

République Algérienne Démocratique et Populaire
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE



Ecole Nationale Polytechnique
Département d'Automatique

**Mémoire De Master
EN AUTOMATIQUE**

Thème

**Gestion D'UNE MAQUETTE DE STATION DE
POMPAGE PAR UN AUTOMATE
PROGRAMMABLE ZELIO**

Proposé et dirigé par :

Pr BERKOUK El Madjid

Etudié par :

M^{lle} FERHAOUI Bouchra

Promotion : 2015

Ecole Nationale Polytechnique
10, Avenue Hassen Badi, BP. 182, 16200 El Harrach, Alger, Algérie

Remerciement

Avant tout, je remercie Dieu, le tout puissant, pour m'avoir assistée et donnée patience et volonté afin d'accomplir ce modeste travail.

L'aboutissement à la réalisation de ce travail est le fruit de toutes les années de formation, je tiens donc à exprimer tout d'abord mon respect et ma gratitude à tous mes enseignants

J'exprime mes remerciements à notre prometteur Monsieur E.M BERKOUK pour l'assistance qu'il m'a témoignée tout au long de ce travail, qu'il trouve ici l'expression de ma gratitude pour ses conseils.

Je tiens à remercier également les membres de jury pour avoir accepté de jurer ce mémoire.

J'adresse mes vifs remerciements, à ma famille, mes amies et tous ceux qui m'ont encouragé chacun de son nom.

Bouchra

ملخص

العملا لمقدم في هذا لأطروحة يتمحور حول استخدام مسير صناعي مبرمج "تيليميكانيك" التابع لشركة "شنايدر اليكتريك". لقد قمنا بتسيير نموذج محطة ضخ المياه بواسطة مسير صناعي مبرمج "زيلييو". لقد بدأنا بعرض كلال مسيرات الصناعية "تيليميكانيك", ثم عرضنا البرنامج "زيلييو سوفت", في الأخير قمنا بوضع دفاتر الشروط والتي قمنا بدراستها وبرمجتها و من ثم نقل البرنامج إلى المسير الصناعي ليتم تطبيقه على النموذج. اختتمنا المذكرة باستنتاجات عامة حول العمل المنجز.

الكلمة المفتاح:

نموذج محطة ضخ المياه, مسير صناعي مبرمج "تيليميكانيك", زيلييو سوفت

Résumé

Le travail présenté dans ce mémoire est basé essentiellement sur l'utilisation des automates programmables Télémécanique de SCHNEIDER ELECTRIC. Nous avons assuré la gestion d'un prototype de station de pompage hydraulique par un Automate Zelio.

Nous avons commencé par la présentation de tous les Automates Télémécanique, ensuite on a présenté logiciel Zelio Soft, en fin on a élaboré des cahiers des charges qu'on a analysé, programmé et transféré vers l'automate pour les appliquer sur le prototype. Nous avons clôturé le mémoire par une conclusion générale sur le travail effectué.

Mot clé :

Prototype de station de pompage, Automate programmable Télémécanique, Zelio Soft.

Abstract

The work presented in this memory is based primarily on the use of programmable automats SCHNEIDER ELECTRIC. We assured the management of a hydraulic pump station prototype with a Zelio PLC.

We started with the presentation of all Telemecanique PLCs, then we presented Zelio Soft software, we developed some sets of specifications that we analyzed, and transferred to the PLC in order to apply them to the prototype. We closed the memory by a general conclusion on the work done.

Key Word :

Pumping station prototype, PLC Telemecanique, Zelio Soft

SOMMAIRE

Introduction générale	1
CHAPITRE I : Les différentes gammes d'automates Télémécanique	
I.1 Introduction	3
I.2 Modules programmables Zelio Logic	3
I.2.1 Présentation	3
I.2.2 Programmation	
I.2.3 Modules logiques compacts	3
I.2.4 Modules logiques modulaires	4
I.2.5 Communication	4
I.3 Contrôleurs programmables Twido	5
I.3.1 Présentation	5
I.3.2 Programmation	7
I.4 Modicon TSX Micro	7
I.4.1 Présentation	7
I.4.2 TSX 37 05 et TSX 37 08	7
I.4.3 TSX 37 10	8
I.4.4 TSX 37 21/22	8
I.4.5 Programmation	9
I.4.6 Communication	9
I.5 Automates Modicon M340	9
I.5.1 Présentation	9
I.5.2 Performances	10
I.5.3 Solutions intégrées	10
I.5.4 Programmation	11
I.6 Modicon Premium	11
I.6.1 Processeur Unity	11
I.6.2 Processeur PL7	13
I.7 Modicon Quantum	14
I.7.1 Processeur standard Unity Pro	14
I.7.2 Processeurs standard Concept / ProWORX	15
I.8 Conclusion	16

Chapitre II : Le logiciel de programmation Zelio Soft

II.1	Introduction	18
II.2	Présentation	18
II.3	Mode de fonctionnement	18
II.3.1	Mode saisie	18
II.3.2	Mode mise au point	18
II.4	La programmation en langage LADDER	19
II.4.1	Modes de saisie des schémas de commande	19
II.4.2	Fonctionnalités	20
II.4.3	Description d'un réseau de contacts	21
II.5	La programmation en langage FBD	22
II.5.1	Présentation	22
II.5.2	Fonctions en langage FBD	22
II.5.3	Blocs Fonctions FBD (Function Bloc Diagram)	23
II.5.4	Fonctions logiques	25
II.5.5	Grafcet / SFC (Sequential Function Chart)	26
II.6	Transfert d'une application	27
II.6.1	Ecrire d'un PC vers le Zelio Logic	27
II.6.2	Transfert du programme Zelio Logic vers le PC	28
II.6.3	Fonction mot de passe	29
II.7	Verrouillage façade	29
II.8	Exemples d'application	29
II.8.1	Exemple d'application en langage LADDER	29
II.8.2	Exemple d'application en langage FBD	34
II.9	Conclusion	38

ChapIII : Réalisation et Gestion du prototype de la station de pompage par l'automate Zelio

III.1	Introduction	40
III.2	Description du Prototype de la station de pompage	40
III.3	Eléments de commande du Prototype de la station de pompage	42
III.4	Développement du projet pour gestion du prototype de la station de pompage	42
III.5	Définition du Cahier de charge	43
III.6.	Création du programme de gestion de la station de pompage	44
III.7.	Programmation sur écran	45

III.8.Conclusion	46
Conclusion générale et perspectives	47
Bibliographie	48
ANNEXE A : Les programmes des 3 manipulations	49
ANNEXE B : Edition du programme sur écran de l'automate Zelio	53

Liste Des Figures

Figure I.1 : *Module Zelio Logic compact*

Figure I.2 : *Association entre modules logiques modulaires et extensions*

Figure I.3 : *Contrôleurs programmables compacts Twido*

Figure I.4 : *Contrôleurs programmables modulaires Twido*

Figure I.5: *Automates TSX 37 05 et TSX 37 08*

Figure I.6 : *Automates TSX 37 10 et TSX 37 21 et TSX 37 22*

Figure I.7 : *plate-forme Modicon M340*

Figure I.8 : *Différents processeurs TSX 57*

Figure I.9 : *Automate Modicon Quantum*

Figure II.1: *Différents outils de programmation en langage LADDER*

Figure II.2 : *Structure d'un réseau de contacts*

Figure III.1: *Prototype de la station de pompage*

Figure III.2: *les deux grafjets principales à réaliser sur la maquette*

Figure III.3: *Un grafjet supplémentaire à réaliser sur la maquette*

Figure III.3: *programme sur l'afficheur de l'automate Zelio*

Figure III.3: *Règles de saisie du schéma de commande*

Liste Des Tableaux

Tableau I.1: *Caractéristiques des différentes bases compactes Twido*

Tableau I.2 : *Caractéristiques des différentes bases modulaires Twido*

Tableau II.1: *Fonctions en symboles LADDER et symboles électriques*

Tableau II.2 : *Fonctions logiques*

Tableau II.3: *Fonctions FC*

Notations

AC	Alternative Curent
A.P.I	Automate Programmable Industriel
AS-i	Actuators Sensors Interface
CPU	Central Processing Unit
DC	Direct current (courant continu)
DI	Digital Input
DO	Digital Output
DP	Decentralized Peripher
EEPROM	Erasable Programmable Read Only Memory
EPROM	Erasable Programmable Read Only Memory
E/S	Entrées/Sorties
FBD	Function Bloc Diagram
IL	Instructed List
I/O	Input/Output
IP	Internet Protocol
LCD	Liquid-Cristal Display
LD	LADDER Diagram
Modbus	Modicon bus
MPI	Multi-Point Interface
PC	Personnel Computer
PCMCIA	Personal Computer Memory Card International Association
PEHD	Polyéthylène Haute Densité
PLC	Programmable Logic Controller
PROFIBUS	Process Field Bus
RAM	Random Access Memory
RS	Recommanded Standard
RTU	Remote Terminal Unit
SFC	Sequential function chart
ST	Structured Text

TCP Transmission Control Protocol
TOR Tout Ou Rien
USB Universal Serial Bus

Introduction Générale

A nos jours, l'automate programmable industriel A.P.I est devenu le constituant le plus répandu pour réaliser des automatismes. On le trouve pratiquement dans tous les secteurs de l'industrie, et cela est dû à la puissance de son environnement de développement et aux larges possibilités d'interconnexions, c'est la solution la plus efficace pour répondre à des besoins d'adaptation et de flexibilité pour un grand nombre d'opération.

La force principale d'un automate programmable industriel API réside dans sa grande capacité de communication avec l'environnement industriel. Il est constitué principalement d'une unité centrale, un module d'alimentation, des modules d'entrées/sorties, et des modules spécialisés, qui lui servent d'interface de communication avec le processus industriel de conduite.

Il a comme rôles principaux dans un processus :

- D'acquérir les informations fournies par les capteurs ;
- En faire le traitement ;
- Elaborer la commande des actionneurs ;
- Assurer la communication avec l'environnement pour l'échange d'informations.

De nombreux modèles d'automates sont aujourd'hui disponible, depuis les nano automates qui sont adaptés aux machines et installations simple avec un petit nombre d'entrées/sorties, jusqu'aux automates multifonctions qui sont utilisés pour la gestion des processus complexes avec des milliers d'entrées sorties.

Dans ce travail on propose de gérer le prototype de station de pompage réalisé au sein du laboratoire de l'automatique avec un automate programmable Zelio de la filiale Télémécanique offerte par la société Schneider.

Nous avons réparti notre travail en trois chapitres, le premier est consacré à la présentation des différentes gammes d'automates programmable de la filiale Télémécanique du Schneider. Au deuxième chapitre on se basera sur la description du logiciel de programmation Zelio Soft 2, avec les langages de programmations qu'il admet et les différentes fonctions qu'il offre.

Au troisième chapitre, on va proposer deux cahiers de charge à réaliser sur la maquette de station de pompage, nous établirons les grafkets et on les programmera à l'aide du logiciel Zelio Soft 2 , en choisissant comme automate le module programmable Zelio.

Chapitre I :
Les différentes gammes d'automates
Télémécanique

I.1.Introduction

Ce chapitre présente les différents automates programmables offerts par la filiale Télémécanique du Schneider Electric, leurs logiciels de programmations, ainsi que d'autres produits d'automatisations qui peuvent être associés à ces automates

I.2. Modules programmables Zelio Logic

I.2.1.Présentation

Les modules Zelio Logic sont destinés à la réalisation de petits équipements d'automatisme. Ils sont utilisés dans les secteurs d'activité de l'industrie et du tertiaire.

I.2.1.1.Pour l'industrie :

On trouve les modules Zelio Logic dans : des automatismes de petites machines de finition, de confection, d'assemblage ou d'emballage, petits équipements d'automatisme fonctionnant à a 48 V (application de levage,...), automatismes décentralisés sur les annexes de grosses et moyennes machines (domaines du textile, du plastique, de la transformation de matériaux,...), automatismes pour machines agricoles (irrigation, pompage, serre,...).

I.2.1.2.Pour le tertiaire/bâtiment :

Pour ce domaine, les modules Zelio Logic sont utilisés pour : des automatismes de barrières, de volets roulants, de contrôle d'accès, automatismes d'éclairage, automatismes de compresseurs et de climatisation, ...

Leur compacité et leur facilité de mise en œuvre en font une alternative compétitive aux solutions à base de logique câblée ou de cartes spécifiques.

I.2.2.Programmation

La simplicité de leur programmation, garantie par l'universalité des langages, satisfait aux exigences de l'automaticien et répond aux attentes de l'électricien.

La programmation peut être effectuée :

- de façon autonome en utilisant le clavier du module Zelio Logic (langage à contacts),
- sur PC avec le logiciel "Zelio Soft 2".

Sur PC, la programmation peut être réalisée soit en langage à contacts (LADDER), soit en langage blocs fonctions (FBD).

I.2.3.Modules logiques compacts

Les modules logiques compacts répondent aux besoins d'automatismes simples.

Les entrées/sorties sont au nombre de :

- 12 ou 20 E/S, alimentées en a 24 V ou c 12 V,
- 20 E/S, alimentées en a 48 V,

- 10, 12 ou 20 E/S, alimentées en a 100...240 V ou c 24 V.

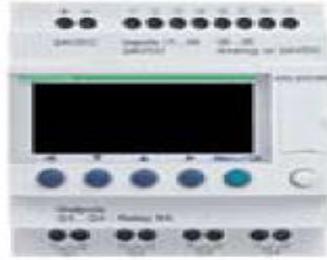


Figure I.1 : *Module Zelio Logic compact [2]*

I.2.4. Modules logiques modulaires

Les entrées/sorties pour les modules logiques modulaires sont au nombre de :

- 26 E/S, alimentées en c 12 V,
- 10 ou 26 E/S, alimentées en a 24 V, a 100...240 V ou c 24 V.

Pour plus de performance et de flexibilité, les modules Zelio Logic modulaires peuvent recevoir des extensions afin d'obtenir un maximum de 40 E/S :

- Extensions de communication réseau Modbus ou Ethernet, alimentées en c 24 V par le module Zelio Logic de même tension.
- Extension d'entrées/sorties analogiques avec 4 E/S, alimentée en c 24 V par le module Zelio Logic de même tension,
- Extensions d'entrées/sorties TOR avec 6, 10, ou 14 E/S, alimentées par le module Zelio Logic de même tension.

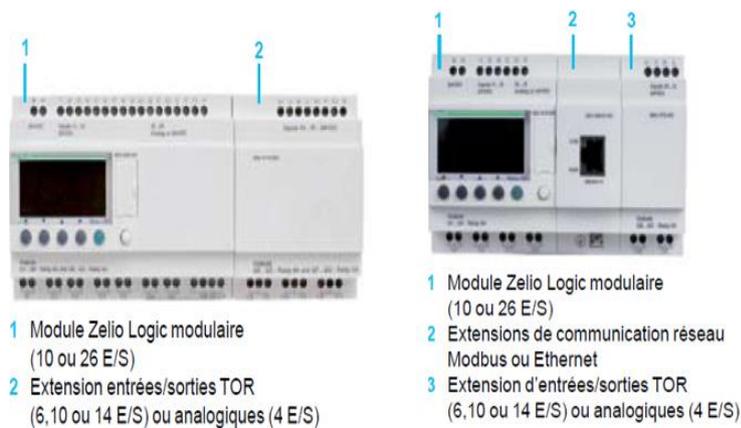


Figure I.2: *Association entre modules logiques modulaires et extensions [2]*

I.2.5. Communication [2]

Les outils de programmation permettent de connecter le module Zelio Logic au PC équipé du logiciel "Zelio Soft 2" :

- Liaison par câbles : Câble RS 232(SR2 CBL01) sur port série 9 contacts ou Câble USB (SR2 USB01) sur port USB
- Liaison sans fil : Interface Bluetooth SR2 BTC01

I.3. Contrôleurs programmables Twido

I.3.1.Présentation

Il existe deux modèles de bases Twido :

- Bases compactes
- Bases modulaires

Les bases compactes sont disponibles avec 10, 16, 24 ou 40 E/S.

Les bases modulaires sont disponibles avec 20 ou 40 E/S.

Il est possible d'ajouter des E/S supplémentaires aux bases à l'aide de modules d'expansion d'E/S.

Il s'agit des modules suivants :

- 15 modules d'expansion de type E/S TOR ou relais ;
- 10 modules d'expansion de type E/S analogique.



Figure I.3 : Contrôleurs programmables compacts Twido [6]

Nom de la base	Référence	Voies	Type de voie	Type d'entrée/sortie	Alimentation
Compact 10 E/S	TWDLCAA10DRF	6	Entrées	24 V CC	100/240 V ca
		4	Sorties	Relais	
Compact 10 E/S	TWDLCDA10DRF	6	Entrées	24 V CC	24 V CC
		4	Sorties	Relais	
Compact 16 E/S	TWDLCAA16DRF	9	Entrées	24 V CC	100/240 V ca
		7	Sorties	Relais	
Compact 16 E/S	TWDLCDA16DRF	9	Entrées	24 V CC	24 V CC
		7	Sorties	Relais	
Compact 24 E/S	TWDLCAA24DRF	14	Entrées	24 V CC	100/240 V ca
		10	Sorties	Relais	
Compact 24 E/S	TWDLCDA24DRF	14	Entrées	24 V CC	24 V CC
		10	Sorties	Relais	
Compact 40 E/S	TWDLCAA40DRF	24	Entrées	24 V CC	100/240 V ca
		16	Sorties	Relais X 14 Transistors X 2	
Compact 40 E/S	TWDLCAE40DRF	24	Entrées	24 V CC	100/240 V ca
		16	Sorties	Relais X 14 Transistors X 2 Port Ethernet	
Compact 40 E/S	TWDLCDA40DRF	24	Entrées	24 V CC	24 V CC
		16	Sorties	Relais X 14 Transistors X 2	
Compact 40 E/S	TWDLCDE40DRF	24	Entrées	24 V CC	24 V CC
		16	Sorties	Relais X 14 Transistors X 2 Port Ethernet	

Tableau I.1: Caractéristiques des différentes bases compactes Twido [6]



Figure I.4 : Contrôleurs programmables modulaires Twido [6]

Nom de la base	Référence	Voies	Type de voie	Type d'entrée/sortie	Alimentation
Modulaire 20 E/S	TWDLMDA20DUK	12	Entrées	24 V CC	24 V CC
		8	Sorties	Transistor logique négative	
Modulaire 20 E/S	TWDLMDA20DTK	12	Entrées	24 V CC	24 V CC
		8	Sorties	Transistor logique positive	
Modulaire 20 E/S	TWDLMDA20DRT	12	Entrées	24 V CC	24 V CC
		6 2	Sorties Sorties	Relais Transistor logique positive	
Modulaire 40 E/S	TWDLMDA40DUK	24	Entrées	24 V CC	24 V CC
		16	Sorties	Transistor logique négative	
Modulaire 40 E/S	TWDLMDA40DTK	24	Entrées	24 V CC	24 V CC
		16	Sorties	Transistor logique positive	

Tableau I.2 : Caractéristiques des différentes bases modulaires Twido [6]

I.3.2. Programmation

Le logiciel TwidoSuite offre une programmation aisée à partir des instructions langage liste d'instructions ou des éléments graphiques du langage à contacts.

I.4. Modicon TSX Micro

I.4.1. Présentation

L'automate TSX Micro est développé pour satisfaire au mieux les exigences d'adaptabilité et de maintenabilité des machines. Sa modularité et sa compacité répondent de manière économique à l'automatisation aussi bien de machines simples à quelques dizaines d'entrées/sorties que de machines plus complexes, jusqu'à 480 E/S. Pour AC, un simplifier le câblage des machines, le TSX Micro supporte la connexion du bus AS-i.

I.4.2. TSX 37 05 et TSX 37 08

L'automate TSX 37 05 comprend un bac intégrant une alimentation de 100/240 V processeur incluant une mémoire RAM de 11 K mots (programme, données et constantes), une mémoire de sauvegarde Flash EPROM, un module d'entrées/sorties "Tout ou Rien" TSX DMZ 28DR (16 entrées et 12 sorties à relais) et un emplacement disponible.

L'emplacement disponible peut recevoir :

- Un module d'entrées/sorties TOR au format standard de tout type ;
- Deux modules demi-format de type entrées/sorties TOR, sécurité, entrées/sorties analogiques et comptage.

La seule différence entre l'automate TSX 37 05 et le TSX 37 08, c'est que ce dernier comprend deux modules E/S TOR TSX DMZ 28DR et le premier ne comprend qu'un seul module



Figure I.5 : Automates TSX 37 05 et TSX 37 08 [3]

I.4.3.TSX 37 10

Les automates TSX 37 10 compacts et modulaires se différencient par leur tension d'alimentation et le type de module d'entrées/sorties "Tout ou Rien" implanté de base dans le premier emplacement. Chaque configuration TSX 37 10 comprend un bac intégrant une alimentation (24 V DC ou 100/240 V AC), un processeur incluant une mémoire RAM de 14 K mots (programme, données et constantes), une mémoire de sauvegarde Flash EPROM, un horodateur, un module d'entrées/sorties "Tout ou Rien" (28 ou 64 entrées/sorties) et un emplacement disponible. Un mini bac d'extension TSX RKZ 02 permet d'augmenter le nombre d'emplacements de 2 (4 positions).

Chaque emplacement disponible peut recevoir :

- 1 module d'entrées/sorties TOR au format standard de tout type
- 2 modules demi format de type entrées/sorties TOR, sécurité, entrées/sorties analogiques et comptage. De plus, les automates TSX 37 10 peuvent se connecter au réseau Ethernet TCP/IP ou à un Modem via le coupleur autonome externe TSX ETZ 410/510.

I.4.4.TSX 37 21/22

Les automates TSX 37 21/22 modulaires se différencient entre eux par leur tension d'alimentation et/ou la possibilité d'effectuer sur la base, du comptage rapide et des fonctions analogiques.

Chaque automate comprend : un bac à 3 emplacements libres intégrant une alimentation (24 V DC ou 100/240 V AC), un processeur incluant une mémoire RAM de 20 K mots (programme, données et constantes), une mémoire de sauvegarde Flash EPROM, un horodateur, 2 emplacements pour carte PCMCIA (1 carte communication et 1 carte extension mémoire de 128 K mots maximum). Un mini bac d'extension TSX RKZ 02 permet d'augmenter le nombre d'emplacements de 2 (4 positions).

Chaque emplacement disponible peut recevoir :

- 1 module d'entrées/sorties TOR au format standard ;

- 2 modules demi-format de type entrées/sorties TOR, sécurité, entrées/sorties analogiques et comptage.

De plus, les automates TSX 37 21/22 peuvent se connecter au réseau Ethernet TCP/IP ou à un Modem via le coupleur autonome externe TSX ETZ 410/510.



Figure I.6 : Automates TSX 37 10 et TSX 37 21 et TSX 37 22 [3]

I.4.5.Programmation

Les automates TSX Micro disposent, de base, des fonctions de régulation accessibles par l'utilisateur à travers le logiciel de programmation PL7 Micro/Junior/Pro.

I.4.6.Communication

Les automates TSX Micro proposent plusieurs possibilités :

- Communication intégrée qui offre des fonctions de dialogue économiques réalisées par la prise terminal pour les automates TSX 37 05/08/10 ou par les prises terminal et de dialogue opérateur pour les automates TSX 37 21/22. Ces liaisons de type RS 485 non isolées, disposent du protocole Uni-Telway maître/esclave, Modbus RTU esclave ou mode caractères. De plus les automates TSX 37 10/21/22 intègrent le protocole Modbus RTU maître ;
- Carte de communication au format PCMCIA pour les automates TSX 37 21/22. Ils sont équipés d'un emplacement dédié recevant une carte de communication au format PCMCIA (liaison série asynchrone "Full-duplex", bus CANopen, Fipio, Uni-Telway ou Modbus, et réseau Modbus Plus ou Fipway ;
- Module externe Ethernet TCP/IP 10/100 MHz. Ce module externe se connecte à la prise terminal des automates TSX 37 10/21/22 et dispose de la messagerie Uni-TE et Modbus. Il permet aussi la connexion sous protocole PPP à un modem externe.

I.5.Automates Modicon M340

I.5.1.Présentation

L'automate Modicon M340 est "la solution" pour les constructeurs de machines dans des applications telles que le packaging secondaire, la manutention, le textile, l'imprimerie, l'agroalimentaire, les machines à bois, la céramique...

L'intégration des variateurs de vitesse Altivar et Lexium, des afficheurs Magelis et des modules de sécurité Preventa a été particulièrement poussée pour simplifier la mise en œuvre et l'exploitation des solutions Télémécanique.

Modicon M340 est également le compagnon de Modicon Premium et Modicon Quantum pour répondre aux exigences d'automatisation des procédés industriels et des infrastructures, au cœur des architectures Transparent Ready.

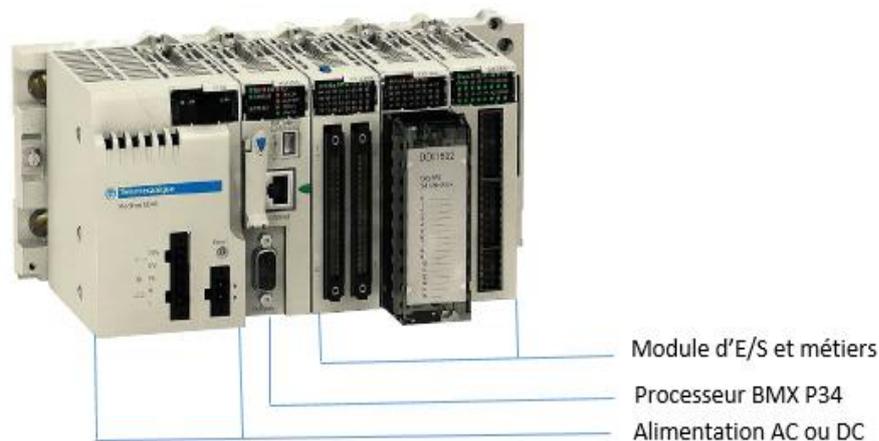


Figure I.7 : *plate-forme Modicon M340 [3]*

I.5.2.Performances

Modicom M340 est doté de performances importantes :

- 7 K instructions/ms ;
- 4 Mo de mémoire programme ;
- 70 K instructions.

I.5.3.Solutions intégrées

Modicon M340 offrent les solutions intégrées suivantes :

- Liaison USB et 2 ports de communication intégrés sur l'UC ;
- Drives sur CANopen ;
- Alimentation directe des afficheurs Magelis ;
- Fonctions métiers performantes ;
- Comptage, décomptage, mesure ;
- Commande de mouvements (position/vitesse) par librairie de Motion Function ;
- Blocks (MFB) conforme PLCopen ;

- Régulation par langage FBD et librairie complète de fonctions spéciales.

Modicon M340 est doté d'une robustesse au-delà des standards : Chocs, vibrations, immunités électriques, température et altitude.

I.5.4.Programmation [3]

Un environnement fédérateur pour les plates-formes Modicon.Unity Pro est un logiciel commun de programmation, mise au point et exploitation des automates Modicon M340, Premium, Quantum et slot PLC Atrium.

Logiciel IEC 61131-3, Unity Pro reprend les valeurs d'usage reconnues de PL7 et Concept. Il ouvre les portes d'un ensemble complet de nouvelles fonctionnalités pour plus de productivité :

- Une haute conception fonctionnelle ;
- Une standardisation aisée pour une réutilisation des développements ;
- De nombreux outils de tests programme et d'amélioration de l'exploitation des systèmes ;
- De nouveaux services de diagnostic intégré.

Le catalogue logiciel Unity propose des logiciels spécialisés pour encore plus de productivité :

- Ouverture pour des développements en langage C ou en VBA (Visual Basic Applications).
- Conception et génération d'applications batch/process avec intégration PLC/HMI...

I.6.Modicon Premium

I.6.1.Processeur Unity

I.6.1.1.Présentation

Les processeurs des plates-formes d'automatisme Modicon Premium TSX P57 pp4M, TSX P57 pp34M et TSX Hp4M gèrent l'ensemble d'une station automate constituée de :

- Modules d'entrées/sorties "Tout ou Rien".
- Modules de sécurité Preventa.
- Modules d'entrées/sorties analogiques.
- Modules métiers (comptage, mouvement, pesage, communication).

L'offre processeurs Premium est complétée par 3 références :

- TSX P57 6634M, processeur de haut de gamme intégrant 1 port Ethernet Modbus/TCP et proposant une RAM interne de 2 Mo ;
- TSX H57 24M/44M, processeurs permettant la réalisation de système de redondance Hot Standby (avec automates "Primaire" et "Secondaire").

Les processeurs proposés se différencient par leurs capacités mémoire, vitesses de traitement, nombre d'E/S et nombre de ports de communication. De plus, selon le modèle, les processeurs proposent :

- De 1 à 16 racks interconnectés par le bus X (sur une distance de 700 m maxi).
- De 192 à 2040 entrées/sorties "Tout ou Rien".
- De 12 à 512 entrées/sorties analogiques.
- De 4 à 64 voies métiers. Chaque module métier comptage, commande de mouvement, communication ou pesage comporte une ou plusieurs voies métiers.
- De 1 à 4 réseaux (Ethernet Modbus/TCP, EtherNet/IP, Fipway, Modbus Plus, Ethway), de 1 à 8 bus AS-Interface.
- De 0 ou 1 bus Fipio, de 0 ou 1 bus CANopen ou Modbus Plus et de 0 à 5 bus de terrain INTERBUS ou Profibus DP (1).
- De 0 à 30 voies de régulation, une voie pouvant avoir jusqu'à 3 boucles.



Figure I.18 : Différents processeurs TSX 57 [4]

Selon les modèles, les processeurs Premium intègrent :

- Un port Ethernet Modbus/TCP 10BASE-T/100BASE-TX (raccordement RJ45) ;
- Une liaison bus Fipio 1 Mbit/s (gestionnaire de bus) ;
- Une communication via 2 prises terminal (TER et AUX) sous protocole Uni-Telway ou mode caractères (typiquement un terminal de programmation 19 ou 115 Kbit/s et un terminal de dialogue opérateur).
- Une prise TER de type USB (pour connexion d'un terminal de programmation).

Chaque processeur dispose de 2 emplacements pour carte PCMCIA :

- Un supérieur (n° 0) pour l'accueil de cartes d'extension mémoire sauvegardée (programme, symboles, constantes et/ou fichiers de données).
- Un inférieur (n° 1) pour l'accueil (1) d'une carte réseau (Fipway, Modbus Plus) ou bus (CANopen, Fipio Agent, Modbus, Uni-Telway et liaisons séries).

Cet emplacement peut également recevoir les cartes d'extension mémoire spécifiques au stockage de données.

I.6.1.2. Conception et mise en œuvre des applications Premium

La mise en œuvre de ces processeurs Premium nécessite :

- Le logiciel de programmation Unity Pro Medium, Large ou Extra Large. Il est identique à celui mettant en œuvre les plates-formes Modicon M340 et Modicon Quantum.

Eventuellement, selon les besoins :

- Le logiciel spécialisé Unity Application Generator (UAG) pour la modélisation et la génération des applications process,
- Le logiciel Unity EFB toolkit pour le développement en langage C de bibliothèques de blocs fonction EFs et EFBs,
- Le logiciel Unity SFC View pour la visualisation et le diagnostic des applications écrites en langage diagramme fonctionnel en séquence (SFC) ou Grafcet.

I.6.2. Processeur PL7

I.6.2.1. Présentation

Les processeurs des plates-formes d'automatisme Premium TSX P57 pp3M/3AM et TSX P57 pp23M/23AM gèrent l'ensemble d'une station automate constituée de modules d'E/S "Tout ou Rien", modules de sécurité Preventa, modules d'E/S analogiques et modules métiers pouvant être répartis sur un ou plusieurs racks connectés sur le bus X ou pouvant être distribués sur bus de terrain.

Les processeurs proposés sont segmentés par des capacités différentes au niveau de la mémoire, des entrées/sorties "In rack", des communications ainsi que par leurs vitesses de traitement. Selon le modèle :

- De 4 à 16 racks.
- De 512 à 2040 entrées/sorties "Tout ou Rien".
- De 24 à 256 entrées/sorties analogiques.
- De 8 à 64 voies métiers. Chaque module métier (comptage, commande de mouvement, liaison série ou pesage) compte pour 1 ou plusieurs voies métiers.
De 1 à 4 réseaux (Ethernet Modbus/TCP, Fipway, Ethway, Modbus Plus), de 2 à 8 bus capteurs/actionneurs AS-Interface, de 1 à 2 bus de terrain (CANopen, INTERBUS, Profibus DP), 0 ou 1 bus de terrain Fipio, des liaisons séries (Modbus, Uni-Telway).
- De 10 à 20 voies de régulation.

I.6.2.2. Conception et mise en œuvre des applications Premium

Les licences logiciels PL7 Junior/Pro version V4.5 sont proposées, selon les besoins, en différentes versions :

- Monoposte,
- Multipostes sous forme de poste locaux indépendants (Junior/Pro), à distance à partir d'un serveur via réseau (Pro OpenTeam pour 3 à 10 postes ou Pro OpenSite pour > 10 postes).

Ces licences sont compatibles avec les terminaux PC équipés du système d'exploitation Windows 2000 Professional ou Windows XP.

I.7.Modicon Quantum

I.7.1.Processeur standard Unity Pro

I.7.1.1.Présentation

Les unités centrales de la plate-forme d'automatisme Modicon Quantum sont basées sur des processeurs haute performance et sont compatibles avec le logiciel Unity Pro. De nombreuses fonctionnalités sont incluses de base dans les processeurs Quantum :

- Temps de cycle réduit avec acquisition rapide des entrées/sorties ;
- Traitement d'interruption sur événement de temps ou en provenance d'entrées ;
- Traitement possible en tâche rapide comme en tâche maître ;
- Extension des capacités mémoire par cartes mémoire PCMCIA ;
- Multiples ports de communication intégrés au processeur ;
- Diagnostic et maintenance aisés grâce au bloc de visualisation LCD en face avant des processeurs haut de gamme.

Les processeurs proposés se différencient par leurs capacités mémoire, leurs vitesses de traitement et leurs possibilités de communication.



Figure I.9 : Automate Modicon Quantum [5]

I.7.1.2.Ports de communication intégrés

Les processeurs Quantum intègrent, selon modèle :

- 2 ports Modbus RS 232 (1 port Modbus RS 232/485 pour les processeurs **140 CPU 6pp pp**),
- 1 port Modbus Plus,
- 1 port Ethernet TCP/IP 10BASE-T/100BASE-TX (100BASE-FX pour les processeurs Hot Standby **140 CPU 67p 6p**),
- Port USB pour connexion d'un terminal PC de programmation pour les processeurs.

I.7.1.3. Visualisation LCD

Selon modèle, les processeurs disposent d'une visualisation LCD (2 lignes de 16 caractères) avec luminosité et contraste réglables. Les touches associées à la visualisation permettent le diagnostic, l'accès à certains paramètres de configuration et la commande d'exécution du processeur.

I.7.1.4. Conception et mise en œuvre des applications Quantum

La mise en œuvre de ces processeurs Quantum nécessite :

- Le logiciel de programmation Unity Pro Large ou Extra Large. Ces logiciels sont compatibles avec les plates-formes Premium et M340.

Eventuellement, selon les besoins :

- Le logiciel spécialisé Unity Application Generator (UAG) pour la modélisation et la génération des applications process,
- Le logiciel Unity EFB toolkit pour le développement en langage C de bibliothèques de blocs fonctions EFs et EFBs,
- Le logiciel Unity Dif pour la comparaison d'applications Unity Pro,
- Le logiciel Unity Loader pour la mise à jour des projets Unity Pro.

I.7.2. Processeurs standard Concept / ProWORX

I.7.2.1. Présentation

Les processeurs Quantum, compatibles avec les logiciels Concept et ProWORX, sont logés dans un boîtier n'occupant qu'une seule position. Ils intègrent la mémoire système, la mémoire application et les ports de communication. Tous les composants mémoire étant intégrés, il n'est pas nécessaire de recourir à des extensions mémoire pour finaliser une configuration.

I.7.2.2. Ports de communication

Tous les processeurs s'intègrent dans les architectures de réseaux Modbus et Modbus Plus. Des commutateurs rotatifs en face arrière des modules permettent de définir l'adresse du (des) port(s) Modbus Plus. Chaque équipement réseau Modbus Plus doit avoir une adresse unique dans la plage 1...64. Les réglages des ports Modbus comprennent : la vitesse, la parité, le nombre de bits de

données, le nombre de bits de stop, le protocole et l'adresse de l'Esclave. Par défaut, ces réglages sont 9600 bit/s, parité paire, 8 bits de données, 1 bit de stop, mode RTU et adresse 1.

Un commutateur en face avant des processeurs permet de paramétrer le port Modbus comme support de communication modem (2400 bit/s, parité paire, 7 bits de données, 1 bit de stop, mode ASCII et adresse 1).

Les processeurs 140 CPU 434 12A et 140 CPU 534 14B disposent de 2 ports séries Modbus :

- Port 1 Modbus, paramétrable comme modem.
- Port 2 Modbus, gestion de flux RIS/CTS (ne supporte pas la liaison modem).

I.7.2.3.Choix des langages

Tous les processeurs peuvent utiliser les langages de programmation suivants :

- Langages évolués CEI 66631-3 :
 - Langage diagramme fonctionnel en séquence (SFC) ou Grafcet,
 - Langage blocs fonctionnels (FBD),
 - Langage à contacts (LD),
 - Langage littéral structuré (ST),
 - Langage liste d'instructions (IL),
- Langage à contacts 984 : langage "bas niveau" à haute performance dont le code source de l'application réside dans l'automate.

I.8.Conclusion

Schneider Electric propose des solutions d'automatisation complète par les différentes gammes d'automates quelle met à notre disposition.

Dans le cas de notre projet, on a utilisé l'automate ZELIO compacte doté d'un afficheur(RS 2B 201 BD°, avec un nombre d'entrées/sorties bien adaptés à nos besoins.

Chapitre II :
Le logiciel de programmation
Zelio Soft

II.1. Introduction

Zelio Soft est le logiciel nécessaire pour la programmation de l'automate ZELIO , sachant que ce dernier est programmable directement sur son afficheur uniquement en langage LADDER , mais comme notre projet est une maquette sur laquelle plusieurs manipulations peuvent être effectués alors nous écrivons nos programmes avec le logiciel Zelio Soft ,les programmes seront donc sauvegardés. Pour effectuer une manipulation il suffira de transférer le programme correspondant vers l'automate,

II.2.Présentation

Le logiciel “Zelio Soft 2” permet :

- La programmation en langage à contacts (LADDER) ou en langage à blocs fonctions (FBD),
- La simulation, le monitoring et la supervision,
- Le chargement et le déchargement de programmes,
- L'édition de dossiers personnalisés,
- La compilation automatique de programmes,
- L'aide en ligne.

II.3.Mode de fonctionnement [2]

Il existe deux modes de fonctionnement pour l'atelier de programmation :

II.3.1.Mode saisie

Le mode saisie sert à construire les programmes en mode LD ou FBD, cela correspond au développement de l'application.

II.3.2.Mode mise au point

Ce mode sert à finaliser l'application, cette opération peut se faire :

- **en mode simulation:** le programme est exécuté en local directement dans l'atelier de programmation (simulé sur le PC). Dans ce mode chaque action sur le graphe (changement d'état d'une entrée, forçage d'une sortie) met à jour les fenêtres de simulation.

- **en mode Monitoring** : le programme est exécuté sur le module logique, l'atelier de programmation est connecté au module (connexion PC ↔ module).

Les différentes fenêtres sont mises à jour cycliquement.

Dans ces deux modes il est possible de :

- visualiser en dynamique (dans les fenêtres: Edition / Supervision / Face Avant), l'état des sorties et des paramètres des blocs fonctions du programme correspondant à la feuille de câblage
- forcer les entrées/sorties pour tester le comportement du programme dans des conditions particulières.

II.4.La programmation en langage LADDER

Le langage à contacts permet d'écrire un programme LADDER avec des fonctions élémentaires, des blocs fonctionnels élémentaires et des blocs fonctionnels dérivés, ainsi qu'avec des contacts, des bobines et des variables. Les contacts, les bobines et les variables peuvent être commentés. Du texte peut être inséré librement sur le graphique. [2]

II.4.1.Modes de saisie des schémas de commande

Le mode "saisie Zelio" permet à l'utilisateur ayant programmé directement sur le produit Zelio Logic de retrouver la même ergonomie, à la première prise en main du logiciel.

Le mode "saisie libre", plus intuitif, apporte à l'utilisateur un grand confort d'utilisation et de nombreuses fonctionnalités supplémentaires.

En langage de programmation LADDER, 2 types d'utilisation sont possibles :

- symboles LADDER,
- symboles électriques.

Le mode "saisie libre" permet aussi la création de mnémoniques et de commentaires associés à chaque ligne de programme. Le passage d'un mode de saisie à l'autre est possible à tout instant, par

un simple clic de souris. Il est possible de programmer jusqu'à 120 lignes de schémas de commande, avec 5 contacts et 1 bobine par ligne de programmation.

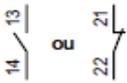
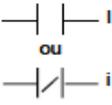
Fonction	Schéma électrique	Langage LADDER	Commentaire
Contact			i correspond à l'image réelle du contact câblé sur l'entrée du module. i correspond à l'image inverse du contact câblé sur l'entrée du module.
Bobine classique			La bobine est excitée lorsque les contacts auxquels elle est reliée sont passants.
Bobine à accrochage (Set)			La bobine est excitée lorsque les contacts auxquels elle est reliée sont passants. Elle reste enclenchée lorsque les contacts ne sont plus passants.
Bobine de décrochage (Reset)			La bobine est désexcitée lorsque les contacts auxquels elle est reliée sont passants. Elle reste inactivée lorsque les contacts ne sont plus passants.

Tableau II.1: Fonctions en symboles LADDER et symboles électriques [2]

II.4.2.Fonctionnalités

Le logiciel Zelio Soft 2 offre les fonctionnalités suivantes :

- 16 blocs fonctions Textes,
- 16 temporisateurs, chacun paramétrable parmi 11 types différents (1/10ème de secondes à 9999 heures),
- 16 compteurs/décompteurs de 0 à 32767,
- 1 compteur rapide (1 kHz),
- 16 comparateurs analogiques,
- 8 horloges, disposant chacune de 4 canaux,
- 28 relais auxiliaires,
- 8 comparateurs de compteurs,
- écran LCD avec rétroéclairage programmable,
- passage automatique heure d'été/heure d'hiver,
- diversité des fonctions : bobine, à mémoire (Set/Reset), télérupteur, contacteur,
- 28 blocs messages.

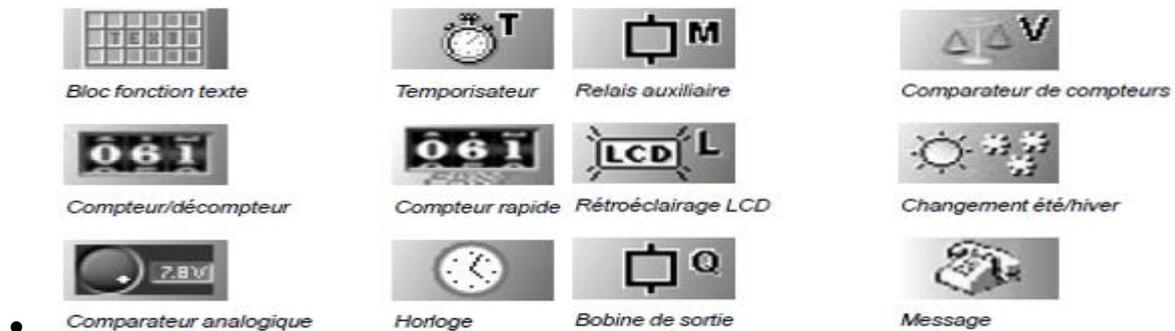


Figure II.1: Différents outils de programmation en langage LADDER [2]

II.4.3. Description d'un réseau de contacts

Un réseau de contacts est composé d'un ensemble d'éléments graphiques disposés sur une grille de :

- 120 lignes maximum de programme,
- chaque ligne est composée de 5 contacts maximum et d'une bobine.

Il est réparti en deux zones :

- la zone test, dans laquelle figurent les conditions nécessaires au déclenchement d'une action (contacts),
- la zone action, qui applique le résultat consécutif à une combinaison logique de test (bobines).

La figure ci-après décrit la structure d'un réseau de contacts.

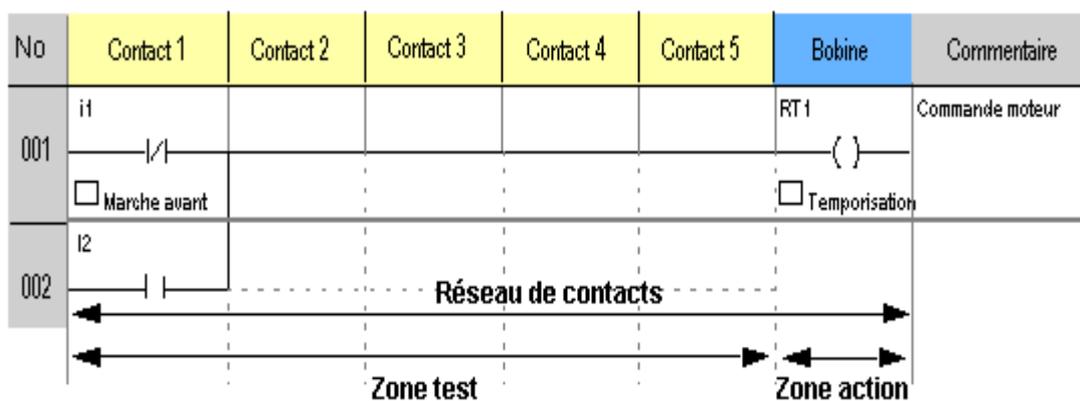


Figure II.2 : Structure d'un réseau de contacts [1]

II.5.La programmation en langage FBD [1]

II.5.1.Présentation

Le langage FBD permet une programmation graphique basée sur l'utilisation de blocs fonctionnels prédéfinis, il propose l'utilisation de :

- 34 fonctions préprogrammées pour le comptage, la temporisation, la minuterie, la définition de seuil de commutation, (exemple : régulation de température), la génération d'impulsion, la programmation horaire, le multiplexage, l'affichage,
- 7 fonctions SFC,
- 6 fonctions logiques.

II.5.2.Fonctions en langage FBD

- **Les Entrées TOR (Tout Ou Rien)** 

Il est possible de personnaliser l'application en choisissant une autre Icône pour matérialiser par exemple un détecteur de présence ou un bouton poussoir lumineux.

- **Les Entrées analogiques** 

Ce type d'entrée accepte en entrée une tension de 0 à 10 V correspondant à une valeur de 0 à 255. Il est possible de personnaliser l'application en choisissant une autre Icône pour matérialiser par exemple un capteur de température ou un potentiomètre.

- **Les Entrées filtrées** 

On peut insérer dans le câblage des entrées digitales ou analogiques filtrées. Ces types d'entrées peuvent servir à éliminer des parasites.

- **L'Entrée d'un entier (NUM IN)** 
- **Les Constantes** 

On peut faire intervenir dans le câblage des constantes. Il existe des constantes analogiques et des constantes numériques.

- **L'Horloge 1 sec** 

On peut câbler en entrée une horloge de 1 seconde.

- **Changement d'heure été/hiver** 

La sortie de cette fonction est à l'état ARRET pendant toute la durée de l'heure d'hiver et passe à l'état MARCHÉ pendant toute la durée de l'heure d'été. Elle permet, par exemple, de signaler à l'écran le changement d'horaire.

- **Les boutons** 

On peut utiliser les 4 boutons de façade du Zelio **Z1, Z2, Z3, Z4** dans notre application.

- **La sortie TOR (Tout ou Rien)**

Il est possible de personnaliser l'application en choisissant une autre Icône pour matérialiser par exemple un ventilateur ou une résistance chauffante.

Pour changer d'Icône, placez un bloc TOR sur la feuille de câblage

- **La sortie d'un entier (NUM OUT)** 
- **La sortie rétro-éclairage** 

Cette sortie permet de piloter le rétro-éclairage de l'écran du module.

II.5.3. Blocs Fonctions FBD (Function Bloc Diagram)

- **Le temporisateur A/C** 

Il permet d'appliquer au signal de sortie soit un retard de passage à ON soit un retard de passage à OFF ou les deux retards par rapport au signal d'entrée. Ce bloc peut-être utilisé pour faire une minuterie fonction A ou fonction C.

- **Le temporisateur B/H** 

Il permet de mettre à l'état haut le signal de sortie pendant un temps choisi, celui-ci est déclenché par une impulsion en entrée (fonction B) ou lorsque l'entrée est à l'état haut (fonction H).

- **Le temporisateur BW** 

Il fournit une impulsion de la durée d'un cycle sur le front montant ou descendant ou sur les deux fronts d'une entrée en fonction du réglage choisi dans les paramètres.

- **Le Clignotant** 

Il permet de générer des impulsions sur front montant de l'entrée.

- **Le bistable** 

Le principe de ce bloc est bien connu puisqu'il s'agit du télérupteur. Il suffit d'une première impulsion pour mettre la sortie à 1 puis il en faut une seconde pour faire passer la sortie à 0.

- **La bascule** 

C'est un élément qui est composé de deux entrées : **R** et **S**. R pour Reset et S pour Set. Pour activer la sortie, il suffit de générer une impulsion sur S, pour la désactiver, il faut générer une impulsion sur R. La priorité sert à définir l'état de la sortie lorsque les deux entrées sont à 1.

- **La fonction booléenne** 

Elle accepte quatre entrées. La sortie réagit en fonction de la table de vérité décrite dans les paramètres.

Pour accéder aux paramètres de la fonction booléenne, il suffit de double-cliquer sur le bloc ou de faire un clic droit et de sélectionner fenêtre de paramétrage.

- **L'arbre à Came** 

Cette fonction sert à réaliser un programmeur à Cames

- **Le compteur** 

Cette fonction permet de compter jusqu'à une valeur définie dans la fenêtre de paramétrage.

Lorsque cette valeur est atteinte, la sortie passe à 1 jusqu'à la remise zéro si la sortie fixe est choisie ou durant un certain temps si la sortie impulsionnelle est choisie. La valeur de comptage et la valeur maximum peuvent être visualisées. Il est possible de compter de zéro vers la valeur définie (comptage) ou de la valeur définie vers zéro (décomptage).

Le bloc UP DOWN COUNT permet de mettre en entrée la valeur de présélection, alors qu'elle est programmable pour le bloc PRESET COUNT.

- **Le compteur horaire** 

Cette fonction mesure la durée de l'état à 1 de l'entrée. Au-delà d'une durée présélectionnée, la sortie change d'état. Ce bloc peut par exemple servir d'alerte sur une machine pour la maintenance.

- **Le programmeur horaire hebdomadaire et annuel** 

Cette fonction sert à activer ou désactiver la sortie à un moment bien précis dans la journée, la semaine ou l'année. Ce bloc fonctionne avec un principe d'événements.

On peut utiliser le calendrier à droite de l'écran.

- **La fonction gain** 

Cette fonction permet l'utilisation d'un facteur d'échelle, elle est applicable à toutes les données analogiques.

- **Le trigger de Schmitt** 

La sortie change d'état si l'entrée est inférieure à la valeur minimale, la sortie change à nouveau d'état si l'entrée est supérieure à la valeur maximale. Si l'entrée est située entre les deux, la sortie

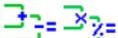
reste inchangée. Cette fonction est utilisée pour situer un seuil haut et un seuil bas par rapport à une variable analogique.

- **La fonction multiplexeur** 

Cette fonction permet de sélectionner la voie A ou la voie B en sortie.

- **La comparaison de zone** 

Utilisée pour des applications utilisant des données analogiques.

- **Les opérations + - x /** 

La combinaison de ces deux blocs nous permet de réaliser de nombreuses opérations avec les constantes numériques.

- **L' affichage sur le LCD** 

Ce bloc permet d'afficher du texte ou un entier sur l'afficheur LCD en face avant du module. On peut par exemple afficher un décimal à partir d'un entier.

- **La comparaison de deux valeurs** 

Ce bloc permet de comparer deux valeurs analogiques grâce aux opérateurs =, >, >=, <, <=, =. La sortie est du type TOR et elle est activée si la comparaison est vraie.

- **La fonction état module** 

Cette fonction permet de connaître l'état du module.

- **La fonction d'archivage**  ARCHIVE

Proposant en sortie plusieurs informations, dont notamment l'heure et la date, cette fonction sert par exemple à afficher ces informations à l'écran et à les rendre modifiables.

II.5.4.Fonctions logiques

Dans le langage FBD il est possible d'utiliser dans les schémas blocs des fonctions logiques. Les fonctions disponibles sont :

- la fonction NON,
- la fonction ET,
- la fonction OU,
- la fonction NON ET,
- la fonction NON OU,

- la fonction OU EXCLUSIF.

Le tableau suivant présente les différentes fonctions logiques :

Fonction	Symbole	Description	Nombre d'entrée	Type entrée
NON		Si l'entrée est inactive ou pas connectée, la sortie est active. Si l'entrée est active, la sortie est inactive.	1	TOR
ET		Si toutes les entrées sont actives ou pas connectées, la sortie est active. Si au moins une entrée est inactive, la sortie est inactive.	4	TOR
OU		Si au moins une entrée est active, la sortie est active. Si toutes les entrées sont inactives ou pas connectées, la sortie est inactive.	4	TOR
NON ET		Si au moins une entrée est inactive, la sortie est active. Si toutes les entrées sont actives ou pas connectées, la sortie est inactive.	4	TOR
NON OU		Si toutes les entrées sont inactives ou pas connectées, la sortie est active. Si au moins une entrée est active, la sortie est inactive.	4	TOR
OU EXCLUSIF		Si une entrée est inactive et l'autre entrée est active ou pas connectée, la sortie est active. Si les 2 entrées sont actives ou inactives ou pas connectées, la sortie est inactive.	2	TOR

Tableau II.2 : Fonctions logiques [1]

II.5.5. Grafcet / SFC (Sequential Function Chart)

Les fonctions SFC (Diagramme Fonctionnel en Séquence) sont assimilables au langage Grafcet de la norme IEC 1131-3.

Le Grafcet permet de représenter graphiquement et de façon structurée le fonctionnement d'un automate séquentiel.

Le principe est simple, un graphe avec des fonctions SFC se lit de haut en bas et se compose principalement :

- d'étapes,
- de transitions.

Les étapes se succèdent les unes aux autres encadrées par des transitions. Lorsqu'une étape est active, il faut attendre que la transition qui suit soit active pour passer à l'étape suivante. A chaque

étape est associée une action (Sortie d'étape) qui transmet des ordres à d'autres fonctions (Sortie TOR, logiques, standards). Le tableau suivant présente les différentes fonctions qui composent un programme SFC :

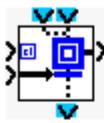
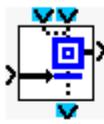
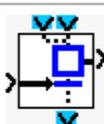
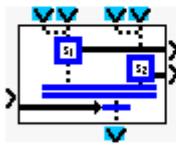
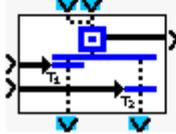
Désignation	Symbole	Description
<u>Etape Initiale</u>		Etape initiale d'un graphe SFC.
<u>Etape Initiale réinitialisable</u>		Etape initiale d'un graphe SFC avec initialisation de l'étape par une commande. Initialise la totalité du graphe connexe qui contient le reset init.
<u>Etape</u>		Etape qui transmet un ordre à une autre fonction FBD.
<u>Divergence ET</u>		Transition de une ou deux étapes vers deux étapes.
<u>Convergence ET</u>		Transition de deux étapes simultanées vers une étape.
<u>Divergence OU</u>		Transition d'une étape vers une ou deux étapes.
<u>Convergence OU</u>		Transition de une à quatre étapes vers une seule.

Tableau II.3: Fonctions FC [1]

II.6. Transfert d'une application

II.6.1 Ecrire d'un PC vers le Zelio Logic

Lorsque notre application est au point, on peut la transférer dans le Zelio Logic. Pour envoyer un programme vers le Zelio Logic, on doit aller dans le menu **Transfert, Transférer Programme** puis cliquez sur **PC->Module**.

Si le type de module sélectionné n'est pas compatible avec le type de module connecté, on peut changer le type de module dans **Module** puis **Choix du Module/Programmation**. Il est également possible d'effectuer un diagnostic du module connecté dans **Module** puis **Diagnostic du Module**.

Si le module connecté est en mode **RUN**, il est impossible de transférer le programme. On peut le mettre en mode **STOP** à partir du logiciel en sélectionnant **Transfert** puis **STOP Module**.

Si le type de module sélectionné est le même que le type de module connecté, la fenêtre de dialogue suivante s'affiche :



En (2) on choisit de rendre ou non les touches Zx (qui sont utilisées en entrée dans le programme) accessibles. Il est possible de protéger le programme présent dans le module par un mot de passe (3) qu'on saisit en (4). Pour enregistrer notre application sur notre ordinateur, on coche (5). Enfin pour lancer le mode monitoring, on coche (6). En cliquant ensuite sur **OK** le programme est transféré.

Le programme qui était présent dans le module avant le transfert est écrasé par le module.

Pour mettre en marche le module à partir du logiciel, on clique sur **Transfert** puis **RUN Module**.

Toutefois, lorsqu'on transfère un programme sur le module, celui-ci se met en mode **RUN** automatiquement.

II.6.2. Transfert du programme Zelio Logic vers le PC

Cette fonction de transfert permet de récupérer une application d'un module à l'aide du logiciel.

A partir du logiciel, On va dans le menu **Transfert, Transférer Programme** puis on clique sur **Module-> PC**. Après une demande de confirmation, le transfert s'effectue. Le logiciel charge alors le programme présent dans le module.

Si le programme du module est verrouillé, le code vous sera demandé par le module avant le transfert.

II.6.3.Fonction mot de passe

Le mot de passe protège l'accès à un programme. Lorsqu'on transfère un programme dans le Zelio Logic, la fenêtre option d'écriture s'ouvre et on peut cocher la case « Protéger par mot de passe... ». Une fois le mot de passe activé, on ne peut plus écrire vers le module ni lire le programme sans connaître ce mot de passe. Le programme est ainsi protégé.

II.7.Verrouillage façade

La fonctionnalité de verrouiller la façade sert à interdire tout accès aux menus. Le verrouillage est effectif lorsque le programme est en marche, mais aussi lorsqu'il est à l'arrêt. Pour mettre le programme en marche ou en arrêt une fois le verrouillage activé, il faut passer par le logiciel. Néanmoins, le verrouillage de la façade avant n'interdit pas l'utilisation des boutons de façade dans un programme.

Lorsqu'on écrit un programme dans le Zelio Logic, la fenêtre option d'écriture s'ouvre. Il suffit alors de cocher la case « mettre un verrou sur la face avant du Module ».

II.8.Exemples d'application

On prend l'exemple de la gestion d'un parking souterrain. A partir d'un cahier des charges donné, on développe l'application à programmer dans le module logique.

II.8.1.Exemples d'application en langage LADDER

➤ Cahier des charges

Nous désirons compléter et centraliser la gestion du parking souterrain d'un bâtiment administratif.

- **Barrière automatique**

L'entrée et la sortie de ce parking sont réalisées par une barrière automatique usuelle.

La barrière intègre des fonctionnalités de base telles que :

- ❖ l'ouverture et la fermeture temporisée au passage des véhicules,
- ❖ la gestion des tickets de paiement,
- ❖ l'interphone de sécurité,
- ❖ la commande externe de blocage de l'entrée en position fermée, etc.

- **Comptage des véhicules**

En complément, nous voulons comptabiliser le nombre de véhicules stationnés dans le parking.

On peut alors commander un panneau lumineux qui signale aux usagers que toutes les places sont occupées et interdire l'accès par le blocage de la barrière d'entrée en position fermée.

L'automobiliste sait qu'il doit alors trouver une place en surface.

Il doit être possible d'inhiber ce blocage s'il est nécessaire de faire intervenir des véhicules (pompiers, médecins...).

- **Horaires d'ouverture**

Nous désirons également interdire l'accès au parking lors de la fermeture du centre.

On permet cependant aux agents de sécurité d'inhiber ce blocage lors d'évènements exceptionnels.

Les horaires d'ouverture sont les suivants : du lundi au vendredi de 8H30 à 17H30, le samedi de 9H30 à 12H00 et fermeture complète le dimanche.

- **Evacuation des gaz toxiques**

Pour des raisons de sécurité, nous devons également évacuer les émanations de gaz toxique tel que le CO₂. Un ventilateur est mis en marche lorsque la concentration mesurée dépasse les normes autorisées.

On utilise un capteur spécialisé fournissant une valeur de sortie comprise entre 0 et 10V.

- **Eclairage**

De plus, nous voulons piloter l'éclairage à l'arrivée d'un véhicule et par l'intermédiaire de boutons poussoir situés aux différents accès piétons. Pour des raisons d'économies, l'éclairage s'éteindra au bout de 10 minutes. C'est le temps usuellement constaté pour se garer, sortir de son véhicule et monter dans les ascenseurs, ou bien pour atteindre son véhicule et sortir du parking.

- **Comptage manuel**

En complément, une intervention manuelle devra permettre de mettre à jour le nombre de véhicules situés dans le parking. On pourra incrémenter ou décrémenter manuellement le nombre de véhicules connus par le module logique.

- **Analyse du cahier des charges**

- Les entrées :

Label module logique	Désignation
Entrée I1	Détection de l'entrée d'un véhicule.
Entrée I2	Détection de la sortie d'un véhicule.
Entrées I3 et I4	Boutons poussoir des accès piétons. Ils permettent d'éclairer le parking. Un pour l'ascenseur et un pour l'escalier (aucun accès par l'entrée véhicules n'est autorisé pour les piétons).
Entrée analogique IB	Capteur de niveau de CO ₂ .

- Les boutons :

Label module logique	Désignation
Touche fonction Z1	Incrémentation manuelle du nombre de véhicules situés dans le parking.
Touche fonction Z2	Reprise de la gestion automatique de l'entrée.
Touche fonction Z3	Décrémentation manuelle du nombre de véhicules situés dans le parking.
Touche fonction Z4	Blocage manuel de l'entrée

- Les sorties :

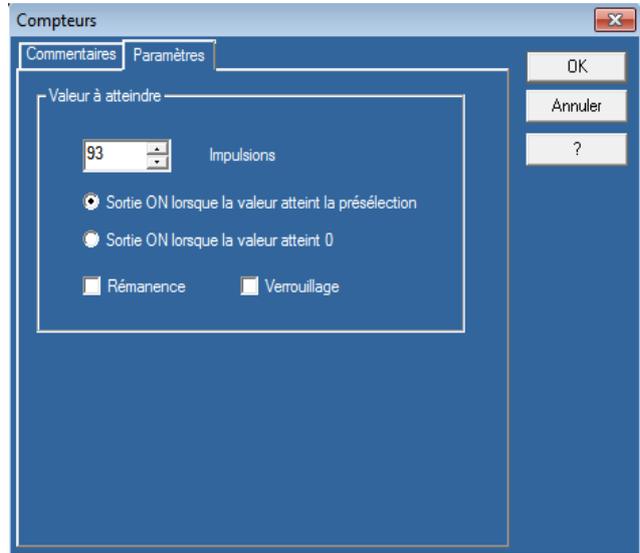
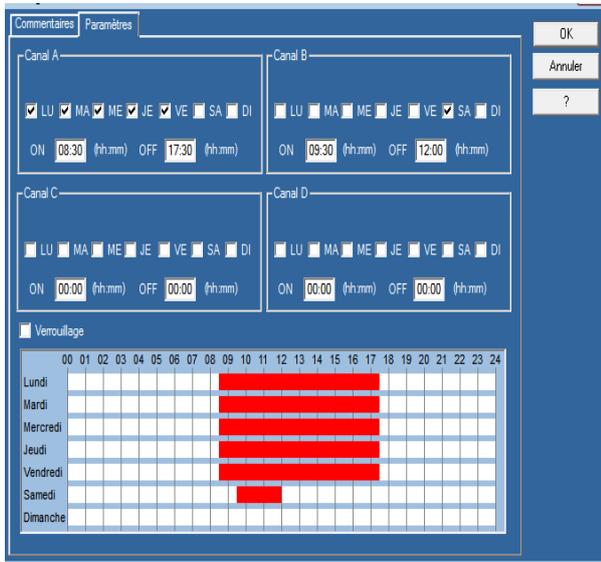
Label module logique	Désignation
Sortie Q1	Indication que le parking est complet.
Sortie Q2	Blocage de l'entrée (interdiction d'ouverture de la barrière d'entrée) lorsque le parking est complet ou que les horaires d'ouverture sont dépassés.
Sortie Q3	Eclairage du parking
Sortie Q4	Commande du ventilateur d'extraction d'air vicié.

- Les blocs fonctions spéciaux

Label module logique	Désignation
Compteur C1	Comptage du nombre de véhicules situés dans le parking (maximum 93).
Bloc fonction Horloge H1	Gestion des horaires d'accès au parking.
Bloc fonction Temporisateur T1	Temporisation de l'éclairage (10 minutes).
Bloc fonction analogique A1, la valeur de seuil autorisée correspond à 8,5 Volts.	Comparaison de la mesure de CO ₂ avec le seuil autorisé.
Bloc fonction Temporisateur T2	Temporisation de la ventilation (15 minutes).

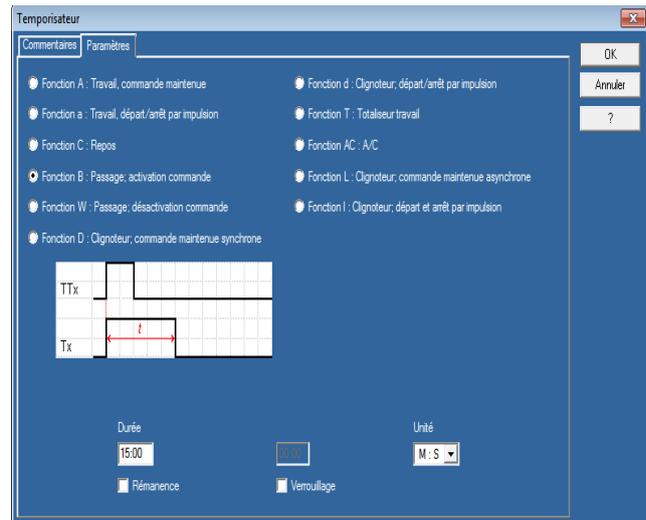
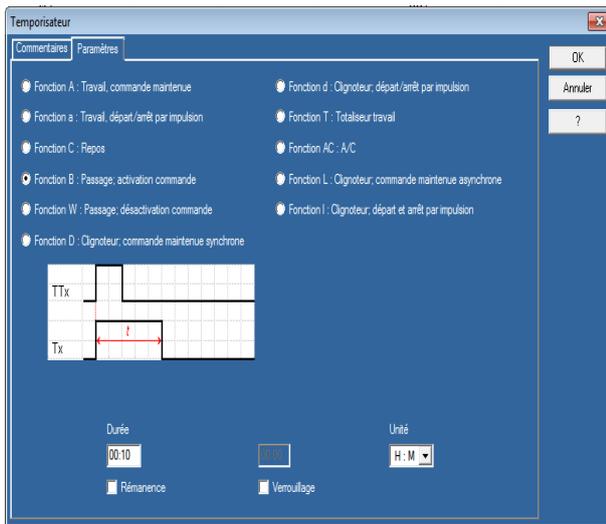
➤ Réalisation du schéma de commande

- Paramétrage de l'horloge :
- Paramétrage du compteur :



• Paramétrage du temporisateur T1

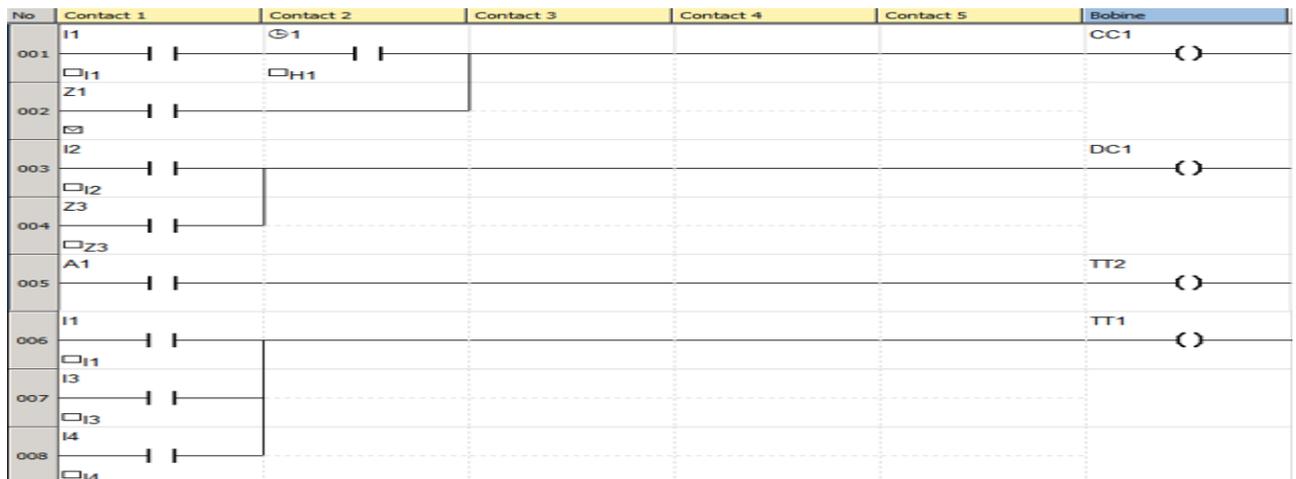
• Paramétrage du temporisateur T2



• Paramétrage du bloc de comparaison analogique (pour la mesure de la quantité du CO2)



- **Comptage et décomptage du nombre de voiture dans le parking et le lancement des temporisateurs T1 et T2**



- **Commande des sorties**

No	Contact 1	Contact 2	Contact 3	Contact 4	Contact 5	Bobine	Commentaire
009	<input type="checkbox"/> I4 Z4					SQ2 ()	
010	<input type="checkbox"/> Z2					<input type="checkbox"/> Q2 RQ2 ()	
011	C1					<input type="checkbox"/> Q1 [Q1 ()	parking complet
012	<input type="checkbox"/> H1 T1	<input type="checkbox"/> 1				<input type="checkbox"/> Q1 [Q2 ()	interdiction de passage (parking complet ou horaires dépassés)
013						<input type="checkbox"/> Q2 [Q3 ()	Eclairage du parking
014						<input type="checkbox"/> Q3 [Q4 () <input type="checkbox"/> Q4	commande du ventilateur d'extraction du CO2

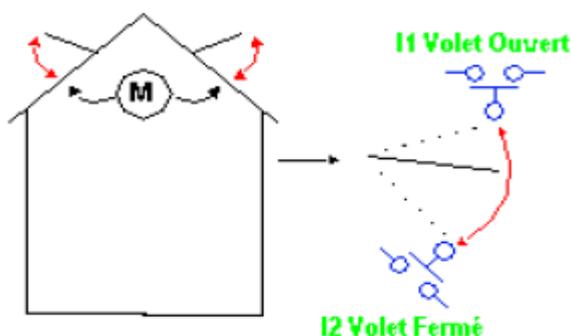
II.8.2.Exemple d'application en langage FBD

Cet exemple décrit la gestion de l'ouverture automatique des volets d'une serre.

➤ Cahier des charges

Le propriétaire d'une serre désire s'équiper d'une installation capable de gérer l'ouverture et la fermeture des volets d'aérations situés sur le toit de sa serre.

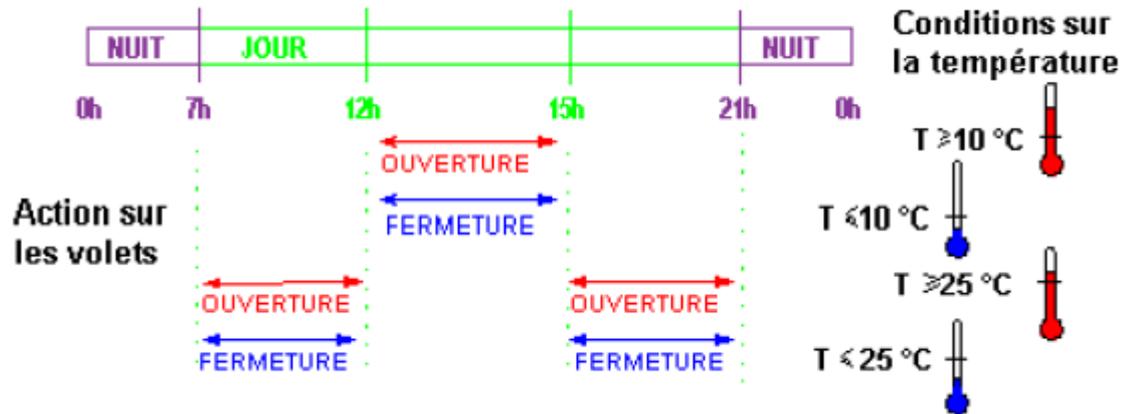
La serre présente 2 volets assurant le renouvellement de l'air. L'ouverture de ces volets est commandée par un moteur et 2 capteurs indiquent si les volets sont ouverts ou fermés :



La journée, les volets ouvrent de 12h à 15h pour aérer au moment où la température est en principe, la plus élevée. Toutefois, si la température se situe en dessous de 10°C, les volets ne s'ouvrent pas ou se ferment dans le cas où ils seraient déjà ouverts. De plus, les volets s'ouvrent la journée de 7h à 12h et de 15h à 21h lorsque la température atteint 25°C. Si la température redescend en dessous de 25 °C, les volets doivent se refermer. Enfin, lorsqu'il fait nuit, les volets restent fermés quelle que soit la température. Trois plages horaires sont utilisées :

- Plage 1 : Nuit, de 21h à 7h
- Plage 2 : Journée, de 7h à 12h et de 15h à 21h

- Plage 3 : Midi, de 12 h à 15 h



➤ **Analyse du cahier des charges**

- Les entrées :

Entrée	Description
I1	Volets ouverts (TOR)
I2	Volets fermés (TOR)
IB	Température (analogique)

- Les sorties :

Entrée	Description
Q1	Ouverture des volets (TOR)
Q2	Fermeture des volets(TOR)

La température provient d'un capteur délivrant en sortie une tension de 0 à 10 V.

Pour cette application il faut un contrôleur possédant une horloge et des entrées analogiques.

➤ **Réalisation du schéma de commande**

- Comparateur analogique B10

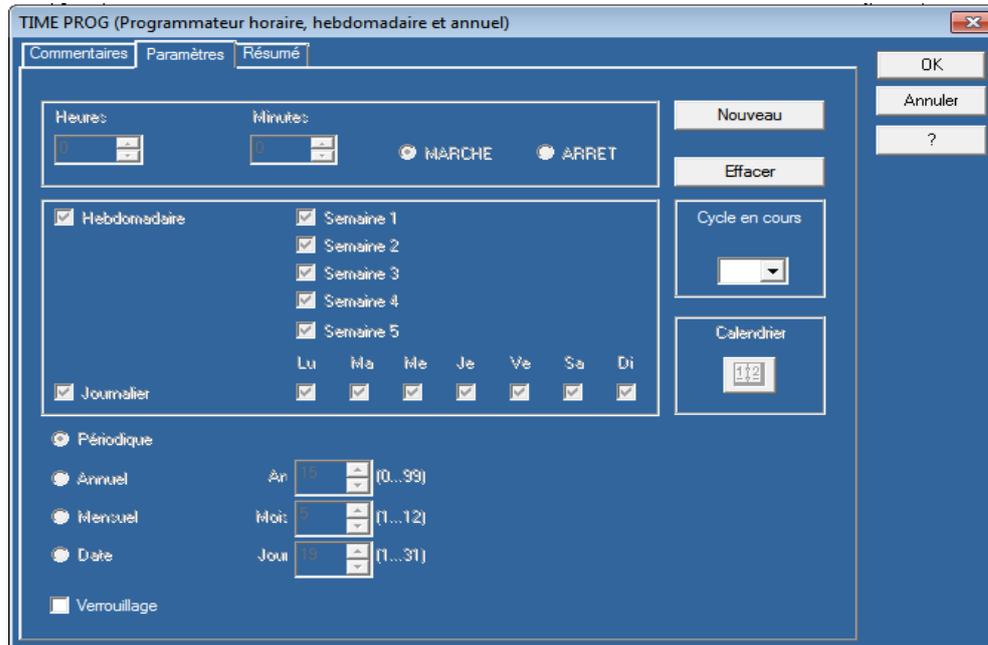
Valeur1 (température) > Valeur2 (25 deg)

- Comparateur analogique B18

Valeur1 (température) > Valeur2 (25 deg)

- Programmateur horaire B12

Cycle en cours : 00



- Heure: 12,
- Minute: 00,
- MARCHE est sélectionné,
- Toutes les semaines et les jours sont sélectionnés
- On fait un programme périodique

Cycle en cours : 01

- Heure: 15,
- Minute: 00,
- ARRET est sélectionné,
- Tous les autres paramètres sont identiques que pour MARCHE

➤ Programmeur horaire B13

Cycle en cours : 00

- Heure: 07,
- Minute: 00,
- MARCHE est sélectionné,
- Tous les autres paramètres sont identiques que pour le programmeur B12

Cycle en cours : 01

- Heure: 12,
- Minute: 00,
- ARRET est sélectionné,
- Tous les autres paramètres sont identiques que pour MARCHE

Cycle en cours : 02

- Heure: 15,
- Minute: 00,
- MARCHE est sélectionné,
- Tous les autres paramètres sont identiques que pour le programmeur B1

Cycle en cours : 03

- Heure: 21,
 - Minute: 00,
 - ARRET est sélectionné,
 - Tous les autres paramètres sont identiques que pour MARCHE
- Le programmeur : B14

Cycle en cours : 00

- Heure: 21,
- Minute: 00,
- MARCHE est sélectionné,
- Tous les autres paramètres sont identiques que pour le programmeur B12

Cycle en cours : 01

- Heure: 07,
- Minute: 00,
- ARRET est sélectionné,
- Tous les autres paramètres sont identiques que pour MARCHE

➤ La fonction booléenne B11 et B15

On va définir les tables de vérité de ces fonctions booléennes de la façon suivante :

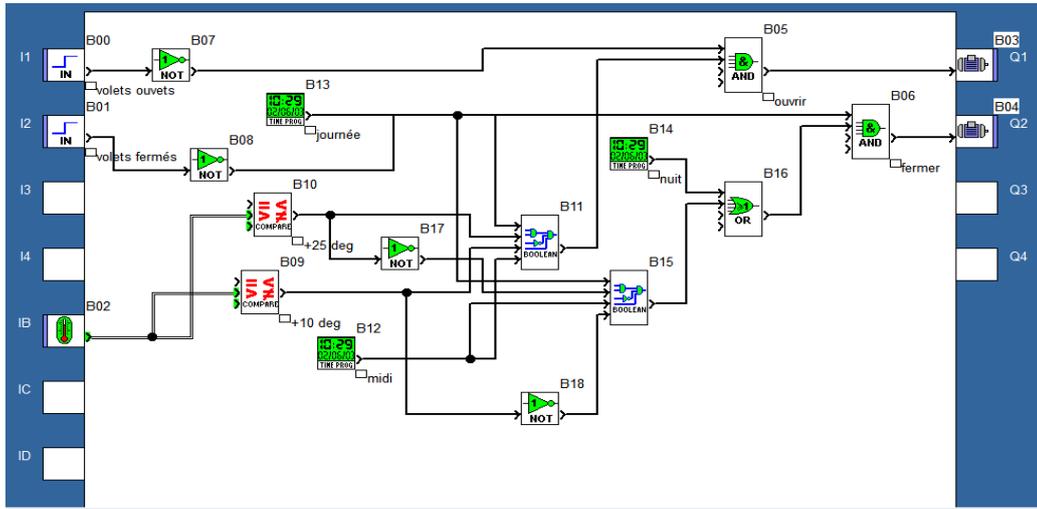
Option

Sortie ARRET si résultat VRAI

Sortie MARCHE si résultat VRAI

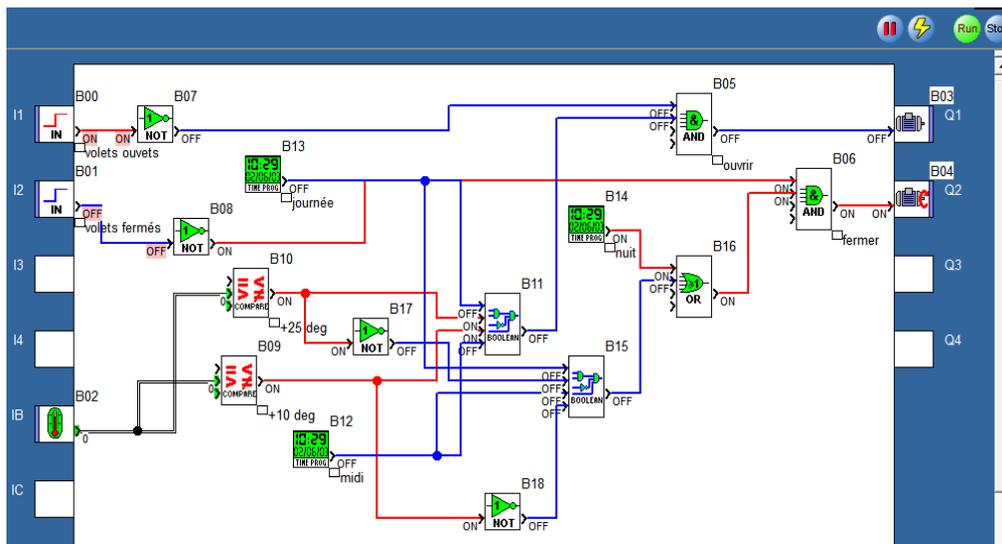
Entrée 1	Entrée 2	Entrée 3	Entrée 4	Sortie
0	0	0	0	0
1	0	0	0	0
0	1	0	0	0
1	1	0	0	1
0	0	1	0	0
1	0	1	0	0
0	1	1	0	0
1	1	1	0	1
0	0	0	1	0
1	0	0	1	0
0	1	0	1	0
1	1	0	1	1
0	0	1	1	1
1	0	1	1	1
0	1	1	1	1
1	1	1	1	1

Voici le schéma de commande final :



➤ **Simulation**

On simule le cas où il est 21 heure et à 21 heure 00 min les volets étaient ouverts alors on doit actionner le moteur pour les fermer :



II.9.Conclusion

Intégrant les langages et les outils nécessaires pour une programmation complète et souple, le logiciel Zelio Soft 2 nous offre la possibilité et la facilité de commander de petits équipements d'automatisme, comme on le fera avec le prototype de la station de pompage au sein du laboratoire de l'automatique.

ChapIII :

Réalisation et Gestion du prototype de la station de pompage par
l'automate Zelio

III.1.Introduction

Au sein du laboratoire de l'automatique , on a réalisé une maquette de station de pompage, la gestion de cette maquette est effectuée à la base par un automate programmable Siemens , dans ce chapitre on va élaborer un cahier de charge à réaliser sur le prototype et on va le programmer par le logiciel Zelio soft 2 afin qu'il soit transféré vers un automate Zelio de Schneider, On aura donc la possibilité d'effectuer des travaux pratique sur la maquette pour les étudiants de la 2^{ème} année (sur Zelio soft 2) et pour la 3^{ème} année (sur step7 et WinCC).

III.2. Description du Prototype de la station de pompage

Les éléments et composants utilisés pour la construction de notre prototype, ont été choisis fonction des paramètres, qualité, disponibilité et coût.

La Figure donne une vue générale de la maquette, qui permet réaliser différentes manipulations de même principe que les stations de pompage hydraulique.

Le principe de fonctionnement sera décrit en détail dans le cahier de charge établi.

L'ensemble des éléments et les caractéristiques du prototype réalisé (en considérant que la station n'es gérée que par le module programmable ZELIO, sont présentés comme suit :

- Le prototype repose sur un support métallique 1.50 m sur 1m et d'une hauteur de 1.50m qui comporte toutes les organes de la station.
- Les trois réservoirs sont en plexi glace transparent dimensionnés comme suit :
Deux réservoirs (A et B) de 0.15m sur 0.15m sur une hauteur de 0.30m
Un réservoir (C) de 0.50m sur 0.20m sur une hauteur de 0.20m
- Une pompe centrifuge QB50, couplé à un moteur asynchrone monophasé alimenté à 220 V La hauteur manométrique maximale de cette pompe est de 20 m, protégé par un cerveau pour éviter la marche à sec de la pompe.
- Un mélangeur(Agitateur) alimenté en 220 v pour agiter le liquide.
- Un démarreur progressif ATS01N03FT qui améliore les performances de démarrage du mélangeur en permettant un démarrage progressif et contrôlé.
- Les éléments de la tuyauterie sont en PEHD (Polyéthylène Haute Densité) Tigre de diamètre DN40,
- Les deux robinets d'arrêt à opercule DN40, utilisés pour permettre la flexibilité du prototype afin qu'on peut l'utiliser dans deux manipulations différentes (ou plus).

- Deux relais de niveau JYB-714B alimenté en 220V et muni 2 bougies de sonde pour détecter les niveaux haut des réservoirs A et B.
- Une alimentation stabilisée (Harvest electric HAS-120-24) alimenté en monophasé 220 V et qui délivre du 24V.
- Deux flotteurs de niveau pour détecter les niveaux bas.
- Deux électrovannes DN40 TOR alimentées en 24V DC.
- Un Automate Zelio pour la gestion
- Deux relais électromagnétiques 8 broches servant comme interface de sortie entre l'automate et les actionneurs (pompe et l'agitateur).
- Deux disjoncteurs : un pour la sécurité de toute l'installation, l'autre pour la sécurité de l'automate ZELIO



1 Automate Zelio

2 Voyants

3 Boutons(Marche –Reset)

4 Reservoir B

5 Agitateur

6 Reservoir C

7 Reservoir A

8 Bougie de sonde

9 servomoteur

10 Pompe

Figure III.1: *Prototype de la station de pompage*

III.3. Eléments de commande du Prototype de la station de pompage

Pour gérer la station avec un automate ZELIO nous avons utilisé :

- Un automate Zelio de Schneider Electric (SR2 B201BD) des caractéristiques suivant :

- 4 entrées analogiques
- 6 entrées TOR
- 8 sorties TOR (à relais : 8 A)
- Un port de raccordement au PC
- Un afficheur LCD, 4 lignes, 18 caractères
- Touche Menu/OK et touches de navigation

- PC avec un système d'exploitation Windows XP.
- Logiciel Zelio Soft 2 (v 4.5)
- Câble de communication entre le PC et l'automate (adaptateur PC)

III.4. Développement du projet pour gestion du prototype de la station de pompage

L'étape suivante consiste à créer et développer le programme de contrôle et de commande qui sera implémenté dans l'automate afin de gérer notre installation.

Le graphique ci-après montre ces étapes de configuration de projet et les objets à créer :

Etape	Contenu de la tâche	Réalisation
1	Création du projet « Maquette de station de pompage »	<ul style="list-style-type: none"> • Démarrer Zelio Soft 2 • Choisir le module programmable • Créer un nouveau projet.
2	Editer le programme(en langage LADDER	<ul style="list-style-type: none"> • Insérer les entrées et les sorties nécessaires. • Insérer les différentes fonctions (timer, compteur,...).

3	Simulation du Programme	<ul style="list-style-type: none"> • On lance le simulateur • On force les entrées pour tester le fonctionnement du programme
4	Transfert du programme	<ul style="list-style-type: none"> • On installe le pilote du câble de communication • On fait le transfert du PC vers l'automate

III.5. Définition du Cahier de charge :

Manipulation 1 :

Lorsque les deux réservoirs A et B sont remplis (niveau max atteints : $H_A=H_B=1$) et en appuyant sur le bouton marche l'électrovanne A s'ouvre pour remplir le réservoir C et le voyant V1 s'allume, jusqu'à ce que le niveau min au niveau du réservoir A soit atteint ($L_A=1$), à ce moment l'électrovanne B s'ouvre pour remplir à son tour le réservoir C et le voyant V2 s'allume, jusqu'à ce que le niveau min du réservoir B soit atteint ($L_B=1$) et là, on lance une temporisation de 3 secondes avant d'actionner l'agitateur et allumer le voyant V4 pendant 5 secondes. Si

Manipulation 2 :

Le réservoir C n'étant pas vide, lorsque le réservoir A n'est pas rempli ($\overline{H_A}=1$) et en appuyant sur le bouton marche, la pompe démarre et le voyant V3 s'allume ($P=1$) jusqu'au remplissage du réservoir A ($H_A=1$), puis on ouvre l'électrovanne EVA et on allume le voyant V1 jusqu'à ce que le réservoir B soit rempli ou le réservoir A soit vide, on lance alors une temporisation de 3 secondes et on allume le voyant V4. Après les 3 secondes et si on appuie sur le bouton marche, l'électrovanne EVB s'ouvre et le voyant V2 s'allume jusqu'à ce que le réservoir B soit vide ($L_B=1$), on ce moment, on revient à l'étape initiale

Voici les grafjets qui décrivent le fonctionnement de la maquette :

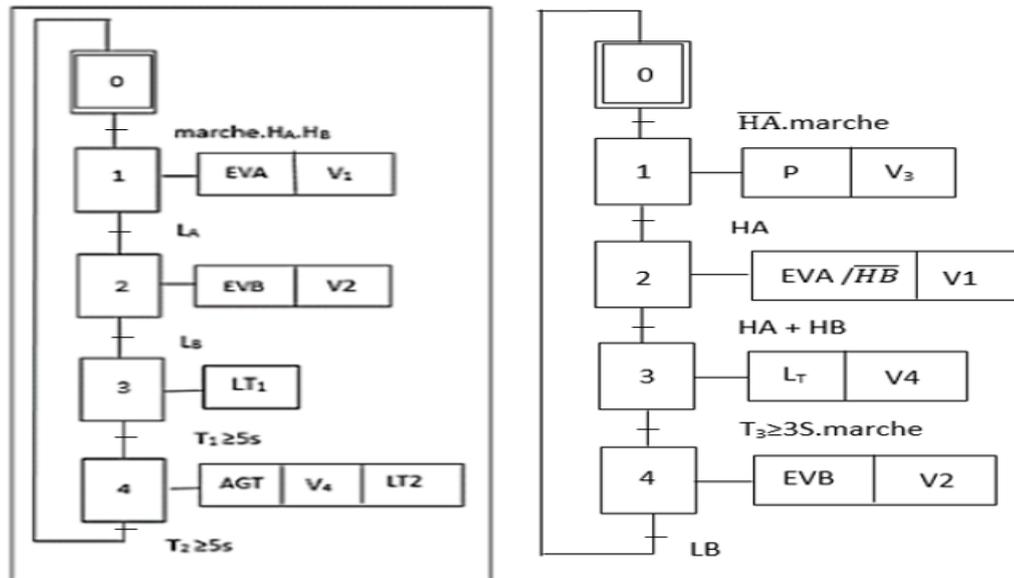


Figure III.2: les deux grafjets à réaliser sur la maquette

Manipulation supplémentaire

On lance une manipulation supplémentaire qui sert à remplir les 2 réservoirs A et B comme une préparation pour commencer la manipulation 1

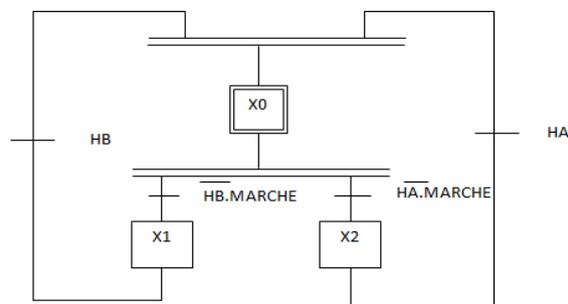


Figure III.3: Un grafjet supplémentaire à réaliser sur la maquette

III.6.Création du programme de gestion de la station de pompage

Le module programmable utilisé, et grâce à son afficheur LCD et ses boutons sur la face avant, offre la possibilité de le programmer directement sur l'écran sans passer par le logiciel Zelio Soft, mais seulement en langage LDDER

On a essayé dans un premier temps de programmer l'automate directement sur son écran, ce qui a été facile et le programme a bien fonctionné

On a passé ensuite à l'utilisation du logiciel Zelio Soft2, on a d'abord installé le logiciel sur un PC avec un système Windows XP, on a installé aussi le pilote du câble de communication.

Après l'édition du programme et la simulation, on l'a transféré vers l'automate, et on a testé les trois manipulations sur la maquette de station de pompage, qui a répondu correctement aux trois cahiers des charges.

III.7. Programmation sur écran

Cette programmation est permise uniquement en langage LADDER, l'automate étant en mode stop.

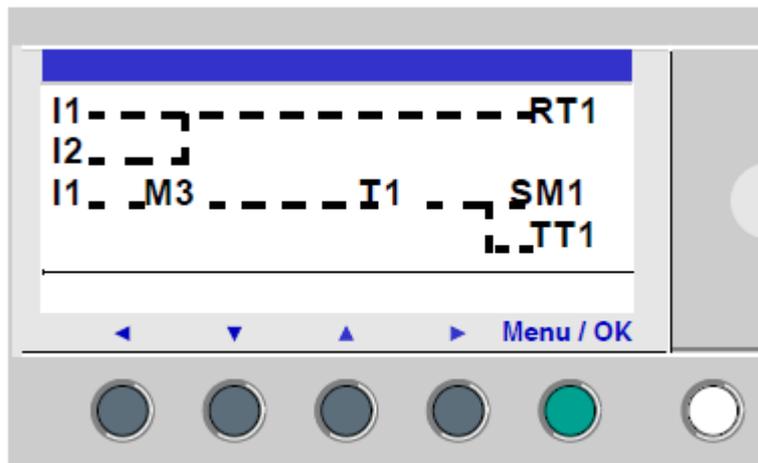
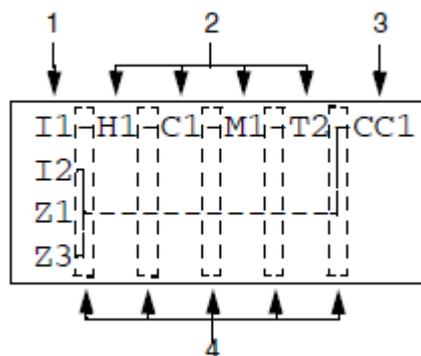


Figure III.3: programme sur l'afficheur de l'automate Zelio

Le module logique permet la saisie de 120 lignes de schémas de commande. L'écran du module logique permet de visualiser ces lignes 4 par 4 et se présente de la façon suivante :



Repère	Élément
1	Colonne réservée aux contacts (conditions).
2	Colonne réservée aux contacts (conditions) ou aux liaisons.
3	Colonne réservée aux bobines (actions).
4	Colonne réservée aux liaisons.

Conclusion Générale et Perspectives

Les travaux pratiques sont indispensables pour la formation des étudiants qui vont s'orienter soit vers le monde industriel ou bien vers les laboratoires de recherches, car dans les deux cas, il leur faut une base solide fondée sur la théorie et complétée par la manipulation des maquettes et des prototypes.

Ce travail est une petite contribution qui vise à permettre aux étudiants d'appliquer les connaissances acquises à partir des cours des Automates Programmables industriels pour les mieux assimiler.

On a proposé la gestion du prototype de station de pompage réalisé au sein de laboratoire de l'automatique par un automate Zelio, qui forme une unité de commande flexible, et qu'on a pu exploiter d'une manière optimale grâce au logiciel de programmation Zelio Soft 2.

La réalisation des cahiers de charges sur la maquette a passé par les étapes suivantes :

- Etablir les grafjets qui correspondent à chaque cahier de charge,
- Identifier les entrées et les sorties ainsi que les temporisateurs à utiliser,
- Créer un nouveau projet dans le logiciel Zelio Soft 2 en choisissant le module programmable dont on dispose au sein du laboratoire
- Elaborer les programmes sur le logiciel,
- Simuler les programmes avant le transfert vers l'automate.
- Transférer le programme et tester la maquette

En fin, nous espérons que ce projet soit utile pour les étudiants de l'automatique, il serait intéressant qu'ils puissent par la suite le développer, pour manipuler des grandeurs analogiques tels que le débit, et le niveau du liquide, et pour se faire nous proposons d'ajouter :

- Un débitmètre pour la mesure du débit du liquide
- Des capteurs de niveaux analogiques
- Un variateur de fréquence ou une vanne proportionnelle pour la régulation du débit et/ou niveau

Bibliographie

[1] documentation logiciel Zelio soft-CD d'installation

Catalogues

[2] modules logiques Zelio Logic, 2010

[3] Telecatalog - Automatismes & Contrôle, 2008

[4] catalogue-plate-forme d'automatisme Modicon Premium, 2012

[5] catalogue-plate-forme d'automatisme Modicon Quantum, 2012

Sit internet

[6] www.schneider-electric.com (consulté le 2015/04/29)

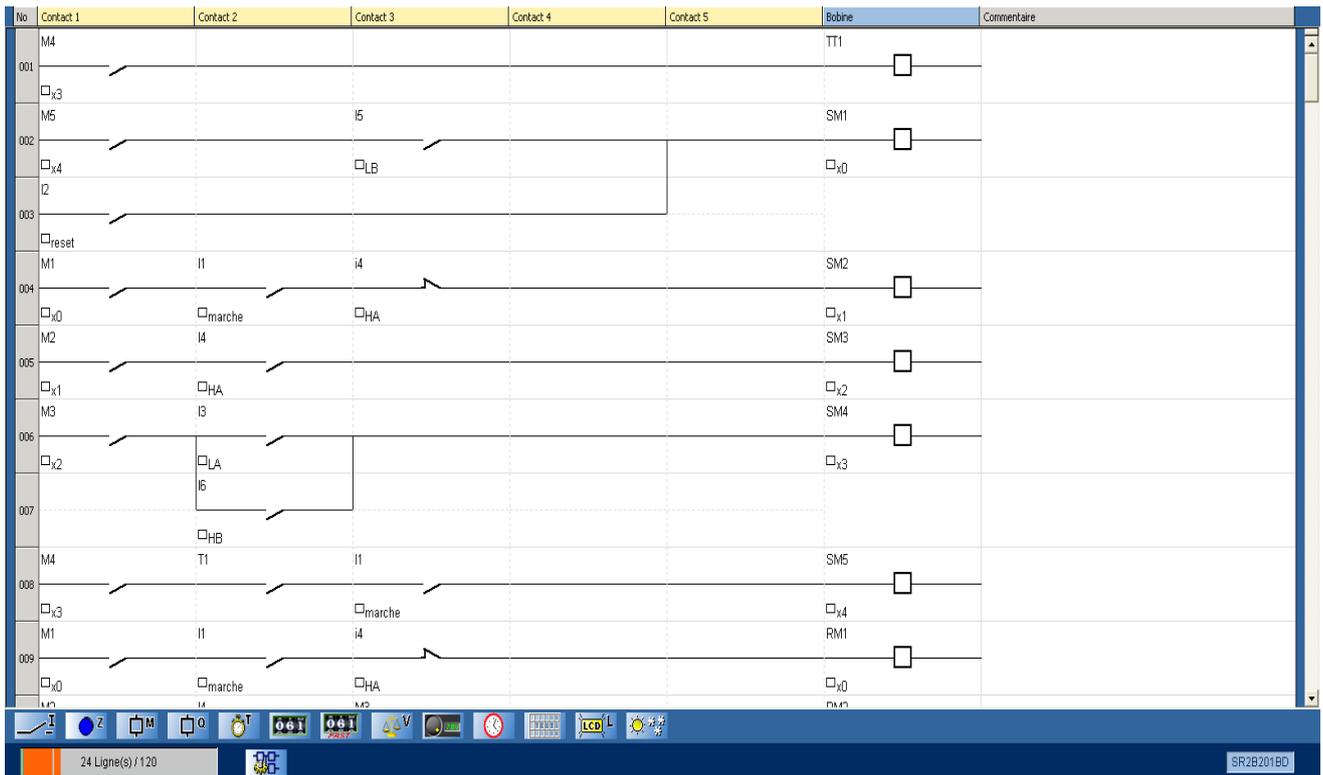
ANNEXE A : Les programmes des 3 manipulations

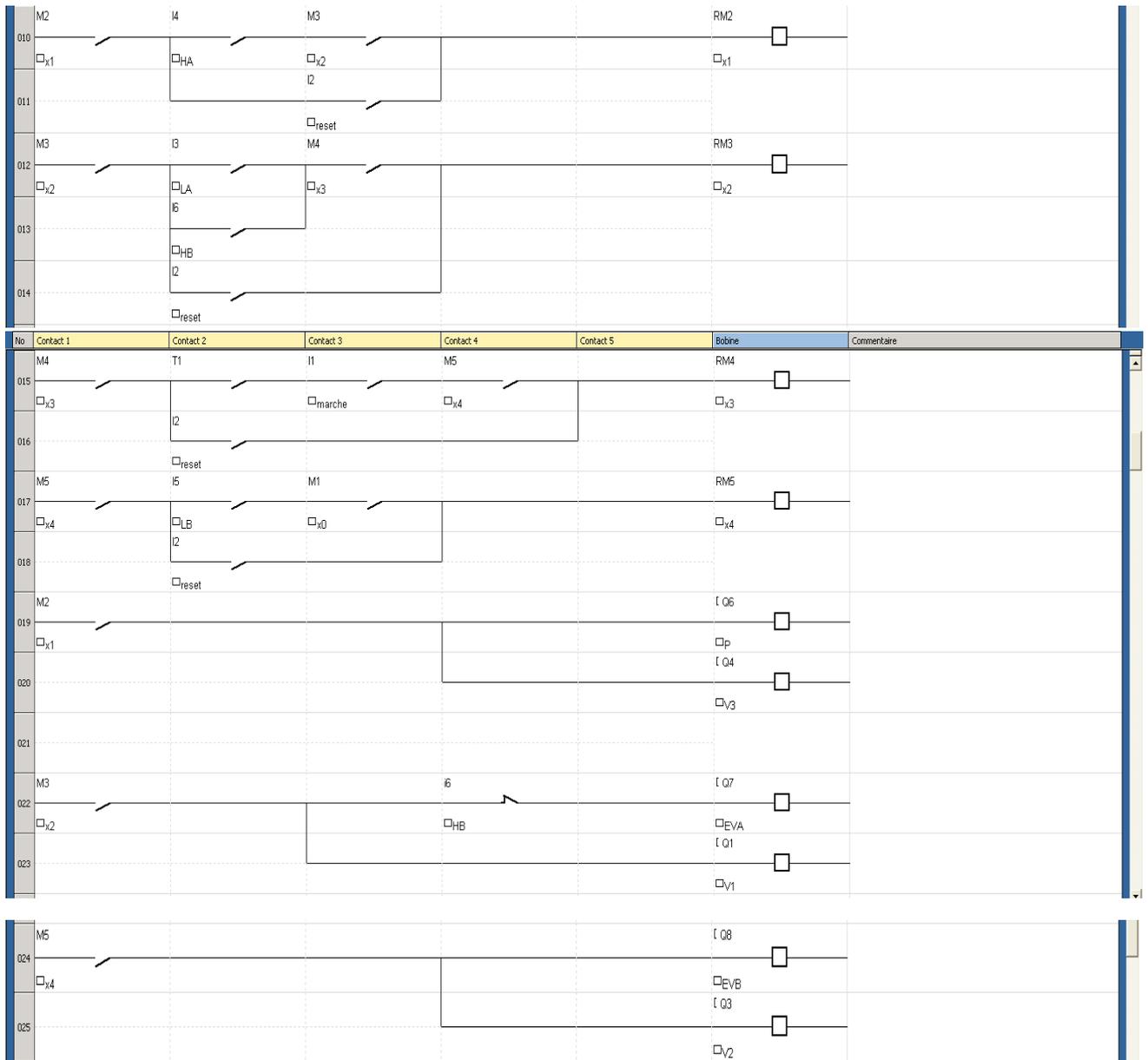
A.1. Programme de la 1^{ère} manipulation

A.2. Programme de la 2^{ème} manipulation

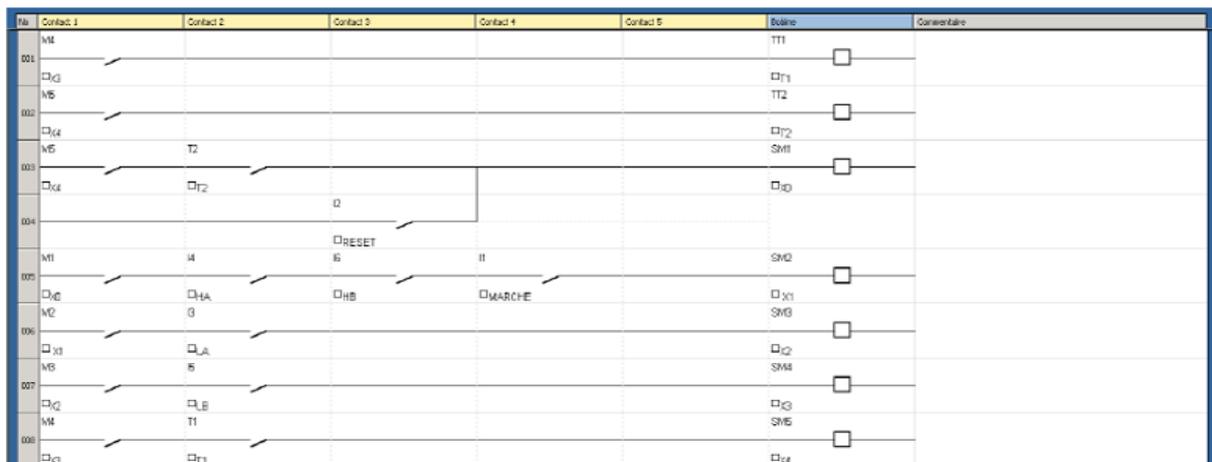
A.3. Programme de la manipulation supplémentaire

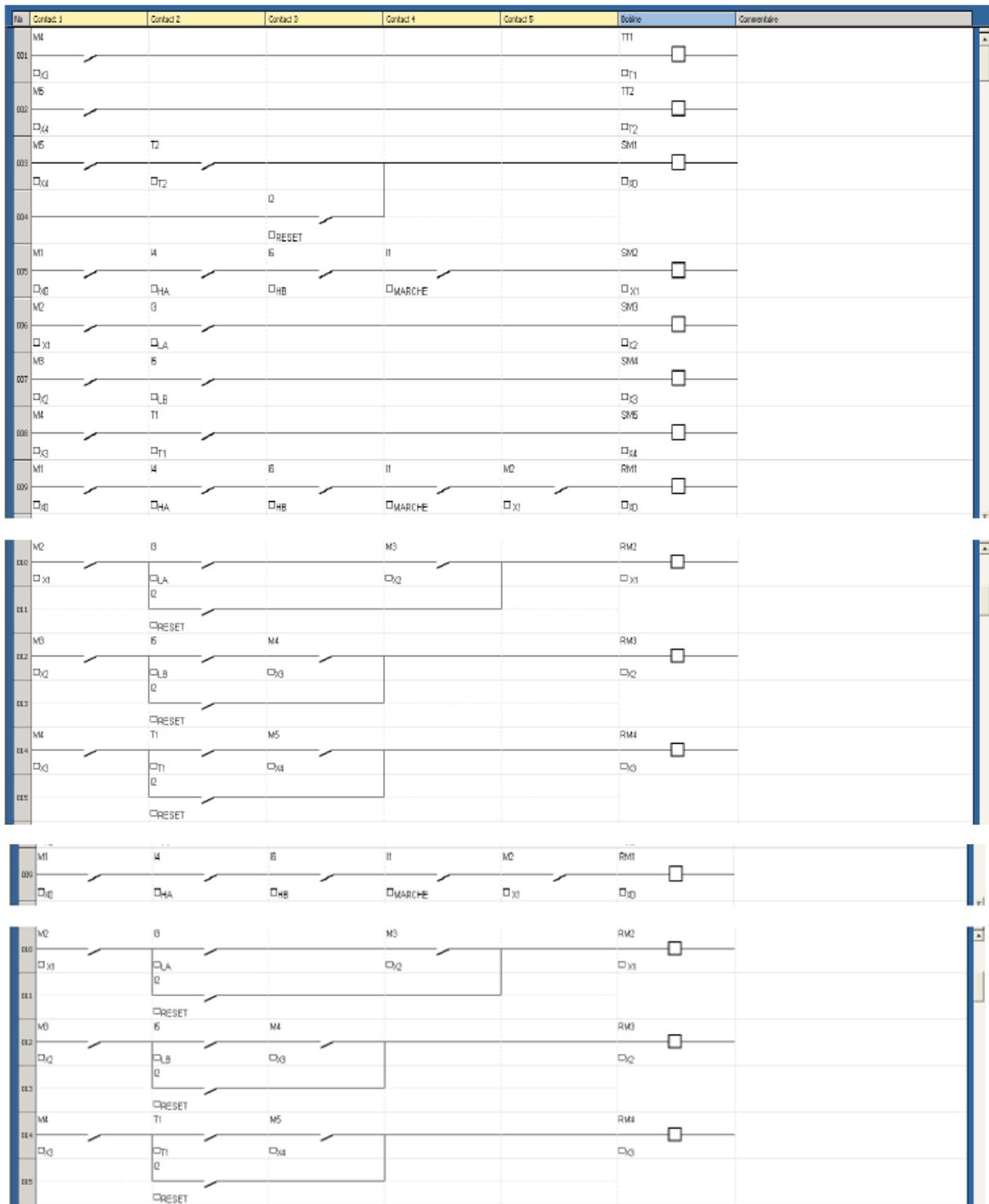
A.1. Programme de la 1^{ère} manipulation

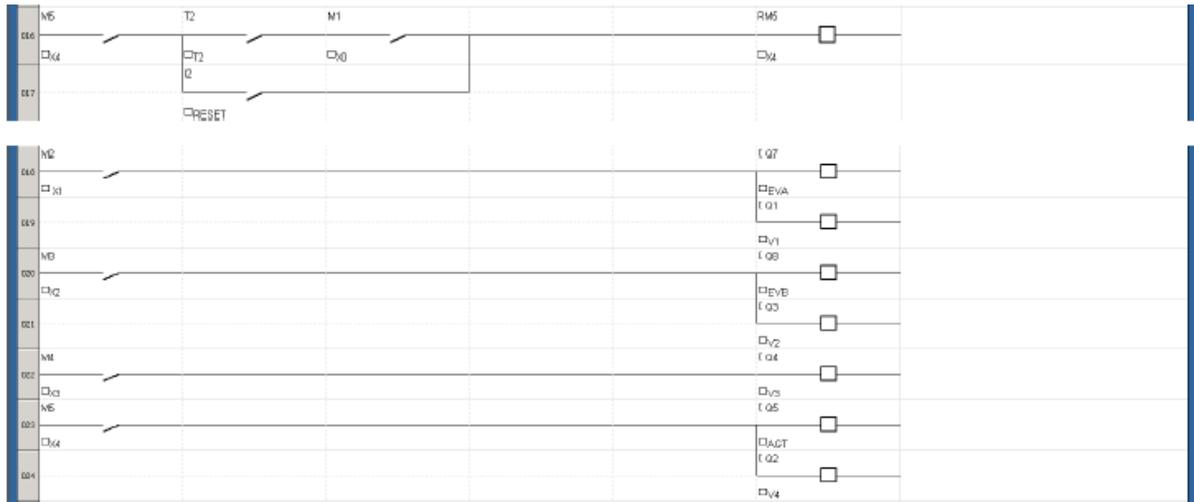




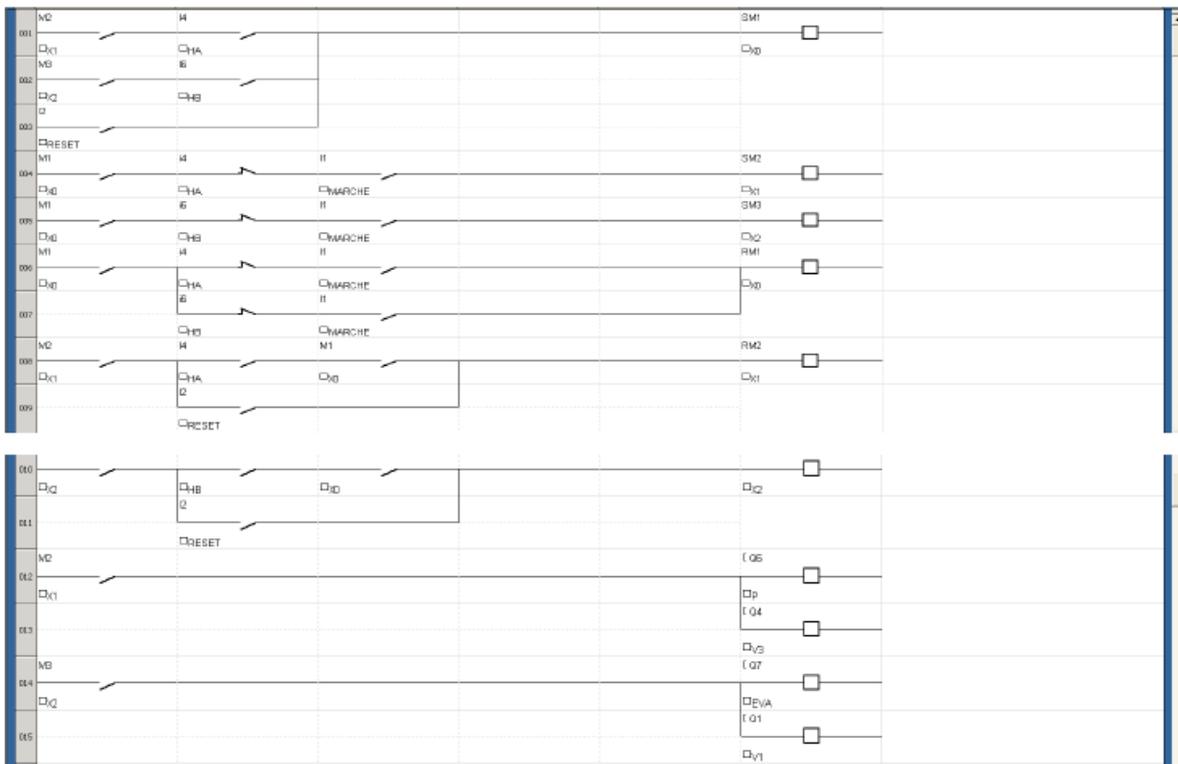
A.2. Programme de la 2ème manipulation







A.3. Programme de manipulation supplémentaire (remplissage des réservoirs A et B)



ANNEXE B : Edition du programme sur écran de l'automate Zelio

B.1 Méthode de saisie d'un contact ou d'une bobine

B.2. Saisie d'une liaison

B.3 .Saisie des paramètres des blocs fonctions

B.4. Suppression et insertion de lignes de schéma

B.1 Méthode de saisie d'un contact ou d'une bobine

Cette section décrit les procédures pour effectuer les opérations suivantes :

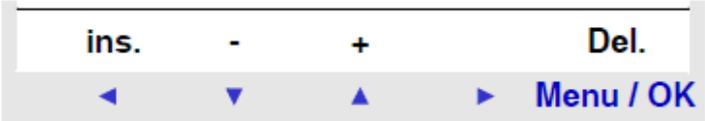
- saisie d'un élément,
- modification d'un élément,
- suppression d'un élément.

Ceci, quel que soit le type d'élément : contact ou bobine, qu'il soit paramétrable ou non.

➤ **Saisie d'un élément** : La saisie des éléments suit les règles suivantes :

- **contact** : sur toutes les colonnes à l'exception de la dernière,
- **bobine** : uniquement sur la dernière colonne.

La présence du curseur carré clignotant indique la possibilité d'insérer un élément.

Etape	Action
1	<p>Positionner le curseur clignotant à l'endroit désiré.</p> <p>Les touches de navigation permettent de déplacer le curseur dans le sens des flèches de navigation ◀ ▼ ▲ ▶.</p> <p>Illustration :</p> 
2	<p>Appuyer sur la touche Shift pour faire apparaître le menu contextuel.</p> <p>Illustration :</p>  <p>L'appui simultané sur Shift et sur l'une des touches ▼ ▲ (- et +) insère la première lettre de l'élément : I pour un contact et Q pour une bobine, suivi du chiffre 1.</p>
3	<p>Choisir le type d'élément désiré en appuyant simultanément sur Shift et + ou -. Ceci fait défiler de manière cyclique les différents types d'éléments possibles, dans l'ordre suivant :</p> <ul style="list-style-type: none"> ● Pour les contacts : I, i, Z, z, M, m, Q, q, T, t, C, c, K, k, V, v, A, a, H, h, W, w, S, s. ● Pour les bobines : M, Q, T, C, K, X, L, S. <p>Se référer au chapitre <i>Les éléments du langage LD</i>, p. 89.</p>

Etape	Action
4	<p>Relâcher la touche Shift pour avoir accès aux touches de navigations : ◀ ▼ ▲ ▶.</p> <p>L'appui sur la touche ▶ positionne le curseur sur le chiffre 1 correspondant.</p>
5	<p>Appuyer simultanément sur les touches Shift et + pour incrémenter le numéro de l'élément (2, 3, 4, ..., 9, A, etc.).</p> <p>Note : les numéros des blocs fonctions sont limités au nombre de blocs du type disponibles dans le module logique. Dans le cas des modules logiques extensibles, les numéros des entrées et des sorties permettent de programmer l'extension de taille maximum.</p> <p>Dans le cas de la saisie d'un contact, une fois cette étape réalisée, la saisie est terminée. Dans le cas de la saisie d'une bobine, il faut en plus choisir la fonction de la bobine.</p>
6	<p>Relâcher la touche Shift pour avoir accès aux touches de navigations : ◀ ▼ ▲ ▶.</p>
7	<p>Les étapes 7 à 9 ne sont à réaliser que dans le cas de la saisie d'une bobine.</p> <p>Positionner le curseur sur la fonction de la bobine en appuyant 2 fois sur la touche ◀.</p>
8	<p>Choisir la fonction désirée en appuyant simultanément sur Shift et + ou -, ce qui fait défiler les différentes fonctions possibles pour la bobine.</p>
9	<p>Relâcher la touche Shift pour avoir accès aux touches de navigations : ◀ ▼ ▲ ▶.</p>

Pour modifier un élément de schéma de commande existant, il suffit de :

- se positionner sur l'élément à modifier : étape 1 du tableau précédent,
- choisir le nouvel élément désiré : étapes 3 à 6.

Pour supprimer un élément, il suffit de :

- positionner le curseur sur l'élément à supprimer : étape 1,

- appuyer simultanément sur les touches Shift et Menu/OK.

Selon la position du curseur au moment de la suppression, deux cas peuvent se présenter :

- sur un élément : ce dernier est effacé
- sur un emplacement vide de la ligne : la ligne entière est effacée.

B.2.Saisie d'une liaison

Etape	Action
1	Positionner le curseur clignotant à l'endroit désiré. Les touches de navigation permettent de déplacer le curseur dans le sens des flèches de navigation ◀ ▼ ▲ ▶. Illustration : 
2	Appuyer sur la touche Shift pour faire apparaître le menu contextuel. Illustration : 
3	Tracer les connexions en appuyant simultanément sur Shift et sur une des touches : ← ↑ ↓ →. Shift et → trace une connexion jusqu'à la position du contact suivant ou jusqu'à la bobine de fin de ligne. Shift et ↑ ↓ permettent de tracer des connexions perpendiculaires vers la ligne précédente ou la suivante.
4	Relâcher la touche Shift pour avoir accès aux touches de navigations : ◀ ▼ ▲ ▶.
5	Répéter l'opération autant de fois que nécessaire pour relier les éléments les uns aux autres selon vos besoins.

Pour modifier une liaison existante, il suffit de :

- se positionner sur la liaison à modifier : étape 1
- modifier la liaison : étapes 2 à 5.

Pour supprimer une liaison, il suffit de :

- positionner le curseur sur l'élément à supprimer : étape 1.
- appuyer simultanément sur les touches **Shift** et **Menu/OK**.

Selon la position du curseur au moment de la suppression, deux cas peuvent se présenter :

- sur une liaison : cette dernière est effacée,
- sur un emplacement vide de la ligne : la ligne entière est effacée.

Pour remplacer une liaison par un contact, il suffit de :

- positionner le curseur (carré clignotant) sur la liaison à transformer : étape 1.
- suivre la procédure de saisie d'un élément (voir *Méthode de saisie d'un contact ou d'une bobine, p. 45*) : étapes 3 à 6.

B.3 .Saisie des paramètres des blocs fonctions

Lors de la saisie d'un schéma de commande, il faut renseigner les paramètres des fonctions d'automatisme configurables. Les fonctions d'automatisme possédant des paramètres sont :

- Relais auxiliaires,
- Sorties Tout ou Rien,
- Horloges,
- Compérateurs analogiques,
- Temporisateurs,
- Compteurs,
- Compteur rapide.

Quel que soit l'écran de paramétrage, la saisie des paramètres est identique :

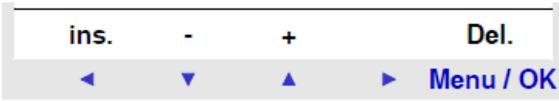
Etape	Action
1	Saisir la fonction d'automatisme désirée : étape 1 de la procédure de saisie d'un élément (voir <i>Méthode de saisie d'un contact ou d'une bobine, p. 45</i>). Lorsque la fonction possède des paramètres, Param apparaît dans le menu contextuel (lorsqu'on appuie sur la touche Shift). Illustration : 
2	Maintenir la touche Shift enfoncée et appuyer sur Param (touche ►). Résultat : l'écran de paramétrage de la fonction apparaît.
3	Se positionner sur les champs des paramètres modifiables à l'aide des touches de navigations : ◀ ▶.
4	Modifier la valeur du paramètre à l'aide des touches + et - avec la touche Shift enfoncée.
5	Valider les modifications en appuyant sur Menu/OK , ce qui ouvre la fenêtre de validation. Valider à nouveau en appuyant sur la touche Menu/OK pour sauvegarder.

Pour modifier les paramètres d'un élément existant, il suffit de :

Etape	Action
1	Se positionner sur l'élément à modifier à l'aide des touches de navigation : étape 1 de la procédure de saisie d'un élément (voir <i>Méthode de saisie d'un contact ou d'une bobine, p. 45</i>)
2	Appuyer simultanément sur les touches Shift et Param pour ouvrir la fenêtre de paramétrage.
3	Continuer avec les étapes 3 à 5 ci-dessus.

B.4. Suppression et insertion de lignes de schéma

La suppression des lignes de schéma s'effectue ligne par ligne. La procédure est la suivante :

Etape	Action
1	Positionnez le curseur sur la ligne à supprimer.
2	Supprimer tous les éléments de la ligne (voir <i>Méthode de saisie d'un contact ou d'une bobine, p. 45</i>) : (liaisons, contacts et bobines) pour obtenir une ligne vide.
3	Appuyer sur la touche Shift pour faire apparaître le menu contextuel. Illustration :  <p>L'appui simultané sur Shift et Del ouvre la fenêtre de confirmation.</p>
4	Valider en appuyant sur Menu/OK .

Pour l'insertion, la procédure est la suivante :

Etape	Action
1	Positionner le curseur sur la ligne située immédiatement au-dessous de la ligne à créer.
2	Appuyer sur la touche Shift pour faire apparaître le menu contextuel.
3	Appuyer sur Ins (en maintenant la touche Shift enfoncée) pour créer la ligne.