الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITE MOHAMED SEDDIK BENYAHIA JIJEL

Faculté des sciences et de la technologie

Département d'électrotechnique

N° : ……/2025

MEMOIRE DE MASTER

DOMAINE: Scienceset Technologies

FILIERE: ÉLECTROMECANIQUE

SPECIALITE : Électromécanique

Thème

Automatisation d'un Système Virtuel de Tri de Caisses par Automate Programmable Industriel

Présenté Par : BEN HALOULOU ADEL **Encadré Par : N. Lourci**

CHOUIT ANIS MAHREZ

Date de soutenance : 28 / 06 / 2025

Jury de Soutenance

Président : M.Hadef	**Grade MCA**	**Univ MSB jijel**
Encadreur : N. Lourci	**Grade MCB**	**Univ MSB jijel**
Examinateur 1:A.Kimouche	**Grade MCB**	**Univ MSB jijel**

Promotion :2024 /2025

Remerciements

Nous exprimons nos plus sincères remerciements à Monsieur Lourci Nabil, notre encadreur, pour son accompagnement exemplaire tout au long de la réalisation de ce mémoire de fin d'études du master.

Nous lui sommes profondément reconnaissants pour son engagement constant, sa disponibilité et ses précieux conseils qui ont grandement contribué à la réussite de ce travail. Ses efforts ne se sont pas limités à cette étape, mais se sont étendus à l'ensemble de notre parcours académique, où il n'a cessé de nous encadrer, de nous motiver et de nous transmettre les compétences nécessaires à notre formation dans notre spécialité.

Grâce à sa pédagogie, sa rigueur et sa passion pour l'enseignement, il a su nous préparer efficacement à intégrer le monde professionnel.

Nous lui témoignons tout notre respect et notre profonde gratitude.

Dédicace

Je dédie ce travail :

À mes chers parents, Pour votre amour inconditionnel, vos sacrifices silencieux et votre soutien sans faille. Vous êtes les fondations de mon parcours, et c'est grâce à vous que j'ai pu rêver, persévérer et atteindre cet objectif. Je vous dédie cette réussite avec tout mon respect et ma reconnaissance.

Aux cadres des entreprises pour lesquelles j'ai travaillé, merci pour la confiance que vous m'avez accordée, pour vos conseils, votre exigence et votre professionnalisme. Vous m'avez permis de me former sur le terrain, de me confronter à la réalité du métier et de me construire jour après jour.

Grâce à vous, je suis devenu l'ingénieur que je suis aujourd'hui.

Adel.

Dédicace

Je dédie ce mémoire :

À mes chers parents, pour leur amour inconditionnel, leur soutien constant et leurs sacrifices tout au long de mon parcours.

À mes frères et sœurs, pour leur encouragement, leur compréhension et leur présence rassurante.

À mes enseignants et encadrants, pour leur encadrement, leurs conseils et leur patience qui ont contribué à forger mes compétences.

Merci du fond du cœur.

Anis Mahrez.

Résumé

Le développement d'un système de tri automatisé moderne repose sur la synergie entre l'automate haute performance "Siemens S7-1500", son environnement de programmation unifié "TIA Portal", et le logiciel de simulation 3D "Factory I/O". d'automatisation.

L'API S7-1500 fournit la puissance de calcul et la réactivité nécessaires pour le contrôle à grande vitesse, tandis que TIA Portal centralise la configuration et le développement de la logique de contrôle avec des langages adaptés comme SCL et LAD. Cette approche est complétée par Factory I/O, qui permet la création d'un "jumeau numérique" pour tester, valider et optimiser le système de manière virtuelle et sans risque. L'intégration de ces trois technologies constitue un écosystème de développement performant qui réduit les coûts, accélère le temps de mise en service et améliore la qualité globale du projet.

Mots Clés : Automatisation, Simulation, Factory IO, TIA Portal, Siemens S7-1500, STEP 7, PLCSIM.

Abstract

The development of a modern automated sorting system is based on the synergy between the high-performance PLC "Siemens S7-1500", its unified programming environment "TIA Portal", and the 3D simulation software "Factory I/O". automation.

The S7-1500 PLC provides the computing power and responsiveness needed for high-speed control, while TIA Portal centralizes the configuration and development of control logic with suitable languages such as SCL and LAD. This approach is complemented by Factory I/O, which enables the creation of a "digital twin" for virtual and risk-free testing, validation, and optimization of the system. The integration of these three technologies creates a powerful development ecosystem that reduces costs, accelerates time to commission, and improves overall project quality..

Keywords : Automatisation, Simulation, Factory IO, TIA Portal, Siemens S7-1500, STEP 7, PLCSIM.

TABLE DES MATIÈRES

Remerciements i

Dédicace i

Résumé iii

Abstract iv

Table Des Matières v

Liste des Figures vii

Liste des Tables viii

Introduction générale 1

1 Automate Programmable Industriel : S7-1500 3
- 1.1 Introduction . . . 3
- 1.2 L'API (Automate Programmable Industriel) . . . 4
  - 1.2.1 Définition et rôle d'un API . . . 4
  - 1.2.2 Architecture générale d'un API . . . 5
  - 1.2.3 Cycle de fonctionnement d'un API . . . 6
- 1.3 Présentation de l'API Siemens S7-1500 . . . 7
  - 1.3.1 Aperçu de la gamme S7-1500 . . . 7
  - 1.3.2 Focus sur les caractéristiques techniques spécifiques . . . 8
  - 1.3.3 Avantages de l'utilisation du S7-1500 . . . 10
- 1.4 Logiciel de Programmation TIA Portal (Totally Integrated Automation Portal) . . . 11
  - 1.4.1 Introduction à TIA Portal . . . 11
  - 1.4.2 Les différents composants de TIA Portal . . . 11
  - 1.4.3 Présentation des langages de programmation IEC 61131-3 supportés par le S7-1500 et TIA Portal . . . 14
  - 1.4.4 Interface utilisateur de TIA Portal . . . 17
- 1.5 Conclusion . . . 19

2 Logiciel de Simulation des Systèmes Automatisés et d'Usines Virtuels FACTORY I/O 20
2.1 Introduction . . . 20
2.2 Présentation et Fonctionnalités de Factory I/O . . . 21
2.2.1 Aperçu général de Factory I/O . . . 21
2.2.2 Bibliothèque d'équipements industriels . . . 22
2.2.3 Physique réaliste et comportement des objets . . . 37
2.2.4 Interface utilisateur de Factory I/O . . . 37
2.2.5 Modes de fonctionnement . . . 40
2.3 Intégration de Factory I/O avec l' API S7-1500 et TIA Portal . . . 41
2.3.1 Principe de la communication API-Simulation . . . 41
2.3.2 Mapping des E/S virtuelles et réelles . . . 46
2.3.3 Utilisation de PLCSIM Advanced . . . 48
2.4 Conception du Système de Tri de Caisses dans Factory I/O . . . 48
2.4.1 Présentation des éléments nécessaires à la modélisation . . . 48
2.4.2 Processus de construction de la scène 3D . . . 49
2.4.3 Configuration des propriétés des objets . . . 49
2.5 Avantages de l'Utilisation de Factory I/O . . . 50
2.6 Conclusion . . . 51

3 Conception et Mise en Œuvre d'un Système de Tri Automatisé sur Factory I/O 52
3.1 Introduction . . . 52
3.2 Description du Système Virtuel sur Factory I/O . . . 53
3.2.1 Structure générale du système . . . 53
3.2.2 Cahier des charges : . . . 57
3.3 Programmation de l'API avec TIA Portal : . . . 58
3.3.1 Configuration Matériel : . . . 58
3.3.2 La réalisation du programme : . . . 61
3.4 Conclusion . . . 66

Conclusion générale 68

Bibliographie 72

Annexe 73

TABLE DES FIGURES

1.1 Structure d'un automate programmable industriel (API) 6
1.2 Affichage de l'interface de configuration 12
1.3 Affichage du module défaillant dans la vue des appareils 13
1.4 Affichage de l'interface de simulation 14
1.5 Exemple de programme LAD 14
1.6 Exemple de programme FBD 15
1.7 Exemple de programme ST 15
1.8 Exemple de programme GRAFCET 16
1.9 Exemple des composants d'une vue de projet 18

2.1 l'interface utilisateur et le flux de travail général 38
2.2 L'interface d'E/S 38
2.3 L'interface liste des paramètres 39
2.4 Barre d'outil 39
2.5 Liste des objets au sein de la scène 40

3.1 Poste de Tri de Caisses 53
3.2 Plan d'exploitation du système 54
3.3 Poste Palettiseur N°1 55
3.4 Poste de Tri 56
3.5 Poste Stockage 57
3.6 main[OB],(FBD) 62
3.7 Suite main[OB],(FBD) 62
3.8 Suite main[OB],(FBD) 63
3.9 Suite main[OB],(FBD) 63
3.10 PP_01[FC],(Ladder) 64
3.11 Suite PP_01[FC],(Ladder) 64
3.12 Suite PP_01[FC],(Ladder) 65
3.13 Suite PP_01[FC],(Ladder) 65
3.14 Suite PP_01[FC],(Ladder) 66

LISTE DES TABLEAUX

1.1 Spécifications techniques clés du Siemens S7-1500 (Exemple CPU 1513-1 PN) 10
1.2 Comparaison des Langages de Programmation IEC 61131-3 17

2.1 Exemple de Mappage des E/S Virtuelles (Factory I/O) vers les Variables API (TIA Portal) 47

3.1 Table des variables API 61

INTRODUCTION GÉNÉRALE

Dans le paysage industriel contemporain, l'automatisation est devenue un pilier fondamental pour optimiser l'efficacité, la précision et la sécurité des processus de fabrication. Face aux défis d'un marché mondialisé et hautement concurrentiel, les entreprises sont contraintes d'adopter des solutions innovantes pour accroître leur productivité et réduire leurs coûts opérationnels. Les systèmes de tri automatisés, en particulier, jouent un rôle crucial dans la gestion des flux de matériaux, permettant une classification rapide et fiable des produits, qu'il s'agisse de marchandises, de matières premières ou de produits finis.

Ce mémoire explore l'automatisation d'un système virtuel de tri de caisses à l'aide d'un Automate Programmable Industriel (API). La démarche intègre trois outils technologiques majeurs : l'API Siemens S7-1500, l'environnement de développement TIA Portal, et le simulateur 3D Factory I/O, offrant un environnement réaliste pour la conception et le test.

L'objectif principal de ce travail est de démontrer la faisabilité et les avantages d'une telle approche pour la validation et l'optimisation des systèmes automatisés avant leur déploiement physique. Nous détaillerons les principes fondamentaux de ces technologies, leur intégration pour créer un "jumeau numérique" fonctionnel, et la mise en œuvre pratique d'un système de tri complexe, soulignant ainsi la transition des concepts théoriques vers des solutions d'ingénierie concrètes et performantes.

Le premier chapitre présente l'automatisation industrielle comme un levier stratégique pour améliorer l'efficacité, la précision et la sécurité dans la production. Il décrit l'API comme le centre du contrôle industriel, en détaillant son architecture (CPU, mémoire, E/S, alimentation, bus) et son cycle de fonctionnement en temps réel. Le Siemens S7-1500 est mis en avant pour ses performances élevées, sa modularité et ses fonctionnalités avancées, en faisant une solution adaptée aux systèmes de tri. Le chapitre aborde aussi le TIA Portal comme environnement de développement complet, avec ses outils de configuration, diagnostic, simulation et son support des langages IEC 61131-3.

Le deuxième chapitre met en avant l'importance de la simulation dans l'automatisation moderne, permettant de modéliser, tester et optimiser les processus dans

un environnement virtuel avant toute mise en œuvre réelle. Il présente Factory I/O comme un simulateur 3D interactif de référence pour le développement et le débogage de systèmes de contrôle. Le chapitre détaille ses principales fonctionnalités, telles que sa bibliothèque d'équipements, son moteur physique et son interface intuitive. Il explique également l'intégration avec l'API Siemens S7-1500 via PLCSIM Advanced et TIA Portal, en abordant le mappage des E/S virtuelles, ainsi que la modélisation du système de tri et les bénéfices apportés par cette approche.

Le troisième chapitre décrit en détail le système de tri virtuel développé dans Factory I/O, en présentant sa structure générale et son cahier des charges. Il aborde ensuite la programmation de l'API via TIA Portal, incluant la configuration matérielle et le développement du programme. Ce chapitre illustre l'application concrète des outils et concepts abordés précédemment pour concevoir un système de tri automatisé opérationnel en simulation.

L'utilisation de PLCSIM Advanced pour l'intégration avec Factory I/O démocratise l'accès à la formation et au développement avancés en automatisation industrielle, permettant aux étudiants et ingénieurs d'acquérir une expérience pratique de la programmation API complexe et du comportement des systèmes en temps réel dans un environnement virtuel sûr et rentable, réduisant ainsi les obstacles à l'apprentissage des technologies d'automatisation de pointe et favorisant une main-d'œuvre plus qualifiée.

CHAPITRE 1

AUTOMATE PROGRAMMABLE INDUSTRIEL : S7-1500

1.1 Introduction

L'automatisation industrielle représente un pilier fondamental des processus de fabrication modernes, marquant une évolution significative dans la manière dont les industries opèrent. Son importance ne cesse de croître, portée par la nécessité d'améliorer l'efficacité opérationnelle, d'accroître la précision des tâches et de renforcer la sécurité des environnements de travail. L'intégration de technologies avancées, telles que les robots industriels, les systèmes de vision artificielle et les capteurs de pointe, permet d'automatiser une vaste gamme de tâches manufacturières, allant de l'assemblage à la manutention des matériaux, en passant par le contrôle qualité et l'emballage.[1]

Dans un contexte industriel en constante évolution, l'automatisation représente une transformation majeure aux retombées profondes. Elle constitue un levier stratégique permettant d'accroître significativement la productivité, de stabiliser la qualité des produits, de réduire les coûts opérationnels et d'améliorer la sécurité sur les lieux de travail. En limitant les interventions humaines dans les tâches répétitives ou dangereuses, elle favorise un environnement plus sûr et conforme aux normes en vigueur. De plus, la capacité des systèmes automatisés à fonctionner en continu, tout en minimisant les erreurs, renforce la maîtrise des processus et la réactivité face aux aléas. Dans un marché mondialisé et hautement concurrentiel, ces avantages ne sont plus des options, mais des impératifs pour assurer la pérennité et la croissance des entreprises. Ce chapitre se propose donc d'analyser en détail les bénéfices de l'automatisation industrielle ainsi que son rôle dans l'optimisation des performances globales des systèmes de production. [2]

Le choix de l'API Siemens S7-1500 pour ce projet repose sur ses capacités de pointe en matière d'automatisation. Cet automate programmable industriel est reconnu pour sa vitesse, sa précision et sa polyvalence, ce qui le rend particulièrement adapté aux applications complexes. Sa conception modulaire offre une flexibilité d'ex-

pansion et de personnalisation, permettant d'adapter la solution d'automatisation aux besoins spécifiques du projet grâce à la prise en charge de divers modules de signaux et d'unités centrales (CPU). [3]

Le S7-1500 de Siemens est particulièrement bien adapté aux systèmes de tri automatisés grâce à sa puissance de traitement élevée, permettant une exécution rapide des tâches critiques [3]. La CPU 1513-1 PN offre des temps de cycle très courts (25 ns pour les opérations binaires) et une mémoire étendue, jusqu'à 32 Go avec carte SIMATIC, ce qui facilite la gestion d'algorithmes complexes [4]. Sa compatibilité avec une large gamme de modules d'E/S (SM 521, SM 522) permet une intégration flexible de capteurs et actionneurs. Il dispose en standard de PROFINET et peut être étendu à d'autres protocoles (PROFIBUS, Ethernet/IP), assurant une communication fluide avec des systèmes comme Factory I/O . Enfin, son intégration dans le TIA Portal permet de réduire les erreurs, accélérer le développement et garantir la pérennité du système.[5]

Ce chapitre a pour objectif de fournir une présentation exhaustive de l'API Siemens S7-1500, en détaillant ses principes fondamentaux, son architecture et ses caractéristiques techniques spécifiques. Il introduira également le logiciel TIA Portal, l'environnement de développement intégré de Siemens, en expliquant ses composants, les langages de programmation pris en charge et son interface utilisateur. L'ensemble de ces informations vise à établir une compréhension solide des outils matériels et logiciels essentiels à la conception et à la mise en œuvre de solutions d'automatisation industrielle, en particulier pour le projet de système de tri.

1.2 L'API (Automate Programmable Industriel)

1.2.1 Définition et rôle d'un API

Un Automate Programmable Industriel (API), ou PLC (Programmable Logic Controller) en anglais, est un ordinateur numérique spécialisé, conçu spécifiquement pour opérer dans des environnements industriels afin de contrôler et de surveiller divers équipements et processus. Son rôle fondamental est d'automatiser des processus spécifiques, des fonctions de machine ou des lignes de production entières. Les API ont été développés comme une alternative électronique aux systèmes de logique à relais câblés traditionnels, offrant une flexibilité, une fiabilité et une facilité de reprogrammation considérablement améliorées par rapport à ces systèmes rigides. Ils sont intrinsèquement robustes et résistants aux facteurs environnementaux difficiles tels que les variations de température, les vibrations et le bruit électrique, ce qui les rend parfaitement adaptés aux applications industrielles.[6]

La transition des systèmes à relais câblés vers les API représente une évolution majeure dans l'automatisation. Historiquement, toute modification du processus de contrôle nécessitait un recâblage physique fastidieux, ce qui rendait les systèmes coûteux et lents à adapter. L'avènement des API a marqué un passage de l'automatisation fixe à une automatisation flexible basée sur des logiciels. Cette flexibilité est devenue indispensable dans la fabrication moderne, où les lignes de production doivent souvent s'adapter rapidement à de nouveaux produits ou à des processus modifiés.

La capacité d'un API à être facilement reprogrammé n'est pas seulement une caractéristique technique, mais un changement de paradigme qui permet un prototypage rapide, une amélioration continue et une fabrication agile, influençant directement le délai de mise sur le marché et la réactivité opérationnelle des entreprises.[6]

1.2.2 Architecture générale d'un API

Un système API typique est composé de plusieurs éléments clés qui travaillent en synergie pour exécuter les tâches de contrôle. Les trois composantes principales sont l'unité centrale de traitement (CPU), les modules d'entrées/sorties (E/S) et une alimentation électrique (figure 1.1). À cela s'ajoutent la mémoire et un système de bus de communication.[7]

— **Unité Centrale de Traitement (CPU) :** La CPU est le "cerveau" de l'API. Elle est chargée d'exécuter la logique de programme, de contrôler toutes les opérations de l'API et de gérer la communication avec les autres composants et périphériques. Elle intègre un microprocesseur et des éléments de mémoire.[7]

— **Mémoire :** La mémoire de l'API est divisée en plusieurs types. La mémoire morte (ROM) stocke le système d'exploitation et les pilotes, tandis que la mémoire vive (RAM) conserve l'état et les détails des données d'entrée/sortie, ainsi que les programmes d'application.[7] La mémoire programme contient le code de l'API exécuté par la CPU, et la mémoire de données stocke les variables et les paramètres du système. La mémoire peut être volatile (RAM, perd son contenu à la coupure de courant) ou non volatile (ROM, conserve son contenu).[8]

— **Modules d'Entrées/Sorties (E/S) :** Ces modules constituent l'interface essentielle entre l'API et les dispositifs de terrain, tels que les capteurs et les actionneurs. Ils sont responsables de la conversion des signaux analogiques et numériques en un format compréhensible par la CPU. Les E/S numériques gèrent les signaux tout ou rien (par exemple, des boutons-poussoirs, des lampes), tandis que les E/S analogiques traitent les signaux variables (par exemple, des capteurs de température, des variateurs de fréquence).[7]

— **Alimentation :** L'alimentation convertit le courant alternatif (CA) en courant continu (CC) nécessaire au fonctionnement de la CPU et des modules d'E/S. Les alimentations de qualité industrielle pour API incluent souvent des protections contre les surtensions, les courts-circuits et les surcharges.[7]

— **Bus de Communication :** Un système de bus de contrôle assure le transfert d'informations vers et depuis la CPU. Les API utilisent divers protocoles de communication (par exemple, Modbus, Profibus, Ethernet/IP, PROFINET) pour interagir avec les composants de terrain, les réseaux et d'autres dispositifs. La CPU et les modules d'E/S sont généralement connectés via un fond de panier ou un système de bus, permettant leur interaction. La diversité des protocoles de communication (RS-232, Modbus RTU, Ethernet

TCP/IP, Profinet, etc.) souligne la capacité de l'API à se connecter à une multitude de dispositifs et de réseaux. Cette capacité est fondamentale pour l'interopérabilité industrielle, un concept clé de l'Industrie 4.0 et de l'Internet Industriel des Objets (IIoT). Dans un environnement industriel hétérogène, les API doivent s'intégrer de manière fluide avec des machines, des systèmes de contrôle et des logiciels d'entreprise pour permettre l'échange de données en temps réel. Cette fonction de l'API en tant que hub de données est cruciale pour l'optimisation des opérations, la maintenance prédictive et la prise de décision basée sur les données dans une usine connectée.[7]

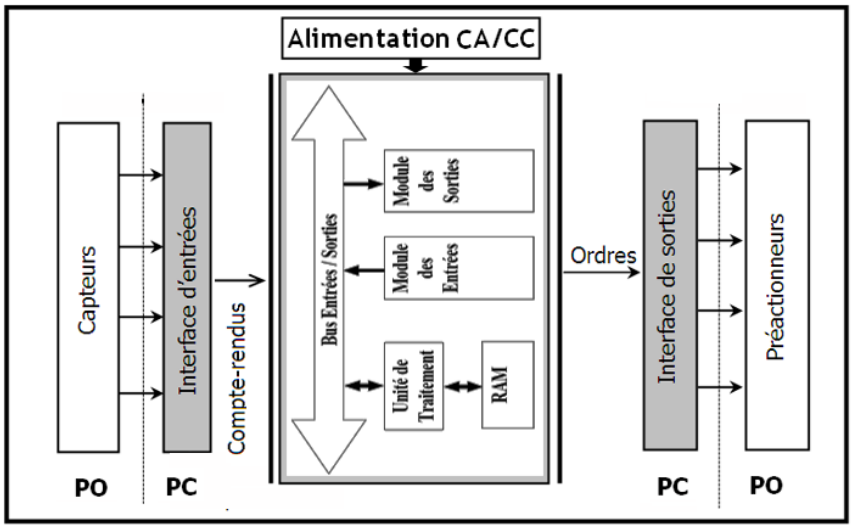


FIGURE 1.1 – Structure d'un automate programmable industriel (API)

1.2.3 Cycle de fonctionnement d'un API

Le fonctionnement d'un API repose sur un processus continu et répétitif appelé "cycle de balayage" ou "scan cycle".Ce cycle assure un contrôle en temps réel et une réactivité constante aux changements de conditions dans l'environnement industriel.[6] Le cycle de balayage se déroule en quatre étapes fondamentales :

1. **Lecture des entrées (Input Scan) :** Au début de chaque cycle, l'API lit l'état de tous les dispositifs d'entrée connectés à ses modules d'entrée. Cette phase permet de capturer les conditions réelles du processus, telles que la présence d'un objet détectée par un capteur ou l'activation d'un bouton-poussoir.[9]

2. **Exécution du programme (Program Scan/Execution) :** Une fois les entrées lues, la CPU exécute la logique de contrôle définie par l'utilisateur et stockée dans sa mémoire. Durant cette phase, l'API traite les données d'entrée, évalue les conditions programmées et prend des décisions en fonction des instructions. C'est le cœur de l'intelligence de l'API, où les calculs et les opérations logiques sont effectués.[9]

3. **Mise à jour des sorties (Output Scan) :** Après l'exécution du programme, l'API met à jour l'état de tous les dispositifs de sortie. Les résultats des décisions prises par le programme sont traduits en commandes actionnables pour les actionneurs, tels que le démarrage d'un moteur, l'ouverture d'une vanne ou l'activation d'un voyant lumineux.[9]

4. **Tâches de communication et d'auto-diagnostique :** Cette dernière phase comprend des tâches de maintenance interne, telles que la communication avec les dispositifs de programmation ou les réseaux, ainsi que l'exécution de diagnostics internes pour vérifier l'intégrité du système.16 Ces tâches garantissent le bon fonctionnement et la fiabilité de l'API.[9]

Ce comportement déterministe, où les entrées sont lues, la logique exécutée et les sorties mises à jour dans un laps de temps prévisible, est essentiel pour le contrôle en temps réel dans les applications industrielles. Contrairement aux ordinateurs à usage général, les API sont conçus pour garantir des temps de réponse spécifiques, ce qui est primordial pour les systèmes critiques en matière de sécurité et les processus nécessitant une synchronisation précise (par exemple, le contrôle de mouvement ou le tri à grande vitesse). Le temps de cycle de balayage influence directement la réactivité du système, et la compréhension des facteurs qui l'affectent (tels que la complexité de la logique, les configurations d'E/S et la vitesse du processeur) est cruciale pour optimiser les performances de l'API et assurer l'efficacité et la sécurité opérationnelles. [9]

1.3 Présentation de l'API Siemens S7-1500

1.3.1 Aperçu de la gamme S7-1500

La gamme Siemens S7-1500 représente une avancée majeure dans le domaine des automates programmables industriels, étant qualifiée de "contrôleur programmable de pointe conçu pour les tâches d'automatisation haute performance".Ces API sont reconnus pour leur vitesse, leur précision et leur polyvalence, ce qui les rend parfaitement adaptés aux applications complexes dans des secteurs variés tels que la fabrication, le traitement et les infrastructures.

Le S7-1500 se distingue par des fonctions de sécurité intégrées (Safety Integrated), un contrôle de mouvement avancé et des capacités de communication étendues. Sa conception modulaire est un atout majeur, permettant une expansion et une personnalisation aisées grâce à la prise en charge de diverses CPU et modules de signaux. Cette modularité assure une adaptabilité aux exigences spécifiques de chaque application. De plus, le S7-1500 est entièrement pris en charge par le TIA Portal, l'environnement d'ingénierie intégré de Siemens, ce qui simplifie considérablement les processus de programmation et de configuration, améliorant ainsi la facilité d'utilisation et la connectivité transparente avec d'autres dispositifs et systèmes.

La description du S7-1500 met en évidence son caractère "à la pointe de la technologie" et sa "viabilité intégrée pour l'avenir". Cela ne concerne pas seulement ses performances actuelles, mais implique une philosophie de conception qui anticipe les besoins industriels futurs. Sa modularité, ses vastes options de communication et ses fonctions avancées intégrées (applications d'IA avec le module TM NPU, contrôle de mouvement étendu, systèmes redondants) suggèrent que Siemens positionne le S7-1500 non pas comme un simple contrôleur, mais comme une plateforme flexible et évolutive capable de s'adapter aux tendances de l'Industrie 4.0. Cette capacité à anticiper et à s'adapter réduit le risque d'obsolescence et permet des mises à niveau et des extensions progressives, offrant une valeur à long terme significative au-delà du déploiement initial du projet.[3]

1.3.2 Focus sur les caractéristiques techniques spécifiques

Les spécifications techniques du Siemens S7-1500 sont un facteur déterminant de son adéquation aux applications d'automatisation exigeantes, comme un système de tri.

— **Puissance du processeur et temps de cycle :** Le S7-1500 offre des performances accrues grâce à une exécution plus rapide des commandes, des extensions de langage, de nouveaux types de données et un bus de fond de panier plus rapide[5]. Par exemple, la CPU 1513-1 PN présente des temps d'opération typiques de 25 ns pour les opérations binaires et de 170 ns pour l'arithmétique en virgule flottante.[4] Cette vitesse élevée est cruciale pour garantir des temps de réponse rapides dans des systèmes dynamiques comme le tri.

— **Capacité mémoire :** L'API dispose d'une mémoire de travail intégrée pour le programme (par exemple, 600 Ko) et pour les données (par exemple, 2,5 Mo pour la CPU 1513-1 PN). La mémoire de chargement est extensible via une carte mémoire SIMATIC, pouvant atteindre jusqu'à 32 Go. La zone de données rémanente est de 256 Ko, avec une zone rémanente étendue pouvant atteindre 2,5 Mo. Cette mémoire abondante est essentielle pour prendre en charge des algorithmes de tri complexes et la journalisation des données.[4]

— **Types de modules d'entrées/sorties disponibles et ceux que vous envisagez d'utiliser :** Les modules de signaux (SM 521 pour les entrées numériques, SM 522 pour les sorties numériques) constituent l'interface entre le contrôleur et le processus. Ils sont disponibles en diverses versions (standard, haute performance, étroites de 25 mm) et offrent une densité de canaux élevée. Des modules d'entrée/sortie analogiques sont également disponibles, proposant des fonctions étendues comme le sur-échantillonnage. [10]

Pour un système de tri, des entrées numériques seraient utilisées pour les capteurs de présence (par exemple, capteurs photoélectriques, de proximité), les boutons-poussoirs (démarrage/arrêt) et les interrupteurs de fin de course. Les sorties numériques contrôleraient les convoyeurs, les poussoirs/déviateurs pneumatiques et les voyants lumineux. Des entrées analogiques pourraient

être utilisées si le tri par poids ou taille nécessite des capteurs analogiques (par exemple, capteurs de charge, capteurs de distance), et des sorties analogiques pour les variateurs de vitesse des convoyeurs.[7]

— **Interfaces de communication intégrées :** Le PROFINET IO (avec un commutateur à 2 ports) est une interface intégrée standard, permettant une communication haute performance. L'API est extensible avec des modules de communication pour PROFIBUS (CM 1542-5) et d'autres protocoles Industrial Ethernet comme TCP/IP, ISO-on-TCP, UDP (CM 1542-1 pour contrôleur/dispositif PROFINET IO). Un serveur/client OPC UA est disponible en option pour une connexion facile à des dispositifs tiers. Ces interfaces sont cruciales pour la connexion à Factory I/O et l'intégration du système de tri dans un réseau plus vaste.[5]

— **Fonctionnalités avancées :**

  — **Sécurité intégrée (Safety Integrated) :** Permet le traitement des programmes standard et de sécurité sur le même contrôleur, assurant la protection du personnel et des machines.[5]

  — **Contrôle de mouvement (Motion Control) :** Intégré sans modules supplémentaires, il prend en charge les axes de vitesse, les axes de positionnement, le fonctionnement synchrone et les codeurs externes. Cela est très pertinent pour un contrôle précis des vitesses de convoyeurs et des mouvements des déviateurs dans un système de tri.[5]

  — **Diagnostic système intégré :** Fournit des informations en temps réel sur les performances du système.[3]

  — **Intelligence Artificielle :** Le module TM NPU permet des applications d'IA telles que la reconnaissance intelligente d'objets et les contrôles visuels de qualité , ce qui pourrait constituer des fonctionnalités avancées pour de futures améliorations du tri.[10]

Le Tableau 1.1 ci-dessous récapitule les spécifications techniques clés du Siemens S7-1500, en utilisant la CPU 1513-1 PN comme exemple représentatif. Cette présentation tabulaire offre un aperçu concis et rapide des capacités fondamentales de l'API, permettant une évaluation rapide de sa puissance et de sa capacité pour les exigences du projet.

Catégorie	Paramètre	Valeur
Processeur	Temps d'opération binaire (typ.)	25 ns
	Temps d'opération mot (typ.)	32 ns
	Arithmétique à virgule fixe (typ.)	42 ns
	Arithmétique à virgule flottante (typ.)	170 ns
Mémoire	Mémoire de travail intégrée (programme)	600 Ko
	Mémoire de travail intégrée (données)	2.5 Mo
	Mémoire de chargement (max.)	32 Go
	Zone de données rémanente (max.)	256 Ko
E/S	Nombre de modules d'E/S	2048
Communication	Contrôleur IO intégré	1

TABLE 1.1 – Spécifications techniques clés du Siemens S7-1500 (Exemple CPU 1513-1 PN)

1.3.3 Avantages de l'utilisation du S7-1500

L'intégration du Siemens S7-1500 dans le système de tri présente plusieurs avantages stratégiques, directement liés aux exigences du projet :

— **Réactivité du tri :**La vitesse de traitement élevée et les temps de cycle courts (25 ns pour les opérations binaires) du S7-1500 sont fondamentaux pour un système de tri à grande vitesse. Cette performance assure une détection rapide des articles et une activation précise des mécanismes de tri, minimisant ainsi les retards et maximisant le débit du système.[3]

— **Capacités de communication pour l'intégration avec Factory I/O :** L'interface PROFINET intégrée et la prise en charge de divers protocoles de communication simplifient grandement la connexion et l'échange de données avec Factory I/O. Cela permet une mise en service virtuelle et des tests fluides de la logique de tri, facilitant le mappage des E/S virtuelles aux variables de l'API.[11]

— **Modularité et Scalabilité :** La conception modulaire du S7-1500 3 offre la flexibilité d'adapter le système à différentes complexités de tri (par exemple, l'ajout de convoyeurs, capteurs ou déviateurs supplémentaires) ou d'envisager des expansions futures sans nécessiter le remplacement complet du contrôleur.[3]

— **Fonctionnalités avancées pour un tri précis :** Le contrôle de mouvement intégré peut être exploité pour une commande précise des vitesses de convoyeurs ou des manipulateurs robotiques si le système de tri implique des mouvements complexes. Les diagnostics intégrés facilitent le dépannage rapide de la ligne de tri, réduisant les temps d'arrêt.[3]

— **Robustesse et Fiabilité :** Conçu pour les environnements industriels exigeants, le S7-1500 garantit un fonctionnement fiable, minimisant les interruptions du processus de tri.[3]

1.4 Logiciel de Programmation TIA Portal (Totally Integrated Automation Portal)

1.4.1 Introduction à TIA Portal

TIA Portal, ou Totally Integrated Automation Portal, est la suite logicielle d'ingénierie unifiée de Siemens, conçue pour la configuration, la programmation, la simulation et la mise en service des systèmes d'automatisation industrielle. Il intègre de manière transparente divers outils de développement, notamment SIMATIC STEP 7 (pour les API), WinCC (pour les IHM), SINAMICS Startdrive (pour les variateurs) et SIMOTION SCOUT TIA (pour le contrôle de mouvement), le tout au sein d'un environnement unique et intuitif.[12] L'objectif principal de TIA Portal est d'accroître la flexibilité, la vitesse et la productivité des projets d'automatisation en offrant un flux de travail intégré et cohérent. Il permet une "chaîne de travail numérique" où des représentations virtuelles des machines et des systèmes peuvent être simulées et testées avant même la construction physique. La valeur fondamentale de TIA Portal réside dans son concept de "Totally Integrated Automation". Cette intégration dépasse la simple compilation d'outils dans un seul logiciel ; elle vise fondamentalement à créer un "jumeau numérique" du système d'automatisation. Cette capacité permet une mise en service et des tests virtuels, ce qui réduit considérablement le besoin de prototypes physiques coûteux et de débogage sur site. L'implication est une réduction significative du temps de mise sur le marché et des coûts de développement, tout en améliorant la qualité et la fiabilité du système final. TIA Portal agit ainsi comme un catalyseur essentiel de la transformation numérique dans l'automatisation industrielle, passant d'un développement séquentiel et cloisonné à une approche parallèle, intégrée et virtualisée.[6]

1.4.2 Les différents composants de TIA Portal

TIA Portal offre une vue structurée de tous les composants d'un projet. Ses principaux éditeurs et outils comprennent :

— **Configuration des appareils et des réseaux :** Cette section permet aux utilisateurs de configurer graphiquement les dispositifs matériels (par exemple, la CPU S7-1500, les modules d'E/S) et les connexions réseau (par exemple, PROFINET, PROFIBUS). Cela inclut l'attribution d'adresses IP, la définition des paramètres de communication et la disposition physique du système d'automatisation(figure1.2).[13]

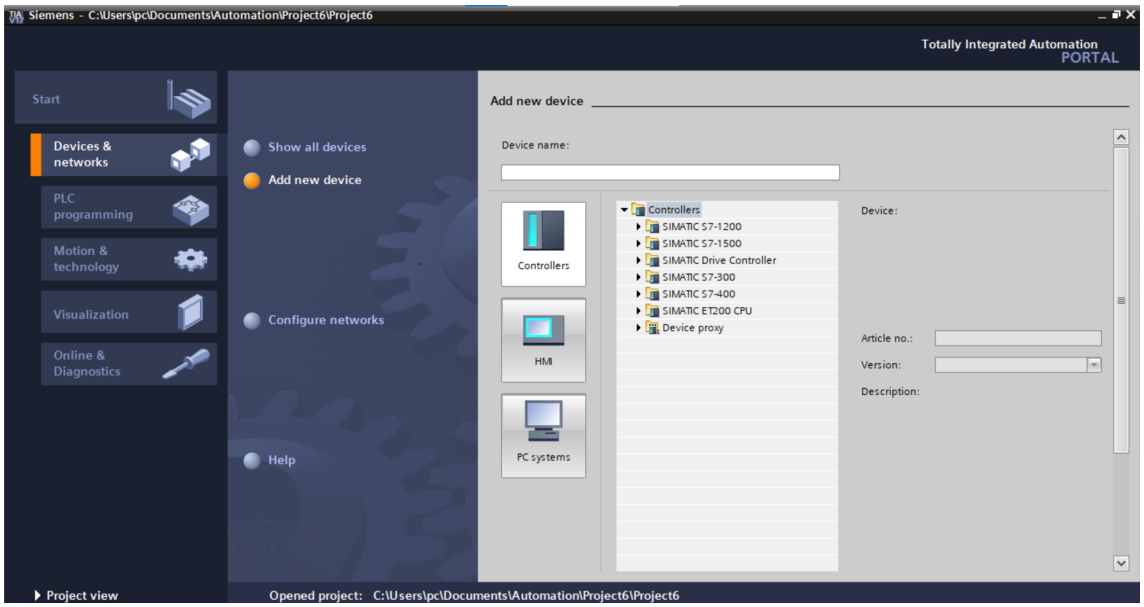


FIGURE 1.2 – Affichage de l'interface de configuration

— **Outils de diagnostic et de mise en service :** TIA Portal fournit des fonctions de diagnostic complètes pour la surveillance des performances du système, l'identification des pannes et le dépannage. Cela inclut la visualisation des dispositifs accessibles, la modification des configurations en ligne et la surveillance de l'état des dispositifs d'E/S(figure1.3).[14]

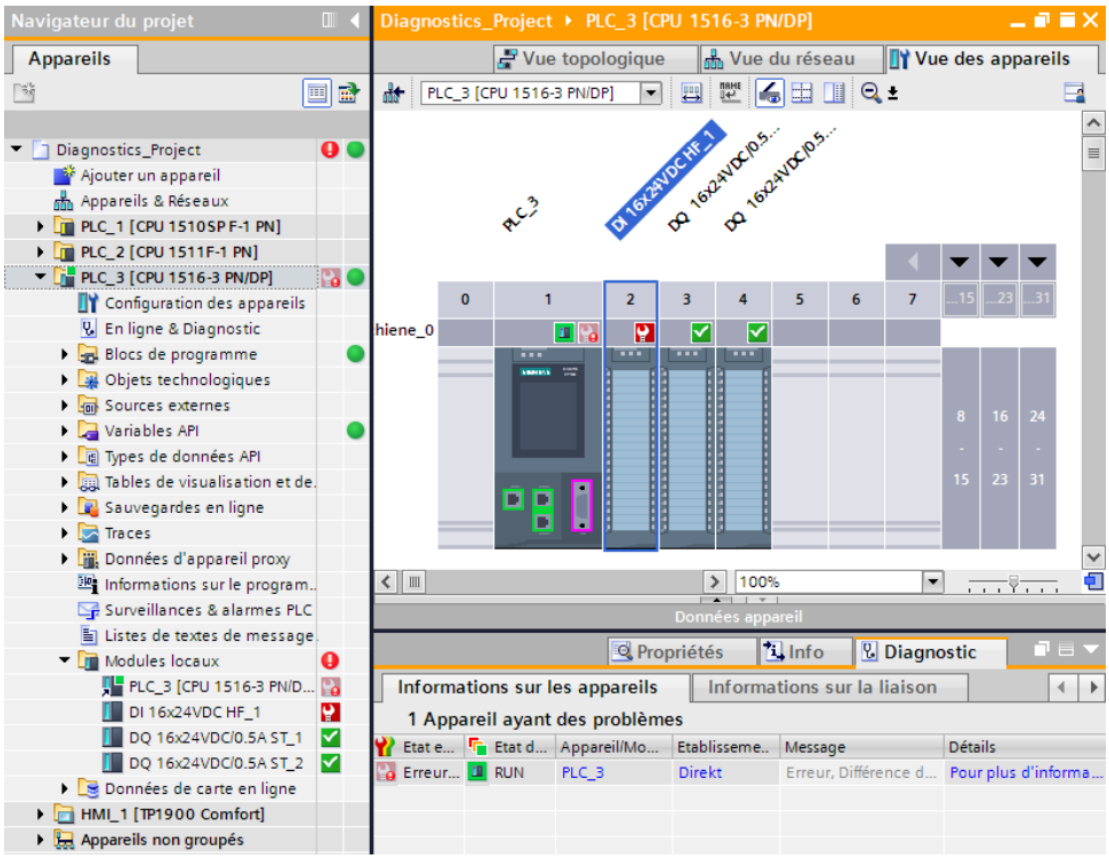


FIGURE 1.3 – Affichage du module défaillant dans la vue des appareils

— **Interface de simulation (en lien avec Factory I/O) :** TIA Portal intègre un logiciel de simulation (SIMATIC S7-PLCSIM et PLCSIM Advanced) qui permet d'exécuter et de tester le matériel et le logiciel dans un environnement simulé directement sur un PC, sans nécessiter de matériel physique supplémentaire. Cet environnement virtuel est crucial pour la connexion et l'interaction avec Factory I/O afin de valider le système(figure1.4). [11]

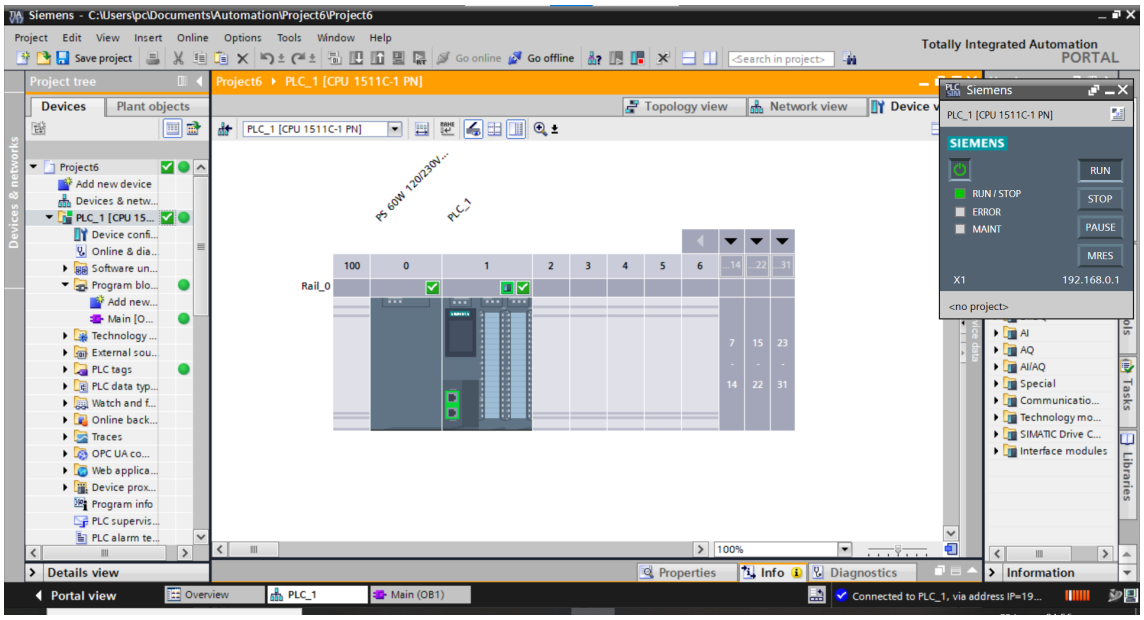


FIGURE 1.4 – Affichage de l'interface de simulation

D'autres composants incluent des éditeurs pour la programmation (STEP 7), le développement d'IHM (WinCC), le contrôle de mouvement (Startdrive), ainsi que des outils pour la gestion de projet, les bibliothèques et l'accès en ligne.[12]

1.4.3 Présentation des langages de programmation IEC 61131-3 supportés par le S7-1500 et TIA Portal

L'API S7-1500 et le TIA Portal prennent en charge plusieurs langages de programmation conformes à la norme IEC 61131-3, offrant une flexibilité considérable aux ingénieurs :

— **Ladder Diagram (LAD) :**C'est un langage graphique qui ressemble aux schémas de relais électriques, ce qui le rend intuitif pour les ingénieurs ayant une formation en électricité (figure1.5). Il est idéal pour la logique de machine de base, les interverrouillages de sécurité, la surveillance des signaux et les conditions simples d'activation/désactivation.[15]

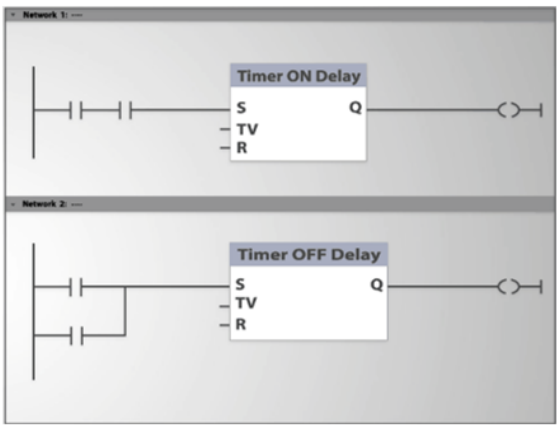


FIGURE 1.5 – Exemple de programme LAD

— **Function Block Diagram (FBD) :** Un autre langage visuel, similaire aux organigrammes, où les éléments logiques sont représentés sous forme de blocs interconnectés (figure1.6). Il est excellent pour la logique répétitive et modulaire, le traitement des signaux analogiques et les boucles PID.[15]

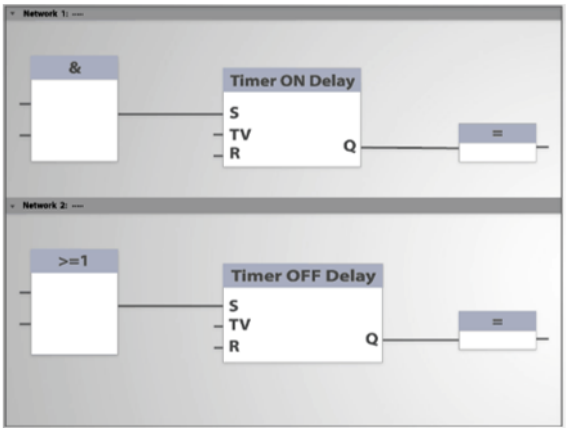


FIGURE 1.6 – Exemple de programme FBD

— **Structured Control Language (SCL) / Structured Text (ST) :** Un langage de programmation de haut niveau, textuel, similaire à Pascal ou C (figure 1.7).Il est le mieux adapté aux algorithmes de contrôle complexes, aux calculs mathématiques, à la manipulation de données, aux boucles et aux instructions conditionnelles. Il est très lisible, maintenable et évolutif pour les grands projets.[15]

```
18 if (tank_lvl_1<=tank_max_lvl) then
19  pump_start_1[:=]1;
20  end_if;
21
22 if (tank_lvl_1>=tank_max_lvl) then
23  pump_start_1[:=]0;
24  end_if;
25
26 if (tank_lvl_2<=tank_max_lvl) then
27  pump_start_2:=1;
28  end_if;
29 if (tank_lvl_2>=tank_max_lvl) then
30  pump_start_2[:=]0;
31  end_if;
```

FIGURE 1.7 – Exemple de programme ST

— **Sequential Function Chart (SFC) / GRAPH :** Une méthode pour programmer des systèmes de contrôle complexes à un niveau plus structuré, idéale pour les opérations basées sur des états et séquentielles. Chaque étape peut contenir une logique programmée dans d'autres langages (figure1.8).[16]

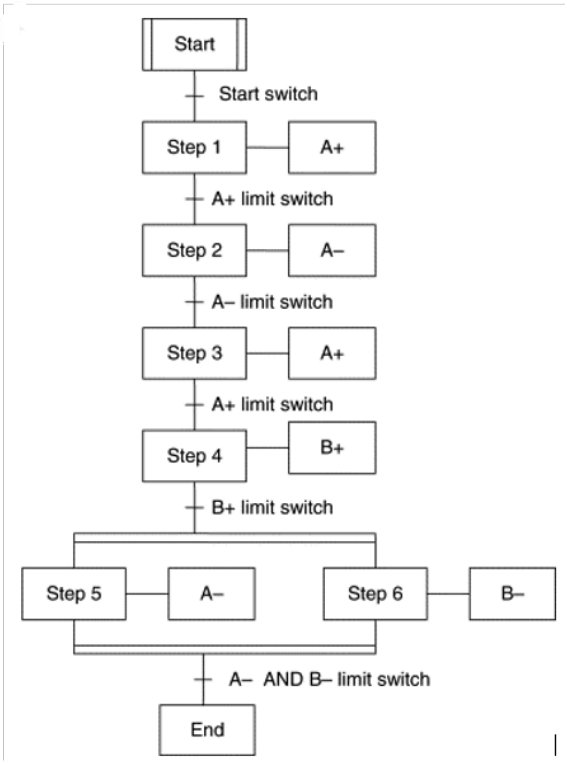


FIGURE 1.8 – Exemple de programme GRAFCET

La prise en charge par le S7-1500 de plusieurs langages IEC 61131-3 est un atout majeur pour la "programmation hybride". Cela signifie que les ingénieurs peuvent sélectionner le langage le plus approprié pour chaque tâche spécifique au sein d'un même projet, en exploitant les forces de chacun. Par exemple, le LAD peut être utilisé pour les interverrouillages simples, le SCL pour les calculs complexes, et le SFC pour les processus séquentiels. Cette flexibilité conduit à un code plus optimisé, plus lisible et plus facile à maintenir, réduisant le temps de développement et améliorant le support à long terme du système. Elle permet également à des équipes aux compétences diverses (par exemple, électriciens, ingénieurs logiciels) de collaborer efficacement sur le même projet. [15]

Cette adaptabilité est un avantage significatif pour les applications industrielles complexes comme un système de tri.

Pour le projet de système de tri, une combinaison de ces langages serait optimale :

— **LAD** serait utilisé pour la logique de contrôle de base, les interverrouillages de sécurité (par exemple, arrêts d'urgence, activation/désactivation de convoyeurs) et les activations simples de capteurs, en raison de sa clarté visuelle et de sa facilité de dépannage par les techniciens.

— **FBD** pourrait être employé pour des fonctions modulaires comme les blocs de commande de moteur ou le traitement des capteurs, permettant ainsi la réutilisation du code.

— **SCL** Une méthode pour programmer des systèmes de contrôle complexes

à un niveau plus structuré, idéale pour les opérations basées sur des états et séquentielles. Chaque étape peut contenir une logique programmée dans d'autres langages.

Le Tableau 1.2 offre une comparaison détaillée des langages de programmation IEC 61131-3, soulignant leurs forces et leurs applications typiques, ainsi que leur pertinence pour le système de tri. Cette présentation structurée facilite la compréhension des choix de conception et justifie l'approche de programmation adoptée.

Langage	**Description**	**Forces**	**Applications Typiques**	**Pertinence pour le Projet de Tri**
LAD	Graphique, ressemble aux schémas de relais.	Facile à comprendre pour les électriciens, visualisation claire.	Interverrouillage de sécurité, contrôle ON/OFF simple, logique de machine.	Idéal pour la logique de base, les interverrouillages et le dépannage rapide des capteurs/actionneurs.
FBD	Graphique,basé sur des blocs fonctionnels.	Modulaire, réutilisable, visualisation du flux de données.	Boucles PID, traitement de signaux analogiques, logique répétitive.	Convient pour des blocs fonctionnels réutilisables (ex : contrôle moteur, gestion capteurs) et des fonctions modulaires.
ST	Textuel, de haut niveau (similaire à Pascal/C).	Calculs complexes, manipulation de données, algorithmes avancés, lisibilité, maintenabilité.	Contrôle de gestion des recettes, algorithmes mathématiques.	Essentiel pour la logique de décision complexe du tri (ex : classification par taille/poids/couleur),comparaison avancée.
SFC	Graphique, basé sur des étapes et des transitions.	Structuration de processus séquentiels, gestion d'états.	Machines d'état, systèmes batch, séquences complexes.	Utile pour la gestion de la séquence globale du processus de tri (ex : attente, détection, diversion).

TABLE 1.2 – Comparaison des Langages de Programmation IEC 61131-3

1.4.4 Interface utilisateur de TIA Portal

L'interface utilisateur de TIA Portal est conçue pour être conviviale et optimiser la productivité. Elle propose deux vues principales : la "vue portail" pour un accès rapide aux outils orientés tâche, et la "vue projet" pour une vue structurée de tous

les composants du projet (figure1.9).

Les zones clés de l'interface incluent :

— **Barre de menus et barre d'outils :**Elles offrent un accès rapide aux commandes et fonctions principales.[13]

— **Arborescence du projet :**Une vue hiérarchique de tous les composants du projet, tels que les dispositifs, les programmes et les bibliothèques. Elle permet d'ajouter, de modifier et d'afficher les propriétés des composants.[13]

— **Zone de travail/Éditeur :** C'est l'espace principal où la programmation, la configuration et les diagnostics sont effectués. Elle peut être divisée et personnalisée pour s'adapter aux préférences de l'utilisateur.[13]

— **Fenêtre d'inspection/Volet des propriétés :** Affiche les propriétés et les détails des objets sélectionnés, facilitant la configuration et le dépannage.[13]

— **Barre d'état (Status Bar) :** Affiche des informations sur les processus en cours et l'état du système.[13]

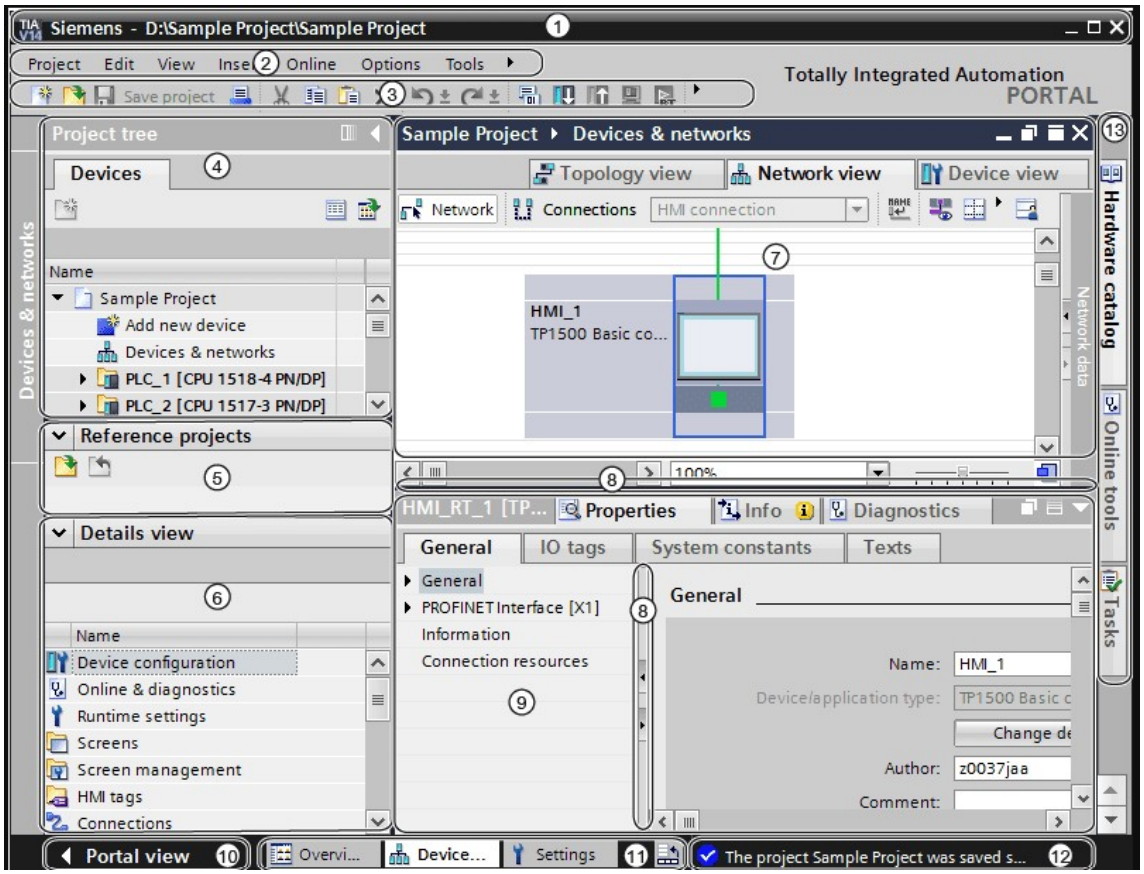


FIGURE 1.9 – Exemple des composants d'une vue de projet

1	Barre de titre	7	Lieu de travail
2	Barre de menu	8	Séparateurs (diviseurs)
3	barre d'outils	9	Fenêtre de l'inspecteur
4	Arborescence du projet	10	Passer à la vue du portail
5	Projets de référence	11	Barre de l'éditeur
6	Vue des détails	12	Barre d'état avec indicateur de progression
13	Fiches de tâches		

1.5 Conclusion

Dans ce chapitre ont posé les bases essentielles à la compréhension et à la mise en œuvre d'un système d'automatisation industrielle. On a d'abord mis en lumière l'automatisation industrielle comme un impératif stratégique pour l'efficacité, la précision et la sécurité dans la fabrication moderne. Ensuite, on a défini l'API comme le cœur du contrôle industriel, en détaillant son architecture fondamentale (CPU, mémoire, E/S, alimentation, bus de communication) et son cycle de balayage opérationnel, soulignant le comportement déterministe essentiel au contrôle en temps réel.

Le Siemens S7-1500 a été présenté comme un contrôleur haute performance, modulaire et polyvalent, dont les caractéristiques techniques spécifiques (puissance de traitement, mémoire, types d'E/S, interfaces de communication, fonctions avancées) en font un choix idéal pour le projet de système de tri, notamment grâce à sa vitesse, ses capacités de communication et ses fonctionnalités intégrées.

Enfin, le TIA Portal a été exploré en tant qu'environnement de développement intégré complet de Siemens, mettant en évidence ses composants pour la configuration, le diagnostic et la simulation, ainsi que son support des langages de programmation IEC 61131-3, permettant un développement de projet flexible et efficace.

Ayant établi une base solide concernant l'API Siemens S7-1500 et le TIA Portal, le chapitre suivant approfondira le rôle crucial de la simulation dans l'automatisation industrielle. Il se concentrera spécifiquement sur Factory I/O, un puissant simulateur d'usine virtuelle 3D, et démontrera comment il s'intègre de manière transparente avec le S7-1500 et le TIA Portal pour permettre des tests, une validation et une optimisation robustes du système de tri avant sa mise en œuvre physique.

CHAPITRE 2

LOGICIEL DE SIMULATION DES SYSTÈMES AUTOMATISÉS ET D'USINES VIRTUELS FACTORY I/O

2.1 Introduction

La simulation est devenue un outil indispensable dans l'automatisation manufacturière moderne, offrant aux fabricants la possibilité de modéliser, tester et optimiser leurs processus dans un environnement virtuel avant toute implémentation physique[17]. Elle joue un rôle central dans la conception, le développement et l'optimisation continue des systèmes automatisés, depuis la planification initiale des dépenses d'investissement jusqu'à l'optimisation des itinéraires de production.[18]

Les avantages de la simulation sont multiples et significatifs :

— **Validation du programme API :** Elle permet de tester et de déboguer les systèmes de contrôle dans un environnement virtuel, évitant ainsi les coûts et les risques associés aux modifications dans des environnements de production réels.[18]

— **Détection précoce des erreurs :** La simulation aide à identifier les goulots d'étranglement, les défauts de conception, les blocages et les collisions dès les premières phases de conception.[17]

— **Réduction des coûts de mise en service :** Les fabricants peuvent éviter des dépenses inutiles en affinant virtuellement les processus avant d'investir dans de nouveaux équipements ou de réorganiser les aménagements. La mise en service virtuelle valide les performances avant le déploiement réel.[17]

— **Formation des opérateurs :** La simulation offre une plateforme sûre et

efficace pour former le personnel aux nouveaux processus et équipements avant que le système physique ne soit opérationnel.[18]

— **Visualisation claire du fonctionnement du système :** Elle fournit un environnement 3D interactif pour visualiser et comprendre des processus industriels complexes.[19]

— **Possibilité de tester différents scénarios et optimisations :** La simulation permet d'explorer des solutions innovantes et de réaliser des analyses de type "et si" sans perturber les opérations en cours ni risquer d'endommager l'équipement réel.[17]

Les bénéfices de la simulation, tels que la réduction des coûts, la détection des erreurs et la formation , révèlent un changement fondamental dans la méthodologie d'ingénierie. En créant un "jumeau numérique" , la simulation permet une réduction significative des risques opérationnels et des investissements en capital avant tout engagement physique. Cette atténuation des risques, combinée à la capacité de tester rapidement de multiples scénarios, accélère fondamentalement le cycle d'innovation. Les entreprises peuvent expérimenter de nouvelles conceptions ou des changements de processus radicaux virtuellement, apprendre rapidement de leurs échecs dans un environnement sûr, et commercialiser des solutions optimisées plus rapidement. Cela positionne la simulation comme un catalyseur stratégique pour le développement agile et l'amélioration continue dans l'automatisation industrielle.

Ce chapitre vise à présenter Factory I/O comme un logiciel de simulation de premier plan pour les systèmes automatisés et les usines virtuelles. Il détaillera ses fonctionnalités, y compris sa riche bibliothèque d'équipements industriels, son moteur physique réaliste et son interface utilisateur. Un objectif central est d'expliquer l'intégration transparente de Factory I/O avec l'API Siemens S7-1500 (via PLCSIM Advanced) et TIA Portal, y compris le processus crucial de mappage des E/S virtuelles. Enfin, il décrira la méthodologie de conception et de modélisation du système de tri de caisses dans Factory I/O et soulignera les avantages significatifs que cette approche de simulation apporte au projet.

2.2 Présentation et Fonctionnalités de Factory I/O

2.2.1 Aperçu général de Factory I/O

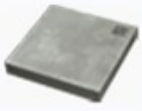
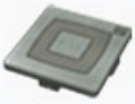
Factory I/O est un simulateur 3D d'API (Automate Programmable Industriel) qui offre une plateforme interactive pour le développement, le test et le débogage de systèmes de contrôle dans un environnement industriel virtuel. Ce logiciel permet aux utilisateurs de créer et de simuler des processus industriels réels, incluant l'automatisation, la robotique et les systèmes mécatroniques.Il se distingue par son environnement 3D intuitif et interactif, facilitant la visualisation et la compréhension du comportement des processus industriels complexes.[19]

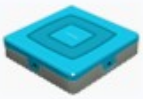
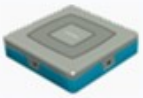
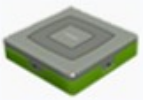
2.2.2 Bibliothèque d'équipements industriels

Factory I/O dispose d'une bibliothèque complète de modèles 3D et de composants, basés sur des équipements industriels courants. Ces éléments sont organisés en diverses catégories, notamment :

— **Articles :** Boîtes, palettes, Boîte empilable...[20]

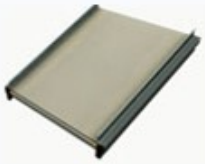
<table>
<tr><td rowspan="2">Boîtes
Il existe quatre types de boîtes différents, chacune ayant une taille et un poids différents.</td><td>Petit
8 kg</td><td>Moyen
10 kg</td><td>Grand
15 kg</td><td>Caisse de palettisation
3 kg</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="2">Palettes
Palettes en bois utilisées pour empiler et transporter toutes sortes de marchandises.</td><td colspan="2">Palette carrée de
20 kg</td><td colspan="2">Palette
20 kg</td></tr>
<tr><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td rowspan="2">Boîte empilable
Les boîtes empilables sont utilisées pour transporter des articles tels que des matières premières, des bases de produits et des couvercles de produits.</td><td colspan="4">Caisse empilable
15 kg</td></tr>
<tr><td colspan="4"></td></tr>
</table>

Matière première La matière première utilisée pour la fabrication des couvercles et des bases de produits dans le centre d'usinage est disponible en trois types de matériaux : plastique bleu ou vert et métal.	**Matière première bleue** 8 kg	**Matière première verte** 8 kg	**Matière première métallique** 10 kg
			
Couvercle du produit Pièce pouvant être assemblée dans une base de produit pour produire un produit final.	**Couvercle de produit bleu** 5 kg	**Couvercle de produit vert** 5 kg	**Couvercle pour produit métallique** 7 kg
			
Base de produits Pièce pouvant être assemblée en un Produit Lit pour produire un Produit Final.	**Base de produit bleu** 7 kg	**Base de produit verte** 7 kg	**Base de produit métallique** 9 kg
			

Produit final Créé en assemblant une base et un couvercle.	**Couvercle bleu - Base bleue** 12 kg	**Couvercle bleu - Base verte** 12 kg	**Couvercle bleu - Base en métal** 14 kg
			
	Couvercle vert - Base bleue 12 kg	**Couvercle vert - Base verte** 12 kg	**Couvercle vert - Base en métal** 14 kg
			
	Couvercle en métal - Base bleue 14 kg	**Couvercle en métal - Base verte** 14 kg	**Couvercle en métal - Base en métal** 16 kg
			

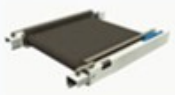
— **Pièces de charge lourde :** Cette catégorie comprend toutes les pièces adaptées à la manutention de charges lourdes. L'équipement est robuste, large, de faible hauteur et fonctionne à faible vitesse.[20]

Convoyeur à rouleaux Convoyeur à rouleaux robuste, peut être contrôlé par des valeurs numériques et analogiques selon la configuration sélectionnée.	• Rayon du rouleau : 0,046 m • Longueurs disponibles : 2, 4 et 6 m • Vitesse de transport maximale : 0,45 m/s (numérique) 0,8 m/s (analogique)	
Convoyeur à rouleaux courbes Convoyeur à rouleaux courbes robuste, peut être contrôlé par des valeurs numériques et analogiques selon la configuration sélectionnée.	• Rayon du rouleau : 0,046 m • Périmètre extérieur (au bord du rouleau) : 2,94 m • Vitesse de transport maximale : 0,45 m/s (numérique) 0,8 m/s (analogique)	
Convoyeur de chargement Convoyeur robuste, principalement utilisé pour charger/décharger des marchandises sur un transstockeur . Contrôle possible par valeurs numériques ou analogiques.	• Rayon du rouleau : 0,046 m • Longueurs disponibles : 2, 4 et 6 m • Vitesse de transport maximale : 0,45 m/s (numérique) 0,8 m/s (analogique)	
Arrêt du rouleau Ce dispositif à commande pneumatique peut être utilisé pour arrêter, accumuler ou empêcher les collisions de matériaux sur les convoyeurs.	• Normalement en panne • Course : 0,1 m	
Rouleau libre Rouleau non motorisé utilisé pour le montage et le réglage du convoyeur.		

Transfert en chaîne Utilisé pour transférer des marchandises sur des convoyeurs adjacents, il fonctionne mieux avec des granulés carrés. Il est composé de rouleaux de chargement et de trois chaînes.	• Course de la chaîne : 0,04 m • Vitesse de transport maximale : 0,45 m/s • Vitesse maximale de la chaîne : 0,45 m/s	
Platine Table tournante motorisée robuste, généralement utilisée pour le tri de palettes. Elle est équipée de rouleaux périphériques à rotation libre et de capteurs préinstallés.	• Rayon du rouleau : 0,045 m • Vitesse de transport maximale : 0,45 m/s • Vitesse de rotation de la table : 0,7 rad/s • Plage de capteurs capacitifs : 0 - 0,1 m	
Convoyeur à goulotte bas Convoyeur à goulotte droite, couramment utilisé pour expédier des palettes.		

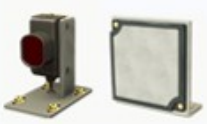
— **Pièces de charge légère :** La catégorie Pièces pour charges légères comprend des pièces adaptées à la manutention de charges légères. Conçues pour exécuter rapidement une tâche et suivre un flux de travail intensif, ces pièces sont légères et fonctionnent à grande vitesse.[20]

Convoyeurs à bande Les convoyeurs à bande sont utilisés pour le transport de charges légères. Ils peuvent être contrôlés par des valeurs numériques ou analogiques.	• Longueurs disponibles : 2, 4 et 6 m • Vitesse de transport maximale : 0,6 m/s (numérique) 3 m/s (analogique)	
Convoyeur à bande courbe Les convoyeurs à bande sont utilisés pour le transport de charges légères. Ils peuvent être contrôlés par des valeurs numériques ou analogiques.		

Porte de convoyeur à bande Convoyeur à bande motorisé à glissières conçu pour le passage du personnel. Équipé d'un capteur fermé, il peut être commandé par des valeurs numériques ou analogiques.	• Angle d'ouverture possible : 100° • Vitesse de transport maximale : 0,6 m/s (numérique), 3 m/s (analogique)	
Convoyeur à bande inclinée Les convoyeurs à bande inclinés sont utilisés pour le transport de charges légères. Ils peuvent être contrôlés par des valeurs numériques ou analogiques.		
Balance à convoyeur Balance à convoyeur à grande vitesse utilisée pour le contrôle du poids. Mesure différentes plages de poids selon la configuration sélectionnée.	• Vitesse de transport maximale : 0,6 m/s • Capacités : 20 et 100 kg	
Convoyeur à éperons droits Convoyeur multi-bandes d'alimentation/sortie, couramment utilisé pour le transfert précis des charges dans les convoyeurs à bande. Des valeurs numériques et analogiques permettent de contrôler le convoyeur.	• Vitesse de transport maximale : 0,8 m/s (numérique) 3 m/s (analogique)	
Trieur à bras pivotant Un bras de déviation à 45°, actionné par un motoréducteur, est équipé d'une courroie permettant de dévier les articles transportés vers la pièce suivante. Le bras peut pivoter à gauche ou à droite	• Vitesse de la bande : 2 m/s • Vitesse angulaire du bras : 5 rad/s	

Trieur à roue escamotable Un trieur à roues pop-up double face à 45° utilisé pour détourner les articles vers trois directions différentes via des rouleaux pivotants.	• Rayon de roue : 0,05 m • Vitesse de transport : 1,5 m/s • Normalement centré • Course des roues : 0,003 m	
Poussoir Trieur pneumatique à poussoir équipé de deux capteurs Reed indiquant les limites avant et arrière. Il comprend également une servovalve permettant de régler et de mesurer la position de la tige. Le contrôle peut être effectué par des valeurs numériques ou analogiques selon la configuration sélectionnée.	• Vitesse par défaut : 1 m/s • Vitesse rapide : 4 m/s • Course : 0,9 m	
Convoyeur à goulotte Convoyeur à goulotte droite, principalement utilisé pour expédier des articles à partir de convoyeurs à bande.		
Arrêter la lame La lame d'arrêt est un dispositif à commande pneumatique utilisé pour arrêter ou accumuler du matériau.	• Normalement en panne • Course : 0,12 m	
Support et coin métallique Le support est une structure métallique servant de barrière de hauteur et permettant de fixer des capteurs. Le coin métallique peut servir à plusieurs fins, notamment pour le positionnement des capteurs.	 Bracket  Metal Corner	
Barres de positionnement Dispositif permettant de positionner successivement des objets au même endroit par serrage. Disponible en deux configurations : gauche et droite. Il est couramment utilisé avec les systèmes Pick and Place.	• Course verticale : 0,373 m • Course de la pince : 0,48 m	

— **Capteurs :** Les capteurs sont utilisés pour détecter la présence de pièces, mesurer les distances et même déterminer le type d'une pièce. [20]

Capteur capacitif Capteur de proximité permettant la détection rapprochée de tout matériau. Il est équipé d'une LED indiquant la présence d'un objet à portée. La valeur de sortie peut être numérique ou analogique.	• LED : verte (détection) • Matériaux détectables : solides et liquides. • Plage de détection : 0 - 0,2 m	
Capteur diffus Capteur photoélectrique diffus capable de détecter n'importe quel objet solide.	• LED : rouge (détection) • Matériaux détectables : solides • Portée de détection : 0 - 1,6 m	
Capteur inductif Capteur de proximité utilisé pour la détection rapprochée de matériaux conducteurs. Il est équipé d'une LED indiquant la présence d'un objet à proximité. La valeur de sortie peut être numérique ou analogique.	• LED : rouge (détection) • Matériaux détectables : conducteurs • Plage de détection : 0 - 0,1 m	
Capteur de vision Le capteur de vision reconnaît les matières premières , les couvercles et les bases des produits , ainsi que leurs couleurs respectives.	• LED : rouge (détection) • Matériaux détectables : matières premières, bases de produits, couvercles de produits • Portée de détection : 0,3 - 2 m	
Capteur et réflecteur rétroréfléchissants Contrairement aux autres capteurs, le capteur rétroréfléchissant nécessite un réflecteur. Pour fonctionner correctement, il doit être aligné avec le réflecteur. Il est équipé de deux LED qui indiquent l'alignement correct (vert) et l'état de détection (jaune).	• LED verte : alignée avec le réflecteur • LED jaune : faisceau lumineux non interrompu • Matériaux détectables : solides • Portée de détection : 0 - 6 m	

Réseau lumineux Un ensemble de faisceaux lumineux parallèles permet de créer un champ de détection. Pour assurer la synchronisation des deux appareils (émetteur et récepteur), ceux-ci doivent être correctement alignés et face à face. Une fois cet alignement assuré (indiqué par une LED verte), tous les faisceaux peuvent être interrompus sans interrompre la synchronisation. Peut-être configuré numérique, digital ou analogique.	• Nombre de poutres : 8 • Portée : 1,5 m • Matériaux détectables : solides • Valeur analogique = 10 * nombre de faisceaux interrompus / 8	
Codeur incrémental Plusieurs composants incluent un codeur incrémental, activable par un clic droit sur le composant et en choisissant « Utiliser le codeur » dans le menu contextuel. Ce codeur utilise deux bits (A et B) pour produire deux formes d'onde déphasées de 90 degrés (en quadrature). La résolution du codeur est de 0,0225 m, calculée pour une vitesse de 0,45 m/s à 20 Hz (vitesse par défaut d'un convoyeur à rouleaux). La fréquence de 20 Hz a été choisie car elle est facilement prise en charge par la plupart des pilotes d'E/S.	• Convoyeur de chargement • Convoyeur à rouleaux (2/4/6 m) • Convoyeur à rouleaux courbes • Convoyeur à bande (2/4/6 m) • Convoyeur à bande courbe • Convoyeur à bande inclinée	

— **Opérateurs :** Arrêts d'urgence, boutons-poussoirs, voyants lumineux... [20]

Arrêt d'urgence Gâchette rouge à deux positions et bouton-poussoir non lumineux (tête champignon). Généralement utilisé en cas d'urgence.	• Type de contact normalement fermé	
Boutons poussoirs Bouton-poussoir lumineux disponible en trois couleurs (vert : démarrage, jaune : réinitialisation et rouge : arrêt). Il peut être utilisé en mode momentané ou alternatif, selon la configuration choisie.	• Le bouton d'arrêt est normalement fermé	

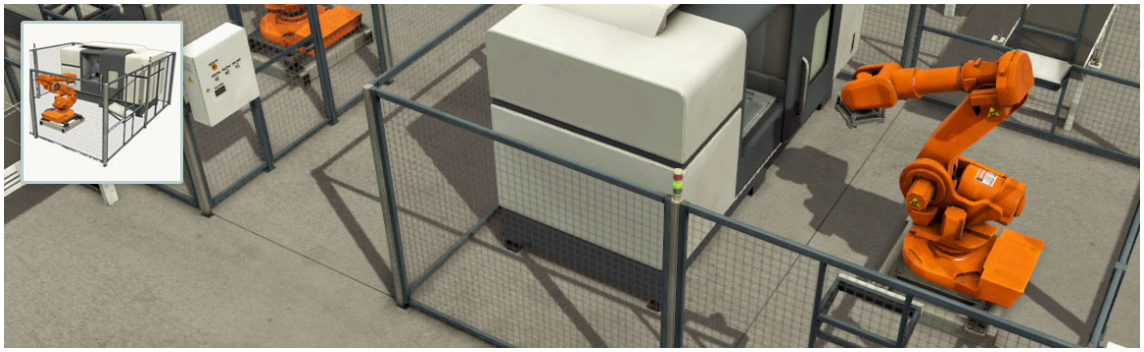
Indicateurs lumineux Voyant lumineux monté sur panneau. Couramment utilisé sur les tableaux de bord pour des applications de sécurité ou pour indiquer l'état.	• Disponible en quatre couleurs (bleu, rouge, vert et jaune)	
Sélecteur Sélecteur non lumineux principalement utilisé pour définir l'état actuel d'une variable ou d'une tâche. Faites un clic gauche et faites glisser pour basculer entre les états.	• L'état 0 est sélectionné par défaut.	
Potentiomètre Permet de générer une valeur analogique proportionnelle à la rotation du bouton.	• La configuration sélectionnée permet de définir la plage de valeurs.	
Affichage numérique Permet d'afficher des valeurs numériques pendant la simulation.	• La configuration sélectionnée définit la plage de valeurs et le type de données.	
Tableau électrique Utilisé pour concevoir un tableau électrique avec des opérateurs.		
Colonne Structure métallique généralement utilisée pour supporter un tableau électrique.		

Stations :

Centres d'usinage :

Le centre d'usinage est une station utilisée pour fabriquer des couvercles et des fonds à partir de matières premières. Tout d'abord, le robot articulé attend que la matière première soit placée au poste d'entrée. Lorsqu'une nouvelle matière est détectée, elle est chargée dans la machine CNC, qui lance la fabrication d'une pièce. Chaque type de pièce nécessite un temps de production différent (couvercles : 6 secondes ; fonds : 3 secondes). Une fois l'opération terminée, le robot place la pièce

au poste de sortie. [20]



— **Urgence :** déclenche un arrêt d'urgence, interrompant le robot articulé et la CNC. Après un arrêt d'urgence, la station ne peut être réinitialisée qu'en désactivant le bouton d'urgence et en appuyant sur le bouton de réinitialisation.

— **Démarrer :** démarre la station.

— **Arrêt :** arrête le robot articulé et la commande numérique. Pour redémarrer, appuyez sur le bouton Démarrer ou Réinitialiser.

— **Réinitialiser :** réinitialise la station.

Deux voyants placés sur le dessus de la protection fournissent des informations sur l'état actuel.

— **Feu vert :** pas occupé

— **Feu jaune :** occupé

— **Voyant rouge :** erreur (élément incorrect détecté à l'entrée)

Ascenseur :

Ascenseur robuste à chaîne, utilisé pour le transport de toutes sortes de marchandises entre les étages. Il est équipé de deux capteurs rétroréfléchissants, placés à chaque extrémité de la plateforme. Il peut être contrôlé par des valeurs numériques ou analogiques.[20]



- **Rayon du rouleau :** 45 mm
- **Course de la plate-forme :** 7 m
- **Vitesse de la plateforme :** 0,68 m/s
- **Vitesse de transport maximale :** 0,45 m/s (numérique) ; 0,8 m/s (analogique)
- **Capteur rétroréfléchissant LED :** vert (détection du réflecteur), jaune (faisceau non interrompu)

Pick and place :

Station de prélèvement et de placement à portique à trois axes commandés par servomoteurs. Souvent utilisée pour déplacer des charges légères (par exemple, des cartons) vers d'autres convoyeurs ou palettes.

Le Pick and Place possède quatre degrés de liberté : trois correspondent au mouvement linéaire des axes et un autre à la rotation de la pince. Celle-ci est actionnée par des ventouses et intègre un capteur de proximité. Le Pick and Place peut être contrôlé par des valeurs numériques et analogiques, selon la configuration sélectionnée. Avec une E/S numérique, le mouvement des axes est incrémental (pas à pas) à chaque front montant de la valeur de la balise de contrôle. [20]



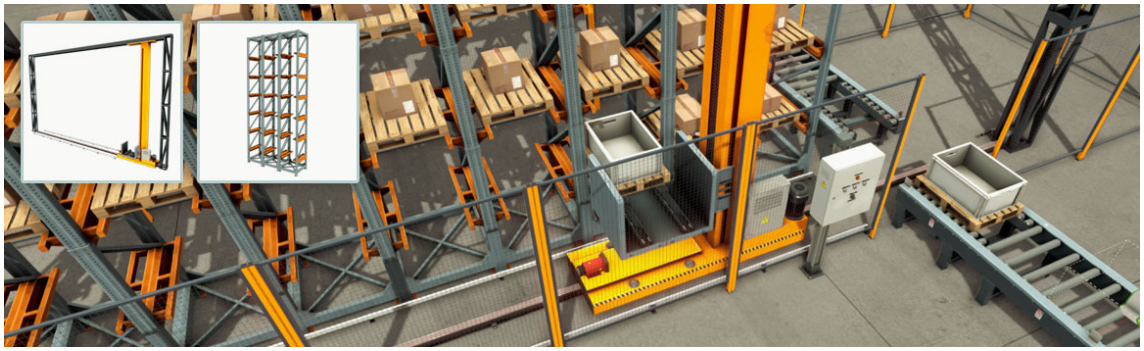
- **Course de l'axe Y :** 1,25 m
- **Course de l'axe X :** 2,125 m
- **Course de l'axe Z :** 0,5 m
- **Pas :** 0,125 m
- **Vitesse du faisceau :** 1,5 m/s
- **Vitesse angulaire de la pince :** 4,6 rad/s

Transstockeur et rayonnage :

Transstockeur sur rails utilisé pour le stockage de charges lourdes. Il comprend un chariot, une plateforme verticale et deux fourches coulissantes.

Deux télémètres laser placés sur le chariot et la plateforme mesurent la position horizontale et verticale de cette dernière. Les racks sont des cadres verticaux en acier reliés par des poutres horizontales en acier destinés au stockage des charges. Le type de rack disponible est un rack simple profondeur, également appelé rack sélectif, qui permet de stocker des charges sur une seule palette. Les charges peuvent être stockées des deux côtés du rack.

Chaque rack doit être aligné avec l'une des extrémités du rail, ce qui permet au transstockeur de s'arrêter à la bonne position. Le transstockeur peut être contrôlé par des valeurs numériques, numériques et analogiques. [20]



- **Course des fourches :** 1,2 m
- **Course du chariot :** 10,5 m
- **Course de la plate-forme :** 6,625 m
- **Vitesse du chariot :** 1,4 m/s
- **Vitesse des fourches :** 0,5 m/s
- **Vitesse de la plateforme :** 1,7 m/s
- **Nombre de cellules :** 18
- **Numérique :** la cellule cible peut être définie par une valeur entière comprise entre 1 et 54. Si cette valeur est égale à zéro, le transstockeur s'arrête à la position actuelle. En revanche, si elle est supérieure à 54, il se déplace vers la position de repos (55).
- **Analogique :** la position cible/actuelle de chaque axe peut être définie/mesurée avec des valeurs analogiques.
- **Numérique :** chaque numéro de cellule est codé par cinq valeurs numériques.

Palettiseur :

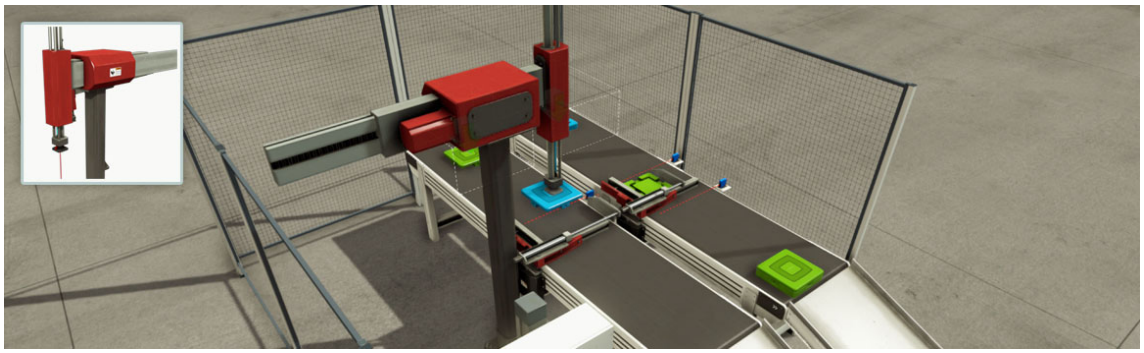
Palettiseur de haut niveau utilisé pour empiler des boîtes en carton (Palletizing Box) sur des palettes.[20]



- **Course du poussoir :** 0,88 m
- **Course de l'ascenseur :** 1,75 m
- **Vitesse de l'ascenseur :** 2 m/s

Pick and Place à deux axes :

Cette pièce permet d'assembler des couvercles sur des bases ou de déplacer des articles. Pour garantir un ajustement parfait, les bases et les couvercles doivent être correctement alignés à l'aide de barres de positionnement. [20]



- **Course de l'axe X :** 1,125 m
- **Course de l'axe Z :** 0,625 m
- **Vitesse du bras et du sélecteur :** 2 m/s

Réservoir :

Réservoir de liquide comprenant deux vannes de régulation et un capteur de niveau capacitif permettant de réguler le débit de liquide entrant et sortant. Les vannes de régulation sont commandées par des actionneurs pneumatiques réglables avec des signaux compris entre 0 et 10 V. Les capteurs capacitifs peuvent également être utilisés pour détecter les niveaux de liquide. Le réservoir de liquide est principalement destiné à la régulation de niveau et de débit à l'aide d'un PID. [20]



- **Hauteur :** 3 m
- **Diamètre :** 2 m
- **Rayon du tuyau de refoulement :** 0,125 m
- **Débit d'entrée max. :** 0,25 m^3/s
- **Débit de sortie max. :** 0,3543 m^3/s
- Les capteurs capacitifs peuvent détecter les liquides.

Dispositifs d'avertissement :

Pièces utilisées pour attirer l'attention sur certaines tâches et certains événements au moyen d'alertes visuelles ou sonores. [20]

Sirène d'alarme La sirène d'alarme est audible dans les environnements bruyants en raison du volume élevé et du hurlement pénétrant de la sirène.	
Lumière de pile Les voyants lumineux sont couramment utilisés comme indicateurs visuels de l'état et des processus d'une machine. Ils sont disponibles en trois couleurs : rouge, jaune et vert.	
Voyant d'avertissement Feu rotatif utile lorsqu'un signal d'avertissement visuel puissant est requis.	

— **Passerelles :**

Cette catégorie comprend les pièces utilisées pour construire des passages pour les ouvriers ou pour isoler des périmètres. La caméra à la première personne offre une perspective réaliste lors de l'utilisation de ces passages.[20]



— **Émetteur :**

Émet un objet à utiliser dans une scène. Tant qu'un objet est dans le volume de l'émetteur, aucun autre objet n'est émis. Vous pouvez choisir la pièce ou la base à émettre, l'intervalle entre les émissions, le nombre d'objets à émettre et la prise en compte ou non de la position et/ou de l'orientation aléatoires. Vous pouvez activer ou désactiver un émetteur en activant ou désactivant son étiquette.[20]



2.2.3 Physique réaliste et comportement des objets

Factory I/O simule de manière réaliste le comportement physique des objets, y compris la détection, le mouvement et les interactions. La simulation physique est hautement optimisée pour la précision et la stabilité, garantissant un comportement réaliste des pièces et des articles (par exemple, plus de palettes qui se tordent sur les convoyeurs à rouleaux [21]). Le logiciel prend en charge les E/S numériques et analogiques pour la plupart des pièces, permettant un contrôle réaliste (par exemple, une valeur numérique pour démarrer/arrêter un convoyeur, une valeur analogique pour peser des articles [22]).

Le logiciel intègre également une fonction d'"échelle de temps", qui permet d'exécuter les simulations au ralenti (0,1x) ou en avance rapide (jusqu'à 4x). Cette fonctionnalité est particulièrement utile pour l'analyse comportementale ou la génération de grandes quantités de données. La physique réaliste et les fonctions d'échelle de temps sont plus que de simples améliorations visuelles ; elles constituent de puissants outils de débogage et d'optimisation. Le ralenti permet aux ingénieurs d'observer méticuleusement les interactions subtiles et les problèmes de synchronisation qui seraient impossibles à détecter en temps réel sur du matériel physique. Inversement, l'avance rapide permet de tester rapidement de longs cycles opérationnels ou de soumettre la logique de contrôle à des scénarios de stress, identifiant rapidement les cas limites ou les goulots d'étranglement de performance. Cette capacité réduit considérablement le temps et les efforts nécessaires à la mise en service et au réglage fin, conduisant à des programmes de contrôle plus robustes et efficaces.[21]

2.2.4 Interface utilisateur de Factory I/O

Factory I/O dispose d'une interface conviviale et intuitive. Ses éléments clés comprennent :

— **Vue 3D :**L'espace de travail principal où le système virtuel est construit, visualisé et interagi.[23]

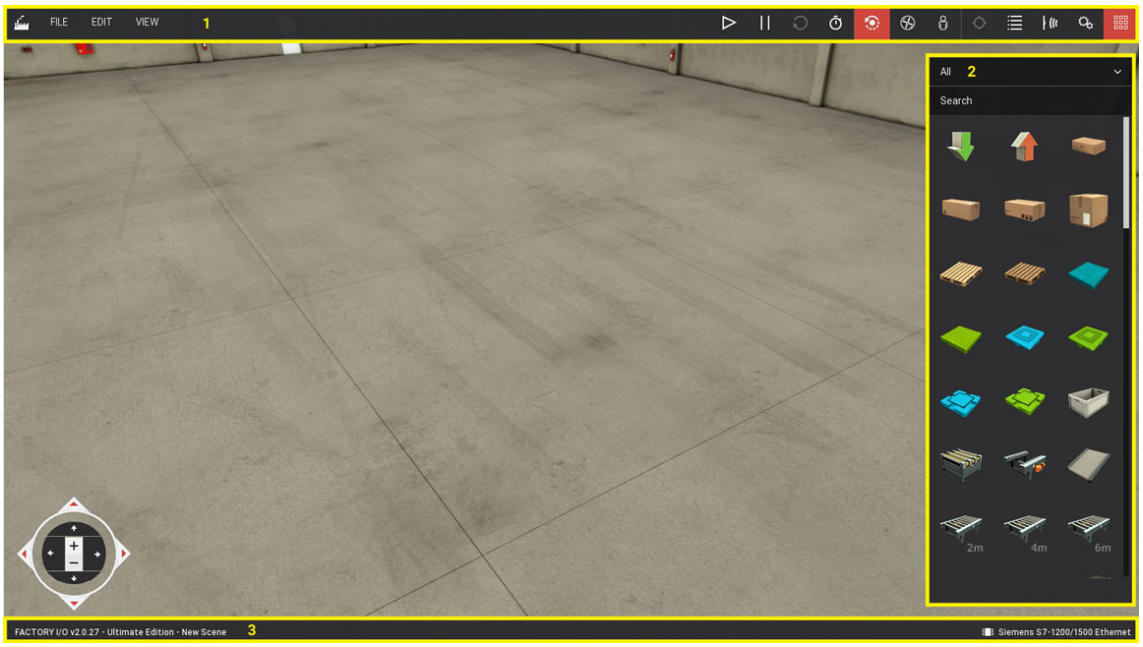


FIGURE 2.1 – l'interface utilisateur et le flux de travail général

— **Explorateur de scènes :** Organise la structure hiérarchique des objets au sein de la scène, facilitant la sélection et la navigation.[24]

— **Navigateur d'E/S :**Permet de mapper les entrées et sorties virtuelles aux variables de l'API, affichant les connexions configurées et les propriétés définies. Il affiche les tags, leurs définitions (nom, adresse) et leurs valeurs actuelles.[24]

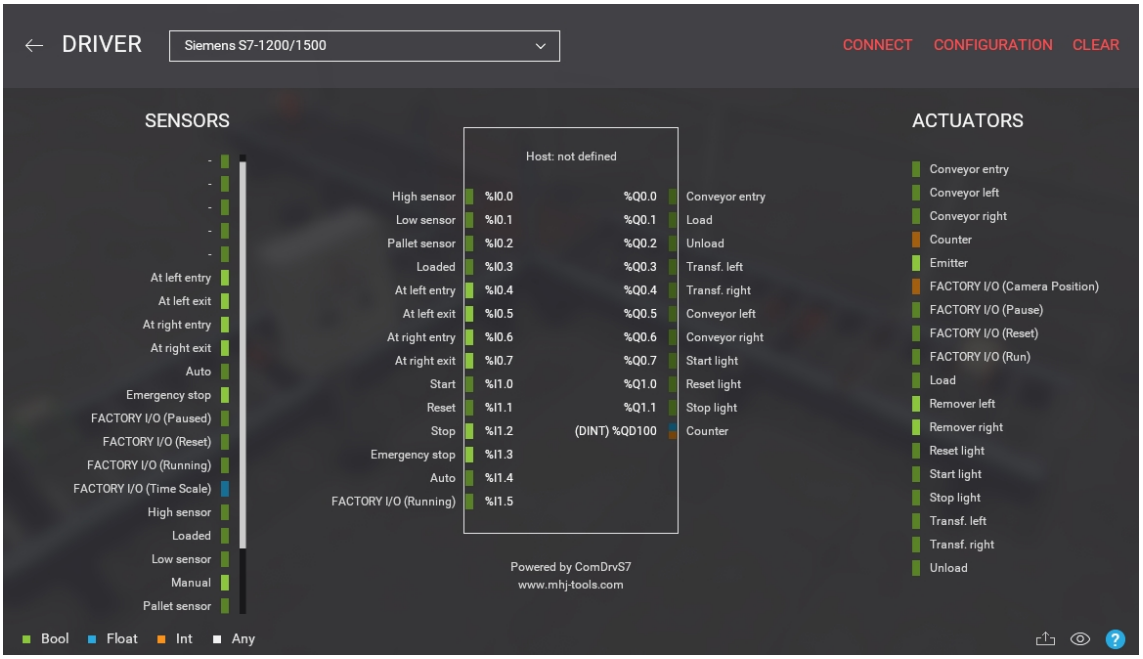


FIGURE 2.2 – L'interface d'E/S

— **Éditeur de propriétés :** Permet de configurer les paramètres des objets individuels (par exemple, la vitesse d'un convoyeur, la portée d'un capteur).[24]

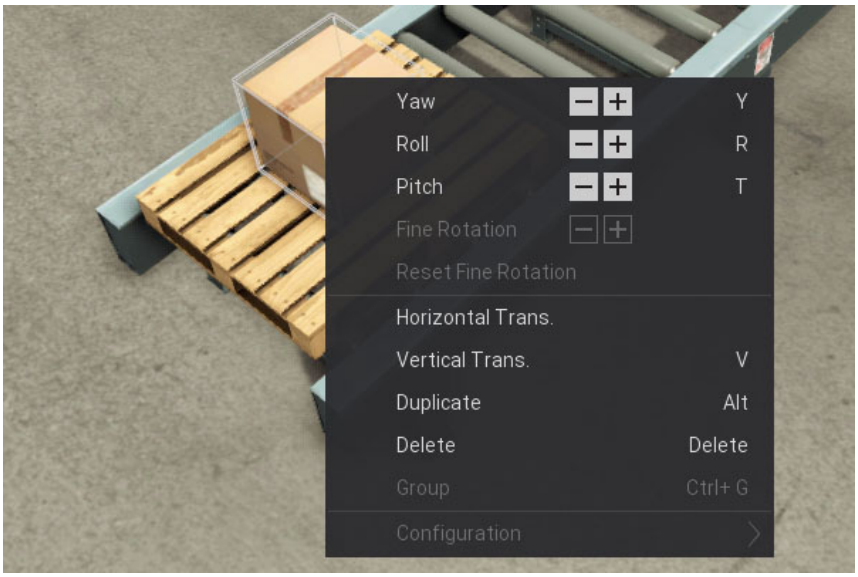


FIGURE 2.3 – L'interface liste des paramètres

— **Barre d'outils :** Offre un accès rapide aux fonctions principales comme les modes Exécution/Édition, Pause, Réinitialisation, contrôles de caméra et opérations de fichier.[23]



FIGURE 2.4 – Barre d'outil

— **Palette :**Affiche toutes les pièces disponibles dans Factory I/O, organisées par catégorie, à partir desquelles les utilisateurs peuvent glisser-déposer des composants dans la scène 3D.[23]



FIGURE 2.5 – Liste des objets au sein de la scène

2.2.5 Modes de fonctionnement

Factory I/O fonctionne selon deux modes distincts : le mode Édition et le mode Exécution.

— **Mode Édition (Edit Mode) :** Ce mode est utilisé pour ouvrir, enregistrer, créer et éditer des scènes. Les utilisateurs placent, transforment (traduisent, font pivoter) et modifient les pièces en les faisant glisser de la Palette vers l'environnement 3D. Il intègre des algorithmes de collision intelligents qui aident à placer les pièces de manière valide.[25]

— **Mode Exécution (Run Mode) :** Dans ce mode, la scène est simulée en temps réel et peut être contrôlée manuellement ou par un contrôleur externe (comme un API). Les utilisateurs peuvent interagir avec les pièces en cliquant et en les faisant glisser. La simulation peut être mise en pause, ralentie ou accélérée.[25]

2.3 Intégration de Factory I/O avec l' API S7-1500 et TIA Portal

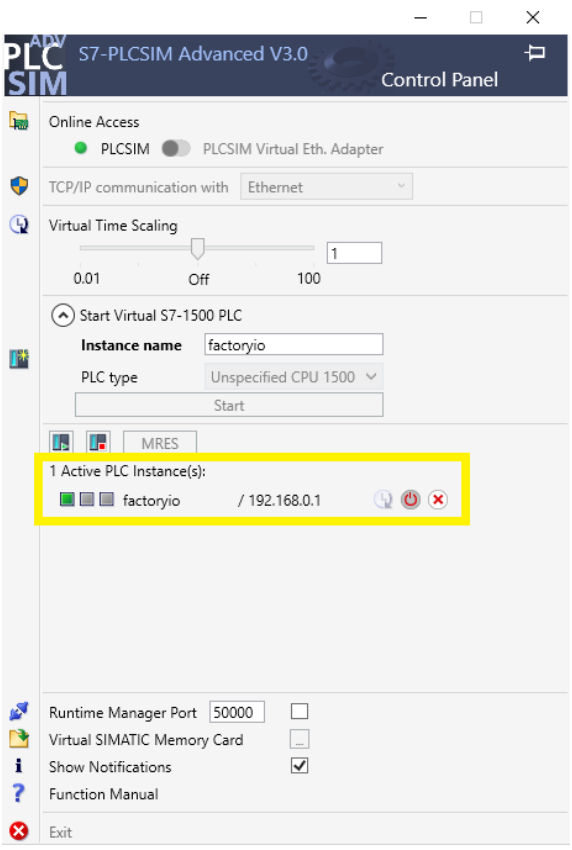
2.3.1 Principe de la communication API-Simulation

Factory I/O communique avec l'API (ou un simulateur d'API comme SIMATIC S7-PLCSIM Advanced) pour simuler l'interaction entre la logique de commande et le système physique virtuel. Cette communication repose sur le mappage des entrées/sorties virtuelles dans Factory I/O aux variables correspondantes dans le programme API. Lorsqu'un capteur est activé dans Factory I/O, son état d'entrée virtuelle est envoyé à l'API ; l'API traite cette information, met à jour sa logique interne et renvoie des commandes de sortie à Factory I/O, qui active alors les actionneurs virtuels.[26]

Configuration de la connexion :

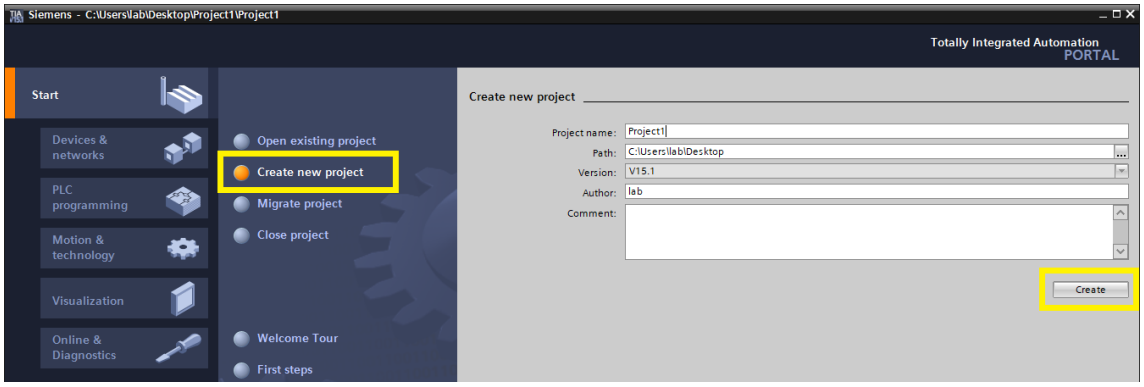
L'établissement d'une connexion entre Factory I/O et le simulateur d'API SIMATIC S7-PLCSIM Advanced via TIA Portal suit des étapes générales précises :

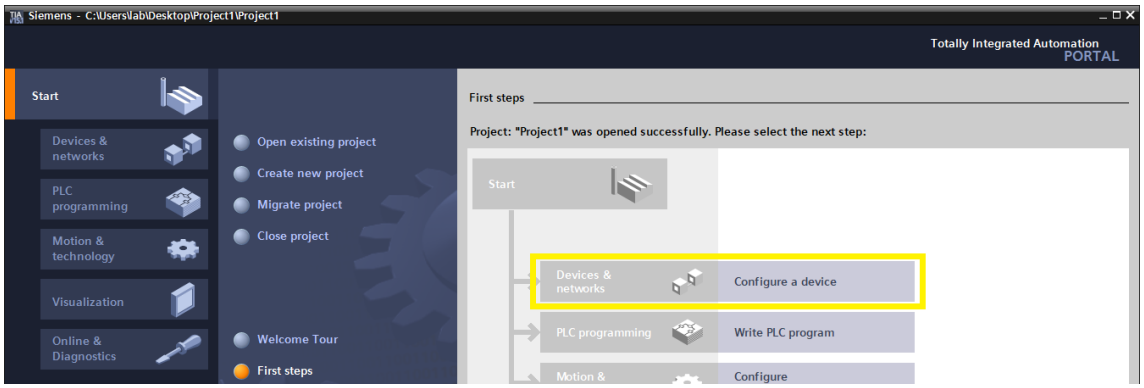
1. **Démarrage de PLCSIM Advanced :** Ouvrez le panneau de contrôle de PLCSIM Advanced et spécifiez un nom d'instance (par exemple, 'factoryio'). Démarrez l'instance.[11]

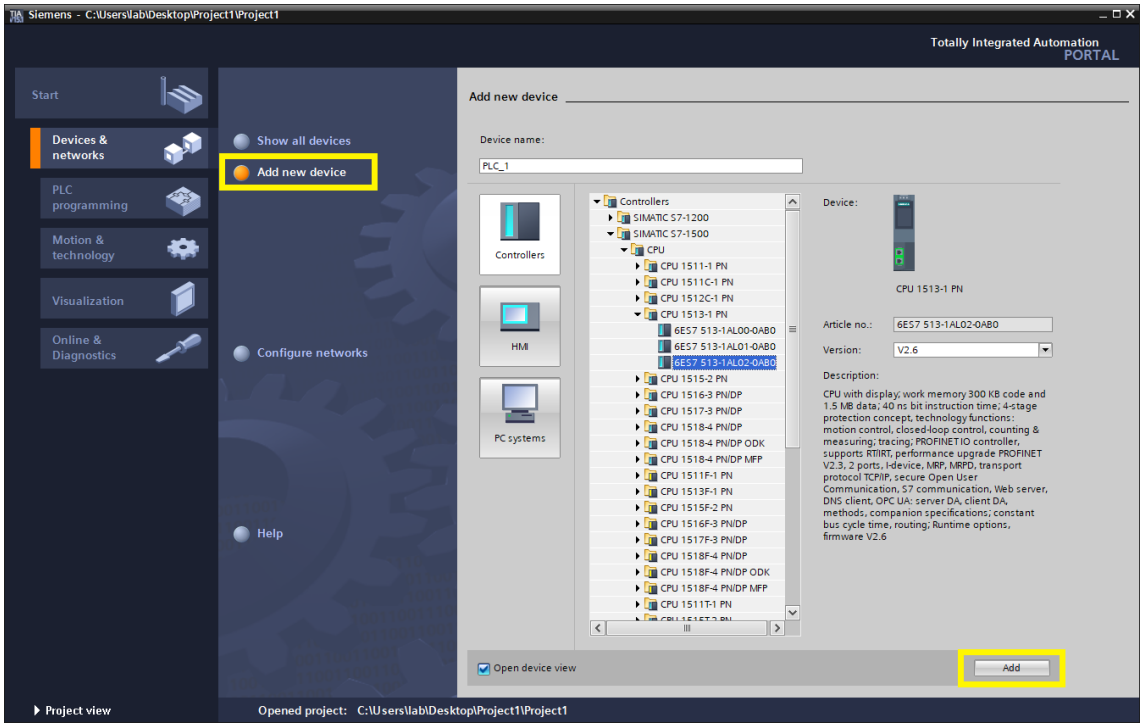


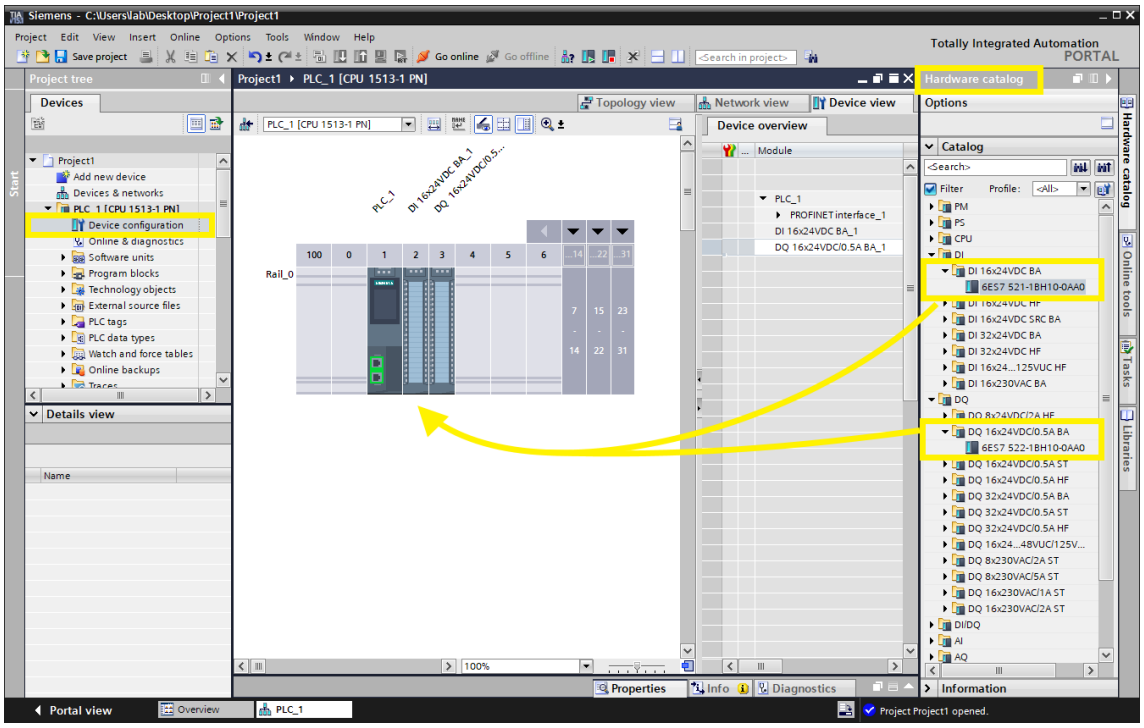
2. **Configuration du projet TIA Portal :** Ouvrez TIA Portal, créez un nouveau projet et ajoutez la CPU S7-1500 souhaitée (par exemple, S7-1513-1

PN) et les modules d'E/S pertinents (par exemple, DI 16x24VDC BA, DQ 16x24VDC/0.5A BA).[11]

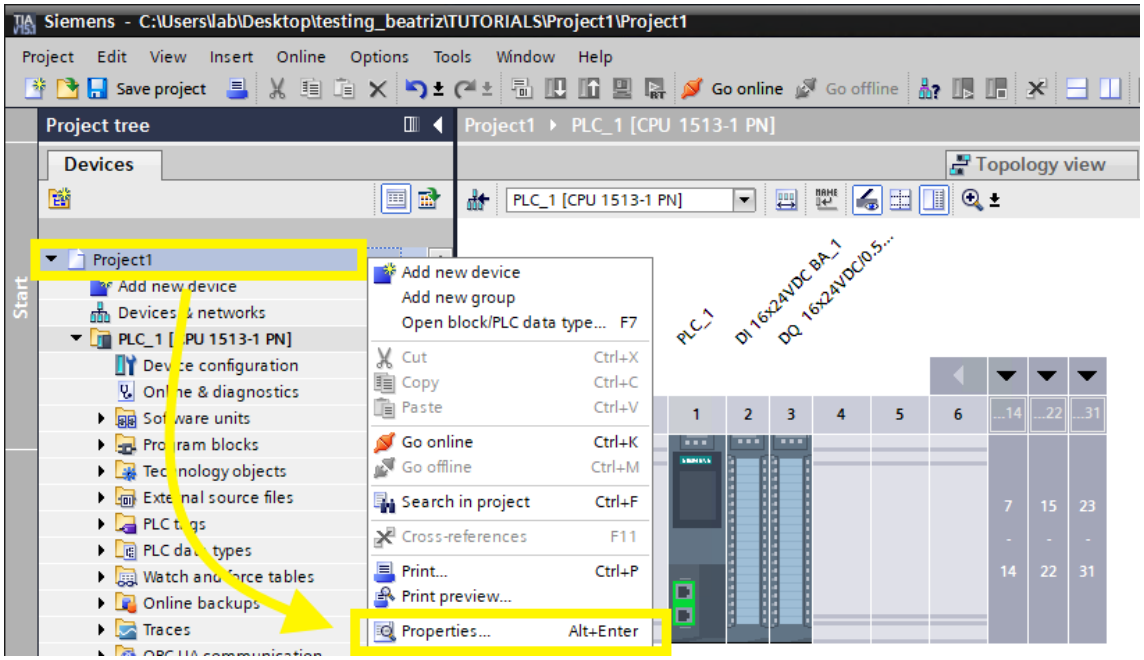


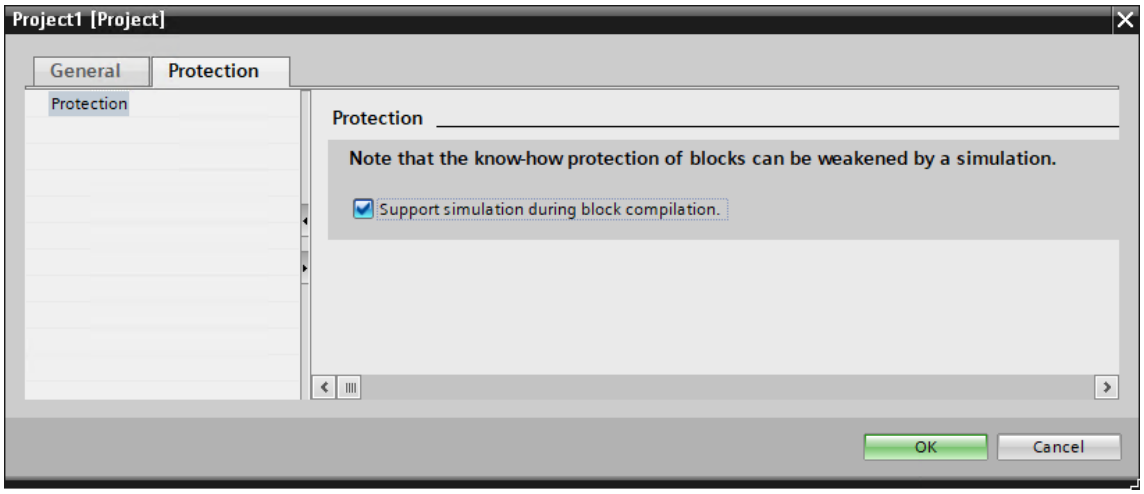




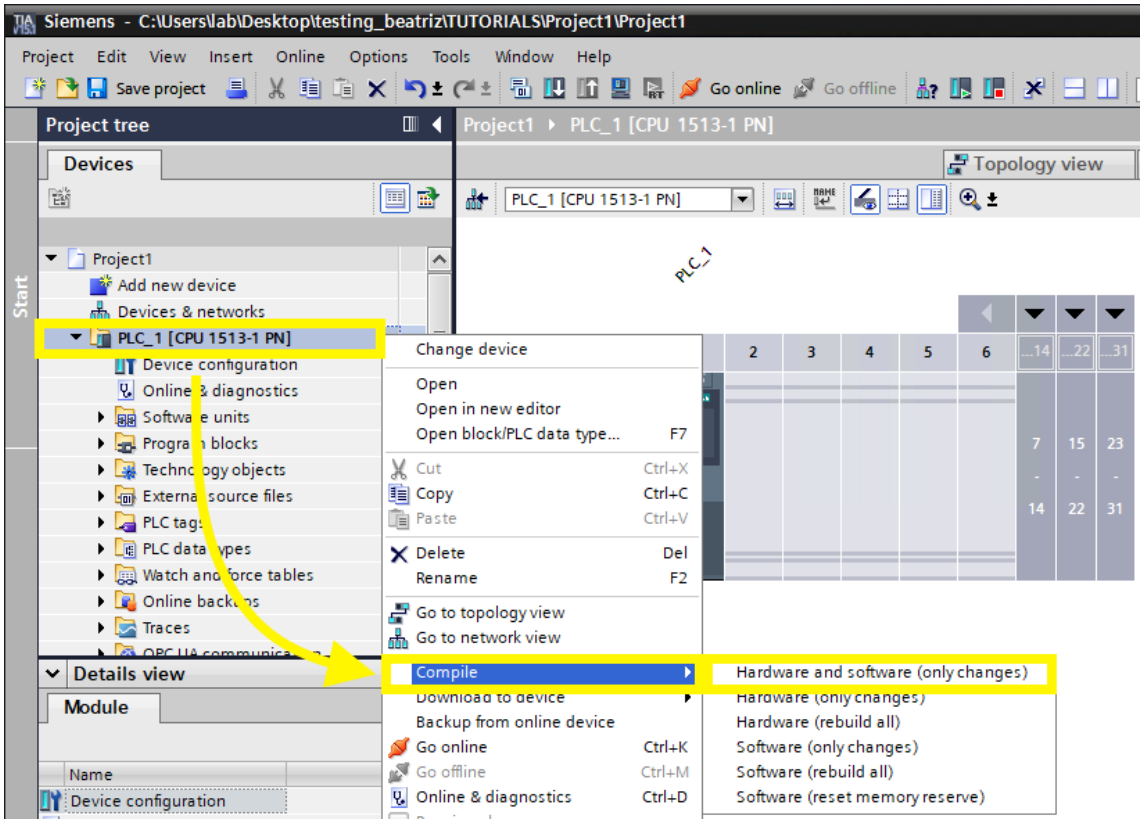


3. **Activation du support de simulation :** Dans TIA Portal, faites un clic droit sur le nom du projet, accédez à Propriétés > onglet Protection, et cochez "Support simulation during block compilation".[11] Pour un API réel, assurez-vous que "Accès complet (aucune protection)" est sélectionné dans les paramètres de l'API > Protection et sécurité.[26]



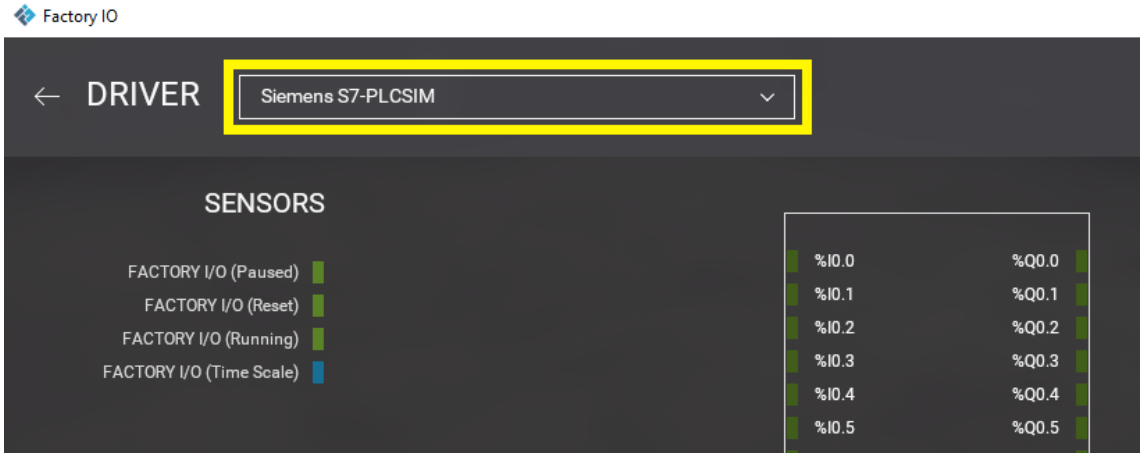


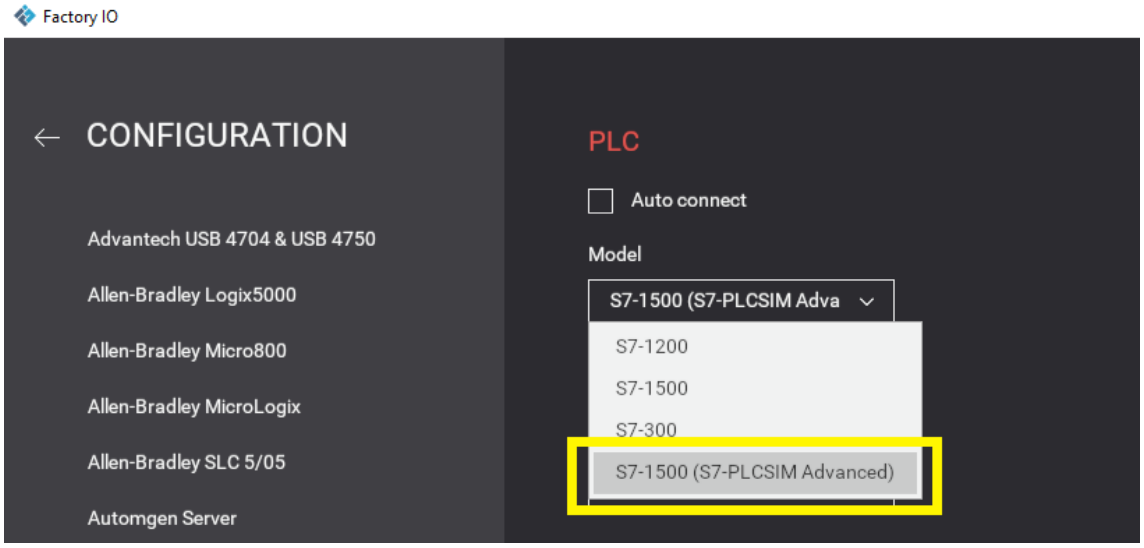
4. **Compilation et téléchargement vers PLCSIM Advanced :** Compilez les modifications matérielles et logicielles dans TIA Portal et téléchargez-les vers l'instance PLCSIM Advanced configurée. Sélectionnez "Démarrer le module" dans les résultats du chargement.[11]



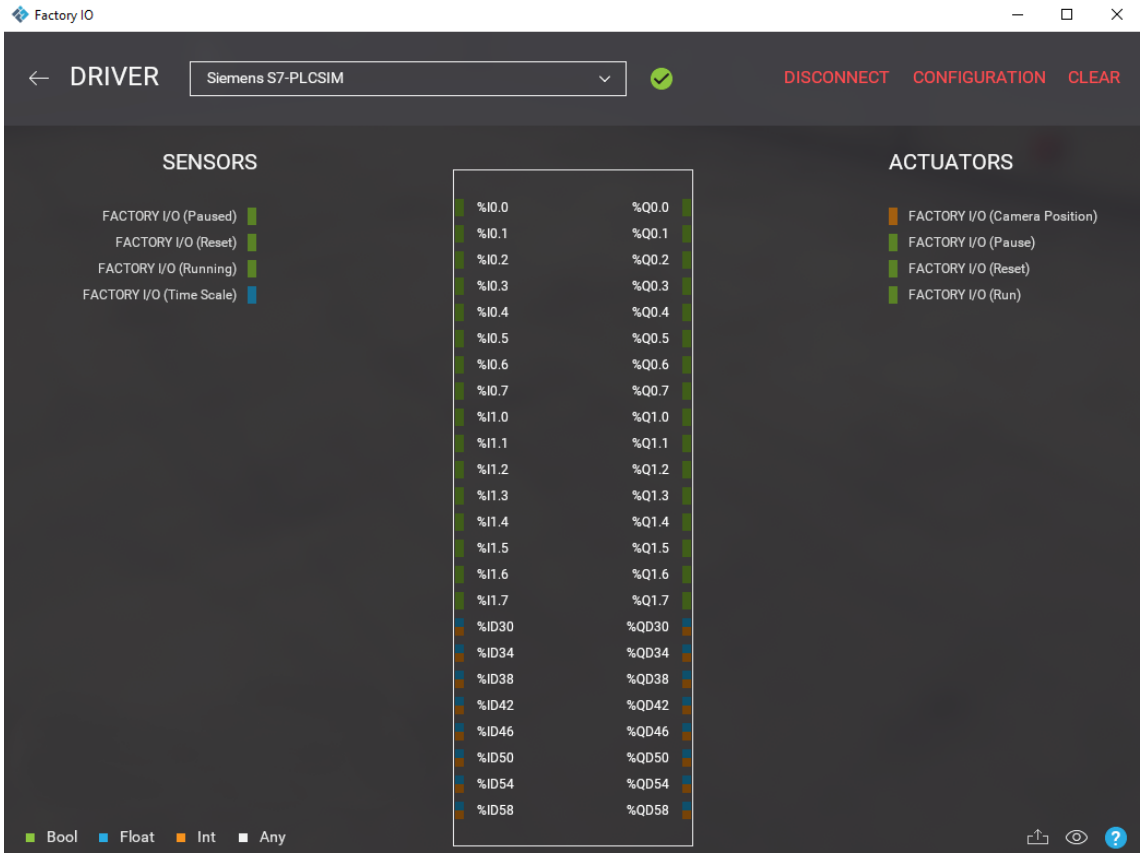
5. **Connexion de Factory I/O :** Ouvrez Factory I/O, allez dans Fichier > Pilotes, sélectionnez "Siemens S7-PLCSIM" dans la liste des pilotes. Dans le panneau de configuration, sélectionnez "S7-1500 (S7-PLCSIM Advanced)" comme modèle.[11]







6. **Établissement de la connexion :** Cliquez sur "CONNECT" dans Factory I/O. Une icône verte indique une connexion réussie.[11]



La communication entre PLCSIM Advanced et TIA Portal peut utiliser le protocole Softbus ou TCP/IP. Factory I/O se connecte à PLCSIM Advanced via TCP/IP ou un adaptateur Ethernet virtuel.[11]

PROFINET est le protocole principal pour la communication entre le S7-1500 et les E/S distribuées, ce qui est simulé par PLCSIM Advanced.[5]

2.3.2 Mapping des E/S virtuelles et réelles

Ce processus implique l'assignation des entrées/sorties virtuelles de Factory I/O à des adresses mémoire spécifiques (variables) dans le programme API au sein de TIA Portal.[26]

Dans le navigateur d'E/S de Factory I/O, les tags virtuels (par exemple, Conveyor Start, Sensor Presence) sont liés aux adresses d'E/S de l'API (par exemple, I0.0, Q0.0).[24]

Voici des exemples concrets liés au système de tri :

— Un **capteur de présence** virtuel dans Factory I/O (par exemple, Sensor Entry) serait mappé à une entrée numérique dans l'API (par exemple, I0.0). Lorsqu'une boîte passe devant le capteur dans Factory I/O, I0.0 dans l'API passe à VRAI.

— Un **moteur de convoyeur** virtuel dans Factory I/O (par exemple, Conveyeur Main) serait mappé à une sortie numérique dans l'API (par exemple, Q0.0).

Lorsque Q0.0 est défini à VRAI dans le programme API, le convoyeur virtuel démarre dans Factory I/O.

— Un **poussoir/déviateur** virtuel (par exemple, Pusher Left) serait mappé à une autre sortie numérique (par exemple, Q0.1).

— Si le tri se fait par hauteur, **un capteur de hauteur** virtuel (analogique) dans Factory I/O pourrait être mappé à une entrée analogique (par exemple, IW64) dans l'API, permettant à l'API de lire la hauteur de la boîte.[6]

Le mappage des E/S virtuelles aux variables réelles de l'API est le lien essentiel qui transforme Factory I/O d'un simple outil de visualisation 3D en un "jumeau numérique" fonctionnel du système de tri. Ce mappage permet à la logique de commande de l'API d'interagir avec un environnement virtuel réaliste, fournissant un retour d'information en temps réel sur le comportement du programme. Cela permet un test complet de la logique de commande dans divers scénarios, y compris les conditions d'erreur, sans le risque ou le coût associés au matériel physique. C'est la réalisation pratique du concept de "jumeau numérique" pour la validation du système de contrôle.[26]

Le Tableau 2.1 ci-dessous illustre des exemples de mappage des E/S virtuelles de Factory I/O aux variables de l'API dans TIA Portal. Cette table fournit des exemples concrets et pratiques de la manière dont les composants virtuels de Factory I/O sont liés aux variables de l'API dans TIA Portal. Pour un rapport destiné aux étudiants ou aux jeunes ingénieurs, ce mappage est crucial pour comprendre l'implémentation pratique de la logique de contrôle. Il fait le pont entre la compréhension théorique des API et de la simulation et la réalité concrète de la connexion du logiciel à un environnement virtuel, rendant le concept abstrait du mappage des E/S tangible et exploitable.

Objet/Tag	**Type**	**Nom de la Variable API**	**Adresse**	**Description**
Entry Sensor	Entrée Numérique	Sensor Entree	I0.0	Détecte la présence d'une boîte à l'entrée.
Main Conveyor	Sortie Numérique	Convoyeur Principal	Q0.0	Active le convoyeur principal.
Pusher A	Sortie Numérique	Poussoir A	Q0.1	Dévie la boîte vers le chemin A.
Height Sensor	Entrée Analogique	Capteur Hauteur	IW64	Mesure la hauteur de la boîte.
Exit Counter	Compteur	Compteur Sortie	DB1.DW0	Compte le nombre de boîtes triées.

TABLE 2.1 – Exemple de Mappage des E/S Virtuelles (Factory I/O) vers les Variables API (TIA Portal)

2.3.3 Utilisation de PLCSIM Advanced

SIMATIC S7-PLCSIM Advanced est le simulateur d'API logiciel de Siemens qui s'exécute sur un PC et émule le comportement d'une véritable CPU S7-1500.Il agit comme un intermédiaire, recevant le programme API compilé de TIA Portal et l'exécutant.18 Factory I/O se connecte ensuite à cette instance d'API virtuelle, échangeant des signaux d'E/S. Cela crée une simulation en boucle fermée où la logique de l'API contrôle l'usine virtuelle, et l'usine virtuelle fournit un retour d'information à l'API, permettant des tests et un débogage complets du programme de contrôle. L'utilisation de PLCSIM Advanced pour l'intégration avec Factory I/O a une implication plus large que la simple validation de projet : elle démocratise l'accès à la formation et au développement avancés en automatisation industrielle. Sans la nécessité d'un coûteux matériel S7-1500 physique, les étudiants et les ingénieurs peuvent acquérir une expérience pratique de la programmation API complexe et du comportement des systèmes en temps réel dans un environnement virtuel sûr et rentable. Cela réduit les obstacles à l'apprentissage des technologies d'automatisation de pointe, favorisant une main-d'œuvre plus qualifiée capable de répondre aux exigences de l'Industrie 4.0.[11]

2.4 Conception du Système de Tri de Caisses dans Factory I/O

2.4.1 Présentation des éléments nécessaires à la modélisation

Pour construire le système de tri virtuel dans Factory I/O, les composants suivants de sa bibliothèque seraient utilisés :[20]

- **Convoyeurs :**Convoyeurs à bande droits, convoyeurs à bande courbes ou convoyeurs à rouleaux pour transporter les boîtes.
- **Capteurs de différents types :**
  - Capteurs diffus pour la détection de présence à divers points (entrée, points de décision de tri, sortie).
  - Barrière lumineuse ou capteur de vision si le tri est basé sur la hauteur, la couleur ou des caractéristiques spécifiques.
  - Codeur incrémental pour un comptage précis des boîtes ou un retour de vitesse.
- **Actionneurs pour le tri :**
  - Poussoir ou trieur à bras pivotant pour dévier les boîtes sur différents chemins.
  - Lames d'arrêt pour retenir les boîtes aux points de décision.

— **Éléments structurels** : Passerelles, garde-corps et plates-formes pour créer un environnement d'usine réaliste.

— **Éléments de signalisation :** Colonnes lumineuses ou voyants pour indiquer l'état du système (par exemple, en fonctionnement, en défaut, nombre trié).

— **Articles :** Boîtes (de différentes tailles/couleurs si applicable pour les critères de tri).

Factory I/O propose des scènes pré-construites telles que "Tri par hauteur" ou "Station de tri" qui peuvent servir de points de départ ou d'inspiration.

2.4.2 Processus de construction de la scène 3D

La construction de la scène 3D implique l'assemblage des éléments de la palette Factory I/O dans la vue 3D. Les étapes comprennent :

1. **Placement :**Glisser-déposer les pièces de la Palette dans l'environnement 3D.[25]
2. **Orientation :**Faire pivoter les pièces (lacet, roulis, tangage) pour les aligner correctement.[25]
3. **Connexion :** Connecter physiquement les convoyeurs et s'assurer que les chemins de flux des boîtes sont corrects. L'algorithme de collision intelligent de Factory I/O aide à placer les pièces de manière valide.[25]
4. **Conception de l'agencement :**Disposer les convoyeurs, les capteurs et les actionneurs pour former la logique de tri souhaitée (par exemple, convoyeur principal, plusieurs chemins de déviation).[27]
5. **Regroupement** : Regrouper les pièces connexes pour faciliter les manipulations. [25]

2.4.3 Configuration des propriétés des objets

Après avoir placé les objets, leurs propriétés doivent être configurées pour simuler un comportement réel.Voici quelques exemples :

— Réglage de la **vitesse d'un convoyeur** (par exemple, 0,5 m/s).

— Ajustement de la **portée d'un capteur** (par exemple, distance de détection d'un capteur diffus).

— Configuration du **comportement d'un poussoir** (par exemple, vitesse d'extension, vitesse de rétraction).

— Définition du **type d'articles** émis par un émetteur (par exemple, des boîtes de tailles/couleurs aléatoires).

— Configuration de l'**affichage numérique** pour montrer les comptes triés.

2.5 Avantages de l'Utilisation de Factory I/O

L'utilisation de Factory I/O pour le projet de système de tri offre une multitude d'avantages, optimisant le processus de développement et de validation :

— **Validation du programme API avant la mise en service réelle :** Factory I/O permet un test et un débogage complets de la logique de commande de l'API dans un environnement virtuel. Cela permet d'identifier et de corriger les erreurs avant le déploiement sur le matériel physique , réduisant ainsi considérablement le risque de dommages matériels et de temps d'arrêt coûteux.[28]

— **Visualisation claire du fonctionnement du système :** L'environnement 3D interactif fournit une représentation visuelle intuitive du processus de tri, facilitant la compréhension des interactions complexes entre les capteurs, les actionneurs et la logique de commande.[19]

— **Facilitation de la détection et de la correction des erreurs de logique :** La capacité de mettre en pause, de ralentir ou d'accélérer la simulation est inestimable pour analyser le comportement du système, identifier les défauts logiques et déboguer efficacement le programme API.[28]

— **Possibilité de tester différents scénarios et optimisations sans risque pour l'équipement réel :** Les ingénieurs peuvent expérimenter divers algorithmes de tri, débits ou scénarios de défaillance (par exemple, défaillance d'un capteur, convoyeur bloqué) sans compromettre l'équipement physique ni perturber la production. [17]

— **Outil pédagogique puissant pour la compréhension du système :** Factory I/O est un excellent outil pédagogique, offrant un environnement pratique et sans risque pour les étudiants et les ingénieurs afin d'apprendre l'automatisation industrielle, la programmation API et l'intégration de systèmes.[28]

Il aide à développer des compétences pratiques telles que la détection de pannes et la compréhension des protocoles opérationnels. Alors que des outils de simulation traditionnels existent (par exemple, Matlab/Simulink), la force de Factory I/O réside dans sa capacité à fournir un "contexte industriel réel". Cela comble directement le fossé entre les connaissances théoriques et l'application pratique, un défi courant dans l'enseignement de l'ingénierie. En permettant aux étudiants de "pratiquer dans des conditions industrielles réelles" virtuellement , il favorise le développement de compétences critiques comme la détection de pannes et le respect des protocoles, qui sont difficiles à enseigner dans un cadre académique sans laboratoires physiques coûteux. Cela fait de Factory I/O un outil transformateur pour l'apprentissage basé sur les compétences en automatisation.[29]

2.6 Conclusion

Ce chapitre a souligné le rôle essentiel de la simulation dans l'automatisation industrielle moderne, en mettant en évidence ses avantages pour la validation des programmes, la détection des erreurs, la réduction des coûts et la formation des opérateurs. Factory I/O a été présenté comme un puissant simulateur interactif 3D, doté d'une vaste bibliothèque d'équipements industriels, d'une physique réaliste et d'une interface conviviale avec des modes d'édition et d'exécution distincts.

Un point crucial a été présenté, c'est le détail du processus d'intégration transparente de Factory I/O avec l'API Siemens S7-1500 via PLCSIM Advanced et TIA Portal, expliquant les principes de communication et les étapes essentielles pour configurer la connexion et mapper les E/S virtuelles aux variables de l'API.

Le chapitre a également décrit la méthodologie de conception du système de tri de caisses dans Factory I/O, depuis la sélection des composants jusqu'à la construction de la scène 3D et la configuration des propriétés des objets.

Enfin, les avantages significatifs de l'utilisation de Factory I/O pour ce projet ont été soulignés, en particulier sa capacité à valider les programmes API, à visualiser le fonctionnement du système, à faciliter la correction des erreurs, à permettre des tests de scénarios sans risque et à servir d'outil pédagogique puissant.

CHAPITRE 3

CONCEPTION ET MISE EN ŒUVRE D'UN SYSTÈME DE TRI AUTOMATISÉ SUR FACTORY I/O

3.1 Introduction

Après avoir posé les fondements théoriques concernant l'automate programmable industriel Siemens S7-1500 et son environnement de développement TIA Portal dans le premier chapitre , et exploré les capacités du logiciel de simulation Factory I/O pour la création d'usines virtuelles dans le second chapitre, ce troisième chapitre est consacré à l'application pratique de ces connaissances. L'objectif est de démontrer la mise en œuvre concrète d'un système d'automatisation, en passant de la phase conceptuelle à une solution fonctionnelle simulée.

La démarche suivie dans ce chapitre consistera à présenter, dans un premier temps, le système de tri de caisses tel qu'il a été modélisé et construit dans l'environnement 3D de Factory I/O. Par la suite, un cahier des charges détaillé sera établi pour définir les spécifications techniques et les exigences fonctionnelles du système. Finalement, la dernière partie de ce chapitre sera dédiée à la présentation et à l'explication du programme de commande, développé en langage Ladder et FBD sur TIA Portal, qui orchestre l'ensemble des opérations du système de tri.

L'objectif de cette transition n'est pas simplement une application technique, mais plutôt une démonstration systématique de la manière dont les concepts théoriques peuvent être traduits en solutions d'ingénierie pratiques. Ce chapitre met en évidence la valeur pédagogique et pratique du travail réalisé, en soulignant que la compréhension de l'architecture API, la programmation dans TIA Portal et les principes de simulation sont des bases indispensables pour résoudre les problèmes industriels du monde réel. Par conséquent, ce chapitre fournit non seulement une solution technique, mais présente également une méthodologie intégrée pour le développement de l'ingénierie, qui favorise une compréhension approfondie de l'interrelation entre la théorie et la pratique.

3.2 Description du Système Virtuel sur Factory I/O

Le système que nous avons conçu pour ce projet, illustré dans la figure ci-dessous, est une simulation complète d'une ligne de production et de logistique automatisée. Il a été entièrement conçu dans l'environnement de développement Factory I/O 3D pour représenter un processus industriel réaliste, du tri et de l'emballage automatisés des matières premières au stockage et à la distribution des produits finis.



FIGURE 3.1 – Poste de Tri de Caisses

3.2.1 Structure générale du système

Le système présenté est une ligne de production et de logistique entièrement automatisée, simulée à l'aide du logiciel Factory I/O. Son rôle est de prendre en charge des produits entrants, de les trier, de les palettiser et de les stocker de manière organisée dans un entrepôt. Le processus peut être décomposé en plusieurs étapes clés, comme illustré dans le schéma fonctionnel.

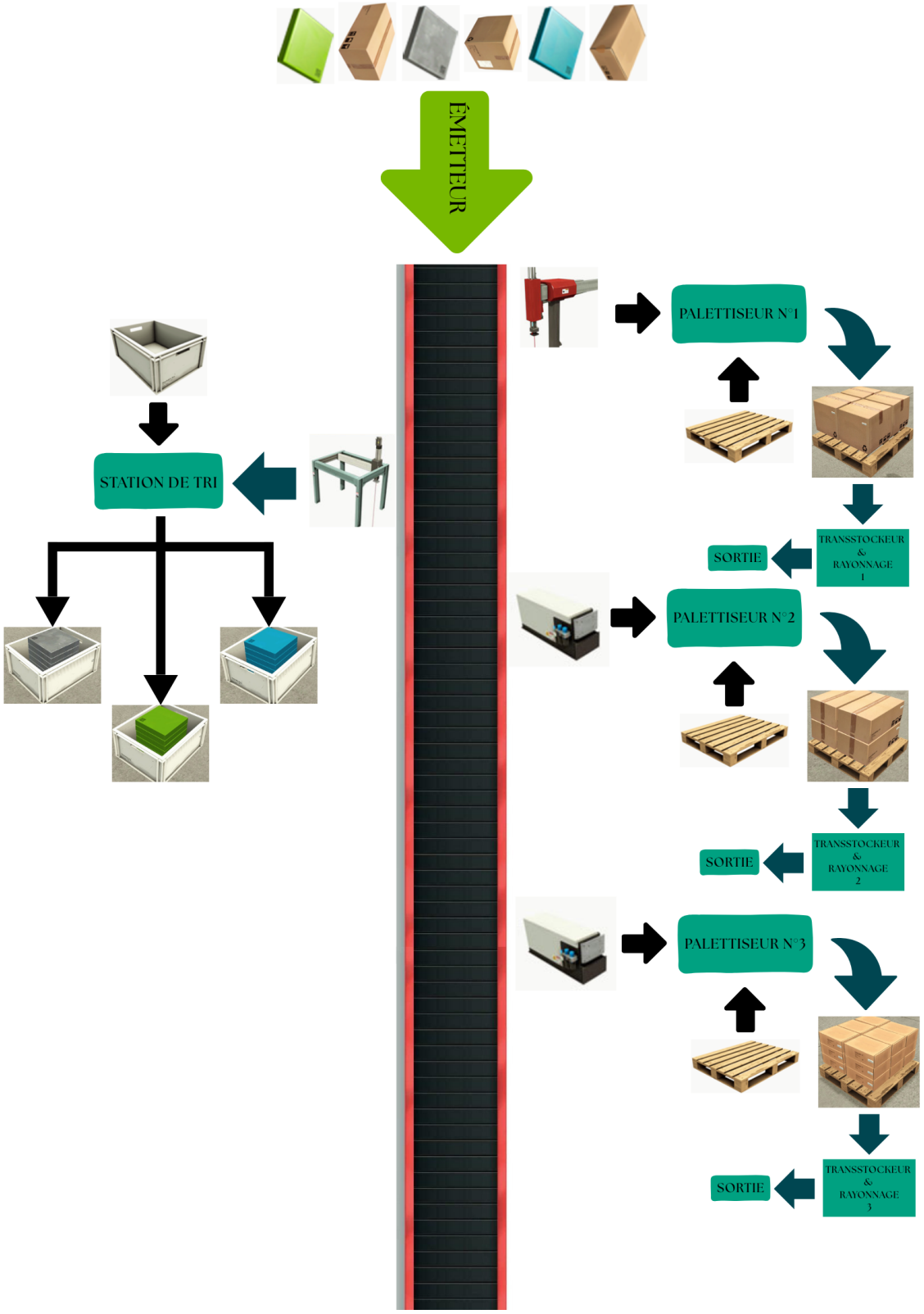


FIGURE 3.2 – Plan d'exploitation du système

1. **Arrivée des Produits (ÉMETTEUR) :**

   Le processus commence à l'émetteur, qui introduit différents types de produits sur un convoyeur central. Ces produits peuvent varier en nature, comme le montrent les différentes boîtes (cartons, articles avec des couvercles

de couleur bleue, verte ou grise).

2. **Palettiseur (N°1, N°2, N°3) :** Une partie des produits circulant sur le convoyeur est destinée à être empilée sur des palettes. Le système dispose de trois stations d'empilage distinctes :

   — Les capteurs de volume déterminent quels produits spécifiques seront placés sur les palettes.

   — Un bras robotisé (N°1) ou un Poussoir (N°2, N°3) récupère ces produits et les redirige vers une station d'empilage.

   — Chaque poste d'empilage semble dédié à un type de produit, permettant la création de palettes homogènes prêtes à être stockées.



FIGURE 3.3 – Poste Palettiseur N°1

3. **Station de Tri (SORTING STATION) :** Les produits non palettisés continuent leur parcours sur le convoyeur principal jusqu'à la station de tri. À cette étape :

   — Les produits sont déviés vers une zone de tri avec un bras robotisé d'un autre modale.

   — Des capteurs de la vision des couleurs identifient chaque article.

   — En fonction de leurs caractéristiques, les produits sont orientés et placés dans des bacs ou conteneurs correspondants.

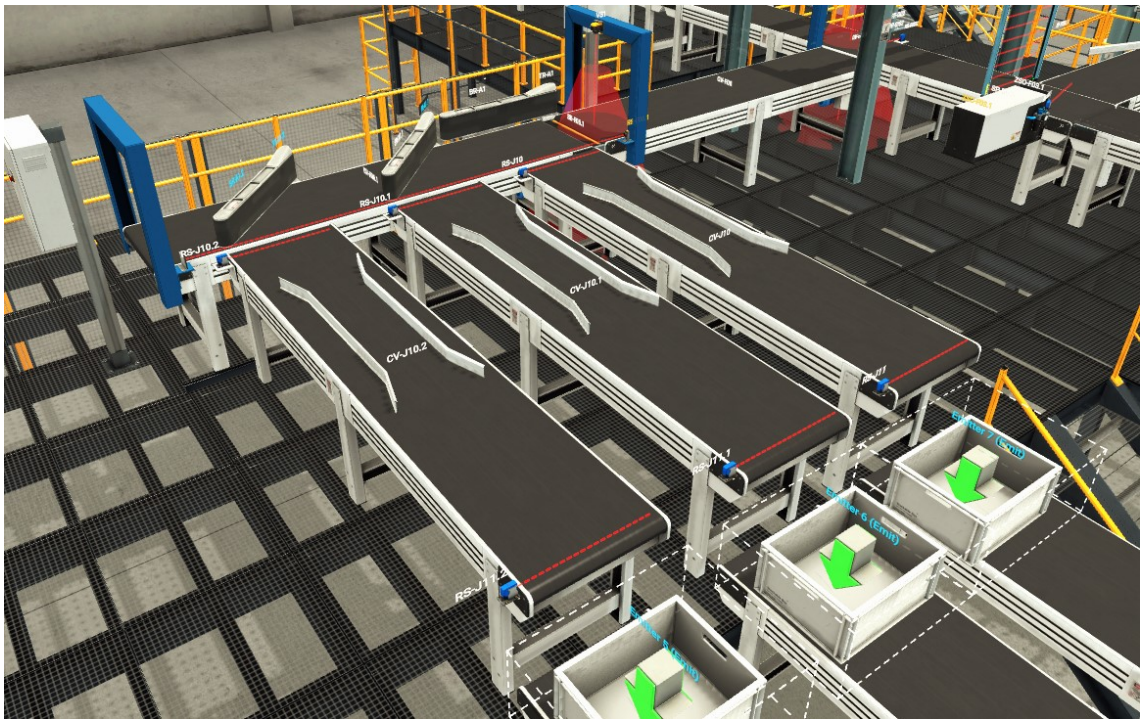


FIGURE 3.4 – Poste de Tri

4. **Stockage Automatisé (TRANS STOCKEUR ,RAYONNAGE)** Systèmes de stockage et de récupération automatisés . Ces tours de rayonnages sont desservies par un chariot élévateur automatisé qui se déplace verticalement et horizontalement pour déposer ou récupérer les produits (palettes ou cartons) aux emplacements définis dans le rayonnage.

   — **Rayonnage :** La structure physique où les produits sont stockés.

   — **Chariot de Trans stockage :** L'élément mobile qui interagit avec les rayonnages. Il reçoit des commandes du système de contrôle pour les opérations de stockage et de déstockage.

   Ils sont essentiels pour optimiser l'espace de stockage et automatiser les opérations d'inventaire.



FIGURE 3.5 – Poste Stockage

3.2.2 Cahier des charges :

Toutes les marchandises entrent dans le système et circulent le long **d'une ligne principale** constituée de plusieurs convoyeurs successifs. Chaque convoyeur décharge les articles sur le suivant, assurant une progression continue.

Station de Palettisation des Grandes Boîtes

1. **Capteur de Taille :** Au début de la ligne principale, un capteur de taille est positionné pour détecter spécifiquement les grandes boîtes.
2. **Pick & Place** : Lorsque de grandes boîtes sont détectées, un système de pick and place est activé.
3. **Déplacement et Palettisation :** Ce pick and place déplace les grandes boîtes vers une station de palettisation. À cette station, les boîtes sont empilées sur des palettes en bois, avec deux boîtes par palette.

Station de Tri des Matières Premières

1. **Pick & Place :** Au milieu de la ligne principale, un autre système de pick and place est installé.
2. **Séparation des Matières Premières :** Ce pick and place a pour rôle de séparer les matières premières en fonction de leur couleur ou type : vertes, bleues et métalliques.

3. **Station de Tri avec Capteur de Vision :** Les matières premières séparées sont dirigées vers une station de tri, qui est équipée d'un capteur de vision pour une identification et un tri précis.
4. **Unité de Tri :** Les matières premières bleues, vertes et minérales (métalliques) sont ensuite acheminées vers une unité de tri spécifique. Cette unité est actionnée par des convoyeurs et des trieurs à bras pivotant. Elles sont empilées dans des boîtes en plastique, chaque type ou couleur ayant sa propre boîte.

Unités d'Emballage (Plastination) pour Boîtes (M,S)

1. **Capteur de Taille :** Après la station de tri, un autre capteur de taille est présent sur la ligne principale pour détecter les fûts de taille moyenne.
2. **Pousseur :** Un pousseur est activé pour dévier ces fûts de taille moyenne.
3. **Système de Plastination :** Les fûts sont dirigés vers un système de plastination qui les emballe sur des palettes en bois.
4. **Unité Similaire pour Petites Boîtes :** Immédiatement après cette unité, il existe une autre unité similaire qui effectue la même tâche d'emballage sur palettes, mais cette fois-ci pour les petites boîtes.

Stockage et Distribution des Produits Finis

1. **Transport vers l'Unité de Stockage :** Les sorties des stations de palettisation (pour les grandes boîtes et les articles plastiqués) se déplacent via des convoyeurs vers l'unité de stockage dédiée à chaque type de produit fini.
2. **Distribution :** Enfin, les produits finis sont distribués à l'aide d'un chariot élévateur spécial conçu pour cette tâche.

3.3 Programmation de l'API avec TIA Portal :

3.3.1 Configuration Matériel :

Avant d'entamer toute phase de programmation dans le système automatisé, par le logiciel Factory I/O, il est impératif d'effectuer une configuration matérielle rigoureuse. Cette étape consiste à définir les différentes entrées et sorties (E/S) qui seront utilisées par l'automate programmable industriel (API). En effet, chaque capteur intégré dans le processus — comme une cellule de détection de passage, un capteur de position ou un détecteur de proximité — constitue une entrée du système.

La configuration ci-dessous présente la configuration des entrées et des sorties dans en logiciel factory io

Name	Data Type	Address	Name	Data Type	Address
ZSC-F03.1	Bool	%I0.0	CV-F03.1	Bool	%Q0.6
ZSC-F03.2	Bool	%I0.1	CV-F03.2	Bool	%Q0.7
ZSO-F03.1	Bool	%I0.2	CV-F06.1	Bool	%Q1.0
ZSO-F03.2	Bool	%I0.3	CV-FP01	Bool	%Q1.1
ZSO-P01	Bool	%I0.4	CV-FP02	Bool	%Q1.2
ZSO-P02	Bool	%I0.5	CV-FP03	Bool	%Q1.3
ZSO-P03	Bool	%I0.6	CV-FP01.1	Bool	%Q1.4

Auto	Bool	%I0.7	CV-FP02.1	Bool	%Q1.5
RS	Bool	%I1.0	CV-FP03.1	Bool	%Q1.6
D0-J20	Bool	%I1.1	CV-FSTR	Bool	%Q1.7
rs(1)	Bool	%I1.2	CV-J01	Bool	%Q2.0
D90-J20	Bool	%I1.3	CV-J02	Bool	%Q2.1
EI-P01	Bool	%I1.4	CV-J03	Bool	%Q2.2
EI-P02	Bool	%I1.5	CV-J04	Bool	%Q2.3
EI-P03	Bool	%I1.6	CV-J10	Bool	%Q2.4
M	Bool	%I1.7	CV-J11	Bool	%Q2.5
ES-J20	Bool	%I2.0	CV-J04.1	Bool	%Q2.6
ES-P01	Bool	%I2.1	CV-J04.2	Bool	%Q2.7
ES-P02	Bool	%I2.2	CV-J04.3	Bool	%Q3.0
ES-P03	Bool	%I2.3	CV-J10.1	Bool	%Q3.1
ET-P01	Bool	%I2.4	CV-J10.2	Bool	%Q3.2
ET-P02	Bool	%I2.6	CV-J11.1	Bool	%Q3.3
HD-01	Bool	%I2.7	CV-J11.2	Bool	%Q3.4
HD-02	Bool	%I3.0	CV-PB01	Bool	%Q3.5
HD-03	Bool	%I3.1	CV-PB02	Bool	%Q3.6
HD-04	Bool	%I3.2	CV-PB03	Bool	%Q3.7
HD-05	Bool	%I3.3	ED-P01	Bool	%Q4.0
HD-06	Bool	%I3.4	ED-P02	Bool	%Q4.1
HD-07	Bool	%I3.5	ED-P03	Bool	%Q4.2
SC.L.LIMIT	Bool	%I3.6	EF-P01	Bool	%Q4.3
SC.R.LIMIT	Bool	%I3.7	EF-P02	Bool	%Q4.4
SC-M	Bool	%I4.0	EF-P03	Bool	%Q4.5
Manual	Bool	%I4.1	EML-P01	Bool	%Q4.6
PP-GI01	Bool	%I4.2	EML-P02	Bool	%Q4.7
PP-GI02	Bool	%I4.3	EML-P03	Bool	%Q5.0
Reset	Bool	%I4.4	EU-P01	Bool	%Q5.1
RS-F06.1	Bool	%I4.5	EU-P02	Bool	%Q5.2
RS-FP01	Bool	%I4.6	EU-P03	Bool	%Q5.3
RS-FP02	Bool	%I4.7	CV-J04.5	Bool	%Q5.4
RS-FP03	Bool	%I5.0	FD-J20	Bool	%Q5.5
RS-FP01.1	Bool	%I5.1	PP-G01	Bool	%Q5.6
RS-FP02.1	Bool	%I5.2	PP-G02	Bool	%Q5.7
RS-FP03.1	Bool	%I5.3	L	Bool	%Q6.0
RS-J01	Bool	%I5.4	Reset light	Bool	%Q6.1
RS-J02	Bool	%I5.5	R	Bool	%Q6.2
RS-J03	Bool	%I5.6	RF-J20	Bool	%Q6.3
RS-J10	Bool	%I5.7	SB-F06.1	Bool	%Q6.4
RS-J11	Bool	%I6.0	SC-P01	Bool	%Q6.5
RS-J04.3	Bool	%I6.1	SC-P02	Bool	%Q6.6
RS-J10.1	Bool	%I6.2	SC-P03	Bool	%Q6.7
RS-J10.2	Bool	%I6.3	SG-P01	Bool	%Q7.0
RS-J11.1	Bool	%I6.4	SG-P02	Bool	%Q7.1
RS-J11.2	Bool	%I6.5	SG-P03	Bool	%Q7.2
SGL-P01	Bool	%I6.6	SP-F03.1	Bool	%Q7.3
SGL-P02	Bool	%I6.7	SP-F03.2	Bool	%Q7.4

SGL-P03	Bool	%I7.0	SP-P01	Bool	%Q7.5
SPL-P01	Bool	%I7.1	SP-P02	Bool	%Q7.6
SPL-P02	Bool	%I7.2	SP-P03	Bool	%Q7.7
SPL-P03	Bool	%I7.3	Start light	Bool	%Q8.0
SR-SGTR	Bool	%I7.4	SC-L	Bool	%Q8.1
Start	Bool	%I7.5	SC-R	Bool	%Q8.2
Stop	Bool	%I7.6	Stop light	Bool	%Q8.3
RS-PP01	Bool	%I7.7	STR-P01	Bool	%Q8.4
PP-RI01	Bool	%I8.0	STR-P02	Bool	%Q8.5
RS-F03.1	Bool	%I8.1	STR-P03	Bool	%Q8.6
RS-F03.2	Bool	%I8.2	TR-A1	Bool	%Q8.7
RS-PP02	Bool	%I8.3	TR-A1.1	Bool	%Q9.0
RS-J04	Bool	%I8.4	TR-A1.2	Bool	%Q9.1
RS-J05	Bool	%I8.5	Unload Conveyor	Bool	%Q10.5
m(1)	Bool	%I8.6	WL-P01	Bool	%Q9.3
RS-J06	Bool	%I8.7	WL-P02	Bool	%Q9.4
M2	Bool	%I9.0	WL-P03	Bool	%Q9.5
RS-J08	Bool	%I9.1	PP-RL01	Bool	%Q9.6
RS-FP05	Bool	%I9.2	PP-RR01	Bool	%Q9.7
RS-FP06	Bool	%I9.3	MOVING Z	Bool	%Q10.0
RS-P07	Bool	%I9.4	CV-J04.4	Bool	%Q10.1
RS-FP03.5	Bool	%I9.5	Lift	Bool	%Q10.2
Moving-Z	Bool	%I9.6	Left	Bool	%Q10.3
Moving-X	Bool	%I9.7	Right	Bool	%Q10.4
Left Limit	Bool	%I10.0	JUMP	Bool	%Q10.6
Middle Limit	Bool	%I10.1	LC-1	Bool	%Q10.7
Right Limit	Bool	%I10.2	L2	Bool	%Q11.0
X	Bool	%I10.3	R2	Bool	%Q11.1
Z(1)	Bool	%I10.4	un-con	Bool	%Q11.2
load	Bool	%I10.5	JUMP2	Bool	%Q11.3
Unload	Bool	%I10.6	I(1)	Bool	%Q11.4
L-L	Bool	%I10.7	r(1)	Bool	%Q11.5
R-L	Bool	%I11.0	jump(1)	Bool	%Q11.6
Z-M	Bool	%I11.1	LC-2	Bool	%Q11.7
X-M	Bool	%I11.2	l2(1)	Bool	%Q12.0
L2-L	Bool	%I11.3	r2(1)	Bool	%Q12.1
R2-L	Bool	%I11.4	jump 2	Bool	%Q12.2
Z2-M	Bool	%I11.5	UN-CON(1)	Bool	%Q12.3
X2-M	Bool	%I11.6	PP-XP01	Word	%IW18
I-L(1)	Bool	%I11.7	PP-XP02	Word	%IW20
r-L(1)	Bool	%I12.0	PP-YP02	Word	%IW22
z-m(1)	Bool	%I12.1	PP-ZP02	Word	%IW24
X-m(1)	Bool	%I12.2	PP-ZP01	Word	%IW26
m2(1)	Bool	%I12.3	SI-F03	Word	%IW28
l2-L(1)	Bool	%I12.4	SI-F03.1	Word	%IW30
r2-L(1)	Bool	%I12.5	VS1	Int	%IW34
z2-m(1)	Bool	%I12.6	VS02	Int	%IW36
x2-m(1)	Bool	%I12.7	PP-X01	Word	%QW16

load1	Bool	%I13.0	PP-X02	Word	%QW18
unload1	Bool	%I13.1	PP-Y02	Word	%QW20
load2	Bool	%I13.2	PP-Z01	Word	%QW22
unload2	Bool	%I13.3	PP-Z02	Word	%QW24
BR-A1	Bool	%Q0.0	SC-Target Position	Int	%QW26
BR-A1.1	Bool	%Q0.1	Target Position	Int	%QW28
BR-A1.2	Bool	%Q0.2	tp	Int	%QW30
CV-F02	Bool	%Q0.3	TP(1)	Int	%QW32
CV-F03	Bool	%Q0.4	tp 2	Int	%QW34
CV-F06	Bool	%Q0.5	TP 2(1)	Int	%QW36

TABLE 3.1: Table des variables API

3.3.2 La réalisation du programme :

La méthode que nous avons suivie pour programmer ce système est une méthode hybride. Cette approche combine la visualisation structurée de FBD avec la simplicité logique de Ladder, permettant une programmation d'API puissante et maintenable.

En FBD, le programme est organisé en blocs fonctionnels qui représentent des tâches ou des modules spécifiques (ex : commande des convoyeur, gestion de stocke, régulation PID,système de sécurité). Ces blocs sont connectés entre eux pour montrer le flux de données ou le séquencement. Chaque bloc FBD peut ensuite encapsuler sa propre logique interne, qui peut être écrite dans un autre langage, comme le Ladder Diagramme.

main[OB],(FBD)

— La structure du programme est présentée sous la forme d'un arbre, où chaque nœud représente un bloc de programme composé d'un élément. Par exemple, le programme précédent est nécessaire pour celui qui suit, et celui qui suit ne fonctionne pas s'il y a une erreur dans celui qui le précède. Cela facilite également le processus de maintenance de chaque élément, voire la modification de certaines conditions de fonctionnement d'un élément lorsque le capteur est endommagé.

Totally Integrated Automation Portal

Main [OB1]

Main Properties							
General							
Name	Main	**Number**	1	**Type**	OB	**Language**	LAD
Numbering	Automatic						
Information							
Title	"Main Program Sweep (Cycle)"	**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value
Input		
Initial_Call	Bool	
Remanence	Bool	
Temp		
Constant		

Network 1:



Network 2:

FIGURE 3.6 – main[OB],(FBD)

Totally Integrated Automation Portal

Network 2: (1.1 / 3.1)

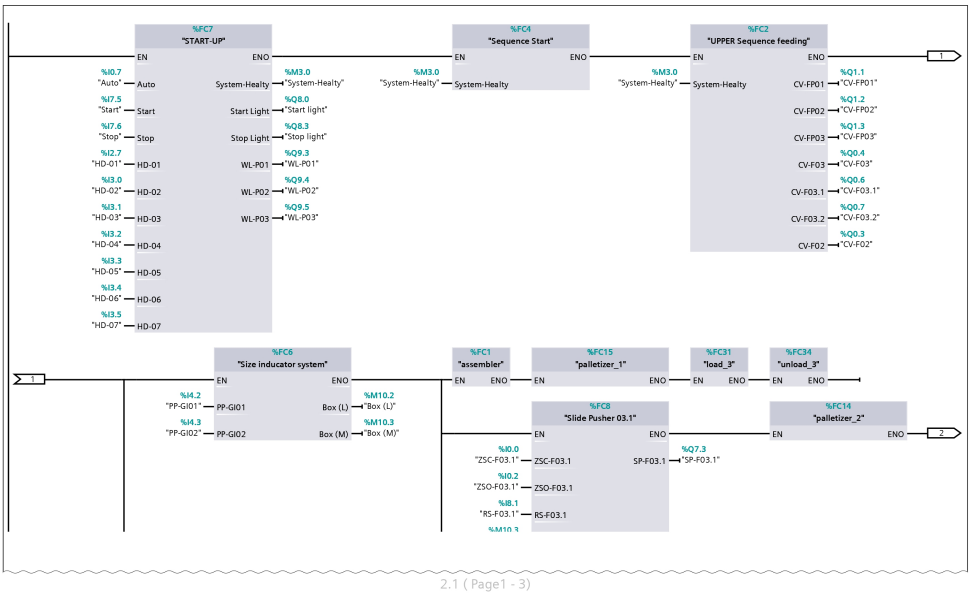


FIGURE 3.7 – Suite main[OB],(FBD)

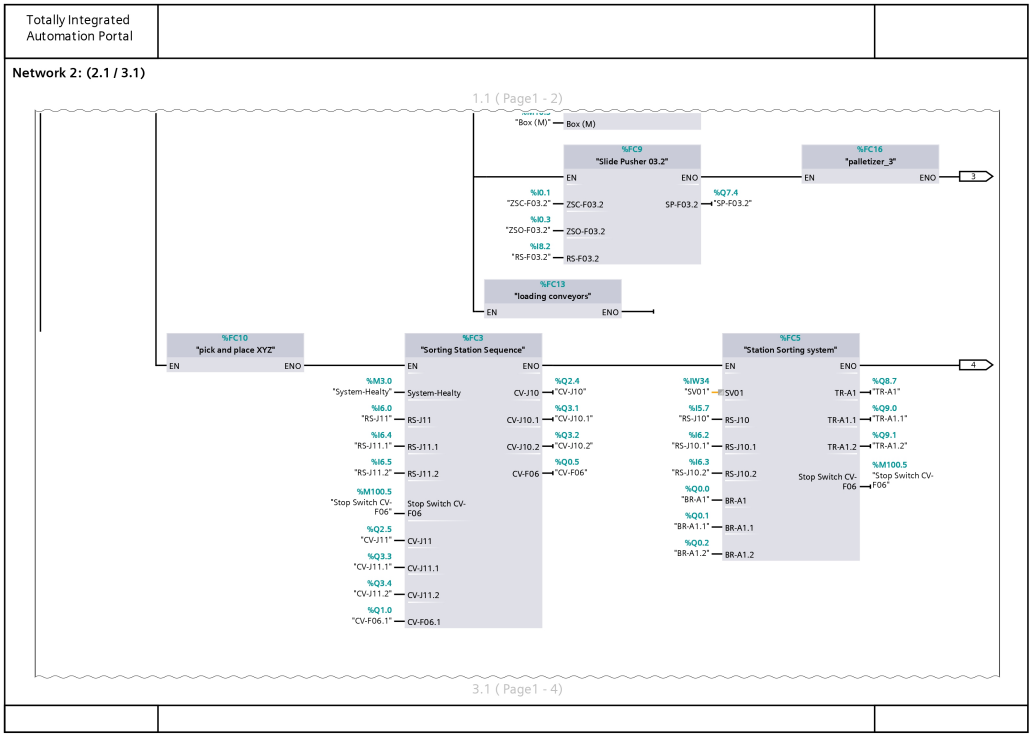


FIGURE 3.8 – Suite main[OB],(FBD)

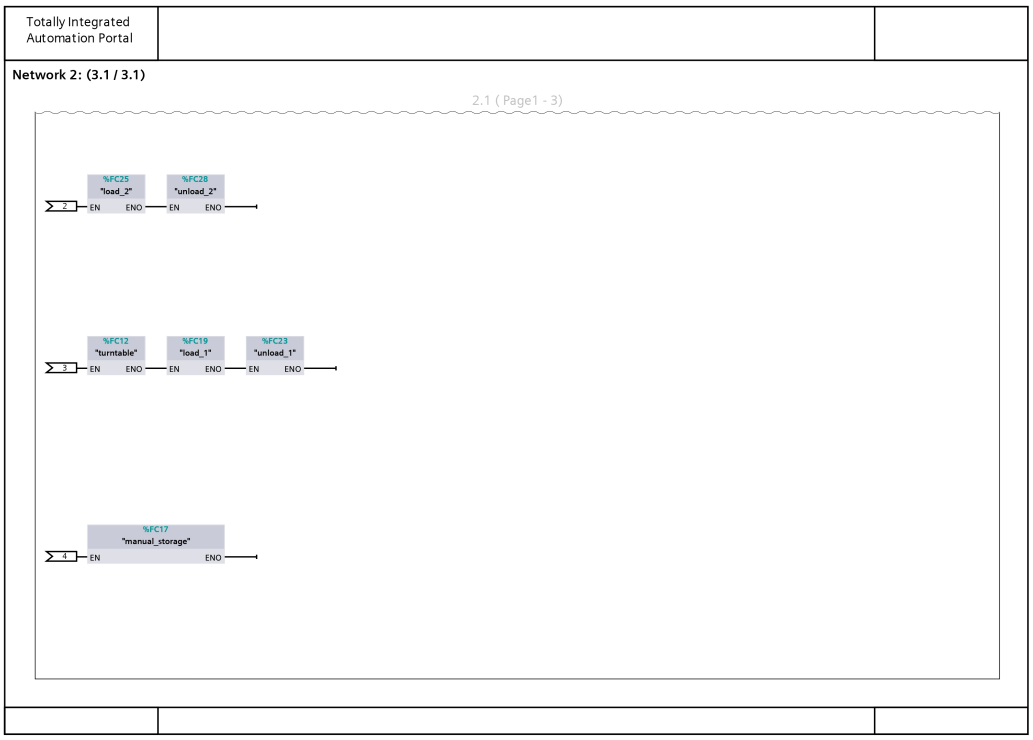


FIGURE 3.9 – Suite main[OB],(FBD)

PP_01[FC],(Ladder)

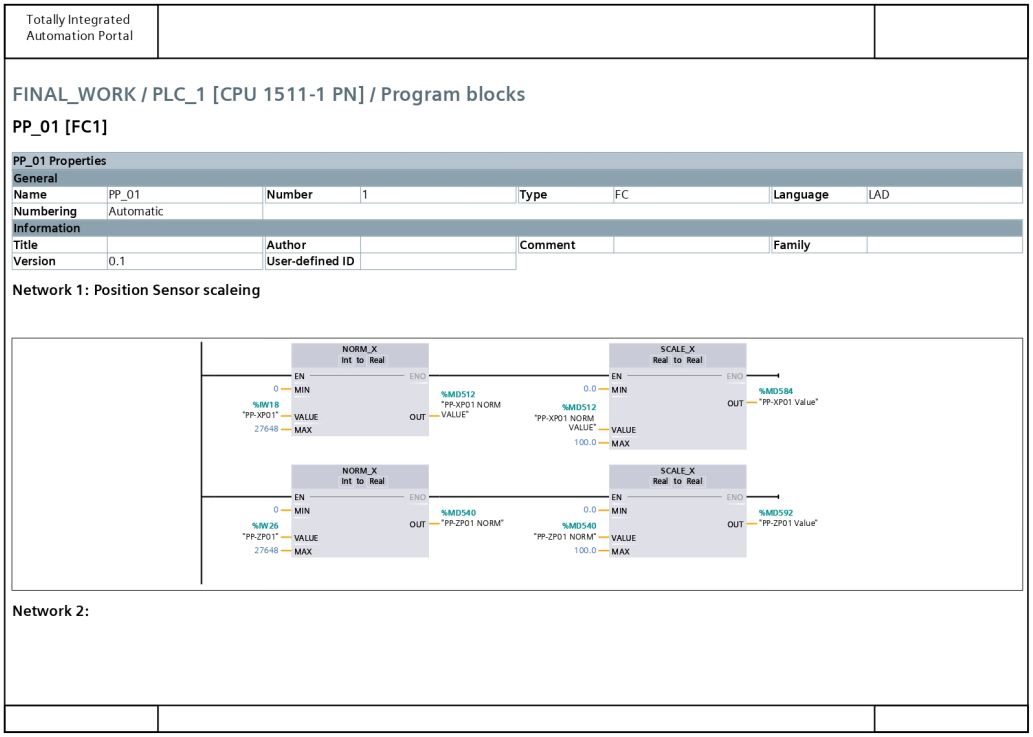


FIGURE 3.10 – PP_01[FC],(Ladder)

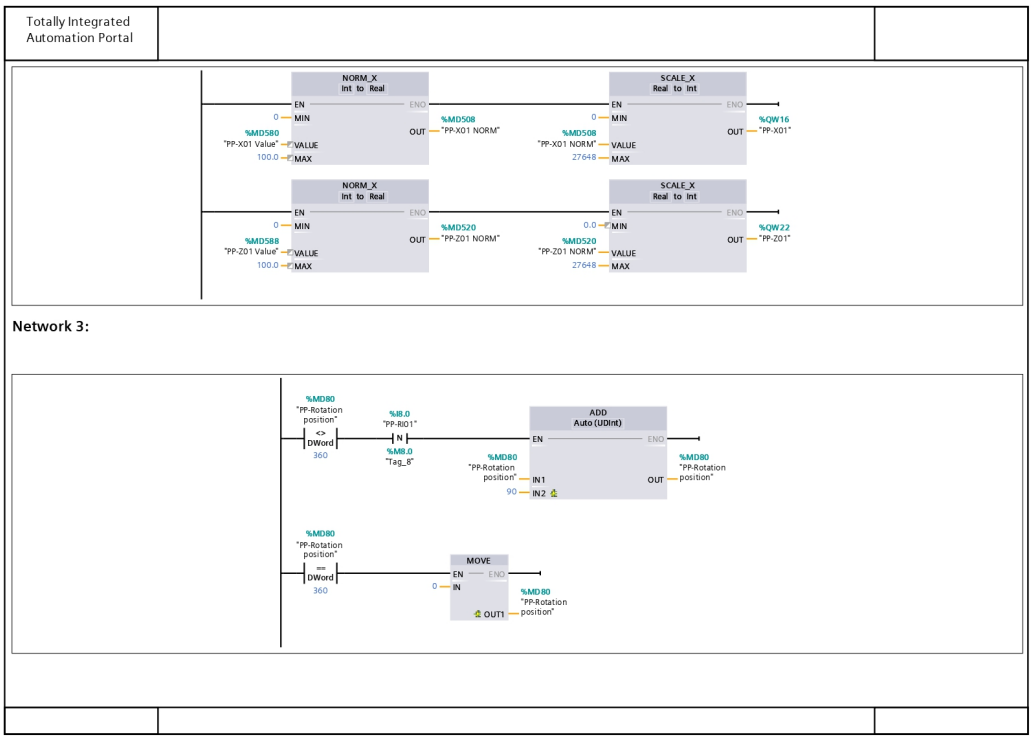


FIGURE 3.11 – Suite PP_01[FC],(Ladder)

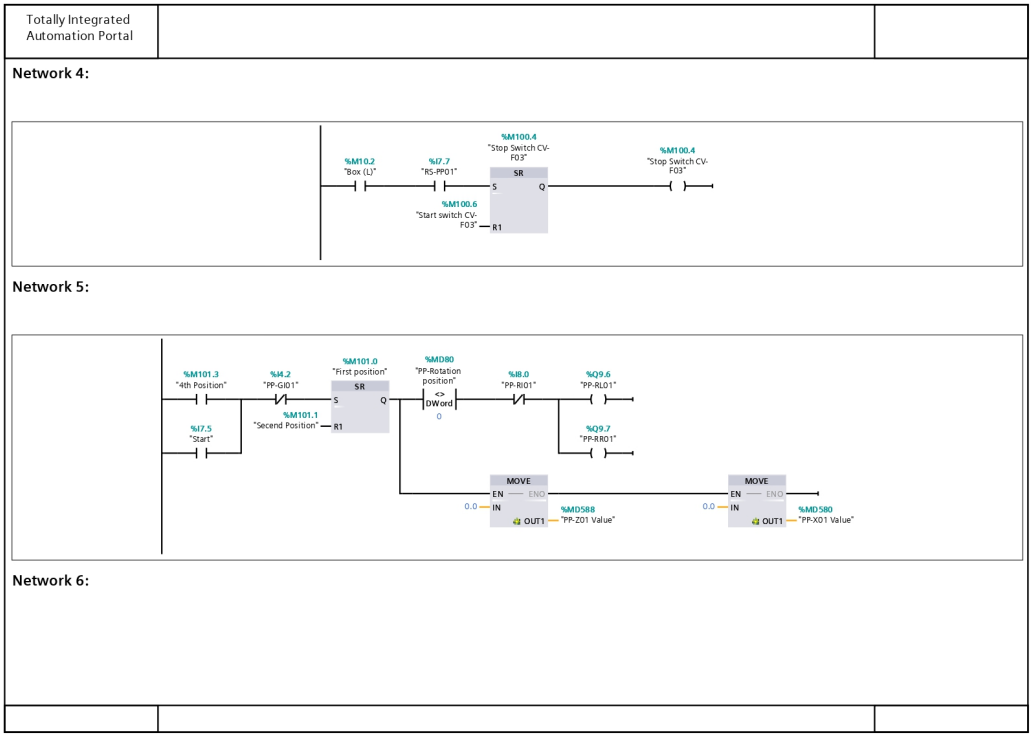


FIGURE 3.12 – Suite PP_01[FC],(Ladder)

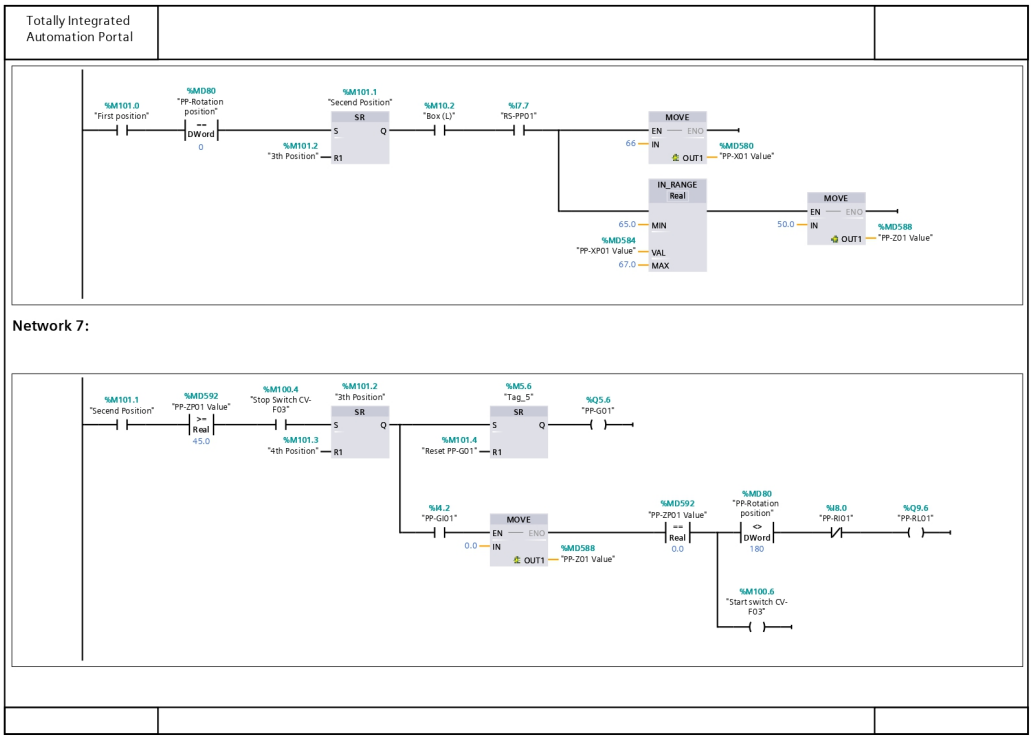


FIGURE 3.13 – Suite PP_01[FC],(Ladder)

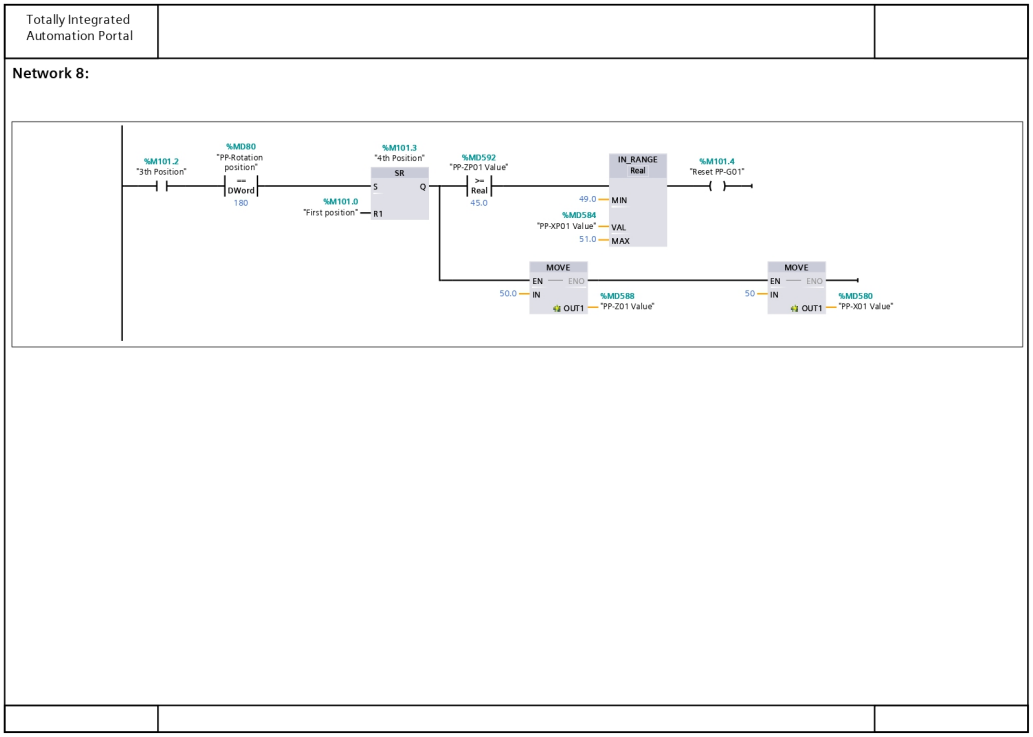


FIGURE 3.14 – Suite PP_01[FC],(Ladder)

— **NOTE : Le reste du programme est dans L'annexe.**

3.4 Conclusion

Ce troisième chapitre a concrétisé les bases théoriques établies précédemment en détaillant la conception et la mise en œuvre pratique d'un système de tri automatisé sur Factory I/O.

Nous avons d'abord présenté la structure générale de notre système virtuel, illustrant une ligne de production et de logistique entièrement automatisée, capable de trier, palettiser et stocker diverses marchandises. Le cahier des charges détaillé a ensuite permis de définir précisément les exigences fonctionnelles et les spécifications techniques de chaque station, de l'arrivée des produits au stockage final. Enfin, la section sur la programmation de l'API avec TIA Portal a mis en lumière notre approche hybride, combinant les langages Ladder et FBD pour orchestrer les opérations complexes du système.

Ce chapitre démontre non seulement la faisabilité technique de notre solution, mais il souligne également l'importance cruciale de l'intégration entre la modélisation virtuelle et la programmation API. La mise en œuvre de ce système simulé a permis de valider les choix de conception, de déceler les éventuelles erreurs logiques et d'optimiser le comportement de l'ensemble du processus dans un environnement sans risque. C'est une illustration éloquente de la manière dont les outils de simulation, couplés à une programmation rigoureuse, peuvent transformer la phase de développement en

ingénierie d'automatisation, garantissant ainsi une solution robuste et performante pour des applications industrielles réelles.

CONCLUSION GÉNÉRALE

Ce mémoire a exploré en profondeur l'automatisation d'un système virtuel de tri de caisses par automate programmable industriel, en s'appuyant sur l'intégration synergique de technologies de pointe : l'API Siemens S7-1500, l'environnement TIA Portal et le simulateur 3D Factory I/O. Nous avons démontré comment ces outils peuvent être combinés pour concevoir, programmer et valider un système d'automatisation complexe de manière efficace et sans risque.

Le premier chapitre a posé les fondements théoriques, en détaillant l'importance stratégique de l'automatisation industrielle et en présentant l'architecture, les fonctionnalités avancées et les avantages du Siemens S7-1500 et de TIA Portal.

Le deuxième chapitre a mis en lumière le rôle indispensable de la simulation, en introduisant Factory I/O comme un outil clé pour la création de "jumeaux numériques", et en expliquant son intégration transparente avec l'API Siemens S7-1500 via PLCSIM Advanced, notamment par le mappage des E/S virtuelles.

Le troisième chapitre a concrétisé ces concepts en décrivant la conception et la mise en œuvre d'un système de tri automatisé réel dans Factory I/O, accompagné de sa programmation API détaillée en langages hybrides.

Les résultats de ce travail mettent en évidence plusieurs conclusions majeures. Premièrement, l'utilisation combinée de Factory I/O et de l'écosystème Siemens (S7-1500 et TIA Portal) permet une validation rigoureuse des programmes API avant toute mise en service physique, réduisant significativement les coûts, les risques et les délais de développement. Deuxièmement, la simulation 3D offre une visualisation intuitive et interactive du comportement du système, facilitant la détection et la correction précoce des erreurs logiques. Troisièmement, cette approche constitue un outil pédagogique puissant, offrant un environnement d'apprentissage pratique et sûr pour les ingénieurs et techniciens.

En perspective, les systèmes d'automatisation continueront d'évoluer, notamment avec l'intégration croissante de l'intelligence artificielle et de l'apprentissage automatique pour des prises de décision plus complexes, comme le tri adaptatif ou la maintenance prédictive. L'optimisation énergétique, la modularité accrue et

la cybersécurité seront également des axes majeurs de développement. Ce projet ouvre la voie à des applications industrielles plus intelligentes, plus flexibles et plus résilientes, et souligne la nécessité de maîtriser ces outils d'ingénierie intégrés pour répondre aux exigences de l'Industrie 4.0 et au-delà.

BIBLIOGRAPHIE

[1] What is industrial automation and why is it important ? | ibt. https://ibtinc.com/what-industrial-automation-why-important/.

[2] Top 10 benefits of industrial automation you need to know. https://milestone.ac.in/blog-mit/10-benefits-of-industrial-automation/.

[3] What is siemens s7-1500 plc ? | features, applications, overview - indmall. https://www.indmall.in/faq/what-is-siemens-s7-1500-plc/.

[4] Simatic s7-1500 cpu 1513-1 pn - id : 109972214 - industry support siemens. https://support.industry.siemens.com/cs/mdm/109972214?c=172564620171&lc=en-cr.

[5] S7-1500 - industry mall - siemens usa. https://mall.industry.siemens.com/mall/en/us/Catalog/Products/10204162.

[6] Definition of programmable logic controller or "plc" ? - automation ready panels. https://www.automationreadypanels.com/plc-hmi/what-is-the-definition-of-a-programmable-logic-controller-or-plc/.

[7] What is a plc ? definition and details - paessler. https://www.paessler.com/it-explained/plc.

[8] Plc architecture. https://www.plctable.com/plc-architecture/.

[9] What is scan cycle in plc ? | definition, phases, and tips - indmall. https://www.indmall.in/faq/what-is-scan-cycle-in-plc/.

[10] Simatic s7-1500 - siemens us. https://www.siemens.com/us/en/products/automation/systems/industrial/plc/simatic-s7-1500.html.

[11] Setting up s7-plcsim advanced v3.0. https://docs.factoryio.com/tutorials/siemens/setting-up-s7-plcsim-advanced-V3/.

[12] Totally integrated automation portal - siemens us. https://www.siemens.com/us/en/products/automation/industry-software/automation-software/tia-portal.html.

[13] Tia portal - user interface view. https://sklep-plcspace.pl/en/blog/tia-portal-view-of-the-user-interface/.

[14] Simulation of devices - tia portal. https://docs.tia.siemens.cloud/r/en-us/v20/using-online-and-diagnostics-functions/simulating-devices-with-s7-plcsim/simulation-of-devices.

[15] What language is used in s7-1500 ? | tia portal basics. https://www.indmallautomation.com/faq/what-language-is-used-in-s7-1500/.

[16] Advanced micro controls inc : : What is a plc ? - amci. https://www.amci.com/industrial-automation-resources/plc-automation-tutorials/what-plc/.

[17] The future of factory automation : How simulation drives efficiency and innovation. https://www.autodesk.com/blogs/design-and-manufacturing/2025/05/13/the-future-of-factory-automation-how-simulation-drives-efficiency-and-innovati

[18] Industry insights : How simulation software powers real-world efficiency. https://www.automate.org/industry-insights/how-simulation-software-powers-real-world-efficiency.

[19] Download factory i/o 3d plc simulator. https://www.teckhme.com/2023/02/download-factory-io-3d-plc-simulator.html.

[20] Parts - factory i/o. https://docs.factoryio.com/manual/parts/.

[21] Factory i/o 2.3.0 now available. https://factoryio.com/blog/2018/06/15/factory-i-o-2-3-0-now-available/.

[22] Features - factory i/o. https://factoryio.com/features/.

[23] User interface - factory i/o. https://docs.factoryio.com/manual/user-interface/.

[24] Using the io browser - web store login. https://store.sim3d.com/demo3d_2023/introduction_to_the_tag_browser.

[25] Edit and run - factory i/o. https://docs.factoryio.com/manual/edit-and-run/.

[26] Approach to sorting line control using plc and factory i/o. https://www.ijitee.org/wp-content/uploads/papers/v13i10/I994113090824.pdf.

[27] Design of a new sorting colors system based on plc, tia portal, and factory i/o programs. https://www.researchgate.net/publication/378862876_Design_of_a_new_sorting_colors_system_based_on_PLC_TIA_portal_and_factory_IO_programs.

[28] The ultimate guide to plc training : Skills, certifications, and future trends in industrial automation. https://www.solisplc.com/tutorials/plc-training.

[29] Teaching automation with factory i/o under a competency-based curriculum. https://www.researchgate.net/publication/364643689_Teaching_automation_with_Factory_IO_under_a_competency-based_curriculum.

ANNEXE

Table of contents

Main [OB1]	3 - 1
actuator load2 [FC24]	4 - 1
actuatorunload1 [FC22]	5 - 1
assembler [FC1]	6 - 1
load_1 [FC19]	7 - 1
load_2 [FC25]	8 - 1
load_3 [FC31]	9 - 1
load1 sensors [FC18]	10 - 1
load1ACTUATOR [FC20]	11 - 1
load3actuators [FC30]	12 - 1
loading conveyors [FC13]	13 - 1
manual_storage [FC17]	14 - 1
MHJ-PLC-Lab-Function-S71500 [FC9000]	15 - 1
palletizer_1 [FC15]	16 - 1
palletizer_2 [FC14]	17 - 1
palletizer_3 [FC16]	18 - 1
pick and place XYZ [FC10]	19 - 1
sensors load2 [FC11]	20 - 1
sensorsload3 [FC29]	21 - 1
Sequence Start [FC4]	22 - 1
Size inducator system [FC6]	23 - 1
Slide Pusher 03.1 [FC8]	24 - 1
Slide Pusher 03.2 [FC9]	25 - 1
Sorting Station Sequence [FC3]	26 - 1
START-UP [FC7]	27 - 1
Station Sorting system [FC5]	28 - 1
turntable [FC12]	29 - 1
unload_1 [FC23]	30 - 1

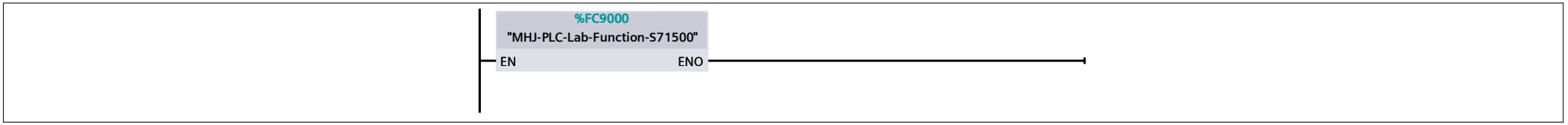
unload_2 [FC28]	31 - 1
unload_3 [FC34]	32 - 1
unloadactuators3 [FC33]	33 - 1
unloading actuator2 [FC27]	34 - 1
unloadsensors1 [FC21]	35 - 1
unloadsensors2 [FC26]	36 - 1
unloadsensors3 [FC32]	37 - 1
UPPER Sequence feeding [FC2]	38 - 1

Main [OB1]

Main Properties							
General							
Name	Main	**Number**	1	**Type**	OB	**Language**	LAD
Numbering	Automatic						
Information							
Title	"Main Program Sweep (Cycle)"	**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

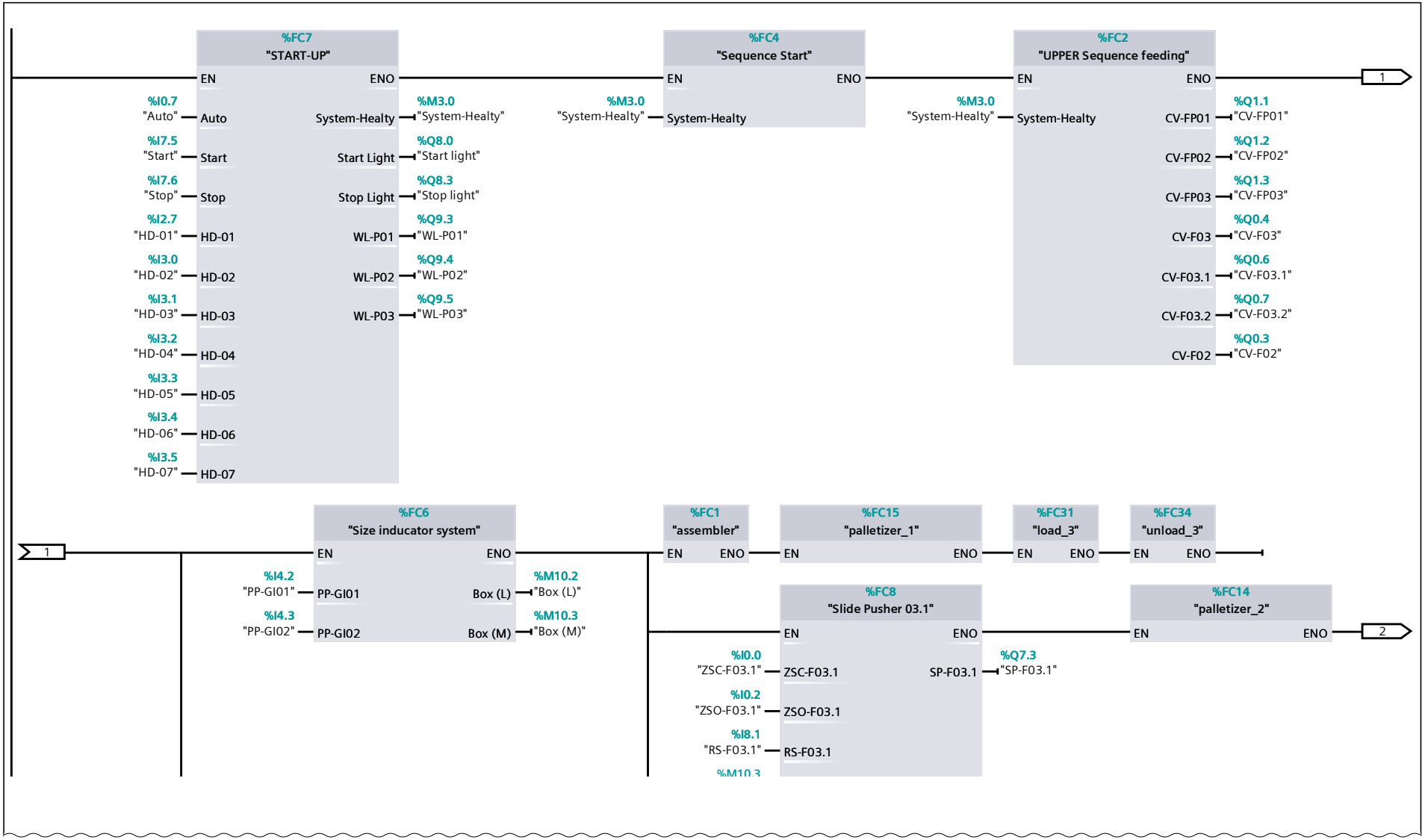
Name	Data type	Default value	Comment
Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
Temp			
Constant			

Network 1:



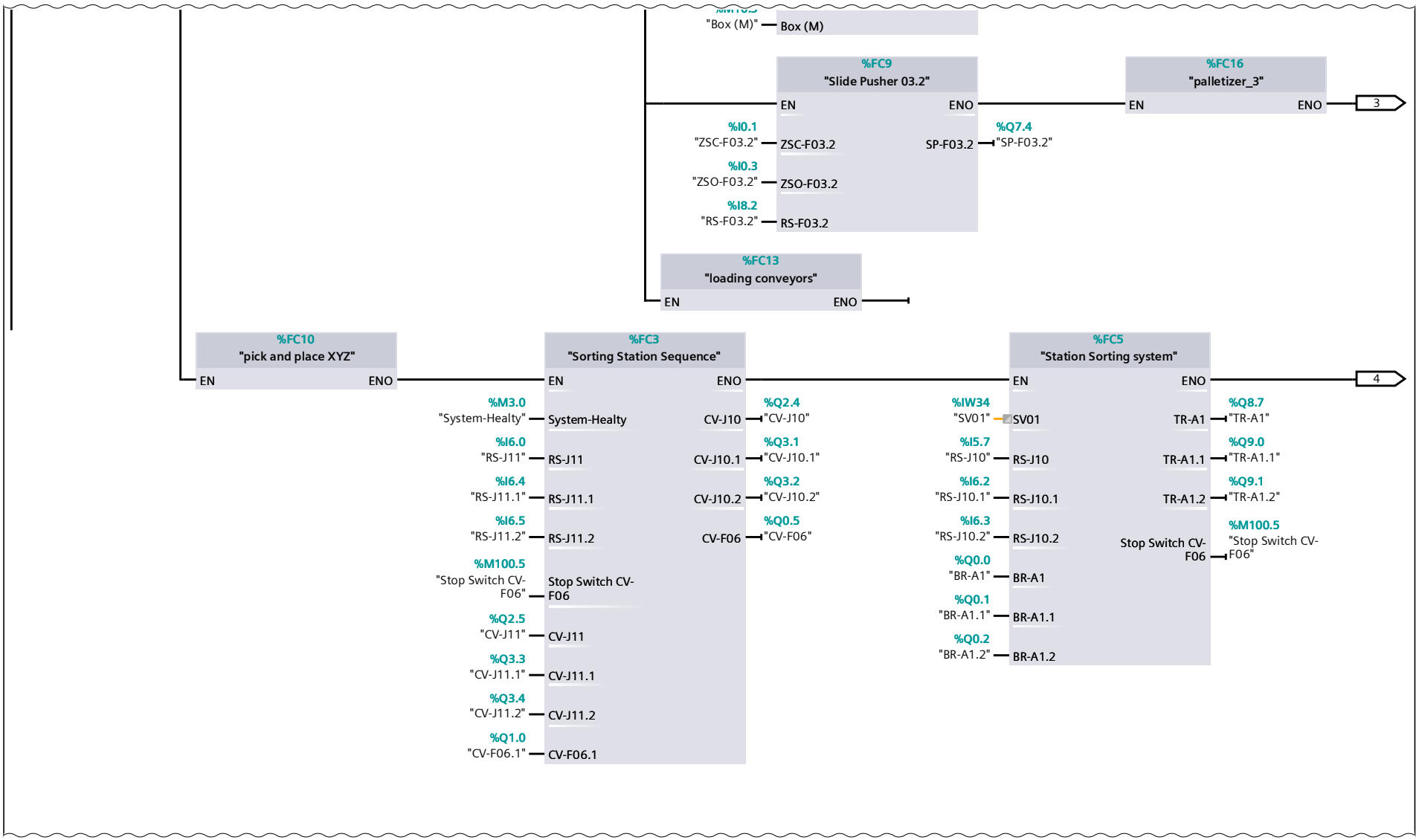
Network 2:

Network 2: (1.1 / 3.1)



Network 2: (2.1 / 3.1)

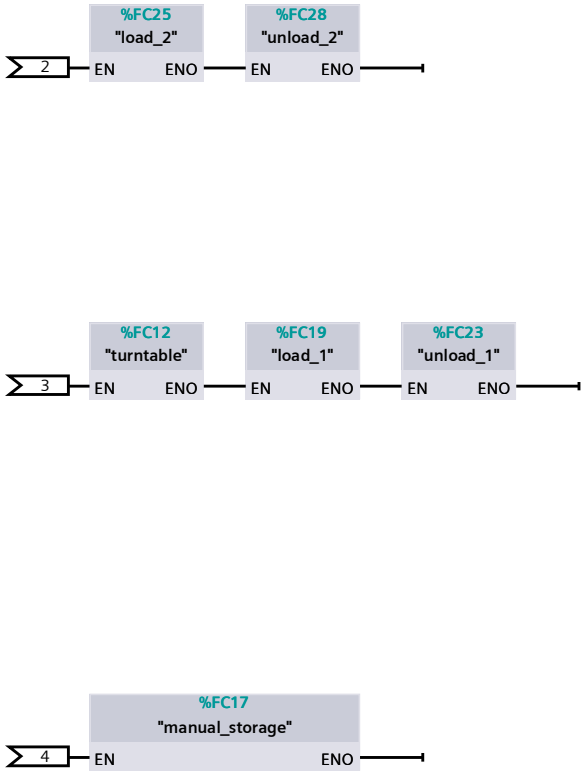
1.1 (Page3 - 2)



3.1 (Page3 - 4)

Network 2: (3.1 / 3.1)

2.1 (Page3 - 3)

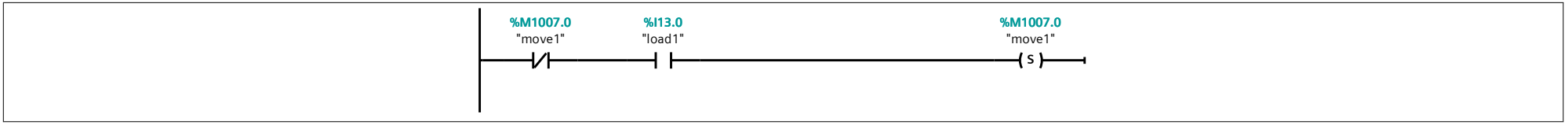


actuator load2 [FC24]

actuator load2 Properties							
General							
Name	actuator load2	**Number**	24	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
▼ Temp			
num	Int		
Constant			
▼ Return			
actuator load2	Void		

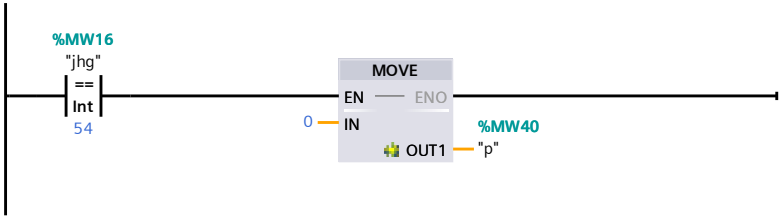
Network 1:



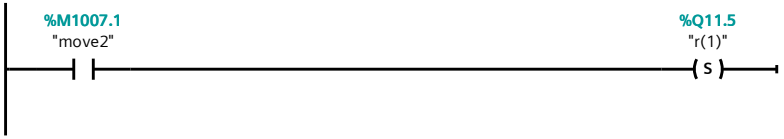
Network 2:



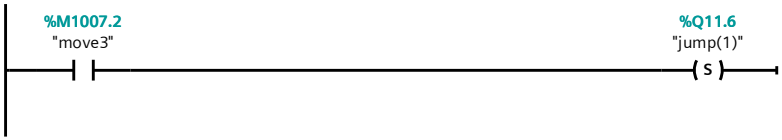
Network 3:



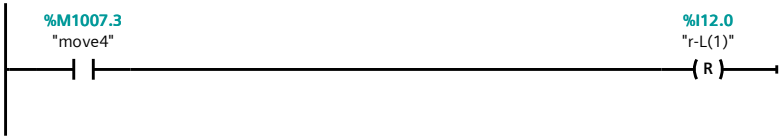
Network 4:



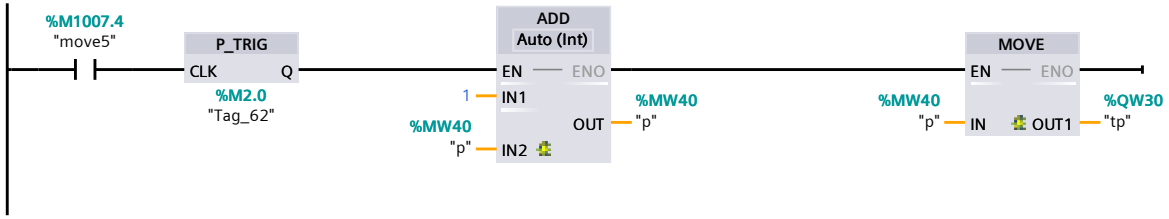
Network 5:



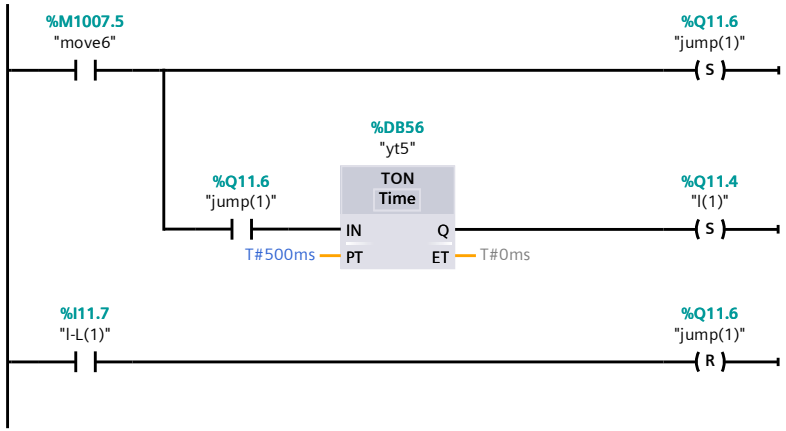
Network 6:



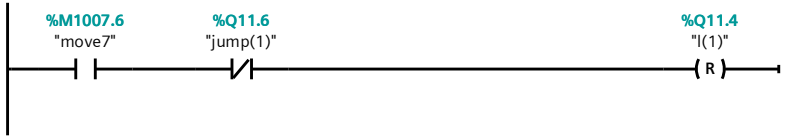
Network 7:



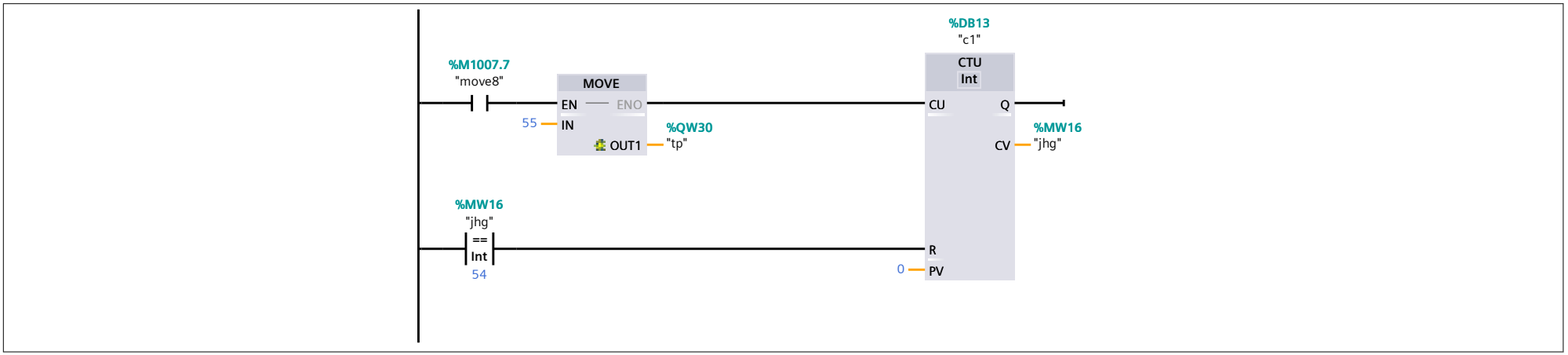
Network 8:



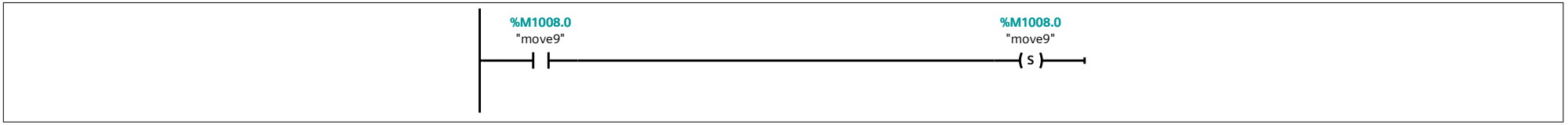
Network 9:



Network 10:


%M1007.7
"move8"
MOVE
EN
ENO
55
IN
OUT1
%QW30
"tp"
%DB13
"c1"
CTU
Int
CU
Q
CV
%MW16
"jhg"
%MW16
"jhg"
==
Int
54
R
0
PV

Network 11:

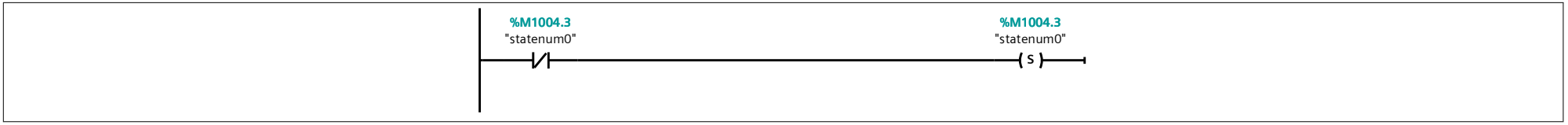

%M1008.0
"move9"
%M1008.0
"move9"
S

actuatorunload1 [FC22]

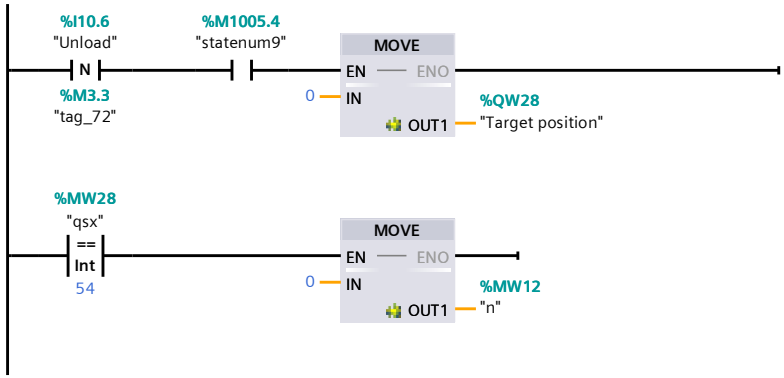
actuatorunload1 Properties							
General							
Name	actuatorunload1	**Number**	22	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
▼ Temp			
tag_23	Bool		
Constant			
▼ Return			
actuatorunload1	Void		

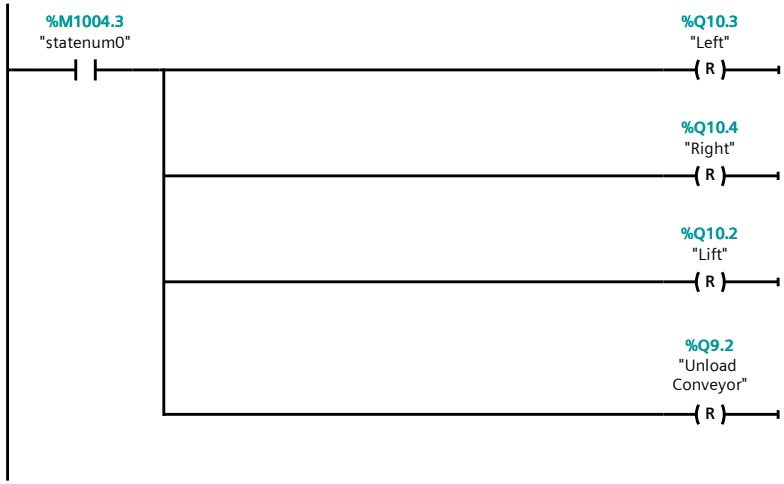
Network 1:



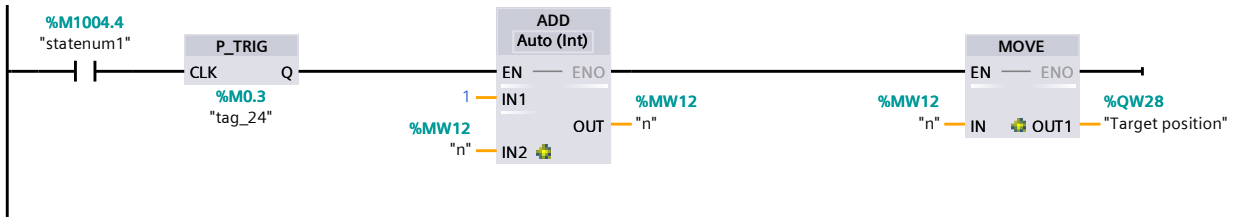
Network 2:



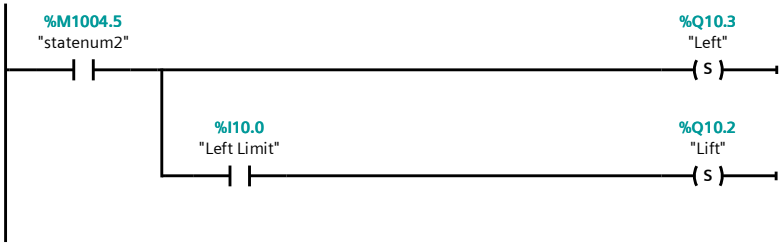
Network 3:



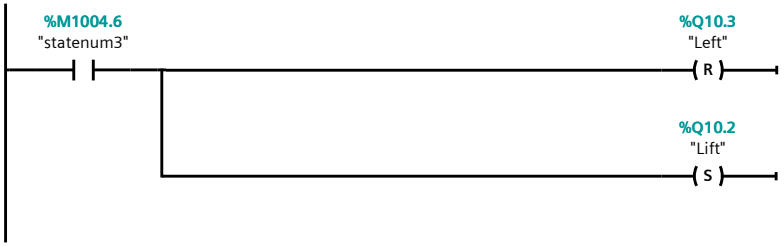
Network 4:



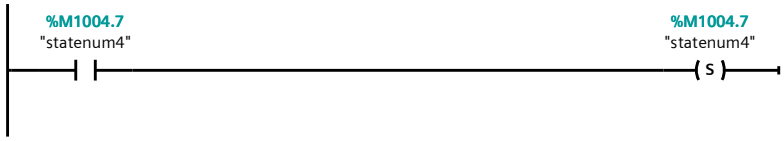
Network 5:



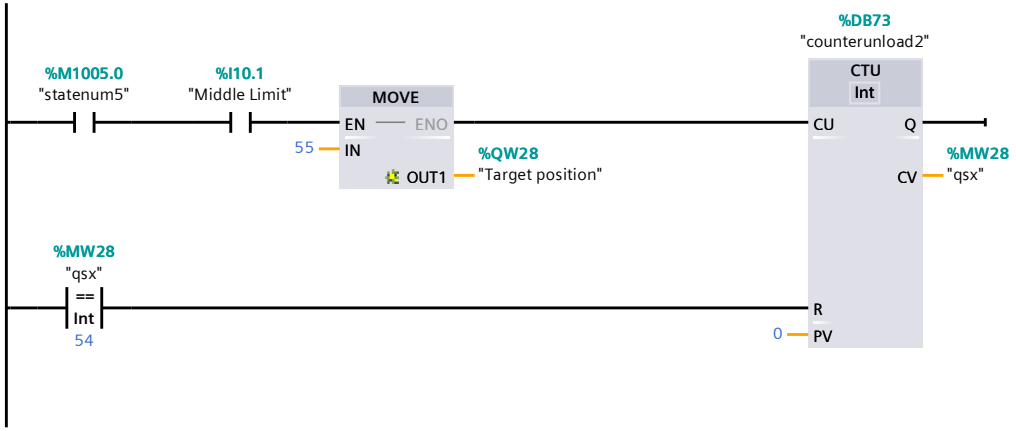
Network 6:



Network 7:



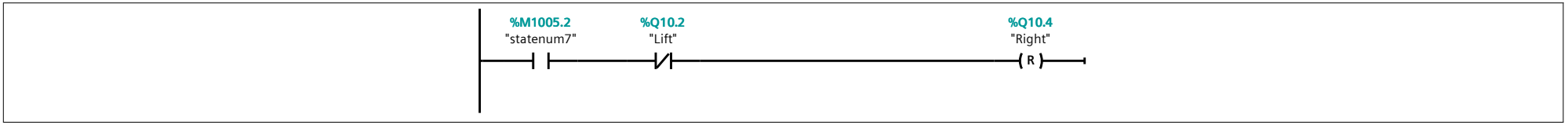
Network 8:



Network 9:



Network 10:



Network 11:

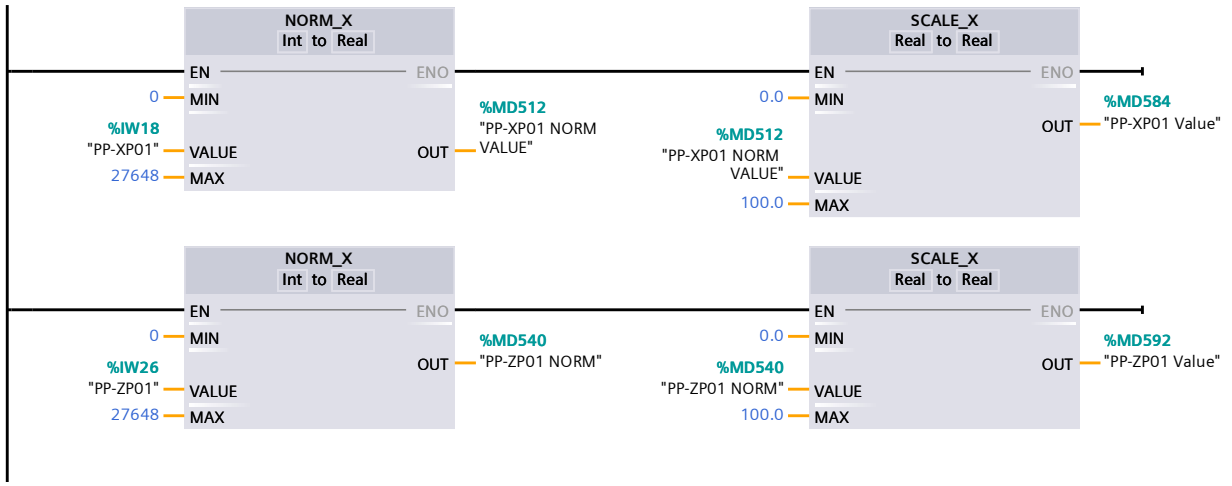


assembler [FC1]

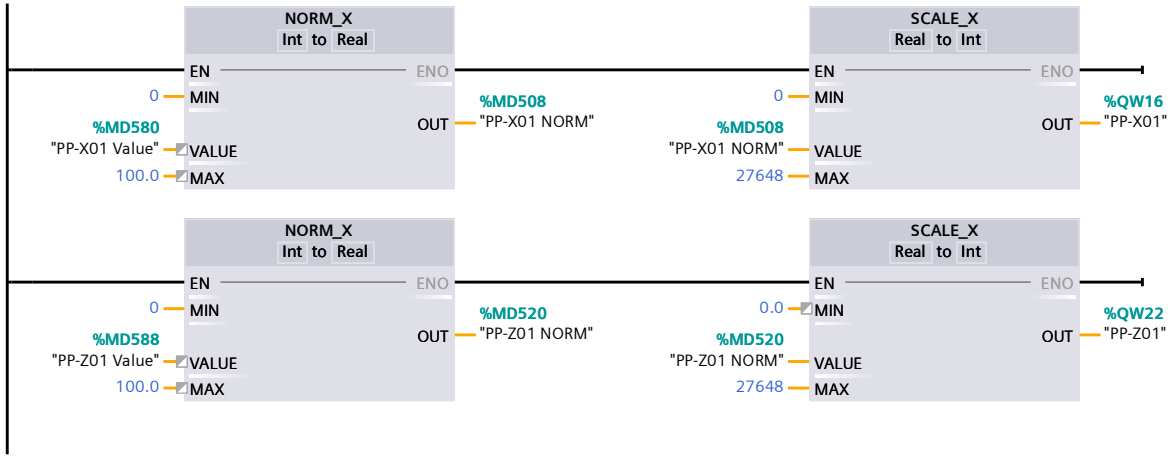
assembler Properties							
General							
Name	assembler	**Number**	1	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
assembler	Void		

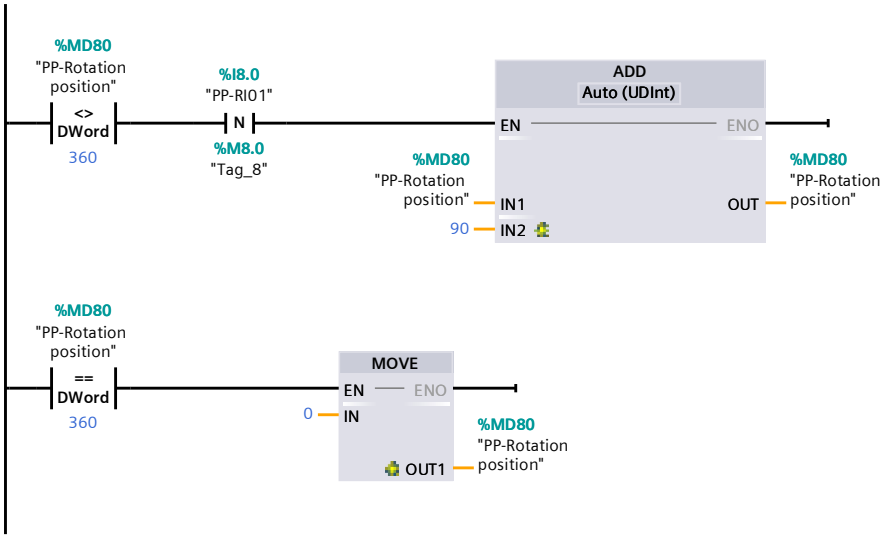
Network 1: Position Sensor scaleing



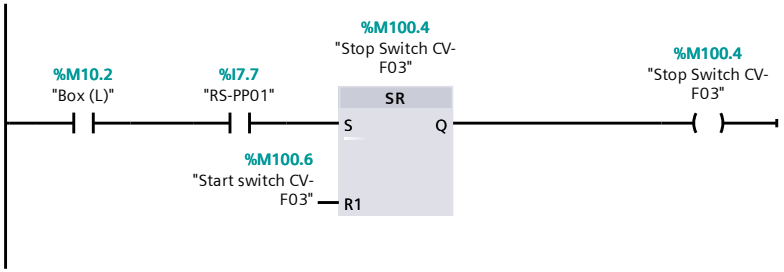
Network 2:



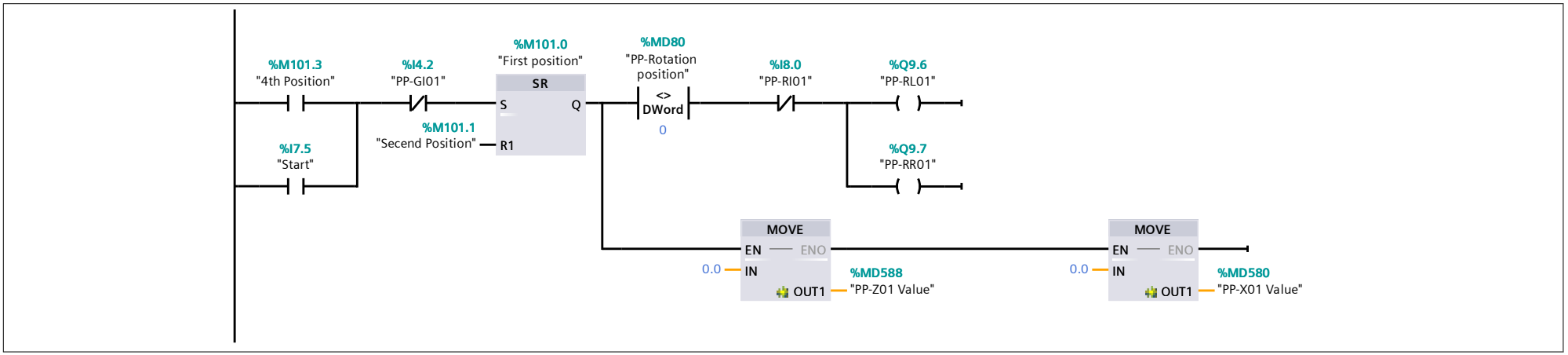
Network 3:



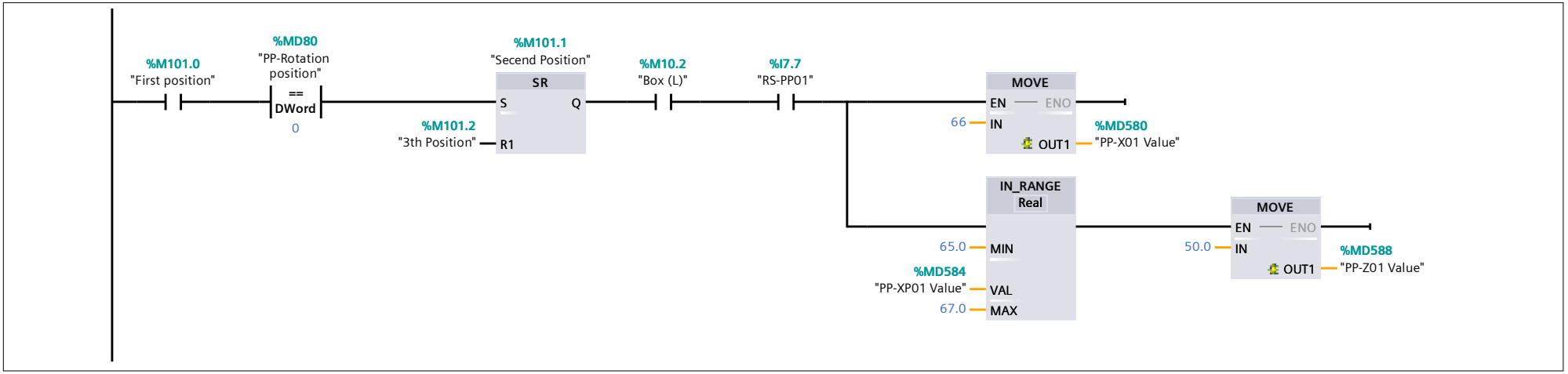
Network 4:



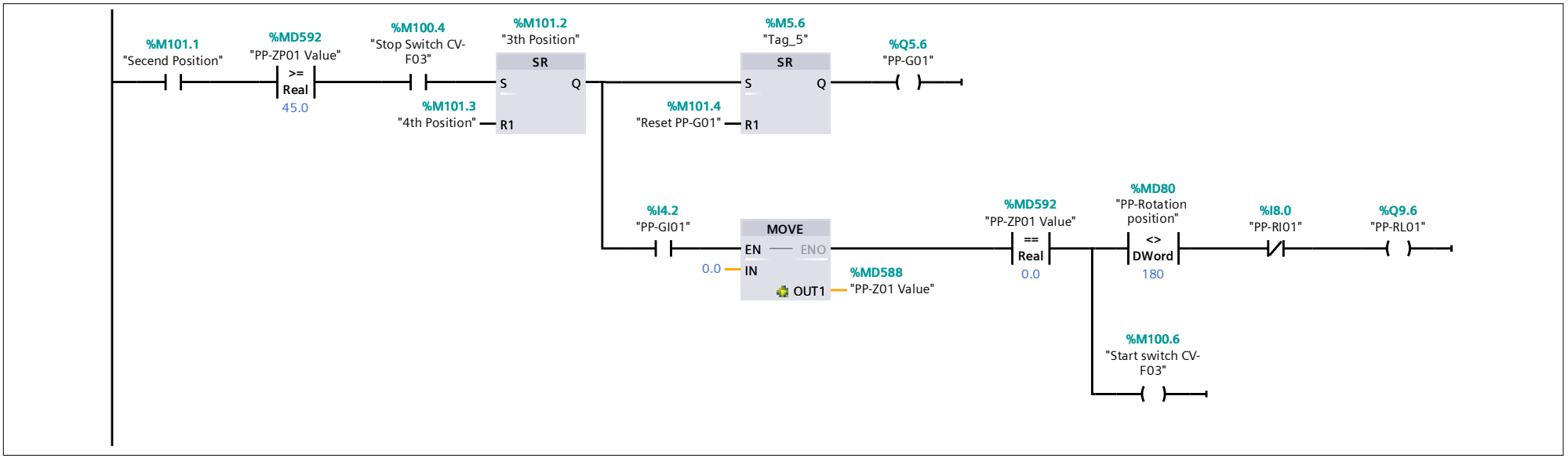
Network 5:



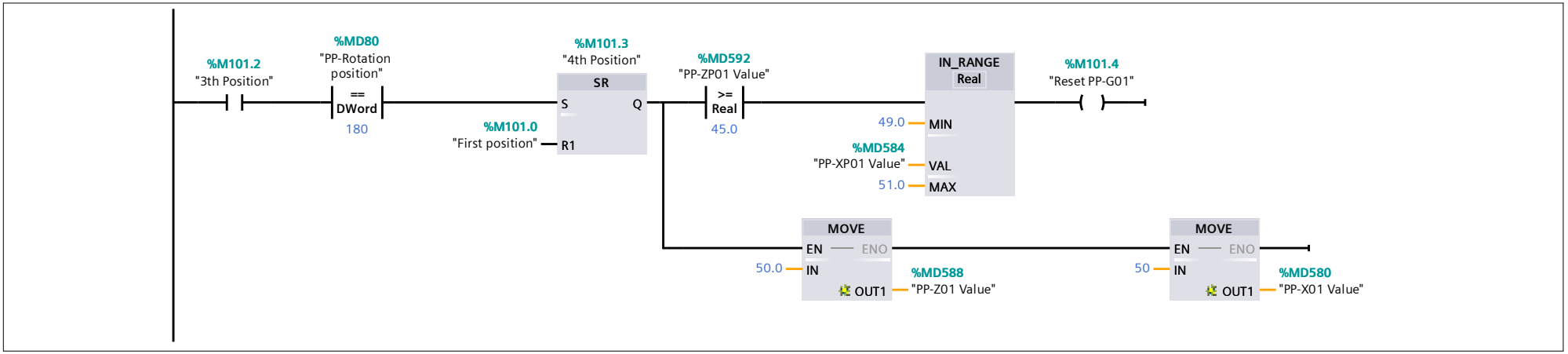
Network 6:



Network 7:


%M101.1
"Secend Position"
%MD592
"PP-ZP01 Value"
>=
Real
45.0
%M100.4
"Stop Switch CV-F03"
%M101.2
"3th Position"
SR
S
Q
%M101.3
"4th Position"
R1
%M5.6
"Tag_5"
SR
S
Q
%M101.4
"Reset PP-G01"
R1
%Q5.6
"PP-G01"
%I4.2
"PP-GI01"
MOVE
EN
ENO
0.0
IN
OUT1
%MD588
"PP-Z01 Value"
%MD592
"PP-ZP01 Value"
==
Real
0.0
%MD80
"PP-Rotation position"
<>
DWord
180
%I8.0
"PP-RI01"
%Q9.6
"PP-RL01"
%M100.6
"Start switch CV-F03"

Network 8:

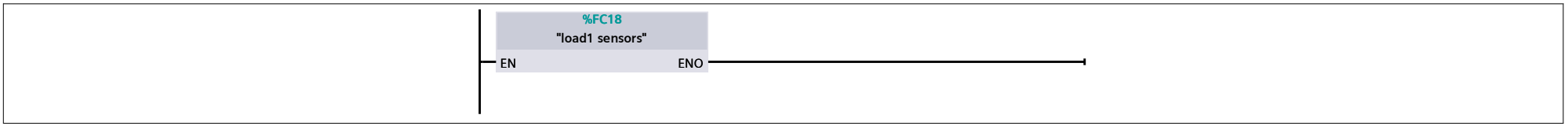

%M101.2
"3th Position"
%MD80
"PP-Rotation position"
==
DWord
180
%M101.3
"4th Position"
SR
S
Q
%M101.0
"First position"
R1
%MD592
"PP-ZP01 Value"
>=
Real
45.0
IN_RANGE
Real
49.0
MIN
%MD584
"PP-XP01 Value"
VAL
51.0
MAX
%M101.4
"Reset PP-G01"
MOVE
EN
ENO
50.0
IN
OUT1
%MD588
"PP-Z01 Value"
MOVE
EN
ENO
50
IN
OUT1
%MD580
"PP-X01 Value"

load_1 [FC19]

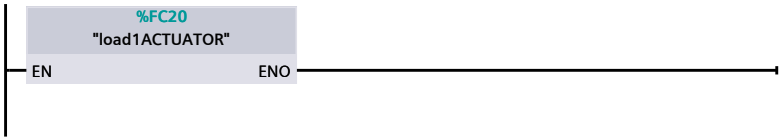
load_1 Properties							
General							
Name	load_1	**Number**	19	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
load_1	Void		

Network 1:



Network 2:

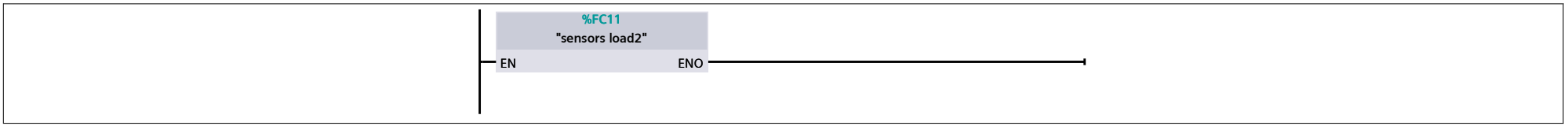

%FC20
"load1ACTUATOR"
EN
ENO

load_2 [FC25]

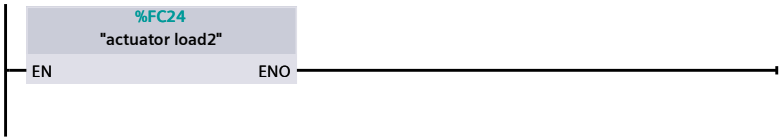
load_2 Properties							
General							
Name	load_2	**Number**	25	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
load_2	Void		

Network 1:



Network 2:

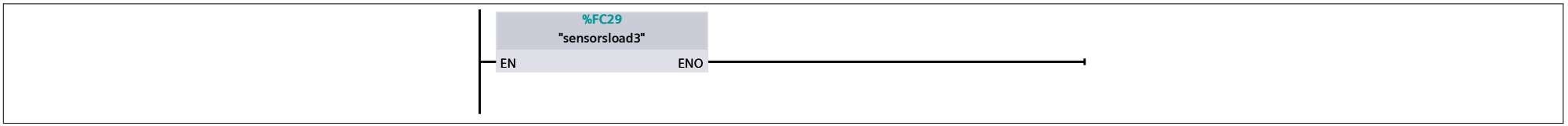

%FC24
"actuator load2"
EN
ENO

load_3 [FC31]

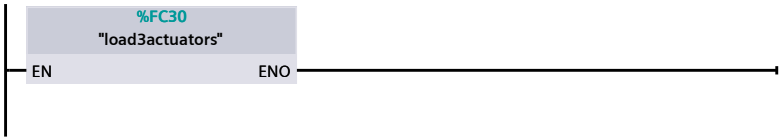
load_3 Properties							
General							
Name	load_3	**Number**	31	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
load_3	Void		

Network 1:



Network 2:

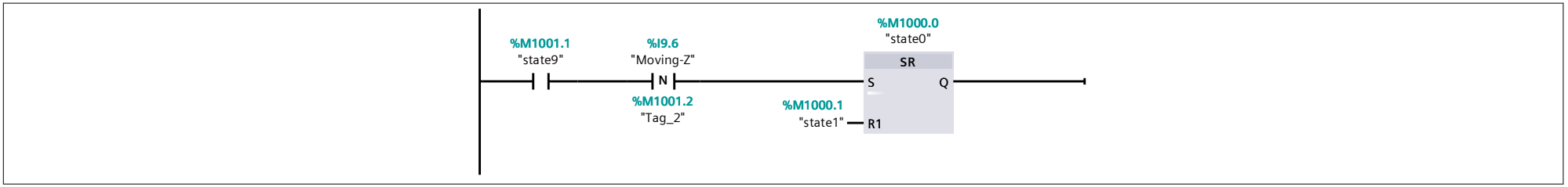

%FC30
"load3actuators"
EN
ENO

load1 sensors [FC18]

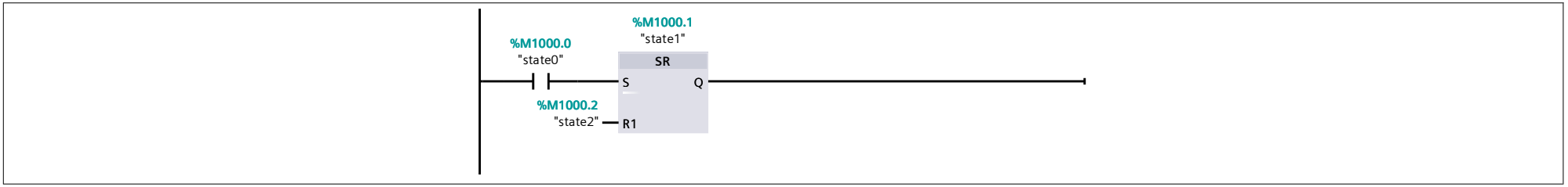
load1 sensors Properties							
General							
Name	load1 sensors	**Number**	18	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
▼ Temp			
position	Int		
position2	Int		
e	Int		
Constant			
▼ Return			
load1 sensors	Void		

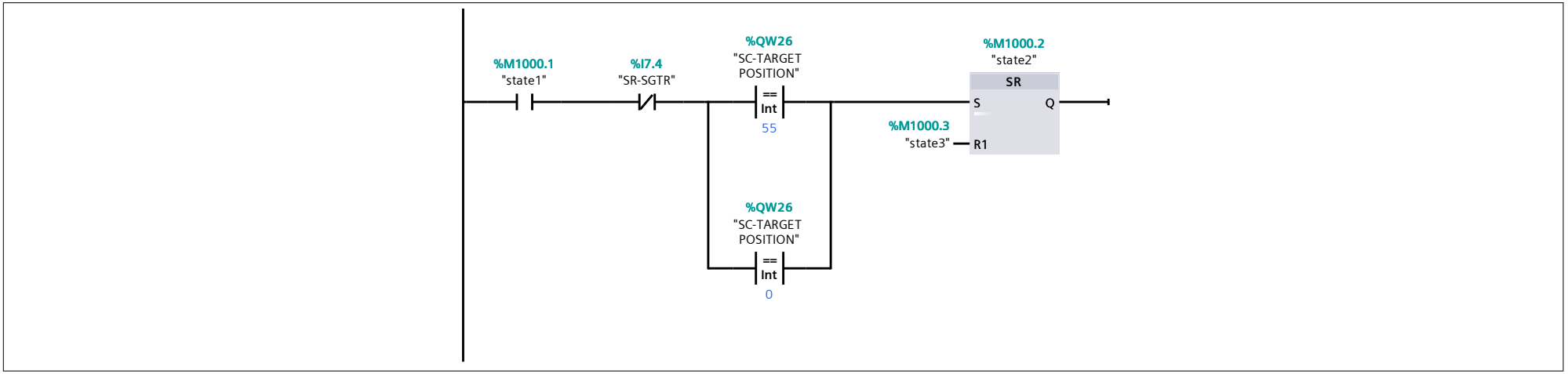
Network 1:



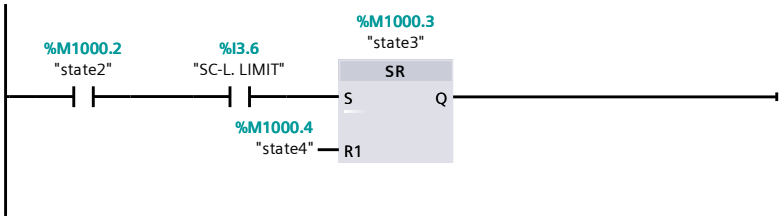
Network 2:



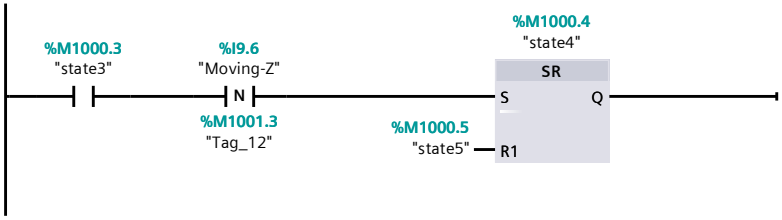
Network 3:



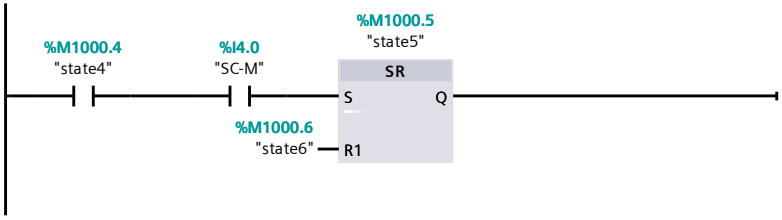
Network 4:



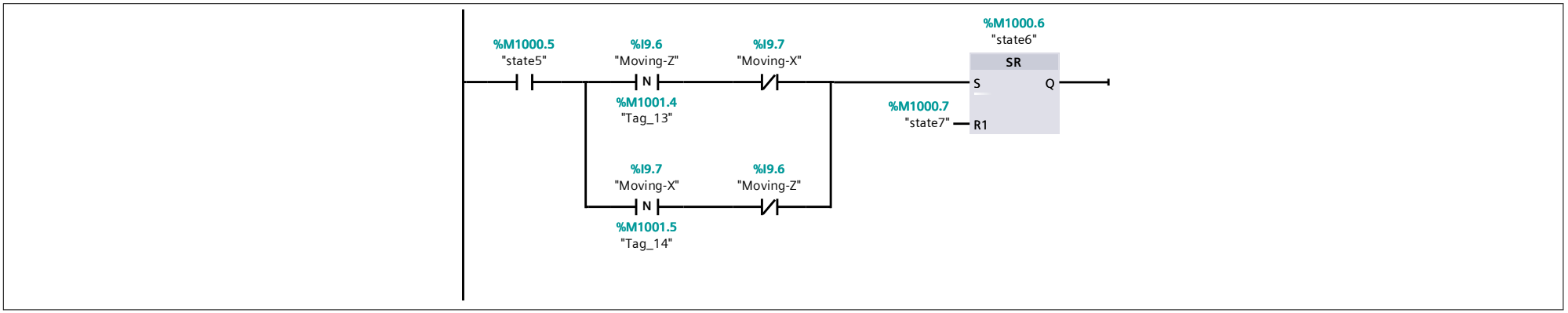
Network 5:



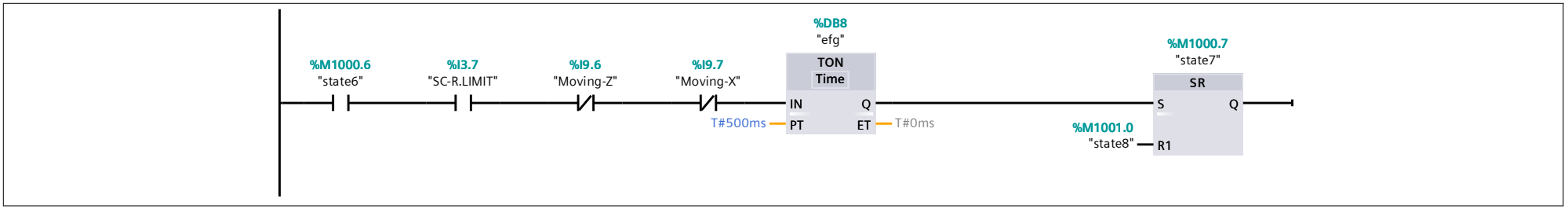
Network 6:



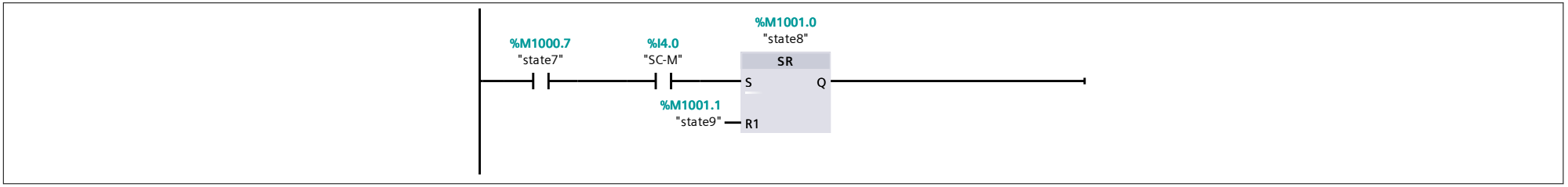
Network 7:



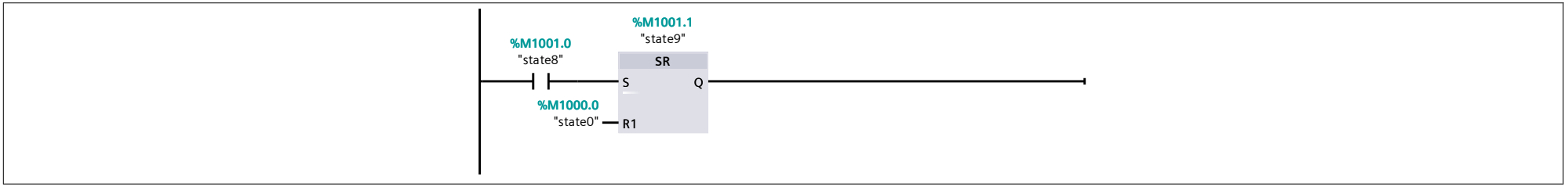
Network 8:



Network 9:



Network 10:

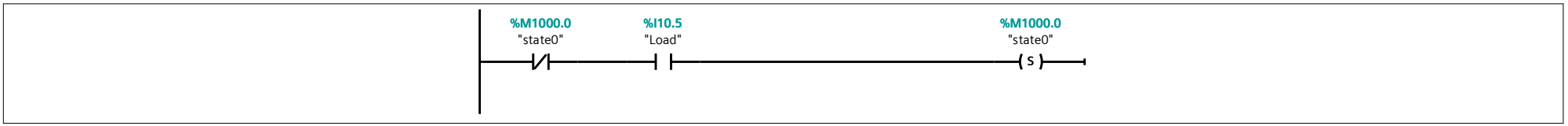


load1ACTUATOR [FC20]

load1ACTUATOR Properties							
General							
Name	load1ACTUATOR	**Number**	20	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
load1ACTUATOR	Void		

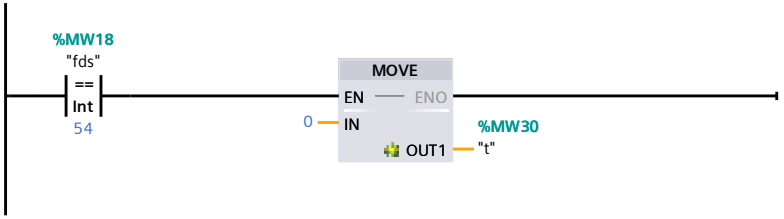
Network 1:



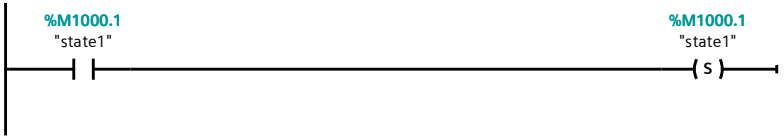
Network 2:



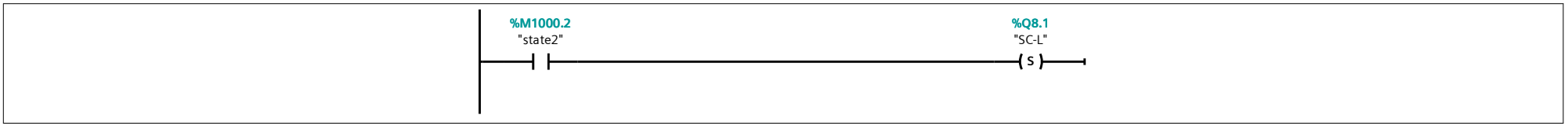
Network 3:



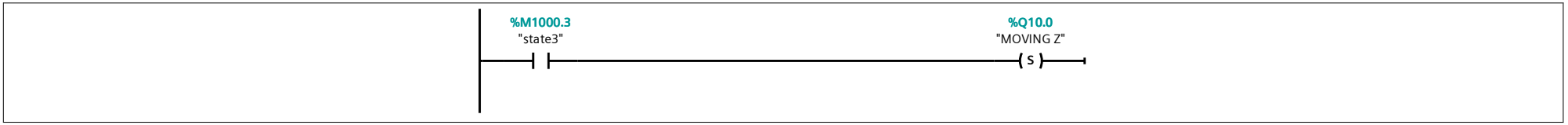
Network 4:



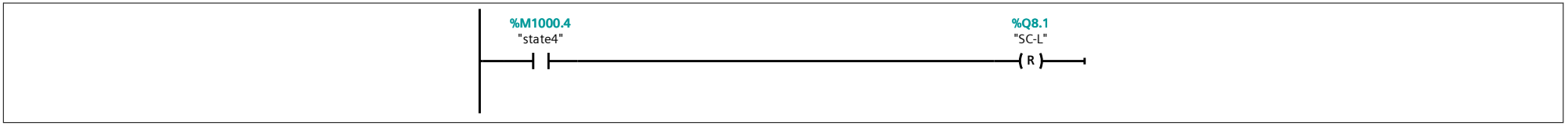
Network 5:



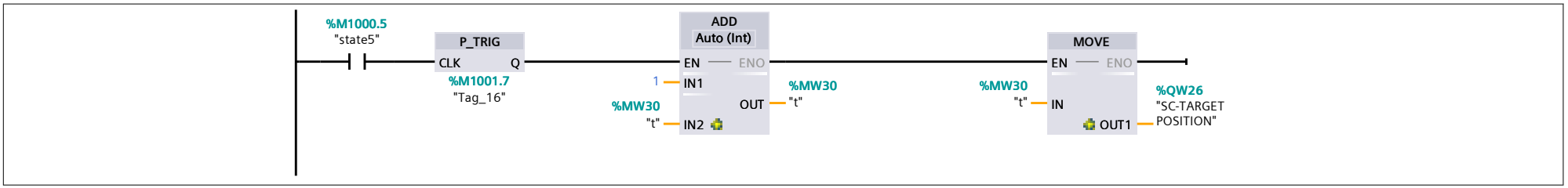
Network 6:



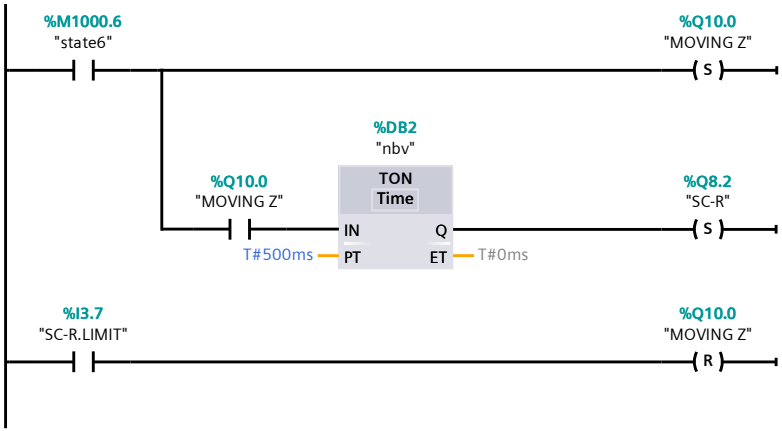
Network 7:



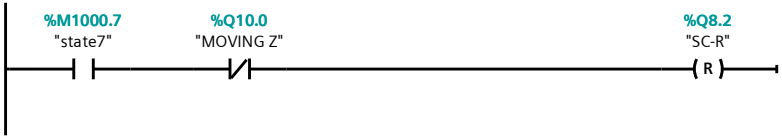
Network 8:



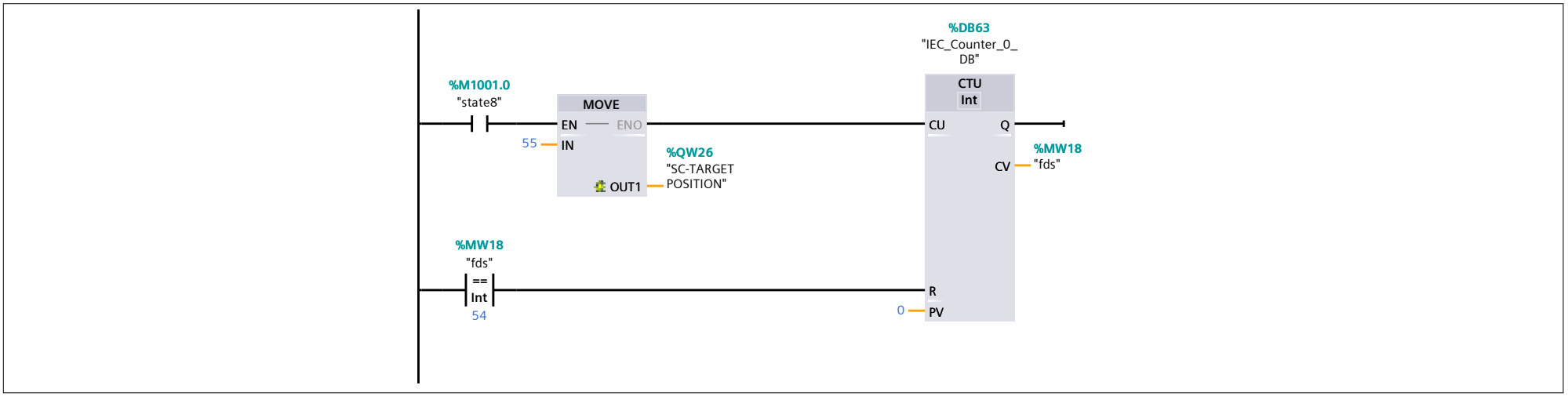
Network 9:



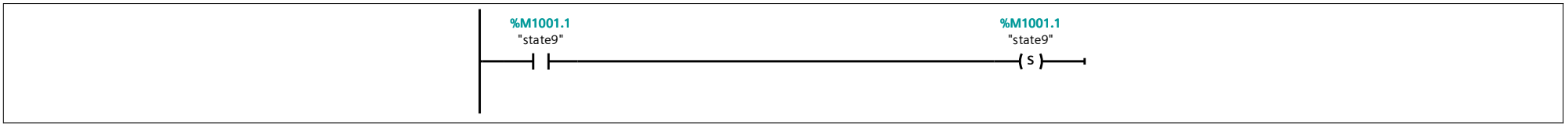
Network 10:



Network 11:


%DB63
"IEC_Counter_0_
DB"
%M1001.0
"state8"
MOVE
EN
ENO
55
IN
OUT1
%QW26
"SC-TARGET
POSITION"
CTU
Int
CU
Q
CV
%MW18
"fds"
%MW18
"fds"
==
Int
54
R
0
PV

Network 12:

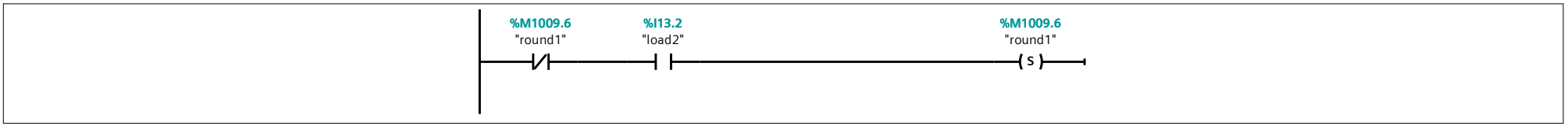

%M1001.1
"state9"
%M1001.1
"state9"
S

load3actuators [FC30]

load3actuators Properties							
General							
Name	load3actuators	**Number**	30	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
load3actuators	Void		

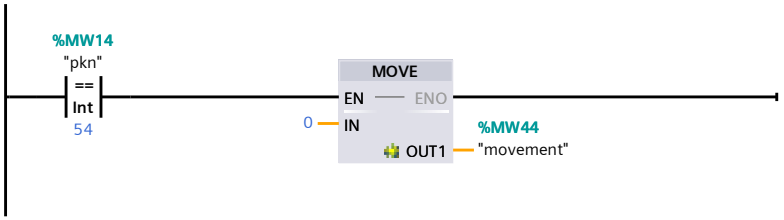
Network 1:



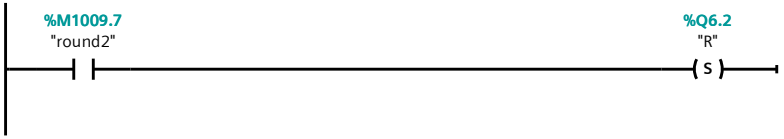
Network 2:



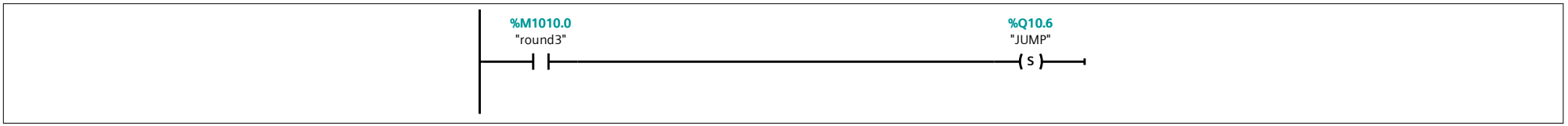
Network 3:



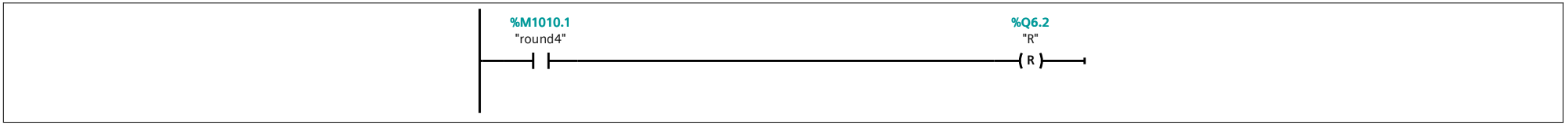
Network 4:



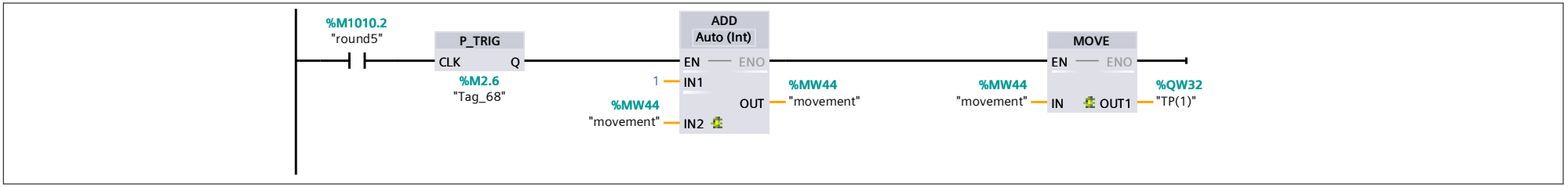
Network 5:



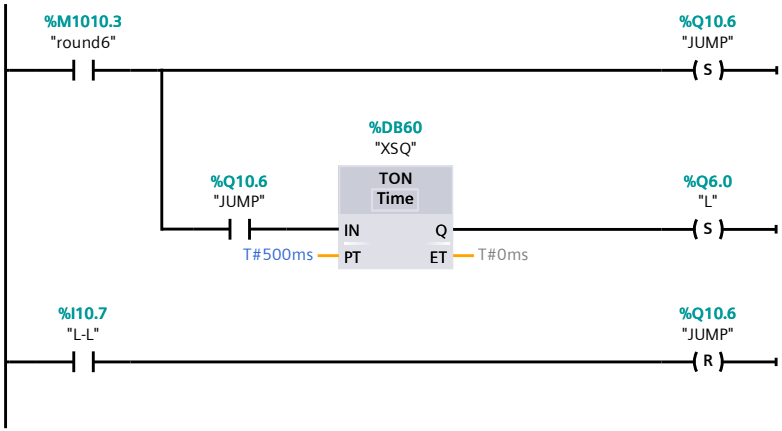
Network 6:



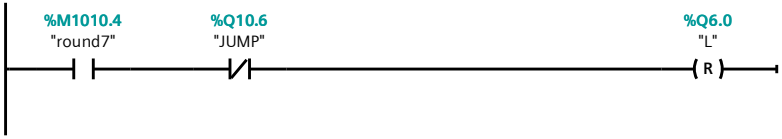
Network 7:



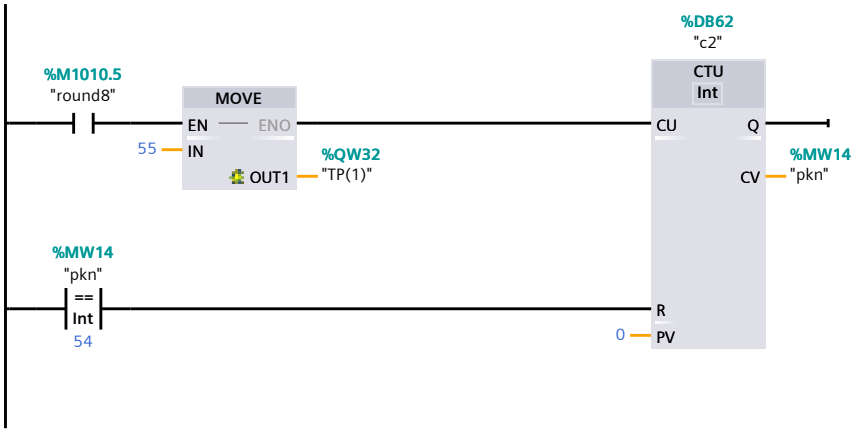
Network 8:



Network 9:



Network 10:

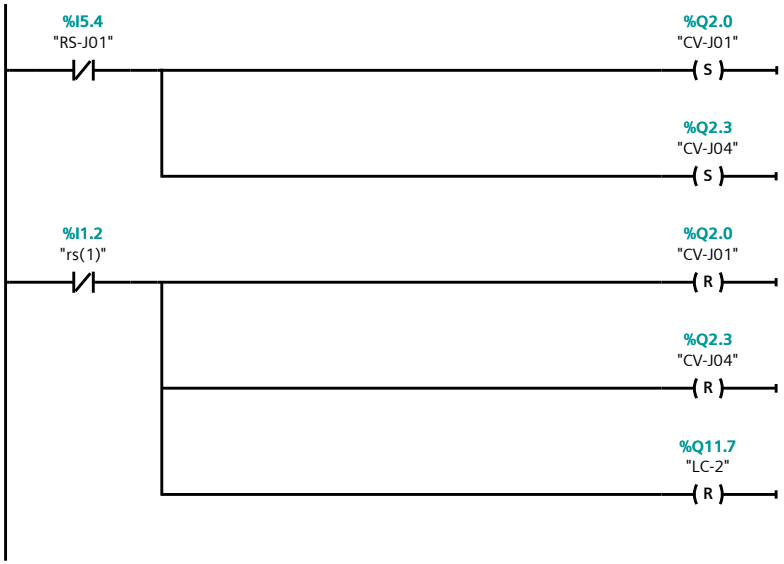

%M1010.5
"round8"
MOVE
EN
ENO
55
IN
OUT1
%QW32
"TP(1)"
%DB62
"c2"
CTU
Int
CU
Q
CV
%MW14
"pkn"
%MW14
"pkn"
==
Int
54
R
0
PV

loading conveyors [FC13]

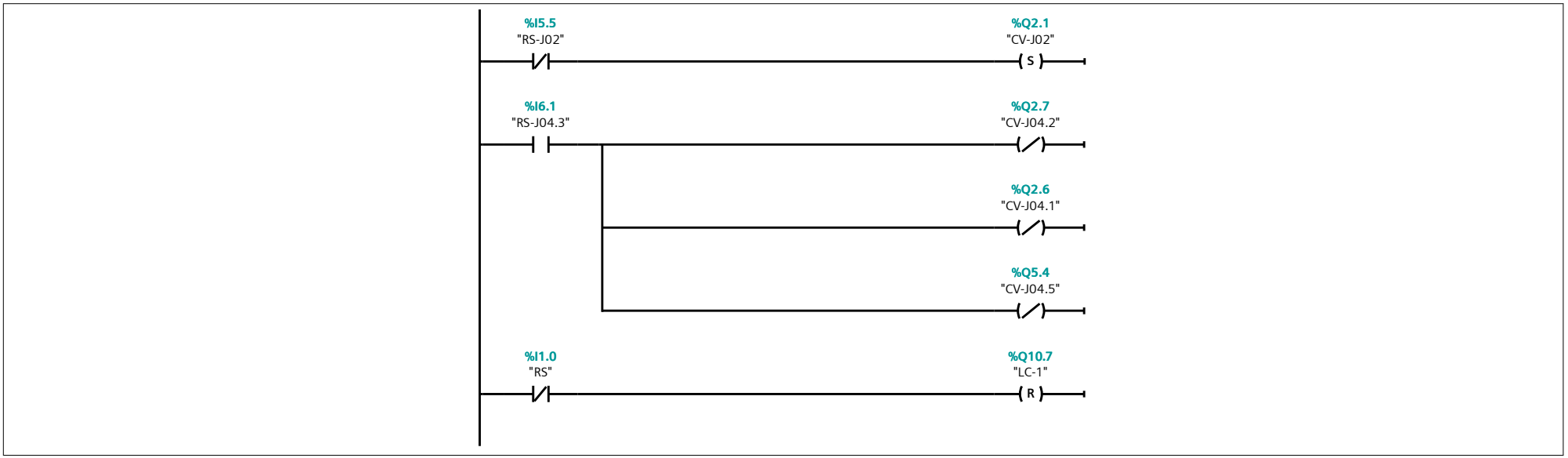
loading conveyors Properties							
General							
Name	loading conveyors	**Number**	13	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
loading conveyors	Void		

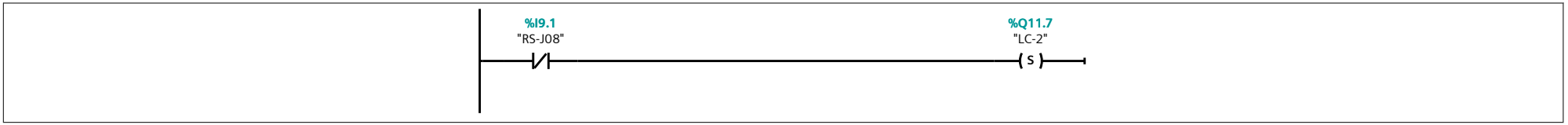
Network 1:


%I5.4
"RS-J01"
%Q2.0
"CV-J01"
S
%Q2.3
"CV-J04"
S
%I1.2
"rs(1)"
%Q2.0
"CV-J01"
R
%Q2.3
"CV-J04"
R
%Q11.7
"LC-2"
R

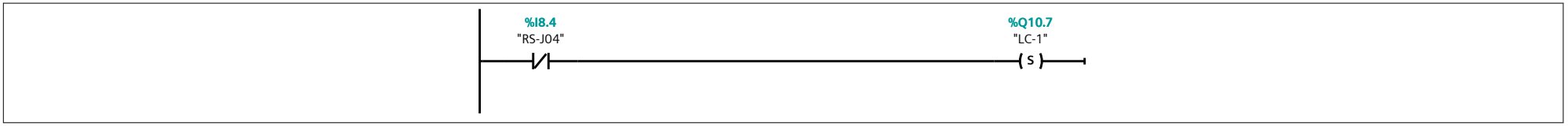
Network 2:



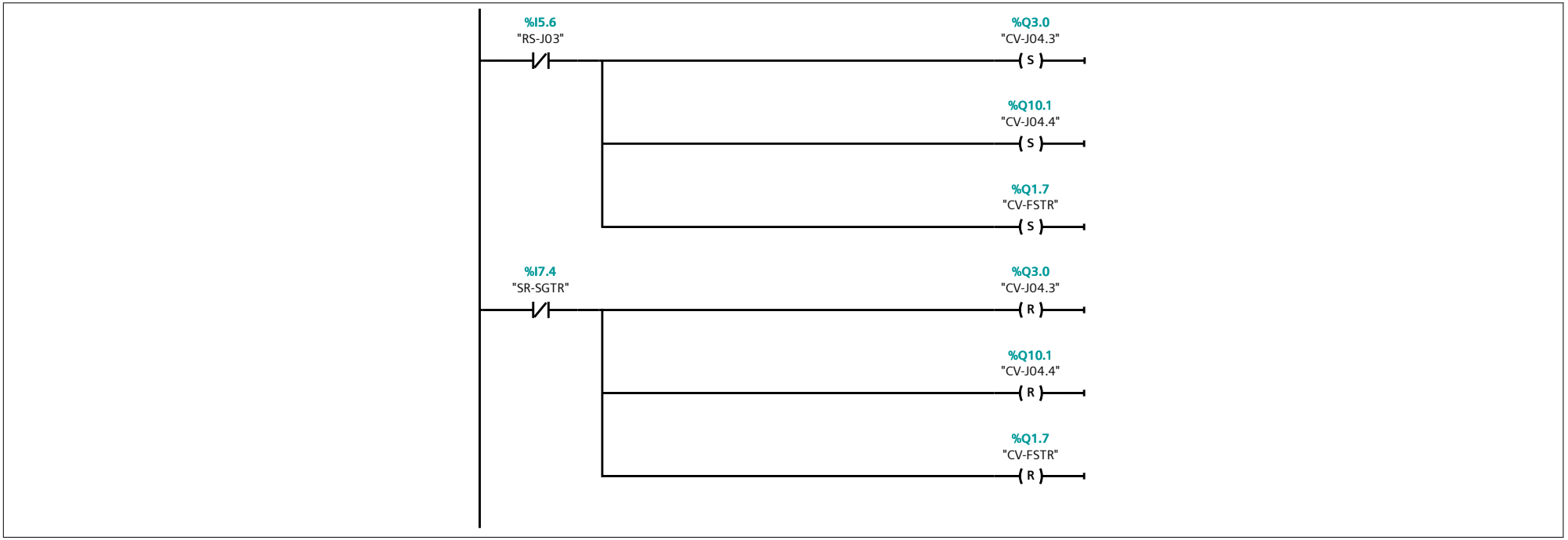
Network 3:



Network 4:



Network 5:

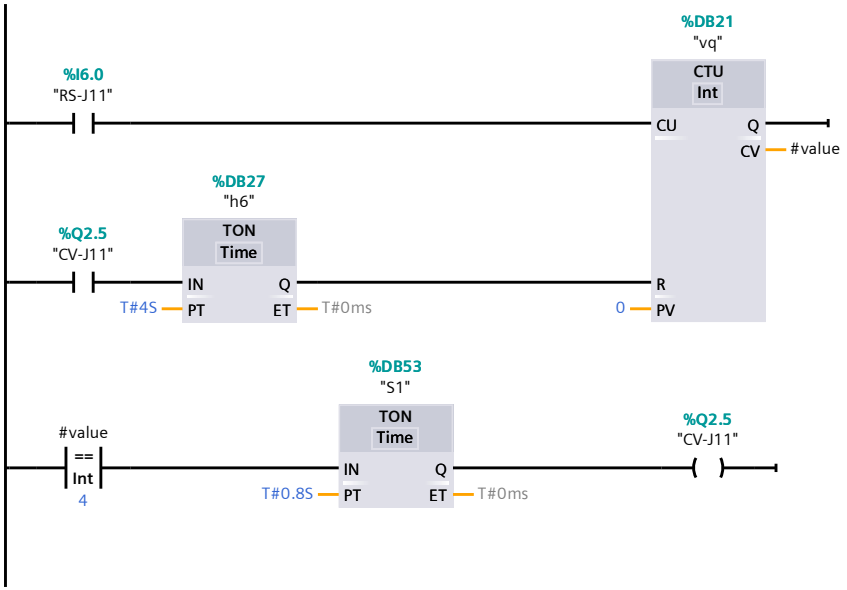

%I5.6
"RS-J03"
%Q3.0
"CV-J04.3"
S
%Q10.1
"CV-J04.4"
S
%Q1.7
"CV-FSTR"
S
%I7.4
"SR-SGTR"
%Q3.0
"CV-J04.3"
R
%Q10.1
"CV-J04.4"
R
%Q1.7
"CV-FSTR"
R

manual_storage [FC17]

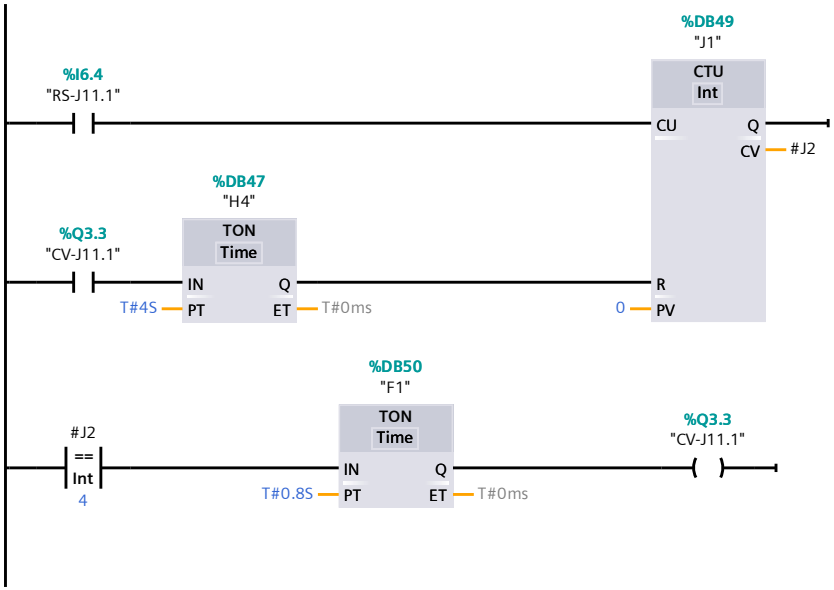
manual_storage Properties							
General							
Name	manual_storage	**Number**	17	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
▼ Temp			
value	Int		
value2	Int		
value3	Int		
tre	Bool		
trw	Bool		
trq	Bool		
G2	Int		
J2	Int		
Constant			
▼ Return			
manual_storage	Void		

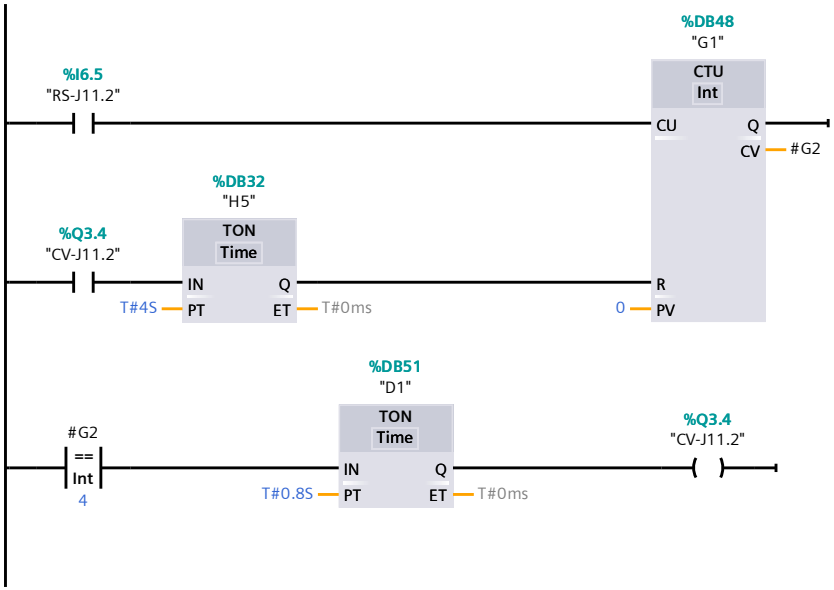
Network 1:


%DB21
"vq"
%I6.0
"RS-J11"
CTU
Int
CU
Q
CV
#value
%DB27
"h6"
%Q2.5
"CV-J11"
TON
Time
IN
Q
T#4S
PT
ET
T#0ms
R
0
PV
%DB53
"S1"
#value
==
Int
4
TON
Time
IN
Q
T#0.8S
PT
ET
T#0ms
%Q2.5
"CV-J11"

Network 2:


%DB49
"J1"
%I6.4
"RS-J11.1"
CTU
Int
CU
Q
CV
#J2
%DB47
"H4"
%Q3.3
"CV-J11.1"
TON
Time
IN
Q
T#4S
PT
ET
T#0ms
R
0
PV
%DB50
"F1"
#J2
==
Int
4
TON
Time
IN
Q
T#0.8S
PT
ET
T#0ms
%Q3.3
"CV-J11.1"

Network 3:


%DB48
"G1"
%I6.5
"RS-J11.2"
CTU
Int
CU
Q
CV
#G2
%DB32
"H5"
%Q3.4
"CV-J11.2"
TON
Time
IN
Q
T#4S
PT
ET
T#0ms
R
0
PV
%DB51
"D1"
#G2
==
Int
4
TON
Time
IN
Q
T#0.8S
PT
ET
T#0ms
%Q3.4
"CV-J11.2"

MHJ-PLC-Lab-Function-S71500 [FC9000]

MHJ-PLC-Lab-Function-S71500 Properties							
General							
Name	MHJ-PLC-Lab-Function-S71500	**Number**	9000	**Type**	FC	**Language**	SCL
Numbering	Manual						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
▼ Temp			
Value	Byte		
ForCounter	Int		
▼ Constant			
Value_01_DW	DWord	16#1223_5486	
Value_02_DW	DWord	16#A6C9_D1F5	
▼ Return			
MHJ-PLC-Lab-Function-S71500	Void		

```
0001
0002 #Value:=PEEK(area := 16#82,
0003    dbNumber := 0,
0004    byteOffset := 511);
0005 #Value := #Value + 1;
0006
0007 POKE(area := 16#82,
0008    dbNumber := 0,
0009    byteOffset := 511,
0010    value := #Value);
0011
```

```
0012 POKE(area := 16#82,
0013    dbNumber := 0,
0014    byteOffset := 1016,
0015    value := #Value_01_DW);
0016 POKE(area := 16#82,
0017    dbNumber := 0,
0018    byteOffset := 1020,
0019    value := #Value_02_DW);
0020
0021 FOR #ForCounter := 0 TO 63 DO
0022   #Value:=PEEK(area := 16#1,
0023      dbNumber := 0,
0024      byteOffset := #ForCounter);
0025   POKE(area := 16#81,
0026      dbNumber := 0,
0027      byteOffset := #ForCounter,
0028      value := #Value);
0029 END_FOR;
0030 #Value := PEEK(area := 16#1,
0031          dbNumber := 0,
0032          byteOffset := 512);
0033 POKE(area := 16#82,
0034    dbNumber := 0,
0035    byteOffset := 512,
0036    value := #Value);
0037
0038
```

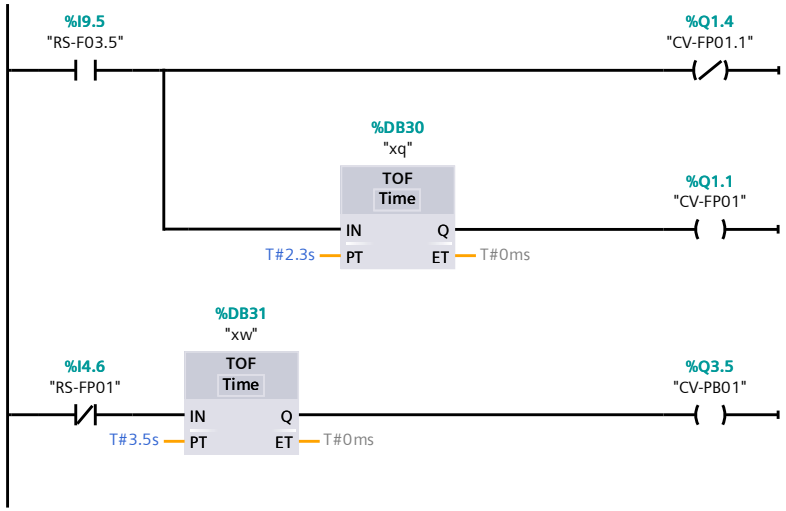
Symbol	Address	Type	Comment
#ForCounter		Int	
#Value		Byte	
#Value_01_DW	16#1223_5486	DWord	
#Value_02_DW	16#A6C9_D1F5	DWord	

palletizer_1 [FC15]

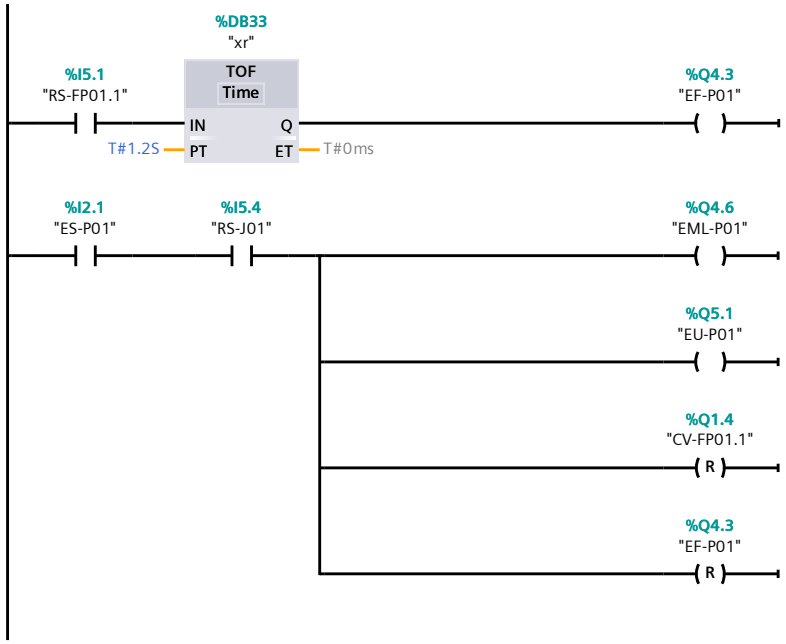
palletizer_1 Properties							
General							
Name	palletizer_1	**Number**	15	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
▼ Temp			
j1	Bool		
A1	Bool		
V1	Bool		
G1	Bool		
TGB	Bool		
Constant			
▼ Return			
palletizer_1	Void		

Network 1:

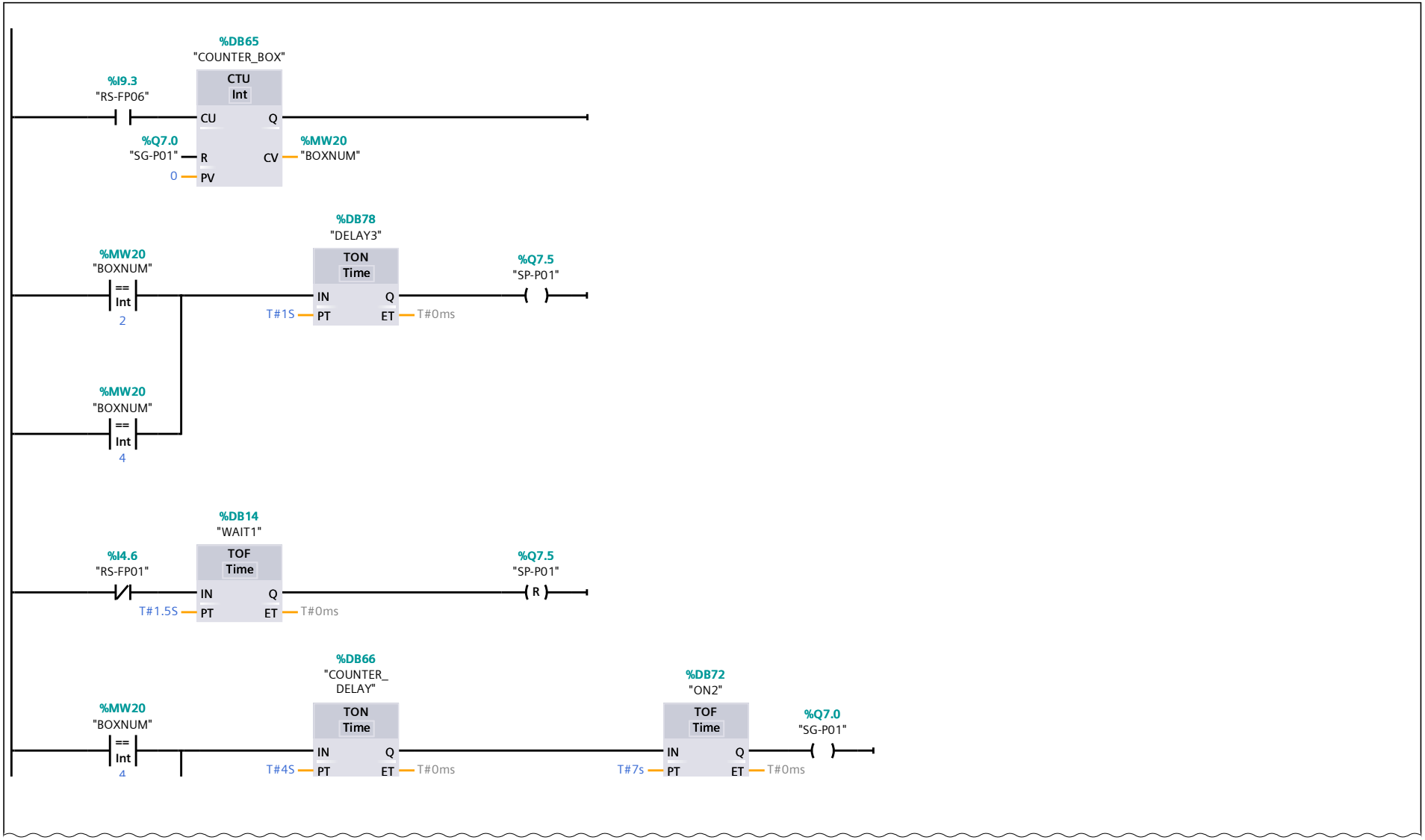

%I9.5
"RS-F03.5"
%Q1.4
"CV-FP01.1"
%DB30
"xq"
TOF
Time
IN
Q
T#2.3s
PT
ET
T#0ms
%Q1.1
"CV-FP01"
%DB31
"xw"
%I4.6
"RS-FP01"
TOF
Time
IN
Q
T#3.5s
PT
ET
T#0ms
%Q3.5
"CV-PB01"

Network 2:


%DB33
"xr"
%I5.1
"RS-FP01.1"
TOF
Time
IN
Q
T#1.2S
PT
ET
T#0ms
%Q4.3
"EF-P01"
%I2.1
"ES-P01"
%I5.4
"RS-J01"
%Q4.6
"EML-P01"
%Q5.1
"EU-P01"
%Q1.4
"CV-FP01.1"
R
%Q4.3
"EF-P01"
R

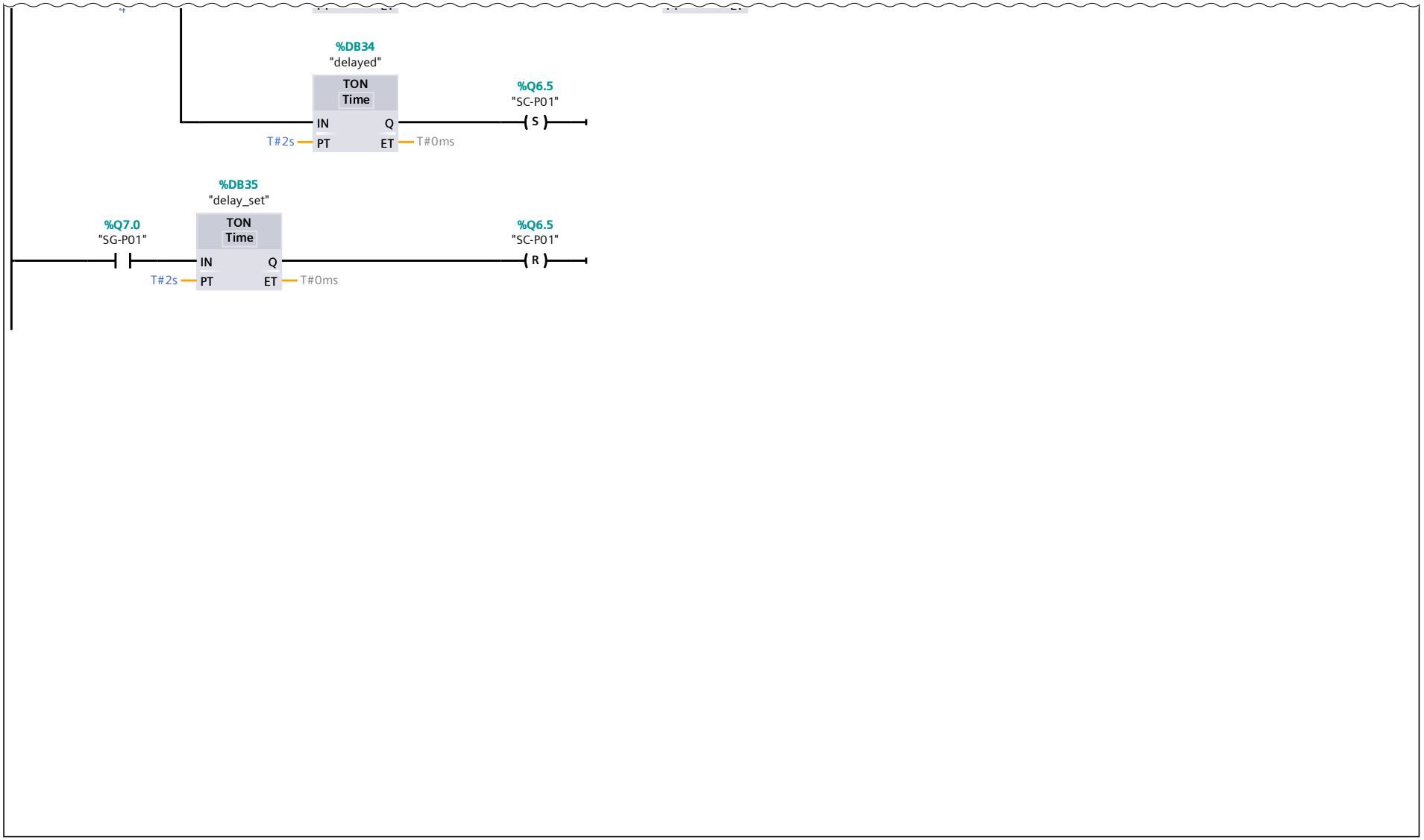
Network 3:

Network 3: (1.1 / 2.1)

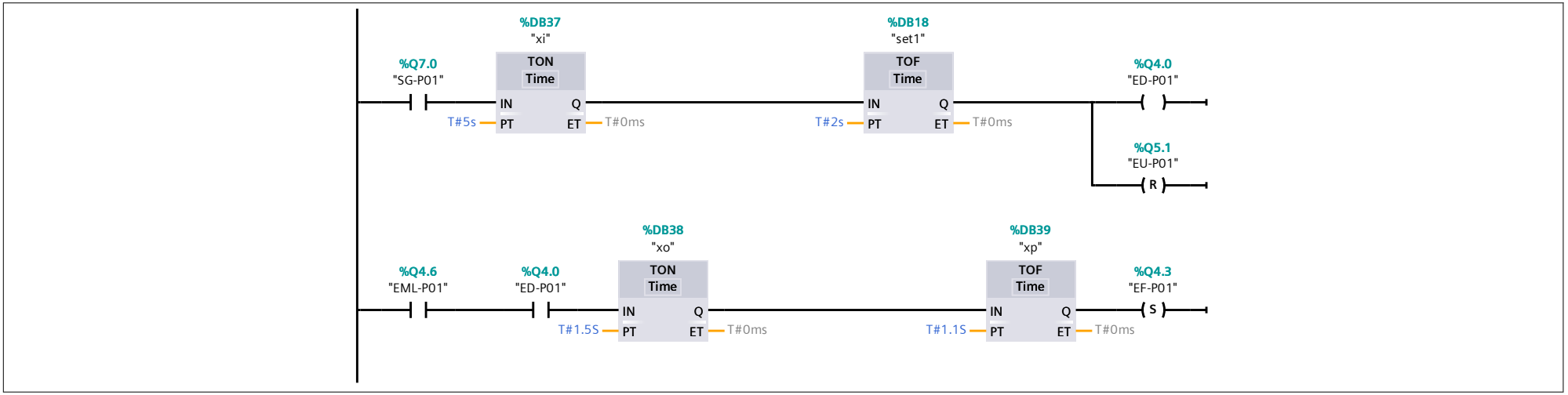


Network 3: (2.1 / 2.1)

1.1 (Page16 - 4)



Network 4:

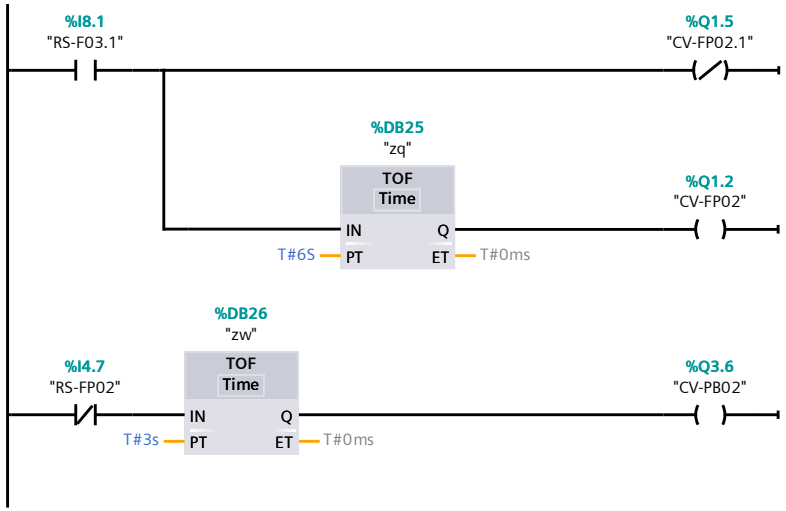

%DB37
"xi"
%Q7.0
"SG-P01"
TON
Time
IN
Q
T#5s
PT
ET
T#0ms
%DB18
"set1"
TOF
Time
IN
Q
T#2s
PT
ET
T#0ms
%Q4.0
"ED-P01"
%Q5.1
"EU-P01"
R
%DB38
"xo"
%Q4.6
"EML-P01"
%Q4.0
"ED-P01"
TON
Time
IN
Q
T#1.5S
PT
ET
T#0ms
%DB39
"xp"
TOF
Time
IN
Q
T#1.1S
PT
ET
T#0ms
%Q4.3
"EF-P01"
S

palletizer_2 [FC14]

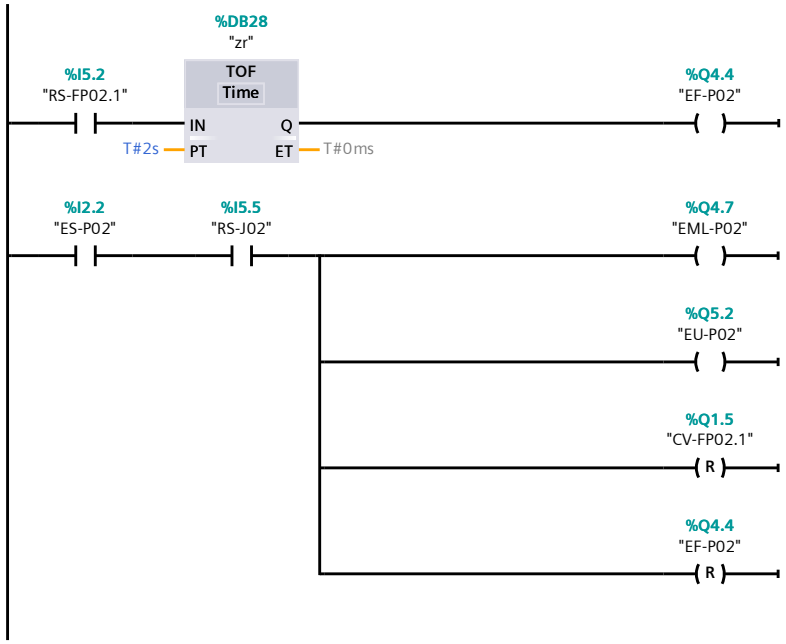
palletizer_2 Properties							
General							
Name	palletizer_2	**Number**	14	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
▼ Temp			
B1	Bool		
F1	Bool		
G1	Bool		
AZQ	Bool		
Constant			
▼ Return			
palletizer_2	Void		

Network 1:

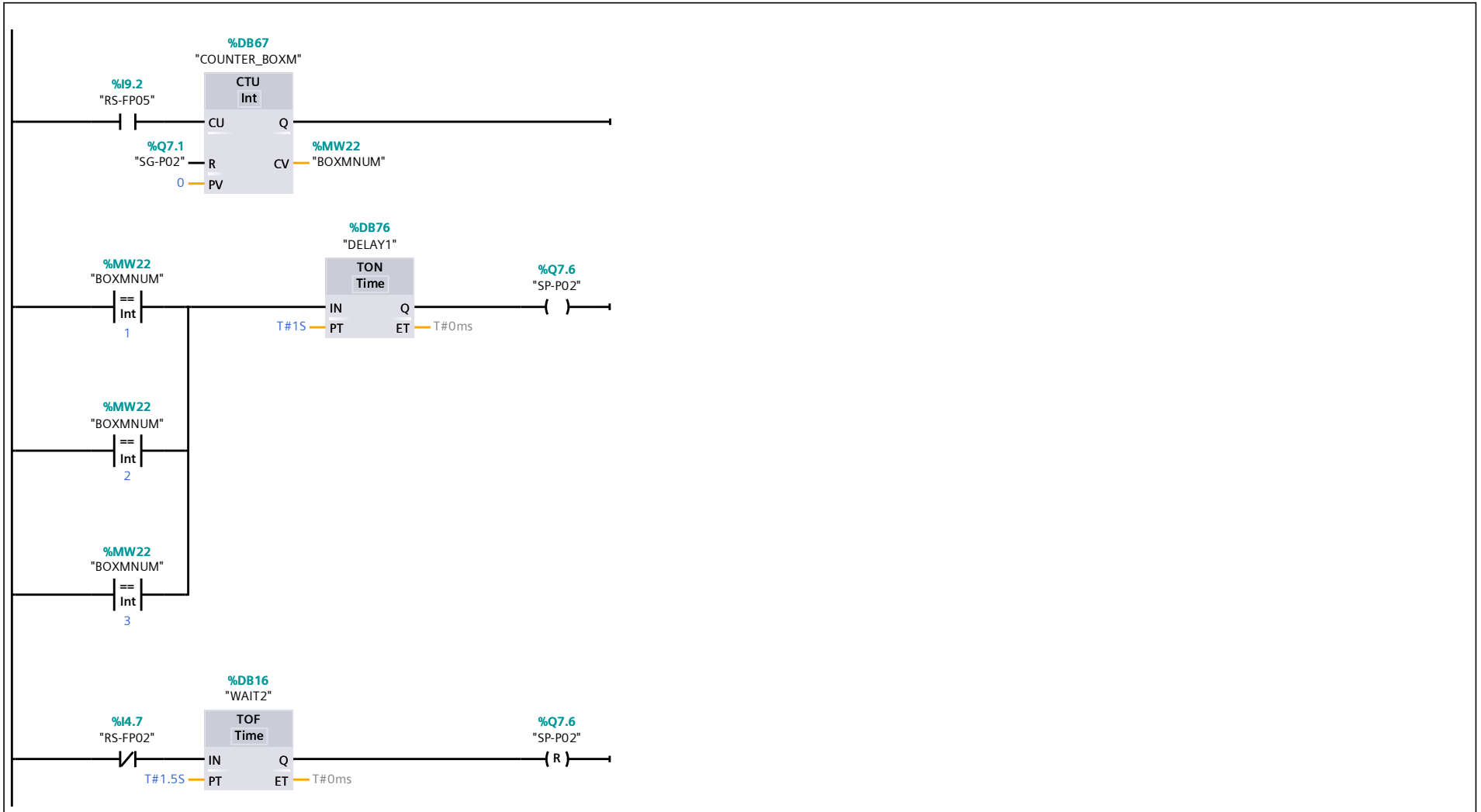

%I8.1
"RS-F03.1"
%Q1.5
"CV-FP02.1"
%DB25
"zq"
TOF
Time
IN
Q
T#6S
PT
ET
T#0ms
%Q1.2
"CV-FP02"
%DB26
"zw"
%I4.7
"RS-FP02"
TOF
Time
IN
Q
T#3s
PT
ET
T#0ms
%Q3.6
"CV-PB02"

Network 2:


%DB28
"zr"
%I5.2
"RS-FP02.1"
TOF
Time
IN
Q
T#2s
PT
ET
T#0ms
%Q4.4
"EF-P02"
%I2.2
"ES-P02"
%I5.5
"RS-J02"
%Q4.7
"EML-P02"
%Q5.2
"EU-P02"
%Q1.5
"CV-FP02.1"
R
%Q4.4
"EF-P02"
R

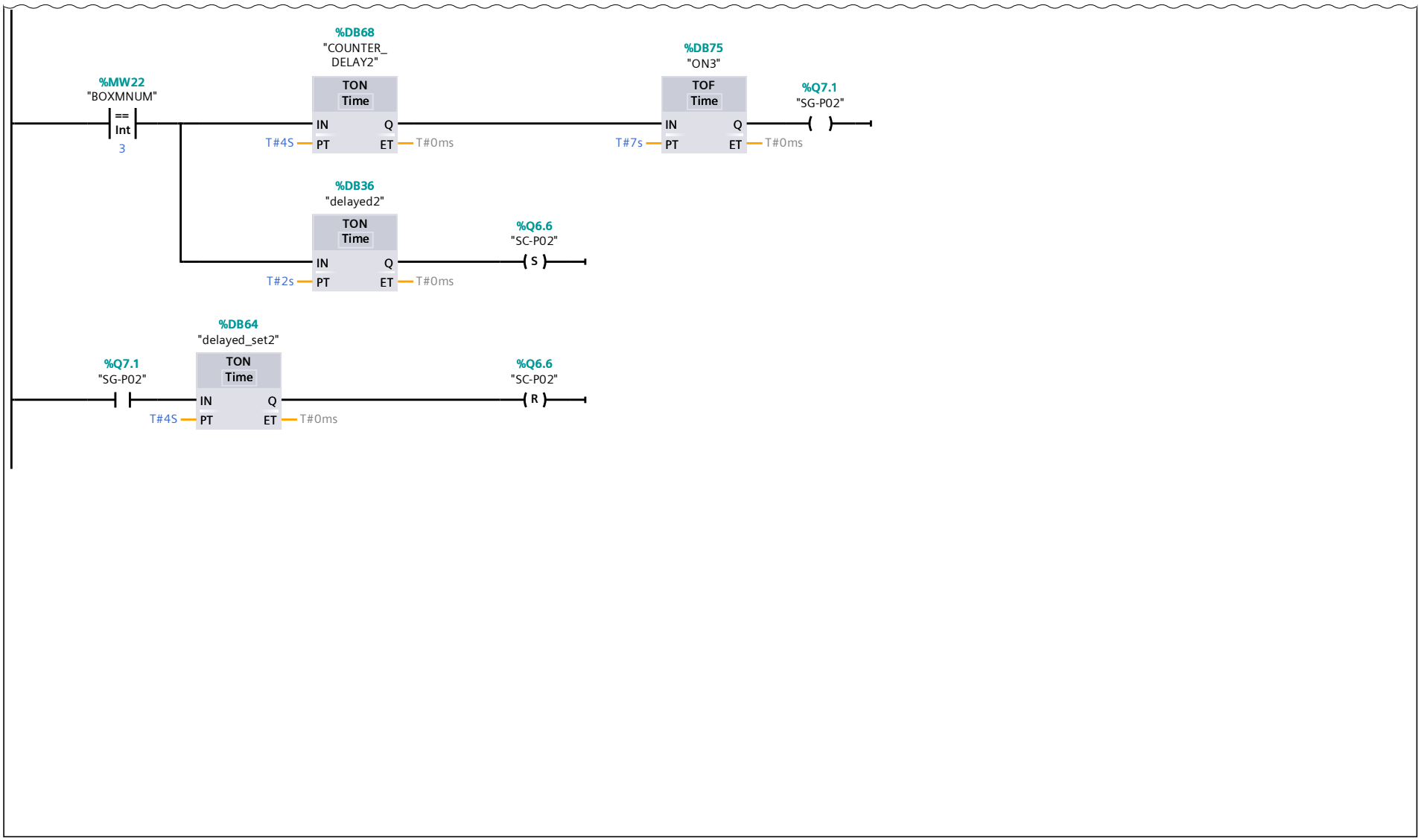
Network 3:

Network 3: (1.1 / 2.1)

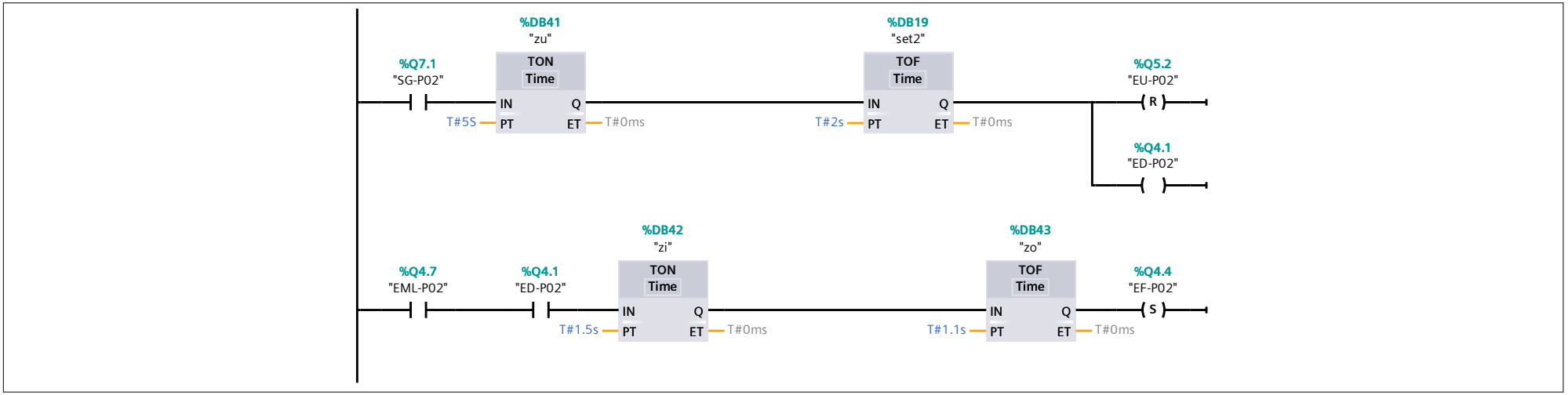


Network 3: (2.1 / 2.1)

1.1 (Page17 - 4)



Network 4:

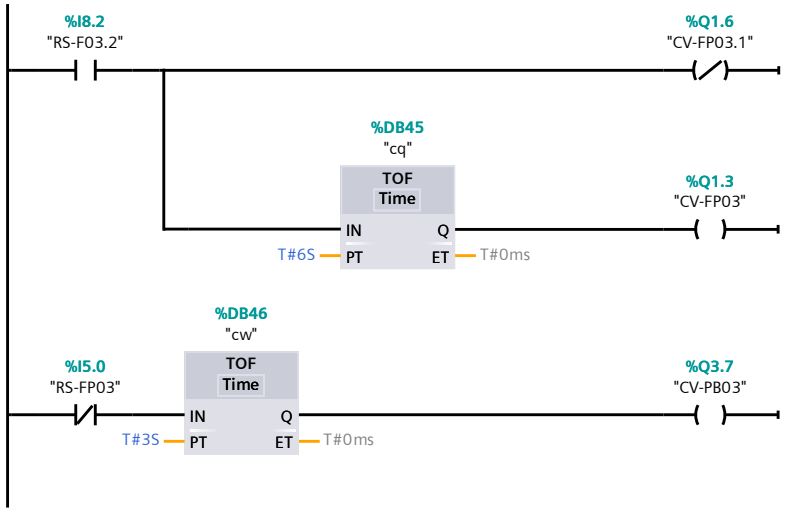

%DB41
"zu"
%Q7.1
"SG-P02"
TON
Time
IN
Q
T#5S
PT
ET
T#0ms
%DB19
"set2"
TOF
Time
IN
Q
T#2s
PT
ET
T#0ms
%Q5.2
"EU-P02"
R
%Q4.1
"ED-P02"
%DB42
"zi"
%Q4.7
"EML-P02"
%Q4.1
"ED-P02"
TON
Time
IN
Q
T#1.5s
PT
ET
T#0ms
%DB43
"zo"
TOF
Time
IN
Q
T#1.1s
PT
ET
T#0ms
%Q4.4
"EF-P02"
S

palletizer_3 [FC16]

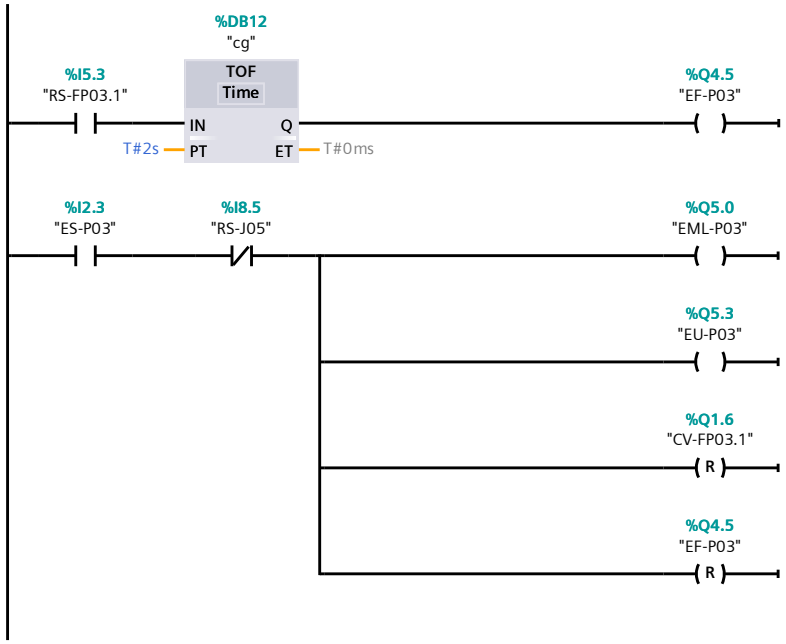
palletizer_3 Properties							
General							
Name	palletizer_3	**Number**	16	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
▼ Temp			
ISD	Bool		
Constant			
▼ Return			
palletizer_3	Void		

Network 1:

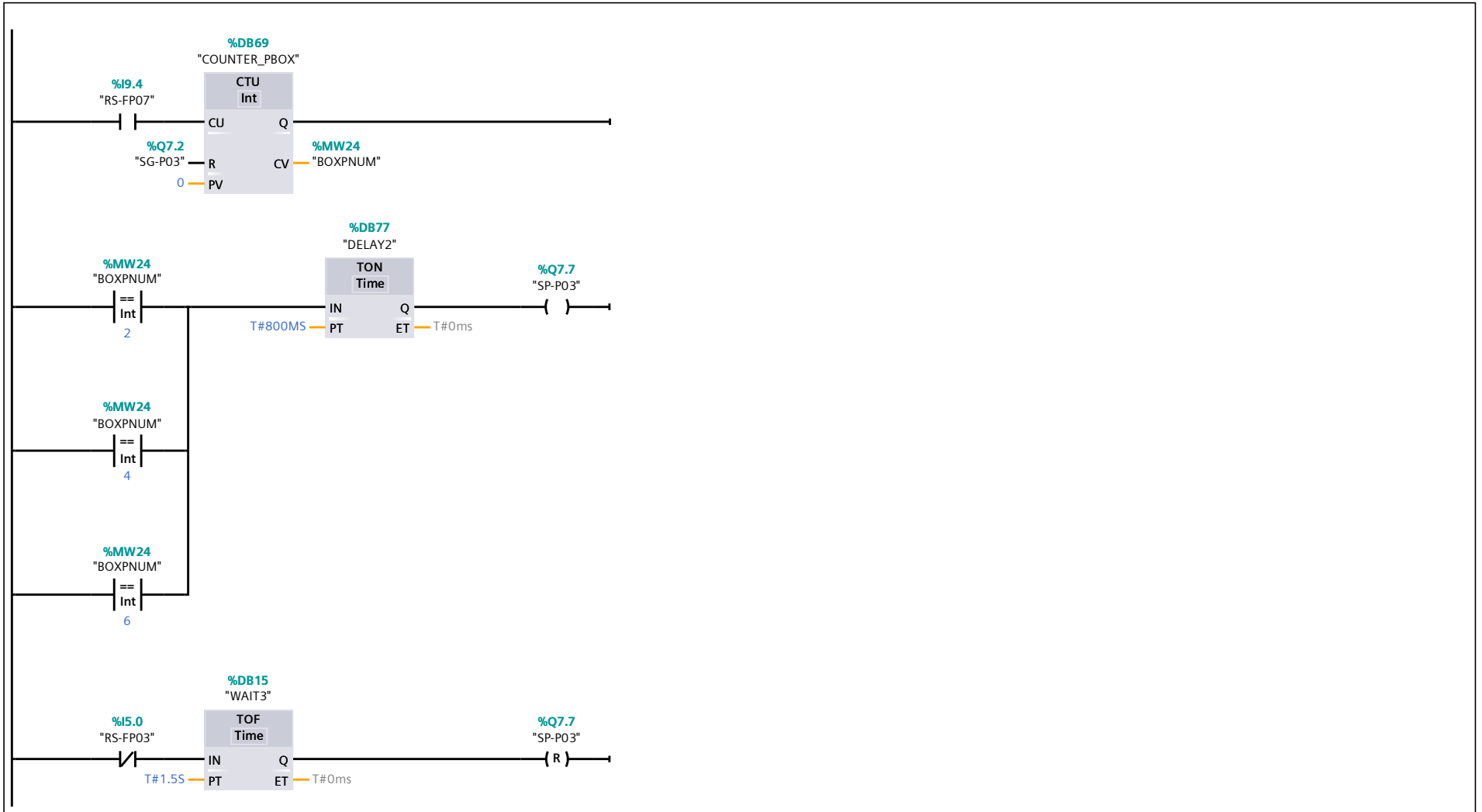

%I8.2
"RS-F03.2"
%Q1.6
"CV-FP03.1"
%DB45
"cq"
TOF
Time
IN
Q
T#6S
PT
ET
T#0ms
%Q1.3
"CV-FP03"
%DB46
"cw"
%I5.0
"RS-FP03"
TOF
Time
IN
Q
T#3S
PT
ET
T#0ms
%Q3.7
"CV-PB03"

Network 2:


%DB12
"cg"
%I5.3
"RS-FP03.1"
TOF
Time
IN
Q
T#2s
PT
ET
T#0ms
%Q4.5
"EF-P03"
%I2.3
"ES-P03"
%I8.5
"RS-J05"
%Q5.0
"EML-P03"
%Q5.3
"EU-P03"
%Q1.6
"CV-FP03.1"
R
%Q4.5
"EF-P03"
R

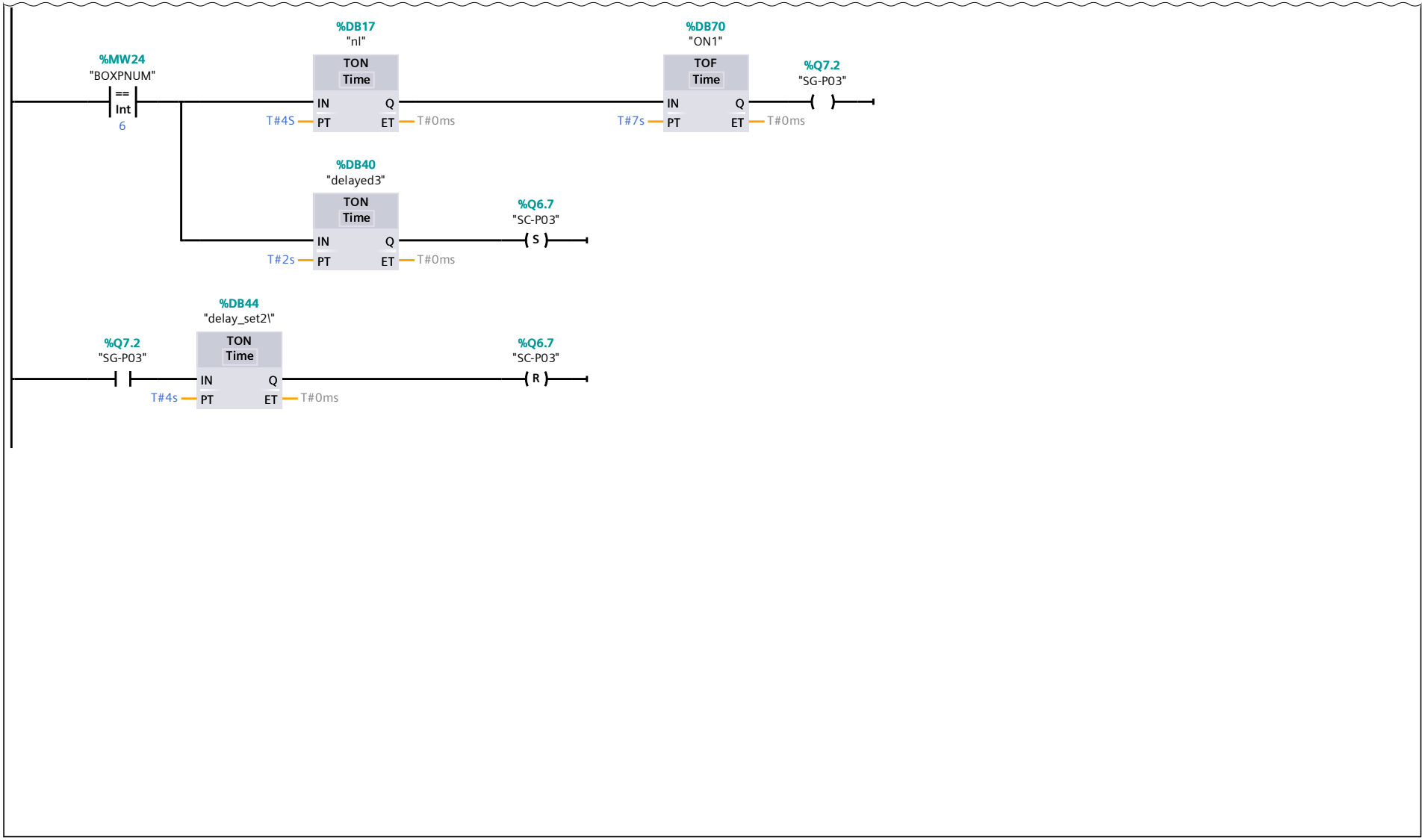
Network 3:

Network 3: (1.1 / 2.1)

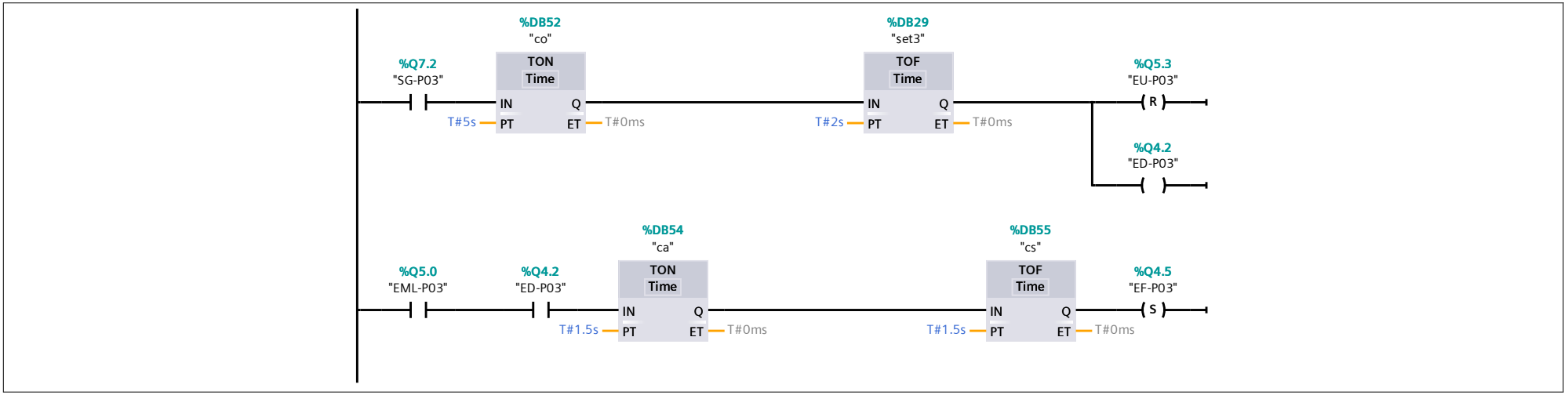


Network 3: (2.1 / 2.1)

1.1 (Page18 - 4)



Network 4:

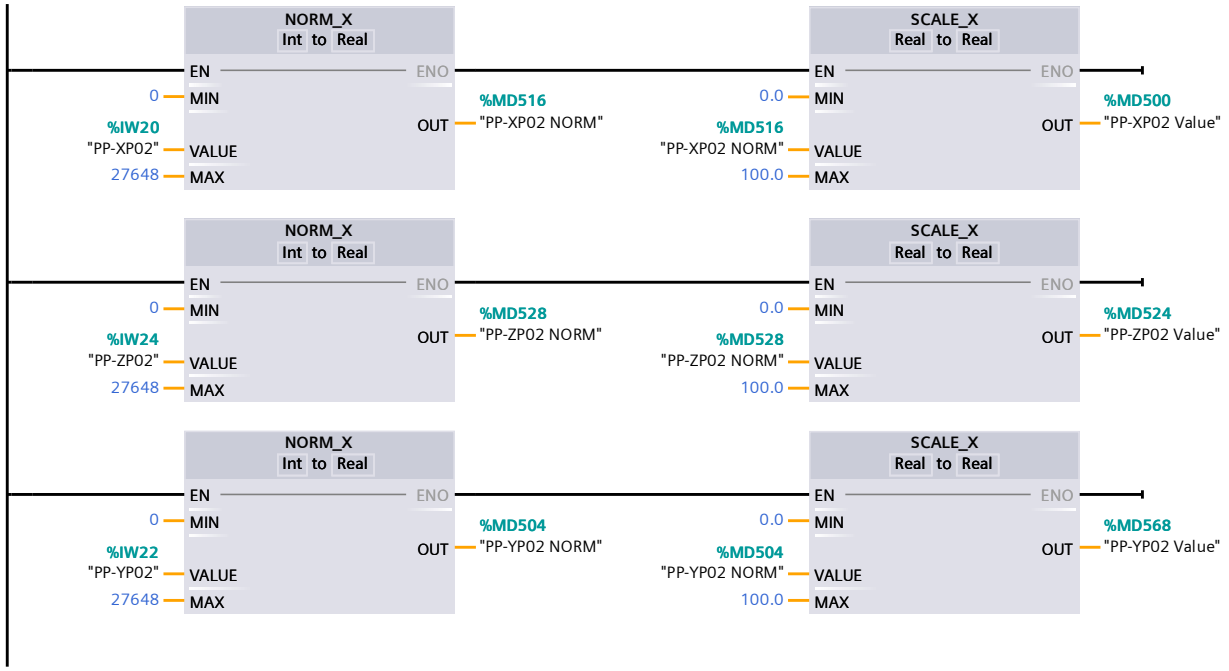

%DB52
"co"
%Q7.2
"SG-P03"
TON
Time
IN
Q
T#5s
PT
ET
T#0ms
%DB29
"set3"
TOF
Time
IN
Q
T#2s
PT
ET
T#0ms
%Q5.3
"EU-P03"
R
%Q4.2
"ED-P03"
%DB54
"ca"
%Q5.0
"EML-P03"
%Q4.2
"ED-P03"
TON
Time
IN
Q
T#1.5s
PT
ET
T#0ms
%DB55
"cs"
TOF
Time
IN
Q
T#1.5s
PT
ET
T#0ms
%Q4.5
"EF-P03"
S

pick and place XYZ [FC10]

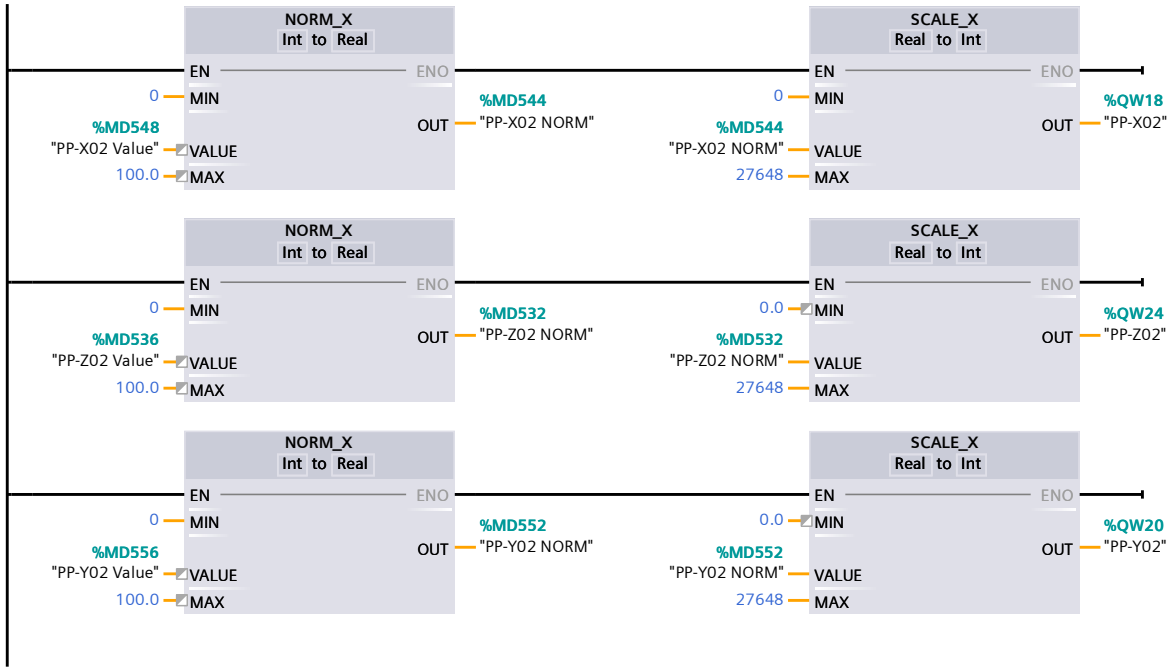
pick and place XYZ Properties							
General							
Name	pick and place XYZ	**Number**	10	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
pick and place XYZ	Void		

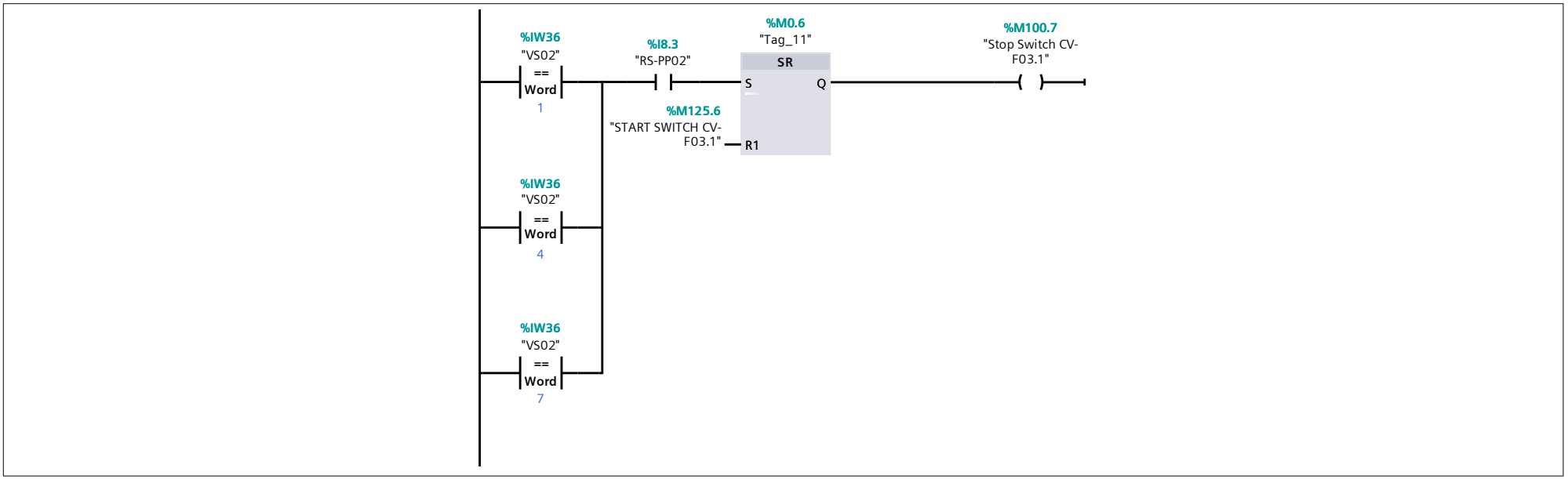
Network 1: Position Sensor scaleing


NORM_X
Int to Real
EN
ENO
0
MIN
%IW20
"PP-XP02"
VALUE
27648
MAX
OUT
%MD516
"PP-XP02 NORM"
SCALE_X
Real to Real
EN
ENO
0.0
MIN
%MD516
"PP-XP02 NORM"
VALUE
100.0
MAX
OUT
%MD500
"PP-XP02 Value"
NORM_X
Int to Real
EN
ENO
0
MIN
%IW24
"PP-ZP02"
VALUE
27648
MAX
OUT
%MD528
"PP-ZP02 NORM"
SCALE_X
Real to Real
EN
ENO
0.0
MIN
%MD528
"PP-ZP02 NORM"
VALUE
100.0
MAX
OUT
%MD524
"PP-ZP02 Value"
NORM_X
Int to Real
EN
ENO
0
MIN
%IW22
"PP-YP02"
VALUE
27648
MAX
OUT
%MD504
"PP-YP02 NORM"
SCALE_X
Real to Real
EN
ENO
0.0
MIN
%MD504
"PP-YP02 NORM"
VALUE
100.0
MAX
OUT
%MD568
"PP-YP02 Value"

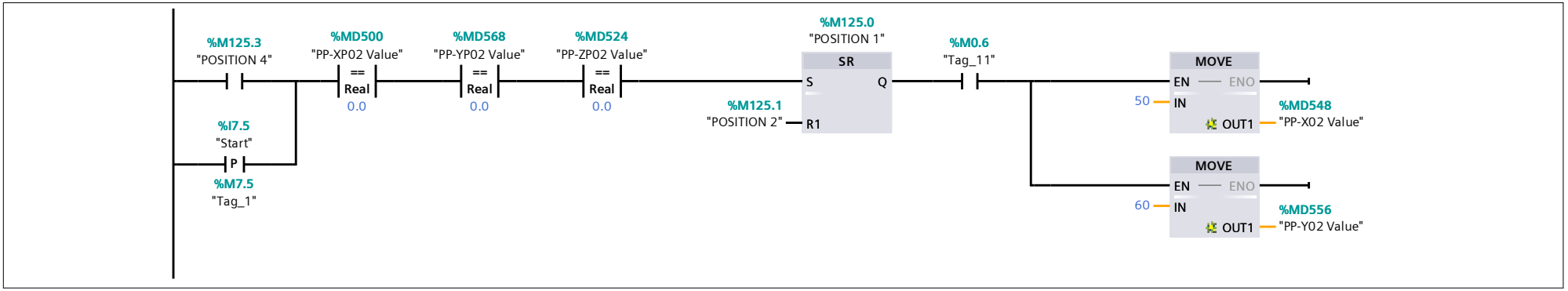
Network 2:


NORM_X
Int to Real
EN
ENO
0
MIN
%MD548
"PP-X02 Value"
VALUE
100.0
MAX
%MD544
"PP-X02 NORM"
OUT
SCALE_X
Real to Int
EN
ENO
0
MIN
%MD544
"PP-X02 NORM"
VALUE
27648
MAX
%QW18
"PP-X02"
OUT
NORM_X
Int to Real
EN
ENO
0
MIN
%MD536
"PP-Z02 Value"
VALUE
100.0
MAX
%MD532
"PP-Z02 NORM"
OUT
SCALE_X
Real to Int
EN
ENO
0.0
MIN
%MD532
"PP-Z02 NORM"
VALUE
27648
MAX
%QW24
"PP-Z02"
OUT
NORM_X
Int to Real
EN
ENO
0
MIN
%MD556
"PP-Y02 Value"
VALUE
100.0
MAX
%MD552
"PP-Y02 NORM"
OUT
SCALE_X
Real to Int
EN
ENO
0.0
MIN
%MD552
"PP-Y02 NORM"
VALUE
27648
MAX
%QW20
"PP-Y02"
OUT

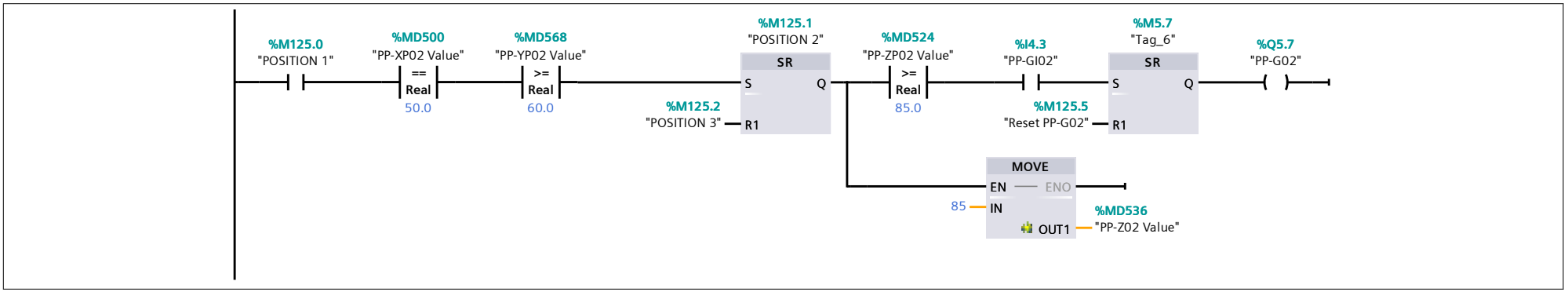
Network 3:



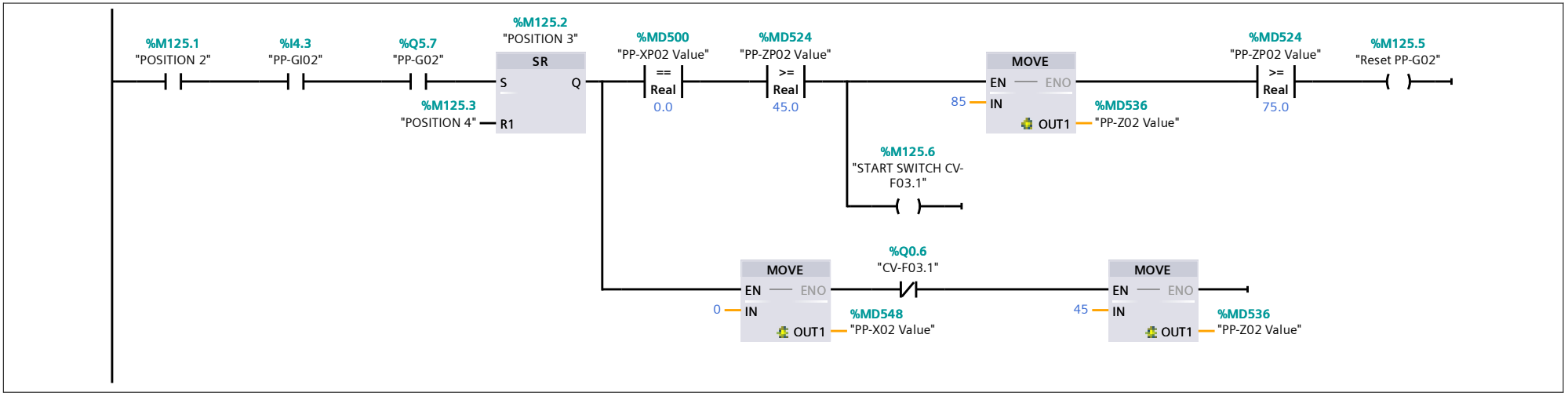
Network 4:



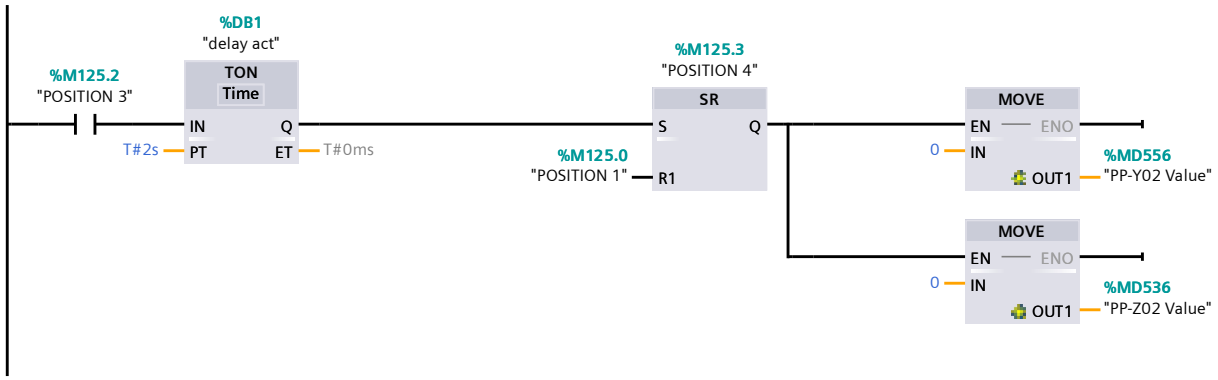
Network 5:



Network 6:



Network 7:

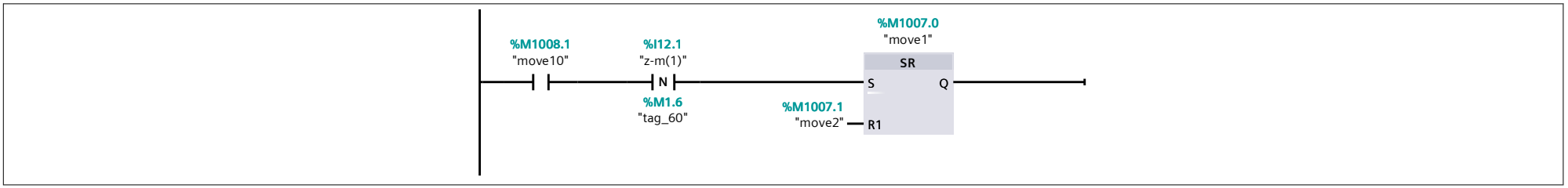

%DB1
"delay act"
%M125.2
"POSITION 3"
TON
Time
IN
Q
T#2s
PT
ET
T#0ms
%M125.3
"POSITION 4"
SR
S
Q
%M125.0
"POSITION 1"
R1
MOVE
EN
ENO
0
IN
OUT1
%MD556
"PP-Y02 Value"
MOVE
EN
ENO
0
IN
OUT1
%MD536
"PP-Z02 Value"

sensors load2 [FC11]

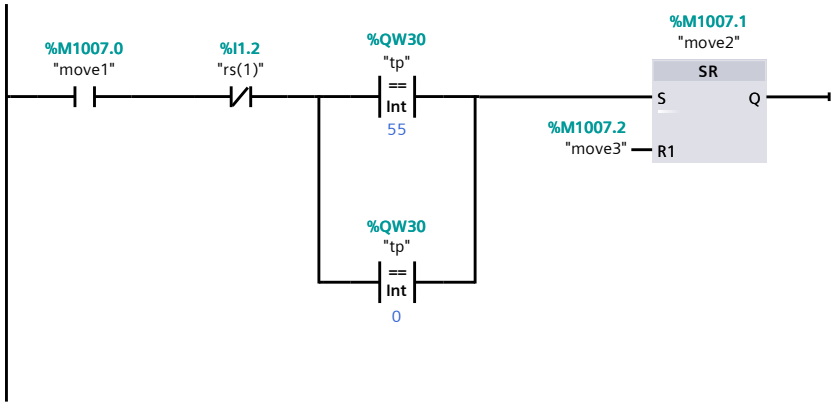
sensors load2 Properties							
General							
Name	sensors load2	**Number**	11	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
sensors load2	Void		

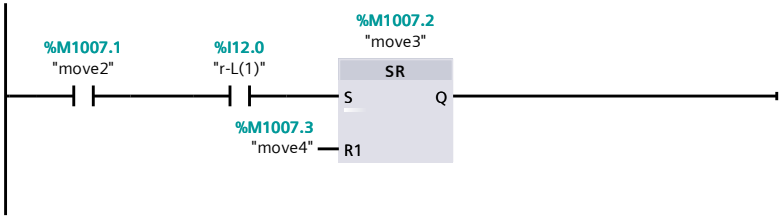
Network 1:



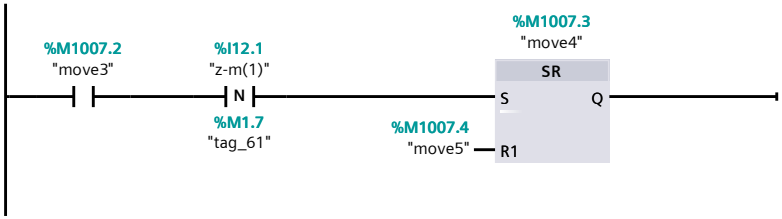
Network 2:



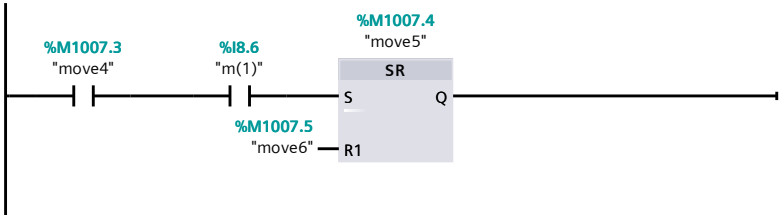
Network 3:



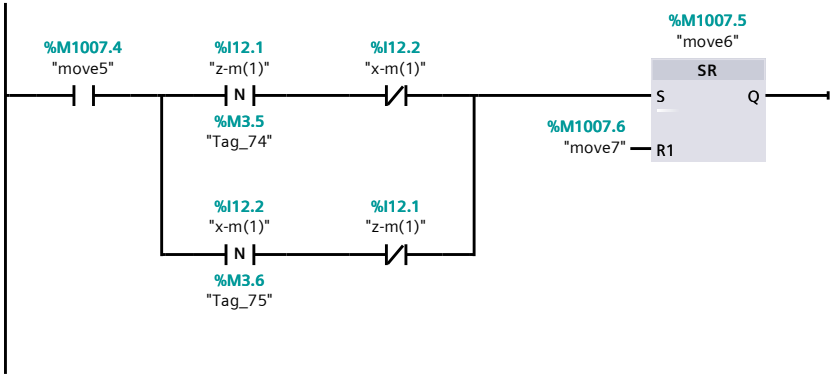
Network 4:



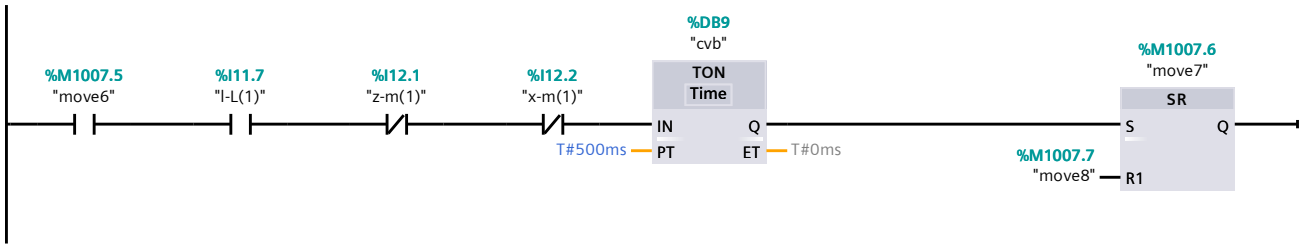
Network 5:



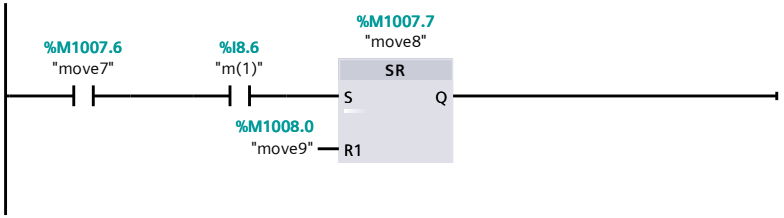
Network 6:



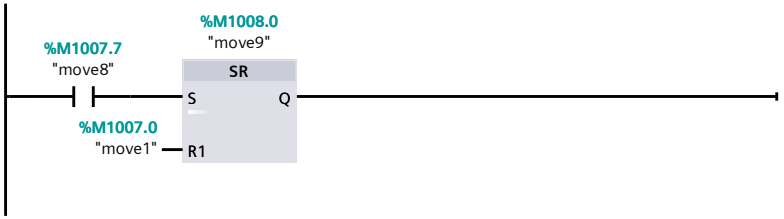
Network 7:



Network 8:



Network 9:

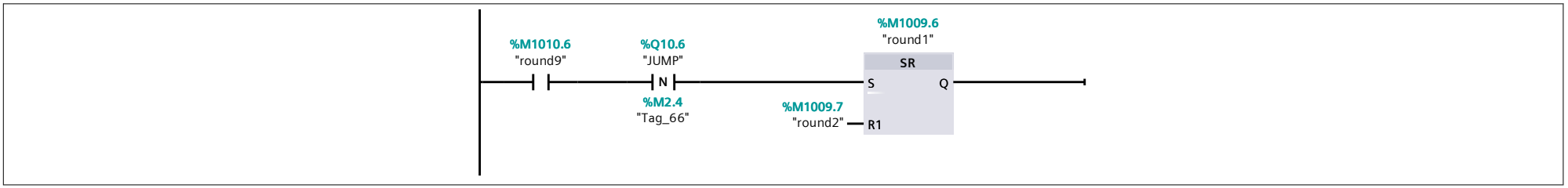


sensorsload3 [FC29]

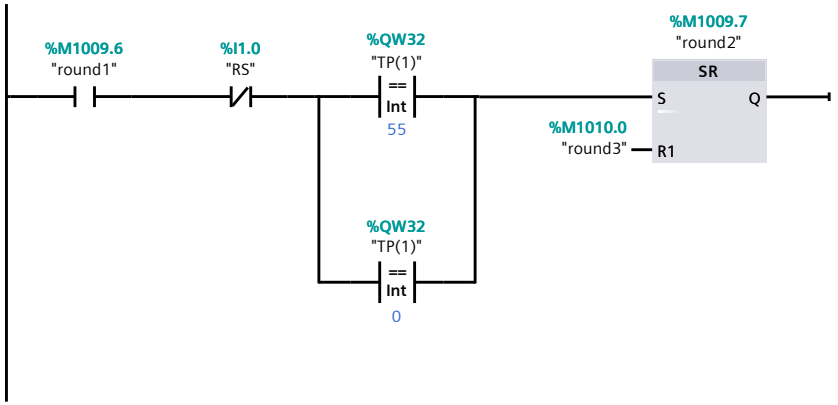
sensorsload3 Properties							
General							
Name	sensorsload3	**Number**	29	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
sensorsload3	Void		

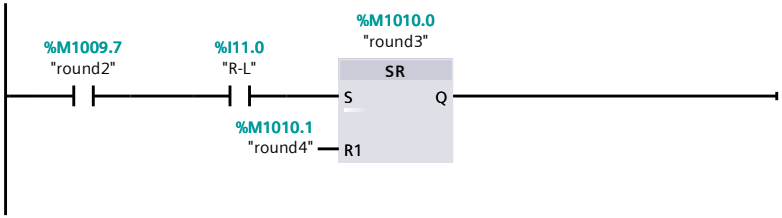
Network 1:



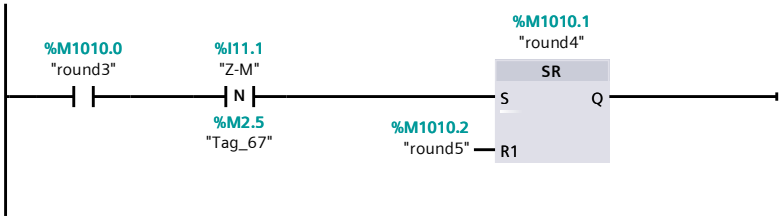
Network 2:



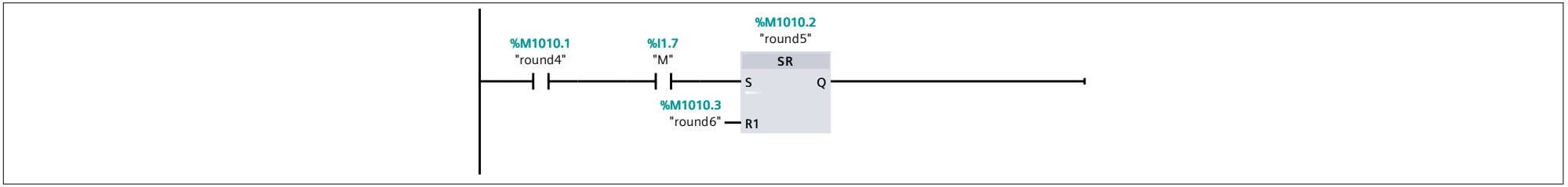
Network 3:



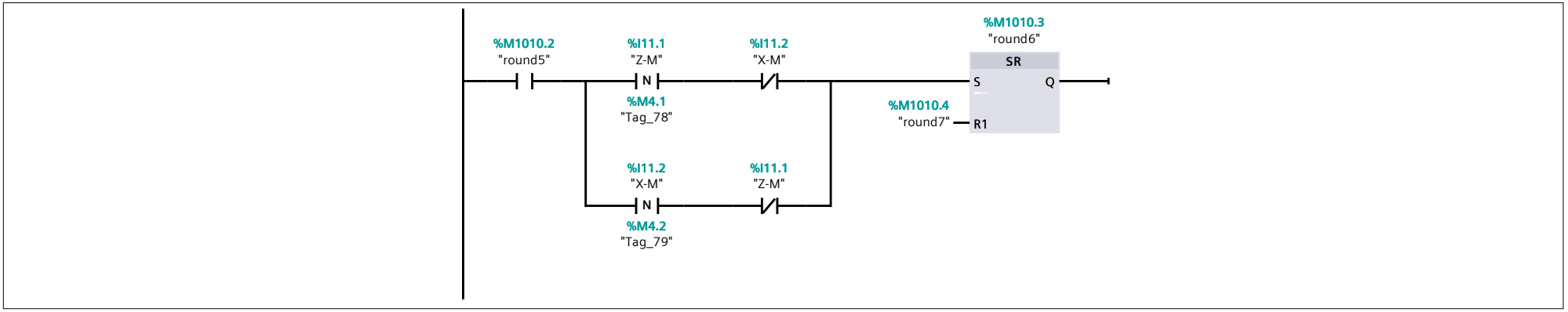
Network 4:



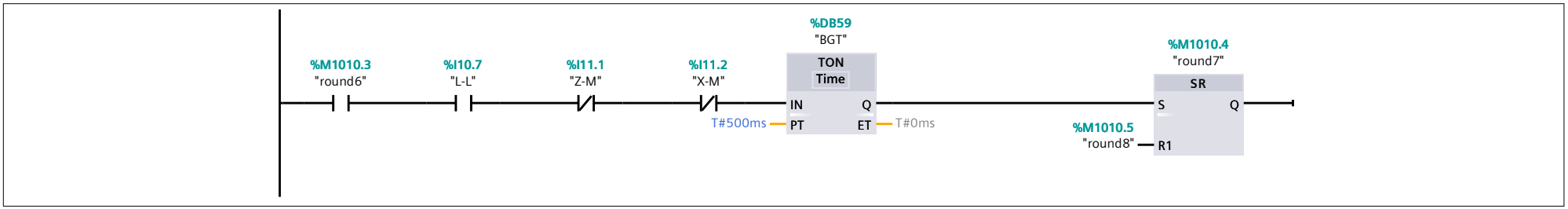
Network 5:



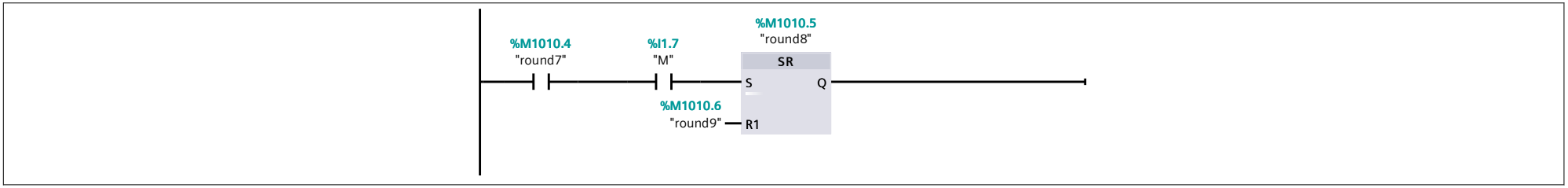
Network 6:



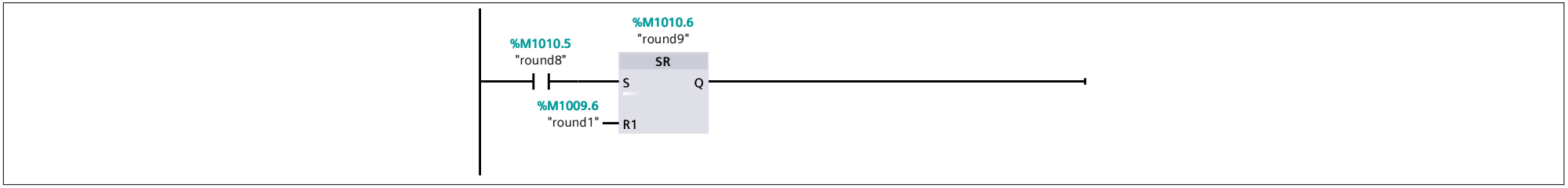
Network 7:



Network 8:



Network 9:

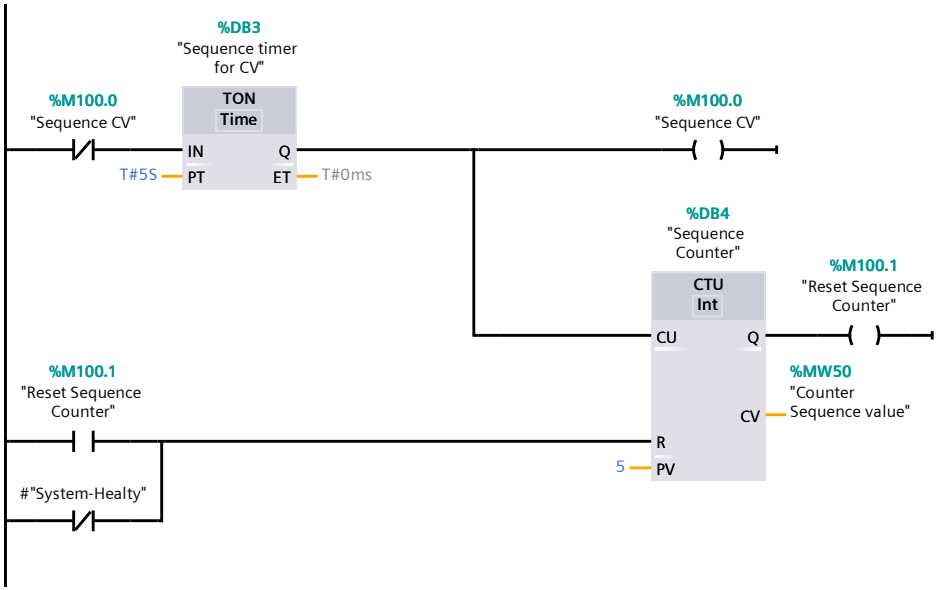


Sequence Start [FC4]

Sequence Start Properties							
General							
Name	Sequence Start	**Number**	4	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
▼ Input			
System-Healty	Bool		
Output			
InOut			
Temp			
Constant			
▼ Return			
Sequence Start	Void		

Network 1:

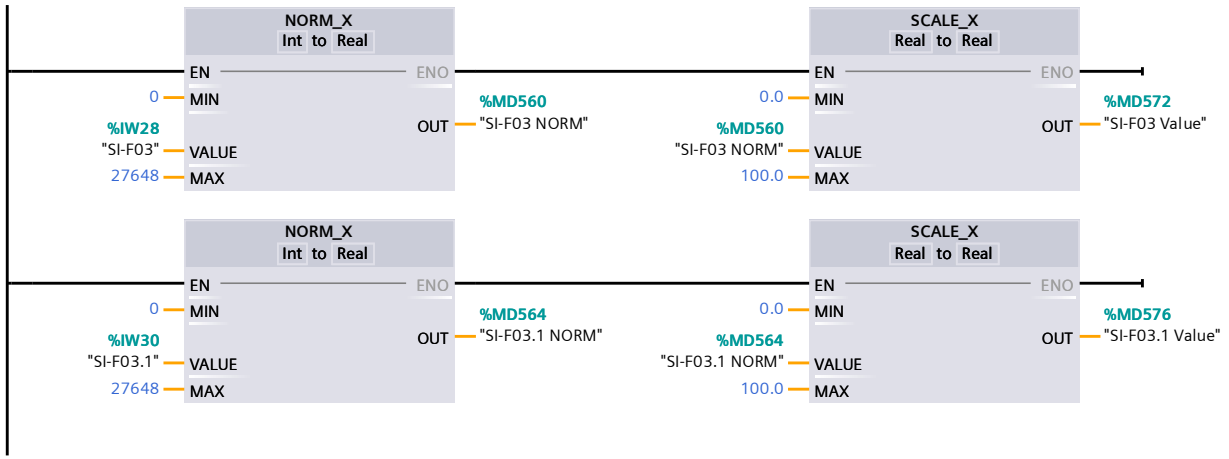

%DB3
"Sequence timer for CV"
%M100.0
"Sequence CV"
TON
Time
IN
Q
T#5S
PT
ET
T#0ms
%M100.0
"Sequence CV"
%DB4
"Sequence Counter"
CTU
Int
CU
Q
%M100.1
"Reset Sequence Counter"
%M100.1
"Reset Sequence Counter"
%MW50
"Counter Sequence value"
CV
R
5
PV
#"System-Healty"

Size inducator system [FC6]

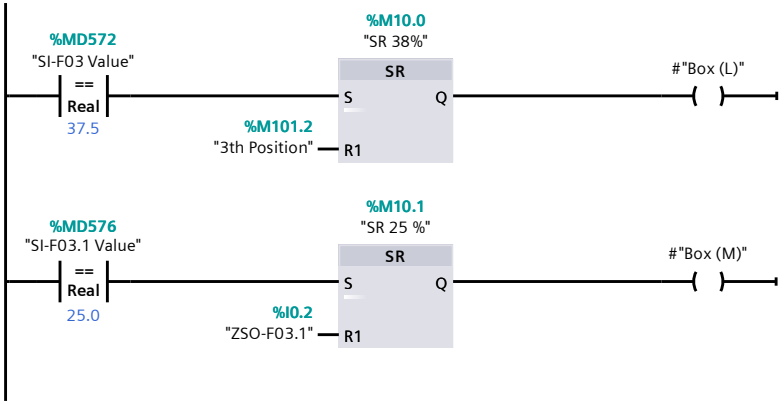
Size inducator system Properties							
General							
Name	Size inducator system	**Number**	6	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
▼ Input			
PP-GI01	Bool		
PP-GI02	Bool		
▼ Output			
Box (L)	Bool		
Box (M)	Bool		
InOut			
Temp			
Constant			
▼ Return			
Size inducator system	Void		

Network 1: Size Box Scaling



Network 2: Box (L) Memory

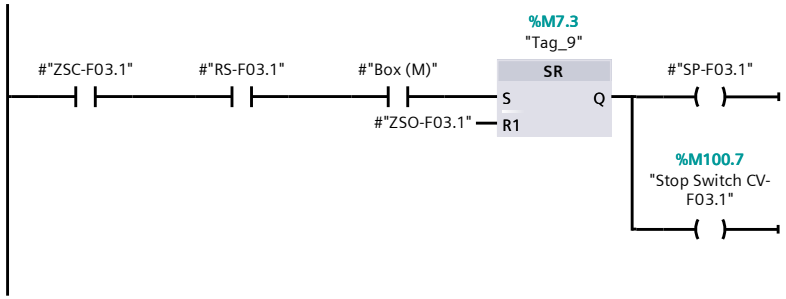


Slide Pusher 03.1 [FC8]

Slide Pusher 03.1 Properties							
General							
Name	Slide Pusher 03.1	**Number**	8	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
▼ Input			
ZSC-F03.1	Bool		
ZSO-F03.1	Bool		
RS-F03.1	Bool		
Box (M)	Bool		
▼ Output			
SP-F03.1	Bool		
InOut			
Temp			
Constant			
▼ Return			
Slide Pusher 03.1	Void		

Network 1:

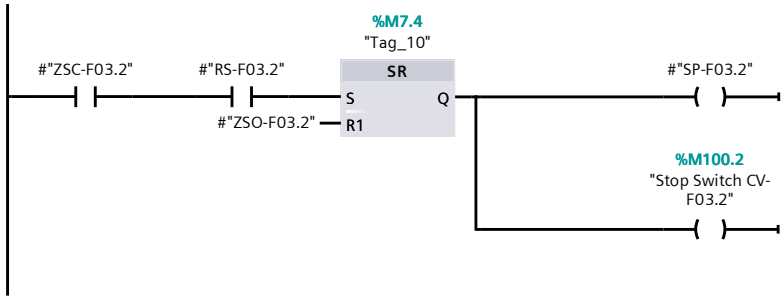

%M7.3
"Tag_9"
#"ZSC-F03.1"
#"RS-F03.1"
#"Box (M)"
SR
S
Q
#"ZSO-F03.1"
R1
#"SP-F03.1"
%M100.7
"Stop Switch CV-
F03.1"

Slide Pusher 03.2 [FC9]

Slide Pusher 03.2 Properties							
General							
Name	Slide Pusher 03.2	**Number**	9	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
▼ Input			
ZSC-F03.2	Bool		
ZSO-F03.2	Bool		
RS-F03.2	Bool		
▼ Output			
SP-F03.2	Bool		
InOut			
Temp			
Constant			
▼ Return			
Slide Pusher 03.2	Void		

Network 1:

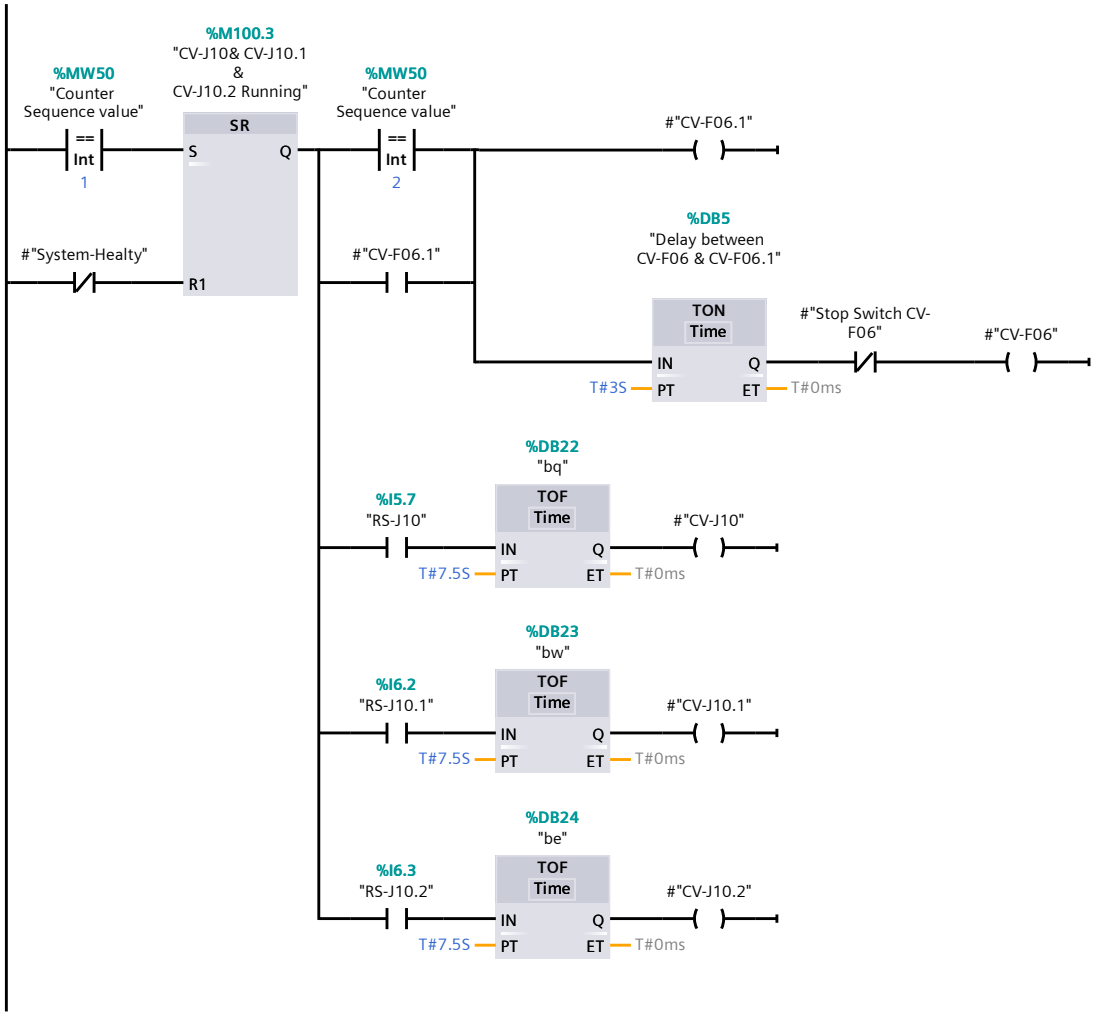

%M7.4
"Tag_10"
#"ZSC-F03.2"
#"RS-F03.2"
SR
S
Q
#"ZSO-F03.2"
R1
#"SP-F03.2"
%M100.2
"Stop Switch CV-
F03.2"

Sorting Station Sequence [FC3]

Sorting Station Sequence Properties							
General							
Name	Sorting Station Sequence	**Number**	3	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
▼ Input			
System-Healty	Bool		
RS-J11	Bool		
RS-J11.1	Bool		
RS-J11.2	Bool		
Stop Switch CV-F06	Bool		
CV-J11	Bool		
CV-J11.1	Bool		
CV-J11.2	Bool		
▼ Output			
CV-J10	Bool		
CV-J10.1	Bool		
CV-J10.2	Bool		
CV-F06	Bool		
▼ InOut			
CV-F06.1	Bool		
Temp			
Constant			
▼ Return			
Sorting Station Sequence	Void		

Network 1: Sorting Station conveyor



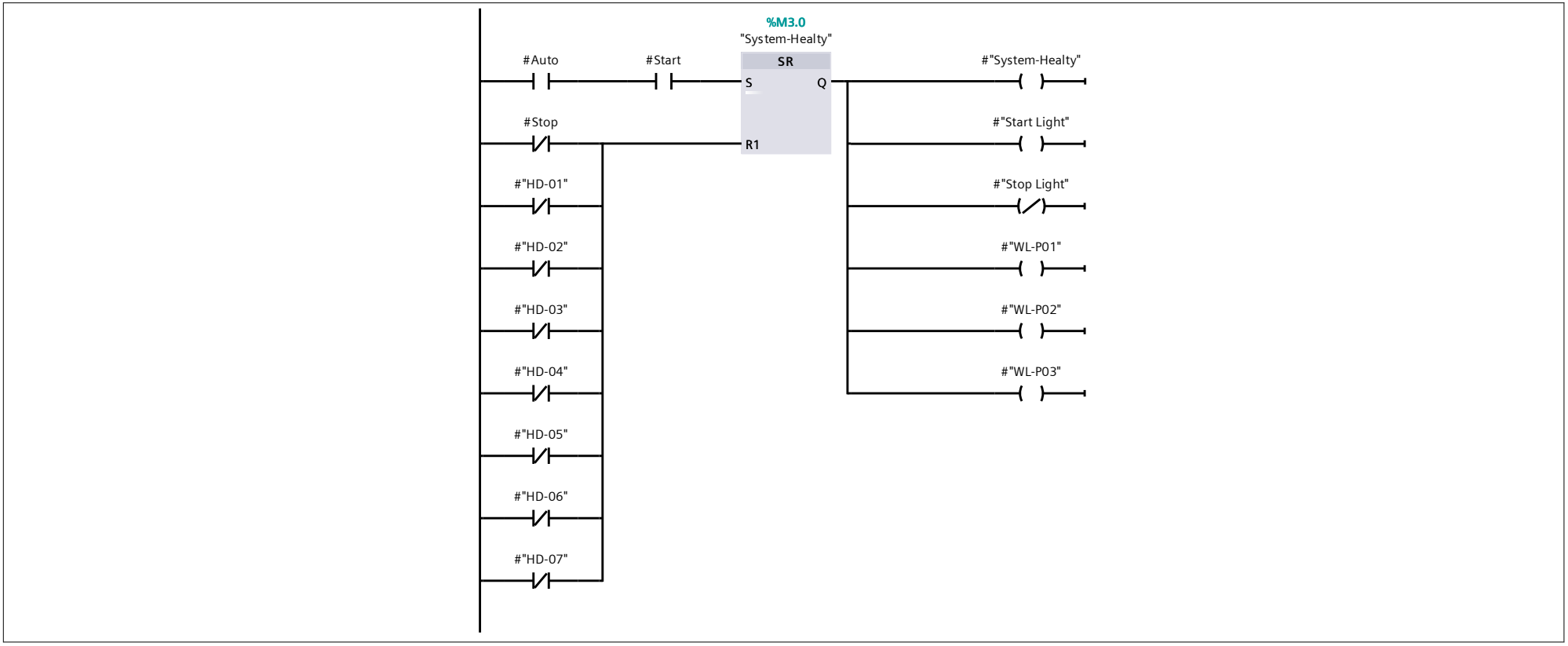
START-UP [FC7]

START-UP Properties							
General							
Name	START-UP	**Number**	7	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
▼ Input			
Auto	Bool		
Start	Bool		
Stop	Bool		
HD-01	Bool		
HD-02	Bool		
HD-03	Bool		
HD-04	Bool		
HD-05	Bool		
HD-06	Bool		
HD-07	Bool		
▼ Output			
System-Healty	Bool		
Start Light	Bool		
Stop Light	Bool		
WL-P01	Bool		
WL-P02	Bool		
WL-P03	Bool		
InOut			
Temp			
Constant			
▼ Return			

Name	Data type	Default value	Comment
START-UP	Void		

Network 1:

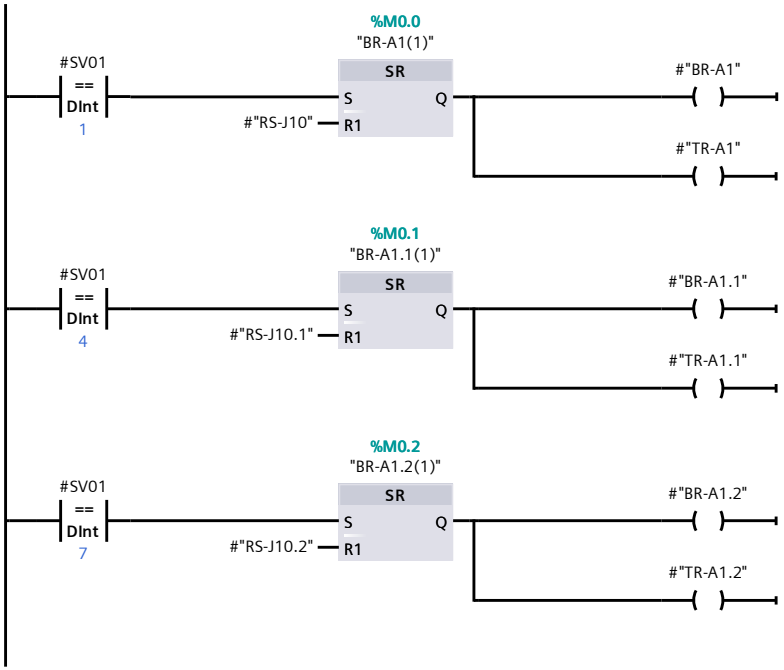


Station Sorting system [FC5]

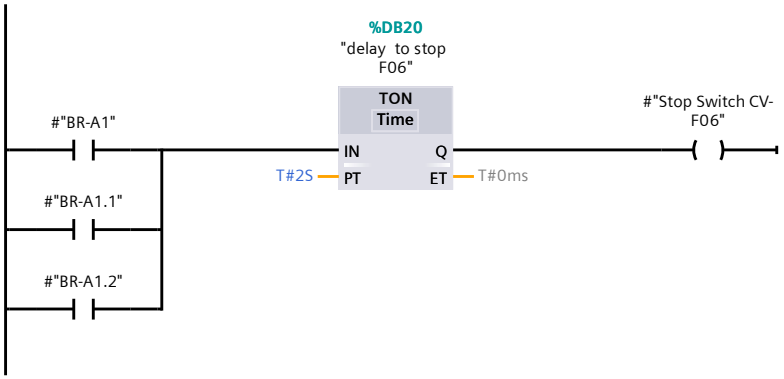
Station Sorting system Properties							
General							
Name	Station Sorting system	**Number**	5	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
▼ Input			
SV01	DInt		
RS-J10	Bool		
RS-J10.1	Bool		
RS-J10.2	Bool		
▼ Output			
TR-A1	Bool		
TR-A1.1	Bool		
TR-A1.2	Bool		
Stop Switch CV-F06	Bool		
▼ InOut			
BR-A1	Bool		
BR-A1.1	Bool		
BR-A1.2	Bool		
Temp			
Constant			
▼ Return			
Station Sorting system	Void		

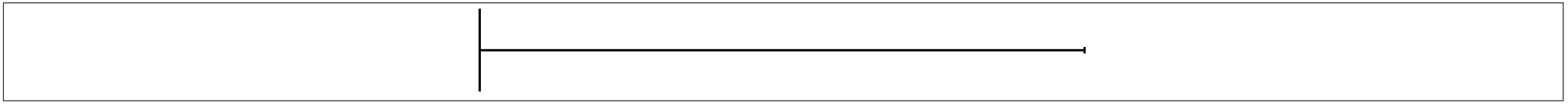
Network 1:


%M0.0
"BR-A1(1)"
#SV01
==
DInt
1
SR
S
Q
#"RS-J10"
R1
#"BR-A1"
#"TR-A1"
%M0.1
"BR-A1.1(1)"
#SV01
==
DInt
4
SR
S
Q
#"RS-J10.1"
R1
#"BR-A1.1"
#"TR-A1.1"
%M0.2
"BR-A1.2(1)"
#SV01
==
DInt
7
SR
S
Q
#"RS-J10.2"
R1
#"BR-A1.2"
#"TR-A1.2"

Network 2:


%DB20
"delay to stop
F06"
TON
Time
#"BR-A1"
IN
Q
#"Stop Switch CV-
F06"
T#2S
PT
ET
T#0ms
#"BR-A1.1"
#"BR-A1.2"

Network 3:

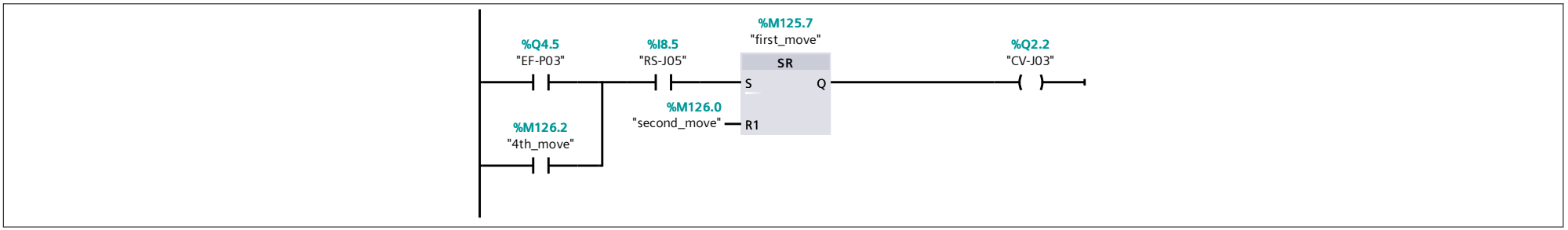


turntable [FC12]

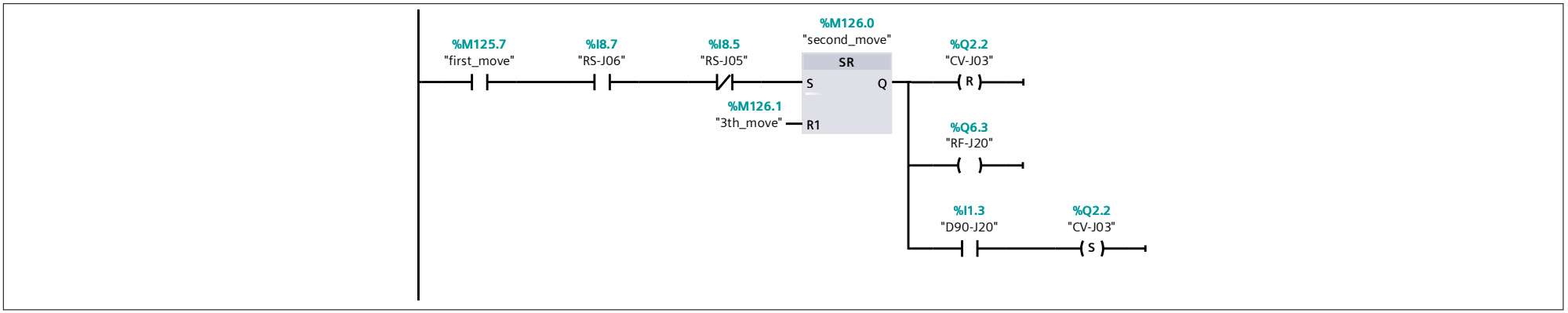
turntable Properties							
General							
Name	turntable	**Number**	12	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
turntable	Void		

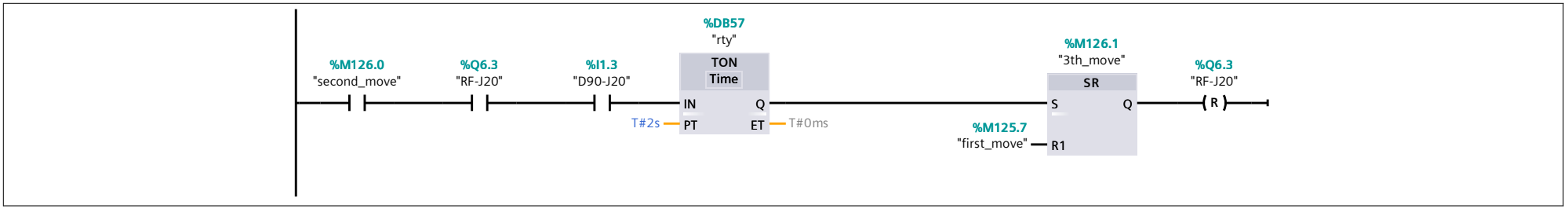
Network 1:



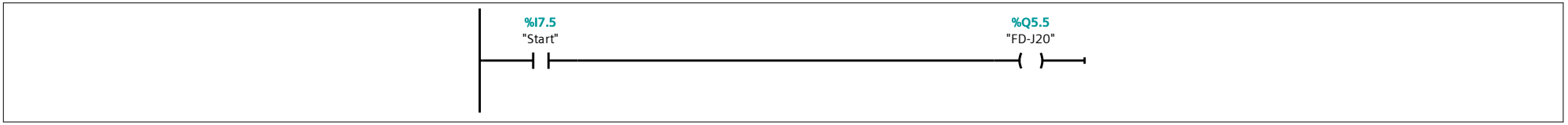
Network 2:



Network 3:



Network 4:

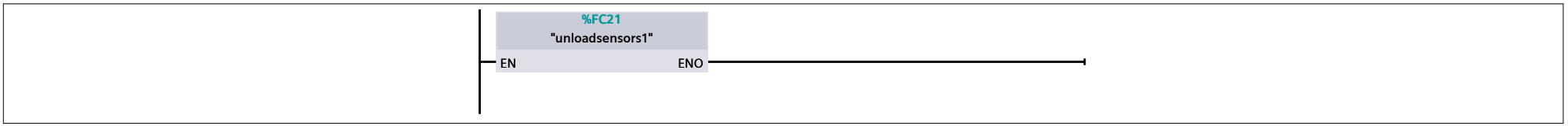


unload_1 [FC23]

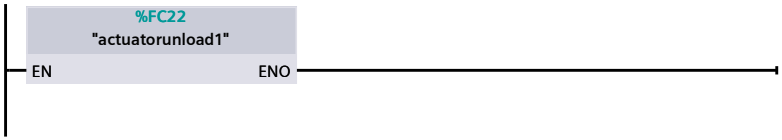
unload_1 Properties							
General							
Name	unload_1	**Number**	23	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
unload_1	Void		

Network 1:



Network 2:

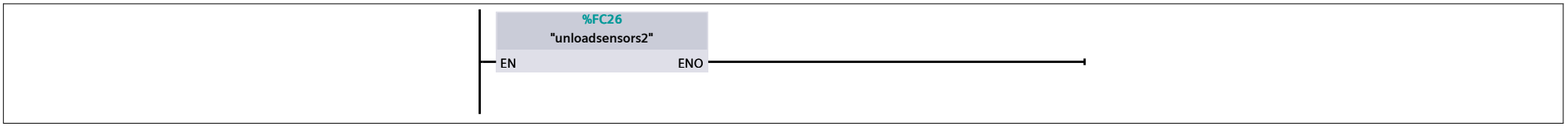

%FC22
"actuatorunload1"
EN
ENO

unload_2 [FC28]

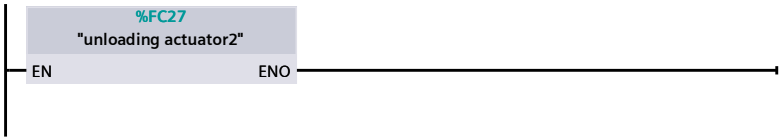
unload_2 Properties							
General							
Name	unload_2	**Number**	28	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
unload_2	Void		

Network 1:



Network 2:

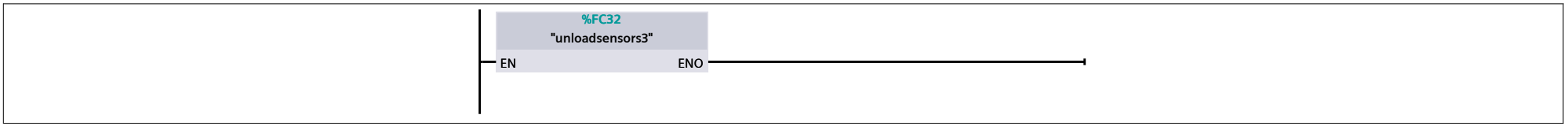

%FC27
"unloading actuator2"
EN
ENO

unload_3 [FC34]

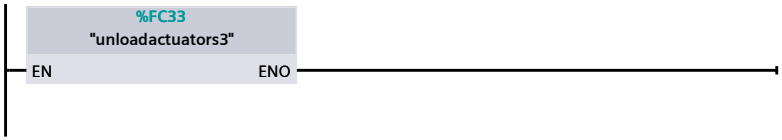
unload_3 Properties							
General							
Name	unload_3	**Number**	34	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
unload_3	Void		

Network 1:



Network 2:

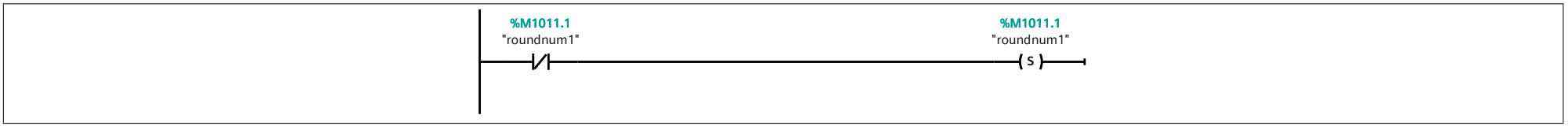

%FC33
"unloadactuators3"
EN
ENO

unloadactuators3 [FC33]

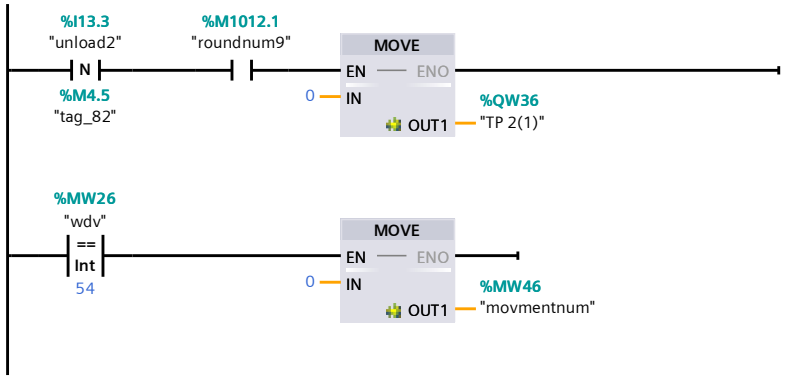
unloadactuators3 Properties							
General							
Name	unloadactuators3	**Number**	33	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
unloadactuators3	Void		

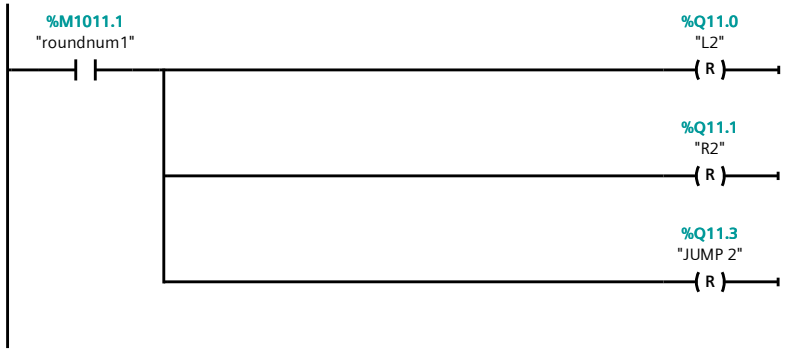
Network 1:



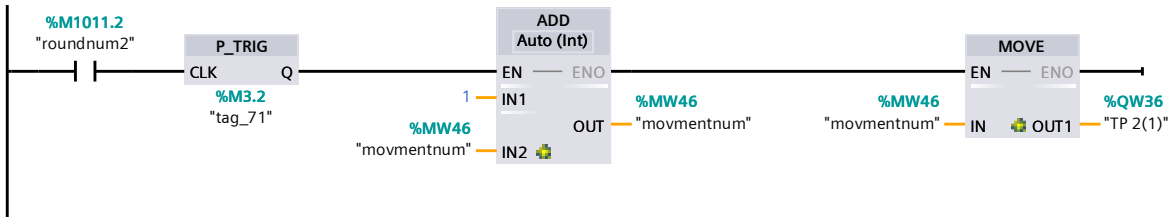
Network 2:



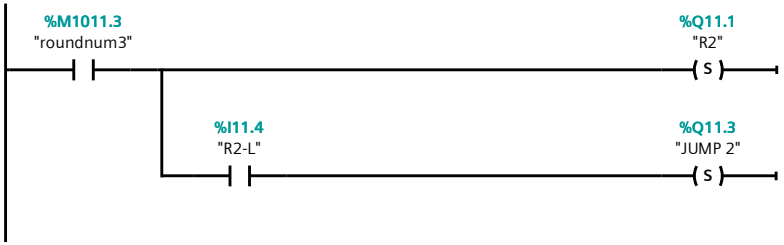
Network 3:



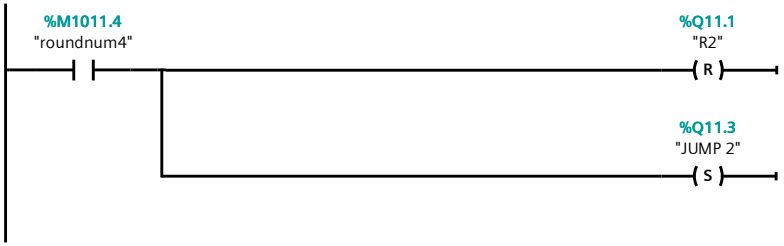
Network 4:



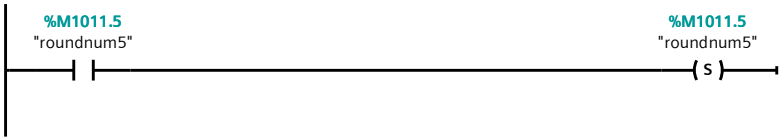
Network 5:



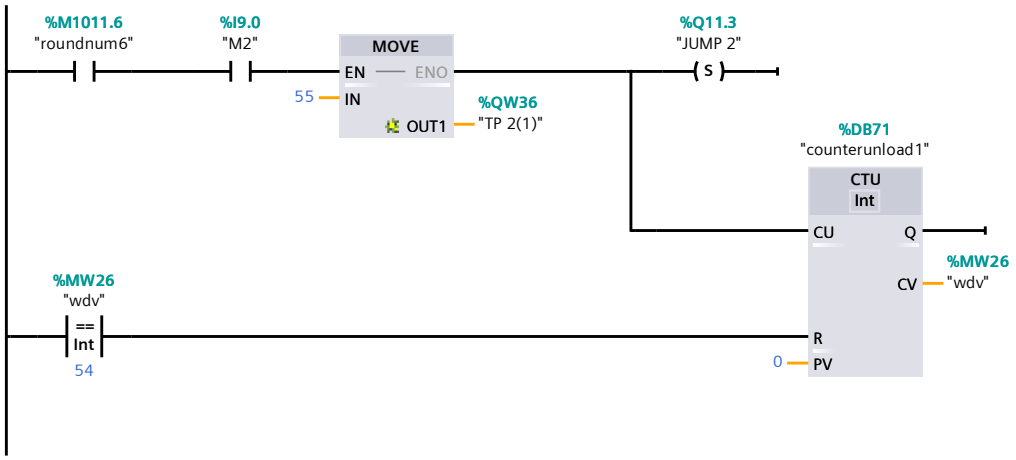
Network 6:



Network 7:



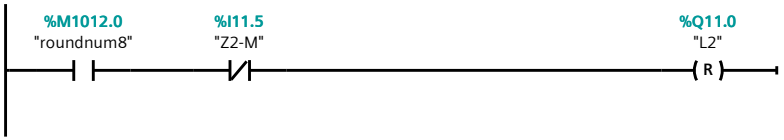
Network 8:



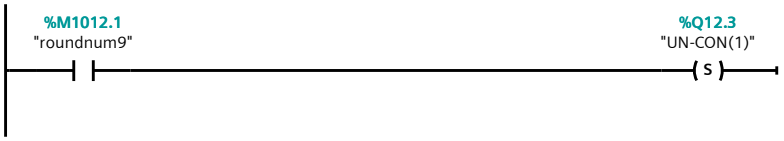
Network 9:



Network 10:



Network 11:

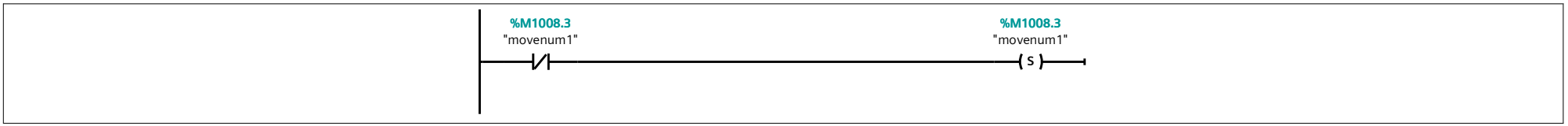


unloading actuator2 [FC27]

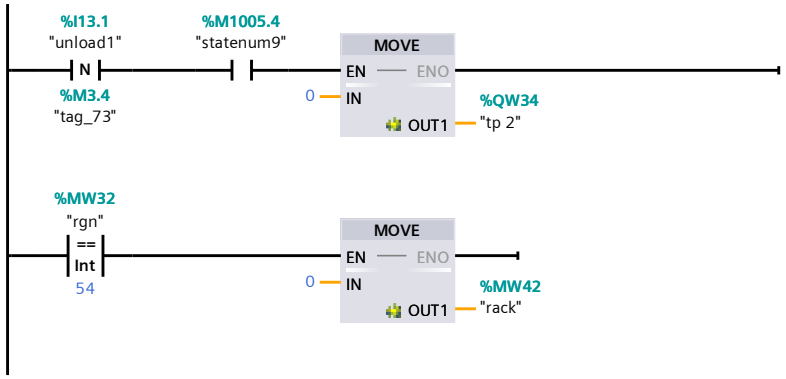
unloading actuator2 Properties							
General							
Name	unloading actuator2	**Number**	27	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
unloading actuator2	Void		

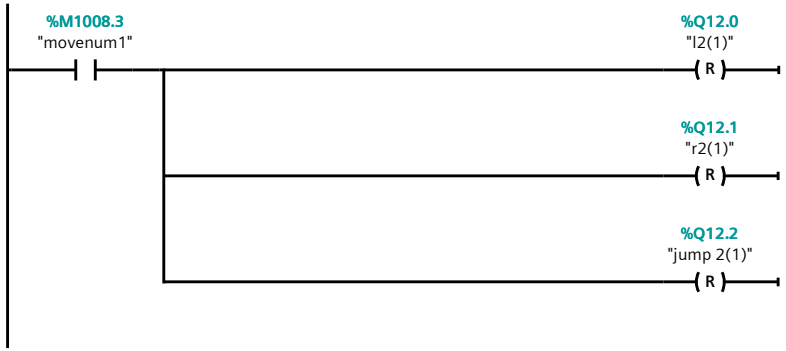
Network 1:



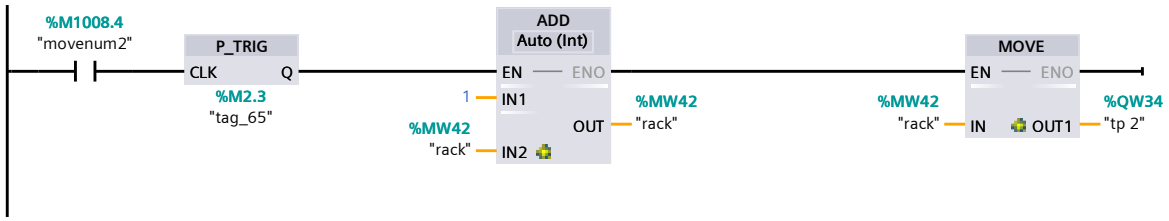
Network 2:



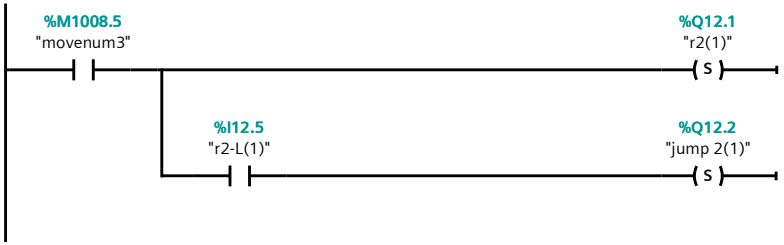
Network 3:



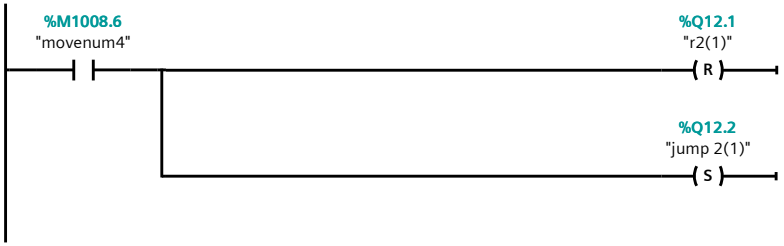
Network 4:



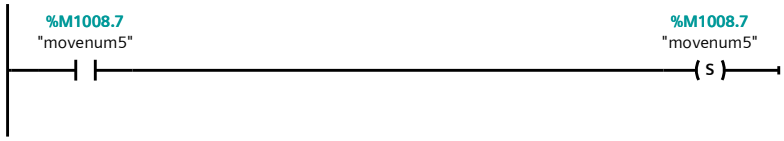
Network 5:



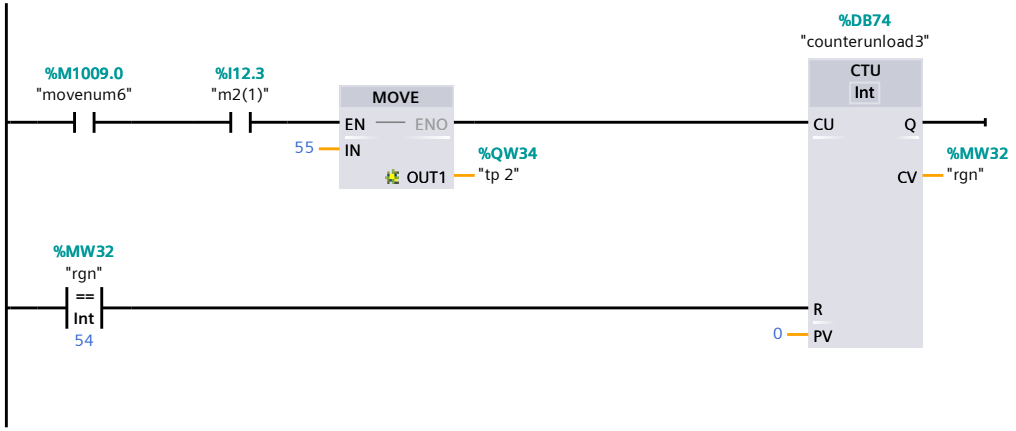
Network 6:



Network 7:



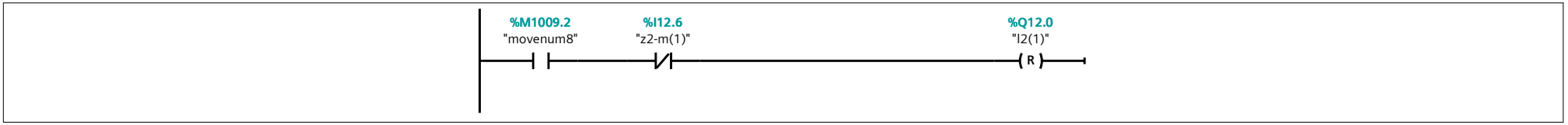
Network 8:



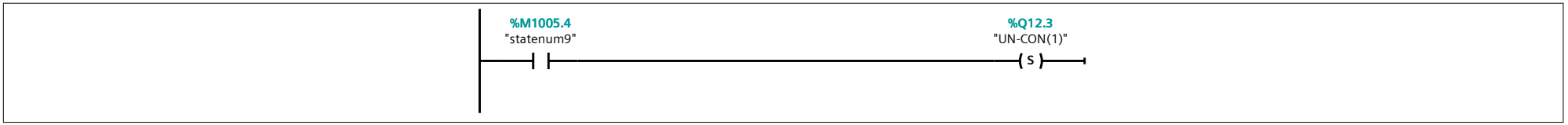
Network 9:



Network 10:



Network 11:

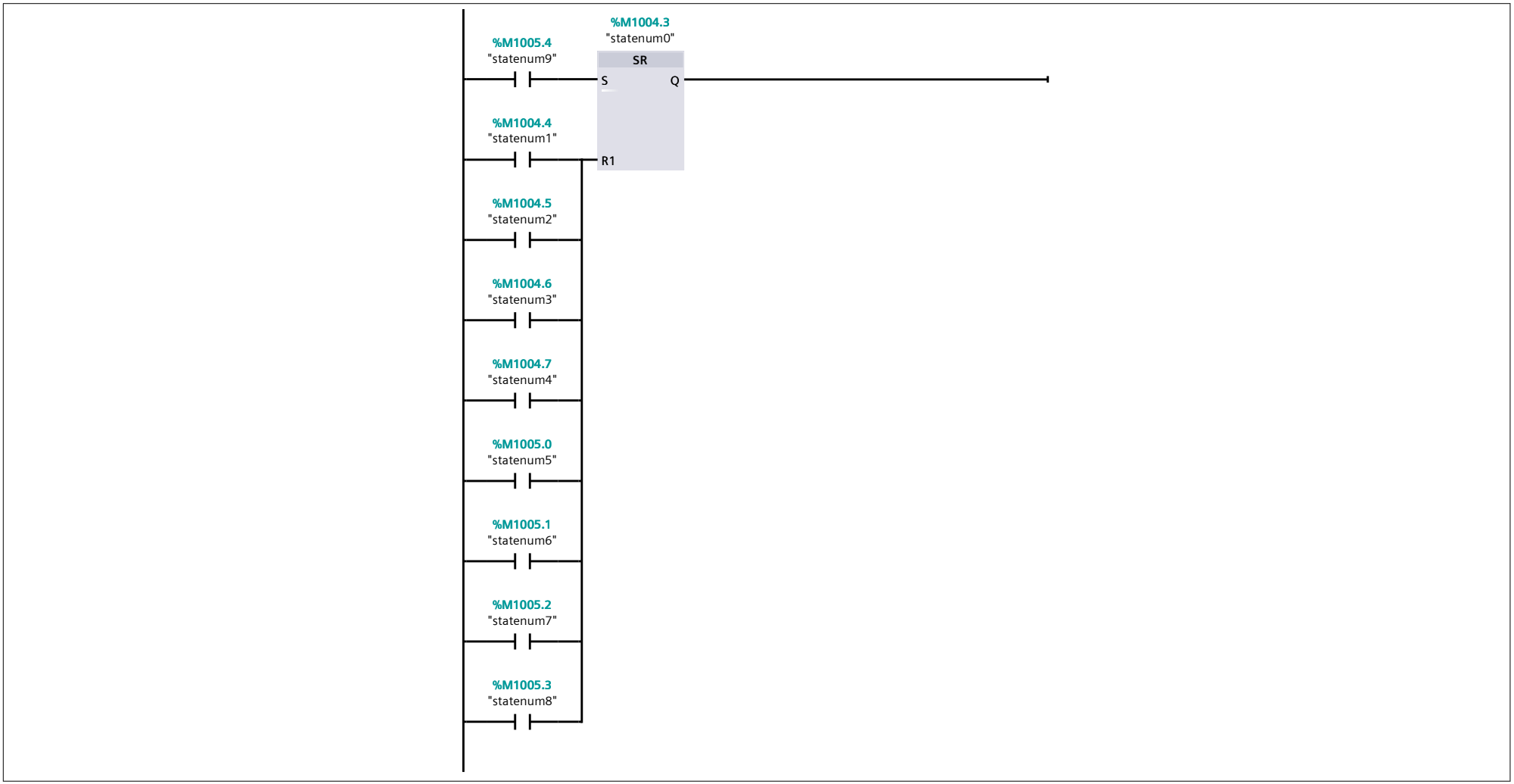


unloadsensors1 [FC21]

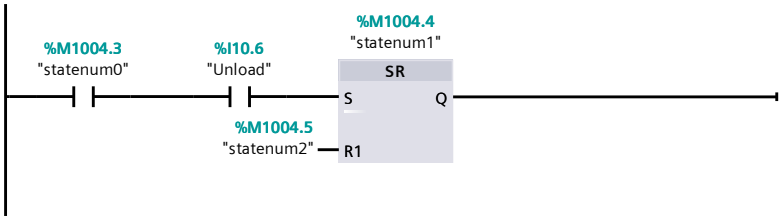
unloadsensors1 Properties							
General							
Name	unloadsensors1	**Number**	21	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
▼ Temp			
statenum9	Bool		
statenum0	Bool		
statenum1	Bool		
statenum2	Bool		
statenum3	Bool		
statenum4	Bool		
tag_37	Bool		
Constant			
▼ Return			
unloadsensors1	Void		

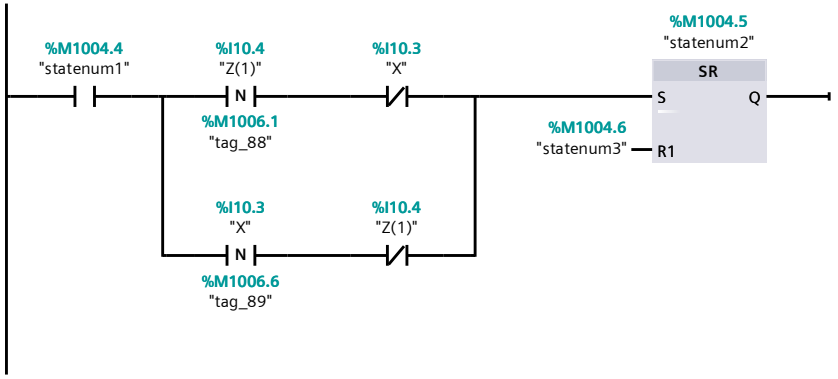
Network 1:


%M1004.3
"statenum0"
%M1005.4
"statenum9"
SR
S
Q
%M1004.4
"statenum1"
R1
%M1004.5
"statenum2"
%M1004.6
"statenum3"
%M1004.7
"statenum4"
%M1005.0
"statenum5"
%M1005.1
"statenum6"
%M1005.2
"statenum7"
%M1005.3
"statenum8"

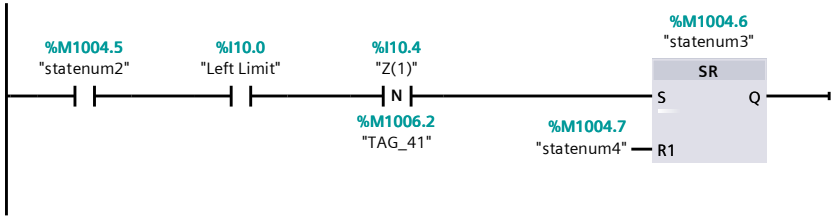
Network 2:



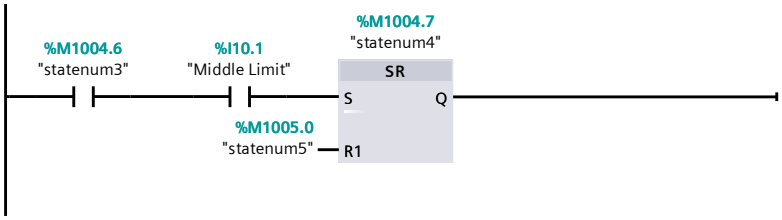
Network 3:



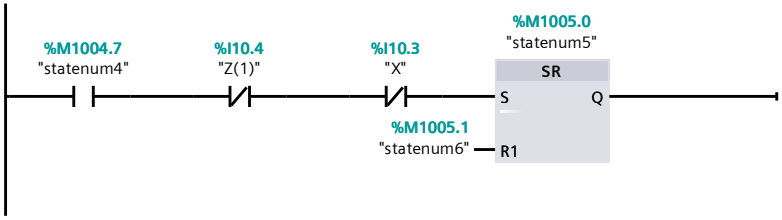
Network 4:



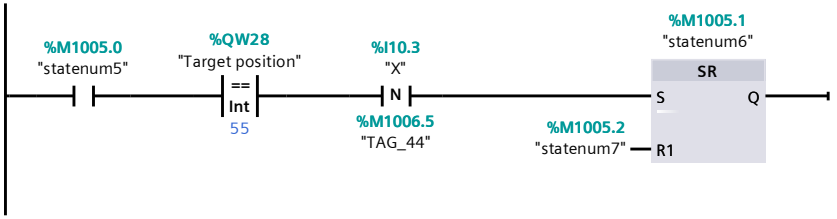
Network 5:



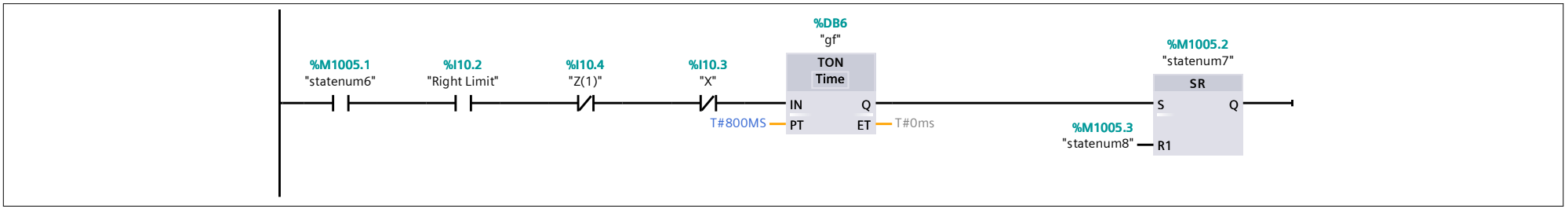
Network 6:



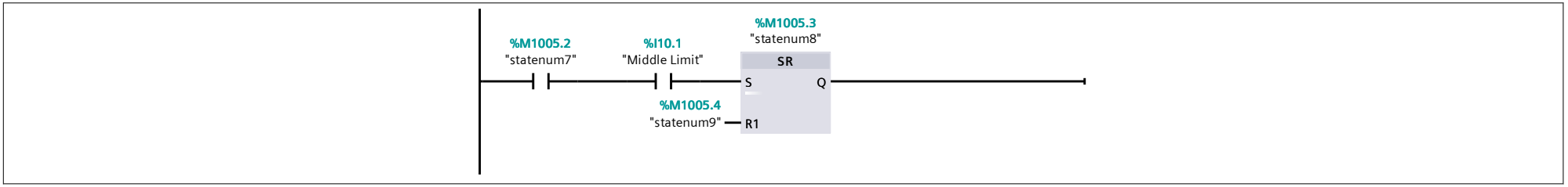
Network 7:



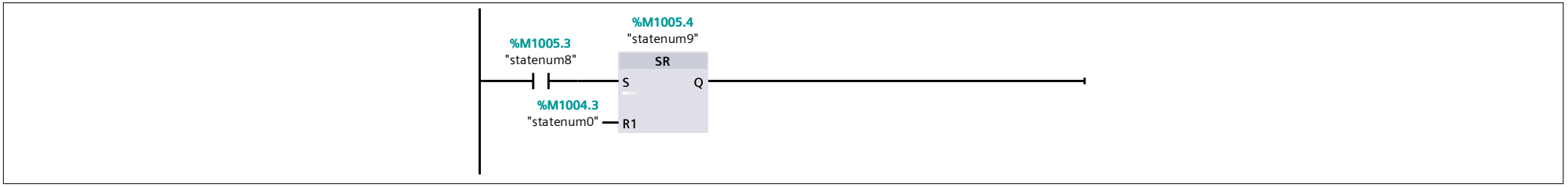
Network 8:



Network 9:



Network 10:

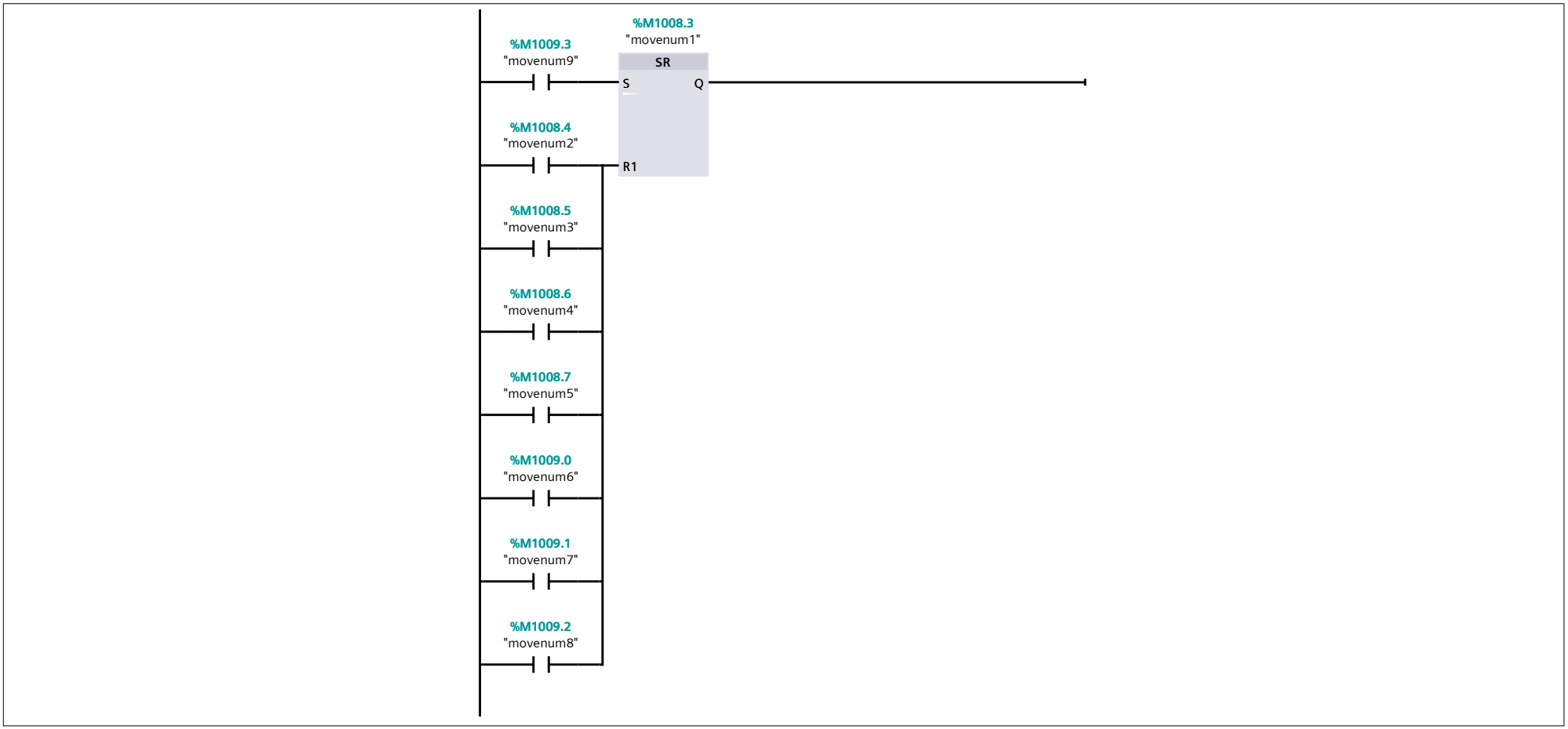


unloadsensors2 [FC26]

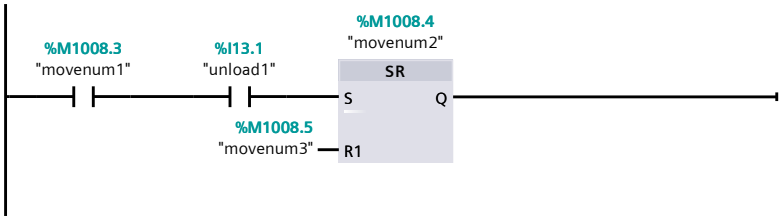
unloadsensors2 Properties							
General							
Name	unloadsensors2	**Number**	26	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
unloadsensors2	Void		

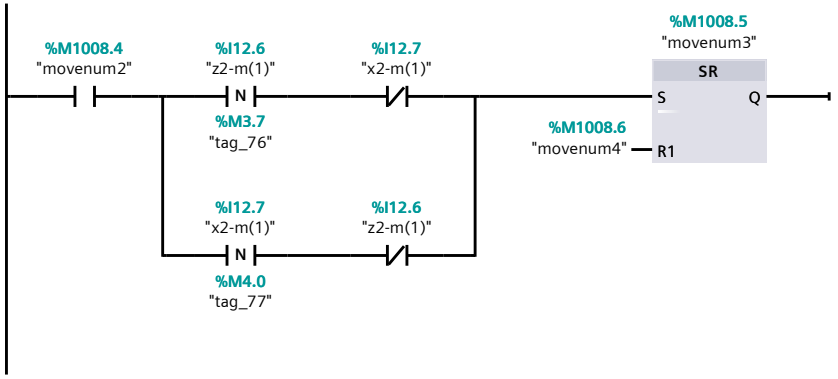
Network 1:


%M1008.3
"movenum1"
%M1009.3
"movenum9"
SR
S
Q
%M1008.4
"movenum2"
R1
%M1008.5
"movenum3"
%M1008.6
"movenum4"
%M1008.7
"movenum5"
%M1009.0
"movenum6"
%M1009.1
"movenum7"
%M1009.2
"movenum8"

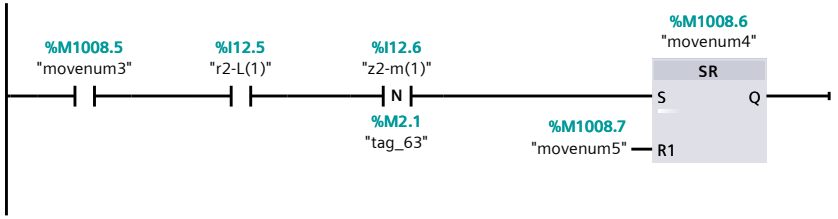
Network 2:



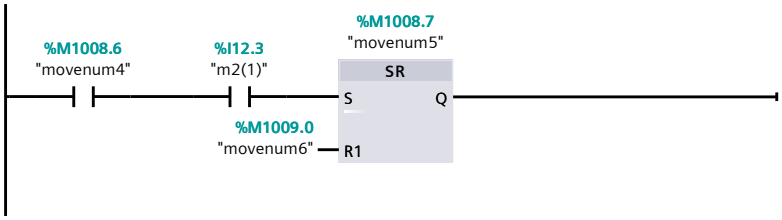
Network 3:



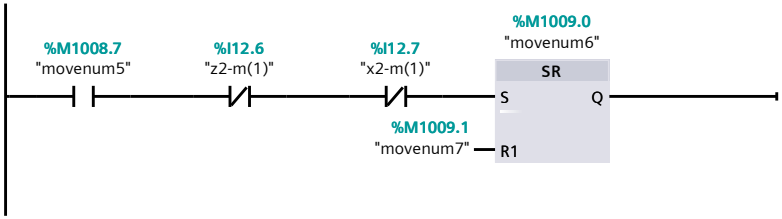
Network 4:



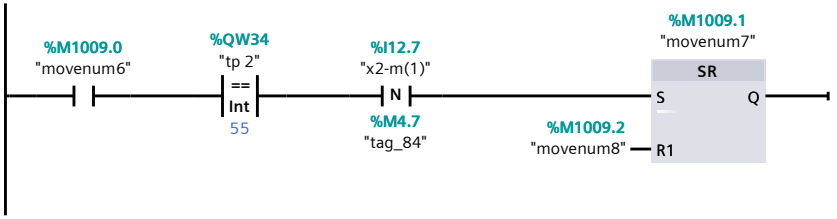
Network 5:



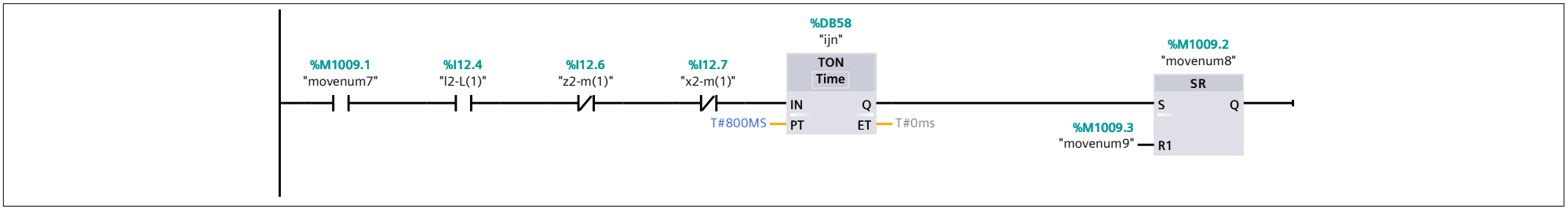
Network 6:



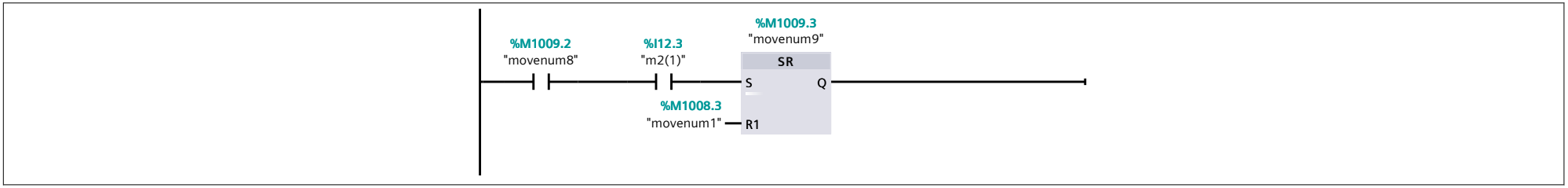
Network 7:



Network 8:


%M1009.1
"movenum7"
%I12.4
"I2-L(1)"
%I12.6
"z2-m(1)"
%I12.7
"x2-m(1)"
%DB58
"ijn"
TON
Time
IN
Q
T#800MS
PT
ET
T#0ms
%M1009.2
"movenum8"
SR
S
Q
%M1009.3
"movenum9"
R1

Network 9:

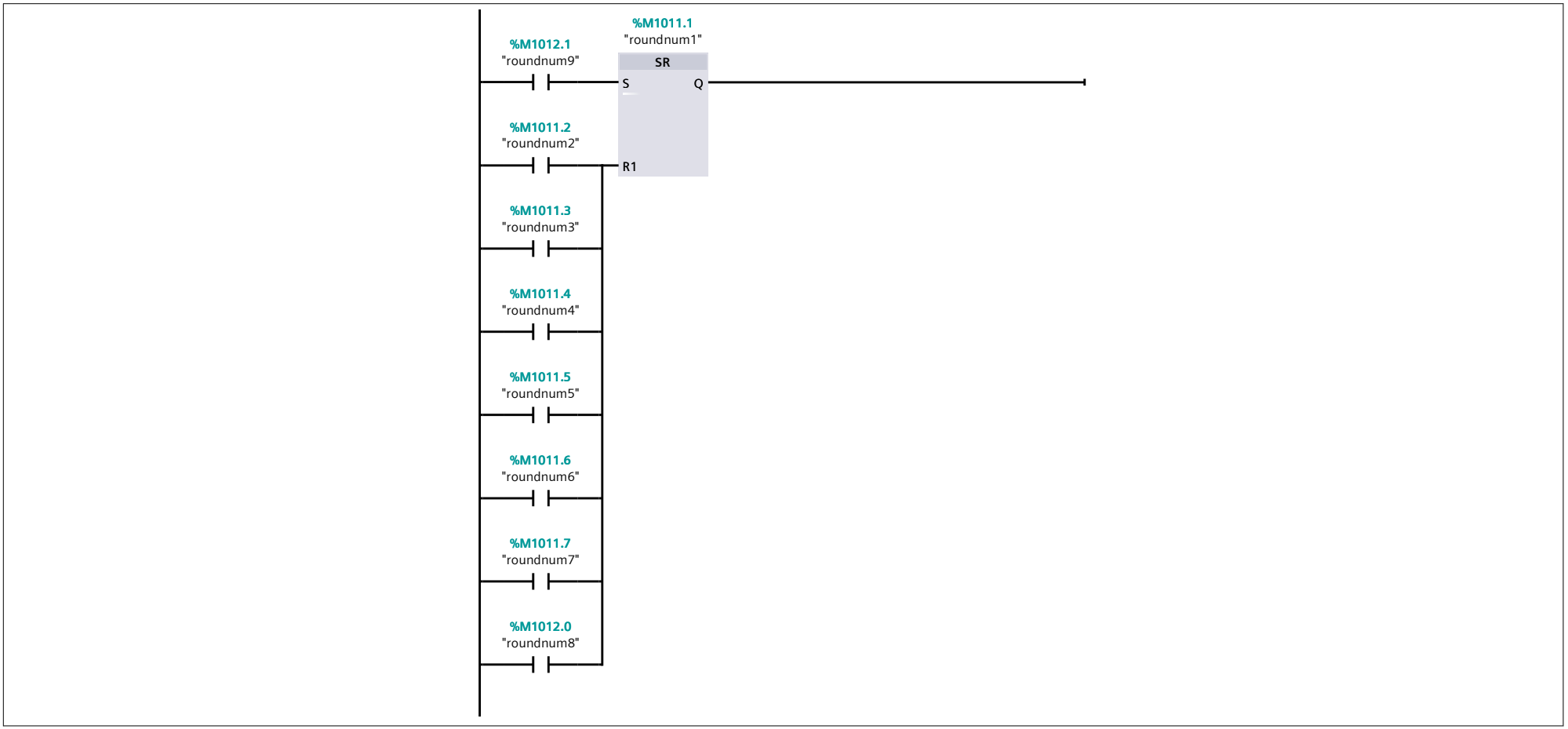

%M1009.2
"movenum8"
%I12.3
"m2(1)"
%M1009.3
"movenum9"
SR
S
Q
%M1008.3
"movenum1"
R1

unloadsensors3 [FC32]

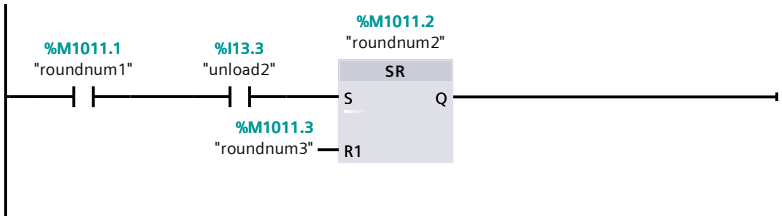
unloadsensors3 Properties							
General							
Name	unloadsensors3	**Number**	32	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
unloadsensors3	Void		

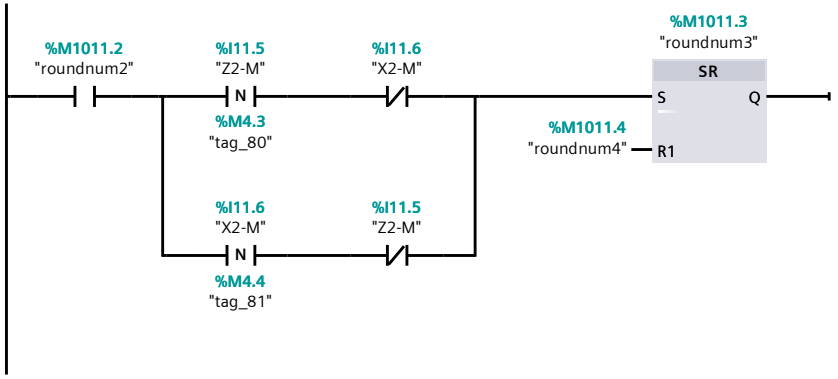
Network 1:


%M1012.1
"roundnum9"
%M1011.1
"roundnum1"
SR
S
Q
%M1011.2
"roundnum2"
R1
%M1011.3
"roundnum3"
%M1011.4
"roundnum4"
%M1011.5
"roundnum5"
%M1011.6
"roundnum6"
%M1011.7
"roundnum7"
%M1012.0
"roundnum8"

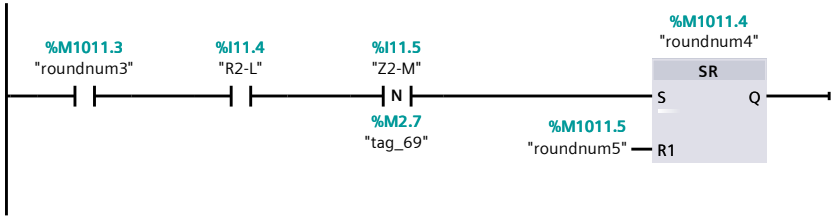
Network 2:



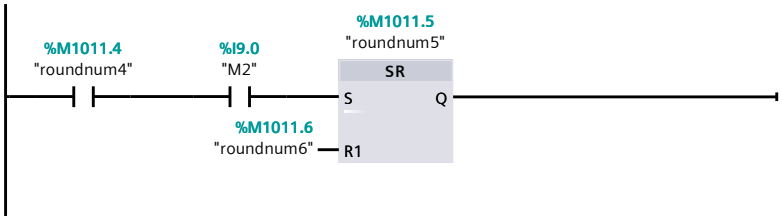
Network 3:



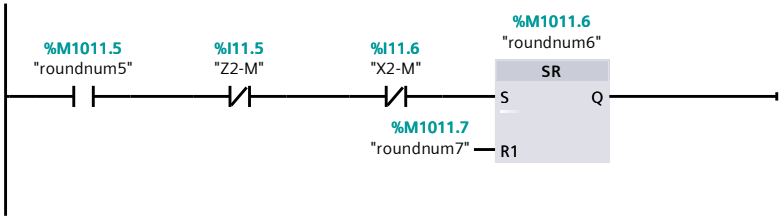
Network 4:



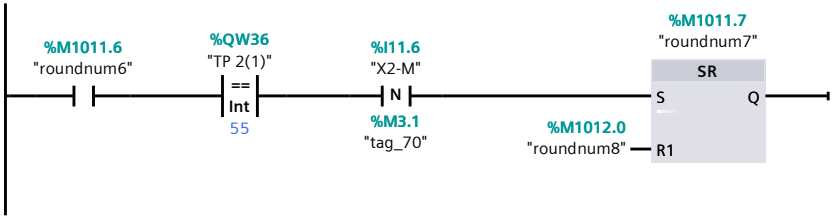
Network 5:



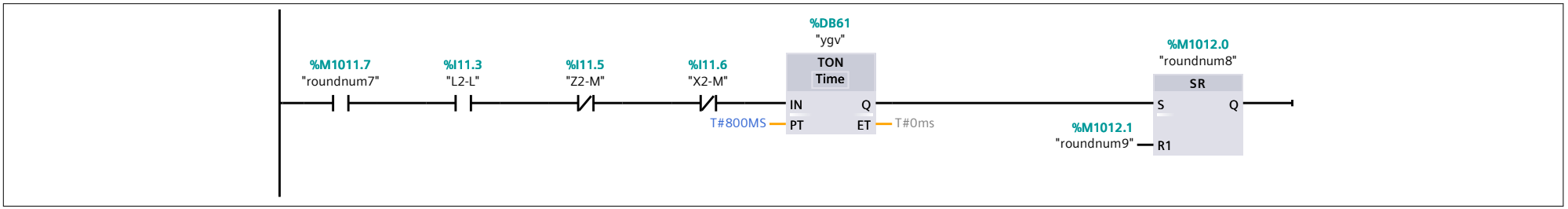
Network 6:



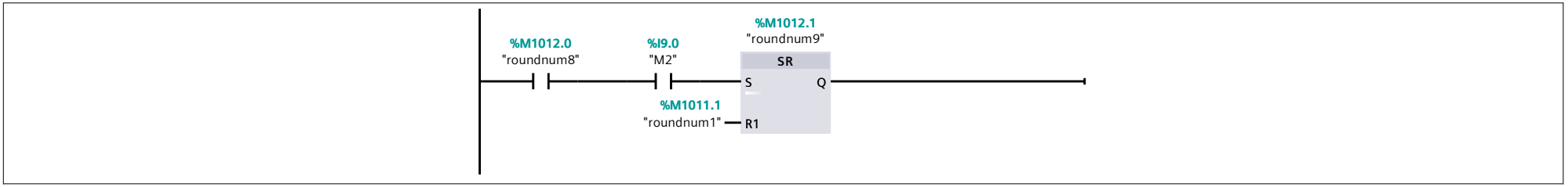
Network 7:



Network 8:


%M1011.7
"roundnum7"
%I11.3
"L2-L"
%I11.5
"Z2-M"
%I11.6
"X2-M"
%DB61
"ygv"
TON
Time
IN
Q
T#800MS
PT
ET
T#0ms
%M1012.0
"roundnum8"
SR
S
Q
%M1012.1
"roundnum9"
R1

Network 9:

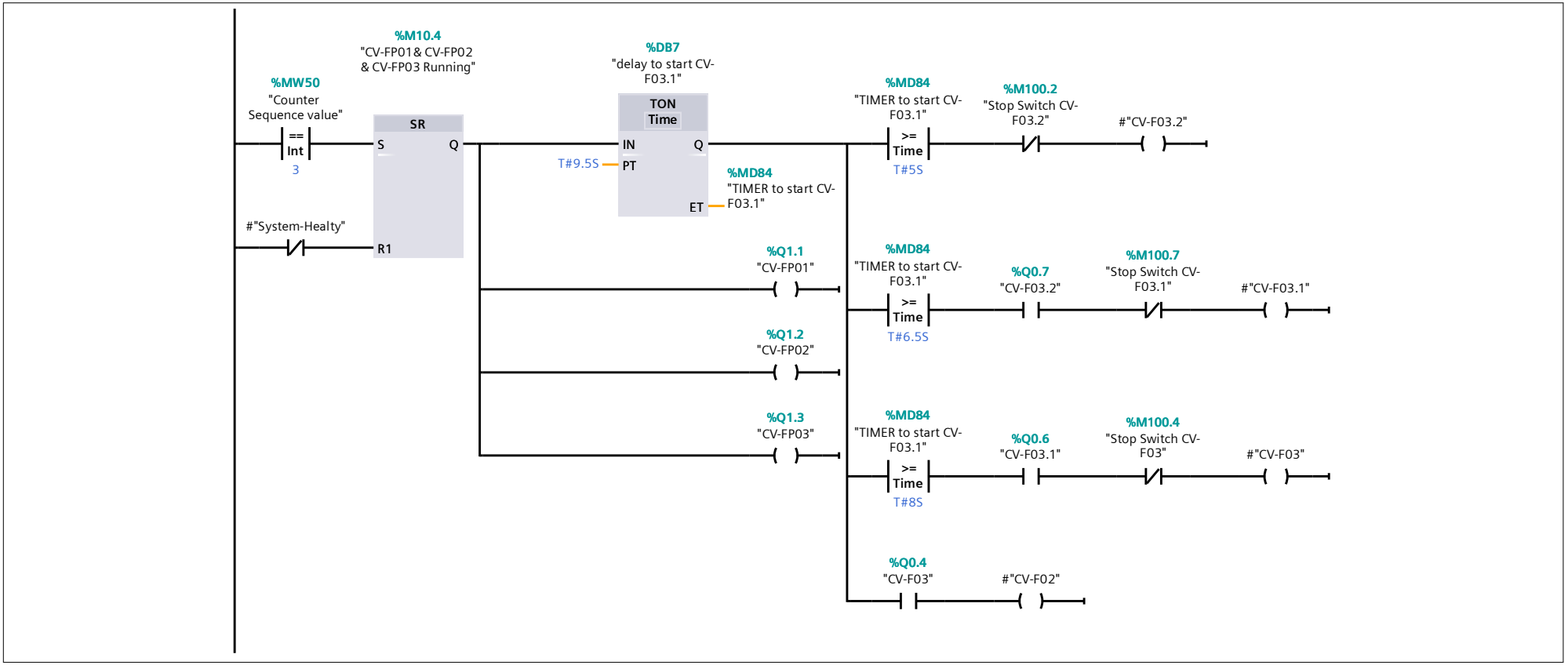

%M1012.0
"roundnum8"
%I9.0
"M2"
%M1012.1
"roundnum9"
SR
S
Q
%M1011.1
"roundnum1"
R1

UPPER Sequence feeding [FC2]

UPPER Sequence feeding Properties							
General							
Name	UPPER Sequence feeding	**Number**	2	**Type**	FC	**Language**	LAD
Numbering	Automatic						
Information							
Title		**Author**		**Comment**		**Family**	
Version	0.1	**User-defined ID**					

Name	Data type	Default value	Comment
▼ Input			
System-Healty	Bool		
▼ Output			
CV-FP01	Bool		
CV-FP02	Bool		
CV-FP03	Bool		
CV-F03	Bool		
CV-F03.1	Bool		
CV-F03.2	Bool		
CV-F02	Bool		
InOut			
Temp			
Constant			
▼ Return			
UPPER Sequence feeding	Void		

Network 1: Box feeder Conveyor


%M10.4
"CV-FP01& CV-FP02
& CV-FP03 Running"
%DB7
"delay to start CV-
F03.1"
%MW50
"Counter
Sequence value"
==
Int
3
SR
S
Q
R1
#"System-Healty"
TON
Time
IN
Q
PT
T#9.5S
ET
%MD84
"TIMER to start CV-
F03.1"
%MD84
"TIMER to start CV-
F03.1"
>=
Time
T#5S
%M100.2
"Stop Switch CV-
F03.2"
#"CV-F03.2"
%Q1.1
"CV-FP01"
%Q1.2
"CV-FP02"
%Q1.3
"CV-FP03"
%MD84
"TIMER to start CV-
F03.1"
>=
Time
T#6.5S
%Q0.7
"CV-F03.2"
%M100.7
"Stop Switch CV-
F03.1"
#"CV-F03.1"
%MD84
"TIMER to start CV-
F03.1"
>=
Time
T#8S
%Q0.6
"CV-F03.1"
%M100.4
"Stop Switch CV-
F03"
#"CV-F03"
%Q0.4
"CV-F03"
#"CV-F02"