

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة  
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا  
Faculté de Technologie

قسم الإلكترونيك  
Département d'électronique



## Mémoire de Master

Filière Automatique  
Spécialité Automatique et Informatique Industrielle

présenté par

Lakehal Hamza

&

Berrekia Youcef Abdelhamid

# Développement d'un système contrôle-commande et acquisition des données à base de labview

Proposé par : Bennila Nour-Eddine

Année Universitaire 2018-2019

## Remerciements

---

Nous commençons par rendre grâce à ALLAH qui nous a guidé sur le droit chemin tout au long du travail, nous a inspiré les bons pas et les justes reflexes et nous a donné la capacité d'écrire et réfléchir, la force, la patience d'aller jusqu'au bout. Sans sa miséricorde, ce travail n'aura pas abouti. Merci Allah

Il nous serait impossible de citer nommément toutes les personnes qui nous aidé, encouragé et soutenu afin que ce travail puisse voir le jour. Que toutes ces personnes trouvent ici l'expression de notre sincère reconnaissance.

Ces remerciements vont tout d'abord au corps professoral et administratif de la Faculté (science et technologie département électronique) des Sciences Economiques, Juridiques et sociales, pour la richesse et la qualité de leur enseignement et qui déploient de grands efforts pour assurer à leurs étudiants une formation actualisée.

La première personne que nous tenons à remercier est notre promoteur & encadrant Mr.BENNILA, pour l'orientation, la confiance, la patience qui a constitué un apport considérable sans lequel ce travail n'aurait pas pu être menée au bon port. Qu'il trouve dans ce travail un hommage vivant à sa haute personnalité.

On n'oublie pas nos parents pour leur contribution, leur soutien et leur patience. Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours encouragées au cours de la réalisation de ce mémoire. Merci à tous et à toutes.

*Je dédie ce travail :*

*Qui n'aura jamais pu voir le jour sans les soutiens indéfectibles et sans limite de*

*Mes chers parents qui ne cessent de me donner*

*Avec amour le nécessaire pour que je puisse arriver à ce que je suis aujourd'hui.*

*J'espère que Dieu vous gardera en sécurité et que le succès sera toujours à portée de main afin que je puisse vous garder heureux.*

*Je dédie aussi ce travail:*

A ma chère sœur pour ses encouragements constants et son soutien moral.

A mon cher frère ABDOU et ma petite famille.

A mon binôme **YOUCEF** et toute la famille **BERREKIA**.

*Je dédie ce modeste travail :*

*A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études, que dieu leur procure bonne santé et longue vie.*

*A toute ma famille pour leur soutien tout au long de mon parcours universitaire, Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infailible, et Merci d'être toujours là pour moi.*

*A mon binôme **HAMZA** et toute la famille **LAKEHAL**.*

---

ملخص: في هذا المشروع، نقوم بتطوير منصة برمجية تستند إلى LABVIEW، وأجهزة تعتمد على الأدوات الوطنية وأجهزة سيمنز، لرصد أنشطة شبكة API والحصول على البيانات. نحن نستخدم DAQ-NI-USB-6001، الذي يتكون من: ثمانية مداخل و مخرجين لإشارات قياسية و ثلاثة عشر مدخل/مخرج رقمي. من بين هذه الإشارات القياسية الثمانية نستخدم مدخلين لإشارات الفارقة لقياس درجة الحرارة و الضغط وواحدة لإشارات المرجعية بهدف تغيير سرعة محرك. كذلك نستعمل مخرج واحد من المخرجين في تغيير السرعة لكن عن طريق جهاز الكمبيوتر. أما فيما يخص الإشارات الرقمية نستعمل واحدة كمدخل رقمي لتوقف المحرك بطريقة آمنة و مخرج رقمي لتحكم الآلي و البعدي في تشغيل و توقيف المحرك . و إضافة علي هذا نقوم بربط ثلاثة أجهزة التحكم المنطقي قابلة للبرمجة siemens S7-312c بواسطة شبكة MPI وإنشاء واجهة الإيفيو مع هذه الأجهزة بواسطة serveur OPC .

كلمات المفاتيح: الابطفيو, MPI, siemens S7-312c, serveur OPC,

---

**Résumé :** nous développons une plateforme logicielle à base de LABVIEW, et matériel à base de national instrument et siemens, pour l'acquisition des données, Nous utilisons le DAQ-NI-USB-6001. Sur les 08 entrées nous utilisons deux différentielles pour la mesure de température et pression et une référentielle pour la variation de vitesse, sur les 02 sorties nous utilisons une pour la variation de vitesse, sur les 13E/S digitales nous utilisons une entrée pour l'arrêt du moteur en sécurité et une sortie pour le démarrage du moteur. Et nous connectons 3 automates programmables siemens S7-312c par un réseau MPI Et lié ces automates avec une interface labview via le serveur OPC.

**Mots clés :** LabVIEW, DAQ-NI-USB-6001, Serveur OPC, siemens S7-312c, profibus

---

**Abstract: we** are developing a software platform based on LABVIEW, and hardware based on national instrument and siemens equipment, for data acquisition. We use the DAQ-NI-USB-6001, On the 08 inputs we use two differentials for the measurement of temperature and pressure and a reference for the variation of speed, on the 02 outputs we use one for the variation of speed, on the digital 13E / S we use an entry for the shutdown of the engine in safety mode and an output for starting the engine. And we connect 3 programmable logic controller siemens S7-312c by MPI network, and interface LabVIEW with PLCs using the OPC server.

**Keywords:** LabVIEW, DAQ-NI-USB-6001, S7-312c, OPC Server, MPI

---

## Listes des acronymes et abréviations

**SCADA:** Supervisory Control and Data Acquisition

**DCS:** Distributed control system

**NI:** National instrument

**TIA:** Totally Integrated Automation

**API:** Automate programmable industrial

**VI:** Virtual Instrument

**OPC:** OLE for Process Control

**DA:** Data Access

**HDA:** historisation des données

**UA:** Unified architecture

**MPI:** multi point interface

**IHM:** Interfaces Homme Machine

**DAQ:** Data Acquisition

**AI:** Analog Input

**AO:** Analog Output

**DI:** Digital Input

**DO:** Digital Output

**RSE:** Référentielle

**Diff:** différentielle

**GND:** Ground

**PFI:** Programmable function interface

**SW:** switch

**E/S:** Entrées/Sorties

**USB:** Universel Serial Bus

## Table des matières

Introduction générale .....	1
Chapitre 1 : Les systèmes de contrôle- commande industriels .....	3
1.1 Les automates programmables.....	3
1.1.1 Introduction .....	3
1.1.2 Pourquoi l'automatisation.....	3
1.1.3 Structure générale des API.....	4
1.1.4 Structure interne d'un automate programmable industriel.....	5
1.1.5 fonctionnement.....	6
1.1.6 description des éléments d'un API.....	7
1.1.7 jeu d'instruction.....	9
1.2 Les systèmes de contrôle-commande.....	10
1.2.1 Introduction.....	10
1.2.2 Acquisition des données.....	10
1.3 superviseurs industriels.....	14
1.3.1 introduction.....	14
1.3.2 systèmes SCADA.....	14
1.3.3 système DCS.....	16
1.4 les différents constructeurs des superviseurs.....	19
1.5 conclusion.....	24
Chapitre 2 : développement et réalisation.....	25
2.1 Introduction.....	25
2.2 Système d'acquisition de données.....	25
2.2.1 E/S numérique.....	26
2.2.2 Entrées analogique .....	27
2.2.3 Sortie analogique.....	28
2.2.4 Brochage et description des signaux.....	29
2.2.5 Utilisation des entrées digitale DI.....	31
2.2.6 Utilisation des sorties digitale DO.....	35
2.2.7 Utilisation des sorties analogique AO.....	43
2.2.8 Utilisation des entrees analogiques Ai.....	44
2.3 SCADA labview PLC5-312c.....	50
2.3.1 Introduction .....	50
	51
	52

2.3.2	Les feux de circulation.....
2.3.3	Variation de vitesse d'un moteur .....
2.3.4	Le chariot .....
2.4	Communication entre 3 API s7-312c.....
2.4.1	Introduction.....
2.4.2	Bus de terrain profibus .....
2.4.3	Branchement.....
2.5	Conclusion .....
Chapitre 3 programmation et résultats obtenus .....	
3.1	Introduction .....
3.2	Système d'aquisition.....
3.2.1	NI-DAQmx.....
3.2.2	Configuration.....
3.2.3	Programmation sur labview .....
3.3	SCADA labiew.....
3.3.1	Programmation sur le TIA PORTAL.....
3.3.2	Transmission des variables .....
3.4	Conclusion .....
Conclusion générale.....	

## Liste des figures

Figure 1.1. Automate siemens s5-95u.....	4
Figure 1.2. Aspect extérieur d'un automate s7-200 cpu222.....	4
Figure 1.3. Structure interne d'un automate programmable industriel API.....	6
Figure 1.4. Fonctionnement cyclique d'un API.....	6
Figure 1.5. Temps de scrutation vs temps de repensé.....	7
Figure 1.6. La mémoire.....	7
Figure 1.7. Symboles usuels en langage LD.....	10
Figure 1. 8. Fenêtre "face avant" & "diagramme.....	13
Figure 1.9. Architecteur du SCADA.....	15
Figure 1.10. National instruments.....	19
Figure 1.11. Siemens.....	20
Figure 1.12. TIA PORTAL.....	21
Figure 2.1. NI USB -6001.....	25
Figure 2.2. Les canaux de NI USB 6001.....	26
Figure 2.3. Diagramme de bloc NI USB 6001.....	26
Figure 2.4. Circuit de entres analogique .....	27
Figure 2.5. Circuit de sortie analogique .....	28
Figure 2.6. NI USB 6001.....	29
Figure 2.7. Convertisseur de huit canaux 24v vers 5v.....	31
Figure 2.8. Schéma électrique de convertisseur de 8 canaux.....	31
Figure 2.9. Détecteur photoélectrique.....	32
Figure 2.10. Photo cellule FAR2/BP-OE.....	33
Figure 2.11. Faisceau renvoyé par la cible.....	33
Figure 2.12. Principe du système proximité.....	33
Figure 2.13. Détection avec réglage d'effacement d'arrière-photo coupleur.....	34
Figure 2.14. Les détections photoélectriques.....	34
Figure 2.15. Câblage de la photocellule avec DIO.....	34
Figure 2.16. Schéma de fonctionnement d'un relais électromécanique.....	35
Figure 2.17. Moteur asynchrone.....	36
Figure 2.18. Couplage étoile.....	37
Figure 2.19. Couplage triangle.....	38

Figure 2.20. OMRON VS mini j7.....	38
Figure 2.21 .nomenclature d'un variateur de vitesse.....	39
Figure 2.22. Console numérique du variateur de vitesse.....	39
Figure 2.23. Bornes du circuit de contrôle .....	39
Figure 2.24. Câblage de démarrage par bouton .....	42
Figure 2.25. Relais .....	43
Figure 2.26. Câblage pour la variation de vitesse de moteur.....	44
Figure 2.27. La commande du moteur sur la face avant.....	44
Figure 2.28. Thermocouple.....	45
Figure 2.29. Câblage de thermocouple.....	47
Figure 2.30. Affichage de mesure de température sur la face avant .....	48
Figure 2.31. Détection de pression .....	48
Figure 2.32. Câblage du capteur de pression.....	49
Figure 2.33. Affichage de mesure de pression sur la face avant.....	50
Figure 2.34. API SIEMENS.....	51
Figure 2.35. Maquette de feu rouge.....	51
Figure 2.36. GRAFCET de feu de circulation.....	52
Figure 2.37. Potentiomètre.....	53
Figure 2.38. Schéma de fonctionnement de AI.....	53
Figure 2.39. Schéma de branchement de potentiomètre avec AI.....	53
Figure 2.40. Affichage et commande de chariot .....	54
Figure 2.41. Montage de chariot.....	54
Figure 2.42. Le chariot.....	55
Figure 2.43. profibus.....	56
Figure 2.44. Connecteur de profibus .....	56
Figure 2.45. Débit et longueur de segment .....	57
Figure 2.46. Branchement de 3 PLC .....	57
Figure 3.1. DAQ Assistant.....	59
Figure 3.2. Création d'un canal pour DI0.....	60
Figure 3.3. Canal physique DI0anal .....	60
Figure 3.4. Configuration d'un de photocellule.....	61
Figure 3.5. Création d'un canal virtuel NI-DAQmx.....	61

Figure 3.6. Canaux physique du périphérique.....	62
Figure 3.7. visualisation du brochage .....	62
Figure 3.8. Configuration d'un canal de thermocouple dans max.....	63
Figure 3.9. Canal de pression .....	63
Figure 3.10. Création d'un canal voltage output.....	64
Figure 3.11. Canal physique ao0.....	64
Figure 3.12. Configuration d'un canal de ao0.....	65
Figure 3.13.programmation des entrées/sorties digitales .....	65
Figure 3.14. Programmation des entrées analogique .....	66
Figure 3.15. Programmation des sorties analogique .....	66
Figure 3.16. GRAFCET de feu de circulation.....	68
Figure 3.17. Table de variation .....	68
Figure 3.18. Programmation de l'API.....	69
Figure 3.19. La valeur renvoyée par le potentiomètre.....	74
Figure 3.20. Les variables d'un bloc scal .....	75
Figure 3.21. IBH OPC Editeur.....	77
Figure 3.22. Editeur OPC.....	77
Figure 3.23. Nouveau PLC.....	78
Figure 3.24. Propriétés de PLC.....	78
Figure 3.25. OPC Editeur Simatic S7.....	78
Figure 3.26. Simatic Net Settings.....	79
Figure 3.27. Interface PG/PC.....	79
Figure 3.28. Paramètre de PG/PC.....	80
Figure 3.29. Configuration des variables.....	80
Figure 3.30. Transfer des variables au serveur OPC IBH.....	81
Figure 3.31. Configuration sur labview .....	81
Figure 3.32. Paramètres de variables .....	82
Figure 3.33. Variables .....	82
Figure 3.34. Fin de configuration .....	83
Figure 3.35. Interconnexion entre 3 PLC .....	83
Figure 1. Vue du Portail .....	85
Figure 2. Vue du Portail .....	86

Figure 3. Création d'un projet.....	87
Figure 4. Configuration et paramétrage du matériel .....	88
Figure 5. Configuration et paramétrage du matériel .....	88
Figure 6. Adressage des E/S.....	89
Figure 7. Memento de cadence .....	90
Figure 8. Adresse Ethernet de la CPU .....	90
Figure 9. Compilation et chargement de la configuration de la CPU .....	91
Figure 10. Compilation et chargement de la configuration de la CPU .....	91
Figure 11. Compilation et chargement de la configuration de la CPU .....	92
Figure 12. Compilation et chargement de la configuration de la CPU .....	92
Figure 13. Création d'un canal pour DO0 .....	93
Figure 14. Canal physique DO0 .....	93
Figure 15. Configuration d'un canal de démarrage.....	94

## Liste des tableaux

Tableau 2.1. description des signaux .....	29
Tableau 2.2. Bornes de circuit du contrôle .....	39
Tableau 2.3. L'utilisation de l'opération numérique.....	41
Tableau 3.1. Paramètres du bloc scale.....	75

# Introduction générale

---

Un système de supervision scada est un système de contrôle et commande permettant de superviser et de prendre en main un système industriel complet à distance. Avec un système de supervision scada, on peut donc simuler un système physique en temps réel via des PCs de contrôle ou des panels de supervision industriels.

Les systèmes SCADA existent dans presque toutes les industries, que ce soit dans l'industrie pétrolière, la production d'eau potable, le traitement des eaux usées, les cimenteries, les usines de production agroalimentaire, etc.

En fonction de cela, nous avons concevoir un mini SCADA qui contient le système de commande et de contrôler et acquisition des données à base de LabVIEW et communication entre 3 automate programmable industriel S7-312c à travers le réseau MPI.

LabVIEW est un environnement de développement complet, graphique, compilé et particulièrement bien adapté au domaine de l'acquisition et de la mesure. Son approche totalement graphique offre une souplesse et une dimension intuitive inégalée.

Comparativement aux langages textuels il offre la même puissance de programmation mais sans le coté abstrait et complexe lié à la syntaxe.

En lisant ce mémoire, vous pourriez avoir le sentiment que LabVIEW est un outil non seulement dédié à la mesure, mais il peut être utilisé pour superviser le déroulement d'un processus industriel.

Pour faire communiquer entre 3 automates programmables industriels S7-312c Siemens, on a besoin de réseau d'interface. Comme réseau d'interface, on peut avoir le MPI, le Profibus ou le Profinet. Dans notre cas nous avons utilisé le réseau MPI pour communiquer entre 3 CPU.

# Introduction générale

Pour la bonne compréhension du travail que nous avons développé au cours de ce projet nous avons organisé ce mémoire en trois chapitres.

- Le premier chapitre présente les concepts et l'intérêt des systèmes de contrôle-commande industriels et les outils de programmation que nous avons utilisé tout au long de ce projet, à savoir le TIA Portal de Siemens, LabVIEW de National Instrument et OPC server d'IBH.
- Le chapitre deux, nous avons développé les détails de la démarche suivie pour la conception et la réalisation d'un mini SCADA , et la ligne de communication entre LabVIEW et l'API S7-312C de Siemens et aussi la communication entre 3 l'API.
- le dernier chapitre, nous avons développé tous les programmes pour faire la supervision et le contrôle au niveau de LabVIEW et au niveau de TIA PORTAL.

# Chapitre 1 : les systèmes de contrôle-commande industriels

---

## 1.1 Les automates programmables

### 1.1.1 Introduction

Les **Automates Programmables Industriels (API)** sont apparus aux Etats-Unis vers 1969 où ils répondaient aux désirs des industries de l'automobile de développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des techniques et des modèles fabriqués.

Un Automate Programmable Industriel (**API**) est une machine électronique programmable par un personnel pas nécessairement informaticien et destiné à piloter en ambiance industrielle et en temps réel des procédés industriels. Un **automate programmable** est adaptable à un maximum d'application, d'un point de vue traitement, composants, langage. C'est pour cela qu'il est de construction modulaire.

Il est en général manipulé par un personnel électromécanicien. Le développement de l'industrie à entraîner une augmentation constante des fonctions électroniques présentes dans un automatisme c'est pour ça que l'API s'est substitué aux armoires à relais en raison de sa souplesse dans la mise en œuvre, mais aussi parce que dans les coûts de câblage et de maintenance devenaient trop élevés [1].

### 1.1.2 Pourquoi l'automatisation ?

L'automatisation permet d'apporter des éléments supplémentaires à la valeur ajoutée par le système. Ces éléments sont exprimables en termes d'objectifs par :

- Accroître la productivité (rentabilité, compétitivité) du système ;
- Améliorer la flexibilité de production ;
- Améliorer la qualité du produit ;
- Adaptation à des contextes particuliers tels que les environnements hostiles pour l'homme (milieu toxique, dangereux.. nucléaire...), adaptation à des tâches ;

- physiques ou intellectuelles pénibles pour l'homme (manipulation de lourdes charges, tâches répétitives parallélisées...);
- Augmenter la sécurité, etc [1].

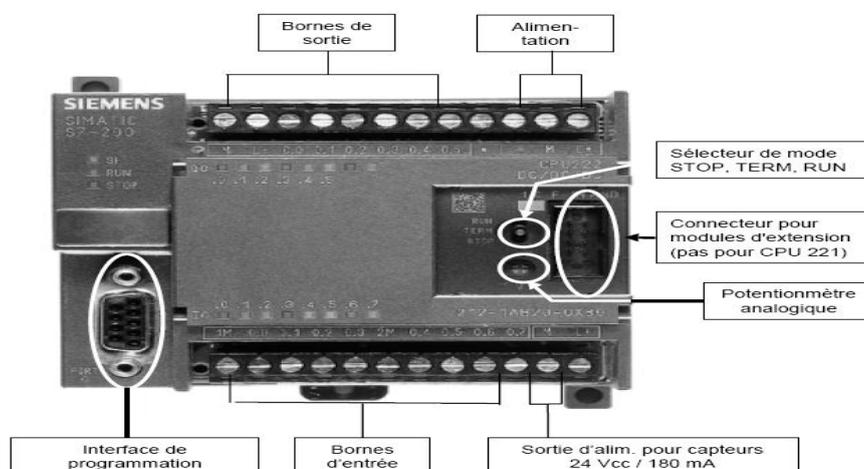


**Figure 1.1.** Automate SIEMENS S5-95U

### **1.1.3 Structure générale des API**

Les caractéristiques principales d'un automate programmable industriel(API) sont : coffret, rack, baie ou cartes.( Figure 1.2)

- Compact ou modulaire ;
- Tension d'alimentation ;
- Taille mémoire ;
- Sauvegarde (EPROM, EEPROM, pile, ...);
- Nombre d'entrées / sorties ;
- Modules complémentaires (analogique, communication,..) ;
- Langage de programmation ;



**Figure 1.2.** Aspect extérieur d'un automate S7-200 CPU222

Des API en boîtier étanche sont utilisées pour les ambiances difficiles (température, poussière, risque de projection ...) supportant ainsi une large gamme de température, humidité ... L'environnement industriel se présente sous trois formes :

- environnement physique et mécanique (poussières, température, humidité, vibrations);
- pollution chimique ;
- perturbation électrique. (parasites électromagnétiques) [1].

#### **1.1.4 Structure interne d'un automate programmable industriel**

Les API comportent quatre principales parties (figure1.3):

- Une unité de traitement (un processeur CPU);
- Une mémoire ;
- Des modules d'entrées-sorties ;
- Des interfaces d'entrées-sorties ;
- Une alimentation 230 V, 50/60 Hz (AC) - 24 V (DC).

La structure interne d'un **automate programmable industriel** (API) est assez voisine de celle d'un système informatique simple, L'unité centrale est le regroupement du processeur et de la mémoire centrale. Elle commande l'interprétation et l'exécution des instructions programme.

Les instructions sont effectuées les unes après les autres, séquencées par une horloge.

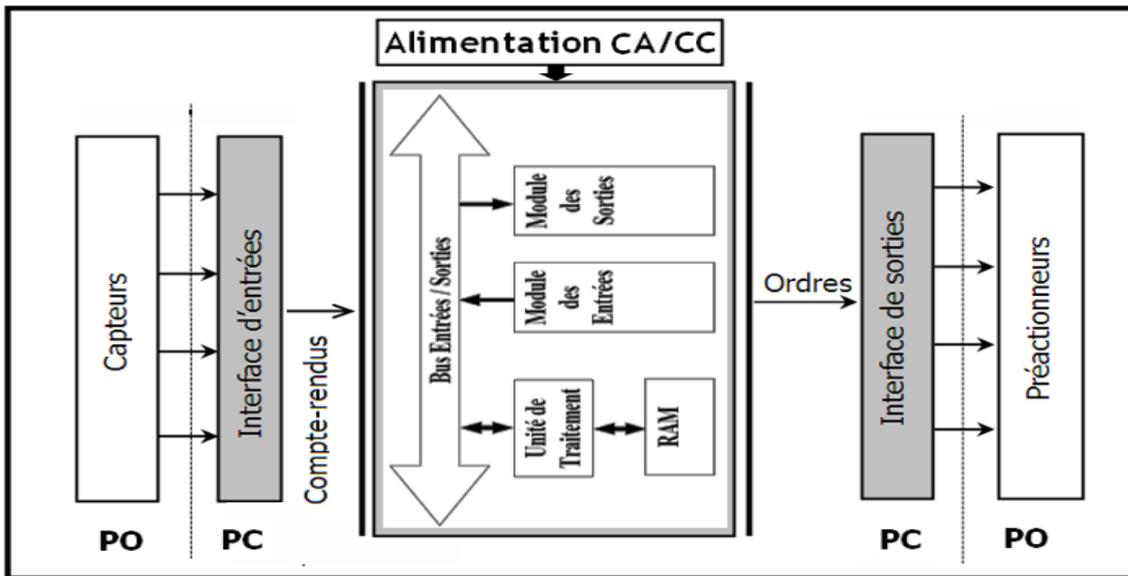
##### **Deux types de mémoire cohabitent :**

- **La mémoire Programme**

Où est stocké le langage de programmation. Elle est en général figée, c'est à dire en lecture seulement. (ROM : mémoire morte)

- **La mémoire de données**

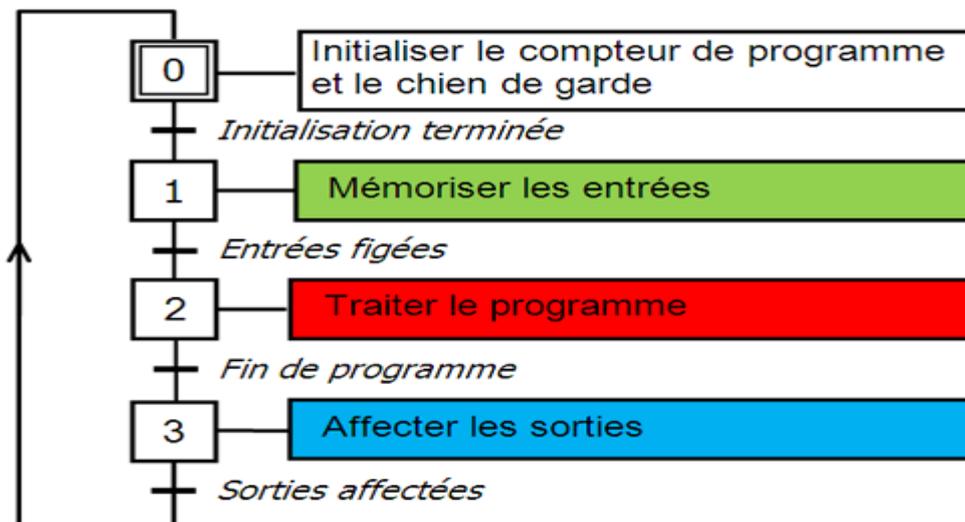
Utilisable en lecture-écriture pendant le fonctionnement c'est la RAM (mémoire vive). Elle fait partie du système entrées-sorties. Elle fige les valeurs (0 ou 1) présentes sur les lignes d'entrées, à chaque prise en compte cyclique de celle-ci, elle mémorise les valeurs calculées à placer sur les sorties [1].



**Figure 1.3.** Structure interne d'un automate programmable industriel (API)

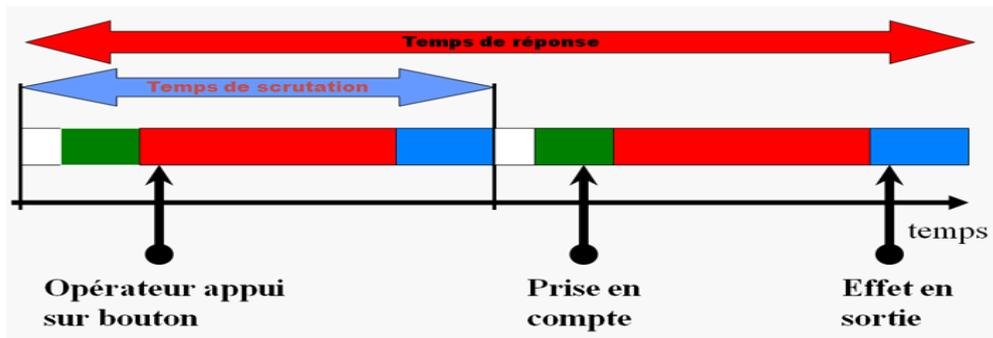
### 1.1.5 Fonctionnement

L'automate programmable **reçoit** les informations relatives à l'état du système et puis **commande** les pré-actionneurs suivant le programme inscrit dans sa mémoire. Généralement les automates programmables industriels ont un fonctionnement cyclique (Figure 1.4). Le **microprocesseur** réalise toutes les fonctions logiques ET, OU, les fonctions de temporisation, de comptage, de calcul... Il est connecté aux autres éléments (mémoire et interface E/S) par des liaisons **parallèles** appelées '**BUS**' qui véhiculent les informations sous forme binaire. Lorsque le fonctionnement est dit synchrone par rapport aux entrées et aux sorties, le cycle de traitement commence par la prise en compte des entrées qui sont figées en mémoire pour tout le cycle. [1]



**Figure1.4.** Fonctionnement cyclique d'un API

Le processeur exécute alors le programme instruction par instruction en rangeant à chaque fois les résultats en mémoire. En fin de cycle les sorties sont affectées d'un état binaire, par mise en communication avec les mémoires correspondantes. Dans ce cas, le temps de réponse à une variation d'état d'une entrée peut être compris entre un ou deux temps de cycle (durée moyenne d'un temps de cycle est de 5 à 15 ms Figure 1.5) [1].



**Figure 1.5.** Temps de scrutation vs Temps de réponse

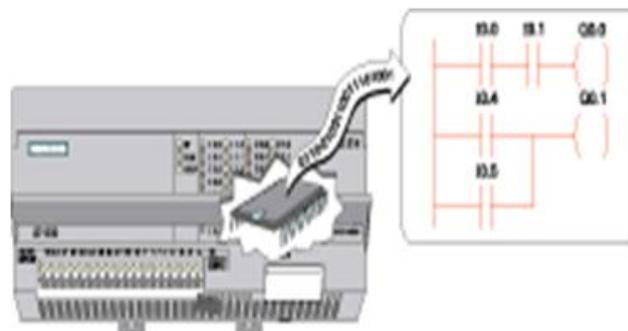
**Il existe d'autres modes de fonctionnement, moins courants :**

- synchrone par rapport aux entrées seulement ;
- asynchrone ;

### **1.1.6 Description des éléments d'un API**

#### **a. La mémoire :**

Elle est conçue pour recevoir, gérer, stocker des informations issues des différents secteurs du système que sont le terminal de programmation (PC ou console) et le processeur, qui lui gère et exécute le programme. Elle reçoit également des informations en provenance des capteurs [1].



**Figure 1.6.** La mémoire

Il existe dans les automates deux types de mémoires qui remplissent des fonctions différentes:

- **La mémoire Langage :**

Où est stocké le langage de programmation. Elle est en général Figée, c'est-à-dire en lecture (ROM : mémoire morte).

- **La mémoire Travail :**

Utilisable en lecture-écriture pendant le fonctionnement c'est la RAM (mémoire vive). Elle s'efface automatiquement à l'arrêt de l'automate (nécessite une batterie de sauvegarde).

**Répartition des zones mémoires :**

- Table image des entrées ;
- Table image des sorties ;
- Mémoire des bits internes ;
- Mémoire programme d'application ;

***b. Le processeur :***

Son rôle consiste d'une part à organiser les différentes relations entre la zone mémoire et les interfaces d'entrées et de sorties et d'autre part à exécuter les instructions du programme.

***c. Les interfaces et les cartes d'Entrées / Sorties :***

L'interface d'entrée comporte des adresses d'entrée. Chaque capteur est relié à une de ces adresses. L'interface de sortie comporte de la même façon des adresses de sortie. Chaque pré actionneur est relié à une de ces adresses. Le nombre de ces entrées est sorties varie suivant le type d'automate. Les cartes d'E/S ont une modularité de 8, 16 ou 32 voies. Les tensions disponibles sont normalisées (24, 48, 110 ou 230V continu ou alternatif ...).

**-Cartes d'entrées :**

Elles sont destinées à recevoir l'information en provenance des capteurs et adapter le signal en le mettant en forme, en éliminant les parasites et en isolant électriquement l'unité de commande de la partie opérative.

**-Cartes de sorties :**

Elles sont destinées à commander les pré-actionneurs et éléments des signalisations du système et adapter les niveaux de tensions de l'unité de commande à celle de la partie opérative du système en garantissant une isolation galvanique entre ces dernières.

**d. Exemple des cartes :**

- **Cartes de comptage rapide** : elles permettent d'acquérir des informations de fréquences élevées incompatibles avec le temps de traitement de l'automate. (Signal issu d'un codeur de position.
- **Cartes de commande d'axe** : Elles permettent d'assurer le positionnement avec précision d'élément mécanique selon un ou plusieurs axes. La carte permet par exemple de piloter un servomoteur et de recevoir les informations de positionnement par un codeur. L'asservissement de position pouvant être réalisé en boucle fermée.
- **Cartes d'entrées / sorties analogiques** : Elles permettent de réaliser l'acquisition d'un signal analogique et sa conversion numérique (CAN) indispensable pour assurer un traitement par le microprocesseur. La fonction inverse (sortie analogique) est également réalisée. Les grandeurs analogiques sont normalisées : 0-10V ou 4-20mA.
- **Cartes de régulation PID ;**
- **Cartes de pesage ;**
- **Cartes de communication (RS485, Ethernet ...)** ;
- **Cartes d'entrées / sorties déportées ;**

**e. L'alimentation électrique :**

Tous les automates actuels sont équipés d'une alimentation 240 V 50/60 Hz, 24 V DC. Les entrées sont en 24 V DC et une mise à la terre doit également être prévue.

**1.1.7 Jeu d'instructions**

Le processeur peut exécuter un certain nombre d'opérations logiques; l'ensemble des instructions booléennes des instructions complémentaires de gestion de programme (saut, mémorisation, adressage ...) constitue un jeu d'instructions [1].

Chaque automate possède ses propres jeux d'instructions. Mais par contre, les constructeurs proposent tous une interface logicielle de programmation répondant à la norme CEI1131-3. Cette norme définit cinq langages de programmation utilisables, qui sont :

- **Les langages graphiques:**
  - **LD** : **Ladder Diagram** (Diagrammes échelle) ;
  - **FBD** : **Function Block Diagram** (Logigrammes) ;

- **SFC** : Sequential Function Chart ( Grafcet);
- **Les langages textuels** :
  - **IL** : Instruction List (Liste d'instructions) ;
  - **ST** : Structured Text (Texte structure);

Le langage à relais (Ladder Diagram) est basé sur un symbolisme très proche de celui utilisé pour les schémas de câblage classiques. Les symboles les plus utilisés sont donnés au tableau suivant :

Fonction	Symbole	
	Européen	Américain
Contact ouvert au repos	---o o---	
Contact fermé au repos	---o/o---	
Début de branchement		
Fin de branchement		
Affectation	---( )---	---( )

*Figure 1.7.* Symboles usuels en langages LD

## 1.2 Les systèmes de contrôle – commande

### 1.2.1 Introduction

Les systèmes de contrôle-commande sont quasi indispensables sur toute ligne de production automatisée moderne. Il est presque impensable de gérer des installations industrielles qui couvrent des hectares et des hectares de superficie sans systèmes de supervision et de commande à distance [2].

### 1.2.2 Acquisitions des données

L'acquisition de données est le processus d'échantillonnage des signaux qui mesurent les conditions physiques du monde réel et de convertir les échantillons résultants en valeurs numériques qui peuvent être manipulés par un ordinateur. Les systèmes d'acquisition de données, abrégés par les acronymes DAS ou DAQ, convertissent typiquement des formes d'onde analogiques en valeurs numériques pour le traitement. Les composants des systèmes d'acquisition de données comprennent :

- Capteurs, pour convertir des paramètres physiques en signaux électriques ;

- Circuit de conditionnement de signal, pour convertir les signaux de capteur en une forme qui peut être convertie en valeurs numériques ;
- Convertisseurs analogique-numérique, pour convertir les signaux des capteurs conditionnés en valeurs numériques ;

Les applications d'acquisition de données sont généralement contrôlées par des logiciels développés à l'aide de langages de programmation à usage général tels que Assembleur, BASIC, C, C ++, C #, Fortran, Java, LabVIEW, Lisp, Pascal, etc.

Il existe également des logiciels libres qui fournissent tous les outils nécessaires pour acquérir des données à partir d'équipements matériels différents. Ces outils viennent de la communauté scientifique où l'expérience complexe nécessite des logiciels rapides, souples et adaptables. Ces paquets sont habituellement personnalisés, mais un paquet DAQ plus général comme le système d'acquisition de données maximum intégré peut être facilement adapté et est utilisé dans plusieurs expériences de physique dans le monde entier [3].

#### **a. LabVIEW**

LabVIEW est un environnement de développement complet, graphique, compilé et particulièrement bien adapté au domaine de l'acquisition et de mesure, son approche totalement graphique offre une souplesse et une dimension intuitive inégalée. Comparativement aux langages textuels il offre la même puissance de programmation mais sans le côté abstrait lié à syntaxe.

Ce langage s'adresse à tous les techniciens ou ingénieurs qui désirent développer une application de mesures, de tests, de supervision ou de contrôle /commande avec une interface utilisateur de très grande interactivité. Mais plus généralement à tous les développeurs qui désirent utiliser un langage, puissant et ouvert sur la totalité du génie logiciel.

Le LabVIEW étend les avantages de la programmation graphique au développement de SCADA (Contrôle de supervision et acquisition de données) ou aux applications d'enregistrement de données à grand nombre de canaux. Utilisez les outils pour communiquer avec les automates programmables conventionnels (API), gérer les données dans les bases de données, gérer les alarmes et les événements et créer des IHM (Interfaces Homme Machine) [4].

### **- Concepts de base de la programmation graphique**

La programmation à l'aide d'un langage graphique date des années 1980. Les langages de programmation textuels conventionnels impliquent un mode de conception séquentiel contraint par l'architecture même de la machine. La représentation textuelle est moins performante puisqu'elle ne permet qu'une représentation séquentielle de l'information. D'ailleurs, les efforts mis en œuvre pour rendre un texte plus rapidement compréhensible reposent sur des moyens graphiques. Lorsque des objets peuvent être manipulés graphiquement, la communication homme/machine est grandement facilitée. Le but de la programmation graphique est donc de faciliter la mise en œuvre d'applications informatiques [4].

### **- Avantages**

- Facilité d'utilisation par les non-programmeurs : les images sont en général préférées aux mots, la programmation devient alors plus intuitive ;
- La sémantique des images est plus puissante que celle des mots ;
- Les images ne sont pas sujettes aux barrières des langues.

Il est évident que toute méthode a ses inconvénients. Ainsi, les principaux problèmes liés à la programmation graphique sont les suivants :

- Difficulté de visualisation pour les programmes de taille importante nécessitant une architecture modulaire et hiérarchique ;
- Ambiguïté dans l'interprétation du graphisme ;
- Nécessité de disposer d'un environnement de développement efficace [4].

### **- Environnement de la programmation LabVIEW**

Un instrument virtuel est un programme qui présente une interface sous forme graphique pour l'apparenter à un instrument physique. Dans LabVIEW, les utilisateurs manipulent des instruments depuis l'ordinateur (virtuels) comme s'il s'agissait d'instruments physiques sur « étagère » (réels). Une application, développée sous LabVIEW, est donc appelée Instrument Virtuel (Virtual Instrument : VI).

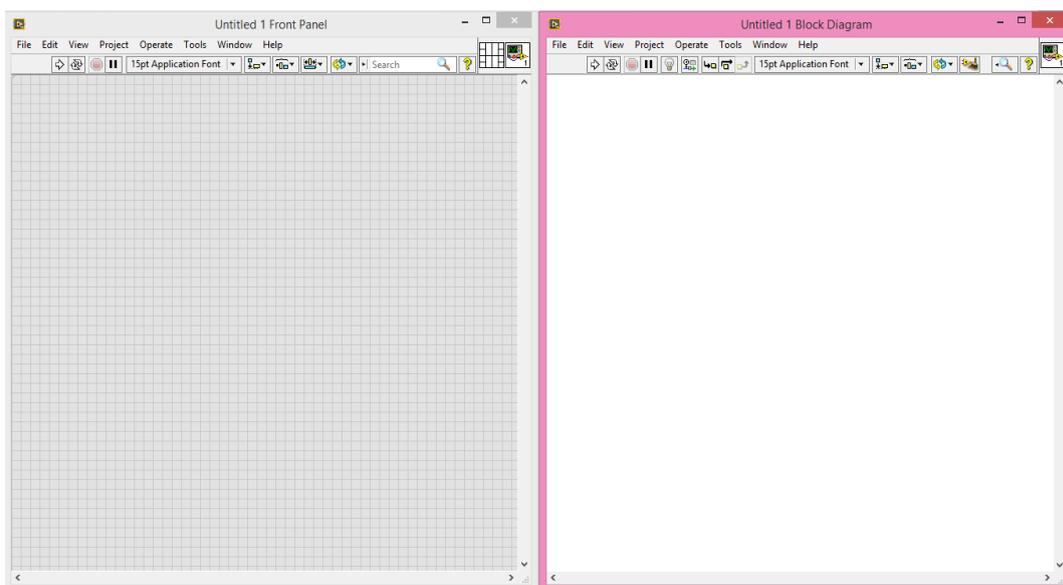
Un programme ou VI, développé dans l'environnement LabVIEW, se compose principalement de deux éléments étroitement associés et regroupés sous le même nom « nom\_application.vi ». nous avons :

- **La face-avant (front panel)**

Représente le panneau de contrôle de l'instrument virtuel composé d'objets variés (boutons, indicateurs, graphes,...). Cette fenêtre est l'interface utilisateur du programme au sens génie logiciel : définition des entrées/sorties de données accessibles par l'utilisateur du programme. En imitant l'interface classique des appareils de mesures (générateurs, oscilloscope,..), LabVIEW apporte une continuité pour les utilisateurs des appareils d'instrumentation. La réalisation de cette interface peut être également effectuée de manière indépendante par rapport au programme (figure 1.8).

- **Le diagramme (diagram)**

C'est le programme de l'application ou code source. Il écrit sous la forme d'un diagramme flux de données en langage G : ensemble des icônes et des liaisons entre ces icônes utilisées. Le diagramme contient les fonctions de l'instrument virtuel. Contrairement à la procédure qui consistait pour le technicien ou le scientifique à dessiner le schéma d'une application et ensuite à le convertir en un code propre au langage choisi, pour LabVIEW, le diagramme et le programme. Représenté en image, le programmes »explique de lui-même et est donc facile à adapter et à comprendre lorsque la représentation graphique garde une taille acceptable. Cette partie de l'application est ce que l'on appelle le code source par opposition à l'interface utilisateur (Figure 1.8) [4].



**Figure 1. 8.** Fenêtre "face avant" & "diagramme"

Une application LabVIEW ou VI est en fait un module logiciel que l'on peut soit exécuter soit inclure dans une autre application. En effet, lorsqu'une partie de l'application est réalisée et testée, il est possible de créer un nouvel objet graphique ou une icône qui représente alors une fonction bien définie par son interface utilisateur (variables en entrée/sortie et fonctionnalité). Cette caractéristique montre que LabVIEW intègre le concept de programmation modulaire. Ainsi, une autre composante d'un VI est la présentation graphique de la hiérarchie (VI Hierarchy) de l'ensemble des modules utilisés, appelés aussi sous-VI.

En résumé, une application développée dans l'environnement LabVIEW est caractérisée de façon complète avec les 4 éléments suivants : « Face-avant » ou interface utilisateur, « Diagramme » ou programme, structure hiérarchique des sous-VI utilisés et l'icône de l'application elle-même [4].

## **1.3 Superviseurs industriels**

### **1.3.1 Introduction**

La supervision est une technique industrielle de suivi et de pilotage informatique de procédés de fabrication automatisés. La supervision concerne l'acquisition de données (mesures, alarmes, retour d'état de fonctionnement) et des paramètres de commande des processus généralement confiés à des programmables, la supervision est la surveillance du bon fonctionnement d'un système ou d'une activité. [5]

### **1.3.2 systèmes SCADA**

#### **a. Supervision dans un environnement SCADA**

Le système SCADA fonctionne par l'acquisition de données provenant de l'installation, ces dernières sont affichées sur une interface graphique, les opérations sont exécutées en temps réel, ainsi les systèmes SCADA donnent aux opérateurs le maximum d'information pour une meilleure décision, il permet un très haut niveau de sécurité, pour les personnels et pour l'installation et permet aussi la réduction des coûts des opérations. [5]

#### **b. Définition du SCADA**

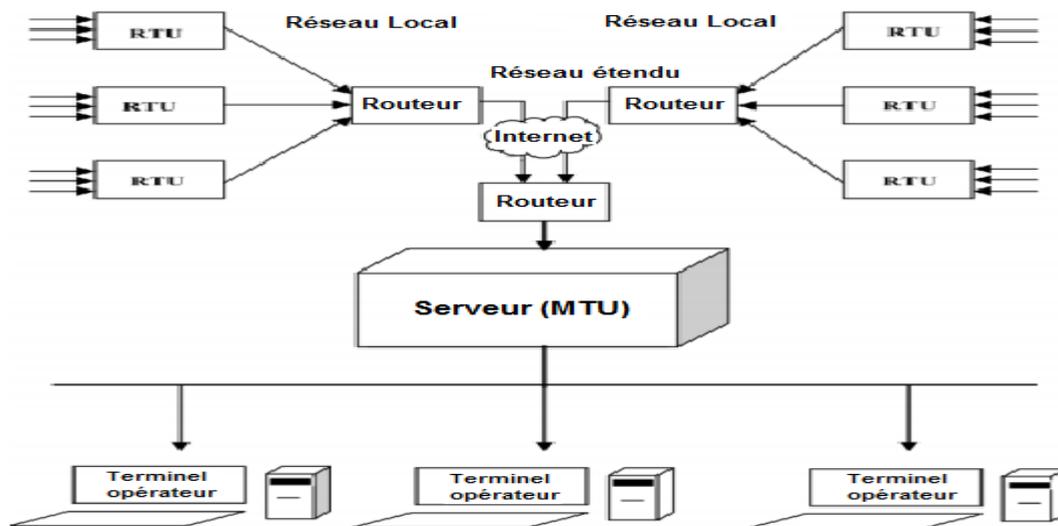
SCADA est un acronyme qui signifie le contrôle et la supervision par acquisition de données (en anglais: Supevisory Control And Data Acquisition).

L'environnement SCADA collecte des données de divers appareils d'une quelconque installation, puis transmet ces données à un ordinateur central, que ce soit proche ou

éloigné, qui alors contrôle et supervise l'installation, ce dernier est subordonné par d'autres postes d'opérateurs [5].

### **c. Architecture du SCADA**

SCADA assure un transfert de données entre le Serveur (MTU, master terminal units) et une ou plusieurs unités terminales distantes (Remote Terminal Units RTUs), et entre le Serveur et les terminaux des opérateurs, la figure ci-dessous représente un schéma sur l'architecture d'un réseau SCADA qui utilise des routeurs pour joindre le poste de pilotage par à travers de l'Internet (Figure 1.9) [6].



**Figure 1. 9.** Architecture du SCADA

Les logiciels de supervision sont une classe de programmes applicatifs dédiés au contrôle de processus et à la collecte d'informations en temps réel depuis des sites distants (ateliers, usines), en vue de maîtriser un équipement (machine, partie opérative) [6].

Les éléments hardware assurent la collecte des informations qui sont à disposition du calculateur sur lequel est implanté le logiciel de supervision, le calculateur traite ces données et en donne une représentation graphique réactualisée périodiquement, le système SCADA enregistre les événements dans des fichiers ou envoie vers une imprimante, mail..., ainsi le système surveille les conditions de fonctionnement anormal et génère des alarmes [6].

#### **d. Avantage du SCADA**

Parmi les avantages du SCADA, on retrouve :

- Le suivi de près du système ; voir l'état du fonctionnement de procédé dans des écrans même s'il se situe dans une zone lointaine ;
- Le contrôle et l'assurance que toutes les performances désirées sont atteintes ; de visualiser les performances désirées du système à chaque instant, et s'il y aurait une perte de performance, une alarme se déclenche d'une manière automatique pour prévenir l'opérateur.
- Produire une alarme lorsqu'une faute se produit et visualise même la position où se situent la faute et l'élément défectueux, ce qui facilite la tâche du diagnostic et de l'intervention de l'opérateur ;
- Donne plusieurs informations sur le système ainsi aide l'opérateur à prendre la bonne décision, et ne pas se tromper dans son intervention ;
- Diminue les tâches du personnel en les regroupant dans une salle de commande ;
- Elimination ou réduction du nombre de visite aux sites éloignés ; avec une interface graphique, on peut suivre l'état de l'installation à chaque instant, ainsi on n'aura pas besoin de faire des visites de contrôle [7].

#### **e. Interfaces graphiques du SCADA**

Les interfaces graphiques sont un outil très important pour le bon déroulement de la procédure d'aide à la décision, elles sont le seul point d'interaction entre l'opérateur et les algorithmes d'aide à la décision, ainsi, elles aident l'opérateur dans sa tâche d'interprétation et de prise de décision, en lui offrant une très bonne visibilité sur l'état et l'évolution de l'installation, avec l'affichage en différentes couleurs des résidus, des alarmes et des propositions sur l'action à entreprendre.

### **1.3.3 Système DCS**

#### **a. Définition du DCS**

Un système numérique de contrôle-commande (SNCC, ou DCS pour distributed control system en anglais) est un système de contrôle d'un procédé industriel doté d'une interface homme-machine pour la supervision et d'un réseau de communication numérique [8].

L'avantage de ces systèmes est leur modularité, qui permet de les installer et de les modifier facilement [9].

### **b. Domaine d'utilisation**

Les DCS sont principalement utilisés dans les industries de procédés dont le mode de production est en continu ou par lots. Par exemple, les procédés où un DCS peut être utilisé incluent :

- pétrochimie et raffineries ;
- métal et mines ;
- installations de traitement métallurgique ;
- fabrication pharmaceutique ;
- centrales nucléaires ;
- fabrication d'automobiles ;
- applications agricoles ;

### **c. Architecture du système**

Les équipements de commande d'un DCS sont distribués. À la différence des systèmes de contrôle centralisés à base d'automates programmables industriels, qui comportent un seul contrôleur central qui gère toutes les fonctions de contrôle-commande du système.

Les DCS sont constitués de plusieurs contrôleurs modulaires qui commandent les sous-systèmes ou unités de l'installation globale.

L'attribut clé d'un DCS est sa fiabilité en raison de la distribution du traitement de contrôle autour des nœuds dans le système. Cela atténue une défaillance d'un seul processeur. Si un processeur tombe en panne, cela affectera seulement une section du processus de l'usine, par opposition à une défaillance d'un ordinateur central qui affecterait l'ensemble du processus. Cette répartition de la puissance de calcul locale sur les baies de connexion de champ d'entrées-sorties (E / S) sur le terrain garantit également des temps de traitement rapides du contrôleur en supprimant les éventuels retards de traitement réseau et central [10].

#### **- Les niveaux de fabrication fonctionnels en utilisant un contrôle informatisé :**

- Le niveau 0 contient les appareils de terrain tels que les capteurs et actionneurs
- Le niveau 1 contient les modules d'entrées-sorties (E/S) industrialisés et leurs processeurs électroniques distribués associés.

- Le niveau 2 contient les ordinateurs de supervision, qui collectent des informations à partir des nœuds du processeur sur le système et fournissent les écrans de contrôle de l'opérateur.
- Le niveau 3 est le niveau de contrôle de la production, qui ne contrôle pas directement le processus, mais qui concerne le suivi des objectifs de production et de surveillance.
- Le niveau 4 est le niveau d'ordonnancement de la production.

Les niveaux 1 et 2 sont les niveaux fonctionnels d'un DCS traditionnel, dans lequel tous les équipements font partie d'un système intégré provenant d'un seul fabricant.

Les niveaux 3 et 4 ne sont pas strictement des processus de contrôle dans le sens traditionnel, mais où le contrôle de la production et l'ordonnancement ont lieu [10].

#### **d. Fonctionnalités du système**

Un DCS est composé à la base d'un ensemble d'équipements à base d'un microcontrôleur assurant chacun une tâche spécifique :

- **Acquisition des données**

Raccordé généralement et essentiellement avec les équipements du terrain (capteur, actionneur), ils permettent de filtrer et conditionner le signal pour assurer la bonne communication entre le régulateur et ces équipements de terrain.

- **Régulation et traitement de donnée**

Ce sont les équipements qui assure la fonction de commande de régulation et d'asservissement en utilisant des boucles de régulation PID ou bien à l'aide d'une logique combinatoire.

- **Un système de communication**

En utilisant des protocoles de communications et des topologies spécifiques pour assurer la communication entre les différents équipements du système.

- **Surveillance**

Ce sont des équipements qui veillent sur le bon fonctionnement de l'ensemble d'équipement du système (alarme système), ainsi que le bon fonctionnement du processus (alarme procès).

- **Historique**

L'historique des données procès, les alarmes système et les alarmes process, les logs...

- **archivage**

Cette fonction assure l'archivage et l'enregistrement de différentes données du système dans la base de données du serveur.

- **Supervision**

Comme son nom l'indique, cette fonction permet de superviser le système et afficher l'état instantané de chaque équipement du système [10].

### **1.3.4 Les différents constructeurs des superviseurs**

#### **a. National Instruments**

National Instruments Corporation, communément appelée NI, est une entreprise américaine cotée au NASDAQ (sous le sigle NATI) qui compte (en 2011) plus de 6000 employés répartis dans 41 pays. Son siège social est basé à Austin, Texas. Pionnier de l'instrumentation virtuelle, National Instruments est un fournisseur de matériels et de logiciels permettant aux scientifiques et aux ingénieurs de concevoir, prototyper et déployer des systèmes aux applications de test et mesure, de contrôle/commande et embarqués (Figure 1.10) [11].



*Figure 1. 10.* National Instruments

➤ **La principale plate-forme de développement logiciel de l'entreprise**

- NI LabVIEW, intègre le langage de programmation graphique G.

Parmi les autres produits logiciels, on compte :

- NI LabWindows/CVI (outils de développement d'applications en C/C++);
- NI Measurement Studio (outils de développement d'applications avec Visual Studio .NET);
- NI TestStand (séquencement et gestion de programmes de test);

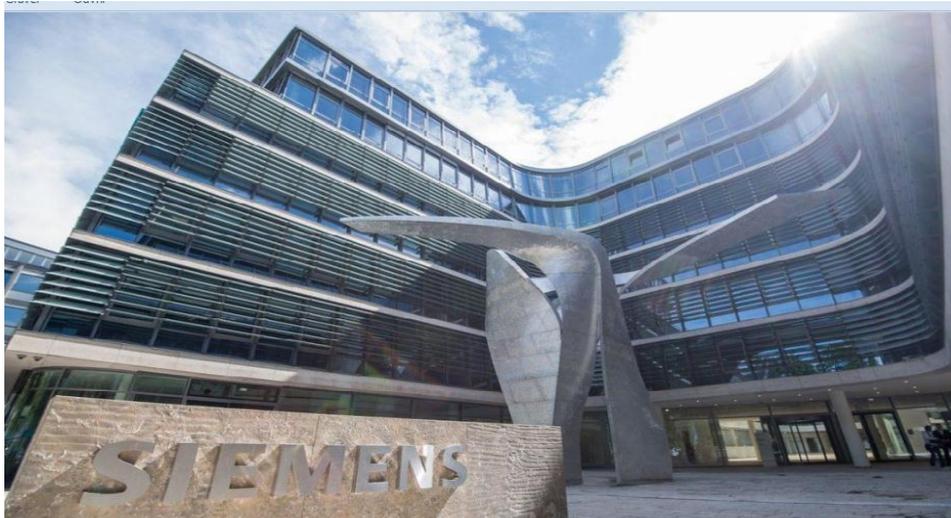
- NI VeriStand (développement configuratif d'applications de test temps réel) ;
- Multisim (anciennement ElectronicsWorkbench, outil de conception de circuits électroniques) ;
- DIAdem (post-traitement de données et génération de rapports) ;

➤ **Matériel**

National Instruments propose notamment la plate-forme des modules d'acquisition de données, de mesure, de contrôle, de vision, de commande d'axes moteurs, de communication par bus industriel... permettant de construire des systèmes complets de test automatique ou d'automatisation industrielle [11].

**b. Siemens**

Siemens est un groupe international d'origine allemande spécialisé dans les hautes technologies et présent dans les secteurs de l'industrie, de l'énergie et de la santé. Il a été fondé en 1847 par Werner von Siemens. Le groupe, dont le siège est à Munich, est le premier employeur privé d'Allemagne, et la plus grande société d'ingénierie (en termes d'effectifs) en Europe (Figure 1.11) [12].



**Figure 1. 11.** Siemens

➤ **Logiciel**

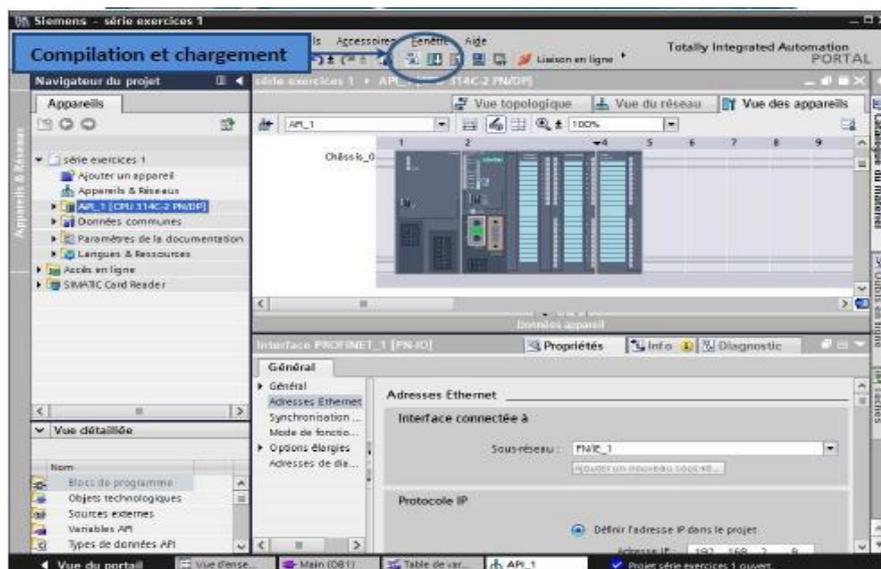
Le TIA portal est le logiciel de programmation le plus connu au monde et le plus utilisé dans le domaine de l'automatisation industrielle, il contient toutes les séries de communication.

Le TIA Portal est la clé ouvrant l'accès au potentiel intégral de la Totally Integrated Automation. Le logiciel optimise l'ensemble des procédures au niveau planification, machine et processus.

Son interface utilisateur intuitive, ses fonctions simples et sa transparence totale des données le rendent extrêmement convivial. Des données et projets déjà existants peuvent être intégrés aisément, ce qui garantit la sécurité de l'investissement [13].

La plateforme **Totally Integrated Automation Portal** est le nouvel environnement de travail Siemens qui permet de mettre en œuvre des solutions d'automatisation avec un système d'ingénierie intégré comprenant les logiciels SIMATIC STEP 7 V13 et SIMATIC WinCC V13 (dans la version du programme disponible au CTA de Virton) (voir la figure 1.12) [14].

SIMATIC WinCC est un système de supervision de processus modulable qui offre des fonctions performantes de surveillance d'automatismes. WinCC offre des fonctionnalités SCADA complètes sous Windows pour tous les secteurs depuis la configuration monoposte jusqu'aux configurations multipostes distribuées avec serveurs redondants et solutions multi-sites avec clients Web [15] [16].



**Figure 1. 12.** TIA Portal

- **Les variables API**

Dans TIA Portal, toutes les variables globales (entrées, sorties, mémentos,..) possède une adresse symbolique et une adresse absolue.

- ✓ L'adresse absolue représente l'identificateur d'opérande (I, Q, M,...) et son adresse et numéro de bit.
- ✓ L'adresse symbolique correspond au nom que l'utilisateur a donné à la variable [14].

➤ **protocoles de communication**

S7comm (S7 Communication) est un mode de communication propriétaire de Siemens. Il permet de faire communiquer des automates Siemens de la famille S7-300 / 400/1200/1500. Il permet l'échange de données entre automates, l'accès aux données de l'API à partir du SCADA (télésurveillance et acquisition de données) et pour des fins de diagnostic.

On peut réaliser une liaison S7 communication via les interfaces réseau MPI, Profibus ou industriel Ethernet. La communication va ainsi être gérée via les couches physique / Réseau, comme RS-485 pour MPI (Multi-Point-Interface) ou Profibus DP et ISO ou TCP pour l'Ethernet industriel.

Avant d'établir une connexion S7, on doit d'abord faire la configuration du matériel sur le logiciel Simatic manager Step 7 ou sur TIA portal. Puis, la communication entre les deux partenaires pourra être configurée sur NetPro [17].

Dans notre projet Pour faire la communication entre le TIA portal et Le LabVIEW, L'OPC est la méthode la plus simple, la plus sûre et la plus facile pour communiquer avec un automate. Un Serveur OPC est basiquement un programme relié à un automate en passant par des drivers de communication spécifiques qui permettent d'accéder aux données de cet automate de manière standardisée et indépendante du type d'automate utilisé.

- **Définition**

L'OPC est un standard de communication qui permet la communication entre plusieurs applications et permet l'interopérabilité entre les différents fabricants de logiciels et de matériels [17].

- **La norme OPC se compose de plusieurs spécifications :**

- **OPC DA (ou Data Access)**: permet l'accès aux données en temps réel
- **OPC HDA**: pour l'historisation des données
- **OPC A & E**: pour la gestion des alarmes et événements
- **OPC UA (Unified architecture)** : dernière spécification

- **L'architecture d'un réseau fonctionnant avec L'OPC**

Le réseau se compose toujours d'au moins 3 parties :

**- Un dispositif matériel ou logiciel**

Ce dispositif matériel ou logiciel contient les données à exploiter. Il peut être un automate, un SNCC, une base de données, un fichier Excel, un RTU, un logiciel SCADA etc.

Dans notre cas nous avons utilisé un automate programmable siemens S7-312c.

**- Un serveur OPC**

Un serveur OPC est un logiciel qui «connaît» le langage propriétaire du matériel ou du logiciel où il va collecter les données à exploiter. Il existe des serveurs OPC pour les différents marques d'automates (Siemens, Allen Bradley, Omron, GE, Schneider, Honeywell, Emerson, Yokogawa, ABB, etc ...) et des serveurs OPC pour des applications comme Excel (DDE), pour les bases de données (ODBC), pour des protocoles de communication connus comme le Modbus ou encore le DNP3.

En général, les fabricants d'automates disposent de leur propre serveur OPC. C'est le cas de Siemens avec sa suite logicielle Simatic Net, Rockwell avec RSLINX, National Instruments, Schneider avec OFS OPC Server etc. Il existe aussi des entreprises spécialisées dans la conception de serveurs OPC. On peut citer par exemple : Matrikon, Kepware avec KepserverEx, Softing etc. [17].

➤ **NI OPC SERVEUR**

Le pont NI OPC Servers convertit les protocoles industriels propriétaires en protocoles ouverts OPC Classic et OPC Unified Architecture (UA). Cette conversion vers OPC permet ensuite au logiciel NI LabVIEW de communiquer avec de nombreux automates programmables (API) [11].

➤ **IBH OPC SERVEUR**

Une application de visualisation peut être couplée avec un API SIMATIC S5, S7-1200, S7-200, **S7-300** et S7-400 ou un SoftPLCIBHsoftec. Une opération mixte est également possible. Accès aux variables d'un automate Control via OPC. L'adressage symbolique utilisé dans le programme API et les blocs de données peuvent être directement utilisés dans l'IHM. En quelques clics de souris, toutes ou uniquement les variables souhaitées dans l'automate peuvent être sélectionnées.

La communication entre le serveur IBH OPC et les automates Siemens peut être établie via:

- S5-AS511 ;
- S5-TCP / IP ;
- S5 via le lien IBH S5 ++ ;
- IBH-S5-SoftPLC ;
- **S7-300, S7-400** via Profibus / **MPI à l'aide d'un adaptateur PC (série / USB) ;**
- S7-200 via PPI à l'aide d'un adaptateur PC (série / USB) ;
- S7-200, S7-300, S7-400 via IBH Lien S7 ++ / S7 ++ HS;
- S7-200, S7-1200, S7-1500, S7-300, S7-400 avec S7-TCP / IP via Ethernet;

Le Serveur utilisé pour développer notre projet est l'IBH OPC Serveur.

La non-utilisation du serveur NI OPC dans le cadre de notre travail est liée au problème de connexion avec l'automate S7-312c (MPI-USB).

Le constructeur NI ne donne pas de possibilités de connexion avec MPI-USB ce qui nous a mis dans l'obligation de procéder à une solution de rechange NI OPC serveur par IBH (adaptable) et faciliter la conduite et le bon déroulement de notre expérience [18].

#### **- Un Client OPC**

Un client OPC est un logiciel qui met en œuvre les spécifications du standard OPC et qui peut communiquer avec tout serveur OPC. Un client OPC peut se connecter à un serveur OPC de n'importe quels fabricants. Si l'on veut créer une application de supervision, on fait appel au standard OPC DA. Un client OPC peut être une application Visual Basic, C#, LabVIEW etc...

Le client qui nous avons utilisé pour développer notre projet est le LabVIEW.

## **1.4 Conclusion**

Ce chapitre porte essentiellement sur l'intérêt de la supervision industrielle pour le contrôle et commande des processus industriels et les outils de programmation que nous avons utilisée tout au long de ce projet.

Après avoir détaillé tous les fondements théoriques proposés dans la littérature ainsi que dégagé les notions de bases, nous allons aborder la partie suivante qui représente la partie pratique.

# Chapitre 2 développement et Réalisation

---

## 2.1 Introduction

Cette partie présente l'importance d'utilisation de systèmes d'acquisition de données NI USB-6001 pour lire et afficher la variation de température de thermocouple et mesure et afficher la pression de capteur de pression et aussi commande la vitesse d'un moteur asynchrone et l'arrêt en sécurité par le photocellule.

Et de plus, Cette partie est présente aussi l'utilisation de LabVIEW pour développer la programmation graphique de SCADA (Supervision Control and Data Acquisition) et pour communiquer avec les automates programmables à travers L'OPC server.

La partie la plus importante est la communication entre les trois automates programmables S7-312c qui continent trois programmes Différent a travers d'un réseau MPI.

## 2.2 Système d'acquisition de données

Un système de contrôle informatisé se compose généralement d'un ordinateur numérique, d'une carte d'acquisition de données, d'un circuit d'interface électronique et d'un logiciel permettant d'obtenir, d'analyser et d'afficher les données obtenues lors de l'acquisition. Pour la communication entre l'ordinateur et les appareils externes sont disponibles sur le marché plusieurs solutions. Dans notre cas, nous avons utilisé une carte d'acquisition de données (DAQ) programmable par un logiciel basé sur LabVIEW.



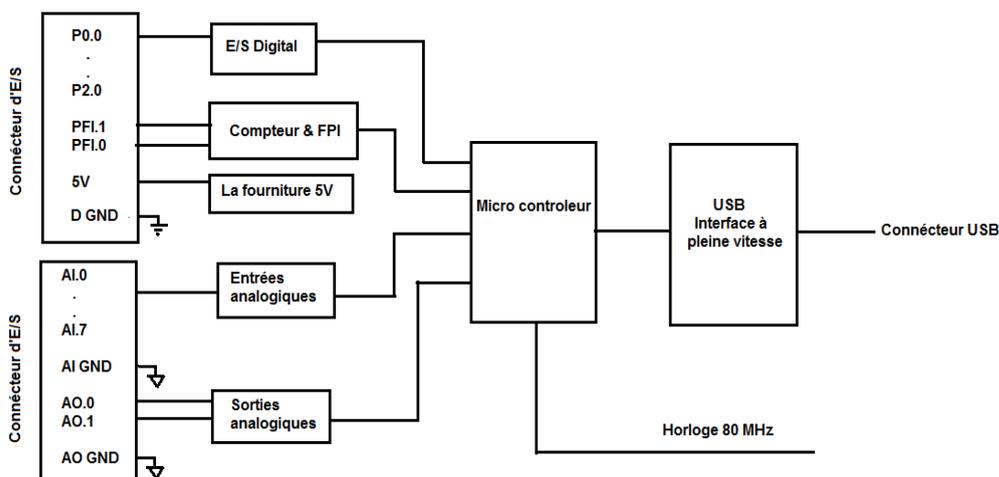
*Figure 2.1.* NI USB-6001

Le NI USB-6001 est un périphérique USB fournit huit canaux d'entrée analogique (AI), qui peuvent également être configurés comme quatre canaux différentiels. Il comprend également deux canaux de sortie analogique (AO), 13 canaux d'entrée / sortie numérique (DIO) et un compteur 32 bits [19]. (Voir la Figure 2.2)



**Figure 2.2.** Les canaux de NI USB-6001

La Figure 2.3 montre les principaux composants fonctionnels du dispositif NI-USB-6001



**Figure 2.3.** Diagramme de bloc NI-USB-6001

### 2.2.1 E / S numériques

Le NI USB-6001 possède 13 lignes numériques: P0. <DI0..DI7>, P1. <DO0..DO3> et P2.0(PFI). D GND est le signal de référence de masse pour les E / S numériques. Nous pouvons programmer individuellement chaque ligne en entrée ou en sortie. Toutes les mises à jour et échantillons numériques d'entrée et de sortie numérique sont programmés par logiciel.

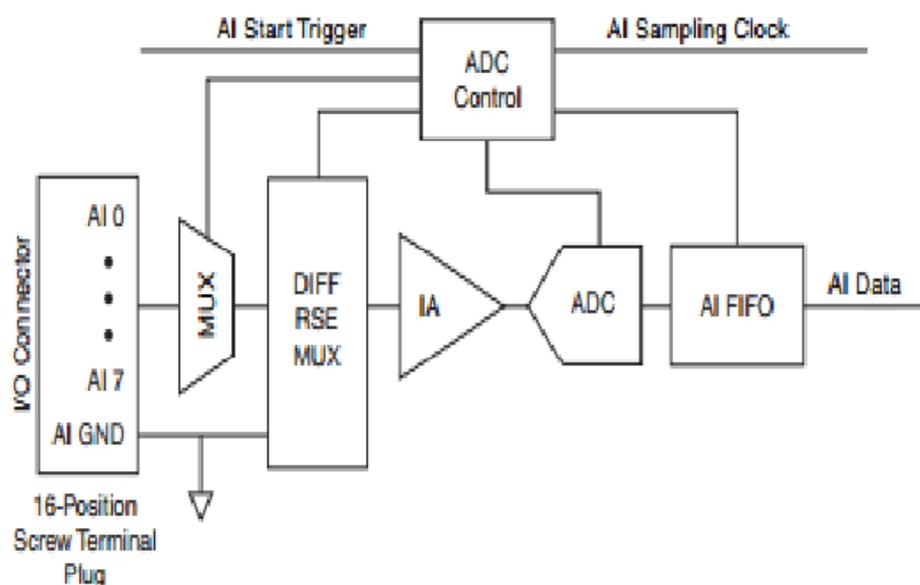
Utilisation de PFI comme source de comptage. Nous pouvons configurer PFI 0 ou PFI 1 comme source de comptage d'arêtes numériques. Dans ce mode, les fronts montants ou descendants sont comptés à l'aide d'un compteur 32 bits [19].

**Remarque : Les fronts ne peuvent être comptés qu'à partir de 0 ou d'une valeur initiale telle que 1, 2, 3 .... Le décompte n'est pas pris en charge.**

### 2.2.2 Entrées analogiques

USB-6001 dispose de huit canaux d'entrée analogique que nous pouvons utiliser pour quatre mesures d'entrée analogique différentielle ou huit mesures d'entrée analogique asymétriques.

Nous pouvons configurer les canaux d'entrée analogiques sur le périphérique NI DAQ pour prendre jusqu'à quatre mesures différentielles (DIFF) ou huit mesures asymétriques référencées (RSE) [19].



**Figure 2.4.** Circuit de entres analogique

#### a. Mode différentiel

En mode DIFF, le périphérique NI DAQ mesure la différence de tension entre deux signaux AI.

Pour les signaux différentiels, le signal positif ou de la source connecter à la borne AI+, et le signal négatif à la borne AI- [19].

**b. Mode référencé à une seule extrémité:**

En mode RSE, le périphérique NI DAQ mesure la tension d'un signal « AI » par rapport à AI GND. Les modes d'entrée analogiques sont programmés par canal.

Par exemple, Nous pouvons configurer l'appareil, deux canaux en mode différentiel et quatre canaux en mode asymétrique. Les paramètres AI déterminent la manière dont nous connectons nos signaux AI au périphérique NI DAQ.

• **Prise de mesures unilatérales référencées :**

Pour la connexion des signaux de tension à extrémité unique référencés (RSE) au dispositif NI DAQ, nous avons connectées le signal de tension positive à une borne AI et le signal de masse à une borne AI GND.[19]

**c. Plage d'entrée**

Le périphérique NI DAQ a une plage d'entrée de  $\pm 10$  V.

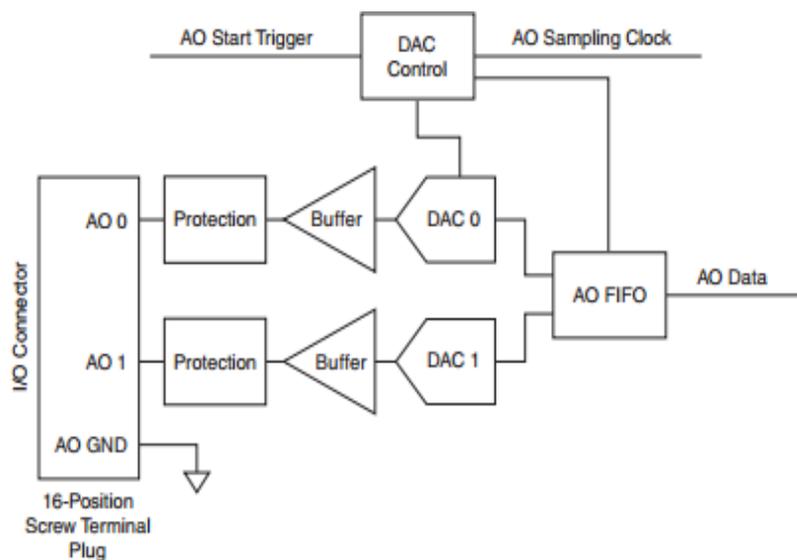
Pour le mode différentiel, chaque AI doit rester à  $\pm 10$  V par rapport à AI GND, et la tension entre les entrées positives et négatives doit être inférieure ou égale à  $\pm 10$  V.

Pour le mode RSE, les signaux de  $\pm 10$  V sur toute borne d'entrée analogique par rapport à AI GND sont mesurés avec précision [19].

**2.2.3 Sortie analogique**

Les signaux de sortie analogiques sont référencés à AO GND.

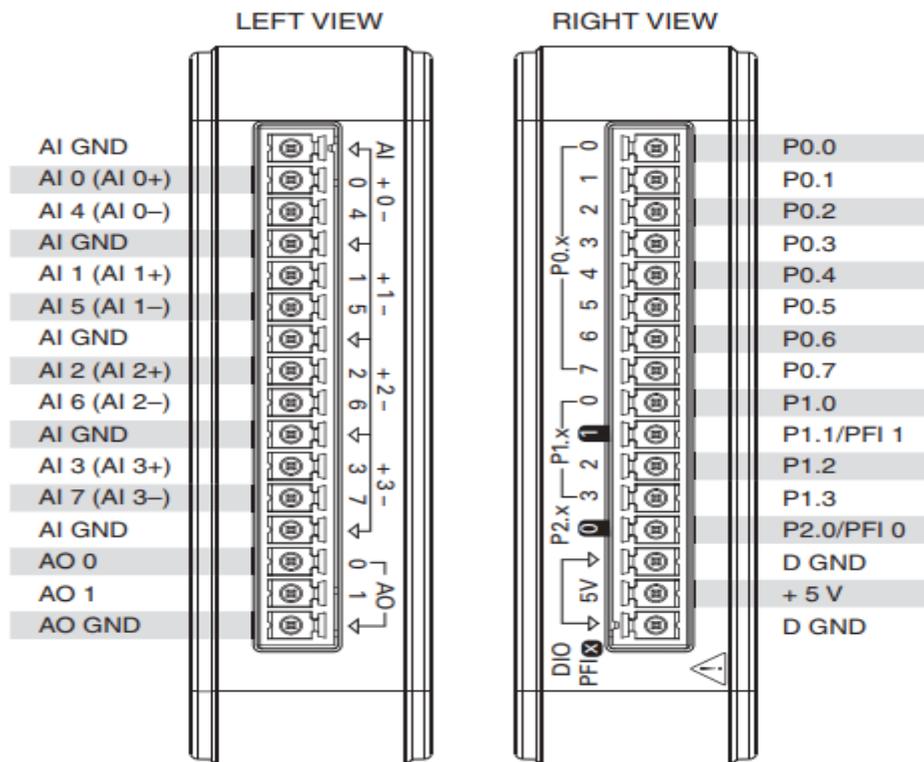
La plage AO est de  $\pm 10$  V. (Figure 2.5)



**Figure 2.5.** Circuit de sortie analogique

### 2.2.4 Brochage et descriptions des signaux

La figure 2.6 montre le brochage du périphérique NI DAQ USB-6001. Les noms des signaux d'entrée analogiques sont répertoriés comme nom d'entrée analogique AI x, puis le nom de l'entrée analogique différentielle (AI x +/-), et le tableau 2.1 présente une description détaillée de chaque signal.



**Figure 2. 6 .NI-USB-6001**

Nom du signal	Référence	Direction	Description
AI GND	---	---	Ground Analog Input (Entrée analogique): point de référence pour les mesures d'entrée analogiques asymétriques, Canaux d'entrée analogiques 0 à 7 - Pour les mesures asymétriques, chaque signal correspond à un canal de tension d'entrée analogique. Pour les mesures différentielles

AI <0..7>	AI GND	Entrée	Canaux d'entrée analogiques 0 à 7 - Pour les mesures asymétriques, chaque signal correspond à un canal de tension d'entrée analogique. Pour les mesures différentielles, AI 0 et AI 4 sont les entrées positives et négatives du canal d'entrée analogique différentiel 0. Les paires de signaux suivantes forment également des canaux d'entrée différentiels: AI<0,4>,AI<1,5>, AI <2, 6> et AI <3 , 7>.
AO GND	---	---	Ground Analog Output - Point de référence pour la sortie analogique.
AO <0, 1>	AO GND	Sortie	Canaux de sortie analogique 0 et 1 - Fournit la sortie de tension des canaux AO.
P0.<0..7>	D GND	Entrée ou sortie	Port 0 Canaux d'E / S numériques 0 à 7- Nous pouvons configurer chaque signal individuellement comme une entrée ou une sortie.
P1.<0..3>	D GND	D GND	Port1 Canaux d'E / S numériques 0 à 3- Nous pouvons configurer chaque signal individuellement comme une entrée ou une sortie.
P2.0	D GND	Entrée ou sortie	Port 2 Canal E / S numérique 0- Nous pouvons configurer chaque signal individuellement comme une entrée ou une sortie.
PFI 0, 1	D GND	Entrée	Interface de fonction programmable ou canaux d'E / S numériques: entrée de compteur de bord ou entrée de déclenchement numérique.
D GND	---	---	Digital Ground - Le point de référence pour les signaux numériques
+5 V	D GND	Sortie	Source d'alimentation +5 V - Fournit une puissance de +5 V jusqu'à 150 mA.

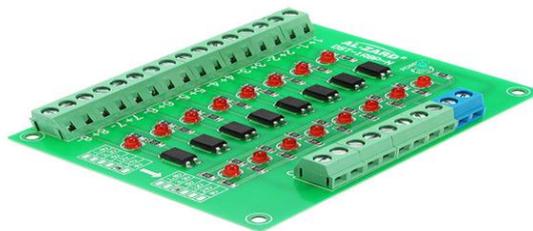
**Tableau 2. 1.** Description des signaux

### 2.2.5 Utilisation des entrées digitales DI

Nous avons configuré le port0 (p0) comme entrées digital, donc nous avons huit entrées numériques, qui peuvent également être programmées comme entrées ou sorties numériques DI/O,

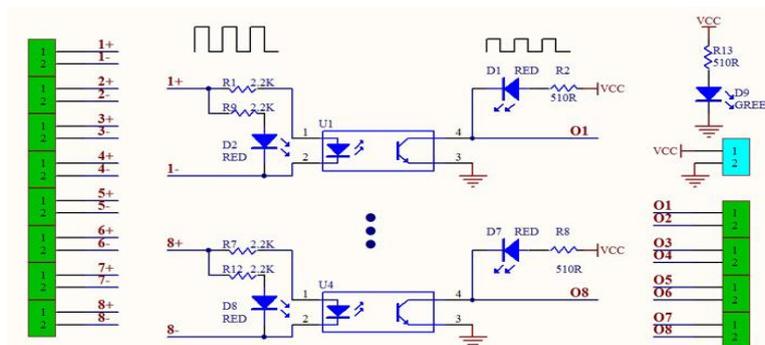
nous avons utilisé une de ces huit entrées digital (DI0) avec une photocellule qui vas permettre d'arrêter un moteur lorsque la cible pénètre dans le faisceau c'est-à-dire un modèle de sécurité. La photocellule utilisé est de type industrielle qui nécessite une alimentation de 24V ( délivrer par l'alimentation ) donc le signal généré par la photocellule a une amplitude de 24V, ce qui est une contrainte pour le module NI-USB-6001 qui accepte au niveau de ces entrées une tension ne dépassant pas 5V d'après sa fiche technique, ce qui nous a obligé d'utilisé un convertisseur DC/DC pour faire baisser la tension de 24V à une tension de 5V tolérer par le module NI-USB-6001. Le convertisseur que nous avons utilisé est un convertisseur de huit canaux représenté à la figure 2.7 qui à un double avantage à savoir :

- Réduire la tension de 24V à 5V ;
- Isoler le NI-USB-6001 des perturbations éventuelle sur la photocellule ;



**Figure 2.7.** Convertisseur de huit canaux 24v vers 5v

Tel que la figure 2.8 c'est le schéma électrique de convertisseur de huit canaux



**Figure 2.8.** Schéma électrique de convertisseur de 8 canaux

**a. Les détecteurs photoélectriques**

Permettent la détection d'objets de toutes natures (opaques, transparents, réfléchissants...) dans des applications industrielles et tertiaires les plus diverses. La détection s'appuie sur les cinq systèmes de base suivants :

Barrage, reflex, reflex polarisé, proximité, proximité avec effacement de l'arrière-plan. Les appareils existent sous différentes formes : compacts, miniatures, su miniatures, à fibres optiques, rectangulaires avec boîtier plastique ou métallique, ou cylindriques avec corps en plastique, métal ou inox [20].

L'union des systèmes et des formes assure une adaptation optimale à la nature de l'objet à détecter, à la place disponible et aux conditions d'environnement (Figure2.9).



**Figure 2. 9.** Détecteur photoélectriques

Un détecteur photoélectrique réalise la détection d'une cible (objet ou personne) au moyen d'un faisceau lumineux. Ses deux constituants de base sont un émetteur et un récepteur de lumière [20].

**Principe de la détection optique :**

La détection est effective quand la cible pénètre dans le faisceau lumineux et modifie suffisamment la quantité de lumière reçue par le récepteur pour provoquer un changement d'état de la sortie [20].

Elle s'effectue selon deux procédés :

- Blocage du faisceau par la cible. (émetteur et récepteur sont séparées) ;
- Renvoi du faisceau sur le récepteur par la cible. (émetteur et récepteur se trouve dans le même boîtier) ;

Tous les détecteurs photoélectriques ont un émetteur à diode électroluminescente (DEL) et un récepteur à phototransistor [20].

Dans notre projet nous avons utilisé un photocellule de type **FAR2/BP-0 E**.

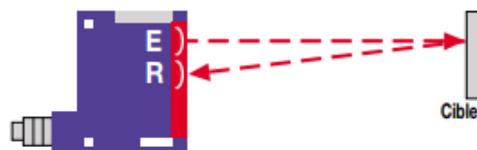
**Photocellule FAR2/BP-0 E** : c'est un détecteur infrarouge (émetteur et récepteur dans le même boîtier) de la série FA qui peut être alimenté par une tension de 10V à 30V, délivrant un courant maximal de 100 mA et dont la distance de détection ne dépasse pas 100mm [20].



**Figure.2. 10.** Photo cellule FAR2/BP-OE

Elle fonctionne en principe selon le procédé de renvoi du faisceau. En cas d'absence de cible, le faisceau lumineux n'arrive pas sur le récepteur et quand une cible pénètre dans le faisceau, elle renvoie ce dernier sur le récepteur (Figure 2.11).

Lumière sur le récepteur = détection

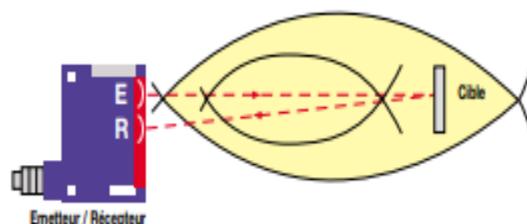


**Figure 2. 11.** Faisceau renvoyé par la cible

Deux systèmes de base fonctionnent selon ce procédé fondé sur les propriétés de réflexion de la lumière des objets à détecter : proximité, proximité avec effacement de l'arrière-plan [20].

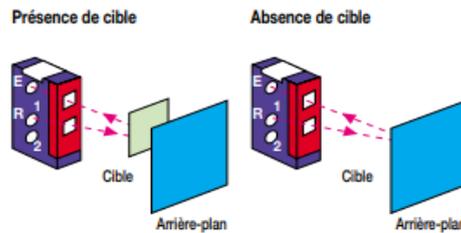
**Proximité** : Comme pour le système reflex, émetteur et récepteur sont regroupés dans un même boîtier.

Le faisceau lumineux est renvoyé vers le récepteur par tout objet suffisamment réfléchissant qui pénètre dans la zone de détection (figure 2.12).[20]



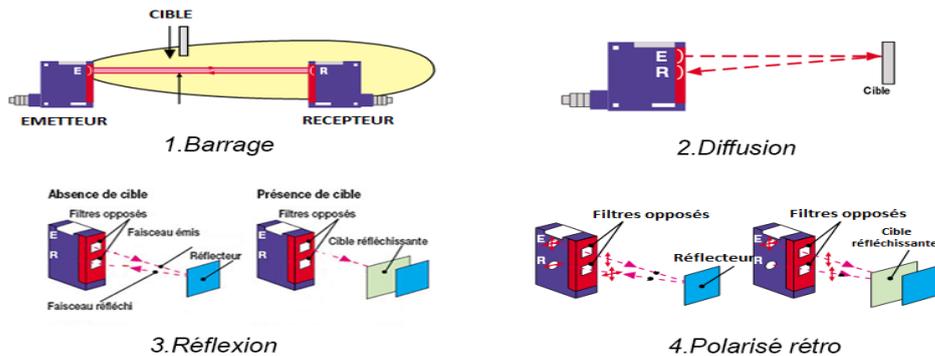
**Figure 2. 12.** Principe du système proximité

**Proximité avec effacement de l'arrière-plan :** Les détecteurs de proximité avec effacement de l'arrière-plan sont équipés d'un potentiomètre de réglage de portée. Ce dernier permet de se "focaliser" sur une zone de détection en évitant toute réflexion parasite sur l'arrière-plan [20].



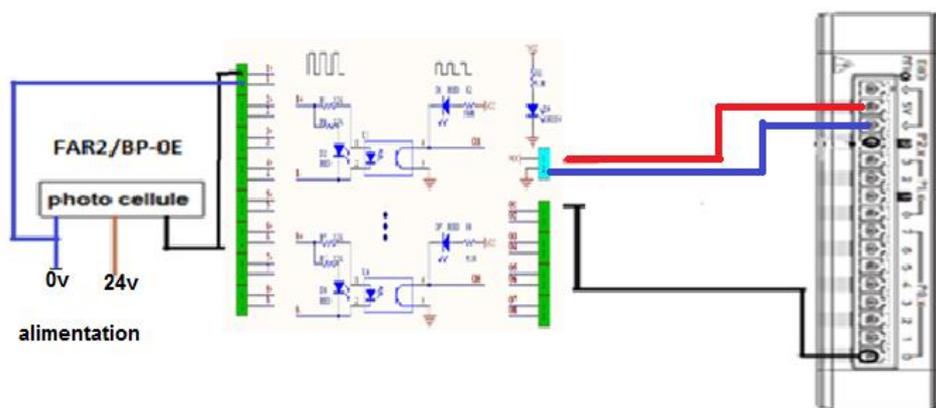
**Figure 2.13.** Détection avec réglage d'effacement d'arrière- Photo coupleur

La figure 2.14 résume toutes les détections



**Figure 2.14.** Les détections photoélectriques

Nous avons branché la photocellule selon le schéma de câblage de la figure 2.15 :



**Figure 2.15.** Câblage de la photocellule avec DIO

**Le fonctionnement de montage.**

l'anode de la photodiode (pin n°1) est relié à une résistance ( pull down) de 2.2K. Lorsque l'anode de la photodiode (pin n1) est alimenté à travers la photocellule externe FAR2/BP-0E par une tension de 24V, la photodiode émet une lumière vers le phototransistor interne ce qui fait saturer ce dernier et la tension de son collecteur (5V) se trouve appliquer à son émetteur a une différence de 0.3V (VCE sat=0.3V).

Le signal ainsi obtenu au niveau du pin de sortie n1 (o1) appliqué directement à l'entrée digital DIO de NI-USB-6001.

Dans notre cas, lorsque une cible pénètre dans le faisceau de la photocellule provoque l'arrêt du moteur.

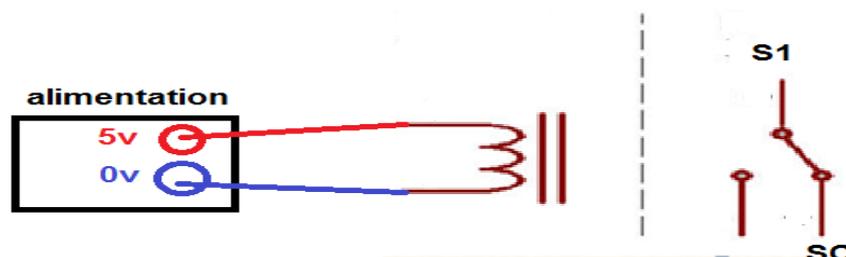
**2.2.6 Utilisation des sorties digitales DO**

On a utilisé un moteur asynchrone commandé par un variateur de vitesse électronique OMRON VS mini J7 qui permet de faire démarrer le moteur à travers les commandes qui provienne du NI-USB-6001. Ces commandes nécessite généralement un appel de courant important ce qui constitue une contrainte pour le module NI-USB-6001 car le courant délivré par ces ports des sorties digital ne dépasse pas 4mA, ce qui nous a mis dans l'obligation d'utiliser de Relais électromécanique

**a. Relais électromécanique**

➤ **Introduction**

Un relais électromécanique est doté d'un bobinage en guise d'organe de commande. La tension appliquée à ce bobinage va créer un courant, ce courant produisant un champ électromagnétique à l'extrémité de la bobine (il ne s'agit ni plus ni moins que d'un électro-aimant). Ce champ magnétique va être capable de faire déplacer un élément mécanique métallique monté sur un axe mobile, qui déplacera alors des contacts mécaniques [21]. (Voir la Figure 2.16)



**Figure 2.16.** Schéma de fonctionnement d'un relais électromécanique

➤ **Avantages du relais électromécanique**

- Capacité de commuter aussi bien des signaux continus qu'alternatifs sur une large gamme de fréquences.
- Fonctionnement avec une dynamique considérable du signal commuté.
- Aucun ajout de bruit ou de distorsion.
- Résistance de contact fermé très faible (il est moins facile de trouver des valeurs aussi faibles avec des composants électroniques).
- Résistance de contact ouvert très élevée (il est moins facile de trouver des valeurs aussi élevées avec des composants électroniques).
- Très grande isolation entre circuit de commande (bobine) et circuit commuté (contacts).
- Possibilité de résoudre des problèmes d'automatisme de façon parfois plus simple qu'avec un circuit électronique.

Dans notre projet nous avons utilisé un relais électromécanique de deux canaux.

**b. Moteur asynchrone**

La machine asynchrone, connue également sous le terme anglo-saxon de machine à induction, est une machine électrique à courant alternatif sans connexion entre le stator et le rotor.

Comme les autres machines électriques (machine à courant continu, machine synchrone), la machine asynchrone est un convertisseur électromécanique basé sur l'électromagnétisme permettant la conversion bidirectionnelle d'énergie entre une installation électrique parcourue par un courant électrique (ici alternatif) et un dispositif mécanique (Figure 2.17) [22].



**Figure 2. 17.** Moteur Asynchrone

Cette machine est réversible et susceptible de se comporter, selon la source d'énergie, soit en « moteur » soit en « générateur ».

La machine se compose de deux pièces principales :

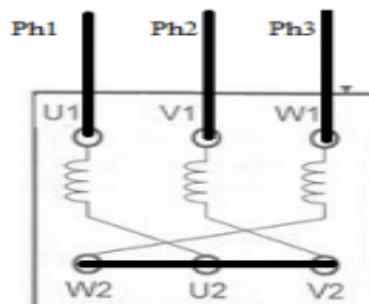
Le stator est une pièce construite en matériau ferromagnétique, servant de support et incluant un bobinage relié au réseau ou à un variateur de vitesse ;

Le rotor est un cylindre en matériau ferromagnétique relié au stator par des paliers. Il comporte un enroulement constitué de conducteurs en court-circuit parcourus par des courants induits par le champ magnétique créé par les courants satiriques. C'est la principale différence avec une machine synchrone, laquelle a un rotor avec un champ magnétique provenant d'aimants permanents ou de bobines alimentées en courant continu [22].

Nous utilisons dans notre expérience un moteur asynchrone de 3000 tour/min à 50 HZ.

➤ **Couplage du stator** : Le stator peut être alimenté selon deux couplages : étoile ou triangle. La tension aux bornes des enroulements (bobinages) ne sera pas la même suivant le couplage.

➤ **Couplage étoile**: Le schéma de raccordement est donné ci-contre : Dans ces conditions, l'enroulement voit à ses bornes la tension simple du réseau (Figure 2.18).

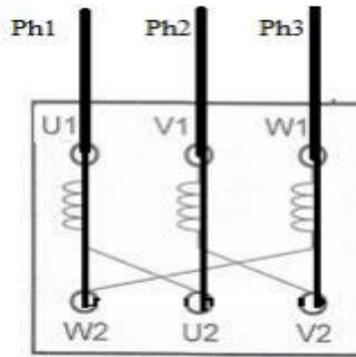


**Figure 2. 18.** Couplage étoile

➤ **Couplage triangle** : Dans ces conditions, l'enroulement voit à ses bornes la tension composée du réseau.

Exemple : sur le réseau de 230/400 V, un moteur couplé en triangle aurait une tension sur chaque bobinage du stator de 400V.

Les plaques signalétiques des Moteurs asynchrone indiquent quel couplage réaliser en fonction de la tension composée du réseau, puis les grandeurs nominales du moteur pour le couplage considéré (Figure 2.19) [23].



**Figure 2. 19.** Couplage triangle

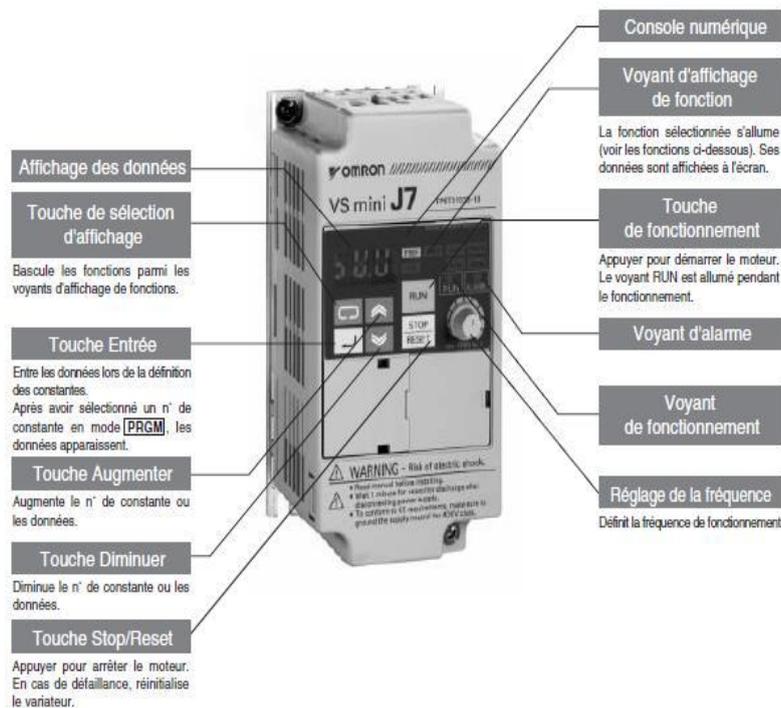
**c. Variateur de vitesse**

Le variateur de vitesse moteur est idéal pour toutes les applications de commande simple (marche / arrêt) ou avancées (variation de vitesse).

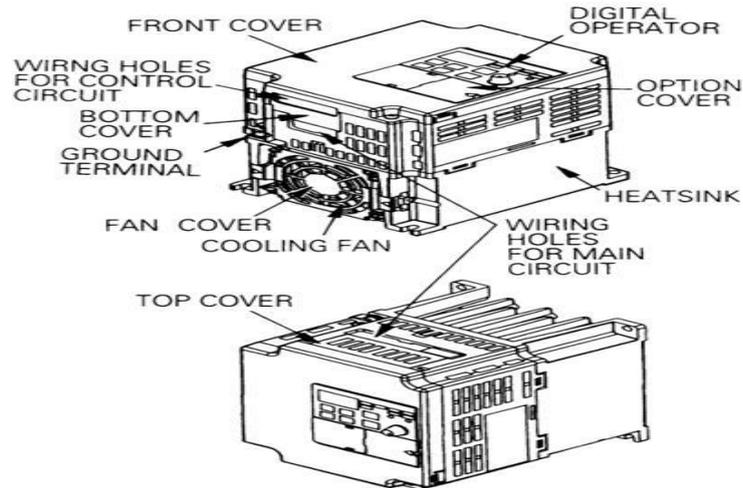
Fonctionnant sur le principe de la modulation de la fréquence d'alimentation (modification de largeur d'impulsion ou MLI) le variateur moteur permet de faire varier finement la vitesse de votre moteur sans faire l'impasse sur ses spécifications d'origine (peu de perte de couple, possibilité de faire tourner le moteur à quelques trs/min).

Dns notre cas nous avons utilisé un variateur de vitesse de type mini j7 (1.1kw OMRON). (Voir la figure 2.20).

➤ **Nomenclature :**



**Figure 2. 20.** OMRON VS mini J7



**Figure 2. 21.** Nomenclature d'un variateur de vitesse

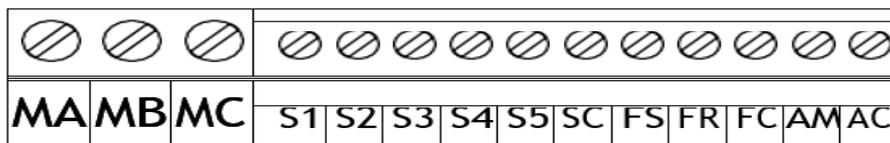
➤ **Opérateur numérique**



**Figure 2. 22.** console numérique du variateur de vitesse

Console numérique (avec potentiomètre) Utilisé pour définir ou modifier des constantes. La fréquence peut être réglée en utilisant un potentiomètre.

➤ **Bornes du circuit de contrôle :**



**Figure 2. 23.** Bornes du circuit de contrôle

symbole	Nom	Fonction	Niveau du signal
Entré	<b>S1</b>	Avant / arrêt Avant en position ON. Arrêt en position OFF.	Optocoupleur 8 mA à 24 Vcc. Remarque : NPN est le paramètre par défaut de ces bornes. Nous Raccordons les en fournissant une terre commune. Aucune alimentation
	<b>S2</b>	Entrée multifonction 1 (S2) Définie par le paramètre n36 (arrière / arrêt)	

	<b>S3</b>	Définie par le paramètre n36 (arrière / arrêt)	Définie par le paramètre n37 (réinitialisation erreur)	externe n'est nécessaire. Toutefois, pour fournir une alimentation externe et raccorder les bornes via une ligne positive commune, réglez le commutateur SW7 sur PNP et assurez-vous que l'alimentation est réglée sur 24 Vc.c. ±10 %.
	<b>S4</b>	Entrée multifonction 4 (S5)	Définie par le paramètre n38 (erreur externe :	
	<b>S5</b>	Entrée multifonction 4 (S5)	Définie par le paramètre n39 (référence à pas multiples 1)	
	CS	Commun d'entrée de séquence	Commun pour S1 à S5	
	FS	Alimentation de la fréquence de référence	Alimentation c.c. pour la fréquence de référence	0 à 10 Vc.c. (impédance d'entrée : 20 kΩ)
	FR	Entrée de fréquence de référence	Borne d'entrée pour la fréquence de référence	
	FC	Commun de fréquence de référence		

**Tableau 2. 2.** Bornes de circuit du contrôle

SW7 peut être modifié en fonction de la polarité du signal d'entrée de séquence (S1 à S5).

0V commun: côté NPN (réglage d'usine)

+24 commun: côté PNP

Si la commande de marche FWD (REV) est donnée pendant la sélection de référence de fonctionnement (n02 = 1) à partir de la borne du circuit de commande, le moteur démarre automatiquement après que l'alimentation d'entrée du circuit principal est activée [24].

➤ **Test exécuté:** L'onduleur fonctionne en réglant la fréquence (vitesse). Il existe trois types de modes de fonctionnement pour le VS mini:

- Exécuter la commande depuis la console numérique (potentiomètre / réglage numérique).
- Exécuter la commande depuis le terminal (bornes du circuit de contrôle) (figure 23) du circuit de commande.
- Exécuter la commande à partir d'une ligne de communication (MEMOBUS)

**Les références de fonctionnement ou les constantes de référence de fréquence peuvent être sélectionnées séparément comme indiqué ci-dessus.**

N° paramètre	Nom	Description
n01	Sélection d'interdiction d'écriture de paramètres / initialisation de paramètres	Sert à interdire l'écriture de paramètres, régler des paramètres ou modifier la plage de surveillance de paramètres. Sert à rétablir la valeur par défaut des paramètres. 0 : Règle ou surveille le paramètre n01. Les paramètres n02 à n79 peuvent uniquement être surveillés. 1 : Règle ou surveille les paramètres n01 à n79. 6 : Efface le journal d'erreurs. 8 : Rétablit la valeur par défaut des paramètres en séquence à 2 fils. 9 : Rétablit la valeur par défaut des paramètres en séquence à 3 fils.
n02	Sélection de la commande de fonctionnement	Sert à sélectionner la méthode d'entrée des commandes RUN et STOP en mode distant. 0 : Les touches RUN et STOP/RESET de la console numérique sont activées. 1 : Entrée multifonction via les bornes du circuit de contrôle en séquence à 2 ou 3 fils. 2 : Les commandes de fonctionnement via une communication RS-422A/485 sont activées.
n03	Sélection de la fréquence de référence	Sert à définir la méthode d'entrée de la fréquence de référence en mode distant. 0 : Console numérique 1 : Fréquence de référence 1 (n21) 2 : Borne du circuit de contrôle de fréquence de référence (0 à 10 V) 3 : Borne du circuit de contrôle de fréquence de référence (4 à 20 V) 4 : Borne du circuit de contrôle 41 de fréquence de référence (0 à 20 V) 6 : Fréquence de référence via

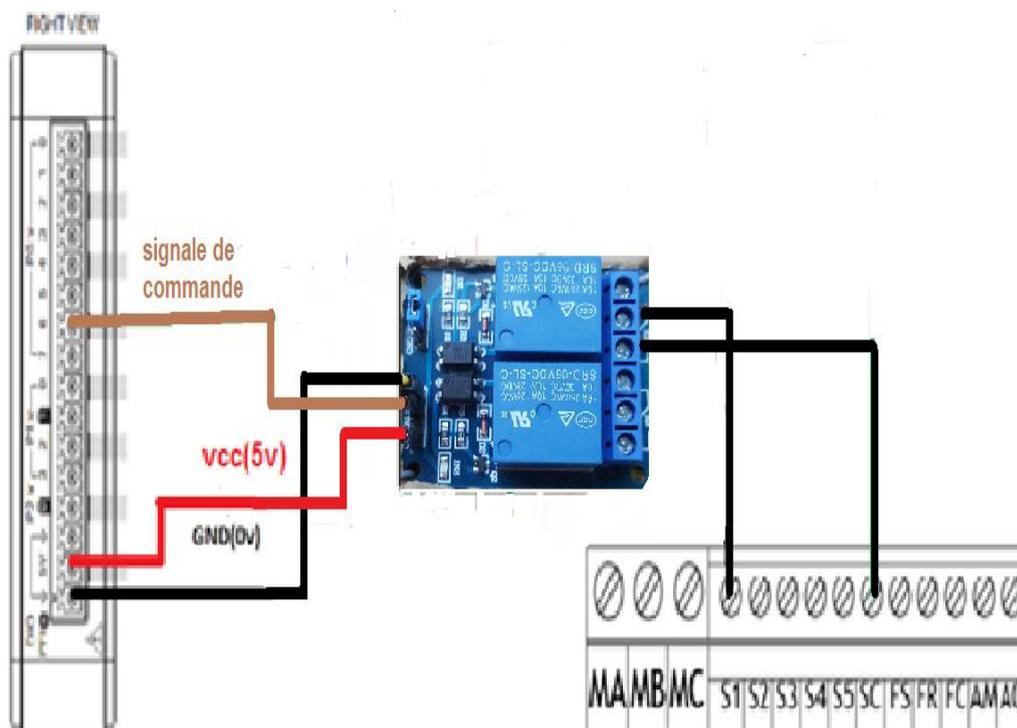
		communication RS-422A/485
n16	Temps d'accélération	Temps d'accélération : temps nécessaire pour passer de 0 à 100 % de la fréquence maximale.
n17	Temps de décélération	Temps de décélération : temps nécessaire pour passer de 100 à 0 % de la fréquence maximale.
n21	Fréquence de référence	Sert à régler les fréquences de référence internes. Remarque : La fréquence de référence 1 est activée en mode distant en réglant le paramètre n03 (sélection de la fréquence de référence) sur 1

**Tableau 2. 3.** L'utilisation de l'opération numérique

**Remarque : Avant d'utiliser la console numérique, nous assurons que le potentiomètre de fréquence (FREQ) est réglé sur MIN.**

Nous contrôlons l'affichage de la console (IOUT ou surveillance multifonction U03) pour assurer que le courant de sortie n'est pas excessif [23].

La figure 2.24 montre le câblage que nous avons réalisé pour faire démarrer et arrêter le moteur à travers le port de sorties de NI USB-6001 (port6), et varie la vitesse via le VS mini J7:



**Figure 2. 24.** Câblage de démarrage par bouton

**a. Fonctionnement de montage**

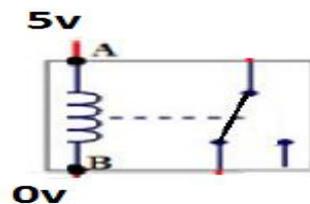
Pour activer le relais il faut Le signal de commande d'entrée est de 0V, le relais donc est allumé, c.-à-d. reçoit une différence de potentielle de 5v.

On a ( $V_A - V_B = V_L$ )

Pour que  $V_L$  soit égale à 5v il faut que le  $V_B$  soit égale à 0v. La tension  $V_B$  est générée par le port P0.6 de NI-USB-6001.

Tandis que le signal de commande d'entrée est de 5V, le relais est éteint. C.-à-d. reçoit une différence de potentielle de 0v.

Pour que  $V_L$  soit égale à 0v il faut que le  $V_B$  soit égale à 5v. La tension  $V_B$  est générée par le port P0.6 de NI-USB-6001.



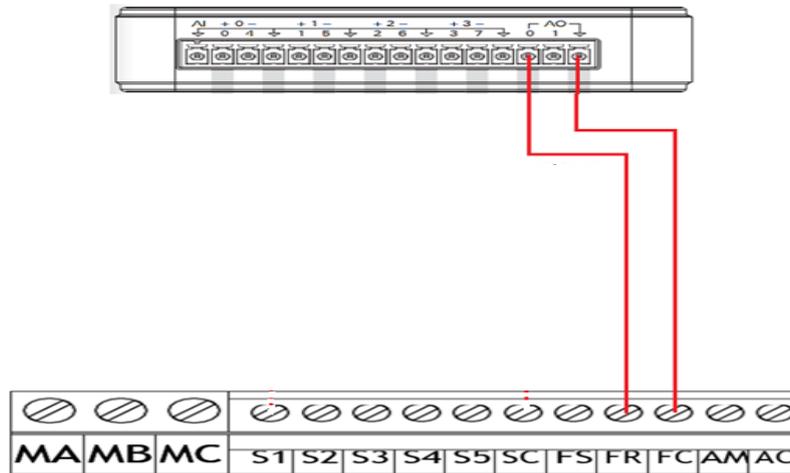
**Figure 2.25.** Relais

D'après le constructeur du variateur est conformément au paramètre numéro 2 qui est à «1» le moteur démarre par la mise en connexion de la borne SC avec S1 (voir le tableau 2.3).

**2.2.7 Utilisation des sorties analogiques AO**

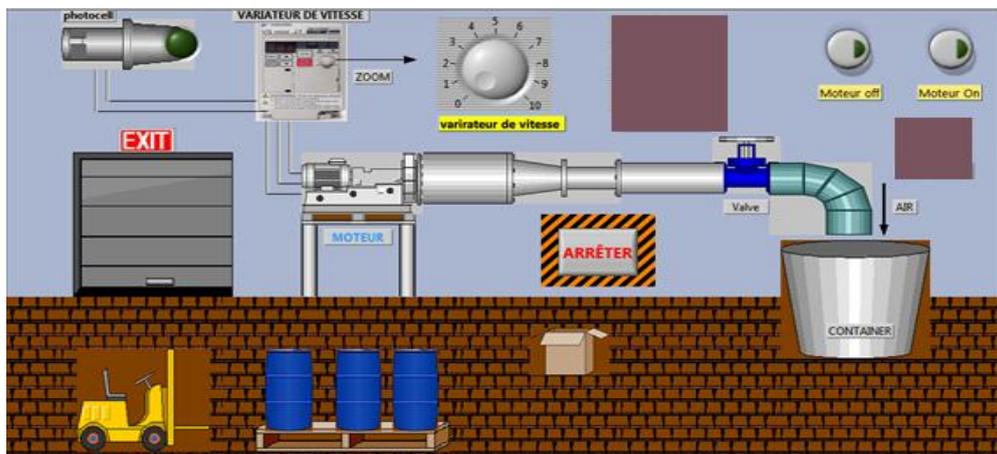
- Nous avons fait un travail qui permet de faire varier la vitesse du moteur à travers le potentiomètre de face avant de LabVIEW ;
- Le NI-USB-6001 délivre un signal analogique de sortie qui permet de faire varier la vitesse du moteur à travers l'OMRON VS mini J7 ;
- Nous avons utilisé l'une de ces sorties (ao0) pour varier la vitesse à partir de la face avant de LabVIEW ;
- Nous avons utilisé le cas référentiel ;

Le montage de ce travail est présenté dans la (Figure 2.26).



**Figure 2. 26.** Câblage pour la variation de vitesse de moteur

Dans notre cas lorsque nous appuyons sur **moteur on** de la face avant de LabVIEW provoque le démarrage du moteur, et lorsque nous appuyons sur **moteur off** provoque l'arrêt du moteur (figure 2.27), en peut aussi varie la vitesse d'un moteur à travers le bouton de **variateur de vitesse**.



**Figure 2. 27.** La commande du moteur sur la face avant

### 2.2.8 Utilisation des entrées analogiques

Le NI-USB-6001 dispose de 8 entrées analogiques. Ces derniers peuvent être configurée soit en différentielle soit en référentielle.

Pour utiliser le cas différentielle, nous avons branché un thermocouple de type K au niveau d'une entrée analogique (a0, a4) doté d'un convertisseur qui délivre un signal analogique de 4mA à 20 mA, et nous avons branché un détecteur de pression à (a1, a5) qui permet également de délivré un signal analogique de 4mA à 20 mA et nous avons utilisé ces capteur pour tester les entrées analogiques différentielles.

### b. Thermocouple

Un thermocouple est un capteur servant à mesurer la température. Il se compose de deux métaux de natures différentes reliés à une extrémité. Quand la jonction des métaux est chauffée ou réfrigérée, une tension variable est produite, qui peut être ensuite transcrite en température. Les alliages thermocouple sont généralement disponibles en fils

Les thermocouples sont des capteurs de température polyvalents et sont, par conséquent, communément utilisés pour de nombreuses applications, du thermocouple à usage industriel au thermocouple standard équipant les appareils et les équipements domestiques. (Figure 2.28)



**Figure 2. 28.** Thermocouple

- **Principe de fonctionnement:**

Un thermocouple utilise principalement l'effet Seebeck afin d'obtenir une mesure de la température. Si on réunit à une extrémité deux fils métalliques de natures différentes et que l'on élève la température de cette extrémité, il apparaît une tension  $e_{AB}$  aux extrémités restées libres. Il est possible de déterminer la température de l'extrémité chauffée à partir de la mesure de  $e_{AB}$  [25].

On appelle :

**Soudure chaude** : Jonction de l'ensemble thermocouple soumis à la température à mesurer : c'est la jonction Capteur.

**Soudure froide:** Jonction de l'ensemble thermocouple maintenu à une température connue ou à  $0\text{ }^{\circ}\text{C}$  : c'est la jonction Référence [26].

Le thermocouple que nous avons utilisé est le type à base de métaux usuels (typek) :

**Composition** : Chromel (alliage nickel + chrome) / Alumel (alliage nickel + aluminium)

(5 %) + silicium). Usage continu de 0 °C à 1 100 °C ; usage intermittent de -180 °C à 1 200 °C. Sa table de référence s'étend à 1 370 °C<sup>4</sup>.

➤ **Convertisseur** : La conversion du signal du capteur est faite à partir d'un convertisseur de température thermocouple en signal électrique 4-20 mA (ou 20 - 4 mA) à microprocesseur réglable pour capteur de température thermocouple type K. Il permet de convertir les variations de température relevées par un capteur thermocouple pour une étendue de mesure allant de 0 à +255°C en signal linéaire de courant à 2 fils dans le domaine 4-20 mA, ce qui implique la forme **Y = a\*x + b**

**Si la température égale à 0 °C → le courant égale à 4 mA**

**Si la température égale à 255 °C → le courant égale à 20 mA**

Y : Température

x : Courant

$$y = a * x + b \quad 0 = a * (4 * 10^{-3}) + b \dots \dots \dots (1)$$

$$255 = a * (20 * 10^{-3}) + b \dots \dots \dots (2)$$

C'est un système de deux équations à deux inconnues du premier degré

Méthode de résolution : Le principe général est d'éliminer une inconnue pour se ramener à la résolution d'une équation du premier degré à une inconnue. On distingue la résolution par combinaison.

Principe : On multiplie l'une ou les deux équations par des nombres convenablement choisis de manière à ce que l'une des inconnues disparaisse par addition membre à membre.

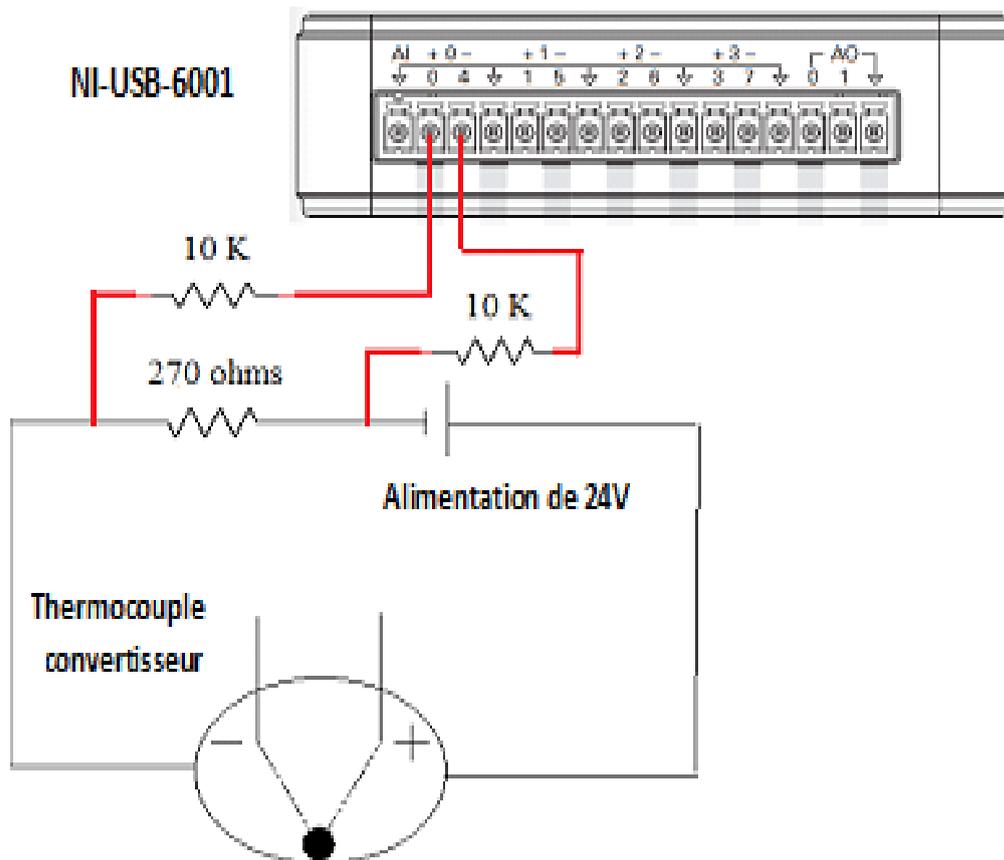
Alors :

$$[0 = a * (4 * 10^{-3}) + b] * (-1)$$

$$(1) + (2) \rightarrow 255 = a * (16 * 10^{-3}) \rightarrow a = 255 / (16 * 10^{-3})$$

$$b = a * (4 * 10^{-3}) - 255 / (16 * 10^{-3}) * (4 * 10^{-3}) = -255 / 4$$

Le montage se présente comme indique la (Figure 2.29) :



**Figure 2. 29.** Câblage de thermocouple

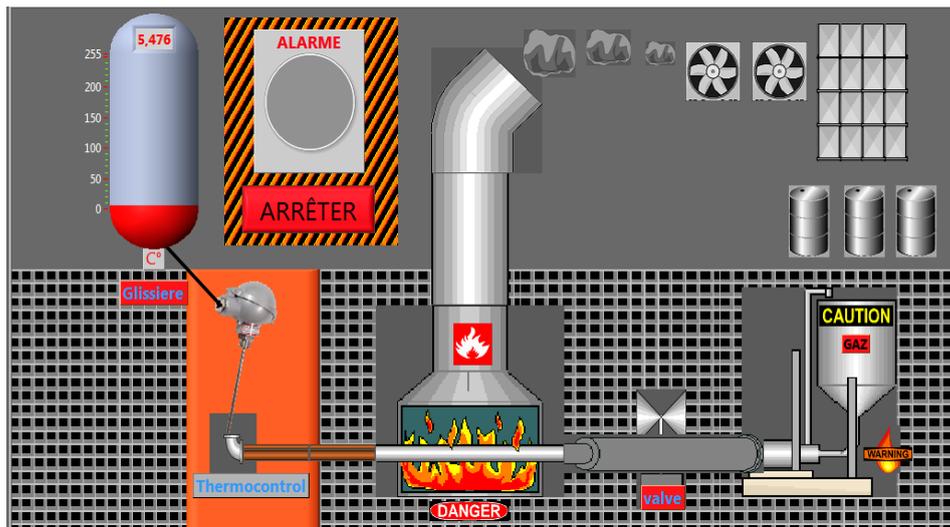
Nous avons connecté le fil positif du thermocouple à la borne positive de l'alimentation de 24v et le fil négatif du thermocouple à la borne de résistances de 270 ohms et le deuxième borne de la résistance est connecté à la borne négative du l'alimentation de 24v, l'entrées analogique différentielle A0 (a0, a4) est connecté au bornes de la résistance de 270 ohms.

**NB : Nous avons ajouté une résistance pull-down de 10 K à chaque entrée analogique pour éliminer la perturbation et le bruit électronique.**

➤ **Résistance pull-down :**

Une résistance de pull-down est connectée à la masse assure que la broche est dans un état bas, tout en utilisant également une faible quantité de courant.

Dans notre projet, lorsque le thermocouple reçoit une température nous aurons un affichage au niveau de la face avant de LabVIEW (figure 2.30).



**Figure 2. 30.** Affichage de mesure de température sur la face avant

**a. Détecteur de pression**

Un capteur de pression est un dispositif destiné à convertir les variations de pression en variations de courant électrique (Figure 2.31).



**Figure 2. 31.** Détecteur de pression

C'est le même principe précédent il permet aussi de convertir les variations de pression relevées par un capteur de pression pour une étendue de mesure allant de -200mbar à +200mbar en signal linéaire de courant dans le domaine 4-20 mA, ce qui implique la forme [27].

$$Y = a \cdot x + b$$

**Si la pression égale à -200mbar → le courant égale à 4 mA**

**Si la pression égale à 200mbar → le courant égale à 20 mA**

Y : Pression

x : Courant

$$Y = a \cdot x + b$$

$$-200 = a \cdot (4 \cdot 10^{-3}) + b \dots \dots \dots (1)$$

$$200 = a \cdot (20 \cdot 10^{-3}) + b \dots \dots \dots (2)$$

C'est un système de deux équations à deux inconnues du premier degré donc :

Le même Principe pour le calcul de la conversion de la température.

Alors :

$$(1) + (2) \rightarrow 0 = a \cdot (4 \cdot 10^{-3}) + b \rightarrow b = -12 \cdot 10^{-3} a$$

$$b = -300$$

$$a = 25000$$

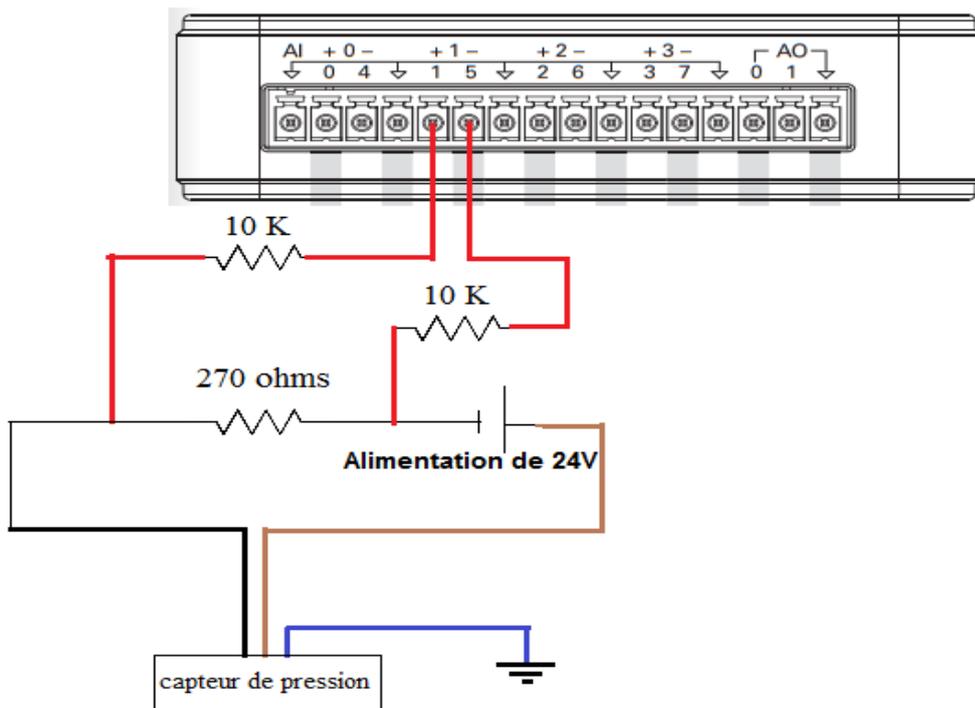
$$y = 25000x - 300$$

➤ **le câblage de périphérique NI DAQ à utiliser avec un capteur de pression :**

L'étape suivante consiste à connecter physiquement le thermocouple à notre périphérique DAQ.

Le fil marron du capteur est une avance positive, le noir est une avance négative, le bleu c'est pour le case qui va à la masse, le blanc c'est NC. Le diagramme de câblage de la (Figure 3.31) indique quelles broches de notre périphérique DAQ.

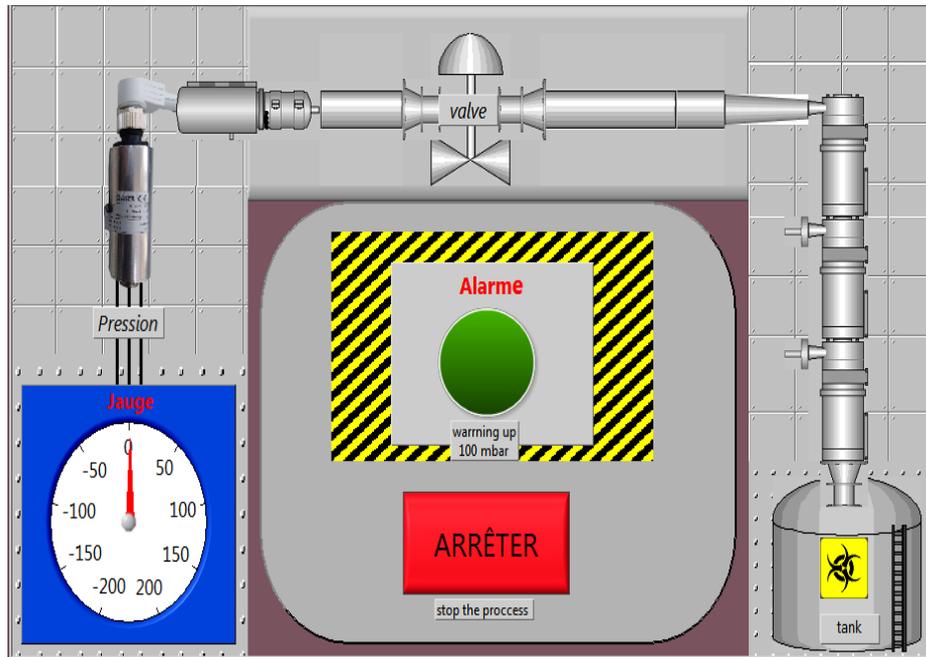
**Remarque : La manière de câblage est la même manière précédente**



**Figure 2. 32.** Câblage du capteur de pression

### Fonctionnement de montage

Dans notre cas, lorsque le détecteur reçoit une pression ou dépression nous aurons un affichage au niveau de la face avant de LabVIEW (Figure 2.33).



**Figure 2. 33.** Affichage de mesure de pression sur la face avant

## 2.3 SCADA LabVIEW PLC S7-312c

### 2.3.1 Introduction:

Cette partie présente la conception d'interfaces de commande graphiques de LabVIEW permettant d'afficher et de contrôler les paramètres du système et programmable par le Siemens S7-312c à l'aide du serveur OPC.

Nous avons réalisé 3 applications de supervision par LabVIEW qui consiste à :

- Contrôler et surveiller le déroulement de programme de l'API ;
- Représenter graphiquement les données acquises ;
- Conserver ces données en fichier lorsque l'application est fermée ;

### 2.3.2 Les feux de circulation



**Figure 2. 34.** API SIEMENS

Nous pouvons également visualiser le carrefour via une maquette se présente comme suit :



**Figure 2. 35.** Maquette de feu rouge

Nous avons développé une petite application qui permet de gérer les couleurs d'une intersection selon deux modes :

➤ **Le premier mode c'est le mode jour**

**Le temps de la séquence :** Nous avons mis un délai de 20s secondes entre le feu vert et le feu orange. Un délai de 4s seconde entre le feu orange et le feu rouge. Et un délai de 20s secondes entre le feu rouge et le feu vert.

**Le bouton S0 permet d'activer le mode jour**

- Allumer les feux verts de la voie 1 et les feux rouges de la voie 2 pour 20 secondes ;
  - Allumer les feux orange de la voie 1 et les feux rouges de la voie 2 pour 4 secondes ;
  - Allumer les feux rouges de la voie 1 et les feux verts de la voie 2 pour 20 secondes ;
  - Allumer les feux rouges de la voie 1 et les feux orange de la voie 2 pour 4 secondes ;
- Refaire le cycle, passer au cycle de S1.

➤ **Le deuxième mode c'est le mode nuit**

Puisque il n'Ya pas trop de circulation au niveau des routes, les feux oranges commence à clignoté, à ce moment-là, la priorité est à droite.

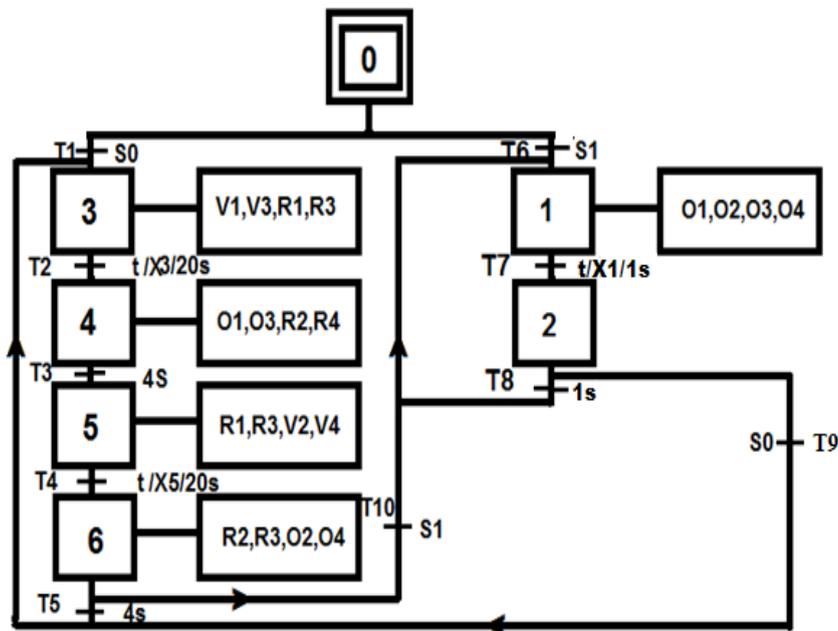
**Le bouton S1 permet d'activer le mode nuit**

- Allumer les feux orange de la voie 1 et la voie 2 pour 1 secondes ;
- Eteindre les feux orange de la voie 1 et la voie 2 pour 1 secondes ;

Refaire le cycle, passer au cycle de S0.

Le changement du mode peut être fait à travers de la face avant de LabVIEW.

Avant de faire le programme sur le TIA portal, nous avons commencé par un grafcet se présente comme suit :



**Figure 2. 36.** GRAFCET de feu rouge

**2.3.3 variation de vitesse d'un moteur**

**a. potentiomètre**

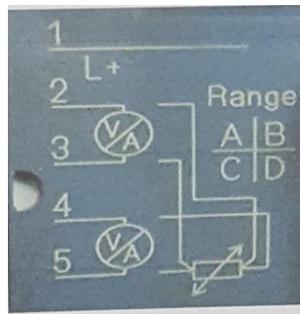
- Le potentiomètre est une résistance variable utilisées pour diviser une tension, contrôler un niveau, compenser des erreurs ....etc ;
- L'utilisation de potentiomètre c'est pour varie la vitesse d'un moteur à l'aide de API s7-312c ;
- Le potentiomètre qu'on a utilisé est linéaire de 220 kohms en un seul tour, appliquée pour une plage de variation de la tension entre 0 à 5v ; (Figure 2.37)



**Figure 2.37.** Potentiomètre

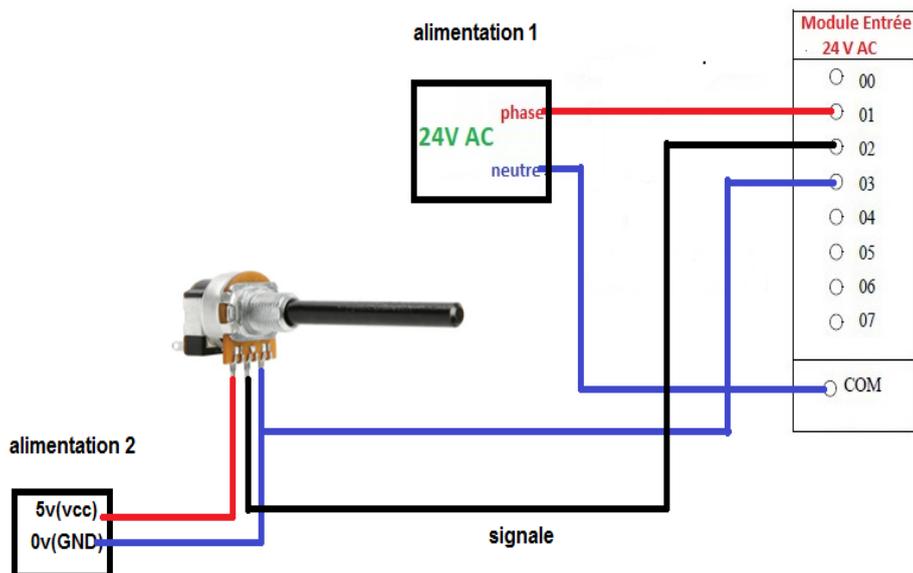
- Le potentiomètre est alimenté par 5v ce qui représente seul max du potentiomètre à travers le variateur de fréquence et deux fils d'information vers le module d'entrée analogique, l'un est relié à la borne 2 et l'autre à la borne 3 vers la masse.

**b. Le module d'entre analogique :** les borne 2 et 3 réservé au potentiomètre, et les bornes 1 et 20 réservé au l'alimentation de module AI (figure 2.38).



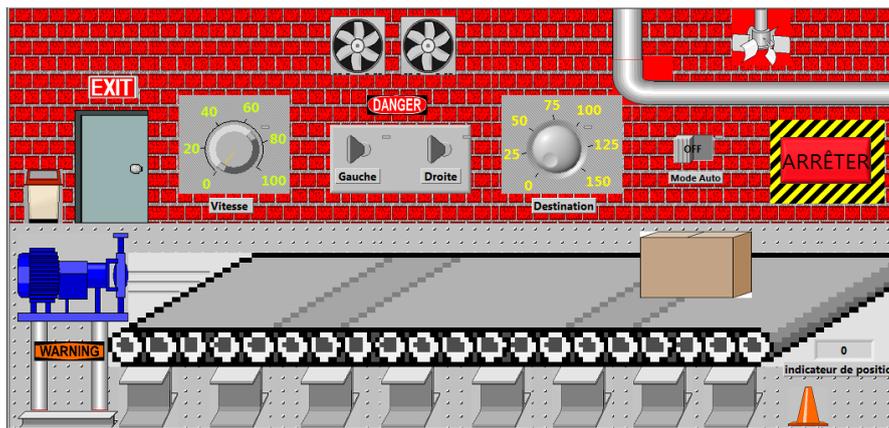
**Figure 2.38.** Schéma de fonctionnement de AI

**c. Schéma de Branchement**



**Figure 2.39.** Schéma de branchement de potentiomètre avec AI

### 2.3.4 Le Chariot



**Figure 2.40.** Affichage et commande de chariot

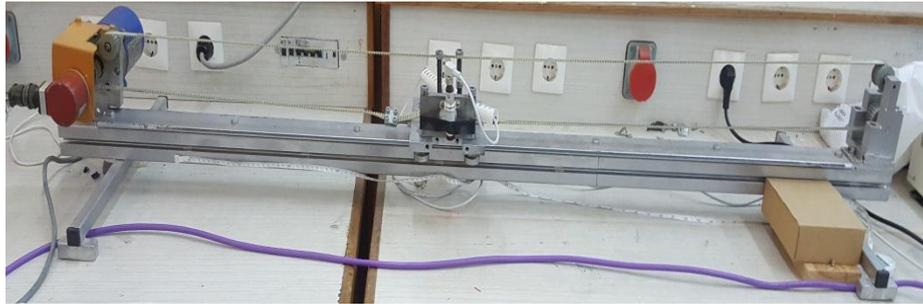


**Figure 2.41.** Montage de chariot

#### a. Principe de fonctionnement :

Dans cette partie on peut visualiser et contrôler le fonctionnement et le mouvement d'un chariot à partir de deux modes.

- **Mode manuel** : lorsque on choisit la vitesse entre 0 et 100 qui équivalent à 0Hz et 60Hz et lorsque on appuie sur le bouton gauche ou bien la droite, le chariot se déplace selon la vitesse et la direction que nous choisissons.
- **Mode automatique** : on active le mode automatique et on choisit la destination entre 0 et 130 cm, on remarque que le chariot se déplace sans utiliser le bouton gauche et droit et sans choisir la vitesse (**vitesse et la direction automatique**).



*Figure 2.42. Le Chariot*

## **2.4 communication entre 3 API s7-312c**

### **2.4.1 Introduction**

Pour faire communiquer 3 CPU Siemens, on a besoin de réseau d'interface. Comme réseau d'interface, on peut avoir le MPI, le Profibus ou le Profinet. Cependant, dans certains cas le CPU utilisé peut ne pas disposer de ports Profibus ou Profinet, il faudra donc dans ce cas de figure utiliser un processeur de communication ou CP. On distingue plusieurs processeurs de communication adaptés aux différentes gammes d'automates (des CP avec port Profibus ,Profinet, MPI etc...) [28].

Dans notre projet nous avons utilisé comme réseau d'interface un réseau MPI ,Ce qui nous permettant de donner à chaque CPU sa propre adresse MPI, et communiquer entre 3 CPU facilement.

### **2.4.2 Bus de terrain profibus**

#### **a. Présentation**

- Procès Field BUS.
- Projet allemand : Bosch, Siemens, ...
- Standardisation : DIN 19245 (1991), EN50170 (1996), IEC 61158 (1999).

#### **b. variantes**

- **Profibus FMS** –Fieldbus Message Specification (Messagerie industrielle entre automates), le premier à avoir été utilisé.
- **Profibus PA** –Process Automation (Alimentation et signal sur un même câble, jusqu'à 31.25 kbits/s.)
- **Profibus DP** – Decentralized Periphery (Jusqu'à 12 Mbits/s.)

**c. Câble**

Paire torsadée blindée, soigneusement isolée par une feuille conductrice et une tresse. Avec 2 conducteurs nommés A et B.

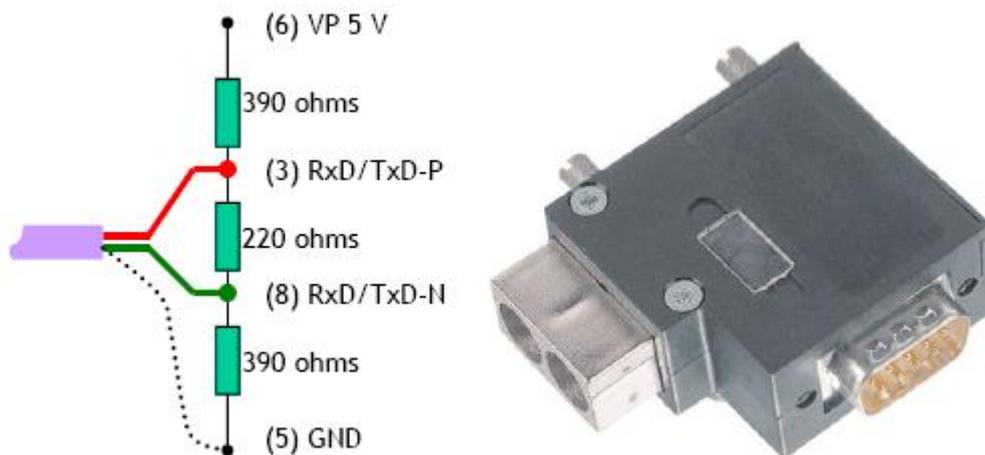
- Fil vert : A, transporte le signal RxD/TxD-N
- Fil rouge : B, transporte le signal RxD/TxD+N



*Figure 2.43.* profibus

**d. Terminaison de ligne**

Résistances de terminaison équivalentes à l'impédance du câble pour polariser la ligne en l'absence de signal. En général, intégrées dans le connecteur et activables par un Interrupteur.



*Figure 2.44.* Connecteur de profibus

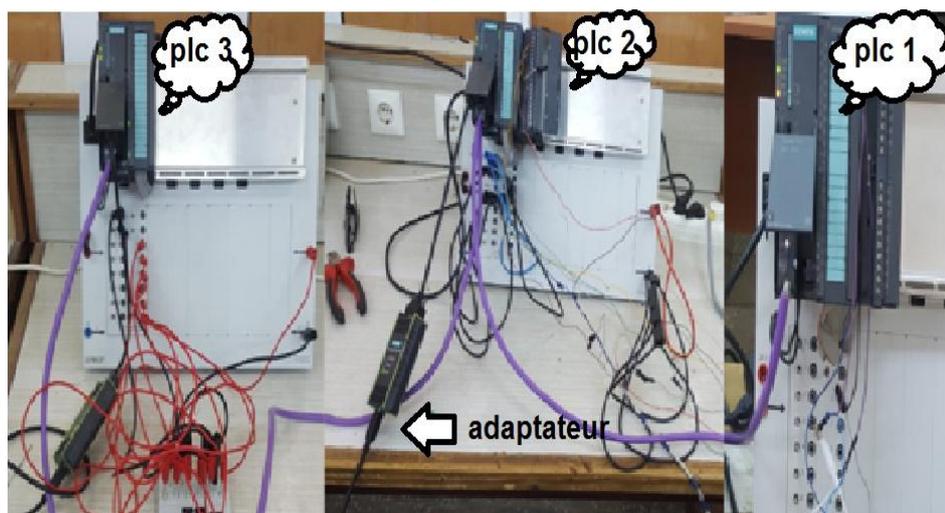
**e. Débit et distance de Profibus**

La distance maximale et le débit sont liés, le bus accepte jusqu'à 32 équipements sans répéteur et 126 équipements avec répéteur. L'utilisation d'un répéteur régénérant le signal permet de cascader les segments. Il ne doit pas y avoir plus de 9 répéteurs entre un équipement et le maître [29]. (Voir la Figure 2.45).

Débit (kbits / s)	Longueur de segment (m)
9.6 ; 19.2 ; 45.45 ; 93.75	1200
187.5	1000
500	400
1500	200
3000 ; 6000 ; 12'000	100

*Figure 2.45.* Débit et longueur de segment

### 2.4.3 Branchement



*Figure 2.46.* Branchement de 3 plcs

## 2.5 Conclusion

Dans ce chapitre nous avons défini tous les composants électroniques et développé les détails de la démarche pour conception et réalisation d'un mini SCADA a base de LabVIEW et la ligne de communication entre l'API S7-312c de siemens et LabVIEW de national instrument et aussi nous avons défini et détaillé les programmes des 3 API et comment communiquer entres eux ,En utilisant le réseau MPI et câble profibus ,nous allons aborder la partie suivante qui représente la partie de programmation.

## Chapitre 3 Programmation et résultats obtenus

---

### 3.1 Introduction

Après avoir expliqué l'essentiel des outils utilisés dans notre projet et la procédure suivie pour le mettre en œuvre dans les précédents chapitres, ce chapitre a pour principale vocation d'exposer et de décrire la méthode que nous avons utilisée lors de la configuration et la programmation au niveau de LabVIEW et le TIA Portal.

### 3.2 Système d'acquisition

#### 3.2.1 NI-DAQmx:

Depuis le développement de NI-DAQmx, les utilisateurs du matériel d'acquisition de données (DAQ) de National Instruments tirent pleinement parti de ses nombreuses fonctionnalités conçues à la fois pour gagner du temps de développement et améliorer les performances de leurs applications d'acquisition de données.

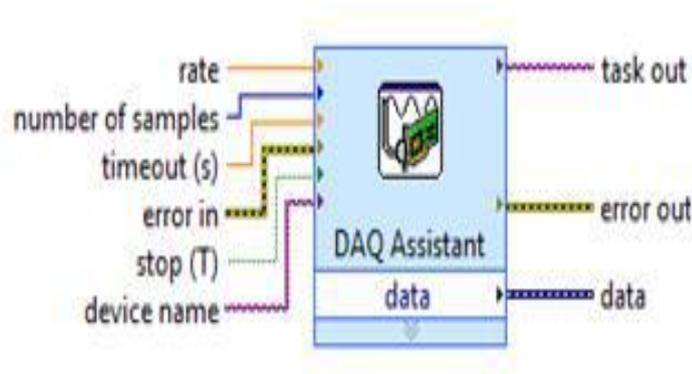
L'interface de programmation d'application (IPA) NI-DAQmx, qui est la même pour les fonctionnalités de périphérique et les familles de périphériques, est une fonctionnalité qui permet d'économiser un temps de développement considérable. Cela signifie que toutes les fonctionnalités d'un appareil multifonction sont programmées avec le même ensemble de fonctions (entrée analogique, sortie analogique, E/S numérique et compteurs). De plus, un dispositif d'E/S numérique et un dispositif de sortie analogique sont programmés en utilisant ce même ensemble de fonctions. Dans LabVIEW, cela est possible en raison du polymorphisme. Un VI polymorphe accepte plusieurs types de données pour un ou plusieurs terminaux d'entrée et / ou de sortie. L'API NI-DAQmx est également cohérente dans tous ses environnements de programmation applicables.

Une autre caractéristique dans le NI-DAQmx qui améliore notre expérience de développement est DAQ Assistant. Cet outil nous aide à créer nos applications sans programmation grâce à une interface graphique permettant de configurer des tâches d'acquisition de données aussi simples que complexes.

De plus, la synchronisation, un processus généralement difficile à mettre en œuvre car les signaux de déclenchement et / ou d'horloge doivent être routés manuellement, est

Facile avec NI-DAQmx, qui effectue automatiquement le routage des signaux entre les différentes zones fonctionnelles d'un même périphérique et entre plusieurs périphériques [30].

**DAQ Assistant** : DAQ Assistant est une interface graphique permettant de créer, d'éditer et d'exécuter de manière interactive des canaux et des tâches virtuels NI-DAQmx (figure 3.1).



**Figure 3.1.** DAQ Assistant

Un canal virtuel NI-DAQmx se compose d'un canal physique sur un périphérique DAQ et des informations de configuration pour ce canal physique, telles que la plage d'entrée et la mise à l'échelle personnalisée. Une tâche NI-DAQmx est une collection de canaux virtuels, d'informations de synchronisation et de déclenchement, ainsi que d'autres propriétés concernant l'acquisition ou la génération. Dans la figure suivante, l'Assistant DAQ est configuré pour effectuer une mesure de contrainte finie [29].

### **3.2.2 Configuration**

#### **a. Entrée digital**

Nous avons configuré l'entrée digitale DI0 qui correspond le mode de sécurité d'un moteur à partir d'une photocellule dans un DAQ selon les étapes suivantes :

- Avec DAQ assistant ouvert, nous avons sélectionné **acquiere signal** → **Digital Input** → **Line input** (voire la Figure 3.2)

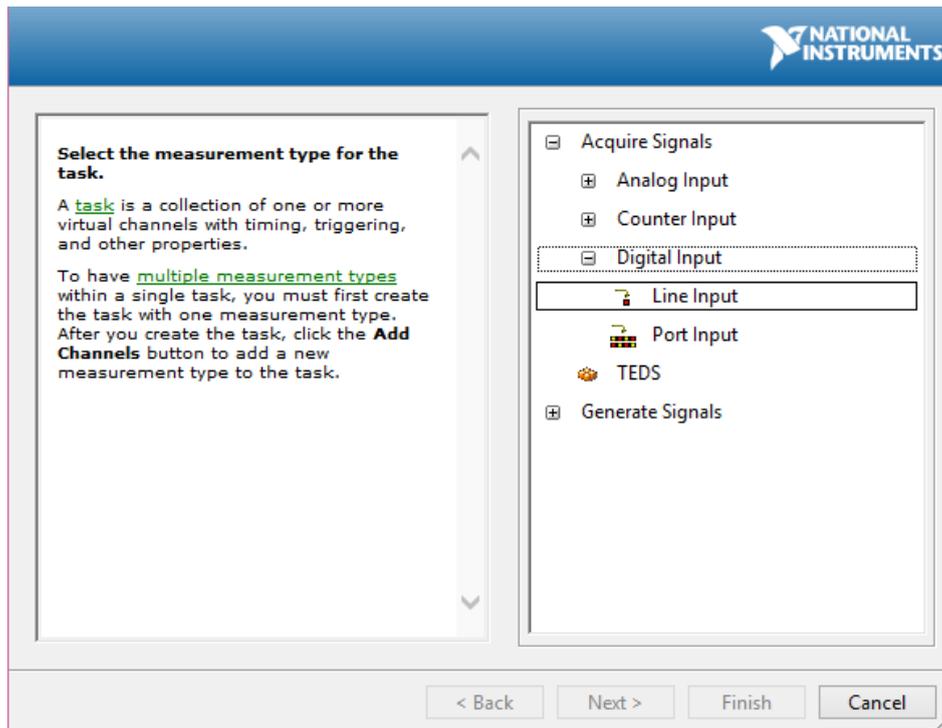


Figure 3. 2. Création d'un canal pour DIO

- ❖ Nous avons sélectionné port0/line0 que nous avons l'intention de connecter à notre photocellule.

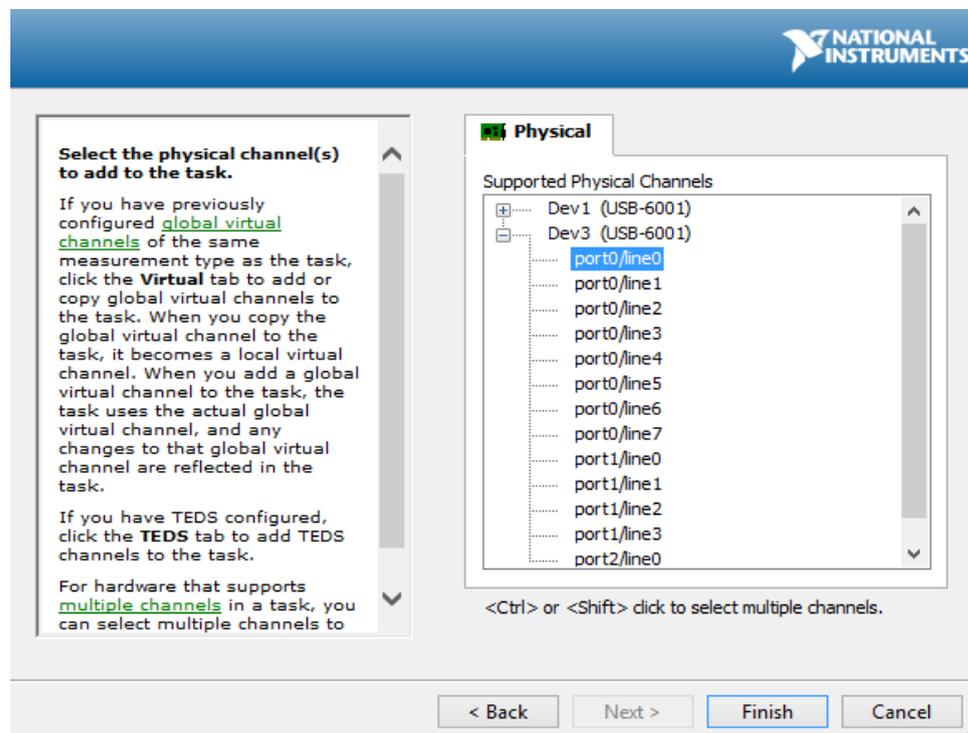
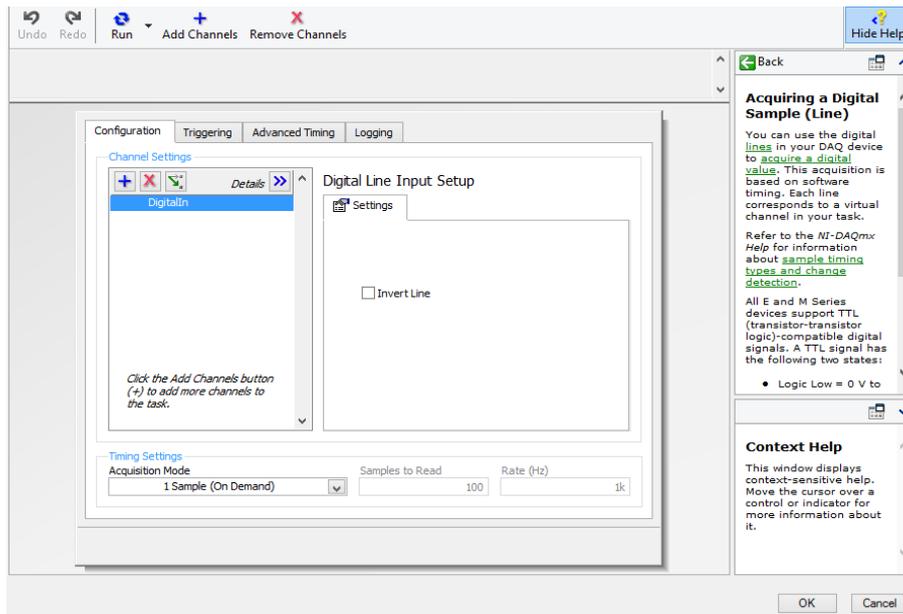


Figure 3. 3. Canal physique DIO



**Figure 3. 4.** Configuration d'un canal de photocellule

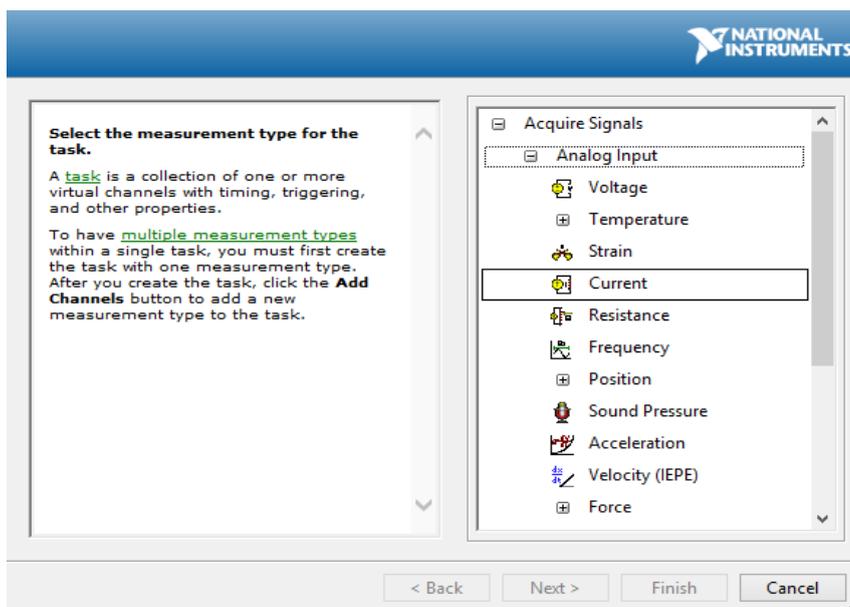
**b. Sorties digitales**

Nous avons configuré la sortie digitale dans un DAQ qui correspond se démarrage du moteur selon les mêmes étapes précédentes (le détail est au niveau de l'annexe B).

**c. Entrées et sortie analogiques**

Nous avons configuré les entrées analogiques (RSE et DIFF) dans un seul DAQ selon les étapes suivantes :

- Avec DAQ assistant ouvert, nous avons sélectionné **acquiere signals** → **Analog Input** → **current** (voire le Figure 3.5).



**Figure 3.5.** Création d'un canal virtuel NI-DAQmx

- ❖ Nous avons sélectionné ai0 que nous avons l'intention de connecter à notre thermocouple.

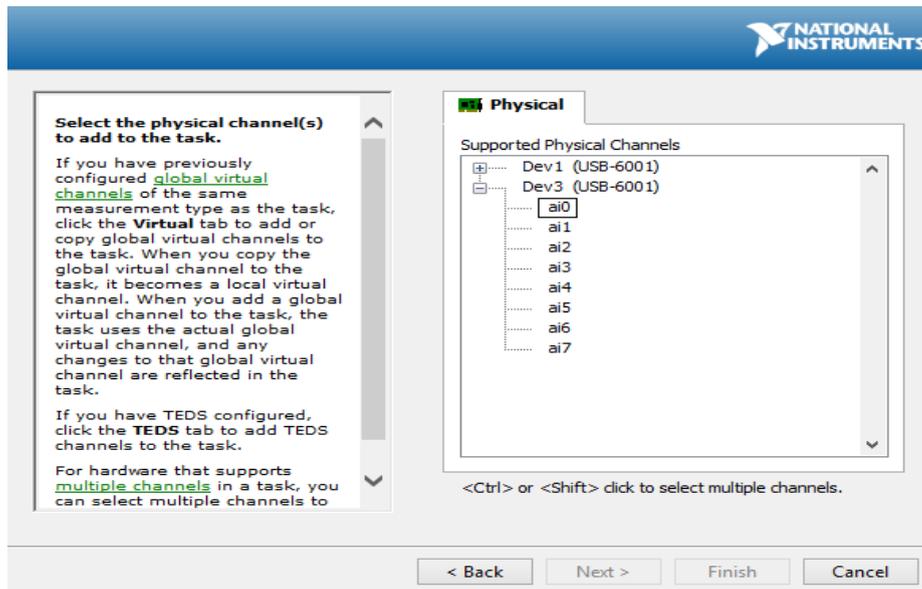


Figure 3. 6. Canaux physiques du périphérique

Un canal physique est un terminal ou une broche à laquelle nous pouvons mesurer ou générer un signal analogique ou numérique. Un seul canal physique peut inclure plus d'un terminal ou une broche, comme dans le cas d'un canal d'entrée à thermocouple différentiel. Dans ce cas, ai0 correspond à CH + et CH- sur le schéma de brochage NI USB-6001. (Voir la figure 3.7)

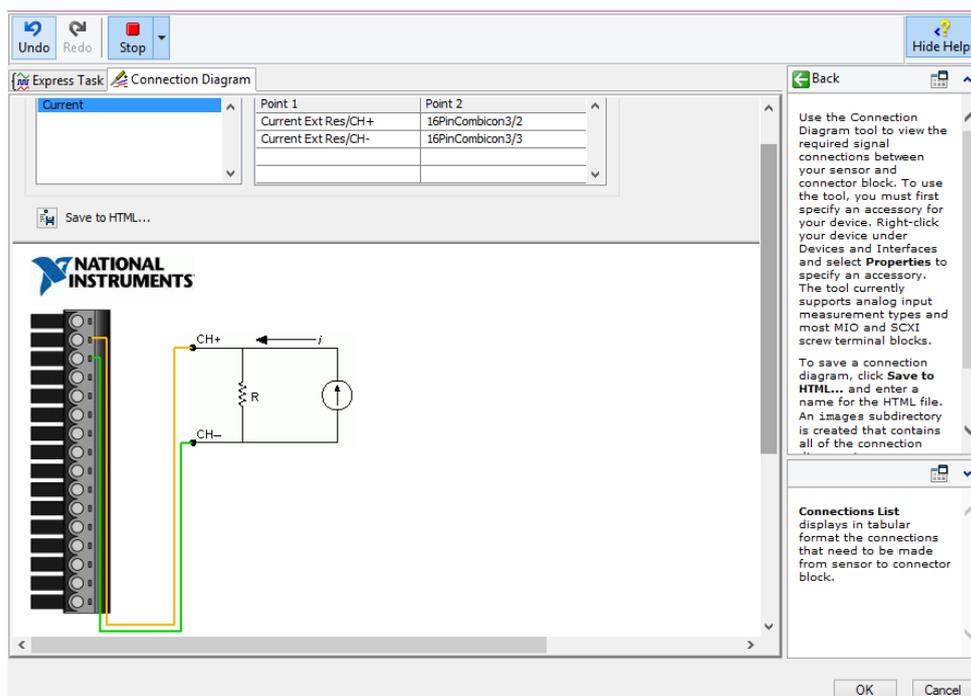
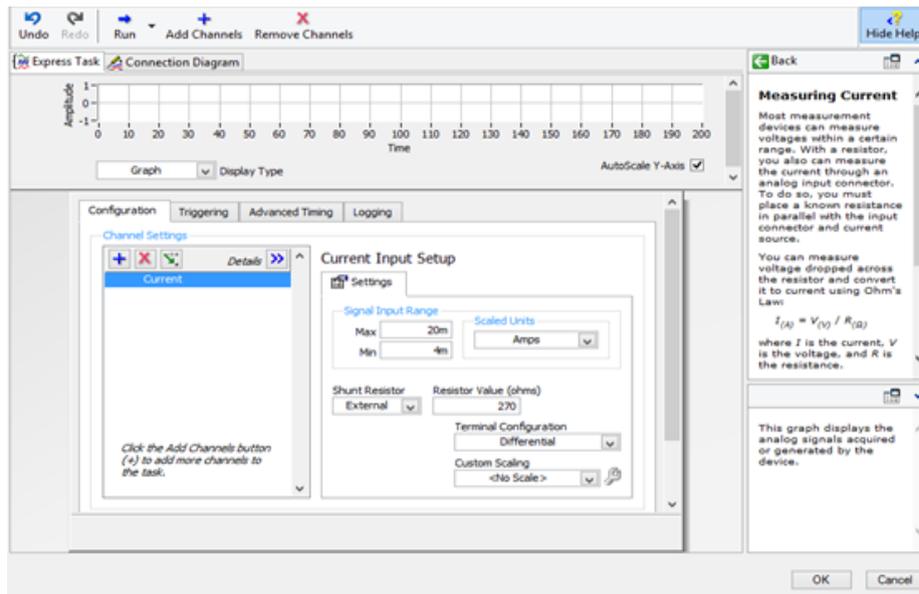
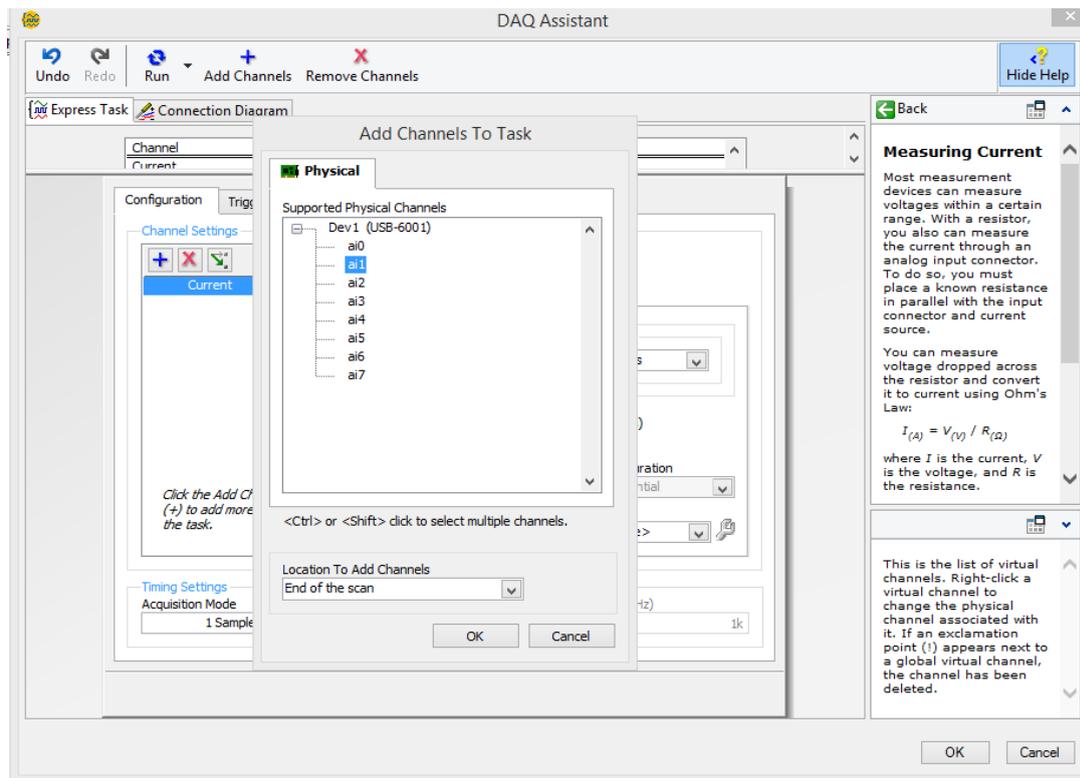


Figure 3. 7. Visualisation du brochage



**Figure 3. 8.** Configuration d'un canal de thermocouple dans MAX

- ❖ Nous avons configuré la mesure de pression dans le même DAQ assistant, pour ce faire nous avons intégré un nouveau canal avec la sélection ai1 :



**Figure 3. 9.** Canal de pression

Pour la configuration de la variation de vitesse du moteur à partir de face avant, nous avons ajouté un nouveau DAQ assistant, la configuration se présente comme suit :

- ❖ Nous avons sélectionné **Generate Signals**→**Analog Output**→**Voltage** (voire la Figure 3.10).

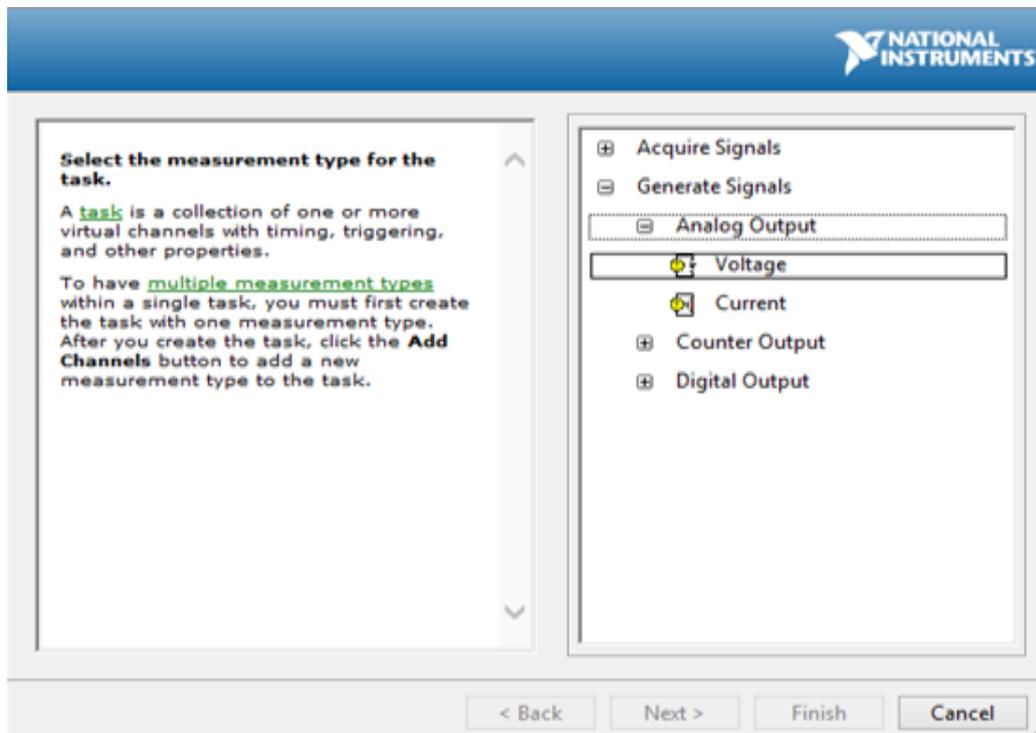


Figure 3. 10. Création d'un canal voltage output

- ❖ Nous avons choisi ao0

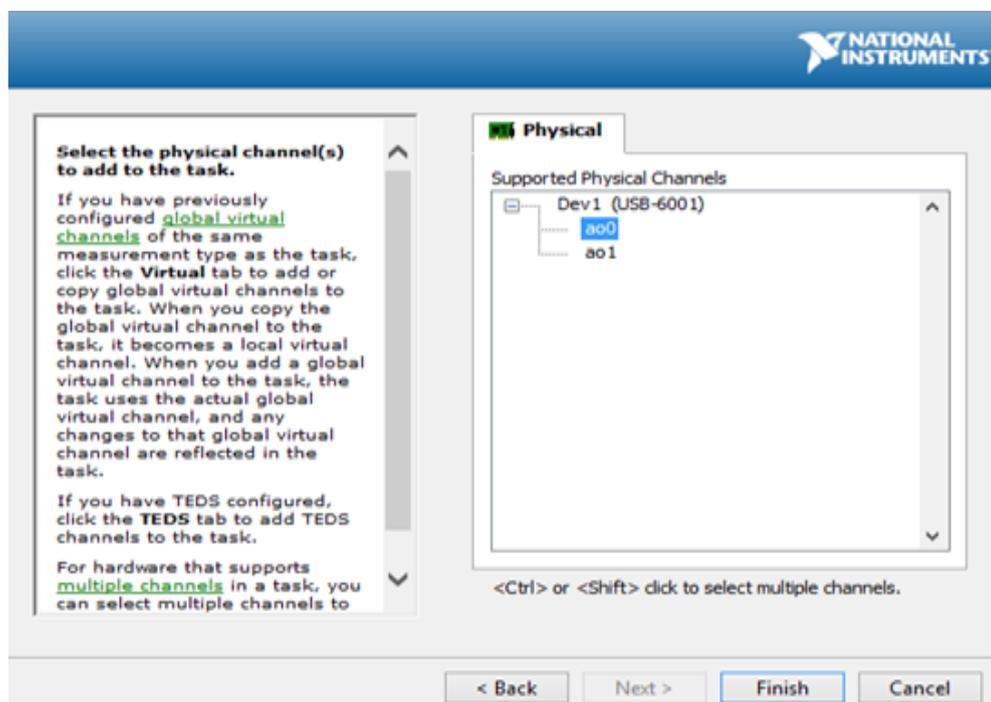
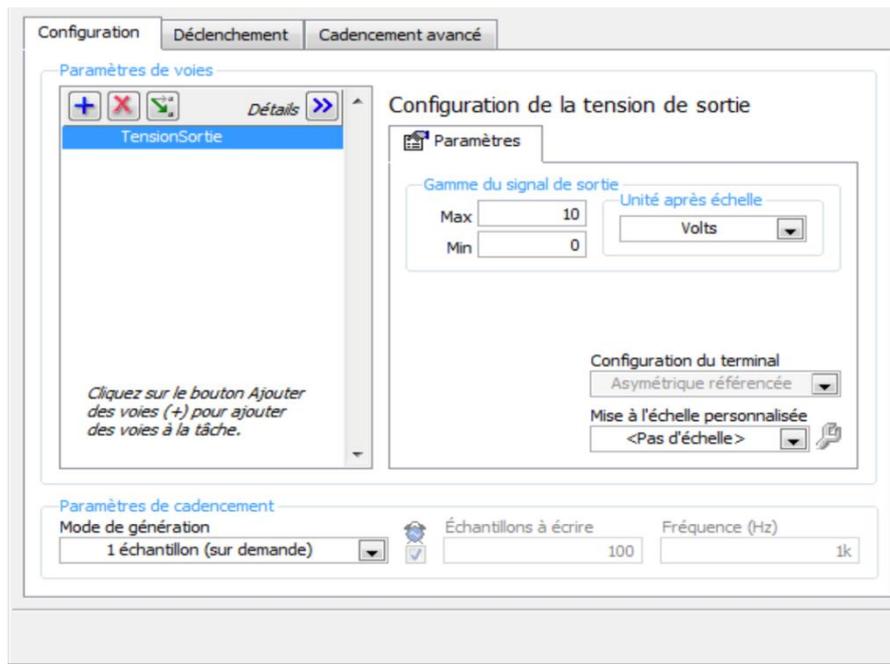


Figure 3. 11. Canal physique ao0

- ❖ Nous avons saisi la valeur minimale et maximale de tension

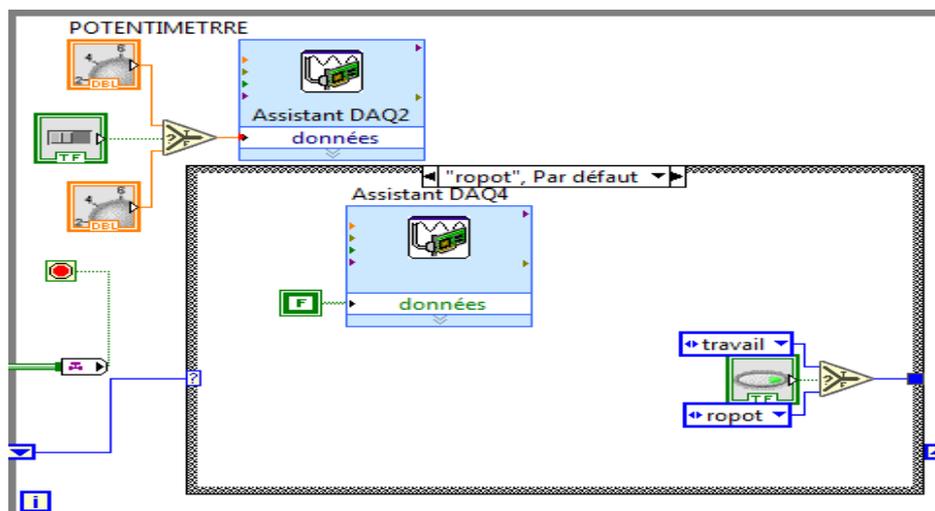


**Figure 3. 12.** Configuration d'un canal de ao0

### 3.2.3 Programmation sur LabVIEW

#### a. E /S digitales

Après la configuration, la programmation de l'entrée digitale que nous avons utilisée pour le démarrage/arrêt de moteur se présente comme suit:

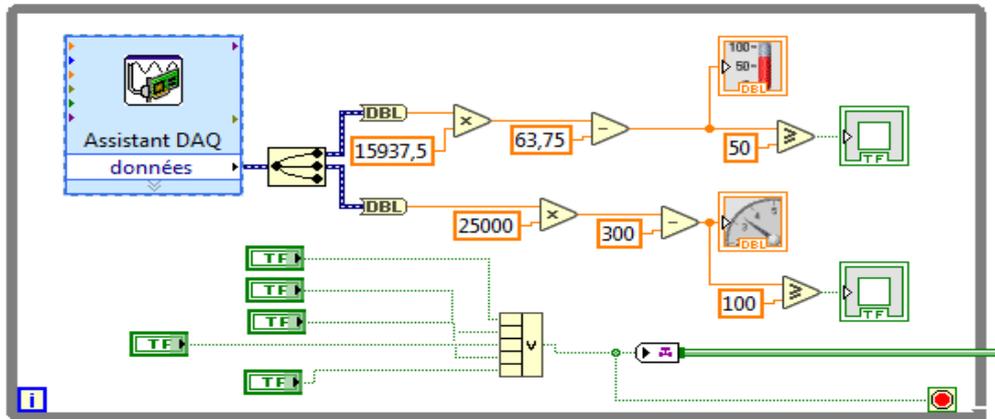


**Figure 3. 13.** Programmation des entrées/sorties digitales

Ce programme est conçu pour permettre de démarrer le moteur à partir d'un bouton sur la face avant de LabVIEW, d'arrêter le moteur en mode de sécurité à partir d'une photocellule.

**b. Entres analogiques**

Après la configuration, la programmation des entrées analogique que nous avons utilisées pour la mesure de la température et la pression se présente comme suit:

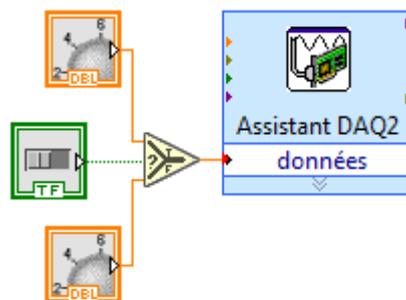


*Figure 3. 14.* programmation des entrées analogique

Ce programme est conçu pour permettre de calculer la conversion effectuée (température/courant et pression/courant) et d’afficher graphiquement sur la face avant du LabVIEW le résultat de mesure (température, pression).

**c. Sorties analogique**

Après la configuration, la programmation de sortie analogique que nous avons utilisées pour varie la vitesse d’un moteur à partir de la face avant du LabVIEW se présente comme suit:



*Figure 3. 15.* programmation des sorties analogique

Nous avons utilisé un interrupteur pour passer de mode à un autre (varié la vitesse a partir de la fac avant de LabVIEW ou bien a partir de potentiomètre qui relié avec l’automate programmable).

### 3.3 SCADA LabVIEW

#### 3.3.1 Programmation sur le TIA Portal

##### a. Variables et équations de feu de circulation

➤ **Temporisateurs :**

- $t/X3/20s \rightarrow To$
- $t/X4/4s \rightarrow To2$
- $t/X5/20s \rightarrow To3$
- $t/X6/4s \rightarrow To4$
- $t/X1/1s \rightarrow To5$
- $t/X2/1s \rightarrow To6$

➤ **Equations de transitions :**

- $T1 = X0 * (s0 + sr0)$
- $T2 = To * X3$
- $T3 = To2 * X4$
- $T4 = To3 * X5$
- $T5 = To4 * X6$
- $T6 = X0 * (s1 + sr1)$
- $T7 = To5 * X1$
- $T8 = To6 * X2$
- $T9 = s0 * x2$
- $T10 = s1 * x6$

➤ **Equations d'étapes :**

- $X0 = int + X0 * (\overline{T1} * \overline{T6})$
- $X1 = T10 + T6 + T8 + X1 + \overline{T7}$
- $X2 = T7 + X2 * (\overline{T9} * \overline{T8})$
- $X3 = (T1 + T5 + T9 + X3) * \overline{T2}$
- $X4 = (T2 + X4) * \overline{T3}$
- $X5 = (T3 + X5) * \overline{T4}$
- $X6 = (T4 + X6) * \overline{T5} * \overline{T10}$

Le Xi est activé lorsque Xi=1 et désactivé lorsque Xi=0. (Voire la figure)

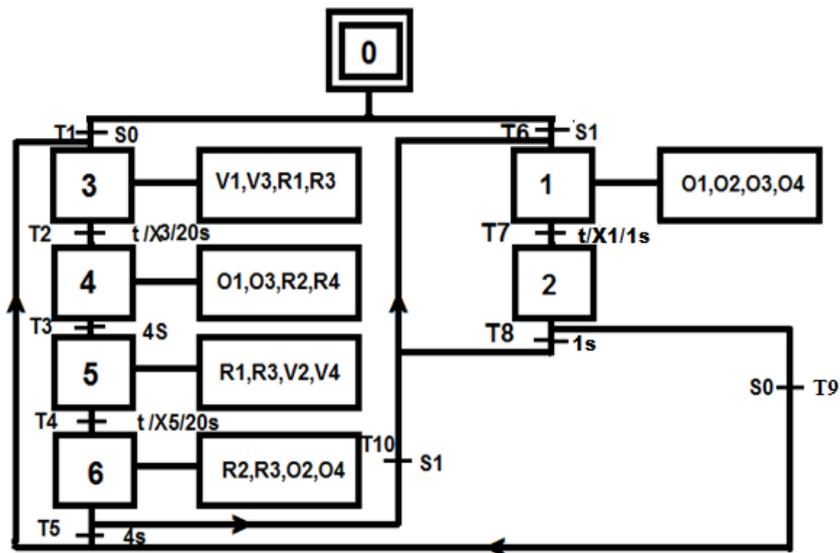


Figure 3.16. GRAFCET de feu de circulation

La table de variables est présentent comme suit :

Table de variables standard							
	Name	Data type	Address	Retain	Visibl...	Acces...	
1	t1	Bool	%M0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	t2	Bool	%M0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	t3	Bool	%M0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	t4	Bool	%M0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	t5	Bool	%M0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	t6	Bool	%M0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	t7	Bool	%M0.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	t8	Bool	%M0.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	t9	Bool	%M1.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	t10	Bool	%M1.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	x0	Bool	%M2.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	x1	Bool	%M2.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	x2	Bool	%M2.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	x3	Bool	%M2.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	x4	Bool	%M2.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	x5	Bool	%M2.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	x6	Bool	%M2.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
18	v1v3	Bool	%Q124.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
19	o1o3	Bool	%Q124.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
20	r1r3	Bool	%Q124.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
21	v2v4	Bool	%Q124.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
22	o2o4	Bool	%Q124.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
23	r2r4	Bool	%Q124.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
24	s0	Bool	%M124.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
25	s1	Bool	%M124.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
26	int	Bool	%M124.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
27	sr0	Bool	%I124.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
28	sr1	Bool	%I124.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figure 3.17. Table de variation

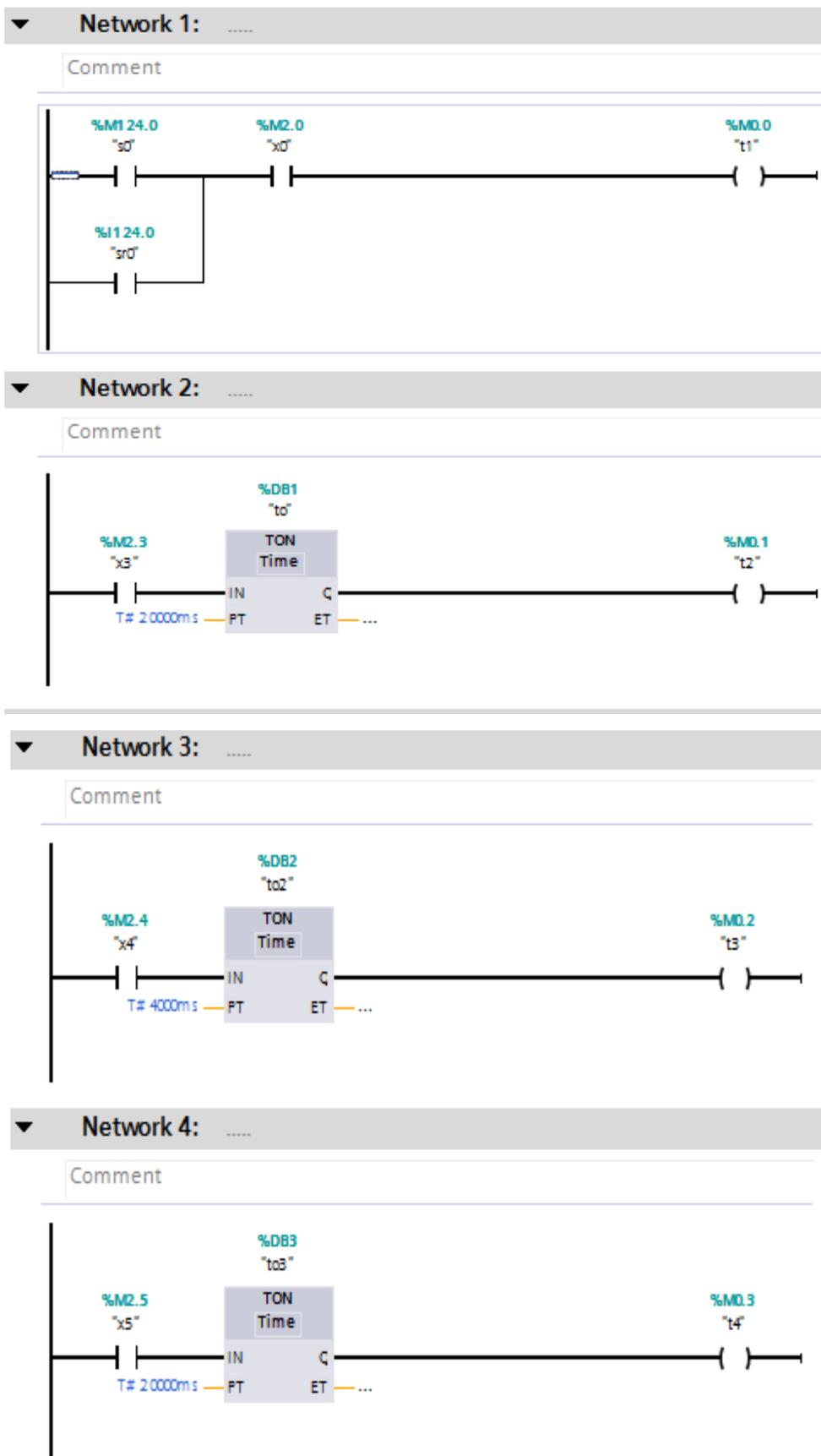


Figure 3.18. Programmation de l'API

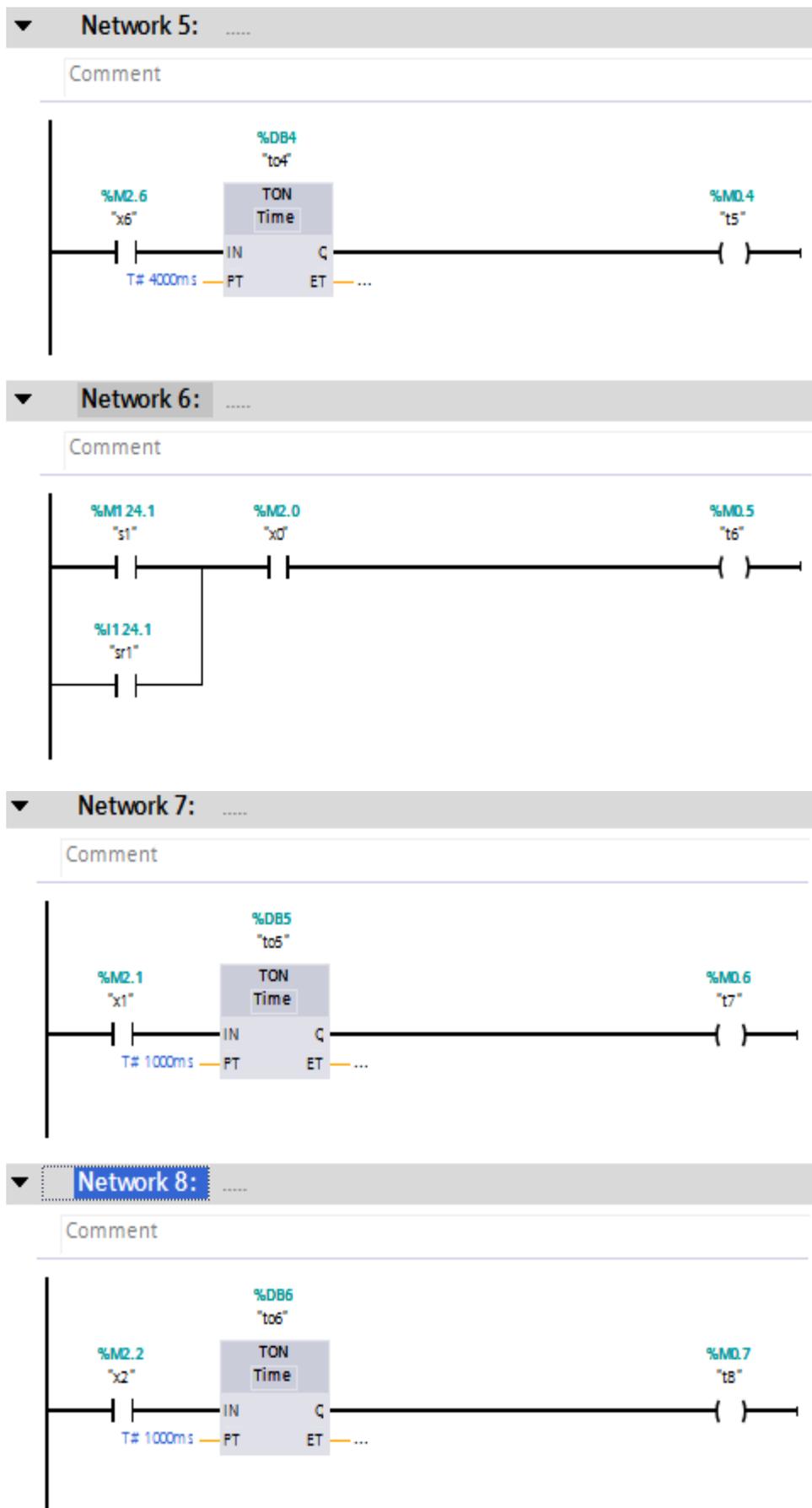


Figure 3.18. Programmation de l'API

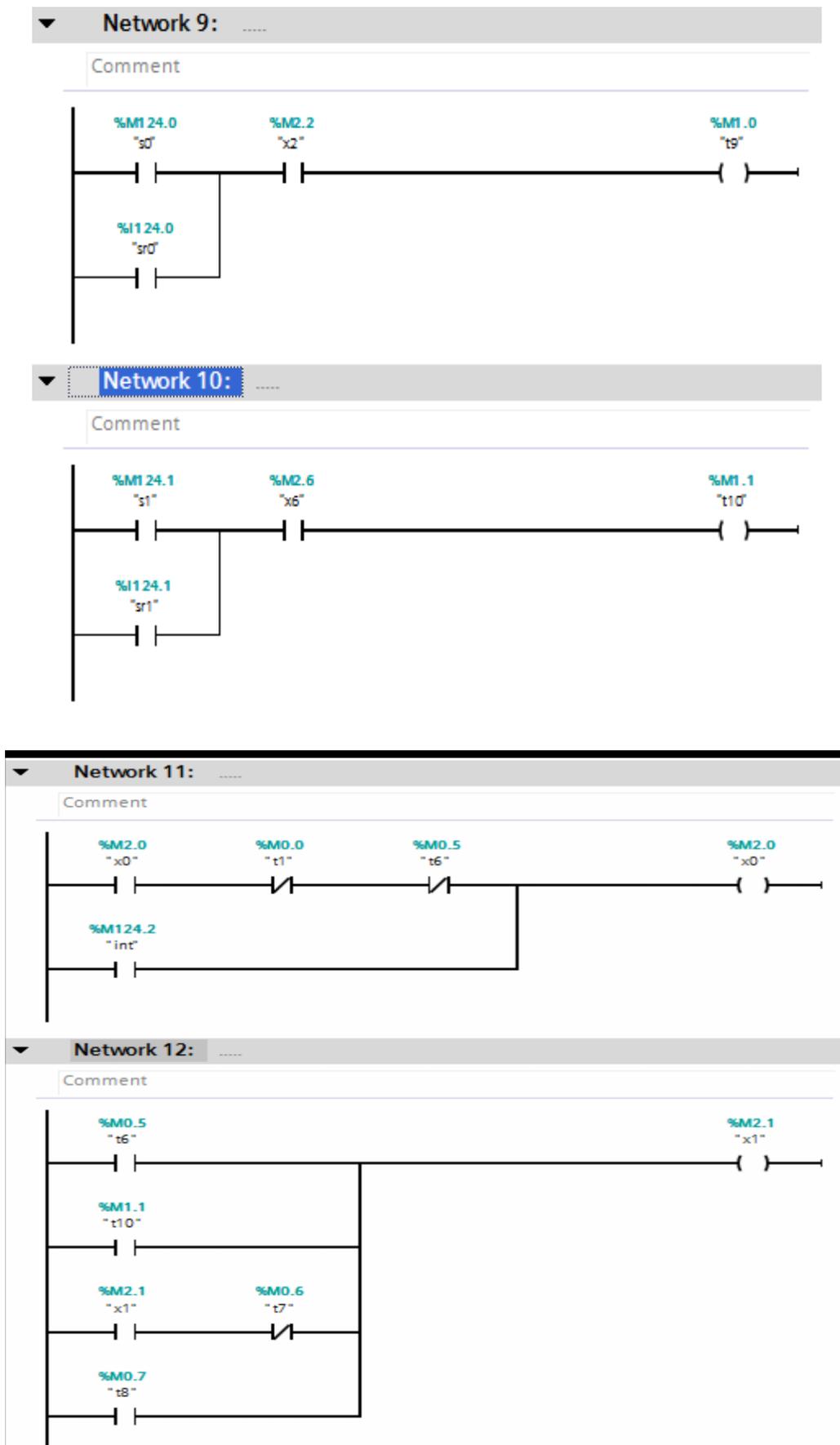


Figure 3.18. Programmation de l'API

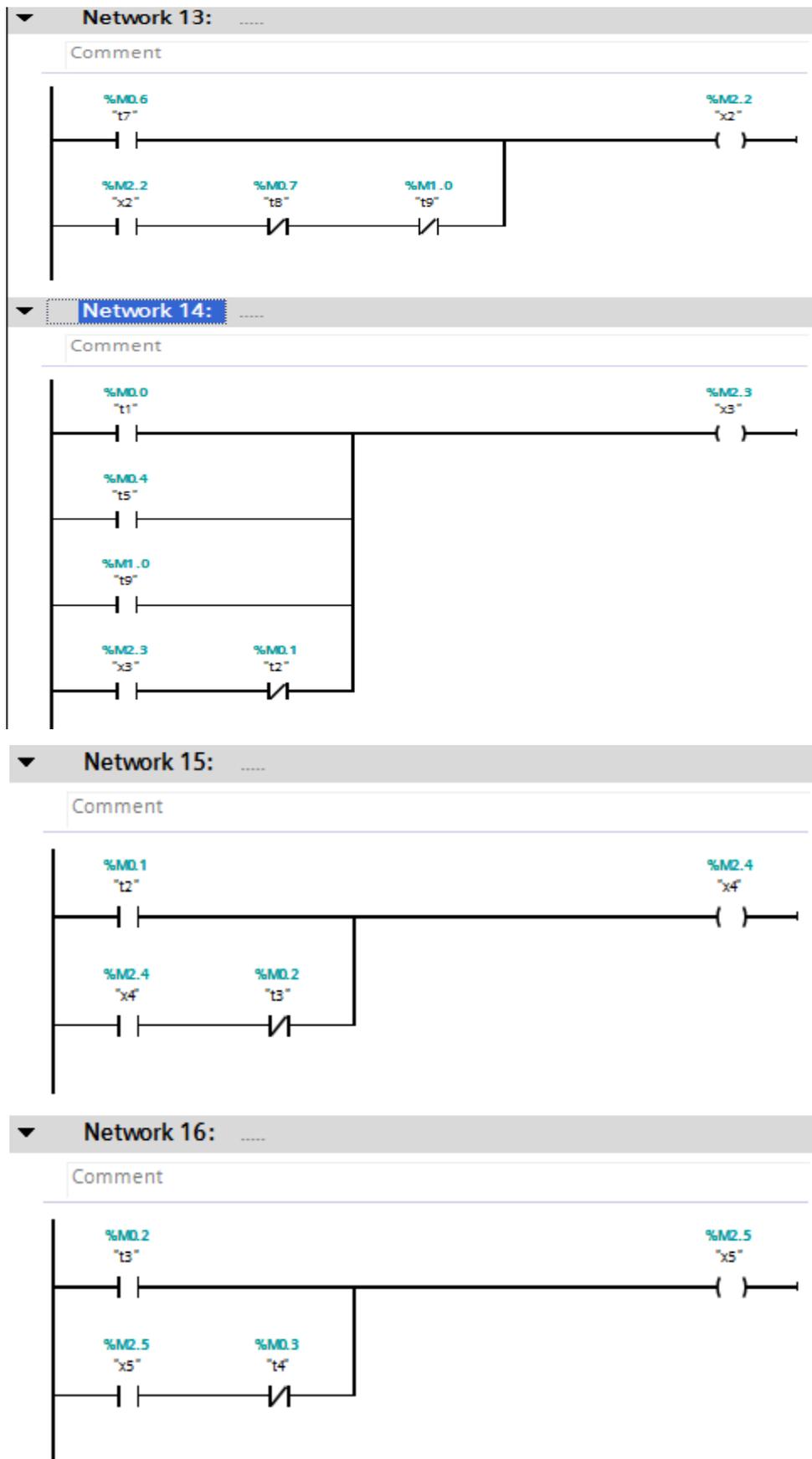


Figure 3.18. Programmation de l'API

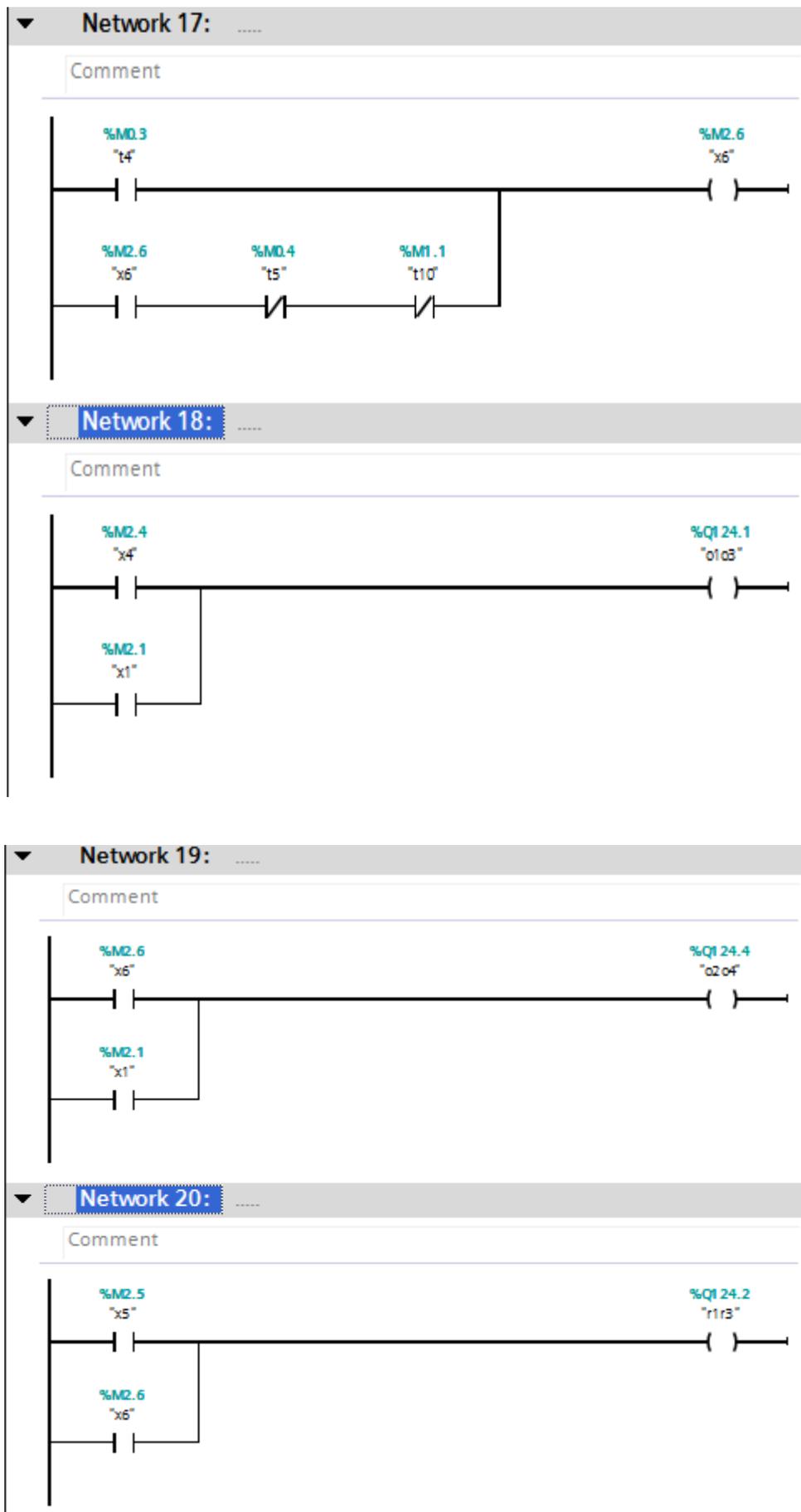


Figure 3.18. Programmation de l'API

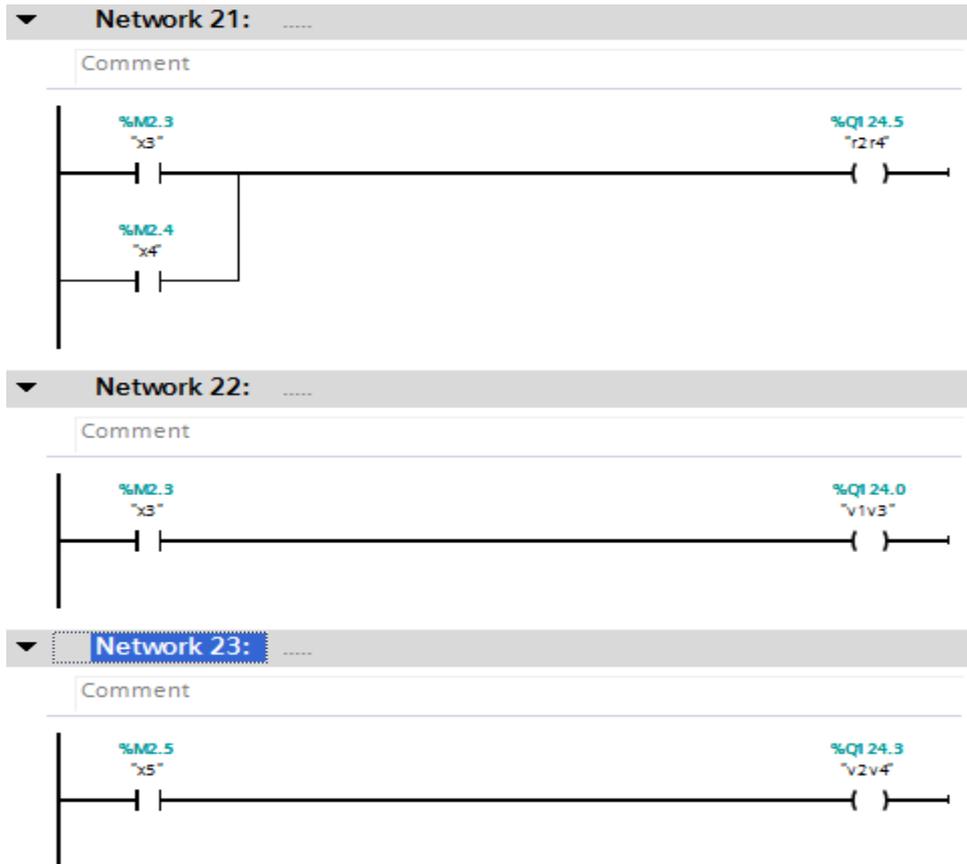


Figure 3.18. Programmation de l'API

a. variateur de vitesse d'un moteur

➤ structure du programme

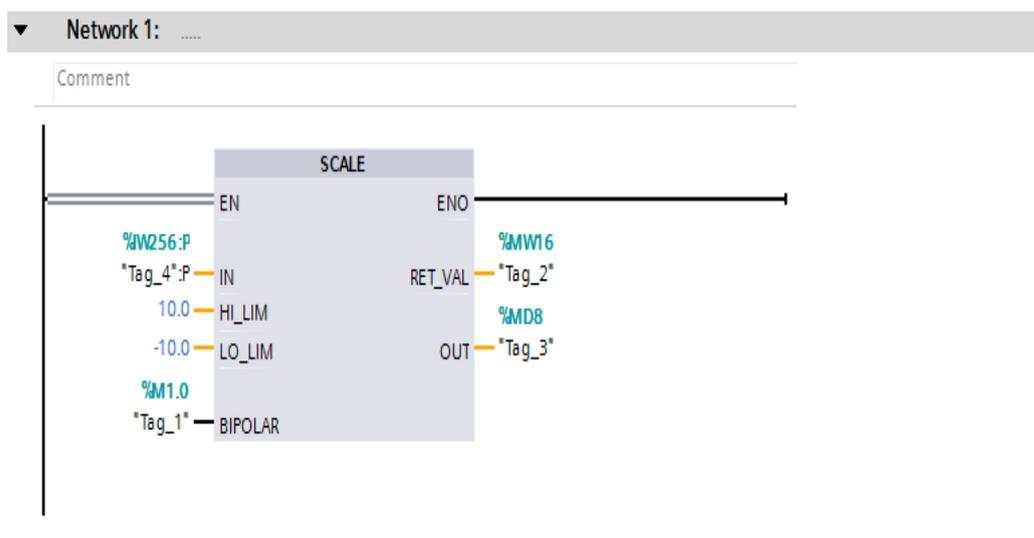
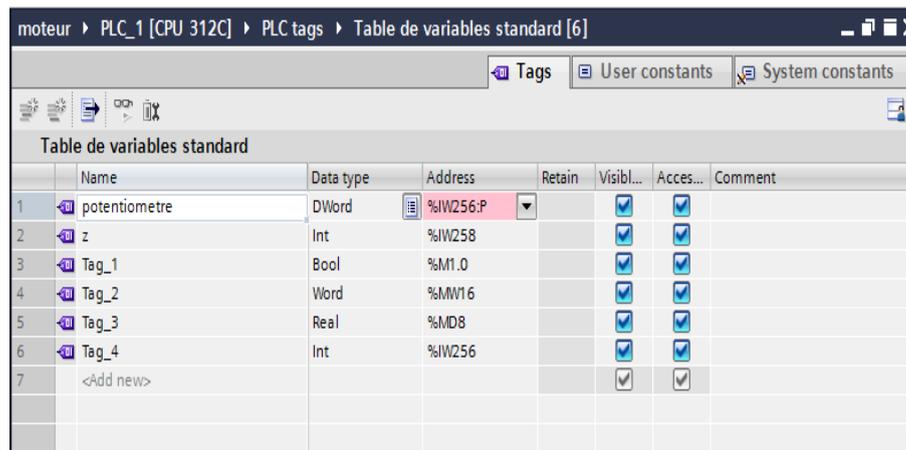


Figure 3.19. La valeur renvoyée par le potentiomètre



**Figure 3.20.** Les variables d'un bloc scale

➤ **paramètres du bloc scale :**

Le tableau suivant montre les paramètres de l'instruction " **UNSCALE** "

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
EN	Input	BOOL	I ,Q,M,D,L	Entrée de variation
END	Output	BOOL	I,Q,M,D,L	Sortie de validation
IN	Input	INT	I,Q,M,D,L,P ou Constante	Valeur d'entrée à mettre a l'échelle
HI_LIM	Input	REAL	IQMDLP ou Constante	Valeur limite supérieure
LO_LIM	Input	REAL	I,Q,M,D,L,P ou Constant	Valeur limite inférieure
BIPOLAR	Input	BOOL	IQMDLP ou Constante	Indique si la valeur du paramètre IN est interprète comme bipolaire ou unipolaire ce paramètre peut prendre les valeurs suivantes :

				1 : bipolaire 0 : unipolaire
OUT	Output	REAL	I,Q,M,D,L,P	Résultat de l'instruction
RET_VAL	Output	WORD	I,Q,M,D,L,P	Information d'erreur

**Tableau 3.1.** Paramètres du bloc scale

**Paramètre de RET\_VAL**

La signification des valeurs du paramètre RET\_VAL :

Le code d'erreur est en hexadécimal (W#16#...).

Le nombre 0000 indique qu'il n'y a aucune erreur, et le nombre 0008 indique que la valeur IN est supérieure à la valeur de la constante K2 ou inférieure à la valeur de la constante K1.

➤ **Le fonctionnement du bloc scale :**

L'instruction scale ou mise à l'échelle permet de convertir le nombre entier du paramètre NI en un nombre à virgule flottante mis à l'échelle en unités physiques comprises entre une valeur limite inférieure et une valeur limite supérieure. On définit la valeur limite inférieure et supérieure de la plage de valeurs sur laquelle la valeur d'entrée est mise à l'échelle par le biais des paramètres LO\_LIM et HI\_LIM. Le résultat de l'instruction est inscrit dans le paramètre OUT.

**L'instruction mise à l'échelle utilise l'équation suivante :**

$$OUT = [((FLOAT(IN) - K1) / (K2 - K1)) * (HI\_LIM - LO\_LIM)] + LO\_LIM$$

Les valeurs des constantes "K1" et "K2" sont déterminées par l'état logique du paramètre BIPOLAR.

Le paramètre BIPOLAR peut prendre les états suivants :

- Etat logique "1" : on suppose que la valeur dans le paramètre NI est bipolaire et se situe dans une plage de valeurs allant de -27648 à +27648. Dans ce cas, la constante "K1" a la valeur "-27648,0" et la constante "K2" la valeur "+27648,0"
- Etat logique "0" : on suppose que la valeur dans le paramètre NI est unipolaire et se situe dans une plage de valeurs allant de 0 à 27648. Dans ce cas, la constante "K1" a la valeur "0,0" et la constante "K2" la valeur "+27648,0".

Lorsque la valeur au paramètre Ni est supérieure à celle de la "K2", le résultat de l'instruction prend la valeur de la limite supérieure (HI\_LIM) et une erreur est signalée.

Lorsque la valeur au paramètre NI est supérieure a celle de la constante "K1", le résultante l'instruction prend la valeur de la limite inférieure (LO\_LIM) et une erreur est signalée.

Lorsque la valeur limite inferieur indiquée est supérieure a la valeur limite supérieure (LO\_LIM>HI\_LIM), le résultat est mis a l'échelle de manière inversement proportionnelle a la valeur d'entrée.

On a utilisé le bloc scale pour convertir la tension du potentiomètre en nombre real (binaire de 12 bits) [31].

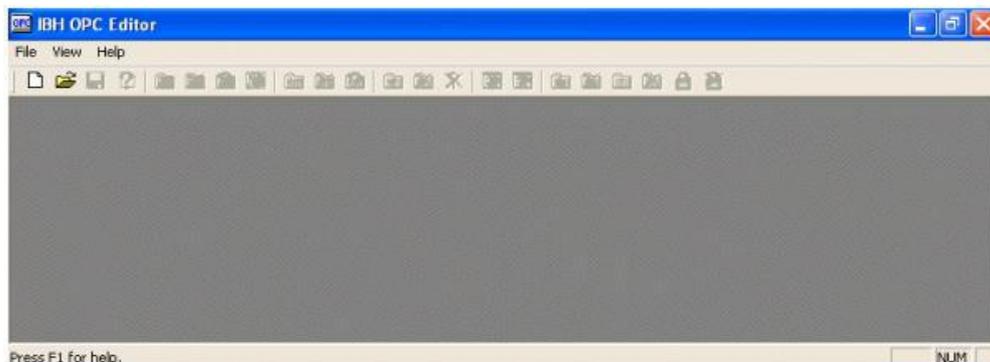
### **3.3.2 Transmission des variables**

Après la configuration et le chargement dans l'automate (annexe A), nous passons à l'étape suivante.

#### **a. Serveur OPC :**

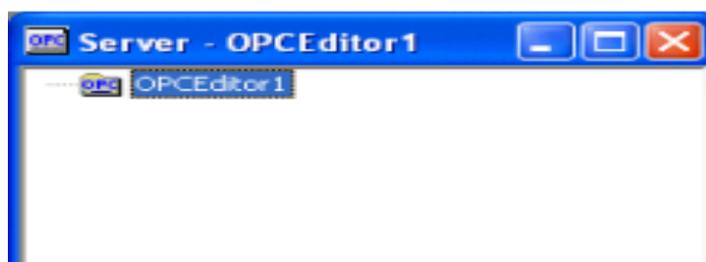
La communication entre S7-300 et Le serveur OPC se fait selon les étapes suivantes :

- **Créer un projet** : L'éditeur OPC nous permettra de créer un nouveau projet ou d'en ouvrir un existant. Pour créer un nouveau projet, nous cliquons sur Fichier→Nouveau (voire le Figure 3.21)



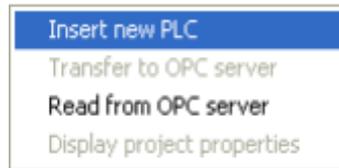
**Figure 3. 21.** IBH OPC Editeur

- Cela ouvre l'arborescence du projet de l'éditeur OPC.



**Figure 3. 22.** Editeur OPC

- Avec un clic droit sur le "OPCEditor1", nous apparaissent le menu suivant et nous choisissons "Insérer un nouveau automate".



**Figure 3. 23.** Nouveau PLC

Ceci expose les propriétés de l'API.

- Ici, nous donnerons un nom significatif à notre automate dans le projet et définissons le protocole sur lequel communiquer (S7 Simatic Net). Nous utilisons "S7 MPI converter serial / USB" pour notre cas d'un automate s7 312 MPI (Message Passing Interface).



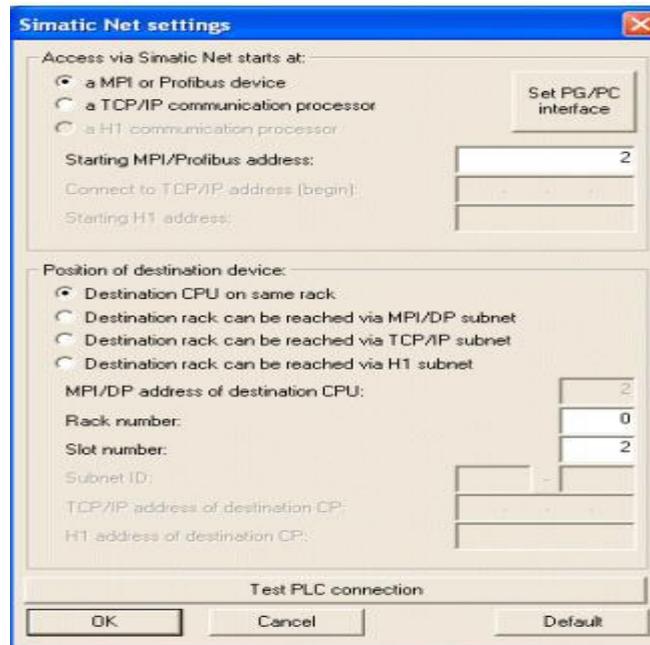
**Figure 3. 24.** Propriétés de PLC

- Sur les propriétés de PLC Nous Cliquons avec le bouton droit sur S7 Simatic Net et nous sélectionnons les paramètres de connexion. Nous trouvons la boîte de dialogue Simatic Net Settings, comme indiqué dans la Figure ci-dessous.



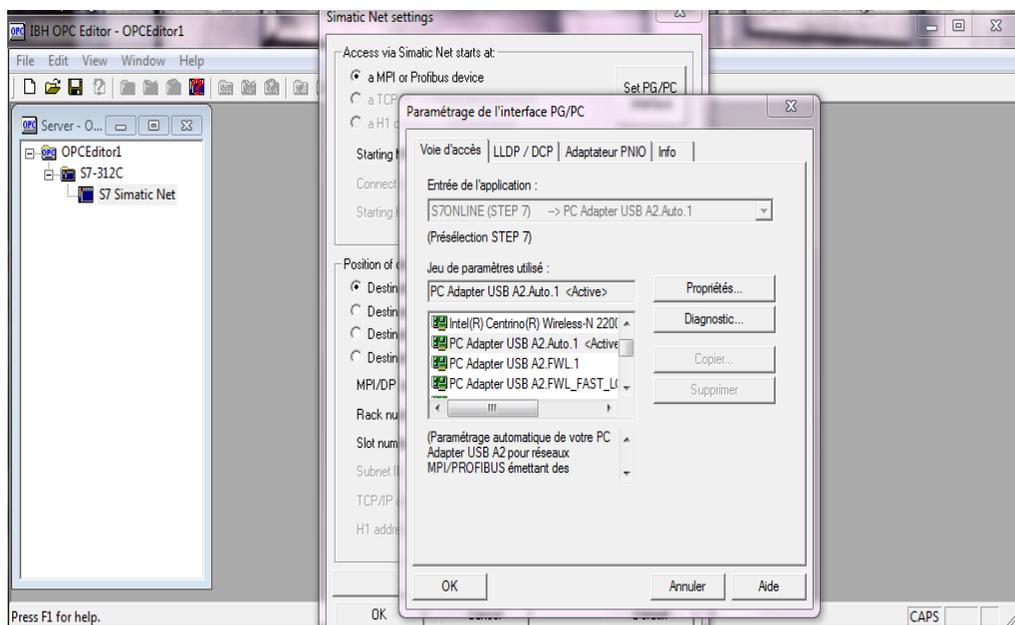
**Figure 3.25.**OPC Editeur Simatic S7

- Lorsque la boîte de dialogue Simatic Net Settings s'ouvre, nous vérifions les paramètres par rapport à ceux de notre réseau et de l'appareil que nous essayons de connecter. Puis nous cliquons sur "Set PG / PC Interface".



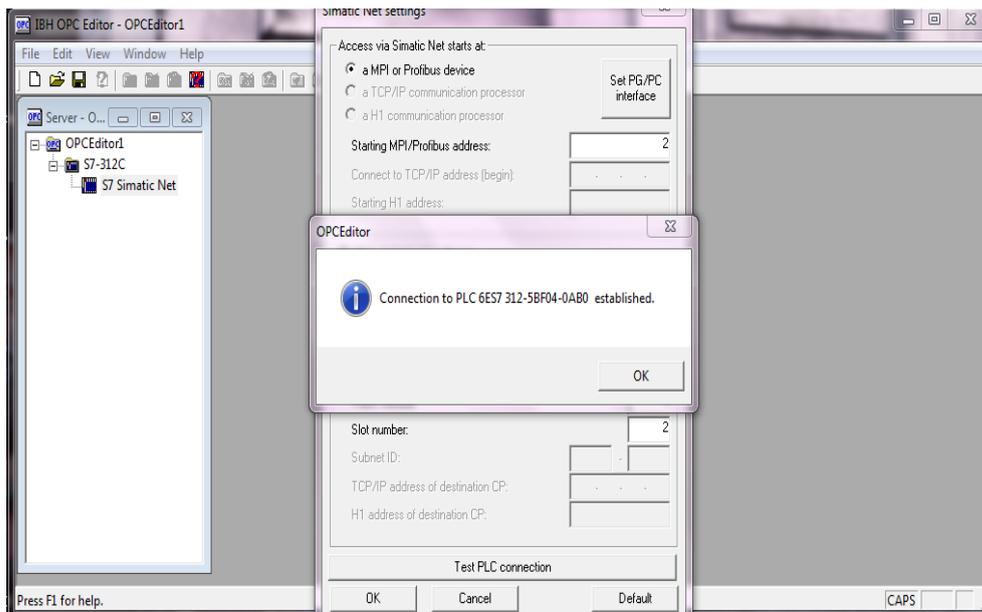
**Figure 3. 26.** Simatic Net Settings

- Après avoir cliqué sur "Set PG / PC Interface" dans le dialogue Simatic Net Settings, nous verrons le dialogue d'interface PG / PC. Ici nous allons configurer notre interface d'adaptateur pour utiliser USB, plutôt qu'un port COM standard.



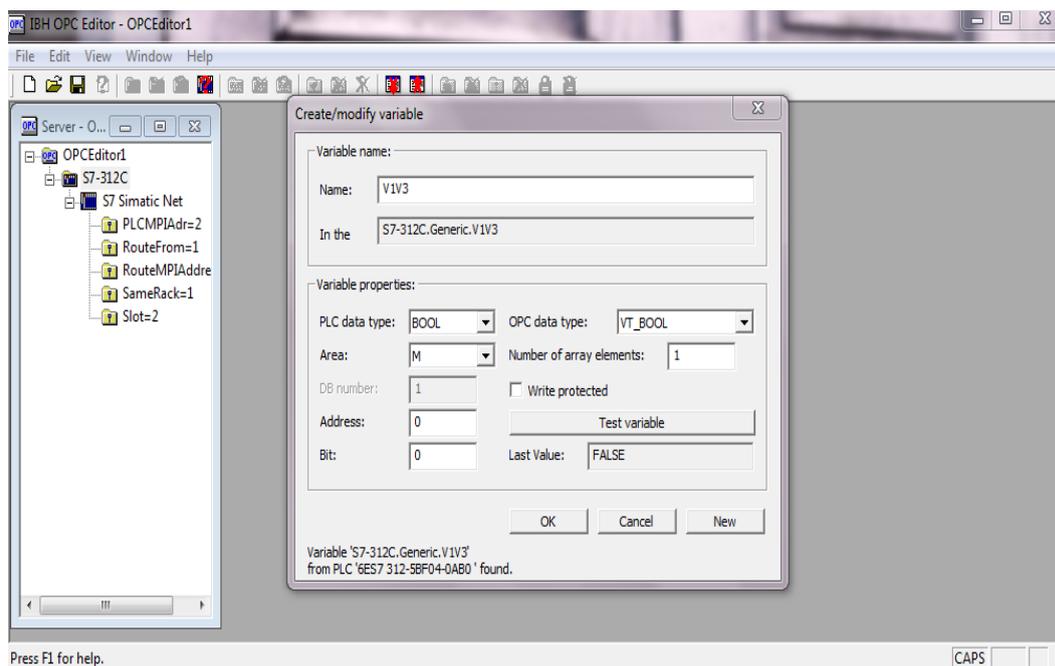
**Figure 3. 27.** Interface PG/PC

- Nous mettons en évidence PC adapter USB A2 AUTO.1 (MPI) pour configurer l'interface à utiliser avec le câble Siemens USB -MPI.



**Figure 3. 28.** Paramètre de PG/PC

- Nous devons sélectionner les variables à surveiller, avec un clic droit sur PLC 1 comme le montre la figure 3. 29, nous sélectionnons Select Variables de TIA portal STEP7 qui a été enregistré avant.



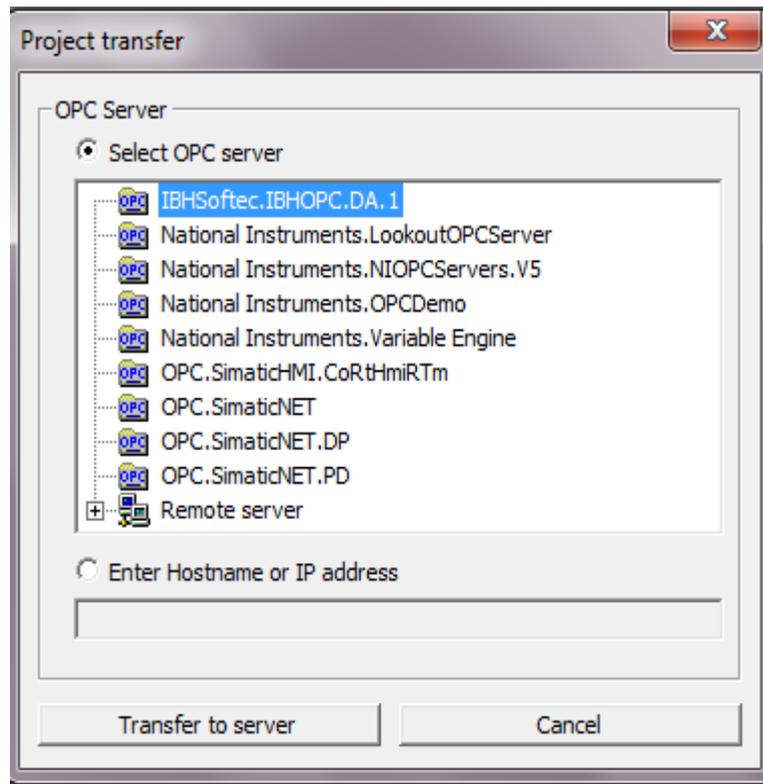
**Figure 3. 29.** Configuration des variables

Le choix du nom est le même qui existe dans le programme de TIA protale.

- Nous choisissons les types de variable et aussi les adresses.

**Remarque : les adresses doit être les mêmes de notre programme de TIA protal.**

- Nous transférons tous les variables qui nous avons configuré voire le (Figure 3. 30)



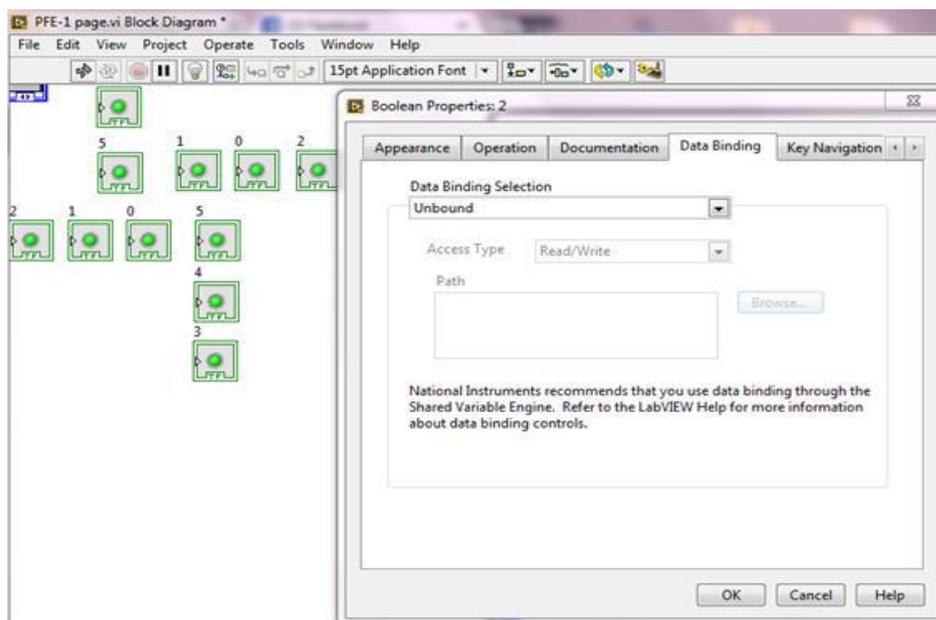
**Figure 3. 30.** Transfer des variables au serveur OPC IBH

Nous choisissons le serveur OPC qui nous avons utilisé « IBHSoftec.IBHOPC.DA.1 »

**a. LabVIEW**

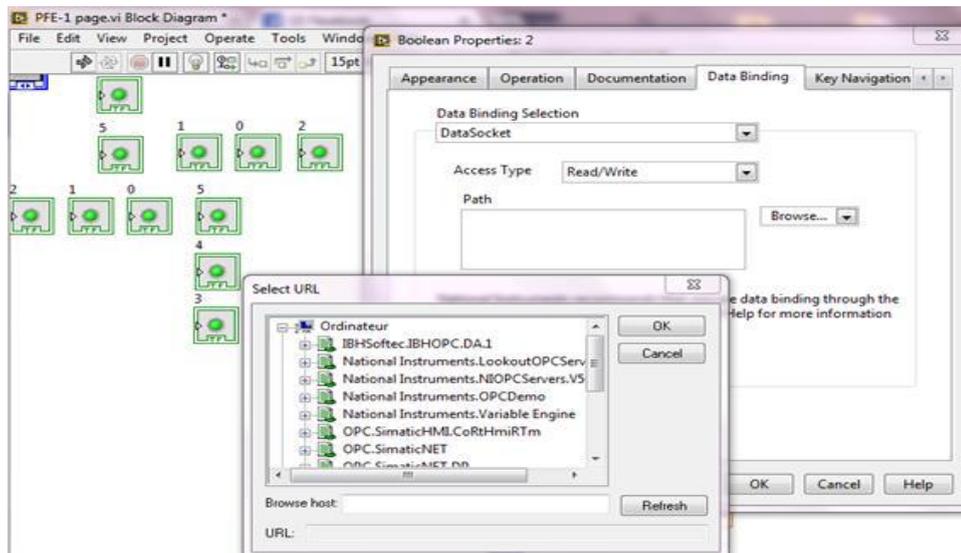
Sur LabVIEW nous allons appeler les variables de notre programme selon les étapes suivantes :

- Sur le block diagram, nous cliquons à droite sur le booléen pour ouvrir la fenêtre de propriétés



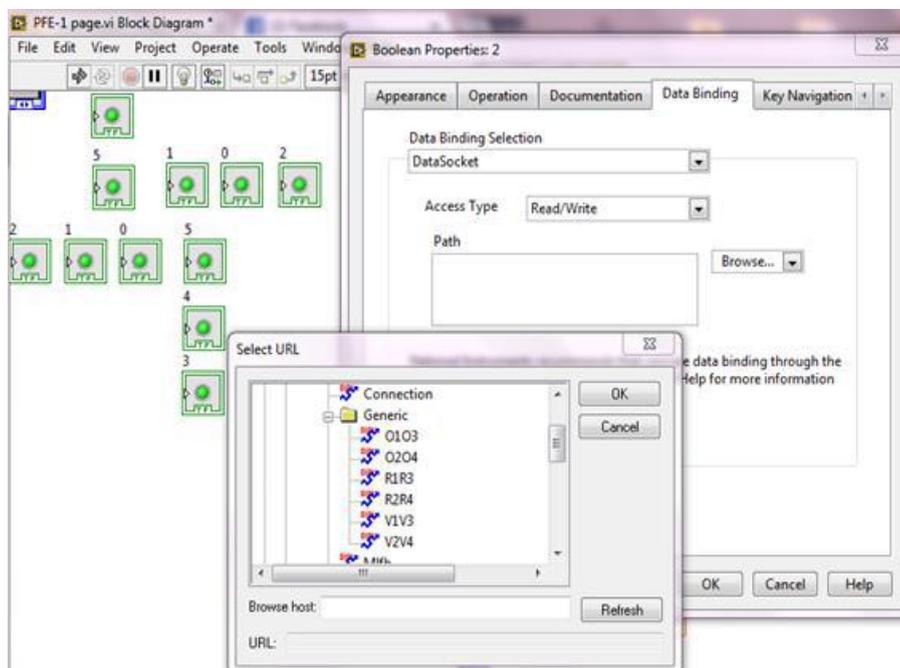
**Figure 3. 31.** Configuration sur LabVIEW

- Nous cliquons sur « **DATA Binding** ». Et nous choisissons **DAT SOKET**  
➔ **Read/Write** pour le type (voire le Figure 3. 32)

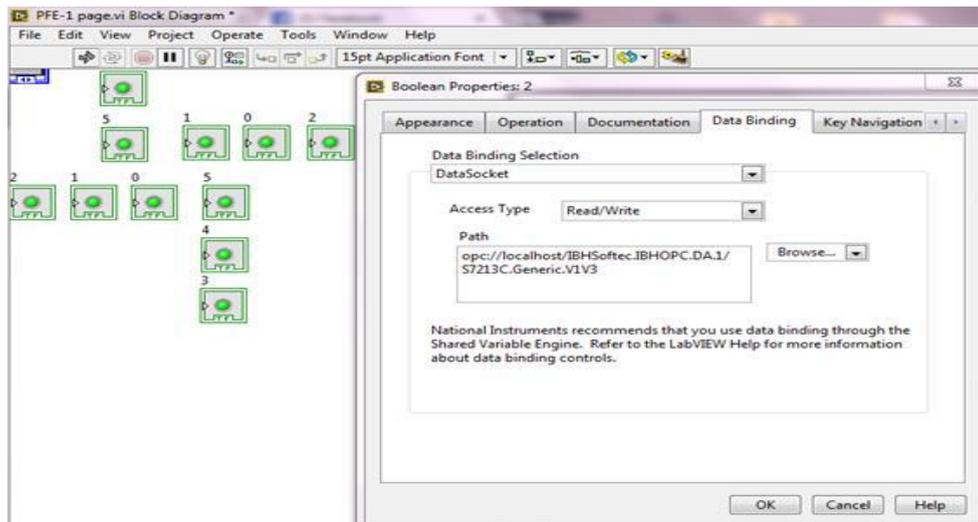


**Figure 3. 32.** Paramètres de variables

- A partir de la liste URL nous choisissons **IBHSoftec IBHOPC.DA .1 ➔S7213c**  
➔ **Generic** (voire le Figure 3. 33)



**Figure 3. 33.** les Variables



**Figure 3. 34.** Fin de configuration

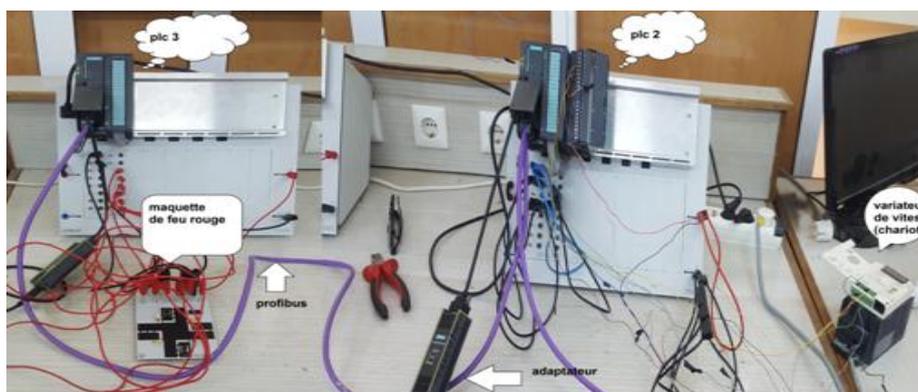
➤ chaque booléen prendre sa propre variable.

La communication est réussie entre le TIA portal et LabVIEW.

**Rq : la même chose pour la transmission des variables pour le chariot**

Nous avons réalisé une supervision par LabVIEW afin de visualiser et contrôler le système de feu de circulation et le chariot que nous avons programmé au niveau de l'automate, comme nous pouvons également commander les deux systèmes via la face avant de LabVIEW.

Dans ce cas nous pouvons visualisé, contrôlé et commandé à partir de LabVIEW comme se présente dans la figure ci-dessous :



**Figure 3. 35.** Interconnexion entre 2 PLC

### 3.4 Conclusion

Ce chapitre a été consacré à l'association de tous les méthodes qui nous avons utilisé lors de la configuration et la programmation au niveau de LabVIEW et le TIA Portal pour faire la supervision et le contrôle en temps réel.

## Conclusion général

---

Nous avons concevoir dans notre projet un exemple de système SCADA industriel qui contient 3 automate programmable de type siemens S7-312c, chaque automate programmable il a son propre programme, à titre d'exemple nous avons programmes le PLC1 pour varié la vitesse d'un moteur asynchrone à travers le potentiomètre et aussi nous avons programmes le 2eme PLC pour visualise le feu de circulation et la 3eme PLC pour commander et contrôlé le mouvement d'un chariot.

Pour communique entre ces 3 automate programmable il Faut d'un réseau d'interface, dans notre cas nous avons utilisé le réseau MPI pour communique entre ces automate programmables.

Nous avons utilisé une carte d'acquisition de type NI-USB 6001 pour contrôler certain périphériques électronique. Dans notre cas nous avons utilisé le thermocouple pour mesure la température et le capteur de pression pour mesuré la pression et aussi le photocellule pour faire l'arrêt de moteur en mode de sécurité.

Nous avons utilisé logiciel labVIEW comme d'interface de supervision.

Nous avons utilisé le OPC server pour adapter la communication entre TIA Portal et LabVIEW.

## Annexe A : le TIA Portail

### 1. Vue du portail et vue du projet

Lorsque l'on lance TIA Portal, l'environnement de travail se décompose en deux types de vue :

#### 1.1 La vue du portail

Elle est axée sur les tâches à exécuter et sa prise en main est très rapide.

Chaque portail permet de traiter une catégorie de tâche (actions). La fenêtre affiche la liste des actions pouvant être réalisées pour la tâche sélectionnée.

#### 1.2 La vue du projet

Elle comporte une arborescence avec les différents éléments du projet. Les éditeurs requis s'ouvrent en fonction des tâches à réaliser. Données, paramètres et éditeurs peuvent être visualisés dans une seule et même vue (voir la figure 1).

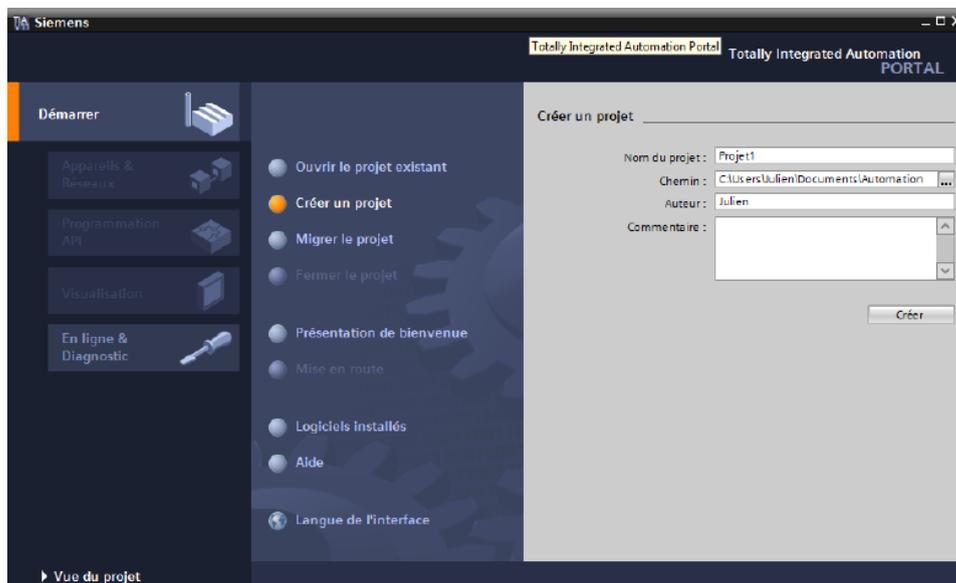


Figure 1. Vue du Portail

L'élément « Projet » contient l'ensemble des éléments et des données nécessaires pour mettre en œuvre la solution d'automatisation souhaitée (voir la figure 2).

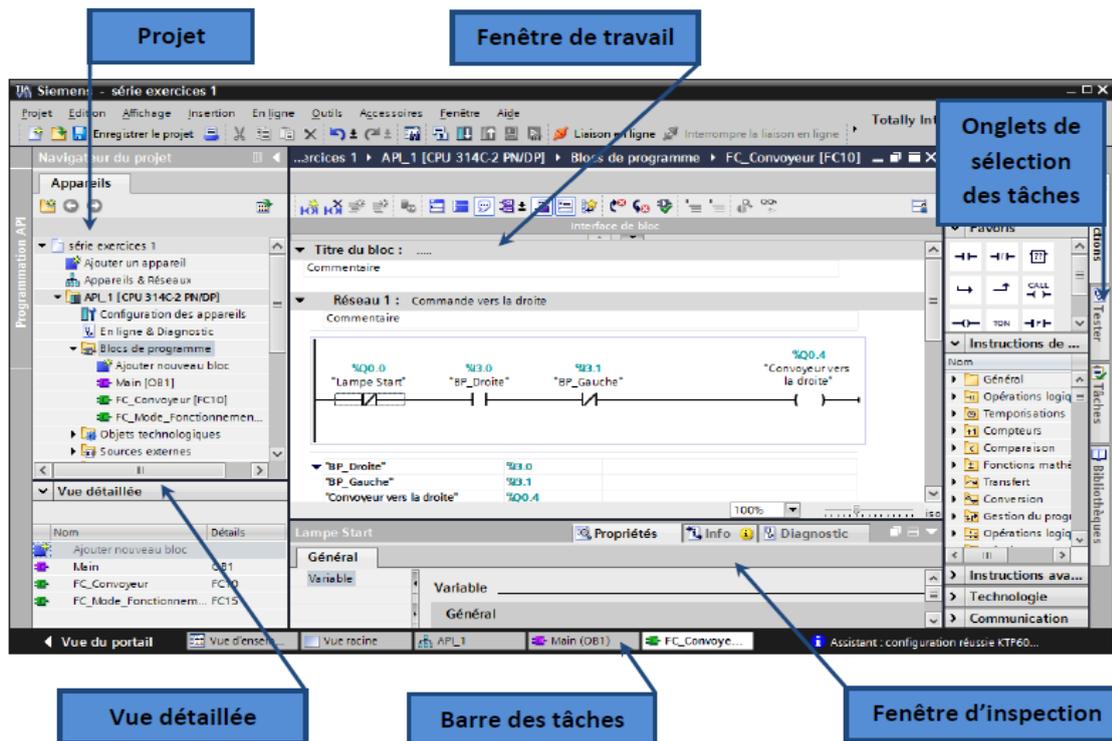


Figure 2. Vue du Portail

- La fenêtre de travail permet de visualiser les objets sélectionnés dans le projet pour être traités. Il peut s'agir des composants matériels, des blocs de programme, des tables des variables, des HMI,...
- La fenêtre d'inspection permet de visualiser des informations complémentaires sur un objet sélectionné ou sur les actions en cours d'exécution (propriété du matériel sélectionné, messages d'erreurs lors de la compilation des blocs de programme,...).
- Les onglets de sélection de tâches ont un contenu qui varie en fonction de l'objet sélectionné (configuration matérielle → bibliothèques des composants, bloc de programme → instructions de programmation).

Cet environnement de travail contient énormément de données. Il est possible de masquer ou réduire certaines de ces fenêtres lorsque l'on ne les utilise pas.

Il est également possible de redimensionner, réorganiser, désancrer les différentes fenêtres.

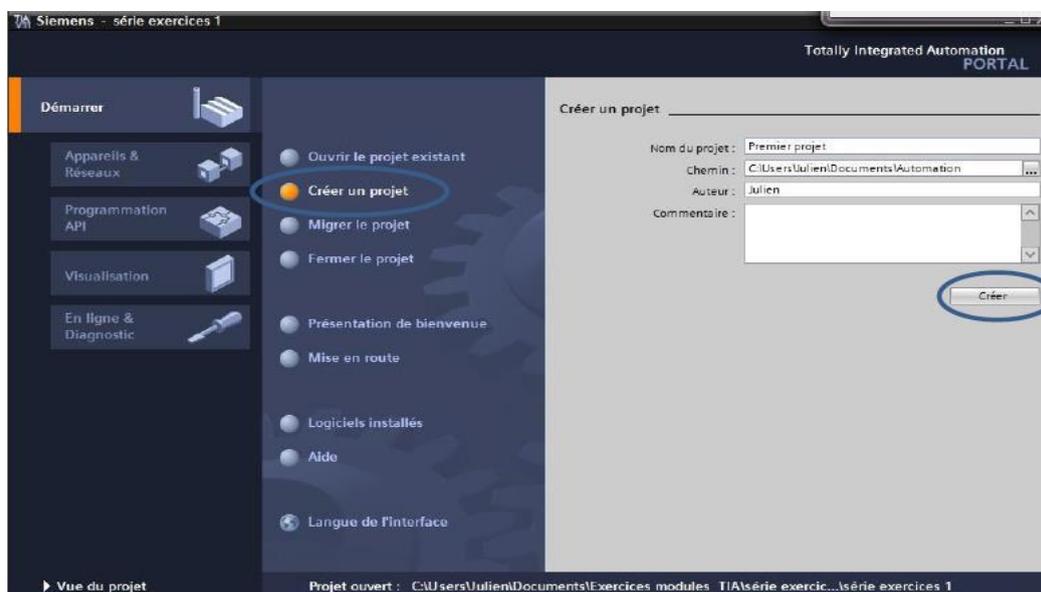
## 2. Création d'un projet et configuration d'une station de travail

### 2.1 Création d'un projet

Pour créer un projet dans la vue du portail, il faut sélectionner l'action « Créer un projet ».

On peut donner un nom au projet, choisir un chemin où il sera enregistré, indiquer un commentaire ou encore définir l'auteur du projet.

Une fois que ces informations sont entrées, il suffit de cliquer sur le bouton « créer » (voir la figure 3).



**Figure 3** .Création d'un projet

### 2.2 Configuration et paramétrage du matériel

Une fois votre projet créé, on peut configurer la station de travail.

La première étape consiste à définir le matériel existant. Pour cela, on peut passer par la vue du projet et cliquer sur « ajouter un appareil » dans le navigateur du projet.

La liste des éléments que l'on peut ajouter apparaît (API, HMI, système PC). On commencera par faire le choix de notre CPU pour ensuite venir ajouter les modules complémentaires (alimentation, E/S TOR ou analogiques, module de communication AS-i,...). (Voir la figure 4).

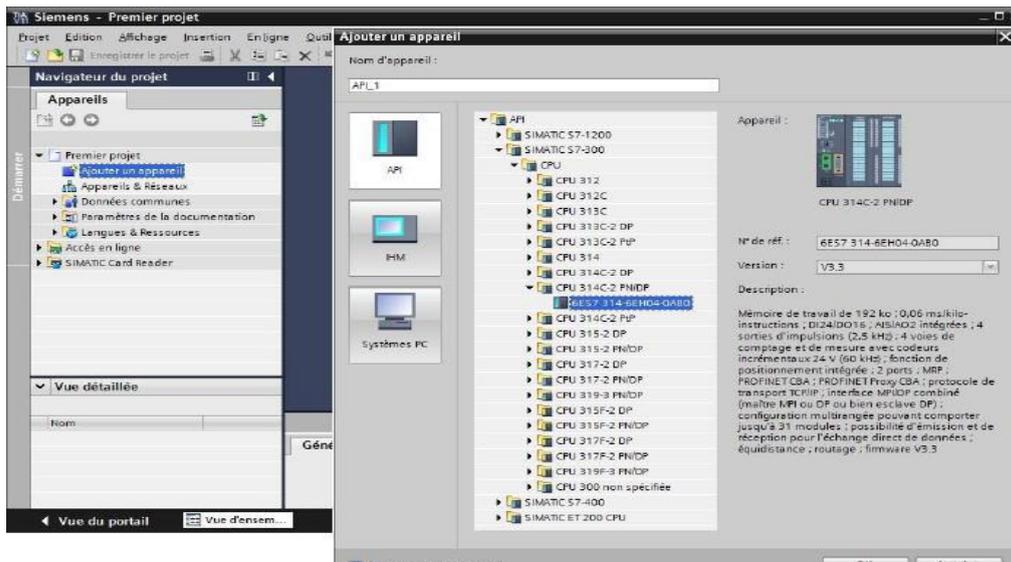


Figure 4. Configuration et paramétrage du matériel

Les modules complémentaires de l'API peuvent être ajoutés en utilisant le **catalogue**. Par exemple pour ajouter un module analogique il faut repasser par la commande « **ajouter un appareil** » dans le navigateur du projet.

Lorsque l'on sélectionne un élément à insérer dans le projet, une description est proposée dans l'onglet information. (Voir la figure 5)

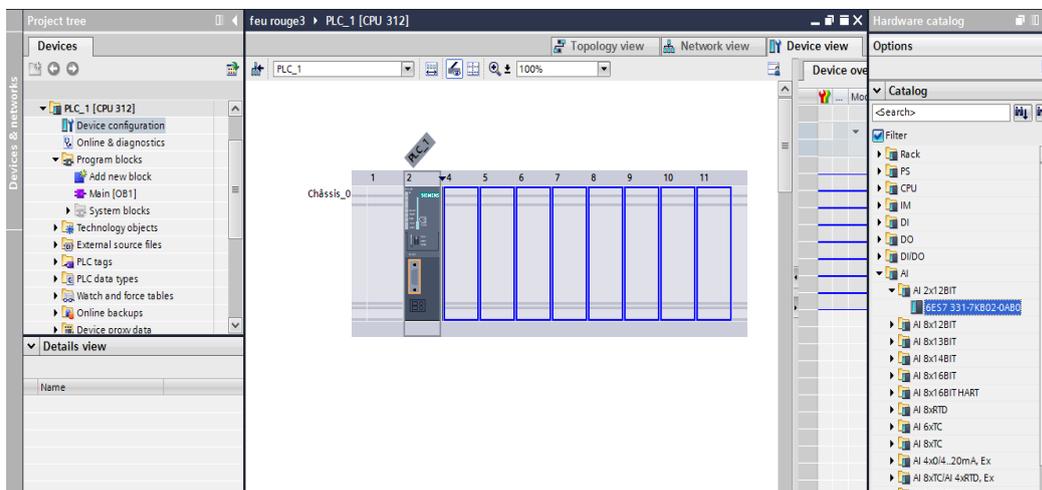


Figure 5. Configuration et paramétrage du matériel

### a. Adressage des E/S

Pour connaître l'adressage des entrées et sorties présentes dans la configuration matériel, il faut aller dans « appareil et réseau » dans le navigateur du projet.

Dans la fenêtre de travail, on doit s'assurer d'être dans l'onglet « Vue des appareils » et de sélectionner l'appareil voulu. (voir la figure 6).

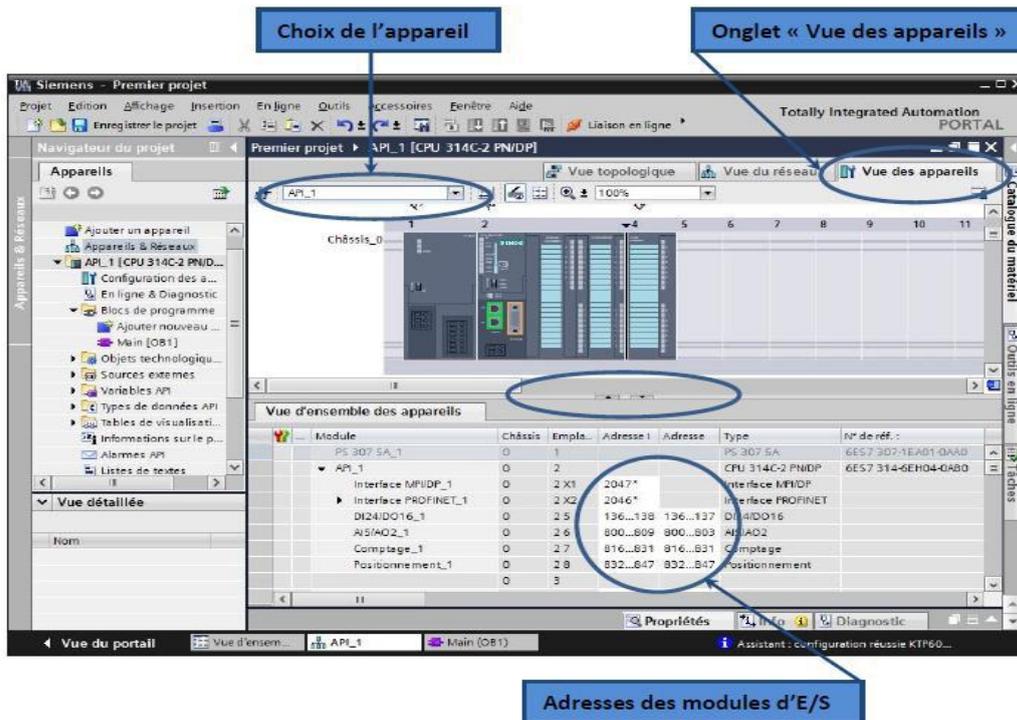


Figure 6. Adressage des E/S

On sélectionne la CPU puis à l'aide des deux petites flèches (voir figure), on fait apparaître l'onglet « Vue d'ensemble des appareils » (n'hésitez pas à masquer certaines fenêtres et à en réorganiser d'autres). Les adresses des entrées et sorties apparaissent. Vous pouvez les modifier en entrant une nouvelle valeur dans la case correspondante.

### b. Mémento de cadence

Une fois la CPU déterminée, on peut définir le memento de cadence. Pour cela, on sélectionne la CPU dans la fenêtre « Vue des appareils » et l'onglet « propriété » dans la fenêtre d'inspection.

Dans le menu « Général », choisir l'option « Memento de cadence », cocher la case « Mémento de cadence » et choisir l'octet du memento de cadence que l'on va utiliser. (Voir la figure 7).

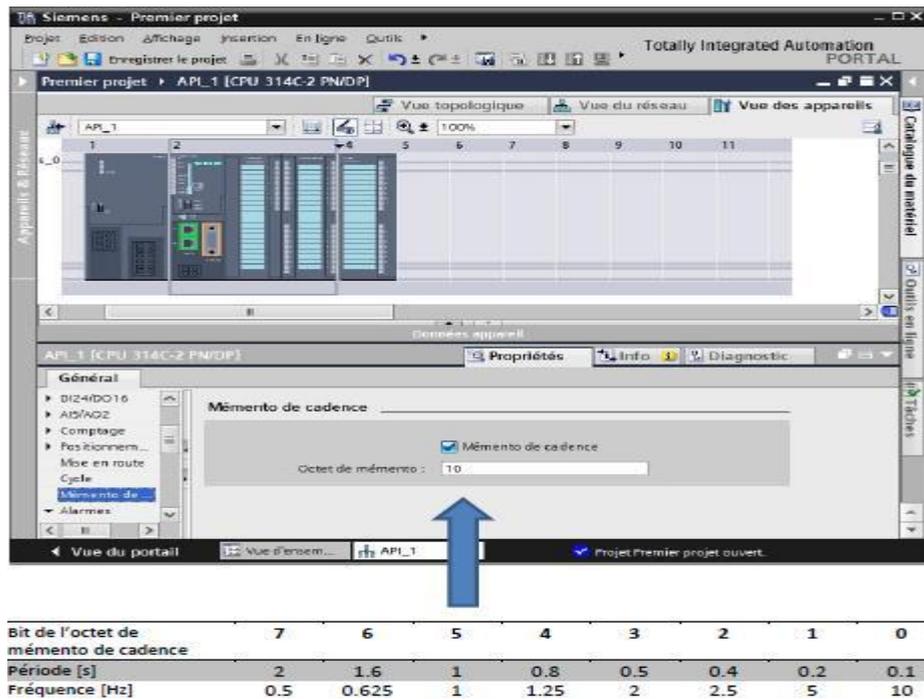


Figure 7. Memento de cadence

### c. Adresse Ethernet de la CPU

Toujours dans les propriétés de la CPU, il est possible de définir son adresse Ethernet. Un double clic sur le connecteur Ethernet de la station fait apparaître la fenêtre d'inspection permettant de définir ses propriétés.

Pour établir une liaison entre la CPU et la console de programmation, il faut affecter aux deux appareils des adresses appartenant au même réseau. On utilisera comme adresse pour l'automate 192.168.2.n° de l'automate. (Voir la figure 8).

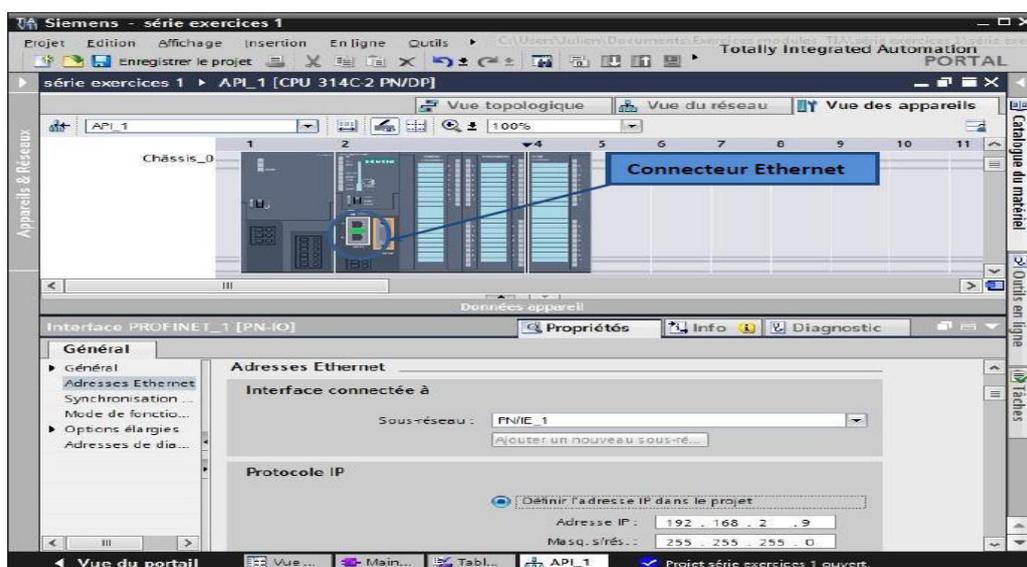


Figure 8. Adresse Ethernet de la CPU

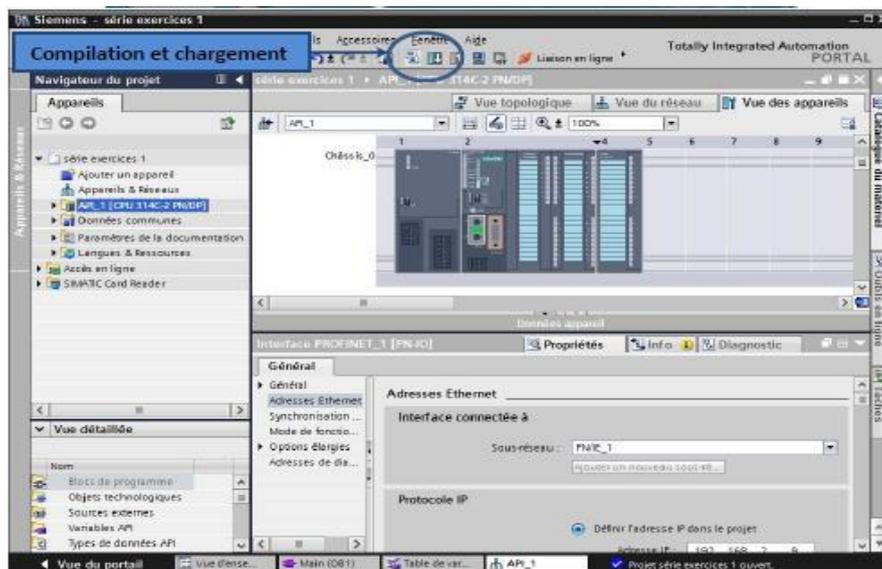
**d. Compilation et chargement de la configuration matérielle**

Une fois la configuration matérielle réalisée, il faut la compiler et la charger dans l'automate.

La compilation se fait à l'aide de l'icône « compiler » de la barre de tâche. On sélectionne l'API dans le projet puis cliquer sur l'icône « compiler ».

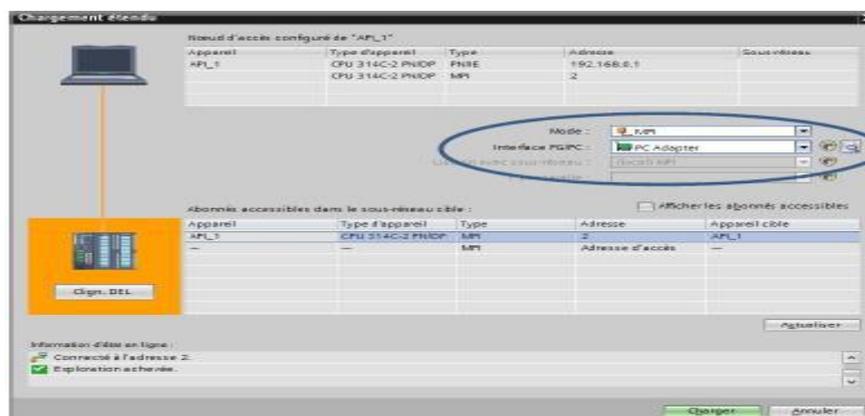
En utilisant cette manière, on effectue une compilation matérielle et logicielle.

Une autre solution pour compiler est de faire un clic droit sur l'API dans la fenêtre du projet et de choisir l'option « Compiler → Configuration matérielle ». (Voir la figure 9).



**Figure 9.** Compilation et chargement de la configuration de la CPU

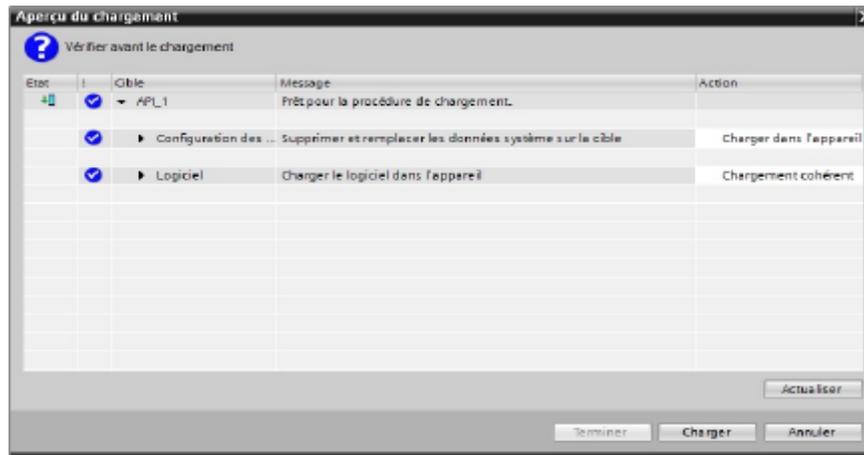
Pour charger la configuration dans l'automate, on effectue un clic sur l'icône « charger dans l'appareil ». La fenêtre ci-dessous s'ouvre et vous devez faire le choix du mode de connexion (PN/IE, Profibus, MPI). Si vous choisissez le mode PN/IE, l'API doit posséder une adresse IP. (Voir la figure 10).



**Figure 10.** Compilation et chargement de la configuration de la CPU

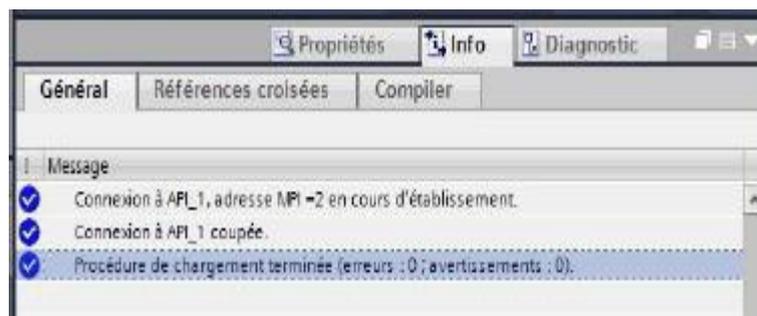
Pour une première connexion ou pour charger l'adresse IP désirée dans la CPU, il est plus facile de choisir le mode de connexion MPI et de relier le PC à la CPU via le « PC Adapter ».

Si le programme trouve un appareil, il figure dans la liste en bas de la fenêtre. La touche « Clign. DEL » permet de faire clignoter une LED sur la face avant de l'appareil afin de s'assurer que l'on est connecté à l'appareil désiré. (voir la figure 11).



**Figure 11.** Compilation et chargement de la configuration de la CPU

Une fois la configuration terminée, on peut charger le tout dans l'appareil. Des avertissements / confirmations peuvent être demandés lors de cette opération. Si des erreurs sont détectées, elles seront visibles via cette fenêtre. Le programme ne pourra pas être chargé tant que les erreurs persistent. (Voir la figure 12).



**Figure 12.** Compilation et chargement de la configuration de la CPU

Le programme de l'automate programmable est créé sur le PC avec le logiciel TIA portal est temporairement enregistré. Une fois que nous avons relié le PC avec l'interface MPI de l'automate, le programme peut alors être chargé à l'aide de la fonction de chargement dans la mémoire de l'automate programmable.

## Annexe B : LabVIEW

### 1. Configuration

#### 1.1 Sorties digitales

- ❖ Avec DAQ assistant ouvert, nous avons sélectionné **Generate signals** → **digital output** → **line output**

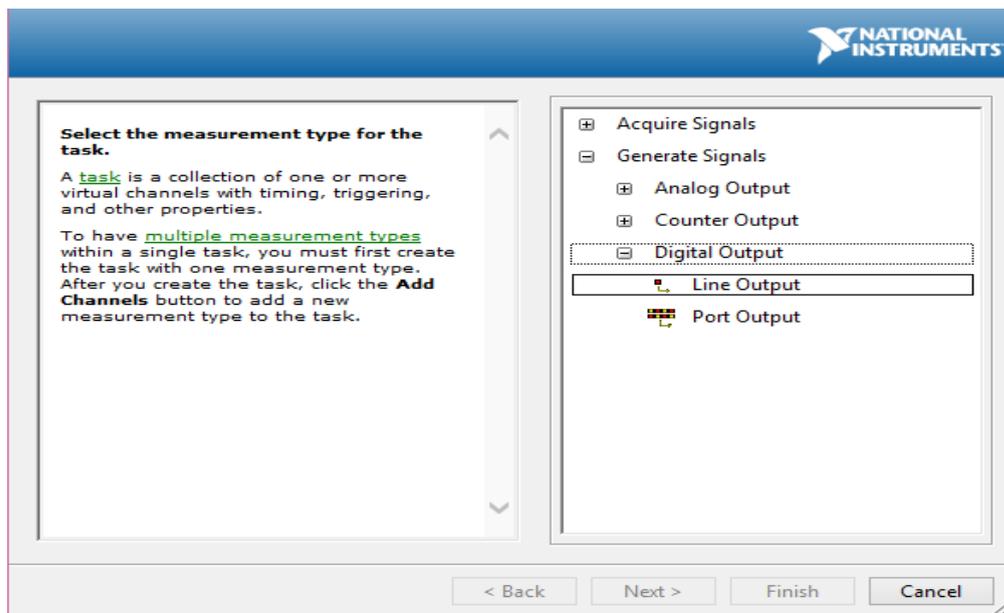


Figure 13. Création d'un canal pour DO0

- ❖ Nous avons sélectionné port1/line0 que nous avons l'intention de programmer un bouton de démarrage.

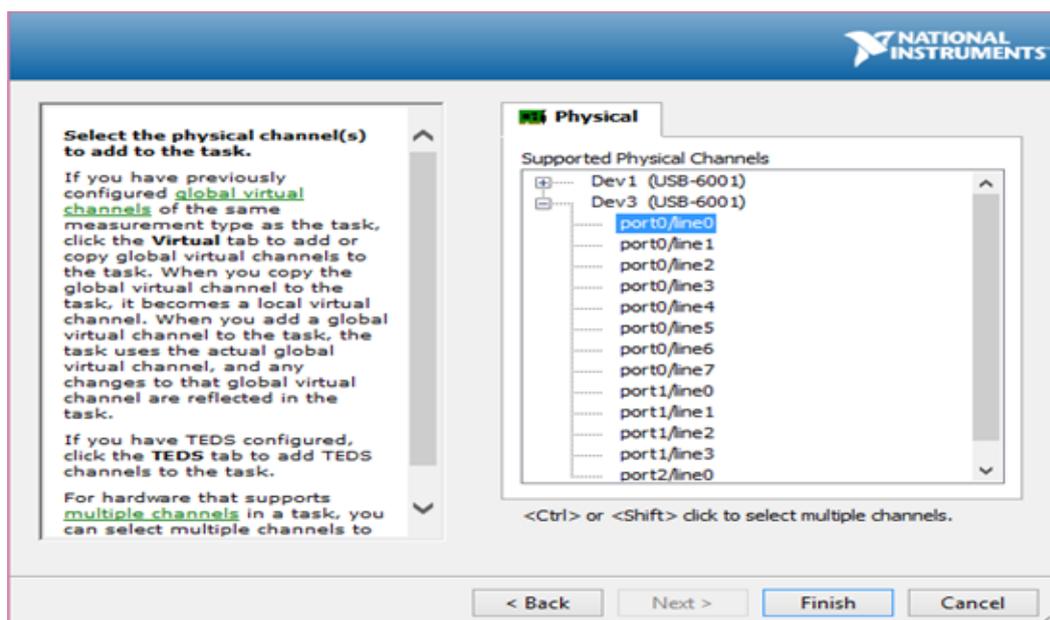
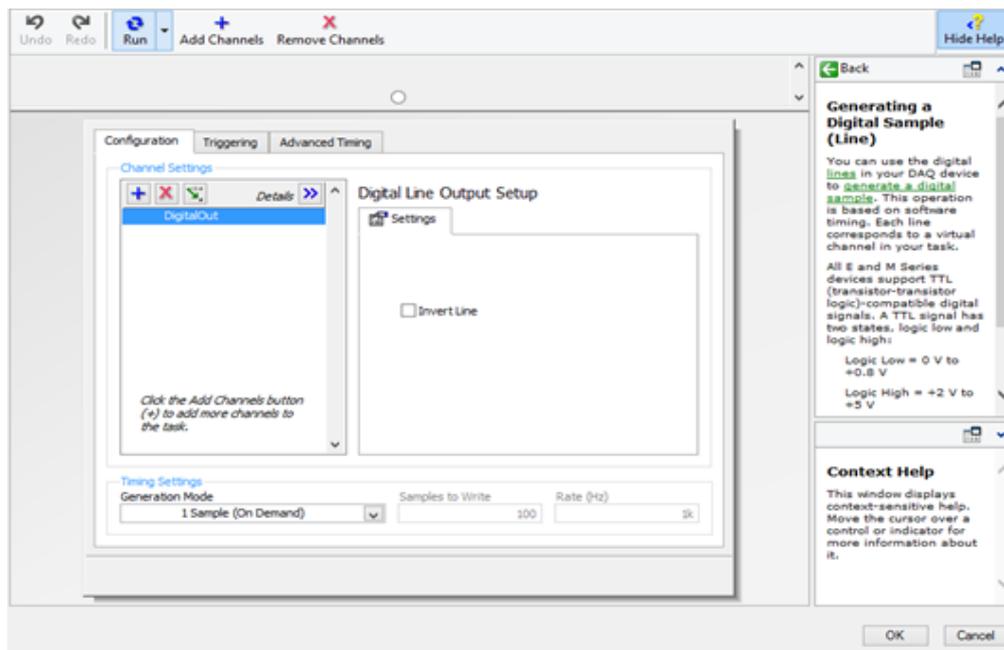


Figure 14. Canal physique DO0



**Figure 15.** Configuration d'un canal de démarrage

# Bibliographie

---

- [1] Site : [www.technologiepro.com](http://www.technologiepro.com),cours-automate-programmable-industriel, consulté le: 18-11-2017.
- [2] site : [www.automation-sense.com](http://www.automation-sense.com), consulté le 18/12/2015.
- [3] site : [www.acquisys.fr/fonction/acquisition-de-donnees](http://www.acquisys.fr/fonction/acquisition-de-donnees).
- [4] le titre du document : LabVIEW Programmation et application. Auteurs : Francis Cottet, Michel Pinard, Luc Desruelle. Créé le 20 mai 2015, consulté le 31/05/2018.
- [5] Site : [www.actors-solutions.com](http://www.actors-solutions.com) année de création 2014, date de consultation 25/05/2018.
- [6] site : [www.teamnet.ro](http://www.teamnet.ro) année de création 2001-2018 TEAMNET INTERNATIONAL S.A., date de consultation 25/05/2018.
- [7] mémoire de master : l'auteur « BOULALI Abdelmonim & BAAZIZ Abderrahim » , titre : Réalisation d'un système de comptage à l'usine FCI : Partie Communication OURAGLA , Filière : Génie Electrique Spécialité : Instrumentation , année : 2015 /2016.
- [8] Les Éditions Techniques de l'Ingénieur ou Éditions T.I , Fondées en 1946 par « Maurice Postel » sur le site : [www.techniques-ingenieur.fr](http://www.techniques-ingenieur.fr) , date de consultation 25 / 05/2018.
- [9] site : [www.semanticscholar.org](http://www.semanticscholar.org) , Créateur : Allen Institute for Artificial Intelligence Date de lancement : novembre 2015, date de consultation : 25 /05/2018.
- [10] Le Laboratoire d'Informatique Signal et Image de la Côte d'Opale , créé le 1er janvier 2010 de l'Université du Littoral Côte d'Opale sur le site : <http://www-lisic.univ-littoral.fr> ; Journée d'études SEE – Lille 2010 l'auteur « Jérôme Hostein » Training and Maintenance Manager – SAS Montpellier Expert DCS date de consultation : 30 /05/2018.
- [11] entreprise américain crée en 1976 par « James Truchard, Bill Nowlin, Jeff Kodosky » dans le site : <http://www.ni.com> crée en 2018 consultation en 30/05/2018.
- [12] Article de journal : le FEGARO, crée par Entreprises Le 09/08/2007 « Siemens va créer 10 000 emplois en 2007 » consulté le 30/05/2018.

- [13] site : [www.industry.siemens.com](http://www.industry.siemens.com) créée par siemens 1996 – 2018, consulté le 31/05/2018
- [14] brochure de central de technologie d'avancé , Programmation des automates S7-300 – Introduction au logiciel TIA Portal consulté le 31/05/2018.
- [15] brochure SIMATIC STEP 7 dans le portail TotallyIntegrated Automation Portal créée par siemens en novembre 2012, N° de référence : 6ZB5310-0PX03-0BA3, sur le site : [www.automation.siemens.com](http://www.automation.siemens.com) Consulté le 30/05/2018.
- [16] brochure SIMATIC WinCC Supervision de process avec Plant Intelligence créée par siemens en avril 2009, N° de référence : 6ZB5370-1CB03-0BA8, sur le site : [www.automation.siemens.com](http://www.automation.siemens.com) Consulté le 10/05/2018.
- [17] site : [www.automation-sense.com](http://www.automation-sense.com) créée le SEPTEMBER 2009 – 2017, consulté le 30/05/2018.
- [18] site : [www.ibhsoftec.com](http://www.ibhsoftec.com) consulté le 31/05/2018.
- [19] brochure guide de NI-USB-6001, sur le site : <http://www.ni.com> consulté le 01/06/2018.
- [20] Document guide technique Détecteurs photoélectriques, crée le novembre 2003, consulté le 01/06/2018.
- [21] site : [www.zpag.net/Electroniques/relais.htm](http://www.zpag.net/Electroniques/relais.htm)
- [21] site : [www.physique.vije.ne](http://www.physique.vije.ne) créée par Olivier Wajsfelner, professeur au lycée Pablo Neruda, St-Martin D'Hères (Isère). Consulté le 02/06/2018.
- [22] document de FORMATION CONTINUE - TECHNICIENS SUPERIEURS – INGENIEURS ELECTROTECHNICIENS MACHINE ASYNCHRONE Edition revue 2010 Bapio BAYALA, consulté le 02/06/2018.
- [23] Document Fiche technique de VS mini J7 créée par OMRON , No. I63E-FR-01.
- [24] site : [www.omega.fr](http://www.omega.fr) créée en 2003 par Omega Engineering, date de consultation : 05/06/2018.
- [25] site : [www.thermocoupleinfo.com](http://www.thermocoupleinfo.com) créée le date 2011 par THERMOCOUPLEINFO.COM, fournies par REOTEMP Instrument Corporation. Consulté le : 06/06 /2018.
- [26] site : [www.surplus.motionconstrained.com](http://www.surplus.motionconstrained.com), créée en 2018, consulté le 07/06/2018

[27] site : [www.automation-sense.com](http://www.automation-sense.com),automatisme,communication-entre-3-cpu-  
Consulté le 26/08/2015.

[28] site : [www.technologuepro.com](http://www.technologuepro.com),cours-automate-programmable-industriel,Bus-  
de-terrain-Profibus.Consulté le 18-11-2017.

[29] site : [www.ni.com](http://www.ni.com), crée en 2014 par : « National Instruments » , titre : NI-DAQmx.

[30] l'aide du logiciel TIA PORTAL V13