

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

Ecole Nationale Polytechnique



Département Automatique
Laboratoire de Commande des processus



Mémoire de Master en Automatique

Thème

Commande et supervision d'une station de pompage
didactique avec l'automate Zelio

Réalisé par : **Lyes Heythem BETTACHE**

Sous la direction de M. El Madjid Berkouk Professeur à l'ENP Alger

Présenté et soutenu publiquement le 29/06/2017

Composition du Jury :

Président :	L.ABDELOUL	Chargé de Cours à l'ENP
Rapporteur :	E.M.BERKOUK	Professeur à l'ENP
Examineur :	H.ACHOUR	Chargé de Cours à l'ENP

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

Ecole Nationale Polytechnique



Département Automatique
Laboratoire de Commande des processus



Mémoire de Master en Automatique

Thème

Commande et supervision d'une station de pompage
didactique avec l'automate Zelio

Réalisé par : **Lyes Heythem BETTACHE**

Sous la direction de M. El Madjid Berkouk Professeur à l'ENP Alger

Présenté et soutenu publiquement le 29/06/2017

Composition du Jury :

Président :	L.ABDELOUL	Chargé de Cours à l'ENP
Rapporteur :	E.M.BERKOUK	Professeur à l'ENP
Examineur :	H.ACHOUR	Chargé de Cours à l'ENP

ملخص

العنوان: التحكم والمراقبة في نموذج محطة ضخ المياه باستعمال المسير الصناعي المبرمج "زليو"
العمل المنجز في المذكرة يتمحور حول تحكم و مراقبة محطة ضخ تعليمية بواسطة المبرمج زليو, زليو سوفت
المحطة المنجزة تحتوي على عدة أدوات تسمح باختيار نمط التسيير :

- كلي (TOR) او نسبي (ANALOGIQUE)
- مراقبة مباشرة لعملية الضخ
- مراقبة و تسيير عن قرب أو عن بعد (باستعمال الشبكة المحلية)

الكلمات المفتاحية

مسير صناعي مبرمج , برنامج "ستابسات", محطة ضخ المياه.

Abstract :

The present work is mainly about the control and supervision of didactic pumping station using the programmable logic controller Zelio, ZelioSoft softwares.

The realized station has many tools that allow us to choose the management type :

- logic or analogic
- Real time visualization of the process
- Close or Remote visualization and control (local network)

Keywords:

Siemens PLC Zelio, ZelioSoft software, didactic pumping station, Arduino.

Résumé :

Ce mémoire présente essentiellement sur la commande et supervision d'une station de pompage didactique par le biais des automates programmables Zelio, Et le logiciel de programmation Zelio Soft

- La station réalisée a des outils qui permettent de choisir le type de fonctionnement :Digital or analogique
- Visualisation du procédé en temps réel
- Control et visualisation sur place ou à distance

Mots clés : Automate Zelio, ZelioSoft, station de pompage didactique.

Remerciements

Nous tenons à remercier dieu de nous avoir donné la force morale, physique et l'aide pour

accomplir ce modeste travail.

Nous tenons à remercier notre promoteur Pr. BERKOUK pour nous avoir acceptées

encadrées et dirigées durant l'élaboration de ce travail ainsi que pour son assistance et tous

ses conseils.

Nous remercions également nos professeurs H.ACHOUR, R.ILLOUL et O.STIHI, pour leurs

conseils et leurs aides.

Nous remercions chaleureusement les membres du jury pour l'honneur qu'ils nous ont fait en

acceptent d'évaluer notre projet.

Nous souhaitons aussi remercier tous les enseignants de l'Ecole Nationale Polytechnique d'Alger, et en particulier, Nos professeurs d'Automatique qui nous ont

encadrées auparavant et tous nos enseignants pour les connaissances qu'ils nous ont

transmis, leur disponibilité et leurs efforts.

Que tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste travail

trouvent ici l'expression de notre sincère gratitude.

Dédicace

Je dédie ce modeste travail tout d'abord à mon père qui m'a donné tous sans rien demandé en retour.

A ma chère mère sans elle je ne serais pas l'homme que je suis.

A mes sœurs

A mon petit frère

A mon ami et binôme Ahmed sur qui j'ai toujours pu compter.

A toute ma famille

A mes amis et camarades avec qui j'ai passé de bons moments.

A tous ceux qui ont contribué de loin ou de prêt à notre travail.

BETTACHE Lyes Heythem

Table des matières

Liste des tableaux

Liste des figures

Liste des abréviations

Introduction générale..... 10

Chapitre 1 : Les Modules programmables Zelio Logic et Le logiciel de programmation Zelio Soft

1.1 Introduction.....	12
1.2 Modules programmables Zelio Logic.....	12
1.2.1 Présentation.....	12
1.2.1.1 Pour l'industrie	12
1.2.1.2 Pour le tertiaire/bâtiment	12
1.2.2 Programmation.....	12
1.2.3 Modules logiques compacts.....	12
1.2.4 Modules logiques modulaires1	13
1.2.5 Communication.....	14
1.3 Le logiciel de programmation Zelio Soft	14
1.3.1 Présentation.....	14
1.3.2 Mode de fonctionnement.....	14
1.3.2.1 Mode saisie.....	14
1.3.2.2 Mode mise au point.....	14
1.3.3 La programmation en langage LADDER.....	14
1.3.3.1 Modes de saisie des schémas de commande.....	15
1.3.3.2 Fonctionnalités.....	15
1.3.3.3 Description d'un réseau de contacts.....	16
1.3.4 Programmation en langage FBD.....	16
1.3.4.1 Présentation.....	16
1.3.4.2 Fonctions en langage FBD.....	17
1.3.4.3 Blocs Fonctions FBD (Function Bloc Diagram).....	17
1.3.4.4 Fonctions logiques.....	19
1.3.4.5 Grafcet / SFC (Sequential Function Chart).....	20
1.3.5 Transfert d'une application.....	21
1.3.5.1 Ecrire d'un PC vers le Zelio Logic.....	21
1.3.5.2 Transfert du programme Zelio Logic vers le PC.....	22
1.3.5.3 Fonction mot de passe.....	22

1.3.6	Programmation sur écran.....	22
1.3.7	Verrouillage façade.....	24
1.4	Conclusion.....	24

Chapitre 2 : Commande et supervision du prototype de la station de pompage par l'automate Zelio

2.1	Introduction	26
2.2	Description de la station de pompage	27
2.3	Cahiers de charges et Grafjets	30
2.3.1	Grafjet et définition du cahier de charges de la manipulation TOR (logique)	31
2.3.2	Grafjet et définition du cahier de charges de la manipulation analogique	32
2.4	Implémentation sous Zelio Soft 2.....	33
2.5	Conclusion	34
	Conclusion générale et perspectives	35
	Bibliographie,webographie.....	36

Liste des tableaux

Tableau 1.1	page 15
Tableau 1.2	page 20
Tableau 1.3	page 21
Tableau 1.4	page 23
Tableau 2.1	page 29

Liste des figures

Figure 1.1	13
Figure 1.2	13
Figure 1.3	16
Figure 1.4	16
Figure 1.5	23
Figure 2.1	26
Figure 2.2	27
Figure 2.3	30
Figure 2.4	31
Figure 2.5	32
Figure 2.6	33
Figure 2.7	34

Symboles et abréviation

AA sortie analogique

AC Alternative Current

AE entrée analogique

AI Analog Input

AO Analog Output

API Automate Programmable Industriel

ASCII American Standard Code for Information Interchange

CHAR character

COUNT Schéma à Contact

CP Communication Processor

CPU Central Processing Unit

DA Sortie Digitale

DB Data Bloc

DC Direct Current (courant continu)

DE entrée digitale

DI Digital Input

DO Digital Output

DP Decentralized Periphery

E/S Entrée/Sortie

FB Fonctionnel Bloc

FC Fonction

FM Function Module

HW Config HardWare Configuration

IHM Interface Homme-Machine

HMI Human-Machine Interface

IM Coupleurs d'extension

INT INTeger

LED Light Emitting Diode

LIST Le langage de liste d'instructions

LOG LOGigramme

MPI Multi-Point Interface

OB Organisationnel Bloc

OP Operator Pupiter

PC Personnel Computer

PCMCIA Personal Computer Memory Card International Association

PG la console de programmation sur le terrain

PLC Programmable Logic Controller

PROFIBUS Process Field Bus

PROM Programmable Read Only Memory

EEPROM Electrically Ersable Programmable Read Only Memory

PS Gamme des alimentations stabilisees de Siemens

RAM Random Access Memory

ROM Read Only Memory

SFB System Fonctionnel Bloc

SFC System Fonction

SIMATIC Siemens Automatic

SM Signal Module

S7 Step 7

TOR Tout Ou Rien

Introduction générale :

De nos jours, l'automatisme occupe une place importante dans les processus industriels. Cette technologie revêt désormais une importance primordiale.

A cela s'ajoute que les nouvelles solutions industrielles, telles la décentralisation et la visualisation, exigent de nouveaux systèmes didactiques pour leur enseignement.

Dans le but de former ses élèves ingénieurs automaticiens à la maîtrise des automates programmables, le Département d'Automatique de l'ENP a proposé le projet d'une station de pompage didactique, où la mise en marche et le développement de cette dernière permet aux élèves ingénieurs de se familiariser avec les systèmes industrielles, comme les API, et interface HMI.

Dans ce travail, on propose de gérer le prototype de la station de pompage réalisé au sein du Laboratoire de Commande des Processus avec un automate programmable Zelio de Schneider.

Le présent rapport est organisé en deux chapitres décrivant les volets principaux de notre projet :

Le premier chapitre sera consacré à la description des automates programmables de la filiale Télémécanique du Schneider, particulièrement les modules programmables Zelio Logic. Aussi, le logiciel de programmation Zelio Soft 2, avec les langages de programmations qu'il admet et les différentes fonctions qu'il offre, sera développé dans ce chapitre.

Le deuxième chapitre présentera les différentes réalisations effectuées tout au long du projet.

Chapitre 1

Les Modules programmables Zelio Logic et Le logiciel de programmation Zelio Soft

1.1 Introduction

Un Automate Programmable Industriel (API, en anglais PLC pour Programmable Logic Controller) est une machine électronique programmable par un personnel non informaticien et destiné à piloter en environnement industriel et en temps réel des procédés industriels. Particulièrement, les automates offerts par la filiale Télémécanique du Schneider Electric, exactement (Modules programmables Zelio Logic) sont programmables par le biais du logiciel Zelio Soft. Il permet la programmation et la simulation de procédures automatisées avec différents langages normalisés. Dans ce chapitre, on va présenter le Module programmable Zelio Logic et Le logiciel de programmation Zelio Soft.

1.2 Modules programmables Zelio Logic [1]

1.2.1 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.

1.2.1.1 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 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,...).

1.2.1.1 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.

1.2.2 Programmation

La simplicité de leur programmation, garantie par l'universalité des langages, satisfait aux exigences de l'automatisme 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).

1.2.3 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 1.1 : Module Zelio Logic compact [1]

1.2.4 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.

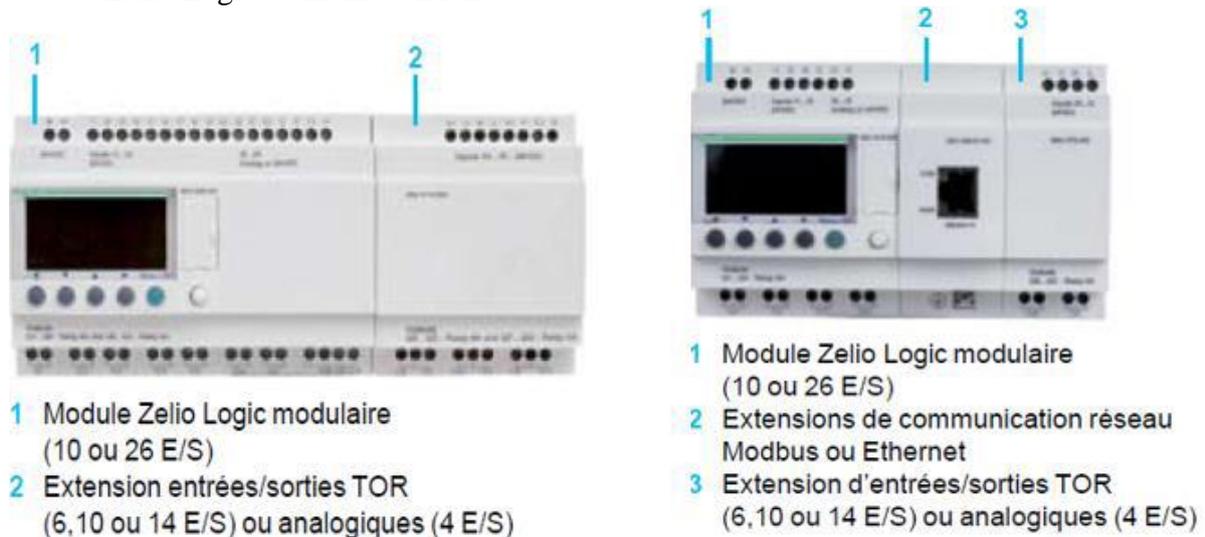


Figure 1.2 : Association entre modules logiques modulaires et extensions [1]

1.2.5 Communication [1]

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

1.3 Le logiciel de programmation Zelio Soft [1]

1.3.1 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.

1.3.2 Mode de fonctionnement

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

1.3.2.1 Mode saisie

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

1.3.2.2 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.

1.3.3 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]

1.3.3.1 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.

Fonctions			
Fonction	Schéma électrique	Langage LADDER	Commentaire
Contact			<p>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.</p>
Bobine classique			<p>La bobine est excitée lorsque les contacts auxquels elle est reliée sont passants.</p>
Bobine à accrochage (Set)			<p>La bobine est excitée lorsque les contacts auxquels elle est reliée sont passants. Elle reste enclenchée lorsque les contacts ne sont plus passants.</p>
Bobine de décrochage (Reset)			<p>La bobine est désexcitée lorsque les contacts auxquels elle est reliée sont passants. Elle reste inactivée lorsque les contacts ne sont plus passants.</p>

Tableau 1.1 : Fonctions en symboles LADDER et symboles électriques [1]

1.3.3.2 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.

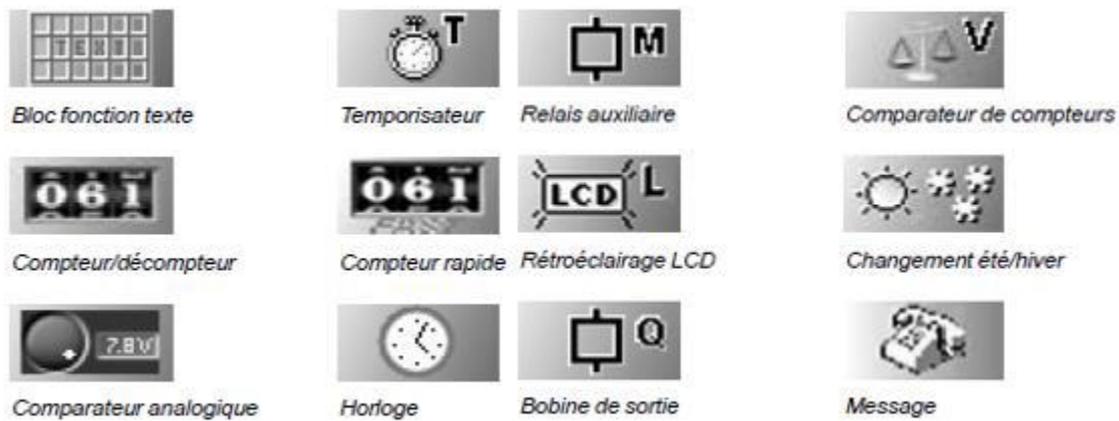


Figure 1.3 : Différents outils de programmation en langage LADDER [1]

1.3.3.3 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 reparti 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.

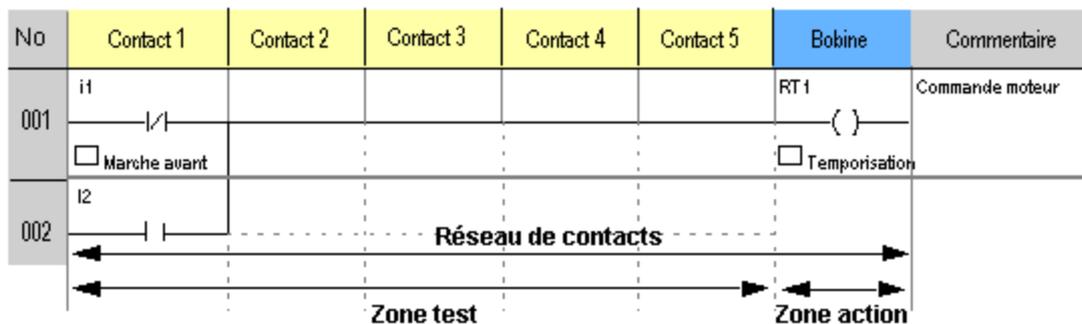


Figure1.4 : Structure d'un réseau de contacts [2]

1.3.4 programmation en langage FBD [2]

1.3.4.1 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.

1.3.4.2 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 MARCHE 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.

1.3.4.3 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** TIMER 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** TIMER LI

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** BOOLEAN

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** CAM

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.



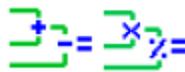
- **La fonction multiplexeur**

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**

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.

1.3.4.4 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

Tableau1.2 : Fonctions logiques [2]

1.3.4.5 Grafset / SFC (Sequential Function Chart)

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

Le Grafset 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 :

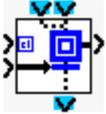
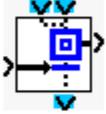
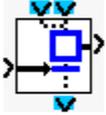
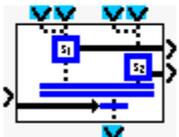
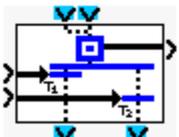
Désignation	Symbole	Description
<u>Etape Initiale</u>		Etape initiale d'un graphe SFC.
<u>Etape Initiale réinitialisable</u>		Etape initiale d'un graphe SFC avec initialisation de l'étape par une commande. Initialise la totalité du graphe connexe qui contient le reset init.
<u>Etape</u>		Etape qui transmet 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 1.3 : Fonctions FC [2]

1.3.5 Transfert d'une application

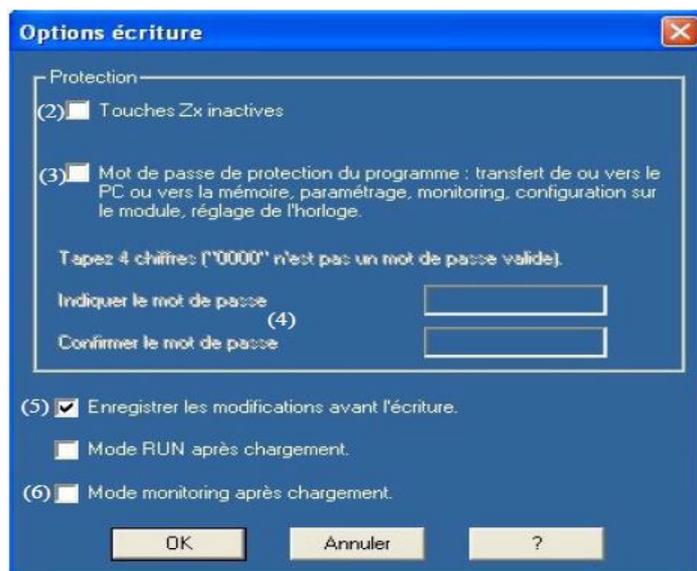
1.3.5.1 Ecrire d'un PC vers le Zelio Logic

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.

1.3.5.2 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.

1.3.5.3 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.

1.3.6 Programmation sur écran [3]

Le module Zelio, 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.

1.3.7 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 ».

1.3 Conclusion

De part la mobilité, la flexibilité de son architecture, la facilité de sa programmation, de sa connexion et de son adaptation dans les milieux industriels, l'automate programmable est devenu un produit incontournable dans les systèmes automatisés de production.

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

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.

Chapitre 2

Commande et supervision du prototype de la station de pompage par l'automate Zelio

2.2 Introduction

Au sein du Laboratoire de Commande des Processus, on a remis en marche une maquette de station de pompage dans le but de réaliser des applications et manipulations plus avancées, la gestion de cette maquette est effectuée à la base par un automate programmable Siemens.

Dans ce chapitre on va commander cette station en utilisant un automate Zelio de Schneider. Cet automate est programmé en utilisant le logiciel ZelioSoft. Deux cahiers Charges sont développés pour cette station.

L'organigramme suivant montre les étapes suivies dans la réalisation de notre projet :

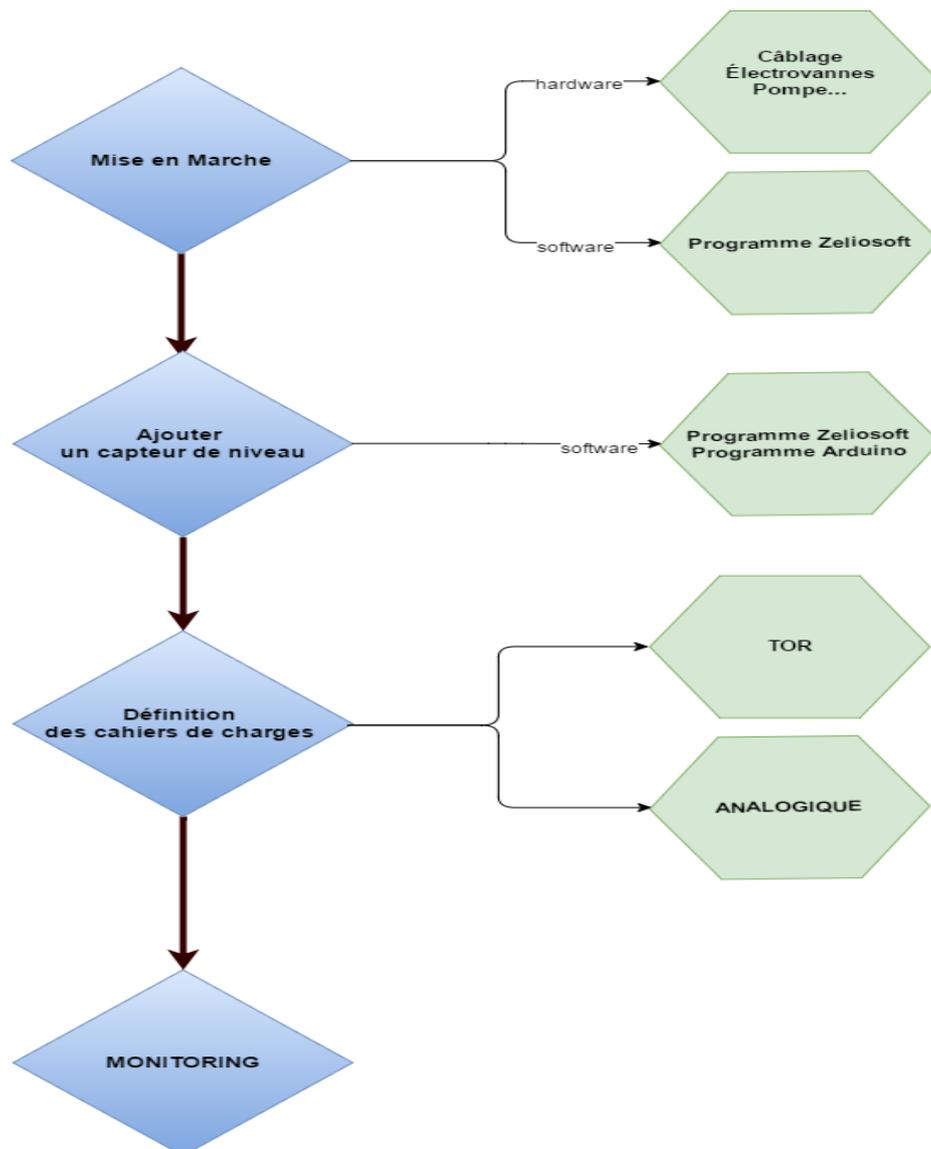


Figure 2.1 : Organigramme des étapes de réalisation

2.2 Description de la station de pompage

La station de pompage est un dispositif éducatif destiné à l'étude et le contrôle de plusieurs systèmes. Le contrôle de ces systèmes se fait par le biais des automates Zelio.

Notre prototype a été basé sur le système contrôle de niveau. Les éléments et composants utilisés pour la construction de notre prototype, ont été choisis en fonction des paramètres, disponibilité et coût.

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

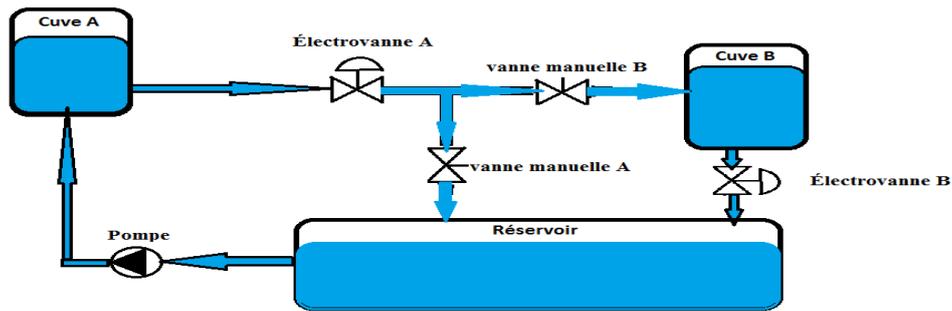


Figure 2.2 : représentation Synoptique de la station

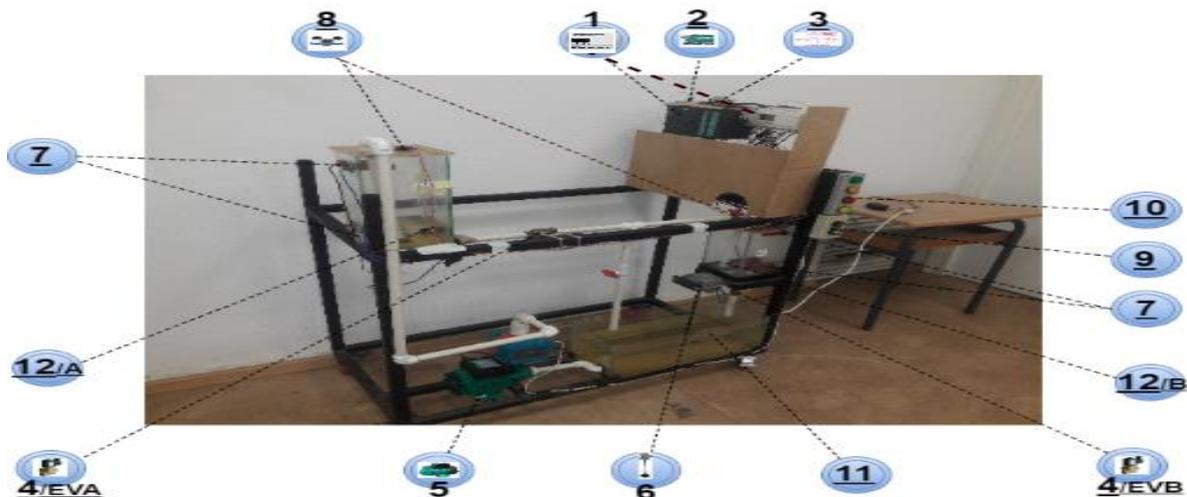


Figure 2.2 : Station de pompage réalisée

1-Automate (API/PLC)	7-Capteur de niveau logique (TOR)
2-Arduino	8-Capteur de niveau (Ultrason hc-05)
3-Circuit convertisseur PMW → Analogique	9- Boutons
4-Électrovannes (EVA, EVB)	10-Voyants
5-Pompe	11-Réservoir (C)
6-Agitateur	12- Cuves A et B

Figure 2.2 : représentation Synoptique de la station

L'ensemble des éléments et les caractéristiques du prototype réalisé :

Objet	Description
support métallique	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.
Réservoir	Un réservoir (C) de 0.50m sur 0.20m sur une hauteur de 0.20m
Cuves	Deux Cuves (A et B) de 0.15m sur 0.15m sur une hauteur de 0.30m/0.25m.
Pompe	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.
Agitateur	Un mélangeur(Agitateur) alimenté en 220 v pour agiter le liquide.
Démarrreur	Un démarreur progressif ATS01N03FT qui améliore les performances de démarrage du mélangeur en permettant un démarrage progressif et contrôlé.
Relais	Quatre relais de niveau JYB-714B alimenté en 220V et muni chacune de 2 bougies de sonde pour détecter les niveaux haut et les niveaux bas des réservoirs A et B. Deux relais électromagnétiques 8 broches servant comme interface de sortie entre l'automate et les actionneurs (pompe et l'agitateur).
robinets d'arrêt à opercule	Les deux robinets d'arrêt à opercule, utilisés pour permettre la flexibilité du

	prototype afin qu'on peut l'utiliser dans deux manipulations différentes (ou plus).
Alimentation	Une alimentation stabilisée (Harvest electric HAS-120-24) alimenté en monophasé 220 V et qui délivre du 24V.
Electrovannes	Deux électrovannes DN40 TOR alimentées en 24V DC.
Disjoncteur	Un disjoncteur pour la sécurité de toute l'installation.
Répartiteur	Deux répartiteurs de tension pour le 24 V et 220 V
Arduino	Une carte de développement qui réalise des calculs pour mesurer le niveau
Automate (API/PLC)	Un automate Zelio de Schneider Electric (SR2 B201BD) qui exécute les cahiers de charges et qui contrôle la station
Ultrason	Deux capteurs ultrason hc-05 sont utilisés pour mesurer le niveau
Circuit convertisseur	Deux circuits qui reçoit un signal PWM et délivre un signal analogique
Variateur de vitesse	un variateur de vitesse afin de la faire fonctionner de manière analogique pour contrôle le niveau
PC	Une station PC pour la supervision.

Tableau 2.1 : Eléments et composants de la station

La figure 2.3 donne une vision générale sur notre projet :

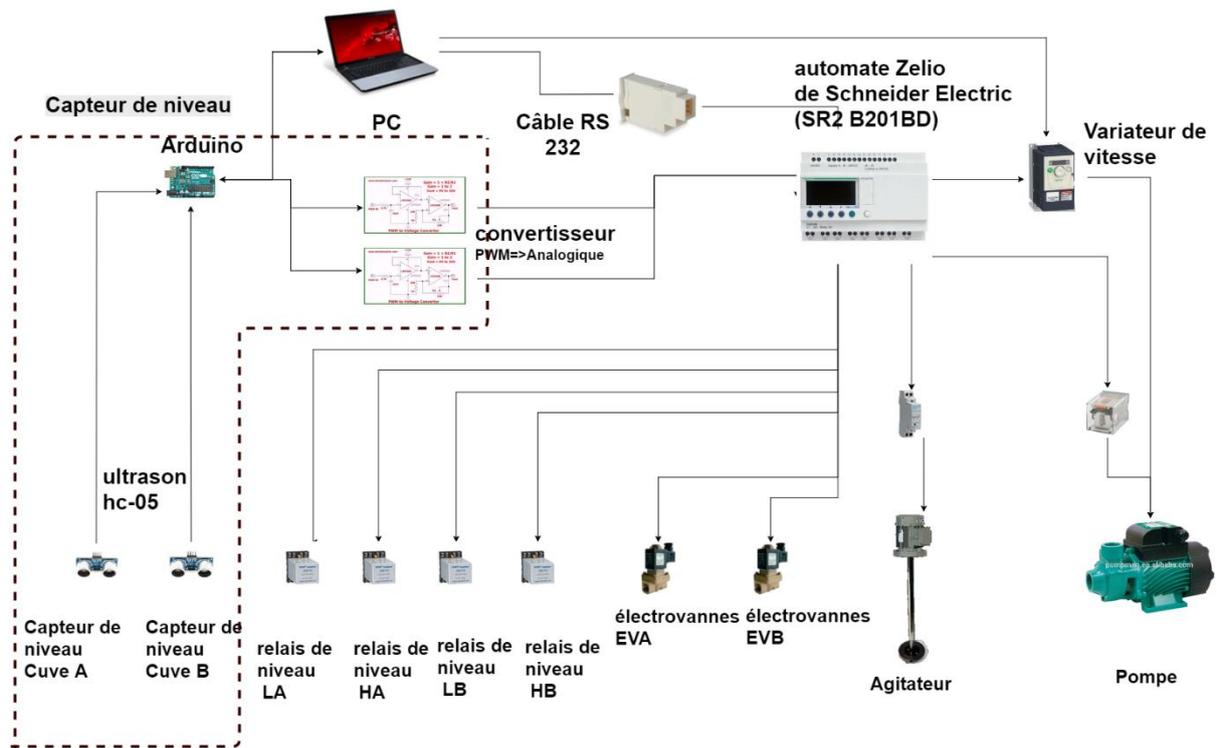


Figure 2.3 : Synoptique du prototype.

2.6 Cahiers de charges et Grafquets

Dans cette partie, on présente les grafquets et les manipulations réalisées sur notre station de pompage.

2.3.1 Grafquet et définition du cahier de charges de la manipulation TOR (logique) :

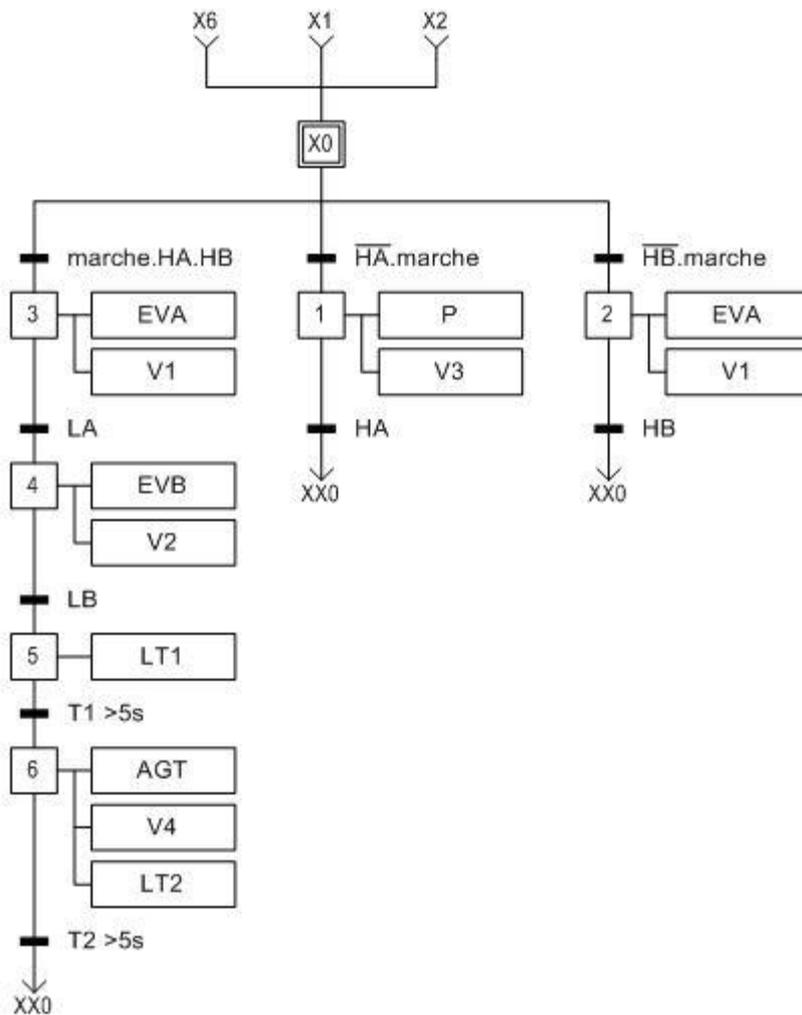


Figure 2.4 : Grafcet représentant la manipulation logique

La station étant à l'arrêt et en appuyant sur le bouton « marche », trois cas se présentent :

- Le niveau haut de la cuve A n'est pas atteint :
La pompe démarre pour ramener de l'eau du réservoir à la cuve A et le voyant V3 s'allume jusqu'à ce que le niveau haut de cette cuve soit atteint.
- Le niveau haut de la cuve B n'est pas atteint :
L'électrovanne EVA s'ouvre pour ramener de l'eau de la cuve A à la cuve B et le voyant V1 s'allume jusqu'à ce que le niveau haut de B soit atteint.
- Les niveaux hauts des cuves A et B sont atteints :
L'électrovanne EVA s'ouvre et l'eau émanant de la cuve A est versé dans le réservoir par le biais du robinet, une fois cette cuve est vidée, la même procédure est exécutée avec l'électrovanne EVB et la cuve B. une fois les deux cuves sont vides, il y a une temporisation de 5 secondes suivie du lancement de l'agitateur pour la même durée.
Les voyants V1 et V2 indiquent, respectivement, que les électrovannes EVA et EVB sont ouvertes tandis que le voyant V4 indique l'activation de l'agitateur.

2.3.2 Grafset et définition du cahier de charges de la manipulation analogique :

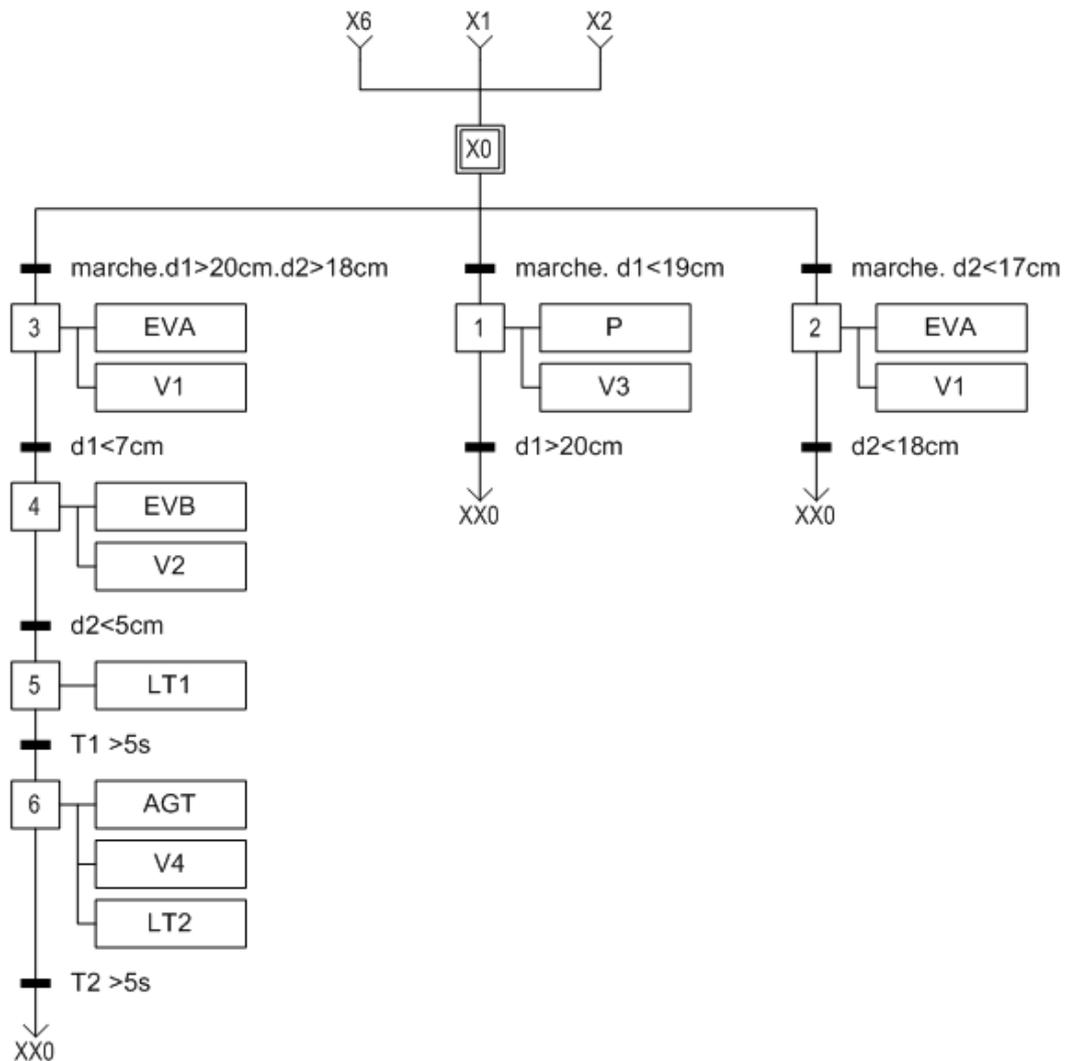


Figure 2.5 : Grafset représentant la manipulation analogique

Les symboles $d1$ et $d2$ représentent, respectivement, le niveau d'eau dans les cuve A et B.

La station étant à l'arrêt et en appuyant sur le bouton « marche », trois cas se présentent :

- Si $d1$ est inférieur à 19 cm :
La pompe démarre pour ramener de l'eau du réservoir à la cuve A et le voyant V3 s'allume jusqu'à ce que $d1$ atteigne 20 cm.
- Si $d2$ est inférieur à 17 cm :
L'électrovanne EVA s'ouvre pour ramener de l'eau de la cuve A à la cuve B et le voyant V1 s'allume jusqu'à ce que $d2$ atteigne 18 cm.
- Si $d1$ est supérieur à 20 cm et $d2$ est supérieur à 18 cm :
L'électrovanne EVA s'ouvre et l'eau émanant de la cuve A est versé dans le réservoir par le biais du robinet, une fois $d1$ atteint 7 cm, la même procédure est exécutée avec l'électrovanne EVB et la cuve B jusqu'à ce que $d2$ atteigne 5 cm. Après, il y a une temporisation de 5 secondes suivie du lancement de l'agitateur pour la même durée. Les voyants V1 et V2 indiquent, respectivement, que les électrovannes EVA et EVB sont ouvertes tandis que le voyant V4 indique l'activation de l'agitateur.

2.7 Implémentation sous Zelio Soft 2 :

On a essayé dans un premier temps de programmer l'automate directement sur son écran, ce qui a été facile et le programma 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 7, on a installé aussi le pilote du câble de communication.

Après l'édition du programme (SFC) et la simulation, on l'a transféré vers l'automate, et on a testé les deux manipulations (TOR/Analogique) sur la maquette de station de pompage, qui a répondu correctement aux deux cahiers des charges.

Voici les programmes des 2 manipulations en langage SFC :

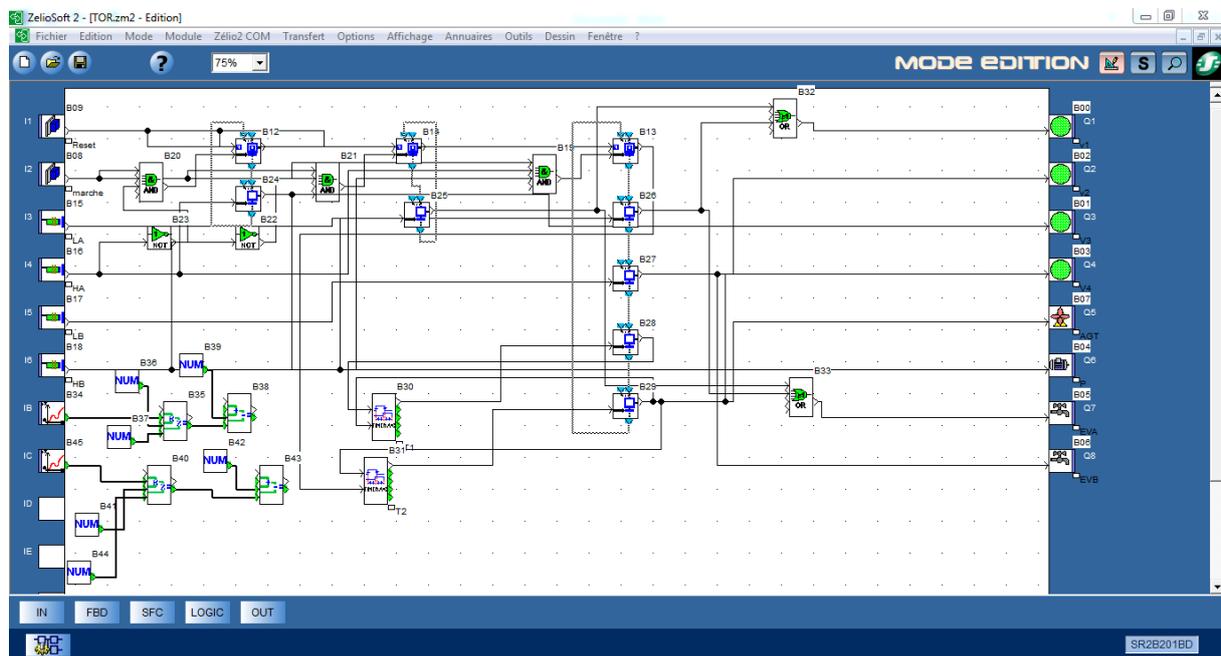


Figure 2.6 : Manipulation avec seulement des entrées TOR

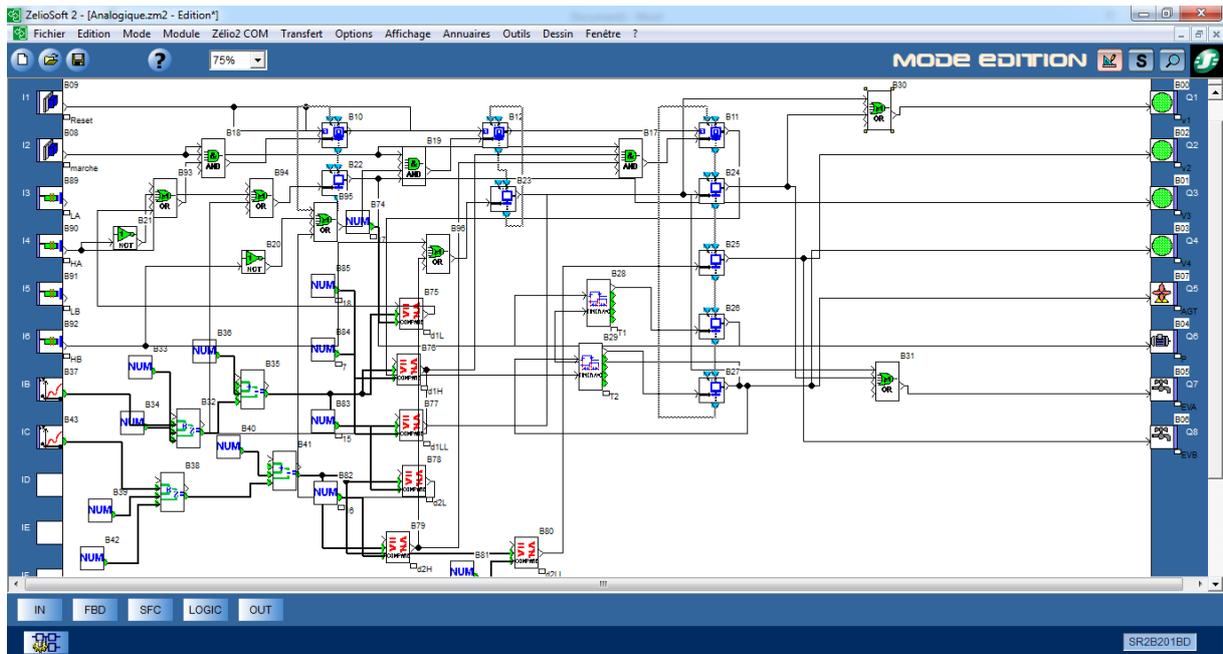


Figure 2.7 : Manipulation avec des entrées TOR et Analogiques

Pour la visualisation de l'état de la station, on a utilisé le mode MONITORING de Zelio Soft2

2.8 Conclusion

A la fin de ce chapitre, on peut conclure que la gestion de la maquette de la station de pompage par un automate Zelio de Schneider Electric est simple et efficace, et cela grâce au logiciel Zelio Soft.

Conclusion générale et perspectives

Un élève ingénieur ne doit pas se satisfaire d'acquérir des connaissances théoriques, il est plutôt primordiale qu'il complète sa formation par une bonne expérience pratique pour mettre en œuvre, justement, ce qu'il a appris en théorie. Il doit aussi gérer le budget et les moyens disponibles et ne doit pas lâcher devant les problèmes rencontrés.

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.

Enfin, nous espérons que notre travail sera utile aux étudiants en automatique, pour appliquer ce qu'ils apprennent aux cours de l'informatique industrielle et d'Automatique avancé, et qu'il participe à forger des ingénieurs automaticiens avec beaucoup d'expérience qui leur permet de transiter vers une nouvelle phase où ils seront obligés à manipuler des systèmes réels.

Perspectives

La maquette réalisée en mode logique et analogique, Il serait intéressant de réguler des grandeurs analogiques (niveau, débit, pression, ...) et d'implémenter différents lois de commandes pour comprendre la différence entre ces dernières et assimiler les avantages de chacune.

On peut ajouter par exemple :

- Un variateur de vitesse monophasé pour réguler le niveau
- Un débitmètre pour mesurer le débit et une vanne proportionnelle pour le réguler.
- Capteur de pression
- La supervision « Vijeo Citect »

Bibliographie :

Bibliographie

[1] Schneider-Electric, Home, [en ligne], [consulté le 27 juin 2017]. Disponible sur
« http://www.schneiderelectric.es/html/Zelio2/files/zeliologic_compact_modulaire.pdf »

[2] Schneider-Electric, documentation logiciel Zelio soft-CD d'installation

[3] B.FERHAOUI. Gestion D'UNE MAQUETTE DE STATION DE POMPAGE PAR UN AUTOMATE PROGRAMMABLE ZELIO. 57 pages. Mémoire de master en automatique : Alger, Ecole Nationale Polytechnique : 2015.