

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement Supérieur et de la Recherche scientifique



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Automatique
Option : Automatique et informatique industriel

Réf :

Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme :

MASTER

Thème

*Commande d'un ascenseur didactique Par un API Schneider
M340*

Présenté par :
GHALEM Ala Eddine
Soutenu le : 24 Juin 2018

Devant le jury compose de :

M ^r Kraa Okba	MCB	Président
M ^r Achour SAADOUNE	MCA	Encadreur
M ^r Khaled ABADA	MAA	Examineur

Année universitaire: 2017/2018

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement Supérieur et de la recherche scientifique



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Automatique
Option : Automatique et Informatique Industriel

Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme :

MASTER

Thème

*Commande d'un ascenseur didactique Par un API
Schneider M340*

Présenté par :

- *Ghalem Ala Eddine*

Avis favorable de l'encadreur :

M^r. Achour SAADOUNE

Signature

Avis favorable du Président du Jury

M^r Kraa Okba

Signature

Cachet et signature

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement Supérieur et de la recherche scientifique



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Automatique
Option : Automatique et Informatique Industriel

Thème

Commande d'un ascenseur didactique Par un API Schneider M340

Préparé par : Ghalem Ala Eddine

Dirigé par : Dr. SAADOUNE Achour

RESUME

Dans ce projet on a fait la commande et la programmation d'un ascenseur didactique 4 étages par L'automate Programmable Industriel Schneider M340 disponible au niveau de notre laboratoire d'automatique avec le logiciel de programmation UNITY PRO.

Ce travail nous a permis de comprendre les différentes étapes suivies pour l'automatisation des systèmes, maîtriser le logiciel de programmation UNITY PRO des automates de la gamme M340 et par ailleurs assure une bonne formation pour les étudiants de la filière automatique.

Mots-clés : Automatisation, ascenseur 4 étages, Le logiciel UNITY PRO, l'automate programmable industriel M340.

ملخص

في هذا المشروع قمنا بالتحكم وبرمجة نظام مصعد تعليمي بواسطة جهاز التحكم المنطقي القابل للبرمجة شنايدر (M340) المتوفر على مستوى مخبر الالية مع نظام البرمجة UNITY PRO.

هذا المشروع سمح لنا بمعرفة مختلف المراحل المتبعة من أجل جعل الأنظمة الية، التمكن من الاستخدام الجيد لبرنامج UNITY PRO الخاص بأجهزة التحكم المنطقي شنايدر M340 ومن جهة أخرى ضمان تكوين جيد لطلبة شعبة الالية.

الكلمات المفتاحية : التشغيل الآلي، مصعد 4 طوابق، برنامج UNITY PRO، وحدة التحكم الصناعي القابل للبرمجة M340.

Dédicace

Je dédie ce modeste travail

A celle qui a attendu avec impatience les fruits de

Sa bonne éducation ... ma chère mère.

A celui m'a indiqué la bonne voie en me rappelant

Que la volonté fait toujours les grandes personnes

Mon cher père.

A mes sœurs

A mon frère Saad Ayoub.

A Ma fiancé fifi

A mes amies

Spécial dédicace a mon ami Elias "babay"

A membres de SB9_Gang

A mes amis(es) et camarades du primaire jusqu'à l'université

A tous ceux qui m'aiment et que j'aime

REMERCIEMENTS

Je tiens à dire que j'ai réussi à réaliser ce modeste travail grâce à « Dieu » qui m'a donné le pouvoir, la volonté et le courage d'être arrivé jusqu'au là.

Je remercie vivement mon encadreur « Dr. Achour Saadoune » qui m'a honoré par son encadrement, sa générosité en information, sa direction, sa modestie, sa patience, ses conseils et toutes ces remarques constructives pour le bon déroulement de mon travail. Mes remerciements vont également aux membres de jury : M^{elle}.

Kraa Okbaet Mr. Abada Khaled d'avoir accepté de juger mon travail

J'adresse aussi un grand merci à tous les enseignants de Département de Génie électrique, À toute personne qui a posé son empreinte de loin ou de près pour faire aboutir ce modeste travail

Merci du fond du cœur

Résumé

Dans ce projet on a fait la commande et la programmation d'un ascenseur didactique 4 étages par L'automate Programmable Industriel Schneider M340 disponible au niveau de notre laboratoire d'automatique avec le logiciel de programmation UNITY PRO.

Ce travail nous a permis de comprendre les différentes étapes suivies pour l'automatisation des systèmes, maîtriser le logiciel de programmation UNITY PRO des automates de la gamme M340 et par ailleurs assure une bonne formation pour les étudiants de la filière automatique.

Mots-clés : Automatisation, ascenseur 4 étages, Le logiciel UNITY PRO, l'automate programmable industriel M340.

ملخص

في هذا المشروع قمنا بالتحكم وبرمجة نظام مصعد تعليمي بواسطة جهاز التحكم المنطقي القابل للبرمجة شنايدر (M340) المتوفر على مستوى مخبر الآلية مع نظام البرمجة UNITY PRO.

هذا المشروع سمح لنا بمعرفة مختلف المراحل المتبعة من أجل جعل الأنظمة الآلية، التمكن من الاستخدام الجيد لبرنامج UNITY PRO الخاص بأجهزة التحكم المنطقي شنايدر M340 ومن جهة أخرى ضمان تكوين جيد لطلبة شعبة الآلية.

الكلمات المفتاحية: التشغيل الآلي، مصعد 4 طوابق، برنامج UNITY PRO، وحدة التحكم الصناعي القابل للبرمجة M340.

LISTE DES FIGURES ET TABLEAUX

CHAPITRE I :

Figure I.1 : structur type d'un automate en boucle fermée.....	4
Figure I.2 : structure générale d'un système automatisé de production.....	5
Figure I.3 : Structure des automates.....	8
Figure I.4: Architecture interne d'un automate programmable.....	10

CHAPITRE II :

Figure II.5 :Diagnostics intégrés aux plates-formes d'automatisme Modicon M340, Modicon M580, Premium et Quantum.....	26
Figure II.2 : régulateur PID avec commande manuelle M.....	31

CHAPITRE III :

Figure III-6: exemple de configuration de la station automate Modicon M340 avec un rack..	37
Figure III-2: rack BMX XPB 0400.....	38
Figure III-7: module d'extension de rack BMX XBE 1000.....	39
Figure III-4:un module d'alimentation BMX CPS •••.....	39
Figure III-8: un module d'entrées/sorties TOR avec connecteurs 40 points.....	40
Figure III-6: un module d'entrées/sorties TOR avec bornier 20 points.....	40
Figure III-7: un module d'entrées/sorties analogique avec un connecteur 40 points.....	41
Figure III-8 : un module d'entrées/sorties analogique avec bornier 20 points.....	41
Figure III-9 :le module de comptage BMX EHC 0200.....	42
Figure III-10 : le module de comptage BMX EHC 0800.....	42

Figure III-91: réseau Modbus.....	43
Figure III-110: un réseau Ethernet.....	44
FigureI-111: une architecture de bus de terrain CANopen.....	44

CHAPITRE IV :

Figure IV.1 : Socle support de l'ascenseur.....	49
Figure IV.2 : automate Modicon M340.....	50
Figure IV.3:Commutateur "Stop/Run".....	51
Figure IV.4 : Boutons de demande extérieure.....	52
Figure IV.5 : Boutons de demande extérieure.....	52

CHAPITRE II :

Tableau II.1: Conditionnement des données d'entrée.....	32
TableauII.2: Régulateurs.....	32
Tableau II.3: Fonctions mathématiques.....	32
Tableau II.4 : Fonctions de la bibliothèque CONT_CTL.....	34
Tableau II.5 : Drivers de communication.....	34

CHAPITRE III :

Tableau III.4:la composition de la station automate.....	37
Tableau III.5 : les caractéristiques d'entrées/sorties TOR.....	40
Tableau III.6 : les caractéristiques d'entrées/sorties analogiques.....	41

Sommaire

Introduction général 2

Chapitre I Généralités sur les systèmes automatisés

I.1 Définitions..... 4

I.2 Structure générale d'un système automatisé de production..... 5

 I.2.1 La Partie Opérative 6

 I.2.2 La Partie Commande..... 6

 I.2.3 Interface Homme Machine..... 6

 I.2.4 Frontière PC - PO..... 6

I.3 Les automate Programmables Industriels 6

I.4 Objectifs..... 7

I.5 Structure des automates 8

I.6 Architecture interne d'un automate programmable 9

 I.6.1 L'unité centrale 10

 I.6.2 Le module d'entrées..... 12

 I.6.3 Le module de sorties 13

 I.7 Le module d'alimentation 14

I.8 Le module de communication..... 15

Chapitre II Logiciel de programmation Unity pro

II.1 Introduction..... 17

II.2 Cybersécurité 17

II.3 Fonction FDT/DTM..... 17

II.4 Langages de programmation..... 19

 II.4.1 Les cinq langages conformes à la norme IEC 61131-3..... 19

 II.4.1.1 Langage LL984 19

II.5 Fonction 19

 II.5.1 Editeur de données 19

 II.5.2 Blocs fonctions utilisateur DFBs..... 20

Sommaire

II.5.3 Bibliothèques des blocs fonctions	21
II.5.4 Outils de mise au point.....	23
II.5.5 Simulateur d'automate	25
II.5.6 Outil d'analyse des tendances	25
II.5.7 Editeur de documentation.....	25
II.5.8 Système "Time stamping"	26
II.5.9 Modification de programme, automate en exécution	27
II.5.11 Fonction import/export.....	28
II.5.12 Convertisseurs d'application Les outils	29
II.5.13 Utilitaires de mise à jour du système d'exploitation	29
II.5.14 Modification de configuration Quantum en ligne.....	30
II.5.15 Régulation de procédés dans les machines.....	30
II.6 Drivers de communication	34
II.6.1 Kits de mise à niveau pour logiciels Concept, PL7 Pro et ProWORX.....	35
II.6.2 Composition et compatibilité OS Windows®.....	35

Chapitre III Automates **Modicon M340**

III.1 Introduction	37
III.2 Présentation des stations automates Modicon M340.....	37
III.3 Présentation générale des composants d'une station automate	38
III.3.1 Présentation générale des processeurs	38
III.3.2 Présentation générale des racks.....	38
III.3.3 Présentation générale des modules d'alimentation.....	39
III.3.4 Présentation générale du module d'extension du rack	39
III.3.5 Présentation générale des modules d'entrées/sorties.....	40
III.3.6 Présentation générale des modules de comptage	41
III.3.7 Présentation générale de la communication.....	42
III.3.8 Mise à la terre des modules installés.....	42
III.4 Présentation générale des réseaux d'automates.....	43
III.4.1 Présentation générale du protocole Modbus	43

Sommaire

III.4.2	Présentation générale d'un réseau Ethernet	43
III.4.3	Présentation générale du bus de terrain CANopen	44
III.5	Normes et conditions de mise en service	45
III.5.1	Normes et certifications	45
III.5.2	Conformité aux normes et certifications	45
 Chapitre IV Commande et programmation de l'ascenseur 		
IV.1	Introduction	48
IV.2	Cahier des charges de fonctionnement	48
IV.3	Description de l'équipement	48
IV.3.1	Partie opérative	49
IV.3.2	Partie commande	50
IV.4	Conditions initiales	50
IV.5	Fonctionnement	51
IV.6	Repérage des entrée/sorties	53
IV.6.1	Capteurs	53
IV.6.2	Commandes manuelles	53
IV.6.3	Actionneurs	54
IV.7	Programmation	54
IV.7.1	Création d'un nouveau projet et déclaration de Rack automate	54
IV.7.2	Déclaration de la Configuration Matérielle de l'automate	55
IV.7.3	Déclaration des capteurs et des pré-Actionneurs dans "l'Editeur de Variables Elémentaire"	56
IV.7.4	Création d'une nouvelle section	57
IV.7.5.1	Configuration des étapes et transitions	59
IV.7.5.2	Les transitions se compose plus d'une condition	60
IV.7.5.3	Configuration des étapes	61
IV.8	Généation puis transfert le programme vers l'automate	67
Conclusion général		70

Introduction
générale

Introduction générale

L'automate Programmable Industriel API est aujourd'hui le constituant le plus répandu pour réaliser des automatismes. On le trouve pratiquement dans tous les secteurs de l'industrie car il répond à des besoins d'adaptation et de flexibilité pour un grand nombre d'opérations. Cette émergence est due en grande partie, à la puissance de son environnement de développement et aux larges possibilités d'interconnexions.

Les automates programmables industriels effectuent des tâches d'automatisation traduites sous formes de programme d'application qui définit la manière dont l'automate doit commander le système par une suite d'instructions, le programme doit être écrit dans un langage déterminé avec des règles définies pour que l'automate puisse l'exécuter, pour cela les automates de la famille SCHNEIDER sont programmés grâce au logiciel UNITY PRO via une console de programmation ou PC et sous un environnement WINDOWS.

Dans ce contexte, on s'est intéressé à la commande d'un ascenseur didactique 4 étages par API Schneider M340 à l'aide d'un logiciel de programmation UNITY PRO.

Ce mémoire est organisé autour de quatre chapitres ; le premier chapitre contient Généralités sur les systèmes automatisés, le deuxième chapitre présente la description de logiciel Unity Pro. Le troisième chapitre présente l'automate Modicon M340. Le quatrième chapitre est consacré de programmer l'ascenseur didactique.

Chapitre I

Généralités sur les systèmes automatisés

I.1 Définitions

L'automatisation d'un procédé (c'est-à-dire une machine, un ensemble de machines ou plus généralement un équipement industriel) consiste à en assurer la conduite par un dispositif technologique. Le système ainsi conçu sait prendre en compte les situations pour lesquelles sa commande a été réalisée. L'intervention d'un opérateur est souvent nécessaire pour assurer un pilotage global du procédé pour surveiller les installations et prendre en commande manuelle (non automatique) tout ou partie du système.

Les mesures des paramètres du procédé sont effectuées par des capteurs qui prélèvent l'information, la mettent en forme et souvent en transforment la nature physique. A l'opposé, les ordres de commande sont transmis au procédé par les actionneurs. Cette transmission est généralement accompagnée d'une transformation de la nature physique de l'information et d'un apport de puissance.[1]

L'organe de commande élabore les actions à partir des mesures et des consignes selon la loi de commande de l'automatisme.

Un système automatisé est un moyen d'assurer l'objectif primordial d'une entreprise et la compétitivité de ses produits. Il permet d'ajouter une valeur aux produits entrants.[2]

I.2 structure générale d'un système automatisé de production

La notion de système automatisé peut s'appliquer aussi bien à une machine isolée qu'à une unité de production, voire même à une usine ou un groupe d'usines. Il est donc indispensable, avant toute analyse, de définir la frontière permettant d'isoler le système automatisé étudié de son milieu extérieur.[3]

Un système automatisé de production comporte deux parties :

- Une *PARTIE OPERATIVE* (PO) dont les actionneurs (moteur électrique, vérin hydraulique,...) agissent sur le processus automatisé,
- Une *PARTIE COMMANDE* (PC) qui coordonne les différentes actions de la partie opérative.

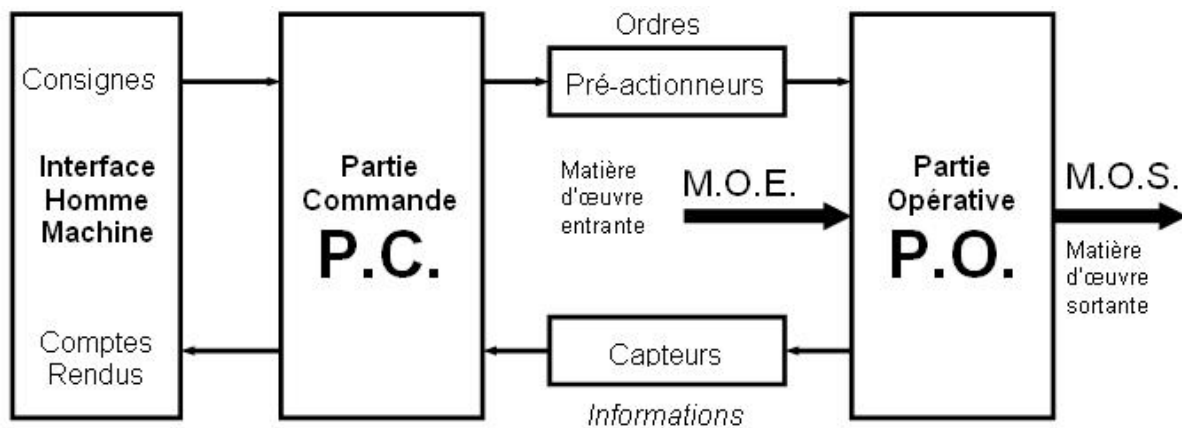


Figure I.1: structure générale d'un système automatisé de production

Les émissions d'ordres ou de signaux de commande vers la partie opérative sont transmises par les pré-actionneurs, les comptes rendus sont fournis à la partie commande par les capteurs.[4]

I.2.1 La Partie Opérative

C'est l'ensemble des dispositifs permettant d'apporter la valeur ajoutée. Elle met en œuvre un ensemble de processus physiques qui permettent la transformation de ces produits. Ces processus physiques nécessitent obligatoirement un apport d'énergie.

I.2.2 La Partie Commande

Automatiser la production consiste à transférer tout ou partie des tâches de coordination et des commandes, auparavant exécutées par des opérateurs humains, dans un ensemble d'objets techniques appelé Partie Commande.[1]

La partie commande reproduit le savoir-faire des concepteurs pour obtenir la suite des actions à effectuer sur les produits afin d'assurer la valeur ajoutée désirée. Pour ce faire elle émet des ordres vers la Partie Opérative et en reçoit, en compte rendu, un ensemble d'informations.[6]

I.2.3 Interface Homme Machine

Par ailleurs, la Partie Commande est en interaction avec son milieu extérieur par des liaisons informationnelles avec l'environnement humain, au travers de l'Interface Homme Machine (IHM).[7]

I.2.4 Frontière PC - PO

Les échanges d'informations entre la PC et la PO sont de deux types :

- émission d'ordres ou de signaux de commande vers des pré-actionneurs de la PO ;
- réception de comptes rendus par la PC par l'intermédiaire d'organes de saisie de l'information (capteurs).[8]

I.3 Les automate Programmables Industriels

Un automate programmable est une machine électronique, programmable par un personnel et destinée à piloter en ambiance industrielle et en temps réel des procédés automatiques.[4]

La définition d'un API d'après la norme NFC 63-850 est donnée par :

« Appareil électronique qui comporte une mémoire programmable par un utilisateur automatique (et non informaticien) à l'aide d'un langage adapté, pour le stockage interne des instructions composant les fonctions d'automatisme comme par exemple :

- Logique séquentielle et combinatoire .
- Temporisation, comptage, décomptage, comparaison .
- Calcul arithmétique .
- Réglage, asservissement, régulation, etc,

pour commander, mesurer et contrôler au moyen d'entrées et de sorties (logiques, numériques ou analogiques) différentes sortes de machines ou de processus, en environnement industriel. »[10]

I.3.1 Objectifs

La compétition économique impose à l'industrie de produire en qualité et en quantité pour répondre à la demande dans un environnement très concurrentiel. En terme d'objectifs il s'agit de :

- Produire à qualité constante,
- Fournir les quantités nécessaires au bon moment,
- Accroître la productivité et la flexibilité de l'outil.

les automates programmables industriels ou API comme on les appelle le plus souvent, sont apparus aux Etats-Unis vers 1969 ou ils répondaient aux désirs des industries de l'automobile de

développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des technologies et des modèles fabriqués. L'API s'est ainsi substitué aux armoires à relais en raison de sa souplesse, mais aussi parce que dans les automatismes de commande complexe, les coûts de câblage et de mise au point devenaient très élevés.[11]

Les premiers constructeurs américains étaient les entreprises Modicon et Allen-Bradley.

Les API offrent de nombreux avantages par rapport aux dispositifs de commande câblés, comme :

- Fiabilité
- simplicité de mise en œuvre
- Souplesse d'adaptation
- Maintenabilité
- Intégration dans un système de production [12]

I.3.2 Structure des automates

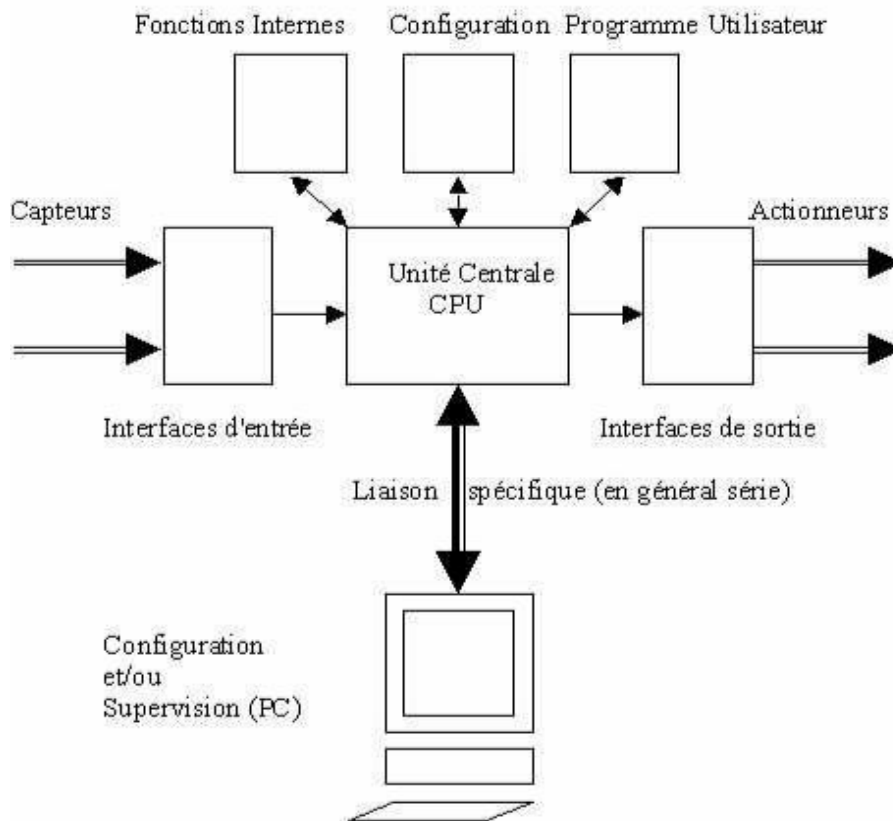


Figure I.2 : Structure des automates

- **Unité centrale** : elle gère l'ensemble du processus, elle contient le processeur, les mémoires vives et des mémoires mortes pour une taille débutant à 40 Koctets. Elle est programmable directement par console ou par le biais d'une liaison série et d'un logiciel adapté. Cette CPU peut être en RUN (elle exécute le programme) ou en STOP (toutes les sorties sont au repos, contacts ouverts).
- **Configuration** : elle contient les paramètres liés à la structure de l'API et à la structure du réseau informatique.
- **Fonctions Internes** : ce sont des fonctions pré-programmées livrées avec l'API qui permettent par exemple d'assurer des temporisations, des régulations... Ces fonctions peuvent être résidentes dans l'automate ou disponible dans le logiciel de programmation.
- **Programme Utilisateur** : c'est la loi de commande, il assure la gestion des sorties en fonction de l'état des entrées et éventuellement du temps. Ce programme est exécuté sous forme cyclique par l'API, le temps de cycle est dépendant de la taille du programme et ne doit pas excéder la centaine de millisecondes.
- **Supervision** : c'est un ordinateur standard. Il contient le logiciel de programmation (Orphée pour April et Step7 pour Siemens). Ce logiciel permet d'écrire le programme, de le compiler et de le transférer à l'automate. L'ordinateur peut également servir de poste opérateur pour assurer la conduite de l'unité. Un autre logiciel est alors nécessaire pour assurer le dialogue avec l'automate et une interface opérateur conviviale. Si la liaison entre le PC et la CPU est rompue, l'API continue de dérouler son programme.
- **Interfaces** : elles assurent le lien avec le procédé. Ces interfaces peuvent alimenter les boucles d'entrées ou de sorties, dans ce cas, l'automate sera doté d'une alimentation 24V continue. Elles peuvent être garnies de contacts secs, dans ce cas, une alimentation extérieure devra être intégrée aux boucles d'entrée et de sortie [13].

I.3.3 Architecture interne d'un automate programmable

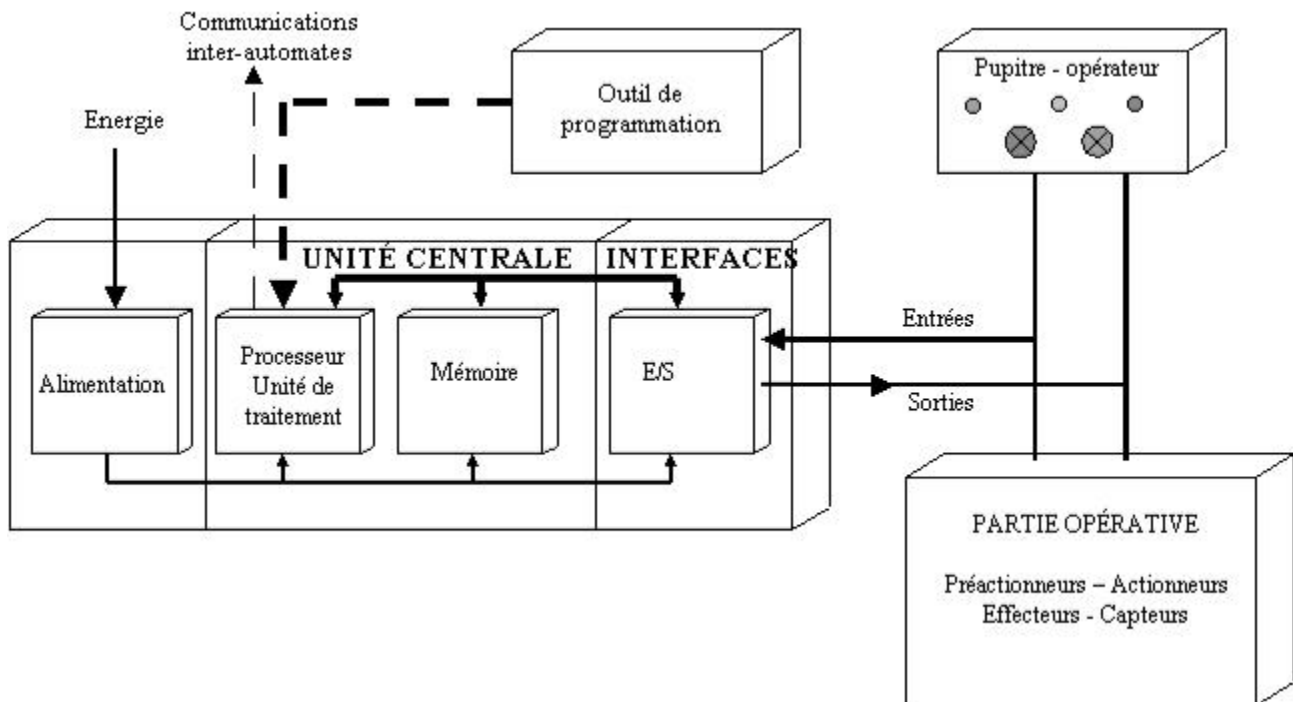


Figure I.3: Architecture interne d'un automate programmable

Un automate programmable est constitué essentiellement de 5 modules :

L'unité centrale :

L'unité centrale représente le cœur de la machine, et comprend le regroupement du processeur et de la mémoire centrale. Elle commande l'interprétation et l'exécution des instructions programmes. Les instructions sont effectuées les unes après les autres, séquencées par une horloge.[14]

❖ **Le processeur :**

Un processeur est l'unité fonctionnelle capable d'interpréter et d'exécuter les instructions du programme. Dans un API le processeur gère l'ensemble des échanges informationnels en assurant :

- ✓ La lecture des informations d'entrée.
- ✓ L'exécution des instructions du programme mis en mémoire.
- ✓ La commande ou l'écriture des sorties.

Pour réaliser ces différentes fonctions, le processeur se compose :

- d'une Unité Logique (UL) qui traite les opérations logiques ET, OU et Négation.[15]

- d'une Unité Arithmétique et Logique (UAL) qui traite les opérations de temporisation, de comptage et de calcul.[16]
- d'un Accumulateur qui est un registre de travail dans lequel se range une donnée ou un résultat.[17]
- d'un Registre d'Instruction qui contient, durant le temps de traitement, l'instruction à exécuter. [18]
- d'un Décodeur d'Instruction qui décode l'instruction à exécuter en y associant les microprogrammes de traitement. [19]
- d'un Compteur Programme ou Compteur Ordinal qui contient l'adresse de la prochaine instruction à exécuter et gère ainsi la chronologie de l'exécution des instructions du programme. [20]

❖ **La mémoire**

La mémoire centrale est l'élément fonctionnel qui peut recevoir, conserver et restituer. Elle est découpée en zones où l'on trouve :

- La zone mémoire programme (programme à exécuter) ;
- La zone mémoire des données (état des entrées et des sorties, valeurs des compteurs, temporisations) ;
- Une zone où sont stockés des résultats de calcul utilisés ultérieurement dans le programme ;
- Une zone pour les variables internes.

Ces mémoires peuvent être :

Durant la phase d'étude et de mise au point du programme :

- des mémoires vives RAM (Random Access Memory) volatiles
- des mémoires EAROM (Electrically Alterable Read Only Memory) non volatiles et effaçables partiellement par voie électrique.

Durant la phase d'exploitation:

- des mémoires vives RAM qui imposent un dispositif de sauvegarde par batterie rechargeable pour éviter la volatilité de leur contenu en cas de coupure de courant
- des mémoires mortes ROM à lecture seulement ou PROM programmables à lecture seulement.

- des mémoires reprogrammables EPROM (Erasable PROM) effaçables par un rayonnement ultraviolet et EEPROM (Electric Erasable PROM effaçables électriquement).[12]

I.3.3.1 Le module d'entrées

Un module d'entrées doit permettre à l'Unité Centrale de l'automate, d'effectuer une "lecture" de l'état logique des capteurs qui lui sont associés (module 4, 8, 16 ou 32 entrées). A chaque entrée correspond une voie qui traite le signal électrique pour élaborer une information binaire, le bit d'entrée qui est mémorisé. L'ensemble des bits d'entrées forme le "mot" d'entrées. Périodiquement, le Processeur de l'automate programmable vient questionner (adresser) le module: le contenu du mot d'entrées du module est alors recopié dans la mémoire DONNEES de l'automate programmable.[14]

Un module d'entrées est principalement défini par sa modularité (nombre de voies) et les caractéristiques électriques acceptées (tension, nature du courant...).

❖ Les cartes d'entrées logiques

Les cartes d'entrées logiques (cartes d'entrées tout ou rien) permettent de raccorder à l'automate les différents capteurs logiques tels que :

- boutons poussoirs
- fins de course
- capteurs de proximité inductifs ou capacitifs
- capteurs photoélectriques
- etc...

Elles assurent l'adaptation, l'isolement, le filtrage et la mise en forme des signaux électriques.

Une diode électroluminescente située sur la carte donne l'état de chaque entrée.[22]

❖ Les cartes d'entrées analogiques

Les cartes d'entrées analogiques permettent de gérer des grandeurs analogiques en faisant varier un code numérique au sein du module.

Les entrées analogiques disposent d'un seul convertisseur analogique /numérique, elles sont scrutées les unes à la suite des autres par un multiplexeur à relais.[23]

I.3.3.2 Le module de sorties :

Un module de sorties permet à l'automate programmable d'agir sur les actionneurs. Il réalise la correspondance: état logique signal électrique. Périodiquement, le processeur adresse le module et provoque l'écriture des bits d'un mot mémoire sur les voies de sorties du module.

L'élément de commutation du module est soit électronique (transistors, triac) soit électromécanique (contacts de relais internes au module).[24]

❖ *Les cartes de sorties logiques*

Les cartes de sorties logiques (tout ou rien) permettent de raccorder à l'automate les différents pré actionneurs tels que :

- Les contacteurs
- Les voyants
- Les distributeurs
- Les afficheurs...

Les tensions de sorties usuelles sont de 5 volts en continu ou de 24, 48, 110, 220 volts en alternatif.

Ces cartes possèdent soit des relais, soit des triacs, soit des transistors.

L'état de chaque sortie est visualisé par une diode électroluminescente.[25]

❖ *Les cartes de sortie analogiques*

Les cartes de sortie analogiques permettent de gérer des grandeurs analogiques en faisant varier un code numérique au sein du module. Ces modules assurent la conversion numérique/analogique.

Les sorties analogiques peuvent posséder un convertisseur par voie. Le nombre de voies sur ces cartes est de 2 ou 4.[26]

I.3.3.3 Le module d'alimentation

Composé de blocs qui permettent de fournir à l'automate l'énergie nécessaire à son fonctionnement. A partir d'une alimentation en 220 volts alternatif, ces blocs délivrent des sources de tension dont l'automate a besoin : 24V, 12V ou 5V en continu. En règle générale, un voyant positionné sur la façade indique la mise sous tension de l'automate.[27]

I.3.3.4 Le module de communication

Comprend les consoles, les boîtiers de tests et les unités de dialogue en ligne :

❖ *Les consoles*

Il existe deux types de consoles. L'une permet le paramétrage et les relevés d'informations (modification des valeurs, et visualisation), l'autre permet en plus la programmation, le réglage et l'exploitation. Cette dernière dans la phase de programmation effectue :

- L'écriture
- La modification
- L'effacement
- Le transfert d'un programme dans la mémoire de l'automate ou dans une mémoire REPRM.

La console peut également afficher le résultat de l'autotest comprenant l'état des modules d'entrées et de sorties, l'état de la mémoire, de la batterie, etc. Les consoles sont équipées souvent d'un écran à cristaux liquides. Certaines consoles ne peuvent être utilisées que connectées à un automate, d'autres peuvent fonctionner de manière autonome grâce à la mémoire interne et à leur alimentation.[10]

❖ *Les boîtiers de tests*

Destinées aux personnels d'entretien, ils permettent de visualiser le programme ou les valeurs des paramètres. Par exemple :

- Affichage de la ligne de programme à contrôler
- Visualisation de l'instruction (code opératoire et adresse de l'opérande)
- Visualisation de l'état des entrées
- Visualisation de l'état des sorties.[29]

❖ *Les unités de dialogue en ligne*

Elles sont destinées aux personnels spécialistes du procédé et non de l'automate programmable, elle leur permet d'agir sur certains paramètres :

- Modification des constantes, compteurs temporisations
- Forçage des entrées/sorties
- Exécution de parties de programme
- Chargement de programmes en mémoire à partir de cassettes.[30]

Chapitre II
Logiciel de
Programmation
Unity pro

II.1 Introduction

Unity Pro est le logiciel commun de programmation, mise au point et exploitation des gammes d'automates Modicon M340, M580, Premium, Momentum et Quantum. Unity Pro est un logiciel multitâche qui offre les fonctionnalités suivantes :

- Logiciel “tout en un”.
- 5 langages de programmation IEC 61131-3.
- Langage de programmation LL 984.
- Bibliothèque de blocs fonctions (DFB) intégrée et personnalisable.
- Simulateur automate sur PC pour valider votre programme avant installation.
- Tests intégrés (Built-in test) et diagnostic.
- Large gamme de services en ligne.
- Cybersécurité.

II.2 Cybersécurité

Schneider Electric a toujours veillé à la sécurité de ses systèmes. Des conseils sur la sécurité sont mis à disposition de nos clients afin d'assurer que leurs systèmes sont protégés des attaques.

Avec les plates-formes d'automatisme Modicon M340, M580, Quantum et Premium :

- protection contre les modifications de programmation à distance grâce à un mot de passe.
- option permettant d'activer ou de désactiver les services HTTP ou FTP.

Avec la plate-forme d'automatisme Modicon M580 :

- Historique des événements de sécurité dans la base de données SYSLOG.
- Gestion des services Ethernet étendus (DHCP, etc.) attribuable à chaque utilisateur dans la liste de contrôle d'accès.
- Sécurisation accrue de la communication IPSec entre Unity Pro ou SCADA et le régulateur. [31]

II.3 Fonction FDT/DTM

Unity Pro facilite l'intégration d'architectures de bus de terrain dans les systèmes de contrôle d'ingénierie à l'aide de la technologie FDT/DTM :

- FDT (Field DeviceTool) est le conteneur qui permet d'accueillir les DTM des équipements.

• DTM (Device Type Manager) est l'outil de configuration d'un équipement qui intègre ses propres interfaces graphiques. Il regroupe les propriétés qui définissent cet équipement.

Outre la norme FDT/DTM, Unity Pro utilise des informations spécifiques du Master DTM créé pour le module ProfibusRemote Master (PRM), le module HART, le module de pesage Premium ISPY101, l'intégration des variateurs AltivarProcess, les modules réseau Modbus/TCP, Ethernet/IP BMXNOC0401 et BMENOC03*1.

L'utilisation du Master DTM permet à Unity Pro d'effectuer les actions suivantes :

- Gérer la scrutation des entrées/sorties de l'automate.
- Créer les variables application grâce à la description des objets process disponibles à partir des équipements DTM connectés.
- Gérer la synchronisation avec la configuration de l'automate.
- Créer un DTM générique à partir des fichiers de description (GSD ou EDS).

Afin d'assurer le téléchargement complet de l'application, la configuration DTM est stockée dans la mémoire de l'automate. Elle est également enregistrée dans le fichier de projet de l'automate (STU), dans le fichier d'archive (STA) et dans le fichier complet d'échange de l'application (ZEF).

Un DTM tiers peut être installé dans le catalogue matériel DTM.

Le catalogue matériel DTM permet de trier ou de filtrer les DTM selon différents critères tels que Équipements, Fournisseurs, Groupes ou Protocoles.

À partir du navigateur DTM, Unity Pro :

- Affiche les topologies de bus de terrain dans une arborescence.
- Permet la configuration d'équipements DTM :
 - Ajout et suppression de DTM.
 - Connexion et déconnexion des DTM à/de leurs équipements physiques.
 - Affichage et impression des propriétés d'un DTM.
 - Transfert des informations de configuration DTM vers et depuis l'équipement physique.
 - Fonctions spécifiques du DTM, via le menu Device.

La recherche de bus de terrain est utilisée pour scruter les équipements physiques dans un réseau de bus de terrain et pour renseigner le navigateur DTM avec les équipements sélectionnés.[28]

II.4 Langages de programmation

II.4.1 Les cinq langages conformes à la norme IEC 61131-3

Les cinq langages de type graphiques ou textuels du logiciel Unity Pro permettent la programmation des plates-formes d'automatisme Modicon M340, Modicon M580, Modicon Momentum, Premium et Quantum.

- Les 3 langages graphiques sont :
 - Langage à contacts (LD).
 - Langage blocs fonctionnels (FBD).
 - Langage diagramme fonctionnel en séquence (SFC) ou Grafset.
- Les 2 langages textuels sont :
 - Langage littéral structuré (ST).
 - Langage liste d'instructions (IL).

Pour ces 5 langages, l'utilisation du jeu d'instructions de base conforme à la norme IEC 61131-3 permet de créer des applications portables d'une plate-forme sur une autre. De plus, le logiciel Unity Pro apporte des extensions à ce jeu d'instructions de base. Ces extensions spécifiques aux automates Modicon M340, Modicon M580, Modicon Momentum, Premium et Quantum autorisent le développement d'applications plus complexes et ainsi permettent de tirer profit des spécificités de chacune de ces plates-formes.

II.4.1.1 Langage LL984

Le langage LL984 (LadderLogic 984) permet la migration des anciennes gammes Modicon. Il est utilisé pour la programmation des plates-formes d'automatisation Modicon M580, Modicon M340, Momentum et Quantum.[31]

II.5 Fonction

II.5.1 Editeur de données

L'éditeur de données offre en un seul outil les services d'édition suivants :

- Déclaration des données incluant les variables et blocs fonctions (déclaration de leur type, leur instance et leurs attributs),

- Utilisation et archivage des types de données des blocs fonctions dans les différentes bibliothèques,
- Visualisation hiérarchique des structures de données,
- Recherche, tri et filtrage des données,
- Création d'un hyperlien au niveau du commentaire de toute variable pour accéder à un descriptif.

Les données sont présentées sous quatre onglets :

- Onglet "Variables" pour la création et la gestion des instances de données bits, mots, double mots, entrées/sorties, tableaux et structures.
- Onglet "DDT types" pour la création de types de données dérivées (tableaux et structures).
- Onglet "Function Blocks" pour la déclaration des blocs fonctions EFBs et DFBS.
- Onglet "DFB types" pour la création des types de données des blocs fonctions utilisateur DFBS.

Chaque donnée comporte plusieurs attributs parmi lesquels :

- le nom et le type de la variable sont obligatoires.
- le commentaire, l'adresse physique en mémoire ou les valeurs initiales sont optionnels.

Il est possible de configurer les colonnes de l'éditeur de données (nombre de colonnes, ordre). Une fenêtre de propriétés permet de visualiser l'ensemble des attributs associés à une variable.

Cet éditeur est accessible à tout moment en programmation par la sélection des variables permettant ainsi la modification ou la création de données.

II.5.2 Blocs fonctions utilisateur DFBS

Le logiciel Unity Pro offre à l'utilisateur la possibilité pour les plates-formes Modicon M340, Modicon M580, Modicon Momentum, Premium et Quantum, de créer ses propres blocs fonctions répondant aux spécificités de ses applications.

Une fois créés et mis en bibliothèque, ces blocs fonctions utilisateur pourront être réutilisés avec la même facilité que les blocs fonctions élémentaires EFBs.

Ces blocs fonctions utilisateur permettent de structurer une application. Ils sont utilisés dès qu'une séquence de programme se trouve répétée à plusieurs reprises dans l'application ou pour figer une programmation standard. Ils peuvent être protégés en lecture seule ou en lecture/écriture. Ils peuvent être distribués vers toutes autres applications Unity Pro.

L'utilisation d'un bloc fonction DFBs dans une ou plusieurs applications permet :

- de simplifier la conception et la saisie des programmes.
- d'accroître la lisibilité et la compréhension du programme.
- de faciliter sa mise au point (toutes les variables manipulées par le bloc fonction DFBs sont identifiées grâce à l'éditeur de données).
- d'utiliser des variables privées spécifiques aux DFBs donc indépendantes de l'application.

La mise en œuvre d'un bloc fonction DFBs s'effectue en différentes phases :

- La conception du DFBs se composant d'un nom, d'un ensemble de paramètres (entrées, sorties, variables internes publiques et privées) et d'un commentaire via l'éditeur de données.
- La création du code dans une ou plusieurs sections de programme avec, selon les besoins, le choix des langages : littéral structuré, liste d'instructions, à contacts ou blocs fonctionnels (ST, IL, LD ou FBD).
- Son rangement éventuel, avec un numéro de version associé, dans une bibliothèque.
- La création d'une instance DFBs dans l'éditeur de données ou lors de l'utilisation de la fonction dans l'éditeur de programme.
- L'utilisation de cette instance dans le programme de façon identique à un bloc fonction élémentaire EFB (la création de l'instance peut se faire à partir du programme).[29]

II.5.3 Bibliothèques des blocs fonctions

Le gestionnaire de bibliothèques de fonctions et de blocs fonctions regroupe l'ensemble des éléments fournis avec le logiciel Unity Pro. Les fonctions et les blocs fonctions sont organisés en bibliothèques comprenant elles-même des familles. Selon le type d'automate sélectionné et le modèle de processeur, l'utilisateur dispose d'un sous-ensemble de ces bibliothèques pour écrire ses applications. Toutefois, la bibliothèque "Base Lib" comporte l'ensemble des fonctions et des blocs fonctions dont la compatibilité, pour la majorité d'entre-eux, est indépendante des plates-formes. En particulier, elle comprend les blocs conformes à la norme IEC 61131-3.

La bibliothèque “Base Lib” est structurée en familles :

- Temporisateurs et compteurs
- Régulation sur entier
- Gestion de tableaux
- Comparaison
- Gestion du temps, date et heure
- Traitement logique
- Traitement mathématiques
- Traitement statistiques
- Traitement sur chaînes de caractères
- Conversion de type de données.

La bibliothèque “Base Lib”, qui répond aux fonctions standard d’automatismes, est complétée par d’autres bibliothèques plus orientées métiers et des fonctions dépendant des plates-formes:

- **Bibliothèque communication:** qui permet une intégration aisée des programmes de communication entre automates programmables et équipements de dialogue homme/machine à partir du programme d’application de l’automate. Comme les autres blocs fonctionnels, ces EFBs peuvent être utilisés dans tous les langages pour échanger des données entre automates, ou pour fournir des données à afficher sur l’équipement IHM.
- **Bibliothèque régulation :** La bibliothèque CONT_CTL permet la mise en place des boucles de régulation spécifiques au process. Elle offre notamment des fonctions de type régulateur, dérivé et intégrateur, et des algorithmes complémentaires, par exemple : EFBs pour le calcul de valeur moyenne, la sélection de valeur maximale, la détection de front, l’affectation d’une hystérésis sur des variables process...
- **Bibliothèque diagnostic :** qui permet la surveillance des actionneurs et contient les EFBs de type diagnostic sur action, diagnostic sur réaction, diagnostic d’interverrouillage, diagnostic sur conditions permanentes du procédé, diagnostic dynamique, surveillance de groupes de signaux...
- **Bibliothèque de gestion des entrées/sorties :** qui fournit des services gérer les informations échangées avec les modules matériels (formattage des données, mises à l’échelle...).
- **Bibliothèque “Motion Function Blocks” :** contenant un ensemble de fonctions et de structures de données prédéfinies pour gérer les mouvements commandés par les variateurs et servo-variateurs connectés via CANopen.

- **Bibliothèque de commande** : de mouvement et de comptage rapide.
- **Bibliothèque “System”** : qui fournit des EFBs dédiés à l’exécution de fonctions système : évaluation du temps de cycle, mise à disposition de diverses horloges du système, contrôle de section SFC, affichage de l’état du système, gestion de fichiers en cartouche mémoire de processeur Modicon M340, M580...
- **Enfin, une bibliothèque nommée “obsolète”** : comprend l’ensemble des blocs fonctions issus de logiciels de programmation antérieurs nécessaires pour assurer la conversion des applications.[26]

II.5.3.1 Gestion des standards utilisateurs

Les utilisateurs peuvent créer des bibliothèques et familles afin d’y archiver leurs propres blocs fonctions DFBs et structures de données DDTs. Cet enrichissement permet de bénéficier des standards de programmation adaptés avec une gestion de version. En effet, il est possible de :

- Vérifier la version des éléments utilisés dans un programme application par rapport à celle des éléments archivés en bibliothèque.
- Faire une mise à niveau, si nécessaire.

II.5.4 Outils de mise au point

Le logiciel Unity Pro offre un ensemble d’outils complet pour la mise au point des applications Modicon M340, M580, Momentum, Premium ou Quantum. Une palette d’outils permet d’accéder directement aux fonctions principales :

- Animation dynamique du programme.
- Pose de point de visualisation ou d’arrêt (non autorisée dans les tâches événementielles).
- Exécution pas à pas du programme. Une fonction de ce mode autorise une exécution section par section. L’exécution instruction par instruction est lancée à partir du point d’arrêt précédemment posé. Trois commandes d’exécution sont alors possibles lorsque l’élément à exécuter est un sous-programme SR ou une instance de bloc utilisateur DFB :
 - pas à pas détaillé “StepInto”, cette commande permet de se positionner sur le premier élément du SR ou DFB, v pas à pas principal “Step Over”, cette commande permet d’exécuter l’ensemble du SR ou DFB,
 - pas à pas sortant “Step Out”, cette commande permet de se positionner à l’instruction suivant l’élément SR ou DFB.

- Exécution indépendante des tâches maître MAST, rapide FAST, auxiliaires AUX et événementielle EVTi.

II.5.4.1 Animation des éléments du programme

L'animation dynamique est gérée section par section de programme. Un bouton proposé par la barre d'outils permet l'activation ou la désactivation de l'animation de chaque section. Lorsque l'automate est en exécution, ce mode permet de visualiser simultanément :

- l'animation d'une partie du programme quel que soit le langage utilisé.
- la fenêtre de variables contenant les objets de l'application créée automatiquement à partir de la section visualisée.

II.5.4.2 Table d'animation

Des tables contenant les variables de l'application à surveiller ou à modifier peuvent être créées par saisie ou automatiquement initialisées à partir de la portion de programme sélectionnée. Ces tables peuvent être sauvegardées dans l'application et ainsi être restituées lors d'une intervention ultérieure.

Unity Pro permet de sauvegarder, importer et exporter des tables d'animation avec des valeurs prédéfinies. Grâce à la population de valeurs définies en parallèle des valeurs usuelles de l'automate, et vice versa, les tables d'animation peuvent être utilisées comme modèle pour simplifier le réglage de l'application.

II.5.4.3 Mise au point des blocs fonctions utilisateur DFBs

Les paramètres et variables publiques de ces blocs sont affichés et animés en temps réel par l'intermédiaire des tables d'animation, avec la possibilité de modifier et forcer les objets souhaités. D'une manière identique aux autres composants programme, il est possible, afin d'analyser le comportement des blocs DFBs, d'utiliser les fonctions de point de visualisation ou d'arrêt, d'exécution pas à pas et de diagnostic du code programme. La pose d'un point d'arrêt dans une instance de bloc fonction utilisateur DFB stoppe l'exécution de la tâche incluant ce bloc.

II.5.4.4 Mise au point du diagramme fonctionnel de séquence (SFC)

Les différents outils de mise au point sont également disponibles en langage SFC. Cependant, une section SFC exécutée en pas à pas, contrairement aux autres sections (IL, ST, LD ou

FBD), ne stoppe pas l'exécution de la tâche mais fige l'évolution du graphe SFC. Plusieurs points d'arrêt peuvent être déclarés simultanément à l'intérieur d'une même section SFC.[24]

II.5.5 Simulateur d'automate

Le simulateur intégré au logiciel Unity Pro permet, à partir du terminal PC de tester le programme application automate Modicon M340, M580, Momentum, Premium ou Quantum et ce, sans faire appel à une connexion au processeur automate. Les fonctions offertes par les outils de mise au point sont disponibles pour la mise au point des tâches maître, rapide et auxiliaires. Le simulateur ne gérant pas les entrées/sorties de l'automate, l'utilisation des tables d'animation permet de simuler, par forçage à 1 ou à 0, l'état des entrées. Le simulateur peut être connecté à des applications tierces via un serveur OPC avec logiciel OFS (OPC Factory Server).

II.5.6 Outil d'analyse des tendances

L'outil d'analyse des tendances permet un contrôle simple des variables en détectant les problèmes d'exploitation ou en améliorant les performances de processus. Vous pouvez sélectionner toutes les variables de votre application et lancer une acquisition, effectuer des enregistrements et analyser les enregistrements à l'aide d'outils intégrés ou d'Excel. Jusqu'à 16 variables peuvent être scrutées au temps de cycle MAST de l'automate.

II.5.7 Editeur de documentation

L'éditeur de documentation s'articule autour du Navigateur Documentation qui montre sous forme arborescente la constitution du dossier. Il permet d'imprimer tout ou partie du dossier application sur toute imprimante graphique accessible sous Windows et utilisant la technologie True Type, en format d'impression A4 ou lettre US.

L'éditeur de documentation permet de constituer son propre dossier à partir des rubriques suivantes :

- Page de garde.
- Sommaire.
- Informations générales.
- Cartouche b Configuration.
- Blocs fonctions de type EFs, EFBs et DFBs.
- Variables utilisateur.
- Communication.
- Structure projet.

- Programme.
- Tables d'animation et références croisées.
- Ecrans d'exploitation.

II.5.8 Système "Time stamping"

Le mode système permet d'horodater les entrées/sorties ou les variables internes pour faciliter la gestion via OFS et SCADA. La programmation ou l'utilisation des blocs fonctions n'est pas obligatoire car la configuration est facile à réaliser dans l'éditeur de données Unity Pro supporté par OFS version 3.51

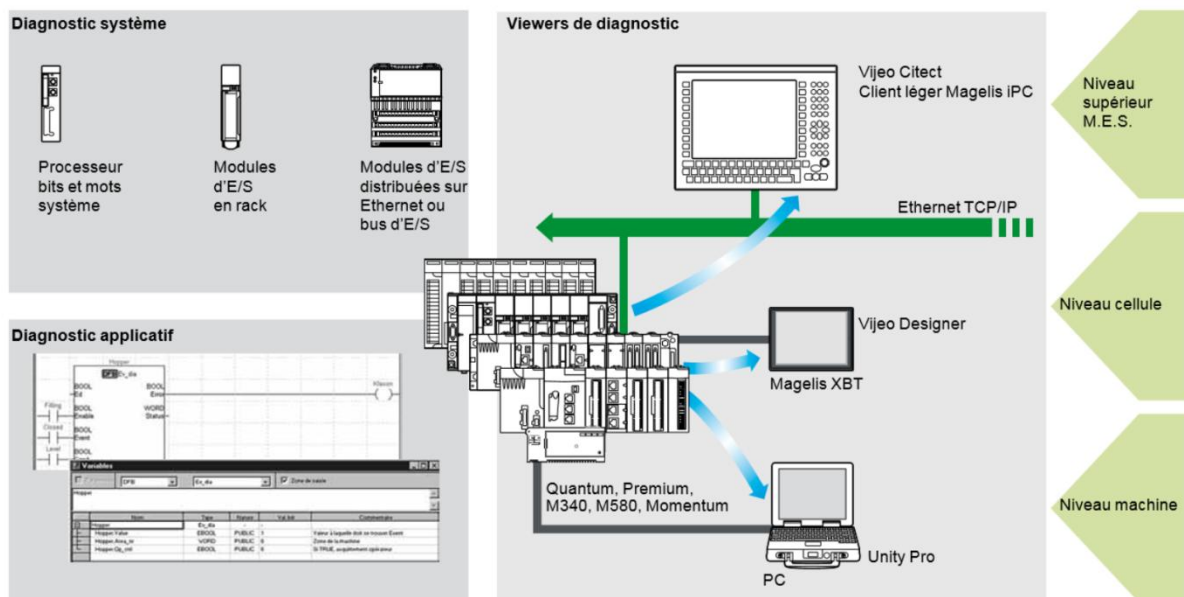


Figure II.1 : Diagnostics intégrés aux plates-formes d'automatisme Modicon M340, Modicon M580, Premium et Quantum

L'offre diagnostic des plates-formes Modicon M340, M580, Momentum, Premium et Quantum s'appuie sur trois composantes qui sont :

- Diagnostic système.
- Blocs fonctions DFBs et EFBs de diagnostic (système et application).
- Système de visualisation des messages d'erreur appelé "viewers" inclus de base dans les terminaux Magelis XBT, les logiciels de supervision VijeoCitect et les logiciels de mise en œuvre Unity Pro.

✓ Diagnostic système

Le diagnostic système des plates-formes Modicon M340, M580, Momentum, Premium et Quantum inclut la surveillance des bits/mots système, des modules d'entrées/sorties et des temps d'activité (mini/maxi) des étapes SFC. Par simple choix lors de la configuration de l'application, tout événement entraîne affichage automatique de messages horodatés enregistrés dans le buffer diagnostic dans l'automate. Ces événements sont visualisés automatiquement sur un viewer de diagnostic (1) sans aucune programmation supplémentaire. A l'aide du diagnostic intégré des logiciels Unity Pro, cette fonction permet le diagnostic de premier niveau des éléments constituant la configuration. Ce diagnostic est effectif jusqu'au niveau de la voie de chaque module d'E/S. [26]

II.5.9 Modification de programme, automate en exécution

Le logiciel Unity Pro permet d'effectuer des évolutions de programme lorsque l'automate connecté au terminal de programmation est en fonctionnement. Ces modifications s'effectuent en différentes opérations :

- Transfert éventuel de l'application contenue dans l'automate vers le terminal PC équipé du logiciel Unity Pro.
- Préparation des évolutions de programme. Ces modifications de programme peuvent être de tous types et dans tous les langages (IL, ST, LD, FBD et SFC), par exemple de type ajout/suppression d'étapes ou d'actions SFC. De plus, les modifications de code d'un bloc fonction utilisateur DFB sont possibles (cependant, les modifications de son interface ne sont pas autorisées).
- Mise à jour dans l'automate (en exécution) de ces évolutions de programme.

Cette fonction donne la possibilité d'ajouter ou de modifier du code programme et des données en différents endroits de l'application, et ce, en une seule session de modification (rendant ainsi cette modification homogène et cohérente par rapport au process contrôlé). Cette souplesse accrue entraîne un coût en termes de volume de mémoire de programme requis.

II.5.10 Fonction références croisées

La fonction références croisées du logiciel Unity Pro, disponible en mode autonome et en mode connecté à l'automate en exécution, permet de visualiser tous les éléments d'une appli-

cation automate en recherchant une variable de tout type. Cette visualisation indique où la variable déclarée est utilisée ainsi que la manière dont elle est utilisée (écriture, lecture...). Cette fonction donne également accès à la recherche/remplacement du nom des variables. La recherche de variables peut être initialisée à partir de tout éditeur (langage, données, écran d'exploitation, table d'animation...).

II.5.11 Fonction import/export

La fonction import/export, disponible sous les logiciels Unity Pro, permet à partir des vues structurelles et fonctionnelles du projet :

- par la fonction import, de réutiliser dans le projet en cours, l'ensemble ou seulement une partie d'un projet précédemment créé.
- par la fonction export, de copier dans un fichier l'ensemble ou seulement une partie du projet en cours, et ce à des fins de réutilisation.
- Les fichiers générés lors d'un export sont généralement au format XML (1).

Cependant, l'export ou l'import des variables peut, en plus du format XML, être réalisé sous :

- un format .xvm compatible avec le logiciel serveur de données OFS.
- un format source, dans un fichier .scy compatible avec le logiciel de conception PL7.
- un format texte avec séparateur (TAB), dans un fichier .txt pour compatibilité avec tout autre système.

Lors d'un import, un assistant permet de réaffecter les données sur de nouvelles instances de :

- blocs fonctions DFBs.
- structures de données DDTs.
- données simples.

De plus, lors de l'import d'un module fonctionnel, les données associées aux tables d'animation et aux écrans d'exploitation sont également réaffectées.

La fonction import XML permet, de plus, le transfert de configuration automate Modicon M340, Modicon M580, Premium ou Quantum établie dans l'outil de configuration et de chiffrage SIS Pro, en vue de la réalisation d'un projet dans le logiciel Unity Pro. Cet import évite à l'utilisateur de redéfinir la configuration automate lorsque celle-ci a déjà été réalisée avec l'outil SIS Pro.

Unity Pro inclut un outil import/export Excel qui permet de gérer simplement les variables en utilisant un fichier export XML dans Excel.

II.5.12 Convertisseurs d'application Les outils

Les conversions intégrées au logiciel Unity Pro permettent de convertir en application Unity Pro, les applications automatées créées avec les logiciels de programmation et de mise en œuvre ModSoft, ProWorX, Concept et PL7.

II.5.12.1 Convertisseur Concept/Unity Pro (automates Quantum et Momentum)

La conversion est assurée à partir d'une application Concept de niveau V2.5 (possible à partir de la version V2.11 mais après mise à jour en version V2.5). Elle nécessite un export, sous le logiciel Concept, de l'application en fichier ASCII. Automatiquement, le fichier d'export est converti en fichier source Unity Pro. Ce fichier source est ensuite analysé par Unity Pro. En fin de procédure, un rapport de conversion est généré et une fenêtre de sortie affiche les éventuelles erreurs de conversion détectées donnant un accès direct à la partie de programme à modifier. Le convertisseur d'applications Concept convertit l'application en Unity Pro mais ne garantit pas un fonctionnement. Il est donc indispensable d'effectuer des tests ou une mise au point de toute application convertie.

II.5.12.2 Convertisseur PL7/Unity Pro (automates Premium et slot-PLC Atrium)

La conversion est assurée à partir d'une application PL7 de niveau V4 minimum (automate Premium ou slot-PLC Atrium). Elle nécessite un export, sous le logiciel PL7, du fichier source (application complète ou bloc fonction utilisateur). La procédure de conversion est similaire à celle de la conversion de Concept décrite ci-dessus. Nota : La conversion d'applications créée avec Concept, Modsoft, ProWorx en LL984 est possible. Consultez notre centre de relations clients.

II.5.13 Utilitaires de mise à jour du système d'exploitation

Le logiciel OS-Loader est destiné à la mise à jour du système d'exploitation des plates-formes Premium et Quantum. Il est fourni avec le logiciel Unity Pro. Il permet de mettre à jour les processeurs et modules Unity ainsi que de mettre à niveau les processeurs PL7 et Concept pour les rendre compatibles avec Unity Pro.

Le logiciel OS-Loader supporte :

- les processeurs Premium.
- les processeurs Quantum.
- les modules de communication Ethernet.

- les modules de communication EtherNet/IP.

Les mises à jour de système d'exploitation s'effectuent avec :

- la liaison terminal Uni-Telway RS 485 pour processeurs Premium.
- la liaison terminal Modbus ou Modbus Plus pour processeurs Quantum.
- le réseau Ethernet TCP/IP pour port Ethernet intégré aux processeurs Premium et modules Ethernet Premium et Quantum.

II.5.14 Modification de configuration Quantum en ligne

Cette fonction, également dénommée Change Configuration On The Fly (CCOTF), permet la modification d'une configuration Modicon M580 ou Modicon Quantum en ligne (application en mode RUN) :

- ajout ou enlèvement de modules d'E/S TOR ou analogiques.
- modification des paramètres de configuration de modules d'E/S TOR ou analogiques déjà présents ou nouvellement installés.

Sur Quantum, la fonction CCOTF est supportée par les processeurs standalone pour les trois types d'architecture d'E/S (locales, RIO, DIO) à partir de la version 5 de Unity Pro, et pour les processeurs Hot Standby à partir de la version 4.1 de Unity Pro.

La fonction CCOTF doit être validée au préalable dans l'écran de configuration Unity Pro. Un écran de confirmation apparaît, lorsqu'une modification de configuration en ligne est réalisée.

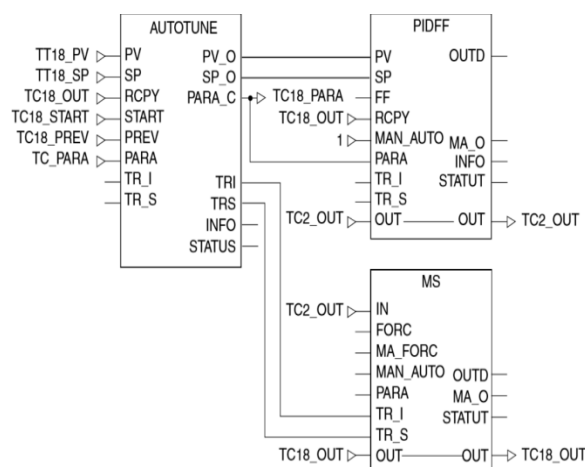
II.5.15 Régulation de procédés dans les machines

Unity Pro fournit CONT_CTL, une bibliothèque de 36 blocs fonctions pour la réalisation de boucles de régulation orientées vers la commande de machines.

La richesse fonctionnelle de la bibliothèque et la liberté d'association des blocs par programmation sont telles que toutes les fonctions de commande en boucle fermée dans les machines trouvent une solution adéquate avec les plates-formes d'automatisme Modicon M340, Modicon M580, Modicon Momentum, Premium et Quantum. La solution élimine ainsi le recours à des régulateurs externes, simplifiant l'architecture générale du contrôle de la machine, sa conception, son déploiement, et son exploitation.

Les blocs fonctions, EF ou EFB, peuvent être utilisés dans tous les langages de Unity Pro : LD, ST, IL, FBD. Ce dernier éditeur procure, sous Unity Pro, un accès particulièrement adapté aux traitements de régulation, grâce à l'assistant de saisie et de visualisation des paramètres et des variables des blocs.

Figure II.2 : Régulateur PID avec commande manuelle M



II.5.15.1 Fonctions de la bibliothèque CONT_CTL

La bibliothèque se compose de 6 familles de fonctions :

- Conditionnement des données d'entrée.
- Régulateurs.
- Fonctions mathématiques.
- Traitement des mesures.
- Traitement des valeurs de sorties.
- Traitement des valeurs de consigne

DTIME	Retard pur
INTEGRATOR	Intégrateur avec limitation
LAG_FILTER	Retard du premier ordre
LDLG	Action proportionnelle et dérivée avec lissage
LEAD	Action dérivée avec lissage
MFLOW	Calcul de débit/masse selon mesure de pression différentielle ou vitesse de flux avec compensation de pression et température
QDTIME	Terme de temps mort
SCALING	Mise à l'échelle
TOTALIZER	Intégrateur (typiquement de débit) jusqu'à ce qu'une limite soit atteinte (typiquement un volume) et réinitialisation automatique
VEL_LIM	Limiteur de variation, avec limitation de la grandeur de commande

Tableau II.1: Conditionnement des données d'entrée

PI_B	Régulateur PI simple : algorithme PI à structure mixte (série/parallèle)
PIDFF	Régulateur PID complet : algorithme PID à structure parallèle ou mixte (série/parallèle)
AUTOTUNE	Réglage automatique du régulateurPIDFF (PID complet) ou PI_B (PI simple): <ul style="list-style-type: none"> • Identification par méthodes de type Ziegler-Nichols. • Modélisation selon processus du premier ordre. • Génération des paramètres de régulation avec critère pour privilégier soit le temps de réaction aux perturbations (dynamique), soit la stabilité du procédé
.	IMC Régulateur modélisé. Le modèle est celui du premier ordre avec retard. Ce correcteur est intéressant : <ul style="list-style-type: none"> • lorsqu'il existe des retards purs importants par rapport à la constante de temps principale du procédé ; cas qui ne peut être traité de façon satisfaisante par une régulation PID classique. • pour réguler un procédé non linéaire. IMC peut traiter tout procédé stable et apériodique, d'ordre quelconque.
SAMPLETM	Contrôle du démarrage et de l'échantillonnage des régulateurs
STEP2	Régulateur simple à deux positions
STEP3	Régulateur à trois positions pour la régulation de température

TableauII. 2: Régulateurs

COMP_DB	Comparaison de deux valeurs, avec bande morte et hystérésis
K_SQRT	Racine carrée, avec pondération et seuil, utile pour la linéarisation des mesures de débit
MULDIV_W	Multiplication/Division pondérée de 3 grandeurs numériques
SUM_W	Addition pondérée de 3 grandeurs numériques

Tableau II.3: Fonctions mathématiques

II.5.15.2 Mise en œuvre des blocs fonctions de régulation

Basé sur l'enchaînement de blocs fonctionnels, le langage FBD intégré dans Unity Pro est un langage de programmation tout à fait adapté à la construction des boucles de régulation. Les concepteurs y associent facilement les blocs de la bibliothèque CONT_CTL avec leurs propres blocs DFB écrits en langage ST, IL, LD de Unity Pro, ou en langage C.[31]

II.5.15.3 Mise au point, exploitation

Tous les services de mise au point de Unity Pro sont disponibles. Le simulateur d'automate permet notamment de vérifier off-line la bonne exécution des traitements. [21]

II.5.15.4 Compatibilité

La bibliothèque des blocs fonctions de régulation CONT_CTL est disponible dans toutes les versions d'Unity Pro. Elle est compatible avec tous les processeurs des gammes Modicon M340, Modicon M580, Modicon Momentum, Premium et Quantum.

II.5.15.5 Bibliothèques spécialisées optionnelles

La bibliothèque des blocs fonctions de régulation CONT_CTL peut être complétée par des bibliothèques spécialisées optionnelles, permettant de répondre à des besoins spécifiques comme : contrôle prédictif, contrôleur logique floue, climatisation et ventilation (HVAC) et calcul de débit massique. [11]

II.5.15.6 Ressources

La documentation technique apporte un grand nombre d'exemples de mise en œuvre des blocs fonctions de régulation programmables dans les langages FBD, LD, IL, ST.

Les techniques de réglage des boucles de contrôle de procédé sont rappelées dans le document "Régulation de procédé, Unity V3.0" disponible sur le site www.schneider-electric.com. [24]

Traitement des mesures	
AVGMV	Moyenne glissante sur nombre d'échantillons fixe (50 maxi).
AVGMV_.	K Moyenne glissante avec facteur de correction constant, 10 000 échantillons maxi
DEAD_ZONE	"Dead zone" (Bande morte)
LOOKUP_TABLE1	Linéarisation de courbes caractéristiques par interpolation du premier ordre
SAH	Détection de front montant
HYST_XXX	Détection de seuil haut avec hystérésis
INDLIM_XXX	Détection de seuils haut et bas avec hystérésis
Traitement des valeurs de sorties	
MS	Commande manuelle d'une sortie
MS_DB	Commande manuelle d'une sortie avec bande morte
PWM1	Commande par modulation de largeur d'impulsion SERVO Commande de servo moteur
SPLRG	Commande à deux actionneurs "Split Range"
Traitement des valeurs de consigne	
RAMP	Générateur de rampe, rampes croissantes et décroissantes distinctes
RATIO	Régulateur proportionnel
SP_SEL	Sélection de la valeur de consigne : local (opérateur) ou "remote" (traitement)

Tableau II.4 : Fonctions de la bibliothèque CONT_CTL

II.6 Drivers de communication

Les drivers de communication les plus utilisés avec les plates-formes Modicon M340, M580, Momentum, Premium et Quantum sont installés en même temps que le logiciel Unity Pro

De plus, le logiciel Unity Pro inclut les drivers de communication suivants, à installer selon besoins :[31]

Protocole - Type de matériel	Windows® 7 Editions 32 et 64 bits	Windows® 8.1 Editions 32 et 64 bits
Adaptateur Fip - CUSBFIP		
Modbus Serial - Port COM		
PCIway - Atrium TPCI57 carte PCI		
Uni-Telway - Port COM		
USB pour automate de haut de gamme		
XIP - XWay sur TCP/IP		

Driver disponible
 Driver non disponible

Tableau II.4 : Drivers de communication

II.6.1 Kits de mise à niveau pour logiciels Concept, PL7 Pro et ProWORX

Les kits de mise à niveau pour logiciels Concept, PL7 Pro et ProWORX permettent aux utilisateurs, qui possèdent déjà un de ces logiciels dans leur parc installé, ainsi qu'un abonnement en cours, de s'équiper, à moindre coût, des logiciels Unity Pro version V4.1.

Ces mises à niveau ne sont possibles que pour des licences de même niveau (de licence groupe Concept XL vers licence groupe Unity Pro Extra Large).[25]

II.6.2 Composition et compatibilité OS Windows®

Les logiciels multi langues Unity Pro sont compatibles avec les systèmes d'exploitation Windows 7 (32 et 64 bits), Windows 8.1 (32 et 64 bits) et Windows Server 2012 (64 bits). Un système d'exploitation Windows 64 bits avec une mémoire RAM de 4 Go minimum est préconisé pour les projets utilisant des DTM. Ils comprennent :

- la documentation sous forme électronique en 6 langues (allemand, anglais, chinois, espagnol, français et italien).
- les convertisseurs permettant la conversion des applications créées avec les logiciels de conception et de mise en œuvre Concept et PL7 Pro.
- le simulateur d'automate. Les câbles de raccordement processeur au PC de programmation sont à commander séparément[31]

Chapitre III

Automates

Modicon M340

III.1 Introduction

Cette section présente de façon générale les configurations automate Modicon M340, les différents sous-ensembles pouvant les composer, ainsi que les réseaux et bus de terrain utilisés.

III.2 Présentation des stations automates Modicon M340

Station automate Modicon M340

Les processeurs de plate-forme automatisée Modicon M340 gèrent l'ensemble de la station automate, qui se compose de modules d'entrée/sortie TOR, de modules d'entrée/sortie analogiques, de modules de comptage, de modules experts et de modules de communication. Ces modules sont répartis sur un ou plusieurs racks raccordés au bus local. Chaque rack doit comporter sa propre alimentation ; le rack principal accueille l'unité centrale

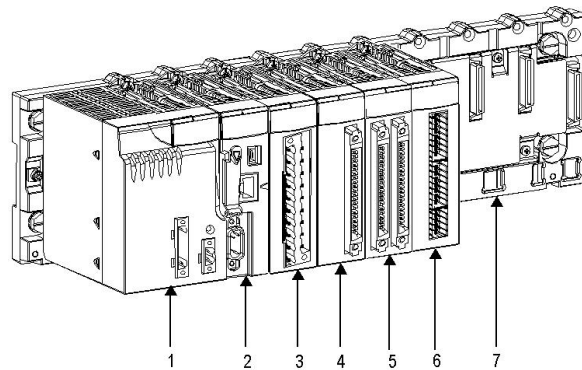


Figure III.1 : exemple de configuration de la station automate Modicon M340 avec un rack

Repère	Description
1	Module d'alimentation
2	Processeur
3	Module d'entrées/sorties à bornier 20 points
4	Module d'entrées/sorties à 1 connecteur 40 points
5	Module d'entrées/sorties à 2 connecteurs 40 points
6	Module de comptage
7	Rack à 8 emplacements

Tableau III.1 : la composition de la station automate

III.3 Présentation générale des composants d'une station automate

III.3.1 Présentation générale des processeurs

Chaque station automate est pourvue d'un processeur, choisi en fonction des caractéristiques suivantes :

- puissance de traitement (nombre d'entrées/sorties gérées).
- capacité mémoire.
- ports de communication.

Pour plus d'informations, reportez-vous à la *Présentation des processeurs BMX P34 xxxx*.

III.3.2 Présentation générale des racks

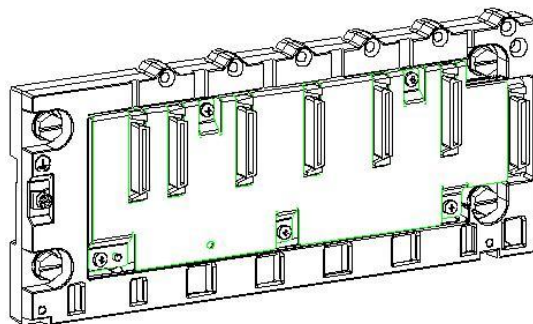
Il existe quatre formats de racks, choisis selon le nombre de modules que vous souhaitez utiliser.

- Rack BMX XBP 0400 (4 emplacements)
- Rack BMX XBP 0600 (6 emplacements)
- Rack BMX XBP 0800 (8 emplacements)
- Rack BMX XBP 1200 (12 emplacements)

La liste ci-dessus donne le nombre d'emplacements disponibles.

Chaque rack inclut un emplacement supplémentaire réservé au module d'alimentation et un emplacement sur la droite est réservé au module d'extension de rack BMX XBE 1000.

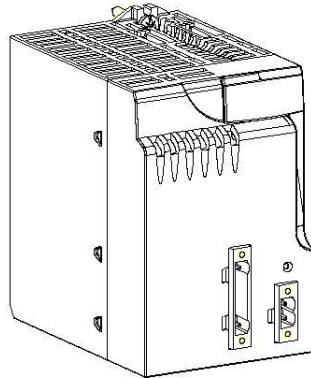
Pour plus d'informations, reportez-vous à la présentation des racks.



**Figure III.2 :rack BMX XBP
0400**

III.3.3 Présentation générale des modules d'alimentation

Chaque rack nécessite 1 module d'alimentation défini en fonction du réseau distribué (courant alternatif ou courant continu) et de la puissance nécessaire au niveau du rack.



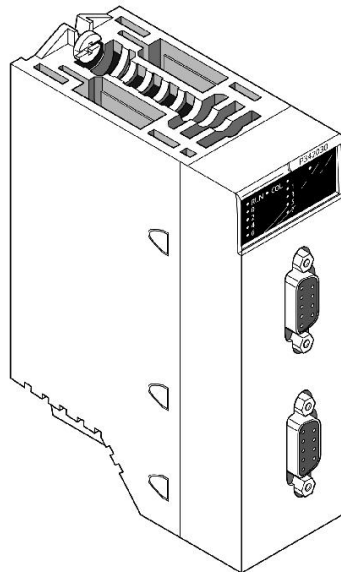
**Figure III.3 : un module
d'alimentation BMX CPS**

...

III.3.4 Présentation générale du module d'extension du rack

Ce module permet de connecter un maximum de 4 racks en chaîne, selon l'unité centrale, répartis sur une longueur maximale de 30 mètres.

Consultez la section Module d'extension de rack.



**Figure III.4 : module d'extension
de rack BMX XBE 1000**

III.3.5 Présentation générale des modules d'entrées/sorties

La gamme Modicon M340 est composée de modules d'entrées/sorties TOR et analogiques.

III.3.5.1 Entrées/sorties TOR

Une large gamme de modules d'entrées/sorties TOR permet de s'adapter aux mieux à vos besoins. Ces modules se différencient par les caractéristiques suivantes :

Caractéristiques	Description
Modularité	8 voies 16 voies 32 voies 64 voies
Type d'entrées	Modules avec entrées à courant continu (24 V cc et 48 V cc) Modules avec entrées à courant alternatif (24 V ca, 48 V ca et 120 V ca)
Type de sorties	Modules avec sorties à relais Modules avec sorties statiques à courant continu (24 V cc/0,1 A - 0,5 A - 3 A) Modules avec sorties statiques à courant alternatif (24 V ca/240 V ca/3 A)
Type de connectique	Borniers 20 points Connecteurs de type 40 points permettant le raccordement aux capteurs et aux pré-actionneurs par l'intermédiaire du système de précâblage TELEFAST 2

Tableau III.2 : les caractéristiques d'entrées/sorties TOR

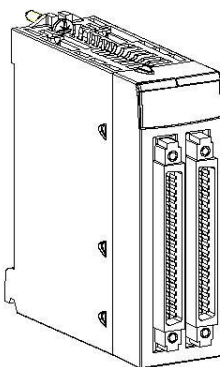


Figure III.2 : un module d'entrées/sorties TOR avec connecteurs 40 points

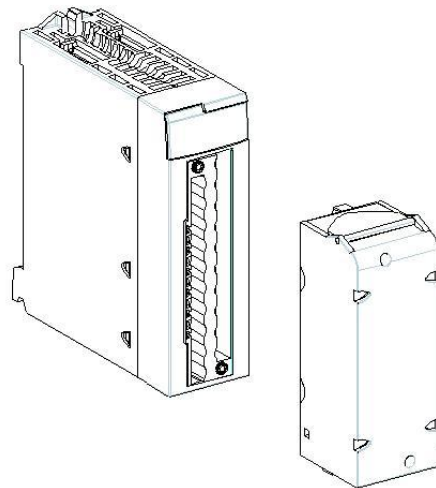


Figure III.6 : un module d'entrées/sorties TOR avec bornier 20 points

III.3.5.2 Entrées/sorties analogiques

Une large gamme de modules d'entrées/sorties analogiques permet de s'adapter aux mieux à vos besoins. Ces modules se différencient par les caractéristiques suivantes :

Caractéristiques	Description
Modularité	2 voies 4 voies
Performances et gammes de signaux proposés	Tension/courant Thermocouple Thermosonde
Type de connectique	Borniers 20 points Connecteurs de type 40 points permettant le raccordement aux capteurs et aux pré-actionneurs par l'intermédiaire du système de précâblage TELEFAST 2

Tableau III.3 : les caractéristiques d'entrées/sorties analogiques

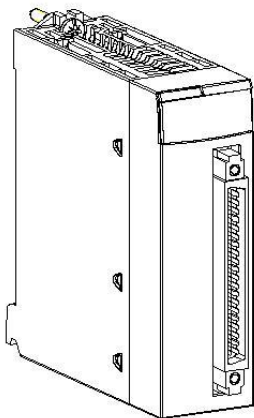


Figure III.7 : un module d'entrées/sorties analogique avec un connecteur 40 points

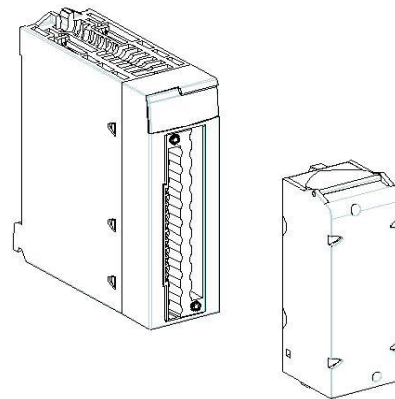


Figure III.8 : un module d'entrées/sorties analogique avec bornier 20 points

III.3.6 Présentation générale des modules de comptage

Les automates de la gamme Modicon M340 proposent des fonctions de comptage (décomptage, comptage, comptage/décomptage) grâce aux modules métiers de comptage.

Deux modules de comptage sont disponibles :

- module BMX EHC 0200 à 2 voies de comptage avec fréquence maximale d'acquisition de 60 kHz,
- module BMX EHC 0800 à 8 voies de comptage avec fréquence maximale d'acquisition de 10 kHz.

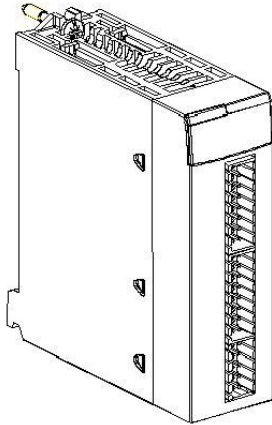


Figure III.9 :le module de comptage BMX EHC 0200

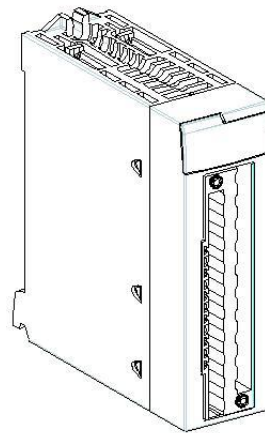


Figure III.10 :le module de comptage BMX EHC 0800

III.3.7 Présentation générale de la communication

Les automates de la gamme Modicon M340 peuvent être utilisés avec différents modes de communication :

- USB
- Série
- Ethernet
- CANopen
- AS-Interface

III.3.8 Mise à la terre des modules installés

La mise à la terre des modules Modicon M340 est indispensable pour éviter tout choc électrique.

Tous les modules Modicon M340 possèdent des contacts de terre en face arrière pour la mise à la terre.

Ces contacts connectent le bus de mise à la terre des modules au bus de mise à la terre du rack.

III.4 Présentation générale des réseaux d'automates

III.4.1 Présentation générale du protocole Modbus

Le protocole Modbus est un protocole créant une structure hiérarchisée (un maître et plusieurs esclaves).

Le maître gère l'ensemble des échanges selon deux types de dialogues :

- le maître échange avec l'esclave et attend la réponse,
- le maître échange avec l'ensemble des esclaves sans attendre de réponse (diffusion générale).

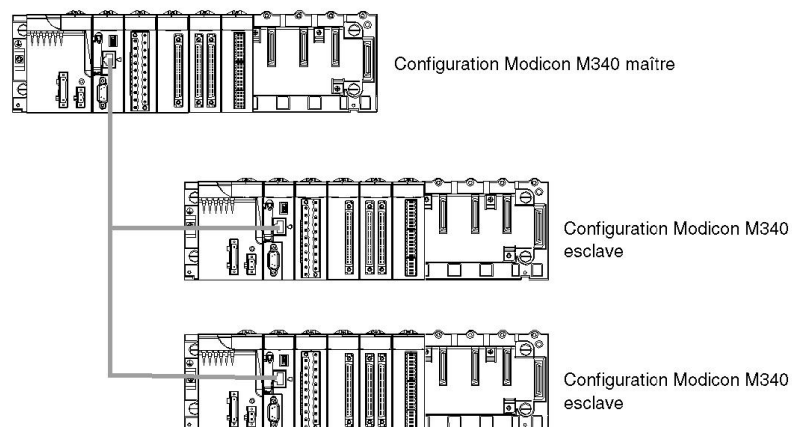


Figure III.11 :réseau Modbus

III.4.2 Présentation générale d'un réseau Ethernet

La communication Ethernet vise essentiellement les applications de :

- coordination entre automates programmables,
- supervision locale ou centralisée,
- communication avec l'informatique de gestion de production,
- communication avec les entrées/sorties distantes. La communication Ethernet supporte également, en fonction agent, la gestion du standard de supervision réseau SNMP.

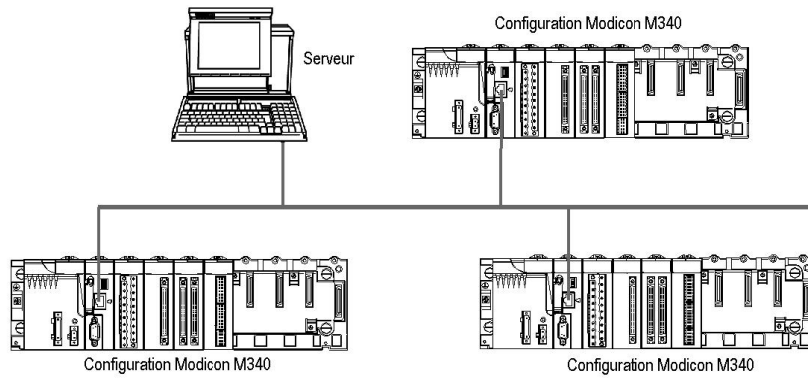


Figure III.12 : un réseau Ethernet

III.4.3 Présentation générale du bus de terrain CANopen

Une architecture CANopen comprend :

- un maître du bus,
- des équipements esclaves appelés aussi nœuds.

Le bus fonctionne selon un mode d'échange point à point. A tout moment, chaque équipement peut envoyer une requête sur le bus et les équipements concernés répondent.

La priorité des requêtes circulant sur le bus est déterminée par un identifiant au niveau de chaque message.

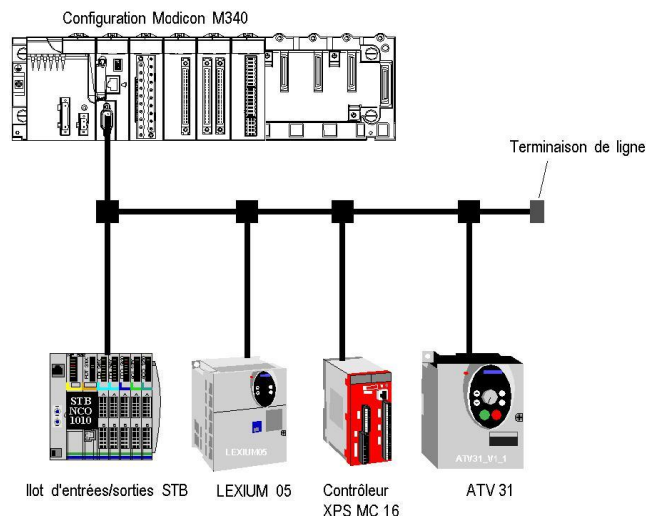


Figure III.13 : une architecture de bus de terrain CANopen

III.5 Normes et conditions de mise en service

III.5.1 Normes et certifications

Les automates Modicon M340 ont été conçus pour être conformes aux normes et règles appropriées applicables aux équipements électriques destinés à l'industrie.

III.5.2 Conformité aux normes et certifications

Les normes et certifications respectées par les automates Modicon M340 sont les suivantes :

- Exigences spécifiques aux automates programmables qui concernent les caractéristiques fonctionnelles, l'immunité, la robustesse et la sécurité :
 - CEI 61131-2 Ed. 2 (2003)
 - CSA 22.2 N° 142
 - UL 508
- Exigences pour la marine marchande des principales organisations internationales :
 - ABS
 - BV
 - DNV
 - GL
 - LR
 - RINA
 - RMRS
- Directives européennes :
 - Basse tension : 72/23/CEE amendement 93/68/CEE
 - Compatibilité électromagnétique : 89/336/CEE amendements 92/31/CEE et 93/68/CEE
- Exigences concernant les zones dangereuses :
- CSA 22.2 N 213, classe 1, division 2, groupes A, B, C et D.
 - Cet équipement ne peut être utilisé que dans les zones dangereuses de classe 1, division 2, groupes A, B, C et D ou dans les zones dangereuses non classées (voir AVERTISSEMENT ci-dessous).
- Règles ACA (pour C-Tick)
- Règles CEI/ECO (pour GOST)
- Respect de l'environnement :

- Directive européenne 2002/95/CE relative à la limitation de l'utilisation de certaines substances dangereuses (RoHS). Ne contient pas de plomb, mercure, chrome hexavalent, PBB ou PBDE.
- Directive européenne 2002/96/CE relative aux déchets d'équipements électriques et électroniques (DEEE)
- Directives Schneider Electric (pas de matières halogènes, augmentation du taux de recyclabilité, etc.)

***Chapitre IV Com-
mande et programmation
de l'ascenseur***

IV.1 Introduction

Pour concevoir et commander l'ascenseur, on essayera, dans ce chapitre d'éclaircir notre travail, on procédera étape par étape et en séparant ses différentes parties. On commencera par le cahier des charges de l'ascenseur envisagé, puis on parlera la partie opérative et la partie commande, ensuite on présentera les différentes étapes de programmation de l'ascenseur.

IV.2 Cahier des charges de fonctionnement

On considère un ascenseur desservant quatre étages et dont la boîte d'appel contient deux boutons de demande.

Le principe de ce type d'ascenseur consiste à optimiser la demande d'ascenseur et à améliorer l'utilisation en permettant la demande et la mémorisation de la demande de la cabine quand elle est en marche.

Un utilisateur désirant « Monter » appuie sur le bouton dont la flèche est dirigée vers le haut, un utilisateur désirant « Descendre » appuie sur le bouton dont la flèche est dirigée vers le bas. Si la direction de l'ascenseur est la même que celle demandée, et que l'étage d'où provient la demande n'est pas encore atteint.

Au niveau 5 il y a un bouton poussoir lumineux appel cabine pour descendre.

Aux niveaux 2, 3 et 4 il y a, un bouton poussoir lumineux appel cabine pour descendre et un bouton poussoir lumineux appel cabine pour monter.

Au niveau 1 il a un bouton poussoir lumineux appel cabine pour monter.

Sur le socle support il a cinq boutons poussoirs lumineux représentant les boutons se trouvant dans la cabine pour donner l'ordre de destination et un bouton arrêt d'urgence à accrochage mécanique et déverrouillage par clé.

IV.3 Description de l'équipement

L'ensemble ascenseur est composé de deux parties.

- La partie opérative : représentant un ascenseur monté sur un support et intégrant les actionneurs, les capteurs et les éléments de dialogues (boutons et voyants) nécessaires à son fonctionnement.
- La partie commande : intégrant l'automate M340 avec son programme ainsi que l'alimentation 24 Vcc.

L'automatisme est réalisé par un automate programmable M340.

IV.3.1 Partie opérative

L'ensemble fonctionne comme un ascenseur d'immeuble sur cinq niveaux, il est monté sur un socle support, il comprend les éléments suivants.

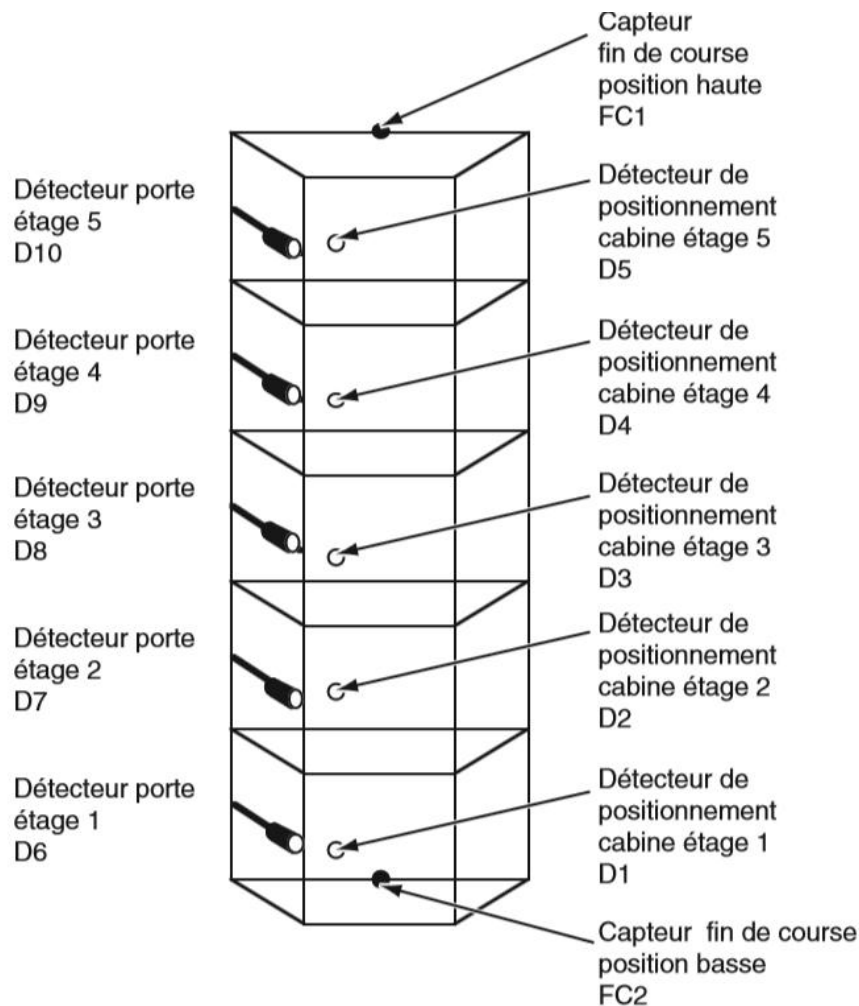


Figure IV.1 : Socle support de l'ascenseur

- Un capteur de sécurité "Position haute".
- Un capteur de sécurité "Position basse".
- Un capteur de présence cabine à l'étage et un capteur de contrôle de la fermeture de la porte palière.
- Un capteur de présence cabine à l'étage et un capteur de contrôle de la fermeture de la porte palière
- Un capteur de présence cabine à l'étage et un capteur de contrôle de la fermeture de la porte palière.
- Sur l'arrière du support deux connecteurs "sub D " mâles, un de 37 broches pour le raccordement des entrées, l'autre de 25 broches pour le raccordement des sorties.

- A l'intérieur du support quatre relais :
 - deux relais pour l'inversion du sens de marche de la cabine,
 - deux relais en relayage des capteurs de fin de course de sécurité haut et bas.

IV.3.2 Partie commande

Sur un support sont montés :



Figure IV.2 : automate Modicon M340

- Un automate Modicon M340, équipé d'une alimentation, d'un processeur avec une carte mémoire et de deux cartes 32 entrées et 32 sorties TOR.
- Une alimentation 24V DC 4,2A suffisamment puissante pour alimenter les capteurs, les afficheurs et les actionneurs de l'équipement.
- Un commutateur 2 positions "Run/Stop".
- Un voyant blanc "Sous tension".
- Sur le côté deux connecteurs (25 et 37 broches) pour le raccordement à la partie opérative.
- Une prise type "europa" équipée d'un fusible de protection pour le raccordement au réseau par le cordon secteur fourni.

IV.4 Conditions initiales

- Il n'y a pas de position particulière au démarrage pour faire fonctionner la cabine en montée ou en descente, toutefois toutes les portes palières doivent être fermées et le bouton coup de poing arrêt d'urgence en position déverrouillé.
- Chaîne de sécurité
 - Un bouton coup de poing "arrêt d'urgence" à accrochage mécanique, déverrouillage par rotation horaire du bouton, est équipé de deux contacts à ouvertures :
 - Le premier coupe les préactionneurs, stoppant ainsi immédiatement tout mouvement de la cabine.
 - Le second informe l'automate.

- L'ouverture d'une porte palière quand la cabine n'est pas présente au palier, stoppe ou interdit le fonctionnement de l'ascenseur
- Dans tous les cas d'arrêt décrit ci-dessus l'afficheur sera éteint.

IV.5 Fonctionnement

- Dès que les raccordements sont effectués :
 - Le voyant blanc "sous tension" situé sur la partie commande s'allume,
 - Le voyant "Run/Stop" de l'automate passe au vert clignotant.

L'ensemble "Ascenseur" est en mesure de fonctionner.

- Utilisation :
 - Après avoir positionné le commutateur "Stop/Run" sur "Run", le voyant vert de l'automate passe au vert fixe.



Figure IV.3:Commutateur "Stop/Run"

- L'accès au programme automate ou à la mémoire de donnée est possible en connectant un micro-ordinateur équipé du logiciel UNITY-PRO sur la prise USB ou RJ 45 (Ethernet) du M340.
- Pour faire fonctionner l'ascenseur le bouton coup de poing arrêt d'urgence doit être en position déverrouillé et les portes palières fermées.
- L'ensemble fonctionne de façon identique à un ascenseur réel.
- L'ordre vient d'un palier

L'appel vient d'un bouton pour monter (↑) ou pour descendre (↓) se trouvant sur l'un des paliers, l'ascenseur se déplacera en fonction de sa position pour rejoindre le lieu de l'appel. Si plusieurs appels sont effectués, il les traitera de façon logique. Pour rejoindre un appel venant d'un palier supérieur il s'arrêtera en montant à tous les paliers ayant envoyés un ordre pour monter. Il traitera les ordres de descente quand il retournera vers les niveaux inférieurs.

- L'ordre vient de la cabine

Les boutons poussoirs repérés 1 à 5 représentent les boutons de la cabine de l'ascenseur, ils donnent le niveau de destination. L'ordre sur l'un de ces boutons est pris en mémoire, l'ascenseur démarre en montée ou en descente pour rejoindre le niveau désiré.

Pendant son déplacement il s'arrêtera, s'il se dirige vers un niveau supérieur aux étages qui auraient émis un ordre (↑) pour monter, ou s'il se dirige vers un niveau inférieur aux étages qui auraient émis un ordre (↓) pour descendre.

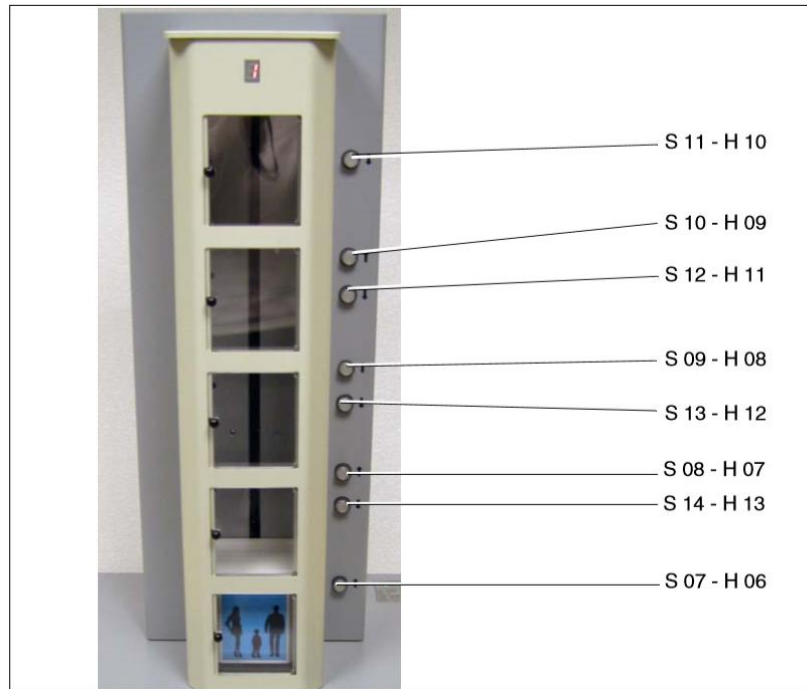


Figure IV.4 : Boutons de demande extérieure

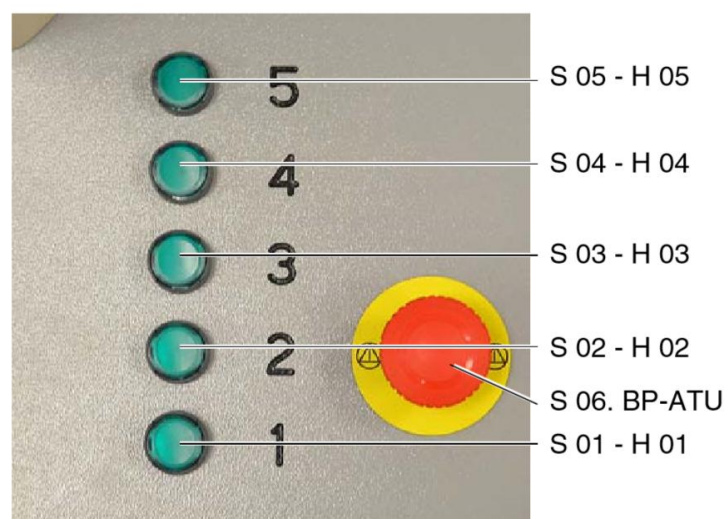


Figure IV.5 : Boutons de demande extérieure

IV.6 Repérage des entrée/sorties

IV.6.1 Capteurs

Désignation	Nom	Adresse	Type
FC1 - Fin de course haut	Fdc_haut	%I0.1.7	EBOOL
FC2 - Fin de course bas	Fdc_bas	%I0.1.1	EBOOL
D1 - Détecteur étage 1	Det_et_1	%I0.1.2	EBOOL
D2 - Détecteur étage 2	Det_et_2	%I0.1.3	EBOOL
D3 - Détecteur étage 3	Det_et_3	%I0.1.4	EBOOL
D4 - Détecteur étage 4	Det_et_4	%I0.1.5	EBOOL
D5 - Détecteur étage 5	Det_et_5	%I0.1.6	EBOOL
D6 - Détecteur porte étage 1	Porte_et_1	%I0.1.14	EBOOL
D7 - Détecteur porte étage 2	Porte_et_2	%I0.1.15	EBOOL
D8 - Détecteur porte étage 3	Porte_et_3	%I0.1.20	EBOOL
D9 - Détecteur porte étage 4	Porte_et_4	%I0.1.21	EBOOL
D10 - Détecteur porte étage 5	Porte_et_5	%I0.1.22	EBOOL

IV.6.2 Commandes manuelles

Désignation	Nom	Adresse	Type
Run/Stop	run_stop	%I0.1.0	EBOOL
S1 - Demande étage 1 en cabine	Ap_cab_1	%I0.1.17	EBOOL
S2 - Demande étage 2 en cabine	Ap_cab_2	%I0.1.18	EBOOL
S3 - Demande étage 3 en cabine	Ap_cab_3	%I0.1.19	EBOOL
S4 - Demande étage 4 en cabine	Ap_cab_4	%I0.1.12	EBOOL
S5 - Demande étage 5 en cabine	Ap_cab_5	%I0.1.13	EBOOL
S6 - Arrêt d'urgence	Bp_atu	%I0.1.16	EBOOL
S7 - Appel étage 1 pour monter	Ap_mo_1	%I0.1.23	EBOOL
S8 - Appel étage 2 pour monter	Ap_mo_2	%I0.1.28	EBOOL
S9 - Appel étage 3 pour monter	Ap_mo_3	%I0.1.29	EBOOL
S10 - Appel étage 4 pour monter	Ap_mo_4	%I0.1.30	EBOOL
S11 - Appel étage 5 pour descendre	ap_des_5	%I0.1.11	EBOOL
S12 - Appel étage 4 pour descendre	ap_des_4	%I0.1.10	EBOOL
S13 - Appel étage 3 pour descendre	ap_des_3	%I0.1.9	EBOOL
S14 - Appel étage 2 pour descendre	ap_des_2	%I0.1.8	EBOOL

Remarque : L'entrée %I0.1.0 est configurée en entrée RUN/STOP

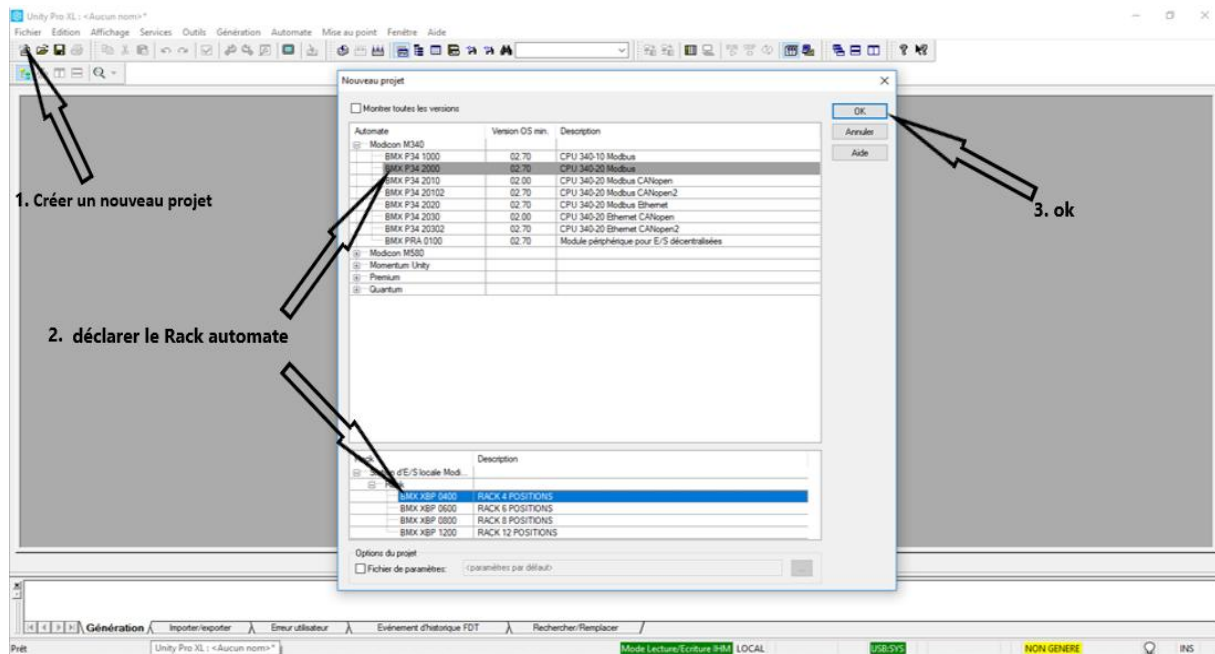
IV.6.3 Actionneurs

Désignation	Nom	Adresse	Type
KM1 - Montée de la cabine	Mont_cab	%Q0.2.0	EBOOL
KM2 - Descente de la cabine	Desc_cab	%Q0.2.1	EBOOL
H1 - Voyant d'appel cabine étage 1	V_cab_1	%Q0.2.2	EBOOL
H2 - Voyant d'appel cabine étage 2	V_cab_2	%Q0.2.3	EBOOL
H3 - Voyant d'appel cabine étage 3	V_cab_3	%Q0.2.4	EBOOL
H4 - Voyant d'appel cabine étage 4	V_cab_4	%Q0.2.5	EBOOL
H5 - Voyant d'appel cabine étage 5	V_cab_5	%Q0.2.10	EBOOL
H6 - Voyant d'appel monté étage 1	Voy_m_1	%Q0.2.6	EBOOL
H7 - Voyant d'appel monté étage 2	Voy_m_2	%Q0.2.7	EBOOL
H8 - Voyant d'appel monté étage 3	Voy_m_3	%Q0.2.8	EBOOL
H9 - Voyant d'appel monté étage 4	Voy_m_4	%Q0.2.9	EBOOL
H10 - Voyant d'appel descente étage 5	Voy_d_5	%Q0.2.11	EBOOL
H11 - Voyant d'appel descente étage 4	Voy_d_4	%Q0.2.20	EBOOL
H12 - Voyant d'appel descente étage 3	Voy_d_3	%Q0.2.19	EBOOL
H13 - Voyant d'appel descente étage 2	Voy_d_2	%Q0.2.18	EBOOL

IV.7 Programmation

Voir les listes des programmes ci-après :

IV.7.1 Création d'un nouveau projet et déclaration de Rack automate



IV.7.2 Déclaration de la Configuration Matérielle de l'automate

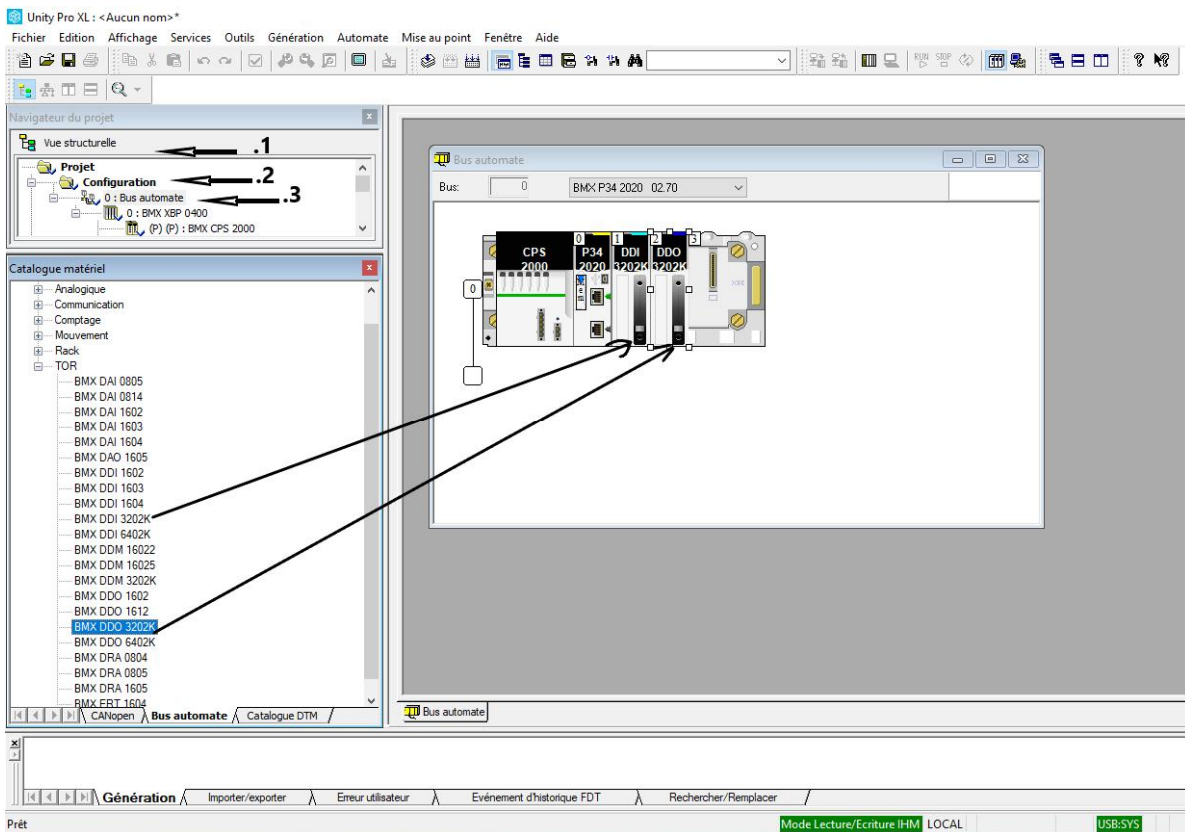
Sélectionner l'éditeur de configuration "Bus Automate", pour déclarer le Rack automate et les cartes associées, il se compose de :

Rack BMX XBP 0400 / Rack 4 positions.

- Processeur CPU : BMX P34 2020 avec ports Modbus Ethernet.
- Carte d'entrées TOR : BMX DDI 3202 K
- Carte de sorties TOR : BMX DDO 3202 K

Structure programme :

- Cliquer dans la fenêtre "Vue structurelle" sur "Configuration" puis "Bus automate" pour définir le Rack, l'alimentation, le processeur ainsi que les interfaces E/S TOR. (En fonction de la configuration matérielle de la platine de commande de la maquette "Ascenseur").



IV.7.3 Déclaration des capteurs et des pré-Actionneurs dans "l'Editeur de Variables Elémentaire"

Dans l'arbre "Vue structurelle", rentrer dans l'éditeur de "Variables Elémentaires" puis dans l'onglet "Variable" cocher la case "EDT" (Elementay Data Type). Vous pouvez alors sélectionner :

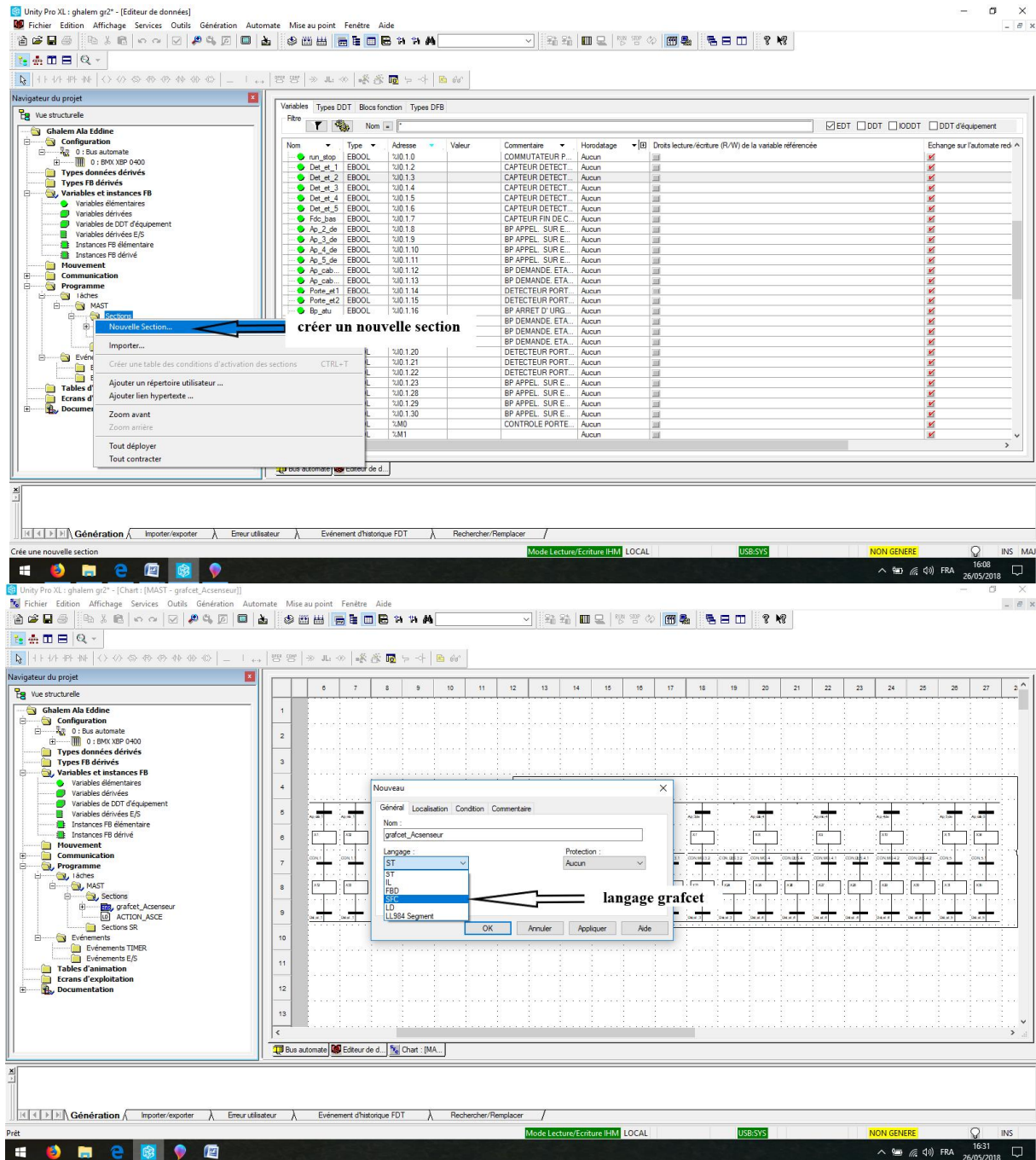
1. leur nom dans la colonne "Nom"
2. leur type dans la colonne "Type".
3. leur adresse dans la colonne "Adresse"
4. leur commentair(Désignation) dans la colonne "Commantair"

The screenshot displays the 'Unity Pro XL : ghaem gr2' - [Editeur de données]' interface. The left sidebar shows the project tree under 'Ghaem Ala Eddine', with 'Variables et instances FB' selected. The main window shows the 'Variables' editor with a table of variables. The table has columns: Nom, Type, Adresse, Valeur, Commentaire, Horodatage, and Echange sur l'automate red. The 'Echange sur l'automate red' column contains checkboxes, with the 'EDT' checkbox checked for the first row. Arrows labeled 1, 2, and 3 indicate the steps described in the text: 1 points to the project tree, 2 points to the 'Variables' tab, and 3 points to the 'EDT' checkbox.

Nom	Type	Adresse	Valeur	Commentaire	Horodatage	Echange sur l'automate red
run_stop	EBOOL	%I0.1.0		COMMUTATEUR P...	Aucun	<input checked="" type="checkbox"/>
Det_et_1	EBOOL	%I0.1.2		CAPTEUR DETECT...	Aucun	<input checked="" type="checkbox"/>
Det_et_2	EBOOL	%I0.1.3		CAPTEUR DETECT...	Aucun	<input checked="" type="checkbox"/>
Det_et_3	EBOOL	%I0.1.4		CAPTEUR DETECT...	Aucun	<input checked="" type="checkbox"/>
Det_et_4	EBOOL	%I0.1.5		CAPTEUR DETECT...	Aucun	<input checked="" type="checkbox"/>
Det_et_5	EBOOL	%I0.1.6		CAPTEUR DETECT...	Aucun	<input checked="" type="checkbox"/>
Fdc_bas	EBOOL	%I0.1.7		CAPTEUR FIN DE C...	Aucun	<input checked="" type="checkbox"/>
Ap_2_de	EBOOL	%I0.1.8		BP APPEL SUR E...	Aucun	<input checked="" type="checkbox"/>
Ap_3_de	EBOOL	%I0.1.9		BP APPEL SUR E...	Aucun	<input checked="" type="checkbox"/>
Ap_4_de	EBOOL	%I0.1.10		BP APPEL SUR E...	Aucun	<input checked="" type="checkbox"/>
Ap_5_de	EBOOL	%I0.1.11		BP APPEL SUR E...	Aucun	<input checked="" type="checkbox"/>
Ap_cab_1	EBOOL	%I0.1.12		BP DEMANDE ETA...	Aucun	<input checked="" type="checkbox"/>
Ap_cab_2	EBOOL	%I0.1.13		BP DEMANDE ETA...	Aucun	<input checked="" type="checkbox"/>
Porte_et1	EBOOL	%I0.1.14		DETECTEUR PORT...	Aucun	<input checked="" type="checkbox"/>
Porte_et2	EBOOL	%I0.1.15		DETECTEUR PORT...	Aucun	<input checked="" type="checkbox"/>
Bp_atu	EBOOL	%I0.1.16		BP ARRET D'URG...	Aucun	<input checked="" type="checkbox"/>
Ap_cab_3	EBOOL	%I0.1.17		BP DEMANDE ETA...	Aucun	<input checked="" type="checkbox"/>
Ap_cab_4	EBOOL	%I0.1.18		BP DEMANDE ETA...	Aucun	<input checked="" type="checkbox"/>
Ap_cab_5	EBOOL	%I0.1.19		BP DEMANDE ETA...	Aucun	<input checked="" type="checkbox"/>
Porte_et3	EBOOL	%I0.1.20		DETECTEUR PORT...	Aucun	<input checked="" type="checkbox"/>
Porte_et4	EBOOL	%I0.1.21		DETECTEUR PORT...	Aucun	<input checked="" type="checkbox"/>
Porte_et5	EBOOL	%I0.1.22		DETECTEUR PORT...	Aucun	<input checked="" type="checkbox"/>
Ap_mo_1	EBOOL	%I0.1.23		BP APPEL SUR E...	Aucun	<input checked="" type="checkbox"/>
Ap_mo_2	EBOOL	%I0.1.28		BP APPEL SUR E...	Aucun	<input checked="" type="checkbox"/>
Ap_mo_3	EBOOL	%I0.1.29		BP APPEL SUR E...	Aucun	<input checked="" type="checkbox"/>
Ap_mo_4	EBOOL	%I0.1.30		BP APPEL SUR E...	Aucun	<input checked="" type="checkbox"/>
Cond_init	EBOOL	%M0		CONTROLE PORTE...	Aucun	<input checked="" type="checkbox"/>
Mem_1	EBOOL	%M1			Aucun	<input checked="" type="checkbox"/>

IV.7.4Créationd'une nouvelle section

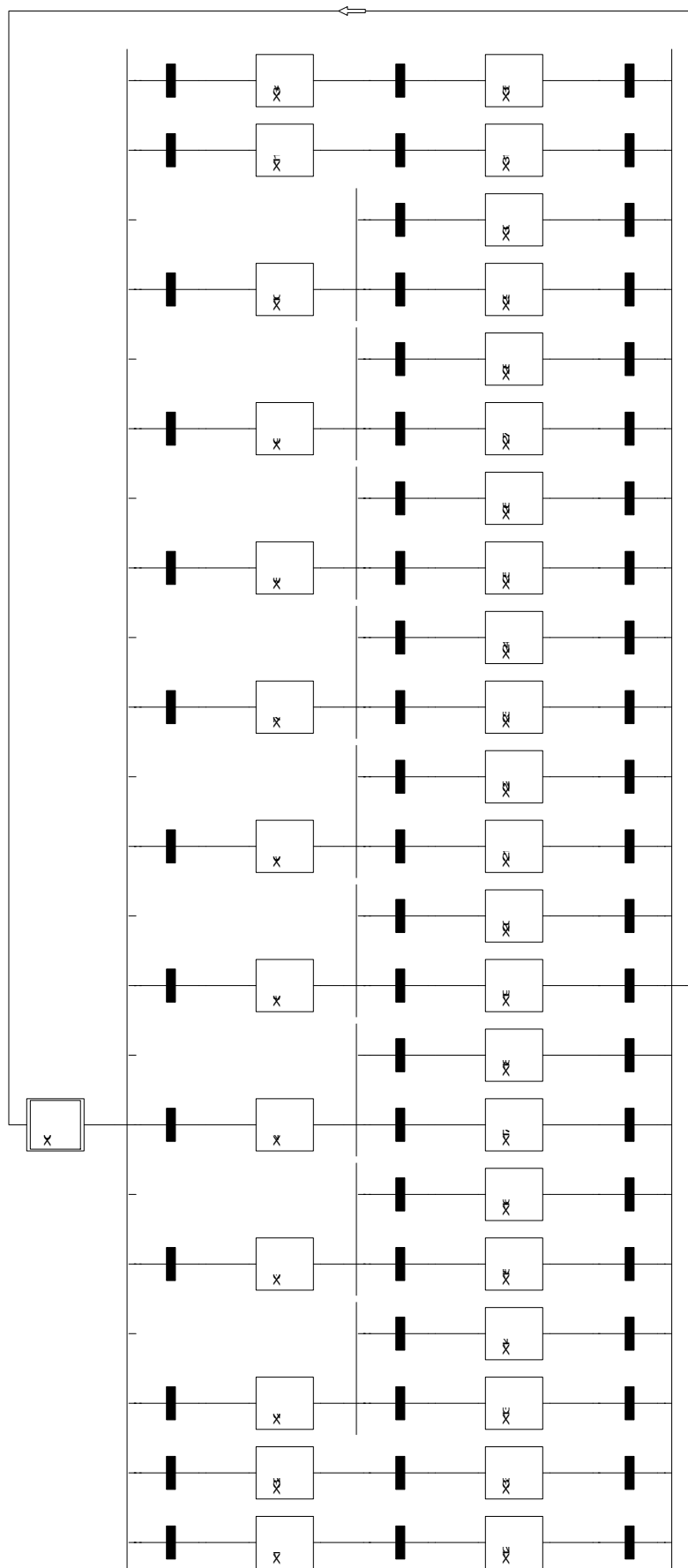
Dans l'éditeur de programme, créer un nouvelle section dans "Tâche" puis "Mast" que vous numerez : grafcet_Acenseur. La programmation se fera en grafcet



Remarque :

- Section de transition utilisée une seul fois
- Configuration des transitions en ladder

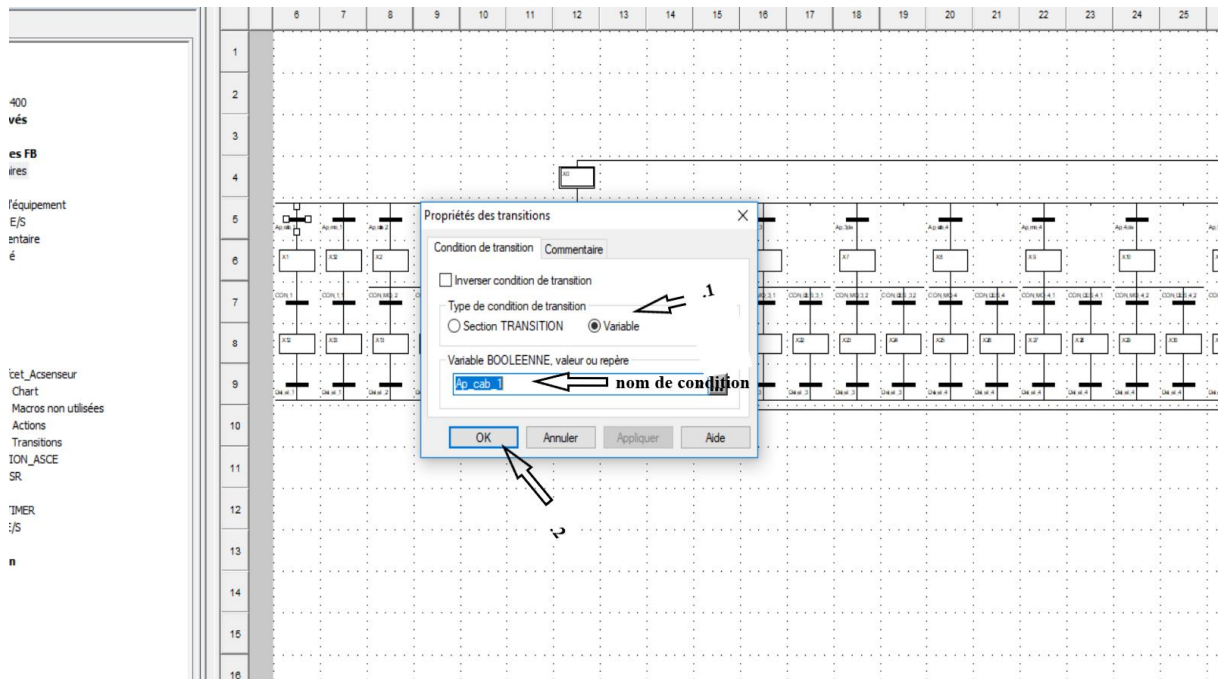
IV.7.5 Grafset de commande de l'ascenseur



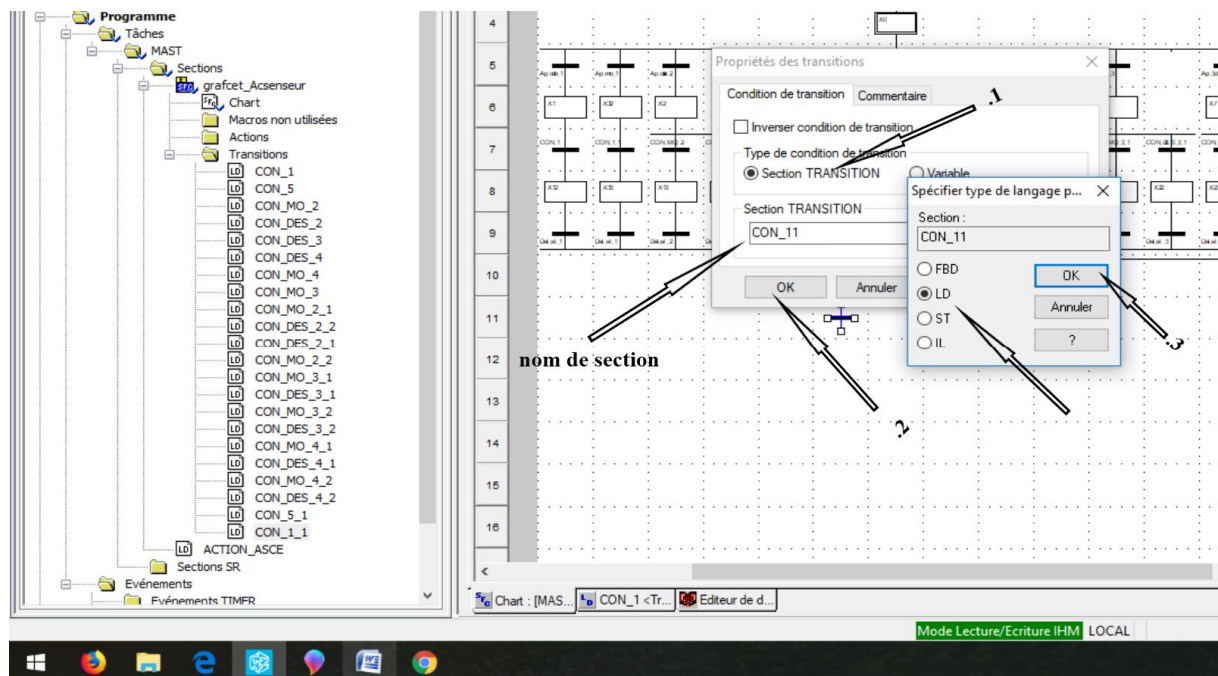
IV.7.5.1 Configuration des étapes et transitions

Pour configurer les transitions appuyez deux fois sur la transition :

1. Si la transition se compose d'une condition

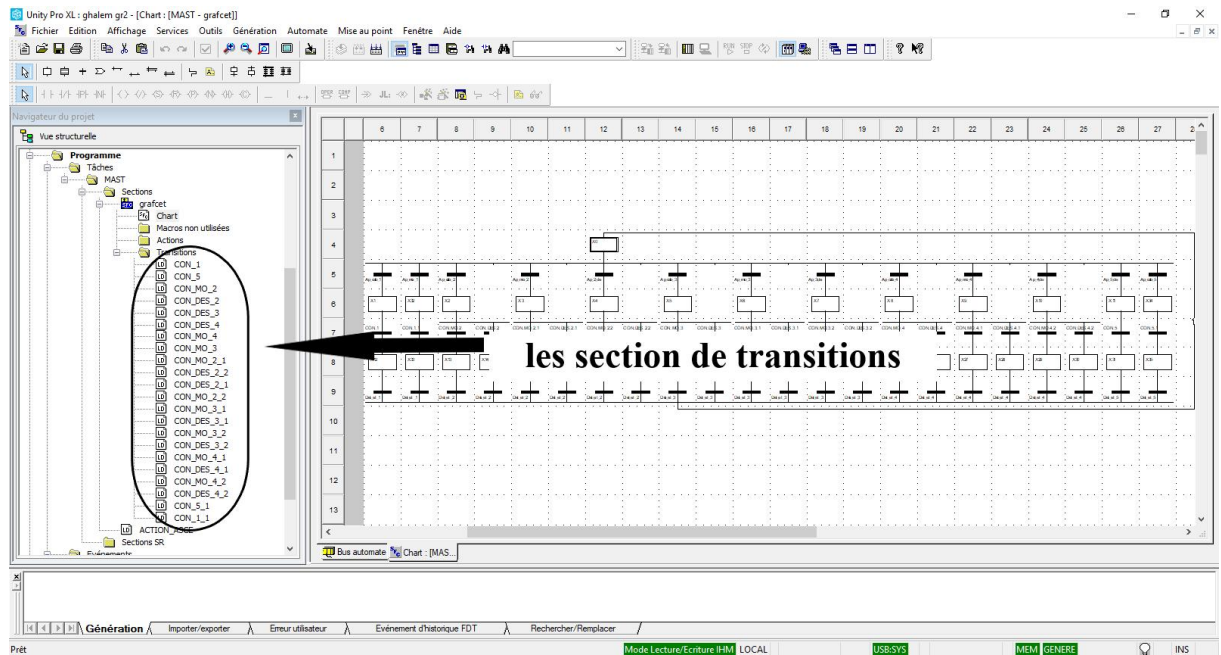


2. Si la transition se compose plus d'une condition



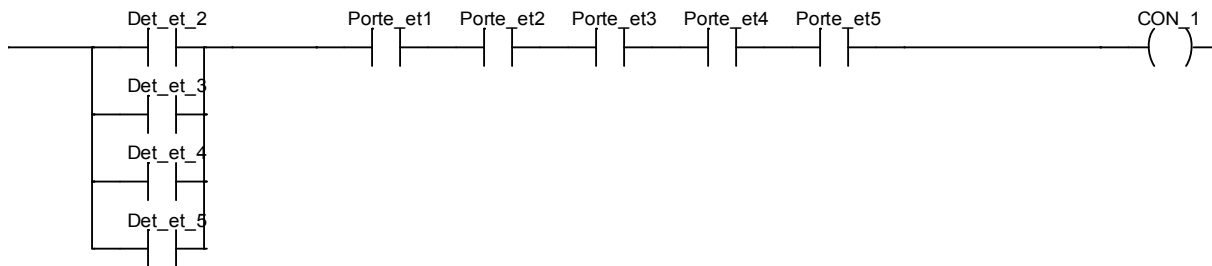
Remarque :

- section de transition utilisée une seul fois
- Configuration des transitions en ladder

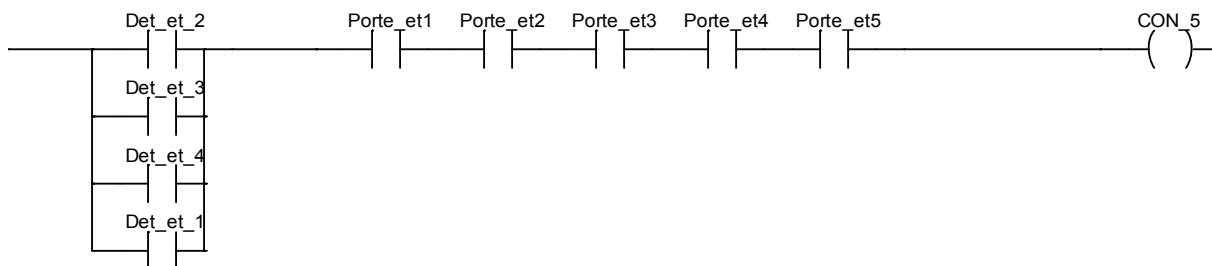


IV.7.5.2 Les transitions se compose plus d'une condition

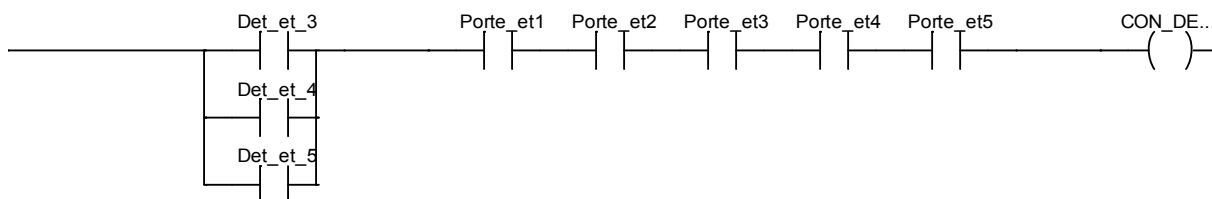
CON_1 et CON_1_1 :



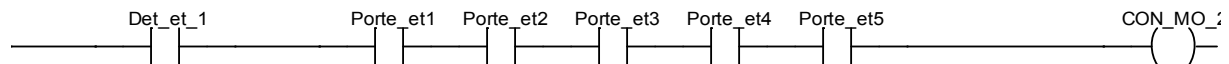
CON_5 et CON_5_1 :



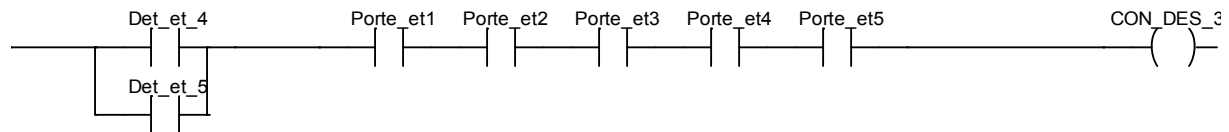
CON_DES_2 et CON_DES_2_1 et CON_DES_2_2



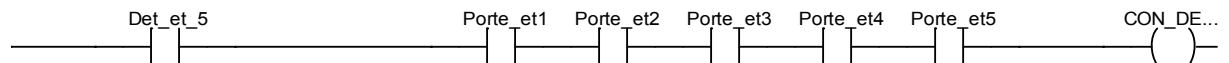
CON_MO_2 et CON_MO_2_1 et CON_MO_2_2



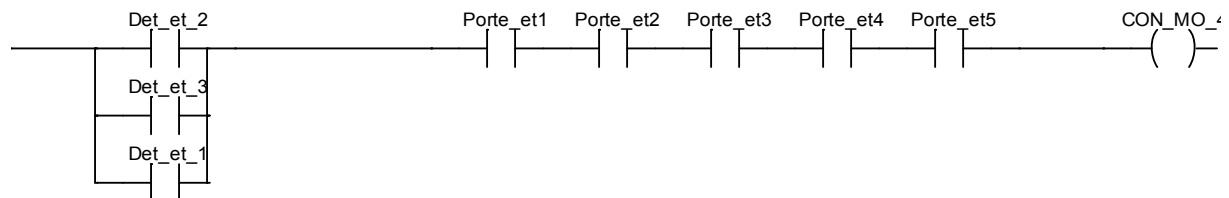
CON_DES_3 et CON_DES_3_1 et CON_DES_3_2



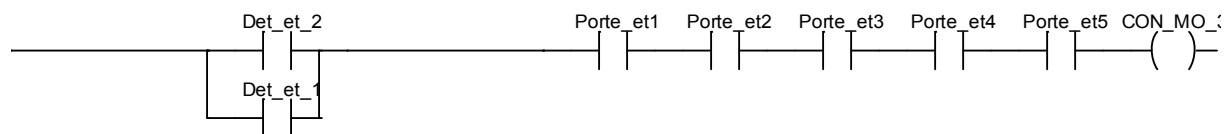
CON_DES_4 et CON_DES_4_1 et CON_DES_4_2



CON_MO_4 et CON_MO_4_1 et CON_MO_4_2



CON_MO_3 et CON_MO_3_1 et CON_MO_3_2



IV.7.5.3 Configuration des étapes

Le traitement séquentiel permet la saisie du graphe et des transitions associées.

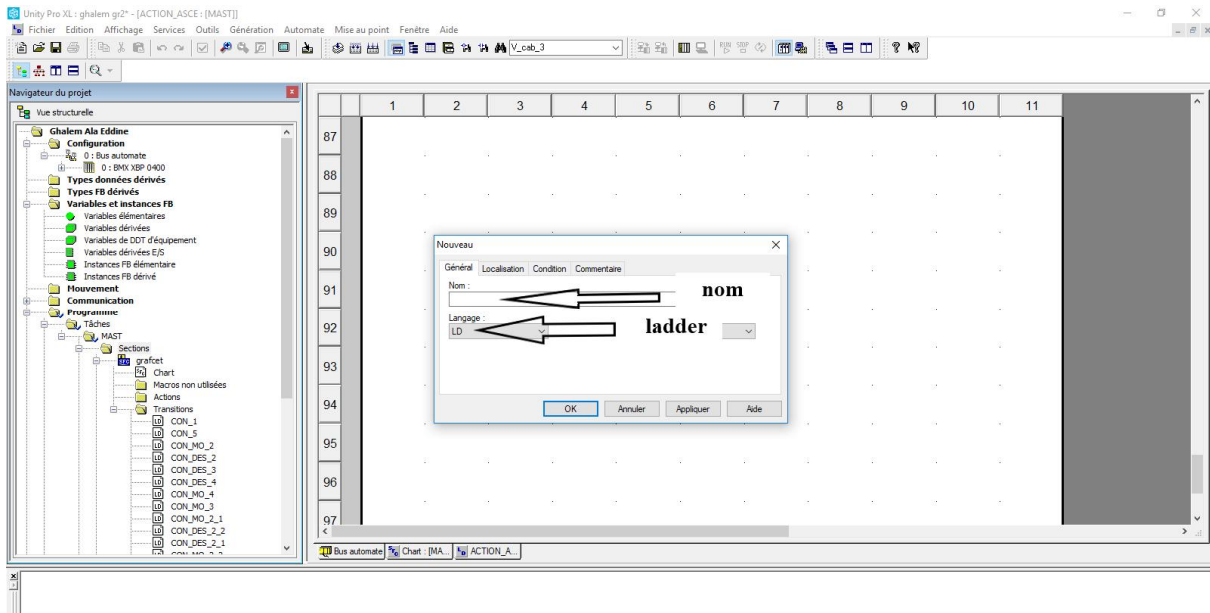
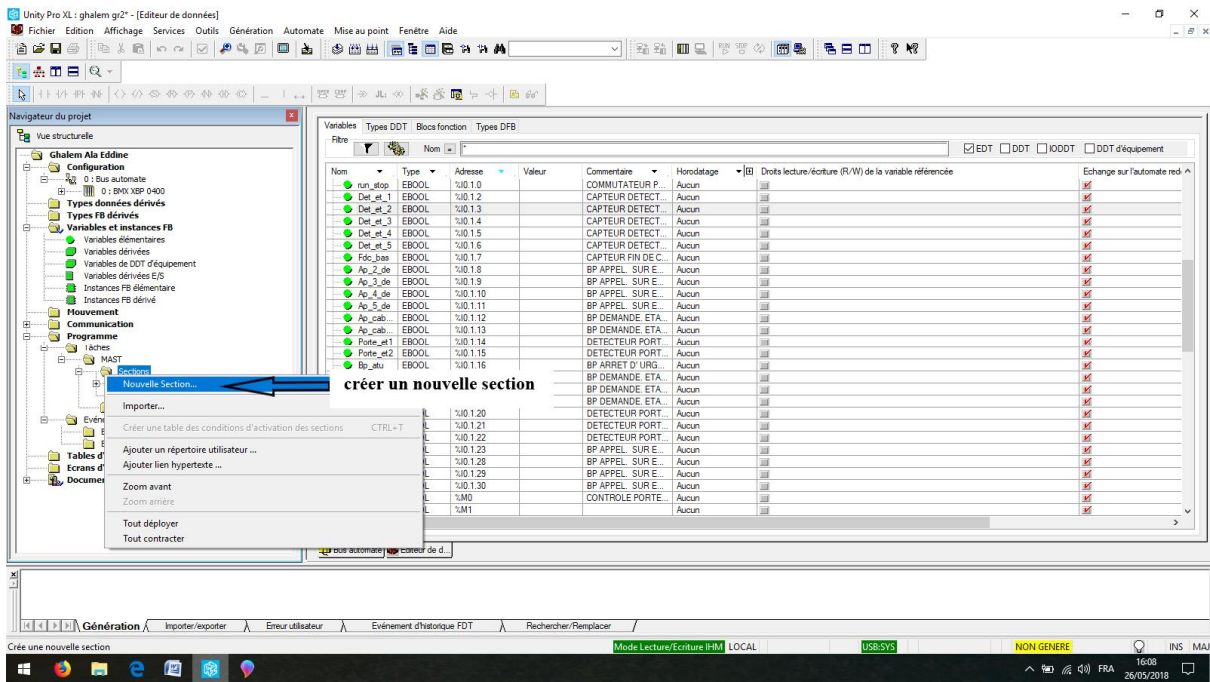
Le traitement postérieur permet d'écrire les actions associées aux étapes en utilisant les bits d'étape %Xi.x.

Remarque :

- On peut utiliser plusieurs étapes pour exécuter une action précisée
- Configuration des étapes en ladder

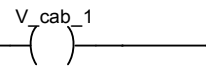
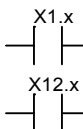
Créer une nouvelle section :

Dans l'éditeur de programme, créer une nouvelle section dans "Tâche" puis "Mast" que vous nommerez : ACTION_ASCE. La programmation se fera en ladder

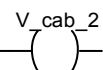
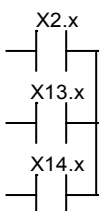


Dans cette section tapez le suivant :

V_cab_1



V_cab_2



V_cab_3



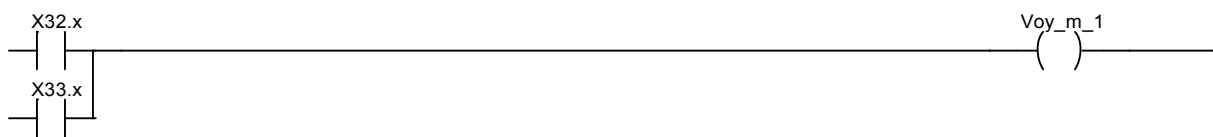
V_cab_4



V_cab_5



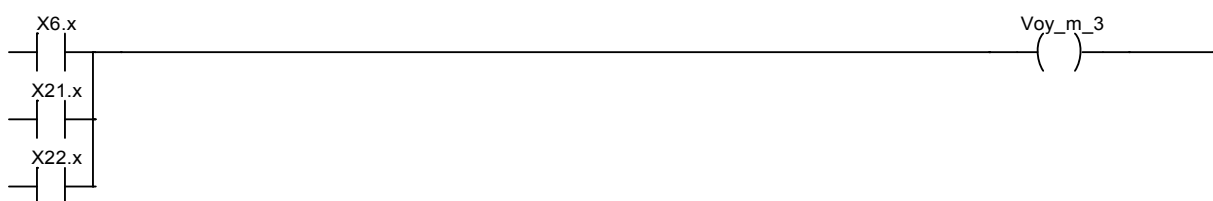
Voy_m_1



Voy_m_2



Voy_m_3



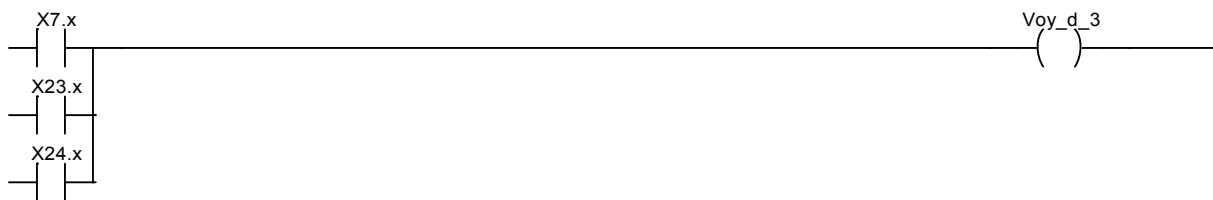
Voy_m_4



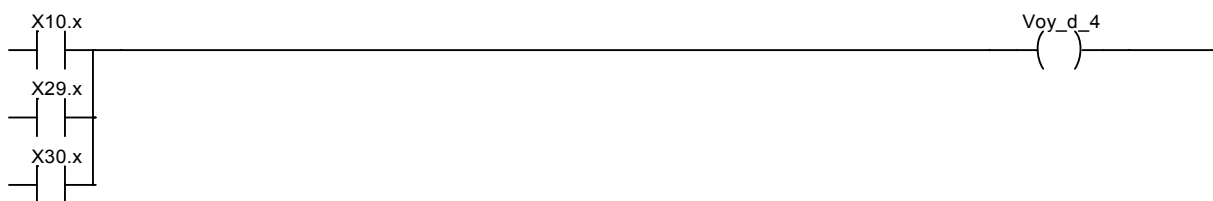
Voy_d_2



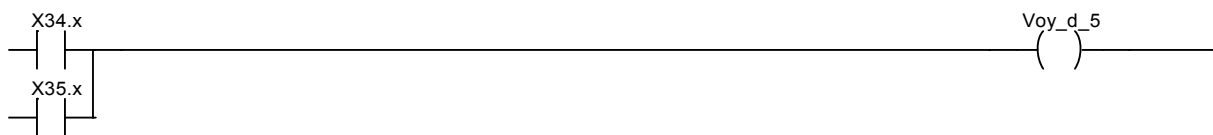
Voy_d_3



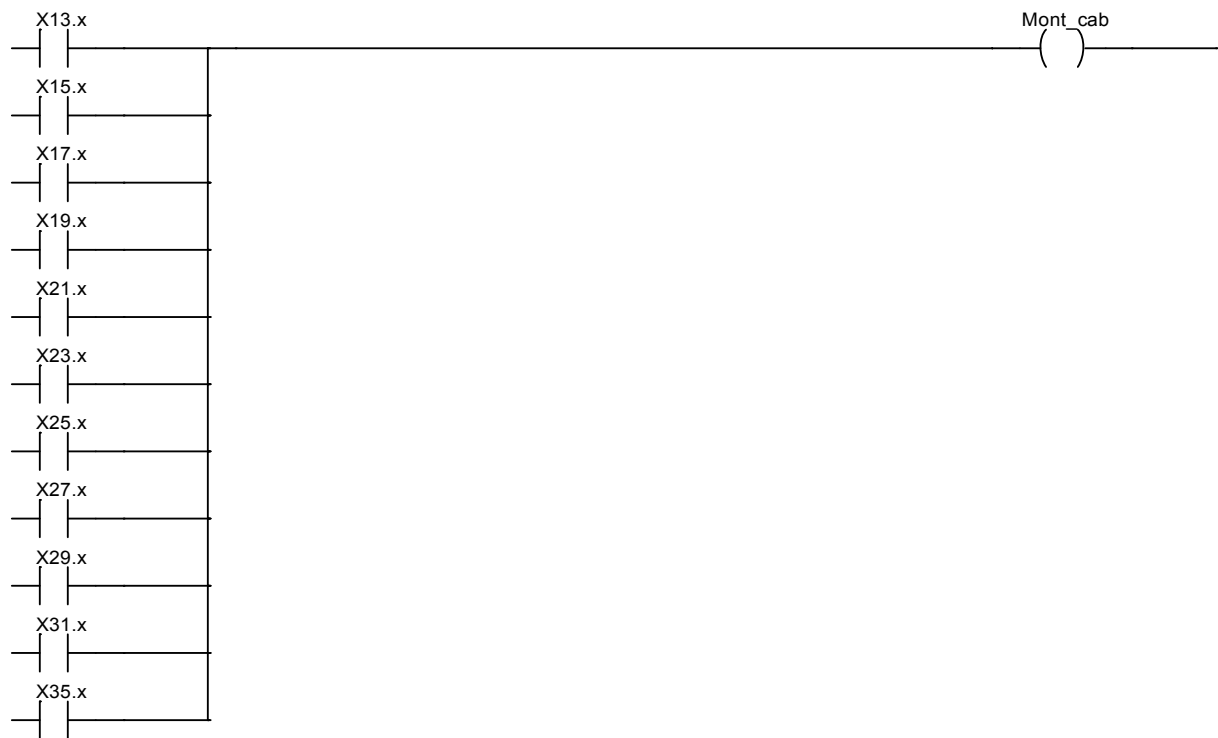
Voy_d_4



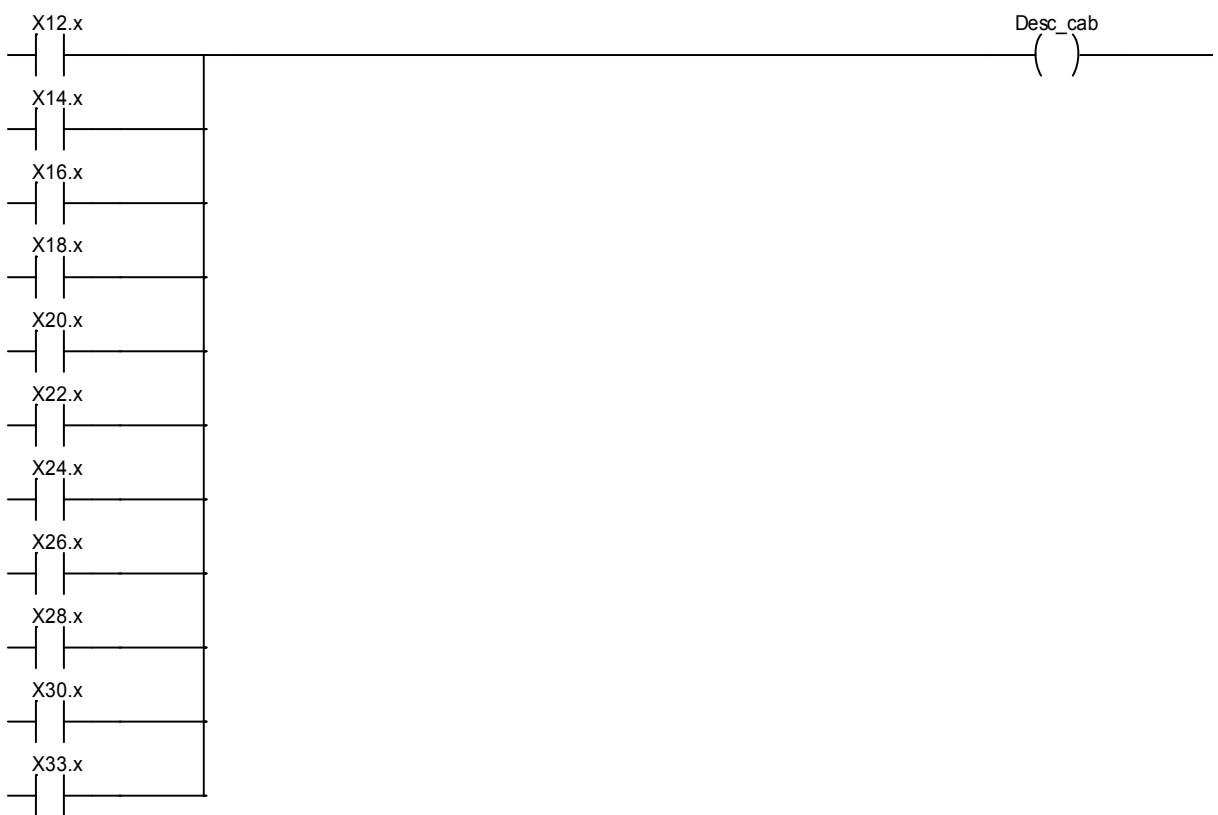
Voy_d_4



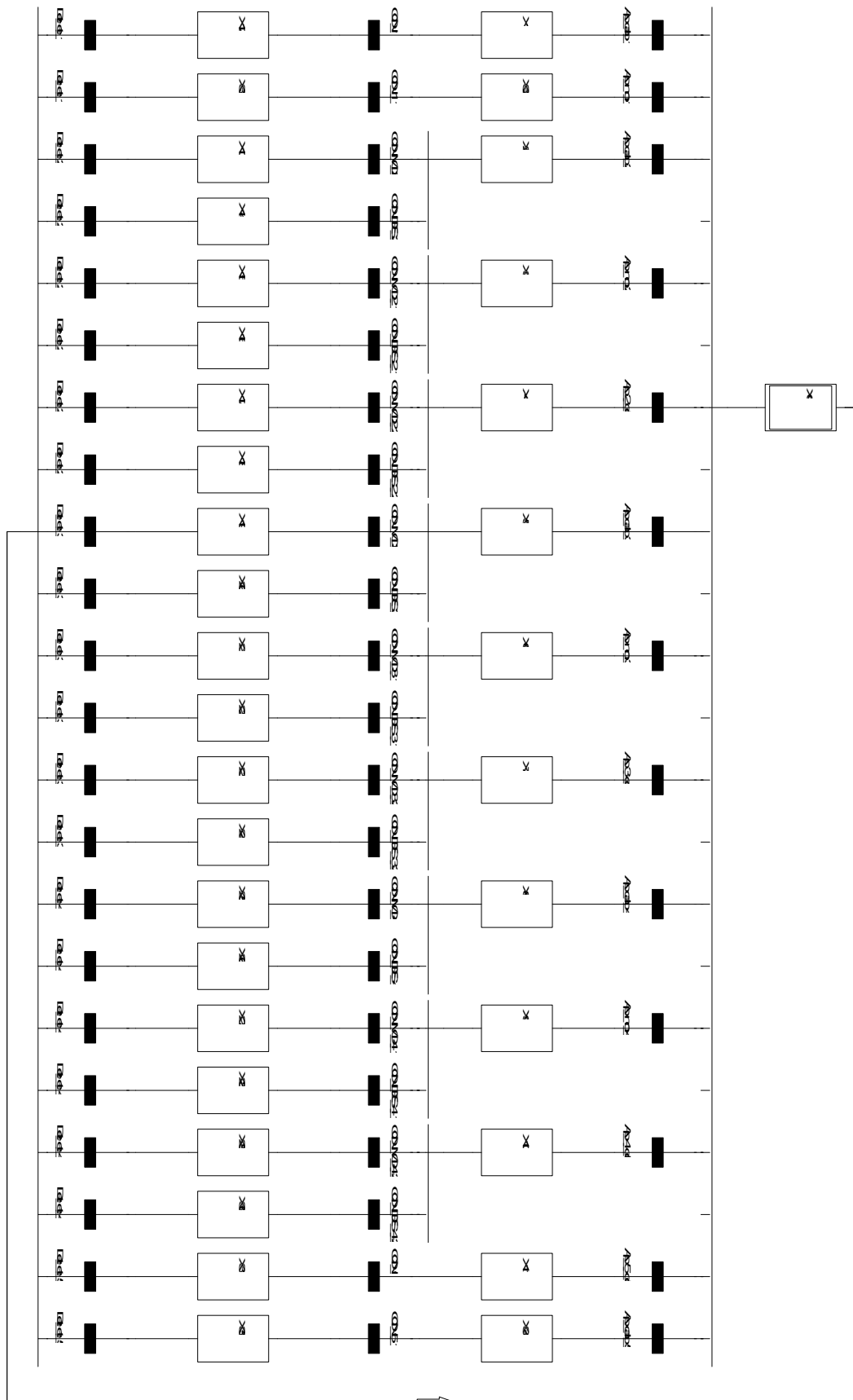
Mont_cab



Desc_cab



IV.7.5.4 Grafset final de l'ascenseur

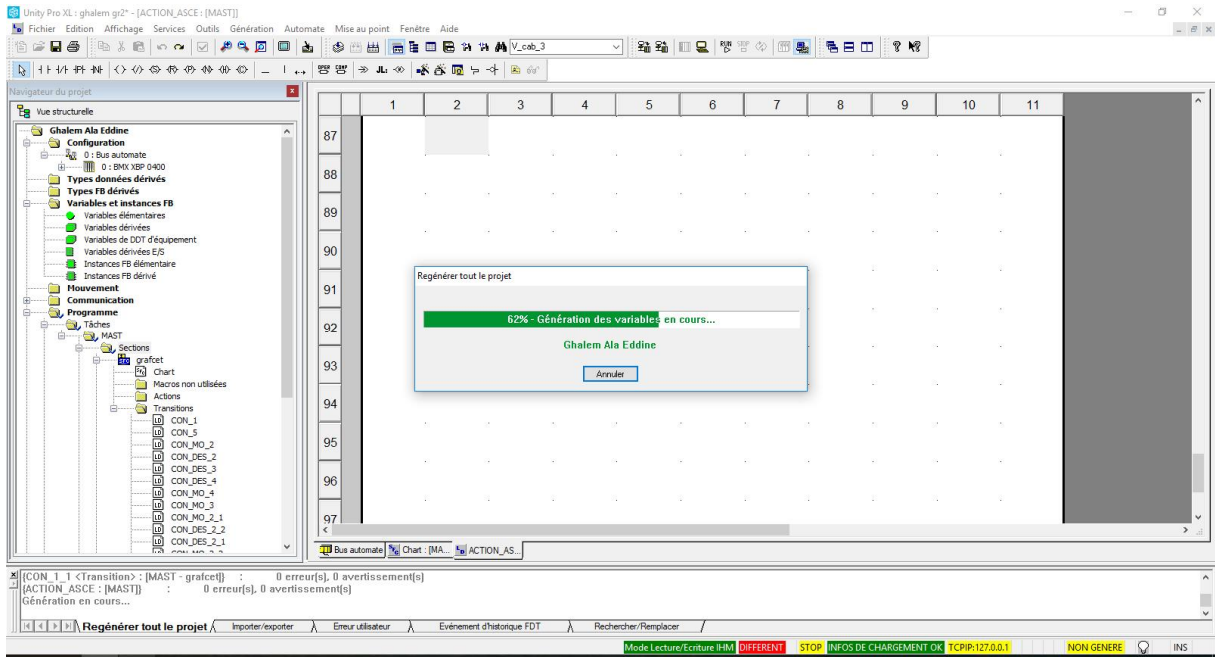
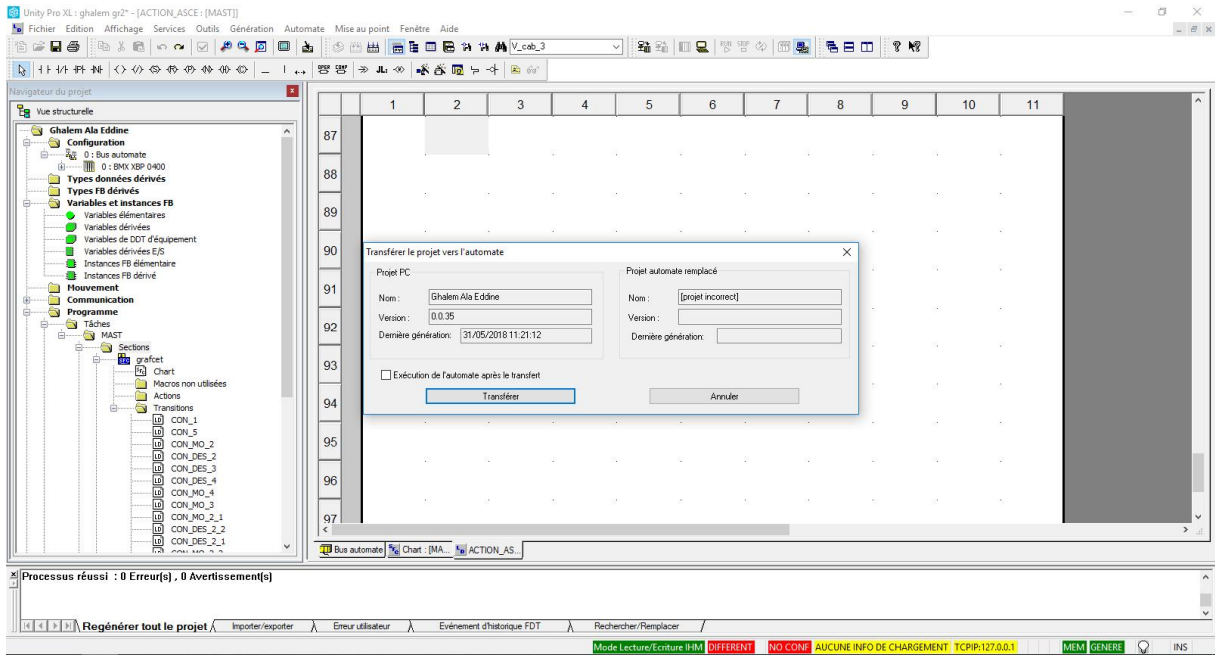


Etap	Action
X1	V_cab_1
X2	V_cab_2
X3	Voy_m_2
X4	Voy_d_2
X5	V_cab_3
X6	Voy_m_3
X7	Voy_d_3
X8	V_cab_4
X9	Voy_m_4
X10	Voy_d_4
X11	V_cab_5
X12	V_cab_1 \Desc_cab
X13	V_cab_2 \Mont_cab
X14	V_cab_2 \Desc_cab
X15	Voy_m_2 \Mont_cab
X16	Voy_m_2 \Desc_cab
X17	Voy_d_2 \Mont_cab
X18	Voy_d_2 \Desc_cab
X19	V_cab_3 \Mont_cab
X20	V_cab_3 \Desc_cab
X21	Voy_m_3 \Mont_cab
X22	Voy_m_3 \Desc_cab
X23	Voy_d_3 \Mont_cab
X24	Voy_d_3 \Desc_cab
X25	V_cab_4 \Mont_cab
X26	V_cab_4 \Desc_cab
X27	Voy_m_4 \Mont_cab
X28	Voy_m_4 \Desc_cab
X29	Voy_d_4 \Mont_cab
X30	Voy_d_4 \Desc_cab
X31	V_cab_5 \Mont_cab
X32	Voy_m_1
X33	Voy_m_1 \Desc_cab
X34	Voy_d_5
X35	Voy_d_5 \Mont_cab

Généation puis transfert le programme vers l'automate

- Sauvegarder votre application.
- Générer votre projet, puis "Connecter" et "Transférer" vers AP.
- Tester le fonctionnement sur l'appareil

The screenshot displays the Unity Pro XL software interface for a project named 'Ghalem Ala Eddine'. The main window shows a ladder logic diagram with rungs numbered 87 to 95. The left sidebar contains a project navigator with a tree view showing folders like 'Configuration', 'Types données dérivés', 'Variables et instances FB', 'Mouvement', 'Communication', and 'Programme'. The top menu bar includes 'Fichier', 'Edition', 'Affichage', 'Services', 'Outils', 'Génération', 'Automate', 'Mise au point', 'Fenêtre', and 'Aide'. The 'Génération' menu is open, showing options like 'Analyser le projet' and 'Regénérer tout le projet'. The 'Connexion' option is highlighted, opening a sub-menu with options such as 'Définir l'adresse...', 'Mode Standard', 'Mode Simulation', 'Comparer...', and 'Transférer le projet vers l'automate'. Three arrows labeled .1, .2, and .3 point to 'Mode Standard', 'Connexion', and 'Transférer le projet vers l'automate' respectively.



*Conclusion
générale*

Conclusion générale

L'évolution de la technologie et du mode de vie nous permet aujourd'hui de prévoir des espaces de travail et de logement mieux adaptés pour assurer le confort et la sécurité de individus et des travailleurs .Dans ce projet on a essayé de programmer un ascenseur en utilisant un automate programmable de type Schneider et référence M340. L'API Schneider M340 est du nouvel arrivage à l'université de Mohamed Kheider Biskra. Le choix de cet exemple d'application est pour approfondir nos connaissances théoriques et pratiques dans le domaine des automatismes industriels et la pédagogie.

L'objectif de notre modeste travail est faire programmation didactique pour l'ascenseur .Pour la programmation de l'automate programmable industriel M340, on a utilisé le logiciel UNITY PRO

En perspectives, nous pouvons dire que ce travail n'est qu'une simple application dans le domaine de l'industrie, il peut être plus autonome, plus pratique, Parmi les perspectives ouvertes à ce projet :

- sécuriser l'utilisation par un code.
- Faire une amélioration sur notre application
- Mettre en œuvre les connaissances acquises sur l'ascenseur.

Bibliographie

Bibliographie

- [1] “ knitting technologie and praticalguide” edition spencer 1989
- [2] bianco spa –vidaleindustria 4-12051 alba (CN) – italiawww.bianco-spa.com
- [3] manuel d’utilisation du compacteur tubolairewww.bianco-spa.com
- [4] M.KARIM , A.YASSIN << Etude d’adaptation d’un automate S7-300 sur une aléuseuse GSP Ebauchée implémentation d’une interface homme/machine >>PFE faculté des science de l’ingénieure 2015
- [5]Henri Nussbaumer, « informatique industrielle III »,Press polytechniques de Romandes 1987
- [6] MICHEL Groult et Patrick Salam, Instrumentation industriel : 3eme édition
- [7] C.VRIGNON et M.THENAISSIE, ISTI
- [8] G .Michel, « Architecture et application des automates programmable industriels »DUNOD ,1987
- [9] Jean-Claude Humbolt, « Automate programmables Industriels » édition DUNOD.1993.
- [10] William Bolton, « Les automates programmable industriels » [http://www.course .free.fr](http://www.course.free.fr)
- [11] Philippe LE BRUN « les automates programmable industriel », lycée louis ARMAND, 1999
- [12] Gilles MICHEL, Claude LAURGEAU, Bernard ESPIAU « les automates programmables industrielle », édition DUNOD.1979.

Bibliographie

- [13] « système automatisées, bus de terrain, API SIEMENS »ELWE, système didactiques pour l'enseignement et la formation en science et technique Industriel, mai 2001
- [14] André SIMON, « Automate programmables programmables » édition L'ELAN, 1983
- [15] Henri Nussbaumer « Informatique Industriel » édition 1986 grafcet
- [16] Jean-Bernard DELUCHE « pour un bon usage du GRAFCET » Lycée Raoul DAUTRY LIMOGES/Faculté des sciences & Technique LIMOGES
- [17] La S.N.V.I << présentation de l'entreprise >>. [2] : Document technique S.N.V.I / Rouïba.
- [18] H.GEORGE : 'Capteurs en instrumentation', Edition DUNON, Paris,1999.
- [19] www.techno-science.net
- [20] www.ceci.fr
- [21] www.mrw.wallonie.be
- [22] www.syndic-benevole.fr
- [23] www.inter-coproprietes.com
- [24] www.schneider-electric.com
- [25] KAMKUMO, Cédric TAYO. Conception de l'installation électrique et du système de contrôle-commande de la station de relevage sur l'étage de Korofina. Bamako, Mali : s.n., 2015.
- [26] Electric, Schneider. Catalogue 2012. 2012.

Bibliographie

[27] BONNET, Pierre. Introduction à la supervision. Lille : Université Lille 1, Sciences et technologies, 2010.

[28] Electric, Schneider. Quickstart tutorial V7.20 CitectSCADA. Australia : s.n., April 2011.

[29] Guide de programmation du variateur de vitesse pour le moteur asynchrone www.schneider-electric.com.

[30] Catalogue Modicon M340 sous Unity Pro Processeurs, racks et modules d'alimentation Manuel de configuration 05/2010 Electric, Schneider

[31] Catalogue Janvier 2016 Logiciels Unity et OPC Electric, Schneider