

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 Projet de Fin d'Études

présenté par

BOUGRESSI RABEA

pour l'obtention du diplôme Master 2 en Électronique option Signaux en Ingénierie des
Système, Informatique Industrielle(SISII)

Thème

Intégration d'un système IHM-API OMRON pour le contrôle et la supervision d'un processus industriel

Proposé par : NOUREDDINE BENNILA

Année Universitaire 2014-2015

ملخص:

هذه الدراسة لمشروع نهاية الدراسة لطور الماستر تقدم عملية صناعية باستعمال المعدات القابلة للبرمجة مترابطة فيما بينها بما في ذلك

NS 12, (واجهة التحكم و التردد) IHM, CPU61, (جهاز الاتمته المبرمجة الصناعية) API وحدة تحكم (VS mini J7) من الشركة المصنعة . الإلكترونيات للسرعة للمحركات غير المتزامنة OMRON. من اجل التحكم و المراقبة و الاشراف على المنتوجات المنقولة بواسطة حزام ناقل نحو الروبوت لطبع و نقل المنتوج الى مكان التخزين

البرامج المستخدمة هي : CX مصمم ومبرمج-NS

كلمات المفاتيح:

API CQM1H CPU 61 -OMRON - VSmini J7. - آلة واجهة NS 12 -الإشراف – الرقابة

Résumé :

Ce projet de fin d'étude, présente un procédé industriel, mettant en œuvre des équipements programmables interconnectés notamment : un API CQM1H CPU 61 du constructeur OMRON, un variateur électronique de vitesse pour les moteurs asynchrones, VS mini J7 du même constructeur, et un IHM NS12 du même constructeur aussi, pour le contrôle et la supervision d'un produit transféré par une bande transporteuse vers un robot pour être stocké dans des conteneurs.

Les logiciels mis en œuvre sont : NS-Designer et CX Programmer

Mots clés : AUTOMATIQUE INDUSTRIEL - IHM NS 12 - API CQM1H CPU 61 - OMRON - VS mini J7- Supervision – Contrôle.

Abstract :

This final project study presents an industrial process, implementing programmable interconnected equipment including: a PLC OMRON CQM1H CPU 61 of the manufacturer, an electronic speed controller for asynchronous motors, VS mini J7 of the same manufacturer, and NS12 HMI the same manufacturer as, for the control and supervision of a product transferred by a conveyor belt to a robot to be stored in containers.

Software used are : NS-Designer and CX Programmer.

Keywords : AUTOMATIQUE INDUSTRIEL - IHM NS 12 - API CQM1H CPU 61 - OMRON - VS mini J7- Supervision - Control

Introduction générale

Depuis plusieurs décennies, l'automatisation industrielle n'a cessée de connaître des progrès spectaculaires. En effet, l'évolution rapide des technologies nouvelles a permis de contourner la plupart des difficultés rencontrées dans le monde industriel, et a fourni plusieurs possibilités pour satisfaire les exigences et les critères demandés tels que la productivité, la sécurité, l'optimisation des coûts de production et l'amélioration des conditions de travail.

Actuellement, en raison d'une modernisation incessante des outils de production, les systèmes industriels deviennent de plus en plus complexes et sophistiqués.

L'automatisation des procédés industriels est actuellement l'un des axes où on fait appel, de plus en plus, aux technologies évoluées à mesure que les exigences du monde industriel ont aussi évolué. Parmi celles-ci, figurent les Automates Programmables Industriels (API), qui offrent la solution adaptée aux besoins exigés.

L'automate programmable industriel est l'organe principal de la boucle de réglage placée dans un procédé industriel, en vue de le contrôler. Il a pour tâche principale, la récolte des informations relatives à l'état du système, à partir des différents capteurs via ses interfaces d'entrées, et les traiter pour prendre une décision; et ainsi commander les actionneurs via ses interfaces de sorties suivant une logique de fonctionnement mise en évidence.

Ils peuvent être affectés par des défauts provenant de leurs composants internes (défaillance d'un moteur, défaillance ou dérive d'un capteur). Une défaillance d'une partie d'un processus peut s'aggraver et paralyser ou endommager tout le

système de production ce qui peut entraîner des pertes en vies humaines et des dommages sur le plan écologique et économique, surtout s'il s'agit des systèmes se trouvant dans des usines de productions.

Il est alors indispensable de mettre en œuvre des techniques fiables de surveillance et de contrôle de ces systèmes afin de détecter de façon précise l'apparition de signes d'anomalies. Des systèmes de surveillance sont alors mis en œuvre dont le but est de fournir à tout instant l'état de fonctionnement des différents organes qui constituent le processus technologique et de faire piloter et donner des commande a partir des stations de surveillance.

Aujourd'hui grâce au développement de la technologie et de l'informatique, le problème de la communication entre les équipements d'automatisme est résolu.

Le but de ce travail est l'élaboration d'une connexion entre des équipements OMRON tel que l'API et IHM, pour développer un procédé industriel et de l'intégré dans une chaine de production.

Le travail présenté dans ce mémoire comporte quatre chapitres :

- ◆ Le premier chapitre présente une généralité sur l'automatisation industrielle dans tous ces niveaux.
- ◆ Dans le deuxième chapitre, nous avons présenté les techniques de supervision, de pilotage, et de contrôle de procédés de fabrications.
- ◆ Le troisième chapitre a pour objectif de traiter l'application qu'on a développée, pour mettre en évidence les systèmes de contrôle et de surveillance évoqués dans le chapitre précédent, et de présenter le fonctionnement et la configuration du matériels mis en œuvre.
- ◆ Le quatrième et dernier chapitre sera consacré à la réalisation mis au point pour concrétiser tout ce qui a été développé dans les chapitres précédents.

Enfin, on terminera l'étude par une conclusion générale qui discutera les avantages apportés et les perspectives visées en termes de réalisation et installation.

Liste des tableaux

Tableau 3.1 : les différents modèles de VARISPEED J7. [22].....	88
Tableau 3.2 : Disruption de la console numérique.....	89
Tableau 3.3 : Disruptives des bornes du circuit de contrôle.....	91
Tableau 3.4 : Symboles Globaux.....	99
Tableau 3.5 : Zones mémoire de l'API DM et EM.....	102
Tableau 3.6 : Format de trame de transmission.....	104
Tableau 3.7 : la vitesse de transmission du trame.....	104
Tableau 4.1 : la mnémonique de notre programme.....	125
Tableau 4.2 : (suite) la mnémonique de notre programme.....	126

Tables des matières

CHAPITRE 1	AUTOMATISATION INDUSTRIELLE.....	ERREUR ! SIGNET NON DÉFINI.
1.1	Introduction.....	Erreur ! Signet non défini.
1.2	Les révolutions industrielles.....	Erreur ! Signet non défini.
1.2.1	La première révolution industrielle.....	<i>Erreur ! Signet non défini.</i>
1.2.2	La deuxième révolution industrielle.....	<i>Erreur ! Signet non défini.</i>
1.2.3	La troisième révolution industrielle.....	<i>Erreur ! Signet non défini.</i>
1.2.4	La quatrième révolution industrielle.....	<i>Erreur ! Signet non défini.</i>
1.3	L'Automatisation industrielle.....	Erreur ! Signet non défini.
1.3.1	Les systèmes automatisés de production :.....	<i>Erreur ! Signet non défini.</i>
1.3.2	Structure d'un système automatisé :.....	<i>Erreur ! Signet non défini.</i>
1.3.3	Traitement du programme automate :.....	<i>Erreur ! Signet non défini.</i>
1.4	L'automatisation et l'informatique.....	Erreur ! Signet non défini.
1.4.1	Conception et fabrication assistées par ordinateur..	<i>Erreur ! Signet non défini.</i>
1.4.2	Systèmes de production flexibles.....	<i>Erreur ! Signet non défini.</i>
1.4.3	Systèmes experts.....	<i>Erreur ! Signet non défini.</i>
1.5	L'informatique dans l'industrie.....	Erreur ! Signet non défini.
1.6	Evolution des architectures d'automatisme.....	Erreur ! Signet non défini.
1.6.1	Les automatismes centralisés:.....	<i>Erreur ! Signet non défini.</i>
1.6.2	Les automatismes décentralisés.....	<i>Erreur ! Signet non défini.</i>
1.6.3	La décentralisation des entrées/sorties et de la périphérie d'automatisme	<i>Erreur ! Signet non défini.</i>
1.7	RESEAUX DE TERRAIN :.....	Erreur ! Signet non défini.
1.8	Réseaux Locaux Industriels.....	Erreur ! Signet non défini.
1.8.1	Qu'est-ce qu'un réseau local industriel ?.....	<i>Erreur ! Signet non défini.</i>
1.8.2	Critères de choix d'un réseau (RLI).....	<i>Erreur ! Signet non défini.</i>
1.8.3	Hierarchie d'une Entreprise de fabrication:.....	<i>Erreur ! Signet non défini.</i>
1.8.4	Les réseaux de communication en automatisme industriel	<i>Erreur ! Signet non défini.</i>
a	Bus capteurs et actionneurs.....	Erreur ! Signet non défini.
b	Bus de terrain.....	Erreur ! Signet non défini.
c	Réseaux locaux industriels (RLI).....	Erreur ! Signet non défini.
d	Réseaux Ethernet TCP/IP.....	Erreur ! Signet non défini.
1.9	Evolution de la communication, les grandes tendances...	Erreur ! Signet non défini.
1.9.1	L'impact des NTIC sur les architectures d'automatismes	<i>Erreur ! Signet non défini.</i>
1.10	Le modèle OSI (Open System Interconnections).....	Erreur ! Signet non défini.

1.10.1	L'encapsulation.....	<i>Erreur ! Signet non défini.</i>
1.11	Topologies des réseaux.....	Erreur ! Signet non défini.
1.11.1	Classification selon la taille.....	<i>Erreur ! Signet non défini.</i>
1.11.2	Classification suivant le type c'est la partie physique du réseau :	<i>Erreur ! Signet non défini.</i>
1.11.3	Différents types d'interconnexion de réseaux.....	<i>Erreur ! Signet non défini.</i>
1.12	Systèmes de contrôle commande.....	Erreur ! Signet non défini.
1.12.1	Le système de contrôle commande par PLC.....	<i>Erreur ! Signet non défini.</i>
1.12.2	Le système de contrôle, commande et supervision SCADA	<i>Erreur ! Signet non défini.</i>
1.12.3	Le système de contrôle, commande et supervision DCS	<i>Erreur ! Signet non défini.</i>
1.13	L'utilisation des tablettes électroniques en automatisation industrielle	Erreur ! Signet non défini.
1.13.1	Visualisation et contrôle des opérations.....	<i>Erreur ! Signet non défini.</i>
1.13.2	Calibration et ajustements.....	<i>Erreur ! Signet non défini.</i>
1.14	Conclusion.....	Erreur ! Signet non défini.
CHAPITRE 2 TECHNIQUES INDUSTRIELLES DE CONTRÔLE ET PILOTAGE DE PROCÉDÉS DE FABRICATION ERREUR ! SIGNET NON DÉFINI		
2.1	Introduction.....	Erreur ! Signet non défini.
2.2	Les Automates Programmables Industriels (API).....	Erreur ! Signet non défini.
2.3	La supervision et son intérêt.....	Erreur ! Signet non défini.
2.3.1	Définition.....	<i>Erreur ! Signet non défini.</i>
2.3.2	L'intérêt de la supervision.....	<i>Erreur ! Signet non défini.</i>
2.3.3	l'IHM (Interface Homme-Machine).....	<i>Erreur ! Signet non défini.</i>
2.3.4	Les systèmes de contrôle SCADA.....	<i>Erreur ! Signet non défini.</i>
a	Définition.....	Erreur ! Signet non défini.
b	Caractéristiques du SCADA.....	Erreur ! Signet non défini.
c	SCADA hardware.....	Erreur ! Signet non défini.
d	SCADA software.....	Erreur ! Signet non défini.
e	SCADA et les réseaux locaux.....	Erreur ! Signet non défini.
2.3.5	Le DCS (Distributed control System).....	<i>Erreur ! Signet non défini.</i>
a	Application des systèmes DCS :	Erreur ! Signet non défini.
2.4	Les constructeurs des systèmes SCADA et DCS et les logiciels mis en œuvre	Erreur ! Signet non défini.
2.4.1	Introduction.....	<i>Erreur ! Signet non défini.</i>
2.4.2	Siemens.....	<i>Erreur ! Signet non défini.</i>
a	Simatic Manager S7 (logiciel pour API).....	Erreur ! Signet non défini.
b	WinCC flexible (logiciel pour IHM).....	Erreur ! Signet non défini.
c	Wincc (logiciel pour SCADA).....	Erreur ! Signet non défini.
2.4.3	Schneider.....	<i>Erreur ! Signet non défini.</i>
a	Unity pro (logiciel pour API).....	Erreur ! Signet non défini.

b	Vijeo Designer (logiciel pour IHM).....	Erreur ! Signet non défini.
c	Vijeo Citect (logiciel pour SCADA).....	Erreur ! Signet non défini.
2.4.4	OMRON.....	<i>Erreur ! Signet non défini.</i>
a	CX-Programmer.....	Erreur ! Signet non défini.
b	CX-Designer.....	Erreur ! Signet non défini.
c	CX-Supervisor.....	Erreur ! Signet non défini.
2.5	Conclusion.....	Erreur ! Signet non défini.

CHAPITRE 3 DÉVELOPPEMENT D'UN PROCÉDÉ DE FABRICATION PILOTÉ PAR DES ÉQUIPEMENTS OMRON.. **ERREUR ! SIGNET NON DÉFINI.**

3.1	Introduction	Erreur ! Signet non défini.
3.2	Description du procédé industriel	Erreur ! Signet non défini.
3.3	Les outils d'automatisme mis en œuvre	Erreur ! Signet non défini.
3.3.1	L'API CQM1H	<i>Erreur ! Signet non défini.</i>
a	Configuration du système	Erreur ! Signet non défini.
a.1	Interruptions	Erreur ! Signet non défini.
a.2	Sortie d'impulsions	Erreur ! Signet non défini.
a.3	Communications	Erreur ! Signet non défini.
b	Les modules du CQM1H :	Erreur ! Signet non défini.
B-1	Unités d'alimentation	Erreur ! Signet non défini.
B-2	Unités centrales	Erreur ! Signet non défini.
B-3	les modules d'entrées/sorties	Erreur ! Signet non défini.
3.3.2	L'IHM NS 12	<i>Erreur ! Signet non défini.</i>
3.3.3	Le variateur de vitesse VS mini J7	<i>Erreur ! Signet non défini.</i>
3.3.4	L'encodeur (réf : RE5402508 /24, constructeur : elap).....	<i>Erreur ! Signet non défini.</i>
3.3.5	Les cellules photo électrique	<i>Erreur ! Signet non défini.</i>
3.4	Programmation de l'API CQM1H et l'IHM NS12	Erreur ! Signet non défini.
a	Présentation du Logiciel	Erreur ! Signet non défini.
1)	Création et gestion de projets :	Erreur ! Signet non défini.
b	Les outils et les instructions de Programmation	Erreur ! Signet non défini.
1)	Déclaration des contacts et bobines :	Erreur ! Signet non défini.
d	Mode de fonctionnement	Erreur ! Signet non défini.
c	Transferts du programme dans l'IHM :	Erreur ! Signet non défini.
3.4.3	Configuration du variateur VS mini J7	<i>Erreur ! Signet non défini.</i>
3.5	Conclusion	Erreur ! Signet non défini.

CHAPITRE 4 DÉVELOPPEMENT DE LA RÉALISATION ET RÉSULTATS OBTENUS.. **ERREUR ! SIGNET NON DÉFINI.**

4.1	Introduction	Erreur ! Signet non défini.
-----	--------------------	------------------------------------

- 4.2 Présentation des interfaces des programmes (API, IHM, VS mini J7) **Erreur ! Signet non défini.**
 - 4.2.1 Programmation de l'API **Erreur ! Signet non défini.**
 - a Mnémonique du programme utilisé : **Erreur ! Signet non défini.**
 - b Aperçu du programme dans le CX Programmer **Erreur ! Signet non défini.**
 - 4.2.2 La configuration effectué du variateur VS mini J7 **Erreur ! Signet non défini.**
 - 4.2.3 Les résultats des fenêtres réalisées pour l'IHM **Erreur ! Signet non défini.**
 - 4.3 Conclusion **Erreur ! Signet non défini.**

Figure 1.1 : l'époque de la première révolution industrielle.....	3
Figure 1.2 : l'époque de la deuxième révolution industrielle.....	4
Figure 1.3 : l'époque de la troisième révolution industrielle.....	5
Figure 1.4 : La dernière révolution industrielle.....	6
Figure 1.5 : un système automatisé de production.....	8
Figure 1.6 : Décomposition d'un système automatisé.....	9
Figure 1.7 : logigramme du traitement d'un programme automate.....	10
Figure 1.8 : Les automatismes centralisés.....	15
Figure 1.9 : Les automatismes décentralisés.....	16
Figure 1.10 : La décentralisation des E/S et de la périphérie d'automatisme.....	17
Figure 1.11 : hiérarchie d'une Entreprise de fabrication.....	20
Figure 1.12: Positionnement des principaux Réseaux et Bus.....	26
Figure 1.13 : L'impact des NTIC sur les architectures d'automatismes.....	27
Figure 1.14 : Le modèle OSI.....	28
Figure 1.15 : Mécanisme d'encapsulation.....	29
Figure 1.16 : topologie point à point.....	30
Figure 1.17 : topologie en étoile.....	31
Figure 1.18: topologie en Bus.....	31
Figure 1.19 : topologie en arbre.....	32
Figure 1.20: topologie en maille.....	32
Figure 1.21: topologie en anneau.....	33
Figure 1.22 : Différents types d'interconnexions de réseaux.....	33
Figure 1.23 : Les différents types des tablettes industriels.....	37
Figure 2.1 : CPU d'un automate.....	41
Figure 2.2 : Quelques APIs vendus par des constructeurs de renommé mondiale.....	42
Figure 2.3 :Exemple d'un synoptique d'un process de gestion de traitement de boue.....	45

Figure 2.4 : Un IHM industriel.....	45
Figure 2.5 : Gestion des alarmes, défaillances.....	47
Figure 2.6 : exemple de supervision des pipelines.....	48
Figure 2.7 : Communication entre SCADA Master et RTU Site.....	50
Figure 2.8 : Typique d'un Système SCADA.....	51
Figure 2.9 : SCADA et le réseau LAN.....	53
Figure 2.10 : Ethernet utilisé pour transférer des données sur un système SCADA....	53
Figure 2.11: <i>Distributed</i> control system (DCS).....	54
Figure 2.12: Distributed control system (DCS).....	55
Figure 2.13 : Un exemple des domaines d'applications du système DCS.....	56
Figure 2.14 : Plateforme logiciel SIMATIC S7.....	58
Figure 2.15. Plateforme logiciel Siemens TIA PORTAL.....	59
Figure 2.16. Plateforme logiciel WinCC flexible.....	60
Figure 2.17 : Exemple d'une vue crée par WINCC.....	62
Figure 2.18 : Plateforme logiciel de l'UNITY PRO.....	64
Figure 2.19 : Plateforme logiciel Vijeo Designer.....	65
Figure 2.20 : Plateforme logiciel Vijeo Citect.....	66
Figure 2.21 : Exemple d'une vue crée par Vijeo Citect.....	66
Figure 2.22 : Exemple d'un programme crée par CX Programmer.....	67
Figure 2.23 : Exemple d'une vue crée par CX Designer.....	69
Figure 2.24 : Vue crée par CX-Supervisor.....	70
Figure 2.25 : Plateforme du logiciel CX-Supervisor.....	70
Figure 3.1 : emballage des paquets de 1m ²	73
Figure 3.2 : photo de la réalisation, les équipements utilisés.....	74
Figure 3.3 : API OMRON CQM1H.....	75
Figure 3.4 : les étapes à respecter pour l'installation et l'utilisation d'un système CQM1H.	76
Figure 3.5 : Description d'un Automate Programmable Industriel CQM1H.....	77
Figure 3.6 : Unité Centrale et les unités d'E/S d'un API CQM1H CPU 51/61.....	78

Figure 3.7 : les différents types de liaisons.....	80
Figure 3.8 : assemblage des modules du CQM1H.....	81
Figure 3.9 : unité d'alimentation CQM1-PA203.....	81
Figure 3.10 : Unité centrale du l'API CQM1H CPU 61.....	82
Figure 3.11 : composant de bas des cartes d'E/S.....	83
Figure 3.12 : NS 12 d'OMRON.....	83
Figure 3.13 : Vue côté connecteur du NS12.....	84
Figure 3.14 : la manière d'ouvrir le menu de notre IHM.....	85
Figure 3.15 : le menu du l'écran NS.....	86
Figure 3.16 : Variateur de vitesse mini J7.....	87
Figure 3.17 : console numérique du VS mini J7.....	88
Figure 3.18 : les bornes du circuit de contrôle.....	90
Figure 3.19 : Encodeur rotatif absolue.....	92
Figure 2.20 : modèles de capteurs infrarouges.....	93
Figure 3.21 : les trois types des capteurs à proximité.....	93
Figure 3.22 : fenêtre d'accueil du CX Programmer.....	95
Figure 3.23 : fenêtre du choix d'API.....	96
Figure 3.24 : fenêtre Paramètres type API.....	96
Figure 3.25 : la fenêtre de programmation	97
Figure 3.26 : L'onglet du projet.....	98
Figure 3.27 : Paramètres d'API.....	100
Figure 3.28 : instruction étendue.....	100
Figure 3.29 : Mémoires de l'API.....	101
Figure 3.30 : programme dans l'onglet projet	105
Figure 3.31 : les types des câbles qu'on peut utiliser.....	106
Figure 3.32 : le chemin de chargement du programme dans l'API.....	106
Figure 3.33 : L'instruction KEEP.....	107
Figure 3.34 : L'instruction Différentiation Front montant : DIFFU(13).....	107
Figure 3.38 : L'instruction de déplacement (MOV).....	108

Figure 3.39 : L’instruction de comparaison CMP(20).....	108
Figure 3.40: les types des câbles qu’on peut utiliser.....	111
Figure 3.41 : le chemin de chargement du programme dans l’API.....	112
Figure 3.42 : CX Simulator.....	112
Figure 3.43 : l’espace de travail du CX Designer.....	114
Figure 3.44 : configuration du Bouton ON/OFF.....	116
Figure 3.45 : paramétrage d’adresse du bouton ON/OFF crée.....	117
Figure 3.46 : le transfert du programme vers l’IHM.....	118
Figure 3.47 : schéma de câblage du VS mini J7.....	119
Figure 4.1 : Description des matériels utilisés.....	123
Figure 4.3 : une partie de la programmation de notre API.....	126
Figure 4.4 : Ecrans d’accueil de notre projet.....	127
Figure 4.5 : le menu de notre IHM.....	128
Figure 4.6 : info du notre IHM.....	128
Figure 4.7 : le SETUP de l’IHM.....	129
Figure 4.8 : Historique des alarmes.....	129
Figure 4.9 : la commande manuelle de la vitesse.....	130
Figure 4.10 : la fenêtre de la commande automatique.....	131
Figure 4.11 : fenêtre de contrôle-commande de fréquence.....	132

1.1 Introduction

L'automatisation s'est généralisée à l'ensemble des activités de production, tant dans l'industrie, que dans les activités de services. Quelque soit son domaine d'application et les techniques auxquelles elle fait appel, l'automatisation s'est constamment développée dans l'unique but de réduire la pénibilité du travail humains et d'améliorer la productivité.

Dans ce chapitre, nous allons d'abord donner une généralité sur l'automatisation, dans tous ses niveaux industriels.

1.2 Les révolutions industrielles

Une révolution industrielle a toujours été précédée de progrès scientifiques, techniques et organisationnels. C'est encore le cas, aujourd'hui, avec l'adoption de la numérisation par des pans entiers de l'économie ; Informatique et télécommunications associées portent ce phénomène qui annonce la 4^{ème} révolution industrielle. [1]

1.2.1 La première révolution industrielle



FIGURE 1.1 : L'ÉPOQUE DE LA PREMIÈRE RÉVOLUTION INDUSTRIELLE.

La première révolution industrielle repose sur le charbon, la métallurgie, le textile et la machine à vapeur. Elle démarre en Grande-Bretagne à la fin du XVIIIe siècle, puis se propage en France au début du XIXe siècle avant de s'étendre en Allemagne, aux États-Unis, au Japon et à la Russie. [1]

Mais l'industrie n'aurait jamais vu le jour sans progrès scientifiques et techniques.

Plusieurs évènements se sont produits à cette période, à savoir :

- les améliorations apportées en 1705 par Thomas Newcomen, à la machine à vapeur et l'extension de son utilisation à l'industrie;
- la première machine à tisser mécanique avec moteur hydraulique (Richard Arkwright en 1769);
- L'amélioration de la machine à vapeur (James Watt en 1769) ;
- le premier essai d'une locomotive à vapeur (1804)...

1.2.2 La deuxième révolution industrielle



Figure 1.2 : L'époque de la deuxième révolution industrielle.

La deuxième révolution industrielle, qui a commencé à la fin du XVIIIe siècle, trouve ses fondements dans l'électricité, la mécanique, le pétrole et la chimie ou encore l'apparition de moyens de communication (télégraphe et téléphone) et le succès qu'a connu le transport collectif grâce au développement des chemins de fer

et des bateaux à vapeur. Les moyens de communication et de transport favorisent largement les échanges internationaux. [1]

1.2.3 La troisième révolution industrielle



Figure 1.3 : L'époque de la troisième révolution industrielle.

Une troisième révolution se produit au milieu du XXe siècle, dont la dynamique vient d'être générée par l'électronique, les télécommunications, de l'informatique, de l'audiovisuel et le nucléaire. Ces derniers rendent possibles la production de matériels miniaturisés, de robots et l'automatisation poussée de la production, le développement des technologies spatiales et celui des biotechnologies.

La troisième révolution industrielle a vu naître également Internet, au crépuscule du XXe siècle. Le véritable démarrage de l'électronique miniature date de l'arrivée du transistor et des circuits intégrés, c'est l'origine du microprocesseur, pièce maîtresse de tous les produits électroniques dits intelligents, notamment les ordinateurs.

L'informatique en général et la commutation de paquets en particulier n'existerait pas sans l'invention du datagramme par le Français Louis Pouzin.

Deux inventions industrielles : l'automate et le robot produits ont particulièrement impacté la production.

Inventé en 1968 par l'Américain Richard Morley, l'automate programmable industriel (API), destiné au contrôle-commande d'une machine ou d'un processus, s'est imposé dans toutes les industries, puis au fil du temps aux transports, à la gestion technique des bâtiments, etc.

Le robot industriel, sorti de l'imagination de Georges Devol et le visionnaire Joseph Engelberger, fut d'abord destiné aux opérations de manutention, puis aux tâches de production: soudage, assemblage, etc. [1]

1.2.4 La quatrième révolution industrielle



Figure 1.4 : La dernière révolution industrielle.

La dernière révolution industrielle est en train de prendre forme sous nos yeux, à l'aube de ce XXI^e siècle. Elle sera mûre au plus tôt vers 2020. Toutes les briques technologiques sur lesquelles elle est bâtie sont là. Cette révolution peut être résumée par la numérisation poussée à l'extrême des échanges économiques et productifs.

L'Industrie 4.0 suppose une intégration horizontale. On réalise tout de A à Z en interaction entre les produits et les machines, et les machines entre elles. Nous sommes dans un système global interconnecté. [1]

1.3 L'Automatisation industrielle

L'automatisation industrielle est l'art d'utiliser les machines afin de réduire la charge du travailleur tout en gardant la productivité et la qualité.

L'automatisation industrielle fait appel à des systèmes électroniques qui englobent toute la hiérarchie de contrôle-commande depuis les capteurs de mesure, en passant par les automates, les bus de communication, la visualisation, l'archivage jusqu'à la gestion de production et des ressources de l'entreprise.[1]

Un automatisme est un sous-ensemble d'une machine, destinée à remplacer l'action de l'être humain dans des tâches en général simples et répétitives, réclamant précision et rigueur. L'industrie est passée d'un système dit manuel, à un système mécanisé, puis au système automatisé.

Dans l'industrie, les automatismes sont devenus indispensables: ils permettent d'effectuer quotidiennement les tâches les plus ingrates, répétitives et, dangereuses.

Parfois, ces automatismes sont d'une telle rapidité et d'une telle précision, qu'ils réalisent des actions impossibles pour un être humain.

L'automatisme est donc synonyme de productivité et de sécurité.

Le savoir-faire de l'opérateur est transposé dans le système automatisé, il devient le processus.

Un processus peut-être considéré comme un système organisé d'activités qui utilise des ressources (personnel, équipement, matériels et machines, matière première et informations) pour transformer des éléments entrants en éléments de sortie dont le résultat final attendu est un produit.

Le système automatisé de production industrielle doit également répondre à des contraintes économiques et donc à une certaine flexibilité. Il existe deux concepts technologiques :

- Les automatismes séquentiels,
- Les asservissements, la régulation ou « feed-back control ».

Cependant, dans la plupart des systèmes complexes modernes, ces deux types d'automatismes se trouvent imbriqués. [2]

1.3.1 Les systèmes automatisés de production :

L'objectif de l'automatisation des systèmes est de produire, en ayant recours le moins possible à l'homme, des produits de qualité et ce pour un coût le plus faible.

Un système automatisé est un ensemble d'éléments en interaction, et organisés dans un but précis : agir sur une matière d'œuvre afin de lui donner une valeur ajoutée. [3]

Le système automatisé est soumis à des contraintes : énergétiques, de configuration, de réglage et d'exploitation qui interviennent dans tous les modes de marche et d'arrêt du système.

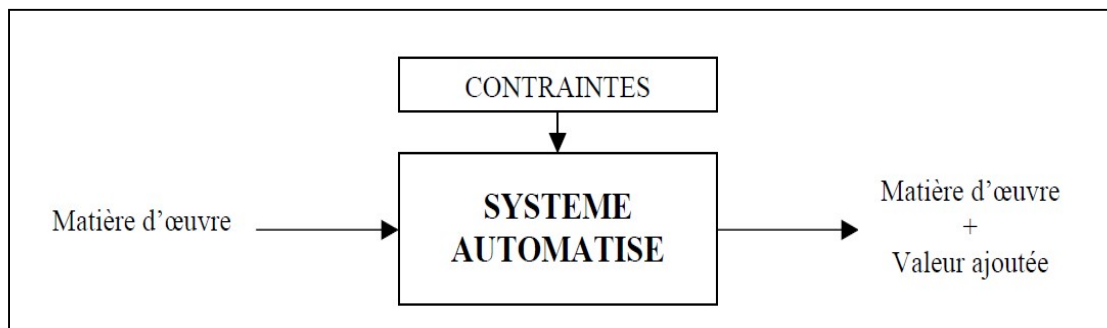


Figure 1.5 : Un système automatisé de production.[3]

1.3.2 Structure d'un système automatisé :

Tout système automatisé peut se décomposer selon le schéma ci-dessous :

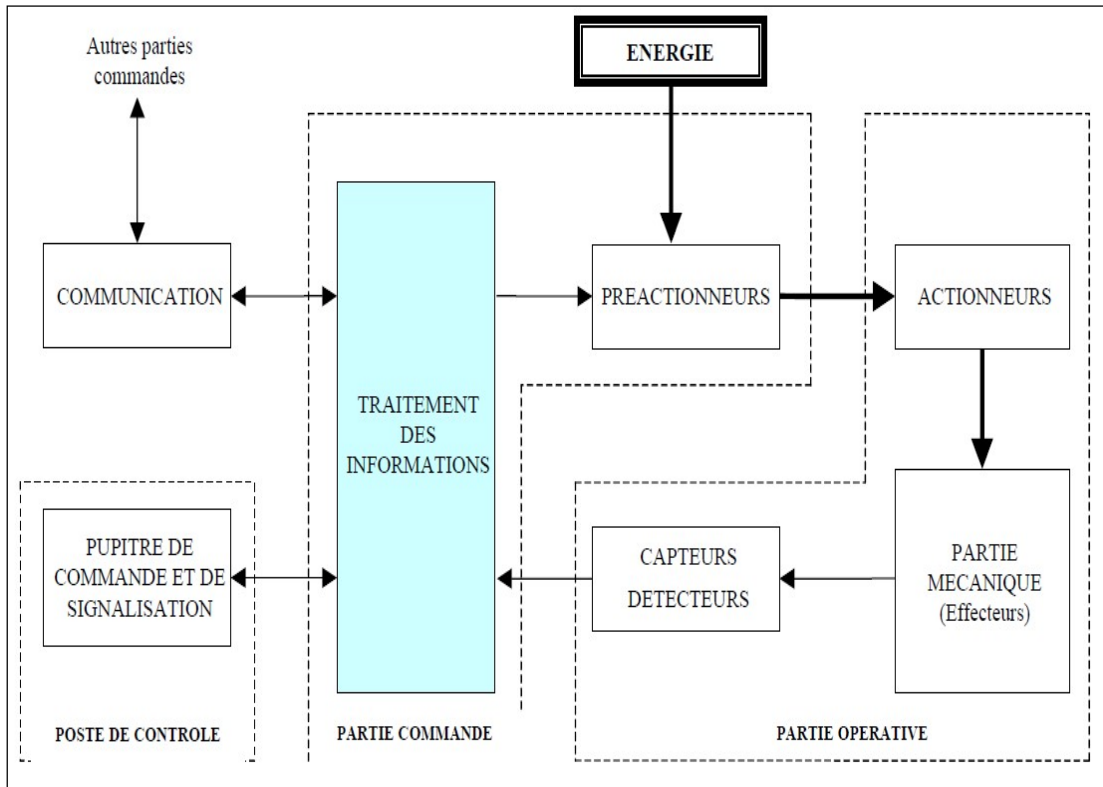


Figure 1.6 : Décomposition d'un système automatisé. [3]

- Partie opérative :

Elle agit sur la matière d'œuvre afin de lui donner sa valeur ajoutée.

Les actionneurs (moteurs, vérins) agissent sur la partie mécanique du système qui agit à son tour sur la *matière d'œuvre*.

Les capteurs / détecteurs permettent d'acquérir les divers états du système.

- Partie commande :

Elle donne les ordres de fonctionnement à la partie opérative.

Les pré-actionneurs permettent de commander les actionneurs ; ils assurent le transfert d'énergie entre la source de puissance (réseau électrique, pneumatique ...) et les actionneurs. Exemple : contacteur, distributeur ...

Ces pré-actionneurs sont commandés à leur tour par le bloc traitement des informations.

Celui-ci reçoit les consignes du pupitre de commande (opérateur) et les informations de la partie opérative transmises par les capteurs / détecteurs.

En fonction de ces consignes et de son programme de gestion des tâches (implanté dans un automate programmable ou réalisé par des relais (on parle de logique câblée)), elle va commander les pré-actionneurs et renvoyer des informations au pupitre de signalisation ou à d'autres systèmes de commande et/ou de supervision en utilisant un réseau et un protocole de communication.

- Poste de contrôle :

Composé des pupitres de commande et de signalisation, il permet à l'opérateur de commander le système (marche, arrêt, départ cycle ...).

Il permet également de visualiser les différents états du système à l'aide de voyants, de terminal de dialogue ou d'interface homme-machine (IHM). [3]

1.3.3 Traitement du programme automate :

Tous les automates fonctionnent selon le même mode opératoire :

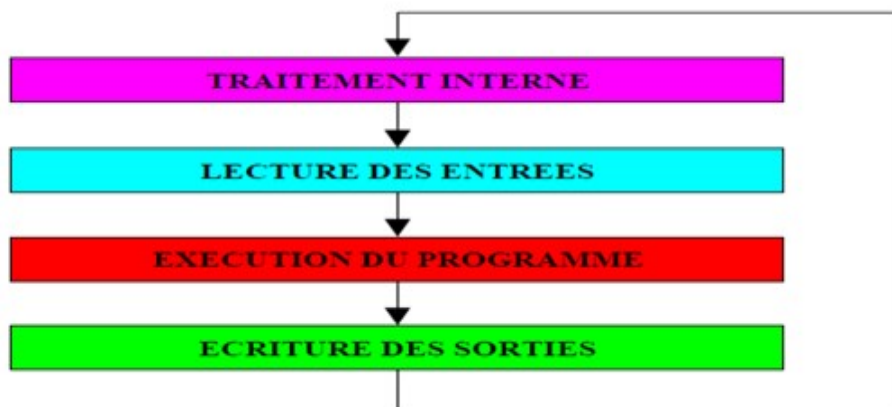


Figure 1.7 : Logigramme du traitement d'un programme automate. [3]

- Traitement interne : L'automate effectue des opérations de contrôle et met à jour certains paramètres systèmes (détection des passages en RUN / STOP, mises à jour des valeurs de l'horodateur, ...).
- Lecture des entrées : L'automate lit les entrées (de façon synchrone) et les recopie dans la mémoire image des entrées.
- Exécution du programme : L'automate exécute le programme instruction par instruction et écrit les sorties dans la mémoire image des sorties.
- Ecriture des sorties : L'automate bascule les différentes sorties (de façon synchrone) aux positions définies dans la mémoire image des sorties.

Ces quatre opérations sont effectuées continuellement par l'automate (fonctionnement cyclique). On appelle *scrutation* l'ensemble des quatre opérations réalisées par l'automate et le *temps de scrutation* est le temps mis par l'automate pour traiter la même partie de programme. Ce temps est de l'ordre de la dizaine de millisecondes pour les applications standards. [3]

1.4 L'automation et l'informatique

L'introduction de l'informatique dans les processus de fabrication a considérablement accéléré l'automatisation, en facilitant l'utilisation des boucles de rétroaction et en augmentant la flexibilité des systèmes de production.

En effet, avec l'avènement des ordinateurs sont apparues des machines à commande numérique, dont les mouvements sont enregistrés sur une unité de stockage, et qui peuvent accomplir plusieurs opérations d'usinage différentes. Les ordinateurs mis en réseau ou isolés, peuvent accomplir des tâches complexes, non seulement de contrôle par des boucles de rétroaction, mais aussi de pilotage de machines, de traitement de données, de circulation de l'information et de simulation. Ils sont utilisés à tous les stades du processus productif (conception, production, contrôle de la qualité des produits).[4]

1.4.1 Conception et fabrication assistées par ordinateur

Plus récemment, les progrès de l'informatique ont donné naissance aux techniques de conception et de fabrication assistées par ordinateur, qui favorisent encore davantage l'automatisation des processus de production.

Ce système permet d'intégrer et de visualiser toutes les étapes nécessaires à la fabrication d'un produit. [4]

1.4.2 Systèmes de production flexibles

Les systèmes de production flexibles développés par l'informatique caractérisent les manifestations actuelles de l'automatisation. Ces systèmes permettent d'équiper des entreprises qui produisent en petites séries et qui, par souci de rentabilité, ne peuvent donc automatiser complètement leur production.

Dans ces entreprises, un ou plusieurs ordinateurs sont utilisés afin de surveiller et diriger tout le fonctionnement de l'usine, depuis la planification de chaque étape de la production jusqu'au suivi des stocks de pièces et de l'utilisation des outils. L'opérateur humain reste cependant maître des décisions, mais dispose ainsi à tout moment d'informations précises sur la production, les stocks, le carnet de commandes. [4]

1.4.3 Systèmes experts

Les techniques d'automatisation décrites jusqu'ici permettent d'effectuer par des machines des tâches manuelles entrant dans le processus de production. Les systèmes experts peuvent être considérés comme le prolongement de ces techniques, qui tentent de modéliser et d'automatiser des travaux de nature intellectuelle. Ces systèmes informatiques se composent de bases de données associées à un

ordinateur : une base d'expertise est élaborée qui consiste en un ensemble de faits décrivant de manière précise des types de situations reliées entre elles par un moteur d'inférence capable de proposer face à une situation inédite une solution appropriée, en effectuant des déductions à partir des faits en s'aidant des règles contenues dans la base d'expertise.

Il est certain que les progrès attendus dans le domaine des systèmes experts et plus généralement de l'intelligence artificielle permettront de franchir de nouvelles étapes dans l'automatisation des tâches techniques. [4]

1.5 L'informatique dans l'industrie

L'informatique est un domaine d'activité scientifique, technique et industriel concernant le traitement automatique de l'information via l'exécution de programmes informatiques par des machines : des systèmes embarqués, des ordinateurs, des robots, des automates, etc.

L'informatique industrielle et technologique constitue un domaine très vaste, qui s'étend des applications utilisées dans la recherche et le développement jusqu'à celles qui sont dédiées à la production. Elle englobe un grand nombre d'environnements : le monde de la prestation informatique et du conseil, l'industrie ou les bureaux d'études. De nombreux secteurs y ont recours : le médical, les télécommunications, l'aéronautique, l'automobile...

Dans l'industrie, l'informatique représente en général le système nerveux de l'entreprise. Elle comprend principalement la CAO (conception assistée par ordinateur), la FAO (fabrication assistée par ordinateur), qui permet de programmer et de commander les outils de production ainsi que la GPAO (gestion de production assistée par ordinateur), destinée à assurer, par le biais de l'informatique, le suivi de tout le cycle de production.

1.6 Evolution des architectures d'automatisme

Un réseau local industriel est utilisé dans une usine ou tout système de production, pour connecter diverses machines afin d'assurer la commande, la surveillance, la supervision, la conduite, la maintenance, le suivi de produit, la gestion, en un mot, l'exploitation de l'installation de production.

Depuis que les automatismes sont réalisés sur la base d'unités de traitement (automates programmables), les architectures ont fortement évolué et sont passées par différents stades pour arriver aux architectures actuelles, basées sur l'adoption des grands standards de communication et sur l'arrivée des NTIC (Nouvelles Technologies de l'Information et de la Communication). [5]

1.6.1 Les automatismes centralisés:

Jusqu'aux années 80, les automatismes, s'appuyant sur des automates programmables industriels (API) traitaient essentiellement des fonctions séquentielles. En simplifiant, les API :

- ✓ Géraient des demandes d'exécution et d'état de l'automatisme (image des entrées),
- ✓ Elaboraient des demandes d'exécution d'actions (positionnement des sorties).

Par la suite, les API ont été amenés à gérer de nombreuses fonctions complémentaires comme des fonctions métier, des fonctions de diagnostic système et application, etc.

Les automatismes centralisés géraient tout un ensemble de fonctions qui n'avaient pas forcément d'interactions entre elles.

Lorsqu'il y avait déjà un automate dans l'usine, les automaticiens qui devaient intégrer une fonction supplémentaire se posaient simplement la question : l'automate ou le système d'automatisme en place peut-il gérer les E/S supplémentaires et quelle est la capacité de mémoire disponible ?

Bien souvent, l'automatisation supplémentaire était réalisée avec cet automate existant, même si elle n'avait aucun rapport avec l'automatisme résident.

Ces automatismes centralisés amenaient des nombreuses contraintes :

- aucune autonomie des différents sous-ensembles,
 - mise en service et maintenance lourdes et difficiles à effectuer du fait de la quantité d'E/S gérées,
 - arrêt de l'ensemble des fonctions gérées par l'API en cas de défaut système de cet API ou d'arrêt pour la maintenance du moindre élément de l'outil de production.
- [5]

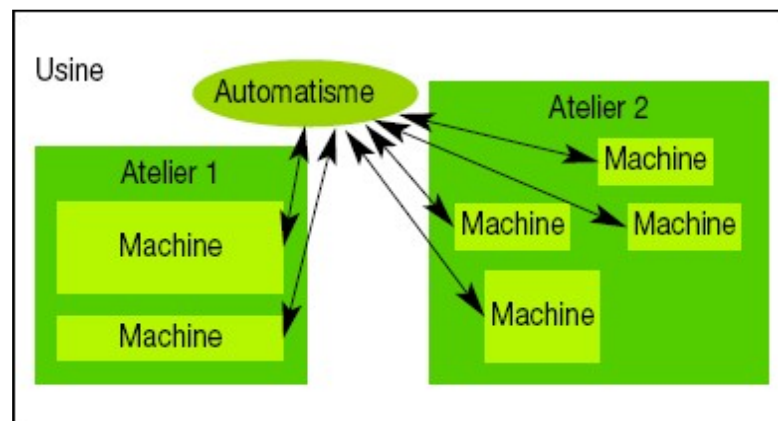


Figure 1.8 : Les automatismes centralisés. [5]

1.6.2 Les automatismes décentralisés

Du fait des contraintes imposées par les systèmes centralisés, les utilisateurs se sont orientés vers une segmentation de l'architecture. Celle-ci a été faite en découpant l'automatisme en entités fonctionnelles.

Elle permet de simplifier les automatismes en réduisant le nombre d'E/S gérées et présente donc l'avantage de faciliter la mise en service et la maintenance.

Cette segmentation a généré le besoin de communication entre les entités fonctionnelles. La fonction de communication est devenue la clef de voûte de la conception des architectures d'automatismes.

Les constructeurs d'API ont donc créé des offres de réseaux locaux industriels (RLI) afin d'assurer une communication efficace entre les différents API (voir figure 1.9). [5]

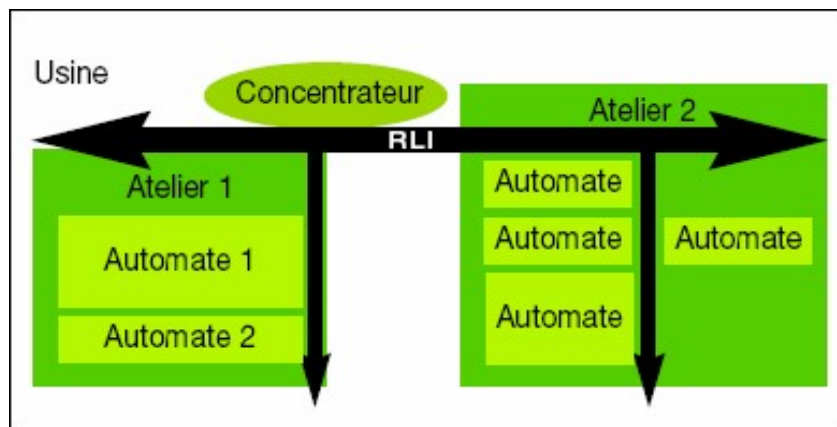


Figure 1.9 : Les automatismes décentralisés. [5]

1.6.3 La décentralisation des entrées/sorties et de la périphérie d'automatisme

A la demande des utilisateurs finaux, notamment pour faire baisser les coûts de câblage, il a été nécessaire de prendre en compte la topologie des automatismes.

Sur des sites plus étendus, il est souvent nécessaire de gérer un nombre de points diffus importants et de prendre en compte les fonctions métier réparties (variation de vitesse, dialogue homme/machine, pesage...).

La réponse des constructeurs de produits d'automatismes est arrivée avec les réseaux et bus de terrain.

Ceux-ci ont permis de gérer dans un premier temps des E/S décentralisées puis la périphérie d'automatisme (voir figure 1.10).

Ces réseaux de terrain contribuent à réaliser des gains de câblage importants, mais surtout ils permettent de rendre accessibles des services (diagnostic, programmation...) sur tout le site. [5]

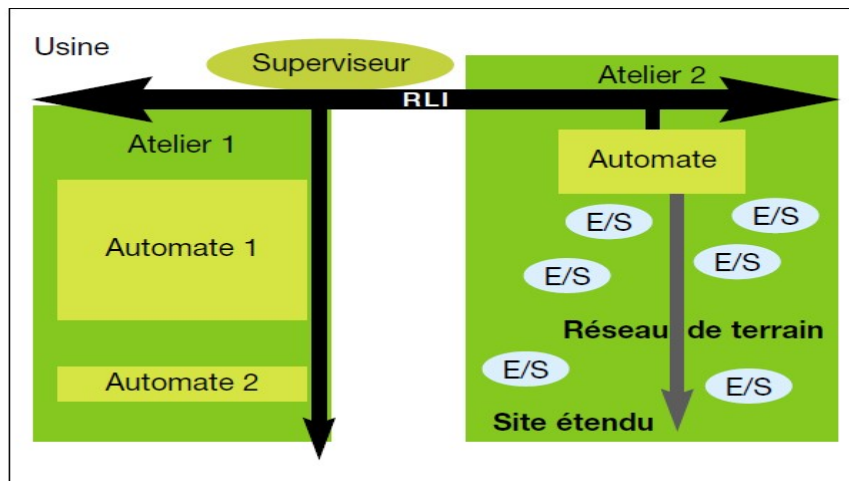


Figure 1.10 : La décentralisation des E/S et de la périphérie d'automatisme. [5]

1.7 RESEAUX DE TERRAIN :

Ceux-ci ont permis de gérer dans un premier temps des E/S décentralisées puis la périphérie d'automatisme.

Avantage :

- installation locale ou distante du produit plus facile;
- extension possible des applications;
- transferts de données vers des hôtes répartis pour le traitement et la supervision automatiques du produit;
- gestion, diagnostics et réparation à distance de l'équipement. [6]

❖ Ces réseaux de terrain contribuent à réaliser :

- des gains de câblage importants (suppression des boucles analogiques 4-20mA);
- intelligence déportée;
- mais surtout ils permettent de rendre accessibles des services sur tout le site: diagnostic, programmation ;
- réduction des coûts de maintenance.

Inconvénients :

- prise en compte de l'information en temps réel (en cours d'évolution) ;
- sécurité des informations ;
- choix du réseau :
 - Topologie Accès au bus,
 - Vitesse,
 - Standard ou Propriétaire,
 - coût direct du prix du Bus de Terrain. [6]

1.8 Réseaux Locaux Industriels

1.8.1 Qu'est-ce qu'un réseau local industriel ?

L'automatisation, puis l'informatisation des systèmes de production ont montré que pour l'amélioration de la productivité et de la diversité de la production, il ne suffisait pas de maîtriser la gestion de production, ou d'avoir une machine-outil à commande numérique sophistiquée, ou encore un système de contrôle-commande évolué et quelques capteurs et actionneurs autour. Encore faut-il faire communiquer tous ces éléments constitutifs d'un système de production dans sa globalité. Faire communiquer, c'est permettre d'échanger des informations de natures différentes et ce parfois nécessairement en temps réel.

Un nouvel outil informatique apparaît alors, avec sa problématique scientifique : le réseau local industriel. Son rôle est donc de faire dialoguer des équipements nombreux et divers, en intégrant une caractéristique importante, celle de fournir des services contraints par le temps.

Ces réseaux locaux industriels, parfois appelés abusivement réseaux de terrain, s'inscrivent donc à différents niveaux au sein de la pyramide CIM1 (Computer Integrated Manufacturing) classique, qui de par l'émergence des technologies de communication (industrielles notamment) donne lieu à une organisation de fonctions autour de réseaux (voir figure 11).

Ces réseaux ont pour objectif de faire dialoguer de nombreux équipements selon trois niveaux hiérarchiques (classiques au sein de la pyramide CIM).

Au plus haut, niveau 3, ils relient les calculateurs de gestion et les systèmes de contrôle-commande temps réel (classiquement appelé *réseau d'usine*).

Au niveau 2, celui de l'atelier, ils relient les systèmes de contrôle-commande et les cellules de production (classiquement appelé *réseau de cellule*).

Au niveau 1, les *réseaux de terrain* font communiquer les capteurs, les actionneurs et les automates programmables. Ces trois réseaux requièrent une sécurité de fonctionnement supérieure à celle des réseaux informatiques utilisés dans le cadre de la bureautique. Ils doivent tolérer les pannes tout en assurant l'acheminement des données avec un haut degré de fiabilité (voir figure 11). [7]

1.8.2 Critères de choix d'un réseau (RLI)

- Standardisation, normalisation
- Interopérabilité, compatibilité
- Complexité mise en œuvre, maintenance
- Coût
- Performances
- Respect des contraintes temporelles, fiabilité. [4]

1.8.3 Hiérarchie d'une Entreprise de fabrication:

L'émergence de ces nouvelles technologies à conduit à la définition de plusieurs catégories de réseaux locaux industriels (pyramide CIM) :

- les réseaux de terrain,
- les réseaux de cellule,
- les réseaux de supervision et de commande.

Aujourd'hui le système commande-contrôle est conçu sous le modèle CIM (Computer Integrated Manufacturing)

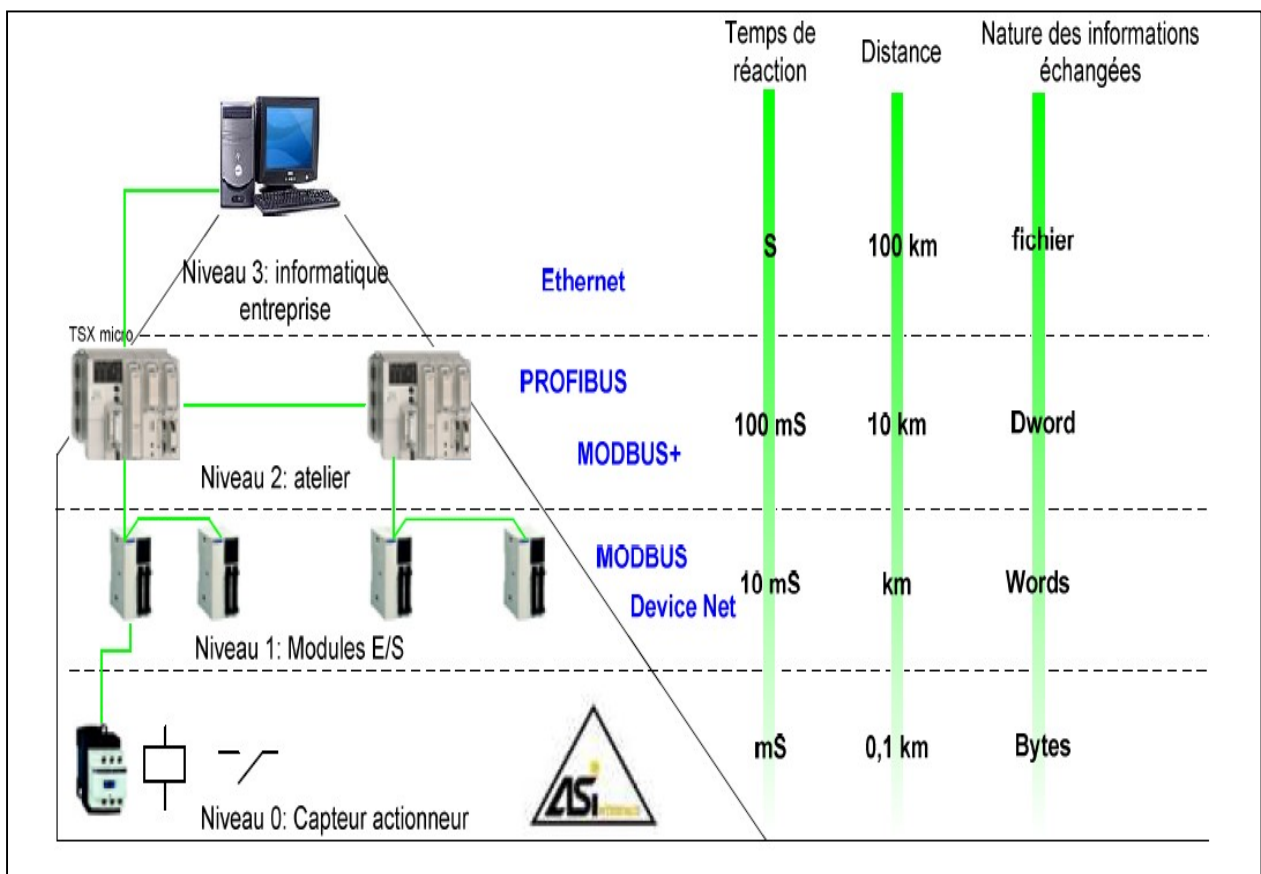


Figure 1.11 : Hiérarchie d'une Entreprise de fabrication (pyramide CIM).

Le réseau Ethernet pénètre de plus en plus la structure d'automatisme : automate, variateur et bientôt capteurs.

Le CIM décrit les différents niveaux de communication sous une forme quantitative des données à véhiculer. Le niveau 0, niveau capteur/actionneur,

nécessite un transfert performant (quelques millisecondes) mais concernant peu d'informations (données binaires), alors que le niveau 4 nécessite quant à lui de véhiculer de gros paquets de données, des fichiers et la performance n'est plus forcément un critère prédominant (voir figure 11).

Le CIM ne résout donc pas la problématique de la gestion de l'augmentation du trafic sur les médiums.

Les offreurs de communication adaptent les performances de leurs réseaux en fonction des niveaux du CIM sur lesquels ceux-ci seront positionnés. De ce fait, la quasi-totalité des réseaux d'automatismes sont dits propriétaires et répondent essentiellement en la mise en conformité d'une approche produite et non pas exactement de la problématique utilisateur qui est son besoin.

Le CIM a contribué à la segmentation de l'automatisme en niveaux :

- Capteur/Actionneur (niveau 0),
- Automatisme (niveau 1),
- Supervision (niveau 2),
- Informatique (niveaux 3 et 4) en associant le couple volume/performance.

1.8.4 Les réseaux de communication en automatisme industriel

a Bus capteurs et actionneurs

❖ *Bus AS-i*

AS-i (Actuator Sensor interface) est un bus de capteurs et d'actionneurs. C'est un bus déterministe aux temps de réponse très courts qui s'appuie sur un standard industriel ouvert soutenu par l'association AS-i. Cette association compte parmi ses membres les leaders du marché des capteurs, des actionneurs, des automates programmables et de la connectique.

AS-i offre donc l'avantage de ne pas être un réseau propriétaire. Son raccordement vers le niveau supérieur dans la hiérarchie des réseaux peut être réalisé au travers des passerelles (comme la passerelle FiPio/AS-i) ou en utilisant les capacités de communication d'un coupleur de bus (automate...).

La topologie d'AS-i est libre: point à point, en ligne ou en arbre en anneau.

❖ *Bus CAN-open*

Le bus CAN-open s'appuie sur la technologie CAN (Controller Area Network), développée à l'origine pour les systèmes embarqués des véhicules automobiles ; la technologie CAN est maintenant utilisée dans de nombreux domaines comme : le transport, les équipements mobiles, les équipements médicaux...

Le bus CAN-open s'impose désormais dans les automatismes industriels et en particulier sur les machines. Il utilise une double paire torsadée blindée sur laquelle le raccordement d'un maximum de 127 équipements s'effectue par simple dérivation. Le débit binaire variable entre 10 Kbits/s et 1 Mbits/s est conditionné par la longueur du bus.

b Bus de terrain

❖ *Bus Fip*

Sur la base du standard World-Fip, deux profils ont été développés :

- Fipio, bus d'E/S de l'automate (niveau 1),
- Fipway, bus de synchronisation (niveau 2).

Le bus de terrain Fipio est un standard de communication entre différents constituants d'automatisme (niveau 1). Il permet la connexion de 127 équipements à partir du point de connexion intégré au processeur. Ce bus de terrain est basé sur les mécanismes producteurs/consommateurs. Il est destiné au déport d'E/S jusqu'à 15 km et accueille des équipements tiers.

c Réseaux locaux industriels (RLI)

❖ *Réseau Fipway*

Le réseau Fipway est un réseau local industriel assurant la communication entre les différents automates programmables Micro, Premium et TSX Série 7. Il sert de bus de synchronisation entre automates ; il est dérivé de la norme FIP et inclut de nombreux services complémentaires.

❖ *Bus Modbus ET Jbus*

Le bus Modbus (le bus J-bus est une variante de Modbus) répond aux architectures Maître/Esclave. Il a été créé par la société MODICON en 1978 pour interconnecter les automates programmables.

Le bus est composé d'une station Maître et de stations Esclaves. Seule la station Maître peut être à l'initiative de l'échange (la communication directe entre stations Esclaves n'est pas réalisable).

❖ *Réseau Modbus Plus*

Le réseau Modbus plus est un réseau local industriel, déterministe et performant, répondant à des architectures étendues de type Client/serveur, combinant haut débit (1 Mbit/s), supports de transmission simples et économiques ainsi que de nombreux services de messagerie. 64 nœuds peuvent être connectés sur 1800 m (32 nœuds sur 450 m sans répéteur) sur paire torsadée.

d Réseaux Ethernet TCP/IP

Ethernet sans protocole de communication n'est pas un réseau mais un médium normé. Il concerne les couches Liaison (2) et Physique (1) du modèle OSI. Bien qu'il existe une différence entre Ethernet et IEEE 802.3, par abus de langage, Ethernet est assimilé à la norme 802.3. Il est à noter qu'Ethernet TCP/IP est un réseau non déterministe.

Les principales caractéristiques de configuration d'Ethernet sont :

- un tronçon (ou segment) principal Ethernet ne peut excéder 500 m,
- un tronçon ne peut pas accepter plus de 100 nœuds (par nœud, on entend toute entité unique adressable sur Ethernet),

La distance minimale entre deux nœuds devant être supérieure à 2,5 m.

Le câble est généralement marqué tous les 2,5 m.

- le chemin entre 2 nœuds distants ne doit pas :
 - Traverser plus de 2 répéteurs (ou 4 demi-répéteurs ou 1 répéteur et 2 demi-répéteurs),
 - Présenter plus de 1500 m de câble coaxial,
 - Présenter plus de 1000 m de liaisons point à point (entre demi-répéteurs présents),
- le câble de transceiver (ou dérivation) ne peut excéder 50 m (liaison transceiver vers coupleur station, vers répéteur, vers demi-répéteur),
- la distance théorique maximale entre deux stations est de 2,8 km.

Toutes ces limitations sont liées à la technique CSMA/CD (Carrier Sense Multiple Access with Collision Detection) et à la vitesse de propagation des signaux électriques sur les médias ou dans les constituants, ainsi qu'aux phénomènes de réflexion.

Il existe une multitude de protocoles associables à Ethernet. Deux sont devenus des standards, à savoir : TCP et IP.

❖ IP (Internet Protocol)

IP est le protocole principal de la couche réseau qui est utilisé à la fois par TCP, UDP, ICMP et IGMP.

Une application peut également accéder directement à IP (rare mais possible).

Chaque bloc de données qui circulent sur l'Internet traverse la couche IP de tous les hôtes en extrémités réseau ou routeurs intermédiaires. Il assure le routage des messages qui est direct si le destinataire est sur le même réseau ou indirect via routeur ou passerelle. Le service est non fiable, il n'existe aucune garantie que le datagramme arrive à destination. Il fournit un service qualifié de "au moindre effort" ou "au mieux". En cas de saturation de buffers, IP ne sait que rejeter un datagramme et essayer de prévenir l'émetteur via un message ICMP.

La fiabilité doit être assurée par les couches supérieures.

❖ TCP (Transmission Control Protocol)

TCP est destiné à être implémenté sur la couche transport du modèle OSI. C'est un protocole de transport fiable orienté connexion et flux de données. TCP se charge de traiter la non-fiabilité d'IP.

L'offre Ethernet TCP/IP implémente sur la couche application les protocoles applicatifs natifs tels que Modbus.

Ceci afin de satisfaire aux besoins de dialogue d'application à application et ainsi assurer l'interopérabilité des différentes plates-formes automates.

Ceux-ci seront distribués sur le réseau Ethernet TCP/IP pour ainsi en faire bénéficier la totalité des acteurs. Diag viewer est accessible par un navigateur Internet.

Ces fonctions sont fournies avec les coupleurs concernés et ne nécessitent aucun développement.

De plus, certains coupleurs disposent d'une mémoire utilisateur permettant le chargement de pages HTML et ainsi autoriser la conception d'un dialogue homme/machine temps réel embarqué.[5]

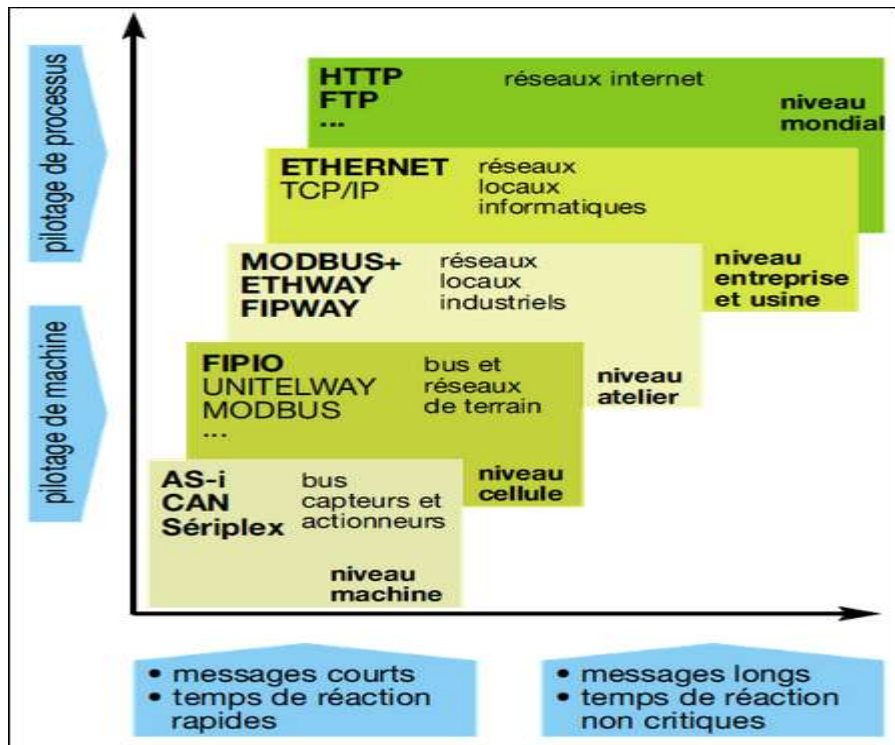


Figure 1.12 : Positionnement des principaux Réseaux et Bus.

1.9 Evolution de la communication, les grandes tendances

Avec l'adoption progressive de standards communs par les deux mondes, la frontière entre automatisme et informatique tombe. La communication entre ces deux mondes converge grâce à l'adoption de protocoles standard mondiaux Ethernet et TCP/IP, mais aussi grâce à la prise en compte de mécanismes normés tels qu'OPC (Open Platform Communications). Ces nouvelles technologies, associées aux Extranet, Intranet et Internet, autorisent un accès aux données de l'automatisme en temps réel, en tout lieu, à toute personne autorisée. Cette convergence est renforcée par l'arrivée de nombreux constituants d'automatismes basés sur cette technologie. Le concept "Transparent Factory" s'appuie sur toutes ces nouvelles technologies de l'information et de la communication (NTIC).[5]

1.9.1 L'impact des NTIC sur les architectures d'automatismes

La dernière décennie a vu l'apparition des nouveaux langages de programmation et des mécanismes d'échanges de données dans le monde

informatique. Depuis dix ans, Ethernet TCP/IP s'est développé sur les architectures d'automatismes et est devenu un réseau fédérateur à partir du niveau 2 du CIM.

La généralisation de ce réseau autorise une prise en compte simple de nouveaux outils de conduite et d'exploitation des données : MES (Manufacturing Execution System) et ERP (Enterprise Resource Planning). Parallèlement à l'apparition du MES, les postes de supervision ont tendance à se repositionner au niveau de la conduite.

Globalement, les NTIC ont tendance à aplanir la pyramide du CIM, c'est à dire à fusionner certains niveaux, car il n'est plus utile de décrire les niveaux sous une approche quantitative mais fonctionnelle.[5]

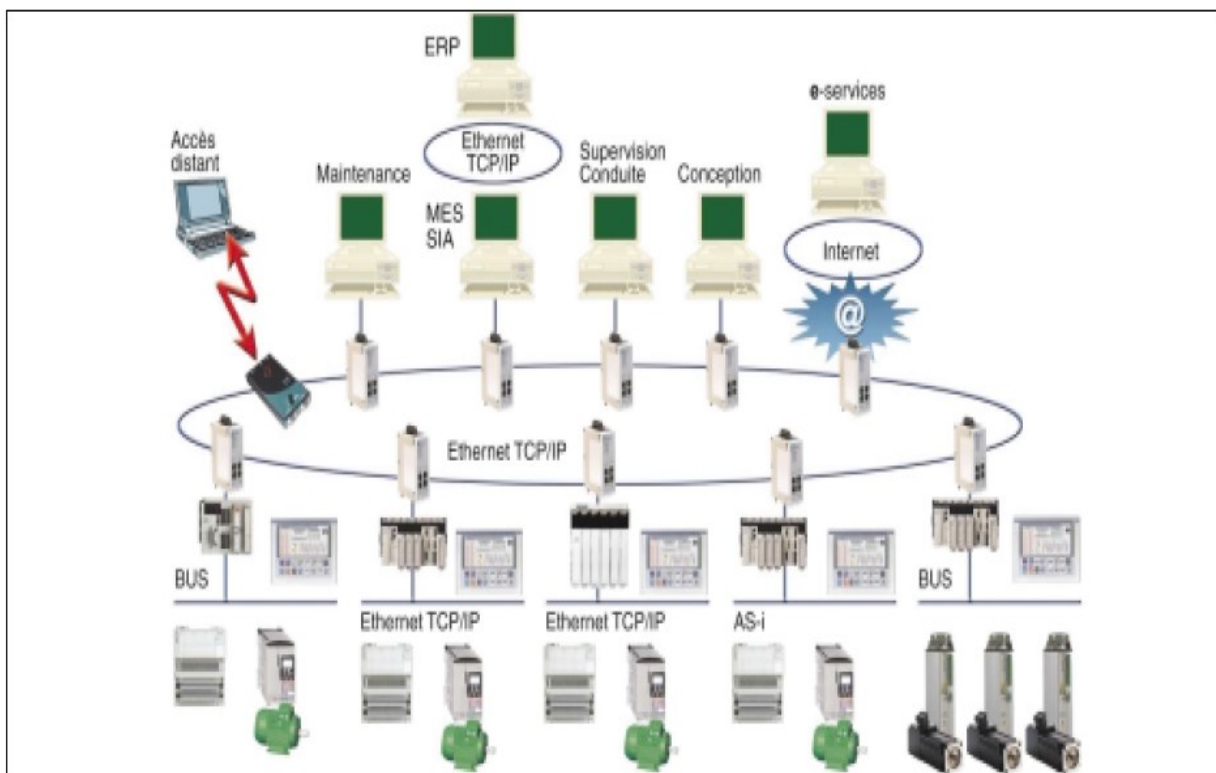


Figure 1.13 : L'impact des NTIC sur les architectures d'automatismes.

1.10 Le modèle OSI (Open System Interconnections)

Le modèle OSI est un modèle académique constitué de sept couches à savoir : couche 7 (Application), couche 6 (Présentation), couche 5 (Session),

couche 4 (Transport), couche 3(Réseau), couche 2 (Liaison), et couche 1 (Physique) (figure 1.14).

Les couches Transport, Session et Présentation peuvent être vides. C'est ce qu'on nomme un profil simplifié, largement utilisé pour les réseaux industriels.

Fonctionnellement, l'ensemble des sept couches (figure 14) est divisible en deux familles bien distinctes : la première, constituée des couches 1 à 4, offre les services de communication (transfert des données), la seconde constituée des couches 5 à 7, les services d'application (utilisation des données). [5]

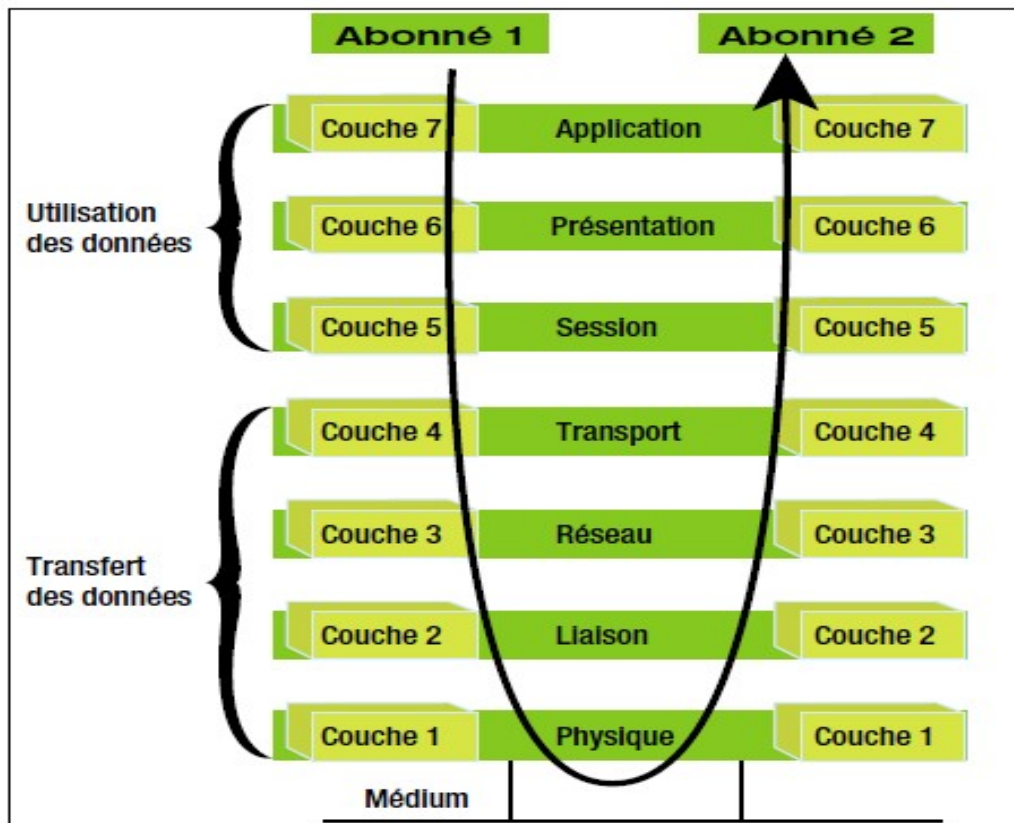


Figure 1.14 : Le modèle OSI. [5]

1.10.1 L'encapsulation

C'est un mécanisme de transmission de données. Lorsqu'une application envoie des données vers le réseau, chaque couche traversée exécute un processus

d'encapsulation de l'unité de données fournie par la couche supérieure en ajoutant un entête, voire une remorque, qui lui est propre.

Réciproquement à la réception, chaque couche exécutera une dés-encapsulation pour en final ne restituer que les données utilisateurs à l'application. La figure 14 matérialise le processus d'encapsulation. [5]

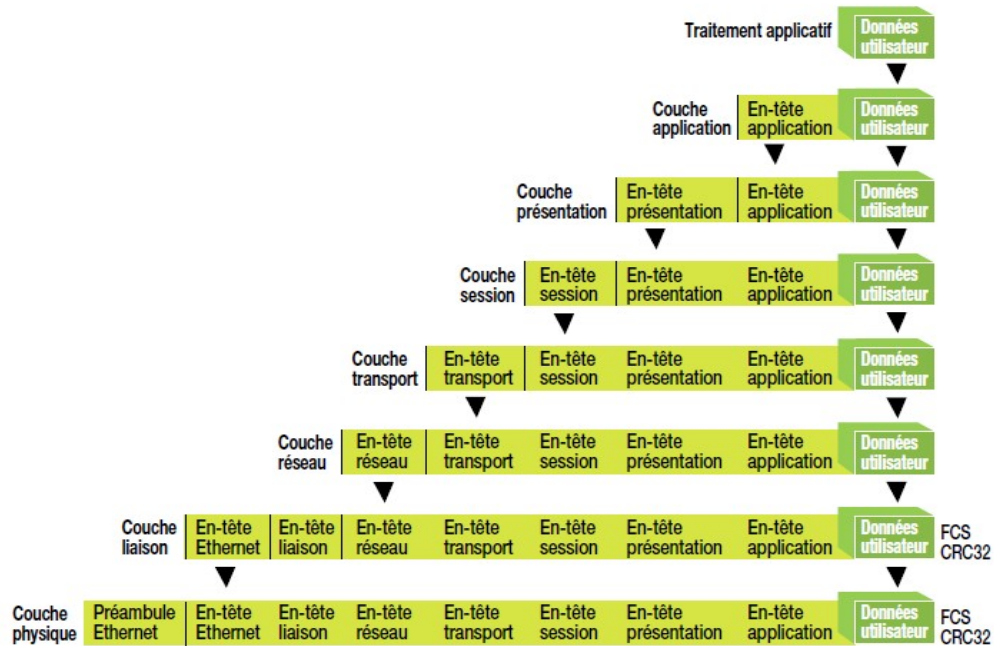


Figure 1.15 : Mécanisme d'encapsulation. [5]

1.11 Topologies des réseaux

La topologie d'un réseau est caractérisée par le système de câblage du réseau; il existe deux niveaux de classification : suivant la taille et selon le type.

1.11.1 Classification selon la taille

Les réseaux sont divisés en trois grandes familles : les LAN, MAN et WAN.

- LAN = Local Area Network ou réseau local.

Ce type de réseau s'étend de 1 mètre à 2 kilomètres et peut compter de 2 à 200 abonnés. Le débit courant est de 1 à 100 Mbits/s.

- MAN = Metropolitan Area Network ou réseau métropolitain.

Ce type de réseau s'étend de 1 mètre à 100 kilomètres et peut compter de 2 à 1000 abonnés. Le débit courant est de 1 à 100 Mbits/s.

- WAN = Wide Area Network ou réseau grande distance.

Ce type de réseau s'étend sur plus de 1000 kilomètres et peut compter plusieurs milliers d'abonnés. Le débit, étant donné la distance à parcourir, est plus faible, de 50 bits/s à 2 Mbits/s. Les deux premiers types de réseaux utilisent des connexions multipoints tandis que le WAN utilise des connexions point à point. Par exemple, le réseau Transpac, réseau français est constitué de liaisons point à point entre des nœuds situés dans les grandes villes françaises.[5]

1.11.2 Classification suivant le type

c'est la partie physique du réseau :

- ***Point à point***

C'est la forme la plus élémentaire qui implique deux machines. Elle correspondra souvent à la topologie d'une partie d'un réseau. En général, cela concerne une liaison série dont les vitesses sont vite limitées par la distance, l'utilisation de la fibre optique pouvant compenser cette faiblesse.[5]

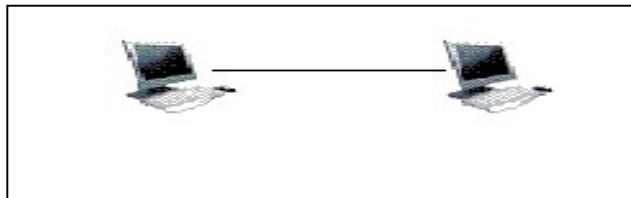


Figure 1.16 : Topologie point à point.

- **Etoile**

Une approche ancienne, non normalisée, correspondait à une organisation de machines esclaves reliées par liaisons point à point à une machine maîtresse.[5]

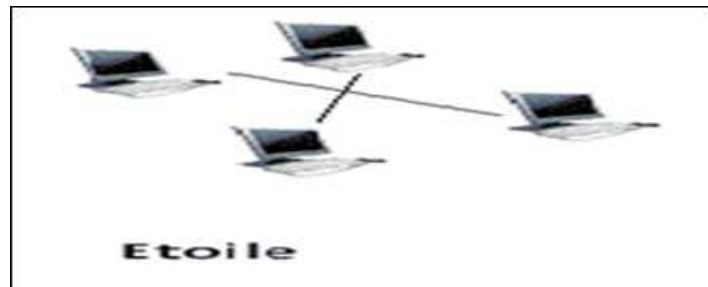


Figure 1.17 : Topologie en étoile.

- **Bus**

C'est la topologie la plus commune aux LAN (Local Access Network) car la plus économique. Chaque nœud est raccordé au bus par l'intermédiaire de modules de dérivation actifs ou passifs suivant le protocole de réseau.[5]

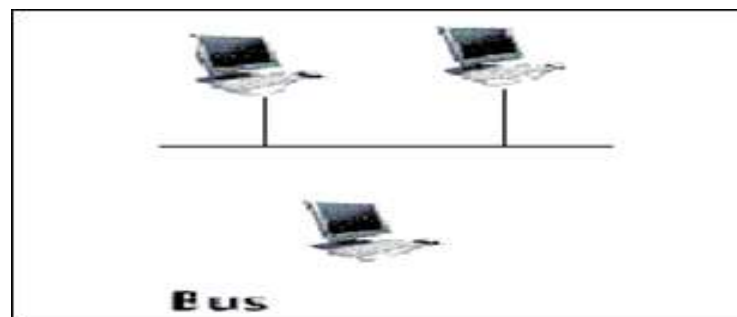


Figure 1.18 : Topologie en Bus.

- **Arbre**

Chaque nœud peut être un abonné ou un hub. C'est une variante de la topologie en étoile avec les mêmes faiblesses. Cette architecture revient en force avec les techniques du type Ethernet en fibre optique 10 bases F et Ethernet commuté du type 100 bases VG. [5]



Figure 1.19 : topologie en arbre.

- **Maille**

C'est la forme classique d'organisation WAN (World Access Network). Les équipements sont reliés entre eux pour former une toile (Web) d'araignée (comme Internet). Chaque nœud a un rôle de routeur. Pour atteindre un nœud, les chemins sont multiples et choisis en fonction de critères comme la disponibilité d'un nœud ou d'un WAN, la qualité de transmission ou la charge ponctuelle sur un tronçon WAN donné.[5]

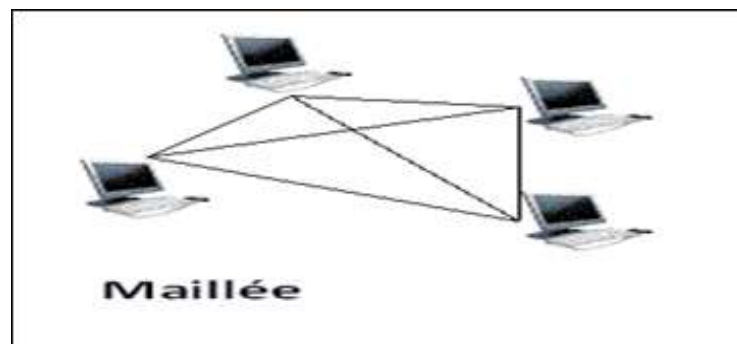


Figure 1.20 : Topologie en maille.

- **Anneau**

Chaque nœud est relié à ses voisins pour former une boucle fermée et a un rôle actif dans la propagation des échanges. Cette structure est bien adaptée aux LAN, en particulier si l'on recherche la disponibilité.

Une rupture de liens entre deux nœuds peut être gérée pour garantir la communication. Chaque nœud a la possibilité de régénérer le signal et la structure

se prête facilement à l'utilisation de la fibre optique, les distances de couverture pouvant être grandes. [5]

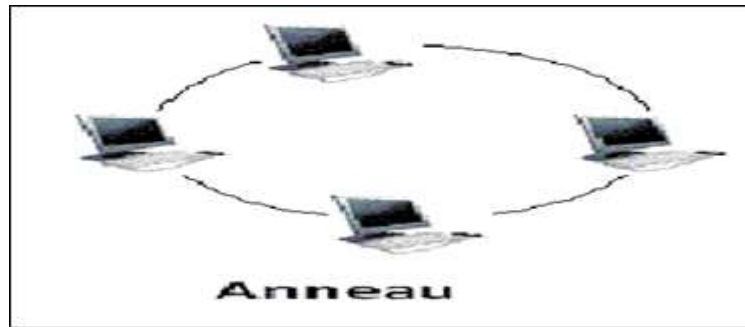


Figure 1.21 : Topologie en anneau.

1.11.3 Différents types d'interconnexion de réseaux

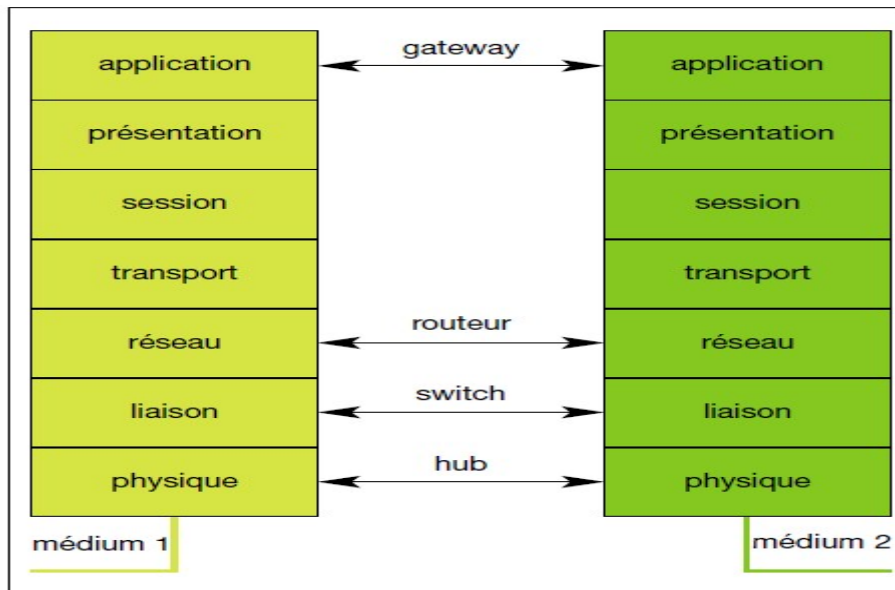


Figure 1.22 : Différents types d'interconnexions de réseaux. [5]

a Gateway

Unité fonctionnelle qui permet l'interconnexion de deux réseaux d'architecture différente.

b *Routeur*

Il crée une segmentation logique de réseaux. Il assure le passage de l'information entre deux sous-réseaux logiques distincts en choisissant le meilleur chemin. C'est la couche réseau qui assure ce routage. Il n'est pas transparent, il faut donc l'adresser pour le traverser.

c *Switch*

Il transmet les données reçues sur un port, seulement vers le port sur lequel la station destinatrice est connectée. Il assure la prolongation du support au delà des limites en distance du standard (segment) en réalisant une remise en forme des signaux. Il supprime les collisions et les paquets non valides et réduit la charge moyenne sur le réseau entier. Synonyme : Bridge.

d *Hub*

Les données reçues sur un port sont envoyées à tous les autres ports. Le hub ne possède pas de mémoire interne et diffuse les collisions ; plus il y a d'équipements, plus il y a de collisions et plus la charge est importante.

Pour de plus longues distances, utiliser des Switch (répéteur). [5]

1.12 Systèmes de contrôle commande

Les systèmes de contrôle commande sont divers et différents dans l'industrie. La modernisation remarquable de ces systèmes laisse apparaître de nouvelles technologies de commande qui permettent de mieux gérer les process industriels devenus de plus en plus complexes. [8]

Parmi les systèmes de contrôle commande les plus courants, et s'intégrant dans la logique programmée, on cite :

1.12.1 Le système de contrôle commande par PLC

Ce type de système de contrôle commande est basique et simple d'utilisation. Le PLC (Programmable Logic Controller), qui représente le cerveau de la commande, est programmé en tenant compte des entrées logiques et analogiques qu'il reçoit via ses modules d'entrées.

Après exécution du programme implémenté dedans, il reçoit les commandes adéquates via les modules de sorties vers les différents actionneurs et pré-actionneurs équipant les machines à piloter. [8]

Son inconvénient majeur est l'absence d'une interface de supervision permettant un contrôle visuel par l'opérateur dans la salle de contrôle du processus industriel.

1.12.2 Le système de contrôle, commande et supervision SCADA

Ce type de logique programmé est basé sur des PLCs, la supervision SCADA (Supervisory Control And Data Acquisition) est une solution très performante pour la commande des systèmes industriels complexes.

Le poste opérateur intègre une interface utilisateur permettant à l'opérateur de superviser la machine à partir d'un tableau de bord virtuel comportant des boutons, des voyants, des alertes et toutes les données dont il a besoin pour la prise de décision. L'ensemble des couples PLC/HMI (Human Machine Interface) forme ce qu'on appelle le SCADA.

Bien entendu, le SCADA peut comporter plusieurs PLCs qui sont extensibles en plusieurs modules d'entrées/sorties. Il présente une souplesse et une adaptabilité dans son installation puisque les fonctions logiques sont toutes rassemblées en un seul programme qui peut être aisément modifié. [8]

1.12.3 Le système de contrôle, commande et supervision DCS

Le système de contrôle commande DCS (Distributed Control System), développé au début des années 70, ressemble en grande partie au SCADA. Sauf que ce dernier est destiné pour gérer des processus plus étendus et plus complexes.

La principale différence entre un DCS et un SCADA réside essentiellement dans la nature de l'architecture et la criticité du procès supervisé. En effet le DCS gère beaucoup plus rapidement le transfert de données et se distingue par un temps de réponse remarquable.

Le DCS présente une architecture très organisée qui empêche toutes sortes de conflits et de collisions de données.

Ses contrôleurs sont reliés entre eux via un réseau Profinet pour échanger des données partagées. Chaque contrôleur est doté de modules d'entrées/sorties qui lui sont propres via lesquels il communique avec le système. [8]

1.13 L'utilisation des tablettes électroniques en automatisation industrielle

Les développements technologiques majeurs sont souvent initiés par l'industrie militaire, aérospatiale ou industrielle.

Or, il arrive parfois qu'un produit créé et commercialisé spécifiquement pour le domaine public soit utilisé par l'industrie développée à priori pour l'usage personnel, la tablette électronique en est un bon exemple. Ce produit offre des possibilités extrêmement intéressantes dans le domaine industriel, en particulier au niveau de l'automatisation. Plusieurs fabricants offrent une gamme croissante de tablettes opérant sur des systèmes d'exploitation standards ou propriétaires. [9]

1.13.1 Visualisation et contrôle des opérations

La visualisation et le contrôle d'un procédé ou d'une ligne de production, à distance ou localement, sont sans doute parmi les avantages les plus incontestables de la tablette électronique. Avec sa tablette dans la main, un opérateur, un technicien d'entretien ou un ingénieur de procédé peut se déplacer le long de la ligne de production et visualiser, en temps réel, l'état d'opération d'une machine. Cela peut se faire à travers les différentes pages d'une interface opérateur existante, d'un serveur WEB intégré au système de contrôle, ou tout simplement d'une

application embarquée. En réalité, toute opération qui peut être effectuée sur une interface opérateur standard, comme démarrer un moteur, modifier un paramètre ou accepter une alarme, devient possible à partir de la tablette électronique. Il est également facile de créer des pages spécialement dédiées à une utilisation depuis cette dernière. Par exemple, une console d'opération physique, composée de boutons de commande, de sélecteurs et de témoins lumineux, peut être recrée virtuellement sur une interface opérateur et utilisée à distance à partir de la tablette électronique. Cela offre la possibilité à un opérateur ou à un technicien d'opérer une machine à partir d'un endroit plus approprié lors de travaux de maintenance, d'analyse, de dépannage ou d'optimisation. [9]



Figure 1.23 : Les différents types des tablettes industriels.

1.13.2 Calibration et ajustements

Une seconde utilisation très intéressante de la tablette électronique sur une ligne de production est le paramétrage et l'ajustement des différents instruments. Dans certains cas, un contact physique avec les composantes de l'équipement est requis. De façon générale, les ajustements sont effectués à partir de la console d'opération, souvent située à une certaine distance de l'équipement en question.

Avec la tablette électronique, il devient facile de se déplacer à l'endroit le plus approprié pour procéder à la calibration ou aux ajustements.

Il est probable que les manufacturiers d'équipement d'instrumentation offriront sous peu des applications spécifiques pour leurs instruments. Celles-ci seront exécutables directement sur les tablettes électroniques courantes. Avec le multimètre et le tournevis, cette dernière pourrait bientôt faire partie de la boîte à outils standard du technicien en instrumentation. [9]

- Visualisation des alarmes.
- Accès aux contrôleurs.
- Visualisation des dessins techniques.
- Manuels techniques.
- Enregistrement et visualisation de séquences vidéo.
- Support à distance.
- Communiquer partout et en tout temps.

1.14 Conclusion

Dans ce chapitre, nous avons essayé de donner une idée générale sur l'automatisation industrielle, l'historique de l'industrie, l'informatique dans l'industrie et les différents réseaux de communication dans les différents niveaux d'automatisation ainsi que les topologies et les améliorateurs de communication. Enfin l'utilisation des tablettes électroniques dans l'industrie ; ce tour d'horizon nous a permis de présenter l'application qui fera l'objet de ce travail.

Dans le chapitre suivant, nous commencerons tous d'abord par la description des techniques industrielles de suivi et de pilotage informatique de procédés de fabrication automatisés.

Chapitre 2 Techniques industrielles de contrôle et pilotage de procédés de fabrication

2.1 Introduction

Dans ce chapitre nous allons traiter les APIs, leurs apports dans le secteur industriel, la façon de les programmer, de les mettre en réseau, et les logiciels mis en œuvre pour leur configuration et programmation, nous allons aussi évoquer dans ce même chapitre la supervision des procédés industriels avec toutes ces formes, à savoir : l'IHM, le SCADA, et le DCS.

2.2 Les Automates Programmables Industriels (API)

Il existe beaucoup d'API qui ont chacune leur particularité de programmation. Les plus "simples" ont un mode de programmation pas à pas et un déroulement du programme de manière séquentielle (il faut attendre que la condition soit remplie pour que le programme passe à l'instruction suivante). Nous ne traiterons pas de ce genre d'API. La gamme suivante se programme en codes à l'aide d'un ordinateur ou d'un programmeur. Ceux qui acceptent la programmation par ordinateur ont, en principe, des logiciels qui permettent une programmation facilitée, donc une lecture, une modification, un dépannage simplifié (LADDER diagramme-symbolique américaine, diagramme des flux, portes logiques, etc.).

Quel que soit le logiciel utilisé le mode de fonctionnement de l'API est le même. A partir d'un schéma représentant une commande le programmeur traduit ce schéma en un programme mis dans des mémoires. Ce programme lisible par un microprocesseur contient différents éléments:

- un pas de programme, (une ligne numérotée)
- une instruction ou code (par exemple: lire l'état d'une entrée, comparer deux valeurs, etc....)
- une référence (par exemple: numéro d'une sortie).

Toute "l'intelligence" de l'automate réside dans son CPU (central processing unit) en français c'est l'unité centrale. Elle se compose des éléments suivant (voir figure 2.1) :

- un microprocesseur (traitement des informations)
- une horloge interne (fonction de temporisation, synchronisation des opérations)
- unité de calcul (addition binaire, compteur, décompteur)
- plusieurs entrées (réception des données externes)
- plusieurs sortie (émission des données vers l'extérieur, - pour relais, écran, imprimante)
- une mémoire ROM ou EPROM (contient le système d'exploitation)
- une mémoire RAM divisée en zone.
- programme utilisateur (code binaire, assembleur)
- données utile au programme.
- données utile au programmeur (programme de base, registres, drapeaux, constantes).
- bus de connexion pour des modules déportés.

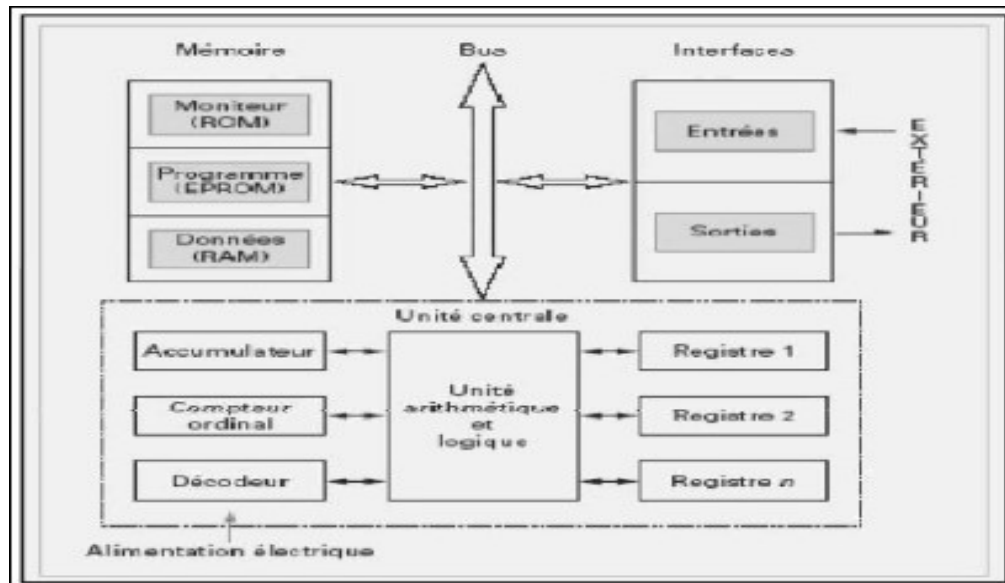


Figure 2.1 : CPU d'un automate.

L'unité centrale n'est pas à même de supporter des intensités de courant telles qu'il y en a au travers des bornes d'entrées et de sorties de l'API. Elles sont donc protégées des surtensions éventuelles ou de ces courants trop intenses. Pour séparer électriquement le CPU du reste de l'API, il est souvent utilisé des photocoupleurs.

Lorsqu'on fait un programme, on peut choisir un langage évolué (LADDER, GRAFCET, Logigramme). Les API ne comprenant que les langages en liste d'instructions, il faut compiler le programme (si le logiciel de programmation ne le fait pas automatiquement) avant de la transférer dans l'API.

On cherche également à documenter (joindre des commentaires) au maximum le programme pour qu'une autre personne puisse le comprendre et le modifier facilement. On peut également déclarer les entrées, sorties, etc..... Par des noms courts et/ou des textes plus longs - par exemple : sur la sortie O : 000.01 est raccordé le relais K1 sur lequel commande la ventilation : O : 000.01 - K1 - ventilation 1. Toutes ces adjonctions au programme de base ne sont pas prises en charge par l'API.

L'ordinateur va créer pour ces données des fichiers spécifiques qui resteront dans la mémoire d'archivage (disque dur). Sans ces fichiers, un rappel du programme

de l'API vers l'ordinateur ne donnera que la liste d'instructions et le reste des données peut être considéré comme perdu.

Les figures ci-dessous montre quelques APIs vendu par des constructeur de renommé mondiale.



Automate compact (Allen-bradley)



Automate modulaire (Modicon)



Automate Modulaire (OMRON)



API modulaire SIEMENS

Figure 2.2 : Quelques APIs vendus par des constructeurs de renommé mondiale.

Pour la suite de notre application, nous avons choisi le constructeur OMRON très connu par la robustesse de son matériel.

Ayant donné dans ce qui a précédé une brève description sur les APIs, nous allons traiter dans ce qui suit l'intérêt de la supervision des procédés de fabrication, soit localement c'est-à-dire au pied de la machine, on parle dans ce cas des Interfaces Homme Machine (IHM), soit globalement pour la supervision d'un nombre important de machines au d'ateliers, et on parle dans ce cas des systèmes SCADA ou DCS.

2.3 La supervision et son intérêt

2.3.1 Définition

La supervision concerne l'acquisition de données (mesures, alarmes, retour d'état de fonctionnement) et des paramètres de commande/réglages des processus généralement confiés à des APIs, régulateurs et des fédérateurs.

A l'origine, la supervision est une technique industrielle de suivi et de pilotage informatique de procédés de fabrication automatisés.

Aujourd'hui, elle est utilisée dans tout type de domaine pour le suivi d'un ensemble de sous équipements depuis un poste informatique.

La supervision dans le secteur industriel peut être divisée en deux catégories selon son étendue, en effet, si le superviseur surveille une machine ou un nombre très réduit de machines on parle dans ce cas d'IHM (Interface Homme Machine), si le nombre de machine a contrôlé augmente, le flux d'informations devient vite très important et insupportable pour les IHMs. Pour pouvoir traiter ce flux en temps réel et préserver la sécurité des biens et des personnes, les constructeurs des APIs se sont vite penchés vers une solutions qui mis en œuvre la mise en réseau des différents constituants des moyens de production et de décentraliser le traitement.[11]

2.3.2 L'intérêt de la supervision

Comme il a été évoqué précédemment, la supervision permet de surveiller, en termes de disponibilité et de performance l'ensemble de l'infrastructure d'une entreprise. Elle assure la surveillance du bon fonctionnement du système à savoir [11] :

- ◆ Fournir une représentation synthétique, dynamique et instantanée de l'ensemble des moyens de production de l'unité.
- ◆ Permettre à l'opérateur d'interagir avec le processus et de visualiser le comportement. normal, les courbes donnent une représentation graphique de différentes données du processus.

- ◆ Permettre la sauvegarde périodique de grandeurs (archivage au fil de l'eau).
- ◆ Permettre la sauvegarde d'événements horodatés (archivage sélectif).
- ◆ Fournir les outils de recherche dans les données archivées.
- ◆ Fournir la possibilité de refaire fonctionner le synoptique avec les données archivées (Fonction de magnétoscope ou de replay).
- ◆ Permettre de garder une trace validée de données critiques (traçabilité de données de production).
- ◆ Gestion des Alarmes.

De nombreuses techniques de supervision sont possibles du plus bas niveau à la plus intégrée et professionnelle, où chaque technique a ces caractéristiques.

2.3.3 L'IHM (Interface Homme-Machine)

Une interface homme-machine (IHM) est un dispositif informatique qui présente les processus à l'opérateur humain et lui permet de contrôler le processus. Les IHM sont généralement reliés à la base de données du système SCADA et à des programmes capables de calculer des tendances, sélectionner des données de diagnostic et des informations de gestion telles que les procédures d'entretien prévisionnels. Des schémas détaillés d'un capteur ou d'une machine particulière, et d'un guide de dépannage. Les IHM présentent des informations graphiques aux opérateurs sous une forme synoptique. Ainsi l'opérateur peut voir une représentation schématique de la chaîne de production ou de l'usine qu'il contrôle. Par exemple le dessin d'une pompe connectée à une canalisation peut montrer à l'opérateur que la pompe fonctionne et quel est le débit instantané de fluide s'écoulant dans le tuyau. Si l'opérateur arrête la pompe, l'IHM lui montrera la diminution du débit de la pompe en temps réel (voir figure 2.3).

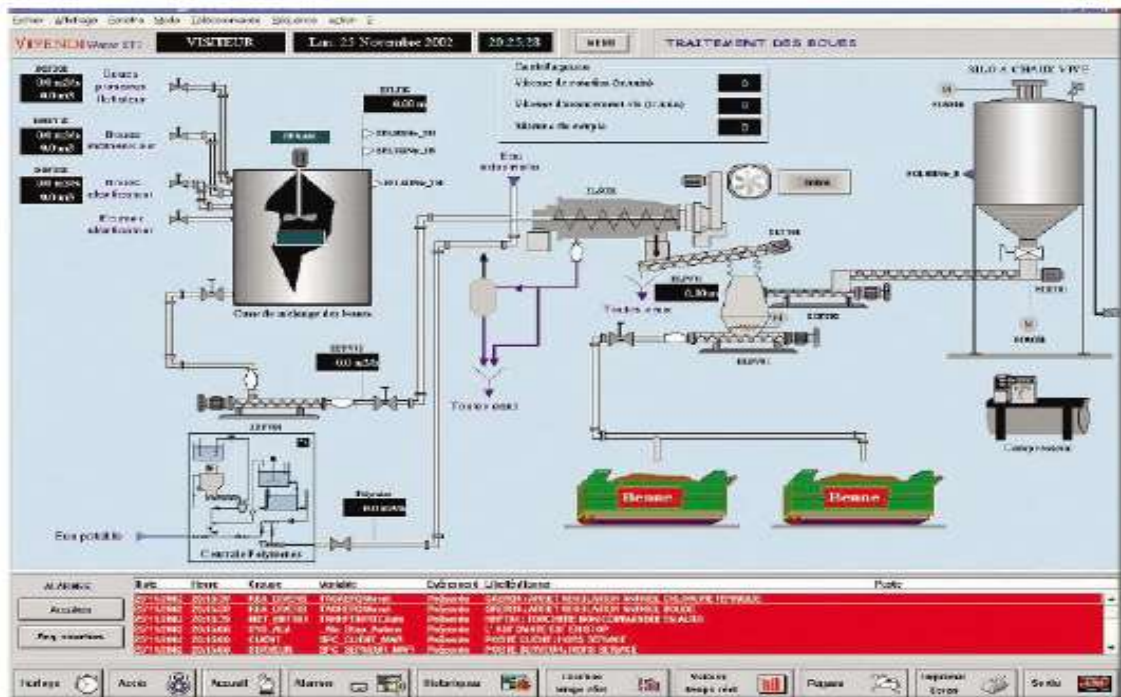


Figure 2.3 : Exemple d'un synoptique d'un process de gestion de traitement de boue.

L'IHM est généralement livrée avec un programme de conception graphique permettant aux opérateurs ou au personnel de maintenance de modifier la représentation des composants. Les objets schématisés, peuvent être aussi élémentaires, qu'un indicateur lumineux sur l'écran représentant l'état d'un feu de circulation sur le terrain, qu'un affichage à plusieurs multi-projecteurs représentant la position de tous les ascenseurs d'un gratte-ciel ou des trains dans une gare ferroviaire.



Figure 2.4. Un IHM industriel.

La gestion des alarmes est une partie essentielle de la plupart des IHM et des systèmes SCADA (voir figure 2.5). Le système contrôle si certaines conditions d'alarme sont réunies, afin de déterminer si un événement alarmant s'est produit. Si un événement d'alarme est ainsi détecté, des mesures réactives sont prises comme l'activation de signaux ou indicateurs d'alarme, l'envoi de messages SMS ou de courriels, ... afin d'alerter les opérateurs SCADA distants ou la direction. Dans de nombreux cas, un opérateur IHM ou SCADA peut avoir à valider (ou acquitter) l'événement d'alarme, ce qui peut désactiver certains indicateurs d'alarme, alors que d'autres indicateurs restent actifs jusqu'à ce que les conditions d'alarme sont réinitialisées. Les conditions d'alarme peuvent être explicites, par exemple, un élément d'alarme peut être un état numérique qui a soit la valeur *normale* ou la valeur *alarme* et qui soit le résultat de l'évaluation d'une formule tenant compte de multiples valeurs analogiques ou logiques : le système SCADA peut automatiquement contrôler si la valeur analogique en un point de mesure se trouve en dehors des valeurs limites haute et basse associées à ce point.

Exemples d'indicateurs d'alarme comprennent une sirène, une fenêtre surgissant sur un écran ou une zone colorée ou clignotant sur un écran (qui pourrait agir de manière similaire à la lumière « réservoir vide » dans une voiture) ; dans chaque cas, le rôle de l'indicateur d'alarme est d'attirer l'attention de l'opérateur sur la partie du système « en alerte » afin que des mesures appropriées puissent être prises. Dans les systèmes SCADA conception, des précautions doivent être prises lorsqu'une cascade d'événements d'alarmes se produit dans un court laps de temps, sinon la cause sous-jacente (qui pourrait ne pas être le premier cas détecté) peut se perdre dans le bruit.

Time	Event	Description - Value	Domain - Nature	Priority
23/02/12 15:51:54	Alarm on - not ack.	Water stream	Water	0
23/02/12 15:51:54	Alarm on - not ack.	Fire temperature	Fire - Temperature	4
23/02/12 15:51:54	Alarm on - not ack.	Electric power supply	Electrical	2
23/02/12 15:51:54	Alarm off	762x221 Electric over-tension	Electrical - Overload	3
23/02/12 15:51:54	Alarm on - not ack.	Water stream 1	Water	11
23/02/12 15:51:54	Alarm on - not ack.	Electric power supp...	Electrical	12
23/02/12 15:51:54	Alarm off	Electric over-tensio...	Electrical - Overload	13
23/02/12 15:51:54	Alarm on - not ack.	Fire temperature 1	Fire - Temperature	14
23/02/12 15:51:55	Alarm off	Temperature - High...	Temperature - Level	0
23/02/12 15:52:08	Alarm off - not ack.	Water level	Water - Level	1
23/02/12 15:52:08	Alarm off - not ack.	Water level 1	Water - Level	11

Figure 2.5 : Gestion des alarmes, défaillances.

Les IHMs sont en évolution rapide.

2.3.4 Les systèmes de contrôle SCADA

a Définition

SCADA "Supervisory Control and Data Acquisition" est un système de contrôle de processus industriel en temps réel utilisées pour surveiller et contrôler l'équipement industriel local ou à distance tels que les moteurs, les vannes, pompes, relais, etc.[10]

Prenons comme exemple la détection d'une fuite au niveau d'un pipeline, le système transfère les informations vers un site central, puis avertit la station d'accueil qu'une fuite a eu lieu pour effectuer les analyses et le contrôle nécessaires, afin de déterminer si la fuite est critique ou pas, et affiche les informations d'une manière logique et organisée (voir figure 2.6).

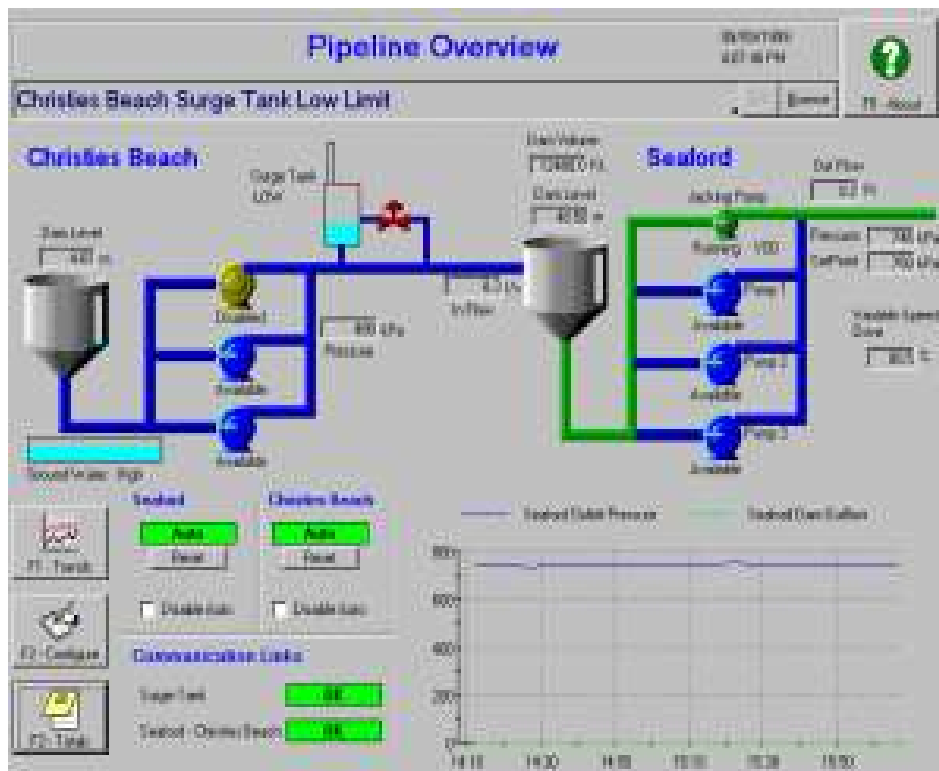


Figure 2.6 : exemple de supervision des pipelines.

b Caractéristiques du SCADA

Les systèmes SCADA peuvent être caractérisés par leur [10] :

- Simplicité et convivialité.
- Traitements graphiques (icônes, bibliothèques, ...).
- fonction de Supervision à savoir :
 - La Commande
 - La Surveillance
 - Le Traitement des alarmes
- Archivage des informations au cours de traitement.
- Programmation.
- rapport Performances/Prix :
 - Prix : matériel + système d'exploitation, logiciel, mise à jour, assistance, documentation.

c SCADA hardware

Un système SCADA se compose d'un certain nombre d'RTU (Remote Terminal Unit) distants, ces derniers s'occupent de la collection des informations pour être acheminés à la centrale via un système de communication.

La station centrale traite ces informations et permet à l'opérateur d'effectuer des tâches de commande à distance. Le traitement ainsi fait (précis et mis à jour) permet l'optimisation de l'exploitation de l'unité de production. Il en résulte un moindre coût de fonctionnement par rapport aux autres systèmes non automatisés.[11]

Pour les autres systèmes SCADA plus complexes, il existe essentiellement cinq niveaux hiérarchiques :

- dispositif d'instrumentation et de contrôle niveau capteur actionneur (field level).
- Les dispositifs de sélection et RTU.
- Les protocoles de communication (qui sont souvent propriétaires).
- Les stations maîtres.
- le département informatique de traitement des données commerciales (le niveau hiérarchiquement le plus haut).

Les RTUs servent d'interface pour les capteurs analogiques et numériques situés sur des sites distants.

Le système de communication fournit un protocole de communication entre les stations maîtres et les sites distants. La liaison physique peut être filaire, fibres optiques, liaison radio (voir figure 2.7), ligne téléphonique, micro ondes, et peut être même par satellite. Les protocoles de communication sont optimisés pour garantir l'arrivée des informations sans erreurs aux stations maîtres en provenance des divers RTUs. [11]

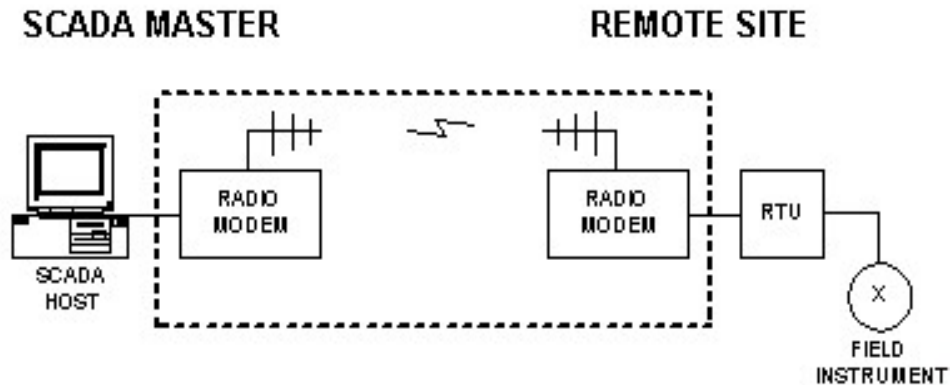


Figure 2.7 : Communication entre SCADA Master et RTU Site.

d SCADA software

Les logiciels SCADA peuvent être divisés en deux types :

- Propriétaire : Les constructeurs d'API et les développeurs des logiciels SCADA développent parfois des logiciels propriétaires pour communiquer avec leur propre matériel. ces systèmes sont vendus comme (clés en main).
Le principal problème de ce système est la dépendance écrasante du fournisseur du système
- Ouverts : les logiciels ouverts ont beaucoup gagné en popularité en raison de l'interopérabilité qu'ils apportent au système. L'interopérabilité est la capacité de mélanger les équipements des différents fabricants sur le même système.
CITECT et Wonderware ne sont que deux des logiciels ouverts disponibles dans le marché des systèmes SCADA. [11]

Des composants typiques d'un système SCADA sont représentés dans le schéma suivant :

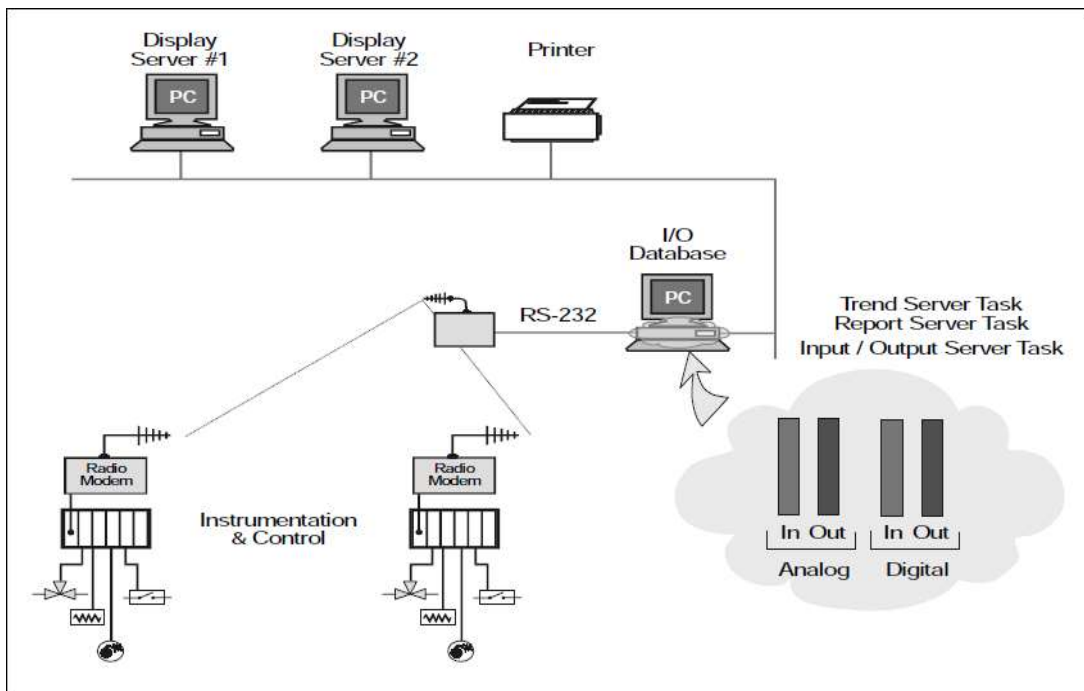


Figure 2.8 : Typique d'un Système SCADA.

✚ Principales caractéristiques du logiciel SCADA sont :

- L'interface utilisateur
- L'affichage graphique
- Traitements des alarmes
- Le graphe des tendances
- L'interface avec les RTUs
- Evolutivité
- Accès aux données
- Base de données
- Mise en réseau
- Tolérance aux pannes et la redondance
- Traitement distribué client- serveur.

e SCADA et les réseaux locaux

Les réseaux locaux traitent généralement tous le partage d'informations et de ressources.

Pour permettre à tous les nœuds des réseaux SCADA de partager des informations, ces derniers doivent être reliés par un support de transmission. Le procédé de connexion de ces nœuds est connue sous le nom « topologie du réseau ». Les nœuds du réseau partagent le même support de transmission de telle manière à permettre l'accès au support sans perturber les informations établis par l'expéditeur.

Un réseau local (LAN) est un lien de communication entre les ordinateurs, les fichiers-serveurs, les terminaux, les stations de travail et beaucoup d'autres périphériques intelligents, qui sont généralement dénommés hôte. Ethernet est plus largement utilisé dans les réseaux locaux d'aujourd'hui pour sa facilité à l'utilisation (voir figure 2.10).

Raccordement du réseau SCADA au LAN permet à quiconque au sein de l'entreprise avec le bon logiciel et la permission, d'accéder au système. Pour des raisons de sécurité certains utilisateurs sont limités qu'à la lecture des informations (voir figure 2.9). [11]

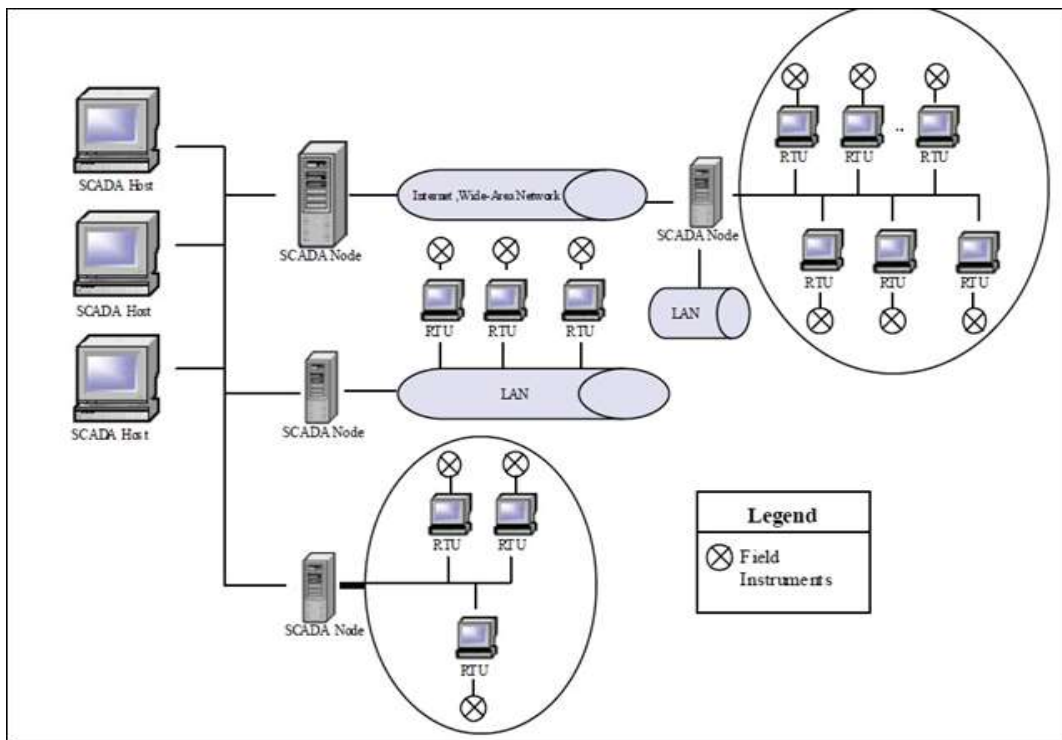


Figure 2.9 : SCADA et le réseau LAN.

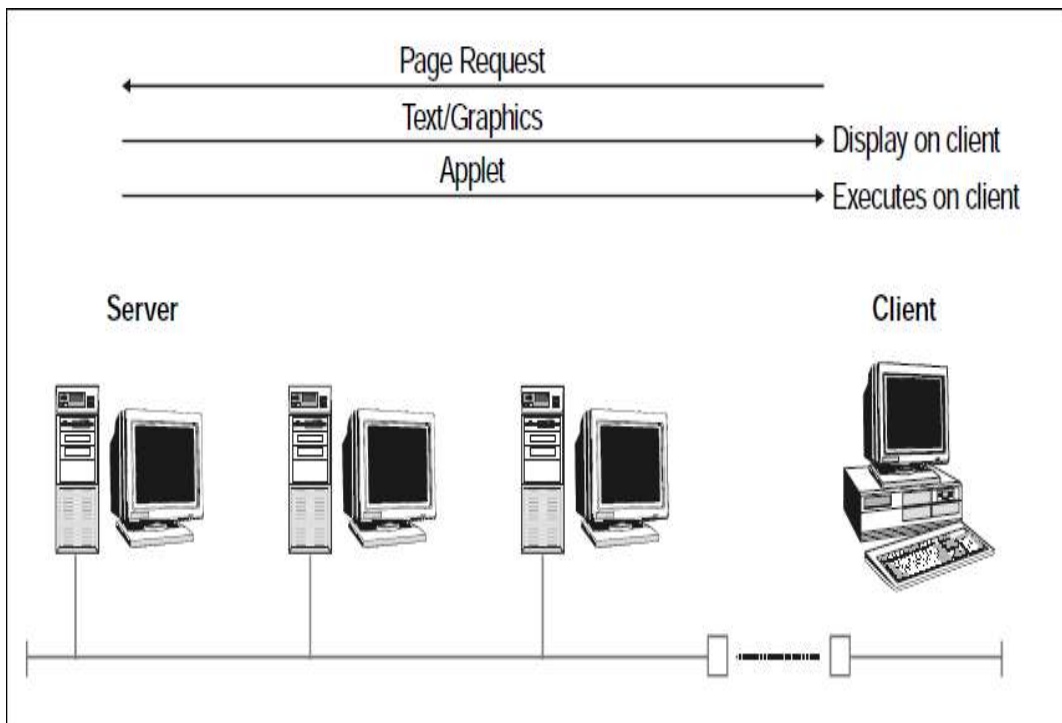


Figure 2.10 : Ethernet utilisé pour transférer des données sur un système SCADA.

2.3.5 Le DCS (Distributed control System)

Dans un DCS, les fonctions d'acquisition et de contrôle des données sont effectuées par un certain nombre d'unités à microprocesseur, qui sont distribuées et situées à proximité des appareils qui lui sont connectés, ou l'instrument à partir duquel les données sont recueillies (voir figure 2.11).

L'évolution du système DCS a permis un contrôle très sophistiqué des boucles analogiques ; et à travers l'ensemble des interfaces qui lui sont étroitement intégrés, permet de faciliter la configuration du système et le contrôle par l'opérateur (la vitesse de transmission peut atteindre les 100 Mbps) (voir figure 2.12). [11]

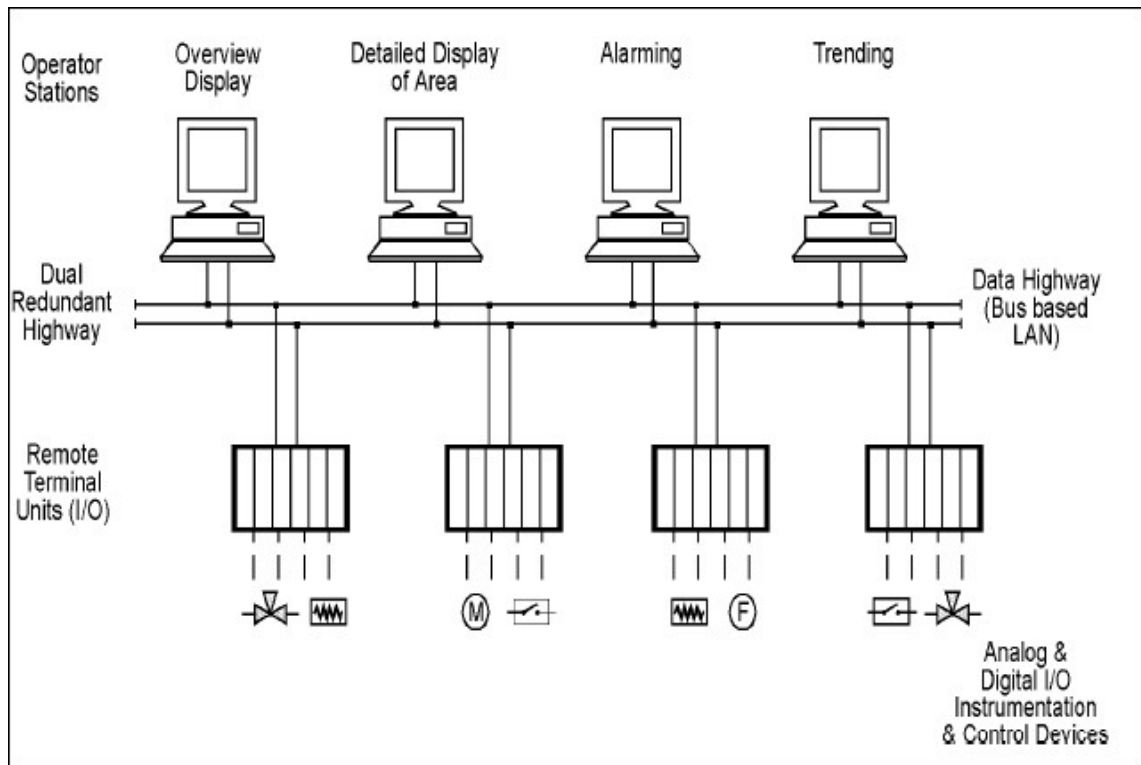


Figure 2.11: Distributed control system (DCS).

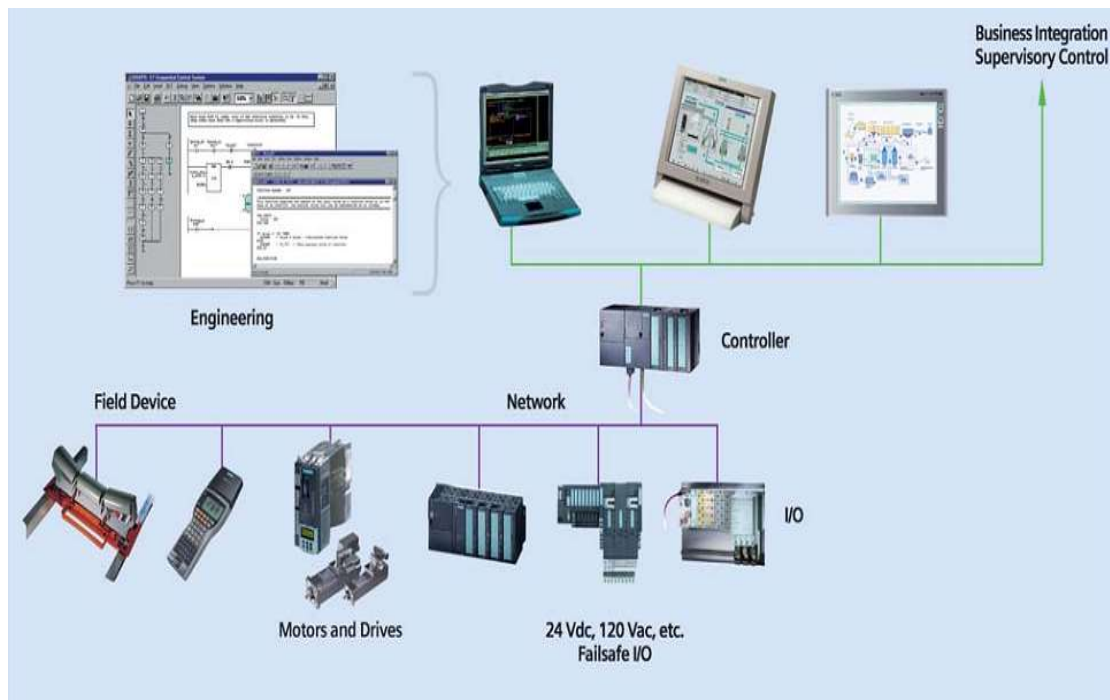


Figure 2.12: Distributed control system (DCS).

Les objectifs de DCS et SCADA sont très différentes. Il est possible qu'un système unique puisse effectuer les fonctions DCS et SCADA, mais les systèmes conçus pour fonctionner sous ce principes son peu nombreux.

Le DCS est orienté "processus", il considère le processus à contrôler (usine de produit chimique ou autres) comme étant son centre d'intérêt, il présente les données à l'opérateur DCS (qui peuvent être des API) comme élément de travail. Alors que le SCADA est orienté "collecte-de-données", il considère les opérateurs (RTU) qui s'occupent de la collecte des données et le centre de contrôle comme étant ces centres d'intérêt. Les équipements distants servent à recueillir les données.

L'opérateur DCS (généralement API) est étroitement lié au entrées / sorties, lorsqu'il a besoin d'une donnée, il a récupère directement, soit a partir du câblage, soit a partir du bus de terrain, soit a partir du réseau. Alors que pour le système SCADA la qualité des données échanger avec les opérateurs (RTUs) est un aspect très important pour son bon fonctionnement. C'est pour cette raison le système SCADA a besoin d'obtenir des données sécuriser pour un support potentiellement lent et peut fiable. Le

Le système DCS est toujours connecté à sa source de données alors que le SCADA est relié à une base de données qui doit être continuellement mise à jour.

Une différence très significative s'impose alors entre les deux architectures.

a Application des systèmes DCS :

Les systèmes DCS sont appliqués dans plusieurs usages tels que [11] :

- Les réseaux électriques et les usines de production d'électricité.
- Systèmes de contrôle de l'environnement.
- Les feux de circulation.
- Les signaux radio.
- Systèmes de gestion de l'eau.
- Usines de raffinage du pétrole.
- Usines de procédé métallurgique.
- Les usines chimiques.
- Fabrication de produits pharmaceutiques.
- Les réseaux de capteurs (voir figure 2.13).
- Marchandises sèches en vrac et les navires de support de pétrole.

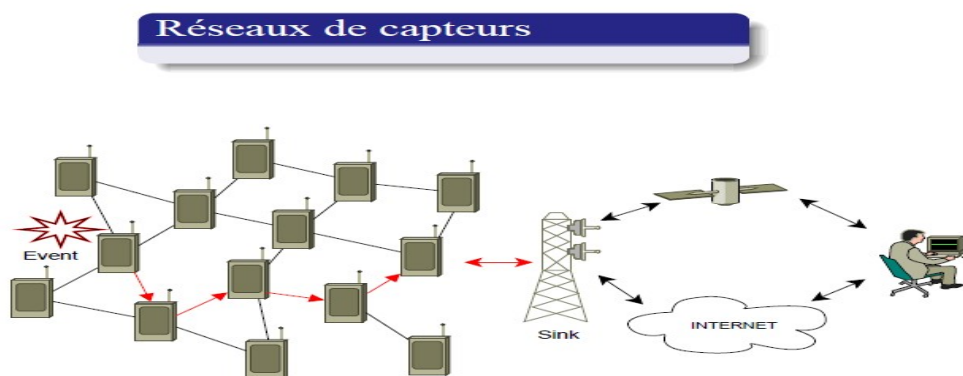


Figure 2.13 : Un exemple des domaines d'applications du système DCS.

2.4 Les constructeurs des systèmes SCADA et DCS et les logiciels mis en œuvre

2.4.1 Introduction

Quelque soit le system utilisé, distribué (DCS) ou centralisé (SCADA), les API se placent souvent dans une hiérarchie de communication dont le niveau le bas se trouve les dispositifs d'entrées et de sorties, tels que les capteurs et les actionneurs. Le niveau suivant fait intervenir de petits API, reliés par un réseau, avec, au niveau supérieur, des API plus importants et des ordinateurs mettant en œuvre un contrôle local. A leur tour, ils peuvent faire partie d'un réseau dans lequel se trouve un mainframe qui contrôle l'ensemble. Et la façon de collecter et traiter les données, fait la différence majeur entre les systèmes DCS et SCADA. Pour répondre à toutes ces exigences, et mettre tout l'automatisme des activités de fabrication en mesure de communiquer, les constructeurs ont développé des logiciels de supervision flexible, permettant aux utilisateurs de construire la solution qui répond au plus près aux besoins de supervision de leurs installations. Parmi ces constructeurs nous citons SIEMENS, SCHNEIDER, et OMRON, bien qu'il existe beaucoup d'autre constructeurs de renommé mondial, mais leur existence sur le territoire national est très timide.

Dans ce qui suit nous parlerons brièvement des logiciels développés par ces constructeurs.

2.4.2 Siemens

a Simatic Manager S7 (logiciel pour API)

L'atelier logiciel STEP 7 Professional fédère toutes les fonctions requises pour concevoir, configurer, programmer, tester, mettre en service et maintenir les systèmes d'automatisation (figure 2.14 et 2.15).

SIMATIC STEP 7 Professional apporte une productivité de l'ingénierie sans précédent. Il concrétise tous les atouts du concept "Totally Integrated Automation" :

- Interface utilisateur orientée objet, commune à tous les outils logiciels SIMATIC Industrial Software.
- Base de données unique des projets garantissant la cohérence des applications même complexes.
- Cohérence des réseaux de communication entre les composants d'une application.[12]

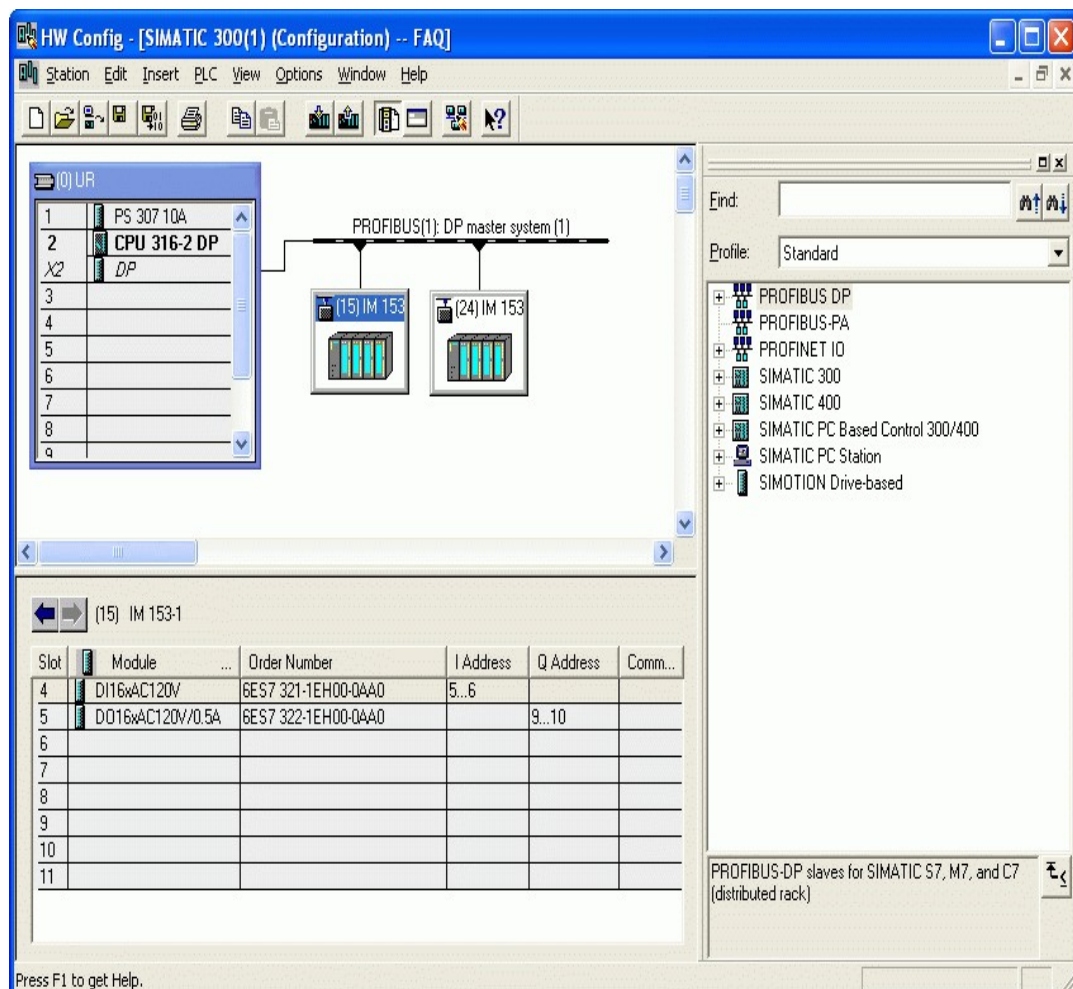


Figure 2.14 : Plateforme logiciel SIMATIC S7.

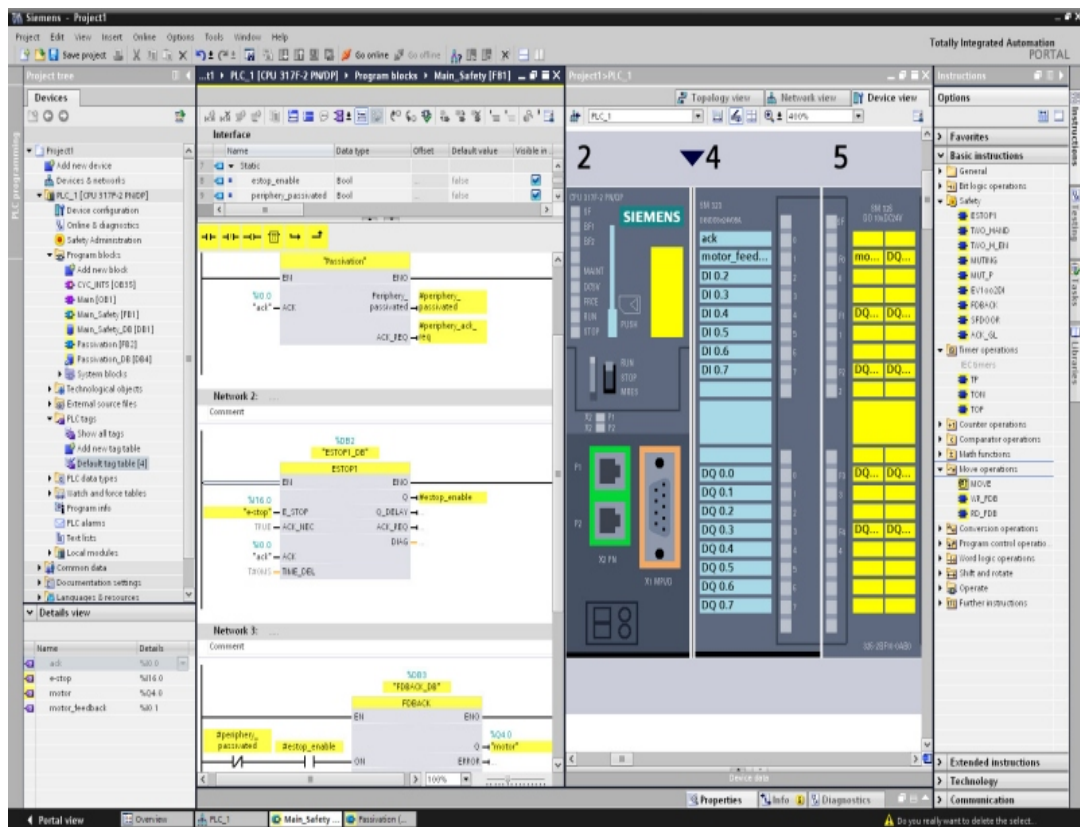


Figure 2.15 : Plateforme logiciel Siemens TIA PORTAL.

❖ **Caractéristiques techniques:**

STEP 7 Professional intègre en particulier les outils suivants :

- Interface utilisateur SIMATIC Manager commune à tous les outils logiciels intégrés et optionnels
- Tous les langages de programmation pour automates programmables définis dans le standard CEI 61131-3: schémas contact, logigrammes, listes d'instructions, graphes séquentiels (S7-GRAPH) et langages structuré (S7-SCL)
- Le logiciel de simulation automate S7-PLCSIM pour la mise au point de programmes sans disposer des automates cible
- Outil configuration graphique des composants matériels et des réseaux de communication. [12]

STEP 7 Professional comporte de nombreuses fonctions évoluées qui contribuent à l'efficacité du travail des automaticiens, par exemple:

- Programmation structurée, avec des blocs fonctions paramétrables et instanciables.
- Vérification globale de la cohérence d'un programme.
- Gestion multilingue de projet.
- Interfaces d'import/export de code source des programmes, de la liste des noms symboliques de variables, de configurations matérielles.
- Modèle objet STEP 7 pour réalisation de scripts destinés à "automatiser le travail de l'automaticien".[12]

b WinCC flexible (logiciel pour IHM)

SIMATIC WinCC flexible est le fruit d'un perfectionnement systématique des logiciels d'interface homme-machine.

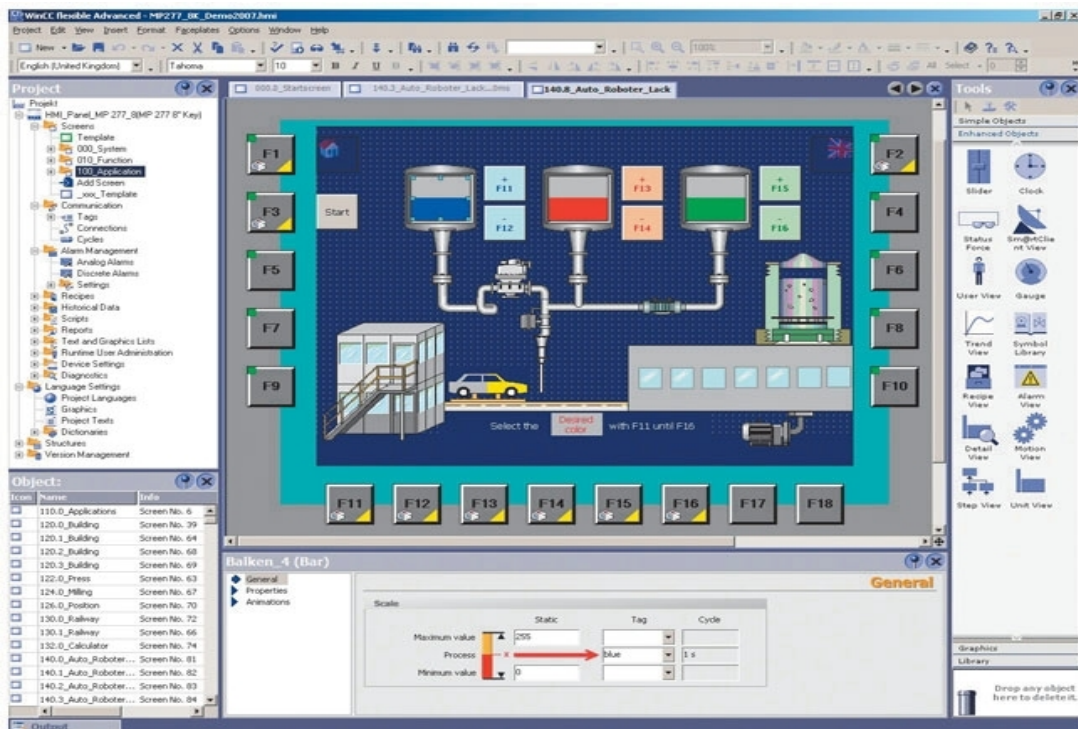


Figure 2.16 : Plateforme logiciel WinCC flexible.

❖ Fonctionnalités: [13]

Le WinCC flexible offre, pour les applications au niveau machine (couvert jusqu'à présent par la famille ProTool), un considérable gain d'efficacité dans la configuration ainsi que des concepts d'automatisation innovateurs.

Dans les secteurs proches du process, de la construction d'installations et de machines ainsi que de la construction de machines de série, SIMATIC WinCC flexible 2005 SP1 permet en outre :

- d'améliorer la productivité (efficacité de la configuration) lors de la création de projets IHM
- de réaliser des concepts d'IHM et d'automatisation innovants dans le cadre de réseaux TCP/IP et du Web
- d'accroître la disponibilité des machines et installations par de nouveaux concepts de maintenance
- d'accéder facilement, en toute sécurité aux données de process à partir de n'importe quel endroit du globe.

❖ Caractéristiques techniques: [13]

- Intégration dans les automates programmables,
- Manipulation du projet,
- Editeurs de tableau,
- Gestion de données orientée objet avec possibilités d'édition et de recherche confortables,
- Bibliothèques d'objets de configuration prédéfinis ou confectionnés par l'utilisateur,
- Prise en charge linguistique,
- Visual Basic Script Support,
- Run time,
- Test et assistance à la mise en service,
- Communication ouverte entre systèmes HMI et systèmes de niveau supérieur.

c Wincc (logiciel pour SCADA)

SIMATIC WinCC est un système de supervision de processus modulaire 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. WinCC se distingue en particulier par son ouverture totale. Ses fonctions sont facilement combinables avec des programmes standards et utilisateur pour donner naissance à des solutions IHM répondant exactement aux besoins de la pratique (voir figure 2.17). [14]

Par la mise en œuvre de l'option serveur, vous pouvez étendre un système WinCC monoposte en un système client/serveur performant. De cette manière, vous pouvez exploiter plusieurs postes de conduite et de supervision coordonnés en association avec des systèmes d'automatisation en réseau. Un serveur peut approvisionner 32 clients en données de process et d'archive, messages, vues et journaux. La condition préalable à ceci est une connexion réseau (TCP/IP) entre l'ordinateur servant de serveur et les clients raccordés. [14]

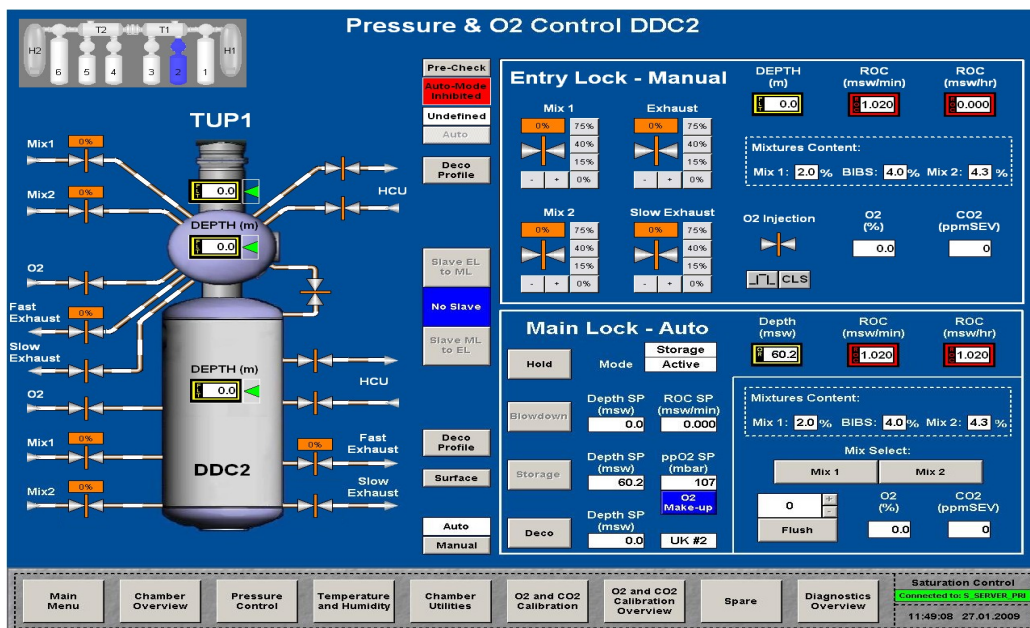


Figure 2.17 : Exemple d'une vue créée par WINCC.

2.4.3 Schneider

a Unity pro (logiciel pour API)

Unity Pro est le nouvel atelier logiciel de SCHNEIDER ELECTRIC pour la programmation de ses automates. C'est un logiciel "tout en un" permettant le développement de programmes et l'exploitation des automatismes Schneider.

UNITY-Pro par le biais de SOFTLINK offre un accès bidirectionnel à la quasi-totalité des informations, qui peuvent être écrites aussi bien que lues par des logiciels tiers. Il présente en outre la faculté de gérer des variables structurées définies par l'utilisateur ; or c'est par la définition de ce type de variables qu'il est possible d'établir les liens entre les traitements effectués dans l'automate et la supervision pour des objets génériques (voir figure 2.18).

Dans le cadre d'un accord de partenariat entre AREAL et SCHNEIDER ELECTRIC (programme CAPP), des développements ont été menés à bien pour permettre de façon très poussée l'intégration et l'unicité des données partagées par Unity-Pro et TOPKAPI (développeur de programme de supervision). Concrètement cela signifie qu'après avoir défini des variables dans la programmation automate, vous pourrez aisément les incorporer à l'application de supervision ou, réciproquement, qu'après avoir placé des objets dans les synoptique graphiques et par là même créé les variables de supervision, vous pourrez demander que les instances correspondantes soient intégrées au programme automate.

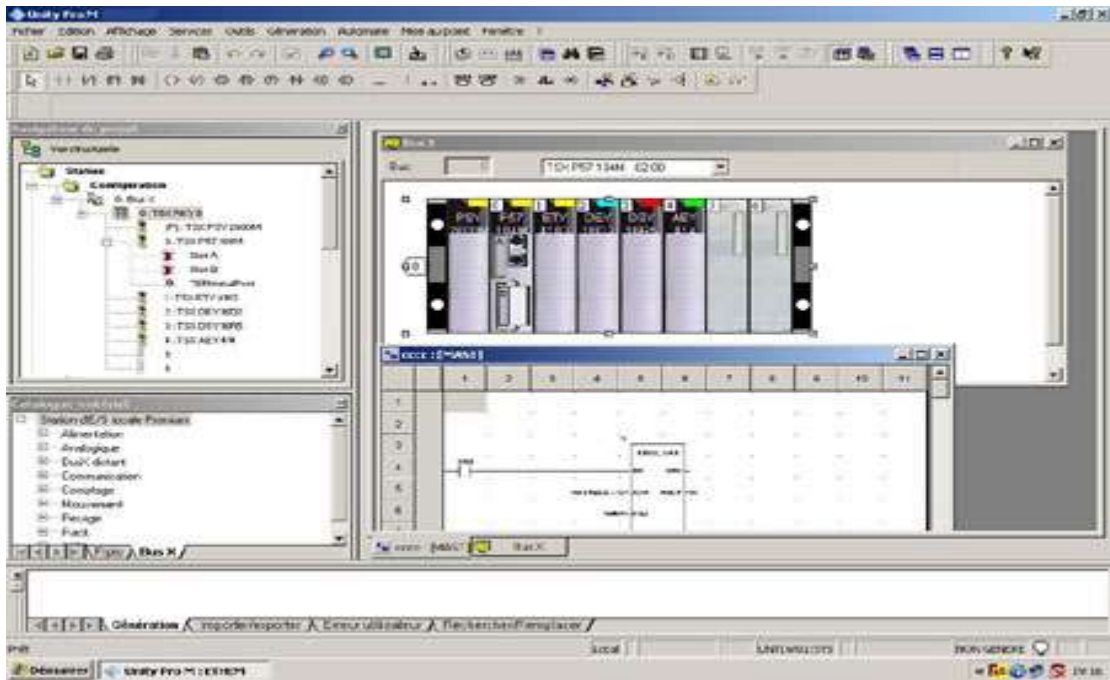


Figure 2.18 : Plateforme logiciel de l'UNITY PRO.

b Vijeo Designer (logiciel pour IHM)

Vijeo Designer est le logiciel de configuration pour les IHM, développé par SCHNEIDER ELECTRIC pour les terminaux graphiques et PC industriels.

Flexible, Vijeo Designer conçoit de la plus simple application de conduite à la plus complexe des installations IHM. Ouvert, ce logiciel fonctionne sur les compatibles PC équipés du système d'exploitation Windows et s'intègre aussi bien dans les solutions Télémécaniques que les architectures tierces. Ingénieux, il offre des fonctions de scripts avancées pour personnaliser votre IHM et répondre à vos moindres contraintes. [15]

Vijeo Designer est souvent employé dans, le secteur de construction de machines simples ou complexes (automobile, composants électroniques, produits pharmaceutiques, chimie), le bâtiment, le secteur agroalimentaire, et le traitement de l'eau et des déchets.

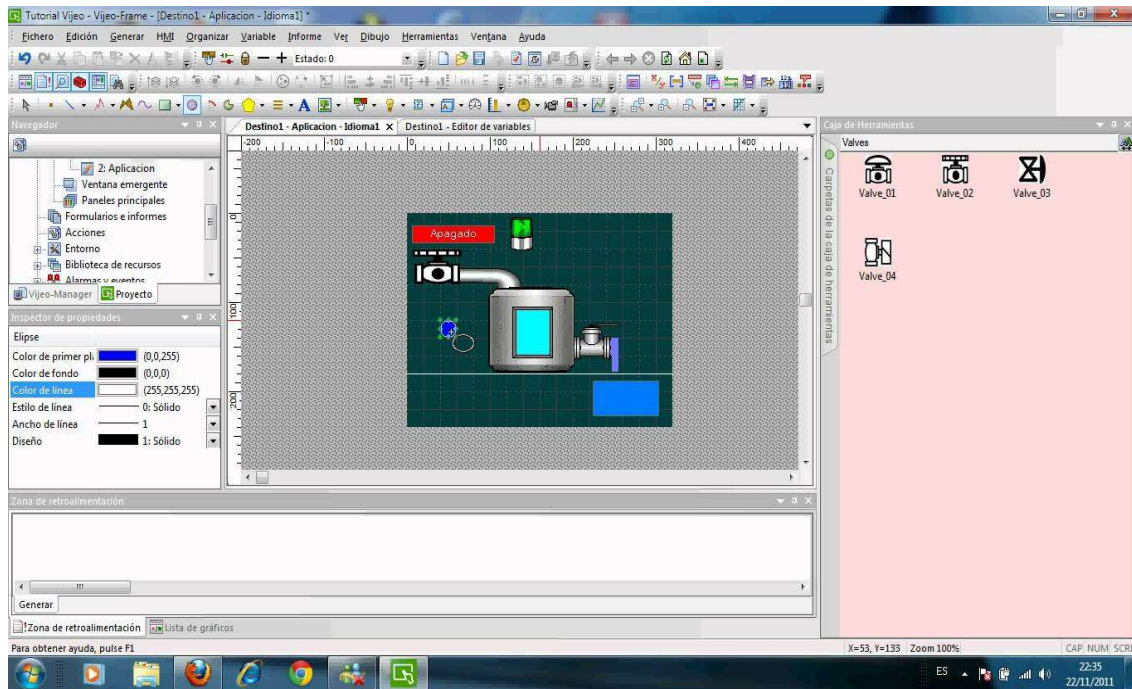


Figure 2.19 : Plateforme logiciel Vijeo Designer.

c Vijeo Citect (logiciel pour SCADA)

Citect SCADA offre toutes les fonctions d'un superviseur moderne. Son architecture distribuée client-serveur s'applique à une multitude d'applications dans les domaines les plus variés:

- Énergie et infrastructures: aéroports, routes et tunnels, eau, pétrole, gaz
- Industrie: agro-alimentaire, mines, métallurgie, matières premières minérales.

Un outil de développement unique permet le développement de toutes les applications de supervision, depuis les petits systèmes autonomes, jusqu'aux grands systèmes distribués redondants.

Les utilisateurs et les créateurs d'applications bénéficient de la compétence de Schneider Electric pour la maîtrise de l'intégration de systèmes et l'optimisation du cycle de vie des produits. Enfin la stratégie de renouvellement de l'offre Citect SCADA garantit qu'elle est toujours compatible avec les technologies massivement adoptées par le marché. [16]



Figure 2.20 : Plateforme logiciel Vijeo Citect.

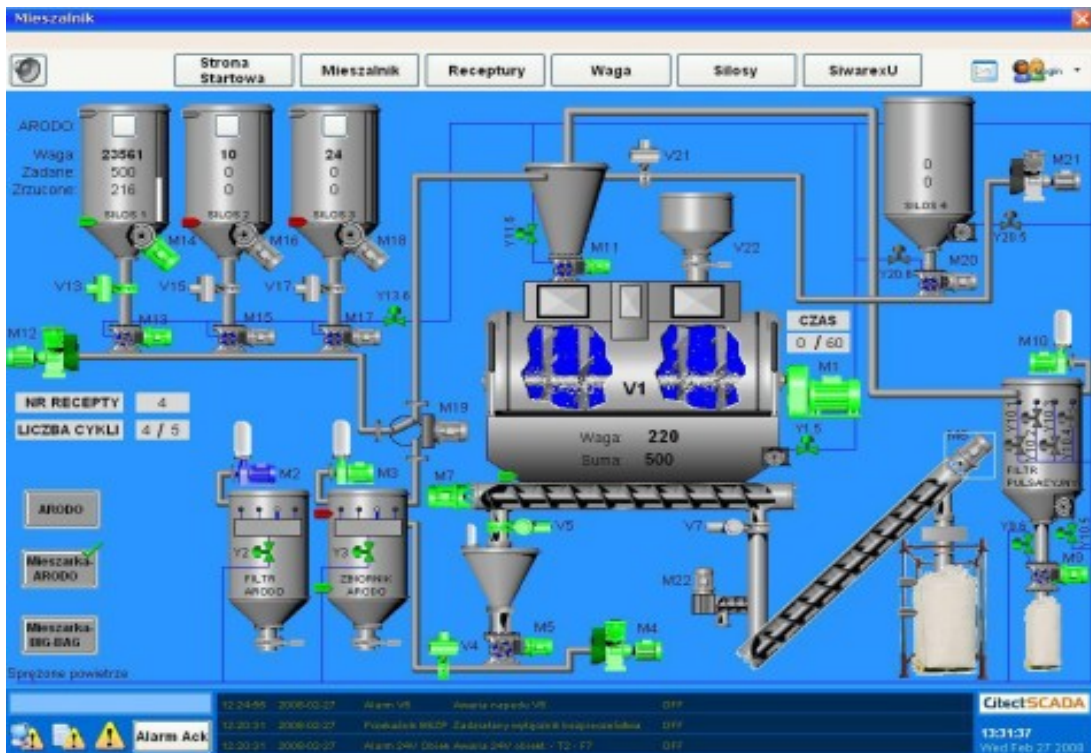


Figure 2.21 : Exemple d'une vue crée par Vijeo Citect.

2.4.4 OMRON

a CX-Programmer

CX-Programmer, le logiciel de programmation de toutes les séries d'API d'OMRON, est entièrement intégré dans la suite logicielle CX-One.

CX-Programmer comprend de nombreuses fonctions pour accélérer le développement du programme API. Les boîtes de dialogue de réglage des paramètres réduisent la durée de la configuration et avec les blocs fonctions standard en texte structuré ou en langage de schéma à contacts, CX-Programmer fait du développement des programmes API une simple configuration par glisser-déplacer [17]. La figure (2.22) montre la plateforme du logiciel CX Programmer.

Et c'est ce logiciel que nous avons utilisé pour développer et maitre au point notre application.

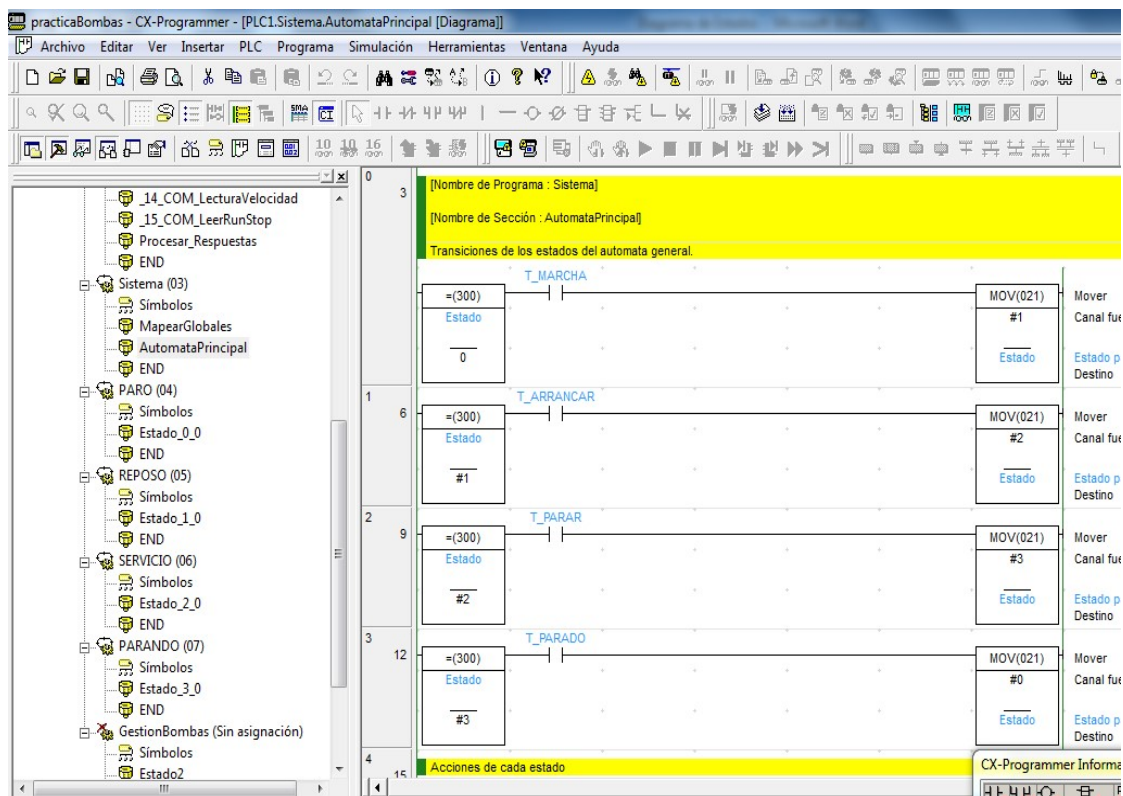


Figure 2.22 : Exemple d'un programme crée par CX Programmer.

b CX-Designer

CX-Designer est le logiciel utilisé pour le développement des applications des IHM de la série NS de 5,7 à 12,1 pouces.

Il est exceptionnellement facile à utiliser et à charger, avec des fonctions puissantes telles que le partage de balises avec les API et une importation et une exportation simplifiées d'étiquettes dans différentes langues. Il possède également une interface utilisateur entièrement personnalisable avec des icônes pour la plupart des fonctions. [18]

Intégré à CX-One, il permet de partager des balises entre les API et les IHM, il suffit de faire un glisser-déplacer depuis CX-Programmer ou de copier et coller depuis Excel. La réutilisation de projets et d'écran se fait aussi avec le glisser-déplacer.

CX-Designer possède une bibliothèque de plus 1 500 composants actifs intelligents préprogrammés, et toutes les propriétés des objets sont accessibles [18]. La figure (2.23) montre la plateforme du logiciel CX-Designer.

Et c'est le deuxième logiciel de la suite CX-one qu'on a utilisé pour la mise au point de notre application.

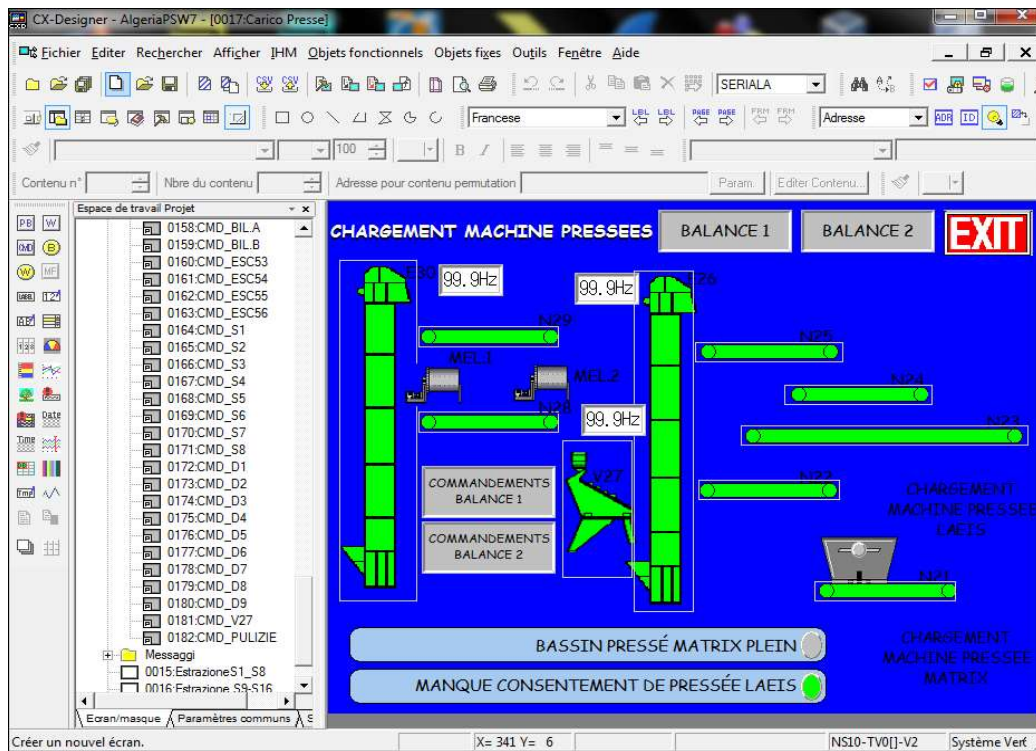


Figure 2.23 : Exemple d'une vue créée par CX Designer.

c CX-Supervisor

CX-Supervisor est dédié à la conception et à l'utilisation de la visualisation et du contrôle des machines sur ordinateur. Il est non seulement simple à utiliser pour de petites tâches de surveillance et de contrôle, mais il offre également des fonctionnalités très intéressantes pour la conception des applications les plus sophistiquées.[19]

CX-Supervisor dispose de fonctions puissantes pour une large gamme de besoins d'interface homme-machine sur PC. Les applications simples peuvent être créées rapidement avec l'aide de bibliothèques et de nombreuses fonctions prédéfinies, et les applications les plus complexes peuvent être générées avec un langage de programmation puissant, ou en VBScript™. CX-Supervisor est d'une utilisation extrêmement simple, intuitive et très conviviale. L'importation de composants ActiveX® permet de créer des applications flexibles aux nombreuses fonctions. [19]

La figure (2.25) montre la plateforme du logiciel CX-Supervisor, ce logiciel n'existe pas dans la suite CX-one, OMRON a préféré le vendre seule.

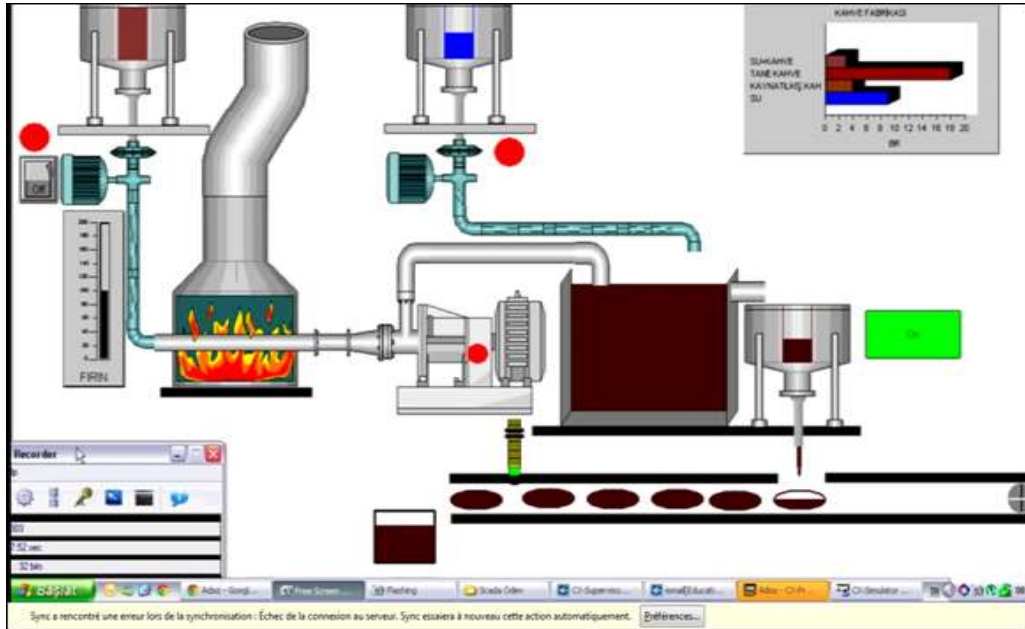


Figure 2.24 : Vue créée par CX-Supervisor.

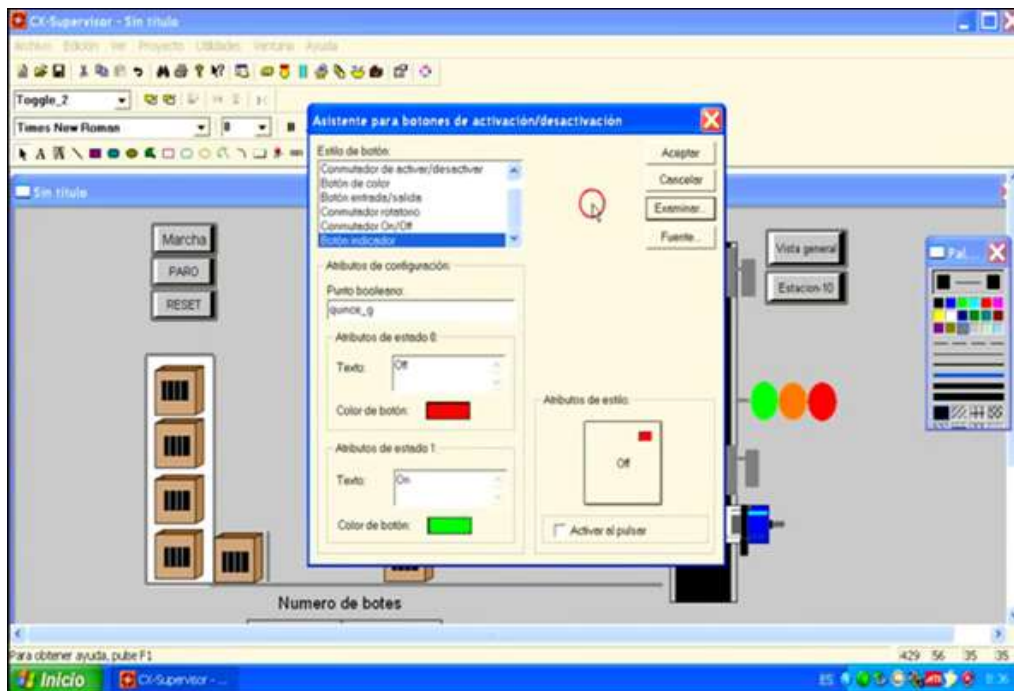


Figure 2.25 : Plateforme du logiciel CX-Supervisor.

2.5 Conclusion

Dans ce chapitre nous avons essayé de simplifier le maximum, le principe de la supervision et du contrôle des procédés industriels, et rendre le plus clairement possible la différence entre les techniques couramment utilisées dans l'industrie, en allant du plus simple, c'est-à-dire celui qui se trouve à proximité des machines comme l'IHM, vers le plus compliqué qui se trouve généralement sur des sites distants comme le SCADA et DCS.

La façon de collecter et traiter les données fait la différence entre SCADA et DCS.

Dans le chapitre suivant nous allons traiter l'application que nous avons réalisée à base d'équipement OMRON.

Chapitre 3 Développement d'un procédé de fabrication piloté par des équipements OMRON

3.1 Introduction

Dans ce chapitre, nous allons traiter l'application qu'on a développé, pour mettre en évidence les système de contrôle et de surveillance évoqués dans le chapitre précédent, mais avant de ce faire, nous avons jugé utile de présenter d'abord le fonctionnement et la configuration du matériel mis en œuvre, qui sont majoritairement du constructeur OMRON.

L'application qu'on a traité, rentre dans le cadre de la supervision par IHM, vu sa simplicité de mise en œuvre par rapport au système SCADA et DCS qui nécessitent des moyens très importants, qui sort de nos capacités financières et physiques.

3.2 Description du procédé industriel

Cette application s'intègre dans la fabrication des pièces en céramique, et l'étape qu'on a visé, c'est la phase finale où les pièces emballées en paquets de 1 m² (voir figure 3.1) sont transportées par un tapis et acheminées vers un robot pour être imprimées et stockées dans des centaines circulaires.



Figure 3.1 : emballage des paquets de 1m².

Les paquets qui contiennent les pièces sont distribués par une machine en amont appelée empaqueteuse, vers un tapis qui les transportes au robot, la vitesse du tapis est contrôlée par un variateur de vitesse OMRON VS mini j7 entièrement programmable, la variation de la vitesse de ce tapis est faite de tel manière à ce que le flux des paquets n'engorge pas le robot. Enfin à l'arrivé au niveau du robot, le tapis s'arrête un petit moment pour laisser le temps au robot d'imprimer le paquet.

Un system de compensation en amont permet de récupérer les paquets qui sortent de l'empaqueteuse lorsque le tapis s'arrête momentanément. Le contrôle et la supervision du procédé a été faite par quatre entêtés communicantes à savoir :

- L'API OMRON CQM1H, c'est le maitre du réseau communicant. Il contient le programme de gestion du procédé, tous les capteurs et actionneurs du processus analogiques et numériques y sont connectés. Ajoutant à ceci une ligne de communication série entre l'API, l'IHM, le variateur de vitesse, et le codeur incrémental.
- L'IHM OMRON NS12, c'est un esclave du réseau il permet de visualiser le procédé, structuré en un certain nombre de pages, où chaque page permet la visualisation d'une partie du procédé pour en donner plus de détails, et pour d'éventuelle intervention manuel.

- Le VS mini J7, c'est un variateur de vitesse OMRON, utilisé principalement pour le déplacement du tapis sous le contrôle de l'API.
- Le codeur incrémental type « elap », relié sur l'arbre du moteur du tapis dans le but de contrôler la vitesse du tapis et la position des paquets sur le tapis.

Ayant donné un descriptif du procédé industriel développé, nous allons par la suite donner une fiche technique des différents équipements utilisés.

3.3 Les outils d'automatisme mis en œuvre

Comme la montre la figure (3.1), les équipements mis en œuvre pour la réalisation du projet, afin de concrétiser ce qui a été dit précédemment sur les procédés de supervision, sont principalement issus du constructeur OMRON très connu pour la robustesse de ces équipements industriels.



Figure 3.2 : Photo de la réalisation, les équipements utilisés.

3.3.1 L'API CQM1H

C'est l'équipement essentiel de l'installation, il contient le programme de gestion.



Figure 3.3 : API OMRON CQM1H.

Le schéma suivant montre les étapes à respecter pour l'installation et l'utilisation d'un système CQM1H.

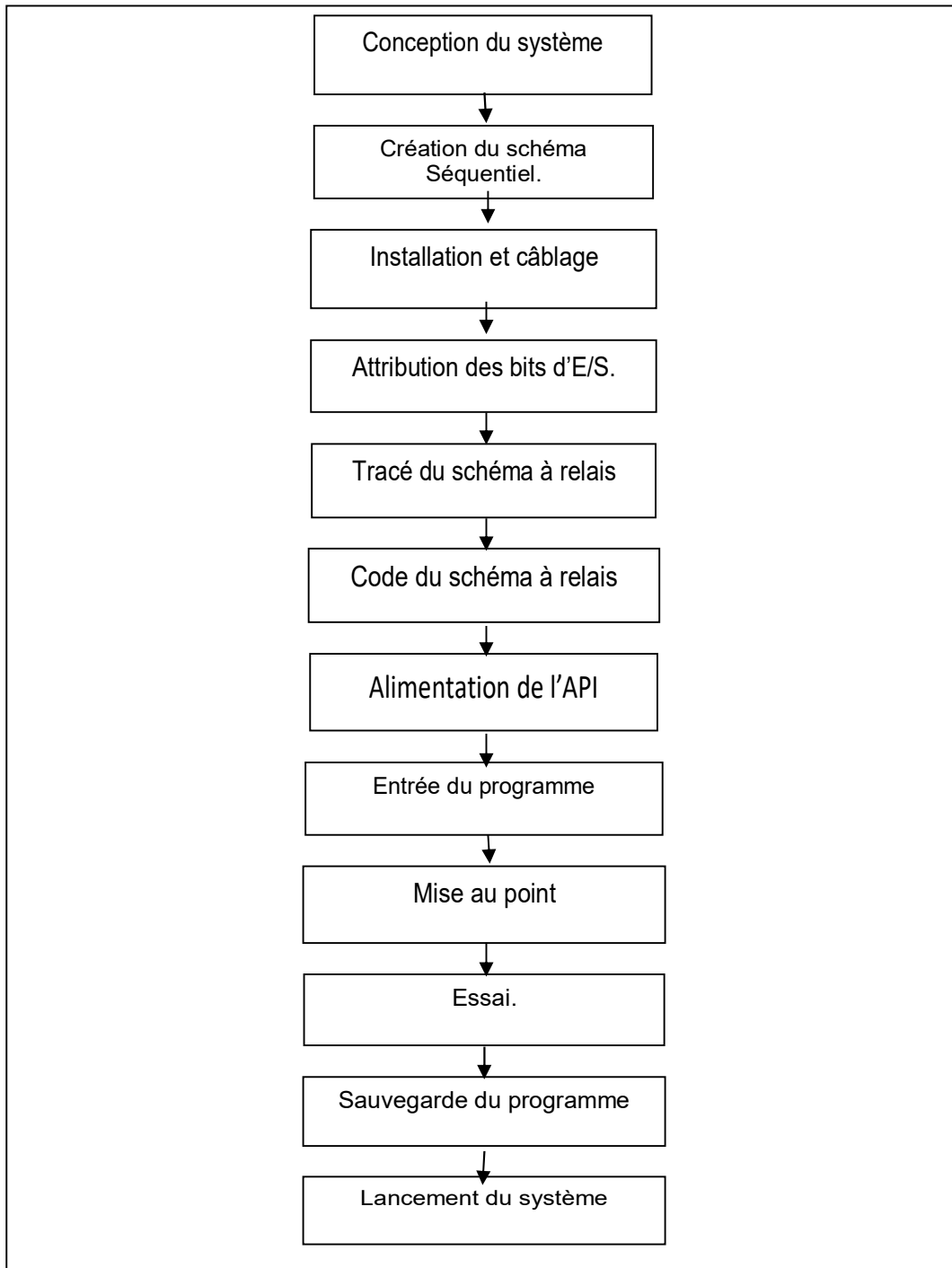


Figure 3.4 : schémas des étapes à respecter pour l'installation et l'utilisation d'un système CQM1H.

a Configuration du système

Le CQM1H est un API compact, rapide, composé d'une alimentation, d'une unité centrale (UC) et de cartes d'E/S. Toutes ces cartes se connectent sur le côté pour constituer un API unique qui se monte généralement sur un rail DIN.

Toutes les UC CQM1H sont équipées d'un port RS-232C qui peut être connecté directement à un ordinateur, un autre CQM1H, ou d'autres appareils en série.

Le schéma de la figure 3.5 montre les configurations possibles avec le CQM1H. [20]

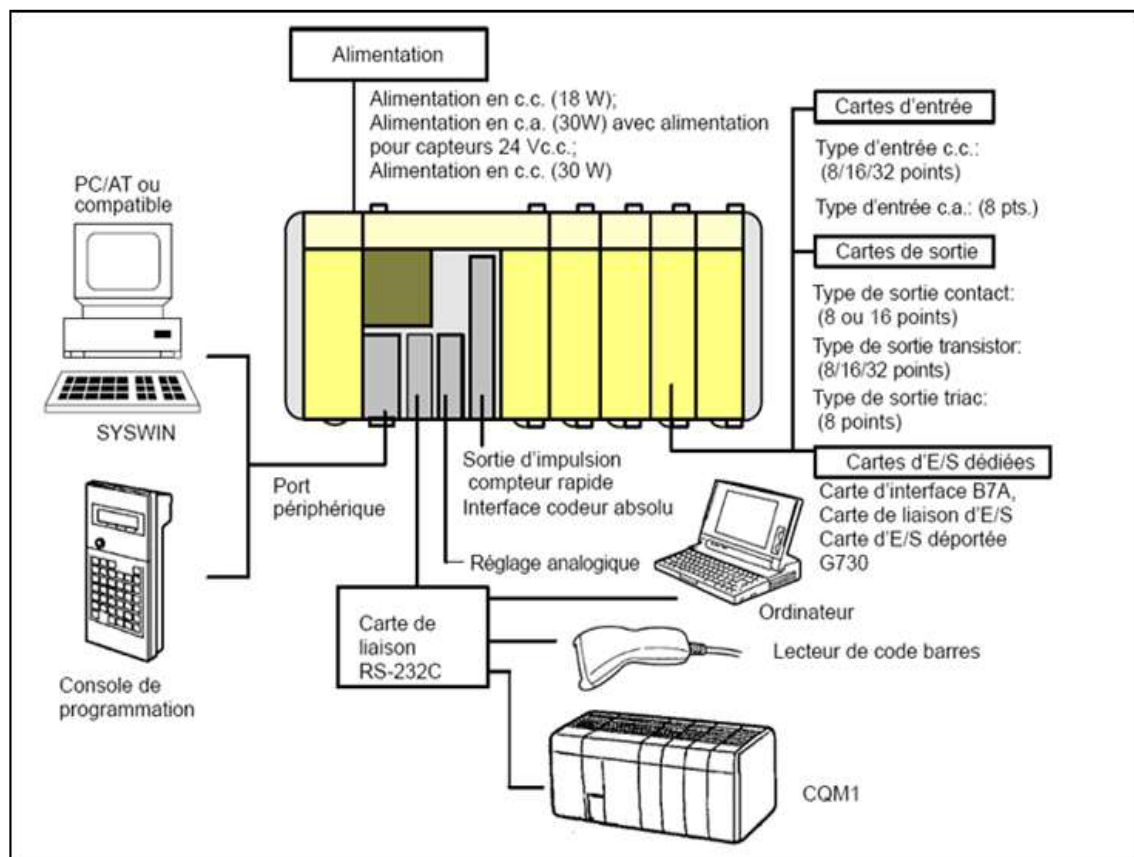


Figure 3.5 : Description d'un Automate Programmable Industriel CQM1H.

Le CQM1H offre la plus grande flexibilité de tous les systèmes API de sa catégorie, il offre également de nombreuses caractéristiques avancées, dont les suivantes:

- 4 processeurs de bases différentes à choisir (CPU 11, 21, 51 ou 61) où chaque CPU intègre 16 entrées à courant continu qui peuvent être étendu jusqu'à 512.

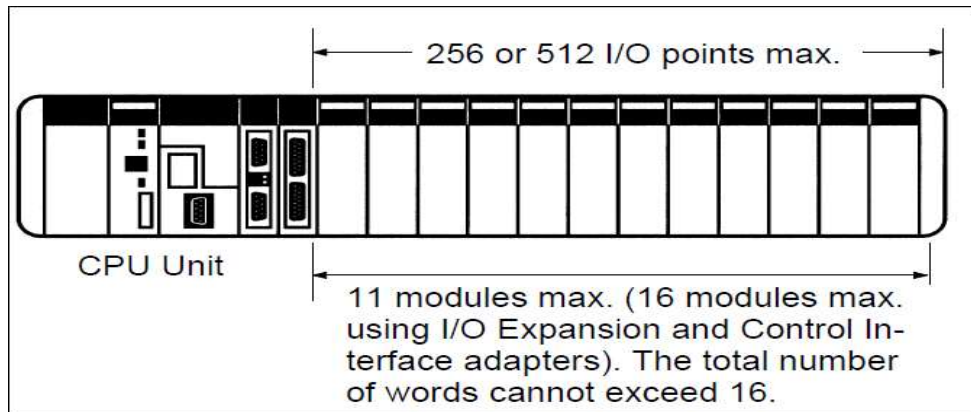


Figure 3.6 : Unité Centrale et les unités d'E/S d'un API CQM1H CPU 51/61.

- Cartes internes permettent une configuration "sur mesure" de la CPU.
- Cassettes de mémoire optionnelles permettent la sauvegarde des données sensibles.
- Des temporisations et des compteurs rapides sont intégrés.
- jeu d'instructions comprenant les régulateurs PID, traitement en virgule flottante, le traitement des protocoles macro-instructions.
- Des sorties sont traitées lorsque les instructions sont exécutées (sorties directes).
- Ajoutant à ceci les interruptions, Sortie d'impulsions, et Communications. [20]

A.1 INTERRUPTIONS

Le CQM1H supporte trois types d'interruptions:

➤ Interruptions d'entrées

Des interruptions d'entrées sont utilisées pour traiter des signaux d'entrée venant d'un appareil externe qui sont plus courts que le temps d'exécution du programme. Des signaux d'entrée avec une largeur d'impulsion aussi courte que 0,1 ms peuvent être utilisés. [20]

➤ **Interruptions programmées**

Des interruptions programmées peuvent être effectuées à l'aide d'une temporisation d'intervalle à grande vitesse.

- Interruptions par compteur rapide.

Des impulsions unidirectionnelles pouvant atteindre 5 kHz et des impulsions bidirectionnelles pouvant atteindre 2,5 kHz peuvent être admis. Des interruptions par compteur rapide peuvent être combinées avec des sorties impulsion pour des applications comme la commande d'un moteur. Le

CQM1H-CPU61 accepte des impulsions unidirectionnelles pouvant atteindre 100 kHz et des impulsions bidirectionnelles pouvant atteindre 50 kHz. Le compteur rapide du CPU61 (entrée du codeur absolu) comporte deux entrées 1 kHz. [20]

A.2 SORTIE D'IMPULSIONS

Des impulsions pouvant atteindre 1 kHz peuvent être sorties par les contacts de la carte de sortie. Le CQM1H CPU 61 peut supporter des ports pour la sortie des impulsions qui peuvent arriver jusqu'au 100 kHz.

A.3 COMMUNICATIONS

Un port périphérique et un port RS-232C sont disponibles et utilisés pour communiquer avec des appareils externes à l'aide des méthodes suivantes.

- ✓ Communication protocole SYSMAC-WAY

Le CQM1H communique avec un ordinateur et un terminal opérateur programmable à l'aide des instructions SYSMAC-WAY.

- ✓ Communication ASCII

Le CQM1, par le port RS-232C, peut lire des données provenant d'un lecteur de codes barres ou d'un appareil de mesure et sortir ces données sur une imprimante.

- ✓ Liaison CQM1 point-à-point

Une liaison de données peut être créée avec une zone de données dans un autre CQM1 pour contrôler l'état de l'autre API et synchroniser les processus contrôlés par les API.

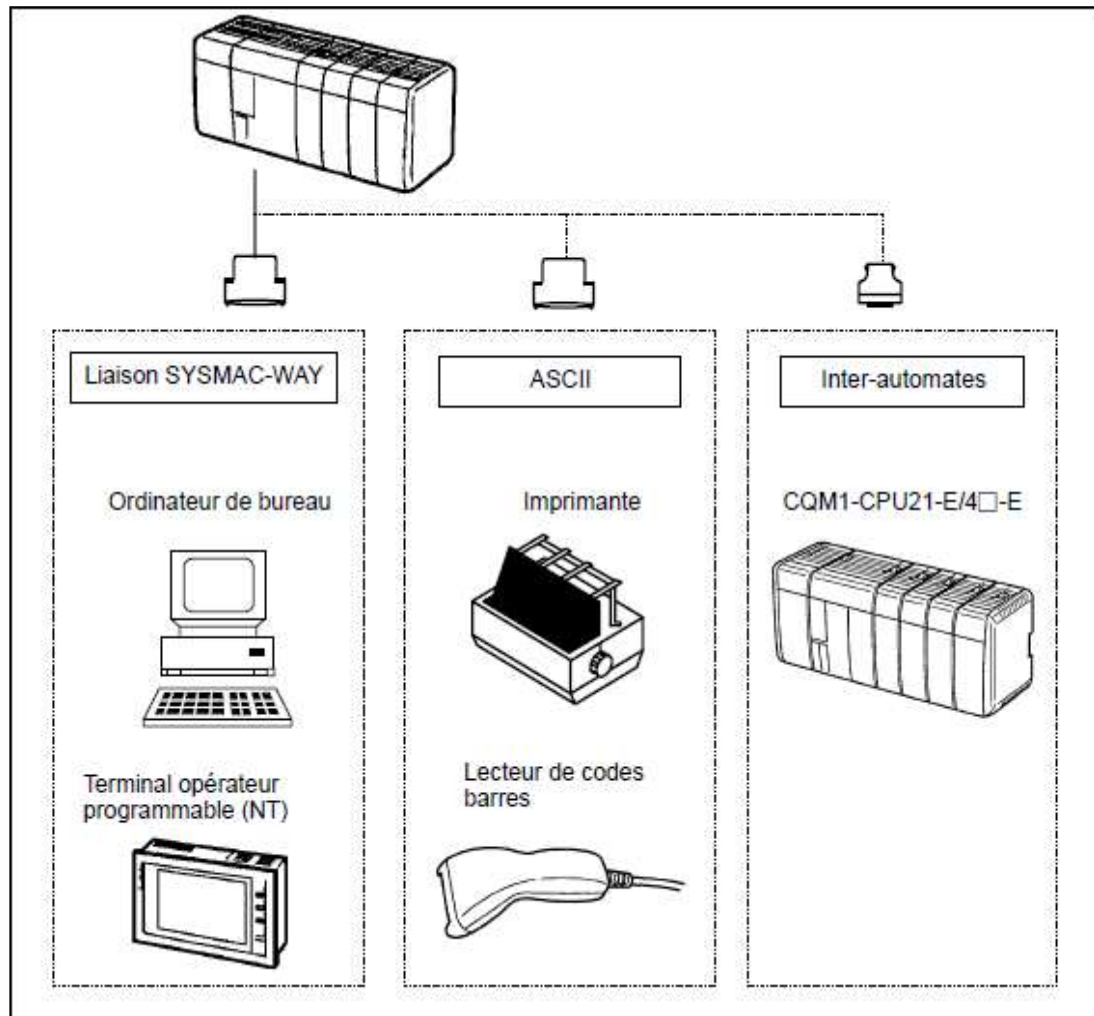


Figure 3.7 : les différents types de liaisons.

b Les modules du CQM1H :

Les modules du CQM1H sont interconnectés à travers des connecteurs se trouvant sur chaque côté d'un module, ce qui permet des connections en amont et en aval de ce module, sauf le module d'alimentation qui dispose d'un seul connecteur sur son coté droit et le module de terminaisons un connecteur sur son coté gauche. Exemples ci-dessous :

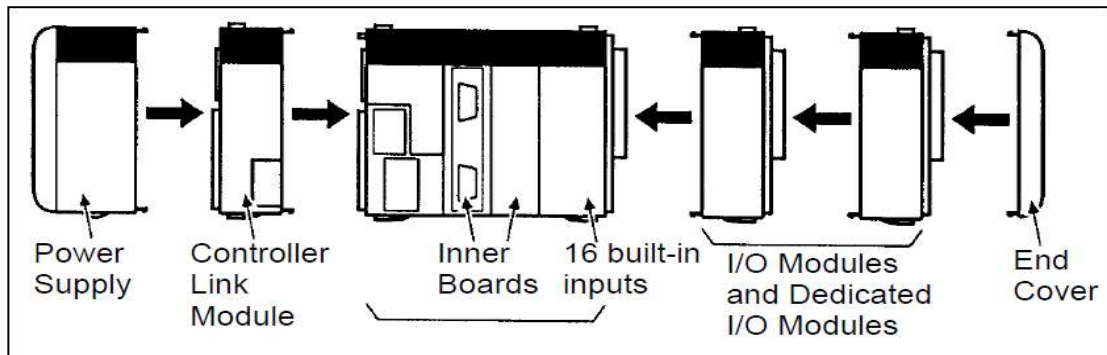


Figure 3.8 : assemblage des modules du CQM1H.

Ces modules peuvent être les suivants :

B-1 UNITÉS D'ALIMENTATION

Les unités d'alimentation à courant continu (C.C) ou à courant alternatif (C.A) peuvent être installées selon le lieu d'utilisation de l'API. Le module d'alimentation C.A exige une entrée d'alimentation de 100 à 240 Vc.a 50/60 Hz, comme dans le modèle CQM1-PA203 utilisé dans notre application.

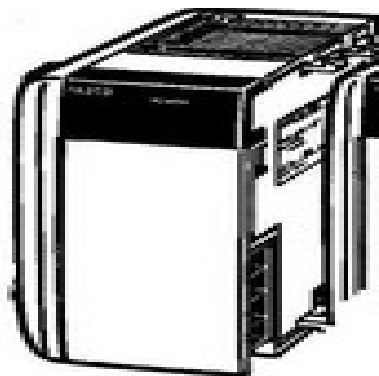


Figure 3.9 : unité d'alimentation CQM1-PA203.

B-2 UNITÉS CENTRALES

Les quatre modèles d'unités centrales peuvent se diviser en deux groupes : les unités centrales économiques (CPU11, CPU21) et les unités centrales à haute fonctionnalité (CPU51, CPU61). Ces unités centrales varient également dans leurs

capacités de programme, capacités d'entrée/sortie, capacités de mémoire et la présence de 1 à 2 ports de communication.

Vue d'ensemble d'une unité centrale du CQM1H-CPU61 :

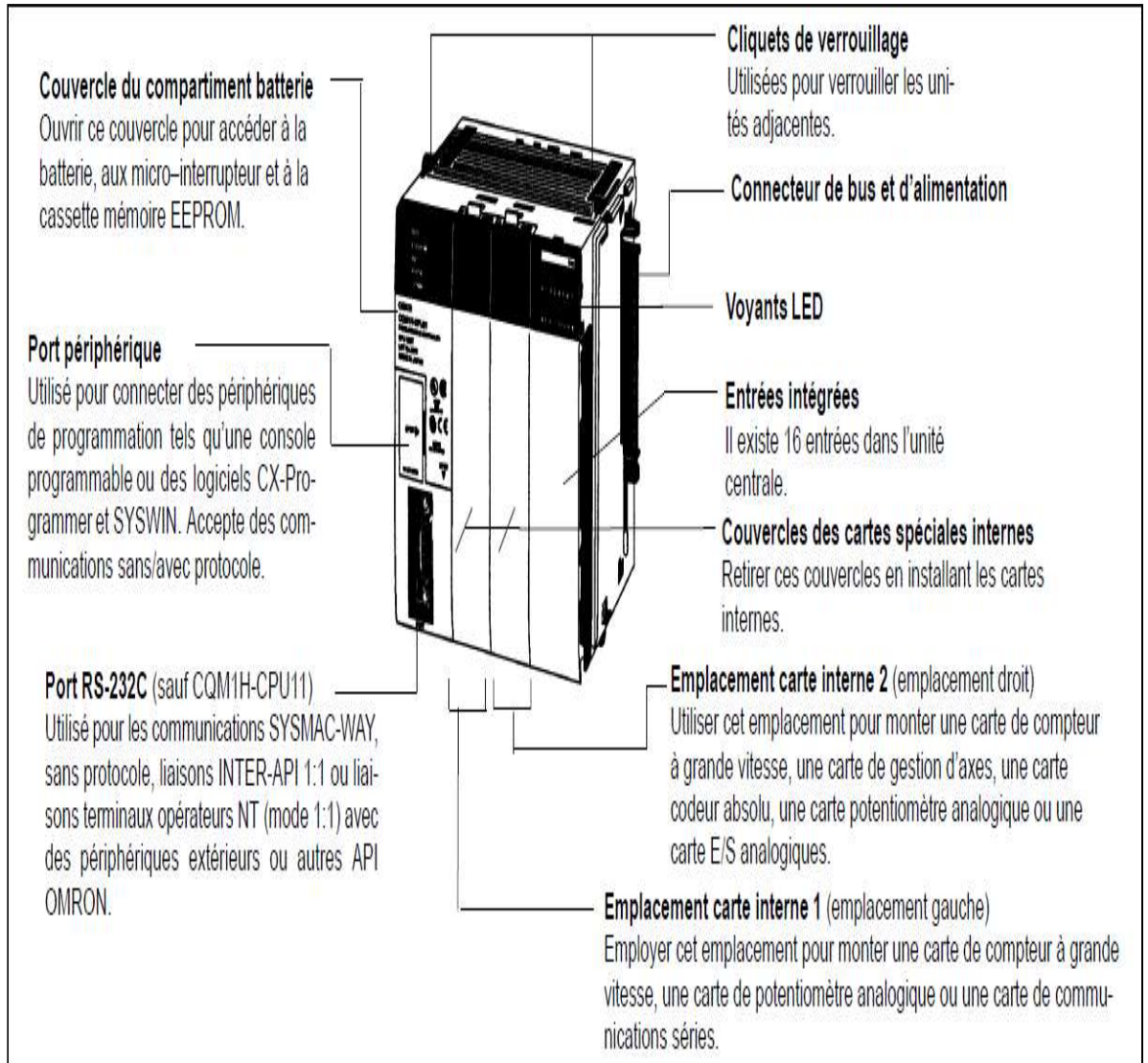


Figure 3.10 : Unité centrale du l'API CQM1H CPU 61.

B-3 LES MODULES D'ENTRÉES/SORTIES

La figure suivant présente les composants de base d'une carte d'E/S.

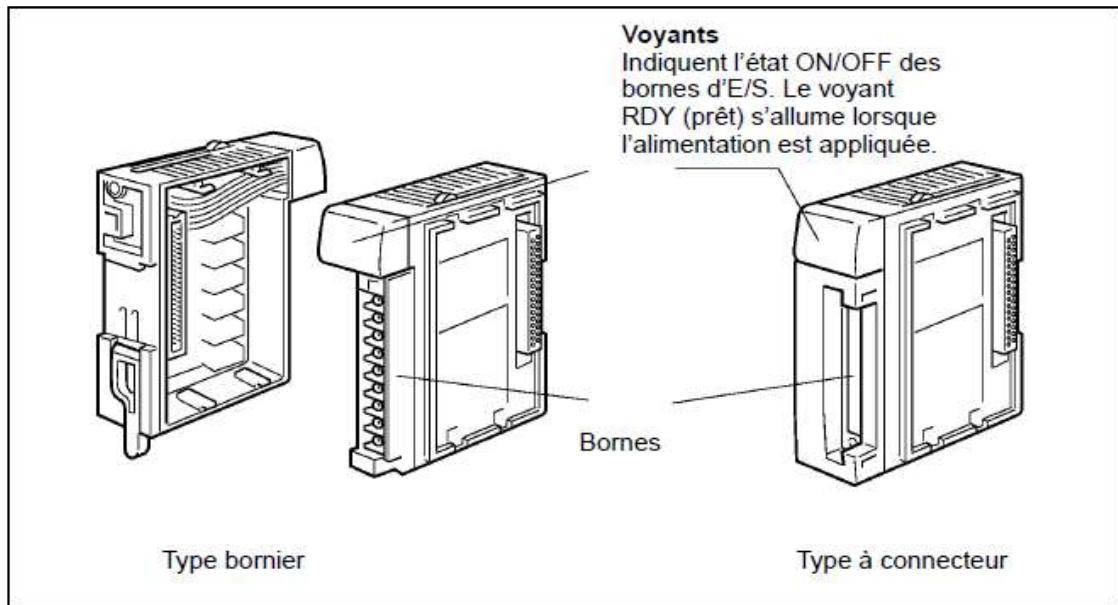


Figure 3.11 : composant de bas des cartes d'E/S.

3.3.2 L'IHM NS 12

L'IHM c'est le deuxième élément essentiel pour notre application, et notre choix c'est porté sur le NS 12 du même constructeur OMRON, il appartient à la nouvelle génération de terminaux OMRON à écrans tactiles, il existe d'autres formats dans cette gamme tel que NS 5", 8", 10".



Figure 3.12 : NS 12 d'OMRON.

Ce modèle offre 256 couleurs, deux connexions USB pour le téléchargement ou l'impression et des connexions supplémentaires pour les cartes Ethernet, Controller Link et vidéo (voir figure 3.13). Le NS à l'avantage d'être configuré et mis en service rapidement grâce à l'utilisation des Smart Active Parts (SAP) uniques d'OMRON. Les Smart Active Parts sont des composants logiciels graphiques préprogrammés et pré testés qui contiennent un code de communication intégré. Ils permettent de créer une application IHM en toute simplicité, à l'aide de la fonction " glisser-déplacer ". En plus on peut citer les caractéristiques suivantes :

- Lisibilité parfaite et basculement rapide entre les écrans.
- Durée de vie extrêmement longue du rétro-éclairage (jusqu'à 75 000 heures).
- Prise en charge de toutes les langues européennes, asiatiques et cyrilliques.
- Enregistrement facile des données sur compact flash.
- Taille de mémoire importante (60 Mo).

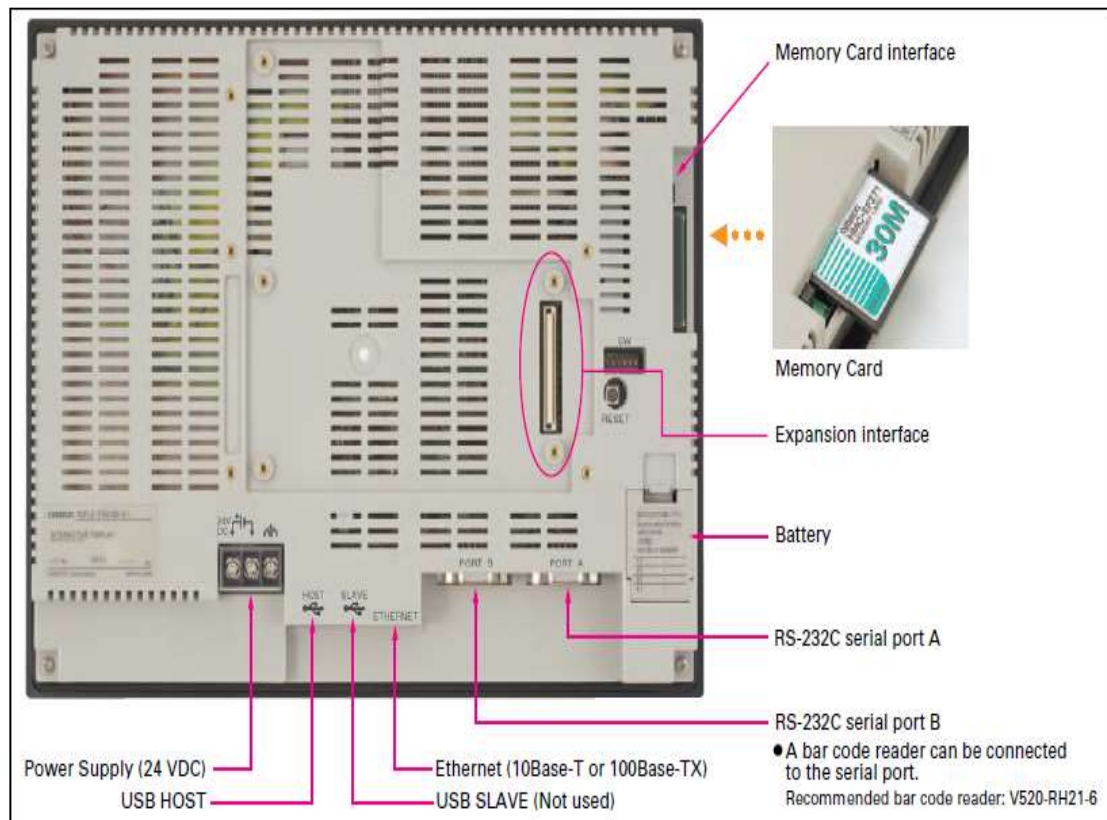


Figure 3.13 : Vue côté connecteur du NS12.

Les terminaux NS sont idéaux à la fois pour le contrôle universel de machines uniques, et pour le fonctionnement de systèmes en réseaux. La série NS amène le traitement de l'information avancé au niveau de l'usine, de manière simple et fiable. Les protocoles NT-Link 1:1, NT-Link 1:N assurent des performances de communications impressionnantes sur les réseaux OMRON.

Pour accéder au menu de configuration du terminal, il faut appuyer sur deux angles de l'écran (voir figure 3.14).



Figure 3.14 : la manière d'ouvrir le menu de notre IHM.

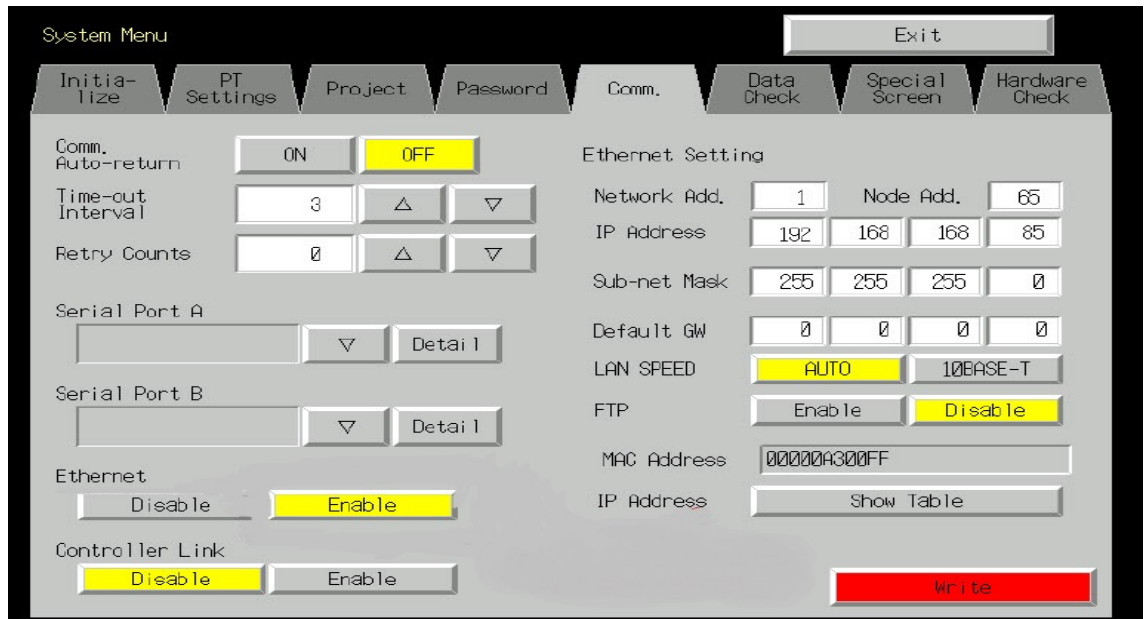


Figure 3.15 : le menu de l'écran NS.

Les terminaux à écrans tactiles offrent une plus grande flexibilité pour la programmation (une cellule tactile peut permettre d'accéder à n'importe quelle adresse d'un API) et une plus grande simplicité d'utilisation. Plus d'instructions difficiles à suivre pour expliquer l'utilisation des différentes touches de fonction – il suffit de dessiner la zone sensible et d'en spécifier l'utilisation (changement d'un bit dans l'API, changement d'écran, saisie d'une valeur dans le système de contrôle, etc.) avec un logiciel simple. Beaucoup d'utilisateurs utilisent l'augmentation de la taille de l'écran pour augmenter les fonctionnalités du terminal en ajoutant plus d'informations pour l'utilisateur, de l'aide, des images et des informations de support.

Avec les améliorations récentes, ces terminaux sont maintenant accessibles, à des prix économiques.

3.3.3 Le variateur de vitesse VS mini J7

C'est l'un des équipements importants utilisé dans notre application pour le contrôle de la vitesse de déplacement du tapis. Dans ce qui suit on va donner un bref descriptif du VS mini J7. [22]

Le VS mini J7 est simple, compact, et plus convivial à utiliser que les modèles conventionnels. (figure 3.15)

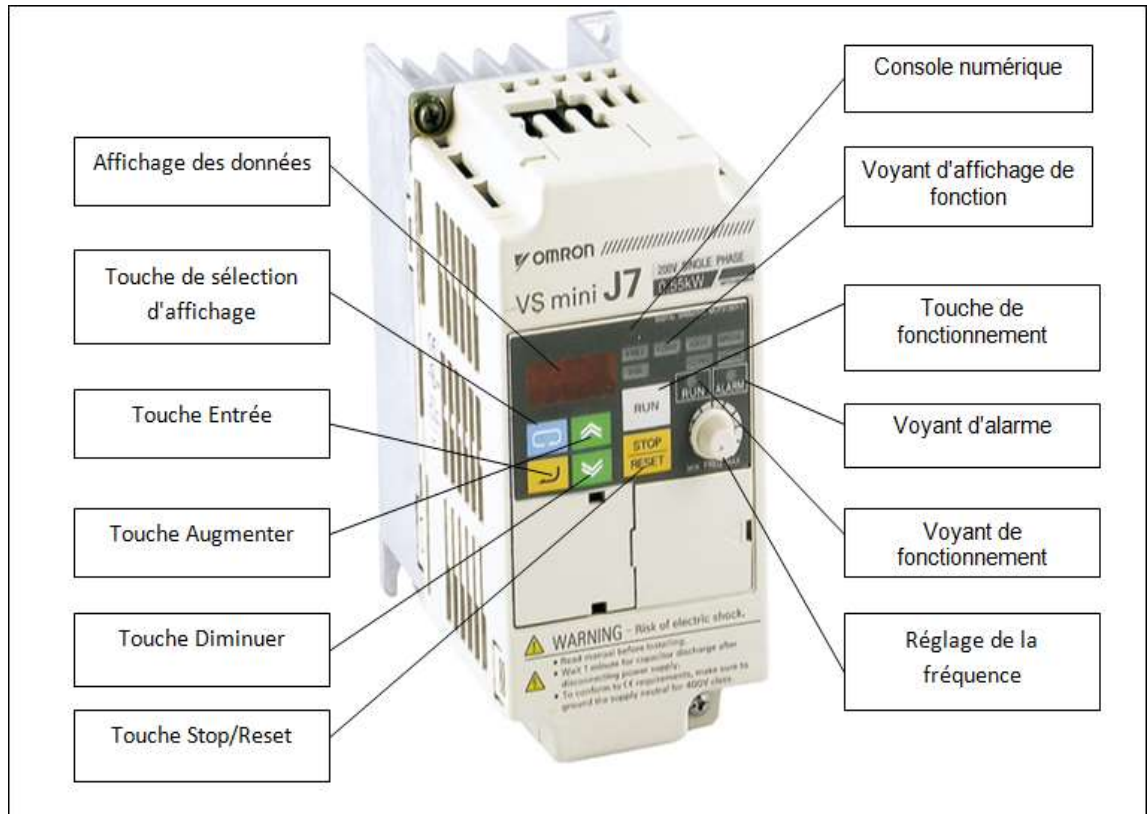


Figure 3.16 : Variateur de vitesse mini J7.

Le VS mini J7 existe sous formes de plusieurs modèles selon la puissance et la tension du réseau électrique mis en jeu. (tableau 3.1) . Notre choix c'est porté sur le modèle CIMR-J7AZ20P7.

Tension nominale	Structure de protection	Puissance moteur maximum applicable (kW)	Modèle
Triphasé 200 Vc.a.	Modèles à montage sur panneau (conformes IP20)	0,1	CIMR-J7AZ20P1
		0,25	CIMR-J7AZ20P2
		0,55	CIMR-J7AZ20P4
		1,1	CIMR-J7AZ20P7
		1,5	CIMR-J7AZ21P5
		2,2	CIMR-J7AZ22P2
		4,0	CIMR-J7AZ24P0
Monophasé 200 Vc.a.	Modèles à montage sur panneau (conformes IP20)	0,1	CIMR-J7AZB0P1
		0,25	CIMR-J7AZB0P2
		0,55	CIMR-J7AZB0P4
		1,1	CIMR-J7AZB0P7
		1,5	CIMR-J7AZB1P5
Triphasé 400 Vc.a.	Modèles à montage sur panneau (conformes IP20)	0,37	CIMR-J7AZ40P2
		0,55	CIMR-J7AZ40P4
		1,1	CIMR-J7AZ40P7
		1,5	CIMR-J7AZ41P5
		2,2	CIMR-J7AZ42P2
		4,0	CIMR-J7AZ44P0

Tableau 3.1 : les différents modèles de VARISPEED J7. [22]

Sur la face avant du VS mini J7 on trouve la console de programmation à travers laquelle on change les paramètres de fonctionnement. Pour plus d'information le tableau ci-dessous résume l'utilisation de la console. [22]

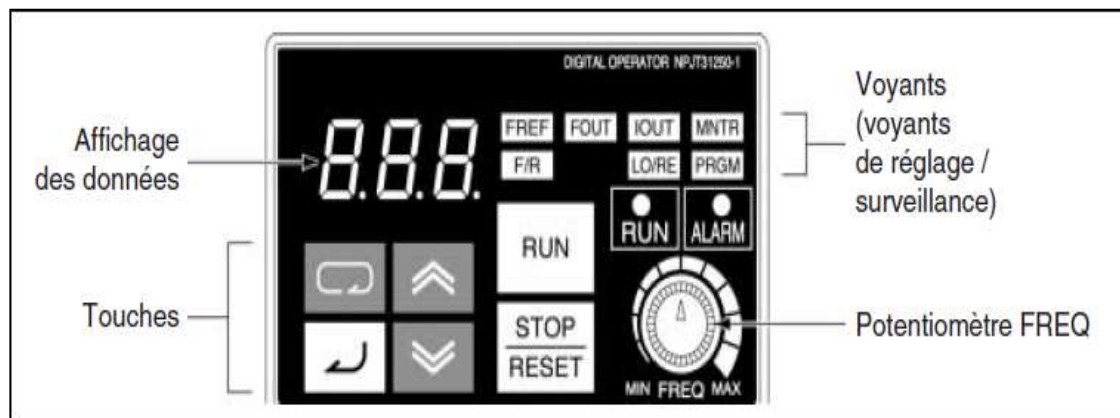


Figure 3.17 : console numérique du VS mini J7. [22]
















Elément	Nom	Fonction
	Affichage des données	Affiche les données pertinentes, telles la fréquence de référence, la fréquence de sortie et les points de consigne des paramètres.
	Potentiomètre FREQ	Définit la fréquence de référence dans une plage entre 0 Hz et la fréquence maximale.
	Voyant FREF	La fréquence de référence peut être surveillée ou définie quand ce voyant est allumé.
	Voyant FOUT	La fréquence de sortie du variateur peut être surveillée quand ce voyant est allumé.
	Voyant IOUT	Le courant de sortie du variateur peut être surveillé quand ce voyant est allumé.
	Voyant MNTR	Les valeurs définies dans U01 à U10 sont surveillées quand ce voyant est allumé.
	Voyant F/R	Le sens de rotation peut être sélectionné quand ce voyant est allumé pendant le fonctionnement du variateur via la touche RUN.
	Voyant LO/RE	Le fonctionnement du variateur via la console numérique ou selon les paramètres définis peut être sélectionné quand ce voyant est allumé. Remarque L'état de cet indicateur ne peut être surveillé que lorsque le variateur fonctionne. Toute entrée de commande RUN est ignorée quand ce voyant est allumé.
	Voyant PRGM	Les paramètres dans n01 à n79 peuvent être définis ou surveillés quand ce voyant est allumé. Remarque Pendant le fonctionnement du variateur, les paramètres peuvent seulement être surveillés et il n'est possible de modifier que certains paramètres. Toute entrée de commande RUN est ignorée quand ce voyant est allumé.
	Touche Mode	Commute les voyants de réglage et de surveillance de manière séquentielle. Le paramètre en cours de réglage est annulé en cas d'activation de cette touche avant l'entrée du paramètre.
	Touche Augmenter	Augmente les numéros de surveillance multifonction, les numéros de paramètres et les points de consigne des paramètres.
	Touche Diminuer	Diminue les numéros de surveillance multifonction, les numéros de paramètres et les points de consigne des paramètres.
	Touche Entrée	Entre les numéros de surveillance multifonction, les numéros de paramètres et les valeurs de données internes après leur définition ou modification.
	Touche RUN	Démarre le variateur quand le J7AZ fonctionne avec la console numérique.
	Touche STOP/RESET	Arrête le variateur, sauf si le paramètre n06 est défini pour désactiver la touche STOP. Fait office de touche de réinitialisation en cas d'erreur du variateur (voir remarque).

Tableau 3.2 : Disruption de la console numérique.

❖ Remarque

Pour des raisons de sécurité, la réinitialisation est impossible lors de l'application d'une commande RUN (avant ou arrière). Il faut attendre que la commande RUN soit désactivée avant de réinitialiser le variateur. [22]

La disposition des bornes du circuit de contrôle du variateur est comme suit :

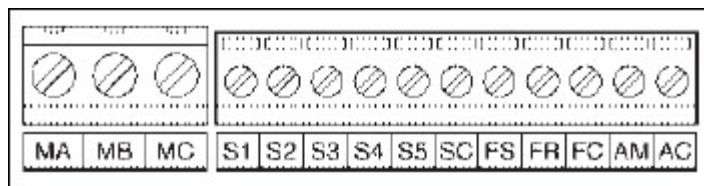


Figure 3.18 les bornes du circuit de contrôle.

Symbole	Nom	Fonction	Niveau du signal	
Entrée	S1	Avant / arrêt	Avant en position ON. Arrêt en position OFF.	
	S2	Entrée multifonction 1 (S2)	Définie par le paramètre n36 (arrière / arrêt)	
	S3	Entrée multifonction 2 (S3)	Définie par le paramètre n37 (réinitialisation erreur)	
	S4	Entrée multifonction 3 (S4)	Définie par le paramètre n38 (erreur externe :	
	S5	Entrée multifonction 4 (S5)	Définie par le paramètre n39 (référence à pas multiples 1)	
	SC	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	20 mA à 12 Vc.c.
	FR	Entrée de fréquence de référence	Borne d'entrée pour la fréquence de référence	0 à 10 Vc.c. (impédance d'entrée : 20 kΩ)
	FC	Commun de fréquence de référence	Commun pour l'utilisation de la fréquence de référence	
Sortie	MA	Sortie de contact multifonction (normalement ouverte)	Sortie relais 1 A max. à 30 Vc.c. 1 A max. à 250 Vc.a.	
	MB	Sortie de contact multifonction (normalement fermée)		
	MC	Commun de sortie de contact multifonction		Commun pour l'utilisation de MA et MB
	AM	Sortie console analogique	Définie par le paramètre n44 (fréquence de sortie)	2 mA max. à 0 à 10 Vc.c.
	AC	Commun de sortie console analogique	Commun pour l'utilisation AM	

Tableau 3.3 : Disruptives des bornes du circuit de contrôle.

3.3.4 L'encodeur (réf : RE5402508 /24, constructeur : elap)

L'encodeur rotatif utilisé dans notre application appartient au constructeur « elap »

Très connue dans le secteur industriel, dont voici un bref aperçu.

Les encodeurs électromécaniques, linéaires ou rotatifs, génèrent un signal électrique selon la position ou le déplacement. Les rotatifs délivrent une

information d'angle en mesurant la rotation effectuée autour d'un axe. Plus le codeur rotatif tourne lentement, plus la déduction de vitesse perd en précision.

Il existe 2 principaux types :

- ❖ Le codeur rotatif incrémental : Qui ajoute ou soustrait (selon le sens de rotation) une unité à un compteur à chaque rotation supérieure à la résolution du capteur. Le compteur est généralement remis à zéro lorsque l'appareil est allumé.
- ❖ Le codeur rotatif absolu : Qui intègre son propre compteur. Ce genre de capteur est généralement calibré et initialisé une seule fois, et il conserve normalement sa valeur lors de l'arrêt de l'appareil.



Figure 3.19 : Encodeur rotatif absolue.

3.3.5 Les cellules photo électrique

Et enfin parmi les composants essentiels utilisés dans notre application, nous citons les détecteurs de position infrarouge. Là aussi on a choisie le constructeur OMRON, dans ce qui suit nous expliquons le principe de fonctionnement et les différents modèles de ce type de capteur. [21]

Un capteur photoélectrique est un capteur de proximité. Il se compose d'un émetteur de lumière associé à un récepteur. La détection d'un objet se fait par coupure ou variation d'un faisceau lumineux. Le signal est amplifié pour être exploité par la partie commande.

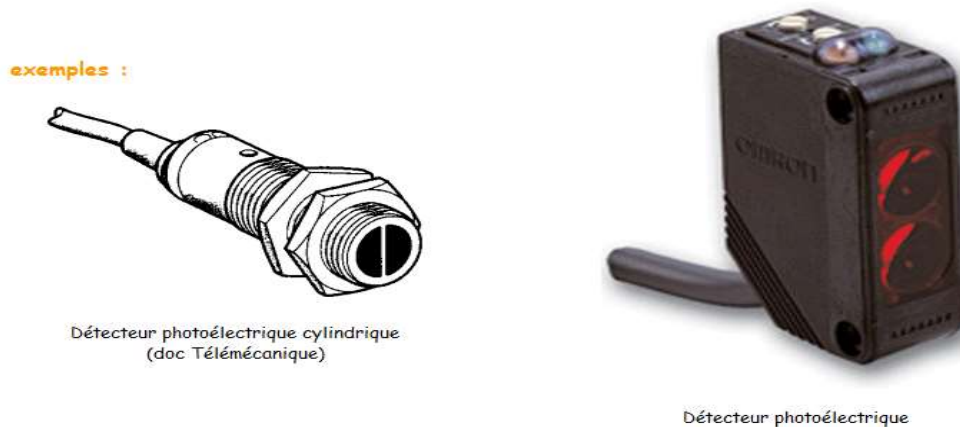


Figure 2.20 : modèles de capteurs infrarouges.

❖ Les différents types de détection :

Il existe trois grands types de détection :

- *La détection par barrage* où l'objet à détecter coupe un faisceau lumineux situé entre l'émetteur et le récepteur,
- *La détection par barrage* où un faisceau réfléchi est coupé par l'objet à détecter,
- *Le système réflex* où le faisceau émis par le récepteur est renvoyé par la pièce à détecter sur le récepteur situé sur le même capteur.

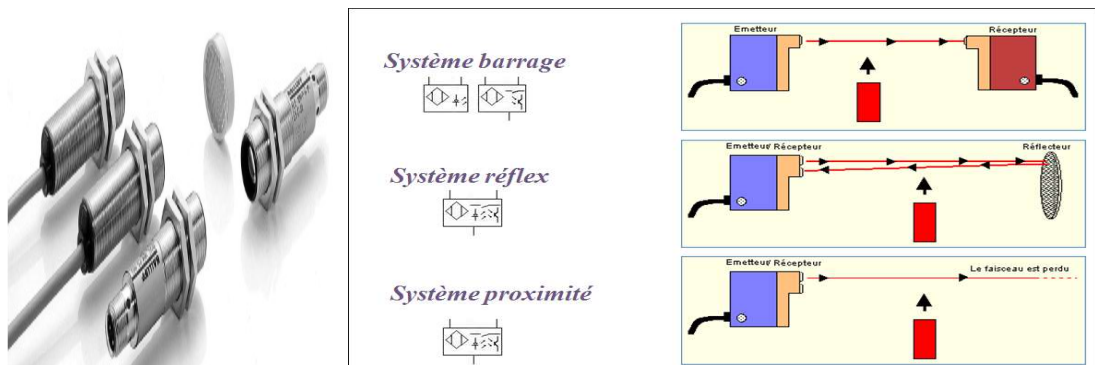


Figure 3.21 : les trois types des capteurs à proximité.

3.4 Programmation de l'API CQM1H et l'IHM NS12

3.4.1 Développement du programme de L'API CQM1H par le logiciel CX Programmer

Après avoir installé le Pack logiciel CX One, on ouvre le CX Programmer pour la programmation de l'API CQM1H qu'on a choisis. Tout d'abord nous allons présenter ce logiciel et les outils nécessaires pour pouvoir démarrer et effectuer une programmation d'un projet.

a Présentation du Logiciel

CX-Programmer fournit une plate-forme logicielle API commune pour tous les types d'automates programmables OMRON, des micro-API aux systèmes à processeurs en duplex. Il permet une conversion facile et la réutilisation de code API entre différents types d'API ainsi que la réutilisation totale des programmes de contrôle créés avec les logiciels de programmation d'API précédents.

Les tâches de bases qu'il offre à son utilisateur lors de la création d'une solution d'automatisation sont :

- ✓ La création et gestion de projets.
- ✓ La configuration et le paramétrage du matériel et de la communication.
- ✓ La gestion des mnémoniques.
- ✓ La création des programmes.
- ✓ Le chargement de programmes dans les systèmes cibles.
- ✓ Le test de l'installation d'automatisation.
- ✓ Le diagnostic lors des perturbations dans l'installation.

1) CRÉATION ET GESTION DE PROJETS :

Lorsque nous ouvrons le logiciel la fenêtre suivante apparaît :

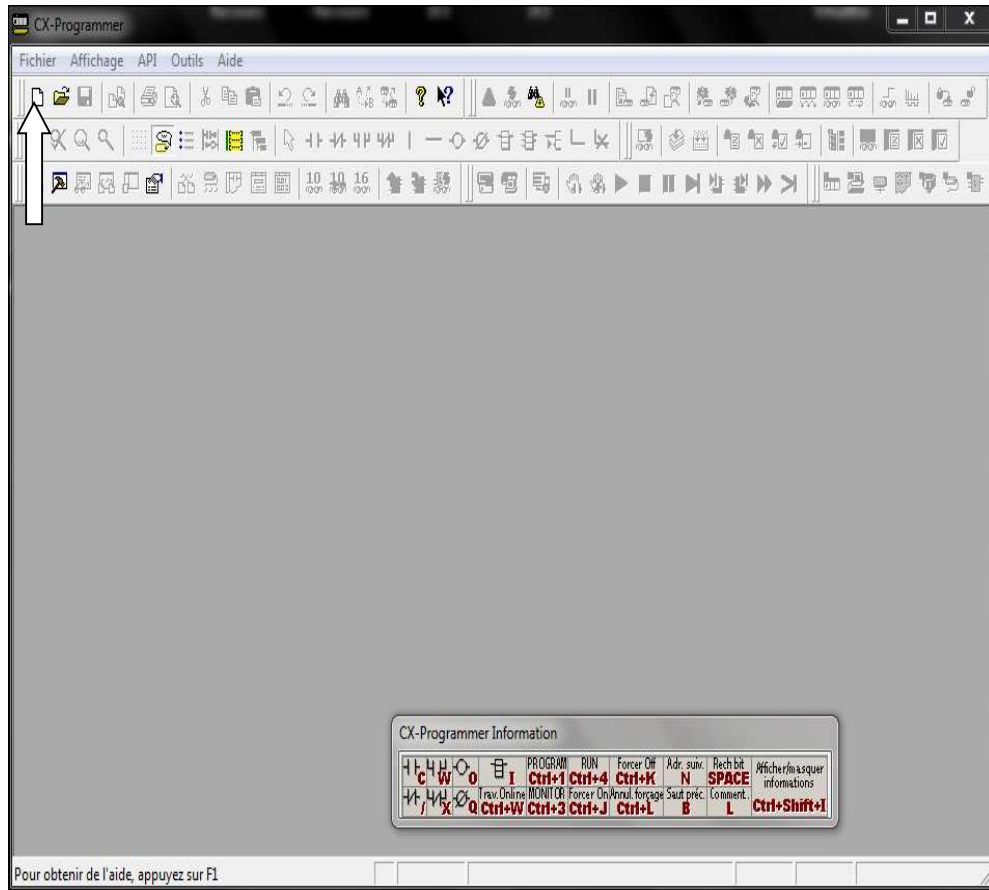



Figure 3.22 : fenetre d'acceuil du CX Programmer.

Pour crée un nouveau projet cliquer sur le bouton  dans la barre des outils, ou cliquer sur Fichier puis nouveau.

Une fenêtrre apparaît du choix de nom, type d'API ainsi le type du réseau.

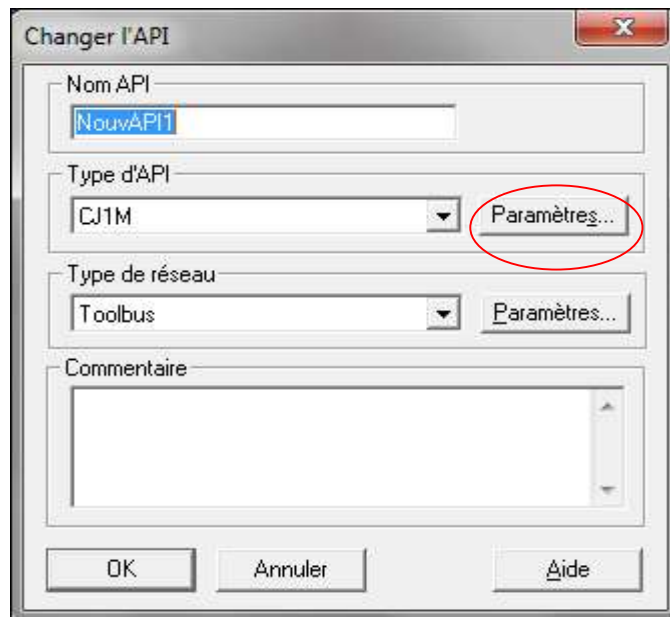


Figure 3.23 : fenêtre du choix d'API.

Le bouton Paramètres du type API permet de choisir le type d'UC (CPU) selon le CPU du matériel utilisé.

Dans cette version le type de réseau change automatiquement avec l'API choisie.

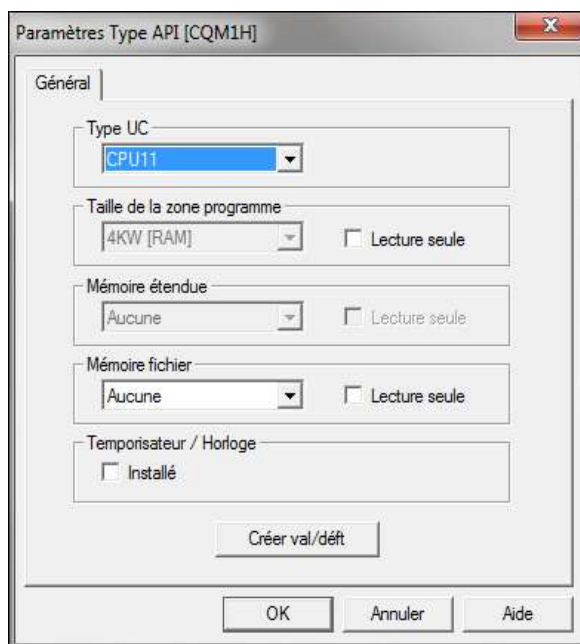



Figure 3.24 : fenêtre Paramètres type API.

En fin nous validons notre choix en cliquant sur OK.

✚ La configuration choisie pour notre travail est :

- Types d'API : CQMH,
- Paramètre du type : CPU 61,
- Type de réseau : SYSMAC WAY.

On enregistre le programme en cliquant sur Fichier → enregistré ou par le raccourci qui se trouve dans l'icône .

Après la validation du choix .la fenêtre du travail apparait (voir figure 3.24).

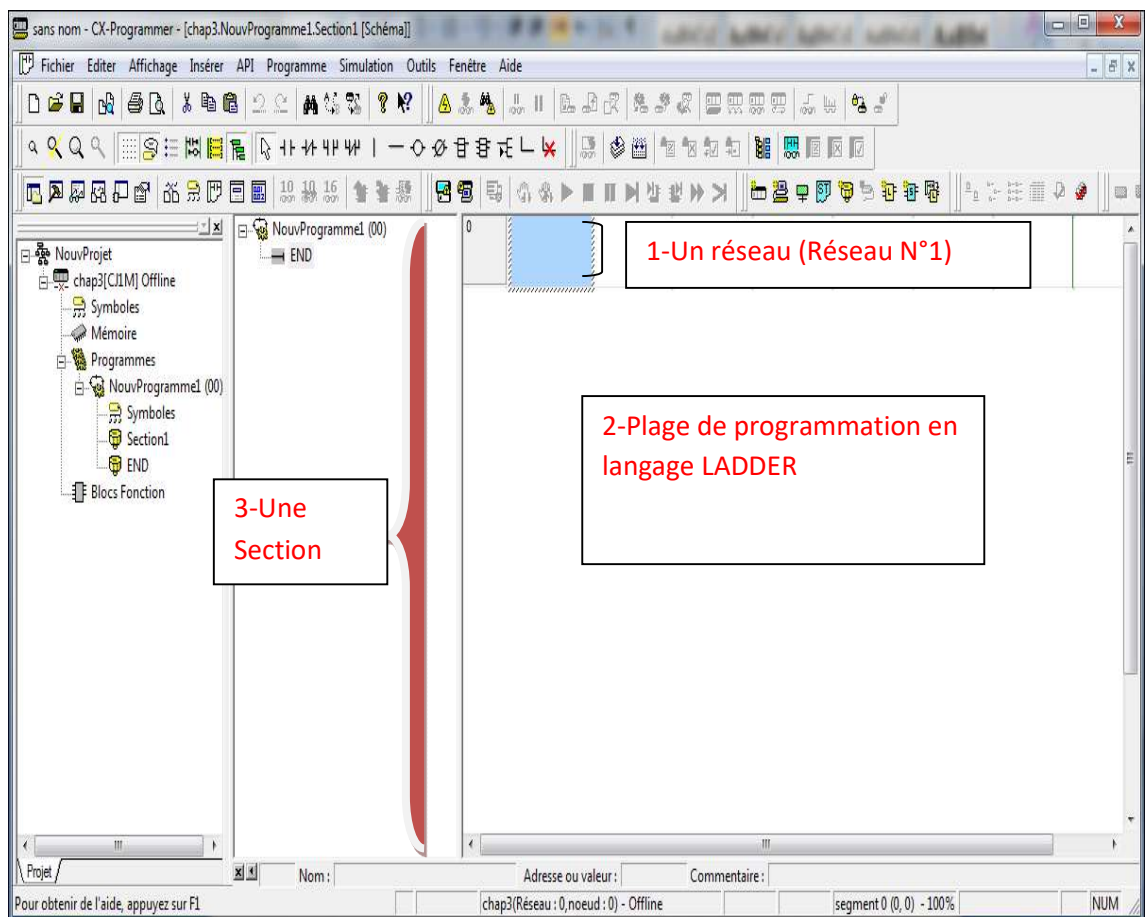


Figure 3.25 : la fenêtre de programmation.

✚ La fenêtre du travail contient :

1. Un réseau (Réseau N°1) : Un réseau est une partie d'une section où ce derniers est formée de plusieurs réseaux.
2. Plage de programmation en langage LADDER.
3. Une Section :

Pour la gestion plus facile des grands programmes, un programme peut être divisé en un certain nombre de sections, nommées définissables. Une section est comme un chapitre dans un livre - l'automate balaye les sections dans l'ordre.

Une liste des sections dans un programme est affichée dans l'espace de travail de projet sous le nom de programme.

✚ L'onglet du Projet contient :

Les Symboles globaux, les paramètres d'API, les instructions étendues, Mémoire et les programmes qui contiennent aussi les Symboles locaux et des sections.

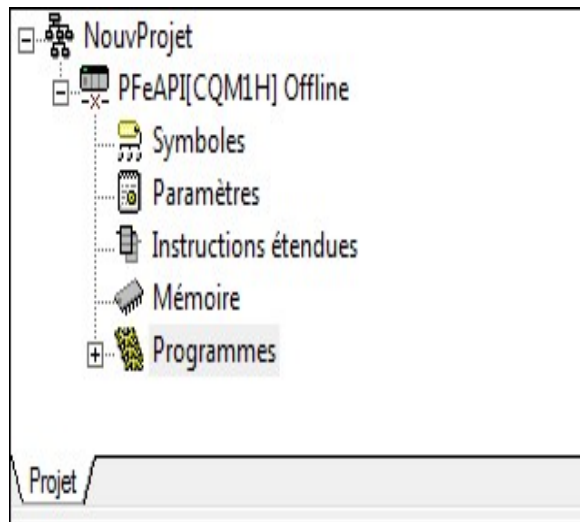



Figure 3.26 : L'onglet du projet.

 Description des éléments des fenêtres de l'onglet du projet :

❖ **Symboles :**

Il est possible de définir les symboles d'un projet pour chaque API, et à deux niveaux différents :

1. Symboles Globaux

Disponibles pour tous les programmes d'un API spécifique. CX-Programmer insère ici un ensemble de symboles prédéfinis en fonction du type d'API.

2. Symboles Locaux

Disponibles seulement dans un programme d'API particulier, c'est-à-dire propres au programme.

Nom	Type de don...	Adresse / Valeur	Emplaceme...	Utilis...	Commentaire
∧ P_1min	BOOL	254.00			Bit impulsion horloge 1 minute
∧ P_1s	BOOL	255.02			Bit impulsion horloge 1,0 seconde
∧ P_CY	BOOL	255.04			Drapeau Retenue (CY)
∧ P_Cycle_Time_Error	BOOL	253.09			Drapeau Erreur temps de cycle
**** P_Cycle_Time_Value	UINT_BCD	AR27			Temps de scrutation actuel
∧ P_EQ	BOOL	255.06			Drapeau Egalité (EQ)
∧ P_ER	BOOL	255.03			Drapeau Erreur exécution instructi...
∧ P_First_Cycle	BOOL	253.15			Drapeau Premier cycle
∧ P_GT	BOOL	255.05			Drapeau Plus que (GT)
**** P_Hour_Date	UINT_BCD	AR19			Heures (00-07) & Date (08-15)
∧ P_Low_Battery	BOOL	253.08			Drapeau Batterie faible
∧ P_LT	BOOL	255.07			Drapeau Moins que (LT)
**** P_Max_Cycle_Time	UINT_BCD	AR26			Temps de cycle maximum
**** P_Month_Year	UINT_BCD	AR20			Mois (00-07) & Année (08-15)
∧ P_OF	BOOL	254.04			Drapeau Dépassement (OF)
∧ P_Off	BOOL	253.14			Drapeau Toujours OFF
∧ P_On	BOOL	253.13			Drapeau Toujours ON
∧ P_Output_Off_Bit	BOOL	252.15			Bit sortie OFF
**** P_Sec_Min	UINT_BCD	AR18			Secondes (00-07) & Minutes (08-1...
∧ P_Step	BOOL	254.07			Drapeau Pas
∧ P_UF	BOOL	254.05			Drapeau Dépassement négatif (UF)

Tableau 3.4 : Symboles Globaux.

❖ Paramètres d'API :

Les API comportent généralement des options internes qui régissent leur comportement et sont importantes pour le bon déroulement des programmes. Le composant Paramètres API permet de visualiser et de modifier la configuration de l'API. Le paramétrage de l'API peut être enregistré dans un projet CX-Programmer et transféré avec le ou les programmes de l'API.

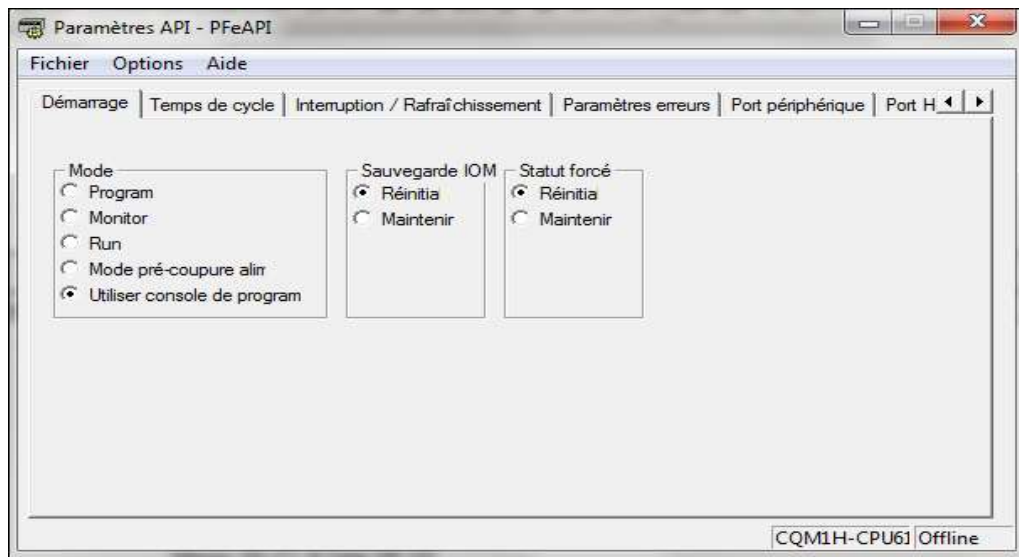


Figure 3.27 : Paramètres d'API.

❖ Instructions étendues

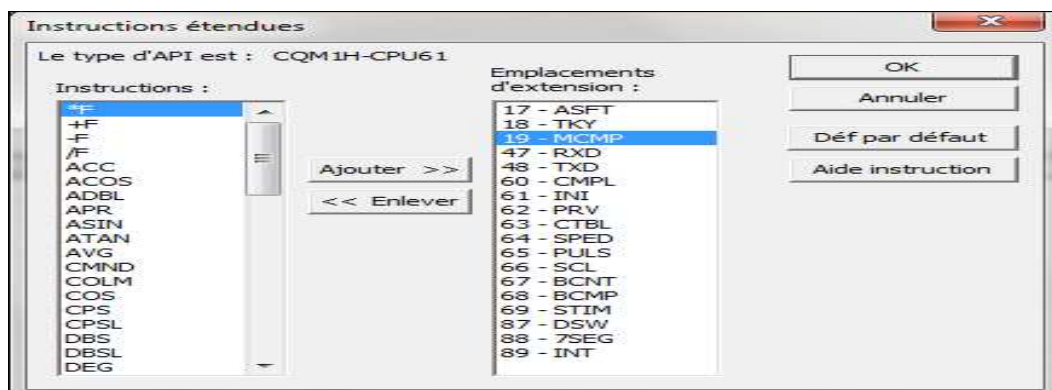


Figure 3.28 : instruction étendue.

❖ Mémoires

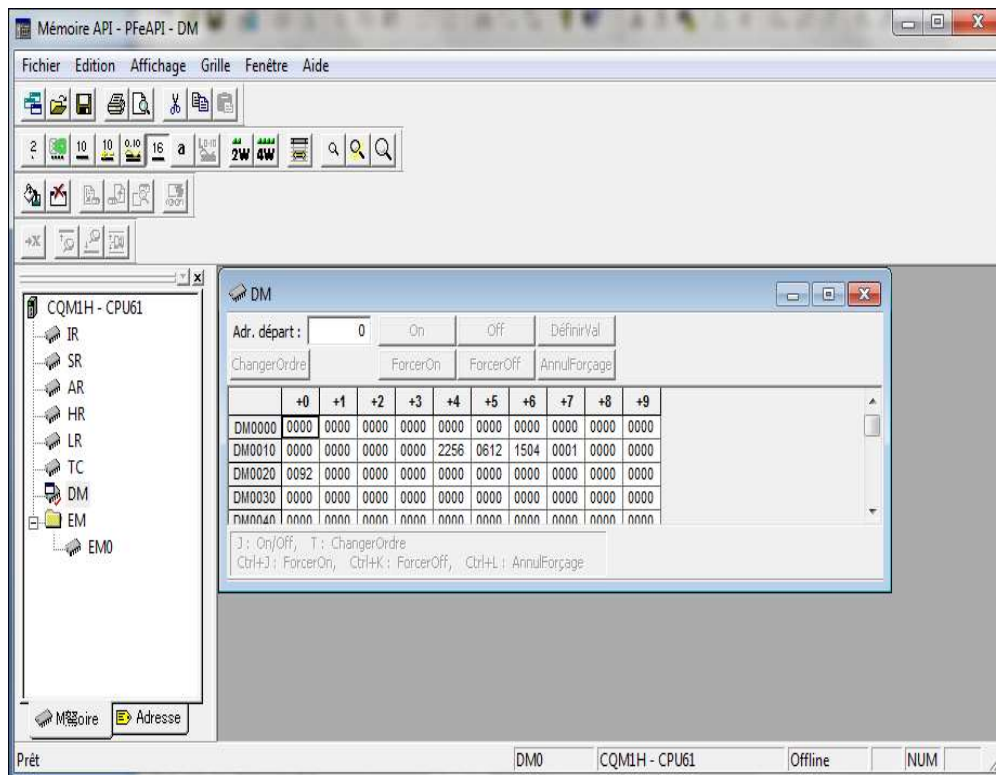
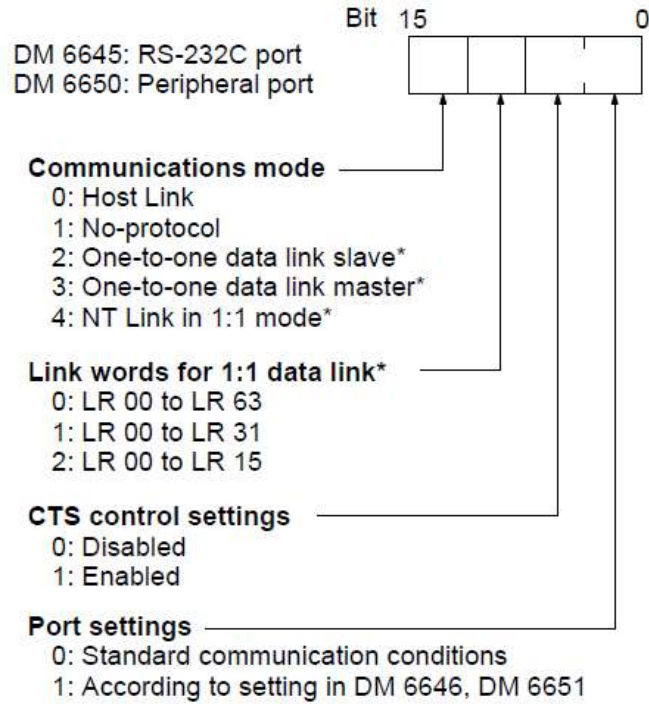


Figure 3.29 : Mémoires de l’API.

Cette fenetre donne l'accé aux différentes zones mémoire de l’API et en particulier DM (Data Memory) et EM (Extented Memory) (tableau 3.5), c’est une zone mémoire qui contient les données de calcul, de transfert, et surtout de configuration . La zone de confuguration est extremement sensible car une erreur de configuration risque de bloquer définitivement l’API.

Une partie de cette zone peut être lue ou écrit à partir du programme de l’application , la zone qui concerne la configuration ne peut être ni lue ou ni écrit par le programme, pour pouvoir les modifier il faut utiliser un péripherique de programmation (tableau 3.5).

- ❖ **DM6645** « réservé a la communication via le port RS232 », on lui a affecté la valeur "4001" .

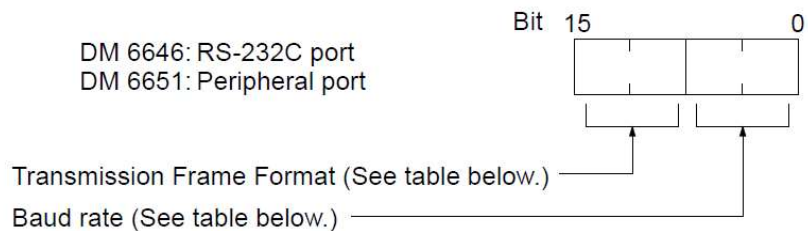


Default (0000): Host Link using standard parameters, no CTS control

Note *These settings can be made for the RS-232C port (DM 6645), but not for the peripheral port (DM 6650).

C'est-à-dire le port RS232 de l'API est configurer pour communiquer en mode NT link 1 :1, de cette manière l'API peut communiquer avec l'IHM

- ❖ **DM6646** « réservé pour fixer la vitesse de communication », on lui a affecté la valeur "0304".



Default: Standard communication conditions.

Transmission Frame Format

Setting	Stop bits	Data length	Stop bits	Parity
00	1	7	1	Even
01	1	7	1	Odd
02	1	7	1	None
03	1	7	2	Even
04	1	7	2	Odd
05	1	7	2	None
06	1	8	1	Even
07	1	8	1	Odd
08	1	8	1	None
09	1	8	2	Even
10	1	8	2	Odd
11	1	8	2	None

Tableau 3.6 : Format de trame de transmission.

Baud Rate

Setting	Baud rate
00	1,200 bps
01	2,400 bps
02	4,800 bps
03	9,600 bps
04	19,200 bps

Tableau 3.7 : la vitesse de transmission du trame.

C'est-à-dire 1 bit de start, longueur 7 bits, 2bits de stop, parité impaire, et 9600 bits/s.

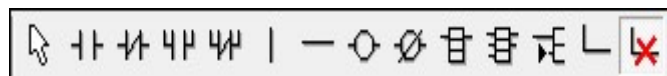
❖ Programmes



Figure 3.30 : programme dans l'onglet projet.





Dans cette zone on peut ajouter ou supprimer des sections et modifier ou créer des nouveaux symboles du programme.

b Les outils et les instructions de Programmation




Les symboles du LADDER :

❖ Les contacte :

- contacte ouvert 
- contact fermé 
- contact OR 
- contact Fermé OR 

❖ Les lignes de connexion : Verticale ou Horizontale 

❖ Les bobines : Bobine ou Bobine fermé 

❖ Les instructions et les Blocs 

c Déclaration des outils

1) Déclaration des contacts et bobines :

Pour éditer un contact, on clique sur le contact qu'on veut utiliser, et on le dépose dans la case du réseau choisie.

La fenêtre suivante apparait pour donner au contact son adresse et le commentaire. Voici un exemple.

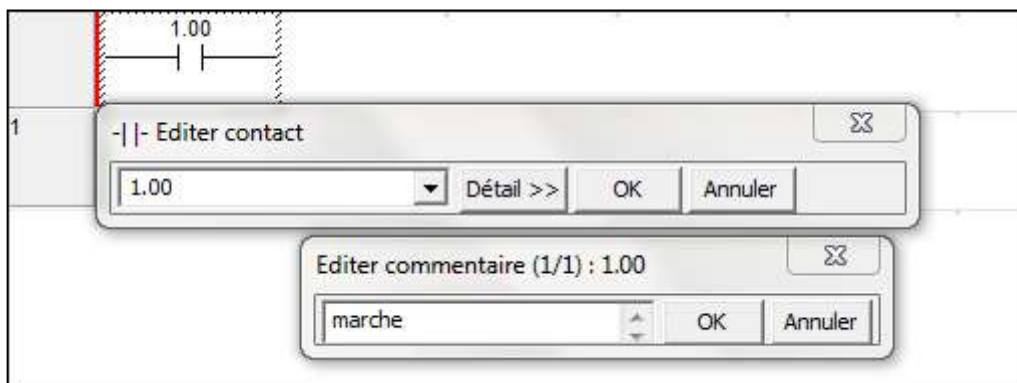


Figure 3.31: exemple d'édition d'un contact.

Et c'est la même chose dans l'édition des bobines de sortie .voici un exemple :

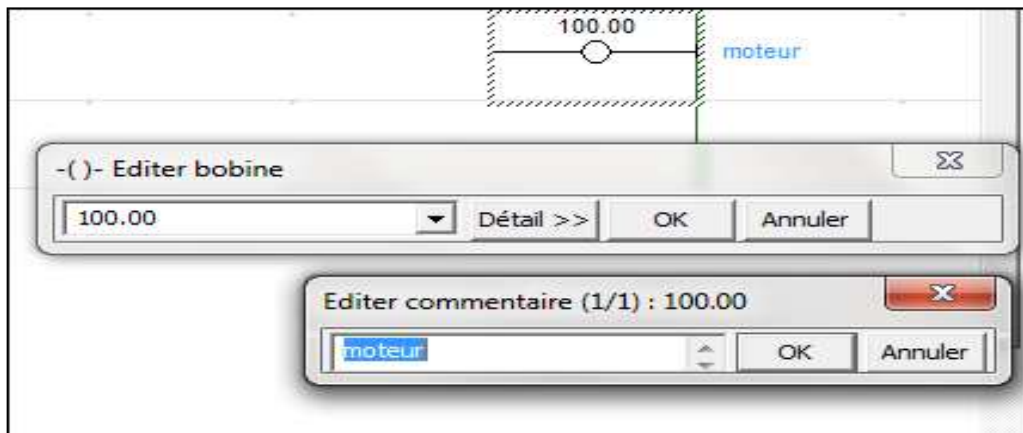


Figure 3.32 : exemple d'édition d'une bobine (sortie).

2) Déclaration des instructions :

L'édition des instructions est comme suit : le nom d'instruction (timer : TIM, diviseur : DIV, déplacement : MOV ...etc.) → l'adresse → le numéro d'instruction → la valeur ou donnée déclaré par #valeur.

Voici quelques exemples des instructions :

- Instruction pour garder l'état : KEEP(11), on écrit seulement l'adresse du bit à garder

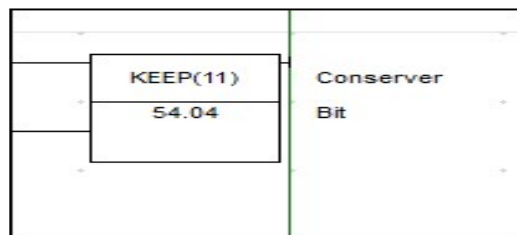


Figure 3.33 : L'instruction KEEP.

- Différentiation Front montant : DIFU(13)

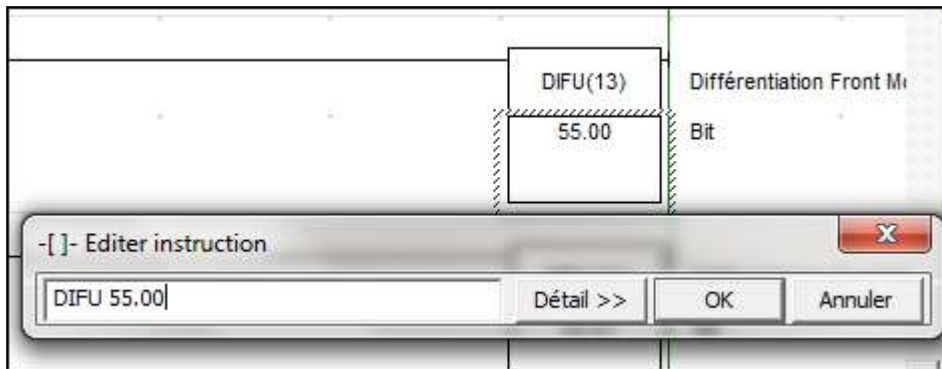


Figure 3.34 : L'instruction Différentiation Front montant : DIFU(13)

- L'instruction de déplacement : MOV(21), elle sert à déplacer le mot source (un bit ou une case mémoire) dans la destination déclaré. (Figure 3.35)

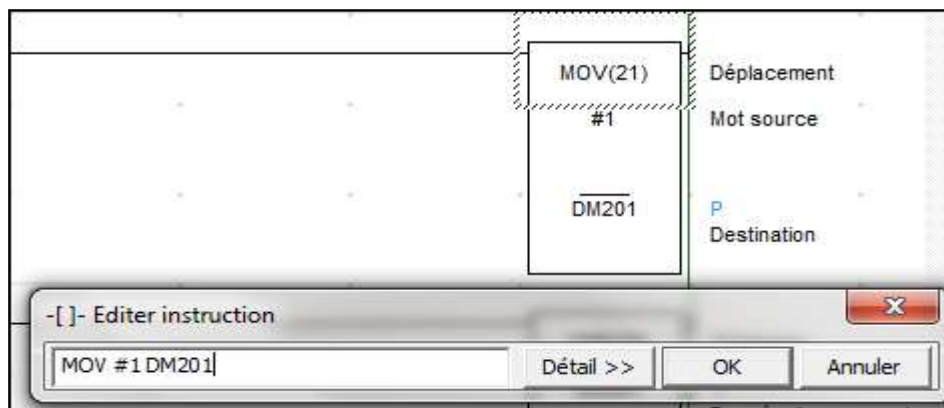


Figure 3.38 : L'instruction de déplacement (MOV).

- L'instruction de comparaison : CMP(20), on écrit le nom : 'CMP 'espace, la 'donné de comparaison', espace la valeur de comparaison' #Valeur'. voir l'exemple dans la figure suivante

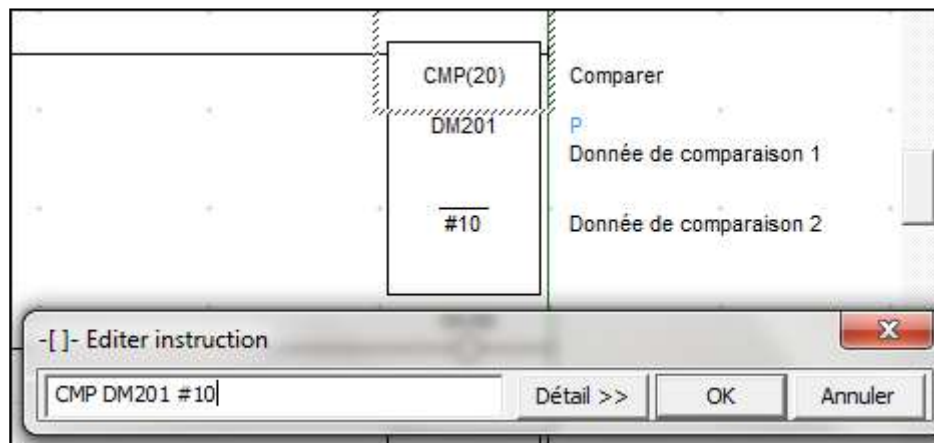


Figure 3.39 : L'instruction de comparaison CMP(20).

d Mode de fonctionnement

Le mode de fonctionnement d'un API ne peut être défini que lorsque l'API est on-line.

Il existe 4 modes de fonctionnement API :

1. Programme,
2. DEBUG,

3. Monitor,
4. RUN.

❖ **Changement du mode**

1. Avant de tenter de changer le mode de fonctionnement d'un API, nous assurons que cet API est on-line.

Si l'API est offline :

Nous cliquons avec le bouton droit de la souris sur l'objet API voulu dans l'espace de travail Projet, puis nous sélectionnons l'option Travail Online du menu déroulant.

Ou Sélectionnons l'option Travail Online du menu API propre à l'API sélectionné.

Après un court instant, la barre d'état et l'icône API de l'espace de travail Projet indiquent le mode de fonctionnement courant de l'API connecté.

2. Il existe plusieurs méthodes de sélection du mode de fonctionnement d'un API :
 - En cliquant sur l'objet API souhaité dans l'espace de travail Projet avec le bouton droit de la souris.
 - A partir du menu API (option Mode d'opération), après avoir sélectionné l'API dans l'espace de travail Projet.
 - A partir de la boîte de dialogue Propriétés de l'API.

Après l'une de ces opérations, les différents modes sont affichés.

3. On sélectionne le mode PROGRAM, DEBUG, MONITOR, ou RUN.

Une boîte de dialogue nous invite à confirmer le changement de mode, qui va affecter le fonctionnement des programmes de l'API.

- Le mode **PROGRAM** d'un API est utilisé pour transférer des programmes vers l'API.
- Le mode **DEBUG** d'un API sert à vérifier l'exécution du programme et le fonctionnement des entrées/sorties. Les points de sortie des modules de sortie restent Off même si le bit de sortie correspondant est On
- En mode **MONITOR**, les programmes de l'API sont exécutés et les entrées/sorties sont actives. Toutefois, vous pouvez écrire dans la mémoire de l'API. En mode MONITOR, nous pouvons :
 - Modifier des données dans toutes les zones de mémoire
 - Effectuer une édition on line de la logique à contacts
 - Modifier les valeurs TIM/CNT
 - Activer ou désactiver des bits (contacts ou bobines), forcer des bits à 1 ou à 0.
- Le mode **RUN** d'un API permet l'exécution du ou des programmes de l'API. Il ne permet pas d'écrire ou de forcer des valeurs dans la mémoire de l'API à partir d'un ordinateur. Par contre, il est possible de lire (surveiller) la mémoire de l'API.

CX-Programmer fait passer l'API dans le mode sélectionné. La barre d'état et l'icône API de l'espace de travail Projet indiquent le nouveau mode de fonctionnement.

-En mode **PROGRAM**, la CPU est arrêtée autorisant ainsi le transfert de tout type de données (programme, table d'E/S)

- En mode **MONITOR**, le programme fonctionne et le forçage de bit, mot, timer, compteur est autorisé.

-En mode **RUN**, le programme fonctionne et aucune variable ne peut être forcée. Il faudra impérativement envoyer une commande FINS pour passer en mode MONITOR.

e Chargement du programme dans L'API ciblés

Pour charger le programme dans l'API, il faut d'abord se connecté avec un câble spécial.

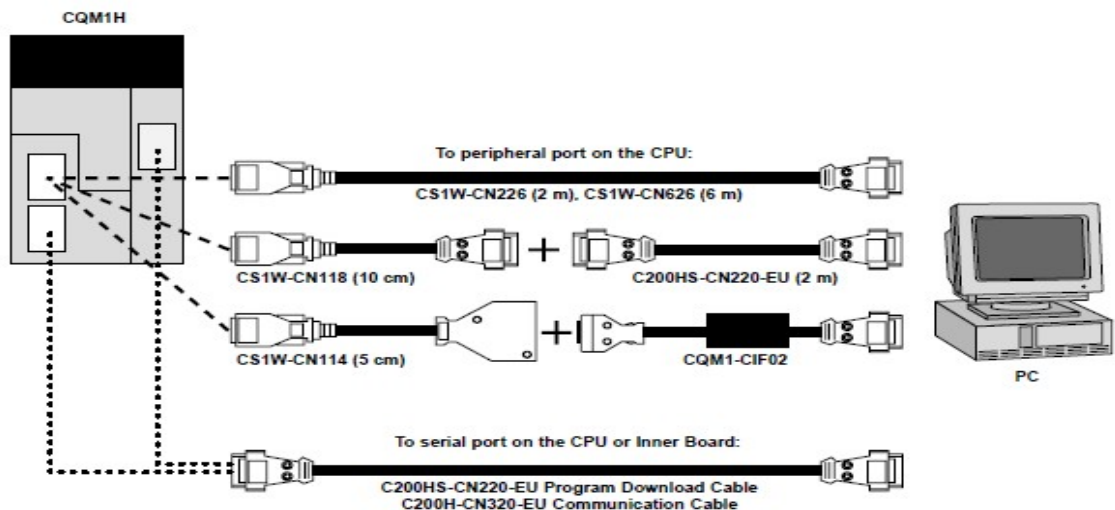


Figure 3.40: les types des câbles qu'on peut utiliser.

Après l'assurance de la communication entre l'API et le PC, le chargement du programme ce fait par les étapes suivantes :

1. Dans la barre d'édition on click sur « API » puis on choisie (Travail Online).
2. Puis dans la même liste on appuie sur (transferts → " Vers API") et pour charger le programme de L'API vers notre PC on choisie →" depuis API". (voir figure 3.37)
3. Le logiciel lance le téléchargement, lorsqu'il termine on click sur ok.

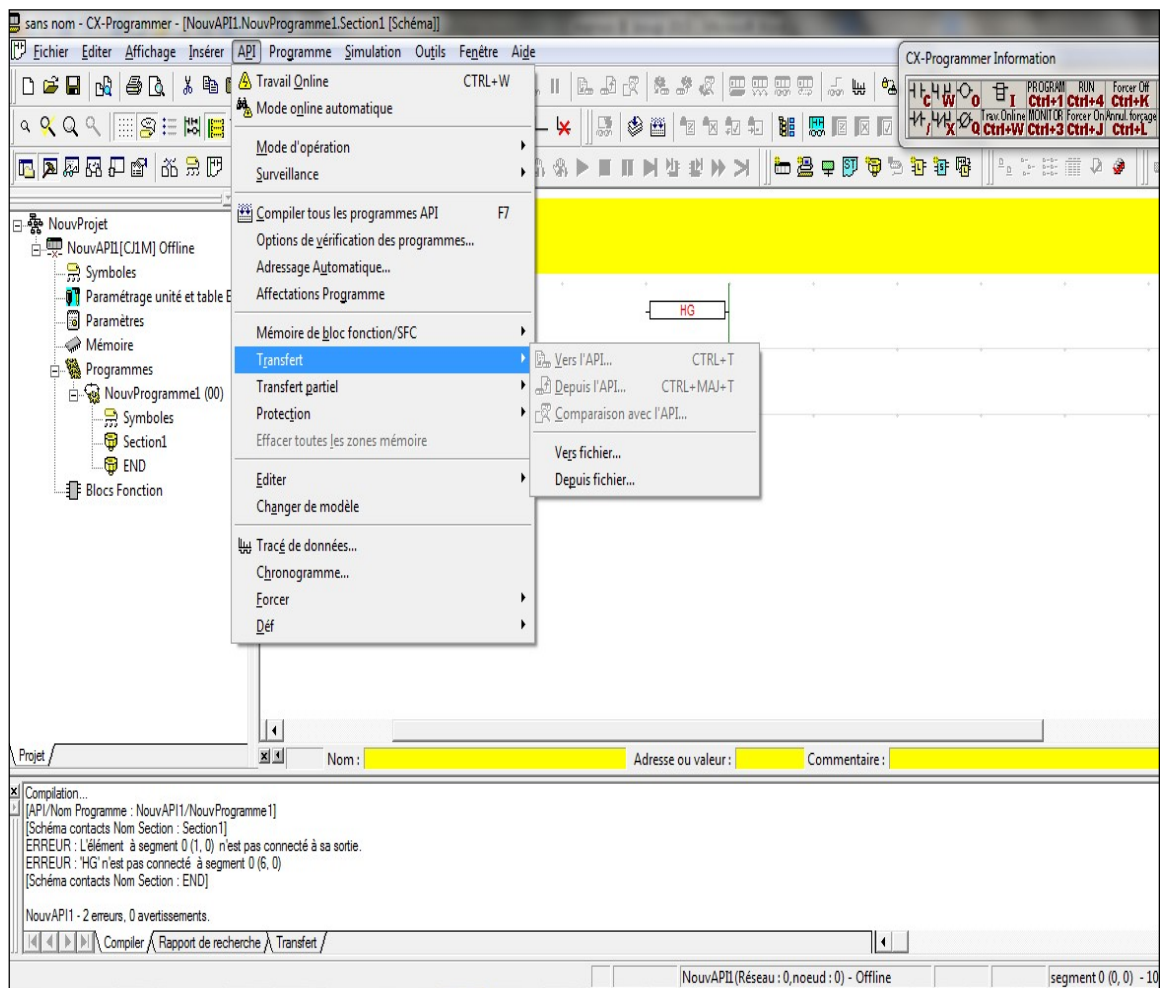


Figure 3.41 : le chemin de chargement du programme dans l'API.

f Le simulateur CX Simulator


Le simulateur des programmes est intégré dans le CX Programmer . Il sert à simuler pas à pas le programme développé après sa compilation.



Figure 3.42 : CX Simulator.

3.4.2 Programmation de L'IHM NS12 par le logiciel CX Designer

CX Designer permet de programmer les IHM de type NS, et c'est le deuxième logiciel qu'on a utilisé pour le développement de notre application.

Ce logiciel de supervision permet le dialogue avec un automate dans le but de visualiser un processus.

Il s'agit d'assembler adroitement matériels et logiciels afin d'obtenir un système capable de :

- _ Connaître toutes les informations délivrées par l'installation ;
- _ Réagir et d'informer l'utilisateur sur les diverses informations le concernant ;
- _ Gérer le fonctionnement de l'ensemble de l'installation.

Ce logiciel va permettre de dialoguer avec l'installation, de surveiller des niveaux d'alarmes, de charger des recettes de fabrication,

CX-Designer sert à créer des données d'écran pour les terminaux programmables NS. CX-Designer peut également contrôler le fonctionnement des données d'écran créées sur l'ordinateur. CX-Designer garantit un processus de développement efficace pour la création et la simulation d'écrans ainsi que le déploiement de projets. Les utilisateurs peuvent développer des écrans de manière plus efficace grâce au logiciel de support convivial. CX-Designer possède environ 1 000 objets fonctionnels standards associés à des graphiques et à des fonctions avancées. De ce fait, même les utilisateurs débutants peuvent créer des écrans en toute simplicité : il leur suffit de disposer les objets fonctionnels sur un écran.

a Création d'un projet

C'est le même principe que le CX Programmer, la fenêtre lors de l'ouverture du logiciel est comme suit (voir figure 3.43) :

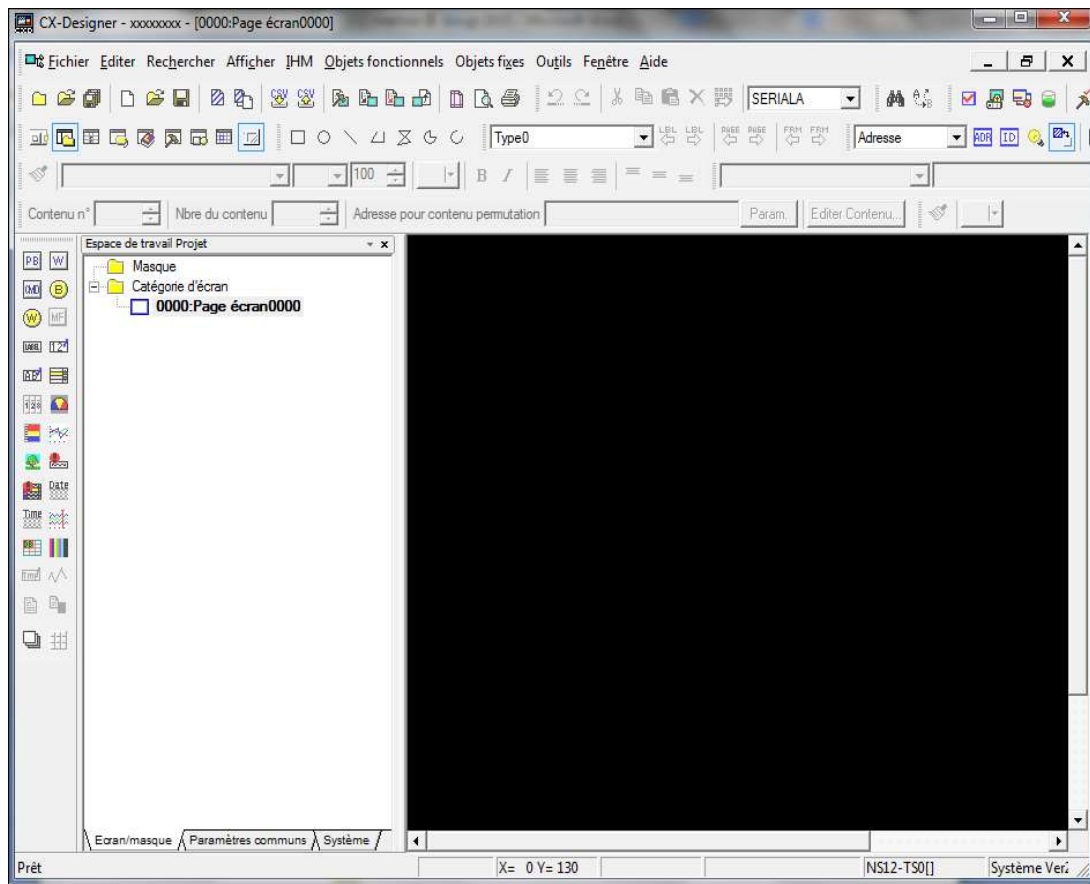








Figure 3.43 : l'espace de travail du CX Designer.






b Les outils de création d'écrans

Les icônes qu'on va les définir c'est des outils pour créer notre interface de contrôle-commande et de surveillance, pour les utiliser il faut cliquer sur l'icône choisie et sélectionner un carré sur la zone de travail puis on fait un double clic pour effectuer le paramètre qu'on va le démontrer par la suite.

Les outils de création de l'écran de travail sont :

- Bouton ON/OFF  Contrôle le statut ON/OFF d'une adresse spécifiée. L'action peut être du type momentané, alterné, SET, ou RESET.
- Bouton "Word"  Ecriture ou incrément/décément d'un mot à l'adresse spécifiée.

- Bouton de commande  Exécution de commande particulière tel que les changements d'écran, affichage vidéo, exécution de macro, etc.
- Voyant Bit  Affiche le statut ON/OFF du bit à l'adresse spécifiée.
- Voyant mot  Affiche une des 10 couleurs suivant la valeur 0 à 9 du mot spécifié.
- Texte  Affichage de texte fixe.
- Affichage/saisie numérique  Affiche et/ou modifie la valeur numérique de l'adresse spécifiée.
- Affichage/saisie texte  Affiche et/ou modifie le texte consigné à l'adresse spécifiée.
- Liste de Sélection  Affiche une liste textes consignés à une adresse spécifiée ou dans un fichier texte. Le texte sélectionné peut être chargé dans une adresse spécifiée.
- Roue codeuse  Affiche/modifie la valeur numérique de l'adresse spécifiée.
- Vue-mètre  Affiche sous forme de vue mètre en cercle, demi-cercle ou quart de cercle 3 couleurs, la valeur numérique de l'adresse spécifiée.
- Indicateur de niveau  Affiche sous forme de bar graphe 3 couleurs, la valeur numérique de l'adresse spécifiée.
- Graphique  Affiche sous forme de graphique un ensemble de point consignés à partir de l'adresse spécifiée.
- Bitmap  Affiche une image format BMP ou JPEG.
- Alarme  Affiche par ordre de priorité les alarmes et événements qui se sont produit.
- Historique d'alarme  Affiche l'historique des alarmes et événements qui se sont produit.
- Date  Affiche/modifie la date.
- Heure  Affiche/modifie l'heure.

- Courbe de tendance  Affiche la courbe de tendance des valeurs d'une ou plusieurs adresses spécifiées.
- Recette  Transfert de recette depuis ou bien vers l'API.
- Affichage Vidéo  Affiche les images vidéo en provenance d'une caméra ou d'un système de vision.
- Boite à onglets  Affiche une boite à onglets multiples.
- Matrice  Crée une matrice d'objet identique (bouton, voyant etc..).

Chaque icône à des paramètres tel que information générale (type d'action (voir figure 3.44), adresse st sa configuration...), la taille et position, l'adresse, couleur et forme, étiquette...etc. Voici un exemple du paramètre d'un bouton ON/OFF.

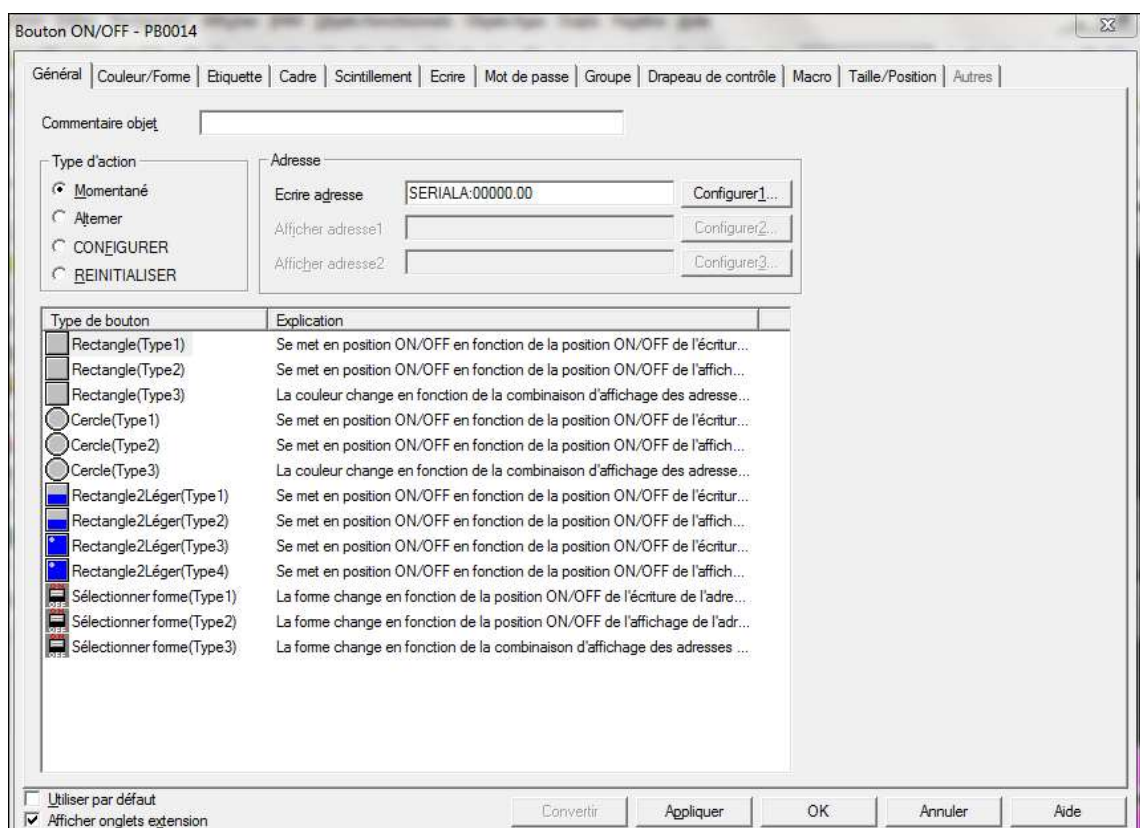


Figure 3.44 : configuration du Bouton ON/OFF.

La figure suivante nous montre la configuration d'adresse on cliquant sur le bouton confuration1 qui apparait dans la figure précédente.

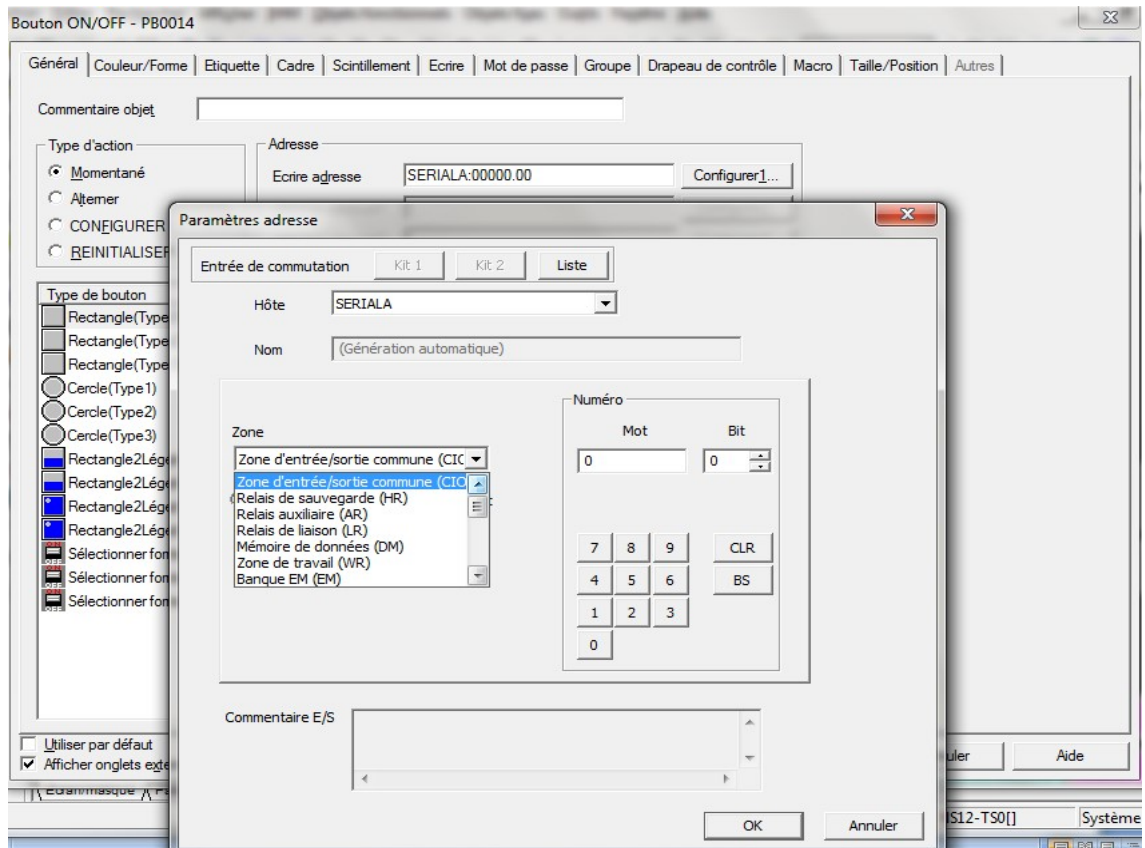


Figure 3.45 : paramétrage d'adresse du bouton ON/OFF créé.

Dans la fenêtre du paramètre adresse on écrit l'adresse et on choisie la zone du placement de l'adresse (zone d'entrée/sortie, relai de sauvegarde, relai auxiliaire, relai de liaison, zone de travail, ...etc.).

Tous ces paramètres doit être bien faite pour le bon fonctionnement, donc il faut éviter les erreurs à ce niveau.

c Transferts du programme dans l'IHM :

Pour pouvoir transférer le projet dans l'IHM NS12, nous connectons le PC avec l'IHM avec le câble RS-232C Ensuite on click sur le bouton IHM dans la barre des taches et on choisie "Transfère vers IHM ". (Voir figure 3.46)

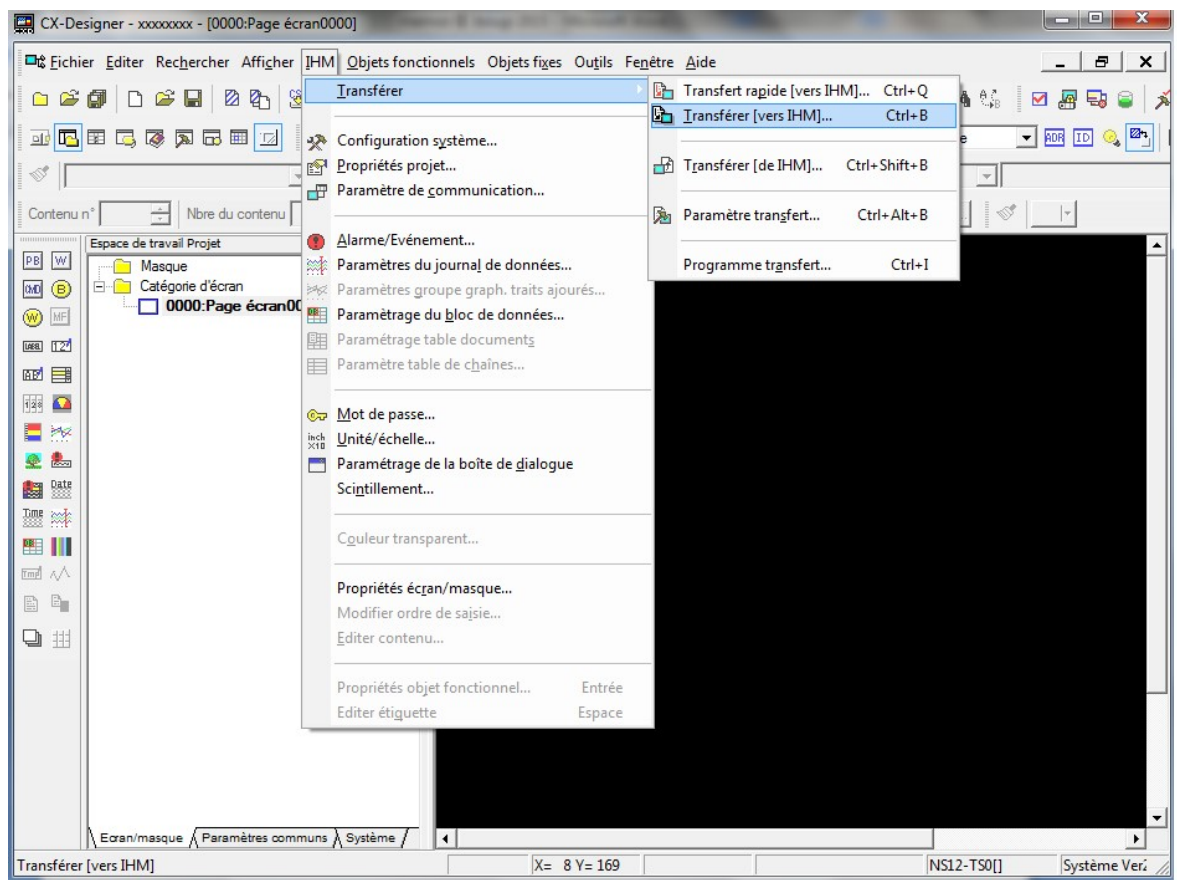


Figure 3.46 : le transfert du programme vers l'IHM.

3.4.3 Configuration du variateur VS mini J7

Le VS mini J7 est le variateur de vitesse qu'on a utilisé pour la réalisation de ce projet, il permet de faire varier la vitesse des moteurs asynchrones en faisant varier la fréquence et la tension d'alimentation du moteur ; il fonctionne sous le principe de la double conversion. Il convertit la tension du réseau en une tension continue puis en une tension alternative d'une amplitude et fréquence programmées pour avoir une vitesse particulière au moteur.

Le schéma de câblage est donné par la figure (3.47) :

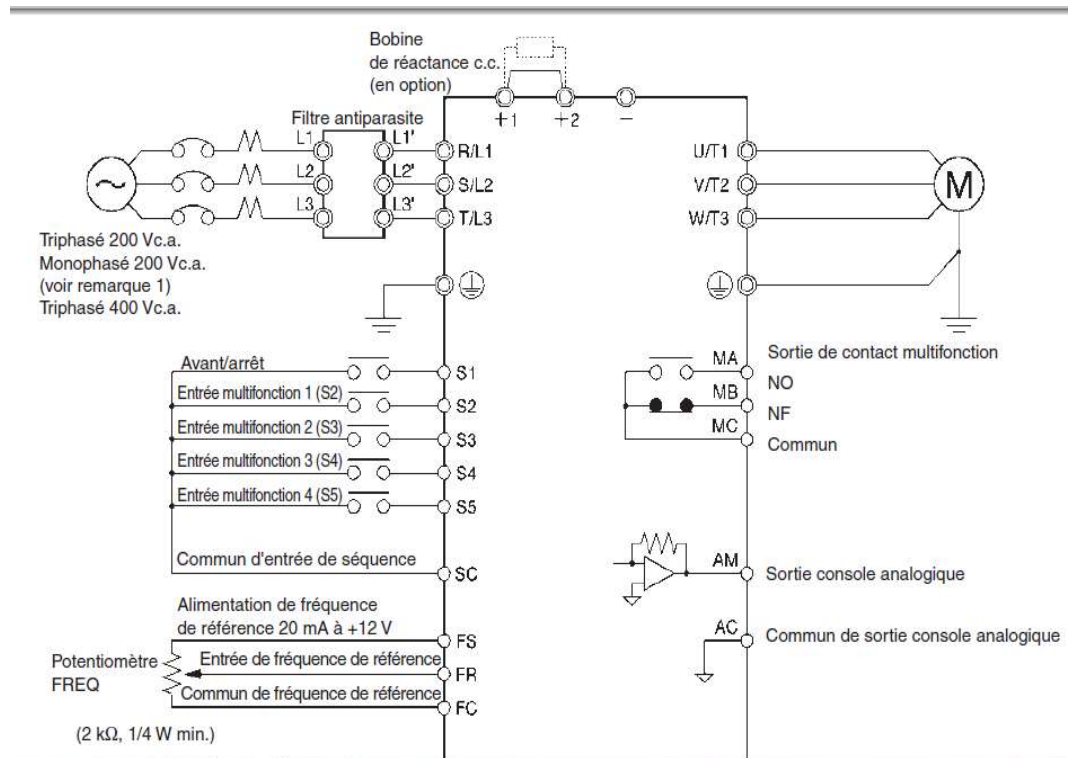


Figure 3.47 : schéma de câblage du VS mini J7.

Après le branchement du variateur au reste des équipements, nous passons au paramétrage du variateur.

Le VS mini J7 possède 79 paramètres configurables par l'utilisateur, soit à travers la console intégré, soit par ordinateur à travers un logiciel dédié tel que le CX-Drive ou CX-Motion du pack CX-ONE. Les paramètres qu'on a adaptés pour notre application sont les suivants :

- Le paramètre (n01) 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. Il peut prendre les valeurs suivantes :

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.

Dans notre cas on a fixé le paramètre à « 1 » pour pouvoir modifier les autres paramètres et après avoir terminé le paramétrage il faut le remettre à « 0 » pour éviter un changement accidentel des paramètres.

- Le paramètre (n02) sert à sélectionner la méthode d'entrée des commandes RUN et STOP en mode distant. Il peut prendre les valeurs suivantes :

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.

Remarque La commande RUN via des séquences d'opérations sur la console numérique n'est acceptée qu'en mode local.

Dans notre cas on a fixé le paramètre à « 1 » pour pouvoir commander le RUN et STOP par l'API.

- Le paramètre (n03) sert à définir la méthode d'entrée de la fréquence de référence en mode distant. Il peut prendre les valeurs suivantes :

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 mA).

4 : Borne du circuit de contrôle de fréquence de référence (0 à 20 mA).

6 : Fréquence de référence via communication RS-422A/485.

Dans notre cas on a fixé le paramètre à « 1 », dans ce il est possible de régler jusqu'à neuf fréquences de référence (1 à 8) et une commande de fréquence pas à pas. ces fréquences sont enregistrés dans les paramètres (n21) jusqu'au (n28).

Les paramètres du (n36) au (n39) sont réservés aux E/S multiples de fonctionnement externes.

La manière dont l'API est connecté au variateur est comme suit :

- Le bornier S1 du variateur est connecté à l'entrée "0.00".
- Le bornier S2 du variateur est connecté à l'entrée "0.03".
- Le bornier S4 du variateur est connecté à l'entrée "0.04".
- Le bornier SC du variateur est connecté au commun.

3.5 Conclusion

Dans ce chapitre nous avons exposé les équipements utilisés pour l'installation qu'on a réalisée, et fourni les moyens de leurs programmations et configurations.

On a choisi le constructeur OMRON non seulement pour la disponibilité de son matériel, mais pour sa robustesse et son faible coût d'installation, citons par exemple les câbles de communications entre PC---API et API---IHM, coutent quatre fois moins chère que ceux de son concurrent SIEMENS.

Chapitre 4 Développement de la réalisation et résultats obtenus

4.1 Introduction

Dans ce qui suit nous allons présenter le prototype qu'on a réalisé pour donner une image rapproché de la machine où le développement du projet est censé être installé. Vue la complexité de la partie mécanique et de son installation et de son coût, il nous est impossible de le reproduire.

Dans ce chapitre nous allons aussi exposer la manière dont l'application a été développé et comment va être utilisé par l'utilisateur final, qu'il soit opérateur de la machine ou personne qualifié pour un éventuel changement de quelques paramètres du procédé de fabrication, l'accès de ce dernier se fait par mot de passe.

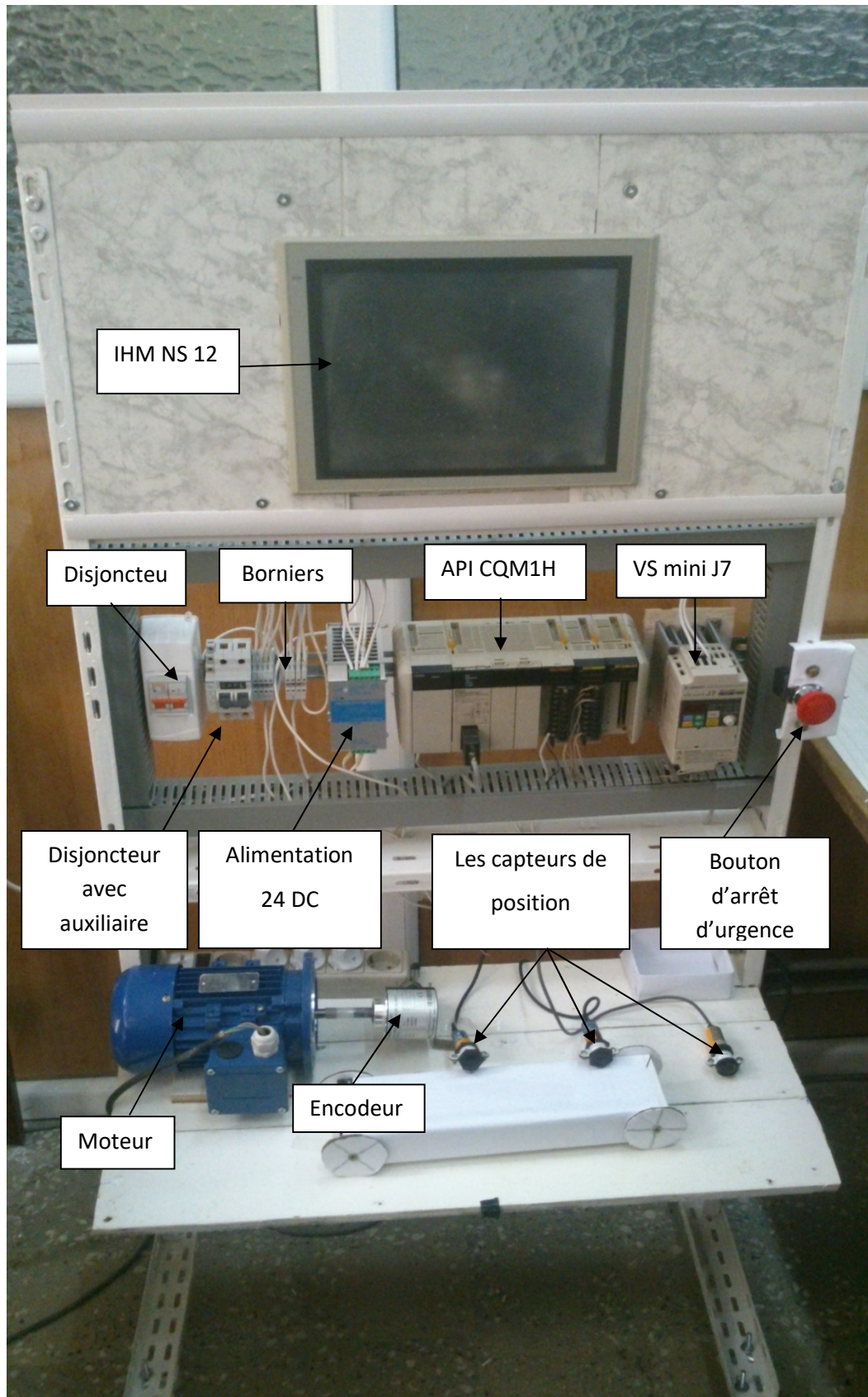


Figure 4.1 : Description des matériels utilisés.

4.2 Présentation des interfaces des programmes (API, IHM, VS mini J7)

4.2.1 Programmation de l'API

Avant de commencer la programmation proprement dite, Il faut d'abord configurer l'API de façon à pouvoir vérifier l'exactitude des adresses utilisées dans le projet. Ensuite on crée les sections et les blocs nécessaires au projet. Enfin on commence par programmer les sections.

Par "configuration", on entend dans ce qui suit la disposition de profilés support ou châssis, de modules, d'appareils de la périphérie décentralisée et de cartouches interface dans une fenêtre de station. Les profilés support ou châssis sont représentés par une table de configuration, dans laquelle l'on peut enficher un nombre défini de modules.

CX-Programmer affecte automatiquement une adresse à chaque module dans la table de configuration. Vous pouvez modifier les adresses des modules d'une station, à condition que la CPU permette l'adressage libre.

Vous pouvez copier la configuration aussi souvent que vous le désirez dans d'autres projets, la modifier si besoin est et la charger dans une ou plusieurs installations existantes.

A la mise en route de l'automate programmable, la CPU compare la configuration prévue créée avec CX-Programmer à la configuration sur site de l'installation. Aussi, les erreurs éventuelles sont-elles immédiatement détectées et signalées.

Une fois la configuration terminée on passe au paramétrage.

Par "paramétrage", on entend dans ce qui suit :

- Le réglage des paramètres des modules paramétrables pour la configuration centralisée et pour un réseau. Exemple : une CPU est un module paramétrable. La surveillance du temps de cycle est un paramètre que vous pouvez définir ;
- La définition des paramètres de bus, des maîtres et d'esclaves pour un réseau maître ou d'autres définitions pour l'échange de données entre des composants.

Ces paramètres sont chargés dans la CPU qui, lors de son démarrage, les transmet aux modules correspondants. Il est très facile de remplacer des modules, car les paramètres définis avec CX-Programmer sont automatiquement chargés dans le nouveau module à la mise en route.

a Mnémonique du programme utilisé :

Les tableaux ci-dessus montrent la mnémonique du programme que nous avons utilisé pour développer notre chaîne de production.

LD	100.00	MOV(21)	#32	AND	P_EQ
OUT	50.00		DM100	ANDNOT	HR0.02
LD	100.00	LD	50.01	LD	TR0
MOV(21)	#6	MOV(21)	230	LD	P_LT
	DM100		DM2	OR	P_GT
LD	100.00	OUT	252.00	ANDLD	
AND	100.03	LD	P_1s	KEEP(11)	HR0.02
MOV(21)	#11	LD	50.01	LD	HR0.02
	DM100	DIFU(13)	HR0.03	OR	HR0.03
LD	100.00	LD	P_On	OR	HR0.05
AND	100.04	MUL(32)	DM2	OR	HR0.06
MOV(21)	#16		#24	OUT	100.04
	DM100		DM6	LD	100.00
LD	100.00	' VITESS DE CONSIGN		OUT	TR0
AND	100.05	LD	100.00	CMP(20)	DM110
MOV(21)	#9	OUT	TR0		#0009
	DM100	CMP(20)	DM110	AND	P_EQ
LD	100.00		#000B	ANDNOT	HR0.00
AND	100.03	AND	P_EQ	LD	TR0
AND	100.04	ANDNOT	HR0.01	LD	P_LT
MOV(21)	#30	LD	TR0	OR	P_GT
	DM100	LD	P_LT	ANDLD	
LD	100.00	LD	P_GT	KEEP(11)	HR0.00
AND	100.03	OR		LD	HR0.00
AND	100.05	ANDLD		OR	HR0.04
MOV(21)	#14	KEEP(11)	HR0.01	OR	HR0.05
	DM100	LD	HR0.01	OR	HR0.06
LD	100.00	OR	HR0.03	OUT	100.05
AND	100.04	OR	HR0.04	LD	100.00
AND	100.05	OR	HR0.06	OUT	TR0
MOV(21)	#19	OUT	100.03	CMP(20)	DM110
	DM100	LD	100.00		#001E
LD	100.00	OUT	TR0	AND	P_EQ
AND	100.03	OUT	DM110	ANDNOT	HR0.03
AND	100.04	CMP(20)	#0010	LD	TR0
AND	100.05			LD	P_LT
				OR	P_GT

(1)

(2)

(3)

Tableau 4.1 : la mnémonique de notre programme.

ANDLD		KEEP(11)	HR0.06
KEEP(11)	HR0.03	LD	100.00
LD	100.00	ANDNOT	HR0.01
OUT	TR0	ANDNOT	HR0.00
CMP(20)	DM110	ANDNOT	HR0.02
	#000E	ANDNOT	HR0.03
AND	P_EQ	OUT	52.00
ANDNOT	HR0.04	LD	100.00
LD	TR0	ANDNOT	HR0.04
LD	P_LT	ANDNOT	HR0.05
OR	P_GT	ANDNOT	HR0.06
ANDLD		OUT	52.01
KEEP(11)	HR0.04	LD	52.00
LD	100.00	AND	52.01
OUT	TR0	OUT	52.02
CMP(20)	DM110		
	#0013		
AND	P_EQ		
ANDNOT	HR0.05		
LD	TR0		
LD	P_LT		
OR	P_GT		
ANDLD			
KEEP(11)	HR0.05		
LD	100.00		
OUT	TR0		
CMP(20)	DM110		
	#0020		
AND	P_EQ		
ANDNOT	0.06		
LD	TR0		
LD	P_LT		
OR	P_GT		
ANDLD			

(4)

(5)

Tableau 4.2 : (suite) la mnémorique de notre programme.

b Aperçu du programme dans le CX Programmer

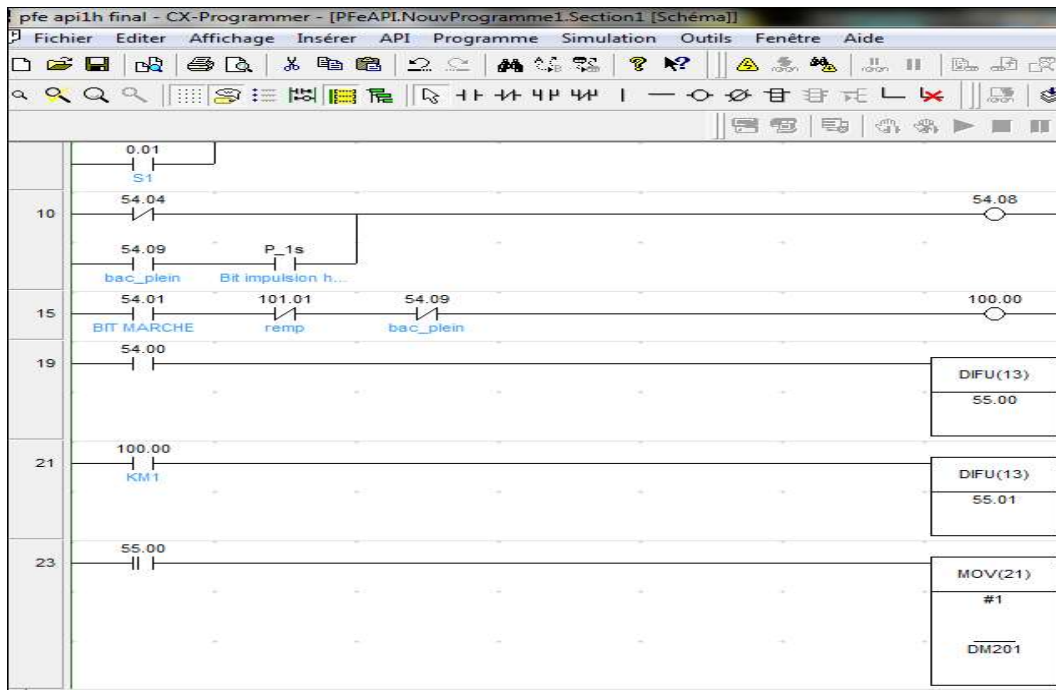


Figure 4.2 : une partie de la programmation de notre API.

4.2.2 La configuration effectuée du variateur VS mini J7

Les huit fréquences de vitesses configurées pour varier la vitesse du notre tapis est comme suit : 6 Hz, 9 Hz, 11 Hz, 14 Hz, 16 Hz, 19 Hz, 30 Hz, 32 Hz. Ces fréquences sont les résultats de combinaison entre 3 vitesses programmées dans le variateur.

4.2.3 Les résultats des fenêtres réalisées pour l'IHM

Les différents fenêtres que nous avons programmé et interprété pour notre projet est comme suit :

Nous présentons tout d'abord la page d'accueil, pour les entreprise ils utilisent souvent leur logo comme accueil de l'IHM.



Figure 4.3 : Ecrans d'accueil de notre projet.

La seconde page est réservé au menu du l'IHM, le se répartie en cinq commande, la configuration de l'IHM (heure/ date, langue), info générale sur l'IHM utilisé et l'API associer ...etc. La commande automatique ou manuel du tapis et l'historique des alarmes.la figure (4.4) montre les commande de menu.

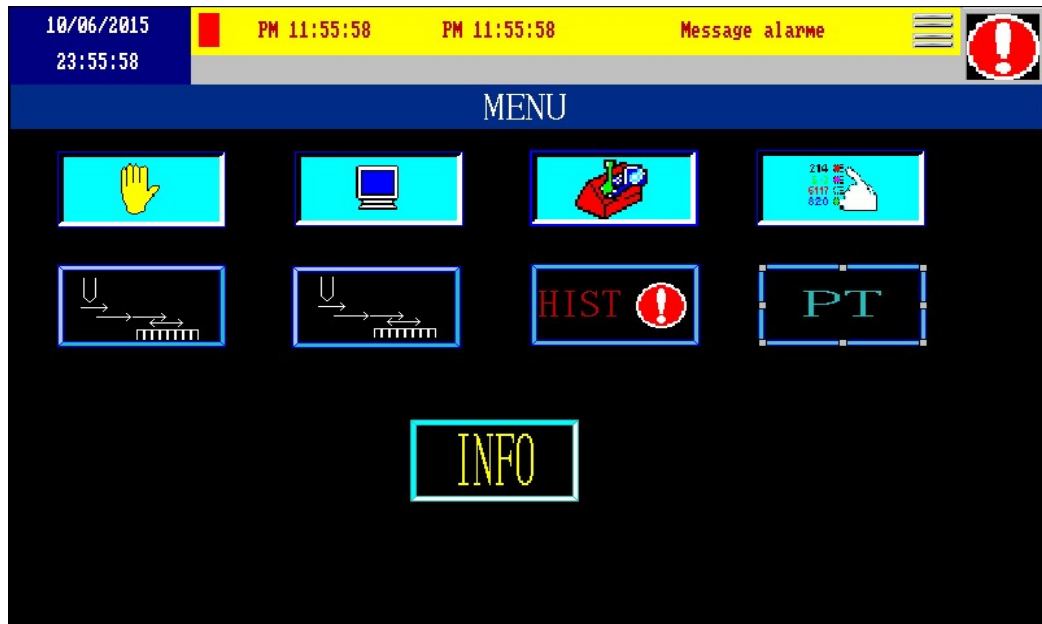


Figure 4.4 : le menu de notre IHM.

1. **INFO** : l'information générale.

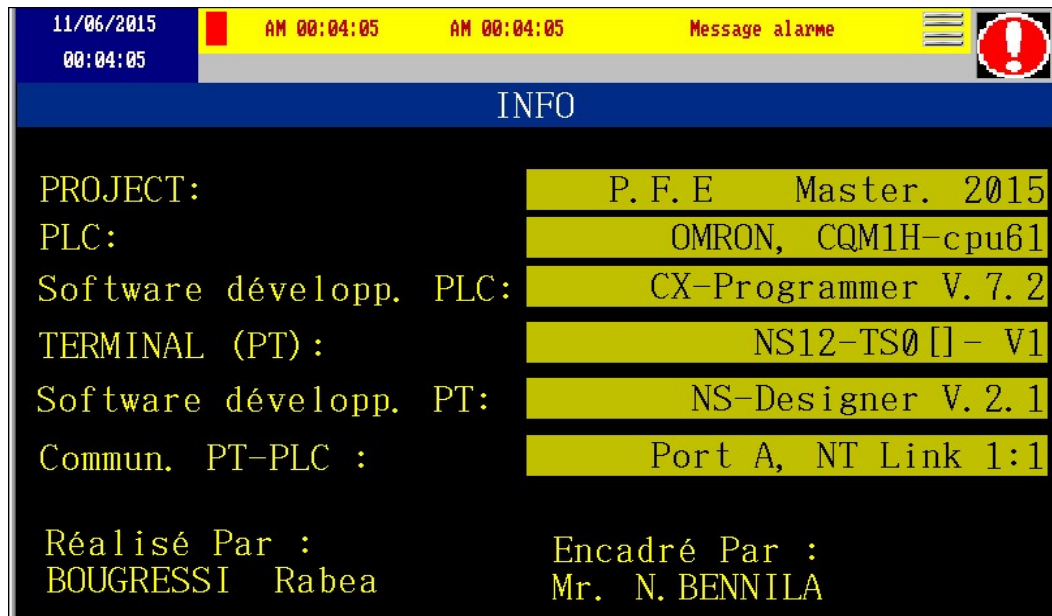


Figure 4.5 : info du notre IHM.

2. **PT** : pour la configuration de l'IHM.

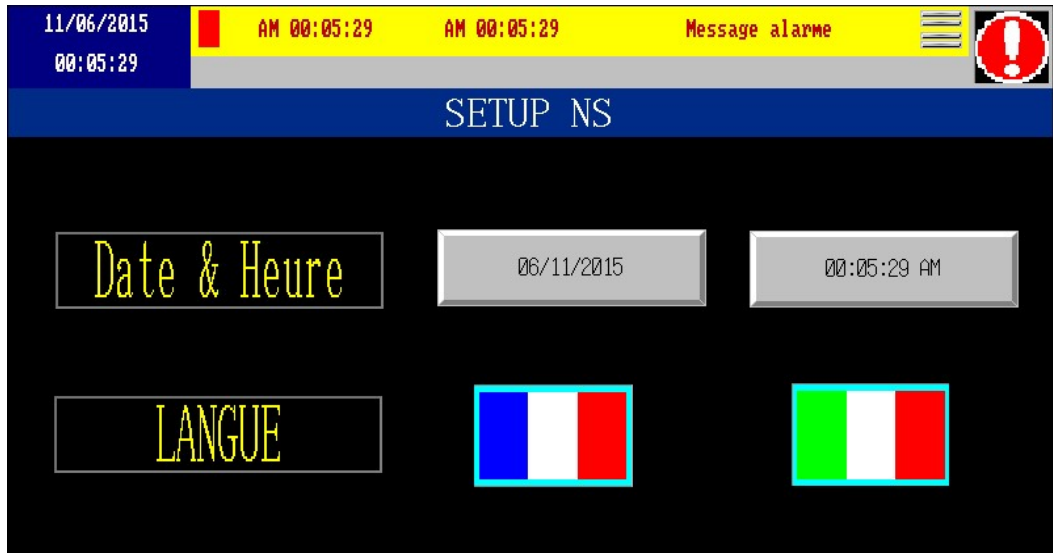


Figure 4.6 : le SETUP de l'IHM.

Pour le changement de la langue ainsi le réglage le l'heurs et date.

3. **HIST** : la fenêtre qui affiche l'historique des alarmes.



Figure 4.7 : Historique des alarmes.

Pour afficher tous les alarmes et la mise à zéro (reset) est codé par un mot de passe pour qu'il soit réinitialisé par des utilisateurs qualifié.

4. La commande manuelle

Le commande manuel permet de varier la vitesse manuellement a l'aide des boutons, ai dans notre cas on a crée des boutons mais virtuellement (dans l'écran).la figure ci-dessus montre la page de manipulation manuelle de notre réalisation.



Figure 4.8 : la commande manuelle de la vitesse.

5. La commande automatique :

La commande automatique du tapis est le but de notre application car elle permet de contrôler et de commander le tapis automatiquement et a partir de l'IHM.La fenêtre suivante présente une vue du la partie automatique par la suite une vue de paramétrage de vitesse.

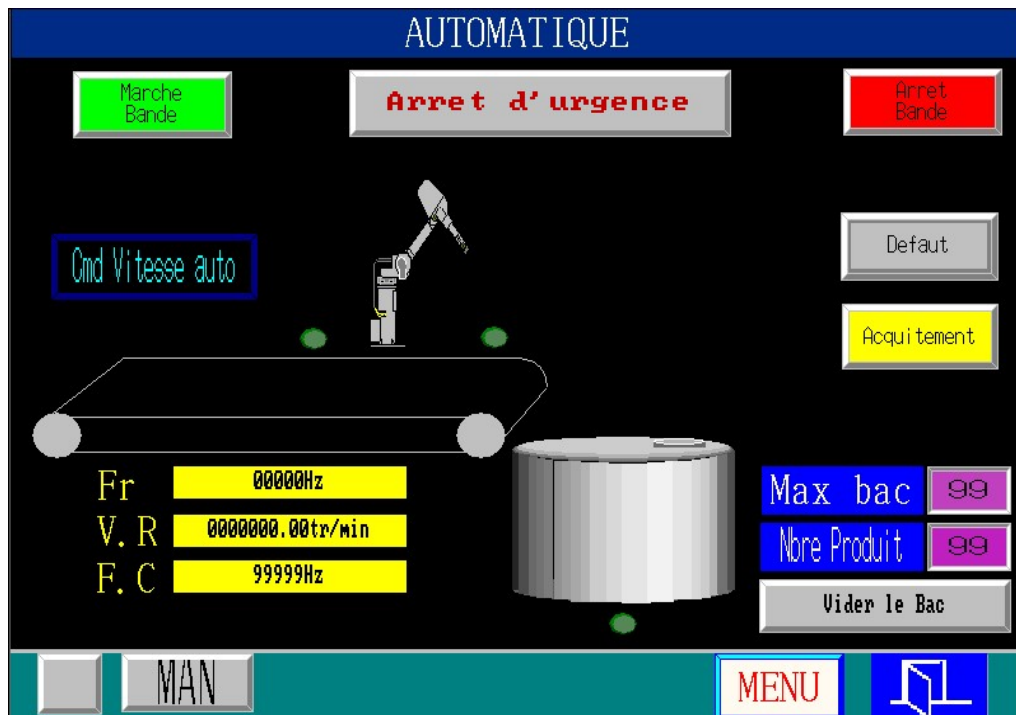


Figure 4.9 : la fenêtre de la commande automatique.

Le passage du commande automatique au manuel est verrouillé en raison de sécurité, on a fait dont un verrouillage, lorsque on choisi le commande automatique et on revient dans le menue pour changé l'état de la commande du automatique en manuel le bouton d'accès au manuel est verrouillé. Le déverrouillage se trouve donc dans la barre des commandes inferieur (commande MAN), lorsqu'on revient dans la page menu le bouton de la commande manuel est maintenant déverrouillé.

Le bouton Cmd Vitesse auto indiqué dans la figure précédente va nous permettre de changer la vitesse, (voir figure 4.10).

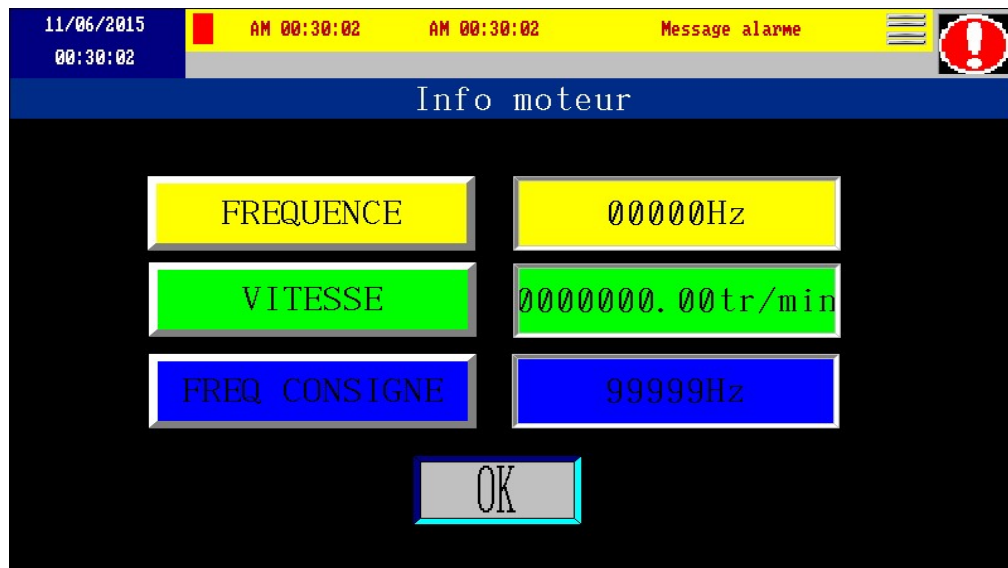


Figure 4.10 : fenêtre de contrôle-commande de fréquence.

4.3 Conclusion

Nous avons présenté dans ce chapitre le prototype qu'on a réalisé, c'est une image très réduite du processus industriel réel, comme il a été susmentionné sa partie mécanique est très complexe à réaliser, mais elle peut être piloté par la partie contrôle et commande qu'on vient de réaliser.

Toutes les mesures de sécurité on été prise pour la protection des biens et du personnelle.

Conclusion générale

Au cours de développement de ce projet, nous avons appris à se familiariser avec les logiciels de programmation des équipements industriels les plus sophistiqués, utilisés pour le contrôle et la supervision des procédés industriels quel que soit leur degré de complexité. Et c'est le but qui a été tracé pour nous, afin de servir le secteur industriel algérien, de le moderniser, et de lui assurer la maintenance, ce qui permet de réduire l'intervention de la main d'œuvre étrangère souvent très coûteuse.

Pour l'application que nous avons développée, très complète au niveau électrique, électronique et informatique industrielle, on aimerait bien qu'elle soit complétée par une bonne partie mécanique et intégrée dans une chaîne de production.

Glossaire

AS-i: Actuator Sensor Interface. Norme de bus de terrain de niveau 0 (ou bus de capteurs/actionneurs).

ASIC: Application Specific Integrated Circuit. Circuit intégré (composant électronique) dédié à une application spécifique.

Il gère, par exemple, la gestion d'un protocole de communication par opposition à un circuit généraliste comme un microprocesseur.

Bus déterministe : Bus temps réel ; les temps de réponse du système sont compatibles avec les appareils commandés.

CIM : Concept élaboré dans les années 80, le CIM (Computer Integrated Manufacturing, traduit en français par Système Intégré de Production, soit SIP) s'est avéré quelque peu "académique" vis-à-vis des réalités industrielles. Néanmoins, il a eu le mérite de bien structurer les différentes fonctions dans l'entreprise. Cette structuration a permis de mieux appréhender les besoins, la nature et l'importance des flux d'informations qui conditionnent le choix des solutions optimales de communication dans les différents niveaux de la pyramide CIM.

Diag Viewer : Fonction de diagnostic disponible pour la plate-forme automate Premium.

Équipement : Désigne tout produit d'automatisme raccordé au bus : automate, variateur de vitesse, distributeur pneumatique, robot, interface homme/machine, etc.

FIP (Factory Instrumentation Protocol) : Bus de terrain couvrant les niveaux 1 et 2.

HTML : Langage de programmation qui permet la définition et la réalisation de documents (textes, images...) conformes aux algorithmes de décodage et d'affichage propres aux logiciels de navigation sur Internet.

Java : Langage informatique orienté objet dédié à internet.

Médium : Support physique de la communication (Paire torsadée, câble coaxial, fibre optique).

Protocole : Désigne les règles de dialogue entre mêmes couches des entités communicantes.

Réseau d'entreprise : Réseau local utilisé dans les applications de bureautique et de gestion.

Réseau local : Réseau limité à une zone ne dépassant pas quelques kilomètres ; en général, il s'agit de réseaux restreints à un bâtiment ou une entreprise, c'est-à-dire restant dans un domaine privé et ne traversant pas le domaine public.

Par opposition, on parle de WAN (Wide Area Network) pour les réseaux comme le Réseau Téléphonique Commuté ou le réseau Internet.

RLI : Abréviation pour Réseau Local Industriel.

Réseau local utilisé dans un environnement industriel (production...). Il permet la communication entre les équipements d'automatisme.

TCP/IP (Transport Control Protocol/Internet Protocol) :

Protocoles de communication utilisés notamment pour la transmission des informations sur le web.

Trame : Suite de bits ou caractères émis de façon ininterrompue par un équipement sur le réseau et dont l'ensemble constitue une information cohérente interprétable par le ou les destinataires (messages, questions ou réponses, diffusion d'une valeur...).

La longueur des trames est toujours limitée.

WAN (Wide Area Network) : Par opposition aux LAN, réseau couvrant une large zone, en général réseau public : Réseau Téléphonique Commuté, Internet...

A noter que depuis quelques années, on parle aussi de MAN (Metropolitan Area Network) ; il s'agit de réseaux rapides couvrant quelques dizaines, voire centaines de kilomètres (des zones métropolitaines).

WWW: World Wide Web ou Web.

Permet un accès à l'échelle planétaire à des services et des documents avec un outil hypertexte, tels que Internet Explorer ou Netscape.

Ces logiciels (ou navigateurs Web) utilisent le protocole http (HyperText Transfer Protocol).

Les documents échangés sont basés sur le langage HTML (HyperText Mark-up Language).

Ladder Diagram :(LD) ou Langage Ladder ou schéma à contacts est un langage graphique très populaire auprès des automaticiens pour programmer les Automates Programmables Industriels. Il ressemble un peu aux schémas électriques, et est facilement compréhensible.

Listes des acronymes et abréviations

API : automate programmable industriel.

PLC : Programmable Logic Controller.

IHM : interface homme-machine, en anglais (HMI).

CAO : conception assistée par ordinateur.

FAO : fabrication assistée par ordinateur.

GPAO : gestion de production assistée par ordinateur.

NTIC : nouvelles technologies de l'information et de la communication.

RLI : réseaux locaux industriels.

AS-I : Actuator Sensor interface.

Fipio : passerelle d'entrée /sortie.

CAN : Controller Area Network.

OSI: Open System Interconnections.

IEEE: institute of electrical and electronics engineers.

TCP: Transmission Control Protocol.

IP: Internet Protocol.

CSMA/CD: Carrier Sense Multiple Access with Collision Detection.

UDP: user datagram protocol.

ICMP: internet control message protocol.

IGMP: internet group management protocol.

OPC: Open Platform Communications.

UNI-TE: unified terminal equipment.

HTML: hyper text markup language.

MES: Manufacturing Execution System.

ERP:Enterprise Ressource Planning.

LAN: Local Area Network ou réseau local.

MAN: Metropolitan Area Network.

WAN: Wide Area Network.

SCADA: Supervisory Control and Data Acquisition.

DCS : Systèmes de Commande Distribuée.

CPU: central processing unit.

ROM: Read Only Memory.

EPROM: erasable programmable read only memory.

RAM : Random Access Memory.

GRAFCET : graphe du groupe AFCET (Association française pour la cybernétique économique et technique).

RTU: remote terminal unit.

Mbps: megabits per second.

C.C: courant continue.

C.A: Courant alternatif.

Liste des paramètres du variateur de vitesse VS mini J7

N° paramètre (n° registre (Hex))	Nom	Description	Plage de réglage	Unité de réglage	Réglage par défaut	Modifications pendant fonctionnement	Page de référence
n01 (0101)	Sélection d'interdiction d'écriture de paramètres / initialisation de paramètres	<p>Sert à interdire l'écriture de paramètres, régler des paramètres ou modifier la plage de surveillance de paramètres.</p> <p>Sert à rétablir la valeur par défaut des paramètres.</p> <p>0 : Règle ou surveille le paramètre n01. Les paramètres n02 à n79 peuvent uniquement être surveillés.</p> <p>1 : Règle ou surveille les paramètres n01 à n79.</p> <p>6 : Efface le journal d'erreurs.</p> <p>8 : Rétablit la valeur par défaut des paramètres en séquence à 2 fils.</p> <p>9 : Rétablit la valeur par défaut des paramètres en séquence à 3 fils.</p>	0, 1, 6, 8, 9	1	1	Non	5-2
n02 (0102)	Sélection de la commande de fonctionnement	<p>Sert à sélectionner la méthode d'entrée des commandes RUN et STOP en mode distant.</p> <p>0 : Les touches RUN et STOP/RESET de la console numérique sont activées.</p> <p>1 : Entrée multifonction via les bornes du circuit de contrôle en séquence à 2 ou 3 fils.</p> <p>2 : Les commandes de fonctionnement via une communication RS-422A/485 sont activées.</p> <p>Remarque La commande RUN via des séquences d'opérations sur la console numérique n'est acceptée qu'en mode local.</p>	0 à 2	1	0	Non	5-7

n03 (0103)	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 de fréquence de référence (0 à 20 V) 6 : Fréquence de référence via communication RS-422A/485	0 à 4, 6	1	0	Non	5-8
n04 (0104)	Sélection du mode d'interruption	Sert à régler la méthode d'arrêt à appliquer lors de l'entrée de la commande STOP. 0 : Arrêt par décélération dans le temps pré-réglé. 1 : Arrêt sur inertie (sortie désactivée par la commande STOP)	0, 1	1	0	Non	5-16
n05 (0105)	Sélection de la rotation arrière interdite	Sert à sélectionner le fonctionnement avec l'entrée de commande arrière. 0 : Fonctionnement arrière activé 1 : Fonctionnement arrière désactivé	0, 1	1	0	Non	5-15
n06 (0106)	Sélection de la fonction de la touche STOP/RESET	Sert à sélectionner la méthode d'arrêt en mode distant avec n02 (sélection du mode de fonctionnement) réglé sur 1. 0: Touche STOP/RESET de la console numérique activée. 1 : Touche STOP/RESET de la console numérique désactivée.	0, 1	1	0	Non	5-7
N° paramètre (n° registre (Hex))	Nom	Description	Plage de réglage	Unité de réglage	Réglage par défaut	Modifications pendant fonctionnement	Page de référence
n07 (0107)	Sélection de la fréquence en mode local	Sert à définir la méthode d'entrée de la fréquence de référence en mode local. 0 : Potentiomètre FREQ de la console numérique activé 1 : Séquences d'opérations de la console numérique activées	0, 1	1	0	Non	5-8
n08 (0108)	Réglage de la fréquence séquentielle de touches	Sert à activer la touche Entrée pour régler la fréquence de référence à l'aide des touches Augmenter et Diminuer. 0 : Valeur entrée en appuyant sur la touche Entrée 1 : Valeur activée lors de l'entrée de la valeur	0, 1	1	0	Non	5-12

n09 (0109)	Fréquence maximale (FMAX)	Sert à régler le schéma V/f comme caractéristique de base du variateur avec tension de sortie par fréquence.	50,0 à 400	0,1 Hz (voir remarque 1)	60,0	Non	5-4
n10 (010A)	Tension maximale (VMAX)	<p>Tension de sortie V(VMAX)</p> <p>n14 (FMIN) n12 (FB) n11 (FMAX) n09 (FMAX)</p> <p>Fréquence (HZ)</p>	1 à 255 (voir remarque 2)	1 V	200 (voir remarque 2)	Non	5-4
n11 (010B)	Fréquence de tension maximale (FA)		0,2 à 400	0,1 Hz (voir remarque 1)	60,0	Non	5-4
n12 (010C)	Fréquence de sortie moyenne (FB)	<p>Remarque Réglez les paramètres afin de respecter la condition suivante : $n14 \leq 012 < n11 \leq n09$</p> <p>Remarque La valeur réglée pour n13 est ignorée si les paramètres n14 et n12 ont la même valeur.</p>	0,1 à 399	0,1 Hz (voir remarque 1)	1,5	Non	5-4
n13 (010D)	Tension de fréquence de sortie moyenne (VC)		1 à 255 (voir remarque 2)	1 V	12 (voir remarque 2)	Non	5-4
n14 (010E)	Fréquence de sortie minimale (FMIN)		0,1 à 10,0	0,1 Hz	1,5	Non	5-4
n15 (010F)	Tension de fréquence de sortie minimale (VMIN)		1 à 50 (voir remarque 2)	1 V	12,0 (voir remarque 2)	Non	5-4
n16 (0110)	Temps d'accélération 1	Temps d'accélération : temps nécessaire pour passer de 0 à 100 % de la fréquence maximale.	0,0 à 999	0,1 s	10,0	Oui	5-13
n17 (0111)	Temps de décélération 1	Temps de décélération : temps nécessaire pour passer de 100 à 0 % de la fréquence maximale.			10,0	Oui	5-13
n18 (0112)	Temps d'accélération 2	<p>Remarque La formule ci-dessous permet de calculer le temps d'accélération ou de décélération réel :</p> <p>Temps d'accélération / décélération = (Point de consigne du temps d'accélération / décélération) × (Valeur de fréquence de référence) ÷ (Fréquence max.)</p>			10,0	Oui	5-13
n19 (0113)	Temps de décélération 2				10,0	Oui	5-13

N° paramètre (n° registre (Hex))	Nom	Description	Plage de réglage	Unité de réglage	Réglage par défaut	Modifications pendant fonctionnement	Page de référence
n20 (0114)	Caractéristique d'accélération / décélération en S	Sert à régler les caractéristiques d'accélération / décélération en S. 0 : Pas d'accélération / décélération en S (accélération / décélération trapézoïdales) 1 : Temps de caractéristique d'accélération / décélération en S 0,2 s 2 : Temps de caractéristique d'accélération / décélération en S 0,5 s 3 : Temps de caractéristique d'accélération / décélération en S 1,0 s Remarque Une fois le temps caractéristique d'accélération / décélération en S réglé, les temps d'accélération et de décélération sont allongés en fonction de la forme en S au début et à la fin de l'accélération / décélération.	0 à 3	1	0	Non	5-14
n21 (0115)	Fréquence de référence 1	Sert à régler les fréquences de référence internes.	0,0 à la fréquence max.	0,1 Hz (voir remarque 1)	6,0	Oui	5-10
n22 (0116)	Fréquence de référence 2	Remarque La fréquence de référence 1 est activée en mode distant en réglant le paramètre n03 (sélection de la fréquence de référence) sur 1. Remarque Ces fréquences de référence sont sélectionnées à l'aide de références de vitesse à pas multiples (entrée multifonction). Reportez-vous aux pages de référence pour connaître le rapport entre les références de vitesse à pas multiples et les fréquences de référence.			0,0	Oui	5-10
n23 (0117)	Fréquence de référence 3				0,0	Oui	5-10
n24 (0118)	Fréquence de référence 4				0,0	Oui	5-10
n25 (0119)	Fréquence de référence 5				0,0	Oui	5-10
n26 (011A)	Fréquence de référence 6				0,0	Oui	5-10
n27 (011B)	Fréquence de référence 7				0,0	Oui	5-10
n28 (011C)	Fréquence de référence 8				0,0	Oui	5-11
n29 (011D)	Commande de fréquence pas à pas				Sert à régler la commande de fréquence pas à pas. Remarque La commande de fréquence pas à pas est sélectionnée avec la commande pas à pas (entrée multifonction). La commande de fréquence pas à pas est prioritaire sur la référence de vitesse à pas multiples.	6,0	Oui

n30 (011E)	Limite supérieure de la fréquence de référence	Sert à régler les limites supérieure et inférieure de la fréquence de référence sous la forme d'un pourcentage, la fréquence maximale étant égale à 100 %. Remarque Si le paramètre n31 est réglé sur une valeur inférieure à la fréquence de sortie minimale (n14), le variateur n'a aucune sortie lorsqu'une fréquence de référence inférieure à l'entrée de fréquence de sortie minimale est entrée.	0 à 110	1 %	100	Non	5-9
n31 (011F)	Limite inférieure de la fréquence de référence		0 à 110	1 %	0	Non	5-9
n32 (0120)	Courant nominal du moteur	Sert à régler le courant nominal du moteur pour la détection de surcharge du moteur (OL1) en fonction du courant nominal du moteur. Remarque La fonction de détection de surcharge du moteur (OL1) est désactivée en réglant le paramètre sur 0,0. Remarque Le courant nominal du moteur correspond par défaut au courant nominal standard du moteur applicable maximal.	0,0 à 120 % du courant de sortie nominal du variateur	0,1 A	Varie selon la capacité	Non	5-2
N° paramètre (n° registre (Hex))	Nom	Description	Plage de réglage	Unité de réglage	Réglage par défaut	Modifications pendant fonctionnement	Page de référence
n33 (0121)	Caractéristiques de protection du moteur	Sert à régler la détection de surcharge du moteur (OL1) pour les caractéristiques thermoélectroniques du moteur. 0 : Caractéristiques de protection pour les moteurs à induction universels 1 : Caractéristiques de protection pour les moteurs dédiés au variateur 2 : Aucune protection Remarque En cas de raccordement d'un seul variateur à plusieurs moteurs, réglez ce paramètre sur 2 (aucune protection). Le paramètre est également désactivé en réglant n32 (moteur nominal) sur 0,0.	0 à 2	1	0	Non	6-14
n34 (0122)	Réglage du temps de protection du moteur	Sert à régler les caractéristiques thermoélectriques du moteur à connecter par incréments d'1 minute. Remarque Le réglage par défaut ne nécessite aucune modification du fonctionnement normal. Remarque Pour régler le paramètre en fonction des caractéristiques du moteur, demandez au fabricant du moteur de fournir la constante de temps thermique et prévoyez une marge. En d'autres termes, réglez ce paramètre sur une valeur légèrement inférieure à la constante de temps thermique. Remarque Pour détecter la surcharge du moteur de manière plus rapide, diminuez le point de consigne, à condition que cela n'entraîne aucun problème d'application.	1 à 60	1 mn	8	Non	6-14

n35 (0123)	Fonction d'activation du ventilateur	Sert à activer le ventilateur du variateur lorsque celui-ci est sous tension ou en cours de fonctionnement. 0 : Tourne uniquement lorsque la commande RUN est entrée, et ce pendant 1 minute après l'arrêt de fonctionnement du variateur. 1 : Tourne lorsque le variateur est sous tension. Remarque Ce paramètre n'est disponible que si le variateur intègre un ventilateur. Remarque Si la fréquence de fonctionnement du variateur est faible, il est possible de prolonger la durée de vie du ventilateur en réglant le paramètre sur 0.			0, 1	1	0	Non	6-14
n36 (0124)	Entrée multifonction 1 (borne d'entrée S2)	Sert à sélectionner la fonction des bornes d'entrée multifonction S2 à S5.			2 à 8, 10 à 22	1	2	Non	5-17
		Point de consigne	Fonction	Description					
n37 (0125)	Entrée multifonction 2 (borne d'entrée S3)	0	Commande de rotation avant / arrière	Séquence à 3 fils (à régler sur n37 uniquement) En réglant le paramètre n37 sur 0, le point de consigne de n36 est ignoré et les réglages suivants sont forcés : S1 : entrée RUN (RUN quand ON) S2 : entrée STOP (STOP quand OFF) S3 : commande de rotation avant / arrière (OFF : avant ; ON : arrière)	0, 2 à 8, 10 à 22	1	5	Non	5-17
n38 (0126)	Entrée multifonction 3 (borne d'entrée S4)				2 à 8, 10 à 22	1	3	Non	5-17
N° paramètre (n° registre (Hex))	Nom	Description			Plage de réglage	Unité de réglage	Réglage par défaut	Modifications pendant fonctionnement	Page de référence
n39 (0127)	Entrée multifonction 4 (borne d'entrée S5)	2	Arrière / Arrêt	Commande de rotation arrière en séquence à 2 fils (Inversée avec borne activée)	2 à 8, 10 à 22, 34, 35	1	6	Non	5-17
		3	Erreur externe (NO)	ON : erreur externe (détection FP□ : □ correspond à un numéro de borne)					
		4	Erreur externe (NF)	OFF : erreur externe (détection EF□ : □ correspond à un numéro de borne)					
		5	Réinitialisation erreur	ON : réinitialisation erreur (désactivée lors de l'entrée de la commande RUN)					
		6	Référence de vitesse à pas multiples 1	Signaux permettant de sélectionner les fréquences de référence 1 à 8. Voir la section 5-5-4 Réglage des fréquences de référence via les séquences d'opérations pour connaître le rapport entre les références de vitesse à pas multiples et les fréquences de référence.					
		7	Référence de vitesse à pas multiples 2						
		8	Référence de vitesse à pas multiples 3						

		10	Commande de fréquence pas à pas	ON : commande de fréquence pas à pas (prioritaire sur la référence de vitesse à pas multiples)					
		11	Basculement du temps d'accélération / décélération	ON : le temps d'accélération 2 et le temps de décélération 2 sont sélectionnés					
		12	Commande de circuit externe (NO)	ON : sortie désactivée (pendant l'arrêt du moteur et clignotement de « bb »)					
		13	Commande de circuit externe (NF)	OFF : sortie désactivée (moteur en rotation libre et clignotement de « bb »)					
		14	Commande de recherche (début de la recherche à partir de la fréquence maximale)	ON : recherche de vitesse (début de la recherche à partir de n09)					
		15	Commande de recherche (début de la recherche à partir de la fréquence pré-réglée)	ON : recherche de vitesse					
		16	Commande d'interdiction d'accélération / décélération	ON : accélération / décélération maintenues (exécution à la fréquence paramétrée)					
		17	Sélection du mode local / distant	ON : mode local (utilisation de la console numérique)					

N° paramètre (n° registre (Hex))	Nom	Description		Plage de réglage	Unité de réglage	Réglage par défaut	Modifications pendant fonctionnement	Page de référence	
n39 (0127)	Entrée multifonction 4 (borne d'entrée S5)	18	Sélection de l'entrée de communication ou distante	ON : activation de l'entrée de communication RS-422A/485 OFF : les réglages de n02 et n03 sont activés	2 à 8, 10 à 22, 34, 35	1	6	Non	5-17
		19	Erreur d'arrêt d'urgence (NO)	Le variateur s'arrête en fonction du réglage du paramètre n04 (sélection du mode d'interruption) lorsque l'entrée d'arrêt d'urgence est activée.					
		20	Alarme d'arrêt d'urgence (NO)	NO : arrêt d'urgence avec contact fermé. NF : arrêt d'urgence avec contact ouvert.					
		21	Erreur d'arrêt d'urgence (NF)	Erreur : sortie d'erreur activée et réinitialisée à l'aide de l'entrée RESET. Sortie alarme activée (pas de réinitialisation requise).					
		22	Alarme d'arrêt d'urgence (NF)	STP s'affiche (allumé avec entrée d'erreur activée et clignotant avec entrée alarme					
		34	Commande haut ou bas	Commande haut ou bas (réglée sur n39 uniquement) En réglant le paramètre n39 sur 34, le point de consigne pour n38 est ignoré et les réglages suivants sont forcés : S4 : commande haut S5 : commande bas					
		35	Test d'auto-diagnostic	ON : test d'auto-diagnostic de communication RS-422A/485 (réglé sur n39 uniquement)					

Références bibliographique

[1] Wikipedia.

[2] C. JOSSIN. *'Buts_de_l'automatisme'.DOC. AUTOM.*

[3] Alain GONZAGA. LES AUTOMATES PROGRAMMABLES INDUSTRIELS .GEEA.2012

Site :[http://www.geea.org/IMG/pdf/LES AUTOMATES PROGRAMMABLES INDUSTRIELS pour GEEA.](http://www.geea.org/IMG/pdf/LES_AUTOMATES_PROGRAMMABLES_INDUSTRIELS_pour_GEEA.pdf)

[4] site : <http://www.thomasta.com/>.

[5]Le magazine Schneider Electrique de L'enseignement technologique et professionnel.' Les bus et les réseaux de terrain en automatisme industriel'. Inter-section. novembre 2002.

[6] *Cours Réseaux de terrain support cours*

[7] support cours Réseaux « *Les Réseaux Locaux Industriels* ».

[8] Chaker ABDELJAOUED, 'Elaboration d'un système d'automatisme et de régulation d'une unité d'aérocondenseur de vapeur d'eau en remplacement d'un condenseur de vapeur à eau de mer'. Université du 7 novembre à Carthage .2010.

[9] François Tremblay, ' L'UTILISATION DES TABLETTES ÉLECTRONIQUES EN AUTOMATISATION INDUSTRIELLE'. PUBLICATION CPA-2011-100-R1 - Avril 2011.

[10] *Belkacem OULD BOUAMAMA : ' Supervision des Systèmes Industriels', Laboratoire d'Automatique, Génie Informatique et Signal de Lille (LAGIS -UMR CNRS 8219),2014.*

[11] David Bailey BEng, Bailey and Associates, Perth,' Practical SCADA for Industry', IDC Technologies, Australia, 2003, ISBN 07506 58053

[12] Site web:

<http://www.univ-reims.fr/site/laboratoires/meserp/descriptif-deslogiciels,948627015.html>

[13] Site web: <http://www.feinc.com/view.php?id=69>

[14]

https://www.swe.siemens.com/belux/portal/fr/offre/industrie/automatisation/hmi/Pages/scada_wincc.aspx

[15]

http://france.fileschneider.com/visualisation_fiche.php?fic_id=34&fon_id=18&fam_id=84

[16]

<http://www.ip-systemes.fr/Produits/SupervisionScadaMes/CitectSCADA.aspx>

[17]

http://industrial.omron.fr/fr/products/catalogue/automation_systems/software/configuration/cx-one/cx-programmer.html

[18]

http://industrial.omron.fr/fr/products/catalogue/automation_systems/software/configuration/cx-one/cx-designer.html

[19]

http://industrial.omron.fr/fr/products/catalogue/automation_systems/software/runtime/cx-supervisor/default.html

[20] OMRON ELECTRONICS LLC, 'Datasheet OMRON CQM1H', Cat. No. GC-MSPLC1.2003.

[21] http://www.contrinex.com/fr/business_unit/photoelectric-sensors-2/.

[22] VS mini J7, 'Variateur industriel compact'. MANUEL D'UTILISATION. Cat. No. I63E-FR