

# WAGO → I/O → SYSTEM 750

## Mise en service des contrôleurs Ethernet 32 bits sous CoDeSys 2.3



### Note d'application

A47070d, Français  
Version 1.2.0  
22/08/2011

Copyright © 2011 by WAGO Kontakttechnik GmbH & Co. KG  
Tous droits réservés.

### **WAGO Kontakttechnik GmbH & Co. KG**

Hansastraße 27  
D-32423 Minden

Phone: +49 571-887-0

Fax: +49 571 887-169

E-Mail: [info@wago.com](mailto:info@wago.com)

Web: <http://www.wago.com>

### **Technical Support**

Phone: +49 571 887 555

Fax: +49 571 887 8555

E-Mail: [support@wago.com](mailto:support@wago.com)

### **WAGO Contact SAS**

83 rue des chardonnerets  
BP 55065 - ZI Paris Nord 2  
95947 Roissy Charles de Gaulle Cedex

Tel. : +33 (0) 1 48 17 25 90

Fax : +33 (0) 1 48 63 25 20

E-Mail : [info-fr@wago.com](mailto:info-fr@wago.com)

Web: <http://www.wago.fr>

### **Support technique**

Tel. : +33 (0) 1 48 17 54 54

Fax : +33 (0) 1 48 17 54 08

E-Mail : [support-fr@wago.com](mailto:support-fr@wago.com)

Toutes les mesures imaginables ont été prises pour garantir l'exactitude et la complétude de la présente documentation. Comme il est impossible, malgré un travail consciencieux, d'éviter toutes les erreurs, nous recevons avec gratitude vos remarques et suggestions.

Nous attirons votre attention sur le fait que dans ce manuel, les désignations de logiciels et de matériels et plus généralement les noms de marques des entreprises concernées sont soumis à une protection des marchandises, à une protection des marques ou à une protection liée aux droits de brevet.

# SOMMAIRE

<b>1 Informations importantes .....</b>	<b>2</b>
1.1 Bases juridiques .....	2
1.1.1 Protection des droits d'auteur .....	2
1.1.2 Qualification du personnel .....	2
1.1.3 Utilisation dans le cadre prévu .....	2
1.2 Domaine de validité .....	3
1.3 Symboles .....	3
<b>2 Description .....</b>	<b>4</b>
<b>3 Domaine d'application .....</b>	<b>4</b>
<b>4 Matériel de référence .....</b>	<b>5</b>
4.1 Matériel .....	5
4.2 Logiciel .....	5
<b>5 Installation et câblage .....</b>	<b>6</b>
5.1 Montage sur le rail .....	6
5.2 Alimentation .....	7
5.2.1 Câblage avec une unique alimentation .....	8
5.2.2 Câblage avec deux alimentations .....	8
5.3 Réseau Ethernet .....	9
5.4 Vérification de l'établissement du lien physique .....	10
<b>6 Paramétrage IP .....</b>	<b>11</b>
6.1 Paramétrage du PC .....	11
6.1.1 Windows 2000/XP .....	11
6.1.2 Windows 7 .....	12
6.2 Paramétrage du contrôleur .....	13
6.2.1 Paramétrage par micro-interrupteurs .....	13
6.2.2 Paramétrage par logiciel .....	14
6.3 Test de la communication .....	18
<b>7 Accès au serveur web .....</b>	<b>19</b>
<b>8 CoDeSys .....</b>	<b>22</b>
8.1 Installation .....	22
8.2 Aperçu de l'éditeur .....	24
8.3 Les langages de programmation .....	24
8.4 Les modules .....	27
8.4.1 Les programmes .....	27
8.4.2 Les blocs fonctionnels .....	27
8.4.3 Les fonctions .....	28
8.5 Les types de données .....	29
8.5.1 Les booléens .....	29
8.5.2 Les entiers .....	29
8.5.3 Les réels .....	30

8.5.4	Les données temporelles .....	30
8.5.5	Les Chaînes de caractères .....	31
8.5.6	Les tableaux .....	31
8.5.7	Les structures et les énumérations .....	32
8.6	Les adresses.....	33
<b>9</b>	<b>Réalisation d'un premier programme.....</b>	<b>34</b>
9.1	Préambule .....	34
9.2	Sélection de la cible .....	34
9.3	Création du programme principal .....	36
9.4	Définition de la configuration matérielle et déclaration des variables d'entrées/sorties .....	36
9.5	Saisie du programme principal .....	41
9.6	Création d'une fenêtre de visualisation.....	45
<b>10</b>	<b>Exécution du programme en simulation.....</b>	<b>48</b>
<b>11</b>	<b>Téléchargement et sauvegarde dans le contrôleur.....</b>	<b>51</b>
11.1	Téléchargement et sauvegarde du code compilé.....	51
11.2	Téléchargement du code source .....	55
11.3	Récupération du code source .....	56
11.4	Activation des changements en ligne.....	57
<b>12</b>	<b>Activation de la visualisation web .....</b>	<b>60</b>
<b>13</b>	<b>Utilisation du modèle de projet.....</b>	<b>62</b>
<b>14</b>	<b>Zones mémoires des contrôleurs.....</b>	<b>67</b>
<b>15</b>	<b>Affectation des bornes de sortie.....</b>	<b>70</b>
<b>16</b>	<b>Modbus.....</b>	<b>72</b>
16.1	Introduction.....	72
16.2	Codes fonction .....	73
16.3	Format des adresses Modbus .....	74
16.4	Correspondance entre les adresses internes et les adresses Modbus....	75
16.4.1	Accès mots.....	75
16.4.2	Accès bits.....	75
16.5	Organisation des tables des entrées/sorties physiques .....	76
16.5.1	Exemple 1 .....	77
16.5.2	Exemple 2 .....	78
16.6	Test de la communication Modbus .....	79
<b>17</b>	<b>Autres protocoles de communication .....</b>	<b>81</b>
17.1	EtherNet/IP.....	81
17.2	KNX IP .....	81
17.3	BACnet/IP.....	81
17.4	CEI 60870-5.....	82
17.5	CEI 61850 .....	82

<b>18 Utilisation des bibliothèques .....</b>	<b>83</b>
18.1 Dossiers de stockage des bibliothèques .....	83
18.2 Insertion de bibliothèques dans le projet.....	84
<b>19 Gestion des entrées/sorties analogiques .....</b>	<b>87</b>
19.1 Exemple pour les entrées analogiques .....	87
19.2 Exemple pour les sorties analogiques .....	89
19.3 Remarque sur les entrées température .....	90
<b>20 Annexe A : LED I/O .....</b>	<b>91</b>
<b>21 Annexe B : affectation de l'adresse IP par protocole BootP .....</b>	<b>94</b>
<b>22 Annexe C : lecture de la configuration matérielle depuis WAGO-I/O- CHECK.....</b>	<b>99</b>
<b>23 Annexe D : mise à jour du firmware du contrôleur.....</b>	<b>104</b>
23.1 WAGO FBC Update .....	104
23.2 WAGO Ethernet Update .....	109
<b>24 Informations complémentaires .....</b>	<b>113</b>

# 1 Informations importantes

Pour assurer à l'utilisateur une installation et une mise en service rapides des appareils décrits dans ce manuel, il est nécessaire de lire et de respecter scrupuleusement les informations et les explications suivantes.

## 1.1 Bases juridiques

### 1.1.1 Protection des droits d'auteur

Ce manuel, y compris toutes les illustrations qui s'y trouvent, est protégé par la législation sur les droits d'auteur. Toute autre utilisation de ce manuel s'écartant de la réglementation concernant les droits d'auteur est interdite. Sa reproduction, sa traduction dans une autre langue, de même que son archivage et modification électronique et phototechnique nécessitent une autorisation expresse écrite de WAGO Kontakttechnik GmbH & Co. KG, Minden. Toute infraction fera l'objet d'une demande de dommages et intérêts.

### 1.1.2 Qualification du personnel

L'utilisation des produits telle qu'elle est décrite dans ce manuel s'adresse exclusivement à des personnes possédant une formation dans la programmation d'un API, à des personnes formées en électricité ou à des personnes placées sous la responsabilité de personnes formées en électricité, et qui de plus sont familiarisées avec les normes en vigueur. WAGO Kontakttechnik GmbH & Co. KG décline toute responsabilité pour des mauvaises manipulations ou des dommages causés sur des produits WAGO ou des produits tiers, dus au non-respect des informations contenues dans ce manuel.

### 1.1.3 Utilisation dans le cadre prévu

Les composants sont livrés depuis l'usine pour chacun des cas d'application avec une configuration fixe, matérielle et logicielle. Les modifications ne sont permises que dans le cadre des possibilités contenues dans les manuels. Toute autre modification sur les matériels et logiciels, de même qu'une utilisation non conforme à la réglementation entraîne l'exclusion de la responsabilité de la société WAGO Kontakttechnik GmbH & Co. KG.

Veillez vous adresser directement à la société WAGO Kontakttechnik GmbH & Co. KG pour toute demande de modification, voire de nouvelle configuration.

## 1.2 Domaine de validité

Cette note d'application est basée sur des versions logicielles et matérielles spécifiques, ainsi que sur la documentation s'y référant. La validité de cette note d'application est donc limitée à l'installation décrite. De nouvelles versions logicielles et matérielles pourraient donner lieu à des manipulations différentes.

Veillez respecter les descriptions détaillées dans les manuels respectifs.

## 1.3 Symboles



### **Danger**

Respecter impérativement ces informations afin de préserver toute personne d'un éventuel dommage



### **Avertissement**

Respecter impérativement ces informations afin de préserver l'appareil de tout dommage matériel



### **Attention**

Respecter impérativement les conditions additionnelles permettant d'assurer un fonctionnement sans erreur



### **ESD** (*Electrostatic Discharge* – Décharge électrostatique)

Attention aux composants sensibles aux décharges électrostatiques. Respecter les mesures de précaution pour le maniement des composants à risques.



### **Remarque**

Procédures ou conseils pour une utilisation efficace de l'appareil et pour une optimisation logicielle



### **Informations complémentaires**

Renvoi à des références de littérature supplémentaires, manuels, fiches techniques, et pages INTERNET

## 2 Description

Cette note d'application décrit la mise en service des contrôleurs Ethernet 32 bits sous CoDeSys 2.3.

Les différentes étapes y sont détaillées, du câblage de l'alimentation à la programmation en elle-même. Des fonctionnalités complémentaires comme l'utilisation de la visualisation web y sont également abordées.

## 3 Domaine d'application

Cette note d'application s'applique aux contrôleurs programmables listés dans les tableaux ci-dessous.

Contrôleurs de seconde génération :

Référence	750-881	750-880	750-882
Interface réseau	2xRJ45 switch intégré	2xRJ45 switch intégré	2xRJ45 2 adresses IP
Interface additionnelle	-	Slot SD	-
Protocoles	EtherNet/IP	EtherNet/IP	Modbus/TCP
	Modbus/TCP	Modbus/TCP	
	Modbus/UDP	Modbus/UDP	
Programme	1024 ko	1024 ko	1024 ko
Données	512 ko	1024 ko	512 ko
Données rémanentes	16 ko	16 ko	16 ko
Données rémanentes adressées (%M)	16 ko	16 ko	16 ko
Mémoire flash	1.7 Mo	1.7 Mo	1.7 Mo
Firmware ≥	2	2	1

Contrôleurs de première génération :

Référence	750-841	750-871	750-873	750-872	750-849	750-830
Interface réseau	1xRJ45	2xRJ45 switch intégré	1xRJ45	1xRJ45	2xRJ45 switch intégré	1xRJ45
Interface additionnelle	-	-	1xRS232	1xRS232	-	1xRS232
Protocoles	EtherNet/IP	EtherNet/IP	EtherNet/IP	CEI 60870-5 CEI 61850	KNX IP	BACnet/IP
	Modbus/TCP	Modbus/TCP	Modbus/TCP	Modbus/TCP	Modbus/TCP	Modbus/TCP
	Modbus/UDP	Modbus/UDP	Modbus/UDP	Modbus/UDP	Modbus/UDP	Modbus/UDP
Programme	512 ko	1024 ko	1024 ko	1024 ko	512 ko	512 ko
Données	256 ko	256 ko	1024 ko	1024 ko	256 ko	256 ko
Données rémanentes	16 ko	16 ko	16 ko	16 ko	16 ko	16 ko
Données rémanentes adressées (%M)	8 ko	12 ko	12 ko	12 ko	8 ko	8 ko
Mémoire flash	1.7 Mo	1.7 Mo	1.7 Mo	1.7 Mo	1.4 Mo	4.5 Mo
Firmware ≥	18	5	3	3	4	3



## 4 Matériel de référence

### 4.1 Matériel

Référence	Description
750-881	Contrôleur Ethernet 10/100 MBaud
750-432	Borne 4 entrées digitales 24 V DC - 3.0 ms - 2 fils
750-531	Borne 4 sorties digitales 24 V DC - 0.5 A - 2 fils
750-600	Borne d'extrémité finale de bus
787-712	Alimentation à découpage 230 V AC / 24 V DC - 2.5 A
759-333/000-923	Outil de programmation CoDeSys 2.3 (Logiciel + Câble USB)



#### Remarque

Pour que le contrôleur fonctionne, la présence d'au moins une borne d'entrée ou d'une borne de sortie est nécessaire, ainsi que la borne d'extrémité finale de bus !

### 4.2 Logiciel

Logiciel	Version
CoDeSys for Automation Alliance	2.3.9.28
WAGO Ethernet Settings	5.2.1
Pilote de câble USB WAGO Séries 750 et 857	6.1.0.0

Les outils **WAGO Ethernet Settings** et le pilote de câble USB peuvent être téléchargés sur [www.wago.fr](http://www.wago.fr), et sont également présents sur le DVD **AUTOMATION Tools and Docs**, livré avec le câble USB :

- > Services > Téléchargements > Téléchargements > AUTOMATION
- > WAGO-I/O-SYSTEM 750/753



#### Remarque

Toutes ces versions logicielles sont compatibles avec Windows 7, éditions 32 et 64 bits.

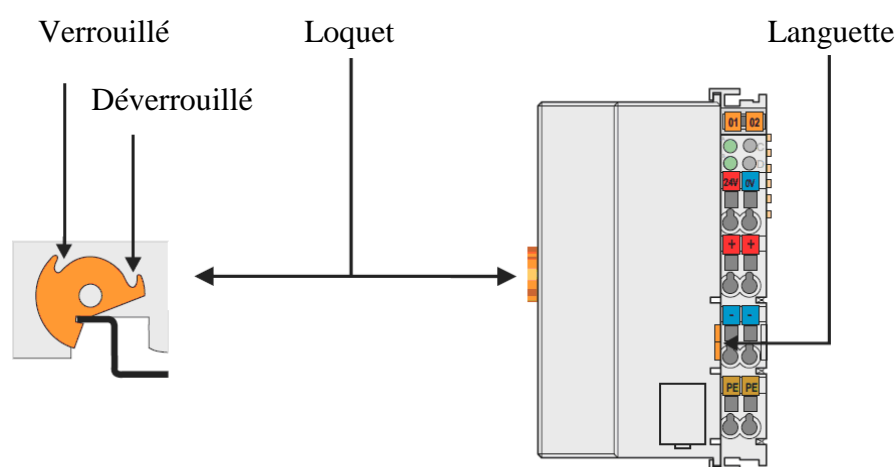
## 5 Installation et câblage

### 5.1 Montage sur le rail

Le contrôleur, tout comme les bornes d'entrées/sorties, se montent sur un rail standard DIN 35. Le contrôleur et les bornes d'entrées/sorties disposent d'un point de verrouillage individuel sur le rail.

Le contrôleur dispose en plus d'un loquet de verrouillage qui le maintient perpendiculaire au rail et assure ainsi un contact optimal au niveau du bus interne.

-Vérifier que ce loquet est bien verrouillé :



Pour extraire le contrôleur du rail, déverrouiller le loquet et tirer sur la languette orange. Les bornes d'E/S disposent également d'une languette orange pour être extraites individuellement.

La position de montage peut être indifféremment horizontale ou verticale. En position verticale, il faut ajouter une butée d'arrêt en bas du bornier d'E/S. En position horizontale, l'utilisation de butées d'arrêt n'est pas nécessaire.

## 5.2 Alimentation

Les contrôleurs programmables permettent le câblage de deux alimentations :  
- l'une pour l'électronique du contrôleur et le bus interne (K-Bus)  
- l'autre pour l'alimentation des capteurs/actionneurs (contacts de puissance)

L'alimentation de l'électronique et du bus interne est repérée sur les contacts 24V et 0V.

L'alimentation des capteurs/actionneurs est repérée sur les contacts + et -. Les deux contacts + et les deux contacts - sont liés ensemble par un pontage interne.

Il est possible d'utiliser la même alimentation pour les deux fonctions. Dans ce cas, deux shunts sont à réaliser au niveau du contrôleur.



---

### Avertissement

WAGO préconise l'utilisation d'alimentations régulées pour l'alimentation de la partie électronique

---



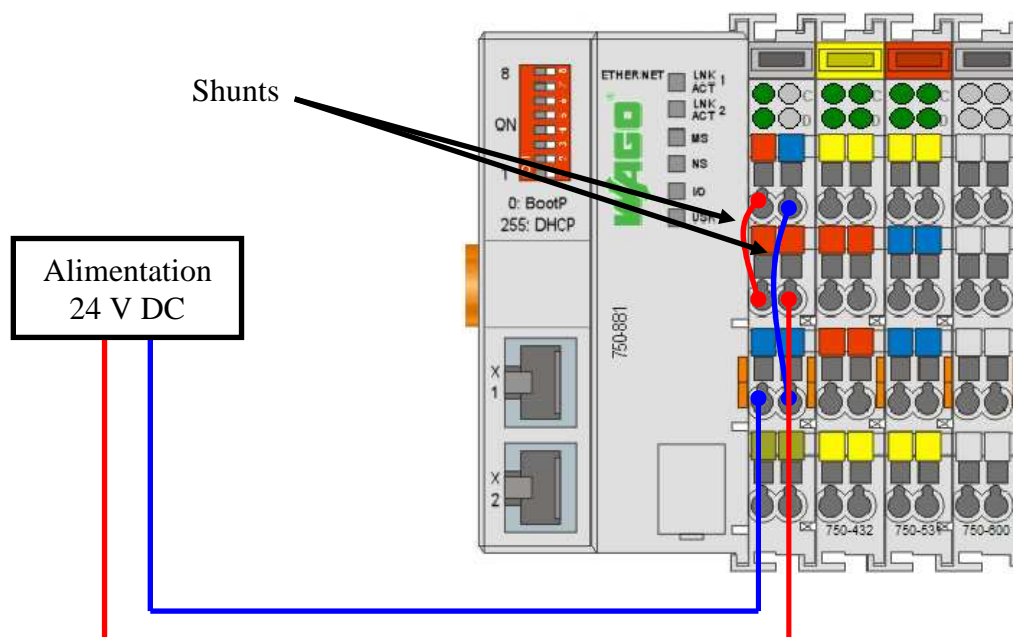
---

### Remarque

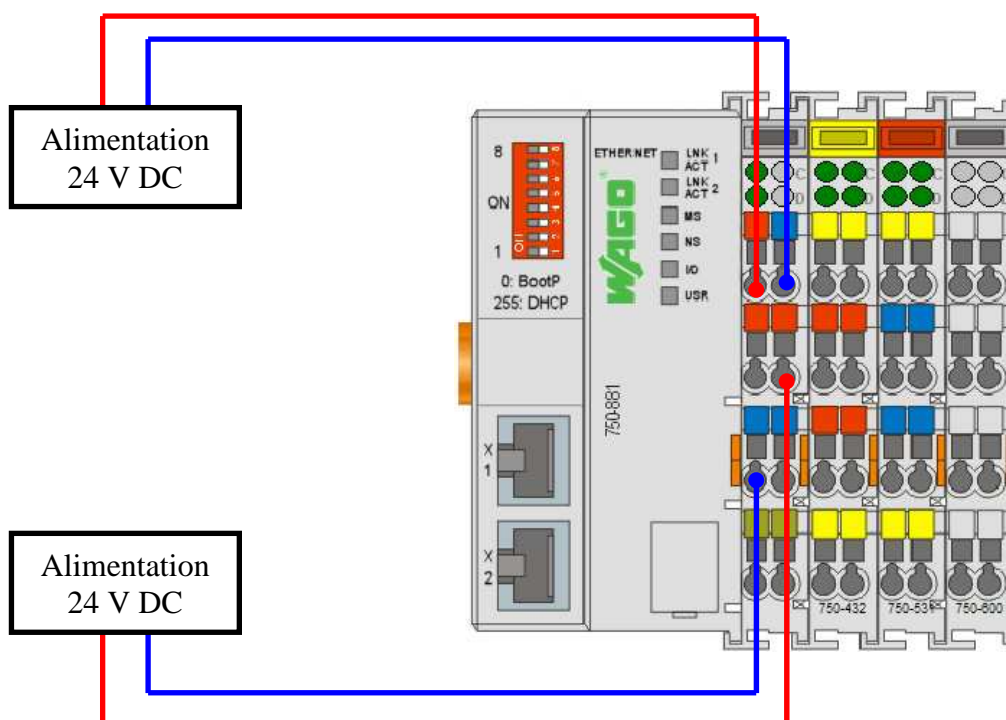
WAGO propose une gamme complète d'alimentations à découpage dans la série 787 (EPSITRON®)

---

### 5.2.1 Câblage avec une unique alimentation



### 5.2.2 Câblage avec deux alimentations

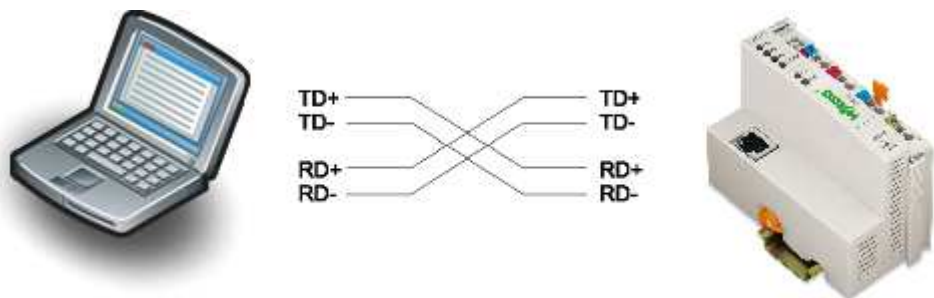


## 5.3 Réseau Ethernet

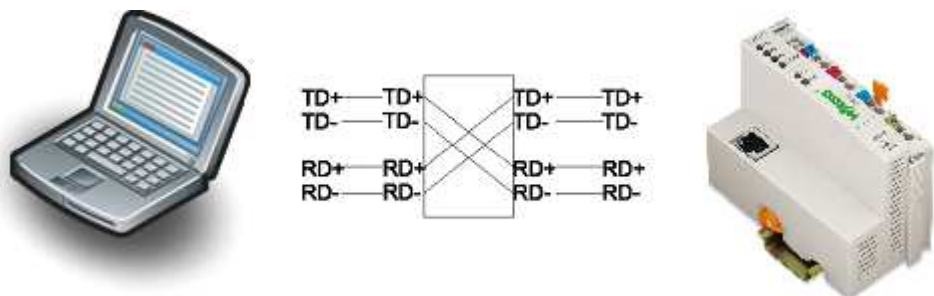
Excepté le contrôleur 750-841, tous les contrôleurs Ethernet 32 bits disposent de la fonctionnalité « Auto MDI/MDI-X », qui permet d'utiliser n'importe quel type de câble (droit/croisé), quelque soit le type de câblage.

Pour le 750-841 uniquement, il est nécessaire de tenir compte du type de câblage.

Si le PC de programmation est relié directement au contrôleur 750-841, il faut utiliser un câble **croisé** :



Si le PC de programmation est relié au contrôleur 750-841 au travers d'un switch, il faut utiliser des câbles **droits** :



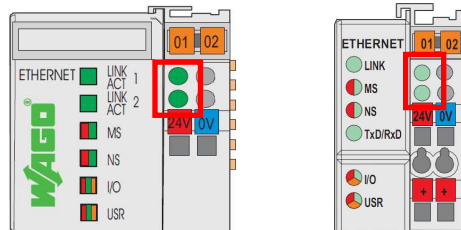
### Remarque

WAGO propose une gamme de switches Ethernet industriels dans la série 852. Tous ces switches supportent également la fonctionnalité « Auto MDI/MDI-X ».

## 5.4 Vérification de l'établissement du lien physique

- Mettre le contrôleur sous tension

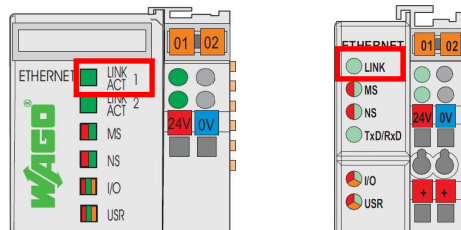
Les deux LEDs d'alimentation doivent alors être allumées :



- Sur les contrôleurs disposant d'un seul port, connecter le câble Ethernet sur le port RJ45. La LED **LINK** doit être allumée fixe.

- Sur les contrôleurs 750-882, connecter le câble Ethernet sur le port repéré **X1**. La LED **LINK/ACT 1** doit être allumée, et clignote en cas de trafic réseau.

- Sur les autres contrôleurs disposant de 2 ports, se relier indifféremment sur les ports repérés **X1** ou **X2**. La LED **LINK/ACT** correspondante doit être allumée, et clignote en cas de trafic réseau.



## 6 Paramétrage IP

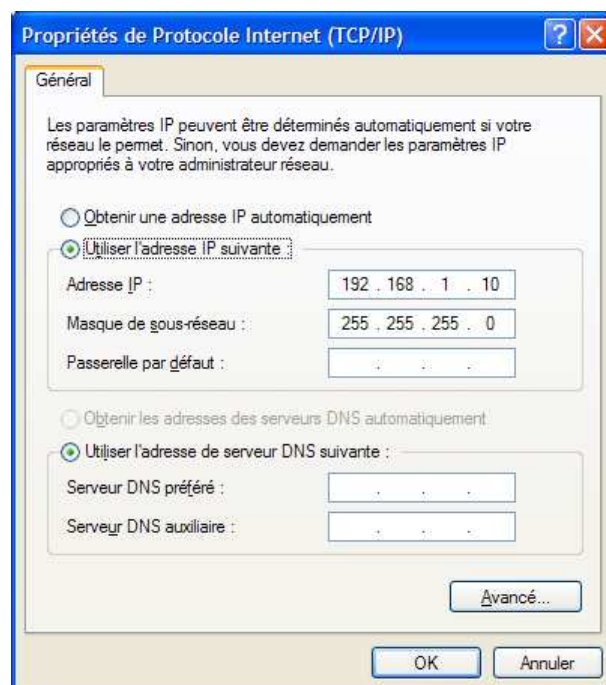
Afin de communiquer directement ensemble, le contrôleur et le PC de programmation doivent être situés dans le même **sous-réseau IP**. Si les adresses IP ne sont pas imposées sur l'installation, WAGO préconise l'utilisation du sous-réseau suivant : **192.168.1.0**, ce qui signifie que les adresses IP commencent toutes par 192.168.1. La valeur du dernier octet de l'adresse IP peut être comprise entre 1 et 254. Ceci correspond au paramétrage par défaut sur les contrôleurs disposant de micro-interrupteurs pour le réglage du dernier octet de l'adresse IP.

La première étape consiste à affecter au PC une adresse IP fixe.

### 6.1 Paramétrage du PC

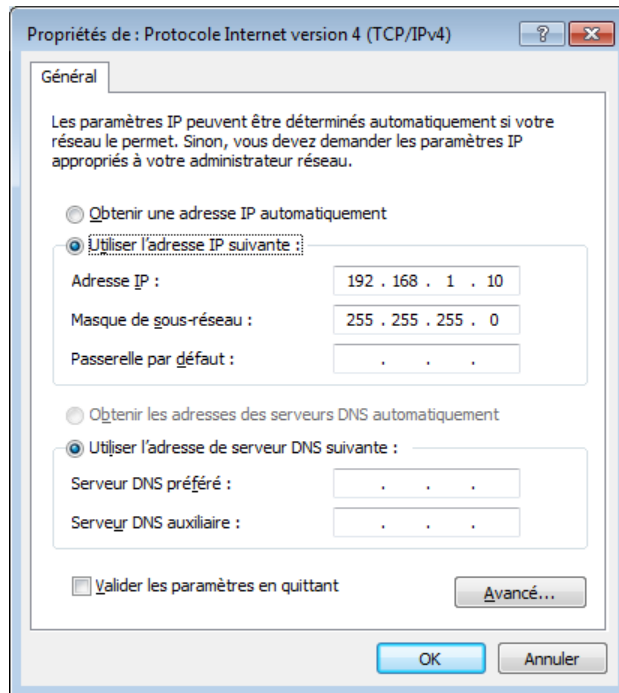
#### 6.1.1 Windows 2000/XP

- Cliquer sur *Démarrer \ Panneau de configuration \ Connexions réseau et Internet \ Connexion réseau*
- Faire un clic droit sur **Connexion au réseau local**, puis cliquer sur **Propriétés**
- Sélectionner **Protocole Internet (TCP/IP)**, puis cliquer sur le bouton **Propriétés**
- Saisir l'**adresse IP** à affecter au PC (ici, 192.168.1.10) ainsi que le **Masque de sous-réseau** (ici, 255.255.255.0), puis valider par **OK** :



## 6.1.2 Windows 7

- Cliquer sur *Démarrer \ Panneau de configuration \ Réseau et Internet \ Centre Réseau et partage \ Modifier les paramètres de la carte*
- Faire un clic droit sur **Connexion au réseau local**, puis cliquer sur **Propriétés**
- Sélectionner **Protocole Internet version 4 (TCP/IPv4)**, puis cliquer sur le bouton **Propriétés**
- Saisir l'**adresse IP** à affecter au PC (ici, 192.168.1.10) ainsi que le **Masque de sous-réseau** (ici, 255.255.255.0), puis valider par **OK** :





## 6.2 Paramétrage du contrôleur

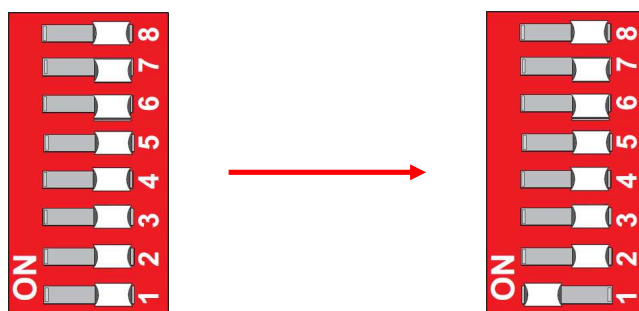
### 6.2.1 Paramétrage par micro-interrupteurs

Certains contrôleurs disposent de micro-interrupteurs permettant de régler le dernier octet de l'adresse IP (750-88x, 750-871). Par défaut, les 3 premiers octets sont fixés en usine à 192.168.1.xxx.

Si le PC a été paramétré avec ce même sous-réseau, il n'est pas nécessaire d'utiliser de logiciel pour régler l'adresse IP.

Il suffit de basculer les micro-interrupteurs correspondants :

- basculer le micro-interrupteur n°1

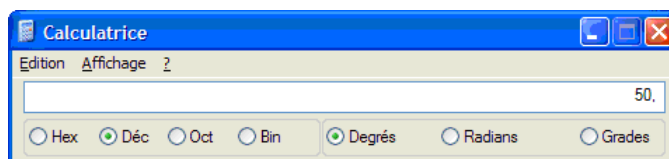


- redémarrer le contrôleur en créant une coupure d'alimentation, afin qu'il prenne en compte ce nouveau réglage

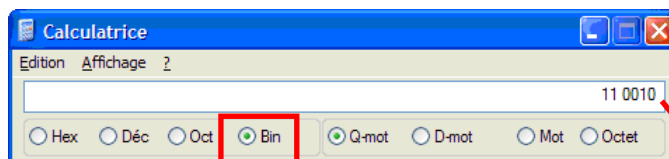
Le contrôleur a maintenant l'adresse IP **192.168.1.1**.

Pour calculer la position pour des valeurs plus élevées, la méthode la plus simple est d'utiliser la calculatrice de Windows en mode **Scientifique** (*Affichage\Scientifique*).

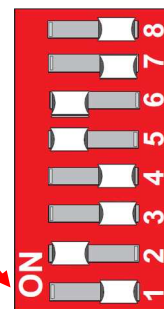
- Saisir la valeur décimale de l'octet (ex : 50)



- Cliquer sur le bouton **Bin**



- Reporter l'état des bits sur les micro-interrupteurs, en commençant par les bits de poids les plus faibles



## 6.2.2 Paramétrage par logiciel

Si l'on ne dispose pas du câble de configuration USB (ou RS232), se reporter au chapitre **Annexe B : affectation de l'adresse IP par protocole BootP**.

Sinon, la méthode la plus simple est l'utilisation du logiciel **WAGO Ethernet Settings**, décrite dans le présent chapitre.

Pour paramétrer le contrôleur par logiciel, il faut tout d'abord installer le pilote de câble USB, ainsi que le logiciel **WAGO Ethernet Settings**.

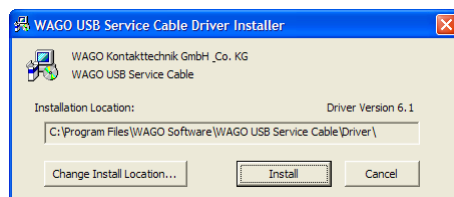


### Avertissement

Ne pas connecter le câble USB au PC avant d'avoir installé le pilote !

Pour installer le pilote du câble USB :

- extraire le contenu du dossier compressé **759-923**
- Aller jusqu'au dossier  
*WAGOServiceCable\_DriverInstall\WIN2k\_XP\_VISTA\_WIN7*,
- exécuter le fichier **Setup.exe**
- Cliquer sur le bouton **Install**



Pour installer le logiciel **WAGO Ethernet Settings** :

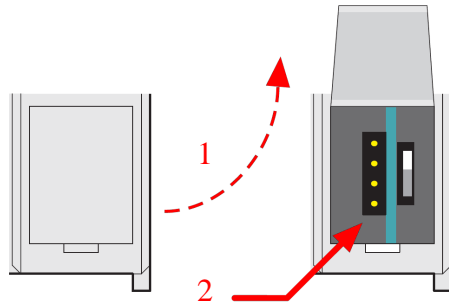
- extraire le contenu du dossier compressé **WAGO\_EthernetSettings\_Setup(Vx.y.z).zip**, puis exécuter **WAGO\_EthernetSettings\_Setup(Vx.y.z).exe**
- sélectionner la langue, puis valider par **OK**
- valider l'écran de bienvenue en cliquant sur **Next**
- Cocher le bouton **I accept the terms of the Licence Agreement**, puis cliquer sur **Next**
- renseigner le nom d'utilisateur et la société, puis cliquer sur **Next**
- sélectionner le chemin d'installation, puis cliquer sur **Install**
- valider la fenêtre de création de raccourcis, puis cliquer sur **Next**
- décocher la case **Run WAGO Ethernet Settings**, puis cliquer sur **Finish**



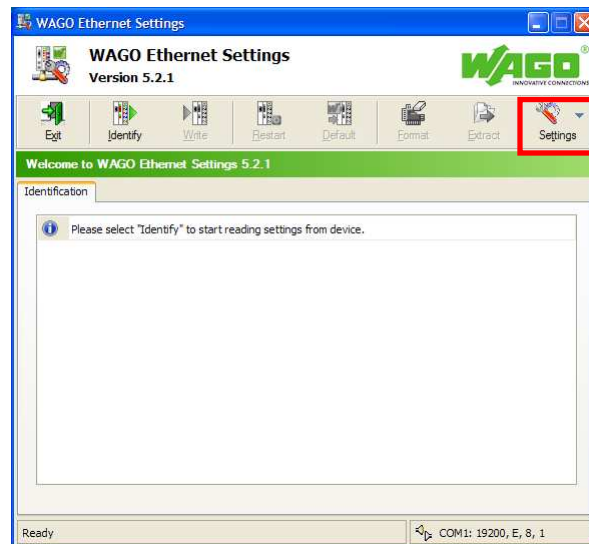
### Avertissement

Le câble USB ne doit jamais être connecté à chaud, lorsque le contrôleur est sous tension. Risque de destruction matérielle !

- Mettre le contrôleur hors tension
- Ouvrir la trappe située en face avant du contrôleur (1) :



- Insérer délicatement le connecteur spécifique du câble sur les 4 broches situées du côté gauche du circuit imprimé (2)
- Raccorder le connecteur USB du câble au PC
- Remettre le contrôleur sous tension
- Lancer **WAGO Ethernet Settings** à partir du menu *Démarrer \ Programmes \ WAGO Software \ WAGO Ethernet Settings*
- Cliquer sur le bouton **Settings** :



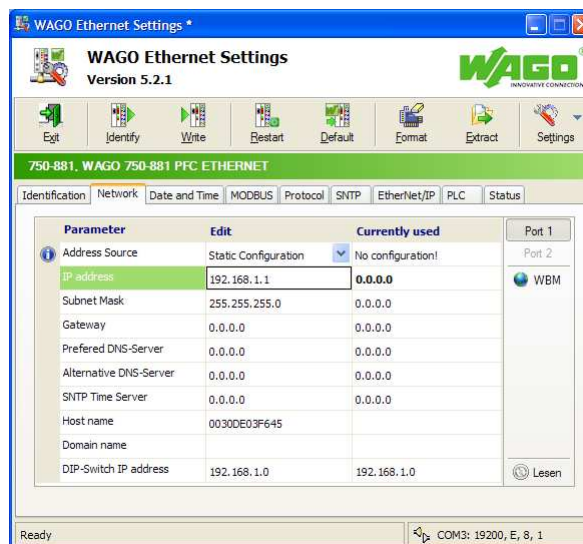
- Dans le groupe **Settings**, champ **Connection**, sélectionner le port COM qui a été créé suite à la connexion du câble USB : **COMx: WAGO USB Service Cable**).



- Valider la fenêtre par **OK**
- Cliquer ensuite sur le bouton **Identify**
- Cliquer sur l'onglet **Network**

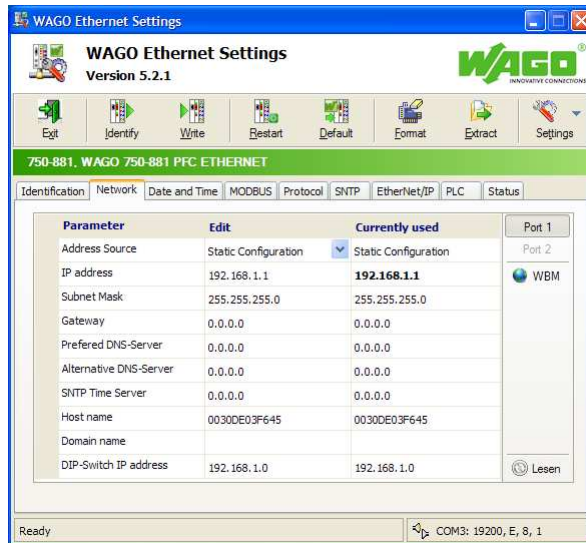
Dans la colonne **Edit** :

- sélectionner **Static Configuration** dans le champ **Address Source**
- renseigner l'adresse à affecter au contrôleur dans le champ **IP address** (ici, 192.168.1.1)



- Cliquer sur **Write**

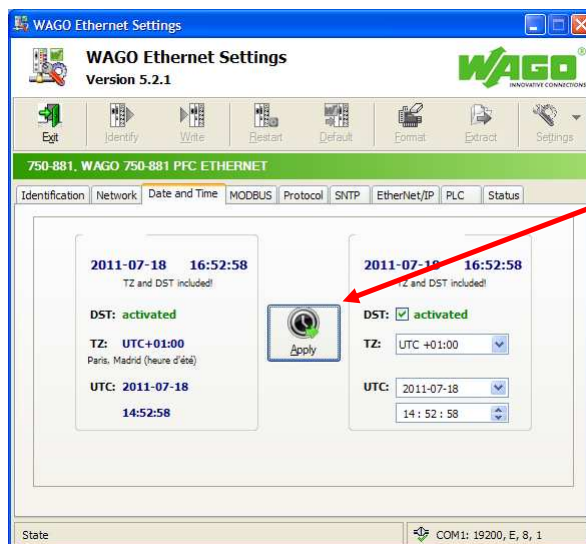
Le contrôleur redémarre, puis les nouveaux paramètres sont affichés dans la colonne **Currently Used** :



### Remarque

Pour les contrôleurs disposant de micro-interrupteurs de réglage de l'adresse IP : pour que l'adresse paramétrée soit prise en compte, il est nécessaire que les micro-interrupteurs de réglage soient sur 0.

- Ensuite, cliquer sur l'onglet **Date/Time**
- Cliquer sur le bouton **Apply**, pour mettre à l'heure l'horloge du contrôleur :



### Remarque

En période d'heure d'été, il est possible que l'horloge du contrôleur soit décalée d'une heure par rapport à celle du PC. Le passage automatique heure d'été / heure d'hiver est abordé plus loin dans ce document.

- Quitter **WAGO Ethernet Settings** en cliquant sur **Exit**

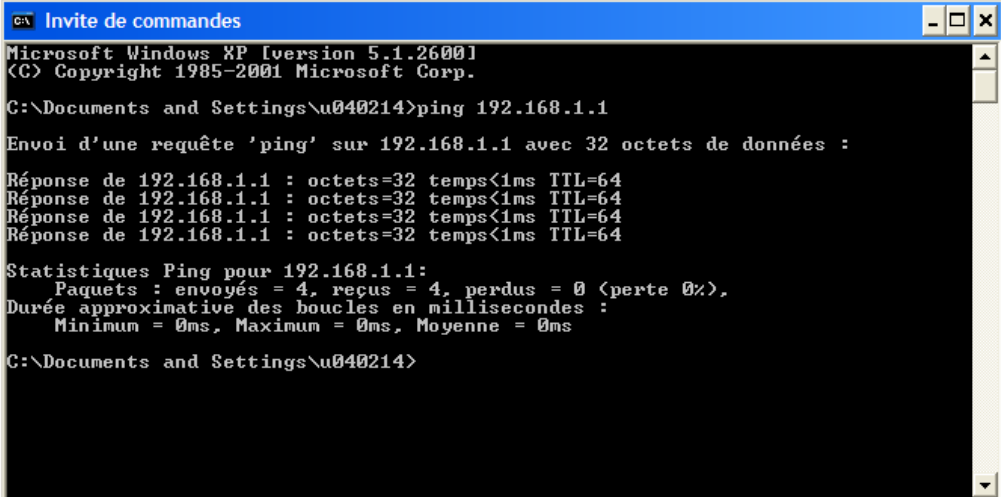
## 6.3 Test de la communication

- Ouvrir une **Invite de commandes MS-DOS** depuis le menu *Démarrer \ Programmes \ Accessoires \ Invite de commandes MS-DOS*

- Saisir la commande **ping**, suivie d'un espace et de l'adresse IP du contrôleur

Exemple : ping 192.168.1.1

Lorsque la communication est correcte, l'écran suivant apparaît :



```

C:\> Invite de commandes
Microsoft Windows XP [version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\u040214>ping 192.168.1.1
Envoi d'une requête 'ping' sur 192.168.1.1 avec 32 octets de données :
Réponse de 192.168.1.1 : octets=32 temps<1ms TTL=64
Réponse de 192.168.1.1 : octets=32 temps<1ms TTL=64
Réponse de 192.168.1.1 : octets=32 temps<1ms TTL=64
Réponse de 192.168.1.1 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 192.168.1.1:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms

C:\Documents and Settings\u040214>
  
```

Si le texte « Délai d'attente de la demande dépassé. » apparaît, vérifier :

- que l'adresse IP du PC et celle du contrôleur sont bien dans le même sous-réseau
- que la LED **I/O** du contrôleur est bien allumée (vert fixe)
- que la LED **LINK** ou **LINK/ACT** du contrôleur est bien allumée



### Remarque

Si la LED **I/O** n'est pas au vert fixe, se reporter au chapitre **Annexe A : LED I/O**, afin de déterminer quel est le problème rencontré.

Une fois la communication validée, fermer l'**Invite de commandes** en tapant la commande **exit**.

Si l'adresse IP a été paramétrée avec **WAGO Ethernet Settings** :

- mettre le contrôleur hors tension
- déconnecter le câble USB
- remettre le contrôleur sous tension

Pour toute la suite, la communication se fera directement par le réseau Ethernet.

## 7 Accès au serveur web

Les contrôleurs Ethernet 32 bits disposent d'un serveur web, qui offre à la fois des possibilités de diagnostic et des possibilités de paramétrage.

Pour y accéder, ouvrir un navigateur web et saisir l'adresse IP du contrôleur dans la barre d'adresse :

The screenshot shows the WAGO Ethernet Web-Based Management interface. The browser address bar contains the URL `http://192.168.1.1/webserver/ln`. The page title is "WAGO Ethernet Web-Based Management". The interface includes a navigation menu on the left with the following items: Information, Ethernet, TCP/IP, Port, SNMP, SNMP V3, Watchdog, Clock, Security, PLC, Features, IO config, Disk Info, and WebVisu. The main content area displays the following sections:

- Status information**
- Coupler details**

Order number	750-881
Mac address	0030DE03F645
Firmware revision	01.02.05 (03)
- Actual network settings**

IP address	192.168.1.1
	Determined by Dip Switch
Subnet mask	255.255.255.0
Gateway	0.0.0.0
Hostname	0030DE03F645
Domainname	
(S)NTP-Server	0.0.0.0
DNS-Server 1	0.0.0.0
DNS-Server 2	0.0.0.0
- Module status**

State Modbus Watchdog:	Disabled
Error code:	0
Error argument:	0
Error description:	Coupler running, OK

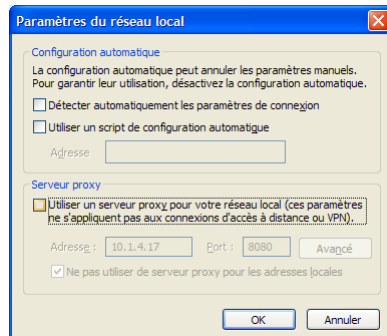
Si la page n'apparaît pas, se reporter à la remarque en page suivante.



### Remarque

Certains PC sont configurés pour passer par un serveur proxy pour l'accès à Internet. Si la page web ne s'affiche pas, il est possible qu'un serveur proxy soit utilisé. Pour dévalider cette option (sous Internet Explorer) :

- Outils \ Options Internet
- Onglet **Connexions**, bouton **Paramètres réseau**
- Décocher la case **Utiliser un serveur proxy**



- Valider les différentes fenêtres, puis saisir de nouveau l'adresse IP du contrôleur dans la barre d'adresse du navigateur

Si la configuration de l'adresse IP a été réalisée à l'aide des micro-interrupteurs, il faut mettre à l'heure l'horloge :

- Cliquer sur la page **Clock**
- le nom d'utilisateur est **admin**, et le mot de passe est **wago** :





- Régler la date et l'heure, puis valider par **SUBMIT** (pour la France, laisser le champ **Timezone** à 1)

**Clock configuration**

Configuration Data	
Time on device	12:18:23
Date (YYYY-MM-DD)	2011-03-15
Timezone (+/- hour:minute)	+1:00
Daylight Saving Time (DST)	<input type="checkbox"/>
12 hour clock	<input type="checkbox"/>



### Remarque

Si l'on souhaite modifier les 3 premiers octets de l'adresse IP, se rendre sur la page **TCP/IP**, et modifier le paramètre **Switch IP-Address** :

EEPROM Configuration Data	
IP-Address	192.168.1.1
Subnet Mask	255.255.255.0
Gateway	192.168.2.1
Hostname	0030DE03F645
Domain name	
DNS-Server1	0.0.0.0
DNS-Server2	0.0.0.0
Switch IP-Address	192.168.1
(S)NTP-Server	0.0.0.0
SNTP Update Time (sec, max. 65535)	3600

- Cliquer ensuite sur **SUBMIT**
  - Redémarrer le contrôleur pour que le réglage soit pris en compte
- Il est possible de redémarrer le contrôleur depuis le serveur web, en allant sur la page **Security**, puis en cliquant sur le bouton **Software Reset**.



### Informations complémentaires

L'ensemble des paramètres accessibles depuis le serveur web est décrit dans le manuel technique du contrôleur. Le manuel technique peut être téléchargé sur [www.wago.fr](http://www.wago.fr), et est également présent sur le DVD **AUTOMATION**

#### Tools and Docs :

> Services > Documentation Technique > Documentation > WAGO-I/O-SYSTEM 750 > Fieldbus Coupler and Programmable Fieldbus Controller

- Fermer le navigateur web

## 8 CoDeSys

CoDeSys (*Controller Development System*) est un environnement de développement pour automates, conforme à la norme CEI 61131-3. Cet outil est multi-constructeurs, et notamment utilisé depuis de nombreuses années par WAGO pour la programmation de ses contrôleurs programmables série 750, ses PC industriels embarqués série 758 ou encore ses terminaux tactiles série 762 (PERSPECTO®).

CoDeSys permet l'édition du programme automate, et dispose également d'un éditeur de visualisations. Sur les contrôleurs Ethernet 32 bits, les visualisations peuvent être rendues accessibles depuis le serveur web, en tant que visualisations web.

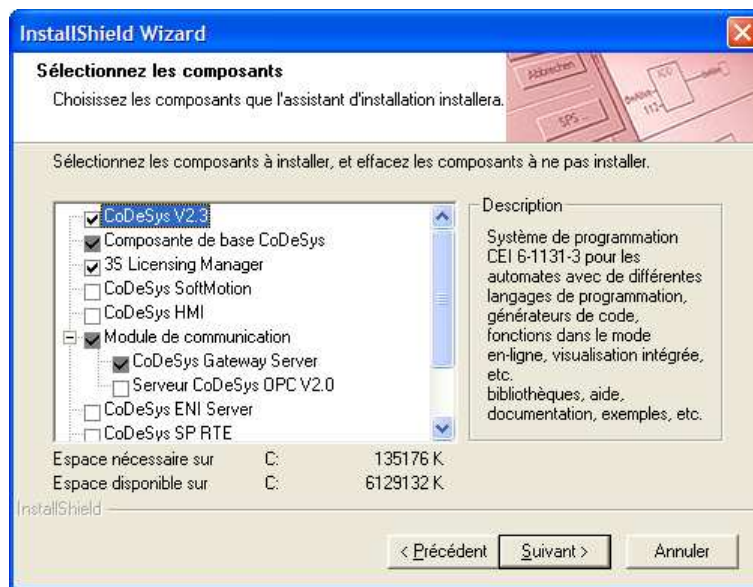
### 8.1 Installation

- Extraire les fichiers du dossier compressé **CoDeSys\_2.3.9.xx.zip**
- Exécuter **Setup.exe**
- Sélectionner la langue d'installation, puis cliquer sur OK



- Valider l'écran de bienvenue en cliquant sur **Suivant**
- Accepter le Contrat de licence en cliquant sur **Oui**
- Sélectionner le chemin cible et cliquer sur **Suivant**

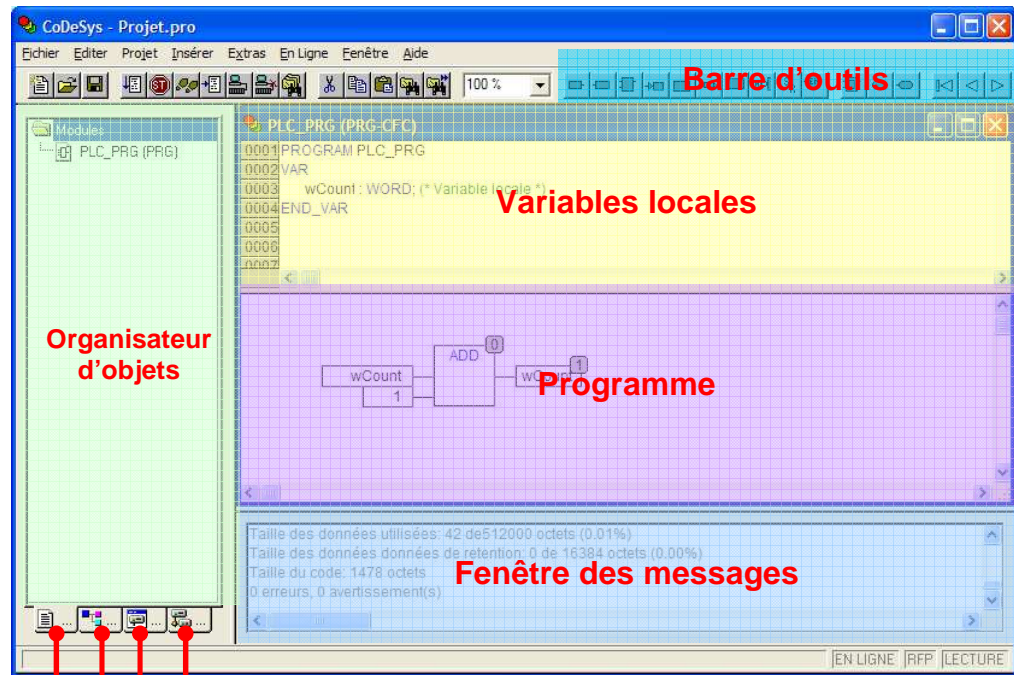
- Sélectionner l'installation des composants suivants, puis cliquer sur **Suivant**



- Sélectionner le dossier programme, puis cliquer sur **Suivant**
- Valider la vue générale des configurations en cliquant sur **Suivant**
- Patienter jusqu'à ce que l'installation se termine

## 8.2 Aperçu de l'éditeur

CoDeSys 2.3 se présente comme suit :



Onglet **Ressources** (variables globales, configuration automate...)

Onglet **Visualisations** (fenêtres de visualisation)

Onglet **Types de données** (structures, énumérations)

Onglet **Modules** (programmes, blocs fonctionnels et fonctions)

## 8.3 Les langages de programmation

CoDeSys intègre 6 langages de programmation, qui peuvent être librement combinés au sein d'un même projet. Cette flexibilité permet au développeur de choisir le langage le plus approprié à son besoin.

Les 5 premiers langages sont définis dans la norme CEI 61131-3. Le langage CFC est un langage complémentaire, particulièrement convivial.

IL	<i>Instruction List</i>	Liste d'Instructions
LD	<i>Ladder Diagram</i>	Langage à Contacts
FBD	<i>Function Block Diagram</i>	Schéma en blocs fonctionnels
SFC	<i>Sequential Function Chart</i>	Diagramme Fonctionnel en Séquences
ST	<i>Structured Text</i>	Littéral Structuré
CFC	<i>Continuous Function Chart</i>	Langage CFC

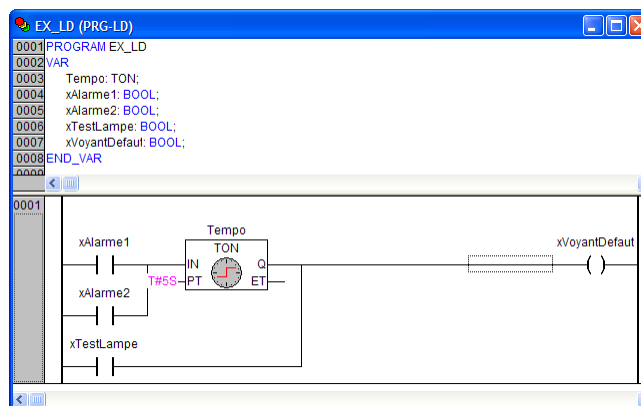
Exemple de programme en langage **IL** :

```

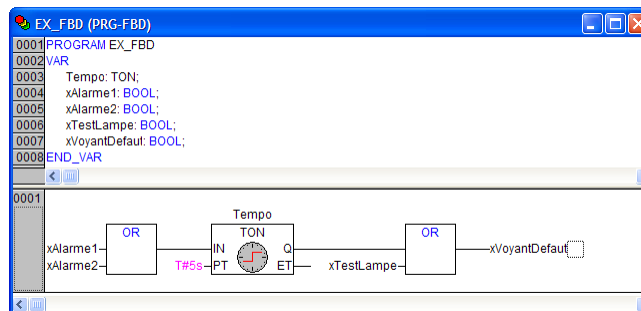
EX_IL (PRG-IL)
0001 PROGRAM EX_IL
0002 VAR
0003   Tempo: TON;
0004   xAlarme1: BOOL;
0005   xAlarme2: BOOL;
0006   xTestLampe: BOOL;
0007   xVoyantDefault: BOOL;
0008   xTmpVar: BOOL;
0009 END_VAR
0001 LD   xAlarme1
0002 OR   xAlarme2
0003 ST   xTmpVar
0004
0005 CAL   Tempo(IN := xTmpVar, PT := T#5s)
0006
0007 LD   Tempo.Q
0008 OR   xTestLampe
0009 ST   xVoyantDefault

```

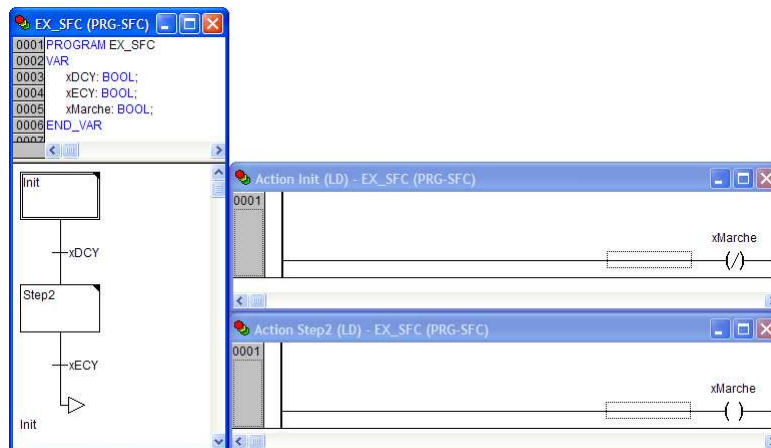
Exemple de programme en langage **LD** :



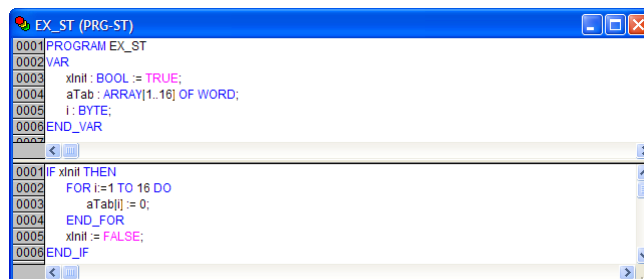
Exemple de programme en langage **FBD** :



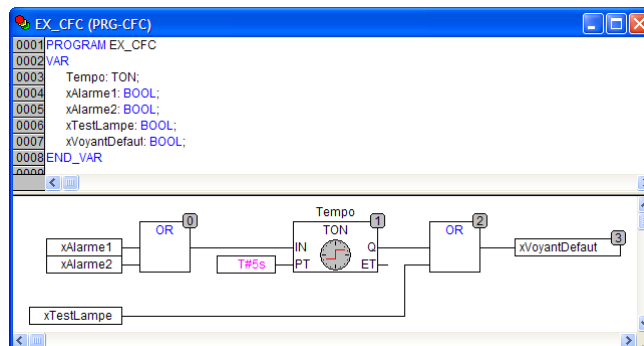
Exemple de programme en langage SFC :



Exemple de programme en langage ST :



Exemple de programme en langage CFC :



## 8.4 Les modules

Les modules, ou « POU » (*Program Organization Units*), se décomposent en 3 types :

- les programmes
- les blocs fonctionnels
- les fonctions

Chaque module peut être programmé dans un langage différent, au sein d'un même projet.

### 8.4.1 Les programmes

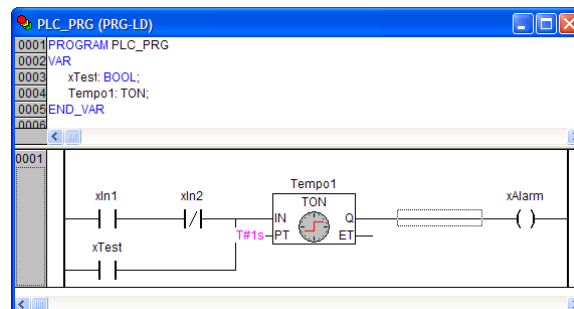
Un programme est unique et ne comporte pas la notion d'instance. En général, il n'a pas de paramètres d'entrées/sorties. Un programme est visible depuis l'ensemble du projet.

Les variables locales d'un programme conservent leur état entre deux appels (variables statiques).

Un programme peut ensuite faire appel à d'autres programmes, tout comme à des blocs fonctionnels et des fonctions.

Dans le cadre d'une implémentation mono-tâche, le programme principal s'appelle « PLC\_PRG ».

Exemple :



### 8.4.2 Les blocs fonctionnels

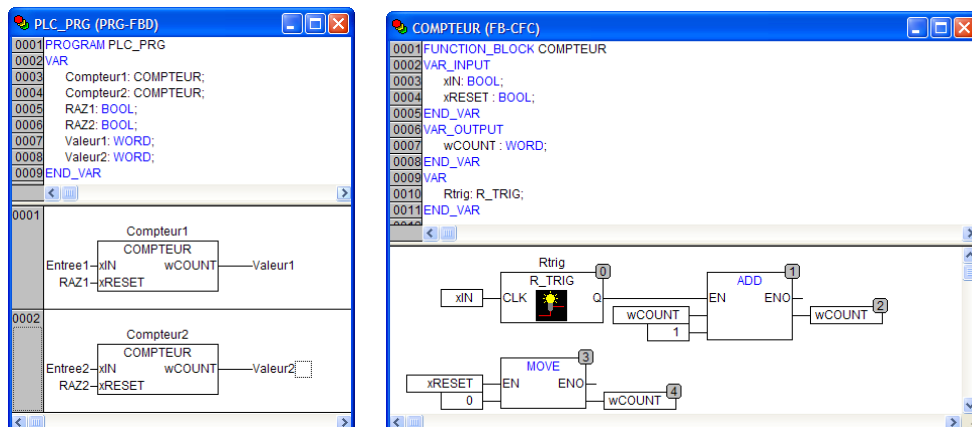
Les blocs fonctionnels sont des objets comportant la notion d'instance. Ils peuvent avoir zéro, un ou plusieurs paramètres d'entrée, et zéro, un ou plusieurs paramètres de sortie. Chaque instance dispose de variables locales, mais le code est unique pour toutes les instances. Comme pour les programmes, les variables locales à un bloc fonctionnel conservent leurs valeurs entre deux appels.

L'utilisation de blocs fonctionnels est particulièrement intéressante pour éviter de dupliquer plusieurs fois le même code dans un programme. Elle permet de faciliter la lecture du code, ainsi que sa maintenance : le code n'étant écrit

qu'une seule fois, il n'y a pas de risque d'erreur « de copier/coller » en cas de modification.

Les blocs fonctionnels sont par exemple utilisés pour les temporisateurs, les compteurs, les régulateurs PID, etc. Il est possible de développer ses propres blocs fonctionnels.

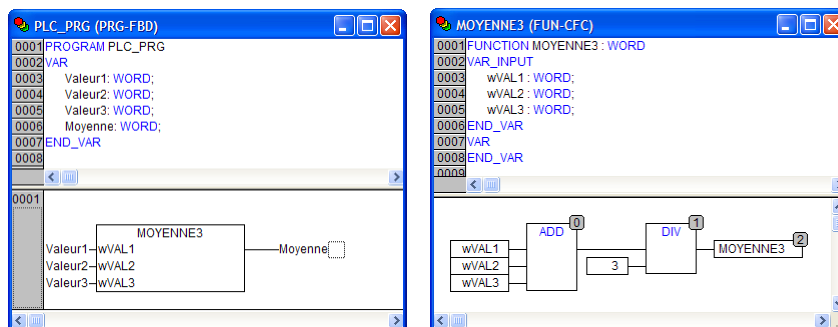
Exemple :



### 8.4.3 Les fonctions

Les fonctions disposent de un ou plusieurs paramètres d'entrée et ne renvoient qu'une seule valeur. Les fonctions ne contiennent pas de variables statiques, c'est-à-dire que l'état des variables locales à une fonction n'est pas conservé entre deux appels.

Exemple :





## 8.5 Les types de données

Les types de données standard sont tous définis dans la norme CEI 61131-3.

### 8.5.1 Les booléens

Le type **BOOL** correspond à une variable booléenne. Ce type de variable ne peut prendre que deux états : **TRUE** (vrai) ou **FALSE** (faux).

Exemples :

```
xStart : BOOL ;
xFirstCycle : BOOL := TRUE ;
```

### 8.5.2 Les entiers

Les entiers occupent de 8 à 32 bits de mémoire. Ils peuvent être signés ou non. Le tableau ci-dessous synthétise les types d'entiers disponibles :

Type	Valeur minimale	Valeur maximale	Espace mémoire
<b>BYTE</b>	0	255	8 bits
<b>WORD</b>	0	65535	16 bits
<b>DWORD</b>	0	4294967295	32 Bits
<b>SINT</b>	-128	127	8 bits
<b>USINT</b>	0	255	8 bits
<b>INT</b>	-32768	32767	16 bits
<b>UINT</b>	0	65535	16 bits
<b>DINT</b>	-2147483648	2147483647	32 bits
<b>UDINT</b>	0	4294967295	32 bits



#### Remarque

Les valeurs entières peuvent être saisies en décimal, en hexadécimal ou en binaire. Pour saisir une valeur en hexadécimal, il suffit d'ajouter 16# devant la valeur. Pour une valeur en binaire, ajouter 2#. En binaire, il est également possible de grouper les bits par 4, en les séparant par des tirets bas ( \_ ), afin d'améliorer la lisibilité.

Exemples :

```
Déclaration : MonMot: WORD;
Code (en ST) : MonMot := 250; (* Décimal *)
                MonMot := 16#FA; (* Hexadécimal *)
                MonMot := 2#11111010; (* Binaire *)
                MonMot := 2#1111_1010; (* Binaire, en groupes de 4 bits *)
```

### 8.5.3 Les réels

Le type **REAL** est utilisé pour les variables réelles (« nombres à virgules »). Une variable de type **REAL** occupe 32 bits. Le séparateur est le point. Le format utilisé est IEEE 754.

Exemple :

PI : **REAL** := 3.141593;

### 8.5.4 Les données temporelles

Il existe 4 types de données temporelles. Chacun de ces types occupe 32 bits de mémoire.

Le type **TIME** est utilisé pour exprimer des durées, et sa résolution est la milliseconde. C'est notamment le type utilisé en entrée des temporisateurs.

Le type **TIME\_OF\_DAY** permet d'indiquer l'heure du jour. Sa résolution est également à la milliseconde.

Le type **DATE** permet d'indiquer une date, et contient un nombre de secondes depuis le 1<sup>er</sup> janvier 1970.

Le type **DATE\_AND\_TIME** permet de spécifier l'heure du jour en plus de la date, et contient également un nombre de secondes depuis le 1<sup>er</sup> janvier 1970.

Type	Résolution	Exemple	Etendue
<b>TIME</b>	1ms	T#5ms T#1h5s130ms	0 à 49d17h2m47s295ms (4194967295 ms)
<b>TIME_OF_DAY</b> (ou <b>TOD</b> )	1ms	TOD#00:00:00 TOD#15:36:30.123	00:00:00 à 23:59:59.999
<b>DATE</b>	1s	D#2010-09-01 D#1980-05-24	1970-01-01 à 2106-02-06
<b>DATE_AND_TIME</b> (ou <b>DT</b> )	1s	DT#2010-09-01-14:00:00 DT#1980-05-24-03:10	1970-01-01-00:00:00 à 2106-02-06-06:28:15



#### Remarque

Les données temporelles peuvent ensuite être converties en entiers à l'aide d'opérateurs de conversion.

Exemple (pour un temps d'une minute) :

MyDWORD := **TIME\_TO\_DWORD**(T#1m); (\* Le résultat est 60000, puisque le type **TIME** contient un nombre de millisecondes \*)

## 8.5.5 Les Chaînes de caractères

Le type de données **STRING** permet le stockage de chaînes de caractères. Le format utilisé est dit « à zéro terminal », c'est-à-dire que la chaîne est terminée par un caractère nul (code ASCII 00).

La taille de la chaîne peut être spécifiée lors de la déclaration. Si la taille n'est pas spécifiée, elle sera dimensionnée à la valeur par défaut, soit 80 caractères.

La taille maximale d'une chaîne de caractères est limitée par la cible (le contrôleur) utilisée.

Néanmoins, les fonctions de traitement sur les chaînes de caractères étant limitées à 255, il est préférable de ne pas excéder cette valeur.

Exemples :

```
sMyString : STRING := 'Alarm tank 1$R$L';
MaChaine : STRING(20) := 'ABCD';
```



### Remarque

L'espace mémoire utilisé est toujours augmenté d'un octet à cause du zéro terminal. Par exemple, une variable de type **STRING**(255) occupera 256 octets en mémoire.

## 8.5.6 Les tableaux

Il est possible de déclarer des tableaux de données standard, structurées, ou encore des tableaux d'instances de blocs fonctionnels. Les tableaux peuvent avoir de 1 à 3 dimensions. Les index de début et de fin sont libres, il n'est pas nécessaire de commencer à 0.

Exemples :

```
myTab : ARRAY[1..10] OF INT ; (* Tableau d'entiers *)
aTempo : ARRAY[0..255] OF TON; (* Tableau d'instances de FB *)
DaliValues : ARRAY[1..5,1..64] OF BYTE; (* Tableau d'octets à deux dimensions *)
```

## 8.5.7 Les structures et les énumérations

Il est possible de créer des données structurées, afin qu'une variable contienne plusieurs valeurs.

Exemple :

```
TYPE typRecette :  
  STRUCT  
    wProduit1 : WORD;  
    wProduit2 : WORD;  
    tDuree    : TIME;  
  END_STRUCT  
END_TYPE
```

Ensuite, on accède aux composants de structures au moyen de la syntaxe suivante :

<Nom\_de\_structure>.<Nom\_de\_composant>

Exemple :

```
Déclaration :   MaRecette : typRecette;  
Code (en ST) : MaRecette.wProduit1 := 200;
```

Les énumérations permettent d'associer des noms symboliques à des valeurs numériques, afin de faciliter la programmation et la lisibilité du programme.

Les énumérations sont souvent utilisées dans les bibliothèques de fonctions.

Exemple :

```
TYPE enumCouleur :  
  (  
    VERT := 0,  
    ORANGE := 1,  
    ROUGE := 2  
  );  
END_TYPE
```

```
Déclaration :   MaCouleur : enumCouleur;  
Code (en ST) : IF MaCouleur = VERT THEN [...] END_IF
```

## 8.6 Les adresses

Le format des adresses est défini dans la norme CEI 61131-3. L'adresse spécifie la zone mémoire, la taille de la variable et l'adresse de celle-ci.

Une adresse commence par le caractère %, suivi de la zone, de la taille et de l'adresse :

### Zone mémoire

I	<i>Input</i>	Entrée
Q	<i>Output</i>	Sortie
M	<i>Memory location</i>	Mémoire interne

### Taille de la variable

X	<i>Bit</i>	Bit	
B	<i>Byte</i>	Octet	(8 bits)
W	<i>Word</i>	Mot	(16 bits)
D	<i>Double word</i>	Double mot	(32 bits)

### Adresse de la variable

Pour les booléens, l'adresse est décomposée en 2 nombres : le premier identifie le numéro du mot et le second le numéro de bit dans le mot.

Exemples :

%MX0.0	Premier bit du premier mot de la mémoire interne
%MX0.15	Dernier bit du premier mot de la mémoire interne
%IX16.8	9ème bit du 17ème mot de la zone d'entrée

Pour les autres variables, un seul nombre est nécessaire :

%MB2	3ème octet de la mémoire interne
%IW0	Premier mot de la zone d'entrée
%QD4	5ème double mot de la zone de sortie



### Remarque

Pour toutes les variables internes au programme, qui ne correspondent pas à des E/S et qui ne sont pas des variables à rendre accessible depuis Modbus/TCP ou EtherNet/IP, il n'est pas nécessaire de spécifier d'adresse.

Exemple : MotInterne1 : **WORD** ;

## 9 Réalisation d'un premier programme

### 9.1 Préambule

Le premier projet consistera à réaliser un programme qui activera la première sortie tout ou rien si les deux premières entrées tout ou rien sont actives. En parallèle, un bouton de test sera prévu.

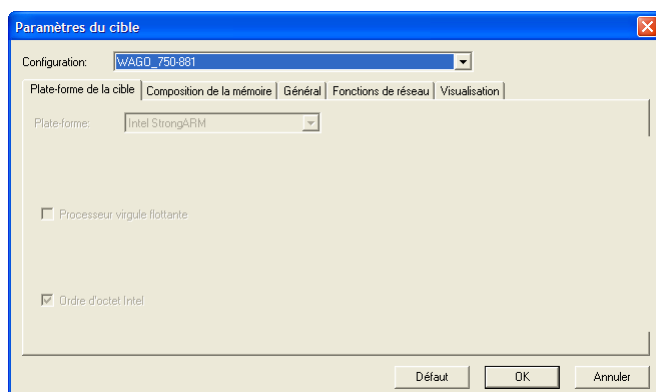
Ceci peut être traduit par la table de vérité suivante :

Entrée 1	Entrée 2	Test	Sortie 1
0	0	0	0
1	0	0	0
0	1	0	0
1	1	0	1
0	0	1	1
1	0	1	1
0	1	1	1
1	1	1	1

Ce programme sera testé sur la configuration matérielle décrite en début de document. Néanmoins, il est possible de suivre ce chapitre sans disposer du matériel.

### 9.2 Sélection de la cible

- ouvrir **CoDeSys V2.3** depuis le menu *Démarrer \ Programmes \ WAGO Software \ CoDeSys for Automation Alliance \ CoDeSys V2.3*
- cliquer *Fichier \ Nouveau*
- sélectionner le contrôleur utilisé (ici, **WAGO\_750-881**) :



- Valider la fenêtre en cliquant sur **OK**



### Remarque

L'ajout de nouvelles fonctionnalités dans les contrôleurs a parfois nécessité la création de nouvelles cibles. Pour certains contrôleurs, il existe donc plusieurs cibles.

Exemple :

**WAGO\_750\_871\_(...-FW04)**

**WAGO\_750\_871\_(FW05-...)**

La première cible est à utiliser pour les versions de firmware de 01 à 04.

La seconde cible est à utiliser pour les versions 05 et supérieures.

La version de firmware peut être lue depuis la page web du contrôleur :

- Relever le nombre entre parenthèses dans le champ **Firmware revision** :

The screenshot displays the WAGO Web-based Management interface. On the left, there is a 'Navigation' menu with options: Information, Ethernet, TCP/IP, Port, SNMP, SNMP V3, Watchdog, Clock, Security, PLC, Features, IO config, and WebVisu. The main content area is divided into three sections: 'Status information', 'Coupler details', and 'Network details'. The 'Coupler details' section contains the following information:

Order number	750-871/000-000
Mac address	0030DE02C40F
Firmware revision	03.01.04 (06)

The 'Network details' section contains the following information:

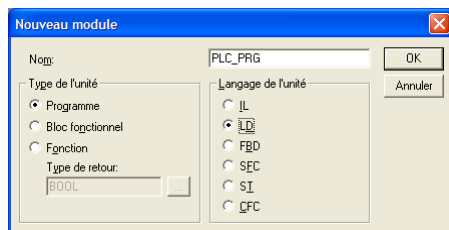
IP address	192.168.1.1
	Determined by Dip Switch
Subnet mask	255.255.255.0
Gateway	0.0.0.0
Hostname	
Domainname	
(SNTP-Server	0.0.0.0
DNS-Server 1	0.0.0.0

The 'Firmware revision' value '03.01.04 (06)' is highlighted with a red box, indicating that the number in parentheses (06) is the relevant information.

## 9.3 Création du programme principal

Dans le cadre d'une implémentation mono-tâche, le programme principal doit s'appeler **PLC\_PRG**.

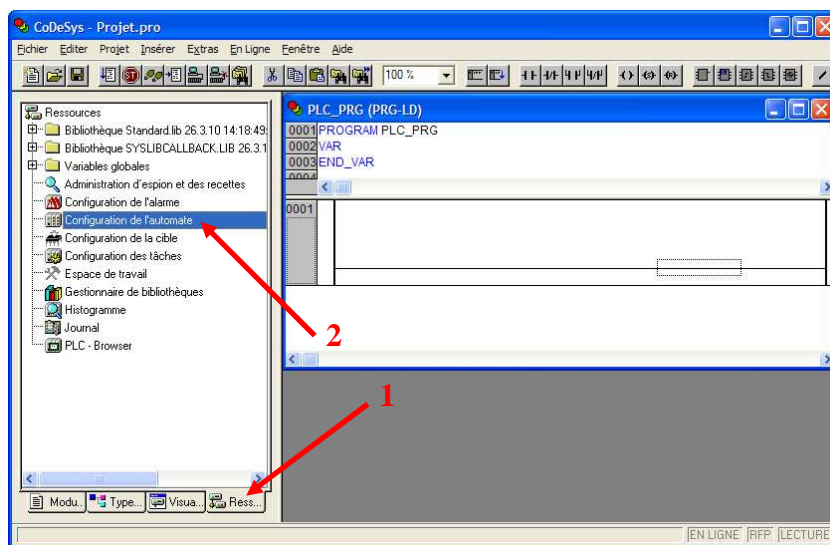
- Sélectionner le langage à contacts en cliquant sur **LD**, puis valider par **OK** :



L'étape suivante est la définition de la configuration matérielle.

## 9.4 Définition de la configuration matérielle et déclaration des variables d'entrées/sorties

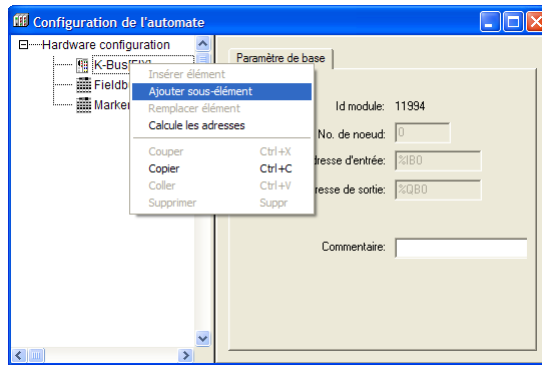
- Cliquer sur l'onglet **Ressources**, puis double-cliquer sur **Configuration de l'automate** :



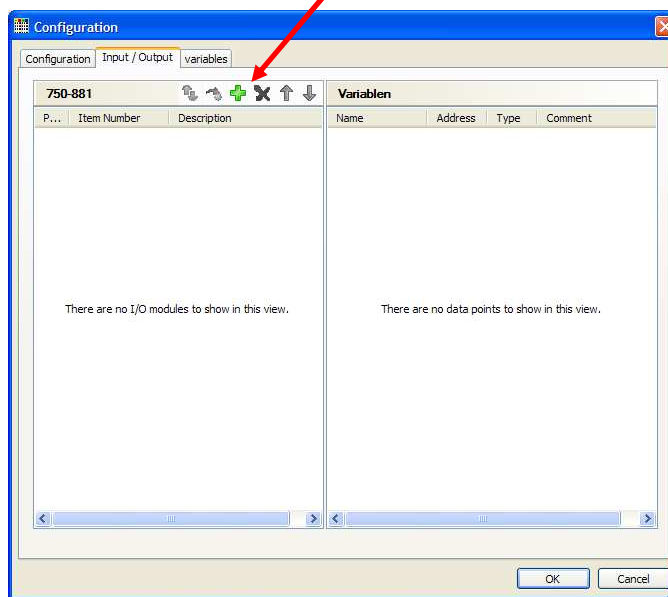
- Développer l'arborescence **Hardware Configuration**



- Faire un clic droit sur **K-Bus[FIX]**, puis sélectionner **Ajouter sous-élément**

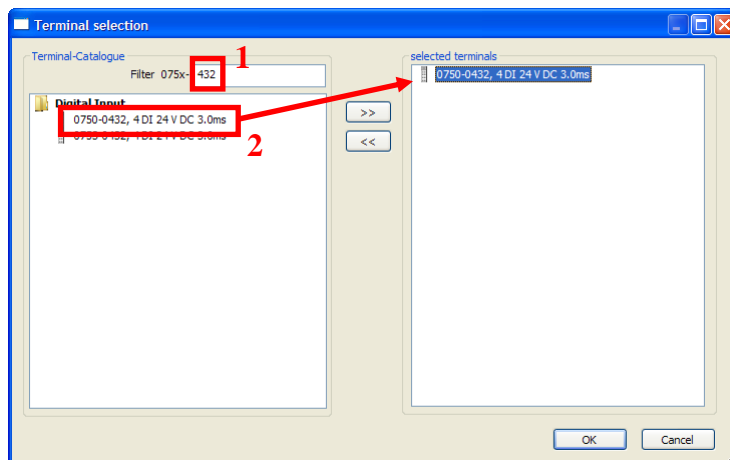


- Cliquer sur le bouton **Add**



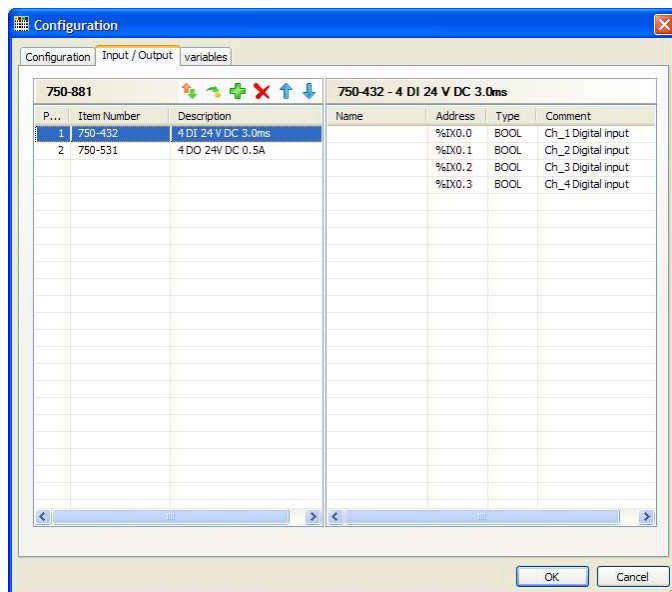
Les bornes d'entrées/sorties sont classées par catégories. Afin de faciliter la recherche, il est possible de saisir un filtre. Dans cet exemple, on dispose d'une borne de 4 entrées 750-432. Il suffit de saisir la fin de la référence dans le champ **Filter** et de double-cliquer sur la borne recherchée.

Les bornes sélectionnées apparaissent ensuite dans la partie droite de la fenêtre :

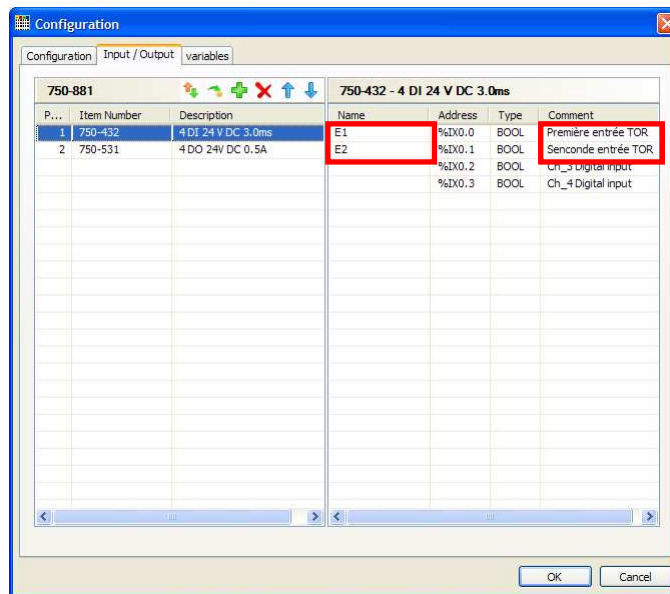


- Répéter l'opération pour l'ensemble des bornes d'entrées/sorties présentes derrière le contrôleur, puis valider par **OK**.

Le configurateur crée automatiquement les variables correspondant aux entrées/sorties. Pour les entrées/sorties tout ou rien, le configurateur crée des variables de type **BOOL**. Pour les entrées analogiques, ce sont en général des variables de type **WORD** (16 bits, non signé) ou **INT** (16 bits, signé).



- Renseigner des noms de variables en regard des deux premières entrées, en les nommant respectivement **E1** et **E2** dans le champ **Name**. Il est également possible d'ajouter un commentaire dans le champ **Comment** :



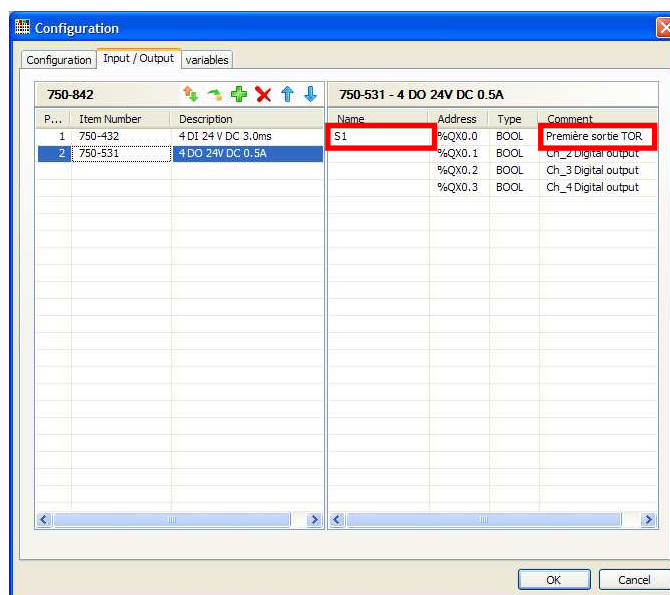
### Remarques

Les noms des variables ne doivent contenir ni accents, ni espaces, ni caractères spécifiques. Le tiret bas est autorisé ( \_ ).

Les noms de variables peuvent contenir indifféremment des majuscules et des minuscules. Le compilateur n'est pas sensible à la casse (une variable déclarée en majuscules peut être utilisée en minuscules, et inversement).

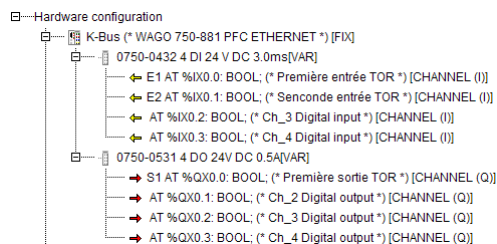
Le nombre de caractères composant le nom d'une variable n'est pas limité.

- Sélectionner ensuite la borne de sortie ou rien dans la partie gauche de la fenêtre, puis nommer la première sortie **S1**.



- Valider ensuite la fenêtre en cliquant sur **OK**.

De retour dans la **Configuration de l'automate**, il est possible de visualiser les variables saisies :



### Remarque

Pour ajouter, modifier ou supprimer des variables et/ou la configuration matérielle, il est nécessaire de faire à nouveau un clic droit sur **K-Bus** et de cliquer sur **Ajouter sous-élément** pour faire apparaître la fenêtre de saisie.



### Remarque

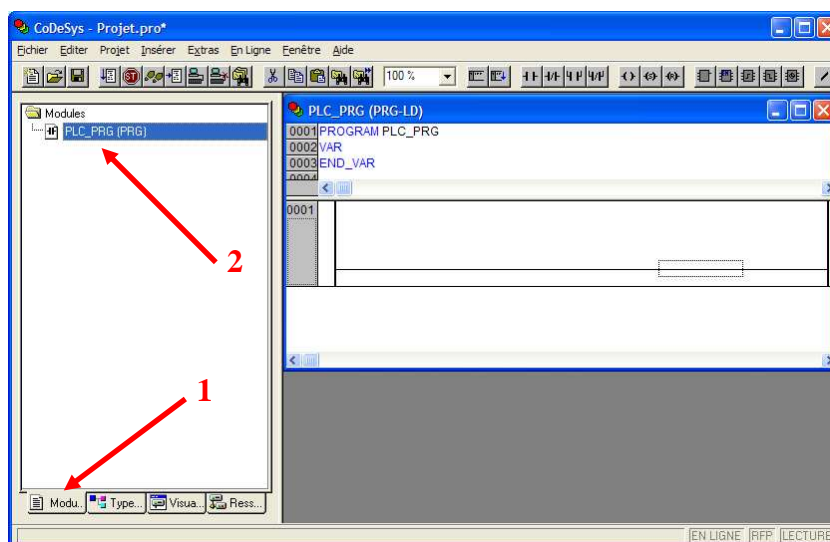
Les variables saisies dans le configurateur matériel sont déclarées comme des variables globales. Elles sont donc accessibles depuis n'importe quel programme, bloc fonctionnel ou fonction du projet.



### Remarque

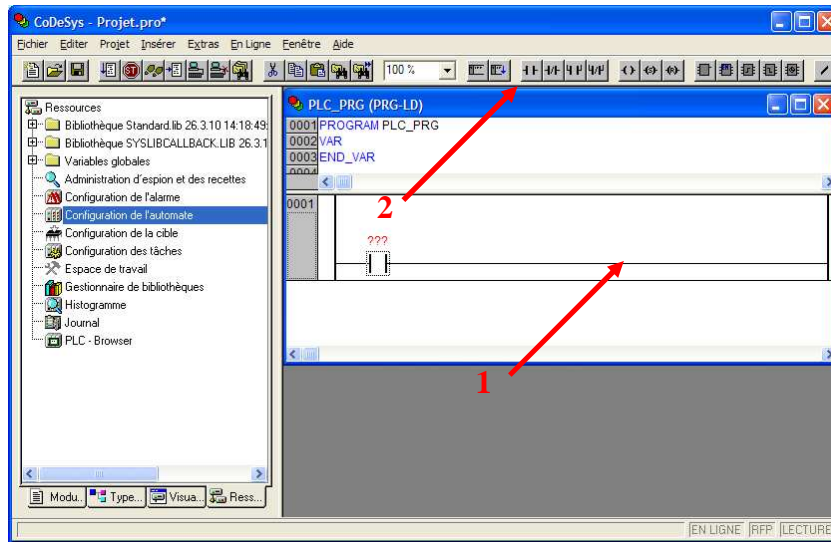
Si le logiciel WAGO-I/O-CHECK est installé sur le PC, il est également possible de faire une lecture automatique de la configuration matérielle. Se reporter au chapitre **Annexe C : lecture de la configuration matérielle depuis WAGO-I/O-CHECK**

- Fermer ensuite la fenêtre de **Configuration de l'automate** et revenir au programme **PLC\_PRG**.



## 9.5 Saisie du programme principal

- Cliquer sur le premier réseau du programme, puis cliquer sur le bouton **Contact** :

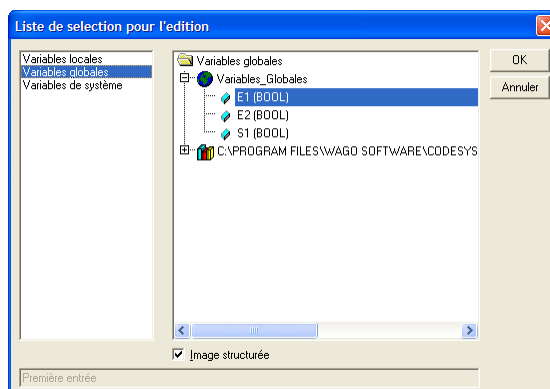


- Cliquer sur les 3 points d'interrogations situés au dessus du contact, puis appuyer sur la touche <F2>.

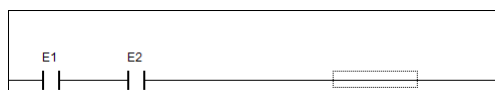


Une fenêtre de sélection apparaît.

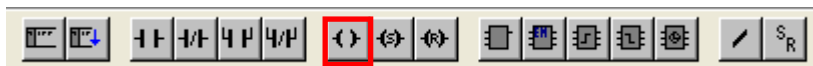
- Cliquer sur **Variables globales**, sélectionner **E1** et valider par **OK** :



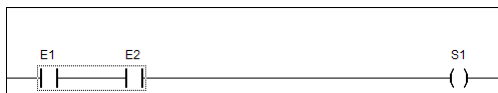
Cliquer à nouveau sur le réseau et réitérer l'opération pour l'entrée **E2**.



- Cliquer à nouveau sur le réseau, puis cliquer sur le bouton **Bobinage** et y associer la variable **S1**.



- Sélectionner ensuite les deux contacts en cliquant successivement sur l'un et sur l'autre, tout en maintenant la touche **<Shift>** enfoncée :

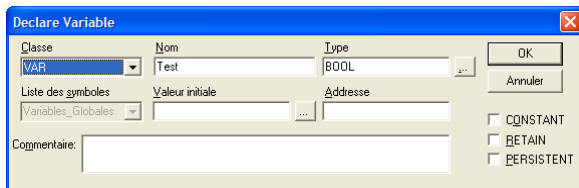


- Une fois les deux contacts sélectionnés, cliquer sur le bouton **Contact parallèle** :



- Cliquer sur les 3 points d'interrogations situés au dessus du contact, puis saisir **Test** et valider par la touche **<Entrée>**.

On vient d'utiliser un nom de variable sans l'avoir déclaré au préalable. L'assistant de déclaration apparaît :



Voici la description des différents champs :

### Classe

VAR	Variable locale
VAR_INPUT	Variable déclarée en tant que paramètre d'entrée du programme, du bloc fonctionnel ou de la fonction (équivalent à un passage par valeur)
VAR_OUTPUT	Variable déclarée en tant que paramètre de sortie du programme ou du bloc fonctionnel
VAR_IN_OUT	Variable déclarée en tant que paramètre d'entrée/sortie du programme, du bloc fonctionnel ou de la fonction (équivalent à un passage par adresse)
VAR_GLOBAL	Variable globale

### Nom

Contient le nom de la variable

### Type

Contient le type de la variable (**BOOL**, **WORD**, etc.)

Il est possible d'accéder à la liste des types disponibles en cliquant sur le bouton suivant :



### Liste des symboles

Permet de sélectionner la liste de variables globales dans laquelle la variable doit être insérée. Il est en effet possible de créer plusieurs listes de variables globales. Celles-ci sont accessibles depuis l'onglet **Ressources**, dossier **Variables globales**.

### Valeur initiale

Permet d'initialiser la variable à une valeur au démarrage du programme (facultatif).

### Adresse

Adresse de la variable. Notamment utile pour les variables à mettre à disposition du réseau.

### Commentaire

Commentaire qui apparaîtra en regard de la variable à l'endroit de sa déclaration.

### CONSTANT

Permet de déclarer une constante au lieu d'une variable. La valeur d'une constante ne peut être modifiée dans le programme. Contrairement aux variables, les constantes sont stockées dans la mémoire programme et non dans la mémoire données.

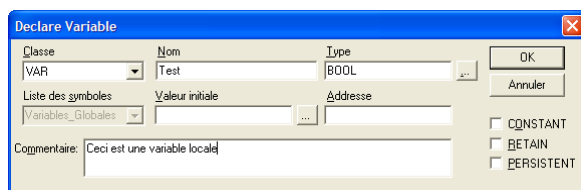
## RETAIN

Permet de définir la variable comme une variable rémanente, c'est-à-dire que son état est conservé même en cas de coupure d'alimentation. Les variables RETAIN sont stockées en NOVRAM (*Non Volatile RAM*). Il n'y a pas de pile dans les contrôleurs WAGO.

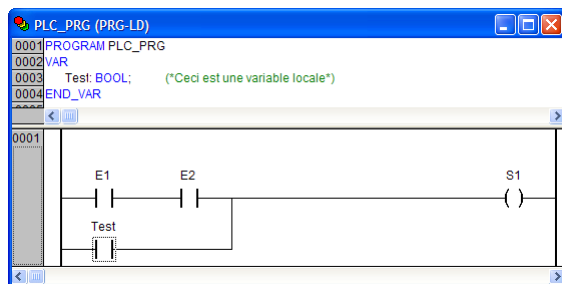
## PERSISTENT

Permet de définir une variable comme une variable persistante, c'est-à-dire que son état est conservé même en cas de mise à jour du programme. Il est possible de combiner les attributs RETAIN et PERSISTENT, afin que l'état d'une variable soit à la fois maintenu en cas de coupure d'alimentation, et à la fois en cas de modification du programme.

- Saisir simplement un commentaire, puis valider par **OK**



La variable est venue s'insérer automatiquement dans les variables locales du programme **PLC\_PRG** :



- Cliquer sur *Projet \ Compiler*, et vérifier dans la fenêtre des messages qu'il n'y a pas d'erreur :



S'il y a des erreurs, appuyer sur la touche **<F4>** pour les faire défiler, et les corriger.

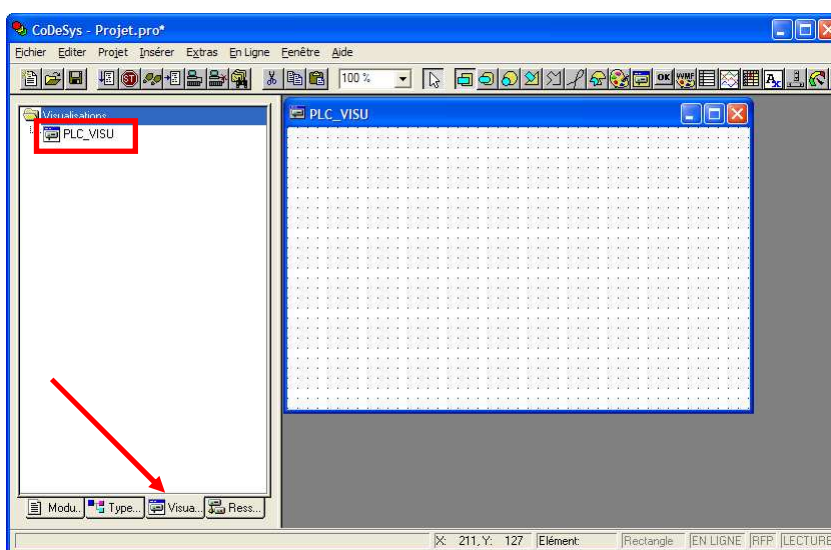


## 9.6 Création d'une fenêtre de visualisation

CoDeSys permet de créer des visualisations graphiques, qui peuvent être exploitées depuis CoDeSys lorsque le logiciel est connecté au contrôleur. Celles-ci permettent d'avoir une vue synthétique de l'installation, et permet également d'agir sur des variables en créant les boutons adéquats.

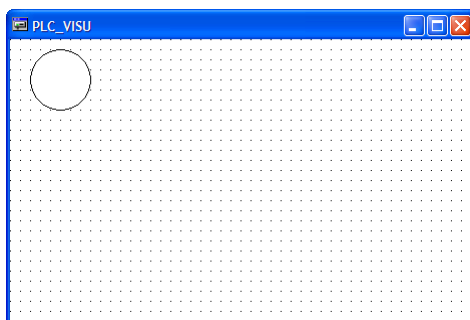
Sur les contrôleurs Ethernet 32 bits, ces visualisations peuvent également être rendues accessibles depuis un simple navigateur web. Cette fonctionnalité s'appelle la **visualisation web**.

- Cliquer sur l'onglet **Visualisations**
- Faire un clic droit dans l'**Organisateur d'Objets** et cliquer sur **Insérer objet**
- Nommer la visualisation **PLC\_VISU** et valider par **OK**



Les étapes suivantes vont permettre de créer les objets graphiques correspondant aux 4 variables du programme.

- Cliquer sur le bouton **Ellipse** et dessiner un rond dans l'espace de travail :



- Double-cliquer sur le rond

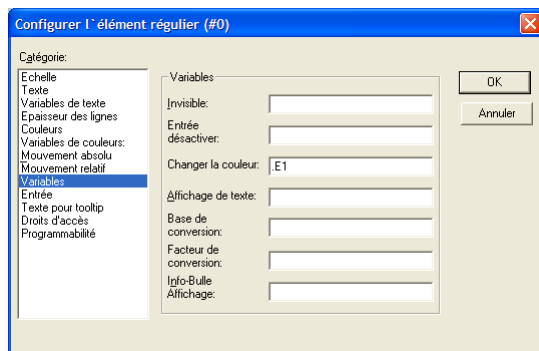
Une fenêtre **Configurer l'élément régulier** apparaît.

- Cliquer sur la catégorie **Texte**
- Saisir **E1** dans le champ **Contenu**
- Cliquer sur la catégorie **Couleurs**

Le groupe **Couleur** définit les couleurs qui seront affichées lorsque l'entrée ne sera pas active (**FALSE**).

Le groupe **Couleur d'alarme** définit les couleurs qui seront affichées lorsque l'entrée sera active (**TRUE**).

- Cliquer sur le bouton **Dedans** du champ **Couleur** et sélectionner une couleur grise
- Cliquer sur le bouton **Dedans** du Champ **Couleur d'alarme** et sélectionner une couleur verte
- Cliquer sur la catégorie **Variables**
- Placer le curseur dans le champ **Changer la couleur** et appuyer sur la touche **<F2>**
- Sélectionner **E1** dans la liste **Variables\_Globales**



- Valider la fenêtre en cliquant sur **OK**

De retour dans la fenêtre de visualisation :

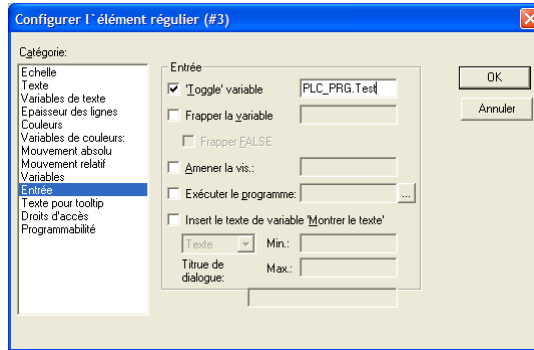
- appuyer sur les touches **<Ctrl>+<C>**, puis 3 fois sur **<Ctrl>+<V>**, afin de dupliquer 3 fois l'objet.
- agencer correctement les cercles sur la fenêtre
- éditer chacun des 3 objets en remplaçant **E1** par **E2**, **S1** et **Test**.

La variable **Test** étant une variable locale au programme **PLC\_PRG**, il faudra la sélectionner dans **PLC\_PRG** et non dans **Variables\_Globales**, au moment de remplir le champ **Changer la couleur**. Le nom complet de la variable apparaîtra alors dans le champ (**PLC\_PRG.Test**).

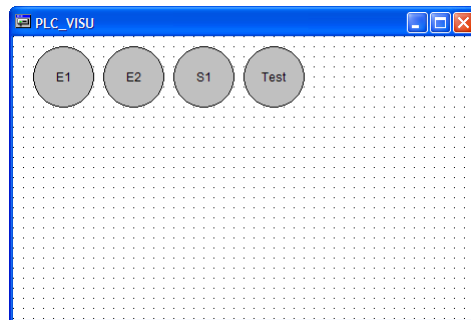
Pour l'objet **Test**, il faut que l'on ait la possibilité de cliquer sur le cercle afin d'inverser la valeur de la variable et ainsi agir sur la sortie.

- double-cliquer sur le rond correspondant à la variable **Test**
- cliquer dans la catégorie **Entrée**
- cocher la case '**Toggle**' variable

- renseigner le champ correspondant en saisissant **PLC\_PRG.Test** (ou en sélectionnant la variable après avoir appuyé sur <F2>).



Une fois l'opération terminée, la fenêtre terminée aura l'aspect suivant :



## 10 Exécution du programme en simulation

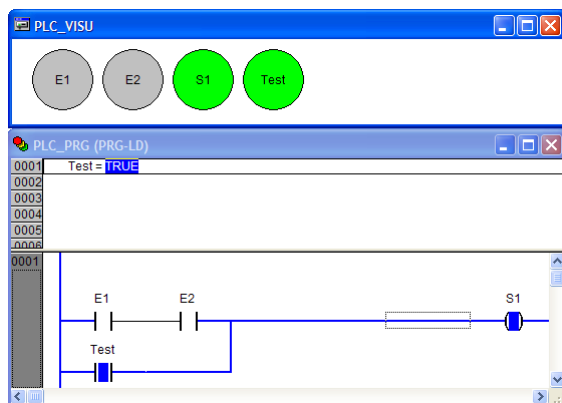
CoDeSys permet de simuler l'exécution d'un programme sans être connecté au contrôleur. La simulation comporte certaines limitations puisqu'elle ne permet pas d'émuler certaines fonctions système, mais elle permet néanmoins l'exécution du reste du code.

- Cliquer sur *En Ligne \ Simulation*
- Cliquer sur *En Ligne \ Accéder au système*
- Cliquer sur *En Ligne \ Démarrer*

Une fois ces commandes exécutées, la barre d'état en bas à droite indique que l'on est en ligne, en simulation et que le programme est en marche :



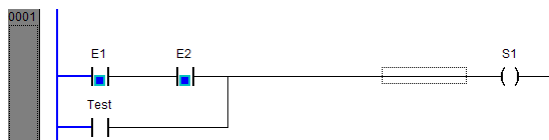
Si l'on clique sur le bouton **Test**, la sortie **S1** doit s'activer :



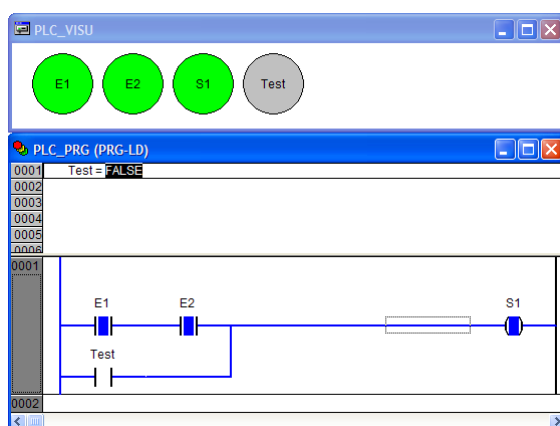
Les cercles correspondant à **E1** et **E2** ne permettent pas de changer l'état des variables, puisque dans l'application finale, elles seront le reflet de l'état des entrées de l'automate, qui sont rafraîchies en permanence.

Pour les besoins de la simulation, il est néanmoins possible de forcer leur état.

- Cliquer à nouveau sur le bouton **Test** pour désactiver la variable
- Double-cliquer sur les contacts **E1** et **E2** dans la fenêtre du programme :



- Valider l'écriture en cliquant sur *En Ligne \ Ecrire valeurs des variables* :



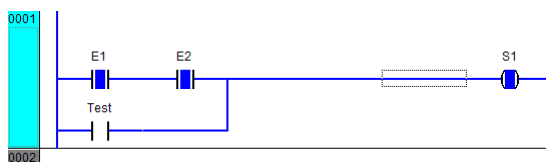
### Remarque

La commande *En Ligne \ Ecrire les valeurs des variables* n'écrit la valeur qu'une seule fois. Si le programme modifie la valeur de la variable, la valeur écrite n'est pas maintenue

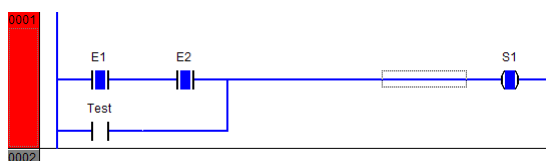
La commande *En Ligne \ Forcer valeurs des variables* force la réécriture de la valeur à chaque début de cycle. Il faut ensuite utiliser la commande *En ligne \ Arrêter de forcer* pour libérer les valeurs des variables.

Il est également possible de positionner des points d'arrêt afin d'exécuter le programme en mode pas à pas. Il suffit de cliquer sur le numéro du réseau.

La couleur bleue indique la présence d'un point d'arrêt :



Lorsque le pointeur programme arrive sur le point d'arrêt, le numéro de réseau passe en rouge :



Ensuite, pour faire avancer le programme d'un pas, il faut cliquer sur *En Ligne \ Etape individuelle sur* ou *En ligne \ Etape individuelle dans*. En cas d'appel à un programme, un bloc fonctionnel ou une fonction, la première commande exécute le module sans entrer dedans. La seconde permet d'entrer dans le module.

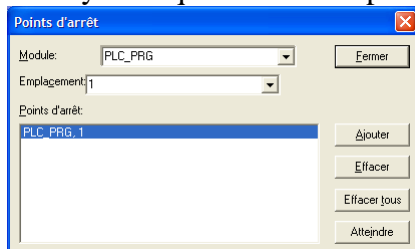
Pour supprimer un point d'arrêt, il suffit de cliquer à nouveau sur le numéro de la ligne.

Pour libérer à nouveau l'exécution du programme, il suffit de cliquer sur *En Ligne \ Démarrer*. Le programme ne s'arrêtera de nouveau que si le pointeur programme arrive sur un point d'arrêt.



### Remarque

La commande *En Ligne \ Dialogue des points d'arrêt* permet d'avoir une vision synthétique de tous les points d'arrêt actifs.

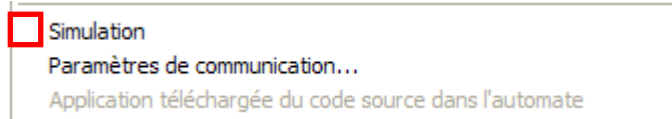


- Avant d'exécuter le programme dans le contrôleur, cliquer sur

*En Ligne \ Quitter le système*

*En Ligne \ Simulation*

Il ne doit plus y avoir de symbole « ✓ » en regard de **Simulation** :

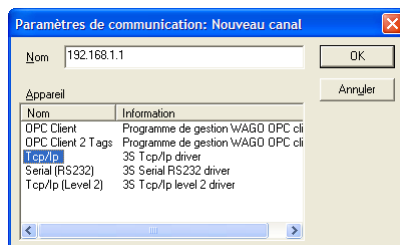


# 11 Téléchargement et sauvegarde dans le contrôleur

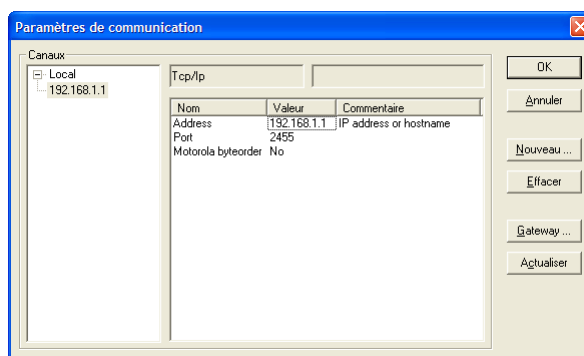
## 11.1 Téléchargement et sauvegarde du code compilé

L'étape suivante consiste à télécharger le programme au travers de la liaison Ethernet.

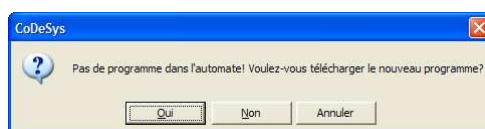
- Cliquer sur *En Ligne \ Paramètres de communication*
- Cliquer sur **Nouveau**
- Dans le champ **Nom**, saisir l'adresse IP du contrôleur. Ceci permettra de distinguer les différents contrôleurs sur des applications plus importantes.
- Sélectionner **Tcp/Ip 3S Tcp/Ip driver**



- Valider par **OK**
- Renseigner ensuite l'adresse IP du contrôleur dans le champ **Address**, et veiller à cliquer ailleurs dans la fenêtre pour sortir du champ et ainsi valider la saisie :



- Après s'être assuré que la saisie était validée, cliquer sur **OK**.
- Cliquer ensuite sur *En ligne \ Accéder au système*
- Cliquer sur **Oui** :

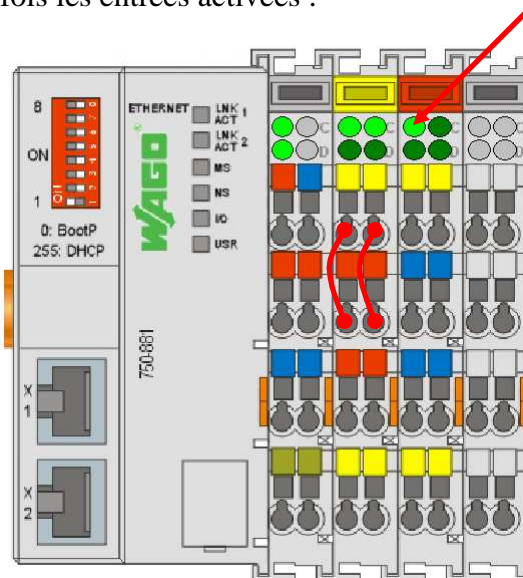


- Cliquer sur *En ligne \ Démarrer*

La barre d'état doit indiquer que l'on est en ligne, que l'on n'est plus en simulation et que le programme est en marche :

EN LIGNE: 192.168.1.1 SIM EN MARCHÉ PA FORCE RFP LECTURE

- Activer les deux entrées en faisant un shunt entre un contact « + » et les contacts des deux entrées.
- Vérifier que la LED correspondant à la première sortie s'allume bien, une fois les entrées activées :



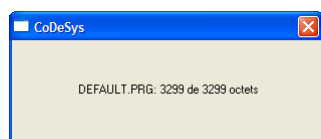
### Attention

A ce stade, le programme est uniquement chargé en RAM. S'il y a une coupure d'alimentation, le programme du contrôleur est perdu !

Afin que le programme redémarre automatiquement à la mise sous tension, deux conditions sont indispensables :

- le programme doit avoir été sauvegardé dans la mémoire flash
- le micro-interrupteur de mode de fonctionnement doit être placé sur la position RUN

Pour sauvegarder le code compilé, exécuter la commande *En Ligne \ Créer projet d'initialisation*.



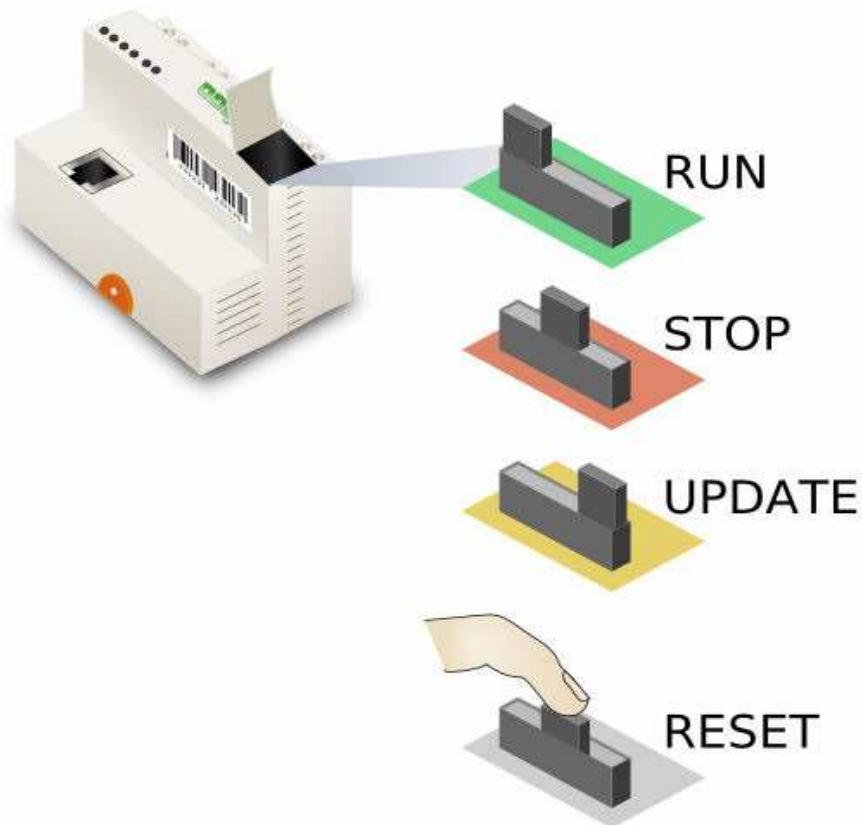




### Remarque

A ce stade, seul le code compilé a été sauvegardé dans le contrôleur. Le code compilé ne peut être décompilé, il est donc impératif de garder le projet source en lieu sûr !

- Ensuite, ouvrir la trappe située en face avant du contrôleur. Le micro-interrupteur de mode de fonctionnement se trouve sur le côté droit du circuit-imprimé.
- Le basculer délicatement vers le haut pour le passer en position **RUN** :



Quelque soit la position du micro-interrupteur (**RUN** / **STOP**), le démarrage et l'arrêt du programme depuis CoDeSys est possible.

A la mise sous tension, si le micro-interrupteur est en position **RUN**, le programme démarre automatiquement.

A la mise sous tension, si le micro-interrupteur est en position **STOP**, le programme ne démarre pas.

Durant le fonctionnement, le fait de passer le micro-interrupteur de la position **RUN** à la position **STOP** arrête le programme.

Durant le fonctionnement, le fait de passer le micro-interrupteur de la position **STOP** à la position **RUN** démarre le programme.

Le micro-interrupteur **ne doit pas** être en position **UPDATE**.

Le fait d'appuyer (avec précaution) sur le micro-interrupteur redémarre le contrôleur. Ceci équivaut à une coupure d'alimentation.



---

**Remarque**

En utilisation normale, le micro-interrupteur doit être en position haute (RUN).

---



---

**Remarque**

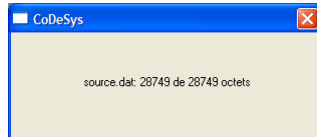
Le contrôleur ne dispose pas de LED de fonctionnement du programme. Néanmoins, il est possible d'animer la LED **USR** (*user*) en face avant du contrôleur, afin d'indiquer l'état du programme. La procédure est décrite au chapitre **Utilisation du modèle de projet**.

---

## 11.2 Téléchargement du code source

Il est possible de télécharger le code source du projet dans le contrôleur, afin de pouvoir le récupérer ultérieurement. L'intégralité du projet source est alors transférée, y compris les commentaires.

- Cliquer sur *En Ligne \ Accéder au système*
- Cliquer sur *En Ligne \ Application téléchargée du code source dans l'automate*



Par défaut, les bibliothèques sont également chargées dans le contrôleur. En cas de problème d'espace insuffisant, il est possible de ne charger que le fichier du projet, sans les bibliothèques.

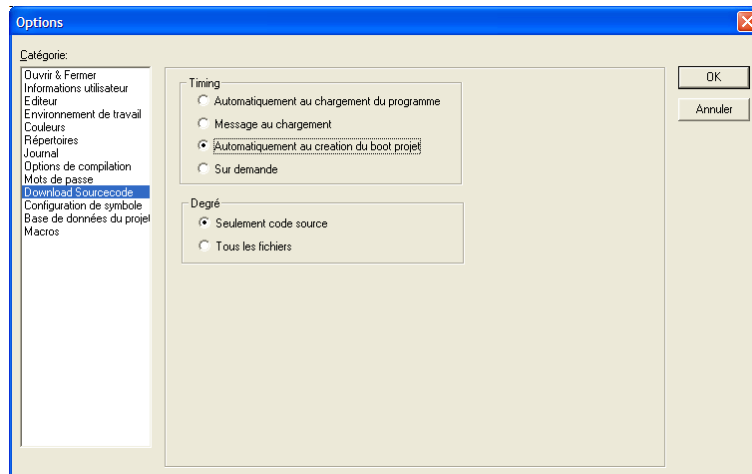
Pour accéder à ce réglage :

- cliquer sur *En Ligne \ Quitter le système*
- cliquer sur *Projet \ Options \ Download Sourcecode*
- Dans le groupe **Degré**, sélectionner **Seulement code source**

Il est également possible de configurer le logiciel afin que le code source se charge automatiquement. Dans le groupe **Timing** :

- **Automatiquement au chargement du programme** : le code source est écrit dès le téléchargement
- **Message au chargement** : un message apparaît au moment du téléchargement
- **Automatiquement au création du boot projet** : le code source est écrit en même temps que l'on sauvegarde le code compilé
- **Sur demande** : le code source n'est écrit que sur exécution de la commande *En Ligne \ Application téléchargée du code source dans l'automate*.

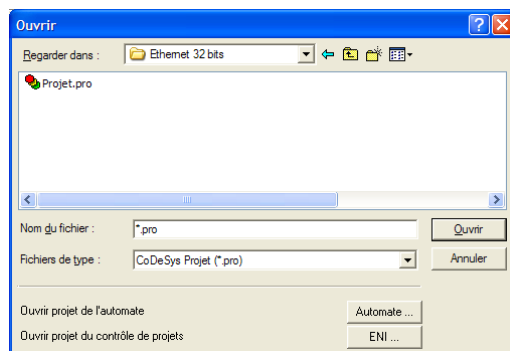
Si le programmeur souhaite charger le code source, WAGO conseille de sélectionner la position **Automatiquement au création du boot projet**, car celle-ci garanti qu'il n'y aura pas d'incohérence entre le code compilé et le code source.



## 11.3 Récupération du code source

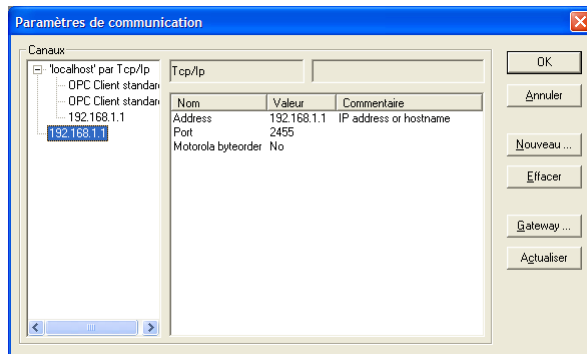
Pour récupérer le code source :

- ouvrir une nouvelle application **CoDeSys V2.3** depuis le menu *Démarrer \ Programmes \ WAGO Software \ CoDeSys for Automation Alliance \ CoDeSys V2.3*
- cliquer sur *Fichier \ Ouvrir*
- cliquer sur le bouton **Automate...**

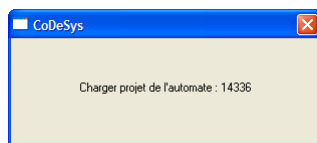


- Sélectionner la cible et valider par **OK**

- Sélectionner le canal de communication précédemment créé, ou en recréer un au besoin



- Valider par **OK**



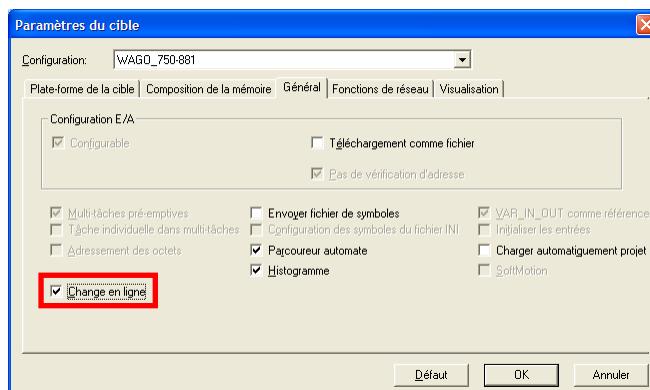
- Sauvegarder ensuite le projet sur le PC

## 11.4 Activation des changements en ligne

Les contrôleurs Ethernet 32 bits supportent les changements en ligne. Les modifications du programme sont réalisées hors ligne, mais le chargement de ces modifications n'interrompt pas l'exécution du programme. D'un cycle automatique à l'autre, le contrôleur passe de l'ancienne à la nouvelle version de programme.

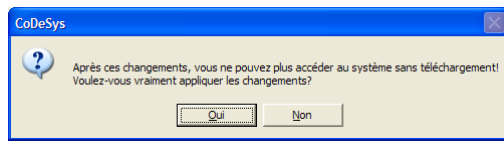
Pour activer les changements en ligne :

- cliquer sur l'onglet **Ressources**
- double-cliquer sur **Configuration de la cible**
- cliquer sur l'onglet **Général**
- cocher la case **Change en ligne**



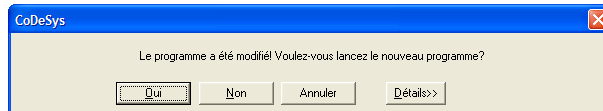
- Valider la fenêtre en cliquant sur **OK**

- Valider le message d'avertissement qui apparaît en cliquant sur **Oui** :

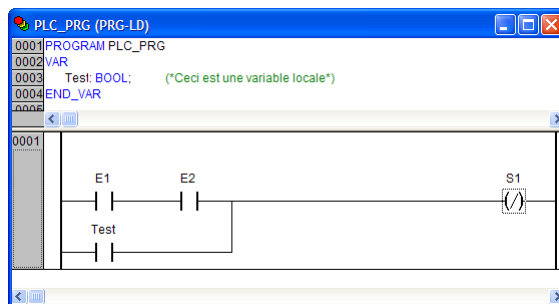


- Cliquer ensuite sur *En Ligne \ Accéder au système*

Le message suivant apparaît :

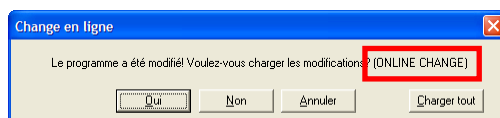


- Valider le message en cliquant sur **Oui**
  - Cliquer sur *En Ligne \ Démarrer*
  - Cliquer sur *En Ligne \ Quitter le système*
- Procéder à une modification du programme, par exemple en complétant la sortie **S1** :



- Cliquer de nouveau sur *En Ligne \ Accéder au système*

Le message de confirmation apparaît désormais avec la mention **ONLINE CHANGE** :



En cliquant sur **Oui**, seules les modifications sont chargées, et le programme n'est à aucun moment arrêté.

En cliquant sur **Charger tout**, le programme est arrêté et l'intégralité de celui-ci est chargée dans le contrôleur.

- Cliquer sur **Oui**

Le programme n'ayant pas été arrêté, la barre d'état indique toujours le mode **EN MARCHÉ** :

EN LIGNE: 192.168.1.1 | SIM | EN MARCHÉ | PA | FORCE | RFP | LECTURE



---

**Remarque**

Certaines modifications majeures, comme l'ajout d'une bibliothèque, nécessitent un rechargement complet du programme, et ne permettent donc pas la mise à jour du programme sans interruption.

---

## 12 Activation de la visualisation web

Les contrôleurs Ethernet 32 bits disposent de la fonctionnalité de **visualisation web**.

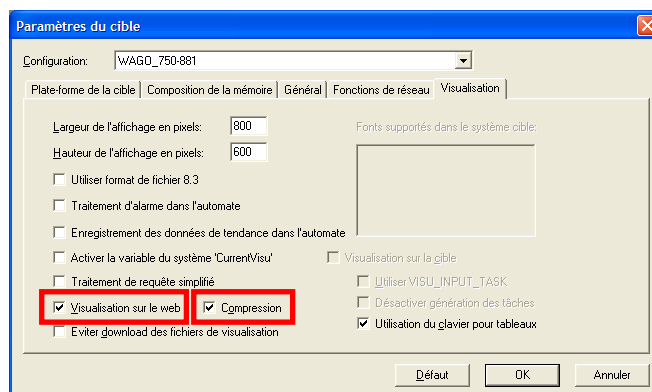
Celle-ci permet d'accéder aux fenêtres de visualisation développées sous CoDeSys à partir d'un simple navigateur web. Ainsi, il est possible d'accéder à des vues d'exploitation ou de paramétrage sans qu'aucun logiciel propriétaire ne soit à installer sur le PC de l'utilisateur final.

Par défaut, la visualisation à laquelle on accède est **PLC\_VISU**. Il est donc nécessaire qu'au moins une visualisation porte ce nom dans le projet. Ensuite, il est bien sûr possible de créer d'autres visualisations et de créer une navigation entre les pages.

La visualisation web de CoDeSys utilise une technologie **Java**. Il est nécessaire que le logiciel **Java** soit installé sur le PC de l'utilisateur final.

Le logiciel **Java** est gratuit, il peut être téléchargé sur le site [www.java.com](http://www.java.com).

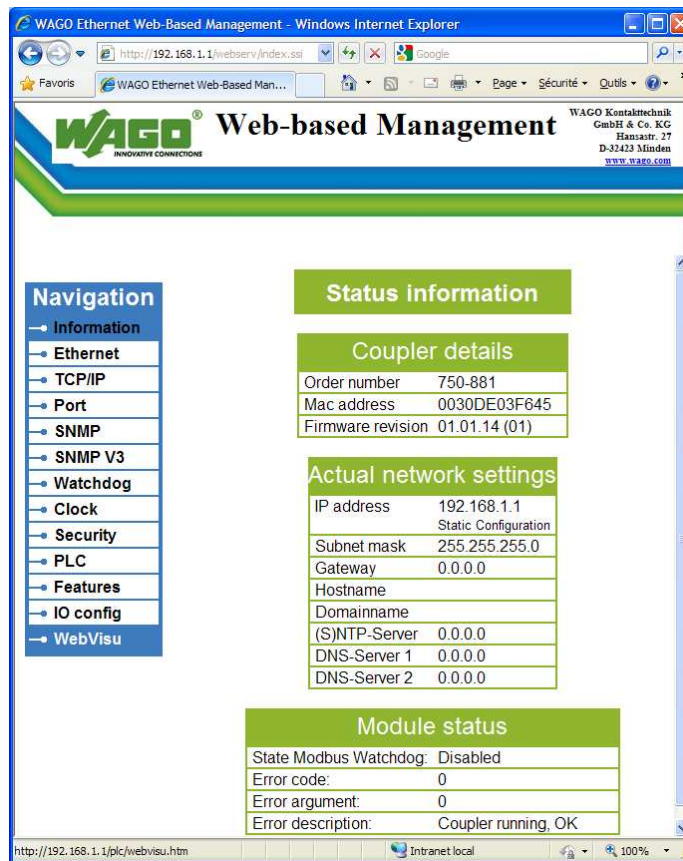
- ouvrir le projet précédemment créé
- cliquer sur l'onglet **Ressources**, puis double-cliquer sur **Configuration de la cible**.
- Cliquer sur l'onglet **Visualisation** et cocher les cases **Visualisation sur le web** et **Compression** :



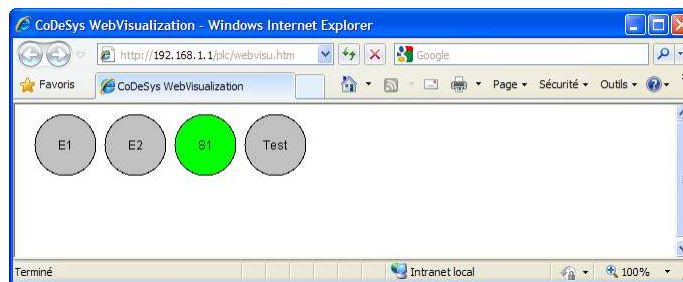
- Valider par **OK**
- Cliquer ensuite sur *En Ligne \ Accéder au système*, pour charger l'ensemble dans le contrôleur
- Cliquer sur *En Ligne \ Démarrer*
- Cliquer sur *En Ligne \ Quitter le système*
- Ouvrir un navigateur web et saisir l'adresse IP du contrôleur dans la barre d'adresses



- Dans le menu de gauche, cliquer sur **WebVisu**



Une nouvelle page s’ouvre, et on retrouve la fenêtre de visualisation telle qu’elle apparaissait sous CoDeSys :



### Informations complémentaires

La visualisation web offre des possibilités d’application très vastes. Pour plus d’information sur cette fonctionnalité, consulter le document :

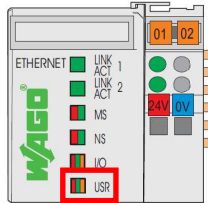
**Mise en service de la visualisation web**

## 13 Utilisation du modèle de projet

Comme évoqué précédemment, les contrôleurs ne disposent pas de LED d'état de fonctionnement du programme. Cette fonctionnalité est très utile en mise en service et en cas de maintenance, car elle permet d'avoir un premier diagnostic visuel sur l'état du contrôleur.

La LED **USR** (*user*) en face avant du contrôleur est librement programmable depuis CoDeSys. WAGO a donc mis au point un modèle de projet pour la gestion de cette LED. Ce modèle permet d'indiquer l'état du programme (RUN/STOP), ainsi que 3 états complémentaires, librement utilisables : intia-  
lisation, défaut mineur, défaut majeur.

Le tableau suivant synthétise les différents états :

	Etat de la LED USR	Signification
	Vert, clignotant	RUN, en cours d'initialisation
	Vert, fixe	RUN, pas de défaut
	Orange clignotant	RUN, défaut mineur
	Orange fixe	RUN, défaut majeur
	Rouge	STOP
	Eteinte	STOP

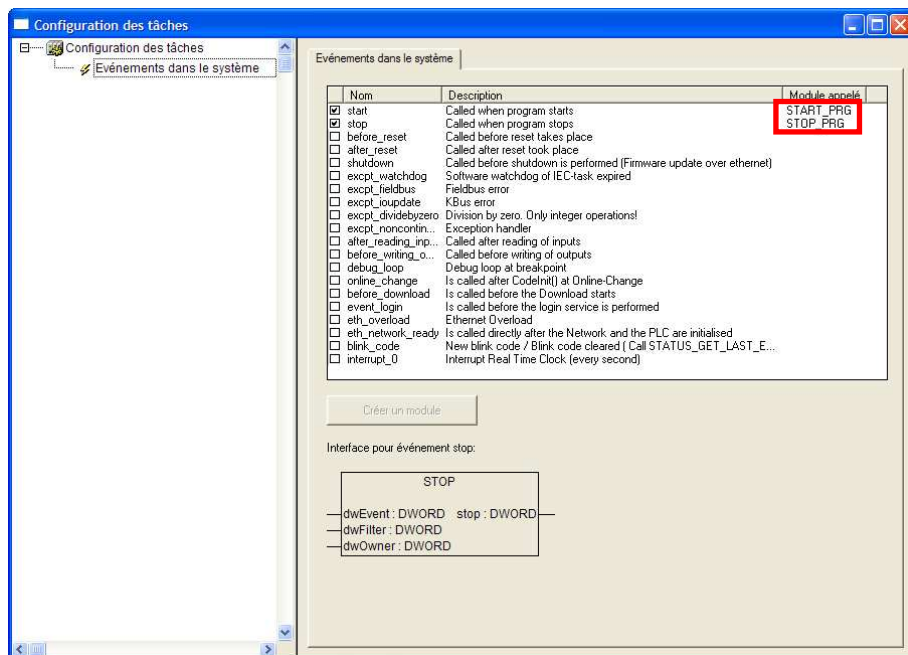
De plus, le modèle de projet contient les appels de fonctions pour la gestion de l'horloge du contrôleur. Les contrôleurs Ethernet 32 bits disposent en effet d'une horloge temps réel (RTC – *Real Time Clock*), qui est secourue pendant une période approximative de 6 jours.

Le modèle de projet permet l'actualisation automatique de la date et de l'heure du contrôleur dans une variable globale (dtActualTime), ainsi que le passage automatique heure d'été / heure d'hiver.

- Enregistrer et fermer le projet, à l'aide des commandes *Fichier \ Enregistrer* et *Fichier \ Fermer*
- cliquer *Fichier \ Nouveau du modèle*
- sélectionner le projet **Template\_Eth\_32bits\_V1.pro**
- cliquer sur *Fichier \ Enregistrer* et sauvegarder le projet sous un autre nom
- cliquer sur l'onglet **Ressources**, puis double-cliquer sur **Configuration de la cible**
- sélectionner la bonne référence de contrôleur, puis valider par **OK**.

Pour animer la LED USR et détecter le démarrage et l'arrêt du programme, le modèle utilise des **événements système**. En cas de changement de cible, les appels de programmes liés à des événements systèmes sont perdus. Il est donc nécessaire de les réactiver :

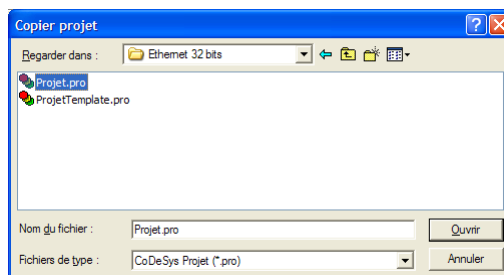
- depuis l'onglet **Ressources**, double-cliquer sur **Configuration des tâches**
- cliquer sur **Événements dans le système**
- cocher les cases correspondant aux événements **start** et **stop**
- cliquer dans la case **Module appelé** de l'événement **start**, appuyer sur <F2> et sélectionner le programme **Template \ USR\_LED \ START\_PRG**
- cliquer dans la case **Module appelé** de l'événement **stop**, appuyer sur <F2> et sélectionner le programme **Template \ USR\_LED \ STOP\_PRG**



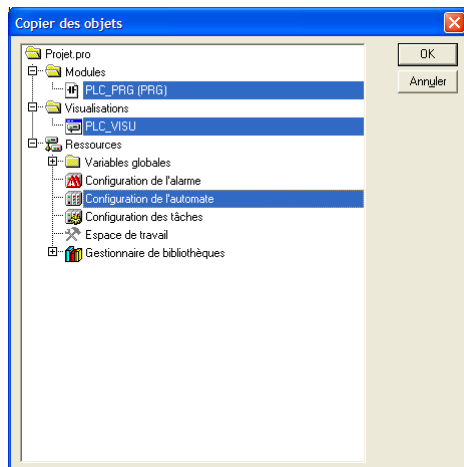
- une fois ces deux opérations réalisées, fermer la fenêtre de **Configuration des tâches**.

Pour récupérer le programme et les visualisations créées dans le projet précédent :

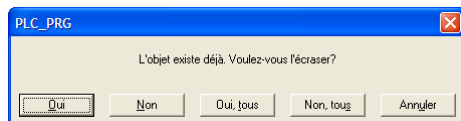
- Cliquer sur **Projet \ Copier**
- Sélectionner le projet précédent, puis cliquer sur **Ouvrir** :



- Sélectionner les éléments à importer en maintenant la touche **<Ctrl>** enfoncée, puis valider par **OK**

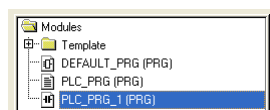


L'objet **PLC\_PRG** étant existant dans le modèle et dans le précédent projet, un message apparaît :

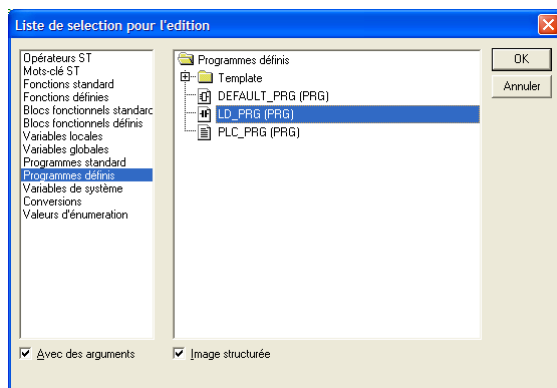


- Cliquer sur **Non**

Le **PLC\_PRG** de l'ancien projet a été renommé en **PLC\_PRG\_1** :



- Faire un clic droit sur **PLC\_PRG\_1**, puis sélectionner **Renommer Objet**
- Le renommer **LD\_PRG**, puis valider par **OK**
- Double cliquer sur **PLC\_PRG**, puis positionner le curseur sous le dernier commentaire
- Appuyer sur **<F2>**
- Sélectionner **Programmes définis**, puis **LD\_PRG** et valider par **OK** :



L'appel au sous programme **LD\_PRG** apparaît maintenant dans le **PLC\_PRG** :



```

0001 PROGRAM PLC_PRG
0002 VAR
0003 END_VAR
0004
0005 (* Attention : en cas de changement de cible (target), valider de nouveau les événements systèmes "start" et "stop" afin qu'ils
0006    fassent respectivement appel aux programmes "START_PRG" et "STOP_PRG" *)
0007
0008 IF xINIT THEN
0009     (* Appeler ici les modules qui ne doivent être exécutés qu'au premier cycle *)
0010     xINIT := FALSE;
0011 END_IF
0012
0013 USR_LED_PRG(); (* Appel au programme d'animation de la LED USR *)
0014 DEFAULT_PRG(); (* Appel au programme d'exemple de positionnement des défauts mineurs et majeurs *)
0015 RTC_PRG(); (* Appel au programme de gestion de l'horloge *)
0016
0017 (* Ajouter ci-dessous les appels de programmes utilisateur -----*)
0018 LD_PRG();
0019
0020
0021
0022

```

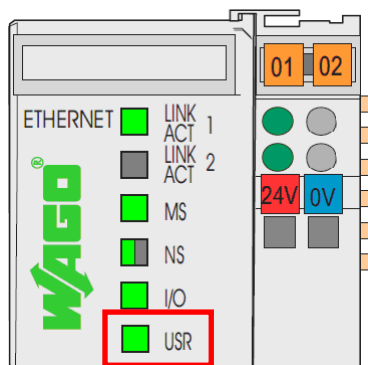


### Remarque

Dans la visualisation **PLC\_VISU**, le bouton **Test** était lié à une variable locale au programme **PLC\_PRG**. Celui-ci ayant été renommé en **LD\_PRG**, il sera nécessaire de reconfigurer ce bouton en changeant **PLC\_PRG.Test** en **LD\_PRG.Test** dans les champs **Variable \ Changer la couleur et Entrée \ Toggle variable**.

- Ensuite, paramétrer de nouveau le canal de communication depuis le menu *En Ligne \ Paramètres de communication*, puis *En Ligne \ Accéder au système*, et enfin *En Ligne \ Démarrer*

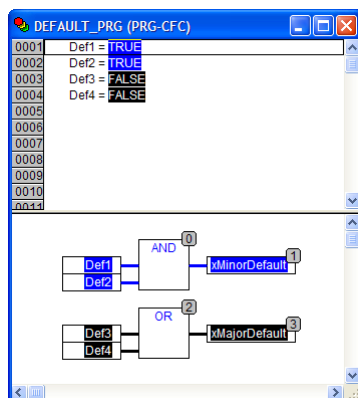
La LED **USR** du contrôleur doit être allumée en vert fixe :



Le programme **DEFAULT\_PRG** est fourni à titre d'exemple pour signaler les défauts mineurs (ex : une entrée analogique en défaut) ou majeurs (ex : perte de communication avec un équipement).

Ce programme sera à adapter à l'application.

- A titre d'exemple, forcer les variables **Def1** et **Def2** à **TRUE** (double clic sur chaque variable, puis <Ctrl> + <F2>) :



La LED **USR** doit maintenant clignoter en orange.

En cas d'arrêt du programme (*En Ligne \ Arrêter le système*), la LED passe en rouge.

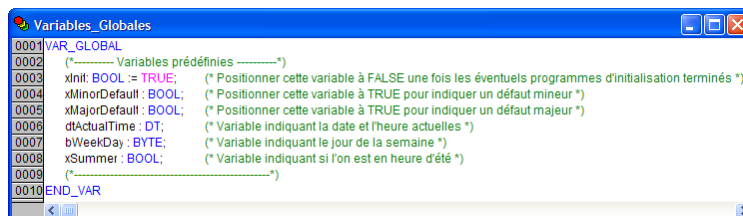
Les variables **xMinorDefault** et **xMajorDefault** sont déclarées en variables globales et sont prévues pour la signalisation des défauts sur la LED **USR**.

La variable **dtActualTime** est une variable de type **DATE\_AND\_TIME**, et renvoie la date et l'heure courante du contrôleur.

La variable **bWeekday** est une variable de type **BYTE** et indique le jour de la semaine :

- 1 > lundi
- 2 > mardi
- 3 > mercredi
- 4 > jeudi
- 5 > vendredi
- 6 > samedi
- 7 > dimanche

La variable **xSummer** est une variable de type **BOOL**. Son état est à **FALSE** en heure d'hiver et à **TRUE** en heure d'été.



### Remarque

WAGO recommande de partir de ce modèle pour de nouveaux projets, plutôt que de partir d'un projet vierge !

## 14 Zones mémoires des contrôleurs

Les contrôleurs disposent de différentes zones mémoire pour les variables.

Toutes les variables non adressées sont stockées dans la mémoire RAM. Le compilateur alloue automatiquement les adresses.

Les données rémanentes non adressées sont stockées dans la mémoire NOVRAM. Le compilateur alloue automatiquement les adresses.

Les données adressées peuvent être situées :

- dans la zone d'entrées physiques ou de sorties physiques
- dans une zone de variables réseau, d'entrée ou de sortie
- dans la zone de mémoire interne

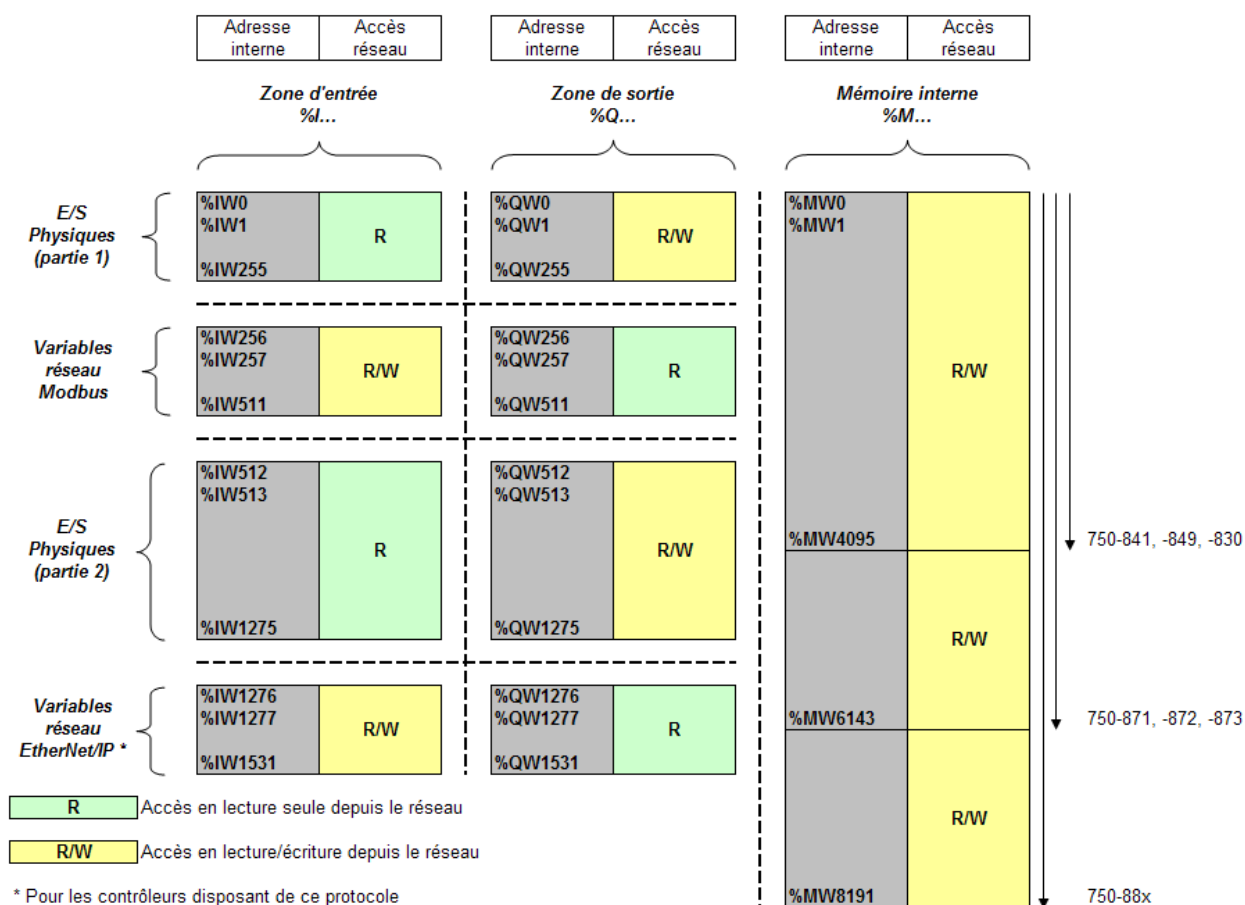
La zone d'entrées et la zone de sorties physiques sont toutes deux décomposées en deux parties : la première est de 256 mots, et la seconde de 764 mots. Dans la majorité des applications, seule la première partie est utilisée.

Il existe 2 zones de variables réseau : l'une pour les protocoles Modbus/TCP et Modbus/UDP, l'autre pour le protocole EtherNet/IP (pour les contrôleurs disposant de ce protocole). Ces zones sont des zones d'échange entre le contrôleur et un système de plus haut niveau (automate, PC, etc.). Chacune de ces zones dispose de 256 mots d'entrée, et de 256 mots de sortie :

- la zone d'entrée (%I...) est accessible lecture seule depuis le contrôleur, et en lecture/écriture depuis le réseau
- la zone de sortie (%Q...) est accessible en lecture/écriture depuis le contrôleur et en lecture seule depuis le réseau

En complément, la zone de mémoire interne peut être utilisée pour étendre le nombre de données échangées en Modbus. En fonction du contrôleur et de la configuration, celle-ci contient de 4096 mots à 12288 mots adressables depuis le réseau Modbus. De plus, cette zone est la seule étant à la fois en lecture/écriture depuis le contrôleur et depuis le réseau Modbus.

Le tableau ci-dessous synthétise les différentes zones disponibles :



La zone de mémoire interne (%M...) peut contenir à la fois des variables volatiles et des variables rémanentes. Pour qu'une variable soit rémanente (adressée ou non), elle doit être déclarée en tant que variable RETAIN.

Les variables RETAIN déclarées sans adresse sont stockées dans la mémoire NOVRAM. Celle-ci est contiguë à la zone de mémoire interne (non représentée sur le schéma précédent).



Il est possible de modifier la répartition entre la zone de mémoire interne et la mémoire NOVRAM. Le réglage se fait dans les **Paramètres de la cible**, accessible depuis l'onglet **Ressources**.

Le tableau ci-dessous représente la configuration par défaut du contrôleur 750-841 :

Le diagramme à gauche illustre la répartition mémoire par défaut : 8 ko de Mémoire interne (%M) et 16 ko de NOVRAM. La capture d'écran de l'interface 'Paramètres du cible' (onglet 'Composition de la mémoire') confirme ces valeurs :

- Code: 16#CD0000
- Globale: 16#4DD0000
- Mémoire: 16#30000000
- Entrée: 16#4CCD810
- Sortie: 16#4CCEC08
- Maintenir: 16#30002000
- Taille: 16#80000
- 16#3D000 par segment
- 16#2000
- 16#BF8
- 16#BF8
- 16#4000
- Nombre max. de blocs fonctionnels: 1023
- Taille du mémoire de données entier: 16#40000
- Nombre max. des segments de données globales: 1

Pour augmenter la taille de la mémoire interne, il faut :

- augmenter la **Taille** du champ **Mémoire**
- augmenter d'autant l'adresse de **Base** sur champ **Maintenir**
- réduire d'autant la **Taille** du champ **Maintenir**

Exemple : si l'on souhaite augmenter la taille de la mémoire interne à 12 ko au lieu de 8, on décale les paramètres précédemment décrits de 4 ko, soit 16#1000. Le paramétrage devient donc le suivant :

Le diagramme à gauche illustre la nouvelle répartition mémoire : 12 ko de Mémoire interne (%M) et 12 ko de NOVRAM. La capture d'écran de l'interface 'Paramètres du cible' (onglet 'Composition de la mémoire') confirme ces nouvelles valeurs :

- Code: 16#CD0000
- Globale: 16#4DD0000
- Mémoire: 16#30000000
- Entrée: 16#4CCD810
- Sortie: 16#4CCEC08
- Maintenir: 16#30003000
- Taille: 16#80000
- 16#3D000 par segment
- 16#3000
- 16#BF8
- 16#BF8
- 16#3000
- Nombre max. de blocs fonctionnels: 1023
- Taille du mémoire de données entier: 16#40000
- Nombre max. des segments de données globales: 1

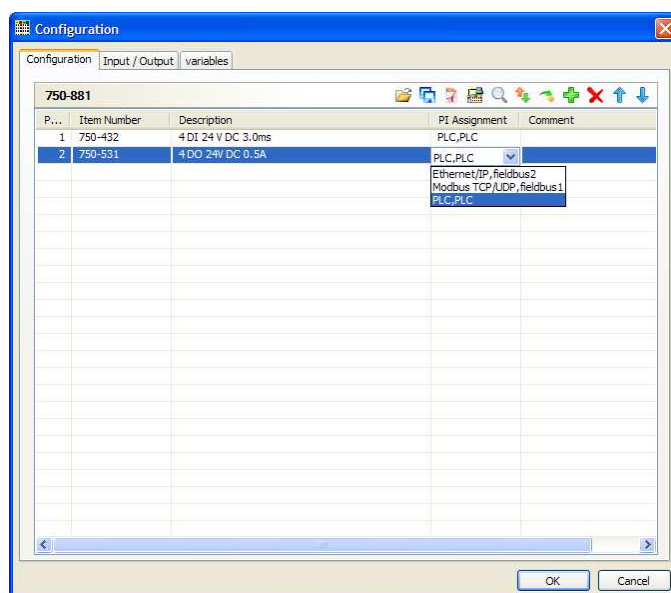
## 15 Affectation des bornes de sortie

L'écriture des sorties est protégée contre les accès concurrents depuis plusieurs sources. Une borne de sortie, ou plus généralement une borne contenant des données dans la table de sortie, est affectée soit au contrôleur, soit aux protocoles Modbus/TCP et Modbus/UDP, soit au protocole de communication supplémentaire du contrôleur (Ethernet/IP, KNX IP, BACnet/IP...).

Par défaut, toutes les bornes de sorties sont affectées au contrôleur. Il est possible de changer l'affectation d'une borne depuis le configurateur matériel.

- A partir de l'onglet **Ressources**, double-cliquer sur **Configuration de l'automate**
- Développer l'arborescence **Hardware Configuration**
- Faire un clic-droit sur **K-Bus**, et sélectionner **Ajouter sous-élément**
- Cliquer sur l'onglet **Configuration**

L'affectation peut être modifiée dans la colonne **PI Assignment** :



Pour les contrôleurs disposant du protocole EtherNet/IP, les affectations suivantes sont possibles (750-881, -880, -841, -871, -872, -873) :

PLC	Contrôleur
fieldbus1	Modbus/TCP ou Modbus/UDP
fieldbus2	EtherNet/IP

Pour le contrôleur KNX IP (750-849) :

PLC	Contrôleur
fieldbus1	Modbus/TCP ou Modbus/UDP

Pour le contrôleur BACnet/IP (750-830) :

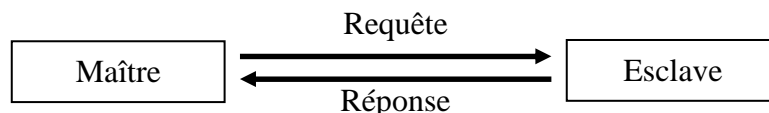
PLC	Contrôleur
Modbus TCP/UDP	Modbus/TCP ou Modbus/UDP
Modbus RTU	Modbus RTU, via liaison série RS232 embarquée
BACnet	BACnet/IP

La société WAGO recommande d'utiliser des variables intermédiaires via une table d'échange située dans la zone d'entrée des variables réseau, puis de réaliser une recopie vers les sorties dans le programme. Cette méthode permet notamment de pouvoir gérer des modes dégradés dans le contrôleur, en cas de perte de communication avec le système de plus haut niveau.

## 16 Modbus

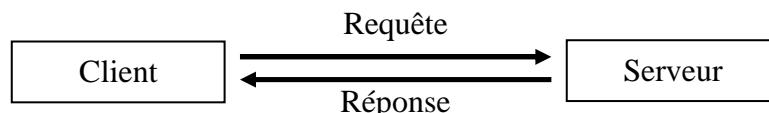
### 16.1 Introduction

Le protocole **Modbus** a été développé en 1979 par la société Modicon. Ce protocole, simple et ouvert, a été conçu pour fonctionner sur des liaisons séries (RS232, RS422, RS485, ou encore boucles de courant). Il fonctionne en mode maître/esclave : le maître envoie une requête à un esclave, puis l'esclave lui renvoie une réponse. Il ne peut y avoir qu'un seul maître sur un même réseau Modbus.



**Jbus** est un protocole utilisant une partie du protocole Modbus. Quelques différences existent entre les deux protocoles, mais le format des trames étant identique, les équipements Modbus et Jbus sont en général interopérables.

Le protocole Modbus a été adapté en 1999 pour fonctionner sur les réseaux IP. Les spécifications du protocole **Modbus/TCP** sont libres d'accès. Le mode client/serveur a remplacé le mode maître/esclave. La principale différence entre un réseau Modbus sur liaison série et un réseau Modbus sur IP est que le réseau peut contenir plusieurs clients. Il est ainsi possible de créer des réseaux « multi-maîtres ».



#### Remarque

Les coupleurs et contrôleurs Ethernet WAGO supportent également le protocole **Modbus/UDP**. Ce protocole offre des performances supérieures, car le nombre d'échanges au niveau de la couche de transport (TCP ou UDP) est moins important.

## 16.2 Codes fonction

Le protocole Modbus définit des codes fonction permettant différents types d'accès aux données d'un esclave.

Voici la liste des codes fonction supportés par les contrôleurs Ethernet 32 bits :

Code fonction	Nom	Description	Accès
FC01	Read coils	Lecture n bits	E/S TOR, variables réseau, mémoire interne
FC02	Read input discretes	Lecture n bits	E/S TOR, variables réseau, mémoire interne
FC03	Read multiple registers	Lecture n mots	E/S analogiques et TOR, variables réseau, mémoire interne
FC04	Read input registers	Lecture n mots	E/S analogiques et TOR, variables réseau, mémoire interne
FC05	Write coil	Ecriture 1 bit	Sorties TOR, variables réseau, mémoire interne
FC06	Write single register	Ecriture 1 mot	Sorties analogiques et TOR, variables réseau, mémoire interne
FC11	Get comm event counters	Lecture compteur événements	Compteur d'événements (nombre d'échanges Modbus corrects)
FC15	Force multiple coils	Ecriture n bits	Sorties TOR, variables réseau, mémoire interne
FC16	Write multiple registers	Ecriture n mots	Sorties analogiques et TOR, variables réseau, mémoire interne
FC22	Mask write register	Ecriture 1 mot avec masque	Sorties analogiques et TOR, variables réseau, mémoire interne
FC23	Read/write register	Lecture/Ecriture mots	E/S analogiques et TOR, variables réseau, mémoire interne



### Remarque

Sur les coupleurs et contrôleurs programmables WAGO, les codes fonction 01 et 02 accèdent aux mêmes données. Idem pour les codes fonction 03 et 04.



### Remarque

Toutes les données (E/S TOR et analogiques, variables réseau, mémoire interne) sont accessibles depuis les codes fonction 03 (lecture mots) et 16 (écriture mots). Une implémentation minimale est donc possible du côté client puisque seuls deux codes fonction sont nécessaires.



### Remarque

Le code fonction 23 permet à la fois de lire et d'écrire des données sur le serveur. L'utilisation de ce code fonction permet une réduction du trafic réseau, ainsi qu'une implémentation minimale du côté client. Une requête avec le code fonction 23 équivaut à une requête d'écriture mots avec le code fonction 16, suivie d'une requête de lecture mots avec le code fonction 03.

## 16.3 Format des adresses Modbus

Une adresse Modbus est constituée d'un préfixe et d'une adresse de registre, augmentée de 1. L'adresse Modbus correspondant à l'accès au registre 0 d'un équipement sera par exemple notée : **40001**.

Le préfixe détermine indirectement le code fonction utilisé :

Préfixe	Nom	Codes fonction lecture	Codes fonction écriture
0	Output coils	FC 01	FC 05 / 15
1	Discrete inputs	FC 02	-
3	Input registers	FC 04	-
4	Holding registers	FC 03 / 23	FC 16 / 23

Le préfixe est suivi d'une adresse. Historiquement, l'adresse est représentée sur 4 digits, en décimal. Sur les contrôleurs WAGO, les adresses pouvant être supérieures à 9999, nous choisirons par convention de représenter les adresses sur 5 digits. Ex : **400001**

Sur les contrôleurs WAGO, les codes fonction 01 et 02 permettant d'accéder aux mêmes zones, les adresses correspondant à des accès **bit** seront notées **0xxxxx**.

De la même manière, les codes fonction 03 et 04 permettant d'accéder aux mêmes zones, les adresses correspondant à des accès **mot** seront notées **4xxxxx**.



### Remarque

Le format des adresses est dépendant du maître utilisé.

Par exemple, l'adresse du registre 256 d'un équipement peut être :

- 256 (décimal, sans préfixe, sans décalage de 1)
- 0x100 (hexadécimal, sans préfixe, sans décalage de 1)
- 40257 (décimal, avec préfixe, avec décalage de 1)
- 400256 (décimal, avec préfixe, avec décalage de 1)

Consulter la documentation du maître Modbus pour déterminer la bonne syntaxe à utiliser.

# 16.4 Correspondance entre les adresses internes et les adresses Modbus

Les tables suivantes synthétisent les correspondances entre les adresses internes et les adresses Modbus.

## 16.4.1 Accès mots

	Adresse interne	Accès	Registre (HEX)	Registre (DEC)	Adresse Modbus	Adresse interne	Accès	Registre (HEX)	Registre (DEC)	Adresse Modbus	Adresse interne	Accès	Registre (HEX)	Registre (DEC)	Adresse Modbus	
<b>Zone d'entrée %I...</b>																
E/S Physiques (partie 1)	%IW0	R	0000	0	400001	%QW0	R/W	0200	512	400513	%MW0	R/W	3000	12288	412289	
	%IW1		0001	1	400002	%QW1		0201	513	400514			%MW1	3001	12289	412290
	%IW255		00FF	255	400256	%QW255		02FF	767	400768						
<b>Zone de sortie %Q...</b>																
Variables réseau Modbus	%IW256	R/W	0300	768	400769	%QW256	R	0100	256	400257	%MW4095	R/W	3FFF	16383	416384	
	%IW257		0301	769	400770	%QW257		0101	257	400258			%MW6143	47FF	18431	418432
	%IW511		03FF	1023	401024	%QW511		01FF	511	400512			%MW8191	4FFF	20479	420480
<b>Mémoire interne %M...</b>																
E/S Physiques (partie 2)	%IW512	R	6000	24576	424577	%QW512	R/W	7000	28672	428673	Pas d'accès bit					
	%IW513		6001	24577	424578	%QW513		7001	28673	428674						
	%IW1275		62FC	25340	425341	%QW1275		72FC	29436	429437						

R Accès en lecture seule depuis le réseau (codes fonction 03 et 04)

R/W Accès en lecture/écriture depuis le réseau (codes fonction 03, 04, 06, 16, 23)

## 16.4.2 Accès bits

	Adresse interne	Accès	Adresse du bit (HEX)	Adresse du bit (DEC)	Adresse Modbus	Adresse interne	Accès	Adresse du bit (HEX)	Adresse du bit (DEC)	Adresse Modbus	Adresse interne	Accès	Adresse du bit (HEX)	Adresse du bit (DEC)	Adresse Modbus	
<b>Zone d'entrée %I...</b>																
E/S Physiques (partie 1)	1ère entrée TOR	R	0000	0	000001	1ère sortie TOR	R/W	0200	512	000513	%MX0.0	R/W	3000	12288	012289	
	2ème entrée TOR		0001	1	000002	2ème sortie TOR		0201	513	000514			%MX0.1	3001	12289	012290
	512ème entrée TOR		01FF	511	000512	512ème sortie TOR		02FF	767	000768						
<b>Zone de sortie %Q...</b>																
Variables réseau Modbus	%IX256.0	R/W	2000	8192	008193	%QX256.0	R	1000	4096	004097	%MX1279.15	R/W	7FFF	32767	032768	
	%IX256.1		2001	8193	008194	%QX256.1		1001	4097	004098			%MX1280.0			
	%IX511.15		2FFF	12287	012288	%QX511.15		1FFF	8191	008192						
<b>Mémoire interne %M...</b>																
E/S Physiques (partie 2)	513ème entrée TOR	R	8000	32768	032769	513ème sortie TOR	R/W	9000	36864	036865	Pas d'accès bit					
	514ème entrée TOR		8001	32769	032770	514ème sortie TOR		9001	36865	036866						
	2040ème entrée TOR		85F7	34295	034296	2040ème sortie TOR		95F7	38391	038392						

R Accès en lecture seule depuis le réseau (codes fonction 01 et 02)

R/W Accès en lecture/écriture depuis le réseau (codes fonction 01, 02, 05, 15)

## 16.5 Organisation des tables des entrées/sorties physiques

Les contrôleurs créent automatiquement les tables d'entrée et de sortie physiques, en fonction des bornes d'entrées/sorties installées. Cette opération est réalisée automatiquement à chaque mise sous tension.

Il y a une table pour les entrées et une table pour les sorties.

Le contrôleur commence par placer dans la table d'entrée les données correspondant aux entrées analogiques et aux bornes spécifiques (comptage rapide, interfaces séries, etc.). Ces données sont placées dans l'ordre des bornes, en partant du contrôleur, et en allant vers la borne finale. Viennent ensuite les bits correspondant aux entrées TOR. Ces bits sont concaténés dans le mot suivant le dernier mot occupé par les bornes d'entrées analogiques et les bornes spécifiques. Si le nombre de bits occupés par les entrées TOR dépasse 16 bits, un nouveau mot est automatiquement commencé.

Le principe est identique pour les sorties.

En résumé, les tables d'E/S sont constituées de la manière suivante :

Table d'entrée :

- mots occupés par les entrées analogiques et bornes spécifiques
- bits occupés par les entrées TOR (concaténation)

Table de sortie :

- mots occupés par les sorties analogiques et bornes spécifiques
- bits occupés par les sorties TOR (concaténation)

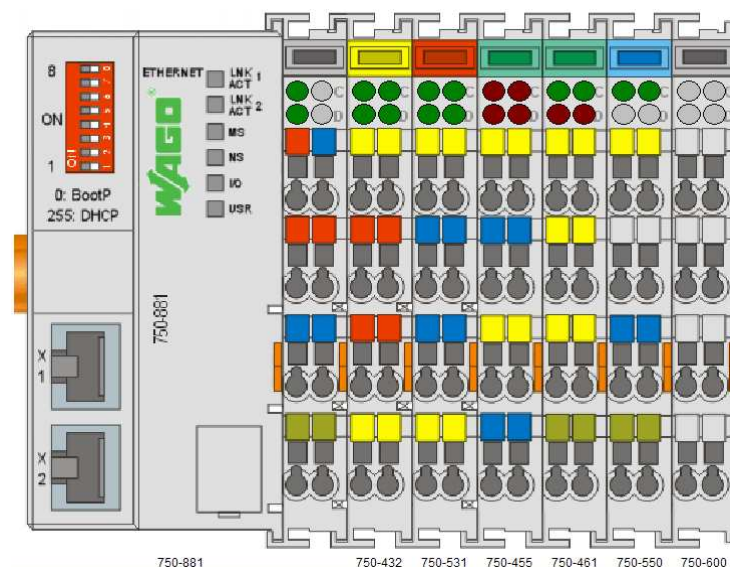
Des exemples sont donnés ci-après.



## 16.5.1 Exemple 1

Le bornier suivant est constitué de :

- 4 entrées TOR (750-432)
- 4 sorties TOR (750-531)
- 4 entrées analogiques 4-20mA (750-455)
- 2 entrées analogiques Pt100 (750-461)
- 2 sorties analogiques 0-10V (750-550)



Dans la table d'entrée, le contrôleur placera d'abord les 4 mots occupés par les entrées 4-20mA, puis les deux mots occupés par les entrées Pt100, et ajoutera dans le mot suivant les 4 bits occupés par les entrées TOR.

Dans la table de sortie, le contrôleur placera d'abord les 2 mots occupés par les sorties 0-10V, puis ajoutera dans le mot suivant les 4 bits occupés par les sorties TOR.

Adr. Interne	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Adr. Registre	Adr. Modbus
%IW0	750-455, entrée 1																0	400001
%IW1	750-455, entrée 2																1	400002
%IW2	750-455, entrée 3																2	400003
%IW3	750-455, entrée 4																3	400004
%IW4	750-461, entrée 1																4	400005
%IW5	750-461, entrée 2																5	400006
%IW6	750-432																6	400007

Adr. Interne	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Adr. Registre	Adr. Modbus
%QW0	750-550, sortie 1																512	400513
%QW1	750-550, sortie 2																513	400514
%QW2	750-531																514	400515

## 16.5.2 Exemple 2

Dans ce second exemple, on a ajouté une borne d'interface série configurable 750-653/003-000, ainsi qu'une borne de 16 sorties TOR 750-1500.



La borne d'interface série est configurée en 5 octets, et occupe donc 3 mots d'entrée (5 octets de données + un octet d'état) et 3 mots de sortie (5 octets de données + un octet de contrôle). Cette interface série étant placée après les entrées analogiques, ses 3 mots d'entrée seront placés après ceux des entrées analogiques, et avant les bits des entrées TOR.

Les bits de la borne 16 sorties 750-1500 sont placés à la suite de ceux de la borne 4 sorties 750-531. Comme le nombre de bits restants dans le mot %QW5 n'étant pas suffisant, le mot %QW6 est automatiquement commencé afin d'accueillir les 4 bits restants.

**Table d'entrée**

Adr. Interne	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Adr. Registre	Adr. Modbus
%IW0	750-455, entrée 1																0	400001
%IW1	750-455, entrée 2																1	400002
%IW2	750-455, entrée 3																2	400003
%IW3	750-455, entrée 4																3	400004
%IW4	750-461, entrée 1																4	400005
%IW5	750-461, entrée 2																5	400006
%IW6	750-653/003-000 configurée en 5 octets																6	400007
%IW7																	7	400008
%IW8																	8	400009
%IW9																	9	400010

750-432

**Table de sortie**

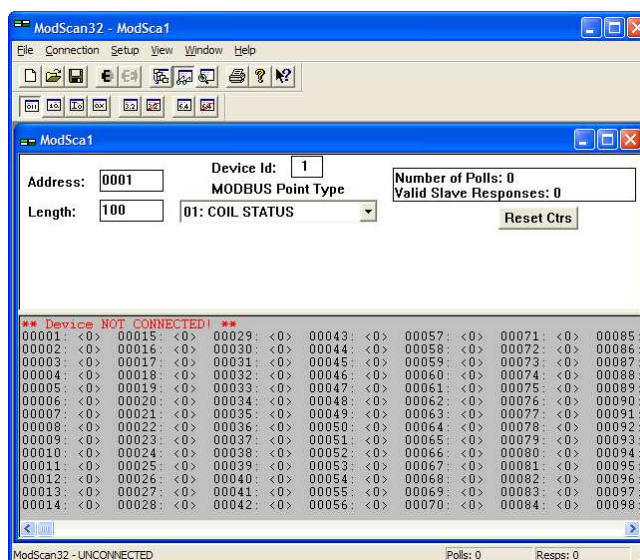
Adr. Interne	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Adr. Registre	Adr. Modbus
%QW0	750-550, sortie 1																512	400513
%QW1	750-550, sortie 2																513	400514
%QW2	750-653/003-000 configurée en 5 octets																514	400515
%QW3																	515	400516
%QW4																	516	400517
%QW5																	517	400518
%QW6	750-1500, sorties 1 à 12										750-1500, sorties 13 à 16				518	400519		

## 16.6 Test de la communication Modbus

Une version d'essai de l'application **ModScan32** est téléchargeable sur le site [www.win-tech.com](http://www.win-tech.com). Ce logiciel permet d'émettre des trames Modbus/TCP.

- Lancer l'application **ModScan32**
- Valider les deux écrans de démarrage

L'écran suivant apparaît :



Voici une brève description des différents champs :

Champ	Description
<b>Address</b>	Adresse du premier mot ou bit à lire
<b>Length</b>	Nombre de mots ou de bits à lire
<b>Device Id</b>	Numéro d'esclave Modbus (uniquement lors de communications avec une passerelle Modbus/TCP vers Modbus)
<b>MODBUS Point Type</b>	Type de données à lire : 01: Coil Status           ⇒ Lecture bits 02: Input Status         ⇒ Lecture bits 03: Holding Register     ⇒ Lecture mots 04: Input Register       ⇒ Lecture mots
<b>Number of Polls</b>	Nombre de requêtes émises du PC vers le serveur Modbus/TCP
<b>Valid Slave Responses</b>	Nombre de réponses correctes du serveur Modbus/TCP
<b>Reset Ctrs</b>	Remet les compteurs de requêtes et de réponses à 0



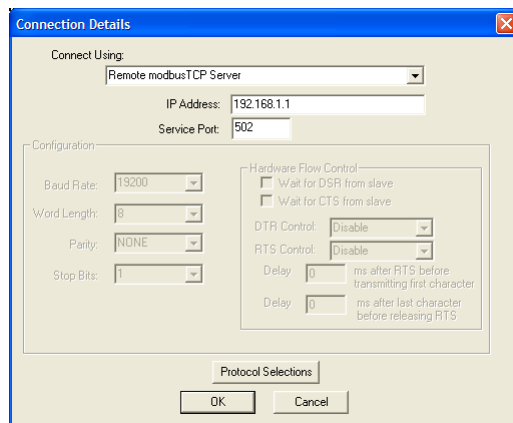
### Remarques

Sous **Modscan 32**, il faut ajouter 1 aux adresses de registres précédemment décrites, mais sans saisir le préfixe, qui est ajouté automatiquement en fonction de la valeur du champ **MODBUS Point Type**.

Par exemple, pour accéder en lecture mots au premier registre d'entrée, il faudra sélectionner **03: Holding Register** dans le champ **MODBUS Point Type**, et saisir **0001** dans le champ **Address**.

- Pour établir la communication, entrer dans le menu *Connection \ Connect*.

L'écran suivant apparaît :



- Dans la liste **Connect Using**, choisir **Remote TCP/IP Server**
- Dans la section **IP Address**, saisir l'adresse IP affectée au contrôleur
- Saisir **502** dans le champ **Service Port**
- Valider cette fenêtre en cliquant sur le bouton **OK**.

Le dialogue commence alors entre le PC et le contrôleur. On peut ensuite changer l'adresse de début, ainsi que le nombre d'entrées / sorties à visualiser.

Il est possible de changer le type de fonction utilisé pour lire soit les mots, soit les bits, en modifiant la valeur sélectionnée dans le champ **MODBUS Point Type**.

Des écritures bit et des écritures mots peuvent être déclenchées respectivement par les menus *Setup \ Extended \ Force coils* et *Setup \ Extended \ Preset Regs*.



### Remarque

Pour écrire directement sur une sortie, TOR ou analogique, il est nécessaire d'avoir au préalable affecté la borne correspondante au fiibus1, dans le configurateur matériel sous CoDeSys.

Se référer au besoin au chapitre **Affectation des bornes de sortie**.

Il est également possible de choisir le temps entre l'émission de 2 trames dans le menu *Setup \ Data Definition*. Cette durée est exprimée par défaut en millisecondes dans la section **Scan Rate**, et doit être comprise entre 5 et 10 000.

## 17 Autres protocoles de communication

Les contrôleurs Ethernet 32 bits disposent d'un ou plusieurs protocoles de communication supplémentaires, en plus des protocoles Modbus/TCP et Modbus/UDP. Ces protocoles peuvent fonctionner en même temps que les connexions Modbus et au travers des mêmes ports Ethernet.

La disponibilité des protocoles dépend de la référence du contrôleur.

### 17.1 EtherNet/IP

Section en préparation.

### 17.2 KNX IP

Les **objets de communication** KNX sont liés aux variables du programme CoDeSys par l'utilisation de blocs fonctionnels disponibles dans la bibliothèque **KNX\_Standard**.

Les **objets de communication** sont liés aux **adresses de groupe** dans le logiciel de configuration standard **ETS**. WAGO fournit un *plug-in* permettant la configuration du contrôleur KNX IP sous **ETS**.

Une procédure de mise en service complète est décrite dans le document :

#### Mise en service des équipements KNX

### 17.3 BACnet/IP

Les objets BACnet peuvent être liés à des variables simples ou à des structures dans lesquelles on retrouve l'ensemble des propriétés des objets BACnet. Ces structures et des fonctions de conversion sont disponibles dans la bibliothèque **BACnet\_01\_easy.lib**.

WAGO fournit le logiciel **WAGO BACnet Configurator** afin de réaliser la liaison entre les variables internes et les objets BACnet, ainsi que pour lier les objets BACnet du contrôleur à ceux d'autres équipements.

Une procédure de mise en service complète est décrite dans le document :

#### Mise en service des contrôleurs BACnet

## 17.4 CEI 60870-5

Les objets de communication **CEI-60870-5-101** (série) et **CEI 60870-5-104** (TCP/IP) peuvent être créés directement depuis un configurateur sous CoDe-Sys.

Une procédure de mise en service complète est décrite dans le document de démarrage rapide :

**Quick-Start Manual for ETHERNET Fieldbus Controller 750-872**

## 17.5 CEI 61850

Les objets de communication **CEI 61850** peuvent être créés directement depuis un configurateur sous CoDeSys.

Une procédure de mise en service complète est décrite dans le document de démarrage rapide :

**IEC 61850 Solution for programmable Controls of Remote Control Technology 750-872, 758-870/000-130, /000-131, -875/000-130, /000-131**

## 18 Utilisation des bibliothèques

De nombreuses bibliothèques sont disponibles dans le package logiciel fourni par WAGO. Celles-ci contiennent de nombreux programmes, blocs fonctionnels et fonctions utiles pour le développement d'applications d'automatisme.

### 18.1 Dossiers de stockage des bibliothèques

Les bibliothèques génériques, pouvant être utilisées sur plusieurs plateformes, sont disponibles dans le dossier suivant :

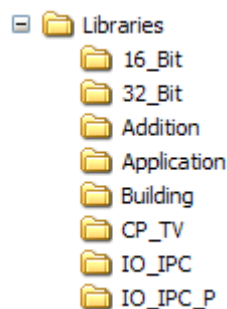
*C:\Program Files\WAGO Software\CoDeSys V2.3\Library*

Ce dossier contient notamment la bibliothèque **Util.lib**, qui intègre des fonctions intéressantes comme le bloc **BLINK**, permettant de générer un signal binaire périodique, dont on précise la période à l'état haut et la période à l'état bas.

Les bibliothèques spécifiques aux cibles WAGO sont situées dans le dossier suivant :

*C:\Program Files\WAGO Software\CoDeSys V2.3\Targets\WAGO\Libraries*

Ce dossier contient plusieurs sous-dossiers :



Voici l'organisation des différents sous-dossiers :

Dossier	Description
16_Bit	Bibliothèques pour les contrôleurs 16 bits (ex : 750-842, -843, etc.).
32_Bit	Bibliothèques pour les contrôleurs 32 bits (ex : 750-881, -841, etc.).
Addition	Bibliothèques ajoutées manuellement, non installées avec le package logiciel de base. Ce dossier est à créer manuellement.
Application	Bibliothèques communes aux différents contrôleurs, générales, surtout orientées pour la communication, via des bornes d'interface série ou le port Ethernet
Building	Bibliothèques communes aux différents contrôleurs pour l'automatisation du bâtiment
CP_TV	Bibliothèques pour les <i>Control Panel</i> de la gamme PERSPECTO®
IO_IPC	Bibliothèques pour les PC industriels de première génération WAGO-I/O-IPC (ex : 750-870, -/000-001, etc.).
IO_IPC_P	Bibliothèques pour les PC industriels de seconde génération WAGO-I/O-IPC (ex : 750-874/000-110, etc.).

Les bibliothèques pour l'automatisation du bâtiment peuvent être téléchargées sur [www.wago.fr](http://www.wago.fr), et sont également présentes sur le DVD **AUTOMATION**

#### **Tools and Docs :**

> Services > Téléchargements > Automatisation du bâtiment  
 > Bibliothèque WAGO-I/O-PRO > WAGO I/O PRO CAA libraries  
 for building automation

- Télécharger le fichier **Libraries\_BA.zip**, et extraire son contenu dans le dossier :

*C:\Program Files \ WAGO Software \ CoDeSys V2.3 \ Targets \ WAGO \ Libraries \ Building*

## 18.2 Insertion de bibliothèques dans le projet

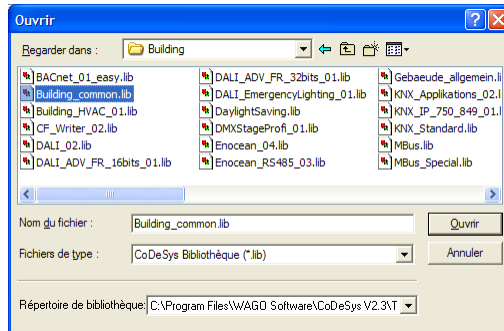
Pour insérer une nouvelle bibliothèque dans le projet :

- cliquer sur *Fenêtre \ Gestion des bibliothèques*
- cliquer sur *Insérer \ Autre bibliothèque*

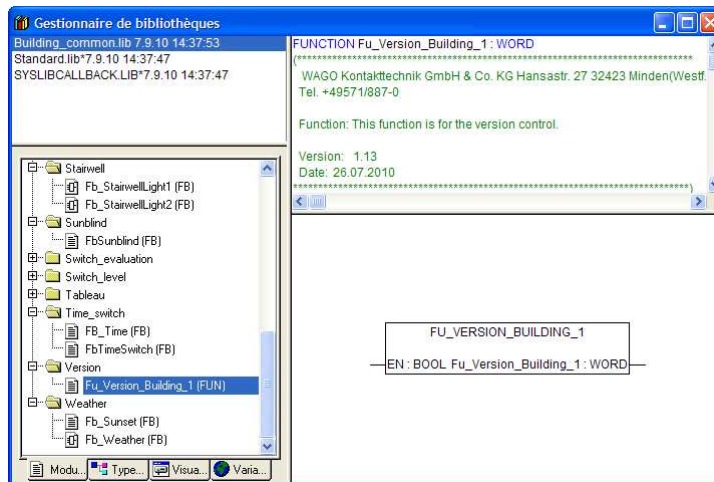
Par défaut, le dossier dans lequel on est situé est le dossier **32\_bit**.

- remonter d'un dossier dans l'arborescence
- se rendre dans le dossier **Building**
- sélectionner la bibliothèque **Building Common**, puis cliquer sur **Ouvrir**

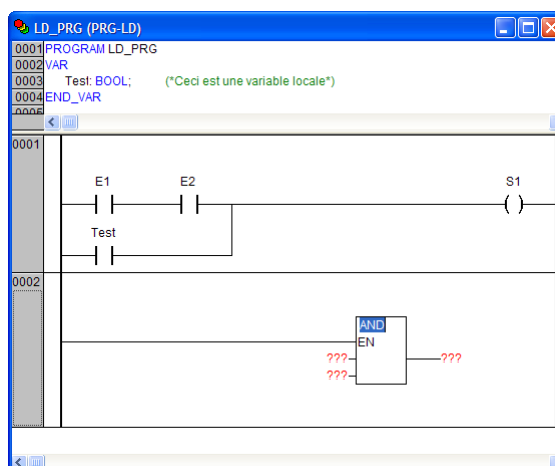




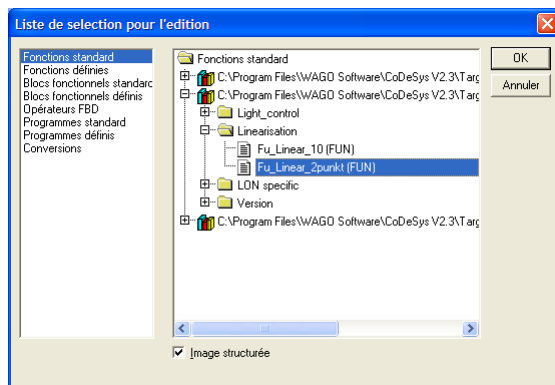
La bibliothèque apparaît ensuite dans la liste des bibliothèques insérées dans le projet :



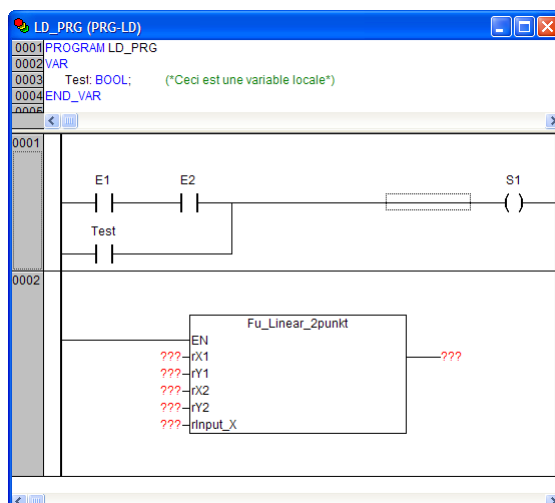
- Retourner dans la fenêtre d'édition du programme
- Ajouter un nouveau réseau
- Ajouter un **Bloc avec EN**
- Placer le curseur sur **AND**, puis appuyer sur la touche <F2> :



La liste de sélection permet désormais l'accès aux fonctions, blocs fonctionnels, programmes et type de données contenus dans la bibliothèque nouvellement insérée :



- Sélectionner par exemple la fonction **Fu\_Linear\_2punkt**, puis valider par **OK** :



Il ne reste plus qu'à placer des variables en regard des paramètres d'entrée et de sortie pour pouvoir utiliser la fonction.

## 19 Gestion des entrées/sorties analogiques

D'une manière générale, les entrées / sorties analogiques sont codées sur 16 bits. Par exemple :

- une borne de 4 entrées 4-20mA occupera 4 mots dans la table d'entrée
- une borne de 2 sorties 0-10V occupera 2 mots dans la table de sortie

Le type **WORD** est utilisé pour les données non signées (signaux 0/4-20mA, 0-10V, etc.).

Le type **INT** (signé) est utilisé dans le cas de signaux symétriques (ex : +/- 10V) ou d'entrées température (Pt100, thermocouples, etc.).

Les informations de diagnostic sont codées dans la valeur. En général, les deux bits de poids les plus faibles signalent un défaut.

Pour connaître la correspondance entre les valeurs physiques et les données renvoyées par la borne (pour les entrées) ou à envoyer à la borne (pour les sorties), il faut se référer au manuel technique de chaque borne.

Les manuels techniques de chaque borne sont disponibles sur [www.wago.fr](http://www.wago.fr) ou sur le DVD **AUTOMATION Tools and Docs** :

- > Service > Documentation technique > Documentation
- > WAGO-I/O-SYSTEM 750 > Analog Input (Output) Modules.

Dans chaque manuel, une section **Process Image** donne la correspondance entre les valeurs physiques et les valeurs numériques.

### 19.1 Exemple pour les entrées analogiques

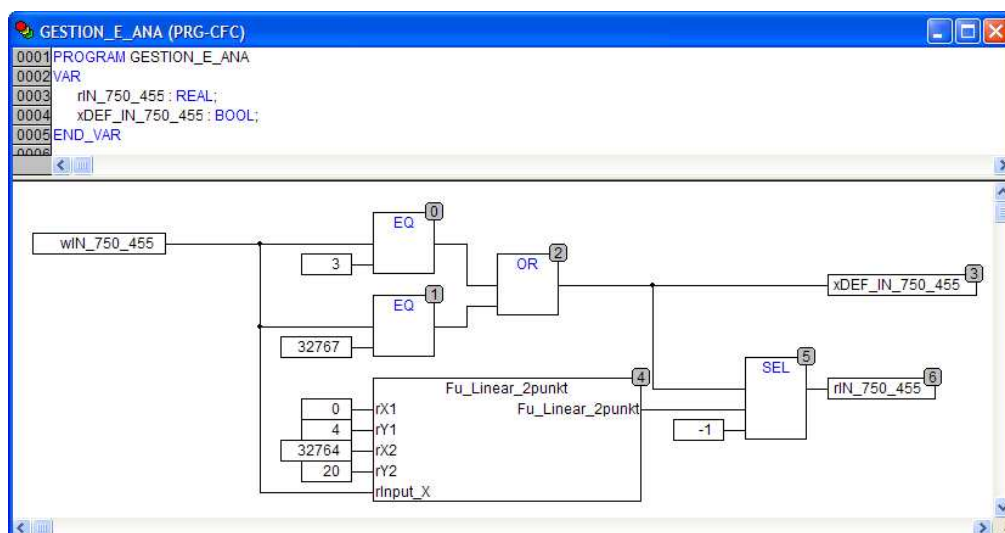
Exemple pour une borne 4 entrées 4-20 mA réf. 750-455 :

Process values of module 750-455						
Input current 4 mA - 20 mA	numerical value			Status-byte hex.	LED Error AI 1...4	
	binary Measured value	<sup>≠</sup> X F Ü	hex. dec.			
<4.0	'0000.0000.0000.0	X11'	0x0003 0	0x41	on	
4.0	'0000.0000.0000.0	X00'	0x0000 0	0x00	off	
5.6	'0000.1100.1100.1	X00'	0x0CCC 3276	0x00	off	
7.2	'0001.1001.1001.1	X00'	0x1998 6552	0x00	off	
8.8	'0010.0110.0110.0	X00'	0x2664 9828	0x00	off	
10.4	'0011.0011.0011.0	X00'	0x3330 13104	0x00	off	
12.0	'0100.0000.0000.0	X00'	0x4000 16384	0x00	off	
13.6	'0100.1100.1100.1	X00'	0x4CCC 19660	0x00	off	
15.2	'0101.1001.1001.1	X00'	0x5998 22936	0x00	off	
16.8	'0110.0110.0110.0	X00'	0x6660 26208	0x00	off	
18.4	'0111.0011.0011.0	X00'	0x7330 29488	0x00	off	
20.0	'0111.1111.1111.1	X00'	0x7FFC 32764	0x00	off	
>20.0	'0111.1111.1111.1	X11'	0x7FFF 32767	0x42	on	

Cette table montre que lorsque l'entrée mesure moins de 4 mA, le défaut est signalé par la présence à 1 des deux bits de poids les plus faibles, ce qui correspond à la valeur 3. Si l'entrée mesure plus de 20 mA, la valeur renvoyée est 32767.

Ces deux valeurs remarquables peuvent être testées dans le programme pour déterminer si l'entrée est en défaut. Si l'entrée n'est pas en défaut, on a généralement besoin de faire une mise à l'échelle avant de traiter la donnée dans le programme. Une fonction de mise à l'échelle est disponible dans la bibliothèque **Building\_Common**, située dans le dossier **Building**.

Voici un exemple de code pour la gestion des défauts et la mise à l'échelle :



Dans cet exemple, la valeur brute est de type **WORD**, et est déclarée dans le configurateur matériel. Elle s'appelle **wIN\_750\_455**.

Deux valeurs sont calculées par le programme : la valeur mise à l'échelle (**rIN\_750\_455**), et un bit de défaut (**xDEF\_IN\_750\_455**).

Si l'entrée analogique est dans la plage de fonctionnement normal, la valeur **rIN\_750\_455** est calculée, et la variable **xDEF\_IN\_750\_455** est à **FALSE**.

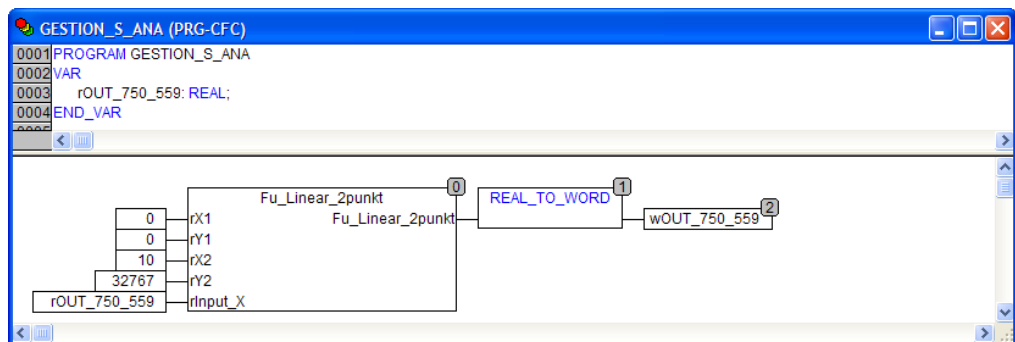
Si l'entrée analogique est hors plage, la valeur **rIN\_750\_455** est mise à une valeur de défaut (-1), et la variable **xDEF\_IN\_750\_455** est à **TRUE**.

## 19.2 Exemple pour les sorties analogiques

Autre exemple, avec une borne de 4 sorties 0-10V, réf. 750-559 :

Process values of module 750-559				
Output value 0 V - 10 V	numerical value			status byte hex.
	binary output value	hex.	dec.	
0.00	'0000.0000.0000.0000'	0x0000	0	0x00
1.25	'0001.0000.0000.0000'	0x1000	4096	0x00
2.50	'0010.0000.0000.0000'	0x2000	8192	0x00
3.75	'0011.0000.0000.0000'	0x3000	12288	0x00
5.00	'0100.0000.0000.0000'	0x4000	16384	0x00
6.25	'0101.0000.0000.0000'	0x5000	20480	0x00
7.50	'0110.0000.0000.0000'	0x6000	24576	0x00
8.75	'0111.0000.0000.0000'	0x7000	28672	0x00
10.00	'0111.1111.1111.1111'	0x7FFF	32767	0x00

Le programme correspondant est légèrement différent : il suffit de faire une mise à l'échelle dans l'autre sens (de 0-10 vers 0-32767), puis de faire une conversion de type de donnée. En effet, fonction **Fu\_Linear\_2punkt** renvoie un réel, alors que la variable qui est liée à la sortie dans le configurateur matériel est de type **WORD**. Une conversion **REAL\_TO\_WORD** est donc nécessaire :



## 19.3 Remarque sur les entrées température

Les bornes d'entrée pour sondes de température RTD (Pt100, Pt1000, Ni1000, etc.) ou thermocouples (K, J, etc.) réalisent la linéarisation en interne, renvoient la température en dixièmes de degrés Celcius.

Par exemple, pour une température mesurée de 25.4°C, la valeur retournée sera 254.

Il suffit donc de faire une division par 10 pour obtenir la température. Par contre, si l'on divise une variable de type **INT** par la valeur 10, le compilateur fera le calcul sur un entier, donc « sans virgule ». Il faut donc forcer l'une des deux opérandes en réel afin de ne pas perdre les décimales. Il existe deux possibilités :

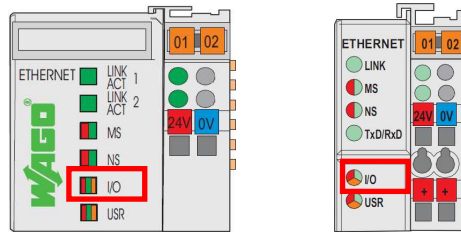
- écrire 10.0 au lieu de 10
- écrire REAL#10 au lieu de 10

Exemples (en code ST) :

```
rTemp := iTemp / 10.0 ;  
rTemp := iTemp / REAL#10 ;
```

## 20 Annexe A : LED I/O

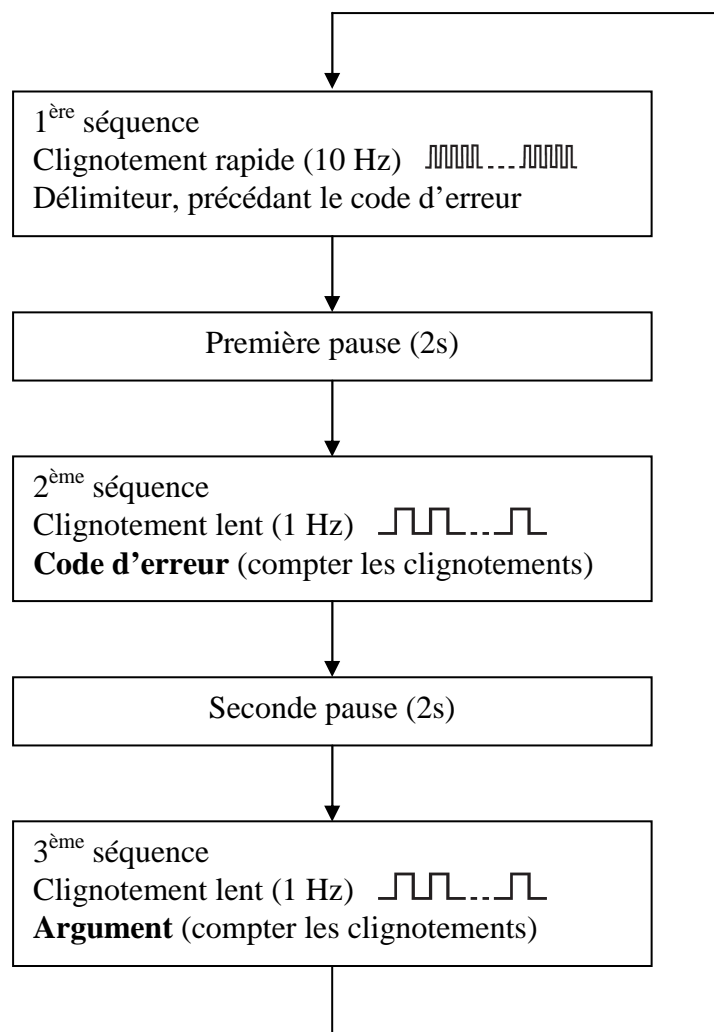
Les contrôleurs disposent d'une LED « I/O » en face avant :



Cette LED donne une information de diagnostic général sur l'état du système. Elle clignote rapidement à la mise sous tension. Ensuite :

- elle reste en VERT (fixe) s'il n'y a aucun défaut.
- s'il y a un défaut, la LED suivra des séquences de clignotement en ROUGE, pour indiquer un code d'erreur et un argument.

Les séquences de clignotement suivent la logique suivante :



La liste exhaustive des codes d'erreur et des arguments peut être trouvée dans les manuels techniques des contrôleurs.

Le tableau ci-dessous synthétise les codes d'erreur les plus fréquemment rencontrés, ainsi que les solutions proposées :

Code d'erreur + argument	Description	Solution
1 + 10	Horloge interne désynchronisée	<ol style="list-style-type: none"> <li>1. Mettre à l'heure l'horloge</li> <li>2. Maintenir le contrôleur sous tension durant au moins 15 minutes pour charger les condensateurs</li> </ol>
3 + 0	Défaut de communication sur le bus interne, le module défectueux n'a pu être déterminé	<p>- Si des bornes de réalimentation du bus interne 750-613 sont présentes :</p> <ol style="list-style-type: none"> <li>1. Vérifier qu'elles sont bien toutes alimentées, en vérifiant que leur LED "A" est bien allumée</li> </ol> <p>- Si toutes les bornes 750-613 sont correctement alimentées, il faut déterminer la borne défectueuse :</p> <ol style="list-style-type: none"> <li>1. Arrêter le programme automate en mettant le switch de mode de fonctionnement sur STOP</li> <li>2. Mettre le bornier hors tension</li> <li>3. Insérer la borne finale 750-600 au milieu du bornier</li> <li>4. Remettre le contrôleur sous tension</li> <li>5. - La LED I/O continue à clignoter : couper l'alimentation et insérer la borne finale 750-600 au milieu de la première moitié du bornier</li> <li>- La LED I/O arrête de clignoter : couper l'alimentation et insérer la borne finale 750-600 au milieu de la seconde moitié du bornier</li> <li>6. Remettre le bornier sous tension</li> <li>7. Répéter la procédure décrite à l'étape 4 jusqu'à détection de la borne défectueuse</li> <li>8. Remplacer la borne défectueuse</li> <li>9. Remettre le bornier sous tension</li> <li>10. Remettre le switch sur la position RUN</li> </ol>
4 + n	Bus interne interrompu après le module n.	<ol style="list-style-type: none"> <li>1. Vérifier que le loquet de verrouillage du contrôleur est bien verrouillé</li> <li>2. Vérifier que les modules d'E/S sont bien correctement encliquetés sur le rail</li> <li>3. Vérifier la présence de la borne finale</li> </ol> <p>Si ces actions n'ont pas résolu le problème :</p> <ol style="list-style-type: none"> <li>1. Mettre le bornier hors tension</li> <li>2. Remplacer le module n+1</li> <li>3. Remettre le bornier sous tension</li> </ol>
6 + 4	Erreur de configuration du réseau : pas d'adresse IP	<p>Si l'adresse IP est affectée depuis WAGO Ethernet Settings, recommencer la procédure décrite au début de ce document.</p> <p>Si l'adresse IP est affectée par protocole BootP</p> <ol style="list-style-type: none"> <li>1. Vérifier le paramétrage du WAGO BootP Server et le démarrer</li> <li>2. Redémarrer le contrôleur</li> <li>3. Dévalider le protocole BootP depuis la page PORT du serveur web</li> </ol> <p>La procédure complète est décrite dans le chapitre "Annexe B : affectation de l'adresse IP par protocole BootP"</p>
6+7	Doublon d'adresse IP sur le réseau	<ol style="list-style-type: none"> <li>1. Affecter au contrôleur une adresse IP disponible</li> <li>2. Redémarrer le contrôleur</li> </ol>
6 + 9 (sur 750-830)	Discordance entre le programme CoDeSys et le fichier SYM_XML importé dans le WAGO BACnet Configurator	<ol style="list-style-type: none"> <li>1. Sauvegarder le programme dans le contrôleur (<i>En ligne \ Créer projet d'initialisation</i>)</li> <li>2. Réimporter le fichier SYM_XML dans le WAGO BACnet Configurator et recharger la configuration en cliquant sur le bouton Store &amp; Download</li> </ol>
6 + 9 (sur 750-849)	Discordance entre le programme CoDeSys et le fichier SYM_XML importé dans le plug-in ETS	<ol style="list-style-type: none"> <li>1. Sauvegarder le programme dans le contrôleur (<i>En ligne \ Créer projet d'initialisation</i>)</li> <li>2. Réimporter le fichier SYM_XML dans le plug-in ETS et recharger le programme d'application de l'IP-Controller</li> </ol>
6 + 10 (sur 750-882)	Erreur de configuration du réseau : pas d'adresse IP sur le port n°2	Voir solution du code d'erreur 6 + 4 et l'appliquer sur le port n°2.
10 + 5	Discordance entre les bornes d'E/S connectées et la configuration matérielle déclarée sous CoDeSys	<ol style="list-style-type: none"> <li>1. Vérifier la configuration matérielle déclarée sous CoDeSys</li> <li>2. Recharger le programme dans le contrôleur</li> <li>3. Démarrer le programme</li> </ol>





---

**Informations complémentaires**

Si le code d'erreur rencontré n'est pas listé dans le tableau précédent, se référer au manuel technique du contrôleur.

Le manuel technique peut être téléchargé depuis Internet, et est également présent sur le DVD **AUTOMATION Tools and Docs** :

[www.wago.fr](http://www.wago.fr) > Services > Documentation Technique > Documentation > WAGO-I/O-SYSTEM 750 > Fieldbus Coupler and Programmable Fieldbus Controller

---



---

**Remarque**

Les valeurs du code d'erreur et de l'argument peuvent respectivement être lues par des requêtes de lectures mots via les registres 0x1020 (adresse Modbus 404129) et 0x1021 (adresse Modbus 404130).

---



---

**Remarque**

Si la LED I/O clignote en ORANGE :

1. Mettre le contrôleur hors tension
2. Déconnecter le câble de configuration USB ou RS232
3. Remettre le contrôleur sous tension

S'il n'y avait pas de câble connecté, ou si le défaut persiste après la déconnexion du câble, contacter le support technique.

---

## 21 Annexe B : affectation de l'adresse IP par protocole BootP

Il est possible d'affecter l'adresse IP d'un contrôleur Ethernet 32 bits en utilisant le protocole **BootP** (*Bootstrap Protocol*). Cette configuration alternative est utile lorsque l'on ne dispose pas du câble de configuration RS232 ou USB, sur les contrôleurs ne disposant pas de micro-interrupteurs de réglage de l'adresse IP.

Le protocole BootP permet d'affecter une adresse IP à un équipement, à partir de son **adresse MAC**. L'adresse MAC est une adresse matérielle unique, fixée par le constructeur en usine.

L'adresse MAC des contrôleurs Ethernet WAGO peut être relevée sur l'étiquette collée sur le côté gauche de l'appareil. Elle est également sérigraphiée sur le côté droit du contrôleur, où cas où l'étiquette ne serait plus présente (démontage nécessaire).



### Remarque

Les contrôleurs 750-882 disposent de deux adresses MAC. L'adresse qui est indiquée sur l'étiquette, et qui est sérigraphiée sur le côté droit, correspond à l'adresse MAC du port 1. Pour déterminer l'adresse MAC du port 2, il suffit d'ajouter 1 à celle du port 1.



### Attention

Si le contrôleur utilisé dispose de micro-interrupteurs de réglage de l'adresse IP, tous les micro-interrupteurs doivent être placés sur la position 0.

Le logiciel **WAGO BootP Server** est disponible sur [www.wago.fr](http://www.wago.fr) ou sur le DVD **AUTOMATION Tools and Docs** :

> Service > Téléchargements > AUTOMATION  
> WAGO-I/O-SYSTEM 750/753 > WAGO BootP Server

- Extraire le contenu du dossier compressé **BootPSer.zip**
- Se rendre dans le dossier **disk1**
- Exécuter le fichier **SETUP.EXE**
- Valider les différentes fenêtres en cliquant sur **Weiter** >
- Terminer l'installation en cliquant sur **Beenden**

Une fois l'installation terminée :

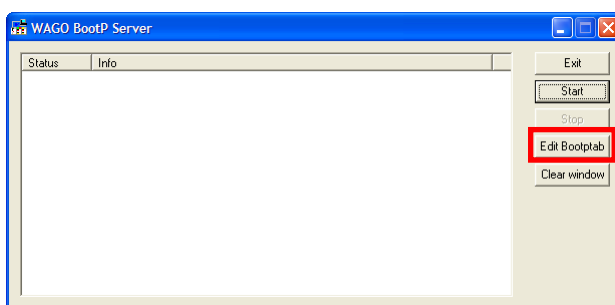
- Exécuter **WAGO BootP Server** à partir du menu *Démarrer \ Programmes \ WAGO Software \ WAGO BootP Server*



### Remarque

Sous Windows 7, lancer le logiciel avec les droits administrateurs.

- Cliquer sur le bouton **Edit Bootptab** :



Le **Bloc-notes** s'ouvre :

 A screenshot of a Notepad window titled 'bootptab.txt - Bloc-notes'. The window shows the following text:
 

```
bootptab.txt: database for bootp server
# Blank lines and lines beginning with '#' are ignored.
#
# Legend:
#   first field -- hostname (not indented)
#   bf -- bootfile
#   bs -- bootfile size in 512-octet blocks
#   cs -- cookie servers
#   df -- dump file name
#   dn -- domain name
#   ds -- domain name servers
#   ef -- extension file
#   gw -- gateways
#   ha -- hardware address
#   hd -- home directory for bootfiles
#   hn -- host name set for client
#   ht -- hardware type
#   im -- impress servers
#   ip -- host IP address
#   lg -- log servers
#   lp -- LPR servers
#   ns -- NIS-116 name servers
#   ra -- reply address
#   rl -- resource location protocol servers
#   rp -- root path
#   sa -- boot server address
#   sm -- subnet mask
#   sw -- swap server
#   tc -- template host (points to similar host entry)
#   td -- TFTP directory
#   to -- time offset (seconds)
#   ts -- time servers
#   vm -- vendor magic number
#   Tn -- generic option tag n
#
# Be careful about including backslashes where they're needed. Weird (bad)
# things can happen when a backslash is omitted where one is intended.
# Also, note that generic option data must be either a string or a
# sequence of bytes where each byte is a two-digit hex value.
#
# Example of entry with no gateway
KeinProxy:ht=1:ha=0030DE000002:ip=10.1.254.202:
# Example of entry with gateway
# The gateway address must be inserted in hexadecimal
# after the T3 parameter
hamburg:ht=1:ha=0030DE000003:ip=10.1.254.203:T3=0A.01.FE.01:
```

Toutes les lignes commençant par un # sont des lignes de commentaires.

Une ligne de paramétrage typique est composée des champs suivants :

xxx	Nom symbolique du contrôleur
ht	Spécification du réseau. Pour Ethernet, cette valeur est 1
ha	Adresse MAC
ip	Adresse IP
sm	Masque de sous-réseau
gw (optionnel)	Adresse de passerelle

Dans cet exemple, on relève l'adresse MAC du contrôleur : 0030DE03F645.

On souhaite ensuite lui affecter les paramètres réseau suivants :

Adresse IP	192.168.1.1
Masque de sous-réseau	255.255.255.0
Adresse de passerelle	sans

- Mettre en commentaire les deux lignes qui ne commencent pas par un #
- ajouter la ligne de configuration en fin de fichier, en ajustant l'adresse MAC et les paramètres réseau à affecter au contrôleur :

```
CTRL:ht=1:ha=0030DE03F645:ip=192.168.1.1:sm=255.255.255.0:
```



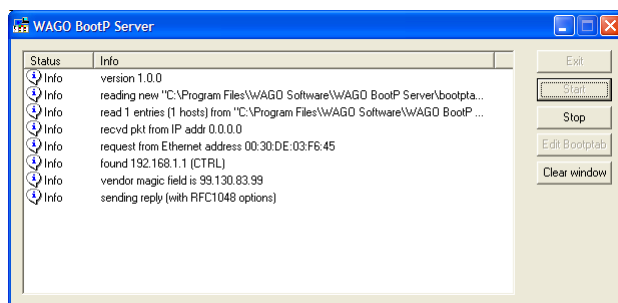
### Remarque

Si l'on souhaite configurer une adresse de passerelle, il faut ajouter un champ GW. Exemple :

```
CTRL:ht=1:ha=0030DE03F645:ip=192.168.1.1:sm=255.255.255.0:gw=192.168.1.254:
```

- Fermer le **Bloc-notes** en enregistrant les modifications
- Cliquer sur **Start**

Après quelques secondes, l'écran suivant apparaît :

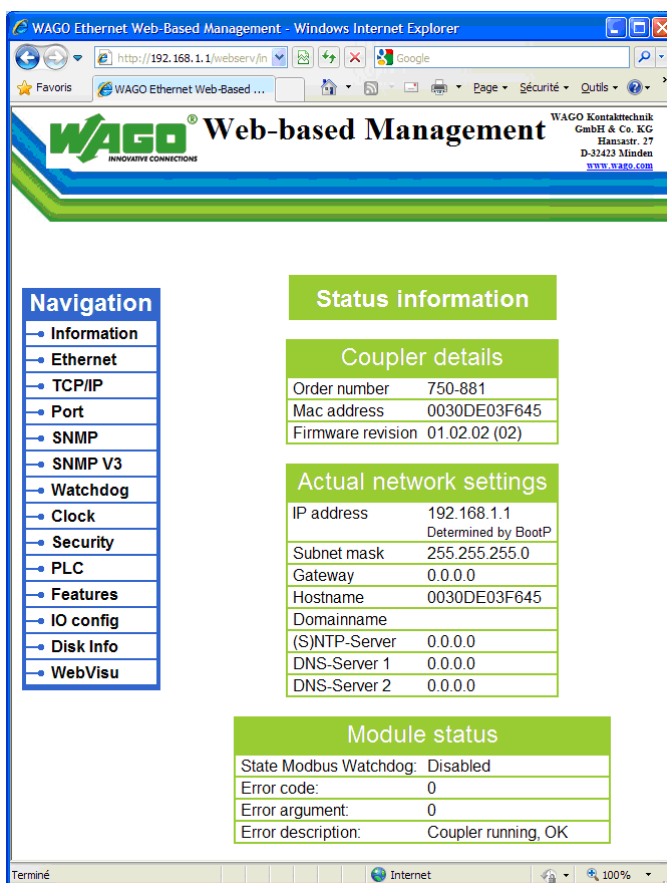


La ligne **found 192.168.1.1 (CTRL)** indique que les paramètres réseau ont bien été affectés au contrôleur.

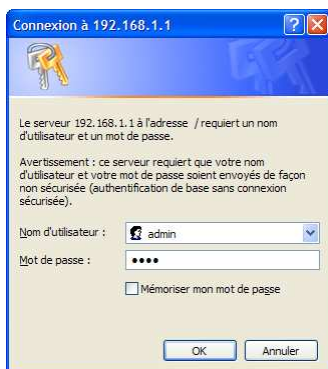
- Cliquer sur **Stop**
- Cliquer sur **Exit** pour quitter l'application

Il est ensuite nécessaire de désactiver les requêtes BootP.

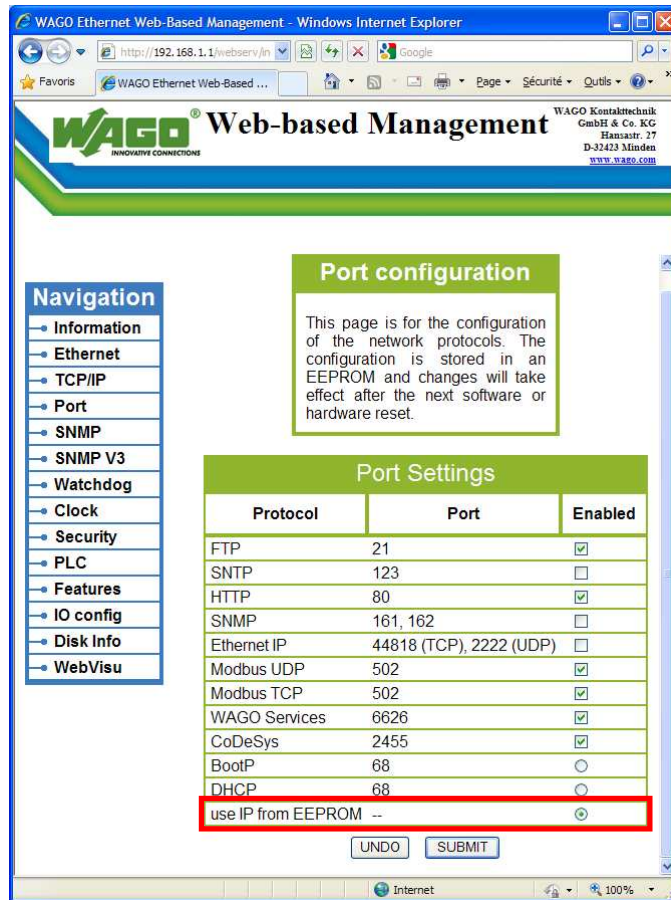
- Ouvrir un navigateur web et saisir l'adresse IP du contrôleur dans la barre d'adresse :



- Cliquer sur la page **Port**
- le nom d'utilisateur est **admin**, et le mot de passe est **wago** :



- Sélectionner **use IP from EEPROM**
- Valider en cliquant sur **SUBMIT**

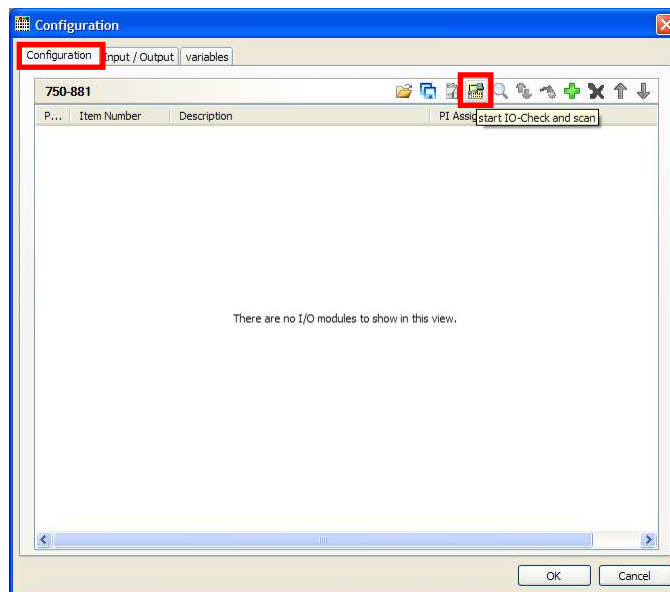


## 22 Annexe C : lecture de la configuration matérielle depuis WAGO-I/O-CHECK

Si le logiciel WAGO-I/O-CHECK est installé sur le PC, il est possible de faire une lecture automatique de la configuration matérielle.

Sous CoDeSys :

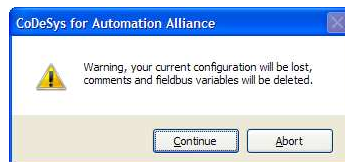
- Développer l'arborescence **Hardware Configuration**
- Faire un clic droit sur **K-Bus[FIX]**, puis sélectionner **Ajouter sous-élément**
- Cliquer sur l'onglet **Configuration**, puis sur le bouton **start IO-Check and scan** :



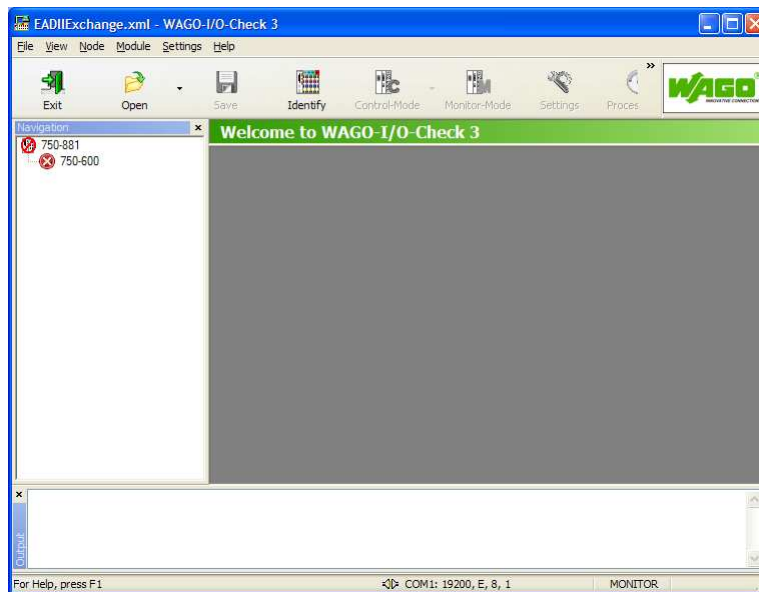
### Attention

La lecture automatique de la configuration matérielle effacera l'éventuelle configuration matérielle précédemment saisie !

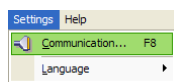
- Valider le message d'avertissement



La fenêtre de **WAGO-I/O-CHECK** s'ouvre :



- Cliquer sur *Settings \ Communication...*



- Dans le champ **Connection**, sélectionner **Ethernet (TCP/IP)**
- Renseigner l'adresse IP du contrôleur dans le champ **IP-Address**



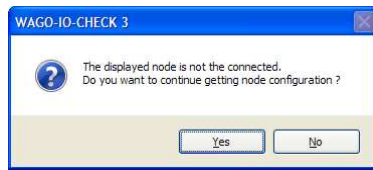
- Valider la fenêtre en cliquant sur **OK**

- De retour dans la fenêtre principale de WAGO-I/O-CHECK, cliquer sur le bouton **Identify** :

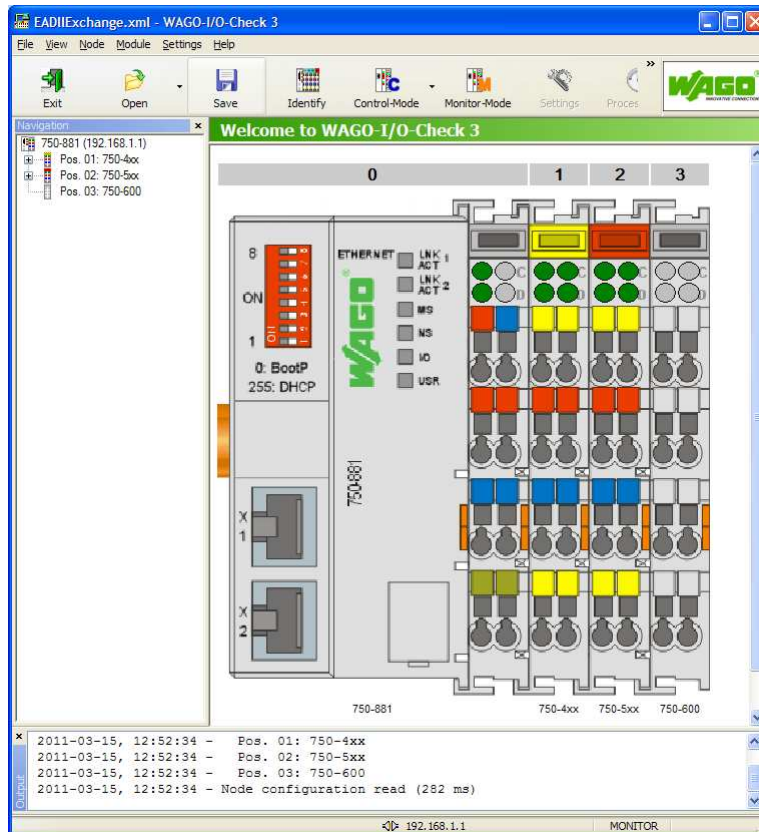




- Valider le message :

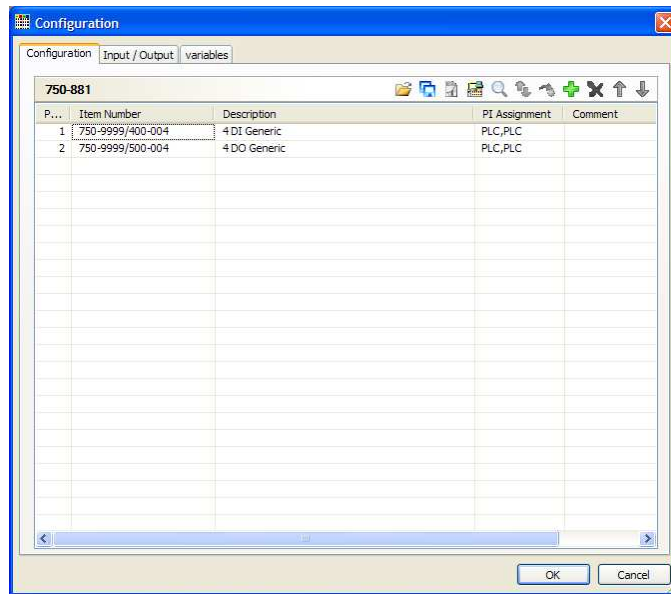


La configuration apparaît :



- Cliquer ensuite sur les boutons **Save** et **Exit**

De retour dans CoDeSys, la configuration matérielle apparaît :

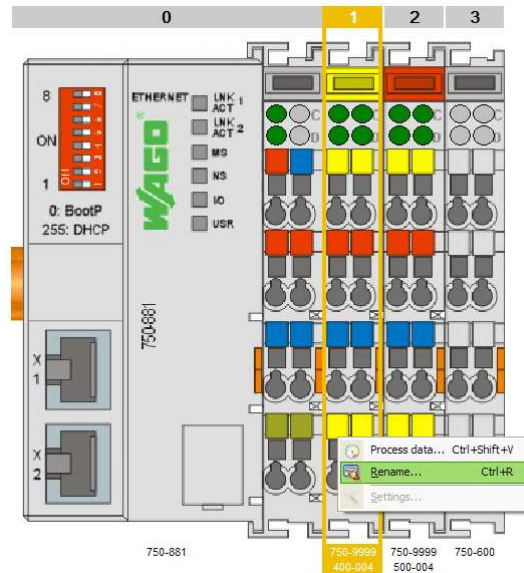




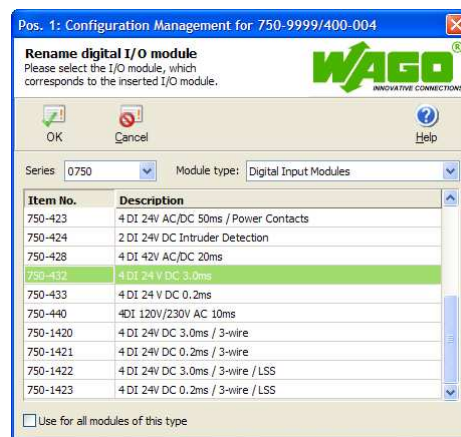
### Remarque

Pour les bornes d'E/S TOR, la référence exacte du produit ne peut être déterminée automatiquement.

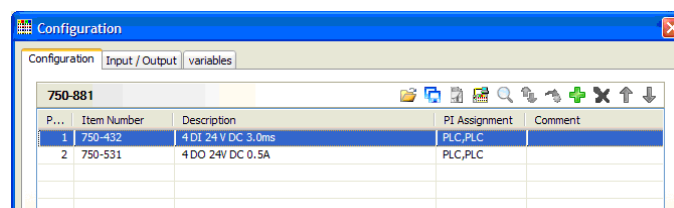
Si l'on souhaite que les références exactes apparaissent sous CoDeSys, au lieu de références génériques, il est possible de spécifier les références sous WAGO-I/O-CHECK, en faisant un clic droit sur la borne concernée, et en cliquant sur **Rename** :



Il suffit ensuite de sélectionner la bonne référence parmi la liste proposée :



Les références exactes apparaîtront ensuite sous CoDeSys :



## 23 Annexe D : mise à jour du firmware du contrôleur

En fonction du contrôleur et de la version de firmware, le support technique WAGO peut fournir deux logiciels pour la mise à jour :

- Le logiciel **WAGO FBC Update** permet de mettre à jour le contrôleur depuis le câble USB ou RS232
- Le logiciel **WAGO Ethernet Update** permet de mettre à jour le contrôleur depuis le réseau Ethernet

Ce chapitre est illustré par la mise à jour d'un contrôleur Ethernet 750-881 en firmware 02 vers la version 03.

### 23.1 WAGO FBC Update



---

#### Attention

Ne pas tenter de mettre à jour un contrôleur à l'aide de l'adaptateur radio Bluetooth réf. 750-921. Le logiciel WAGO FBC Update ne peut être utilisé qu'avec le câble USB réf. 750-923 ou le câble RS232 réf. 750-920 !

---

Pour paramétrer le contrôleur par logiciel, il faut installer le pilote de câble USB, ainsi que le logiciel **WAGO FBC Update**.



---

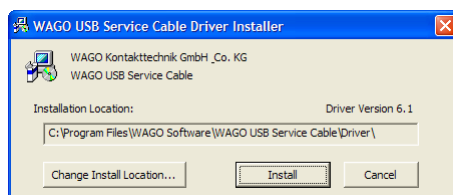
#### Avertissement

Ne pas connecter le câble USB au PC avant d'avoir installé le pilote !

---

Pour installer le pilote du câble USB :

- extraire le contenu du dossier compressé **759-923**
- Aller jusqu'au dossier  
*WAGOServiceCable\_DriverInstall\WIN2k\_XP\_VISTA\_WIN7,*
- exécuter **Setup.exe**
- Cliquer sur le bouton **Install**

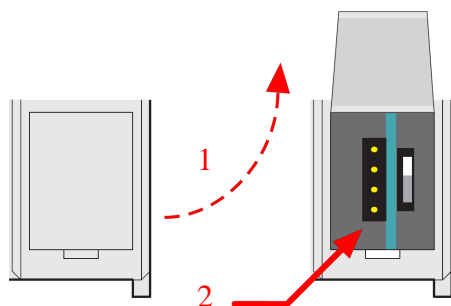




### Avertissement

Le câble USB ne doit jamais être connecté à chaud, lorsque le contrôleur est sous tension. Risque de destruction matérielle !

- Mettre le contrôleur hors tension
- Ouvrir la trappe située en face avant du contrôleur (1) :



- Insérer délicatement le connecteur spécifique du câble sur les 4 broches situées du côté gauche du circuit imprimé (2)
- Raccorder le connecteur USB du câble au PC
- Remettre le contrôleur sous tension

L'étape suivante consiste à installer le logiciel **WAGO FBC Update**, puis le fichier de firmware fourni :

- Exécuter **WAGO\_FBC-Update\_Setup(x.y.z).exe**  
(ici, WAGO\_FBC-Update\_Setup(4.1.1).exe) :

- A la fin de l'installation, décocher la case **Run WAGO-FBC-Update**, puis cliquer sur **Finish** :



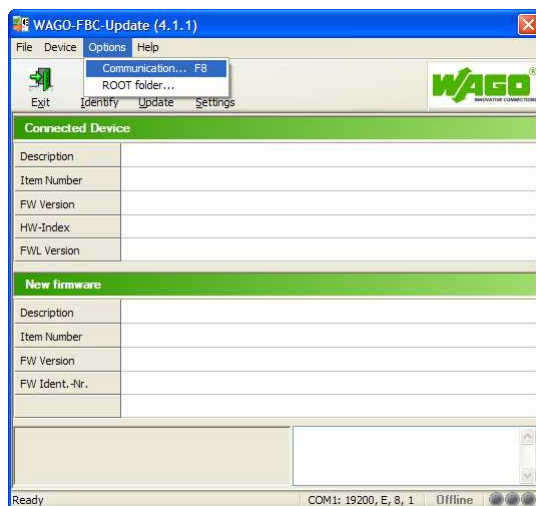
- Exécuter ensuite l'application d'installation du firmware **Setup\_075008xx\_Vyyyyyy\_IXzz.exe**  
(ici, Setup\_07500881\_V010205\_IX03.EXE) :

- A la fin de l'installation, laisser cochée la case **Yes, I want to launch WAGO-FBC-Update now**, puis cliquer sur **Finish** :

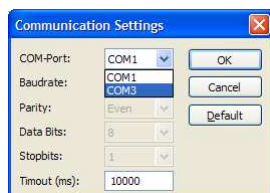


Le logiciel **WAGO-FBC-Update** se lance.

- Cliquer ensuite sur *Options \ Communication...*



- Sélectionner le port COM sur lequel est connecté le câble USB ou RS232 (ici, COM3), puis valider la fenêtre en cliquant sur **OK** :



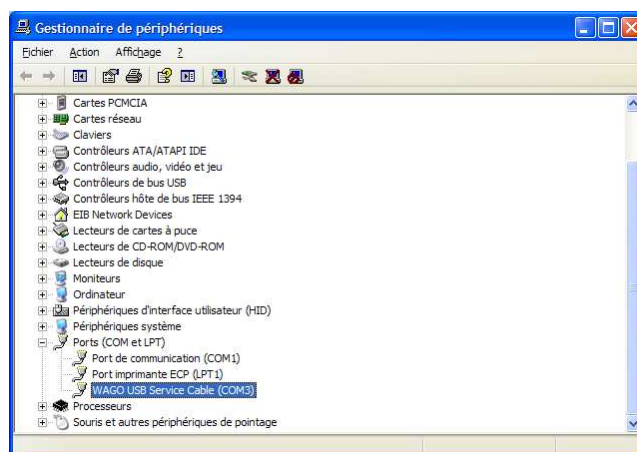


### Remarque

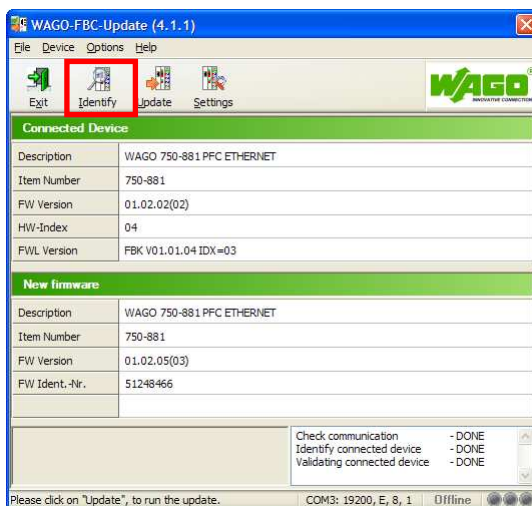
Si le numéro de port COM utilisé par le câble USB n'est pas connu, il est possible de le déterminer de la manière suivante :

- Cliquer sur *Démarrer \ Poste de travail*
- Cliquer sur **Afficher les informations système**
- Cliquer sur l'onglet **Matériel**, puis sur le bouton **Gestionnaire des périphériques**

Le câble USB et le numéro de port COM correspondant apparaissent dans la section **Ports (COM et LPT)** :



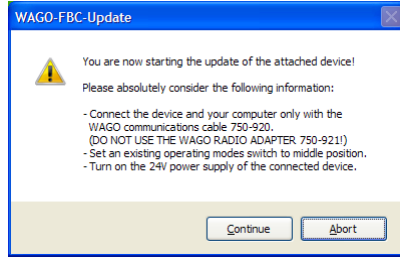
- Cliquer sur le bouton **Identify**



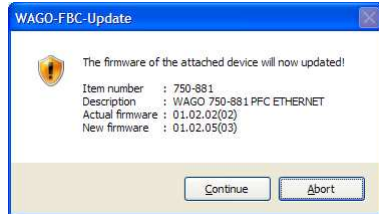
Les informations sur la version actuelle du contrôleur sont affichées dans le champ **Connected (P)FC**. Le logiciel propose la mise à jour avec la version affichée dans le champ **New Firmware**.

- Cliquer sur le bouton **Update**

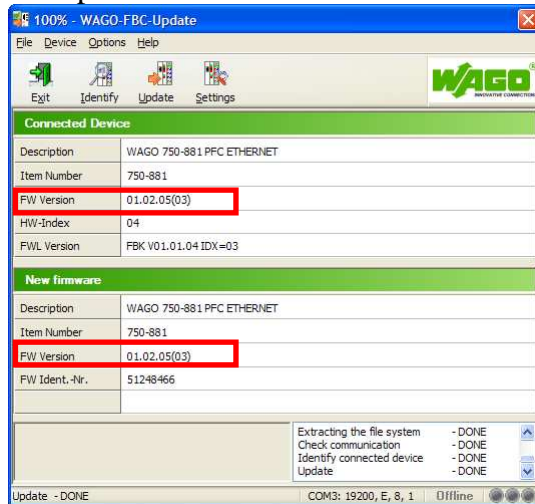
- Valider la fenêtre d'avertissement suivante :



- Valider la fenêtre de confirmation :

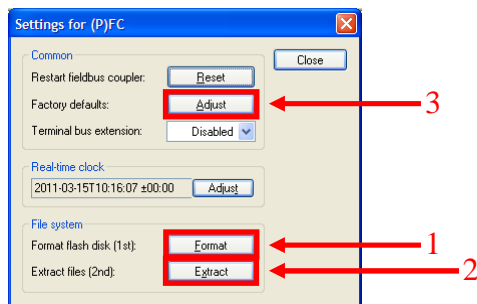


Une fois la mise à jour terminée, le logiciel indique le message **Update – DONE** dans la barre d'état, et les versions de firmware indiquées dans les champs **Connected Device** et **New firmware** sont identiques :



- Cliquer ensuite sur le bouton **Settings**

- Cliquer successivement sur les boutons **Format / Extract / Ajust**



Le contrôleur est alors mis à jour, et tous les paramètres sont réinitialisés tels qu'ils étaient en sortie d'usine.



## 23.2 WAGO Ethernet Update

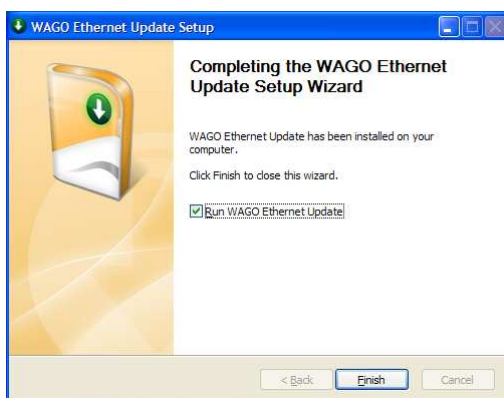


### Attention

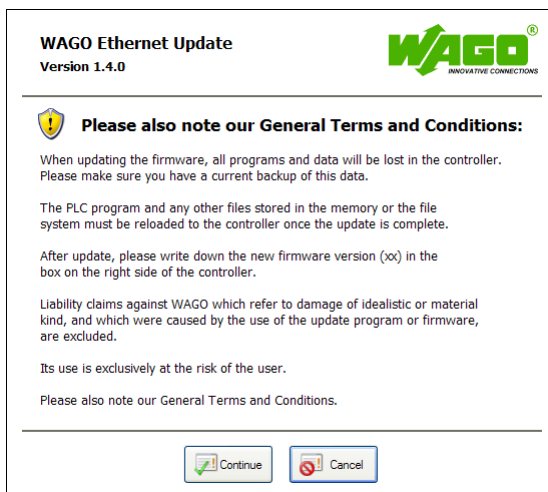
Ne pas tenter de mettre à jour un contrôleur au travers d'une connexion à distance (ADSL, GPRS, etc.). Le logiciel **WAGO Ethernet Update** ne doit être utilisé que sur un réseau local !

- Exécuter **WAGOEthernetUpdate\_Setup(x.y).exe**  
(ici, WAGOEthernetUpdate\_Setup(1.4).exe)

- A la fin de l'installation, laisser cocher la case **Run WAGO Ethernet Update**, puis cliquer sur **Finish**



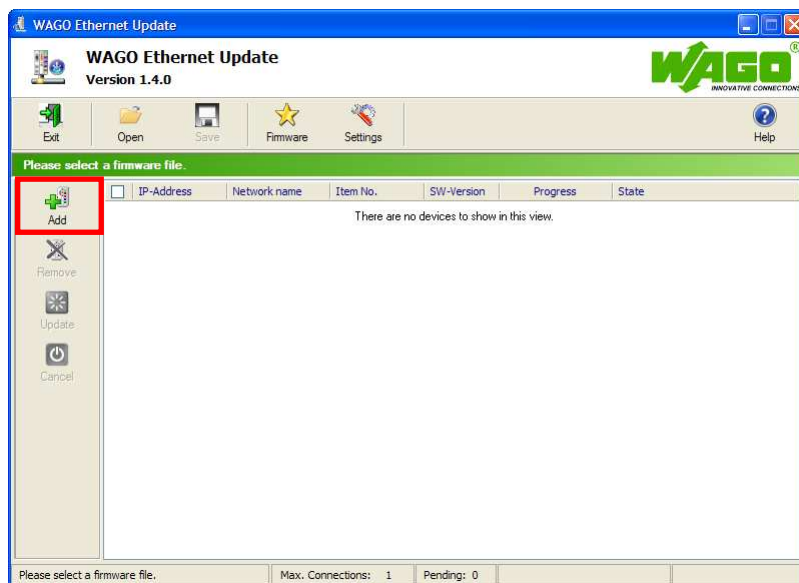
- Valider la fenêtre décrivant les conditions générales d'utilisation en cliquant sur **Continue** :



### Attention

Ne pas couper l'alimentation durant la mise à jour du firmware !

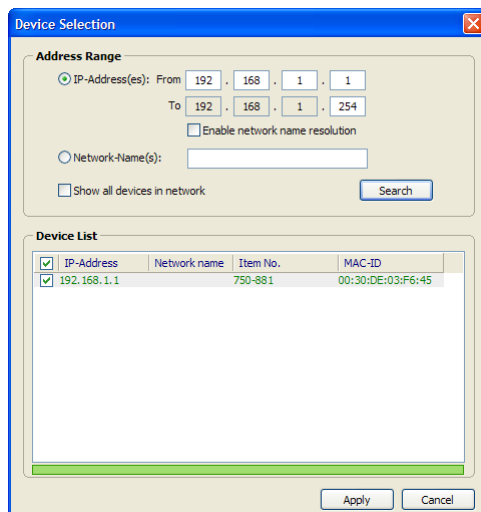
- Cliquer ensuite sur le bouton **Add**



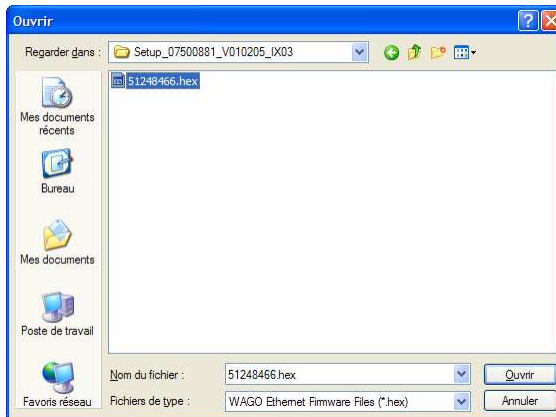
- Renseigner la plage d'adresses IP dans laquelle le(s) contrôleur(s) à mettre à jour est (sont) situé(s), en remplissant le champ **IP-Address(es)**

La liste des contrôleurs disponibles apparaît.

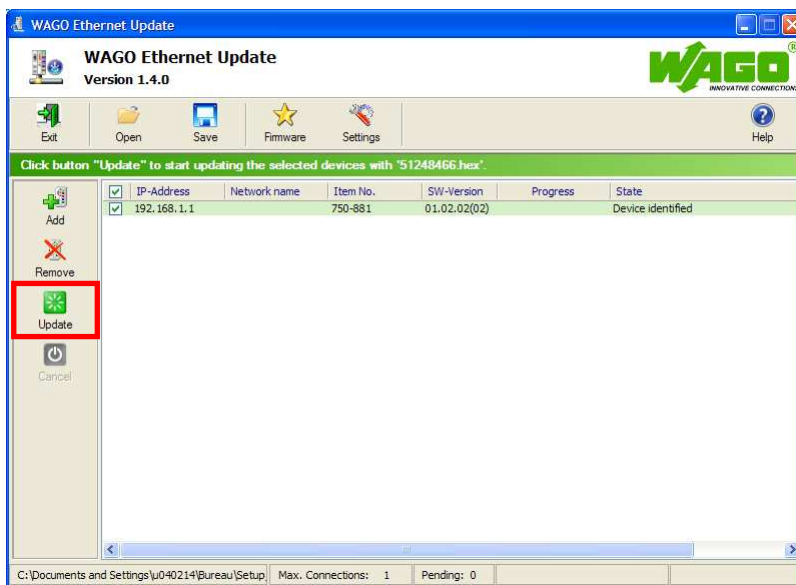
- Laisser uniquement coché(s) le(s) contrôleur(s) à mettre à jour, puis valider en cliquant sur **Apply** :



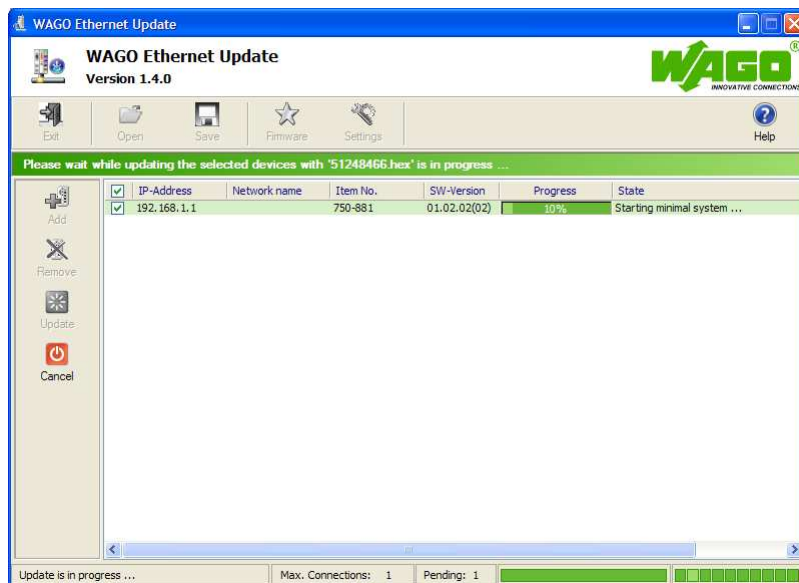
- De retour dans la fenêtre principale, cliquer sur **Firmware**.
- Sélectionner le fichier .hex fourni par WAGO, puis cliquer sur **Ouvrir** :



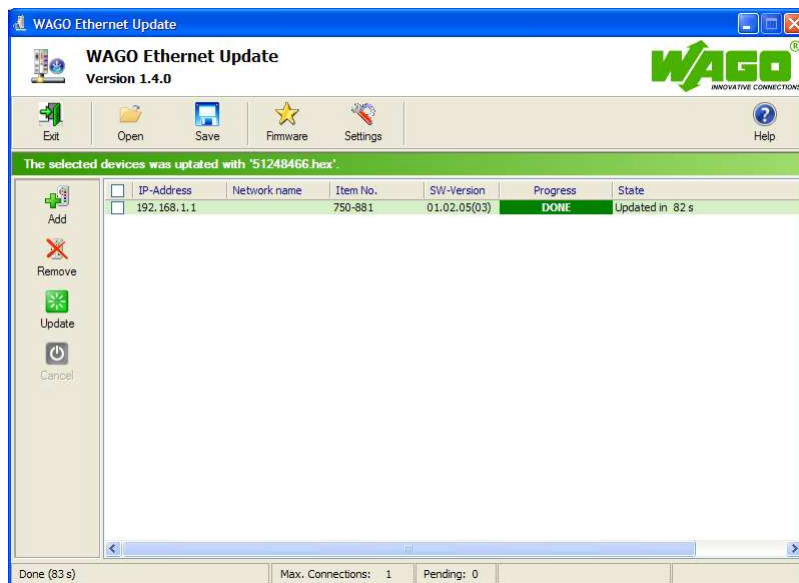
- De retour dans la fenêtre principale, cliquer sur **Update** :



La mise à jour commence :



Une fois la mise à jour terminée, l'information **DONE** apparaît en regard du (des) contrôleurs mis à jour :



### Remarque

Une fois la mise à jour terminée, la mémoire flash a été reformatée, et le système de fichiers a été réinitialisé. Les paramètres stockés en EEPROM, comme les paramètres réseau, sont conservés.

## 24 Informations complémentaires

Un portail web dédié au support technique des produits AUTOMATION est disponible sur Internet à partir de l'adresse suivante :

<http://www.wago.fr/support>

Ce portail donne notamment accès aux manuels techniques des différents produits, ainsi qu'aux notes d'application disponibles.

L'assistance technique AUTOMATION (*hotline*) est librement accessible aux coordonnées suivantes :

Tél : +33 (0)1 48 17 54 54

Fax : +33 (0)1 48 17 54 08

Email : [support-fr@wago.com](mailto:support-fr@wago.com)



WAGO Contact SAS  
83 rue des chardonnerets  
BP 55065 - ZI Paris Nord 2  
95947 Roissy Charles de Gaulle Cedex  
Tél : +33 (0)1 48 17 25 90  
Fax : +33 (0)1 48 63 25 20  
E-Mail: [info-fr@wago.com](mailto:info-fr@wago.com)  
Internet: <http://www.wago.fr>

---