

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Ecole Nationale Polytechnique

Département d'Électrotechnique

Laboratoire de Recherche en Electrotechnique



Mémoire de Projet de Fin d'Etudes

Pour l'Obtention du Diplôme d'Ingénieur d'Etat en Electrotechnique

Thème

**Modernisation d'une Bobineuse à 7 Faisceaux avec un
API S7-1200 et un Ecran Tactile HMI KTP400 basic**

DELLOUL Abdellah

Présenté et soutenu publiquement le : 21/06/2023

Composition du Jury :

Président	M. Redouane TAHMI	Professeur	ENP
Encadreur	M. Achour HARBIT	Ingénieur	Electro-Industries
Co-encadreur	M. Kamel BOUGHRARA	Professeur	ENP
Examineur	M. El Madjid BERKOUK	Professeur	ENP

ENP 2023

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Ecole Nationale Polytechnique

Département d'Électrotechnique

Laboratoire de Recherche en Electrotechnique



Mémoire de Projet de Fin d'Etudes

Pour l'Obtention du Diplôme d'Ingénieur d'Etat en Electrotechnique

Thème

**Modernisation d'une Bobineuse à 7 Faisceaux avec un
API S7-1200 et un Ecran Tactile HMI KTP400 basic**

DELLOUL Abdellah

Présenté et soutenu publiquement le : 21/06/2023

Composition du Jury :

Président	M. Redouane TAHMI	Professeur	ENP
Encadreur	M. Achour HARBIT	Ingénieur	Electro-Industries
Co-encadreur	M. Kamel BOUGHRARA	Professeur	ENP
Examineur	M. El Madjid BERKOUK	Professeur	ENP

ENP 2023

Remerciements

Je voudrais tout d'abord remercier Dieu tout puissant pour m'avoir donnée la force et la patience pour réaliser ce modeste travail.

Je tiens à remercier mes chers parents qui ont été une telle source de courage et persévérance. Merci "Oumina" pour ton soutien émotionnel et physique inconditionnel. Merci "Abina" pour ta présence et encouragement. Merci Khadidja Sérine et Abadi. Que Dieu vous garde.

Merci à mes grands parents, mes grands frères et grandes soeurs pour avoir cru en moi et encouragé depuis le début.

Un grand merci Mr. HARBIT et Pr. BOUGHRARA pour votre soutien et votre patience, j'ai été honoré de travailler avec vous sur un tel sujet. Merci Messieurs pour votre encadrement de qualité.

Je remercie les membres du jury pour l'intérêt porté à mon humble travail et d'avoir accepté de l'examiner et l'enrichir par leurs propositions.

Merci mes amis pour tous les bons moments, les sourires et les joies partagés.

Merci à tous ceux qui me sont chers.

Merci "Moi", sans toi je n'y serai jamais parvenu.

ملخص

في السنوات الأخيرة ، أصبحت الأتمتة ، التي لها تاريخ طويل بالفعل ، مصدر قلق كبير للعديد من الشركات. لهذا قدمت لي شركة Electro-Industries هذا المشروع ، الذي يركز على تحسين جزء التحكم في آلة اللف وأتمتتها باستخدام أتمتة قابلة للبرمجة (CPU S7-1200 (1214C) من شركة Siemens. سعينا أيضاً إلى تحسين نظام الإشراف باستخدام لوحة اللمس KTP400.

لقد استخدمت بوابة TIA كأداة نمذجة ، برنامج برمجة متعدد اللغات ، والذي يتضمن Step-7 و WinCC. سمح لي هذا بتطوير حل قابل للبرمجة لسلسلة المعالجة هذه ، ثم تكوين لوحة اللمس KTP400 باستخدام برنامج WinCC المرن.

الكلمات الرئيسية : وحدة التحكم المنطقية القابلة للبرمجة (PLC) ، واجهة الإنسان والآلة (HMI) ، بوابة TIA ، اللف ، التشفير ، المحرك ثلاثي الأطوار ، عدد الدورات ، عدد الحزم ، الإزاحة ، الحماية الكهربائية ، محرك السرعة المتغيرة ، المدخلات \ المخرجات ، التعليمات ، البرنامج ، لغة السلم ، الأتمتة ، المراقبة ، الأوامر.

ABSTRACT

In recent years, automation, which already has a long history, has again become a major concern for many companies. For this the company Electro-Industries offered me this project, centered on the improvement of the control part of a winding machine and its automation using a programmable automaton S7-1200 (CPU 1214C) of the Siemens company. We also sought to improve the supervision system by using a KTP400 touch panel.

We used the TIA Portal as a programming tool, a software which includes Step-7 and WinCC. This allowed us to develop a programmable solution for this processing chain, then to configure the KTP400 touch panel using the WinCC Flexible software.

Keywords : Programmable logic controller (PLC), Human-machine interface (HMI), TIA Portal, Winder, Encoder, Three-phase motor, Number of turns, Number of beams, Displacement, Electrical protection, Variable speed drive, Inputs /outputs, Instructions, Program, Ladder Language, Automation, Monitoring, Command.

RÉSUMÉ

Au cours des dernières années, l'automatisation, qui possède déjà une longue histoire, est devenue une préoccupation majeure pour de nombreuses entreprises. Pour cela l'entreprise Electro-Industries m'a proposé ce projet, axé sur l'amélioration de la partie commande d'une machine bobineuse et son automatisation à l'aide d'un automate programmable S7-1200 (CPU 1214C) de la société SIEMENS. Nous avons également cherché à améliorer le système de supervision en utilisant un écran tactile KTP400.

Nous avons utilisé le TIA Portal comme un outil de programmation, un logiciel qui regroupe le Step-7 et le WinCC. Cela nous a permis de développer une solution programmable pour cette chaîne de traitement, puis de configurer l'écran tactile KTP400 à l'aide du logiciel WinCC Flexible.

Mots Clés : Automate programmable industriel (API), Interface homme-machine (IHM), TIA Portal, Bobineuse, Encodeur, Moteur triphasé, Nombre de spires, Nombre de faisceaux, Déplacement, Protection électrique, Variateur de vitesse, Entrées/sorties, Instructions, Programme, Langage Ladder, Automatisation, Surveillance, Commande.

Table des matières

Table des figures

Introduction générale	9
1 Description de la bobineuse	11
Introduction	11
1 Présentation de l'organe d'accueil	11
2 Présentation du complexe d'Electro-Industries	11
1.2.1 Unité moteurs, alternateurs et groupes électrogènes	11
1.2.2 Unité de transformateurs	12
3 Présentation des différents blocs constituant la machine	12
4 Désignation des éléments de conduite et de réglage	12
5 Caractéristiques techniques de la machine	16
6 Description des éléments constitutifs	17
7 Mise au point de la machine	17
8 Réglage de la commande	17
9 Mise au point de la machine	18
10 Fonction et mode de travail	18
11 Description de l'armoire électrique	19
1.11.1 La protection électrique	19
1.11.1.1 Sectionneur	19
1.11.1.2 Fusible	19
1.11.1.3 Disjoncteur	20
1.11.1.4 Contacteur	20
12 Le transformateur d'isolement	21
13 Le variateur de vitesse	21
14 Encodeur rotatif	22
15 Capteur de proximité inductif	23
16 Problématique	23
17 Objectifs	24
Conclusion	24
2 Description de l'automate S7-1200 CPU 1214C	25
Introduction	25
1 Architecture des automates	25
2.1.1 Unité centrale	26
2.1.1.1 Unité de contrôle	26
2.1.1.2 Unité arithmétique et logique	26
2.1.1.3 Registres	26

2.1.2	Mémoires	27
2.1.3	Entrées et sorties	27
2.1.3.1	En modules d'entrées ou de sorties	27
2.1.3.2	Intégrées	28
2.1.3.3	Déportées	28
2.1.4	Modules de communication	29
2.1.5	Alimentation électrique	29
2.1.6	Éléments d'interface homme-machine	29
2	Fonctions de base	29
2.2.1	Fonctions logiques ou booléennes	29
2.2.2	Fonctions arithmétiques et sur mots	30
2.2.3	Fonctions de communication	30
2.2.4	Autres fonctions	31
3	Langages	31
2.3.1	Langage à contacts (ladder diagram : LD)	32
4	Exécution du programme	33
5	Rôle de l'API dans un système de production	34
6	Les objectifs de l'automatisation	35
7	Présentation de l'automate programmable S7-1200	36
8	Caractéristiques techniques de l'API S7-1200 1214C	37
	Conclusion	38
3	Programmation de L'automate	39
	Introduction	39
1	La conception d'un programme avec TIA Portal V13	39
2	Création d'un nouveau projet	40
3	Vue du portail	41
4	Configuration matérielle	42
5	Création du bloc de données	44
6	Programme	47
3.6.1	Mise à l'échelle des données de l'encodeur	47
3.6.2	Affectation des consignes	48
3.6.3	Contrôle du moteur de la vis-mère	49
3.6.4	Commande du moteur de bobinage	55
3.6.4.1	Moteur en vitesse lente	55
3.6.4.2	Moteur vitesse rapide	58
3.6.5	Comptage de l'opération	61
3.6.6	Fin du bobinage	63
3.6.7	Surveillance de la machine	66
3.6.8	Remise à zéro du compteur rapide	71
	Conclusion	73
4	Configuration de l'écran tactile	74
	Introduction	74
1	Définition de la supervision	74
2	Avantages de la supervision	74
3	Interface homme machine	75
4	Choix de l'Interface Homme-Machine	75
5	WinCC flexible	75

6	Configuration de l'IHM	76
4.6.1	Ajout de l'interface	76
4.6.2	Configuration des vues	78
4.6.3	Vue globale	84
4.6.4	Alarmes IHM	84
	Conclusion	85
	Conclusion générale	86

Bibliographie	87
----------------------	-----------

Table des figures

1.1	Photographie de la machine	14
1.2	Contacteur	20
1.3	Variateur de vitesse	22
1.4	Encodeur rotatif	23
1.5	Capteur à proximité inductif	23
2.1	Architecture d'un API	26
2.2	Circuit d'entrée TOR d'API avec protection	27
2.3	Exemple d'opérations logiques sur TIA Portal	30
2.4	Exemple d'addition et soustraction sur le TIA Portal	30
2.5	Compteur incrémental	31
2.6	Réseau représentant la fonction logique ET	32
2.7	Réseau utilisant le saut	33
2.8	Ordonnancement de l'exécution	33
2.9	Automate en commande	34
2.10	Automate programmable S7-1200 1214C	37
3.1	Organigramme pour création d'un projet sous TIA Portal V13	40
3.2	Fenêtre de création d'un nouveau projet	41
3.3	Vue du portail	41
3.4	Ajout de la CPU	42
3.5	Configuration de l'appareil	42
3.6	Activation du compteur rapide	43
3.7	Configuration du compteur rapide	43
3.8	Activation du memento de cadence	44
3.9	Ajout d'un bloc de données	44
3.10	Remplissage du bloc	45
3.11	Compilation du bloc de données	46
3.12	Conversion de DInt à Real	47
3.13	Calcul de l'équivalent en millimètres	48
3.14	Conversion de DInt à Real	48
3.15	Affectation du nombre de spires	49
3.16	Contrôle du moteur de la vis-mère partie 1	50
3.17	Contrôle du moteur de la vis-mère partie 2	50
3.18	Contact de maintien	51
3.19	Coupure par contacts à ouvertures	51
3.20	Les affectations des sorties	52
3.21	Les mementos de redémarrage	52
3.22	Vue globale du réseau	53
3.23	Condition redémarrage moteur 2	53

3.24	Déplacement sens direct	54
3.25	Choix de la vitesse de rotation	54
3.26	Contrôle du moteur 1 vitesse lente partie 1	55
3.27	Contrôle du moteur 1 vitesse lente partie 2	55
3.28	Affectation 1	56
3.29	Affectation 2	56
3.30	Vue globale du réseau	57
3.31	Condition redémarrage moteur1 vitesse lente	57
3.32	Bobinage vitesse lente	58
3.33	Contrôle du moteur 1 vitesse rapide	58
3.34	Affectation	59
3.35	Vue globale du réseau	59
3.36	Condition redémarrage moteur 1 vitesse rapide	60
3.37	Bobinage vitesse rapide	60
3.38	Comptage des faisceaux	61
3.39	Comptage du cycle	62
3.40	Mise à 0 des sorties et mémentos de commande	63
3.41	Retour à la position initiale	63
3.42	Conditions de calcul du déplacement total	64
3.43	Calcul du déplacement total dans le premier cycle	64
3.44	Calcul du déplacement total dans le deuxième cycle	65
3.45	Vue globale du réseau	66
3.46	Déclaration de la variable "Défauts"	66
3.47	Mise à 1 du memento %M3.5	67
3.48	Mise à 1 des deux mémentos %M3.0 et %M3.1	67
3.49	Conditions de la mise à 1	68
3.50	Mise à 1 de chaque memento	68
3.51	Affectation des sorties des alarmes	69
3.52	Désactivation des alarmes	70
3.53	Désactivation du "Défaut capot ouvert"	71
3.54	Contrôle du compteur rapide	72
3.55	Condition de remise à zéro du HSC	72
4.1	Ajout de l'interface	76
4.2	Connexion API	77
4.3	Paramètres des alarmes	77
4.4	Adresse Ethernet	78
4.5	Vue racine avant la configuration	78
4.6	Vue racine	79
4.7	Évènement "Presser" du bouton "Marche"	79
4.8	Évènement "Relâcher" du bouton "Marche"	80
4.9	Vue des consignes	81
4.10	Configuration du champ d'E/S de la consigne du faisceau 1	81
4.11	Configuration du champ d'E/S de la distance entre deux faisceaux	82
4.12	Vue de alarmes	82
4.13	Ajout du bouton d'acquiescement	83
4.14	Navigateur du projet	83
4.15	Vue globale	84
4.16	Table des alarmes IHM	85

Introduction générale

Aujourd'hui, l'industrie des machines électriques continue d'évoluer pour répondre aux besoins en constante évolution de notre société, en intégrant des technologies innovantes telles que l'intelligence artificielle, l'Internet des objets et l'automatisation des processus de production. Cette évolution permet de concevoir des machines plus efficaces, plus fiables et plus durables, tout en réduisant leur coût de production et leur impact environnemental.

La science de l'automatisation a une longue histoire qui remonte à plusieurs siècles. Dès la fin du XVIIIe siècle, les procédés de production ont été divisés en plusieurs étapes successives, ce qui a permis une augmentation de la production. En identifiant les tâches élémentaires qui composent tout cycle de production, il a été possible de construire des machines qui reproduisent les tâches correspondantes initialement réalisées par les travailleurs humains. Cela a ouvert la voie à la création de chaînes de montage et d'assemblage.

Ces méthodes ont été initiées dans l'industrie automobile, en particulier dans les usines de Ford aux États-Unis, avant d'être adoptées dans l'ensemble de l'industrie. Cela a permis à Ford d'accroître sa productivité.

L'introduction de l'informatique dans les processus de fabrication a considérablement accéléré le développement de l'automatisation. L'arrivée des ordinateurs et des machines à commande numérique a permis d'enregistrer les mouvements sur une unité de stockage, qui peut accomplir plusieurs opérations d'usinage différentes. Alors que les API étaient peu courants dans les années 1950, ils sont devenus omniprésents moins de cinquante ans plus tard, et peuvent accomplir des tâches complexes telles que le contrôle, la commande de machines, le traitement de données, la circulation de l'information et la simulation. Ils sont utilisés à tous les niveaux de production, de la conception à la production en passant par le contrôle. Bien que l'opérateur reste maître des décisions, il dispose à tout moment d'informations précises sur la production et les stocks.

Lorsqu'un ingénieur souhaite automatiser une machine, il doit examiner à la fois sa structure et son fonctionnement. L'aspect structurel concerne l'organisation et la combinaison des différents éléments qui composent la machine. L'aspect fonctionnel décrit les opérations que la machine doit effectuer ainsi que la manière dont les échanges entre les différents éléments et leur environnement ont lieu.

Les automates programmables ont une place de plus en plus importante dans les systèmes de commande automatique des machines, en remplaçant efficacement les systèmes en logique câblée tels que les relais ou les systèmes électroniques dans la plupart des applications industrielles. Les fonctions d'automatisation sont programmées, permettant ainsi une adaptation facile de l'application chargée dans la mémoire de l'automate aux conditions de fonctionnement de la machine.

Depuis sa création en 1983, l'entreprise Electro-Industries d'AZAZGA s'est spécialisée dans la fabrication et la vente de moteurs électriques, d'alternateurs, de transformateurs de distribution et de groupes électrogènes. À ses débuts, l'entreprise utilisait des machines à commande électromagnétique et de la logique câblée. Cependant, avec l'évolution de l'industrie, l'entreprise a progressivement investi dans des machines automatisées et automatisé ses machines existantes pour réduire les pannes fréquentes et améliorer sa productivité.

Dans ce cadre, les dirigeants de l'entreprise nous ont mandaté pour automatiser une machine bobineuse. Les exigences du projet impliquent le retrait de la logique câblée et des cartes électroniques et leur remplacement par un automate programmable, ainsi que l'installation d'un écran tactile pour remplacer le pupitre de manoeuvre actuel.

Pour réaliser cette tâche, on a divisé le plan de travail en 4 chapitres :

Chapitre 1 : on a étudié l'état actuel de la machine puis donné sa description.

Chapitre 2 : on a effectué une étude sur l'automate S7-1200.

Chapitre 3 : consacré à la programmation de la CPU.

Chapitre 4 : on a configuré l'écran tactile HMI.

Le travail sera terminé par une conclusion générale.

Chapitre 1

Description de la bobineuse

Introduction

On va commencer par présenter la machine bobineuse (état actuel), tout en donnant une vue générale sur l'entreprise où on a effectué le travail, ensuite on va élaboré les insuffisances de la machine et établir les objectifs à atteindre pour moderniser cette dernière.

1 Présentation de l'organe d'accueil

Electro-Industries est spécialisé dans la fabrication et la commercialisation des moteurs électriques, alternateurs, transformateurs de distribution et groupes électrogènes.

2 Présentation du complexe d'Electro-Industries

L'entreprise est divisée en trois unités :

- Unité de fabrication de moteurs électriques, alternateurs et groupes électrogènes.
- Unité de fabrication des transformateurs.
- Unité prestation technique.

1.2.1 Unité moteurs, alternateurs et groupes électrogènes

Sa capacité de production est de :

- 49500 moteurs de 0,25 à 400 kW.
- 2000 alternateurs de 16 à 180 kW.
- Montage de 400 groupes électrogènes.

Alternateurs synchrones

Les caractéristiques des alternateurs synchrones triphasés sont :

- Puissance : 17,5 à 200 kVA
- Tension : 380 V
- Fréquence : 50Hz

1.2.2 Unité de transformateurs

Cette entreprise fabrique des transformateurs de puissance allant de 50 à 2000 kVA, avec une capacité de production de 5000 unités par an.

Elle possède tous les équipements nécessaires pour la fabrication et les tests des produits finis.

Transformateurs de distribution

Les caractéristiques des transformateurs de distributions sont :

- Puissance : 50 à 2000 kVA
- Tension primaire : 5,5 -10-30 kV
- Tension secondaire : 400 V

3 Présentation des différents blocs constituant la machine

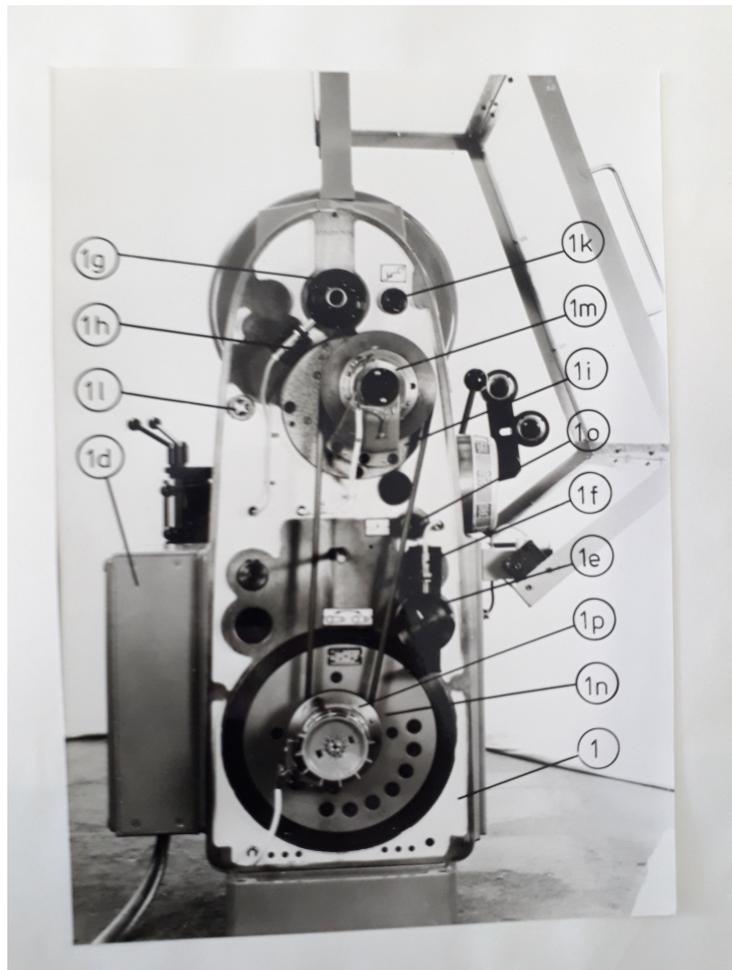
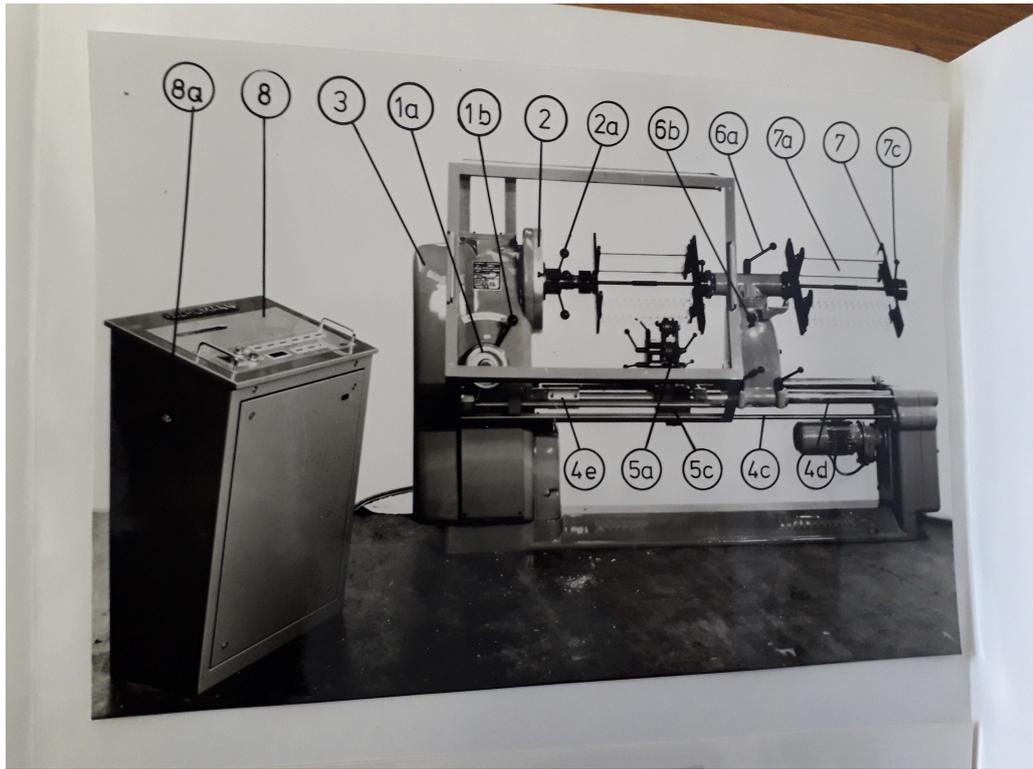
La machine bobineuse est utilisée dans l'industrie pour fabriquer des bobinages pour les machines tournantes telles que les machines à courant alternatif et à courant continu ainsi que les transformateurs. Le bobinage consiste à enrouler un fil de cuivre autour d'un circuit magnétique.

Elle se compose de différentes unités, on a :

- Pupitre de manoeuvre.
- Groupe de la machine.
- Plaque d'embase.
- Chariot-guide.
- Poupée revolver.
- Appareil bobineur.

4 Désignation des éléments de conduite et de réglage

La figure 1.1 qui se compose de quatre photographies représente la machine bobineuse et ses différents éléments de conduite et de réglage. Juste après on a donné la désignation de chaque symbole dans la figure 1.1 [1].



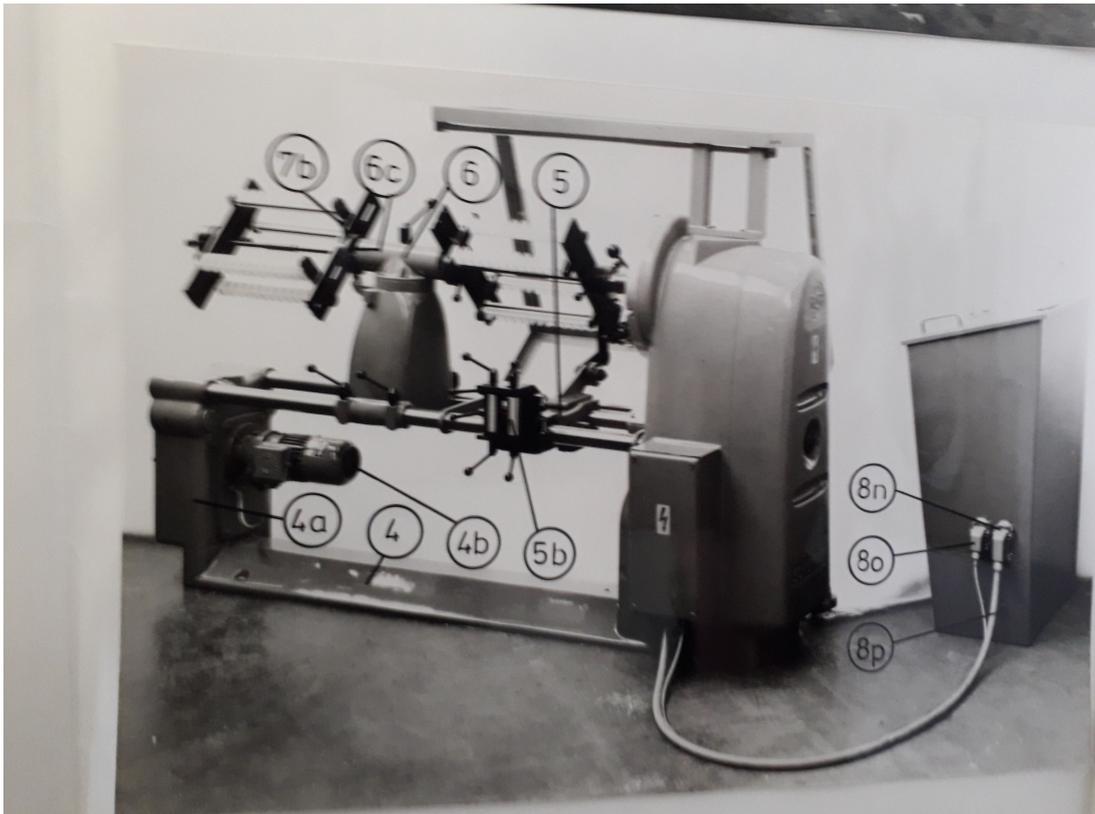


FIGURE 1.1 – Photographie de la machine [1].

- 1 : Groupe de la machine
- 1a : Volant à main de présélection des vitesses de rotation
- 1c : Levier d'enclenchement de présélection des vitesses
- 1b : Boîte de vitesse (rapport présélectionné)
- 1d : Armoire de distribution
- 1e : Disque d'impulsion (vis-mère)
- 1f : Initiateur (vis-mère)
- 1g : Poulie d'impulsion (arbre mené)
- 1h : Initiateur (arbre mené)
- 1i : Boîte de vitesse à engrenage à plusieurs rapports
- 1k : Remplissage d'huile
- 1l : Niveau d'huile
- 1m : Frein électromagnétique
- 1n : Moteur principal
- 1o : Vidange d'huile
- 1p : Accouplement électromagnétique
- 2 : Plateau
- 2a : Adaptateur rapide
- 3 : Capot de machine
- 4 : Plaque d'embase
- 4a : Bâti latéral
- 4b : Moteur d'entraînement de vis-mère
- 4c : Vis-mère
- 4d : Colonne
- 4e : Manchon de butée
- 5 : Chariot-guide
- 5a : Sortie de serrage de fil
- 5b : Entrée de fil
- 5c : Levier de recul du chariot-guide
- 6 : Poupée revolver
- 6a : Levier d'immobilisation
- 6b : Levier de déverrouillage
- 6c : Bride de baïonnette
- 7 : Appareil bobineur
- 7a : Gabarit de bobinage
- 7b : Levier de détente de l'axe de bobinage
- 7c : Levier de déverrouillage des joues
- 8 : Pupitre de manoeuvre

- 8a : Interrupteur principal
- 8b : Lampe témoin "Marche"
- 8c : Interrupteur de contrôle du fil (1 à 5)
- 8d : Lampe témoin "Machine Stop" (rupture de fil)
- 8h : Présélecteur du nombre de bobines par groupe
- 8i : Compteur d'impulsions (nombre de compartiments)
- 8k : Bouton-poussoir "Marche"
- 8l : Bouton-poussoir "Stop"
- 8n : Prise spéciale de force motrice
- 8o : Prise spéciale pour les lignes de signalisation
- 8p : Traversée de câble (branchement réseau)
- 8e : Commutateur de programmes, spires/Compartiments, vitesse de travail, tri décimal
- 8f : Commutateur de programmes, spires/compartiments, marche au ralenti
- 8g : Visualisation "État effectif", spires/compartiments
- 8m : Commutateur de programmes, changement décimal, tri décimal
- 8q : Lampe témoin, mémoire à séquence xPT72, postes 1 à 8

5 Caractéristiques techniques de la machine

Les caractéristiques techniques de la machine sont les suivants [1] :

- Vitesses de rotation sans harnais 26 à 560 tr/min progression 1,41, 9 échelons
- Couple moteur maximal à la plus faible vitesse de rotation sans harnais 25 mkg = 245 N.m
- Hauteur de pointes au-dessus du sol 950 mm
- Diamètre au plateau 400 mm
- Rainure en T du plateau T 12 DIN 650
- Charge utile au plateau (poids de la bobine + traction du fil) 400 kg maximal
- Diamètre maximal de bobine au-dessus des colonnes 800 mm
- Diamètre maximal de bobine au-dessus du chariot-guide 700 mm
- Hauteur de pointes au-dessus de la plaque d'embase 850 mm
- Charge utile des pointes (poids des bobines + traction du fil) 800 kg
- cône morse de la broche et de la contre-poupée
- Puissance du moteur et vitesse d'entraînement de la vis-mère 0.12 kW, 147 tr/min
- Paliers à l'avant de la broche (côté arbre mené-roulement à billes rigide AL 50)
- Logement arrière de la broche-roulement à billes rigides AL 35
- Présélecteur, 9 échelons, dans les deux sens de rotation
- Accouplement ortlinghaus 24 volts
- Frein ortlinghaus 24 volts
- Courroie trapézoïdale - machine sans harnais : 1 courroie 9.5 1475 (Conti-Ultraflex)
- Masse environ 500 kg (entre-pointes 1000 mm)
- "Saut" de compartiment maximale (3 chiffres)
- Présélection programmée de l'ordre d'utilisation des compartiments, réglable de 1 à 6

6 Description des éléments constitutifs

Le fonctionnement de chaque élément est le suivant [1] :

- Entraînement de la boîte à plusieurs rapports (1i) par le moteur (1n) avec interposition de l'embrayage (1p) et du frein (1m). L'arbre mené de la boîte de vitesses entraîne le plateau par l'intermédiaire de roues d'engrenages.
- Le moteur d'entraînement, l'embrayage et le frein sont commandés automatiquement par le pupitre de manoeuvre (8).
- Variation de vitesse à l'arrêt de la machine au moyen du volant à main du présélecteur (1a) et du levier de vitesse, qui doit être déplacé pour obtenir le rapport présélecté (1b).
- Enregistrement du nombre de spires par le compteur d'impulsions présélecteur du pupitre de manoeuvre (8).
- Enregistrement de la vis-mère par le moteur (4b) pour "les sauts décimaux".
- Commande du chariot-guide par la vis-mère (4c).
- Libre déplacement de la contre-poupée (6) sur les colonnes (4d).
- Déplacement manuel du chariot-guide en tirant sur le levier de recul (5c).
- Contrôle des impulsions de spires par le disque (1g) et l'initiateur (1h).
- Contrôle par le disque (1e) et l'initiateur (1f) des impulsions totalisées correspondant à la largeur des compartiments

7 Mise au point de la machine

Pour pouvoir mettre la machine en marche on suit les instructions suivantes [1] :

- Pré-sélectionner la vitesse de l'arbre mené.
- Monter et régler l'arrivée de fil (5b).
- Régler le galet-guide réglable de la sortie de fil (5a) en fonction du diamètre du fil ou de la largeur des fils parallèles.
- Amener à la main le Chariot-guide à sa position de départ.
- La position de départ du chariot-guide est déterminée par le manchon-butée (4e).
- Amener la poupée revolver (6) munie les appareils bobineurs contre l'accouplement rapide (2a) monté sur le plateau puis l'immobiliser en cette position.

8 Réglage de la commande

Après on effectue les réglage de la commande pour démarrer le bobinage comme suit [1] :

- Exemple : Nombre des bobines à confectionner pendant un cycle opératoire = 18, autrement dit 6 groupes de bobines de différentes largeurs. Nombre de spires effectif d'une bobine = 22 spires, première et troisième bobines d'un groupe et une spire en plus = 23 spires (indispensable en fin d'opération), nombre des compartiments du gabarit de bobinage 18, vitesse de bobinage 250 trs/min.
- Régler à 18 le compteur d'impulsions déterminant le nombre de compartiments (8i).

- Régler le commutateur de programme spires/compartiments (8e) :
Programme 1 - 2 - 3
20 19 20
Régler à 3 le commutateur de programme, spires/compartiments, marche au ralenti (5f), Régler à 3 le présélecteur de cycle programmé (3h).
- Relier les appareils de contrôle du fil munis de micro-interrupteurs au pupitre de manoeuvre pour l'arrêt automatique de la machine en cas de rupture de fil ou en fin de rouleau de fil.

9 Mise au point de la machine

Le déplacement du chariot-guide de compartiment à un autre est assuré par le moteur d'entraînement de la vis-mère (4b). Le "saut" d'un compartiment à l'autre doit être réglé en mm au commutateur de programme - "saut décimal" (8m) [1].

- Régler au compteur (8i) le nombre des compartiments du gabarit.
- Régler le galet de guidage réglable du dispositif de dévidement et de tension du fil (5a) en fonction du diamètre ou de la largeur des fils parallèles.
- Ajuster le chariot-guide muni du tendeur de fil pour le remplissage du premier compartiment de gabarit. Déplacer le manchon-butée (4e) contre le chariot-guide et l'immobiliser en cette position.
- La position angulaire de la base de la vis-mère (4c) peut être modifiée au moyen du disque d'impulsions (4e).
- La correction éventuelle de la position angulaire de l'appareil bobineur (7) peut se faire à l'aide du disque à impulsions (1g).
- Passer le fil sur les tendeurs (54) et (5b), puis le fixer à l'appareil bobineur.
- Le renvoi de fil au tendeur (5a) doit être tel que le fil à bobiner arrive par le bas entre le dernier rouleau de fil et le gabarit e bobinage. Cette disposition garantit une développante très régulière aux diverses bobines confectionnées fil par fil.

10 Fonction et mode de travail

Le fonctionnement et mode de travail de la machine est le suivant [1] :

- Fermer l'interrupteur principal (8e).
- Faire démarrer la machine en appuyant sur le bouton poussoir (8k).
- La machine peut être arrêtée à tout moment en appuyant sur le bouton poussoir (8l).
- La machine, une fois mise en marche, va au point de mise en attente d'arrêt réglé au sélecteur de programme (8e), puis elle passe la zone de marche lente dont le début est réglé au commutateur de programmes (8f). Le plateau est ensuite freiné et le moteur d'entraînement de la vis-mère (4b) assure le passage au compartiment suivant. La machine passe ensuite automatiquement à son régime normal et le cycle opératoire ci-dessus se répète un nombre de fois égal au nombre de compartiments réglé au compteur (8i).

- Les opérations manuelles subséquentes se déroulent comme suit :
 1. Effectuer la ligature de la dernière développante
 2. Sectionnement du fil
 5. Recul du chariot-guide (5)
 4. Libération du levier d'immobilisation (6a) de la poupée revolver (6)
 5. Actionnement du levier : déverrouillage (6b) et pivotement dans le sens des aiguilles d'une montre de l'appareil bobineur sans spires.
- Remettre à zéro le compteur d'impulsions pour 1 ., la détermination des compartiments (8i), fixer le fil et appuyer sur le bouton poussoir "Start" (8k) pour amorcer le bobinage suivant. On procédera en même temps au ligature de la développante suivante sur l'appareil bobineur dégagé de la machine.

11 Description de l'armoire électrique

L'armoire électrique renferme l'ensemble des dispositifs de protection et de contrôle des équipements électriques tels que les actionneurs, on va citer l'équipement qu'on va utiliser pour la nouvelle machine, car l'équipement de l'état actuel est très ancien et ne peut pas fonctionner correctement avec l'automate.

1.11.1 La protection électrique

Les organes de protection nécessaires sont :

- Sectionneurs.
- Fusibles.
- Disjoncteurs.
- Contacteurs.

1.11.1.1 Sectionneur

Le sectionneur est un appareil électromécanique permettant de séparer, de façon visible, un circuit électrique en aval de son alimentation et qui assure en position ouverte une distance de sectionnement satisfaisante électriquement.

Le principe mécanique est d'ouvrir un circuit électrique, afin d'isoler une partie de l'installation raccordée en aval du sectionneur.

En conséquence, le sectionneur, à la différence du disjoncteur, n'a pas de pouvoir de coupure, ni de fermeture. Il est impératif d'arrêter l'équipement aval pour éviter une ouverture en charge. Dans le cas contraire de graves brûlures pourraient être provoquées, liées à un arc électrique provoqué par l'ouverture.

Le sectionneur pour satisfaire aux normes en vigueur doit pouvoir être condamné en position ouverte [2].

1.11.1.2 Fusible

Les cartouches fusibles permettent d'ouvrir un circuit électrique par fusion d'un élément calibré pour couper le courant en cas de court-circuit ou de surcharges. La figure 1.2 représente la symbolisation d'un fusible.

Les fusibles assurent la protection des installations contre les courts-circuits afin d'éviter [3] :

- Les échauffements anormaux.
- Les arcs électriques.

1.11.1.3 Disjoncteur

Un disjoncteur est un dispositif de protection qui permet de protéger les circuits électriques contre les surcharges et les courts-circuits. Il est conçu pour interrompre automatiquement la circulation de l'électricité lorsqu'un courant trop important est détecté, afin d'éviter tout risque de surchauffe ou d'incendie.

Le disjoncteur est composé d'un interrupteur thermique et d'un mécanisme de déclenchement qui détecte la surcharge ou le court-circuit. Lorsqu'un courant trop important circule dans le circuit électrique, l'interrupteur thermique se déclenche et coupe l'alimentation électrique. Le disjoncteur peut être réarmé manuellement ou automatiquement après la correction de la cause de la surcharge ou du court-circuit.

1.11.1.4 Contacteur

Un contacteur est un organe électromécanique qui permet de commuter un circuit électrique en ouvrant ou fermant un ou plusieurs contacts électriques. Il est souvent utilisé pour la commande à distance de machines ou d'appareils électriques.

Le contacteur se compose d'une bobine électromagnétique et d'un jeu de contacts électriques. Lorsque la bobine est alimentée en courant électrique, elle crée un champ magnétique qui attire un noyau mobile, qui permet d'ouvrir ou de fermer les contacts électriques. Les contacts peuvent être utilisés pour commuter des circuits de puissance ou de commande, en fonction de l'application.

Les contacteurs sont souvent utilisés dans les installations électriques industrielles et commerciales, où ils permettent de contrôler la mise en marche ou l'arrêt de grandes machines ou équipements électriques, tels que les moteurs électriques de la bobineuse ou les compresseurs. Ils sont également utilisés dans les systèmes de climatisation et de chauffage, les systèmes d'éclairage public, les systèmes de traitement de l'eau et de nombreux autres domaines d'application électrique.

La figure 1.2 donne un schéma du circuit de puissance et de commande ainsi que la bobine, qui sont présent à l'intérieur du contacteur, avec le modèle physique de ce dernier.

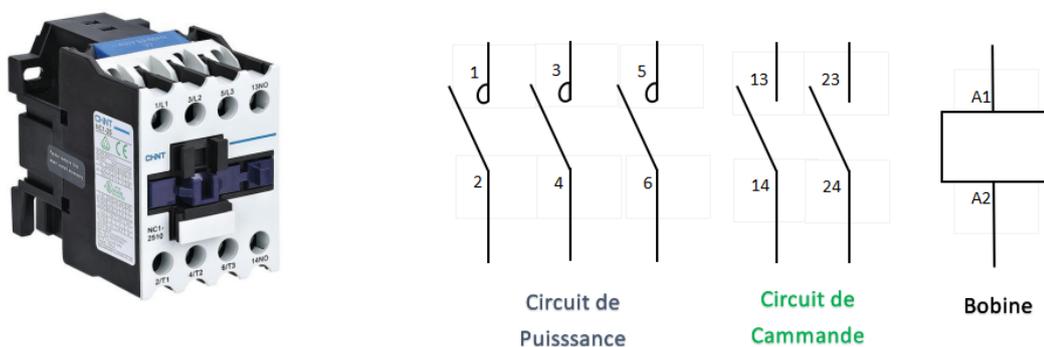


FIGURE 1.2 – Contacteur [4]

12 Le transformateur d'isolement

Un transformateur galvanique est un type de transformateur électrique conçu pour isoler électriquement un circuit de l'autre, afin de protéger les personnes et les équipements contre les chocs électriques et les interférences électromagnétiques.

Le transformateur galvanique est constitué de deux enroulements, un enroulement primaire et un enroulement secondaire, qui sont complètement isolés l'un de l'autre électriquement. Le courant électrique circule dans l'enroulement primaire, créant un champ magnétique qui induit un courant électrique dans l'enroulement secondaire. Les deux circuits sont isolés électriquement les uns des autres, ce qui permet d'éliminer tout risque de choc électrique ou de dommage électrique.

On a utilisé ce transformateur dans cette machine pour assurer l'isolement de la partie puissance de la partie commande.

13 Le variateur de vitesse

Un variateur de vitesse est un dispositif utilisé pour contrôler la vitesse de rotation d'un moteur électrique. Il permet de réguler la vitesse d'un moteur en ajustant la fréquence et la tension de l'alimentation électrique.

Le principe de fonctionnement d'un variateur de vitesse repose sur la conversion de l'alimentation électrique alternative (généralement en courant alternatif) en une alimentation à fréquence variable pour le moteur. Cela permet de faire varier la vitesse de rotation du moteur de manière continue.

Voici les étapes générales du fonctionnement d'un variateur de vitesse :

- Redressement : L'alimentation électrique alternative est généralement convertie en courant continu par un pont redresseur.
- Filtre : le courant continu est ensuite filtré pour éliminer les ondulations indésirables et obtenir une tension continue plus lisse.
- Conversion : la tension continue est ensuite convertie en une tension alternative de fréquence variable, à l'aide d'un onduleur MLI.
- Commande : le variateur de vitesse ajuste la fréquence et la tension de sortie à l'aide de circuits électroniques de commande qui reçoivent des signaux de référence et effectuent les ajustements nécessaires.
- Alimentation du moteur : La tension alternative de fréquence variable est fournie au moteur électrique. La variation de la fréquence de l'alimentation modifie la vitesse de rotation du moteur en fonction des caractéristiques du moteur et des paramètres de commande du variateur.

Ce dispositif représenté dans la figure 1.3, aidera à remplacer le moteur de bobinage qui fonctionne en deux vitesses à l'aide de son montage spécial (montage Dahlander), avec un moteur asynchrone standard qui est fabriqué par l'entreprise, donc la disponibilité contrairement au moteur Dahlander qui est beaucoup plus moins disponible, on peut brancher les sorties de l'automate qui commande le moteur directement avec les entrées à vitesse pré-sélectionnée du variateur.



FIGURE 1.3 – Variateur de vitesse

14 Encodeur rotatif

Sur de nombreux systèmes automatisés, le contrôle de la position et du déplacement d'un objet mobile est un problème commun. Pour y remédier, il existe plusieurs types de capteurs disponibles :

- Les capteurs simples.
- Les encodeurs rotatifs.

Un encodeur rotatif est un dispositif de mesure de mouvement qui permet de convertir le mouvement rotatif en un signal numérique ou analogique. L'encodeur rotatif se compose d'un disque rotatif et d'un capteur optique ou magnétique qui détecte le mouvement du disque.

Le disque rotatif est doté d'une série de marques ou de fentes régulières qui sont détectées par le capteur optique ou magnétique. Lorsque le disque tourne, les marques ou les fentes passent devant le capteur, ce qui génère une série d'impulsions électriques qui sont converties en signaux numériques ou analogiques.

Les encodeurs rotatifs comme celui de la figure 1.4, sont utilisés dans une variété d'applications, notamment dans les machines-outils, les équipements médicaux, les instruments de mesure, les systèmes de positionnement, les robots industriels, les équipements d'automatisation et les systèmes de contrôle de mouvement. Ils permettent de mesurer avec précision la vitesse, la position, la direction et l'angle de rotation des axes de mouvement, et de les convertir en signaux électriques pour une utilisation dans les systèmes de contrôle et d'automatisation.

L'encodeur utilisé dans notre machine est relié à l'arbre de rotation d'un moteur triphasé. Lorsque le moteur tourne, l'encodeur génère des impulsions qui sont ensuite traitées par des cartes électroniques connectées à l'encodeur.

L'information indispensable dans notre cas est la distance parcouru par le chariot-guide plus le nombre de spires produit, l'encodeur est l'appareil qui sert à fournir ce genre d'information d'après ce qu'on a expliqué précédemment.



FIGURE 1.4 – Encodeur rotatif [5]

15 Capteur de proximité inductif

Les capteurs de proximité inductifs permettent de détecter tout objet métallique qui se trouve à proximité de la tête de détection.

Un capteur inductif se compose essentiellement d'un oscillateur comme indique la figure 1.5.

Le champ électromagnétique est créé à l'avant de la face sensible.

Toute pièce métallique pénétrant dans ce champ devient le siège de courants de Foucault qui provoquent l'arrêt des oscillations.

C'est donc l'arrêt des oscillations qui est détecté [1].

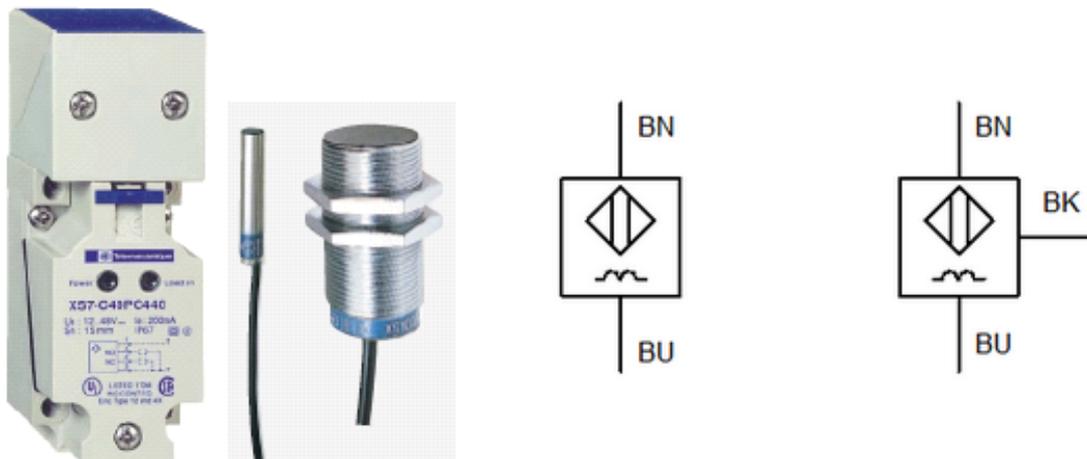


FIGURE 1.5 – Capteur à proximité inductif [6]

16 Problématique

Après une étude approfondie sur le fonctionnement et la commande de la bobineuse (état actuel), on a pu détecter plusieurs inconvénients :

- Pannes assez fréquentes des dispositifs de commande et leur non disponibilité sur le marché.
- La logique câblée est une technique figée, la modification est quasiment impossible.
- Le nombre de spires qu'elle peut produire est limité.
- Difficulté de maintenance en cas de pannes.

Pour remédier aux insuffisances et aux problèmes précédemment rencontrés, nous avons décidé d'automatiser la bobineuse à l'aide d'un automate programmable industriel (API S7-1200) et d'une commande à écran tactile KTP400, Nous avons fixé les objectifs suivants pour améliorer le suivi du fonctionnement de la machine :

17 Objectifs

- Sur le groupe machine on va enlever la commande manuel de la vitesse du moteur (la commande sera automatique).
- Toute Logique câblée trouvant sur l'armoire électrique, même les systèmes de comptage et de commutation, seront remplacé par l'automate, l'encodeur rotatif et les appareils de protection modernes nécessaires.
- L'automate va contrôler les bobines des contacteurs de manière à démarrer et arrêter les moteurs triphasés.
- La bobineuse peut faire un bobinage à trois faisceaux, qui se répète une deuxième fois par l'intermédiaire d'un passage à une spire c'est à dire deux groupes de bobinage, c'est un bobinage à sept faisceaux. Ou bien faire un bobinage simple à quatre faisceaux, c'est à dire un groupe de bobinage.
- On va remplacer le pupitre de manoeuvre par l'interface homme-machine de type KTP400, pour envoyer les consignes vers l'automate et superviser l'avancement de l'opération à tout moment, plus l'indication des alarmes en cas de défaut

Conclusion

Dans ce chapitre, nous avons donné une description de la bobineuse ainsi que son fonctionnement. À partir de là, nous avons formulé les principales insuffisances de la machine. Ces insuffisances ont motivé l'établissement de plusieurs objectifs, qui seront détaillés dans les prochains chapitres. Dans le chapitre suivant on va présenter la CPU qu'on a utilisé.

Chapitre 2

Description de l'automate S7-1200 CPU 1214C

Introduction

L'Automate Programmable Industriel (API) est devenu le composant fondamental de la plupart des systèmes d'automatisation, depuis son apparition dans les années 1970. Initialement développé à la demande des constructeurs automobiles pour améliorer leur productivité tout en réduisant les coûts, l'API a depuis été adopté dans tous les secteurs industriels grâce à ses nombreux avantages. Bien que les API OMRON, ALLEN BRADLEY, ABB, SIEMENS, SCHNEIDER... puissent différer en termes de puissance de travail, de dimensions physiques, de programmation et de prix, leur structure de base et leur objectif général sont essentiellement similaires. Toutefois, ces critères évoluent rapidement avec le temps et les besoins de la fabrication.

Dans ce chapitre, on va découvrir l'architecture et les composants fondamentaux des Automates Programmables Industriels (API), ainsi que se familiariser avec l'automate de marque SIEMENS, en étudiant sa structure globale, ses règles de fonctionnement et de configuration de l'unité centrale.

L'automate est un dispositif électronique programmable conçu pour fonctionner dans des environnements industriels. Il est utilisé pour automatiser les processus en assurant la commande des contacteurs à partir d'informations logiques, analogiques ou numériques.

1 Architecture des automates

La meilleure façon de concrétiser la définition de l'API est d'examiner les éléments matériels et logiciels qu'il doit comporter, et la manière dont ils s'agrègent pour constituer un ensemble opérationnel.

Quel que soit le type d'automate – compact (monobloc), modulaire, c'est-à-dire formé de modules fonctionnels en boîtiers enfichables sur un châssis spécifique, ou encore en cartes à monter dans des supports standard (rack 19 pouces) – il comporte les éléments détaillés ci-après dans la figure 2.1 [7].

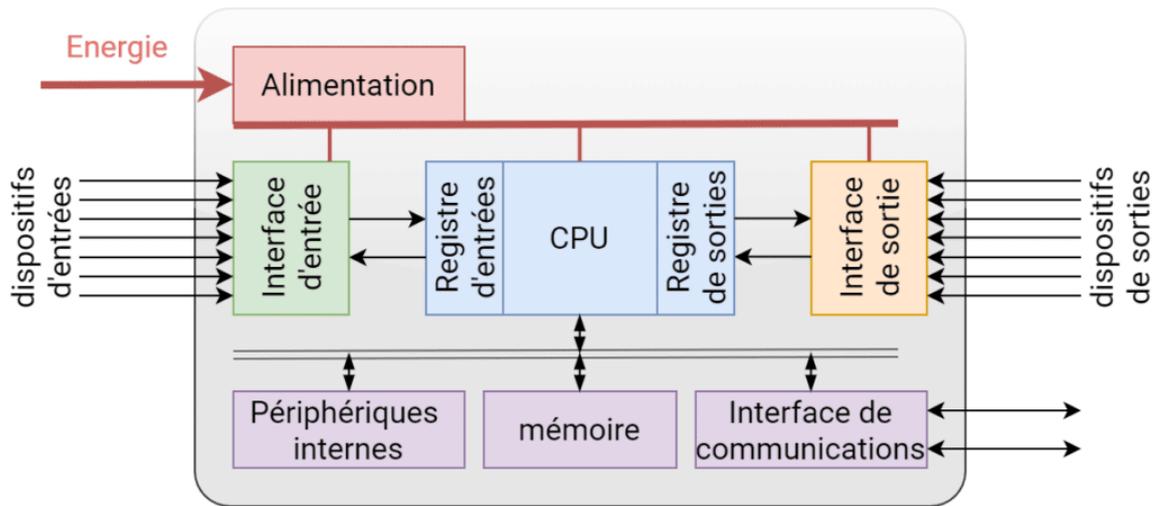


FIGURE 2.1 – Architecture d'un API [8]

2.1.1 Unité centrale

Fondamentalement un automate est un système numérique. Il comprend donc une unité centrale bâtie autour d'un processeur. Ce dernier peut être spécifique ; c'est le plus souvent aujourd'hui un processeur de PC. La course aux performances globales présente peu d'intérêt ici, et il peut s'agir d'un composant considéré comme obsolète pour des applications imagerie, voire bureautique. L'automate PLC102 de Wöhrlé comporte ainsi deux processeurs 80386 d'Intel, un pour l'unité centrale, l'autre pour l'interface homme/machine (IHM) contenue dans le même boîtier. Ce n'est pas là le dernier cri en matière de processeurs.

Le processeur proprement dit est donc un circuit de conception ancienne, éprouvé et bon marché. L'amélioration des performances en temps d'exécution viendra plutôt des circuits d'interfaçage, les coupleurs, gérant les échanges selon les différents modes, de l'incorporation de micro-contrôleurs et processeurs spécialisés, voire du remplacement partiel par des circuits spécifiques (Application specific integrated circuit : ASIC) si le facteur temps est primordial [7].

2.1.1.1 Unité de contrôle

Elle recherche les instructions en mémoire, les décode et génère les signaux de commande pour les exécuter.

2.1.1.2 Unité arithmétique et logique

C'est la partie responsable à effectuer les opérations mathématiques telles que l'addition, la soustraction et la multiplication, et les opérations logiques sur bit telles que : AND, OR et XOR.

2.1.1.3 Registres

Le compteur programme, également appelé compteur ordinal, est l'un des registres les plus importants. Il contient en permanence l'adresse de l'instruction en cours d'exécution ou celle de l'instruction suivante pour accélérer la recherche. Son évolution est automatique : il est incrémenté de +1 à la fin de chaque instruction, sauf dans le cas d'une instruction de saut où l'adresse de la nouvelle instruction lui est imposée. Le registre d'instruction est utilisé pour stocker les codes des instructions une fois qu'elles ont été décodées. L'accumulateur est utilisé dans toutes les opérations arithmétiques et logiques. Comme son nom l'indique, il accumule les résultats de ces opérations.

2.1.2 Mémoires

Elles sont étroitement associées à l'unité centrale.

- Le système d'exploitation est contenu dans une mémoire morte (Read only memory : ROM), que le processeur ne peut que lire ;
- Une mémoire vive (Random access memory : RAM) contient programme et données de l'application, elle est sauvegardée par batterie pour protéger ces données lors de coupures de l'alimentation électrique.

Le programme, une fois au point, est stocké dans une mémoire reprogrammable avec un matériel adapté (EPROM), automatiquement transférée dans la RAM après une coupure secteur. La taille de cette mémoire, exprimée en kilo-octets (1 024 groupes de 8 bits), est faible par rapport à celles rencontrées en bureautique, même si elle s'est nettement accrue suite au développement de la communication numérique et de ses besoins. L'utilisation de mémoires flash de 1 024 kilo-octets ou davantage, qui conservent leurs données en cas de perte de l'alimentation électrique, facilite la sauvegarde.

La taille des mémoires programme et données n'est qu'une indication de la capacité de l'API dans ce domaine. Le système d'exploitation impose aussi des limites par type de langage pour le programme, de variables pour les données : la mémoire d'un micro-automate pourra ainsi recevoir au plus 256 mots variables de 16 bits, 64 mots constants, 128 bits internes [7].

2.1.3 Entrées et sorties

Un automate ne se conçoit qu'en liaison avec des grandeurs physiques caractérisant le comportement de la Partie Opératrice, c'est-à-dire le substrat du système de production.

Au minimum on aura des E/S tout-ou-rien (TOR), aussi appelées logiques, binaires, numériques, digitales [7].

2.1.3.1 En modules d'entrées ou de sorties

Ils regroupent des voies de liaison filaire de mêmes caractéristiques physiques, conformes à la norme NF 61131, Partie 2, dans un boîtier, souvent selon un nombre puissance de 2 pour faciliter l'adressage, le plus courant étant 16. Il faut noter la nécessité de les protéger contre les nuisances de l'environnement industriel : parasites, surtensions, court-circuits, etc., et la variabilité des standards de liaison électrique. La figure 2.2 montre un mode usuel de protection des entrées, avec optocouplage et circuits de conditionnement électrique assurant filtrage et limitation du courant pour assurer une tension V_e adéquate entre la borne d'entrée et la masse [7].

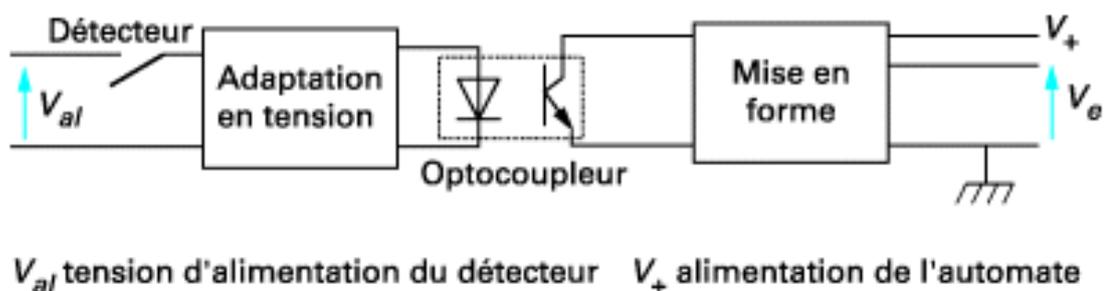


FIGURE 2.2 – Circuit d'entrée TOR d'API avec protection [7]

Pour les sorties, protégées contre les surcharges et l'inversion de polarité, certaines sont à relais statiques, d'autres à relais électromagnétiques :

- modules à sorties statiques : relais statiques intégrant des composants spécialisés (transistors bipolaires, IGBT (Insulated gate bipolar transistor), thyristors, triacs. Ils ne subissent pas d'usure mécanique et leurs caractéristiques de commutation se maintiennent dans le temps) ;
- modules à relais électromagnétiques, dits simplement sorties relais. Ils assurent le découplage entre signaux de commande et de travail grâce à l'existence de deux circuits électriques : circuit d'excitation (bobine) et circuit de puissance (contacts). D'une durée de vie plus limitée que les relais statiques (moins de 100 000 cycles pour des contacts soumis à 10 A sous 125 V alternatif) et plus lents, les relais électromagnétiques ont aussi des avantages : faible résistance de contact, faible capacité de sortie, faible coût. Il faut leur adjoindre un circuit anti-rebond pour obtenir une commutation unique lors du changement d'état de la sortie correspondante.

2.1.3.2 Intégrées

C'est le cas des automates compacts type PS4 de Moeller ou TSX07 de Schneider. Les caractéristiques des E/S sont alors choisies parmi les standards les plus répandus (entrées 24 V isolées, sorties transistorisées à alimentation continue, ou entrées 110 V alternatif isolées, sorties relais, par exemple). Le nombre de voies des modules intégrés reflète le rapport moyen entre entrées et sorties dans les problèmes industriels, qui se traduit par un déséquilibre au profit des entrées. Le micro-automate programmable NAIS FP-e de Matsushita possède ainsi huit entrées et six sorties TOR (cinq transistorisées et une à relais). La possibilité de configurer des voies d'accès en entrée ou en sortie est rarement utilisée, pour des raisons de sécurité. La distinction modulaire/compact est en partie gommée par l'existence, sur plusieurs API monobloc, de modules d'extension renforçant si besoin la faible configuration initiale [7].

2.1.3.3 Déportées

Cette dernière solution est en progrès car elle minimise le câblage. Le bloc d'E/S doit comporter un processeur et communiquer avec le processeur central, grâce à l'un des modules de communication. Les modules d'E/S sans fil se rattachent à cette catégorie.

Les modules d'E/S possèdent un voyant (diode électroluminescente) par voie ; cette visualisation est fort utile pour la mise au point et la maintenance, même s'il existe aujourd'hui d'autres outils de diagnostic, et participe à la sûreté de fonctionnement de l'API. Il ne faut pas oublier par contre l'existence d'un commun électrique entre les différentes voies, au moins par groupe de quatre par exemple.

Les API ayant besoin de mesures, pour le contrôle et/ou la régulation, disposent d'E/S analogiques. Elles sont groupées par quatre, quelquefois huit, avec, là encore, une grande variété de gammes de liaison électriques :

- En tension 0-5 V, 1-5 V, 0-10 V, ± 5 V, ± 10 V ;
- En courant 0-20 mA, 4-20 mA, ± 20 mA.

Les plus répandues sont 0-10 V et 4-20 mA, ce dernier standard assure une transmission plus fiable que les liaisons en tension, et fournit une détection immédiate d'une rupture de la liaison ou du capteur.

Il existe encore d'autres types d'E/S, notamment des entrées adaptées à des capteurs particuliers (sondes à résistances de platine PT100, thermocouples, jauges de contrainte), à associer à des fonctions métier. On les trouve même parfois sur des API monoblocs : le NAIS FP-e déjà cité a ainsi deux entrées thermocouple [7].

2.1.4 Modules de communication

Un API a aujourd'hui plusieurs possibilités de communication numérique [7] :

- Par le port série du processeur (liaison RS232 ou RS485);
- Par un port USB (Universal serial bus);
- Par un coupleur Ethernet;
- Par d'autres canaux : CAN (Controller area network), AS-I (Actuator sensor interface), etc. Dont le coupleur est quelquefois intégré.

2.1.5 Alimentation électrique

À partir du secteur, plus rarement d'une source de 24 V continu, l'alimentation principale fournit l'énergie nécessaire à l'UC et aux interfaces (sous des tensions de 5, 12, 15, 24 V continu). Elle ne peut que rarement alimenter des sorties, et pas toujours les entrées. Elle doit faire face aux micro-coupures du réseau. Un onduleur évite les risques de coupure si celles-ci risquent de dépasser les tolérances admises, 10 ms au plus à 50 Hz avec au moins 1 seconde entre deux coupures.

L'alimentation fait l'objet d'une protection avec notamment un disjoncteur magnéto-thermique. Il est parfois nécessaire, pour lutter contre les perturbations électriques du secteur, d'introduire un transformateur d'isolement. C'est notamment le cas pour les raccordements à un réseau électrique à « neutre flottant ».

Il ne faut pas oublier que les châssis d'extension, les E/S déportées, voire certaines E/S directes, exigent aussi une alimentation. Son dimensionnement doit prendre en compte les ajouts possibles de modules et/ou de liaisons fil à fil directes lors de modifications ultérieures de l'installation [7].

2.1.6 Éléments d'interface homme-machine

Il s'agit de voyants, témoins de l'état du processeur (sous tension, programme en cours d'exécution, en défaut), de la charge de la batterie, et parfois d'un poussoir de réinitialisation (reset). Il n'y a pas en revanche d'interrupteur marche/arrêt [7].

2 Fonctions de base

Que sait faire un automate? La réponse dépend de nombreux facteurs, dont sa taille et son constructeur, mais la base minimale caractéristique comprend les éléments énumérés ci-après.

2.2.1 Fonctions logiques ou booléennes

On les appelle aussi fonctions binaires ou sur bits, leurs variables correspondant à 1 bit (BInary digiT). Elles mettent en jeu des variables binaires, dites aussi discrètes, booléennes, ou tout-ou-rien (TOR), à deux états notés 0 et 1. Les fonctions ET (AND), OU (OR), NON (NOT) sont indispensables pour réaliser d'autres fonctions plus complexes. ET et OU sont généralisables à plus de deux entrées. On rencontre souvent NON ET (NAND), NON OU (NOR), DILEMME (EXCLUSIVE OR).

L'API possède également une fonction mémoire, quasi-indispensable pour le traitement des structures séquentielles comme représente la figure 2.3, il s'agit d'une fonction logique dont le résultat est mémorisé : une entrée (SET) met à 1 la sortie qui reste à 1 même si l'entrée revient à 0; une autre entrée (RESET) remet de même la sortie à 0. Il faut prêter attention au comportement de la sortie lorsque les deux entrées sont à 1, ce qui peut survenir transitoirement ou suite à une erreur ou

défaillance. Attention aussi au fait que selon la norme NF 61131-3, ce n'est pas une fonction mais un bloc fonctionnel [7].

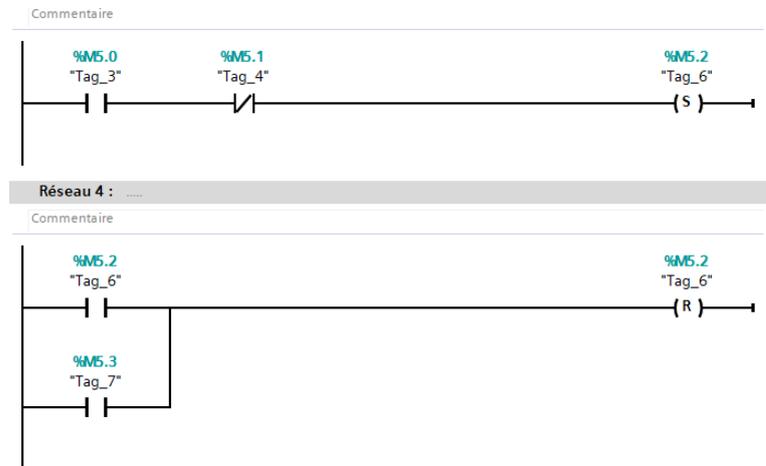


FIGURE 2.3 – Exemple d'opérations logiques sur TIA Portal

2.2.2 Fonctions arithmétiques et sur mots

Pour qu'un API effectue du traitement arithmétique (calcul de proportions par exemple), de la régulation, et même du comptage, qu'il envoie un message, il doit travailler sur un groupe de bits. Nombres entiers et mots standard s'expriment sur 16 bits. Il existe donc des fonctions sur mot (lecture, écriture, comparaison, etc.), et des fonctions de calcul qui opèrent sur 16, 32 ou 64 bits, la figure 2.4 montre un exemple de ces fonctions.

La bibliothèque de fonctions arithmétiques et sur mots est une des caractéristiques de la capacité de traitement d'un API, ou de son orientation vers un type d'activité donné : régulation, télésurveillance de locaux, etc [7].



FIGURE 2.4 – Exemple d'addition et soustraction sur le TIA Portal

2.2.3 Fonctions de communication

Les fonctions de communication (émission, réception, etc.) selon un protocole donné font désormais partie du bagage minimum de l'API. Certains modèles possèdent même plusieurs jeux pour différents modes d'échange [7].

2.2.4 Autres fonctions

Comptage et temporisation méritent d'être mentionnés explicitement car ils occupent une place importante dans l'univers de l'API. Les compteurs permettent de mémoriser l'occurrence d'événements : ce sont les changements d'état d'une entrée TOR ou d'une variable binaire interne, dénommés front montant pour le passage de 0 à 1, front descendant pour celui de 1 à 0. Outre ses applications directes (détermination d'un nombre de pièces, de passages, de défauts, etc.) le comptage s'emploie dans certaines fonctions métier, par exemple en positionnement. Toujours numériques, les temporisations s'expriment en fonction d'une base de temps choisie dans une gamme de valeurs : 1 s, 0,1 s, 0,01 s par exemple ; elles réalisent un comptage d'événements internes particuliers, les impulsions de la base de temps. La présélection définit le nombre d'impulsions attendues. On peut ainsi retarder le début ou la fin d'une action.

Un API dispose aussi d'un saut avant, c'est-à-dire de la possibilité de court-circuiter une partie de programme sur test d'une condition vraie ou fausse, mais pas nécessairement d'un saut arrière, qui risque d'allonger le temps d'exécution et va à l'encontre des exigences du temps réel.

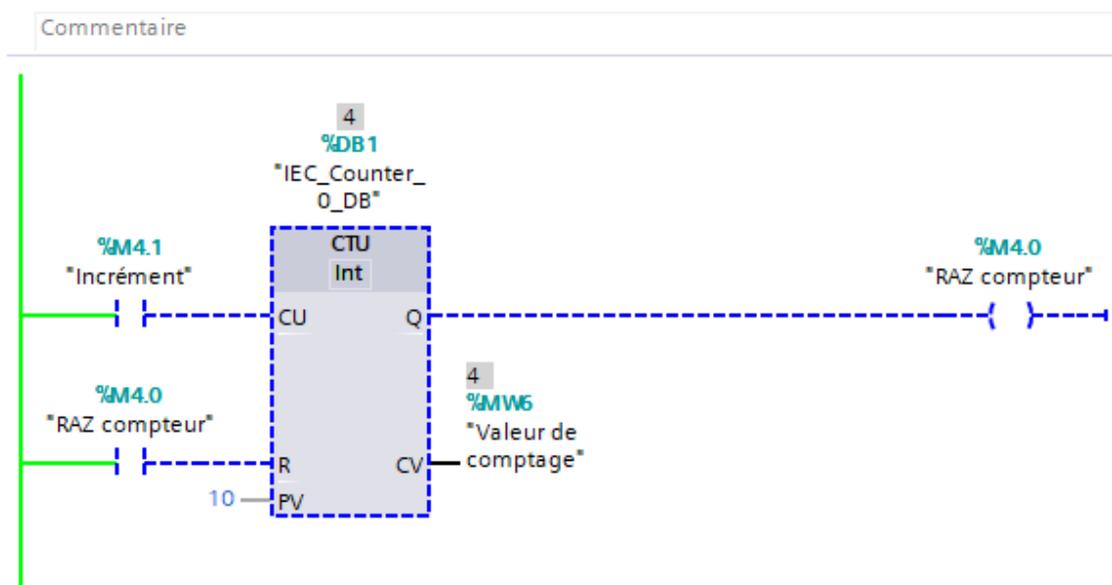


FIGURE 2.5 – Compteur incrémental

La figure 2.5 représente un exemple d'un compteur à 10, il commence de la valeur 0, à chaque fois le bit %M4.0 s'active, le compteur s'incrémente et on peut voir la valeur de comptage au dessus du bloc, quand le comptage atteint la valeur 10 la sortie du compteur s'active, et le bit %M4.1 va réinitialiser le compteur [7].

3 Langages

Ils assurent la mise en œuvre des fonctions précédentes, ainsi que de toutes les fonctions spécialisées. L'une des particularités de l'API est l'utilisation de langages spécifiques, qui en permettent la programmation par des personnes issues d'horizons techniques différents, avec pour certains un apprentissage extrêmement rapide.

Ces langages font l'objet d'une norme internationale, en définissant les types et leurs caractéristiques, et cette multiplicité encadrée a sans doute facilité le succès de l'automate. Il faut toutefois noter que la norme n'impose pas la portabilité des langages, qui restent propriétaires, et qu'un API ne dispose pas forcément d'un langage de chaque type.

La norme CEI 61131-3 qu'on a suit dans ce projet, est défini d'une manière générale :

- La symbolique des variables : débutant toujours par « % », elle comporte une lettre définissant le type (« I ou E » pour une entrée, « Q ou A » pour une sortie, « M » pour une mémoire interne), une autre la taille (« X » pour 1 bit, « B » pour 8, « W » pour 16, « D » pour 32.);
- Le graphisme, car plusieurs de ces langages ont une forme graphique ou semi-graphique, il est très simple car initialement réalisable avec une machine à écrire ;
- Les « éléments communs » des programmes tels que commentaires, déclarations, date, etc.

Elle décompose le logiciel en trois entités, présentées ci-dessous dans un ordre ascendant [7] :

- La fonction (par exemple le ET booléen, la comparaison de valeurs numériques), à sortie unique ne dépendant que de ses entrées ;
- Le bloc fonction, composant logiciel réutilisable dans tous les langages (par exemple un compteur), qui peut comporter plusieurs sorties et des variables internes influant sur les sorties, comme dans les fonctions mémoire ;
- Le programme.

2.3.1 Langage à contacts (ladder diagram : LD)

C'est le langage qu'on a utilisé dans ce travail pour établir le programme de l'automate Il traduit la vocation première de l'automate programmable, qui était le remplacement des volumineuses armoires à relais caractéristiques des premiers temps de l'automatisation. Il s'adressait donc plutôt à des électriciens mais sa grande simplicité l'a rendu très populaire au delà d'une corporation et il l'est resté, notamment aux États-Unis.

C'est une adaptation des schémas électriques. L'application à réaliser se représente par des réseaux, c'est-à-dire un ensemble de contacts et de bobines reliés aux bornes d'une source de tension électrique ; les bornes sont matérialisées par deux traits verticaux aux extrémités du schéma, les contacts par ou suivant qu'ils sont normalement ouverts (NO), ou normalement fermés (NF), donc passants au repos. Un contact NO est fermé lorsque la variable booléenne associée vaut 1. Le résultat est affecté à une bobine, positivement ou négativement, suivant que cette bobine doit être alimentée ou non. Le flux d'information circule de gauche à droite.

Ainsi le réseau de la figure 2.6 traduit en langage ladder le fait que la sortie physique repérée %Q0.0 doit être mise à 1 si, et seulement si, l'entrée %I0.0 est à 1 (le contact NO est alors fermé) et l'entrée %I0.1 à 0 (contact NF). La mise en série de contacts équivaut donc à un ET, la mise en parallèle à un OU.

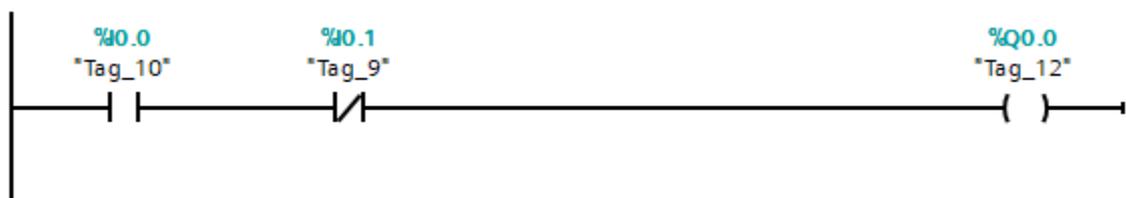


FIGURE 2.6 – Réseau représentant la fonction logique ET

Des extensions ont été apportées pour lancer des traitements arithmétiques, des temporisations, du comptage, en incorporant des blocs fonction au réseau, ou exécuter des opérations informatiques (sauts). Ainsi le réseau de la figure 2.7 représente un saut conditionnel vers un autre réseau étiqueté LABELB [7].

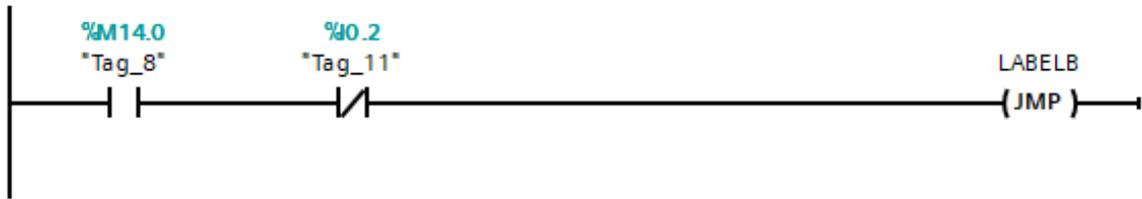


FIGURE 2.7 – Réseau utilisant le saut

4 Exécution du programme

Le fonctionnement de l'automate pour réaliser l'application repose sur l'exécution permanente, certains parlent d'une boucle infinie, d'un cycle opératoire à quatre phases, dont trois directement utiles au résultat :

- L'acquisition de la valeur des entrées (lecture);
- Le traitement des données selon le programme écrit dans un ou plusieurs des langages du § 3 ;
- L'affectation de la valeur des sorties (écriture).

La quatrième phase, correspondant à des traitements et diagnostics internes, ralentit un peu la cadence mais constitue l'un des maillons essentiels de la sûreté de fonctionnement propre de l'API. Différentes modalités d'organisation du cycle sont présentées à la figure 2.8.

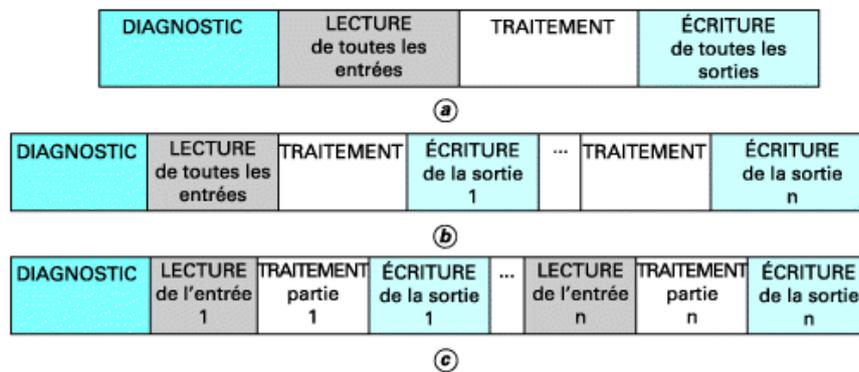


FIGURE 2.8 – Ordonnancement de l'exécution [7]

- Dans le cas (a), cas courant où l'application ne comprend qu'une seule « tâche », le résultat de la scrutation voie par voie des entrées est rangé dans une mémoire tampon, puis transféré en bloc à l'UC. De même les sorties calculées sont mémorisées avant leur transfert aux interfaces de sortie. On parle de synchronisme par rapport aux entrées et aux sorties, les changements de celles-ci s'opérant en même temps, et leurs valeurs restant constantes pendant le traitement.
- Les cas (b) et (c) décrivent d'autres possibilités, d'une gestion plus délicate car une même variable peut prendre différentes valeurs en différents points du programme, mais permettant de raccourcir le temps de réaction de l'automate à un événement extérieur, matérialisé par le changement de valeur d'une variable TOR.

En raison des tests et des sauts avant, la durée du traitement, appelé temps d'exécution, peut varier d'un cycle à l'autre. Son évaluation, à partir d'une vitesse d'exécution exprimée en kilomots de mémoire programme par seconde, ou du temps d'exécution d'une instruction logique (en μ s), ne peut donc être qu'approximative.

La durée totale des quatre phases de la figure 2.8 constitue le temps de cycle, elle est caractéristique du programme dans l'optique du temps réel et varie dans une large fourchette, de l'ordre de 1 à 20 ms. Le mode de fonctionnement fait qu'un changement de valeur d'une entrée pendant moins d'un temps de cycle risque d'être ignoré, d'où l'existence de « mémoires d'événements » sur certaines entrées TOR.

Des délais de réaction plus brefs sont parfois exigés pour certaines parties d'application, des commandes de mouvement par exemple. Il faut alors fonctionner par interruption. Un événement atteignant directement le processeur provoque la suspension de l'exécution « normale » pour lancer un programme spécifique. Une autre solution consiste à utiliser des modules spécialisés, tels les modules de comptage rapide, qui, intégrés ou non à des cartes de positionnement, autorisent l'acquisition d'événements à des fréquences pouvant atteindre 100 kHz [7].

5 Rôle de l'API dans un système de production

L'API est donc d'abord un composant de commande, envoyant des signaux vers les actionneurs, en fonction des informations reçues de l'instrumentation du système matériel, la Partie Opérative, et des ordres reçus (consignes), selon une algorithmique appropriée définie par le programme.

La figure 2.9 explicite ce fonctionnement, la Partie Commande (pour nous l' API) agit dans le cadre d'un système bouclé sur les éléments matériels de l'installation, en vue d'une production, au sens large du terme : fabrication bien sûr, mais aussi emballage, autorisation d'accès, etc. De plus en plus, elle gère aussi l'énergie, incluant donc celle-ci, s'il s'agit d'élaborer matières premières ou objets, dans une Partie Opérative élargie.

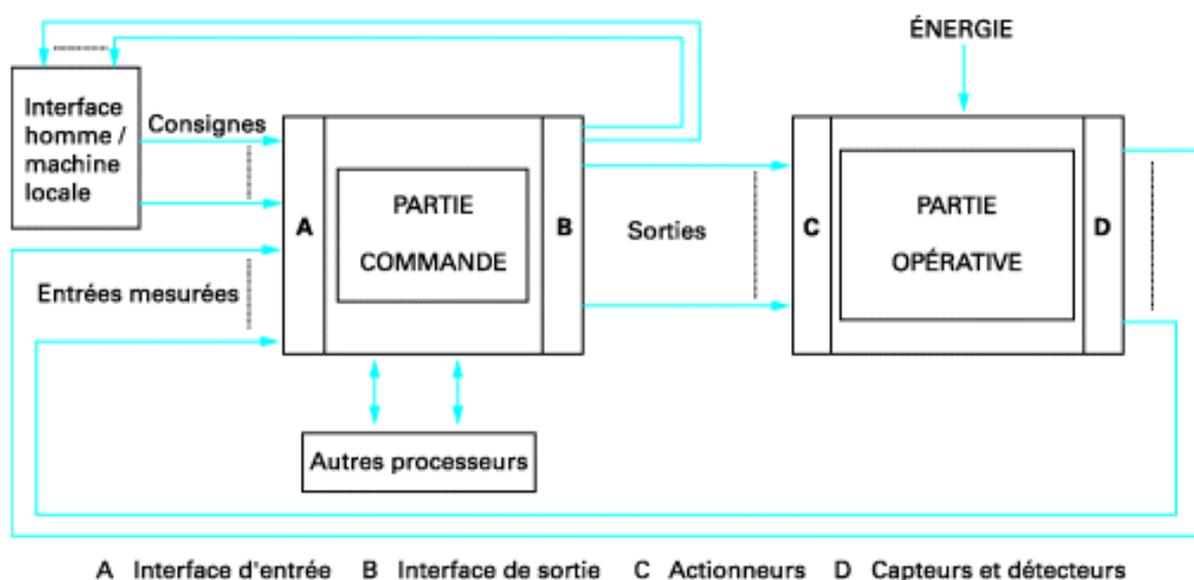


FIGURE 2.9 – Automate en commande [7]

Les flux d'informations qui apparaissent sur la figure 2.9 véhiculent les consignes, les signaux de commande et ceux émis par la Partie Opérative (mesures), mais aussi les informations en provenance ou à destination d'autres systèmes numériques. Au travers de la multiplicité des échanges possibles entre l'automate et son environnement apparaissent d'autres rôles éventuels de l'API : supervision locale, gestion de réseau, conditionnement de mesures, etc. mais aussi la possibilité de modifier de l'extérieur la commande, au niveau consignes comme à celui du paramétrage, voire de l'algorithme lui-même.

L'API s'intègre dans un ensemble de processeurs spécialisés et interconnectés, suivant une architecture décentralisée qui en facilite la conception et l'installation en permettant de fractionner les études, la mise en place, les tests, et qui en améliore aussi la maintenance (modification aisée des programmes, de parties du système automatisé). Elle entraîne toutefois, du fait des multiples sous-ensembles fonctionnels, un fort accroissement des besoins de communication et de gestion pour assurer une bonne coordination de cet ensemble [7].

6 Les objectifs de l'automatisation

Les objectifs de l'automatisation les plus essentiels sont les suivants :

- **Améliorer l'efficacité** : L'automatisation vise à augmenter la productivité en réduisant les tâches manuelles et en optimisant les processus. Cela permet de réaliser des économies de temps et de ressources, ainsi que d'améliorer la précision et la qualité du travail accompli.
- **Accroître la productivité** : En automatisant certaines tâches répétitives et chronophages, les entreprises peuvent libérer du temps pour se concentrer sur des activités à plus forte valeur ajoutée. Cela permet d'augmenter la productivité globale de l'entreprise et augmenter les heures de travail.
- **Réduire les erreurs humaines** : L'automatisation élimine les erreurs potentielles liées à la fatigue, à la distraction ou à l'inattention humaine. Les systèmes automatisés sont conçus pour exécuter des tâches de manière cohérente et précise, ce qui réduit les risques d'erreurs coûteuses.
- **Améliorer la qualité** : En automatisant les processus, il est possible de mettre en place des contrôles et des vérifications automatisés qui garantissent la conformité aux normes de qualité. Cela permet d'améliorer la fiabilité et la cohérence des produits et services.
- **Accélérer les délais de production** : L'automatisation permet d'accélérer la réalisation des tâches en réduisant les délais d'exécution. Les machines et les systèmes automatisés sont souvent capables de travailler plus rapidement et de manière plus efficace que les travailleurs humains, ce qui permet de gagner du temps.
- **Réduire les coûts** : L'automatisation peut réduire les coûts liés à la main-d'œuvre, aux erreurs, aux interruptions de production et à d'autres inefficacités. Bien que l'investissement initial dans l'automatisation puisse être élevé, les économies à long terme peuvent être significatives.
- **Améliorer la sécurité** : En remplaçant les tâches dangereuses ou risquées par des machines ou des robots, l'automatisation contribue à améliorer la sécurité des travailleurs. Cela réduit les risques d'accidents et de blessures professionnelles.

7 Présentation de l'automate programmable S7-1200

Pour automatiser notre machine on a utilisé l'automate S7-1200 qui est un contrôleur programmable développé par Siemens. Il fait partie de la famille de produits SIMATIC S7, qui est largement utilisée dans l'automatisation industrielle, il nous convient pour cette application à cause du compteur rapide et son nombre d'E/S est suffisant.

Caractéristiques clés de l'automate S7-1200 :

- **Puissance de traitement** : L'automate S7-1200 dispose d'un processeur intégré puissant qui permet d'exécuter rapidement les programmes et de traiter les entrées/sorties.
- **Entrées/sorties (E/S) intégrées** : Il est équipé d'une gamme d'interfaces d'E/S numériques et analogiques intégrées qui permettent de se connecter à différents capteurs, actionneurs et autres équipements sur le terrain.
- **Communication** : L'automate S7-1200 prend en charge plusieurs protocoles de communication, tels que PROFIBUS, PROFINET, Modbus et Ethernet TCP/IP. Cela permet d'établir des connexions avec d'autres automates, systèmes de supervision, périphériques et réseaux industriels.
- **Programmation** : Il est programmé à l'aide du logiciel STEP 7 Basic, qui offre une interface conviviale pour le développement et la configuration des applications. Le langage de programmation utilisé est le langage de programmation graphique par blocs fonctionnels (FBD), ainsi que le langage d'instructions structurées (LADDER) et d'autres langages.
- **Flexibilité modulaire** : L'automate S7-1200 peut être étendu avec des modules d'extension, tels que des modules d'E/S supplémentaires, des modules de communication ou des modules de fonction spécifique, afin de répondre aux besoins spécifiques de l'application.
- **Fonctions de sécurité** : Il intègre des fonctionnalités de sécurité intégrées, telles que la surveillance des entrées de sécurité, les arrêts d'urgence et les fonctions de sécurité programmables, permettant de répondre aux exigences de sécurité des machines et des processus industriels.
- **Maintenance et diagnostic** : L'automate S7-1200 dispose de fonctionnalités avancées de maintenance et de diagnostic, telles que la surveillance en temps réel, les journaux d'événements, la gestion des alarmes et la possibilité de mise à jour du firmware.

L'automate S7-1200 représenté dans la figure 2.10 est utilisé dans une variété d'applications industrielles, telles que le contrôle des processus, l'automatisation des machines, la surveillance et la gestion des systèmes. Sa taille compacte, sa flexibilité et ses performances en font un choix populaire pour les petites et moyennes applications d'automatisation.

L'entreprise Electro-Industries (ENEL) a fait le choix de la firme SIEMENS en raison de l'évolution de cette dernière. ENEL a toujours été fidèle à SIEMENS depuis la première génération d'automates jusqu'au S7. SIEMENS est réputée pour sa technologie évolutive, ainsi que pour la fiabilité et la robustesse de ses produits.



FIGURE 2.10 – Automate programmable S7-1200 1214C [9]

8 Caractéristiques techniques de l'API S7-1200 1214C

Les caractéristiques techniques sont :

- 14 entrées et 10 sorties TOR
- 2 entrées analogiques
- Une mémoire de travail 100 kilo-octets
- Une mémoire de chargement de 4 Méga-octets
- Processeur intégré avec une fréquence de fonctionnement de 100 MHz
- Temps de traitement par la CPU 0.08 μs pour opérations sur bits et 2.3 μs pour opérations à virgule flottante
- Interface de communication PROFINET de type RJ 45(Ethernet), d'une vitesse de transmission de 100 Mbit/s
- 6 compteurs rapides (HSC) incrémentation/décrémentation avec une fréquence de comptage maximale de 100 kHz

Conclusion

Dans ce chapitre on a donné une description détaillée de l'automate programmable, ses constituants, fonctions de base, son rôle dans le système automatisé et on a mentionné l'avantage apporter par cet appareil à l'industrie. Ensuite on a présenté l'API S7-1200 1214C qu'on a utilisé dans ce projet avec ses caractéristiques techniques.

On va entamer la programmation de l'automate à partir du logiciel TIA Portal V13 en langage Ladder (langage à contacts), qui est l'intermédiaire entre l'opérateur et l'API dans le chapitre suivant.

Chapitre 3

Programmation de L'automate

Introduction

Afin d'automatiser la machine, on a opté pour l'utilisation de l'automate programmable S7-1200. Le logiciel TIA Portal V13 de programmation offre un environnement convivial pour concevoir, modifier et superviser la logique requise pour contrôler une application. Le projet permet d'associer le programme à toutes les informations nécessaires pour communiquer avec l'API et charger le programme dans ce dernier.

Tout d'abord, on décrira la procédure à suivre pour créer et configurer le matériel d'un projet d'automatisation, ainsi que la structure d'un projet dans le logiciel TIA Portal, ensuite on expliquera la logique derrière le programme et comment il servira à effectuer le bobinage nécessaire d'après les consignes données par l'opérateur de la machine.

1 La conception d'un programme avec TIA Portal V13

Pour concevoir un programme avec le logiciel TIA Portal V13 il faut suivre les étapes suivantes :

- Créer un nouveau projet
- Sélectionner le matériel à utiliser d'après les références et activer les options nécessaires (Compteur rapide...etc.)
- Ajouter les blocs de données et les blocs fonctionnels
- Remplir la table des variables et les blocs de données
- Construire le programme dans le bloc main OB1 et les blocs fonctionnels
- Charger la configuration après compilation
- Exécuter le programme et effectuer les tests nécessaires, pour confirmer la bonne exécution de l'opération.

Dans ce qui suit on va élaborer le travail qu'on a effectué pour concevoir le programme. la figure 3.1 montre les étapes de la conception.

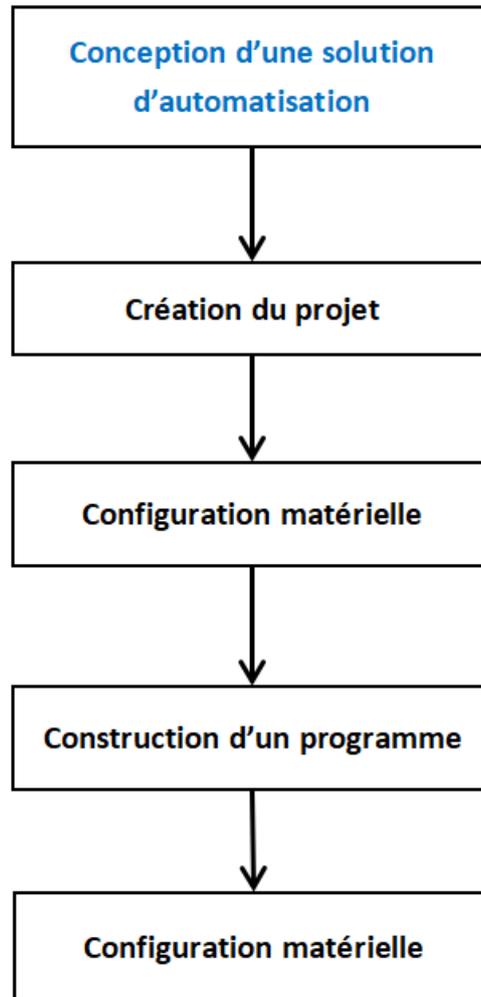


FIGURE 3.1 – Organigramme pour création d'un projet sous TIA Portal V13

2 Création d'un nouveau projet

Après démarrage du logiciel comme indique la figure 3.2 on suit les étapes suivantes pour créer un projet :

- Sélectionner l'action "créer un projet"
- Choisir un nom et le chemin souhaités pour le projet et définir l'auteur
- Ajouter un commentaire
- Cliquer sur le bouton "créer".

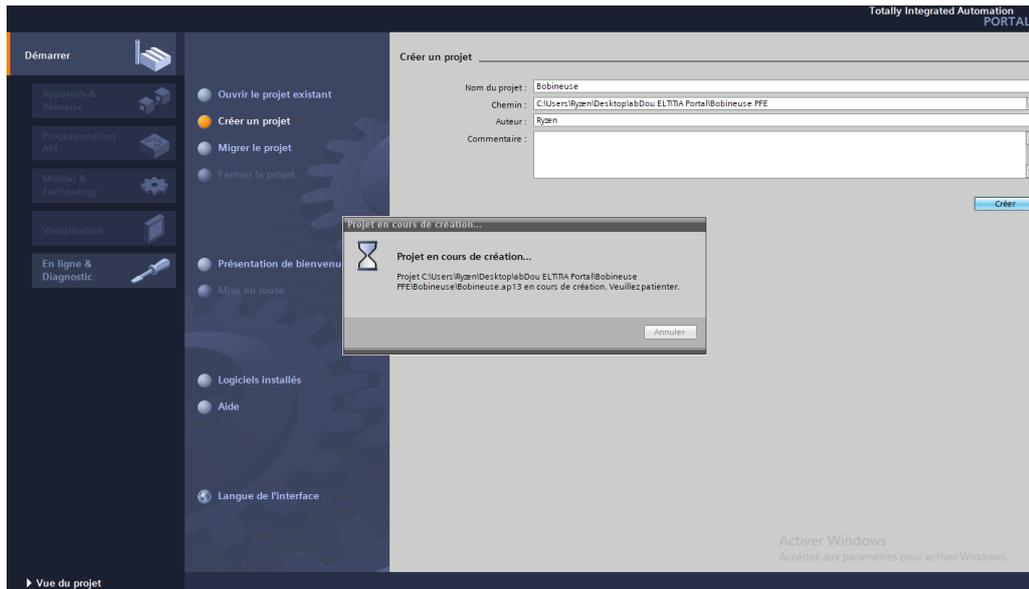


FIGURE 3.2 – Fenêtre de création d’un nouveau projet

3 Vue du portail

On peut configurer plusieurs tâches à partir de cette vue comme indique la figure 3.3, mais on préfère ouvrir d’abord la vue du projet ensuite régler la configuration matérielle.

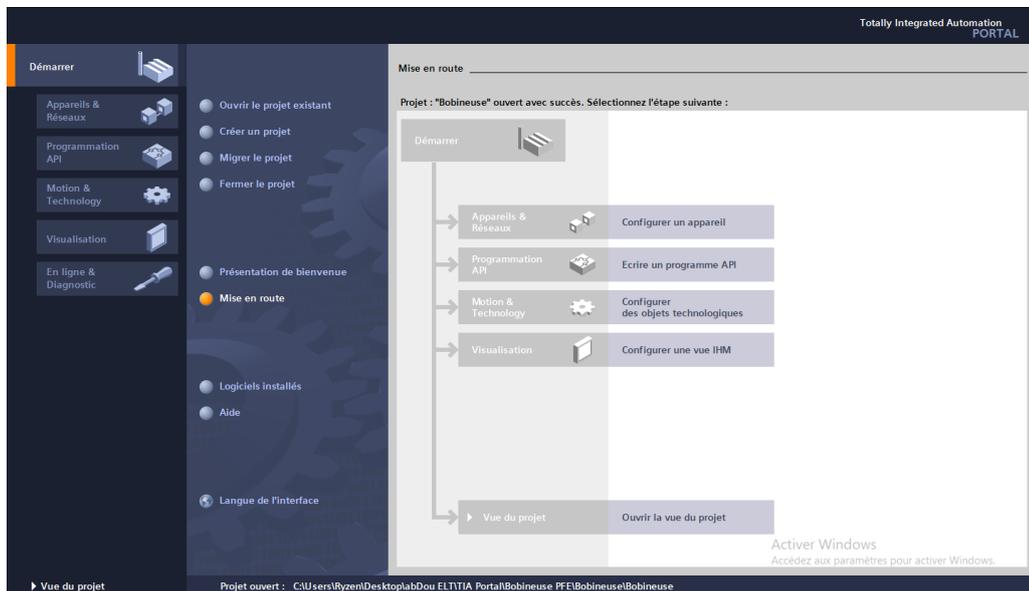


FIGURE 3.3 – Vue du portail

4 Configuration matérielle

Après ouvrir la vue du projet, on clique deux fois sur le l'action "Ajouter un appareil" qui se trouve en haut du navigateur du projet, une autre fenêtre apparaît au centre de l'écran comme sur la figure 3.4, qui affiche tous les appareils qu'on peut ajouter à notre projet.

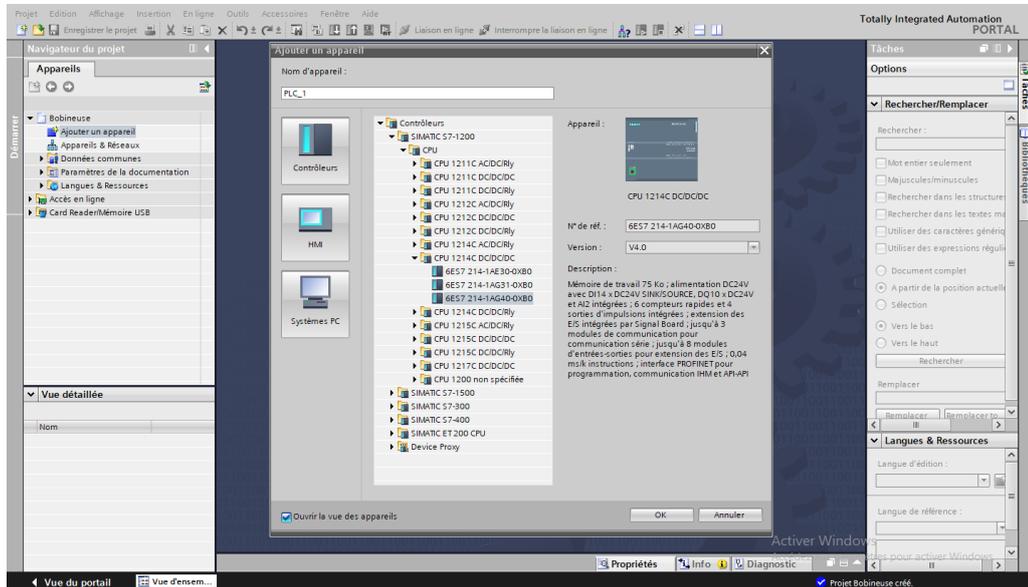


FIGURE 3.4 – Ajout de la CPU

- A gauche de la fenêtre on sélectionne "contrôleurs"
- on ouvre le dossier "SIMATIC S7-1200" puis le dossier "CPU 1214C DC/DC/DC"
- On choisit la référence "6ES7 214-1AG40-0XB0" comme l'appareil disponible à l'établissement
- Enfin on clique sur "OK" et la fenêtre "configuration de l'appareil" s'ouvre, comme indique la figure 3.5.

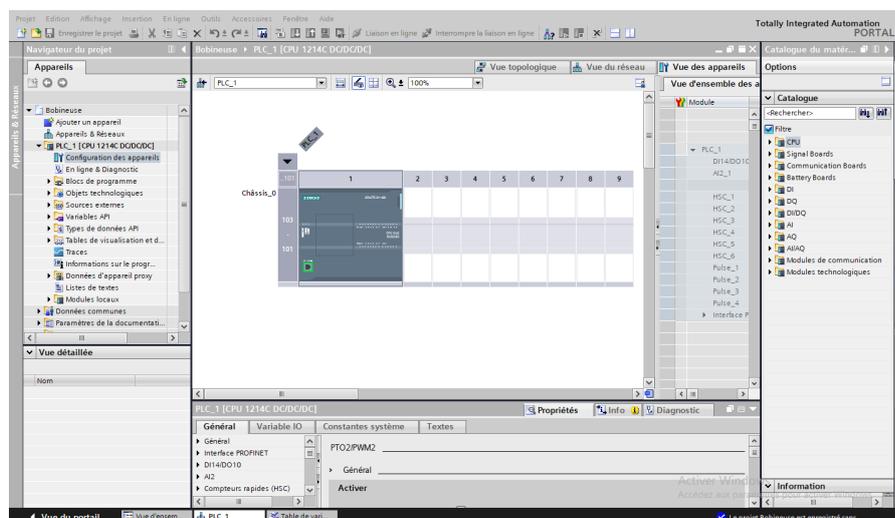


FIGURE 3.5 – Configuration de l'appareil

A partir de cette fenêtre on accède les propriétés de la CPU afin d'effectuer les tâches suivantes :

- Activer le compteur rapide "HSC1" en sélectionnant à gauche "général", comme indique la figure 3.6

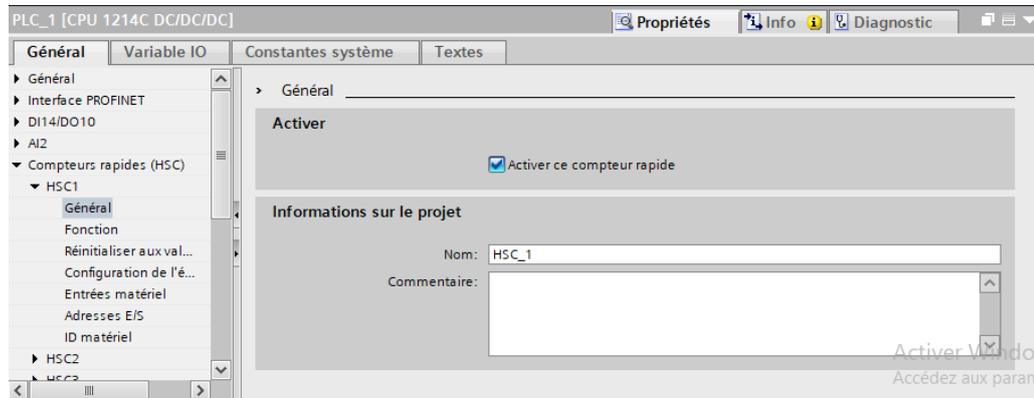


FIGURE 3.6 – Activation du compteur rapide

- La figure 3.7 montre comment définir le type de comptage "Compteur A/B" au lieu de comptage monophasé, et vérifier que le sens de comptage initial est incrémental
- Mémoriser les informations suivantes :
 - Les entrées matériels :
 - Entrée horloge A : %I0.0
 - Entrée horloge B : %I0.1
 - Adresses E/S :
 - Début : 1000
 - Fin : 1003
 - Le ID matériel : 257

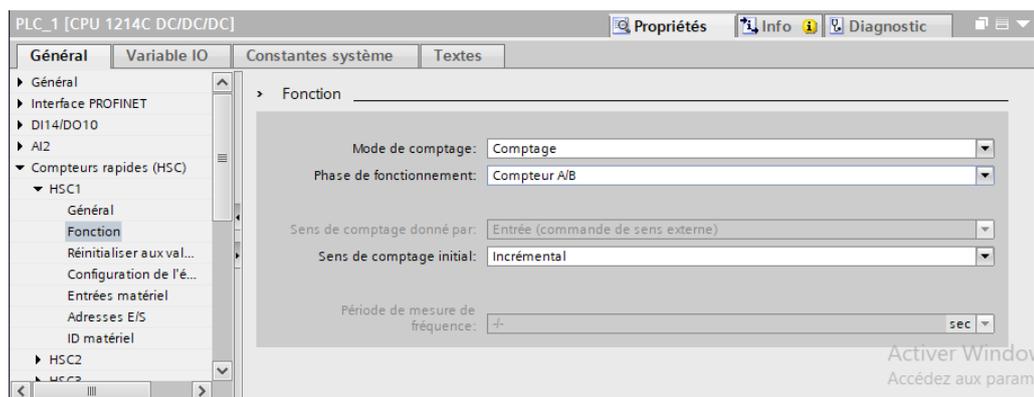


FIGURE 3.7 – Configuration du compteur rapide

- Activer le memento de cadence qui servira à ajouter l'effet de clignotement des LEDs, dans ce cas on a réservé l'octet 40 pour la cadence comme dans la figure 3.8, chaque bit de cet octet a une fréquence. Par exemple le bit %M40.0 de fréquence 10 Hz s'active 10 fois par seconde.

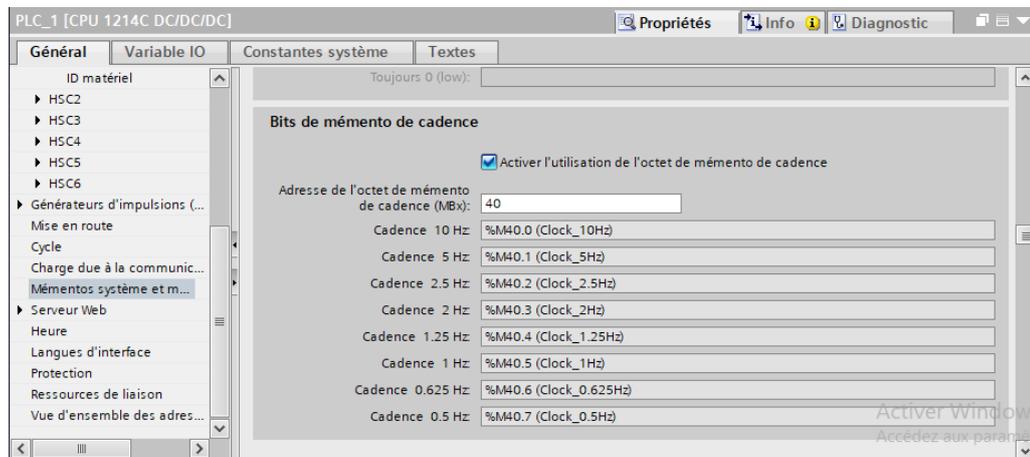


FIGURE 3.8 – Activation du memento de cadence

5 Création du bloc de données

Dans ce programme il existe plusieurs valeurs de consigne qu'on va les déclarés comme variables dans un bloc de données, la procédure ce fait comme suit :

- On sélectionne dans le navigateur de projet : blocs du programme -> ajouter un nouveau bloc, la figure 3.9 montre la fenêtre "Ajouter nouveau bloc"

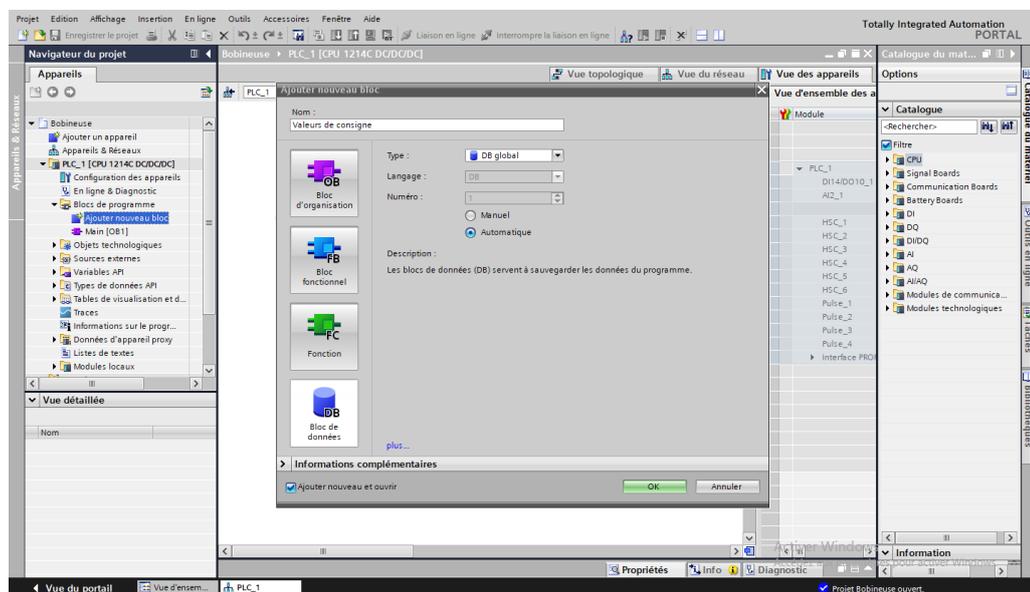


FIGURE 3.9 – Ajout d'un bloc de données

- Dans la fenêtre qui s'ouvre on clique sur bloc de données puis on définit un nom pour ce bloc et on clique sur "OK"
- Après avoir créer le bloc de données "valeurs de consigne" on remplit le bloc comme dans la figure 3.10 avec les variables suivants :

Nom	Type de données	Valeur de départ	Rémanence	Accessible ...	Visible da...	Valeur de ...	Commentaire
1	Static		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	Consigne1	Int 0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	Consigne2	Int 0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	Consigne3	Int 0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5	Consigne4	Int 0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6	Nbr de spires	Int 0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
7	Nbr de spires - 2	Int 0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
8	Nbr de faisceaux	Int 0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
9	Impulsions ns	Int 0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
10	Conversion ns	Int 0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
11	Valeur max de compt...	Int 0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
12	Comptage	Int 0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
13	Cycle	Int 0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
14	Déplacement	Real 0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
15	Déplacement total	Real 0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
16	Impulsions	Real 0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
17	Conversion en mm	Real 0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
18	Marche	Bool false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
19	Arrêt	Bool false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
20	Redémarrer	Bool false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
21	Retour	Bool false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
22	Arrêt alarmes	Bool false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
23	Capot	Bool false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

FIGURE 3.10 – Remplissage du bloc

- Consigne1 : c'est le nombre de spires qu'on veut avoir dans le premier faisceau
- Consigne2 : c'est le nombre de spires qu'on veut avoir dans le deuxième faisceau
- Consigne3 : c'est le nombre de spires qu'on veut avoir dans le troisième faisceau
- Consigne4 : c'est le nombre de spires qu'on veut avoir dans quatrième faisceau
- Nbr de spires : c'est la variable qui prends les valeurs des consignes précédentes en fonction de la position du chariot-guide.
- Nbr de spires - 2 : c'est la soustraction de la valeur de la variable précédente de 2
- Nbr de faisceaux : c'est le nombre de faisceaux du gabarit de bobinage choisit avant démarrer la machine (3 ou 4)
- Impulsions ns : l'encodeur envoie des impulsions en type DInt (double-integer), cet variable est la conversion de l'information envoyée en type Int, pour obtenir la mise à l'échelle en nombre de spires
- Conversion ns : la mise à l'échelle en nombre de spires de la variable précédente
- Valeur max de comptage : c'est le nombre maximum de faisceaux à effectuer par cycle de bobinage (pour 3 faisceaux il y a 2 cycles, pour 4 faisceaux il y a un seul cycle)
- Comptage : c'est la valeur du compteur qui compte les faisceaux effectués en temps réel
- Cycle : c'est la valeur de comptage du cycle en temps réel
- Déplacement : la distance entre deux faisceaux, on peut la modifier comme les 4 premières variables
- Déplacement total : c'est la distance totale parcouru par le chariot-guide en cours de l'opération

- Impulsions : c'est la conversion en type "Real" de l'information produite par l'encodeur pour avoir la mise à l'échelle en millimètres
 - Conversion en mm : la mise à l'échelle en millimètres de la variable précédente
 - Marche : c'est le bouton qui sert à commencer l'opération de bobinage, on peut l'activer à partir du bouton "Marche" définie dans l'écran
 - Arrêt : c'est le bouton qui sert à arrêter en tout moment l'opération de bobinage, on peut l'activer à partir du bouton "Arrêt" définie dans l'écran
 - Redémarrer : c'est le bouton qui sert à redémarrer le bobinage en cas d'arrêt au milieu de l'opération, on peut l'activer à partir du bouton "Redémarrer" définie dans l'écran
 - Retour : c'est le bouton qui sert à renvoyer le chariot-guide vers sa position initiale, en cas de fin de bobinage ou bien en cas d'arrêt au milieu de l'opération, on peut l'activer à partir du bouton "Retour" définie dans l'écran
 - Arrêt alarmes : c'est le bouton qui sert à désactiver les alarmes, on peut l'activer à partir du bouton "Retour" définie dans l'écran
 - Capot : c'est la variable qui suit l'ouverture et la fermeture du capot, elle est activé quand le capot est fermé, et elle se désactive dès que le capot s'ouvre
- Après avoir remplir le bloc, il faut le compiler seul pour pouvoir utiliser ces variables, la figure 3.11 représente le résultat de la compilation.

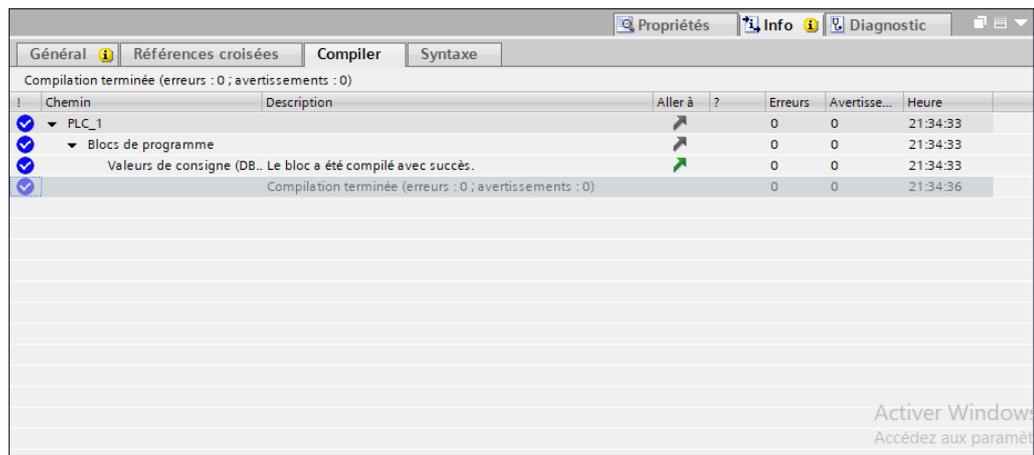


FIGURE 3.11 – Compilation du bloc de données

Par la suite on commence la programmation dans le bloc "main OB1" en langage Ladder.

6 Programme

Dans la programme on a utilisé plusieurs instructions dont on va expliquer leurs principes de fonctionnement au fur et à mesure dans ce qui suit.

3.6.1 Mise à l'échelle des données de l'encodeur

Comme on a expliqué dans le premier chapitre, le mouvement de rotation de l'encodeur génère des impulsions qui seront envoyés vers l'automate, le nombre d'impulsions sera compter par le compteur rapide qu'on a activé précédemment.

La variable de type "DInt" %ID1000 portera le nombre d'impulsions, on va utiliser cette variable afin d'obtenir son équivalent en "Int" et en "Real", ces derniers vont servir à calculer l'équivalent du nombre d'impulsions en nombre de spires et en millimètres, en tenant compte que l'encodeur utilisé génère 600 impulsions par tour.

L'instruction de conversion se trouve dans le dossier "Conversion" sous le nom "CONVERT" comme dans la figure 3.12, et la mise à l'échelle se fait avec la fonction mathématique "CALCULATE" comme dans la figure 3.13, qui définit et exécute, en fonction du type de données choisi, une expression pour le calcul d'opérations mathématiques ou l'exécution d'opérations logiques complexes.

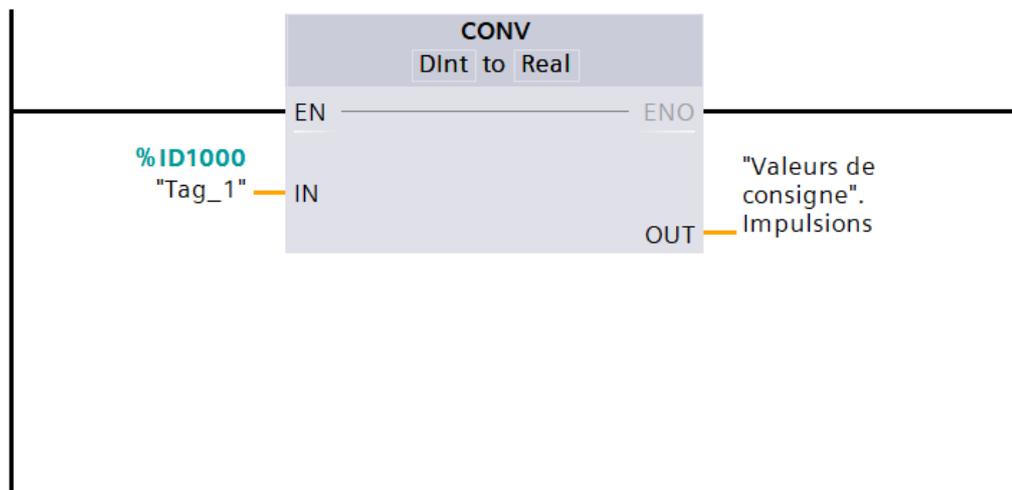


FIGURE 3.12 – Conversion de DInt à Real

On suppose que la distance parcourue par tour est de 10 mm, c'est à dire que l'équivalent du nombre d'impulsions en millimètres se trouve par l'opération mathématique : ($\text{"Impulsions"} \cdot 10$) / 600

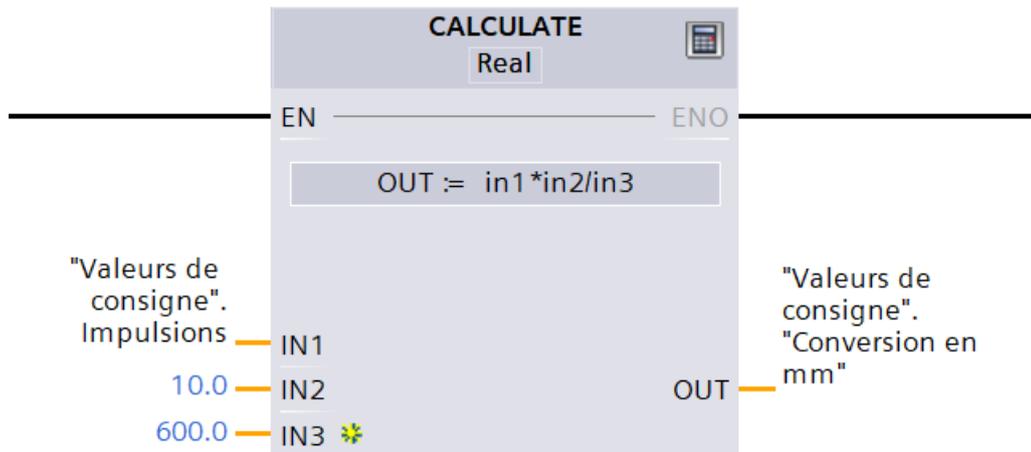


FIGURE 3.13 – Calcul de l'équivalent en millimètres

On obtient l'équivalent en nombre de spires en convertissant la variable %ID1000 en un "Int", puis on calcule l'équivalent par une simple division, la figure 3.14 montre le réseau qui effectue ces instructions.

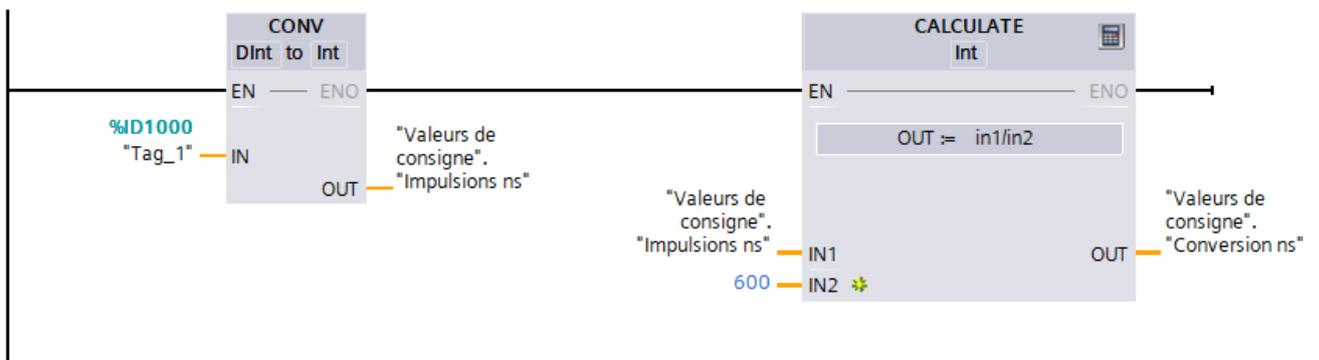


FIGURE 3.14 – Conversion de DInt à Real

3.6.2 Affectation des consignes

Pour avoir la consigne de nombre de spire de chaque faisceau au bon moment, on utilise la variable du compteur ("Comptage") qui sert à compter le nombre de faisceaux effectuées, la figure 3.15 montre comment on a utilisé l'instruction "MOVE" pour affecter la valeur de chaque consigne à la variable "Nbr de spires", en tenant compte de la valeur de comptage.

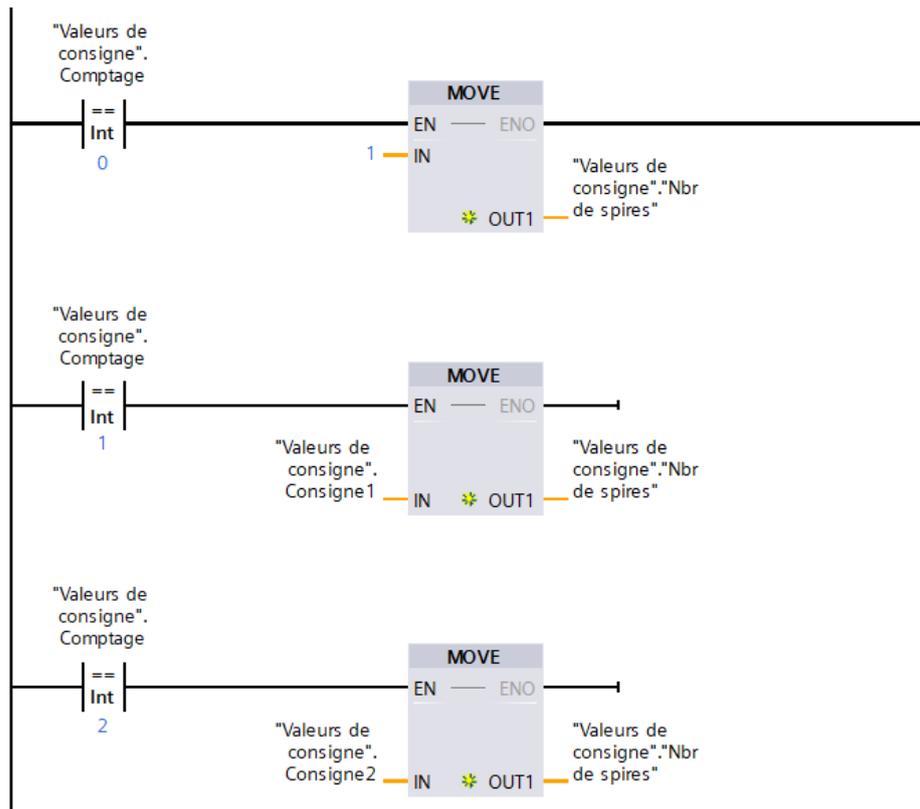


FIGURE 3.15 – Affectation du nombre de spires

Le contact utilisé est un contact de comparaison qui se ferme si et seulement si la valeur en haut est égale à celle en bas du contact.

3.6.3 Contrôle du moteur de la vis-mère

Cette partie est un peu large pour pouvoir l'afficher en une seule figure, donc on va la diviser en plusieurs parties, il faut juste savoir que tout ce qu'on va montrer est du même réseau et non pas de réseaux différents

- Le bouton "Marche" sert à démarrer la machine bobineuse seulement quand elle est dans la position initiale, c'est pour cela on a mis plusieurs contacts en série avec le contact "Marche" comme indique la figure 3.16, ces derniers servent à éliminer l'influence du bouton "Marche" (en cas de son activation) en milieu de l'opération.

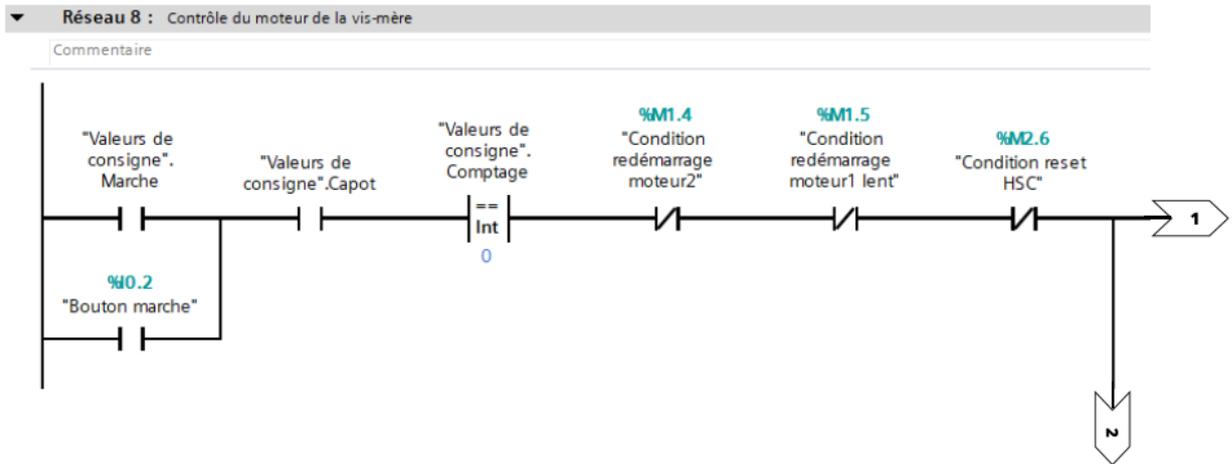


FIGURE 3.16 – Contrôle du moteur de la vis-mère partie 1

- Sur la figure 3.17, on voit le premier contact en parallèle sert à démarrer le moteur de la vis-mère après le bobinage d'un faisceau es terminé, les autres contacts en parallèle servent à déclencher le retour du chariot-guide à sa position initiale ou redémarrer le bobinage en cas d'un arrêt au milieu de l'opération.

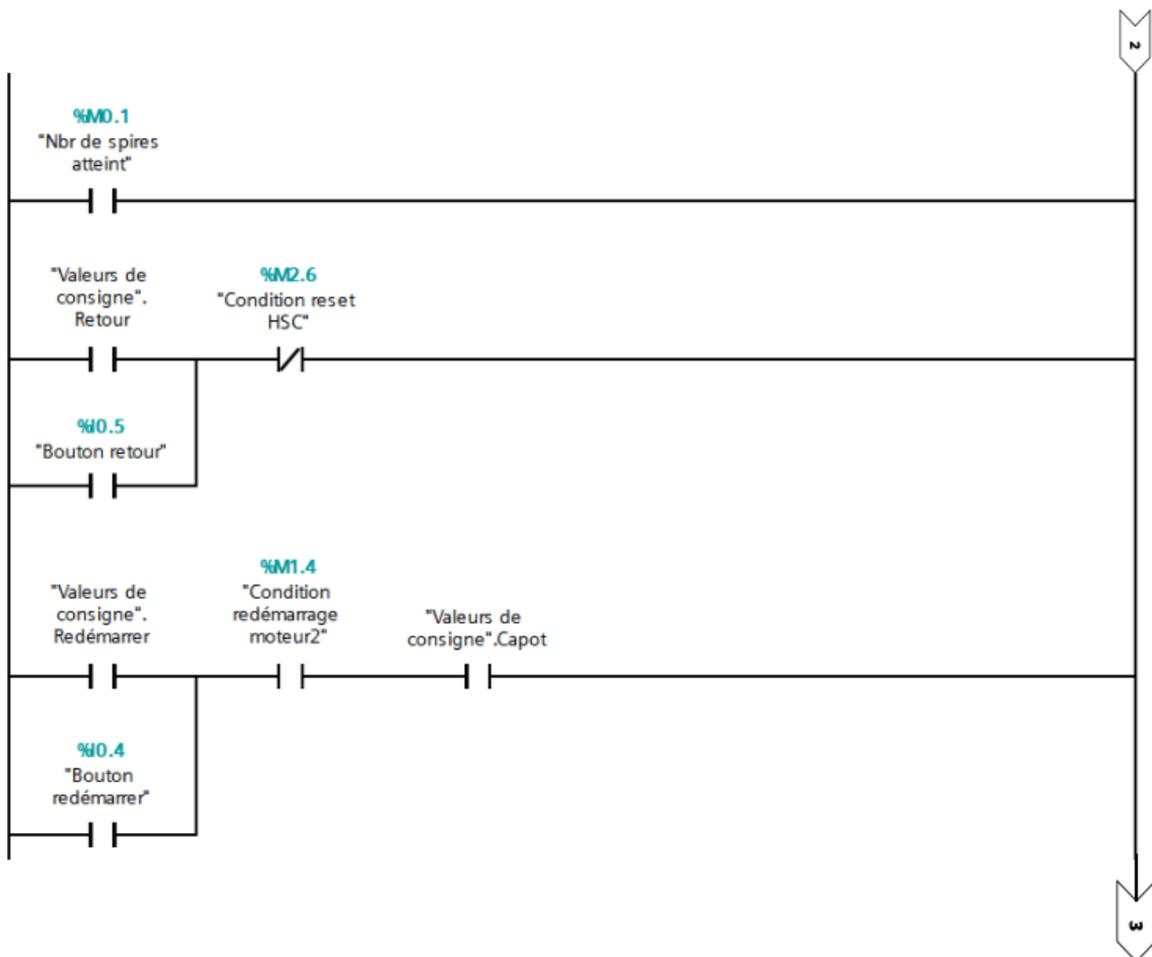


FIGURE 3.17 – Contrôle du moteur de la vis-mère partie 2

- Comme dans la figure 3.18, le contact en parallèle de la variable %Q0.2 sert à maintenir l'activation de la même variable qui va apparaître une autre fois sur une affectation.



FIGURE 3.18 – Contact de maintien

- Les contacts à ouverture représentés dans la figure 3.19, se trouvent au milieu, ils servent à arrêter l'affectation qui vient juste après, à cause de fin du déplacement.

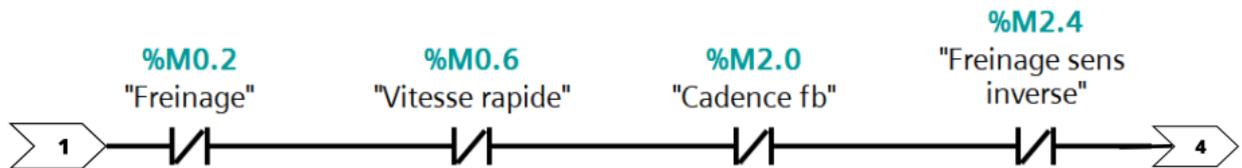


FIGURE 3.19 – Coupure par contacts à ouvertures

- Les sorties qui commande le moteur de la vis-mère et son frein, représentées dans la figure 3.20, alimentent les actionneurs quand la valeur de l'affectation est à 1, le rôle de la temporisation consiste à créer un laps de temps entre le démarrage du moteur et le défreinage.

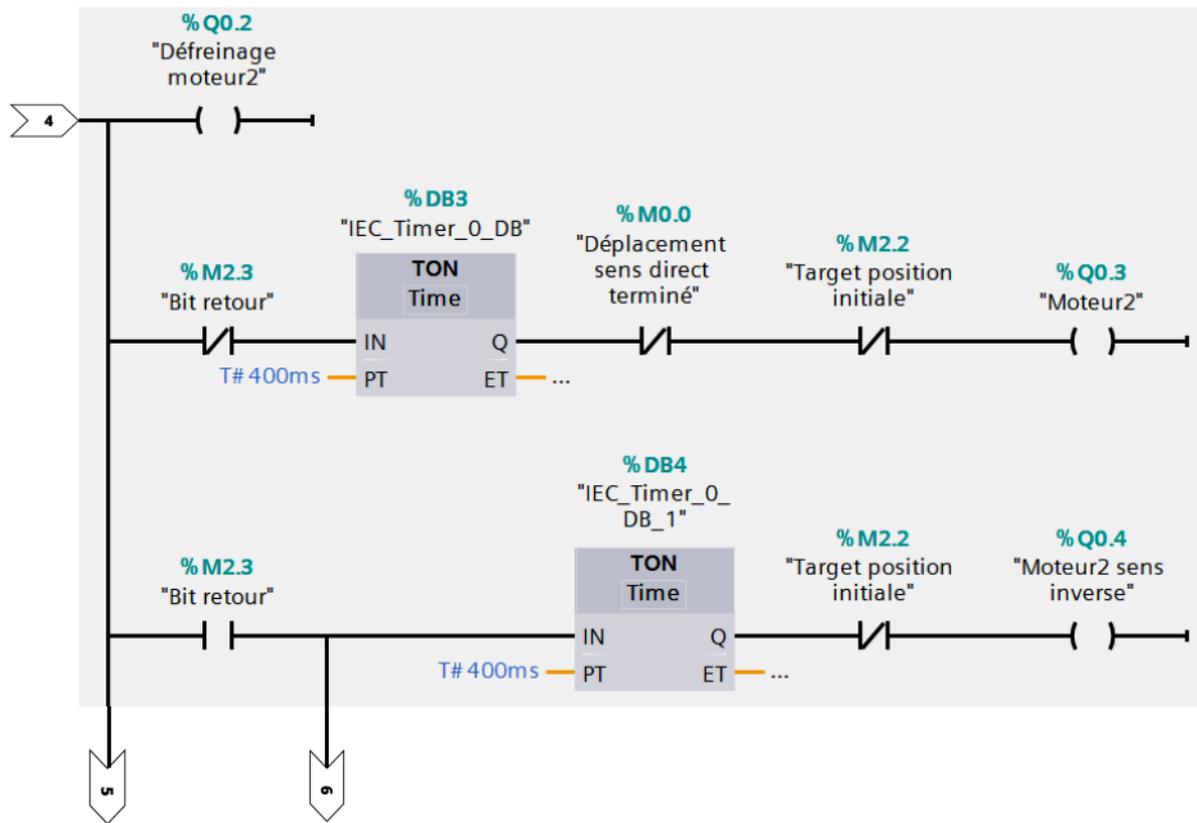


FIGURE 3.20 – Les affectations des sorties

- Les méments en bobine "S" (mise à 1 sortie), indiqués dans la figure 3.21, sont utilisés pour contrôler le redémarrage en cas d'un arrêt au milieu.

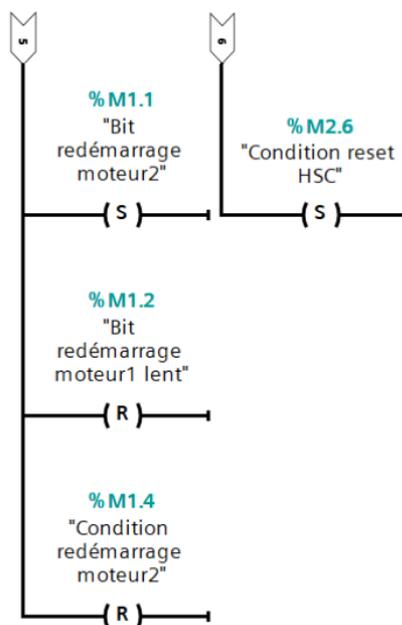


FIGURE 3.21 – Les méments de redémarrage

La figure 3.22 montre une vue globale du réseau de contrôle du moteur de la vis-mère :

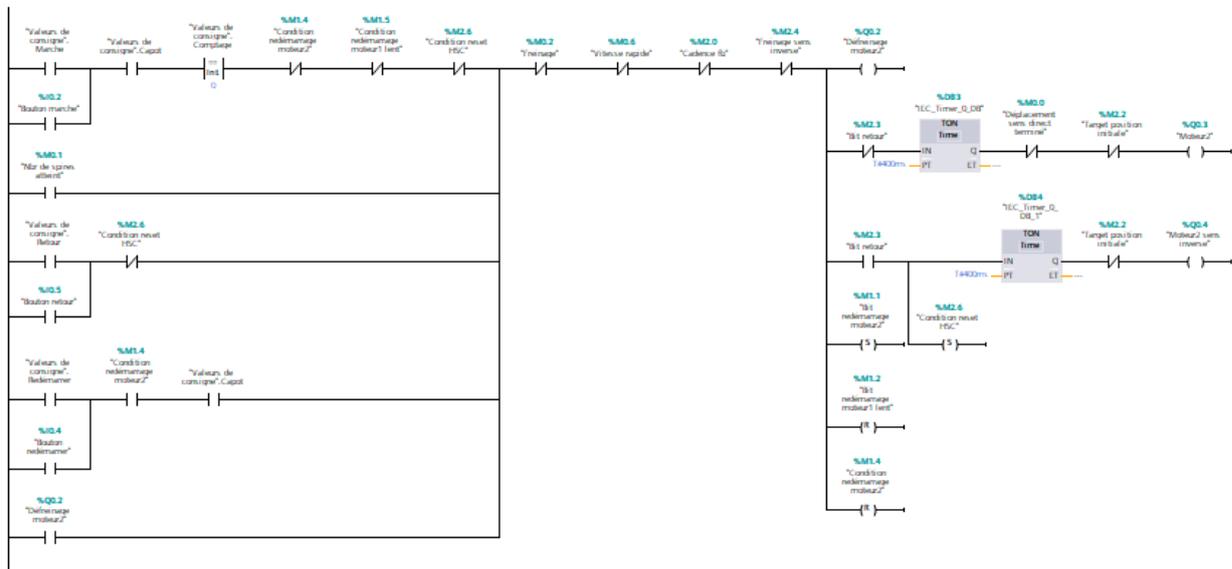


FIGURE 3.22 – Vue globale du réseau

- En cas d'arrêt de l'opération au moment du déplacement du chariot-guide, le memento **%M1.4**, indiqué sur la figure 3.23 en bobine "S", s'active pour pouvoir redémarrer le bobinage.

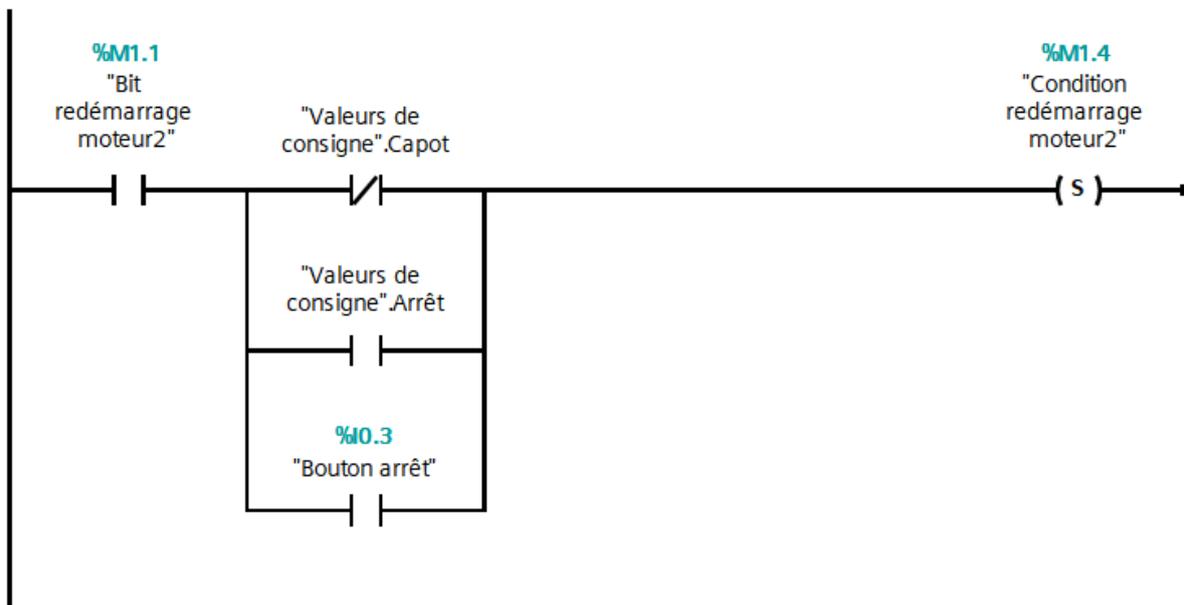


FIGURE 3.23 – Condition redémarrage moteur 2

- Le contact de comparaison, représenté dans la figure 3.24, assure l'activation de la sortie jusqu'à ce que la variable "Conversion en mm" devienne supérieure ou égale à la variable "Déplacement", c'est à dire le chariot-guide a bien complété le déplacement d'un faisceau à l'autre.

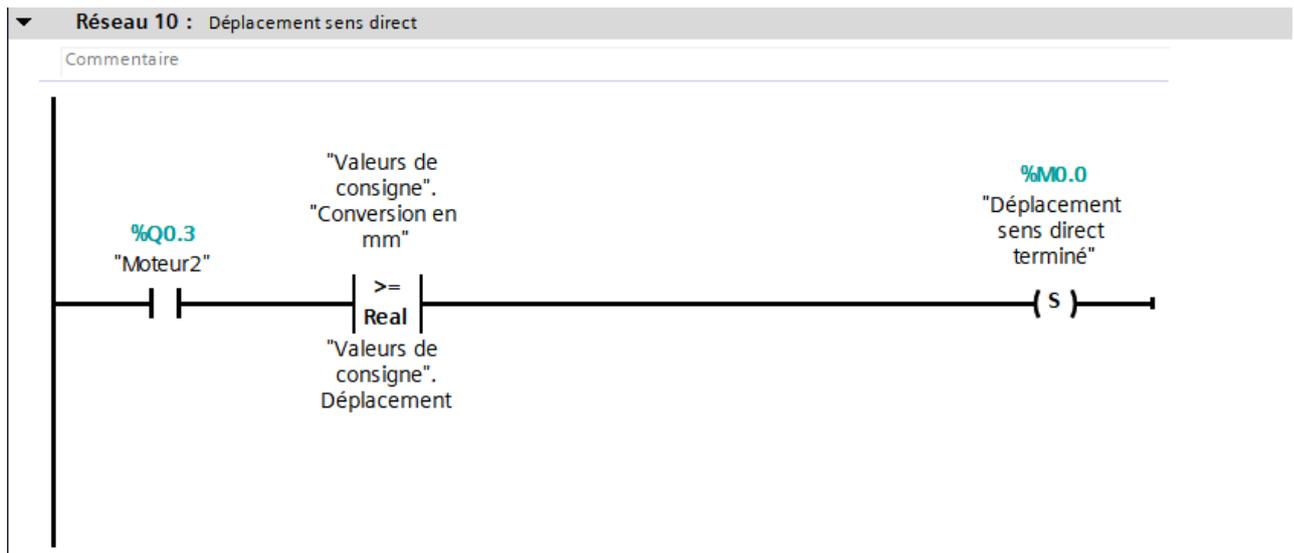


FIGURE 3.24 – Déplacement sens direct

- Après avoir terminé le déplacement il faut choisir la vitesse du moteur de bobinage, comme indique la figure 3.25, soit la vitesse rapide pour des consignes supérieur à 2, ou bien la vitesse lente pour des consignes inférieur à 3, cela pour pouvoir effectuer à chaque fois les deux dernières spires en vitesse lente.

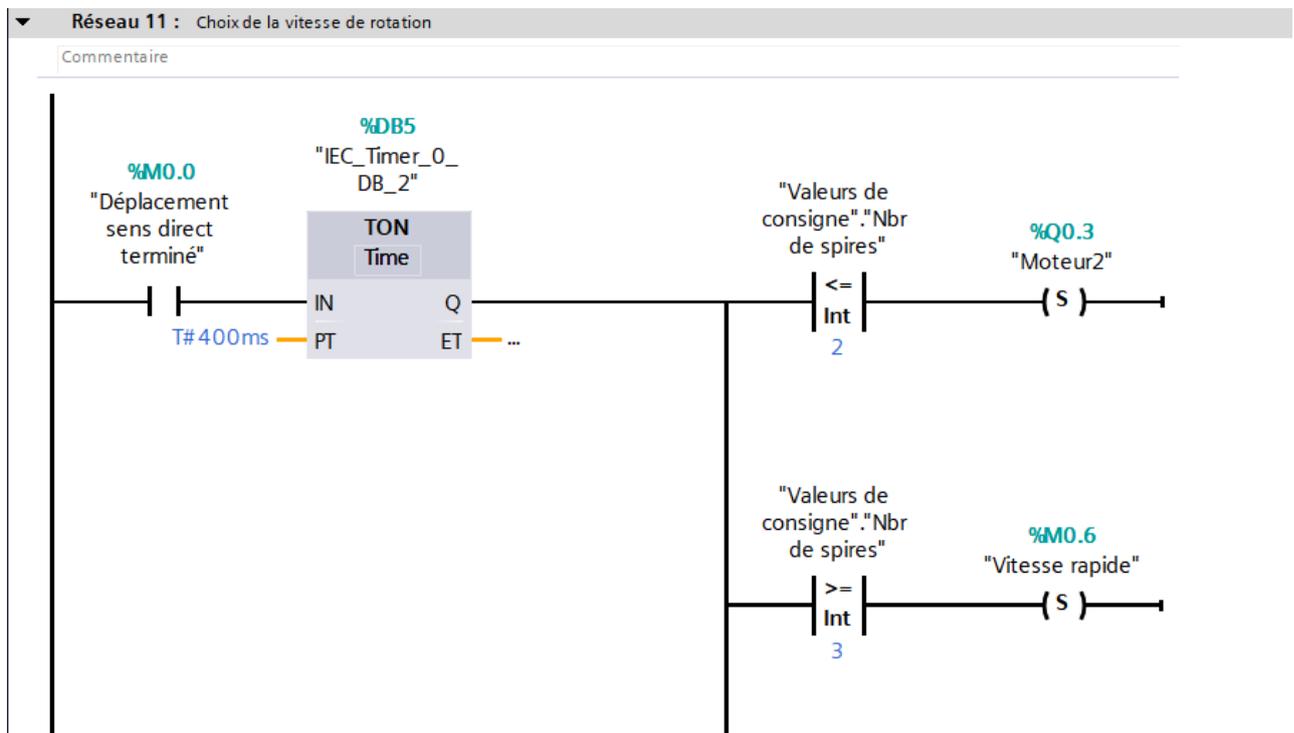


FIGURE 3.25 – Choix de la vitesse de rotation

3.6.4 Commande du moteur de bobinage

Le réseau de contrôle de la sortie "moteur 1 vitesse lente" sera diviser en plusieurs morceaux pour pouvoir l'afficher clairement, de même pour le réseau de contrôle de la sortie "moteur 1 vitesse rapide", ces deux sorties sont en mesure de commander la vitesse de rotation du moteur de bobinage, soit pour le moteur Dahlander ou pour un moteur standard par l'intermédiaire d'un variateur de vitesse.

3.6.4.1 Moteur en vitesse lente

- Le memento `%M0.2` sert à activer la sortie (moteur 1 vitesse lente) lorsque le déplacement est terminé et le nombre de spires à effectuer est inférieur à 3, alors que le memento `%M0.3` active la sortie lorsque le bobinage en vitesse rapide est terminé, les deux contacts à ouverture qui viennent après servent à désactiver la sortie lorsque le bobinage est terminé, comme indique la figure 3.26.

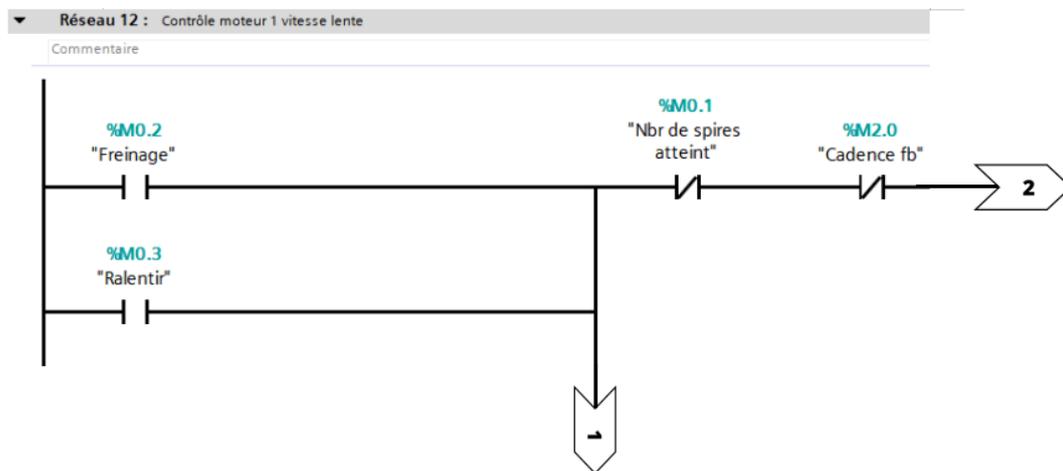


FIGURE 3.26 – Contrôle du moteur 1 vitesse lente partie 1

- La figure 3.27 montre la partie en parallèle qui permet de redémarrer le moteur lorsque le bobinage s'est arrêté, plus le contact de maintien qui assure l'activation de la sortie `%Q0.1`.

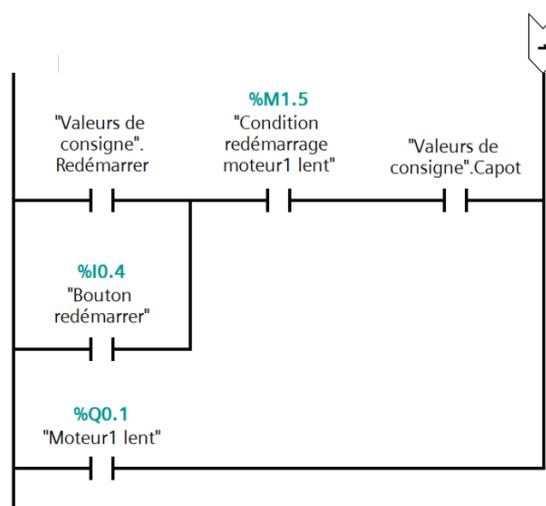


FIGURE 3.27 – Contrôle du moteur 1 vitesse lente partie 2

- Après les parties de contrôle vient l'affectation de la sortie %Q0.1 et la mise à 1 et à 0 des mementos de commande, comme indique la figure 3.28, le contact à ouverture de la sortie %Q0.0 empêche l'activation de la sortie %Q0.1 en cas du choix de la vitesse rapide.

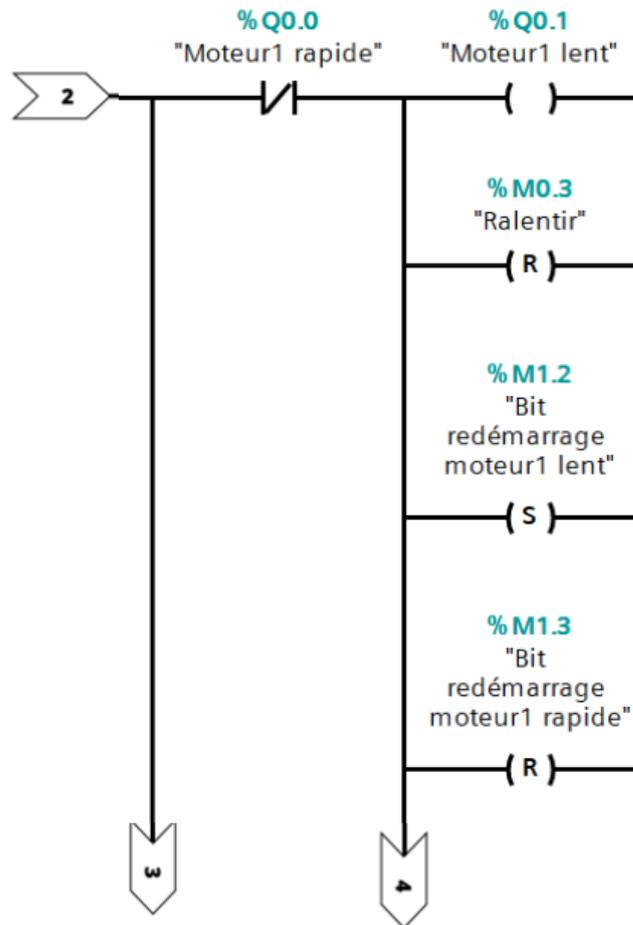


FIGURE 3.28 – Affectation 1

- Après on a la mis à 1 du memento de redémarrage, et la mise à 0 du memento de freinage avec le contact à ouverture qui assure la désactivation de la sortie avant la mise à 0 du memento, comme dans la figure 3.29.

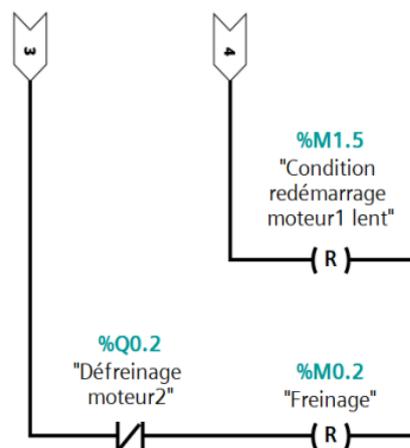


FIGURE 3.29 – Affectation 2

La figure 3.30 montre une vue globale du réseau de contrôle de la sortie "moteur 1 vitesse lente" :

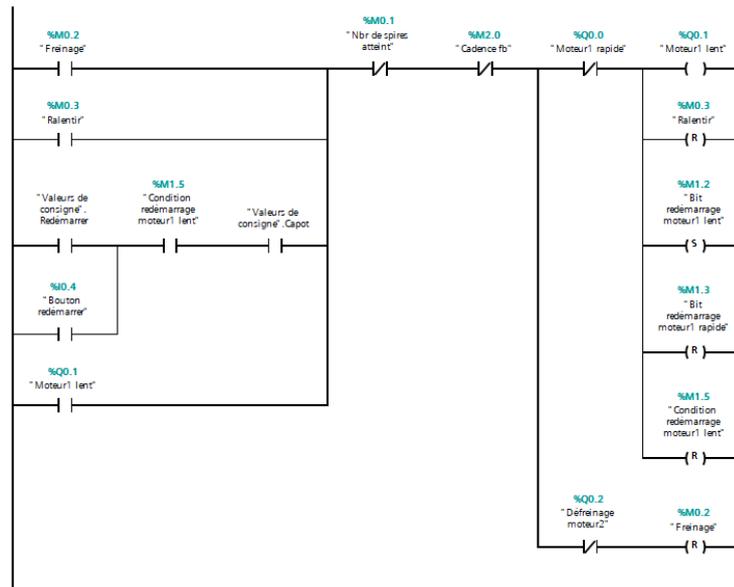


FIGURE 3.30 – Vue globale du réseau

- Ce réseau montré dans la figure 3.31, donne la possibilité de redémarrer le bobinage.

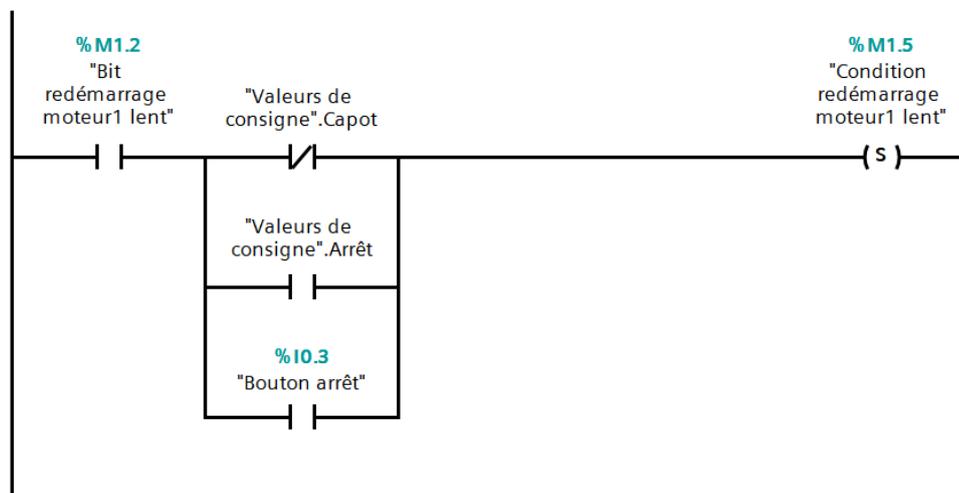


FIGURE 3.31 – Condition redémarrage moteur1 vitesse lente

- Le contact de comparaison, représenté sur la figure 3.32, assure l'activation de la sortie jusqu'à ce que la variable "Conversion ns" devienne supérieur ou égale à la variable "Nbr de spires", c'est à dire le moteur 1 a bien complété le bobinage en vitesse lente et le memento %M0.1 qui désactive le moteur 1 et réactive le moteur de la vis-mère sera mis à 1.

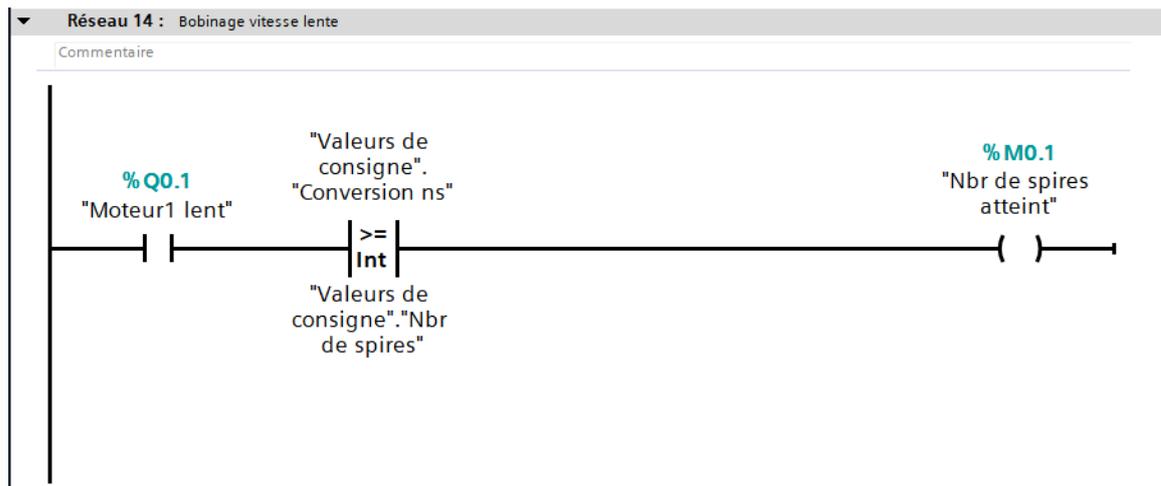


FIGURE 3.32 – Bobinage vitesse lente

3.6.4.2 Moteur vitesse rapide

- La figure 3.33 montre comment le memento `%M0.6` sert à activer la sortie (moteur 1 vitesse rapide) lorsque le déplacement est terminé, et le nombre de spires à effectuer est supérieur à 2, les deux contacts à ouverture qui viennent après servent à désactiver la sortie lorsque le bobinage est terminé, la partie en parallèle permet de redémarrer le moteur 1 en vitesse rapide en cas d'arrêt, plus le contact de maintien qui est en parallèle.

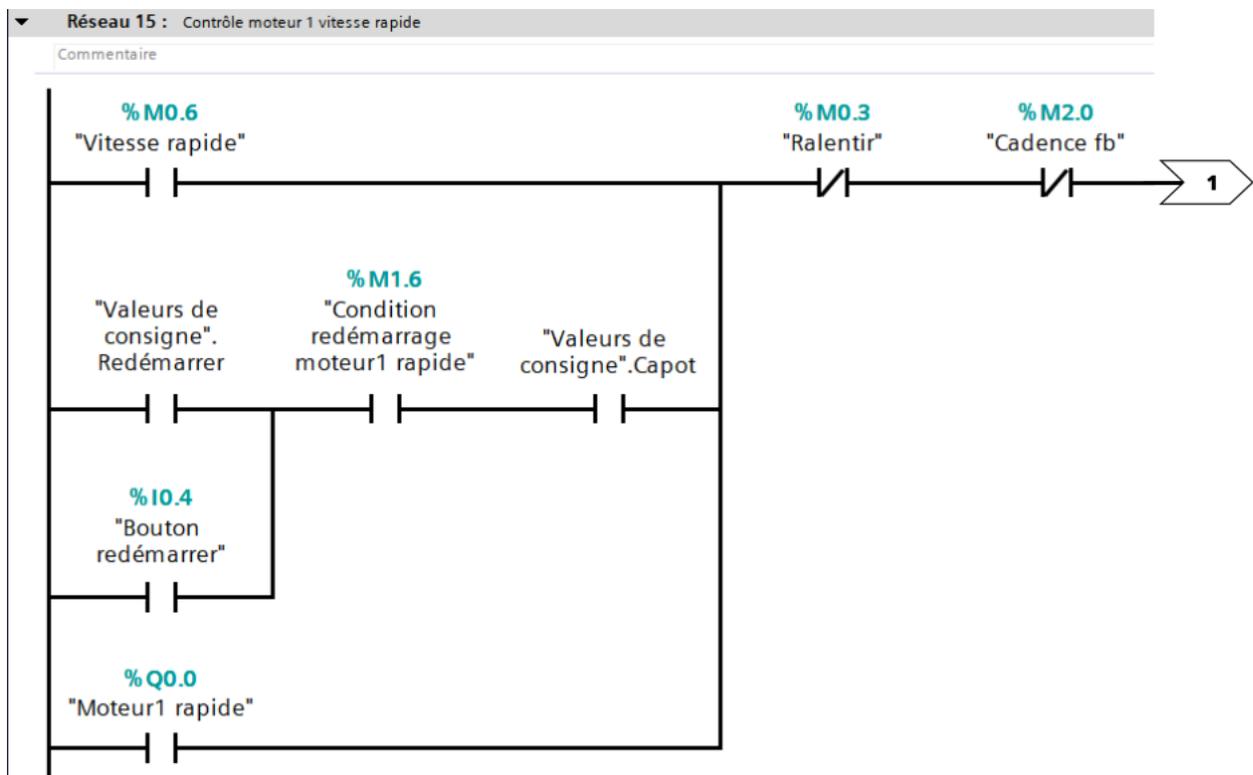


FIGURE 3.33 – Contrôle du moteur 1 vitesse rapide

- Juste après vient l'affectation de la sortie `%Q0.0` et la mise à 0 du memento `%M0.6` qui a activé la sortie `%Q0.0`, après assurer que la sortie `%Q0.2` est désactivé, plus la mise à 1 et à 0 des mementos de redémarrage, tout est représenté dans la figure 3.34

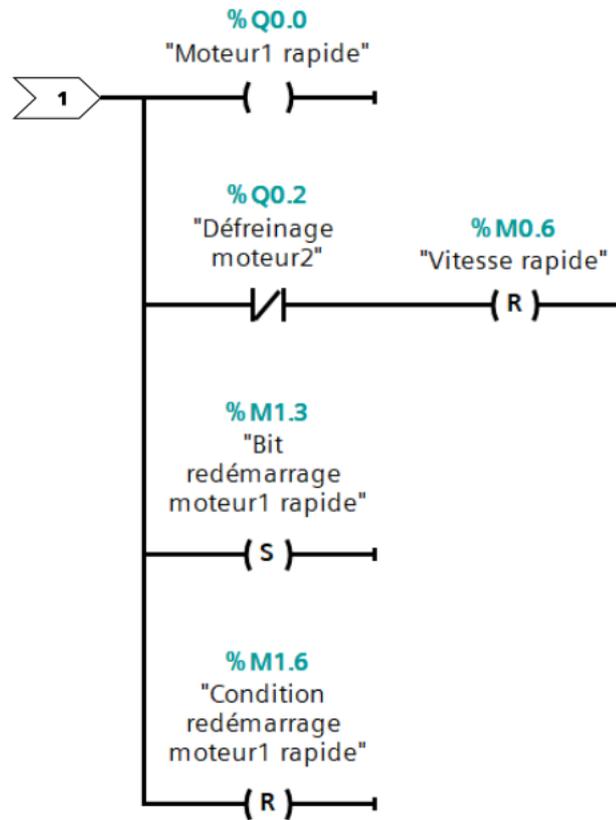


FIGURE 3.34 – Affectation

- La figure 3.35 montre une vue globale du réseau de contrôle de la sortie "moteur 1 vitesse rapide" :

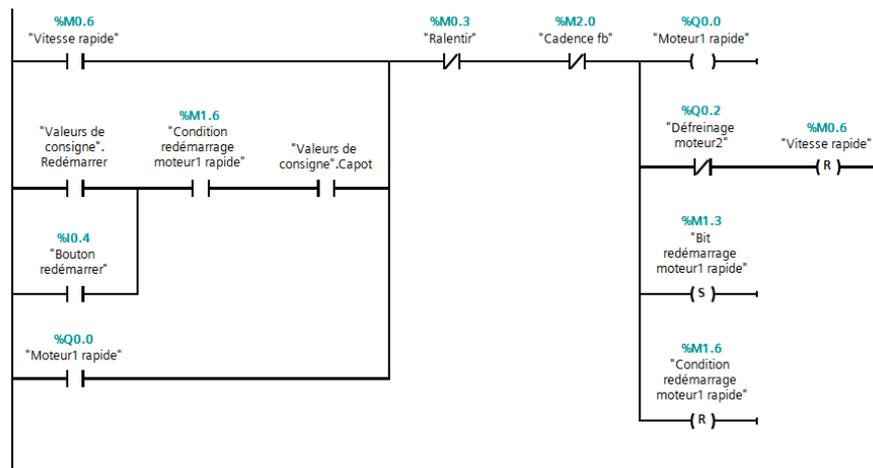


FIGURE 3.35 – Vue globale du réseau

- La figure 3.36 indique le réseau qui donne la possibilité de redémarrer le bobinage.

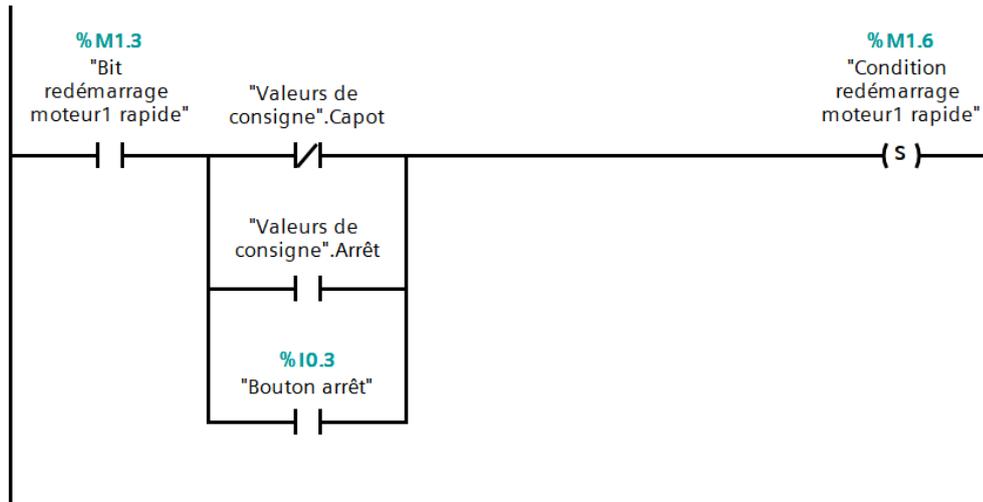


FIGURE 3.36 – Condition redémarrage moteur 1 vitesse rapide

- Le contact de comparaison, représenté sur la figure 3.37, assure l'activation de la sortie jusqu'à ce que la variable "Conversion ns" devienne supérieur ou égale à la variable "Nbr de spires - 2", c'est à dire le moteur 1 a bien complété le bobinage en vitesse rapide et le memento %M0.3 qui diminue la vitesse du moteur s'active.

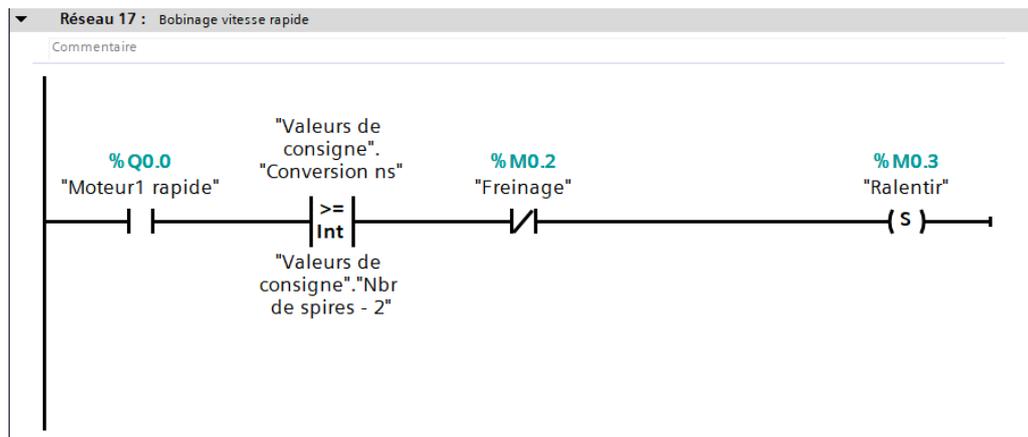


FIGURE 3.37 – Bobinage vitesse rapide

3.6.5 Comptage de l'opération

Dans notre projet on a choisit de faire deux comptage au cours du bobinage, un qui s'incrémente à chaque fois le bobinage d'un faisceau est terminé, et l'autre qui s'incrémente une fois quand le cycle est terminé.

Un cycle représente la complétion de plusieurs faisceaux, pour notre cas si la consigne de nombre de faisceaux est 3 la machine doit faire deux cycles, c'est à dire $2 \cdot 3 = 6$ faisceaux bobinés, mais dans le cas de 4 faisceaux la machine doit effectuer un seul cycle c'est à dire 4 faisceaux bobinés.

Cette configuration peut être généraliser sans devoir effectuer des changement au corps du programme, il suffit juste de modifier les réseaux représentés dans les figures 3.38 et 3.39.

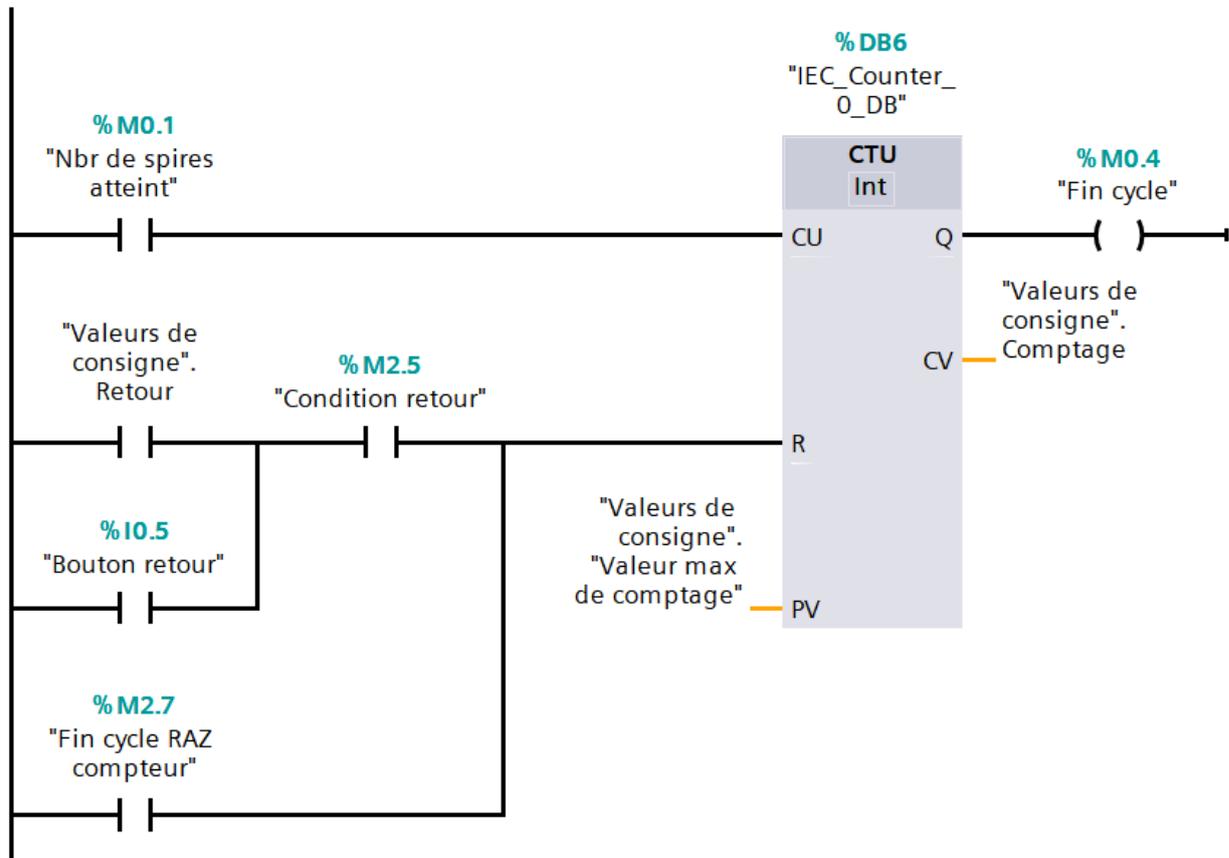


FIGURE 3.38 – Comptage des faisceaux

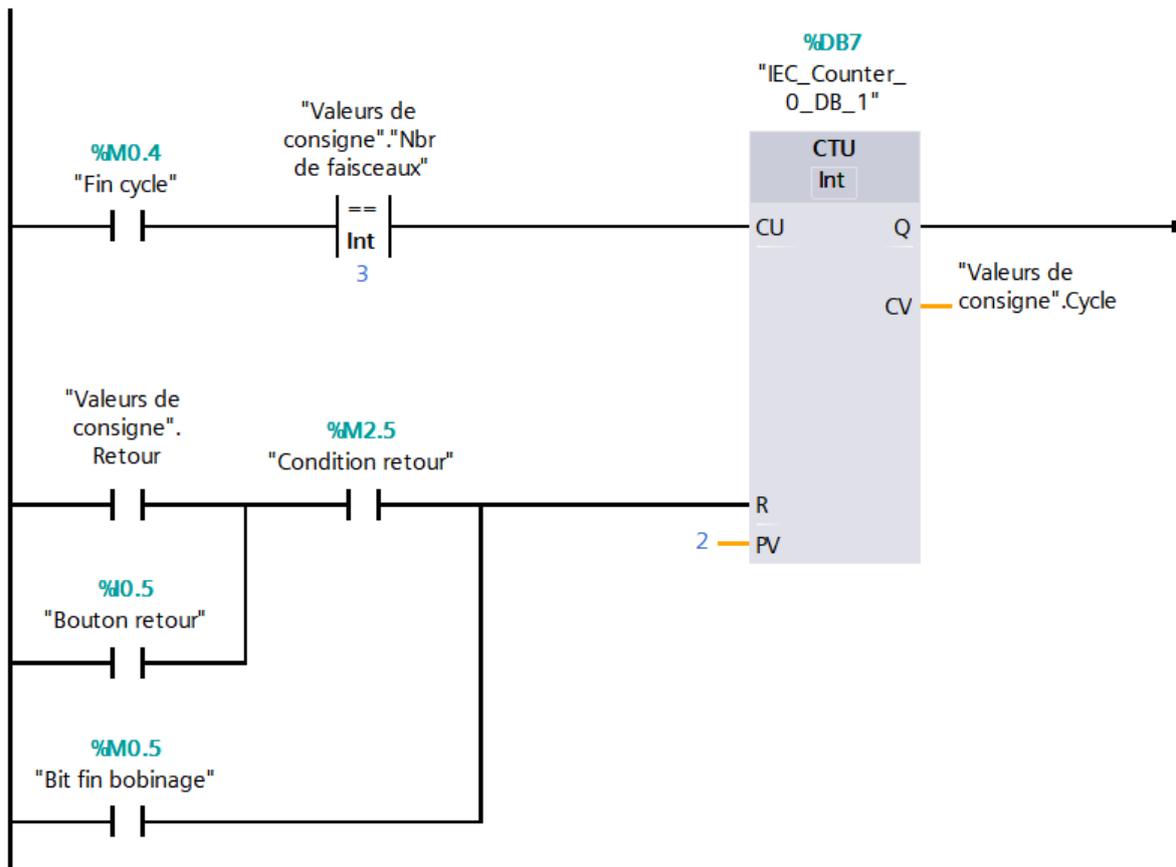


FIGURE 3.39 – Comptage du cycle

Un compteur à incréméntation se compose de :

- Entrée "CU" : le compteur s'incréménte quand le front montant du résultat logique de cette entrée devient 1.
- Remise à zero 0 "R" : le compteur se réinitialise quand le front montant du résultat logique devient 1.
- Valeur maximale de comptage "PV" : c'est là où on introduise la valeur où le compteur cesse de s'incréménte et la sortie "Q" du compteur s'active.
- Sortie du compteur "Q" : elle s'active quand le compteur atteint la valeur "PV".
- Comptage actuel "CV" : c'est la valeur de comptage à chaque incréméntation du début jusqu'à la fin.

Il faut tenir en compte que le compteur à incréméntation n'a rien avoir avec le compteur rapide, chacun a son rôle dans le programme et l'un ne peut jamais effectuer le travail de l'autre.

3.6.6 Fin du bobinage

Quand le bobinage est terminé il faut désactiver tous les mémentos et les sorties de commande, c'est le rôle de l'instruction de transfert "MOVE", représentée dans la figure 3.40.

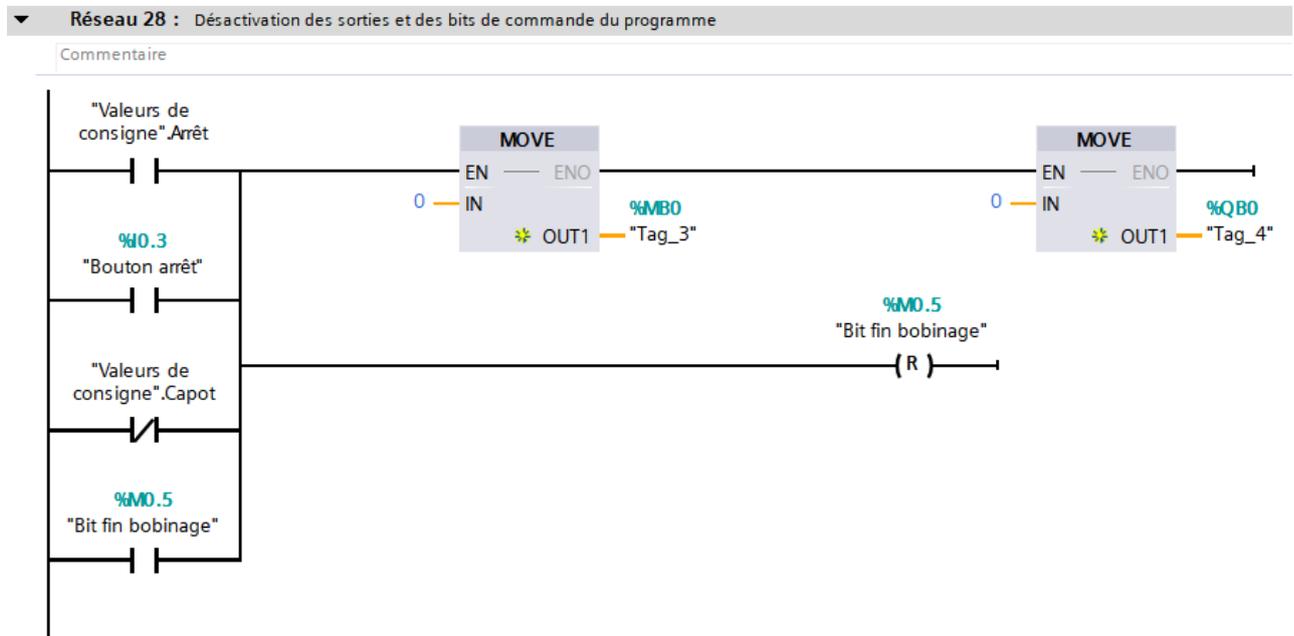


FIGURE 3.40 – Mise à 0 des sorties et mémentos de commande

A ce niveau on peut remettre le chariot-guide à sa position initiale en sélectionnant le bouton "Retour" qui active la sortie %Q0.4 (moteur 2 sens inverse), le contact de comparaison assure l'activation de la sortie jusqu'à que la variable "Conversion en mm" devienne supérieur ou égale à la variable "Déplacement total", et le memento %M2.2 sera mis à 1 indiquant l'arrivée à la position initiale, comme indique la figure 3.41.

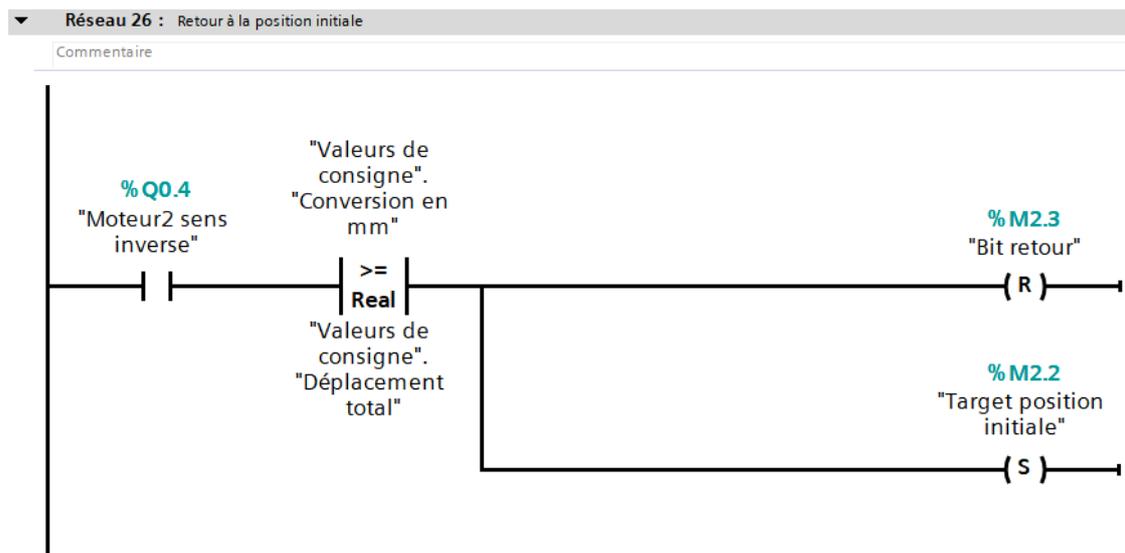


FIGURE 3.41 – Retour à la position initiale

Afin de pouvoir faire le retour il faut calculer la valeur du déplacement total que le chariot-guide a effectué, le réseau qui calcul cette variable est le suivant :

- D'abord on place les contacts nécessaires, indiqués sur la figure 3.42, pour assurer que l'instruction de calcul du déplacement total n'est effectué que lorsque le moteur de la vis-mère (en sens direct) est en marche, c'est à dire le chariot-guide est entrain de se déplacer d'un faisceau à l'autre.

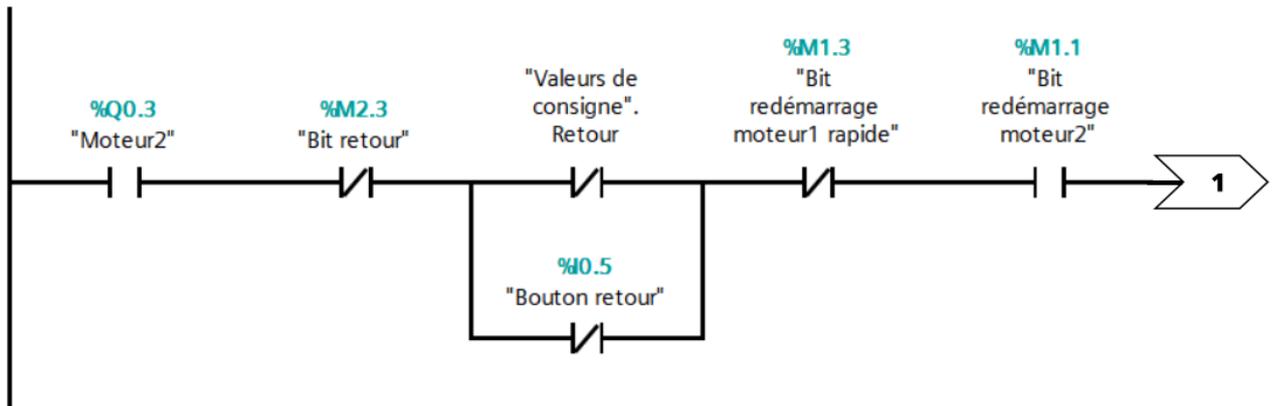


FIGURE 3.42 – Conditions de calcul du déplacement total

- Ensuite on met l'instruction "CALCULATE", comme indique la figure 3.43, avec l'opération mathématique : "Conversion en mm" + "Déplacement" "Comptage" = "Déplacement total"

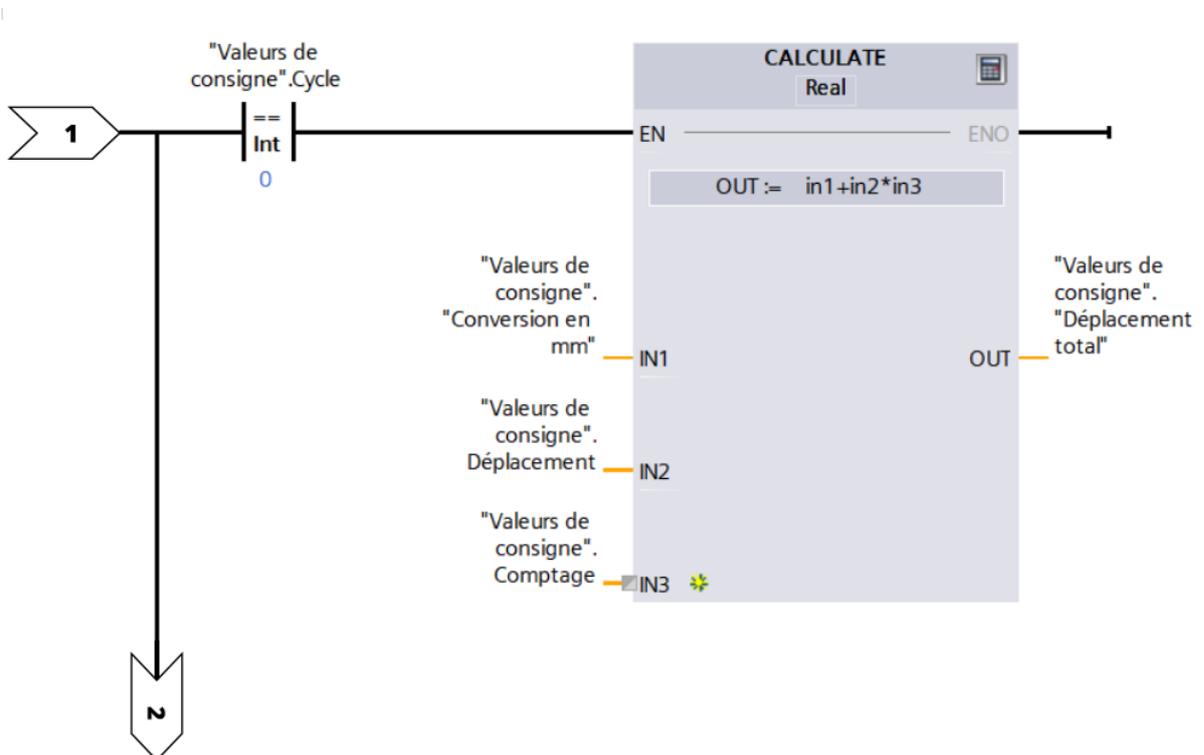


FIGURE 3.43 – Calcul du déplacement total dans le premier cycle

- Pour le deuxième cycle le premier compteur se réinitialise, ce qui nécessite l'ajout du déplacement effectuer au premier cycle, comme indique la figure 3.44, d'où l'opération mathématique suivante : "Conversion en mm" + "Déplacement" "Comptage" + "Déplacement" "Valeur max de comptage" = "Déplacement total"

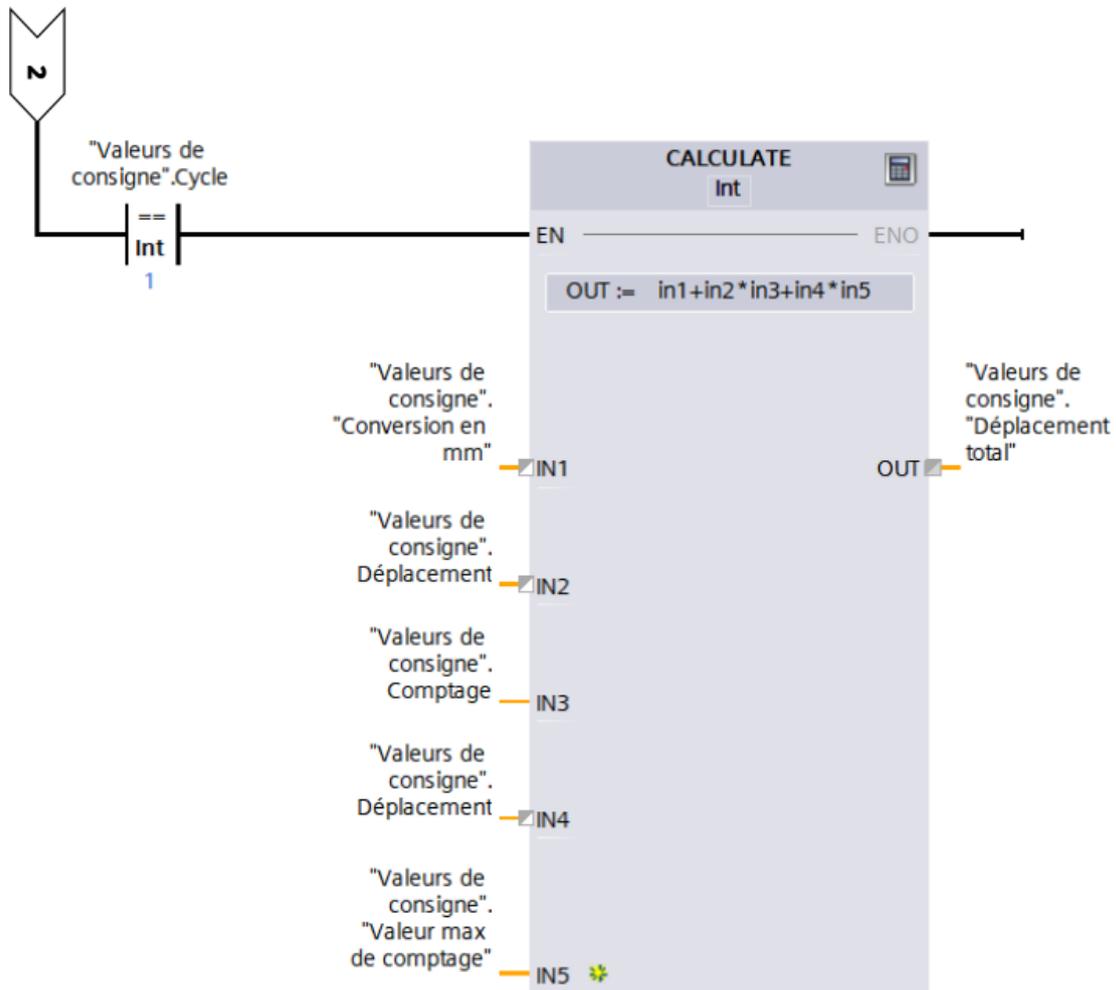


FIGURE 3.44 – Calcul du déplacement total dans le deuxième cycle

On a mis ce réseau au début du programme (réseau 4), car c'est important de prioriser le calcul du déplacement total afin que le calcul soit exact.

La figure 3.45 montre la vue générale de ce réseau.

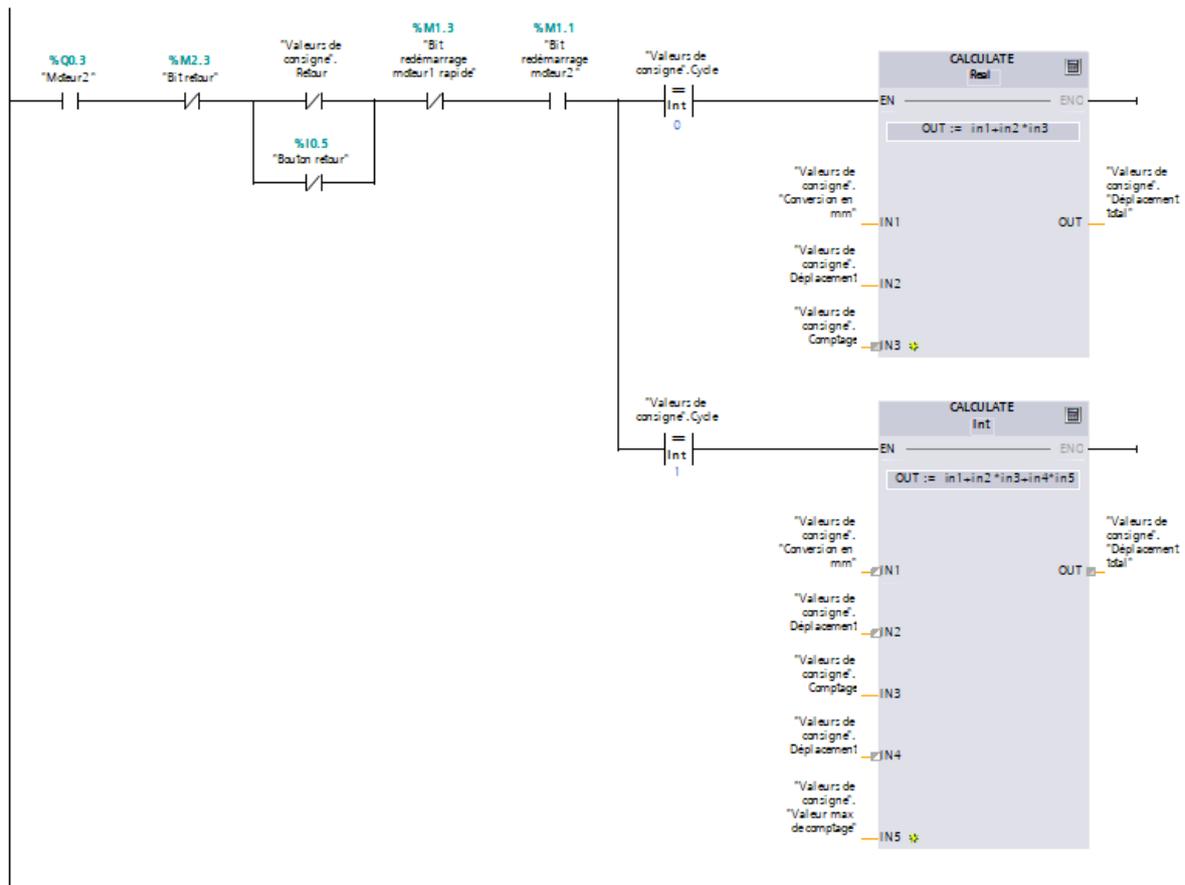


FIGURE 3.45 – Vue globale du réseau

3.6.7 Surveillance de la machine

Afin de superviser la machine et son bon fonctionnement, on a ajouté des alarmes qui s'enclenchent lorsque un défaut apparaît sur l'un des deux moteurs, alors on a déclaré la variable %MW3 de type "Int" dans le tableau des variables standards comme indique la figure 3.46, les bits de ce mot seront réservées aux défauts et on ne peut pas les utiliser dans d'autres fonctions.

	Défauts	Int	%MW3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Défaut moteur1 rapide	Bool	%M3.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Défaut moteur1 lent	Bool	%M3.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Défaut défreinage moteur2	Bool	%M3.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Défaut moteur2 sens direct	Bool	%M3.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Défaut moteur2 sens inverse	Bool	%M3.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Défaut capot ouvert	Bool	%M3.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

FIGURE 3.46 – Déclaration de la variable "Défauts"

On a mis les conditions de la mise à 1 des mémentos de défauts comme dans la figure 3.47 et la figure 3.48.

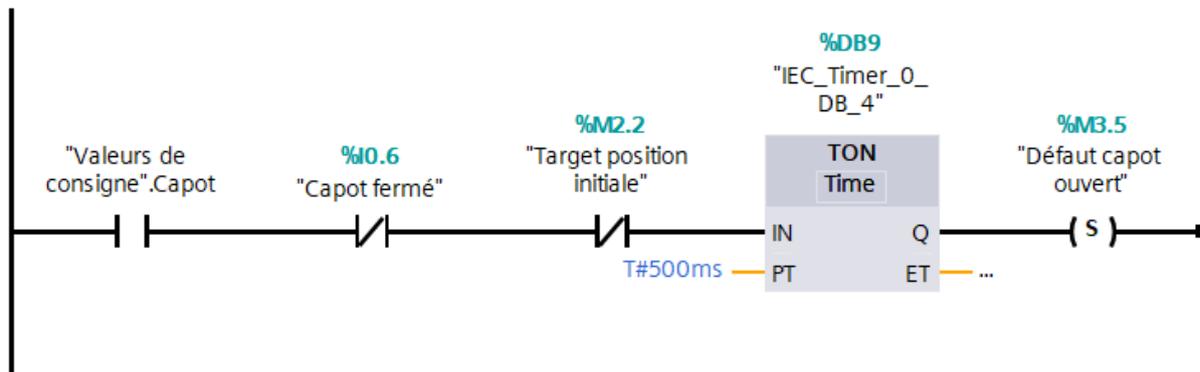


FIGURE 3.47 – Mise à 1 du memento %M3.5

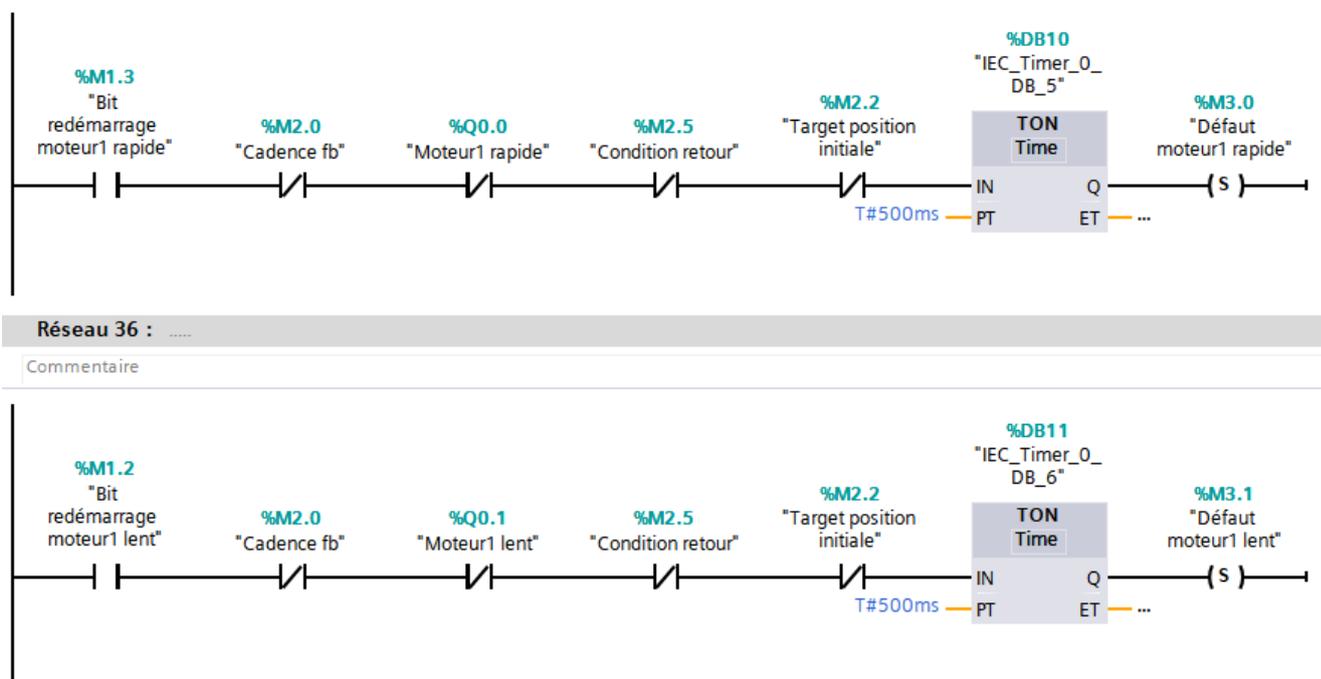


FIGURE 3.48 – Mise à 1 des deux mementos %M3.0 et %M3.1

Le reste des mementos sont regroupés dans un seul réseau avec les mêmes conditions, puis la mise à 1 de chaque bit de défaut, si par exemple le moteur 2 est en marche et il s'est arrêté au milieu du déplacement, le contact à ouverture (%Q0.3) s'active et le bit "Défaut moteur 2 sens direct" sera mis à 1, indiquant l'existence d'un défaut au niveau de ce moteur, comme montrent les figures 3.49 et 3.50.

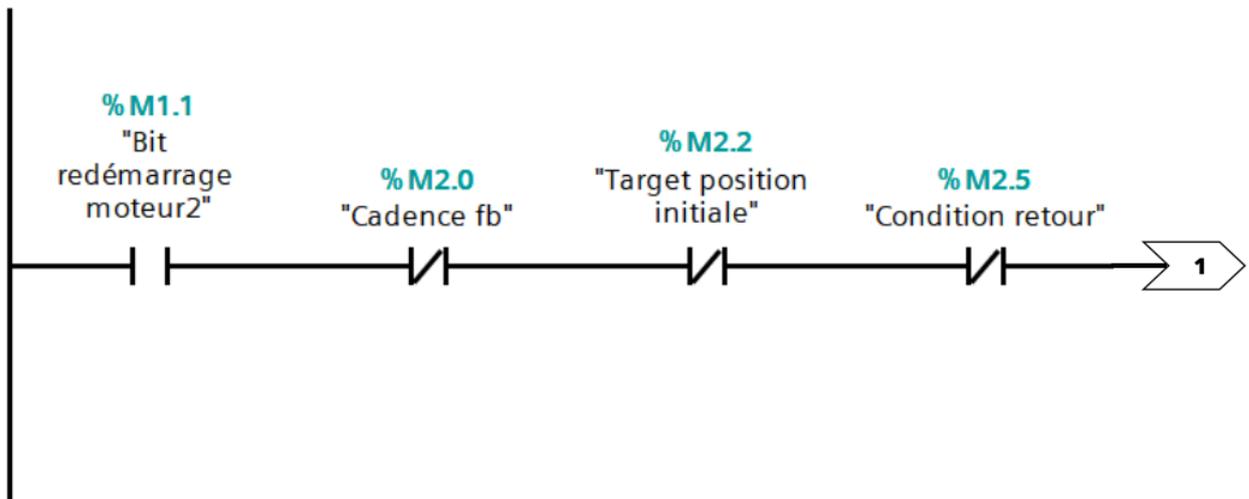


FIGURE 3.49 – Conditions de la mise à 1

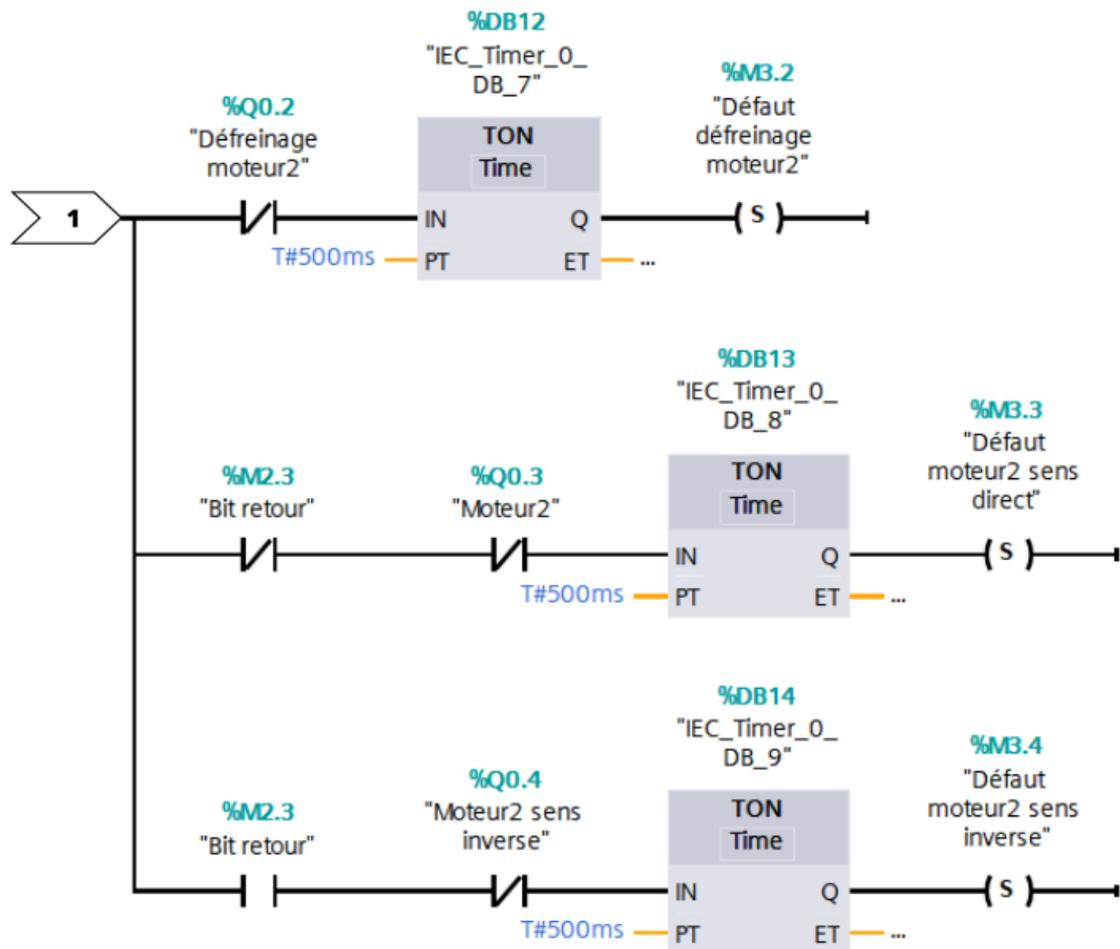


FIGURE 3.50 – Mise à 1 de chaque memento

Ensuite on va utiliser ces mémentos comme contacts à fermeture qui enclenchent les sorties représentant les alarmes, comme indique la figure 3.51.

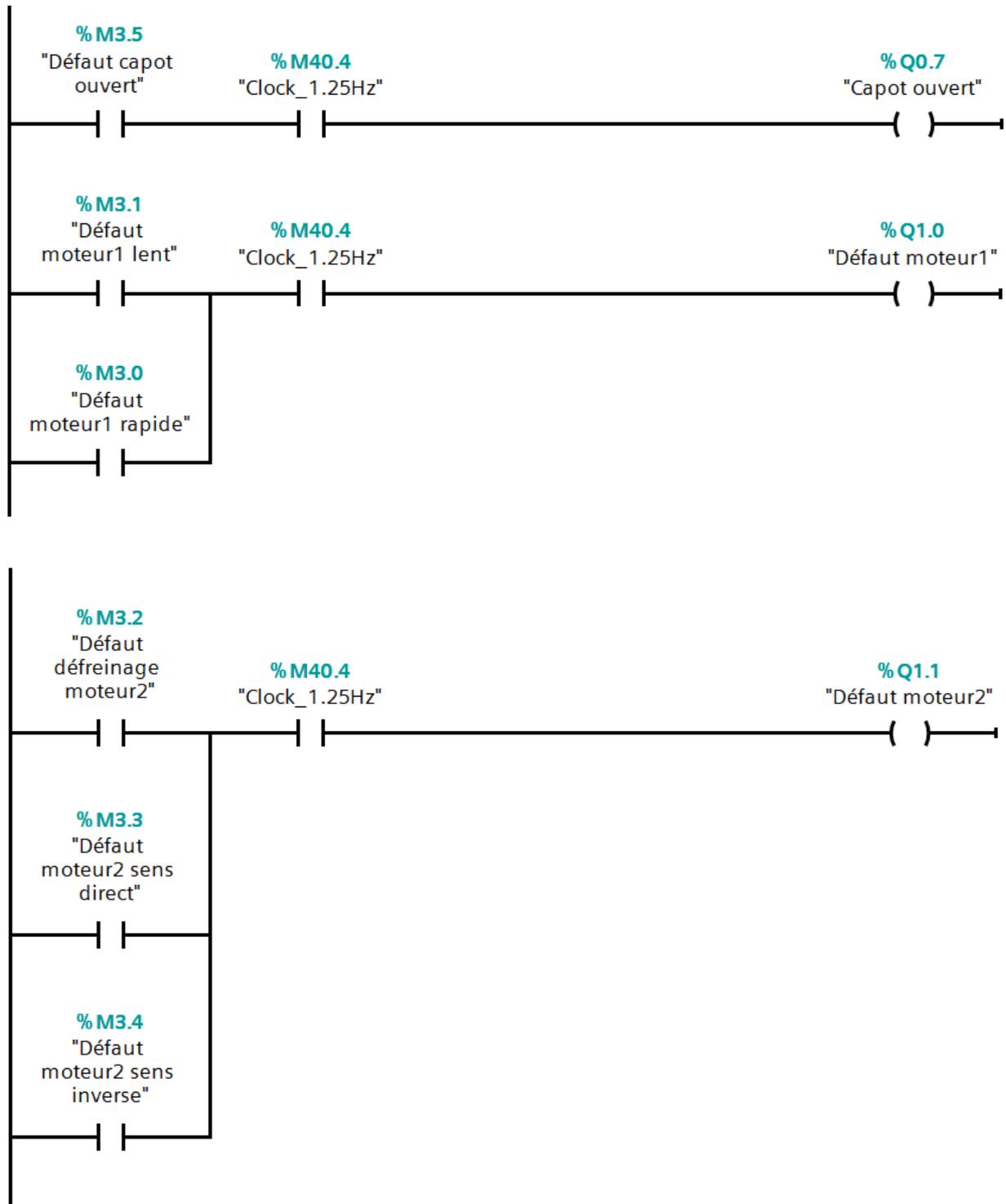


FIGURE 3.51 – Affectation des sorties des alarmes

Dans cette partie on a utilisé le memento de cadence %M40.4 pour ajouter l'effet de clignotement aux sorties des alarmes, ce mementos s'active 5 fois chaque 4 secondes.

Enfin si une alarme est enclenché on peut l'arrêter par le bouton "Arrêt alarmes" qui désactive par la mise à 0 "RESET" les mémentos de défauts mentionnés précédemment, la figure 3.52 montre le réseau qui effectue la désactivation.

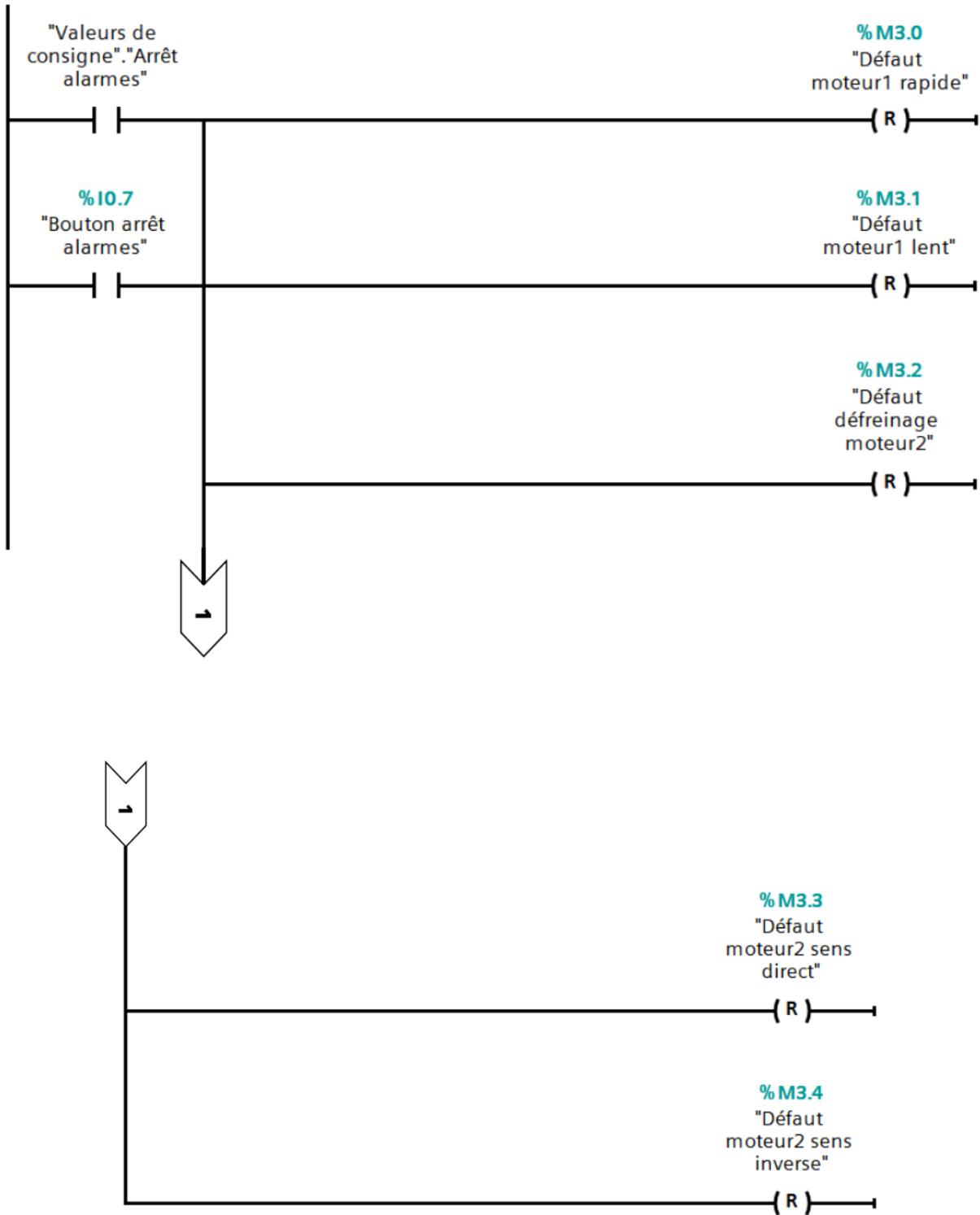


FIGURE 3.52 – Désactivation des alarmes

La figure 3.53 montre comment se fait la désactivation du memento %M3.5.



FIGURE 3.53 – Désactivation du "Défaut capot ouvert"

Remarque : le memento %M3.5 s'active quand le capot est ouvert accidentellement au milieu du bobinage.

3.6.8 Remise à zéro du compteur rapide

Dans ce projet on avait un seul encodeur pour compter la distance parcourue en millimètres et pour compter le nombre de spires effectuées, ce qui fait que ce programme est basé sur un seul encodeur, alors à chaque fois le déplacement d'un faisceau à l'autre ou le bobinage d'un faisceau sont terminés il faut remettre à zéro le compteur rapide qui compte les impulsions envoyées par l'encodeur.

La remise à zéro s'effectue par le bloc appelé "CTRL_HSC", indiqué dans la figure 3.54, qui se trouve dans Technologie > Compter, on sélectionne ce bloc et on remplit l'entrée du bloc nommée "HSC" par l'ID matériel que l'on trouve dans les propriétés du compteur rapide, comme indique la figure ci-après.

La seule entrée qu'on a besoin est "CV", elle remet à zéro le compteur rapide quand elle est activée, alors on met les conditions en parallèle liés à cette entrée pour assurer la remise à zéro du compteur à chaque fois un bobinage ou un déplacement sont terminés.

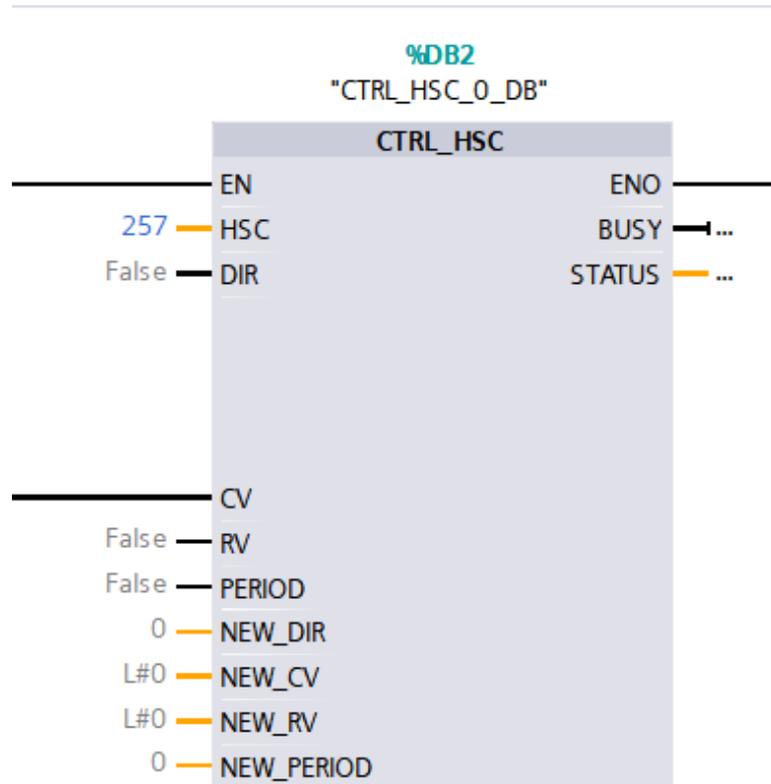


FIGURE 3.54 – Contrôle du compteur rapide

La figure 3.55 montre un exemple d'une condition de remise à zéro du compteur rapide.

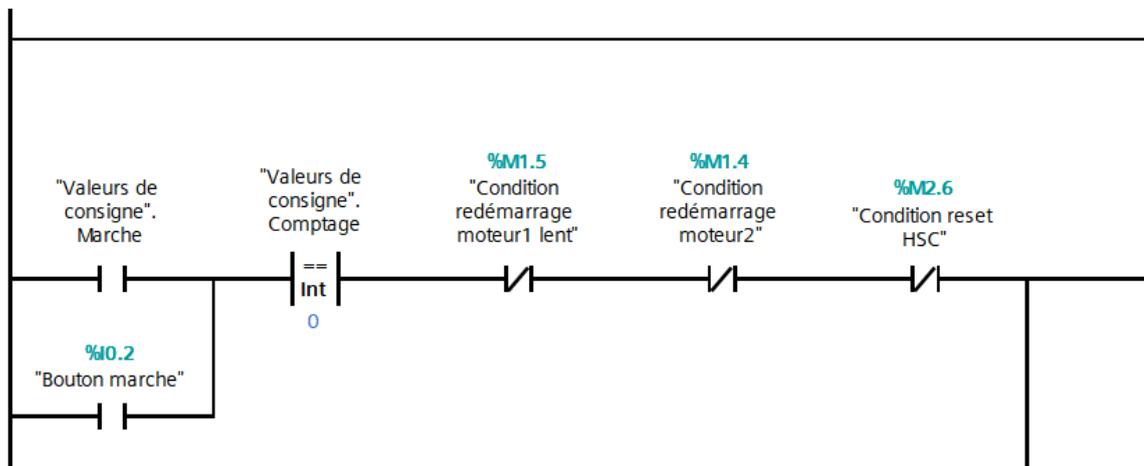


FIGURE 3.55 – Condition de remise à zéro du HSC

Il y a d'autres contacts en parallèle qui enclenche la remise à zéro, c'est les mémentos de commande du programme qui activent et désactivent automatiquement les sorties.

Conclusion

Dans ce chapitre on a programmé l'automate qui sera en mesure de remplacer la logique câblée, tout en donnant un nouveau fonctionnement de la machine basant sur la logique séquentielle et les principes de l'automatisation. Le prochain chapitre élabore la configuration de l'écran tactile (interface homme-machine) qu'on va l'utiliser pour commander et superviser l'automate et la machine bobineuse.

Chapitre 4

Configuration de l'écran tactile

Introduction

Lorsque les machines et les installations doivent satisfaire à des spécifications de fonctionnalité de plus en plus strictes, il est essentiel de fournir à l'opérateur un niveau élevé de transparence. Cette transparence est obtenue grâce à l'utilisation d'une Interface Homme-Machine (IHM). Une IHM est un ensemble de vues préalablement créées et configurées à l'aide d'un logiciel approprié, qui sont ensuite présentées à l'opérateur.

Au cours de ce chapitre, on a créé une plateforme de supervision en utilisant le logiciel WinCC. Cette plateforme permettra de visualiser l'état de fonctionnement du procédé de bobinage, le commander à distance et détecter d'éventuels défauts survenant pendant son fonctionnement.

1 Définition de la supervision

La supervision est une forme avancée d'interaction entre l'homme et la machine. Elle consiste à surveiller l'état de fonctionnement d'un processus, offrant des capacités bien plus étendues que celles des fonctions de conduite et de surveillance réalisées par les interfaces classiques. Les fonctions de la supervision sont multiples, voici quelques exemples :

- Assure la communication entre les équipements d'automatisation et les outils informatiques pour la planification et la gestion de la production.
- Assiste l'opérateur dans les opérations de diagnostic et de maintenance.
- Coordonne le fonctionnement d'un ensemble de machines enchaînées constituant une ligne de production, en assurant l'exécution d'ordres communs tels que la marche, l'arrêt, et en gérant des tâches telles que la synchronisation.
- Fournit en temps réel une visualisation de l'évolution des paramètres du processus, ce qui permet à l'opérateur de prendre rapidement des décisions adaptées à ses objectifs.
- Répond à des besoins qui nécessitent généralement une puissance de traitement importante.

2 Avantages de la supervision

La supervision d'un système automatisé présente plusieurs avantages significatifs :

- **Surveillance continue à distance** : La supervision permet une surveillance continue de l'état et des performances du système automatisé, sans devoir être proche au système.

- **Détection précoce des problèmes** : En surveillant en temps réel les paramètres et les indicateurs clés du système, la supervision permet de détecter les problèmes potentiels avant qu'ils ne se transforment en problèmes majeurs.
- **Optimisation des performances** : La supervision permet d'optimiser les performances du système automatisé en identifiant les goulots d'étranglement, les inefficacités ou les zones d'amélioration.
- **Prise de décision éclairée** : Grâce à la visualisation en temps réel des données et des informations sur l'état du système, la supervision permet aux opérateurs et aux décideurs de prendre des décisions éclairées.
- **Amélioration de la maintenance** : La supervision facilite la maintenance préventive en fournissant des informations sur l'état des équipements et des composants du système.

3 Interface homme machine

L'IHM (Interface Homme-Machine) est une plateforme qui permet aux utilisateurs d'interagir avec un système informatique ou une application. Elle agit comme un pont entre l'utilisateur et la machine, facilitant la communication et l'échange d'informations.

L'IHM englobe différents éléments, tels que les menus, les boutons, les icônes, les fenêtres, les boîtes de dialogue et les éléments interactifs, qui permettent à l'opérateur d'effectuer des actions et de recevoir des informations de l'automate.

Elle a donc la capacité de remplacer les stations de boutons, c'est à dire la possibilité de commander l'automate par cette plateforme, bien que visualiser les états des entrées/sorties et afficher les alarmes du système.

4 Choix de l'Interface Homme-Machine

SIMATIC HMI de l'entreprise SIEMENS propose une vaste gamme d'écrans tactiles pour toutes les applications des différentes machines, il existe dans cette gamme plusieurs types de pupitres programmables via trois logiciels : Protocol, WinCC et WinCC flexible.

Dans ce projet on a choisi d'utiliser le logiciel WinCC flexible pour la configuration d'un écran tactile de type KTP400 basic, qui était disponible à l'entreprise Electro-Industries.

5 WinCC flexible

WinCC flexible est un logiciel développé par Siemens qui fait partie de l'environnement d'ingénierie TIA Portal (Totally Integrated Automation). Il est spécifiquement conçu pour la création et la configuration d'interfaces homme-machine (IHM) pour les systèmes automatisés.

WinCC flexible offre des fonctionnalités avancées pour la conception et la visualisation d'IHM conviviales et interactives. Il permet aux utilisateurs de créer des écrans graphiques, des interfaces utilisateur et des vues personnalisées pour surveiller et contrôler les processus automatisés. Le logiciel prend en charge une variété d'éléments visuels tels que les boutons, les menus, les listes déroulantes, les graphiques, les tableaux, les alarmes et les tendances.

L'une des caractéristiques principales de WinCC flexible est sa flexibilité. Il permet aux utilisateurs de créer des IHM adaptées à leurs besoins spécifiques en utilisant des fonctions de configuration conviviales. Les utilisateurs peuvent facilement ajouter, modifier et configurer des objets graphiques, des états, des alarmes et des actions.

WinCC flexible offre également des fonctionnalités de connectivité étendues. Il peut se connecter à différents types de contrôleurs et d'appareils d'automatisation, tels que les automates programmables industriels (API), les variateurs de fréquence et les dispositifs de surveillance. Il prend en charge les protocoles de communication courants tels que Profibus, Profinet et Ethernet.

Le logiciel propose également des fonctionnalités avancées telles que la gestion des recettes, l'enregistrement des données, la surveillance à distance et la visualisation en temps réel des informations de production. Il facilite l'intégration de l'IHM avec d'autres outils d'automatisation et de gestion de production, permettant une meilleure interaction entre les différents composants du système.

En résumé, WinCC flexible de TIA Portal de Siemens est un logiciel puissant pour la création et la configuration d'IHM dans les systèmes automatisés. Il offre une grande flexibilité, des fonctionnalités avancées et une connectivité étendue, facilitant la surveillance, le contrôle et l'optimisation des processus industriels.

Par la suite on va élaborer la configuration de l'IHM réalisé par ce logiciel dans le TIA Portal, à l'intérieur du même projet où on a construit notre programme Ladder sur main [OB1], cette configuration se compose de trois vues, une table de variables standards et une table des alarmes IHM.

6 Configuration de l'IHM

4.6.1 Ajout de l'interface

- Sur le navigateur du projet on sélectionne "ajouter un appareil", dans la fenêtre qui s'ouvre on choisit IHM -> SIMATIC Basic Panel -> 4" Écran -> KTP400 Basic -> 6AV2 123-2DB03-0AX0 c'est la référence qu'on trouve sur l'appareil, puis on clique "OK", comme sur la figure 4.1

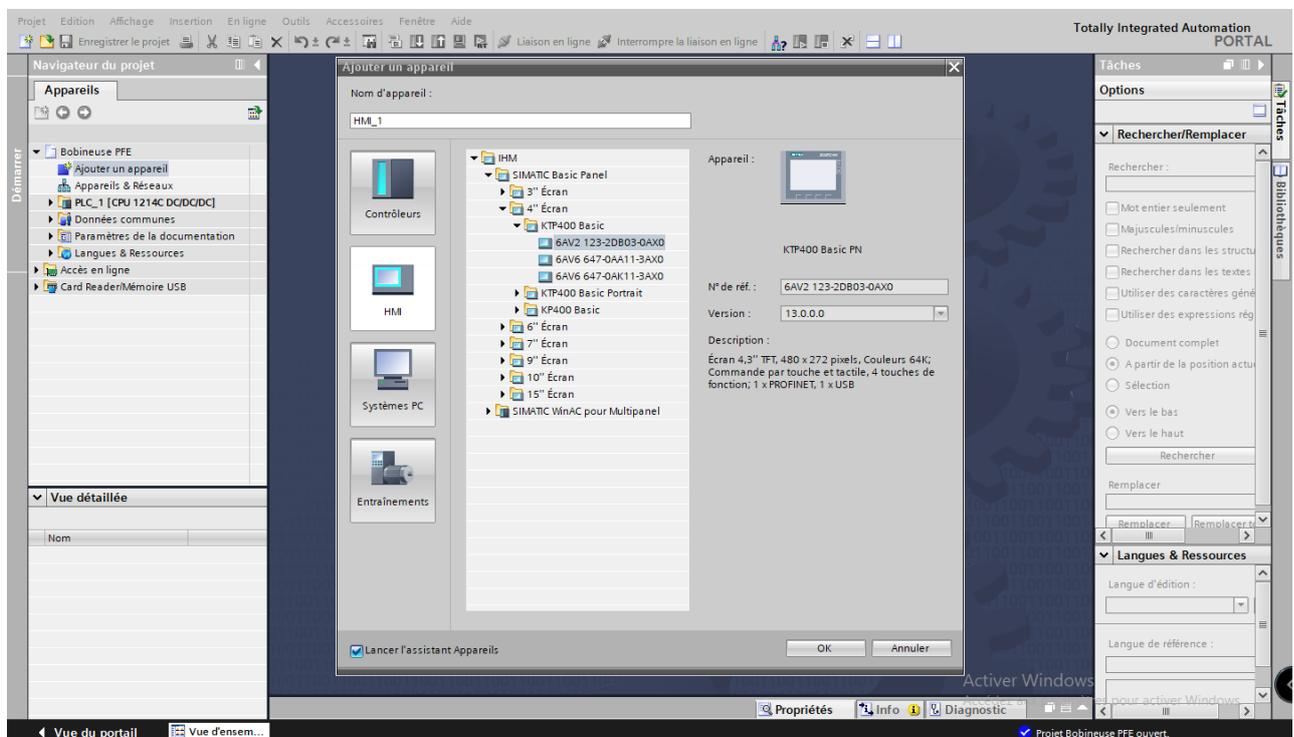


FIGURE 4.1 – Ajout de l'interface

- On met l'IHM en liaison avec l'automate, la figure 4.2 montre la fenêtre de connexion API

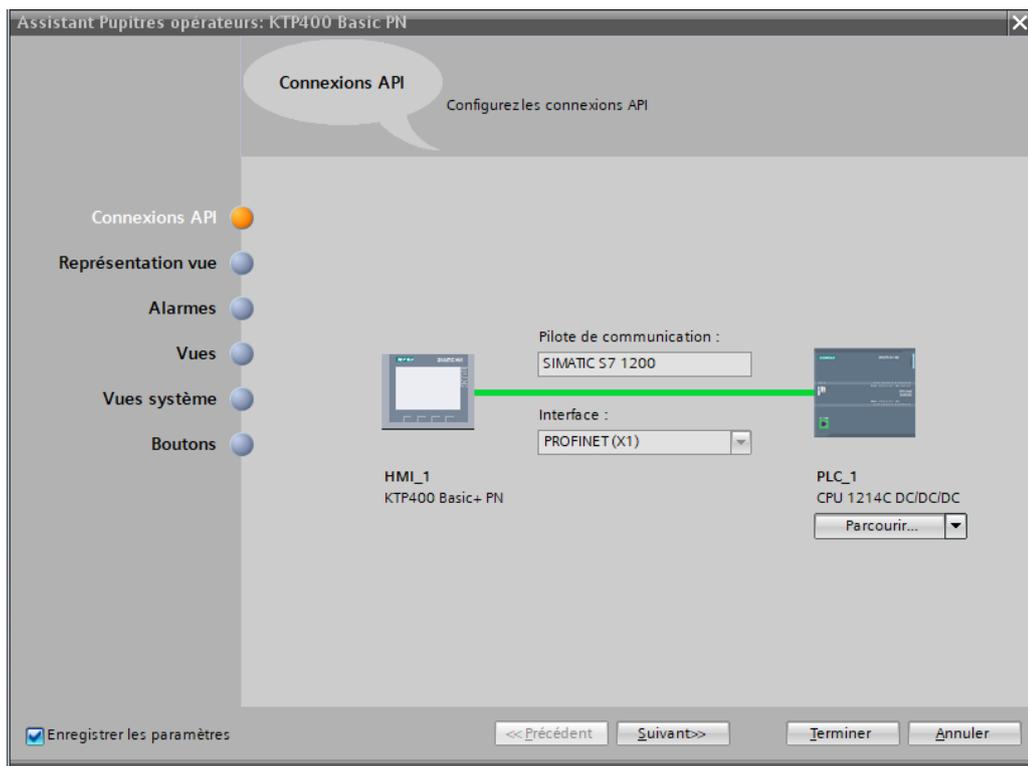


FIGURE 4.2 – Connexion API

- On sélectionne "Alarmes" à gauche de la fenêtre, et désactive les options d'alarmes, comme indique la figure 4.3, et on clique sur terminer

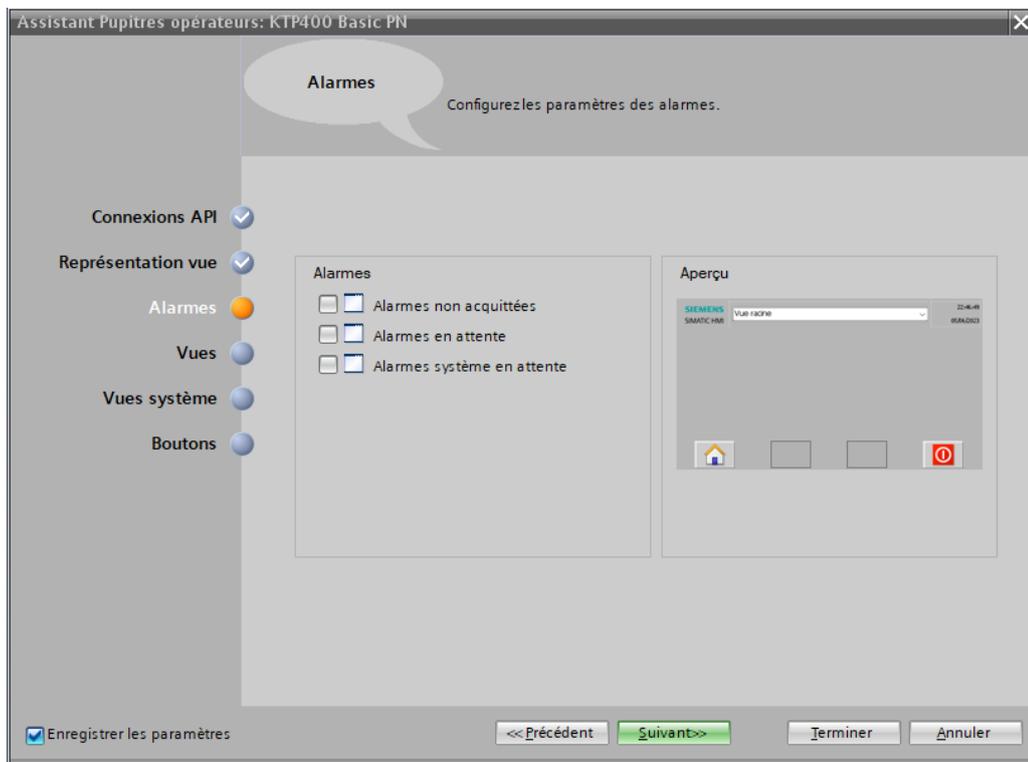


FIGURE 4.3 – Paramètres des alarmes

- Finalement, il est nécessaire de vérifier que l'adresse Ethernet de l'IHM ajouté dans notre programme, est compatible avec celle de l'interface actuelle qu'on va utiliser en temps réel, cela peut se faire à partir de la fenêtre des propriétés de l'IHM, qu'on peut sélectionner dans "Appareils et réseaux", dans le navigateur du projet, comme indique la figure 4.4.

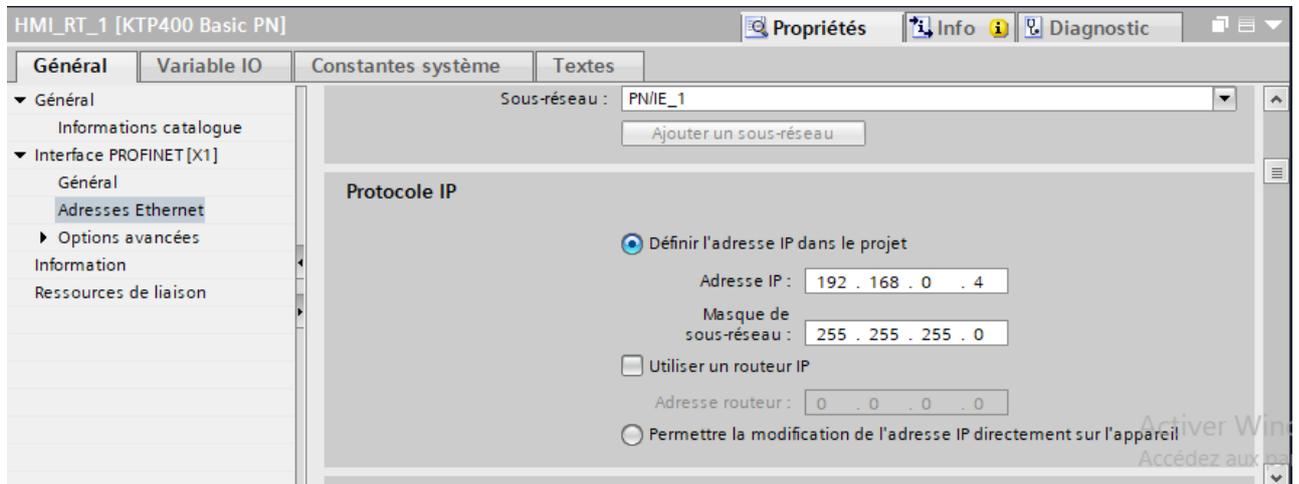


FIGURE 4.4 – Adresse Ethernet

4.6.2 Configuration des vues

Une fois qu'on a ajouté l'IHM, on commence de personnaliser les différentes vues de notre écran tactile, la figure 4.5 montre la vue racine initialement.

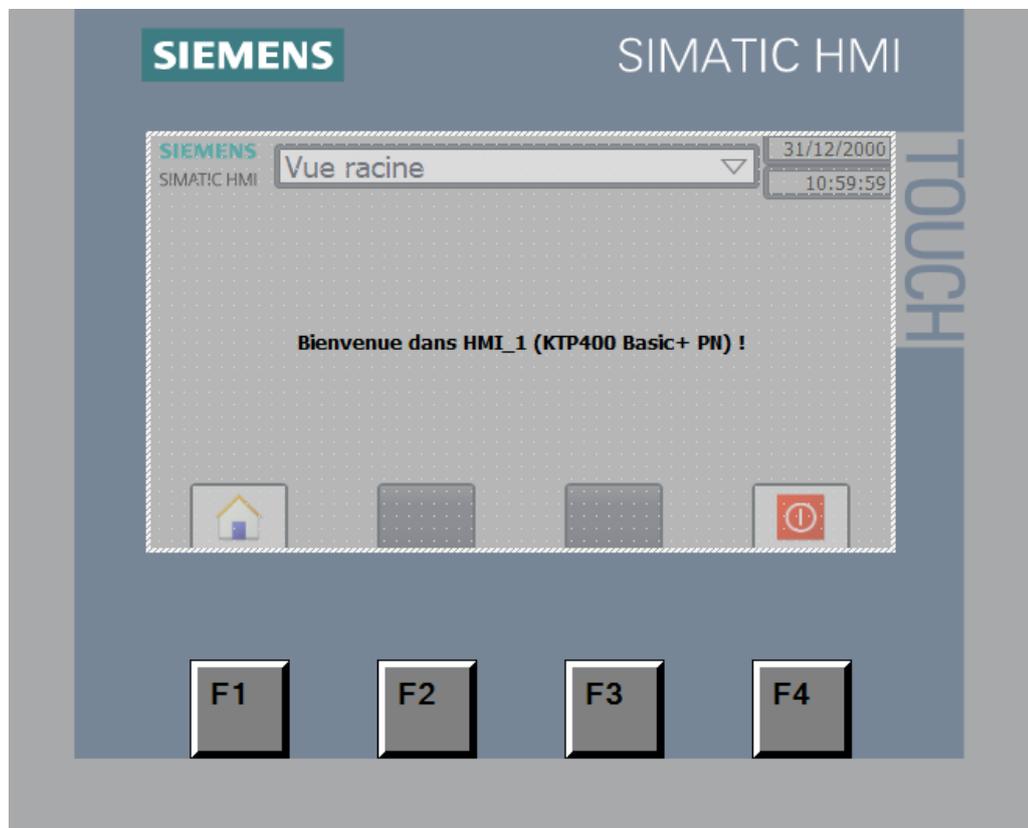


FIGURE 4.5 – Vue racine avant la configuration

- **Vue racine** : Sur la vue racine on ajoute tous les boutons qui contrôlent les actions de la machine (marche, arrêt, redémarrer,...etc.), comme dans la figure 4.6, le bouton se trouve à droite dans le groupe "Éléments", sauf pour les boutons "Vue des consignes" et "Vue des alarmes", qu'on peut les ajoutés d'après le navigateur du projet.

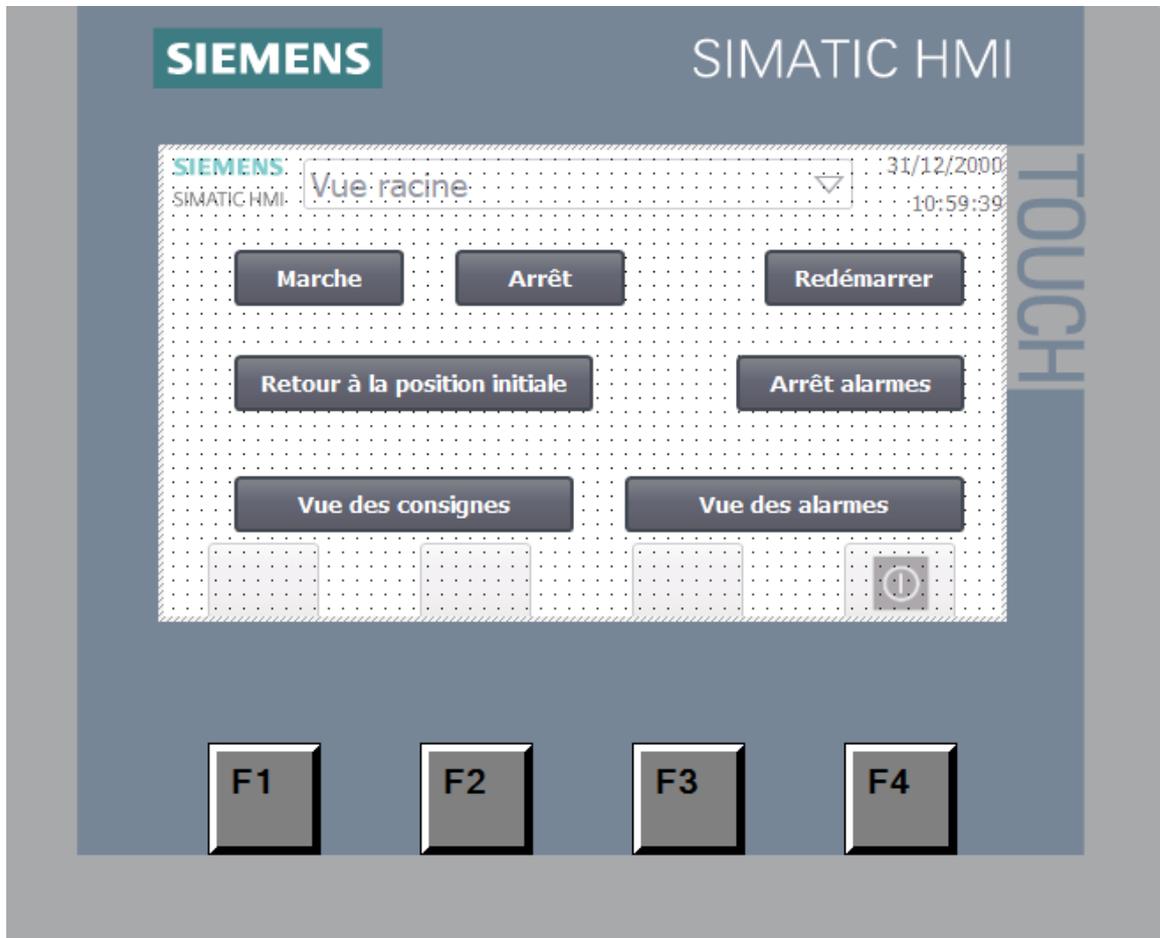


FIGURE 4.6 – Vue racine

Pour configurer un bouton, on sélectionne l'évènement qui se trouve dans les propriétés de ce bouton, comme indique la figure 4.7, puis on désigne la variable qui sera activer en appuyant sur ce bouton, et désactiver en relâchant le bouton.

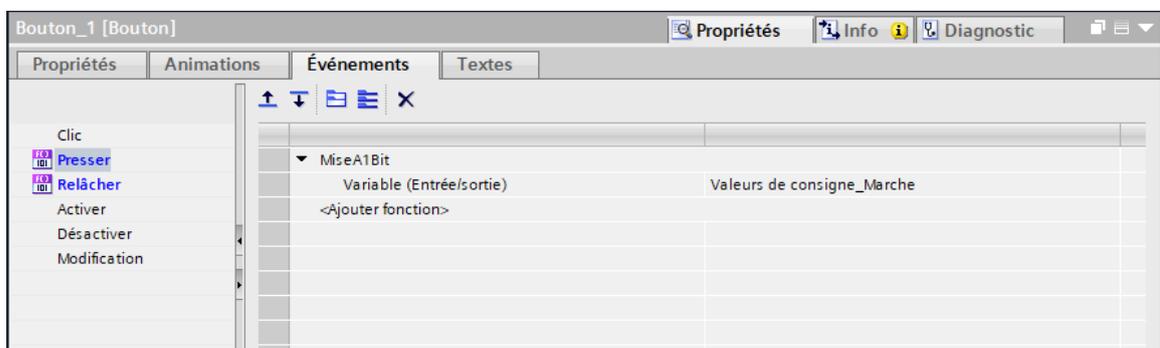


FIGURE 4.7 – Évènement "Presser" du bouton "Marche"

La figure 4.7 montre comment on a pu configurer l'évènement "Presser" du bouton "Marche", après avoir sélectionné l'évènement "Presser", on ajoute une fonction, puis on choisit l'action "MiseA1Bit" et on ajoute la variable "Marche" à partir du bloc de données.

On définit de la même façon l'évènement "Relâcher", sauf qu'on choisit l'action "RAZBit" au lieu de "MiseA1Bit", comme dans la figure 4.8, ce paramétrage permet d'activer la variable "Marche" (c'est à dire sa valeur logique est 1), quand on appuie sur ce bouton, mais dès qu'on relâche le bouton, la variable se désactive, donnant la valeur logique 0.

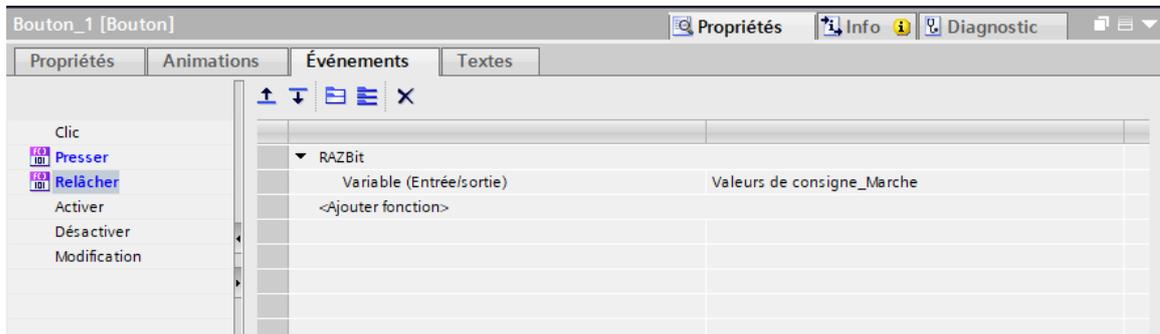


FIGURE 4.8 – Évènement "Relâcher" du bouton "Marche"

Il faut répéter cette procédure de configuration pour les autres boutons, afin d'avoir la vue racine près pour utilisation par l'opérateur, avec chaque bouton qui commande une variable spécifique donc contrôle les actions majeurs de la machine.

- **Vue des consignes :** Par la suite on ajoute une nouvelle vue, à partir du dossier "Vues" qui se trouve dans le navigateur du projet, on l'appelle "Vue des consignes" et on met dans cette vue le bouton "Vue racine", pour qu'on puisse retourner à la vue racine.

La figure 4.9 illustre la vue des consignes, avec les différents champs d'entrée/sortie ajoutés à partir du groupe "Éléments", ces derniers représentent les consignes de bobinage qu'il faut établir, plus les variables qui représentent la progression du bobinage, cette vue alors sert à paramétrer et superviser le travail de la machine.

Chaque champ d'entrée/sortie sert à donner une valeur quelconque à sa variable qui lui est attribuée, selon le type de celle-ci, car pour une variable entière on ne peut pas entrer un nombre avec la virgule, on peut définir les valeurs des consignes qui sont : "Consigne1", "Consigne2", "Consigne3", "Consigne4", "Déplacement" et "Nbr de faisceaux", mais on ne doit pas modifier les valeurs des variables : "Conversion en mm", "Déplacement total", car elles représentent l'état d'avancement du bobinage en temps réel.

Rappel : il faut tenir compte que le nombre de faisceaux dans ce cas peut être 3 ou 4 seulement, à cause du gabarit de bobinage de la machine actuelle qui est composé de deux parties, une qui contient des groupes de 3 faisceaux et l'autre qui contient des groupes de 4 faisceaux.

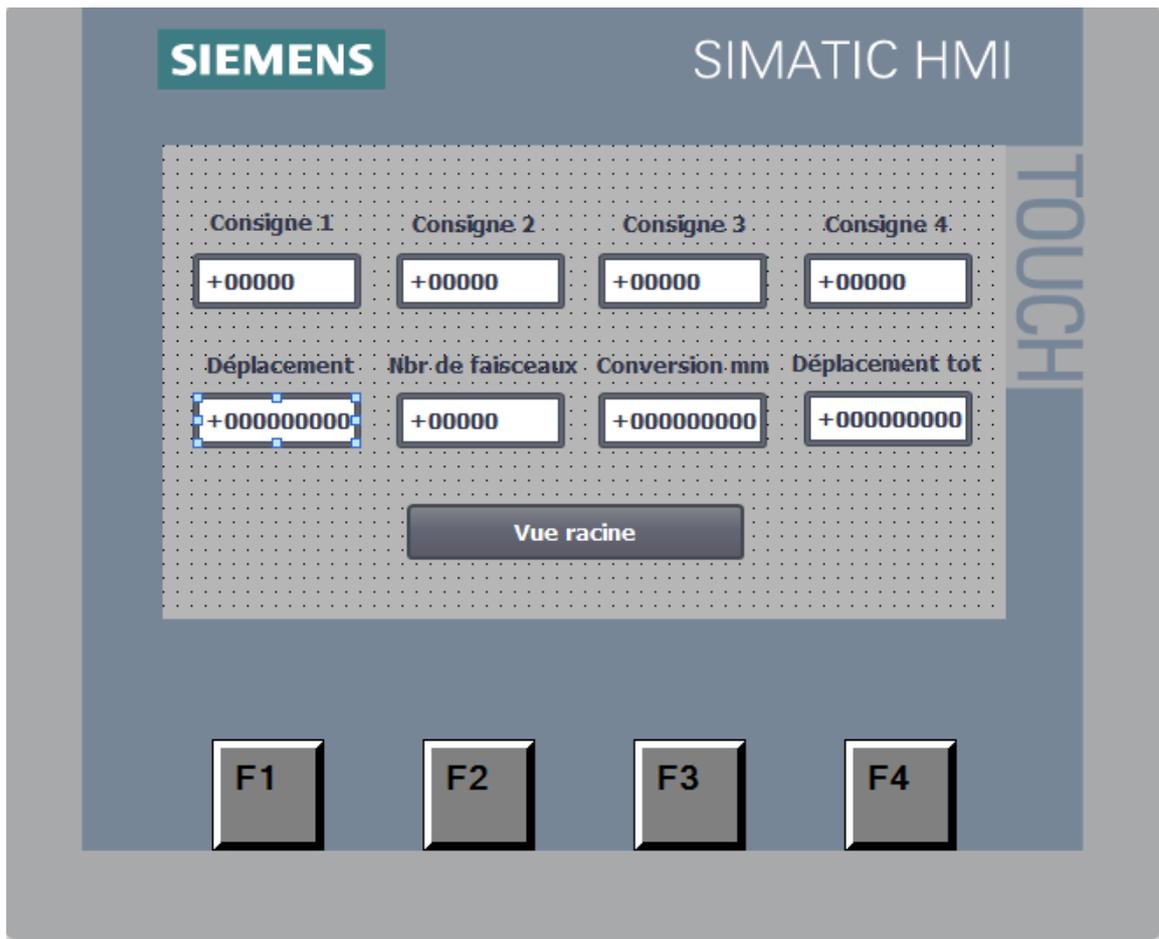


FIGURE 4.9 – Vue des consignes

La configuration d'un champ d'entrée/sortie est très simple, il suffit juste de sélectionner "Attributs" dans le menu "Propriétés" de ce champ, puis ajouter la variable comme la figure 4.10 l'indique.

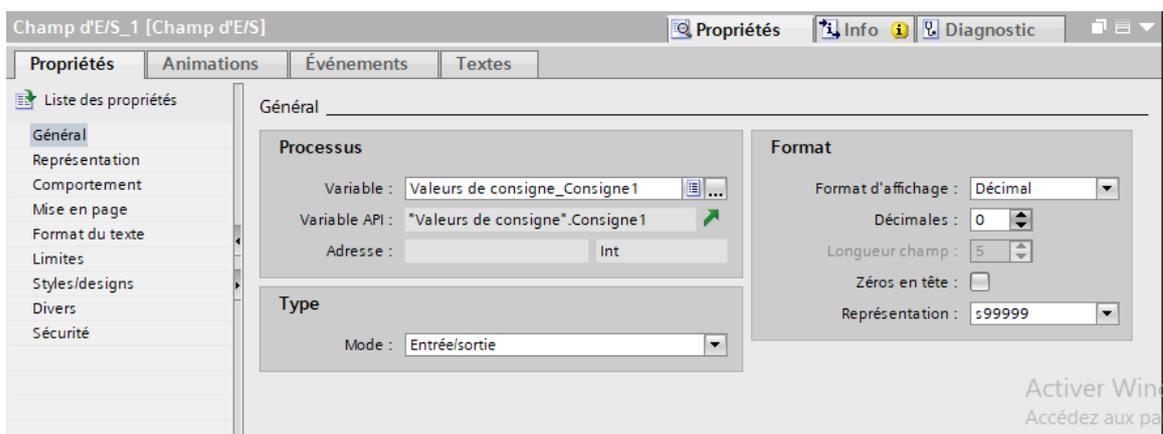


FIGURE 4.10 – Configuration du champ d'E/S de la consigne du faisceau 1

Il faut juste confirmer que le format d'affichage est décimale et le mode du champ est "Entrée/sortie".

On programme le reste des champs d'entrée/sortie de la même façon, la figure 4.11 montre le paramétrage du champ d'entrée/sortie de la variable "Déplacement" qui est de type "Real" contrairement à la variable précédente.

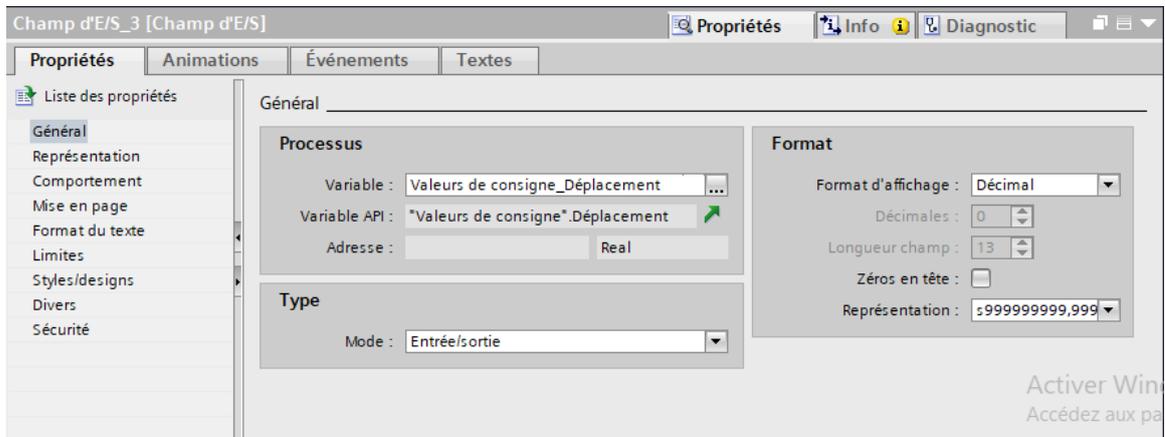


FIGURE 4.11 – Configuration du champ d'E/S de la distance entre deux faisceaux

- **Vue des alarmes** : Dernièrement on va ajouter la vue la plus simple appelée "Vue des alarmes", cette vue, représentée sur la figure 4.12, sert à visualiser les alarmes avec un texte explicatif, qui informe l'opérateur sur la cause de cette alarme, on doit juste ajouter la table "Vue des alarmes" qui se trouve dans le groupe "Contrôles" et le bouton "Vue racine".

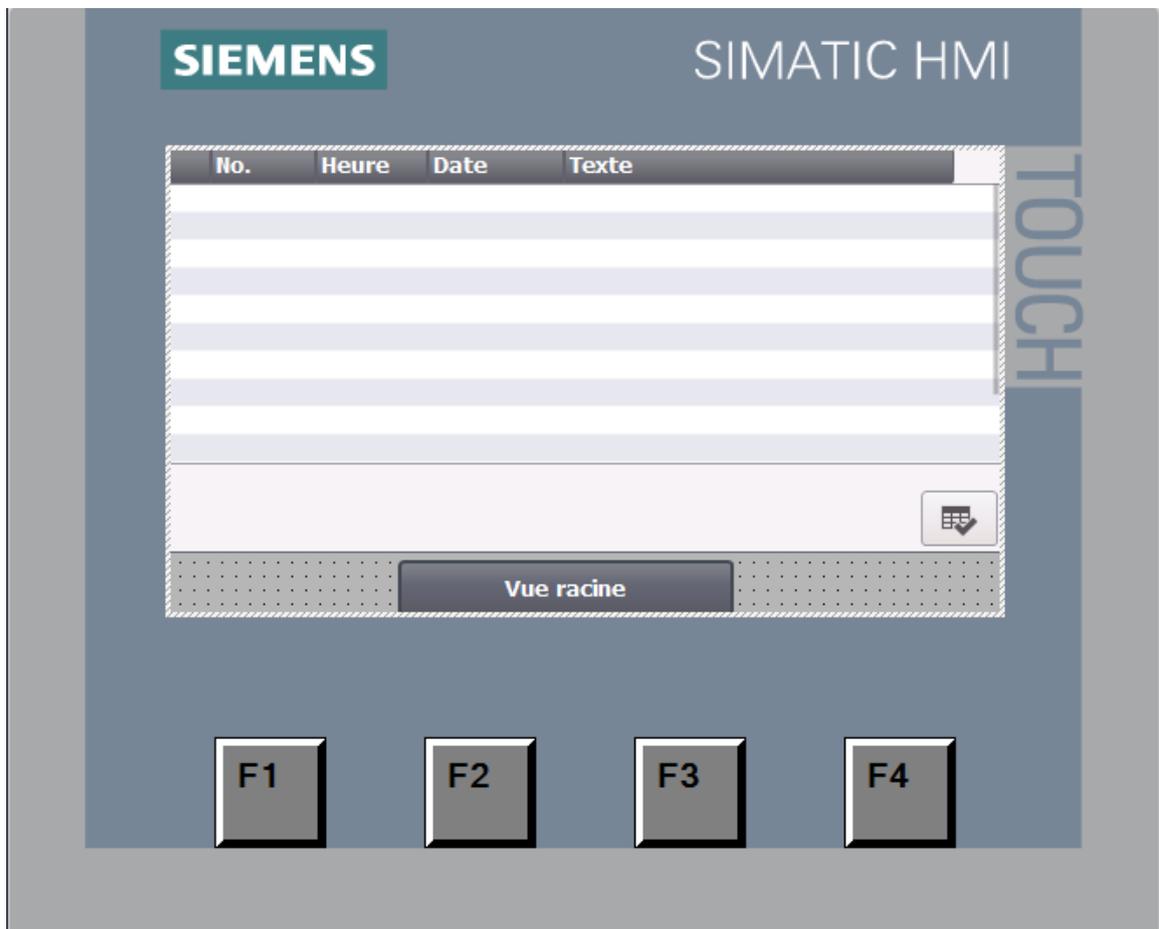


FIGURE 4.12 – Vue de alarmes

Le bouton qui se trouve en bas à droite de la table sert à acquitter les alarmes, après avoir sélectionné la table on peut l'ajouter à partie de Propriétés -> Barre d'outils -> Boutons -> Acquitter, comme la figure 4.13 l'indique.

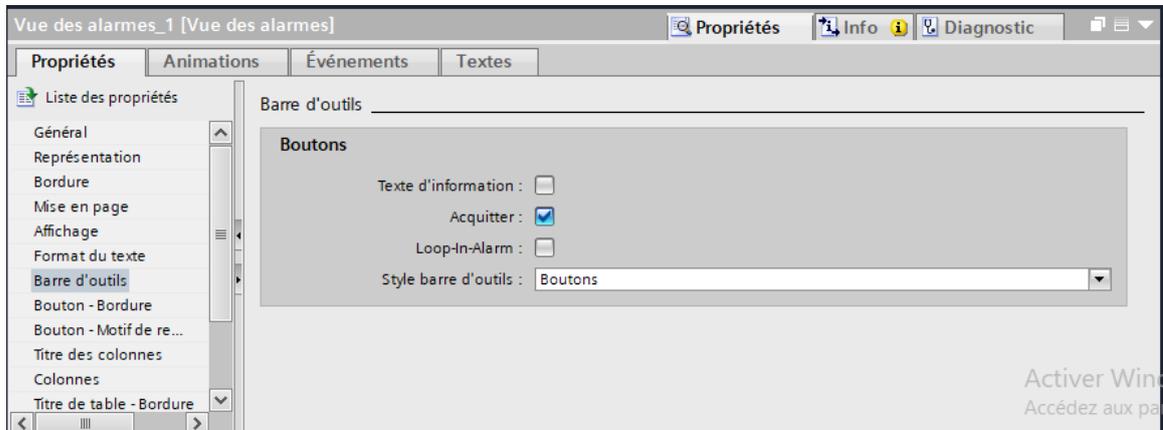


FIGURE 4.13 – Ajout du bouton d'acquiescement

Le navigateur de projet devrait être comme la figure 4.14 le montre.

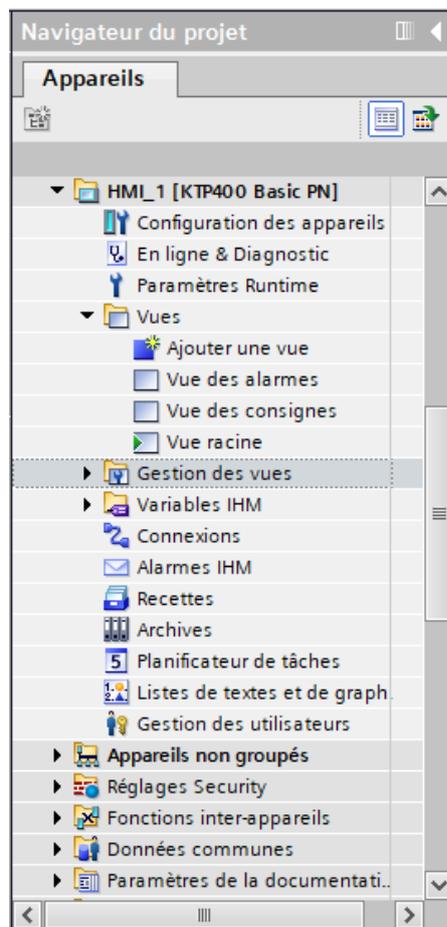


FIGURE 4.14 – Navigateur du projet

4.6.3 Vue globale

A partir du navigateur du projet, on ouvre le dossier "Gestion des vues" et on sélectionne la vue globale, sur cette vue on peut ajouter des éléments qui seront visibles sur tous les vues qui existent. Dans notre cas on va ajouter l'indicateur d'alarme qui se trouve dans le groupe "Contrôles", comme indique la figure 4.15, cet indicateur est impératif pour savoir si une alarme est apparue au moment du bobinage, elle se comporte comme un bouton qui ouvre la vue des alarmes lorsque on l'appuie.

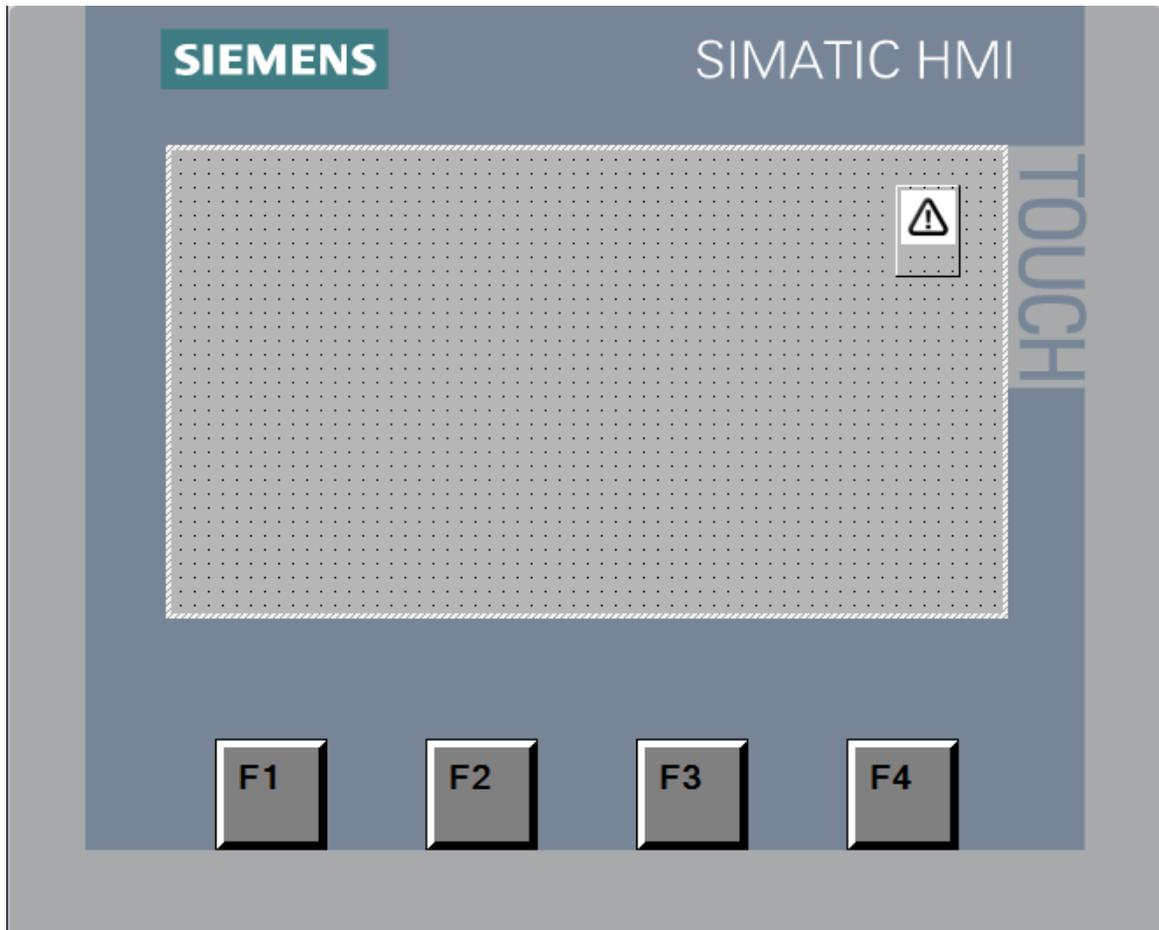


FIGURE 4.15 – Vue globale

4.6.4 Alarmes IHM

Sur le navigateur de projet on sélectionne "Alarmes IHM", on ouvre le tableau "Alarmes de bit", on va le remplir avec les bits de défauts qui enclenchent les alarmes en cas de leur activation par occurrence d'un défaut, pour cela on va sélectionner la variable "Défauts" de type "Int" d'après la table des variables standards, dans la colonne "Variable de déclenchement", puis on choisit le bit de déclenchement qui est désigné par son numéro dans le mot %MW3, par exemple : le bit numéro 8 est le %M3.0, ensuite on définit le texte d'alarme qui va s'afficher dans la vue des alarmes indiquant la défaut qui s'est produit, la figure 4.16 montre la table des alarmes de bit.

ID	Nom	Texte d'alarme	Classe d'alar...	Variable de déclen..	Bit de ..	Adresse de dé..
1	Alarme de bit_1	Défaut moteur1 vitesse rapide	Errors	Défauts	8	Défauts.x8
2	Alarme de bit_2	Défaut moteur1 vitesse lente	Errors	Défauts	9	Défauts.x9
3	Alarme de bit_3	Défaut défreinage	Errors	Défauts	10	Défauts.x10
4	Alarme de bit_4	Défaut moteur2 sens direct	Errors	Défauts	11	Défauts.x11
5	Alarme de bit_5	Défaut moteur2 sens inverse	Errors	Défauts	12	Défauts.x12
6	Alarme de bit_6	Défaut capot ouvert	Errors	Défauts	13	Défauts.x13

FIGURE 4.16 – Table des alarmes IHM

Conclusion

Dans ce dernier chapitre on a donné une description du WinCC et on a élaboré la configuration de l'écran tactile avec ce logiciel, qui va servir à remplacer le pupitre de manoeuvre à entrées physiques.

Cette amélioration offre à l'opérateur de la bobineuse une utilisation meilleure, et un meilleur suivi de l'évolution du processus.

Conclusion générale

Dans le cadre de ce projet d'automatisation d'une bobineuse, nous avons remplacé la logique câblée qui existait avant avec un automate, des actionneurs et des éléments de protection modernes, plus l'écran tactile.

Ce projet qui m'a été confié par l'entreprise algérienne Electro-Industries d'Azazga, consiste à l'amélioration de la machine bobineuse en l'automatisant par un automate programmable S7-1200 (CPU 1214C) de la firme SIEMENS et améliorer la commande et la supervision du fonctionnement de la machine en remplaçant son pupitre de manoeuvre par un écran tactile KTP400.

Après avoir étudié la machine, nous avons acquis une compréhension de son fonctionnement et identifié ses différentes Limitations, ce qui nous a permis de compléter le cahier des charges et de fixer nos objectifs.

Nous avons élaboré un programme en langage Ladder avec le TIA Portal, basé sur la logique séquentielle derrière le fonctionnement de la bobineuse, et les nécessités de cette machine, ce programme est ensuite chargé dans la CPU et exécuté pour pouvoir simuler l'opération de la machine.

Pour commander la bobineuse, nous avons configuré un écran tactile à l'aide du logiciel WinCC Flexible. Ce panneau est utilisé pour contrôler et visualiser l'état de la bobineuse, ce qui a considérablement réduit la taille de l'unité de commande.

Une fois l'automatisation effectuée, toutes les modifications désirées peuvent être configurées directement dans le programme, sans avoir à manipuler les circuits.

Cependant, mon travail représente une solution pour remplacer la logique câblée qui, je l'espère, fonctionnera convenablement et fournira l'efficacité et la sûreté nécessaires dans l'industrie.

Bibliographie

- [1] *Manuel d'utilisation de la bobineuse, Selektemat 235.*
- [2] <https://www.techno-science.net/definition/3213.html>.
- [3] https://formation-energetique.fr/images/LPBatiment/Les_fusiblesPP.pdf.
- [4] <https://zoneelec.com.tn/wp/product/contacteur-chint/>.
- [5] <https://www.omch.co/fr/incremental-rotary-encoder/>.
- [6] https://sti2d.ecolelamache.org/les_capteurs_de_proximit_inductifs.html.
- [7] Bertrand Michel. *Automates programmables industriels, techniques de l'ingénieur.*
- [8] <https://scietech.fr/architectures-des-automates-programmables-industriels-scietech/>.
- [9] <https://www.automation24.fr/siemens-cpu-1214c-6es7214-1ag40-0xb0>.