

MANIP 5 : Premiers pas avec l'API WAGO

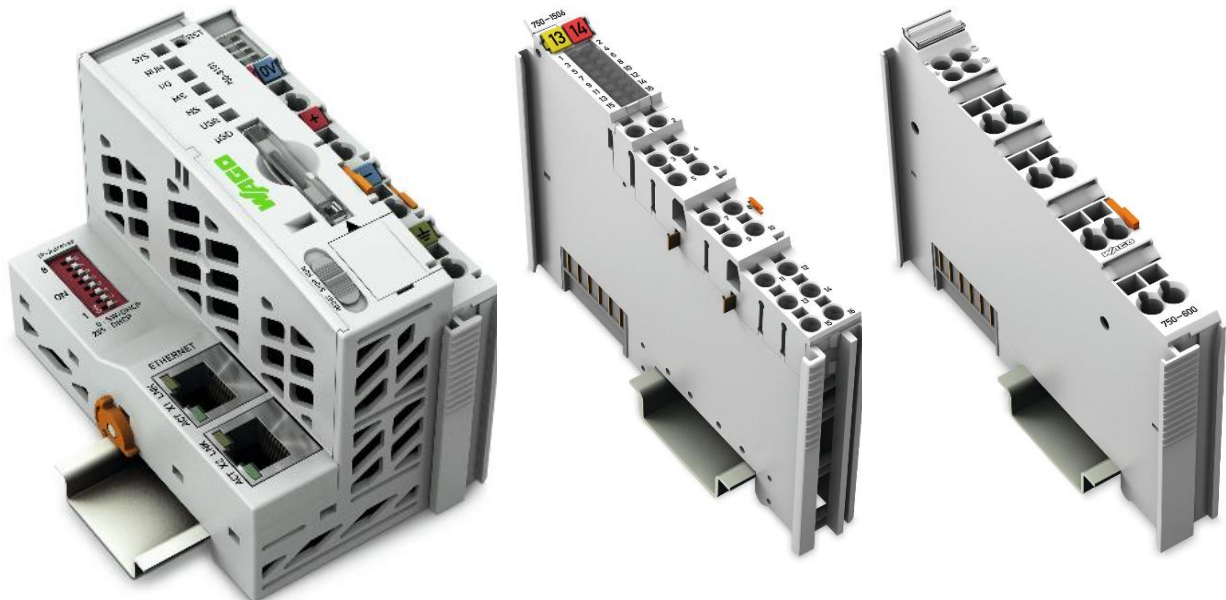
Introduction au matériel

Nous aurons besoin d'alimenter les différents éléments avec du 24 V DC ; nous pouvons choisir d'utiliser l'alimentation WAGO ou une autre source de tension.

Nous conserverons les conventions bleu / brun pour les potentiels respectifs 0V / 24V.

Nous fixerons sur un rail DIN trois modules qui seront assemblés et interconnectés :

- Le contrôleur PFC100 (article 750-8101)
- Le module d'entrées/sorties type PNP (article 750-1506)
- Le module de terminaison (article 750-600)



• Le contrôleur PFC100

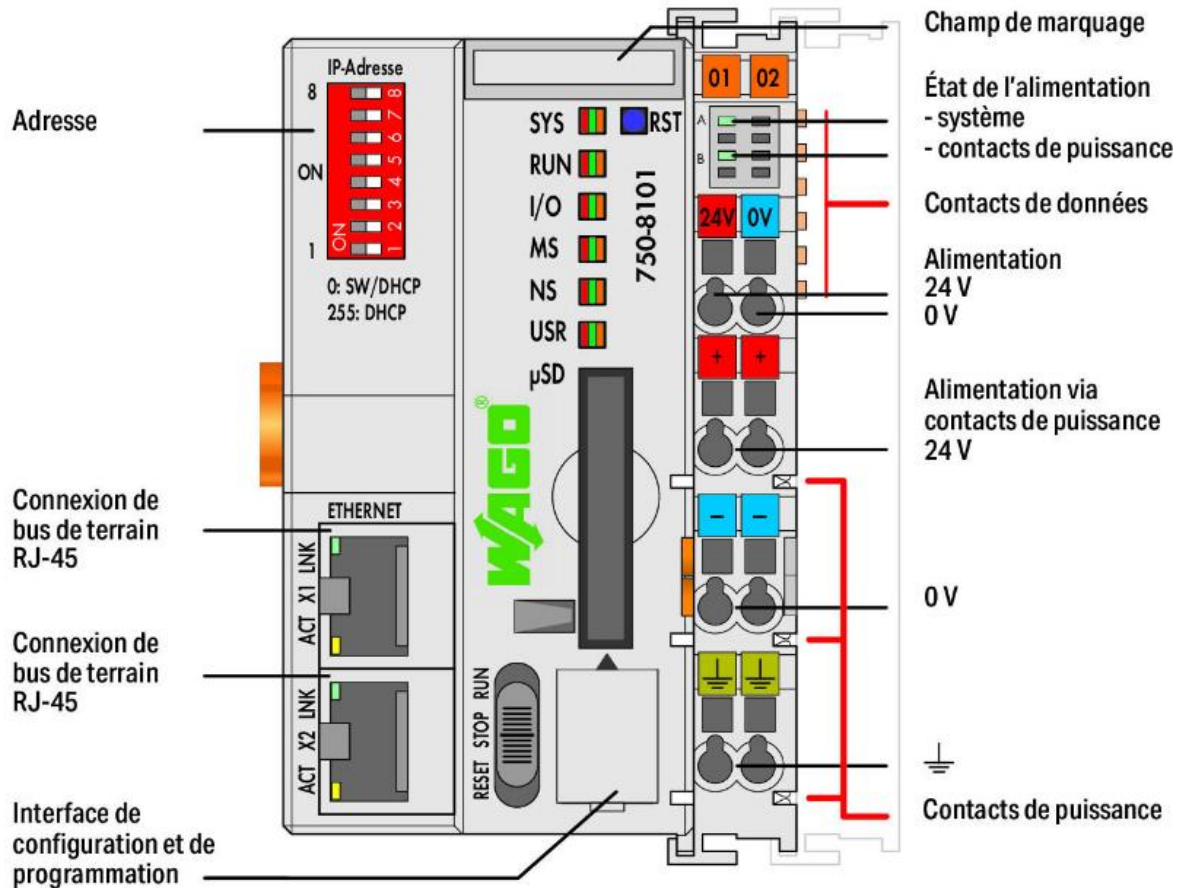
Le contrôleur est l'élément à gauche ci-dessus, on trouvera :

- La possibilité de configurer le dernier octet de l'adresse IP.
- Deux connecteurs RJ45 pour l'interconnexion au PC + API,...
- Un interrupteur 3 positions (Reset, Stop, Run).
- Un emplacement pour une microSD
- Des voyants lumineux + un accès au reset
- Une partie destinée aux interconnexions électriques

Il faut savoir que même si les connecteurs RJ45 laissent penser qu'on connecte le contrôleur à Internet, ce n'est pas toujours vrai. On peut très bien décider de travailler au sein d'un réseau privé (sans connexion Internet) ou sur un réseau public (Internet).

Le contrôleur peut utiliser Ethernet comme bus de terrain (Modbus TCP par exemple) et est capable de prendre en charge certains protocoles classiques (HTTP, FTP, DHCP, ...).

MANIP 5 : Premiers pas avec l'API WAGO

Descriptif du module PFC100 :


Le dernier octet de l'adresse IP peut être configuré grâce à l'emplacement représenté en haut à gauche. Le LSB correspond au n°1 et le MSB au n°8.

Le contrôleur doit être alimenté avec une alimentation typique 24V DC.

On assurera également un pontage entre 24V ↔ + et entre 0V ↔ - afin d'utiliser les contacts de puissance pour les autres modules.

- **Le module d'entrées / sorties**

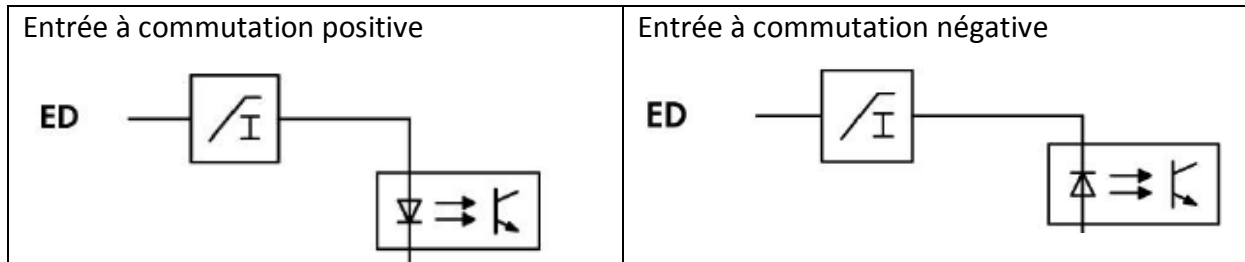
Lorsqu'un API reçoit par exemple un signal d'entrée d'un capteur, il faut une compatibilité entre le mode de fonctionnement du capteur et le module d'entrée. Si on a un capteur de type PNP (le plus courant en Europe), l'API doit avoir sa résistance d'entrée du module entre l'entrée et la masse. Le module Wago que nous utilisons (article 750-1506) utilise l'appellation « à commutation positive » au lieu de PNP.

Il existe aussi des modules appelés « à commutation négative » qui correspondent à une entrée NPN (par exemple l'article 750-1407).

Pour l'étage de sortie du module on parle aussi de commutation positive ou négative. Dans notre cas, nous avons une sortie à commutation positive → le module fournit le courant.

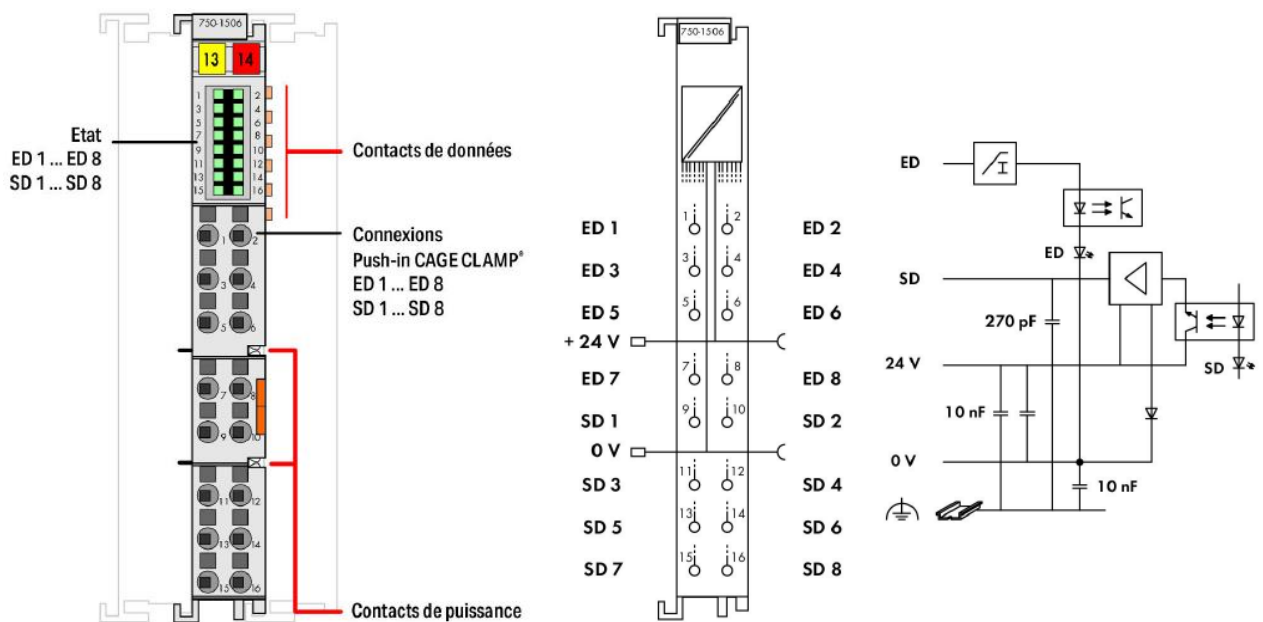
MANIP 5 : Premiers pas avec l'API WAGO

Le datasheet montre pour l'étage de sortie un optocoupleur suivi d'un étage tampon. Pour l'étage d'entrée on trouve également un optocoupleur dont le sens de la LED informe sur le type de commutation. Ci-dessous une illustration partielle de l'étage d'entrée :



Notre module a 8 entrées et 8 sorties. Les entrées se trouvent au niveau de la partie supérieure et les sorties au niveau de la partie inférieure.

Les états des entrées/sorties sont indiqués à l'aide de 16 LEDs.



- **Le module de terminaison**

Lorsqu'on connecte des dispositifs ensemble à l'aide d'un câble, il peut apparaître des effets indésirables comme de la distorsion d'amplitude ou de phase. Des études ont montré qu'il est possible de réduire fortement ces effets indésirables en utilisant des impédances de ligne. C'est le rôle de ce module pour le bus de terrain utilisé sur les connecteurs RJ45.

MANIP 5 : Premiers pas avec l'API WAGO

Introduction à l'environnement de développement

- Installation**

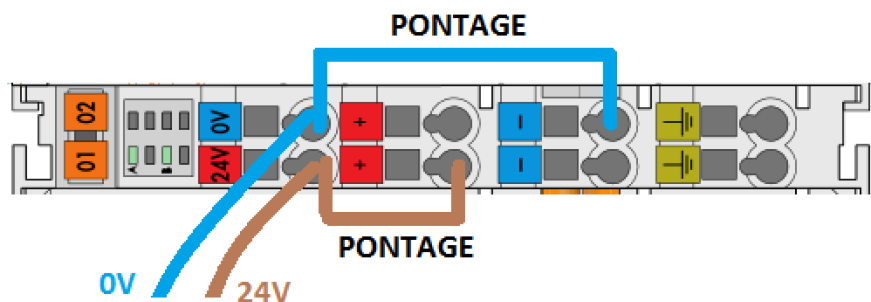
Le constructeur (Wago) fournit une version d'évaluation de eCOCKPIT valable 30 jours. Sa mise en œuvre s'appuie sur l'environnement de développement CODESYS disponible gratuitement.

Nous installerons CODESYS V3 et une version d'évaluation de eCOCKPIT (V1.4 par exemple).

- Configuration matérielle de l'automate**

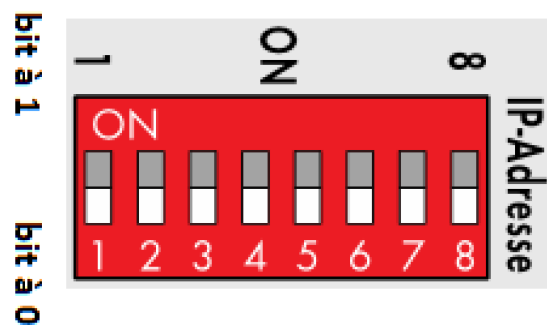
Alimentation des modules :

Comme illustré ci-contre, On doit fournir du 24V_{DC} au module et le rendre disponible pour les autres modules grâce à des pontages.



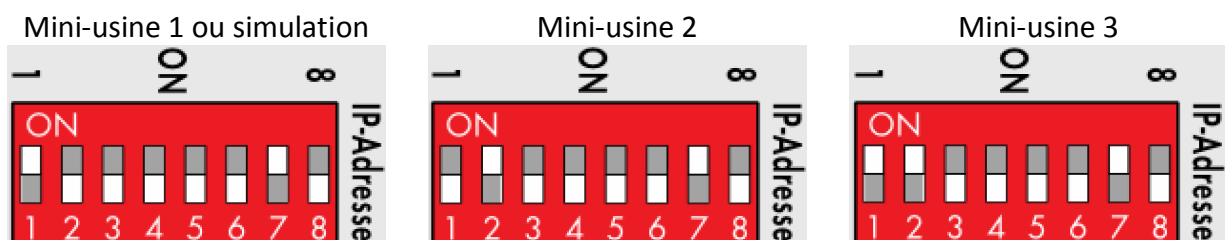
Configuration IP de l'automate :

Comme illustré ci-contre, on peut paramétrer l'adresse IP de l'automate avec des petits commutateurs à glissière. Lorsqu'on est en position ON, le bit correspondant est à '1', sinon il est à '0'. On paramètre en réalité le dernier octet de l'IP où le commutateur 1 correspond au LSB (bit 0) et le commutateur 8 correspond au MSB (bit 7).



À priori, on travaillera toujours au sein d'un réseau local (donc pas sur Internet) qu'il s'agisse d'un réseau « PC↔API », d'un réseau « API↔API » ou d'un réseau plus complexe.

On adoptera la convention que les commutateurs 7 et 8 seront respectivement à 1 et 0, les autres commutateurs seront positionnés suivant l'application comme illustré ci-dessous :



MANIP 5 : Premiers pas avec l'API WAGO

• **Configuration réseau du PC**

Une adresse IPV4 est composée de 32 bits, on l'écrit souvent sous la forme de 4 nombres entiers ou chaque nombre correspond à un octet. Exemple : 192.168.1.207

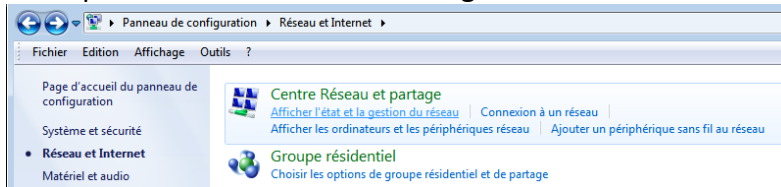
L'adresse IP permet d'identifier deux parties : le réseau et l'hôte au sein de ce réseau. La frontière de séparation entre ces parties n'est pas toujours la même, cependant nous utiliserons 24 bits pour identifier le réseau (notation /24) et les 8 bits restants pour l'hôte. Par exemple, 192.168.1.207 /24 donne le réseau 192.168.1 (noté 192.168.1.0) et l'hôte 207.

Pour pouvoir connecter notre PC à l'automate à l'aide du câble réseau, nous devons nous assurer que l'automate et le PC soient dans le même réseau → les 3 premiers nombres doivent être identiques pour le PC et l'automate. Nous utiliserons 192.168.1.X.
Pour l'adresse de l'hôte, nous choisirons par défaut $(1100\ 0000)_2 = 192$ pour rester compatible avec la convention $(01xx\ xxxx)_2$ adoptée pour le paramétrage IP des automates.

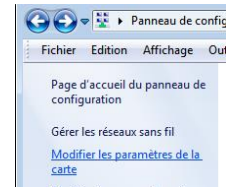
→ Par défaut, on fixera l'adresse IP du PC à **192.168.1.192**

On devra modifier les paramètres de configuration de la carte réseau du PC et préciser l'adresse IP ainsi que le masque de sous-réseau 255.255.255.0 (voir ci-dessous).

1° Cliquer sur Panneau de configuration → Réseau et Internet
2° Cliquer sur « Afficher l'état et la gestion du réseau »

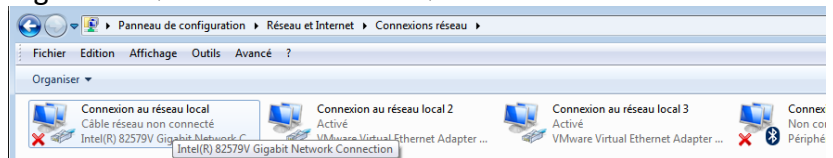


3° Sélectionner « Modifier les paramètres de la carte »

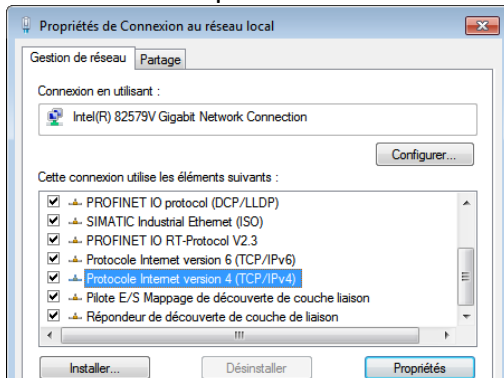


4° On atteint Panneau de configuration → Réseau et Internet → Connexions réseau

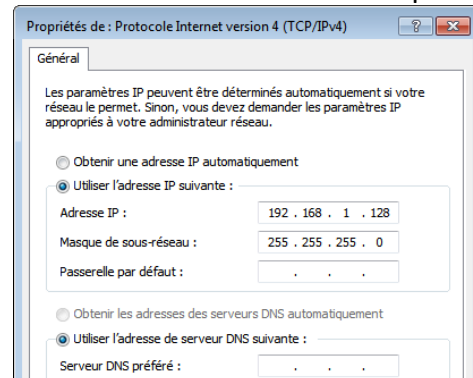
5° On fait un click droit sur l'icône de la carte réseau.



6° Sélectionner Propriétés de IPV4



7° Introduire l'adresse IP et le masque



MANIP 5 : Premiers pas avec l'API WAGO

• **Configuration du projet**

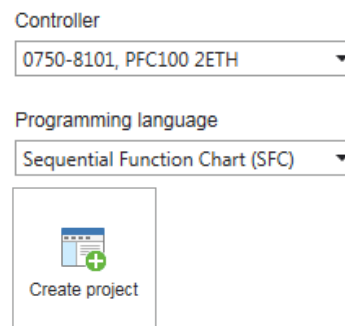
On commence par lancer le logiciel eCOCKPIT et le configurer pour notre matériel et l'adresse IP paramétrée.

Le matériel répond aux articles 750-8101 (PFC100), 750-1506 (E/S), 750-600 (terminaison). Le paramétrage IP correspond à $(0100\ 0001)_2$ **pour une simulation** → l'hôte 65, l'adresse IP de l'automate sera alors **192.168.1.65**.

Au lancement de eCOCKPIT, on choisira de démarrer un projet « Product Series 750 ».

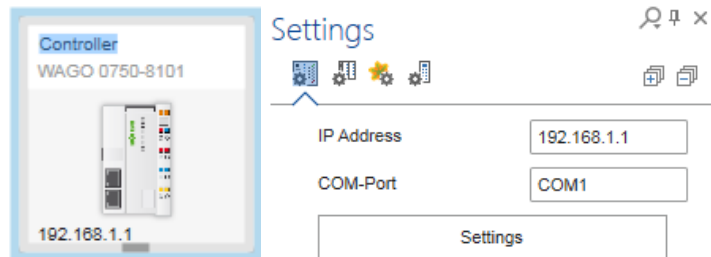
Ensuite, comme illustré ci-contre, on choisira le contrôleur approprié et le langage SFC.

On crée ensuite le projet.



On peut alors voir apparaître la référence du contrôleur et certaines configurations pré remplies.

On peut introduire l'IP 192.168.1.65 manuellement ou lancer un Scan pour vérification.

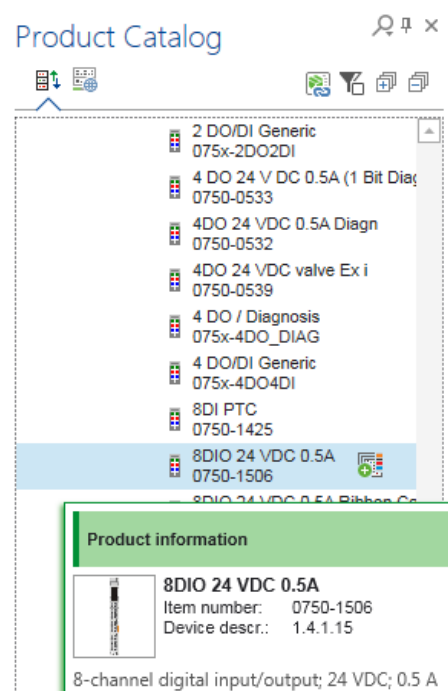


Pour ajouter notre module d'entrée/sortie, on peut sélectionner le menu « View→Product Catalog »

On sélectionne notre contrôleur à la gauche de la fenêtre. Il apparaît alors à droite de l'écran, comme représenté ci-contre, une liste de différents produits liés à notre contrôleur.

On choisit le module d'entrée/sortie approprié, il doit porter le numéro 0750-1506.

La structure des modules se complète au niveau de l'arborescence et sur la représentation visuelle des blocs

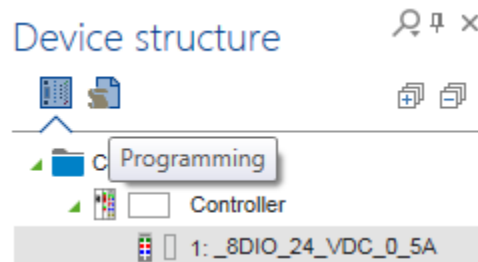


MANIP 5 : Premiers pas avec l'API WAGO

• **Programme de test**

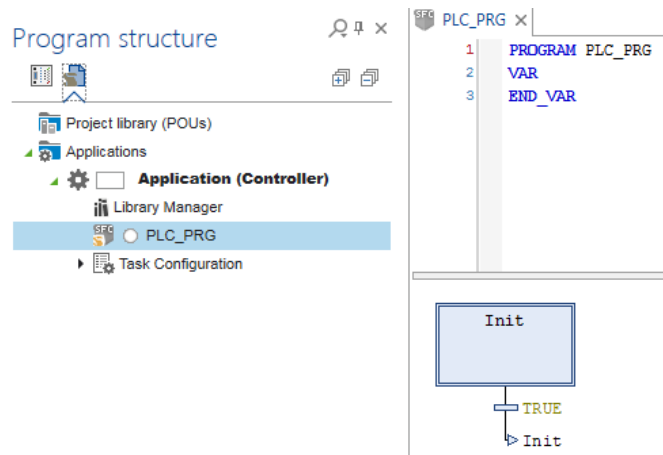
Une fois la configuration faite, sauvez votre projet au sein d'un dossier dédié.

Pour visualiser le programme, on clique sur le 2^{ème} onglet de « Device structure » comme représenté ci-contre.



Il apparaît alors le programme et sa structure.

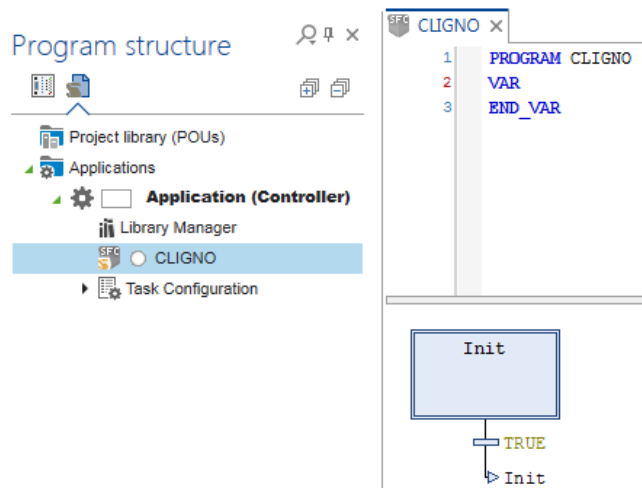
Au sein de « Program structure », le programme proprement dit est nommé PLC_PRG par défaut.



On remarque qu'il contient :

- une partie texte (en haut), on y déclarera les variables.
- une partie SFC (en bas), on l'utilisera pour implémenter notre programme.

Si on souhaite renommer « PLC_PRG » (par exemple en « CLIGNO », comme ci-contre), on peut le faire en commençant par un double-click sur l'élément « PLC_PRG » de l'arborescence.



Une fenêtre apparaît pour nous demander notre accord pour adapter toutes les références liées à ce nom → On accepte.

Une autre fenêtre apparaît pour nous signaler les changements sur le point d'être faits → On accepte.

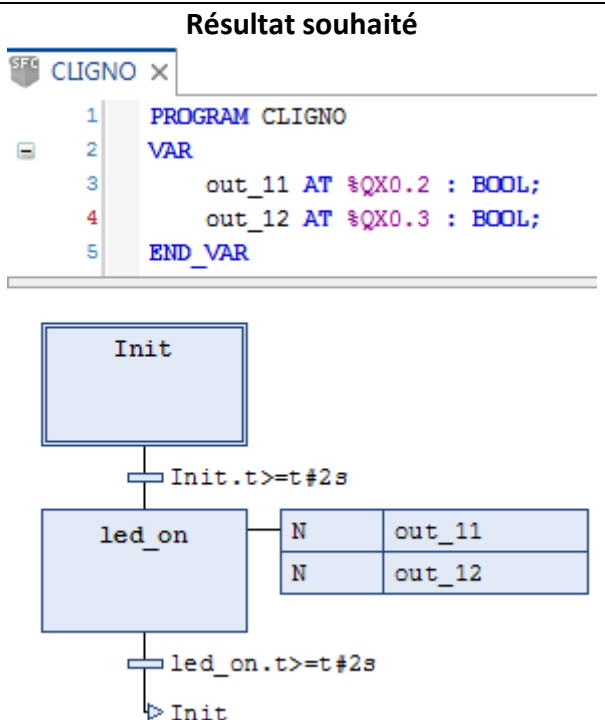
Pour l'édition, on accède aux outils SFC :



MANIP 5 : Premiers pas avec l'API WAGO

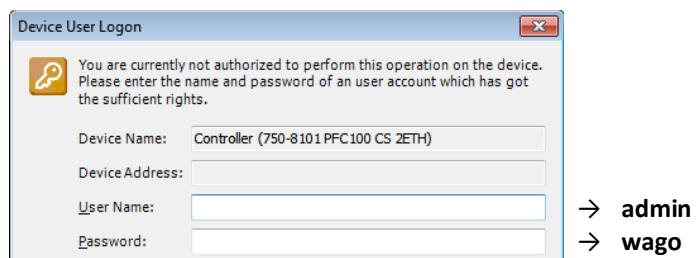
On essayera d'arriver au résultat ci-après. Pour ce faire :

- Insérer une étape + transition
- Renommer la deuxième étape en « led_on »
- Recopier les lignes pour les variables
- Click-droit sur l'étape « led_on » et choisir « Insert action association »
- Recopier les deux actions associées à l'étape « led_on »
- Recopier la syntaxe pour les deux conditions-transitions

Résultat souhaité	Quelques explications
 <p>The screenshot shows the SFC editor for 'CLIGNO'. The code defines two boolean variables: out_11 at %QX0.2 and out_12 at %QX0.3. The ladder logic diagram consists of three states: 'Init', 'led_on', and 'Init'. Transitions are: 'Init' to 'led_on' on condition 'Init.t >= t#2s'; 'led_on' to 'Init' on condition 'led_on.t >= t#2s'. The 'led_on' state has two parallel actions: 'N out_11' and 'N out_12'.</p>	<p>Pour les variables :</p> <ul style="list-style-type: none"> Q correspond à une sortie. X indique qu'il s'agit d'un bit. 0 correspond dans notre cas au port de sortie. .2 et .3 sont les bits 2 et 3 du port. <p>L'étape « Init » n'a pas d'action associée</p> <p>« Init.t » correspond à la durée écoulée depuis la dernière activation de « Init »,</p> <p>« Init.t >= t#2s » est une condition-transition qui deviendra vraie lorsque « Init » aura été active au moins 2 secondes.</p> <p>L'étape « led_on » est associée à deux actions qui seront effectuées en continu tant que l'étape est active.</p> <p>La séquence reboucle sur « Init ».</p>

Lorsque votre programme est prêt, vous pouvez sélectionner l'onglet **PROGRAM** pour utiliser les actions : **Connect, Build** et **Start**

La connexion nécessite les accès (ici à droite les accès par défaut).



Une fois la connexion établie avec l'automate et le projet compilé et chargé dans l'automate, on peut l'exécuter avec Start.

On voit alors apparaître sur le PC l'état d'exécution du programme SFC ainsi que l'état des variables (voir ci-contre).

