0 et ASi_.1…15
Sorties de sécurité	Q_.0 …Q_.3
Mémoires globales	GM_.0 … GM_.11
	Variables locales:
Entrées de sécurité	I_.1..3, 10..13 et ASi_.16..31
Entrées non sécurisées	Entrées esclaves: ASi_.X.Y*
Sorties de sécurité	-
Sorties non sécurisées	Q_.10..13 et Sorties esclaves: ASq_.X.Y*
Mémoires	M_.0 … M_.149 (avec jeu d'instructions nº 2) M_.0 … M_.599 (avec jeu d'instructions nº 3)
Registres	R_.0 ..149
*Registres doubles***	*DR_.0…DR_.148 (seulement nombres pairs)*
Mémoires du système	SM_.0 ..199
Registres du système	SR_.0 ..99

***Avec le jeu d'instructions nº 3 seulement. Un registre double est constitué de deux registres consécutifs. Voir la deuxième partie du manuel.*

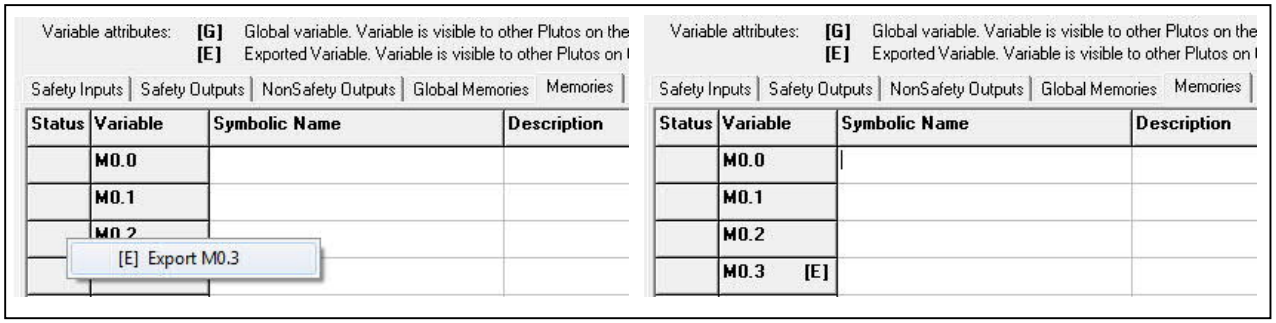
Chaque type/Famille	Pluto B42 AS-i
	Variables globales:
Entrées de sécurité	I_.0…3
Sorties de sécurité	-
Mémoires globales	GM_.0 … GM_.27
	Variables locales:
Entrées de sécurité	I_.10..17, 20..27, 30..37, 40..47 et ASi_.1..31
Entrées non sécurisées	Entrées esclaves: ASi_.X.Y*
Sorties de sécurité	Q_.0…Q_.5
Sorties non sécurisées	Q_.10 ...17, 20 ..27 et Sorties esclaves: ASq_.X.Y*
Mémoires	M_.0 … M_.599
Registres	R_.0 ..149
*Registres doubles***	*DR_.0…DR_.148 (seulement nombres pairs)*
Mémoires du système	SM_.0 ..199
Registres du système	SR_.0 ..99

*X = 1…31 (1B…31B), Y = 1…4
Si, par exemple, ASi_.1 (ASi_.1.1…ASi_.1.4) est un esclave standard non sécurisé (Nonsafe Std) avec 4 entrées, il ne peut exister de ASi_.1B simultanément. En revanche, si ASi_.1 est un esclave A/B (Nonsafe A), il peut aussi exister un ASi_.1B (Nonsafe B).

***Avec le jeu d'instructions nº 3 seulement. Un registre double est constitué de deux registres consécutifs. Voir la deuxième partie du manuel.*

12.2.1 Exportation de variables

Les Pluto avec le jeu d'instructions 3 et une version d'OS 3.2 ou ultérieure permettent de sélectionner plusieurs variables locales (registres, registres doubles, mémoires, sorties de sécurité, sorties non sécurisées et/ou entrées de sécurité) et de les exporter pour les rendre disponibles pour les autres Pluto sur le bus. Cliquer du bouton droit sur la Variable dans Pluto manager, puis cliquer du bouton gauche pour sélectionner le nom de la variable dans le menu contextuel.



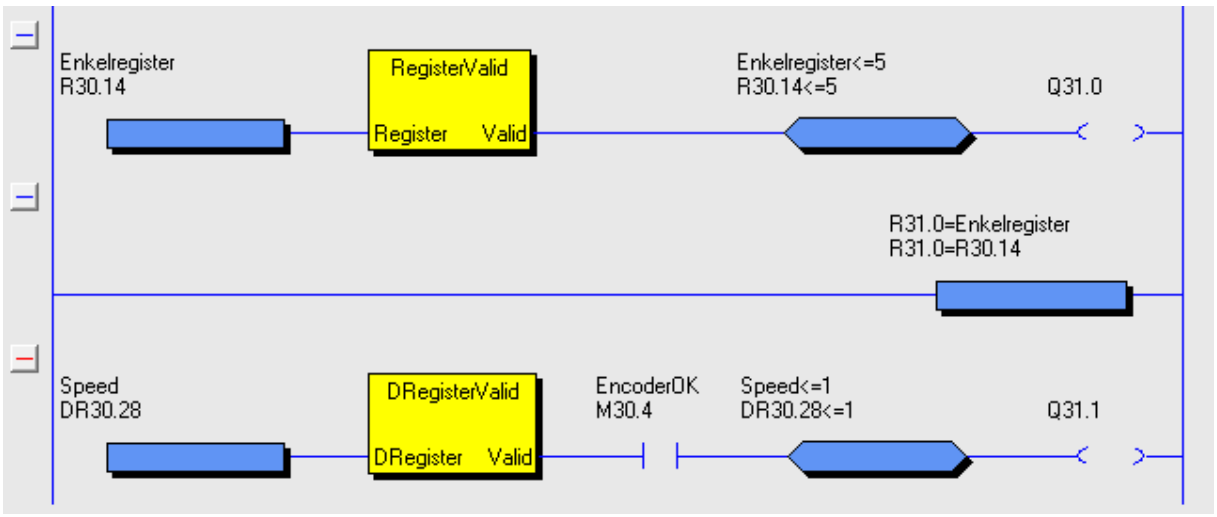
L'exportation de variables ajoute des télégrammes à la communication sur le bus Pluto bus et le nombre de variables pouvant être exportées est limité.

Chacune des options suivantes correspond à une paire de télégrammes supplémentaire :
- 32 variables booléennes
- 16 variables booléennes + 1 registre
- 2 registres
- 1 registre double

Le maximum autorisé est de 4 paires de télégrammes supplémentaires par Pluto et 16 paires de télégrammes supplémentaires par projet. Certains désavantages doivent toutefois être pris en compte :

- La charge du bus augmente considérablement, surtout si des registres mis à jour rapidement sont utilisés (codeurs et valeurs analogiques par ex.) puisqu'une combinaison de transmissions cycliques et de changements d'état est utilisée.
- Pour les registres simples et doubles, le temps d'arrêt maximum est augmenté de 10ms comparé aux variables booléennes.
- Comme le mappage des variables exportées est réalisé par le compilateur, les variables ne sont accessibles que pour les Pluto appartenant au même projet.
- Les variables exportées ne peuvent pas être utilisées dans les passerelles.

Les variables peuvent être utilisées directement dans le programme de l'automate dès qu'elles sont exportées. Les deux blocs de fonctions spéciaux, « RegisterValid » (registre valide) et « DRegisterValid » (registre double valide), sont utilisés pour savoir si un registre simple ou double exporté est valide. Cela n'est normalement pas nécessaire mais si une valeur nulle est utilisée pour active une fonction dangereuse, ces blocs doivent être utilisés puisque la valeur zéro peut aussi signifier l'absence de communication. Un cas typique serait le contrôle de vitesse nulle quand la vitesse nulle est représentée par la valeur 0 :



Dans cet exemple, il n'est pas suffisant d'insérer seulement le contact « Encodeur OK/EncoderOk » (sans DR30.28 et DRegisterValid) dans le code Ladder car il n'est pas garantit que la valeur du registre est valide dans Pluto 31.

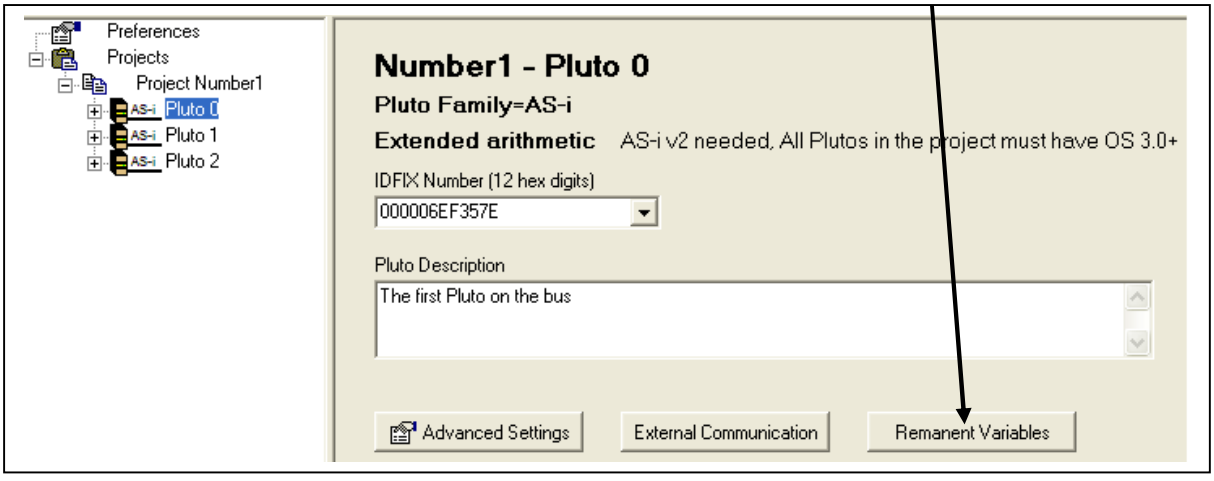
12.3 Variables rémanentes

Une variable rémanente est une variable dont la valeur est conservée quand le Pluto est mis hors tension. Cette fonction est disponible dans les types de Pluto ayant les versions de matériel (HW) et de système d'exploitation (OS) indiqués dans le tableau ci-dessous :

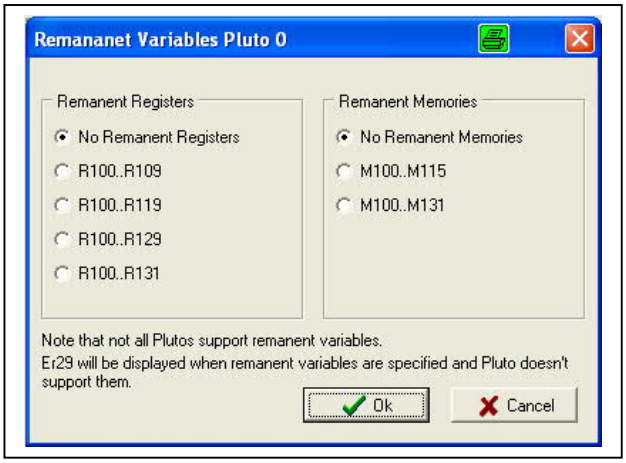
Type de Pluto	Version HW	Version OS
A20 v2	Toutes	Toutes
B20 v2	Toutes	Toutes
S20 v2	Toutes	Toutes
B22	Toutes	Toutes
D20	Toutes	Toutes
B46 v2	2.11 ou plus	3.0 ou plus
S46 v2	2.11 ou plus	3.0 ou plus
D45	Toutes	Toutes
AS-i v2	3.7 ou plus	3.0 ou plus
B42 AS-i	Toutes	Toutes

La version de HW du Pluto est indiquée sur son étiquette. Si aucune version de HW n'est indiquée, le Pluto est trop vieux pour utiliser les variables rémanentes.

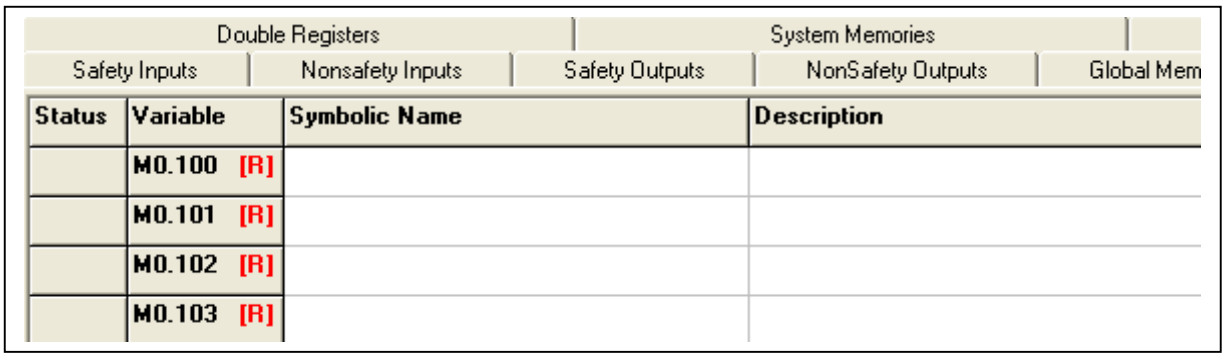
Pour configurer des variables rémanentes, cliquer sur le bouton « Remanent variables ».



Les registres R100 à R131 et/ou les mémoires M100 à M131 peuvent être utilisés comme variables rémanentes selon différentes combinaisons. Toutefois, si tous les registres rémanents (R100..R131) ont été sélectionnés, aucune mémoire rémanente ne peut être sélectionnée.



Dans la liste des variables, les mémoires et les registres pouvant être configurés comme rémanents sont repérés par un [R] rouge.

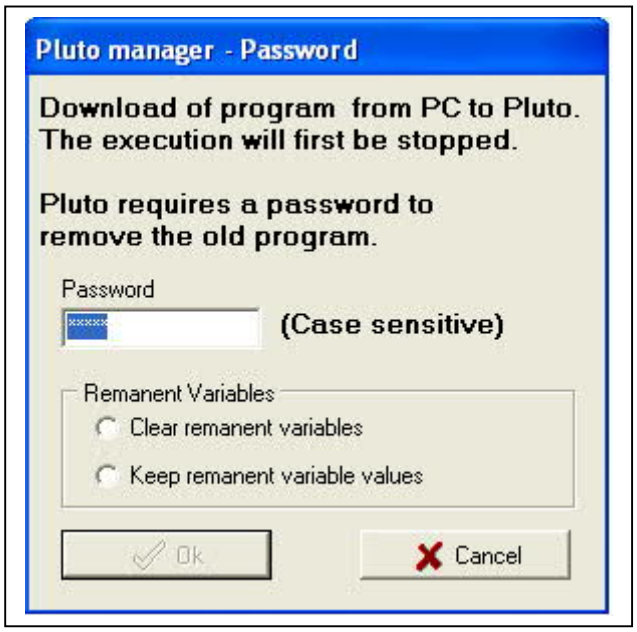


12.3.1 Effacer les variables rémanentes

Lors du transfert d'un programme dans le Pluto, il est demandé si les variables rémanentes doivent être effacées ou conservées.

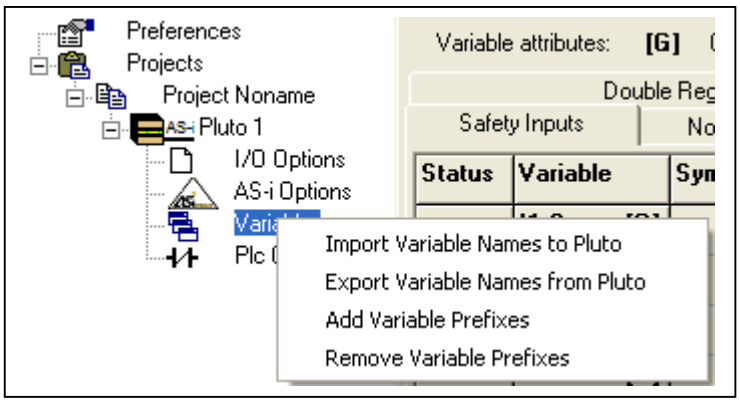
Toutefois, si le nom du projet ou le numéro du Pluto ont été modifiés, les variables sont effacées lors du transfert, même si « Keep remanent variable values » a été coché.

En cas d'erreur 74 (Erreur mémoire rémanente), les variables sont aussi effacées.

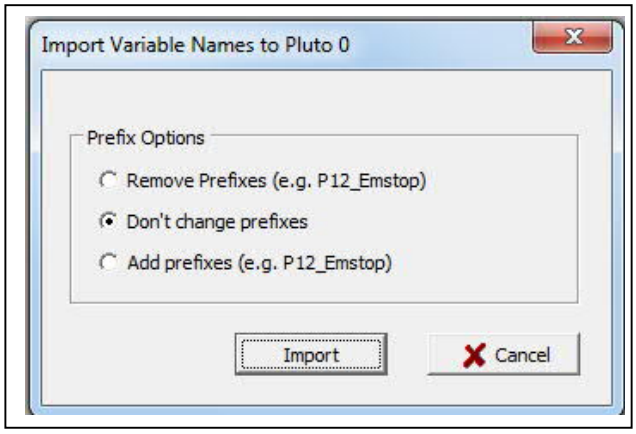


12.4 Exporter et importer les noms de variables

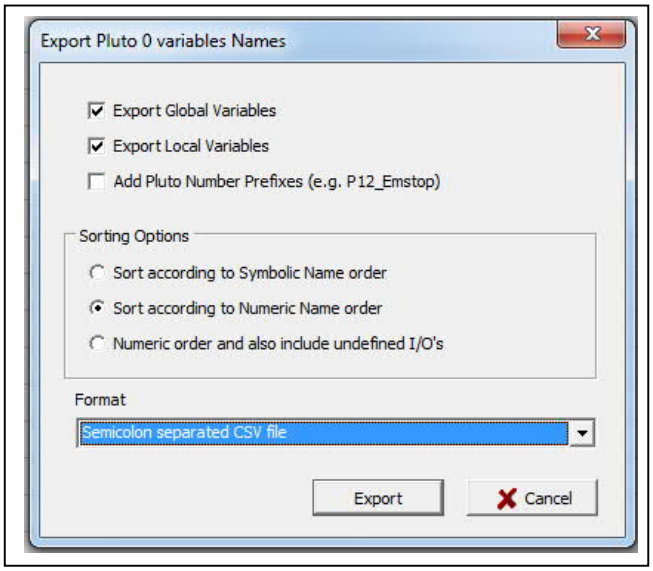
Cliquer du bouton droit sur « Variables » dans le menu arborescent de gauche pour importer les noms de variables ou les exporter dans un fichier .csv qui peut être lu dans Excel par ex.



Cliquer sur « Import Variable Names to Pluto » et la fenêtre suivante s'ouvre. Choisir le préfixe souhaité et cliquer sur « Import » pour importer le fichier choisi.

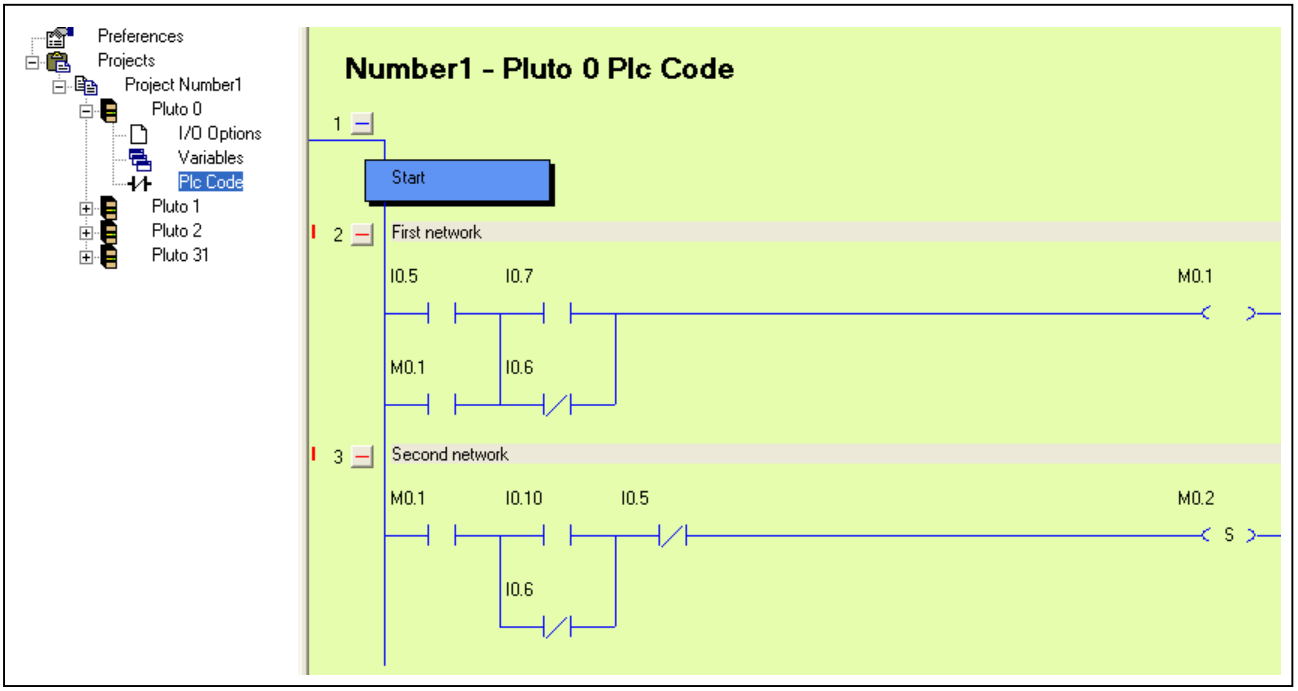


Cliquer sur « Export Variable Names from Pluto » et la fenêtre suivante s'ouvre. Choisir ce qui est souhaité pour les variables globales/locales, le préfixe et l'ordre de tri. Cliquer sur « Export » pour créer le fichier.



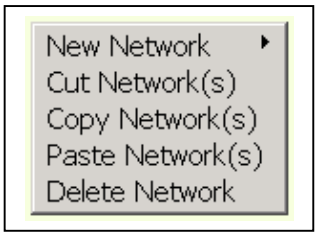
13 Programmation de la logique Ladder

Cliquer sur « PLC Code » dans le menu arborescent pour afficher la page de programmation Ladder.



Le programme Ladder est constitué de réseaux (parfois aussi appelés « rungs ») qui sont numérotés à gauche.

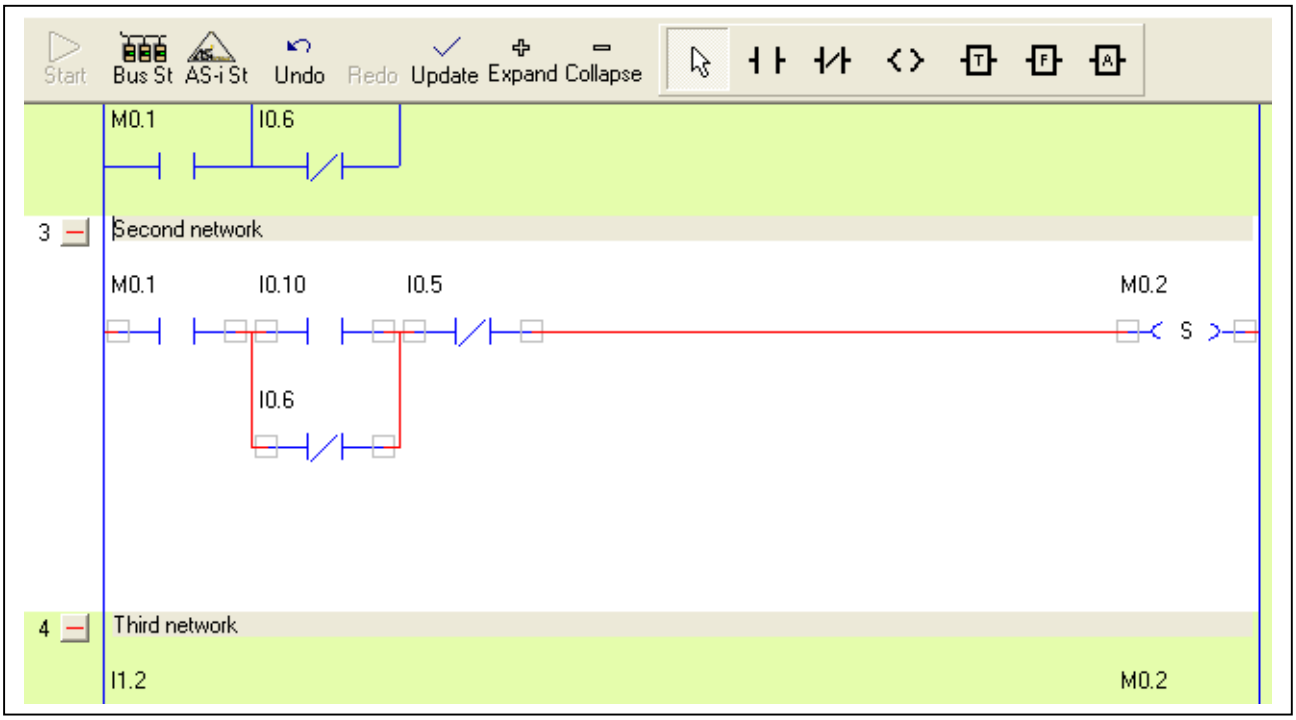
En cliquant du bouton droit de la souris sur un réseau, la boîte de dialogue suivante apparaît. Les options Cut, Copy, Paste et Delete Networks (Couper, Copier, Coller et Effacer Réseaux) fonctionnent comme dans la plupart des autres logiciels sous Windows et ouvrent une nouvelle boîte de dialogue.



Sélectionner « New Network » pour ouvrir un nouveau réseau qui sera inséré sous le réseau sur lequel vous avez cliqué.

13.1 Mode édition

Pour accéder au mode édition, sélectionner « New Network » ou cliquer du bouton gauche de la souris sur un réseau existant. Un seul réseau peut être édité à la fois.



En mode édition, le fond du réseau est blanc, les lignes entre les composants sont rouges et des cases de connexion des lignes sont dessinées à côté des composants. Il est possible de déplacer, d'insérer, de détacher et de supprimer les lignes et les composants Ladder.

Fonctions du mode édition :

Tracer une ligne : Cliquer du bouton gauche de la souris (puis le relâcher) sur un point de connexion du composant. Les points de connexion sont indiqués par des cases (hit box). Déplacer le pointeur là où doit connectée l'autre extrémité de la ligne et cliquer du bouton gauche de la souris pour fixer la ligne.

Modifier une ligne :
- Pour capturer la ligne, cliquer dessus entre deux points de connexion.
- Pour étirer la ligne et la fixer à un troisième point de connexion, cliquer du bouton gauche de la souris sur le point.
- Pour la déconnecter, cliquer du bouton gauche de la souris sur un des points de connexion.
  Une ligne déconnectée peut être connectée à un autre composant.
  Elle peut être supprimée en cliquant en dehors du point de connexion.
- Un clic avec le bouton droit de la souris fait apparaître une boîte de dialogue appelée « Delete line » (Effacer ligne).
- Pour la relâcher, cliquer de nouveau sur le bouton gauche de la souris.

Modifier les propriétés d'un composant : Double-cliquer du bouton gauche de la souris sur un composant pour faire apparaître une boîte de dialogue permettant de changer le nom de la variable, son type (NO, NF, front), etc.

Changer de composant : Cliquer du bouton droit de la souris sur un composant
- pour modifier ses propriétés comme sa fonction et son nom (Components properties..).

- pour supprimer toutes ses connexions (Disconnect component).
- pour supprimer le composant (Delete component).

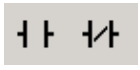
Déplacer un composant : Maintenir enfoncé le bouton gauche de la souris sur un composant puis le faire glisser. Relâcher le bouton au nouvel emplacement.

13.2 Menu Outils

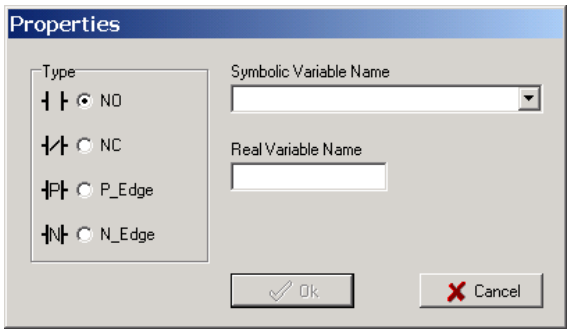
Le menu Outils est affiché en mode édition et sert à insérer des composants.

Cliquer sur un symbole pour insérer le composant correspondant. Le curseur prend alors la forme du symbole. Placer le composant au point désiré dans le réseau et le fixer en cliquant du bouton gauche de la souris. Saisir ses propriétés.

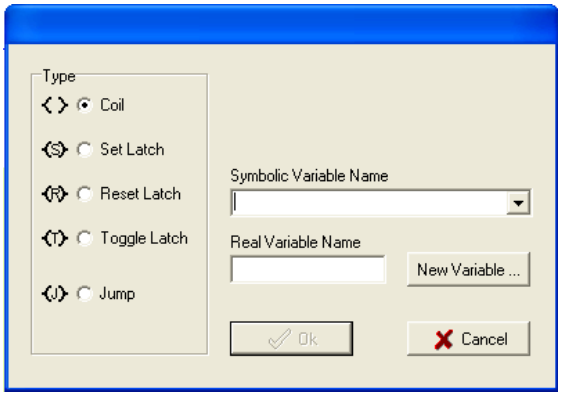
Composants du menu Outils :



Composants Ladder standard. (Ouvre la boîte de dialogue ci-dessous.)



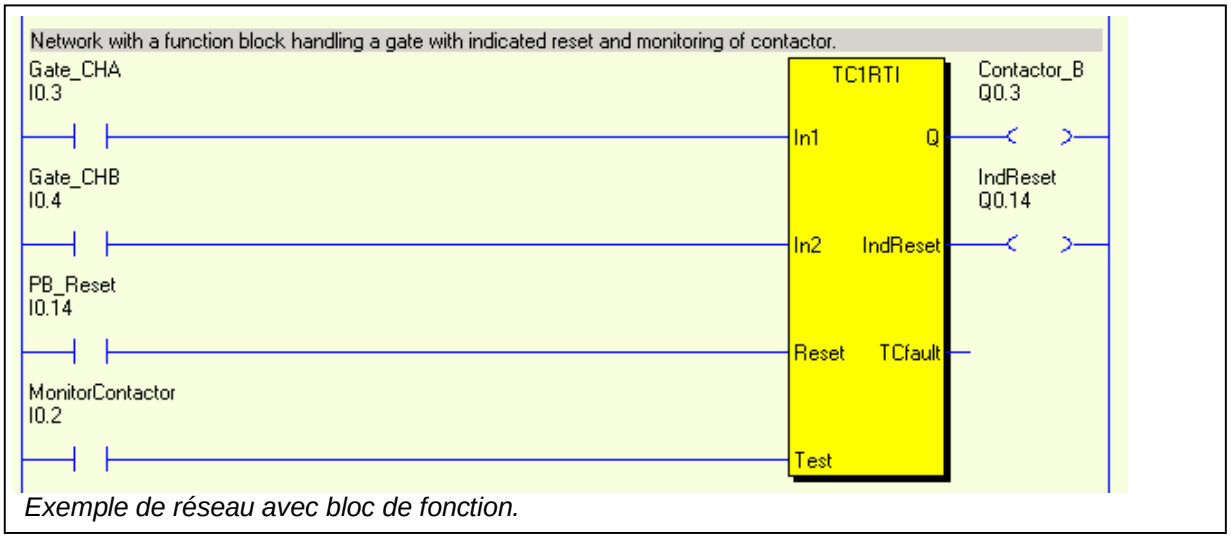
Composants de sortie Ladder standard. (Ouvre la boîte de dialogue ci-dessous.)



Temporisations
Ouvre une boîte de dialogue permettant de sélectionner deux types de temporisations.

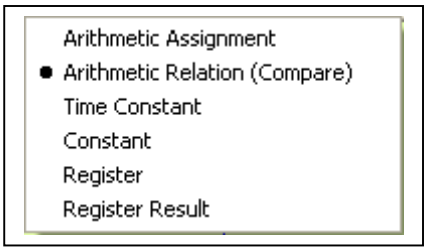
Blocs de fonction.

Cliquer sur « F » pour ouvrir la liste des blocs de fonction disponibles. Le contenu de cette liste dépend des bibliothèques de blocs de fonction choisies. Voir « Choix d'une bibliothèque de blocs de fonction ». Les blocs de fonctions sont décrits dans un document séparé.



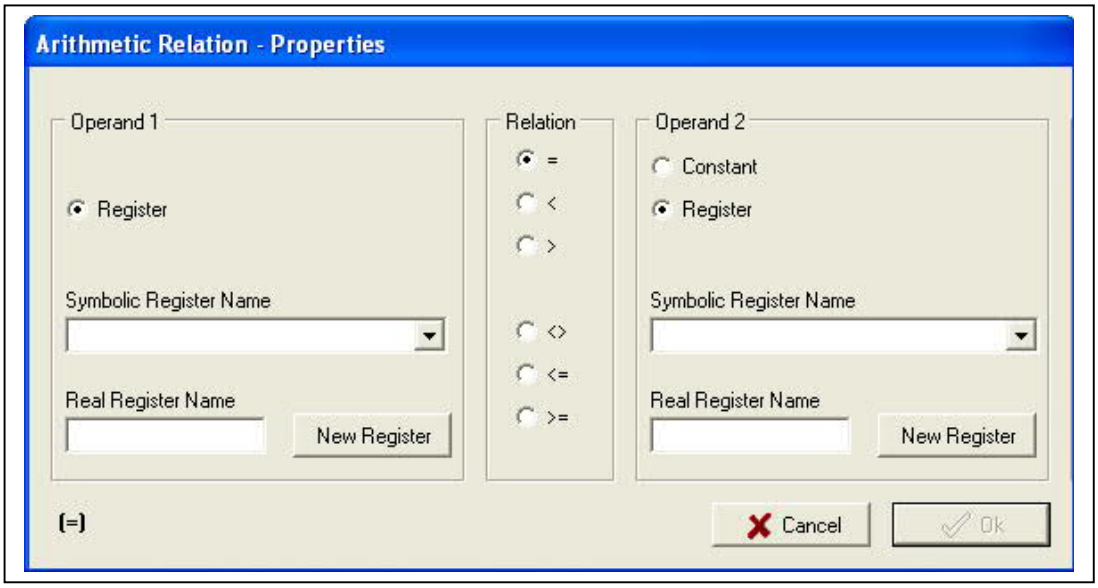
Exemple de réseau avec bloc de fonction.

Fonctions arithmétiques et constantes.
Un clic sur le symbole ouvre le menu déroulant ci-contre.

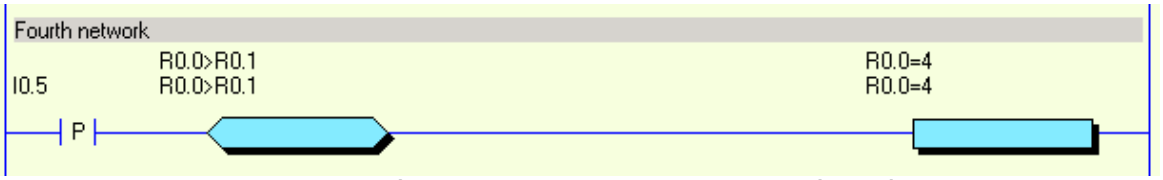


- « Arithmetic Assignment » assigne une valeur à un registre. L'assignation peut revêtir la forme d'une opération mathématique (+, -) ou l'affectation directe d'une valeur.
- « Arithmetic Relation » fait une comparaison avec un registre.
- « Time constant » sert pour les blocs de fonctions exigeant une valeur de temporisation en entrée.
- « Constant » sert pour les blocs de fonctions exigeant une constante en entrée.

Lorsqu'une de ces options est choisie, une nouvelle boîte de dialogue apparaît permettant d'entrer la valeur, la comparaison, etc., sous forme de texte.
(Voir aussi la seconde partie Langage de programmation)



Dans le diagramme Ladder, la fonction arithmétique a l'apparence suivante.



En cas de front montant sur l'entrée I0.5 et si le registre R0.0 est supérieur à R0.1, la valeur 4 est attribuée au registre R0.0.

13.3 Update / Undo

Pour sortir du mode Édition, utiliser soit le bouton « Update », soit le bouton « Undo ». Update confirme les modifications et Undo rétablit le réseau tel qu'il était avant les modifications.

À la place de « Update », on peut utiliser :
- la touche « F3 » ou
- la touche « Échap » suivie de la réponse « Yes » dans la boîte de dialogue.

À la place de « Undo », on peut utiliser :
- la touche « F2 » ou
- la touche « Échap » suivie de la réponse « No » dans la boîte de dialogue.

13.4 Déploiement/Rétraction d'un réseau

Le schéma à contact dit diagramme Ladder peut être déployé ou rétracté. Sous la forme rétractée, seuls les commentaires du réseau apparaissent. Les boutons de la liste d'outils contrôlent les réseaux de tout le diagramme Ladder.
Pour contrôler chaque réseau séparément, utiliser les boutons « + » et « - » situés à gauche de chacun d'eux.

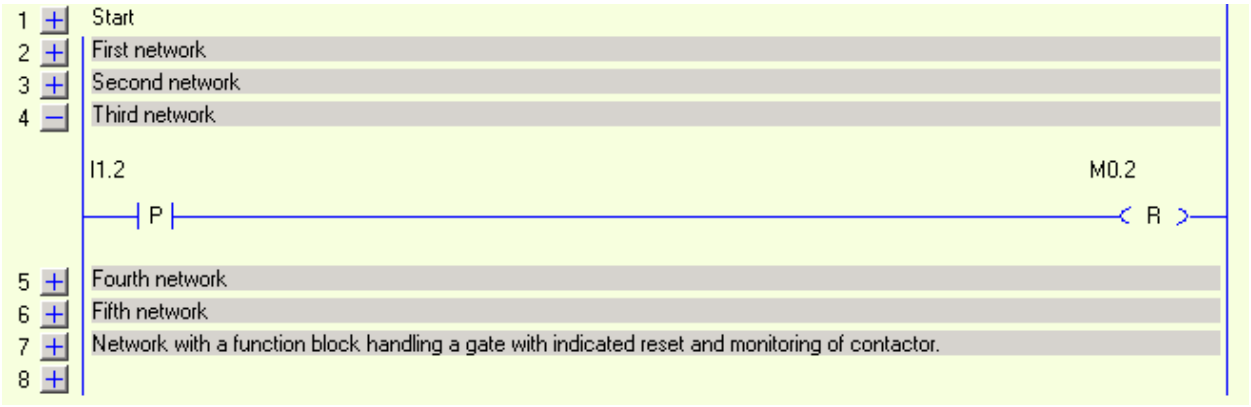
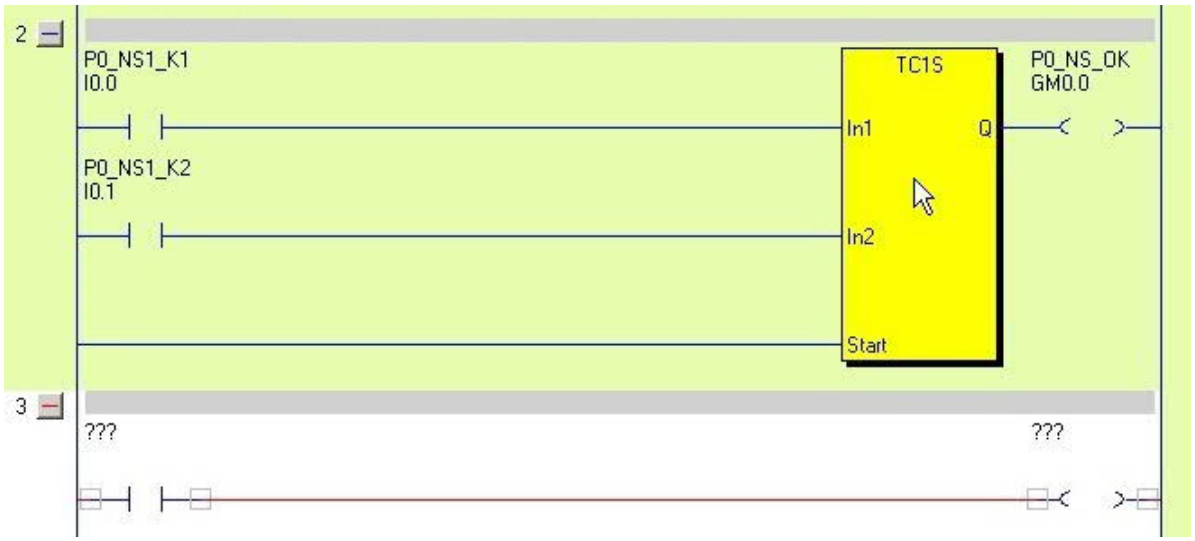


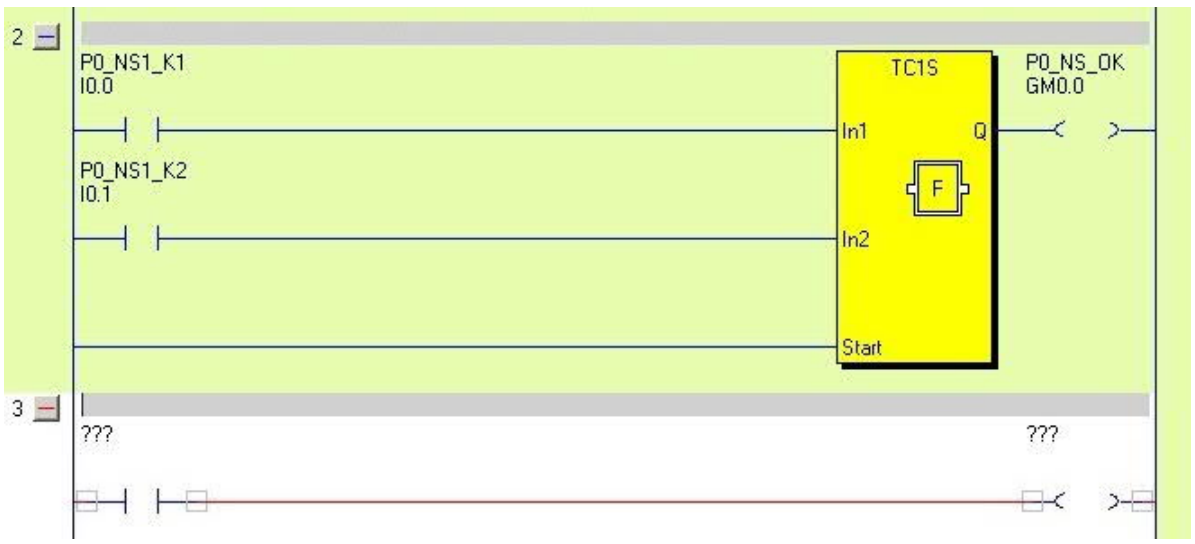
Diagramme Ladder avec réseaux déployés et rétractés

13.5 Glisser-déposer

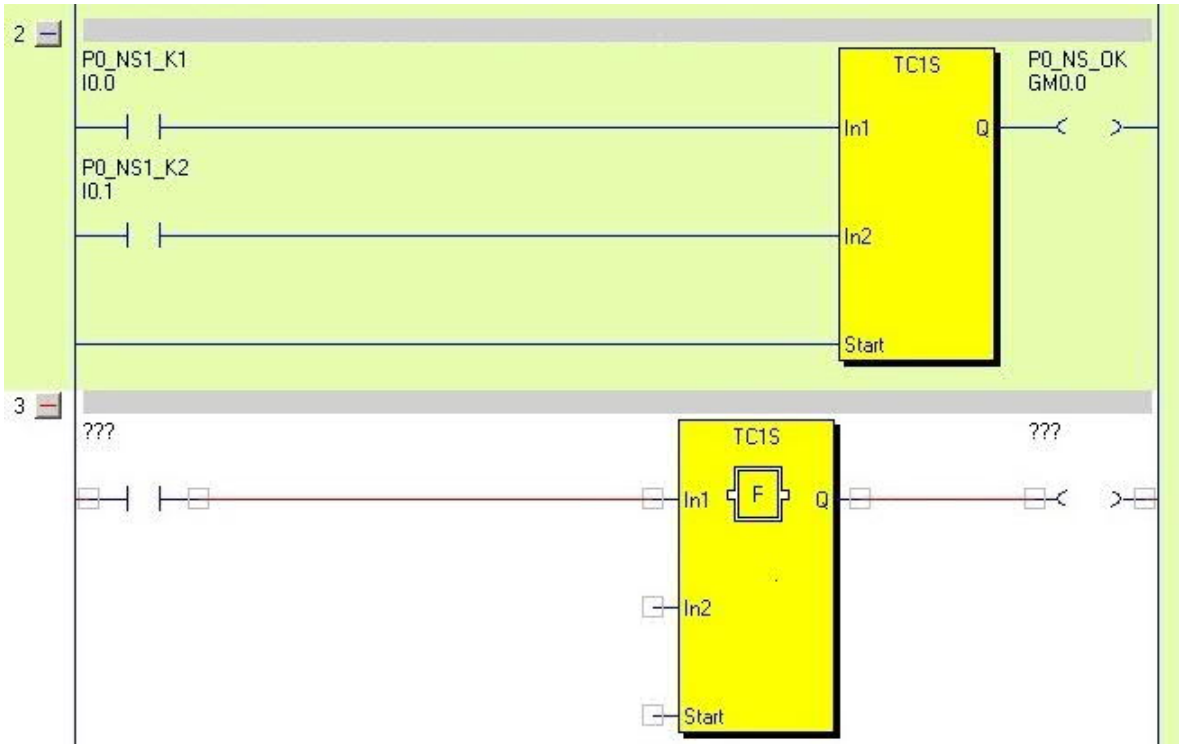
Il est possible d'utiliser le « glisser-déposer » pour copier les composants et blocs de fonctions d'un réseau à un autre. Le réseau dans lequel les composants doivent être déposés doit se trouver en mode d'édition. Placer le curseur sur le composant à copier et cliquer du bouton gauche. Un symbole du composant est affiché. Le faire simplement glisser jusqu'à l'endroit où il doit être déposé et relâcher.



Réseau 3 en mode d'édition, curseur sur le composant à copier.



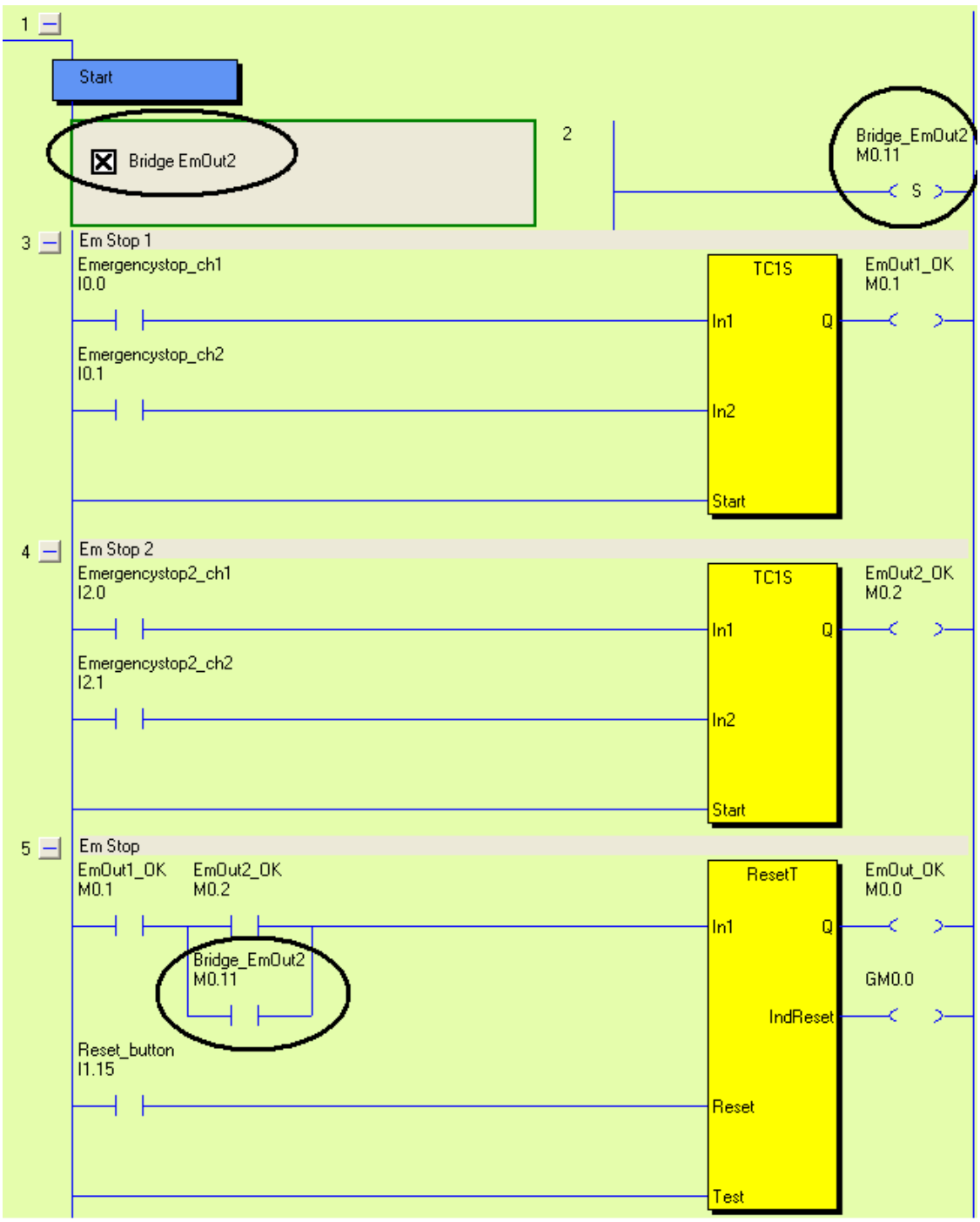
Maintenir le bouton gauche de la souris enfoncé pour afficher le symbole du bloc de fonction.



Faire glisser le composant jusqu'à son nouvel emplacement et relâcher la souris.

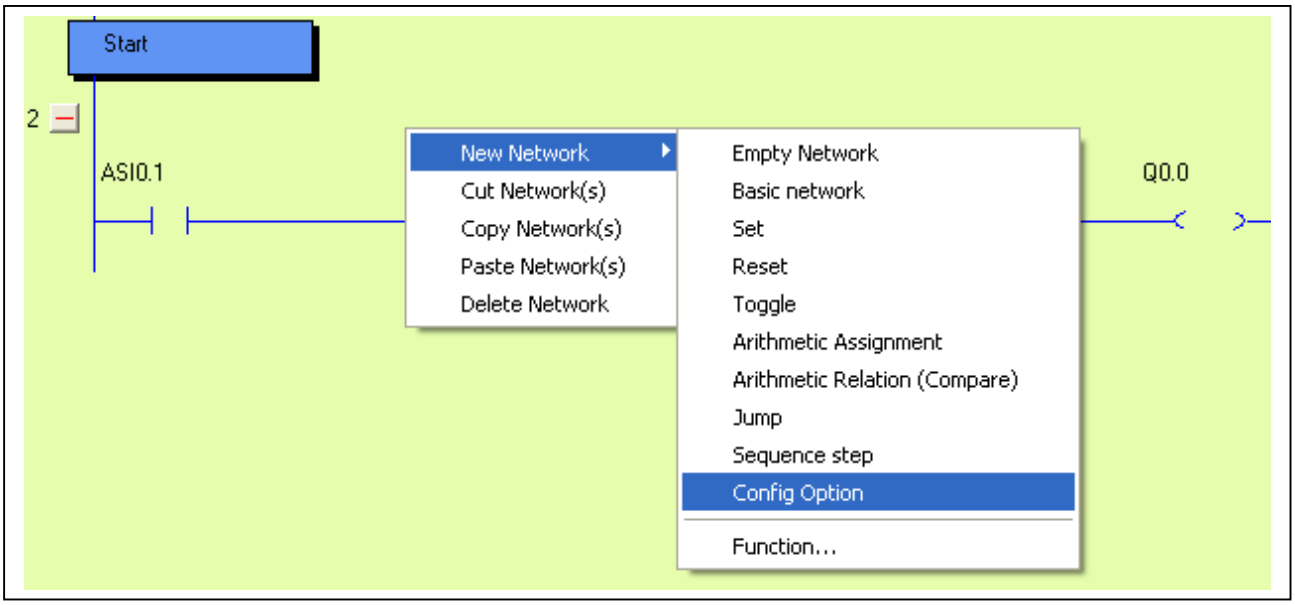
13.6 Options

Les « Options » permettent à l'utilisateur de choisir quelle version du même programme exécuter sans connaître le fonctionnement du programme. Par exemple, le même programme d'automate peut être utilisé pour différents modèles d'une même machine. En fonction du modèle, une case dans le programme doit être cochée ou non, ce qui met une mémoire à 1 ou à 0. Dans l'exemple ci-dessous, cette mémoire est utilisée pour shunter un interrupteur sur un modèle de machine n'étant pas équipé de cet interrupteur. Il est ainsi facile d'adapter le programme à une application spécifique. Les Options fonctionnent bien avec la protection par mot de passe (voir 4.1 Protéger par mot de passe) puisque les Options peuvent avoir un autre niveau de protection que le reste du programme. Noter que les Options doivent être placées au début du programme de l'automate.

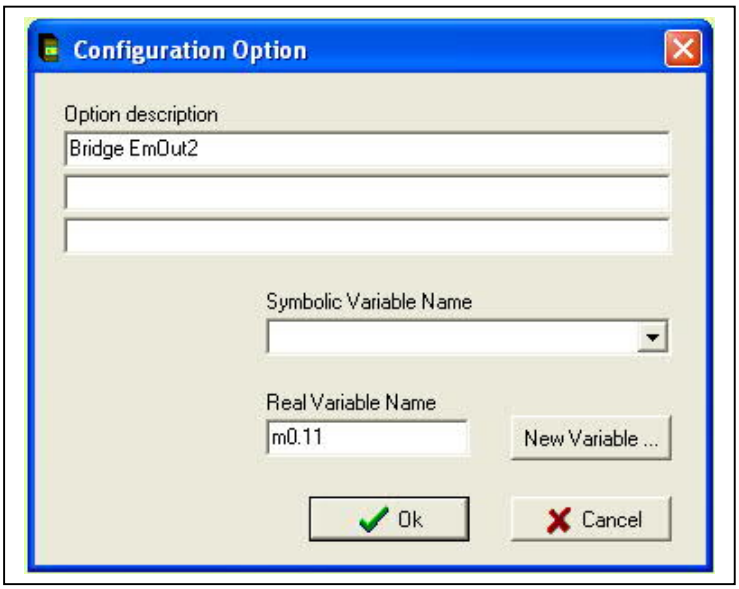


Exemple: si l'Option 1 est cochée, M0.11 est activée et shunte M0.2.

Pour programmer une Option, cliquer du bouton droit dans l'espace du réseau :

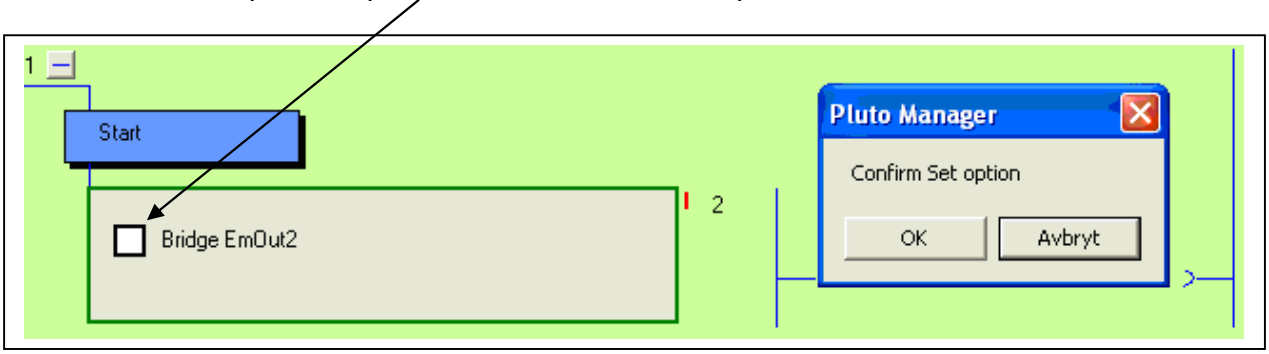


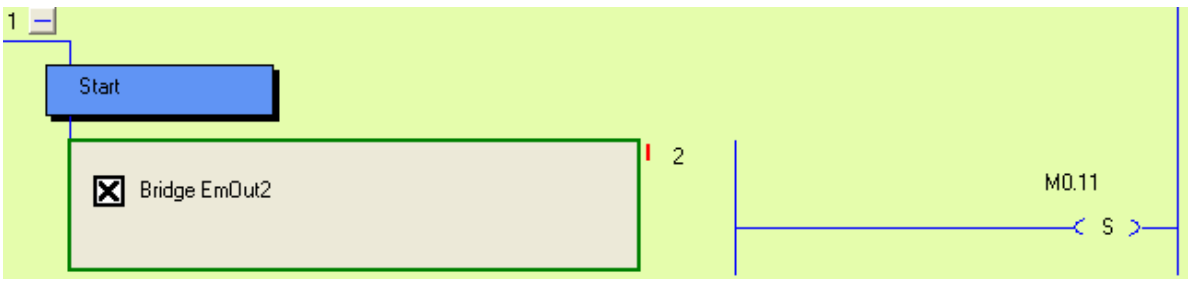
Sélectionner « New network » et « Config Option ».



Indiquer le nom de l'Option sous « Option description » et choisir un nom de variable (seules des mémoires peuvent être utilisées pour les Options). Cliquer sur OK :

Pour activer une Option, cliquer sur la case et confirmer par « OK ».



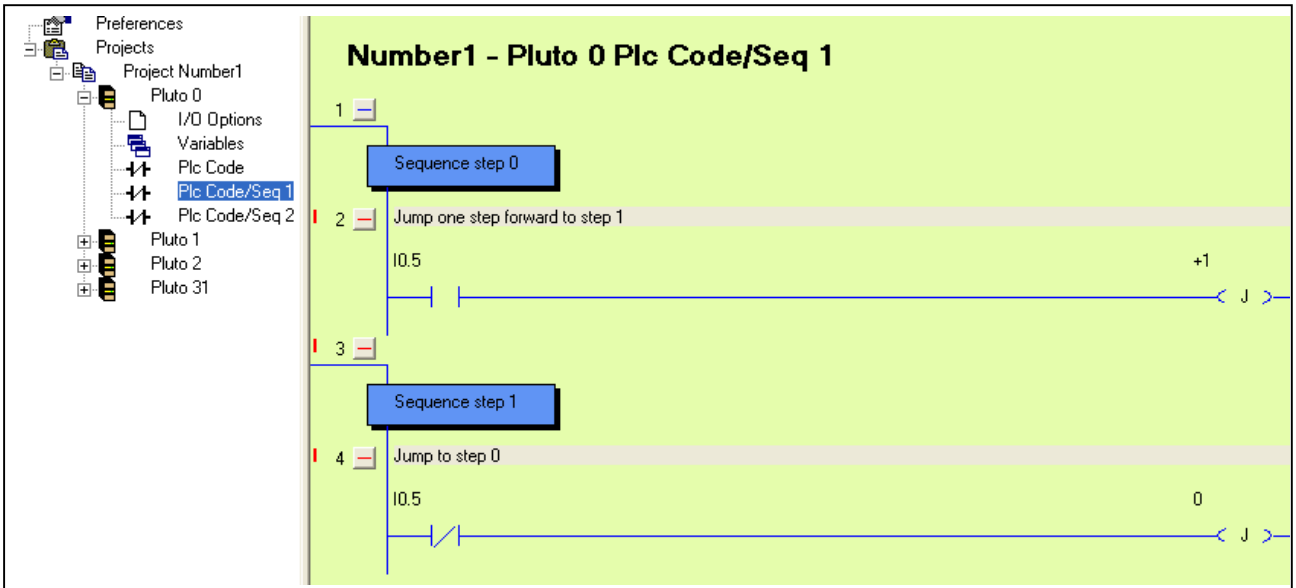


M0.11 est maintenant activée.

13.7 Séquences

Outre le programme principal de l'automate, il est possible d'avoir 9 séquences de 254 pas maximum chacune.

Pour ouvrir une nouvelle séquence :
Cliquer du bouton droit sur le symbole Pluto dans l'arborescence du menu → Sélectionner « New Sequence » → Entrer un numéro de séquence entre 1-9 dans la boîte de dialogue suivante.

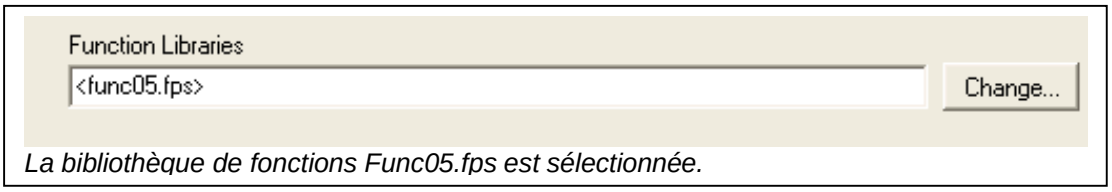


Une séquence très simple.

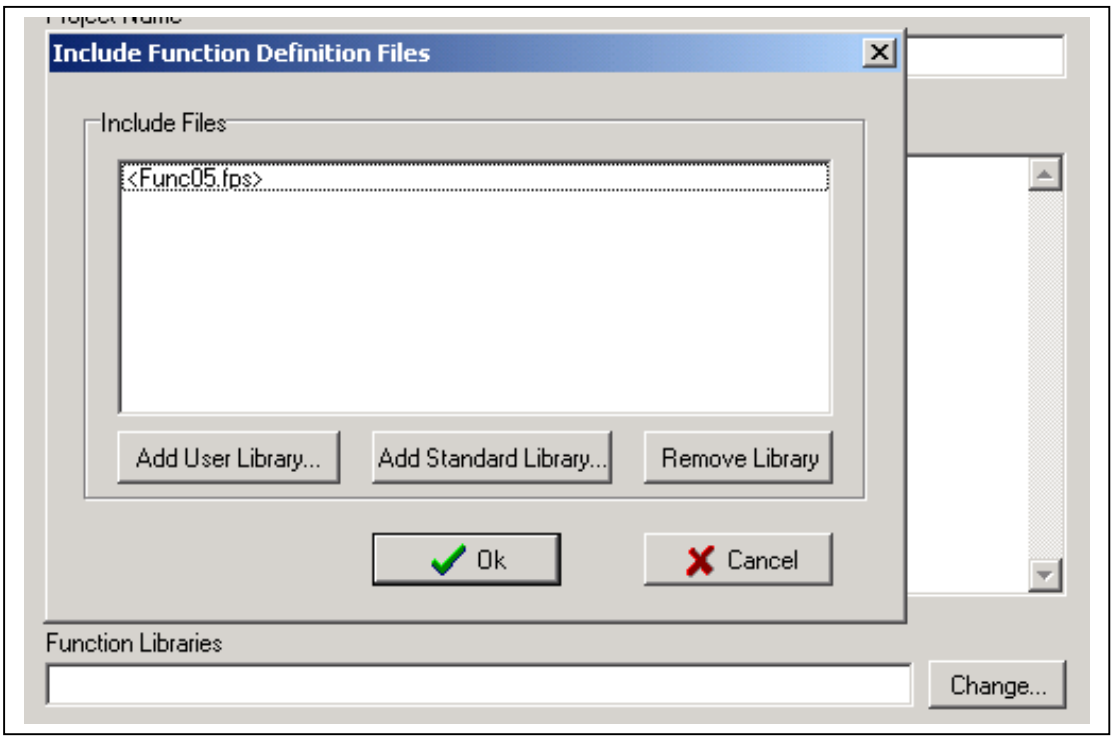
14 Paramètres du projet

14.1 Bibliothèque de fonctions

Le système Pluto permet l'utilisation d'un grand nombre de blocs de fonctions préprogrammés /macros pour des fonctions et des modules de sécurité divers. Ces blocs de fonction sont enregistrés dans une bibliothèque de fonctions à part dont l'extension fichier est « .fps ». La bibliothèque standard est fournie par Jokab. L'utilisateur peut aussi créer ses propres bibliothèques. Plusieurs fichiers de bibliothèques peuvent être importés dans un projet.



La bibliothèque de fonctions Func05.fps est sélectionnée.

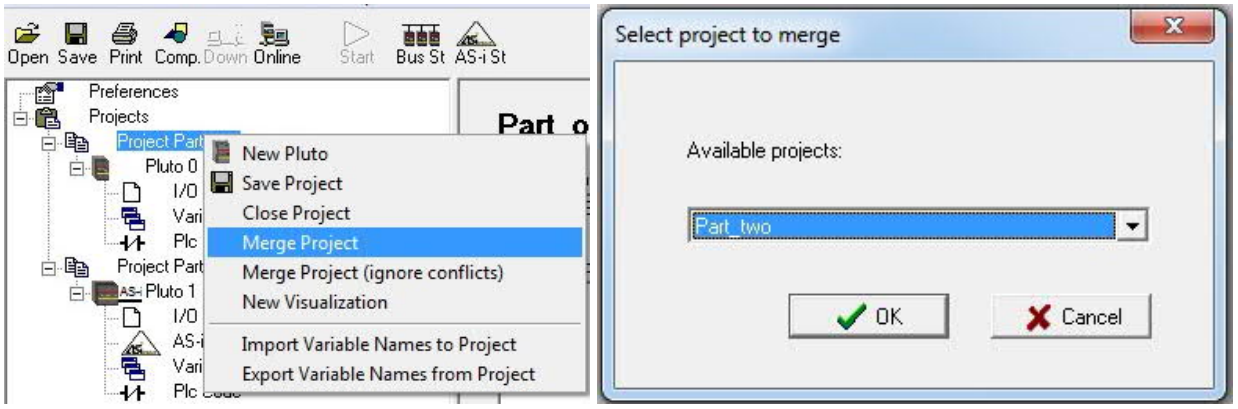


Cliquer tout d'abord sur « Function libraries » « Change », pour ouvrir une boîte de dialogue proposant les trois options suivantes :

- « Add standard Library » : Pluto Manager recherche les fichiers de bibliothèques de Jokab dans le répertoire « ..\PlutoManager\Library » où ils sont normalement sauvegardés par le programme d'installation.
- « Add User Library » : Pluto Manager recherche les fichiers dans le dossier où sont sauvegardés les fichiers de projets. Les bibliothèques utilisateur sont des fichiers contenant les blocs de fonction spécifiques à l'utilisateur.
- « Remove Library » permet de supprimer un fichier de la liste.

14.2 Fusion de projets

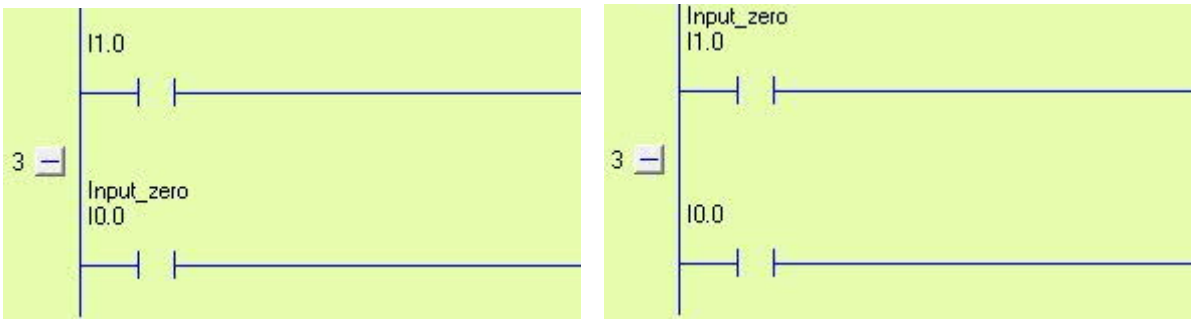
Il est possible de fusionner deux projets en un. Ouvrir les deux projets à fusionner dans Pluto Manager. Cliquer du bouton droit sur des noms de projets et sélectionner « Merge Project » (fusionner projet).



Cliquer sur « Merge Project », choisir le projet à fusionner et cliquer sur OK.

Chaque Pluto doit avoir un numéro unique et chaque nom de variable doit aussi être unique (le même nom de variable ne doit pas être utilisé dans les deux projets).

La commande « Merge Project (ignore conflicts) » (Fusionner projet (ignorer conflits)) permet d'utiliser le même nom de variable dans les deux projets mais le nom de la variable n'est affiché que dans code du Pluto où elle a été définie. Dans l'exemple ci-dessous, I0.0 et I1.0 ont toutes deux le nom "Input_zero".



I0.0 et I1.0 s'appellent toutes deux « Input_zero ». Dans le code du Pluto 0, seul le nom de I0.0 est affiché et dans le code du Pluto 1, seul le nom de I1.0.

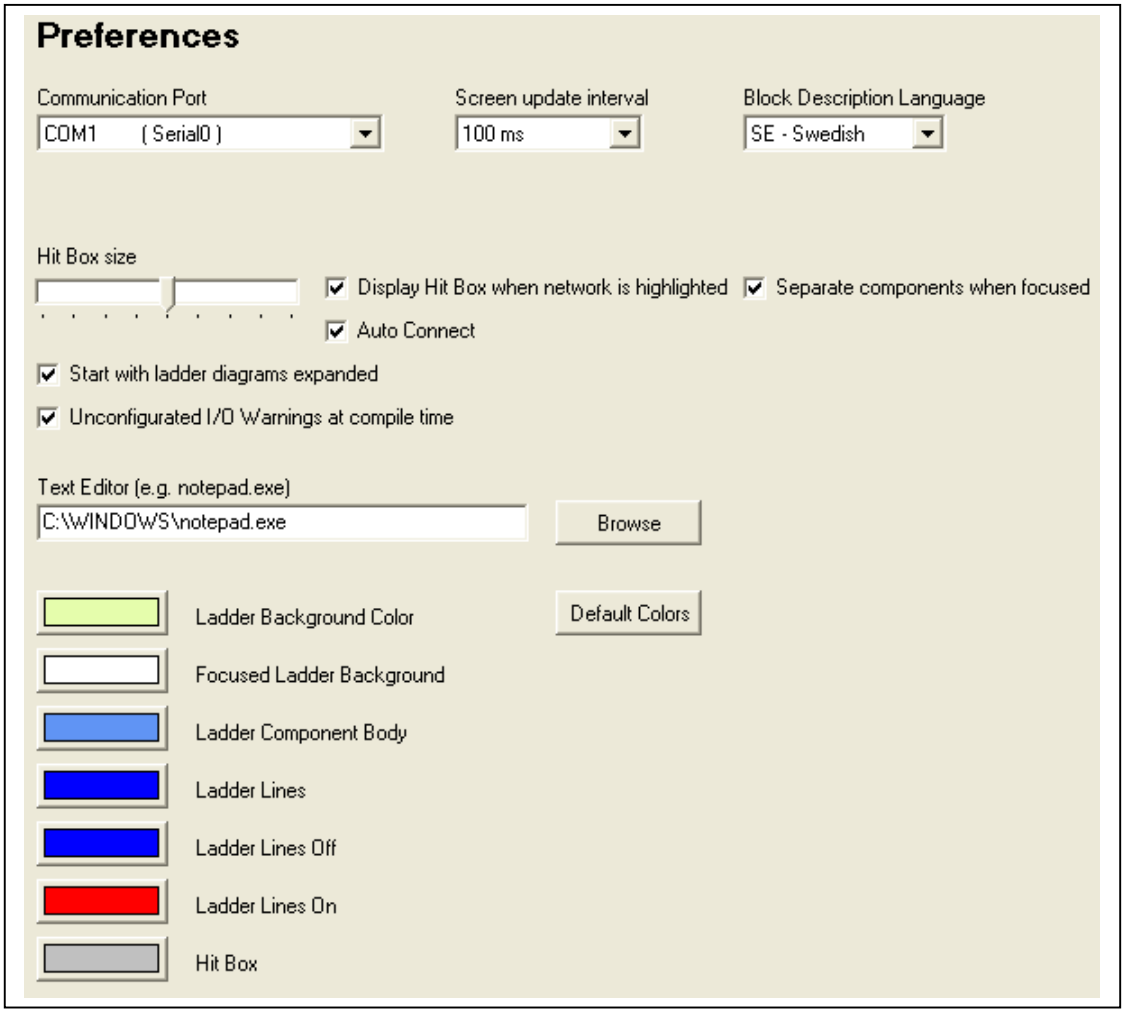
15 Compilation

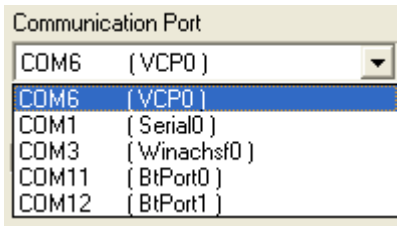
Pluto Manager enregistre le programme dans un fichier portant l'extension « .sps » devant être compilé pour pouvoir être chargé dans un Pluto. Le compilateur vérifie que le code du programme dans le fichier sps ne contient pas d'erreur de syntaxe puis produit un fichier au format hexadécimal (.hps) pouvant être transféré. Cliquer sur le bouton « Comp » pour lancer le compilateur. Une fenêtre de texte apparaît sur l'écran. Après la compilation et si tout s'est bien passé le message « 0 Error (s) detected » …… Result=OK » s'affiche. Pluto Manager prévient la plupart des erreurs de syntaxe mais pas toutes et il peut donc arriver que le compilateur émette un message d'erreur.

Remarque : Pluto Manager et le compilateur ne détectent que les erreurs de syntaxe c'est-à-dire un code contrevenant aux règles du langage de programmation. Les erreurs logiques, comme celle d'un arrêt d'urgence ne commandant pas la bonne sortie, ne peuvent pas être détectées. C'est pourquoi le programme doit être vérifié et les applications de sécurité testées minutieusement avant leur utilisation.

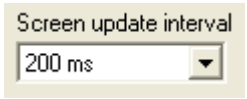
16 Préférences générales

Cette page contient les paramètres pour l'ordinateur et le programme.

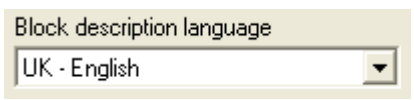




Pour communiquer via le câble USB, choisir le premier port COM « VCP » dans la liste.
Pour communiquer via le port série, choisir le port COM « Serial » dans la liste.



Intervalles de rafraîchissement de l'écran en mode En ligne. Des intervalles de rafraîchissement plus courts ralentissent l'ordinateur.



En double cliquant sur un bloc de fonction en mode édition (voir point 9), une description du bloc est affichée. La langue de description est choisie ici.

On choisit ici la taille des cases de connexion des composants et si elles doivent être affichées ou non dans le diagramme Ladder.

Lorsque « Auto Connect » est coché, les composants Ladder sont connectés automatiquement dès qu'ils sont insérés sur une ligne.

En mode édition, les composants Ladder sont séparés les uns des autres.

Par défaut, les diagrammes s'ouvrent avec le diagramme Ladder en mode déployé.

Couleurs
Les couleurs dans Pluto Manager peuvent être modifiées par l'utilisateur.

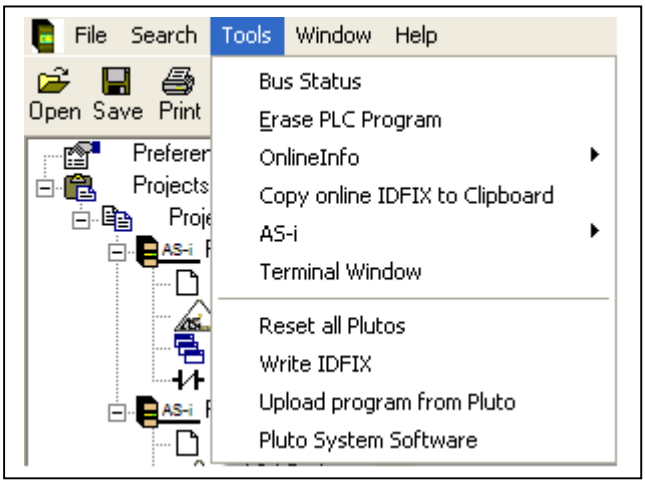
17 Fonctions en ligne

17.1 Communication

La communication avec le système se fait via le port série ou via un port USB de l'ordinateur, un câble spécial étant nécessaire dans les deux cas. Aller à la page « Preferences » et sélectionner le port COM à utiliser.

17.2 Menu « Tools »

La plupart des fonctions En ligne sont accessibles sous le menu « Tools ».



17.2.1 Erase PLC Program (Effacer le programme de l'automate/Changer de mot de passe)

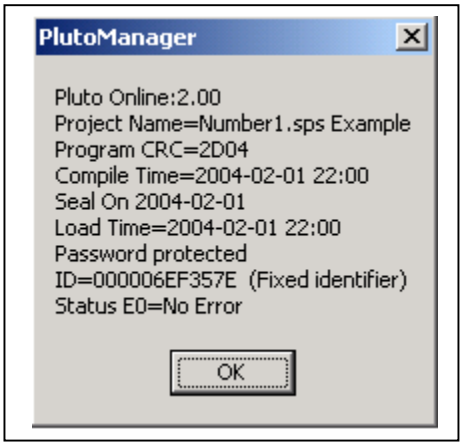
Pour effacer le programme de l'API, aller sous « Tools » → « Erase PLC program » (Mot de passe nécessaire).
Cette fonction permet aussi de changer de mot de passe. Quand le programme d'un Pluto a été supprimé, l'utilisateur peut choisir un nouveau mot de passe lors du chargement d'un nouveau programme.

17.2.2 Online info

« Tools » → « Online Info » permet de lire les informations générales d'un Pluto.

Le nom du projet et la date et l'heure de compilation sont généralement les informations recherchées §

Pour visualiser en ligne, le nom du projet doit correspondre à celui du projet ouvert dans Pluto Manager.



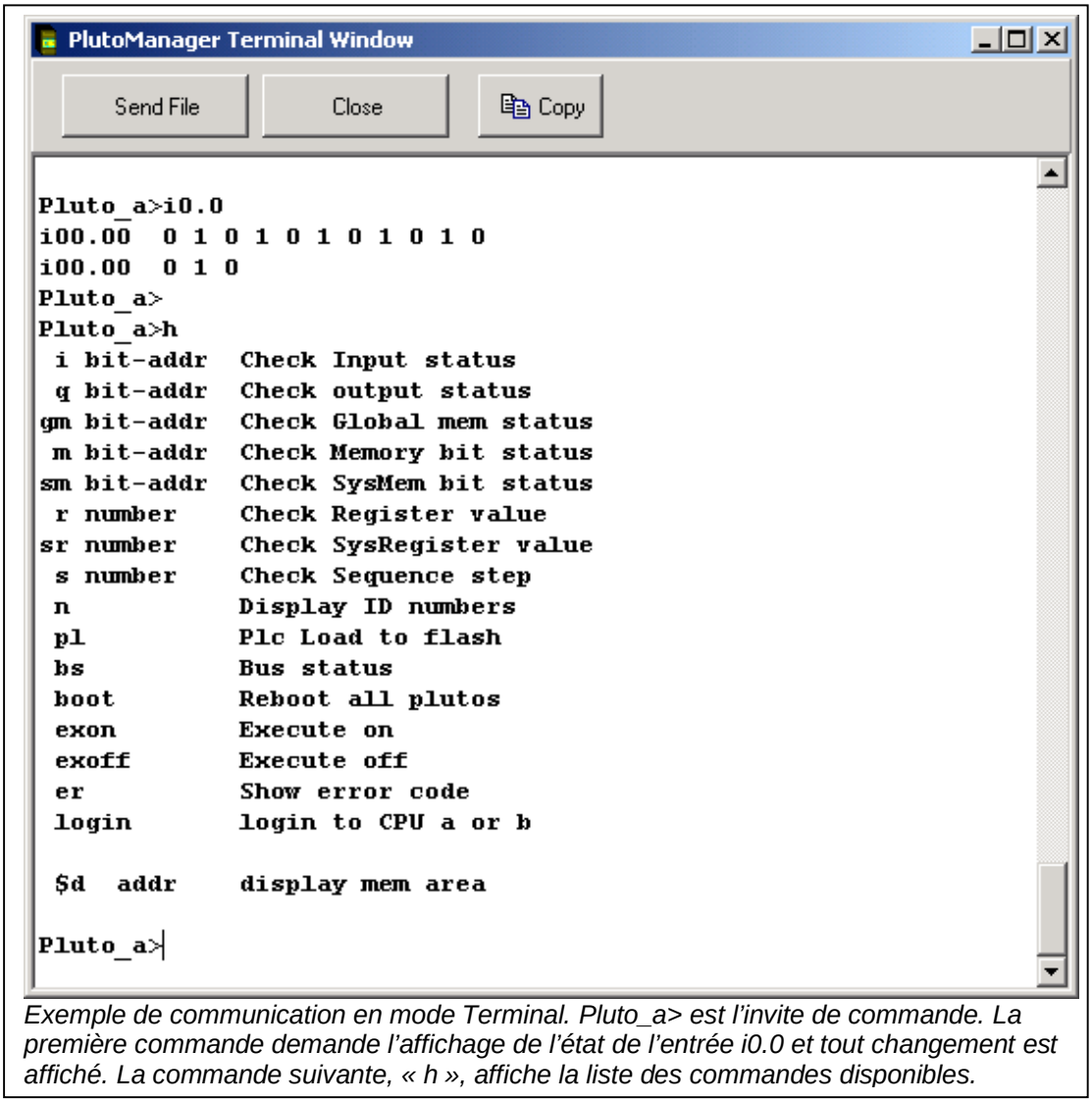
17.2.3 Copy online IDFIX to Clipboard

Le circuit d'identifiant « IDFIX » est lu et copié automatiquement dans le presse-papier. Utiliser les touches Ctrl+V pour le coller dans le champ « Identifier Number ».

17.2.4 Terminal Window (fenêtre terminal)

Une autre façon de communiquer avec un Pluto est de passer en mode Terminal. Dans ce mode, le PC est utilisé comme un simple terminal informatique. Tout ce qui est saisi au clavier est envoyé au Pluto et tout ce qui s'affiche à l'écran est envoyé par le Pluto.
De nombreuses informations sont disponibles en mode Terminal, comme les valeurs des E/S, la date de compilation, le nom du programme, etc. Il est aussi possible de transférer un nouveau programme : « pl » suivi de « yes » à la question puis cliquer sur « Send File ». Saisir « h » (Help/Aide) pour afficher toutes les commandes disponibles.

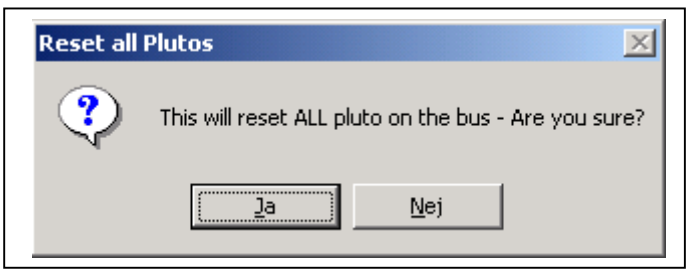
Un logiciel de terminal standard, comme HyperTerm dans Windows, peut être utilisé à la place de la fenêtre Terminal dans Pluto Manager.



Exemple de communication en mode Terminal. Pluto_a> est l'invite de commande. La première commande demande l'affichage de l'état de l'entrée i0.0 et tout changement est affiché. La commande suivante, « h », affiche la liste des commandes disponibles.

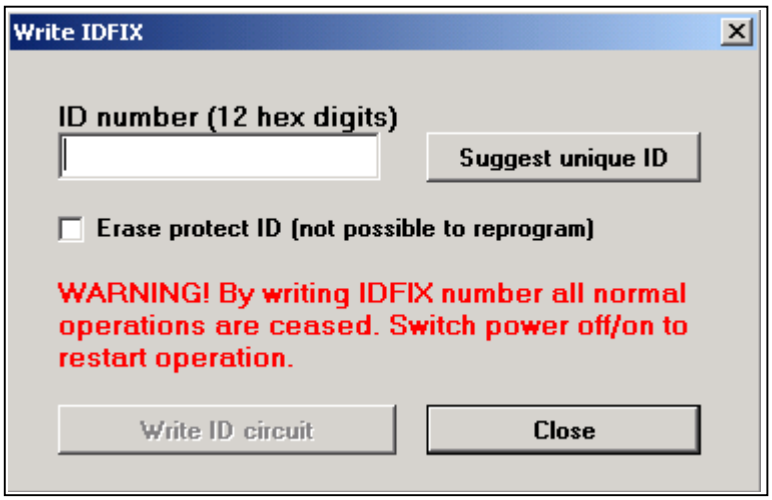
17.2.5 Reset all Plutos

Cette instruction redémarre tous les Pluto connectés au bus. Elle a la même fonction qu'une mise sous et hors tension. Elle est utilisée après modifications de la vitesse de transmission sur le bus ou réinitialisation de certaines erreurs par ex.



17.2.6 Write IDFIX

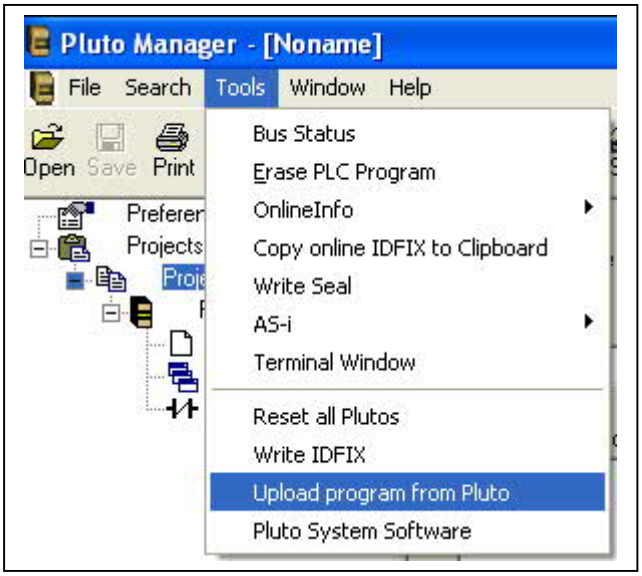
Fonction pour des circuits d'identifiants programmables « IDFIX ».
Il est possible de saisir le numéro manuellement, pour copier un numéro déjà existant par ex., ou de laisser le système proposer un numéro.
La sélection de « Erase protected ID » interdit toute modification ultérieure du circuit.



Noter que pour revenir à un fonctionnement normal après la programmation d'un IDFIX, Pluto doit être réinitialisé (mis hors et sous tension).

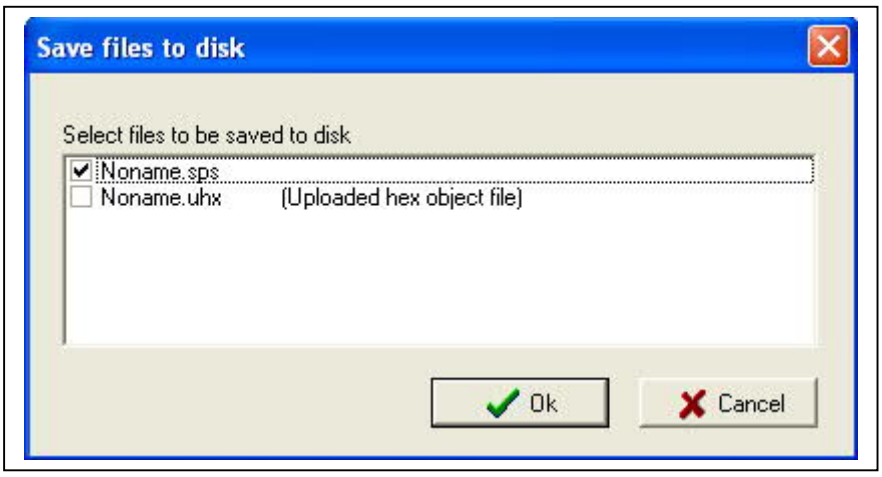
17.2.7 Upload Program from Pluto

Le programme de l'automate peut être chargé sur un PC depuis le Pluto sous forme de fichier .uhx. Si « Include source code in compiled file » a été coché lors du transfert du programme dans le Pluto (voir 3.3 Inclure le fichier source), le fichier source (.sps) peut aussi être récupéré et sauvegardé. Choisir « Upload program from Pluto » (charger le programme du Pluto) dans le menu Tools.



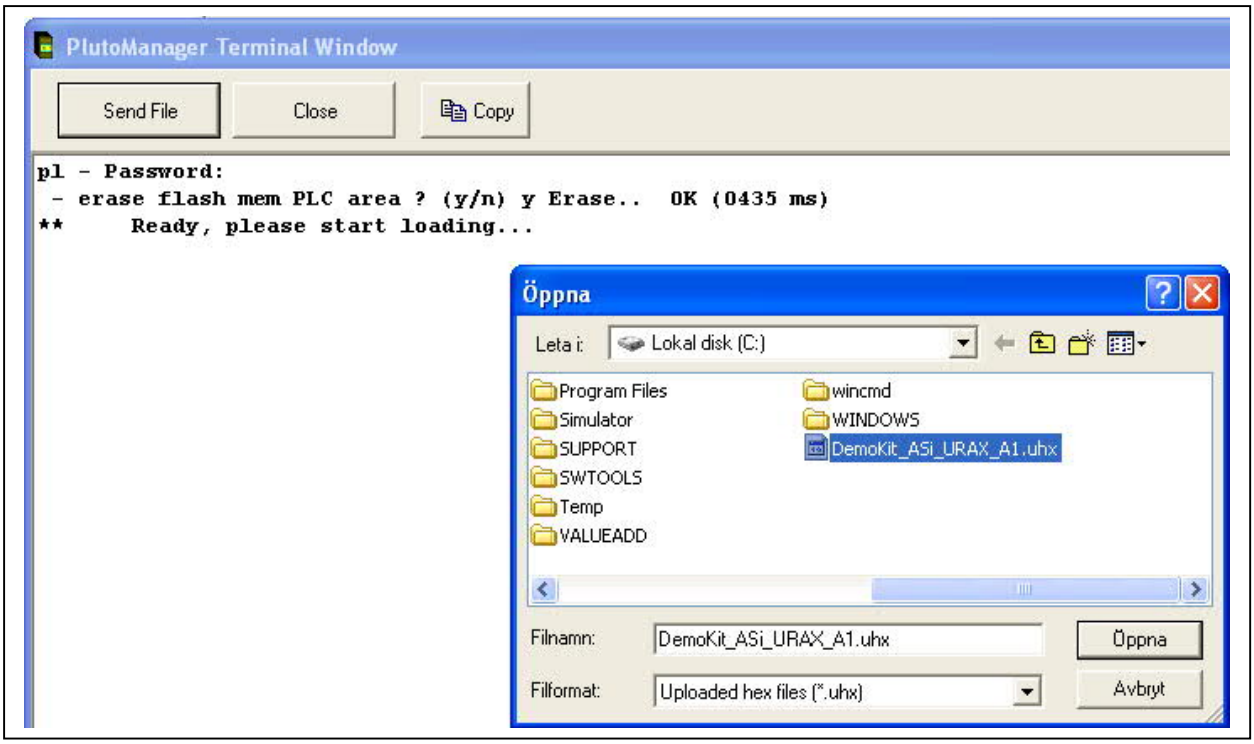
Le mot de passe demandé est le même que celui utilisé pour le transfert du programme.

Si « Include source code in compiled file » a été coché lors du transfert du programme dans le Pluto (voir 3.3), le fichier source (.sps) et le fichier compilé (.uhx) peuvent être récupérés.



Une fois sélectionnés, les fichiers peuvent être sauvegardés sur le PC.

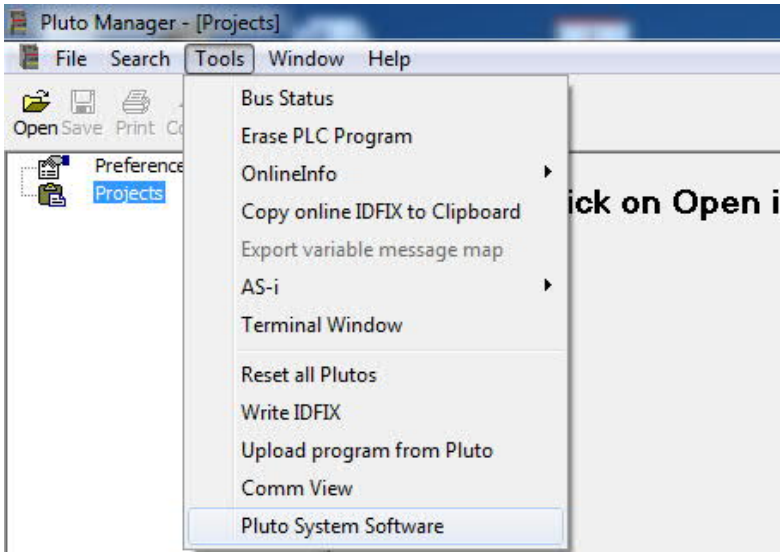
Le fichier obtient une extension .uhx et peut être transféré dans un autre Pluto à l'aide de la commande « pl » dans la fenêtre Terminal. Écrire « pl » et indiquer le mot de passe. Répondre « y » (oui) à la question « erase flash mem PLC area? » (effacer mémoire flash de l'automate ?) Quand « Ready, please start loading…" (Prêt, démarrer le chargement…) s'affiche, cliquer sur « Send File » et sélectionner le fichier .uhx souhaité.



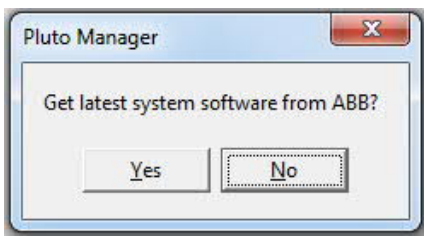
17.2.8 Pluto System Software (Mise à jour des fichiers du système d'exploitation sur Pluto)

Le système d'exploitation peut être téléchargé manuellement depuis la page www.abb.com/jokabsafety ou automatiquement directement de ABB via Pluto Manager.

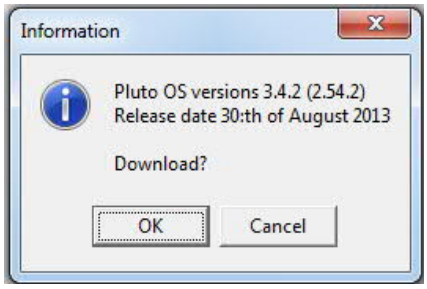
1. Branchez Pluto au PC avec le câble de programmation Pluto.
2. Allez au menu « Tools » dans Pluto Manager et sélectionnez « Pluto System Software ».



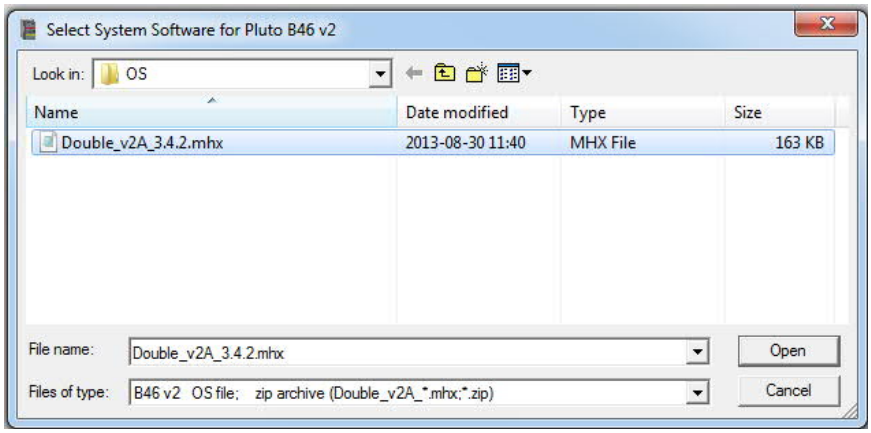
Pour télécharger automatiquement la dernière version directement de la page web d'ABB, cliquez sur « Yes ».
Pour charger le système d'exploitation depuis votre ordinateur local, sélectionnez « No ».



3. Si « Yes » a été sélectionné : La version la plus récente du modèle Pluto connectée est affichée. (Si « No » a été sélectionné, cette boîte de dialogue ne s'affiche pas.)



Pour télécharger, cliquez sur « OK ».



Sélectionnez le système d'exploitation et cliquez sur « Open ». Le téléchargement commence. Si « No » a été sélectionné précédemment (pour ne pas télécharger directement de ABB), localisez le fichier du système d'exploitation sur l'ordinateur local et cliquez sur « Open ».

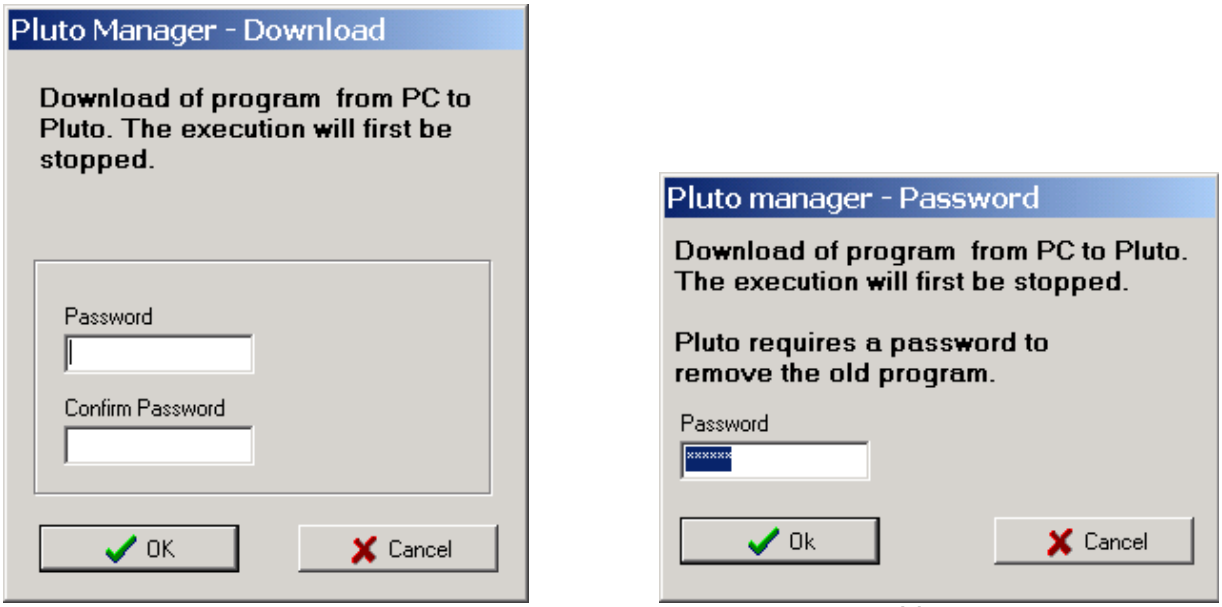
Pour confirmer et redémarrer Pluto, cliquez sur « Yes ».

17.3 Transfert d'un programme

Appuyer sur le bouton « Down » dans la liste d'outils pour transférer un programme du PC vers un Pluto. Noter que le programme doit avoir été compilé avant son transfert. Un message d'erreur s'affiche s'il n'a pas été compilé. Voir Compilation.

Après l'appui sur le bouton« Down », une boîte de dialogue demandant un mot de passe s'ouvre. Ce mot de passe doit comporter de 4 à 6 caractères.
Si la communication échoue, le message d'erreur suivant s'affiche :

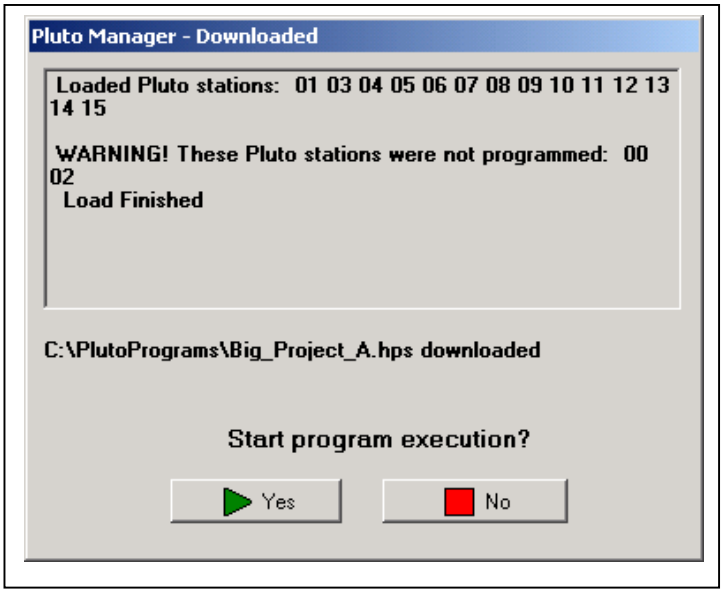
« Couldn't establish connection… » - Impossible d'établir une connexion.
« Connection time out » - communication interrompue.



Une de ces boîtes de dialogue s'affiche en fonction de si le Pluto contient déjà un programme avec un mot de passe.

Si tout se passe bien, un message indique que le fichier a été transféré et propose de l'exécuter. Si « No » est choisi, l'exécution du programme pourra être lancée ultérieurement en appuyant sur le bouton « Start » de la liste des outils.

Un avertissement est donné lorsque le projet comprend plusieurs Pluto et que tous ne sont pas connectés au bus.



17.4 Insertion d’un nouveau Pluto dans un projet existant

Lors du chargement d’un programme dans plusieurs Pluto, ceux-ci se contrôlent les uns les autres pour vérifier que tous possèdent exactement la même version du programme. Si ce n’est pas le cas, ils refusent les E/S des autres.

Si un Pluto compris dans un projet est connecté au bus après les autres, plusieurs scénarios sont possibles en fonction du programme chargé dans ce Pluto :

Scénario 1 : le nouveau Pluto est vide (l’afficheur indique Er20) et équipé de l’identifiant IDFIX attendu. Le programme peut être transféré dans ce Pluto depuis un PC connecté à n’importe quel Pluto du même projet.
Le nouveau Pluto peut aussi être autoprogrammé sans PC. Pour ce faire, appuyer pendant 2 secondes sur le bouton « K » en face avant du Pluto. Un « L » (Load) se met alors à clignoter sur l’afficheur indiquant que l’autoprogrammation peut commencer. Appuyer de nouveau sur « K » pour commencer le transfert du programme indiqué par un « L » continu sur l’afficheur.

Scénario 2 – Le nouveau Pluto a l’IDFIX attendu mais la mauvaise version de programme.
À la connexion, tous les Pluto du projet affichent le code d'erreur Er27 car ils détectent un Pluto appartenant au même projet ayant un programme de version différente. Chaque Pluto exécute son programme d’API sans tenir compte des E/S des Pluto ayant un programme différent.
Il suffit de transférer le programme souhaité dans un des Pluto pour que tous soient mis à jour avec la même version de programme.

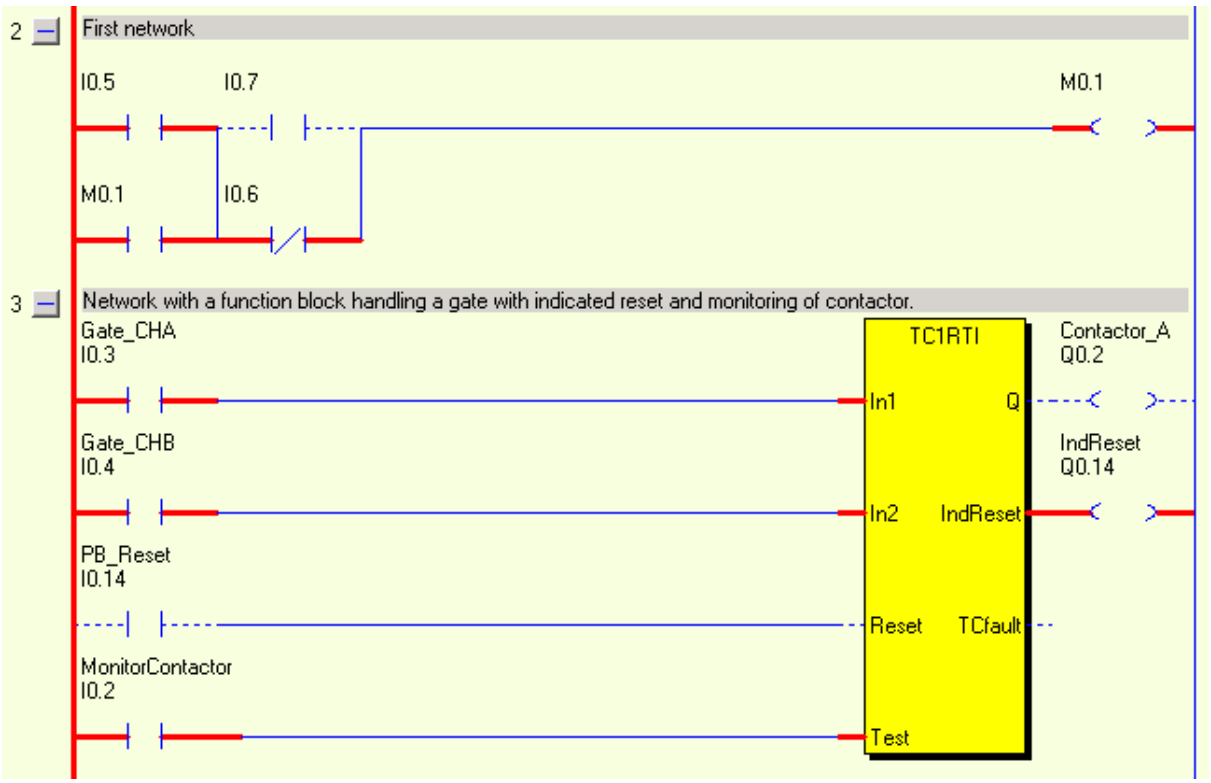
17.5 Modification de la vitesse de transmission sur le bus (baud rate), erreur Er26

La vitesse de transmission sur le bus ne peut pas être modifiée en cours de fonctionnement. Si un programme est transféré avec une nouvelle vitesse (baudrate), le Pluto continue avec l'ancienne vitesse et indique Er26. Er26 signifie qu’un Pluto fonctionne avec une autre vitesse/baud rate que celle pour laquelle il est programmé.
Pour changer de vitesse, redémarrer le Pluto, soit en le mettant hors et sous tension, ou via « Tools » → « Reset all Plutos » dans Pluto Manager.

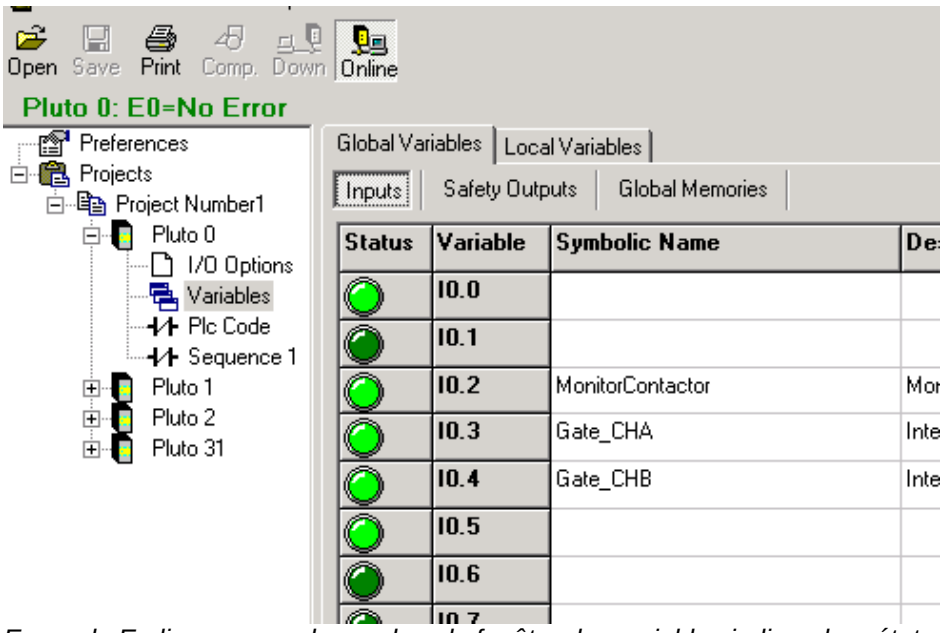
Même un Pluto vide (Er20) seul dans lequel un programme est transféré affiche Er26 et doit être redémarré pour adopter la vitesse de transmission programmée.

17.6 Online

Ce bouton de la barre d'outils permet de passer en mode En ligne et de le quitter.
Le mode En ligne sert à visualiser l'état des variables dans la fenêtre des variables ou le programme Ladder.



En mode En ligne, des lignes rouges dans le diagramme Ladder indiquent quels composants sont activés.



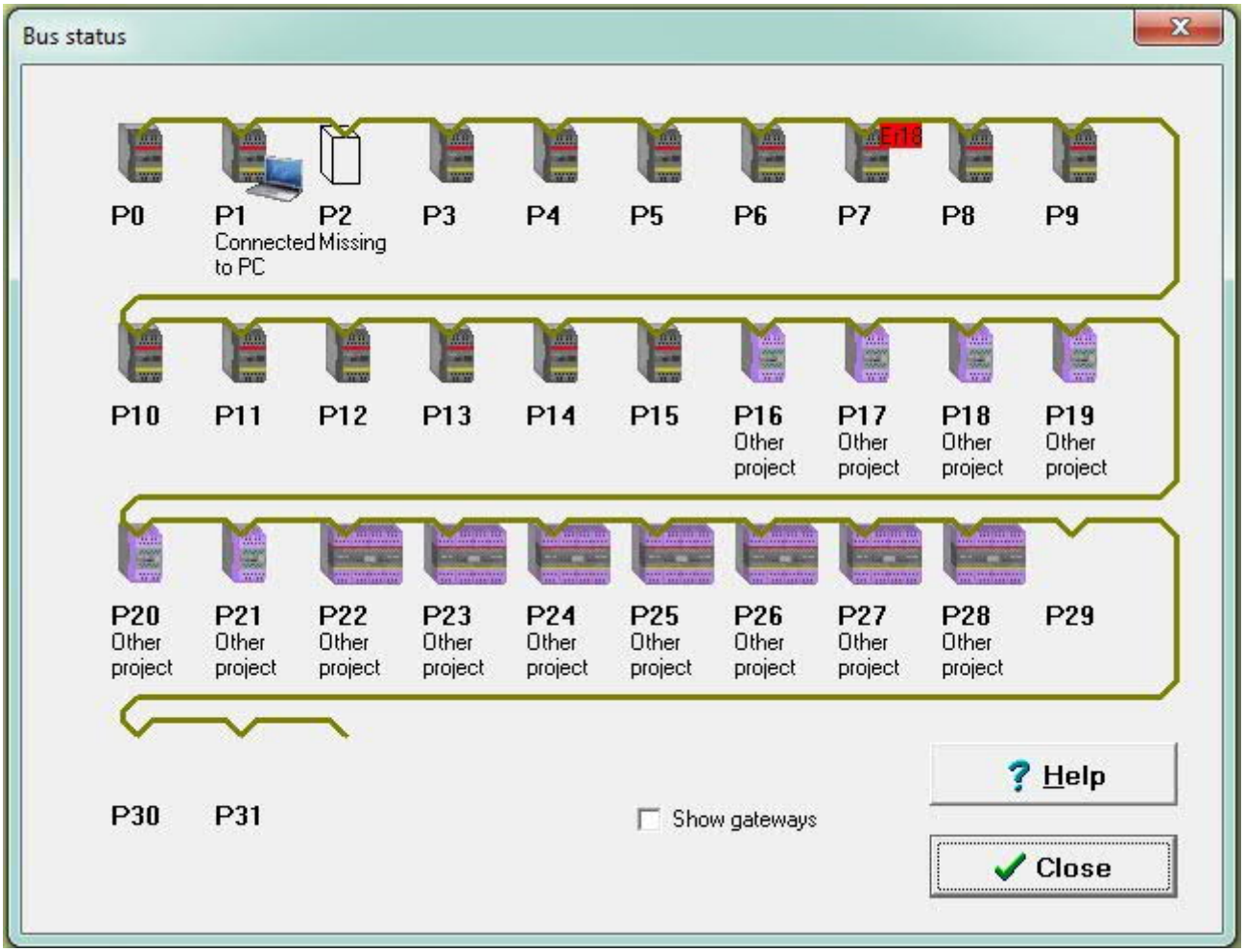
En mode En ligne, une colonne dans la fenêtre des variables indique leur état.
Les codes d'erreurs des Pluto sont indiqués en vert sous la barre d'outils.

Démarrage et arrêt de l'exécution du programme

L'exécution du programme peut être démarrée ou arrêtée en mode En ligne.

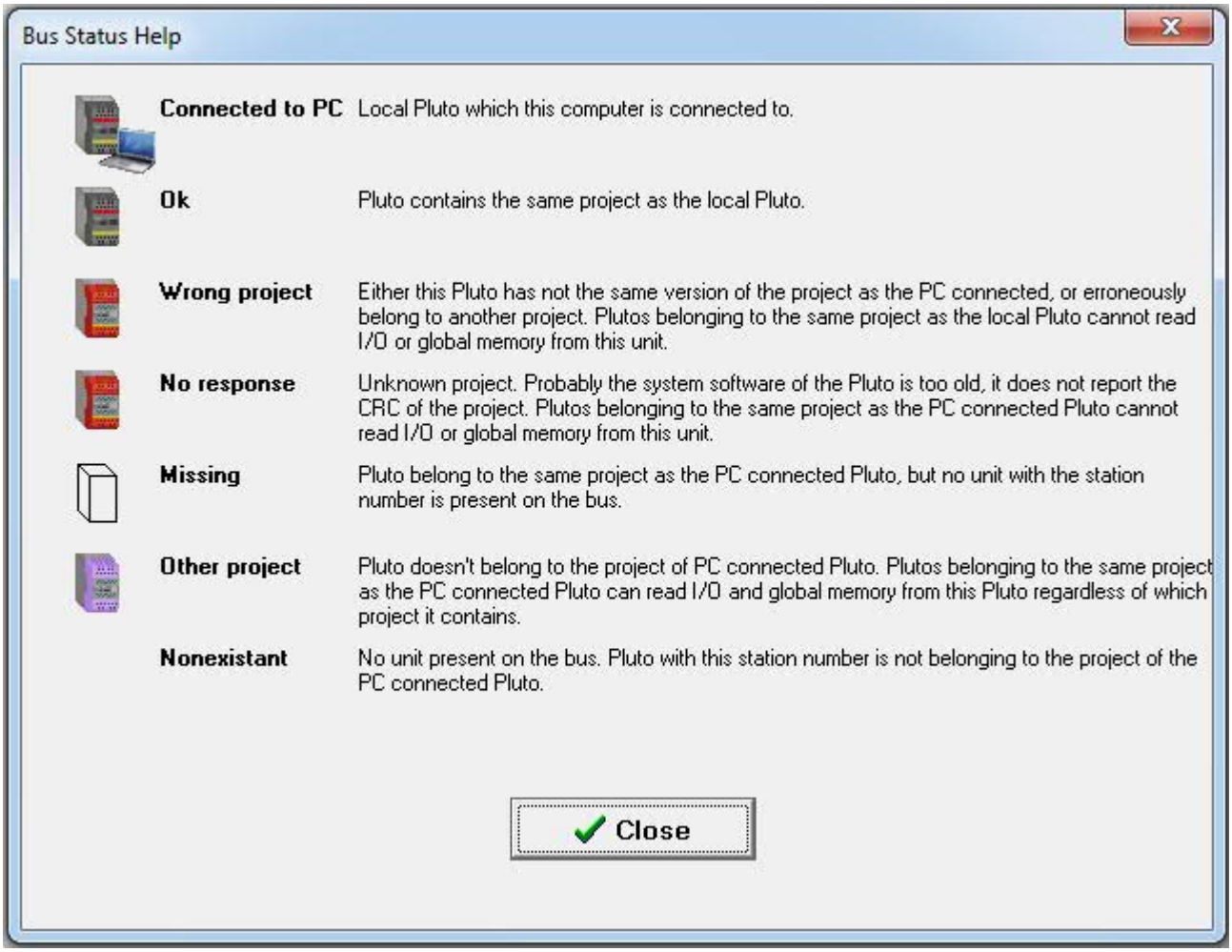
Bus Status (État du bus)

Le mode En ligne permet de visualiser les Pluto connectés au bus via « Tools » → « Bus Status »



L'illustration représente un bus avec 29 Pluto appartenant à au moins deux fichiers de projets différents. Les Pluto 0 à 15 appartiennent au même projet. Le Pluto 1 est connecté au PC. Le Pluto 2 appartient au projet mais n'est pas connecté au bus. Le Pluto 7 présente un code d'erreur. Les Pluto 1 à 28 sont connectés au bus mais n'appartiennent pas au même projet que le Pluto connecté au PC.

Explication de « Bus Status »



Le bouton Aide ouvre la fenêtre d'explications ci-dessus.

Connected to PC Pluto local auquel le PC est connecté.

OK Ce Pluto contient le même projet que le Pluto connecté au PC.

Wrong project Ce Pluto n'a pas la même version de projet que le Pluto connecté au PC ou appartient à un autre projet alors qu'il ne devrait pas. Les Pluto appartenant au même projet que le Pluto connecté au PC ne peuvent lire ni les E/S, ni les mémoires globales de ce Pluto.

No response Projet inconnu. Le système d'exploitation de ce Pluto est probablement trop vieux et il ne communique pas le CRC de ce projet. Les Pluto appartenant au même projet que le Pluto connecté au PC ne peuvent lire ni les E/S, ni les mémoires globales de ce Pluto.

Missing Pluto appartenant au même projet que le Pluto connecté au PC mais aucun Pluto ayant ce numéro n'est connecté au bus.

Other project Ce Pluto n'appartient au projet du Pluto connecté au PC. Les Pluto appartenant au même projet que le Pluto connecté au PC peuvent lire les E/S et les mémoires globales de ce Pluto quel que soit le projet qu'il contient.

Nonexistant Aucun Pluto possédant ce numéro n'est présent sur le bus.

17.7 Sceau

La boîte de dialogue « Online info » contient le texte « Seal On » ou « Seal Off ».
Après le transfert d'un programme, « Seal off » est affiché et indique que le programme est modifié et non approuvé. La fonction du sceau ne sert qu'à indiquer que le programme a été modifié et il n'a aucune influence sur la fonction.

Via le code de licence de Pluto Manager, trois possibilités sont disponibles :
Option 1 : la fonction Sceau n'est pas disponible.
Option 2 : le sceau peut être chargé séparément après le transfert du programme.
Option 3 : le sceau est chargé automatiquement lors du transfert du programme.
Une entreprise peut donc créer un système permettant à certaines personnes autorisées de contrôler les programmes et de les approuver en chargeant le sceau.

Pour écrire un sceau :
« Tool » → « Write Seal » → Le message « Seal written » indique que le sceau a bien été écrit.

Deuxième partie
Langage de programmation

Remarque : *les instructions et fonctions en italique sont disponibles avec le jeu d'instructions nº 3 seulement. (Voir 3.6.1 Jeux d'instructions nº 2 et nº3 dans la première partie de ce manuel.)*

1 Instructions sur les bits

1.1 Adressage des opérandes sur les bits

Dans le langage de programmation de Pluto, les E/S et les mémoires sont adressées ainsi : [*type d'E/S*][n° de Pluto].[*n° d'E/S*].
Un maximum de 32 Pluto, numérotés de 0 à 31, peuvent être connectés au même bus.

Le tableau ci-dessous illustre le principe d'adressage des Pluto de la famille A20.

Type d'E/S:	Dénomination Pluto 0	Dénomination Pluto 1	……	Dénomination Pluto 31
Entrées	I0.0 I0.1 . . I0.17	I1.0 I1.1 . . I1.17	… … … … … … … … … …	I31.0 I31.1 . . I31.17
Sorties	Q0.0 Q0.1 . . Q0.17	Q1.0 Q1.1 . . Q1.17	…… … … … … … … … …	Q31.0 Q31.1 . . Q31.17
Mémoires	M0.0 M0.1 . . M0.599	M1.0 M1.1 . . M1.599	…… … … … … … … … …	M31.0 M31.1 . . M31.599
Mémoires globales	GM0.0 GM0.1 . . GM0.11	GM1.0 GM1.1 . . GM1.11	…… … … … … … … … …	GM31.0 GM31.1 . . GM31.11
*Bits de registres**	*R0.0.0… R0.0.15* *R0.1.0… R0.1.15* . . *R0.149.0… R0.149.15*	*R1.0.0… R1.0.15* *R1.1.0… R1.1.15* . . *R1.149.0…R1.149.15*	…… … … … … … … … …	*R31.0.0… R31.0.15* *R31.1.0… R31.1.15* . . *R31.149.0… R31.149.15*

**Jeu d'instructions nº 3 seulement. Les bits de registres peuvent être adressés individuellement comme R<Pluto>.<reg>.<bit>.*

Exemple :
Q10.1 ⇔ Adressage de la sortie 1 du Pluto n°10

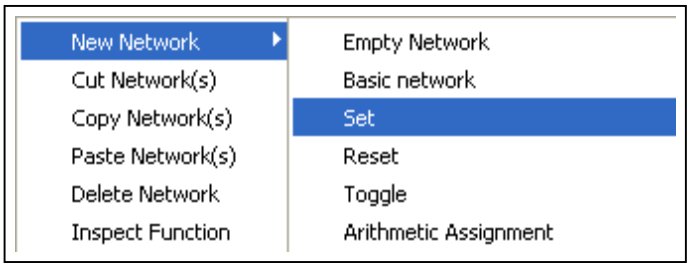
Il est aussi possible d'écrire ainsi : Q10.01, Q10.0001 ou Q10.1

Le tableau ci-dessous illustre le principe d'adressage des entrées et sorties d'un esclave du Pluto AS-i. (Décrit plus en détail au chapitre 7 Fonctions du bus AS-i.)

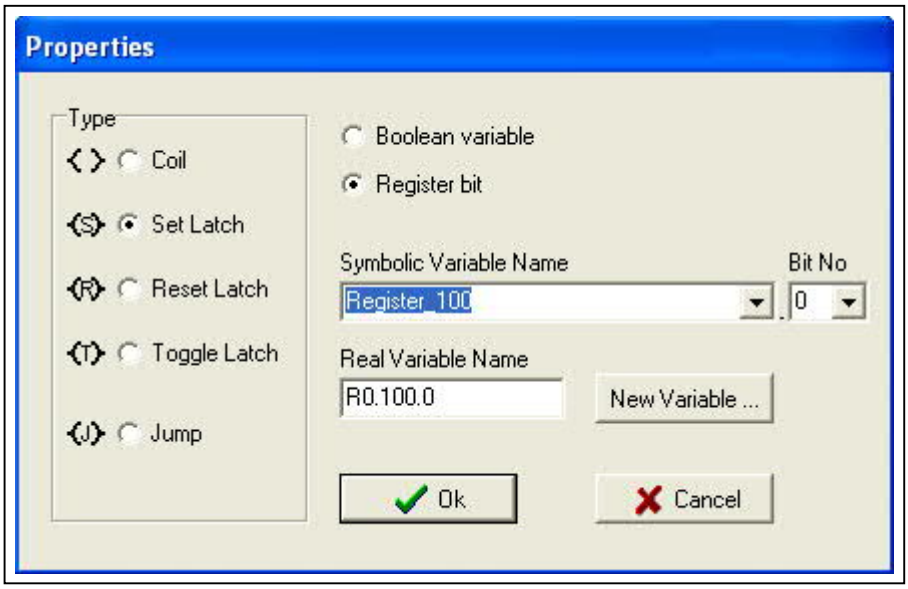
Type d'E/S:	Dénomination Pluto 0	Dénomination Pluto 1	……	Dénomination Pluto 31
Entrées AS-i (de sécurité)	ASi0.1 ASi0.2 . . ASi0.31	ASi1.1 ASi1.2 . . ASi1.31	… … … … … … … … … …	ASi31.1 ASi31.2 . . ASi31.31
Entrées AS-i Esclaves standard non sécurisés	ASi0.1.1... ASi0.1.4 ASi0.2.1... ASi0.2.4 . . ASi0.31.1... ASi0.31.4	ASi1.1.1... ASi1.1.4 ASi1.2.1... ASi1.2.4 . . ASi1.31.1... ASi1.31.4	… … … … … … … … … …	ASi31.1.1... ASi31.1.4 ASi31.2.1... ASi31.2.4 . . ASi31.31.1... ASi31.31.4
Entrées AS-i Esclaves B Non sécurisés	ASi0.1B.1... ASi0.1B.4 ASi0.2B.1... ASi0.2B.4 . . ASi0.31B.1... ASi0.31B.4	ASi1.1B.1... ASi1.1B.4 ASi1.2B.1... ASi1.2B.4 . . ASi1.31B.1... ASi1.31B.4	… … … … … … … … … …	ASi31.1B.1... ASi31.1B.4 ASi31.2B.1... ASi31.2B.4 . . ASi31.31B.1... ASi31.31B.4
Sorties AS-i Esclaves standards non sécurisés	ASq0.1.1... ASq0.1.4 ASq0.2.1... ASq0.2.4 . . ASq0.31.1... ASq0.31.4	ASq1.1.1... ASq1.1.4 ASq1.2.1... ASq1.2.4 . . ASq1.31.1... Asq1.31.4	… … … … … … … … … …	ASq31.1.1... ASq31.1.4 ASq31.2.1... ASq31.2.4 . . ASq31.31.1... Asq31.31.4
Sorties AS-i Esclaves B non sécurisés	ASq0.1B.1... ASq0.1B.4 ASq0.2B.1... ASq0.2B.4 . . ASq0.31B.1... ASq0.31B.4	ASq1.1B.1... ASq1.1B.4 ASq1.2B.1... ASq1.2B.4 . . ASq1.31B.1... Asq1.31B.4	… … … … … … … … … …	ASq31.1B.1... ASq31.1B.4 ASq31.2B.1... ASq31.2B.4 . . ASq31.31B.1... Asq31.31B.4

1.2 Bits de registres (jeu d'instructions nº 3 seulement)

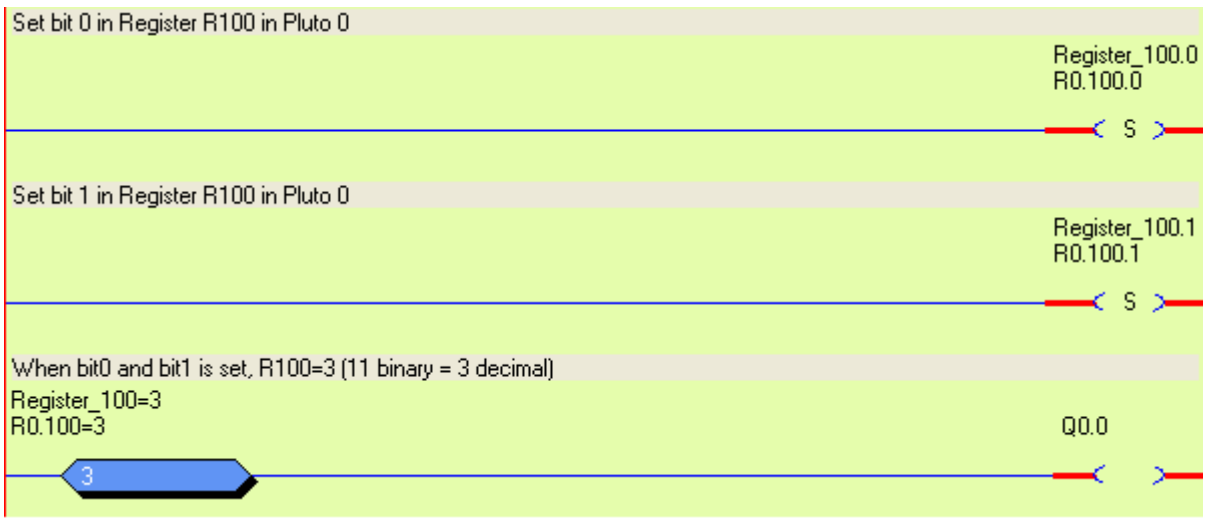
Le jeu d'instructions nº 3 permet des opérations sur les bits des registres. Pour mettre à 1 un bit de registre, choisir « New Network » et « Set ».



Choisir « Register bit », choisir le registre et le numéro de bit et cliquer sur Ok.



<u>Exemple:</u>



Dans cet exemple, les bits 0 et 1 du registre R100 du Pluto 0 sont mis à 1. La valeur de R100 devient 3 qui correspond à la valeur binaire 11 (les deux bits les moins significatifs sont à 1).

1.3 Instructions booléennes

Le langage de programmation Pluto suit les règles de programmation Ladder selon IEC 1131-3 si Pluto Manager est utilisé.

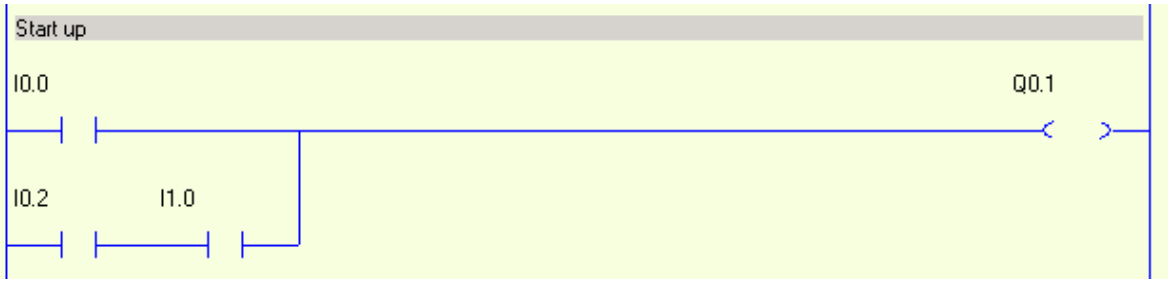
En cas de programmation en format texte avec un éditeur de texte, le langage de programmation suit les lois booléennes et utilise AND, OR, NOT et les commandes d'assignation.

Syntaxe de programme en format texte :

Instruction	Syntaxe de programme
AND	*
OR	+
NOT	/
Assignation	=

Exemple :

En Ladder :



Texte équivalent :

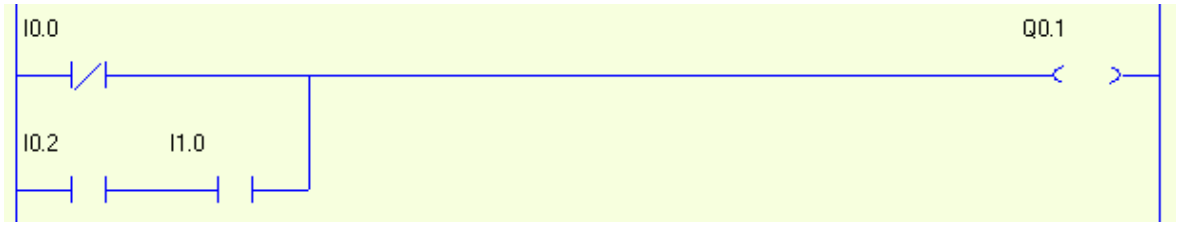
```
Q0.1 = I0.0 + I0.2 * I1.0              ; Start up
```

Le point virgule définit le début d'un commentaire de programme.

Explication : la sortie Q0.1 est activée quand l'entrée I0.0 ou les deux entrées I0.2 et I1.0 sont activées (« 1 »).

Exemple de négation

En Ladder :



Texte équivalent :

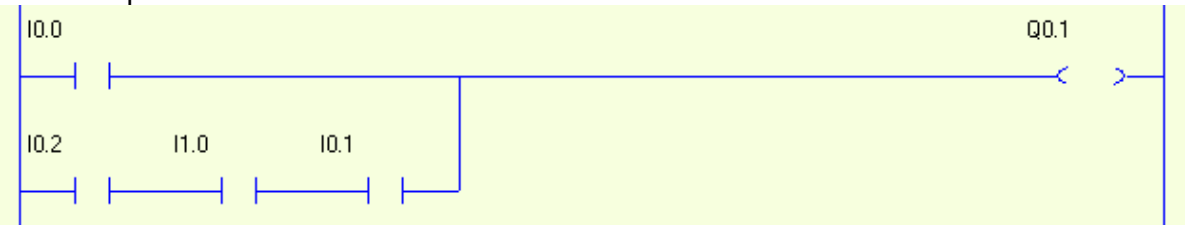
```
Q0.1 = /I0.0 + I0.2*I1.0
```

Selon les lois booléennes, les instructions AND (*) sont exécutées avant les instructions OR (+). L'ordre d'exécution peut être modifié à l'aide de parenthèses.

Exemple :

```
Q0.1 = I0.0 + I0.2*I1.0*I0.1
```

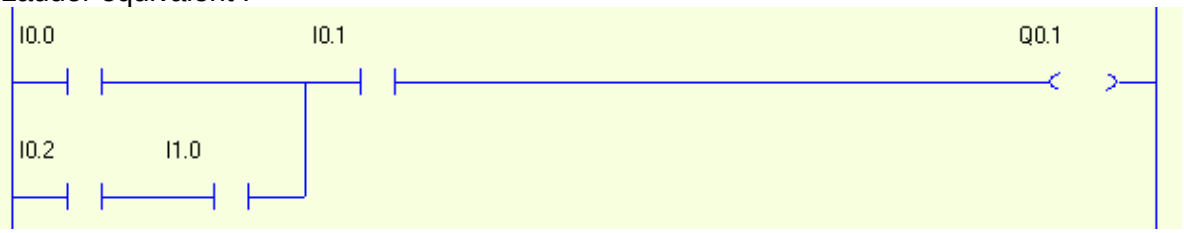
Ladder équivalent :



Exemple avec des parenthèses

```
Q0.1 = (I0.0 + I0.2*I1.0)*I0.1
```

Ladder équivalent :

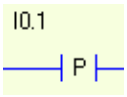


Remarque : en programmation texte, les espaces n'ont aucune influence.

1.4 Détection de front

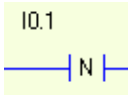
La détection de front peut être utilisée avec certains opérandes. Cette fonction permet de détecter les fronts montants (positifs) et descendants (négatifs). La syntaxe de programme est la suivante :

Front montant :



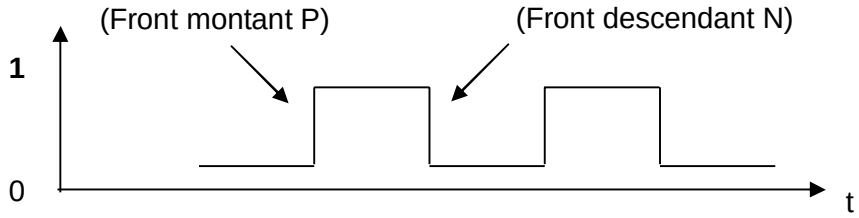
⇔ P(I0.1)

Front descendant :

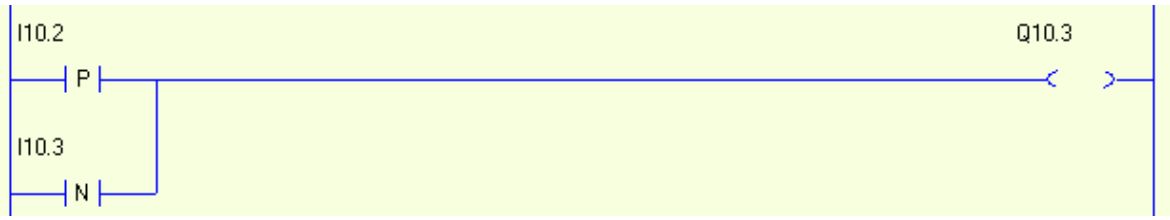


⇔ P(/I0.1)

Fonction : quand un front est détecté, le composant correspondant passe à un « 1 » logique durant un cycle de programme.



Exemple :

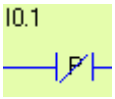


`Q10.3 = P(I10.2) * P(/I10.3)` ⇔ La sortie Q10.3 est activée pendant un temps de cycle quand un front montant est détecté sur l'entrée I10.2 ou quand un front descendant est détecté sur l'entrée I10.3

1.4.1 Détection de front inversée (jeu d'instructions nº 3 seulement)

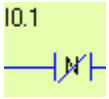
Le résultat de cette fonction est l'inverse de celui de la fonction détection de front normale : sa valeur est normalement « 1 » et elle passe à « 0 » durant la durée d'un cycle quand un front est détecté.

Pas de front positif :



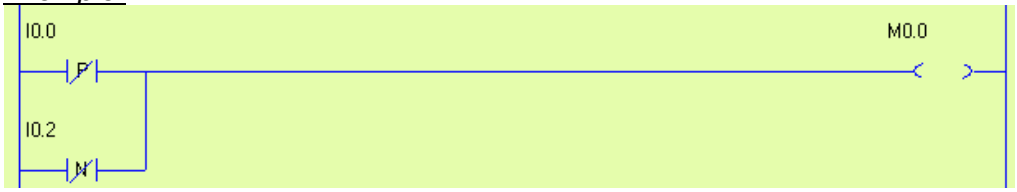
⇔ */P(I0.1)*

Pas de front négatif :



⇔ */P(/I0.1)*

Exemple:



M0.0 est normalement au niveau haut (« 1 »). En cas de front positif sur I0.0 ou de front négatif sur I0.2, M0.0 passe à « 0 » durant un cycle.

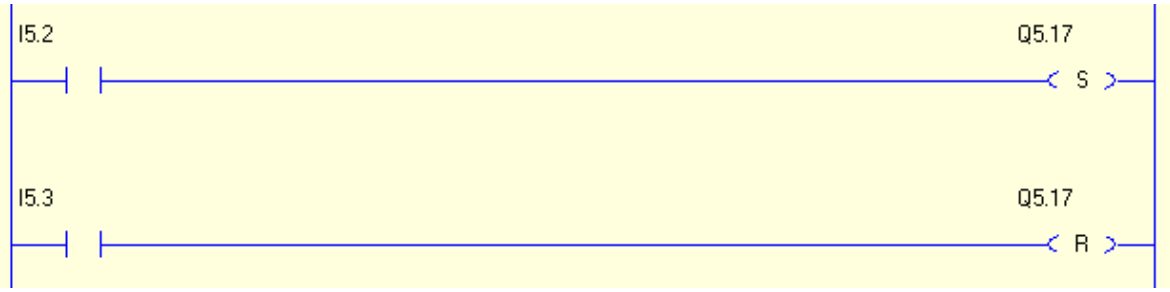
1.5 Fonctions SET et RESET

Les fonctions SET et RESET activent et désactivent une sortie ou une mémoire. La nouvelle valeur se maintient jusqu'à la prochaine activation/désactivation.

Fonction de maintien	Syntaxe du programme
SET/Latch on	S(Q0.1)
RESET/Latch off	R(Q0.1)

Quand une variable est activée avec l'instruction SET, cette variable reste activée quand la condition change d'état. La variable est désactivée à l'aide de l'instruction RESET.

Exemple :



Texte équivalent :

```
S(Q5.17) = I5.2
R(Q5.17) = I5.3
```

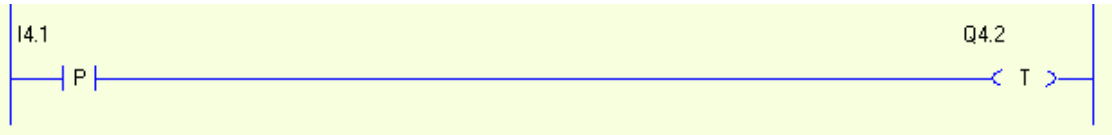
Fonction : la sortie 17 du PLUTO 5 est activée quand l'entrée 2 du PLUTO 5 est activée. La sortie reste activée jusqu'à ce que l'entrée 3 du Pluto 5 soit activée.

1.6 Fonction de basculement (Toggle)

La fonction Toggle inverse l'état d'un opérande (Q, M ou GM).

Fonction de basculement	Syntaxe du programme
Inverse l'état	T(Q0.1)

Exemple :



Texte équivalent :

```
T(Q4.2) = P(I4.1)
```

Fonction : un front montant sur l'entrée 1 du Pluto 4 provoque le changement d'état de la sortie 2 du Pluto 4 (de 0 à 1 ou de 1 à 0).

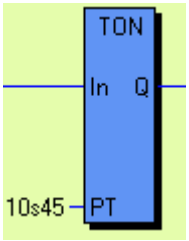
Remarque : cet exemple utilise une instruction de front montant pour éviter que Q4.2 ne bascule plus d'une fois. Sans instruction de front montant, la sortie risque de basculer à chaque cycle de l'automate, ce qui peut générer une erreur CPU (E51).

1.7 Temporisations

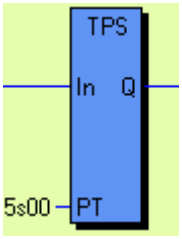
Pluto permet d'utiliser jusqu'à 50 temporisations simultanées dans une étape de séquence active (voir Séquences). Les temporisations ont une résolution de 10 ms et peuvent être définies par intervalles de temps de 0,01 à 655,35 s.

Temporisations:	Valeur:	Syntaxe du programme:	Ancienne syntaxe du programme :
TON	0.01- 655.35 s.	**TON** (*nn***S***nn*)	**T**(*nn***S***nn*)
TPS	0.01- 655.35 s.	**TPS** (*nn***S***nn*)	***I*T**(*nn***S***nn*)
*TOF**	*0.01- 655.35 s.*	***TOF**(nn**S**nn)*	**-**

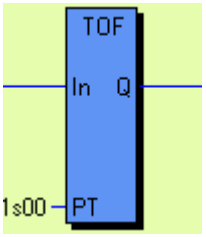
** Jeu d'instructions nº 3 seulement*



⇔ TON(10s45)



⇔ TPS(5s0)



⇔ *TOF(1s0)*

Retard à l'enclenchement (10,45 secondes)

Tempo impulsionnelle (5,0 secondes)

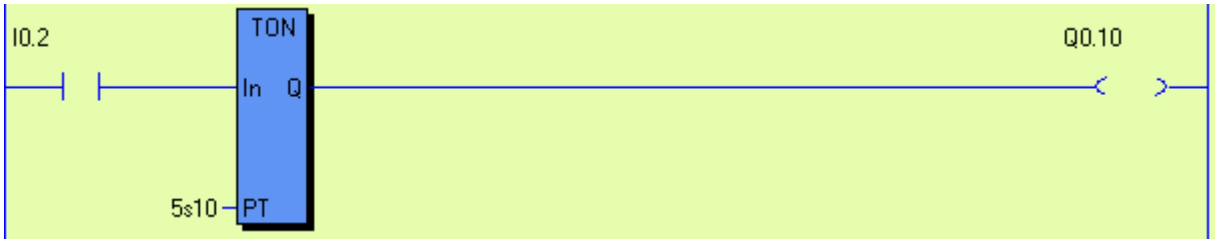
Retard au déclenchement (1,0 s)

Le symbole « s » correspond à la virgule décimale.

Fonction : il existe trois types de temporisations : les temporisations à l'enclenchement, les temporisations impulsionnelles et les temporisations au déclenchement. *(Les temporisations au déclenchement sont définies si le jeu d'instructions nº 3 est sélectionné.)*

Dans le cas du retard à l'enclenchement (TON), la sortie est activée une fois le temps indiqué écoulé si l'entrée est toujours active.
Dans le cas de l'impulsion (TPS), la sortie est activée en même temps que la condition d'entrée, mais au maximum pendant le temps indiqué.
Dans les deux cas, la sortie ne peut pas être active si l'entrée ne l'est pas.
La sortie de la temporisation au déclenchement (TOF) est à « 1 » quand les instructions booléennes en entrée sont à 1. Quand la condition d'entrée passe à « 0 », la temporisation commence à décompter et quand le temps indiqué est écoulé, la sortie passe aussi à « 0 »).

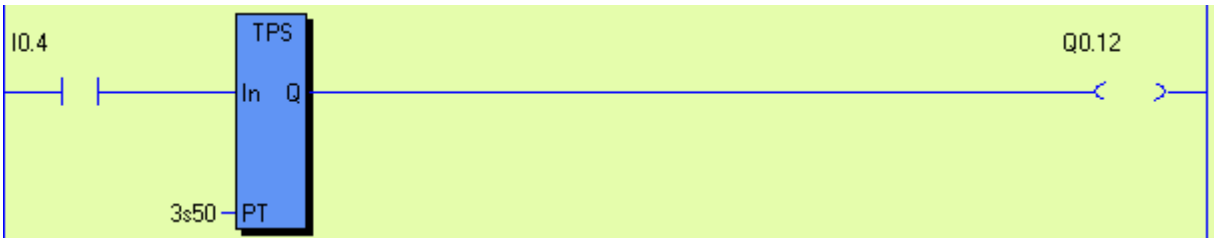
Exemple :



Texte équivalent : `Q0.10 = I0.2 * TON(5s10)`

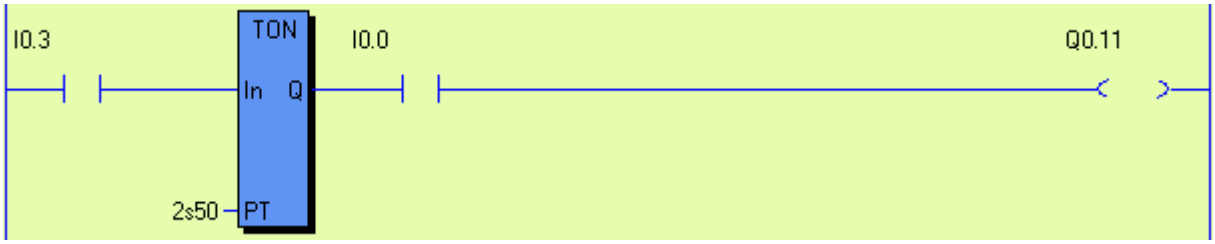
Fonction : quand l'entrée I0.2 est activée, la temporisation est activée. La sortie Q0.10 est activée 5,10 s après l'activation de I0.2 si cette dernière est toujours active.

Exemple :



Texte équivalent : `Q0.12 = I0.4 * TPS(2s5)`

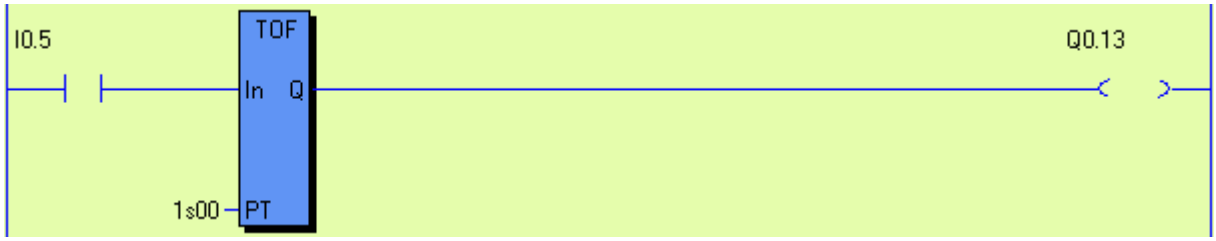
Fonction : quand l'entrée I0.4 est activée, la sortie Q0.12 est immédiatement activée.
Après 3,5 s, la temporisation désactive la sortie Q0.12.



Texte équivalent : `Q0.11 = I0.3 * TON(2s5) * I0.0`

Fonction : quand l'entrée I0.3 est activée, la temporisation est activée.
Après 2,5, si l'entrée I0.0 est activée, la sortie Q0.11 est aussi activée.
Observer que l'expression à droite de la temporisation (I0.0) n'a aucune influence sur elle.

Exemple:



Texte équivalent : `Q0.13 = I0.5 * TOF(1s00)`

***Fonction :** quand l'entrée I0.5 passe à 1, la sortie Q0.13 est immédiatement mise à 1. Quand l'entrée I0.5 passe à 0, le décompte de la temporisation de 1,00s débute. La sortie Q0.13 est mise à 0 quand le délai est écoulé.*

2 Mémoires

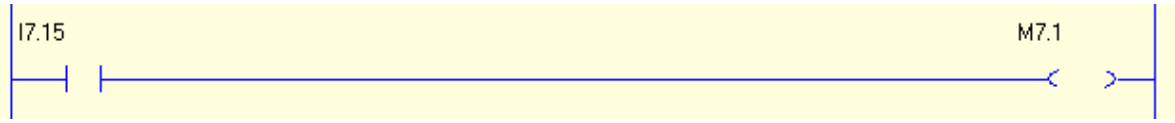
2.1 Mémoires locales (M)

Pluto dispose de 600 mémoires pouvant être utilisées librement. Ces mémoires sont locales et ne peuvent être utilisées que par leur propre Pluto. Par exemple, la mémoire M0.10 ne peut être activée et lue que par le programme du Pluto 0.

Les mémoires sont adressées comme suit :

Famille de Pluto :	Syntaxe du programme :
Tous modèles sauf Pluto AS-i	M_.0 – M_.599
Pluto AS-i	M_.0 – M_.149
Pluto AS-i, jeu d'instructions n° 3	*M_.0 – M_.599*

Exemple :



Texte équivalent : `M7.1 = I7.15`

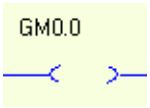
Fonction : la mémoire M7.1 est activée quand l'entrée I7.15 est activée.

Remarque : bien que les mémoires soient locales, le numéro de Pluto doit être indiqué dans leur identification comme indiqué ci-dessus.

2.2 Mémoires globales (GM)

Les mémoires globales sont utilisées comme les mémoires locales. Toutefois, elles sont automatiquement envoyées sur le bus et peuvent être lues et utilisées par les autres Pluto comme conditions d'entrées.
Elles sont utiles quand une fonction complexe doit être utilisée apr plusieurs Pluto par exemple. Au lieu de programmer la même fonction complexe dans tous les Pluto, elle peut être programmée dans un seul Pluto qui stocke le résultat dans une mémoire globale. Pour utiliser cette fonction, il suffit aux autres Pluto de lire cette mémoire globale sur le bus.

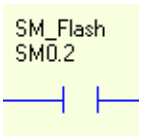
Les mémoires globales sont adressées comme suit :



Famille de Pluto :	Mémoire globale :	Syntaxe du programme :
Tous modèles sauf B42 AS-i	0-11	GM_.0 – GM_.11
B42 AS-i	0-27	GM_.0 – GM_.27

2.3 Mémoires du système (SM)

Pluto dispose d'un ensemble de mémoires du système dont voici les fonctions :



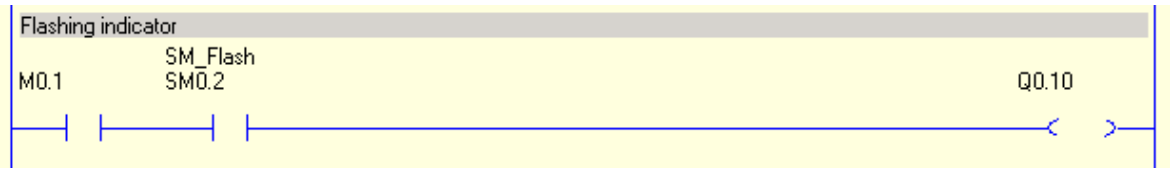
Syntaxe : SM[*enhet*].[*nr*]

Adresse E/S	Nom symbolique	Fonction	Type
SM_.0	SM_StepNew	« 1 » durant le premier cycle d'une étape de séquence.	R
SM_.1	SM_Ditto	Résultat de la dernière opération logique.	R
SM_.2	SM_Flash	Clignote : 0,4 / 0,6 seconde (haut/bas)	R
SM_.3	SM_1Hz	Impulsion 1 Hz	R
SM_.4	SM_10Hz	Impulsion 10 Hz	R
SM_.5	SM_FastFlash	Clignote : 0,17/0,33 seconde (haut/bas)	R
SM_.6	SM_DoubleFlash	Clignotement rapide : 0,11/0,2 /0,11 / 0,67 seconde	R
SM_.9	SM_SysInit	« 1 » durant le premier cycle de l'exécution du programme	R
SM_.11	*SM_Overflow*	*Débordement de l'arithmétique*	*R*
SM_.12	*SM_DivByZero*	*Division par zéro*	*R*
SM_.15**	SM_PlutoB	Il s'agit du processeur B de Pluto	R
SM_.39	SM_Button	Bouton sur la face avant	R
SM_.84*	SM_PlutoB	Il s'agit du processeur B de Pluto	R
SM_.100	SM_Pluto0_Present	Pluto #0 est présent	R
:			:
SM_.131	SM_Pluto31_Present	Pluto #31 est présent	R

* Seulement pour les Pluto simples (famille A20).
** Seulement pour B46, D45, AS-i et B42 AS-i.

Type : R = Read = lecture, W = Write = écriture

Exemple :



Texte équivalent : `Q0.10 = M0.1 * SM0.2` ; voyant clignotant

Fonction : la mémoire du système SM0.2 clignote avec des impulsions de 0,4 s au niveau haut et 0,6 s au niveau bas.
Si M0.1 est au niveau haut, la sortie Q0.10 est pulsée à la fréquence de SM0.2.

3 Séquences

Pluto dispose de 9 registres de séquences de 254 étapes chacune. Les séquences travaillent en parallèle et sont indépendantes les unes des autres.
Dans une séquence, le code d'une seule étape est exécuté (l'étape de séquence active). Des instructions de saut commandent le passage d'une étape à une autre. Les sauts peuvent être assujettis à des conditions. Le résultat de l'étape précédente est remis à zéro au passage à l'étape suivante. Au démarrage du système, l'étape de séquence 0 est automatiquement exécutée, ce qui signifie qu'une séquence doit contenir une étape 0.

3.1 Adressage

Une étape de séquence débute par la déclaration du numéro de séquence et du numéro d'étape. Voir ci-dessous.

Séquence/étape	Syntaxe de programme
1-9/0-254	S*n*.1_00 – S*n*.9_254 (*n*= nº de Pluto)

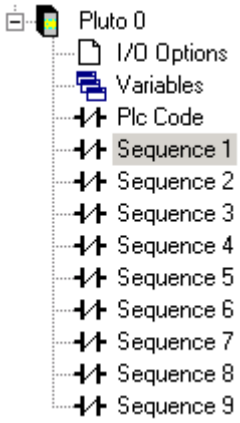
La syntaxe de programme en format texte est interprétée comme suit :

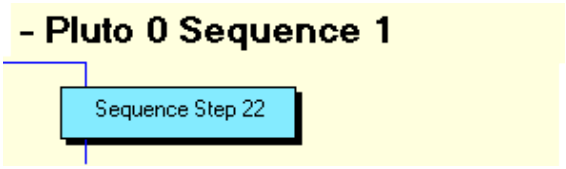
- La première lettre indique le registre de séquence (S)
- Le premier chiffre indique le numéro de Pluto.
- Le deuxième chiffre (après le point) indique le numéro de séquence
- Le troisième chiffre (après le tiret du dessous) indique le numéro d'étape.

Exemple :

S0.1_22 ⇔ Début de l'étape 22 de la séquence 1 du Pluto 0.

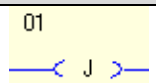
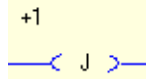
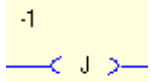
Programmation d'une séquence dans Pluto Manager :



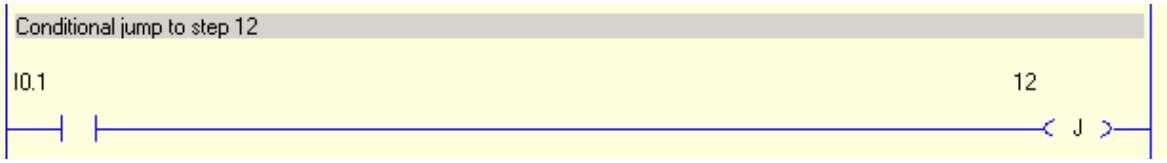


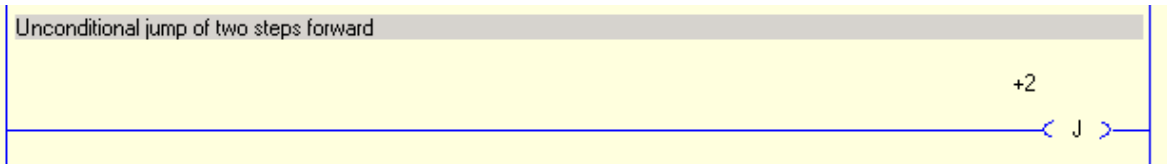
3.2 Saut

Les fonctions de saut sont utilisées dans les séquences pour sauter d'une étape à une autre. Un saut à une étape de la séquence peut être absolu ou relatif à l'étape active.

Fonction de saut	Syntaxe en format texte	Symbole Ladder
Absolu : à l'étape 1	J(01)	
Relatif : d'une étape en avant	J(+1)	
Relatif : d'une étape en arrière	J(-1)	

Un saut peut être assujetti à une condition mais ce n'est pas obligatoire :





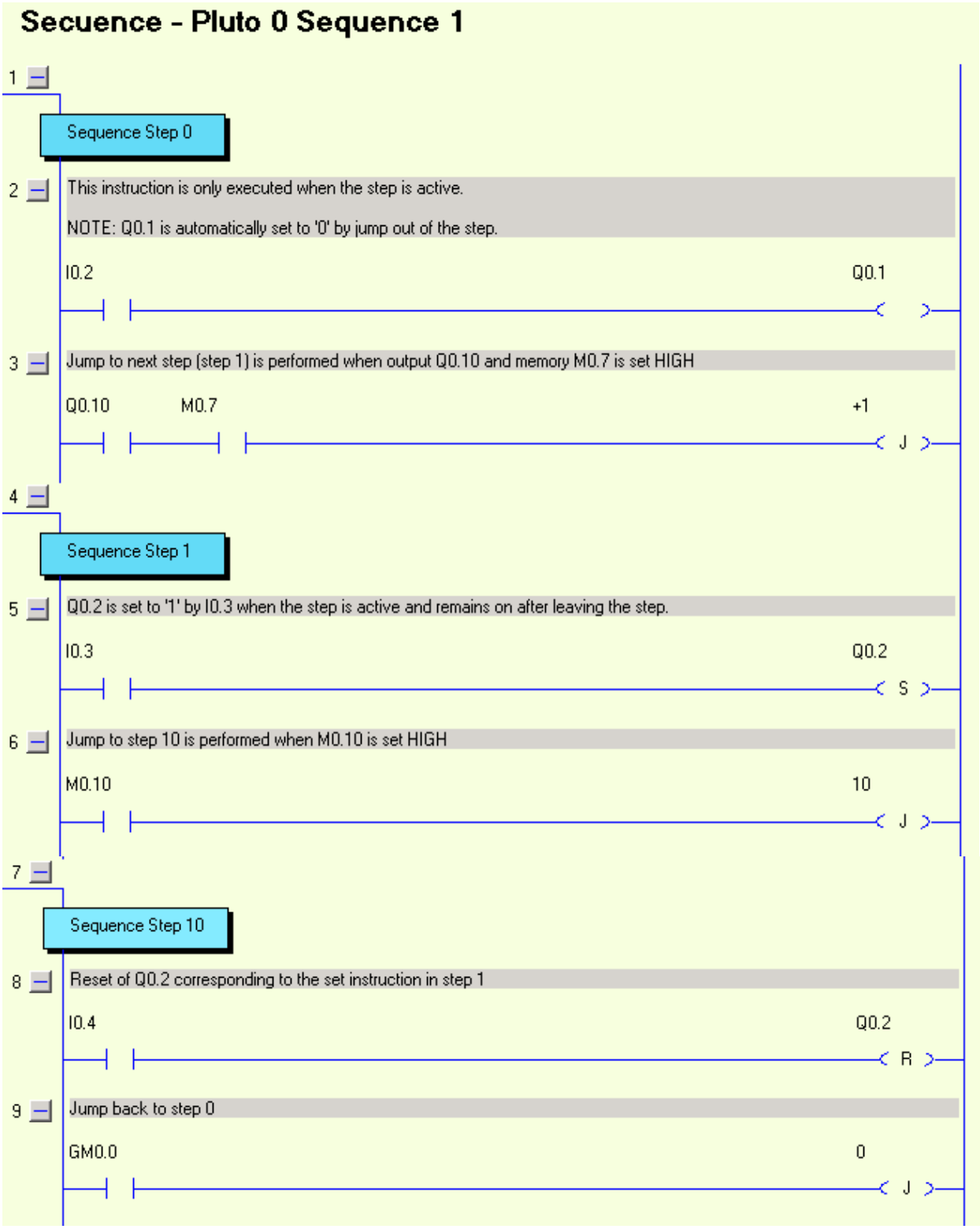
Exemple de séquence en format texte :

```
S0.1_00                 ⇔   Pluto 0, séquence 1, étape 0 :
Q0.1 = I0.2                 Q0.1 est activée par I0.2.
J(+1) = Q0.10*M0.7          Saut à l'étape suivante (étape 1) quand
                            Q0.10 et M0.7 sont activées.

S0.1_01                 ⇔   Pluto 0, séquence 1, étape 1 :
S(Q0.2) = I0.3              La sortie Q0.2 est activée par I0.3.
J(10) = M0.10               Saut à l'étape 10 si M0.10 est activée.

S0.1_10                 ⇔   Pluto 0, séquence 1, étape 10 :
R(Q0.2) = I0.4              La sortie Q0.2 est désactivée quand I0.4
                            est activée.
J(0) = GM0.0                Saut à l'étape 0 quand GM0.0 est activée.
```

Équivalent en Ladder



4 Opérations numériques

4.1 Registres

4.1.1 Adressage

Pluto dispose de 150 registres de 16 bits dans lesquels des valeurs peuvent être sauvegardées, des résultats de calculs, par ex. La plage des registres est : -32 768 … +32 767

Les registres sont adressés comme suit :

Registre	Syntaxe
0-149	R0.0 – R0.149

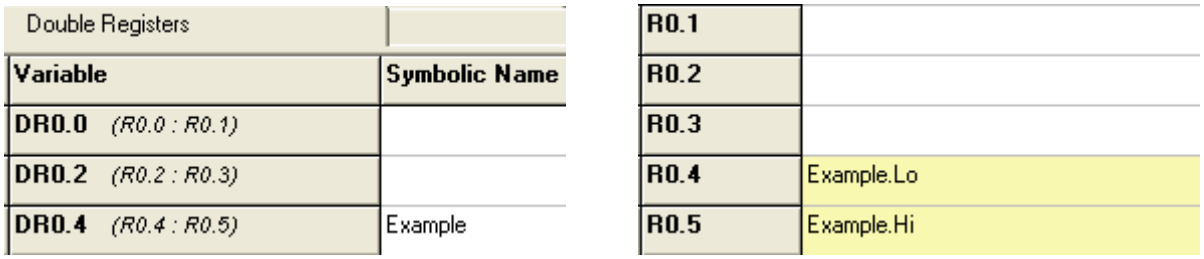
Le jeu d'instructions nº 3 introduit un nouveau type de variable, le registre double (DR, double Register). Un registre double est constitué du registre R correspondant (mot de poids faible) et du registre suivant (mot de poids fort). Par ex., DR1.4 = R1.5 (mot de poids fort) et R1.4 (mot de poids faible). Un registre double ne doit pas avoir de numéro impair. Un registre double peut gérer des valeurs de 32 bits, ce qui correspond aux valeurs numériques suivantes : -2147483648 … +2147483647

Registre double :	Syntaxe :
*0-148**	*DR0.0 – DR0.148*

** Seuls les nombres pairs sont autorisés*

4.1.1.1 Moitiés de registres doubles

Quand un registre double est utilisé, les deux registres simples constituant le registre double ne peuvent pas être adressés directement afin d'éviter les conflits de registres simples/doubles.
Par exemple, si DR0.4 est utilisé dans un programme, R0.4 et R0.5 ne peuvent pas être adressés directement. Il faut utiliser « DR0.4.Lo » (=R0.4) et « DR0.4.Hi » (=R0.5). Quand la syntaxe .Lo et .Hi est utilisée, le compilateur sait que le programmeur souhaite réellement accéder à la moitié d'un double registre.



Le registre double DR0.4 « Example » est constitué de R0.4 et R0.5. Ces moitiés du registre « Example » doivent être adressées comme « Example.Lo » et « Example.Hi ».

4.1.2 Fonctions

Opérations sur les registres (avec jeu d'instructions nº 2)

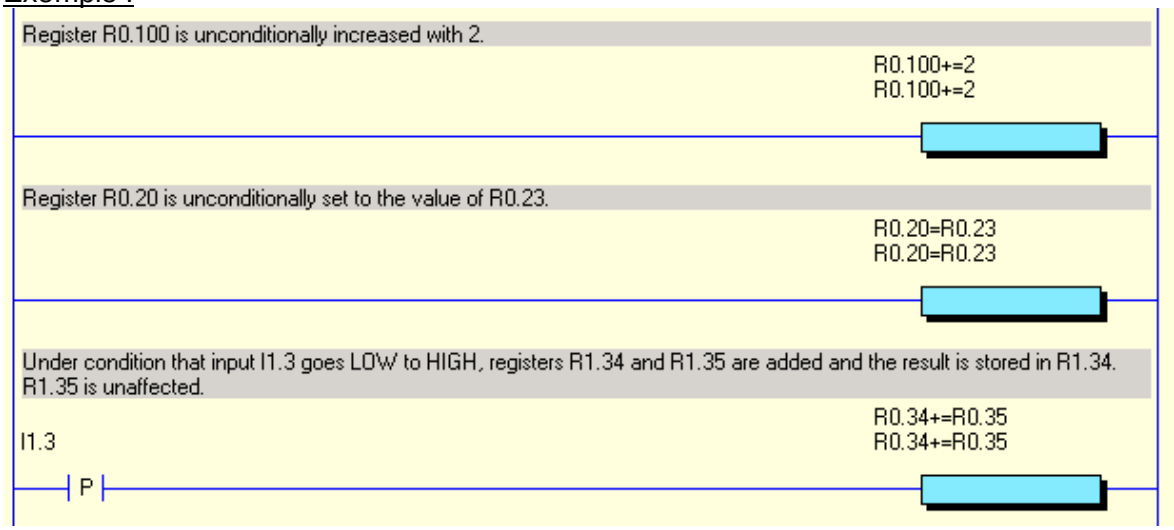
Fonction :	Syntaxe pour les registres :
Incrémenter de 1	(R0.100++)
Décrémenter de 1	(R0.100--)
Ajouter une constante	(R0.100 += 77)
Soustraire une constante	(R0.100 – = 77)
Attribuer une valeur absolue = 1	(R0.100 = 1)
Addition avec un autre registre (R0.100 = R0.100 + R0.102)	(R0.100 += R0.102)
Soustraction d'un autre registre (R0.100 = R0.100 – R0.102)	(R0.100 – = R0.102)
Attribuer la valeur d'un autre registre	(R0.100 = R0.102)

Opérations sur les registres (avec jeu d'instructions nº 3)

Fonction :	*Syntaxe pour les registres*	*Syntaxe pour les registre doubles :*
Incrémenter de 1	*(R0.100++)*	*(DR0.100++)*
Décrémenter de 1	*(R0.100--)*	*(DR0.100--)*
Ajouter une constante	*(R0.100 += 77)*	*(DR0.100 += 77)*
Soustraire une constante	*(R0.100 – = 77)*	*(DR0.100 – = 77)*
Attribuer une valeur absolue, = 1	*(R0.100 = 1)*	*(DR0.100 = 1)*
Attribuer la valeur d'un autre registre	*(R0.100 = R0.102)*	*(DR0.100 = DR0.102)*
Multiplier par une constante	*(R0.100 * = 2)*	*(DR0.100 * = 2)*
Diviser par une constante	*(R0.100 / = 2)*	*(DR0.100 / = 2)* *(OS ≥ 3.6)*
Addition avec un autre registre (R0.100 = R0.100 + R0.102)	*(R0.100 += R0.102)* *ou* *(R0.100=R0.100+R0.102)*	*(DR0.100 += DR0.102)* *ou* *(DR0.100=DR0.100+DR0.102)*
Addition avec un autre registre (et résultat dans un troisième registre) (R0.100 = R0.102 + R0.104)	*(R0.100=R0.102+R0.104)*	*(DR0.100=DR0.102+DR0.104)*
Soustraire un autre registre (R0.100 = R0.100 – R0.102)	*(R0.100 – = R0.102)* *ou* *(R0.100=R0.100-R0.102)*	*(DR0.100 – = DR0.102)* *ou* *(DR0.100=DR0.100-DR0.102)*
Soustraire un autre registre (et résultat dans un troisième registre) (R0.100 = R0.102 – R0.104)	*(R0.100=R0.102-R0.104)*	*(DR0.100=DR0.102-DR0.104)*
*Multiplier avec un autre registre (R0.100 = R0.100 * R0.102)*	*(R0.100 * = R0.102)* *ou* *(R0.100=R0.100*R0.102)*	*(DR0.100 * = DR0.102)* *ou* *(DR0.100=DR0.100*DR0.102)*
*Multiplier avec un autre registre (et résultat dans un troisième registre) (R0.100 = R0.102 * R0.104)*	*(R0.100=R0.102*R0.104)*	*(DR0.100=DR0.102*DR0.104)*
Diviser par un autre registre (R0.100 = R0.100 / R0.102)	*(R0.100 / = R0.102)* *ou* *(R0.100=R0.100/R0.102)*	*(DR0.100 / = DR0.102)* *or* *(DR0.100=DR0.100/DR0.102)*
Diviser par un autre registre (et résultat dans un troisième registre) (R0.100 = R0.102 / R0.104)	*(R0.100=R0.102/R0.104)*	*(DR0.100=DR0.102/DR0.104)* *(OS ≥ 3.6)*
Remarque : *il est impossible de mélanger R et DR dans des attributions.*		*Ex: (DR0.100 * = R0.102)*

En cas de division par zéro, SM_DivByZero (SM_.12) est mis à 1, et le résultat est mis à zéro. En cas de débordement, SM_Overflow (SM_.11) est mis à 1, et le résultat est mis soit à 32767 soit à -32768 en fonction du signe du débordement (pour DR: 2147483647 ou -2147483647). SR_Remain (SR_.2) contient le reste de la division.

Exemple :



Texte équivalent :

```
(R0.100+=2)
(R0.20=R0.23)
(R1.34+=R1.35) = P(I1.3)
```

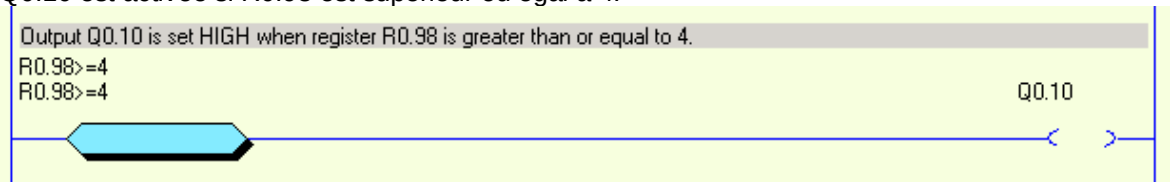
Remarque : en cas d'augmentation ou de diminution de registre, l'opération prend fin quand la limite est atteinte (32 767 ou -32 768).

Comparaison de registres

Comparaison :	Syntaxe pour les registres	*Syntaxe pour les registres doubles*
Égal (constante)	(R0.100=1)	*(DR0.100=1)*
Supérieur à	(R0.100>1)	*(DR0.100>1)*
Inférieur à	(R0.100<1)	*(DR0.100<1)*
Supérieur ou égal à	(R0.100>=1)	*(DR0.100>=1)*
Inférieur ou égal à	(R0.100<=1)	*(DR0.100<=1)*
Égal à (deux registres)	(R0.100=R0.101)	*(DR0.100=DR0.102)*
Supérieur à	(R0.100>R0.101)	*(DR0.100>DR0.102)*
Inférieur à	(R0.100< R0.101)	*(DR0.100< DR0.102)*
Supérieur ou égal à	(R0.100>= R0.101)	*(DR0.100>= DR0.102)*
Inférieur ou égal à	(R0.100<= R0.101)	*(DR0.100<= DR0.102)*
Différent de	(R0.100<> R0.101)	*(DR0.100=1)*

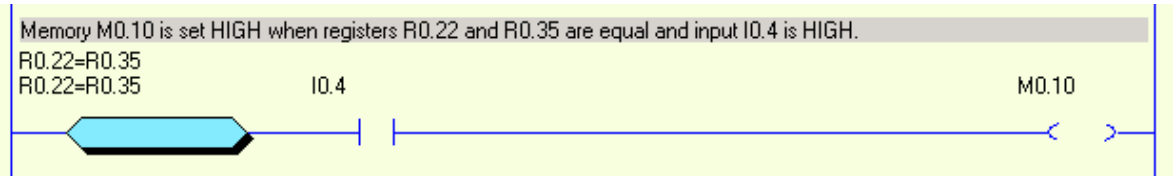
Exemple :

Q0.10 est activée si R0.98 est supérieur ou égal à 4.



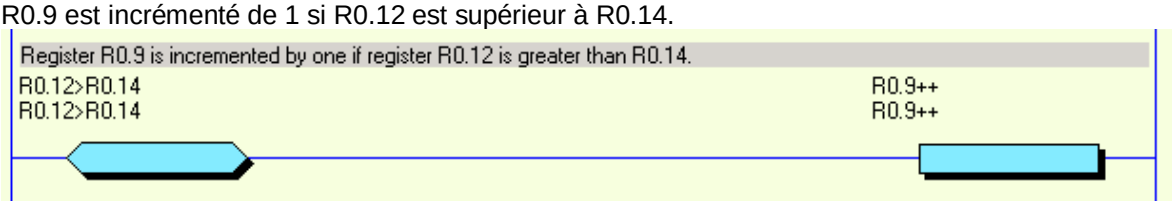
En format texte : `Q0.10 = (R0.98>=4)`

M0.10 est activée si R0.22=R0.35 et I0.4 est active.



En format texte : `M0.10 = (R0.22=R0.35)* I0.4`

R0.9 est incrémenté de 1 si R0.12 est supérieur à R0.14.



En format texte : `(R0.9++) = (R0.12>R0.14)`

4.1.3 Registres système

Pluto dispose de différents registres de système correspondants à plusieurs fonctions.

Registres système
Syntaxe : SR[*nº Pluto*].[*nº*]

Adresse E/S :	Nom symbolique :	Fonction :	Type :
Pour tous les modèles de Pluto :			
SR_.2	*SR_Remain*	*Reste de la division*	*R*
SR_.8*	SR_ExecFreeTime	Temps de cycle de l'API restant à utiliser (µs)	R
SR_.9	SR_ExecTime	Temps d'exécution de l'automate en µs	R
SR_.10	SR_PlutoDisplay	Chiffre sur l'afficheur du Pluto. Pour les codes d'erreurs utilisateur : 200+n°	W
SR_.11	SR_ErrorCode	Code d'erreur	R
SR_.12	SR_ErrorLog1	Dernier code d'erreur	R
SR_.13	SR_ErrorLog2	Avant dernier code d'erreur	R
SR_.14	SR_ErrorLog3	Avant avant dernier code d'erreur	R
SR_.40	SR_SupplVolt	Tension d'alimentation (x10 Volt)	R

A20, B20; D20, S20, B22 :			
SR_.41	SR_I5_Volt	Tension sur l'entrée analogique I5 (x10 volt)	R
SR_.42	SR_Q16_Current	Intensité (mA) sur la sortie Q16 (A20 seulement)	R
SR_.43	SR_Q17_Current	Intensité (mA) sur la sortie Q17 (A20 seulement)	R

B46, S46, D45 :			
SR_.41	SR_I5_Volt	Tension sur l'entrée analogique I5 (x10 volt)	R
SR_.45	SR_I6_Volt	Tension sur l'entrée analogique I6 (x10 volt)	R
SR_.46	SR_I7_Volt	Tension sur l'entrée analogique I7 (x10 volt)	R

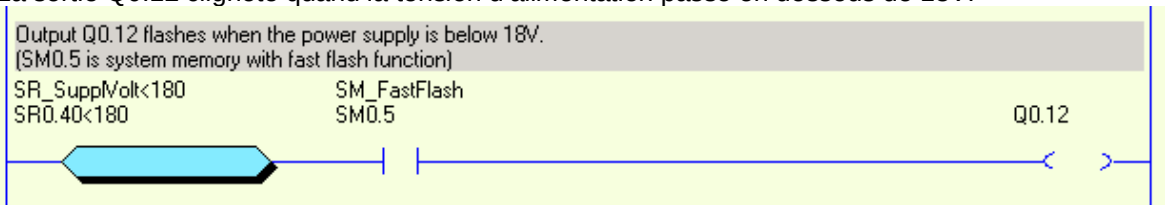
Pluto AS-i			
SR_.15**	SR_ASi_Slave_Missing	Premier esclave AS-i manquant. Esclave B codé comme n° +32	R
SR_.16**	SR_ASi_Slave_Chanf	Première erreur canaux esclave AS-i. Esclave B codé comme n° +32	R
SR_.41	SR_IQ11_Volt	Tension sur l'entrée analogique I11 (x10 volt)	R
SR_.44	SR_IQ10_Volt	Tension sur l'entrée analogique I10 (x10 volt)	R
SR_.45	SR_IQ12_Volt	Tension sur l'entrée analogique I12 (x10 volt)	R
SR_.46	SR_IQ13_Volt	Tension sur l'entrée analogique I13 (x10 volt)	R

B42 AS-i			
SR_.15	SR_ASi_Slave_Missing	Premier esclave AS-i manquant. Esclave B codé comme n° +32	R
SR_.16	SR_ASi_Slave_Chanf	Première erreur canaux esclave AS-i. Esclave B codé comme n° +32	R
SR_.41	SR_I1_Volt	Tension sur l'entrée analogique I1 (x10 volt)	R
SR_.45	SR_I2_Volt	Tension sur l'entrée analogique I2 (x10 volt)	R
SR_.46	SR_I3_Volt	Tension sur l'entrée analogique I3 (x10 volt)	R

*OS version 3.0 ou plus
*OS version 2.10.4 ou plus

Exemple :

La sortie Q0.12 clignote quand la tension d'alimentation passe en dessous de 18V.



En format texte : `Q0.12 = (SR0.40<180) * SM0.5`

4.2 Utilisation des valeurs analogiques

Les valeurs analogiques sont disponibles dans les registres SR40…SR46 (en fonction du modèle de Pluto, voir le tableau ci-dessous). Il existe certaines exigences pour leur utilisation.

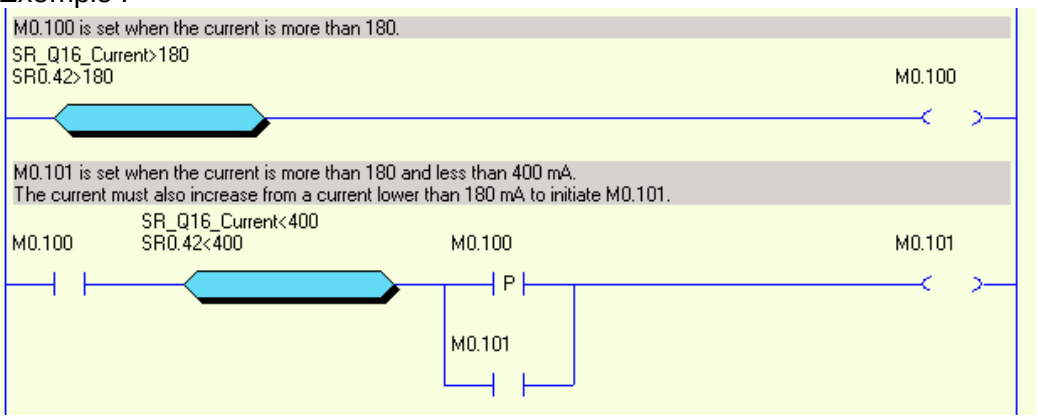
Entrées analogiques :
Comme indiqué dans le tableau ci-dessous, certaines entrées peuvent aussi être utilisées pour mesurer la tension sur la borne. La valeur peut être lue dans le registre du système (SR_) en dixième de volt (240 = 24.0 volt). Une valeur de registre égale à zéro ne peut par être utilisée comme condition dans une application de sécurité si cette valeur n'est pas surveillée de façon dynamique (le programme doit surveiller que la valeur varie). Les registres du système (SR_) sont en effet mis à 0 en cas de défaut interne du système.

Contrôle de courant sur Q16 et Q17 (Pluto A20 seulement) :
Les valeurs des intensités sur les sorties Q16 et Q17 se trouvent dans SR42 et SR43 exprimées en mA. Prévues pour le contrôle du courant dans une lampe d'inhibition, ces valeurs peuvent être utilisées autrement. Comme la mesure de courant n'est pas complètement redondante, les valeurs doivent être utilisées de façon dynamique. Par ex, si le courant dans une lampe doit être contrôlé, le programme doit être écrit pour qu'une modification de l'intensité soit détectée quand la sortie change d'état.

Entrées analogiques comme ci-dessous :

	B16, B20, S20, D20, B22, O2 :	A20 :	B46, S46 :	D45 :	Pluto AS-i :	B42 AS-i :
SR_40	Tension d'alim. (×10 V)	Tension d'alim. (×10 V)	Tension d'alim. (×10 V)	Tension d'alim. (×10 V)	Tension d'alim. (×10 V)	Tension d'alim. (×10 V)
SR_41	Tension entrée I5 (×10 V) *(Pas Pluto O2)*	Tension entrée I5 (×10 V)	Tension entrée I5 (×10 V)	Tension entrée I10 (×10 V)	Tension entrée I11 (×10 V)	Tension entrée I1 (×10 V)
SR_42	-	Intensité sortie Q16	-	-	-	-
SR_43	-	Intensité sortie Q17	-	-	-	-
SR_44	-	-	-	-	Tension entrée I10 (×10 V)	-
SR_45	-	-	Tension entrée I6 (×10 V)	Tension entrée I11 (×10 V)	Tension entrée I12 (×10 V)	Tension entrée I2 (×10 V)
SR_46	-	-	Tension entrée I7 (×10 V)	Tension entrée I12 (×10 V)	Tension entrée I13 (×10 V)	Tension entrée I3 (×10 V)

Exemple :



En format texte :

```
M0.100 = (SR0.42>180)
M0.101 = M0.100 * (SR0.42<400) * (P(M0.100) + M0.101)
```

5 Déclaration de programme en format texte

En début de programme, différentes déclarations décrivent l'environnement matériel du Pluto.

Ces différentes options matérielles sont décrites dans le « Manuel de l'utilisateur, Matériel ».

5.1 Identité, numéro de Pluto

Chaque Pluto doit avoir un numéro compris entre 0 et 31.
Un identifiant externe contenant un numéro hexadécimal à 12 chiffres unique peut être connecté au Pluto et doit alors être déclaré. Le modèle de Pluto doit aussi être déclaré.
Ces deux paramètres sont déclarés ainsi :

! id_pluto:[*numéro de Pluto*]=[*numéro d'identifiant*] pour les Pluto de la famille A20.
! id_pluto_Double:[*numéro de Pluto*]=[*numéro d'identifiant*] pour les Pluto « double ».
! id_pluto_ASi:[*numéro de Pluto*]=[*numéro d'identifiant*] pour les Pluto AS-i.
! id_pluto_ B42_ASi:[*numéro de Pluto*]=[*numéro d'identifiant*] pour les Pluto B42 AS-i.

Si aucun identifiant n'est connecté, le numéro d'identifiant doit être déclaré comme étant 000000000000 (12 zéros).

Exemple :

! id_pluto:00=ffff00007FA3	⇔	Le numéro 0 est attribué au Pluto et un identifiant de numéro ffff00007FA3 doit lui être connecté.
! id_pluto:23=000000000000	⇔	Le numéro 23 est attribué au Pluto et aucun identifiant n'est nécessaire.

5.2 Déclaration de code de programme

Comme il est possible d'avoir le code de programme de plusieurs Pluto sauvegardé dans un seul Pluto, il faut déclarer à quel Pluto appartient le code.

Syntaxe :

! pgm_pluto:*[n° de Pluto]*	⇔	Ici commence le programme de ce Pluto

5.3 Déclaration des E/S

Toutes les entrées de sécurité et les sorties non sécurisées (A20: Q10…17, B46 et B42 AS-i: Q10…27, Pluto AS-i: Q10…13) doivent être déclarées puisqu'elles peuvent être utilisées de différentes façons. Le tableau ci-dessous indique les différentes possibilités pour la famille A20.

Entrées
Syntaxe : ! I[nr],[puls typ],[switch 1],[switch 2]
Exemple : ! I0.5,c_pulse,non_inv,no_filt

Entrées :	Types de créneaux : (Signal dynamique)	Switch 1 : (au choix)	Switch 2 : (au choix)
I_.0 - I_.17	a_pulse b_pulse c_pulse	non_inv	no_filt
I_.0 - I_.17	static∗)		no_filt

∗) I_.10-I_17,**static** n'est pas conforme à la cat. 4 selon EN954-1, en tant qu'entrée à un seul canal

Sorties dynamiques
Syntaxe : ! Q[*nº],[type de créneau]*
Exemple : ! Q0.10,a_pulse

Sorties :	Types de créneaux :
Q_.10 – Q_.17	a_pulse b_pulse c_pulse

Sorties non sécurisées
Syntaxe : ! Q[*nº],static*
Exemple : ! Q0.10,static

Sorties :	Types de créneaux :
Q_.10 –Q_.17	static

Fonction bouton-poussoir lumineux
Syntaxe : ! IQ[*nº],[type de créneau]*
Exemple : ! IQ0.12,a_pulse

Entrée/Sortie :	Types de créneaux :
IQ_.10 –IQ_.17	a_pulse b_pulse c_pulse

Exemple :

```
! i0.1,a_pulse          ; Entrée connectée à un créneau de type A via un inverseur.
! i0.2,a_pulse,non_inv  ; Entrée connectée à un créneau de type A sans inverseur.
! i0.3,static           ; Entrée connectée à un signal +24V.

! q0.10,a_pulse         ; Sortie qui génère un créneau de type A pouvant être utilisé par
                          différents dispositifs.
! q0.11,static          ; Sortie qui délivre +24V.
```

5.4 Noms symboliques

Il est aussi possible d'attribuer un nom symbolique aux variables afin d'améliorer la lisibilité du programme. Dans Pluto Manager, ces noms sont déclarés dans une fenêtre spécifique, voir la 1ère partie.

En format texte, ils sont déclarés comme indiqué ci-dessous. L'emplacement de la déclaration dans le programme dépend de si la variable est locale ou globale. Les variables globales sont déclarées avant le code de programme du premier Pluto puisque ces variables peuvent être utilisées par tous les Pluto. Les variables locales sont déclarées au début du code de programme du Pluto concerné, après la déclaration des E/S.

Exemple :

```
! I0.0=MuteSensor1                ; Noms symboliques, variables globales
! Q0.1=MuteSensor2
! GM0.1=MuteSensor2

! Q0.14=IndReset                  ; Noms symboliques, variables locales
! M0.0=MutingActive
! R0.0=Counter1
```

6 Syntaxe de programme en format texte :

Cet exemple de programme correspond à l'exemple d'application au chapitre 8 du « Manuel d'utilisation, Matériel ».

$name Example, manual

```
! id_pluto:00=000034AD4AE1

! i0.12 = Arrêt_Urgence        ; Déclaration du nom symbolique, E/S globale

! pgm_pluto:00

! q0.10,a_pulse                ; Sortie dynamique A

! i0.00,static                 ; Capteur d'inhibition 1
! i0.01,a_pulse,non_inv        ; Capteur d'inhibition 2
! i0.02,a_pulse,non_inv        ; Test Contacteurs
! i0.12,a_pulse                ; Arrêt d'urgence
! i0.13,a_pulse                ; Mono Faisceau
! iq0.14,a_pulse               ; Réarmement avec voyant

! q0.14 = VoyantReset          ; Déclaration du nom symbolique, variable locale

;*****************************************

s0.0_0                         ; Début séquence principale

q0.2 = Arrêt_Urgence * (i0.13 + m0.0) * ( (p(i0.14) * i0.02) + q0.2)
q0.3 = q0.2
                               ; Toutes les sorties de sécurité actives quand arrêt
                               ; d'urgence (I0.12) et mono faisceau (I.13) ou inhibition
                               ; (M0.0) actifs.
                               ; Réarmement (I0.14) et Test (I0.02) sont aussi
                               ; nécessaires dans  condition de départ.
                               ; (Arrêt_Urgence = nom symbolique de i0.12)

VoyantReset = /q0.2            ; Voyant de réarmement actif quand les sorties ne
                               ; sont pas actives
                               ; (VoyantReset = nom symbolique de q0.14)

;*****************************************

s0.1_0                         ; Séquence d'inhibition
j(+1)=/i0.00*/i0.01*(SR0.43<100)
                               ; Conditions de démarrage : les deux capteurs inactifs

s0.1_1
q0.17 = i0.00 * i0.01 * i0.13
j(+1) = q0.17 * (SR0.43<100)   ; L'inhibition démarre quand les deux capteurs et
                               ; le mono faisceau sont actifs
s0.1_2
m0.0                           ; M0.0, Mémoire d'inhibition active
q0.17                          ; Voyant d'inhibition actif
j(0) = /i0.00 + /i0.01         ; Inhibition interrompue car un des capteurs inactif
```

7 Annexe A, compatibilité avec les Pluto antérieurs

Certaines fonctions décrites dans ce manuel ne sont pas compatibles avec d'anciens modèles de Pluto. Le tableau ci-dessous indique avec quelles versions matérielles et versions d'OS la fonction en question peut être utilisée.

Fonction	Type de Pluto	Version matérielle	Version d'OS
Jeu d'instruction n° 3	A20 v2	Toutes	Toutes
	B20 v2	Toutes	Toutes
	S20 v2	Toutes	Toutes
	B22	Toutes	Toutes
	D20	Toutes	Toutes
	B46 v2	Toutes	≥3.0
	S46 v2	Toutes	≥3.0
	D45	Toutes	Toutes
	AS-i v2	Toutes	≥3.0
	B42 AS-i	Toutes	Toutes
	O2	Toutes	Toutes
Variables rémanentes	A20 v2	Toutes	Toutes
	B20 v2	Toutes	Toutes
	S20 v2	Toutes	Toutes
	B22	Toutes	Toutes
	D20	Toutes	Toutes
	B46 v2	HW ≥ 2.11	≥3.0
	S46 v2	HW ≥ 2.11	≥3.0
	D45	Toutes	Toutes
	AS-i v2	HW ≥ 3.7	≥3.0
	B42 AS-i	Toutes	Toutes
Exportation de variables	Tous les Pluto avec jeu d'instructions n°3	Voir Jeu d'instructions n°3	≥3.2
Désactivation des impulsions de test Q2, Q3	A20 v2	Toutes	Toutes
	B20 v2	Toutes	Toutes
	S20 v2	Toutes	Toutes
	B22	Toutes	Toutes
	D20	Toutes	Toutes
« Read IDFIX number from Pluto »	Toutes	Toutes	≥3.4

Pour nous contacter

Australia
ABB Australia Pty Limited
Low Voltage Products
Tel: +61 (0)1300 660 299
Fax: +61 (0)1300 853 138
Mob: +61 (0)401 714 392
E-mail: kenneth.robertson@au.abb.com
Web: www.abbaustralia.com.au

Austria
ABB AB, Jokab Safety
Tel: +43 (0)1 601 09-6204
Fax: +43 (0)1 601 09-8600
E-mail: aleksander.gauza@at.abb.com
Web: www.abb.at

Belgium
ABB N.V.
Tel: +32 27186884
Fax: +32 27186831
E-mail: tech.lp@be.abb.com

Brazil
ABB Ltda
Produtos de Baixa Tensão
ABB Atende: 0800 014 9111
Fax: +55 11 3688-9977
Web: www.abb.com.br

Canada
ABB Inc.
Tel: +1 514 420 3100 Ext 3269
Fax: +1 514 420 3137
Mobile: +1 514 247 4025
E-mail: alan.m.brown@ca.abb.com
Web: www.abb.com

China
ABB (China) Limited
Tel: 86-21-23287948
Telefax: 86-21-23288558
Mobile: 86-186 2182 1159
E-mail: harry-yarong.zhang@cn.abb.com

Czech Republic
ABB AB, Jokab Safety
Tel: +420 543 145 482
Fax: +420 543 243 489
E-mail: premysl.broz@cz.abb.com
Web: www.abb.cz

Denmark
JOKAB SAFETY DK A/S
Tel: +45 44 34 14 54
Fax: +45 44 99 14 54
E-mail: info@jokabsafety.dk
Web: www.jokabsafety.dk

ABB A/S
Tel: +45 4450 4450
Fax: +45 4359 5920
E-mail: ordre.komp@dk.abb.com
Web: www.abb.dk

Finland
ABB Oy
Web: www.abb.fi

France
ABB France
Division Produits Basse Tension
Tel: 0825 38 63 55
Fax: 0825 87 09 26
Web: www.abb.com

Germany
ABB STOTZ-KONTAKT GmbH
Tel: +49 (0) 7424-95865-0
Fax: +49 (0) 7424-95865-99
E-mail: buero.spaichingen@de.abb.com
Web: www.abb.de

Greece
ABB SA
Tel: +30 210.28.91.900
Fax: +30 210.28.91.999
E-mail: dimitris.voulgaris@gr.abb.com
nikos.makrakos@gr.abb.com
Web: www.abb.com

Ireland
ABB Ltd.
Tel +353 1 4057 381
Fax: +353 1 4057 312
Mobile: +353 86 2532891
E-mail: derek.kelly@ie.abb.com

Israel
ABB Technologies Ltd.
Tel: +972 4 851-9204
Mobile: +972 52 485-6284
E-mail: contact@il.abb.com
Web: www.abb.co.il

Italy
ABB S.p.A.
Tel. +39 02 2414.1
Fax +39 02 2414.2330
Web: www.abb.it

Korea
ABB KOREA
Low-voltage Product
Tel: +82 2 528 3177
Fax: +82 2 528 2350
Web: www.jokabsafety.co.kr

Malaysia
ABB Malaysia
Tel: +60356284888 4282
E-mail: chang-sheng.saw@my.abb.com

Netherlands
ABB b.v.
Tel:+31 (0) 10 - 4078 947
Fax: +31 (0) 10 – 4078 090
E-mail: info.lowvoltageproducts@nl.abb.com
Web: www.abb.nl

Norway
ABB AS
Tel: +47 03500
Fax: +47 32858021
Mobile: +47 40918930
E-mail: Lars-Erik.Arvesen@no.abb.com
Web: www.abb.no

Poland
ABB Sp. z.o.o
Tel: +48 728 401 403
Fax: 22 220 22 23
E-mail: adam.rasinski@pl.abb.com,
safety@pl.abb.com
Web: www.abb.pl

Portugal
Asea Brown Boveri S.A.
Low Voltage Products - Baixa Tensão
Tel: +35 214 256 000
Fax: +35 214 256 390
Web: www.abb.es

Slovenia
ABB d.o.o.
Tel: +386 1 2445 455
Fax: +386 1 2445 490
E-mail: aljosa.dobersek@si.abb.com

Spain
Asea Brown Boveri S.A.
Tel: +34 93 4842121
Fax: +34 93 484 21 90
Web: www.abb.es

South Africa
ABB
Tel: +27 10 202 5906
Fax: +27 11 579 8203
Mobile: +27 82 500 7990
E-mail: Hendrik.Spies@za.abb.com

Sweden
ABB AB, Jokab Safety
Varlabergsvägen 11
SE-434 39 Kungsbacka
Tel: +46 21 32 50 00
Fax: +40 67 15 601
Mail: support.jokabsafety@se.abb.com
Web: www.abb.com/jokabsafety

Switzerland
ABB Schweiz AG
Industrie- und Gebäudeautomation
Tel: +41 58 586 00 00
Fax: +41 58 586 06 01
E-mail: industrieautomation@ch.abb.com
Web: www.abb.ch

Turkey
ABB Elektrik Sanayi A.Ş
Tel: 0216 528 22 00
Fax: 0216 365 29 44

United Kingdom
ABB Ltd/JOKAB SAFETY UK
Tel: +44 (0) 2476 368500
Fax: +44 (0) 2476 368401
E-mail: orders.lvp@gb.abb.com
Web: www.jokabsafety.com

USA/Mexico
ABB Jokab Safety North America
Tel: +1 519 735 1055
Fax: +1 519 7351299
E-mail: jokabnaorderentry@us.abb.com
Web: www.jokabsafetyna.com