

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique d'Alger



المدرسة الوطنية متعددة التقنيات
Ecole Nationale Polytechnique

Département d'Automatique

Mémoire de Projet de Fin d'Etudes
pour l'obtention du diplôme d'ingénieur d'état en Automatique

Commande, Simulation et Supervision d'un prototype de station de pompage avec Automate Programmable Industriel SIEMENS S7-1200

Réalisé par :

Mohamed Amine KOUSSA
Mohamed Ramzi BOUZA

Sous l'examen du jury :

Président	M. Djamel BOUDANA	Pr. ENP
Rapporteur	M. El Madjid BERKOUK	Pr. ENP
Examineur	M. Hakim ACHOUR	MCA. ENP

Présenté et soutenu publiquement le :
13 Juillet 2021

ENP 2021

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique d'Alger



المدرسة الوطنية متعددة التقنيات
Ecole Nationale Polytechnique

Département d'Automatique

Mémoire de Projet de Fin d'Etudes
pour l'obtention du diplôme d'ingénieur d'état en Automatique

Commande, Simulation et Supervision d'un prototype de station de pompage avec Automate Programmable Industriel SIEMENS S7-1200

Réalisé par :

Mohamed Amine KOUSSA
Mohamed Ramzi BOUZA

Sous l'examen du jury :

Président	M. Djamel BOUDANA	Pr. ENP
Rapporteur	M. El Madjid BERKOUK	Pr. ENP
Examineur	M. Hakim ACHOUR	MCA. ENP

Présenté et soutenu publiquement le :
13 Juillet 2021

ENP 2021

ملخص

أطروحة مشروع نهاية الدراسة تغطي عملية أتمتة محطة ضخ المياه بإستعمال جهاز تحكم منطقي قابل للبرمجة من **سيمنس** ، **الاس-7-1200** بالإضافة لواجهة بين الإنسان والآلة من **سيمنس السيماتيک كومفورٹ ٹي بي 900**. لغرض الإشراف على المحطة

من خلال هذا المشروع ، سننظم عملنا خطوة بخطوة من أجل تغطية جميع مراحل دراسة معدات الأجهزة ، تصميم التحكمات الآلية ، تنفيذ البرامج على **تي بي آي إي بورتال نسخة 15.1** والإشراف الكامل لمحطة ضخ المياه ومعداتنا عن طريق الواجهة

كلمات مفتاحية : بي أل سي ، إتش إم آي ، الأتمتة ، تحكم آلي ، تي بي آي إي بورتال نسخة 15.1 ، سيمنس ، برمجة ، محطة ، صناعة

Abstract

The thesis of our project covers the automation process of a pumping station based on a programmable logic controller from **SIEMENS**, the **S7-1200** combined with a human machine interface from **SIEMENS** the **SIMATIC Comfort TP900** in order to supervise the station.

Through this project, we will organize our work step by step with the aim of covering the of all the hardware equipment review, the design of the automatic controllers, the implementation of automation programs on **TIA Portal V15.1** and supervision of the water pumping station by means of the HMI.

Keywords : PLC, HMI, Automation, Control, TIA Portal, SIEMENS, Programing, Station, Industry.

Résumé

Le mémoire de notre projet de fin d'étude enveloppe le processus d'automatisation d'une station de pompage d'eau en utilisant un automate programmable de **SIEMENS**, le **S7-1200** ainsi qu'une interface homme machine de **SIEMENS**, la **SIMATIC Comfort TP900** dédiée à la supervision.

A travers ce projet, nous organiserons notre travail étape par étape dans le but de couvrir toutes les phases d'étude de l'équipement matériel, de la synthèse de commande du processus, l'implémentation sur **TIA Portal V15.1** et la supervision de la station de pompage à travers l'IHM.

Mots clés : API, IHM, Automatisation, Commande, TIA Portal, SIEMENS, Programmation, Station, Industrie.

Remerciements

On remercie Dieu le Tout Puissant de nous avoir accordé le courage, la santé, et la patience pendant les moments difficiles.

*On remercie notre encadreur, Monsieur **El Madjid BERKOUK** de nous avoir aidé dans notre travail et d'avoir été disponible pour nous.*

On remercie également les membres du jury pour l'intérêt qu'ils ont porté à notre travail en acceptant de l'examiner.

On remercie chaleureusement nos familles et tous nos amis pour leur soutien et leurs encouragements.

On souhaite aussi remercier tous les enseignants de l'École Nationale Polytechnique d'Alger, particulièrement nos professeurs d'Automatique pour la quantité et la qualité de savoir et des capacités intellectuelles qu'ils nous ont transmis durant ces trois années.

En espérant que tous ceux qui ont contribué de près ou de loin à ce travail pourront trouver ici nos remerciements les plus honnêtes.

Dédicaces

Je dédie ce travail d'abord à moi même et je me félicite d'avoir accompli mon premier projet en tant qu'ingénieur.

À mes chers parents, Rabah et Khalida, merci de m'avoir soutenu dans le bien et dans le pire depuis le départ de cette aventure.

À ma soeur adorée Yasmine pour tes encouragemetns et tes conseils en or.

À mes petits frères Adem et Ayoub.

À mon cher binôme Ramzy, once a friend, always a friend!

À Katia, merci d'être ma vraie amie pendant ces 5 ans.

À tous mes amis de Polytech, Younes, Mokerane, Anis, Soheib, Oussama, Rahim, Mustapha, nos souvenirs ensemble seront gravées sur mon coeur à l'éternité.

Au CAP pour m'avoir offert de nouvelles perspectives.

Amine

Dédicaces

J'ai fait des efforts dans ce projet. Cependant, cela n'aurait pas été possible sans l'aimable soutien et l'aide de nombreuses personnes et organisations.

Je tiens à leur adresser mes sincères remerciements.

Je suis extrêmement reconnaissant à mes parents pour leur amour, leurs prières, leurs soins et leurs sacrifices pour m'avoir éduqué et préparé pour mon avenir.

J'aimerais exprimer mes plus grands remerciements, ma sincère gratitude et mon amour inconditionnel à ma bien-aimée Meryem pour ses encouragements et pour sa confiance en moi. Elle m'a fait sourire d'une manière que je ne pourrais jamais imaginer.

Je voudrais remercier mon binôme et ami Amine, c'était un honneur d'être ton binôme dans ce projet, tu es une personne incroyable.

Je tiens à remercier mon frère et mes soeurs pour leur amour et leur soutien durant toute ma vie. je suis très content de les avoir à mon coté.

J'adresse également mes plus sincères remerciements à tous mes proches Rahim, Soheib, Chihab, Anis, Oussama, Younes, Katia, Manel, Amira, Meriem, Nacim, OURABAH, Abdennour, Yacine qui m'ont toujours soutenu et encouragé au cours de la réalisation de ce mémoire.

À mon cher Yakoub qui m'a tutoré durant mes années à Polytech, grâce à lui je suis arrivé où je suis aujourd'hui.

Ramzi

Table des matières

Table des figures	9
Liste des tableaux	10
Liste des abréviations	10
Introduction générale	15
1 Généralités sur les API, IHM et Variateurs de vitesse	16
1.1 Introduction	17
1.2 Siemens	17
1.2.1 Siemens	17
1.2.2 Siemens et l'automatisation	17
1.2.3 Simatic	17
1.3 Les automates programmables industriels	18
1.3.1 Définition	18
1.3.2 Historique	18
1.3.3 Architecture des API	18
1.3.4 Langages de programmation des API	20
1.3.5 Choix d'un API	23
1.3.6 Les Avantages des automates	24
1.3.7 Les inconvénients des automates	24
1.4 IHM (Interface Homme-Machine)	25
1.4.1 Définition générale	25
1.4.2 L'utilisation des IHM	25
1.4.3 La différence entre IHM et SCADA	26
1.4.4 Les avantages des IHM	26
1.4.5 SIMATIC HMI	26
1.5 Variateurs de vitesse	27
1.5.1 Définition	27
1.5.2 Historique	27
1.5.3 Principe de fonctionnement	28
1.5.4 Avantages des variateurs de vitesse	29
1.5.5 Protection des variateurs de vitesse contre les creux de tension	29
1.5.6 Les fonctions des variateurs de vitesse	30
1.5.7 Inconvénients des variateurs de vitesse	31
1.6 Conclusion	31

TABLE DES MATIÈRES

2	Étude du matériel de la station de pompage	32
2.1	Introduction	33
2.2	Prototype de la station de pompage	33
2.2.1	Description	33
2.2.2	Schéma électrique de la station de pompage	34
2.2.3	Assemblage et câblage de la station de pompage	35
2.3	Simatic S7-1200	36
2.3.1	Description	36
2.3.2	Les Modules de communication	36
2.3.3	La mémoire	37
2.3.4	Les modules d'entrées/sorties	37
2.3.5	Régulation	37
2.3.6	Communication industrielle	37
2.4	Modules ajoutés à la CPU	38
2.4.1	L'alimentation (SITOP PSU100S)	38
2.4.2	Le module analogique	39
2.5	Écran IHM	40
2.5.1	SIMATIC HMI Comfort	40
2.5.2	SIMATIC HMI 6AV2124-0JC01-0AX0	40
2.6	Variateur de vitesse ATV12H037M2	41
2.6.1	Description	41
2.6.2	Détails du produit	42
2.6.3	Applications	42
2.7	Unité de pompage (Moteur électrique & pompe)	43
2.7.1	Pompe centrifuge	43
2.7.2	Moteur asynchrone	43
2.7.3	Unité de pompage du prototype de la station	44
2.8	Instrumentation et appareils de mesure	45
2.8.1	Capteurs	45
2.9	interfaces d'entrée et de sortie	48
2.9.1	Interfaces d'entrée	48
2.9.2	Interfaces de sortie	51
2.10	Conclusion	52
3	Modélisation et Synthèse de commande	53
3.1	Introduction	54
3.2	Modélisation mathématique	54
3.2.1	Représentation d'État du modèle mathématique	55
3.2.2	Fonction de transfert et synthèse de commande linéaire	56
3.2.3	Synthèse de commande robuste (Correcteur en mode glissant)	58
3.3	Simulation et interprétation des résultats	59
3.3.1	Simulation avec correcteur PI	59
3.3.2	Simulation avec correcteur en mode glissant	63
3.4	Conclusion	66

4	Implémentation et supervision de la station de pompage	67
4.1	Introduction	68
4.2	Cahiers de Charges	68
4.2.1	Cahier de Charges I : Mode opératoire automatique	68
4.2.2	Cahier de Charges II : Mode opératoire cyclique	71
4.2.3	Équations d'état	73
4.3	Implémentation des cahiers de charges sur TIA Portal	75
4.3.1	Programme de l'API	75
4.3.2	Programme de l'IHM	84
4.3.3	Archivage des données	89
4.4	Conclusion	91
	Conclusion générale	92
5	Annexe A :	
	Initiation à TIA Portal	93
	Logiciels de programmation et d'Interfaces	94
	Environnement de travail : TIA Portal	94
	Programmation des API avec TIA Portal	94
	Création du programme de l'API avec TIA Portal (STEP 7)	94
	Création de l'IHM avec TIA Portal (WinCC)	108
6	Annexe B :	
	Montage et câblage du ATV12H037M2	117
	Schéma de câblage général	118
	Pour une alimentation monophasée 100-120 V	118
	Pour une alimentation monophasée 200-240 V	118
	Pour une alimentation triphasée 200-240 V	119
	Conditions de montage et de température	119
7	Annexe C :	
	Perspectives	120

Table des figures

1.1	Architecture d'un API	19
1.2	Exemple d'un programme en Ladder	21
1.3	Exemple d'un programme en IL	22
1.4	Exemple d'un programme en FBD	22
1.5	Exemple d'un programme en ST [2]	22
1.6	Exemple d'un GRAFCET	23
1.7	Écran HMI tactile, KTP 400 [21]	27
1.8	Schéma électrique d'un variateur de vitesse "réalisé par Multisim"	28
1.9	Principe de fonctionnement de la régulation de vitesse [7]	30
2.1	Schéma électrique de la station	34
2.2	Câblage de la station	35
2.3	Le Prototype de la station	35
2.4	Simatic S7-1200 [6]	36
2.5	Le Simatic S7-1200 et ses modules de communication [26]	37
2.6	Alimentation SITOP PSU100S (Réf : 6EP1334-2BA20) [8]	39
2.7	Module de sorties analogiques SM 1232 (Réf : 6ES7232-4HD32-0XB0) [1]	39
2.8	SIMATIC HMI 6AV2124-0JC01-0AX0 [5]	41
2.9	Altivar ATV12H037M2 [12]	41
2.10	Pompe centrifuge [32]	43
2.11	Moteur asynchrone [19]	44
2.12	Unité de pompage LEO APm37 [16]	44
2.13	Capteur de distance UM18-51111 [24]	46
2.14	Schéma de câblage du capteur SICK UM18-51111	46
2.15	Débitmètre YF-S201 [29]	47
2.16	Schéma de câblage du débitmètre YF-S201	47
2.17	Disjoncteur bipolaire ENG [18]	48
2.18	Contacteur SIEMENS 3RH2140-1AP00 [25]	49
2.19	Carte Arduino UNO [10]	49
2.20	Filtre Passe-Bas	50
2.21	Module convertisseur de courant – tension [15]	50
2.22	Amplificateur	51
2.23	Relais Schneider Electric RSB2A080BDS [14]	51
3.1	Réponse indicielle du système non-corrigé	57
3.2	Schéma Fonctionnel : Correcteur PI avec Système	59
3.3	Réponse de la FTBF à une consigne	60

TABLE DES FIGURES

3.4	Réponse du système non-linéaire à une consigne	61
3.5	Réponse du système corrigé avec perturbation $\alpha=0\%$	62
3.6	Réponse du système corrigé avec perturbation $\alpha=30\%$	62
3.7	Réponse du système corrigé avec perturbation $\alpha=60\%$	62
3.8	Schéma blocs : Correcteur MG et système	63
3.9	Réponse à une consigne du système avec le correcteur MG	63
3.10	Réponse à une consigne avec perturbation ($\alpha = 0\%$)	64
3.11	Réponse à une consigne avec perturbation ($\alpha = 30\%$)	64
3.12	Réponse à une consigne avec perturbation ($\alpha = 60\%$)	64
3.13	Réponse à une consigne avec perturbation ($\alpha = 80\%$)	65
4.1	Schéma représentatif de la station de pompage	69
4.2	GRAFCET décrivant le Cahier de Charges I	70
4.3	GRAFCET décrivant le Cahier de Charges II	72
4.4	Équation d'activation de l'état x_0	76
4.5	Équation de désactivation de l'état x_0	76
4.6	Équation d'activation de l'état x_1	76
4.7	Équation de désactivation de l'état x_1	77
4.8	Équation de sortie de l'état x_1	77
4.9	Équation d'activation de l'état x_2	77
4.10	Équation de désactivation de l'état x_2	78
4.11	Équation de sortie de l'état x_2	78
4.12	Équation d'activation de l'état x_3	78
4.13	Équation de désactivation de l'état x_3	79
4.14	Équation de sortie de l'état x_3	79
4.15	Fonction de limitation	80
4.16	Fonction d'initialisation	80
4.17	Fonction PID	81
4.18	Fonction SMC	81
4.19	Tolérance	81
4.20	Fonction Scale	82
4.21	Fonction Unscale	82
4.22	Bloc Fonctionnel - Cahier de charges II	83
4.23	Blocs utilisant les Blocs de Données	84
4.24	Vue principale de l'IHM	85
4.25	Vue Entrées de l'IHM	85
4.26	Vue GRAFCET I de l'IHM	86
4.27	Vue GRAFCET II de l'IHM	86
4.28	Vue Contrôle de l'IHM	87
4.29	Vue Station de l'IHM	87
4.30	Vue Alarmes de l'IHM	88
4.31	Vue recettes	89
4.32	Archivage des données	90
4.33	Représentation graphique de la variation de niveau d'eau par rapport au temps	90
5.1	GRAFCET de l'exemple du projet	94

5.2	Partie 1	95
5.3	Partie 1 (suite)	95
5.4	Partie 2	96
5.5	Partie 2 (suite)	96
5.6	Partie 2 (suite)	97
5.7	Partie 2 (suite)	97
5.8	Partie 3	98
5.9	Partie 3 (suite)	99
5.10	Partie 4	99
5.11	Partie 4 (suite)	100
5.12	Partie 4 (suite)	100
5.13	Partie 5	101
5.14	Partie 5 (suite)	102
5.15	Partie 6	103
5.16	Partie 6 (suite)	103
5.17	Partie 6 (suite)	104
5.18	Partie 6 (suite)	104
5.19	Partie 7	105
5.20	Partie 8	106
5.21	Partie 8 (suite)	106
5.22	Partie 8 (suite)	107
5.23	Partie 8 (suite)	107
5.24	Partie 8 (suite)	108
5.25	Partie 1	109
5.26	Partie 1 (suite)	109
5.27	Partie 1 (suite)	110
5.28	Partie 2	110
5.29	Partie 2 (suite)	111
5.30	Partie 2 (suite)	111
5.31	Partie 3	112
5.32	Partie 3 (suite)	112
5.33	Partie 3 (suite)	113
5.34	Partie 3 (suite)	114
5.35	Partie 3 (suite)	114
5.36	Partie 3 (suite)	115
5.37	Partie 4	115
5.38	Partie 4 (Fin)	116
6.1	Câblage du ATV12....F1	118
6.2	Câblage du ATV12....M2	118
6.3	Câblage du ATV12....M3	118

Liste des tableaux

2.1	Caractéristiques techniques de l'unité de pompage APm37	45
2.2	Détails techniques du capteur UM18-51111	46
2.3	Détails de l'interface de sortie du capteur UM18-51111	46
2.4	Détails techniques du débitmètre YF-S201	47
2.5	Détails techniques du contacteur 3RH2140-1AP00	49
3.1	Table significative des paramètres	55
3.2	Les gains du correcteur PI	58

Liste des abréviations

AC	Alternating Current
AI	Analog Input
AO	Analog Output
API	Automate Programmable Industriel
BF	Boucle Fermée
BO	Boucle Ouverte
C	Counter/Compteur
DB	Data Bloc
DC	Direct Current
DI	Digital Input
DO	Digital Output
E/S	Entrée/Sortie
EV	Electro-Vanne
FB	Function Block
FC	Function
FTBF	Function de Transfert en Boucle Fermée
FTBO	Function de Transfert en Boucle Ouverte
FT	Function de Transfert
HMI	Human Machine Interface
IHM	Interface Homme Machine
INCR	Incrémentation
INIT	Initialisation
MAS	Moteur/Machine ASynchrone
MG	Mode Glissant
MLI	Modulation de la Largeur d'Impulsion

M	Mnémonique
NIV	Niveau
OB	Organization Block
PID	Proportion Intégrale Dérivée
PLC	Programable Logic Controller
PWM	Pulse Width Modulation
P	Pompe
Q	Query
RAZ	Remise A Zéro
REF	Reference
S/R	Set/Reset
SMC	Sliding Mode Controller
TIA Portal	Totally Integrated Automation Portal
TOR	Tout Ou Rien
T	Time/Temps
VAR	Variable
V	Voyant

Introduction générale

Depuis l'introduction des automates programmables, le monde industriel a connu une croissance révolutionnaire conduisant à des avancées technologiques très remarquables.

À l'heure actuelle et dans tous les pays, les industries de production ne peuvent se passer des API, car ces dernières offrent des avantages incontournables sur les contraintes présentes dans ce domaine. Ils permettent également d'exempter l'homme du lieu de travail en raison des dangers et contraintes occasionnés par les influences extérieures de l'environnement et du matériel présent dans les établissements industriels.

Les processus d'automatisation nécessitent une gestion très rigoureuse des matériaux afin de garantir un fonctionnement totalement autonome. C'est pour cette raison que dans notre travail, nous avons fait recours à un Automate Programmable qui confère une implémentation très facile de nos spécifications et cahier de tâches ainsi qu'un fonctionnement souple et évolutif.

Notre projet est dédié à l'étude d'une station de pompage d'eau. **Le chapitre 01** évoque des généralités sur les automates programmables en évoquant leur histoire, leur impact dans le présent et leur évolution dans le futur dans le domaine de l'automatisation industrielle.

Le chapitre 02 traite de l'étude des équipements qui constituent la station de pompage, en mentionnant tous les équipements hydrauliques et électriques, les dispositifs d'instrumentation et les capteurs, et enfin les installations de contrôle et de supervision. L'étude du matériel permet d'acquérir une connaissance technique des équipements de la station de pompage et de comprendre son intérêt pour ces derniers.

Le chapitre 03 concerne le développement de l'étude simulée de la station de pompage en établissant d'abord le modèle mathématique qui est l'objet essentiel afin de réaliser la synthèse de commande permettant ainsi de simuler l'action de pompage et de réglage du niveau d'eau.

Le chapitre 04 comprend l'élaboration des spécifications de la station de pompage et leur implémentation dans le logiciel de programmation et la création d'interfaces de supervision et de contrôle. Enfin, nous avons réalisé des simulations et des tests logiciels directement sur l'objet de notre étude.

Nous clôturons notre projet de fin d'étude par une conclusion générale sur l'ensemble de notre étude en indiquant si nos objectifs ont été atteints.

Chapitre 1

Généralités sur les API, IHM et Variateurs de vitesse

1.1 Introduction

De nos jours, la technologie connaît un développement très majeur, et particulièrement dans le domaine de l'automatisation industrielle. L'un des principales technologies amenant ce domaine a son paroxysme est le développement des Automates Programmables. L'industrie d'aujourd'hui est gouvernée par le fabricant Siemens, un leader international qui détient les dernières technologies que l'humanité ait pu produire. Dans le programme de ce chapitre, nous allons présenter l'entreprise Siemens, nous parlerons sur son histoire et son impact sur le monde industriel. Ensuite nous dirigeront notre travail sur les automates programmables et les interfaces homme machine. Nous évoquerons aussi une partie sur les variateurs de vitesse en les définissant en détail et nous citerons leurs avantages et leurs exploitations dans le domaine industriel.

1.2 Siemens

1.2.1 Siemens

Siemens AG est une société allemande conglomérale mondiale fondée en octobre 1847 par **Werner Von Siemens**. La société a son siège à Munich, en Allemagne. Elle est le leader mondial des solutions intégrées dans le domaine de l'automatisation, des entraînements et des solutions logicielles appliquées à l'industrie.[22]

Siemens est présent dans plus de 190 pays et est implanté dans le monde entier grâce à un réseau de près de 290 usines.

1.2.2 Siemens et l'automatisation

L'industrie se trouve à l'aube de la 4^{ème} révolution industrielle. La numérisation de la production fait maintenant suite à l'automatisation. L'objectif principal de l'industrie 4.0 est d'augmenter la productivité, l'efficacité, la vitesse et la qualité. Les entreprises abordent ainsi le futur de l'industrie avec un avantage compétitif.

Siemens est l'unique fabricant au monde à proposer une gamme complète dans le domaine de l'automatisation et numérisation de la production. pour toutes les exigences et dans tous les secteurs : des produits, des systèmes et des solutions de très grande qualité, parfaitement cohérents.

1.2.3 Simatic

La technologie d'automatisation est l'épine dorsale de la production industrielle. Cela rend les usines plus productives, plus efficaces, plus rapides et plus fiables.

SIMATIC a jeté les bases en 1958 de la position de Siemens sur le marché mondial de la technologie d'automatisation.

Les contrôleurs SIMATIC offrent plus qu'un matériel puissant pour contrôler tous les types de machines et d'installations. Les capacités des contrôleurs SIMATIC sont constamment étendues avec de nouvelles fonctions. En outre, ils constituent un élément central de l'intégration verticale OT / IT avec les mécanismes informatiques. Cela fait des contrôleurs

SIMATIC la base idéale pour des solutions innovantes et un élément essentiel de l'approche d'automatisation totalement intégrée 'TIA'. [28]

1.3 Les automates programmables industriels

1.3.1 Définition

Un automate programmable industriel, API (en anglais Programmable Logic Controller, PLC) est un dispositif électronique programmable qui a pour but d'automatiser des processus tels que la commande des machines et des chaînes de production. On nomme les automaticiens les programmeurs de ces Automates Programmables Industriels.

Trois caractéristiques fondamentales distinguent totalement l'Automate Programmable Industriel (API) :

- Il peut être directement connecté aux capteurs et pré-actionneurs grâce à ses entrées /sorties industrielles.
- Il est conçu pour fonctionner dans des ambiances industrielles sévères (température, vibrations, micro-coupures de la tension d'alimentation, parasites, etc.).
- Sa programmation à partir de langages spécialement développés pour le traitement de fonctions d'automatisme fait en sorte que sa mise en oeuvre et son exploitation ne nécessitent aucune connaissance en informatique.

1.3.2 Historique

Les API sont apparues aux États-Unis vers 1969 avant même que n'existent les microprocesseurs où ils répondaient aux désirs des industries de l'automobile de développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des technologies et des modèles fabriqués. L'A.P.I. s'est ainsi substitué aux armoires à relais en raison de sa souplesse, mais aussi parce que dans les automatismes de commande complexes, les coûts de câblage et de mise au point devenaient trop élevés.[17]

De 1970 à 1974, la technologie des microprocesseurs ajouta une plus grande intelligence et flexibilité à l'automate programmable. Les capacités d'interface avec l'utilisateur s'améliorent. L'automate programmable devient capable d'exécuter les opérations arithmétiques et les opérations logiques ; il manipule les données et les adresses ; il effectue la communication avec d'autres automates ou ordinateurs, donnant ainsi une nouvelle dimension aux applications de l'automate programmable.

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

1.3.3 Architecture des API

L'automate programmable reçoit des données par ses entrées pouvant être des capteurs, commutateurs, boutons poussoirs etc. . . et puis commande les pré actionneurs suivant le programme inscrit dans sa mémoire. Ce cycle de traitement est toujours le même, quel que soit le programme, néanmoins le temps d'un cycle d'un API varie selon la taille du programme et la puissance de l'automate.

Un API se compose donc de cinq grandes parties :

- Le processeur "CPU".
- La mémoire.
- Les interfaces Entrées/sorties.
- Le bus.
- L'alimentation.

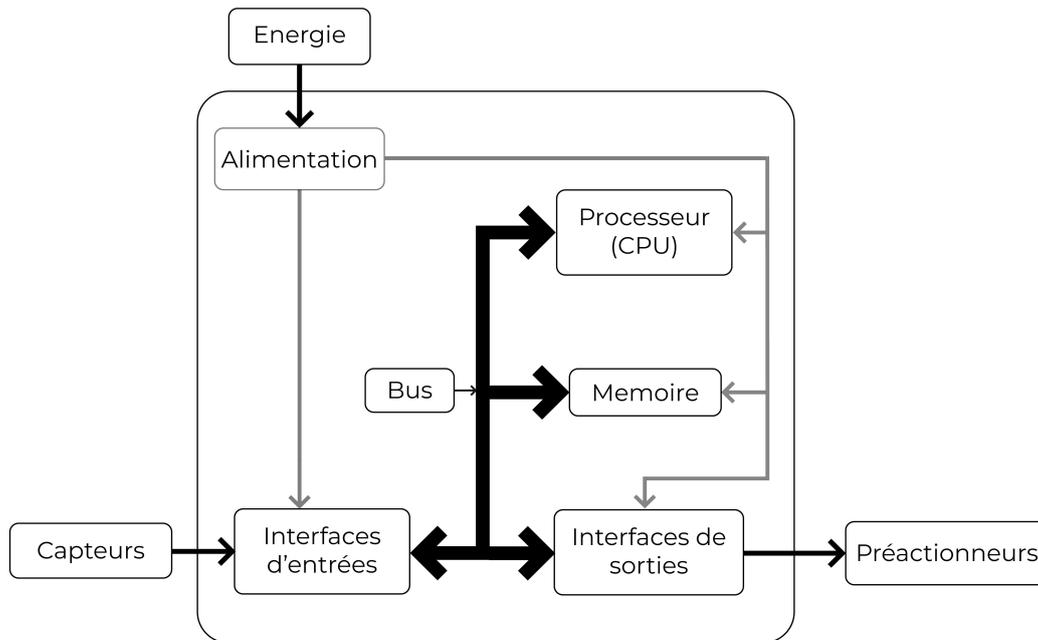


FIGURE 1.1 – Architecture d'un API

Le processeur "CPU" :

Le processeur, ou unité centrale (UC), a pour rôle principal le traitement des instructions qui constituent le programme de fonctionnement de l'application (les fonctions logiques ET, OU, les fonctions de temporisation, de comptage, de calcul PID, etc..). à gérer les entrées et sorties, à surveiller et diagnostiquer l'automate (par des tests lancés régulièrement), à mettre en place un dialogue avec le terminal de programmation .

La mémoire :

Elle permet le stockage des instructions constituant le programme de fonctionnement et les états des entrées/sorties ainsi que diverses informations. Il peut s'agir de mémoire vive (modifiable à volonté mais perdue en cas de coupure de tension) ou de mémoire morte.

Les interfaces entrées/sorties :

Les interfaces entrées/sorties assurent l'intégration directe de l'automate dans son environnement industriel en réalisant la liaison entre le processeur et le processus. Elles

permettent au processeur de recevoir et d'envoyer des informations. Ces dispositifs d'entrée et sortie peuvent produire des signaux discrets, numériques (Sorties de type TOR « tout ou rien ») ou analogiques.

Le bus :

C'est un ensemble de conducteurs qui réalisent la liaison entre les différents éléments de l'automate. Le bus est organisé en plusieurs sous ensembles destinés chacun à véhiculer un type bien défini d'informations :

1. Bus de données.
2. Bus d'adresses.
3. Bus de contrôle pour les signaux de service tels que tops de synchronisation, sens des échanges, contrôle de validité des échanges, etc..
4. Bus de distribution des tensions issues du bloc d'alimentation.

L'alimentation :

Tous les automates actuels sont équipés d'une alimentation à partir d'un réseau 220/240V en courant alternatif ou d'une source 24V en courant continu. Les entrées sont en 24V DC et une mise à la terre doit également être prévue.

1.3.4 Langages de programmation des API

La norme CEI 61131 :

CEI 61131 est une norme de la Commission électrotechnique internationale (CEI) destinée à réglementer plusieurs aspects de la programmation, équipements et communication des automates programmables industriels. Cette norme était nommée CEI 1131 auparavant.

La CEI 61131-3 est la troisième partie de la famille de la norme CEI 61131. Elle se compose des parties suivantes :

- Partie 1 : Vue d'ensemble Générale.
- Partie 2 : Matériel.
- Partie 3 : Langages de Programmation.
- Partie 4 : Directives d'Utilisation.
- Partie 5 : Communication.
- Partie 6 : Sécurité fonctionnelle.
- Partie 7 : Fuzzy Logic.
- Partie 8 : Directives d'application.

La CEI 61131-3, est une norme industrielle publiée la première fois en 1993. Cette norme définit cinq langages de programmation utilisés pour les automates programmables, deux langages textuels (Liste d'instructions et texte structuré) et deux langages graphiques (Diagramme à contacts et diagramme de bloc fonctionnel).

Le dernier représente un ensemble d'éléments graphiques et textuels appelé "diagramme fonctionnel séquentiel" pour structurer l'organisation interne des programmes pour automate programmable et des blocs fonctionnels.

Avantages de la norme CEI 61131 :

- Formation des programmeurs d'applications.
- La création de programmes de plus en plus grands.
- La mise en œuvre de systèmes de programmation de plus en plus complexes.

Les systèmes de programmation API suivent progressivement la tendance du marché des logiciels de masse du monde PC. Ici aussi, la pression des coûts peut être réduite avant tout par la normalisation et la synergie. Étant donné que la norme rapproche des systèmes auparavant spécifiques à chaque fabricant, les fabricants et les clients ont tout à gagner de la norme CEI 61131-3.

Langages de programmation :

Selon la norme CEI 61131-3 on a défini donc cinq langages de programmation pour les automates programmables qui sont : [13]

1. **LD (Ladder Diagram)** : Ou schémas à relais, il se base sur une approche visuelle évoquant des schémas électriques. Ce langage graphique est essentiellement dédié à la programmation d'équations booléennes.

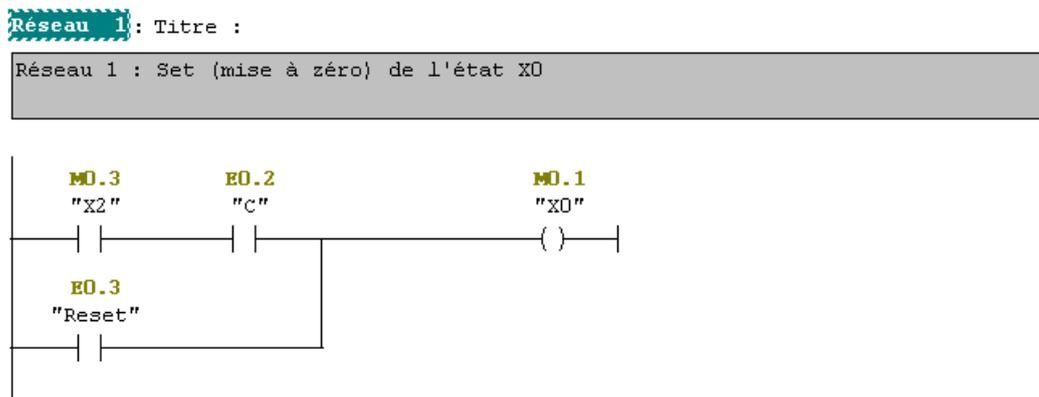


FIGURE 1.2 – Exemple d'un programme en Ladder

2. **IL (Instruction List)** : Ou listes des instructions, ce langage textuel de bas niveau est un langage à une instruction par ligne, il est très proche du langage informatique dit assembleur.

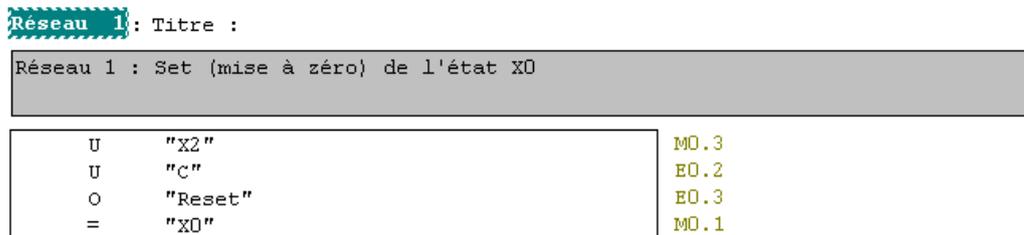


FIGURE 1.3 – Exemple d'un programme en IL

3. **FBD (Function Block Diagram)** : Ce langage permet de programmer graphiquement à l'aide de blocs, représentant des variables, des opérateurs ou des fonctions. Il permet la construction d'équations complexes.

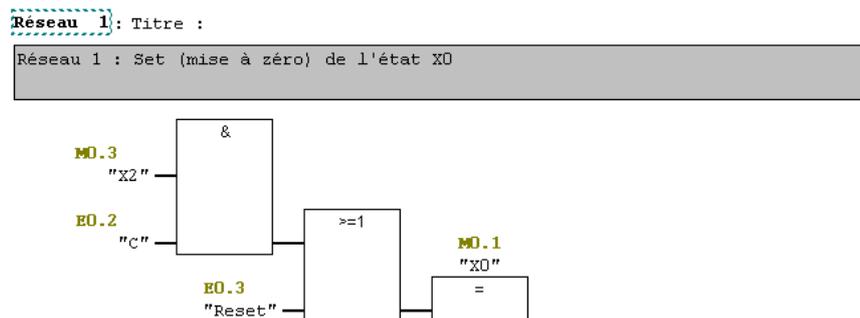


FIGURE 1.4 – Exemple d'un programme en FBD

4. **ST (Structured Text)** : Le texte structuré est un langage textuel de haut niveau qui est utilisé pour décrire des procédures complexes.

```

VAR
  x : BOOL;
END_VAR
x := TRUE;
REPEAT
  x := FALSE;
UNTIL x := FALSE;
END_REPEAT;
END_PROGRAM;

```

FIGURE 1.5 – Exemple d'un programme en ST [2]

5. **SFC (Sequential Function Charts)** : Appelé en français les graphes de fonction séquentielle, est un langage issu du langage GRAFCET il permet la programmation aisée de tous les procédés séquentiels.

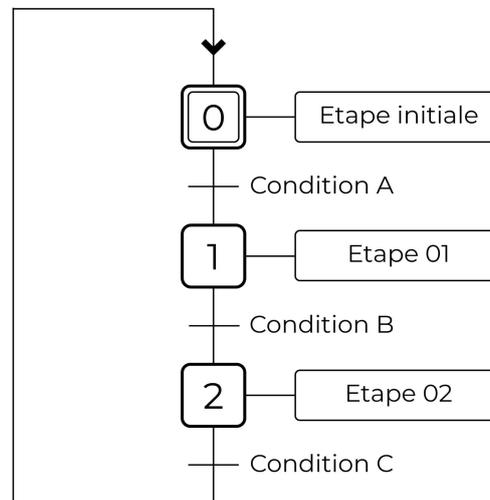


FIGURE 1.6 – Exemple d'un GRAFCET

1.3.5 Choix d'un API

Plusieurs critères sont à prendre en compte au moment de choisir un automate programmable industriel tels que :

- **La marque** : On est plus familier à une certaine marque d'automate, on a donc tendance à choisir un automate car on le maîtrise déjà.
- **Les temps de cycle** : Certains automates ont des temps de cycle plus rapides par rapport à d'autres. Il est donc nécessaire de prêter attention au temps de cycle d'un API si l'on souhaite une bonne réactivité de sa part.
- **L'intuitivité de l'environnement de développement** : Certains automates ont des logiciels de programmation plus aboutis comparés à d'autres. Cela permet un gain de temps énorme lors des développements.
- **La programmation** : Si des programmes spécifiques ont été appliqués à un API bien particulier, il convient de se tourner vers ce type d'automate à l'avenir. Les futures programmations d'automates n'en seront que facilitées.
- **Le prix** : Même si l'on s'accorde à dire qu'un automate industriel constitue un bon investissement à long terme, il convient de garder en tête un prix juste et raisonnable pour le type d'API recherché. S'il s'agit d'un automate pour un petit projet, peut-être il est plus raisonnable d'envisager un micro ou mini automate .
- **La communication** : Il est utile de se renseigner sur les interfaces de communication et les bus de terrain natifs de l'API. Ce qui évite accessoirement d'avoir à racheter d'autres modules par la suite.
- **Les caractéristiques techniques de l'automate** : Il sera nécessaire de s'interroger sur les caractéristiques attendues de l'automate :

- Nombres d'entrées/sorties intégrés.
- Temps de traitement (scrutation).
- Les fonctions spéciales offertes par le processeur.
- Capacité de la mémoire.
- Nombre de compteurs.
- Nombre de temporisateurs.

En résumé, le choix définitif dépendra du cahier de charges et des spécifications techniques du projet. La plupart des fabricants des automates proposent plusieurs gammes d'automates ce qui permet de couvrir une diversité de projets. Certains automates se démarquent par leurs performances et caractéristiques.

1.3.6 Les Avantages des automates

- **Puissance et performance** : Les automates sont un outil très performant, un seul modèle peut être utilisé pour différentes opérations selon les besoins.
- **Rapidité** : Dans les systèmes à relais câblés, le temps d'installation est plus long que celui des panneaux de contrôle basés sur les API.
- **Autonomie et facilité de maintenance** : Le dépannage est plus facile et le nombre de pièces de rechange est réduit et facilement gérable grâce à la conception modulaire. Même la maintenance sur site est commode car elle est similaire à d'autres systèmes à relais ou analogiques.
- **Simplification du câblage** : Les connexions se réduisent au raccordement des capteurs aux entrées et des préactionneurs aux sorties. L'accès aux différents organes de l'automatisme, lots des modifications et des réglages, se trouve ainsi facile.
- **Facilité de modification du programme par rapport à la logique câblée** : Les automates sont fournis avec des langages de programmation simples comme la programmation de type Ladder ou Booléenne.
- **Fiabilité et énorme possibilité d'exploitation** : Les outils de programmation des automates contiennent un grand nombre de contacts internes qui peuvent être utilisés pour tout changement induit dans différentes applications.
- **Robustesse** : Les éléments qui composent les automates sont particulièrement robustes (absence de mécanique tournante pour le refroidissement et le stockage des données, matériaux renforcés) leur permettant de fonctionner dans des environnements particulièrement hostiles (poussière environnante, vibrations des supports, variations de température...).

1.3.7 Les inconvénients des automates

Cependant il existe des inconvénients de l'automate qu'on peut négliger comme : [3]

- **La Cherté de l'appareil et son système** : Ils sont plus chers que des solutions informatiques classiques à base de microcontrôleurs par exemple.
- **Le plantage** : Blocage du programme ou du système qui paralyse par conséquence le fonctionnement de l'API.

1.4 IHM (Interface Homme-Machine)

1.4.1 Définition générale

Interface homme-machine (IHM), désigne un ensemble de moyens et outils informatiques mis en place pour favoriser la communication entre un être humain et une machine. Son rôle principal est de permettre aux utilisateurs de visualiser des données quant aux opérations et de contrôler les machines et faciliter l'interaction entre l'homme et la machine.

Les IHM peuvent prendre différentes formes dans un environnement industriel. Il peut s'agir d'écrans directement intégrés aux machines, d'un tableau de bord attaché à un autre équipement ou d'une tablette tactiles. [9]

1.4.2 L'utilisation des IHM

Les technologies IHM sont utilisées pour un large éventail d'industries ,afin d'interagir avec les machines et d'optimiser les processus industriels. Les interfaces homme-machine sont notamment utilisées dans les industries de l'énergie, de l'agroalimentaire, de la fabrication des produits pharmaceutiques, du pétrole et du gaz, de l'énergie, du recyclage, ou encore du transport.

Les personnes chargées d'utiliser les IHM sont principalement, les opérateurs, les intégrateurs système , et les ingénieurs, surtout les ingénieurs de systèmes de contrôle des processus. Ils utilisent ces interfaces comme des ressources essentielles pour surveiller les processus, diagnostiquer les problèmes, et visualiser les données.

La sophistication de l'IHM varie selon la complexité de la machine ou du système pour lequel elle est utilisée. Elle peut être utilisée pour une seule fonction, par exemple la surveillance d'une machine, ou à des fins multiples, y compris la surveillance des opérations de l'usine et le contrôle des équipements.

Quand le système de contrôle et d'acquisition de données (SCADA, supervisory control and data acquisition; en français : Système de Contrôle et d'Acquisition de Données) communique avec les CPU et les capteurs d'entrée/sortie pour obtenir des informations sur le fonctionnement des équipements, ces informations sont affichées sur une IHM. Elle peut afficher ces informations sous forme de graphiques, de tableaux ou d'autres représentations visuelles qui facilitent la lecture et la compréhension des données. Grâce à l'IHM, toutes les informations de performance des équipements d'une installation peuvent être visionnées à un seul endroit, ce qui améliore la visibilité quant aux opérations des usines. Les opérateurs peuvent également voir et gérer les alarmes à l'aide d'une IHM, cela leur permet d'être sûrs de pouvoir agir rapidement. La possibilité d'effectuer les changements requis sur l'écran rend ce processus plus rapide et plus simple. [9]

1.4.3 La différence entre IHM et SCADA

Les IHM sont proches des systèmes SCADA et sont souvent utilisés simultanément au sein d'un système de contrôle industriel. Cependant, ils présentent plusieurs différences. [9]

- SCADA fait référence à la surveillance et au contrôle de supervision dans les organisations et IHM est un sous-ensemble de SCADA.
- IHM est l'interface homme-machine, elle se connecte à tous les processus, puis présente ces données à un opérateur humain.
- Le bon fonctionnement de l'IHM est essentiel au bon fonctionnement du SCADA.

1.4.4 Les avantages des IHM

Les Interfaces humain-machine offrent un certain nombre d'avantages aux organisations industrielles actuelles, y compris :

- **Amélioration de la visibilité** : Une IHM haute performance offre une meilleure visibilité des opérations, à tout moment. Cela permet de visionner la performance des équipements ou de l'installation.
- **Augmentation de l'efficacité** : La visualisation des données, surtout lorsqu'elle est combinée avec les technologies d'analyse des données, peut aider à identifier les domaines dans lesquels on peut améliorer l'efficacité des procédures automatiques.
- **Diminution des temps d'arrêt** : Grâce aux alertes sur le tableau de bord central, on peut réagir plus rapidement aux problèmes, ce qui réduit les temps d'arrêt.
- **Amélioration de l'ergonomie** : Les interfaces homme-machine facilitent la visualisation et la compréhension des données et des équipements de contrôle pour les utilisateurs. Elles présentent les données à l'aide de graphiques, de tableaux et d'autres visualisations, ce qui permet aux utilisateurs de les interpréter rapidement.
- **Unification du système** : Le contrôle de tous les équipements à l'aide d'une seule plate-forme permet aux opérateurs d'apprendre plus facilement comment contrôler les équipements. [9]

1.4.5 SIMATIC HMI

Les Panneaux SIMATIC HMI présentent un design innovant et une haute performance, ils peuvent être configurés via SIMATIC WinCC dans TIA Portal. Les utilisateurs peuvent bénéficier ainsi d'une efficacité technique très élevée.



FIGURE 1.7 – Écran HMI tactile, KTP 400 [21]

Les différentes familles de panneaux de SIMATIC HMI avec une fonctionnalité intégrée sur toutes les tailles d'écran disponibles :

- **SIMATIC HMI Comfort :**
Premier choix pour les applications exigeantes.
- **SIMATIC HMI Basic :**
Fonctions de base pour des applications IHM simples.
- **SIMATIC HMI Mobile :**
Mobilité maximale pour le fonctionnement et la surveillance.
- **SIMATIC HMI Key :**
Panneaux de commande innovants pour un fonctionnement clé en main.

1.5 Variateurs de vitesse

1.5.1 Définition

Un variateur de vitesse est un équipement permettant de faire varier la vitesse d'un moteur, autrement dit c'est un convertisseur d'énergie, il permet de moduler l'énergie électrique fournie au moteur. Les variateurs de vitesse assurent une accélération et une décélération progressives, ils permettent une adaptation précise de la vitesse aux conditions d'exploitation. [20]

1.5.2 Historique

Les démarreurs rhéostatiques, les variateurs mécaniques et les groupes tournants (Ward Leonard en particulier) ont été les premières solutions pour démarrer les moteurs électriques et contrôler leur vitesse, puis les démarreurs et variateurs électroniques se sont imposés dans l'industrie comme la solution moderne, économique, fiable et sans entretien. [20]

1.5.3 Principe de fonctionnement

Le variateur de vitesse est composé essentiellement :

1. **Un redresseur** : Son rôle est de transformer la tension alternative en tension continue ou plus exactement en tension pseudo continue. En effet la tension continue obtenue n'est pas parfaite car elle comporte des ondulations positives résiduelles. Le redresseur est formé de diodes semi-conductrices. Ce composant électronique fonctionne comme à un clapet anti-retour n'autorisant le passage du courant que dans un sens. C'est cette fonction qui permet la modification de la nature du courant. Le redresseur peut être de type commandé ou pas.
2. **Un circuit intermédiaire** : Agissant principalement sur le "lissage" de la tension de sortie du redresseur (améliore la composante continue). Le circuit intermédiaire peut aussi servir de dissipateur d'énergie lorsque le moteur devient générateur.
3. **Un onduleur** : Il engendre le signal de puissance à tension et/ou fréquence variables.
4. **Électronique de commande** : C'est le cerveau du variateur de fréquence, un calculateur récolte les données et délivre les messages d'erreurs, il pilote le redresseur et l'onduleur, il protège le moteur et l'ensemble du variateur. Suivant sa technologie, le variateur dispose de multiples fonctions de commande et de surveillance.

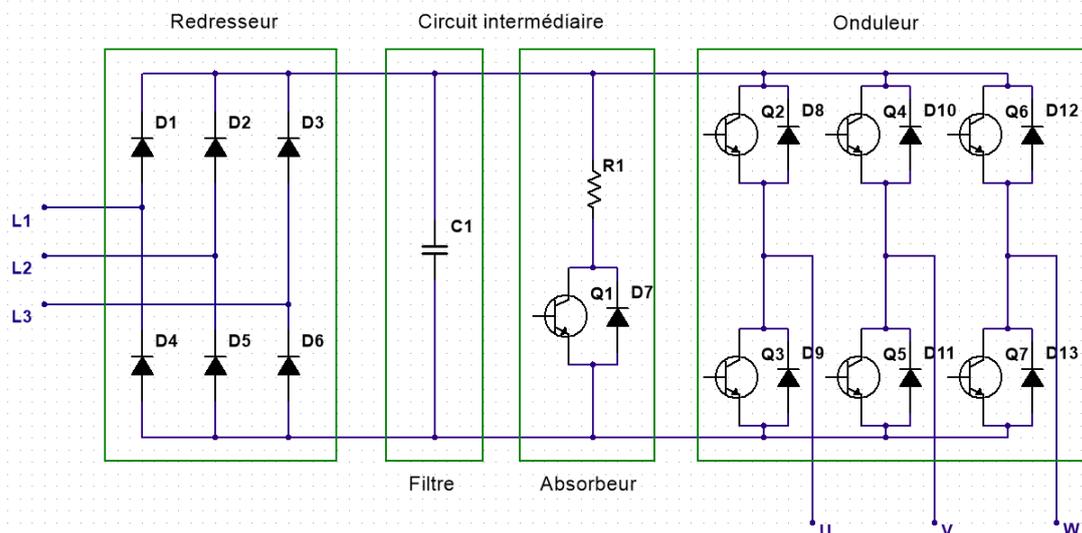


FIGURE 1.8 – Schéma électrique d'un variateur de vitesse "réalisé par Multisim"

Dans le premier cas, les variateurs convertissent la tension alternative du réseau électrique en une tension de sortie également alternative de phase et d'amplitude souhaitées pour le moteur électrique. Les variateurs de vitesse en source de tension sont les plus répandus. Dans le cas où une tension continue est disponible, les variateurs de vitesse sont seulement composés d'un onduleur. Les redresseurs sont généralement des ponts de diodes double alternance triphasés à six pulsations. Dans le cas d'un variateur fonctionnant en source

de tension, le circuit intermédiaire est constitué d'une capacité montée en shunt qui sert à lisser la tension. La tension continue est ainsi constante à l'entrée de l'onduleur. Celui-ci la transforme en une tension sinusoïdale pour alimenter le moteur. Les onduleurs en source de tension ont un facteur de puissance plus élevé et fournissent une tension ayant moins d'harmoniques que ceux en source de courant. Il est également possible d'avoir un contrôleur fonctionnant en convertisseur de phase, avec seulement une phase en entrée et trois phases en sortie.

Dans les dernières années l'évolution des composants semi-conducteurs était énorme, leur tension, courant et fréquence de commutation admissibles ont augmenté considérablement. Ils se sont imposés comme les composants semi-conducteurs les plus utilisés pour les onduleurs associés aux variateurs de vitesse.

1.5.4 Avantages des variateurs de vitesse

Le recours aux variateurs de vitesse offre plusieurs avantages : [20]

- Démarrage progressif des moteurs réduisant les chutes de tension dans le réseau et limitant les courants de démarrage.
- Amélioration du facteur de puissance.
- Précision accrue de la régulation de vitesse.
- Prolongement de la durée de service du matériel entraîné.
- Diminution de la consommation d'électricité.

De nouveaux variateurs de vitesse plus performants peuvent éviter l'interruption des procédés en cas de perturbation du réseau de courte durée.

1.5.5 Protection des variateurs de vitesse contre les creux de tension

- Optimiser les protections (fusibles adéquats pour la protection de l'électronique de puissance).
- Choisir judicieusement les réglages (sous-tension, surtension, surintensité, surcharge, redémarrage automatique, temps d'accélération et de décélération, etc.).
- Prévoir des mesures d'atténuation (transformateurs à tension constante, transformateurs d'isolement, réactance de 3 %, alimentation sans coupure [UPS], conditionneurs et régulateurs de tension) et, dans la mesure du possible, réserver des artères distinctes aux variateurs de vitesse.
- Installer des filtres passifs ou actifs pour réduire l'impact des courants et des tensions harmoniques causés par les variateurs de vitesse et limiter leur propagation dans le reste du réseau électrique.

Les mesures précédentes contribuent à améliorer l'immunité des variateurs de vitesse :

Lors de l'acquisition de nouveaux variateurs de vitesse, il est fortement recommandé de les immuniser suffisamment pour maintenir le bon fonctionnement des équipements en cas de perturbations électriques.

1.5.6 Les fonctions des variateurs de vitesse

- **Accélération contrôlée** : La mise en vitesse du moteur est contrôlée au moyen d'une rampe d'accélération linéaire. Cette rampe est généralement réglable et permet par conséquent de choisir le temps de mise en vitesse approprié à l'application.
- **Variation de vitesse** : C'est un système, rudimentaire, qui possède une commande élaborée à partir des grandeurs électriques du moteur avec amplification de puissance, mais sans boucle de retour, il est donc en boucle ouverte.
- **Régulation de vitesse** : Un régulateur de vitesse est un variateur asservi. Il possède un système de commande avec amplification de puissance et une boucle de retour, il est donc en boucle fermée ».

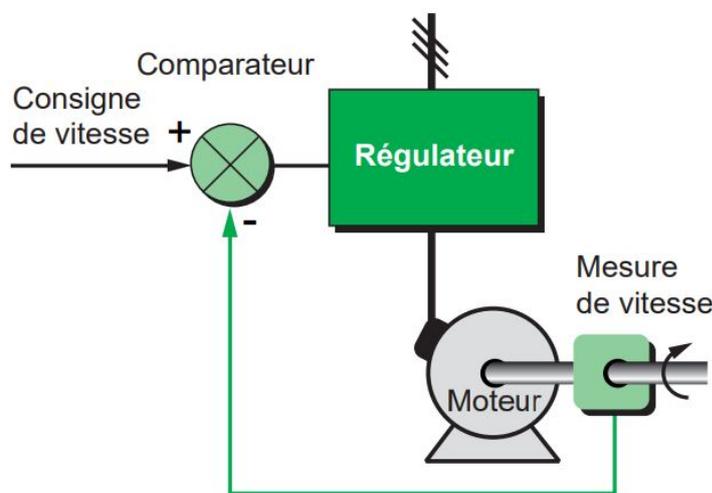


FIGURE 1.9 – Principe de fonctionnement de la régulation de vitesse [7]

- **Décélération contrôlée** : Quand un moteur est mis hors tension, sa décélération est due uniquement au couple résistant de la machine (décélération naturelle). Les variateurs électroniques permettent de contrôler la décélération au moyen d'une rampe linéaire, généralement indépendante de la rampe d'accélération. Cette rampe peut être réglée de manière à obtenir un temps de passage de la vitesse en régime établi à une vitesse intermédiaire ou nulle.
- **Inversion du sens de marche** : La majorité des variateurs actuels permettent cette fonction en standard. L'inversion de l'ordre des phases d'alimentation du moteur est réalisée automatiquement soit par inversion de la consigne à l'entrée, soit par un ordre logique sur une borne, soit par une information transmise par une connexion réseau.
- **Freinage d'arrêt** : Ce freinage consiste à arrêter un moteur sans pour autant contrôler la rampe de ralentissement. Pour les démarreurs et variateurs de vitesse pour moteurs asynchrones, ceci est réalisé de manière économique en injectant du courant continu dans le moteur avec un fonctionnement particulier de l'étage de puissance. Toute l'énergie mécanique est dissipée dans le rotor de la machine et, de ce fait, ce freinage ne peut être qu'intermittent. Sur un variateur pour moteur

à courant continu, cette fonction sera assurée en connectant une résistance aux bornes de l'induit.

- **Protections intégrées :** Les variateurs modernes assurent en général la protection thermique des moteurs et leur propre protection. A partir de la mesure du courant et d'une information sur la vitesse (si la ventilation du moteur dépend de sa vitesse de rotation), un microprocesseur calcule l'élévation de température du moteur et fournit un signal d'alarme ou de déclenchement en cas d'échauffement excessif.

1.5.7 Inconvénients des variateurs de vitesse

Tous les variateurs de vitesse intégrant des dispositifs de commutation (diodes, thyristors, IGBT, etc.), ils forment une charge non linéaire qui engendre des courants harmoniques, sources de distorsion de l'onde (chute ou perturbation de la tension) dans le réseau électrique. Cette dégradation de l'onde peut perturber tant les équipements électriques du système que ceux du réseau électrique si aucune mesure d'immunité n'est prise.

Par ailleurs, des résonances harmoniques peuvent également apparaître entre les variateurs de vitesse et les batteries de condensateurs. Les effets néfastes peuvent se manifester par la défaillance prématurée des équipements électriques (surchauffe des moteurs, des câbles et des transformateurs), par la dégradation de l'isolation des moteurs commandés ou par l'interruption des procédés (fusibles brûlés).

1.6 Conclusion

Dans ce chapitre on a défini en détail les automates programmables et les interfaces homme machine en mentionnant leur constitution et la méthode de programmation ainsi que leurs utilités et avantages sur le domaine de l'automatisation industrielle.

On a clôturé ce chapitre par une définition détaillée rigoureusement des variateurs de vitesse et leur fonctionnement. Nous avons mis l'accent sur leur emploi dans la régulation et la commande des machines en citant des avantages et des inconvénients.

Chapitre 2

Étude du matériel de la station de pompage

2.1 Introduction

À travers le chapitre 2 nous allons citer et expliquer minutieusement tout matériel et équipement utilisés dans la construction du prototype de la station de pompage. Nous évoquerons pareillement la synergie entre l'ensemble des dispositifs ainsi que la manière dont ils se branchent dans l'objectif d'assurer un fonctionnement spécifique de la station.

2.2 Prototype de la station de pompage

2.2.1 Description

Une station de pompage a pour but de réaliser l'extraction d'un fluide tels que l'eau ou le pétrole. L'utilisation de cette dernière est très essentielle car elle permet de faire l'approvisionnement en eau des canaux, le drainage des terres basses ainsi que l'élimination des eaux usées et leur transfert vers les sites de transformation.

La station de pompage est composée essentiellement d'une pompe couplée à un moteur électrique, souvent à courant alternatif asynchrone monophasé ou à 3 phases. Le rôle d'une pompe est l'approvisionnement de l'énergie hydraulique au fluide permettant de le déplacer d'un point A à un point B avec une vitesse déterminée. La pompe est caractérisée par une puissance fournie P et un débit maximal Q .

La station de pompage étudiée contient deux réservoirs à eau, le premier en position inférieure définie : "Réservoir A", contient l'eau qui va être refoulée par la pompe. Le second, en position supérieure, reçoit l'eau provenant du réservoir A est défini : "Réservoir B". Les deux réservoirs sont caractérisés par la capacité de rétention maximale exprimée en volume ainsi qu'une surface rectangulaire.

La pompe et les deux réservoirs sont reliés par des tuyaux cylindriques. La mobilité de l'eau et son déplacement entre les réservoirs sont commandés par deux électrovannes. De plus, il existe une vanne manuelle assurant le retour de l'eau du réservoir B vers le réservoir A.

En dernier, la station possède une panoplie de dispositifs dédiés à l'instrumentation (Débitmètre et Capteur de Niveau) contrôlant les phénomènes physiques de nature variable.

2.2.2 Schéma électrique de la station de pompage

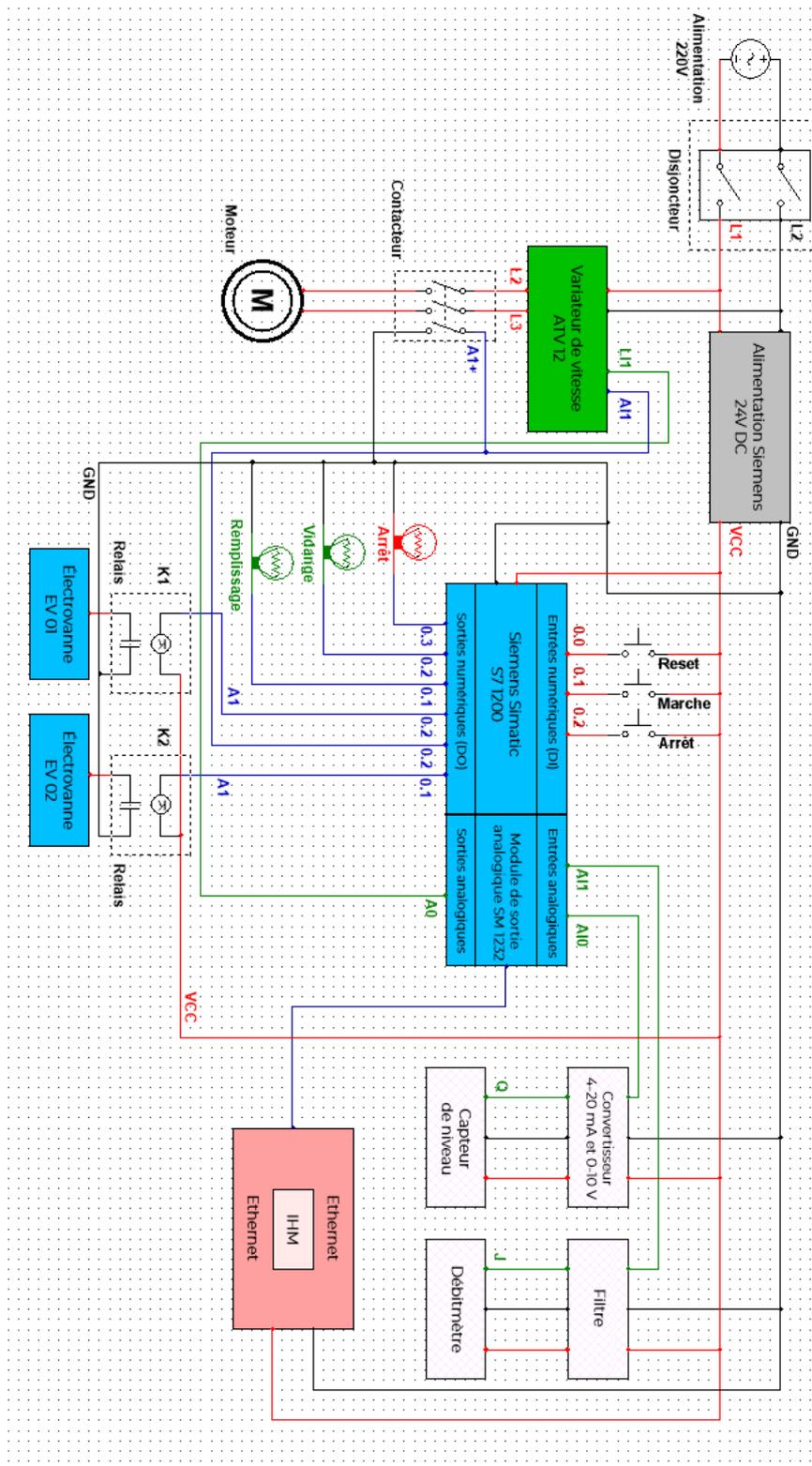


FIGURE 2.1 – Schéma électrique de la station

2.2.3 Assemblage et câblage de la station de pompage



FIGURE 2.2 – Câblage de la station



FIGURE 2.3 – Le Prototype de la station

2.3 Simatic S7-1200

2.3.1 Description

L'automate **SIMATIC S7-1200** est un mini-contrôleur modulaire utilisé pour les petites performances mais avec une précision extrême il offre la souplesse et la puissance nécessaires pour commander une large gamme d'appareils, grâce à sa configuration souple et son important jeu d'instructions il représente une solution idéale pour commander des applications très variées.

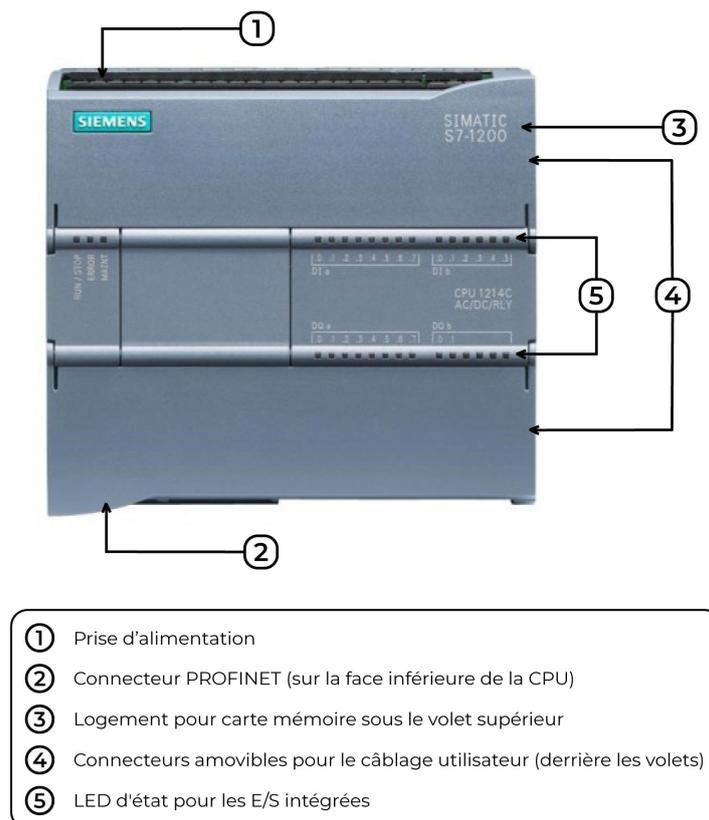


FIGURE 2.4 – Simatic S7-1200 [6]

Ce contrôleur est composé d'une CPU qui est équipée d'un microprocesseur, une alimentation intégrée, des circuits d'entrée et de sortie, un PROFINET intégré, des entrées/sorties rapides de commande de mouvement, ainsi que des entrées analogiques intégrées dans un boîtier compact en vue de créer un contrôleur puissant.

La CPU fournit un port PROFINET permettant de communiquer par le biais d'un réseau PROFINET. Si besoin des modules supplémentaires sont disponibles pour communiquer via les réseaux PROFIBUS, GPRS, RS485 ou RS232. [27]

2.3.2 Les Modules de communication

On peut ajouter à la CPU SIMATIC S7-1200 jusqu'à trois modules de communication maximum.



FIGURE 2.5 – Le Simatic S7-1200 et ses modules de communication [26]

2.3.3 La mémoire

Jusqu'à 50 *Ko* de RAM est présentée avec une part ajustable entre le programme utilisateur et les données utilisateur avec un maximum de 2 *Mo* de mémoire de chargement intégrée et 2 *Ko* de mémoire de données rémanente. Il y a la possibilité d'ajouter une carte mémoire SIMATIC, elle permet de transférer facilement des programmes sur plusieurs CPU. et d'enregistrer des différents fichiers elle est aussi utilisée pour actualiser le microprogramme de l'API. [27]

2.3.4 Les modules d'entrées/sorties

Avec la possibilité d'avoir jusqu'à 8 modules d'entrées/sorties et aussi l'ajout des entrées/sorties TOR et analogiques supplémentaires.

2.3.5 Régulation

Peuvent être utilisées comme sorties MLI pour la régulation de la vitesse d'un moteur, pour le positionnement d'une vanne.

On peut réaliser des applications process simples avec régulation en circuit fermé qui sont faites par des boucles de régulation PID avec fonctionnalité d'auto-ajustement.

2.3.6 Communication industrielle

Interface PROFINET intégrée :

Le SIMATIC S7-1200 est équipé d'une interface PROFINET intégrée qui garantit une communication transparente et optimale avec le système d'ingénierie intégré pour la programmation, avec des pupitres IHM pour la visualisation, avec des contrôleurs supplémentaires pour la communication API-API et avec des appareils d'autres constructeurs pour des possibilités d'intégration avancées.

L'interface de communication SIMATIC S7-1200 se compose d'un connecteur RJ45 insensible au bruit avec fonction de croisement automatique prenant en charge jusqu'à 16 connexions Ethernet et fournissant des débits de transmission de données jusqu'à 10/100 *Mbit/s*. [27]

Mise en réseau simple :

Pour but de minimiser le câblage et d'autoriser une flexibilité maximale du réseau, le nouveau module de commutation compact CSM 1277 peut être utilisé pour configurer facilement un réseau homogène ou mixte composé de topologies linéaires, arborescentes ou en étoile. Le CSM 1277 est un commutateur non géré à 4 ports qui permet de relier le SIMATIC S7-1200 avec jusqu'à 3 appareils supplémentaires.

Communication avec d'autres automates et appareils IHM :

Afin de permettre la communication avec d'autres automates SIMATIC et terminaux IHM, le SIMATIC S7-1200 peut se raccorder à plusieurs appareils communiquant entre eux sur la base du protocole de communication Profinet (TCP/IP) largement éprouvé.

Communication avec des appareils d'autres constructeurs :

L'interface intégrée du SIMATIC S7-1200 permet d'intégrer parfaitement des appareils d'autres constructeurs. Avec les protocoles Ethernet ouverts TCP/IP native et ISO-on-TCP, il est possible de raccorder plusieurs appareils d'autres constructeurs et de communiquer avec ces appareils. Cette possibilité de communication est configurée avec des blocs standard T-Send/T-Receive fournis par le système d'ingénierie intégré SIMATIC STEP 7 Basic, cette capacité offre un niveau de flexibilité encore plus élevé dans la conception des solutions d'automatisation. [27]

2.4 Modules ajoutés à la CPU

2.4.1 L'alimentation (SITOP PSU100S)

SITOP PSU100S fabriquée par Siemens est l'alimentation compacte et performante pour les machines et installations automatisées avec une tension de sortie continue stabilisée de valeur nominale de 24V et un courant de 10 A.

Elle présente un comportement exceptionnel en cas de surcharge. En effet, elle délivre 1,5 fois le courant nominal pendant 5 s/min, permettant d'activer facilement des appareils à fort courant d'appel.

L'option de montage en parallèle est possible pour des meilleures performances.



FIGURE 2.6 – Alimentation SITOP PSU100S (Réf : 6EP1334-2BA20) [8]

2.4.2 Le module analogique

Le module de sorties analogiques SM 1232 - 6ES7232-4HD32-0XB0 de la commande Siemens SIMATIC S7-1200 peut être étendu à 4 sorties analogiques courant ou tensions et il est alimenté par une tension continue de 24 V et a une puissance dissipée de valeur de 1.5 W.

Pour les sorties analogiques les étendues de sortie des courant et tension sont comme suit :

- **Tension** : -10 V à $+10\text{ V}$.
- **Courant** : 0 mA à 20 mA .

Sa Résolution avec domaine de dépassement (bits avec signe) est comme suit :

- **Tension** : 14 bit.
- **Courant** : 13 bit.



FIGURE 2.7 – Module de sorties analogiques SM 1232 (Réf : 6ES7232-4HD32-0XB0) [1]

2.5 Écran IHM

2.5.1 SIMATIC HMI Comfort

Tous les panneaux SIMATIC HMI Comfort offrent la même fonctionnalité haut de gamme dotés d'un affichage d'une haute résolution et d'écrans larges de 4" à 22", en option avec une opération tactile ou par bouton, ils peuvent être parfaitement ajustés à n'importe quelle application. Par ailleurs, ils offrent de nombreux atouts :

- **Écrans flexibles haute résolution** : Avec jusqu'à 40 % d'espace de visualisation en plus, le grand écran offre des options d'affichage étendues pour écrans de commande complexes et une différenciation claire entre la surveillance de l'application et l'exploitation. Une haute résolution avec 16 millions de couleurs facilite les détails de visualisation des processus et offre une lisibilité optimale.
- **Gestion efficace de l'énergie** : Le protocole standardisé PROFINET prend en charge la déconnexion centralisée coordonnée des charges énergétiques redondantes et la détection des valeurs de l'énergie mesurée. Cela facilite la déconnexion des écrans des panneaux SIMATIC COMFORT même pendant les courtes pauses, pour une moindre consommation d'énergie.
- **100% de sécurité des données en cas de panne électrique** : La protection des Comfort Panneaux contre les coupures de tension garantit une conservation des données rentable sans alimentation électrique sans coupure supplémentaire. Il en va de même pour les recettes et les archives au format RDB lorsqu'elles sont enregistrées sur une carte mémoire SIMATIC HMI.
- **Adaptation aux environnements durs** : Les panneaux SIMATIC HMI Comfort sont livrés avec de nombreuses homologations pour les applications internationales et pour une utilisation dans des secteurs soumis à des exigences accrues. Les approbations maritimes sont partiellement disponible pour tous les panneaux Comfort.

2.5.2 SIMATIC HMI 6AV2124-0JC01-0AX0

SIMATIC HMI TP900 Comfort, Comfort Panel, commande tactile, écran large TFT 9", 16 millions de couleurs, interface PROFINET, interface MPI/PROFIBUS DP, 12 Mo de mémoire de configuration, Windows CE 6.0, configurable à partir de WinCC Comfort V11.



FIGURE 2.8 – SIMATIC HMI 6AV2124-0JC01-0AX0 [5]

2.6 Variateur de vitesse ATV12H037M2

2.6.1 Description

Fabriqué par Schneider Electric, ce variateur de vitesse/fréquence est conçu pour des machines et des appareils industriels et commerciaux simples comme des pompes, extracteurs, et tapis roulants. Son encombrement réduit permet un montage dans des châssis d'appareils simples non industriels. Il peut être configuré en amont même dans son emballage et sans avoir à recourir à une alimentation, ce qui permet de réduire au minimum le temps d'installation. La compatibilité avec le logiciel SoMove permet d'enregistrer et de transférer des configurations sur un disque de secours, ce qui permet de réduire les temps d'arrêts de maintenance [23] .



FIGURE 2.9 – Altivar ATV12H037M2 [12]

2.6.2 Détails du produit

- Régulation de vitesse pour petites machines avec moteurs asynchrones triphasés.
- **Entrée** : monophasée, 200...240 V, 50/60 Hz.
- **Sortie** : triphasée, 200...240 V, 0.5...400 Hz.
- **Indice IP** : IP 20.
- Panneau frontal recouvert, avec bouton de navigation, également utilisé pour l'entrée de paramètres.
- Démarrage rapide avec "Plug & Drive", préréglage pour les applications courantes.
- Logiciel SoMove pour une configuration aisée à l'aide d'un PC (téléchargeable gratuitement sur la page d'accueil du fabricant).
- 4 entrées logiques programmables permettent d'enregistrer les sorties dans l'API.

2.6.3 Applications

Applications pour machines industrielles simples :

- Manutention (petits convoyeurs, ...).
- Emballage et conditionnement (petites étiqueteuses, petites ensacheuses, ...).
- Applications de pompage (pompes d'aspiration, pompes centrifuges, pompes de circulation, ...).
- Machines équipées de ventilateur (extraction d'air ou de fumée, machines de fabrication de film plastique, fours, chaudières, ...).

Applications pour machines grand public :

- Manutention (barrières d'accès, panneaux publicitaires rotatifs, ...).
- Machines pour le domaine de la santé (lits médicaux, appareils d'hydromassage, tapis de course, ...).
- Machines pour l'industrie agro-alimentaire (moulins, pétrins, mixeurs, ...).

Autres types d'applications :

- Agro-alimentaire (élevage en batterie, serres de jardinage, ...).
- Applications diverses (machines mobiles et petits appareils équipés d'une prise secteur).
- Applications utilisant traditionnellement d'autres solutions :
 - Moteur à courant continu, variateur mécanique, ...
 - Moteur monophasé pour applications pompe et ventilation utilisant une régulation mécanique ; la solution "Altivar 12 + moteur triphasé" adapte la puissance au besoin de l'application entraînant, ainsi, une réduction de la consommation d'énergie.

Fonctions dédiées aux applications de pompage :

- Sommeil/Réveil.
- Régulation PID.
- La protection :
 - Protection contre les surcharges et les sous-charges en cours d'opération.
 - Protection mécanique des machines avec contrôle des opérations.
 - Protection de l'installation en détectant les surcharges et les sous charges.

2.7 Unité de pompage (Moteur électrique & pompe)

L'unité de pompage est un élément indispensable dans une station de pompage. Elle se compose principalement d'un moteur électrique couplé à une pompe. Le rôle d'une unité de pompage est de déplacer un fluide d'un point A à un point B en lui fournissant une énergie hydraulique suffisante.

2.7.1 Pompe centrifuge

La pompe centrifuge est un dispositif rotatif qui pompe un liquide en l'aspirant d'une manière axiale, puis à travers l'effet rotatif de la roue qui se trouve à l'intérieure (appelée Impulseur), le liquide est accéléré radialement, et enfin refoulé tangentielllement. Cette pompe est la plus utilisée dans le milieu industriel grâce à sa simplicité et ses performances, elle permet un déplacement à des vitesses élevées avec un minimum d'entretien.

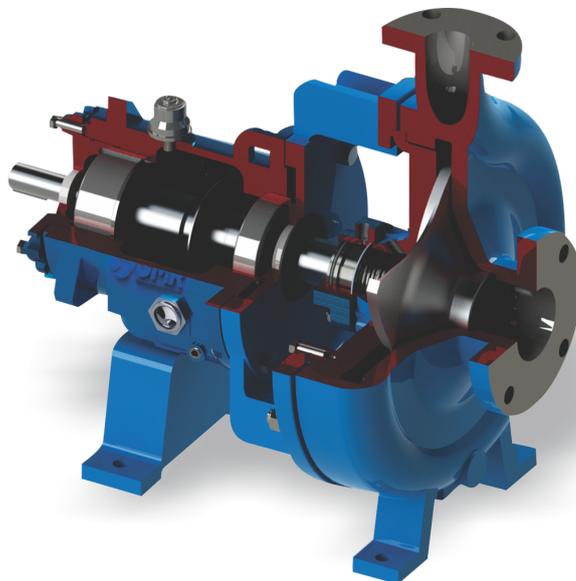


FIGURE 2.10 – Pompe centrifuge [32]

2.7.2 Moteur asynchrone

Le moteur asynchrone est une machine électrique à courant alternatif. Il se compose d'un élément rotatif appelé Rotor et un élément statique appelé Stator. Le moteur

asynchrone est un convertisseur électromécanique basé sur l'électromagnétisme, il permet de convertir l'énergie électrique en une énergie utile (énergie mécanique) à travers la rotation du rotor sous l'effet du champ magnétique. Les moteurs asynchrones sont les plus employés dans les industries en raison de leur simplicité, la facilité de leur maintenance et leur coût abordable.



FIGURE 2.11 – Moteur asynchrone [19]

2.7.3 Unité de pompage du prototype de la station

Le prototype de la station de pompage de l'ENP possède une unité de pompage de la marque LEO, modèle APm37. Cette unité intègre à la fois le moteur asynchrone monophasé ainsi que la pompe centrifuge dans le même dispositif.



FIGURE 2.12 – Unité de pompage LEO APm37 [16]

- **Caractéristiques techniques :**

Température maximale de liquide	+40 C°
Poids	5.5 Kg
Débit maximal	5 - 35 L/min
Aspiration maximale	+8 m
Puissance du moteur	370 W
Indice de protection	IPX4

TABLE 2.1 – Caractéristiques techniques de l'unité de pompage APm37

2.8 Instrumentation et appareils de mesure

L'instrumentation est une technique de mise en œuvre des instruments de mesure, d'actionneurs et des différents appareils présents dans un environnement industriel afin de concevoir un système d'acquisition de données et de commande automatique.

Parmi les instruments de la chaîne de mesure qu'on trouve dans un milieu industriel, on cite :

- **Les capteurs :** Ce sont des dispositifs de mesure permettant de convertir la nature de la grandeur physique mesurée en une autre nature par le biais d'un ou de plusieurs transducteurs. Il existe deux types de capteurs, les capteurs passifs ou capteur à base d'impédances qui nécessitent une alimentation extérieure afin d'exploiter la mesure (ex. Thermistance, potentiomètre). Les capteurs actifs qui sont caractérisés par leur capacité de générer un signal électrique lorsqu'ils sont soumis à une grandeur physique sans faire appel à une excitation externe (ex : Capteur de distance et de vitesse).
- **Les conditionneurs :** Il s'agit d'un circuit électronique précédé par un capteur. Son rôle est de faire le traitement et le conditionnement du signal électrique provenant d'un capteur afin de simplifier l'interface de ceci avec le système d'acquisition des données.

2.8.1 Capteurs

La station de pompage nécessite deux capteurs afin d'acquérir les données nécessaires pour réaliser une commande automatique.

1. **Capteur de distance :**

Le capteur de distance permet de mesurer la distance entre lui et la surface du fluide contenu dans un réservoir. Le prototype de la station possède un capteur de distance à ultrasons SICK UM18.



FIGURE 2.13 – Capteur de distance UM18-51111 [24]

- Détails techniques :

Dimensions	18 mm * 18 mm * 73,3 mm
Design	Cylindrique
Poids	55 g
Tension d'alimentation	10 – 30 V DC
Portée/Portée maximale	30 – 250 mm / 350 mm
Résolution	$\geq 0,08$ mm
Précision	$\leq 2\%$
Temps de réponse	32 ms
Objets de mesure	objets physiques

TABLE 2.2 – Détails techniques du capteur UM18-51111

- Interface de sortie :

Sortie analogique	$4 - 20 \text{ mA} \leq 500 \Omega$
Résolution	12 bit sur module d'entrée

TABLE 2.3 – Détails de l'interface de sortie du capteur UM18-51111

- Câblage des fils :

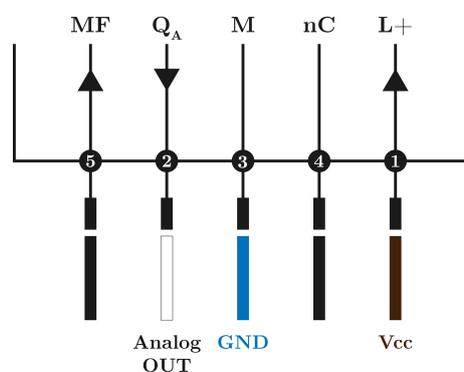


FIGURE 2.14 – Schéma de câblage du capteur SICK UM18-51111

- Fil marron (L+) : Alimentation DC +24 V.
- Fil bleu (M) : Masse.
- Fil blanc (Q_A) : Sortie analogique de type courant 4-20 mA.
- Entrée non connectée/affectée (nC)
- Entrée multifonction (MF)

2. Capteur de débit (Débitmètre) :

Le débitmètre présent à la station permet de mesurer le débit provenant de la pompe. Ce capteur est composé d'un rotor et d'un capteur à effet de Hall et délivre un train d'impulsion à sa sortie. Les impulsions sont en fonction du débit d'eau et la fréquence est proportionnelle à la vitesse du rotor.

Afin de pouvoir exploiter ses données, il est nécessaire de les faire passer par une interface de sortie.



FIGURE 2.15 – Débitmètre YF-S201 [29]

• Détails techniques :

Dimensions	63 mm * 36 mm * 35 mm
Poids	46 g
Tension d'alimentation	5 – 24 V DC
Plage de mesure	1 – 30 L/min
Pression maximale	1.2 MPa
Précision	± 3 %

TABLE 2.4 – Détails techniques du débitmètre YF-S201

• Câblage des fils :

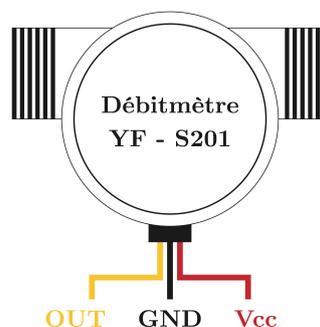


FIGURE 2.16 – Schéma de câblage du débitmètre YF-S201

- Fil rouge : Alimentation DC +24 V.
- Fil noir : Masse.
- Fil jaune : Sortie digitale.

2.9 interfaces d'entrée et de sortie

Les interfaces d'entrée et de sortie ont pour but de contrôler et de mettre en forme les signaux selon leur position dans le circuit électrique par rapport à l'API. Au niveau du prototype il y a des :

2.9.1 Interfaces d'entrée

Au niveau du prototype, il y a 3 interfaces d'entrée.

1. Disjoncteur :

Un disjoncteur est un interrupteur électrique à commande manuelle conçu pour laisser circuler le courant électrique et protéger un circuit électrique contre les dommages causés par un courant excessif provenant d'une surcharge ou d'un court-circuit (contact phase + neutre). Il évite ainsi les échauffement et les risques d'incendie.



FIGURE 2.17 – Disjoncteur bipolaire ENG [18]

2. Contacteur :

C'est un appareil électrique qui permet au courant provenant d'un composant de passer ou de s'interrompre à partir d'une commande lancée à partir de l'API. Dans le prototype, il y a un contacteur SIEMENS modèle 3RH2140-1AP00, ce contacteur est placé entre la sortie du variateur de fréquences et l'entrée de l'unité de pompage. Ce dernier permet de séparer le signal de commande transmis par le variateur et l'alimentation de la pompe. L'ouverture et la fermeture des interrupteurs assurant la séparation est commandé par un relais branché à la sortie digitale de l'API, cela permet de protéger la pompe des anomalies qui peuvent avoir lieu.



FIGURE 2.18 – Contacteur SIEMENS 3RH2140-1AP00 [25]

Nombre de pôles	4 pôles
Type de contact auxiliaire	4 NO
Tension de contact	690 V /AC
Tension de bobine	230 V /AC
Type de terminaison	Vis

TABLE 2.5 – Détails techniques du contacteur 3RH2140-1AP00

3. Conditionneur :

Un conditionneur a la capacité de mettre en forme le signal provenant d'un capteur. Dans notre cas, une carte Arduino a été utilisée dans le prototype afin de transformer le signal généré par le débitmètre (Le train d'impulsions) en un signal MLI (Modulation de la Largeur d'Impulsion). La transformation se fait grâce au micro-contrôleur présent dans la carte, programmé à faire cette tâche. Ainsi, la modulation des largeurs d'impulsion octroie un sens physique au signal transformé, chaque impulsion représente le débit mesuré par le capteur en amont.[10]

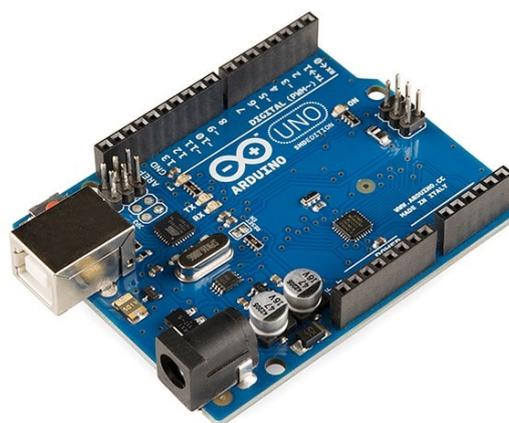


FIGURE 2.19 – Carte Arduino UNO [10]

Un second conditionneur est placé après la carte Arduino, il s'agit d'un filtre passe-bas. Ce dernier sert à filtrer le signal MLI généré par la carte, il converti son amplitude originale $U = 5V$ à une fréquence f égale à $480 Hz$ en un signal d'amplitude

variant de 0 à 5V, ce qui donne au signal MLI sa composante continue qui est exploitable par l'API.

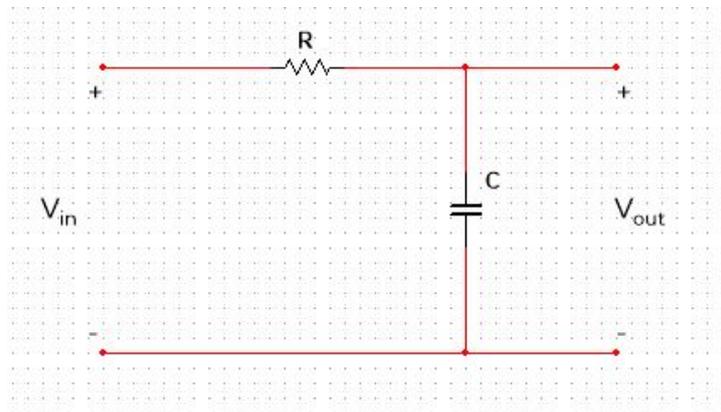


FIGURE 2.20 – Filtre Passe-Bas

4. Convertisseur Courant – Tension :

Un convertisseur de courant est constitué d'un circuit à base d'amplificateurs opérationnels. Il permet d'avoir aux bornes de la sortie la tension électrique voulue à partir du courant qui circule dans le circuit. Puisque le capteur de distance délivre à sa sortie une plage de courant allant de 4 à 20 mA et que l'API nécessite une entrée en tension de 0 à 10 V, il est donc nécessaire de mettre un conditionneur chargé de convertir le courant en tension. [15]



FIGURE 2.21 – Module convertisseur de courant – tension [15]

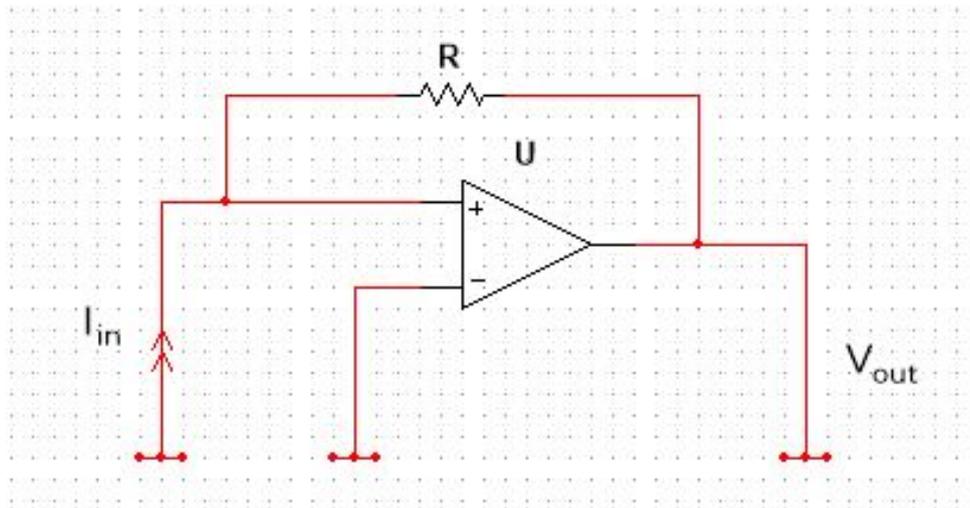


FIGURE 2.22 – Amplificateur

2.9.2 Interfaces de sortie

Il existe dans le prototype une seule interface de sortie.

- **Relais magnétique :**

Les relais magnétiques permettent d'assurer une isolation galvanique entre la partie puissance et la partie commande. Ce sont des dispositifs électriques qui ont un rôle similaire aux interrupteurs sauf que ces derniers peuvent être commandés. Au niveau de la station, il existe 2 relais de Schneider Electric, modèle RSB2A080BDS. Chaque relais sépare l'alimentation principale du signal venant de l'API et allant vers l'électrovanne. Leurs présences dans le circuit est essentielle car elles permettent de commander toute en sécurité, les 2 électrovannes ainsi que les deux voyants LED. [14]



FIGURE 2.23 – Relais Schneider Electric RSB2A080BDS [14]

2.10 Conclusion

Dans ce chapitre on a décrit précisément tout le matériel employé, en mentionnant les caractéristiques de chaque composant et en montrant leurs applications et rôles dans la station de pompage.

Grâce au contenu du chapitre 02, on a arrivé à comprendre le fonctionnement de la station et la façon dont elle a été assemblée afin d'exécuter l'action de pompage d'eau efficacement dans un point en position inférieure vers une position supérieure, en collectant parallèlement des informations importantes utilisées dans la commande et la supervision du processus.

Chapitre 3

Modélisation et Synthèse de commande

3.1 Introduction

La modélisation mathématique consiste à écrire en notation mathématique ce qui est exprimé en phénomènes de la physique en utilisant des variables employées dans des équations en intervenant des contraintes entre ces variables. Le but est de décrire une situation quantitative appelée modélisation.

Le chapitre suivant est consacré à la modélisation mathématique de la station de pompage et à la synthèse des commandes permettant de réaliser le réglage de niveau d'eau dans la station de pompage.

3.2 Modélisation mathématique

La circulation des différents débits au niveau de la station provoquant la variation de volume d'eau dans le réservoir A sert de raisonnement à ce modèle.

Le modèle obtenu permet de synthétiser une loi de commande qui permet d'asservir le système et de simuler le comportement attendu. Le système répond à des excitations durant la simulation sous forme de réponses qui donnent des indications sur la stabilité et les performances globales du système. la validité du modèle est prouvée par les résultats obtenus. [11]

Les équations de la physique :

Dans le réservoir B, il existe deux sorte de débits : débits entrants et débits sortants. Les deux débits font varier le volume d'eau V d'une manière directe :

$$\frac{dV}{dt} = Q_{in} - Q_{out} \quad (3.1)$$

HYPOTHÈSE : L'entraînement du rotor génère un débit noté " Q_{in} ", ce débit présente une proportionnalité à la tension de variation avec un coefficient K . On écrit :

$$Q_{in} = Q_{pompe} = K * v \quad (3.2)$$

D'après la formule de Torricelli [31] énonçant que le carré de la vitesse d'écoulement d'un fluide soumis à sa pesanteur est proportionnel à deux fois la hauteur du fluide dans le cylindre qui le contient et à la gravité terrestre. On note w la vitesse d'écoulement sous l'effet de la gravité :

$$w^2 = 2gh \quad (3.3)$$

L'expression du débit de sortie s'écrit sous la formule suivante.
 α étant le taux d'ouverture de la vanne manuelle :

$$Q_{out} = \alpha.Kv + \alpha.s.w \quad (3.4)$$

On remplace la formule de la vitesse w dans Q_{out} :

$$Q_{out} = \alpha.Kv + \alpha.s\sqrt{2gh} \quad (3.5)$$

La somme des expressions des débits donne :

$$\frac{dV}{dt} = Kv - \alpha.Kv - \alpha.s\sqrt{2gh} \quad (3.6)$$

que le réservoir B est un parallélépipède rectangulaire de surface \mathbf{S} et de hauteur \mathbf{h} , le volume d'eau est exprimé par V qui est le produit de ces deux derniers : $V = S.h$

On aboutit à l'équation régissant sur le système :

$$\boxed{\frac{dh}{dt} = (1 - \alpha) \frac{K}{S} v - \alpha \frac{s}{S} \sqrt{2gh}} \quad (3.7)$$

L'équation ci-dessus représente le modèle mathématique. Les paramètres, leurs symboles et leurs valeurs numériques sont exprimées dans le tableau ci-dessous :

Paramètre	Symbole	Valeur numérique
Surface du réservoir A	S_A	885 cm^2
Surface du réservoir B	$S_B = S$	390 cm^2
Section des tuyaux	s	$4,91 \text{ cm}^2$
Coefficient de proportionnalité	K	$5,833.10^{-5} \text{ m}^3 / s.v$
Gravité terrestre	g	$9,81 \text{ m/s}^2$
Taux d'ouverture	α	$0 \rightarrow 1$
Hauteur du réservoir B	h	$0 \rightarrow 28\text{cm}$
Tension de variation	v	$0 \rightarrow 10v$

TABLE 3.1 – Table significative des paramètres

3.2.1 Représentation d'État du modèle mathématique

Afin de synthétiser une commande, on choisit d'écrire le modèle mathématique sous sa représentation d'état.

On pose :

- \mathbf{x} : la variable d'état représentant la hauteur h du réservoir B
- \mathbf{u} : la variable de commande représentant la tension de commande v
- \mathbf{y} : la hauteur h à observer

$$\Sigma_{NL} : \begin{cases} \dot{x} = F(x, u) = (1 - \alpha) \frac{K}{S} u - \alpha \frac{s}{S} \sqrt{2gx} \\ y = h(x) = x \end{cases} \quad (3.8)$$

Durant la simulation, la valeur de alpha va changer dans le but de tester la robustesse de la commande qu'on va synthétiser.

3.2.2 Fonction de transfert et synthèse de commande linéaire

Afin de faire la synthèse d'une commande linéaire, il faut en premier lieu éliminer la non-linéarité présente dans le modèle : Soit le point d'équilibre $P_0 \left(\begin{smallmatrix} h_0 \\ v_0 \end{smallmatrix} \right)$, tel que pour $x = x_0$, $x \rightarrow$ l'équilibre.

On calcule les paramètres suivants :

$$\begin{aligned} A &= \left. \frac{dF(x, u)}{dx} \right|_{P_0} = -\frac{\alpha g s}{S} \frac{1}{\sqrt{2gh_0}} \\ B &= \left. \frac{dF(x, u)}{du} \right|_{P_0} = (1 - \alpha) \frac{K}{S} \\ C &= \left. \frac{dh(x)}{dx} \right|_{P_0} = 1 \end{aligned}$$

Tous les points de fonctionnement, à l'exception de $\left(\begin{smallmatrix} 0 \\ v_0 \end{smallmatrix} \right)$, sont des points d'équilibre. Donc on choisit une valeur arbitraire de h_0 entre 0 et 0,28 m et v_0 entre 0 et 10 v. Les paramètres A, B et C sont déterminés puis on écrit la représentation d'état du système linéarisé :

Pour $h_0 = 0,25$ m, on écrit :

$$\begin{aligned} A &= -0,055\alpha \\ B &= (1 - \alpha)1,5 \cdot 10^{-3} \\ C &= 1 \end{aligned}$$

$$\Sigma_L : \begin{cases} \Delta \dot{x} = A\Delta x + B\Delta u \\ \Delta y = C\Delta x \end{cases} \quad (3.9)$$

La relation suivante permet d'obtenir la fonction de transfert $G(s)$ du système à partir de la représentation d'état du modèle linéarisé :

$$G(s) = \frac{B}{C[s - A]} \quad (3.10)$$

On donne $\alpha = 0,5$

On aura donc la fonction de transfert suivante :

$$G(s) = \frac{7,5 \cdot 10^{-4}}{s + 0,0275} \quad (3.11)$$

Le pôle $p = -0,275$ est un pôle à partie réelle négative indiquant que le système est stable en BO. A travers la réponse indicielle en BF du système représentée par la figure suivante, on choisira le correcteur linéaire correspondant aux performances voulues.

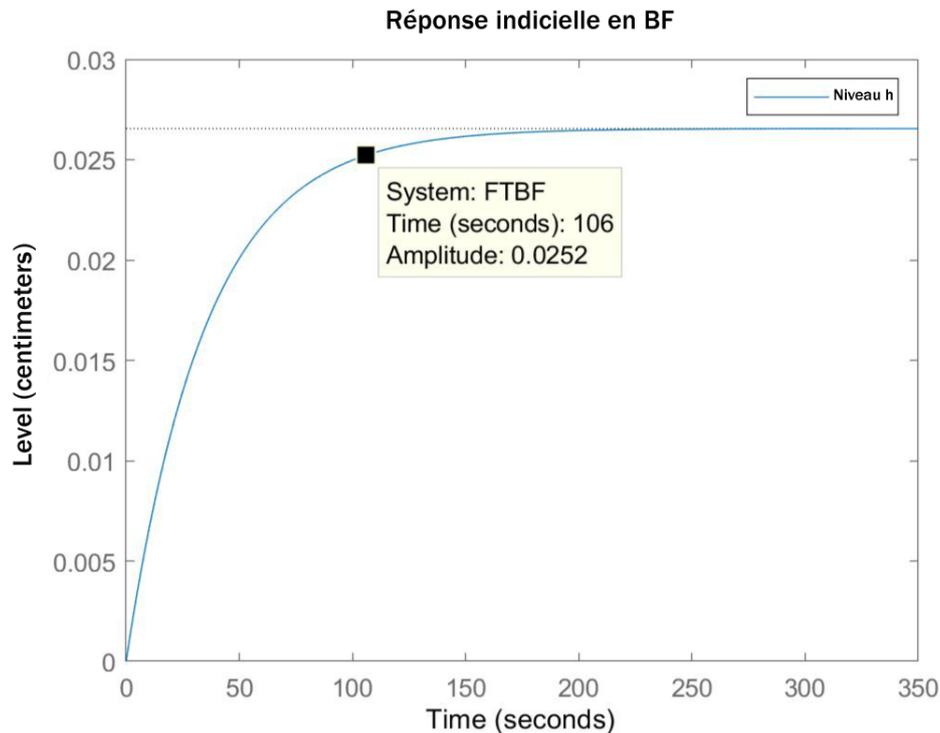


FIGURE 3.1 – Réponse indicielle du système non-corrigé

D'après la réponse indicielle en BF, on choisit les performances suivantes :

- Réponse stable en BF
- Temps de réponse minimal à $\pm 5\%$
- Dépassement minimal
- Erreur statique nulle

Nous avons opté pour un correcteur PI. Ce correcteur assure la stabilité du système en BF et élimine l'erreur statique.

Afin de synthétiser le correcteur PI, nous avons travaillé sur l'environnement MATLAB pour le calcul des paramètres du correcteur PI et son intégration dans la chaîne principale de commande dans le but de simuler tous l'ensemble.

Les gains du correcteur ont été calculés en utilisant la toolbox "PID tuner" sur MATLAB qui prend comme argument la fonction de transfert $G(s)$ citée précédemment. Ce qui donne les valeurs exprimées dans le tableau suivant :

Action	Gain	Valeur numérique
Proportionnelle	K_p	253
Intégrale	K_i	18,5
Dérivée	K_d	0

TABLE 3.2 – Les gains du correcteur PI

La formule du correcteur PI :

$$C(s) = K_p + \frac{K_i}{s} + K_d s = 253 + \frac{18,5}{s}$$

3.2.3 Synthèse de commande robuste (Correcteur en mode glissant)

La commande robuste est un type de commande qui vise à garantir les performances et la stabilité d'un système face à des perturbations du milieu et les incertitudes du modèle. [30]

Les modes glissants constituent une manière simple et robuste de réglage des systèmes automatiques. Cette technique connaît des développements importants dans les domaines de la régulation des processus et de la commande automatique.

Le principe de cette commande est la recherche de l'équation appelée "surface de glissement Σ " déterminée par la surface de Slotine.

$$\Sigma = \left(\frac{d}{dt} + \lambda\right)^{r-1} e(x) \quad (3.12)$$

avec :

- e : l'écart de la variable à régler.
- λ : coefficient positif interprétant la bande passante de commande.
- r : le degré relatif est le nombre de fois qu'il faut dériver la sortie pour que la commande apparaisse.

Le résultat du calcul de la surface de Slotine est exprimée par l'expression suivante :

$$\Sigma = e(x) = x_{ref}(t) - x(t) \quad (3.13)$$

L'objectif est de maintenir l'état x sur la courbe $\Sigma = 0$. Ce qui fait que l'état x revient toujours à suivre la référence d'entrée que nous imposons.

On vérifie les conditions d'invariance et d'attractivité :

1. **Condition d'invariance** ($\dot{\Sigma} = 0$)

$$\frac{d\Sigma}{dt} = \dot{x} - \dot{x}_{ref} = 0 \Rightarrow \dot{x} = 0$$

Si $u = u_e = \frac{s}{K} \frac{\alpha}{(1-\alpha)} \sqrt{2gx}$, on aura donc $\dot{\Sigma} = 0$ et la condition d'invariance est satisfaite.

2. Condition d'attractivité ($\Sigma \cdot \dot{\Sigma} \leq 0$)

Afin de maintenir l'erreur à l'origine ($\Sigma = 0$), on rajoute à la commande équivalente u_e une commande commutante notée u_c .

$u_c = -k \cdot \text{sign}(\Sigma)$ et $u = u_e + u_c$. Cela permet de vérifier la condition d'attractivité :

$$\Sigma \cdot \dot{\Sigma} \leq 0$$

$\Rightarrow -\frac{s}{K(1-\alpha)}k|\Sigma| \leq 0$ est toujours vérifiée pour k **positif**.

Puisque les deux conditions ont été vérifiées pour la surface de glissement Σ , on aura à la fin la commande synthétisée par mode glissant :

$$u_{MG} = \frac{s}{K} \frac{\alpha}{(1-\alpha)} \sqrt{2gx} - k \cdot \text{sign}(x_{ref} - x) \quad (3.14)$$

3.3 Simulation et interprétation des résultats

3.3.1 Simulation avec correcteur PI

Le régulateur PI est mis en place dans la chaîne de commande afin d'observer le comportement du système corrigé en absence/présence de la non-linéarité. Cette étape est essentielle car elle nous permet de savoir si la commande réalisée dans le domaine de Laplace reste valable aussi pour le modèle non-linéaire du système. Les courbes suivantes montrent les deux réponses obtenues par la simulation :

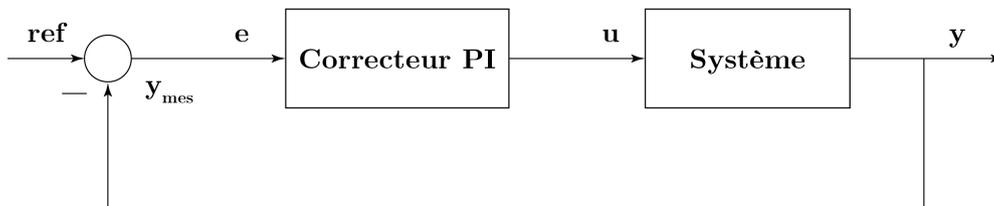


FIGURE 3.2 – Schéma Fonctionnel : Correcteur PI avec Système

Simulation en boucle fermée du système linéaire

La fonction de transfert en boucle fermée (FTBF) du système avec correction est définie sous la forme suivante :

$$FTBF(s) = \frac{C(s)G(s)}{1 + C(s)G(s)} = \frac{0.1898s + 0.0138}{s^2 + 0.2172s + 0.0138} \quad (3.15)$$

La réponse de la FTBF du niveau h à une référence imposée soit 10 cm est présenté dans la figure suivante :

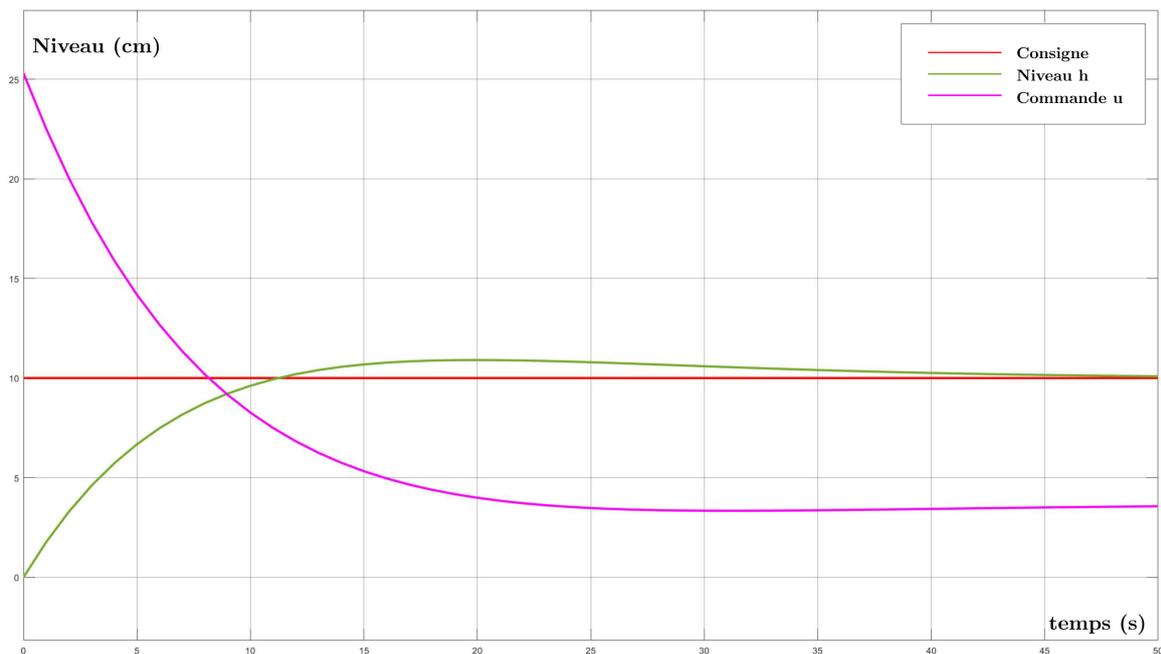


FIGURE 3.3 – Réponse de la FTBF à une consigne

La courbe permet de conclure sur les performances du système avec correcteur PI :

- Stable en BF ;
- réponse en un temps minimal à -5% de l'asymptote horizontale (9,5 s) ;
- dépassement minimal (11 cm) ;
- erreur statique nulle.

Simulation en boucle fermée du système non-linéaire

En gardant la même consigne 10 cm, et en changeant le système linéaire par le système non-linéaire, on observe des changements dans les performances du système.

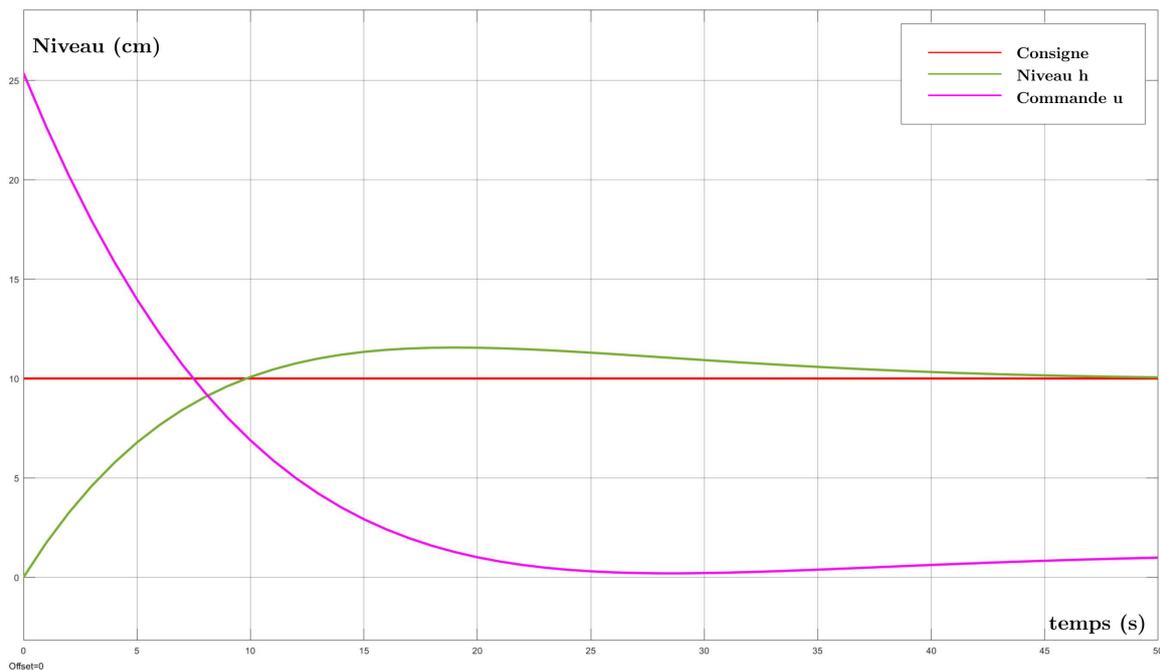


FIGURE 3.4 – Réponse du système non-linéaire à une consigne

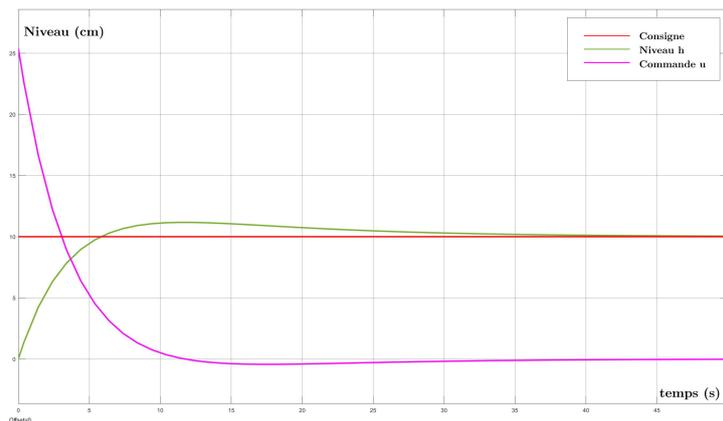
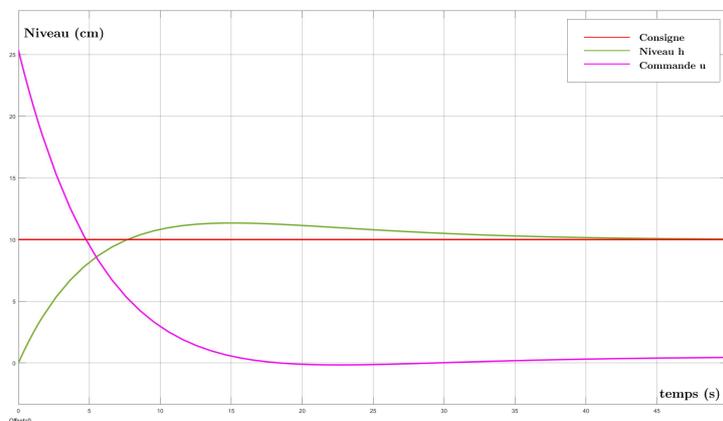
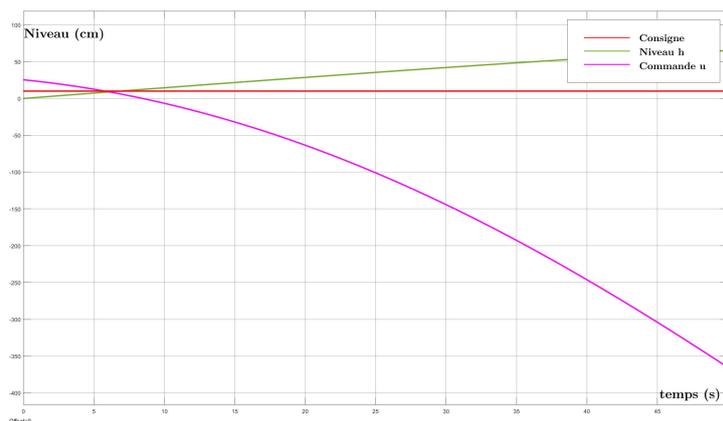
Les performances du système non-linéaire :

- Stable en BF ;
- réponse en un temps minimal a -5% de l'asymptote ; horizontale (8,5 s) ;
- dépassement minimal (12 cm) ;
- erreur statique nulle.

Remarquons que les deux réponses précédentes sont approximativement identiques, cela prouve que la commande synthétisée dans le domaine de Laplace est parfaitement valable en présence de la non-linéarité dans le modèle mathématique.

Test de robustesse du correcteur PI

Le principe est de fixer les coefficient du correcteur PI (K_p et K_i) et de varier le taux d'ouverture α qui entraîne une perturbation dans le modèle mathématique. Si le correcteur permet au système de suivre la consigne et rejeter la perturbation, on conclut que le correcteur est robuste.

FIGURE 3.5 – Réponse du système corrigé avec perturbation $\alpha=0\%$ FIGURE 3.6 – Réponse du système corrigé avec perturbation $\alpha=30\%$ FIGURE 3.7 – Réponse du système corrigé avec perturbation $\alpha=60\%$

D'après les figures, on conclut que le correcteur assure la robustesse seulement pour les valeurs de α inférieures à 50%. Au delà de cette valeur, le système diverge complètement. Par conséquent, il est nécessaire de concevoir un correcteur différent afin de garantir une meilleure performance robuste.

3.3.2 Simulation avec correcteur en mode glissant

Le correcteur synthétisé par mode glissant garantit au système une performance robuste lorsqu'une perturbation aura lieu. Le correcteur en mode glissant est dans l'obligation de satisfaire le cahier de charges imposé dans l'étude.

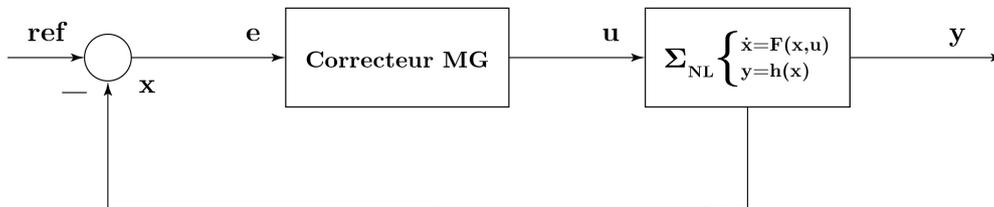


FIGURE 3.8 – Schéma blocs : Correcteur MG et système

En pratique, les performances du système ont été évaluées en présence du correcteur en mode glissant dans le but de tester la correspondance avec le cahier de charges. La figure ci-dessous représente la réponse à une consigne du système corrigé avec le correcteur MG :

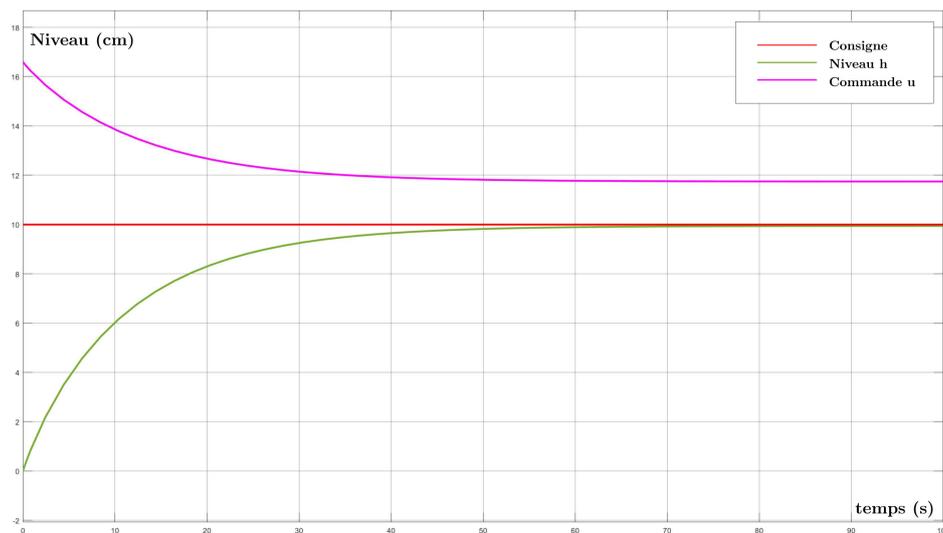


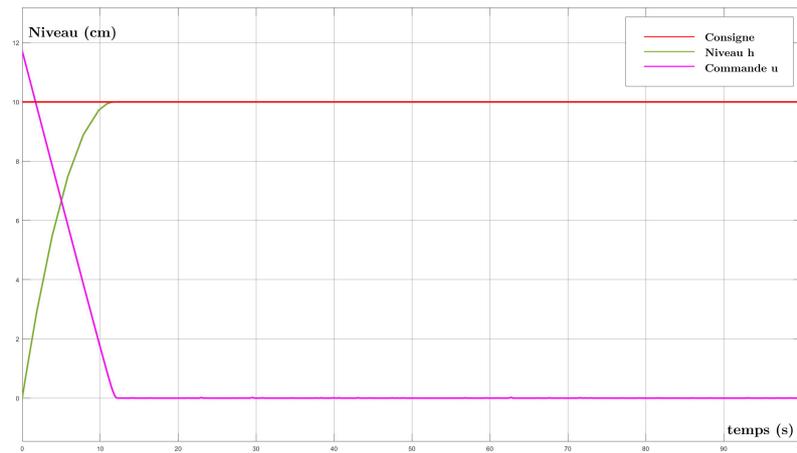
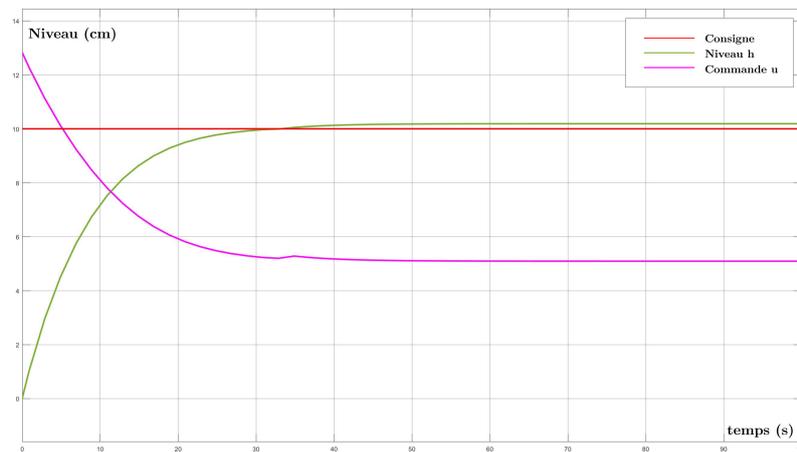
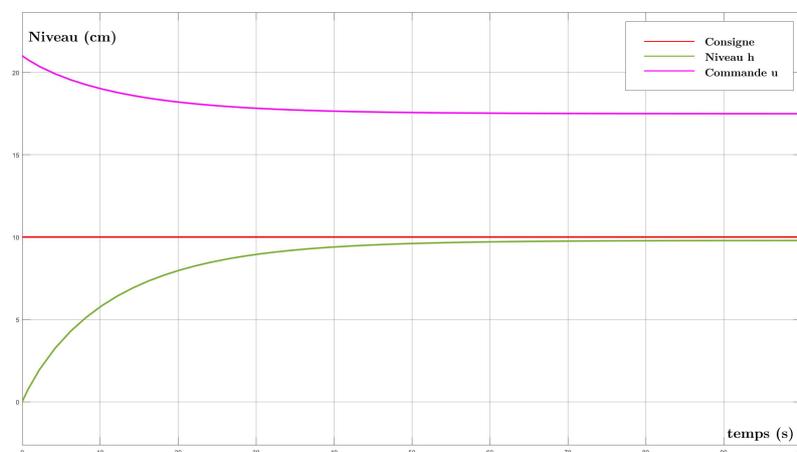
FIGURE 3.9 – Réponse à une consigne du système avec le correcteur MG

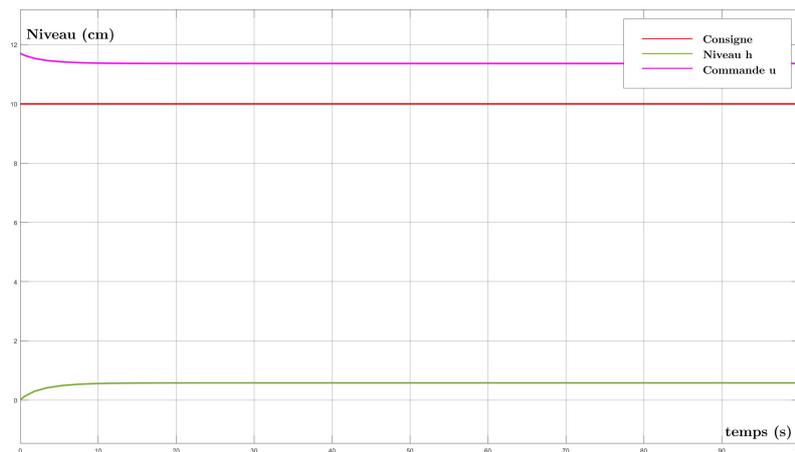
Les performances du système corrigé par le correcteur en MG :

- Stable en BF ;
- réponse en un temps moyennement lent (à -5% $t_r = 35$ s) ;
- absence de dépassement ;
- erreur statique minimale.

Le test de robustesse est réalisé en variant la valeur du taux d'ouverture α causant une perturbation sur le modèle du système. Simultanément, Le correcteur MG est rajouté au système perturbé prouvant sa performance robuste.

Les figures suivantes démontrent les réponses du système corrigé par le correcteur MG et en présence d'une perturbation à trois échelles (0%, 30% et 60%).

FIGURE 3.10 – Réponse à une consigne avec perturbation ($\alpha = 0\%$)FIGURE 3.11 – Réponse à une consigne avec perturbation ($\alpha = 30\%$)FIGURE 3.12 – Réponse à une consigne avec perturbation ($\alpha = 60\%$)

FIGURE 3.13 – Réponse à une consigne avec perturbation ($\alpha = 80\%$)

Interprétation des graphes

On remarque que les dynamiques du système commandé par un régulateur en mode glissant et en présence d'une perturbation changent en fonction du degré de cette dernière, et cela se répercute sur les performances du système lors de la poursuite de consigne.

Dans la première figure, lorsqu'on réduit α à 0%, c'est-à-dire que la vanne manuelle devient complètement fermée, on remarque que le temps de réponse chute de 35 s à 9 s en raison de la traversée sans pertes de l'eau dans la canalisation. En d'autres termes, le temps pour que le niveau réponde à la consigne est largement réduit car il n'y a pas eu de pertes.

Lorsque $\alpha=30\%$, on remarque que le système reste stable et le temps de réponse baisse de 11 mais l'erreur statique augmente légèrement (1% de la valeur finale).

Dans le cas d'origine où $\alpha=50\%$, on remarque une dégradation dans la poursuite de référence, ce qui provoque augmentation de l'erreur statique à 5%.

Dans le cas de $\alpha=60\%$, on remarque une augmentation du temps de réponse à 45s, car les pertes s'accroissent en augmentant la valeur de α .

En dernier quand $\alpha=80\%$, le système est stable mais n'assure pas la poursuite de consigne.

A partir des interprétations précédentes, on conclut que la commande par mode glissant est très robuste et permet au système de poursuivre la consigne qu'on lui impose face à la perturbation introduite par le taux d'ouverture α . Sauf pour la cas dernier où le correcteur ne permet pas la poursuite de consigne. On justifie cela par la limitation en puissance du matériel de la station de pompage, spécifiquement la tension de contrôle du variateur qui est limitée entre 0 à 10 volts et le débit de la pompe plafonné à 35 l/min.

Pour remédier à ce problème, il est recommandé d'installer des dispositifs plus performants afin de bénéficier pleinement de la robustesse de la commande synthétisée par mode glissant.

3.4 Conclusion

La modélisation en Automatique est une étape très essentielle, elle permet d'acquérir les informations déterminant l'identité d'un système dynamique. Ce dernier est résumé par une équation mathématique. L'exploitation de ces informations permet à l'aboutissement aux objectifs de l'étude.

La modélisation a permis de réaliser deux commandes donnant la possibilité d'asservir le système et de simuler son comportement. Les résultats obtenus sont satisfaisants, le système répond aux consignes qu'on lui impose et suit un comportement voulu avec la possibilité d'assurer une poursuite robuste de consigne.

À travers la simulation du système, on conclut que le modèle proposé est valide et les performances du système corrigé sont favorables.

Chapitre 4

Implémentation et supervision de la station de pompage

4.1 Introduction

La station de pompage comprend un ensemble de dispositifs et composants qui interagissent selon un cahier de charges que nous imposons. L'action principale de la station est de déplacer l'eau, souvent trouvée en position inférieure, vers une position supérieure, recueillie par un réservoir. Cette action est maintenue d'une façon continue sous contrôle car ça nécessite une certaine rapidité d'acquisition d'informations afin de prendre les mesures adéquates lorsque la station fait face à des anomalies. C'est pour cela que les Automates Programmables jouent un rôle capital dans l'automatisation industrielle, grâce à leur rapidité de traitement d'informations et la faculté de contrôle de tous l'ensemble d'un système industriel.

A la fin de ce chapitre, nous aurons vu toutes les étapes permettant de développer une gestion autonome de notre station de pompage, en partant d'un cahier des charges élaboré, jusqu'aux programmes de commande de ce processus et interfaces de supervision.

4.2 Cahiers de Charges

Les cahiers de charges définissent le mode de fonctionnement de la station de pompage. Le système exécute des opérations selon des conditions, dictées par l'utilisateur et implémentées dans l'Automate Programmable. Puis, le contrôle de ces opérations est fait à partir de l'Interface Homme/Machine.

4.2.1 Cahier de Charges I : Mode opératoire automatique

Le premier cahier de charges [4] englobe le processus de réglage de niveau dans le réservoir B en réalisant un asservissement avec des performances optimales. Cet asservissement se fait par deux commandes qu'on a synthétisées dans le chapitre précédent 3.2.2.

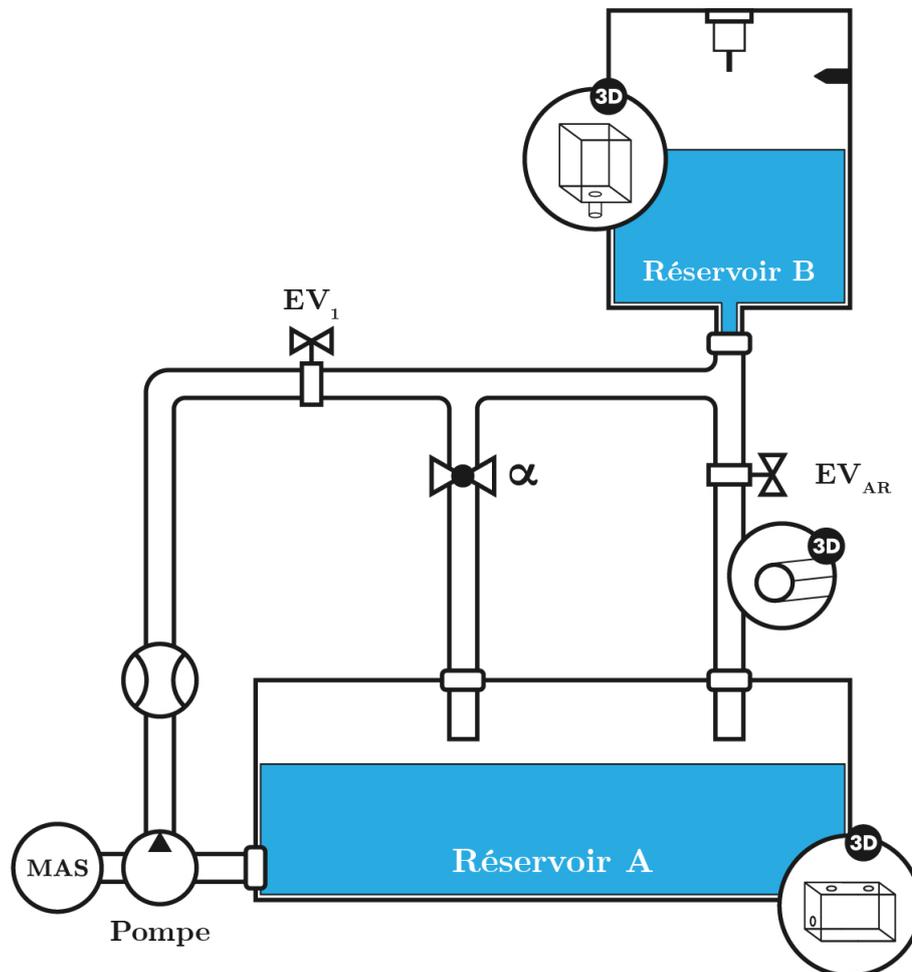


FIGURE 4.1 – Schéma représentatif de la station de pompage

Le mode opératoire autonome est le mode de fonctionnement en boucle fermée de la station. En donnant la consigne à partir de l'IHM, le système effectue les opérations suivantes :

1. **Remplissage** : si le système est en marche et la consigne choisie est strictement supérieure au niveau h , la pompe et l'électrovanne EV_1 se déclenchent, l'électrovanne 2 EV_2 se ferme et le voyant de remplissage s'allume. L'opération s'arrête lorsque la consigne est inférieure au niveau h , puis le système revient à l'état d'arrêt.
2. **Vidange** : si le système est en marche et la consigne choisie est strictement inférieure au niveau h , l'électrovanne 2 EV_2 s'ouvre et le voyant de vidange s'allume. L'opération s'arrête lorsque la consigne est supérieure au niveau h , puis le système revient à l'état d'arrêt.
3. **Arrêt d'urgence** : lorsque le système est en marche et que le bouton d'arrêt

d'urgence est déclenché, le système s'interrompt et s'arrête immédiatement.

Maintenant connaissant le cahier de charges à développer, nous élaborons le GRAFCET suivant :

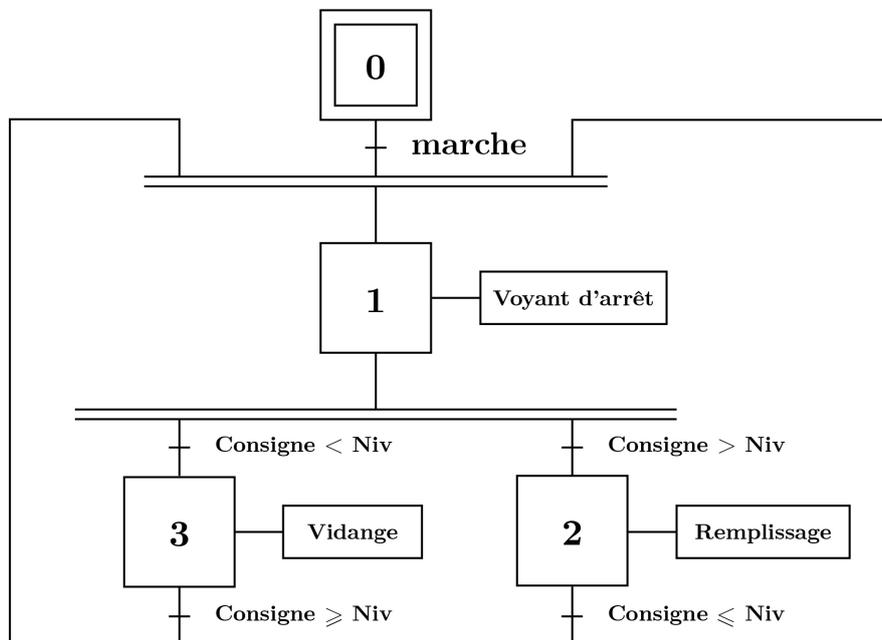


FIGURE 4.2 – GRAFCET décrivant le Cahier de Charges I

Le GRAFCET est une représentation adaptée aux systèmes ayant un aspect séquentiel, il permet de les décomposer en plusieurs étapes. Cela facilite l'implémentation dans les logiciels de programmation et l'interprétation des informations contenues dans le système.

Le GRAFCET se compose de plusieurs étapes appelées **états**, les conditions d'activation et de désactivation de ces états, ainsi que des sorties se déclenchant lorsque l'état est actif.

Détails du GRAFCET

Ci-dessous, nous listons toutes les entrées et sorties de notre premier cahier des charges :

1. **Variables d'entrée :**

- Variables analogiques : Niveau capteur, Débit Capteur.
- Variables TOR : Reset, Marche, Arrêt.

2. **Variables de sortie :**

- Variables analogiques : Commande variateur.
- Variables TOR : Voyants 1 \Rightarrow 3, Relais des électrovannes 1 et 2, Relais du variateur.

4.2.2 Cahier de Charges II : Mode opératoire cyclique

Le mode de fonctionnement cyclique permet au système d'exécuter des fonctions dans un aspect répétitif. L'utilité de ce mode est lorsque on souhaite maintenir un état fixe de la station par rapport à une période de temps. Le fonctionnement cyclique de la station de pompage est déclenché dans notre cas lorsqu'on veut maintenir le niveau d'eau dans le réservoir à un niveau déterminé en fixant une durée de temps \mathbf{T} . On a rajouté aussi un compteur qui s'incrémente \mathbf{n} fois pour s'assurer que le niveau d'eau reste toujours à la valeur déterminée pendant la durée de \mathbf{T}/\mathbf{n} .

Pendant ce mode, le système effectue les opérations suivantes :

1. **Première temporisation** : lorsque le système est en marche, et que le niveau est inférieur à 60% de sa valeur maximale. Le système initialise une temporisation d'une minute et passe à la phase de remplissage. Dans le cas où le niveau est supérieur à 60% de sa valeur maximale, le système passe directement à la seconde temporisation.
2. **Remplissage** : si le niveau h est inférieur à 60% de sa valeur maximale, et que la première temporisation est écoulée, la pompe et l'électrovanne 1 se déclenchent, l'électrovanne 2 se ferme et le voyant de remplissage s'allume jusqu'à ce que le niveau atteigne au moins 60% de sa valeur maximale.
3. **Seconde temporisation** : quand le niveau h est au moins à 60% de sa valeur maximale, la seconde temporisation se déclenche et s'écoule pendant 1 journée (10 s dans le programme de simulation).
4. **Opération de comptage** : le comptage de cycles commence lorsque la deuxième temporisation s'est écoulée. Le cycle est répété 3 fois, de la première temporisation à la seconde. Si la valeur de comptage est inférieure à 3, le cycle est répété. Le comptage se termine au quatrième cycle et la machine entre dans l'état de vidange.
5. **Vidange** : l'étape du vidange s'exécute automatiquement lorsque la valeur de comptage est supérieure à 3, ce qui entraîne l'ouverture de l'électrovanne 2 et l'allumage du voyant de vidange. L'opération s'arrête lorsque le niveau h est au plus égale au niveau h minimal, puis le système revient à son état initial et le compteur est remis à zéro.

A partir du cahier de charges, nous élaborons le GRAFCET suivant :

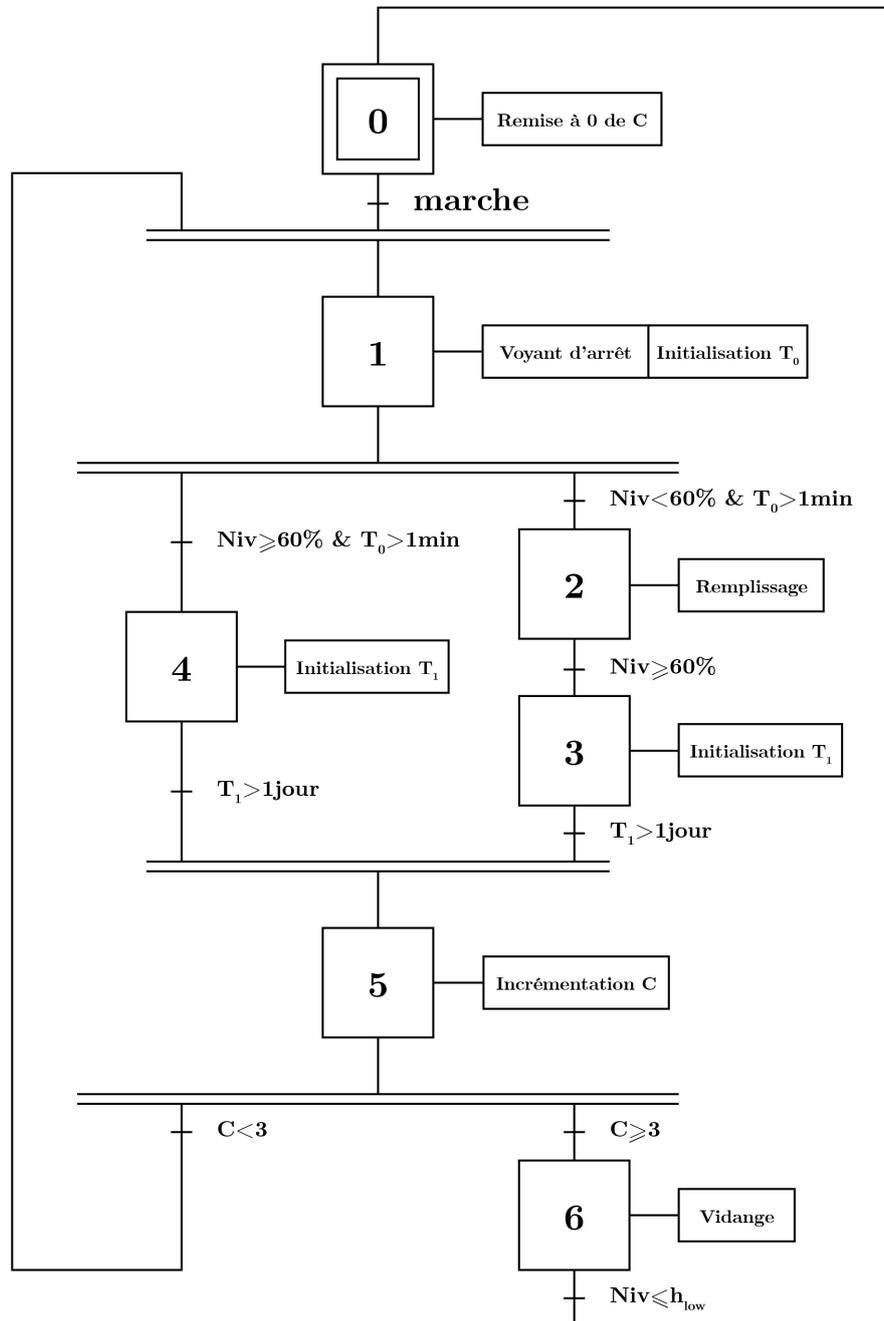


FIGURE 4.3 – GRAFCET décrivant le Cahier de Charges II

Détails du GRAFCET

Ci-dessous, nous listons toutes les entrées et sorties contenues dans le GRAFCET de notre premier cahier des charges :

1. **Variables d'entrée :**
 - Variables analogiques : Niveau capteur, Débit Capteur ;
 - variables TOR : Reset, Marche, Arret.
2. **Variables de sortie :**
 - Variables analogiques : Commande variateur ;
 - variables TOR : Voyants 1 \Rightarrow 3, Relais des électrovannes 1 et 2, Relais du variateur.
3. **Variables de temporisation :**
 - Timer T_0 retardé au travail pendant 5 s ;
 - timer T_1 retardé au travail pendant 10 s.
4. **Variables de comptage :**
 - Compteur C a 3 états.

4.2.3 Équations d'état

Afin de simplifier l'implémentation des GRAFCET, il est recommandé de transcrire cela en plusieurs équations de logique qui déterminent les états du système. Chaque état est représenté par deux équations qui indiquent l'activation et la désactivation de cet état, ainsi qu'une équation de sortie qui désigne l'instruction à faire lorsque le système est dans cet état.

Caractéristiques des équations d'état

- On note les états par x_n
- On note l'équation d'activation S_n et l'équation de désactivation R_n
- Dans une équation, une addition (+) représente un OU logique. Une multiplication (.) représente un ET logique.

Avec n allant de l'état initial à l'état final.

Équations du cahier de charges I

Équations de l'état initial :

$$x_0 : \begin{cases} S_0 = reset \\ R_0 = x_0.marche \end{cases} \quad (4.1)$$

Équations de l'état 1 :

$$x_1 : \begin{cases} S_1 = x_0.marche + x_2.(ref \leq niv) + x_3.(ref \geq niv) \\ R_1 = x_1.(ref > niv) + x_1.(ref < niv) + reset \\ V_1 = x_1 \end{cases} \quad (4.2)$$

Équations de l'état 2 :

$$x_2 : \begin{cases} S_2 = x_1.(ref > niv) \\ R_2 = x_2.(ref \leq niv) + reset \\ V_2 = M_{var} = Pompe = EV_1 = x_2.\overline{arret} \end{cases} \quad (4.3)$$

Équations de l'état 3 :

$$x_3 : \begin{cases} S_3 = x_1.(ref < niv) \\ R_3 = x_3.(ref \geq niv) + reset \\ V_3 = EV_2 = x_3.\overline{arret} \end{cases} \quad (4.4)$$

Équations du cahier de charges II

Équations de l'état initial :

$$x_0 : \begin{cases} S_0 = x_6.(niv \leq h_{low}) + Reset \\ R_0 = x_0.marche \\ RAZ_C = x_0 \end{cases} \quad (4.5)$$

Équations de l'état 1 :

$$x_1 : \begin{cases} S_1 = x_0.marche + x_5.(C < 3) \\ R_1 = x_1.(T_0 > 5s) + reset \\ V_1 = INIT T_0 = x_1 \end{cases} \quad (4.6)$$

Équations de l'état 2 :

$$x_2 : \begin{cases} S_2 = x_1.(niv < 60\%).(T_0 > 1min) \\ R_2 = x_2.(niv \geq 60\%) + reset \\ V_2 = M_{var} = Pompe = EV_1 = x_2.\overline{arret} \end{cases} \quad (4.7)$$

Équations de l'état 3 :

$$x_3 : \begin{cases} S_3 = x_2.(niv \geq 60\%) \\ R_3 = x_3.(T_1 > 1jr) + reset \\ INIT T_1 = x_3 \end{cases} \quad (4.8)$$

Équations de l'état 4 :

$$x_4 : \begin{cases} S_4 = x_1.(niv \geq 60\%).(T_0 > 1min) \\ R_4 = x_4.(T_1 > 1jr) + reset \\ INIT T_1 = x_3 \end{cases} \quad (4.9)$$

Équations de l'état 5 :

$$x_5 : \begin{cases} S_5 = (x_3 + x_4).(T_1 > 1jr) \\ R_5 = x_5 + reset \\ INCRC = x_5 \end{cases} \quad (4.10)$$

Équations de l'état 6 :

$$x_6 : \begin{cases} S_6 = x_5.(C \geq 3) \\ R_6 = x_6.(niv \leq h_{low}) + reset \\ V_3 = EV_2 = x_6.\overline{arret} \end{cases} \quad (4.11)$$

4.3 Implémentation des cahiers de charges sur TIA Portal

4.3.1 Programme de l'API

L'écriture du programme sur TIA Portal exige une certaine structure car un programme d'API sur ce logiciel se compose de plusieurs blocs réalisant des fonctions bien définies. Sur TIA Portal, il existe :

- Des blocs d'organisation (OB : Organization Block)
- Des fonctions (FC : Function)
- Des blocs fonctionnels (FB : Function Block)
- Des blocs de données (DB : Data Block)

En ce qui suit, nous allons montrer le programme écrit sous le logiciel STEP7 inclut dans TIA Portal, en le décomposant en plusieurs parties selon le type de blocs utilisés.

Bloc d'organisation - Main

Les blocs d'organisation représentent l'interface entre l'utilisateur et le système d'exploitation (souvent Microsoft Windows). Ce bloc prend le rôle du **Main** ou le bloc principal de notre programme qui va contenir toutes les instructions de l'API à exécuter.

Dans le Main, les équations d'état établis précédemment sont écrites avec le langage LADDER, chaque réseau contient une équation d'activation ; de désactivation ou de sortie (S_n/R_n). Voici l'exemple des équations du premier cahier de charges écrites dans le Main en utilisant le langage LADDER : L'état x_0 du premier cahier de charges :



FIGURE 4.4 – Équation d'activation de l'état x_0

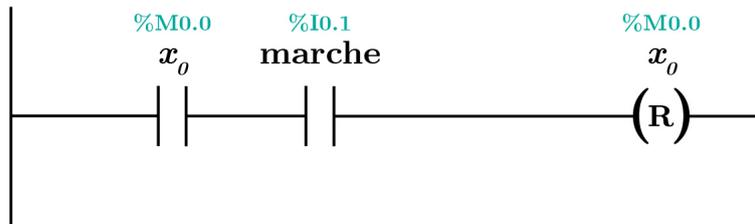


FIGURE 4.5 – Équation de désactivation de l'état x_0

L'état x_1 du premier cahier de charges :

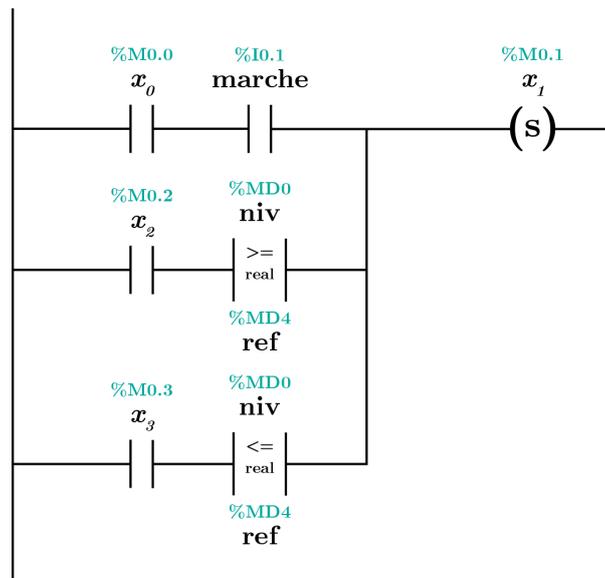


FIGURE 4.6 – Équation d'activation de l'état x_1

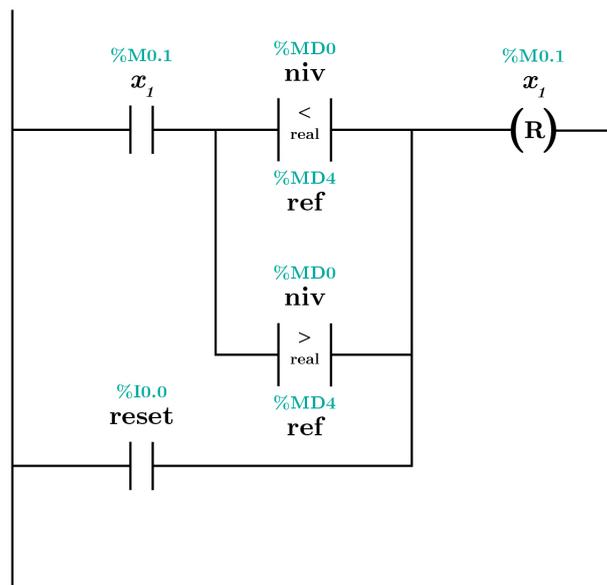


FIGURE 4.7 – Équation de désactivation de l'état x_1

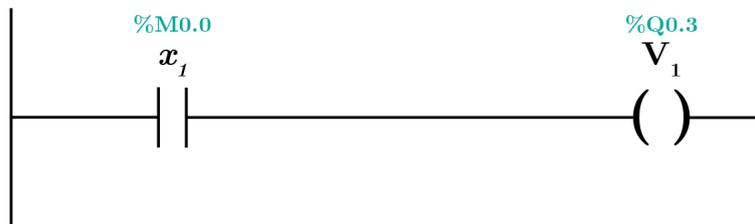


FIGURE 4.8 – Équation de sortie de l'état x_1

L'état x_2 du premier cahier de charges :

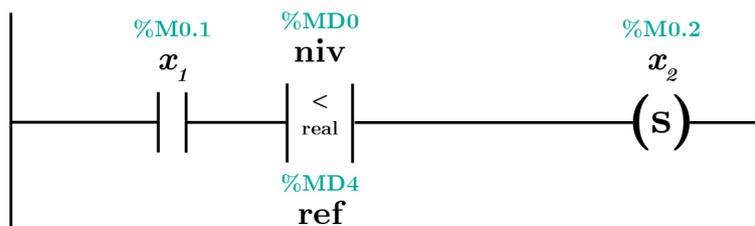


FIGURE 4.9 – Équation d'activation de l'état x_2

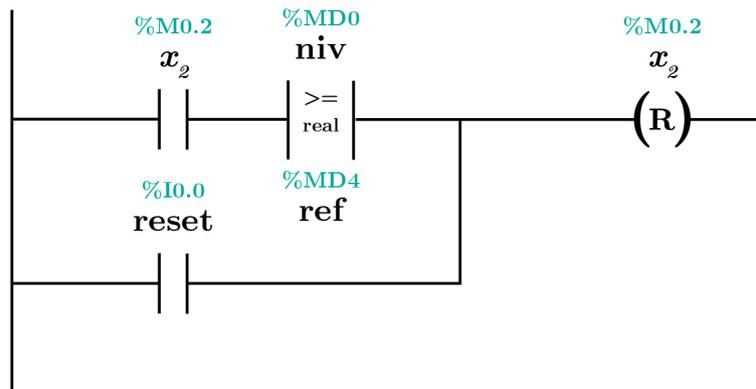


FIGURE 4.10 – Équation de désactivation de l'état x_2

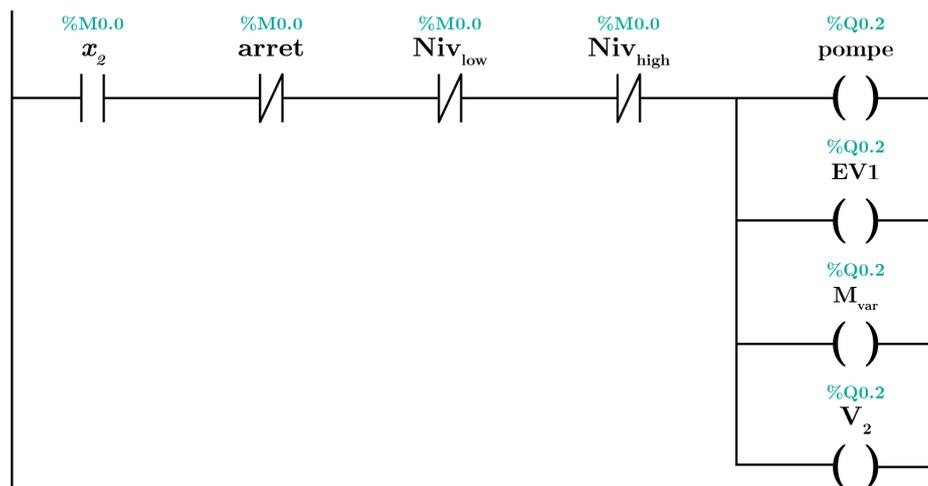


FIGURE 4.11 – Équation de sortie de l'état x_2

L'état x_3 du premier cahier de charges :

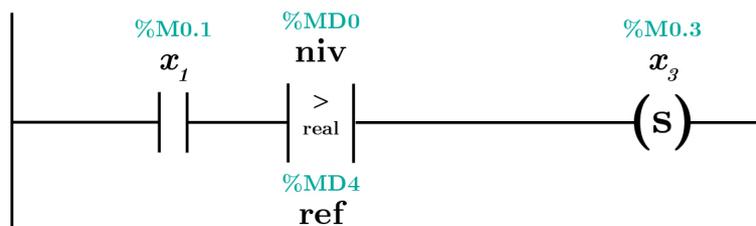


FIGURE 4.12 – Équation d'activation de l'état x_3

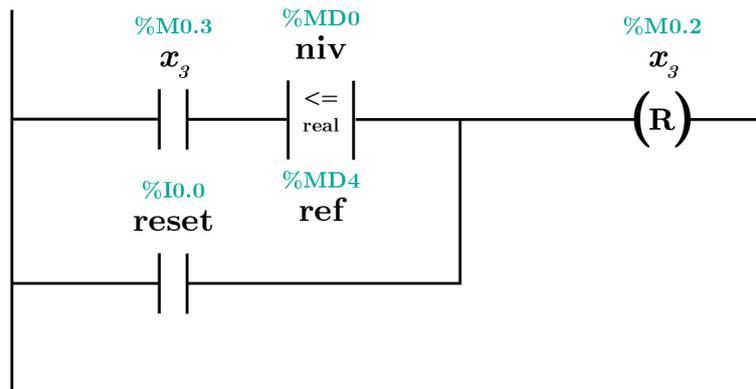


FIGURE 4.13 – Équation de désactivation de l'état x_3

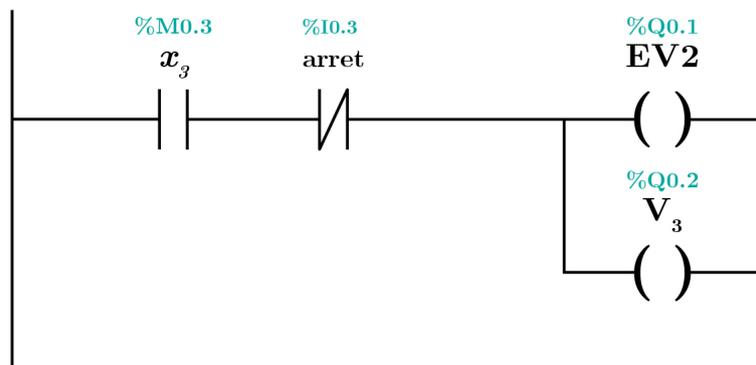


FIGURE 4.14 – Équation de sortie de l'état x_3

Fonctions et Blocs Fonctionnels

Les fonctions et blocs fonctionnels sont des programmes écrits en dehors du programme principal. Ils ont exactement les mêmes fonctionnalités que le Main mais contrairement à cela, ils sont appelés directement par l'utilisateur et non par le système d'exploitation.

Dans notre programme, nous avons utilisé plusieurs fonctions du à la complexité du Main, le but de l'utilisation de ces fonctions est d'avoir un nombre réduit de lignes de code, ce qui aboutit à un programme principal plus au moins compact.

- Fonction de limitation** : Permet de saturer à des seuils définies, la valeur d'une grandeur comme la référence de niveau d'entrée, la tolérance de niveau, les paramètres du correcteur PI...etc. La fonction prend comme arguments la grandeur à limiter, sa valeur maximale et sa valeur minimale.

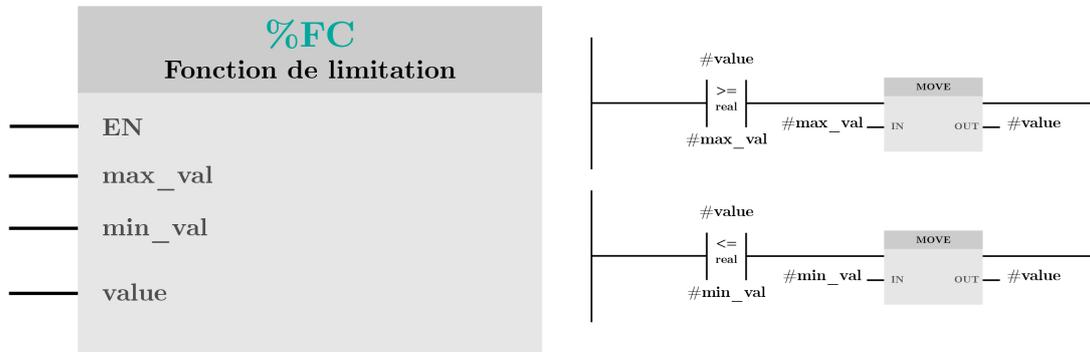


FIGURE 4.15 – Fonction de limitation

- Fonction d'initialisation** : Cette fonction s'active lorsque le système est en marche, son rôle est d'initialiser les paramètres du correcteur PI si ces derniers sont nuls, les paramètres d'initialisation sont les paramètres du correcteur synthétisé dans le chapitre précédent. Elle prend comme arguments les paramètres K_p , K_i et k_d introduits à partir de l'IHM.

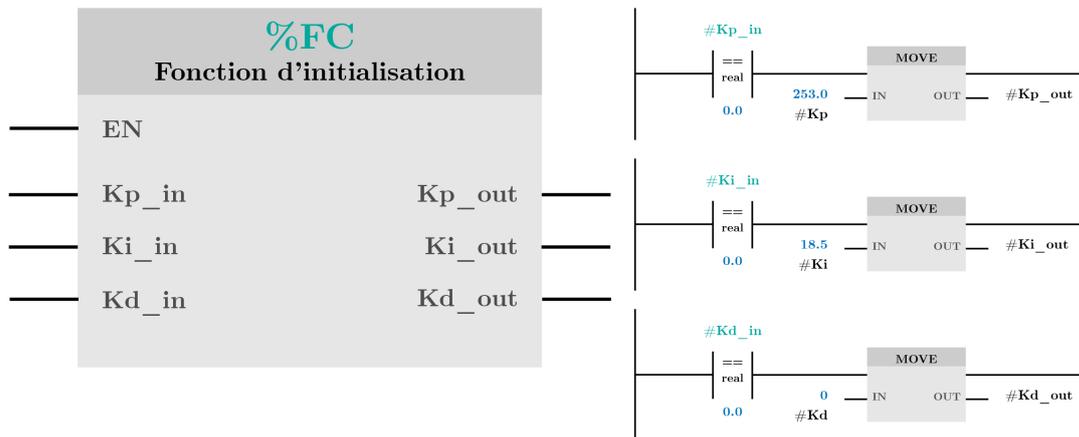


FIGURE 4.16 – Fonction d'initialisation

- Fonction PID** : Cette fonction calcule la commande PI qu'on a synthétisé précédemment. Elle prend comme arguments les paramètres du PID (K_p , K_i , K_d), la référence de niveau, le niveau h et la valeur initiale de la somme intégrale. Elle retourne la commande u écrite sous cette équation :

$$u = K_p * erreur + K_i * somme + K_d * difference \tag{4.12}$$

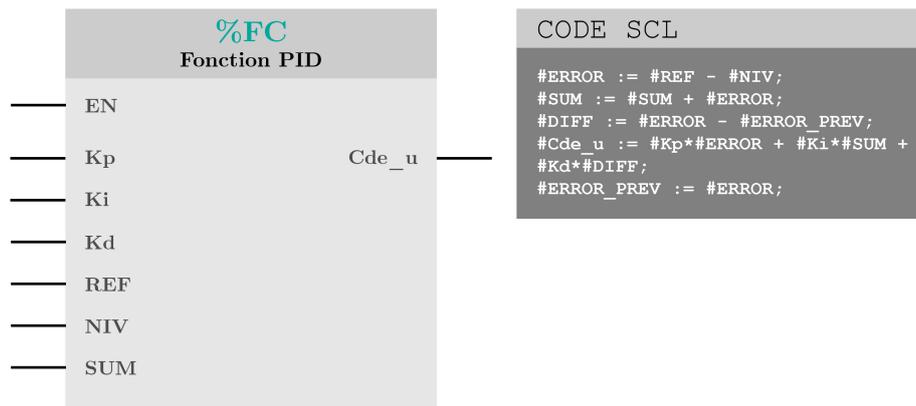


FIGURE 4.17 – Fonction PID

- Fonction SMC** : Cette fonction calcule la commande par mode glissant (SMC) qu'on a synthétisé précédemment. Elle prend comme arguments le gain de commutation K_{SMC} , la référence de niveau et le niveau h. Elle retourne la commande u écrite sous cette équation :

$$u = 37.28 * \text{sqrt}(\text{niveau}) - K_{SMC} * \text{sign}(\text{reference} - \text{niveau}) \quad (4.13)$$

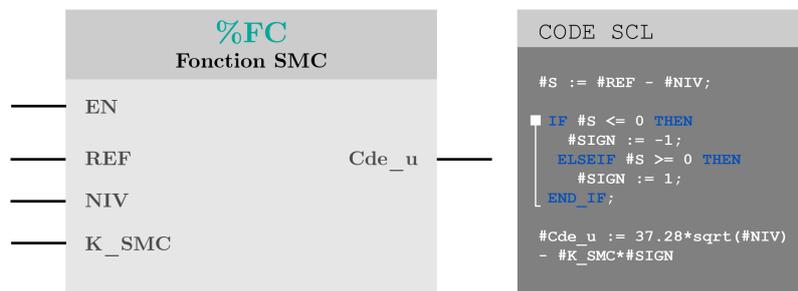


FIGURE 4.18 – Fonction SMC

- Tolérance** : Cette fonction calcule la référence tolérée par le système. L'utilisateur introduit la tolérance en pourcentage et la référence de niveau. La fonction retourne la valeur tolérée de la référence.

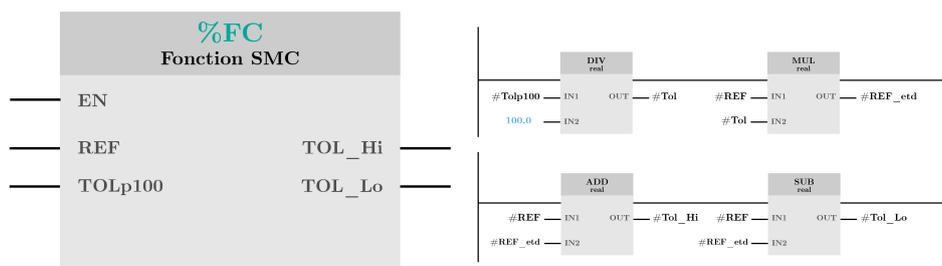


FIGURE 4.19 – Tolérance

- Fonction Scale** : Cette fonction permet de faire la mise en échelle d'une grandeur analogique comme la valeur de lecture du capteur de niveau et le débitmètre, puis elle normalise cette valeur et la multiplie sur un intervalle introduit par l'utilisateur. Elle retourne une grandeur réelle mise à l'échelle.

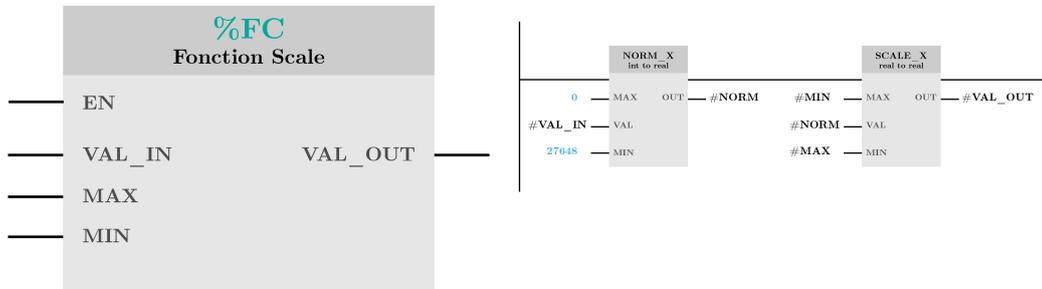


FIGURE 4.20 – Fonction Scale

- Fonction Unscale** : Cette fonction fait le travail inverse de la fonction précédente. Elle normalise une valeur calculée dans l'API comme la commande PI ou la commande par mode glissant sur un intervalle introduit par l'utilisateur, puis la multiplie sur une échelle de 0 à 27648. Elle retourne une grandeur entière.

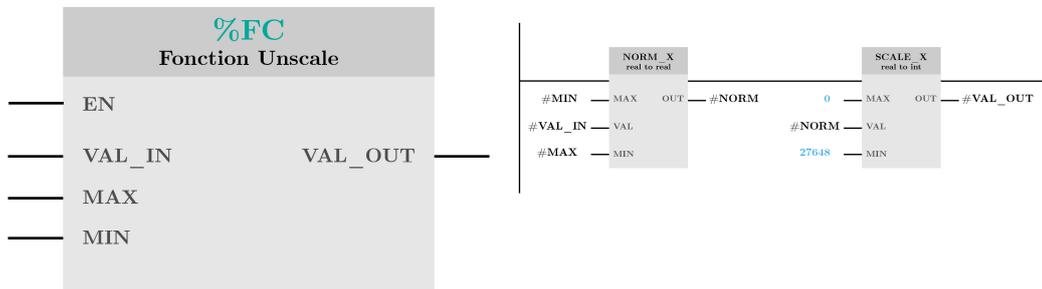


FIGURE 4.21 – Fonction Unscale

- Bloc Fonctionnel - Cahier de charges II** : En raison de la similarité des deux cahiers de charges, nous avons décidé de créer un bloc fonctionnel comportant la totalité du second cahier de charges. Ce bloc prend comme arguments :
 1. Les états du GRAFCET (x_0 à x_6).
 2. Le niveau h.
 3. La condition du niveau inférieure.

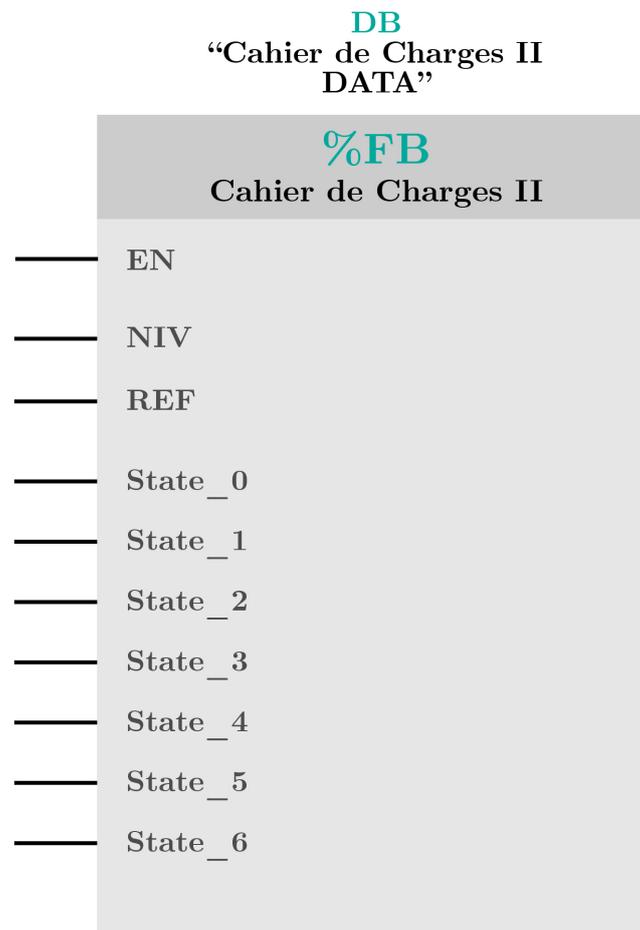


FIGURE 4.22 – Bloc Fonctionnel - Cahier de charges II

Blocs de Données

Dans notre programme principal, nous avons utilisé des blocs de données afin de sauvegarder les données des variables puisque il y a certains blocs qui les nécessitent pour exécuter des instructions. Parmi les blocs qui emploient les blocs de données, il y a :

- Les timers.
- Les compteurs.
- Les blocs fonctionnels.

Les timers et les compteurs ont besoin de sauvegarder la donnée du temps écoulé (ou la donnée de comptage pour le cas du compteur) afin de l'utiliser pour déclencher des conditions par exemple, dans notre cas, nous avons utilisé cette donnée afin d'activer ou désactiver des états dans le second cahier de charges.

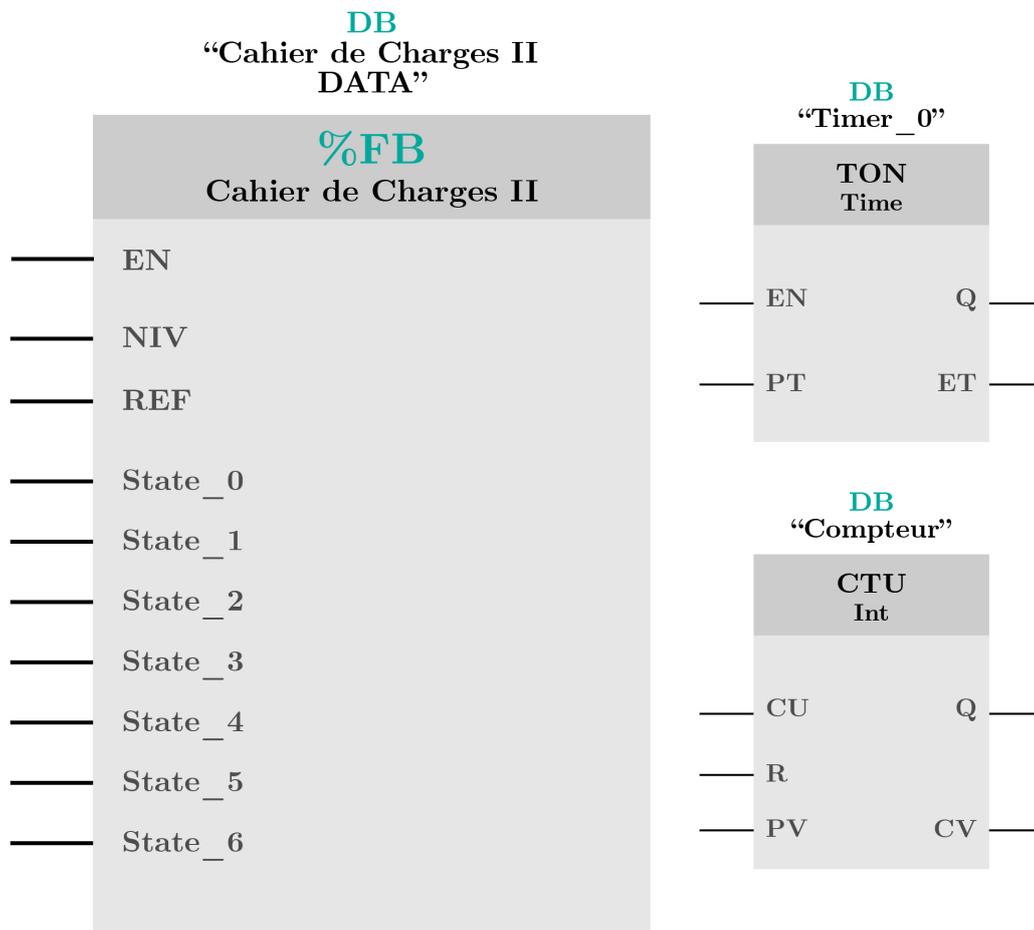


FIGURE 4.23 – Blocs utilisant les Blocs de Données

4.3.2 Programme de l'IHM

Pour superviser et contrôler la station de pompage, il est nécessaire de munir l'automate d'un intermédiaire afin qu'on puisse communiquer avec. Pour se faire, nous utilisons une Interface Homme Machine (IHM).

Le programme IHM est écrit sur WinCC, un système de commande et d'acquisition de données dans TIA Portal.

WinCC nous permet de créer des **Vues**. Chaque Vue est personnalisée selon les besoins de l'utilisateur, elle peut contenir toutes les informations des équipements qu'on trouve dans la station.

Vue Principale

La vue principale est la vue d'accueil de l'IHM, elle donne accès à la navigation sur les autres vues.

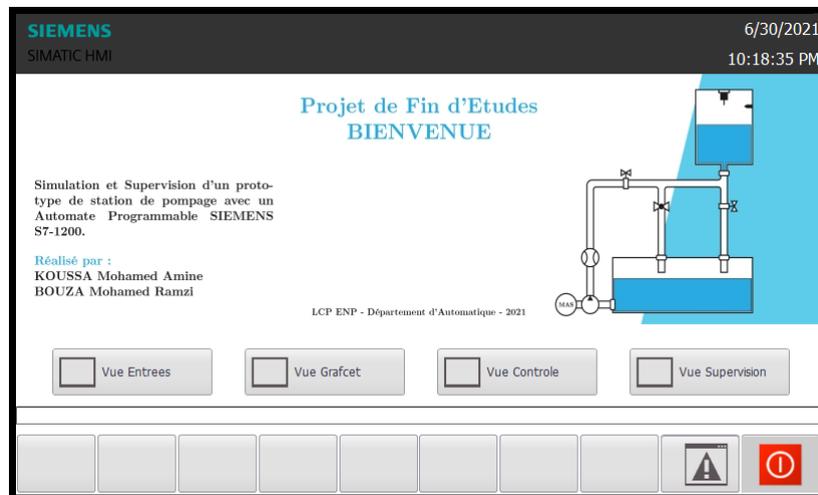


FIGURE 4.24 – Vue principale de l'IHM

Vue Entrées

La vue des entrées contient toutes les entrées de notre système ainsi que les champs de saisi de la référence d'entrée et de tolérance. Les entrées sont :

1. Reset
2. Marche
3. Arrêt



FIGURE 4.25 – Vue Entrées de l'IHM

Vue GRAFCET

Cette vue contient les deux GRAFCET de nos cahiers de charges. Elle permet d’observer les états du GRAFCET quand ils s’activent, ainsi que certaines variables comme le valeur de temporisation d’un timer et de comptage d’un compteur.

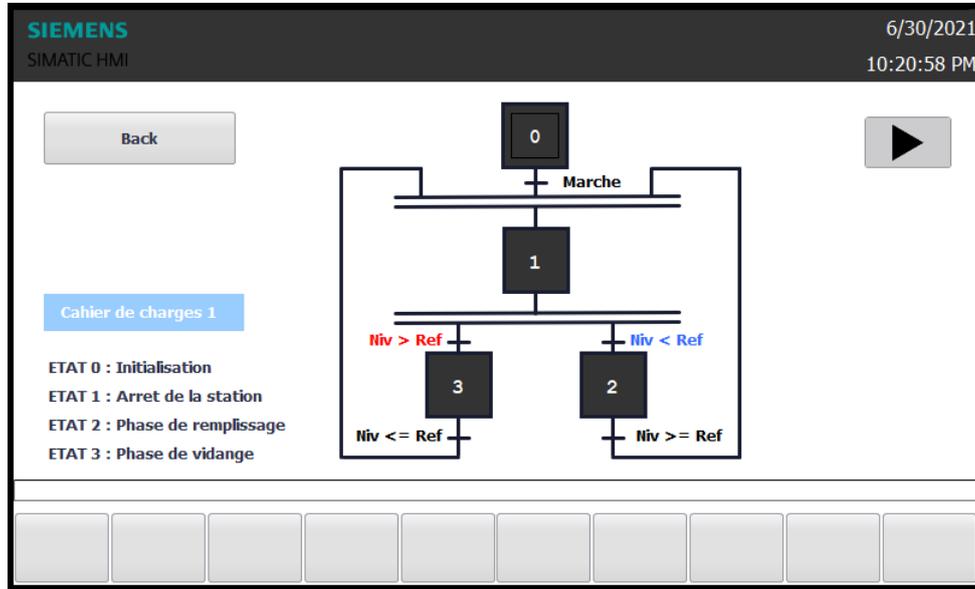


FIGURE 4.26 – Vue GRAFCET I de l’IHM

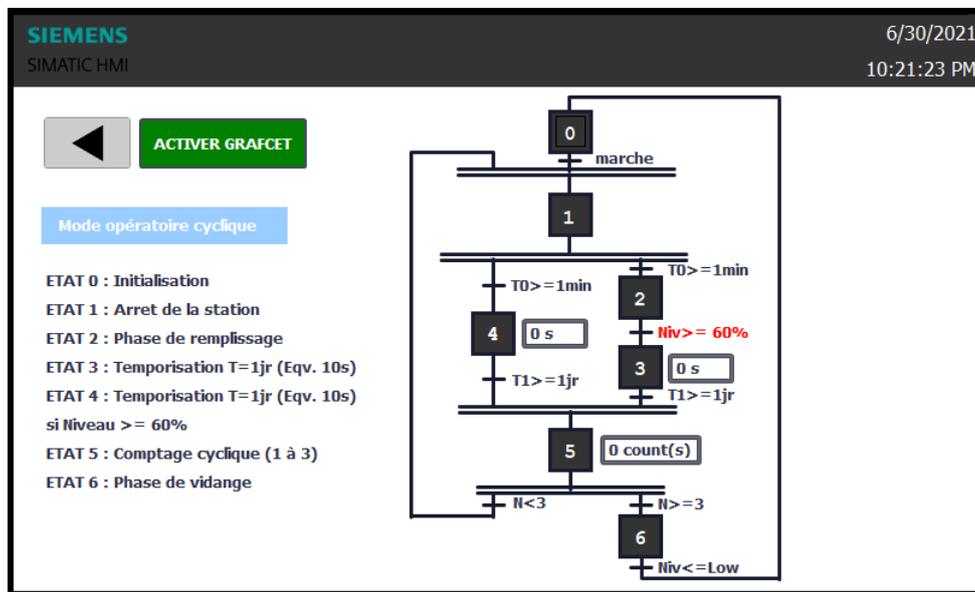


FIGURE 4.27 – Vue GRAFCET II de l’IHM

Vue Contrôle

La vue contrôle permet de superviser graphiquement la courbe du niveau d'eau et de la référence d'entrée. C'est dans cette vue que l'utilisateur peut changer le type de commande ainsi que les paramètres de chaque correcteur. On a pour :

1. La commande PI : K_p , K_i et K_d
2. La commande robuste : K_{SMC}

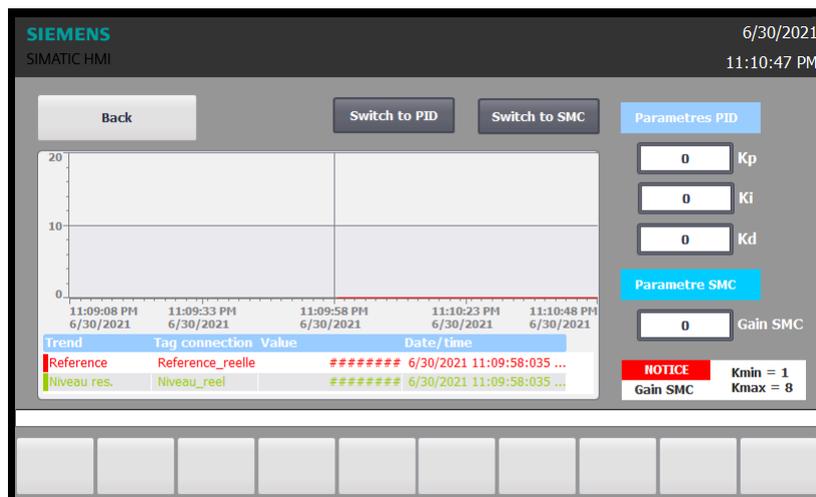


FIGURE 4.28 – Vue Contrôle de l'IHM

Vue Station

C'est la vue où il est possible de lire les valeurs du capteur de niveau, du débitmètre et d'observer les voyants des phases de remplissage, de vidange et de l'arrêt de station.

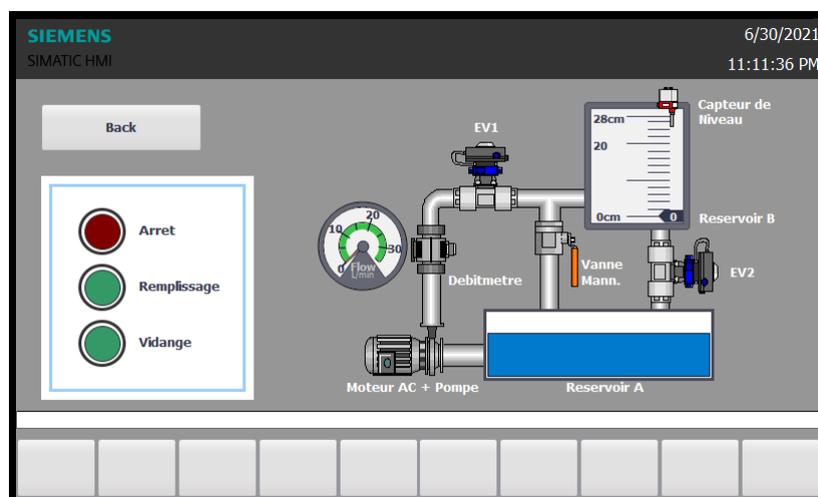


FIGURE 4.29 – Vue Station de l'IHM

Vue Alarmes

La vue alarmes permet d'accéder à toutes les alarmes produite durant le fonctionnement de la machine et permet de faire la gestion de ces dernières. Les alarmes se déclenche lorsque :

1. Alarmes analogiques :
 - Le niveau d'eau atteint la valeur maximale
 - Le niveau d'eau atteint la valeur minimale
 - Le débit d'eau atteint la valeur maximale
 - Le référence dépasse le seuil supérieur
 - Le référence dépasse le seuil inférieur
2. Alarme TOR :
 - Le bouton d'arrêt d'urgence a été déclenché

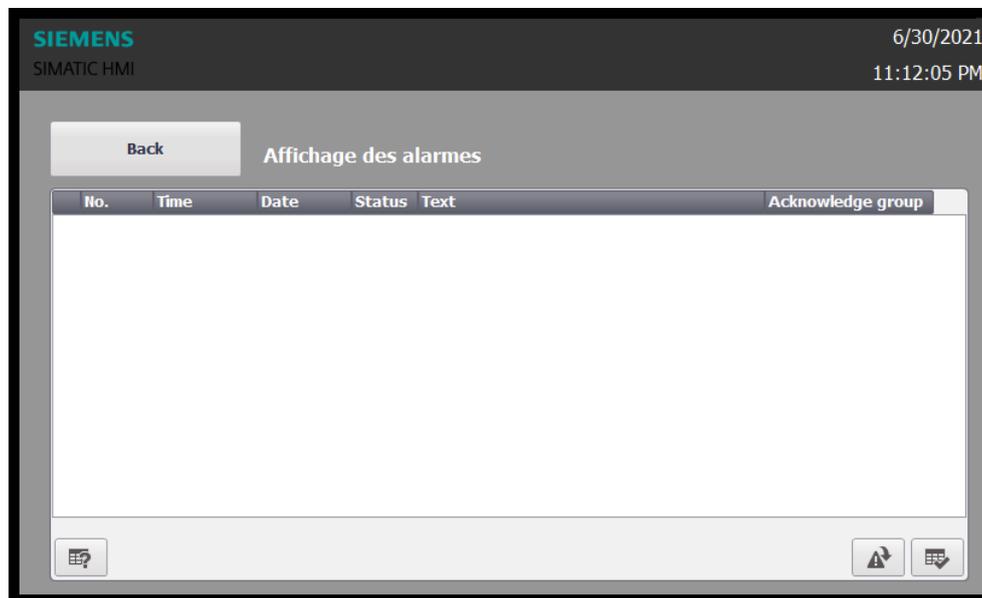


FIGURE 4.30 – Vue Alarmes de l'IHM

Vue recettes

Dans la vue recette, on crée un jeu de paramètres (une structure de données) qui permet de personnaliser un processus en fonction des variables choisies.

Dans notre cas on utilise les recettes dans le deuxième cahier de charges où on définit les paramètres de niveau de seuil du réservoir, la période et le nombre des cycles.

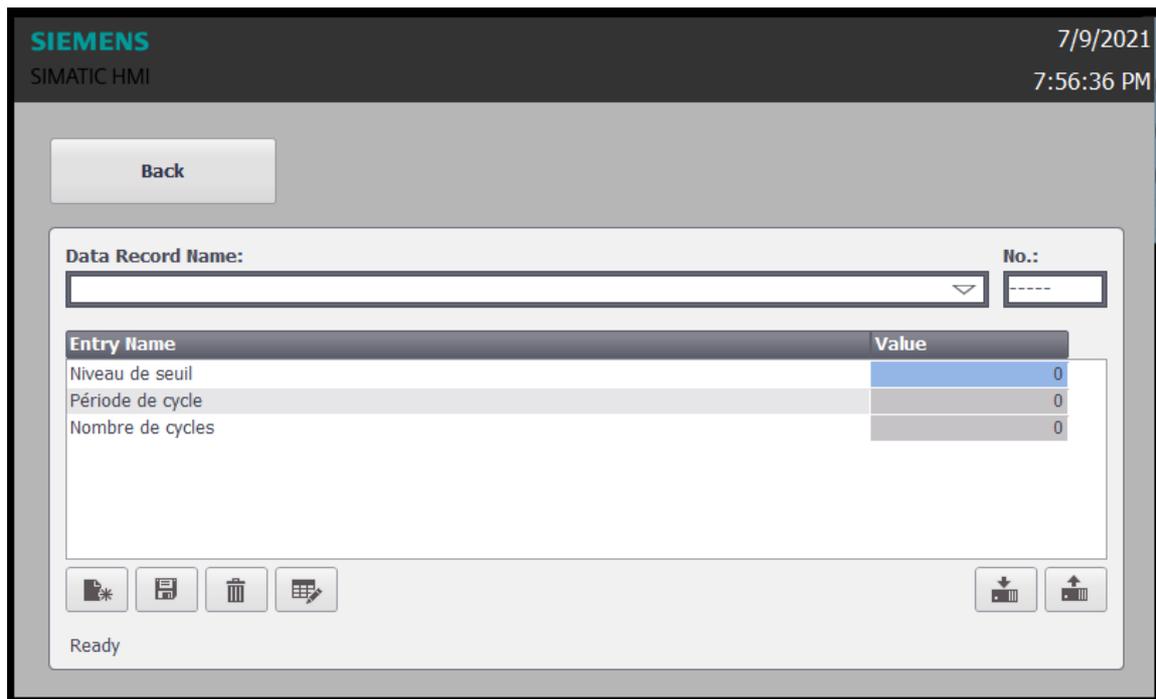


FIGURE 4.31 – Vue recettes

4.3.3 Archivage des données

Sur TIA Portal il est possible d'archiver les données de certaines variables.

Dans notre cas on a archivé d'une façon cyclique dans un fichier .CSV (qui peut s'ouvrir avec Microsoft Excel) les données des variables suivantes :

1. La valeur du gain K_d
2. La valeur du gain K_p
3. Le niveau du réservoir
4. L'état de remplissage
5. L'état de vidange

	A	B	C	D	E
1	VarName	TimeString	VarValue	Validity	Time_ms
2	Remplissage	9/7/2021 11:10	0	1	44386465409
3	Kd	9/7/2021 11:10	1	1	44386465421
4	Ki	9/7/2021 11:10	12	1	44386465421
5	Niveau_reel	9/7/2021 11:10	19		
6	Vidange	9/7/2021 11:10	-1	1	44386465421
7	Remplissage	9/7/2021 11:10	0	1	44386465421
8	Kd	9/7/2021 11:10	1	1	44386465432
9	Ki	9/7/2021 11:10	12	1	44386465432
10	Niveau_reel	9/7/2021 11:10	19		
11	Vidange	9/7/2021 11:10	-1	1	44386465432
12	Remplissage	9/7/2021 11:10	0	1	44386465432
13	Kd	9/7/2021 11:10	1	1	44386465444
14	Ki	9/7/2021 11:10	12	1	44386465444
15	Niveau_reel	9/7/2021 11:10	19		
16	Vidange	9/7/2021 11:10	-1	1	44386465444
17	Remplissage	9/7/2021 11:10	0	1	44386465444
18	Kd	9/7/2021 11:10	1	1	44386465456
19	Ki	9/7/2021 11:10	12	1	44386465456
20	Niveau_reel	9/7/2021 11:10	19		
21	Vidange	9/7/2021 11:10	-1	1	44386465456
22	Remplissage	9/7/2021 11:10	0	1	44386465456

FIGURE 4.32 – Archivage des données

L’option d’archivage offerte par le logiciel TIA Portal engendre plusieurs utilisations et permet d’exploiter le données pour d’autres processus tel que l’analyse et le diagnostic. Dans notre cas, nous avons utilisé l’archivage afin de sauvegarder les données variables du système comme la variation du niveau de réservoir et la consigne et représenter ces dernières par un graphe. Après avoir collecter les données de l’IHM, on traite le fichier sur le logiciel Microsoft Excel afin de faire une représentation graphique de ces dernières. On a décidé de représenter la variation du niveau d’eau suite à une consigne par rapport au temps avec correcteur PI ($K_p = 100$ et $K_i = 15$)

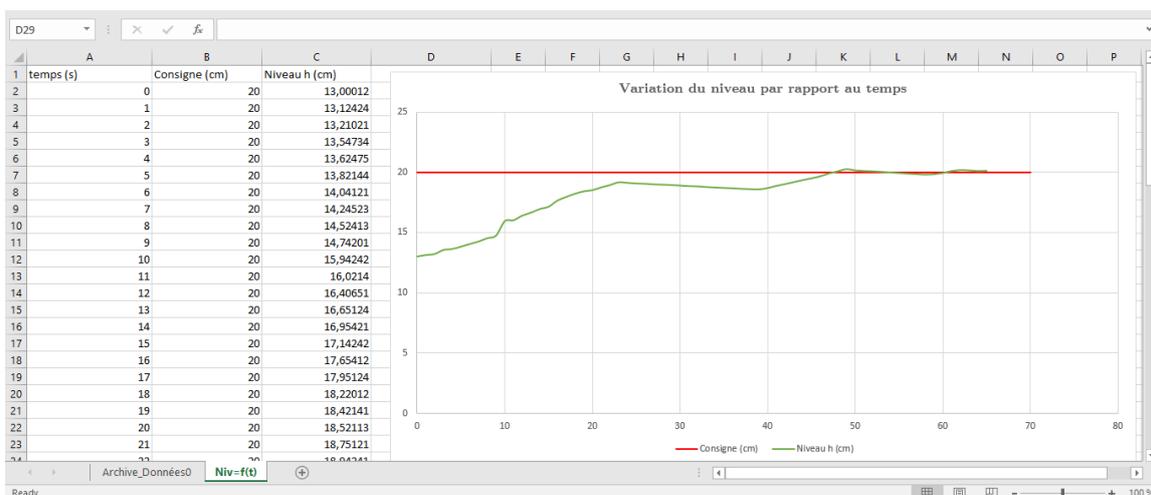


FIGURE 4.33 – Représentation graphique de la variation de niveau d’eau par rapport au temps

4.4 Conclusion

Dans ce chapitre, on a pu voir comment passer d'un cahier de charges élaboré, à un programme de gestion et de supervision. Ce programme permet de superviser à partir de l'IHM, la station de pompage dans son fonctionnement autonome une fois qu'elle commence à marcher.

En raison de la panne au niveau de la pompe et la perte de temps de la réparation de celle ci, les essais pratiques sur la stations n'ont pas été réalisés adéquatement afin de tester nos programmes avec précision. Par contre, la simulation sur TIA Portal nous a permis de voir approximativement le fonctionnement de la station, ce qui a permis de conclure sur la validité de nos programmes. Par conséquent, le programme écrit dans TIA Portal a bien fonctionné .

Conclusion générale

Notre mission à travers ce projet de fin d'études a été de former plusieurs cahiers de charges dans le but d'étendre l'utilisation du prototype de la station de pompage, puisque cette dernière a été assemblée déjà par des étudiants en Automatique de l'École Nationale Polytechnique.

On avait remarqué que sur le plan logiciel, l'API n'a pas été exploité équitablement par rapport au plan matériel, de la station, et c'est pour cette raison qu'on a décidé de consacrer notre travail sur la programmation logicielle afin d'exploiter sérieusement le potentiel de l'automate S7-1200.

À travers les chapitres décrits dans notre étude, nous avons pu réaliser une gestion de la station permettant d'exécuter certaines tâches selon des cahiers de charges qu'on a pu programmer grâce au logiciel TIA Portal. Ce logiciel offre une fiabilité et flexibilité pour le fonctionnement de l'automate S7-1200 et son interface Comfort TP-900.

Par le biais de ce projet, nous avons bénéficié d'un avant-goût sur le monde industriel de l'automatisation des processus, et cela influe sur le choix de nos carrières tant qu'ingénieurs du futur dans l'automatique. On a certainement appris de tas de choses à travers ce projet. On a appris à devenir autonomes, à travailler en dehors de notre zone de confort et à être efficace dans notre mission malgré les circonstances et les imprévus qu'on a eu pendant notre travail.

Dernièrement, on espère que notre travail sera exploité par les étudiants en les incitant à montrer plus de curiosité à ce domaine et à continuer à apprendre de nouvelles connaissances. Notre dernier objectif sera de devenir un bon exemple tant que leurs aînés et de pouvoir se montrer utile avec le travail qu'on a pu réaliser.

5. Annexe A : Initiation à TIA Portal

Logiciels de programmation et d'Interfaces

Un Automate Programmable Industriel, comme indique par son nom, est un dispositif ayant la capacité de lire, d'exécuter et d'afficher un programme écrit dans un logiciel conçu par le même fabricant. Pour notre cas, le constructeur allemand SIEMENS a conçu un écosystème permettant de faire la programmation de plusieurs cahiers de charges et de réaliser les interfaces avec un seul logiciel dédiée à leurs Automates.

Environnement de travail : TIA Portal

Totally Integrated Automation Portal (TIA Portal) donne un accès illimité à la gamme complète de services d'automatisation numérisés, de la planification numérique à l'ingénierie intégrée. Il permet aussi d'augmenter la productivité d'une industrie grâce à des fonctions supplémentaires de diagnostic et de gestion de l'énergie. TIA Portal rassemble à la fois, le logiciel STEP 7 V11 utilisé pour implémenter le cahier de charges comportant les différentes étapes à réaliser pour un système industriel, ainsi qu'un second logiciel WinCC V11 dédié à la création des interfaces intuitives permettant de contrôler et superviser un système industriel.

Programmation des API avec TIA Portal

Création du programme de l'API avec TIA Portal (STEP 7) :

Les étapes qui viennent montreront comment entamer un projet, configurer le matériel en disposition de l'utilisateur, écrire le programme avec les différents langages. Pour cela, nous avons appliqué cette démonstration sur l'exemple suivant :

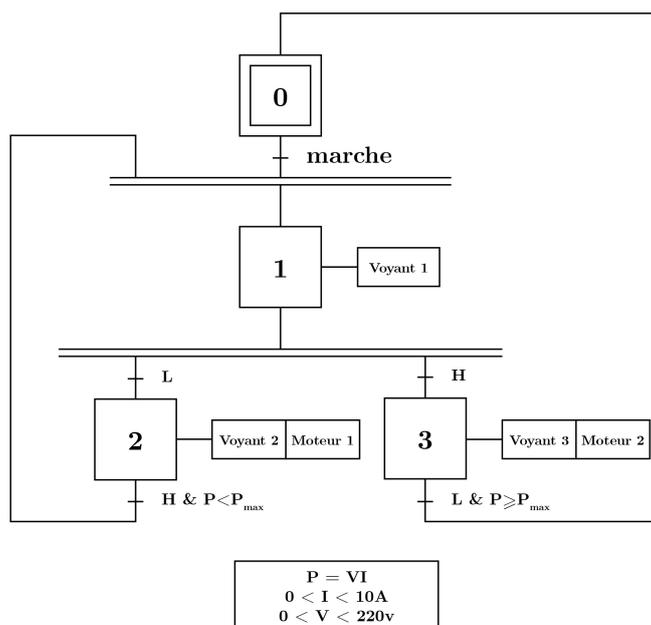


FIGURE 5.1 – GRAFCET de l'exemple du projet

Création du projet :

Après l'installation de TIA Portal, son exécution est accessible à partir du bureau, maintenant cliquer sur l'icône afin de l'ouvrir.

L'ouverture du logiciel affiche la fenêtre suivante :

Nous allons donc montrer en premier lieu comment créer un projet sur TIA Portal. Cette partie est composée de plusieurs points qu'on va décrire avec des étapes très concises.

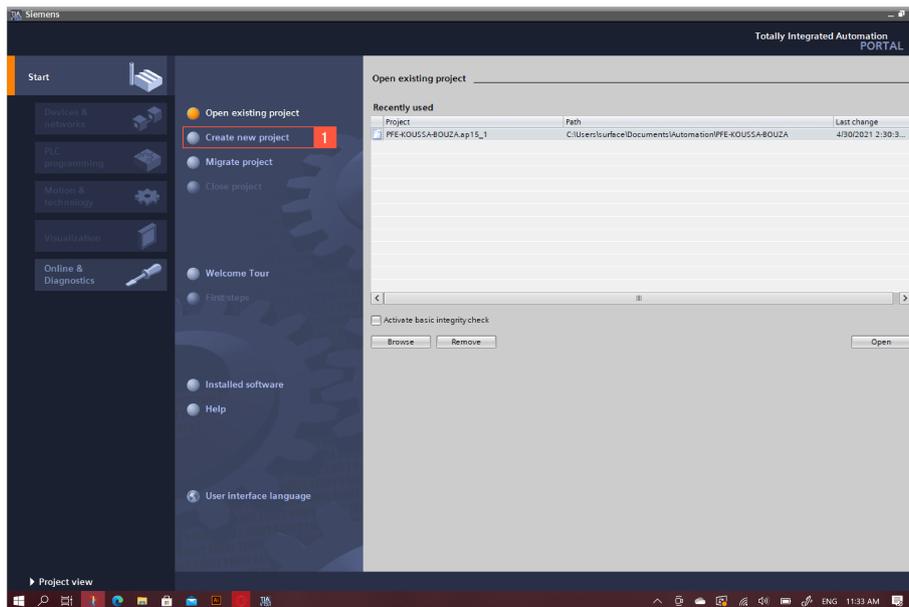


FIGURE 5.2 – Partie 1

1. Cliquer sur créer un nouveau projet.

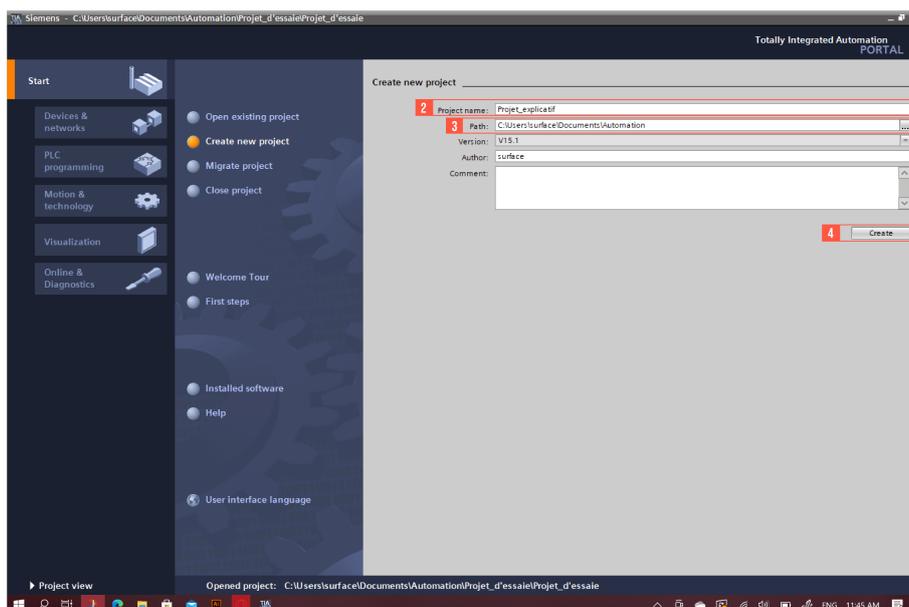


FIGURE 5.3 – Partie 1 (suite)

2. Affecter au projet créé un nom convenable.
3. Choisir le chemin de sauvegarde des fichiers.
4. Cliquer sur créer.

Configuration matérielle :

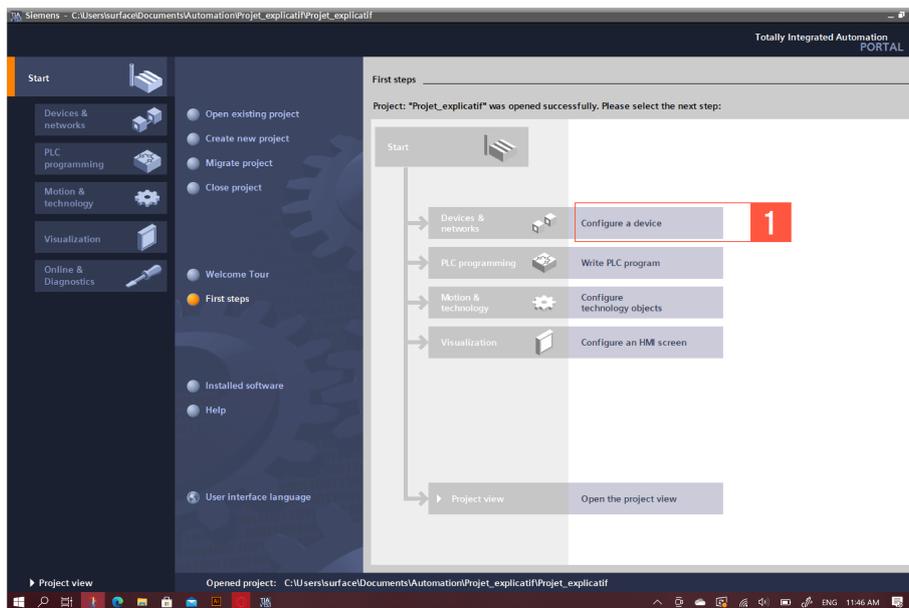


FIGURE 5.4 – Partie 2

1. Cliquer sur “configure a device” ou en français, configurer un appareil afin de lancer la configuration matérielle de l’automate

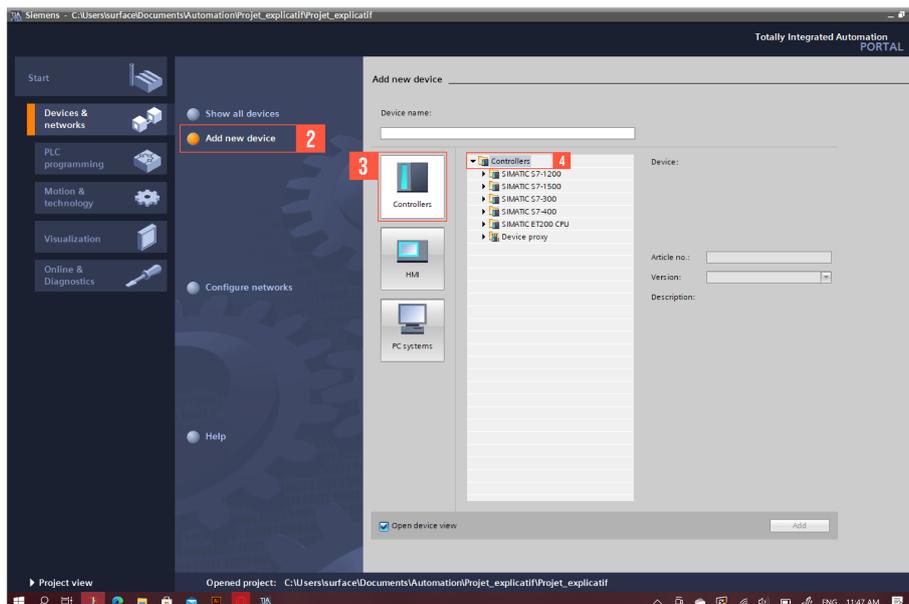


FIGURE 5.5 – Partie 2 (suite)

2. Cliquer sur ajouter un nouvel appareil.
3. Cliquer sur l'icône indiquant la CPU.
4. Puis cliquer sur "Controllers" ou contrôleurs en français afin de choisir la CPU.

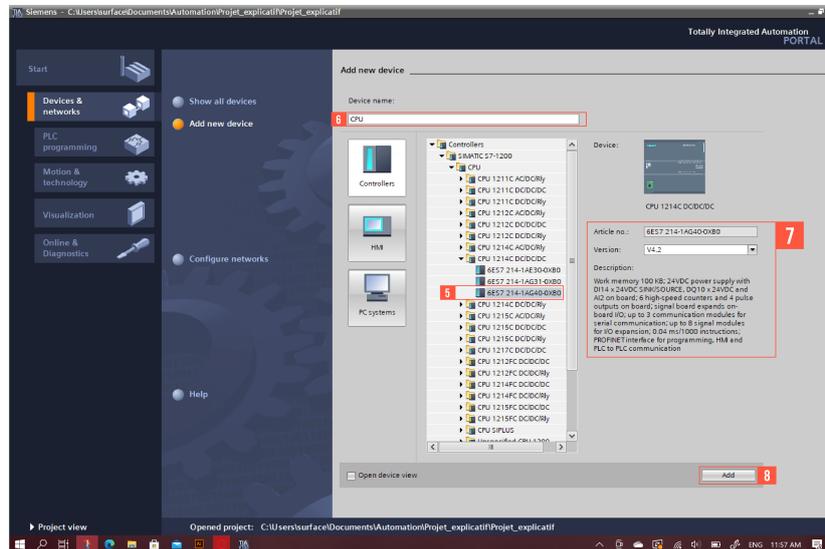


FIGURE 5.6 – Partie 2 (suite)

5. La CPU à choisir dépend du matériel physique disponible, pour cet exemple on a choisi la S7-1200 1214C DC/CD/DC.
6. Affecter un nom à la CPU.
7. Choisir la version du programme et lire les informations concernant l'architecture de la CPU.
8. Ajouter le matériel choisi.

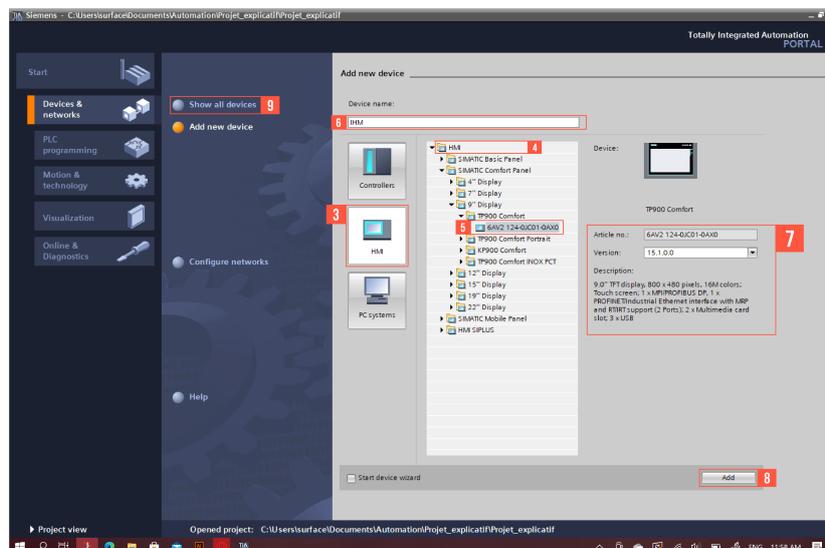


FIGURE 5.7 – Partie 2 (suite)

9. Refaire les mêmes étapes (de 3 à 8) afin de choisir l'IHM, dans cet exemple on a pris l'interface Confort TP900 6AV 124-0JC01-0AX0. Puis cliquer sur "Show all devices" ou afficher tous les appareils en Français

Configuration des modules d'E/S :

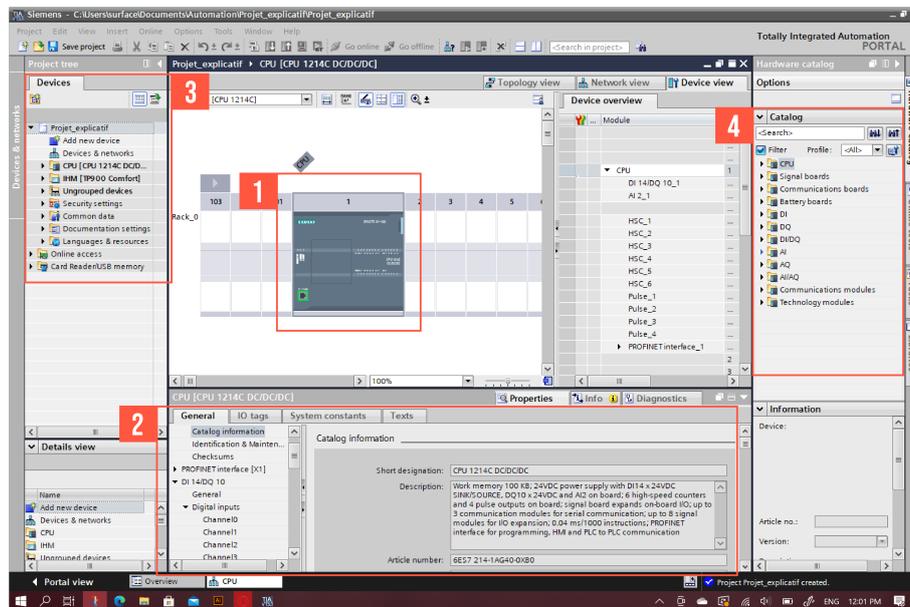


FIGURE 5.8 – Partie 3

1. Cliquer sur la CPU afin de voir plus de détails.
2. Cette case montre les différents détails et informations sur l'architecture de la CPU, son alimentation, ses entrées et sorties et leurs adresses, sa capacité de mémoire, modules de communications...etc.
3. La 3^{me} case permet d'avoir une vue générale sur le matériel choisi et raccourcit le déplacement entre chaque répertoire.
4. Cette case montre le catalogue complet dédié à cette CPU incluant les modules d'entrées et de sorties digitales ou analogiques, les modules de communication et blocs d'alimentation, les batteries à rajouter...etc.

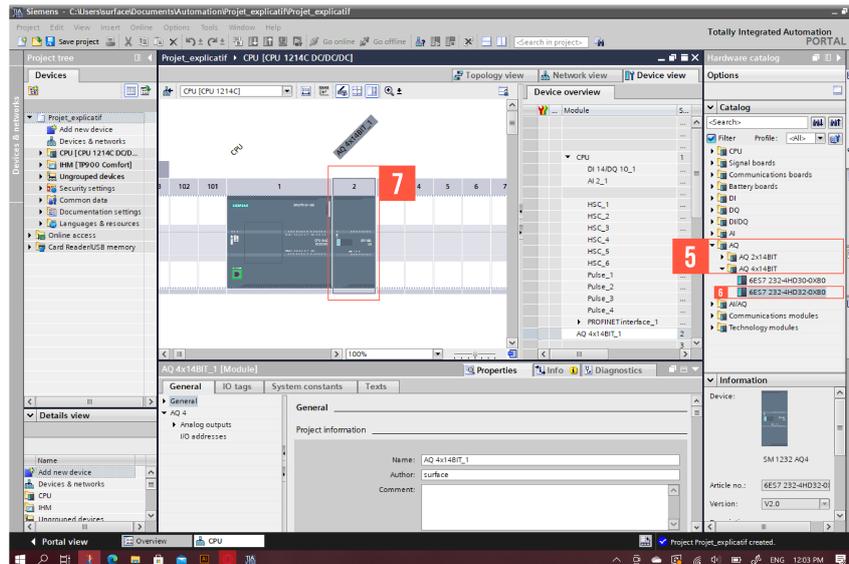


FIGURE 5.9 – Partie 3 (suite)

5. Dans cet exemple, il est nécessaire de rajouter un module de sorties analogiques. Dans le catalogue, cliquer sur AQ, puis sur AQ 4x14BIT.
6. Choisir le module 6ES7 232-4HD32-0XB0 (selon le besoin).
7. La case 7 montre le module analogique rajouté au rack, cliquer dessus pour voir plus de détails en bas.

Adressage des mnémoniques :

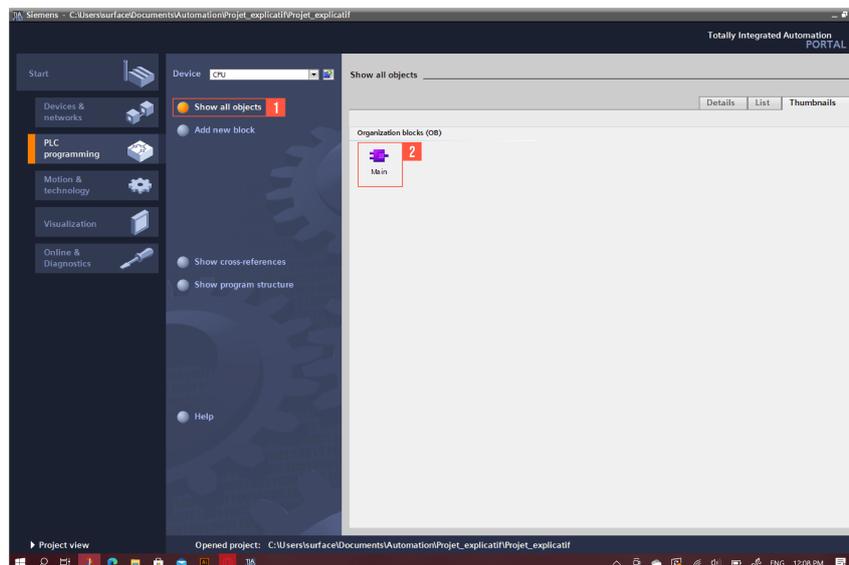


FIGURE 5.10 – Partie 4

1. Revenir au portail principal, cliquer sur “PLC Programming” ou en français Programmation de l’Automate puis “Show all objects” (Afficher tous les objets en français).

2. Double cliquer sur le bloc MAIN.

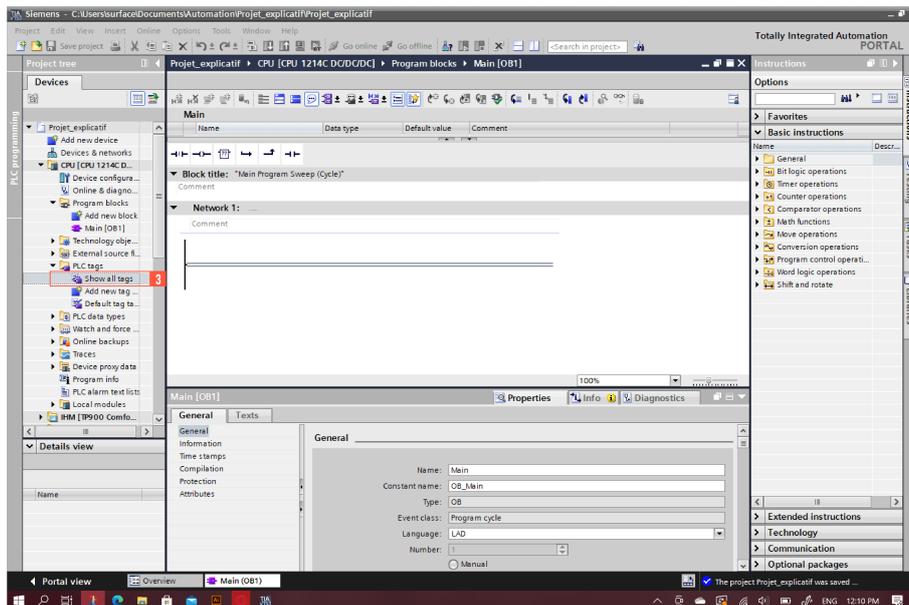


FIGURE 5.11 – Partie 4 (suite)

3. Cliquer sur “PLC tags” puis sur “Show all tags” afin de passer à la configuration des mnémoniques.

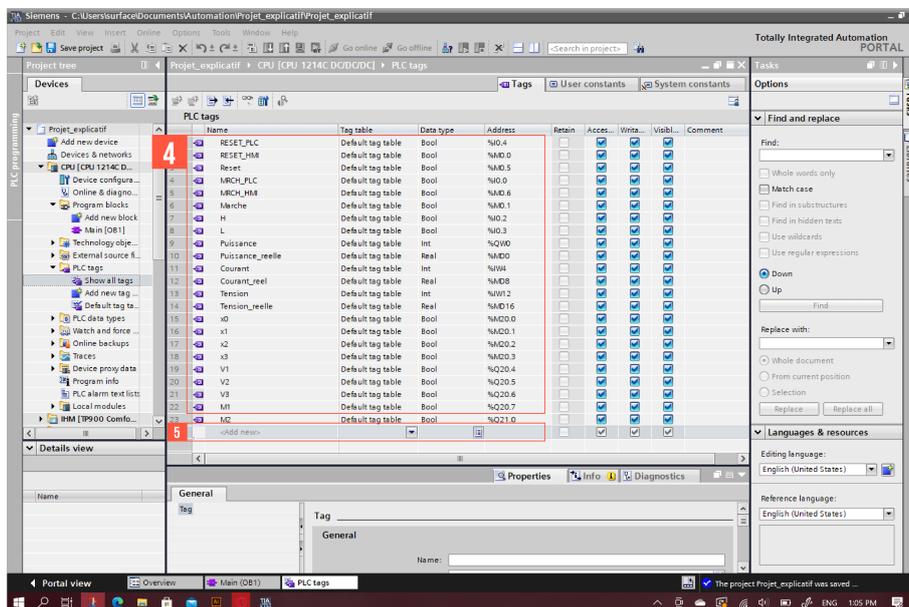


FIGURE 5.12 – Partie 4 (suite)

4. La case 4 montre toutes les variables adressées avec leurs noms, type et adresse dans la mémoire de la CPU.
5. Cliquer sur cette case afin de rajouter de nouvelles variables ou supprimer après avoir rajouter. Après chaque modification, il est nécessaire de sauvegarder la progression afin d'éviter des anomalies lors de la phase de programmation.

Traitement des variables analogiques :

Le traitement des variables analogiques est indispensable lorsqu'on écrit un programme comportant des variables analogiques sur STEP 7. Sachant que les variables analogiques représentent des grandeurs physiques tel que la puissance électrique, la pression ou la température. En réalité ce que l'automate reçoit au niveau de ses entrées analogiques, est une grandeur électrique délivrée par un capteur, soit une tension ou bien un courant selon sa configuration. Dans notre cas, l'automate est configuré de la sorte qu'il y ait seulement une tension continue allant de 0 à 10 V à l'entrée. La partie suivante, montre comment traiter les variables analogiques sous TIA Portal.

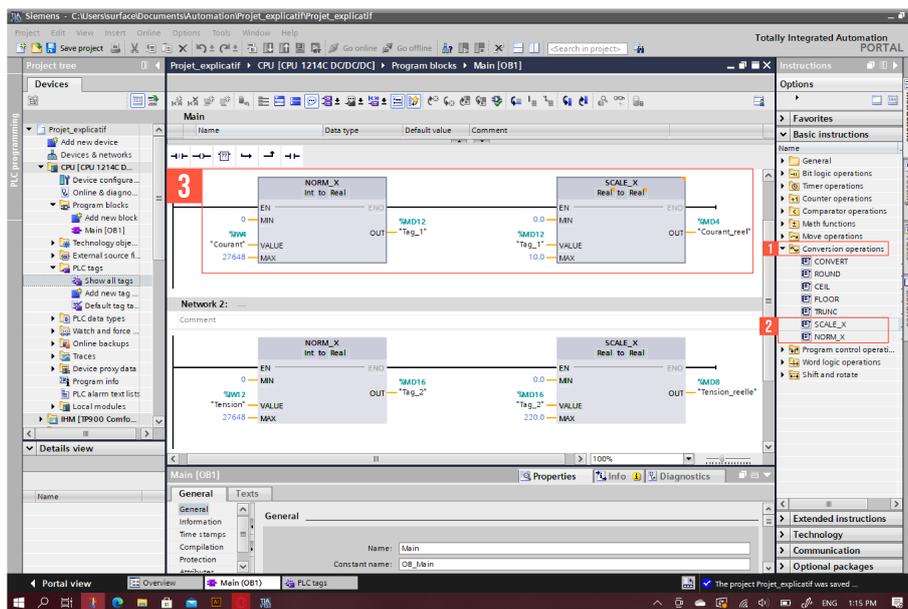


FIGURE 5.13 – Partie 5

1. Dans le répertoire des instructions basiques, cliquer sur Opérations de conversion .
2. Double clique sur les deux blocs NORM_X et SCALE_X respectivement
3. La case suivante montre les deux blocs dans le réseau. Le premier bloc fait la normalisation de la variable entière introduite, cette variable est l'image du signal délivré par un capteur. Son but est de normaliser cette variable en la divisant sur un nombre spécifique (27648). Le résultat de la normalisation est un nombre à virgule flottante, c'est à dire un réel compris entre 0 et 1. Le second bloc permet de faire la mise à l'échelle du réel précédent en le multipliant par des valeurs que l'utilisateur introduit (ici 0 et 10), ce qui donne un sens physique à la variable introduite. Après la normalisation et la mise à l'échelle, le 0 correspond à 0 Ampères et le 1 correspond à 10 Ampères.

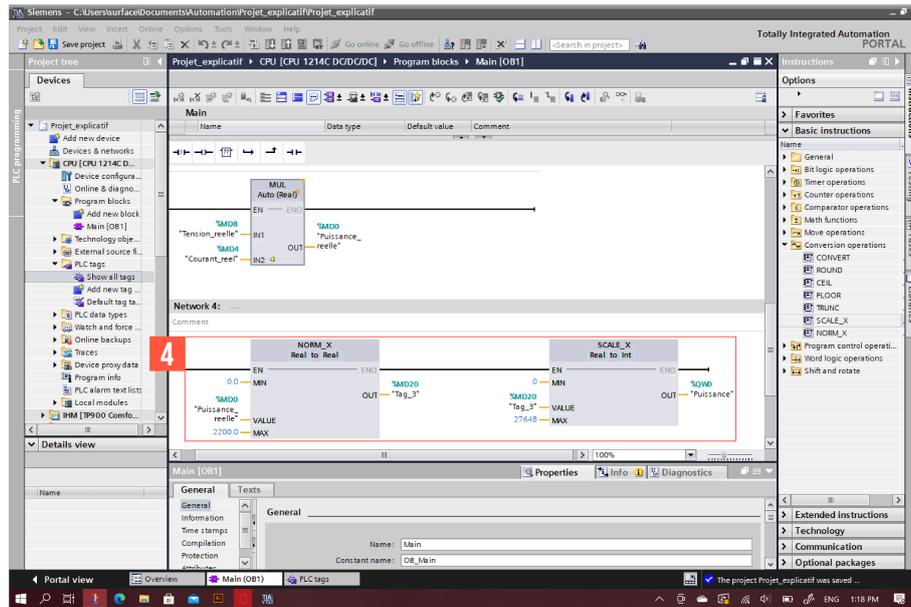


FIGURE 5.14 – Partie 5 (suite)

4. Après avoir effectué les différentes opérations sur les variables analogiques, il est donc nécessaire de traiter la variable analogique de sortie. Comma par avant, la même étape va se répéter ici (voir étape précédente) sauf que dans ce cas, la valeur introduite dans le bloc NORME_X est le résultat effectué sur les autres variables réelles qui va être ensuite normaliser par sa valeur maximale (ici la puissance réelle, normalisée par sa valeur max 2200 W). Ceci donne un nombre réel à virgule flottante compris entre 0 et 1. Ce nombre est introduit dans le bloc suivant et sera mis à l'échelle entière de 0 à 27648 et le résultat final est une grandeur de type entier.

Noter que cette étape est longue et répétitive, il est donc recommandé d'établir une fonction afin de remédier à ce problème. La partie création de fonction est traitée individuellement par la suite.

Création de Fonctions sous TIA Portal :

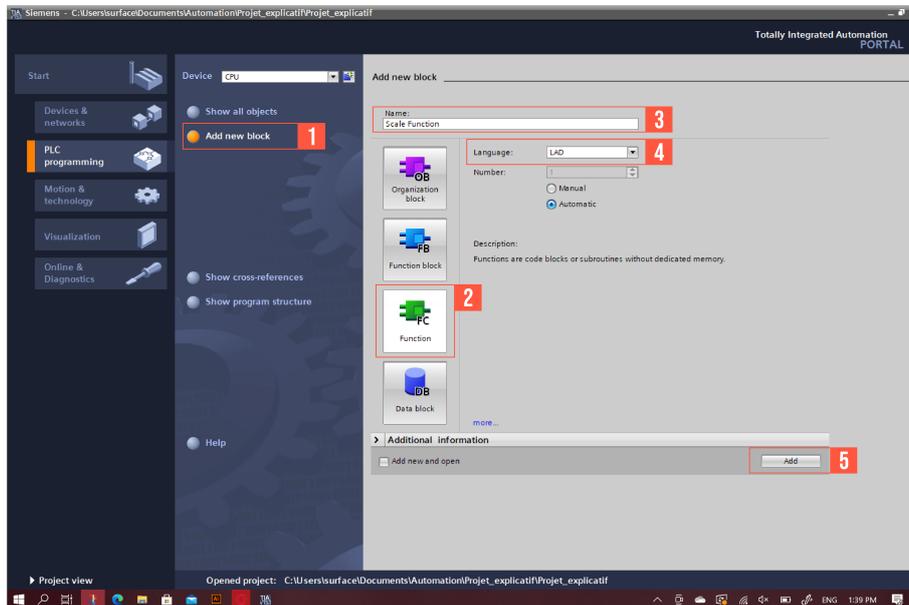


FIGURE 5.15 – Partie 6

1. Revenir au portail principal et cliquer sur “Ajouter un nouveau bloc”.
2. Choisir le bloc “Fonction FC”.
3. Affecter un nom à la fonction.
4. Choisir le langage de programmation.
5. Ajouter le bloc.

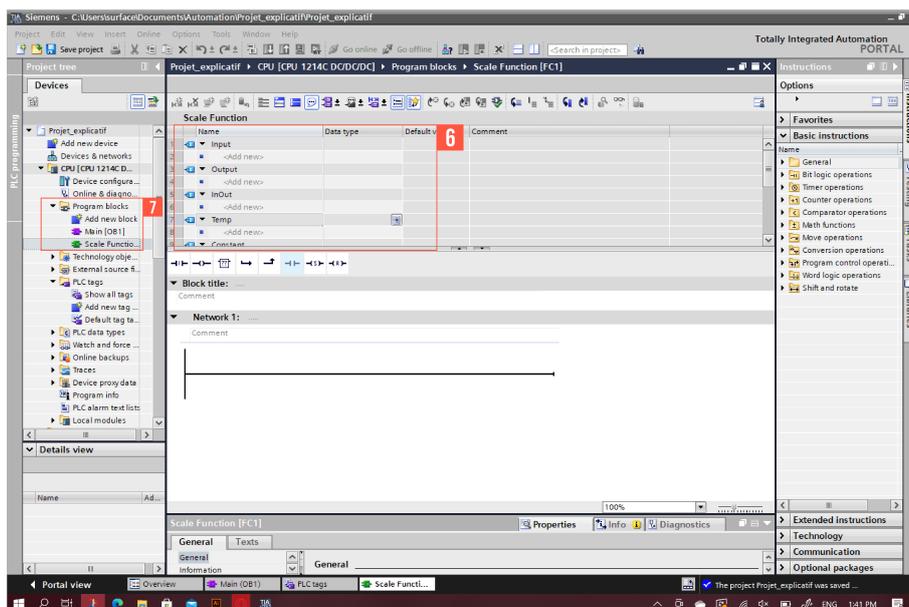


FIGURE 5.16 – Partie 6 (suite)

6. Cette case représente les variables locales propres à la fonction créée. Il y a les entrées, les sorties, les entrées/sorties, les constantes et les variables temporaires.
7. La case suivante montre les différents blocs créés par l'utilisateur (noter qu'il y pas de différence flagrante entre le bloc MAIN et les blocs FC, la seule différence est dans les variables locale du bloc FC qui sont accessible qu'à l'intérieur).

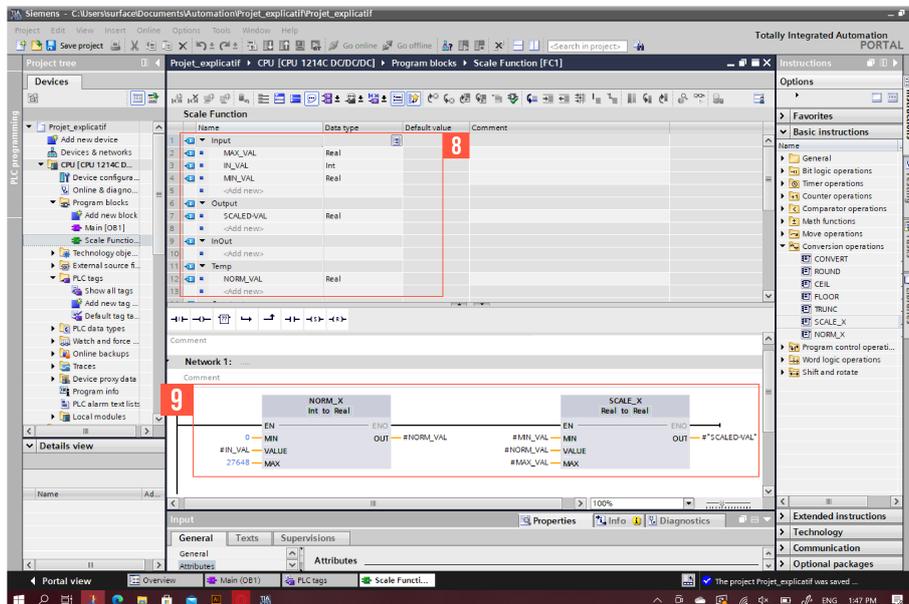


FIGURE 5.17 – Partie 6 (suite)

8. Créer les variables locales utilisées par la fonction.
9. Faire le programme de la fonction (ici c'est la fonction "Scale" qui fait le traitement des variables analogique).

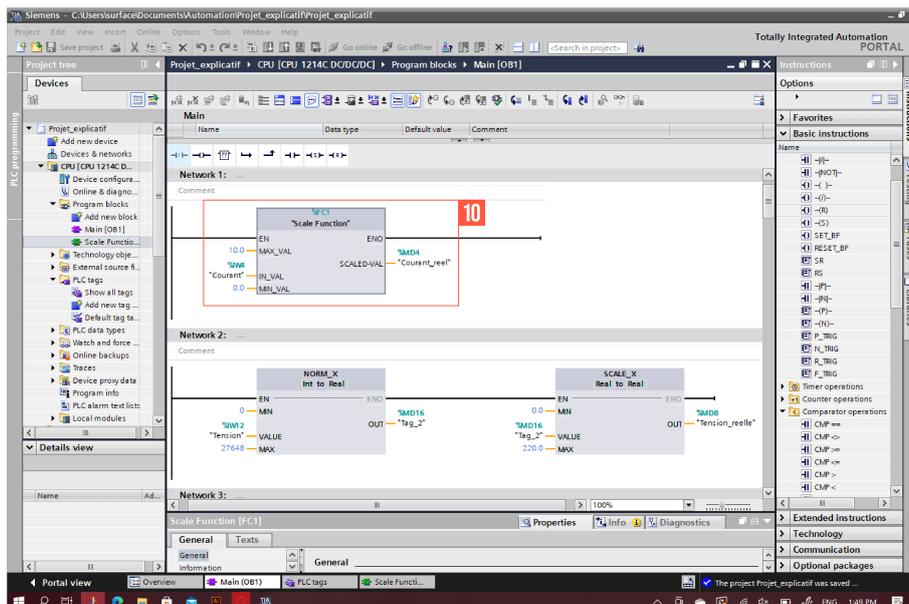


FIGURE 5.18 – Partie 6 (suite)

- Revenir au bloc MAIN après avoir enregistré et cliquer sur la fonction créée. La fonction est affichée en bloc et prend comme arguments la variable d'entrée, le maximum et le minimum de l'échelle et retourne la sortie réelle mise en échelle. Noter que la fonction ne précise pas le type des arguments, il est nécessaire de rester conscient lors de l'affectation des variables et de leur type.

Écrire le programme :

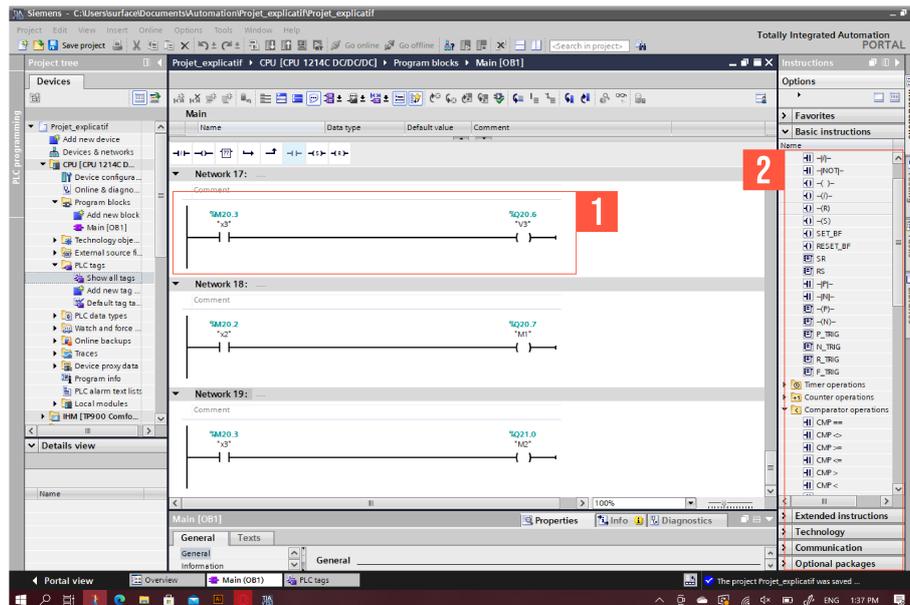


FIGURE 5.19 – Partie 7

- L'écriture du programme se fait par des blocs montés sur des réseaux ou par des lignes de code selon le langage choisi (ici LADDER). Chaque réseau représente une étape ou une équation obtenue à partir du GRAFCET d'un processus industriel.
- La case 2 représente les différentes instructions utiles afin d'écrire le programme, il y a celles dédiées à la conversion de type, la comparaison, les opérations arithmétiques, les mises à l'échelle...etc. Utiliser les instructions selon les équations du GRAFCET sur les réseaux afin de réaliser le programme.

Simulation de l'API :

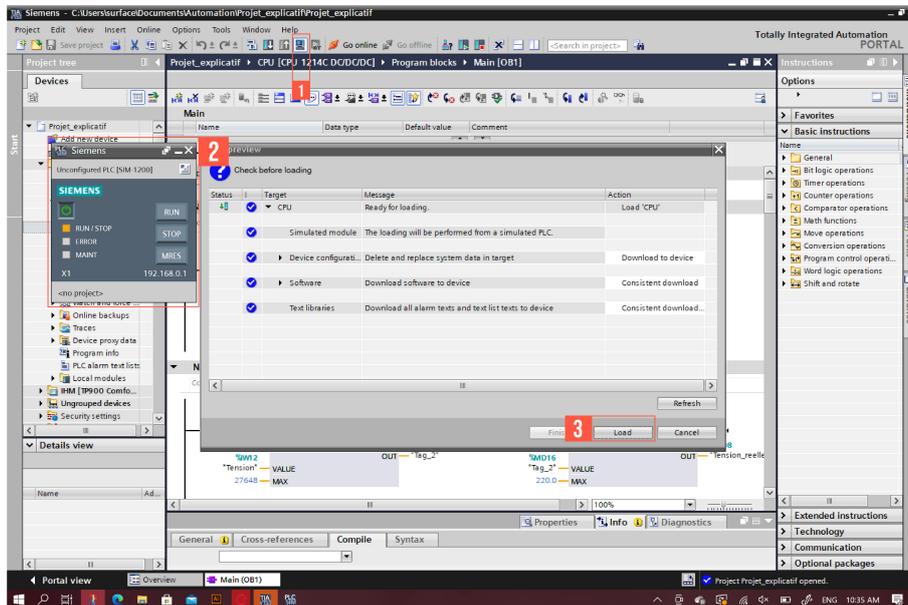


FIGURE 5.20 – Partie 8

1. Cliquer sur l'icône afin de lancer la simulation du programme.
2. Simulateur de l'automate chargé de lancer le programme, le simuler et l'arrêter à tout moment.
3. Cliquer sur "Load" ou charger en français afin de charger le programme.



FIGURE 5.21 – Partie 8 (suite)

4. L'état de la CPU (En marche, arrêt, erreur...etc.).
5. Cliquer sur RUN afin de mettre la CPU en marche.
6. Cliquer sur l'icône pour étendre la fenêtre de simulation.

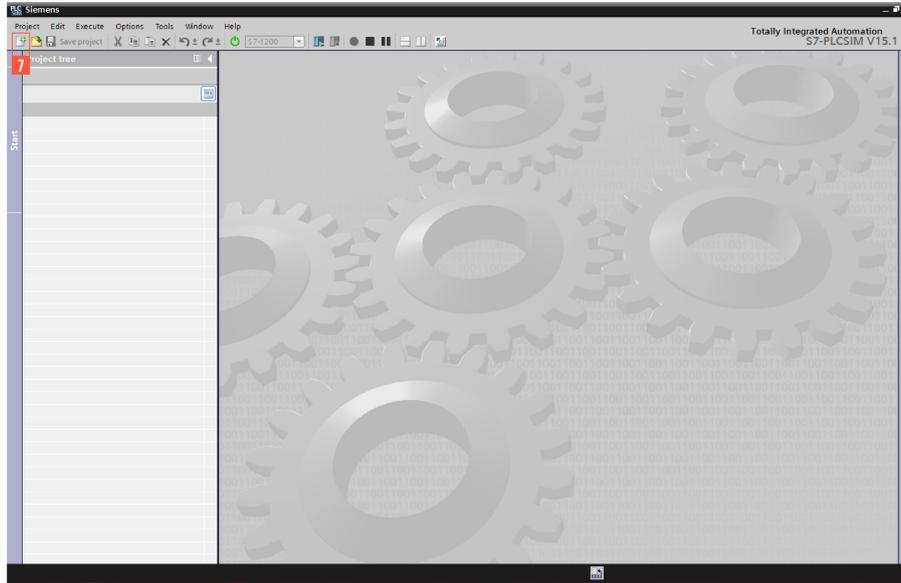


FIGURE 5.22 – Partie 8 (suite)

7. Créer un nouveau projet.

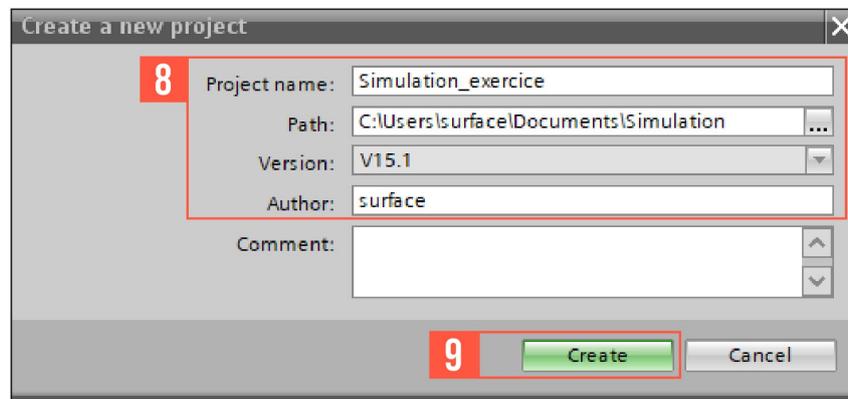


FIGURE 5.23 – Partie 8 (suite)

8. Choisir le nom du projet, son chemin. PLCSIM est un logiciel de simulation sous la même version que TIA Portal.
9. Créer le projet.

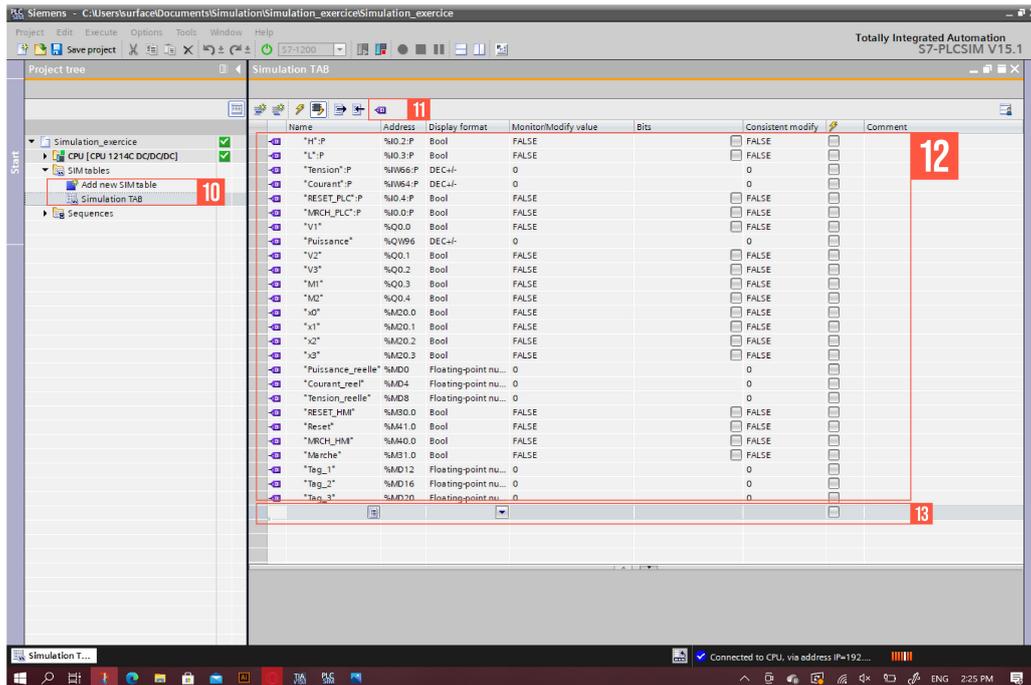


FIGURE 5.24 – Partie 8 (suite)

10. Créer une table de mnémoniques afin de pouvoir modifier les valeurs des variables.
11. Chargé la table avec les mnémoniques du programme à simuler en cliquant sur l'icône.
12. Les variables présentes dans le programme avec leurs adresses et type. Il est possible de modifier les valeurs ou l'état de ces variables au niveau de la colonne "Monitor/Modify value" et de superviser la totalité du programme.
13. Rajouter une variable dans la table.

Création de l'IHM avec TIA Portal (WinCC)

Cette partie consiste à créer l'Interface Homme/Machine (IHM) avec le système de contrôle et d'acquisition des données (SCADA) WinCC sous l'environnement de TIA Portal.

Création de la liaison API-IHM et la table des mnémoniques :

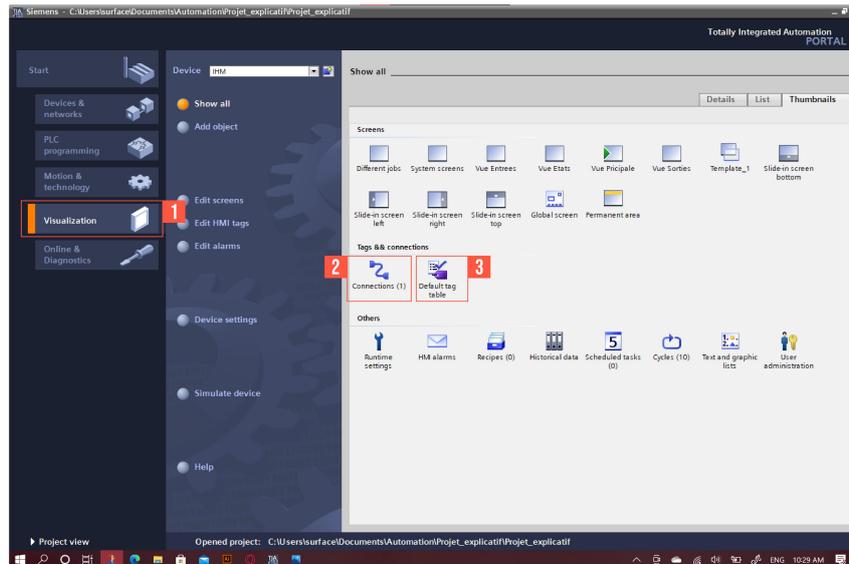


FIGURE 5.25 – Partie 1

1. Revenir à la vue portail et cliquer sur “Visualisation” ou visualisation en français.
2. Cliquer sur “Connections” afin de créer la liaison de communication entre la CPU et l’IHM.
3. Cliquer sur l’icône afin de créer la table des mnémoniques de l’IHM.

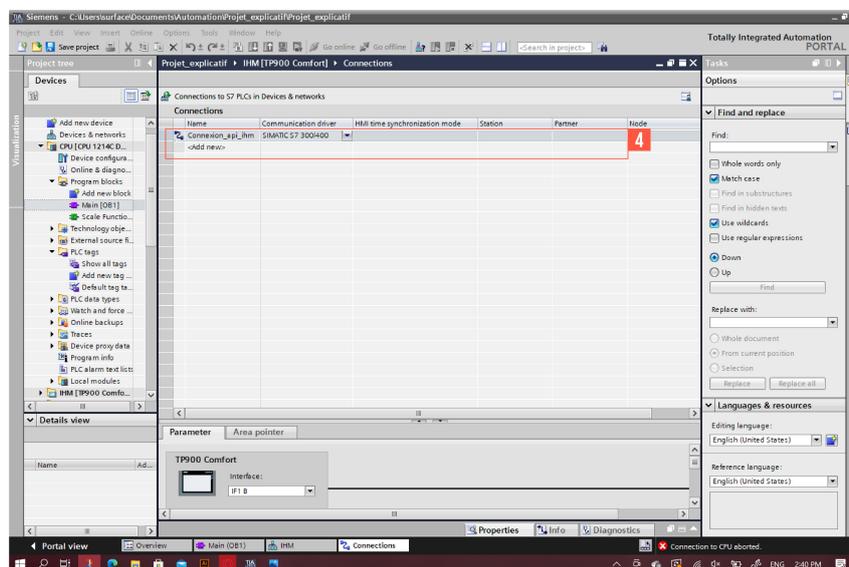


FIGURE 5.26 – Partie 1 (suite)

4. Ajouter une liaison en lui choisissant un nom. Il faut indiquer la CPU correspondante à celle choisie pour le programme afin de pouvoir établir une connexion entre l’IHM et cette dernière.

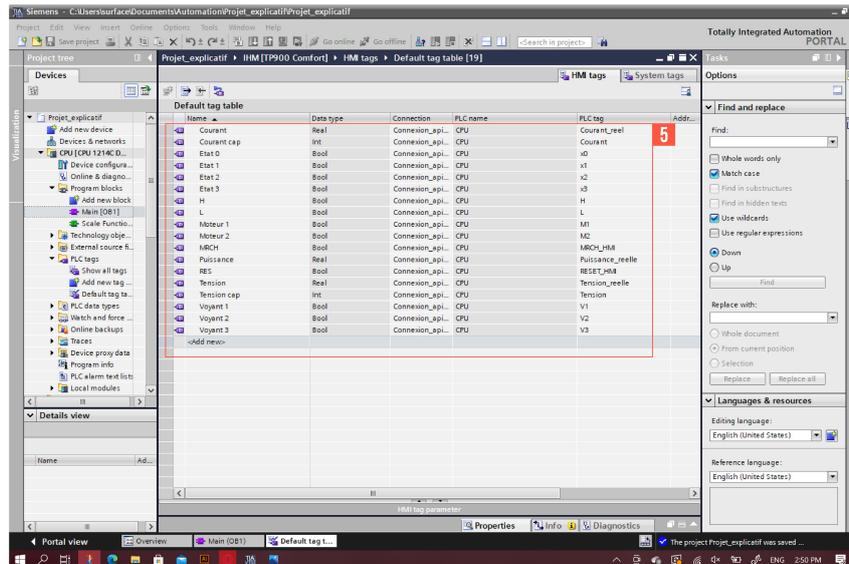


FIGURE 5.27 – Partie 1 (suite)

- Écrire toutes les variables qui seront utilisées dans l'IHM et les faire correspondre à leurs variables équivalentes dans la CPU.

Création des vues :

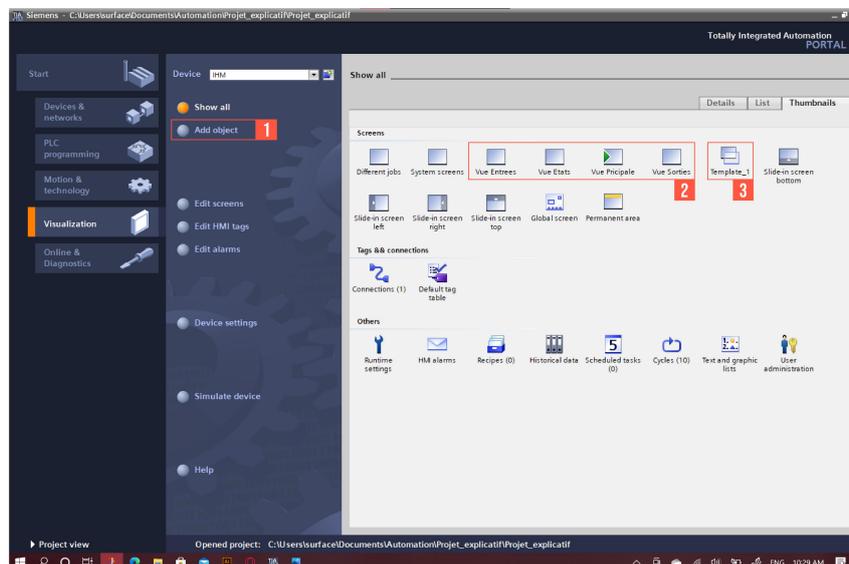


FIGURE 5.28 – Partie 2

- Revenir à la vue portail et cliquer sur “Add object” ou ajouter un objet en français et créer toutes les vues nécessaires dans l'IHM.
- Les vues créées sont affichées dans la case 2 avec leur nom choisit.
- Cette case représente le “Template” ou bien le modèle en français. C’est le Design (Palette de couleurs, arrière-plan, disposition des boutons...etc.) que toutes les vues prennent.

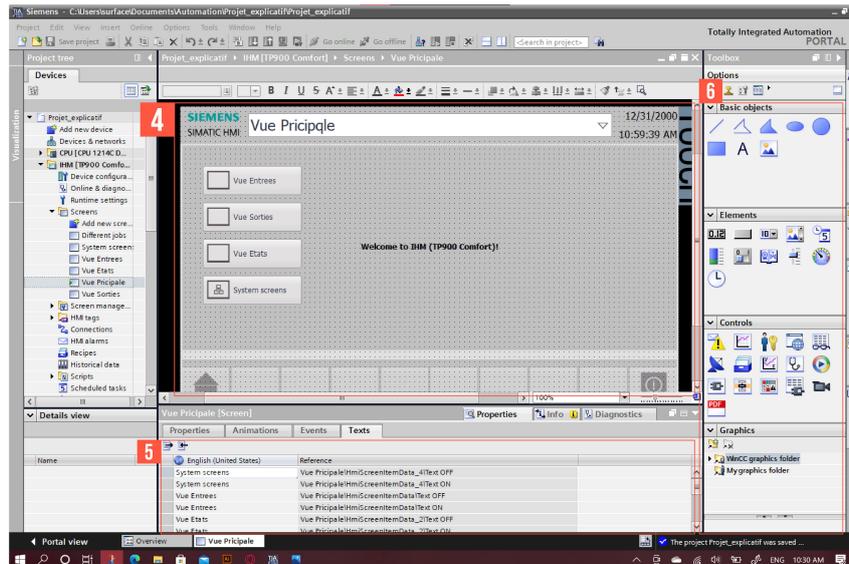


FIGURE 5.29 – Partie 2 (suite)

4. Vue principale de l'IHM.
5. Possibilité de modification des propriétés d'une vue (Les couleurs, la police de caractères, la taille...etc.).
6. "Toolbox" ou boîte à outils en français, c'est les outils à utiliser afin de contrôler, modifier et superviser certaines variables.

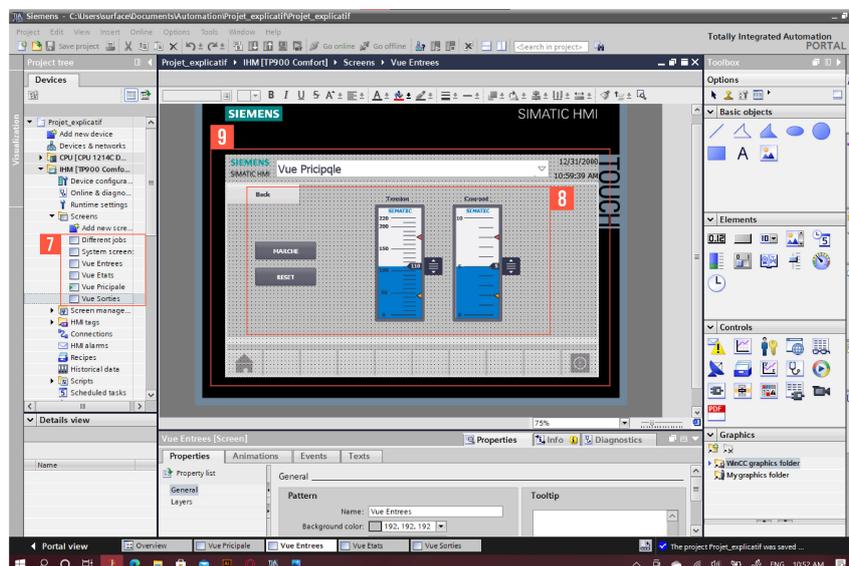


FIGURE 5.30 – Partie 2 (suite)

7. Naviguer sur le répertoire des vues afin de choisir une dans le but de modifier son contenu.
8. Des outils utilisés dans cette vue (2 réglettes pour varier une grandeur analogique, deux boutons d'activation).
9. Vue d'Entrées.

Gestion des alarmes TOR et analogiques :

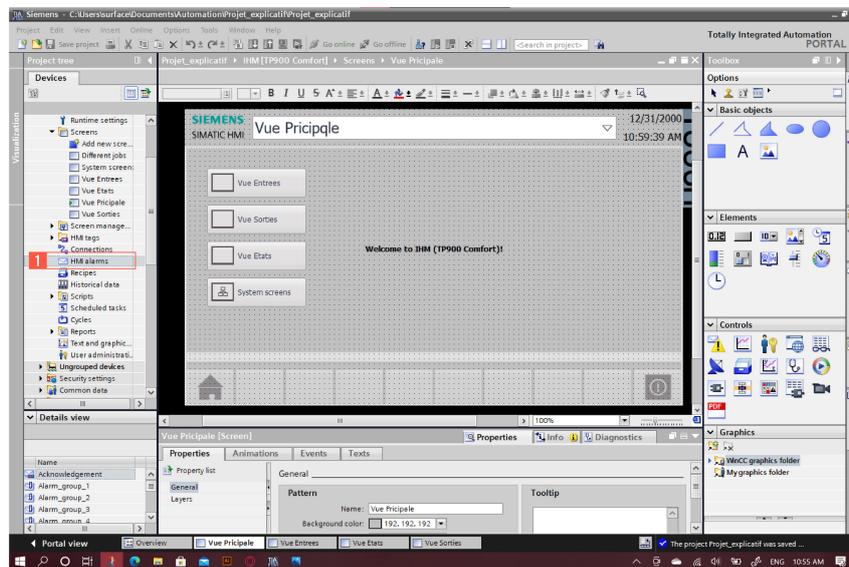


FIGURE 5.31 – Partie 3

1. Cliquer sur “HMI Alarmes” ou en Français alarmes IHM afin d’accéder à la configuration des alarmes. En général, il existe 2 types d’alarmes, les alarmes TOR (Tout Ou Rien) qui s’activent lorsqu’une variable discrète présente une alerte ou une erreur (selon sa classe) ou simplement lorsqu’elle change d’état (de 0 au 1 logique). Il y a aussi les alarmes analogiques qui s’activent lorsque la variable analogique dépasse un seuil défini. Toutes alarmes sont caractérisées par un identifiant, un nom, un message et une classe (Alerte, erreur, danger...etc.).

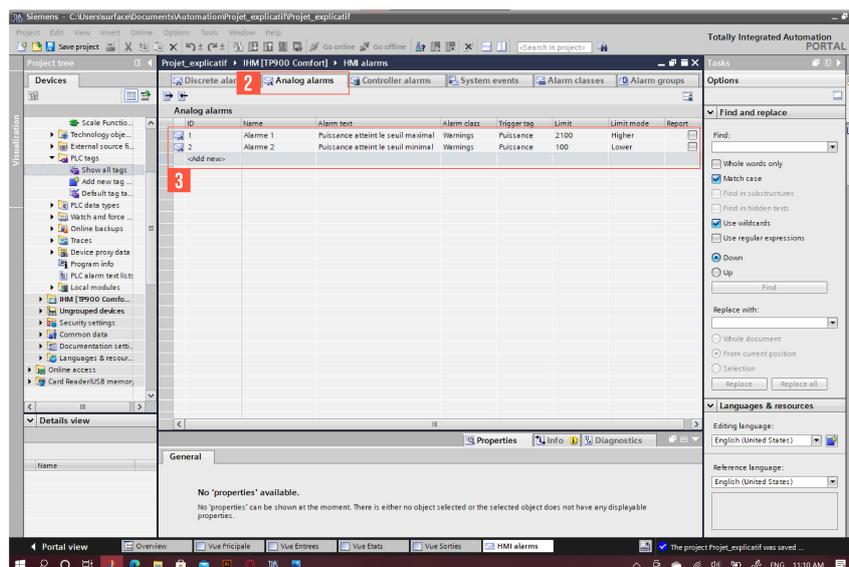


FIGURE 5.32 – Partie 3 (suite)

2. Cliquer sur “Analog alarms” ou alarmes analogiques.
3. Deux alarmes ont été créées dans cet exemple, une alerte qui s’active lorsque la puissance dépasse 2100 W et une autre lorsque la puissance est en dessous de 100 W.

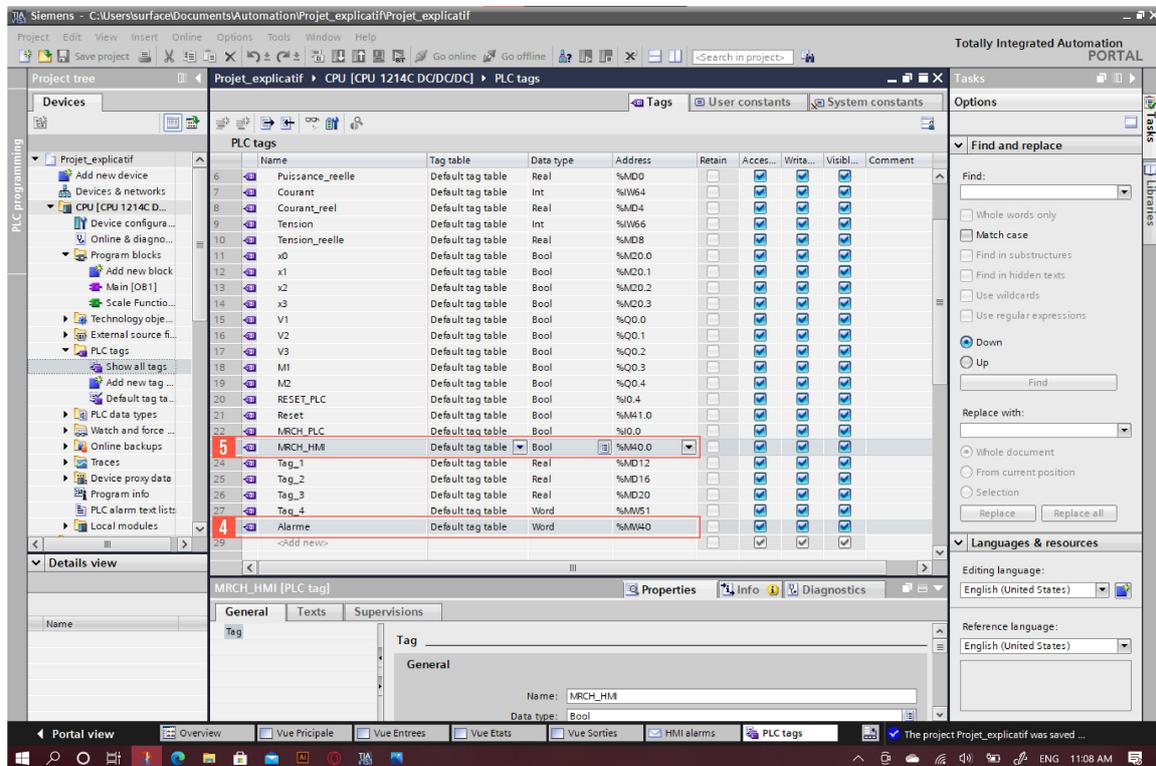


FIGURE 5.33 – Partie 3 (suite)

4. Pour créer une alarme TOR, il est nécessaire de revenir à la table des mnémoniques de la CPU. Créer une variable de type INT et lui attribuer un nom et une case mémoire exclusive à elle seulement.
5. L’adresse de la variable discrète qui active l’alarme doit se trouver dans la même case mémoire attribué à l’alarme TOR (ici, l’adresse de la variable alarme est MW40 et l’adresse de la variable “marche” qui active cette alarme est M40.0). Cela est nécessaire car pour mettre en œuvre une alarme TOR, l’API a besoin d’une case mémoire de 2 octets (une variable entière à 8 bits) contenant jusqu’à 16 adresses dédiées au lancement de l’alarme (soit de MX.0 à MX.7 et de MY.0 jusqu’à MY.7).

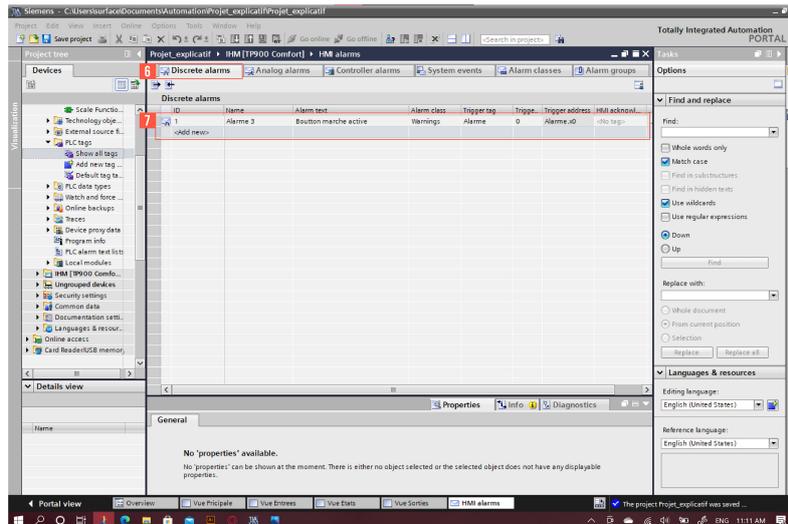


FIGURE 5.34 – Partie 3 (suite)

6. Revenir à la fenêtre des alarmes et cliquer sur “Discrete alarms”.
7. Ajouter une alarme TOR (Il faut mentionner l’adresse du bit d’activation, dans cet exemple c’est le bit 0).

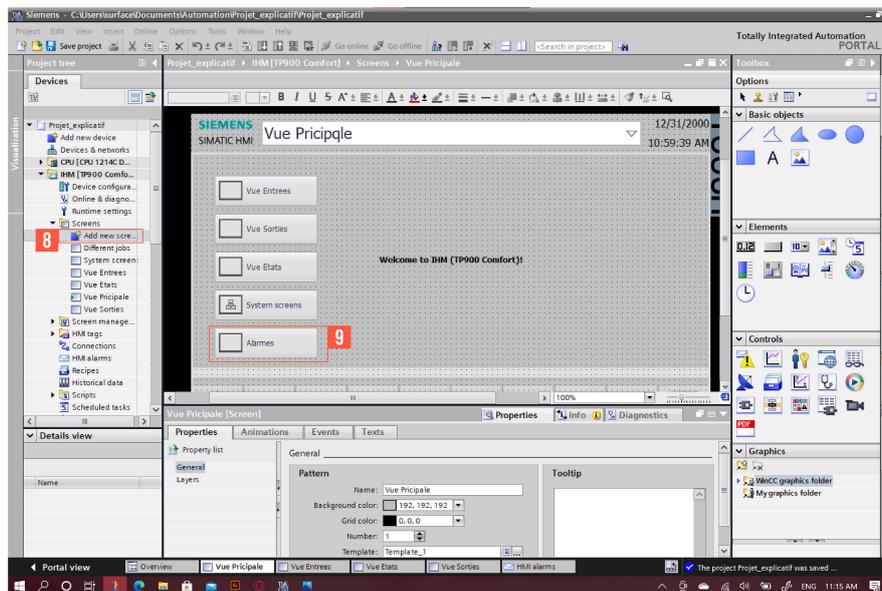


FIGURE 5.35 – Partie 3 (suite)

8. Dans le répertoire des vues, cliquer sur “Add a new screen” ou ajouter une nouvelle vue en Français.
9. Créer un bouton d’accès à la vue des alarmes depuis la vue principale.

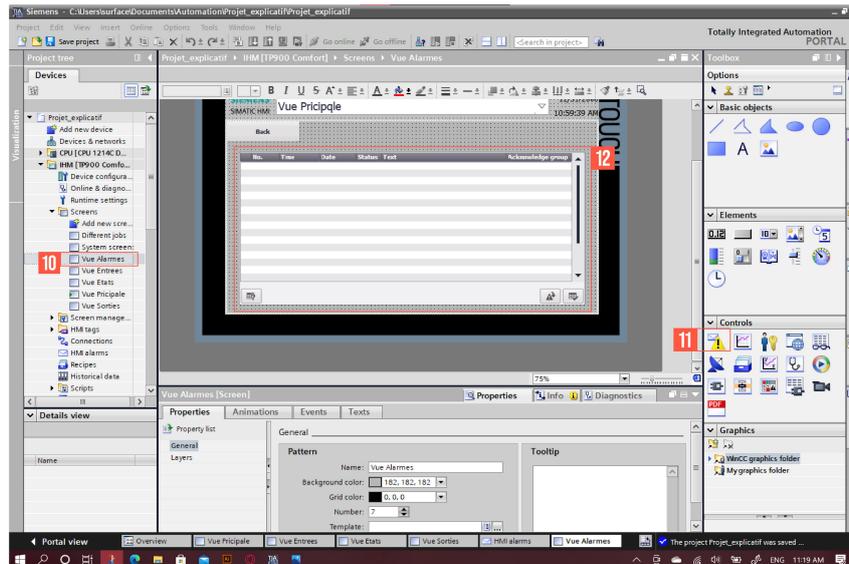


FIGURE 5.36 – Partie 3 (suite)

10. Aller dans la vue alarmes.
11. Cliquer sur l'icône afin de placer le gestionnaire des alarmes dans la vue.
12. Gestionnaire des alarmes a pour but d'afficher en temps réel toutes les alarmes activées durant la progression du programme. La provocation de ces alarmes peut être archivée et enregistré dans un fichier de données (Excel par exemple). Il est aussi possible d'acquiescer certaines alarmes selon leur classe.

Simulation de l'IHM :

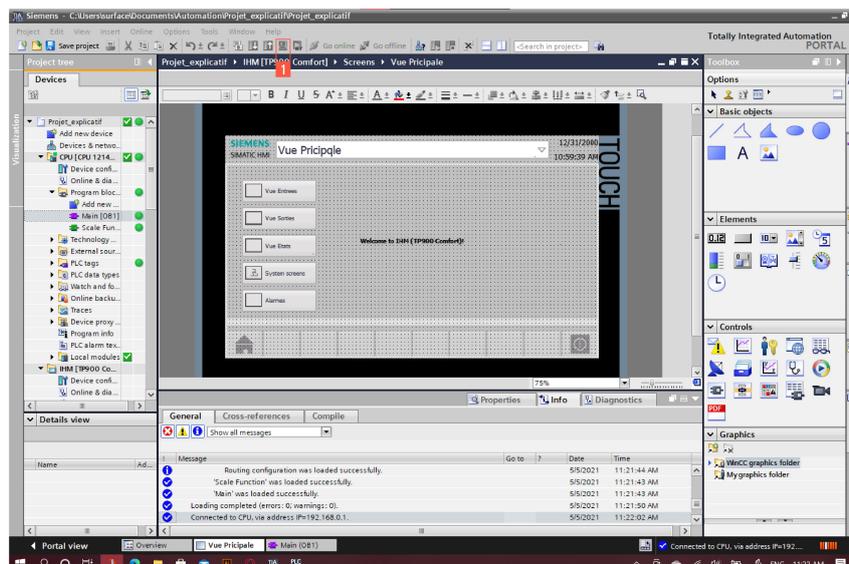


FIGURE 5.37 – Partie 4

1. Cliquer sur l'icône afin de commencer la simulation. Noter qu'il est nécessaire de lancer le simulateur de la CPU en premier lieu (voir la partie 8 précédente).

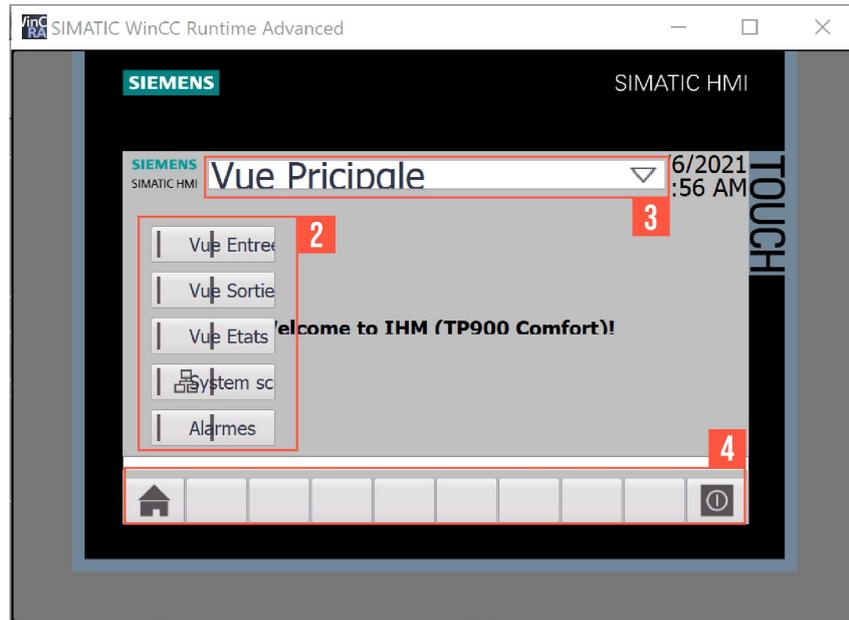


FIGURE 5.38 – Partie 4 (Fin)

2. Cliquer sur les boutons afin de naviguer entre les différentes vues de l'interface.
3. Alternative de navigation entre les vues de l'IHM.
4. Boutons "Home" et "Stop Runtime". Possibilité de rajouter d'autres fonctions.

6. Annexe B :

Montage et câblage du ATV12H037M2

Schéma de câblage général

Pour une alimentation monophasée 100-120 V

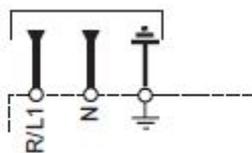


FIGURE 6.1 – Câblage du ATV12....F1

Pour une alimentation monophasée 200-240 V

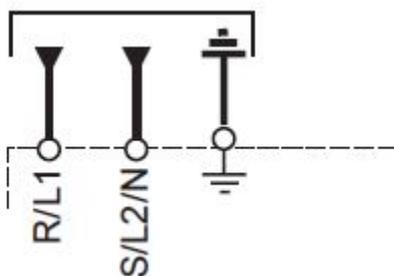


FIGURE 6.2 – Câblage du ATV12....M2

Pour une alimentation triphasée 200-240 V

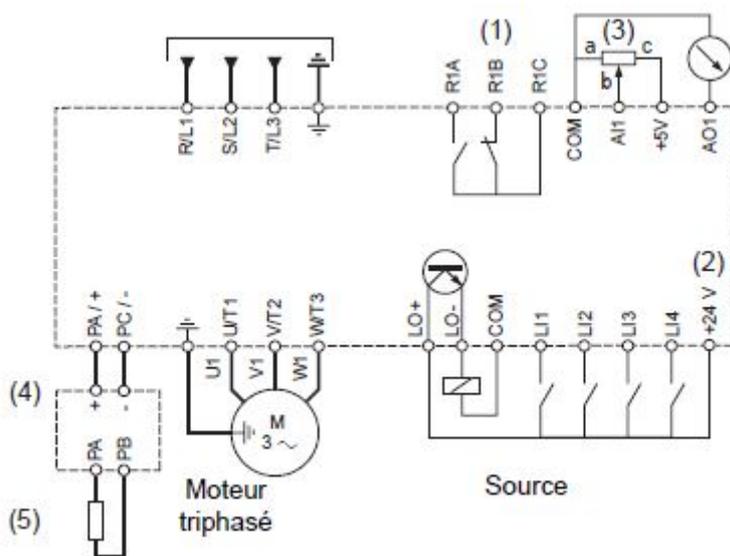


FIGURE 6.3 – Câblage du ATV12....M3

Les composants cités dans la figure

- (1) Contacts de relais R1, pour signalisation à distance de l'état du variateur.
- (2) Interne + 24 V c. Si une source externe est utilisée (+ 30 V c au maximum), le 0 V est connecté de la source sur la borne COM. La borne + 24 V c sur le variateur n'est pas utilisée.
- (3) Potentiomètre de référence SZ1RV1202 (2, 2k Ω) ou similaire (10k Ω maximum).
- (4) Module de freinage optionnel VW3A7005.
- (5) Résistance de freinage optionnelle VW3A7ppp ou autre résistance acceptable. Consultez les valeurs possibles de la résistance dans le catalogue.

Conditions de montage et de température

- Installation du variateur en position verticale, à $\pm 10^\circ$.
- Ne pas le placer à proximité d'une source de chaleur.
- Laisser un espace libre suffisant afin d'assurer la circulation de l'air du bas vers le haut pour le refroidissement correct du radiateur.
- Espace libre devant l'unité : 10mm(0,4in.) au minimum.
- Il est conseillé de retirer les caches des orifices de ventilation situés sur le dessus du variateur, comme l'illustre la figure ci-dessous.
- Il est recommandé d'installer le variateur sur une surface de dissipant thermiquement.

[23]

7. Annexe C :

Perspectives

Le travail qu'on a réalisé est le fruits des efforts qu'on a fournies durant notre cursus en Automatique. Dans ce derniers nous avons appris un tas de notions et on a acquis plusieurs compétences. Notre but était de les appliquer sur notre projet mais afin d'exploiter toutes ces informations, il faudrait se montrer plus motivé et bénéficier de tout le temps qu'on vous donne pour réaliser votre objectif. Comme perspectives pour ce projet, on souhaite de la part des étudiants d'exploiter encore mieux la station :

- Élaborer plus de cahier de charges, utiliser plus d'outils offerts par le logiciel TIA Portal,
- exploiter les données recueillit à travers l'archivage pour réaliser pratiquement l'identification du modèle de la station et le comparer avec celui établi par la méthode théorique,
- utiliser les données d'archivage pour détection de défauts tels que les fuites (jouer sur le taux d'ouverture pour générer des perturbation ou des fuites) et les pannes du matériel,
- réaliser un système SCADA en reliant au PC les deux stations de pompes qui sont au laboratoire pour de la supervision,
- synthétiser plus de commande automatiques et profiter de l'avantage de leurs aspect algorithmique.

Bibliographie

- [1] automation 24. *Siemens SM 1232 AA - 6ES7232-4HB32-0XB0*. URL : <https://www.automation24.fr/siemens-sm-1232-aa-6es7232-4hb32-0xb0>.
- [2] PLC ACADEMY. *structured text tutorial*. URL : <https://www.plcacademy.com/structured-text-tutorial/>.
- [3] *Automates programmable*. 2017. URL : <https://www.automation-sense.com/pages/c-est-quoi-un-automate.html>. (accessed : 03.04.2021).
- [4] BOURAS Kamel CHKEBKEB DAOU. *Réalisation, commande et supervision d'un prototype d'une station de pompage*. 2019.
- [5] PLC CITY. *6AV2 124-0JC01-0AX0 Siemens HMI TP900 COMFORT, COMFORT PANEL TOUCH OPERATION 9 WIDESCREEEN-TFT-DISPLAY*. URL : <https://www.plc-city.com/shop/fr/siemens-simatic-hmi-comfort-panels/6av2124-0jc01-0ax0.html>.
- [6] PLC CITY. *6ES7214-1BG31-0XB0 Siemens S7-1200, CPU 1214C, COMPACT CPU, AC/DC/RLY, ONBOARD I/O*. URL : <https://www.plc-city.com/shop/fr/siemens-simatic-s7-1200-cpu-1214c/6es7214-1bg31-0xb0.html>.
- [7] Daniel CLENET. « Démarreurs et variateurs de vitesse électroniques ». In : Schneider Electric, 2003. Chap. 1.2.
- [8] CONRAD. *Alimentation rail DIN Siemens SITOP PSU100S 24 V/10 A 24 V/DC 10 A 240 W 1 x*. URL : <https://www.conrad.fr/p/alimentation-rail-din-siemens-sitop-psu100s-24-v10-a-24-vdc-10-a-240-w-1-x-513430>.
- [9] COPADATA. *Qu'est-ce qu'une IHM ?* 2021. URL : <https://www.copadata.com/fr/produits/zenon-software-platform/visualisation-contrrole/quest-ce-qu-une-ihm-interface-homme-machine-copa-data/>. (accessed : 03.04.2021).
- [10] Wiki DÉBROUILLARD. *Item :Arduino Uno*. URL : https://www.wikidebrouillard.org/wiki/Item:Arduino_Uno.
- [11] DOCSITY. « Modeling Of The Double Tank Process-Fluid Mechanics- Handout ». In : *Amet University Fluid mechanics* (2012). DOI : <https://www.docsity.com/en/modeling-of-the-double-tank-process-fluid-mechanics-handout/85473/>.
- [12] Schneider ELECTRIC. *Altivar ATV12 - variateur de vit.* URL : <https://www.se.com/fr/fr/product/ATV12H037M2/altivar-atv12---variateur-de-vit---0%5C%2C37kw-0%5C%2C55cv---200..240v---1ph---radiateur/>.
- [13] Schneider ELECTRIC. *Automate programmable industriel (PLC)s*. 2021. URL : <https://www.se.com/fr/fr/work/products/product-launch/guides/plc.jsp>. (accessed : 03.04.2021).

- [14] Schneider ELECTRIC. *Zelio Relay RSB - relais PCB*. URL : <https://www.se.com/fr/fr/product/RSB2A080BDS/zelio-relay-rsb---relais-pcb-embrochable-avec-embase---2of---8a---24vdc/>.
 - [15] Ali EXPRESS. *Convertisseur de tension 4-20 mA à 0-3.3V/5V/10V*. URL : <https://fr.aliexpress.com/item/32821897969.html>.
 - [16] Leo Pompe FRANCE. *APm Pompe Périphérique*. URL : <https://leopompes-france.fr/fr/pompe/APm/>.
 - [17] *Historique de l'API*. 2011. URL : <https://www.institut-numerique.org/iii41-historique-de-lapi-4e1b375ea28df>. (accessed : 03.04.2021).
 - [18] Groupe MAHMOUDI. *Appareillage de tête*. URL : <http://groupe-mahmoudi.com/product-category/distribution-electrique-bt/appareillage-de-tete/>.
 - [19] Pompe et MOTEUR. *Moteur asynchrone une valeur sûre*. 2017. URL : <https://www.pompe-moteur.fr/blog/moteur-asynchrone-triphas-valeur-sure-n16>.
 - [20] Hydro QUÉBEC. *Variateurs de vitesse*. 2018. URL : <https://www.hydroquebec.com/data/affaires/pdf/variateurs-vitesse.pdf>.
 - [21] REICHELT. *TD HMI KTP400 IHM SIMATIC, KTP400 Basic*. URL : <https://www.reichelt.com/de/fr/ihm-simatic-ktp400-basic-td-hmi-ktp400-p201775.html/>.
 - [22] Benoist ROUSSEAU. *Société Siemens*. 2013. URL : <https://www.andlil.com/societe-siemens-138930.html>. (accessed : 02.04.2021).
 - [23] SCHNEIDER. « Variateur de fréquence Schneider Electric ALTIVAR 12 ». In : 2014. Chap. 1.2.
 - [24] SICK. *Capteurs à ultrasons UM18 / UM18-1*. URL : <https://www.sick.com/be/fr/capteurs-de-distance/capteurs-a-ultrasons/um18/um18-51111/p/>.
 - [25] SIEMENS. *3RH21401AP00*. URL : <https://mall.industry.siemens.com/mall/en/us/Catalog/Product/3RH2140-1AP00>.
 - [26] SIEMENS. *Libération de la famille des automates SIMATIC S7-1200*. 2009. URL : <http://support.industry.siemens.com/cs/document/36524193/lib%5C%C3%5C%A9ration-de-la-famille-des-automates-simatic-s7-1200?dti=0&lc=fr-WW>.
 - [27] SIEMENS. « S7 1200 Manuel système ». In : A5E02486682-AG, 2014. Chap. 1.2. URL : https://media.automation24.com/manual/fr/91696622_s71200_System_Manual_fr-FR_fr-FR.pdf.
 - [28] SIEMENS. *SIMATIC - The Automation brand by Siemens*. 2021. URL : <https://new.siemens.com/global/en/products/automation/topic-areas/simatic.html>. (accessed : 03.04.2021).
 - [29] Go TRONIC. *Capteur de débit YFS201*. URL : <https://www.gotronic.fr/art-capteur-de-debit-yfs201-18979.htm>.
 - [30] WIKIPÉDIA. *Commande robuste*. 2020. URL : https://fr.wikipedia.org/wiki/Commande_robuste.
-

Bibliographie

- [31] WIKIPÉDIA. *Formule de Torricelli*. 2020. URL : https://fr.wikipedia.org/wiki/Formule_de_Torricelli.
- [32] Pompe ZANINI. *COMMENT FONCTIONNE UNE POMPE CENTRIFUGE ?* 2019. URL : <https://www.pompezanni.it/fr/news-fr/fonctionnement-pompe-centrifuge/>.