

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique



Département Automatique



Laboratoire de Commande des processus

Mémoire de fin d'études pour l'obtention

du diplôme d'ingénieur d'état

en Automatique

**Commande et supervision d'une station de pompage à
base d'automates programmables
(Automates Schneider et Siemens)**

Amel HAMDY CHERIF

Imane MAYOUF

Sous la direction de M. El Madjid BERKOUK professeur à l'ENP Alger

Présenté et soutenu publiquement le 23/06/2018

Composition du Jury :

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

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique



Département Automatique



Laboratoire de Commande des processus

Mémoire de fin d'études pour l'obtention

du diplôme d'ingénieur d'état

en Automatique

**Commande et supervision d'une station de pompage à
base d'automates programmables
(Automates Schneider et Siemens)**

Amel HAMDY CHERIF

Imane MAYOUF

Sous la direction de M. El Madjid BERKOUK professeur à l'ENP Alger

Présenté et soutenu publiquement le 23/06/2018

Composition du Jury :

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

تلخيص

المذكرة هي مساهمة في تحقيق وحدات تحكم محطة ضخ المياه بمسيرات صناعية مبرمجة. تتم إدارة المحطة عن طريق PLCs مختلفة (سيمنز وشنايدر). تتكون المحطة من خزانين مدعومين بمضخة من خلال صمامات الملف اللولبي. أيضا يتم استخدام المحرض المرتبطة مع الباء التصاعدي. يمكن استخدام مواصفات مختلفة للتحكم في المحطة.

الكلمات المفتاحية

مسير صناعي مبرمج "سيمنس", برنامج "سنايسات", برنامج "وينسيبي فلكسيبل", مسير صناعي مبرمج "زليو", برنامج "زليو سوفت", مسير صناعي مبرمج "لوقو", برنامج "لوقو! سوفت كونفور", نموذج محطة ضخ المياه.

ABSTRACT

The memory is a contribution to the realization of a manipulation based programmable controllers. The station is managed by different PLCs (Siemens and Schneider).

The station consists of two tanks powered by a pump through solenoid valves. Also an agitator associated with a soft starter is used.

Different specifications can be used to control the station.

Keywords

STEP7 software, siemens PLC (S7-314 IFM), WINCC flexible software, ZELIO schneider, zelio software, LOGO!, LOGO! Software, LOGO! TDE, prototype of hydraulic station of pumping.

RESUME

Le mémoire est une contribution à la réalisation d'une manipulation à base d'automates programmables. La station est gérée par différents automates (Siemens et Schneider).

La station est composée de deux tanks alimentés par une pompe à travers des électrovannes. Aussi un agitateur associé à un démarreur progressif est utilisé.

Différents cahiers de charges ont été utilisés pour commander la station.

Mots clés

Logiciel STEP7, automate programmable Siemens (S7-314 IFM), logiciel WinCC flexible, automate Zelio, logiciel Zelio Soft, automate LOGO!, logiciel LOGO!soft comfort, prototype de station de pompage.

Dédicace

J'ai le grand honneur de dédier ce modeste travail

A MON TRÈS CHER PÈRE : HAMDI-CHERIF Farouk

je te serai reconnaissante toute ma vie. Tu as su m'inculquer le sens de la responsabilité, de l'optimisme et de la confiance en soi face aux difficultés de la vie. Tes conseils ont toujours guidé mes pas vers la réussite. Ta patience sans fin, ta compréhension et ton encouragement sont pour moi le soutien indispensable que tu as toujours su m'apporter. Je te dois ce que je suis aujourd'hui et ce que je serai demain et je ferai toujours de mon mieux pour rester ta fierté. Que Dieu tout puissant te préserve.

A MA MAMAN : GACEM Djamila

la plus douce des mamans. Tu m'as comblé avec ta tendresse et ton affection. Tu n'as cessé de me soutenir et de m'encourager, tu as toujours été présente à mes côtés pour me consoler quand il le fallait. Puisse le tout puissant te donner santé, bonheur et longue vie afin que je puisse te combler à mon tour.

A TOUTE MA FAMILLE du plus petit au plus grand. A mes frères HALILOU et MIDOU, ASSIA et mes nièces LEILA et SOFIA.

A ma sœur de cœur MAYOUF Imane sur qui j'ai toujours pu compter. Elle est mon amie ma confidente, la sœur que je n'ai pas eue.

A mes amis AYOUB, MIRA, KARIM, OMAR.

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

HAMDI CHERIF Amel

Dédicace

A la femme la plus courageuse, sensible, généreuse, la plus belle à mes yeux, à celle qui a su me donner amour et joie de vivre, à celle qui a toujours montrée affection et compréhension à mon égard, le symbole de la tendresse. A la personne qui a toujours été présente pour moi, m'encourager et me pousser de l'avant. Celle qui m'a soutenu en toutes circonstances.

*A la femme de ma vie BOUDENNE Akila,
La mère que j'aime.*

A ceux qui m'ont donné joie et bonheur, mes soeurs Houda et Roufiada, ainsi qu'à mon petit ange, mon petit frère Rayan et à mon père MAYOUF Houcin.

A toute ma famille, mes grand-mères Selma et Aicha, ainsi qu'à mes tantes, mes oncles, cousins et cousines.

À ma meilleure amie, mon binôme et ma sœur HAMDI CHERIF Amel avec qui j'ai partagé pleins de beaux moments.

A tous ceux que j'aime, à tous ceux qui m'aiment, je dédie ce modeste travail.

MAYOUF Imane

Remerciements

En préambule à ce mémoire nous remerciant ALLAH de nous avoir donné la patience, le courage et la force d'accomplir ce travail.

Nous tenons à remercier notre encadreur Mr : BERKOUK, ses précieux conseils et son aide durant toute la période du travail.

Nous remercions très chaleureusement Mademoiselle BEHLOULI, pour son amitié, et l'aide précieuse qu'elle nous a apporté tout au long de ce travail.

Nos vifs remerciements vont également aux membres du jury H.ACHOUR et L.ABDELOUL pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail Et de l'enrichir par leurs propositions.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Sommaire

Liste des tableaux

Liste des figures

Liste des abreviations

Introduction générale	13
Chapitre I	14
Présentation de la maquette de station de pompage	14
1 Introduction	15
2 Description du prototype de la station de pompage	15
2.1 Schéma de principe	16
2.2 Descriptions des éléments électriques, de commande et de supervision du prototype	16
2.3 Description du schéma synoptique de la station	18
3 Description des cahiers de charge	18
3.1 Grafcet et définition du cahier de charge de la manipulation TOR -1ère manipulation-	18
3.2 Grafcet et définition du cahier de charge de la manipulation analogique -2ème manipulation-	19
3.3 Présentation du capteur de niveau à base d'Arduino et du circuit de conditionnement	20
3.3.1 Le capteur de niveau à base d'Arduino	20
3.3.1.1 Description du capteur HC-SR04	20
3.3.1.2 Principe de fonctionnement du capteur	21
3.3.1.3 Le montage	21
3.3.2 Le code (le programme) Arduino	22
3.3.3 Présentation du circuit de conversion PWM/ analogique	23
3.3.3.1 Le filtre passe bas	23
3.3.3.2 Le suiveur	23
3.3.3.3 Conversion ou amplification	23
4 Conclusion	24
Chapitre II	25
Commande et supervision du prototype de la station de pompage par l'automate S7 300	25
1 Introduction	26
2 Les automates programmables industriels	26
2.1 Structure interne d'un automate programmable industriel	26
2.2 Choix d'un API	26
2.3 Langage de programmation pour API	26
3 La gamme SIMATIC	27
3.1 Les variantes de la gamme SIMATIC	27
3.2 Description de la gamme SIMATIC S7 300	28
3.2.1 Structure d'un S7 300	28
3.2.2 Interface multipoints MPI	29
4 Logiciel STEP 7	29
4.1 Présentation	29
4.2 Fonctions de base du logiciel	29
4.2.1 Gestionnaire de projets SIMATIC	29
4.2.2 Editeur des mnémoniques	29

4.2.3	Diagnostic du matériel	30
4.2.4	Langages de programmation	30
4.2.5	Paramétrage de l'interface PG-PC	30
4.3	Création du projet SIMATIC Step7	30
4.3.1	Configuration du matériel	31
4.3.2	Edition des programmes	31
4.3.2.1	Blocs d'organisation	31
4.3.2.2	Fonctions et blocs fonctionnels	31
4.3.2.3	Bloc de données	32
4.4	Simulation de modules Avec le logiciel PLCSIM	32
4.5	Chargement du programme dans la CPU	32
4.6	Surveillance du fonctionnement et diagnostic du matériel	32
5	Logiciel de supervision WinCC	32
5.1	Présentation de WinCC	32
5.2	Fonctionnement du WinCC	33
5.2.1	Logiciel de configuration (CS)	33
5.2.2	Éléments de l'interface utilisateur de WinCC flexible	33
5.2.3	Logiciel WinCC flexible Runtime	33
5.3	Création d'un projet WinCC	34
5.3.1	Communication WinCC/STEP 7	34
5.3.2	Définition des variables	34
5.3.3	Création d'une vue de process	35
5.3.4	Création des alarmes	35
5.4	Test et simulation du projet	35
5.4.1	Test du projet	35
5.4.2	Simulation du projet	35
5.5	Transfert du projet	35
6	Commande du prototype de la station de pompage avec l'automate S7 300	36
6.1	Schéma de connexion des entrées et des sorties	36
6.2	Développement du projet sous logiciel STEP 7 et logiciel WinCC	37
6.2.1	1ère manipulation -partie TOR-	37
6.2.1.1	Développement du programme sous logiciel STEP 7	37
6.2.1.2	Supervision sous logiciel WinCC	38
6.2.2	2ème manipulation -partie analogique-	40
6.2.2.1	Développement du programme sous logiciel step 7	40
6.2.2.2	Supervision sous logiciel WinCC	41
7	Conclusion	43
Chapitre III		44
Commande du prototype de la station de pompage par l'automate Zelio		44
1	Introduction	45
2	Les modules programmables Zelio logic	45
2.1	Présentation	45
2.2	Programmation	45
2.3	Modules logiques compacts/modulaires	45
2.4	Module d'extension de communication	46
2.5	Communication	46
2.6	Programmation sur écran	47
2.7	Verrouillage façade	48

3	Le logiciel de programmation Zelio Soft	48
3.1	Présentation	48
3.2	Langages utilisés	48
3.2.1	Langage à contacts	48
3.2.2	Langage FBD	49
3.3	Test des programmes et et maintenance	50
3.4	Transfert d'une application	50
3.4.1	Transfert du programme PC vers le Zelio Logic	50
3.4.2	Transfert du programme Zelio Logic vers le PC	51
3.5	Fonction mot de passe	51
4	Commande du prototype de la station de pompage par l'automate Zelio Logic	52
4.1	Principe de fonctionnement	52
4.2	Schéma de connexion des entrées et des sorties	52
4.3	Développement du projet sous logiciel Zelio soft	54
5	Conclusion	54
Chapitre IV		55
Commande et supervision du prototype de la station de pompage par l'automate LOGO!		55
1	Introduction	56
2	Présentation	56
3	Fonctionnalité du LOGO!	56
4	Les types d'appareils	57
4.1	Les classes de tension du module LOGO! Base	57
4.2	Les versions du module LOGO! Base	57
4.3	Les modules d'extension	57
4.4	Les modules d'affichage	57
5	Présentation du LOGO! TDE	57
5.1	Fonctionnalité du LOGO! TDE	58
6	Programmation sous LOGO! Soft Comfort	58
6.1	Fonctionnalité du LOGO! Soft Comfort V8.0	58
6.2	Interface de programmation	59
6.3	Langage de programmation	60
6.4	Interface de projet	60
7	Commande d'une station de pompage avec LOGO! OBA8	61
7.1	Schéma de connexion des entrées et des sorties	62
7.2	Tableau d'affectation	62
7.3	Configuration de l'automate LOGO! et de son logiciel de programmation	63
7.3.1	Instruction structurée étape par étape	63
7.3.2	les noms des raccordements	66
7.3.3	1ère manipulation -partie TOR-	67
7.3.4	2ème manipulation -partie analogique-	67
7.4	Simulation du circuit	68
7.5	Transfert du programme testé dans le LOGO!	68
7.6	Test en ligne	68
8	Supervision de la maquette avec LOGO! TDE et via un navigateur Web sur PC	69
8.1	Affichage de texte d'alarme sur écran LOGO! TDE	69

8.1.1	Texte d'alarme	70
8.1.1.1	Insérer des textes d'alarme	70
8.1.1.2	Saisir le texte d'alarme	70
8.1.1.3	Affichage de la première manipulation TOR	71
8.1.1.4	Affichage de la première manipulation analogique	72
8.1.2	Définition du rétro-éclairage du texte d'alarme	73
8.1.3	Touches de fonction LOGO! TDE	74
8.2	Affichage de texte d'alarme via un navigateur Web [14]	74
8.2.1	Activation d'un serveur Web dans LOGO!	74
8.2.2	LOGO! Dans le navigateur Web	75
8.2.3	Fonctionnement du module virtuel sur le serveur Web	76
8.2.3.1	Sur le module LOGO! Base	77
8.2.3.2	Sur le LOGO! TDE	77
9	Conclusion	77
	Conclusion générale	78
	Bibliographie	79
	ANNEXE	

LISTE DES TABLEAUX

CHAPITRE I	14
PRESENTATION DE LA MAQUETTE DE STATION DE POMPAGE	14
TABLEAU 1 TABLEAU DES ELEMENTS ELECTRIQUES	16
CHAPITRE II	25
COMMANDE ET SUPERVISION DU PROTOTYPE DE LA STATION DE POMPAGE PAR L'AUTOMATE S7 300	25
TABLEAU 2 TABLE D'AFFECTATION DES ENTREES / SORTIES	37
TABLEAU 3 ADRESSES DES VARIABLES UTILISEES PAR LES FONCTIONS « SCALE »	40
CHAPITRE III	44
COMMANDE DU PROTOTYPE DE LA STATION DE POMPAGE PAR L'AUTOMATE ZELIO	44
TABLEAU 4 TABLEAU D'AFFECTATION DES ENTREES / SORTIES	53
CHAPITRE IV	55
COMMANDE ET SUPERVISION DU PROTOTYPE DE LA STATION DE POMPAGE PAR L'AUTOMATE LOGO!	55
TABLEAU 5 TABLEAU D'AFFECTATION DES ENTRÉES / SORTIES	62

LISTE DES FIGURES

CHAPITRE I	14
PRESENTATION DE LA MAQUETTE DE STATION DE POMPAGE	14
FIGURE 1 PROTOTYPE DE STATION DE POMPAGE	15
FIGURE 2 REPRESENTATION DE LA STATION	16
FIGURE 3 DESCRIPTION DU SCHEMA SYNOPTIQUE DE LA STATION	18
FIGURE 4 GRAFCET REPRESENTANT LA MANIPULATION TOR	19
FIGURE 5 GRAFCET REPRESENTANT LA MANIPULATION ANALOGIQUE	20
FIGURE 6 CAPTEUR HC-SR04	20
FIGURE 7 PRINCIPE DE FONCTIONNEMENT DU CAPTEUR	21
FIGURE 8 VUE DU PROTOTYPE DU MONTAGE	21
FIGURE 9 DIAGRAMME DU CIRCUIT DE DE CONVERSION PWM/ANALOGIQUE	23
CHAPITRE II	25
COMMANDE ET SUPERVISION DU PROTOTYPE DE LA STATION DE POMPAGE PAR L'AUTOMATE S7 300	25
FIGURE 10 STRUCTURE INTERNE D'UN AUTOMATE PROGRAMMABLE	27
FIGURE 11 CONSTITUANTS D'UN AUTOMATE S7-300	28
FIGURE 12 LE GESTIONNAIRE DE PROJET STEP 7	29
FIGURE 13 FENETRE D'UN NOUVEAU PROJET STEP 7	31
FIGURE 14 LOGICIEL DE SIMULATION PLC-SIM	32
FIGURE 15 ELEMENTS DE L'INTERFACE WINCC	33
FIGURE 16 CONFIGURATION DU PC COMME PUPITRE	34
FIGURE 17 LIAISON MPI ENTRE LE PUPITRE OPERATEUR ET LA STATION SIMATIC	34
FIGURE 18 FENETRE DES VARIABLES	35
FIGURE 19 SCHEMA DE CONNEXION DES ENTREES ET DES SORTIES TOR	36
FIGURE 20 SCHEMA DE CONNEXION DES ENTREES ANALOGIQUES	36
FIGURE 21 TABLE DES MNEMONIQUES PARTIE TOR	38
FIGURE 22 TABLE DES VARIABLES DE LA MANIPULATION TOR SUR WINCC FLEXIBLE	38
FIGURE 23 VUE DE DEMARRAGE	39
FIGURE 24 VUE DE LA MAQUETTE DE LA PREMIERE MANIPULATION TOR	39
FIGURE 25 SIMULATION DE LA PREMIERE MANIPULATION TOR	40
FIGURE 26 FONCTION SCALE APPELEE DANS LE BLOC OB 1 POUR LES DEUX ENTREES ANALOGIQUES	40
FIGURE 27 TABLE DES MNEMONIQUES PARTIE ANALOGIQUE	41
FIGURE 28 TABLE DES VARIABLES DE LA MANIPULATION ANALOGIQUE	41

FIGURE 29 VUE DE LA MAQUETTE DE LA MANIPULATION ANALOGIQUE	42
FIGURE 30 TABLE DES ALARMES ANALOGIQUES.....	42
FIGURE 31 VUE DES ALARMES	42
FIGURE 32 SIMULATION DE LA DEUXIEME MANIPULATION ANALOGIQUE	43
CHAPITRE III	44
COMMANDE DU PROTOTYPE DE LA STATION DE POMPAGE PAR L'AUTOMATE ZELIO	44
FIGURE 33 MODULES LOGIQUES COMPACTS	45
FIGURE 34 MODULES D'EXTENSION D'ENTREES/SORTIES	46
FIGURE 35 AFFICHEUR DE L'AUTOMATE ZELIO.....	47
FIGURE 36 REGLES DE SAISIE DU SCHEMA DE COMMANDE.....	47
FIGURE 37 STRUCTURE D'UN RESEAU DE CONTACTS	49
FIGURE 38 EXEMPLE D'UN PROGRAMME EN LANGAGE FBD	50
FIGURE 39 FENETRE DE DIALOGUE D'UN MODULE ZELIO	51
FIGURE 40 AUTOMATE PROGRAMMABLE ZELIO (SR2 B201BD)	52
FIGURE 41 SCHEMA DE CONNEXION DES ENTREES ET DES SORTIES TOR	52
FIGURE 42 SCHEMA DE CONNEXION DES ENTREES ANALOGIQUES.....	53
FIGURE 43 PROGRAMMATION SUR ECRAN AVEC LE LANGAGE LADDER.....	54
CHAPITRE IV	55
COMMANDE ET SUPERVISION DU PROTOTYPE DE LA STATION DE POMPAGE PAR L'AUTOMATE LOGO!	55
FIGURE 44 AUTOMATE LOGO! ET UN AFFICHEUR LOGO! TDE	56
FIGURE 45 AFFICHEUR LOGO! TDE	58
FIGURE 46 INTERFACE VIERGE DU LOGICIEL LOGO ! SOFT COMFORT.....	59
FIGURE 47 BARRE D'OUTILS DE LA COMMANDE « SIMULATION »	60
FIGURE 48 FENETRE DE BLOCS DE PROGRAMME.....	61
FIGURE 49 AUTOMATE PROGRAMMABLE LOGO ! OBA8	61
FIGURE 50 SCHEMA DE CONNEXION DES ENTREES ET DES SORTIES.....	62
FIGURE 51 REGLAGE DE L'ADRESSE IP DANS LE MENU DU MODULE LOGO !.....	63
FIGURE 52 PROGRAMMATION SUR ECRAN	64
FIGURE 53 FENETRE DE CONFIGURATION DES PARAMETRES DU RESEAU	64
FIGURE 54 FENETRE DE CONFIGURATION DES PARAMETRES DU MODULE.....	65
FIGURE 55 FENETRE CONTENANT LES INFORMATIONS DE BASE DU MODULE LOGO!	65
FIGURE 56 FENETRE DE CONFIGURATION DES VOIES ANALOGIQUES.....	66
FIGURE 57 FENETRE DE CONFIGURATION DES MNEMONIQUES	66
FIGURE 58 FENETRE DE CONFIGURATION DES NOMS DES RACCORDEMENTS	67
FIGURE 59 TABLE DES ENTREES/SORTIES PARTIE TOR	67
FIGURE 60 TABLE DES ENTREES/SORTIES PARTIE ANALOGIQUE	67
FIGURE 61 INTERFACE DE COMMUNICATION ENTRE LE PC ET LE MODULE LOGO!	68
FIGURE 62 AFFICHEUR SUR ECRAN (TEST EN LIGNE)	69
FIGURE 63 FENETRE DE CONFIGURATION DES PARAMETRES DE L'AFFICHEUR LOGO! TDE	69
FIGURE 64 FENETRE DE VERIFICATION DES PROPRIETES DE L'AFFICHEUR LOGO! TDE	70
FIGURE 65 FENETRE D'INSERTION DE TEXTE D'ALARME.....	70
FIGURE 66 FENETRE PERMETTANT DE PARAMETRER LE TEXTE D'ALARME	71
FIGURE 67 EXEMPLE D'AFFICHAGE DE LA MANIPULATION TOR	71
FIGURE 68 AFFICHAGE « PARTIE TOR » SUR ECRAN LOGO!.....	72
FIGURE 69 AFFICHAGE « PARTIE TOR » SUR ECRAN LOGO! TDE.....	72
FIGURE 70 EXEMPLE D'AFFICHAGE DE LA MANIPULATION ANALOGIQUE	72
FIGURE 71 FENETRE PERMETTANT DE DEFINIR UN RETRO-ECLAIRAGE DU TEXTE D'ALARME.....	73
FIGURE 72 EXEMPLE DE RETROECLAIRAGE	73
FIGURE 73 EXEMPLE D'UN BLOC MESSAGE TEXTE	73
FIGURE 74 EXEMPLE DE RETRO-ECLAIRAGE DU TEXTE D'ALARME UTILISE	74
FIGURE 75 FENETRE PERMETTANT DE CONFIGURER LA LIAISON PC/MODULE	75

FIGURE 76 FENETRE PARAMETRES POUR LE CONTROLE D'ACCES	75
FIGURE 77 FENETRE PERMETTANT DE CONFIGURER LE LOGO! DANS LE NAVIGATEUR WEB.....	76
FIGURE 78 FENETRE PERMETTANT DE VISUALISER LE PROGRAMME LOGO! DANS LE NAVIGATEUR WEB	76
FIGURE 79 EXEMPLE DE VISUALISATION DE LA MANIPULATION TOR DU MODULE LOGO !	77
FIGURE 80 EXEMPLE DE VISUALISATION DE LA MANIPULATION TOR DU MODULE LOGO ! TDE	77

LISTE DES ABREVIATIONS

AA sortie analogique	PWM Pulse Width Modulation (Modulation à Largeur d'impulsion)
AC Alternative Current	Appareil 0BA8 La dernière version du module LOGO! Basic, décrite dans le présent manuel
AE entrée analogique	TDE Afficheur de texte avec interfaces Ethernet
AI Analog Input	FE Functional Earth, terre fonctionnelle
AO Analog Output	
API Automate Programmable Industriel	
CHAR caractère	
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	
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	
LIST Le langage de liste d'instructions	
LOG LOGigramme	
MPI Multi-Point Interface	
OB Organisationnel Bloc	
OP Operator Pupiter	
PC Personnel Computer	
PG la console de programmation sur le terrain	
PLC Programmable Logic Controller	
PROFIBUS Process Field Bus	
PROM Programmable Read Only Memory	
EEPROM Electrically Erasable Programmable Read Only Memory	
PS Gamme des alimentations stabilisées 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

Dans le but de former des élèves ingénieurs à la maîtrise des automates programmables, le département d'automatique de l'ENP a mis en leurs dispositions une maquette de station de pompage à base d'automates Programmables « Siemens et Schneider », afin de l'utiliser par la suite dans le cadre des Travaux pratiques au sein du laboratoire comme un système d'apprentissage didactique pour l'automatisation, La maquette est conçue dans le but d'appliquer les connaissances acquises dans l'informatique industrielle, et pour répondre à un certain nombre d'exigences de formation professionnelle.

Nous exposons dans le présent rapport quatre grands chapitres décrivant les volets principaux de notre projet:

Le premier chapitre sera porté sur la description du système étudié, une présentation de la maquette de la station de pompage détaillée y sera mise en évidence. Nous y déterminerons les éléments de l'application afin d'établir des cahiers de charges adéquats au processus, par ailleurs nous y détaillerons une partie de l'application qui sera nécessaire pour les chapitres suivants dont la réalisation d'un capteur de niveau à base d'Arduino et d'un circuit de conditionnement du signal.

Dans le deuxième chapitre nous aborderons la description des automates programmables, particulièrement le module S7 300 de la ferme siemens, nous y décrirons le logiciel de programmation STEP7 ainsi que le logiciel de supervision WinCC flexible, qui permettent le contrôle facile de la maquette a fin de maintenir le prototype de la station de pompage en état de marche, nous y présenterons les différentes réalisations effectuées a l'aide de ce même automate.

Le troisième chapitre sera consacré au contrôle du prototype qui s'effectuera par le biais de l'automate ZELIO logic de la filiale Télémécanique de Schneider, nous y aborderons aussi la description du logiciel de programmation Zelio Soft, ainsi que les différentes manipulations effectuées sur l'automate.

Le quatrième et dernier chapitre sera porté sur la Commande et la supervision du prototype de la station de pompage par le biais de l'automate LOGO! de la ferme siemens, une description de l'automate ainsi que de son logiciel de programmation LOGO ! Soft comfort y sera détaillée, nous y aborderons toutes les étapes à suivre pour le traitement des cahiers de charges proposés au premier chapitre, par ailleurs nous y présenterons deux moyens de supervisions, dont la configuration d'un écran d'affichage LOGO ! TDE programmable sur LOGO ! Soft comfort, ainsi que l'exploitation du module LOGO! Base et LOGO! TDE sur un PC classique ou un appareil mobile pour un Affichage via un navigateur Web.

Le but majeur de ce projet est la réalisation de manipulations très simples et très claires aux étudiants, dont les actionneurs peuvent être contrôlés grâce aux API. Les étudiants pourront ainsi programmer des manipulations à l'aide de logiciels spécifiques a chaque automates, puis simuler et charger dans les modules, pour finalement visualiser l'enchaînement de leurs cahiers de charges.

Chapitre I

Présentation de la maquette de station de pompage

1 Introduction

Une maquette de station de pompage a été réalisée au sein du laboratoire d'automatique, nous avons été chargé de séparer la manipulation en deux parties dont la 1ère partie est gérée par l'automate Zelio de Schneider, et la 2ème partie est gérée par deux automates S7 300 et LOGO ! de Siemens que nous aurons à traiter dans un autre chapitre, et ce afin d'obtenir deux manipulations bien séparées et distinctes pour les travaux à venir.

Tout au long de ce chapitre, nous allons décrire notre prototype. Tout d'abord d'un point de vue instrumentation, à savoir les différents capteurs ainsi que les actionneurs utilisés. Ensuite, nous décrivons les cahiers de charges des manipulations (TOR/analogique) effectués sur le prototype de la station de pompage. Néanmoins pour effectuer la manipulation analogique nous aborderons la manière de concevoir un capteur de niveau à l'aide d'Arduino.

2 Description du prototype de la station de pompage

La station de pompage est un dispositif didactique destiné à l'étude et au contrôle de plusieurs systèmes. Le contrôle de ces derniers se fait par le biais d'automates programmables. La figure 1 donne une vue générale de la maquette, qui permet de réaliser différentes manipulations sur les mêmes principes que les stations de pompes hydrauliques.

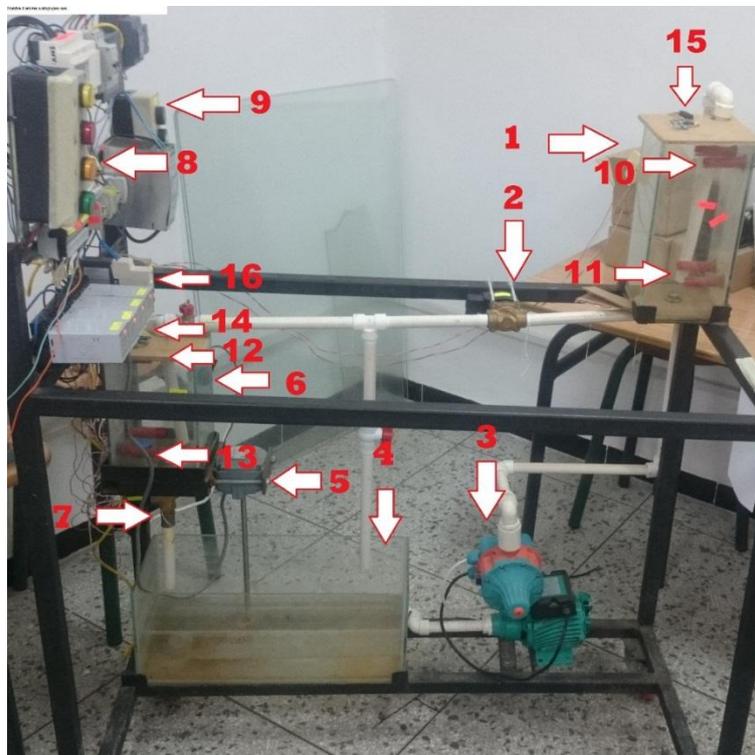


Figure 1 Prototype de station de pompage

- | | |
|---------------------|---|
| 1 Cuve A | 9 Boutons de commande (reset/marche) |
| 2 Electrovanne A | 10 Capteur de niveau TOR, Ha |
| 3 Pompe hydraulique | 11 Capteur de niveau TOR, La |
| 4 Réservoir | 12 Capteur de niveau TOR, Hb |
| 5 Agitateur | 13 Capteur de niveau TOR, Lb |
| 6 Cuves B | 14 Capteur ultrason HC-SR04 de la cuveB |
| 7 Electrovanne B | 15 Capteur ultrason HC-SR04 de la cuveA |
| 8 voyants | 16 Démarreur progressif |

2.1 Schéma de principe

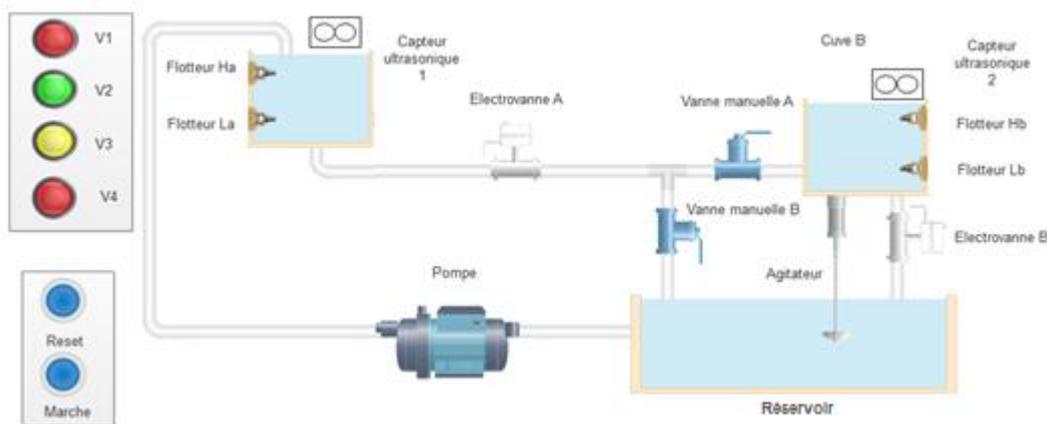


Figure 2 Représentation de la station

2.2 Descriptions des éléments électriques, de commande et de supervision du prototype

Tableau 1 Tableau des éléments électriques

1. Support métallique : le prototype repose sur un support métallique 1.50 m sur 1m et d'une hauteur de 1.50m, tous les éléments du prototype s'y trouvent.
2. Réservoir : réservoir C de 0.50m sur 0.20m sur une hauteur de 0.20m qui sert à réserver de l'eau, et fait partie du procédé.
3. Cuves: Deux Cuves (A et B) de 0.15m sur 0.15m sur une hauteur de 0.30m/0.25m.
4. éléments de tuyauteries en PEHD (Polyéthylène Haute Densité) Tigre de diamètre DN40,
5. Une pompe centrifuge (QB50) TAIFU, couplé à un moteur asynchrone monophasé alimenté à 220 V La hauteur manométrique maximale de cette pompe est de 20 m.

CHAPITRE I : Présentation de la maquette de station de pompage

6. Agitateur: Un mélangeur(Agitateur) OBIMEX alimenté en 220 v pour agiter le liquide.
7. Un démarreur progressif (ATS01N03FT) SCHNEIDER qui améliore les performances de démarrage du mélangeur en permettant un démarrage progressif et contrôlé.
8. Relais : Quatre relais de niveau JYB-714C 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).
9. Alimentation: Une alimentation stabilisée (Harvest electric HAS-120-24) alimenté en monophasé 220 V et qui délivre du 24V.
10. Répartiteurs: Deux répartiteurs de tension pour le 24 V et 220 V.
11. Disjoncteurs: Un disjoncteur HARVEST pour la sécurité de toute l'installation, deux autres pour la sécurité des API.
12. Electrovanes: Deux électrovannes HARVEST (ZSC-1D) DN40 TOR alimentées en 24V DC. Qui servent à régler l'écoulement d'eau
13. Robinet: un robinet d'arrêt à opercule DN40, utilisés pour permettre la flexibilité du prototype afin qu'on puisse l'utiliser dans deux manipulations différentes (ou plus).
14. Automates (API/PLC): Un automate Zelio de Schneider Electric (SR2 B201BD) ainsi qu'un automate LOGO! (0BA8) de la firme siemens qui exécutent les cahiers de charges et qui contrôlent la station
15. Arduino: Une carte de développement qui réalise des calculs pour mesurer le niveau d'eau.
16. Capteur: Deux capteurs ultrason hc-05 sont utilisés pour la mesure du niveau.
17. Circuit convertisseur: Deux circuits qui reçoit un signal PWM et délivre un signal analogique
18. Une station PC pour la supervision.
19. 2 Boutons et 4 voyants
20. 18 Borniers.

2.3 Description du schéma synoptique de la station

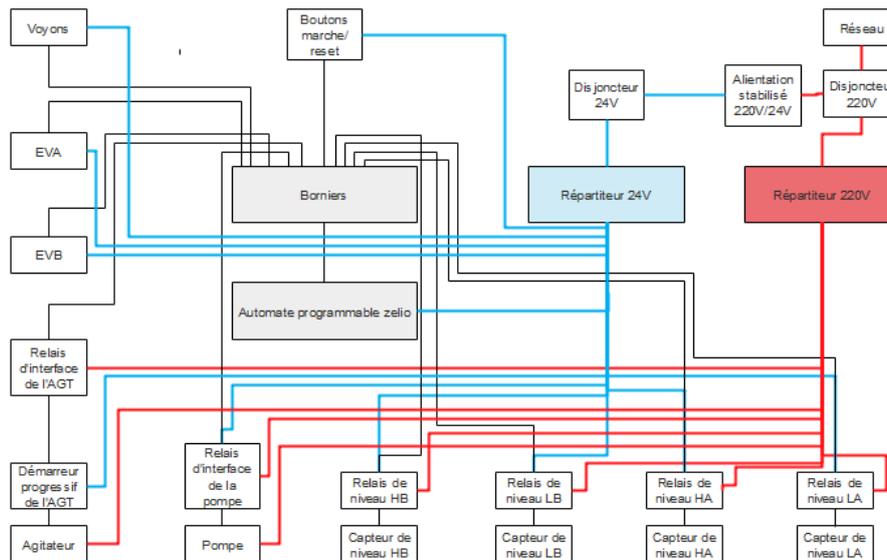


Figure 3 Description du schéma synoptique de la station

3 Description des cahiers de charge

Les cahiers de charge suivants sont un exemple parmi tant d'autres.

3.1 Grafcet et définition du cahier de charge de la manipulation TOR -1ère manipulation-

La station étant à l'arrêt, 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 l'eau du réservoir à la cuve A, 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 déverser l'eau de la cuve A dans la cuve B, le voyant V1 s'allume jusqu'à ce que le niveau haut de la cuve 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 versée dans le réservoir par le biais du robinet, une fois cette cuve vidée, la même procédure est exécutée avec l'électrovanne EVB et la cuve B. une fois les deux cuves vides, une temporisation de 5 secondes suivie du lancement de l'agitateur pour la même durée s'effectue.

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, et pour finir le voyant V3 indique l'activation de la pompe.

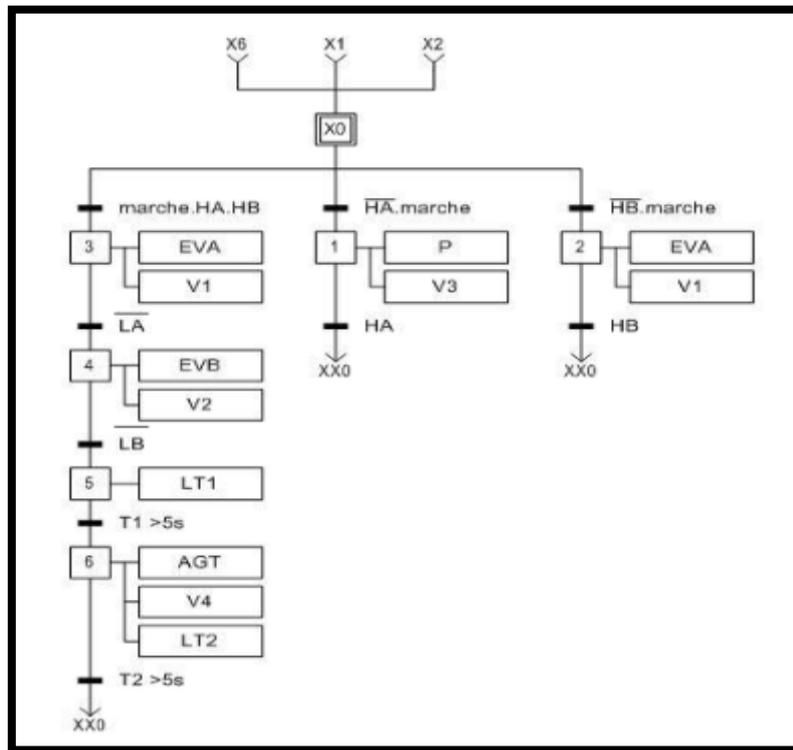


Figure 4 Grafcet représentant la manipulation TOR

3.2 Grafcet et définition du cahier de charge de la manipulation analogique -2ème manipulation-

Les symboles d1 et d2 représentent, respectivement, le niveau d'eau dans les cuve A et B. La station étant à l'arrêt, 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, 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 déverser de l'eau de la cuve A vers la cuve B, 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 versée dans le réservoir par le biais du robinet, jusqu'à ce que d1 atteigne 7 cm, la même procédure est exécutée avec l'électrovanne EVB et la cuve B jusqu'à ce que d2 atteigne 5 cm. 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 les voyants V3 et V4 indiquent respectivement l'activation de la pompe et l'agitateur.

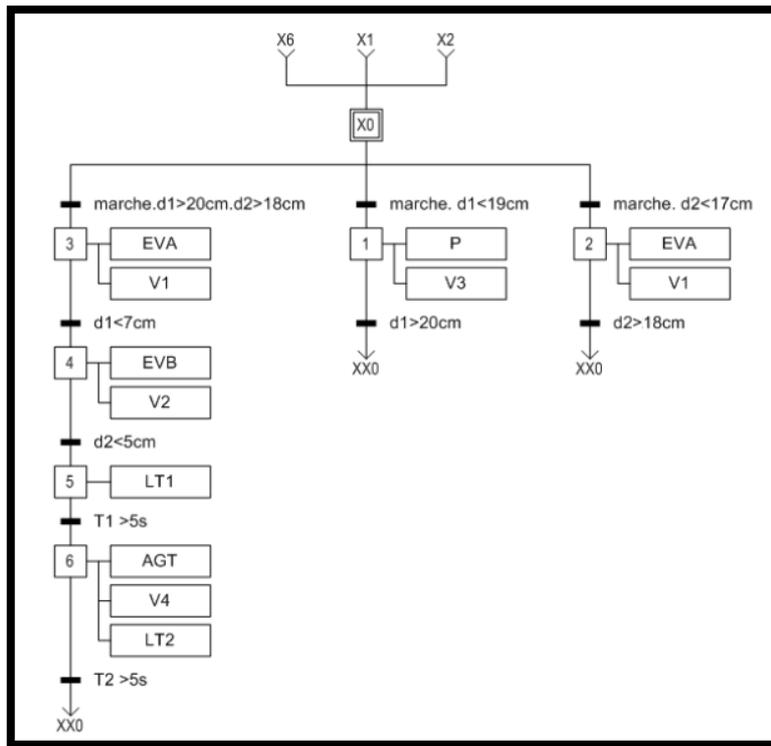


Figure 5 Grafcet représentant la manipulation analogique

3.3 Présentation du capteur de niveau à base d'Arduino et du circuit de conditionnement

Par faute de temps et de budget, le contrôle de niveau du prototype de la station de pompage via les automates programmables a été réalisé par ALLALI Ahmed et BETTACHE Lyes Heithem PFE 2017 grâce a un capteur aux ultrasons HC-SR04 en utilisant une carte Arduino. Un circuit convertisseur est indispensable pour transmettre l'information de l'Arduino à l'API. La carte Arduino fournit un signal PWM et l'API en reçoit de l'analogique.

3.3.1 Le capteur de niveau à base d'Arduino [1]

3.3.1.1 Description du capteur HC-SR04



Figure 6 Capteur HC-SR04

Le capteur HC-SR04 est un capteur à ultrason low cost. Ce capteur fonctionne avec une tension d'alimentation de 5 volts, dispose d'un angle de mesure de 15° environ et permet de faire des mesures de distance entre 2 centimètres et 4 mètres avec une précision de 3mm.

3.3.1.2 Principe de fonctionnement du capteur

Le principe de fonctionnement du capteur est entièrement basé sur la vitesse du son. Le déroulement d'une prise de mesure s'effectue en envoyant une impulsion **HIGH** de 10µs sur la broche **TRIGGER** du capteur. Ce dernier envoie une série de 8 impulsions ultrasoniques à 40KHz, les ultrasons se propagent dans l'air jusqu'à toucher un obstacle et retournent dans l'autre sens vers le capteur. Il détecte l'écho et clôture la prise de mesure. Le signal sur la broche **ECHO** du capteur reste à **HIGH**, ce qui permet de mesurer la durée de l'aller-retour des ultrasons et donc de déterminer la distance.

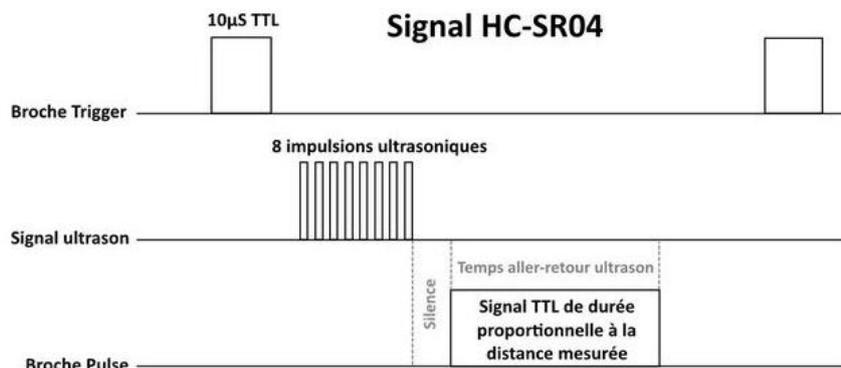


Figure 7 Principe de fonctionnement du capteur

3.3.1.3 Le montage

Pour réaliser le montage, il faut:

Une carte Arduino UNO (et son câble USB), un capteur HC-SR04, une plaque d'essai et des fils pour câbler le montage.

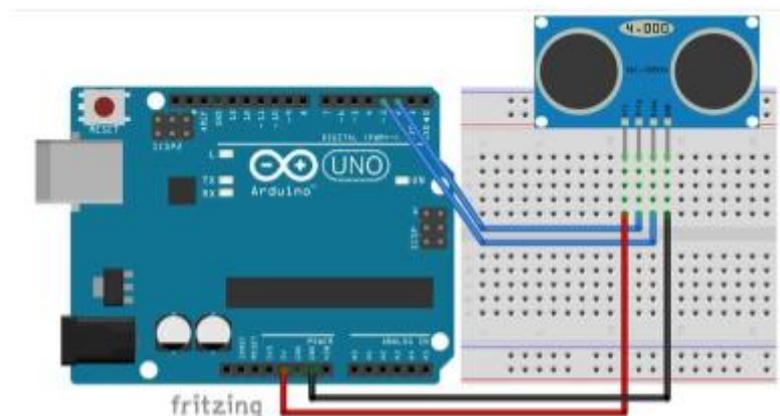


Figure 8 Vue du prototype du montage

- L'alimentation **5V** de la carte Arduino va sur la broche **VCC** du capteur.
- La broche **GND** de la carte Arduino va sur la broche **GND** du capteur.
- La broche **D2** de la carte Arduino va sur la broche **TRIGGER** du capteur.
- La broche **D3** de la carte Arduino va sur la broche **ECHO** du capteur.

3.3.2 Le code (le programme) Arduino

```

1  /* Constantes pour les broches */
2  const byte TRIGGER_PIN = 2; // Broche TRIGGER
3  const byte ECHO_PIN = 3;    // Broche ECHO
4
5  /* Constantes pour le timeout */
6  const unsigned long MEASURE_TIMEOUT = 25000UL; // 25ms = ~8m à 340m/s
7
8  /* Vitesse du son dans l'air en mm/us */
9  const float SOUND_SPEED = 340.0 / 1000;

```

le code commence avec quatre constantes, deux constantes pour les broches `TRIGGER` et `ECHO` du capteur, une constante qui servira de timeout (temps nécessaire avant de considérer qu'il n'y a pas d'obstacle, donc pas de mesure possible) pour la prise de mesure et une constante pour définir la vitesse du son.

```

1  void setup() {
2
3      /* Initialise le port série */
4      Serial.begin(115200);
5
6      /* Initialise les broches */
7      pinMode(TRIGGER_PIN, OUTPUT);
8      digitalWrite(TRIGGER_PIN, LOW); // La broche TRIGGER doit être à LOW au repos
9      pinMode(ECHO_PIN, INPUT);
10 }

```

La fonction `setup()` initialise le port série, met la broche `TRIGGER` du capteur en sortie à `LOW`, et met la broche `ECHO` du capteur en entrée.

```

1  void loop() {
2
3      /* 1. Lance une mesure de distance en envoyant une impulsion HIGH de 10µs sur la broche TRIGGER */
4      digitalWrite(TRIGGER_PIN, HIGH);
5      delayMicroseconds(10);
6      digitalWrite(TRIGGER_PIN, LOW);
7
8      /* 2. Mesure le temps entre l'envoi de l'impulsion ultrasonique et son écho (si il existe) */
9      long measure = pulseIn(ECHO_PIN, HIGH, MEASURE_TIMEOUT);
10
11     /* 3. Calcul la distance à partir du temps mesuré */
12     float distance_mm = measure / 2.0 * SOUND_SPEED;
13
14     /* Affiche les résultats en mm, cm et m */
15     Serial.print(F("Distance: "));
16     Serial.print(distance_mm);
17     Serial.print(F("mm ("));
18     Serial.print(distance_mm / 10.0, 2);
19     Serial.print(F("cm, "));
20     Serial.print(distance_mm / 1000.0, 2);
21     Serial.println(F("m"));
22
23     /* Délai d'attente pour éviter d'afficher trop de résultats à la seconde */
24     delay(500);
25 }

```

La fonction `loop()` s'occupe de la mesure et de l'affichage. Elle génère d'abord l'impulsion `HIGH` de $10\mu\text{s}$ qui déclenche la prise de mesure. Elle mesure ensuite le temps

nécessaire a pour un aller-retour (deux fois la distance mesurée) du signal ultrason avec la fonction `pulseIn()`. Pour finir, elle calcule la distance avant de l'afficher sur le port série.

La carte Arduino va générer des tensions PWM dont le rapport cyclique est proportionnel à la distance mesurée avec une amplitude entre 0 et 5V. Ceci nous conduit à la prochaine étape qui est la conversion et l'amplification de cette amplitude à 0-10V.

3.3.3 Présentation du circuit de conversion PWM/ analogique [2]

Le circuit utilisé est un circuit qui convertit un signal PWM dont l'amplitude est de 5V, et dont la fréquence est de 490 Hz en un signal analogique 0-10V. Ce convertisseur se compose de trois parties :

3.3.3.1 Le filtre passe bas

Le filtre passe-bas est un dispositif qui démontre une réponse en fréquence relativement constante (gain fixe) en basses fréquences et un gain décroissant aux fréquences supérieures à la fréquence de coupure « f_c ». Les filtres passe-bas sont utilisés pour réduire l'amplitude des composantes de fréquences supérieures à la fréquence de coupure.

3.3.3.2 Le suiveur

Le suiveur de tension permet de prélever une tension sans la perturber car l'interaction des impédances des montages amont et aval altère la tension E prélevée. Le rôle du suiveur est de transmettre la tension avec un courant extrait nul.

3.3.3.3 Conversion ou amplification

Un amplificateur électronique est un système électronique augmentant la tension et/ou l'intensité d'un signal électrique. Il est utilisé pour avoir une tension (0-10V) à partir d'une tension (0-5V).

La figure 9 représente le Diagramme du circuit de de conversion PWM/analogique :

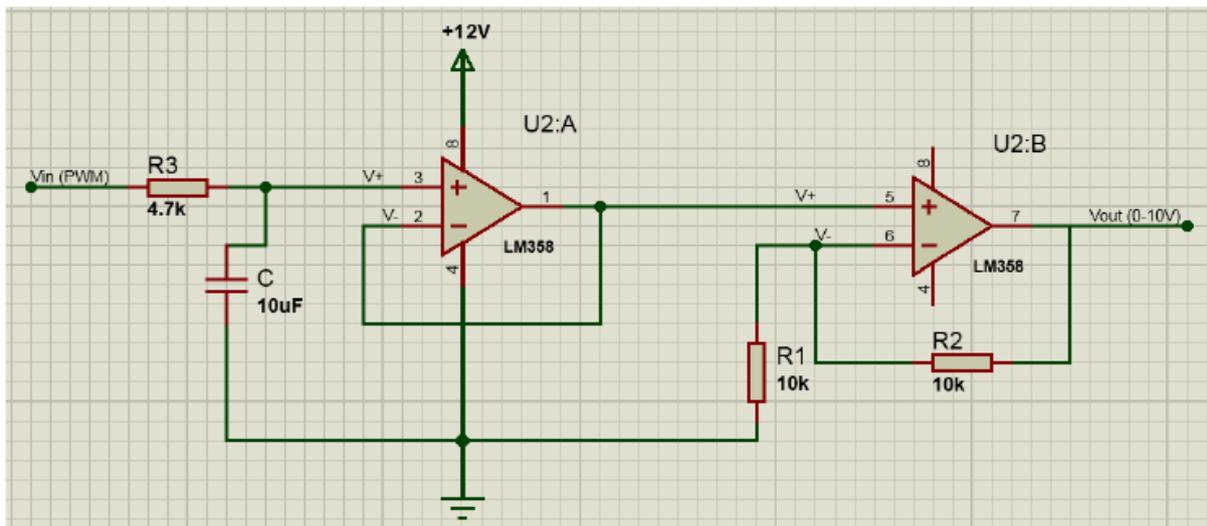


Figure 9 Diagramme du circuit de de conversion PWM/analogique

4 Conclusion

Dans ce chapitre nous avons pu étudier le système de station de pompage, les éléments qui le constitue, à savoir les capteurs TORs et analogiques, ainsi que les actionneurs que nous devons commander.

Nous avons aussi procédé à la définition des cahiers de charge, que nous utiliserons par la suite dans les chapitres pour le contrôle du système grâce aux API.

De plus nous avons pu se familiariser avec les cartes Arduino et leurs programmations dans le cadre d'une mesure de distance analogique du prototype.

Chapitre II

Commande et supervision du prototype de la station de pompage par l'automate S7 300

1 Introduction

Dans ce chapitre nous allons réaliser le contrôle de la station de pompage avec l'automate S7-300 de la firme Siemens à l'aide du logiciel STEP7, ainsi que la supervision de cette dernière à l'aide du logiciel WinCC flexible, nous y présenterons dans un premier temps les généralités sur les API, puis le module S7-300 en particulier, ainsi que ses fonctionnalités. Pour finir nous traiterons les exemples de cahier de charge développés au premier chapitre en les implémentant dans des programmes pour un contrôle et une visualisation du procédé.

2 Les automates programmables industriels

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

2.1 Structure interne d'un automate programmable industriel

Les API comportent quatre principales parties :

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

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

2.2 Choix d'un API [3]

Le choix d'un API est en fonction de la partie commande à programmer. Il est impératif de tenir compte de plusieurs critères :

- le nombre d'E/S, en prenant en considération les possibilités d'extension.
- la taille du programme.
- la vitesse de traitement.
- les fonctions complémentaires: temporisateurs et compteurs.
- Les langages de programmation.
- Le nombre de voies analogiques.
- la capacité de traitement arithmétique.
- la bibliothèque fonctionnelle.

2.3 Langage de programmation pour API [4]

Les programmes utilisés avec les API peuvent être écrits dans différents formats. Pour que les ingénieurs ayant peu de connaissances en programmation puissent élaborer des programmes pour les API.

La norme IEC 1131-3 définit cinq langages qui peuvent être utilisés pour la programmation des automates programmables industriels :

- **Le langage LD (Ladder Diagram) :**

C'est une représentation graphique d'équations booléennes combinant des contacts (en entrée) et des relais (en sortie). Il permet la manipulation de données booléennes, à l'aide de symboles graphiques organisés dans un diagramme comme les éléments d'un schéma électrique.

- **Le langage IL (Instruction List) :**

Un langage textuel de bas niveau. Il est particulièrement adapté aux applications de petites tailles. L'opérateur indique le type d'opération à effectuer entre le résultat courant et l'opérande. Le résultat de l'opération est stocké à son tour dans le résultat courant.

- **Le langage FBD (Function Block Diagram) :**

C'est un langage graphique. Il permet la construction d'équations complexes à partir des opérateurs standards, de fonctions ou de blocs fonctionnels.

- **Le langage ST (Structured Text) :**

Un langage textuel de haut niveau dédié aux applications d'automatisation. Ce langage est principalement utilisé pour décrire les procédures complexes, difficilement modélisables avec les langages graphiques. C'est le langage par défaut pour la programmation des actions dans les étapes et des conditions associées aux transitions du langage SFC.

- **Le langage SFC (Sequential Function Chart), ou GRAFCET :**

C'est un langage graphique utilisé pour décrire les opérations séquentielles. Le procédé est représenté comme une suite connue d'étapes (états stables), reliées entre elles par des transitions, une condition booléenne est attachée à chaque transition.

3 La gamme SIMATIC [5]

Aujourd'hui SIMATIC est devenu synonyme de l'intégration totale. L'intégration totale est un concept révolutionnaire visant à réunir l'univers de la fabrication manufacturière et l'univers des procédés. Toutes les briques matérielles et logicielles nécessaires à la réalisation d'un projet portent désormais un seul nom : SIMATIC.

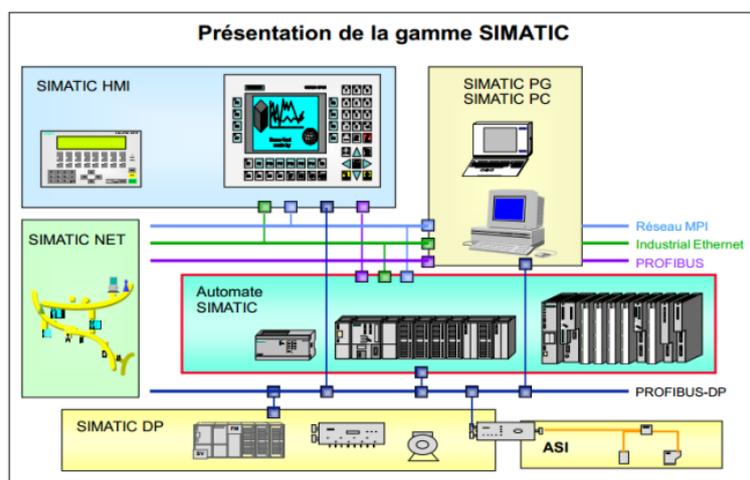


Figure 10 Structure interne d'un automate programmable

3.1 Les variantes de la gamme SIMATIC

Cette gamme est composée de trois familles:

- **SIMATIC S7**

SIMATIC S7 désigne un produit de la société SIEMENS et est synonyme de la nouvelle gamme des automates programmables.

- **SIMATIC C7**

Le SIMATIC C7 comprend deux sections travaillant indépendamment l'une de l'autre et pouvant communiquer entre elles à travers l'interface MPI la CPU et le pupitre opérateur OP.

- **SIMATIC M7**

SIMATIC M7 est un ordinateur industriel compatible PC. Il s'agit d'un système modulaire sous boîtier, construit dans la technique des automates SIMATIC S7. Il peut être intégré dans un automate S7-300/400 ou être utilisé comme système autonome avec une périphérie choisie dans la gamme S7.

3.2 Description de la gamme SIMATIC S7 300 [6]

Cette gamme comprend des CPU de différents niveaux de performance, des modules de signaux pour entrées et sorties TOR et analogiques, des modules de fonction pour différentes fonctions technologiques des CP pour les tâches de communication, des modules d'alimentation pour le raccordement du S7-300 sur secteur 120/230 V, des coupleurs pour configurer un automate sur plusieurs profiles-supports. Tous les modules du S7-300 sont montés sous boîtier procurant le degré de protection IP20 et peuvent fonctionner sans ventilation forcée.

3.2.1 Structure d'un S7 300

Un automate S7-300 est constitué des éléments suivants :

- Alimentation (PS).
- CPU.
- Modules de signaux (SM) .
- Modules de fonction (FM).
- Processeurs de communication.

Plusieurs automates S7-300 peuvent communiquer entre eux et avec d'autres automates SIMATIC S7 au moyen d'un câble-bus PROFIBUS. Le S7-300 est programmé à l'aide d'une console de programmation (PG). Cette PG est à relier à la CPU par un câble PG.

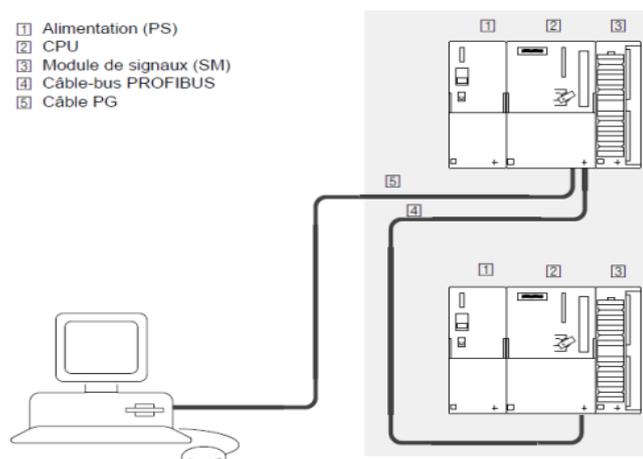


Figure 11 Constituants d'un automate S7-300

3.2.2 Interface multipoints MPI

L'interface MPI est l'interface de communication utilisée pour la console de programmation (PG), le pupitre opérateur (OP) ou pour la communication au sein d'un sous réseau MPI.

4 Logiciel STEP 7 [7]

4.1 Présentation

STEP 7 est le progiciel de base pour la configuration et la programmation de systèmes d'automatisation SIMATIC. Le progiciel de base STEP 7 existe en plusieurs versions :

- STEP 7-Micro/DOS et STEP 7-Micro/Win pour des applications autonomes simples sur SIMATIC S7 - 200.
- STEP 7 pour des applications sur SIMATIC S7-300/400, SIMATIC M7-300/400 et SIMATIC C7.

4.2 Fonctions de base du logiciel

4.2.1 Gestionnaire de projets SIMATIC

Le gestionnaire de projets SIMATIC gère toutes les données relatives à un projet d'automatisation quel que soit le système cible (S7/M7/C7) sur lequel elles ont été créés. Le gestionnaire des projets SIMATIC démarre automatiquement les applications requises pour le traitement des données sélectionnées.

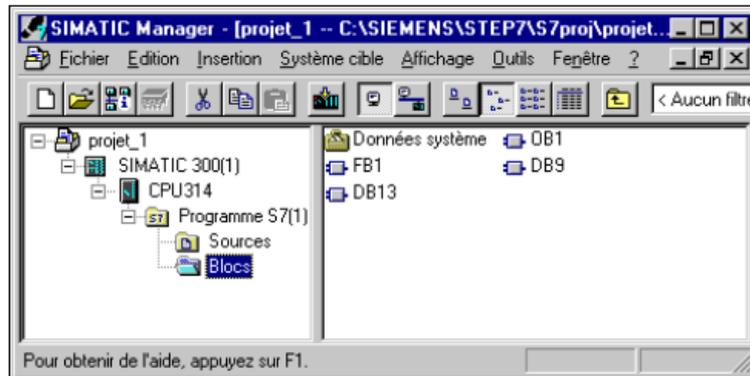


Figure 12 Le gestionnaire de projet STEP 7

4.2.2 Editeur des mnémoniques

L'éditeur des mnémoniques permet de gérer toutes les variables globales.

- Définition de désignations symboliques et de commentaires pour les signaux du processus (entrées/sorties), mémentos et blocs.
- Fonctions de tri.
- Importation/exportation avec d'autres programmes Windows.

La table des mnémoniques qui en résulte est mise à disposition de toutes les applications. La modification de l'un des paramètres d'une mnémonique est de ce fait reconnue automatiquement par toutes les applications.

4.2.3 Diagnostic du matériel

Le diagnostic du matériel fournit un aperçu de l'état du système d'automatisation. Dans une représentation d'ensemble, un symbole permet de préciser pour chaque module, s'il est défaillant ou pas. Un double clic sur le module défaillant permet d'afficher des informations détaillées sur le défaut.

4.2.4 Langages de programmation

Le langage de programmation du STEP 7 dispose de plusieurs modes de représentation, le programme peut être conçu sous forme de liste d'instruction puis converti en un autre mode de représentation. Les langages utilisés dans la programmation sont:

- **CONT**: le schéma à contacts s'apparente aux schémas de circuits électriques, il fait appel à des symboles, tels que les contacts et bobines.
- **LIST**: la liste d'instructions se compose d'opérations STEP7. Ce mode de représentation est plus particulièrement destiné aux programmeurs qui maîtrisent déjà d'autres langages de programmation.
- **LOG**: le programme est un langage de programmation graphique qui utilise des boîtes logiques. Le signe placé dans la boîte symbolise la fonction (& opération logique ET). Ce mode de représentation permet à des non programmeurs d'accéder à la programmation.
- **GRAPH**: Le langage de programmation graphique optionnel GRAPH permet de programmer des commandes séquentielles. Ceci implique la création d'une succession d'étapes, la définition des actions associées à chaque étape et celle des transitions indiquant les possibilités d'évolution entre deux étapes successives.

4.2.5 Paramétrage de l'interface PG-PC

Cet outil sert à paramétrer l'adresse locale des PG/PC, la vitesse de transmission dans le réseau MPI ou PROFIBUS en vue d'une communication avec l'automate et le transfert du projet.

4.3 Création du projet SIMATIC Step7 [8]

Un projet comprend deux données essentielles, les programmes et la configuration du matériel, le programme est l'interface graphique qui permet la manipulation du projet et l'accès aux autres programmes de STEP7. Pour en créer un nouveau, il suffit de cliquer sur le bouton «Nouveau projet», attribuer un nom et valider. Ensuite choisir une station de travail. Une station SIMATIC représente une configuration matérielle S7 comportant un ou plusieurs modules programmables. Il existe différents types:

- SIMATIC 400, SIMATIC 300, SIMATIC S5.
- SIMATIC H: Automate insensible aux défaillances, il se compose de 2 CPU du même type, en cas de problème elle commute de l'une vers l'autre sans perte de données.
- SIMATIC PC: Ou station PC, représente un PC contenant des composants
- SIMATIC: Des applications (WinCC, par ex.).
- Autres stations: se sont soit des appareils d'autres fabricants ou bien des stations de SIMATIC S7 contenus dans un autre projet.

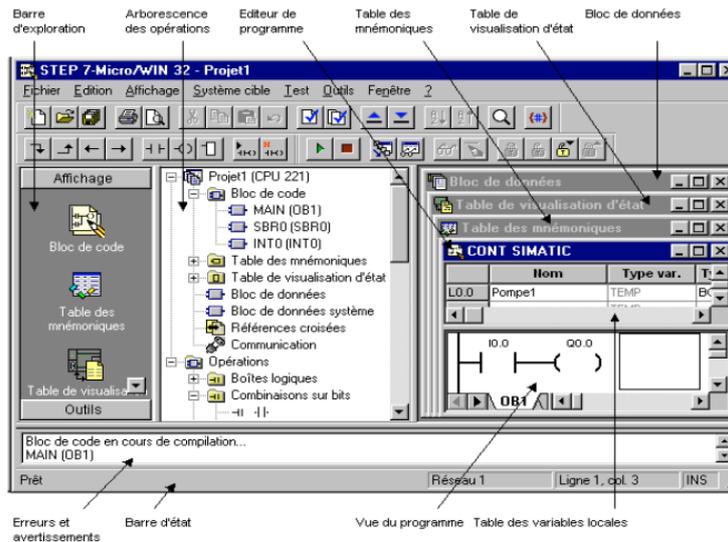


Figure 13 Fenêtre d'un nouveau projet STEP 7

4.3.1 Configuration du matériel

Pour configurer le matériel, il suffit de faire glisser des éléments du catalogue dans l'emplacement approprié, il suffit de choisir le Rack, l'alimentation, la CPU et les E/S... Nous distinguons:

- C7 : Système intégré compact qui regroupe automate programmable et interface homme machine (pupitre opérateur) pour la réalisation de commandes de machines sous encombrement réduit.
- CP : Communication Processor, module de communication (PROFIBUS, Industriel, Ethernet,...).
- FM : (Function Module), il regroupe les modules de fonctions (régulation, comptage...).
- IM : Coupleurs d'extension, il permet l'ajout d'autres modules.
- M7 : Modules d'extension et cartouches interface pour SIMATIC M7.
- PS : Module d'alimentation.
- Rack : Support mécanique.
- Routeur : Relie Industriel Ethernet à PROFIBUS.
- SM : Signal Module, c'est le module d'E/S, il contient le AI module d'entrées analogiques, le AO module de sorties analogiques, le DI module d'entrées TOR et le DO module de sorties TOR.
- CPU : L'Unité Centrale, noté CPU xxx a b, tel que xxx est la famille de la CPU et a, b sont les propriétés de la CPU (éléments additionnels, port de communication).

4.3.2 Edition des programmes

Dans la section 'bloc' du SIMATIC Manager, il existe par défaut le bloc d'organisation 'OB1' qui représente le programme cyclique. Il est possible de rajouter d'autres blocs à tout moment.

4.3.2.1 Blocs d'organisation

Les blocs d'organisation (OB) constituent l'interface entre le système d'exploitation et le programme utilisateur. Ils sont appelés par le système d'exploitation et gèrent le traitement du programme cyclique et des programmes déclenchés par alarmes, ainsi que le comportement à la mise en route de l'automate programmable et le traitement des erreurs.

4.3.2.2 Fonctions et blocs fonctionnels

Il est possible de programmer chaque bloc d'organisation en tant que programme structuré en créant des fonctions (FC) et des blocs fonctionnels (FB).

Les blocs fonctionnels (FB) sont des blocs de code associés à des blocs de données d'instance, dans lesquels sont sauvegardés les paramètres effectifs et les données statiques des blocs fonctionnels. Les fonctions (FC) sont des blocs de code sans rémanence, c'est-à-dire qu'ils ne sont pas associés à des blocs de données, les paramètres effectifs ne sont pas sauvegardés automatiquement.

4.3.2.3 Bloc de données

Les blocs de données (DB) servent à l'enregistrement de données utilisateur. Les blocs de données globales servent à l'enregistrement de données qui peuvent être utilisées par tous les autres blocs. Les blocs de données d'instances sont affectés à des blocs fonctionnels.

4.4 Simulation de modules Avec le logiciel PLCSIM

S7-PLCSIM dispose d'une interface simple permettant de visualiser et de forcer les différents paramètres utilisés par le programme (comme, par exemple, d'activer ou de désactiver des entrées).

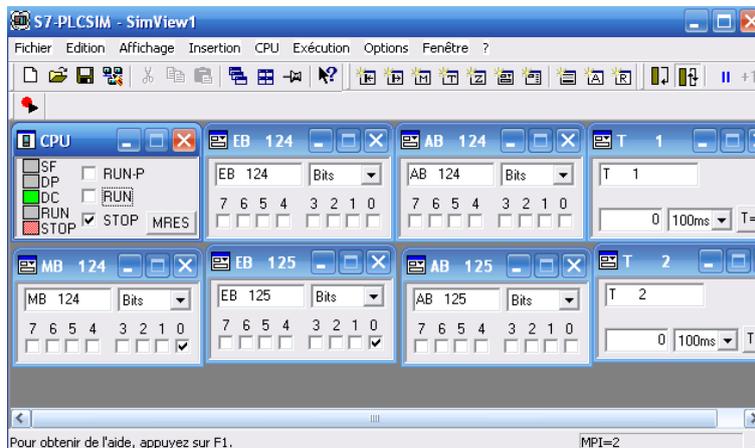


Figure 14 Logiciel de simulation PLC-SIM

4.5 Chargement du programme dans la CPU

Une fois la configuration, le paramétrage et la création du programme terminés, il est possible de transférer le programme utilisateur dans le système cible.

4.6 Surveillance du fonctionnement et diagnostic du matériel

La détermination des causes d'un défaut dans le déroulement d'un programme utilisateur se fait à l'aide de la « Mémoire tampon de diagnostic », accessible depuis le SIMATIC Manager.

5 Logiciel de supervision WinCC [9]

5.1 Présentation de WinCC

Un système IHM constitue l'interface entre l'homme (opérateur) et le processus (machine/Installation). Un système IHM se charge des tâches suivantes :

- Représentation du processus
- Commande du processus
- Vue des alarmes
- Archivage de valeurs processus et d'alarmes
- Gestion des paramètres de processus et de machine

5.2 Fonctionnement du WinCC

WinCC flexible est constitué de deux logiciels de base :

Le logiciel de configuration WinCC flexible Advanced qui permet de créer les configurations sous Windows à partir de l'ordinateur de configuration (PC ou pupitre opérateur). Et le logiciel de visualisation du processus WinCC flexible Runtime qui permet de faire fonctionner la configuration sous Windows et de visualiser le processus. WinCC flexible Runtime est également exécuté sur l'ordinateur de configuration pour tester et simuler le fichier projet compilé.

5.2.1 Logiciel de configuration (CS)

Après le démarrage de WinCC, l'écran affiche l'explorateur WinCC Explorer. ce dernier affiche la structure complète du projet et permet de gérer le projet.

5.2.2 Eléments de l'interface utilisateur de WinCC flexible

L'environnement de travail de WinCC flexible se compose de plusieurs éléments :

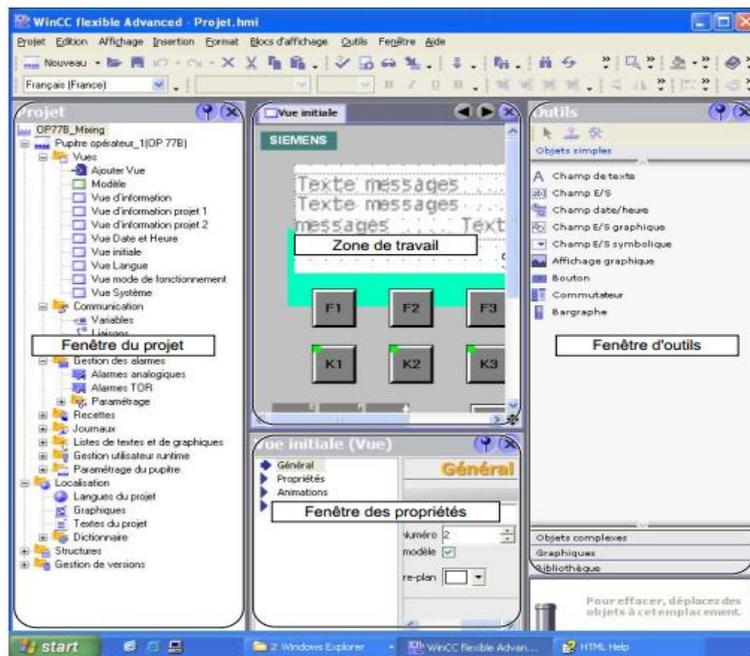


Figure 15 Eléments de l'interface WinCC

- Menus et barres d'outils
- Zone de travail
- Fenêtre de projet
- Fenêtre des propriétés
- Bibliothèque
- Fenêtre des erreurs et avertissements
- Fenêtre des objets

5.2.3 Logiciel WinCC flexible Runtime

WinCC flexible Runtime est conçu pour la visualisation et l'utilisation de machines et de petites installations. Il se distingue par son interface utilisateur entièrement graphique basée sur la technique des fenêtres. Il permet grâce à des temps de réaction rapides une conduite de processus sûre, ainsi qu'une collecte sûre des données.

5.3 Création d'un projet WinCC

Après avoir lancé le SIMATIC WinCC flexible, « Créer un projet vide » ; une fenêtre de dialogue affiche les différents types de pupitre à choisir. 'WinCC flexible Runtime' dans le cas d'une supervision via un PC.

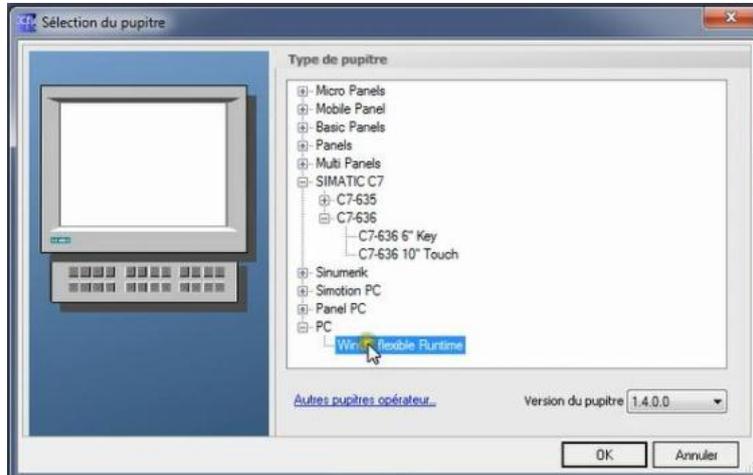


Figure 16 Configuration du PC comme pupitre

5.3.1 Communication WinCC/STEP 7

Il est possible d'établir la communication entre WinCC et le projet STEP 7 déjà créer, par une liaison de type 'MPI' comme le montre la figure 17.

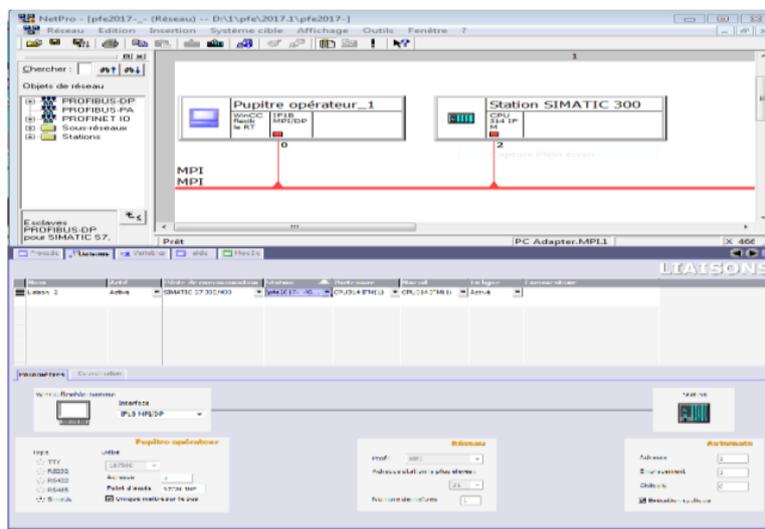


Figure 17 Liaison MPI entre le pupitre opérateur et la station SIMATIC

5.3.2 Définition des variables

Les variables utilisées dans WinCC sont :

- **Variables externes:** en mémoire d'un API.
- **Variables internes:** en mémoire de WinCC, elles assurent les mêmes fonctionnalités qu'un API.

Nom	Liaison	Type de donn...	Mnémonique	Adresse
X4	Liaison_1	Bool	<indéfini>	M 124.4
X3	Liaison_1	Bool	<indéfini>	M 124.3
X2	Liaison_1	Bool	<indéfini>	M 124.2
X1	Liaison_1	Bool	<indéfini>	M 124.1
X0	Liaison_1	Bool	<indéfini>	M 124.0
V4	Liaison_1	Bool	<indéfini>	Q 125.6
V3	Liaison_1	Bool	<indéfini>	Q 125.4
V2	Liaison_1	Bool	<indéfini>	Q 125.2
V1	Liaison_1	Bool	<indéfini>	Q 125.0
reset_logiciel	Liaison_1	Bool	<indéfini>	M 124.6
reset	Liaison_1	Bool	<indéfini>	I 124.0
P	Liaison_1	Bool	<indéfini>	Q 124.5
marche_logiciel	Liaison_1	Bool	<indéfini>	M 124.5
marche	Liaison_1	Bool	<indéfini>	I 124.2
LB	Liaison_1	Bool	<indéfini>	I 125.0
LA	Liaison_1	Bool	<indéfini>	I 124.4
HB	Liaison_1	Bool	<indéfini>	I 125.2
HA	Liaison_1	Bool	<indéfini>	I 124.6
FVR	Liaison_1	Bool	<indéfini>	Q 124.2

Figure 18 Fenêtre des variables

5.3.3 Création d'une vue de process

Pour créer des vues, le logiciel dispose d'objets prédéfinis permettant de représenter notre installation, d'afficher des procédures et de définir des valeurs de processus.

5.3.4 Création des alarmes [10]

La procédure d'alarme détermine le type de l'information déclenchant une alarme. WinCC flexible supporte les procédures d'alarme suivantes :

- Procédure par bit: alarmes TOR.
- Procédure analogique: alarmes analogiques.
- Procédure par numéro d'alarme: L'automate transmet au pupitre un numéro d'alarme.

5.4 Test et simulation du projet

Après la création du projet, le test et la simulation s'effectue à l'aide de la fonction de contrôle de cohérence et du simulateur de WinCC flexible.

5.4.1 Test du projet

Pour tester un projet, il suffit de démarrer le contrôle de cohérence (compilateur) en haut de la barre d'outils, le résultat du contrôle de cohérence est affiché dans la fenêtre des erreurs et avertissements, si la compilation est terminée sans erreurs, il sera possible de passer à la simulation.

5.4.2 Simulation du projet

La simulation permet de détecter des erreurs logiques de configuration, par ex. des valeurs limites incorrectes. Il permet d'observer les états du système, ainsi que les déclenchements des alarmes.

5.5 Transfert du projet

Après la création de l'interface graphique et les variables, et afin de lire les valeurs du processus provenant de l'AS ou de les transmettre vers l'automate. Il suffit de procéder:

- À la vérification des paramètres de connexion par défaut.
- Au transfert du projet sur le pupitre ou dans la station PC de supervision

6 Commande du prototype de la station de pompage avec l'automate S7 300

6.1 Schéma de connexion des entrées et des sorties

Les figures 19 et 20 correspondent aux schémas technologiques avec le câblage correspondant à la table d'affectation du tableau 2.

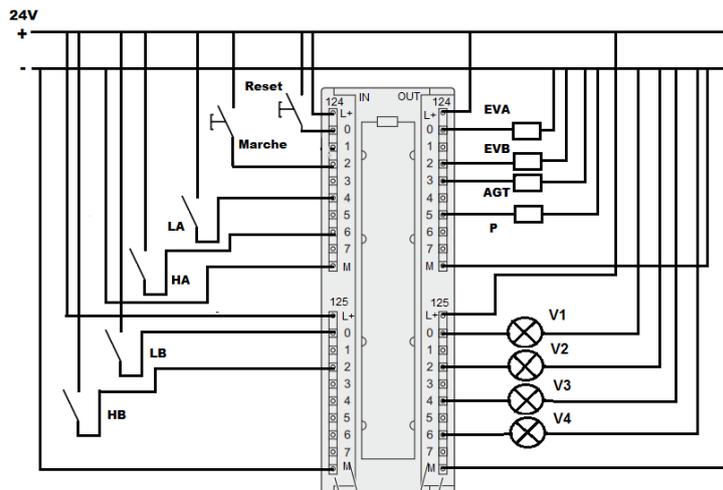


Figure 19 Schéma de connexion des entrées et des sorties TOR

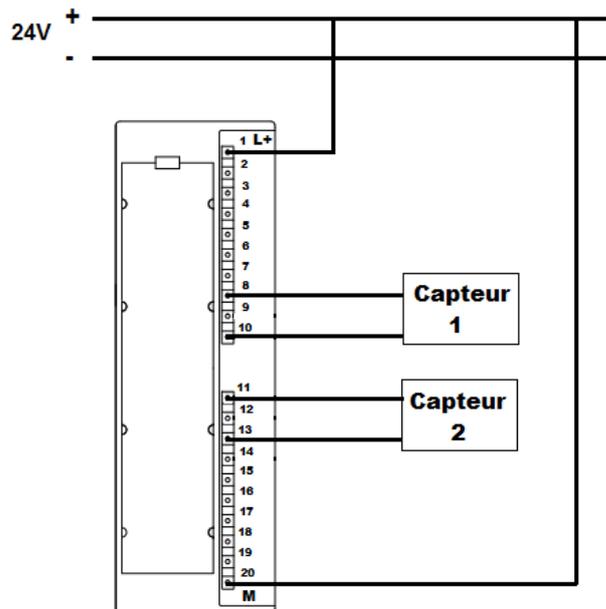


Figure 20 Schéma de connexion des entrées analogiques

Table d'afféctation :

Tableau 2 Table d'afféctation des entrées / sorties

Adresse	E TOR
124.0	Bouton reset
124.2	Bouton marche
124.4	Entrée LA
124.6	Entrée HA
125.0	Entrée LB
125.2	Entrée HB
Adresse	E ANA
128	Entrée analogique cuve A
130	Entrée analogique cuve B
Adresse	S TOR
124.0	Eva
124.2	Evb
124.3	Agt
124.5	Pompe
125.0	Voyant 1
125.2	Voyant 2
125.4	Voyant 3
125.6	Voyant 4

6.2 Développement du projet sous logiciel STEP 7 et logiciel

WinCC

Dans ce qui suit, nous allons créer un nouveau projet sous STEP 7, ou nous allons insérer une station SIMATIC300 en choisissant la CPU 314 IFM avec les caractéristiques citées précédemment, par la suite, nous allons configurer l'ensemble des paramètres adéquats à l'application, et ce pour l'étude et le contrôle de la station de pompage, pour finir nous allons insérer une station SIMATIC PC pour la supervision du processus. Cela en mettant en évidence les cahiers de charge du 1^{er} chapitre.

6.2.1 1ère manipulation -partie TOR-

6.2.1.1 Développement du programme sous logiciel STEP 7

Avant d'éditer le programme, nous allons tout d'abord définir les adresses des entrées et des sorties sous STEP 7, puis les introduire dans la table des mnémoniques comme le montre la figure 21.

Programme S7(1) (Mnémoniques) -- projet10pfe\Station SIMATIC 300\CPU 314IFM					
	Etat	Mnémonique	Opérande	Type de d	Commentaire
1		AGT	A 124.3	BOOL	agitateur
2		Cycle Execution	OB 1	OB 1	
3		EVA	A 124.0	BOOL	electrovanne de cuve A
4		EVB	A 124.2	BOOL	electrovanne de cuve B
5		HA	E 124.6	BOOL	niveau haut de la cuve A
6		HB	E 125.2	BOOL	niveau haut de la cuve B
7		LA	E 124.4	BOOL	niveau bas de la cuve A
8		LB	E 125.0	BOOL	niveau bas de la cuve B
9		marche	E 124.2	BOOL	bouton marche
10		P	A 124.5	BOOL	pompe
11		reset	E 124.0	BOOL	bouton reset
12		t1	M 128.0	BOOL	timer 1= 5s
13		t2	M 128.1	BOOL	timer2= 5s
14		V1	A 125.0	BOOL	voyant de la cuve A
15		V2	A 125.2	BOOL	voyant de la cuve B
16		V3	A 125.4	BOOL	voyant de la pompe
17		V4	A 125.6	BOOL	voyant de l'agitateur
18		X0	M 124.0	BOOL	
19		X1	M 124.1	BOOL	
20		X2	M 124.2	BOOL	
21		X3	M 124.3	BOOL	
22		X4	M 124.4	BOOL	
23		X5	M 124.5	BOOL	
24		X6	M 124.6	BOOL	

Figure 21 Table des mnémoniques partie TOR

Le programme de la 1ère manipulation -partie TOR- développé avec le langage LADDER est présenté dans l'annexe [C].

6.2.1.2 Supervision sous logiciel WinCC

Après avoir établi la liaison entre l'automate et la station SIMATIC HMI, nous devons introduire les variables des entrées et des sorties dans la table des mnémoniques du logiciel WinCC Flexible, comme le montre la figure 22.

Nom	Liaison	Type de donn...	Mnémonique	Adresse
X4	Liaison_1	Bool	<indéfini>	M 124.4
X3	Liaison_1	Bool	<indéfini>	M 124.3
X2	Liaison_1	Bool	<indéfini>	M 124.2
X1	Liaison_1	Bool	<indéfini>	M 124.1
X0	Liaison_1	Bool	<indéfini>	M 124.0
V4	Liaison_1	Bool	<indéfini>	Q 125.6
V3	Liaison_1	Bool	<indéfini>	Q 125.4
V2	Liaison_1	Bool	<indéfini>	Q 125.2
V1	Liaison_1	Bool	<indéfini>	Q 125.0
reset	Liaison_1	Bool	<indéfini>	I 124.0
P	Liaison_1	Bool	<indéfini>	Q 124.5
marche	Liaison_1	Bool	<indéfini>	I 124.2
LB	Liaison_1	Bool	<indéfini>	I 125.0
LA	Liaison_1	Bool	<indéfini>	I 124.4
HB	Liaison_1	Bool	<indéfini>	I 125.2
HA	Liaison_1	Bool	<indéfini>	I 124.6
EVB	Liaison_1	Bool	<indéfini>	Q 124.2
EVA	Liaison_1	Bool	<indéfini>	Q 124.0
AGT	Liaison_1	Bool	<indéfini>	Q 124.3

Figure 22 Table des variables de la manipulation TOR sur WinCC Flexible

- **Vues de la première manipulation :**

Dans la première manipulation TOR, nous allons introduire uniquement deux vues, celle de démarrage et celle de la maquette.



Figure 23 Vue de démarrage

la figure 24 représente la vue de la maquette, les deux entrées reset et marche sont configurées par des boutons, les états de niveaux d'eau dans les cuves A et B (Ha et La pour la cuve A, et Hb et Lb pour la cuve B) sont configurés chacun par un voyant qui s'allume lorsque le niveau est atteint. Tandis que les sorties telles que la pompe, les électrovannes et l'agitateur clignotent automatiquement lorsqu'elles sont en état de marche.

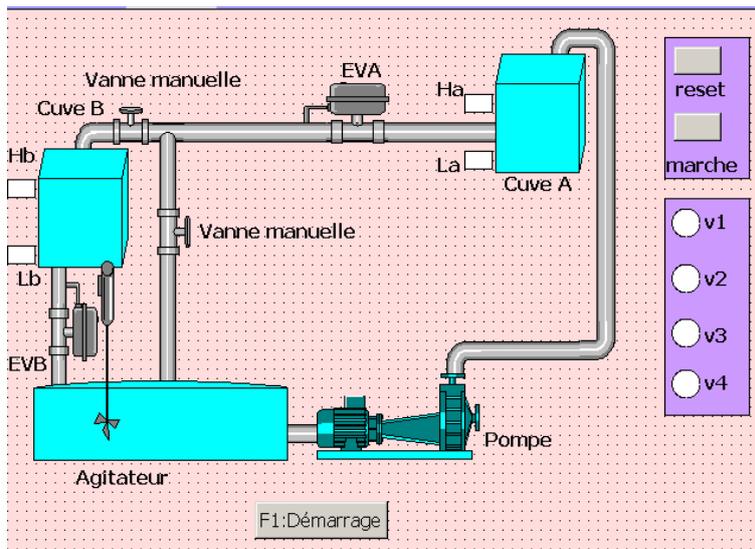


Figure 24 Vue de la maquette de la première manipulation TOR

- **Simulation de la première manipulation avec le simulateur WinCC Runtime:**

La figure 25 montre l'état de l'installation à la troisième étape du grafset, les entrées : marche, Ha, Hb sont activées, le voyant v1 est allumé et l'électrovanne EVA clignote.

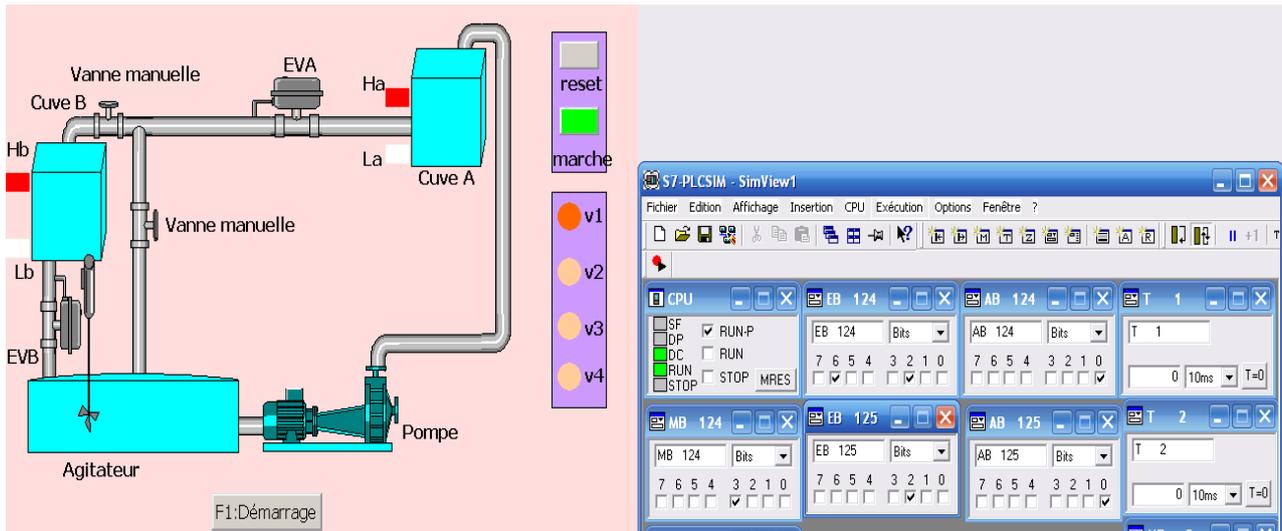


Figure 25 Simulation de la première manipulation TOR

6.2.2 2ème manipulation -partie analogique-

6.2.2.1 Développement du programme sous logiciel step 7

Nous allons dans cette partie introduire la fonction « Scale » (FC105), pour les deux entrées analogiques qui représentent les deux niveaux d'eau 'D1' et 'D2', respectivement des cuves A et B, afin de traduire les valeurs analogiques dans l'intervalle [0 - 10V] obtenues en valeurs réelles, et ce pour réaliser le cahier de charge de cette manipulation. L'adressage des variables est défini comme suit :

Tableau 3 Adresses des variables utilisées par les fonctions « scale »

Adresse	Type de variable	information représentée
PEW128	INT	Tension image du niveau d'eau dans la cuve A donnée par le capteur 1.
MD20	Real	D1 : Niveau d'eau dans la cuve A donnée par le capteur 1.
PEW130	INT	Tension image du niveau d'eau dans la cuve B donnée par le capteur 2.
MD24	Real	D2 : Niveau d'eau dans la cuve B donnée par le capteur 2.

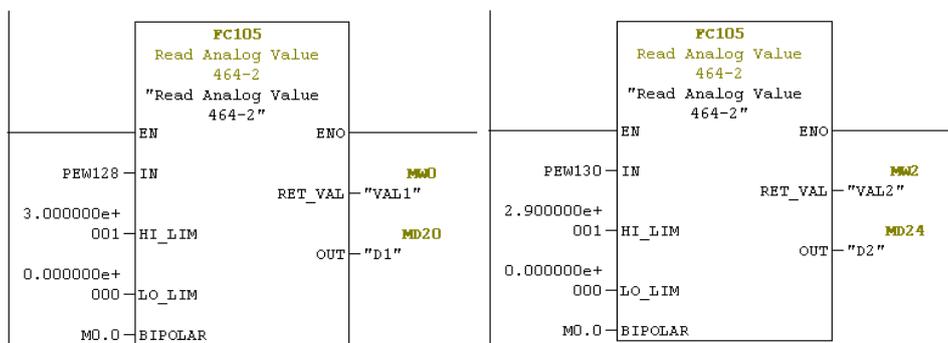


Figure 26 Fonction scale appelée dans le bloc OB 1 pour les deux entrées analogiques

Table des mnémoniques :

	Etat	Mnémonique	Opérande	Type de d	Commentaire
1		AGT	A 124.3	BOOL	agitateur
2		Cycle Execution	OB 1	OB 1	
3		D1	MD 20	REAL	niveau d'eau dans la cuve A
4		d1<19cm	M 1.1	BOOL	
5		d1<7cm	M 1.2	BOOL	
6		d1>20cm	M 1.0	BOOL	
7		D2	MD 24	REAL	niveau d'eau dans la cuve B
8		d2<17cm	M 1.3	BOOL	
9		d2<5cm	M 1.4	BOOL	
10		d2>18cm	M 1.5	BOOL	
11		EVA	A 124.0	BOOL	electrovanne de cuve A
12		EVB	A 124.2	BOOL	electrovanne de cuve B
13		HA	E 124.6	BOOL	niveau haut de la cuve A
14		HB	E 125.2	BOOL	niveau haut de la cuve B
15		LA	E 124.4	BOOL	niveau bas de la cuve A
16		LB	E 125.0	BOOL	niveau bas de la cuve B
17		marche	E 124.2	BOOL	bouton marche
18		P	A 124.5	BOOL	pompe
19		Read Analog Valu...	FC 105	FC 105	Read Analog Value 464-2
20		reset	E 124.0	BOOL	bouton reset
21		t1	M 128.0	BOOL	timer 1= 10s
22		t2	M 128.1	BOOL	timer2=10s
23		UNSCALE	FC 106	FC 106	Unscaling Values
24		V1	A 125.0	BOOL	voyant de la cuve A
25		V2	A 125.2	BOOL	voyant de la cuve B
26		V3	A 125.4	BOOL	voyant de la pompe
27		V4	A 125.6	BOOL	voyant de l'agitateur

Figure 27 Table des mnémoniques partie analogique

Le programme de la manipulation 2 -partie analogique- développé avec le langage LADDER est présenté dans l'annexe [C].

6.2.2.2 Supervision sous logiciel WinCC

La figure 28 représente la table des variables des entrées et des sorties TOR ainsi que les entrées analogiques D1 et D2 dans la table des mnémoniques.

Nom	Liaison	Type de donn...	Mnémonique	Adresse
X4	Liaison_1	Bool	<indéfini>	M 124.4
X3	Liaison_1	Bool	<indéfini>	M 124.3
X2	Liaison_1	Bool	<indéfini>	M 124.2
X1	Liaison_1	Bool	<indéfini>	M 124.1
X0	Liaison_1	Bool	<indéfini>	M 124.0
V4	Liaison_1	Bool	<indéfini>	Q 125.6
V3	Liaison_1	Bool	<indéfini>	Q 125.4
V2	Liaison_1	Bool	<indéfini>	Q 125.2
V1	Liaison_1	Bool	<indéfini>	Q 125.0
reset	Liaison_1	Bool	<indéfini>	I 124.0
P	Liaison_1	Bool	<indéfini>	Q 124.5
marche	Liaison_1	Bool	<indéfini>	I 124.2
EVB	Liaison_1	Bool	<indéfini>	Q 124.2
EVA	Liaison_1	Bool	<indéfini>	Q 124.0
d2	Liaison_1	Real	<indéfini>	MD 24
d1	Liaison_1	Real	<indéfini>	MD 20
AGT	Liaison_1	Bool	<indéfini>	Q 124.3

Figure 28 Table des variables de la manipulation analogique

Nous allons présenter dans cette manipulation 3 vues, la première représente la vue de démarrage. La deuxième représente la maquette incluant, les deux entrées analogique D1 et D2 (variations d'eau dans les cuves A et B) qui sont représentées par deux capteurs, les sorties telles que la pompe les électrovannes et l'agitateur qui clignent lorsqu'elles sont en état de marche, ainsi que

les voyants V1, V2, V3 et V4 associés à chacune d'entre elles. La troisième représente la vue d'alarme.

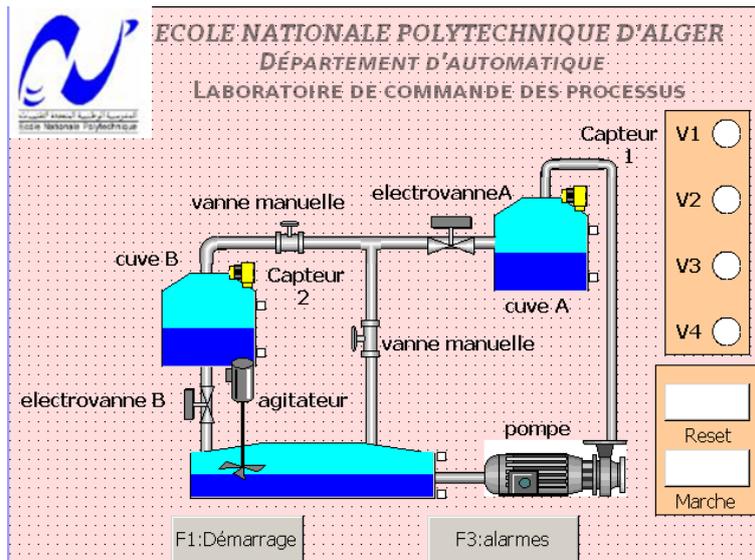


Figure 29 Vue de la maquette de la manipulation analogique

Dans la troisième vue de la manipulation analogique, nous avons configuré des alarmes analogiques pour les niveaux d'eau :

- Deux alarmes qui indiquent le dépassement du niveau max d'eau dans chaque cuves (D1 ou D2), les alarmes s'activent si le niveau d'eau est supérieur à 90% de la valeur maximale de chaque cuve.
- Deux alarmes qui indiquent que le niveau d'eau dans chaque cuves (D1 ou D2) est inférieur au niveau minimal de la cuve, les alarmes s'activent si le niveau d'eau est inférieur à 10% de la valeur minimale de chaque cuve.

La figure 30 présente la configuration des quatre alarmes analogiques :

ALARMES ANALOGIQUES						
Texte	Numéro	Classe	Variable surveil...	Valeur limite s...	Déclencheur	
niveau d1 haut :	1	Erreurs	d1	90	Si front montant	
niveau d1 bas :	2	Erreurs	d1	10	Si front descendant	
niveau d2 haut :	3	Erreurs	d2	90	Si front montant	
niveau d3 bas :	4	Erreurs	d2	10	Si front descendant	

Figure 30 Table des alarmes analogiques

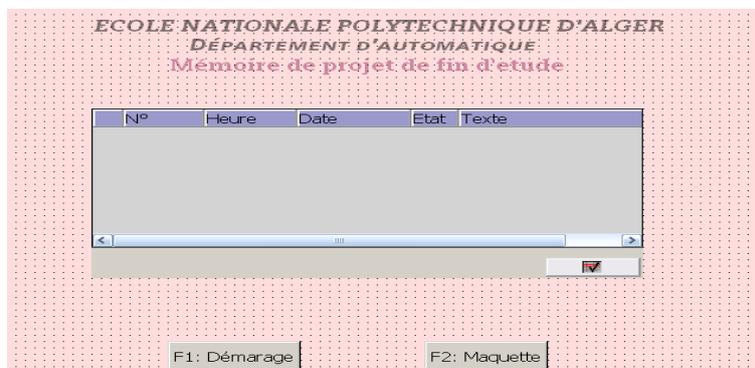


Figure 31 vue des alarmes

- **Simulation de la deuxième manipulation avec le simulateur WinCC Runtime :**

La figure 32 montre l'état de l'installation à la 4ème étape du grafcet ou $d1 < 7\text{cm}$, l'électrovanne B clignote et le voyant V2 est allumé.

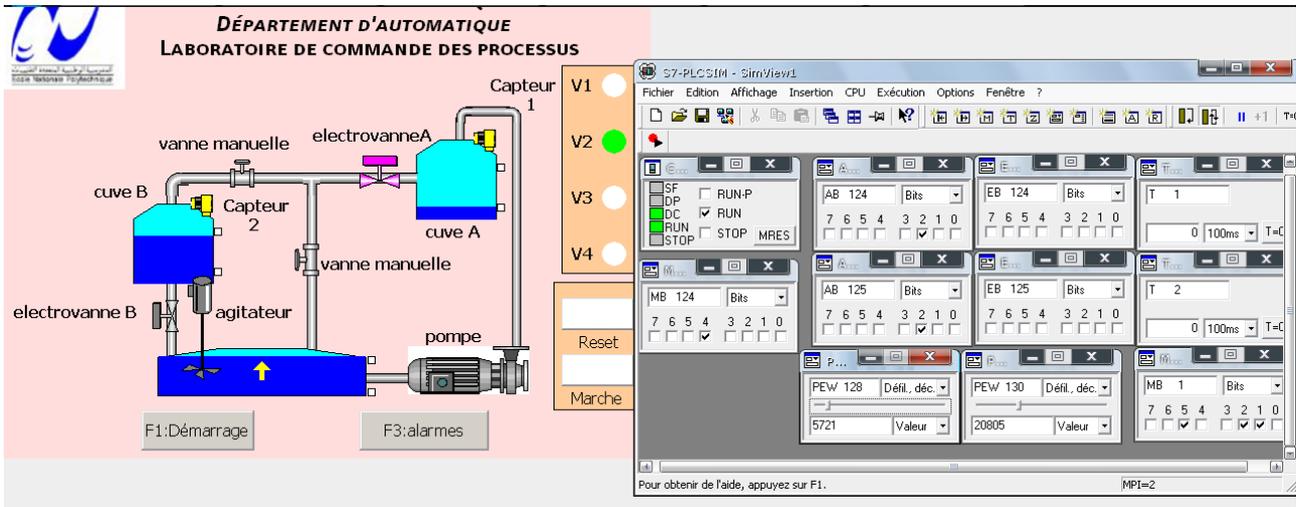


Figure 32 Simulation de la deuxième manipulation analogique

7 Conclusion

Ce chapitre nous a permis de connaître les fonctionnalités des logiciels Step7 et WinCC Flexible et de voir toutes les étapes nécessaires à la création des programmes et la supervision de la station.

La simulation avec le simulateur S7-PLCSIM de Step7 ainsi que l'interface graphique notamment faite sur le simulateur WinCC Flexible Runtime de WinCC Flexible a montré que la liaison entre le programme de la station et l'interface Homme-Machine (IHM) a bien été réalisée. Cette solution proposée permet d'assurer un suivi en temps réel du fonctionnement de la station et du niveau d'eau, ainsi qu'un diagnostic des dépassements ou insuffisances d'eau.

Chapitre III

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

1 Introduction

Les automates offerts par la filiale télémécanique Schneider Electric dispose de plusieurs ateliers logiciels pour la conception d'automatisme parmi lesquelles, nous avons ZELIO soft, PL7 pro, Unity Pro pour la programmation des automates et Vijeo Designer, Monitor Pro pour les systèmes de supervision, ces logiciels de conception permettent la programmation et la simulation de procédures automatisées avec différents langages normalisés. [11]

Dans ce chapitre, nous allons présenter le module programmable ZELIO logic et Le logiciel de programmation ZELIO Soft, par la suite nous allons commander cette station en utilisant l'automate ZELIO (SR2 B201BD) tout en appliquant les cahiers de charges développés au premier chapitre.

2 Les modules programmables Zelio logic

2.1 Présentation

Les modules logiques Zelio logic sont destinés à la réalisation de petits équipements d'automatismes. Ils sont utilisés dans les secteurs d'activité de l'industrie.

La simplicité de leur programmation, garantie par l'universalité des langages LADDER et blocs fonctions FBD, satisfait aux exigences de l'automatisme et répond aux attentes de l'électricien. Les modules logiques compacts répondent aux besoins d'automatismes simples, jusqu'à 20 entrées/sorties. Les modules logiques modulaires autorisent, si besoin, des extensions d'entrées/ sorties et une extension de communication sur réseau Ethernet, pour plus de performance et de flexibilité, de 10 à 40 entrées/sorties. [12]

2.2 Programmation [12]

La programmation peut être effectuée :

- De façon autonome en utilisant le clavier du module logique (langage à contacts)
- Sur PC avec le logiciel "Zelio Soft", la programmation peut être réalisée soit en langage à contacts (LADDER), soit en langage blocs fonctions (FBD).

2.3 Modules logiques compacts/modulaires [12]

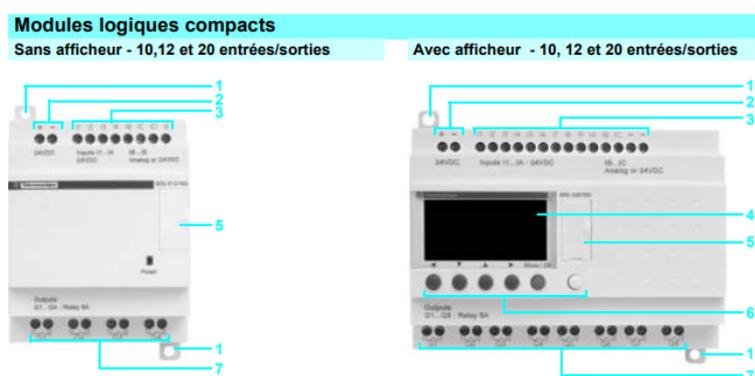


Figure 33 Modules logiques compacts

Les modules logiques compacts et Les modules logiques modulaires comprennent en face avant :

- Deux pattes de fixation rétractables
- Deux bornes d'alimentation
- Des bornes de raccordement des entrées
- Un afficheur LCD rétro éclairé de 4 lignes de 18 caractères
- Un emplacement pour cartouche mémoire et raccordement au PC
- Un clavier de 6 touches pour la programmation et le paramétrage
- Des bornes de raccordement des sorties.

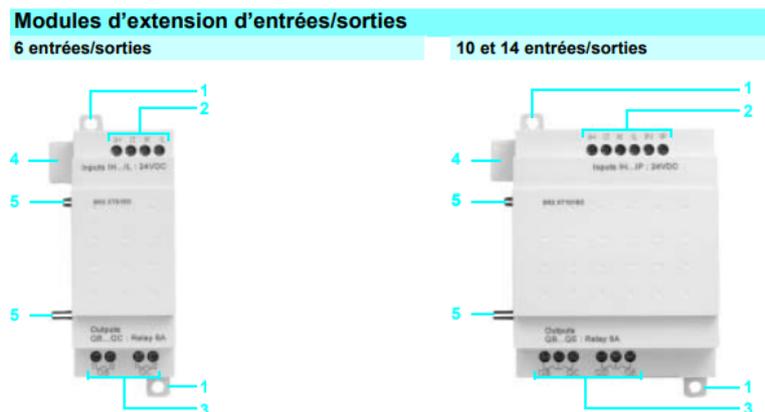


Figure 34 Modules d'extension d'entrées/sorties

Les modules d'extension d'entrées/sorties comprennent en face avant :

- Deux pattes de fixation rétractables
- Des bornes de raccordement des entrées
- Des bornes de raccordement des sorties
- Un connecteur pour raccordement au module logique (alimentation fournie par le module logique)
- Des pions de détrempage.

2.4 Module d'extension de communication

Extensions de communication réseau Modbus ou Ethernet, alimentées en c 24 V. [12]

2.5 Communication [12]

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

2.6 Programmation sur écran [2]

Il est possible de programmer directement sur écran sans passer par le logiciel Zelio Soft, puisque le module Zelio procède un afficheur LCD sur la face avant ainsi que des boutons pour la manipulation, mais cela s'effectue uniquement en langage LADDER.

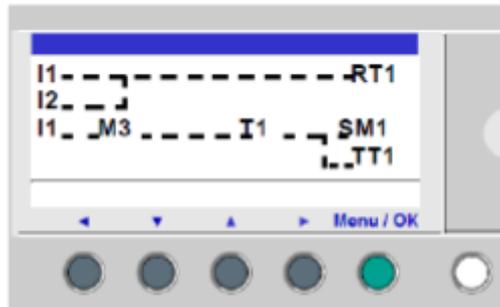
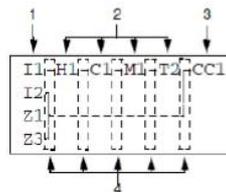


Figure 35 Afficheur de l'automate Zelio

Le module logique permet la saisie de 120 lignes de schémas de commande, tout comme sur le logiciel Zelio Soft. 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.

Règles	Incorrect	Correct
Chaque bobine ne doit être saisie qu'une seule fois dans la colonne de droite.	<pre> I1-----[Q1 I2-I3-----TT1 T1-----[Q1 Z1● </pre>	<pre> I1]-----[Q1 T1]-----TT1 I2-I3-----TT1 Z1● </pre>
Les éléments utilisés comme contacts peuvent être saisis autant de fois que nécessaire dans les 5 colonnes de gauche.		<pre> I1-----TT1 T1-----[Q1 I3-T1-----[M2 I1-M2-----[Q2 </pre>
Les liaisons doivent toujours aller de gauche à droite.	<pre> I1-I2-I3]]---[I4-I5-I6-[Q1 </pre>	<pre> I1-I2-I3-----[M1 M1----I4-I5-I6-[Q1 </pre>
Si dans un schéma, on utilise des bobines S (SET), utiliser aussi une bobine R (Reset).	Si on n'utilise pas de bobine R (Reset), la bobine correspondante sera toujours à 1.	Il est nécessaire d'utiliser une bobine R (Reset) pour la remettre à zéro.

Figure 36 Règles de saisie du schéma de commande

2.7 Verrouillage façade [2]

Pour verrouiller un programme sur Zelio logic, il suffit de cocher sur la case « mettre un verrou sur la face avant du module » une fois la fenêtre option écriture ouverte, sa fonction principale est d'interdire tout accès au menu, lorsque le programme est en marche mais aussi lorsqu'il est à l'arrêt. Les boutons de façade restent utilisables malgré le verrouillage.

3 Le logiciel de programmation Zelio Soft

3.1 Présentation

Le logiciel de programmation Zelio Soft est conçu pour programmer les modules logiques de la gamme Zelio Logic. Zelio Soft nous permet de choisir entre les langages de programmation, d'afficher les données du programme et des paramètres, de charger et télécharger des applications, ainsi que d'imprimer la documentation de l'application. [13]

Le logiciel “Zelio Soft” 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. [12]

3.2 Langages utilisés [13]

Le module logique propose 2 modes de programmation :

- Mode LD : langage à contacts
- Mode FBD : langage en blocs fonction

Ces langages mettent en œuvre :

Des blocs fonction prédéfinis :

- temporisateurs
- compteurs

Des fonctions spécifiques :

- gestion du temps
- chaîne de caractères
- communication...

3.2.1 Langage à contacts [13]

Le langage à contacts (LD) est un langage graphique, il permet la transcription de schémas à relais, il fournit des symboles graphiques : contacts, bobines et blocs. L'exécution de calculs spécifiques est possible à l'intérieur des blocs opérations

Le nombre maximum de lignes en langage à contacts:

- 120 lignes si une interface de communication SR2COM01 est sélectionnée dans la configuration.
- 240 lignes sans interface de communication SR2COM01.
- chaque ligne est composée de 5 contacts maximum et d'une bobine.

Structure d'un réseau de contacts:

- 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).

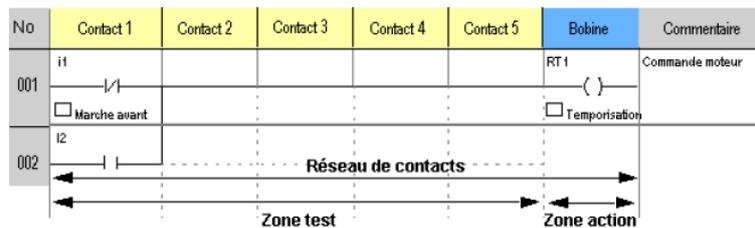


Figure 37 Structure d'un réseau de contacts

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 rétro éclairage programmable,
- passage automatique heure d'été/heure d'hiver,
- diversité des fonctions : bobine, à mémoire (Set/Reset), télérupteur, contacteur,
- 28 blocs messages.

3.2.2 Langage FBD [13]

Le mode FBD permet une programmation graphique basée sur l'utilisation de blocs fonction prédéfinis. Il offre une large gamme de fonctions : temporisateur, compteur, logique, etc..

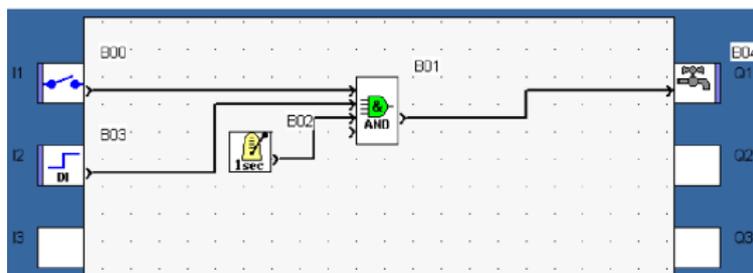


Figure 38 Exemple d'un programme en langage FBD

Fonctionnalités :

- 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

3.3 Test des programmes et et maintenance [13]

Deux modes de test sont proposés : simulation et monitoring.

Le mode simulation de "Zelio Soft" permet de tester l'ensemble des programmes sans module c'est à dire :

- activer les entrées "Tout ou Rien" (TOR).
- visualiser l'état des sorties.
- faire varier la tension des entrées analogiques.
- activer les touches de programmation.
- simuler le programme applicatif en temps réel ou en accéléré.
- visualiser en dynamique et en rouge les différents éléments actifs du programme.

Le mode monitoring de "Zelio Soft" permet de tester le programme exécuté par le module, c'est à dire :

- visualiser "en ligne" le programme,
- forcer les entrées, les sorties, les relais auxiliaires et les valeurs courantes des blocs fonctions,
- régler l'heure,
- passer du mode d'arrêt (STOP) au mode de marche (RUN) et inversement.

En mode simulation ou monitoring, la fenêtre de supervision permet de visualiser l'état des entrées/sorties du module dans l'environnement de votre application.

3.4 Transfert d'une application

3.4.1 Transfert du programme PC vers le Zelio Logic [2]

Une fois la simulation du programme faite, et suit exactement le cahier de charge donc le grafcet, il suffit de transférer le programme sur Zelio logic et pour cela il faut simplement choisir dans le menu **Transfert**, Transférer le programme puis cliquer sur **PC->Module**.

Il faut savoir qu'il est impossible de transférer le programme si le module connecté est en mode **RUN**, il faut donc le mettre en mode **STOP** en choisissant **Transfert** puis **STOP Module**.

Si le type de module sélectionné au préalable n'est pas le même que celui qui est connecté, il faudra changer le type en cliquant sur **Module** puis **choix du module/programmation**, pour effectuer un diagnostic du module connecté il suffit de cliquer sur **Module** puis **Diagnostic du Module**.

Une fois le module sélectionné et le module connecté s'accorde, la fenêtre de dialogue s'affiche comme suit :

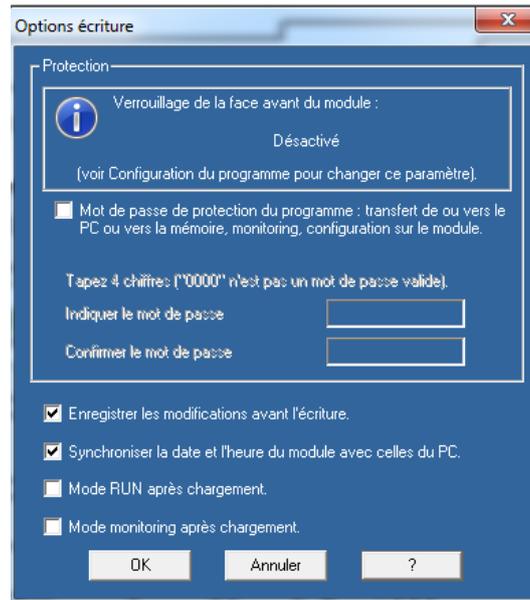


Figure 39 Fenêtre de dialogue d'un module zelio

Le programme présent dans le module avant le transfert est écrasé par le module. Pour mettre en marche le module à partir du logiciel, il suffit de cliquer sur **Transfert** puis **RUN Module**. Toutefois, lorsque le transfert du programme est en cours, le module se met en mode **RUN** automatiquement.

3.4.2 Transfert du programme Zelio Logic vers le PC [2]

La récupération d'un programme d'un module vers le pc est possible, en allant dans le menu, puis en cliquant sur **Transfert**, transférer programme puis **module->pc**.

Le logiciel charge alors le dernier programme présent dans le module. Si le programme du module est verrouillé, un code sera demandé par le module avant la récupération du programme.

3.5 Fonction mot de passe

Le mot de passe protège l'accès à un programme. Lorsque le transfert du programme dans Zelio logic est en cours, la fenêtre option d'écriture s'ouvre et il suffit de cocher la case « Protéger par mot de passe... ». Une fois le mot de passe activé, il sera impossible d'écrire dans le module ni lire le programme sans connaître ce mot de passe. Le programme est ainsi protégé. [2]

4 Commande du prototype de la station de pompage par l'automate Zelio Logic

4.1 Principe de fonctionnement

Le choix de l'automate s'est porté sur Schneider Electric (SR2 B201BD) Zelio logic, dont la tension de fonctionnement est de 24 V/DC, son rôle essentiel est de gérer les actionneurs tels que la pompe, les vannes motorisées et l'agitateur. Il assurera la sécurité de l'installation afin d'éviter toutes fausses manipulations de la part de l'opérateur, mais aussi il permettra la communication entre les équipements et le superviseur.



Figure 40 Automate programmable ZELIO (SR2 B201BD)

4.2 Schéma de connexion des entrées et des sorties

Les figures 41 et 42 correspondent aux schémas technologiques avec le câblage correspondant aux cahiers de charge.

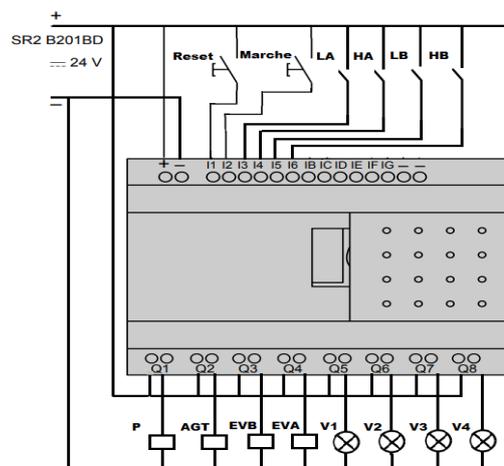


Figure 41 Schéma de connexion des entrées et des sorties TOR

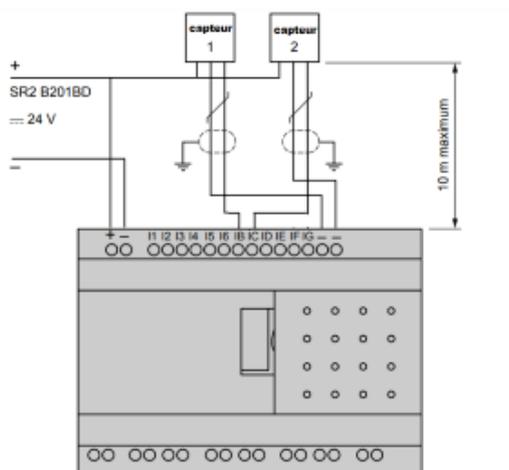


Figure 42 Schéma de connexion des entrées analogiques

Table d'affectation :

Tableau 4 tableau d'affectation des entrées / sorties

Adresse	E TOR
I1	Bouton reset
I2	Bouton marche
I3	Entrée LA
I4	Entrée HA
I5	Entrée LB
I6	Entrée HB
Adresse	E ANA
IB	Entrée analogique cuve A
IC	Entrée analogique cuve B
Adresse	S TOR
Q1	Pompe
Q2	Agt
Q3	Evb
Q4	Eva
Q5	Voyant 1
Q6	Voyant 2
Q7	Voyant 3
Q8	Voyant 4

4.3 Développement du projet sous logiciel Zelio soft

En premier lieu nous avons pu se familiariser avec l'automate et le programmer directement sur son écran. La figure 43 représente l'activation des actionneurs « pompe, agitateur, électrovannes B et A », respectivement les sorties Q1, Q2, Q3 et Q4 en appuyant sur le bouton marche à l'entrée I2.



Figure 43 Programmation sur écran avec le langage LADDER

Ensuite nous avons dû installer le logiciel Zelio Soft, ainsi que le pilote du câble de communication. Après l'édition du programme (FBD) et la simulation des deux manipulations, nous avons effectué le transfert vers l'automate, et testé les deux manipulations (TOR/Analogique) en mode MONITORING sur la maquette de station de pompage, qui a répondu correctement aux deux cahiers des charges proposés dans du premier chapitre.

Aussi, la commande par le langage ladder a été effectuée en annexe [C]

Le programme de la manipulation 1 partie TOR, ainsi que la manipulation 2 partie analogique effectuée avec les langages FBD et LADDER est développée dans l'annexe [C].

5 Conclusion

De part la mobilité, la flexibilité de son architecture, la facilité de sa programmation, l'automate programmable ZELIO forme une unité de commande flexible, que nous avons pu exploiter d'une manière optimale grâce au logiciel de programmation Zelio Soft qui intègre les langages et les outils nécessaires pour une programmation complète et souple. Il offre la possibilité et la facilité de commander de petits équipements d'automatisme comme ceux de la station de pompage au sein du laboratoire d'automatique.

Chapitre IV

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

1 Introduction

Dans ce chapitre nous allons réaliser le contrôle de la maquette de la station de pompage à l'aide d'un tout autre automate jamais utilisé au préalable au laboratoire d'automatique qui est le LOGO!, nous y présenterons dans un premier temps ce module ainsi que ses fonctionnalités dans le domaine de l'informatique industrielle. Par la suite, nous aborderons la description de l'environnement de développement LOGO! Soft Comfort ainsi que les différents langages de programmation qui existent dans ce logiciel, nous verrons les différentes méthodes d'affichages pour la supervision du prototype via un écran d'affichage LOGO! TDE et via un navigateur Web. Et enfin nous traiterons les exemples de cahier de charge afin d'illustrer les différentes étapes de gestion d'un projet.

2 Présentation

LOGO! est un automate logique universel de Siemens intégrant une unité de commande et d'affichage. Cette unité de commande et d'affichage de LOGO! Permet de créer et modifier des programmes et de commander des fonctions système. Cet automate propose un apprentissage rapide permettant de manipuler des modules logiques 0BA8 LOGO! Et d'effectuer des programmations à l'aide du logiciel LOGO! Soft Comfort. [14]



Figure 44 Automate LOGO! Et un afficheur LOGO! TDE

3 Fonctionnalité du LOGO! [15]

Le module logique universel de Siemens intègre les éléments suivants:

- Commande
- Unité de commande et d'affichage avec rétro éclairage
- Alimentation
- Interface pour modules d'extension
- Interface pour une micro carte SD
- Interface pour un module afficheur de textes (TDE) optionnel
- Entrées et sorties selon le type d'appareil LOGO!

LOGO ! 0BA8 intègre en plus les composants suivants :

- Interfaces pour communication Ethernet
- Borne FE (Functional Earth, terre fonctionnelle) pour le raccordement à la terre
- Une DEL de signalisation d'état pour la communication Ethernet

4 Les types d'appareils [15]

4.1 Les classes de tension du module LOGO! Base

- Classe 1 ≤ 24 V, c'est-à-dire 12 V CC, 24 V CC, 24 V CA
- Classe 2 > 24 V, c'est-à-dire 115 V CA/CC à 240 V CA/CC

4.2 Les versions du module LOGO! Base

- LOGO! Basic (avec écran) : 8 entrées et 4 sorties
- LOGO! Pure (sans écran) : 8 entrées et 4 sorties

Chaque module est équipé d'une interface d'extension et d'une interface Ethernet ; il fournit 44 blocs fonctionnels spéciaux et standards préconfigurés pour la création du programme de commande.

4.3 Les modules d'extension

- Les modules TOR LOGO!DM8 sont disponibles pour une exploitation 12 V CC, 24 V CA/CC et 115 V CA/CC à 240 V CA/CC, et sont équipés de quatre entrées et de quatre sorties.
- Les modules TOR LOGO! DM16 sont disponibles pour une exploitation 24 V CC et 115 V CA/CC à 240 V CA/CC, et sont équipés de huit entrées et de huit sorties.
- Des modules analogiques LOGO! Sont disponibles pour une exploitation 24 V CC et d'autres pour une exploitation 12V CC selon le module spécifique. Chaque module dispose de 2 entrées analogiques, 2 entrées Pt100, 2 entrées PT100/PT1000 (PT100 ou PT1000 ou une de chaque) ou 2 sorties analogiques. Chaque module TOR/analogique dispose de 2 interfaces d'extension et peut ainsi être raccordé à d'autres modules.

4.4 Les modules d'affichage

- LOGO! Basic : le LOGO ! De base est un module avec afficheur intégré.
- LOGO! TDE : le LOGO TDE est un afficheur externe qui se raccorde avec le module de base LOGO !

5 Présentation du LOGO! TDE

LOGO! TDE est disponible pour la gamme 0BA8. Il fournit un écran supplémentaire qui est plus large que celui de LOGO! Basic. Il dispose de 4 touches de fonction que nous pouvons configurer comme entrées dans notre programme de commande. Comme LOGO! Basic, il dispose de 4 touches de curseur, d'une touche ESC et d'une touche Enter que nous pouvons également configurer dans notre programme de commande et utiliser pour naviguer dans LOGO! TDE. À l'aide de ce dernier nous pouvons créer et télécharger un écran de démarrage pour le LOGO! TDE à partir de LOGO! Soft Comfort. Cet écran s'affiche

brièvement à la mise sous tension initiale du LOGO! TDE. Nous pouvons également charger l'écran de démarrage du LOGO! TDE vers LOGO! Soft Comfort. [15]

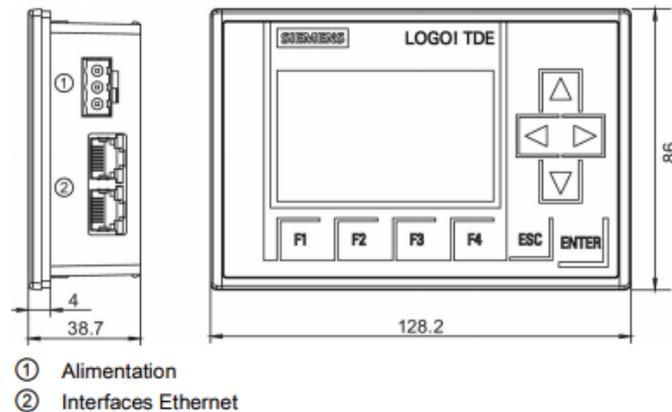


Figure 45 Afficheur LOGO! TDE

5.1 Fonctionnalité du LOGO! TDE

Le module LOGO! TDE est disponible avec deux interfaces Ethernet. Ces deux interfaces Ethernet peuvent également fonctionner comme commutateur à deux ports. Il est possible de connecter le LOGO! TDE à un module Base, à un PC ou à un autre LOGO! TDE via les interfaces Ethernet. LOGO! TDE peut être raccordé à différents modules Base à l'aide de la sélection d'adresse IP.

Le LOGO! TDE est équipé d'un connecteur à 3 broches (P1, P2 et FE) pour la connexion à l'alimentation.

Le LOGO! TDE dispose de trois commandes dans son menu principal, pour la sélection de l'adresse IP d'un module Base, la configuration distante du module Base connecté et la configuration indépendante du module LOGO! TDE. [15]

6 Programmation sous LOGO! Soft Comfort [14]

Le logiciel de programmation adéquat à l'automate LOGO ! De Siemens est le LOGO! Soft Comfort ce dernier permet de lire, via une interface Ethernet ou un câble PC, des programmes externes issus d'un module de programmes. LOGO! Soft Comfort permet, outre l'élaboration de programmes, d'effectuer une simulation de circuit sur ordinateur ou d'imprimer des plans d'ensemble.

6.1 Fonctionnalité du LOGO! Soft Comfort V8.0

Le logiciel présente une interface de commande totalement nouvelle. Elle propose les fonctions suivantes :

- Affichage cohérent des menus de l'application
- Nouveau concept de travail basé sur des projets de réseau
- Affichage partagé pour le mode diagramme et le mode réseau
- Affichage partagé pour la barre d'outils "Standard" dans l'interface principale du logiciel, la barre d'outils "Outil" est affichée dans le mode diagramme et la barre d'outils "Réseau" dans le mode projet

- Affichage de fenêtres partagées avec comme but commutation de vue et fonction copier-coller
- Le travail sur un projet de réseau permet l'enregistrement, le chargement, la création et la fermeture de ce projet de réseau
- Nouveaux paramètres de contrôle d'accès pour l'accès en ligne par différents moyens
- Possibilité de créer des connexions en configurant des blocs fonctionnels NI et NQ
- Nouvelle référence graphique pour le bloc fonctionnel dans le champ de paramètres des schémas LOG
- Possibilité de configurer l'affichage de l'écran pour les messages, l'écran d'accueil et les mémentos 6 lignes pour les appareils LOGO! à partir de la version 0BA8
- Sécurité système étendue grâce au paramétrage de mots de passe utilisateur et de niveaux d'accès via le réglage des contrôles d'accès

6.2 Interface de programmation

L'affichage initial du mode de programmation dans LOGO! Soft Comfort est constitué d'un diagramme vierge.

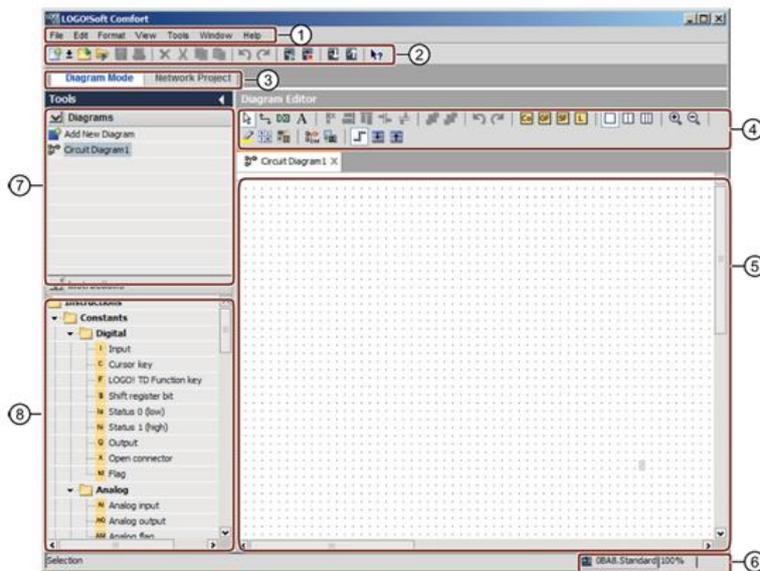


Figure 46 Interface vierge du logiciel logo ! Soft Comfort

- | | |
|------------------------------|-------------------------------|
| ① Barre de menus | ⑤ Interface de programmation |
| ② Barre d'outils "Standard" | ⑥ Barre d'état |
| ③ Barre de sélection du mode | ⑦ Arborescence du diagramme |
| ④ Barre d'outils "Outil" | ⑧ Arborescence des opérations |

Barre des menus : elle propose différentes instructions de commande pour éditer et gérer nos programmes de commande, cela inclut les paramétrages par défaut et les fonctions de transfert du programme de commande et vers LOGO.

Barre d'outils "Standard": Cette barre permet d'accéder directement aux principales fonctions de LOGO Soft Comfort.

Barre de sélection du mode : elle permet de distinguer entre le mode diagramme et le mode projet réseau.

Barre d'outils "Outil"7 : Ces icônes permettent de passer entre les différents modes d'édition afin de créer ou d'éditer facilement et rapidement un programme de commande.

Interface de programmation : c'est le champ de l'application à réaliser.

Barre d'état : Elle fournit les indications sur l'outil actif, l'état de programme, le facteur de zoom, la page du schéma de raccordement et le module LOGO.

Barre d'outils "Simulation" 8: contient comme le montre la figure 47 [1]

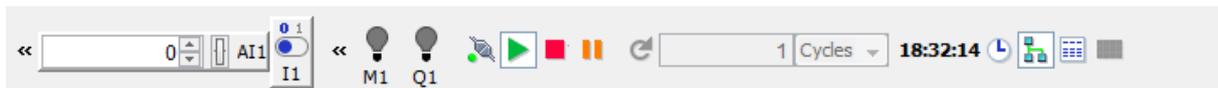


Figure 47 Barre d'outils de la commande « simulation »

- Des icônes (p. ex. commutateurs) de commande d'entrées.
- Une icône de simulation d'une panne de réseau, permettant de tester la rémanence du circuit en cas de défaillance de la tension du réseau,
- Des icônes (p. ex. voyants) de supervision de sorties.
- Des icônes de commande de la simulation.
- Des icônes de commande des temporisations.

6.3 Langage de programmation

LOGO ! Soft Conforts permet de créer des programmes de commande de deux manières différentes, soit sous forme de schéma à contacts (diagramme CONT) où sous forme de diagramme fonctionnel (diagramme de blocs fonctionnels, LOG). [8]

Nous retrouvons donc le même principe que dans les autres automates notamment l'automate Zelio de Schneider.

6.4 Interface de projet

Dans l'interface de projet, LOGO! Soft Comfort affiche une vue du réseau contenant les appareils et les connexions réseau.

Après avoir sélectionné "Ajouter un nouvel appareil", la fenêtre de l'éditeur de diagramme apparaît.

L'éditeur de diagramme affiche les blocs de programme et les liens du programme de commande.

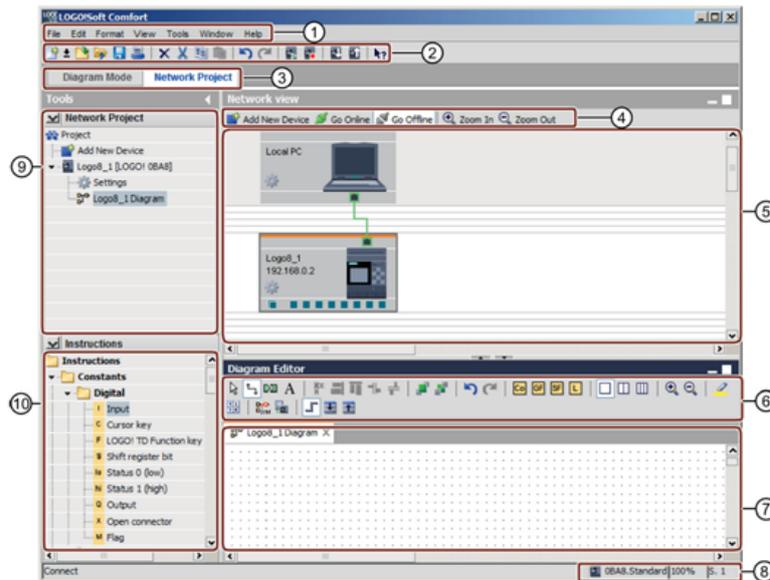


Figure 48 Fenêtre de blocs de programme

Ce qui a été rajouté :

- ④ Barre d'outils "Réseau"
- ⑤ Vue de réseau

7 Commande d'une station de pompage avec LOGO! OBA8

Le choix de l'automate s'est porté sur le LOGO! OBA8, Dont la tension de fonctionnement est de 24 V/DC, son rôle essentiel est de gérer les actionneurs tels que la pompe, les vannes motorisées et l'agitateur. Il assurera la sécurité de l'installation afin d'éviter toutes fausses manipulations de la part de l'opérateur, mais aussi il permettra la communication entre les équipements et le superviseur.



Figure 49 Automate programmable LOGO ! OBA8

7.1 Schéma de connexion des entrées et des sorties

Dans le but de répondre aux cahiers de charge effectués au premier chapitre, étant donné que le nombre de sorties de l'automate LOGO ! est limité à 4, nous avons dû affecter chaque voyant à un actionneur précis en les branchant ensemble, par conséquent les voyants V1, V2, V3 et V4 indiquent respectivement l'activation de la pompe, l'agitateur, l'électrovanne B et l'électrovanne A. ce qui permet l'optimisation du programme.

La figure 50 correspond au schéma technologique avec câblage.

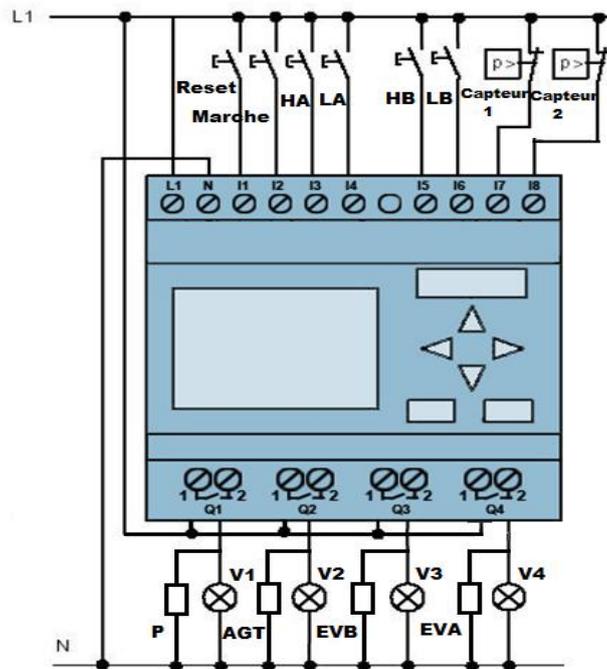


Figure 50 Schéma de connexion des entrées et des sorties

7.2 Tableau d'affectation

Tableau 5 tableau d'affectation des entrées / sorties

Adresse	E TOR
I1	Bouton reset
I2	Bouton marche
I3	Entrée HA
I4	Entrée LA
I5	Entrée HB
I6	Entrée LB
Adresse	E ANA
I7	Entrée analogique cuve A
I8	Entrée analogique cuve B

Adresse	S TOR
Q1	Voyant 1/ Pompe
Q2	Voyant 2/ Agt
Q3	Voyant 3/ Evb
Q4	Voyant 4/ Eva

7.3 Configuration de l'automate LOGO! et de son logiciel de programmation

Pour mieux comprendre le fonctionnement de l'automate LOGO ! Ainsi que de son logiciel LOGO ! Soft Comfort, nous avons traité les deux manipulations précédemment étudiées étape par étape.

7.3.1 Instruction structurée étape par étape

Avant toutes choses nous devons régler l'adresse IP du module LOGO ! 0BA8. Lorsque LOGO! 0BA8 est en mode STOP, il suffit de naviguer jusqu'à la commande de menu **Network (Réseau)**. Pour trouver les paramètres de l'adresse IP, du masque de sous-réseau et de la passerelle. **À l'aide du curseur ►** ou de la **touche OK**, nous pouvons définir les paramètres réseau conformément aux indications de notre administrateur réseau. [14]

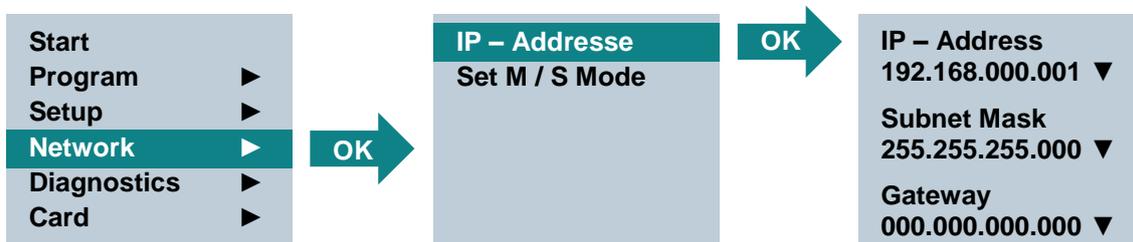


Figure 51 Réglage de l'adresse IP dans le menu du module LOGO !

En premier lieu, nous avons pu se familiariser avec l'automate LOGO! et le programmer directement sur son écran grâce aux touches curseurs dont il dispose. La figure 52 permet de mieux comprendre cette étape avec un exemple : l'activation de la sortie Q1 « la pompe » en appuyant sur l'entrée I1 « reset ».



Figure 52 Programmation sur écran

Ensuite nous avons du installer le logiciel LOGO ! Soft Comfort v8 pour la programmation de l'automate. La première étape est de démarrer le logiciel et d'ajouter le module LOGO! 0BA8.



Puis de configurer les paramètres du réseau ajouté, après avoir sélectionné LOGO! 0BA8 dans la liste des appareils.

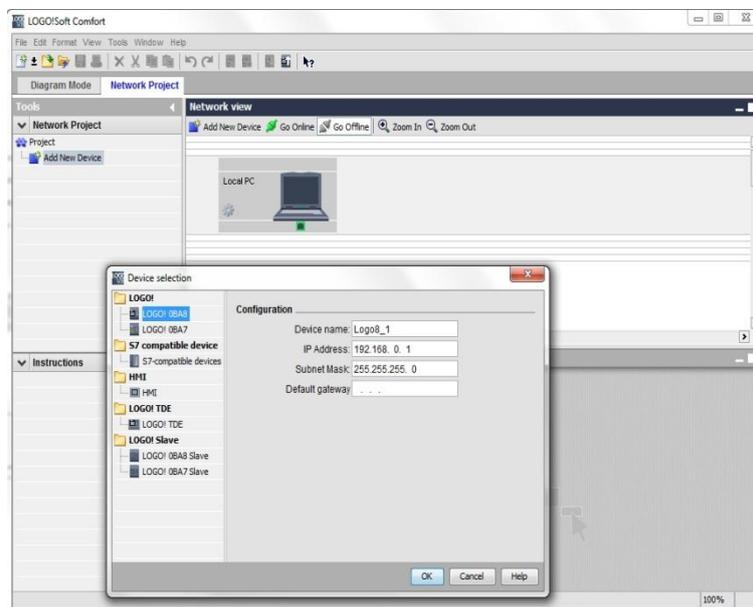


Figure 53 Fenêtre de configuration des paramètres du réseau

Par la suite vient la configuration des paramètres du LOGO! 0BA8:

Dans cette boîte de dialogue il est possible de modifier l'ensemble des paramètres hors/en ligne du LOGO! 0BA8. Le premier onglet permet de définir le nom et l'IP de l'appareil. L'adresse, le masque de sous-réseau et la passerelle pour affecter une adresse IP au module LOGO!

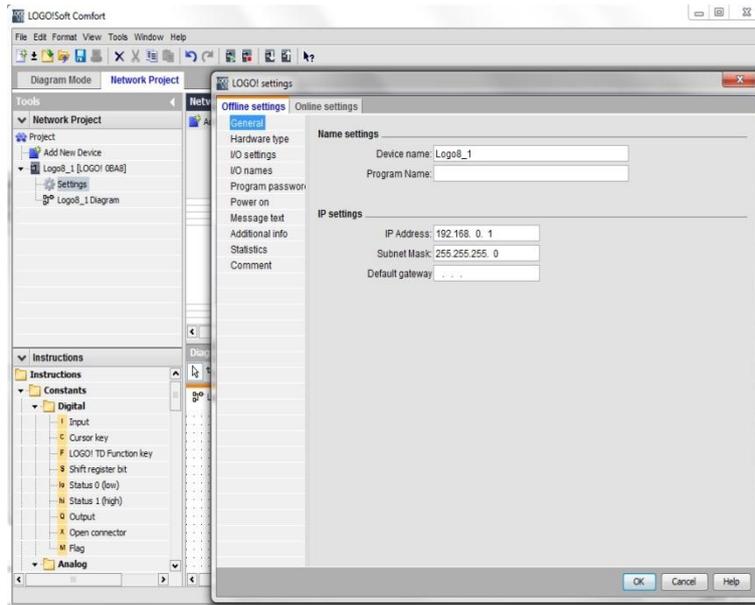


Figure 54 Fenêtre de configuration des paramètres du module

Le second onglet affiche les informations de base de LOGO! Base, tels que les blocs connus, les constantes/connecteurs et le nombre d'entrées TOR.

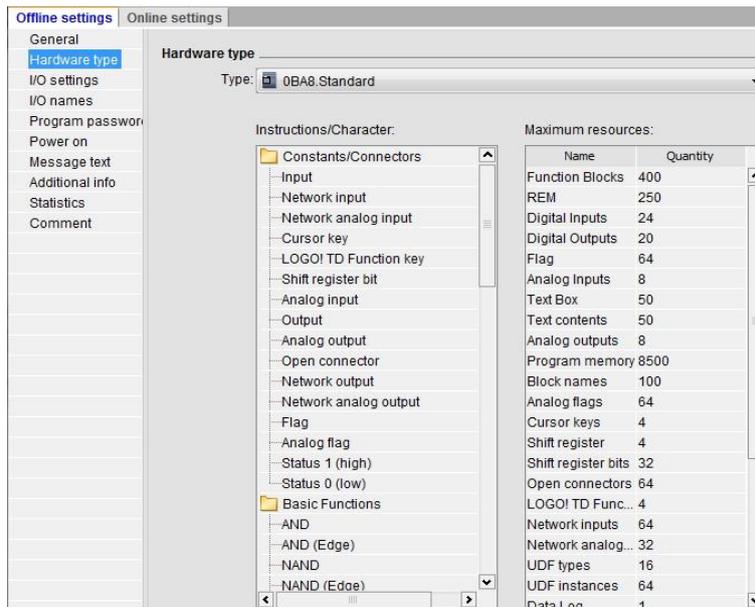


Figure 55 Fenêtre contenant les informations de base du module LOGO!

Puis les paramètres des E/S (I/O) pour la configuration des voies analogiques.

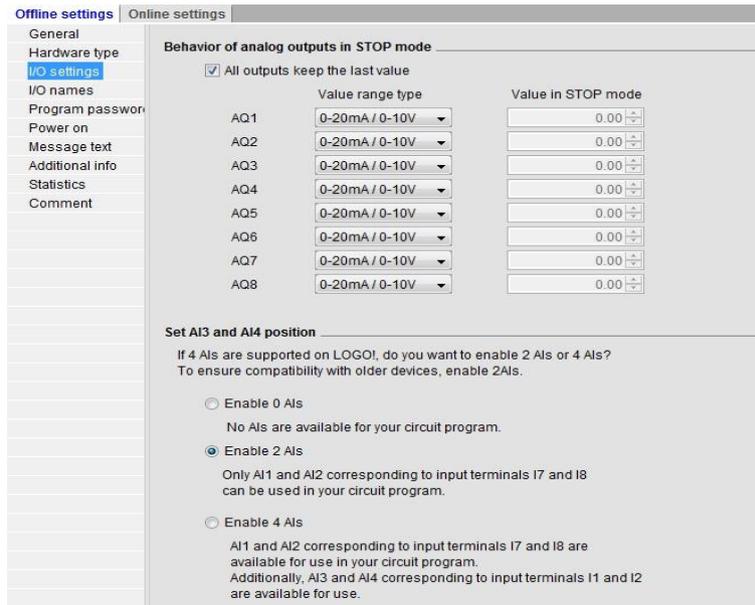


Figure 56 Fenêtre de configuration des voies analogiques

Ensuite les noms des E/S (I/O) pour la désignation des entrées et des sorties.

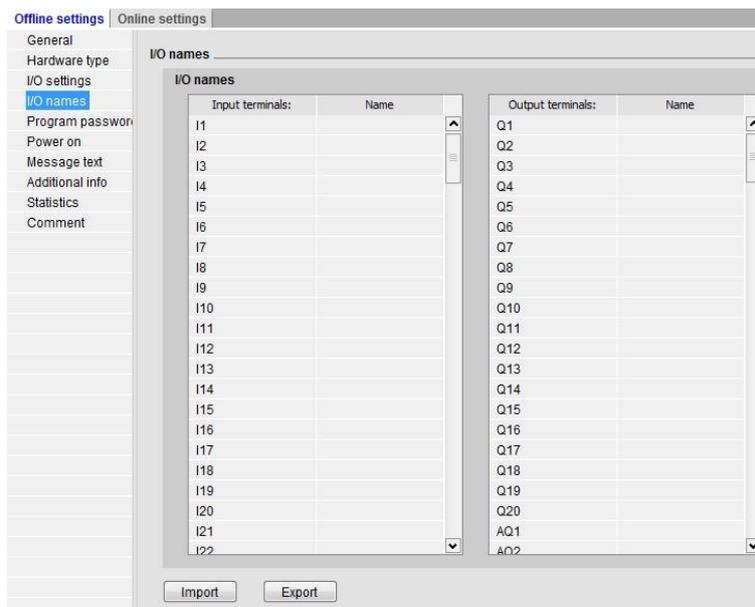


Figure 57 Fenêtre de configuration des mnémoniques

Il est possible de configurer également un mot de passe pour le programme, le contenu d'afficheur sur LOGO! Après la mise sous tension (date/heure, entrées/sortie, menu), mais aussi de configurer les réglages de texte s'appliquant à tous les textes de message etc.

7.3.2 les noms des raccordements [14]

La fenêtre des noms des E/S (I/O) (noms des raccordements) peut également s'ouvrir à l'aide du menu **Editer**. Et ce pour saisir les noms des éléments de la commande.

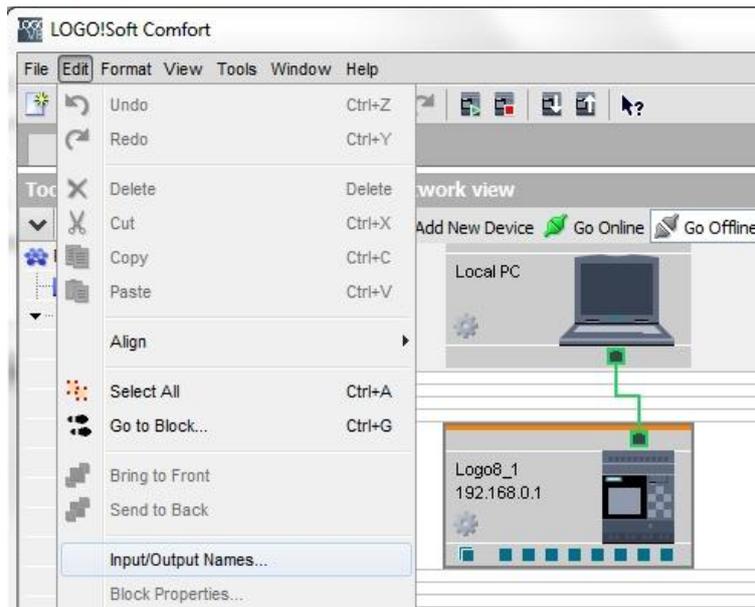


Figure 58 Fenêtre de configuration des noms des raccordements

7.3.3 1ère manipulation -partie TOR-

La table des entrées / sorties :

Bornes d'entrée :		Nom
I1	reset	
I2	marche	
I3	ha	
I4	la	
I5	hb	
I6	lb	

Bornes d'entrée :		Nom
Q1	pompe	
Q2	agt	
Q3	evb	
Q4	eva	
Q5		
Q6		

Figure 59 Table des entrées/sorties partie TOR

Le programme de la manipulation 1 partie TOR effectué avec le langage FBD est développée dans l'annexe [C].

7.3.4 2ème manipulation -partie analogique-

La table des entrées / sorties :

Bornes d'entrée :		Nom
I1	reset	
I2	marche	
I3		
I4		
I5		
I6		
I7	d1	
I8	d2	

Bornes d'entrée :		Nom
Q1	pompe	
Q2	agt	
Q3	evb	
Q4	eva	
Q5		
Q6		
Q7		
Q8		

Figure 60 Table des entrées/sorties partie analogique

Le programme de la manipulation 2 -partie analogique- effectué avec le langage LADDER est développé dans l'annexe [C].

7.4 Simulation du circuit

Après l'enregistrement du schéma de connexion, il est possible de démarrer la simulation, en cliquant sur l'icône  Simulation dans la barre d'outils "Outil". La simulation de programme permet de tester un programme de commande et de modifier son paramétrage. Ainsi, il est possible de s'assurer que le programme de commande opérationnel et optimisé dans le LOGO! [14]

7.5 Transfert du programme testé dans le LOGO!

Une fois que le test du programme dans la simulation LOGO! Soft Comfort est effectué, il suffit de procéder au transfert à l'aide du bouton  PC -> LOGO!. Pour afficher les appareils LOGO! accessibles, il existe le bouton **Actualiser**  [14]

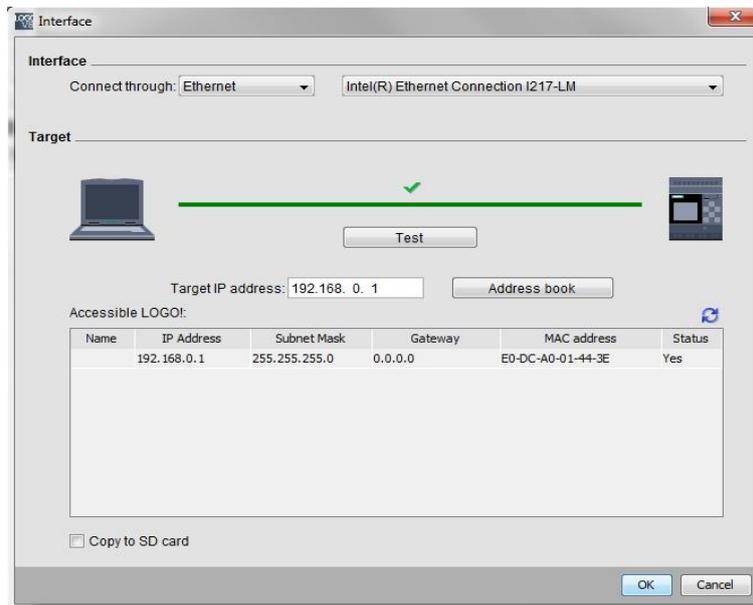


Figure 61 Interface de communication entre le pc et le module LOGO!

Le module doit être en mode stop afin de transférer le programme.

7.6 Test en ligne

Pour charger le programme, nous devons procéder à un test en ligne : Le bouton de test en ligne  permet de tester le programme de commande associé au module LOGO!. Les états des entrées et sorties et des connexions logiques s'affichent. [14]

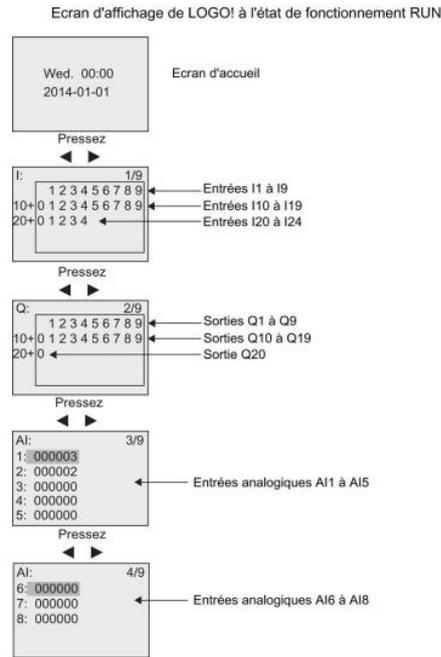


Figure 62 Afficheur sur écran (test en ligne)

8 Supervision de la maquette avec LOGO! TDE et via un navigateur Web sur PC

8.1 Affichage de texte d'alarme sur écran LOGO! TDE

Dans un premier temps, il faudra configurer les paramètres du réseau ajoutés, après avoir sélectionné LOGO ! TDE dans la liste des appareils.

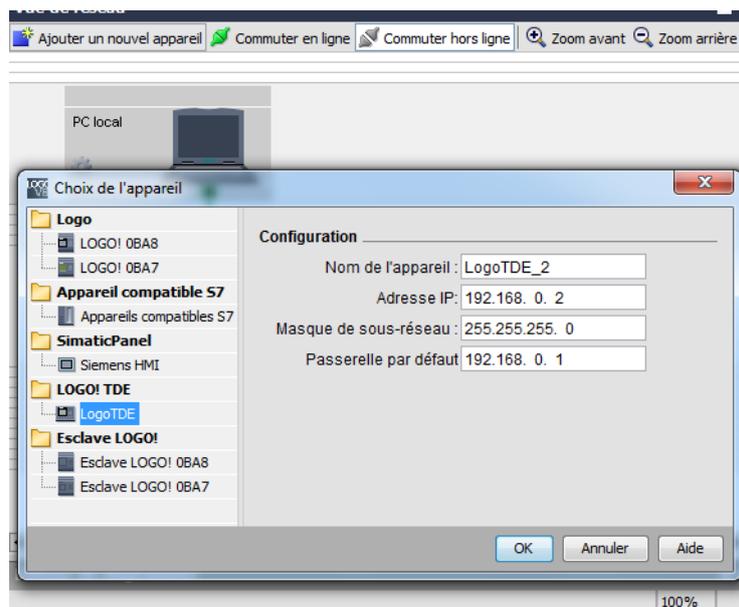


Figure 63 Fenêtre de configuration des paramètres de l'afficheur LOGO! TDE

Le premier onglet permet de définir le nom et l'IP de notre appareil. L'adresse, le masque de sous-réseau et la passerelle pour affecter une adresse IP au module LOGO! TDE

Dans la boîte de dialogue, il est possible de vérifier les propriétés de l'appareil LOGO! TDE.

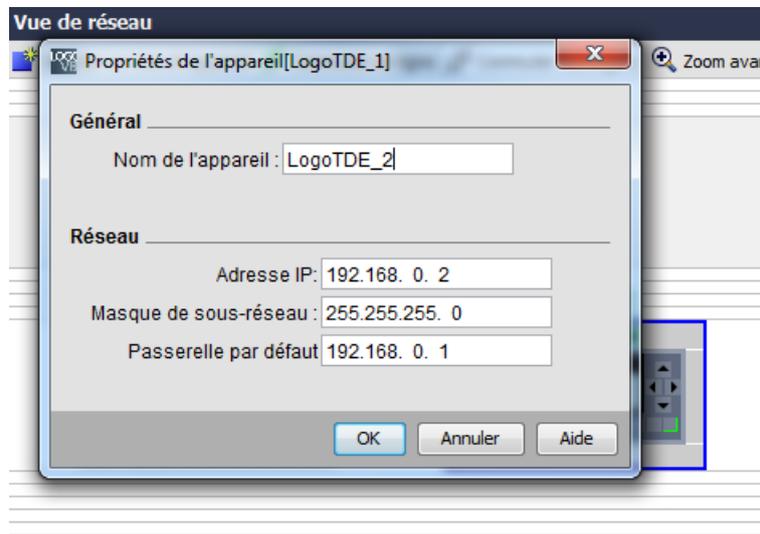


Figure 64 Fenêtre de vérification des propriétés de l'afficheur LOGO! TDE

8.1.1 Texte d'alarme

Il est possible de doter le programme de commande de la station de pompage d'une fonction de texte d'alarme. Le diagramme ainsi complété doit être configuré, programmé et testé. En outre, l'écran du module LOGO! ainsi que l'écran de supervision LOGO! TDE doivent afficher les différents états du système sous la forme d'un texte d'alarme. Afin que le texte d'alarme puisse être également visible via un navigateur Web, le serveur Web doit être sélectionné comme destinataire de texte d'alarme supplémentaire.

8.1.1.1 Insérer des textes d'alarme

Sous autres fonctions, il suffit de tirer des textes d'alarme dans le diagramme et de les connecter. [14]

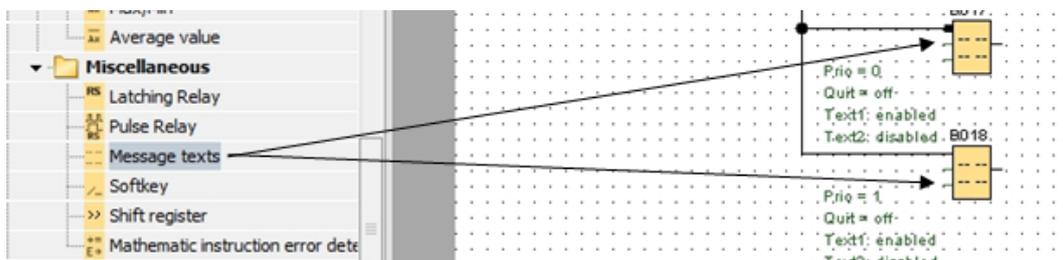


Figure 65 Fenêtre d'insertion de texte d'alarme

8.1.1.2 Saisir le texte d'alarme

le bloc de texte d'alarme par exemple : B037 ouvre une fenêtre permettant de le paramétrer. Chaque texte d'alarme possède une priorité. S'il existe plusieurs textes d'alarme, c'est le texte qui détient la priorité la plus élevée qui s'affiche.

Nous pouvons par ailleurs choisir les destinataires, l'écran LOGO! Display, le serveur Web ou l'afficheur LOGO! TDE.

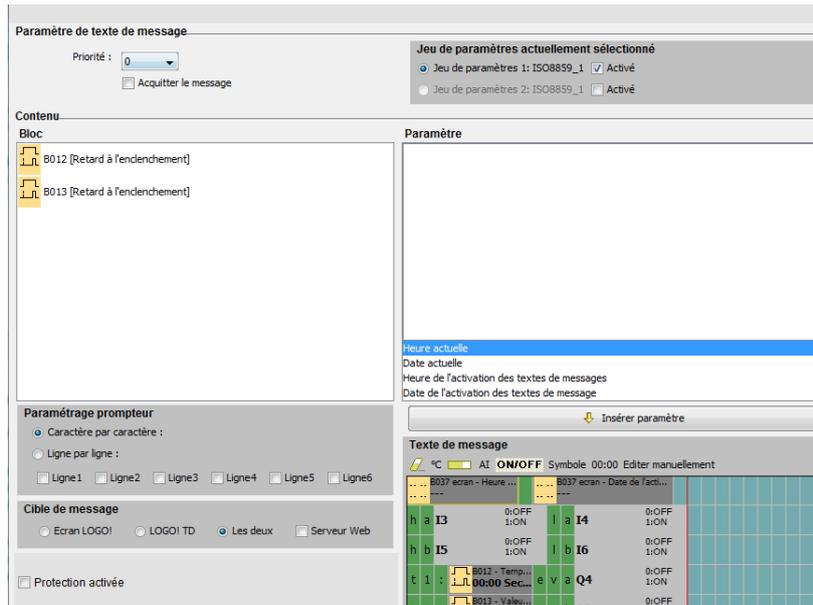


Figure 66 Fenêtre permettant de paramétrer le texte d'alarme

8.1.1.3 Affichage de la première manipulation TOR

A partir du grafcet de la 1ère manipulation TOR, nous pouvons superviser sur LOGO! TDE les sorties dont la pompe, l'agitateur, l'électrovanne B et l'électrovanne A, suivant les voyants V1, V2, V3 et V4, de même que que les entrées HA (high a), LA (low a), HB (high b) et LB (low b), ainsi que les timers t1 et t2.

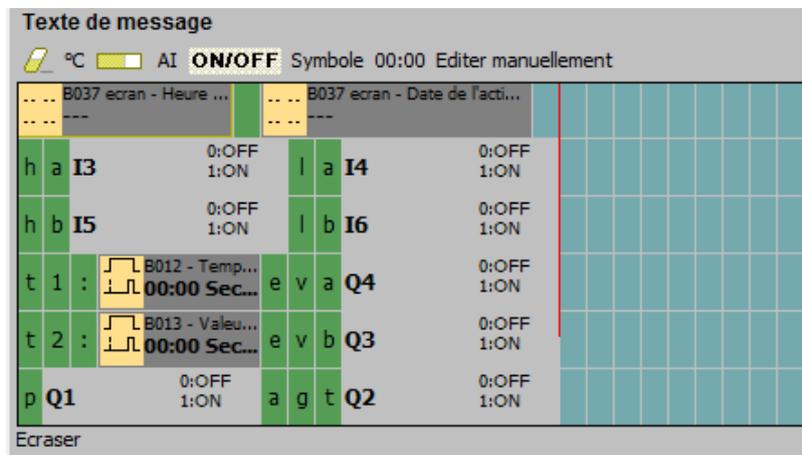


Figure 67 Exemple d'affichage de la manipulation TOR

Les figures 68 et 69 montrent l'état de l'installation à la 6ème étape du grafcet, le voyant v2 est allumé et l'agitateur est en état de marche pendant 5s.



Figure 68 Affichage « partie TOR » sur écran LOGO!



Figure 69 Affichage « partie TOR » sur écran LOGO! TDE

8.1.1.4 Affichage de la première manipulation analogique

A partir du grafctet de la 2ème manipulation analogique, nous pouvons superviser sur LOGO !TDE les mêmes sorties/entrées de plus que les deux entrées analogiques d1 et d2 qui nous révèlent le niveau d'eau dans les cuves a et b.

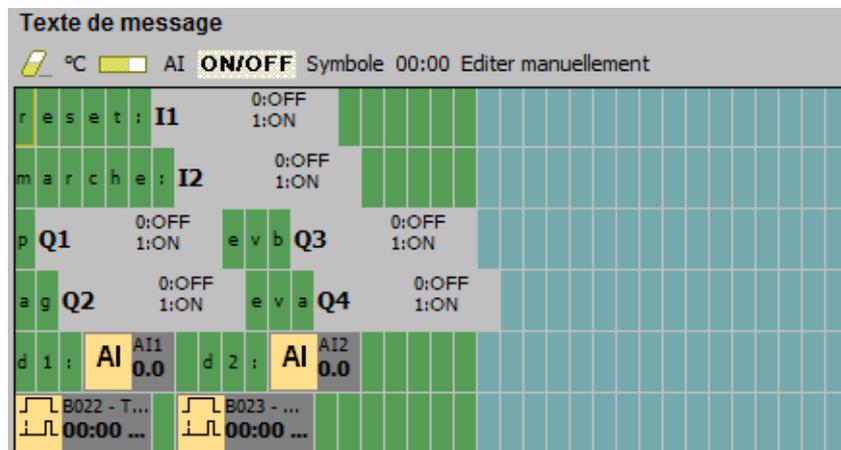


Figure 70 Exemple d'affichage de la manipulation analogique

8.1.2 Définition du rétro-éclairage du texte d'alarme

Des mémentos permettent de définir le rétro éclairage de l'écran LOGO! Display, ainsi que l'écran LOGO ! TDE.

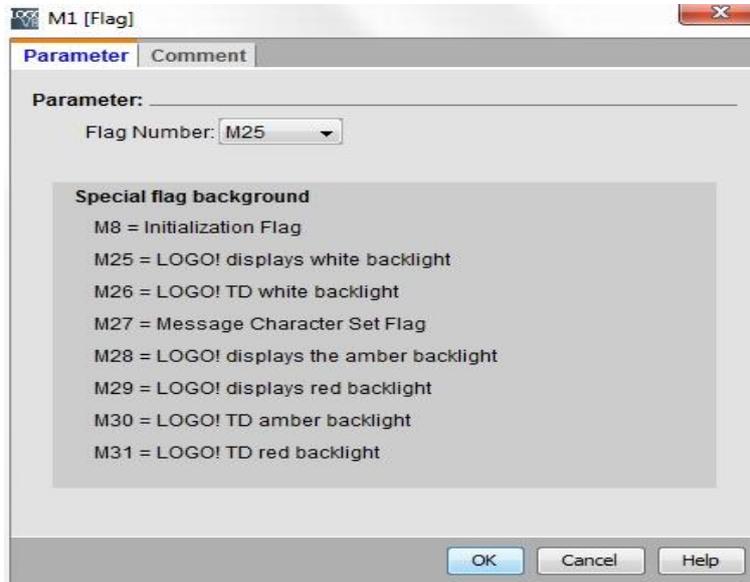


Figure 71 Fenêtre permettant de définir un rétro-éclairage du texte d'alarme

Il est possible de joindre chaque sorties a un memento de couleur en cohérence avec les voyants utilisés auparavant dans l'installation. La figure 72 représente l'affectation de la couleur rouge à l'activation de la pompe.



Figure 72 Exemple de rétroéclairage

Un message texte est joint dans l'afficheur a fin de stipuler le voyant allumé a chaque instant.

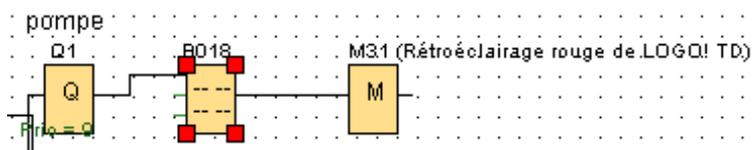


Figure 73 Exemple d'un bloc message texte

8.1.3 Touches de fonction LOGO! TDE



Le module LOGO! TDE possède quatre touches de fonction que nous pouvons utiliser comme entrées numériques dans le programme de commande. Nous les programmons exactement comme les autres entrées. L'utilisation de touches de fonction économise l'utilisation de commutateurs et d'entrées et permet l'intervention manuelle dans le programme de commande. [15]

Nous avons utilisé ces touches pour activer les quatre sorties Q1 Q2 Q3 et Q4 pour une manipulation plus facile sur IHM.

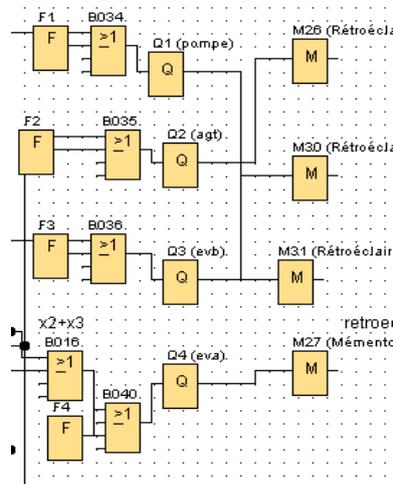


Figure 74 Exemple de rétro-éclairage du texte d'alarme utilisé

8.2 Affichage de texte d'alarme via un navigateur Web [14]

LOGO! 0BA8 possède un serveur Web intégré qui permet d'exploiter le module LOGO! Base ou le LOGO! TDE à partir d'un PC classique ou d'un appareil mobile.

Dans cette optique, l'accès au module LOGO! Base ou au LOGO! TDE à l'aide d'un appareil connecté (PC classique, tablette ou smartphone avec fonction de navigation Web) s'effectue au moyen de son adresse IP.

Le serveur Web permet d'utiliser le pointeur de la souris ou l'écran tactile (en fonction de l'appareil que nous utilisons) pour effectuer des opérations rapides et simples sur le module LOGO! Base et le LOGO! TDE.

8.2.1 Activation d'un serveur Web dans LOGO!

Pour afficher les textes d'alarme du module LOGO! via un navigateur Web, celui-ci doit d'abord être activé dans les paramètres en ligne de LOGO!.

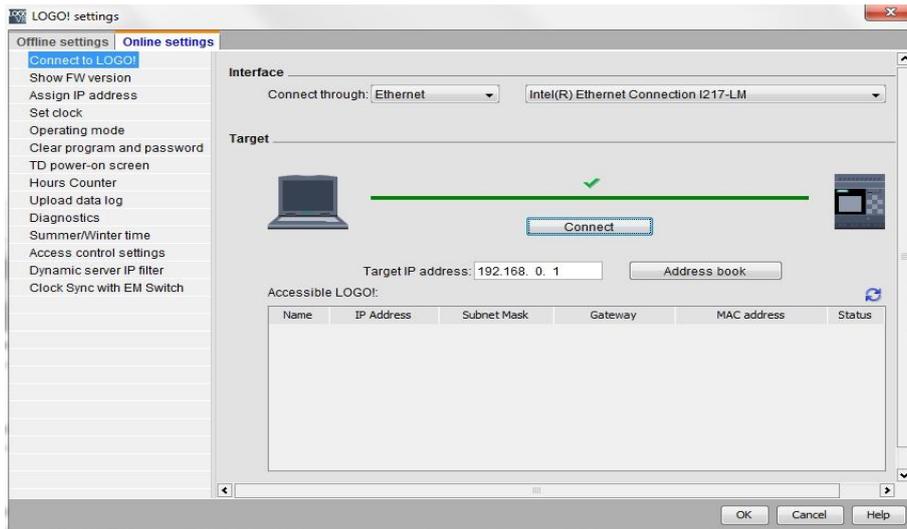


Figure 75 Fenêtre permettant de configurer la liaison pc/module

Le contrôle d'accès s'effectue seulement à la mise en mode ARRÊT du module.
Il suffit ensuite d'activer l'option "Autoriser l'accès au serveur Web" dans le menu "Paramètres pour le contrôle d'accès"

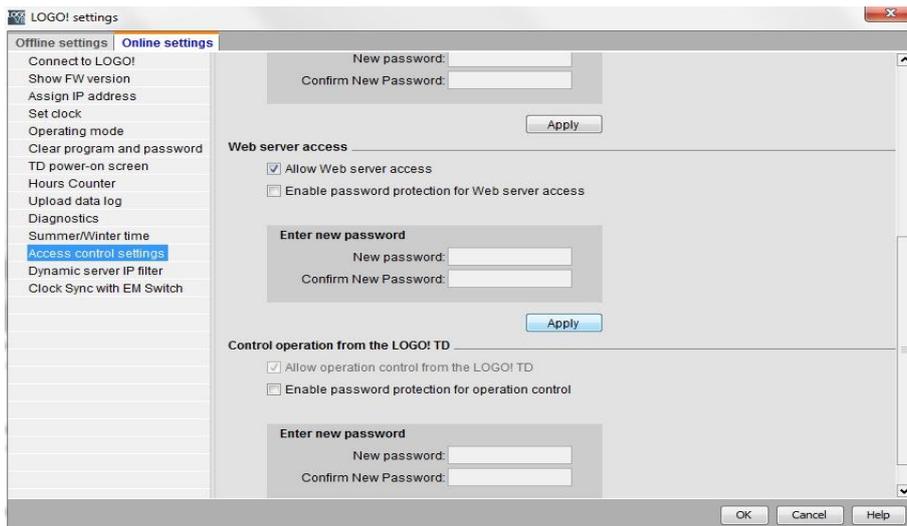


Figure 76 Fenêtre paramètres pour le contrôle d'accès

Une fois le serveur Web est activé, le module LOGO! passe de nouveau en mode MARCHE.

8.2.2 LOGO! Dans le navigateur Web

Dans un premier temps dans le navigateur Web d'un pc d'une tablette ou d'un smartphone, nous devons entrer l'adresse IP du LOGO!. Pour l'accès via Internet avec port TCP (par exemple <http://192.168.0.1>),

Ensuite Dans le champ de saisie, nous devons indiquer le mot de passe avant de se connecter. Dans l'état à la livraison, le mot de passe est "LOGO".

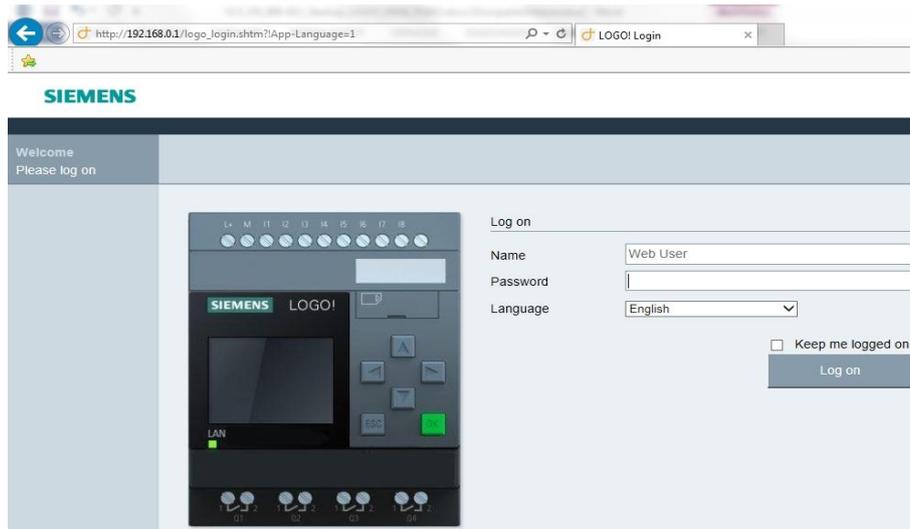


Figure 77 Fenêtre permettant de configurer le LOGO! Dans le navigateur Web



Figure 78 Fenêtre permettant de visualiser le programme LOGO! Dans le navigateur Web

8.2.3 Fonctionnement du module virtuel sur le serveur Web

Le serveur Web LOGO! permet d'exécuter les opérations suivantes sur le module LOGO! virtuel par le biais du menu LOGO! BM et sur le LOGO! TDE virtuel par le biais du menu LOGO! TDE.

8.2.3.1 Sur le module LOGO! Base

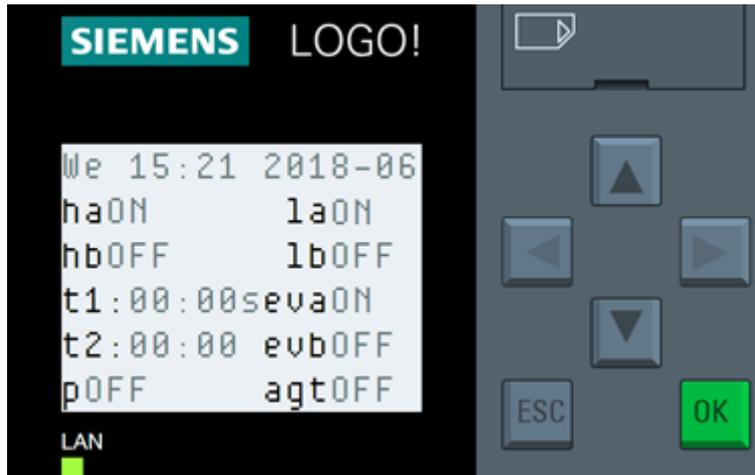


Figure 79 Exemple de visualisation de la manipulation TOR du module LOGO !

8.2.3.2 Sur le LOGO! TDE

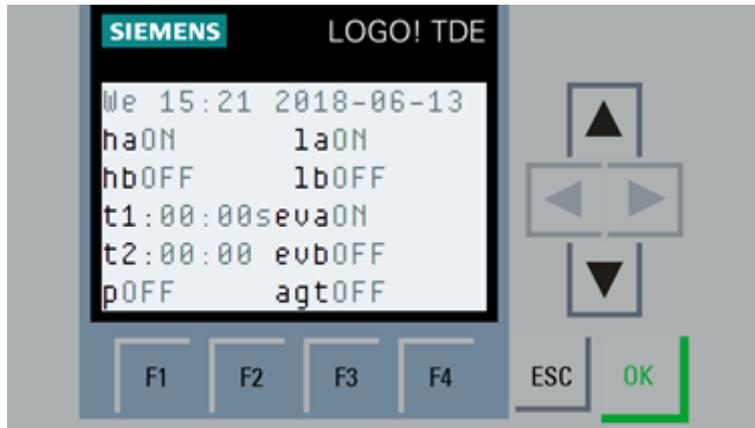


Figure 80 Exemple de visualisation de la manipulation TOR du module LOGO ! TDE

Les figures 79 et 80 montrent l'état de l'installation à la 2ème étape du grafcet, les entrées marche, ha sont activées, le voyant v4 est allumé et l'électrovanne EVA est en état de marche.

9 Conclusion

L'importance de ce chapitre réside dans la mise en évidence du rôle des différents outils utilisés pour implanter et simuler le programme de l'automate LOGO !. En effet, ces outils nous ont permis de simplifier le travail demandé, ainsi que d'automatiser l'application et faciliter son utilisation par l'opérateur. Ce chapitre nous a permis le câblage de l'application, et la mise en œuvre de l'automate programmable industriel.

Nous avons pu par ailleurs exploiter l'automate LOGO ! D'une manière optimale en économisant le nombre de sortie et ce grâce au logiciel de programmation LOGO ! Soft Comfort V8.0.

Et pour finir nous avons pu effectuer la supervision de la maquette à l'aide d'un écran de supervision LOGO! TDE et via navigateur Web sur PC.

Conclusion générale

Toutes connaissances théoriques restent un atout pour l'ingénieur. Cependant les travaux pratiques sont nécessaires à l'apprentissage et au développement de ce dernier. C'est dans cette optique que nous avons voulu améliorer et ajouter un plus à la gestion de la station de pompage.

Pendant la réalisation de ce projet, nous avons acquis de nouvelles connaissances et compétences, avoir étudié au préalable les deux automates Zelio et S7 300 en informatique industrielle durant notre cursus universitaire, nous a permis d'avancer plus rapidement dans l'intégration de ces deniers dans la maquette de la station de pompage. Leurs configurations, programmations, simulations et manipulations ont été faite sans trop de difficultés.

Cependant l'intégration de l'automate LOGO! était pour nous une nouvelle expérience puisque ce fut la première fois que nous manipulions cet automate. L'utilisation de ce dernier nous a permis d'apprendre d'avantage sur cette gamme d'automate très utilisée en industrie. Il faut savoir que l'apprentissage d'un nouveau logiciel de programmation est toujours bénéfique pour forger un ingénieur automaticien.

On a rencontré un problème suite à la configuration d'un pupitre (écran de supervision) 700 IE V3, son logiciel de programmation (WinCC flexible smart) n'est pas disponible au niveau de Siemens Algérie, puisqu'il n'est pas très utilisé en industrie, nous n'avons malheureusement pas pu l'utiliser, néanmoins nous avons pu intégrer un écran d'affichage LOGO! TDE pour la supervision de la maquette.

Enfin, nous espérons que notre travail sera utile aux étudiants en automatique, pour appliquer les connaissances requises en cours d'informatique industrielle.

Il serait très intéressant d'améliorer l'application en ajoutant quelques fonctionnalités et d'intégrer d'autres éléments comme :

- 1 Un capteur de niveau et un capteur de débit.
- 2 Un variateur de fréquence pour alimenter la pompe (avoir un débit variable).
- Réalisation de la commande de niveau et/ou de débit en utilisant ces capteurs et le variateur de fréquence.
- 3 Un pupitre de supervision de la station.

Bibliographie

- [1] BATTEIX Fabia. TamiaLab. [En ligne]. [Consulté le: 17 02 2018].
Disponible sur <https://www.carnetdumaker.net/articles/mesurer-une-distance-avec-un-capteur-ultrason-hc-sr04-et-une-carte-arduino-genuino/>
- [2] ALLALI Ahmed et BETTACHE Lyes Heythem. *Conception et réalisation d'un capteur de niveau Application à la supervision d'une station de pompage*. 97pages.
Projet de Fin d'Etude : Automatique : Alger, Ecole Nationale Polytechnique : 2017.
- [3] SAHRAOUI Hamza et UGERGAZI Khaled. Etude de la tréfileuse et sa commande automatique (entreprise .National .industrie des câbles Biskra). 80 pages.
Thèse d'ingénieur : Electrotechnique : Biskra, Université Mohamed Khider: 2006.
- [4] BOLTON William. Automates Programmables Industriels. DUNOD : Paris, 2015.
- [5] Siemens. STEP 7 Pour une transition facile de S5 à S7 : SIMATIC. Edition 12/2002.
- [6] Siemens. Faites connaissance avec le S7-300 : SIMATIC. Edition 1996.
- [7] Siemens. Programmer avec STEP 7, version 5.2 : SIMATIC, 05/2010.
- [8] FOUKA Madjda Amina Aida et FERHAOUI Bouchera. Réalisation d'une maquette de station de pompage à base d'un automate programmable Siemens. 105 pages.
Projet de Fin d'Etude : Automatique : Alger, Ecole Nationale Polytechnique : 2015.
- [9] Siemens. Mise en route WinCC flexible débutants: SIMATIC HMI. Edition2008.
- [10] Siemens. SIMATIC HMI WinCC flexible 2008: Compact/Standard/Advanced Edition07/2008.
- [11] Automationsense. Cours automatisme Schneider: automate et supervision. Formation automatisme, 06/06/2015.
- [12] Schneiderelectric. Modules logiques Zelio Logic. Edition 1er semestre 2004.
- [13] Schneiderelectric. Guide de programmation du module Zelio Logic.Edition 10/2017.
- [14] S. A. C. Education, Modules complémentaires 900-011, Curriculum SCE, 05/2016.
- [15] Siemens. Manuel LOGO. Edition 06/2014.
- [16] SGHAYER Taiser et THELB Nora, Automatiser un filtre de poussière.
Licence appliquée en informatique industriel système embarqué : Tunisie: Institut supérieur informatique de Medenine , 2011.

ANNEXE A

Description des éléments du prototype de la station de pompage

A.1 Démarreur progressive pour moteur asynchrone
A.2 Relais de niveau
A.3 Relais d'interface
A.4 Alimentation stabilisée
A.5 Electrovannes
A.6 Pompe

A.1 Démarreur progressive pour moteur asynchrone ATS 01N103FT :

A.1.1. Présentation :

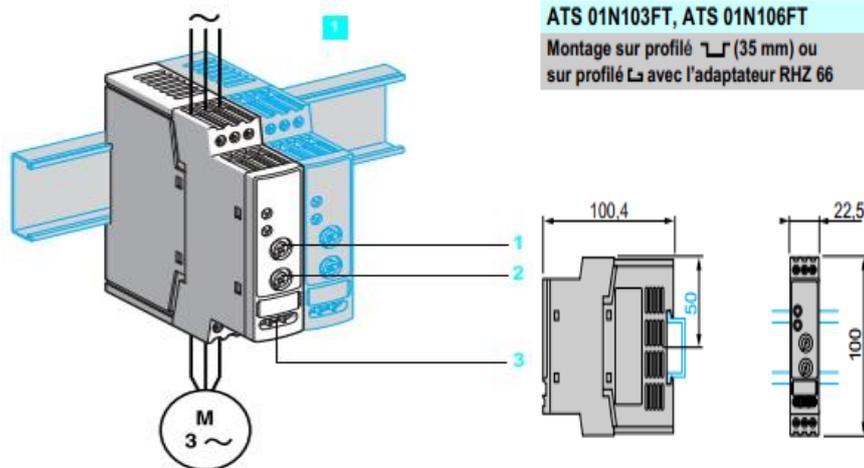
Le démarreur progressif Altistart 01 est soit un limiteur de couple au démarrage, soit un démarreur progressif ralentisseur pour les moteurs asynchrones. L'utilisation de l'Altistart 01 améliore les performances de démarrage des moteurs asynchrones en permettant un démarrage progressif sans à-coup et contrôlé. Son utilisation permet la suppression des chocs mécaniques cause d'usure, d'entretien et de temps d'arrêt de production.

L'Altistart 01 limite le couple de décollage et les pointes de courant au démarrage, sur des machines pour lesquelles un couple de démarrage élevé n'est pas nécessaire. Ils sont destinés aux applications simples suivantes : Convoyeurs, tapis transporteurs, pompes, ventilateurs, compresseurs, portes automatiques, les petits portiques, machines à courroies...

A.1.2. Description :

Les démarreurs progressifs Altistart 01 (ATS 01N1ppp) sont équipés :

- D'un potentiomètre de réglage **1** du temps de démarrage,
- D'un potentiomètre **2** pour ajuster le seuil de tension de démarrage en fonction de la charge du moteur.
- De 2 entrées **3**:
 - 1 entrée z 24 V ou 1 entrée a 110...240 V pour l'alimentation du contrôle qui permet la commande du moteur.



Description des éléments du démarreur et ses dimensions

A.1.3. Schéma électrique :

Alimentation monophasée ou triphasée.

A1 : Démarreur progressif.

Q1 : Disjoncteur GV2 ME.

KM1 : LC1*** +LA4 DA2U.

F1/F2 : Fusible de protection de commande.

S1/S2 : Boutons poussoirs XB4 B XB5 B.

(1) Contacteur de ligne obligatoire dans la séquence.

Note : Pour un moteur monophasé, on utilise l'ATS 01N1ppFT sans connecter la 2ème phase 3/L2,4/T2. Attendre 5 secondes entre une mise hors tension et une mise sous tension du démarreur progressif.

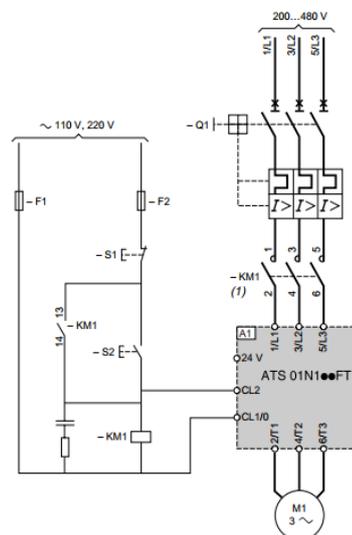


Schéma électrique du démarreur progressif ATS 01N103FT

A.1.4. Diagramme fonctionnel :

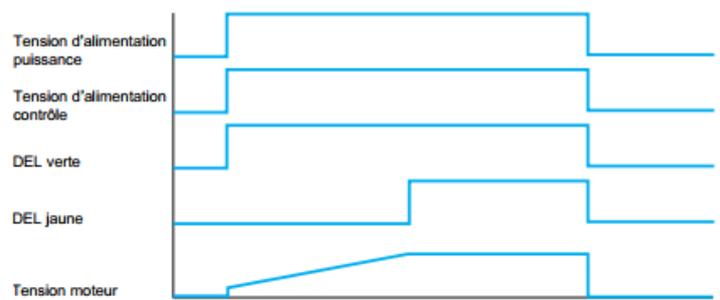


Diagramme fonctionnel du démarreur progressif ATS 01N103FT

A.2 Relais de niveau (CHINT JYB-714C):

A.2.1. Présentation :

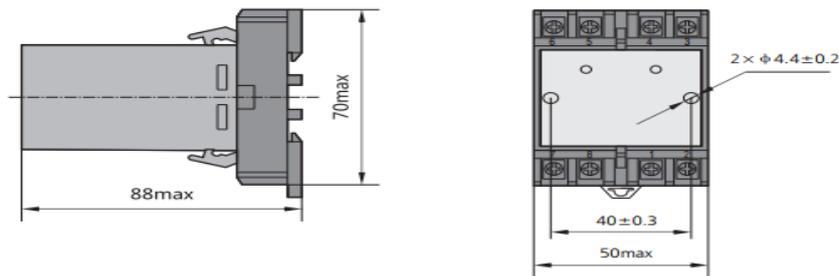
Les relais de contrôle de niveau de liquides sont utilisés pour la protection marche à sec des pompes ou comme régulateur de niveau des liquides. Ils fonctionnent à l'aide de détecteurs, qui mesurent la conductivité. A cette fin un détecteur pour la hauteur de remplissage maximal et un détecteur pour la hauteur de remplissage minimale sont nécessaires. Un troisième détecteur sert comme potentiel de masse.

A.2.2. Fiche technique :

Mode opératoire	Mode continue
Capacité du contact	Ue/Ie: AC 15/220V/1.5A ; Ith :5A
Voltage d'opération	AC 50Hz/60Hz 36V,110V, 220V,380V
Courant maximal de l'électrode de control	50 μ A
Longueur du conducteur	Max 1Km
Température ambiante	-5°C ~ +40°C
Puissance absorbée	<3VA

Caractéristiques techniques du relais CHINT JYB-714C

A.2.3. montage et dimension :

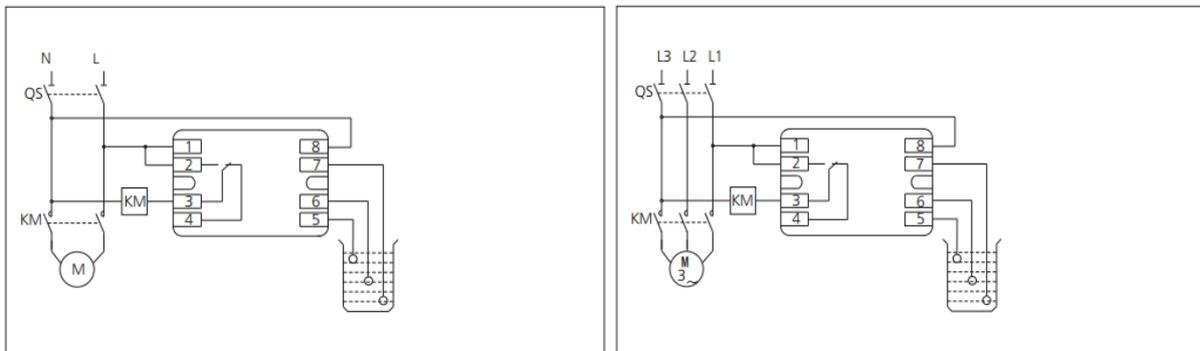


Vue de face et de coté du relais CHINT JYB-714C

A.2.4. Schéma électrique :

Pour une installation 220V :

Pour une installation 380V :



Schémas électriques pour les installations 220V et 380V

A.3. Relais d'interface OMRON MK2P-I

A.3.1 Définition

C'est un appareil composé d'une bobine (électroaimant) qui lorsqu'elle est parcourue par un courant électrique agit sur un ou plusieurs contact. Un relais électromagnétique est un organe électrique permettant de dissocier la partie puissance de la partie commande : Il permet l'ouverture/fermeture d'un circuit électrique par un second circuit complètement isolé (isolation galvanique) et pouvant avoir des propriétés différentes.

A.3.2 Description :

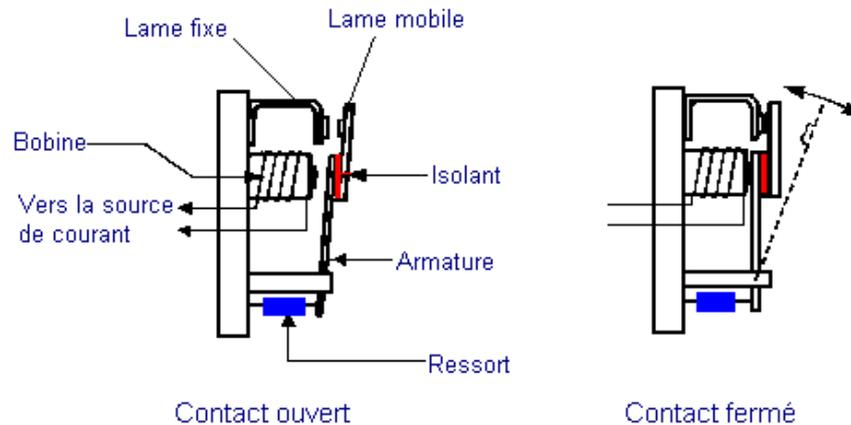


Schéma d'un relais électromécanique

Un relais est composé principalement d'un électroaimant, qui lorsqu'il est alimenté, transmet une force à un système de commutation électrique : les contacts. L'électroaimant peut être, suivant les spécifications et besoins, alimenté en TBT (Très Basse Tension) (12 V, 24 V, 48 V) continu ou alternatif ou en BT (Basse Tension) (230 V, 400 V). Le système de commutation peut être composé d'un ou plusieurs interrupteurs simples effets appelés contacts normalement ouverts (NO) ou normalement fermés (NF), d'un ou plusieurs inverseurs (contacts repos-travail RT). Ces commutateurs sont adaptés aux courants et à la gamme de tensions à transmettre à la partie puissance.

A.3.3. Fiche technique :

Produit:	Relais de puissance
Tension de la bobine:	24 VDC
Forme de contact du relais :	2 Forme C (DPDT-NO, NC)
Résistance de la bobine:	430 Ohms
Courant de la bobine:	56 mA
Consommation d'énergie :	1.5 W
Contact:	250VAC~/28V DC (10A (N.O), 5A (N.C)) 250VAC~7A GEN. $\cos\phi=0.4$

Caractéristiques techniques du relais OMRON MK2P-I

A.3.4. Connexion interne du terminal (vue du bas):

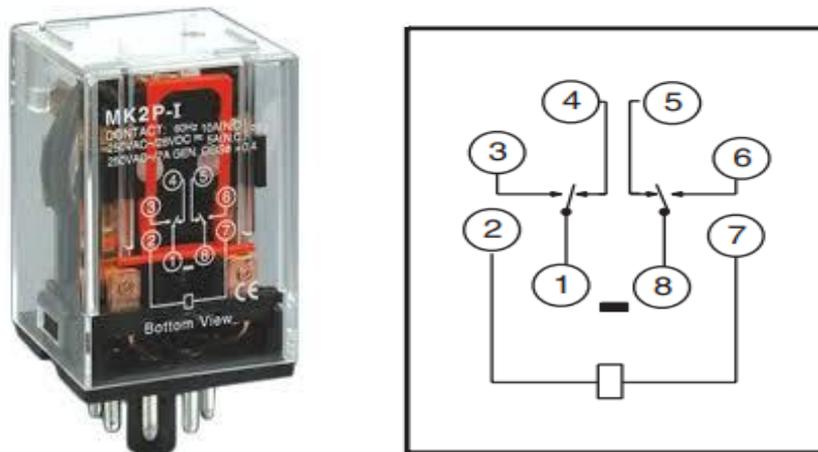


Schéma de connexion interne du relais.

A.4 Alimentation stabilisée

A.4.1 Définition :

Une alimentation stabilisée est un dispositif électronique, utilisant une régulation, permettant le réglage de la tension et du courant nécessaires au fonctionnement d'un appareil électrique ou électronique. Il est généralement conçu pour fonctionner en générateur idéal de tension et de courant.



A.4.2. Description :

Exemple d'une alimentation stabilisée :

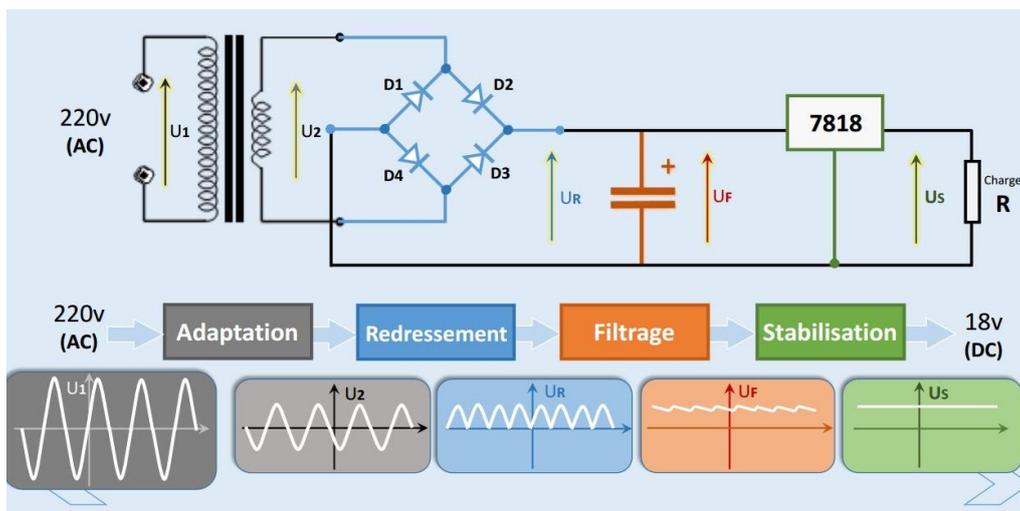


Schéma électrique d'une alimentation stabilisée

A.4.3. Caractéristiques:

Tension et courant d'entrée :	100-120VAC 3.3A 200-240VAC 2.0A 50/60Hz
Tension et courant de sortie :	24V DC 5A

Caractéristique d'alimentation

A.5 Electrovanes :

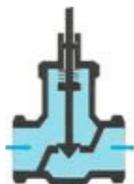
A.5.1. Définition :

Une électrovanne ou électrovalve est un dispositif commandé électriquement par un électroaimant permettant d'autoriser ou d'interrompre par une action mécanique la circulation d'un fluide dans un circuit. Il existe deux types d'électrovanne : "tout ou rien" et "proportionnelle". Elles sont utilisées usuellement dans : l'industrie alimentaire, les industries chimiques et pétrochimiques...

A.5.2. Description :

Une électrovanne est composée de deux parties :

1. Une tête magnétique constituée principalement d'une bobine, tube, culasse, bague de déphasage, ressort(s).
2. Un corps, comprenant des orifices de raccordement, obturés par clapet, membrane, piston, etc. selon le type de technologie employée. L'ouverture et la fermeture de l'électrovanne est liée à la position du noyau mobile qui est déplacé sous l'effet du champ magnétique engendré par la mise sous tension de la bobine.



Normalement Fermée hors tension :
l'électrovanne est ouverte sous tension.



Normalement Ouverte hors tension :
l'électrovanne est fermée sous tension.

Fonctionnement de l'électrovanne

A.6 Pompe :

A.6 .1. Définition :

Une pompe électrique est une pompe à eau qui fonctionne à partir de l'énergie électrique. Elle transforme l'énergie électrique reçue en énergie mécanique servant à activer la

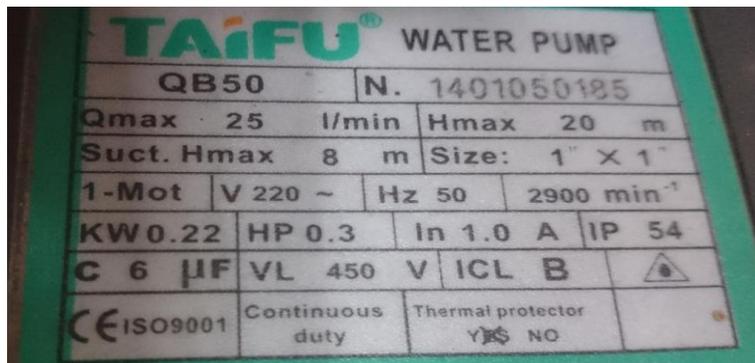
pompe.

L'énergie mécanique sera ensuite transmise au liquide où elle est convertie en énergie cinétique « débit » puis en énergie potentielle « pression ». C'est à partir de ces énergies que le fonctionnement de la pompe à eau électrique s'opère à travers des échanges et consommations d'énergies dans les circuits d'installation.

A.6.2. Le mode de fonctionnement du surpresseur d'eau :

Le surpresseur d'eau sert principalement à faciliter la montée de l'eau. Celui-ci va alors aspirer l'eau pour la faire venir jusqu'à la surface et ce, avec une pression appréciable. Cet appareil est constitué d'un moteur produisant la puissance utile au pompage. Cette puissance sera ensuite émise à la partie hydraulique qui à son tour s'occupe de faire parvenir cette puissance jusqu'à l'eau pour ainsi enclencher son ascension. Dans le cas d'un puits, le tuyau du surpresseur d'eau est inséré au fond du puits, puis le moteur est activé et transmet une grande source d'énergie contribuant alors à l'ascension de l'eau. Le surpresseur s'avère également utile pour les eaux de pluies. Celui-ci maintient les eaux puis les utilise à nouveau notamment pour arroser le jardin.

A.6.3. Caractéristiques:



TAIFU® WATER PUMP			
QB50		N. 1401050185	
Qmax	25 l/min	Hmax	20 m
Suct. Hmax	8 m	Size:	1" X 1"
1-Mot	V 220 ~	Hz	50
			2900 min ⁻¹
KW	0.22	HP	0.3
In	1.0 A	IP	54
C	6 μF	VL	450 V
		ICL	B
CE	ISO9001	Continuous duty	Thermal protector
			YES NO

Plaque signalétique de la pompe

ANNEXE B

Les automates programmables utilisés

- B.1 Automate Siemens S7-300**
- B.2 Automate ZELIO LOGIC**
- B.3 Automate LOGO ! 12/24 RCE**

B.1 Automate Siemens S7-300

L'automate choisi dans le deuxième chapitre c'est l'automate S7-300 de Siemens qui se compose d'une CPU 314 IFM, un module d'alimentation, un module d'entrées/sorties analogiques, et un module d'entrées/sorties TOR. Les références de chaque module se présentent dans le tableau suivant :

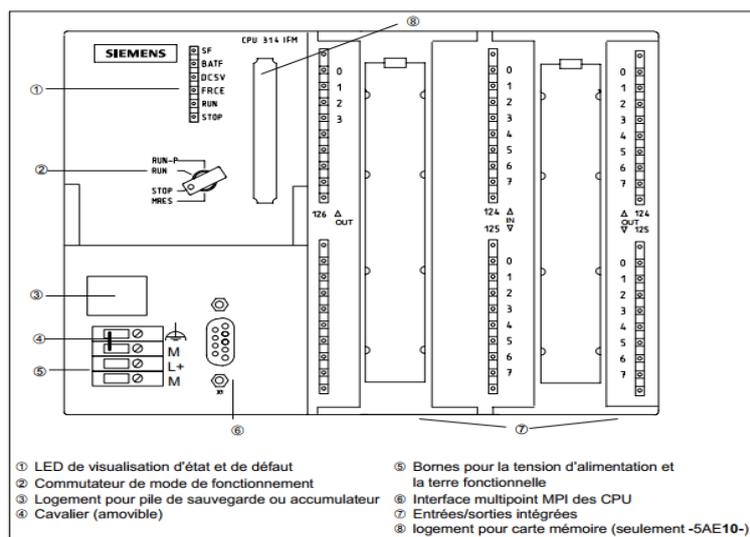
Référence :	6ES7 314-5AE03-0AB0
CPU :	CPU 314 IFM
Module d'alimentation :	PS 307; 2A
Module d'entrées TOR :	SM 323; DI16/DO16*24V
Module d'entrées analogiques :	SM 334 ; DI4*DC24V AI4/AO1*12BIT

Caractéristiques techniques de la CPU 314IFM

Interface MPI :

L'interface MPI est l'interface de la CPU utilisée pour le PG/OP ou pour la communication au sein d'un sous-réseau MPI. La vitesse de transmission typique (par défaut) est de 187,5 kbits/s.

Sur l'interface MPI, la CPU envoie automatiquement ses paramètres réseau (la vitesse de transmission par exemple). Une console de programmation peut ainsi s'inclure automatiquement dans un sous-réseau MPI.



Vue de face de la CPU 314 IFM

B.2 Automate ZELIO LOGIC

Le choix de l'automate s'est porté sur Schneider Electric SR2 B201BD ZELIO LOGIC dont la tension de fonctionnement est de 24 V/DC, son rôle essentiel est de gérer les actionneurs tels que la pompe, les vannes motorisées et l'agitateur. Il assurera la sécurité de l'installation afin d'éviter toute fausse manipulation de la part de l'opérateur, mais aussi il permettra la communication entre les équipements et le superviseur.



Vue de face et de côté de l'automate Zelio logic

Caractéristiques techniques :

Référence	SR2 B201BD
Horloge	Oui
Température ambiante max.	55 °C
Température ambiante min.	-20 °C
Courant permanent max.	250 V/AC – 8A
Possibilité d'extension	Non
Indice de protection	IP20
Tension de fonctionnement	24 V/DC
Nombre d'entrées max.	12
Nombre de sortie relais max.	8
Catégorie	API- Module de commande
Nombre d'entrées analogiques max.	6
Nombre d'entrées numériques max.	12
Affichage	Oui
Nombre de sorties max.	8

Caractéristiques techniques de l'automate Zelio logic

Cordon de raccordement ZELIO LOGIC : Schneider Electric SR2 CBL01



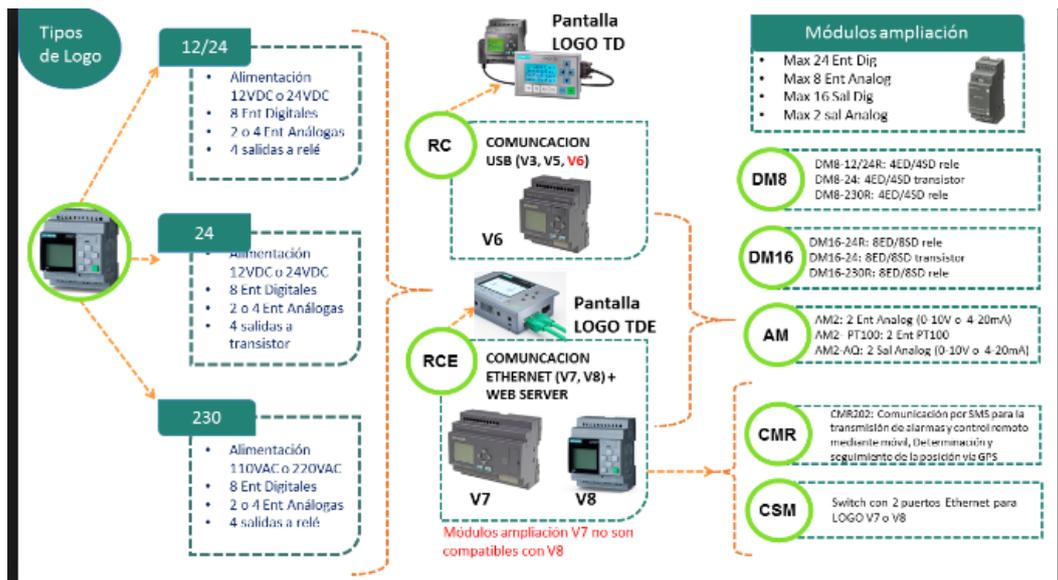
Vue d'un câble SR2CBL01 de Schneider

Caractéristiques techniques :

Catégorie	API- Câble
Modèle	Câble de connexion entre PC et module logique / Longueur : 3m
Référence	SR2 CBL01
Convient pour	Module logique ZELIO LOGIC Schneider Electric

Caractéristiques techniques du câble SR2CBL01

B.3 Automate LOGO ! 12/24 RCE



Types d'automates LOGO !

Caractéristiques techniques :

LOGO! 12/24RCE	
Alimentation :	
Tension d'entrée	12/24 V CC
Plage admissible	10,8 V CC à 28,8 V CC
Protection contre l'inversion de polarité	Oui
Consommation de courant • 12 V CC • 24 V CC	• 30 mA à 140 mA • 15 mA à 90 mA
Temps de maintien en cas de défaillance de la tension • 12 V CC • 24 V CC	• Typ. 2 ms • Typ. 5 ms
Perte de puissance • 12 V CC • 24 V CC	• 0,4 W à 1,7 W • 0,4 W à 2,2 W
Sauvegarde de l'horloge temps réel à 25 °C	Typ. 20 jours
Précision de l'horloge temps réel	Typ. ± 2 s/jour
Séparation galvanique	Non
Entrées TOR :	
Nombre	8
Séparation galvanique	Non
Nombre d'entrées rapides	4 (I3, I4, I5, I6)
Fréquence d'entrée • Entrée normale • Entrée rapide	• max. 4 Hz • max. 5 kHz
Tension continue maximale admissible	28,8 V-
Tension d'entrée L+ • Signal 0 • Signal 1	• < 5 V CC • > 8,5 V CC
Courant d'entrée • Signal 0 • Signal 1	< 0,88 mA (I3 à I6) < 0,07 mA (I1, I2, I7, I8) > 1,5 mA (I3 à I6) > 0,12 mA (I1, I2, I7, I8)
Temps de retard • 0 à 1 • 1 à 0	• Typ. 1,5 ms < 1,0 ms (I3 à I6) • Typ. 1,5 ms < 1,0 ms (I3 à I6)
Longueur de câble (non blindé)	Max. 100 m
Entrées analogiques :	

Nombre	4 (I1=AI3, I2=AI4, I7=AI1, I8=AI2)
Plage	0 V CC à 10 V CC impédance d'entrée 72 kΩ
Temps de cycle pour création des valeurs analogiques	300 ms
Longueur de câble (blindé et torsadé)	Max. 10 m
Limite d'erreur	± 1,5% à FS
Sorties TOR :	
Nombre	4
Type des sorties	Sorties à relais
Séparation galvanique	Oui
Par groupes de	1
Commande d'une entrée TOR	Oui

Caractéristiques techniques de l'automate LOGO !

Interface LOGO ! TDE (afficheur de texte avec interfaces Ethernet) :

Caractéristiques mécaniques :

Clavier Ecran	Clavier à membranes à 10 touches Ecran graphique FSTN 160 x 96 (colonnes x lignes), rétro-éclairage à LED (blanc/ambre/rouge)
------------------	--

Caractéristiques mécaniques de l'afficheur LOGO ! TDE

Caractéristiques techniques :

Alimentation :	
Tension d'entrée	24 V CA /V CC 12 V CC
Plage admissible	20,4 V CA à 26,4 V CA 10,2 V CC à 28,8 V CC
Fréquence de réseau admissible	47Hz à 63 Hz
Consommation (Ethernet et rétro-éclairage blanc actif) • 12 V CC • 24 V CC • 24 V CA	• Typ. 145 mA • Typ. 70 mA • Typ. 75 mA
Indice de protection :	IP20 pour LOGO! TDE à l'exception du panneau avant IP65 pour le panneau avant de LOGO! TDE
Port de communication :	
Performance Ethernet	Deux interfaces Ethernet avec vitesse de transmission des données de 10/100 M duplex/semi duplex
Distance de raccordement	Max. 30 m

Caractéristiques techniques de l'afficheur LOGO ! TDE

ANNEXE C

Programmation sous différents logiciels pour le contrôle de la maquette grâce aux API

- | |
|---|
| C.1 Programmation avec l'automate Siemens S7-300 |
| C.2 Programmation avec l'automate ZELIO LOGIC |
| C.3 Programmation avec l'automate LOGO ! 12/24 RCE |

C.1 Programmation avec l'automate Siemens S7-300

- **Tableau d'affectation des entrées/sorties des manipulations effectuées avec l'automate Siemens S7-300 :**

Adresse	Entrée TOR
124.0	Bouton reset
124.2	Bouton marche
124.4	Entrée LA
124.6	Entrée HA
125.0	Entrée LB
125.2	Entrée HB
Adresse	Sortie TOR
124.0	EVA
124.2	EVB
124.3	AGT
124.5	Pompe
125.0	Voyant 1
125.2	Voyant 2
125.4	Voyant 3
125.6	Voyant 4
Adresse	Entrée analogique
128	Capteur1: Entrée analogique cuve A
130	Capteur2: Entrée analogique cuve B

- **Fonction scale**

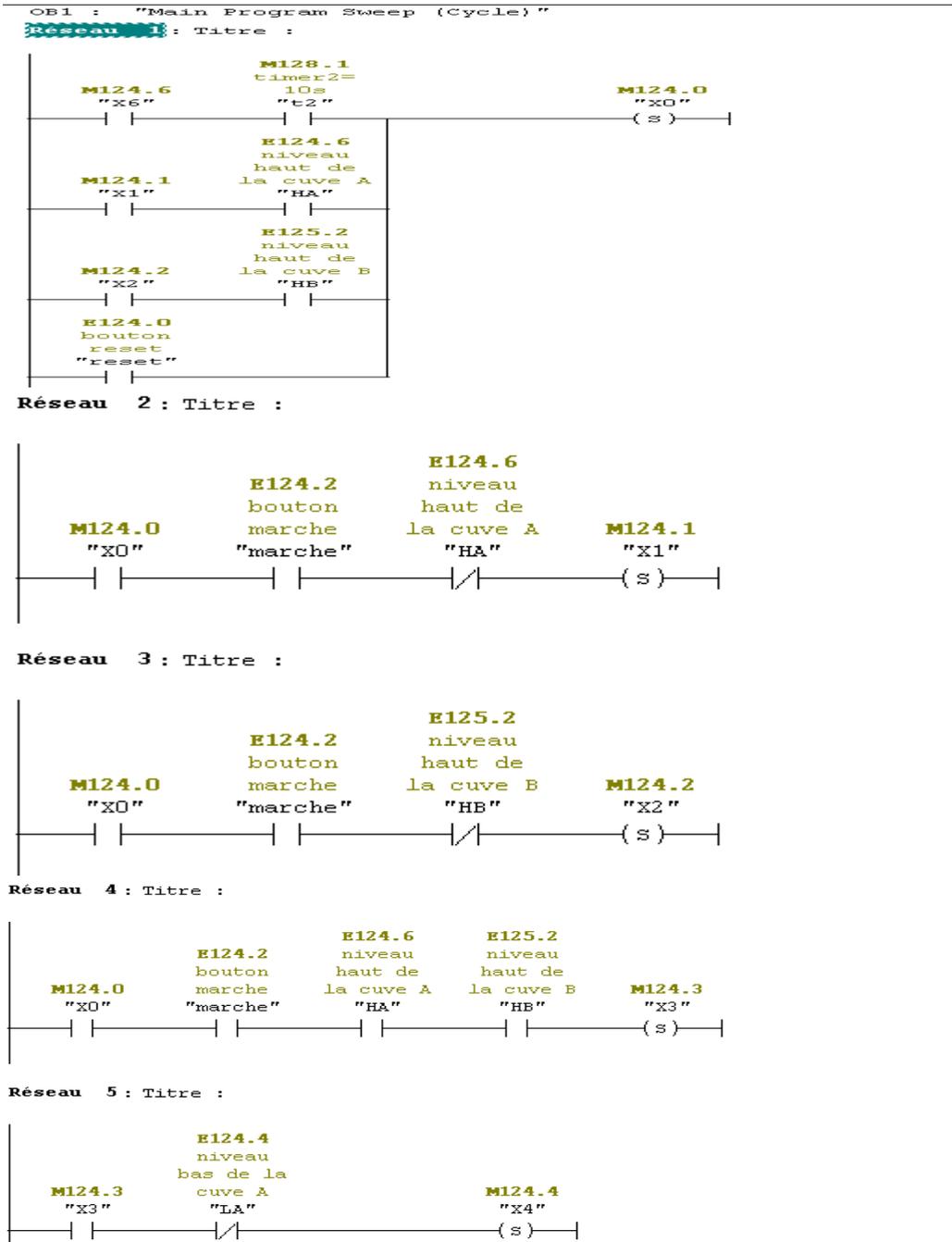
La fonction « SCALE » prend une valeur entière (IN) et la convertit selon l'équation ci-après en une valeur réelle exprimée en unités physiques, comprises entre une limite inférieure (LO_LIM=0.0) et une limite supérieure (HI_LIM=30.0) :

$$\text{OUT} = [((\text{FLOAT}(\text{IN}) - \text{K1}) / (\text{K2} - \text{K1})) * (\text{HI_LIM} - \text{LO_LIM})) + \text{LO_LIM}]$$

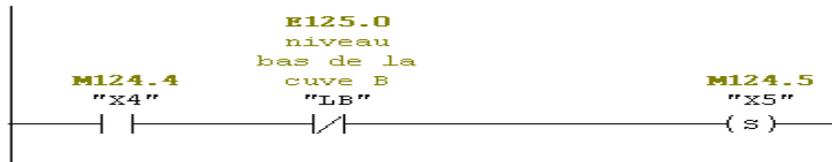
Le résultat est écrit dans OUT, les constantes K1 et K2 sont définies selon que la valeur d'entrée est bipolaire ou unipolaire.

- ✓ Bipolaire : La valeur entière d'entrée est supposée être comprise entre -27648 et 27648, donc : K1 = -27648.0 et K2 = +27648.0
- ✓ Unipolaire : La valeur entière d'entrée est supposée être comprise entre 0 et 27648, donc : K1 = 0.0 et K2 = +27648.0

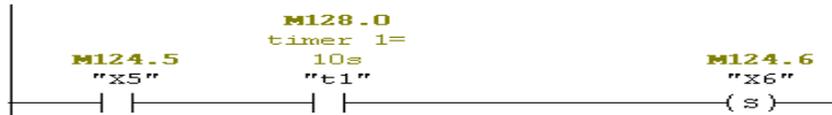
- Le bloc d'organisation OB1 de la première manipulation TOR programmé en langage Ladder :



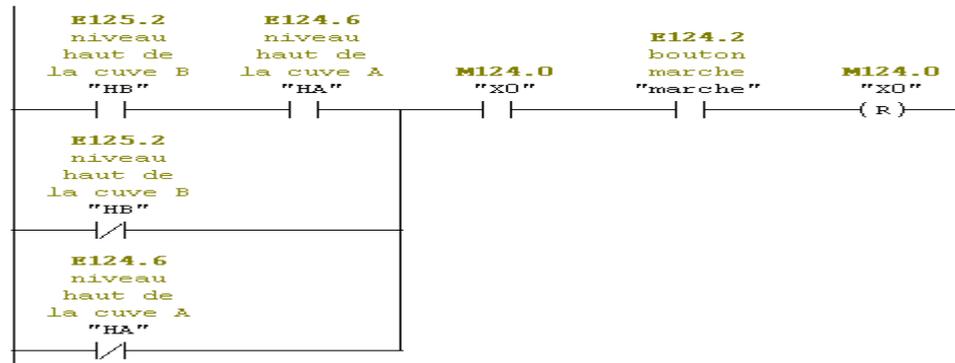
Réseau 6 : Titre :



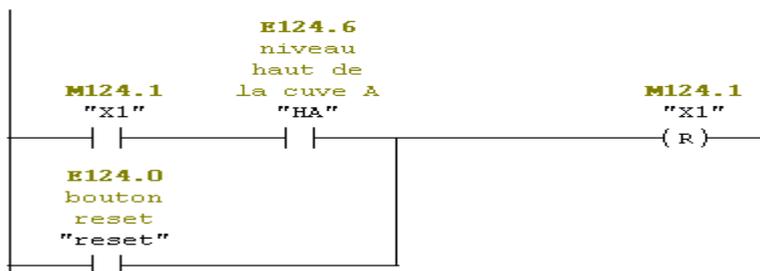
Réseau 7 : Titre :



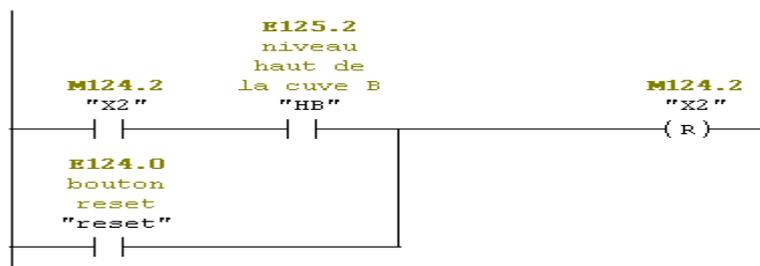
Réseau 8 : Titre :



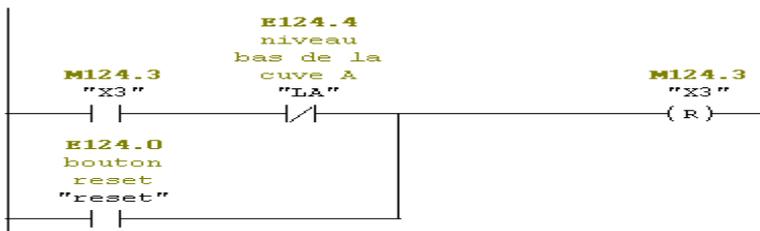
Réseau 9 : Titre :



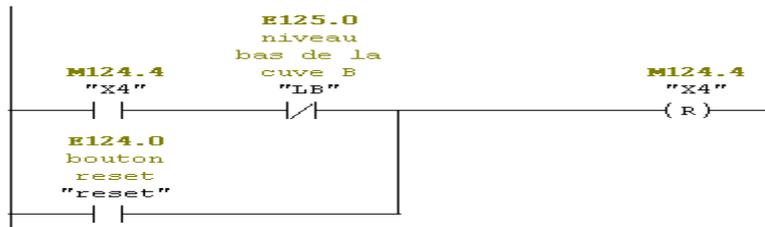
Réseau 10 : Titre :



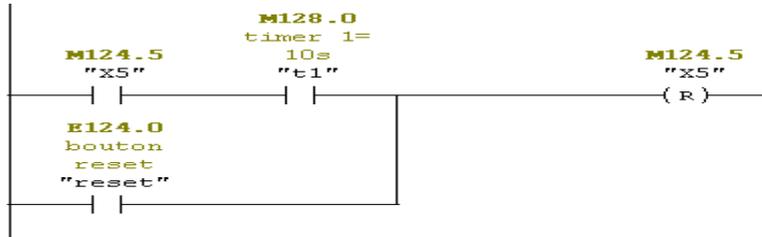
Réseau 11 : Titre :



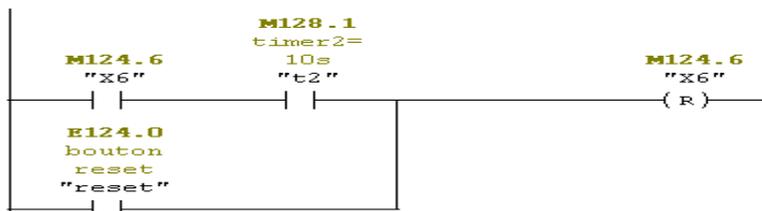
Réseau 12 : Titre :



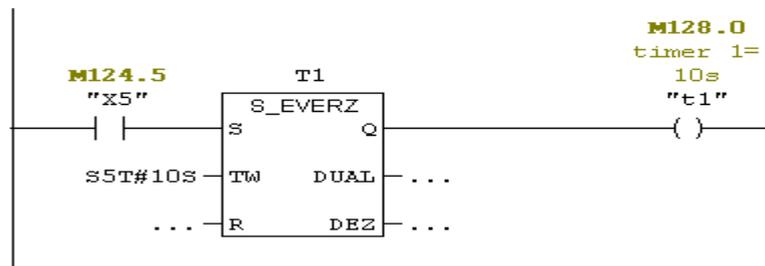
Réseau 13 : Titre :



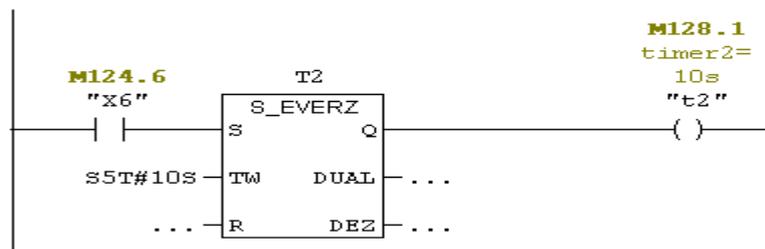
Réseau 14 : Titre :



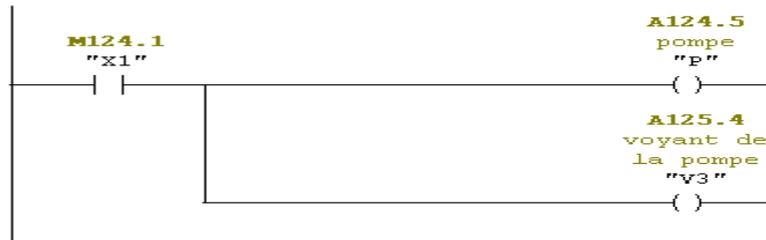
Réseau 15 : Titre :



Réseau 16 : Titre :



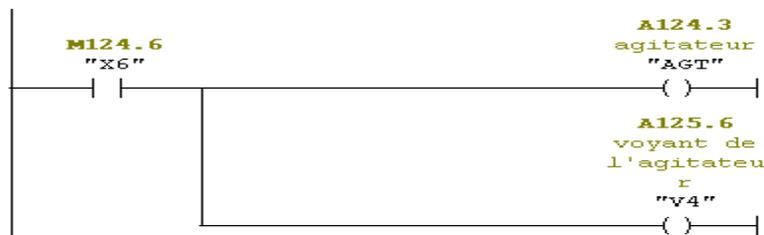
Réseau 17 : Titre :



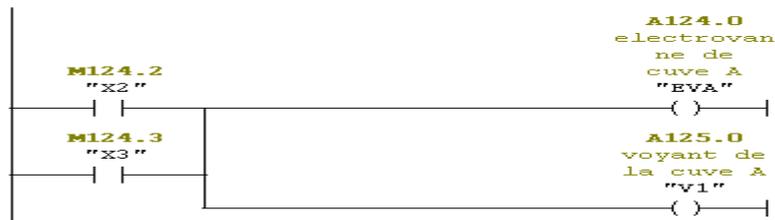
Réseau 18 : electrovanne de cuve B



Réseau 19 : agitateur



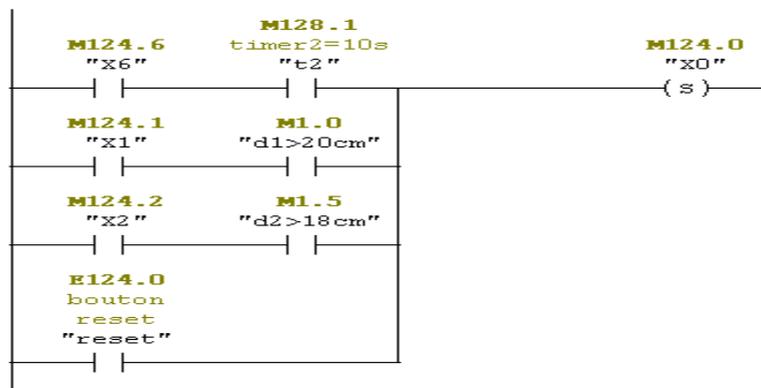
Réseau 20 : electrovanne de cuve A



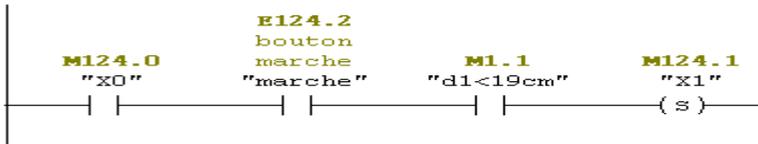
- Le bloc d'organisation OB1 de la deuxième manipulation analogique programmé en langage Ladder

OB1 : "Main Program Sweep (Cycle)"

Réseau 1 : Titre :



Réseau 2 : Titre :



Réseau 3 : Titre :



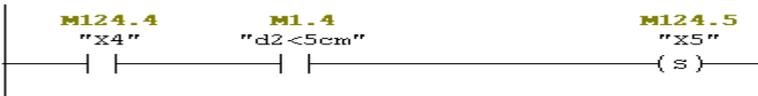
Réseau 4 : Titre :



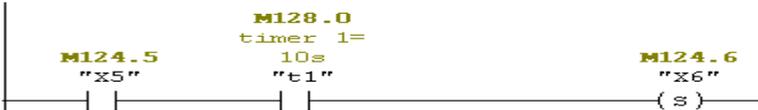
Réseau 5 : Titre :



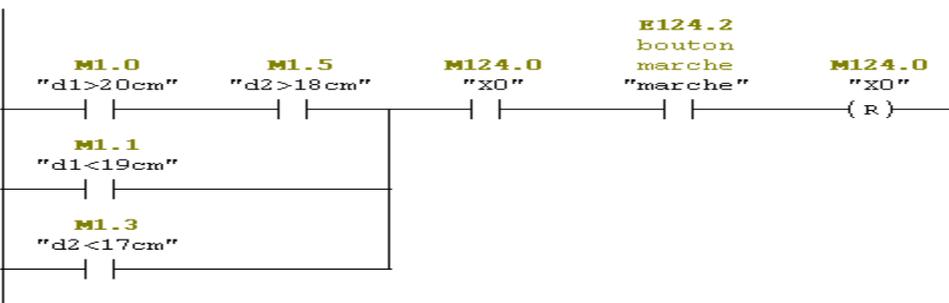
Réseau 6 : Titre :



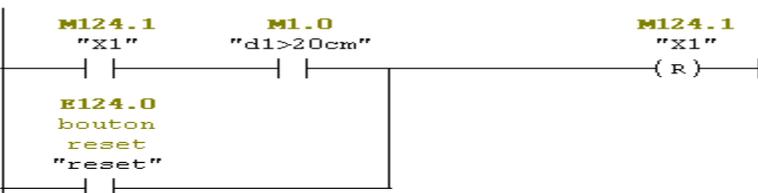
Réseau 7 : Titre :



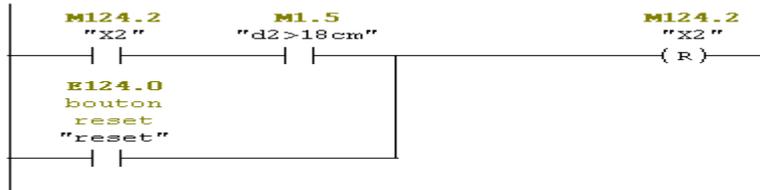
Réseau 8 : Titre :



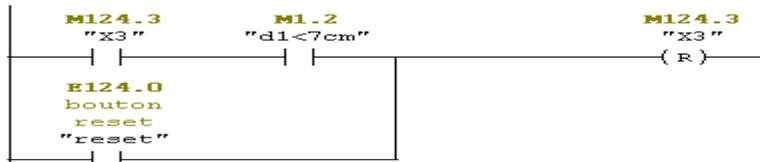
Réseau 9 : Titre :



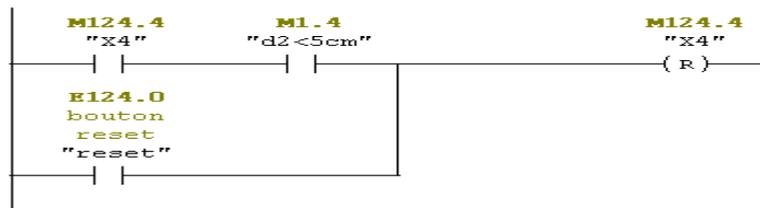
Réseau 10 : Titre :



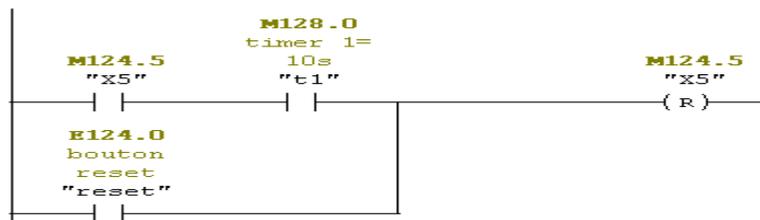
Réseau 11 : Titre :



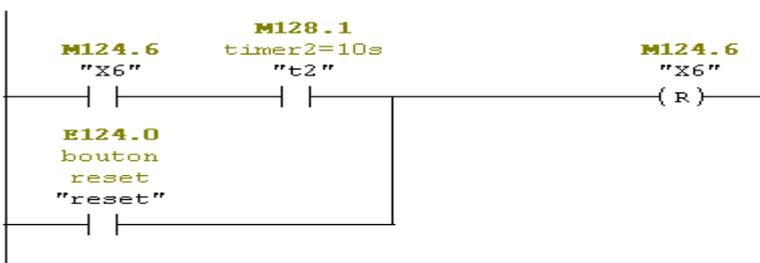
Réseau 12 : Titre :



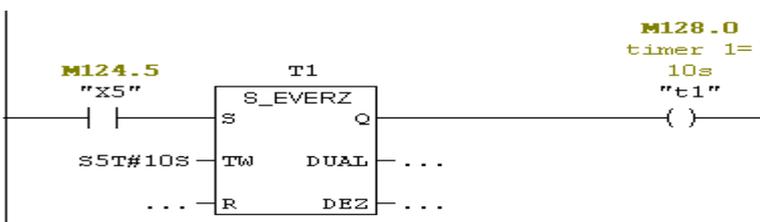
Réseau 13 : Titre :

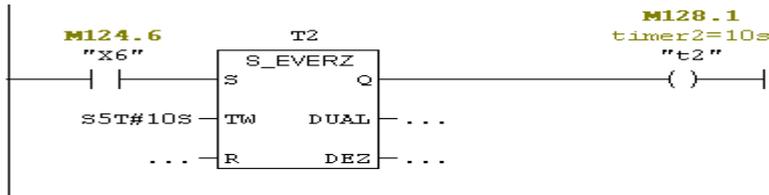


Réseau 14 : Titre :

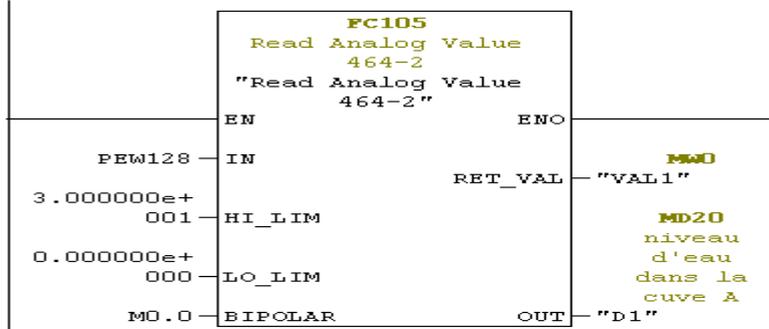


Réseau 15 : Titre :

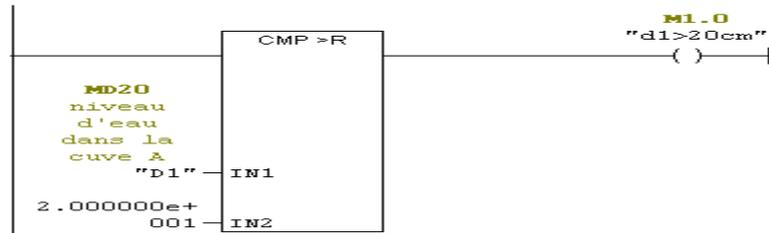




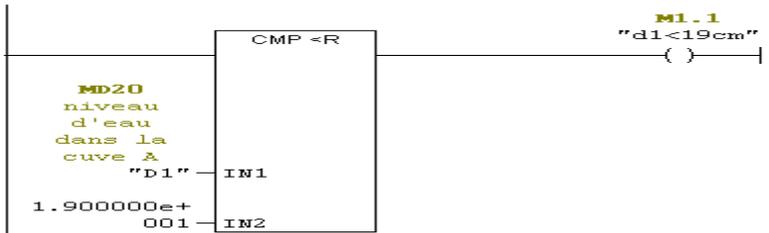
Réseau 17 : Titre :



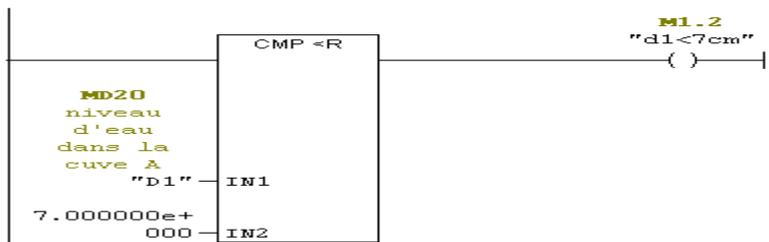
Réseau 18 : Titre :



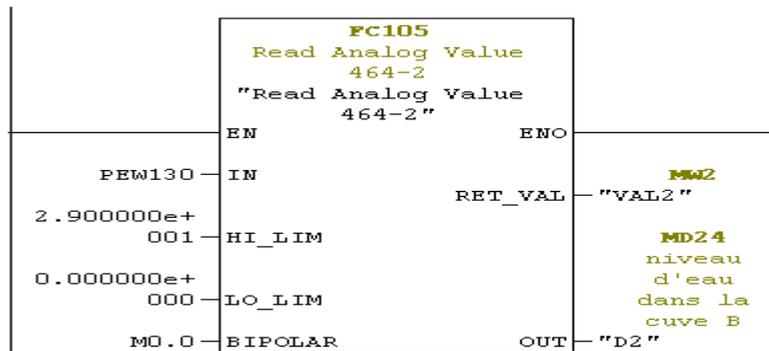
Réseau 19 : Titre :



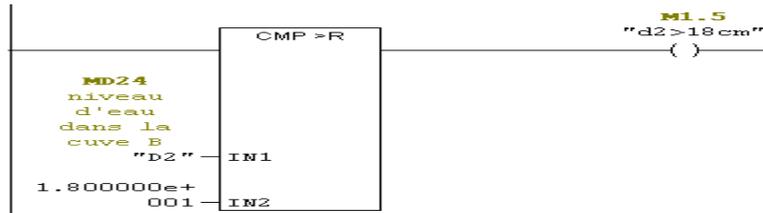
Réseau 20 : Titre :



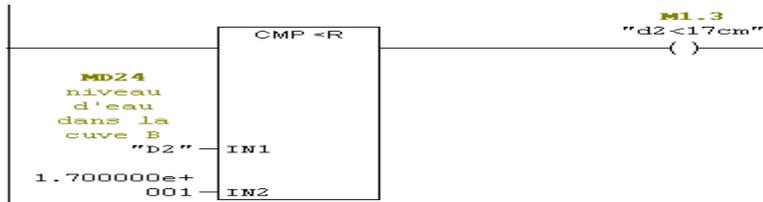
Réseau 21 : Titre :



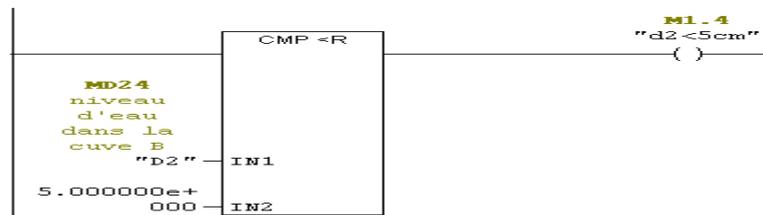
Réseau 22 : Titre :



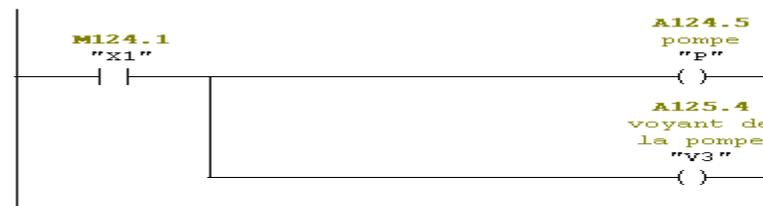
Réseau 23 : Titre :



Réseau 24 : Titre :



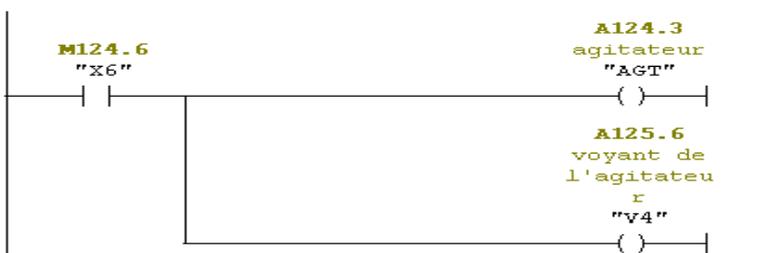
Réseau 25 : Titre :



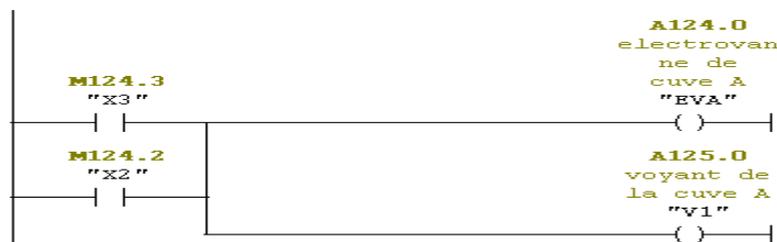
Réseau 26 : Titre :



Réseau 27 : Titre :



Réseau 28 : electrovanne de cuve A

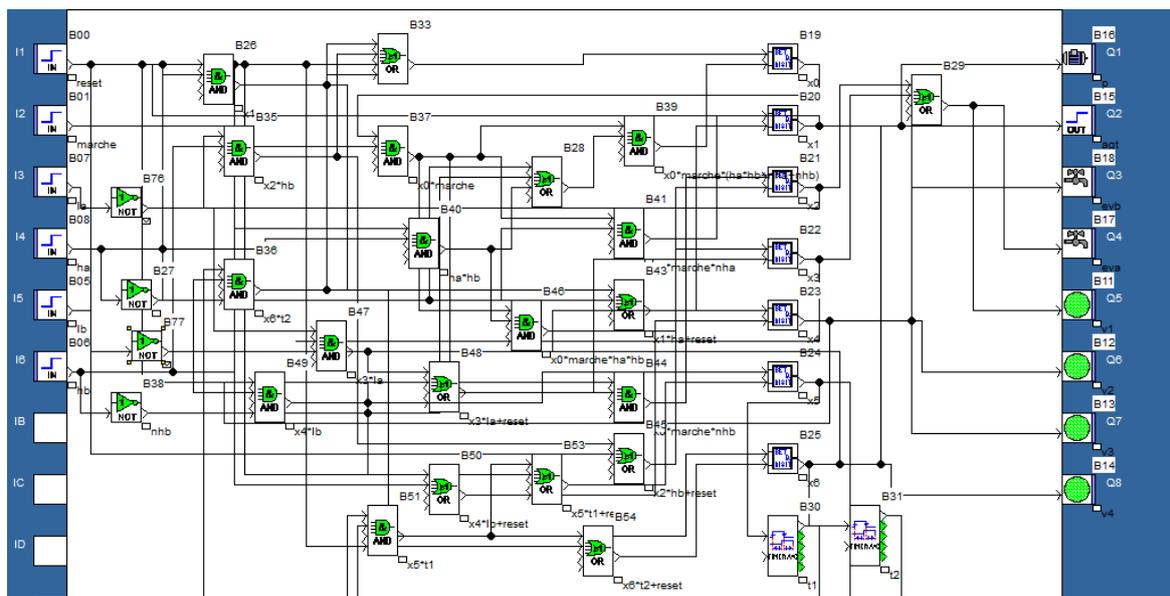


C.2 Programmation avec l'automate ZELIO LOGIC

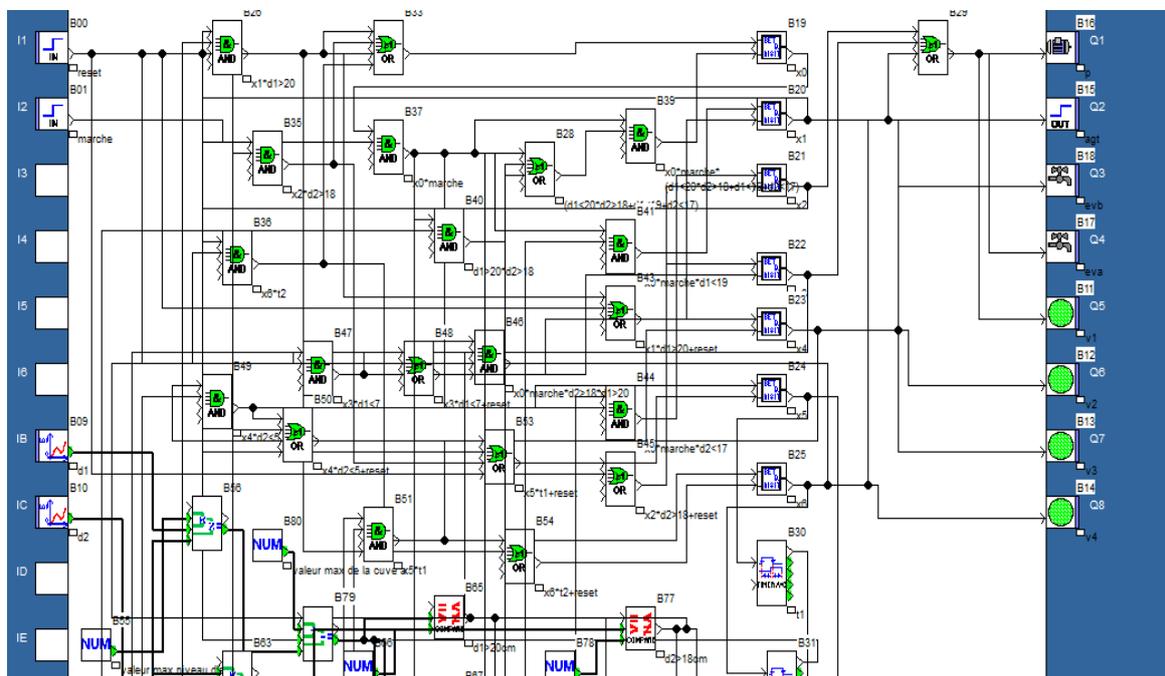
- Tableau d'affectation des entrées/sorties des manipulations effectuées avec l'automate ZELIO LOGIC

Adresse	Entrée TOR
I1	Bouton reset
I2	Bouton marche
I3	Entrée LA
I4	Entrée HA
I5	Entrée LB
I6	Entrée HB
Adresse	Sortie TOR
Q1	Pompe
Q2	Agt
Q3	Evb
Q4	Eva
Q5	Voyant 1
Q6	Voyant 2
Q7	Voyant 3
Q8	Voyant 4
Adresse	Entrée analogique
IB	Capteur1: Entrée analogique cuve A
IC	Capteur2: Entrée analogique cuve B

- Le programme de la première manipulation TOR programmé en langage FBD :



- Le programme de la deuxième manipulation analogique programmé en langage FBD :



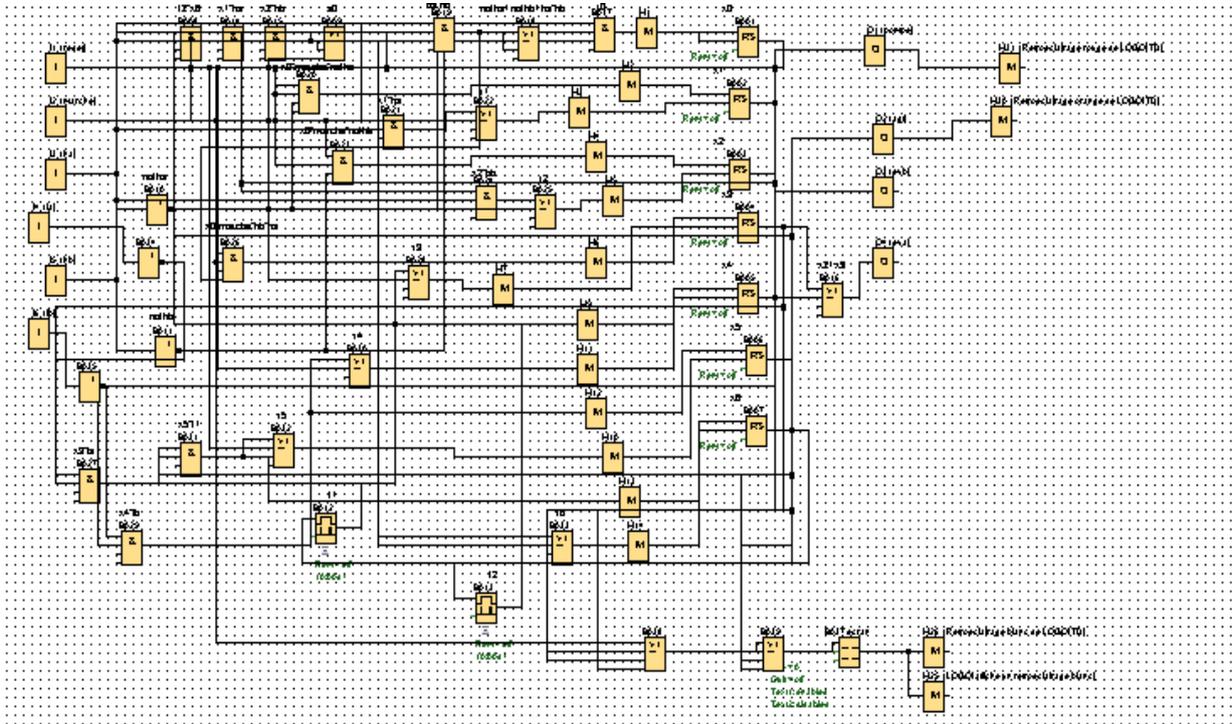
C.3 Programmation avec l'automate LOGO ! 12/24 RCE

- Tableau d'affectation des entrées/sorties des manipulations effectuées avec l'automate LOGO ! 12/24 RCE

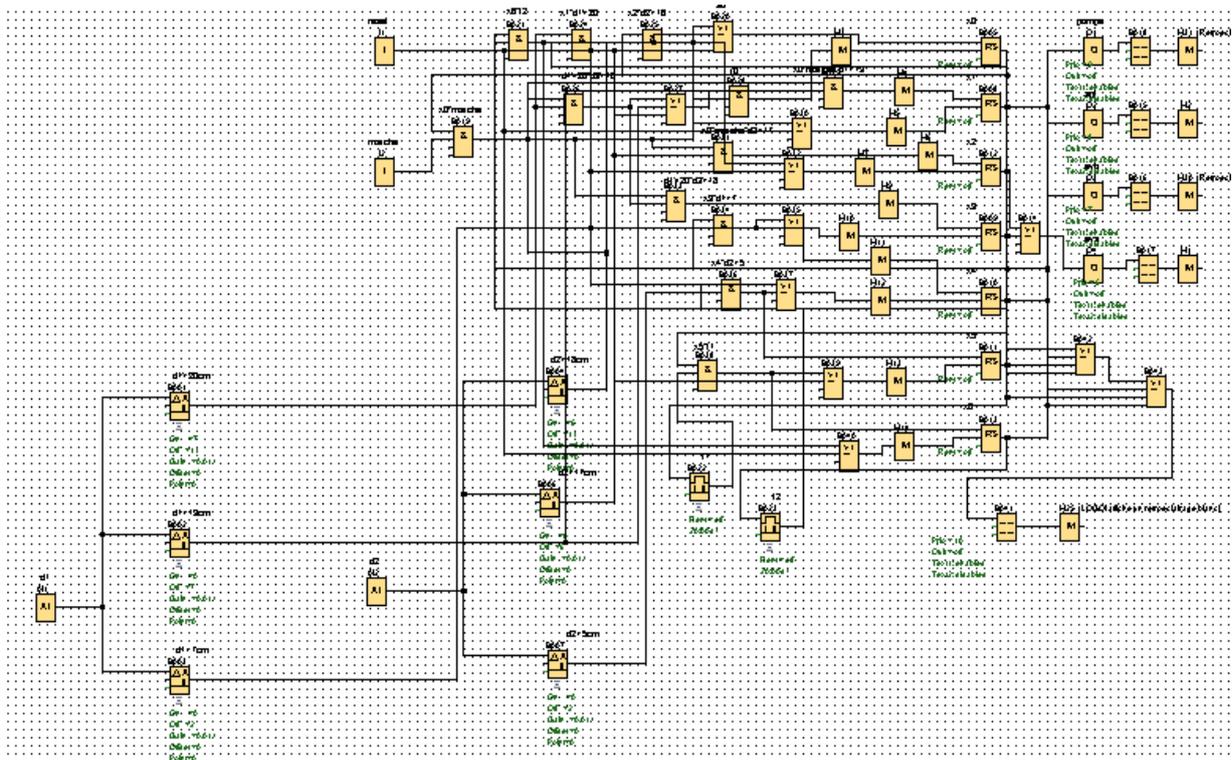
Adresse	Entrée TOR
I1	Bouton reset
I2	Bouton marche
I3	Entrée HA
I4	Entrée LA
I5	Entrée HB
I6	Entrée LB
Adresse	Entrée analogique
I7	Capteur1: Entrée analogique cuve A
I8	Capteur2: Entrée analogique cuve B
Adresse	Sortie TOR
Q1	Voyant 1/ Pompe
Q2	Voyant 2/ Agt

Q3	Voyant 3/ Evb
Q4	Voyant 4/ Eva

- Le programme de la première manipulation TOR programmé en langage FBD :



- Le programme de la deuxième manipulation analogique programmé en langage FBD :



ANNEXE D

Code ARDUINO

```
/* Code pour un capteur à ultrasons HC-SR04*/  
  
/* Constantes pour les broches */  
const byte TRIGGER_PIN1 = 5; // Broche TRIGGER A  
const byte ECHO_PIN1 = 6; // Broche ECHO A  
const byte TRIGGER_PIN2 = 10; // Broche TRIGGER B  
const byte ECHO_PIN2 = 11; // Broche ECHO B  
int v=3; //broche pwm1 (pour stocker la valeur analogique)  
int y=9; //broche pwm2  
  
/* Constantes pour le timeout */  
const unsigned long MEASURE_TIMEOUT = 25000UL; // 25ms = ~8m à 340m/s  
  
/* Vitesse du son dans l'air en mm/us */  
const float SOUND_SPEED = 340.0 / 1000;  
  
/*déclaration des constantes de distances finales (niveau d'eau) pour stocker les valeurs */  
const float distance_finale_1=0;  
const float distance_finale_2=0;  
  
/** Fonction setup() */  
void setup() {  
  /* Initialise le port série */  
  Serial.begin(9600);  
  
  /* Initialise les broches */  
  pinMode(TRIGGER_PIN1, OUTPUT);  
  digitalWrite(TRIGGER_PIN1, LOW); // La broche TRIGGER A doit être à LOW au repos  
  pinMode(ECHO_PIN1, INPUT);  
  delay(5);  
  pinMode(TRIGGER_PIN2, OUTPUT);  
  digitalWrite(TRIGGER_PIN2, LOW); // La broche TRIGGER B doit être à LOW au repos
```

```

pinMode(ECHO_PIN2, INPUT);
delay(5);
}
/** Fonction loop() */
void loop() {

/* 1. Lance une mesure de distance en envoyant une impulsion HIGH de 10µs sur la broche
TRIGGER A/B */

digitalWrite(TRIGGER_PIN1, HIGH);
delayMicroseconds(10);
digitalWrite(TRIGGER_PIN1, LOW);

digitalWrite(TRIGGER_PIN2, HIGH);
delayMicroseconds(10);
digitalWrite(TRIGGER_PIN2, LOW);

/* 2. Mesure le temps entre l'envoi de l'impulsion ultrasonique et son écho (si il existe) */
long measure1 = pulseIn(ECHO_PIN1, HIGH, MEASURE_TIMEOUT);
long measure2 = pulseIn(ECHO_PIN2, HIGH, MEASURE_TIMEOUT);

/* 3. Calcul la distance à partir du temps mesuré */
float distance_1 = measure1 / 2.0 * SOUND_SPEED;
float distance_2 = measure2 / 2.0 * SOUND_SPEED;
float distance_finale_1 =300-distance_1;
float distance_finale_2 =250-distance_2;

/* Affiche les résultats en cm */
Serial.print(F("Distance cuve a: "));
Serial.print( distance_finale_1/ 10);
Serial.print(F("cm, "));
delay(100);
if((distance_1 / 10)<28){
analogWrite(y,(distance_finale_1/10)*255/28);

```

```
delay(900);}

if((distance_finale_1 / 10)>=28){
  analogWrite(v,255);
  delay(900);}

/* Affiche les résultats en cm */

Serial.print(F("Distance cuve b: "));
Serial.print( distance_finale_2/ 10);
Serial.print(F("cm, "));

delay(100);

if((distance_finale_2/ 10)<24){
  analogWrite(y,(distance_finale_2 / 10)*255/24);
  delay(900);}

if((distance_finale_2/ 10)>=24){
  analogWrite(v,255);
  delay(900);}

/* Délai d'attente pour éviter d'afficher trop de résultats à la seconde */
delay(1200) ;

}
```

ANNEXE E

Schéma électrique de l'installation avec l'automate LOGO !

