

Table des matières

1	Introduction générale	4
1.1	Objectif du manuel	4
1.2	Contenu et progression	4
2	Tutoriel	5
2.1	Rappel : qu'est-ce qu'un GRAFCET ?	5
2.2	Passage du GRAFCET à la logique séquentielle	6
2.3	Traduction en Ladder	6
2.3.1	Initialisation : activation de M0	7
2.3.2	Transition M0 → M1	7
2.3.3	Transition M1 → M2	7
2.3.4	Actions : commandes du vérin	8
2.4	Programme test : pilotage d'un vérin	10
2.5	Résumé pédagogique	10
3	Présentation du matériel UniStream	11
3.1	Vue d'ensemble	11
3.2	Composants principaux	12
3.3	Mise sous tension et diagnostic	12
4	Installation du logiciel UniLogic	13
4.1	Téléchargement	13
4.2	Installation pas à pas	13
4.2.1	Procédure d'installation	13
4.2.2	Vérification de l'installation	14
4.3	Mise à jour de l'automate	14
4.3.1	Méthode A : Mise à jour via clé USB (sans connexion PC)	14
4.3.2	Méthode B : Mise à jour directe via câble USB	15
4.3.3	Recommandations importantes	15
4.3.4	Tableau de compatibilité Firmware/UniLogic	16
4.4	Connexion à l'automate	16
4.4.1	Établir la liaison PC PLC	16
4.4.2	Mode Online — Supervision en temps réel	16
4.4.3	Transfert de projet : Upload et Download	17
4.4.4	Vérification du programme avant transfert	17
4.4.5	Procédure complète de Download	18
4.5	Structure d'un projet UniLogic	18

5	Premiers pas dans UniLogic	19
5.1	Création d'un nouveau projet	19
5.1.1	Lancement de UniLogic	19
5.1.2	Configuration du projet	19
5.1.3	Sélection du modèle d'automate	20
5.2	Création de Tags	21
5.2.1	Types principaux de Tags	21
5.2.2	Accès aux Tags dans UniLogic	21
5.3	Configuration des broches	22
5.3.1	Récapitulatif	23
5.4	Éditeur Ladder	23
5.4.1	Programmation en Ladder	24
5.5	Création d'un écran IHM	25
5.5.1	Éléments graphiques disponibles	25
5.5.2	Association entre variables et objets HMI	26
5.5.3	Accès à l'éditeur HMI	26
5.5.4	Utilisation de la Toolbox HMI	26
5.5.5	Ajout d'éléments sur l'écran	27
5.5.6	Liaison d'un élément HMI à une variable	28
5.5.7	Configuration des textes binaires	28
5.5.8	Ajout d'une action interactive	28
5.6	Gestion des alarmes	29
5.6.1	Accès à la gestion des alarmes	29
5.6.2	Création d'un groupe d'alarmes	29
5.6.3	Ajout d'une alarme	30
5.6.4	Définition du déclencheur de l'alarme	30
5.6.5	Logique d'activation dans le Ladder	30
5.7	Connexion UART	31
5.7.1	Configuration de l'UART côté MSP430	31
5.7.2	Choix de l'UART0 pour la communication PLC	32
5.7.3	Configuration logicielle des UARTs	32
5.7.4	Paramètres de communication UART	32
5.7.5	Communication entre le PLC et la carte MSP430	33
5.7.6	Envoi des messages	33
5.7.7	Réception des messages	34
5.8	Connexion Modbus	35
5.8.1	Prérequis	35
5.8.2	Accès aux paramètres de l'automate	35
5.8.3	Configuration réseau de l'automate	37
5.8.4	Configuration réseau dans UniLogic	38
5.8.5	Configuration du protocole Modbus	39
5.8.6	Téléchargement du projet	40
5.8.7	Test de la connexion Modbus	41
5.8.8	Test depuis l'IHM	42
5.9	Transfert vers l'automate	43

6	Travaux pratiques sur les E/S du UniStream	44
6.1	Exercice 1 – Commande TOR (LED)	44
6.1.1	Objectif	44
6.1.2	Matériel requis	44
6.1.3	Cahier des charges	44
6.1.4	Schéma de câblage	45
6.1.5	Programmation dans UniLogic	46
6.2	Exercice 2 – Réception TOR (entrée physique)	53
6.2.1	Objectif	53
6.2.2	Matériel requis	53
6.2.3	Ports utilisés	53
6.2.4	Cahier des charges	53
6.2.5	Schéma de câblage	53
6.2.6	Programmation dans UniLogic	54
6.3	Exercice 3 – Commande PWM (LED variable)	55
6.4	Exercice 4 – Lecture analogique (potentiomètre)	55
6.4.1	Objectif	55
6.4.2	Matériel requis	55
6.4.3	Ports utilisés	55
6.4.4	Cahier des charges	55
6.4.5	Schéma de câblage	56
6.4.6	Programmation dans UniLogic	56
6.4.7	Test et validation	59
6.5	Exercice 5 – Entrée rapide (encodeur)	59
7	Diagnostic et dépannage	60
7.1	Outil UniApps	60
7.2	Erreurs courantes	60
8	Conclusion	61

Chapitre 1

Introduction générale

1.1 Objectif du manuel

Ce manuel s'adresse aux débutants souhaitant apprendre à utiliser l'automate programmable **Unitronics UniStream** ainsi que son environnement de développement **UniLogic**. Il guide l'utilisateur pas à pas, depuis l'installation du logiciel jusqu'à la réalisation de petits projets pratiques, afin de comprendre le fonctionnement des entrées et sorties numériques, analogiques et rapides.

L'objectif final de ce manuel est d'acquérir les bases de la programmation, du câblage et de la supervision d'un système automatisé complet.

1.2 Contenu et progression

Ce manuel propose une approche progressive, permettant au lecteur d'acquérir les compétences essentielles étape par étape. Chaque chapitre introduit de nouvelles fonctionnalités du logiciel **UniLogic** et de l'automate **UniStream**, tout en s'appuyant sur des exercices pratiques afin de valider immédiatement les connaissances acquises.

Le manuel suit une progression pédagogique :

- Introduction au GRAFCET.
- Installation et configuration du logiciel UniLogic.
- Découverte du matériel UniStream et de ses connectiques.
- Premiers pas dans UniLogic.
- Réalisation d'exercices pratiques : commande TOR, lecture TOR, PWM et encodeur.

Chapitre 2

Tutoriel

2.1 Rappel : qu'est-ce qu'un GRAFCET ?

Le GRAFCET (Graphe Fonctionnel de Commande Étape/Transition) est une méthode normalisée permettant de représenter la logique séquentielle d'un système automatisé.

- Il est composé d'étapes (carrés numérotés) pouvant être actives ou inactives.
- Les transitions permettent le passage d'une étape à une autre lorsque leurs conditions sont vraies.
- À chaque étape sont associées des actions (commandes des sorties).

Exemple pour un vérin double effet :

- **Étape 0** : vérin au repos (piston rentré)
- **Transition 0→1** : bouton *Start* appuyé
- **Étape 1** : sortie du vérin (avance)
- **Transition 1→2** : détection du capteur "Sorti"
- **Étape 2** : rentrée du vérin
- **Transition 2→0** : détection du capteur "Rentré"

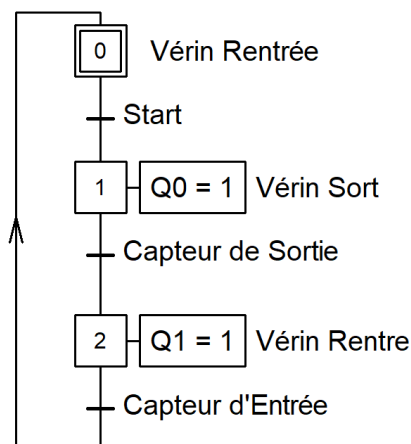


FIGURE 2.1 – Exemple de GRAFCET pour un vérin double effet

2.2 Passage du GRAFCET à la logique séquentielle

Chaque étape du Grafcet correspond à un bit mémoire dans l'automate ($M0, M1, M2, \dots$). Les transitions deviennent des équations booléennes, et les actions sont commandées par les étapes actives.

Correspondance des étapes

- Étape 0 = $M0$ (initialisée à 1)
- Étape 1 = $M1$
- Étape 2 = $M2$

Transitions (logique booléenne)

$$T0 = M0 \cdot Start$$

$$T1 = M1 \cdot Capteur_Sorti$$

$$T2 = M2 \cdot Capteur_Rentr$$

Équations des étapes

$$M1 = (M0 \cdot Start) + (M1 \cdot \neg Capteur_Sorti)$$

$$M2 = (M1 \cdot Capteur_Sorti) + (M2 \cdot \neg Capteur_Rentr)$$

$$M0 = (M2 \cdot Capteur_Rentr) + (M0 \cdot \neg Start)$$

Avec initialisation

$$M1 = (M0 \cdot Start) + (M1 \cdot \neg Capteur_Sorti \cdot \neg Init)$$

$$M2 = (M1 \cdot Capteur_Sorti) + (M2 \cdot \neg Capteur_Rentr \cdot \neg Init)$$

$$M0 = (M2 \cdot Capteur_Rentr + Init) + (M0 \cdot \neg Start)$$

Commandes des sorties

$$Q0 = M1 \quad Q1 = M2$$

2.3 Traduction en Ladder

Hypothèses matérielles :

- Entrées : $I0 = Start, I1 = \text{Fin de course Rentré}, I2 = \text{Fin de course Sorti}$
- Sorties : $Q0 = \text{Avance vérin}, Q1 = \text{Retour vérin}$
- Mémoires : $M0, M1, M2$

2.3.1 Initialisation : activation de M0

$$M0 = M2 \cdot I1 + Init + (M0 \cdot \neg Start)$$

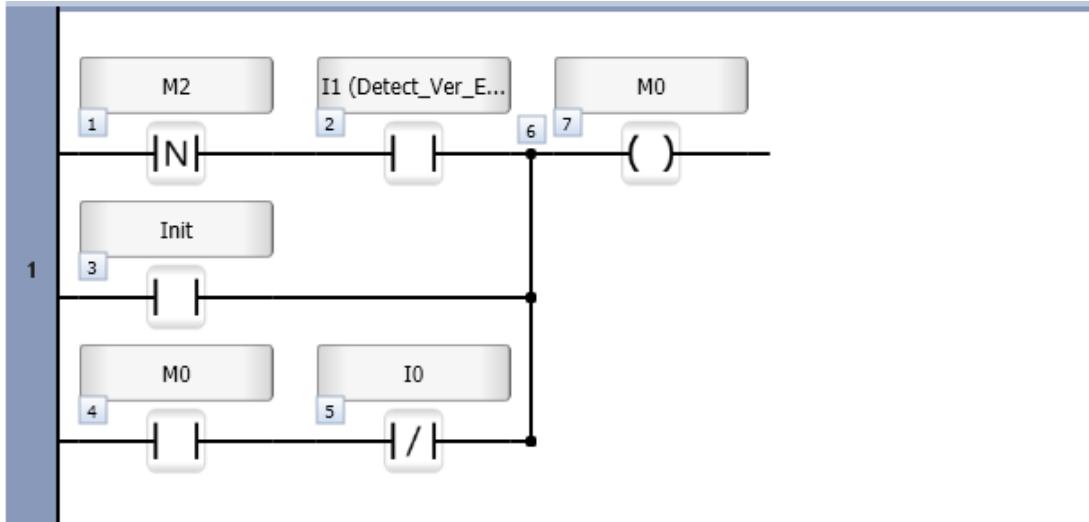


FIGURE 2.2 – Équation Ladder de l'étape M0

2.3.2 Transition M0 → M1

$$M1 = M0 \cdot Start + M1 \cdot \neg I2 \cdot \neg Init$$

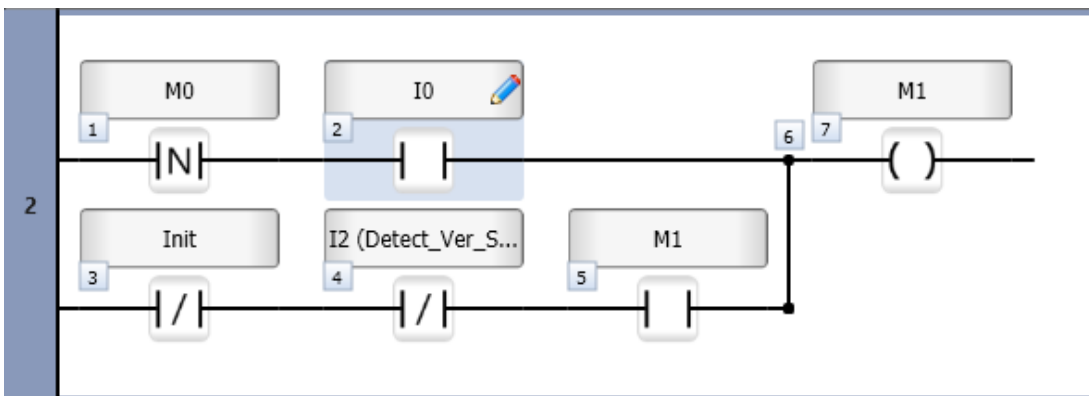


FIGURE 2.3 – Équation Ladder de l'étape M1

2.3.3 Transition M1 → M2

$$M2 = M1 \cdot I2 + M2 \cdot \neg I1 \cdot \neg Init$$

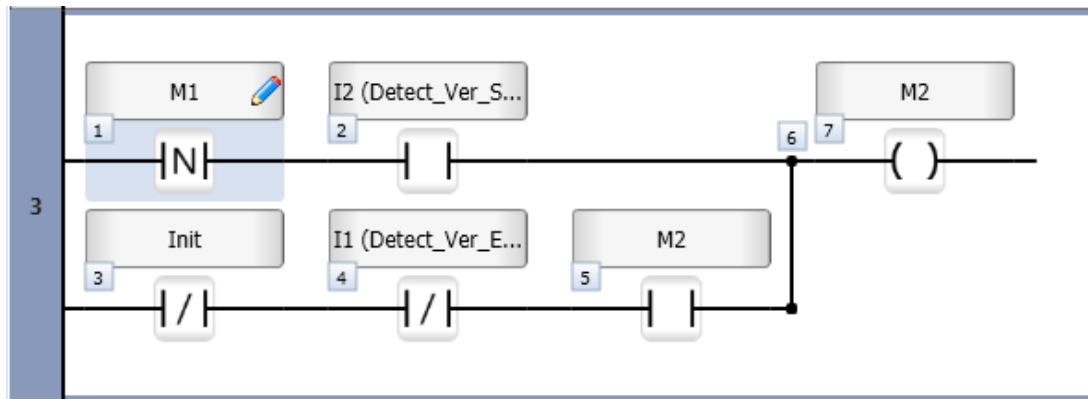


FIGURE 2.4 – Équation Ladder de l'étape M2

2.3.4 Actions : commandes du vérin

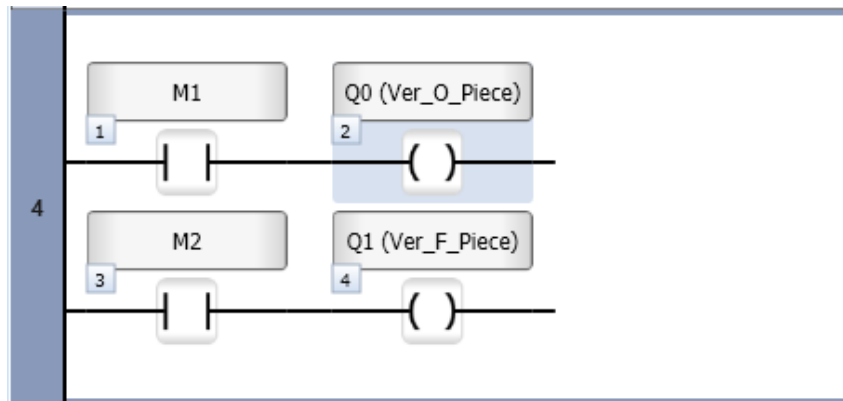


FIGURE 2.5 – Commandes des sorties Q0 et Q1

Fonctionnement du bouton Start

Le bouton utilisé dans l'IHM agit comme un bouton-poussoir : il vaut **1 uniquement pendant l'appui**, et revient à **0 lorsqu'il est relâché**.

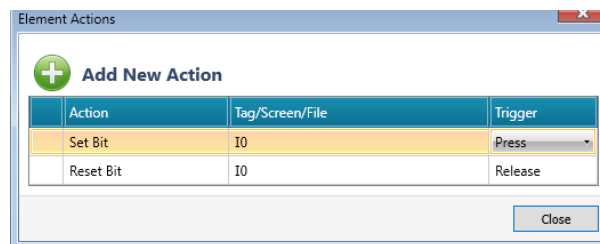


FIGURE 2.6 – Programmation du bouton Start

Remarque importante sur la structure des rungs

Dans notre automate, les transitions se font très rapidement, ce qui peut empêcher le maintien correct de l'état précédent. Pour éviter cela, il est nécessaire d'utiliser les contacts `--[N]--` afin d'obtenir un maintien temporaire (*pulse*).

Structure correcte d'un rung avec une bobine unique

Pour une même bobine (même mémoire ou même sortie), il est recommandé d'utiliser **un seul rung** dans lequel toutes les conditions menant à cette bobine sont regroupées en **branches parallèles**. Ainsi, toutes les conditions pilotent directement la même bobine, ce qui garantit un comportement cohérent et lisible.

Exemple correct (toutes les conditions menant à M0 sont dans un seul rung, en parallèle) :

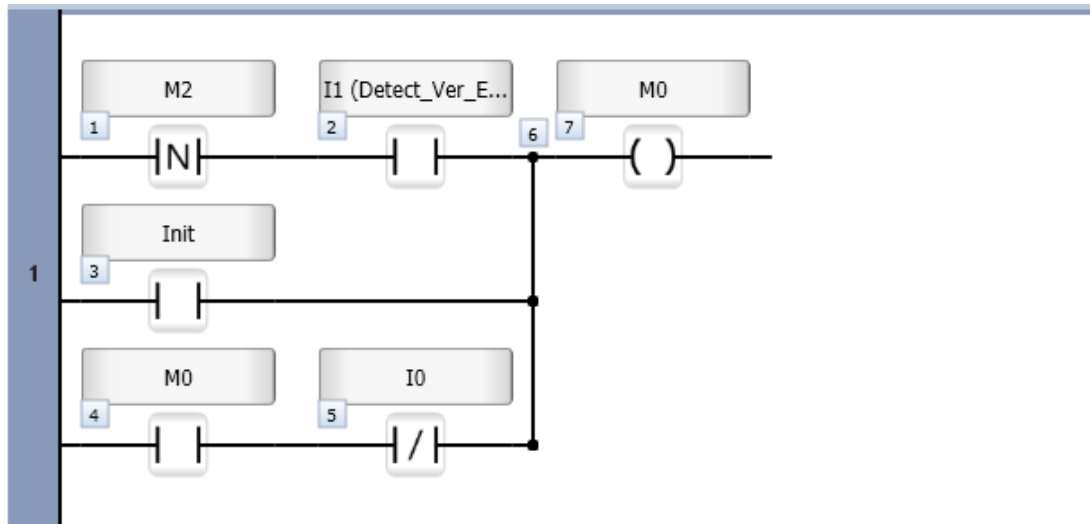


FIGURE 2.7 – Bonne pratique : une seule bobine, un seul rung, conditions en parallèle

Mauvaise pratique à éviter

Il ne faut pas écrire plusieurs rungs séparés pilotant la **même bobine**. Cela crée des comportements imprévisibles, car chaque rung peut activer ou désactiver la bobine indépendamment des autres.

Exemple incorrect (plusieurs rungs différents écrivant tous sur M0) :

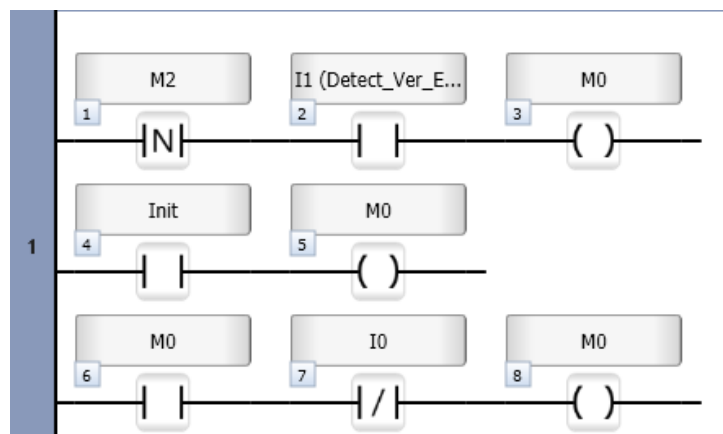


FIGURE 2.8 – Mauvaise pratique à éviter

Pour obtenir un fonctionnement fiable, il faut donc :

- utiliser un **seul rung par bobine** ;

- regrouper toutes les conditions dans des **branches parallèles** ;
- employer les contacts --[N]-- lorsque nécessaire pour assurer le maintien.

Priorité d'initialisation

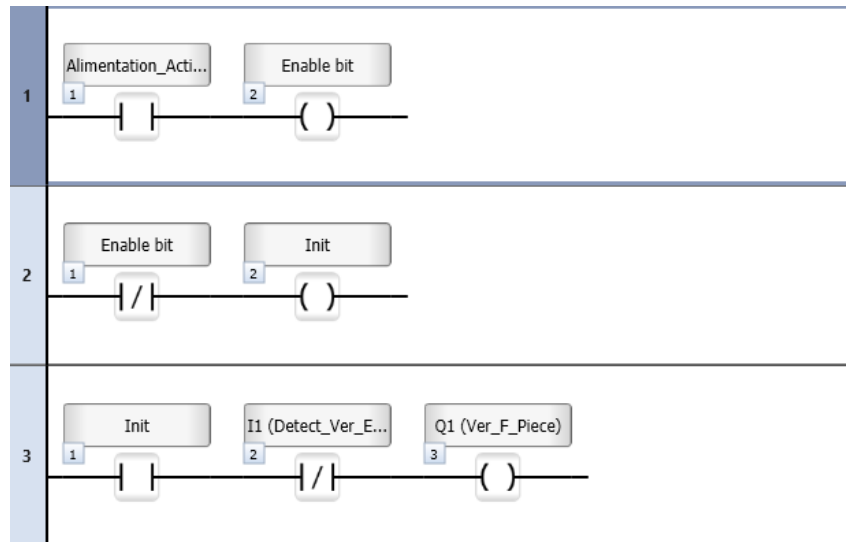


FIGURE 2.9 – Gestion de la priorité d'initialisation

2.4 Programme test : pilotage d'un vérin

Séquence :

1. Au départ : M0 = 1, vérin rentré.
2. Appui sur Start : activation de M1 → sortie Q0 active (avance).
3. Lorsque le capteur "Sorti" passe à 1 : activation de M2 → le vérin rentre.
4. Détection du capteur "Rentré" : retour à M0.

2.5 Résumé pédagogique

- Étape 1 : tracer le Grafcet.
- Étape 2 : associer chaque étape à une mémoire interne.
- Étape 3 : écrire les équations booléennes.
- Étape 4 : traduire en Ladder.
- Étape 5 : tester sur l'automate ou simulateur.

Chapitre 3

Présentation du matériel UniStream

3.1 Vue d'ensemble

L'automate **Unitronics UniStream US5-B10-T42** est un contrôleur tout-en-un regroupant dans un seul équipement :

- un automate programmable (PLC),
- une interface homme-machine tactile (IHM),
- des entrées/sorties locales intégrées.

Ce type d'automate est particulièrement adapté aux applications de contrôle-commande compactes, nécessitant à la fois une logique de traitement et une interface utilisateur pour l'exploitation du système.

L'UniStream US5-B10-T42 permet de réaliser des projets industriels et pédagogiques variés, incluant la commande de moteurs, la lecture de capteurs, la supervision de processus ainsi que des communications réseau. Grâce au logiciel **UniLogic**, la configuration matérielle, la programmation et la création d'écrans IHM sont centralisées dans une seule plateforme de développement conviviale.



UNISTREAM® Built in - Hardware



FIGURE 3.1 – Vue générale de l'automate UniStream US5-B10-T42

3.2 Composants principaux

L'automate **UniStream US5-B10-T42** est un système intégré combinant un automate, une interface homme-machine (IHM) et des entrées/sorties locales. Il permet de concevoir des applications industrielles complètes sans modules externes.

Ses principales caractéristiques sont :

- Écran tactile 5" (IHM intégrée)
- 24 entrées digitales (24V PNP/NPN)
- 16 sorties transistors PNP
- 2 entrées analogiques 0–10V
- 2 sorties PWM (jusqu'à 3 kHz)
- 2 entrées rapides (pour codeur)
- Ports USB, Ethernet, alimentation 24V DC

3.3 Mise sous tension et diagnostic

Mise sous tension de l'automate

Pour mettre sous tension l'automate UniStream US5-B10-T42, procéder comme suit :

- Connecter l'alimentation 24V DC (borne +24V et 0V)
- Vérifier que le voyant **RUN** / **STOP** indique l'état de l'automate
- Accéder au menu **UniApps** pour effectuer le diagnostic et tester les entrées/sorties

Chapitre 4

Installation du logiciel UniLogic

4.1 Téléchargement

Pour obtenir le logiciel UniLogic, procéder comme suit :

1. Se rendre sur le site officiel de Unitronics : <https://unitronicsplc.com/software-unilogic/>
 2. Télécharger la dernière version compatible avec Windows
 3. Enregistrer le fichier d'installation (format `.exe`) dans un répertoire accessible
- **Taille du fichier** : environ 500 Mo
 - **Systèmes supportés** : Windows 10, Windows 11 (64 bits)
 - **Prérequis** : .NET Framework 4.7 ou supérieur

Remarque : Il est recommandé de toujours utiliser la version la plus récente pour bénéficier des dernières fonctionnalités et corrections de bugs.

4.2 Installation pas à pas

L'installation de UniLogic nécessite des privilèges administrateur pour garantir l'accès complet aux ressources système et l'installation correcte des pilotes de communication.

4.2.1 Procédure d'installation

1. Lancement de l'installateur

- Localiser le fichier d'installation téléchargé (ex. : `UniLogic_Setup_v1.33.exe`)
- Clic droit sur le fichier → **Exécuter en tant qu'administrateur**
- Confirmer l'autorisation UAC (User Account Control) si demandée

2. Configuration de l'installation

- Accepter les termes du contrat de licence
- Choisir le répertoire d'installation (par défaut : `C:\Program Files\Unitronics\UniLogic`)
- Sélectionner les composants à installer (recommandé : installation complète)

3. Installation des drivers USB

- Le programme d'installation installe automatiquement les pilotes USB nécessaires à la communication avec l'automate

- Ces pilotes permettent la reconnaissance du PLC UniStream lors de la connexion via câble USB
- En cas d'échec, les pilotes peuvent être installés manuellement depuis le dossier :
C:\Program Files\Unitronics\UniLogic\Drivers

4. Finalisation

- Cliquer sur **Install** pour démarrer l'installation
- Patienter pendant la copie des fichiers (durée : 2 à 5 minutes)
- Cliquer sur **Finish** une fois l'installation terminée

4.2.2 Vérification de l'installation

Après l'installation, vérifier que :

- L'icône UniLogic est présente sur le bureau ou dans le menu Démarrer
- Le logiciel se lance sans erreur
- Les pilotes USB sont correctement installés (vérifiable dans **Gestionnaire de périphériques** → **Ports (COM et LPT)**)

4.3 Mise à jour de l'automate

La mise à jour du firmware de l'automate permet d'assurer une compatibilité optimale avec la version d'UniLogic utilisée, de corriger d'éventuels dysfonctionnements et d'intégrer de nouvelles fonctionnalités. Cette section présente deux méthodes pour mettre à jour le firmware de l'UniStream US5-B10-T42.

4.3.1 Méthode A : Mise à jour via clé USB (sans connexion PC)

Cette méthode est recommandée lorsque l'automate est installé en production et que la connexion directe au PC n'est pas accessible.

Préparation de la clé USB

1. Lancer le logiciel **UniLogic** sur le PC
2. Connecter l'automate au PC via câble USB (communication PC-PLC)
3. Insérer une clé USB ou carte mémoire formatée en **FAT32** dans le PC
4. Dans UniLogic, naviguer dans le menu :

PLC → UniStream Management → Remote Update

5. Sélectionner la version de firmware à installer
6. Choisir la clé USB/carte mémoire comme support d'exportation
7. Lancer le téléchargement des fichiers de mise à jour sur la clé USB
8. Une fois le transfert terminé, retirer la clé USB du PC en toute sécurité

Installation sur l'automate

1. Insérer la clé USB dans l'emplacement prévu sur l'automate UniStream
2. Sur l'écran tactile de l'UniStream, **appuyer plusieurs secondes** en haut à droite de l'écran
3. Ouvrir l'application **UniApps**
4. Naviguer dans le menu :

System → Upgrades → Upgrade PLC
5. Valider l'opération et patienter pendant le processus de mise à jour
6. L'automate redémarre automatiquement avec la nouvelle version du firmware

Vérification post-installation

1. Vérifier la version installée via :

UniApps → System → Information
2. Recharger votre projet depuis UniLogic (Download All) pour garantir la compatibilité entre le firmware et l'application

4.3.2 Méthode B : Mise à jour directe via câble USB

Cette méthode est plus rapide et recommandée lors de la phase de développement ou lorsque l'automate est facilement accessible.

1. Faire les mêmes étapes de la méthode A - Préparation de la clé USB
2. Insérer la clé USB dans l'emplacement prévu sur l'automate UniStream
3. Laisser connecter l'automate au PC via câble USB
4. Lancer **UniLogic**
5. Ouvrir votre projet ou créer un nouveau projet
6. Dans UniLogic, aller dans le menu :

PLC → Update PLC Firmware

7. Sélectionner la version de firmware souhaitée dans la liste proposée
8. Cliquer sur **Start Update**
9. Une barre de progression affiche l'avancement du téléchargement
10. **Ne pas déconnecter** le câble USB pendant l'opération
11. L'automate redémarre automatiquement à la fin de la mise à jour
12. Vérifier la version installée et recharger le projet si nécessaire

4.3.3 Recommandations importantes

Ne jamais couper l'alimentation de l'automate pendant la mise à jour
Toujours **sauvegarder le projet** avant toute mise à jour du firmware
Utiliser une clé USB/carte mémoire formatée en **FAT32** et en bon état
Déconnecter tout équipement externe non indispensable durant l'opération
Vérifier la compatibilité entre la version UniLogic et le firmware de l'automate
Prévoir une interruption de service de 3 à 5 minutes

4.3.4 Tableau de compatibilité Firmware/UniLogic

Version UniLogic	Firmware minimum	Firmware recommandé
1.33.x	1.33.90	1.33.120
1.32.x	1.32.80	1.32.100
1.31.x	1.31.70	1.31.95

TABLE 4.1 – Compatibilité entre versions UniLogic et firmware UniStream

Note : Consultez la documentation officielle Unitronics pour connaître les compatibilités exactes selon votre version.

4.4 Connexion à l'automate

Pour transférer un programme ou superviser l'état du système en temps réel, il est nécessaire d'établir une communication entre le PC et l'automate UniStream. Cette opération s'effectue directement depuis le logiciel UniLogic.

4.4.1 Établir la liaison PC PLC

1. Connexion physique

- Brancher l'automate au PC via câble **USB** ou **Ethernet**
- S'assurer que l'automate est sous tension (24V DC)
- Attendre quelques secondes que le système d'exploitation détecte le périphérique

2. Configuration de la communication dans UniLogic

- Ouvrir UniLogic
- Cliquer sur le bouton **PC-PLC Communication** (barre d'outils supérieure)
- Sélectionner le type de connexion souhaité :
 - **USB** : connexion directe, détection automatique
 - **Ethernet** : nécessite la configuration préalable de l'adresse IP

3. Passage en mode Online

- Une fois l'automate détecté dans la liste des périphériques, cliquer sur **Online**
- La barre d'état en haut de UniLogic affiche alors le message **Online** (en vert)
- La communication est établie et opérationnelle

4.4.2 Mode Online — Supervision en temps réel

Une fois la liaison établie en mode **Online**, les fonctionnalités suivantes deviennent accessibles :

- **Visualisation des E/S** : Les états des entrées et sorties (digitales, analogiques) sont affichés en temps réel dans l'éditeur Ladder et sur l'interface HMI
- **Surveillance des Tags** : Les valeurs de toutes les variables (Tags) peuvent être observées et leur évolution suivie pendant l'exécution du programme

- **Débogage dynamique** : Permet d'identifier les dysfonctionnements en observant le comportement du programme sans l'interrompre
- **Modification de valeurs** : Possibilité de forcer temporairement certaines variables pour effectuer des tests

Note importante : Le mode **Online** permet uniquement l'**observation** du programme en cours d'exécution. Il ne modifie pas le code présent dans l'automate, sauf si des valeurs sont forcées manuellement.

4.4.3 Transfert de projet : Upload et Download

Le transfert de programmes entre le PC et l'automate s'effectue selon deux opérations distinctes :

Fonction	Action	Utilisation
Upload	Récupère le projet présent dans l'automate vers le PC	<ul style="list-style-type: none"> — Sauvegarde du programme depuis l'automate — Récupération en cas de perte du fichier source sur le PC — Archivage de versions déployées
Download	Envoie un nouveau projet du PC vers l'automate	<ul style="list-style-type: none"> — Programmation initiale de l'automate — Mise à jour de l'application — Correction de bugs

TABLE 4.2 – Opérations de transfert entre PC et automate

Attention : Lors d'un **Download**, le programme précédemment présent dans l'automate est **remplacé intégralement**. Il est recommandé de sauvegarder le projet existant via **Upload** avant toute modification.

4.4.4 Vérification du programme avant transfert

Avant d'effectuer un **Download** vers l'automate, il est impératif de vérifier l'intégrité du projet :

1. Cliquer sur le bouton **Verify** (icône de validation, barre d'outils)
2. Le compilateur analyse l'ensemble du projet et détecte :
 - Les erreurs de syntaxe dans le code Ladder
 - Les incohérences de configuration (E/S non définies, conflits d'adressage)
 - Les Tags non initialisés ou inutilisés
 - Les écrans HMI mal configurés
3. Si des erreurs sont détectées, elles s'affichent dans la fenêtre **Output** avec leur emplacement

4. Corriger toutes les erreurs avant de procéder au téléchargement
5. Une fois la vérification réussie, le message **Build succeeded** apparaît

Bonne pratique

Toujours compiler et vérifier le projet avant de télécharger sur le PLC.

Cela permet d'éviter :

- Les arrêts de production dus à des erreurs de programmation
- Les comportements imprévisibles de l'automate
- Les pertes de temps liées au débogage sur site

4.4.5 Procédure complète de Download

1. Vérifier le projet : **Build** → **Verify** (ou F7)
2. Établir la connexion : **PC-PLC Communication** → **Online**
3. Lancer le téléchargement : **PLC** → **Download All**
4. Confirmer l'opération dans la boîte de dialogue
5. Patienter pendant le transfert (durée : 10 secondes à 2 minutes selon la taille du projet)
6. L'automate redémarre automatiquement avec le nouveau programme
7. Vérifier le bon fonctionnement en mode **Online**

4.5 Structure d'un projet UniLogic

Chaque projet UniLogic est organisé autour de cinq composants principaux :

- **Tags** : variables internes et externes
- **Logique Ladder** : programme de contrôle
- **Configuration matérielle** : paramétrage des E/S
- **Écrans HMI** : interface homme-machine
- **Paramètres de communication** : réseaux et protocoles

Chapitre 5

Premiers pas dans UniLogic

5.1 Création d'un nouveau projet

5.1.1 Lancement de UniLogic

Lors du lancement de UniLogic, l'interface d'accueil se présente comme illustré à la figure 5.1. Pour débiter un nouveau projet, cliquer sur **New**.

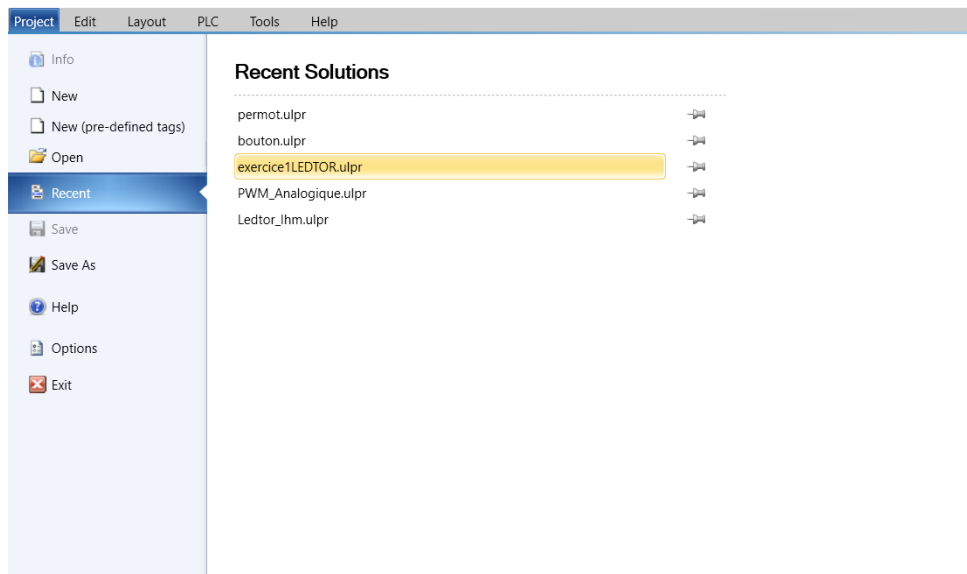


FIGURE 5.1 – Écran d'accueil de UniLogic

5.1.2 Configuration du projet

Remplir le nom du projet, choisir son emplacement et cliquer sur **Next** dans la fenêtre qui s'ouvre (figure 5.2).

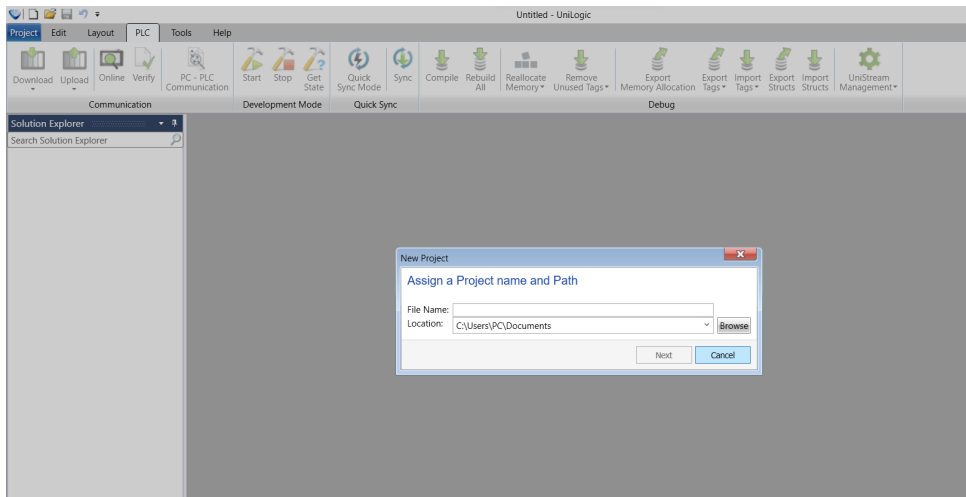


FIGURE 5.2 – Fenêtre de création d'un nouveau projet

5.1.3 Sélection du modèle d'automate

Sélectionner la famille, la gamme et la référence de son automate (figure 5.3). La référence est marquée au dos de l'automate.

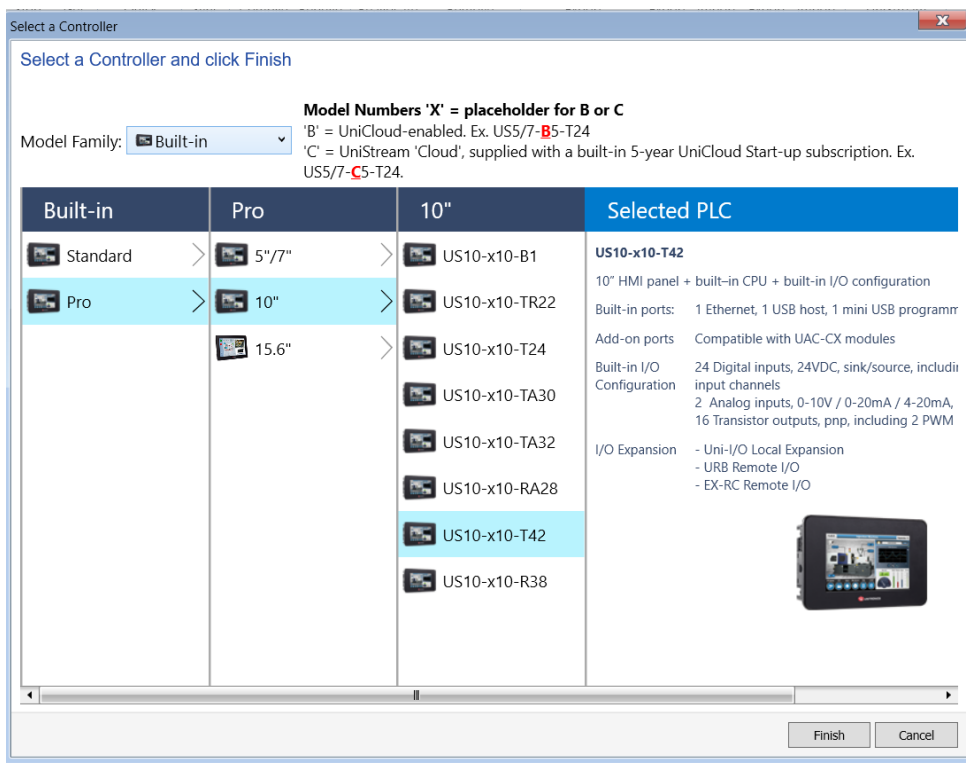


FIGURE 5.3 – Sélection du modèle d'automate

Une fois cette étape terminée, on va créer les variables qui nous seront utiles pour notre projet.

5.2 Création de Tags

Les Tags représentent les variables utilisées dans le programme.

5.2.1 Types principaux de Tags

Les types de données disponibles sont :

- **BOOL** : variable booléenne (0 ou 1)
- **INT16** : entier 16 bits signé (-32768 à +32767)
- **REAL** : nombre réel (virgule flottante)

Les Tags peuvent être liés à des entrées/sorties physiques ou utilisés en interne pour le traitement de données.

5.2.2 Accès aux Tags dans UniLogic

Tous les éléments liés aux variables sont accessibles via la barre horizontale. La section **Global** permet de déclarer les Tags (variables). Pour accéder à cette interface :

- Cliquer sur **Global**
- Tirer la fenêtre vers le haut pour l'agrandir

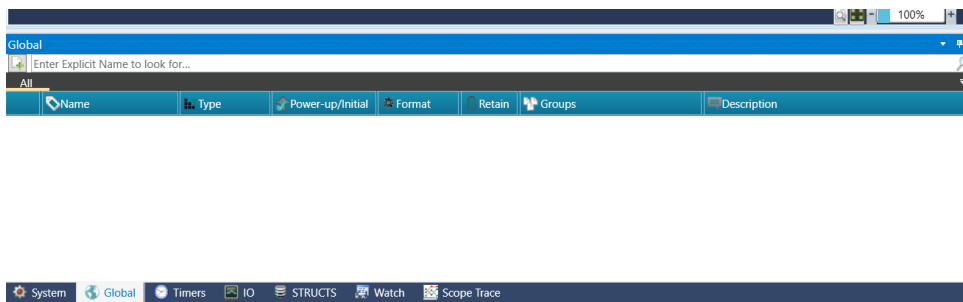


FIGURE 5.4 – Interface de gestion des Tags globaux

Pour créer un nouveau Tag (variable) :

- Cliquer sur l'icône de création de Tag
- Ou faire un clic droit dans l'espace blanc et sélectionner **Add Tag**

Dans la fenêtre qui s'affiche, on a la possibilité de configurer :

- **Type** : type de la variable (**BOOL**, **INT16**, **REAL**, etc.)
- **Format** : mode d'affichage (décimal, binaire ou hexadécimal)
- **Array** : permet de définir si la variable est un tableau
 - Si **Array** est coché, renseigner la taille du tableau

Une fois ces attributs définis :

- Cliquer sur **Save** pour enregistrer le Tag
- Ou cliquer sur **Add** pour ajouter un nouveau Tag

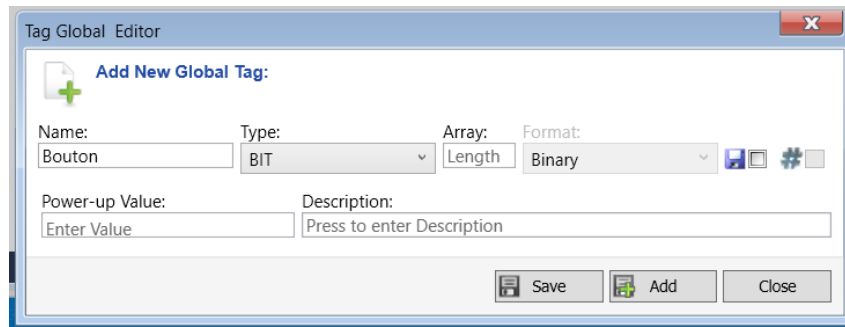


FIGURE 5.5 – Fenêtre de configuration d'un Tag

Une fois cette opération effectuée, le Tag (variable) s'affiche dans la liste et est prêt à être utilisé dans le Ladder, l'IHM ou les deux simultanément.

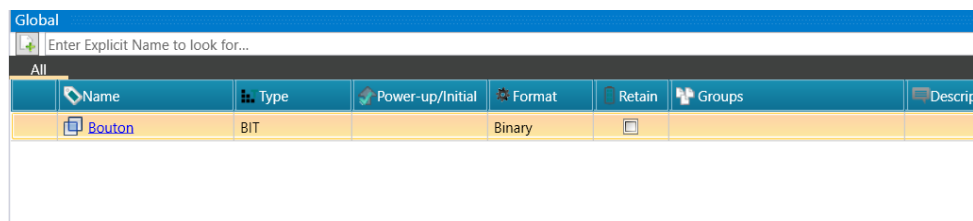


FIGURE 5.6 – Liste des Tags créés

5.3 Configuration des broches

Une fois les Tags créés, on va maintenant configurer les broches nécessaires pour notre application.

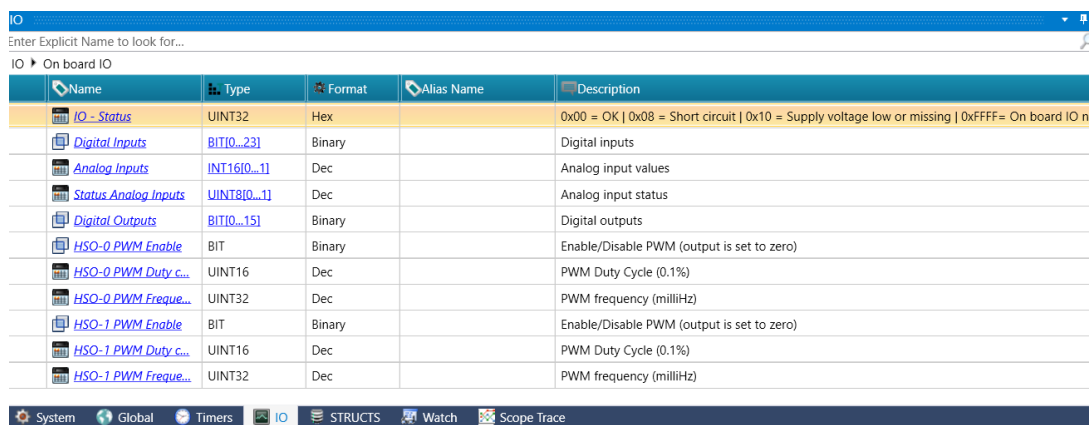


FIGURE 5.7 – Interface de configuration des E/S (On Board IO)

En cliquant sur IO, on se retrouve sur cette fenêtre qui présente la liste des entrées et sorties intégrées (*On Board IO*) de l'automate UniStream, telles qu'elles apparaissent dans le logiciel UniLogic.

On y distingue :

- **Digital Inputs** : entrées digitales regroupées sur 24 voies

- **Digital Outputs** : sorties digitales sur 16 voies, utilisées pour la commande tout ou rien
- **Analog Inputs** : entrées analogiques disponibles sur deux canaux, permettant de lire des signaux continus
- **Status Analog Inputs** : statuts des entrées analogiques servant au diagnostic
- **HSO-0 et HSO-1** : deux sorties rapides PWM, chacune disposant de paramètres d'activation, de fréquence et de rapport cyclique

L'ensemble de ces signaux est accessible sous forme de variables système, directement exploitables dans le programme Ladder et l'IHM. Pour pouvoir spécifier la broche dont on a besoin, il faut d'abord déterminer s'il s'agit d'une entrée ou d'une sortie. Une fois ce choix effectué, on sélectionne la catégorie correspondante dans la colonne **Type**.

Dans le cas présenté sur la figure 5.7, une sortie digitale a été choisie. Il est alors possible de cliquer sur le champ **Type** afin de sélectionner précisément la broche à utiliser.

Il est important de noter que la colonne **Type** indique également la nature des signaux supportés par chaque broche :

- Les entrées et sorties digitales sont de type **BIT**
- Les entrées analogiques sont représentées par des valeurs numériques entières sur 16 bits (**INT16**)

#	Name	Type	Format	Alias Name	Description
0	Digital Outputs 0	BIT	Binary	LED	
1	Digital Outputs 1	BIT	Binary		
2	Digital Outputs 2	BIT	Binary		
3	Digital Outputs 3	BIT	Binary		
4	Digital Outputs 4	BIT	Binary		
5	Digital Outputs 5	BIT	Binary		
6	Digital Outputs 6	BIT	Binary		

FIGURE 5.8 – Attribution d'un alias à une broche

Une fois la broche choisie, on lui donne un **Alias** (nom). C'est par ce nom qu'on pourra agir sur la broche dans le Ladder et dans L'IHM.

5.3.1 Récapitulatif

On a vu comment créer des Tags et comment configurer les broches. Ceci fait, on va voir maintenant comment agir sur ces variables et nos broches configurées.

5.4 Éditeur Ladder

Dans UniLogic, la partie Ladder est accessible dans la barre **Solution Explorer**. Elle est marquée **Ladder**.

Pour avoir accès à la page d'édition du Ladder :

- Cliquer sur **Function1** dans **Module1** si nous n'avons qu'une fonction et un module
- Sinon, il faudra sélectionner la bonne fonction et le bon module
- Pour un début, c'est toujours une fonction et un module



FIGURE 5.9 – Accès à l'éditeur Ladder

Ici, on a la fenêtre d'édition au centre : c'est là que nous éditons notre code Ladder. À gauche, dans la fenêtre **Toolbox**, on a :

- **Basic Element** : tous les éléments de base du Ladder (contacts, bobines, etc.)
- **Blocs fonction** : blocs prédéfinis que nous pouvons utiliser en renseignant simplement les variables d'entrée et de sortie



FIGURE 5.10 – Fenêtre d'édition Ladder et Toolbox

L'utilisation est très pratique et intuitive. Nous ne pouvons pas tout détailler ici. Je vous invite à parcourir les blocs fonction et essayer d'analyser leur fonctionnement.

5.4.1 Programmation en Ladder

Pour programmer, il suffit de :

1. Porter et glisser l'élément dont on a besoin depuis la Toolbox
2. Le coller à la suite de l'élément précédent
3. La connexion se fait automatiquement

Une fois les éléments déposés, il faut attribuer chaque élément à un Tag, une broche ou une variable système.

Dans notre cas :

- Au premier élément, on a attribué le Tag que nous avons créé
- Au deuxième élément, on a rattaché la broche que nous avons paramétrée précédemment

Le principe est le même pour les blocs fonction.

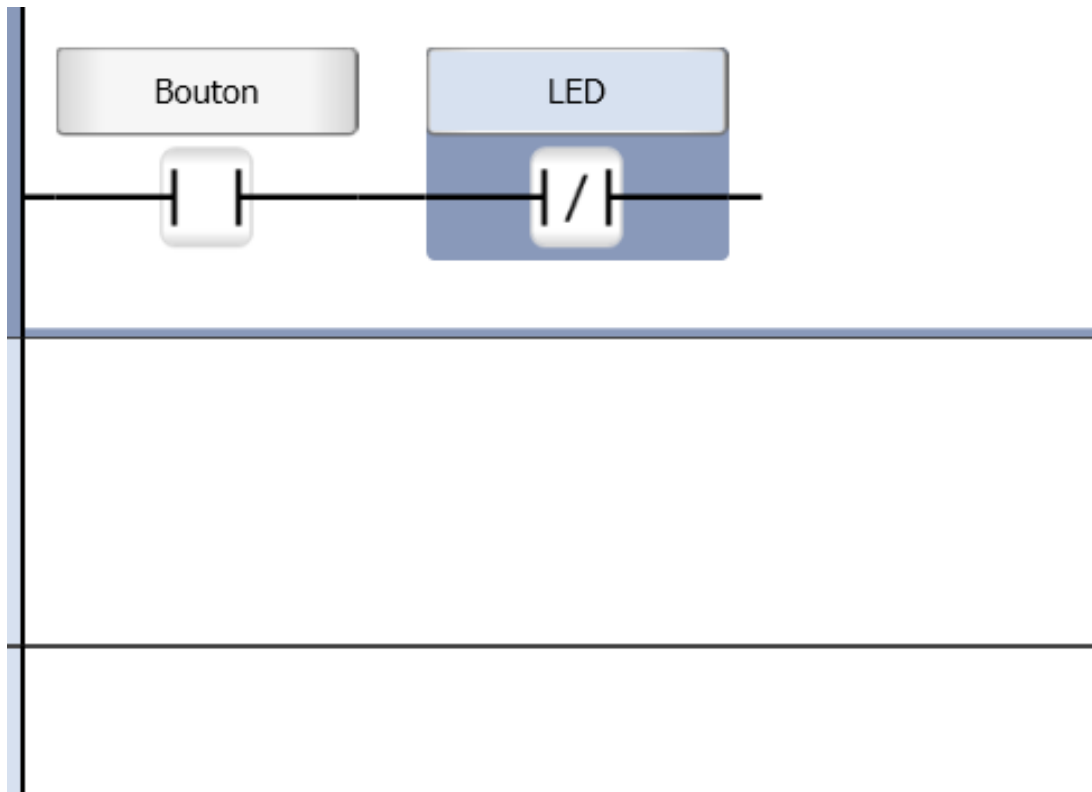


FIGURE 5.11 – Exemple d’attribution de variables aux éléments Ladder

Note : Ce programme Ladder n’a aucun sens. Il est uniquement utilisé pour montrer comment attribuer les variables ou les broches aux éléments du Ladder.

5.5 Création d’un écran IHM

L’interface homme-machine (HMI) permet de créer des écrans de supervision et de commande interactifs affichés sur l’écran tactile de l’automate.

5.5.1 Éléments graphiques disponibles

Il est possible d’ajouter différents types d’objets graphiques :

- **Boutons** : pour les commandes manuelles (démarrage, arrêt, acquittement)
- **Voyants** : pour la visualisation d’états logiques (marche, arrêt, alarme)
- **Jauges** : pour l’affichage de valeurs analogiques sous forme graphique
- **Curseurs** : pour le réglage de consignes ou de paramètres

5.5.2 Association entre variables et objets HMI

Chaque objet HMI peut être lié directement à une variable (Tag) ou une broche. Cette liaison permet :

- L’affichage en temps réel des valeurs
- La modification des variables depuis l’écran tactile
- Une interaction dynamique entre le programme Ladder et l’interface utilisateur

5.5.3 Accès à l’éditeur HMI

L’édition de l’IHM se fait dans l’onglet HMI de la barre **Solution Explorer**. Une fois dans cet onglet, on se retrouve avec un écran vierge prêt à être personnalisé.

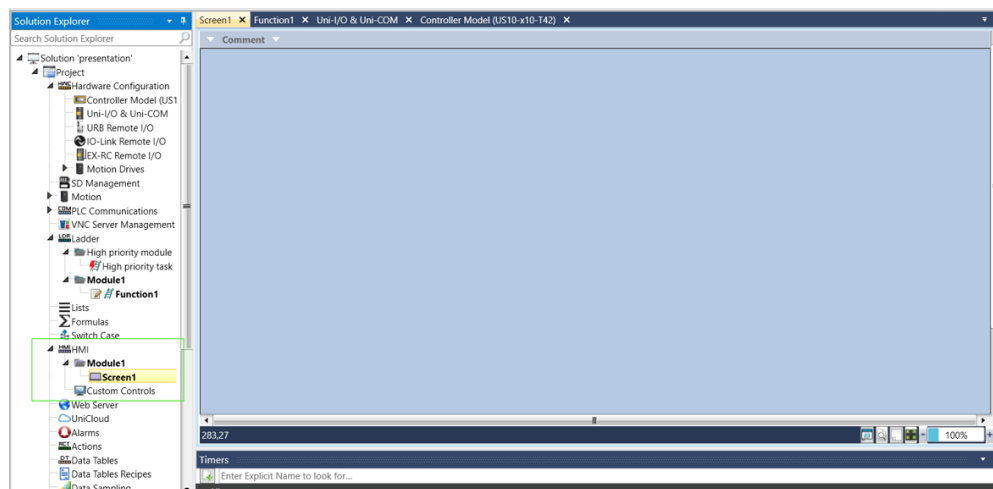


FIGURE 5.12 – Écran vierge de l’éditeur HMI

À gauche se trouve la **Toolbox** qui regroupe toutes les familles d’éléments nécessaires à la conception de l’IHM.

5.5.4 Utilisation de la Toolbox HMI

La Toolbox contient différentes catégories d’éléments graphiques :

- **Simple Elements** : éléments de base
- **Buttons** : boutons de commande
- **User controls** : contrôles utilisateur personnalisés
- **Text Elements** : éléments textuels
- **Numeric elements** : afficheurs et saisies numériques
- **Image Elements** : images et graphiques
- **Tanks** : représentations de cuves et réservoirs
- **Graphs/Meters** : graphiques et jauges
- **Time and Date** : affichage date et heure
- **Timer** : temporisateurs visuels
- **Data Tables** : tableaux de données
- **Schedule** : planificateurs

5.5.5 Ajout d'éléments sur l'écran

Pour ajouter un élément à l'IHM :

1. Sélectionner l'élément souhaité dans la Toolbox
2. Porter et glisser l'élément sur la fenêtre de travail
3. Positionner l'élément à l'emplacement désiré
4. Configurer ses propriétés (taille, couleur, variable liée)

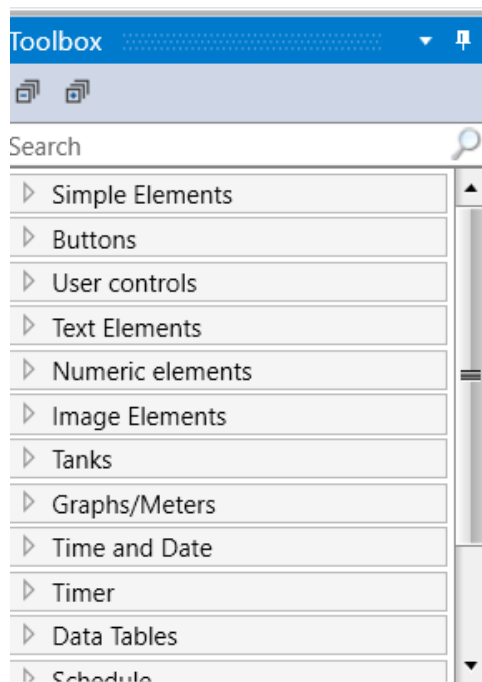


FIGURE 5.13 – Toolbox HMI avec les différentes catégories d'éléments

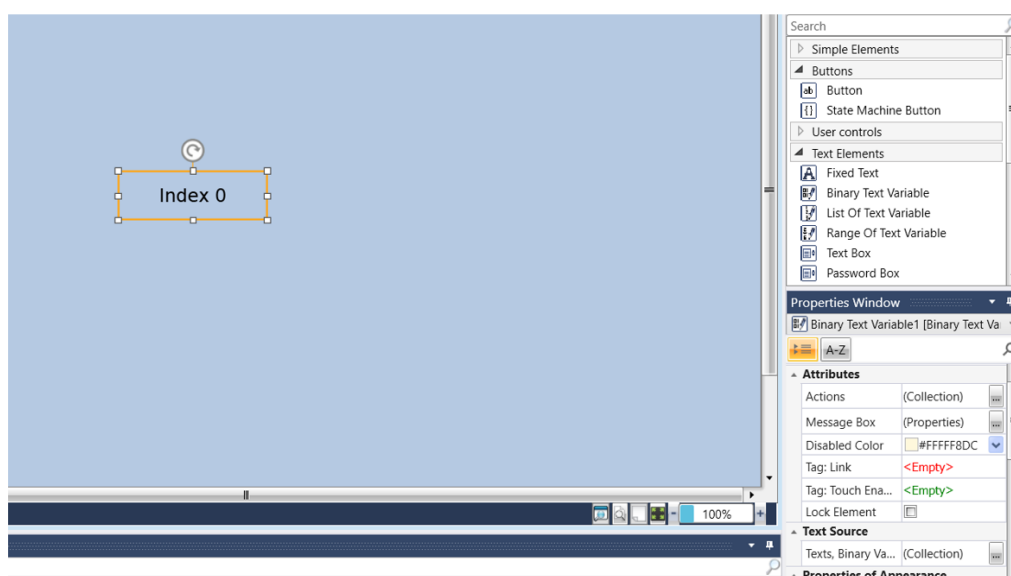


FIGURE 5.14 – Ajout d'un élément Binary Text Variable sur l'écran HMI

Dans cet exemple, j'ai ajouté un élément `Binary Text Variable`. Maintenant, je dois le paramétrer avec la fenêtre `Properties`.

5.5.6 Liaison d'un élément HMI à une variable

Le plus important est de lier l'élément à un Tag. Ce Tag peut être une broche configurée ou une simple variable interne.

Pour effectuer la liaison :

1. Dans la fenêtre `Properties`, aller à `Tag : Link`
2. Sélectionner la variable ou la broche voulue
3. Dans notre cas, on choisira `LED`

5.5.7 Configuration des textes binaires

On va également modifier le texte affiché. Cet élément peut afficher deux textes différents selon l'état de la variable :

- Un texte lorsque la variable est à l'état 0 (éteint)
- Un texte lorsque la variable est à l'état 1 (allumé)

Pour configurer ces textes :

1. Sélectionner la propriété `Text`, `Binary`
2. Renseigner le texte pour l'état 0 (exemple : `"LED OFF"`)
3. Renseigner le texte pour l'état 1 (exemple : `"LED ON"`)

En résumé, si l'état de `LED` change, le texte affiché change également.

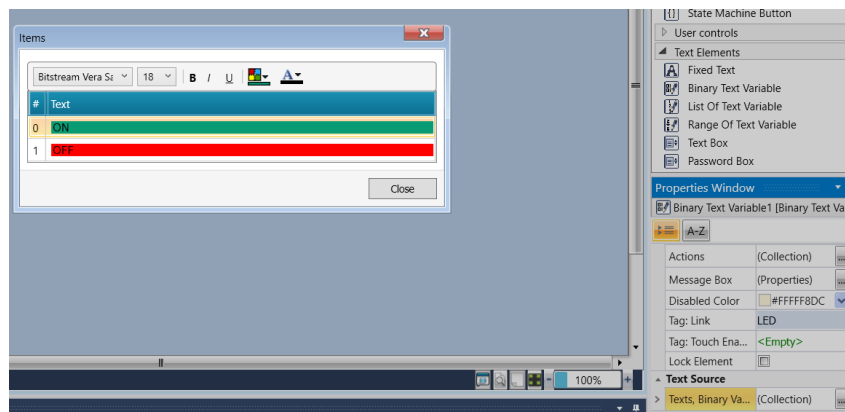


FIGURE 5.15 – Liaison d'un élément HMI à un Tag

5.5.8 Ajout d'une action interactive

Par la suite, on peut ajouter une action sur notre élément pour le rendre interactif.

Configuration de l'action

Pour ajouter une action :

1. Sélectionner `Action → Add New Action`

2. Plusieurs choix d'actions sont disponibles. Nous avons choisi **Toggle Bit**
3. Choisir la variable sur laquelle l'action agit : **LED**
4. Sélectionner le déclencheur (*trigger*) : **Press**

Ainsi, quand on appuie sur l'élément, il inverse l'état du bit de LED, changeant ainsi son état (ON OFF).

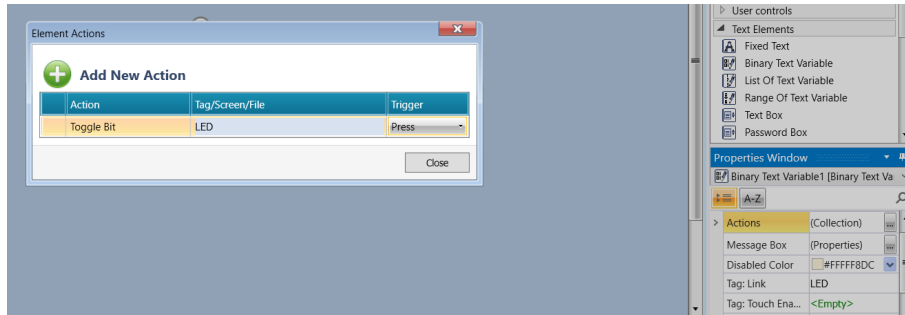


FIGURE 5.16 – Configuration d'une action Toggle Bit

5.6 Gestion des alarmes

Afin d'illustrer le principe de gestion des alarmes dans UniLogic, nous implémentons une alarme simple qui s'active lorsque l'alimentation du système est coupée.

5.6.1 Accès à la gestion des alarmes

Pour intégrer une alarme, il faut tout d'abord se positionner dans le **Solution Explorer**, situé sur la partie gauche de l'interface UniLogic. Plus précisément, on sélectionne le groupe **Alarms** comme montré sur la figure suivante.

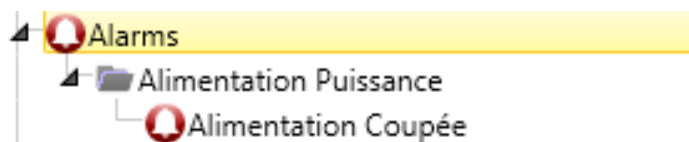


FIGURE 5.17 – Accès au menu de gestion des alarmes

5.6.2 Création d'un groupe d'alarmes

Une fois dans le menu **Alarms**, il est nécessaire de créer un groupe d'alarmes afin d'organiser les différentes alertes du projet. Pour cela, on ajoute un nouveau groupe d'alarme comme illustré ci-dessous.

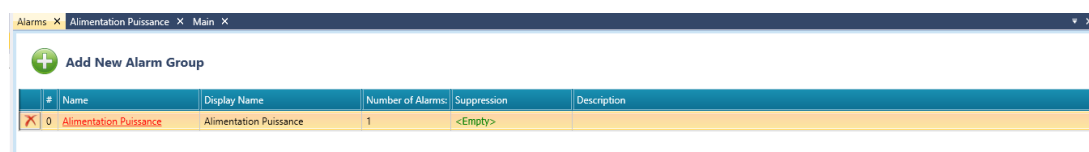
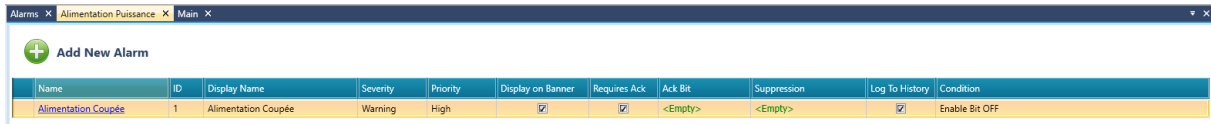


FIGURE 5.18 – Ajout d'un groupe d'alarmes

5.6.3 Ajout d'une alarme

Dans le groupe d'alarme nouvellement créé, on ajoute ensuite une alarme. Chaque alarme correspond à une condition spécifique du système nécessitant une intervention ou une information à l'utilisateur.

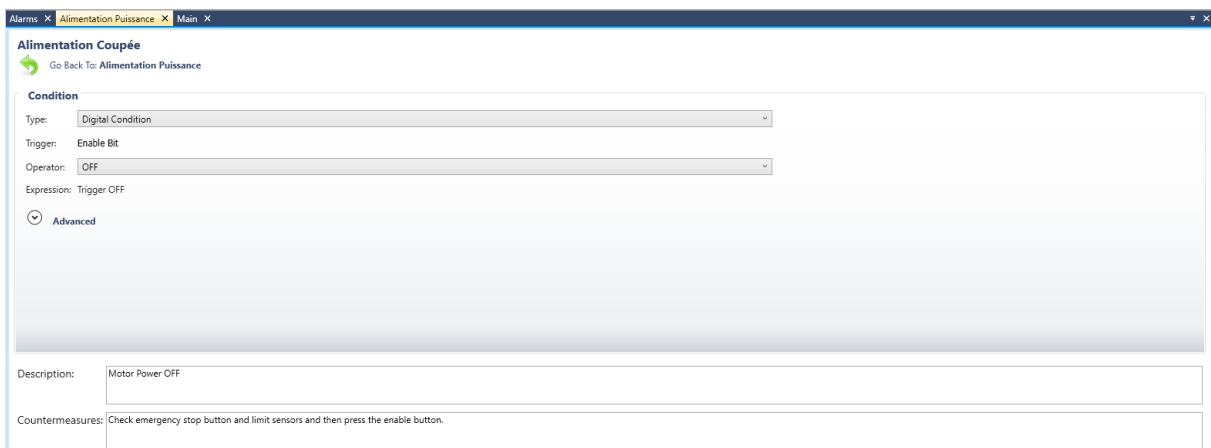


Name	ID	Display Name	Severity	Priority	Display on Banner	Requires Ack	Ack Bit	Suppression	Log To History	Condition
Alimentation Coupée	1	Alimentation Coupée	Warning	High	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<Empty>	<Empty>	<input checked="" type="checkbox"/>	Enable Bit OFF

FIGURE 5.19 – Ajout d'une alarme dans le groupe

5.6.4 Définition du déclencheur de l'alarme

Après la création de l'alarme, il est nécessaire de définir son **trigger** (condition de déclenchement). Dans notre cas, l'alarme doit se déclencher lorsque l'alimentation est coupée. Le déclencheur est donc basé sur un bit logique représentant l'état de l'alimentation.



Alimentation Coupée

Go Back To: Alimentation Puissance

Condition

Type: Digital Condition

Trigger: Enable Bit

Operator: OFF

Expression: Trigger OFF

Advanced

Description: Motor Power OFF

Countermeasures: Check emergency stop button and limit sensors and then press the enable button.

FIGURE 5.20 – Définition du trigger de l'alarme

5.6.5 Logique d'activation dans le Ladder

Enfin, dans la partie **Ladder**, on définit la logique permettant d'activer le bit *Enable Bit* associé à l'alarme. Ce bit passe à l'état actif lorsque l'alimentation est coupée, ce qui provoque automatiquement le déclenchement de l'alarme.

Cette approche permet de séparer clairement :

- la **logique de détection** de la condition d'erreur (Ladder) ;
- la **gestion et l'affichage** de l'alarme (module Alarms).

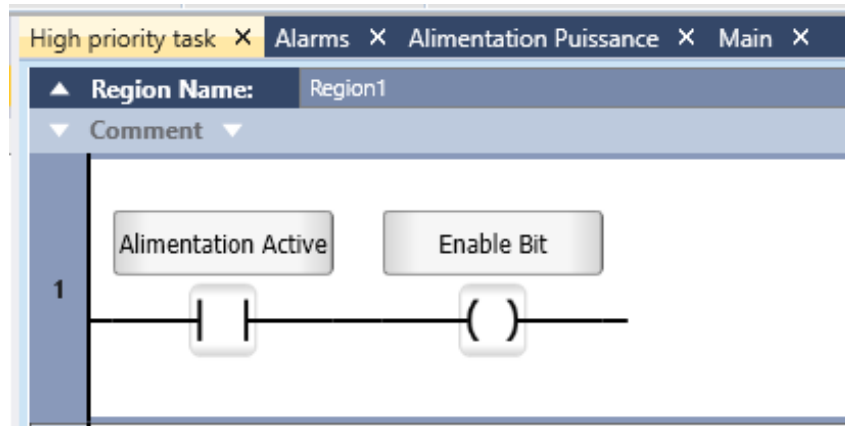


FIGURE 5.21 – Activation du Enable Bit de l’alarme dans le Ladder

5.7 Connexion UART

5.7.1 Configuration de l’UART côté MSP430

Pour que l’automate puisse communiquer avec la carte **MSP430**, il est impératif de configurer correctement l’interface UART de cette dernière. La carte **MSP430FR2355** disponible possède deux interfaces UART : **UART0** et **UART1**.

Un *backchannel* est connecté à l’UART1. Celui-ci permet de communiquer avec l’hôte USB, ce qui est particulièrement utile durant la phase de développement. Il fournit également un canal de communication avec l’ordinateur hôte et peut être utilisé pour créer des interfaces graphiques utilisateur (GUI) ou des outils de débogage sur PC.

Du côté de l’hôte, un port COM virtuel correspondant à l’UART backchannel est automatiquement généré lorsque le kit de développement *LaunchPad* est reconnu par le système (voir figure 5.22). Ce port apparaît sous le nom « **MSP Application UART1** ».

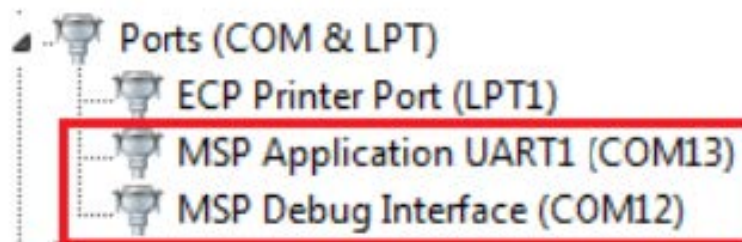


FIGURE 5.22 – Application Backchannel UART visible dans le gestionnaire de périphériques

Remarque importante

Les broches du microcontrôleur associées à l’UART1 ne sont accessibles que via ce backchannel USB. Il n’est donc pas possible de connecter directement des fils de communication sur cette interface UART.

Pour cette raison, nous avons décidé de conserver l'UART1 pour des fins de développement et de débogage, tandis que l'UART0 est utilisée pour la communication entre la carte MSP430 et l'automate (PLC).

5.7.2 Choix de l'UART0 pour la communication PLC

L'avantage principal de l'UART0 par rapport à l'UART1 est qu'elle est accessible physiquement depuis les broches de la carte. Les broches de transmission et de réception sont respectivement :

- **P1.7** : TX (Transmission)
- **P1.6** : RX (Réception)

Ces broches sont utilisées avec un câble série **USB vers TTL RS232** afin d'assurer la liaison avec l'automate.



FIGURE 5.23 – Câble série USB vers TTL RS232

5.7.3 Configuration logicielle des UARTs

La configuration des interfaces UART est réalisée dans le fichier `main.c` du code MSP430. Des constantes nommées `USEUART0` et `USEUART1` sont utilisées pour sélectionner l'UART active.

IMPORTANT

Une seule constante peut être définie à la fois. À l'heure actuelle, il n'est pas possible d'utiliser les deux UARTs simultanément pour commander le robot.

```
17  #define USE_UART0
18  //#define USE_UART1
```

FIGURE 5.24 – Une seule interface UART doit être utilisée à la fois

5.7.4 Paramètres de communication UART

Les différentes configurations possibles ainsi que leurs erreurs de transmission sont détaillées dans la fiche technique du microcontrôleur. Dans notre cas, l'UART est configurée avec les paramètres suivants :

- Débit en bauds : **9600 bauds**
- Bits de données : 8
- Bit d'arrêt : 1
- Parité : aucune

```

//Configure UART module
//SMCLK = 1MHz, Baudrate = 9600
//UCBRx = 8, UCBRFx = 0, UCBSx = 0xD6, UCOS16 = 0
EUSCI_A_UART_initParam param = {0};
param.selectClockSource = EUSCI_A_UART_CLOCKSOURCE_SMCLK; //SMCLK
param.clockPrescaler = 6; // Prescaler
param.firstModReg = 8; // Set Low Freq
param.secondModReg = 17; // Set Modulation
param.parity = EUSCI_A_UART_NO_PARITY; // No parity
param.msborLsbFirst = EUSCI_A_UART_LSB_FIRST; // LSB First
param.numberofStopBits = EUSCI_A_UART_ONE_STOP_BIT; // 1 stop bit
param.uartMode = EUSCI_A_UART_MODE;
param.overSampling = 1;

```

FIGURE 5.25 – Configuration de l'UART

Les valeurs de *prescaler*, *firstModReg* et *secondModReg* ont été déterminées à partir du tableau 15-4 du *datasheet* du microcontrôleur.

Remarque

Nous avons tenté d'augmenter le débit en bauds à **115200**, mais la carte MSP430 ne recevait pas correctement les messages envoyés depuis l'automate.

5.7.5 Communication entre le PLC et la carte MSP430

Pour permettre la communication avec le microcontrôleur, le port USB de l'automate doit être correctement configuré. Cette configuration se fait dans l'onglet :

PLC Communications > Physicals > Controller USB Port

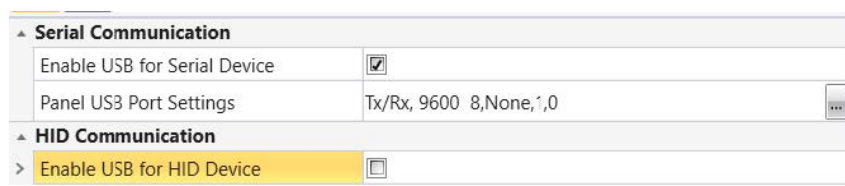


FIGURE 5.26 – Configuration du port USB du PLC

5.7.6 Envoi des messages

Afin d'envoyer des commandes à la carte MSP430, des fonctions spécifiques ont été créées pour chaque type de message. Ces fonctions sont appelées depuis le programme principal du PLC et ont pour rôle d'envoyer la trame correspondante au microcontrôleur.



FIGURE 5.27 – Définition d’une trame dans le Message Composer

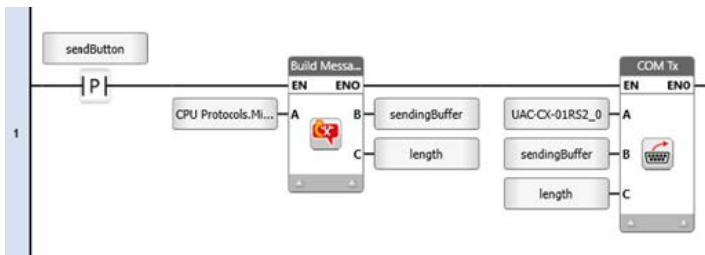


FIGURE 5.28 – Bloc UniLogic pour l’envoi d’un message

5.7.7 Réception des messages

Le même port de communication est utilisé pour recevoir les messages en provenance de la carte MSP430. UniLogic fournit des blocs dédiés permettant de lire les données entrantes.

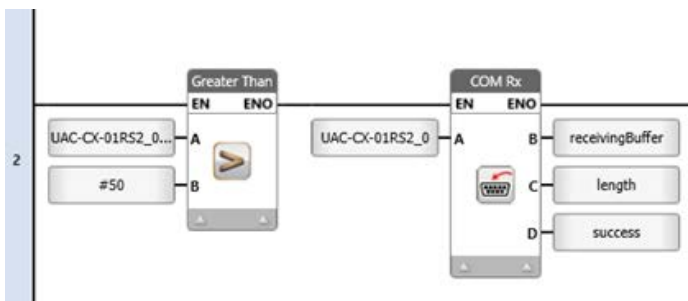


FIGURE 5.29 – Bloc UniLogic pour la réception d’un message

Remarque finale

Pour plus de détails sur la communication UART ou sur la programmation avancée de la carte MSP430 (y compris le code source), il est recommandé de consulter les archives des anciens projets du robot *Pick and Place*, notamment :

- ARCHIVES_2023P
- ARCHIVES_2024P
- ARCHIVES_2024A

5.8 Connexion Modbus

Cette section décrit pas à pas la mise en place d'une connexion Ethernet entre un PC et un automate **UniStream**, ainsi que la configuration du protocole **Modbus** afin de permettre l'échange de données avec une application externe ou une IHM.

5.8.1 Prérequis

Avant de commencer, il est nécessaire de :

- installer le logiciel **UniLogic** sur un PC (disponible uniquement sous Windows) ;
- relier l'automate UniStream et le PC à l'aide d'un câble Ethernet.

5.8.2 Accès aux paramètres de l'automate

Pour configurer le réseau, il faut accéder aux paramètres système directement depuis l'écran de l'automate.

Étape 1 Appuyer longuement en haut à droite de l'écran de l'automate jusqu'à l'apparition d'un menu contextuel.

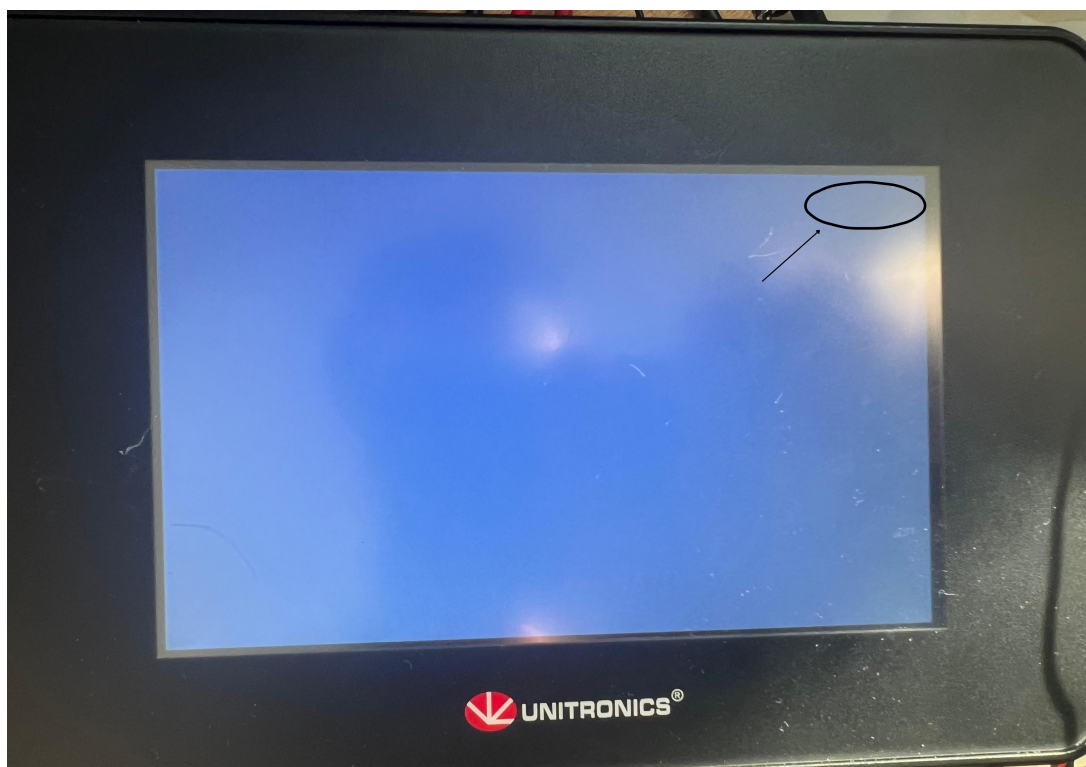


FIGURE 5.30 – Accès au menu système de l'automate

Étape 2 Sélectionner l'application **UniApps**.



FIGURE 5.31 – Sélection de l'application UniApps

Étape 3 Une fois dans le menu système, accéder à la section **Network**.

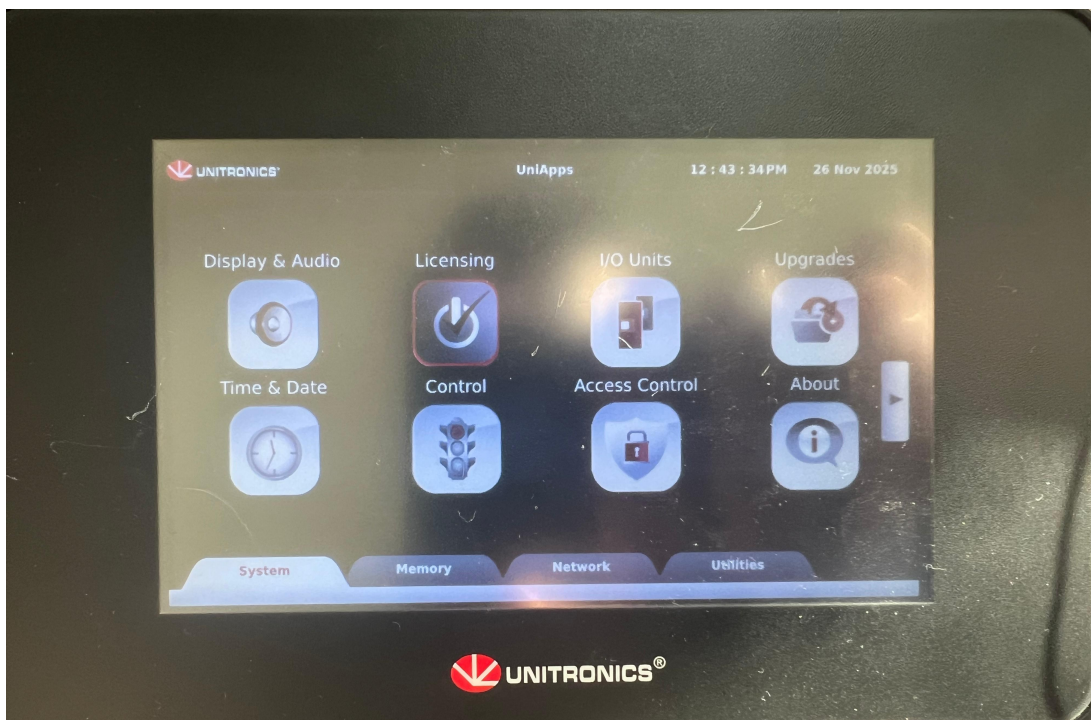


FIGURE 5.32 – Menu système de l'automate

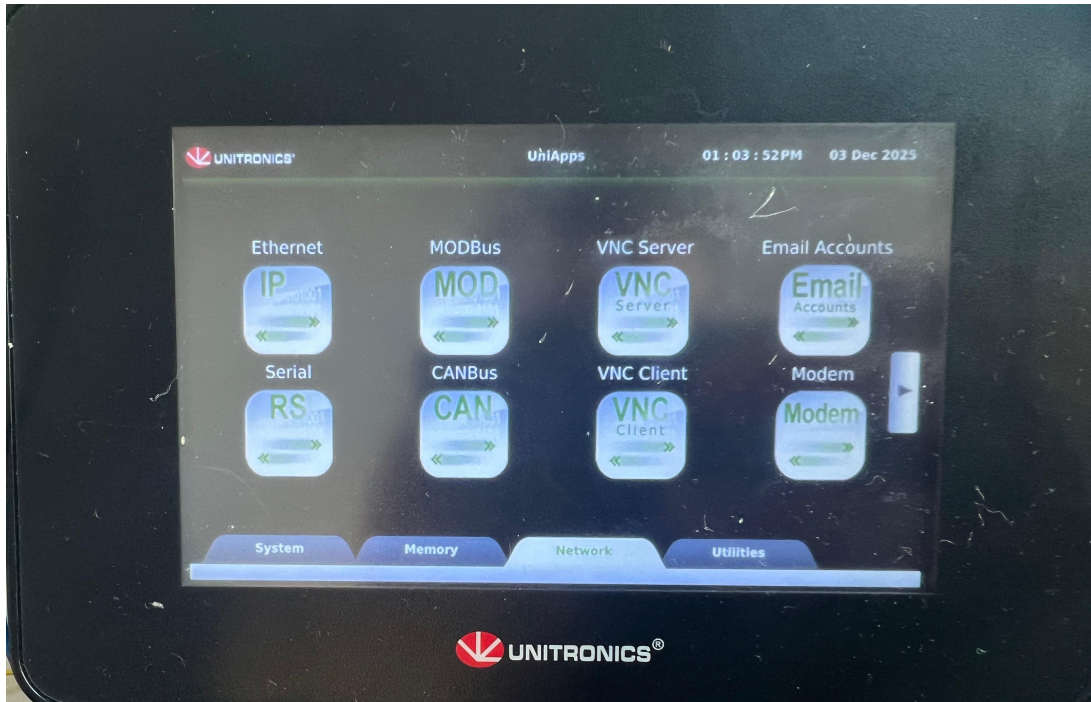


FIGURE 5.33 – Accès au menu Network

5.8.3 Configuration réseau de l'automate

Étape 4 Dans le menu **Network**, sélectionner **Ethernet** afin de configurer les adresses IP de l'automate. Il est possible de définir des adresses IP personnalisées ou d'utiliser une configuration équivalente à celle illustrée ci-dessous.



FIGURE 5.34 – Configuration Ethernet de l'automate

5.8.4 Configuration réseau dans UniLogic

Étape 5 Ouvrir UniLogic puis se rendre dans le menu :

PLC Communication → Physical

Modifier les adresses IP dans les sections :

- **Panel Ethernet** : correspond au *Panel IP Setting* de l'automate ;
- **CPU Ethernet** : correspond au *CPU IP Setting*.

Ces adresses doivent être identiques à celles définies sur l'automate.

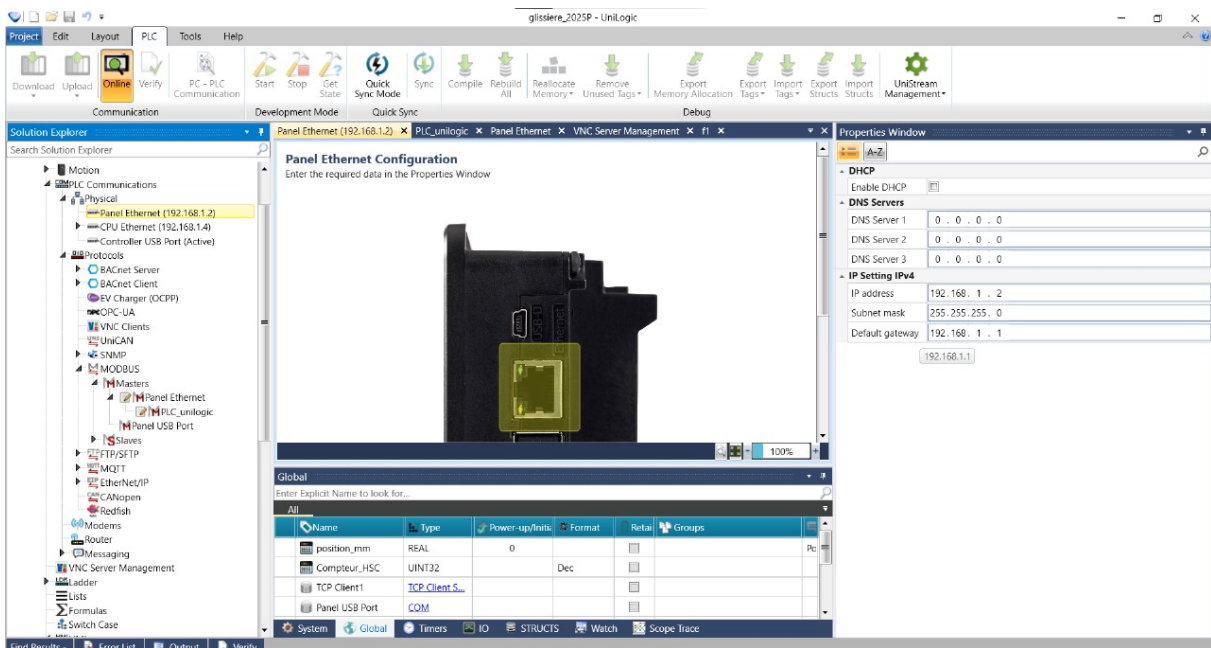


FIGURE 5.35 – Configuration Panel Ethernet dans UniLogic

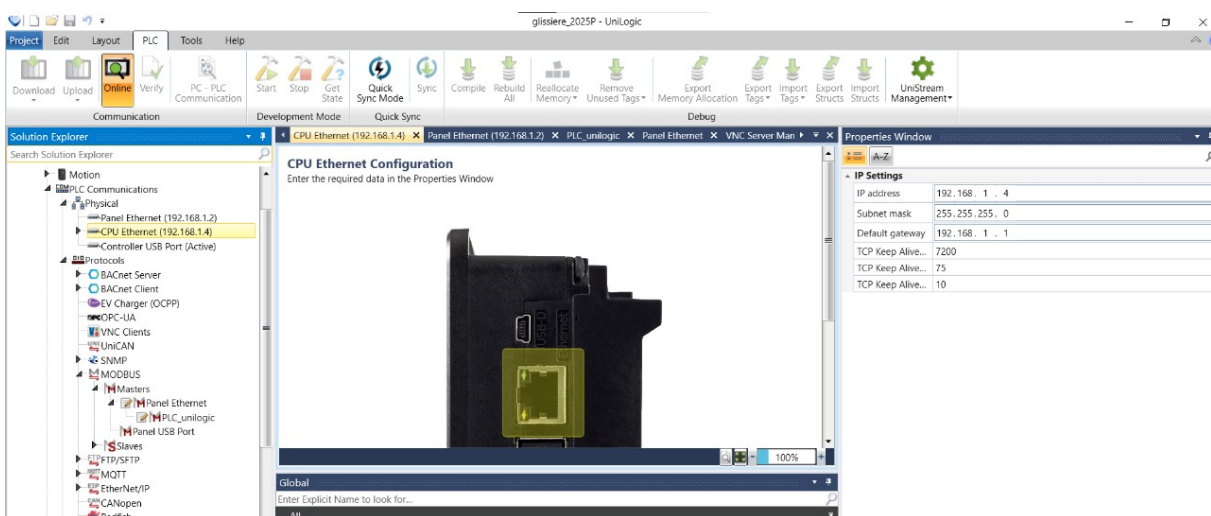


FIGURE 5.36 – Configuration CPU Ethernet dans UniLogic

Étape 6 Pour vérifier la connexion entre le PC et l'automate, ouvrir :

PC-PLC Communication

Sélectionner **Ethernet** et saisir l'adresse IP de l'automate. Si la connexion est correcte, une coche verte apparaît.

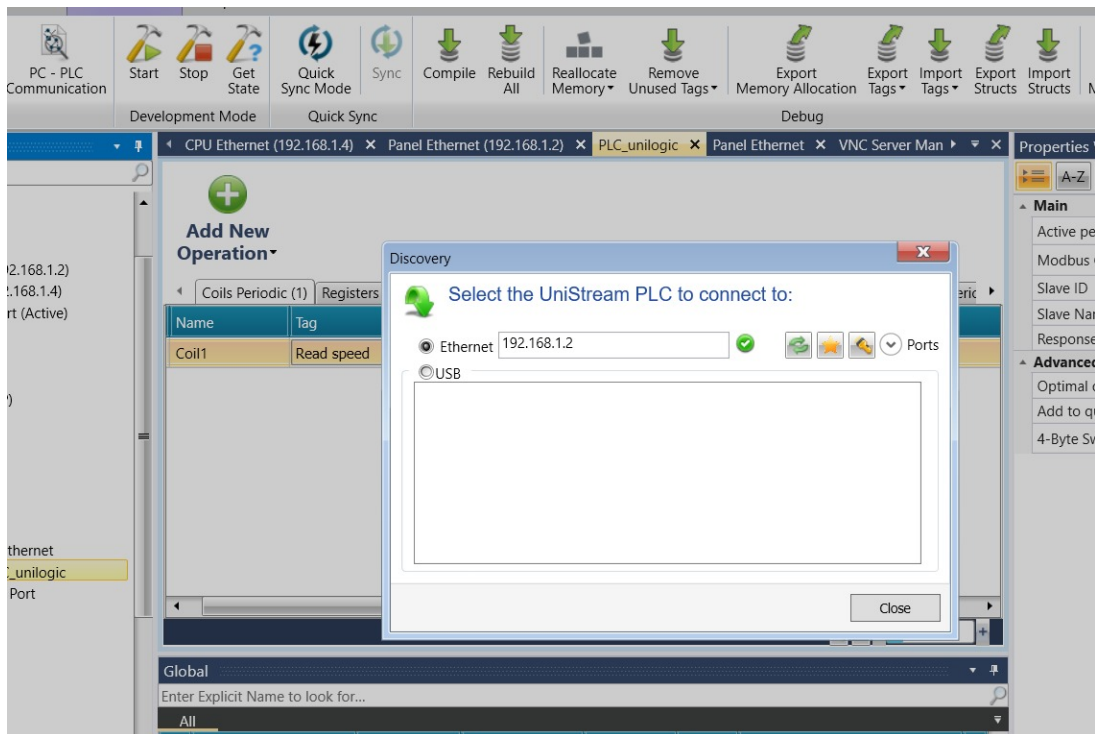


FIGURE 5.37 – Vérification de la connexion PC-PLC

5.8.5 Configuration du protocole Modbus

Étape 7 Dans UniLogic, accéder au menu :

PLC Communication → Protocols → MODBUS → Panel Ethernet

Ajouter un nouvel esclave Modbus (*Add New Slave*) puis reproduire la configuration illustrée ci-dessous. Il est nécessaire d'ajouter au minimum **une opération Modbus** afin de définir un registre à lire ou à écrire.

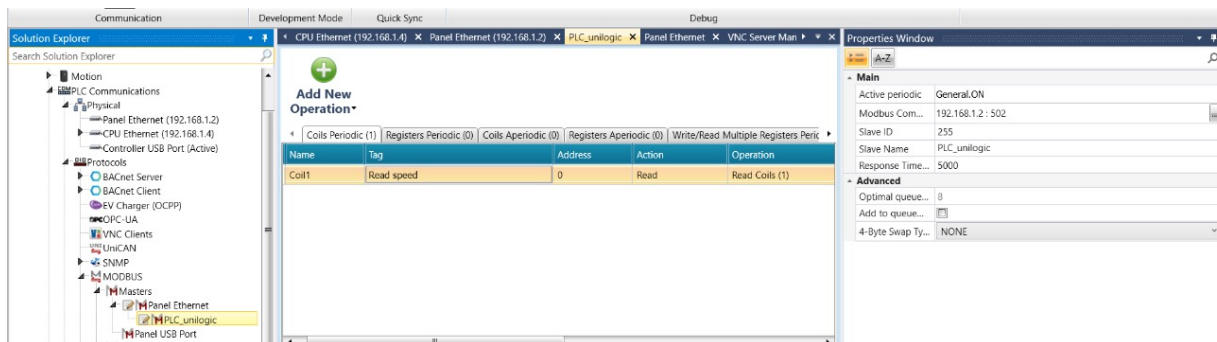


FIGURE 5.38 – Ajout d'un esclave Modbus

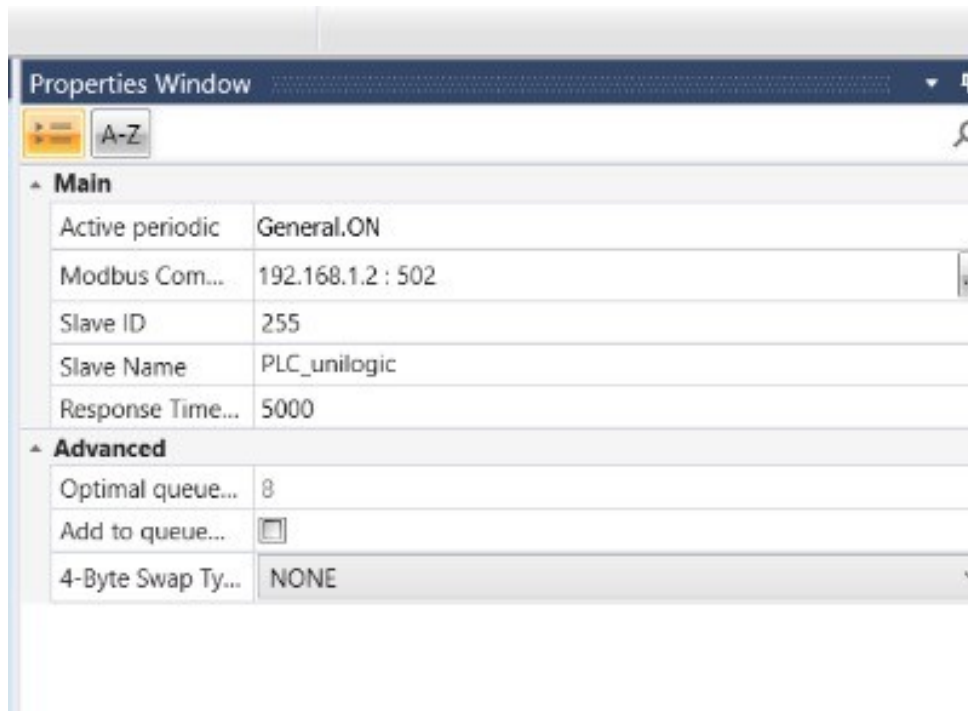


FIGURE 5.39 – Configuration des opérations Modbus

5.8.6 Téléchargement du projet

Étape 8 Une fois la configuration terminée, cliquer sur **Download** afin de transférer le projet vers l'automate. Après le téléchargement, l'automate passe automatiquement en **Online Mode**.

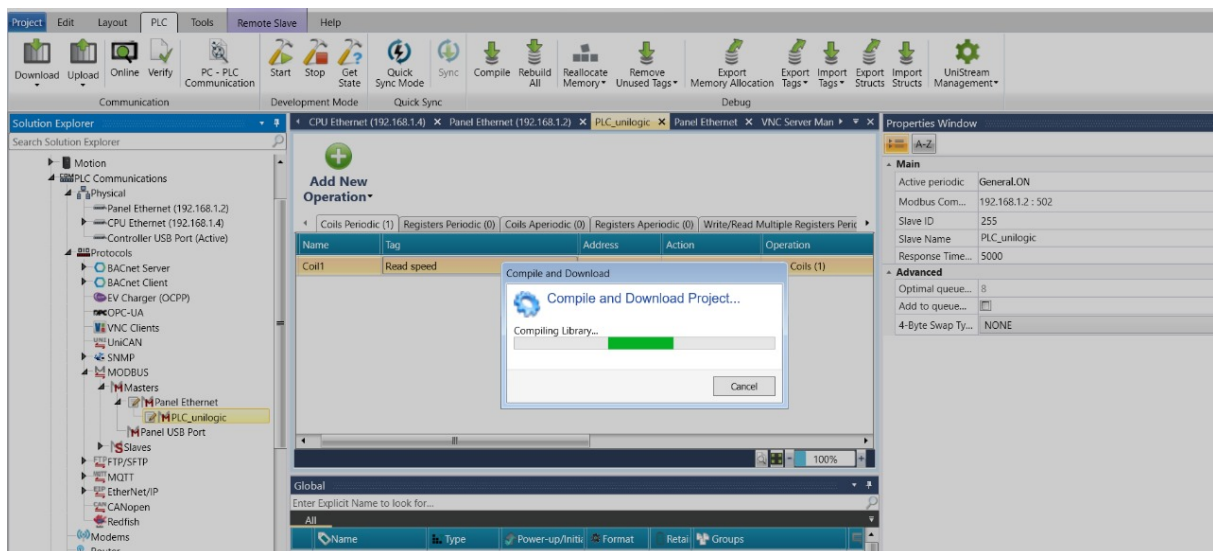


FIGURE 5.40 – Téléchargement du projet

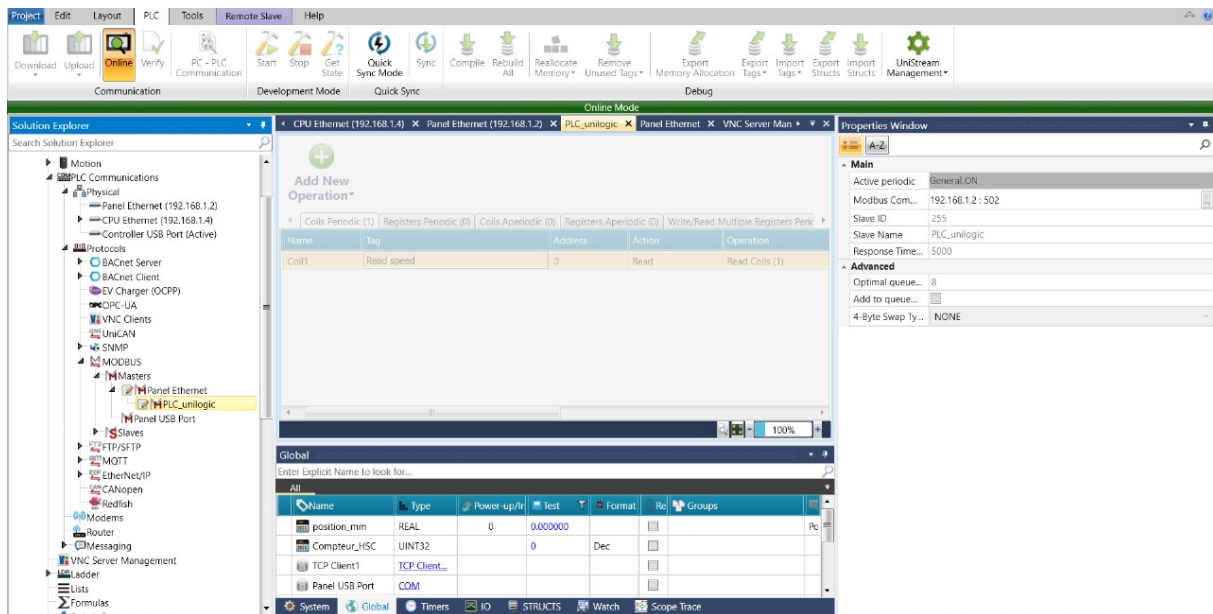


FIGURE 5.41 – Automate en mode Online

5.8.7 Test de la connexion Modbus

Étape 9 : Test via PowerShell Depuis un terminal PowerShell, exécuter la commande suivante :

```
Test-NetConnection xx.xx.xx.xx -Port 502
```

où xx.xx.xx.xx correspond à l'adresse IP de l'automate. Si le port **502** répond, la communication Modbus est fonctionnelle.

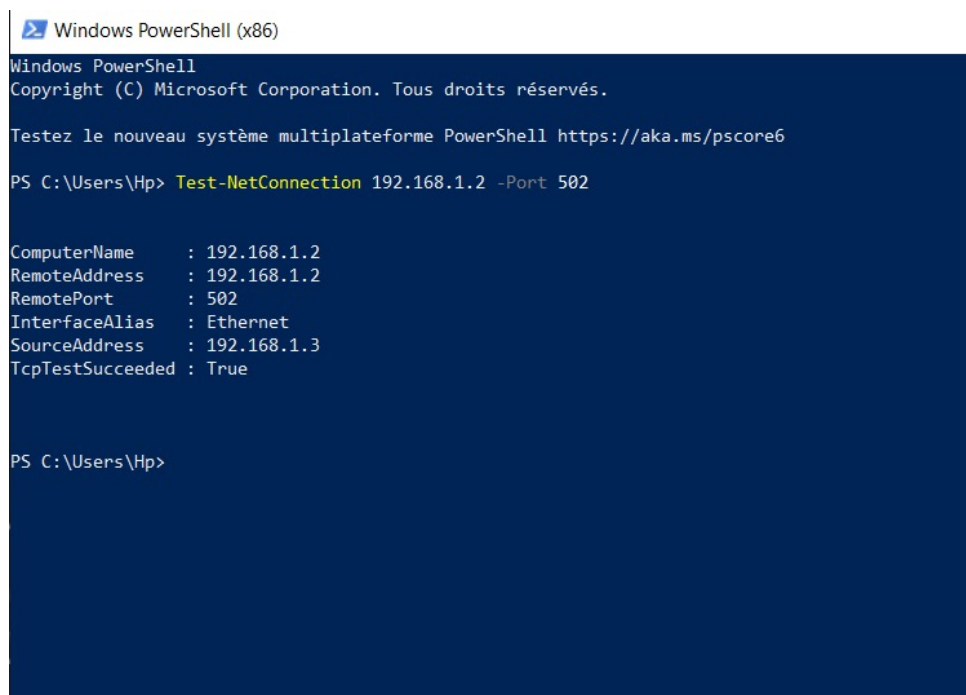


FIGURE 5.42 – Test de la connexion Modbus via PowerShell

5.8.8 Test depuis l'IHM

Étape 10 Lancer l'IHM puis accéder aux paramètres de connexion. Renseigner les informations suivantes :

- adresse IP de l'automate ;
- port Modbus : **502** ;
- serveur Modbus : **1**.

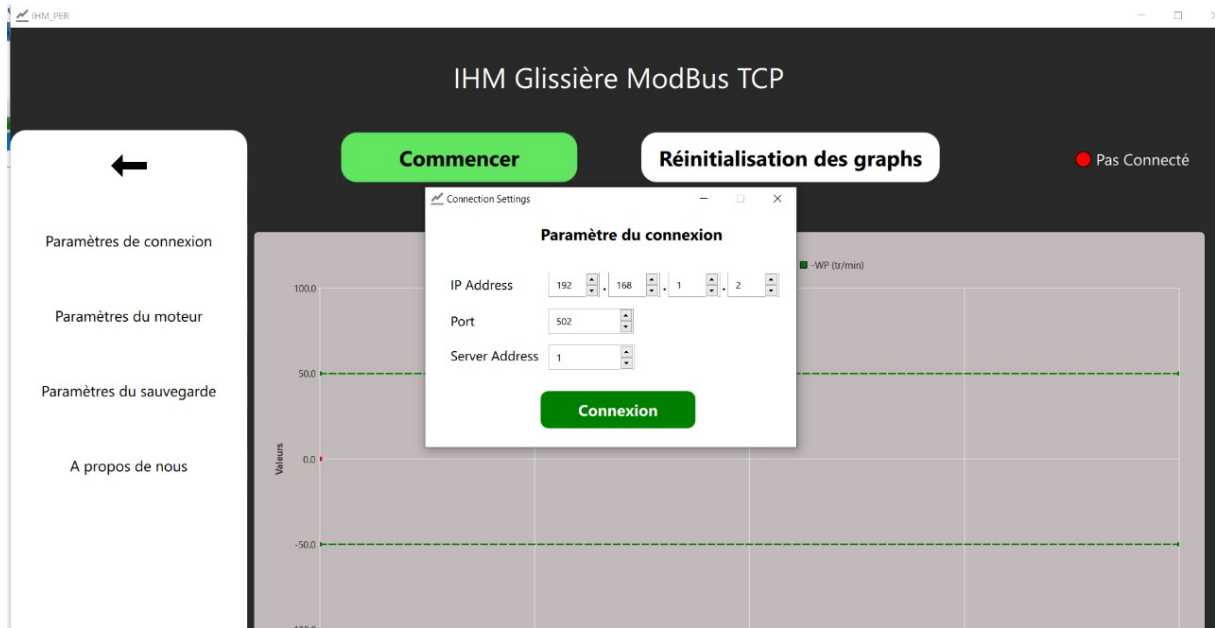


FIGURE 5.43 – Paramétrage de la connexion Modbus dans l'IHM

Lorsque la connexion est établie, l'état passe de « *Pas connecté* » à « *Connecté* ».

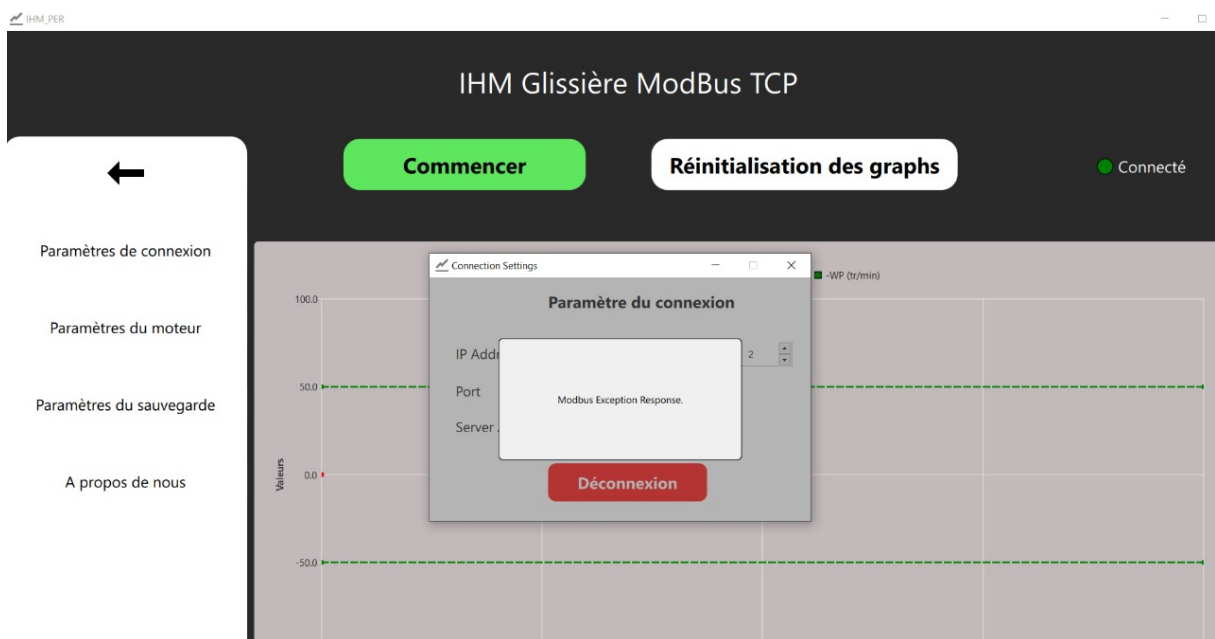


FIGURE 5.44 – Connexion Modbus établie avec succès

5.9 Transfert vers l'automate

Chapitre 6

Travaux pratiques sur les E/S du UniStream

Ce chapitre présente une série d'exercices pratiques permettant de maîtriser progressivement les différentes fonctionnalités de l'automate UniStream US5-B10-T42. Chaque travail pratique est conçu pour mettre en œuvre les concepts abordés précédemment :

6.1 Exercice 1 – Commande TOR (LED)

6.1.1 Objectif

L'objectif est de mettre en œuvre une commande Tout ou Rien (TOR) d'une LED connectée à une sortie digitale de l'automate Unitronics UniStream US5-B10-T42, en utilisant un bouton virtuel sur l'IHM pour activer ou désactiver la LED.

Cet exercice a pour but de découvrir les principes de base de la commande numérique, la création de variables (Tags) et l'interaction entre l'IHM et les sorties de l'automate sans recourir à la programmation Ladder.

6.1.2 Matériel requis

- Automate UniStream US5-B10-T42
- LED 24V
- Alimentation 24V DC
- Câbles de connexion

6.1.3 Cahier des charges

Spécifications fonctionnelles

Le système doit réaliser une commande d'une LED à partir de deux boutons virtuels distincts sur l'IHM : un bouton pour allumer la LED et un bouton pour l'éteindre.

Étape	Action associée	Transition (événement)
1 : LED éteinte	LED_OUT = 0	Appui sur bouton ON
2 : LED allumée	LED_OUT = 1	Maintien de l'état
3 : LED éteinte	LED_OUT = 0	Appui sur bouton OFF

TABLE 6.1 – Séquence de fonctionnement de la commande TOR

Principe de fonctionnement

Le système comporte deux boutons indépendants :

- **Bouton ON** : Met la LED à l'état allumé (LED_OUT = 1)
- **Bouton OFF** : Met la LED à l'état éteint (LED_OUT = 0)

État initial : LED éteinte (LED_OUT = 0)

6.1.4 Schéma de câblage

Vue d'ensemble des borniers

CM0	I0	I1	I2	I3	I4	I5	I6	I7	I8	I9	-	CM1	A0	A1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
00	01	02	03	04	05	06	07	08	09	010	011	-	0V0	+V0

FIGURE 6.1 – Borniers de sorties digitales de l'automate

Procédure de câblage

Le câblage s'effectue en deux étapes :

Étape 1 : Alimentation du bus de sorties Avant de connecter la LED, il est nécessaire d'alimenter le bus des ports de sorties :

- Connecter le **+24V** sur la borne **1** (marquée +V0)
- Connecter le **0V (masse)** sur la borne **2** (marquée 0V0)

Étape 2 : Connexion de la LED Une fois le bus de sorties alimenté :

- Connecter l'**anode** de la LED sur la broche de sortie correspondante : **00**
- Connecter la **cathode** de la LED directement à la **masse (0V)**

Récapitulatif des connexions

Borne	Signal	Connexion
1	+V0	Alimentation +24V DC
2	0V0	Masse (0V)
O0	Sortie digitale 0	Anode LED
Masse	0V	Cathode LED

TABLE 6.2 – Tableau récapitulatif du câblage

6.1.5 Programmation dans UniLogic

Une fois ces étapes de câblage terminées, on va passer au logiciel UniLogic pour développer le programme et le téléverser sur l'automate.

Création du projet

Création du projet

Étape 1 : Créer un nouveau projet

1. Lancer UniLogic
2. Cliquer sur New Project
3. Renseigner le nom du projet (exemple : Lector_1hm)
4. Choisir l'emplacement de sauvegarde
5. Cliquer sur Next

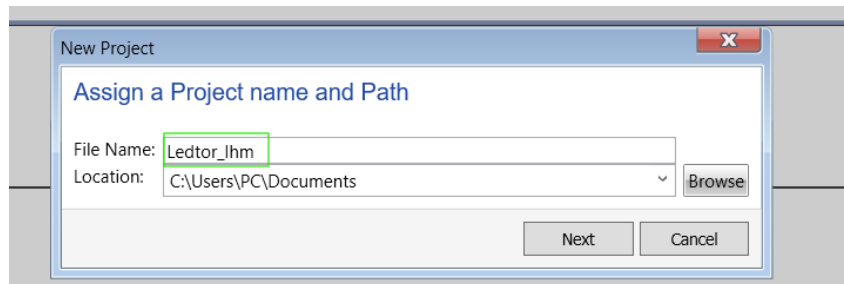


FIGURE 6.2 – Création d'un nouveau projet

Étape 2 : Sélectionner le modèle d'automate

1. Dans la fenêtre Select a Controller, choisir :
 - **Model Family** : Built-in
 - **Catégorie** : 5 /7"
 - **Modèle** : US5/7-x10-T42
2. Cliquer sur Finish

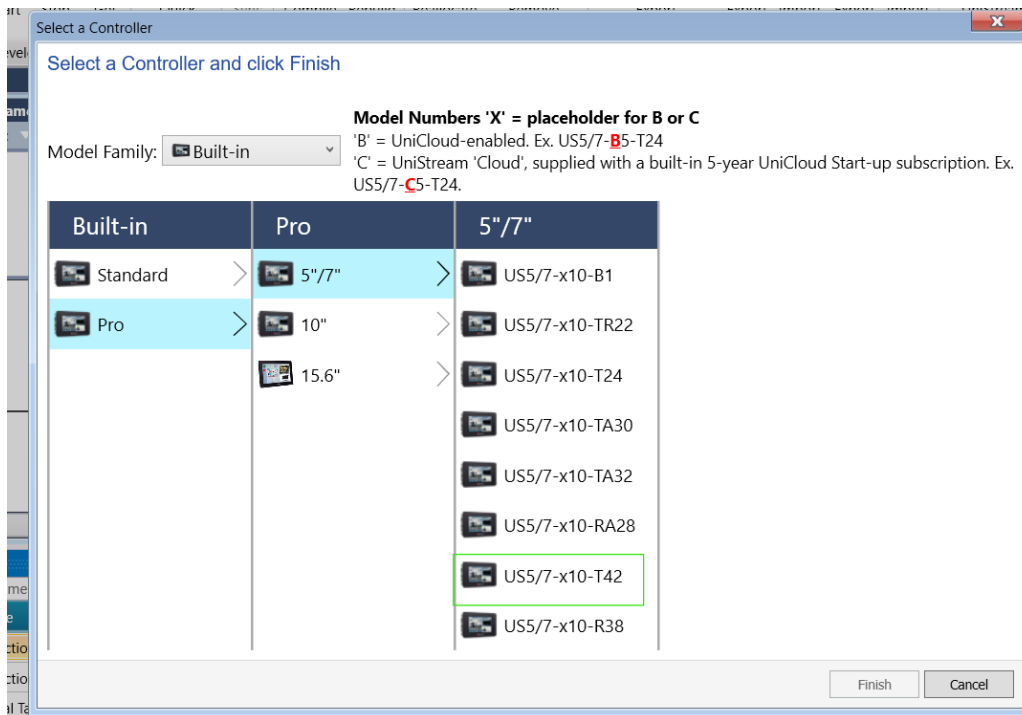


FIGURE 6.3 – Sélection du modèle d'automate

Étape 3 : Configuration de la broche de sortie Pour cet exercice, nous n'avons pas besoin de créer de variables supplémentaires. On peut directement passer à la configuration des broches.

1. Dans la barre Solution Explorer, aller sur IO → On board IO → Digital Outputs
2. Sélectionner la sortie Digital Outputs_0 (O0)
3. Dans la colonne Alias Name, saisir : LED

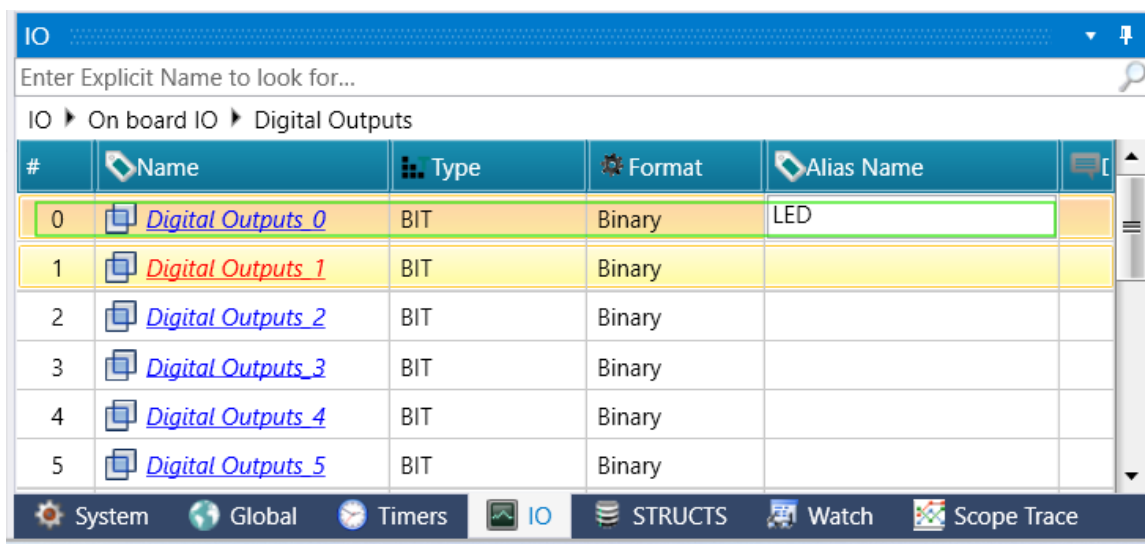


FIGURE 6.4 – Configuration de la sortie digitale pour la LED

La sortie O0 est maintenant configurée et peut être utilisée dans l'IHM sous le nom LED.

Étape 4 : Conception de l'IHM Pour cet exercice, nous n'avons pas besoin de programmer en Ladder. L'IHM seule permettra de respecter notre cahier des charges.

1. Dans la barre **Solution Explorer**, développer **HMI** → **Module1**
2. Double-cliquer sur **Screen1** pour ouvrir l'éditeur d'écran

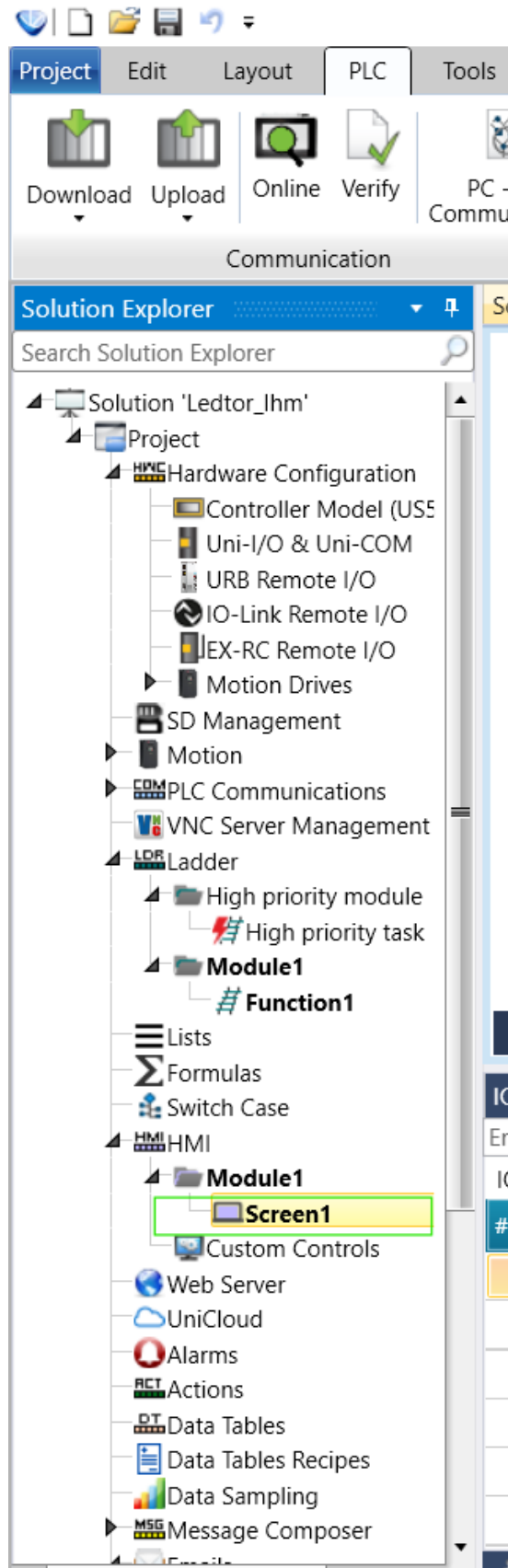


FIGURE 6.5 – Accès à l'écran HMI dans le Solution Explorer

Nous allons maintenant ajouter les éléments graphiques nécessaires : deux boutons

(ON et OFF) et un indicateur visuel de l'état de la LED.

Étape 5 : Ajout des boutons de commande

Placement des boutons

1. Dans la Toolbox, développer Buttons
2. Glisser-déposer deux éléments Button sur l'écran

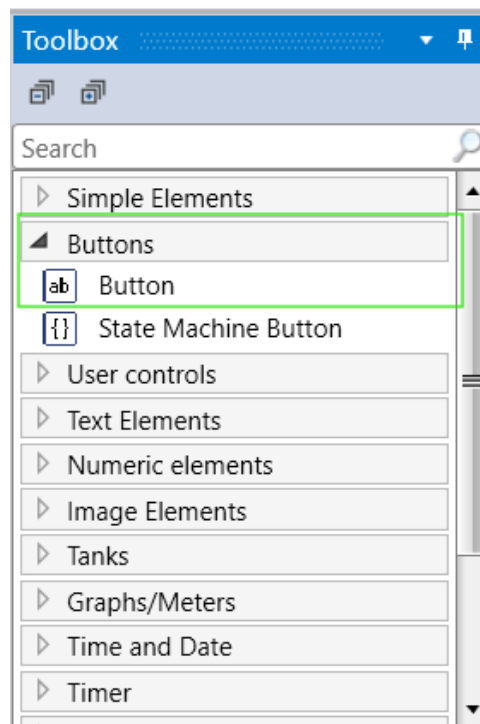


FIGURE 6.6 – Sélection des boutons dans la Toolbox

Configuration du premier bouton (ON) Dans la fenêtre Properties Window :

- Text Label : Saisir ON
- Background Fill : Choisir une couleur verte (par exemple)
- Tag: Touch Enabled : Lier au Tag LED

Configuration du second bouton (OFF) Dans la fenêtre Properties Window :

- Text Label : Saisir OFF
- Background Fill : Choisir une couleur rouge (par exemple)
- Tag: Touch Enabled : Lier au Tag LED

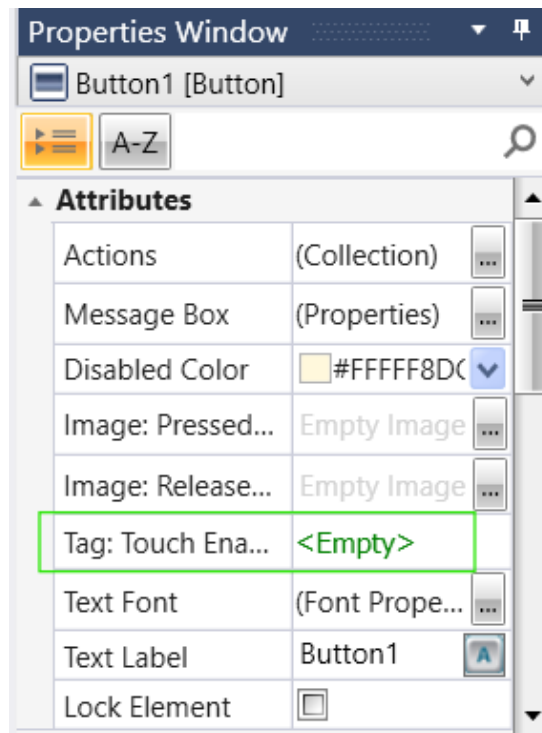


FIGURE 6.7 – Configuration des propriétés d'un bouton

Étape 6 : Configuration des actions des boutons

Action du bouton ON

1. Sélectionner le bouton Button1 (ON)
2. Dans la fenêtre Properties Window, cliquer sur Actions → (Collection)
3. Dans la fenêtre Element Actions, cliquer sur Add New Action
4. Configurer l'action :
 - Action : Sélectionner Set Bit
 - Tag/Screen/File : Saisir LED
 - Trigger : Sélectionner Click/Tap ou Press
5. Cliquer sur Close

Action du bouton OFF

1. Sélectionner le bouton Button2 (OFF)
2. Dans la fenêtre Properties Window, cliquer sur Actions → (Collection)
3. Dans la fenêtre Element Actions, cliquer sur Add New Action
4. Configurer l'action :
 - Action : Sélectionner Reset Bit
 - Tag/Screen/File : Saisir LED
 - Trigger : Sélectionner Click/Tap ou Press
5. Cliquer sur Close

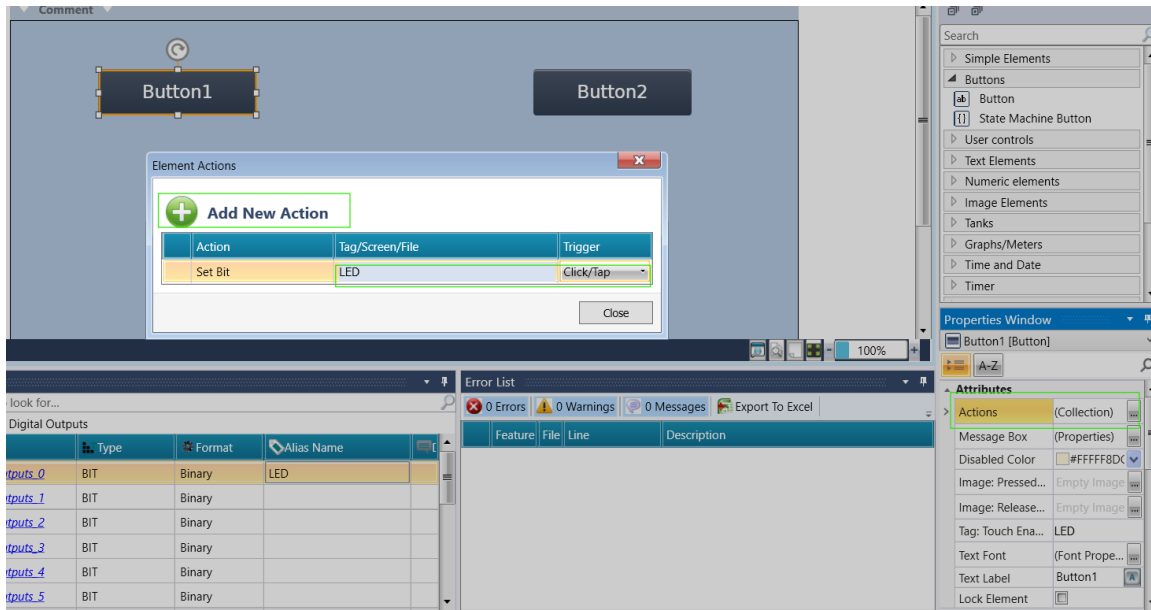


FIGURE 6.8 – Configuration de l'action Set Bit pour le bouton ON

Les deux boutons sont maintenant configurés pour commander directement la sortie LED sans programmation Ladder.

Étape 7 : Finalisation de l'IHM La conception de l'interface est terminée. Voici le rendu final de l'écran :

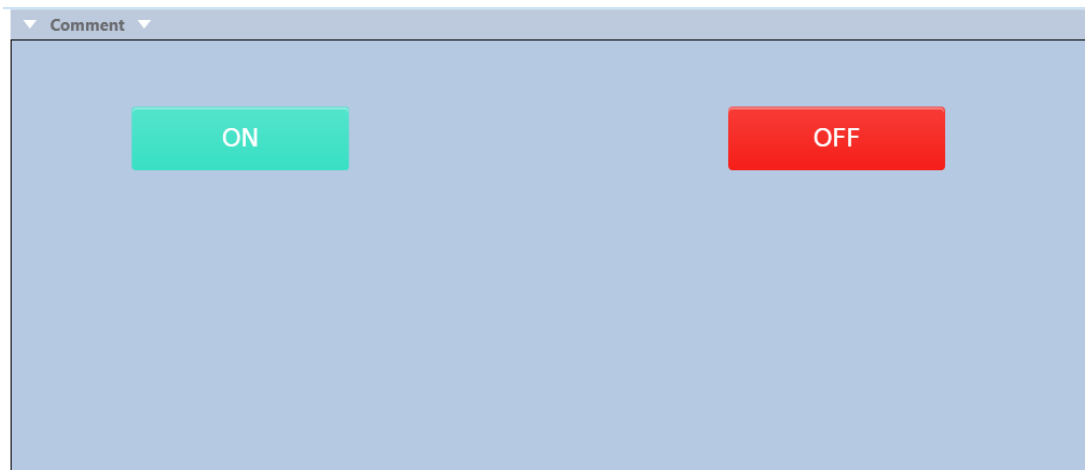


FIGURE 6.9 – Rendu final de l'interface HMI

L'écran comporte :

- Un bouton ON (vert) pour allumer la LED
- Un bouton OFF (rouge) pour éteindre la LED

Il ne reste plus qu'à connecter l'automate et téléverser le programme.

6.2 Exercice 2 – Réception TOR (entrée physique)

6.2.1 Objectif

L'objectif de cet exercice est de réaliser une commande par bascule (*toggle*) d'une LED à l'aide d'un bouton poussoir physique. Chaque appui sur le bouton inverse l'état de la LED.

6.2.2 Matériel requis

- Automate UniStream US5-B10-T42
- LED
- Bouton poussoir NO (Normalement Ouvert) ou NF (Normalement Fermé)
- Alimentation 24V DC
- Câbles de connexion

6.2.3 Ports utilisés

- **Entrée digitale** : I0 (bouton poussoir)
- **Sortie digitale** : O0 (LED)

6.2.4 Cahier des charges

Spécifications fonctionnelles

Étape	Action associée	Transition (événement)
1 : LED éteinte	LED_OUT = 0	Appui bouton physique (front montant)
2 : LED allumée	LED_OUT = 1	Nouvel appui bouton (front montant)

TABLE 6.3 – Séquence de fonctionnement toggle avec bouton physique

Le système fonctionne comme une bascule : chaque appui sur le bouton poussoir physique inverse l'état de la LED.

6.2.5 Schéma de câblage

Câblage de la LED (identique à l'exercice 1)

- Connecter le **+24V** sur la borne **1 (+V0)**
- Connecter le **0V (masse)** sur la borne **2 (0V0)**
- Connecter l'**anode** de la LED (via résistance 330) sur **O0**
- Connecter la **cathode** de la LED à la **masse**

Câblage du bouton poussoir

Étape 1 : Alimentation du bus d'entrées

- Connecter la **masse (0V)** du bus des broches d'entrée : relier **CM0** à la **masse**

Étape 2 : Connexion du bouton poussoir

- Connecter le **commun** du bouton poussoir au **+24V**
- Connecter le contact **NO** (Normalement Ouvert) ou **NF** (Normalement Fermé) à la broche **I0**

Récapitulatif des connexions

Composant	Borne automate	Signal
Alimentation +24V	+V0 (bus sorties)	+24V DC
Alimentation 0V	0V0 et CM0	Masse (0V)
Bouton - Commun	+24V externe	+24V DC
Bouton - Contact NO/NF	I0	Signal entrée digitale
LED - Anode	O0	Sortie digitale
LED - Cathode	Masse	0V

TABLE 6.4 – Tableau récapitulatif du câblage

6.2.6 Programmation dans UniLogic

Une fois les connexions terminées, on crée les variables comme précédemment et on écrit le programme Ladder. Pour cet exercice, nous n'avons pas besoin d'IHM, la commande se fait uniquement via le bouton physique.

Création du projet

1. Créer un nouveau projet : TP2_Toggle_Bouton
2. Sélectionner le modèle : US5-B10-T42

Configuration des broches

Dans I0 → On board I0 :

Entrée digitale

- Digital Inputs_0 (I0) Alias : Bp₁

Sortie digitale

- Digital Outputs_0 (O0) Alias : LED₁

Programmation Ladder

Aller dans Ladder → Module1 → Function1 et écrire le programme suivant :

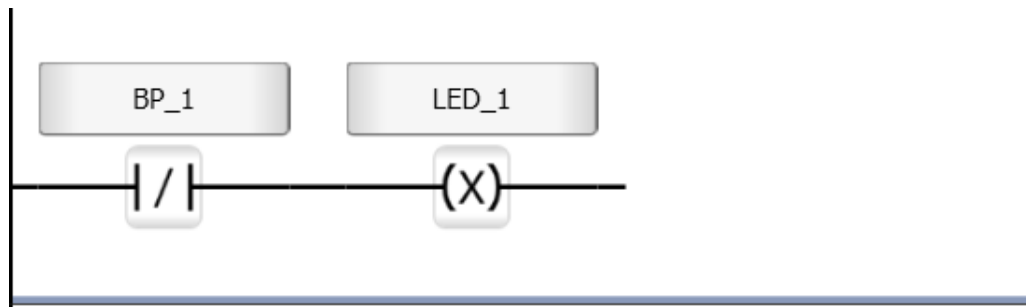


FIGURE 6.10 – Programme Ladder pour la fonction toggle

Principe du programme Le programme utilise la détection de front montant du bouton pour inverser l'état de la LED à chaque appui.

Note : Le détail du code Ladder dépend de l'utilisation d'un bloc fonction Toggle Bit ou d'une logique avec fronts montants et mémorisation.

6.3 Exercice 3 – Commande PWM (LED variable)

6.4 Exercice 4 – Lecture analogique (potentiomètre)

6.4.1 Objectif

L'objectif de cet exercice est de lire une valeur analogique provenant d'une alimentation de laboratoire (simulant un capteur), de convertir cette valeur numérique en tension réelle, et de l'afficher sur l'IHM de l'automate.

6.4.2 Matériel requis

- Automate UniStream US5-B10-T42
- Alimentation de laboratoire réglable (0-10V)
- Câbles de connexion

6.4.3 Ports utilisés

- Entrée analogique : AI0 (0-10V)

6.4.4 Cahier des charges

Spécifications fonctionnelles

Le système doit :

- Lire en continu la tension appliquée sur l'entrée AI0
- Convertir la valeur numérique (0-4095) en tension (0-10V)
- Afficher sur l'IHM :
 - La valeur brute (0-4095)
 - La tension convertie en Volts (0.00-10.00V)

Conversion analogique-numérique

L'entrée analogique de l'automate fonctionne sur **12 bits par défaut**, soit une plage de valeurs de **0 à 4095**.

Formule de conversion Pour convertir la valeur numérique en tension :

$$Tension(V) = \frac{Valeurbrute \times 10}{4095}$$

Valeur brute	Tension (V)
0	0.00 V
2048	5.00 V
4095	10.00 V

TABLE 6.5 – Exemples de conversion

6.4.5 Schéma de câblage

Procédure de câblage

Le câblage de l'entrée analogique s'effectue en trois étapes :

Étape 1 : Alimentation du bus d'entrées analogiques

- Connecter la **masse (COM1)** du bus d'entrées analogiques à la **masse générale**

Étape 2 : Connexion de l'alimentation de laboratoire

- Connecter le **+** de l'alimentation de laboratoire sur la borne **AI0**
- Connecter le **-** de l'alimentation de laboratoire sur la borne **COM1**

Étape 3 : Référence de masse commune

- Relier la masse **COM1** à la **masse de l'alimentation de l'automate** (borne M)

Important : Cette connexion garantit une référence de potentiel commune entre l'alimentation de laboratoire et l'automate, indispensable pour une mesure correcte.

Récapitulatif des connexions

Source	Borne automate	Description
Alimentation lab (+)	AI0	Signal analogique 0-10V
Alimentation lab (-)	COM1	Masse entrées analogiques
COM1	M (masse automate)	Référence de potentiel commune

TABLE 6.6 – Tableau de câblage de l'entrée analogique

6.4.6 Programmation dans UniLogic

Création du projet

1. Créer un nouveau projet : `TP4_Lecture_Analogique`
2. Sélectionner le modèle : `US5-B10-T42`

Configuration de l'entrée analogique

Dans IO → On board IO → Analog Inputs :

- Analog Inputs_0 (AI0) → Configuration :
 - Type : Voltage 0-10V
 - Alias : analogique

Création des Tags

Dans Global Tags, créer les variables suivantes :

Nom	Type	Valeur initiale	Description
analogique	INT16	0	Valeur brute (0-4095)
forma	INT16	10	Plage max en Volts
val_max	INT16	4095	Valeur numérique max
mul	INT16	0	Variable intermédiaire
tens	INT16	0	Tension convertie (en mV)

TABLE 6.7 – Liste des Tags nécessaires

Programmation Ladder

Aller dans Ladder → Module1 → Function1 et programmer la conversion.

Formule de conversion

$$tens = \frac{analogique \times forma}{val_max}$$

Cette formule convertit la valeur brute (0-4095) en tension (0-10V, exprimée en unités pour l'affichage).

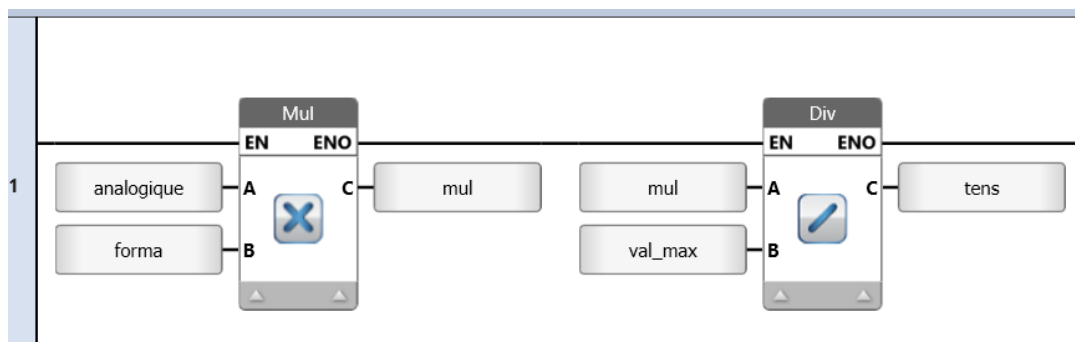


FIGURE 6.11 – Programme Ladder de conversion analogique

Description du programme

1. Bloc Mul (Multiplication) :

- Entrée A : analogique (valeur brute AI0)
- Entrée B : forma (valeur 10)
- Sortie C : mul (résultat intermédiaire)

2. Bloc Div (Division) :

- Entrée A : `mul` (résultat de la multiplication)
- Entrée B : `val_max` (valeur 4095)
- Sortie C : `tens` (tension finale)

Le résultat `tens` représente la tension mesurée et peut être affiché sur l'IHM.

Création de l'IHM

L'interface HMI permet d'afficher en temps réel la valeur brute de l'entrée analogique ainsi que la tension convertie.

Conception de l'écran Aller dans HMI → Module1 → Screen1 et créer l'interface suivante :

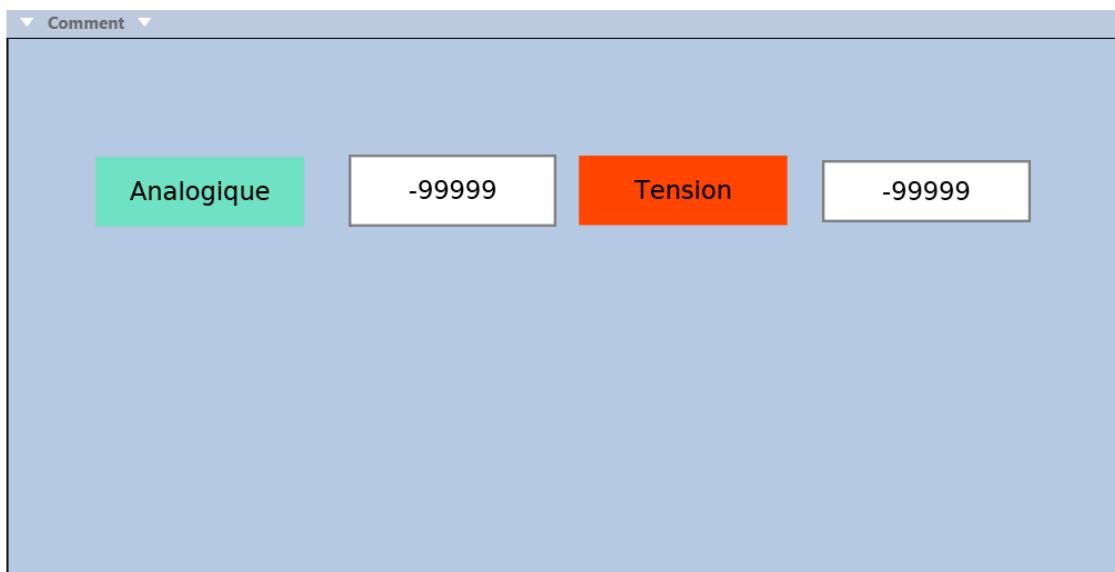


FIGURE 6.12 – Interface HMI pour l'affichage de la valeur analogique

Éléments de l'interface L'écran comporte quatre éléments disposés horizontalement :

1. Label "Analogique" (fond vert clair) :

- Type : Text Element
- Texte : "Analogique"
- Rôle : Identifie l'affichage de la valeur brute

2. Afficheur valeur brute (fond blanc) :

- Type : Numeric Box
- Tag lié : analogique
- Format : Décimal, 0 décimale
- Plage : 0 - 4095
- Lecture seule

3. Label "Tension" (fond orange) :

- Type : Text Element

- Texte : "Tension"
 - Rôle : Identifie l'affichage de la tension convertie
4. **Afficheur tension** (fond blanc) :
- Type : Numeric Box
 - Tag lié : `tens`
 - Format : Décimal, 2 décimales (pour affichage en Volts)
 - Lecture seule

Configuration des afficheurs numériques Pour chaque Numeric Box, dans la fenêtre *Properties* :

- Tag : Link : Sélectionner le Tag correspondant (`analogique` ou `tens`)
- Format : Décimal
- Read Only : Coché (lecture seule)

6.4.7 Test et validation

Procédure de test

1. Connecter l'automate au PC via USB
2. Compiler le projet : `Build` → `Verify`
3. Téléverser le programme : `PLC` → `Download All`
4. Régler l'alimentation de laboratoire sur différentes tensions (0V, 2.5V, 5V, 7.5V, 10V)
5. Observer les valeurs affichées sur l'IHM

Résultats attendus

Tension appliquée (V)	Valeur brute attendue	Tension affichée (V)
0.00	0	0.00
2.50	1024	2.50
5.00	2048	5.00
7.50	3072	7.50
10.00	4095	10.00

TABLE 6.8 – Résultats attendus pour différentes tensions

Note : De légères variations peuvent apparaître en fonction de la précision de l'alimentation de laboratoire et de la résolution de l'entrée analogique.

6.5 Exercice 5 – Entrée rapide (encodeur)

Chapitre 7

Diagnostic et dépannage

7.1 Outil UniApps

7.2 Erreurs courantes

Chapitre 8

Conclusion