

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEURE ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département du Génie Electrique
Option AUTOMATIQUE

Mémoire du projet de fin d'étude
pour l'obtention du diplôme d'ingénieur d'état en Automatique

Thème :

*Réalisation et gestion d'un prototype de station de
pompage à base d'automates programmables industriels
SIEMENS*

Sponsorisé par :

SIEMENS

Réalisé par :

ABRICHE Anissia

BELKAS Salah-eddine

Proposé par :

Mr. E.M. BERKOUK

Mr. R.REMY

Promotion 2006/2007

REMERCIEMENTS

Nous tenons à exprimer nos vifs remerciements à notre promoteur Pr.BERKOUK de l'Ecole Nationale Polytechnique pour nous avoir encadrer durant notre projet de fin d'études et nous conseillé tout le long de notre travail

Nous remercions également notre co-promoteur Mr REMY Roland, responsable de SIEMENS SA, pour son encadrement, sa générosité dans la réalisation du projet, et sa confiance au sein de l'entreprise.

Nous remercions chaleureusement les membres du jury pour l'honneur qu'ils nous ont fait en acceptent d'évaluer notre projet.

Nos sincères remerciements aux ingénieurs de Siemens qui nous ont conseillé et éclairé sur notre travail tout le long de notre projet.

Nous souhaitons aussi remercier tous les enseignants de l'Ecole Nationale Polytechnique d'Alger, et en particulier Mr CHKIREB et Mr R.ILLOUL pour leurs précieux conseils, Nos professeur d'Automatique qui nous ont encadrés auparavant et tous nos enseignants pour les connaissances qu'ils nous ont transmis, leur disponibilité et leurs efforts.

Nous remercions tout le personnel de l'école, de SIEMENS et tout les élèves de génie électrique.

Que tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste travail trouvent ici l'expression de notre sincère gratitude.

DEDICACES

Je dédie ce présent travail à ma mère et mon père qui m'ont soutenu et aider tout le long de mon parcours et sans qui je ne serais pas ou j'en suis.

À mes deux sœurs Souhila et Samira qui ont toujours été la pour moi, et mon petit frère Alilou pour ce qu'il est et son enthousiasme, et ma petite nièce SOFIA que j'adore.

A mes beaux frères Hakim et Nassim qui m'ont encouragé

To my dear Said pour son soutien, son aide et sa présence

A mon binôme SALAH avec qui j'ai travaillé durant trois années et qui ma supporté

Toute l'équipe SIEMENS , surtout Kamel, Mahmoud, Redah, Rabeh, Nawfel,, Redjem, Karim, Kamel, Mouhamed, Nassim, Ghania très gentille, et tout particulièrement Mr Dielman et Mr Kherouba pour nous avoir donner la chance de faire ce projet au sein de l'entreprise.

À mes oncles et tantes et mon petit Grand Père

Mes cousins, Redouane, Lyes et Mohamed qui ont répondu présents, et mes cousines pour leurs encouragements.

À mes très bonnes Amies Soumaya, Ferial et Nanou pour leur présence

A toute ma promotion et la promotion 2006, tout particulièrement Djallil , Merwane, Benyou, Lotfi , Hassan et Nedjma, et tout ceux que je n'ai pas cité.

À tout mes amis de l'Ecole Nationale Polytechnique avec qui j'ai eu d'agréable moments et appris beaucoup de choses.

À tous mes amis que je n'ai pas cité et qui sont présent dans mes pensées.

Anissia Dahbia ABR ICHE

DEDICACE

Je commence d'abord par remercier mon binôme : « Anissia, chapeau »

Je dis bravo à tous mes amis, de l'école et en dehors, ils étaient toujours là quand j'en avais besoin.

Je dédie ce travail à mon cher père et ma chère mère, ma sœur, mes deux frères et mes grands parents que je ne remercierai jamais assez.

Je tiens à remercier toute ma famille, et spécialement ma tante Keira et son mari Mr. ZITOUN pour leur soutien et leur suivi.

Je dédie l'intégralité de ce travail à la personne sans laquelle je ne serais point, ma grand mère Yamina « lah yer7amha »

« Grand mère si vous m'entendez aujourd'hui, sachez que vous nous manquez et que le vide n'est pas comblé. Chaque jour qui passe, on essaye de faire de notre possible pour que vous soyez fiers de nous »

SOMMAIRE

| | |
|--|----|
| Introduction générale | 1 |
| | |
| CHAPITRE I : Les automates programmables industriels SIEMENS | |
| I.1 Historique | 4 |
| I.2 Définition générale | 5 |
| I.3 Architecture des automates programmables industriels | 5 |
| I.4 Structure interne des automates programmables | 7 |
| I.4.1 Le processeur | 7 |
| I.4.2 Les modules d'entrées/sorties | 7 |
| I.4.3 Les mémoires | 8 |
| I.4.4 L'alimentation | 8 |
| I.4.5 Liaisons de communication | 8 |
| I.5 Présentation de la gamme SIMATIC de SIEMENS | 9 |
| I.5.1 Les différentes variantes dans la gamme SIMATIC | 9 |
| I. 5.2 Description du STEP7 | 12 |
| I.6 Stratégie pour la conception d'une structure programme complète et optimisée | 15 |
| I.7 Exemple de Création et d'édition d'un projet S7 | 16 |
| I.7.1 Création du projet dans SIMATIC Manager | 20 |
| I.7.2 Configuration matérielle | 21 |
| I.7.3 Création de la table des mnémoniques | 22 |
| I.7.4 Elaboration du programme S7 | 23 |
| I.7.5 Chargement dans le système cible à partir de la PG/PC | 30 |
| I.7.6 Surveillance du fonctionnement et diagnostic du matériel | 31 |
| I.8 Régulation PID des processus à base d'automates SIMATIC S7 | 32 |

| | | |
|-------|--|----|
| I.8.1 | Principe de régulation | 32 |
| I.8.2 | Concept de la fonction de régulation PID dans SIMATIC S7 | 32 |
| I.8.3 | Le bloc FB 41 « CONT_C » | 33 |
| I.8.4 | Paramétrage du régulateur PID du type continue FB 41 avec STEP 7 | 36 |
| I.9 | Conclusion | 37 |

CHAPITRE II : Le réseau local industriel PROFIBUS

| | | |
|----------|---|----|
| II.1 | Introduction | 39 |
| II.2 | Définitions élémentaires | 39 |
| II.3 | Définition et normalisations PROFIBUS | 39 |
| II.4 | Variantes du réseau PROFIBUS | 41 |
| II.4.1 | PROFIBUS-DP | 41 |
| II.4.2 | PROFIBUS-FMS | 41 |
| II.4.1 | PROFIBUS-PA | 41 |
| II.5 | Principe d'accès au BUS | 42 |
| II.6 | Structure des télégrammes | 43 |
| II.7 | Les techniques de transmissions | 43 |
| II.7.1 | Transmission RS 485 | 43 |
| II.7.2 | Transmission optique | 43 |
| II.8 | Mise en œuvre d'une application à base du PROFIBUS DP avec Maître CPU 315- 2DP et esclave MICROMASTER 440 : « Entraînement d'un chariot sur un plan incliné » | 44 |
| II.8.1 | Le cahier de charge | 44 |
| II.8.2 | La mise en œuvre de l'application | 47 |
| II.8.2.1 | Description du banc d'essai et du profil de communication pour l'application | 48 |

| | | |
|-----------|--|----|
| II.8.2.2 | Création du projet dans STEP7 | 56 |
| a. | Configuration du matériel | 56 |
| b. | Conception et écriture du Programme | 60 |
| II.8.2.3 | Paramétrages du Variateur de vitesse MICROMASTER 440 | 65 |
| a. | Les Méthodes de paramétrages dans <i>DRIVE MONITOR</i> | 68 |
| b. | Les Types de paramètres dans un variateur de vitesse SIMOVERT de SIEMENS | 68 |
| c. | Etapas de paramétrages dans « Drive Monitor » | 69 |
| II.8.2.4. | Test de l'application | 71 |
| II.9 | Conclusion | 72 |

CHAPITRE III : Le progiciel de conception des interfaces Homme/Machine *Win CC flexible*

| | | |
|---------|---|----|
| III.1 | Introduction | 78 |
| III.2 | Progiciel de conception et configuration d'interface <i>Win CC flexible</i> | 78 |
| III.2.1 | Présentation générale | 78 |
| III.2.2 | Elément du progiciel <i>Win CC flexible</i> | 79 |
| III.2.3 | Utilisation des variables | 81 |
| III.2.4 | Compilation et Simulation | 82 |
| III.3 | Conclusion | 82 |

CHAPITRE IV : Réalisation et gestion d'un prototype de station de pompage à base d'automates SIEMENS

| | | |
|--------|---|----|
| IV.1 | Introduction | 84 |
| IV.2 | Présentation des éléments de l'installation | 84 |
| IV.2.1 | Description du Prototype de la station de pompage | 85 |

| | | |
|--------|---|-----|
| IV.2.2 | Eléments de commande, de diagnostic et de supervision du Prototype de la station de pompage | 87 |
| IV.3 | Développement du projet pour gestion du prototype de la station de pompage | 88 |
| IV.3.1 | Définition du cahier des charges | 90 |
| IV.4 | Création du projet S7 pour la station de pompage | 92 |
| IV.4.1 | Insertion des stations dans <i>SIMATIC MANAGER</i> | 92 |
| IV.4.2 | Configuration du matériel dans STEP7 | 93 |
| IV.4.3 | Création du programme de gestion dans STEP7 | 95 |
| IV.4.4 | L'interface homme machine HMI créée par le progiciel Win CC flexible | 96 |
| IV.4.5 | Paramétrage du <i>MASTERDRIVE</i> | 99 |
| IV.5 | Optimisation des fonctions de régulations et mise en service de la station de pompage | 102 |
| IV.5.1 | Détermination de la plage linéaire de fonctionnement | 103 |
| IV.5.2 | Détermination des paramètres du régulateur P.I | 104 |
| | a. Expérimentation | 106 |
| | a.1. L'approche de <i>Ziegler Nichols</i> | 106 |
| | a.2. Les approches de <i>Chien-Hrones-Reswick</i> | 107 |
| | b. Récapitulatif des essais | 111 |
| IV.6 | Analyse des résultats et conclusion | 111 |
| | Conclusion Générale | 113 |
| | BIBLIOGRAPHIE | 115 |

LISTE DES FIGURES

| | |
|---|----|
| Figure I.1 : <i>le sigle de la société Siemens</i> | 4 |
| Figure I.2 : <i>L'automate programmable Siemens</i> | 6 |
| Figure I.3 : <i>structure interne d'un API</i> | 7 |
| Figure I.4 : <i>Présentation de la gamme de SIMATIC</i> | 9 |
| Figure I.5 : <i>L'API S200</i> | 10 |
| Figure I.6 : <i>L'API S300</i> | 10 |
| Figure I.7 : <i>L'API S400</i> | 10 |
| Figure I.8 : <i>La gamme SIMATIC C7</i> | 11 |
| Figure I.9 : <i>La gamme SIMATIC M7</i> | 12 |
| Figure I. 10 : <i>mode de représentation des langages basiques de programmation STEP 7</i> | 13 |
| Figure I.11 : <i>logiciel de simulation PLC-SIM</i> | 15 |
| Figure I.12 : <i>Choix du mode de fonctionnement dans la première application</i> | 17 |
| Figure I.13 : <i>Le fonctionnement en mode manuel dans la première application</i> | 17 |
| Figure I.14 : <i>Le fonctionnement en mode automatique dans la première application</i> | 18 |
| Figure I.15 : <i>La chaîne de remplissage à automatisé</i> | 19 |
| Figure I.16 : <i>Le coffret SIDEMO pour S300</i> | 20 |
| Figure I.17 : <i>SIMATIC Manager</i> | 20 |
| Figure I.18 : <i>Configuration matérielle</i> | 21 |
| Figure I.19 : <i>Création du programme S7</i> | 22 |
| Figure I.20 : <i>Table de mnémonique relative à la chaîne de remplissage</i> | 23 |
| Figure I.21 : <i>Arborescence du programme pour la deuxième application</i> | 24 |
| Figure I.22 : <i>Diagnostic du matériel</i> | 31 |

| | |
|--|----|
| Figure I.23 : <i>Composition d'une boucle de régulation</i> | 32 |
| Figure I.24 : <i>L'algorithme du bloc FB41</i> | 33 |
| Figure I.25 : <i>Paramétrage du régulateur PID</i> | 36 |
| Figure I.26 : <i>Tracé des trois courbe consigne-mesure-commande</i> | 37 |
| Figure II.1 : <i>Les sept couches du modèle OSI</i> | 40 |
| Figure II.2 : <i>Classification des réseaux PROFIBUS</i> | 41 |
| Figure II.3 : <i>Principe d'accès au BUS</i> | 42 |
| Figure II.4 : <i>Schéma de principe de l'application du chariot</i> | 44 |
| Figure II.5 : <i>Chronogramme de fonctionnement</i> | 46 |
| Figure II.6 : <i>l'organigramme de l'application du chariot</i> | 47 |
| Figure II.7 : <i>Schéma de principe de l'application</i> | 48 |
| Figure II.8 : <i>Banc d'essai SIDEMO_MM440</i> | 49 |
| Figure II.9 : <i>Structure générale du télégramme de transmission cyclique des données</i> | 50 |
| Figure II.10 : <i>Données processus de type PPO</i> | 51 |
| Figure II.11 : <i>Configuration matériel pour la CPU315 et ces modules E/S</i> | 56 |
| Figure II.12 : <i>Création d'un Réseau PROFIBUS Maître</i> | 56 |
| Figure II.13 : <i>Paramètres de la transmission PROFIBUS-DP</i> | 57 |
| Figure II.14 : <i>Insertion du réseau maitre CPU – Esclave MM440</i> | 57 |
| Figure II.15 : <i>Insertion du réseau maitre CPU – Esclave MM440</i> | 58 |
| Figure II.16 : <i>Adressage des entrées-sorties pour la communication PROFIBUS-DP</i> | 58 |
| Figure II.17 : <i>Adressage des entrées-sorties virtuelles STW-HSW et ZSW-HIW</i> | 60 |
| Figure II.18 : <i>Arborescence du programme pour la deuxième application</i> | 61 |
| Figure II.19 : <i>blocs programme dans SIMATIC Manager</i> | 62 |
| Figure II.20 : <i>Tableau de mnémonique</i> | 63 |

| | |
|---|----|
| Figure II.21 : <i>Bloc de données DB10</i> | 64 |
| Figure II.22 : <i>Bloc de données DB11</i> | 64 |
| Figure II.23 : <i>La PMU</i> | 65 |
| Figure II.24 : <i>Pupitre opérateur OPIS</i> | 66 |
| Figure II.25 : <i>Le paramétrage via Drive Monitor</i> | 67 |
| Figure II.26 : <i>Le logiciel DriveMonitor et le menu Drive Navigator</i> | 67 |
| Figure II.27 : <i>Les démarches pour la configuration via DriveMonitor</i> | 70 |
| Figure II.28 : <i>Accès à la configuration de la liaison USS PC-Variateur</i> | 71 |
| Figure II.29 : <i>Les paramètres de communication USS PC-Variateur</i> | 72 |
| Figure II.30 : <i>Accès aux partenaires réseau USS</i> | 72 |
| Figure II.31 : <i>Détection et identification du variateur dans la liaison USS</i> | 72 |
| Figure II.32 : <i>Représentation de la liste des paramètres dans DRIVE MONITOR</i> | 73 |
| Figure II.33 : <i>Visualisation du mot d'état et du mot de commande dans DRIVE MONITOR</i> | 75 |
| Figure II.34 : <i>Courbe de vitesse pour l'application « Entraînement d'un chariot sur un plan incliné »</i> | 75 |
| Figure III.1 : <i>L'interface Homme/Machine dans un processus automatisé</i> | 79 |
| Figure III.2 : <i>Etapas pour la conception d'une interface via Win CC flexible</i> | 79 |
| Figure III.3 : <i>Vue d'ensemble du progiciel Win CC flexible</i> | 81 |
| Figure III.4 : <i>éditeur de variable dans le Win CC flexible</i> | 82 |
| Figure IV.1 : <i>Le prototype de station de pompage</i> | 85 |
| Figure IV.2 : <i>Disposition des éléments dans le prototype de station de pompage hydraulique</i> | 87 |
| Figure IV.3 : <i>Organigramme de l'application finale</i> | 89 |
| Figure IV.4 : <i>Organigramme fonctionnel de l'application finale</i> | 90 |
| Figure IV.5 : <i>Synoptique du prototype</i> | 91 |

| | |
|---|-----|
| Figure IV.6 : Principe de régulation en cascade Vitesse/Débit | 92 |
| Figure IV.7 : Création du projet Station Pompage | 93 |
| Figure IV.8 : Configuration matériel | 93 |
| Figure IV.9 : Bloc de donnée pour la commande du variateur | 94 |
| Figure IV.10 : Bloc de donnée pour l'état du variateur | 94 |
| Figure IV.11 : Table des mnémoniques | 95 |
| Figure IV.12 : Création des différents blocs de programme dans SIMATIC MANAGER | 96 |
| Figure IV.13 : Le terminal d'exploitation MP270 | 96 |
| Figure IV.14 : Vue générale crée pour la HMI | 97 |
| Figure IV.15 : Configuration de la liaison Panel-Automate | 98 |
| Figure IV.16 : logiciel de paramétrage DRIVE MONITOR | 99 |
| Figure IV.17 : Identification du moteur | 99 |
| Figure IV.18 : Mode de régulation et type de commande | 100 |
| Figure IV.19 : Assignment des entrées/sorties et d l'adresse PROFIBUS –DP | 100 |
| Figure IV.20 : Paramétrage du mode de régulation en U/f | 101 |
| Figure IV.21 : Validation de la configuration | 101 |
| Figure IV.22 : Paramétrage du régulateur PID | 102 |
| Figure IV.23 : Vue pour la régulation PID de débit dans le panneau MP270 | 103 |
| Figure IV.24 : La caractéristique statique du système en boucle ouverte | 104 |
| Figure IV.25 : Approximation des valeurs T_u , T_g , K_s | 105 |
| Figure IV.26 : Les courbes Consigne- mesure-commande par l'approche Ziegler-Nichols | 106 |
| Figure IV.27 : Rejet des perturbations par l'approche Ziegler-Nichols | 106 |
| Figure IV.28 : Les courbes Consigne- mesure-commande par l'approche I de Chien | 107 |
| Figure IV.29 : Rejet de perturbation par l'approche I de Chien | 107 |

| | |
|---|-----|
| Figure IV.30 : <i>Les courbes Consigne- mesure-commande par l'approche II de Chien</i> | 108 |
| Figure IV.31: <i>Rejet de perturbation par l'approche II de Chien</i> | 108 |
| Figure IV.32 : <i>Les courbes Consigne- mesure-commande par l'approche III de Chien</i> | 109 |
| Figure IV.33: <i>Rejet de perturbation par l'approche III de Chien</i> | 109 |
| Figure IV.34 : <i>Les courbes Consigne- mesure-commande par l'approche IV de Chien</i> | 110 |
| Figure IV.35: <i>Rejet de perturbations par l'approche IV de Chien</i> | 110 |

LISTE DES TABLEAUX

| | |
|---|-----|
| Tableau I.1 : <i>L'adressage absolu des entrées/sorties de la première application</i> | 33 |
| Tableau I.2 : <i>Les paramètres d'entrée du bloc fonctionnel FB 41</i> | 35 |
| Tableau I.3 : <i>Les paramètres de sortie du bloc fonctionnel FB 41</i> | 35 |
| Tableau II.1 : <i>Structure de la PZD dans le PPO3</i> | 52 |
| Tableau II.2 : <i>Le mot de contrôle STW</i> | 54 |
| Tableau II.3 : <i>Le mot d'état ZSW</i> | 55 |
| Tableau II.4 : <i>Adressage virtuel du double mot de commande STW-HSW</i> | 59 |
| Tableau II.5 : <i>Adressage virtuel du double mot d'état ZSW-HIW</i> | 59 |
| Tableau II.6 : <i>Adressage des entrées processus</i> | 60 |
| Tableau II.7 : <i>Etats possibles pour la liaison PC-Variateur</i> | 68 |
| Tableau II.8 : <i>Structure de la liste des paramètres</i> | 74 |
| Tableau IV.1 : <i>valeurs des grandeurs de sortie (débit) en fonction de la commande (vitesse)</i> | 104 |
| Tableau IV.2 : <i>Récapitulatif des résultats obtenus durant l'expérimentation</i> | 111 |

Symboles et abréviation

| | |
|---------------------------|---|
| AOP | Panneau de commande avancé |
| API | Automate Programmable industriel |
| A $n.m$ | Sortie de l'automate numéro m , se trouvant sur le module numéro n |
| AD n | Double Mot de Sortie de l'automate, se trouvant sur l'adresse numéro n |
| AW n | Mot de Sortie de l'automate, trouvant sur l'adresse numéro n |
| AI | entrée analogique |
| AO | Sortie analogique |
| BOP | <i>Basic Operating Panel</i> |
| CONT | Le langage à base de schémas de contacts |
| CPU | <i>Central Processing Unit</i> |
| DP | <i>Decentralized Peripheral</i> |
| DI | Entrée TOR |
| DO | Sortie TOR |
| EEPROM | <i>Electrically Erasable Programmable Read Only Memory</i> |
| E $n.m$ | Entrée « TOR » vers l'automate numéro m , se trouvant sur le module numéro n |
| ED n | Double Mot d'entée de l'automate, se trouvant sur l'adresse numéro n |
| EW n | Mot d'entée de l'automate, trouvant sur l'adresse numéro n |
| FB | Bloc de fonction |
| FC | Fonction |
| FMS | <i>Fieldbus Message Specification</i> |
| HIW | Valeur principale pratique |
| HSW | Valeur principale théorique |
| HMI | <i>Human Machine Interface</i> |
| Kp | Le gain du régulateur PID |
| Ks | Le gain en boucle ouverte |
| LOG | Le langage à base de logigramme |
| LIST | Le langage de liste d'instructions |
| MASTERDRIVE | Variateur de vitesse de la gamme <i>SIMOVERT</i> pour les applications de basses, |

moyennes et hautes puissances.

MICROMASTER Variateur de vitesse de la gamme *SIMOVERT* pour les applications de basses et moyennes puissances

| | |
|------------------|---|
| MPI | <i>Multi Point Interface</i> |
| MM440 | <i>Micro Master 440</i> |
| OSI | <i>Open Systems Interconnection</i> |
| OB | Bloc d'organisation |
| PA | <i>Process Automation</i> |
| PPO | Objet Paramètres/Données processus |
| PKW | Zone des données de paramétrage |
| PZD | Zone des données processus |
| PMU | <i>Parameterization Unit</i> |
| PID | Proportionnel-intégral-dérivée |
| PROM | <i>Programmable Read Only Memory</i> |
| PROFIBUS | <i>Process Field Bus</i> |
| PS | Gamme des alimentations stabilisées de Siemens |
| PG | <i>La console de programmation sur le terrain</i> |
| SIMATIC | <i>Siemens Automatic</i> |
| SITRANS F | <i>Siemens transmitter Flow</i> : filiale Siemens spécialisée en appareils didactiques |
| SITRANS L | <i>Siemens transmitter level</i> : filiale Siemens spécialisée en appareils didactiques |
| S7 | <i>Step 7</i> |
| SM | Gamme des modules E/S des automates de Siemens |
| RAM | <i>Random Access Memory</i> |
| ROM | <i>Programmable Read Only Memory</i> |
| SIDEMO | <i>Siemens Demo</i> : filiale Siemens spécialisée en appareils didactiques |
| SFB | Bloc de fonction spécial |
| SFC | Bloc programme pour le langage évolué textuel |
| STW | Mot de contrôle |
| Tu | Retard pur |
| Tg | Retard compensatoire |
| Ti | Temps d'intégration du régulateur PID |

| | |
|------------|--|
| TFT | <i>Thin Film Transistor</i> |
| USS | Interface Série Universelle |
| VAT | La table de variable dans <i>SIMATIC MANAGER</i> |
| ZSW | Mot d'état |

Introduction Générale

La compétitivité des entreprises impose un recours à la fois fréquent et intensif à des technologies de production avancées. La productique et la complexité des opérations à exécuter, conduisent à la mise en œuvre de dispositifs et systèmes pour l'automatisation des ateliers de fabrication ou de production.

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

Le domaine de l'hydraulique parmi d'autres, est témoin de cette révolution, et de nombreuses sociétés algériennes (*ANBT, SEEAL, Hydro-Aménagement, ...etc.*), orientées dans le stockage, l'épuration et la distribution de l'eau, cherchent à se procurer cette solution d'automatisme au niveau de leurs stations de pompages et de traitements. SIEMENS est une firme compétitive, procurant ce type de service industriel

Dans le but de mettre en relief sa compétence et former les ingénieurs sur ses équipements, SIEMENS nous a proposé la réalisation d'un prototype de station de pompage, connaissant les besoins croissants du pays dans ce domaine.

Notre travail consiste à réaliser un prototype de station de pompage, englobant l'essentiel des systèmes que nous puissions trouver dans une station de pompage réelle, que ce soit du point de vue Hardware ou Software. Cette solution sera à base d'automate SIEMENS qui gère la station du point de vue contrôle du variateur, gestion des électrovannes, régulation du débit, et diagnostic du système. La communication entre les principaux éléments sera via le réseau industriel PROFIBUS.

Plan du travail

Le travail est présenté en quatre chapitres complémentaires :

Le premier chapitre traite les connaissances de base. Après une brève description de toute la gamme SIMATIC S7 des automates **SIEMENS**, on aborde l'environnement de développement **STEP7**. Pour mettre en relief cet environnement, on traite une application en décrivant les différentes étapes pour la création de ce projet. La fin de ce chapitre est consacrée à la régulation PID et sa mise en œuvre dans STEP 7.

Le deuxième chapitre présente l'automate dans son milieu industriel et ses besoins de communications, suivi d'une description pour le réseau local industriel **PROFIBUS**. Après une présentation de ses différentes variantes FMS, PA et DP, un accent particulier a été mis sur cette dernière, laquelle a été illustrée par une deuxième application.

L'exploitation visuelle dans les milieux industriels n'a pas été oubliée dans le troisième chapitre. Une présentation du logiciel *Win CC flexible* de **SIEMENS** sera abordée, en vue de configurer une interface Homme/Machine nécessaire à la commande, le diagnostic et la visualisation à distance.

Ces trois derniers chapitres constituent une préparation au projet qui nous a été soumis. Le dernier chapitre présente une réalisation du prototype et de ses performances. La régulation P.I du débit a été testée avec les approches de **Ziegler-Nichols** et de **Chien-Hrones-Reswick**, et une conclusion conforme aux résultats a été tirée.

On termine par une conclusion générale qui montre les difficultés qu'un étudiant à ses débuts peut rencontrer, des enseignements qu'il en tire, sans oublier les perspectives.

CHAPITRE I

Automates Programmables Industriels SIEMENS

I.1 Historique

Les automates programmables industriels sont apparus à la fin des années soixante aux Etats Unis, à la demande de l'industrie automobile américaine (*General Motors* en leader), qui réclamait plus d'adaptabilité de ses systèmes de commande. Ce n'est qu'en 1971 qu'ils firent leur apparition en France.

Les années soixante-dix connaissent une explosion des besoins industriels dans le domaine de l'automatique, de la flexibilité et l'évolutivité des Systèmes Automatisés de Production (SAP).

Siemens AG est un groupe allemand. Fondé en 1847 par *Werner Von Siemens*, il réalise des équipements électroniques et électrotechniques. Son siège est à Munich, et c'est l'une des plus grosses entreprises européennes.



Figure I.1 : le sigle de la société *Siemens AG*

Dans le cadre de son expansion, *Siemens* crée le 28 janvier 1972, le consortium *Unidata*. Ce projet européen permet, dans les années 1970, l'émergence d'une grande industrie informatique européenne. Les compétences de trois participants ont été mises en commun. La maîtrise d'œuvre, l'architecture des machines et le logiciel ont été attribués à la Compagnie Internationale pour l'Informatique (*CII*), la technologie électronique revenait à *Philips* tandis que *Siemens* se chargeait des périphériques mécaniques.

En 1975 : la France abandonne unilatéralement l'accord *Unidata*, *CII* fusionne avec *Honeywell-Bull*, *Philips* délaisse l'informatique et *Siemens* rejoint *Fujitsu* pour devenir, aujourd'hui, un des plus grands constructeurs mondiaux [23].

I.2 Définition générale

L'automate programmable industriel **A.P.I** ou *Programmable Logic Controller* PLC est un appareil électronique programmable. Il est défini suivant la norme française **EN-61131-1**, adapté à l'environnement industriel, et réalise des fonctions d'automatisme pour assurer la commande de préactionneurs et d'actionneurs à partir d'informations logiques, analogiques ou numériques. C'est aujourd'hui le constituant essentiel des automatismes. On le trouve non seulement dans tous les secteurs de l'industrie, mais aussi dans les services et dans l'agriculture.

La force principale d'un automate programmable industriel API réside dans sa grande capacité de communication avec l'environnement industriel. Outre son unité centrale et son alimentation, il est constitué essentiellement de modules d'entrées/sorties, qui lui servent d'interface de communication avec le processus industriel de conduite.

Et il a comme rôles principaux dans un processus :

- D'assurer l'acquisition de l'information fournie par les capteurs ;
- En faire le traitement ;
- Elaborer la commande des actionneurs ;
- Assurer également la communication pour l'échange d'informations avec l'environnement.

I.3 Architecture des automates programmables industriels

De forme compacte ou modulaire, les automates sont organisés suivant l'architecture suivante :

- Un module d'unité centrale ou CPU, qui assure le traitement de l'information et la gestion de l'ensemble des unités. Ce module comporte un microprocesseur, des circuits périphériques de gestion des entrées/sorties, des mémoires RAM et EEPROM nécessaire pour stocker les programmes, les données, et les paramètres de configuration du système.
- Un module d'alimentation qui, à partir d'une tension 220V/50Hz ou dans certains cas de 24V fournit les tensions continues +/- 5V, +/-12V ou +/- -15V.
- Un ou plusieurs modules d'entrées 'Tout ou Rien' ou analogiques pour l'acquisition des informations provenant de la partie opérative (procédé à conduire).

- Un ou plusieurs modules de sorties ‘Tout ou Rien’ (TOR) ou analogiques pour transmettre à la partie opérative les signaux de commande. Il y a des modules qui intègrent en même temps des entrées et des sorties [1].
- Un ou plusieurs modules de communication comprenant :
 - Interfaces série utilisant dans la plupart des cas comme support de communication, les liaisons RS-232 ou RS422/RS485 ;
 - Interfaces pour assurer l'accès à un bus de terrain ;
 - Interface d'accès à un réseau *Ethernet*.

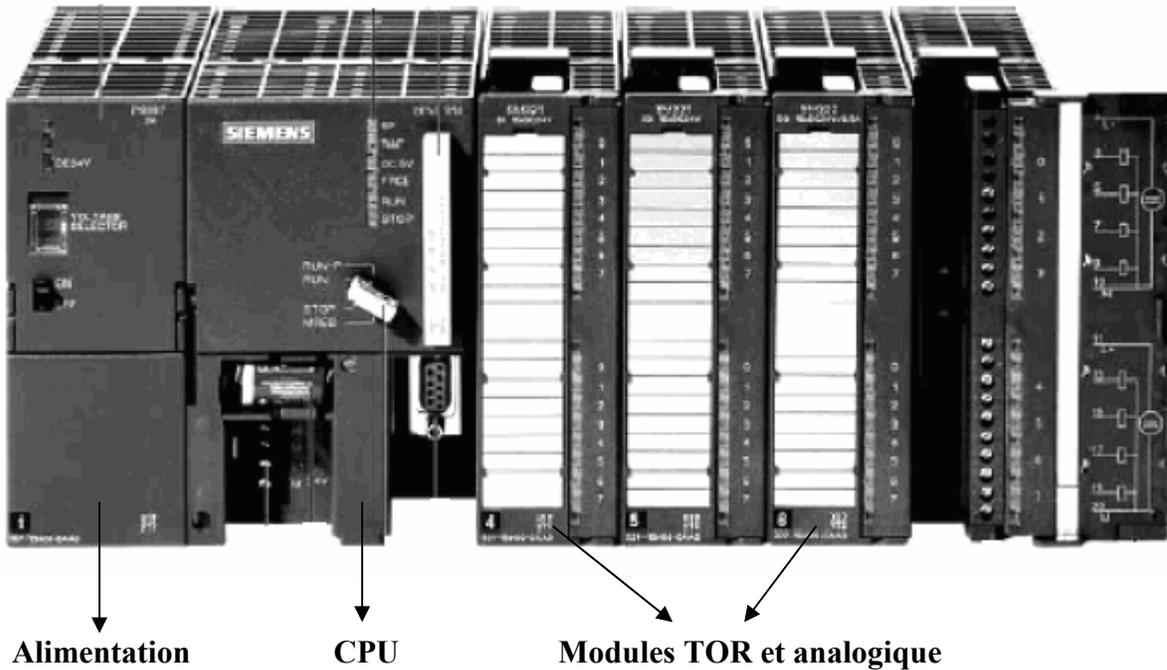


Figure 1.2 : L'automate programmable Siemens [24]

I.4 Structure interne des automates programmables

La structure matérielle interne d'un API obéit au schéma donné sur la figure ci dessous.

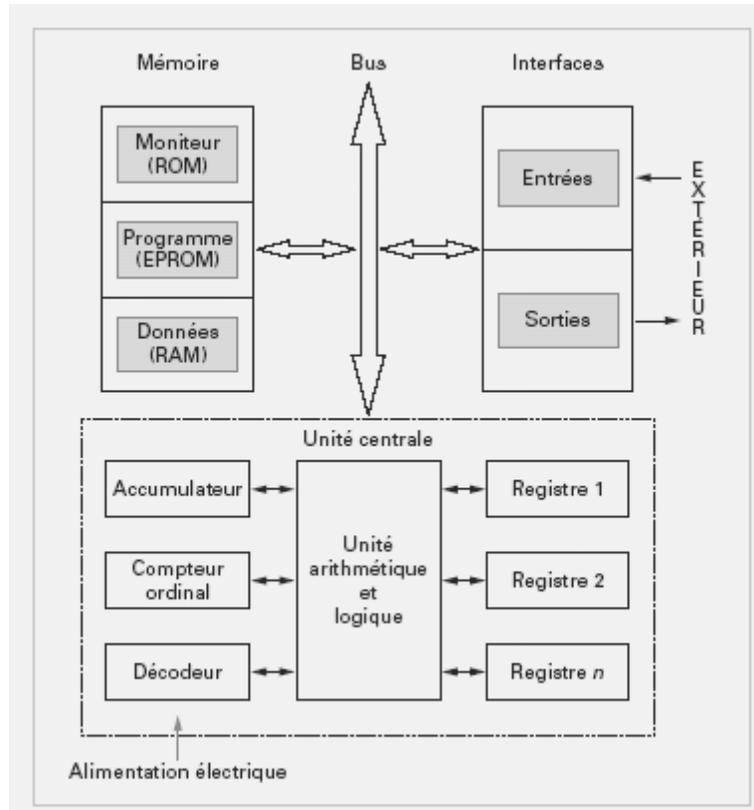


Figure 1.3 : *structure interne d'un API* [18]

Détaillons successivement chacun des composants qui apparaissent sur ce schéma.

I.4.1 Le processeur

Il constitue le cœur de l'appareil dans l'unité centrale ; En fait, un processeur devant être automatisé, se subdivise en une multitude de domaines et processeurs partiels plus petits, liés les uns aux autres.

I.4.2 Les modules d'entrées/sorties

Ils assurent le rôle d'interface entre la CPU et le processus, en récupérant les informations sur l'état de ce dernier et en coordonnant les actions.

Plusieurs types de modules sont disponibles sur le marché selon l'utilisation souhaitée :

- **Modules TOR** : l'information traitée ne peut prendre que deux états (vrai/faux, 0 ou 1 ...)
C'est le type d'information délivrée par une cellule photoélectrique, un bouton poussoir ...etc.

- **Modules analogiques** : l'information traitée est continue et prend une valeur qui évolue dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (débit, niveau, pression, température...etc.).

- **Modules spécialisés** : l'information traitée est contenue dans des mots codés sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent.

I.4.3 Les mémoires

Un système de processeur est accompagné par un ou plusieurs types de mémoires. Elles permettent :

- De stocker le système d'exploitation dans des **ROM** ou **PROM**,
- Le programme dans des **EEPROM**,
- Les données système lors du fonctionnement dans des **RAM**. Cette dernière est généralement secourue par pile ou batterie. On peut, en règle générale, augmenter la capacité mémoire par adjonction de barrettes mémoires type **PCMCIA**.

I.4.4 L'alimentation

Elle assure la distribution d'énergie aux différents modules. L'automate est alimenté généralement par le réseau monophasé 230V-50 Hz mais d'autres alimentations sont possibles (110V ...etc.).

I.4.5 Liaisons de communication

Elles Permettent la communication de l'ensemble des blocs de l'automate et des éventuelles extensions.

Les liaisons s'effectuent :

- avec l'extérieur par des borniers sur lesquels arrivent des câbles transportant le signal électrique ;

- avec l'intérieur par des bus reliant divers éléments, afin de d'échanger des données, des états et des adresses.

I.5 Présentation de la gamme SIMATIC de SIEMENS

Siemens reste le seul à proposer une gamme complète de produits pour l'automatisation industrielle, par le biais de sa gamme *SIMATIC*. L'intégration globale de tout l'environnement d'automatisation est réalisée grâce à :

- Une configuration et une programmation homogène des différentes unités du système.
- Une gestion cohérente des données.
- Une communication globale entre tous les équipements d'automatisme mis en œuvre.

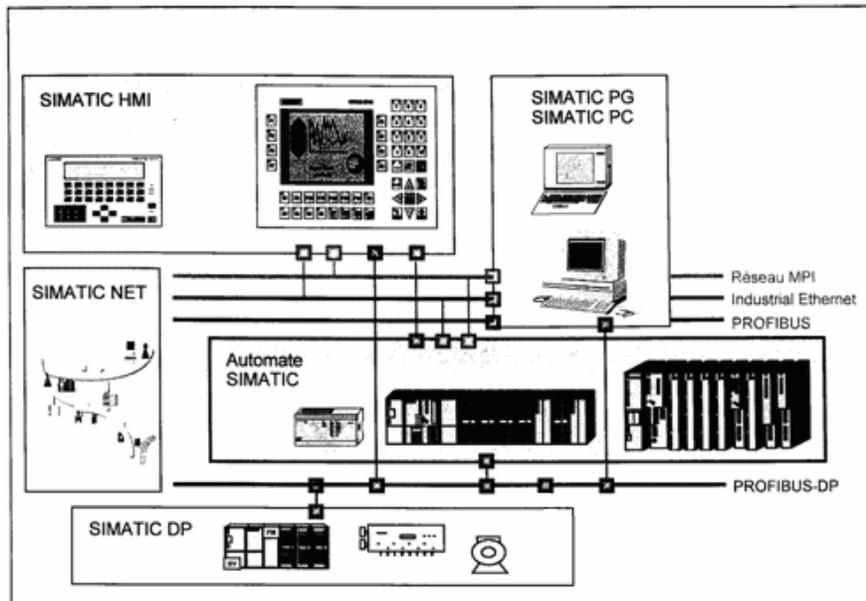


Figure 1.4 : Présentation de la gamme de SIMATIC [18]

I.5.1 Les différentes variantes dans la gamme SIMATIC

a SIMATIC S7

Cette gamme d'automates comporte trois familles :

- **S7 200**, qui est un Micro-automate modulaire pour les applications simples, avec possibilité d'extensions jusqu'à 7 modules, et une mise en réseau par l'interface multipoint (*MPI*) ou *PROFIBUS*.

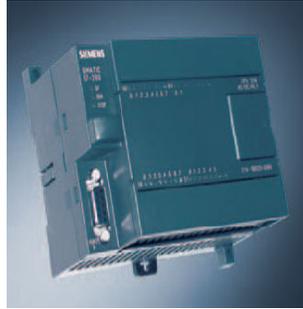


Figure 1.5: *L'API S200* [24]

- **S7300** est un Mini-automate modulaire pour les applications d'entrée et de milieu de gamme, avec possibilité d'extensions jusqu'à 32 modules, et une mise en réseau par l'interface multipoint (*MPI*), *PROFIBUS* et *Industrial Ethernet*



Figure 1.6: *L'API S300* [24]

- **S7400** est un automate de haute performance pour les applications de milieu et haut de gamme, avec possibilité d'extension à plus de 300 modules, et une possibilité de mise en réseau par l'interface multipoint (*MPI*), *PROFIBUS* ou *Industrial Ethernet*.



Figure 1.7: *L'API S400* [24]

b. SIMATIC C7

Le SIMATIC C7 combine automate programmable et panneau opérateur dans une seule unité. L'automate compte la CPU, les modules d'entrées/sorties, et le panneau opérateur qui est utilisé comme une interface Homme/Machine *HMI*.

Le C7 permet la visualisation des états de fonctionnement, des valeurs actuelles du processus et des anomalies.



Figure 1.8 : *La gamme SIMATIC C7* [24]

c. SIMATIC M7

Les SIMATIC M7 sont des calculateurs industriels compatibles PC. Il s'agit d'un système modulaire sous boîtier, construit dans la technique des automates SIMATIC S7. Il peut être intégré dans un automate S7 300/400 ou être utilisé comme système autonome avec une périphérie choisie dans la gamme S7.

Le M7 300/400 est capable d'effectuer simultanément avec une seule CPU des opérations en temps réel, par exemple des algorithmes complexes de commande, de régulation ainsi que des tâches de visualisation et de traitement informatique. Les logiciels sous DOS ou Windows sont exploitables sur le M7-300. Par ailleurs, avec son architecture normalisée PC, il permet une extension programmable et ouverte de la plate-forme d'automatisation S7.



Figure 1.9 : La gamme SIMATIC M7 [24]

I. 5.2 Description du STEP7

STEP 7 est le progiciel de base pour la configuration et la programmation de systèmes d'automatisation SIMATIC S300 et S400. Il fait partie de l'industrie logicielle SIMATIC. Le logiciel de base assiste dans toutes les phases du processus de création de la solution d'automatisation, La conception de l'interface utilisateur du logiciel STEP 7 répond aux connaissances ergonomiques modernes et son apprentissage est très facile [14].

STEP 7 comporte les quatre sous logiciels de base suivants :

a. Gestionnaire de projets *SIMATIC Manager*

SIMATIC Manager constitue l'interface d'accès à la configuration et à la programmation.

Ce gestionnaire de projets présente le programme principal du logiciel STEP7 il gère toutes les données relatives à un projet d'automatisation, quelque soit le système cible (S7/M7/C7) sur lequel elles ont été créées. Le gestionnaire de projets SIMATIC démarre automatiquement les applications requises pour le traitement des données sélectionnées.

b. Editeur de programme et les langages de programmation

Les langages de programmation CONT, LIST et LOG ,font partie intégrante du logiciel de base.

- Le schéma à contacts (CONT) est un langage de programmation graphique. La syntaxe des instructions fait penser aux schémas de circuits électriques. Le langage CONT permet de suivre facilement le trajet du courant entre les barres d'alimentation en passant par les contacts, les éléments complexes et les bobines [2].

- La liste d'instructions (LIST) est un langage de programmation textuel proche de la machine. Dans un programme LIST, les différentes instructions correspondent, dans une large mesure, aux étapes par lesquelles la CPU traite le programme [2].
- Le logigramme (LOG) est un langage de programmation graphique qui utilise les boîtes de l'algèbre de Boole pour représenter les opérations logiques. Les fonctions complexes, comme par exemple les fonctions mathématiques, peuvent être représentées directement combinées avec les boîtes logiques [2].

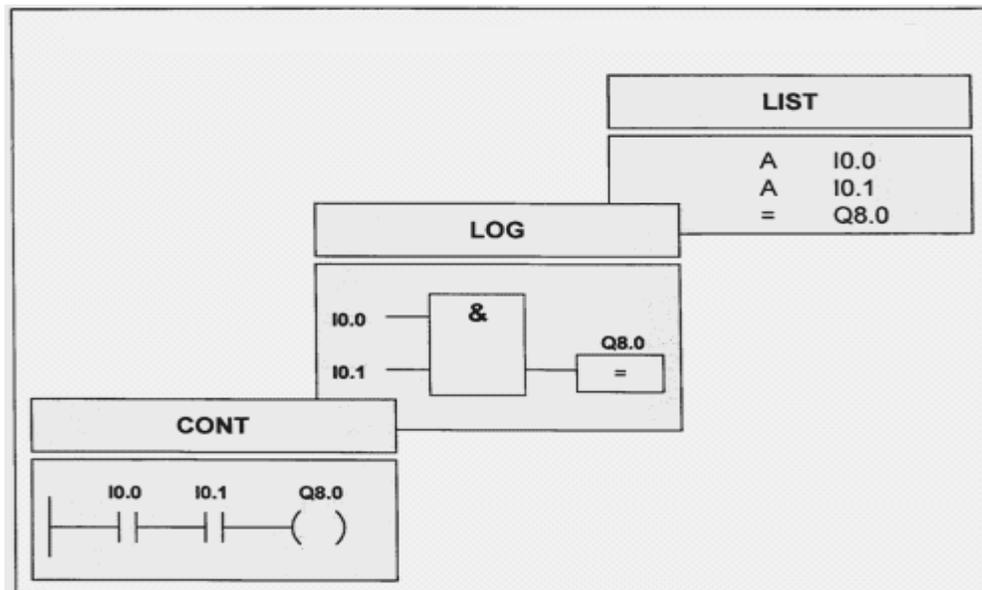


Figure 1. 10 : mode de représentation des langages basiques de programmation STEP 7 [16]

On dispose de langages de programmation plus évolués, au détriment de l'optimisation mémoire:

- **GRAPH** est un langage de programmation permettant la description aisée de commandes séquentielles (programmation de graphes séquentiels). Le déroulement du processus y est subdivisé en étapes. Celles-ci contiennent en particulier des actions pour la commande des sorties. Le passage d'une étape à la suivante est soumis à des conditions de transition [2].
- **HiGraph** est un langage de programmation permettant la description aisée de processus asynchrones non séquentiels sous forme de graphes d'état. A cet effet, l'installation est subdivisée en unités fonctionnelles pouvant prendre différents états. Ces unités fonctionnelles peuvent se synchroniser par l'échange de messages [2].

- **SCL** est un langage évolué textuel. Il comporte des éléments de langage que l'on trouve également sous une forme similaire dans les langages de programmation Pascal et C. SCL convient donc particulièrement aux utilisateurs déjà habitués à se servir d'un langage de programmation évolué. [16].

- **CFC** pour S7 et M7 est un langage de programmation graphique permettant l'interconnexion graphique de fonctions existantes. Ces fonctions couvrent un large éventail allant de combinaisons logiques simples à des régulations et commandes complexes. Un grand nombre de ces fonctions est disponible sous la forme de blocs dans une bibliothèque [16].

c. Paramétrage de l'interface PG-PC

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

d. Le simulateur des programmes PLCSIM

L'application de simulation de modules **S7-PLCSIM** permet d'exécuter et de tester le programme dans un automate programmable (AP) qu'on simule dans un ordinateur ou dans une console de programmation. La simulation étant complètement réalisée au sein du logiciel STEP7, il n'est pas nécessaire qu'une liaison soit établie avec un matériel S7 quelconque (CPU ou module de signaux). L'AP S7 de simulation permet de tester des programmes destinés aux CPU S7-300 et aux CPU S7-400, et de remédier à d'éventuelles erreurs [15].

S7-PLCSIM dispose d'une interface simple permettant de visualiser et de forcer les différents paramètres utilisés par le programme (comme, par exemple, d'activer ou de désactiver des entrées). Tout en exécutant le programme dans l'AP de simulation, on a également la possibilité de mettre en oeuvre les diverses applications du logiciel STEP 7 comme, par exemple, la table des variables (VAT) afin d'y visualiser et d'y forcer des variables.

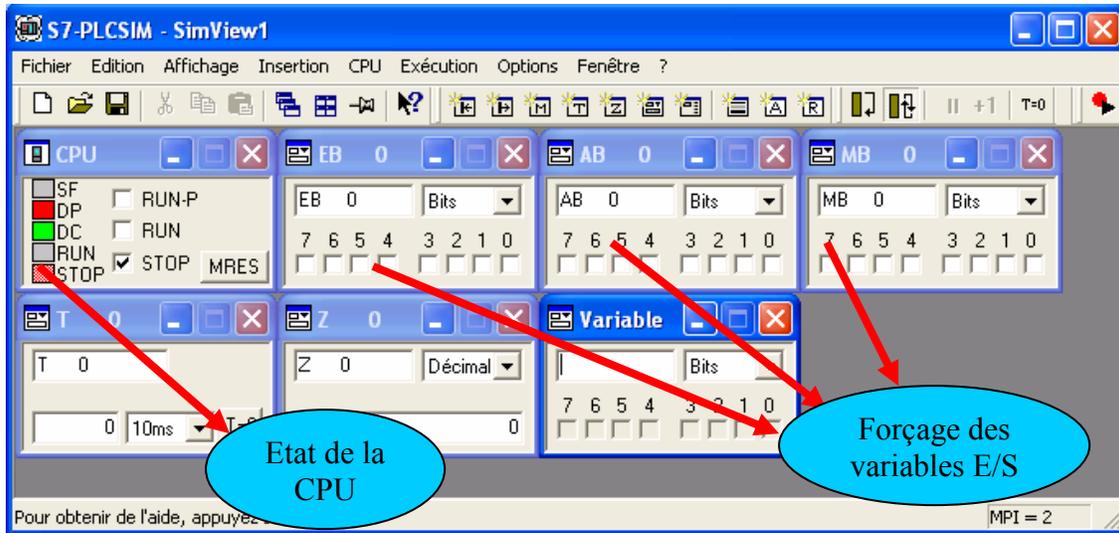


Figure 1.11 : logiciel de simulation PLC-SIM

I.6. Stratégie pour la conception d'une structure programme complète et optimisée

La mise en place d'une solution d'automatisation avec **STEP 7** nécessite la réalisation des tâches fondamentales suivantes :

- *Création du projet SIMATIC Step7*
- *Configuration matérielle HW Config*

Dans une table de configuration, on définit les modules mis en oeuvre dans la solution d'automatisation ainsi que les adresses permettant d'y accéder depuis le programme utilisateur, pouvant en outre, y paramétrer les caractéristiques des modules.

- *Définition des mnémoniques*

Dans une table des mnémoniques, on remplace des adresses par des mnémoniques locales ou globales de désignation plus évocatrice afin de les utiliser dans le programme.

- *Création du programme utilisateur*

En utilisant l'un des langages de programmation mis à disposition, on crée un programme affecté ou non à un module, qu'on enregistre sous forme de blocs, de sources ou de diagrammes.

- ***Exploitation des données:***

Création des données de références : Utiliser ces données de référence afin de faciliter le test et la modification du programme utilisateur et la configuration des variables pour le "contrôle-commande"

- ***Test du programme et détection d'erreurs***

Pour effectuer un test, on a la possibilité d'afficher les valeurs de variables depuis le programme utilisateur ou depuis une CPU, d'affecter des valeurs à ces variables et de créer une table des variables qu'on souhaite afficher ou forcer.

- ***Chargement du programme dans le système cible***

Une fois la configuration, le paramétrage et la création du programme terminés, on peut transférer le programme utilisateur complet ou des blocs individuels dans le système cible (module programmable de votre solution matérielle). La CPU contient déjà le système d'exploitation.

- ***Surveillance du fonctionnement et diagnostic du matériel***

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

I.7 Exemple de Création et d'édition d'un projet S7

Pour bien illustrer les démarches de conception d'un programme dont on a souligné précédemment les étapes, on prend comme exemple d'application l'automatisation d'une chaîne de remplissage de bouteilles, avec le cahier de charge suivant :

Cahier de charge :

Programmer la gestion des modes fonctionnement sur une installation de remplissage d'après les indications suivantes :

- L'installation est mise sous tension avec l'entrée **E0.0**, un témoin s'allume à la sortie **A20.4**, et elle est mise hors tension à l'aide de l'entrée **E0.1**.
- Lorsque l'installation est sous tension, le mode de fonctionnement est sélectionné ainsi :

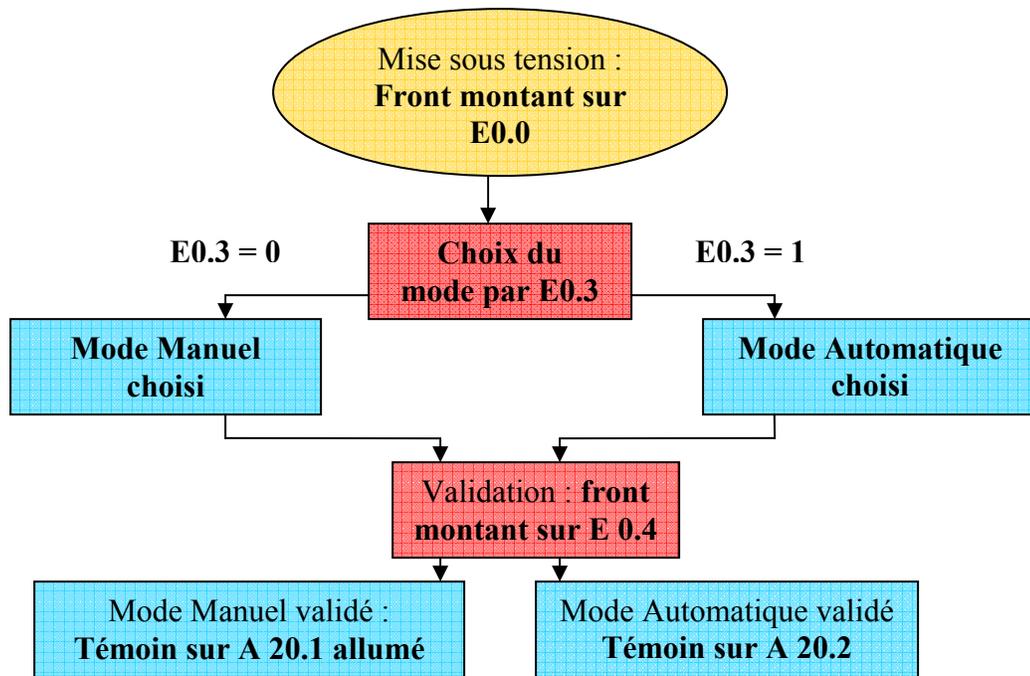


Figure 1.12 : Choix du mode de fonctionnement dans la première application

- En mode manuel :

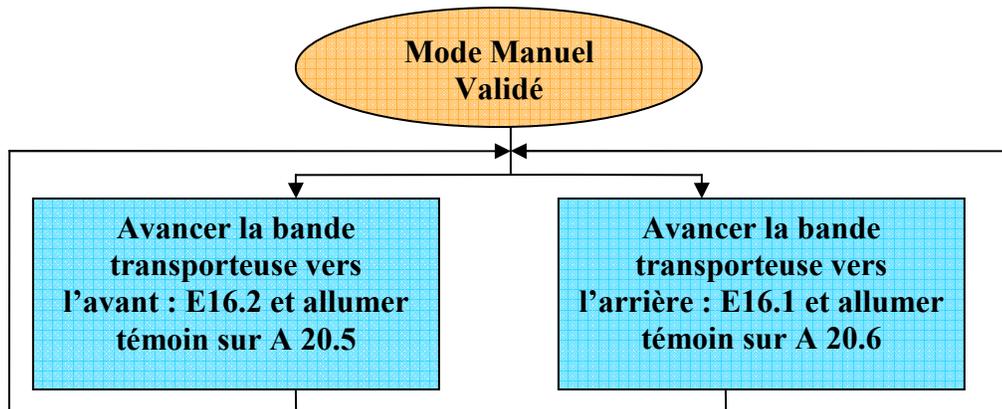


Figure 1.13 : Le fonctionnement en mode manuel dans la première application

- En mode automatique,

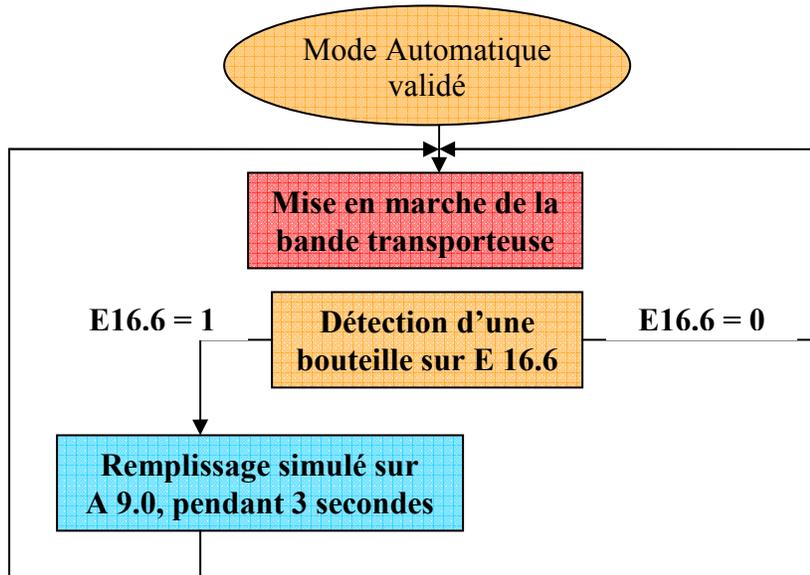


Figure 1.14 : *Le fonctionnement en mode automatique dans la première application*

- En Cas de changement du mode de fonctionnement ou de mise hors tension de l'installation, le mode sélectionné préalablement est désactivé.
- Le nombre de bouteille remplie sera compté et affiché pour l'utilisateur, grâce à un afficheur sept segments muni d'un décodeur BCD, et adressé sur **AW 12**.

| Entrées (Boutons/ capteurs) | Adresse absolue |
|---|------------------------|
| Mise sous tension | E 0.0 |
| Mise hors tension | E 0.1 |
| Mode Manuel/Automatique | E 0.3 |
| Validation du choix de mode | E 0.4 |
| Marche avant par impulsions (Manuel) | E 16.2 |
| Marche arrière par impulsions (Manuel) | E 16.1 |
| Détecteur des bouteilles | E 16.6 |
| Témoin de remplissage | A9.0 |
| Témoin du mode Manuel | A 20.1 |
| Témoin du mode automatique | A 20.2 |
| Témoin de mise sous tension de l'installation | A 20.4 |
| Témoin de marche avant avec impulsions | A 20.5 |
| Témoin de marche arrière avec impulsions | A 20.6 |
| L'afficheur 7 segments (avec décodeur BCD) | AW 12 |

Tableau I.1 : *L'adressage absolu des entrées/sorties de la première application*

La **Figure I.12** schématise le dispositif à automatiser :

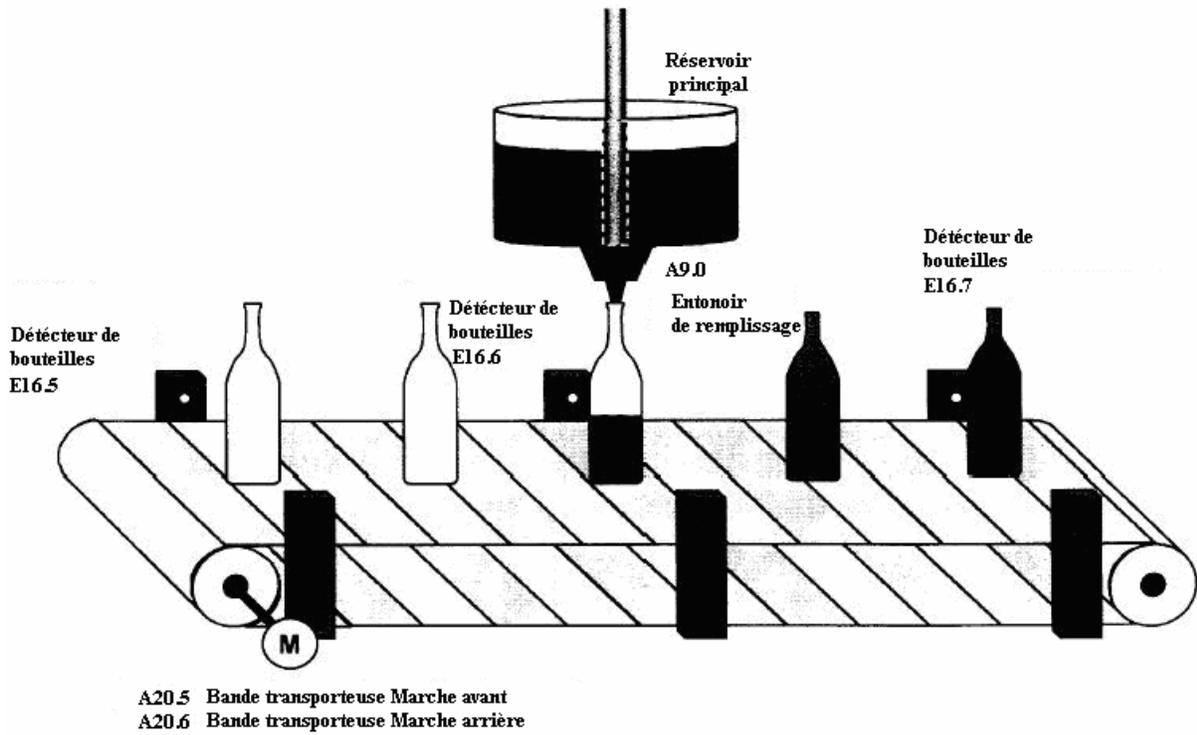
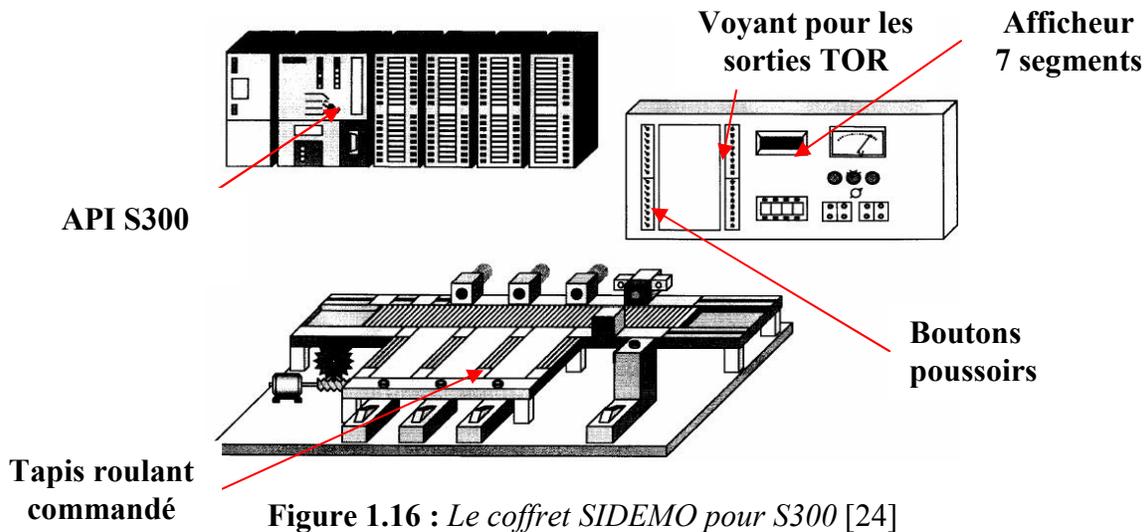


Figure I.15 : *La chaîne de remplissage à automatisé*

Pour la simulation on utilise le coffret de démonstration **SIDEMO** de la figure suivante :





Il inclue un automate S300, avec une CPU315-2DP et des modules E/S qui seront cités plus tard dans la configuration du matériel, et comporte aussi une interface E/S analogique et numérique pour ses modules, ainsi qu'un afficheur sept segments.

I.7.1 Création du projet dans SIMATIC Manager

Afin de créer un nouveau projet STEP7, il nous est possible d'utiliser « l'assistance de création de projet », ou bien créer le projet soi même et le configurer directement, cette dernière est un peu plus complexe, mais nous permet aisément de gérer notre projet.

En sélectionnant l'icône **SIMATIC Manager**, on aura la fenêtre principale qui s'affiche, pour sélectionner un nouveau projet et valider.

Comme le projet est vide il nous faut insérer une station SIMATIC 300/400.

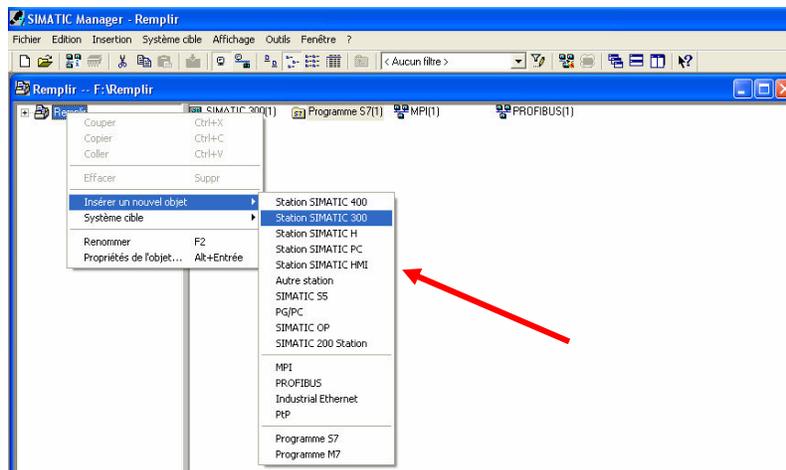


Figure 1.17: Le SIMATIC Manager

Deux approches sont possibles. Soit on commence par la création du programme puis la configuration matérielle ou bien l'inverse.

I.7.2 Configuration matérielle (Partie Hardware)

C'est une étape importante, qui correspond à l'agencement des châssis, des modules et de la périphérie décentralisée.

Les modules sont fournis avec des paramètres définis par défaut en usine. Une configuration matérielle est nécessaire pour :

- modifier les paramètres ou les adresses pré-réglés d'un module,
- configurer les liaisons de communication.

Le choix du matériel SIMATIC S300 avec une CPU315-2DP nous conduit à introduire hiérarchisée suivante :

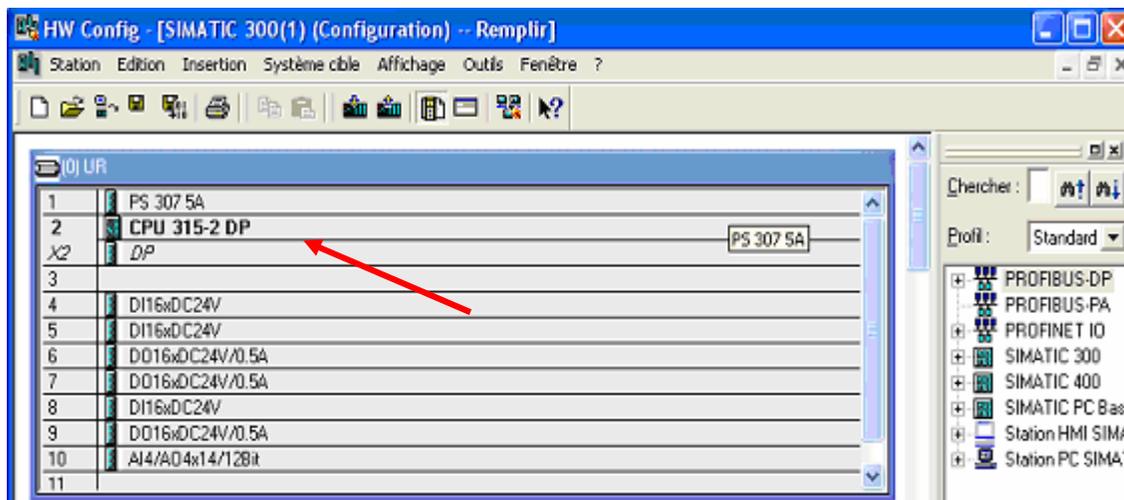


Figure 1.18 : Configuration matérielle

On commence par le choix du châssis selon la station choisie auparavant, Pour la station SIMATIC S300, on aura le châssis « RACK-300 » qui comprend un rail profilé.

Sur ce profilé, l'alimentation préalablement sélectionnée se trouve dans l'emplacement n°1. Parmi celles proposées notre choix s'est porté sur la « PS-307 5A ».

La « CPU 315-2DP » est impérativement mise à l'emplacement n°2.

L'emplacement n°3 est réservé comme adresse logique pour un coupleur dans une configuration multi châssis.

A partir de l'emplacement 4, il est possible de monter au choix jusqu'à 8 modules de signaux (*SM*), processeurs de communication (*CP*) ou modules fonctionnels (*FM*). Après cela il ne nous reste qu'à enregistrer et compiler.

La configuration matérielle étant terminée, un dossier « Programme S7 » est automatiquement inséré dans le projet, comme indiqué dans la figure suivante :

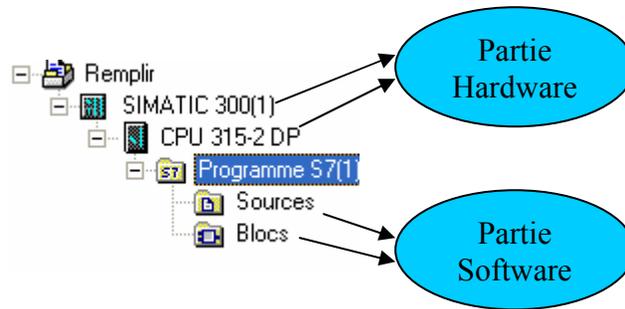


Figure 1.19 : Création du programme S7

I.7.3 Création de la table des mnémoniques (Partie Software)

Dans tout programme il faut définir la liste des variables qui vont être utilisées lors de notre programmation. Pour cela la table des mnémoniques est créée. L'utilisation des noms appropriés rend le programme plus compréhensible et plus facile à manipuler. Ce type d'adressage est appelé « *relatif* » (voir Figure I.20)

Pour créer cette table, on suit le cheminement suivant :

Insérer nouvel objet → table des mnémoniques

On édite la table des mnémoniques en respectant notre cahier de charge, pour les entrées et les sorties.

| | Etat | Mnémonique / | Opérande | Type de d | Commentaire |
|----|------|----------------------|----------|-----------|--------------------|
| 3 | | Bande marche AR | A 20.6 | BOOL | |
| 4 | | Bande marche AV | A 20.5 | BOOL | |
| 5 | | Comptage | DB 1 | DB 1 | |
| 6 | | Comptage des bo... | FC 18 | FC 18 | |
| 7 | | CONT_C | FB 41 | FB 41 | Continuous Control |
| 8 | | Désact_Etapes_M... | M 0.6 | BOOL | |
| 9 | | Détecteur B pleine | E 16.7 | BOOL | |
| 10 | | Détecteur B sous ... | E 16.6 | BOOL | |
| 11 | | Détecteur B vide | E 16.5 | BOOL | |
| 12 | | Fonct_Manuel | FC 16 | FC 16 | |
| 13 | | G7_STD_1 | FC 70 | FC 70 | |
| 14 | | G7_STD_3 | FC 72 | FC 72 | |
| 15 | | Initi_Mode_auto_... | M 0.7 | BOOL | |
| 16 | | Manual/Automati... | E 0.3 | BOOL | |
| 17 | | Marche | M 0.0 | BOOL | |
| 18 | | Marche par impul... | E 16.1 | BOOL | |
| 19 | | Marche par impul... | E 16.2 | BOOL | |
| 20 | | Mode_Auto_activé | M 0.1 | BOOL | |
| 21 | | Mode_manuel_ac... | M 0.2 | BOOL | |

Figure 1.20 : Table de mnémonique relative à chaîne de remplissage

I.7.4 Elaboration du programme S7 (Partie Software)

Le dossier **bloc**, cité auparavant, contient les blocs que l'on doit charger dans la CPU pour réaliser la tâche d'automatisation, il englobe :

- les blocs de code (**OB**, **FB**, **SFB**, **FC**, **SFC**) qui *contiennent les programmes*,
- les blocs de données **DB d'instance** et **DB globaux** qui *contiennent les paramètres du programme*.

a. Les blocs d'organisation (OB)

Les **OB** sont appelés par le système d'exploitation, on distingue plusieurs types :

- ceux qui gèrent le traitement de programmes cycliques
- ceux qui sont déclenchés par un événement,
- ceux qui gèrent le comportement à la mise en route de l'automate programmable
- et en fin, ceux qui traitent les erreurs [16].

Le bloc **OB1** est généré automatiquement lors de la création d'un projet. C'est le programme cyclique appelé par le système d'exploitation.

b. Les blocs fonctionnels (FB), (SFB)

Le **FB** est un sous programme écrit par l'utilisateur et exécuté par des blocs de code. On lui associe un bloc de données d'instance relatif à sa mémoire et contenant ses paramètres.

Les **SFB** système sont utilisés pour des fonctions spéciales intégrées dans la **CPU** [16].

c. Les fonction (FC), (SFC)

La **FC** contient des routines pour les fonctions fréquemment utilisées. Elle est sans mémoire et sauvegarde ses variables temporaires dans la pile de données locales. Cependant elle peut faire appel à des blocs de données globaux pour la sauvegarde de ses données [16].

Les **SFC** sont utilisées pour des fonctions spéciales, intégrées dans la CPU S7, elle est appelée à partir du programme.

Dans notre exemple on a utilisé le **FC15**, **FC16**, **FC18** pour les routines, et **FB17** pour la partie séquentielle. Systématiquement et automatiquement, les **SFC** correspondantes ont été créés.

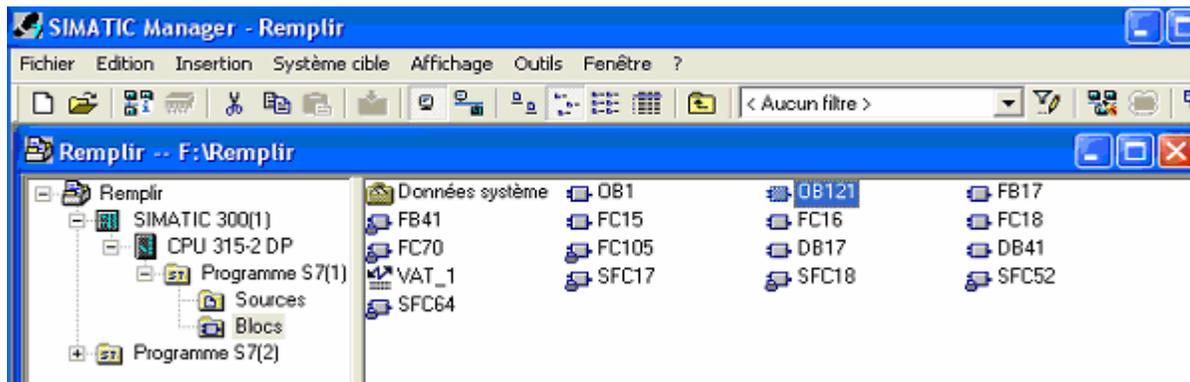


Figure 1.21 : Vue d'ensemble des blocs programme dans SIMATIC Manager

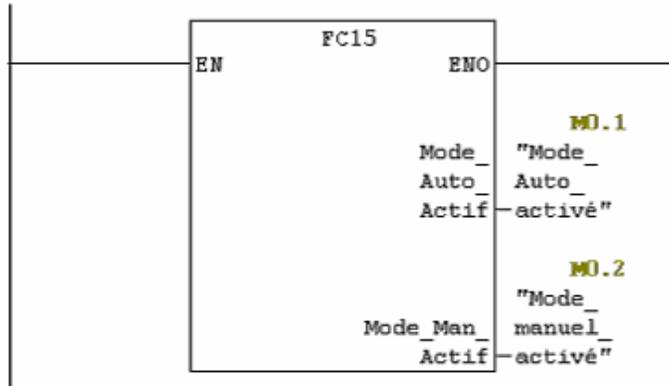
Une fois les mnémoniques et les blocs déclarés, on est prêt pour écrire le programme.

Dans un but didactique, on écrira les programmes en faisant varier les langages (**LOG**, **CONT**, **LIST** et **GRAPH**) :

- **OB1** : édité en **CONT**, dans notre exemple l'**OB1** fait appel à la routine **FC15**, **FC16**, **FB17** et **FC18** comme il est montré si dessous :

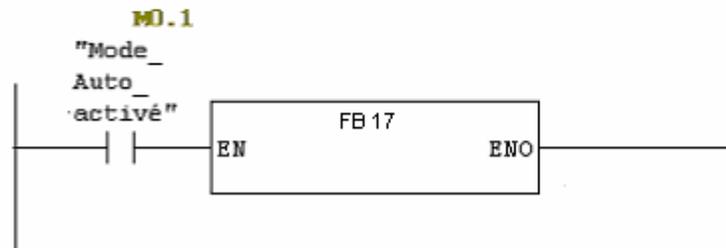
Réseau 1 : mode de fonctionnement

appel du bloc FC15 pour exécuter le programme du mode de fonctionnement et la sélection des modes.



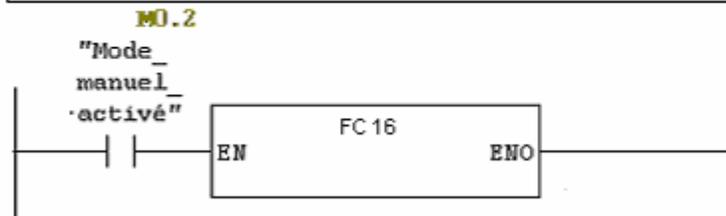
Réseau 2 : MODE AUTOMATIQUE

Appel du bloc fonctionnel FB17, pour le MODE AUTOMATIQUE à condition que se dernier soit activé par le biais du M0.1



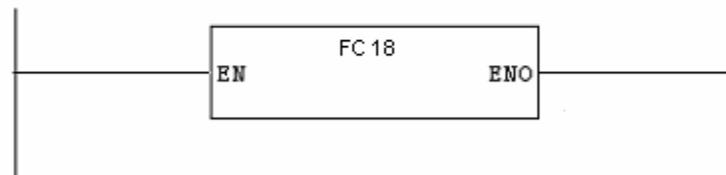
Réseau 3 : MODE MANUEL

Appel de la fonction FC16, pour le "MODE MANUEL", activé par M0.2



Réseau 4 : Comptage des bouteilles

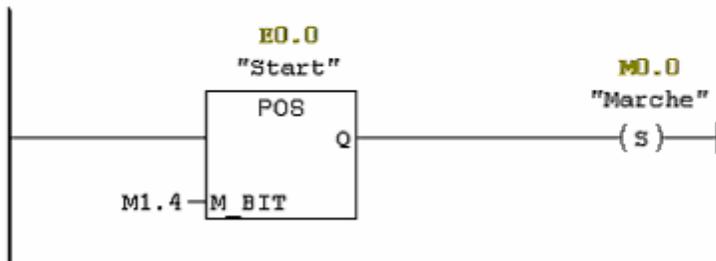
Appel de la fonction de comptage FC18, activé indépendamment du mode de fonctionnement



- **FC15** : édité en **CONT** et en **LIST**, cette fonction sert à la gestion du processus et le choix du mode de fonctionnement comme suit :

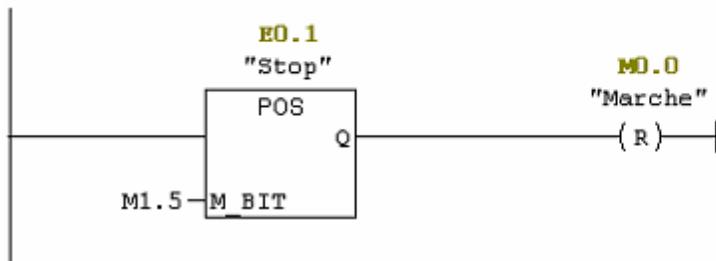
Réseau 1 : Mise en marche

en mettant l'entrée EO.0 à 1 le moteur se met en marche et mettre à 1 le memento MO.0



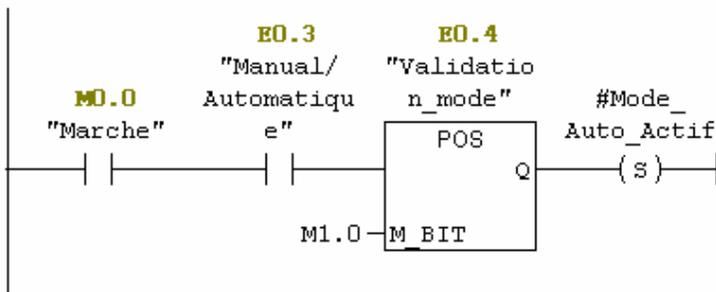
Réseau 2 : Mise à l'arrêt

en mettant l'entrée EO.1 à 1 le moteur s'arrête, et mettre à 0 le memento MO.0



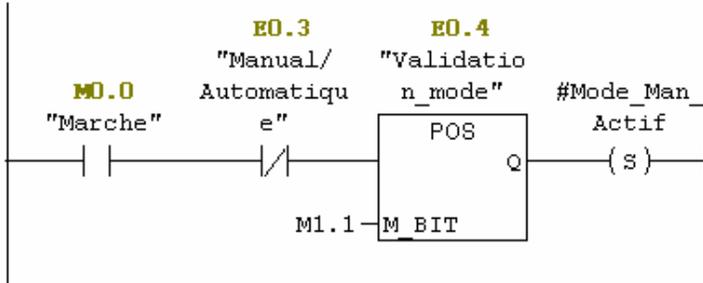
Réseau 3 : mode automatique actif

étant MO.0=1 (moteur marche) et EO.3 à 1 on active par une impulsion sur EO.4; le mode automatique.



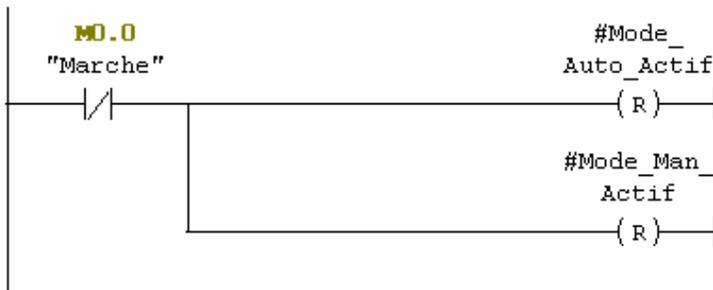
Réseau 4 : mode manuel actif

étant le moteur en marche et EO.3=0 et validation par impulsion sur EO.4 on active le mode manuel.



Réseau 5 : mise a l'arrêt

désactiver les deux modes par mise à l'arrêt du moteur



Réseau 6 : désactivation du mode manuel

lors des changement de modes, inversion de l'état de l'entrée EO.3, le mode en cours se désactive et le moteur s'arrête.

```

U   "Manual/Automatique"      E0.3
BLD 100
FP   M      1.2
R   #Mode_Man_Actif
R   "Bande marche AR"        A20.6
R   "Bande marche AV"        A20.5
    
```

Réseau 7 : désactivation du mode automatique

lors des changement de modes, inversion de l'état de l'entrée EO.3, le mode en cours se désactive et le moteur s'arrête.

```

U   "Manual/Automatique"      E0.3
BLD 100
FN   M      1.3
R   #Mode_Auto_Actif
R   "Bande marche AR"        A20.6
R   "Bande marche AV"        A20.5
    
```

Réseau 8 : témoin lumineux pour mode automatique actif

en mode automatique, la sortie A20.2 se met à 1 et le témoin s'allume

```
U    #Mode_Auto_Actif
=    "Témoin_mode_AUTO"      A20.2
```

Réseau 9 : témoin lumineux pour mode manuel actif

en mode manuel, la sortie A20.1 se met à 1 et le témoin s'allume

```
U    #Mode_Man_Actif
=    "Témoin_mode_MANUEL"    A20.1
```

Réseau 10 : témoin lumineux pour moteur en marche

le moteur étant en marche, la sortie A20.4 se met à 1 et le témoin s'allume

```
U    "Marche"
=    "Témoin_marche"        M0.0
                                A20.4
```

- **FC16** : édité en **LIST**, cette fonction fait la gestion du processus en mode « **Manuel** » comme suit :

Réseau 1 : marche avant

E16.2=1 et verifier bien que E16.1=0 la sortie A20.5=0 la bande transporteuse marche en avant

```
U    "Marche par impulsion AV"  E16.2
UN   "Marche par impulsion AR"  E16.1
=    "Bande marche AV"          A20.5
```

Réseau 2 : marche arrière

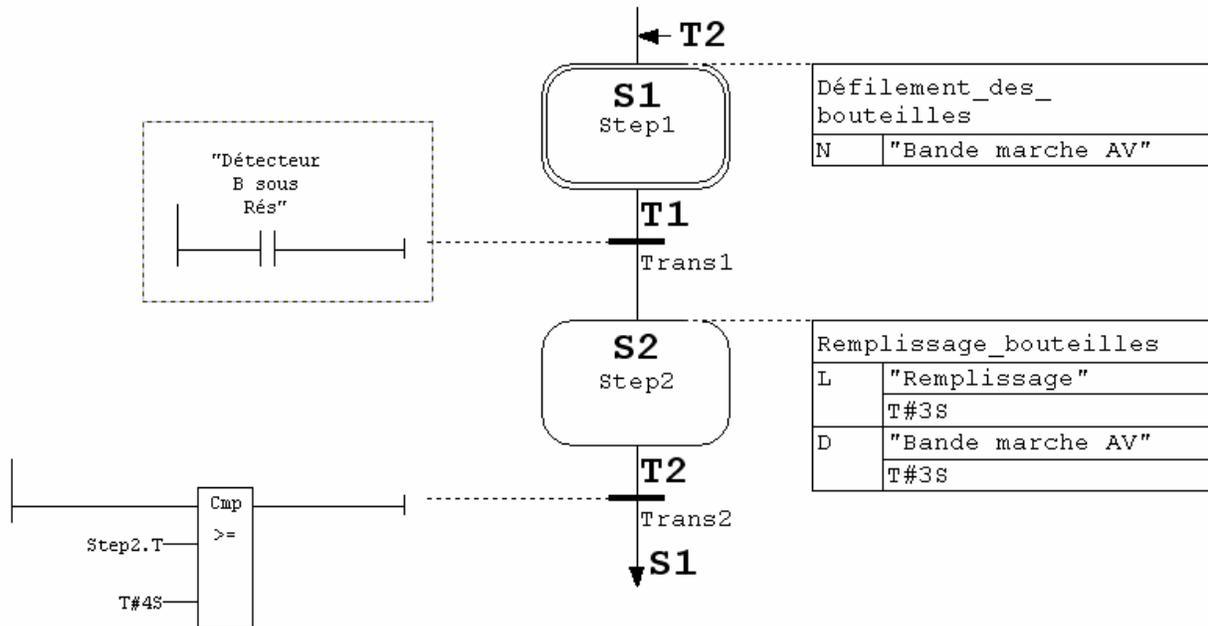
E16.1=1 et verifier bien que E16.2=0 la sortie A20.6=0 la bande transporteuse marche en arrière

```
U    "Marche par impulsion AR"  E16.1
UN   "Marche par impulsion AV"  E16.2
=    "Bande marche AR"          A20.6
```

- **FBI7** : pour le remplissage de bouteilles en « **MODE AUTOMATIQUE** », et édité par le **GRAPH** qui est plus pratique pour les systèmes séquentiels.

La bande transporteuse est en marche, dès la détection d'une bouteille. Dès la détection d'une bouteille vide par capteur le capteur mis à l'entrée E16.6, la bande s'arrête, effectue le

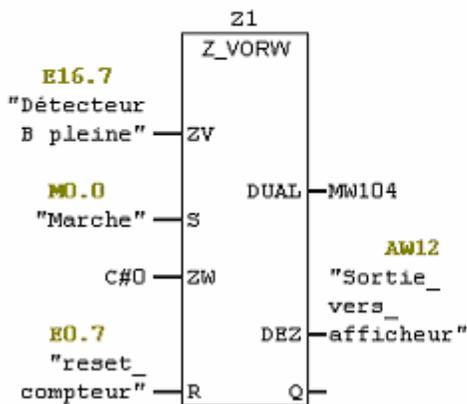
remplissage durant 3 secondes, et continue son avancement ; comme il est montré sur le schéma qui suit.



- **FC18** : édité en LOG, et c'est le langage qu'on a, spécialement, trouvé plus adapté au profil « automaticien ».

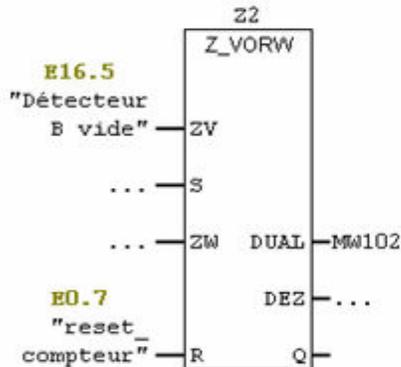
Réseau 1 : le nombre de bouteilles pleines

tant que le moteur est en marche et il y a detection de bouteilles pleines à la sortie par mise à 1 de l'entrée E16.7, le résultat dans MW104 l'affichage sur AW12



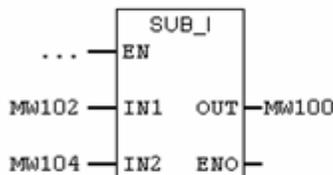
Réseau 2 : comptages des bouteilles vides

dés qu'il y a détection de bouteilles vides à l'entrée par mise à 1 de l'entrée E16.5, comptage et résultat dans dans le mémonto MW102



Réseau 3 : comptage des bouteilles cassées

le nombre de bouteilles cassées est le nombre de bouteilles vides auquel on soustrait le nombre de bouteilles remplies, le résultat est dans le mémonto MW100



Le programme étant écrit, on doit le charger dans la CPU.

I.7.5 Chargement dans le système cible à partir de la PG/PC

Le chargement du programme dans le système cible se fait sous certaines conditions :

- Une liaison est établie entre la PG et la CPU du système cible (via l'interface *MPI* ou *PROFIBUS*).
- L'accès au système cible est possible.
- La compilation du programme à charger doit se faire sans erreur.
- La CPU doit se trouver dans un état de fonctionnement autorisant le chargement (STOP ou RUN-P).

Si la syntaxe est correcte, le bloc est ensuite compilé en code machine, enregistré et chargé par la commande **Système cible > Charger**, ou en sélectionnant tous les blocs et charger par la

fonction de chargement  dans la barre d'outils du **SIMATIC Manager**.

I.7.6 Surveillance du fonctionnement et diagnostic du matériel

Pour le test du bon fonctionnement du programme, le logiciel optionnel de simulation permet d'exécuter et de tester le programme dans un système d'automatisation qu'on simule dans l'ordinateur ou dans la console de programmation en y accédant par la fonction  dans la barre d'outils.

Pour la recherche d'erreurs, des icônes de diagnostic permettent de déceler des défauts sur un module et indiquent l'état de ce dernier. Pour une CPU, par exemple, son état de fonctionnement (RUN, STOP ou RUN-P).

Les icônes de diagnostic s'affichent dans la vue en ligne de la fenêtre du projet, dans la vue rapide (présélection) ou encore dans la vue de diagnostic lors d'un appel de la fonction **"Diagnostic du matériel"**.

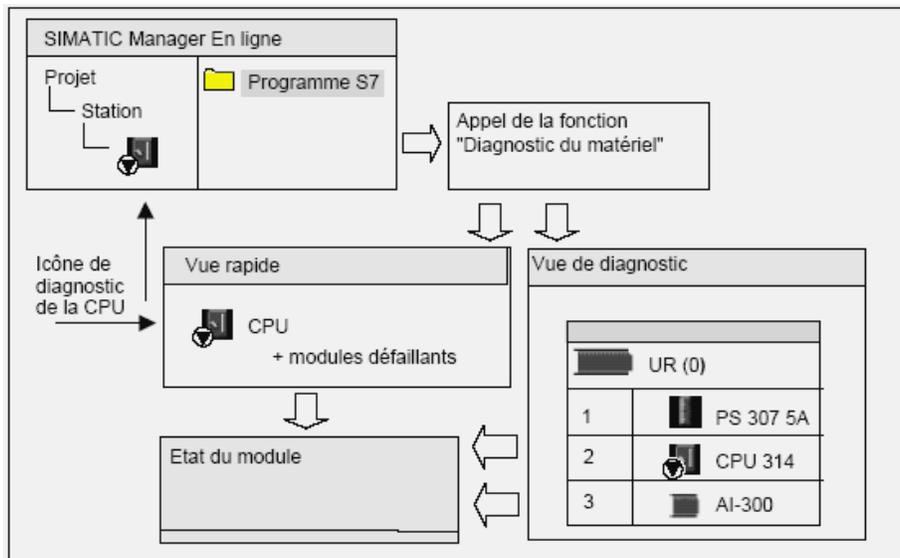


Figure 1.22 : Diagnostique du matériel [16]

Dans la plupart des cahiers de charge, nous avons une régulation à faire. Voyons comment réaliser cette fonction.

I.8 Régulation PID des processus à base d'automates SIMATIC S7

I.8.1 Principe de régulation

La grandeur à réguler est mesurée de façon continue et est ensuite comparée avec une valeur consigne donnée en entrée du système. La régulation devra, en fonction du résultat de la comparaison, rapprocher la valeur à réguler de la consigne [14].

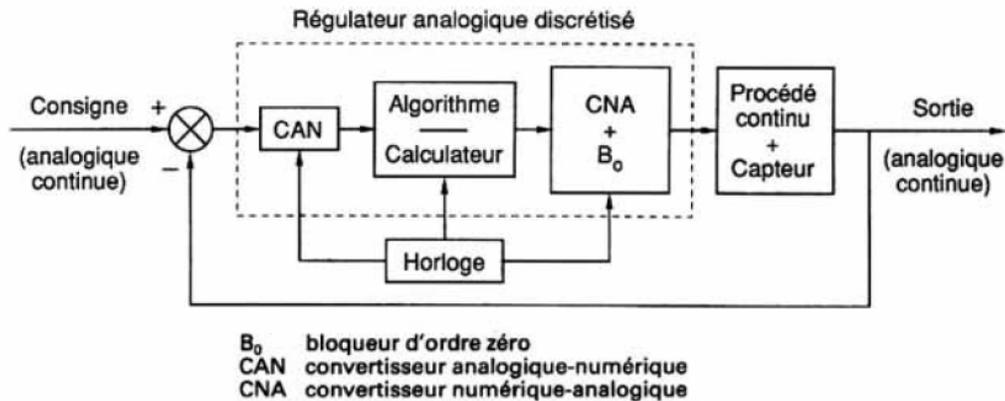


Figure I.23 : Composition d'une boucle de régulation numérique

Le régulateur est l'élément central d'un asservissement. Il évalue l'erreur calculée par le comparateur, c'est à dire l'écart entre la sortie et la consigne. Il en déduit une valeur de correction à transmettre au procédé afin de corriger la sortie.

La manière (l'algorithme) dont le régulateur calcule la valeur de correction (Commande) à partir de l'erreur est la principale activité de l'asservissement.

I.8.2 Concept de la fonction de régulation PID dans SIMATIC S7

La fonction « Régulation PID » se compose des éléments suivants :

- Les blocs fonctionnels avec deux variantes :
 - o **CONT_C continu** : pour un signal de commande continue. Ce bloc sera détaillé par la suite.
 - o **PULSGEN impulsion** : pour un signal de commande modulé en largeur d'impulsion.
- L'interface de paramétrage pour la configuration des blocs fonctionnels.

On peut utiliser le régulateur PID comme régulateur de maintien autonome mais aussi comme régulateur en cascade, de mélange ou de rapport dans des régulations à plusieurs boucles. Sa méthode de travail se base sur l'algorithme PID du régulateur à échantillonnage à sortie analogique, complété le cas échéant par un étage conformateur d'impulsions assurant la formation des signaux de sortie modulés en largeur d'impulsions

I.8.3 Le bloc FB 41 « CONT_C »

Cette variante du régulateur PID, comme déjà précisé, utilisé en tant que régulateur continu. Son rôle sera de calculer une valeur d'ajustement Y_r en fonction de l'erreur (différence mesure/consigne) $e = w - x$, selon l'algorithme d'un régulateur P.I.D, et de livrer cette grandeur d'ajustement y sur sa sortie analogique.

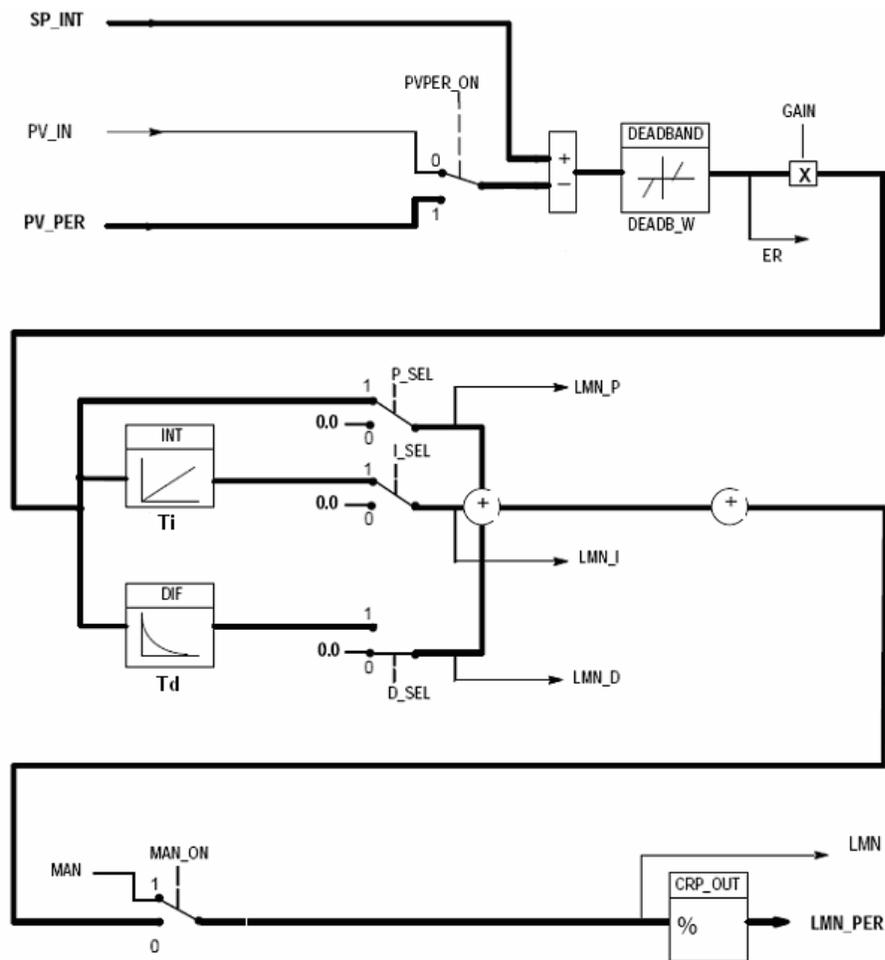


Figure I.24 : L'algorithme du bloc FB41 [14]

Le **FB 41** propose en plus d'une régulation purement logicielle, des fonctions de traitement de la consigne, de la mesure et la grandeur de réglage calculée.

Ainsi la configuration du bloc, nécessite l'ajustement d'un grand nombre de paramètres et de sous-fonctions, dont nous présentons ci-dessous le détail :

- **Les paramètres et les grandeurs d'entrées :**

| Paramètres d'entées | Types de données | Description |
|---------------------|------------------|--|
| MAN_ON | BOOL | MANUAL VALUE ON / Activation du mode manuel Quand cette entrée est à 1, la boucle de régulation est interrompue. La valeur de réglage manuelle est sortie comme grandeur de réglage. |
| PVPER_ON | BOOL | PROCESS VARIABLE PERIPHERY ON / Activation de la mesure de périphérie Pour que la mesure soit lue en périphérie, il faut relier l'entrée PV_PER à la périphérie et mettre à 1 l'entrée PVPER_ON. |
| P_SEL | BOOL | PROPORTIONAL ACTION ON / Activation de l'action proportionnelle |
| I_SEL | BOOL | INTEGRAL ACTION ON / Activation de l'action par intégration |
| D_SEL | BOOL | DERIVATIVE ACTION ON / Activation de l'action par dérivation |
| CYCLE | TEMPS | SAMPLE TIME / Période d'échantillonnage Le temps s'écoulant entre les appels de bloc doit être constant. Il est indiqué au niveau de cette entrée. |
| SP_INT | REEL | INTERNAL SETPOINT / Consigne interne Cette entrée sert à introduire une valeur de consigne. |
| PV_IN | REEL | PROCESS VARIABLE IN / Mesure d'entrée. Cette entrée permet de paramétrer une valeur de mise en service ou d'appliquer une mesure externe en virgule flottante. |
| PV_PER | MOT | PROCESS VARIABLE PERIPHERIE / Mesure de périphérie. La mesure en format de périphérie est appliquée au régulateur par cette entrée. |
| MAN | REEL | MANUAL VALUE / Valeur de réglage manuelle Cette entrée sert à introduire une valeur de réglage manuelle moyennant des fonctions de contrôle-commande. |
| GAIN | REEL | PROPORTIONAL GAIN / Coefficient d'action proportionnelle |

| | | |
|---------|-------|--|
| | | |
| TI | TEMPS | RESET TIME / Temps d'intégration.. |
| TD | TEMPS | DERIVATIVE TIME / Temps de dérivation. |
| DEADB_W | REEL | DEAD BAND WIDTH / Largeur de zone morte. Le signal d'erreur traverse une zone morte. Cette entrée détermine la taille de la zone morte pour l'erreur de réglage. |

Tableau I.2 : Les paramètres d'entrée du bloc fonctionnel FB 41 [14]

- Les paramètres de sorties

| Paramètres de sorties | Types de données | Description |
|-----------------------|------------------|---|
| LMN | REEL | MANIPULATED VALUE / Valeur de réglage. Cette sortie donne en virgule flottante la valeur de réglage agissant réellement. |
| LMN_PER | MOT | MANIPULATED VALUE PERIPHERY / Valeur de réglage de périphérie. Cette sortie fournit la valeur de réglage en format de périphérie. |
| LMN_P | REEL | PROPORTIONALITY COMPONENT / Composante P. Cette sortie contient la composante proportionnelle de la grandeur de réglage. |
| LMN_I | REEL | INTEGRAL COMPONENT / Composante I. Cette sortie contient la composante intégrale de la grandeur de réglage. |
| LMN_D | REEL | DERIVATIVE COMPONENT / Composante D. Cette sortie contient la composante différentielle de la grandeur de réglage. |

Tableau I.3 : Les paramètres de sortie du bloc fonctionnel FB 41 [14]

Pour une CPU donnée, il faut trouver un compromis entre le nombre de régulateurs et la fréquence de traitement de chacun d'eux. Plus les boucles de régulation seront rapides, c'est-à-dire plus souvent les grandeurs réglantes sont à calculer par unité de temps, plus faible sera le nombre de régulateurs.

Il n'y a pas de restriction concernant le type de processus à mettre en oeuvre. Les blocs fonctionnels s'appliquent aux systèmes inertiels (températures, niveaux de remplissages, etc.) comme aux systèmes très rapides (débit, vitesse de rotation, etc.).

I.8.4 Paramétrage du régulateur PID du type continue FB 41 avec STEP 7

Pour l'utilisation et l'optimisation du fonctionnement du régulateur PID **FB**, on fait appel à l'outil « *Paramétrage de la régulation PID* », les paramètres choisis seront sauvegardés dans le DB d'instance local associé au bloc **FB41**.

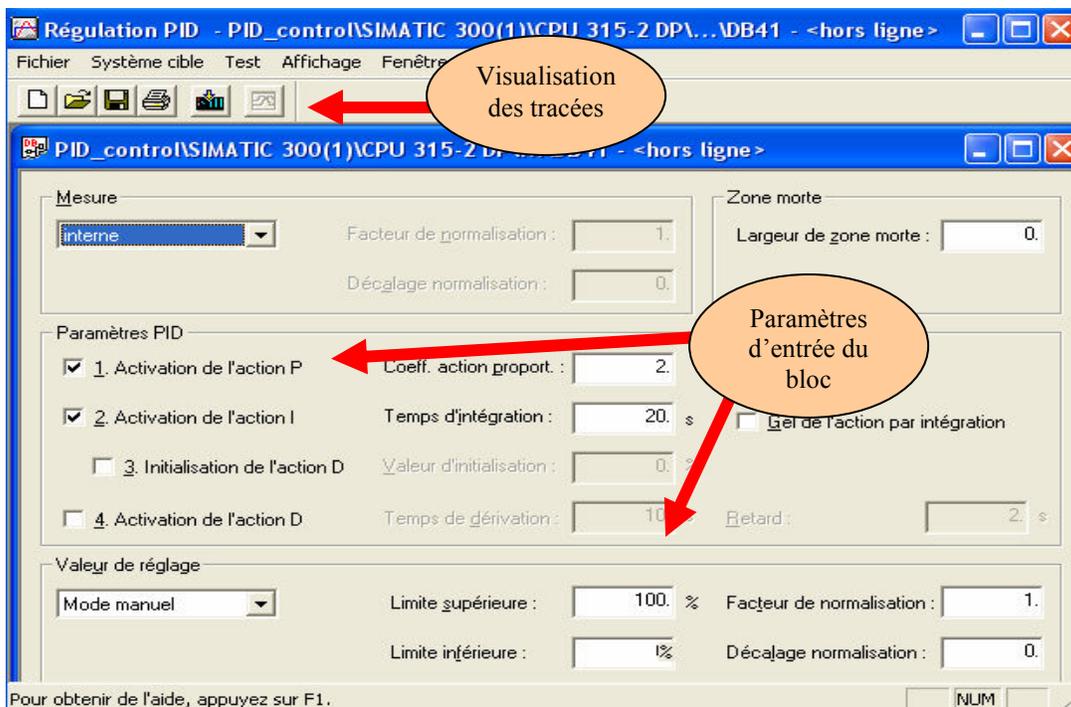


Figure I.25 : Paramétrage du régulateur PID

Une fois le bloc de données paramétré, sauvegardé et chargé dans la CPU, c'est alors que peut intervenir la mise en route de la représentation graphique pour observer le comportement de la régulation.

Les courbes de la valeur souhaitée (consigne), la valeur réelle (mesure) et la grandeur d'ajustement (commande) peuvent être visualisées en temps réel grâce au graphique.

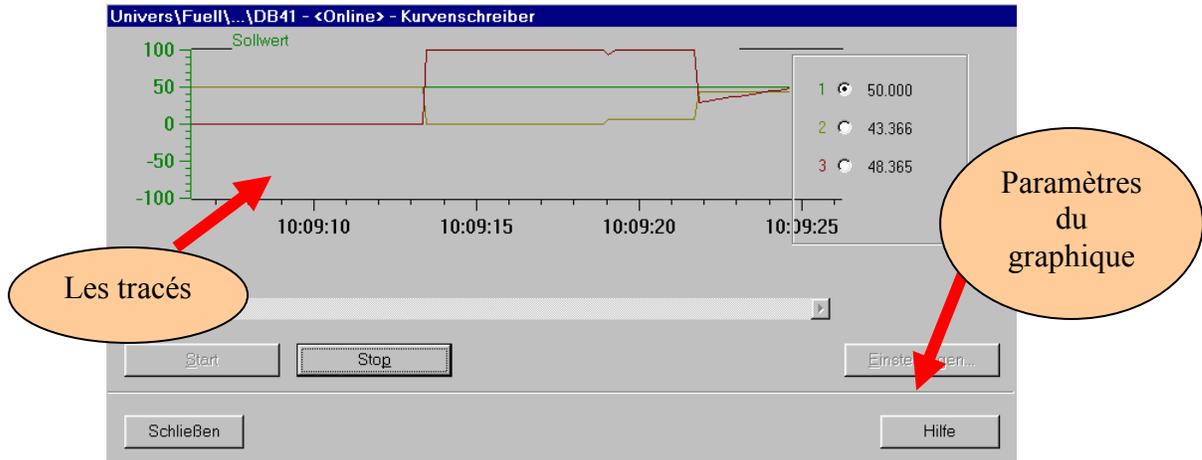


Figure I.26 : Tracée des trois courbe consigne-mesure-commande

Cette description technique de la fonction de régulation PID, exclusivement le bloc de la régulation continue FB41, sera mise en œuvre dans l'application « *Réalisation et Gestion d'un prototype d'une station de pompage à base d'automate Siemens* ».

I.9 Conclusion

On constate la facilité et la souplesse qu'offre L'A.P.I pour sa programmation, connexion, et adaptation aux conditions industrielles, avec toutes les fonctionnalités indispensables à l'automatisation des processus. La diversité des possibilités, mise en œuvre et son coût, le rendent incontournable lors de l'élaboration d'une solution.

Mais il ne faut pas oublier qu'il est nécessaire d'avoir une bonne analyse du problème à résoudre tout en assurant le respect des règles d'installation.

CHAPITRE II

Le réseau local industriel PROFIBUS

II.1 Introduction

Dans le cadre d'une évolution conduisant une automatisation globale, l'automate est de moins en moins acheté « nu ». Et même si c'est le cas, il doit pouvoir être connecter à d'autres matériels à processeur, et pouvoir dialoguer avec les agents d'exploitation.

Il faut donc se pencher sur ses liens avec son environnement et les fonctions qu'il doit assurer, outre son rôle premier de commande d'un dispositif de production.

II.2 Définitions élémentaires

Terrain : indique une espace délimité géographiquement (usine, atelier, voiture...)

BUS : au sens informatique industrielle, conducteur ou ensemble de conducteurs communs à plusieurs circuits permettant l'échange de données entre eux avec :

- Liaisons communes ;
- Plusieurs circuits ;
- Référence à la topologie de la configuration.

RESEAU : ensemble de lignes de communication qui desservent une même unité géographique

BUS / RESEAU DE TERRAIN : Terme générique d'un nouveau réseau de communication numérique, bidirectionnel, multi branche («multi drop»), série reliant différents types d'équipements d'automatisme :

- E/S déportées ;
- Capteur / Actionneur ;
- Automate programmable ;
- Calculateur.

II.3 Définition et normalisations PROFIBUS :

PROFIBUS (*Process Field Bus*) est un réseau de terrain ouvert, permettant de répondre à un large éventail d'applications dans les domaines concernant :

- les procédés manufacturiers (conduite des procédés séquentiels, procédés discontinus par lots « batch »)

- les procédés continus (conduite, régulation)
- la gestion des bâtiments (gestion technique centralisée, gestion technique du bâtiment).

PROFIBUS est issu de travaux initiés en 1987 par le ministère fédéral allemand pour la Recherche technologique, comprenant un groupement de sociétés industrielles et d'instituts de recherche allemands, orchestré par *Siemens AG* [7].

La norme allemande **DIN 19245** a été créée en 1991, normalisée **EN 50170** par le *Cenelec* en 1996, comme norme européenne.

Depuis 1999, le réseau PROFIBUS est reconnu dans la norme internationale **CEI 61158**, avec les autres réseaux *ControlNet*, *P-Net*, *Fieldbus Foundation High-Speed-Ethernet*, *SwiftNet*, *WorldFip*, *Interbus-S*.

PROFIBUS, étant un réseau adhérent à la norme internationale *ISO*, adopte le modèle *OSI* (*Open Systems Interconnections*), qui définit un langage commun aux échanges de données entre stations d'un réseau fondé sur des règles d'interconnexion et des interfaces de transfert désignant un « protocole de communication ». Ce protocole, construit en sept couches, définit les éléments, structures et tâches nécessaires à toute communication. Chaque couche remplit une fonction bien précise dans l'architecture OSI [8]. Toutefois, à défaut d'être utiles, certaines couches peuvent en être exclues, c'est ainsi que **PROFIBUS** se cantonne aux couches 1,2 et 7(Figure 2).

| Émetteur | Récepteur | Désignation et rôle des différentes couches de l'édifice OSI | |
|-------------------------|-----------|--|--|
| 7 | 7 | Application | Véritable interface entre le réseau et le programme d'application, dotée de commandes applicatives (lecture, écriture) |
| 6 | 6 | Présentation | Représentation (codage) des données en vue de permettre leur analyse et interprétation par la couche suivante |
| 5 | 5 | Session | Établissement et libération de liaisons temporaires entre stations ; synchronisation des communications |
| 4 | 4 | Transport | Gestion de la transmission pour la couche 5 (erreurs d'acheminement, découpage en paquets) |
| 3 | 3 | Réseau | Établissement et libération de liaisons, mise en œuvre de mécanismes pour éviter la congestion du réseau |
| 2 | 2 | Liaison de données | Gestion des règles d'accès au bus (<i>Medium Access Control</i> , <i>MAC</i>) et de sécurisation des échanges |
| 1 | 1 | Physique | Caractéristiques mécaniques, électriques et fonctionnelles de la liaison (connectique, codage et débit des signaux) |
| Support de transmission | | | |

Figure II.1 : Les sept couches du modèle OSI [7]

II.4 Variantes du réseau

PROFIBUS se décline en trois variantes de protocoles, répondant chacune à des finalités métiers et applicatives spécifiques :

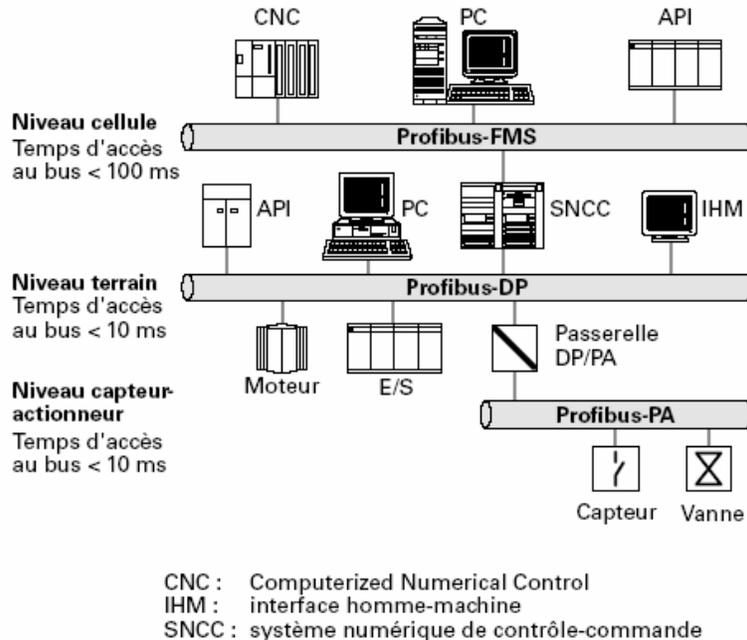


Figure II.2 : Classification des réseaux PROFIBUS [7]

II.4.1 PROFIBUS-DP (*Decentralized Peripheral*) : destiné aux applications de type maître-esclave en mono-maître pour la gestion des équipements d'entrées-sorties déportées avec des temps d'accès extrêmement courts. Le fonctionnement multi-maître est possible ;

II.4.2 PROFIBUS-FMS (*Fieldbus Message Specification*) : destiné aux applications nécessitant l'échange entre maîtres pour la synchronisation d'activités de contrôle-commande, basé sur la messagerie MMS (*Manufacturing Message Specification*) ;

II.4.3 PROFIBUS-PA (*Process Automation*) : destiné aux applications de contrôle de processus nécessitant la communication avec des équipements de terrain (capteurs, actionneurs) permettant une télé alimentation des équipements et un fonctionnement avec sécurité intrinsèque en ambiance explosive.

II.5 Principe d'accès au BUS

PROFIBUS met en œuvre un modèle de communication de type Maître-esclave selon un mode d'accès au bus de nature hybride, comme le montre la (figure 3).

Les équipements **maîtres**, appelés **stations actives**, dirigent la transmission de données sur le bus et émettent librement des messages, sous réserve d'obtenir le droit d'accès au médium, déterminé par le passage d'un **jeton**.

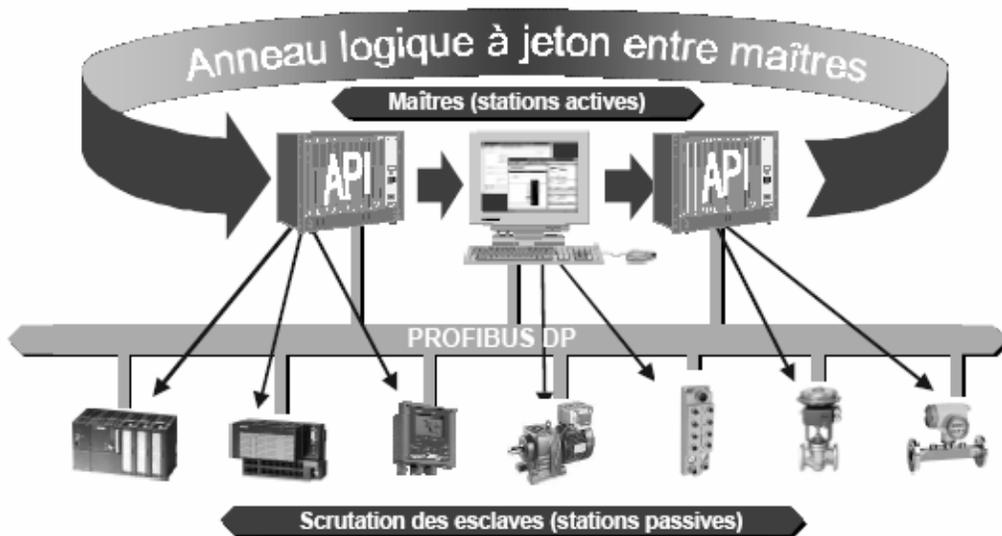


Figure II.3 : Principe d'accès au BUS [10]

Les équipements **esclaves**, appelés **stations passives**, sont des équipements périphériques (blocs d'entrées-sorties, vannes, entraînements et transmetteurs de mesure, etc.) qui n'ont pas le droit d'accès au bus. Leur action se limite à l'acquittement des messages reçus des maîtres ou à la transmission de messages en réponse à une demande des maîtres.

La nature hybride du principe d'accès au réseau implémenté par PROFIBUS permet :

- d'une part une communication entre les stations maîtres par un mécanisme de passage de jeton sur bus, déterministe et adaptatif. La circulation du jeton est effectuée selon un anneau logique, indépendamment de la topologie bus du réseau ;
- d'autre part une communication simple de type maître-esclave entre une station maître et les équipements esclaves auxquels elle veut s'adresser.

Chaque station maître (station active) disposant du droit d'accès au bus figuré par le passage du jeton, qui est constitué d'une trame spéciale, est libre d'accéder à tout esclave (station passive) connecté au réseau.

II.6 Structure des télégrammes

Les informations sont transmises sur les réseaux **PROFIBUS** par des séquences d'octets appelées **télégrammes**, constituées d'une série d'octets contrôlés par un bit de parité paire et transmis en mode asynchrone, encadrés d'un bit **START** (niveau logique 0) et **stop** (niveau logique 1) [10].

II.7 Les techniques de transmissions :

Rappelons que la couche 1 du modèle OSI assure la transmission physique des données. Elle en définit donc les caractéristiques électriques et mécaniques : type de codage et d'interface normalisée.

PROFIBUS spécifie plusieurs versions de couches « physique », selon la technique de transmission, qui sont toutes conformes aux normes internationales **CEI 61158** et la **CEI 61784** :

II.7.1 Transmission RS 485

RS 485 est une technique de transmission simple et économique convenant surtout aux tâches exigeant des débits élevés. Son support de transmission est la paire torsadée blindée.

La structure du bus permet l'ajout ou le retrait de stations ou la mise en service par étapes du réseau sans répercussion sur les autres stations.

Les extensions futures (dans des limites définies) ne pénalisent pas les stations en exploitation.

A cela s'ajoute la possibilité d'exploiter cette liaison en zone à sécurité intrinsèque [10].

II.7.2 Transmission optique

Certaines applications de bus s'accommodent mal de la transmission filaire. C'est le cas des milieux industriels à forte pollution électromagnétique ou des réseaux à longue portée, des contraintes auxquelles remédie parfaitement la fibre optique [10].

II.8 Mise en œuvre d'une application à base du PROFIBUS DP avec Maître CPU 315-2DP et Esclave MICROMASTER 440 : « Entraînement d'un chariot sur un plan incliné »

Pour illustrer ce qu'on a vu précédemment, on va mettre en œuvre une application d'entraînement d'un moteur asynchrone, qui déplace un chariot sur un plan incliné.

Le moteur asynchrone à courant alternatif va être commandé par le variateur de vitesse MICROMASTER 440 et qui sera configuré comme esclave sur le réseau.

Une station SIMATIC S7-300 (CPU315-2DP) avec interface PROFIBUS-DP intégrée va être utilisée pour la gestion du « Monte-chariot » par le biais de l'esclave, et sera configurée comme maître.

II.8.1 Le cahier de charge

Voici le schéma du « Monte-chariot » et son panneau de contrôle :

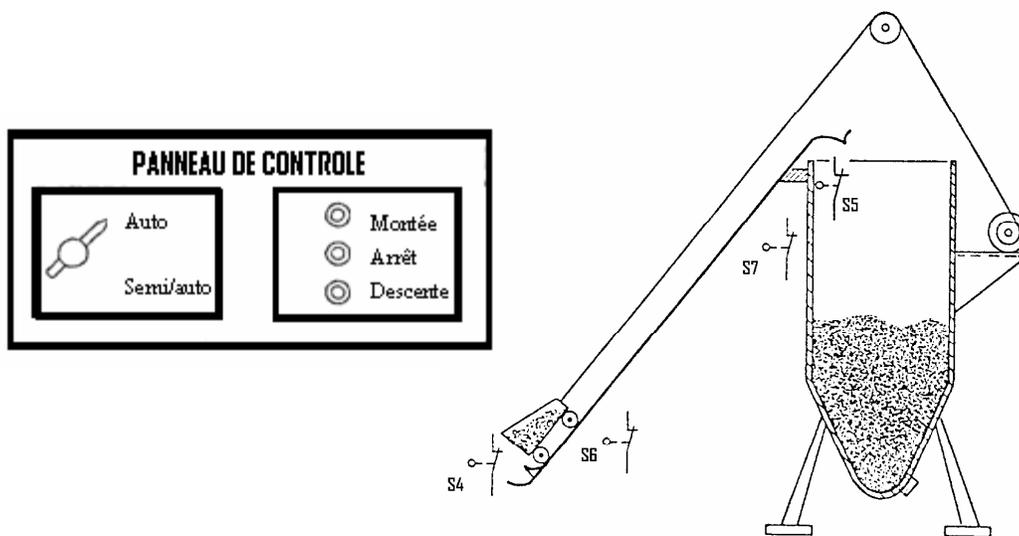


Figure II.4 : Schéma de principe de l'application

Si le commutateur *automatique / semi-automatique* est sur "**Automatique**", alors le container va être monté ou descendu en appuyant de manière brève sur les touches *Montée*, respectivement *Descente*. Le chariot se déplace jusqu'à rencontrer le capteur de fin de course (*S4, resp. S5*). En appuyant sur la touche *Arrêt*, le déplacement du chariot est stoppé.

Si le commutateur *automatique / semi-automatique* est sur "Semi-automatique", alors le chariot monte (descend) tant que l'utilisateur appuie sur le bouton "*Montée*" ("*Descente*")

Dans la phase de montée, le chariot accélère jusqu'à la vitesse requise. Quand le chariot atteint le capteur de limite **S7**, il entre dans la phase de freinage. Le chariot sera donc freiné jusqu'à la vitesse faible exigée et continuera à cette même vitesse jusqu'à atteindre le capteur de fin de course **S5**.

Dans la phase de descente, le chariot accélère jusqu'à sa vitesse maximale. Quand le chariot atteint le capteur de limite **S6**, alors le chariot est freiné et continue à cette basse vitesse jusqu'au capteur de fin de course **S4**.

Voici les normes à respecter pour cela :

- La durée pour atteindre le régime haut de fréquence (50Hz) doit valoir 10 secondes ;
- La durée pour revenir de la fréquence (50Hz) au repos doit valoir 15 secondes ;
- Le moteur doit s'arrêter en 3 secondes (pour une fréquence de 5Hz) ;
- Le chariot sera remonté conformément à la fréquence $F1 = 30\text{Hz}$;
- Le chariot sera redescendu conformément à la fréquence $F2 = 50\text{Hz}$;
- Lors de la phase de décélération, la nouvelle consigne devient $F3 = 5\text{Hz}$.

En détaillant chaque phase comme suit :

Monter le chariot :

- Actionnez « *Montée* »; Le moteur démarre et atteint la fréquence requise $F1=30\text{Hz}$ en 6 secondes ;
- Détecteur de position **S7** actif ; Le moteur freine et atteint la fréquence requise de $F3=5\text{Hz}$ en 7,5 secondes ;
- Détecteur de position **S5** actif ; Le moteur s'arrête en 3 secondes environ (durée de lissage).

Descendre le chariot :

- Actionnez « *Descendre* »; Le moteur démarre et atteint la fréquence max. $F2 = 50\text{ Hz}$ en 10 secondes;
- Détecteur de position **S6** actif; Le moteur freine en 13,5 secondes jusqu'à la fréquence de 5Hz;
- Détecteur de position **S4** actif; Le moteur s'arrête au bout d'environ 3 secondes (Temps de lissage).

Le chronogramme suivant illustre le fonctionnement exigé dans le cahier de charge :

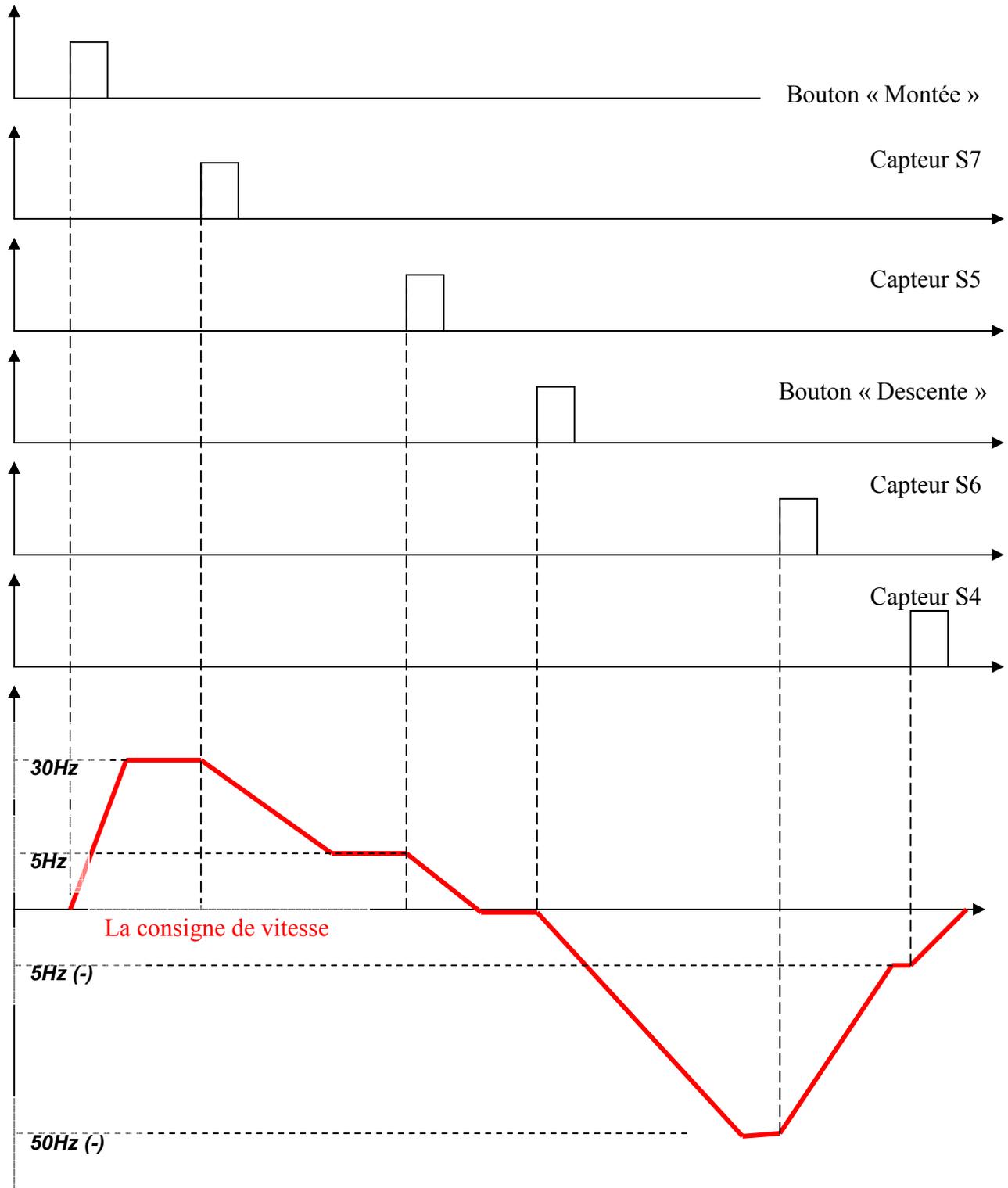


Figure II.5 : Chronogramme de fonctionnement

II.8.2 La mise en œuvre de l'application

Tout au long de cette application, la méthodologie est représentée dans l'organigramme suivant :

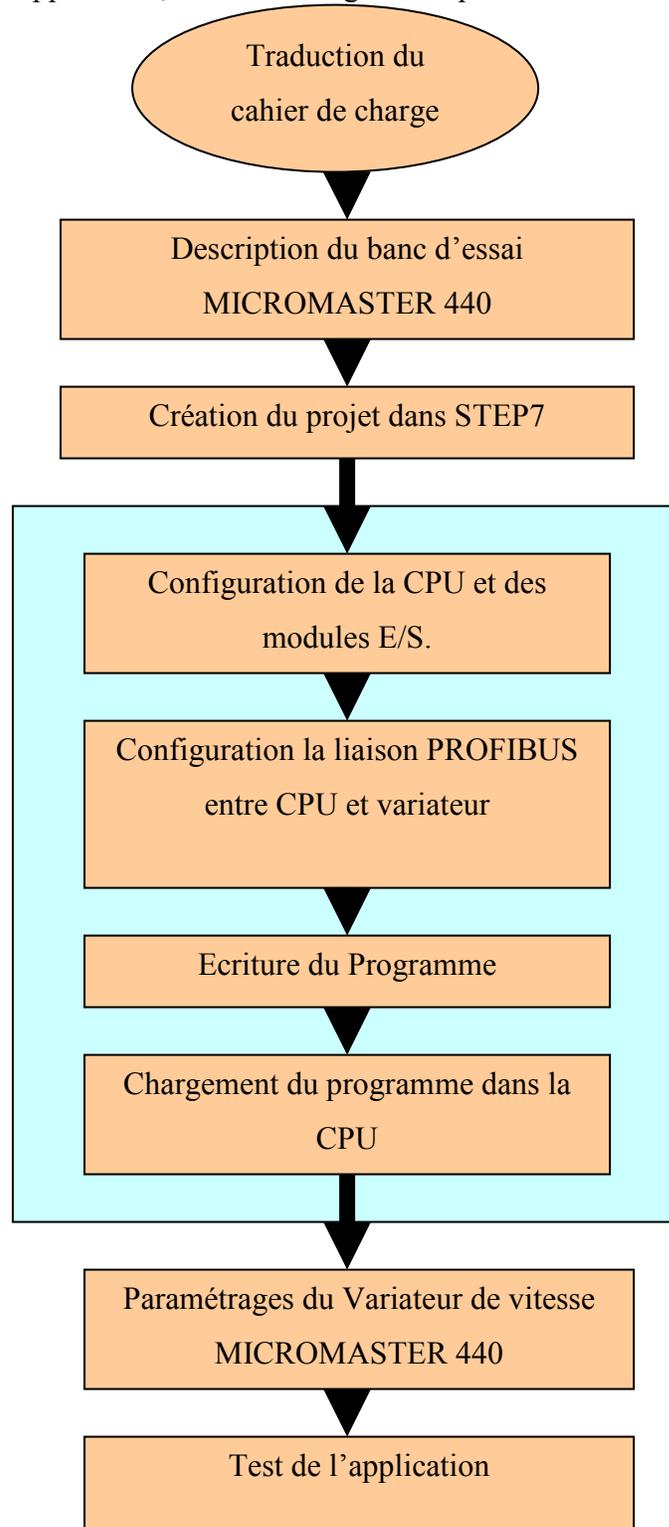


Figure II.6 : l'organigramme de l'application du chariot

II.8.2.1 Description du banc d'essai et du profil de communication pour l'application

II.8.2.1.a. Les éléments du réseau PROFIBUS

Pour cette application, les éléments du réseau sont les suivants :

- 1 PC, système d'exploitation Windows XP ;
- 2 Logiciel STEP 7 V 5.3;
- 3 Interface ordinateur MPI (Ex : PC- Adapter) pour configurer l'automate ;
- 4 API SIMATIC S7-300 avec la CPU 315-2DP, pour cette application on utilisera celle déjà mise en œuvre précédemment, dans l'application du tapis de remplissage (application1) ;
- 5 Vecteur MICROMASTER 440 avec module PROFIBUS CB15 ;
- 6 Câble PROFIBUS avec 2 prises PROFIBUS ;
- 7 Moteur asynchrone Siemens, avec une charge constante pour la simulation.

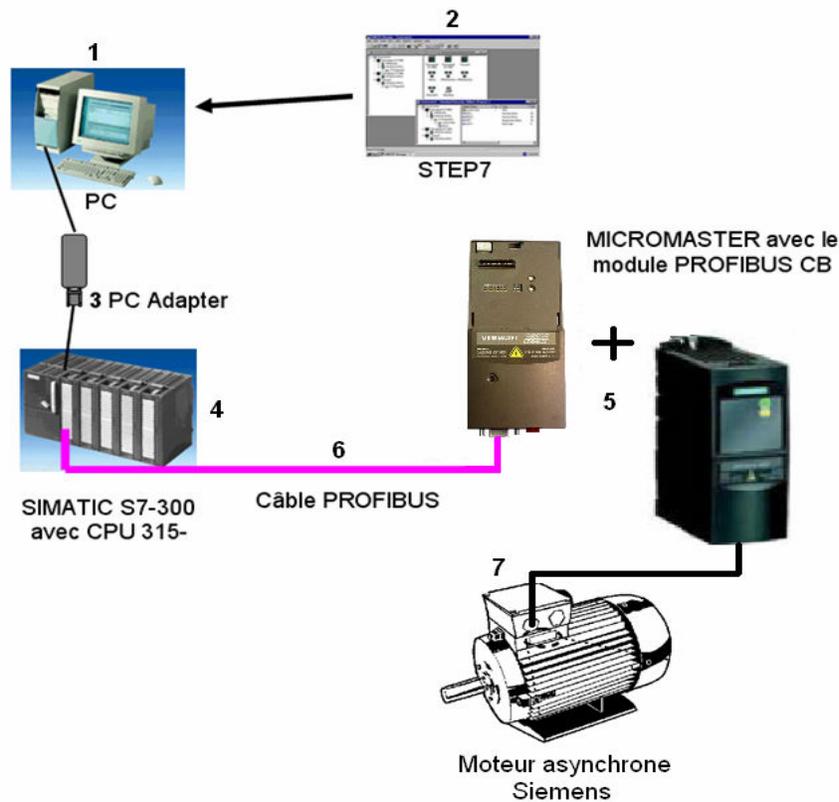


Figure II.7 : Schéma de principe de l'application

Les trois éléments, qui sont le moteur, le variateur et la carte de communication CB pour PROFIBUS-DP sont rassemblés dans un banc d'essai SIDEMO_MM440 présenté dans la figure suivante :



Figure II.8 : Banc d'essai SIDEMO_MM440 [24]

Cette variante de banc d'essai ne comporte pas de capteur de vitesse, ni de frein mécanique, ce qui est très important à considérer pour la suite de l'application.

Remarque : Chaque composant, cité précédemment, est détaillé dans les annexes liées au rapport.

II.8.2.1.b. La communication et les données cycliques CPU315- MICROMASTER 440 par le PROFIBUS-DP

La commande du *MICROMASTER 440* s'effectue sur le canal cyclique de PROFIBUS-DP. De plus il est possible d'échanger des paramètres par cette voie. La structure des données utiles pour le canal cyclique est définie dans le profil *PROFIDrive* version 2.0 et appelée *PPO* (Objet Paramètres Données processus).

Le profil *PROFIDrive* définit la structure des données utiles pour les entraînements, qui permet au maître d'accéder aux entraînements esclaves par un échange de données cyclique.

- Structure des données utiles conforme aux PPO

La structure des données utiles dans la transmission cyclique se subdivise en deux parties qui peuvent être transmises dans chaque télégramme:

La zone des données processus (*PZD*), c.-à-d. des mots de commande et des valeurs de consignes et/ou des informations d'état et des valeurs de mesure.

La zone des paramètres (*PKW*) pour lire / écrire des valeurs de paramètres.

Le choix de chaque type **PPO** est fonction de la tâche affectée au variateur dans le système d'automatisation. Les données processus sont transmises systématiquement. Elles sont traitées dans le variateur avec la priorité maximale et dans les tranches de temps les plus courtes. Le variateur est piloté par les données processus, par ex. mise en marche / arrêt, transmission des consignes...etc. [11].

A l'aide de la zone des paramètres, l'utilisateur accède librement par le bus à tous les paramètres se trouvant dans le variateur. Il peut par exemple lire des informations détaillées concernant des diagnostics, des messages de défaut, etc.

Ainsi, les télégrammes de transmission cyclique des données présentent la structure de base suivante:

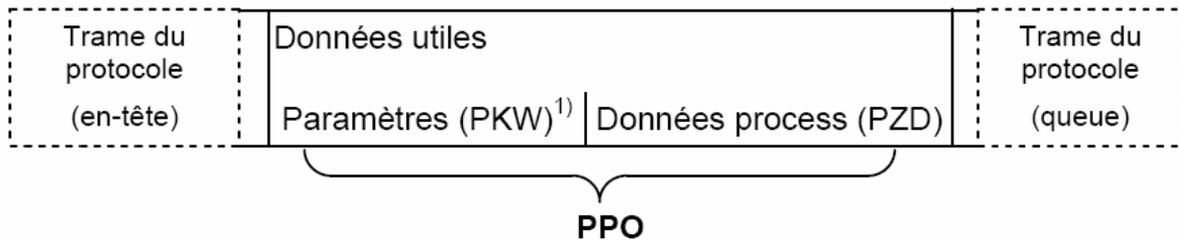


Figure II.9 : Structure générale du télégramme de transmission cyclique des données

Avec :

- **Zone des données de paramétrage (PKW)**

La partie **PKW** du télégramme (identification / valeur de paramètre) permet de surveiller et/ou de modifier n'importe quel paramètre dans le variateur.

- **Zone des données processus (PZD)**

Les données processus permettent de transmettre des mots de commande et des consignes (requêtes: maître → variateur) et/ou des mots d'état et des valeurs de mesure (réponses: variateur → maître).

Il existe cinq types de **PPO**, représentées dans la figure qui suit, et dont la différence réside dans la longueur des trames échangées entre maître-esclave :

| | PKW | | | | PZD | | | | | | | | | |
|------|---------|----------|----------|----------|----------------------|--------------------|----------|----------|----------|----------|----------|----------|----------|-----------|
| | PKE | IND | PWE | | PZD1 STW1 ZSW1 | PZD2 HSW HIW | PZD3 | PZD4 | PZD5 | PZD6 | PZD7 | PZD8 | PZD9 | PZD10 |
| | 1er mot | 2ème mot | 3ème mot | 4ème mot | 1er mot | 2ème mot | 3ème mot | 4ème mot | 5ème mot | 6ème mot | 7ème mot | 8ème mot | 9ème mot | 10ème mot |
| PPO1 | | | | | | | | | | | | | | |
| PPO2 | | | | | | | | | | | | | | |
| PPO3 | | | | | | | | | | | | | | |
| PPO4 | | | | | | | | | | | | | | |
| PPO5 | | | | | | | | | | | | | | |

PKW: Identification valeur de paramètre STW: Mot de commande 1
 PZD: Données process ZSW: Mot d'état 1
 PKE: Identification de paramètre HSW: Consigne principale
 IND: Indice HIW: Valeur de mesure principale
 PWE: Valeur de paramètre

Figure II.10 : Données processus de type PPO [11]

Le temps de réponse du *MICROMASTER 440* concernant les *PZD* est de 20 millisecondes environ.

Il s'agit du temps entre l'arrivée de la consigne auprès de l'esclave DP, et la mise à disposition de la mesure actualisée (et de référence) sur le **PROFIBUS-DP**.

Le temps de réponse du *MICROMASTER 440* pour une modification de paramètre interne (*PKW*) s'élève à 50 millisecondes environ.

Pour notre application, on va utiliser pour l'échange de données commande/état le type *PPO3*, qui sera détaillé dans ce qui suit :

- Description du type PPO3

C'est le type de PPO le plus répandu pour sa simplicité, tout en intégrant tout les fonctionnalités nécessaires dans les entrainements, avec les données de processus, les mots de commande et les valeurs théoriques Maître-variateur ainsi que les mots d'état et les valeurs pratiques variateur-Maître.

Il ne comporte pas de *PKW*, donc le paramétrage du variateur ne pourra se faire que via une console muni de l’outil de paramétrage du variateur *DRIVEMONITOR*, qui sera abordé plus loin dans ce chapitre.

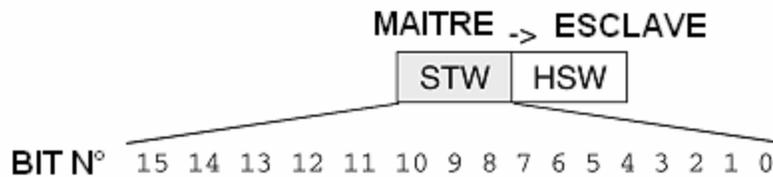
Le tableau suivant donne la structure de la partie de trame existante, qui est le PZD :

| | PZD1 | PZD2 |
|---|---------------------------------|---|
| Trame de requêtes (Maître _ Esclave) | Mot de contrôle (STW) | Valeur principale théorique (HSW) |
| Trame de réponses (Esclave _ Maître) | Mot d’état (ZSW) | Valeur principale pratique (HIW) |

Tableau II.1 : Structure de la PZD dans le PPO3

La structure en bit, avec l’affectation pour chaque bit est donnée comme suit :

o **Le mot de contrôle (STW)**



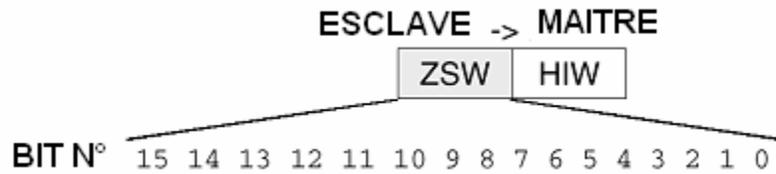
| <i>Bit</i> | <i>Valeur</i> | <i>Signification</i> | <i>Remarques</i> |
|------------|---------------|----------------------|--|
| 0 | 1 | MARCHE | Placer le mutateur à l’état « mise en marche prête », le sens de rotation doit être défini avec le bit 11. |
| | 0 | ARRET 1 | Repos, retour à la rampe HLG, impulsion coupée pour $f < f_{min}$. |
| 1 | 1 | Condition de marche | La commande ARRET 2 est désenclenchée. |
| | 0 | ARRET 2 | Impulsion bloquée immédiate, un arrêt lent. |
| 2 | 1 | Condition de marche | La commande ARRET 3 est désenclenchée. |
| | 0 | ARRET 3 | Pour des temps retour / repos programmés $< 10s$ ($P003 < 10$) avec la moitié du temps retour / repos, pour $P003 > 10$, en 5s. |
| 3 | 1 | Mode libre | Régulation et impulsion de direction rotation |

| | | | |
|-----------|--------|--|--|
| | 0 | Mode bloqué | sont libres. Régulation et impulsion de direction rotation sont bloquées. |
| 4 | 1 0 | Condition de marche Bloquer haut régime | Le haut régime est libre. La sortie qui définit le haut régime est mise à zéro (freinage le plus rapide possible), le mutateur reste à l'état MARCHE. |
| 5 | 1 0 | Haut régime libre Arrêter haut régime | Les valeurs théoriques actuelles du HLG sont figées. |
| 6 | 1 0 | Valeur théorique libre Valeur théorique bloquée | La valeur choisie à l'entrée du HLG va être appliquée. La valeur choisie à l'entrée du HLG va être mise à zéro. |
| 7 | 1 0 | Acquitter Pas de signification | Le message d'erreur va être acquitté par un front montant, le mutateur se place ensuite en 'marche bloquée'. |
| 8 | 1 0 | Semi-Auto droite Pas de Semi-Auto | |
| 9 | 1 0 | Semi-Auto gauche Pas de Semi-Auto | |
| 10 | 1 0 | PZD validé PZD non validé | Du maître, les données du processus valide seront transmises. Les données du processus transmises au maître sont invalides |
| 11 | 1 0 | Déplacement à droite Déplacement à gauche | Marche vers la droite Marche vers la gauche |
| 12 | | | Libre |
| 13 | 1 0 | Potentiomètre motorisé augmente | Libre |
| 14 | 1 0 | Potentiomètre motorisé diminue | |
| 15 | 1 | Commande sur site | Commande sur site active |

| | | | |
|--|---|----------------------------------|----------------------------|
| | 0 | (BOP/AOP) Commande à distance | Commande à distance active |
|--|---|----------------------------------|----------------------------|

Tableau II.2 : Le mot de contrôle STW [11]

o Le mot d'état (ZSW)



| Bit | Valeur | Signification | Remarques |
|----------|--------|---------------------------------|--|
| 0 | 1 | Marche prête | Alimentation en courant branchée, électronique initialisée, impulsion verrouillée. |
| | 0 | Marche non prête | |
| 1 | 1 | Mode de fonctionnement prêt | Le mutateur est en marche (commande marche), il n'y a pas de dérangement. Le mutateur peut être lancé avec la commande 'Mode libre'. Causes : commande de marche non effectuée, erreur, ARRET 2 ou ARRET 3, marche bloquée. |
| | 0 | Mode de fonctionnement non prêt | |
| 2 | 1 | Mode libre | Voir le bit 3 du mot de contrôle. |
| | 0 | Mode bloqué | |
| 3 | 1 | Erreur | Fonctionnement défectueux et donc hors-service, après acquittement et traitement correct des erreurs, passe en marche bloquée. |
| | 0 | Pas d'erreur | |
| 4 | 1 | Pas d'ARRET 2 | |
| | 0 | ARRET 2 enclenché | |
| 5 | 1 | Pas d'ARRET 3 | |
| | 0 | ARRET 3 enclenché | |
| 6 | 1 | Marche bloquée | Redémarrage seulement pour ARRET 1 et ensuite MARCHE. |
| | 0 | Pas de marche bloquée | |

| | | | |
|----|--------|--|--|
| 7 | 1 0 | Avertissement Pas d'avertissement | Toujours en marche ; acquittement non nécessaire. |
| 8 | 1 0 | Pas d'écart consigne-mesure Ecart consigne-mesure | Ecart entre valeurs de consigne et de mesure dans les tolérances 0 |
| 9 | 1 0 | Exécution requise Fonctionnement local | Le système d'automatisation doit entreprendre l'exécution. Exécution uniquement possible sur place (local). |
| 10 | 1 0 | f atteint f dépassé (vers le bas) | Fréquence de sortie du mutateur correspondant à la valeur théorique configurée. Fréquence de sortie du mutateur plus petite que la valeur théorique configurée. |
| 11 | 1 | Alarme : moteur à la limite du courant | |
| 12 | | | Non utilisé |
| 13 | | Surcharge du moteur | Les données du moteur permettent de conclure à une surcharge |
| 14 | 1 0 | Déplacement à droite Déplacement à gauche | Tension de sortie du mutateur a un champ tournant droit. Tension de sortie du mutateur a un champ tournant gauche. |
| 15 | | Surcharge du variateur | Par ex. courant ou température |

Tableau II.3 : Le mot d'état ZSW [11]

Nous allons exploiter ces bits pour la commande du moteur via le variateur, pour ce qui est des bits de d'ordre de commande **STW**, et pour le diagnostic, pour ce qui est des bits du mot d'état **ZSW**.

La consigne de vitesse pour le moteur sera transmise via le mot **HSW**, et la mesure de vitesse estimée par le variateur sera chargée directement du mot **HIW**.

II.8.2.2 Création du projet dans STEP7

Dans cette partie, pour la création du projet dans STEP 7, on va suivre les mêmes principes de programmation que lors de la précédente application. On va suivre la méthodologie présentée auparavant dans l'organigramme du projet (Figure : l'organigramme de l'application).

a. Configuration du matériel

Pour la configuration matérielle, on va suivre les mêmes instructions, et cela en utilisant le même automate. On peut déjà considérer que la configuration de la CPU et des modules est déjà faite.

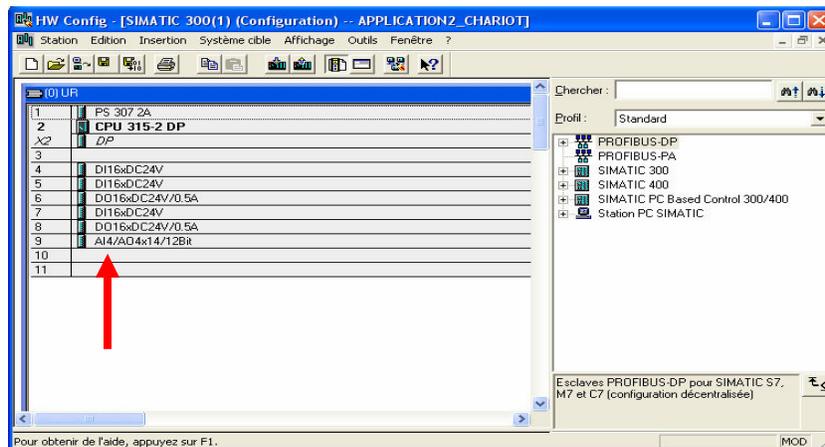


Figure II.11 : Configuration matériel pour la CPU315 et ces modules E/S

Maintenant on va configurer la liaison PROFIBUS-DP, et cela en créant un réseau PROFIBUS maître, comme cela est indiqué :

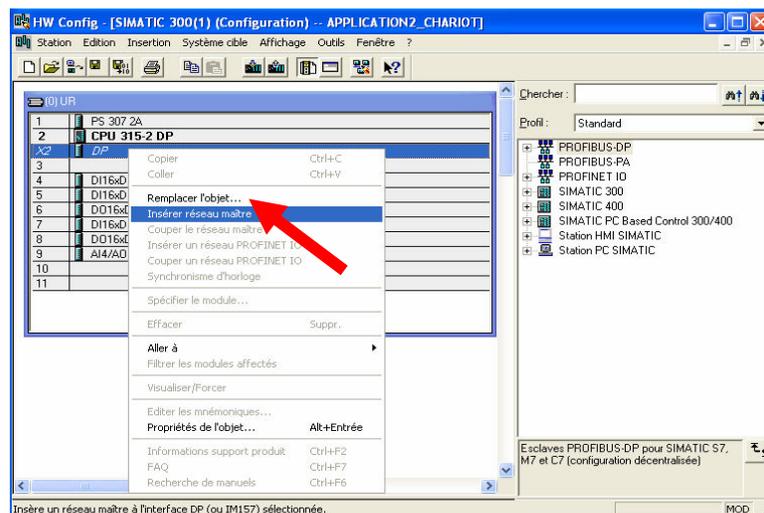


Figure II.12 : Création d'un Réseau PROFIBUS Maître

Après le choix de la tâche dans le menu précédent, la fenêtre « *Création réseau PROFIBUS* » apparaît.

La configuration des paramètres de transmission est accessible dans le menu « *Propriétés* », de la figure précédente :

- La vitesse de transmission choisi : **1,5 Méga Bits/seconde** ;
- Le profil choisi, est bien sure, le **DP**.

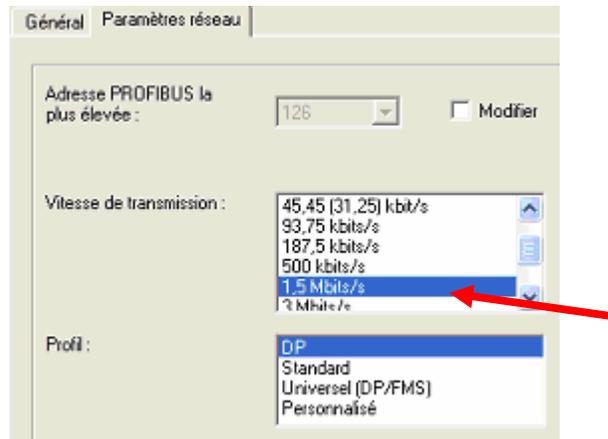


Figure II.13 : Paramètres de la transmission PROFIBUS-DP

Ainsi on aura créé et paramétré le nouveau réseau **PROFIBUS-DP** entre CPU-Variateur.

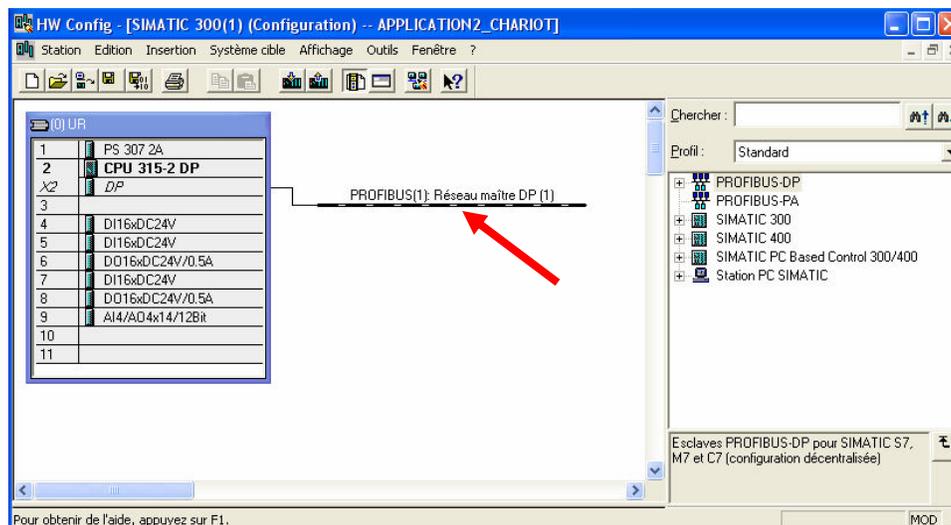


Figure II.14 : Insertion du réseau maître CPU – Esclave MM440

La nouvelle étape consiste à insérer et paramétrer l'esclave, Et pour cela, on va chercher la variante du MicroMaster à insérer dans le catalogue, par le chemin suivant :

PROFIBUS-DP→SIMOVERT→MICROMASTER

Et choisir par la suite le standard de communication approprié, qui est dans cette application le **PPO3** comme indiqué dans la figure suivante.

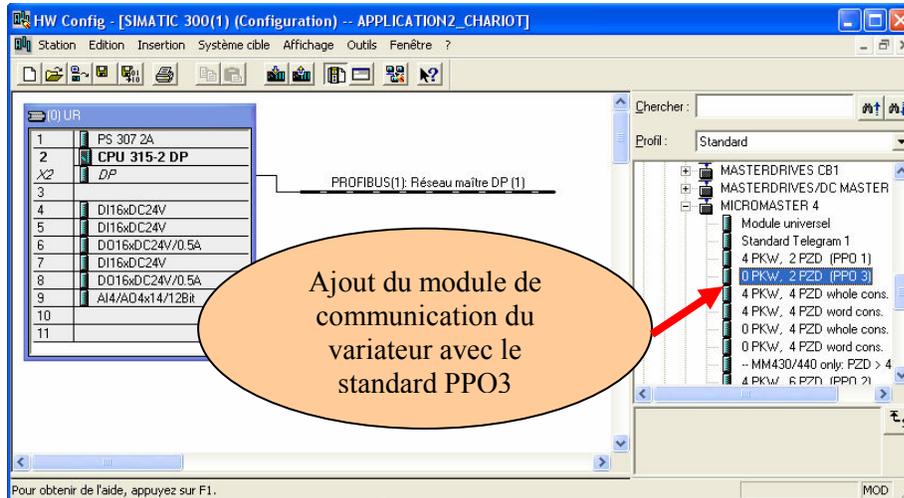


Figure II.15 : Insertion du réseau maître CPU – Esclave MM440

Après la conception de la liaison, il reste maintenant à attribuer les adresses dites « Virtuelles » pour le double mot de sortie-commande **STW-HSW**, et le double mot d'entrée-état **ZSW-HIW** :

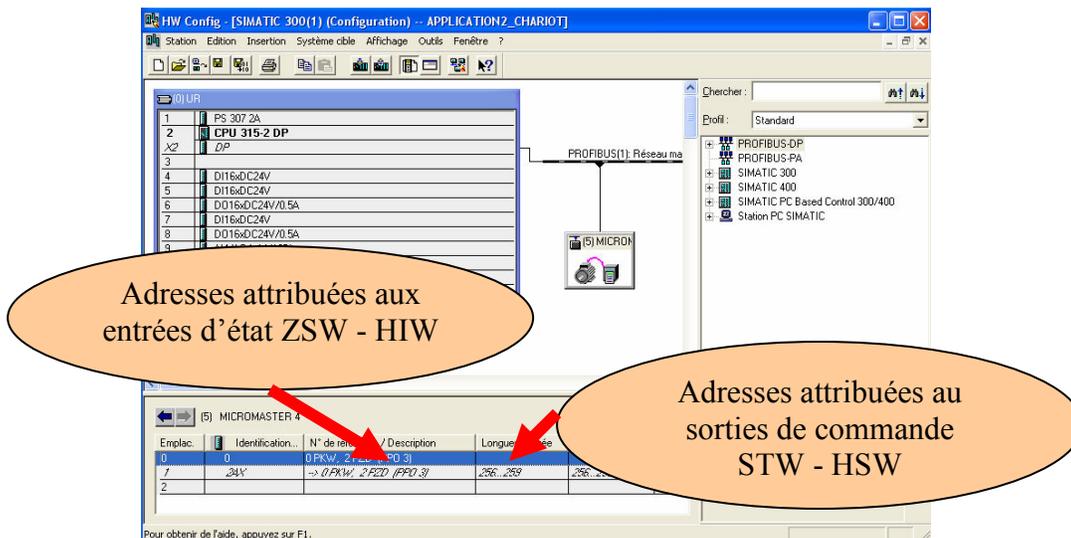


Figure II.16 : Adressage des entrées-sorties pour la communication PROFIBUS-DP

On attribue les adresses comme suit :

- Pour les sorties de commande **STW-HSW** :

| Mot de commande (STW) | | | | | | | | | | | | | | | | Valeur principale théorique (Consigne HSW) | | | | | | | | | | | | | | | |
|-----------------------|----|----|----|----|----|---|---|---------|---|---|---|---|---|---|---|--|----|----|----|----|----|---|---|---------|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PAB 256 | | | | | | | | PAB 257 | | | | | | | | PAB 258 | | | | | | | | PAB 259 | | | | | | | |
| PAW 256 | | | | | | | | | | | | | | | | PAW 258 | | | | | | | | | | | | | | | |

Tableau II.4 : Adressage virtuelle du double mot de commande **STW-HSW**

- Pour les entrées **ZSW-HIW** :

| Mot d'état (ZSW) | | | | | | | | | | | | | | | | Valeur principale pratique (Valeur mesurée de vitesse HIW) | | | | | | | | | | | | | | | |
|------------------|----|----|----|----|----|---|---|---------|---|---|---|---|---|---|---|--|----|----|----|----|----|---|---|---------|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PEB 256 | | | | | | | | PEB 257 | | | | | | | | PEB 258 | | | | | | | | PEB 259 | | | | | | | |
| PEW 256 | | | | | | | | | | | | | | | | PEW 258 | | | | | | | | | | | | | | | |

Tableau II.5 : Adressage virtuelle du double mot d'état **ZSW-HIW**

Et on paramètre ces adresses dans le fenêtre de dialogue « Propriétés de l'esclave », accessible dans le menu de l'icône de l'élément « MicroMaster 440 » :

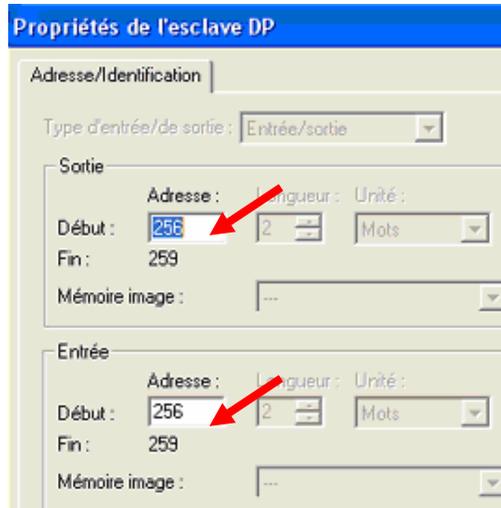


Figure II.17 : Adressages des entrées-sorties virtuelles STW-HSW et ZSW-HIW

b. Conception et écriture du Programme

Dans cette partie, nous allons créer le programme à transférer dans la CPU.

En premier lieu, on attribue des adresses absolues dans la CPU pour les entrées du processus, comme indiqué dans le tableau suivant :

| Entrées (Boutons/ capteurs) | Adresse absolue |
|---------------------------------------|-----------------|
| Auto/Manuel | E 0.0 |
| Montée | E 0.1 |
| Descente | E 0.2 |
| STOP | E 0.3 |
| S4 | E 0.4 |
| S5 | E 0.5 |
| S6 | E 0.6 |
| S7 | E 0.7 |
| Double Mot de commande STW-HSW | PAW 256 |
| Double mot d'état ZSW-HIW | PEW 256 |

Tableau II.6 : Adressage des entrées processus

La structure du programme se présente comme suit :

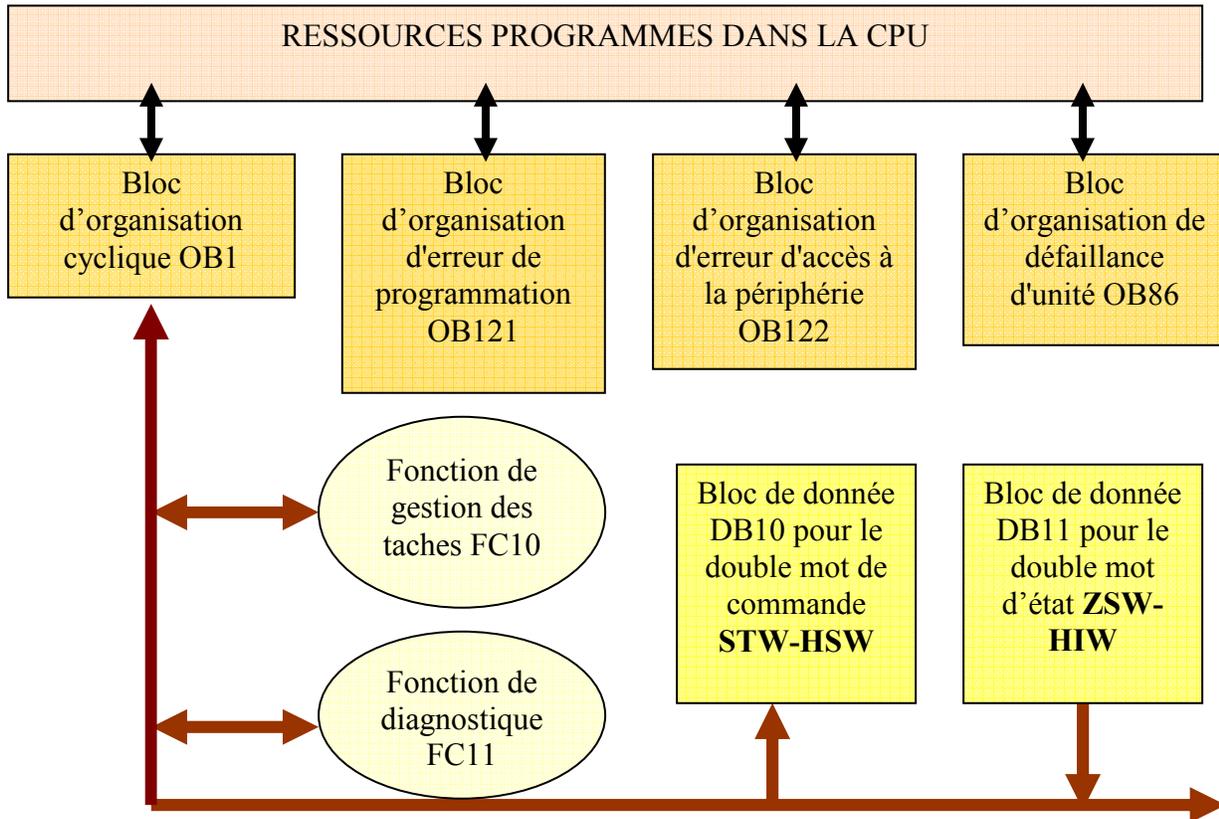


Figure II.18 : Arborescence du programme pour la deuxième application

Avec pour chaque bloc une tache spécifié :

- **Bloc d'organisation cyclique OB1** : c'est le bloc principale qui fait appel au :
 - **Fonction de gestion des taches FC10** : cette fonction a pour tache de gérer les données processus qui sont les capteurs et de donner par la suite les ordres de Marche/Arrêt ainsi que les consignes vitesses conformément au cahier de charge.
 - **Fonction de diagnostic FC11** : elle a pour tache le diagnostic des données électriques du variateur et du moteur, et imposer les arrêts d'urgences dans les situations critiques, qui sont les surtensions, surcharge moteur ou surcharge variateur.
 - **Bloc de donnée DB10 pour le double mot de commande STW-HSW** : ce bloc est réservé pour le double mot de commande avant son envoi vers le variateur via l'adresse virtuelle PAD 256 paramétrée dans l'étape précédente.

- **Bloc de donnée DB11 pour le double mot d'état ZSW-HIW** : ce bloc est réservé pour le double mot d'état après son acquisition de l'adresse de virtuelle PED 256 paramétrée dans l'étape précédente.

Et les Blocs de diagnostic des alarmes et des défauts Hardware/Software :

- **Bloc d'organisation d'erreur de programmation OB121** : Le système d'exploitation de la CPU appelle l'OB121 quand survient un événement provoqué par une erreur au cours du traitement du programme. C'est le cas, par exemple, lors de l'appel d'un bloc qui n'a pas été chargé dans la CPU [16].
- **Bloc d'organisation d'erreur d'accès à la périphérie OB122** : Le système d'exploitation de la CPU appelle l'OB122 quand une erreur survient lors de l'accès aux données d'un module. C'est le cas, par exemple, quand la CPU détecte une erreur de lecture alors qu'elle accède à des données d'un module de signaux [16].
- **Bloc d'organisation de défaillance d'unité OB86** : Le système d'exploitation de la CPU appelle l'OB86 quand il détecte la défaillance d'un réseau maître DP ou d'une station en périphérie décentralisée. L'absence de l'OB86 provoque l'arrêt de la CPU en l'absence d'un partenaire PROFIBUS-DP [16].

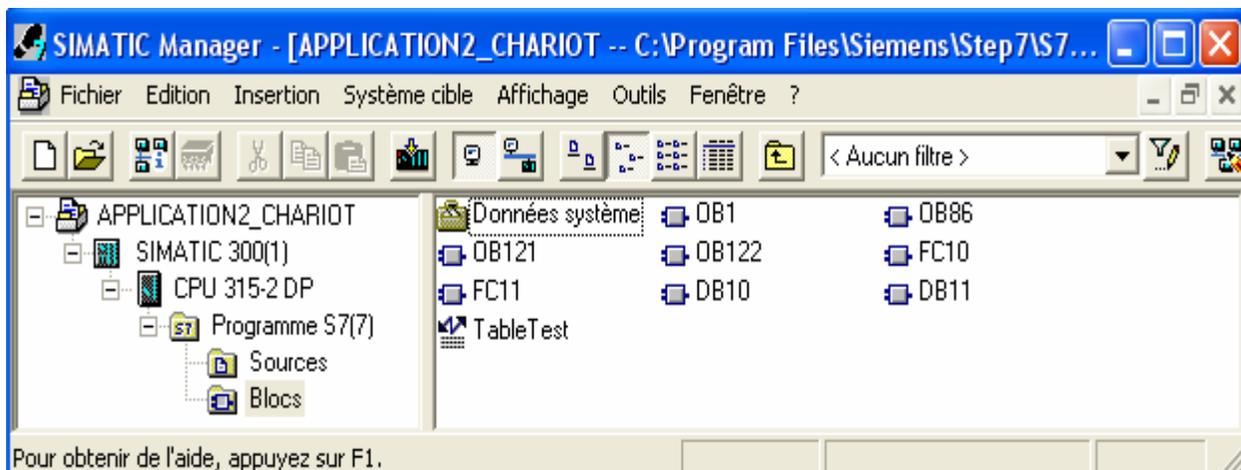


Figure II.19 : blocs programme dans SIMATIC Manager

Pour cette application, on utilisera l'adressage relatif, qui consiste, à créer une table de mnémorique :

| | Mnémorique | Opérande | Type de don | Commentaire |
|----|-----------------|----------|-------------|---|
| 1 | AUTO_SEMIAUTO | E 0.0 | BOOL | COMMUTATEUR AUTO_SEMI-AUTO |
| 2 | DESCENTE | E 0.2 | BOOL | BOUTON DE DESCENTE (FERMETURE DU CONTACT) |
| 3 | M_DESCENTE | M 70.1 | BOOL | MEMENTO DE DESCENTE |
| 4 | M_MONTEE | M 70.0 | BOOL | MEMENTO DE MONTEE |
| 5 | MOD_ERR | OB 122 | OB 122 | Module Access Error |
| 6 | MONTEE | E 0.1 | BOOL | BOUTON DE MONTEE (FERMETURE DU CONTACT) |
| 7 | MOT_COMMANDE | DB 10 | DB 10 | BLOC DE DONNEE A ENVOYER AU VARIATEUR |
| 8 | MOT_ETAT | DB 11 | DB 11 | BLOC DE DONNEE A LIRE DU VARIATEUR |
| 9 | RACK_FLT | OB 86 | OB 86 | Loss of Rack Fault |
| 10 | S4_BAS | E 0.4 | BOOL | FIN DE DESCENTE (OUVERTURE CONTACT) |
| 11 | S5_HAUT | E 0.5 | BOOL | FIN DE MONTEE (OUVERTURE DU CONTACT) |
| 12 | S6_LENTBAS | E 0.6 | BOOL | LENT POUR LA PARTIE INFERIEURE (OUVERTURE DU CONTACT) |
| 13 | S7_LENTHAUT | E 0.7 | BOOL | LENT POUR LA PARTIE SUPERIEURE (OUVERTURE DU CONTACT) |
| 14 | SP_DESCENTE | M 71.1 | BOOL | MEMOIRE POUR DESCENDRE |
| 15 | SP_LENTDESCENTE | M 0.0 | BOOL | MEMOIRE LENT DESCENTE |
| 16 | SP_LENTMONTEE | M 0.1 | BOOL | MEMOIRE LENT MONTEE |
| 17 | SP_MONTEE | M 71.2 | BOOL | MEMOIRE POUR MONTEE |
| 18 | STOP | E 0.3 | BOOL | BOUTON D'ARRET (OUVERTURE DU CONTACT) |
| 19 | TableTest | VAT 1 | | |
| 20 | | | | |

Figure II.20 : Tableau de mnémorique

- **Le bloc de données DB10** : Est la projection du « **Tableau II.4** » dans « SIMATIC Manager », avec une inversion entre l'octet de poids fort 0.0→0.7 et celui du poids faible 1.0→1.7, pour des raisons de programmation. La consigne de vitesse est à l'adresse locale 2.0.

| Adresse | Nom | Type | Valeur initiale | Commentaire |
|---------|----------------------|------------|-----------------|-------------|
| 0.0 | | STRUCT | | |
| +0.0 | SEMI_AUTO_DROITE | BOOL | FALSE | |
| +0.1 | SEMI_AUTO_GAUCHE | BOOL | FALSE | |
| +0.2 | PZD_OUI_NON | BOOL | TRUE | |
| +0.3 | SENS_DROITE_GAUCHE | BOOL | FALSE | |
| +0.4 | LIBRE | BOOL | FALSE | |
| +0.5 | POTENTIOMETRE_AUG | BOOL | FALSE | |
| +0.6 | POTENTIOMETRE_DIM | BOOL | FALSE | |
| +0.7 | COMMANDE_SUR_SITE | BOOL | FALSE | |
| +1.0 | MARCHE_ARRET | BOOL | FALSE | |
| +1.1 | MARCHE_ARRET1 | BOOL | TRUE | |
| +1.2 | MARCHE_ARRET2 | BOOL | TRUE | |
| +1.3 | MODE_LIBRE_BLOQUE | BOOL | TRUE | |
| +1.4 | CONDITION_FONCTION | BOOL | TRUE | |
| +1.5 | DEBLOQUER_GENERATEUR | BOOL | TRUE | |
| +1.6 | LIBRE_CONSIGNE | BOOL | TRUE | |
| +1.7 | AQUITTEMENT | BOOL | FALSE | |
| +2.0 | CONSIGNE_VITESSE | INT | 0 | |
| =4.0 | | END_STRUCT | | |

Figure II.21 : Bloc de données DB10

- **Le bloc de données DB11 :**

C'est la projection du « **Tableau II.5** » dans « SIMATIC Manager », avec une inversion entre l'octet de poids fort 0.0→0.7 et celui du poids faible 1.0→1.7, aussi pour des raisons de programmation. La valeur réelle mesurée de vitesse est à l'adresse locale 2.0.

| Adresse | Nom | Type | Valeur initiale | Commentaire |
|---------|-----------------------|------------|-----------------|-------------|
| 0.0 | | STRUCT | | |
| +0.0 | ECART_CONSIGNE_MESURE | BOOL | FALSE | |
| +0.1 | PILOTAGE_DEMANDE | BOOL | FALSE | |
| +0.2 | FREQUENCE_ATEINTE | BOOL | FALSE | |
| +0.3 | ALARME_COURANT | BOOL | FALSE | |
| +0.4 | FREIN_DE_MAINTIEN | BOOL | FALSE | |
| +0.5 | SURCHARGE_MOTEUR | BOOL | FALSE | |
| +0.6 | DROITE_GAUCHE | BOOL | FALSE | |
| +0.7 | SURCHARGE_VARIATEUR | BOOL | FALSE | |
| +1.0 | DEMARRER_PRET | BOOL | FALSE | |
| +1.1 | MODE_PRET | BOOL | FALSE | |
| +1.2 | MODE_LIBRE_BLOQUE | BOOL | FALSE | |
| +1.3 | DERANGEMENT | BOOL | FALSE | |
| +1.4 | PAS_ARRET2 | BOOL | FALSE | |
| +1.5 | PAS_ARRET3 | BOOL | FALSE | |
| +1.6 | DEMARRER_BLOQUEE | BOOL | FALSE | |
| +1.7 | AVERTISSEMENT | BOOL | FALSE | |
| +2.0 | MESURE_VITESSE | INT | 0 | |
| =4.0 | | END_STRUCT | | |

Figure II.22 : Bloc de données DB11

Remarque : Tous les blocs de programme, seront écrits en langage « **LOG** ». Le code du programme dans ces blocs est donné en annexe

Cette étape de « Création de projet dans STEP7 » va être conclue par la compilation de tous les blocs dans « SIMATIC Manager » et le chargement vers la CPU.

II.8.2.3 Paramétrages du Variateur de vitesse MICROMASTER 440

Les paramètres permettent d'adapter le variateur à une application bien déterminée.

Chaque paramètre est identifié par un numéro de paramètre, un texte de paramètre et par des attributs spécifiques. Le numéro de paramètre est unique dans un système de variateur.

L'accès aux paramètres du MICROMASTER est possible moyennant les panneaux de commande et logiciels suivants :

- **PMU(Parameterization Unit) :**

Le panneau de commande **PMU** sert au paramétrage, à la commande et à la surveillance des variateurs et onduleurs directement au pied de l'appareil. Il fait partie de l'appareil en version de base. Il comporte une visualisation à quatre positions et trois touches.

Il est utilisé pour le paramétrage d'applications simples ne comptant qu'un petit nombre de paramètres à régler ainsi que pour le paramétrage rapide en cours de fonctionnement.

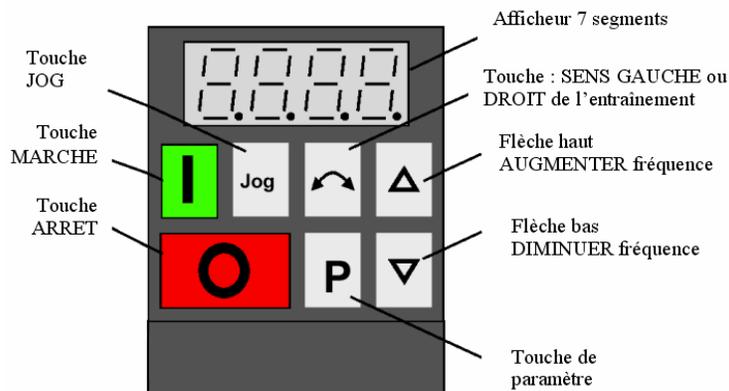


Figure II.23 : La PMU [19]

- **AOP (panneau de commande avancé) :**

Le pupitre opérateur OP1S est un terminal optionnel de saisie/visualisation permettant de procéder au paramétrage et à la mise en service des convertisseurs. Le paramétrage s'effectue de façon confortable grâce à l'affichage en clair sur la visualisation.

L'OP1S dispose d'une mémoire non volatile et est en mesure de stocker de façon permanente des jeux complets de paramètres.

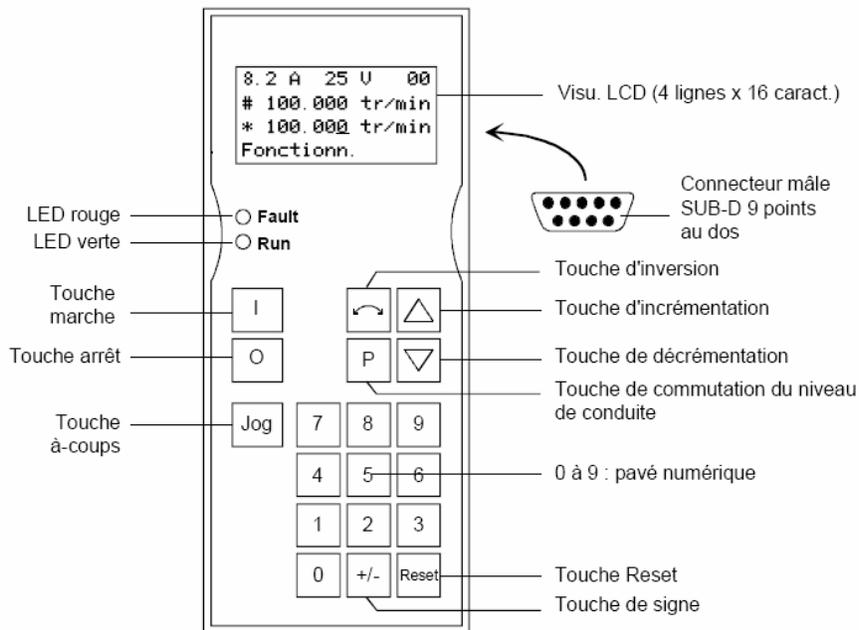


Figure II.24 : Pupitre opérateur OP1S [19]

- **Outil de mise en service de PC avec le progiciel "Drive Monitor"**

Le logiciel DriveMonitor est une interface de paramétrage et de configuration pour tous types de convertisseurs/onduleurs, il réalise un grand nombre de fonctions de commande et de régulation, de fonctions de communication ainsi que de fonctions de diagnostic et de conduite à l'aide de blocs fonctionnels.

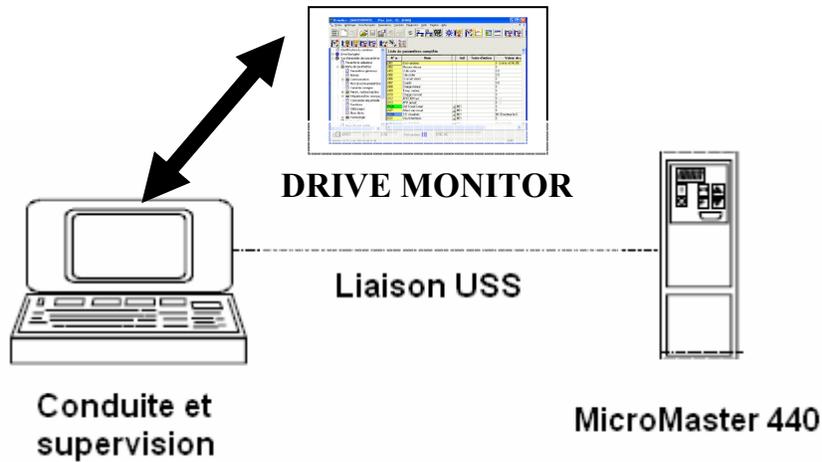


Figure II.25 : Le paramétrage via Drive Monitor

Cette interface se caractérise par sa lisibilité, et cela grâce au navigateur intégré « *Drive Navigator* », qui permet un accès direct et facile à toutes les fonctions qu’offre Drive Monitor.

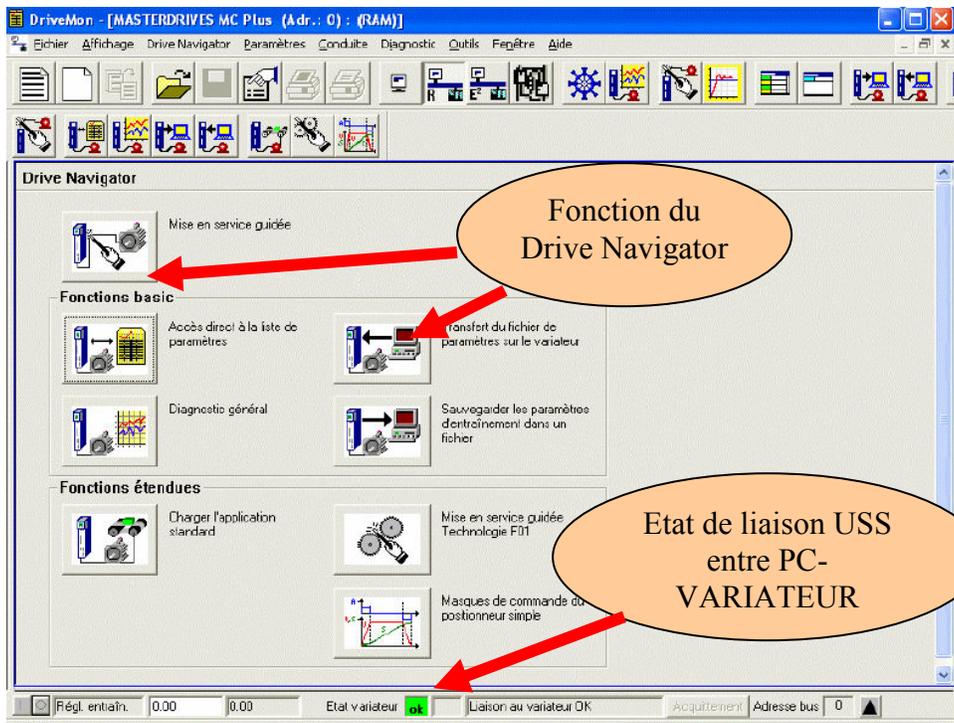


Figure II.26 : Le logiciel DriveMonitor et le menu Drive Navigator

Le protocole USS (Interface série universelle), défini par Siemens, est un protocole simple de transmission série, totalement taillé sur mesure pour les besoins des systèmes d'entraînement.

On décrit dans ce tableau les différents états que peut prendre la liaison USS Pc-Variateur :

| Symbole sur le DRIVE MONITOR | Signification |
|---|---|
|  | Liaison et variateur ok |
|  | Liaison ok, variateur en défaut |
|  | Liaison ok, variateur en alarme |
|  | Le variateur est paramétré en mode offline |
|  | Pas de liaison possible avec le variateur (paramétrage possible seulement en mode offline). |

Tableau II.7 : *Etats possibles pour la liaison PC-Variateur*

Si la liaison avec le variateur ne peut pas être établie en raison de l'absence physique du variateur ou de sa liaison, il est possible d'opter pour un paramétrage hors ligne (offline). Il faut pour cela passer en mode "offline", dans lequel on pourra alors éditer le jeu de paramètres.

Ceci permet de créer un fichier « **Download** » adapté au cas d'application, que l'on pourra charger ultérieurement sur le variateur.

a. Les Méthodes de paramétrages dans DriveMonitor

Pour configurer le Variateur via DriveMonitor, il existe deux approches :

a.1. La Méthode directe de paramétrage :

Le paramétrage direct par le biais de la liste de paramètres reste la méthode de base, et celle qu'on appliquera en cours de cette application.

La liste de paramètres offre les avantages suivants :

- Visibilité simultanée d'un grand nombre de paramètres ;
- Affichage en clair des noms des paramètres, numéros d'indice, texte d'indice, des valeurs;
- Lors d'une modification d'un paramètre : affichage des valeurs limites ou des valeurs possibles.

a.2. La méthode assistée de paramétrage :

On y accède en choisissant la fonction suivante dans le Drive Navigator :



= Mise en service assistée

Cette méthode est favorisée, surtout pour les premières mises en œuvre, par sa simplicité, et elle permet, en quelques étapes, de configurer l'entraînement.

b. Les Types de paramètres dans un variateur de vitesse SIMOVERT de SIEMENS

Dans tous les variateurs de vitesses SIMOVERT du constructeur SIEMENS, on trouve deux types de paramètres :

- **Paramètres de réglage**

Accessibles en écriture et en lecture, ils prennent l'orthographe "**P_XXXX**".

Ces paramètres sont activés/désactivés dans les diverses fonctions ou influent directement sur le comportement d'une fonction. La valeur de ces paramètres est enregistrée dans une mémoire non volatile (**EEPROM**) dans le variateur. On y trouve plusieurs groupes fonctionnels qu'on appelle aussi « **famille de paramètres** », nommé ainsi pour la similitude des tâches qu'on leur associe :

- **Famille des paramètres d'accès au variateur** : de P0003→P0199.
- **Famille des paramètres du variateur** : de P0200→P0299.
- **Famille des paramètres pour caractéristique moteur** : P0300→P0399 et P0600→P0399.
- **Famille des paramètres du capteur de vitesse** : de P0400→P0499.
- **Famille des paramètres de commande et d'ordre E/S TOR** : P0700→P0749 et P0800→P0899.
- **Famille des paramètres pour E/S analogiques** : de P0750→P0799.
- **Famille des paramètres pour le canal de consigne du générateur de rampe** : de P1000→P1199.
- **Famille des paramètres pour les caractéristiques de l'entraînement** : de P1200→P1299.
- **Famille des paramètres pour la commande du moteur** : de P1300→P1799.
- **Famille des paramètres pour la communication** : de P2000→P2099.

- **Paramètres d'observation**

Seulement accessibles en lecture, ils prennent l'orthographe "r_xxxx".

Ces paramètres servent à afficher les grandeurs internes comme par exemple les états ou les mesures. Ces paramètres sont indispensables pour le diagnostic.

On trouve deux familles :

- o Alarmes et surveillances
- o Paramètres pour observation du « PID contrôler ».

- c. **Étapes de paramétrages dans « Drive Monitor »**

Dans cette partie nous allons aborder le paramétrage du **MICROMASTER 440**, de notre application, conformément au cahier de charge.

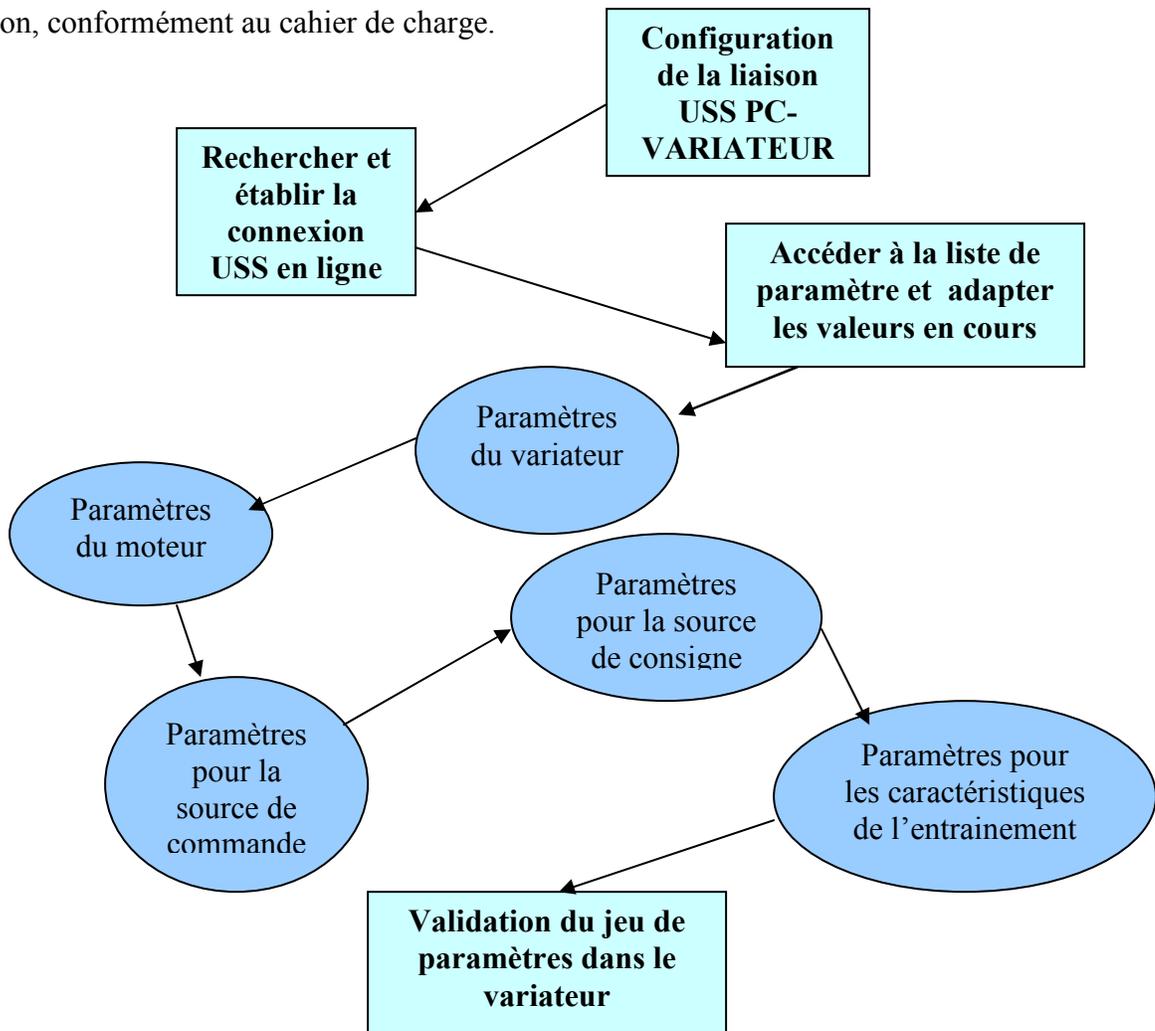


Figure II.27 : Les démarches pour la configuration via DriveMonitor

c.1 Configuration de la liaison USS PC-VARIATEUR :

On accède à la configuration de cette liaison par le chemin :

OUTILS → Paramètres ONLINE

Dans la fenêtre de dialogue Drive Monitor :

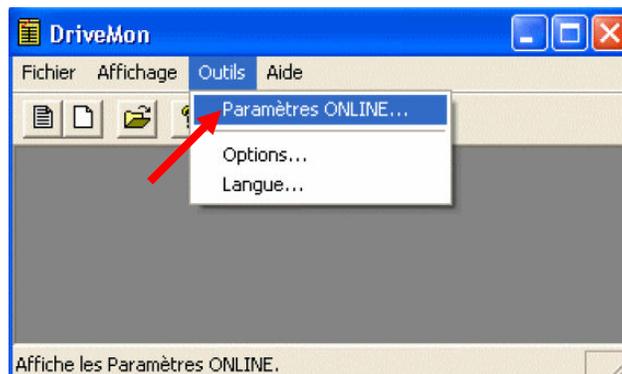
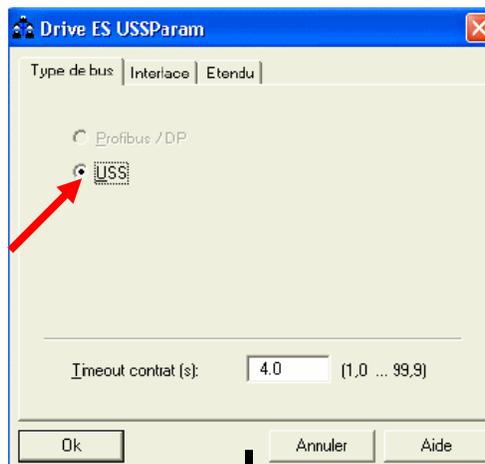


Figure II.28 : Accès à la configuration de la liaison USS PC-Variateur

Dans cette fenêtre, cinq paramètres sont adaptables comme suit:

- o Le type de liaison → **USS**
- o L'interface de communication → **COM**
- o La vitesse de transmission → **9600**
- o Répétition de contrat → **100**
- o Délai de réponse → **40 ms**

Indiqués dans les trois onglets qui suivent :



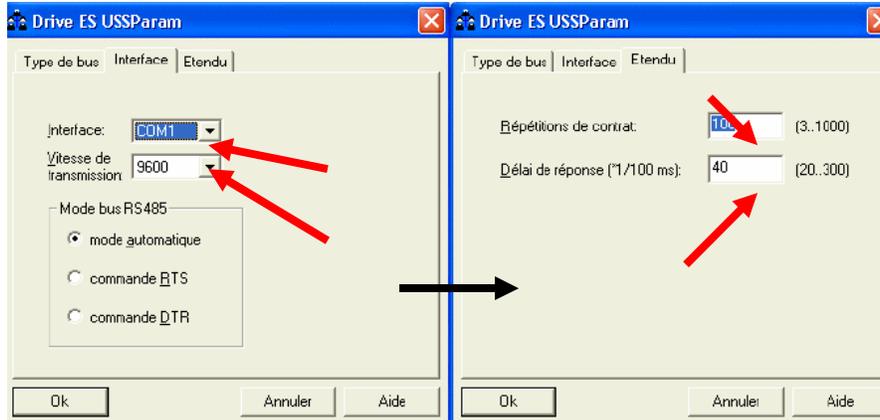


Figure II.29 : Les paramètres de communication USS PC-Variateur

c.2 Rechercher et établir la connexion USS en ligne

Une fois la liaison configurée, il suffit de suivre le chemin :

Fichier → Connexion ONLINE

Pour détecter le variateur en ligne avec le PC.

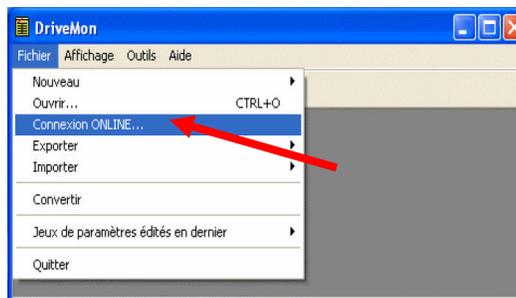


Figure II.30 : Accès aux partenaires réseau USS

La recherche se fait par adresse, et par défaut l'adresse USS du variateur est « 0 ».



Figure II.31 : Détection et identification du variateur dans la liaison USS

Après La détection du variateur, « Drive Monitor» transite directement vers le Menu Drive Navigator.

c.3 Accéder à la liste de paramètres et adapter les valeurs en cours

Dans cette application nous allons suivre la méthode de paramétrage directe, décrite auparavant, et l'accès à la liste des paramètres se fait, comme c'est déjà mentionné par le biais de DRIVE NAVIGATOR, en choisissant la fonction :

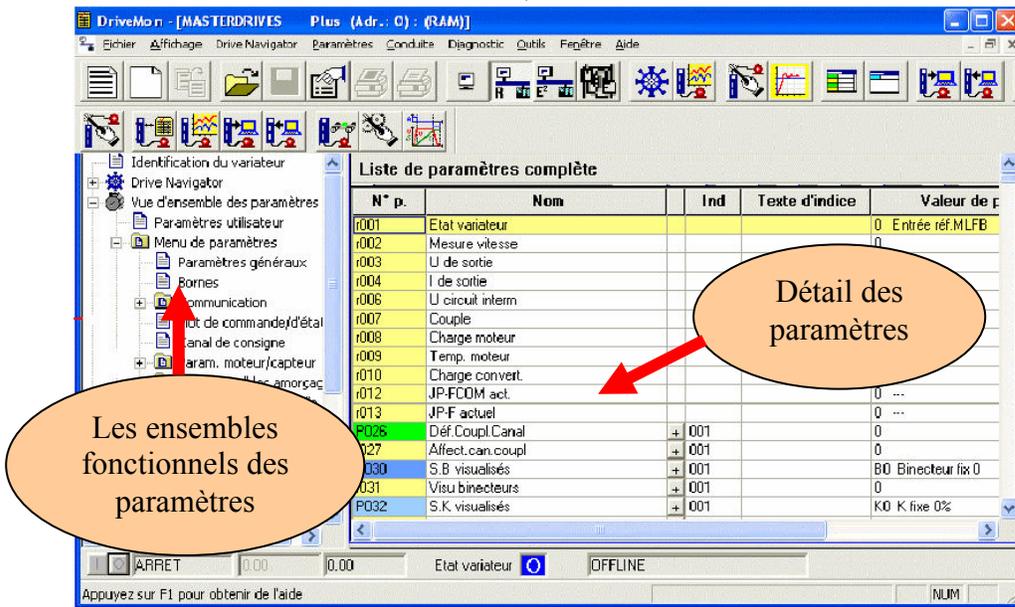
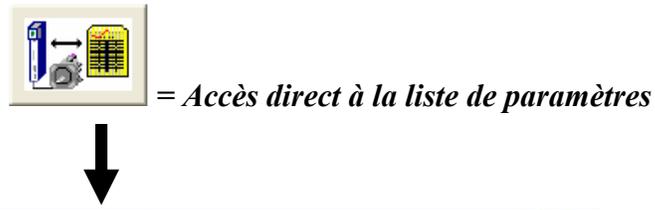


Figure II.32 : Représentation de la liste des paramètres dans **DRIVE MONITOR**

Et cette liste des paramètres offre les détails suivants :

| Colonne | Intitulé | Fonction |
|---------|----------|--|
| 1 | N° P | Numéro du paramètre. Le numéro ne peut être modifié par l'utilisateur que dans le menu « <i>Free parameterization</i> ». |
| 2 | Nom | Nom du paramètre comme indiqué dans la liste des paramètres. |

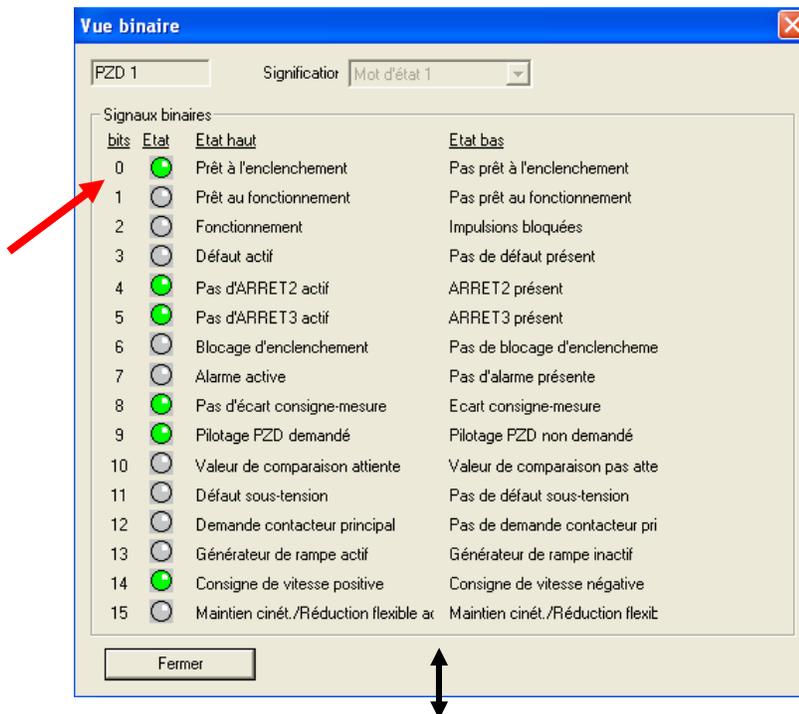
| | | |
|---|----------------------------|---|
| 3 | <i>Ind</i> | Indice d'un paramètre indexé. En affichage standard on ne voit que l'indice 1 ; pour voir les autres indices cliquer sur le symbole [+], l'affichage s'étend alors à tous les indices du paramètre. |
| 4 | <i>Texte d'indice</i> | Signification de l'indice de paramètre. |
| 5 | <i>Valeur de paramètre</i> | Valeur momentanée du paramètre. Modification possible en double-cliquant sur la valeur ou en la sélectionnant puis en actionnant la touche d'entrée. |
| 6 | <i>Dim</i> | Grandeur physique du paramètre s'il en a une. |

Tableau II.8 : Structure de la liste des paramètres

Le jeu de paramètres qu'on va mettre en place, contiendra un nombre de 28 paramètres à adapter, et sera détailler par catégorie sous formes de tableau en annexe.

II.8.2.4 Test de l'application

Pour assurer le bon fonctionnement de l'application, on vérifie à priori l'état des modules API-MICROMASTER440, et de la liaison de communication PROFIBUS-DP, par le biais des voyons d'état qui doivent indiquer « Près à l'enclenchement » :



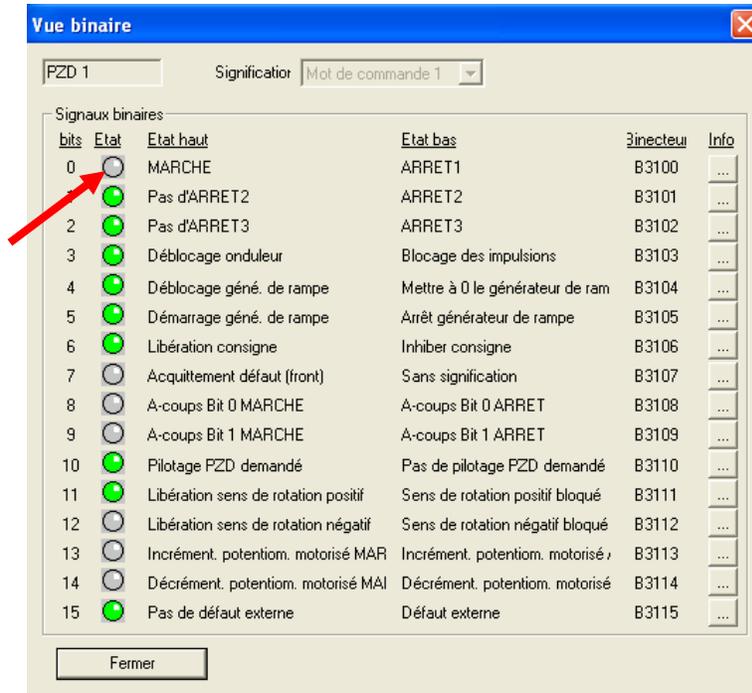


Figure II.33 : Visualisation du mot d'état et du mot de commande dans DRIVE MONITOR

Ensuite on lance l'application en suivant le chronogramme de la *Figure II.9*, on obtient, par la fonction de « traçage » inclus dans *DRIVE MONITOR*, la courbe de vitesse suivante :

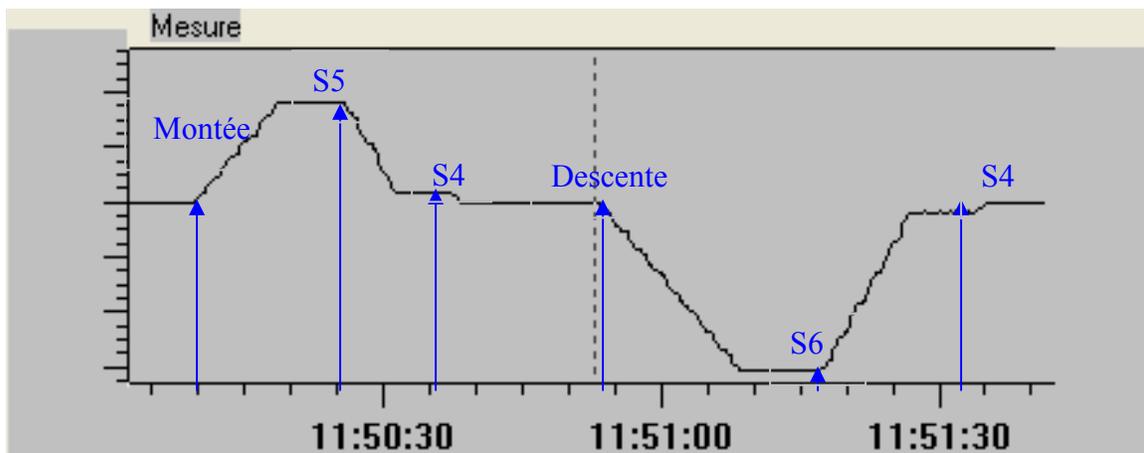


Figure II.34 : Courbe de vitesse

La période d'échantillonnage pour le traçage de cette courbe étant de $T_e = 0.1s$, avec une charge constante sur l'arbre du moteur.

II.9 Conclusion

PROFIBUS propose une architecture de communication ouverte et bien adaptée aux domaines d'application des réseaux locaux industriels, offre une mise en œuvre aisée et un paramétrage flexible, ce qui explique, en grande partie, sa si grande utilisation dans l'industrie.

CHAPITRE III :

Le progiciel de conception des interfaces

Homme/Machine

Win CC flexible HMI

III.1 Introduction

Suite à l'automatisation industrielle, l'opérateur humain a été contraint de **conduire** ou de **superviser** des machines automatisées, en réduisant les prises d'information et les actions directes sur le processus, ce qui conduit à l'élaboration d'interface d'interaction Homme/Machine, flexible et aussi lisible pour un simple opérateur. Le dialogue est d'autant plus facilité que l'écran comporte des images avec des synoptiques, des graphes, des baragraphes...etc.

Siemens avec sa gamme SIMATIC C7, contribue dans ce sens, et met à la disposition de cette demande exigeante, un grand choix en terme de pupitre de supervision et de contrôle, paramétrable par le biais d'outils divers.

Le progiciel *Win CC flexible*, reste le plus approprié, en terme de simplicité, pour la conception ce type d'interface de commande/diagnostic.

III.2 Progiciel de conception et configuration d'interface *Win CC flexible*

III.2.1 Présentation générale

Win CC flexible, est un progiciel partagé dans l'environnement STEP7, et propose pour la configuration de divers pupitres opérateurs, une famille de systèmes d'ingénierie évolutifs adaptés aux tâches de configuration.

Créer l'interface graphique et les variables, c'est pouvoir lire les valeurs du processus via l'automate, les afficher pour que l'opérateur puisse les interpréter, et ajuster, éventuellement, le processus, toujours via l'automate (**Figure III.1**).

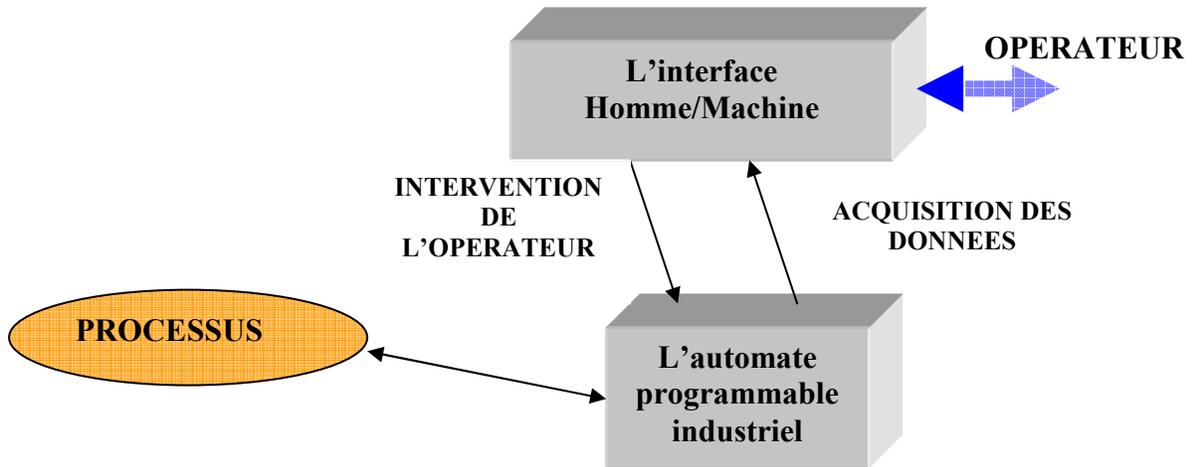


Figure III.1: *L'interface Homme/Machine dans un processus automatisé*

III.2.2 Élément du progiciel Win CC flexible

L'utilisation de ce logiciel pour la conception d'une interface, passe par plusieurs étapes résumé dans la **Figure III.2** :

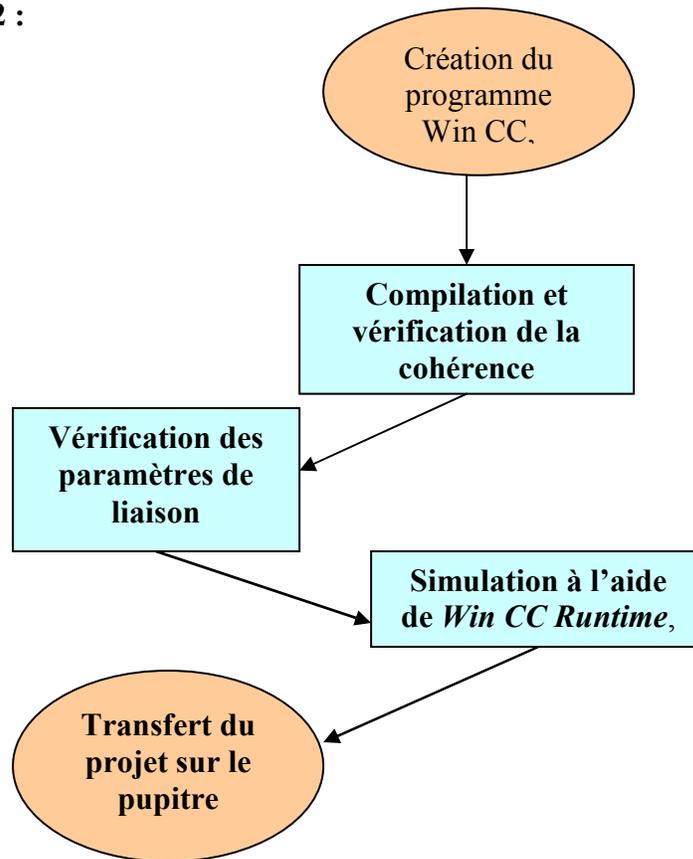


Figure III.2 : *Etapes pour la conception d'une interface via Win CC flexible*

On commence par l'environnement de cet outil, pour terminer expliquer les différentes étapes dans les détails précédemment énumérés.

L'environnement de travail de *Win CC flexible* se compose de plusieurs éléments. Certains de ces éléments sont liés à des éditeurs particuliers, visibles uniquement lorsque cet éditeur est activé. Il met à disposition un éditeur spécifique pour chaque tâche de configuration. On peut configurer p. ex. l'interface utilisateur graphique d'un pupitre opérateur avec l'éditeur "Vues". Pour la configuration d'alarmes, on peut utiliser p. ex. l'éditeur "Alarmes TOR" ou autre.

Dans *Win CC flexible*, chaque projet créé contient principalement des vues que l'on crée pour le contrôle-commande de machines et d'installations. Lors de la création des vues, vous disposez d'objets prédéfinis permettant d'afficher des procédures et de définir des valeurs du processus.

Les différents outils et barres de l'éditeur de vues sont représentés dans la figure qui suit :

- **Barre des menus**

La barre des menus contient toutes les commandes nécessaires à l'utilisation de *Win CC flexible*. Les raccourcis disponibles sont indiqués en regard de la commande du menu.

- **Barres d'outils**

La barre d'outils permet d'afficher tout dont le programmeur a besoin.

- **Zone de travail**

La zone de travail sert à configurer des vues, de façon qu'il soit le plus compréhensible par l'utilisateur, et très facile à manipuler et consulter les résultats.

- **Boîte à outils**

La fenêtre des outils propose un choix d'objets simples ou complexes qu'on insère dans les vues, p. ex. des objets graphiques et éléments de commande.

- **Fenêtre des propriétés**

Le contenu de la fenêtre des propriétés dépend de la sélection actuelle dans la zone de travail, lorsqu'un objet est sélectionné, on peut éditer les propriétés de l'objet en question dans la fenêtre des propriétés.

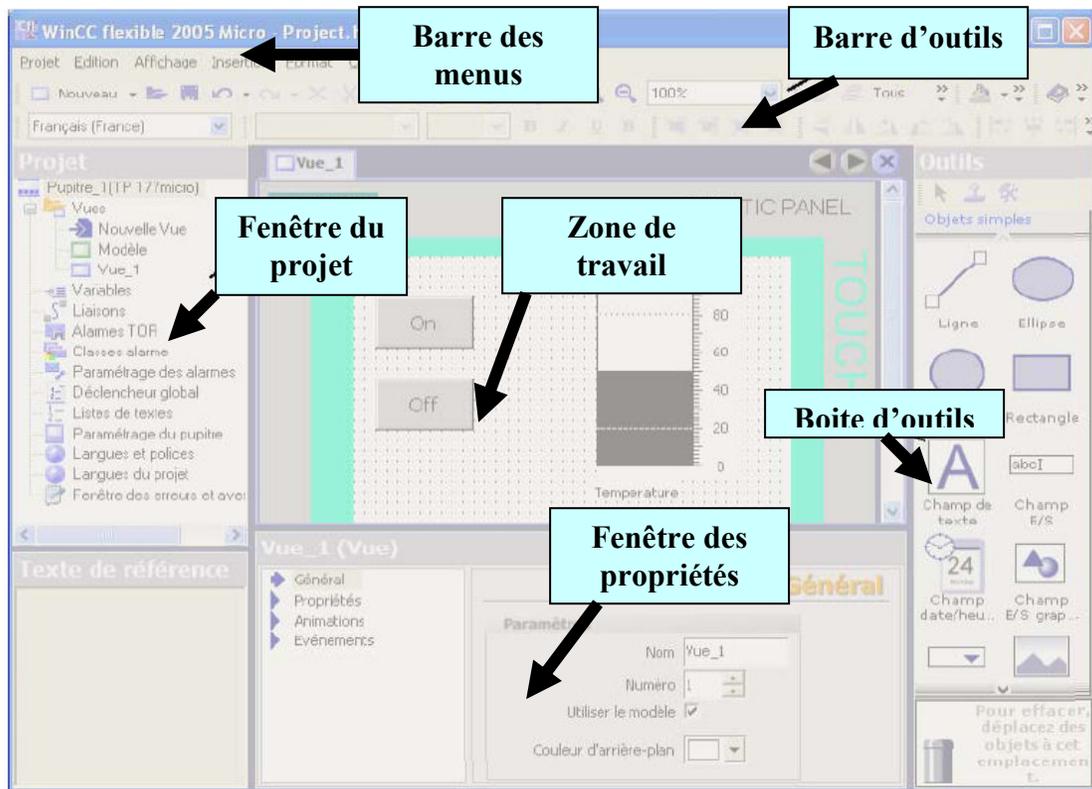


Figure III.3 : Vue d'ensemble du progiciel Win CC flexible

III.2.3 Utilisation des variables

On distingue deux types de variables, les variables externes et les variables internes

Les variables externes permettent de communiquer, c.-à-d. d'échanger des données entre les composants d'un processus automatisé, entre un pupitre opérateur et un automate p. ex.

Une variable externe est l'image d'une cellule mémoire définie de l'automate. L'accès en lecture et en écriture à cette cellule mémoire est possible aussi bien à partir du pupitre opérateur que de l'automate.

Les variables internes ne possèdent aucun lien avec l'automate. Elles sont enregistrées dans la mémoire du pupitre. Le pupitre opérateur en question peut donc accéder en lecture et en écriture aux variables internes. Les variables internes sont créées p. ex. pour exécuter des calculs locaux.

La déclaration des variables s'effectue dans l'éditeur "Variables". Lors de leur déclaration, une configuration de base est attribuée aux variables. L'éditeur "Variables" vous permet d'adapter

la configuration des variables aux besoins de votre projet ; L'éditeur s'ouvre systématiquement par ajout de variable.

| Nom | Liaison | Type de donn... | Adresse | Eléments du tableau | Cycle d'acquisiti... | Comr |
|------------|--------------------|-----------------|---------------|---------------------|----------------------|------|
| Variable_1 | Connection_1 | Int | VW 0 | 1 | 1 s | |
| Variable_2 | Connection_1 | Int | VW 2 | 4 | 1 s | |
| Variable_3 | <Variable interne> | Int | <Pas d'ad...> | 1 | 1 s | |
| Variable_4 | <Variable interne> | Int | <Pas d'ad...> | 1 | 1 s | |

Figure III.4 : éditeur de variable dans le *Win CC flexible*

Chaque variable peut être configurée selon le besoin et cela en accédant à sa fenêtre de priorité après l'avoir sectionnée.

III.2.4 Compilation et Simulation

Après avoir créé le projet et terminé la configuration, il est indispensable de vérifier la cohérence du projet, de contrôler la cohérence et de chercher les erreurs, à l'aide de la commande sur la barre du menu « *contrôle de la cohérence* ». Après le contrôle de cohérence, le système crée un fichier de projet compilé.

La simulation permet de détecter des erreurs logiques de configuration, par exemple, des valeurs limites incorrectes, et cela à l'aide du simulateur *Runtime* par la commande « *démarrer le système Runtime du simulateur* ».

III.3 Conclusion

La description précédente de l'outil *Win CC flexible*, prépare le terrain en vue de l'application « *Réalisation et gestion d'un prototype de station de pompage à base d'automates programmables SIEMENS* », et pendant laquelle une mise en œuvre complète sera décrite pour élaborer une interface Homme/machine.

CHAPITRE IV

Réalisation et gestion d'un prototype de station de pompage à base d'API

SIEMENS

IV.1 Introduction

Les stations de pompages hydrauliques sont primordiales dans les réseaux d'alimentation en eau potable.

La fonction principale est assurée par la pompe, et fonctionne normalement grâce à un environnement constitué de débitmètres, électrovannes, sonde de niveau...etc. Ces accessoires permettent de collecter des informations sur le fonctionnement, et aide à mieux gérer ce dernier.

Dans ce chapitre, nous allons réaliser le prototype d'une station de pompage, ainsi que sa gestion effectuée à base d'un automate programmable Siemens. Cette application est présentée en deux parties, la première décrit le fonctionnement hydraulique d'une station de pompage. La deuxième met en relief le choix de la commande et la technique de régulation afin de développer le programme qui sera implémenté au sein de l'automate pour répondre au dit cahier de charge.

Les étapes de réalisation de notre prototype s'articule sur la :

- Détermination des éléments de l'application pour finaliser le cahier de charge.
- Détermination de la puissance du groupe électropompe du prototype de la station.
- Détermination du type des éléments auxiliaire de l'installation tel que le débitmètre, capteur de niveau, les électrovannes ainsi que la tuyauterie
- fixer le type de commande du variateur de vitesse selon les possibilités offertes par ce dernier.
- Paramétrage de la communication.
- Ecriture du programme pour la gestion de l'installation
- Choisir un mode de régulation et un mode de diagnostic.

IV.2 Présentation des éléments de l'installation

Le prototype de notre installation a été élaboré suivant les phases d'un projet, ces phases sont le design, le développement et la construction. Le design de ce prototype a été réalisé sur le principe de fonctionnement des installations de pompage existantes, et son aspect final a été l'aboutissement de plusieurs idées afin de se rapprocher le plus du fonctionnement d'une

véritable station de pompage en optimisant au plus son encombrement, sa mobilité et son assemblage.

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

IV.2.1 Description du Prototype de la station de pompage

La **Figure IV.1** donne une vue générale de la réalisation, qui permet de pomper l'eau d'un réservoir le refoulant dans un autre réservoir, dès que ce dernier est plein il y'aura permutation des électrovannes et l'eau circule dans un circuit fermé sans arrêter la pompe.

Une station de commande est mise à disposition ainsi qu'un panneau *HMI* de diagnostic.



Figure IV.1 : *Le prototype de station de pompage*

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

L'ensemble des éléments et les caractéristiques du prototype réalisé sont présentés comme suit :

- Le prototype repose sur un support métallique **1.50 m** sur **1m** et d'une hauteur de **1.50m** avec deux parties ; Celle inférieure représente la bêche d'eau d'où l'eau est pompée. La partie supérieure représente le château d'eau pour la distribution (zone de refoulement).
- Les deux réservoirs sont en plexi glace transparent de **0.35m** sur **0.35m** sur une hauteur de **0.60m** qui peut contenir jusqu'à **74Litres** d'eau chacun.
- Une pompe centrifuge **KSB**, couplé à un moteur asynchrone triphasé à cage d'écuruils commandé par un variateur vitesse **SIMOVERT MASTER DRIVE** de **SIEMENS**. La hauteur manométrique de cette pompe était trop importante, nous étions dans l'obligation de la réduire, cette modification est détaillée en **Annexe D.2**.
- Les éléments de la tuyauterie sont en **PEHD** (Polyéthylène Haute Densité) **Tigre** de diamètre **DN40**,
- Les deux vannes manuelles à opercule **DN40**, l'une est utilisée simplement pour vidanger le 2nd réservoir, l'autre vanne est utilisée pour simuler des perturbations externes de débit.
- Des brides DN40 métallique pour le raccordement au débitmètre.
- Un débitmètre électromagnétique **SITRANS F M MAGFLO MAG 5100 W** de **SIEMENS** à la sortie analogique 4-20mA, alimenté en 24 V, DN 40 à brides. Il mesure le débit et le transmet vers l'automate S300.
- Un capteur de niveau ultrasonique électromagnétique **SITRANS PROBE LU** de **SIEMENS** avec sortie analogique 4-20mA alimenté en 24V.
- Une poire pour le niveau bas dans la bêche (Réservoir source du bas), pour éviter la marche à sec de la pompe.
- Des électrovannes DN40 **TOR** alimentées en 24V DC.
- Un variateur de vitesse de type **SIMOVERT MASTERDRIVE** pour varier la vitesse de l'électropompe, et ainsi le débit.
- Un panneau de supervision et diagnostique **MP270**.

D'une manière générale, la figure qui suit illustre la disposition de chaque élément :

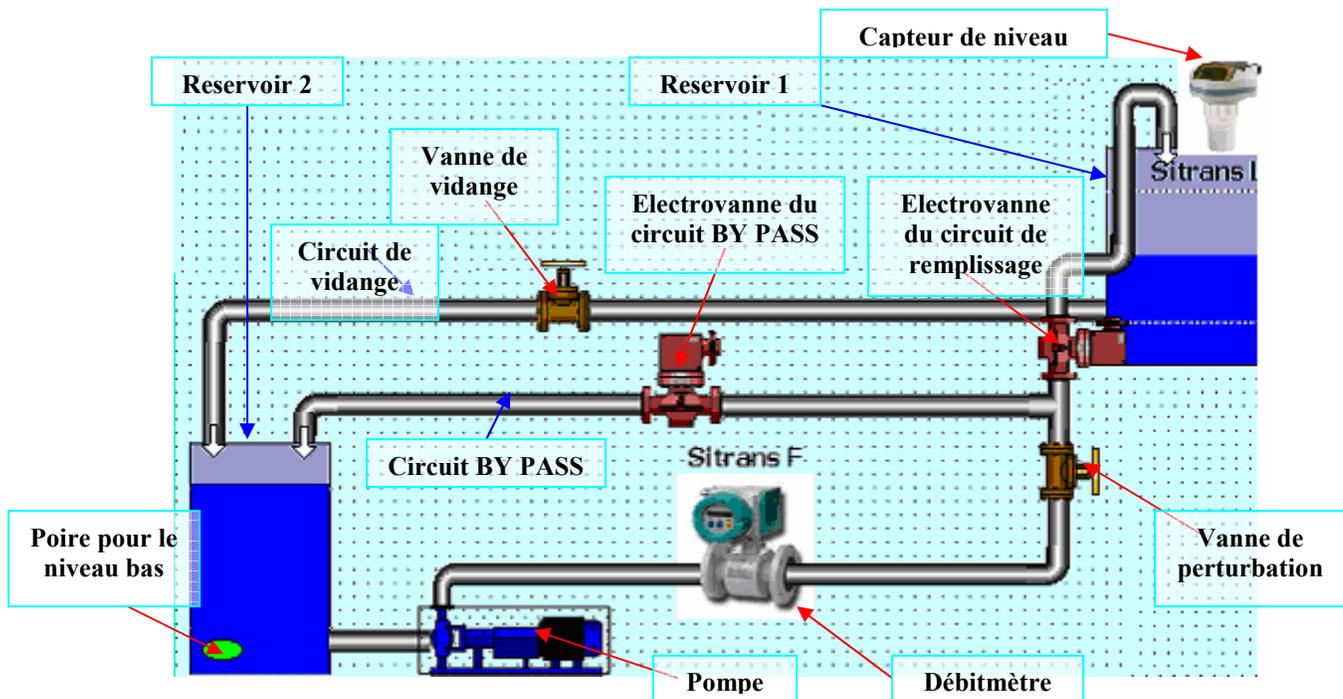


Figure IV.2 : Disposition des éléments dans le prototype de station de pompage hydraulique

IV.2.2 Éléments de commande, de diagnostic et de supervision du Prototype de la station de pompage

La paire Automate/Variateur constitue les deux éléments principaux pour la gestion et l'entraînement dans les installations de pompage.

Pour cette application nous avons utilisé un automate doté des caractéristiques suivant :

- Une **CPU 314 2-DP**, doté d'une mémoire de travail 48Ko, d'un module d'entrées logiques DI24/DO16; d'un module d'entrées/sorties analogiques AI5/AO2 intégrées et de ports MPI+ DP (maître ou esclave DP).
- Une alimentation **S307 5A** alimenté en monophasé 220 V et qui délivre du 24V.
- Modules d'entrées/sorties, on utilise :
 - Module d'entrées/sorties logiques pour la commande des électrovannes.

- Module d'entrées/sorties analogiques pour la mesure de niveau et de débit.
- Une carte CP de communication, pour la communication par télétransmission avec S7 200 via le modem *SATEL* par le protocole *MODBUS*.
- Panneau de supervision *Multi panel 270* est mis à disposition configuré selon le besoin et programmée avec le logiciel *HMI Win CC flexible*, pour la vision des états de chaque élément et le diagnostic pour toute l'installation.

IV.3 Développement du projet pour gestion du prototype de la station de pompage

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

La **Figure IV.3** donne l'organigramme du développement de notre projet, qui consiste à la création du projet, la configuration matérielle, l'écriture du programme ainsi que la création de l'interface Homme/Machine. Le programme développé traduit le cahier des charges établies pour un fonctionnement optimal de notre prototype, et qui se rapproche le plus de celui d'une station de pompage réelle. L'étape finale consiste au paramétrage du variateur et aux tests du programme.

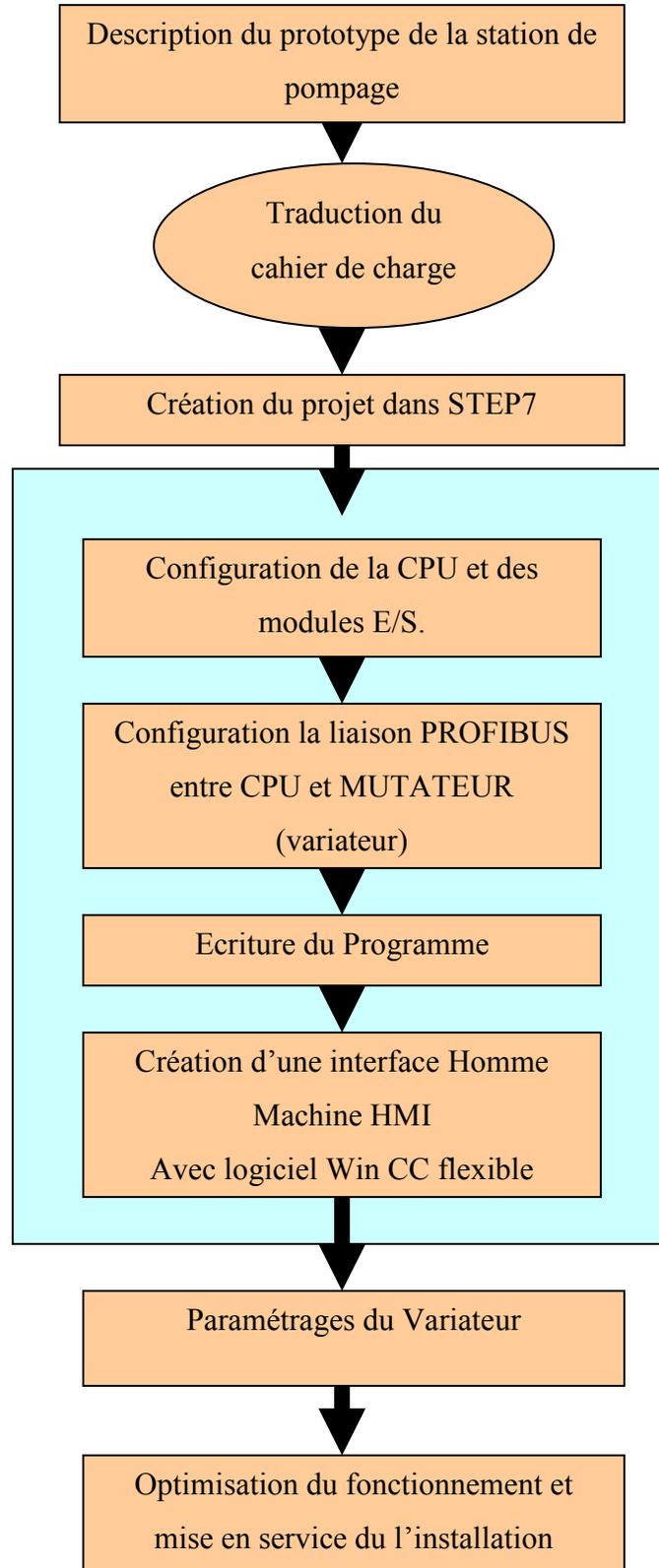


Figure IV.3 : *Organigramme de l'application finale*

IV.3.1 Définition du cahier des charges

Le cahier des charges a été défini comme suit :

a. La mise en marche de la station

Dès la mise en marche du groupe électropompe, le moteur démarrera pour atteindre une vitesse minimale qui sécurise la pompe contre la marche à vide et accompli la phase de remplissage de la tuyauterie qui est de **20Hz**, Si on constate qu'il n y pas de débit, pendant cette phase, on arrête la pompe, pour remédier à d'éventuel problèmes inhérents aux matériels (inversion des phase dans la pompe, etc.)

b. Gestion de la station, la communication et la transmission des grandeurs

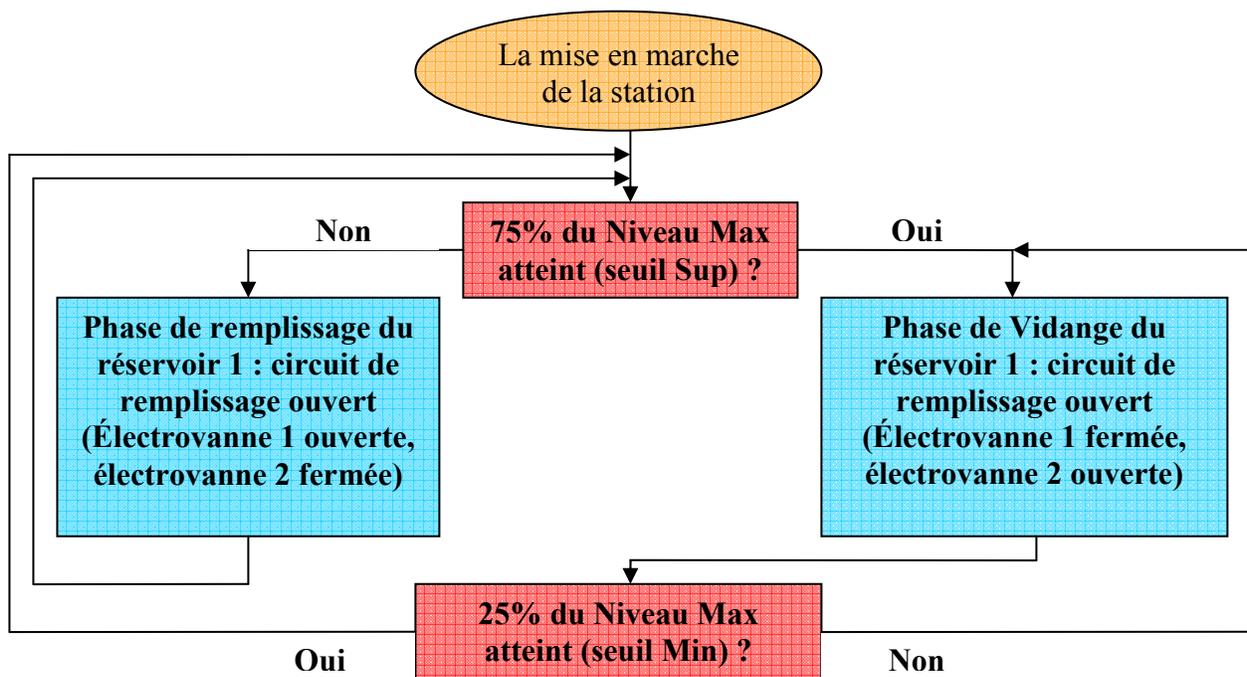


Figure IV.4 : Organigramme fonctionnel de l'application finale

Un réseau **PROFIBUS-DP** avec les éléments **CPU314/MASTERDRIVE-MP270** qui assurera la communication et l'échange de données pour la gestion comme suit :

La **CPU314** est configurée en maître, **MP270** et **MASTERDRIVE** en esclave

Le pilotage des deux électrovannes est assuré par la **CPU 314**, par le biais de son module digitale de sortie, en TOR 0-24V.

La poire du niveau bas, pour éviter la marche à sec, est un contact simple en TOR 0-24V, et son acquisition se fait au niveau de module d'entrée digitale de la CPU314.

La transmission de la mesure de débit se fait en analogique 4-20mA au niveau du module d'entrée analogique de la CPU314 du S300.

L'acquisition de la grandeur niveau se fait en 4-20mA au niveau du module d'entrée analogique du S200, et pour la transmettre vers la S300, on a utilisé un réseau Maître/esclave basée sur le protocole MODBUS, et cela en utilisant le principe de la télétransmission des données par Modem radio **SATEL**, de la S200 vers la S300.

Le schéma suivant résume ce qui était dis précédemment :

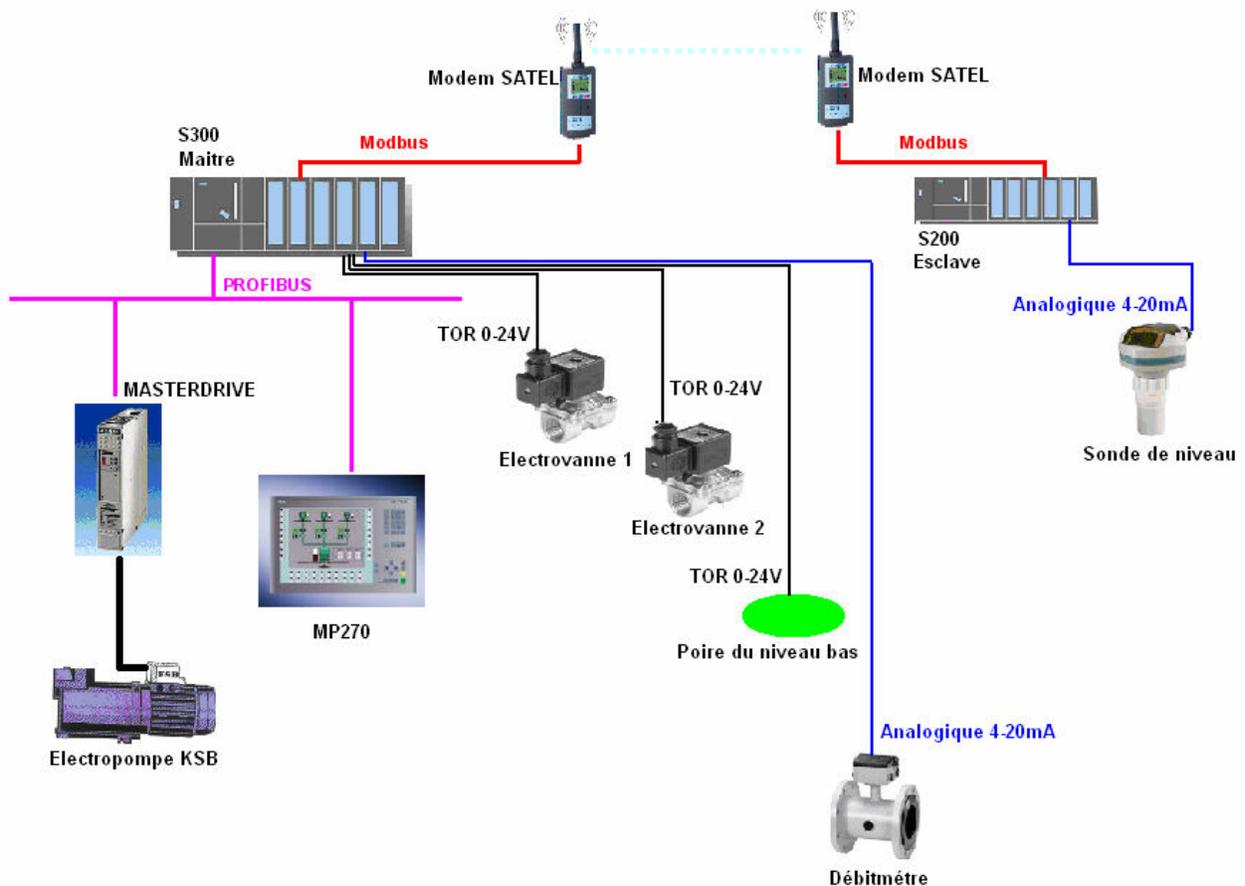


Figure IV.5 : Synoptique du prototype

c. Procédure de régulation du débit

Après avoir ajusté les paramètres pour la commande en $U/f = \text{constante}$ dans le master drive de la vitesse du moteur d'entraînement de la pompe, ceci va constituer la première boucle de régulation interne de vitesse. La deuxième boucle externe, en cascade par rapport à la première, sera pour la régulation de débit dans l'installation, en s'appuyant sur les mesures de débit récolté par le S300, à partir du débitmètre (**Figure IV.5**)

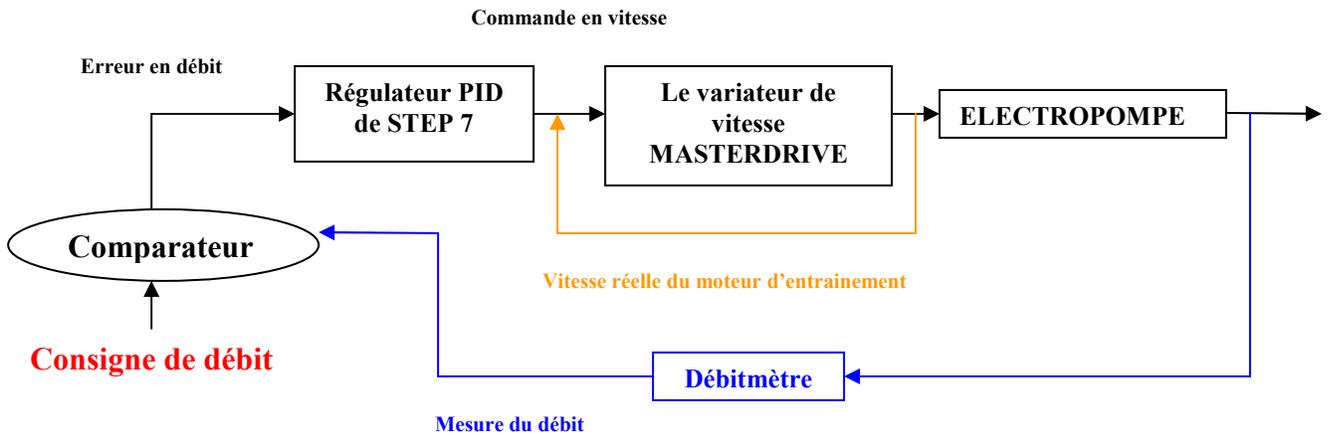


Figure IV.6 : Principe de régulation en cascade Vitesse/Débit

d. Génération de perturbation

Dans le but de tester la régulation, les perturbations peuvent être générés soit par le système lors la commutation (ouverture /fermeture) des électrovannes, soit par une perturbation externe simulée grâce à la vanne manuelle.

IV.4 Création du projet S7 pour la station de pompage :

IV.4.1 Insertion des stations dans *SIMATIC MANAGER*

Le projet est créé comme selon la procédure vue dans le *chapitre I*. Après insertion de la station SIMATIC300 ainsi qu'une station *HMI* pour la supervision telle qu'il est montré dans la figure suivante :

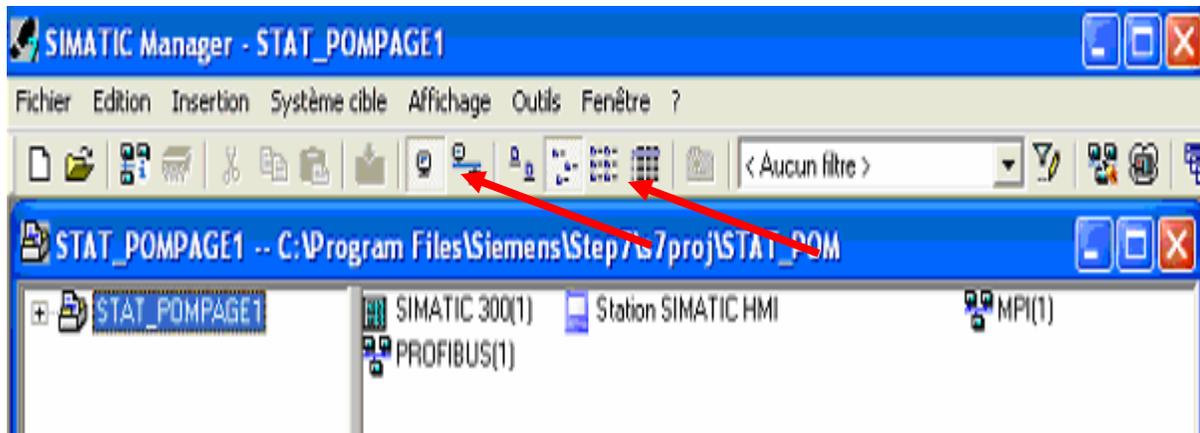


Figure IV.7 : *Création du projet Station Pompage*

IV.4.2 Configuration du matériel dans STEP7

On sélectionne une **CPU 314 2-DP**, avec les caractéristiques citées précédemment, raccordée au **MASTERDRIVE** (Variateur de vitesse) via un réseau **PROFIBUS DP** à l'adresse 3, Le **SIMOVERT MASTER DRIVE** qui apparaît dans le catalogue des esclaves **DP** est enfilé sur le réseau et mis à l'adresse 7, la vitesse de transmission étant de 1.5Mbits/s comme il est montré dans la figure (**Figure IV.8**)

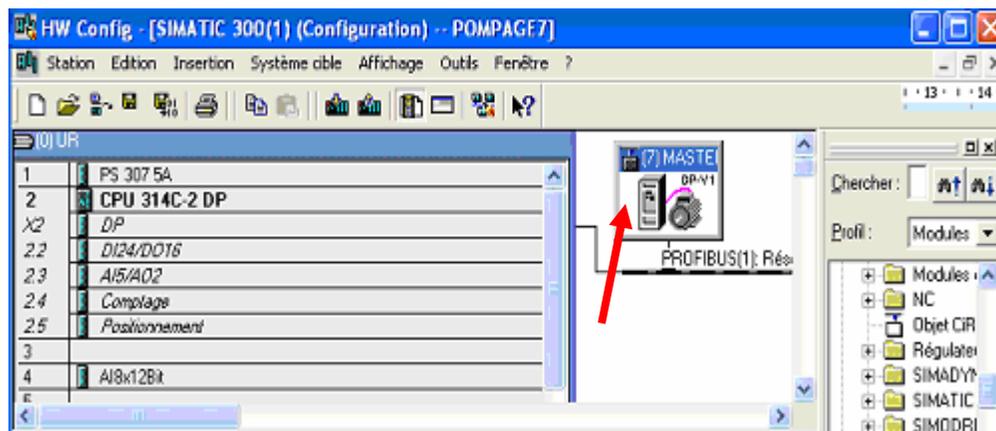


Figure IV.8 : *Configuration matériel*

Les deux figures qui suivent montrent les blocs de donnée pour la commande et l'état :

| Adresse | Nom | Type | Valeur initiale | Commentaire |
|---------|----------------------|------------|-----------------|-------------|
| 0.0 | | STRUCT | | |
| +0.0 | MARCHE_ARRET | BOOL | FALSE | |
| +0.1 | ARRET2 | BOOL | FALSE | |
| +0.2 | ARRET3 | BOOL | TRUE | |
| +0.3 | IMPULS_LIBEREE | BOOL | FALSE | |
| +0.4 | LIBERATION_RAMPE | BOOL | FALSE | |
| +0.5 | DEMARRAGE_RAMPE | BOOL | FALSE | |
| +0.6 | CONSIGNE_LIBEREE | BOOL | FALSE | |
| +0.7 | ACQUITTEMENT | BOOL | FALSE | |
| +1.0 | MARCHE_A_COUP_D | BOOL | FALSE | |
| +1.1 | MARCHE_A_COUP_G | BOOL | TRUE | |
| +1.2 | COMMANDE_DU_PLC | BOOL | TRUE | |
| +1.3 | VALIDER_SENS_POSITIF | BOOL | TRUE | |
| +1.4 | VALIDER_SENS_NEGATIF | BOOL | TRUE | |
| +1.5 | INC_POT_MOT | BOOL | TRUE | |
| +1.6 | DEC_POT_MOT | BOOL | TRUE | |
| +1.7 | PAS_DEFAULT_EXTERNE | BOOL | FALSE | |
| +2.0 | CONSIGNE_VITESSE | INT | 0 | |
| =4.0 | | END_STRUCT | | |

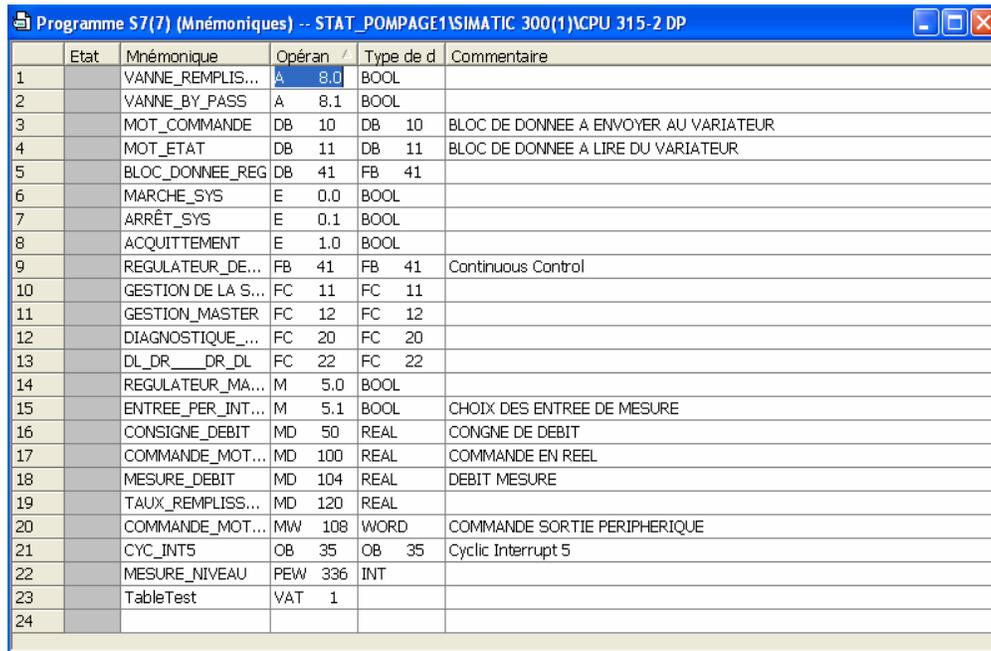
Figure IV.9 : Bloc de donnée pour la commande du variateur

| Adresse | Nom | Type | Valeur initiale | Commentaire |
|---------|--------------------------|------------|-----------------|-------------|
| 0.0 | | STRUCT | | |
| +0.0 | PRET_ENCLENCHEMENT | BOOL | FALSE | |
| +0.1 | PRET_FONCTIONNEMENT | BOOL | FALSE | |
| +0.2 | FONCTIONNEMENT | BOOL | FALSE | |
| +0.3 | DEFAULT_VAR_EN_COURS | BOOL | FALSE | |
| +0.4 | ARRET2_ACTIVEE | BOOL | FALSE | |
| +0.5 | ARRET3_ACTIVEE | BOOL | FALSE | |
| +0.6 | BLOCAGE_ENCLENCHEMENT | BOOL | FALSE | |
| +0.7 | ALARME_ACTIVEE | BOOL | FALSE | |
| +1.0 | ECART_CSG_MESURE | BOOL | FALSE | |
| +1.1 | COMMANDE_PZD | BOOL | FALSE | |
| +1.2 | HZ_MAX_ATTEINTE | BOOL | FALSE | |
| +1.3 | DEFAULT_SOUS_TENSION | BOOL | FALSE | |
| +1.4 | CONTACT_PRINCIPAL_COMMAN | BOOL | FALSE | |
| +1.5 | GENERATEUR_RAMPE_ACTIF | BOOL | FALSE | |
| +1.6 | CONSIGNE_VITESSE_NEGATIF | BOOL | FALSE | |
| +1.7 | MAINTIENT_CINETIQU_ACTIF | BOOL | FALSE | |
| +2.0 | MESURE_VITESSE | INT | 0 | |
| =4.0 | | END_STRUCT | | |

Figure IV.10: Bloc de donnée pour l'état du variateur

IV.4.3 Création du programme de gestion dans STEP7

Le programme S7-300 est créé pour gérer l'installation et pouvoir commander le moteur, il nous est indispensable de créer la table des mnémoniques pour utiliser la méthode d'adressage relatif :



| | Etat | Mnémonique | Opéran | Type de d | Commentaire |
|----|------|--------------------|---------|-----------|---------------------------------------|
| 1 | | VANNE_REMPLIS... | A 8.0 | BOOL | |
| 2 | | VANNE_BY_PASS | A 8.1 | BOOL | |
| 3 | | MOT_COMMANDE | DB 10 | DB 10 | BLOC DE DONNEE A ENVOYER AU VARIATEUR |
| 4 | | MOT_ETAT | DB 11 | DB 11 | BLOC DE DONNEE A LIRE DU VARIATEUR |
| 5 | | BLOC_DONNEE_REG | DB 41 | FB 41 | |
| 6 | | MARCHE_SYS | E 0.0 | BOOL | |
| 7 | | ARRÊT_SYS | E 0.1 | BOOL | |
| 8 | | ACQUITTEMENT | E 1.0 | BOOL | |
| 9 | | REGULATEUR_DE... | FB 41 | FB 41 | Continuous Control |
| 10 | | GESTION DE LA S... | FC 11 | FC 11 | |
| 11 | | GESTION_MASTER | FC 12 | FC 12 | |
| 12 | | DIAGNOSTIQUE_... | FC 20 | FC 20 | |
| 13 | | DL_DR___DR_DL | FC 22 | FC 22 | |
| 14 | | REGULATEUR_MA... | M 5.0 | BOOL | |
| 15 | | ENTREE_PER_INT... | M 5.1 | BOOL | CHOIX DES ENTREE DE MESURE |
| 16 | | CONSIGNE_DEBIT | MD 50 | REAL | CONGNE DE DEBIT |
| 17 | | COMMANDE_MOT... | MD 100 | REAL | COMMANDE EN REEL |
| 18 | | MESURE_DEBIT | MD 104 | REAL | DEBIT MESURE |
| 19 | | TAUX_REMPLISS... | MD 120 | REAL | |
| 20 | | COMMANDE_MOT... | MW 108 | WORD | COMMANDE SORTIE PERIPHERIQUE |
| 21 | | CYC_INT5 | OB 35 | OB 35 | Cyclic Interrupt 5 |
| 22 | | MESURE_NIVEAU | PEW 336 | INT | |
| 23 | | TableTest | VAT 1 | | |
| 24 | | | | | |

Figure IV.11 : *Table des mnémoniques*

On répartira le programme de notre application en plusieurs sous programme et fonctions :

- Le bloc d'organisation cyclique **OB1** comportera essentiellement l'appel aux fonctions :
 - Le **FC11** comportera le programme de gestion de la station.
 - Le **FC12** comportera le programme de gestion du *MASTERDRIVE* et cela à la mise en communication et l'envoi du mot de commande et la réception du mot d'état.
 - Le **FC21** Gestion de l'interface homme machine HMI
- L'**OB35** cyclique qui fait appel au FB41 de régulation « CONT-C » pour réguler le débit de l'installation avec un temps de cycle d'échantillonnage estimé **Te = 10ms**.
- Les différents **OB** d'alarme : **OB121, OB 122, OB 86**.
- Une table de variable pour visualiser les différents entrées/sorties et les variables utilisés dans le programme.

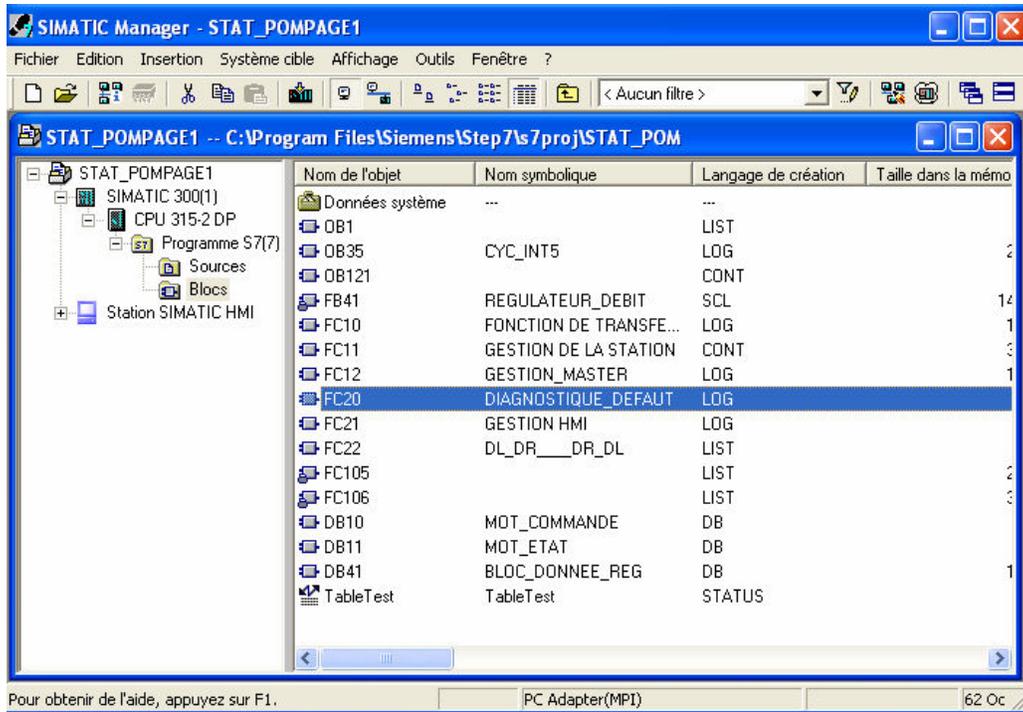


Figure IV.12: *Création des différents blocs de programme dans SIMATIC MANAGER*

IV.4.4 L'interface homme machine HMI configurée par le progiciel « Win CC flexible »

L'interface homme machine, *MP270*, est un terminal d'exploitation de 10 pouces, ou il est possible de visualiser l'état du processus et de ces différentes grandeurs dans notre application, et de détecter les éventuels erreurs de fonctionnement et les anomalies, ainsi que différentes courbes, tel que la consigne de débit, la mesure de débit, et la commande...etc.



Figure IV.13 : *Le terminal d'exploitation MP270*

a. Création de la station HMI

Dans le projet POMPAGE on introduit un nouvel objet qui est la station HMI en choisissant le type de pupitre sur lequel, les informations seront transmises, pour notre application on utilise un *Multi Panel MP 270 10'* en liaison avec l'automate qui a l'adresse 1.

Pour la gestion de cette station on crée les vues suivantes.

- Une vue nommée « *POMPAGE* » pour la gestion des composants de la station hydraulique, et la visualisation.
- Une Vue nommée « *Courbes débit* » pour la visualisation de la consigne de débit, la mesure de débit et la commande.
- Une Vue nommée « *Régulateur* » pour modifier les paramètres du régulateur de débit.
- Une Vue nommée « *Etat variateur* » qui reflète l'état du variateur et les erreurs et alarmes.
- Une Vue nommée « *Réseau* » qui nous permet de diagnostiquer l'état du réseau et de tout ces éléments.

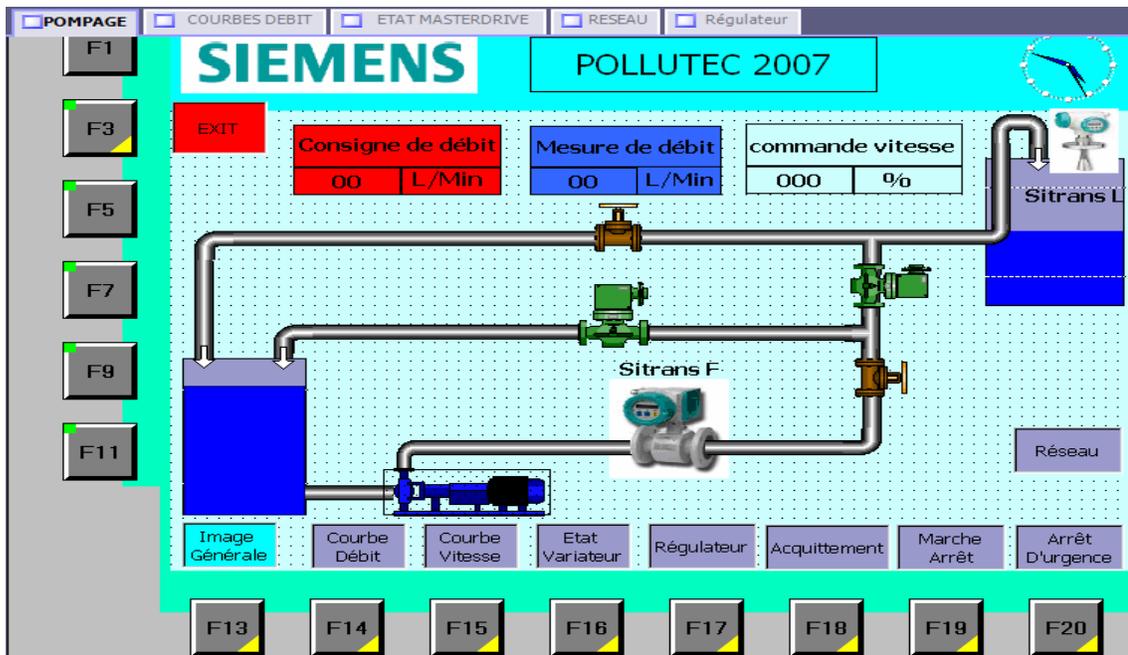


Figure IV.14: Vue générale créée pour la HMI

Les éléments représentatifs de la station sont utilisés à partir de la bibliothèque du *WinCC flexible*, et chaque composant est affecté à une variable, une adresse, une représentation et un événement qui lui est assigné.

Les touches « *F* » correspondent au passage d'une vue à une autre.

La représentation de chaque composant et l'événement correspondant, sont paramétrable dans la propriété de l'objet.

b. Etablissement de la liaison Automate-Panneau MP 270

La liaison est établie en choisissant le protocole de communication qui est dans notre cas **PROFIBUS-DP**, avec une vitesse de transmission **1.5Mb/s**, qui est normalisé sur tout le réseau **API-MASTERDRIVE-MP270** tout en respectant les adresses de chaque élément.

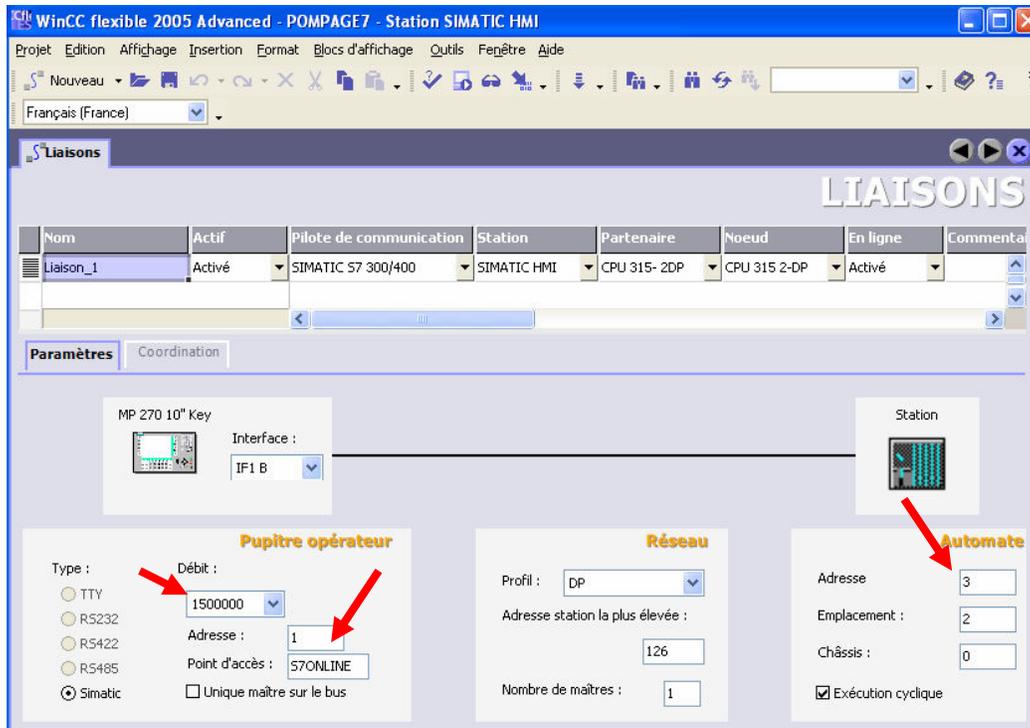


Figure IV.15 : Configuration de la liaison Panel-Automate

IV.4.5 Paramétrage du Master Drive

Pour cette application on choisira le deuxième type de paramétrage qui est le paramétrage par assistance guidé en ligne avec le logiciel Drive Monitor, tel qu'il est montré dans la figure qui suit , pour toute modification ou ajustement on aura recours à la liste complète des paramètres.

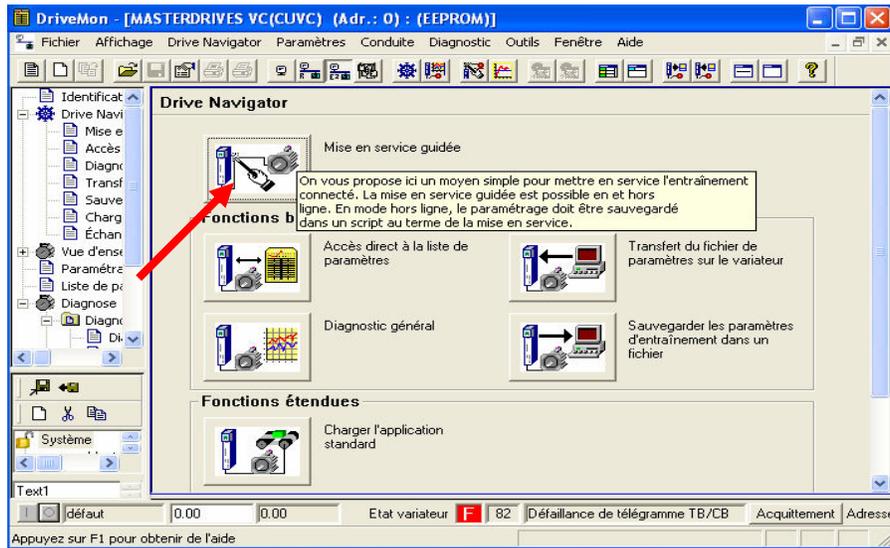


Figure IV.16 : logiciel de paramétrage DRIVE MONITOR

En choisissant la mise en service guidé, nous paramétrons le variateur selon le besoin et la tache, après identification de la série du variateur, nous introduisons les informations du moteur conformément à la plaque signalétique du moteur de la pompe :

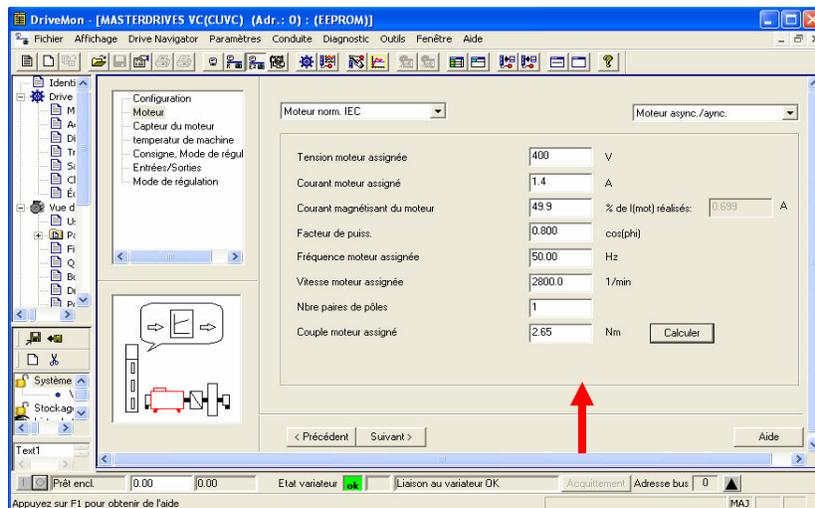


Figure IV.17 : Identification du moteur

Dans l'étape suivante on passera au choix du mode de régulation et la provenance de la commande. Dans notre cas, la commande est scalaire U/f sans capteur de vitesse, la communication entre automate et le *MASTERDRIVE* est via **PROFIBUS-DP**.

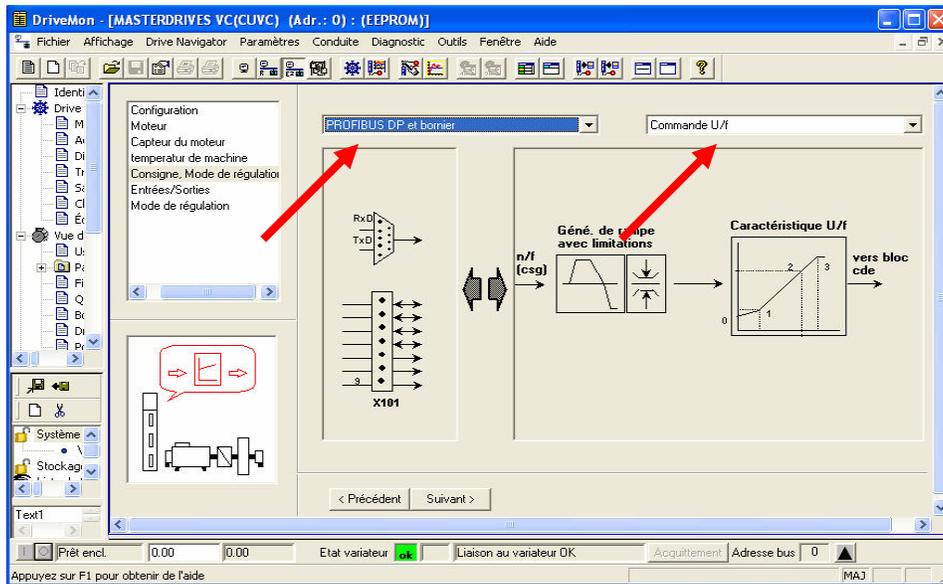


Figure IV.18 : Mode de régulation et type de commande

Dans ce qui suit, nous assignons les entrées/sorties du bornier, l'adresse du **PROFIBUS** qui doit correspondre à l'adresse assigné lors de la configuration du matériel **STEP7** dans l'automate :

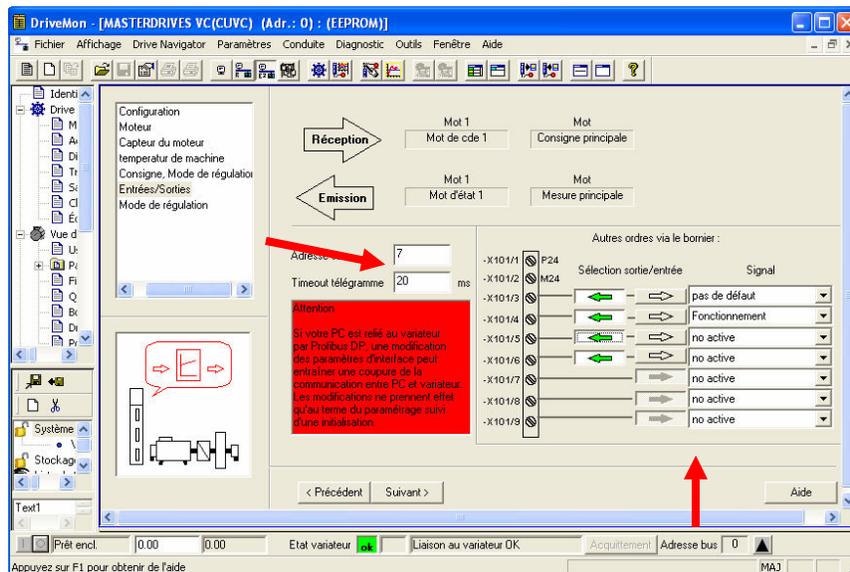


Figure IV.19 : Assignment des entrées/sorties et d l'adresse PROFIBUS -DP

La dernière étape est le paramétrage du mode de régulation, pour affecter essentiellement, le temps de montée jusqu'à une vitesse de référence (rampe d'accélération = gradient positif de la fréquence / temps), ainsi que la rampe de décélération (gradient négatif de la fréquence / temps), on programme une vitesse de référence et le temps pour passer de cette vitesse à zéro :

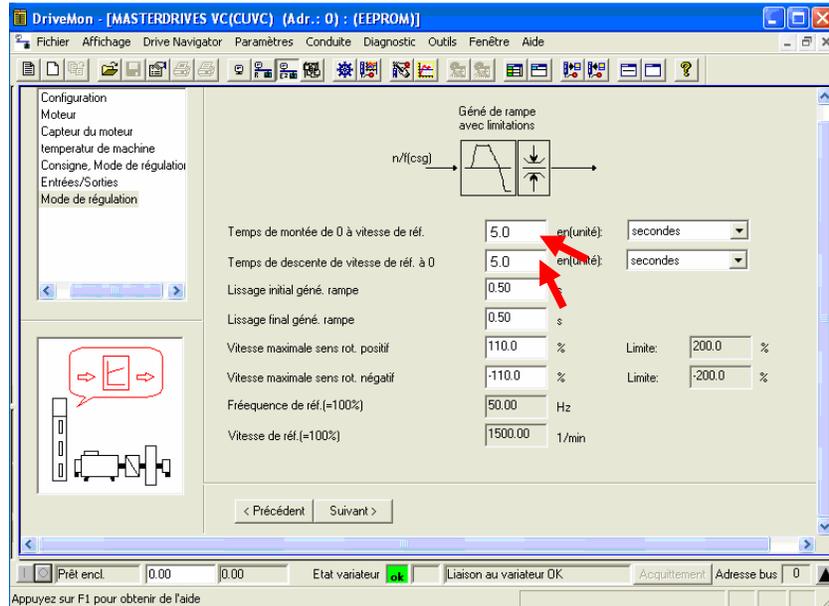


Figure IV.20 : Paramétrage du mode de régulation en U/f

Le paramétrage assisté étant terminé, il suffit de le valider et transférer vers le variateur par la liaison USS :

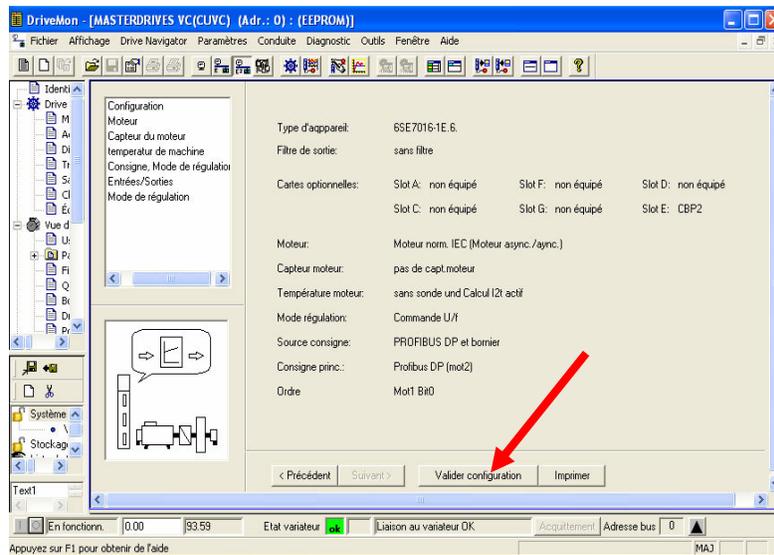


Figure IV.21: Validation de la configuration

Le programme étant chargé dans la CPU, la liaison établie, et le variateur est prêt au fonctionnement, il reçoit un double mot de commande de l'API avec la consigne de vitesse et se met en marche renvoyant un mot d'état du variateur vers l'API.

Nous pouvons visualiser cela à travers le Diagnostic Général du **MASTERDRIVE** dans le **DRIVEMONITOR**, et précisément le diagnostic du bus, qui nous renvoie l'adresse du bus de communication, sa vitesse de transmission, son type de carte optionnelle ainsi que le mot de commande et le mot d'état.

Le Diagnostic général nous renvoie aussi aux défauts et alarmes lors d'un mauvais fonctionnement du variateur, ça nous permet d'une manière très efficace de remédier à ces erreurs et les corriger en temps minimal.

IV.5 Optimisation des fonctions de régulations et mise en service de la station de pompage

Afin de réguler le débit de l'eau à la sortie de la pompe, et d'optimiser les performances, on adapte les paramètres du régulateur *P.I* dans le bloc de paramétrage du **STEP7** tel qu'il est

Montré ci-dessous ou directement dans station HMI :

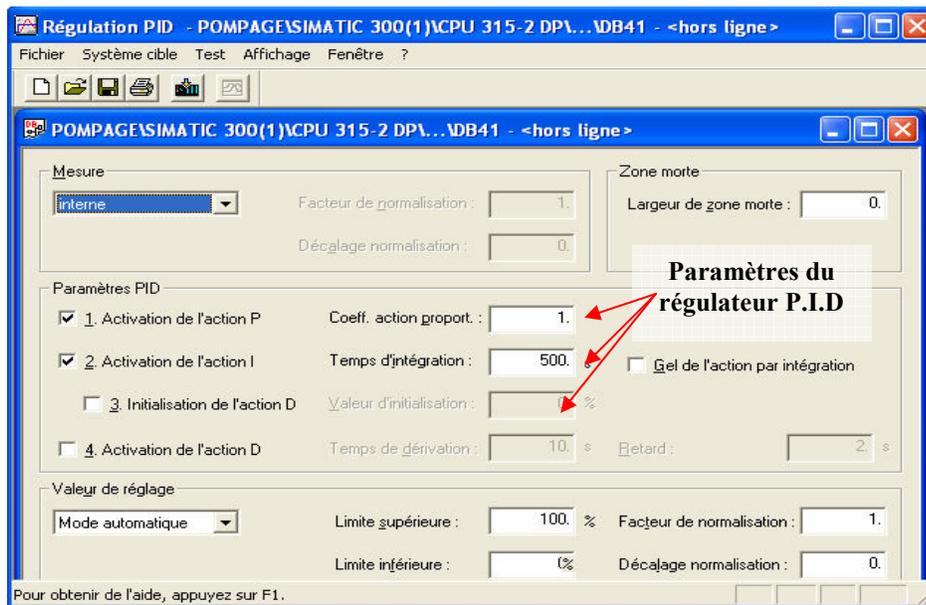


Figure IV.22 : Paramétrage du régulateur PID

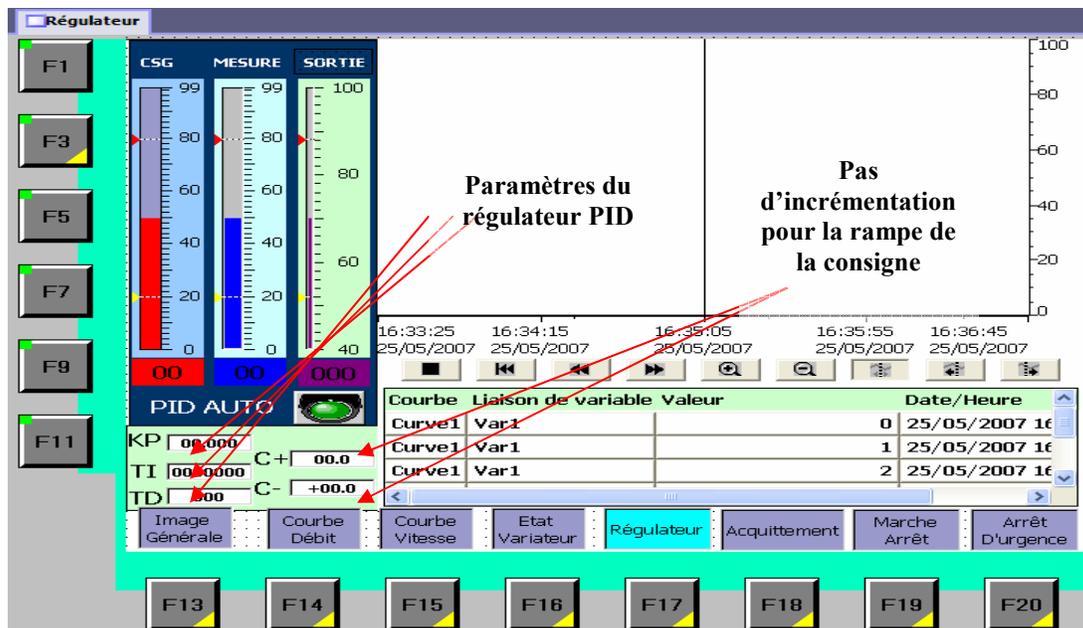


Figure IV.23 : Vue pour la régulation PID de débit dans le panneau MP270

Dans la majorité des systèmes industriels, les régulateur P.I sont les plus utilisés qui sont plus adéquats à la nature assez rapide du système, ainsi qu'à la présence des bruits de mesure.

Le choix des paramètres T_i et K_p se réfère aux méthodes expérimentales, qui sont les plus utilisés en industrie, comme la méthode de *Ziegler- Nichols* [14] ou de *Chien-Hrones-Reswick* [14] avec les différents types de réponses.

Les paramètres en boucle ouverte, K_s (gain), T_u (retard pur), T_g (retard compensatoire), nécessaire à la synthèse du régulateur P.I qui sont K_p et T_i , seront déterminés avec l'étude de la réponse indicielle du système en boucle ouverte, dans la zone linéaire de fonctionnement, ensuite calculer à partir de ces derniers pour chaque approche.

Comme dans notre cas, l'avantage principal de cette méthode est qu'elle s'applique aux systèmes pour lesquels on ne dispose d'aucune description théorique de type analytique.

IV.5.1 Détermination de la plage linéaire de fonctionnement

Une courbe caractéristique est une courbes donnant, en régime permanent les valeurs de la grandeur de sortie du système en fonction de celle de l'entrée, donc pour différentes valeur de l'entrée qui est la « commande manuelle » dans notre cas, on aura les valeurs de la grandeur de sortie « la mesure » qui suit :

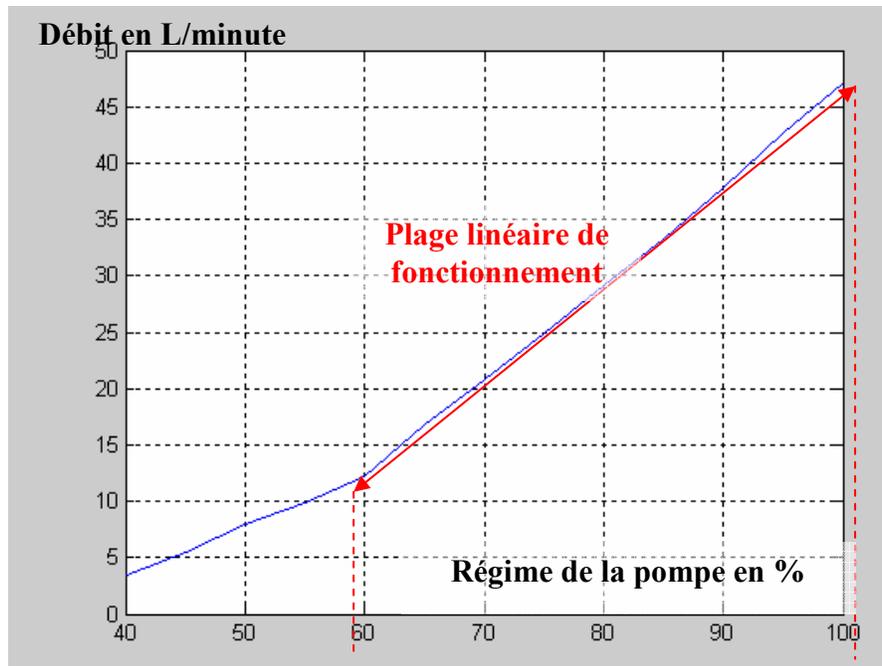


Figure IV.24 : La caractéristique statique du système en boucle ouverte

| Commande « régime de la pompe » en% | 40 | 45 | 50 | 55 | 60 | 65 | 70 | 75 | 80 | 85 | 90 | 95 | 100 |
|-------------------------------------|-----|-----|-----|-----|------|------|------|------|------|------|------|------|------|
| Mesure « débit » en Litre/minute | 3.4 | 5.5 | 7.9 | 9.8 | 12.2 | 16.8 | 20.8 | 24.9 | 29.2 | 32.2 | 37.8 | 42.6 | 47.2 |

Tableau IV.1: valeurs des grandeurs de sortie (débit) en fonction de la commande (vitesse)

Pour ces différentes valeurs, on constatera que la zone linéaire du système se situe entre 60% et 100% ce qui correspond à une vitesse minimale de 30Hz pour notre pompe.

IV.5.2 Détermination des paramètres du régulateur PI

Nous travaillerons dans la zone linéaire qui est entre 60% et 100% de la sortie, en donnant un échelon de à l'entrée du système en boucle ouverte :

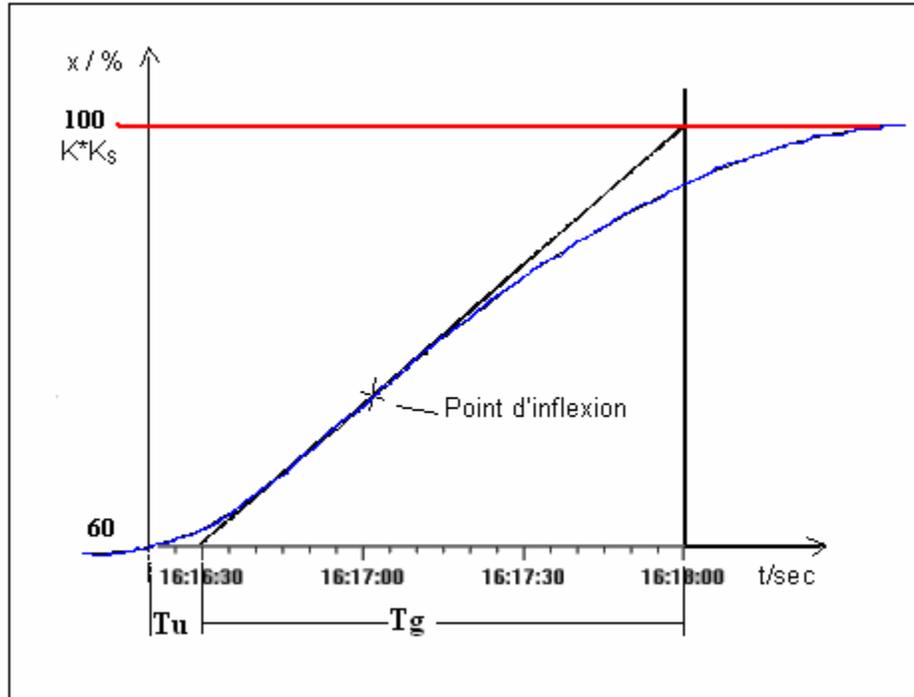


Figure IV.25 : Approximation des valeurs T_u , T_g , K_s

D'après le graphe, on déduit : **$T_g=90\text{sec}$, $T_u=2\text{ sec}$, $K_s=1555$**

Pour les différentes valeurs du gain proportionnel du régulateur **K_p** , ainsi que le temps d'intégration **T_i** , relatives à l'approche de régulation utilisée, on prélève les courbes suivantes :

- Courbes de consigne de débit donnée par l'opérateur.
- Courbes de mesure de débit.
- Courbes du régime de la pompe.

Au démarrage, on donne une consigne minimale de 20 L/mn qui correspond à 60% régime nominale, pour aborder la plage linéaire de fonctionnement, ensuite on donne une consigne assez élevée qui est de **45 L/mn** .

Les différentes courbes de ces réponses indicielles seront prélevées, ainsi que la réaction du système à ces perturbations.

a. Expérimentation

a.1.L'approche de Ziegler Nichols

$$Kp = \frac{0.9T_g}{T_u K_s} = 0.025 \quad (IV.1)$$

$$T_i = 3.33T_u = 6666ms \quad (IV.2)$$

On a les réponses suivantes :

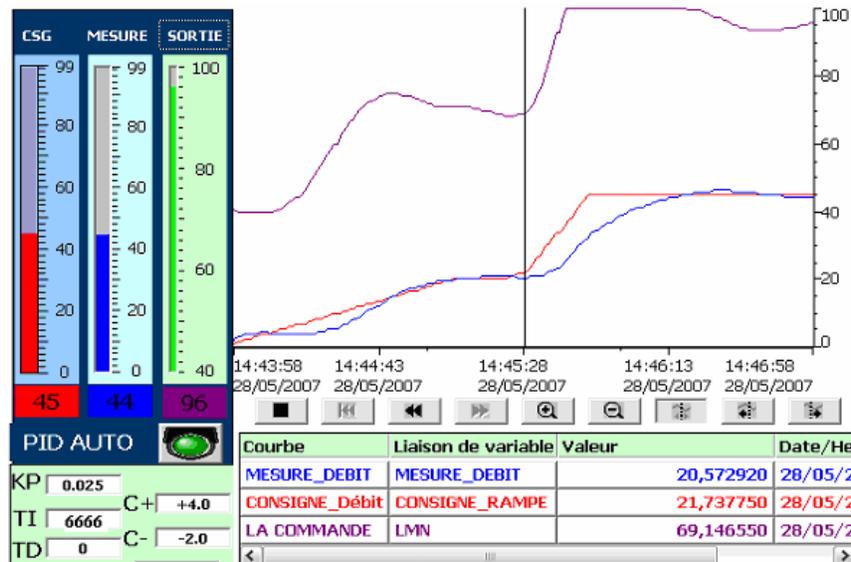


Figure IV.26 : Les courbes Consigne- mesure-commande par l'approche Ziegler-Nichols

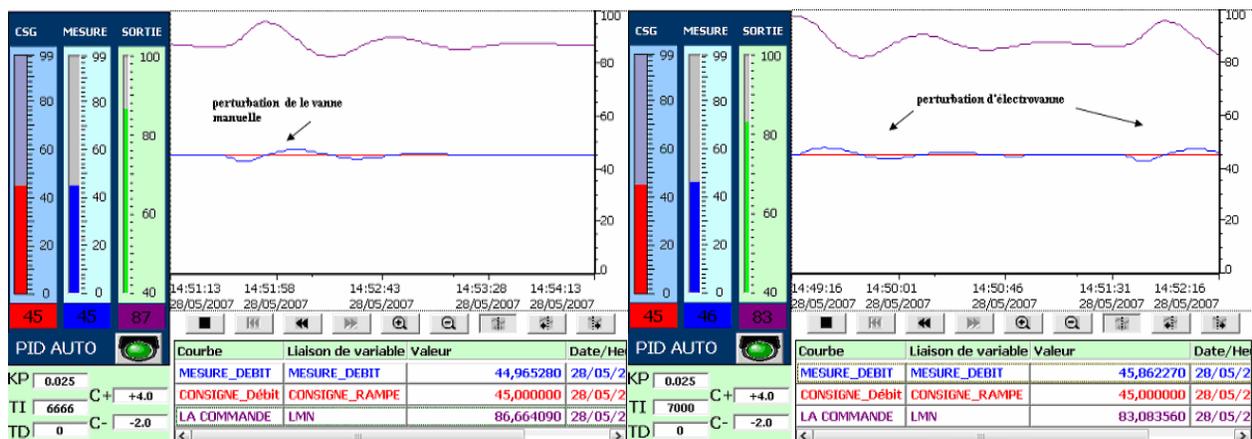


Figure IV.27 : Rejet des perturbations par l'approche Ziegler-Nichols

Commentaire

On remarque qu'au démarrage et avec une très forte pente pour la rampe de la consigne, un seul dépassement de la réponse, cependant le système réagit assez rapidement aux perturbations.

a.2. Les approches de Chien-Hrones-Reswick :

- Approche I : Réponse à une perturbation avec comportement apériodique à courte durée.

$$Kp = \frac{0.6T_g}{T_u K_s} = 0.017. \tag{IV.3}$$

$$T_i = 4T_u = 8000ms. \tag{IV.4}$$

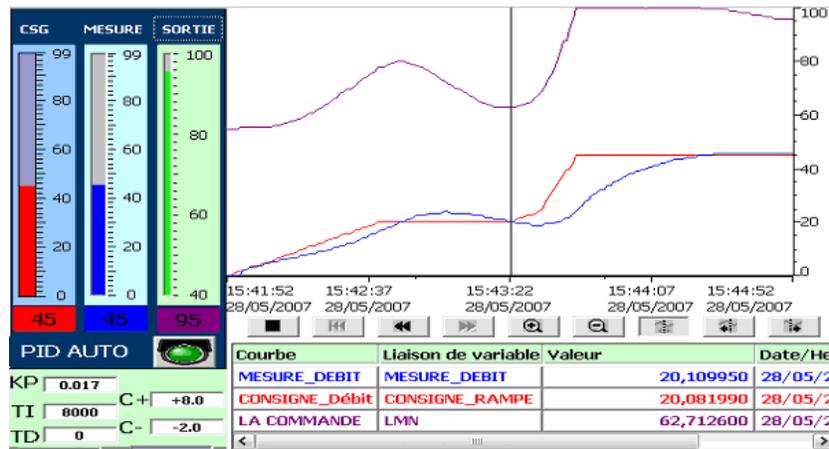


Figure IV.28 : Les courbes Consigne- mesure-commande par l'approche I de Chien

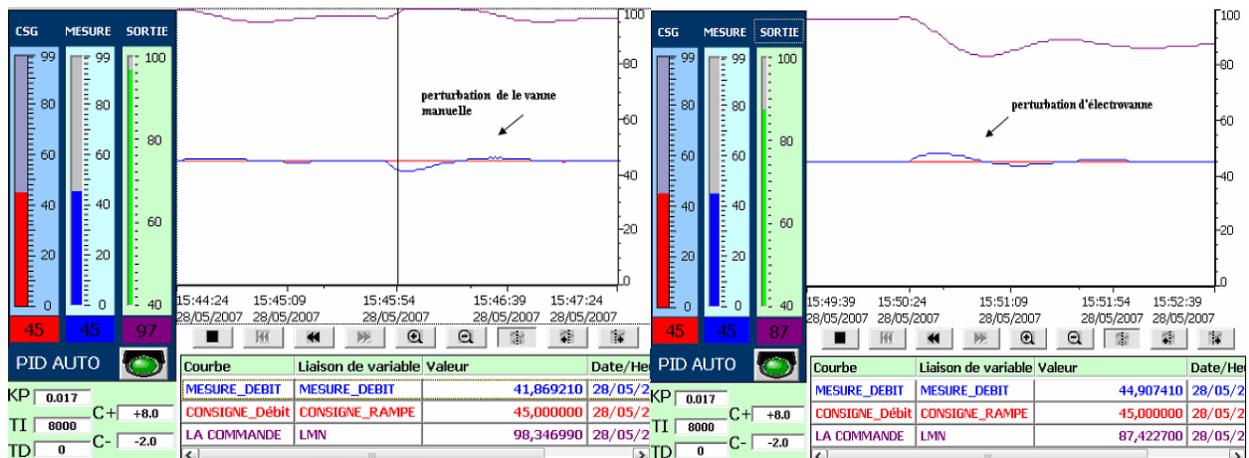


Figure IV.29: Rejet de perturbations par l'approche I de Chien

Commentaire

On remarque qu'au démarrage et avec une très forte pente pour la rampe de la consigne, que la réponse est pratiquement amortie, cependant le système réagit assez rapidement aux perturbations.

- **Approche II : Réponse à une perturbation avec Comportement à 20% d'épassement.**

$$Kp = \frac{0.7T_g}{T_u K_s} = 0.020 \quad (IV.5)$$

$$T_i = 2.3T_u = 4600 \quad (IV.6)$$

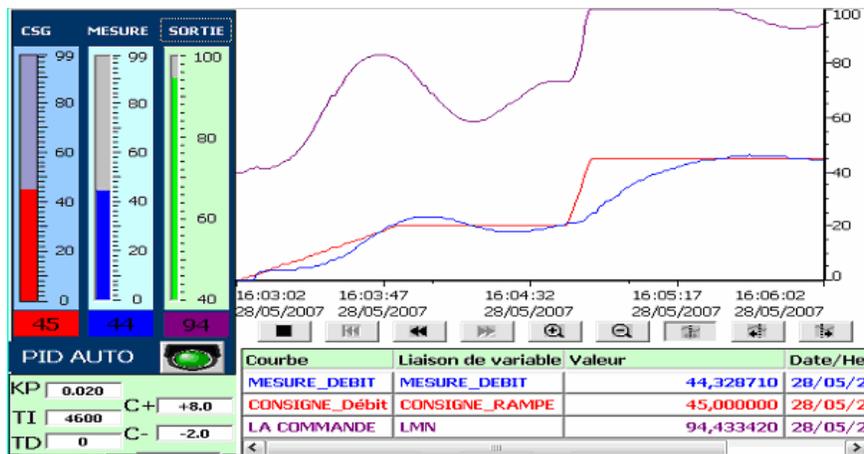


Figure IV.30 : Les courbes Consigne- mesure-commande par l'approche II de Chien

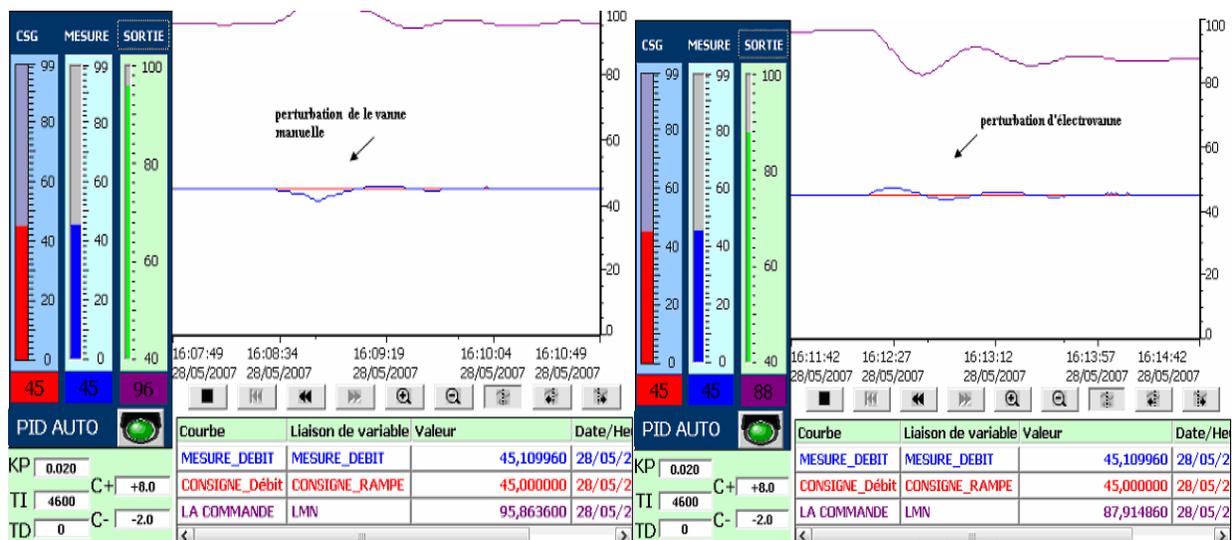


Figure IV.31: Rejet de perturbations par l'approche II de Chien

Commentaire

On remarque qu'au démarrage et avec une très forte pente pour la rampe de la consigne, que la réponse présente un seul dépassement, cependant le système réagit très rapidement aux perturbations.

- **Approche III : Suivi de consigne avec Comportement apériodique de courte durée**

$$Kp = \frac{0.35T_g}{T_u K_s} = 0.010. \tag{IV.7}$$

$$T_i = 1.2T_g = 108000ms. \tag{IV.8}$$

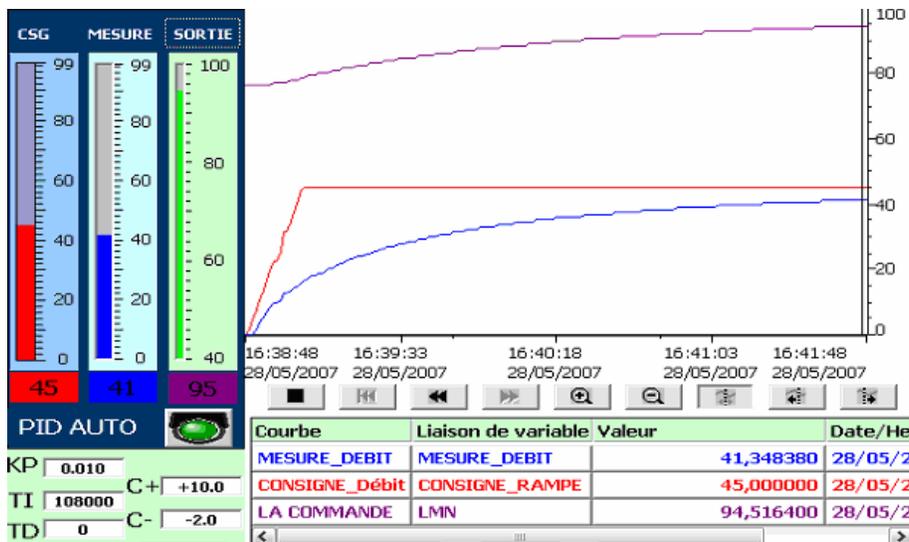


Figure IV.32 : Les courbes Consigne- mesure-commande par l'approche III de Chien

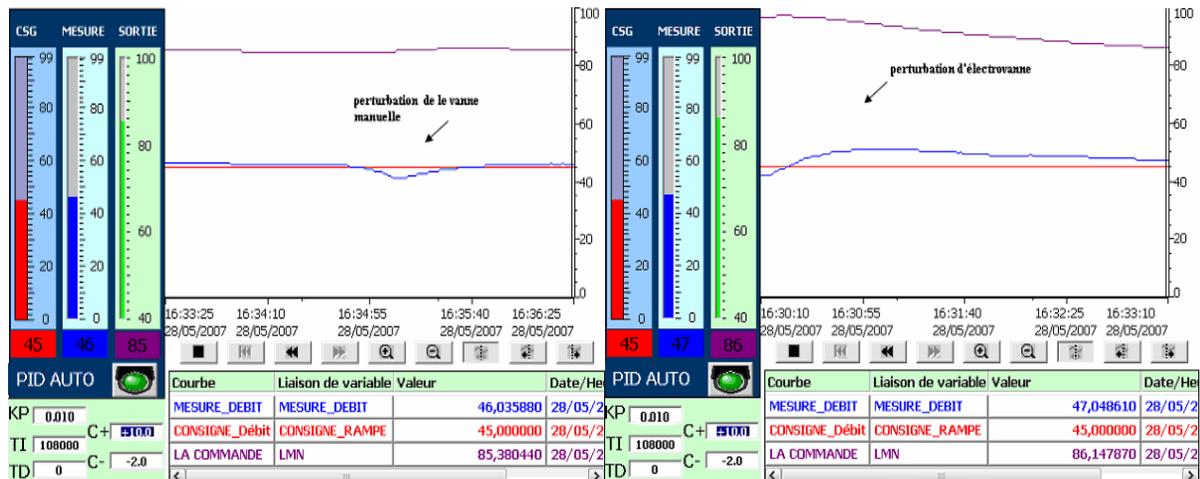


Figure IV.33: Rejet de perturbations par l'approche III de Chien

Commentaire

On remarque qu'au démarrage et avec une très forte pente pour la rampe de la consigne, que la réponse est assez lente et très amortie, cependant le système réagit lentement aux perturbations.

- **Approche IV : Suivi de consigne avec Comportement à 20% de dépassement**

$$Kp = \frac{0.6T_g}{T_u K_s} = 0.017 \quad (IV.9)$$

$$T_i = T_g = 90000ms \quad (IV.10)$$

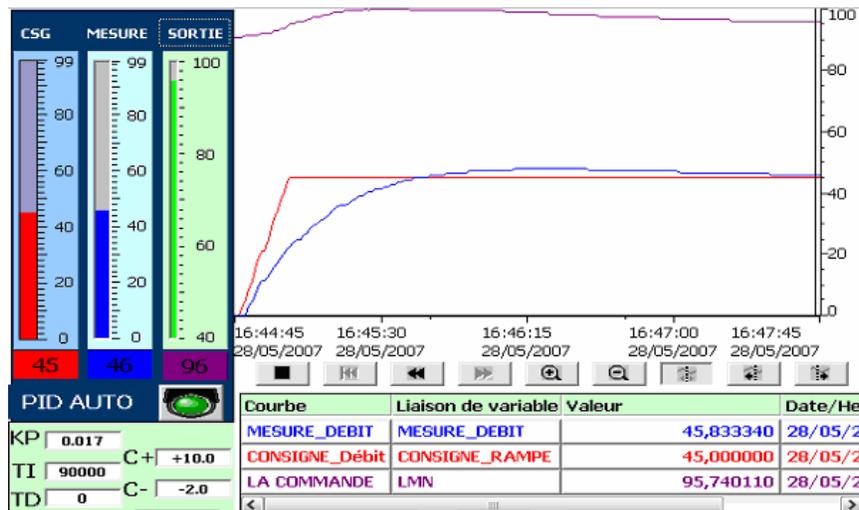


Figure IV.34 : Les courbes Consigne- mesure-commande par l'approche IV de Chien

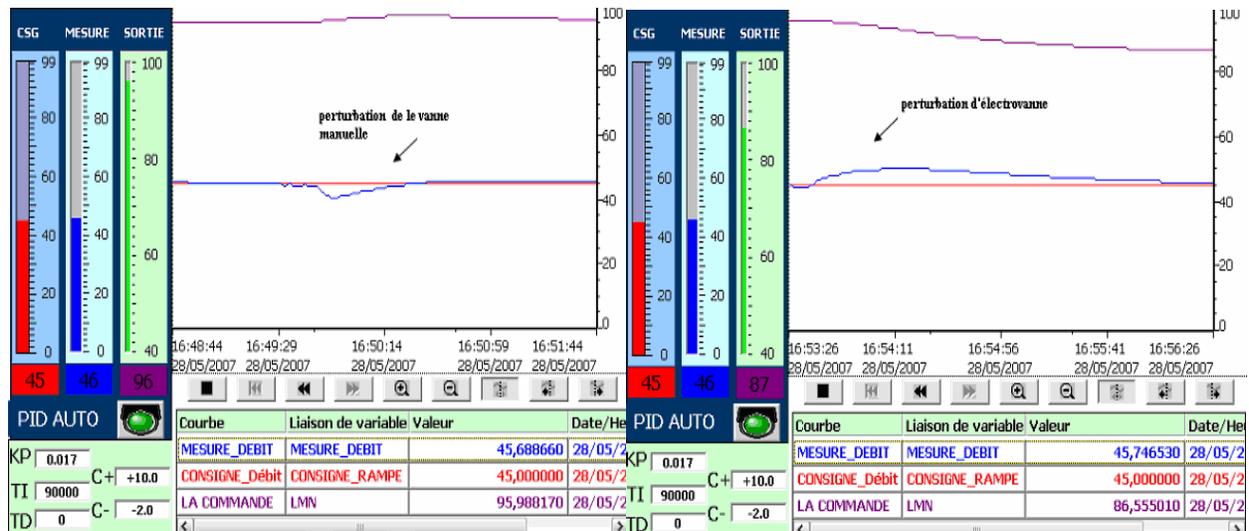


Figure IV.35: Rejet de perturbations par l'approche IV de Chien

Commentaire

On remarque qu'au démarrage et avec une très forte pente de la rampe, il a un seul dépassement de la réponse, cependant le système réagit très lentement aux perturbations.

b. Récapitulatif des essais

| L'approche | Temps de réponse (à 95% en seconde) | Type de la réponse | Le rejet des perturbations |
|--|--|--------------------------------------|-----------------------------------|
| <i>Ziegler-Nichols</i> | 45 | Oscillatoire (une seule oscillation) | ++ |
| Approche I de <i>Chien-Hrones-Reswick</i> | 47 | Amortie | +++ |
| Approche II de <i>Chien-Hrones-Reswick</i> | 34 | Oscillatoire (Un seul dépassement) | ++++ |
| Approche III de <i>Chien-Hrones-Reswick</i> | 192 | Fortement amortie | + |
| Approche IV de <i>Chien-Hrones-Reswick</i> | 43 | Oscillatoire (une seule oscillation) | + |

Tableau IV.2: Récapitulatif des résultats obtenus durant l'expérimentation

IV.6 Analyse des résultats et conclusion

Les méthodes graphiques utilisées pour la régulation sont basées sur le principe de l'optimisation approchée du système asservi, elles concernent essentiellement la précision dynamique.

On a bien remarqué, que pour avoir une bonne poursuite de consigne, les paramètres calculés par *Ziegler-Nichols* répondent plus à ces conditions; et pour répondre à une perturbation et éliminer l'approche III de *Chiens-Hornes-Reswick* est plus adéquate et plus performante.

Pour profiter, simultanément, des performances en poursuite de la première, et des rejets de perturbations de la deuxième, on peut prévoir en perspective, l'ajout d'un commutateur programmé entre les deux approches. La commutation et le choix se fera automatiquement suivant l'écart instantané entre la consigne et la mesure :

- Ecart grand \rightarrow on commute vers l'approche de *Ziegler-Nichols* pour réduire l'écart tout en optimisant la commande ;
- Ecart qui tend vers 0 \rightarrow on commute vers l'approche III de *Chiens-Hornes-Reswick*, et ceci pour éliminer rapidement d'éventuelles perturbations autour du point de fonctionnement.

Conclusion générale

La fonction d'un ingénieur doit être l'optimisation de technique permettant d'aboutir à l'objectif avec simplicité et performance.

Notre contribution s'est portée sur la réalisation du prototype de la station de pompage, l'élaboration du cahier de charge, du programme principale et de supervision, en optimisant la simplicité de compréhension et la précision du résultat.

Les résultats et tests obtenus répondent bien à l'objectif de l'asservissement, la simplicité et performance, le régulateur ainsi choisis réagit bien aux perturbations et approche la consigne désirée dans un temps convenable pour de telles installations.

L'automatisation, la régulation, la télétransmission, la mesure, le traitement de l'information et l'analyse de paramètres ont été traités durant ce projet.

La gestion et l'aboutissement d'un microprojet, comme le nôtre, dépend de plusieurs facteurs techniques et humain tel que :

- La bonne planification des taches en fonction des délais et la durée du microprojet.
- Le bon choix et la compatibilité des éléments utilisés, et aussi la préparation de l'environnement de travail.
- Une bonne entente et un travail de groupe.

La réalisation nous a poussé à faire appel à toute nos connaissances et aptitudes d'élèves ingénieurs et nous a permis d'appréhender les difficultés que les ingénieurs rencontre tous les jours dans le travail tout en prenant des initiatives personnelles.

Tout cela forge un ingénieur, et fait naître une confiance professionnelle, qualité essentielle, pour transiter vers une nouvelle phase ou il sera seul pour faire face aux exigences du terrain.

Perspectives

Ce mémoire étudie les différentes étapes de l'élaboration d'un projet software et l'établissement d'une commande à base d'automate programmable **SIEMENS** grâce au logiciel de programmation **STEP7** et le logiciel pour la supervision **Win CC flexible**. A la lumière des résultats obtenus, de nombreuses perspectives s'ouvrent à nous :

- Mettre en œuvre des commandes avancées au niveau du variateur de vitesse, tel que la commande vectorielle, et au niveau de l'automate pour la régulation du débit (commandes robustes, commande adaptative).
- Traiter en plus approfondi « **la télégestion** » dans le milieu industriel.
- Mettre en œuvre les connaissances acquises sur des installations réelles.

BIBLIOGRAPHIE

- [1] G. MICHEL, « Les A.P.I Architecture et application des automates programmables industriels », Edition DUNOD, 1987
- [2] P.JARGOT, « Langages de programmation pour API. Norme IEC 1131-3 », Techniques de l'ingénieur, Vol. S 8 030.
- [3] M. BERTRAND, « Automates programmables industriels », Techniques de l'ingénieur, Vol. S 8 015.
- [4] J.P.THOMESSE, « Réseaux locaux industriels », Techniques de l'ingénieur, Vol. R 7 574.
- [5] C.KOLSKI, « Ingénierie des systèmes Homme/Machine », Techniques de l'ingénieur, Vol. R 7 614 D.
- [6] G. PUJOLLES, « Les réseaux », Edition Eyrolles, 1997.
- [7] E.BAJIC et B.BOUARD, « Réseau PROFIBUS », Techniques de l'ingénieur, Vol. S 8 190.
- [8] N.BOUAYED, «La télégestion», projet de fin d'étude, Ecole Nationale Polytechnique, 2006.
- [9] R.BOUROUBI, «Mise en œuvre du logiciel STEP 7, application à l'automate programmable S7-314 IFM », projet de fin d'étude, Ecole Nationale Polytechnique 2004.

Manuels:

- [10] SIEMENS, «*PROFIBUS Specification. Normative parts of Profibus-FMS, DP, PA according to the european Standard EN 50170*», Vol. 2, edition 1.0, PNO, 1998.
- [11] SIEMENS, «*PROFIdrive, Profile Drive Technology* », Version 3.0, 2000.
- [12] SIEMENS, «*Drive Monitor* », SIMOVERT, 2002.
- [13] SIEMENS, «*STEP 7, Getting started*», SIMATIC, 2002.
- [14] SIEMENS, «*STEP 7, Régulation PID*», SIMATIC, 2000.
- [15] SIEMENS, «*S7PLCSIM, Testez vos programmes*», SIMATIC, 2002.
- [16] SIEMENS, «*Programmation avec STEP 7* », SIMATIC, 2000.

- [18] SIEMENS, «Appareils de terrain pour l'automatisation des processus», SITRANS, 2005.
- [19] SIEMENS, «*SIMOVERT MASTERDRIVES VC*», SIMOVERT, 2005.
- [20] SIEMENS, «*SIMOVERT MICROMASTER*», SIMOVERT, 2005.
- [21] SATEL, «Modem radio 3AS S.N 03091626», SATELINE, 2004.
- [22] KSB, «MULTI 1000D S.N. 48834969», 2002.
- [23] FASCO, « Electrovanne S.N. SCG238C016 », 2005

Internet:

- [24] www.siemens.com
- [25] www.wikipedia.com

ANNEXE A

Les programmes de l'application « Mise en œuvre d'une application PROFIBUS-DP avec Maître CPU 315-2DP et Esclave MICROMASTER 440 : entraînement d'un chariot sur un plan incliné»

A.1 Le bloc d'organisation OB1

A.1.1. La fonction de gestion des tâches FC10

A.1.2 La fonction de Diagnostique FC11

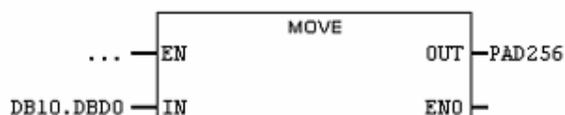
Le programme qui suit concerne la deuxième application : « **Mise en œuvre d'une application PROFIBUS-DP avec Maître CPU 315-2DP et Esclave MICROMASTER 440 : entraînement d'un chariot sur un plan incliné** », on peut voir dans l'ordre, le bloc d'organisation principale OB1, et les fonctions qu'il appelle.

A.1 Le bloc d'organisation OB1

Il se présente sous forme de cinq réseaux suivants :

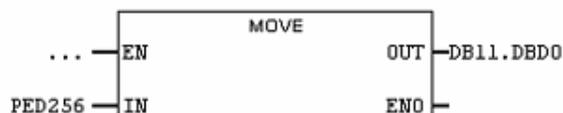
Réseau 1 : ENVOI DU MOT DE COMMANDE

Envoyer le mot de commande, contenu dans le bloc de donnée DB10 vers le variateur via le double mot de sortie PAD256



Réseau 2 : LECTURE DU MOT D'ETAT

Chargement du mot d'état contenu dans le double mot d'entrée PAD256, et l'acquiesionner dans le bloc de donnée DB11



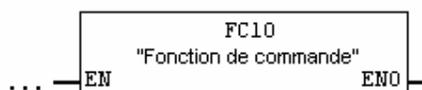
Réseau 3 : AQUITTEMENT DES ERREURS

Aquitter après défauts de communication Maître_Esclave, par un front montant sur l'entrée E1.0



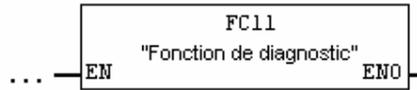
Réseau 4 : FONCTION DE COMMANDE

Appel de la fonction FC10 par l'OB1



Réseau 5 : FONCTION DE

Appel de la fonction FC11 par l'OB1



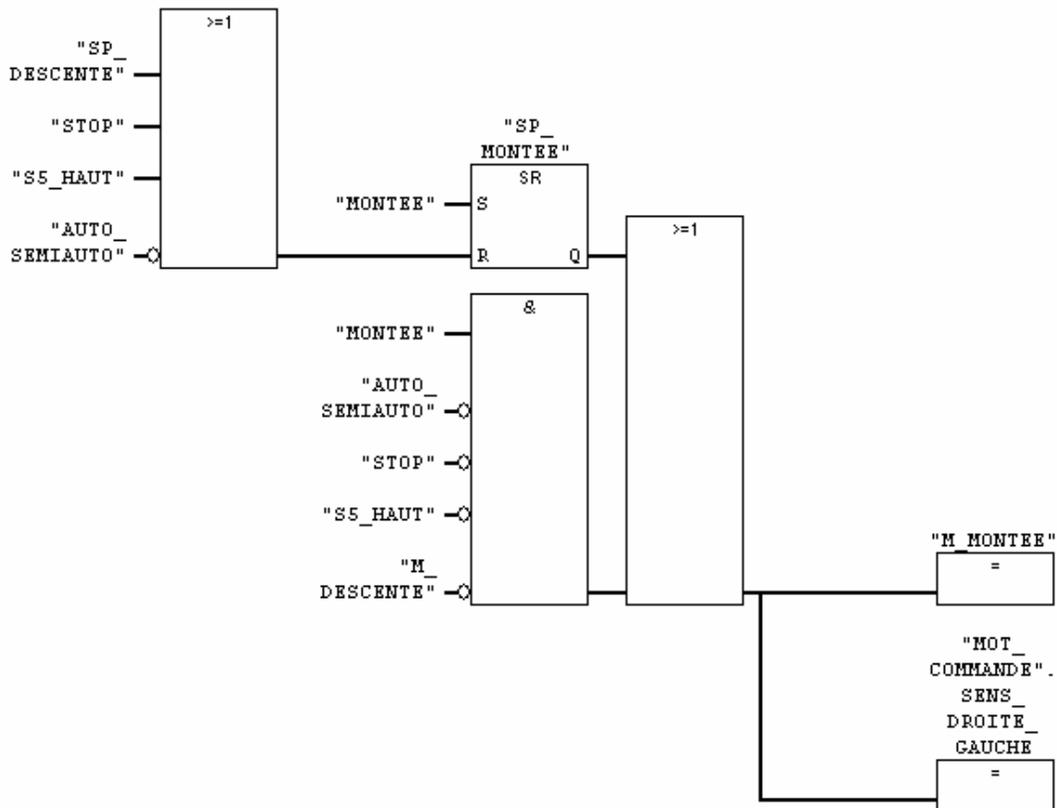
On détaille dans ce qui suit les deux fonctions *FC10* et *FC11* appelés par *l'OB1* :

A.1.1. La fonction de gestion des taches FC10

Elle se compose de sept réseaux détaillés comme suit :

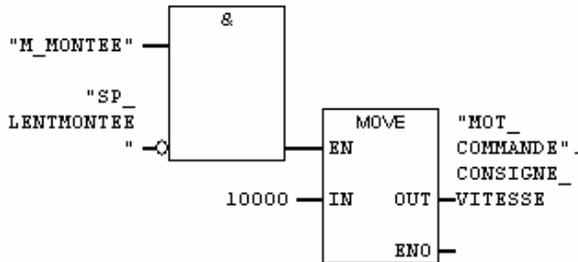
Réseau 1 : MONTEE DU CHARIOT

Combinaison logique pour l'ordre de Marche/Arrêt, et ceci pour la montée du chariot, et choix du sens de rotation vers la droite



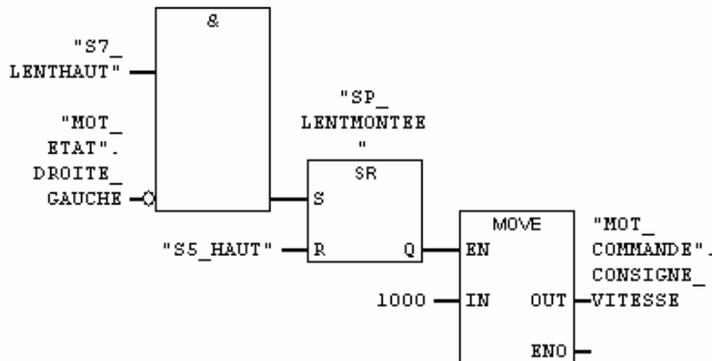
Réseau 2 : VITESSE REQUISE

Affectation de la valeur 10000 comme consigne de vitesse, qui représente un taux de 10000/16834 de la vitesse nominale 1350 tours/minute



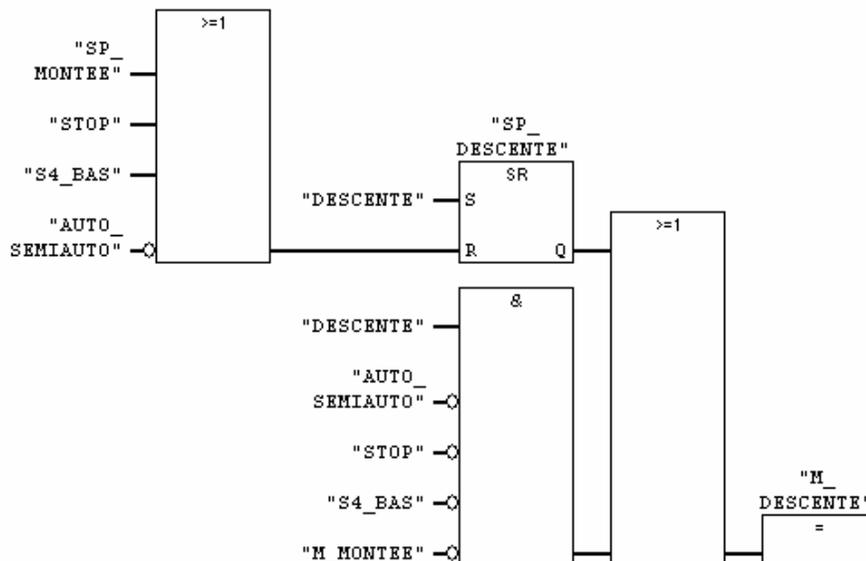
Réseau 3 : PHASE DE FREINAGE

Affectation de la consigne 1000 comme consigne de vitesse, pour la phase de décélération



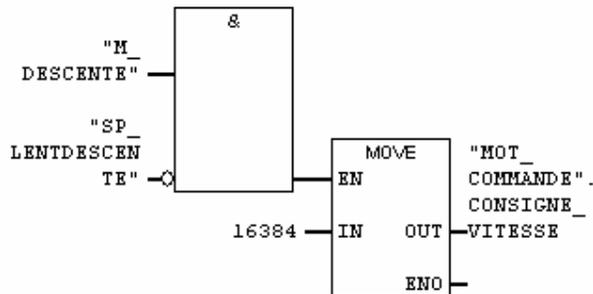
Réseau 4 : DESCENTE DU CHARIOT

Combinaison logique pour l'ordre de Marche/Arrêt, et ceci pour la descente du chariot, et choix du sens de rotation vers la gauche.



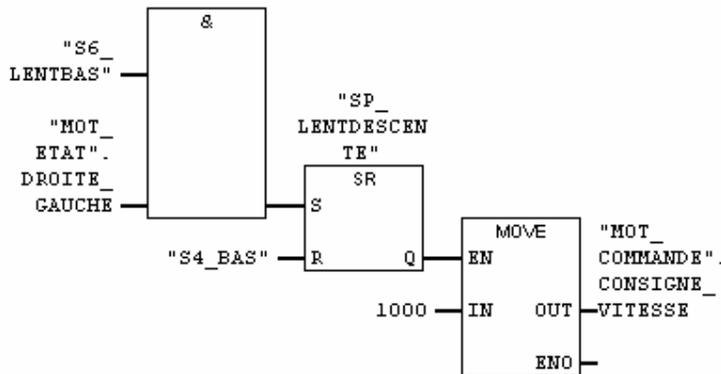
Réseau 5 : VITESSE REQUISE

Affectation de la valeur 16384 comme consigne de vitesse, qui représente la vitesse nominale 1350 tours/minute



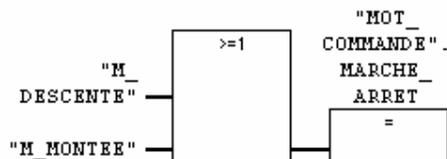
Réseau 6 : PHASE DE FREINAGE INFERIEURE

Affectation de la consigne 1000 comme consigne de vitesse, pour la phase de décélération



Réseau 7 : DEMARRAGE ARRET DU MONTE CHARIOT

Actionner Marche en présence de l'une des requête, montée ou descente, indépendamment du sens de rotation

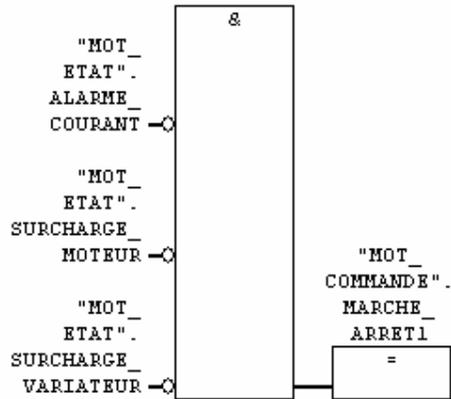


A.1.2 La fonction de Diagnostic FC11

Formé d'un seul réseau détaillé comme suit :

Réseau 1: DETECTION DEFAULTS VARIATEUR

Si un des défaut : surchage variateur, surcharge moteur ou une alarme de courant apparait, le variateur commute vers l'arret2 et le ralentissement naturel du moteur



ANNEXE B

Descriptions des éléments d'entraînement et de communication utilisés au cours de l'application « Mise en œuvre d'une application PROFIBUS-DP avec Maître CPU 315-2DP et Esclave MICROMASTER 440 »

- B.1. La CPU 315-2DP**
- B.2. Le variateur de vitesse MICROMASTER 440**
- B.3. La carte CB pour communication PROFIBUS-DP du MICROMASTER 440**
- B.4. Le moteur asynchrone Siemens**

B.1. La CPU 315-2DP

Elle constitue l'élément principal de la S300 utilisé, et présente les caractéristiques suivantes :

| | | |
|--------------------------------------|---|-----------------------------|
| Version CPU | N° série | 6ES7-315-2AF03-0AB0 |
| | Version du matériel | 0.1 |
| Mémoires | Mémoire de travail | 64Ko non extensible |
| | Mémoire de chargement | RAM de 96 Ko non extensible |
| Temps de traitement (minimal) | Opération sur bit | 0.3 μ s |
| | Opération sur mot | 1 μ s |
| | Opération arithmétique sur nombres entiers | 2 μ s |
| | Opération arithmétique sur nombres à virgule flottante | 50 μ s |

Tableau B.1 : *Caractéristiques de la CPU 315-2DP*

B.2. Le variateur de vitesse MICROMASTER 440**B.2.1. Description**

Les variateurs de la gamme *SIMOVERT MICROMASTER 440* sont des convertisseurs de fréquence permettant d'ajuster la vitesse et le couple de moteurs triphasés. Les modèles disponibles couvrent la plage de puissance 120 W jusqu'à 200 kW (avec couple constant (CT)) et/ou jusqu'à 250 kW (avec couple variable (VT)).



Figure B.1 : *La gamme MicroMaster 440*

La variante XAU703-002938, qui était utilisé lors de la deuxième application « **Mise en œuvre d'une application PROFIBUS DP avec Maître CPU 315-2DP et Esclave MICROMASTER 440** » se caractérise par les données électriques suivantes :



| | |
|-------------------------------------|---------------------|
| N° série Siemens | 6SE6440-2AB11-2AA1 |
| Entrée en tension nominale | 200-240 V +10% -10% |
| Entrée en courant nominale | 1.8 A |
| Fréquence du réseau d'alimentation | 47-63 Hz |
| Puissance Max du moteur à entraîner | 0.12 kW |

Tableau B.2 : *Caractéristique de la variante XAU703-002938*

Des fonctions de protection étendues garantissent une excellente protection du moteur et du variateur.

Le variateur offre une palette de choix pour les algorithmes de commande :

- Contrôle vectoriel avec ces deux types :
 - Contrôle vectoriel sans capteur (*SLVC*)
 - Contrôle vectoriel avec capteur (*VC*)
- Régulation *U/f*.

B.2.2 Paramétrage du variateur de vitesse Micromaster440

Pour adapter le variateur à l'application, il suffit d'adapter ces blocs fonctionnels par le biais des paramètres suivant :

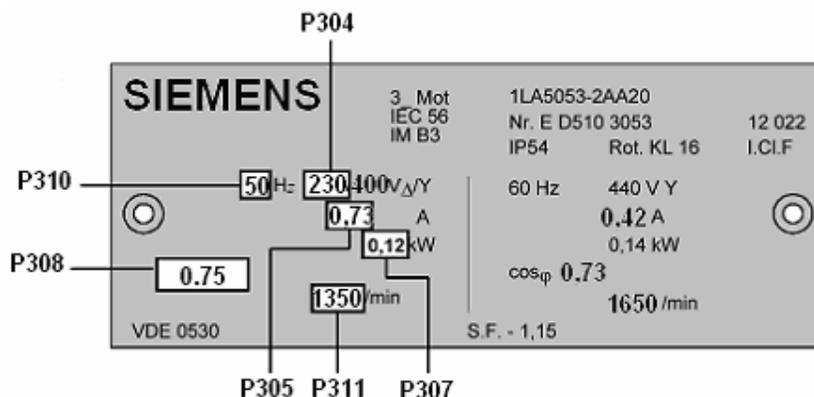
- *Paramètres du variateur :*

| <i>N° du paramètre</i> | <i>Signification et valeurs possibles</i> | <i>Choix pour l'application</i> |
|------------------------|---|---------------------------------|
| P0003 | Niveau d'accès utilisateur 1 Standard (application simple) 2 Etendu (application standard) 3 Expert (application complexe) | 2 |
| P0004 | Filtre des paramètres pour faciliter l'accès 0 Tous les paramètres 2 Variateur 3 Moteur 4 Capteur de vitesse | 0 |
| P0010 | Filtre des paramètres de mise en route 0 Prêt 1 Mise en service rapide 30 Réglages d'usine | 1 |
| P0100 | Europe / Amérique du Nord (fréquence du secteur) 0 Europe [kW], 50 Hz 1 Amérique du Nord [hp], 60 Hz 2 Amérique du Nord [kW], 60 Hz | 0 |
| P0205 | Application de variateur (type de couple requis) 0 couple constante (par ex. compresseurs, machines de fabrication) 1 couple variable (par ex. pompes, ventilateurs) | 0 |

Tableau B.3 : *Paramètres du variateur*

- **Paramètre moteur**

Ces valeurs sont récupérés de la plaque signalétique du moteur comme suit :



Et saisie dans le menu :

| N° du paramètre | Signification et valeurs possibles | Choix pour l'application |
|-----------------|---|--------------------------|
| P0300 | Sélection type de moteur 1 Moteur asynchrone 2 Moteur synchrone | 1 |
| P0304 | Tension assignée du moteur (à relever sur la plaque signalétique en <i>V</i>) Tension nominale du moteur indiquée sur la plaque signalétique. Le diagramme suivant montre une plaque signalétique typique avec la position des caractéristiques essentielles du moteur. | 230 |
| P0305 | Courant assigné moteur (à relever sur la plaque signalétique en <i>A</i>) | 0.73 |
| P0307 | Puissance assignée moteur (à relever sur la plaque signalétique en <i>kW</i>) | 0.12 |
| P0308 | cosPhi assigné du moteur (à relever sur la plaque signalétique en cosPhi) Le réglage 0 entraîne un calcul interne de la valeur. P0100 = 1,2: P0308 sans objet, aucune saisie nécessaire. | 0.75 |
| P0310 | Fréquence moteur assignée (à relever sur la plaque signalétique en <i>Hz</i>) | 50 |

| | | |
|--------------|--|-------------|
| | Le nombre de paires de pôles est recalculé automatiquement dès que le paramètre est modifié. | |
| P0311 | Vitesse moteur nominale (à relever sur la plaque signalétique en tr/min) Avec le réglage 0, la valeur est calculée en interne. | 1350 |
| P0335 | Refroidissement du moteur (Sélectionne le système de refroidissement moteur.) 0 Autoventilé par ventilateur sur l'arbre du moteur 1 Motoventilé séparé par un motoventilateur indépendant (motoventilateur) 2 Autoventilation et ventilateur interne 3 Refroidissement séparé et ventilateur interne | 0 |

Tableau B.4 : Paramètres du moteur

- **Paramètres pour la source de commande**

| | | |
|--------------|--|----------|
| P0700 | Sélection source de commande 0 Réglages usine par défaut 1 BOP (clavier) 2 Borne 4 USS sur liaison BOP 5 USS sur liaison COM (bornes de commande 29 et 30) 6 CB sur liaison COM (CB = carte de communication PROFIBUS-DP) | 6 |
|--------------|--|----------|

Tableau B.5 : Paramètres du variateur

- **Paramètres pour la source de consigne**

| | | |
|--------------|--|----------|
| P1500 | Sélection source de consigne couple (saisie de la source de la consigne de couple, seulement avec contrôle vectoriel p1300 = 22 ou 23) 0 Pas consigne principale 2 Consigne analogique 4 USS sur liaison BOP 5 USS sur liaison COM | 0 |
|--------------|--|----------|

| | | |
|--------------|--|----------|
| | 6 CB sur liaison COM 7 Consigne analogique 2 | |
| P1000 | Sélection consigne de fréquence (vitesse) 1 Consigne BOP 2 Consigne analogique 3 Fréquence fixe 4 USS sur liaison BOP 5 USS sur liaison COM 6 CB sur liaison COM 7 Consigne analogique 2 Une combinaison entre deux source est possible, exemple : 12 Consigne analogique + Consigne MOP | 6 |

Tableau B.6 : Paramètres du variateur

- Paramètres pour les caractéristiques de l'entraînement**

| | | |
|--------------|--|-----------|
| P1080 | Fréquence min. (fréquence minimale du moteur en Hz) Définition de la fréquence minimale de fonctionnement du moteur indépendamment de la consigne de fréquence. La valeur entrée ici est valable pour les deux sens de rotation. | 0 |
| P1082 | Fréquence max. (fréquence maximale du moteur en Hz) Définition de la fréquence maximale sur laquelle le moteur sera limité indépendamment de la consigne de fréquence. La valeur entrée ici est valable pour les deux sens de rotation. | 50 |
| P1120 | Temps de montée (temps d'accélération en s) Temps nécessaire au moteur pour accélérer de 0 à la fréquence maximale du moteur (P1082) en l'absence de lissage. Un temps | 10 |

| | | |
|--------------|---|-----------|
| | de montée trop court peut provoquer l'alarme A0501 (valeur limite de courant) ou mettre le variateur hors tension avec le défaut F0001 (surintensité). | |
| P1121 | <p>Temps de descente (temps de décélération en s)</p> <p>Temps nécessaire au moteur pour décélérer de la fréquence maximale du moteur (P1082) à 0 en l'absence de lissage. Un temps de descente trop court peut provoquer les alarmes A0501 (valeur limite de courant), A0502 (valeur limite de surtension), ou bien mettre le variateur hors tension avec le défaut F0001 (surintensité) ou F0002 (surtension)</p> | 15 |
| P1135 | <p>Temps de descente ARRET3 (temps de descente pour arrêt rapide en s)</p> <p>Définit le temps de descente de la fréquence maximum jusqu'à 0 pour l'ordre ARRET3. Un temps de descente trop court peut provoquer les alarmes A0501 (valeur limite de courant), A0502 (valeur limite de surtension), ou bien mettre le variateur hors tension avec les défauts F0001 (surintensité) ou F0002 (surtension).</p> | 5 |
| P1300 | <p>Mode de commande (type de régulation souhaité)</p> <p>0 U/f avec caract. linéaire 1 U/f 2 U/f avec caractéristique parabolique 3 U/f avec caract. programmable 5 U/f pour les applications textiles 6 U/f pour les applications textiles 19 Commande U/f avec une consigne de tension indépendante 20 Contrôle vectoriel sans capteur de mesure 21 Contrôle vectoriel avec capteur de mesure 22 Contrôle vectoriel de couple sans capteur 23 Contrôle vectoriel de couple avec capteur</p> | 0 |

Tableau B.7 : Paramètres pour les caractéristiques de l'entraînement

- **Validation du jeu de paramètres dans le variateur**

C'est l'ultime étape du paramétrage. Elle consiste à vérifier la cohérence des données saisies du moteur avec les grandeurs réelles, après saisie des caractéristiques d'entraînement, et charger le jeu de paramètres dans le variateur.

Et le voici le tableau des paramètres ultimes pour l'optimisation et la validation :

| | | |
|--------------|---|----------|
| P1910 | Identifications données motrices Utiliser pour se rapprocher plus des données réels du moteur. 0 Bloqué 1 Activé | 1 |
| P3900 | Fin du mode mise en service (démarrage du calcul du moteur) 0 Pas de mise en service rapide (pas de calcul du moteur) 1 Calcul du moteur et réinitialisation de tous les autres paramètres non contenus dans la mise en service rapide (attribut "mise en service rapide" = non), sur réglage usine. 2 Calcul du moteur et réinitialisation des réglages d'E/S sur les réglages usine. 3 Uniquement calcul du moteur. Pas de réinitialisation des autres paramètres. | 1 |

Tableau B.8 : Paramètres pour la validation du jeu de paramètres dans le variateur

B.4. La carte CB pour communication PROFIBUS-DP du MICROMASTER 440

B.4.1 Description

Le module de communication PROFIBUS-DP (module optionnel PROFIBUS) sert à relier des variateurs de la gamme SIMOVERT MICROMASTER 440 aux systèmes d'automatisation de niveau supérieur par le biais du bus de terrain PROFIBUS-DP.



Figure B.3 : Carte CB pour la communication PROFIBUS-DP

B.4.2 Caractéristiques techniques

A fin de nous informer sur son état, le module de communication comporte une LED tricolore (vert, orange, rouge).

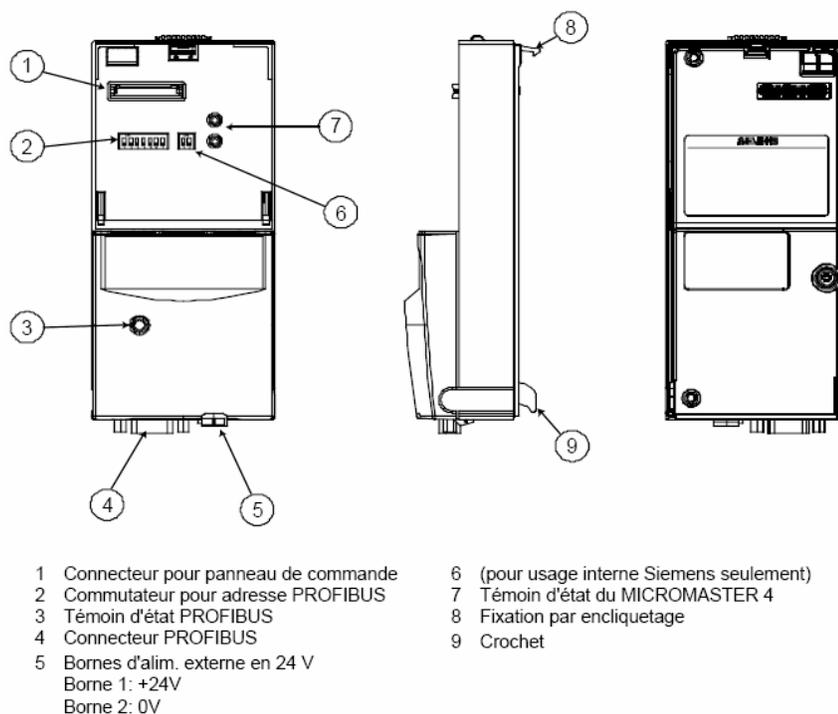


Figure B.3 : *Vue du module de communication*

La tension d'alimentation est assurée par le connecteur système du variateur.

Une source externe de 24 V fournit la tension d'alimentation du module optionnel **PROFIBUS** et de l'électronique du variateur.

Le raccordement au bus **PROFIBUS** s'effectue par un connecteur Sub-D 9 points à la norme **PROFIBUS**. Toutes les connexions de cette interface RS485 sont protégées contre les courts-circuits et sont à séparation galvanique.

B.5. Le moteur asynchrone Siemens

Le moteur asynchrone SIEMENS à cage se présente sous la forme d'un carter (2) entourant le circuit ferromagnétique, statorique avec ses encoches contenant l'enroulement statorique polyphasé (généralement triphasé) bobiné en fil de cuivre isolé (1). A l'intérieur de ce circuit magnétique, qui se présente comme un cylindre creux, séparé par un entre fer, tourne le circuit magnétique rotorique (3) qui accueille dans ses encoches les barreaux de la cage rotorique, en aluminium coulé ou en cuivre, court-circuité à chaque extrémité par des

anneaux réalisés dans le même matériau. Le circuit magnétique rotorique est traversé par l'arbre qui repose sur les paliers montés dans les flasques (4), (5) fixées au carter.

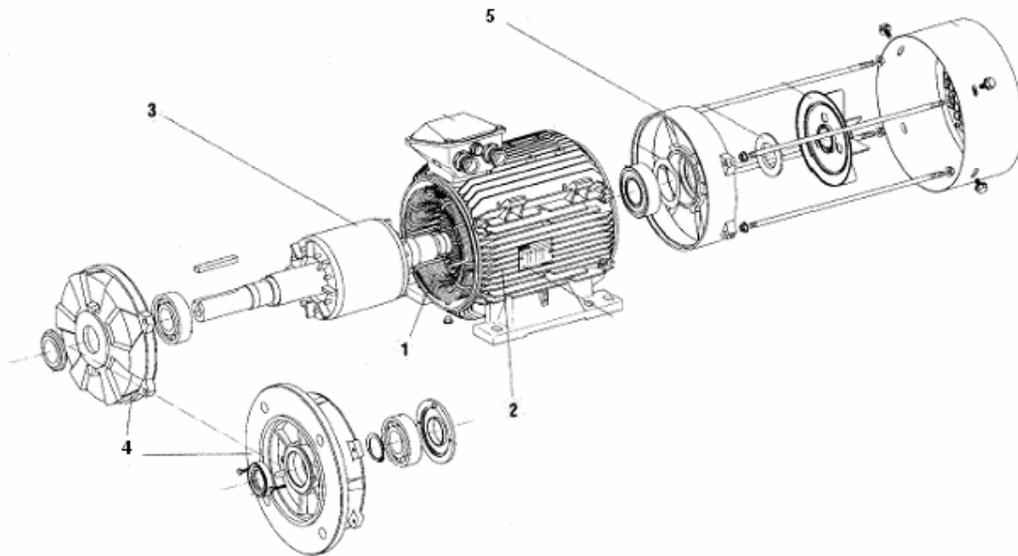


Figure B.4 : Machine asynchrone à cage d'écureuil SIEMENS

La machine asynchrone à cage est le moteur le plus répandu dans l'industrie : elle est robuste, fiable, économique.

Cette machine présente les caractéristiques électriques suivantes :

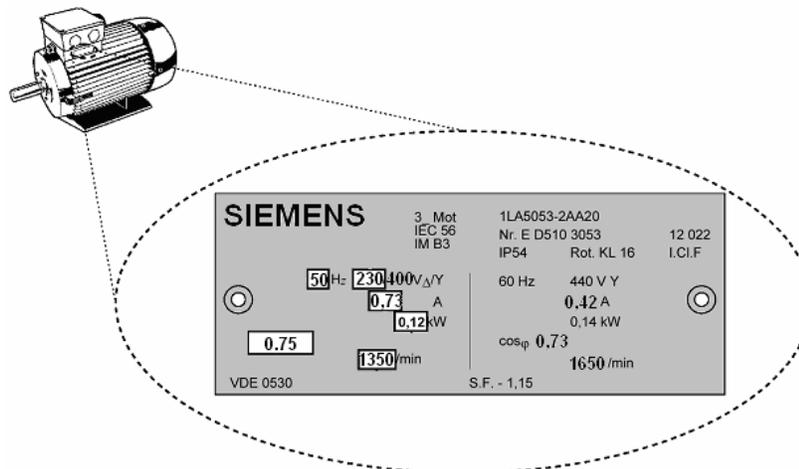


Figure B.5 : Plaque signalétique du moteur asynchrone SIEMENS

Ces données seront exploitées pour le paramétrage du variateur de vitesse **MICROMASTER 440**.

ANNEXE C

Les programmes de l'application « Réalisation et gestion d'un prototype de station de pompages à base d'automates SIEMENS »

C.1. Le bloc d'organisation OB1

C.1.1. La fonction de gestion de la station FC11

C.1.2. La fonction de gestion du MASTERDRIVE FC12

C.1.3. La fonction de gestion de la HMI FC21

C.2. Le bloc d'organisation cyclique OB35

Le programme qui suit concerne la troisième application : « **Réalisation et gestion d'un prototype de station de pompage à base d'API SIEMENS** », on peut voir dans l'ordre le bloc d'organisation principale OB1 et l'OB35, ainsi qu'aux fonctions qu'il appelle.

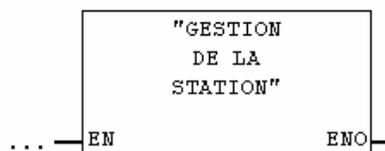
C.1. Le bloc d'organisation OB1

OB1 : "Main Program Sweep (Cycle)"

```
APPEL DES DIFFERENTES FONCTIONS POUR LA GESTION DES DIFFERENTS SOUS PROGRAMMES.
```

Réseau 1 : GESTION DE LA STATION

```
Gestion de la station au demrrage et pour l'arret, ainsi que le remplissage des reservoirs.
```



Réseau 2 : GESTION DE LA HMI

```
Gestion de la station , Interface homme Machine.
```

```
CALL "GESTION HMI"
```

Réseau 3 : GESTION DU MASTERDRIVE

```
Gestion du Master Drive.
```

```
CALL "GESTION_MASTER"
```

On détail dans ce qui suit les deux fonctions *FC11* « *Gestion de la station* », *FC12* « *Gestion du MASTERDRIVE* » et *FC21* « *Gestion de la HMI* » appelés par l'*OB1*.

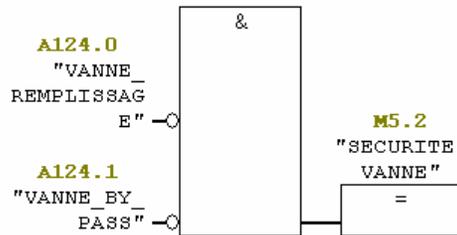
C.1.1. La fonction de gestion de la station FC11

FC11 : Gestion de la station

la gestion de la station du point de vue, démarrage, arrêt, et commande des vannes de remplissage.

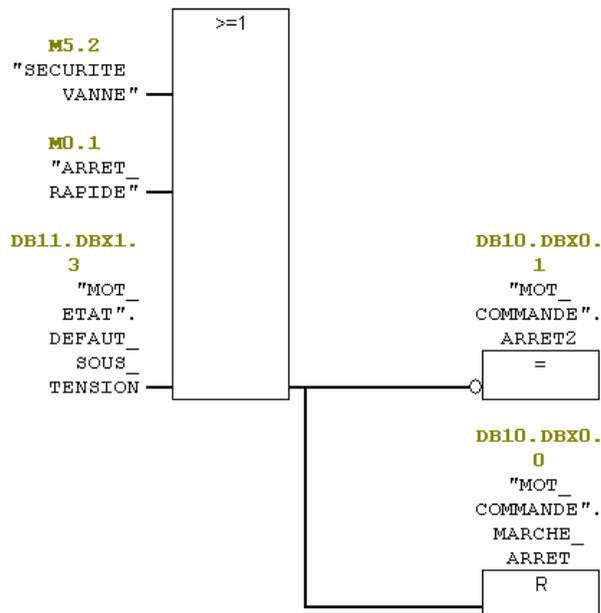
Réseau 1: MARCHÉ_ARRET MOTEUR

Commande en marche du variateur, ou la mise à l'arrêt.



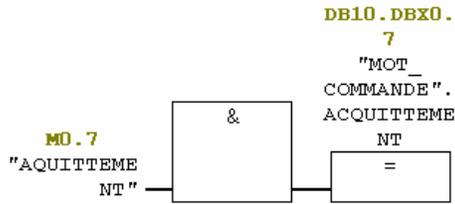
Réseau 2: ARRET D'URGENCE DE SECURITE;

arrêt rapide du variateur.



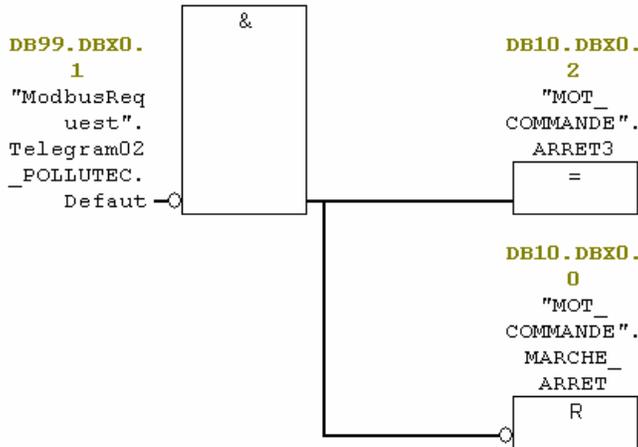
Réseau 3 : AQUITTEMENT

Aquitteemnt des défauts.



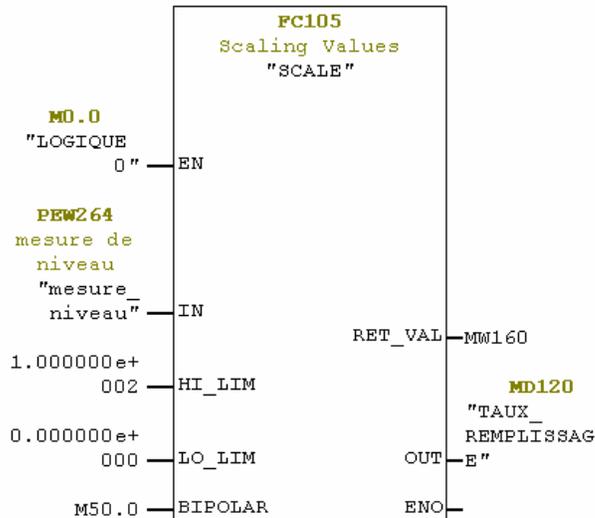
Réseau 4 : ARRET RAPIDE

Si le niveau de l'eau dans le réservoir 2 n'a pas été transmis, on doit arrêter le moteur p éviter le débordement.



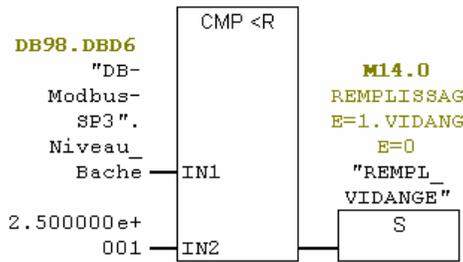
Réseau 5 : MISE A L'ECHELLE DE LA MESURE DE NIVEAU

la mise à l'échelle de la mesure de niveau analogique, la valeurs analogique maxiamale (20mA)correspond à 60L, elle donne une valeur reel pour chaque signal reçu.



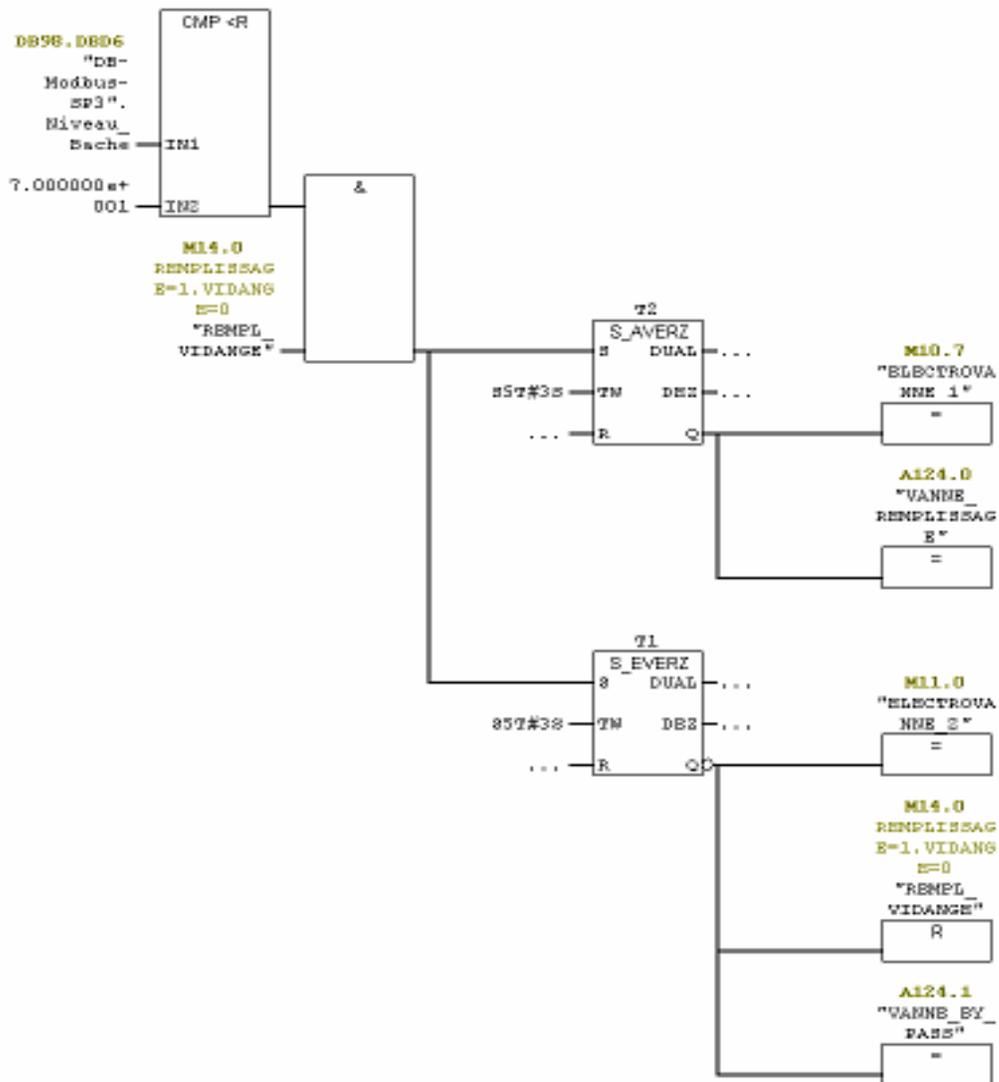
Réseau 6 : GESTION DU REMPLISSAGE DES RESERVOIRS.

Remplissage=1, Vidange =0



Réseau 7 : GESTION DES VANNES

tant que le taux de remplissage est inférieur à 65L, le premier reservoir se remplit, sinon, l'electrovanne se ferme et le reservoir 2 se remplit.



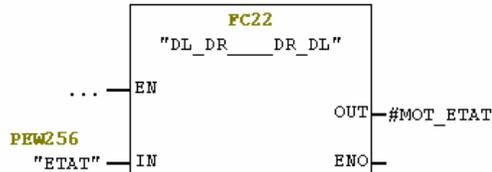
C.1.2. La fonction de gestion du MASTERDRIVE FC12

FC12 : GESTION DU MASTER DRIVE

Envoie du mot de commande et réception du mot d'état.

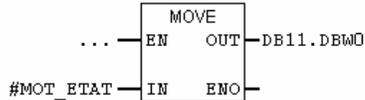
Réseau 1 : INVERSION DU MOT

A la réception du mot d'état, on inverse le mot pour la cohérence.



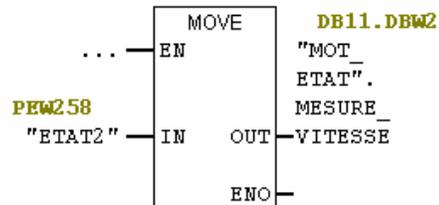
Réseau 2 : LECTURE DU MOT D'ETAT

Mettre le mot d'état sur le bloc de donnée DB11



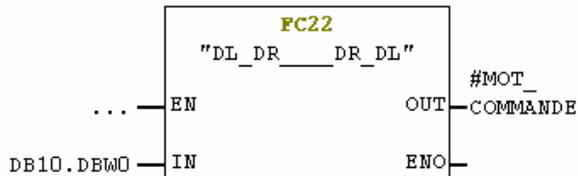
Réseau 3 : RECEPTION DE LA MESURE DE VITESSE

Dans la régulation avec capteur de vitesse on reçoit la mesure de vitesse du variateur.



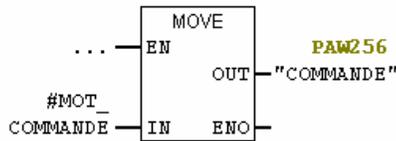
Réseau 4 : INVERSION DU MOT DE COMMANDE AVANT L'ENVOIE

Inversion du mot de commande pour la cohérence.



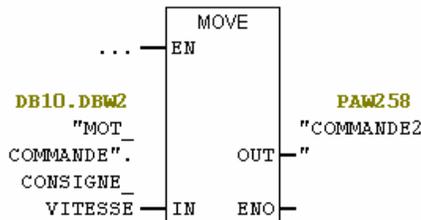
Réseau 5 : ENVOIE DU MOT DE COMMANDE

On envois le mot de commande qui est contenue dans le bloc de donnée DB10



Réseau 6 : ENVOIE DE LA CONSIGNE DE VITESSE

envoi de la consigne de vitesse au variateur.



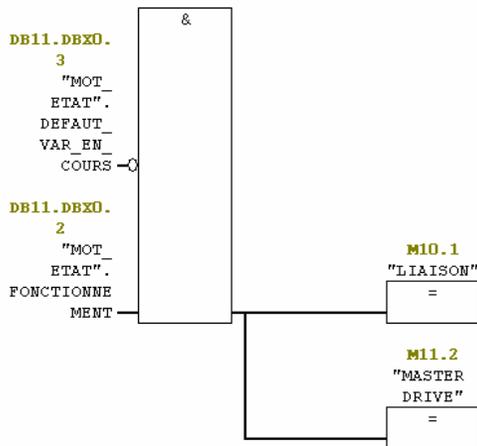
C.1.3. La fonction de gestion de la HMI FC21

FC21 : GESTION DE HMI

LE PROGRAMME POUR GERER L'INTERFACE HOMME MACHINE.

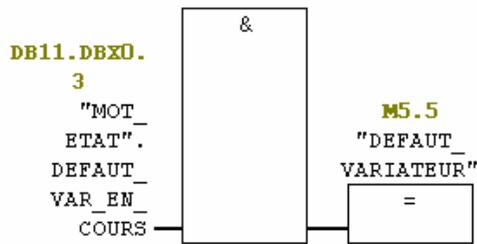
Réseau 1: Etablissement de la liaison

Dés que le moeur est en état prêt à l'enclenchement, la liaison est établie et on le visionne qu l'interface HMI

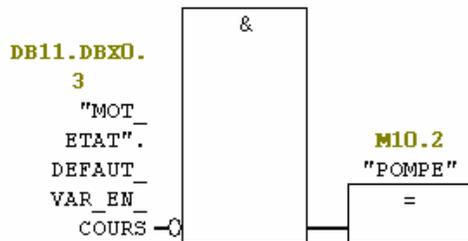


Réseau 2 : DIAGNOSTI DEFAUT DU VARIATEUR

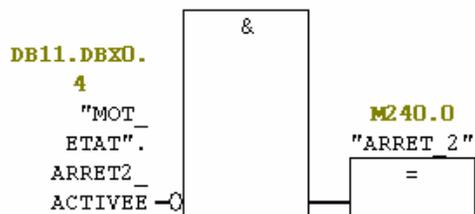
Lors de la réception du mot d'état, on diagnostique les défauts; ça apparaît aussitôt sur l'interface HMI.

**Réseau 3 : ARRET DE LA POMPE**

Dès que le variateur est en défaut, ceci est visonné sur la HMI

**Réseau 4 : ARRET PAR LE VARIATEUR**

Acquisition de l'information d'arrêt du variateur dans la M240.0



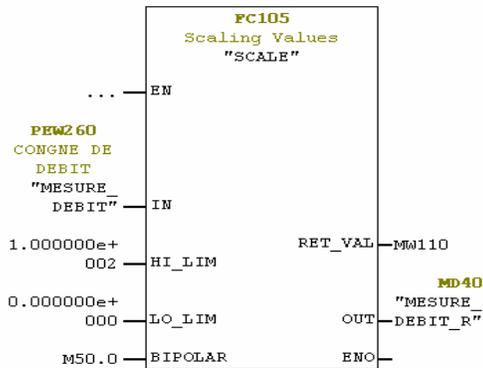
C.2. Le bloc d'organisation cyclique OB35

OB35 : "Cyclic Interrupt"

Ce bloc de Cyclique est utilisé pour la régulation.

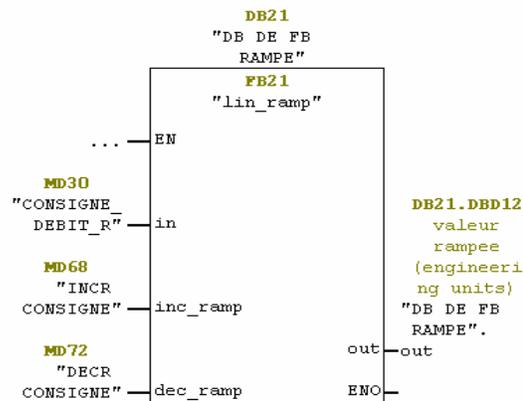
Réseau 1 : mise à l'échelle la consigne de débit

la mise à l'échelle de la consigne de débit afin d'envoyer une valeur réel.



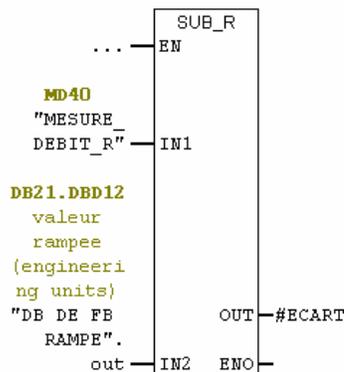
Réseau 2 : GENERATION DE LA RAMPE

Bloc de génération de la rampe au démarrage.



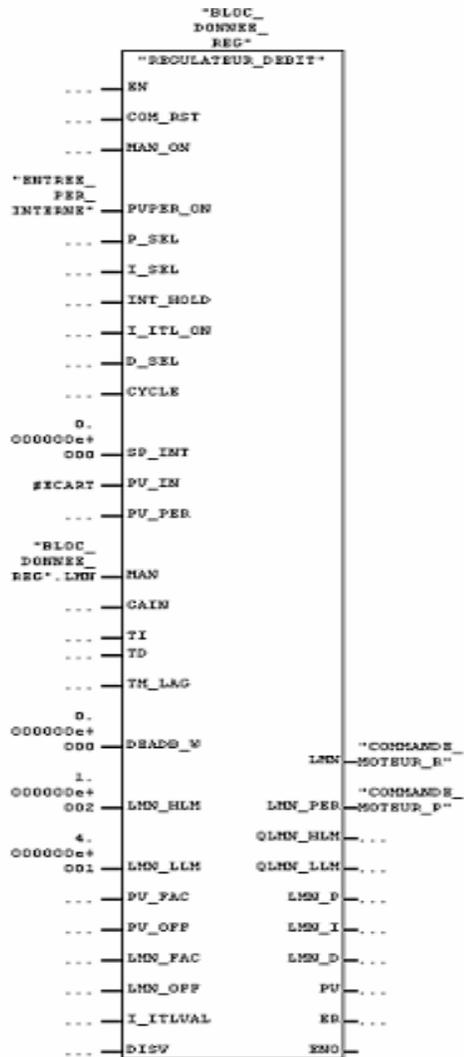
Réseau 3 : CALCUL DE L'ECART ENTRE CONSIGNE ET MESURE

On calcule l'écart entre la mesure et la consigne, afin de l'envoyer à l'entrée du régulateur de débit

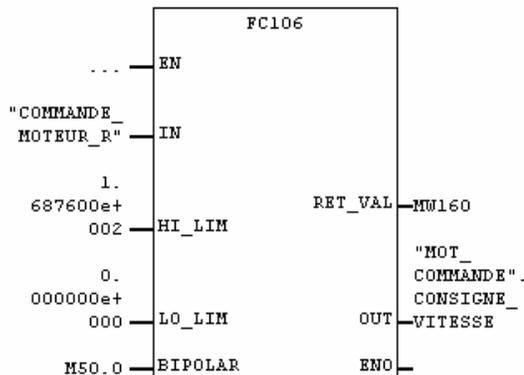


Réseau 5 : REGULATEUR DE DEBIT

Le bloc FB41_Cont pour la régulation du débit. l'entrée est l'ecart consigne/mesure, et à la sortie la commande en vitesse vers la pompe

**Réseau 6 : MISE A L'ECHELLE POUR LA COMMANDE DU MOTEUR**

Adapter la commande de vitesse à la sortie du régulateur, à la plage admissible à l'entrée du variateur



ANNEXE D

Descriptions des éléments électrique et hydraulique utilisés au cours de l'application « Gestion d'une station de pompage à base d'automates Siemens »

- D.1. La CPU 314**
- D.2. Le variateur SIMOVERT MASTERDRIVE de SIEMENS**
- D.3. La pompe *Multi 1000 D* de *KSB***
- D.4. Le capteur de débit SITRANS F M MAGFLO MAG 5100 W de SIEMENS**
- D.5. La sonde de niveau SITRANS PROBE LU de SIEMENS**
- D.6. Les électrovannes FASCO**
- D.7. Le terminal d'exploitation MP270**

D.1. La CPU 314

Elle constitue l'élément principal de la S300 utilisé, et présente les caractéristiques suivantes :

| | | |
|--------------------------------------|---|-----------------------------|
| Version CPU | N° série | 6ES7-314-1AE04-0AB0 |
| | Version du matériel | 0.1 |
| Mémoires | Mémoire de travail | 24Ko non extensible |
| | Mémoire de chargement | RAM de 48 Ko non extensible |
| Temps de traitement (minimal) | Opération sur bit | 0.3 μ s |
| | Opération sur mot | 1 μ s |
| | Opération arithmétique sur nombres entiers | 2 μ s |
| | Opération arithmétique sur nombres à virgule flottante | 50 μ s |

Tableau D.1 : Caractéristiques de la CPU 314

D.2. Le variateur SIMOVERT MASTERDRIVE de SIEMENS

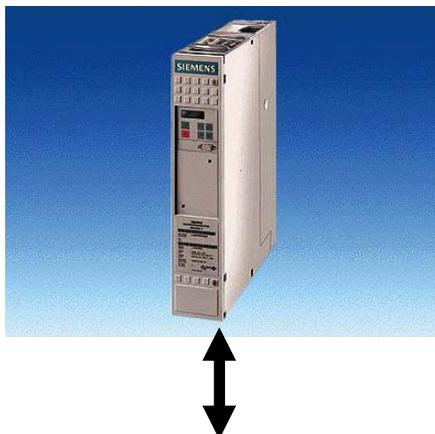
SIMOVERT MASTERDRIVES est une gamme complète de variateur. Les nombreuses variantes matérielles conjuguées à différentes fonctionnalités de régulation permettent une adaptation optimale aux applications les plus variées, et balayant une plage de puissance allant de 0,55 kW à 2300 kW.

La fonctionnalité de régulation est établie par le logiciel contenu dans les modules convertisseurs.



Figure D.1 : La gamme SIMOVERT MASTERDRIVES de SIEMENS

La variante XAU703-002938, utilisée lors de la deuxième application « **Réalisation et Gestion d'une station de pompage à base d'automates SIEMENS** » se caractérise par les données électriques suivantes :



| | |
|-------------------------------------|---------------------|
| N° série Siemens | 6SE6440-2AB11-2AA1 |
| Entrée en tension nominale | 380-480 V +10% -10% |
| Entrée en courant nominale | 6.1 A |
| Fréquence du réseau d'alimentation | 47-63 Hz |
| Puissance Max du moteur à entraîner | 2.2 kW |

Tableau D.2 : Caractéristiques de la variante XAU703-002938

Le groupe de produits **SIMOVERT MASTERDRIVES** comprend les fonctionnalités de régulation suivantes :

- régulation vectorielle avec capteur pour applications très contraignantes en termes de dynamique et de précision du couple ;
- régulation vectorielle sans capteur pour applications simples (par ex. pompes, ventilateurs)
- commande par caractéristique U/f .
- *Possibilité de mise en réseau PROFIBUS-DP grâce à la carte de communication intégrée CBP.*

D.3. La pompe *Multi 1000 D* de KSB

C'est une pompe centrifugeuse, dont le mouvement du fluide résulte de la force centrifuge qui lui est communiqué par une roue à volute spirale.

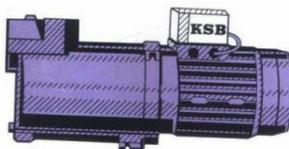


Figure D.2 : La pompe *Multi 1000 D* de KSB

Elle comporte cinq étages de refoulement en série, et aussi:

- une roue à aubes tournant autour de son axe
- un distributeur dans l'axe de la roue
- un collecteur de section croissante, en forme de spirale appelée volute.

Ces pompes sont les plus utilisées dans le domaine industriel pour leur simplicité et de leur faible coût.

Elle présente les caractéristiques techniques suivantes :

- Débit maximum de 133 L/minute au régime nominal.
- Hauteur manométrique maximum d'aspiration de 7m.
- Hauteur manométrique de refoulement de 47m.
- Pression à la bride de refoulement de 6 bars.
- Température de fonctionnement maximum de 70°C.
- Moteur d'entraînement triphasé 230 V, consommant 0,93 W en fonctionnement nominale, et tournant à 2800tr/minute pour 50Hz.



Figure D.3 : La caractéristique Débit/HMT en régime nominale de la pompe 1000D

La pompe présente des caractéristique en refoulement largement supérieure à dont on a besoin pour notre application, et pour l'adapter, on a enlevé trois des cinq étages de refoulement, ce qui réduit la pression au refoulement, et ainsi la HMT en régime nominale de 3/5 (à cause de la disposition série des étages), ce qui donne une nouvelle HMT de 18M.

D.4. Le capteur de débit SITRANS F M MAGFLO MAG 5100 W de SIEMENS

Le capteur électromagnétique *MAGFLO 5100 W* est une sonde pour toutes les applications de l'eau telles que des applications d'eaux souterraines, d'eau potable, d'eaux résiduaires, d'eaux d'égout ou de cambouis. Sa conception et son exactitude même pour les faibles débits la rend particulièrement utile pour la détection de fuite.



Figure D.4 : *La sonde de débit SITRANS F M MAGFLO MAG 5100 W de SIEMENS*

D.4.1. Caractéristiques techniques

- Pression maximum de fonctionnement : 40 bars.
- Température de fonctionnement : de 5° à 70° C.
- La fiabilité est garantie sur toute la plage de température et de pression délimitée.

Le capteur comporte un module d'affichage et de communication *F M MAGFLO MAG 5000*, robuste et approprié pour des applications totales et a une exactitude de mesure de $\pm 0.5\%$ du débit



Figure D.5 : *Le module de communication F M MAGFLO MAG 5000 de SIEMENS*

D.4.2. Caractéristiques techniques du module de transmission

- Menu configurable d'opération d'utilisateur avec la protection par mot de passe.
- Interface multilingue d'affichage et de bloc de touches.
- Température de fonctionnement : -20° à 60°C .
- Auto-diagnostic complet, pour l'indication des erreurs de mesure.
- Transmission analogique, présentant la caractéristique linéaire suivante :

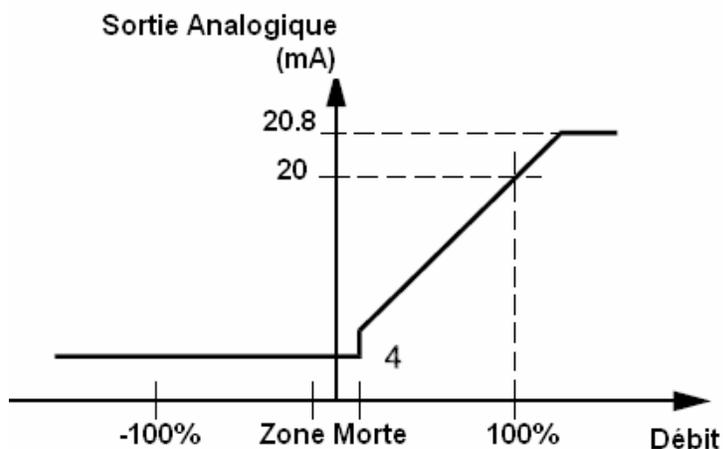


Figure D.5 : Caractéristique de la sortie analogique 4-20mA du capteur de débit

D.5. La sonde de niveau SITRANS PROBE LU de SIEMENS

SITRANS Probe LU est un capteur ultrasonique très performant. Ce système est composé d'une électronique associée à un transducteur et un raccord vers les modules analogiques des automates.

Le SITRANS Probe LU s'utilise avec un transducteur offrant une très grande flexibilité d'utilisation et une résistance optimale aux produits chimiques corrosifs.



Figure D.6 : *le capteur de niveau SITRANS PROBE LU*

Et offre les fonctionnalités suivantes :

- Sortie de mesure analogique 4-20mA.
- Calibrage du *niveau vide* et du *niveau Max* par le biais d'un programmeur portatif.
- Un système anti-écho paramétrable est intégré, pour éviter les mesures erronées.
- Le transducteur ultrasonique est doté d'un capteur pour compenser les variations de température

D.6. Les électrovannes FASCO

Les électrovannes ASCO/JOUCOMATIC, conforme aux Directives CE applicables, vérifiant les tests de fiabilité EN 60730-2-8 (*CENELEC*) et FCI-82-1 (*Fluid Controls Institute*) :



Figure D.7 : *L'électrovanne FASCO à DN40*

Elle est normalement fermée, et Commandable en TOR, à 24 Volt DC, consommant de 4 à 6 Watt, et se présente avec les caractéristiques suivantes :

- Electrovanne à deux orifices.
- Pas de pression minimale de fonctionnement
- Electrovanne conçue pour éliminer les coups de bélier à la fermeture sur eau
- Dispositif spécifique d'auto-nettoyage intégré dans l'orifice de pilotage de la membrane garantissant des performances optimales
- Temps de d'ouverture de 300ms, et un temps de rappel de 500ms.



Figure D.8 : *Coupe verticale de l'électrovanne*

D.7. Le terminal d'exploitation MP270

Le terminal d'exploitation **MP270** représente l'entrée dans la classe des plateformes multifonctionnelles. Elles offrent une exécution optimale. Basé sur le logiciel de conception d'interface homme-machine **Win CC**, puissant et riche sous le SE **Windows**, elles sont caractérisées par flexibilité et possibilités de connectivité variées.



Figure D.7 : Coupe verticale de l'électrovanne

D.7.1. Caractéristique technique

- Ecran 10.4" TFT-Display
- Résolution 640 x 480 pixels
- 64000 couleurs
- 36 boutons système,
- Mémoire utilisateur de 6 MB
- Connexions *MPI-*, *Profibus-DP*, *Ethernet et USB integr.*
- Windows CE
- Configuration via *Win CC flexible* 2005 SP1 Standard et plus récent.
- Dimensions du Panel: 483 x 310 mm (L x H)
- Profondeur: 60 mm.

ANNEXE E

La commande scalaire U/f des moteurs asynchrones

La commande U/f se base sur la mesure de grandeurs scalaires (valeurs d'amplitude en tension et en fréquence). C'est le système de commande de base des variateurs de fréquence.

Afin de garder un flux constant dans le moteur et donc aussi une variation de vitesse à couple constant la tension et la fréquence varient proportionnellement jusqu'à la fréquence nominale du moteur (50 Hz). Lorsque la tension nominale est atteinte, la tension ne sachant plus augmenter, il est toujours possible d'augmenter la fréquence; dans ce cas la variation se fait à puissance constante, le couple diminue avec la vitesse. Ce mode de fonctionnement est intéressant pour des charges à couple constant tels que les ascenseurs. En effet, le couple moteur "colle" mieux au profil du couple résistant; ce qui signifie que les consommations qui en découlent sont moindres.

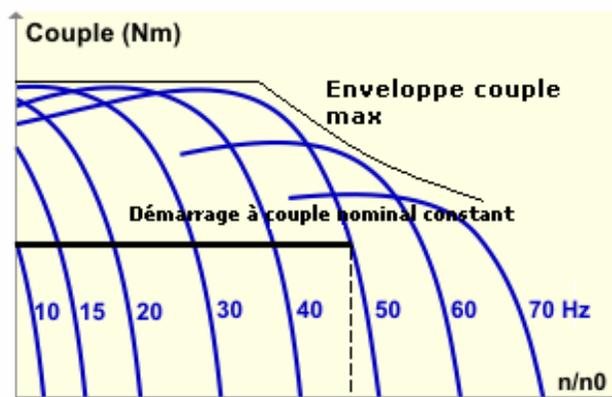


Figure E.1: La courbe du couple en fonction de la vitesse de rotation pour différentes fréquences.

ANNEXE F

La télétransmission mise en œuvre dans l'application « Réalisation et gestion d'un prototype de station de pompage à base d'automates SIEMENS »

- F.1. Qu'est ce que la télétransmission en milieu industriel ?**
- F.2. Le Protocole MODBUS**

F.1. Qu'est ce que la télétransmission en milieu industriel ?

Un système de Télétransmission permet le transport de signaux situés à des endroits distants. Par exemple, pour la distribution d'eau potable, les stations de pompage sont généralement dispersées géographiquement et nécessitent d'échanger les informations relatives aux divers niveaux des châteaux d'eau, aux débits et aux pressions afin de réguler le pompage en fonction de la demande des utilisateurs. Ces stations peuvent ainsi s'échanger des informations à travers un réseau de communication.

N'importe quelle situation anormale est immédiatement détectée et une séquence d'alarmes peut être envoyée vers divers destinataires pour qu'une décision puisse être prise en compte, en vue de corriger l'anomalie.



Figure F.1 : Principe d'accès du maître vers l'esclave

C'est la raison pour laquelle nous avons voulu insérer une communication par télétransmission dans notre application « **Réalisation et gestion d'un prototype de station de pompage à base d'automates SIEMENS** », cette solution de communication et de surveillance sert à transmettre le niveau dans le château d'eau, et ainsi gérer les deux circuits de remplissage et de « **BY PASS** » en pilotant les deux électrovannes.

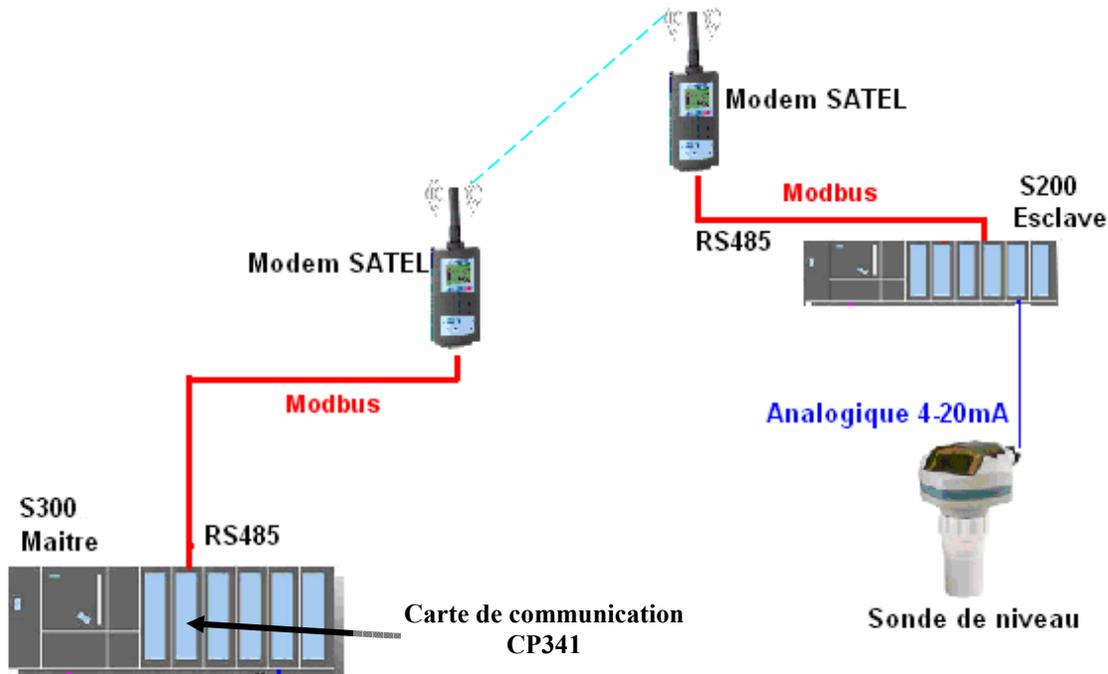


Figure F.2 : *Synoptique de télétransmission*

Un système de télétransmission est basé sur un automate maître, relié à un modem pour la transmission de données via un support de communication choisi selon les conditions géographique et technologiques imposées, et permettant une intelligence sur les sites industriels en assurant une *communication* entre le maître et ses esclaves.

Dans notre application, l'automate **S300** constitue le maître, qui est l'élément central vers qui converge toutes les données processus, et de l'autre côté, le **S200** joue le rôle de l'esclave, qui fait l'acquisition de la grandeur "**niveau**" dans le « **Réservoir 1** », et la transmet par la suite vers le maître.

Le protocole MAITRE-ESCLAVE est de type **MODBUS**, qui se fait via une technique de transmission sans fil par le biais de deux modems SATEL. Le signal est de type **RADIO**, à une fréquence de **411,05 Mhz**, et une portée de **35 Km** avec une puissance de **10Watt**.

Le processeur de communication **CP 341** permet de réaliser un échange de données entre l'automate S300 et d'autres systèmes d'automatisations ou calculateurs, et ceci en intégrant des principaux protocoles de transmission dans le *firmware* du module. Il comporte une interface série intégrée **RS 485**, avec une vitesse de transmission atteignant **19,2 K Bauds**.



Figure F.3 : *Le module de communication CP341*

F.2. Le Protocole MODBUS

Le protocole *MODBUS* (marque déposée par MODICON) est un protocole de dialogue basé sur une structure hiérarchisée entre un maître et plusieurs esclaves.

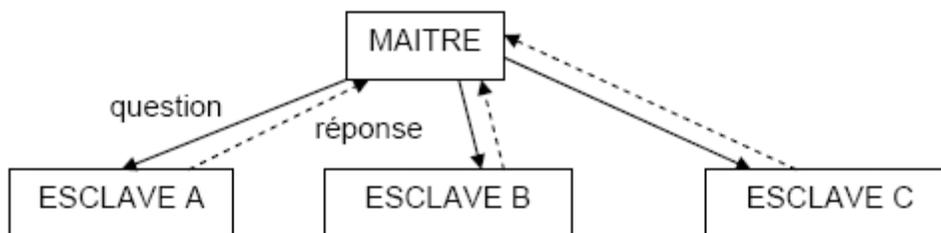


Figure F.4 : *Le principe mono maître du protocole MODBUS*

Le protocole MODBUS consiste en la définition de trames d'échange. Le maître envoie un message constitué de la façon suivante:

- Adresse de l'esclave concerné, pour établir la liaison avec lui,
- Instruction,
- Donnée,
- Contrôle, calculé sur l'ensemble du message et destiné à assurer l'intégrité de l'échange.

Un contrat d'échange doit être créé, et ceci pour définir la table de réception, dédié à l'écriture du maître dans l'esclave, et une table d'émission pour la lecture du maître dans l'esclave.

(Figure F.5).

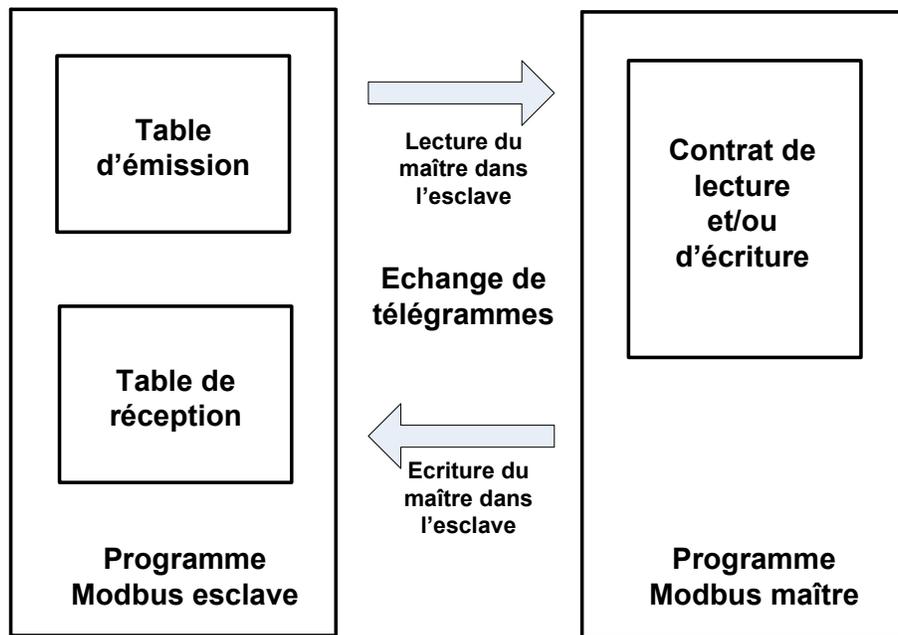


Figure F.5 : Principe des échanges MODBUS

Il existe deux types de dialogue possible en **MODBUS** :

➤ **Echange maître vers l'esclave**

Le maître envoie une **demande** et attend une **réponse** (Relation maître – esclave).

Le CP prend l'initiative de la transmission (= maître), puis attend, après avoir envoyé un télégramme de demande, un télégramme de réponse de l'esclave pendant la durée qui a été paramétrée comme temps de surveillance de la réponse, et c'est le type de dialogue utilisé dans l'application.

➤ **Echange Maître vers tous les esclaves (Message Broadcast)**

Le maître utilise l'adresse 0 pour s'adresser à tous les esclaves du bus. Il diffuse un message à tous les esclaves présents sur le réseau, ceux-ci exécutent l'ordre du message sans émettre une réponse.

ملخص:

العمل المنجز في المذكرة يتمحور حول استعمال مسير صناعي ميرمج "سيمنس". قمنا بإنجاز نموذج محطة ضخ المياه. التسيير، التحصيل ومعالجة المعلومات محقق من طرف هذا المسير الصناعي. لضبط التصريف، استعنا بطريقة "زيكلر-نيكولس" و"شين-رونز-رزيك" لتحديد خصائص الضابط، وبعدها يأتي تحليل النتائج فالخلاصة:

الكلمة المفتاح:

مسير صناعي ميرمج "سيمنس"؛ ضابط السرعة "سيموفر"؛ عداد التصريف الكهرو مغناطيسي؛ عداد المستوى فوق الصوتي؛ شبكة الاتصال المحلية "بروفيبوس"؛ "لوين سيبي فليكسيبل".

RESUME :

Le travail présenté dans ce mémoire est basé essentiellement sur l'utilisation des automates programmables SIEMENS. Nous avons entrepris la réalisation d'un prototype de station de pompage hydraulique. La gestion du prototype, l'acquisition et le traitement des données processus ainsi que le diagnostic sont assurés par un automate S300.

Pour la régulation du débit, nous avons utilisé la méthode de *Ziegler-Nichols* et de *Chien-Hrones-Reswick* pour la détermination des paramètres du régulateur P.I, et ensuite, analyser les résultats et tirer des conclusions.

Mot clé :

Automate programmable Siemens ; Variateur de vitesse SIMOVERT ; Capteur de débit ultrasonique, Capteur de débit électromagnétique ; Réseau local PROFIBUS ; Win CC flexible.

ABSTRACT:

The work presented in this memory is based primarily on the use of the programmable automats SIEMENS. We undertook the realization of a prototype of hydraulic station of pumping. The management of the prototype, the data process acquisition and processing as well as the diagnostic are ensured by S300 automats.

For the regulation of the flow, we used the method of *Ziegler-Nichols* and *Chien-Hrones-Reswick* for the determination of the parameters of regulator P.I, and then, to analyze the results and to draw the conclusions.

Key words:

Programmable Logic Controller, speed driver SIMOVERT; flow transmitters; level transmitters; PROFIBUS DP industrial network, Human Machine Interface Win CC flexible.

AUSZUG:

Die Arbeit, die in diesem Gedächtnis dargestellt wird, basiert hauptsächlich auf dem Gebrauch der programmierbaren Automats SIEMENS. Wir nahmen uns die Realisierung eines Prototyps der hydraulischen Station des Pumpens auf. Das Management des Prototyps, die Daten verarbeiten Erwerb und die Verarbeitung, sowie die Diagnose durch Automats S300 sichergestellt werden.

Auf die Regelung des Flusses, verwendeten wir die Methode von *Ziegler-Nichols* und von *Chien-Hrones-Reswick* für die Ermittlung der Parameter des Reglers P.I und dann, um die Resultate zu analysieren und die Zusammenfassungen zu zeichnen.

Schlüsselwörter:

Programmierbarer Logik-Kontrollleur, Geschwindigkeit Treiber SIMOVERT; Durchflußgeber; waagrecht ausgerichtete übermittler; PROFIBUS DP industrielles Netz, menschlicher Maschine Schnittstelle Gewinn cm flexibel.