

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'AUTOMATIQUE

Mémoire de Fin d'Etudes De MASTER ACADEMIQUE

Domaine : Sciences et Technologies

Filière : Automatique

Spécialité : Automatique et Informatique
Industrielle

Présenté par

Karim BERKAINE

Thème

**Automatisation et supervision de l'unité
de tri de pièces, sous Factory I/O :
Approche de modélisation par réseau de
Petri.**

Mémoire soutenu publiquement le 27/06/2024 devant le jury composé de :

M R.MELLAH

Professeur, UMMTO, Président

M M.CHARIF

MCA, UMMTO, Encadrant

MME G.IDIRI

MCB, UMMTO, Examineur

MME K.TITOUCHE

MCB, UMMTO, Examineur

Remerciment

Je tiens à exprimer mes sincères remerciements et ma profonde reconnaissance à mon promoteur, Monsieur CHARIF, pour ses précieux conseils et orientations tout au long de ce projet.

Je tiens également à exprimer ma reconnaissance envers les membres du jury qui m'honorent en évaluant ce travail.

Dédicace

Je dédie ce modeste travail à :

A mes chers parents,

En témoignage de ma gratitude et reconnaissance, si grandes qu'elles puissent être, pour tous les sacrifices qu'ils ont consentis pour mon bien-être et le soutien qu'ils m'ont prodigué tout au long de mon éducation. Que Dieu les préserve et leur procure santé et longue vie.

A mes sœurs et mon frère,

Pour leur encouragement et leur bonté qu'ils m'ont accordés, j'exprime ma profonde reconnaissance et mon grand respect.

Table des matières

Introduction Générale	1
Chapitre 1 : Présentation du Logiciel de Simulation 3D Factory I/O et Études des Stations de Tri	2
1 Introduction	2
2 Présentation du logiciel Factory I/O	2
2.1 Fonctionnalités Principales	2
2.2 Scènes Prédéfinies	4
2.3 Applications	5
3 Présentation et Étude de la Station de Tri par hauteur	6
3.1 Description de la Configuration :	6
3.2 Analyse du Capteur de la Station de Tri :	6
3.3 Analyse du Système de Tri par hauteur Employé dans la Station avancé - Sorting by Height (Advanced) :	7
3.4 Analyse du Système de Tri par hauteur Employé dans la Station Basique - Sorting by Height (basic) :	7
3.5 Etude du Mécanisme de tri :	8
4 Présentation et Étude de la Station de Tri par poids	9
4.1 Description de la Configuration :	9
4.2 Analyse du Système de tri :	10
4.3 Etude du Mécanisme de tri :	10
5 Présentation et Étude de la Station de Tri par couleur	11
5.1 Description de la Configuration :	11
5.2 Analyse du Capteur de la Station de Tri :	11
5.3 Analyse du Système de tri Employé dans la station de separation de piece :	11
5.4 Analyse du Système de tri Employé dans la station de tri de piece :	12
5.5 Etude du Mécanisme de tri :	13
6 Conclusion	14
Chapitre 2 : Modélisation des Stations de Tri par Réseaux de Petri	15
1 Présentation des Outils de Modélisation des Réseaux de Petri	15
1.1 Introduction	15
1.2 Rappels et notions de base des réseaux de Petri :	15
1.3 Evolution d'un RdP :	17
1.4 Les réseaux de Petri autonomes et non autonomes	18
1.5 Quelques propriétés des RdP :	19
1.6 Réseaux de Petri particuliers :	21
1.7 Eléments de modélisation	23
1.8 théorie des graphes :	24
1.9 Méthodes d'analyse et validation des réseaux de Pétri :	24
2 Modélisation à base de réseaux de pétri	27
2.1 Réseaux de petri de la station de tri par hauteur (station avancée)	27

2.2 Réseaux de petri du capteur de pièce	27
2.3 Nomenclature des places et transitions	29
2.4 resaux de petri du processus de tri	33
2.5 Nomenclature des places et transitions	33
2.6 Réseaux de petri de la station de tri par hauteur (station basique)	36
2.7 Nomenclature des places et transitions	36
2.8 Réseaux de petri de la station de tri par poids	38
2.9 Nomenclature des places et transitions	38
2.10 Réseaux de petri de la station de separation par couleur	41
2.11 Nomenclature des places et transitions	42
2.12 Réseaux de petri de la station de tri par couleur	44
2.13 Nomenclature des places et transitions	44
3 Validation des Réseaux de petri :	46
3.1 Vérification Structurale	46
3.2 Propriétés Dynamiques :	47
4 Transcription des Modèles Réseaux de Petri vers Modèles GRAFCET	47
4.1 Analyse des États et des Événements	47
4.2 Modélisation des Étapes et des Transitions	47
4.3 Établissement des Liaisons et des Conditions	48
4.4 Validation et Vérification du Modèle Converti	48
4.5 Conversions de Base	48
5 Comparaison entre les Réseaux de Petri et le LADDER	49
5.1 Structure et représentation	49
5.2 Expressivité et puissance de modélisation	49
5.3 Analyse et simulation	49
6 Transcription des Réseaux de Petri en Langage Ladder	50
6.1 Tableau de Correspondance : Conversion des Réseaux de Petri en Langage Ladder	50
6.2 Méthodologie de Modélisation et de Traduction des Réseaux de Petri vers le Langage Ladder	50
7 Conclusion	53
Chapitre 3 : Développement de la solution de commande sous Step7.	54
1 Introduction	54
2 Description des Automate programmable industriel (API)	54
2.1 Structure matériel d'un API	54
2.2 Aspect interne d'un automate programmable	54
2.3 Aspect externe d'un automate programmable	55
2.4 Les principaux éléments de l'automate	56
2.5 Programmation d'un API	57
2.6 Choix d'un automate	58
3 Présentation du logiciel SIMATIC Manager STEP7	58
3.1 Le logiciel de programmation STEP7	58
3.2 Programmation sur STEP 7	59
3.3 Description des blocs de programmation :	59
4 Développement de la solution de commande de la station pick&place sous Step 7.	60
4.1 Configuration materielle	60
4.2 La table mnémonique	61
4.3 les blocs de programmation	61
4.4 Développement du programme LADDER	62

4.5 La mise en place d'une liaison entre les deux logiciel (simatic step7 et factory i/o) et Lancement de la Simulation	63
5 Conclusion	64

Chapitre 4 : Développement de la solution de commande supervisée sous WinCC. 65

1 Introduction	65
2 Présentation du logiciel WinCC flexible	65
2.1 WinCC flexible ES	65
2.2 WinCC Runtime	66
3 Développement de la solution de commande supervisée de la station pick&place sous WinCC.	66
4 conclusion	67
Conclusion Générale	69

Table des figures

1	Bibliothèque d'Éléments Industriels	2
2	Interface de Factory I/O	3
3	Protocoles de Communication pris en charge par Factory I/O	3
4	Station de production	4
5	Station élévatrice	4
6	Station d'assemblage	5
7	Station de tri par poids	5
8	Matrices de lumière	6
9	Table tournante	7
10	Chaîne de transfert	8
11	convoyeurs à pesée intégrée	9
12	trieuse à roue escamotable	10
13	capteurs de vision	11
14	poussoir pneumatique	12
15	déviateur à bras incliné	13
16	Exemple d'un Reseau de Petri	16
17	Transition source	16
18	Transition puits	17
19	Transition validée (gauche), non validée (droite)	17
20	RdPs avant franchissement de T1 (gauche) et apres franchissement de T1 (droite)	18
21	Exemple d'evolution d'un RdP	18
22	succession des quatre saisons.	19
23	Exemple sur une transistion non vivante	20
24	RDP vivant	20
25	Graphe d'état	21
26	Graphe d'événements	21
27	Réseau de Petri sans conflits	21
28	Réseau de Petri avec conflits	22
29	RDP sans choix libre(gauche), RDP avec choix libre(droite)	22
30	Exemple d'un RDP à capacités	22
31	Réseaux de Petri P-temporisés.	23
32	Réseaux de Petri T-temporisés.	23
33	deux machines en concurrence.	24
34	Ensemble de marquage.	24
35	Graphe de marquage accessible.	25
36	RDP avec sa matrice post-incidence.	26
37	RDP avec sa matrice pré-incidence.	26
38	arbre de probabilité	28
39	RDP du 1er sous modèle	29
40	RDP du 2eme sous-modèle	33
41	Réseaux de petri de la station de tri "sorting by height (basic)"	36
42	Réseaux de petri de la station de tri "sorting by weight"	38

43	Réseaux de petri de la station de tri "separating station"	41
44	Réseaux de petri de la station de tri "sorting station"	44
45	Fragment d'un Réseau de Petri	51
46	le programme en Ladder	52
47	le programme en Ladder	52
48	Structure interne d'un automate programmable industriel (API).	55
49	Automate programmable compact.	56
50	Automate programmable modulaire	56
51	Configuration matérielle.	59
52	Station pick&place de factory i/o.	60
53	Configuration materielle.	61
54	table des mnémonique	61
55	vue d'Accueil.	66
56	Vue de Commande.	67
57	Vue de la représentation graphique des états des capteurs.	67

Liste des tableaux

1	Capteurs du systeme de tri	7
2	Actionneurs du systeme de tri	7
3	Capteurs du systeme de tri	8
4	Actionneurs du systeme de tri	8
5	Actionneur du systeme de tri	10
6	Capteursr du systeme de tri	12
7	Actionneur du systeme de tri	12
8	Actionneur du systeme de tri	13
9	représentation graphique des éléments du RDP	15
10	Place du 1er sous-modèle	31
11	Transition du 1er sous-modèle	32
12	Place du 2eme sous modèle	34
13	Transition du 2eme sous-modèle	35
14	Places du RDP	37
15	Transitions du RDP	37
16	place du RDP	39
17	Transition du RDP	40
18	Places du RDP	42
19	transitions du RDP	43
20	Places du RDP	45
21	Transition du RDP	46
22	Conversions de Base	48
23	Tableau de Correspondance (conversion des rdp's en LADDER)	50

Introduction Générale

L'industrie moderne se caractérise par une demande croissante d'efficacité et de flexibilité dans les processus de production. Avec l'augmentation de la concurrence mondiale et les attentes des consommateurs, les entreprises doivent constamment améliorer leurs méthodes de production. Dans ce contexte, les systèmes de tri automatisés jouent un rôle fondamental, permettant de gérer une diversité de produits avec une précision et une rapidité accrues. Ces systèmes sont essentiels pour trier efficacement des milliers de produits différents, réduire les temps de traitement, minimiser les erreurs humaines et maximiser l'utilisation des ressources. Pour concevoir et optimiser ces systèmes complexes, il est essentiel d'intégrer des outils de simulation avancés, des modèles théoriques rigoureux, et des solutions de commande performantes. La simulation permet d'anticiper les problèmes potentiels et de tester les solutions avant leur mise en œuvre réelle, réduisant ainsi les coûts et les risques associés aux modifications de dernière minute.

Factory I/O, un logiciel de simulation 3D, émerge comme un outil indispensable pour la création et l'optimisation des stations de tri. En simulant des environnements de production réalistes, Factory I/O permet de tester et d'affiner les paramètres de tri, tels que la hauteur, le poids et la couleur des objets, avant leur déploiement dans le monde réel. Cette capacité à anticiper et résoudre les problèmes potentiels contribue à améliorer l'efficacité et la fiabilité des systèmes de tri. De plus, Factory I/O permet de simuler des scénarios variés et de former les opérateurs dans un environnement virtuel, ce qui réduit les temps d'arrêt et les coûts de formation.

Pour modéliser les stations de tri, nous avons choisi la modélisation avec les réseaux de Petri (RdP), car ils offrent une meilleure représentation pour les systèmes complexes. Les réseaux de Petri permettent de visualiser et d'analyser les flux de travail, d'identifier les goulots d'étranglement et de vérifier la synchronisation des différentes opérations. Cette approche mathématique rigoureuse est particulièrement utile pour garantir que les systèmes de tri fonctionnent sans heurts et peuvent s'adapter aux changements de production.

L'implémentation de notre solution de contrôle se fera sous automate programmable industriel S7-300 de marque Siemens avec Step7, un logiciel de programmation pour automates programmables industriels (API). La programmation avec Step7 permet de développer des solutions de contrôle robustes et flexibles, adaptées aux spécificités de chaque application de tri. Cette étape assure que les systèmes de tri peuvent répondre efficacement aux exigences dynamiques de la production industrielle, en permettant des ajustements rapides et précis en fonction des besoins.

De plus, nous allons intégrer des interfaces utilisateur avec WinCC Flexible 2008, permettant une interaction intuitive et conviviale avec les systèmes de tri. Cette intégration améliore la capacité de surveillance et de contrôle des processus de tri en temps réel, offrant aux opérateurs une vue d'ensemble claire et des outils pour intervenir rapidement en cas de problème. Les interfaces utilisateur bien conçues augmentent l'efficacité des opérations, réduisent les erreurs humaines et permettent une gestion plus agile des flux de production.

En combinant la simulation avancée, la modélisation théorique et les solutions de commande performantes, cette approche intégrée vise à développer des systèmes de tri automatisés qui se distinguent par leur performance, leur fiabilité et leur flexibilité.

Chapitre 1 : Présentation du Logiciel de Simulation 3D Factory I/O et Études des Stations de Tri

1 Introduction

Factory I/O, développé par Real Games, est une plateforme immersive et flexible dédiée à la modélisation, la simulation et la visualisation des processus industriels. Ce logiciel est largement adopté dans l'enseignement, la formation professionnelle et le développement de solutions d'automatisation. Grâce à ses fonctionnalités avancées, Factory I/O permet la création de scénarios de production complexes en intégrant divers éléments comme des capteurs, des actionneurs, des convoyeurs et d'autres équipements industriels. Les utilisateurs peuvent configurer des environnements personnalisés, simuler des processus de fabrication et évaluer les performances des systèmes automatisés sans risque pour les opérations réelles. [11]

2 Présentation du logiciel Factory I/O

2.1 Fonctionnalités Principales

Factory I/O propose une gamme complète de fonctionnalités pour créer des environnements industriels virtuels réalistes :

-Bibliothèque d'Éléments Industriels : Le logiciel comprend une vaste bibliothèque d'éléments industriels préconfigurés, tels que des capteurs, des actionneurs, des convoyeurs, des machines, etc. Ces éléments peuvent être facilement intégrés dans des scènes pour représenter divers processus industriels.



FIGURE 1 – Bibliothèque d'Éléments Industriels

-Interface Intuitive : L'interface conviviale de Factory I/O permet aux utilisateurs de concevoir des systèmes complexes de manière intuitive. Les éléments peuvent être ajoutés, configurés et interconnectés à l'aide d'outils simples de glisser-déposer.

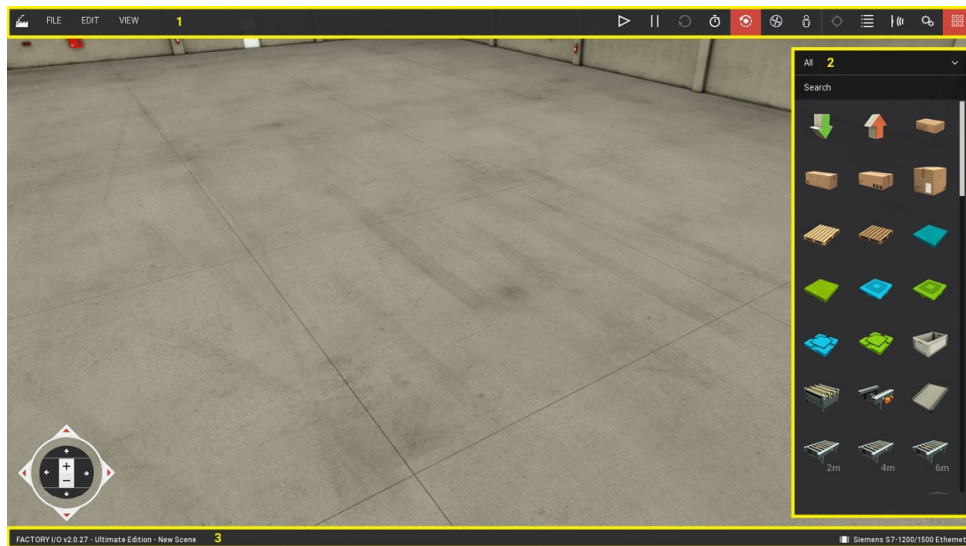


FIGURE 2 – Interface de Factory I/O

-Simulation en Temps Réel : Factory I/O offre une simulation en temps réel, permettant aux utilisateurs de visualiser instantanément l'impact de leurs modifications sur le système. Cela facilite le processus de conception et de débogage des systèmes automatisés.

-Personnalisation Avancée : Les utilisateurs peuvent personnaliser chaque élément de la scène en ajustant ses propriétés et en définissant des comportements spécifiques. Cette fonctionnalité permet de créer des simulations hautement réalistes et adaptées aux besoins spécifiques de chaque projet.

-Compatibilité avec les Protocoles de Communication : Factory I/O prend en charge plusieurs protocoles de communication industriels, tels que Modbus TCP/IP, OPC UA, et Siemens S7, ce qui facilite l'intégration avec des systèmes de contrôle réels.

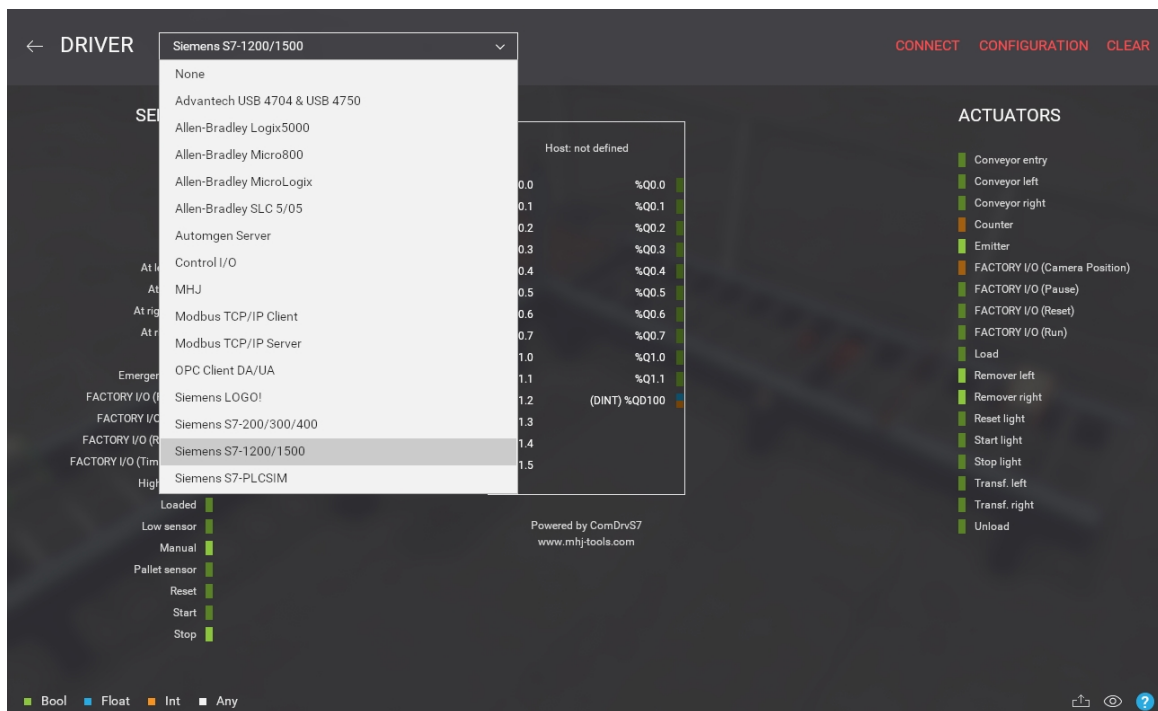


FIGURE 3 – Protocoles de Communication pris en charge par Factory I/O

2.2 Scènes Prédéfinies

Factory I/O propose également une sélection de scènes prédéfinies qui illustrent différents types de processus industriels. Ces scènes offrent aux utilisateurs des exemples concrets d'applications industrielles et servent de point de départ pour explorer les fonctionnalités du logiciel. Les types de scènes incluent :

-Scènes de Production : Simulent des lignes de production typiques avec des convoyeurs, des robots, des machines de traitement, etc.



FIGURE 4 – Station de production

-Scènes de Logistique : Mettent en avant les processus de manipulation et de transport de marchandises dans un entrepôt ou un centre de distribution.



FIGURE 5 – Station élévatrice

-Scènes d'Assemblage : Représentent des processus d'assemblage où différents composants sont assemblés pour former des produits finis.

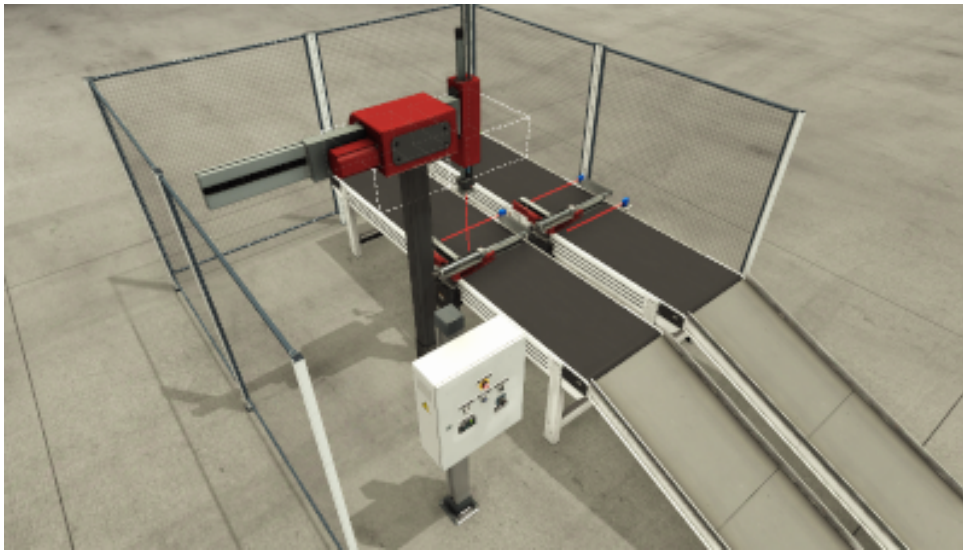


FIGURE 6 – Station d’assemblage

-Scènes de Tri : Simulent des systèmes de tri où des objets sont classés en fonction de différents critères tels que la taille, le poids ou d’autres caractéristiques.



FIGURE 7 – Station de tri par poids

2.3 Applications

Factory I/O trouve des applications dans divers domaines de l’automatisation industrielle, notamment :

-Formation Professionnelle : Il est largement utilisé dans les programmes de formation professionnelle pour familiariser les étudiants avec les concepts et les technologies de l’automatisation industrielle.

-Développement de Solutions Industrielles : Les ingénieurs et les concepteurs utilisent Factory I/O pour prototyper, tester et valider des solutions d’automatisation avant leur déploiement dans des environnements réels.

-Recherche et Développement : Les chercheurs et les développeurs utilisent Factory I/O pour explorer de nouvelles idées, tester des algorithmes et optimiser les processus industriels.

-Débogage et Maintenance : Il est également utilisé pour le débogage et la maintenance des systèmes automatisés, permettant aux techniciens de simuler des scénarios de panne et de tester des solutions de réparation.

Factory I/O est un outil puissant et polyvalent pour la modélisation et la simulation des processus industriels. Sa facilité d'utilisation, ses fonctionnalités avancées et sa compatibilité avec les protocoles de communication en font un choix idéal pour les professionnels de l'automatisation. En fournissant une plateforme immersive et réaliste, Factory I/O contribue à accélérer le développement, l'optimisation et la compréhension des systèmes automatisés.

3 Présentation et Étude de la Station de Tri par hauteur

3.1 Description de la Configuration :

La station de tri "Sorting by Height" dans Factory I/O est conçue pour trier les produits en fonction de leur hauteur. Elle est composée d'un convoyeur principal où les produits sont placés pour être transportés à travers la station. Des capteurs de hauteur sont installés le long du convoyeur pour mesurer les dimensions des produits alors qu'ils se déplacent. En fonction de la hauteur détectée, les produits sont dirigés vers différentes sorties par des mécanismes de tri.

3.2 Analyse du Capteur de la Station de Tri :

Dans la station "Sorting by Height" de Factory I/O, les capteurs de hauteur utilisés sont des capteurs "light array", également appelés en français des "matrices de lumière".

Les capteurs "Light Array" fonctionnent en émettant une série de faisceaux lumineux à travers la zone de détection. Lorsque les produits passent à travers ces faisceaux, ils bloquent partiellement ou complètement la lumière, ce qui est détecté par les photodétecteurs du capteur. En mesurant la quantité de lumière bloquée par chaque faisceau, le capteur peut déterminer avec précision la hauteur des produits.

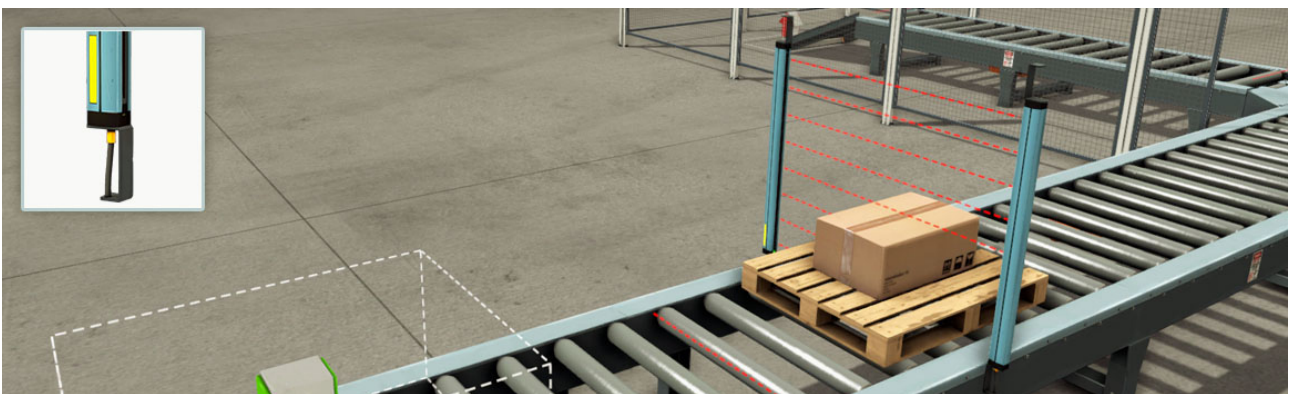


FIGURE 8 – Matrices de lumière

caractéristiques du capteur :

- Nombre de faisceaux : 8
- Portée : 1,5 m
- Matériaux détectables : solides
- Valeur analogique = $10 * \text{nombre de faisceaux interrompus} / 8$

3.3 Analyse du Système de Tri par hauteur Employé dans la Station avancé - Sorting by Height (Advanced) :

Plateau tournant robuste motorisé, généralement utilisé pour trier les palettes. Il est équipé de rouleaux périphériques à rotation libre et de capteurs préinstallés.

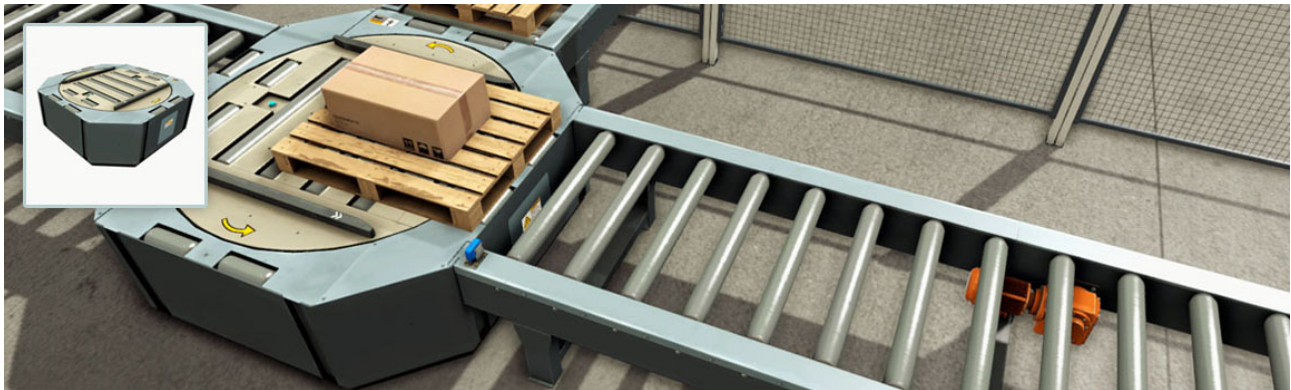


FIGURE 9 – Table tournante

Caractéristique du système de tri :

- Rayon du rouleau : 0,045 m
- Vitesse de transport maximale : 0,45 m/s
- Vitesse de rotation de la table : 0,7 rad/s
- Portée des capteurs capacitifs : 0 - 0,1 m

Capteurs et Actionneurs du système de tri :

Capteur	Type	Adresse	Description
At front	Bool	E0.6	Ce capteur est utilisé pour détecter les objets positionnés à l'avant de la table tournante
At back	Bool	E2.1	Ce capteur est utilisé pour détecter les objets positionnés à l'arrière de la table tournante

TABLE 1 – Capteurs du système de tri

Actionneur	Type	Adresse	Description
Load	Bool	A0.2	Cette sortie est dédiée à l'activation des rouleaux du convoyeur de la table, permettant ainsi aux objets positionnés sur la table de se déplacer vers l'avant
Unload	Bool	A0.3	Cette sortie est dédiée à l'activation des rouleaux du convoyeur de la table, permettant ainsi aux objets positionnés sur la table de se déplacer vers l'arrière
Turn	Bool	A0.4	Cette sortie est destinée à actionner la rotation de la table dans le sens antihoraire

TABLE 2 – Actionneurs du système de tri

3.4 Analyse du Système de Tri par hauteur Employé dans la Station Basique - Sorting by Height (basic) :

transporteur à chaîne, Utilisé pour transférer des marchandises sur des convoyeurs adjacents, il fonctionne mieux avec des palettes carrées. Il est composé de rouleaux de chargement et de

trois pistes de chaînes.

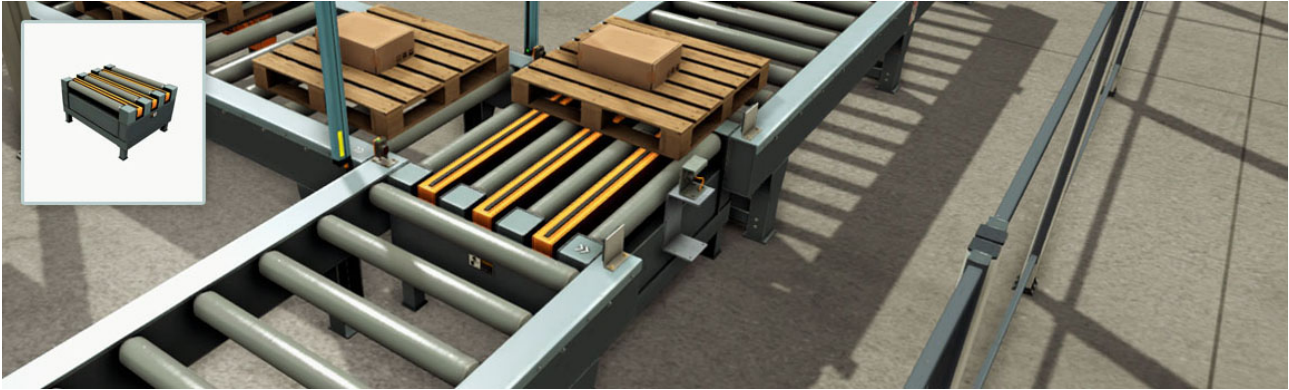


FIGURE 10 – Chaîne de transfert

Caractéristique du système de tri :

- Course de la piste de chaîne : 0,04 m
- Vitesse de transport maximale : 0,45 m/s
- Vitesse maximale de la chaîne : 0,45 m/s

Capteurs et Actionneurs du système de tri :

Capteur	Type	Adresse	Description
Loaded	Bool	E0.3	Ce capteur signale le moment où un objet atteint la limite de la chaîne de transfert

TABLE 3 – Capteurs du système de tri

Actionneur	Type	Adresse	Description
Load	Bool	A0.1	Cette entrée est dédiée à l'activation des rouleaux du convoyeur de la chaîne de transfert, permettant ainsi aux objets positionnés sur elle de se déplacer vers l'avant
Unload	Bool	A0.2	Cette entrée est dédiée à l'activation des rouleaux du convoyeur de la chaîne de transfert, permettant ainsi aux objets positionnés sur la elle de se déplacer vers l'arrière
Transf . left	Bool	A0.3	Cette entrée est dédiée à l'actionnement de la chaîne de transfert afin de déplacer les objets positionnés sur celle-ci vers la gauche
Transf . right	Bool	A0.4	Cette entrée est dédiée à l'actionnement de la chaîne de transfert afin de déplacer les objets positionnés sur celle-ci vers la droite

TABLE 4 – Actionneurs du système de tri

3.5 Etude du Mécanisme de tri :

Le mécanisme de tri de la station utilise les informations de hauteur fournies par les capteurs "Light Array" pour diriger les produits vers les destinations appropriées. En fonction des caractéristiques de hauteur détectées, des actionneurs pneumatiques, des bascules ou des guides mécaniques sont activés pour orienter les produits vers les voies de tri correspondantes,

assurant ainsi un tri précis et efficace.

la station de tri par la hauteur représente une composante essentielle dans les environnements industriels où la classification précise des produits est primordiale. Grâce à l'utilisation de capteurs "light array" ou matrices de lumière, cette station offre une méthode fiable et efficace pour mesurer avec précision les dimensions des produits à trier. La détection rapide et précise de la hauteur des produits permet une manipulation efficace et un routage vers les sorties appropriées, ce qui contribue à optimiser les processus de production et à améliorer l'efficacité globale de la chaîne de fabrication. En intégrant des mécanismes de tri adaptés, cette station offre une solution polyvalente et évolutive pour répondre aux besoins de tri dans une variété d'applications industrielles, soulignant ainsi son importance dans la gestion efficace des flux de production.

4 Présentation et Étude de la Station de Tri par poids

4.1 Description de la Configuration :

La station de tri "Sorting by Weight" dans Factory I/O est conçue pour trier les produits en fonction de leur poids. Elle est équipée d'un convoyeur principal où les produits sont placés pour être transportés à travers la station. Des capteurs de poids sont installés le long du convoyeur pour mesurer le poids des produits alors qu'ils se déplacent. En fonction du poids détecté, les produits sont dirigés vers différentes sorties par des mécanismes de tri.

Analyse du Capteur de la Station de Tri : Dans la station "Sorting by Weight" de Factory I/O, les capteurs utilisés pour mesurer le poids des produits sont des convoyeurs à pesée intégrée, ou "conveyor scales" en anglais. En français, on pourrait les appeler des "convoyeurs peseurs".



FIGURE 11 – convoyeurs à pesée intégrée

caractéristiques du capteur :

- Vitesse maximale de transport : 0,6 m/s
- Capacités : 20 - 100 kg

4.2 Analyse du Système de tri :

Un trieuse à roue escamotable double face à 45 degrés utilisée pour diriger les articles vers trois directions différentes via des rouleaux pivotants.



FIGURE 12 – trieuse à roue escamotable

Caractéristique du système de tri :

- Rayon de la roue : 0,05 m
- Vitesse de convoyage : 1,5 m/s
- Normalement centré
- Course des roues : 0,003 m

Actionneurs du système de tri :

Actionneur	Type	Adresse	Description
Send left	Bool	A0.2	Cette sortie est dédiée à l'actionnement des roues escamotables de la trieuse pour envoyer les objets vers la gauche
Send right	Bool	A0.4	Cette sortie est dédiée à l'actionnement des roues escamotables de la trieuse pour envoyer les objets vers la droite
Send forward	Bool	A0.6	Cette sortie est dédiée à l'actionnement des roues escamotables de la trieuse pour envoyer les objets vers l'avant

TABLE 5 – Actionneur du système de tri

4.3 Etude du Mécanisme de tri :

Les convoyeurs à pesée intégrée sont des dispositifs spécialement conçus pour mesurer le poids des produits pendant leur déplacement le long du convoyeur. Ces convoyeurs sont équipés de cellules de charge ou de capteurs de force intégrés dans la structure du convoyeur. Lorsque les produits passent sur ces convoyeurs, le poids est mesuré en fonction de la déformation ou de la pression exercée sur les capteurs de force.

la station "Sorting by Weight" de Factory I/O offre une solution efficace pour trier les produits en fonction de leur poids. Les convoyeurs à pesée intégrée, équipés de capteurs de force ou de cellules de charge, permettent une mesure précise du poids des produits pendant leur déplacement le long du convoyeur. Cette précision de mesure contribue à un tri fiable et précis des produits en fonction de leur poids.

Associés à des mécanismes de tri tels que des actionneurs pneumatiques, des bascules ou des rampes inclinées, les convoyeurs à pesée intégrée dirigent les produits vers les sorties appropriées en fonction du poids détecté. Cette combinaison de capteurs de poids et de mécanismes de tri offre une solution polyvalente et efficace pour répondre aux besoins de tri dans une variété d'applications industrielles simulées.

En somme, la station "Sorting by Weight" joue un rôle crucial dans l'optimisation des processus de production, en assurant un tri précis et efficace des produits en fonction de leur poids. Son utilisation permet d'améliorer l'efficacité opérationnelle et la gestion des flux de production, contribuant ainsi à une performance globale accrue dans les environnements industriels simulés.

5 Présentation et Étude de la Station de Tri par couleur

5.1 Description de la Configuration :

La station de séparation, ou "Separating Station", dans Factory I/O est conçue pour séparer les produits en fonction de critères prédéfinis tels que la couleur de la boîte. Elle est équipée d'un convoyeur principal où les boîtes sont transportées à travers la station. Des capteurs de vision sont utilisés pour détecter la couleur des boîtes, tandis que des mécanismes de tri sont ensuite activés pour diriger les boîtes vers les sorties appropriées en fonction de leur couleur détectée.

5.2 Analyse du Capteur de la Station de Tri :

Les capteurs utilisés dans cette station sont des capteurs de vision, des dispositifs optiques avancés qui capturent des images des boîtes en mouvement sur le convoyeur.

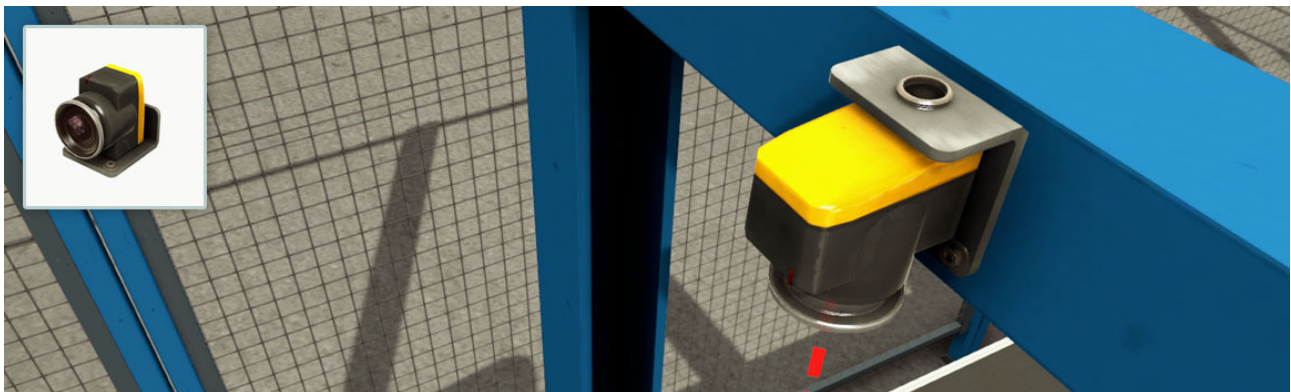


FIGURE 13 – capteurs de vision

caractéristiques du capteur :

- Matériaux détectables : Matières premières, Bases de produits, Couvercles de produits.
- Portée de détection : 0,3 - 2 m

5.3 Analyse du Système de tri Employé dans la station de separation de piece :

Trieur à poussoir pneumatique équipé de deux capteurs à lames indiquant les limites avant et arrière. Il comprend également une soupape servo utilisée pour régler et mesurer la position

de la tige. Le contrôle peut être effectué par des valeurs numériques ou analogiques selon la configuration sélectionnée.



FIGURE 14 – poussoir pneumatique

Caractéristique du système de tri :

- Vitesse par défaut : 1 m/s
- Vitesse rapide : 4 m/s
- Course : 0,9 m

Capteurs et Actionneurs du système de tri :

Capteur	Type	Adresse	Description
Pusher 1 Front	Bool	E0.1	Ces capteurs indiquent la limite avant de leur poussoir pneumatique respectif
Pusher 2 Front	Bool	E0.5	
Pusher 1 Back	Bool	E0.2	Ces capteurs indiquent la limite arrière de leur poussoir pneumatique respectif
Pusher 2 Back	Bool	E0.6	

TABLE 6 – Capteurs du système de tri

Actionneur	Type	Adresse	Description
Pusher 1	Bool	A0.1	Cette sortie est chargée d'actionner le premier poussoir pneumatique
Pusher 2	Bool	A0.4	Cette sortie est chargée d'actionner le deuxième poussoir pneumatique

TABLE 7 – Actionneur du système de tri

5.4 Analyse du Système de tri Employé dans la station de tri de pièce :

Un déviateur à bras incliné de 45°, alimenté par un moto-réducteur. Équipé d'une bande qui aide à dévier les articles transportés vers la partie suivante. Le bras peut tourner vers la gauche ou vers la droite selon la configuration sélectionnée.

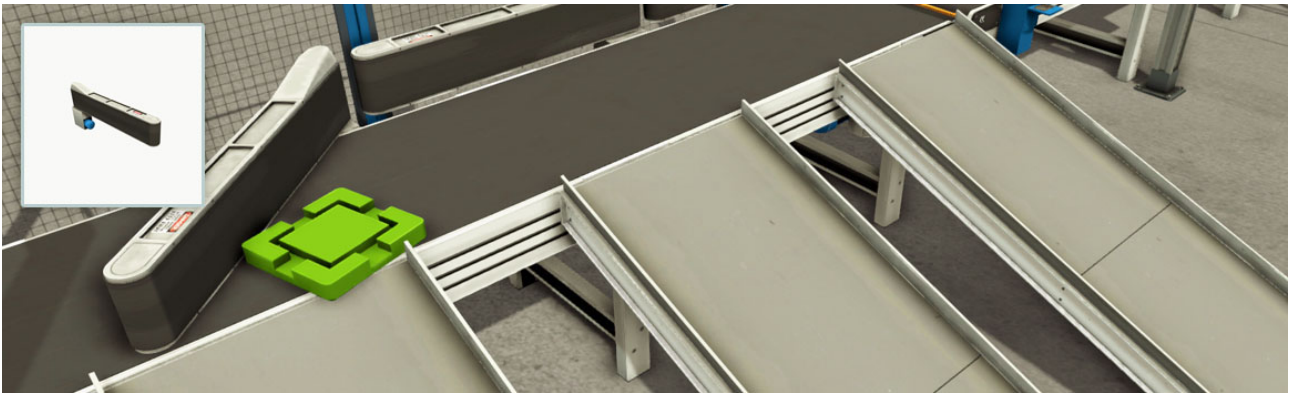


FIGURE 15 – déviateur à bras incliné

Caractéristique du système de tri :

- Vitesse de la bande : 2 m/s
- Vitesse angulaire du bras : 5 rad/s

Actionneurs du système de tri :

Actionneur	Type	Adresse	Description
sorter 1 turn	Bool	A0.3	Chacune de ces sorties est chargée de faire pivoter son Trieur à Bras Pivotant respectif à 45 degrés
sorter 2 turn	Bool	A0.5	
sorter 3 turn	Bool	A0.7	
sorter 1 belt	Bool	A0.4	Chacune de ces sorties est chargée de faire tourner la bande transporteuse de son Trieur à Bras Pivotant respectif
sorter 2 belt	Bool	A0.6	
sorter 3 belt	Bool	A1.0	

TABLE 8 – Actionneur du système de tri

5.5 Etude du Mécanisme de tri :

Ces capteurs analysent ensuite ces images pour détecter la couleur des boîtes. Les mécanismes de tri peuvent inclure des actionneurs pneumatiques, des bascules ou des guides mécaniques, qui sont activés en fonction des informations fournies par les capteurs de vision pour diriger les boîtes vers les sorties correspondant à leur couleur détectée.

la station de séparation dans Factory I/O représente une composante essentielle pour optimiser les processus de production en permettant la séparation efficace des boîtes en fonction de leur couleur. Grâce à l'utilisation de capteurs de vision avancés, cette station offre une précision élevée dans la détection de la couleur des boîtes en mouvement sur le convoyeur. Cette précision est essentielle pour garantir un tri précis et fiable, contribuant ainsi à l'efficacité globale de la chaîne de production.

L'avantage clé de cette station réside dans sa flexibilité, car elle peut être configurée pour détecter une variété de couleurs de boîtes, offrant ainsi une solution polyvalente pour répondre aux besoins spécifiques de différentes applications industrielles. De plus, la combinaison de capteurs de vision avec des mécanismes de tri adaptés permet une séparation rapide et efficace des boîtes, ce qui contribue à optimiser les délais de production et à améliorer la productivité globale.

6 Conclusion

En conclusion, les trois stations prédéfinies, à savoir "Sorting by Height", "Sorting by Weight" et "Separating Station", dans Factory I/O représentent des éléments cruciaux pour l'optimisation des processus de production dans les environnements industriels simulés. Chacune de ces stations offre une fonctionnalité spécifique qui contribue à la gestion efficace des flux de production et à l'amélioration de l'efficacité opérationnelle. La station "Sorting by Height" utilise des capteurs de hauteur, tels que les capteurs de lumière, pour trier les produits en fonction de leur hauteur, offrant ainsi une solution précise et fiable pour le tri des produits dans les environnements industriels. D'autre part, la station "Sorting by Weight" utilise des convoyeurs à pesée intégrée, associés à des capteurs de force ou de charge, pour trier les produits en fonction de leur poids, permettant ainsi une séparation efficace basée sur la masse des produits. Enfin, la station "Separating Station" utilise des capteurs de vision avancés pour détecter des caractéristiques spécifiques telles que la couleur des boîtes, offrant ainsi une solution flexible et précise pour séparer les produits en fonction de critères prédéfinis. Dans l'ensemble, ces trois stations prédéfinies démontrent l'importance des technologies de détection avancées telles que les capteurs de lumière, les convoyeurs à pesée intégrée et les capteurs de vision dans l'automatisation des processus de production. Leur utilisation permet d'optimiser les performances opérationnelles, d'améliorer la qualité des produits et de répondre aux exigences spécifiques de diverses applications industrielles simulées. En combinant efficacement ces technologies avec des mécanismes de tri appropriés, ces stations offrent une solution complète pour la gestion des flux de production dans les environnements industriels simulés, soulignant ainsi leur importance dans l'amélioration continue des processus industriels.

Chapitre 2 : Modélisation des Stations de Tri par Réseaux de Petri

1 Présentation des Outils de Modélisation des Réseaux de Petri

1.1 Introduction

De nos jours les systèmes informatiques sont devenus d'une grande complexité. En plus les systèmes dynamiques ne peuvent pas être décrits en ne prenant en compte que leurs états initiaux et finaux. En effet, on doit tenir compte de leur comportement permanent qui est une séquence d'états, pour qu'on puisse parler d'une description bien fondée. Parmi le grand nombre des techniques formelles pour spécifier, analyser et vérifier ce genre de systèmes, les réseaux de pétri sont l'une des plus utilisés.

Les Réseaux de Petri (RdP) constituent un formalisme graphique propre à la modélisation des Systèmes à Evénements Discrets (SED) introduit en 1962 par Carl Adam Petri. Ils sont particulièrement adaptés à l'étude des processus complexes mettant en jeu des propriétés de synchronisme et de partage de ressources. Leur support mathématique a permis en outre d'établir de nombreux résultats analytiques. Pour l'étude des SED dans l'algèbre des dioïdes, les RdP sont souvent utilisés comme un outil de modélisation intermédiaire. [1]

1.2 Rappels et notions de base des réseaux de Petri :

Systèmes à événements discrets : Un système à événements discrets est un système dont l'espace d'état est discret. Les transitions d'état sont provoquées par l'occurrence d'événements. L'ensemble des événements est également discret. Un événement est validé si l'ensemble des conditions nécessaires à son occurrence sont vérifiées.

Pour prendre en compte le temps, des temporisations peuvent être associées aux événements, une temporisation est un laps de temps qui doit être écoulé avant qu'un événement validé puisse survenir.

Plusieurs outils mathématiques ont été proposés pour modéliser les systèmes à événements discrets on cite par exemple : le GRAFCET, LADER, et les réseaux de Petri.

Réseaux de Petri : Un réseau de Petri est un graphe biparti fait de deux types de sommets : places (représentées par des cercles) et transitions (représentées par des barres). Des arcs orientés relient certaines places à certaines transitions, ou certaines transitions à certaines places. Une place peut contenir un nombre entier de marques ou jeton. A chaque arc on associe un poids (entier positif). Un arc sur lequel il ne figure aucun poids veut dire que son poids est égal à 1.




Elément	Place	Transition	Arc orienté
Symbole graphique			

TABLE 9 – représentation graphique des éléments du RDP

Regle de syntaxe :

Un arc relie soit une place a une transition, soit une transition a une place jamais une place a une place ou une transition a une transition. Tout arc doit avoir a son extrémité un sommet (place ou transition). Un exemple de Reseaux de Petri est representé Figure 9. A chaque place et transition, un nom peut etre associé : par exemple sur le RdP de la Figure 9, les places sont nommées P1, P2, P3 et P4 et les transitions T1, T2 et T3. T1 est reliee a P1 par un arc oriente de T1 vers P1 : on dit que P1 est en sortie ou en aval de T1. P1 est reliee a T2 par un arc oriente de P1 vers T2 : on dit que T2 est en sortie de P1. De meme, on peut dire que P1 est en entrée ou en amont de T2. La place P3 est en entrée de T1

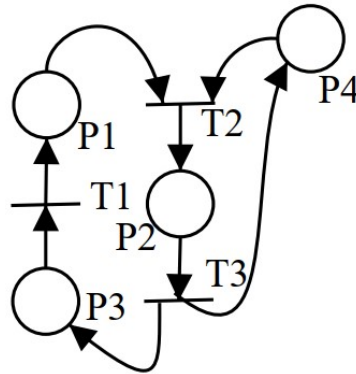


FIGURE 16 – Exemple d'un Reseau de Petri

Place et Transition dans la Modélisation Systémique :

Une place correspond à une "variable d'état" du système qui va être modélisée, et une transition à un événement et/ou une action qui va entraîner l'évolution des variables d'état du système. À un instant donné, une place contient un certain nombre de marques ou jetons qui vont évoluer en fonction du temps, indiquant ainsi la valeur de la variable d'état à cet instant. Lorsqu'un arc relie une place à une transition, cela signifie que la valeur de la variable d'état associée à la place influence l'occurrence de l'événement associé à la transition. De même, lorsque qu'un arc relie une transition à une place, cela indique que l'occurrence de l'événement associé à la transition influence la valeur de la variable d'état associée à la place.

Cas particuliers :

- **Transition source** : pas de place en entree de la transition



FIGURE 17 – Transition source

- **Transition puits** : pas de place en sortie de la transition



FIGURE 18 – Transition puits

Marquage : Chaque place contient un nombre entier positif ou nul de marquage (jeton). Le marquage M définit l'état du système décrit par le réseau à un instant donné. C'est un vecteur colonne de dimension m (le nombre de places dans le réseau). La j^{me} composante de ce vecteur représente le nombre de marque dans la j^{me} place du réseau.

On note : M_0 marquage initial

1.3 Evolution d'un RdP :

L'évolution d'un RdP correspond à l'évolution de son marquage au cours du temps (évolution de l'état du système) : il se traduit par un déplacement de marques ce qui s'interprète comme la consommation/production de ressources déclenchée par des événements ou des actions. Déterminer l'évolution d'un RdP correspond en fait à le simuler, terme plus généralement utilisé en modélisation.

Transition validée : Une transition est dite validée si toutes les places en amont (c'est-à-dire en entrée) de celle-ci possèdent au moins une marque. Une transition source est, par définition, toujours validée.

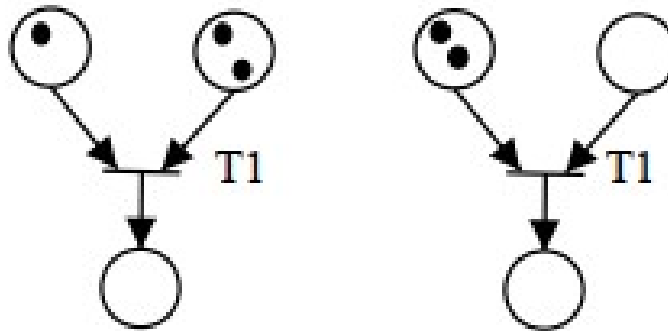


FIGURE 19 – Transition validée (gauche), non validée (droite)

Franchissement : Si la transition est validée, on peut effectuer le franchissement de cette transition, on dit alors que la transition est franchissable. Le franchissement consiste à :

- retirer une marque dans chacune des places en entrée de la transition.
- ajouter une marque à chacune des places en sortie.



FIGURE 20 – RdPs avant franchissement de T1 (gauche) et après franchissement de T1 (droite)

Remarque : L'évolution du RdP se fait par le franchissement d'une seule transition à la fois. Quand plusieurs transitions sont simultanément franchissables, on ne peut pas savoir dans quel ordre elles seront effectivement franchies. L'évolution n'est donc pas unique. (Un exemple d'évolution d'un RdP est représenté dans la Figure 14)

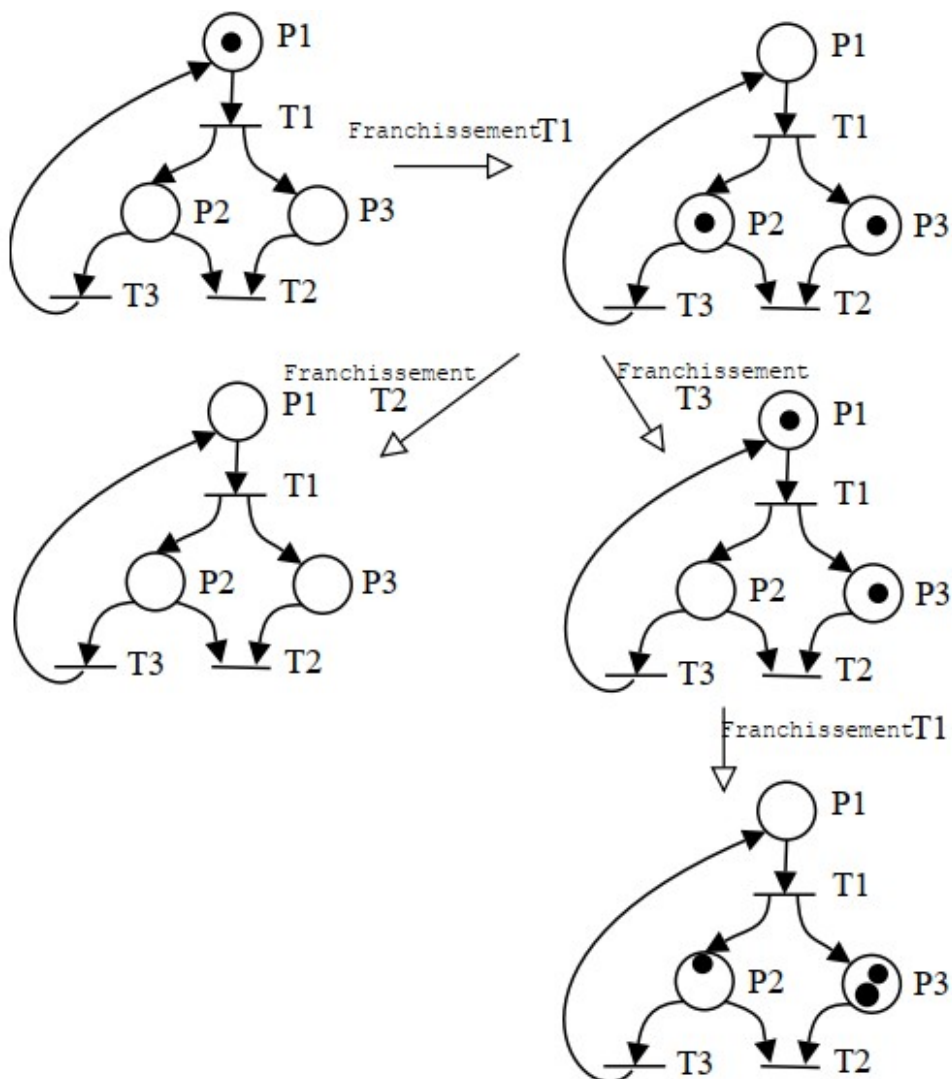


FIGURE 21 – Exemple d'évolution d'un RdP

1.4 Les réseaux de Petri autonomes et non autonomes

Réseau de Petri autonome : Un réseau de Petri autonome décrit le fonctionnement d'un système de manière autonome, c'est-à-dire que son évolution est régie par ses propres lois

internes, sans influence externe. Un exemple concret de système pouvant être modélisé par un réseau de Petri autonome est la succession des quatre saisons.

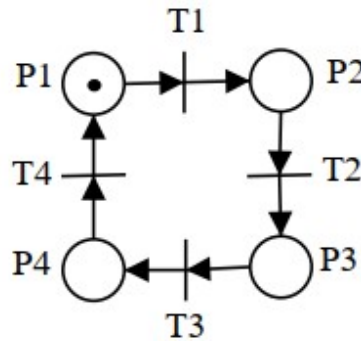


FIGURE 22 – succession des quatre saisons.

Réseau de petri non autonome : Un réseau de Petri non autonome décrit le fonctionnement d'un système dont l'évolution est conditionné par des événements externes ou par le temps. Un réseau de Petri non autonome est synchronisé et/ou temporisé.

1.5 Quelques propriétés des RdP :

Les réseaux de Petri possèdent un certain nombre de propriétés (accessibilité, état d'accueil, la bornitude et la vivacité), dépendent du marquage initial du système et sont liées à l'évolution du réseau. Elles permettent d'apporter des réponses aux questions, concernant l'accessibilité d'un marquage particulier, la bornitude, la vivacité. [3]

Accessibilité : Un marquage M_f est accessible à partir du marquage initial M_0 s'il existe une séquence de franchissement S telle que : du franchissement de cette séquence à partir de M_0 , résulte un nouveau marquage M_f et on note : $M_0[S > M_f]$ Cette propriété permet de savoir si un état non désiré risque de se produire.

Bornitude : Une place P_i est dite bornée pour un marquage initial M_0 si pour tout marquage accessible à partir de M_0 , le nombre de marques dans P_i est fini. Un RdP est dit borné si toutes ses places sont bornées.

Blocage : Un blocage (ou état puits) est un marquage tel qu'aucune transition n'est validée.

Vivacité : Une transition T_j est vivante pour un marquage initial M_0 si pour tout marquage accessible $M_i \in {}^*M_0$, il existe une séquence de franchissement S qui contient T_j à partir de M_i (c-à-d il subsistera toujours une possibilité de franchir T_j).

Exemple :

Sur l'exemple suivant les transitions T_2 et T_3 sont vivantes alors que T_1 ne l'est pas. En effet, elle est franchissable uniquement au démarrage.

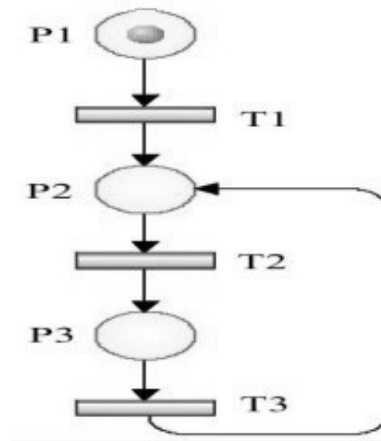


FIGURE 23 – Exemple sur une transition non vivante

un réseau de Petri est dit vivant pour un marquage initial M_0 si pour tout marquage M accessible à partir de M_0 et pour toute transition T_i , il existe une séquence de franchissement S qui inclut la transition T_i .

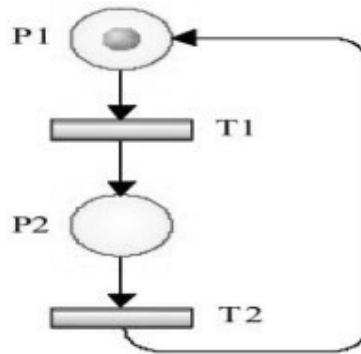


FIGURE 24 – RDP vivant

Cette propriété permet de déduire si le système ne comporte pas de blocage.

Quasi-vivacité : Une transition T_j est quasi vivante pour un marquage initial M_0 , s'il existe une séquence de franchissement qui contient T_j à partir de M_0 . Un RdP est quasi vivant si toutes ses transitions sont quasi vivantes.

Etat d'accueil et réseau de Petri réinitialisable : Un réseau de Petri possède un état d'accueil M_a pour un marquage initial M_0 si pour tout marquage accessible $M \in R(M)$, il existe une séquence de franchissement γ tel que : $M [\gamma > M_a]$

Conflits : Un conflit structurel est défini comme l'existence d'une place qui a au moins deux transitions en sortie. Un conflit effectif est l'existence d'un conflit structurel et d'un marquage tels que le nombre de marques dans la place est strictement inférieur au nombre des transitions en sortie de cette place validées par le marquage.

Lorsqu'un conflit effectif se produit, il est nécessaire de choisir la transition qui va être effectivement franchie.

1.6 Réseaux de Petri particuliers :

Grphe d'état : Un réseau de Petri est un graphe d'état si et seulement si toute transition a exactement une seule place d'entrée et une seule place de sortie. Comme illustrais dans la figure si dessus

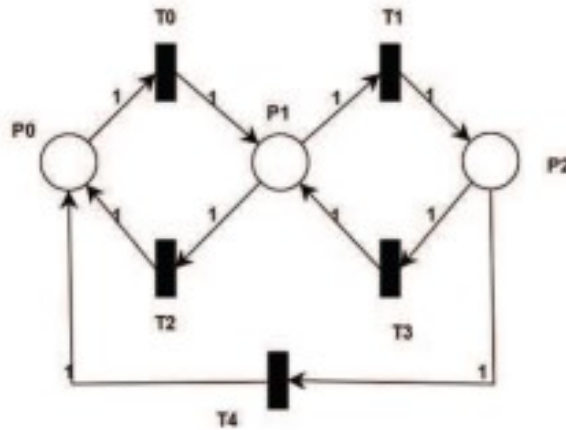


FIGURE 25 – Graphe d'état

Grphe d'événements : Un Réseau de Petri est un graphe d'événement si et seulement si chaque place possède exactement une seule transition d'entrée et une seule transition de sortie.

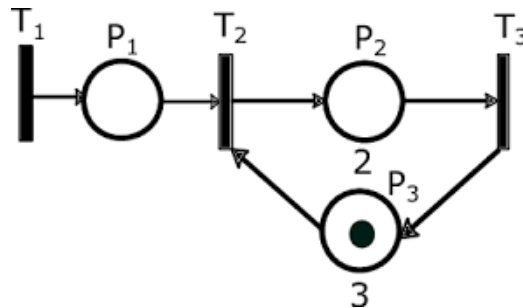


FIGURE 26 – Graphe d'événements

Réseau de Petri avec et sans conflits : Un réseau de Petri sans conflit est un réseau dans lequel chaque place a au plus une transition de sortie.

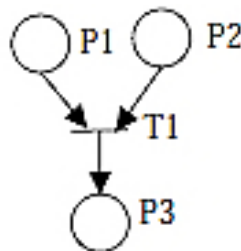


FIGURE 27 – Réseau de Petri sans conflits

Un réseau de Petri avec conflit est un réseau qui possède donc une place avec au moins deux transitions de sorties.

Un conflit est noté : $[P_i, T_1, T_2, \dots, T_n]$; avec T_1, T_2, \dots, T_n : étant les transitions de sorties de la place P_i .

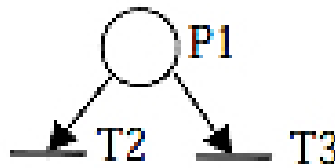


FIGURE 28 – Réseau de Petri avec conflits

RDP à choix libre : Un RDP est à choix libre est un réseau dans lequel pour tout conflit $[P_i, T_1, T_2, \dots, T_n]$ aucune des transitions T_1, T_2, \dots, T_n ne possède aucune autre place d'entrée que P_i . Comme indiqué dans la figure ci-dessous.

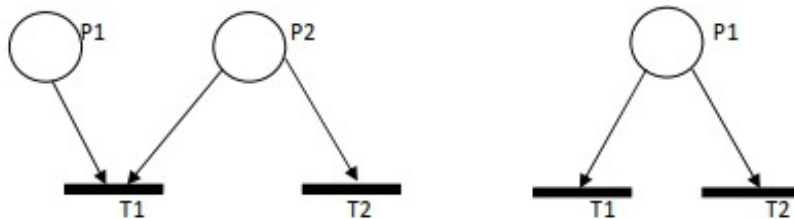


FIGURE 29 – RDP sans choix libre(gauche), RDP avec choix libre(droite)

Réseau de Petri simple : Un réseau de Petri simple est un réseau de petri dans lequel chaque transition ne peut être concernée que par un conflit au plus.

Réseau de petri pur : Un réseau de Petri pur est un réseau dans lequel il n'existe pas de transition ayant une place d'entrée qui soit à la fois place de sortie de cette transition.

RDP à capacités : Un RDP à capacités est un RDP dans lequel des capacités (nombre entiers strictement positifs) sont associées aux places. Le franchissement d'une transition d'entrée d'une place P_i dont la capacité est $cap(P_i)$ n'est possible que si le franchissement ne conduit pas à un nombre de jetons dans P_i qui est plus grand que $cap(P_i)$.

Exemple :

le franchissement de T_1 conduit à 3 jetons dans P_2 d'où T_1 ne peut plus être franchie.

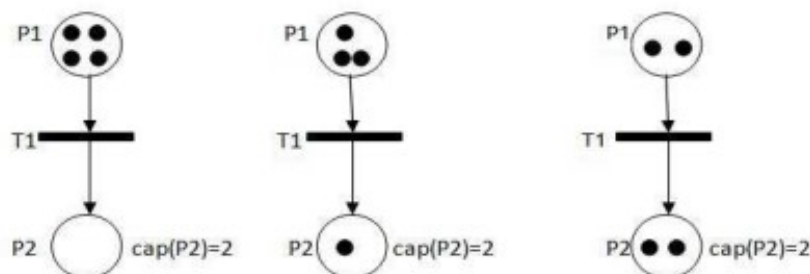


FIGURE 30 – Exemple d'un RDP à capacités

Réseaux de Petri P-temporisés : Pour chaque couple de transition T_i, T_j on note p_{ji} la place qui relie la transition t_i à t_j . Si cette place p_{ji} existe, la temporisation (en valeur rationnelle positive) correspondante est

notée t_{ji} et son marquage noté m_{ji} .
 par : $\tau^{max} = \max\{\tau_{ij}\}_{ij/p_{ij} \in P}$

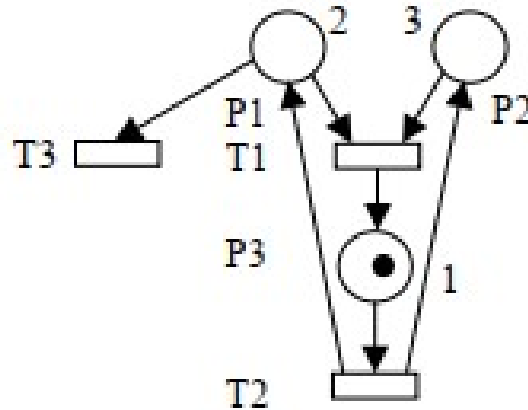


FIGURE 31 – Réseaux de Petri P-temporisés.

Réseaux de Petri T-temporisés : Pour chaque couple de place p_i, p_j on note T_{ji} , la transition qui relie la place p_i à p_j . si cette transition T_{ji} existe, la temporisation (en valeur rationnelle positive) correspondante est notée τ_{ji} .
 La plus grande temporisation du graphe d'évènement considéré est notée τ^{max} , définie par :
 $\tau^{max} = \max\{\tau_{ij}\}_{ij/t_{ij} \in \tau}$.

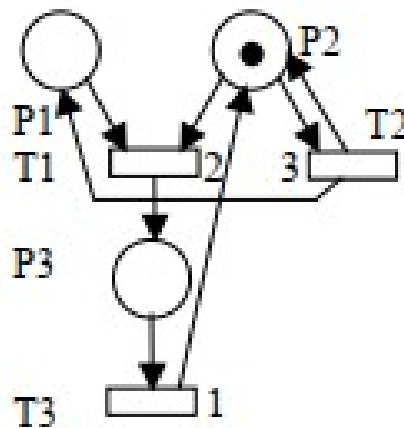


FIGURE 32 – Réseaux de Petri T-temporisés.

1.7 Éléments de modélisation

Parallélisme : Le parallélisme représente la possibilité que plusieurs processus évoluent simultanément au sein du même système.

Synchronisation : La synchronisation ou rendez-vous permet de synchroniser les opérations de deux processus.

Partage de ressources : Cette structure va modéliser le fait qu'au sein du même système plusieurs processus partagent une même ressource, Comme le montre l'exemple ci-dessous.

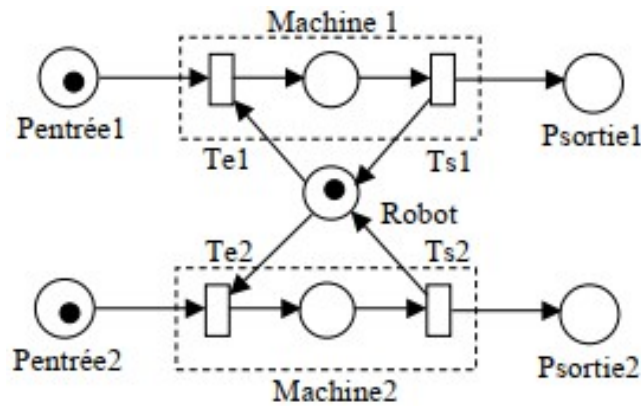


FIGURE 33 – deux machines en concurrence.

Le franchissement d’une transition qui a une place en entrée (ressource partagée) et plusieurs places en sorties validera une seule place et exclura les autres automatiquement.

1.8 théorie des graphes :

1.9 Méthodes d’analyse et validation des réseaux de Pétri :

La modélisation d’un système doit permettre l’analyse de ses propriétés. Les réseaux de pétri offrent des techniques d’analyse puissantes pour valider des modèles de comportement de systèmes à événements discrets. Parmi ces techniques, nous citons le graphe de marquages, l’équation de matrice et la réduction des réseaux de pétri.

Le graphe de marquage :

Introduction à la notion de marquage accessible :

L’ensemble des marquages est l’ensemble M_i qui peuvent être atteint par le franchissement d’une S à partir du marquage initial M_0 . * $M_0 = M_i$ tel que $M_i[s \rightarrow M_j]$, Comme il est représenté par l’exemple suivant :

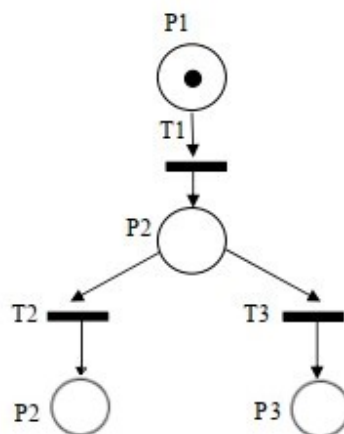


FIGURE 34 – Ensemble de marquage.

L’ensemble des marquages accessible est :

* $M_0 = M_0 \ M_1 \ M_2 \ M_3$ avec $M_0 = [1 \ 0 \ 0 \ 0]$; $M_1 = [0 \ 1 \ 0 \ 0]$; $M_2 = [0 \ 0 \ 1 \ 0]$; $M_3 = [0 \ 0 \ 0 \ 1]$.

Graphe de marquage accessible :

On utilise le graphe de marquage quand le nombre de marquages accessibles est fini.

Exemple :

dans cet exemple nous montrons le marquage accessible et son graphe correspondant.

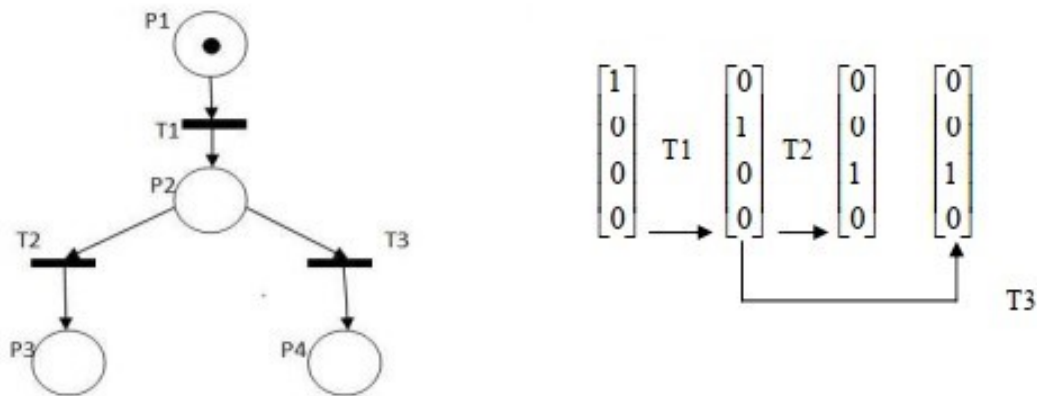


FIGURE 35 – Graphe de marquage accessible.

Déduction des propriétés et Validation du RdP :

Une fois que tous les marquages du réseau ont été construits, les propriétés peuvent ensuite être déduites grâce à des techniques de la théorie des graphes. On peut alors déduire les propriétés du système, telles que la vivacité.

En utilisant ces informations, on peut valider le RdP en vérifiant s'il satisfait les propriétés spécifiées et s'il fonctionne correctement selon les spécifications données.

L'équation de matrice :

Cette méthode consiste à déterminer une représentation matricielle du réseau, où l'utilisation des matrices pré-incidence et post-incidence est nécessaire pour obtenir la matrice d'incidence, qui représente le RdP de manière matricielle. Ensuite grâce aux techniques d'algèbre linéaire, on peut obtenir les propriétés du réseau.

Matrice Post-incidence :

C'est une matrice nm telle que $post(i,j) = W(t_j, p_i)$ avec n le nombre de places et m le nombre de transitions dans le RdP, contient le nombre de marques déposées dans p_i lors du franchissement de la transition t_j .

Exemple :

Un RdP avec sa matrice post-incidence est représenté dans la figure ci-dessous.

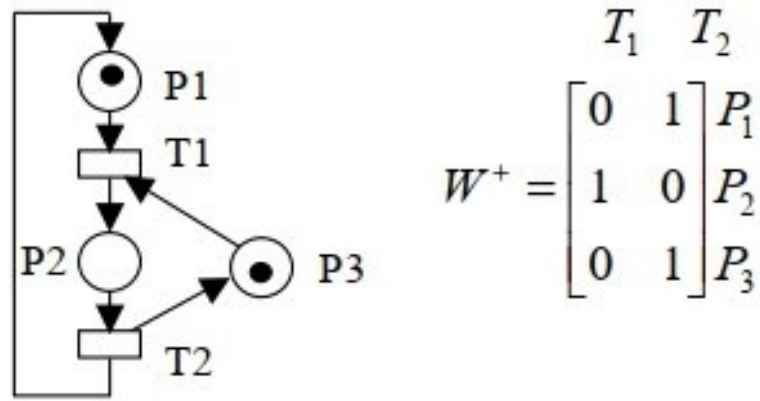


FIGURE 36 – RDP avec sa matrice post-incidence.

Matrice Pré-incidence :

C'est la matrice mn à coefficients dans \mathbb{N} telle que $\text{pré}(i,j) = W(p_i, t_j)$ indique le nombre de marques que doit contenir la place p_i pour que la transition t_j devienne franchissable. n le nombre de places et m le nombre de transitions dans le RdP.

Exemple :

Un RDP avec sa matrice pré-incidence est représenté dans la figure ci-dessous.

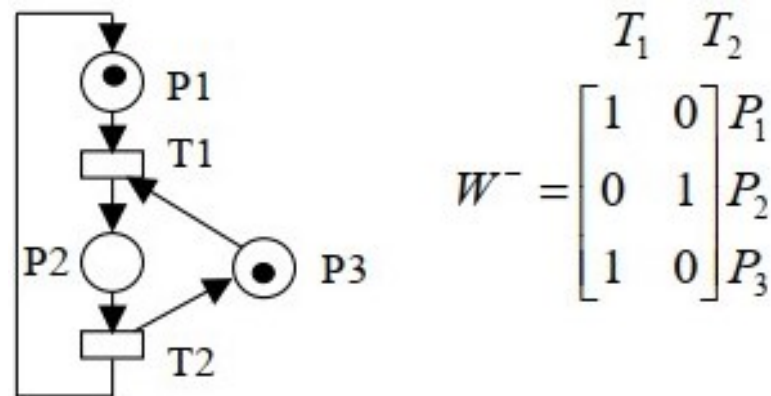


FIGURE 37 – RDP avec sa matrice pré-incidence.

Matrice d'incidence :

C'est une matrice à n ligne et m colonnes avec n le nombre de places et m le nombre de transitions dans le RdP. Elle peut être calculée comme suit :

$$W = W^+ - W^-$$

Exemple :

En reprenant le réseau de Petri utilisé dans l'exemple précédent, voici ce que l'on obtient :

$$W = W^+ - W^- = \begin{matrix} & T_1 & T_2 \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \end{matrix} & \begin{bmatrix} -1 & 1 \\ 1 & -1 \\ -1 & 1 \end{bmatrix} \end{matrix}$$

Cette matrice fait le bilan de l'action de toutes les liaisons du RdP sur le marquage. Il faut cependant noter que les termes issus des matrices d'incidence avant(Pré) et arrière(post) ne doivent pas se compenser par soustraction. Si tel est le cas, il y a perte d'information lors de la transcription du modèle sous la forme matricielle. Pour s'en affranchir, on devra exclure d'une telle analyse les RdP tels qu'une place et une transition soient reliés par deux arcs, l'un amont et l'autre aval à cette transition. Si cette condition est respectée, le RdP est dit pur. Dans notre cas, le RDP utilisé dans les exemples précédents présente un exemple de RdP impur, en ce sens que la place P2 et la transition T2 sont un exemple de ce qui vient d'être décrit.

La réduction des RdPs :

Pour l'analyse des propriétés d'un RdP de taille significative, l'utilisation du graphe de marquage ou de l'équation de matrice s'avère insuffisante. L'objectif de la technique par réduction est de présenter des règles permettant d'obtenir à partir d'un RdP marqué, un RdP marqué plus simple, avec un nombre réduit de places et de transitions.

2 Modélisation à base de réseaux de pétri

Pour modéliser un système avec un réseau de Petri, on définit les places comme des états du système, les transitions comme des événements déclencheurs, et les arcs comme liaisons entre les places et les transitions, permettant ainsi une représentation graphique et formelle des interactions dynamiques du système. [5]

2.1 Réseaux de petri de la station de tri par hauteur (station avancée)

Dans cette station, nous avons opté pour un mode de tri par lot de trois pièces successives, assurant un processus continu sans interruption, contrairement au tri pièce par pièce où le système s'arrêterait jusqu'à l'évacuation de celle-ci. Pour rendre cela possible, nous allons diviser le modèle en deux sous-modèles distincts : le premier sera dédié au capteur de pièce et le deuxième au processus de triage des pièces.

2.2 Réseaux de petri du capteur de pièce

l'arbre de probabilité :

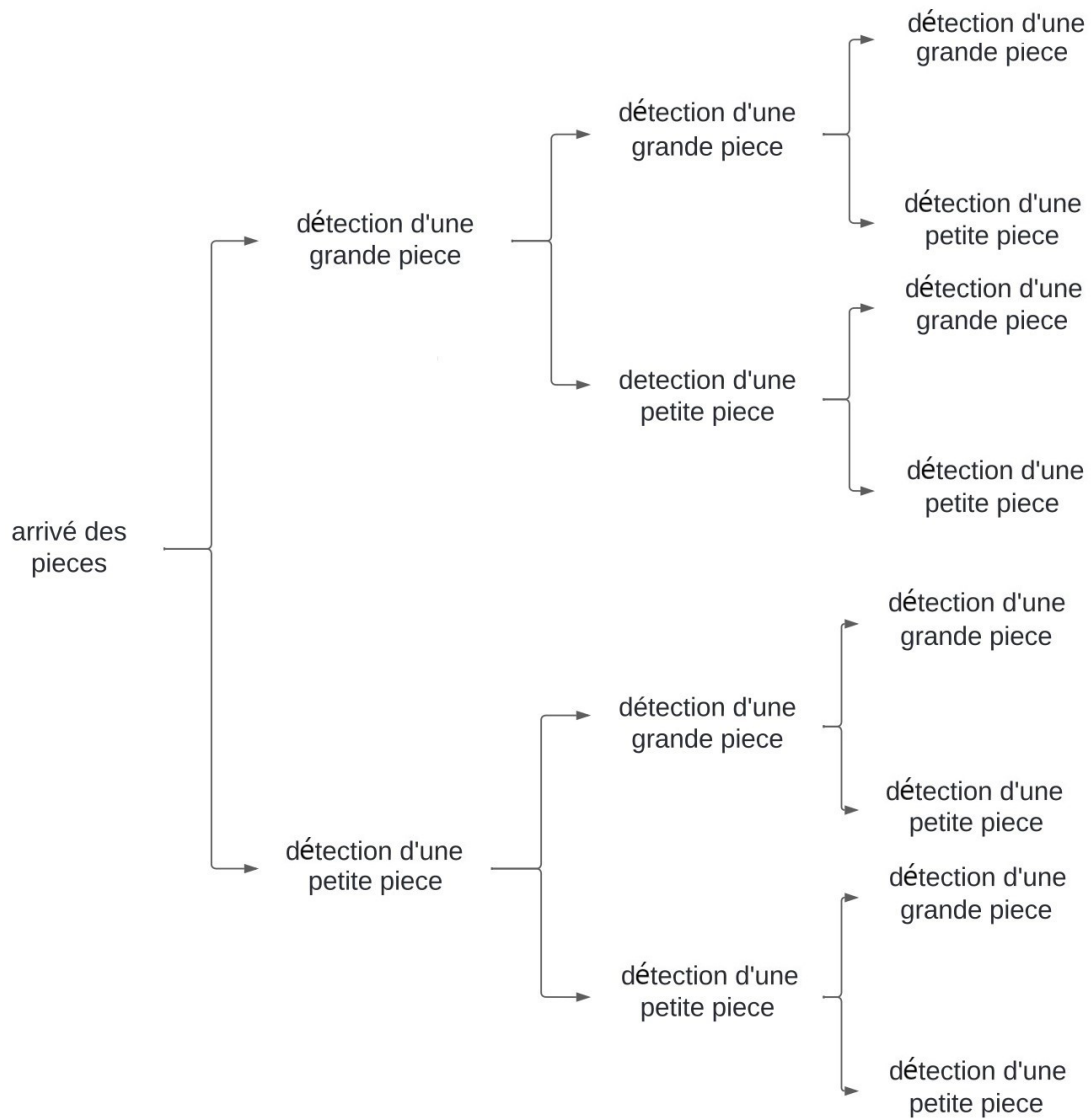


FIGURE 38 – arbre de probabilité

Nous avons utilisé cet arbre de probabilité pour modéliser toutes les possibilités de boîte détectées par le capteur de la station de tri, Pour éviter toute perte d’information, étant donné que le capteur est éloigné du lieu où le tri est effectué.

Dans un Réseau de Petri (RdP), chaque détection de pièce est consignée dans une place spécifique du sous-modèle, permettant ainsi de suivre précisément le processus de détection.

Ensuite, ces données seront exploitées par l’autre sous-modèle afin de récupérer les informations pertinentes.

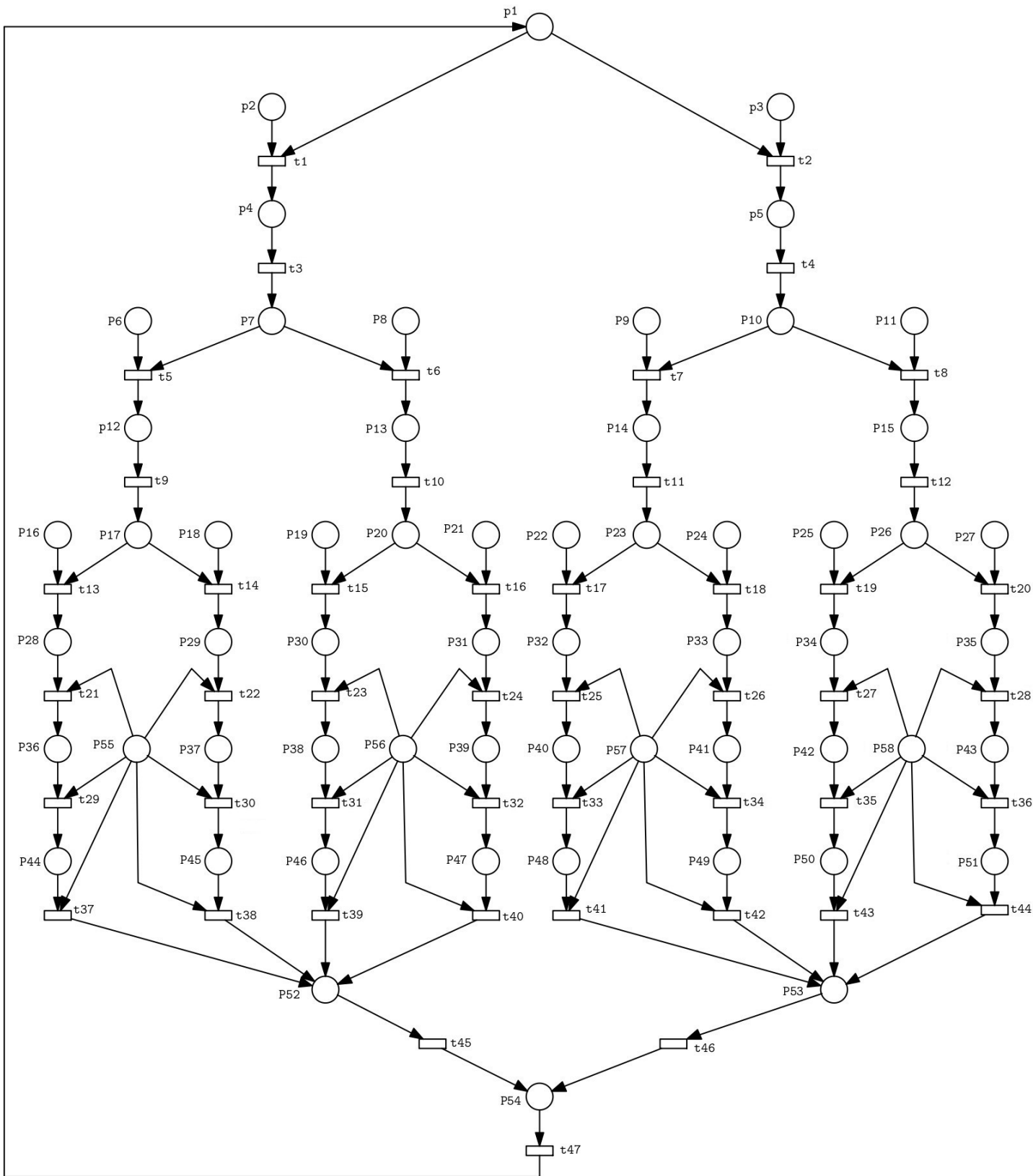


FIGURE 39 – RDP du 1er sous modèle

2.3 Nomenclature des places et transitions

Places :

place	description
P1	Cette place est marquée par un jeton lors du démarrage du deuxième sous-modèle.
P2	Présence d'une pièce de grande taille.
P3	Présence d'une pièce de petite taille.
P4	Cette place représente l'attente de la pièce détectée jusqu'à son passage.
P5	Cette place représente l'attente de la pièce détectée jusqu'à son passage.
P6	détection d'une pièce de grande taille.
P7	Place de divergence en OU (OR)
P8	Présence d'une pièce de petite taille.
P9	Présence d'une pièce de grande taille.
P10	Place de divergence en OU (OR)
P11	Présence d'une pièce de petite taille.
P12	Cette place représente l'attente de la pièce détectée jusqu'à son passage.
P13	Cette place représente l'attente de la pièce détectée jusqu'à son passage.
P14	Cette place représente l'attente de la pièce détectée jusqu'à son passage.
P15	Cette place représente l'attente de la pièce détectée jusqu'à son passage.
P16	Présence d'une pièce de grande taille.
P17	Place de divergence en OU (OR)
P18	Présence d'une pièce de petite taille.
P19	Présence d'une pièce de grande taille.
P20	Place de divergence en OU (OR)
P21	Présence d'une pièce de petite taille.
P22	Présence d'une pièce de grande taille.
P23	Place de divergence en OU (OR)
P24	Présence d'une pièce de petite taille.
P25	Présence d'une pièce de grande taille.
P26	Place de divergence en OU (OR)
P27	Présence d'une pièce de petite taille.
P28	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de grande taille.
P29	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de grande taille.
P30	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de grande taille.
P31	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de grande taille.
P32	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de petite taille.
P33	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de petite taille.
P34	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de petite taille.
P35	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de petite taille.
P36	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de grande taille.
P37	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de grande taille.
P38	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de petite taille.

P39	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de petite taille.
P40	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de grande taille.
P41	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de grande taille.
P42	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de petite taille.
P43	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de petite taille.
P44	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de grande taille.
P45	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de petite taille.
P46	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de grande taille.
P47	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de petite taille.
P48	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de grande taille.
P49	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de petite taille.
P50	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de grande taille.
P51	Cette place est chargée de transmettre l'information au deuxième sous-modèle signalant la détection d'une pièce de petite taille.
P52	Place de convergence en OU (OR)
P53	Place de convergence en OU (OR)
P54	Place de convergence en OU (OR)
P55	Cette place indique que l'évacuation d'une pièce a été effectuée dans le deuxième sous-modèle
P56	Cette place indique que l'évacuation d'une pièce a été effectuée dans le deuxième sous-modèle
P57	Cette place indique que l'évacuation d'une pièce a été effectuée dans le deuxième sous-modèle
P58	Cette place indique que l'évacuation d'une pièce a été effectuée dans le deuxième sous-modèle

TABLE 10 – Place du 1er sous-modèle

Transitions :

transition	description
T1	Détection d'une piece de grande taille.
T2	Détection d'une piece de petite taille.
T3	Transition de divergence en OU (OR)
T4	Transition de divergence en OU (OR)
T5	Détection d'une piece de grande taille.
T6	Détection d'une piece de petite taille.
T7	Détection d'une piece de grande taille.
T8	Détection d'une piece de petite taille.
T9	Transition de divergence en OU (OR)
T10	Transition de divergence en OU (OR)
T11	Transition de divergence en OU (OR)
T12	Transition de divergence en OU (OR)
T13	Détection d'une piece de grande taille.
T14	Détection d'une piece de petite taille.
T15	Détection d'une piece de grande taille.
T16	Détection d'une piece de petite taille.
T17	Détection d'une piece de grande taille.
T18	Détection d'une piece de petite taille.
T19	Détection d'une piece de grande taille.
T20	Détection d'une piece de petite taille.
T21->T44	une évacuation d'une pièce a eu lieu dans le deuxieme sous-modèle.
T45	Transition de convergence en OU (OR)
T46	Transition de convergence en OU (OR)
T47	Transition de réinitialisation (retour)

TABLE 11 – Transition du 1er sous-modèle

2.4 resaux de petri du processus de tri

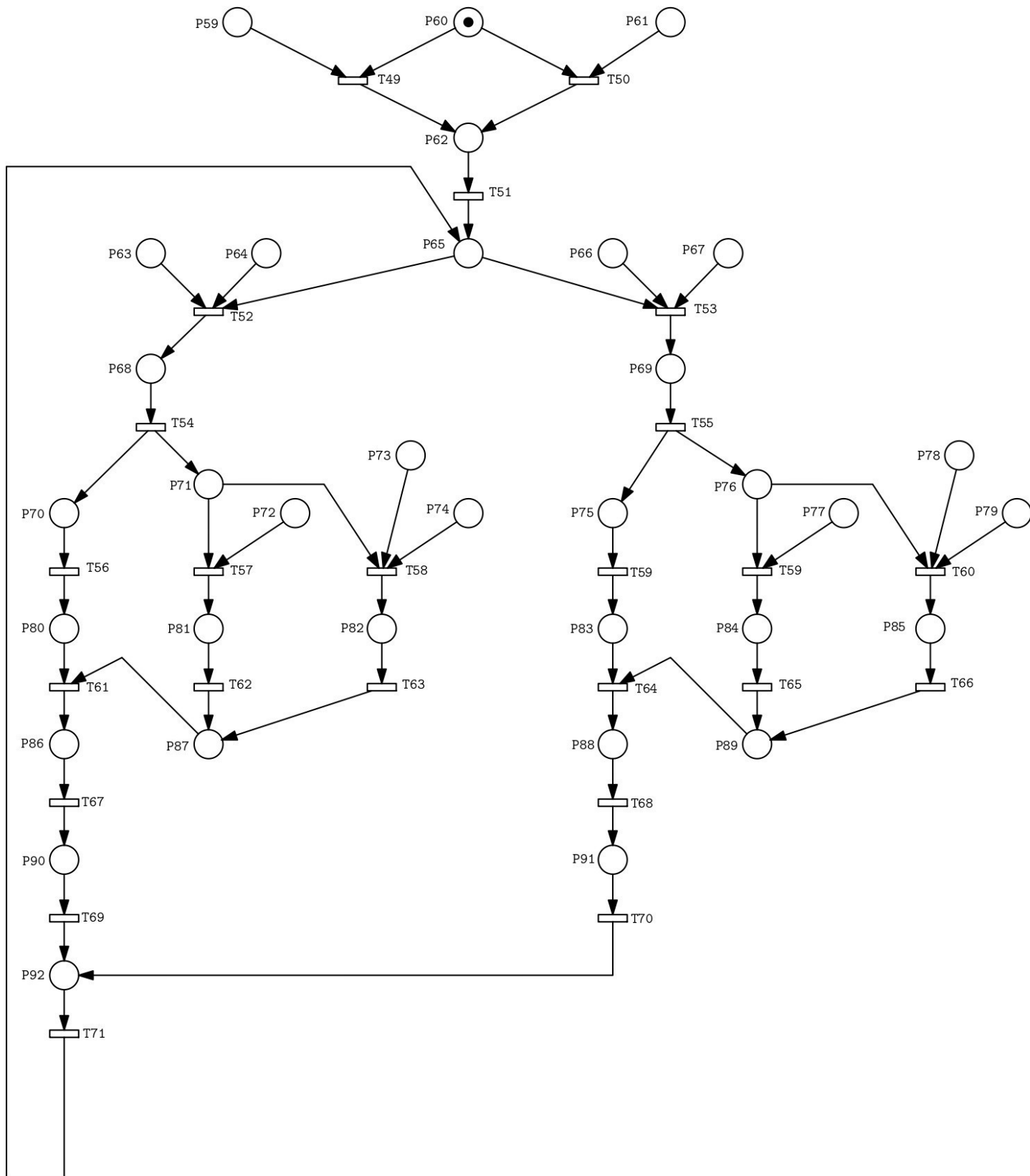


FIGURE 40 – RDP du 2eme sous-modèle

2.5 Nomenclature des places et transitions

Places :

place	description
P59	détection d'une impulsion sur le bouton poussoir "start"
P60	réinitialisation de tous les actionneurs
P61	Effectuer une impulsion en appuyant sur le bouton "start_sup"
P62	Activation du convoyeur d'approvisionnement "feeder convoyeur"
P63	Présence d'une pièce de petite taille.
P64	Présence d'une pièce près de la table tournante.
P65	Activation du convoyeur "entry convoyeur"
P66	Présence d'une pièce de grande taille.
P67	Présence d'une pièce près de la table tournante.
P68	Activation de l'action de chargement "Load"
P69	Activation de l'action de chargement "Load"
P70	Désactivation du chargement "Load" et activation de la rotation "turn" de la table tournante
P71	Place de divergence en OU (OR)
P72	Désactivation du chargement "Load" et activation de la rotation "turn" de la table tournante
P73	Place de divergence en OU (OR)
P74	Présence d'une pièce près de la table tournante "turntable".
P75	Présence d'une pièce dans le convoyeur d'entrée "entry convoyeur"
P76	Aucune pièce n'est présente près de la table tournante.
P77	Présence d'une pièce près de la table tournante "turntable".
P78	Présence d'une pièce dans le convoyeur d'entrée "entry convoyeur"
P79	Aucune pièce n'est présente près de la table tournante.
P80	Activation de la rotation "turn" de la table tournante et activation du chargement "Load".
P81	Désactivation du convoyeur d'entrée "entry convoyeur"
P82	Activation du convoyeur d'entrée "entry convoyeur"
P83	Activation du déchargement "Unload".
P84	Désactivation du convoyeur d'entrée "entry convoyeur"
P85	Activation du convoyeur d'entrée "entry convoyeur"
P86	Place d'attente
P87	Place de convergence en OU (OR)
P88	Place d'attente
P89	Place de convergence en OU (OR)
P90	Désactivation du chargement "Load" et désactivation de la rotation "turn" de la table tournante
P91	Désactivation du déchargement "Unload" et désactivation de la rotation "turn" de la table tournante
P92	Place de convergence en OU (OR)

TABLE 12 – Place du 2eme sous modèle

Transitions :

transition	description
T48	relie la place 62 (2eme sous-modèle) avec la place 1 (1er sous-modèle)
T49	Le bouton poussoir "Start"est enclenché
T50	Le bouton poussoir de supervision "Start_sup" est enclenché
T51	détection d'une piece de petite taille (retour d'infomation depuis 1er sous-modèle) et le capteur "at turn table entry" enclenché
T52	détection d'une piece de grande taille (retour d'infomation depuis 1er sous-modèle) et le capteur "at turn table entry" enclenché
T53	Transition de divergence en ET (AND) et franchissable lorsque le capteur "at front" est enclenché
T54	Transition de divergence en ET (AND) et franchissable lorsque le capteur "at front" est enclenché
T55	capteur "at unload position" enclenché
T56	capteur "at turntable entry" enclenché
T57	capteur "at front" enclenché et le capteur "at turntable entry" dans un état de non-détection
T58	capteur "at unload position" enclenché
T59	capteur "at turntable entry" enclenché
T60	capteur "at front" enclenché et le capteur "at turntable entry" dans un état de non-détection
T61	Transition de convergence en ET (AND)
T62	capteur "entry convoy" dans un état de non-détection
T63	capteur "entry convoy" enclenché
T64	Transition de convergence en ET (AND)
T65	capteur "entry convoy" dans un état de non-détection
T66	capteur "entry convoy" enclenché
T67	capteur "at left entry"
T68	capteur "at right entry"
T69	La rotation de la table tournante est désactivée
T70	La rotation de la table tournante est désactivée
T71	le capteur "at load position" enclenché

TABLE 13 – Transition du 2eme sous-modèle

2.6 Réseaux de petri de la station de tri par hauteur (station basique)

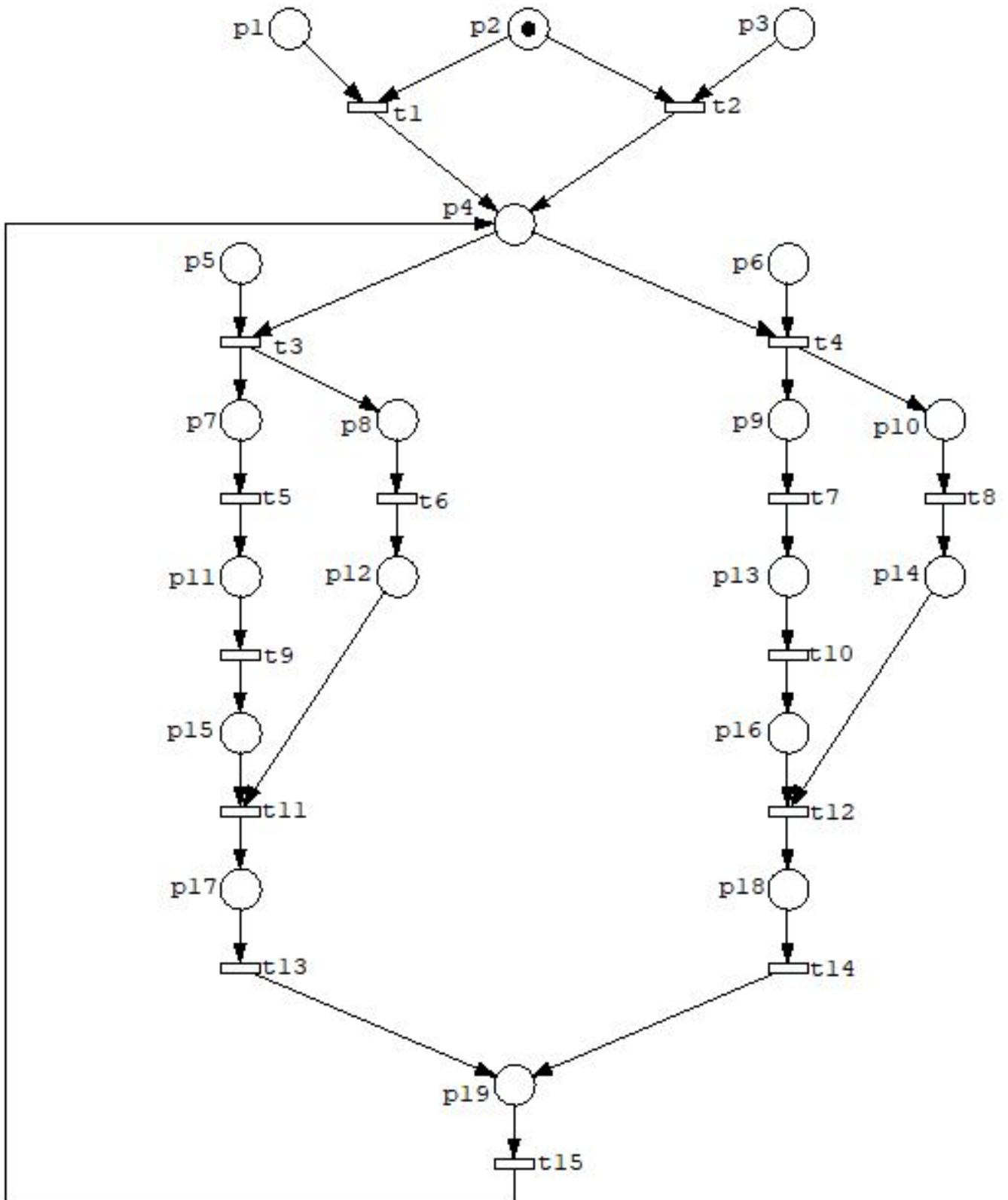


FIGURE 41 – Réseaux de petri de la station de tri "sorting by height (basic)"

2.7 Nomenclature des places et transitions

Places :

place	description
P1	détection d'une impulsion sur le bouton poussoir "start"
P2	réinitialisation de tous les actionneurs
P3	Effectuer une impulsion en appuyant sur le bouton "start_sup"
P4	activation du convoyeur d'entrée "convoyer entry"
P5	présence d'une petite piece
P6	présence d'une grande piece
P7	Activation du transporteur à chaîne dans la direction des flèches (en ligne droite) "Load"
P8	place d'attente
P9	Activation du transporteur à chaîne dans la direction des flèches (en ligne droite) "Load"
P10	place d'attente
P11	désactivation du transporteur à chaîne dans la direction des flèches"load" et activation dans la direction gauche "transfr left"
P12	désactivation du convoyeur d'entrée "convoyer entry"
P13	désactivation du transporteur à chaîne dans la direction des flèches"load" et activation dans la direction droite "transfr right"
P14	désactivation du convoyeur d'entrée "convoyer entry"
P15	place d'attente
P16	place d'attente
P17	desactivation du transporteur à chaîne "transfr left"
P18	desactivation du transporteur à chaîne "transfr right"
P19	place de convergence en OU (OR)

TABLE 14 – Places du RDP

Transitions :

transition	description
T1	Le bouton poussoir "Start"est enclenché
T2	Le bouton poussoir de supervision "Start_sup" est enclenché
T3	détection d'une petite piece (capteur "low sensor" enclenché)
T4	détection d'une grande piece (capteur "low sensor" enclenché)
T5	détection d'une piece sur le transporteur à chaîne (capteur "loaded" enclenché)
T6	le capteur "low sensor" ne detecte aucune piece
T7	détection d'une piece sur le transporteur à chaîne (capteur "loaded" enclenché)
T8	le capteur "high sensor" ne detecte aucune piece
T9	capteur "at left entry" enclaché
T10	capteur "at right entry" enclaché
T11	capteur "at left entry" ne detecte aucune piece
T12	capteur "at right entry" ne detecte aucune piece
T13	transporteur à chaîne désactivé
T14	transporteur à chaîne désactivé
T15	transition de convergence en OU (OR)

TABLE 15 – Transitions du RDP

2.8 Réseaux de petri de la station de tri par poids

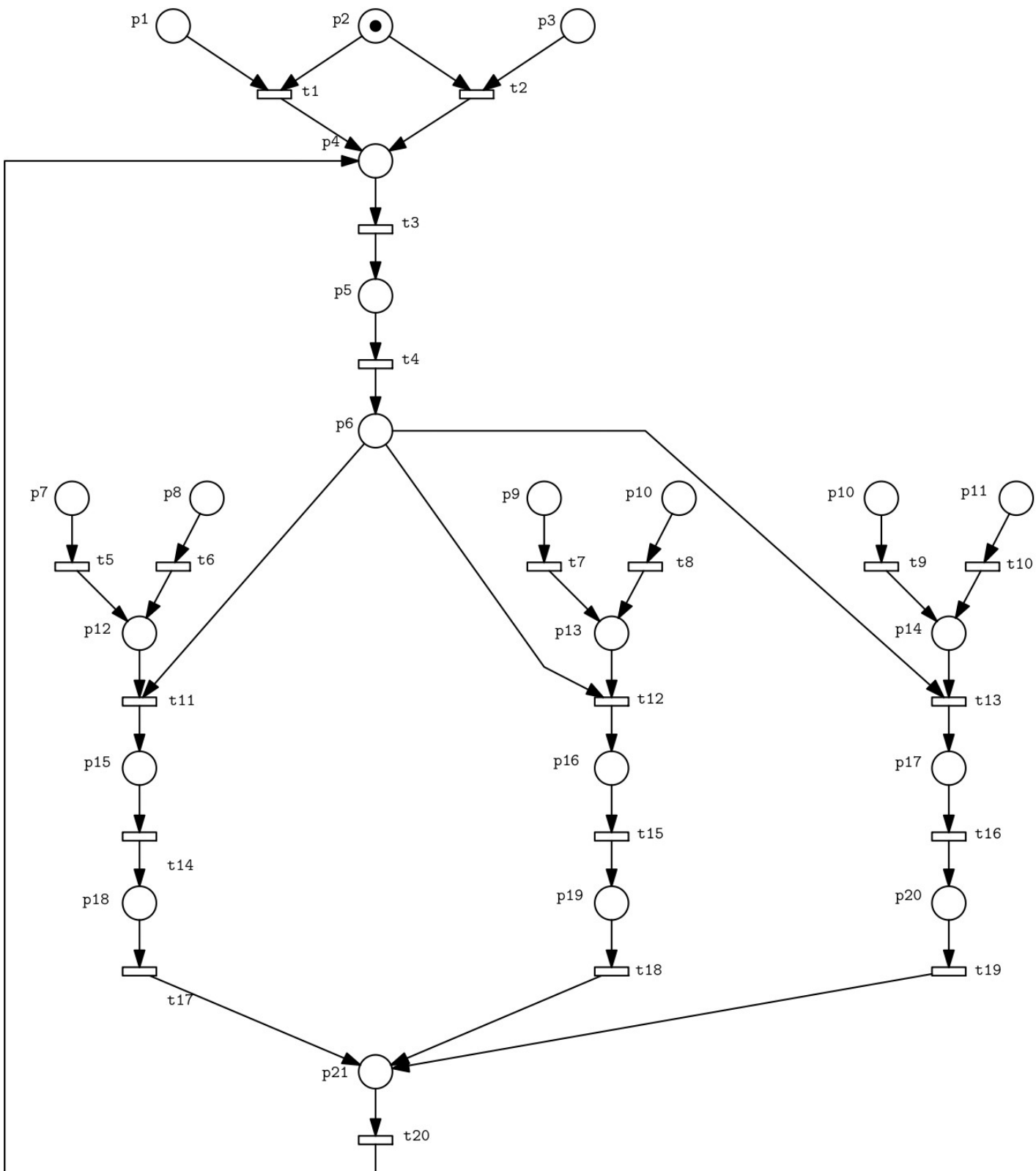


FIGURE 42 – Réseaux de petri de la station de tri "sorting by weight"

2.9 Nomenclature des places et transitions

Places :

place	description
P1	détection d'une impulsion sur le bouton poussoir "start"
P2	réinitialisation de tous les actionneurs
P3	Effectuer une impulsion en appuyant sur le bouton "start_sup"
P4	activation du convoyeur d'entrée "entry convoyer"
P5	activation du convoyeurs à pesée intégrée "load scale"
P6	desactivation du convoyeur d'entrée "entry convoyer" et du convoyeurs à pesée intégrée "load scale"
P7	detection d'une piece d'un poids de 4 KG
P8	detection d'une piece d'un poids de 3,5 KG
P9	detection d'une piece d'un poids de 7,5 KG
P10	detection d'une piece d'un poids de 5 KG
P11	detection d'une piece d'un poids de 2,5 KG
P12	detection d'une piece d'un poids de 1,5 KG
P13	place de convergence en ET (AND)
P14	place de convergence en ET (AND)
P15	place de convergence en ET (AND)
P16	activation du convoyeurs à pesée intégrée "load scale" et du trieuse à roue escamotable ("send forward" et "send left")
P17	activation du convoyeurs à pesée intégrée "load scale" et du trieuse à roue escamotable "send forward"
P18	activation du convoyeurs à pesée intégrée "load scale" et du trieuse à roue escamotable ("send forward" et "send right")
P19	desactivation du trieuse à roue escamotable
P20	desactivation du trieuse à roue escamotable
P21	desactivation du trieuse à roue escamotable
P22	desactivation du convoyeurs à pesée intégrée "load scale"

TABLE 16 – place du RDP

Transitions :

transition	description
T1	Le bouton poussoir "Start"est enclenché
T2	Le bouton poussoir de supervision "Start_sup" est enclenché
T3	capteur "at scale" enclenché
T4	capteur "at scale" enclenché
T5	detection d'une piece d'un poids de 4 KG
T6	detection d'une piece d'un poids de 3,5 KG
T7	detection d'une piece d'un poids de 7,5 KG
T8	detection d'une piece d'un poids de 5 KG
T9	detection d'une piece d'un poids de 2,5 KG
T10	detection d'une piece d'un poids de 1,5 KG
T11	detection d'une piece d'un poids de 4 KG ou 3,5 KG
T12	detection d'une piece d'un poids de 7,5 KG ou 5 KG
T13	detection d'une piece d'un poids de 2,5 KG ou 2,5 KG
T14	capteur "at left entry" enclenché
T15	capteur "at forward entry" enclenché
T16	capteur "at right entry" enclenché
T17	capteur "at left entry" ne detecte aucune piece
T18	capteur "at forward entry" ne detecte aucune piece
T19	capteur "at right entry" ne detecte aucune piece
T20	trieuse à roue escamotable est desactivé

TABLE 17 – Transition du RDP

2.10 Réseaux de petri de la station de separation par couleur

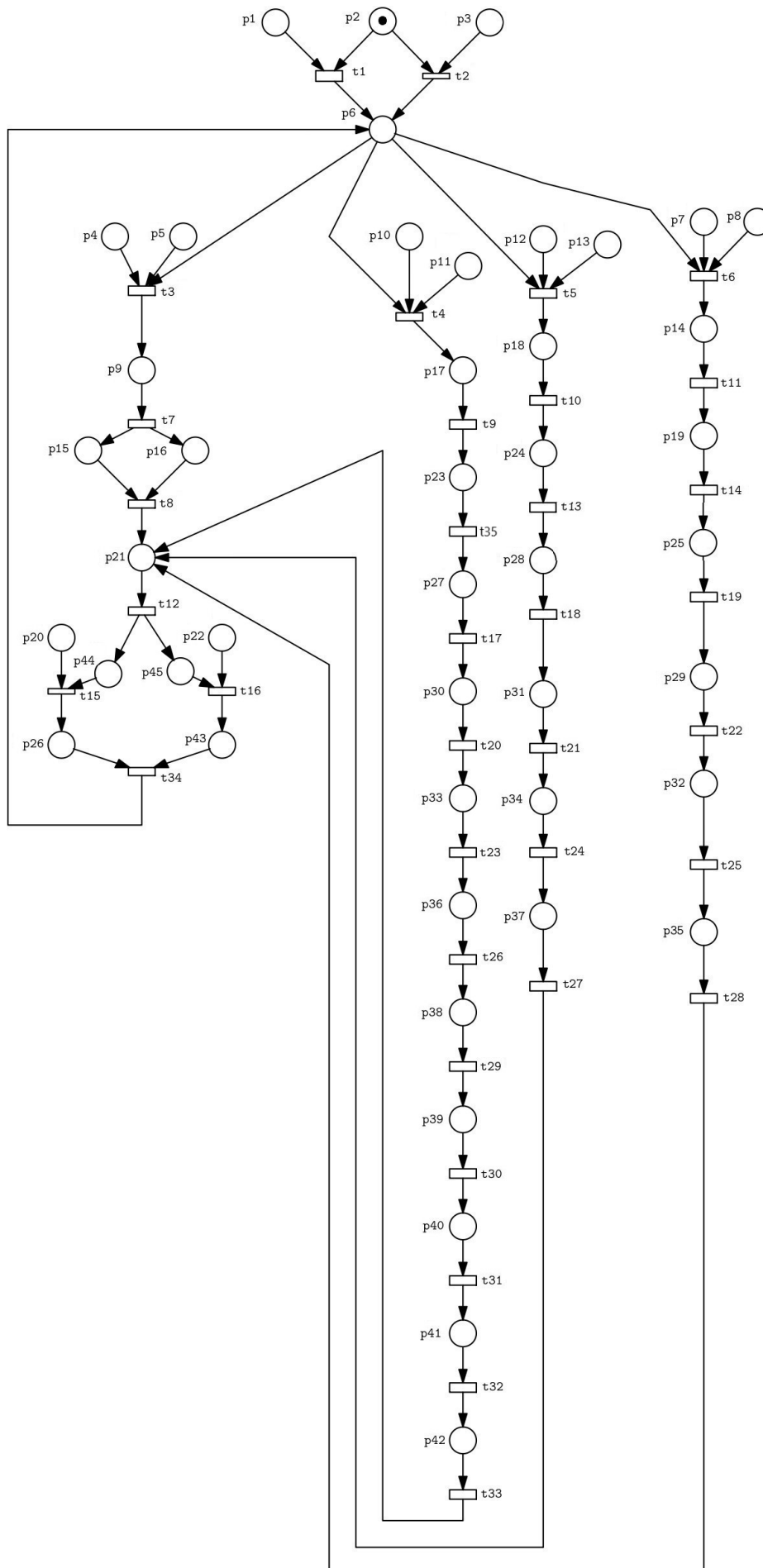


FIGURE 43 – Réseaux de petri de la station de tri "separating station"

2.11 Nomenclature des places et transitions

Places :

place	description
P1	détection d'une impulsion sur le botton poussoir "start"
P2	réinitialisation de tous les actionneurs
P3	Effectuer une impulsion en appuyant sur le bouton "start_sup"
P4	detection d'une piece vert dans le convoyeur d'entré numero 2 "entry convoyer 2"
P5	detection d'une piece blue dans le convoyeur d'entré numero 1 "entry convoyer 1"
P6	activation du convoyeur d'entré (entry convoyer) numero 1 et numero 2
P7	detection d'une piece vert dans le convoyeur d'entré 1
P8	detection d'une piece vert dans le convoyeur d'entré 2
P9	activation du convoyeur 1 et 2
P10	detection d'une piece blue dans le convoyeur d'entré 2
P11	detection d'une piece vert dans le convoyeur d'entré 1
P12	detection d'une piece blue dans le convoyeur d'entré 2
P13	detection d'une piece blue dans le convoyeur d'entré 1
P14	désactivation du convoyeur d'entré 2
P15	Attendre l'évacuation de la pièce
P16	Attendre l'évacuation de la pièce
P17	désactivation du convoyeur d'entré 1
P18	désactivation du convoyeur d'entré 1
P19	activation du convoyeur 1
P20	detection d'une piece quelconque dans le convoyeur d'entré 1
P21	Cette étape permet d'établir une convergence en ET (AND)
P22	detection d'une piece quelconque dans le convoyeur d'entré 2
P23	activation du convoyeur 2
P24	activation du convoyeur 2
P25	desactivation du convoyeur 1 et du convoyeur d'entré 1
P26	desactivation du convoyeur d'entré 2
P27	desactivation du convyeur d'entré 2 et du convoyeur 2
P28	desactivation du convyeur d'entré 2 et du convoyeur 2
P29	activation du poussoir pneumatique 1 (pusher 1)
P30	activation du poussoir pneumatique 2 (pusher 2)
P31	activation du poussoir pneumatique 2 (pusher 2)
P32	desactivation du poussoir pneumatique 1 (pusher 1)
P33	desactivation du poussoir pneumatique 2 (pusher 2)
P34	desactivation du poussoir pneumatique 2 (pusher 2)
P35	activation du convoyeur d'entré 1 et du convoyeur 1 et 2
P36	activation du convoyeur d'entré 1 et convoyeur 2 et desactivation du convoyeur d'entré 2
P37	activation du convoyeur d'entré 2 et du convoyeur 1 et 2
P38	activation du convoyeur 1
P39	desactivation du convoyeur 1 et du convoyeur d'entré 1
P40	activation du poussoir pneumatique 1 (pusher 1)
P41	desactivation du poussoir pneumatique 1 (pusher 1)
P42	activation du convoyeur d'entré (1 et 2) et du convoyeur 1
P43	desactivation du convoyeur d'entré 1
P44	Cette étape permet d'établir une convergence en ET (AND)
P45	Cette étape permet d'établir une convergence en ET (AND)

TABLE 18 – Places du RDP

Transitions :

transition	description
T1	Le bouton poussoir “Start”est enclenché
T2	Le bouton poussoir de supervision “Start_sup” est enclenché
T3	detection d’une piece blue dans le convoyeur d’entré 1 et une piece vert dans le convoyeur 2
T4	detection d’une piece vert dans le convoyeur d’entré 1 et une piece blue dans le convoyeur 2
T5	detection d’une piece blue dans le convoyeur d’entré 1 et une piece blue dans le convoyeur 2
T6	detection d’une piece vert dans le convoyeur d’entré 1 et une piece vert dans le convoyeur 2
T7	convoyeur 1 et 2 en marche
T8	convoyeur 1 et 2 en marche
T9	capteur “sensor 2” est enclenché
T10	capteur “sensor 2” est enclenché
T11	capteur “sensor 1” est enclenché
T12	les capteur “at pusher 1 exit” et “at pusher 2 exit” sont enclenché
T13	temporisation de 1.5 seconde
T14	temporisation de 1.5 seconde
T15	detection d’une pièce quelconque (blue ou vert) dans le convoyeur d’entré 2
T16	detection d’une pièce quelconque (blue ou vert) dans le convoyeur d’entré 1
T17	convoyeur 2 a l’arret est le capteur “pusher 2 back” est enclenché
T18	convoyeur 2 a l’arret est le capteur “pusher 2 back” est enclenché
T19	convoyeur 1 a l’arret est le capteur “pusher 1 back” est enclenché
T20	capteur “pusher 2 front” enclenché
T21	capteur “pusher 2 front” enclenché
T22	capteur “pusher 1 front” enclenché
T23	capteur “pusher 2 back” enclenché
T24	capteur “pusher 2 back” enclenché
T25	capteur “pusher 1 back” enclenché
T26	capteur “sensor 1” enclenché
T27	convoyeur 1 et 2 en marche
T28	convoyeur 1 et 2 en marche
T29	temporisation de 1.5 seconde
T30	convoyeur 1 a l’arret et le capteur “pusher 1 back” est enclenché
T31	capteur “pusher 1 front” enclenché
T32	capteur “pusher 1 front” enclenché
T33	convoyeur 1 et 2 en marche
T34	convoyeur d’entré 1 et 2 a l’arret
T35	temporisation de 1.5 seconde

TABLE 19 – transitions du RDP

2.12 Réseaux de petri de la station de tri par couleur

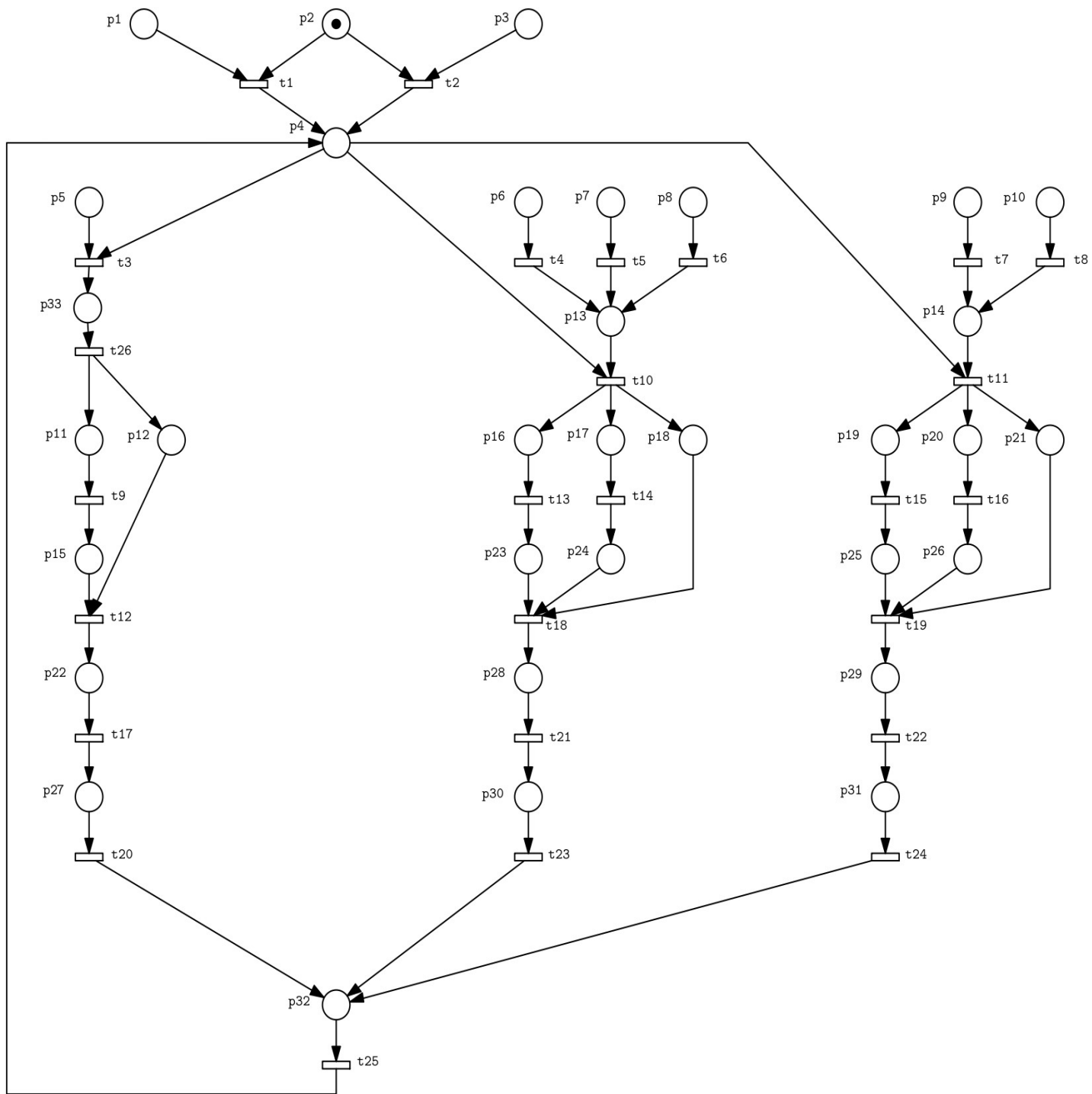


FIGURE 44 – Réseaux de petri de la station de tri "sorting station"

2.13 Nomenclature des places et transitions

Places :

place	description
P1	détection d'une impulsion sur le bouton poussoir "start"
P2	réinitialisation de tous les actionneurs
P3	Effectuer une impulsion en appuyant sur le bouton "start_sup"
P4	activation du convoyeur d'entrée "entry convoyeur"
P5	détection d'une pièce de dimension 1
P6	détection d'une pièce de dimension 4
P7	détection d'une pièce de dimension 6
P8	détection d'une pièce de dimension 5
P9	détection d'une pièce de dimension 3
P10	détection d'une pièce de dimension 2
P11	place d'attente
P12	Activation de la bande transporteuse et Pivotement du 1er bras
P13	activation du convoyeur de sortie "exit convoyeur"
P14	activation du convoyeur de sortie "exit convoyeur"
P15	désactivation du convoyeur d'entrée "entry convoyeur"
P16	place d'attente
P17	place d'attente
P18	Activation de la bande transporteuse et Pivotement du 2eme bras
P19	place d'attente
P20	place d'attente
P21	Activation de la bande transporteuse et Pivotement du 3eme bras
P22	place d'attente
P23	désactivation du convoyeur d'entrée "entry convoyeur"
P24	activation de la lame d'arrêt "stop blade"
P25	activation de la lame d'arrêt "stop blade"
P26	désactivation du convoyeur d'entrée "entry convoyeur"
P27	désactivation du convoyeur de sortie "exit convoyeur" et désactivation de la bande transporteuse et Pivotement du 1er bras
P28	place d'attente
P29	place d'attente
P30	désactivation du convoyeur de sortie "exit convoyeur" et désactivation de la bande transporteuse et Pivotement du 2eme bras
P31	désactivation du convoyeur de sortie "exit convoyeur" et désactivation de la bande transporteuse et Pivotement du 3eme bras
P32	désactivation de la lame d'arrêt "stop blade"
P33	activation du convoyeur de sortie "exit convoyeur"

TABLE 20 – Places du RDP

Transitions :

transition	description
T1	Le bouton poussoir "Start"est enclenché
T2	Le bouton poussoir de supervision "Start_sup" est enclenché
T3	detection d'une de dimension 1 par le capteur de vision
T4	detection d'une de dimension 4 par le capteur de vision
T5	detection d'une de dimension 6 par le capteur de vision
T6	detection d'une de dimension 5 par le capteur de vision
T7	detection d'une de dimension 3 par le capteur de vision
T8	detection d'une de dimension 2 par le capteur de vision
T9	le capteur de vision ne detecte aucune piece
T10	temporisation de 3 seconde
T11	temporisation de 3 seconde
T12	capteur "at exit" enclenché
T13	le capteur de vision ne detecte aucune piece
T14	le capteur de vision ne detecte aucune piece
T15	le capteur de vision ne detecte aucune piece
T16	le capteur de vision ne detecte aucune piece
T17	capteur "at exit" ne detecte aucune piece
T18	capteur "at exit" enclenché
T19	capteur "at exit" enclenché
T20	arret du bras diviateur 1 "sorter 1"
T21	capteur "at exit" ne detecte aucune piece
T22	capteur "at exit" ne detecte aucune piece
T23	arret du bras diviateur 1 "sorter 1"
T24	arret du bras diviateur 1 "sorter 1"
T25	Lame d'arrêt "stop blade" désactivé

TABLE 21 – Transition du RDP

La modélisation par les réseaux de Petri nous a permis de concevoir un modèle qui respecte fidèlement le mode de fonctionnement de notre systèmes automatique des stations de tri, Vu la complexité de notre système, la simplification qu'offre les réseaux de petri par leur fiabilité et leur grande capacité de représentation mets en évidence les structures fonctionnelles indispensables pour le bon fonctionnement de notre systèmes.

3 Validation des Réseaux de petri :

3.1 Vérification Structurelle

Analyse de la connectivité : On a vérifié que le réseau est bien connecté, c'est-à-dire qu'il n'existe aucun sous-graphe isolé. Toutes les places et transitions sont accessibles depuis n'importe quelle autre place ou transition dans le réseau. Cette étape garantit que le réseau de Petri fonctionne comme une unité cohérente et intégrée.

Analyse de la consistance : L'analyse de la consistance a été réalisée pour s'assurer que chaque place du réseau a des transitions correctement définies qui la précèdent et la suivent. Pour chaque place, nous avons vérifié que les transitions entrantes et les transitions sortantes sont bien spécifiées. Cette vérification assure que le flux du réseau de Petri est cohérent et que les transitions peuvent se produire sans ambiguïté ni erreur.

3.2 Propriétés Dynamiques :

Pour l'étude dynamique des cinq réseaux de Petri, nous examinerons plusieurs propriétés essentielles : la bornitude, la vivacité, l'absence de blocages et l'accessibilité. Ces propriétés permettent de garantir que les réseaux sont bien structurés et fonctionnels. Après cette analyse, nous avons constaté que les réseaux étudiés répondent favorablement à ces critères. Ils sont bornés, ce qui signifie que le nombre de jetons dans chaque place reste limité. Ils sont quasi-vivants, permettant à chaque transition d'être éventuellement tirée. De plus, ces réseaux sont exempts de blocages, assurant une évolution continue du système, et ils sont accessibles, permettant d'atteindre tous les marquages possibles.

Les modèle de réseaux de petri développé, simulé ont été validé avec le logiciel snoopy.

Le modèle crée sera transcrit sous forme d'une solution programmable et implantable sur automate programmable industriel (API) de type Siemens S7-300.

Dans la suite nous allons programmer notre modèle proposé grâce à l'outil S7-Graph, permettant de programmer et d'implémenter des modèles qui se basent sur les réseaux de petri (RdP).

4 Transcription des Modèles Réseaux de Petri vers Modèles GRAFCET

La conversion des réseaux de Petri en GRAFCET est un processus méthodique qui nécessite une compréhension approfondie des deux formalismes et de leurs principes fondamentaux. Voici une approche détaillée de ce processus :

4.1 Analyse des États et des Événements

Identification des États : Chaque place dans le réseau de Petri est examinée pour déterminer son rôle dans la représentation de l'état du système. Les propriétés structurelles, telles que la connectivité avec les transitions, sont analysées pour inférer les états du système.

Identification des Événements : Les transitions du réseau de Petri sont étudiées pour comprendre les événements ou les conditions qui influencent le comportement du système. L'analyse des pré-conditions et des post-conditions des transitions permet de déterminer les événements pertinents à inclure dans le GRAFCET.

4.2 Modélisation des Étapes et des Transitions

Association des Étapes : Chaque état identifié est associé à une étape dans le GRAFCET. Les caractéristiques des états, comme leur stabilité, sont prises en compte pour garantir une représentation précise du système.

Définition des Transitions : Les événements identifiés sont traduits en transitions dans le GRAFCET. Les conditions de franchissement sont formulées à l'aide de variables logiques ou temporelles, en tenant compte des contraintes de déclenchement et de désactivation.

4.3 Établissement des Liaisons et des Conditions

Détermination des Liaisons : Les arcs du réseau de Petri guident l'établissement des liaisons entre les étapes du GRAFCET, reflétant l'ordre séquentiel dans lequel les étapes doivent être franchies pour atteindre différents états.

Intégration des Conditions : Les conditions de franchissement des transitions sont intégrées sous forme de variables de contrôle et de conditions de transition, garantissant le respect des contraintes de synchronisation et de temporisation.

4.4 Validation et Vérification du Modèle Converti

Validation Structurelle : Le modèle est soumis à une validation pour vérifier sa cohérence interne et sa conformité aux spécifications. La préservation des propriétés fondamentales est vérifiée.

Validation Comportementale : Le comportement du modèle est évalué pour s'assurer qu'il reproduit fidèlement le comportement du système d'origine, en respectant les exigences fonctionnelles et temporelles. [4]

4.5 Conversions de Base

Le tableau ci-dessous offre une vue d'ensemble des transformations essentielles, telles que l'association des étapes et des transitions, ainsi que l'établissement des conditions et des liaisons. [12]

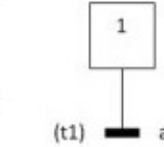
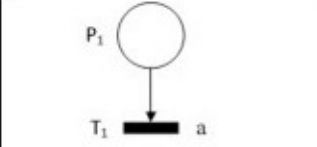


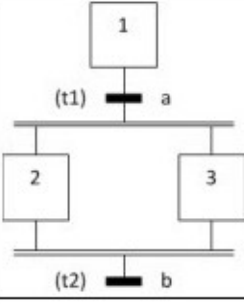
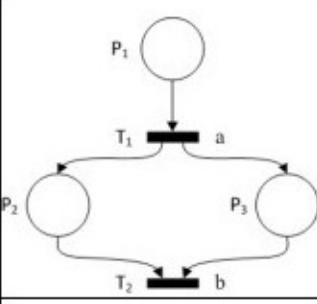
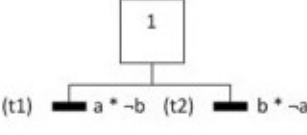
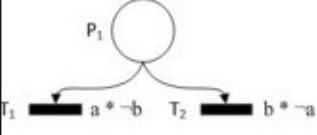
	GRAFCET	rdp
step/place, transition, junction/arc		
initial step/place		
junction "AND"		
Junction "OR"		

TABLE 22 – Conversions de Base

La conversion des réseaux de Petri en GRAFCET nécessite une analyse approfondie et une expertise dans la modélisation des systèmes dynamiques et la logique de contrôle. En suivant une approche rigoureuse, il est possible de réaliser cette conversion de manière efficace et précise, permettant ainsi une représentation formelle et vérifiable du comportement du système.

5 Comparaison entre les Réseaux de Petri et le LADDER

Les Réseaux de Petri et le langage LADDER (ou Ladder Diagram) sont deux outils puissants utilisés dans le domaine de l'automatisation industrielle pour modéliser, analyser et contrôler les systèmes. Bien qu'ils aient des origines et des approches différentes, ils visent tous deux à représenter et à résoudre les problèmes liés à la logique de contrôle. Cette comparaison vise à examiner les différences et les similitudes entre ces deux approches. [10]

5.1 Structure et représentation

Les Réseaux de Petri sont des graphes bipartis composés de places, de transitions et d'arcs. Les places représentent les états du système, les transitions représentent les événements ou les actions, et les arcs décrivent les relations de causalité entre eux. Le langage LADDER est basé sur des symboles électriques et représente le contrôle logique sous forme de contacts et de bobines. Les contacts représentent les conditions ou les états, tandis que les bobines représentent les actions ou les événements.

5.2 Expressivité et puissance de modélisation

Les Réseaux de Petri offrent une modélisation graphique puissante pour représenter les systèmes dynamiques et concurrents. Ils peuvent capturer des comportements complexes tels que la concurrence, la synchronisation et les boucles. Le langage LADDER est plus limité dans sa capacité à représenter des structures complexes. Il est plus adapté aux applications de contrôle logique simples à moyennement complexes trouvées dans les systèmes d'automatisation industrielle.

5.3 Analyse et simulation

Les Réseaux de Petri offrent des méthodes d'analyse formelles telles que la vérification de propriétés, la détection de blocages et l'analyse de performance. Ils peuvent également être simulés pour étudier le comportement dynamique des systèmes. Bien que le langage LADDER ne soit pas aussi formel que les Réseaux de Petri, il est largement utilisé dans l'industrie pour la programmation d'automates programmables industriels (API). Il peut être simulé pour tester le fonctionnement du système avant sa mise en service.

les Réseaux de Petri et le langage LADDER sont des outils complémentaires dans le domaine de l'automatisation industrielle. Les Réseaux de Petri offrent une modélisation formelle et une analyse approfondie des systèmes dynamiques, tandis que le langage LADDER fournit une méthode pratique et intuitive pour programmer des automates industriels. Le choix entre les deux dépend souvent de la complexité du système à contrôler et des besoins spécifiques de l'application. En comprenant les forces et les faiblesses de chaque approche, les ingénieurs peuvent choisir la méthode la plus appropriée pour résoudre efficacement les problèmes de contrôle et d'automatisation.

6 Transcription des Réseaux de Petri en Langage Ladder

La conversion des réseaux de Petri vers le langage Ladder Diagram (LD) est une pratique essentielle dans le domaine de l'automatisation industrielle. Les réseaux de Petri fournissent une représentation graphique puissante pour modéliser et analyser les systèmes dynamiques et concurrents, tandis que le langage Ladder Diagram est largement utilisé dans la programmation des automates programmables industriels (API) pour contrôler les processus de fabrication et les systèmes automatisés.

Cette conversion est cruciale car elle permet de passer des modèles conceptuels abstraits des réseaux de Petri à des implémentations concrètes et exécutables dans des environnements industriels réels. En convertissant un modèle de réseau de Petri en un diagramme en échelle, les ingénieurs peuvent concevoir des algorithmes de contrôle précis et efficaces, en utilisant les conventions et les structures familières du langage Ladder. [13]

6.1 Tableau de Correspondance : Conversion des Réseaux de Petri en Langage Ladder

La transformation des structures complexes des Réseaux de Petri vers la notation en échelle (ladder) nécessite une analyse méticuleuse des éléments et des processus impliqués. Le tableau ci-dessous offre une représentation systématique de cette conversion, éclairant ainsi le chemin vers une implémentation efficace sur des systèmes automatisés industriels. [8]

Construction de Réseau de Petri	Diagramme en échelle (Ladder diagram) correspondant à la variable
Place de Tâche	Variable de processus P_i pour la i -ème tâche (variable interne de LD).
Place de Ressource Inactive	Variable de ressource R_i pour la i -ème ressource. (variable interne de LD).
Place de Capteur	Variable de capteur S_i pour le i -ème capteur.
Transition	Variable de transition T_i pour la i -ème transition. (variable interne de LD).
Jeton dans une place spécifique	La variable correspondante à la place P_N est activée.
Transition activée	La variable de transition correspondante est activée.
Arc émanant d'une place i et se terminant dans une transition j	Variable de fin F_{ij} menant de la fin de la tâche i à la transition j . (variable interne de LD).
Arc émanant d'une transition i et se terminant dans une place j	Un barreau d'échelle avec un contact d'entrée T_i et une bobine de sortie correspondant à la place j (soit P_j ou R_j).

TABLE 23 – Tableau de Correspondance (conversion des rdp's en LADDER)

6.2 Méthodologie de Modélisation et de Traduction des Réseaux de Petri vers le Langage Ladder

Étapes structurées pour attacher des variables dans un réseau de Petri :

- Énumérez les tâches (c'est-à-dire, les places ombragées d'un Réseau de Petri) et attribuez une variable de processus P_i à chaque tâche i . Par exemple, dans la Fig. 3, la tâche 1 est attribuée la variable P_1 (correspondant au processus 1).
- numérez les capteurs et attribuez une variable de capteur S_i à chaque capteur i . Par exemple, dans la figure 45, le capteur qui détecte l'arrivée de la pièce est attribué la variable S_1 (correspondant au capteur 1).
- Énumérez les ressources (par exemple, les machines) et attribuez une variable de ressource R_i à chaque ressource i . Par exemple, dans la figure 45, le robot est attribué la variable de ressource R_0 .
- Chaque transition du Réseau de Petri est attribuée une variable appelée T_i pour la i -ème transition. Par exemple, T_1 et T_2 dans la figure 45

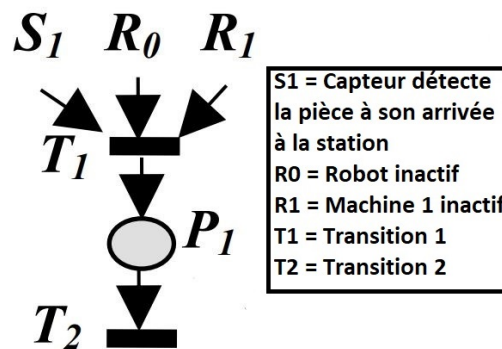


FIGURE 45 – Fragment d'un Réseau de Petri

Transcription des modèles réseaux de Petri vers le langage ladder :

Toutes les variables définies dans la section 6.2 sont incluses dans le programme Ladder. En particulier, les variables de transition T_i sont activées lorsque toutes leurs variables d'entrée sont activées. De même, les variables associées aux places sont activées lorsqu'elles sont vraies.

La traduction en Ladder est réalisée en mettant en œuvre la logique et les conditions suivantes :

- Les variables capteurs (S_i) sont des variables d'entrée activées par des événements externes.
- Toutes les autres variables (P_i , T_i , R_i) sont des variables internes activées par la logique interne et les variables d'entrée.
- Les sorties externes sont activées et désactivées uniquement à l'intérieur des tâches embarquées (ne sont pas visibles dans le PN de haut niveau).
- L'activation de chaque transition se produit au moment où toutes ses entrées sont activées.
- L'activation de chaque transition éteint toutes ses variables de ressources d'entrée (R_i), signalant que les ressources sont occupées.
- L'activation de chaque transition active les variables de ses arcs de sortie du PN.
- L'activation d'une transition qui précède une tâche i active sa variable P_i , qui active la logique de la tâche embarquée correspondante.

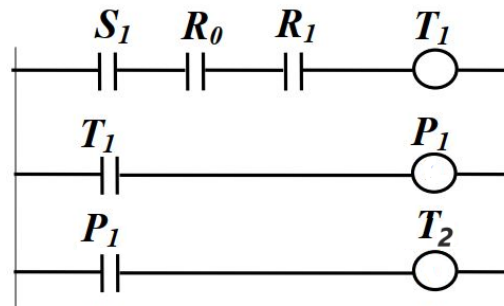


FIGURE 46 – le programme en Ladder

Autres exemples de Transcription des modèles RDP en logique en échelle (ladder) :
 le tableau suivant illustre d'autres exemples de conversion des RDP en langage ladder :

Petri Net	Diagram Ladder

FIGURE 47 – le programme en Ladder

La conversion des réseaux de Petri vers le langage Ladder Diagram (LD) représente une étape fondamentale dans le domaine de l'automatisation industrielle. Ce processus permet de transformer des modèles conceptuels abstraits en des implémentations concrètes et exécutable dans des environnements industriels réels. En passant par cette conversion, les ingénieurs peuvent concevoir des algorithmes de contrôle précis et efficaces, en utilisant les conventions et les structures familières du langage Ladder Diagram.

Le tableau de correspondance fourni offre une représentation systématique de cette conversion, éclairant ainsi le chemin vers une implémentation efficace sur des systèmes automatisés industriels. Chaque transition du réseau de Petri est attribuée une variable dans le langage Ladder, ce qui facilite la traduction et la compréhension du modèle.

En mettant en œuvre la logique et les conditions spécifiées, les variables capteurs sont activées par des événements externes, tandis que les autres variables sont activées par la logique interne et les variables d'entrée. Les sorties externes sont contrôlées à l'intérieur des tâches embarquées, tandis que l'activation de chaque transition et ses effets sont clairement définis.

7 Conclusion

Ce chapitre a présenté en détail les fondements et les concepts clés des réseaux de Petri, offrant ainsi une compréhension approfondie de cet outil puissant pour la modélisation des systèmes à événements discrets. Nous avons exploré les différentes composantes des réseaux de Petri, telles que les places, les transitions et le marquage, ainsi que leurs rôles dans la représentation des systèmes et des événements. De plus, nous avons examiné plusieurs propriétés importantes des réseaux de Petri, telles que l'accessibilité, la bornitude et la vivacité, qui fournissent des informations cruciales sur le comportement et les caractéristiques des systèmes modélisés. Enfin, nous avons abordé les méthodes d'analyse, de validation et de conversion des réseaux de Petri, soulignant leur utilité dans le développement et la vérification des systèmes complexes.

Chapitre 3 : Développement de la solution de commande sous Step7.

1 Introduction

Les Automates Programmables Industriels (API) sont apparus aux Etats-Unis vers 1969 où ils répondaient aux désirs des industries de l'automobile de développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des techniques et des modèles fabriqués.

Il est en général manipulé par un personnel automatique. Le développement de l'industrie à entraîner une augmentation constante des fonctions électroniques présentes dans un automatisme c'est pour ça que l'API s'est substitué aux armoires à relais en raison de sa souplesse dans la mise en œuvre, mais aussi parce que dans les coûts de câblage et de maintenance devenaient trop élevés.

Ce que nous venons de voir dans ce chapitre c'est une représentation des automates programmables industrielles et les outils utilisés pour le développement de notre application tels que l'automate SIEMENS S7-300, le logiciel de programmation STEP7, le simulateur PLCSIM.

2 Description des Automate programmable industriel (API)

L'Automate Programmable Industriel (API) est un appareil électronique programmable, adapté à l'environnement industriel, qui réalise des fonctions d'automatisme pour assurer la commande de préactionneurs et d'actionneurs à partir d'informations logique, analogique ou numérique.

Un automate programmable est adaptable à un maximum d'application, d'un point de vue traitement, composants, langage. C'est pour cela qu'il est de construction modulaire.

2.1 Structure matériel d'un API

Les caractéristiques principales d'un automate programmable industriel (API) sont : coffret, rack, baie ou cartes.

- Compact ou modulaire
- Tension d'alimentation
- Taille mémoire
- Sauvegarde (EPROM, EEPROM, pile, ...)
- Nombre d'entrées/sorties
- Modules complémentaires (analogique, communication,..)
- Langage de programmation

2.2 Aspect interne d'un automate programmable

Les API comportent quatre principales parties :

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

La structure interne d'un automate programmable industriel (API) est assez voisine de celle d'un système informatique simple, L'unité centrale est le regroupement du processeur et de la

mémoire centrale. Elle commande l'interprétation et l'exécution des instructions programme. Les instructions sont effectuées les unes après les autres, séquencées par une horloge. Deux types de mémoire cohabitent :

- **La mémoire Programme** : où est stocké le langage de programmation. Elle est en général figée, c'est à dire en lecture seulement. (ROM : mémoire morte)
- **La mémoire de données** : utilisable en lecture-écriture pendant le fonctionnement c'est la RAM (mémoire vive). Elle fait partie du système entrées-sorties. Elle fige les valeurs (0 ou 1) présentes sur les lignes d'entrées, à chaque prise en compte cyclique de celle-ci, elle mémorise les valeurs calculées à placer sur les sorties.

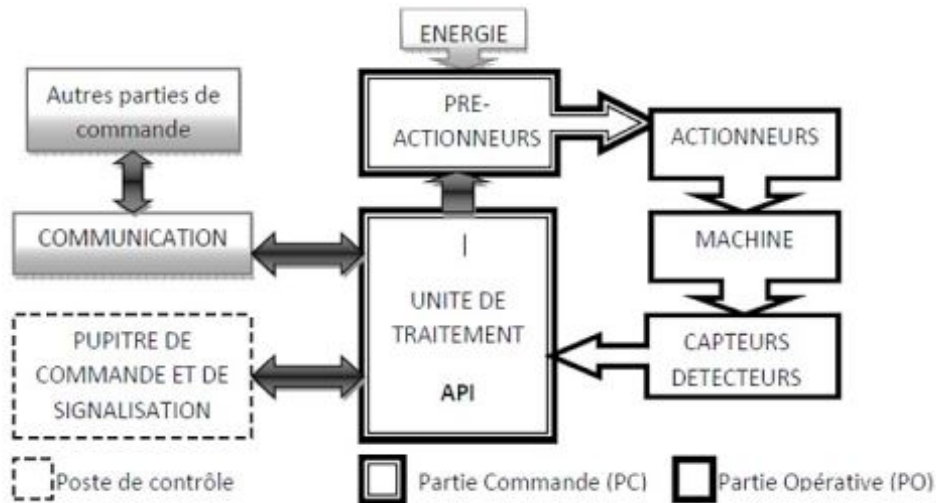


FIGURE 48 – Structure interne d'un automate programmable industriel (API).

2.3 Aspect externe d'un automate programmable

Les automates peuvent être de type compact ou modulaire. De type compact, on distinguera les modules de programmation (LOGO de Siemens, ZELIO de Schneider, MILLENIUM de Crouzet ...) des micro-automates. Il intègre le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage rapide, E/S analogiques ...) et recevoir des extensions en nombre limité. Ces automates, de fonctionnement simple, sont généralement destinés à la commande de petits automatismes. De type modulaire, le processeur, l'alimentation et les interfaces d'entrées / sorties résident dans des unités séparées (modules) et sont fixées sur un ou plusieurs racks contenant le "fond de panier" (bus plus connecteurs). Ces automates sont intégrés dans les automatismes complexes où puissance, capacité de traitement et flexibilité sont nécessaires.



FIGURE 49 – Automate programmable compact.

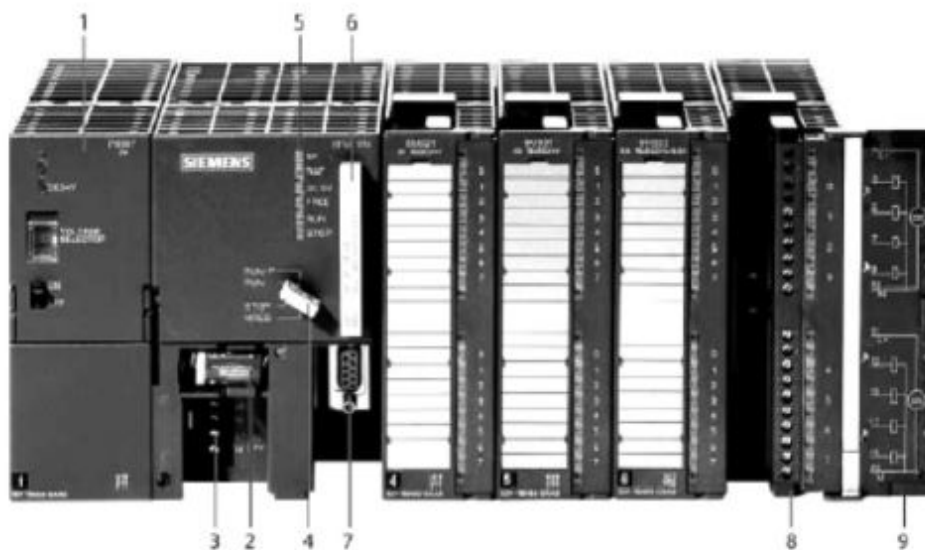


FIGURE 50 – Automate programmable modulaire

- 1) Module d'alimentation
- 2) Pile de sauvegarde
- 3) Connexion au 24V cc
- 4) Commutateur de mode (à clé)
- 5) LED de signalisation d'état et de défauts
- 6) Carte mémoire
- 7) Interface multipoint (MPI)
- 8) Connecteur frontal
- 9) Volet en face avant

2.4 Les principaux éléments de l'automate

Le module d'alimentation :

Le bloc d'alimentation (Power Supply), est nécessaire pour convertir la tension d'entrée alternative (220 V) du secteur en une tension continue (24V,48V...) nécessaire au processeur et aux circuits des modules d'interface d'entrée et de sortie.

La puissance des alimentations varie entre un API et un autre et demandent un courant allant

de 2A à 50A, en fonction du nombre d'interfaces d'E/S alimentées par cette alimentation.

L'unité centrale :

Le module CPU est l'unité contenant le microprocesseur. Cette unité interprète les signaux d'entrée et exécute les actions de commande en fonction du programme enregistré dans sa mémoire, communiquant les décisions sous forme des signaux d'actions aux sorties, Aussi, ce module contient une interface de programmation afin de communiquer avec la console de programmation suivant un protocole bien déterminé (Par exemple TCP/IP, MPI-bus ...etc.).

Le processeur :

Son rôle consiste d'une part à organiser les différentes relations entre la zone mémoire et les interfaces d'entrées et de sorties et d'autre part à exécuter les instructions du programme. Les instructions sont effectuées les unes après les autres, séquencées par une horloge.

Les bus :

C'est un ensemble de conducteurs qui réalisent la liaison entre les différents éléments de l'automate. Dans un automate modulaire, il se présente sous forme d'un circuit imprimé situé au fond du bac et supporte des connecteurs sur lesquels viennent s'enficher les différents modules : processeur, extension mémoire, interfaces et coupleurs.

La mémoire :

Elle est destinée au stockage des instructions qui constituent le programme de fonctionnement de l'automatisme.

Les interfaces des entrées / sorties :

Les interfaces d'entrées, appelés aussi coupleurs d'entrées, sont des cartes électroniques qui assurent la liaison entre l'UC de l'automate programmable et la Partie Opérative (le processus via les capteurs et les pré-actionneurs).

Les interfaces d'entrées reçoivent les signaux des capteurs et des commandes de l'opérateur (via le pupitre de commande) qu'ils traitent pour les rendre compatibles avec les caractéristiques internes de l'A.P.I.

Les interfaces de sorties reçoivent les signaux de l'UC. Ils les amplifient et les rendent compatibles avec les pré-actionneurs commandés et les contrôles du pupitre de commande. Pour la plupart des A.P.I., les coupleurs sont groupés avec l'UC, mais sont modulaires par carte ou par rack. Le coupleur accède d'une part au bus d'E/S, d'autre part au bornier qui se trouve le plus souvent sur la face avant de l'automate. Certains constructeurs proposent des E/S décentralisées (cartes d'entrées / sorties déportées). Les A.P.I. offrent une grande variété d'E/S comme :
- Le module E/S Tout Ou Rien (TOR) : Permet de raccorder l'automate à des capteurs TOR (boutons poussoirs, fins de course, capteurs de proximité, etc...) ou à des pré-actionneurs (vannes, contacteurs, électrovannes, relais de puissance, LED. ...etc.). Le nombre d'entrées sur une carte est de : 4, 8, 16, 32. Chaque entrée possède deux états 0 ou 1 (binaire).

-Le module E/S analogique : Permet de traiter les signaux analogiques des grandeurs qui varient de façon continue (courant, tension). Il est muni d'un convertisseur analogique/numérique pour les entrées et un autre numérique/analogique pour les sorties. Il existe des modules à 2, 4, 8 voies.

2.5 Programmation d'un API

Chaque API à une programmation spécifique à lui (le même type d'automate peut être utilisé pour différentes applications, la différence s'effectue avec le programme installé dans

celui-ci), les plus simples sont programmés d'une manière séquentielle (il faut attendre que la condition soit remplie pour que le programme exécute la prochaine instruction), tandis que d'autres se programme en code à l'aide d'un ordinateur ou d'un programmeur.

Donc la programmation d'un API peut se faire de trois (03) façons distinctes :

- Sur l'API lui-même à l'aide de touches.
- Avec une console de programmation reliée par un câble spécifique à l'API.
- Avec un PC et un logiciel approprié avec lequel la programmation est plus conviviale.

Les Avantages de l'Utilisation des APIs dans le Domaine Industriel :

- Améliorer les conditions de travail en éliminant les travaux répétitifs
- Améliorer la productivité en augmentant la production.
- Amélioration de la qualité des produits et réduction des coûts de production.
- Facilité de maintenance
- Les automates programmables sont facilement programmables et ont un langage de programmation facile à comprendre alors la modification du programme est facile par rapport à la logique câblée.

Les Inconvénients de l'Utilisation des APIs dans le Domaine Industriel :

- Une importante consommation d'énergie.
- Plantage.
- Il y a trop de travail requis dans les fils de connexion.
- Besoin de formation d'un personnel plus qualifié.

2.6 Choix d'un automate

Il revient à l'utilisateur d'établir le cahier des charges de son système et de regarder sur le marché l'automate le mieux adapté aux besoins, en considérant un certain nombre de critères importants :

- Le nombre d'entrées/sorties
- La nature des entrées/sorties (numérique, analogique, etc.)
- La nature du traitement (temporisation, comptage, etc.),
- Le dialogue (la console détermine le langage de programmation),
- La communication avec les autres systèmes,
- Les moyens de sauvegarde du programme (disquette, carte mémoire, etc.).
- La fiabilité et la robustesse,
- Les capacités de traitement du processeur (vitesse, données, opérations, temps réel)

En tenant compte des points soulignés précédemment, nous avons choisis comme système de traitement l'automate SIEMENS S7-300. En plus de cela, Il est le plus utilisé dans les industries.

3 Présentation du logiciel SIMATIC Manager STEP7

3.1 Le logiciel de programmation STEP7

STEP 7 est le nom du logiciel de programmation pour les systèmes SIMATIC S7 et par conséquent le logiciel de programmation de notre S7-300. STEP 7 nous offre toutes les fonctionnalités nécessaires pour configurer, paramétrer et programmer notre S7-300.

3.2 Programmation sur STEP 7

Principe :

Après une compréhension d'un cahier de charge bien élaboré, pour assurer une bonne programmation de l'automatisation d'un dispositif, il est recommandé de suivre les étapes suivantes :

- Configuration, le paramétrage du matériel et de la communication réseau
- Création de la table mnémonique
- Élaboration du programme
- Chargement du programme
- Test du programme sur le simulateur PLCISIM
- Traitement et le diagnostic des erreurs dans le cas de perturbation

Configuration matérielle :

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

La configuration matérielle est utilisée pour :

- Configurer et paramétrer le support matériel.
- Configurer la liaison entre la CPU et le console de programmation.

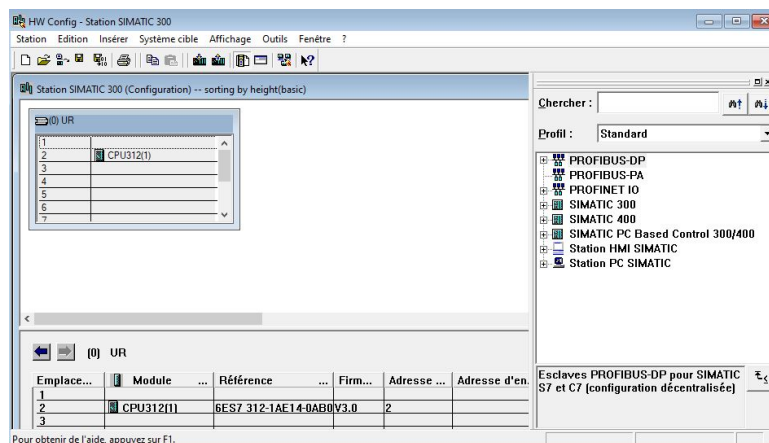


FIGURE 51 – Configuration matérielle.

Quelle que soit la technique de configuration d'une station, il faut toujours s'en tenir aux étapes suivantes pour la configurer :

- Sélectionnez un composant matériel dans la fenêtre "Catalogue du matériel"
- Amenez le composant sélectionné dans la fenêtre de station par la fonction glisser lâcher.

La partie inférieure de la fenêtre de la station donne une vue détaillée du profilé support ou châssis sélectionné ou inséré. Les numéros de référence et les adresses des modules y sont énumérés dans un tableau

3.3 Description des blocs de programmation :

La programmation par blocs, souvent appelée « programmation visuelle », est un paradigme de programmation qui utilise des éléments visuels, tels que des blocs ou des modules, pour représenter des concepts et des fonctions de code. Ces blocs sont manipulés et connectés dans une interface visuelle pour créer un programme. Les blocs de programmation sont les suivants :

-Bloc d'organisation pour le traitement de programme cyclique (OB 1) : Le traitement de programme cyclique constitue le traitement normal pour les automates programmables. Le système d'exploitation appelle l'OB1 cycliquement et déclenche ainsi le traitement cyclique du programme utilisateur.

-Fonction (FC) : Une fonction est un bloc de code sans mémoire, c'est-à-dire sans données statiques. Elle permet la transmission de paramètres dans le programme utilisateur.

-Bloc fonctionnel (FB) : est un bloc de code avec mémoire statique, permettant la transmission de paramètres dans le programme utilisateur. Il est idéal pour la programmation de fonctions complexes et répétitives comme les régulations et la sélection de mode de fonctionnement. Grâce à sa mémoire, les paramètres du FB sont accessibles à tout moment et à toute position du programme utilisateur.

-Bloc de données (DB) : est une zone de données dans un programme utilisateur qui contient des données utilisateur. Il existe des blocs de données globaux accessibles par tous les blocs de code, et des blocs de données d'instance associée à un appel particulier de blocs fonctionnels. Contrairement à tous les autres blocs, les blocs de données ne contiennent pas des instructions.

4 Développement de la solution de commande de la station pick&place sous Step 7.

Pour une meilleure compréhension de la programmation avec le logiciel SIMATIC Step 7, nous allons explorer la station Pick & Place de Factory IO et élaborer son programme.



FIGURE 52 – Station pick&place de factory i/o.

4.1 Configuration matérielle

Nous disposons de 10 entrées numériques et 10 sorties numériques, en plus d'une sortie analogique, nous opterons pour l'utilisation d'un module d'entrée/sortie de 16 bits ainsi qu'un

module analogique.

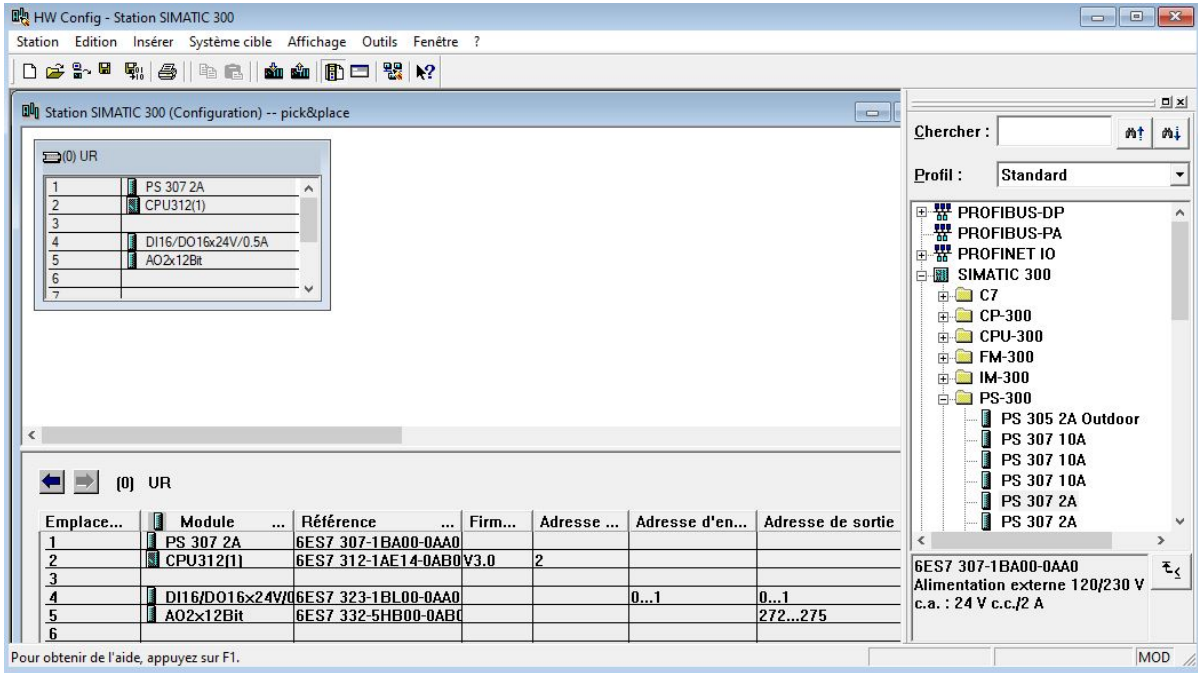


FIGURE 53 – Configuration matérielle.

4.2 La table mnémonique

	Etat	Mnémonique	Opérande	Type de do	Commentaire
1		Cycle Execution	OB 1	OB 1	
2		emergency stop	E 1.0	BOOL	
3		entry convoyer	A 0.0	BOOL	
4		exit convoyer	A 0.1	BOOL	
5		grab	A 0.4	BOOL	
6		init_compteur	M 0.1	BOOL	
7		item at entry	E 0.0	BOOL	
8		item at exit	E 0.1	BOOL	
9		item detected	E 0.4	BOOL	
10		mise en marche	M 0.0	BOOL	
11		move x	A 0.2	BOOL	
12		move z	A 0.3	BOOL	
13		moving x	E 0.2	BOOL	
14		moving z	E 0.3	BOOL	
15		reset	E 0.6	BOOL	
16		reset light	A 0.6	BOOL	
17		start	E 0.5	BOOL	
18		start light	A 0.5	BOOL	
19		stop	E 0.7	BOOL	
20		stop light	A 0.7	BOOL	
21		counter	AW 272	WORD	
22					

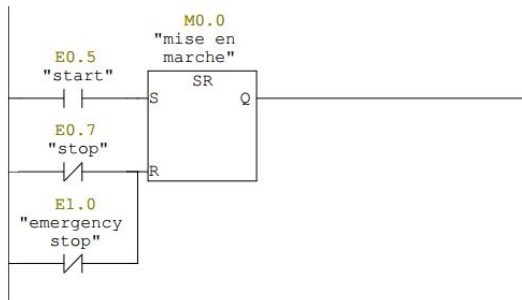
FIGURE 54 – table des mnémonique

4.3 les blocs de programmation

Dans le cas de cette station, nous allons utiliser OB1 et FC1.

4.4 Développement du programme LADDER

RESEAU 1 :



Description du Réseau 1 :

La première bascule SR de ce programme ladder agit comme un mécanisme de mise en marche et d'arrêt sécurisé du système. Elle utilise les entrées suivantes :

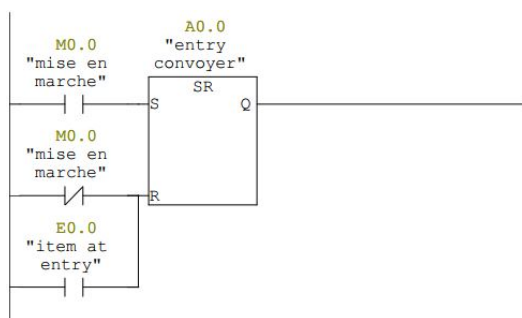
- Entrée S (Set) : start et start de supervision.
- Entrée R (Reset) : stop, stop de supervision, arrêt d'urgence et arrêt d'urgence de supervision.

Fonctionnement :

- Mise en marche du système (Set) : Lorsque l'une des entrées (start ou start de supervision) est activée, la bascule passe en état "Set", activant ainsi le système. Cet état est maintenu même après que le bouton start soit relâché, garantissant que le système continue de fonctionner sans interruption.
- arrêt du système (Reset) : Lorsque l'une des entrées (stop, stop de supervision, arrêt d'urgence ou arrêt d'urgence de supervision) est activée, la bascule passe en état "Reset", désactivant ainsi le système. Cet état est également maintenu empêchant le redémarrage non intentionnel du système.

En résumé, cette bascule SR assure une gestion fiable de l'état du système, en maintenant le fonctionnement après l'activation initiale par l'un des boutons start et en garantissant l'arrêt complet et sécurisé lorsque l'un des boutons stop ou d'arrêt d'urgence est activé.

RESEAU 2 :



Description de la Bascule Réseau 2 :

La deuxième bascule RS commande le convoyeur d'entrée du système. Elle utilise les entrées suivantes :

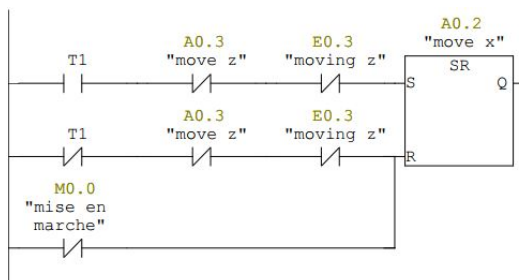
- Entrée S (Set) : mise en marche (normalement ouvert).
- Entrée R (Reset) : mise en marche (normalement fermé) et détection d'un objet par le capteur nommé "item at entry".

Fonctionnement :

- Activation du Convoyeur (Set) : Lorsque l'entrée mise en marche est activée (reliée à la sortie de la première bascule RS), la bascule passe en état "Set", démarrant ainsi le convoyeur d'entrée. Cet état est maintenu tant que les conditions de réinitialisation ne sont pas rencontrées.
- Arrêt du Convoyeur (Reset) : La bascule passe en état "Reset" lorsqu'une des conditions suivantes est remplie : l'entrée mise en marche est désactivée ou le capteur item at entry détecte un objet à l'entrée. Cette action arrête le convoyeur d'entrée.

En résumé, cette bascule SR contrôle le fonctionnement du convoyeur d'entrée, en le maintenant actif lorsque le système global est en marche, et en l'arrêtant soit via la désactivation de la bascule "mise en marche", soit lorsqu'un objet est détecté à l'entrée par le capteur nommé "item at entry".

RESEAU 3 :



Description de la Bascule Réseau 3 :

La troisième bascule SR contrôle l'axe X de la machine utilisée pour le pick & place. Elle utilise les entrées suivantes :

- Entrée S (Set) : t1 (une temporisation de 1 seconde après l'activation de l'actionneur 'grab'), ainsi que 'moving z' et 'move z' désactivés (c'est-à-dire l'axe Z de la machine est inactif)
- Entrée R (Reset) : t1 (normalement fermé), ainsi que 'moving z' et 'move z' désactivés

Fonctionnement :

- Activation de l'Axe X (Set) : La bascule passe en état "Set" lorsque la temporisation t1 est activée (c'est-à-dire une seconde après l'activation de l'actionneur 'grab') et que les mouvements de l'axe Z (moving z et move z) sont désactivés. Cet état active le mouvement de l'axe X de la machine, permettant ainsi le déplacement latéral nécessaire pour le processus de pick & place.
- Arrêt de l'Axe X (Reset) : La bascule passe en état "Reset" lorsque la temporisation t1 est désactivée (c'est-à-dire la désactivation de l'actionneur 'grab') et que les mouvements de l'axe Z sont désactivés. Cet état arrête le mouvement de l'axe X de la machine.

En résumé, cette bascule SR gère le contrôle de l'axe X de la machine de pick & place, assurant que l'axe X se déplace uniquement après une temporisation d'une seconde suite à l'activation de grab, et lorsque l'axe Z est inactif. Cela garantit une coordination précise entre les axes pour un fonctionnement efficace du processus de pick & place.

La suite du programme se trouve dans l'annexe.

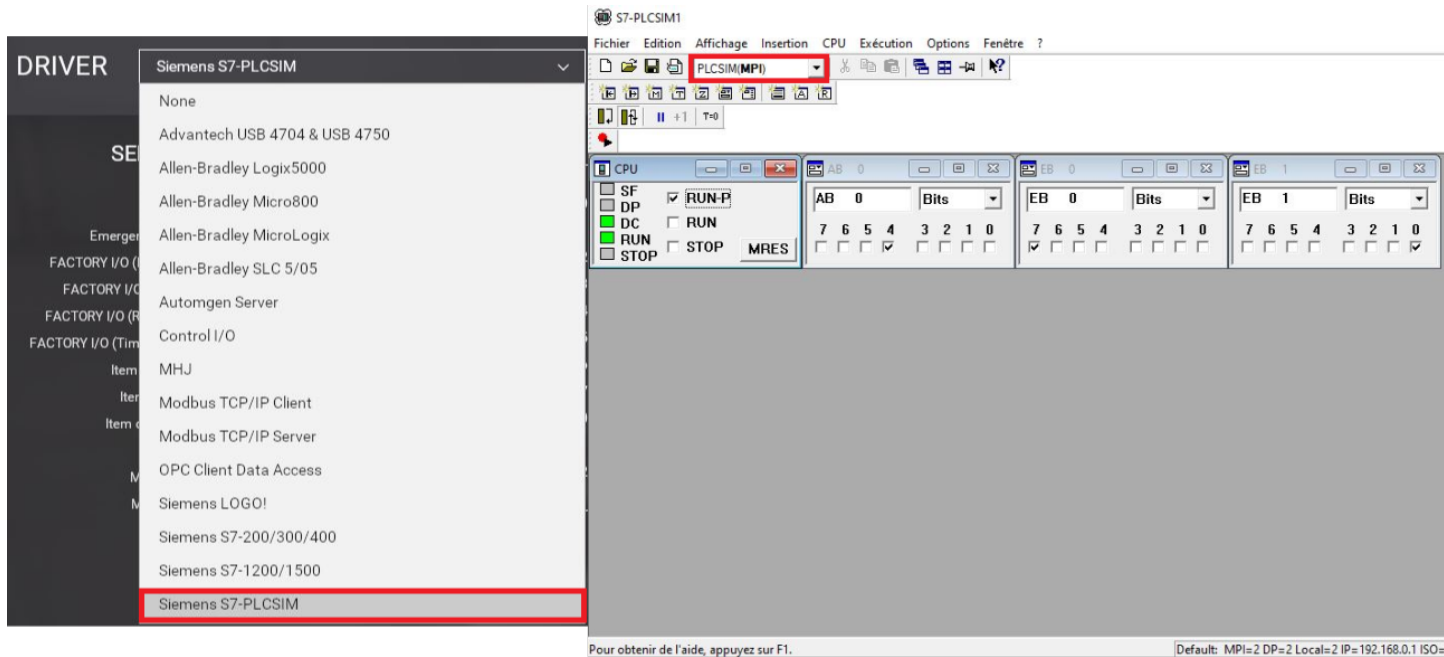
4.5 La mise en place d'une liaison entre les deux logiciel (simatic step7 et factory i/o) et Lancement de la Simulation

Pour configurer la communication entre PLCSIM et Factory IO, tout d'abord, dans PLC-SIM, nous choisissons MPI comme protocole pour établir la connexion. MPI (Multi-Point In-

terface) est un protocole de communication utilisé pour connecter les PLC Siemens.

Ensuite, dans Factory IO, nous sélectionnons le type de PLC Siemens S7-PLCSIM pour configurer la communication avec PLCSIM. Cette option permet à Factory IO de communiquer avec le simulateur PLCSIM.

Une fois que les deux logiciels sont configurés pour utiliser le protocole MPI dans PLCSIM et le type de PLC Siemens S7-PLCSIM dans Factory IO, nous devrions être en mesure de simuler notre système avec succès.



5 Conclusion

Dans ce chapitre, nous avons examiné les automates programmables industriels (API) et leur rôle crucial dans les systèmes de contrôle et d'automatisation. Nous avons également présenté le logiciel de programmation STEP7, un outil essentiel pour la programmation et la configuration des API. Par la suite, nous avons détaillé les étapes de création et de configuration d'un projet sous STEP7, en illustrant les processus nécessaires pour développer des solutions de commande efficaces.

Chapitre 4 : Développement de la solution de commande supervisée sous WinCC.

1 Introduction

Pour améliorer l'efficacité et optimiser les systèmes industriels modernes, la mise en place de solutions de commande supervisée est cruciale. Le logiciel WinCC flexible, développé par Siemens, se distingue comme un outil majeur dans ce domaine, offrant des capacités avancées de supervision et d'acquisition de données (SCADA). Ce logiciel permet la création d'interfaces utilisateur intuitives et assure un contrôle précis des processus industriels. La mise en œuvre de ces solutions permet non seulement d'optimiser les opérations industrielles, mais également d'améliorer la flexibilité et la sécurité des systèmes automatisés.

2 Présentation du logiciel WinCC flexible

WinCC flexible, est un logiciel partagé dans l'environnement STEP7, et proposé pour la configuration de divers pupitres opérateurs dans les automates de type siemens, l'opérateur humain a été contraint de conduire ou de superviser des machines automatisées, en réduisant les prises d'information et les actions directes sur le processus, ce qui conduit à l'élaboration d'interface Homme/Machine (IHM), flexible et aussi lisible pour un simple opérateur. Le dialogue est d'autant plus facilité quand l'écran comporte des images avec des synoptiques, des graphes, des bargraphes...etc.

Le WinCC flexible 2008 est subdivisé en deux sous logiciels :

2.1 WinCC flexible ES

Il est fait pour création de projet ainsi création et configuration des vues, création des liaisons et tous ce qui est comme paramètres à définir.

1- Fenêtre d'accueil :

Cette fenêtre permet l'ouverture et la création d'un projet (sans ou avec assistance).

2- Fenêtre de l'espace de travail :

L'espace de travail est divisé en plusieurs parties, où l'on crée les dessins, animations, texte et boutons nécessaires pour mieux illustrer le processus et faire apparaître les informations utiles.

3- Création des liaisons :

WinCC peut être lié à PLCSIM de différentes manières selon les besoins de l'application. Voici quelques exemples de méthodes de liaison courantes :

- Connexion directe : Dans cette méthode de liaison, la communication entre WinCC et PLCSIM s'effectue via un câble Ethernet / MPI / PROFINET direct entre le PC exécutant WinCC et le PC exécutant PLCSIM.
- Connexion via un routeur : Dans cette méthode de liaison, la communication entre WinCC et PLCSIM s'effectue via un réseau Ethernet, avec un routeur ou un commutateur Ethernet. Cette méthode est couramment utilisée dans les applications distribuées, où les composants de l'application sont répartis sur plusieurs sites distants.

- Connexion via OPC : OPC (Object Linking and Embedding for Process Control) est un standard industriel pour les interfaces de communication entre les systèmes de contrôle et les applications de supervision. WinCC peut se connecter à PLCSIM via une interface OPC, permettant ainsi une intégration facile entre les deux systèmes.

Dans tous les cas, une fois que la liaison est établie, WinCC peut accéder aux données du contrôleur programmé dans PLCSIM et les afficher dans les écrans de supervision. De cette façon, les opérateurs peuvent contrôler et surveiller les processus de manière efficace.

4- Déclaration des variables :

Pour que le WinCC communique avec l'automate programmable on doit déclarer des variables et leur affecter les adresses comme déjà fait sur Step7.

2.2 WinCC Runtime

WinCC Runtime est une version de WinCC qui permet d'exécuter les projets créés avec l'éditeur de configuration WinCC sans nécessiter l'installation de l'ensemble du logiciel WinCC sur le système de contrôle.

En d'autres termes, WinCC Runtime permet de lancer les applications créées avec WinCC sur un poste de travail, une station de travail ou une interface utilisateur, sans avoir besoin de toutes les fonctionnalités de développement du logiciel WinCC. Cela permet de séparer la conception du projet de sa mise en œuvre et de déployer les projets WinCC sur des systèmes de contrôle plus légers.

3 Développement de la solution de commande supervisée de la station pick&place sous WinCC.

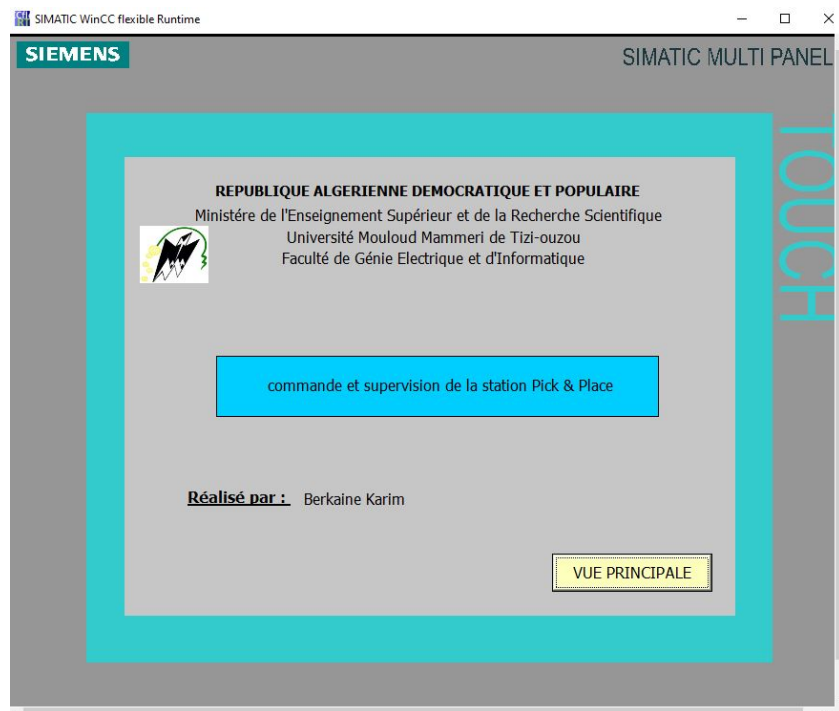


FIGURE 55 – vue d'Accueil.

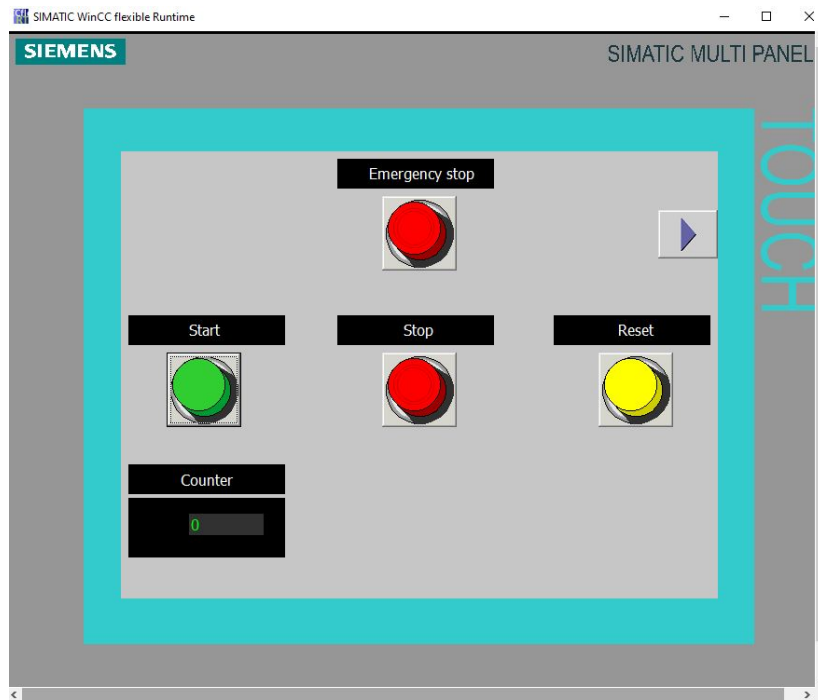


FIGURE 56 – Vue de Commande.

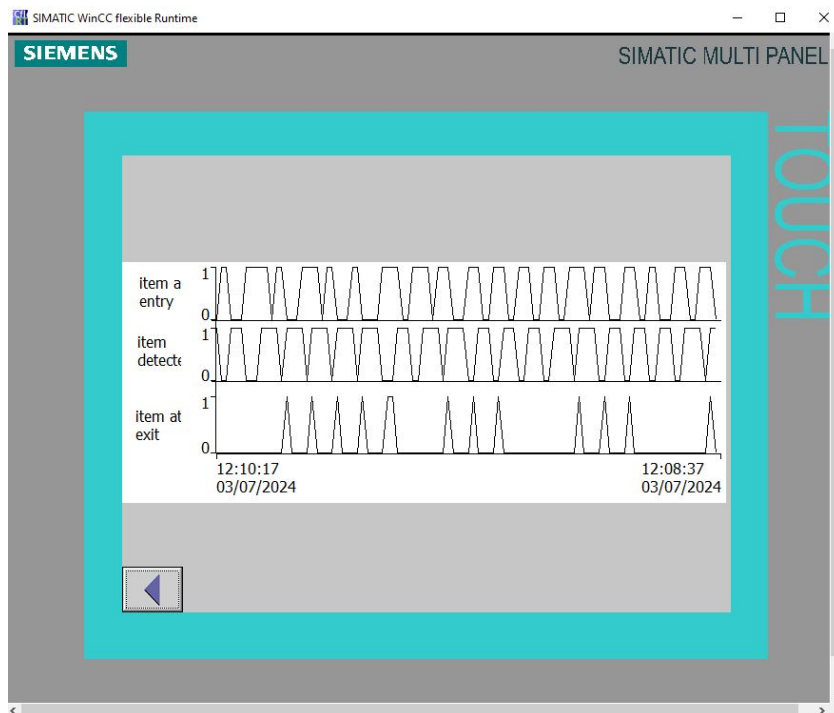


FIGURE 57 – Vue de la représentation graphique des états des capteurs.

4 conclusion

Dans ce chapitre, nous avons exploré le logiciel WinCC flexible, en commençant par une présentation générale de ses deux composantes principales : WinCC flexible ES et WinCC Runtime. Ensuite, nous avons détaillé les étapes de développement de la solution de commande supervisée pour la station pick&place sous WinCC. Cette Etude a permis de comprendre comment

configurer et utiliser WinCC flexible pour superviser et contrôler efficacement les processus industriels.

Conclusion Générale

L'objectif de ce travail est de fournir une analyse approfondie du fonctionnement des stations de tri dans le domaine de l'automatisation industrielle, en explorant leur configuration, leurs mécanismes de tri et leur modélisation à travers l'utilisation des réseaux de Petri. De plus, ce travail vise à développer des solutions de commande supervisée pour ces stations en utilisant des outils tels que Step7 et WinCC, afin de contribuer à une meilleure compréhension et mise en œuvre des processus automatisés de tri dans l'industrie.

Dans le premier chapitre, nous avons exposé les caractéristiques essentielles de Factory I/O ainsi que les spécifications des différentes stations de tri. Cette étape préliminaire a permis de mieux appréhender les défis et les enjeux liés à la mise en œuvre de ces systèmes dans un contexte industriel.

Le chapitre 2 offre une exploration détaillée et méthodique de la modélisation des stations de tri à l'aide des réseaux de Petri. En commençant par une présentation approfondie des outils de modélisation, incluant les rappels fondamentaux sur les réseaux de Petri, le chapitre progresse ensuite vers l'examen des différentes composantes des réseaux de Petri appliqués à la modélisation des processus de tri. Les différentes étapes, du modèle général aux modèles spécifiques de chaque station, sont décomposées avec une attention particulière portée à la nomenclature des places et transitions ainsi qu'à la validation des modèles. De plus, le chapitre aborde également la conversion des modèles réseaux de Petri vers des modèles GRAFCET et LADDER, mettant en lumière les différents aspects de chaque langage de modélisation et offrant ainsi une perspective comparative.

Dans le troisième chapitre, nous avons étudié les automates programmables et le logiciel de programmation STEP7, en mettant l'accent sur le développement pratique d'une solution de commande pour la station pick&place. Cette phase initiale nous a permis de maîtriser les fondamentaux de la programmation et de la configuration des automates, posant ainsi les bases essentielles pour la mise en œuvre d'un système automatisé fonctionnel.

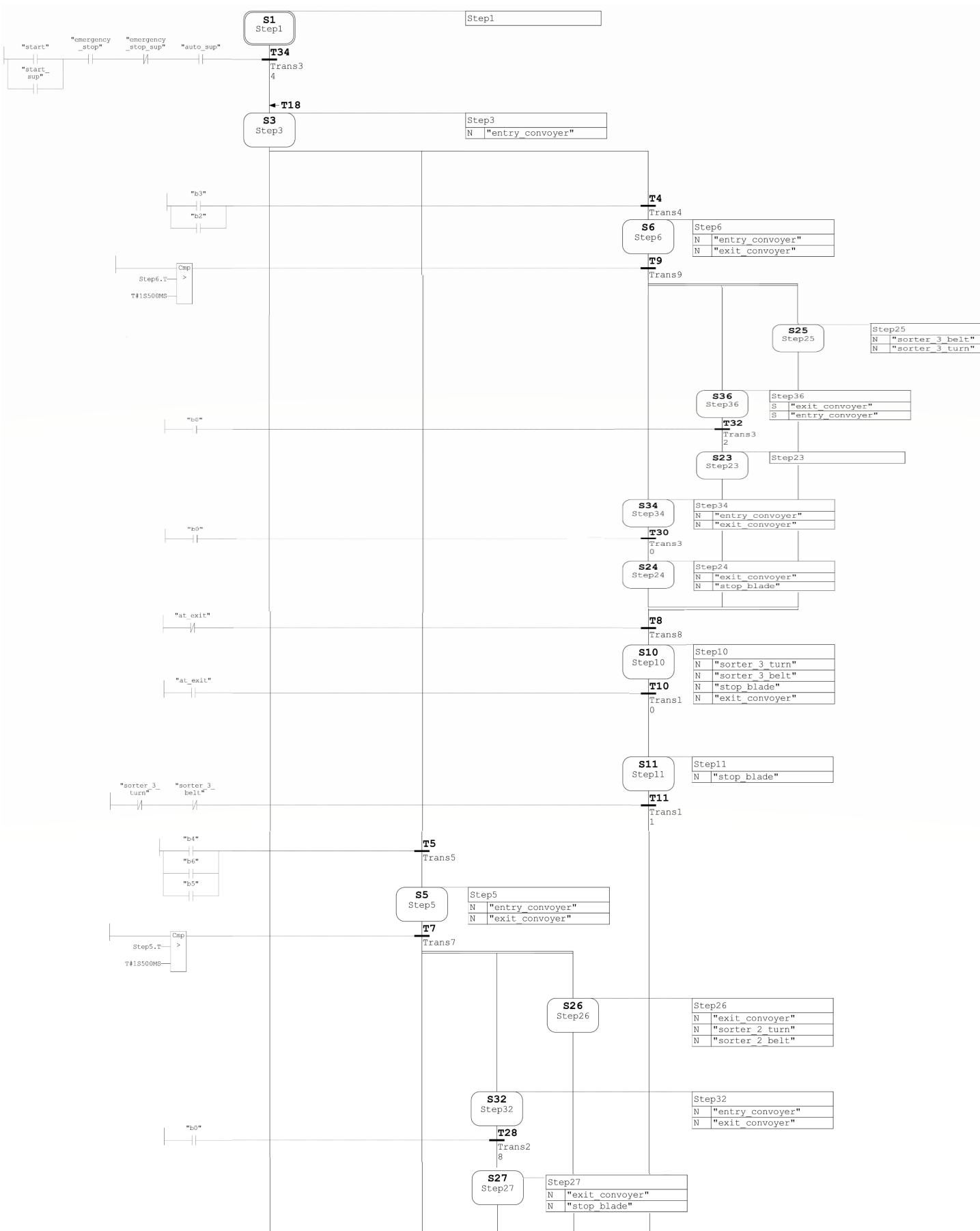
Le quatrième et dernier chapitre s'est concentré sur le logiciel WinCC flexible. Nous avons également procédé à la création des vues de supervision, qui permettent de visualiser et de contrôler efficacement les processus automatisés. Ces interfaces utilisateur ont été spécialement conçues pour la station pick&place, offrant une gestion intuitive des opérations et renforçant la capacité à réagir rapidement aux changements de conditions opérationnelles.

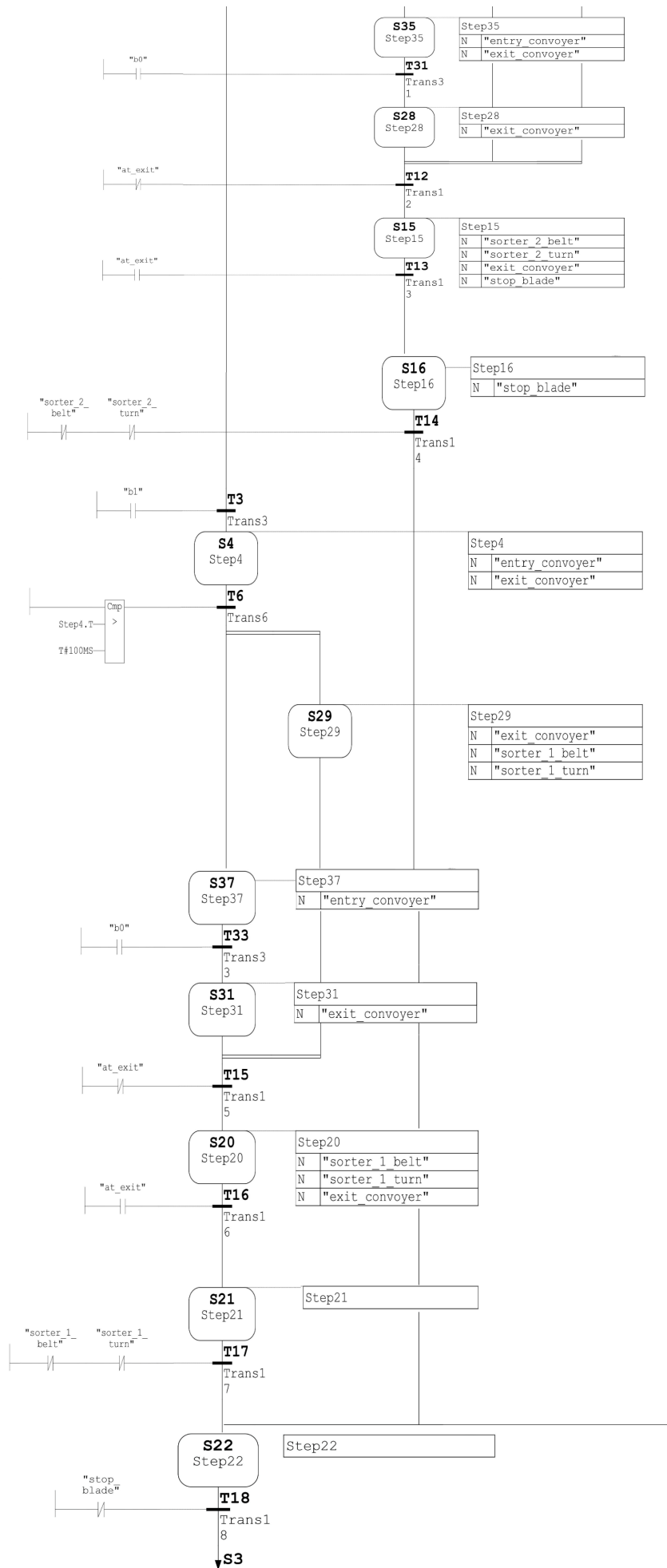
Bibliographie

- [1] P.J.Ramadge, W.M. Wonham. *"The control of Discrete Event Systems"*. january, 1989.
- [2] M.silva, R.valette. *"Petri Nets and Flexible manufacturing"*. 1990.
- [3] T.Murata. *"Petri Nets, properties, analysis and Applications"*. April 1989.
- [4] A.Ichikawa, K.Hiraishi. *"Analysis and control of Discrete Event Systems Represented by Petri Nets"*. 1988
- [5] D.crockett, A.Desrochers, F.Dicesare, T.ward. *"Implementation of Petri net Controller for a Machining Workstation"*. april 1987.
- [6] R.valette. *"Etude comparative de deux outils de représentation : Grafcet et réseaux de Petri"*. Décembre 1979.
- [7] E.Kasturia, F.Dicesare, A.Desrochers. *"Real Time control of multilevel Manufacturing Systems Using Petri Nets"*. june 1988.
- [8] C.Xuekum, L.Lilian, Q.Pengfei. *"Method For Translating Ordinary Petri Nets To Ladder Diagram"*. 2012.
- [9] G.B.Lee, J.S.Lee. *"Constructing Petri Net State Equation for Ladder Diagram"*. 2002.
- [10] K.Venkatesh, M.C.Zhou, R.J.Caudill. *"Comparing ladder logic diagram and Petri nets for sequence controller design through a discrete manufacturing system"*. 1994.
- [11] <https://factoryio.com/>
- [12] F.Schumache, S.Schröck, A.Fay. *"Transforming hierarchical concepts of GRAFCET into a suitable Petri net formalism"*. 2013.
- [13] M.ang, B.Bidanda. *"Automatic Translation of a Petri-Net to a Ladder Diagram"*. 2009.

Annexe

programme de la station de Tri de pieces par couleur :





OB1 - <hors ligne>

"Cycle Execution"

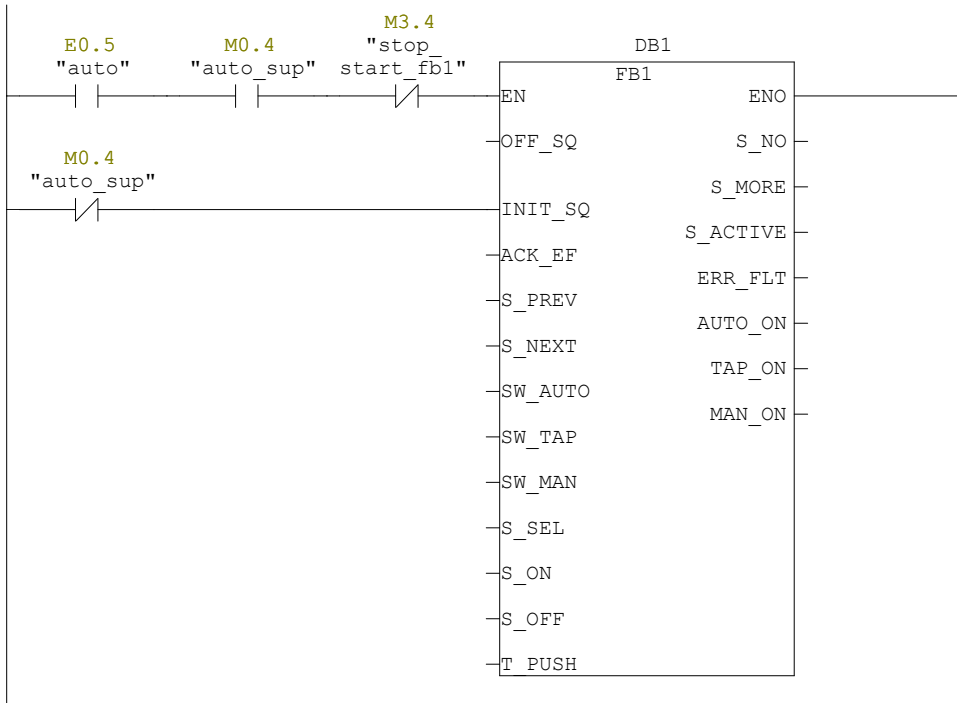
Nom :
Auteur :
Horodatage Code :
Interface :
Longueur (bloc/code /données locales) : 00800 00618 00028

Famille :
Version : 0.1
Version de bloc : 2
 01/06/2024 11:42:02
 15/02/1996 16:51:12

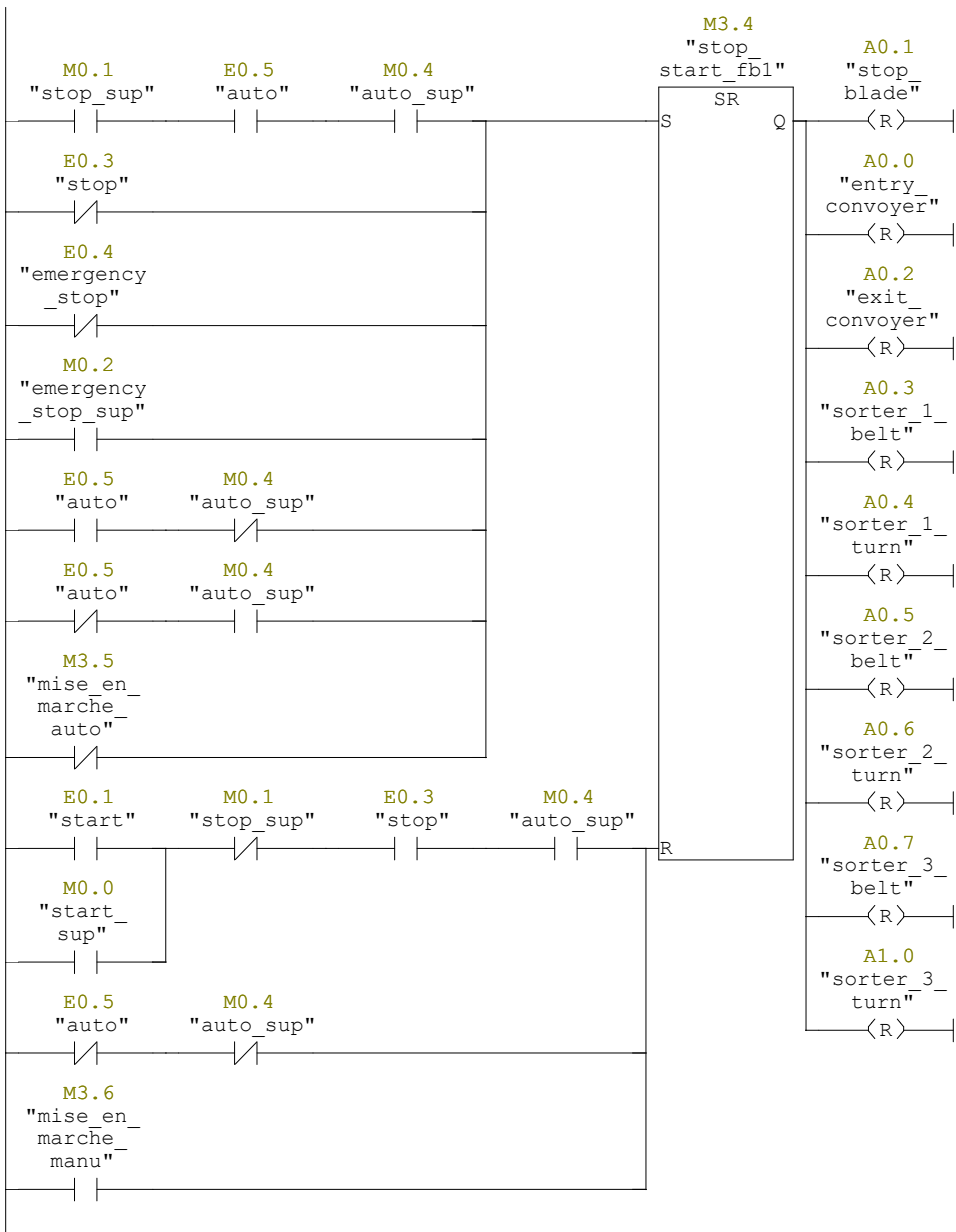
Nom	Type de données	Adresse	Commentaire
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

Bloc : OB1 "Main Program Sweep (Cycle)"

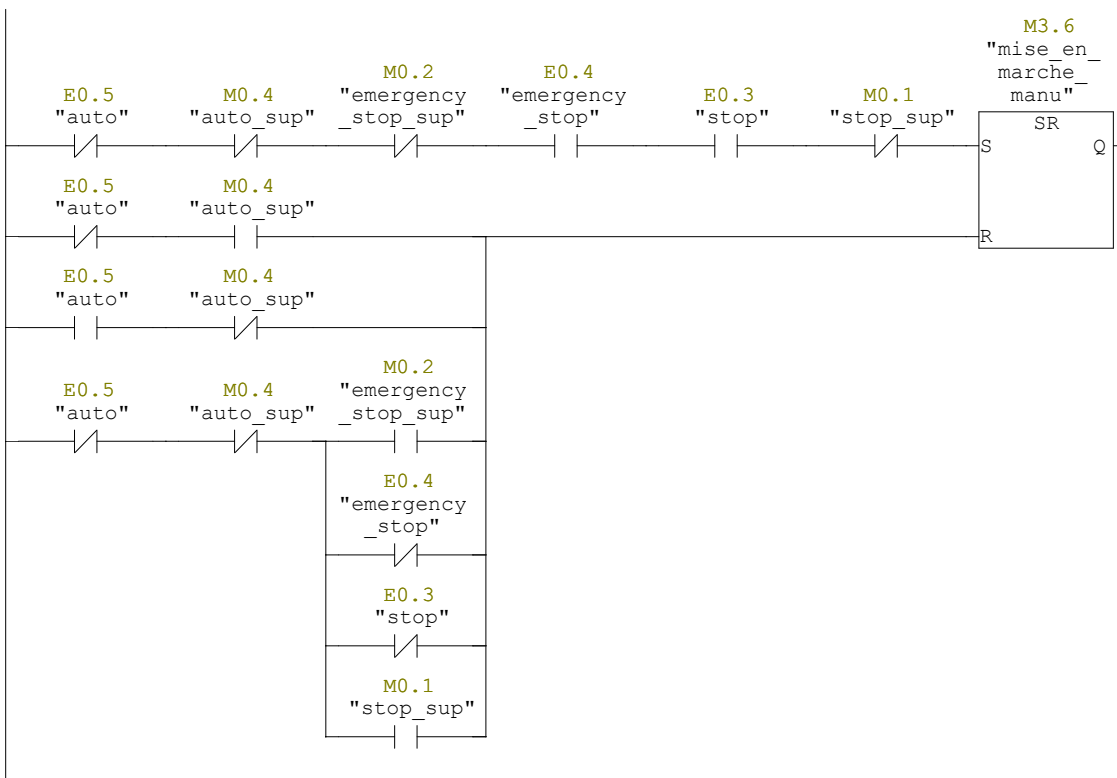
Réseau : 1



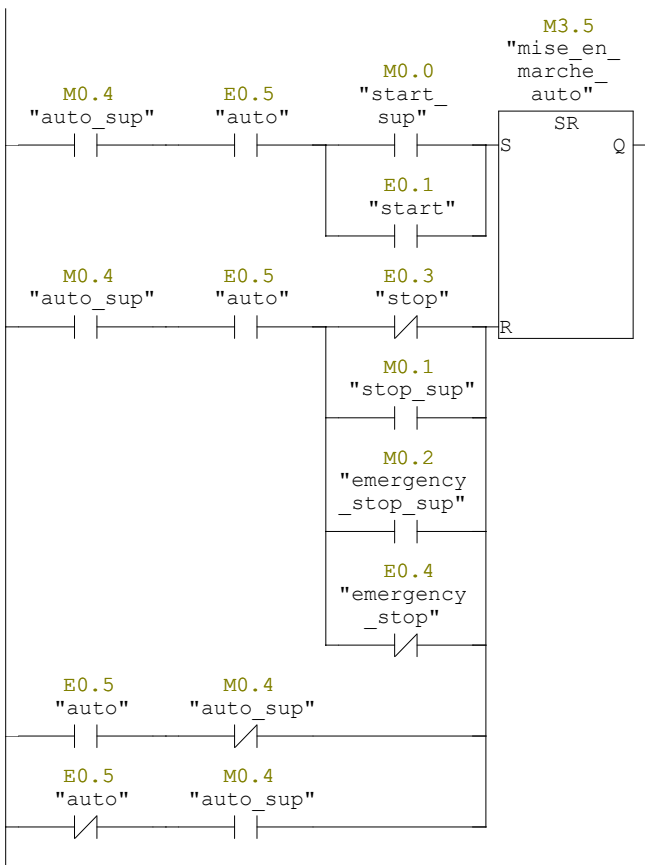
Réseau : 2



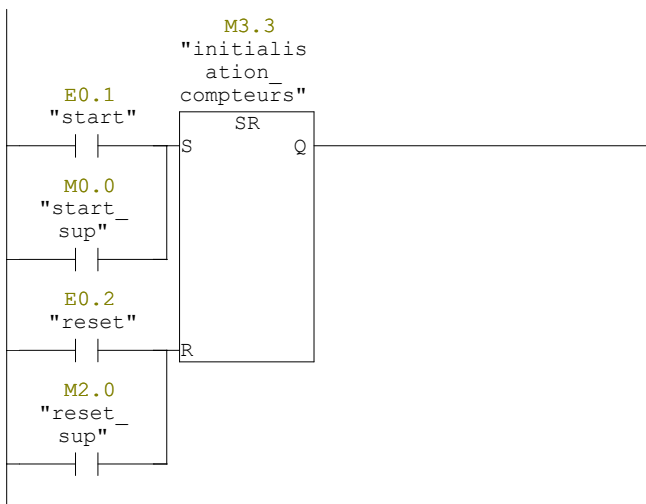
Réseau : 3



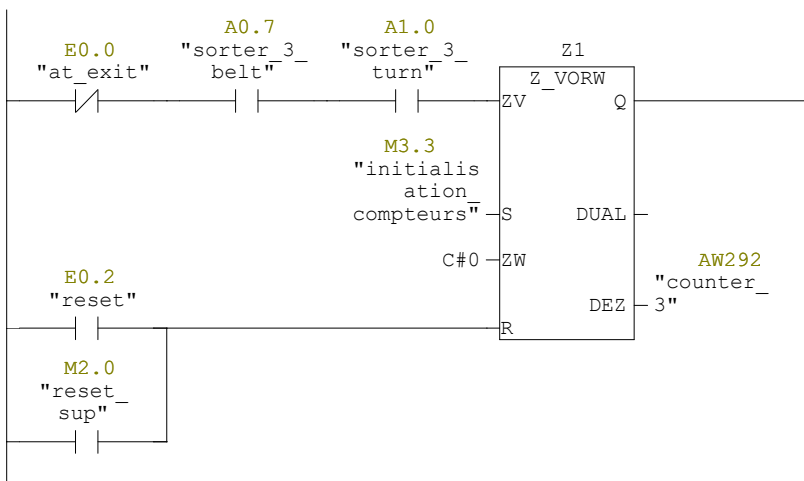
Réseau : 4



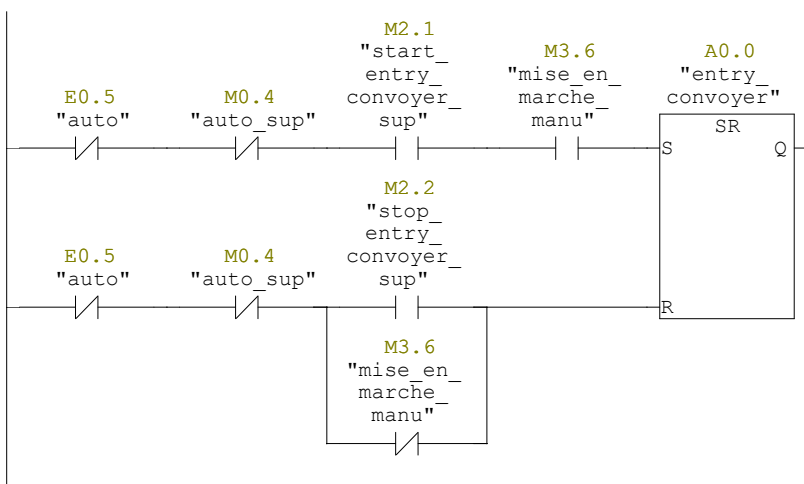
Réseau : 5



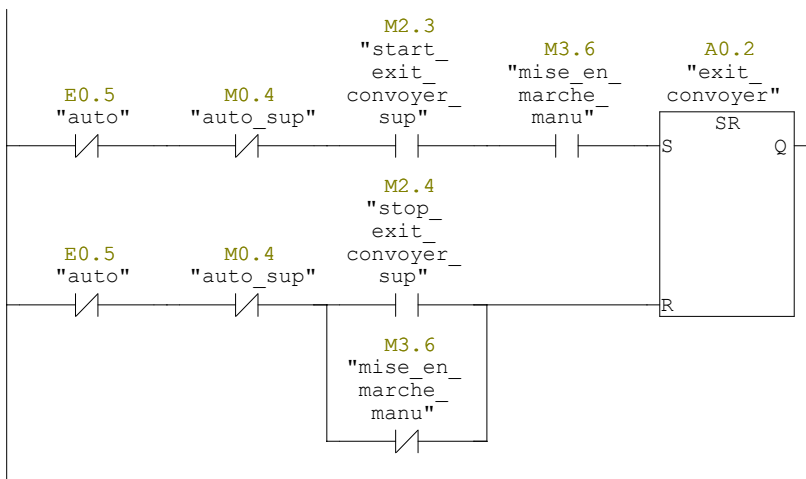
Réseau : 6



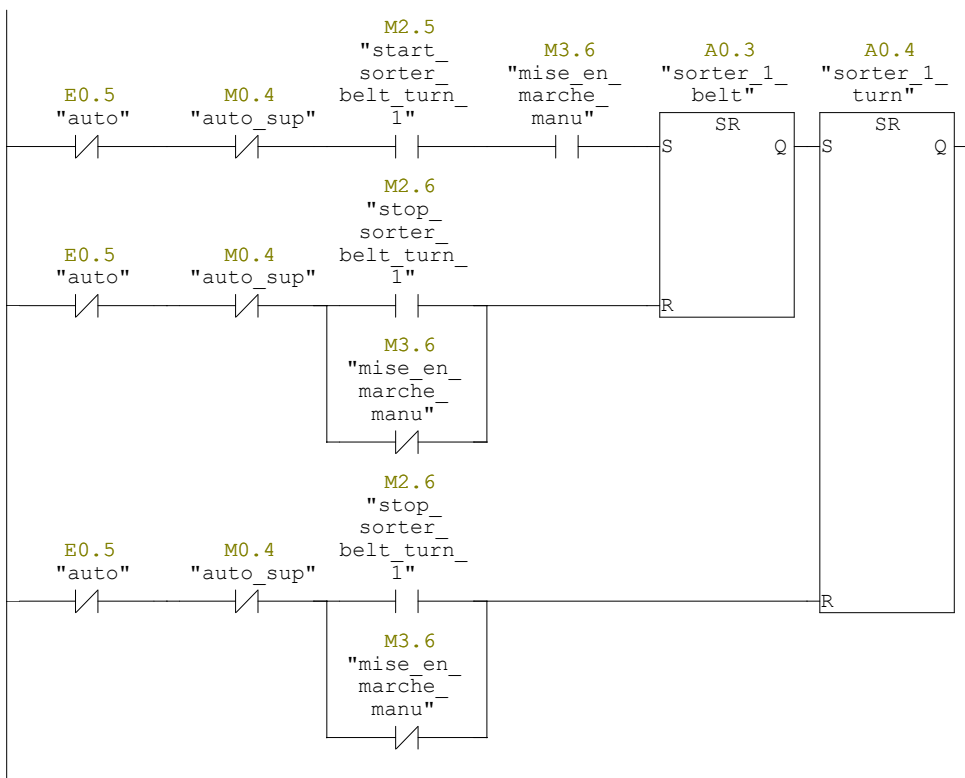
Réseau : 7



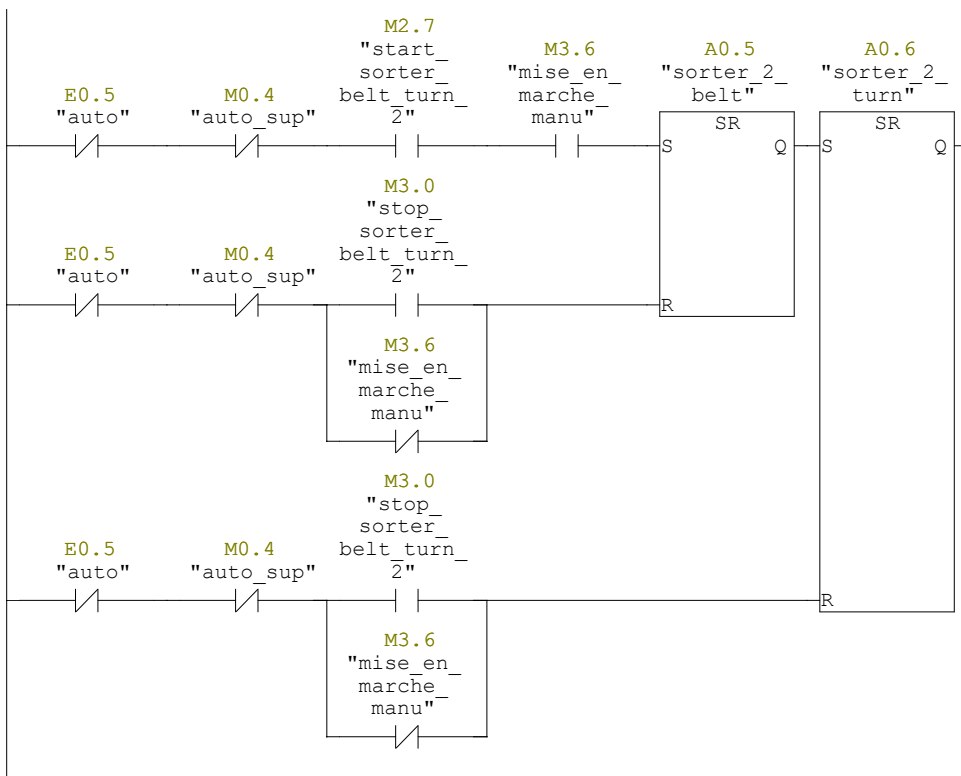
Réseau : 8



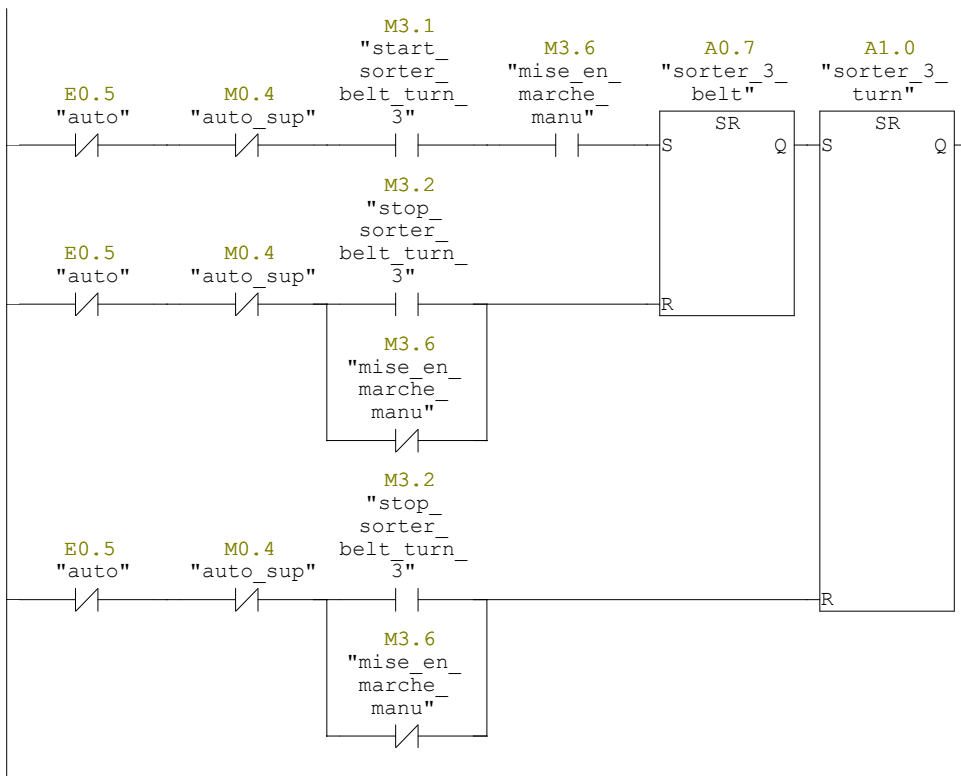
Réseau : 9



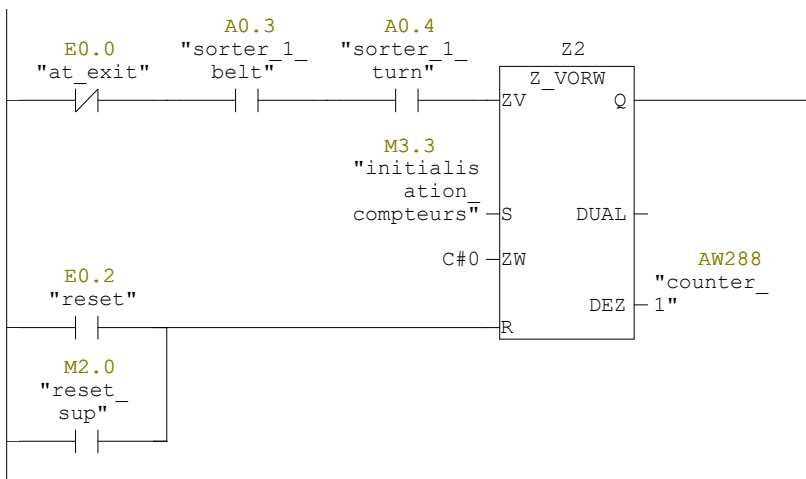
Réseau : 10



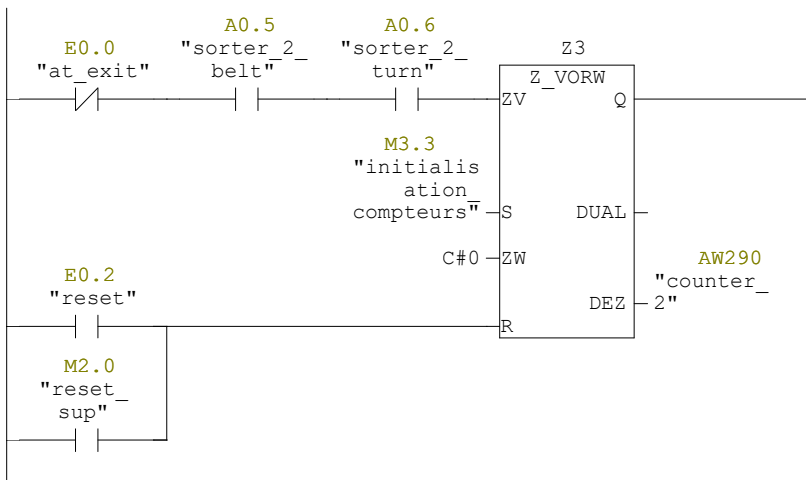
Réseau : 11



Réseau : 12



Réseau : 13



Réseau : 14



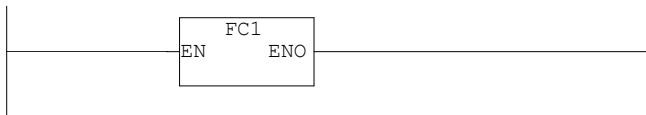
Réseau : 15



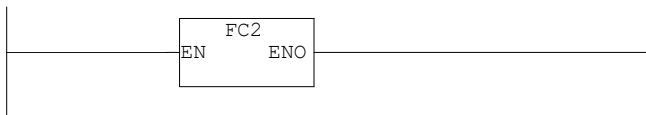
Réseau : 16



Réseau : 17



Réseau : 18



FC1 - <hors ligne>

""

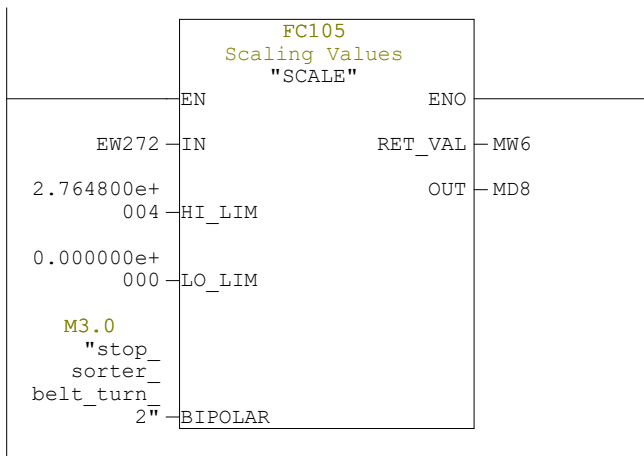
Nom :
Auteur :
Horodatage Code :
Interface :
Longueur (bloc/code /données locales) :

Famille :
Version : 0.1
Version de bloc : 2
 26/05/2024 11:43:22
 20/01/2024 05:18:20
 00260 00154 00010

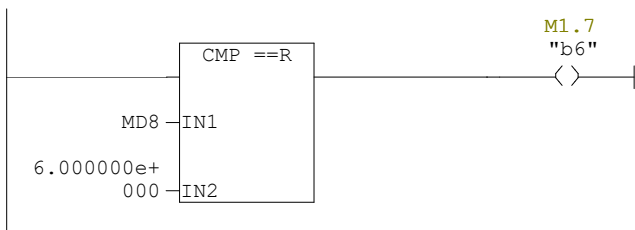
Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC1

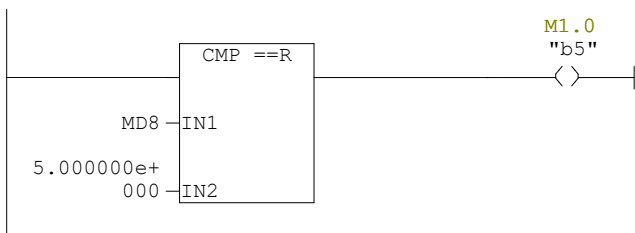
Réseau : 1



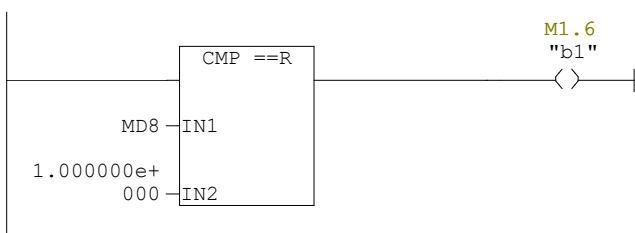
Réseau : 2



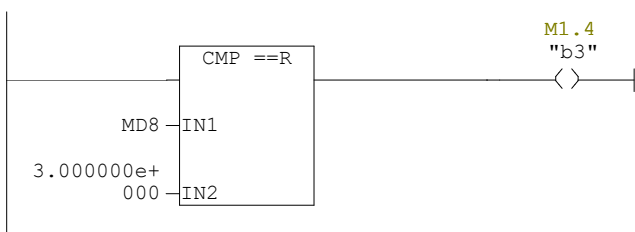
Réseau : 3



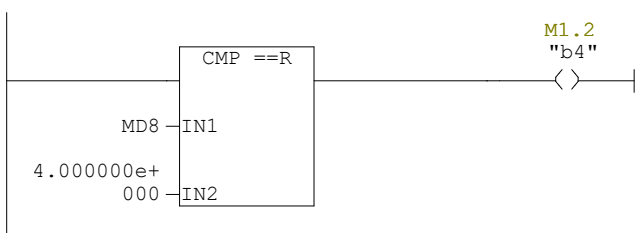
Réseau : 4



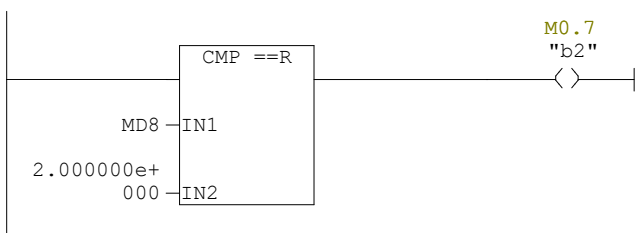
Réseau : 5



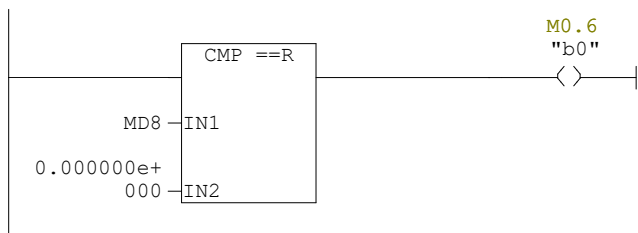
Réseau : 6



Réseau : 7



Réseau : 8



FC2 - <hors ligne>

""

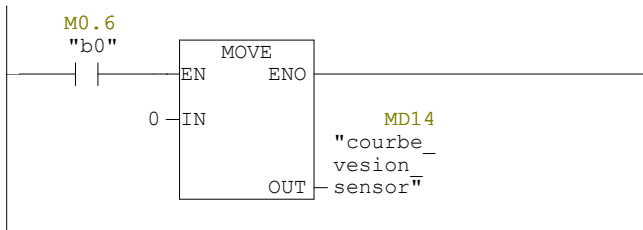
Nom :
Auteur :
Horodatage Code :
Interface :
Longueur (bloc/code /données locales) :

Famille :
Version : 0.1
Version de bloc : 2
 01/06/2024 11:41:45
 01/06/2024 11:39:17
 00270 00146 00000

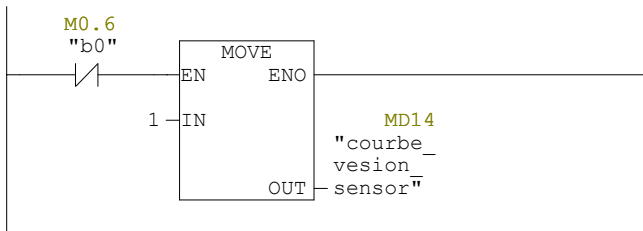
Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC2

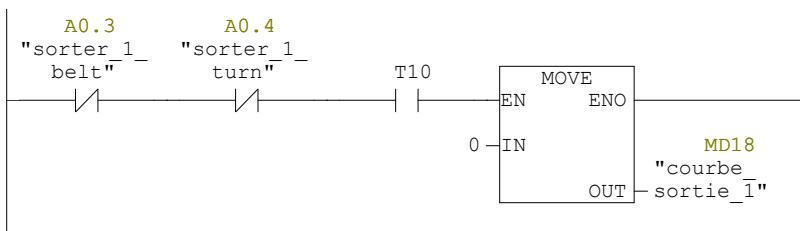
Réseau : 1



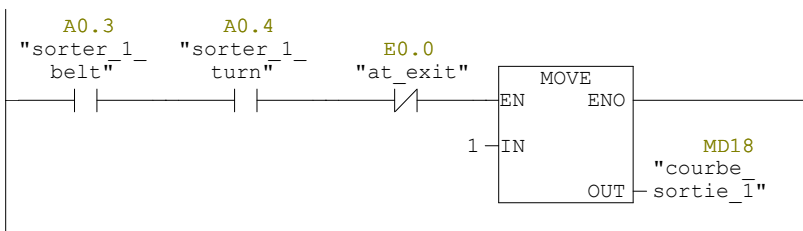
Réseau : 2



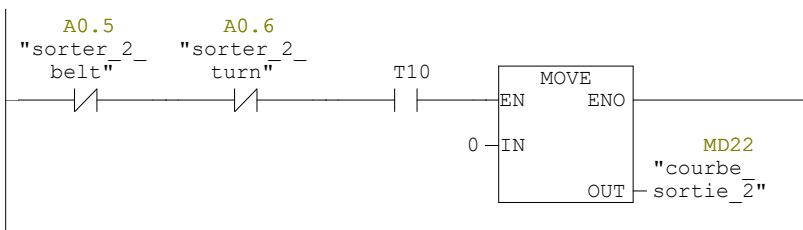
Réseau : 3



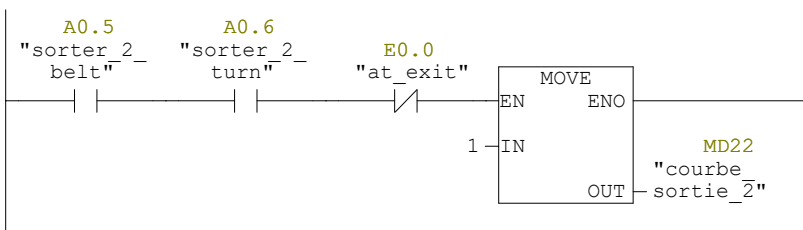
Réseau : 4



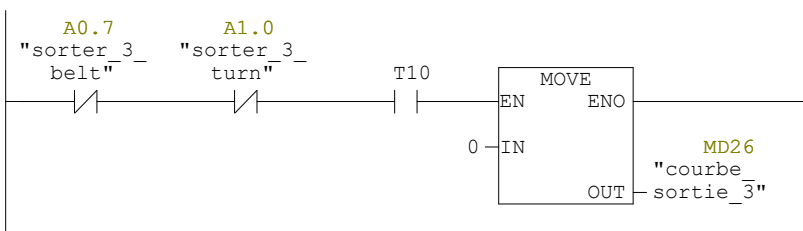
Réseau : 5



Réseau : 6



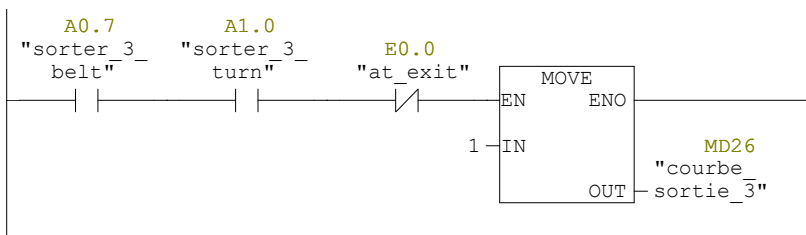
Réseau : 7



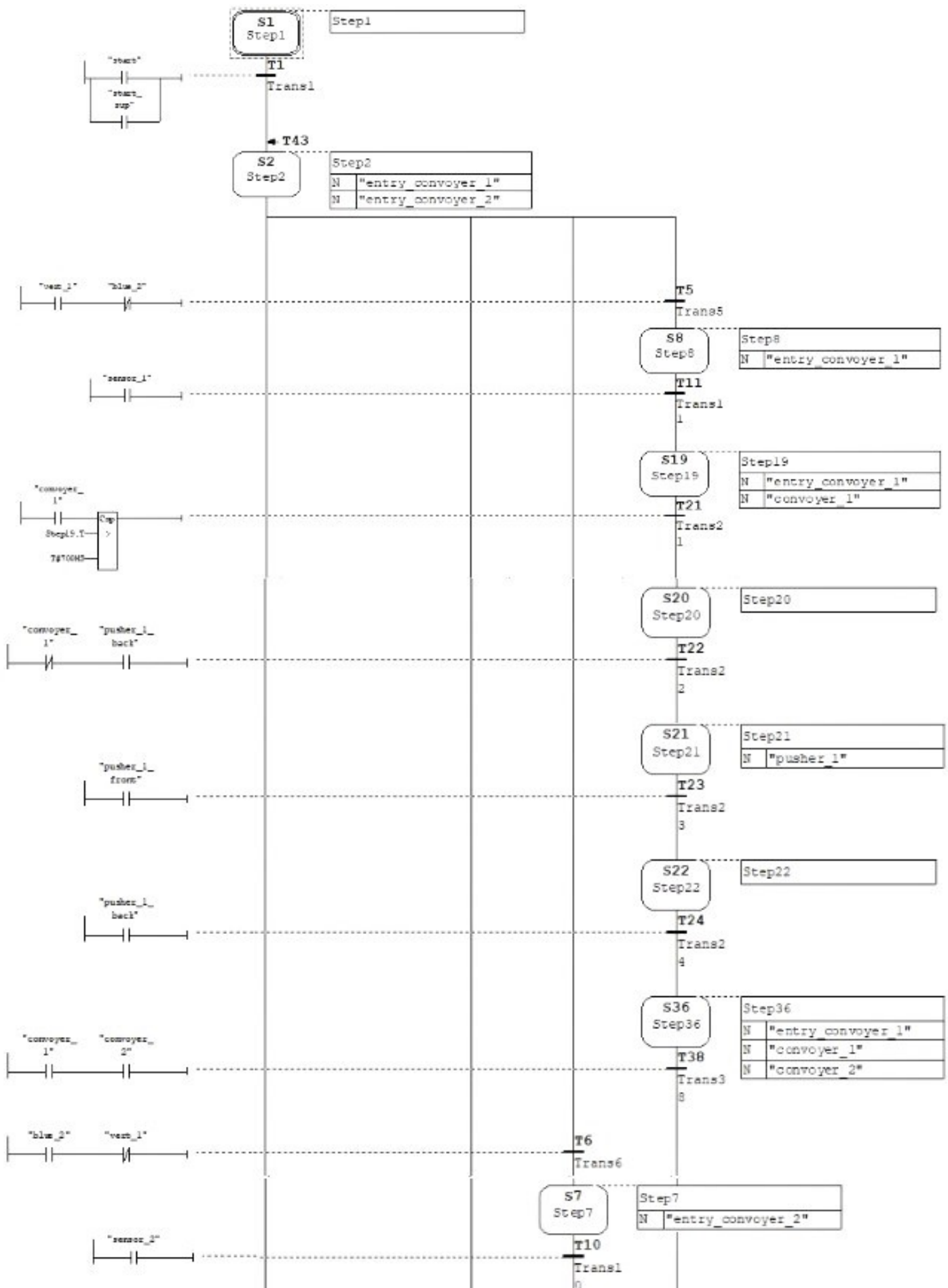
Réseau : 8

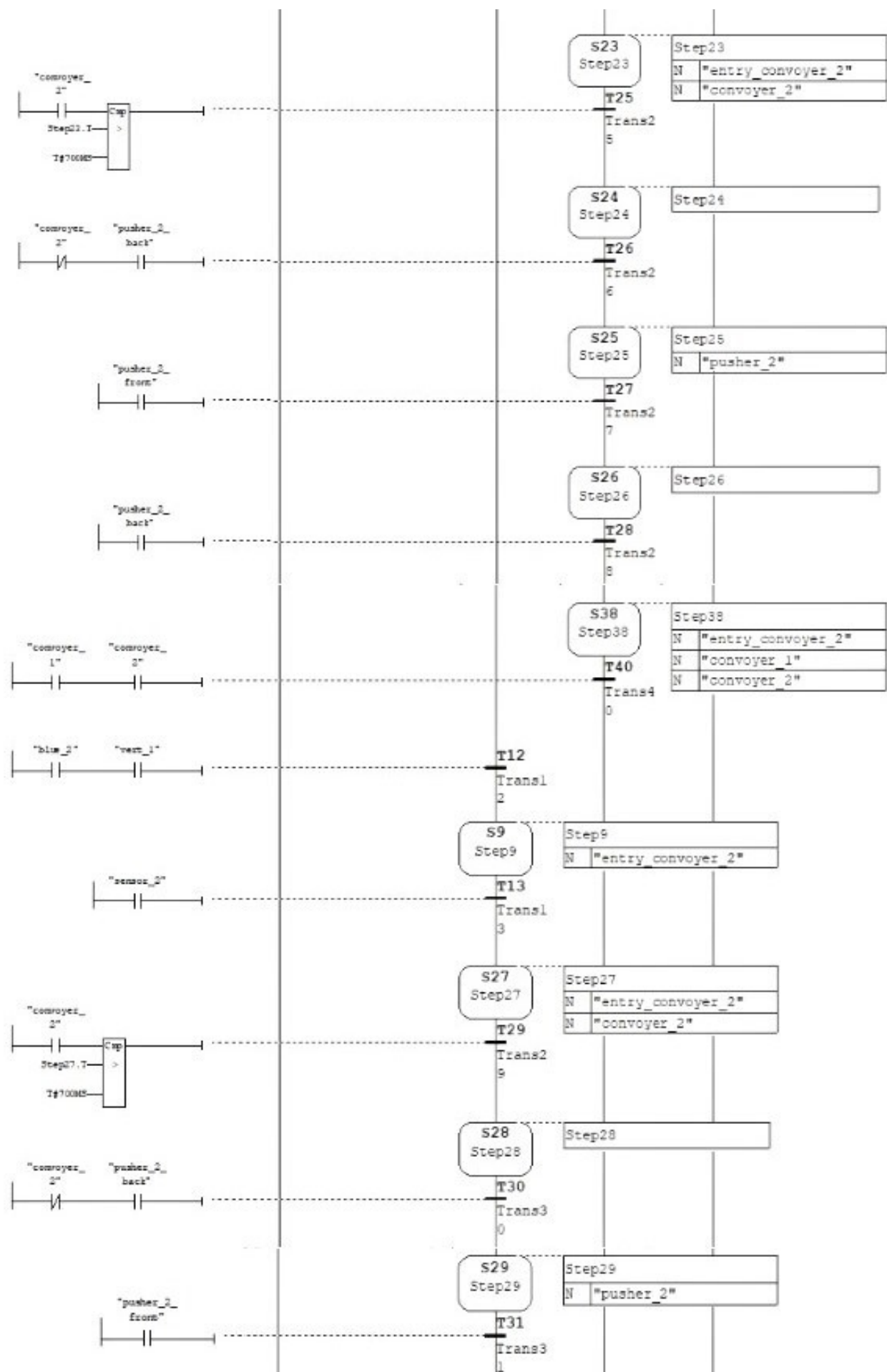


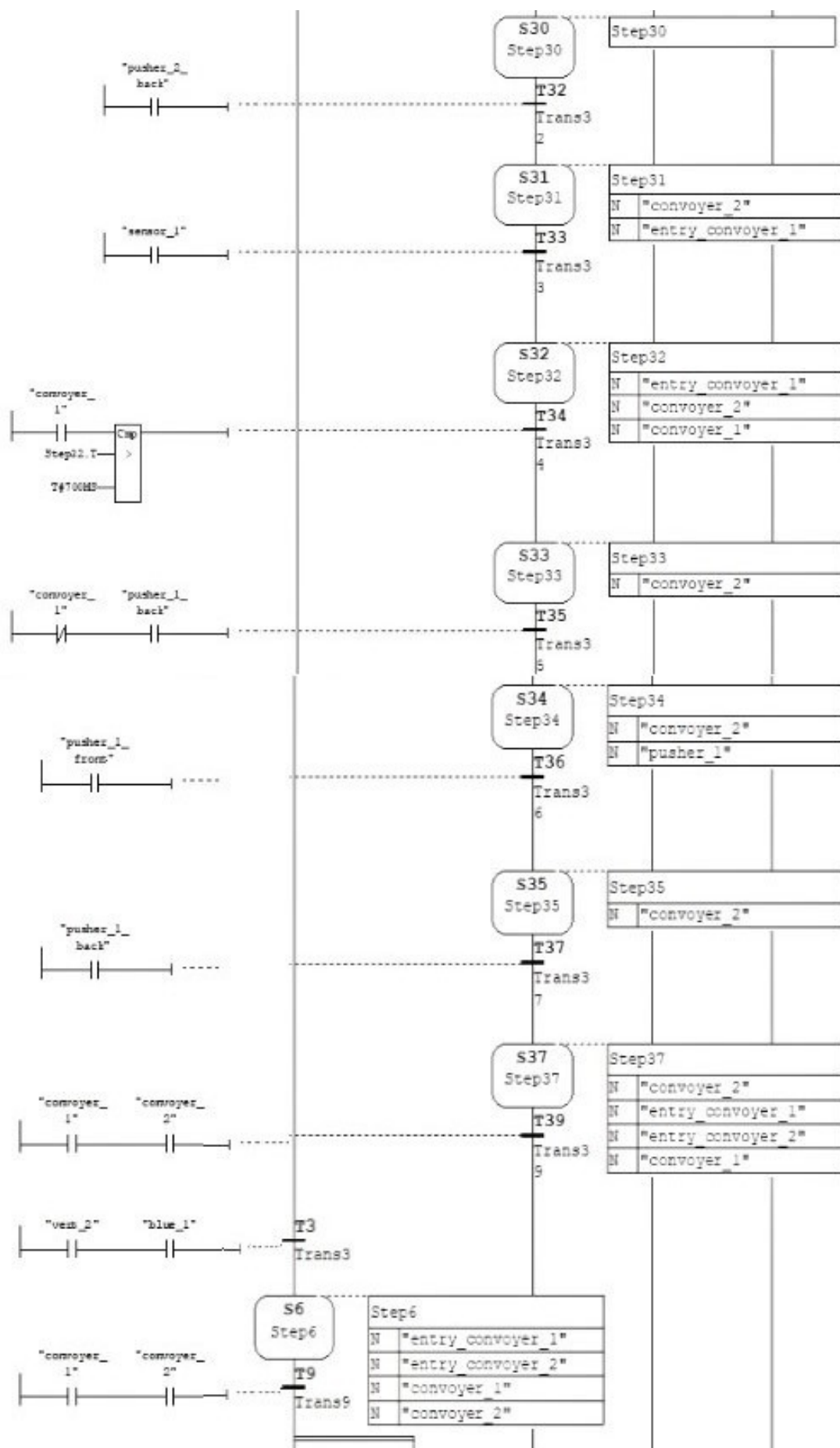
Réseau : 9

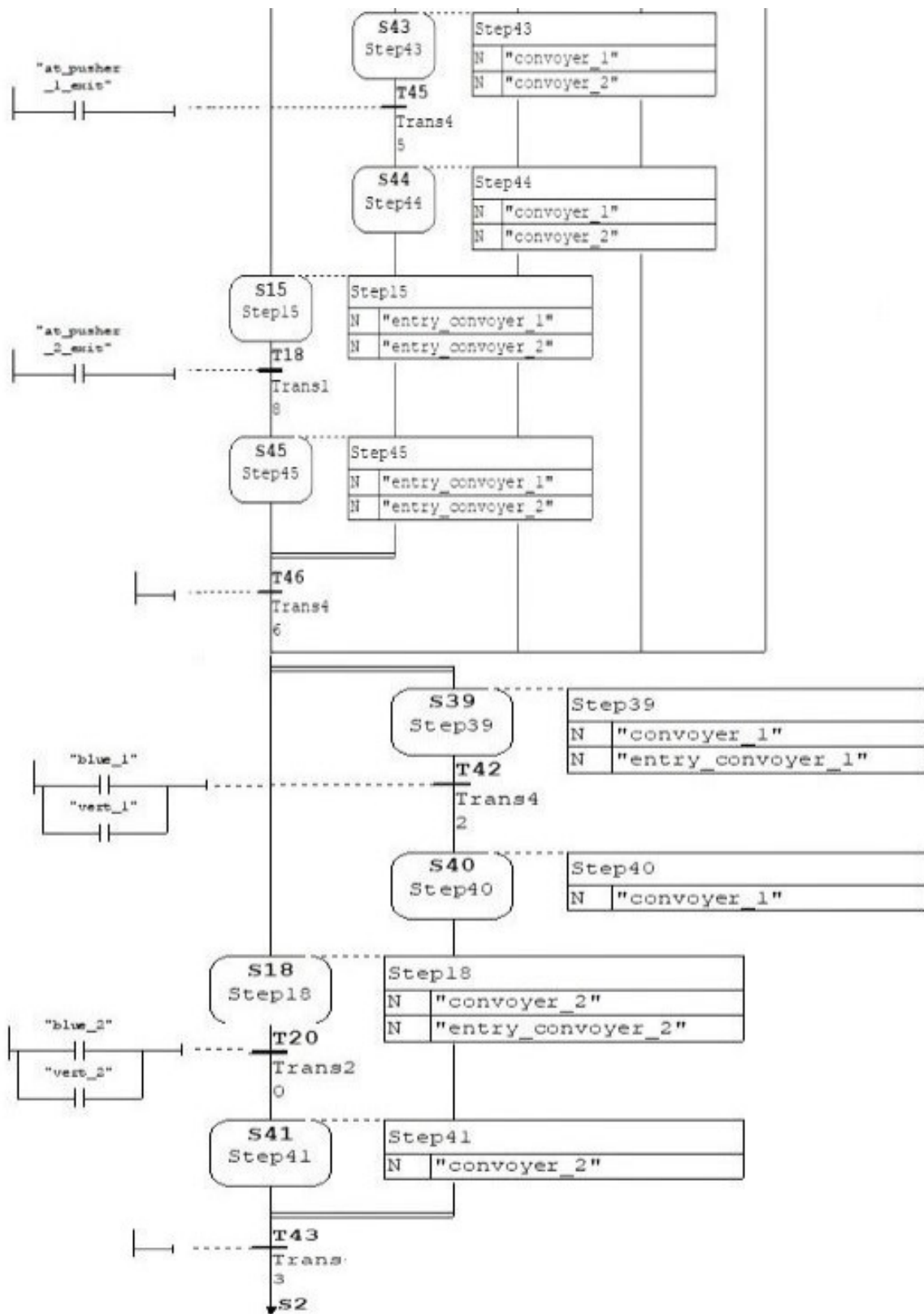


programme de la station de separation de pieces par couleur









OB1 - <hors ligne>

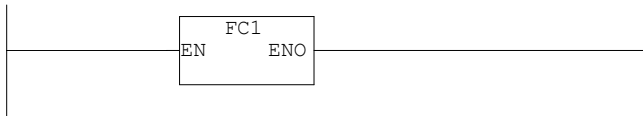
"Cycle Execution"

Nom : **Famille :**
Auteur : **Version :** 0.1
Version de bloc : 2
Horodatage Code : 01/06/2024 11:22:43
Interface : 15/02/1996 16:51:12
Longueur (bloc/code /données locales) : 00414 00234 00028

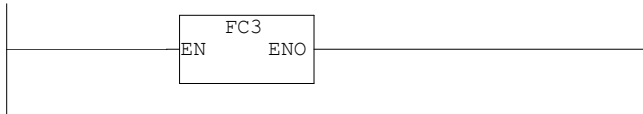
Nom	Type de données	Adresse	Commentaire
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

Bloc : OB1 "Main Program Sweep (Cycle)"

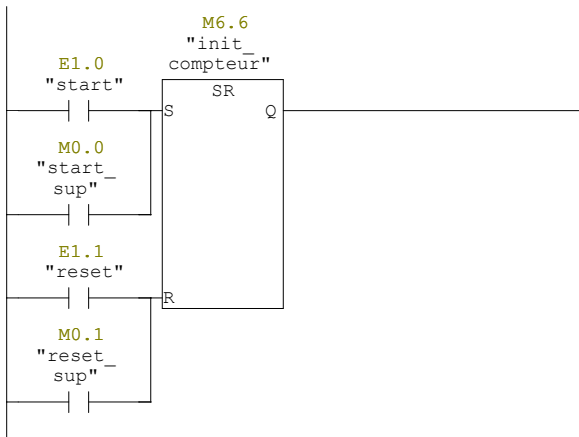
Réseau : 1



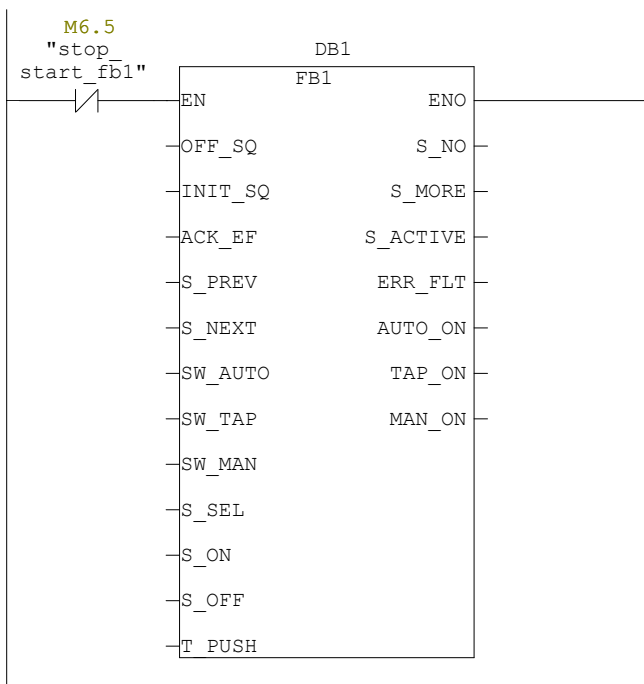
Réseau : 2



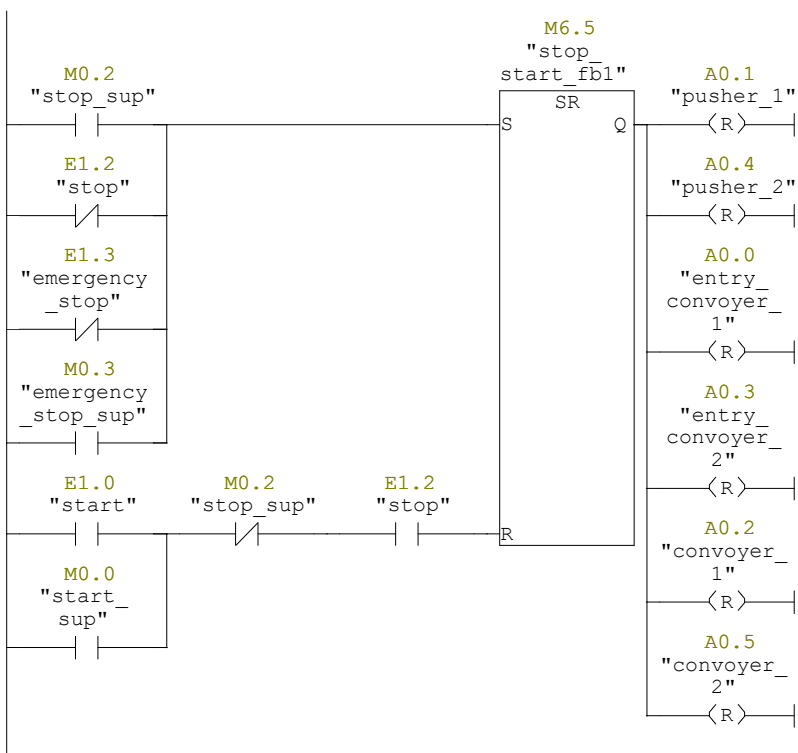
Réseau : 3



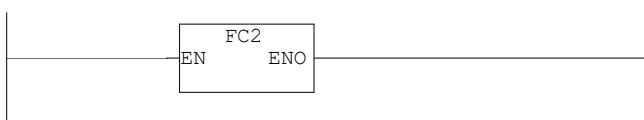
Réseau : 4



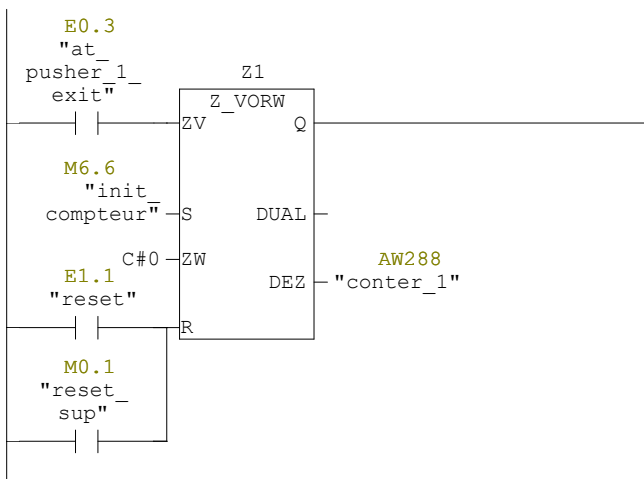
Réseau : 5



Réseau : 6



Réseau : 7



Réseau : 8



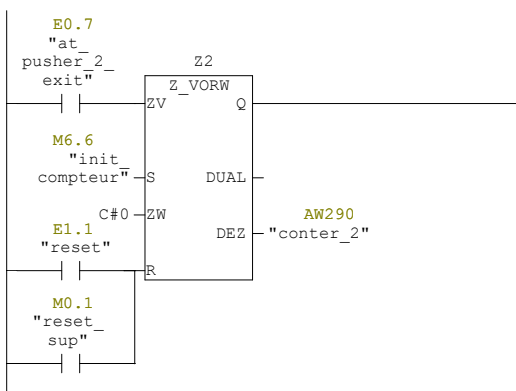
Réseau : 9



Réseau : 10



Réseau : 11



FC1 - <hors ligne>

""

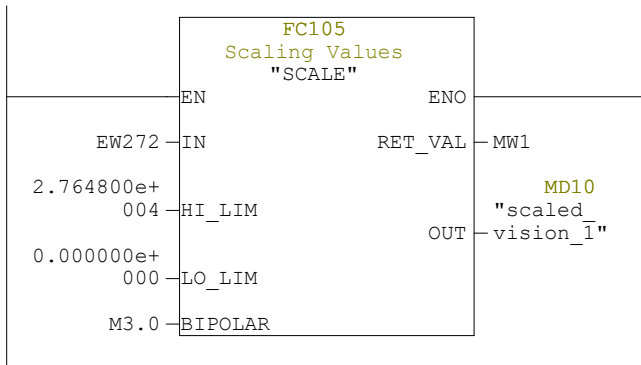
Nom :
Auteur :
Horodatage Code :
Interface :
Longueur (bloc/code /données locales) :

Famille :
Version : 0.1
Version de bloc : 2
 20/05/2024 20:03:47
 05/02/2024 16:46:59
 00190 00094 00010

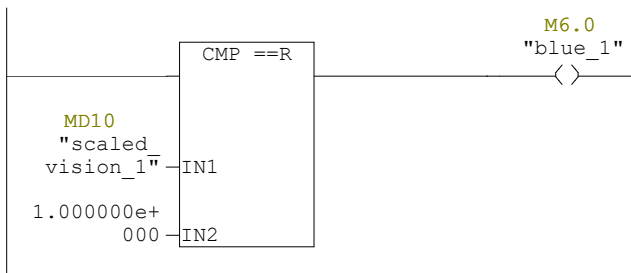
Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC1

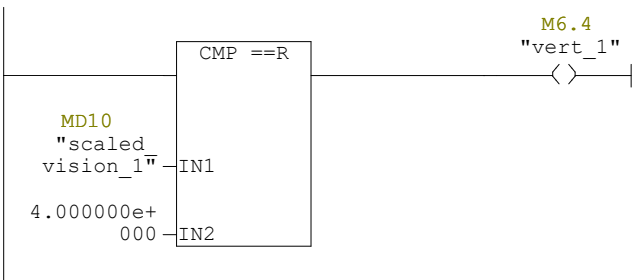
Réseau : 1



Réseau : 2



Réseau : 3



FC2 - <hors ligne>

""

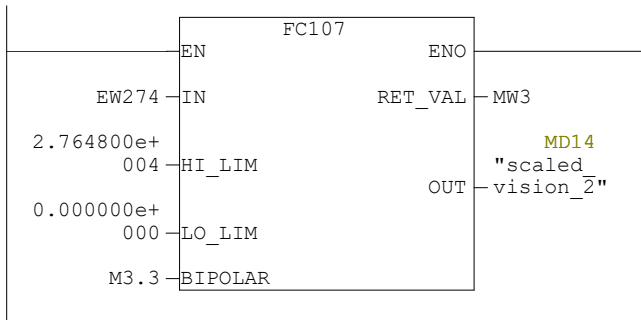
Nom :
Auteur :
Horodatage Code :
Interface :
Longueur (bloc/code /données locales) :

Famille :
Version : 0.1
Version de bloc : 2
 20/05/2024 20:03:58
 05/02/2024 16:47:09
 00190 00094 00010

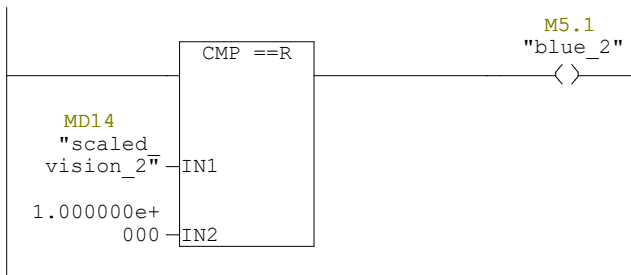
Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC2

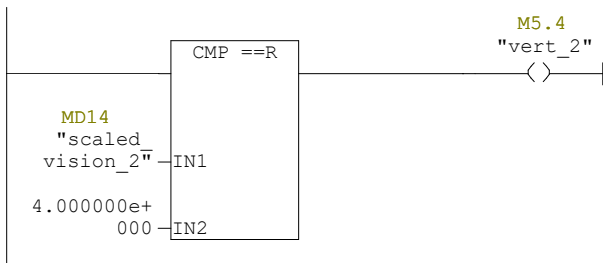
Réseau : 1



Réseau : 2



Réseau : 3



FC3 - <hors ligne>

""

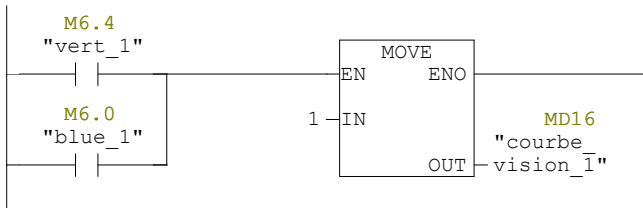
Nom :
Auteur :
Horodatage Code :
Interface :
Longueur (bloc/code /données locales) :

Famille :
Version : 0.1
Version de bloc : 2
 05/06/2024 12:21:00
 01/06/2024 11:21:18
 00324 00186 00000

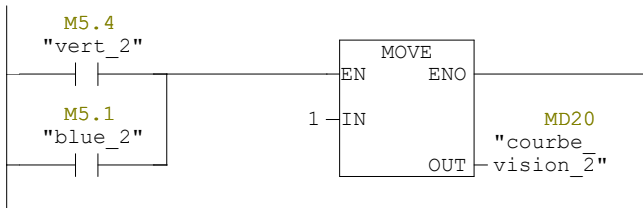
Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC3

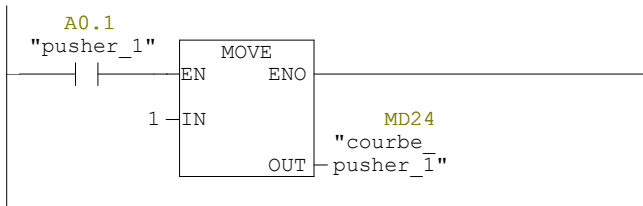
Réseau : 1



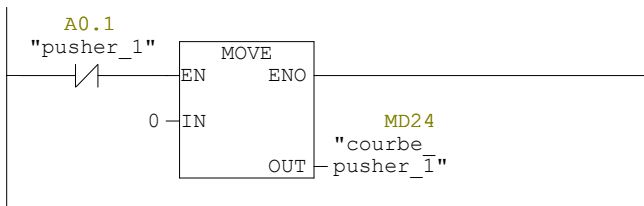
Réseau : 2



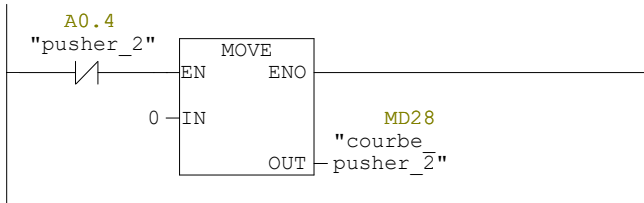
Réseau : 3



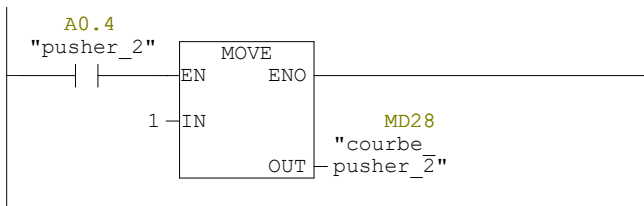
Réseau : 4



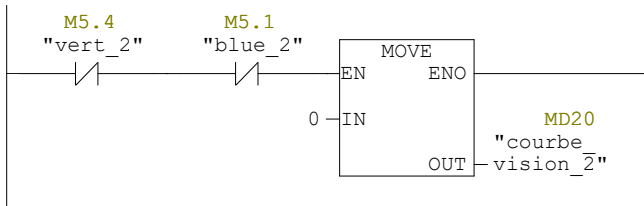
Réseau : 5



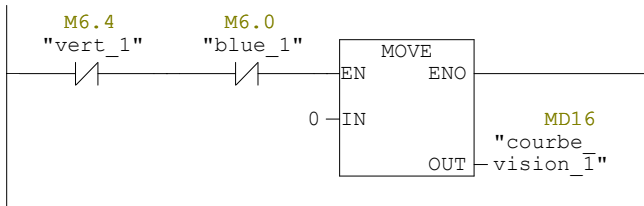
Réseau : 6



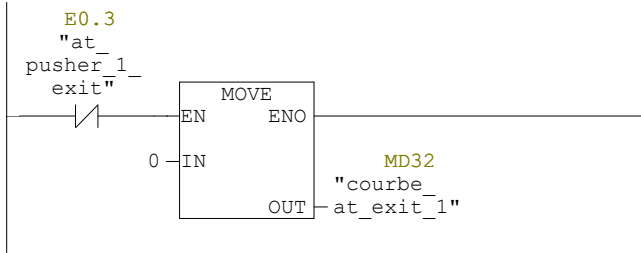
Réseau : 7



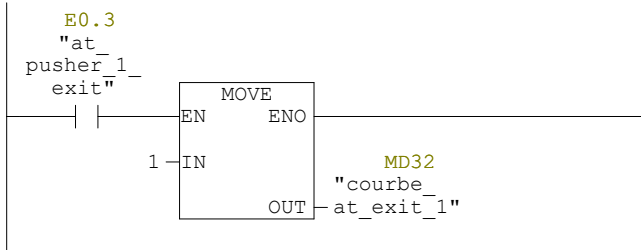
Réseau : 8



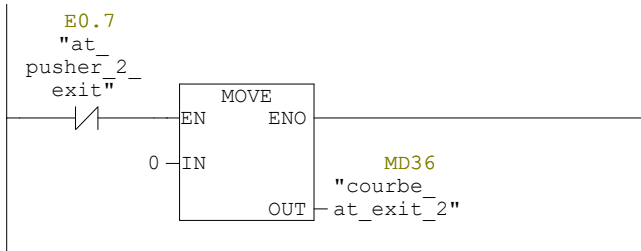
Réseau : 9



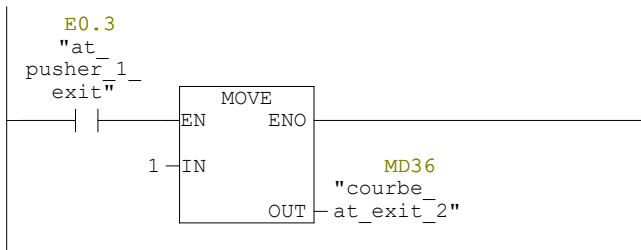
Réseau : 10



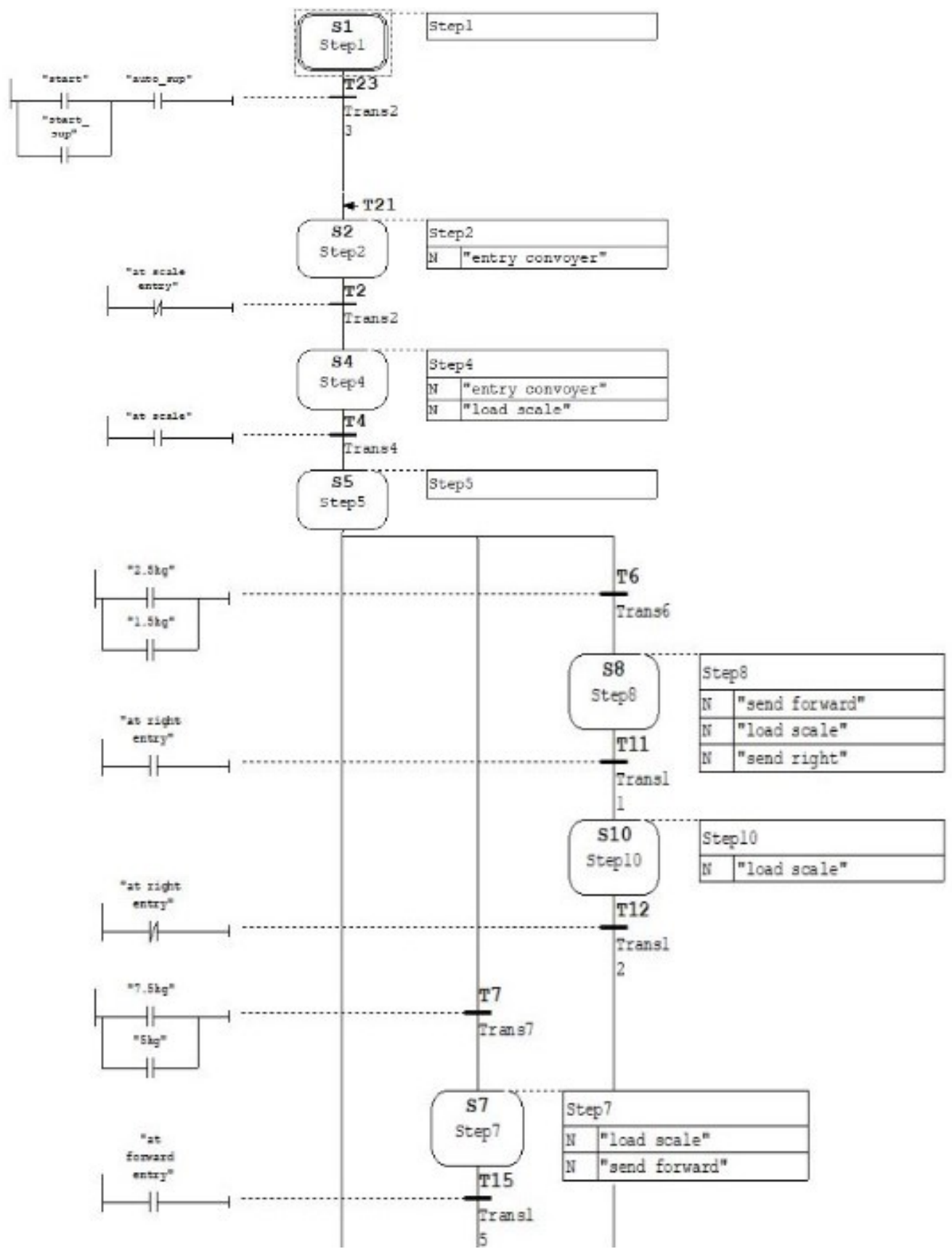
Réseau : 11

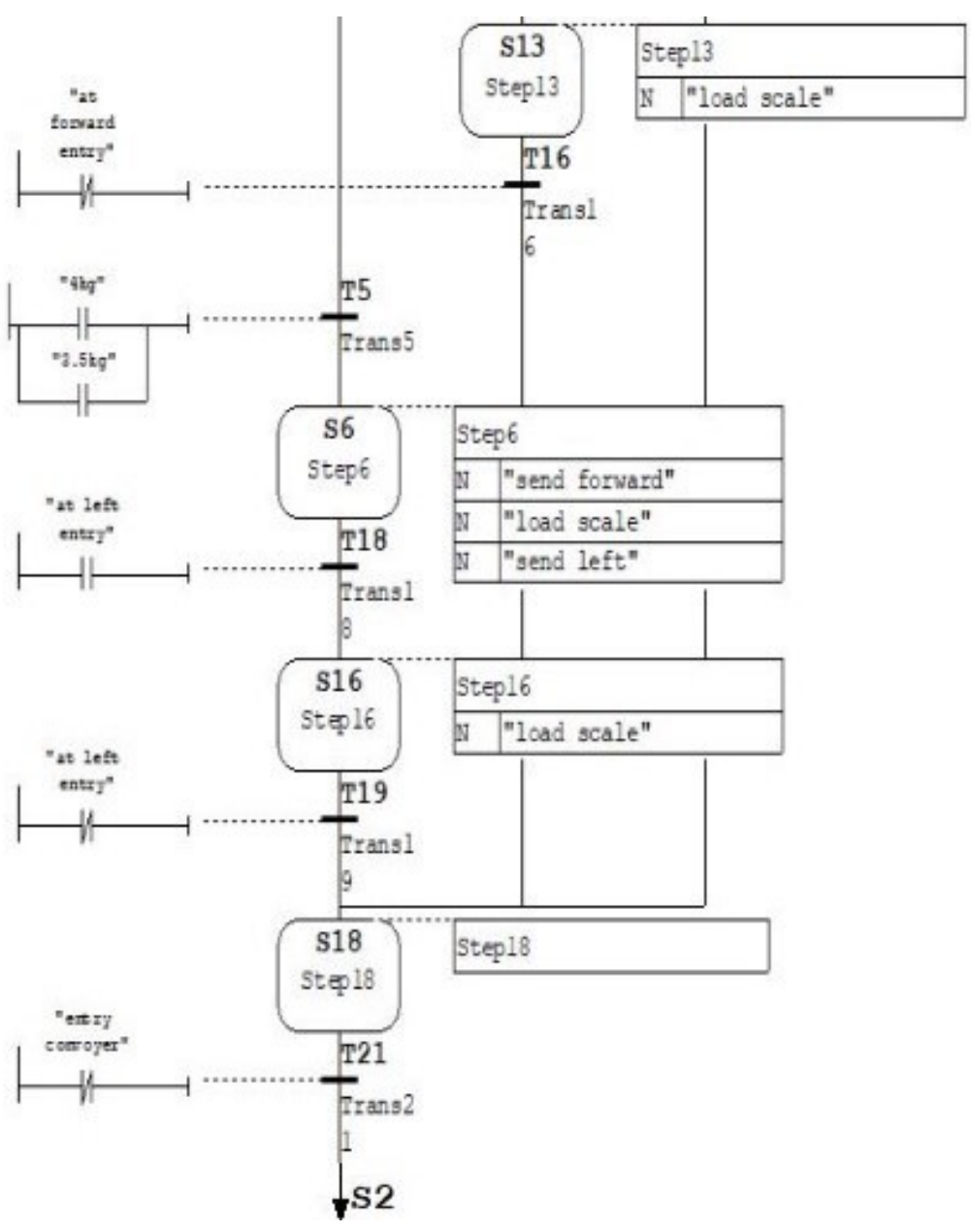


Réseau : 12



programme de la station de tri par poids :





OB1 - <hors ligne>

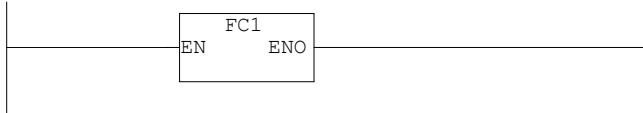
"Cycle Execution"

Nom : **Famille :**
Auteur : **Version :** 0.1
Version de bloc : 2
Horodatage Code : 01/06/2024 11:38:19
Interface : 15/02/1996 16:51:12
Longueur (bloc/code /données locales) : 01024 00802 00028

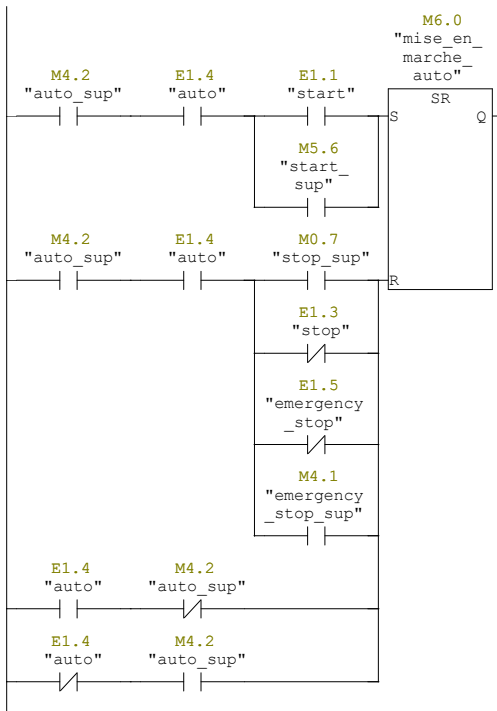
Nom	Type de données	Adresse	Commentaire
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

Bloc : OB1 "Main Program Sweep (Cycle)"

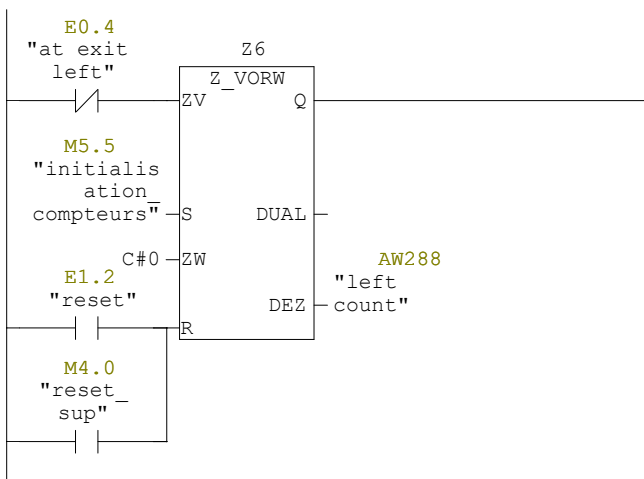
Réseau : 1



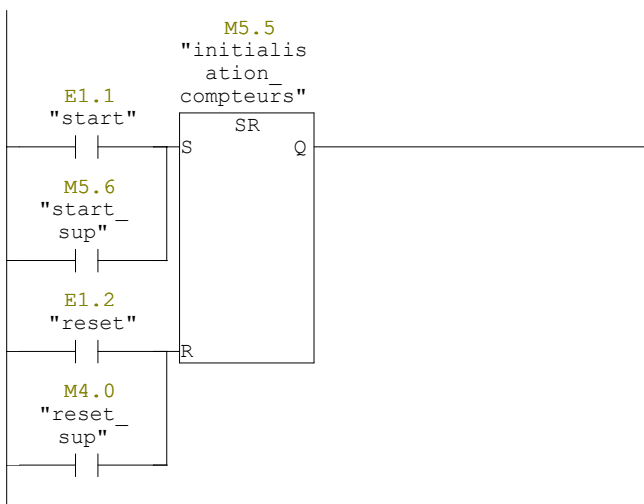
Réseau : 2



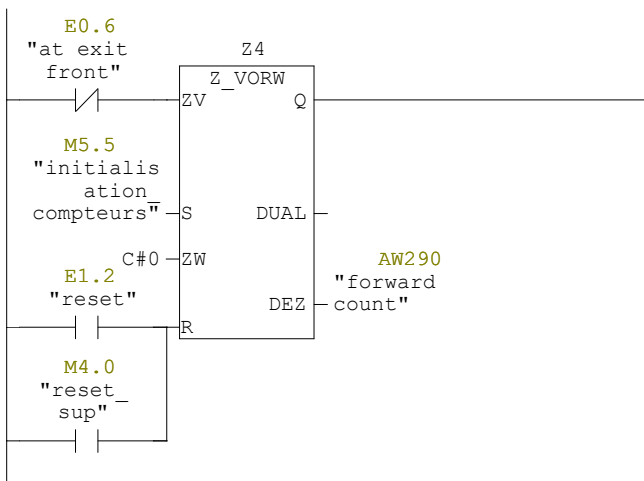
Réseau : 3



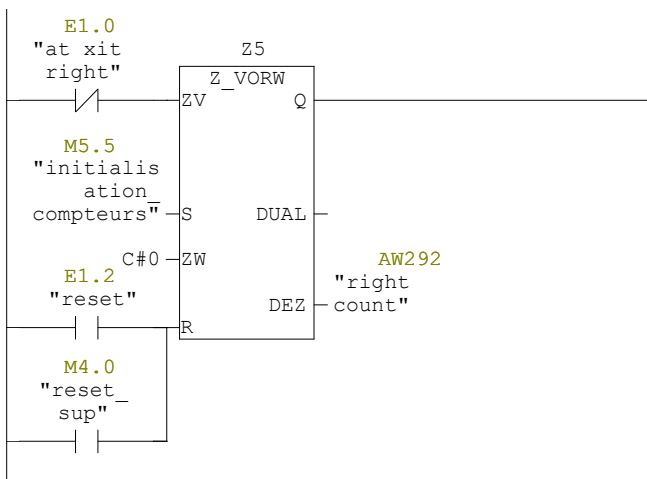
Réseau : 4



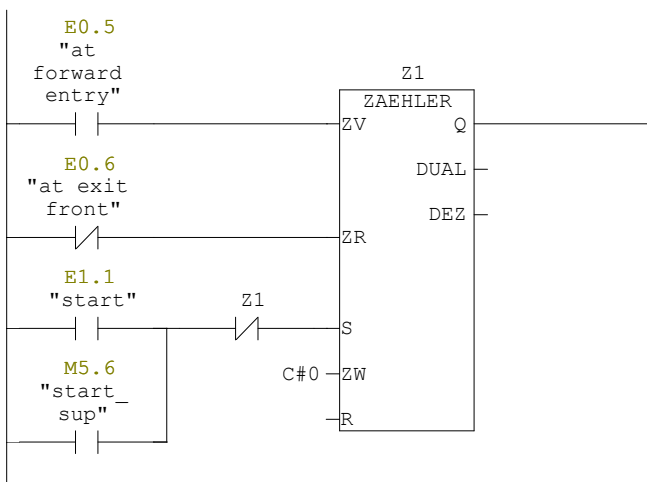
Réseau : 5



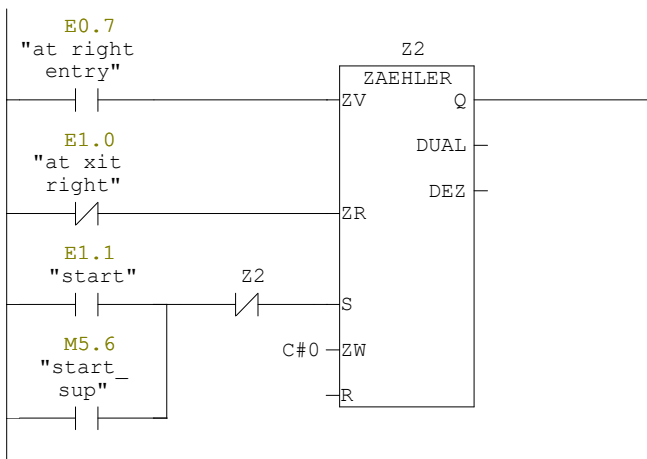
Réseau : 6



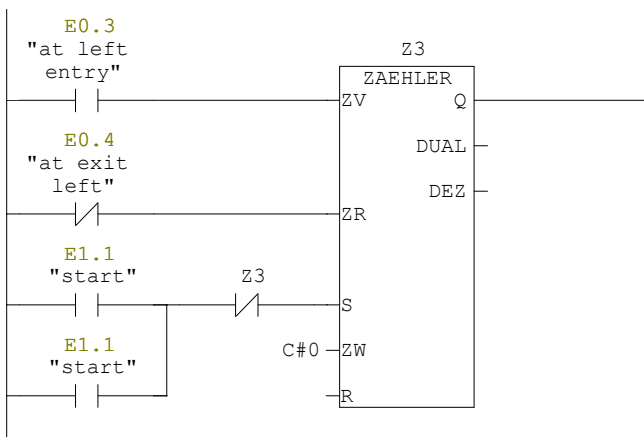
Réseau : 7



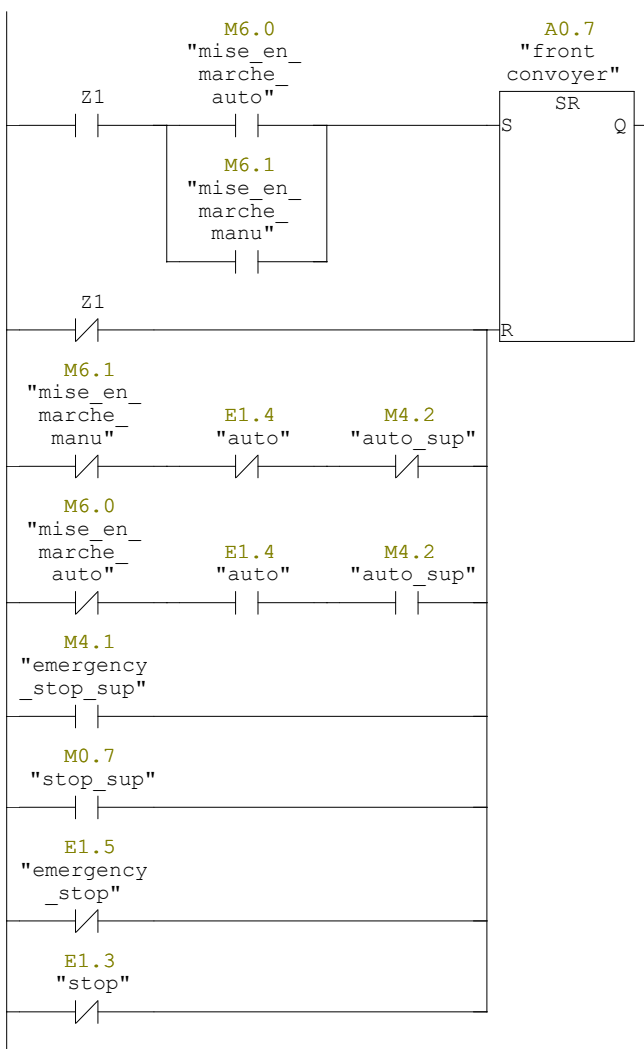
Réseau : 8



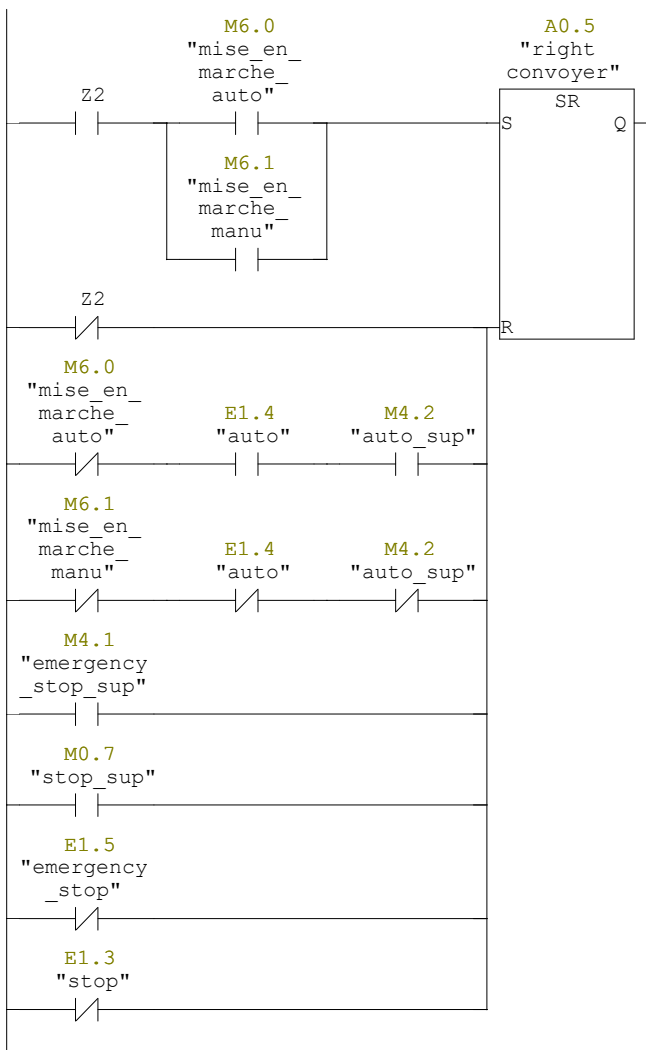
Réseau : 9



Réseau : 10



Réseau : 11



Réseau : 12



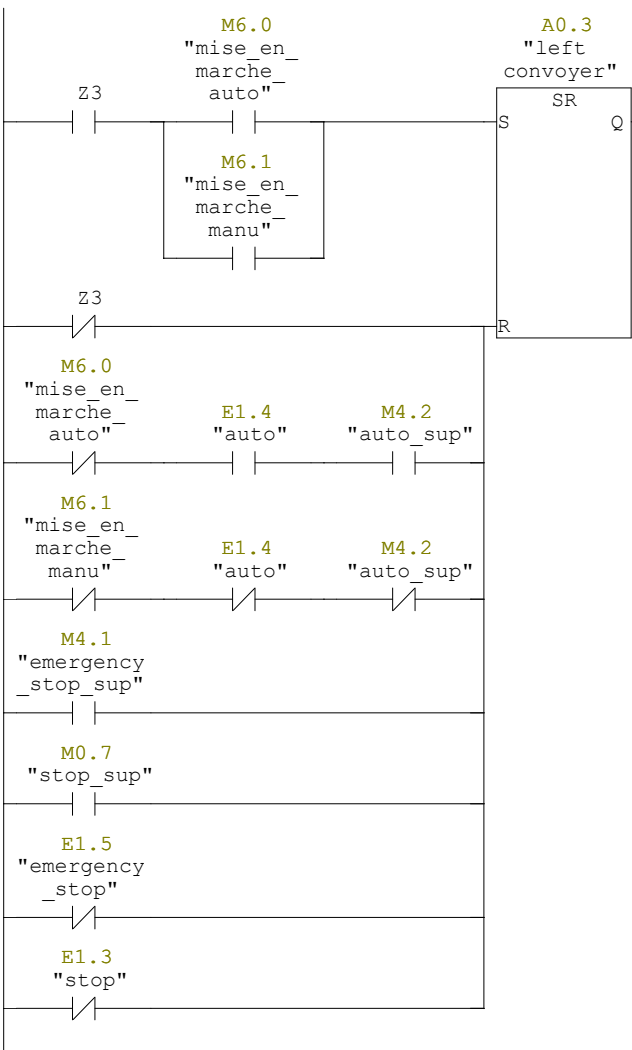
Réseau : 13



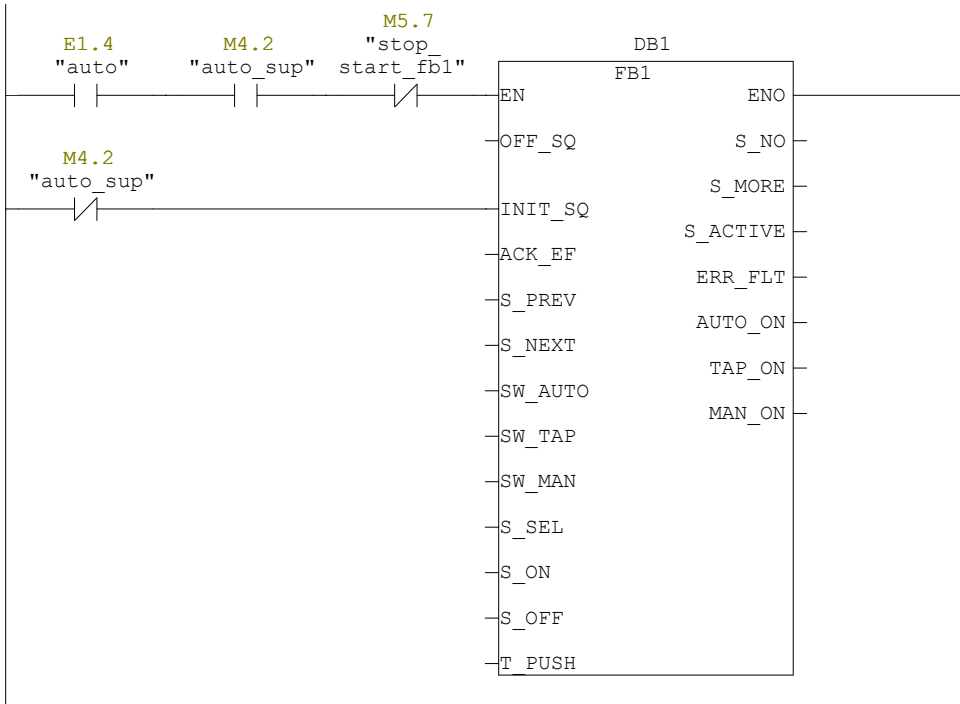
Réseau : 14



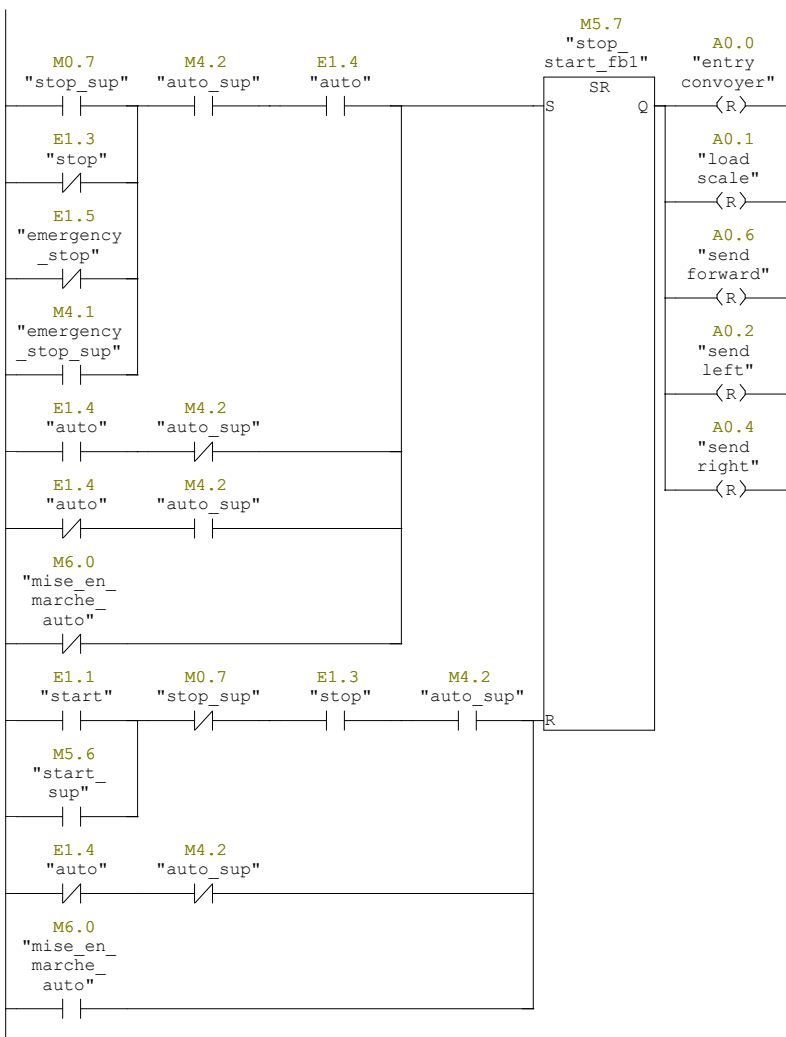
Réseau : 15



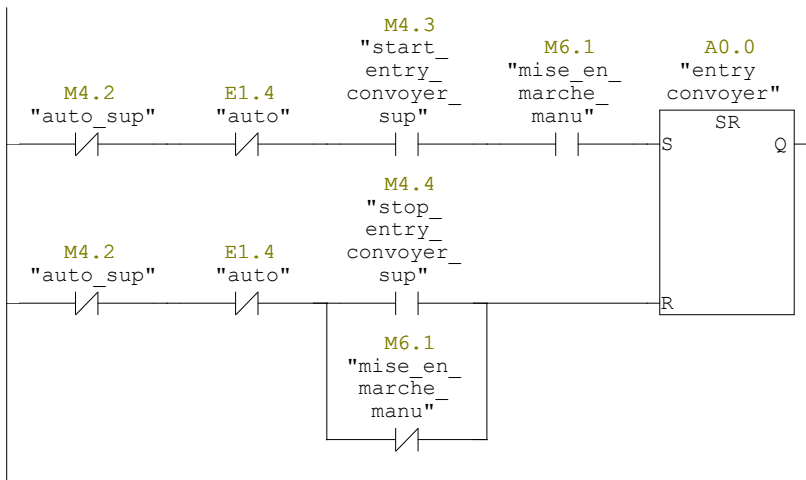
Réseau : 16



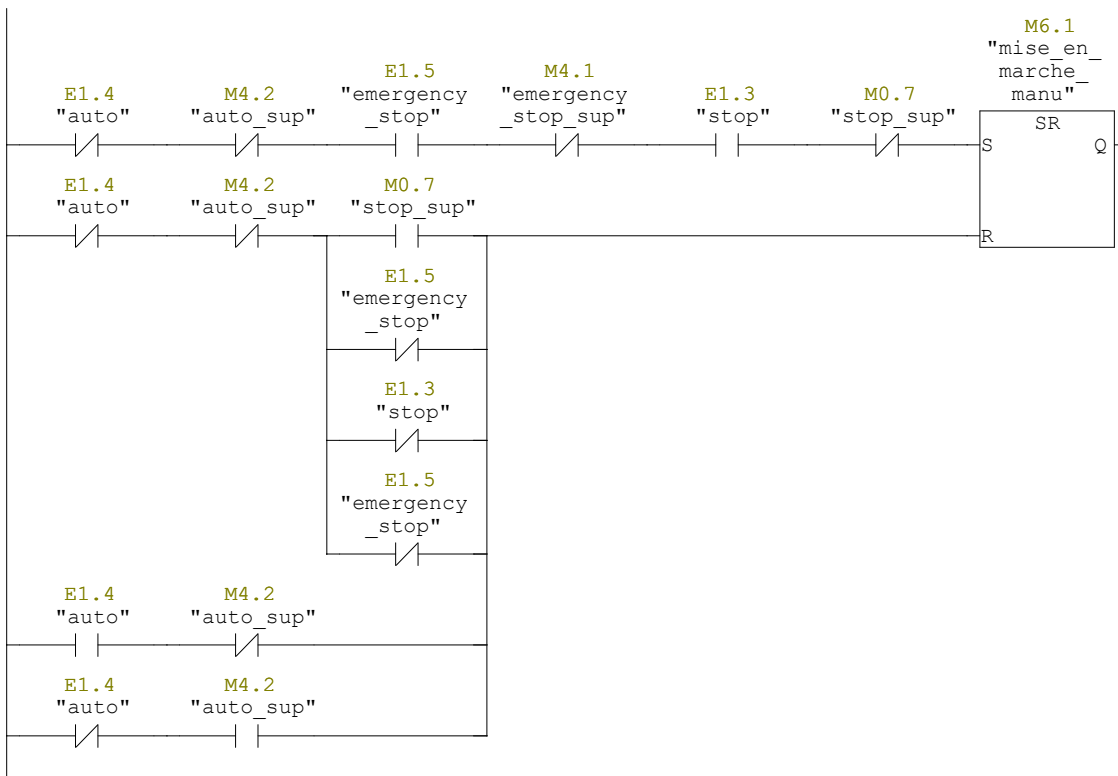
Réseau : 17



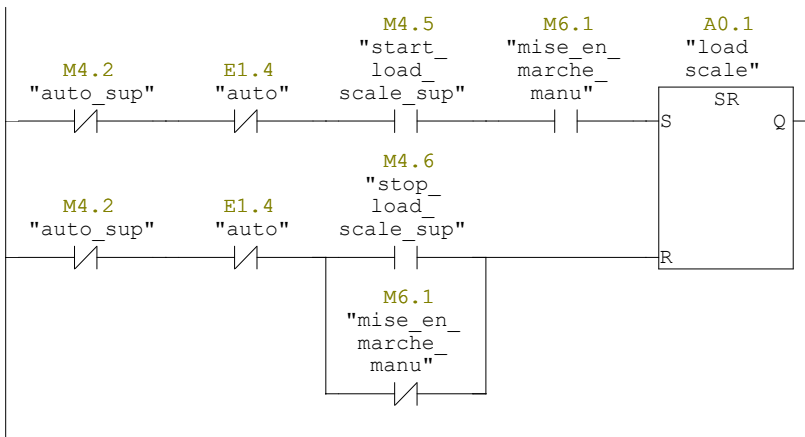
Réseau : 18



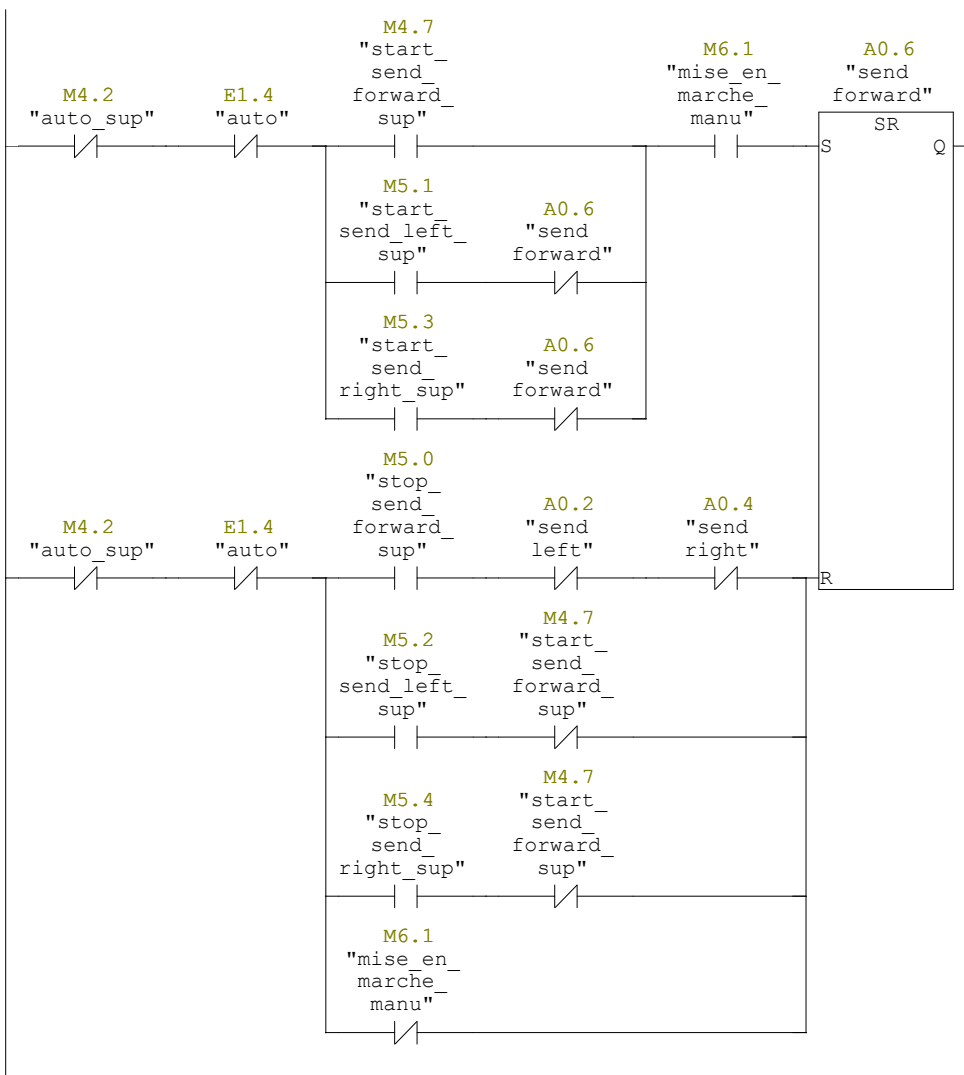
Réseau : 19



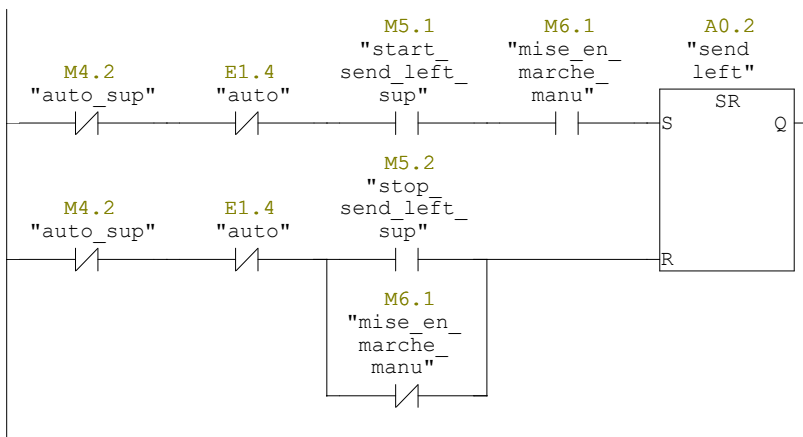
Réseau : 20



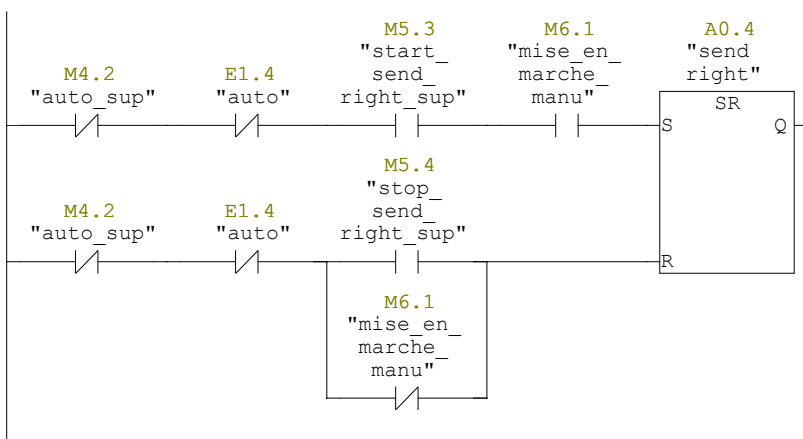
Réseau : 21



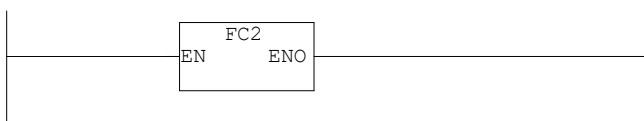
Réseau : 22



Réseau : 23



Réseau : 24



FC1 - <hors ligne>

""

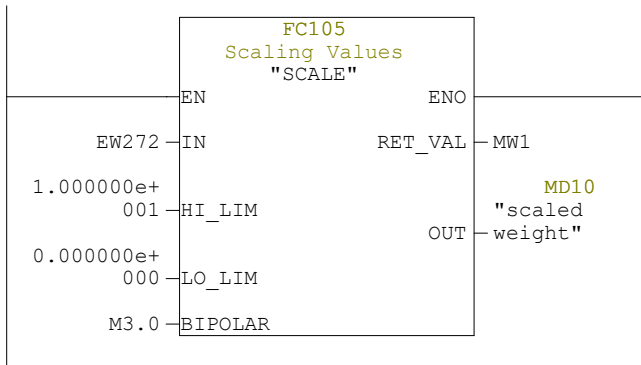
Nom :
Auteur :
Horodatage Code :
Interface :
Longueur (bloc/code /données locales) :

Famille :
Version : 0.1
Version de bloc : 2
 26/05/2024 11:22:47
 28/02/2024 05:52:36
 00338 00230 00010

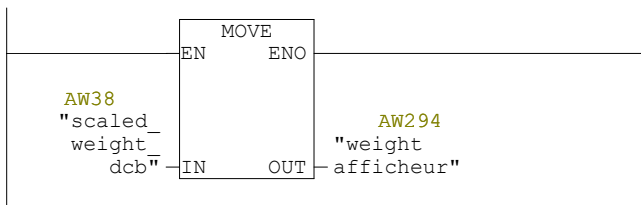
Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC1

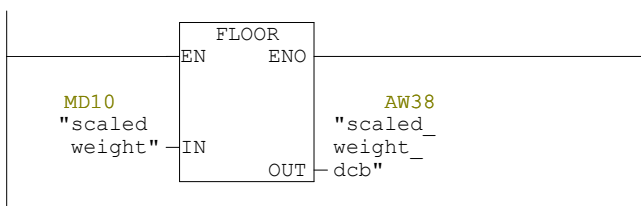
Réseau : 1



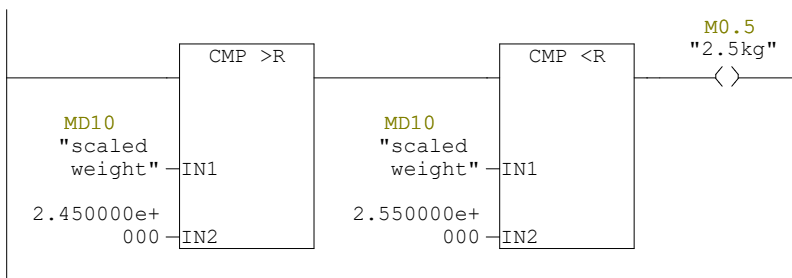
Réseau : 2



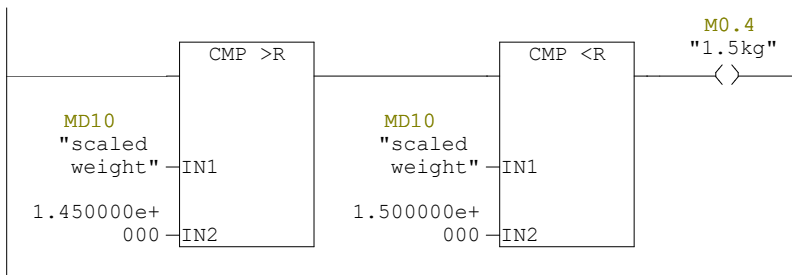
Réseau : 3



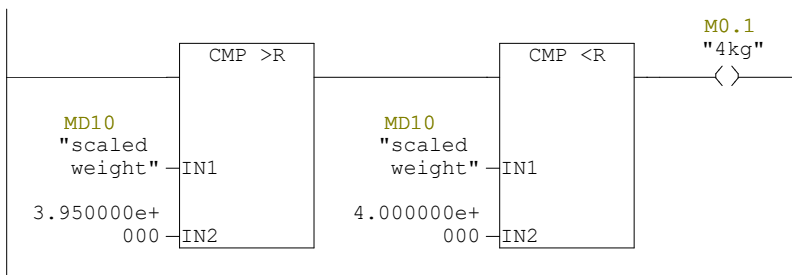
Réseau : 4



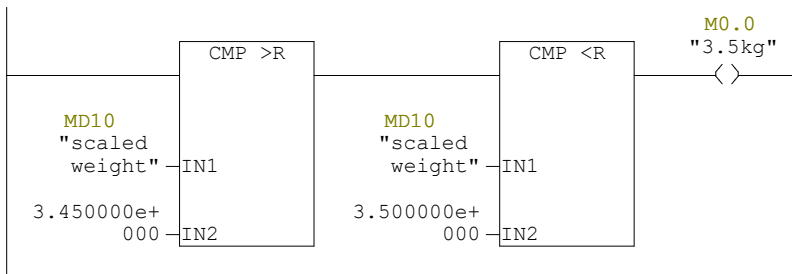
Réseau : 5



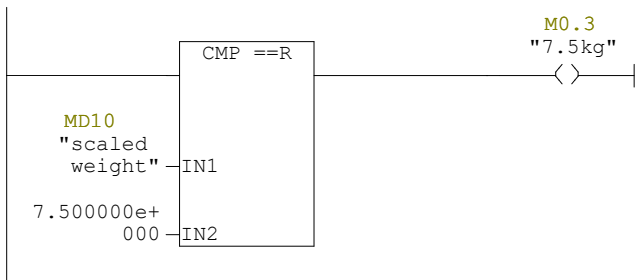
Réseau : 6



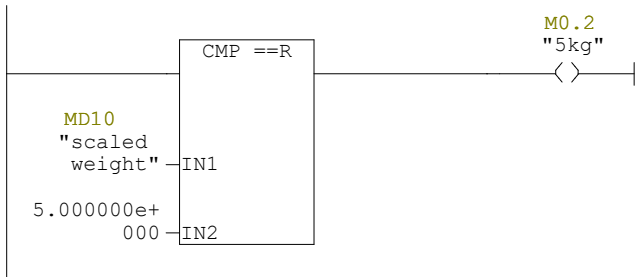
Réseau : 7



Réseau : 8



Réseau : 9



FC2 - <hors ligne>

""

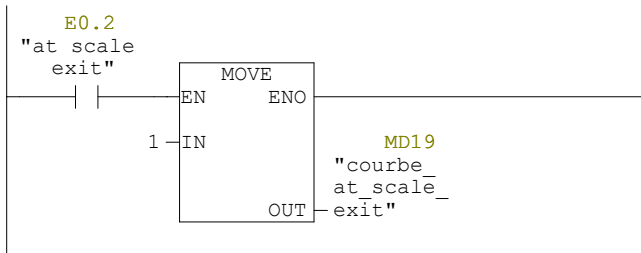
Nom :
Auteur :
Horodatage Code :
Interface :
Longueur (bloc/code /données locales) :

Famille :
Version : 0.1
Version de bloc : 2
 05/06/2024 19:06:36
 01/06/2024 11:36:58
 00410 00250 00000

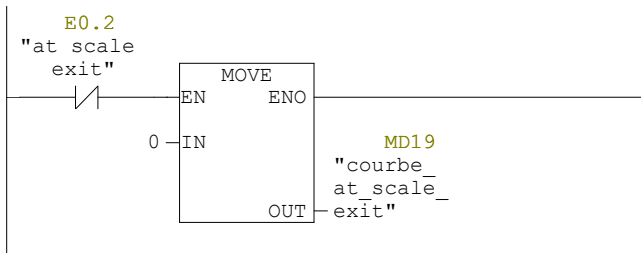
Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC2

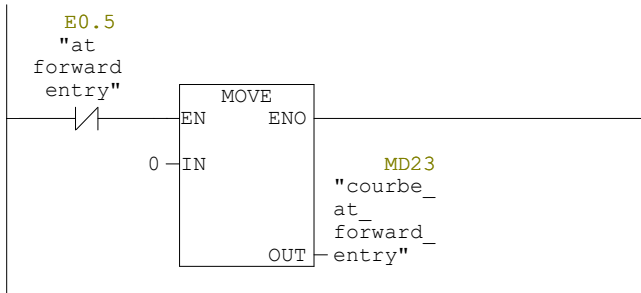
Réseau : 1



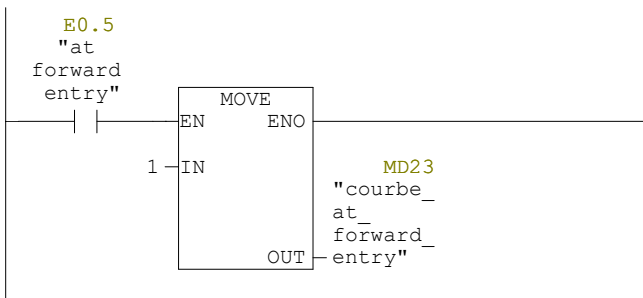
Réseau : 2



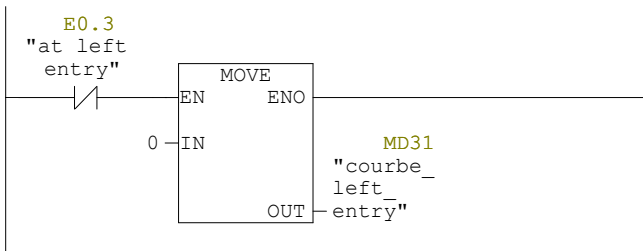
Réseau : 3



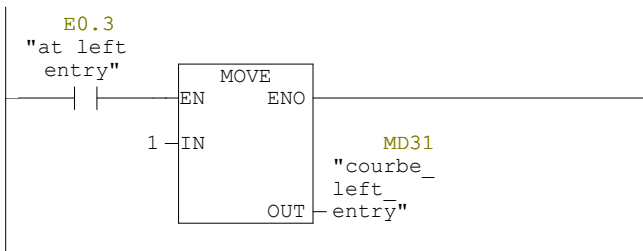
Réseau : 4



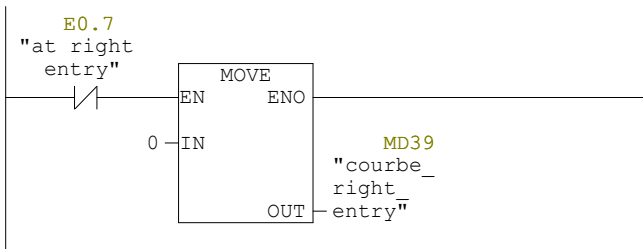
Réseau : 5



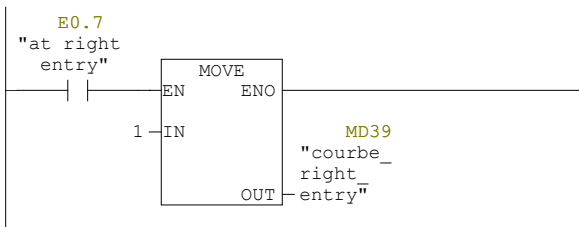
Réseau : 6



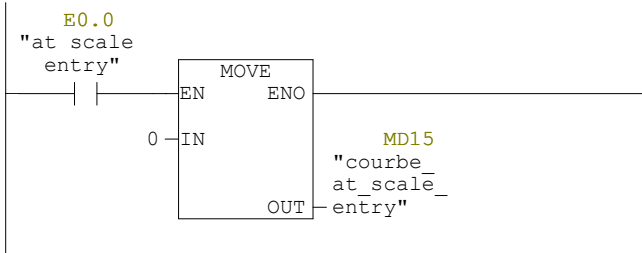
Réseau : 7



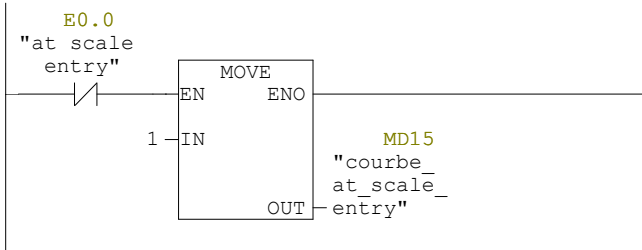
Réseau : 8



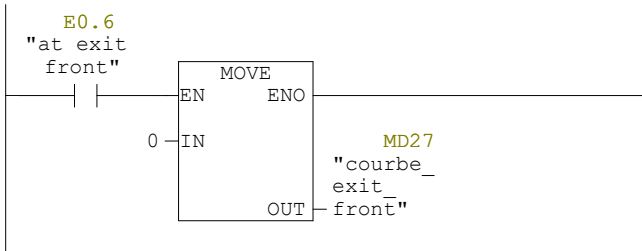
Réseau : 9



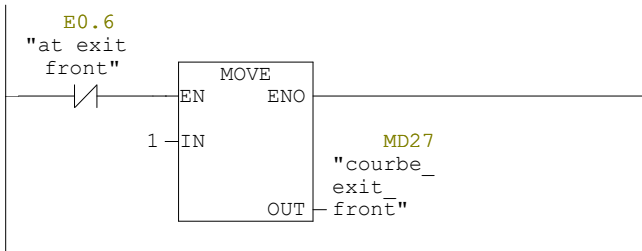
Réseau : 10



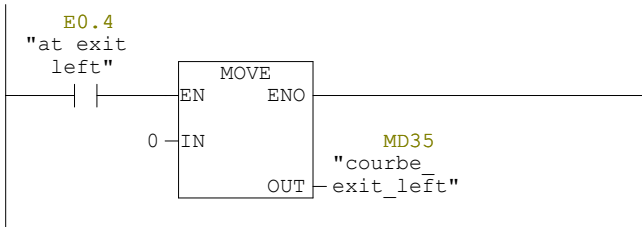
Réseau : 11



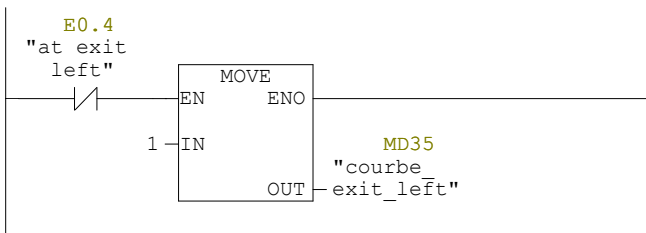
Réseau : 12



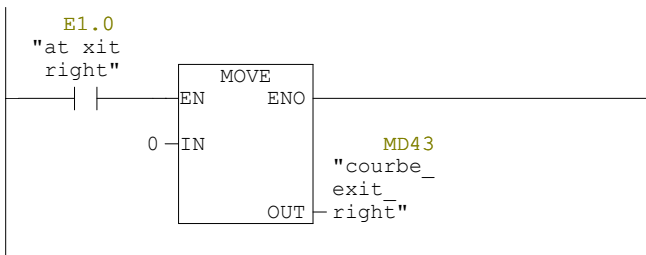
Réseau : 13



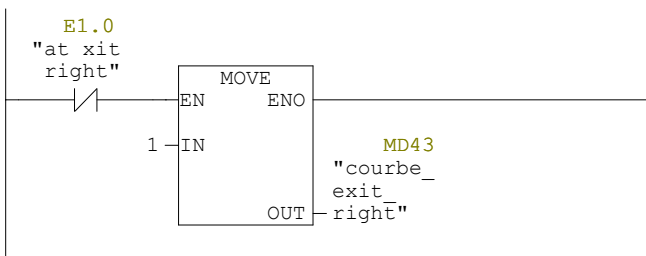
Réseau : 14



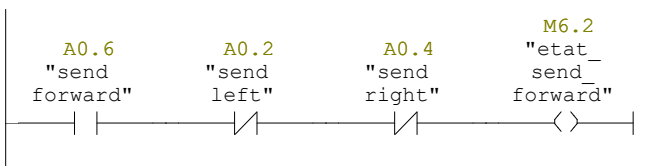
Réseau : 15



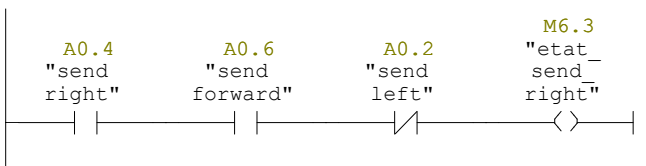
Réseau : 16



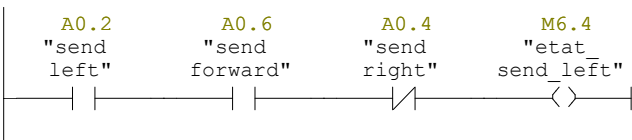
Réseau : 17



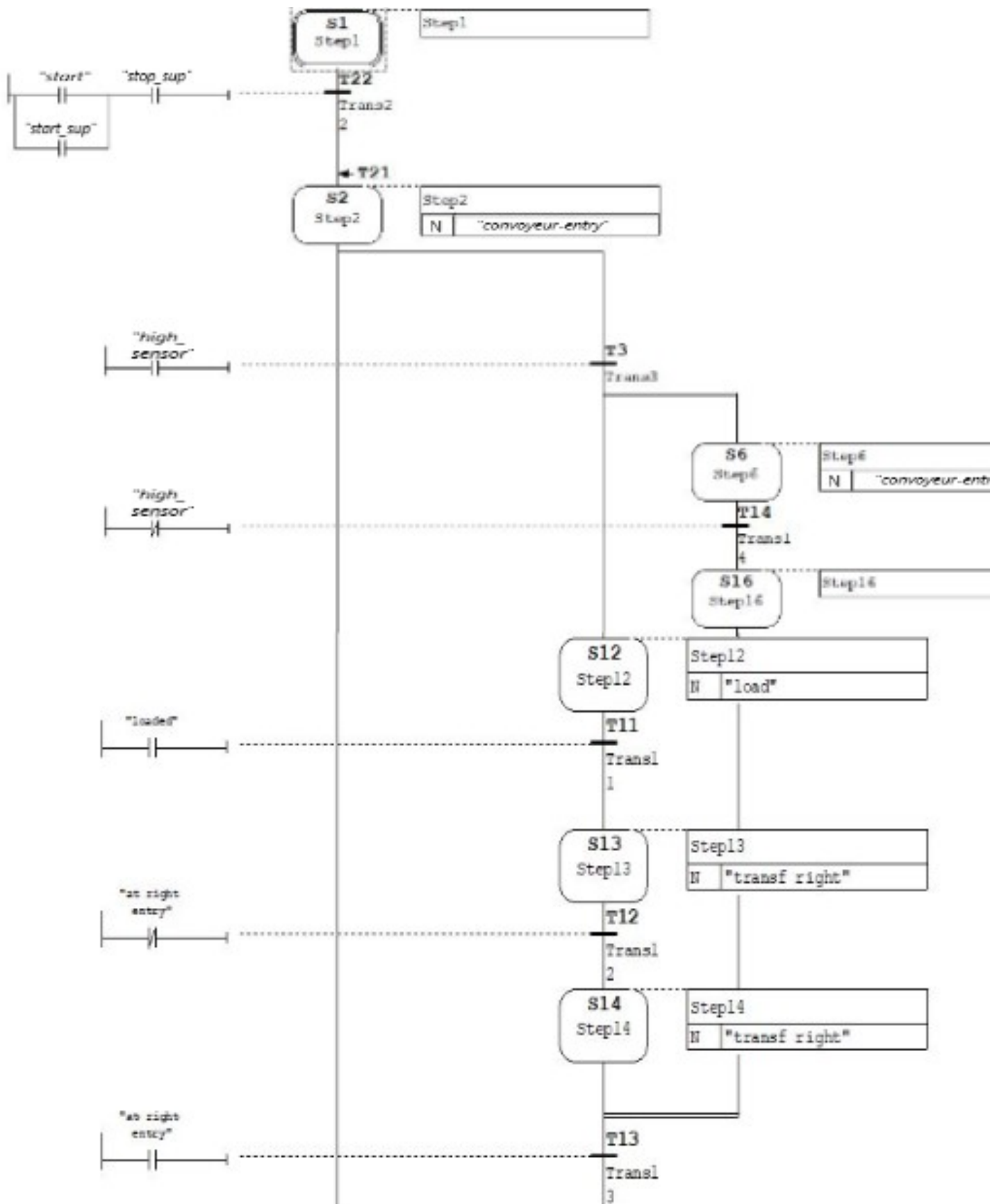
Réseau : 18

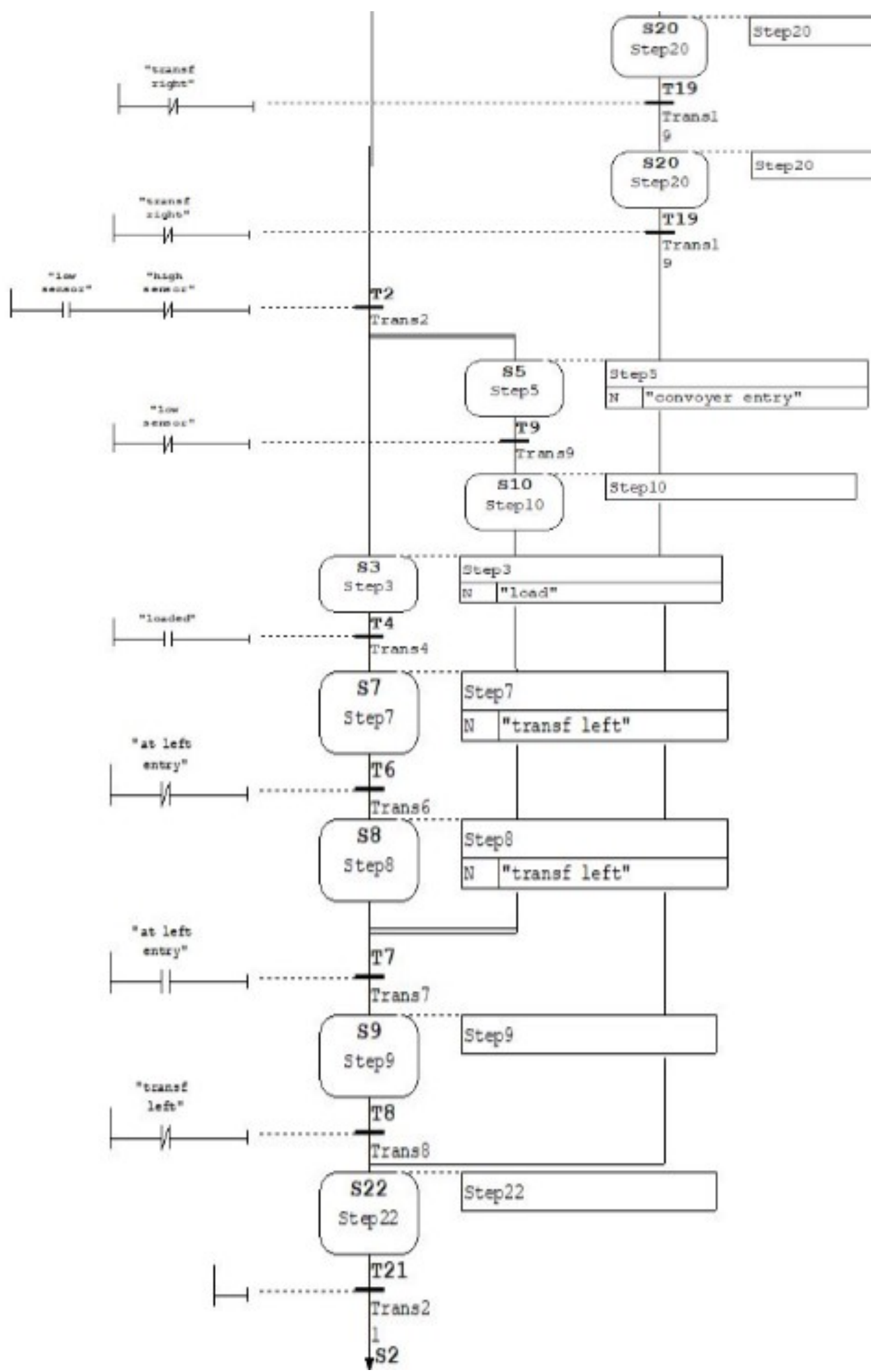


Réseau : 19



programme de la station de tri par Hauteur (station basique) :





OB1 - <hors ligne>

"Cycle Execution"

Nom :
Auteur :
Horodatage Code :
Interface :
Longueur (bloc/code /données locales) : 00814 00614 00028

Famille :
Version : 0.1
Version de bloc : 2
 01/06/2024 11:35:29
 15/02/1996 16:51:12

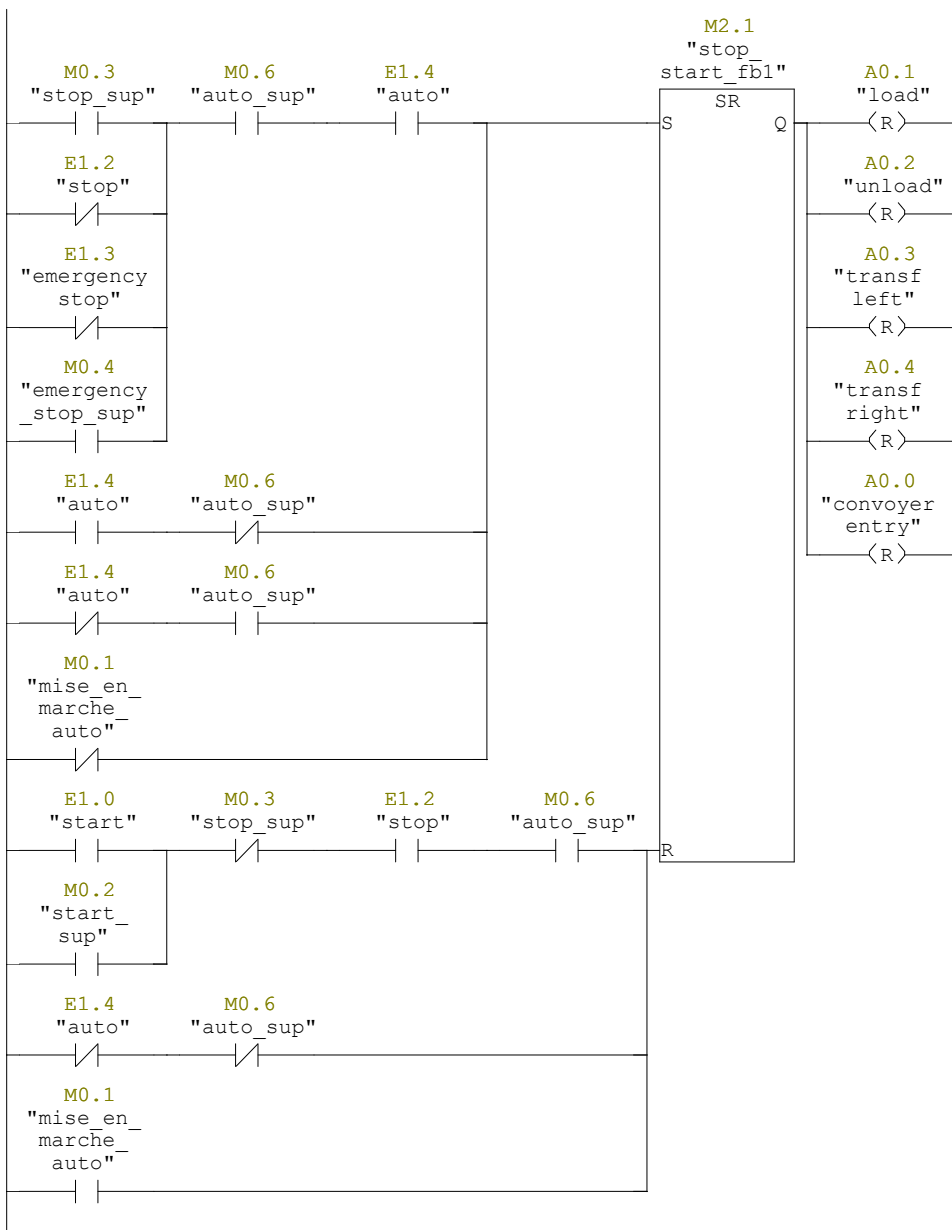
Nom	Type de données	Adresse	Commentaire
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

Bloc : OB1 "Main Program Sweep (Cycle)"

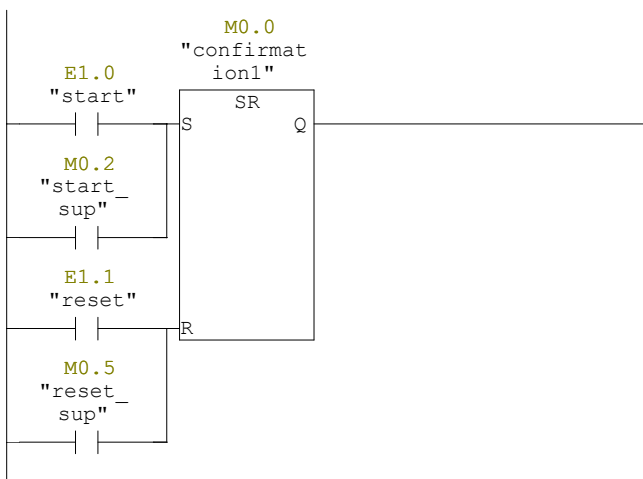
Réseau : 1



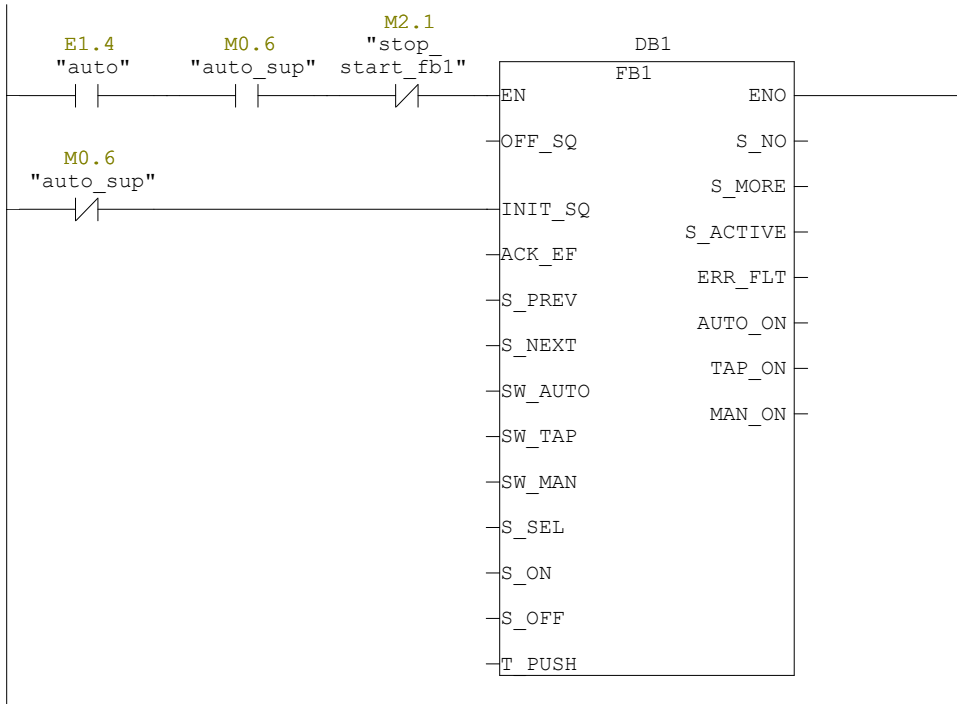
Réseau : 2



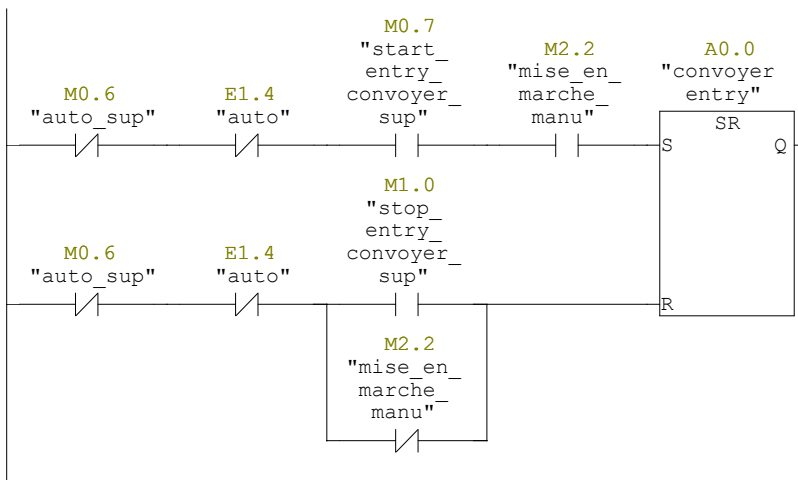
Réseau : 3



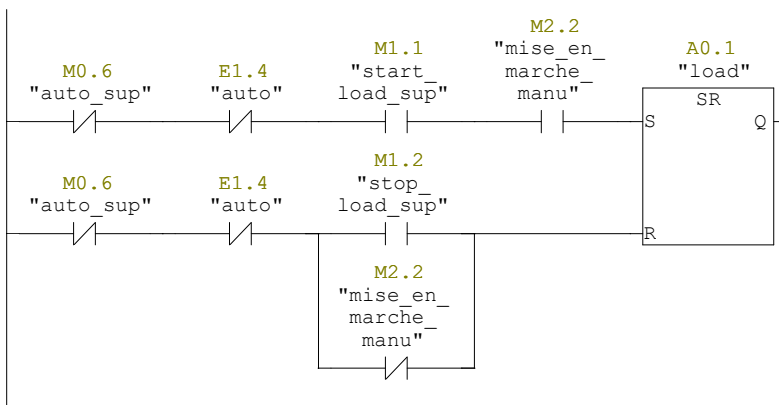
Réseau : 4



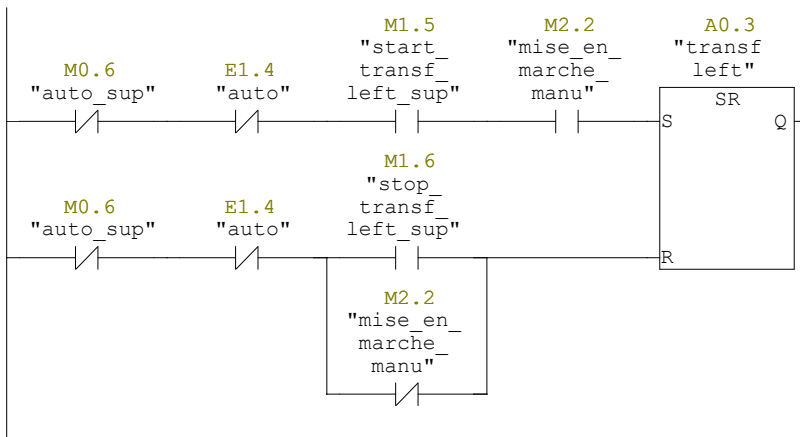
Réseau : 5



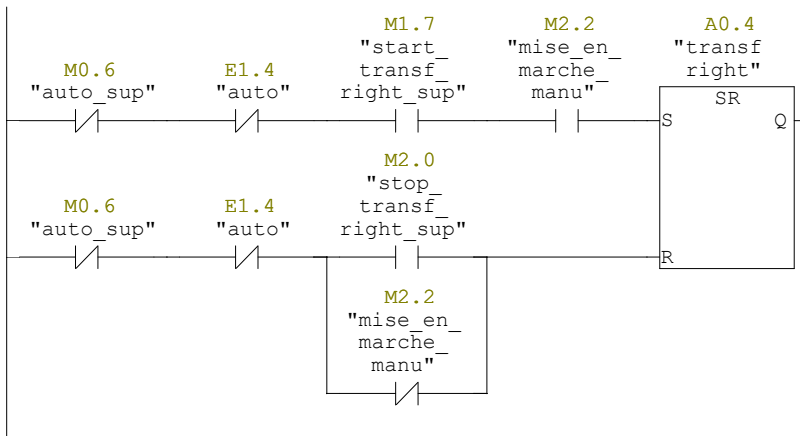
Réseau : 6



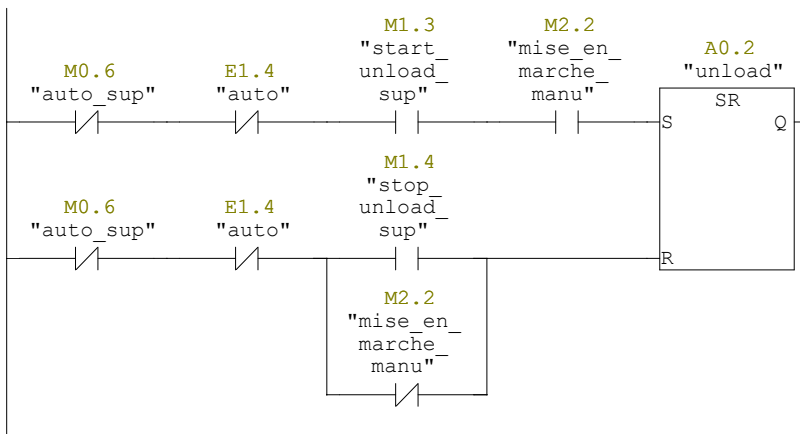
Réseau : 7



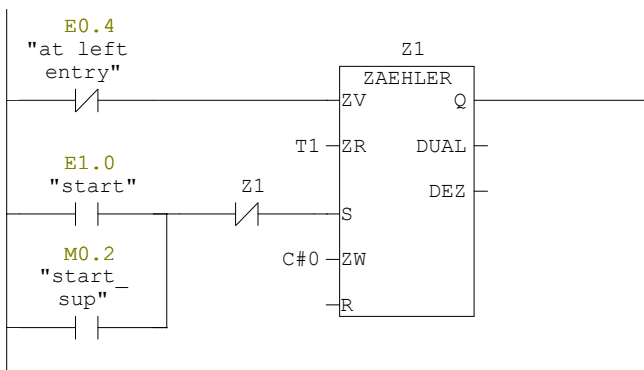
Réseau : 8



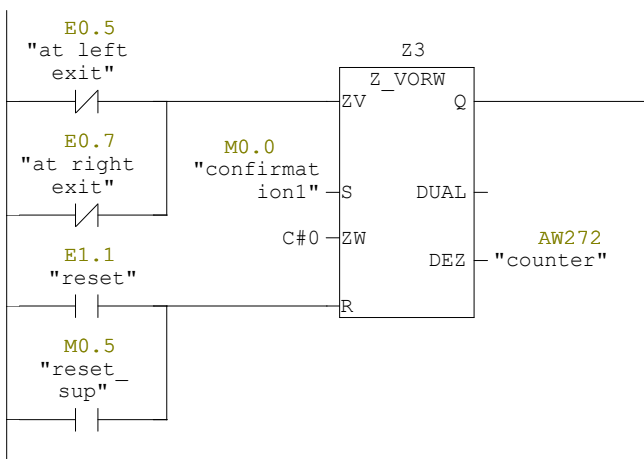
Réseau : 9



Réseau : 10



Réseau : 11



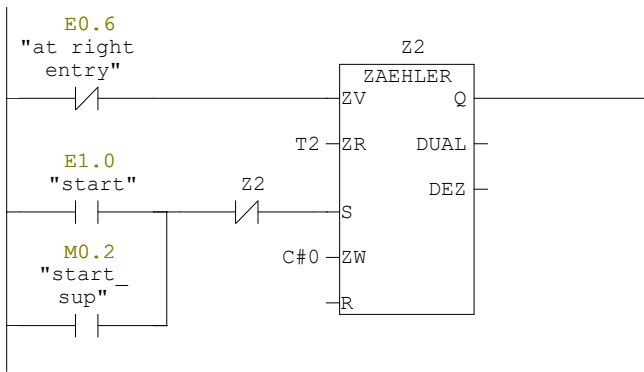
Réseau : 12



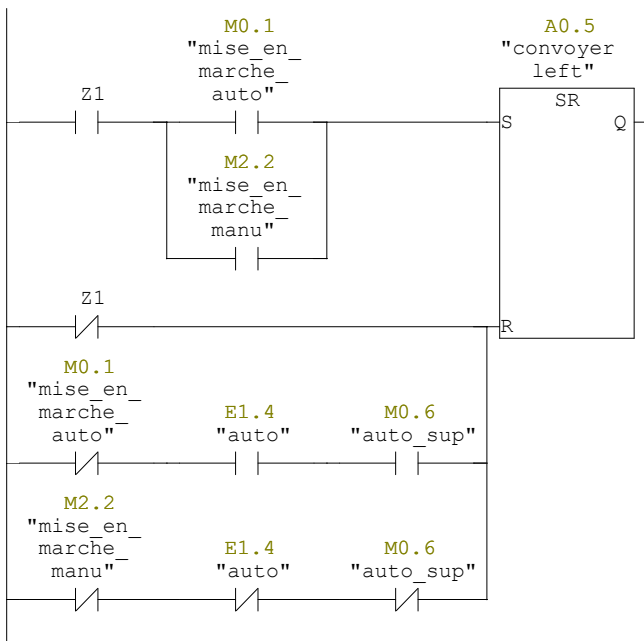
Réseau : 13



Réseau : 14



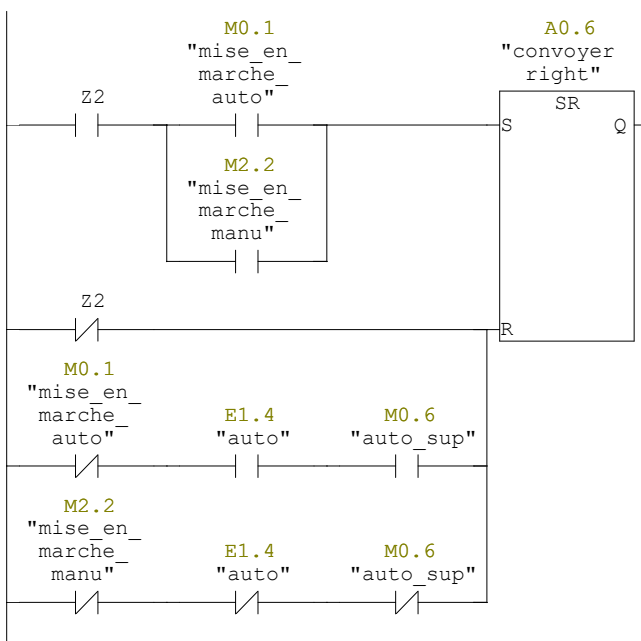
Réseau : 15



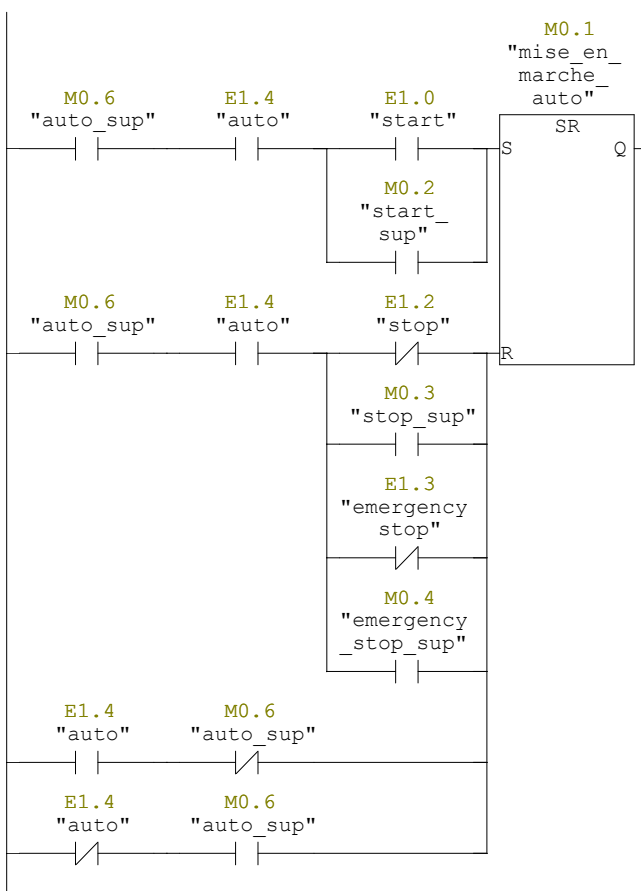
Réseau : 16



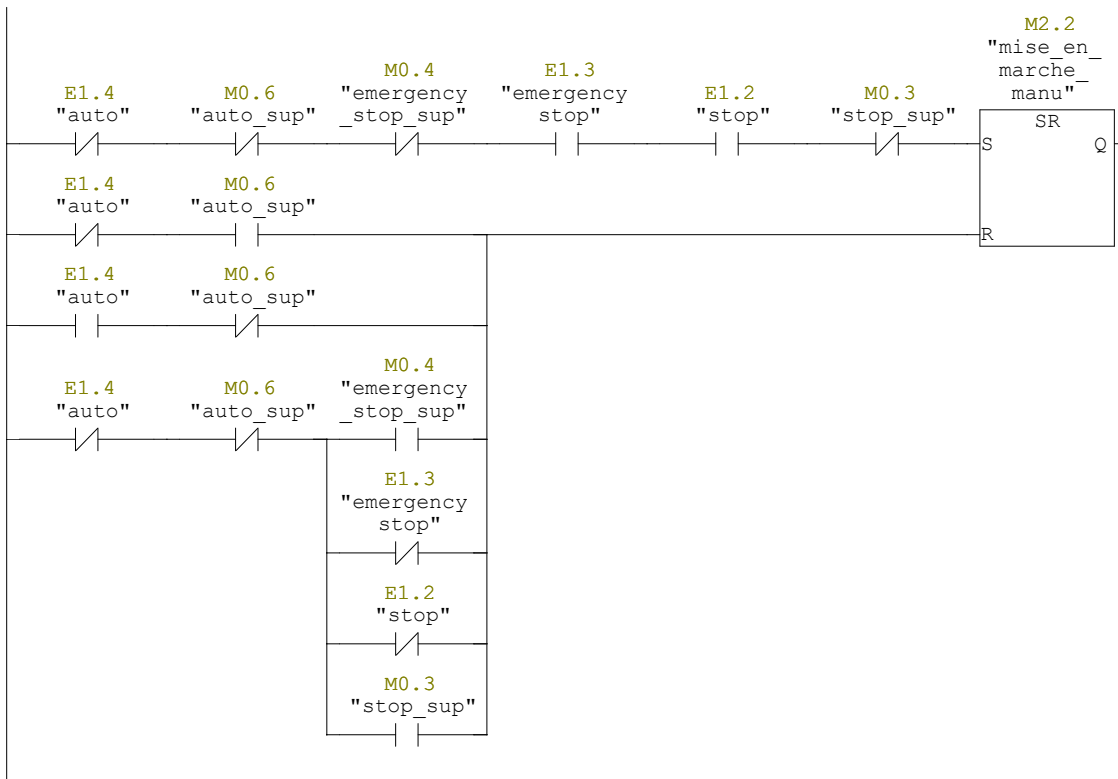
Réseau : 17



Réseau : 18



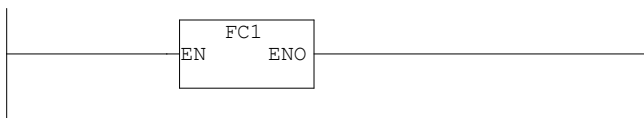
Réseau : 19



Réseau : 20



Réseau : 21



FC1 - <hors ligne>

""

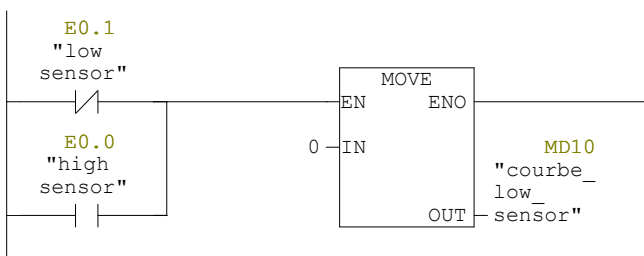
Nom :
Auteur :
Horodatage Code :
Interface :
Longueur (bloc/code /données locales) :

Famille :
Version : 0.1
Version de bloc : 2
 01/06/2024 11:35:15
 01/06/2024 11:34:06
 00316 00178 00000

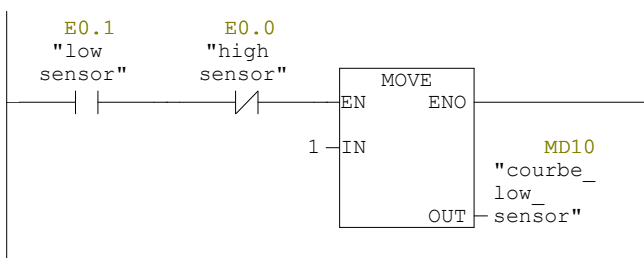
Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC1

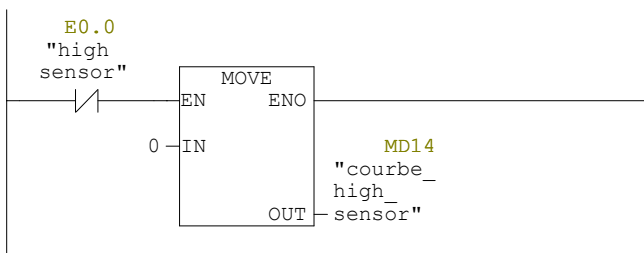
Réseau : 1



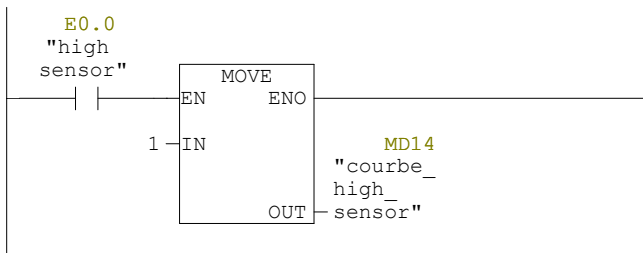
Réseau : 2



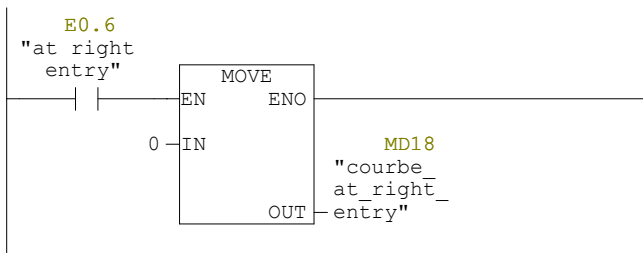
Réseau : 3



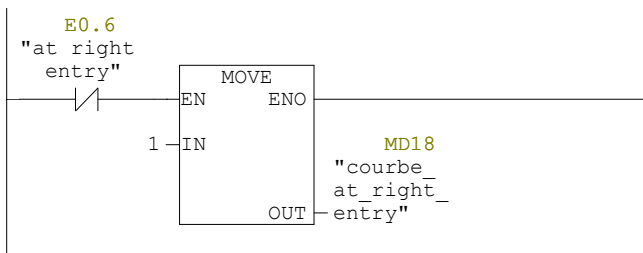
Réseau : 4



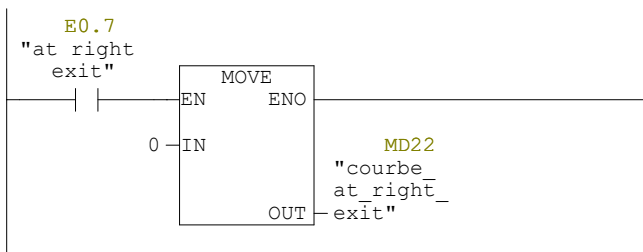
Réseau : 5



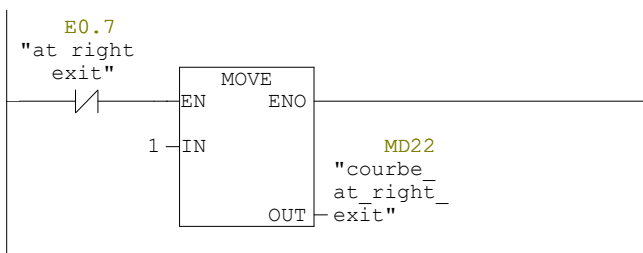
Réseau : 6



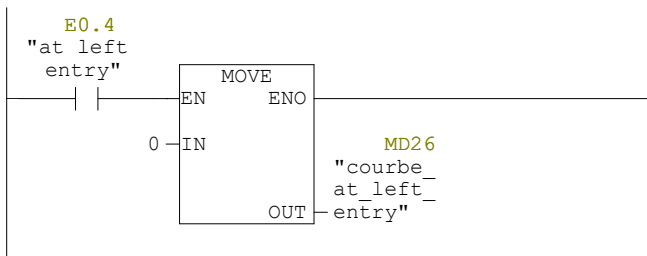
Réseau : 7



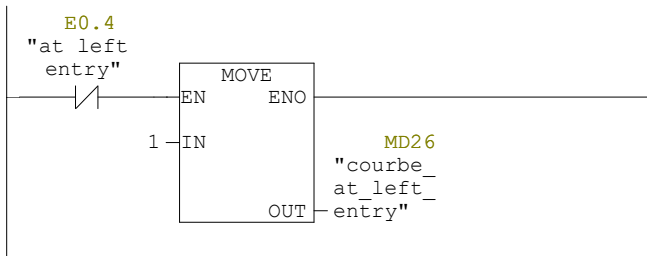
Réseau : 8



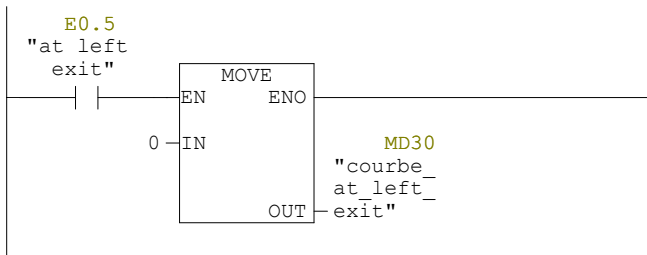
Réseau : 9



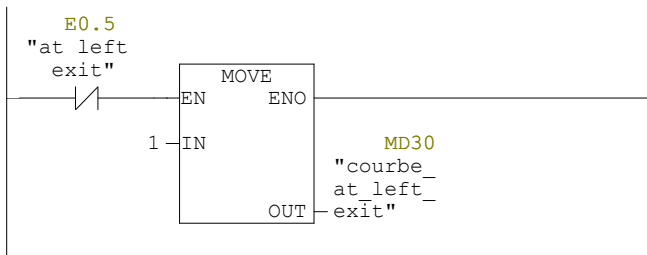
Réseau : 10



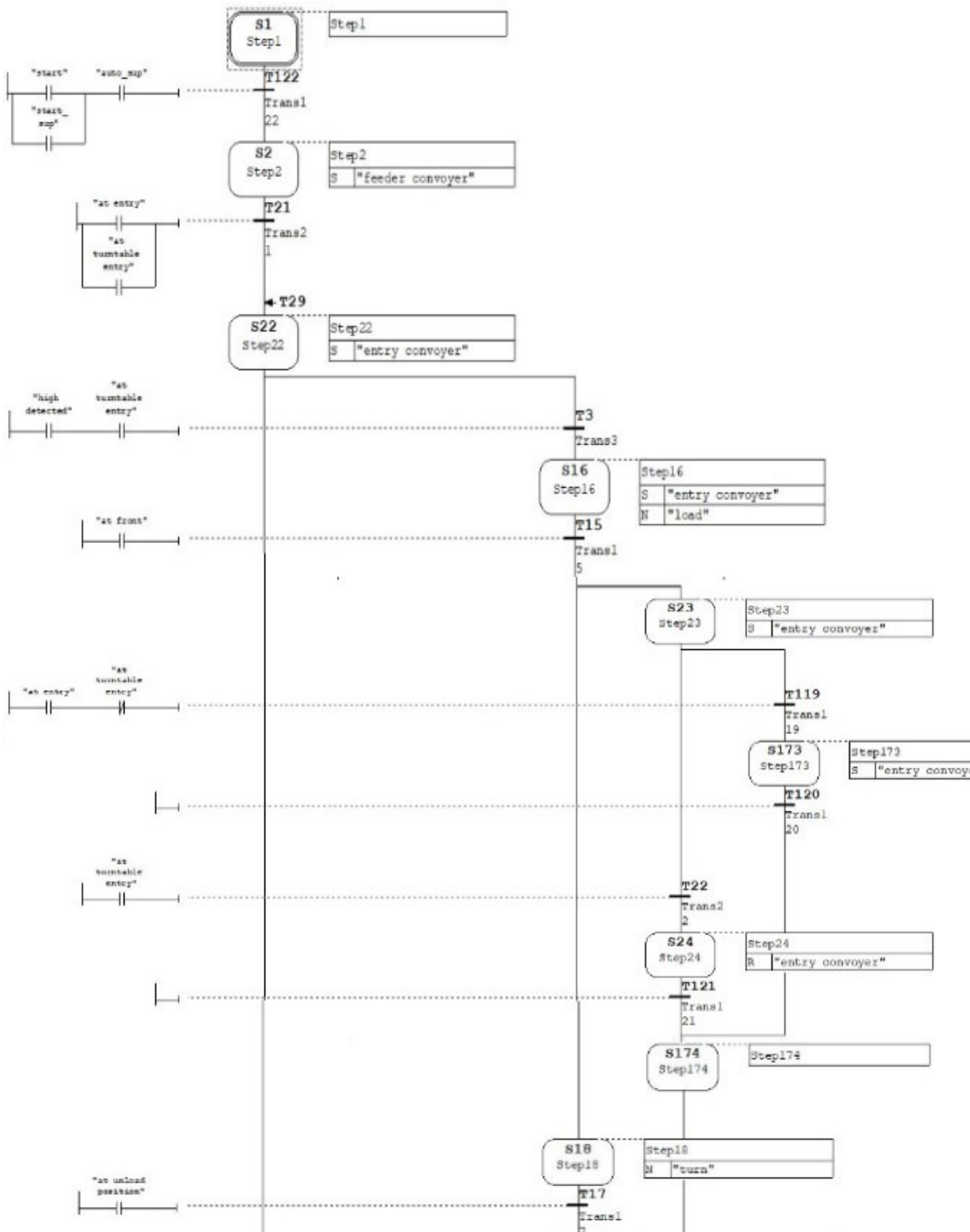
Réseau : 11

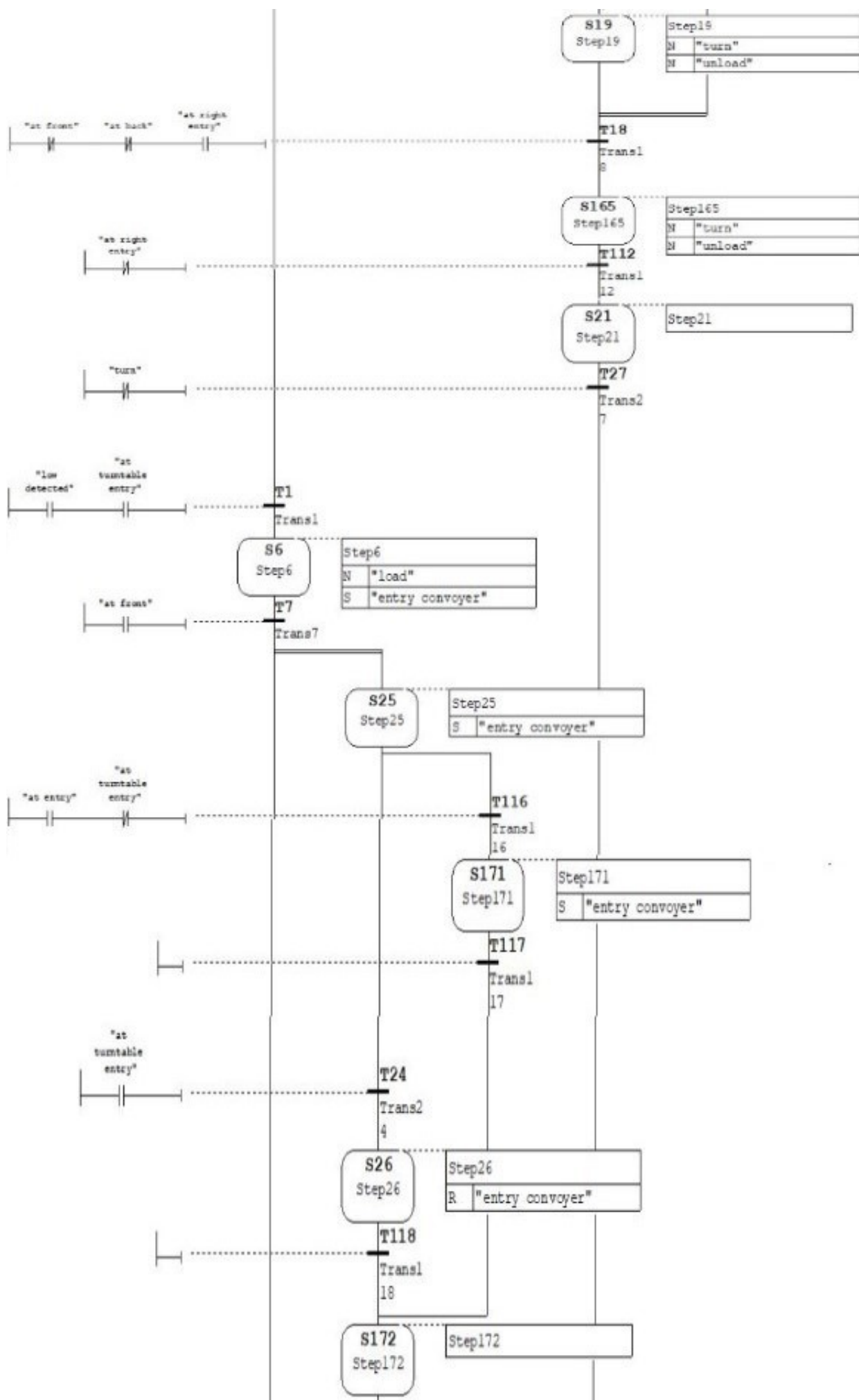


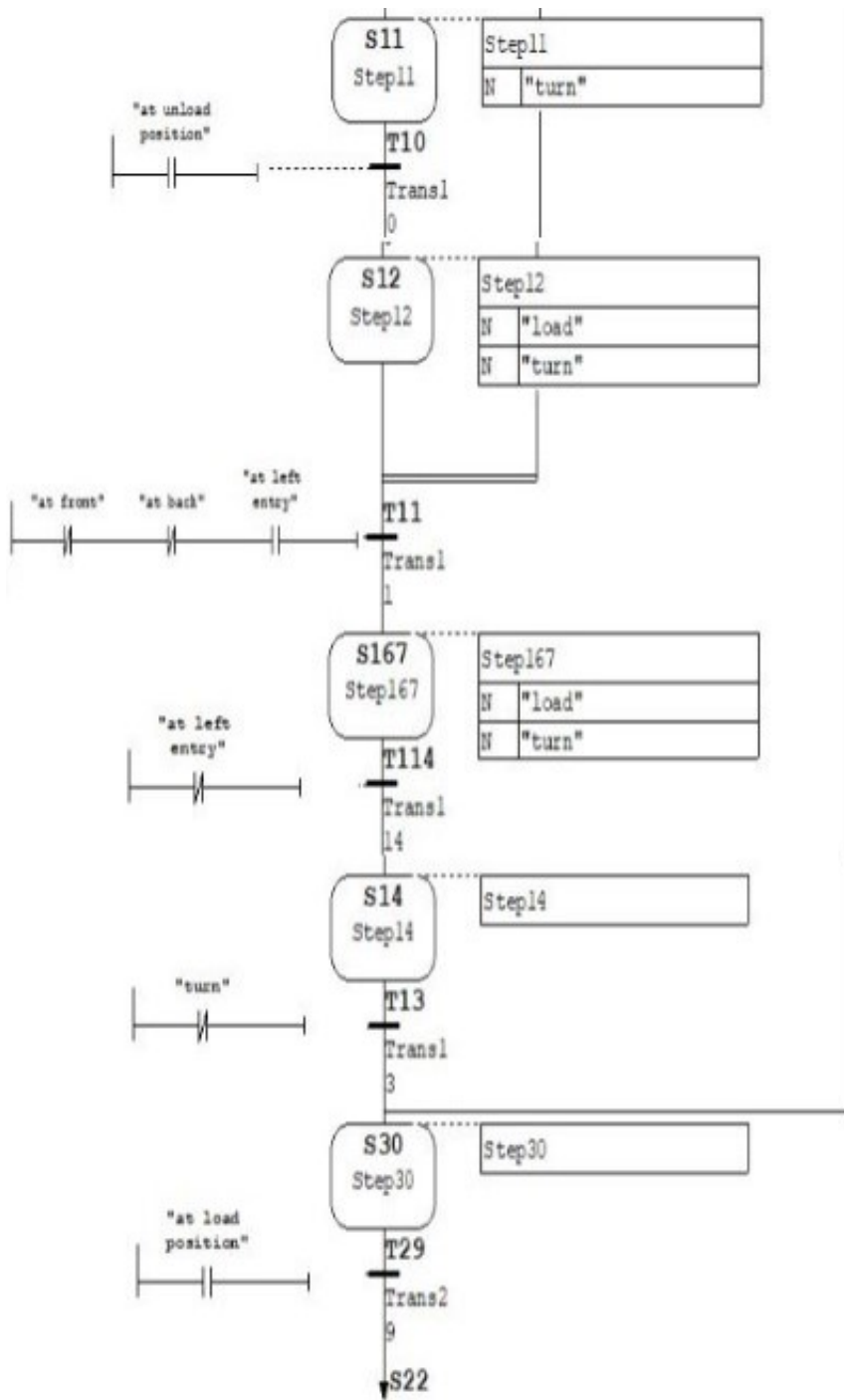
Réseau : 12

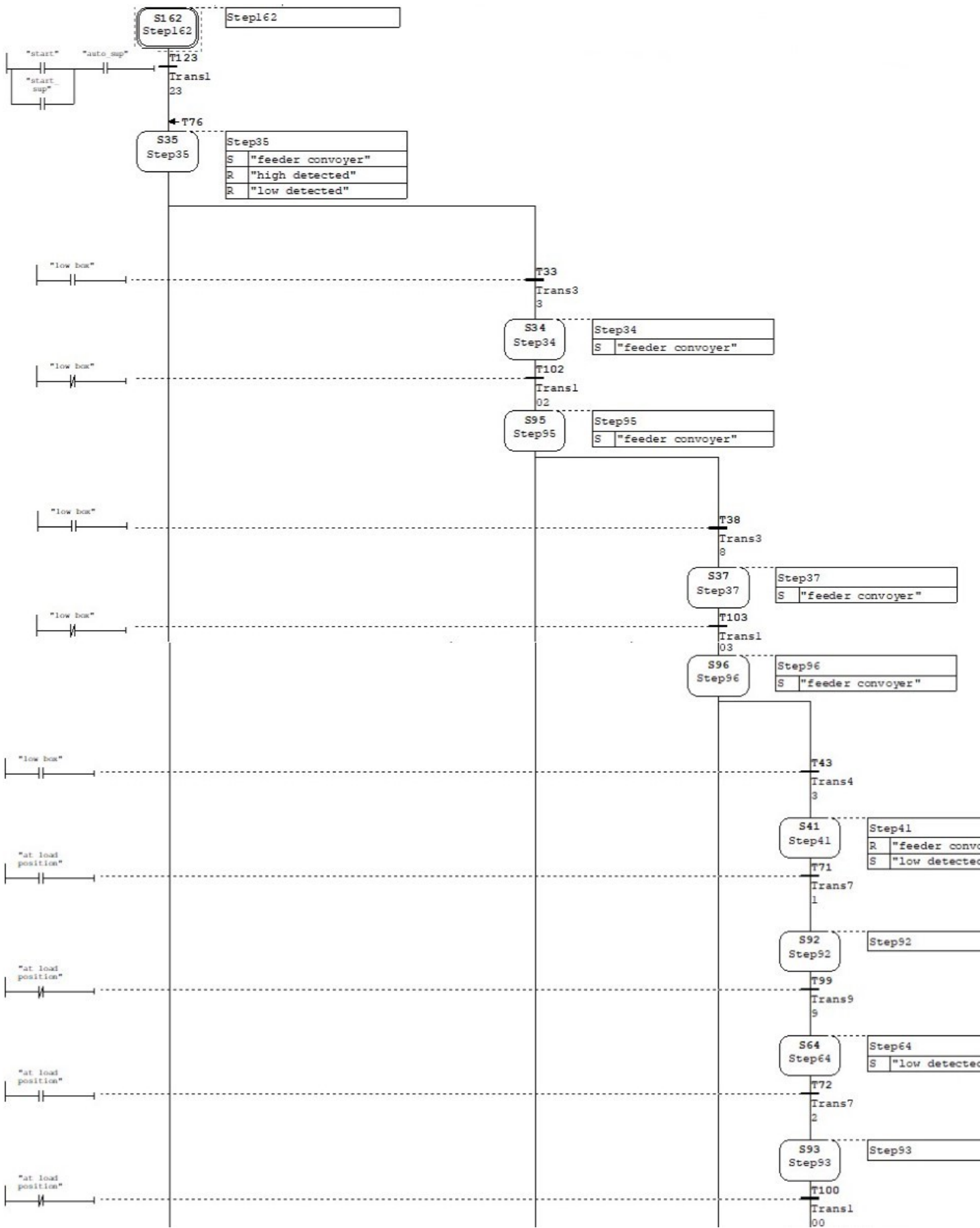


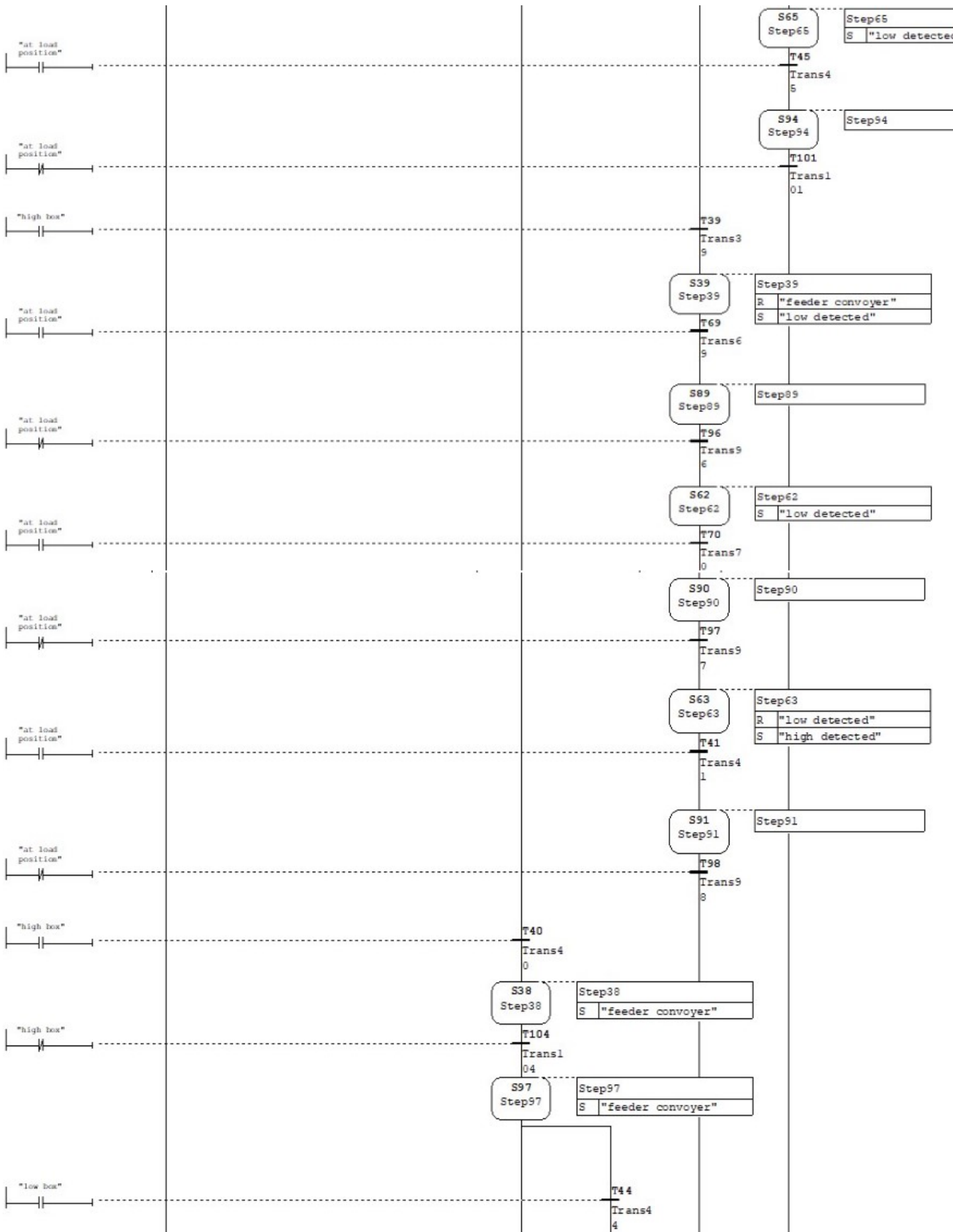
programme de la station de tri par Hauteur (station avancée) :

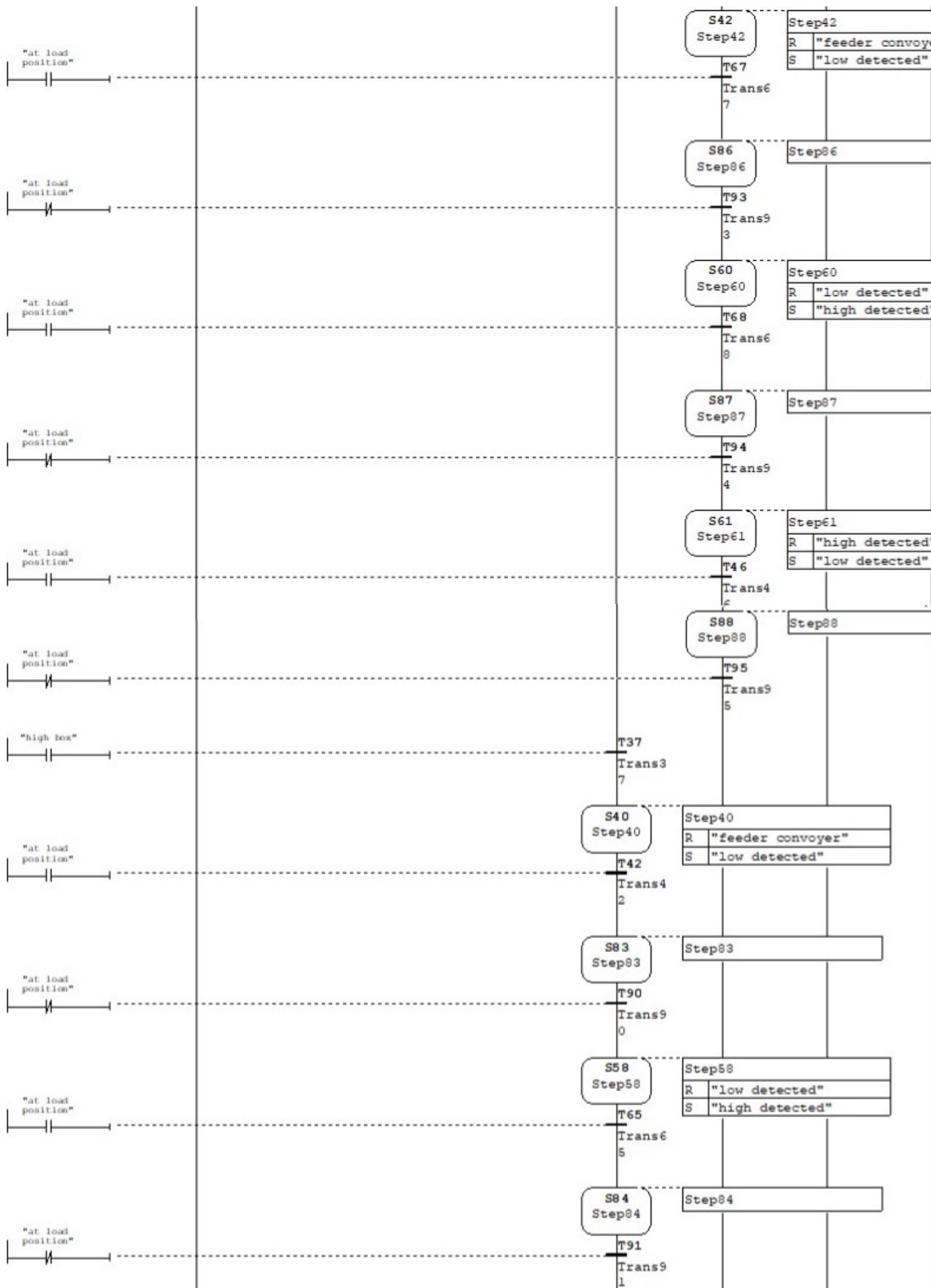


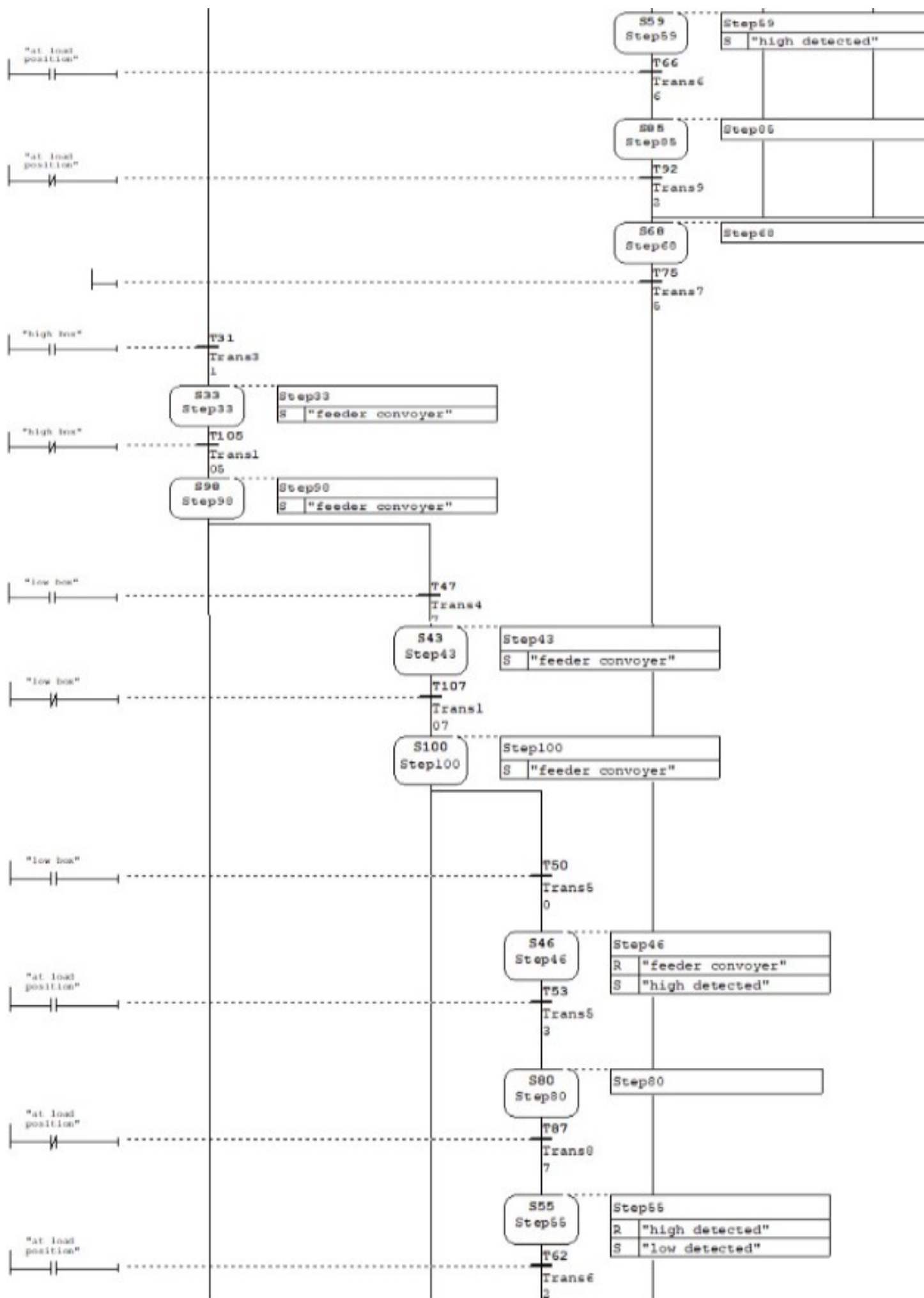


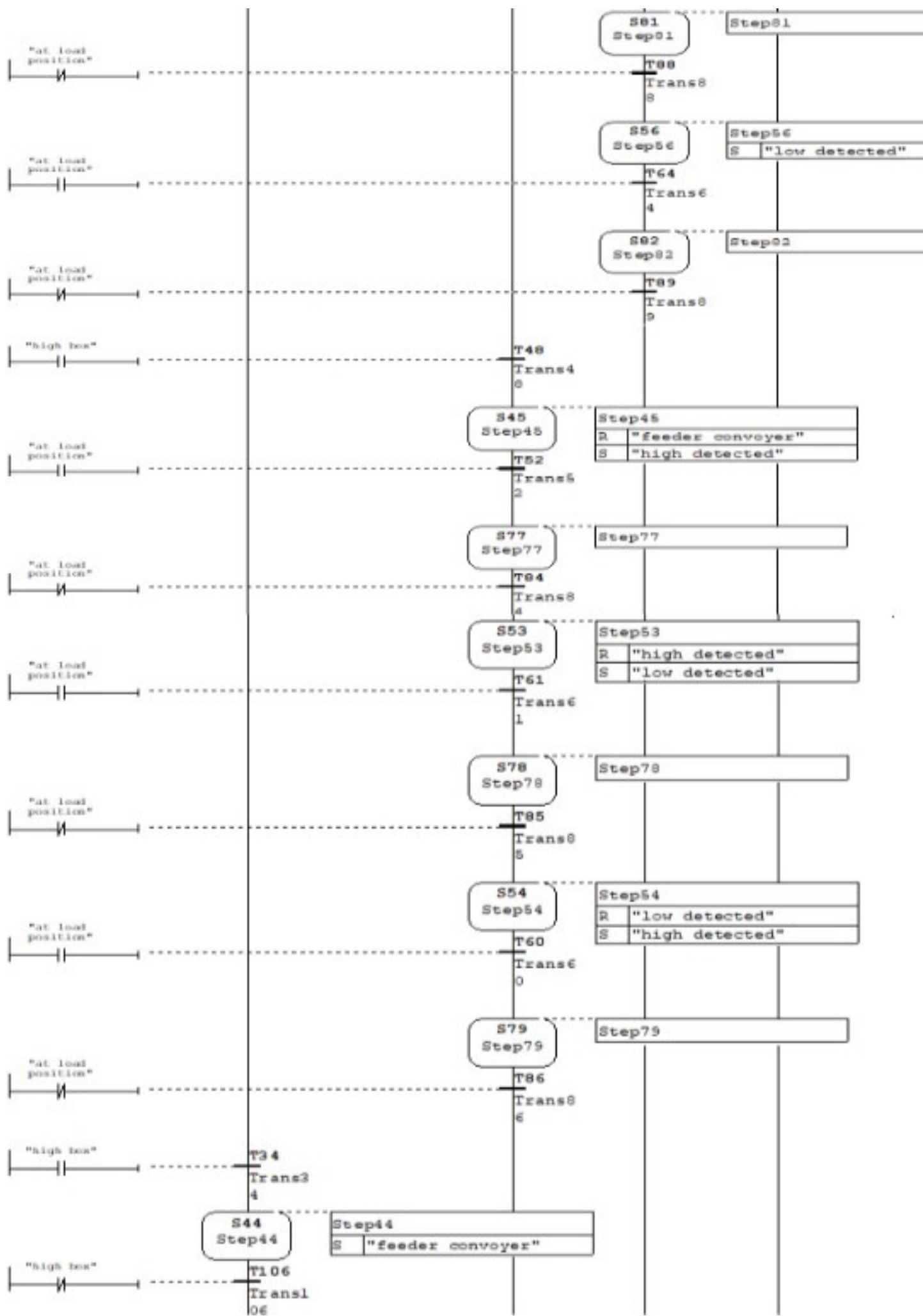


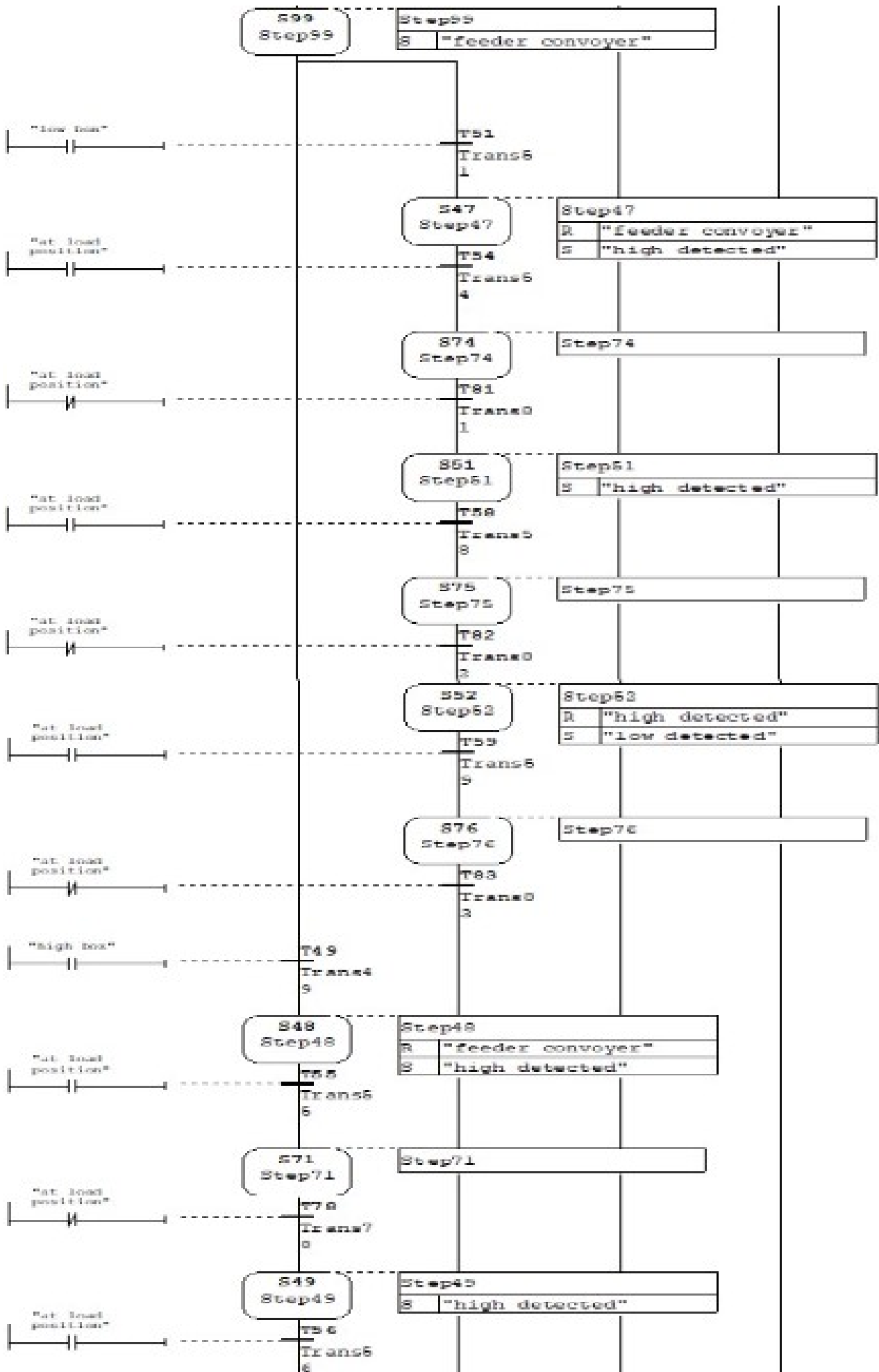


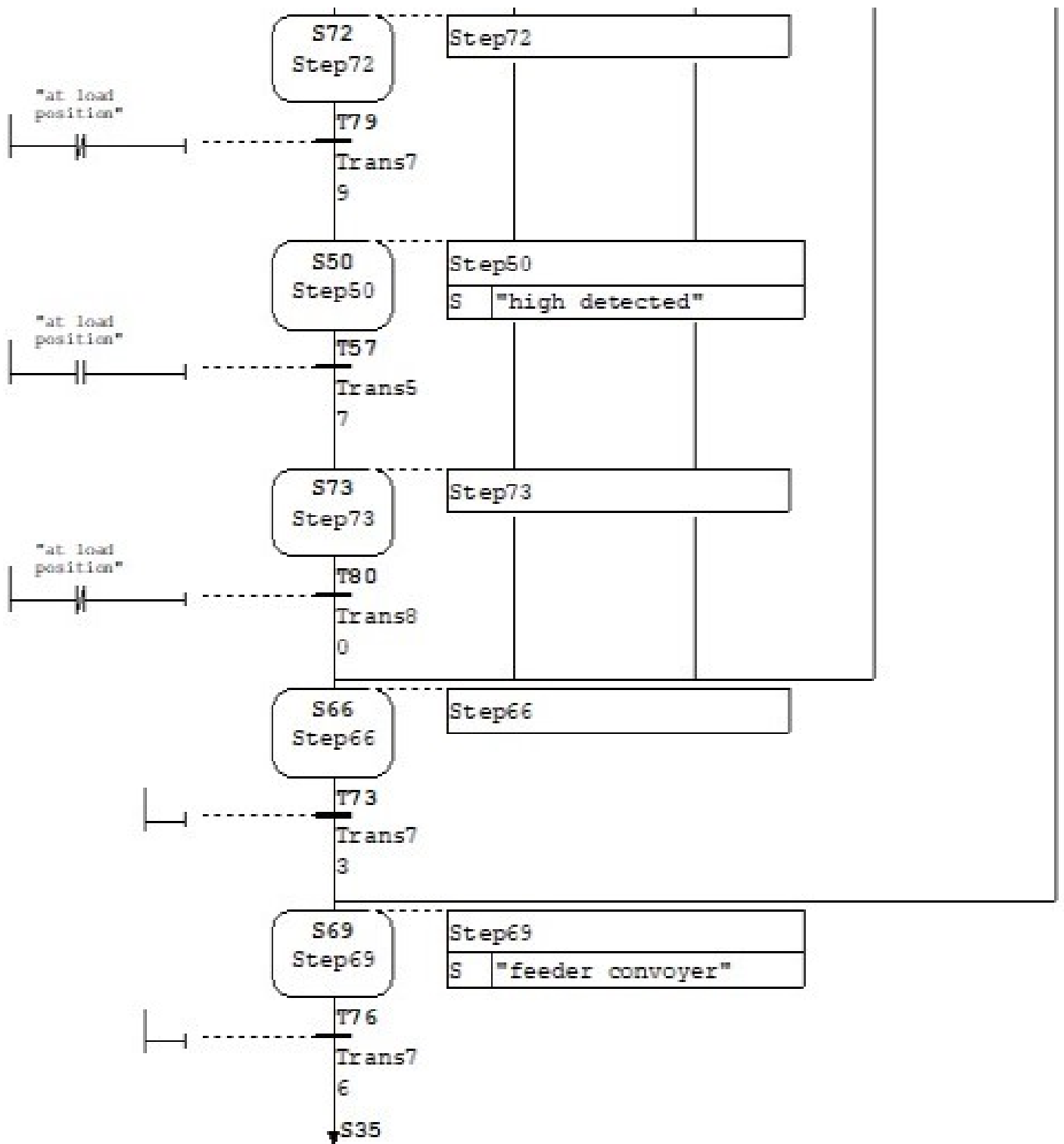












OB1 - <hors ligne>

"Cycle Execution"

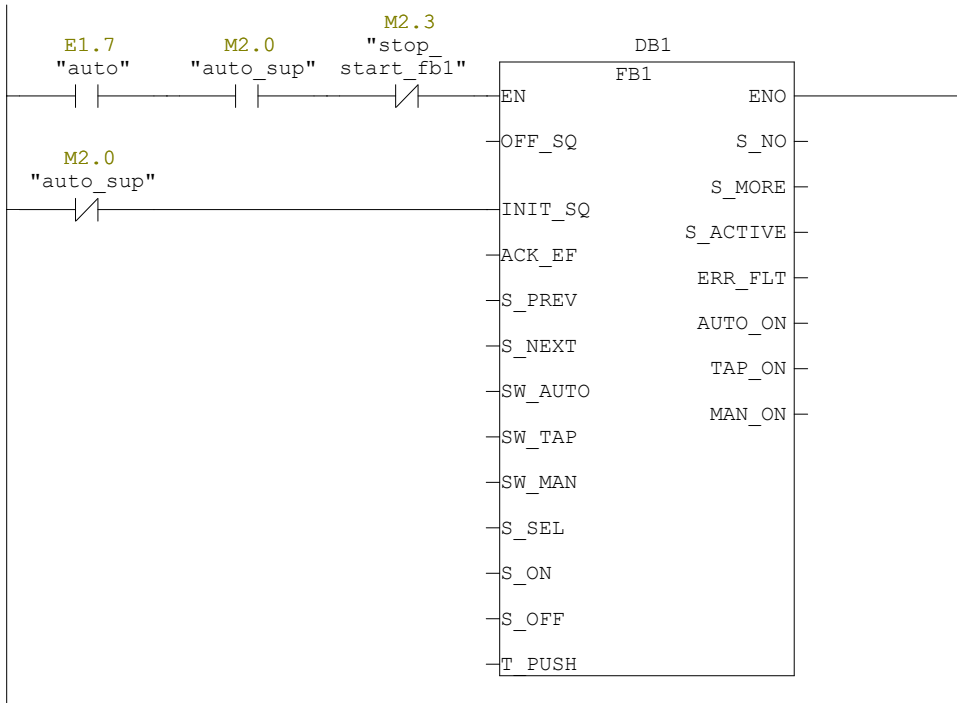
Nom :
Auteur :
Horodatage Code :
Interface :
Longueur (bloc/code /données locales) : 00826 00614 00028

Famille :
Version : 0.1
Version de bloc : 2
 01/06/2024 11:31:18
 15/02/1996 16:51:12

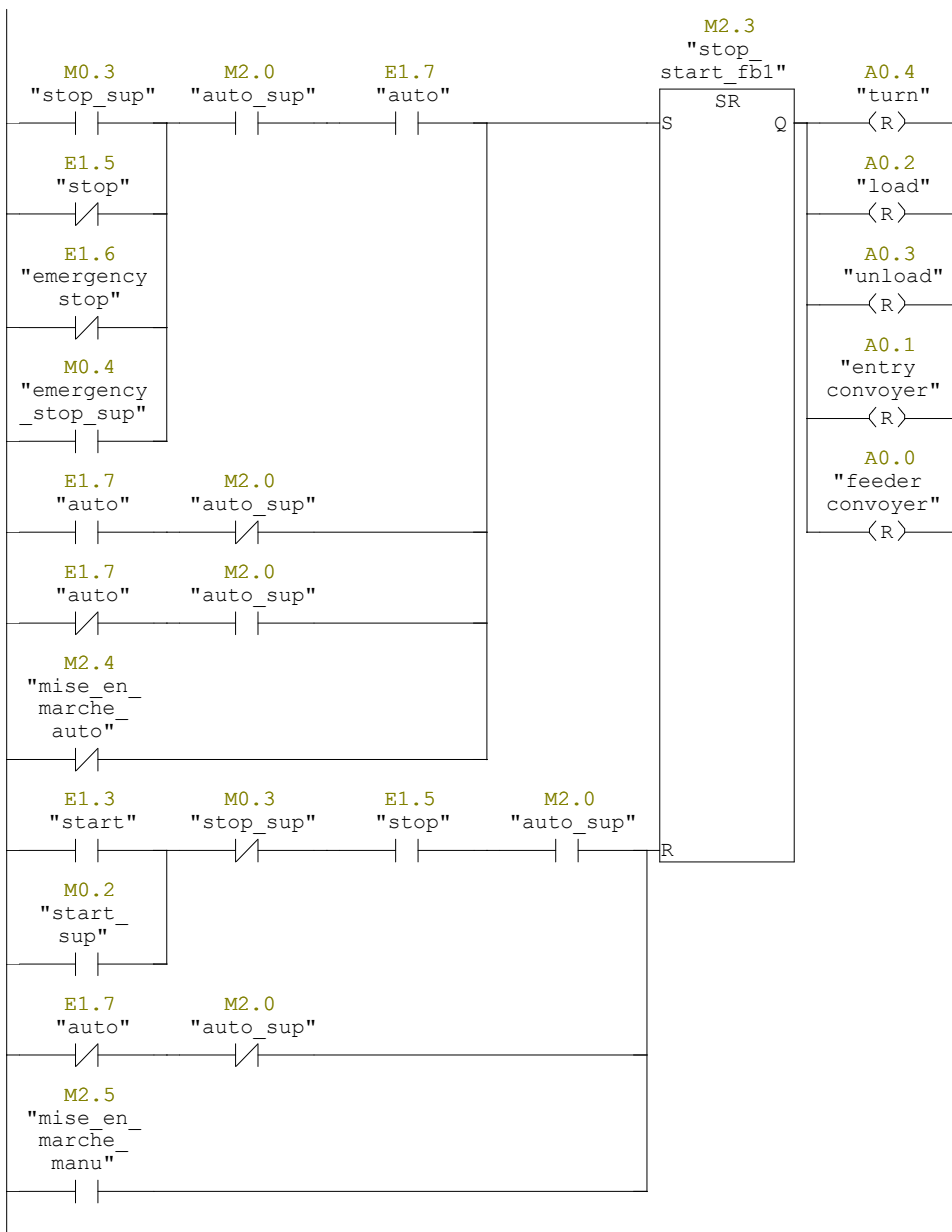
Nom	Type de données	Adresse	Commentaire
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

Bloc : OB1 "Main Program Sweep (Cycle)"

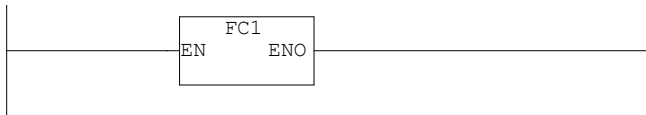
Réseau : 1



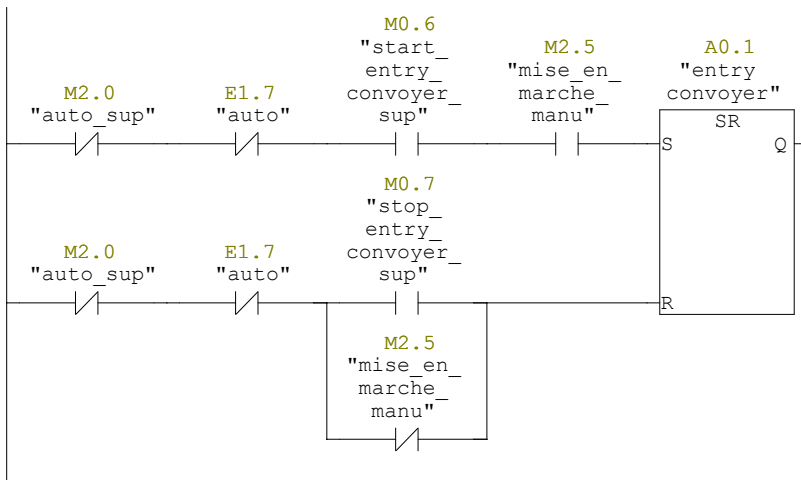
Réseau : 2



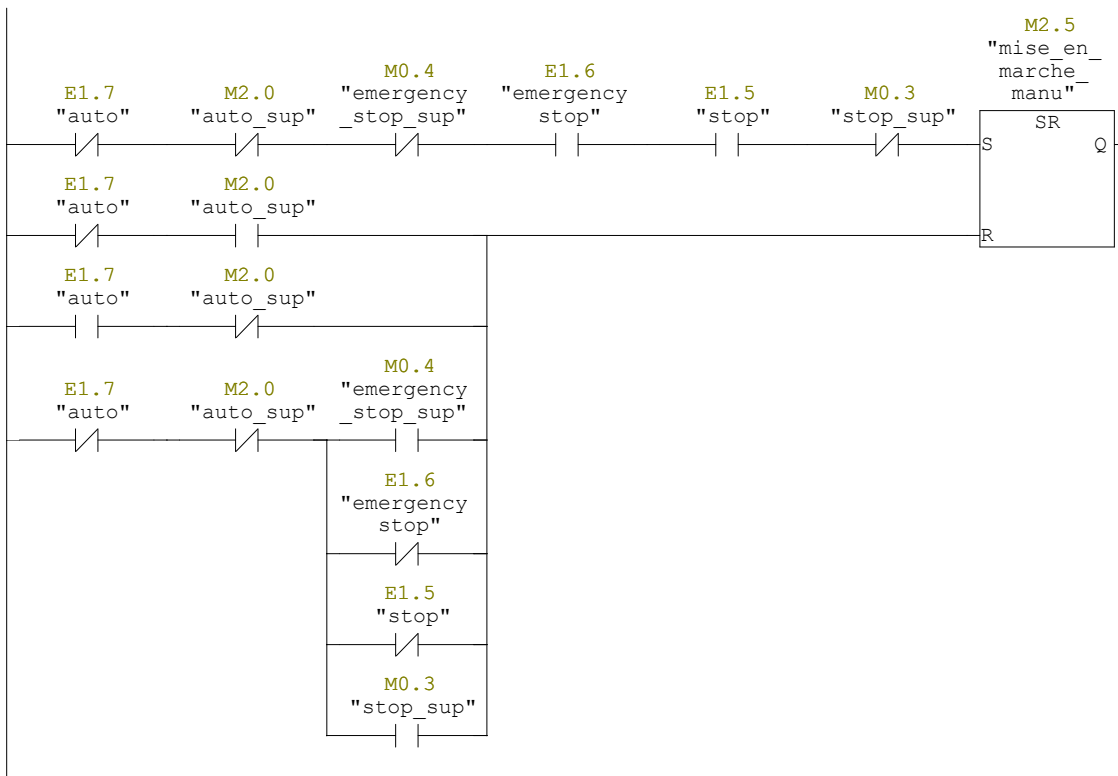
Réseau : 3



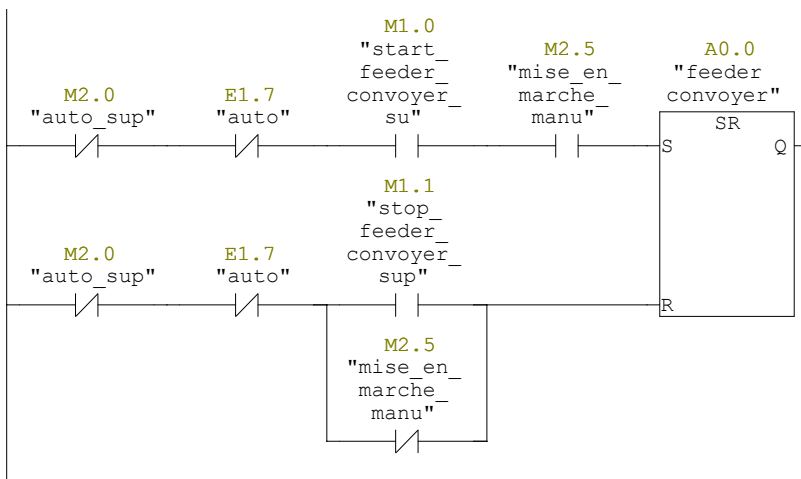
Réseau : 4



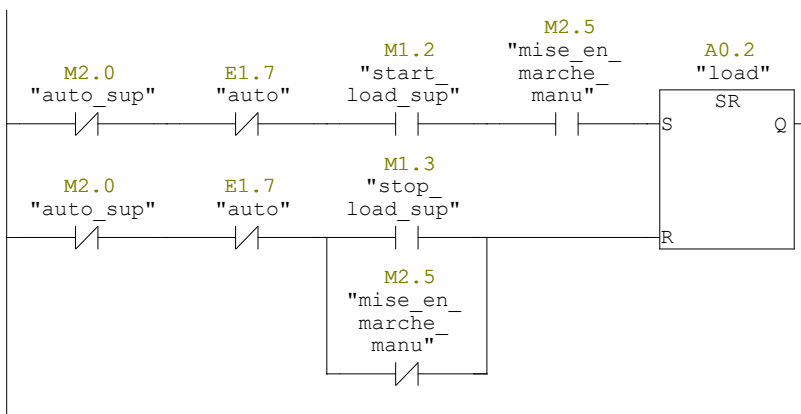
Réseau : 5



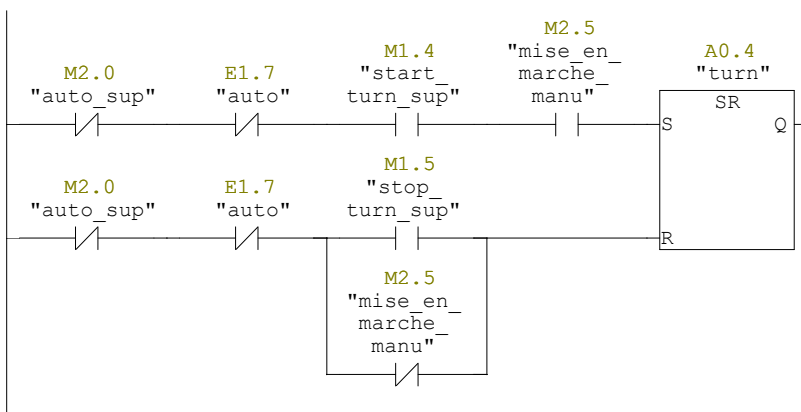
Réseau : 6



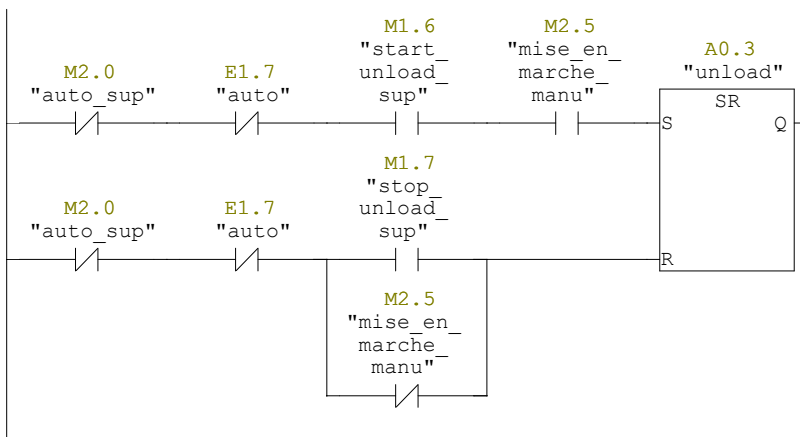
Réseau : 7



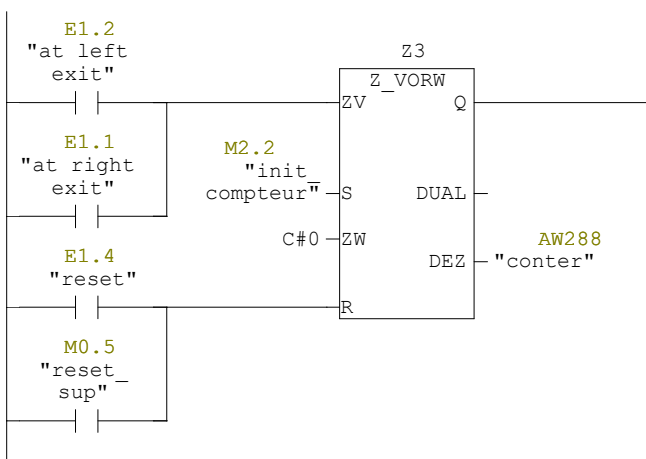
Réseau : 8



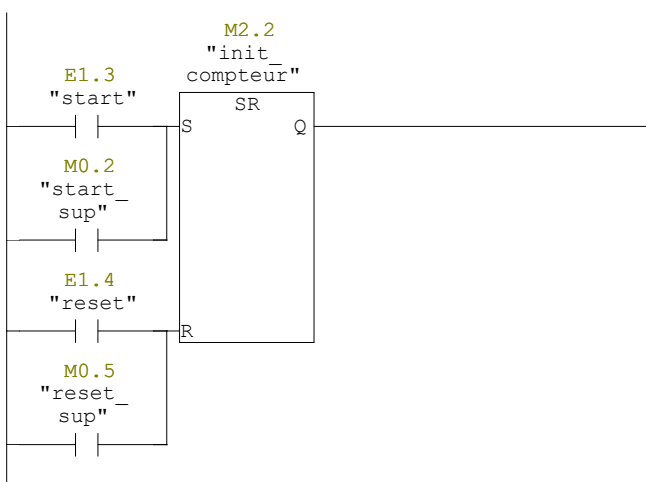
Réseau : 9



Réseau : 10



Réseau : 11



Réseau : 12



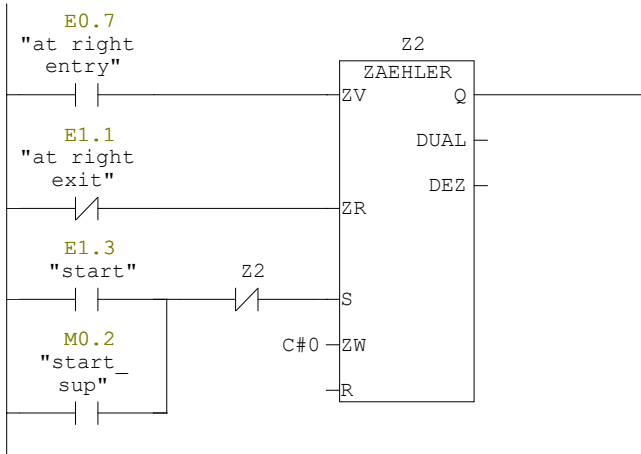
Réseau : 13



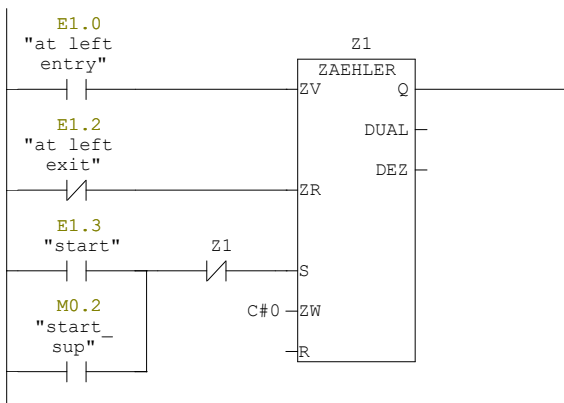
Réseau : 14



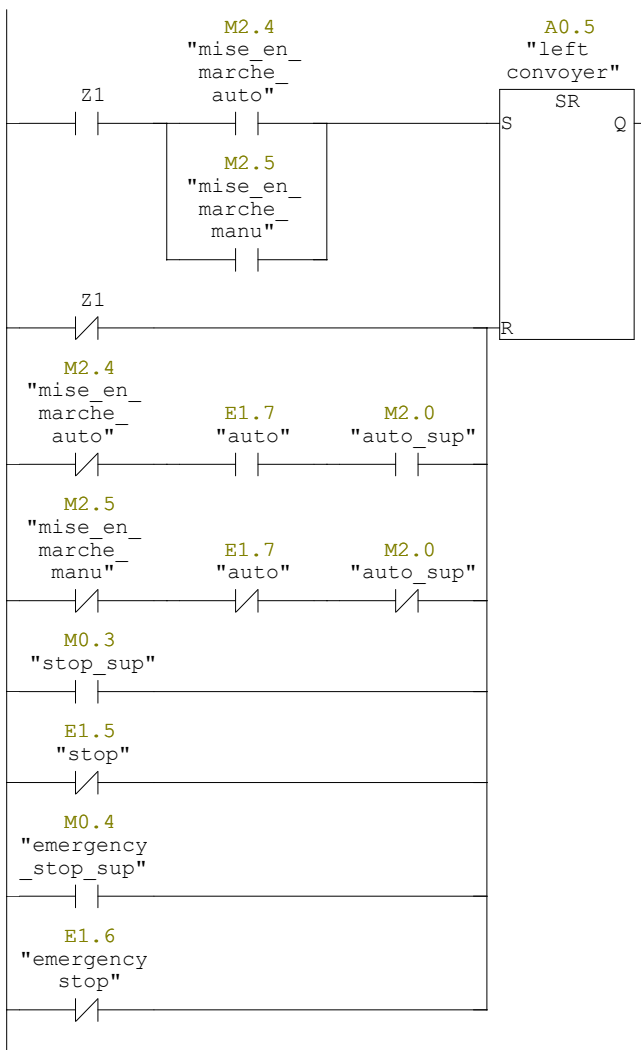
Réseau : 15



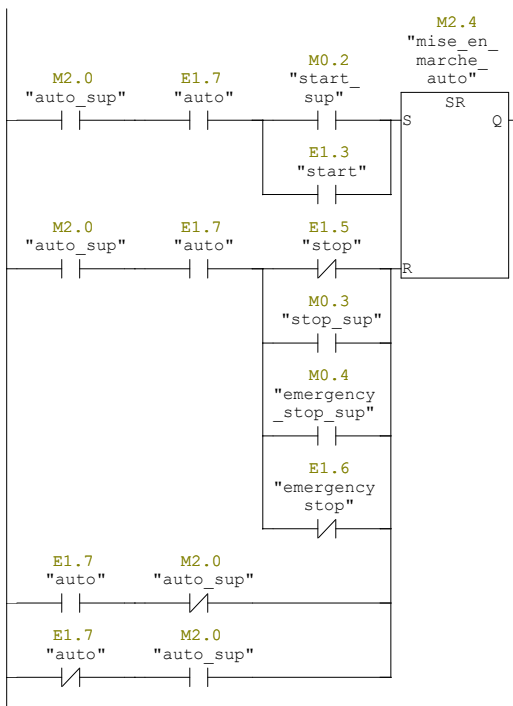
Réseau : 16



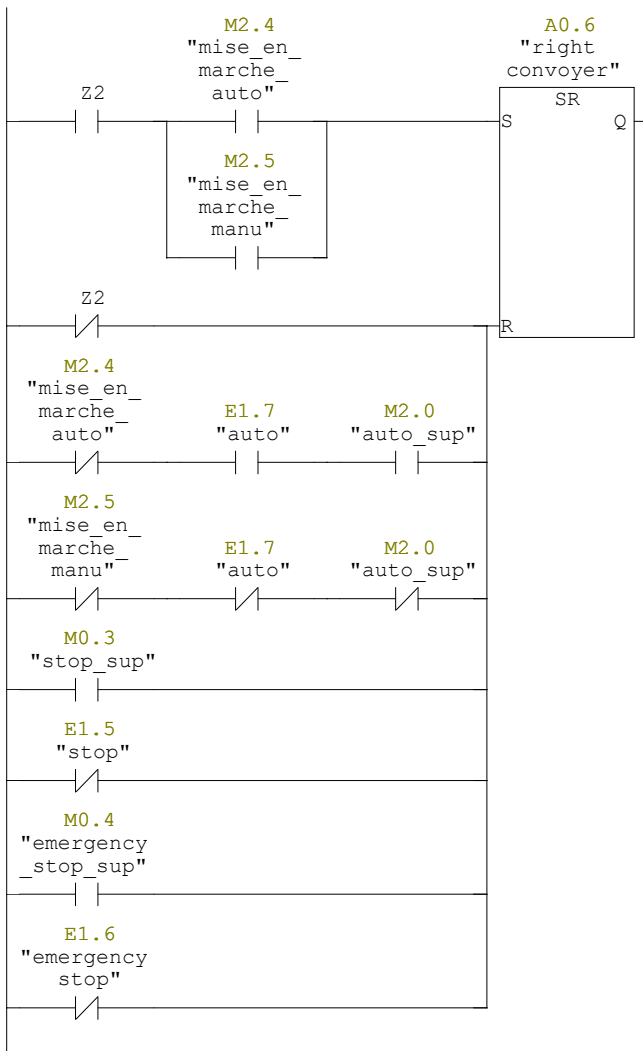
Réseau : 17



Réseau : 18



Réseau : 19



FC1 - <hors ligne>

""

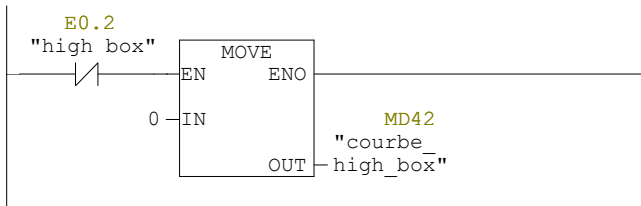
Nom :
Auteur :
Horodatage Code :
Interface :
Longueur (bloc/code /données locales) : 00398 00240 00000

Famille :
Version : 0.1
Version de bloc : 2
 05/06/2024 20:25:17
 01/06/2024 11:29:28

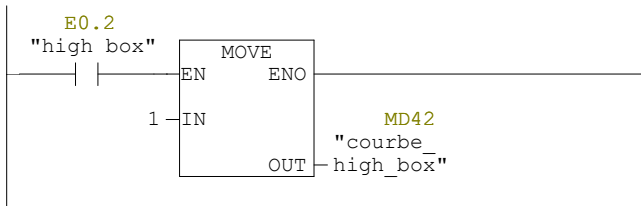
Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC1

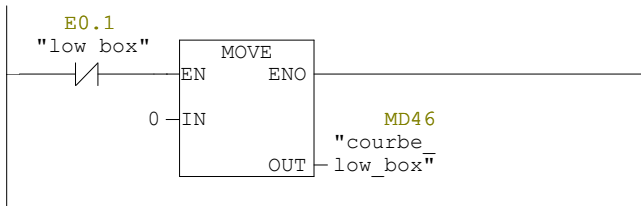
Réseau : 1



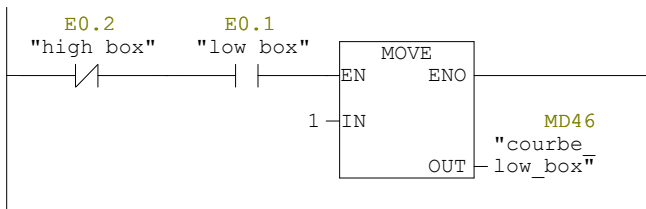
Réseau : 2



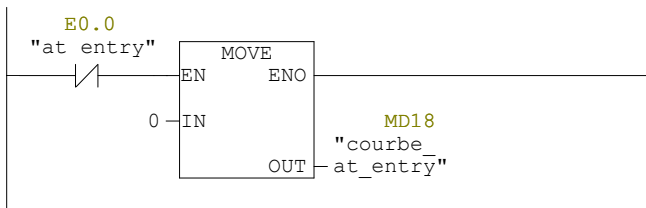
Réseau : 3



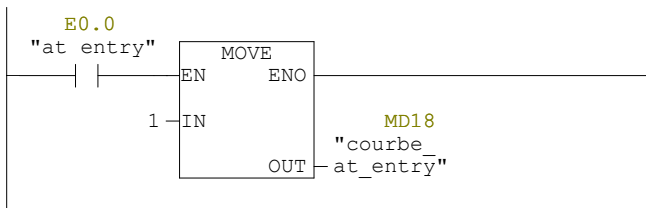
Réseau : 4



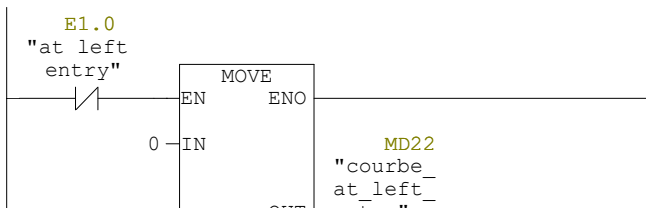
Réseau : 5



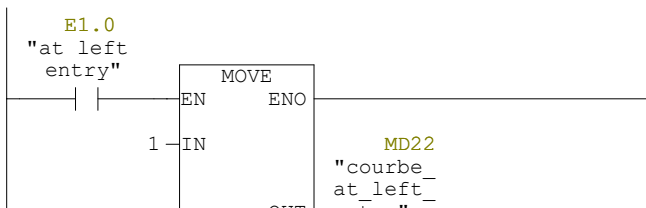
Réseau : 6



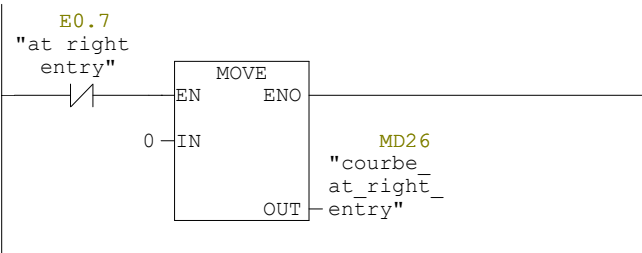
Réseau : 7



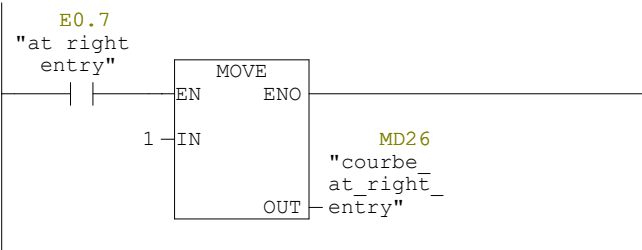
Réseau : 8



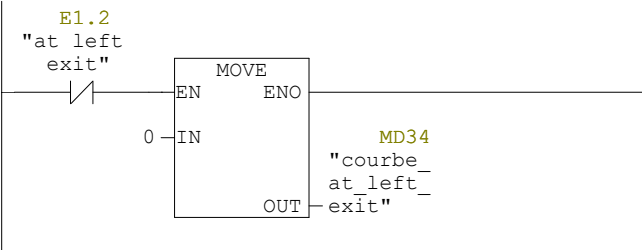
Réseau : 9



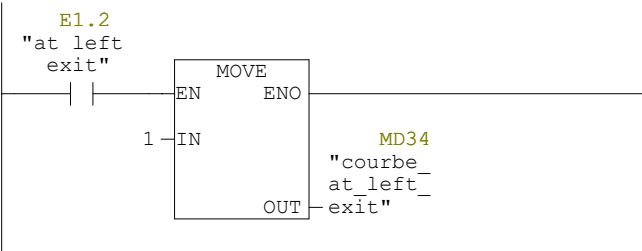
Réseau : 10



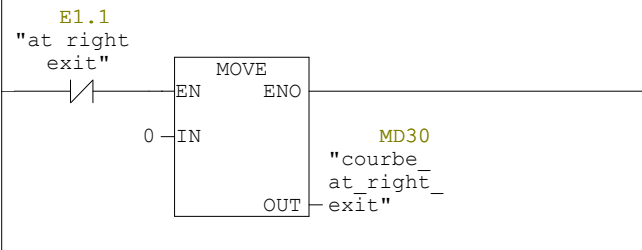
Réseau : 11



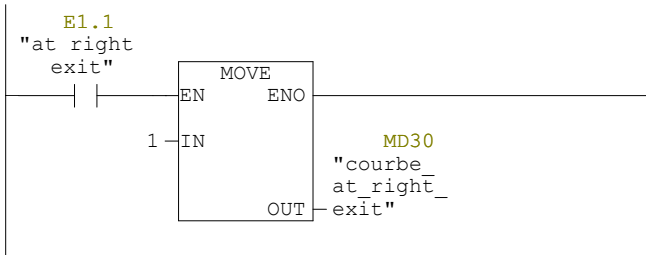
Réseau : 12



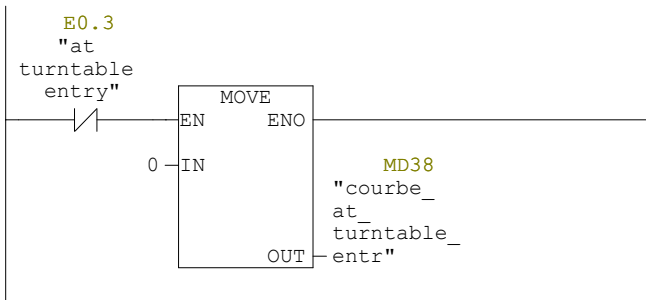
Réseau : 13



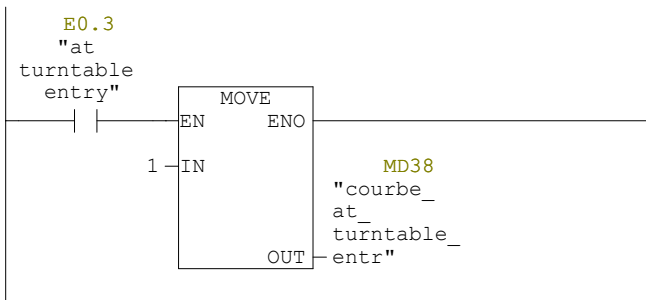
Réseau : 14



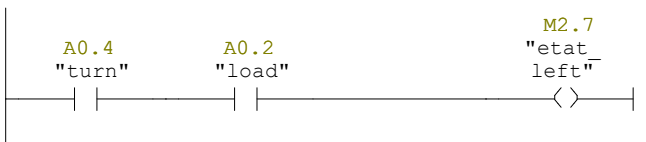
Réseau : 15



Réseau : 16



Réseau : 17



Réseau : 18



Programme de la station pick&place:

SIMATIC

pickandplace\

10/06/2024 11:04:27

SIMATIC 300(1)\CPU 312\...\OB1 - <hors ligne>

OB1 - <hors ligne>

"Cycle Execution"

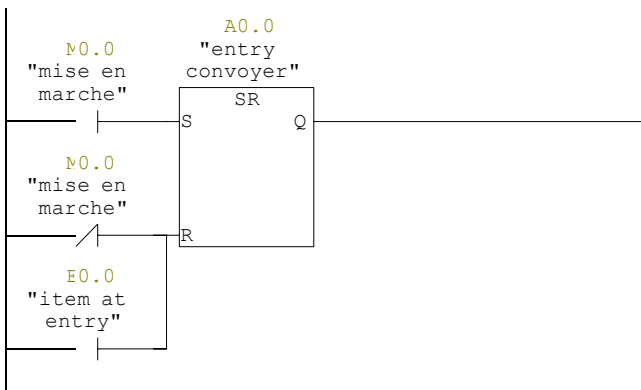
Nom :
Auteur :
Horodatage Code :
Interface :
Longueur (bloc/code /données locales) :

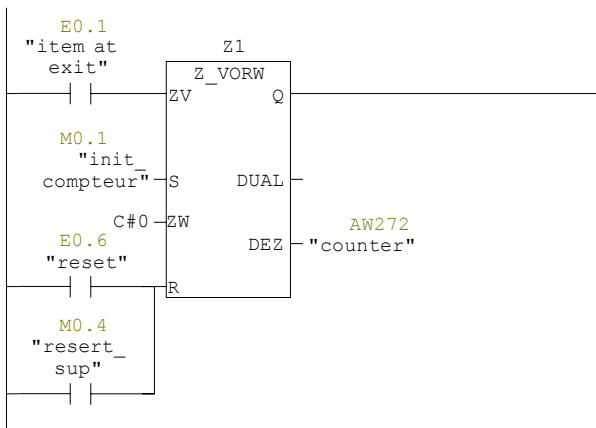
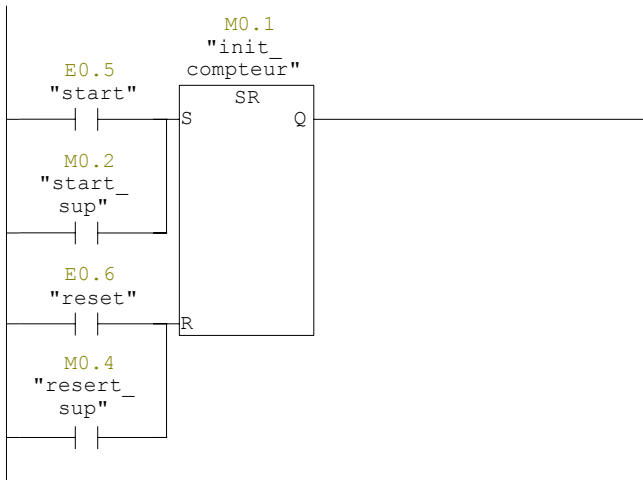
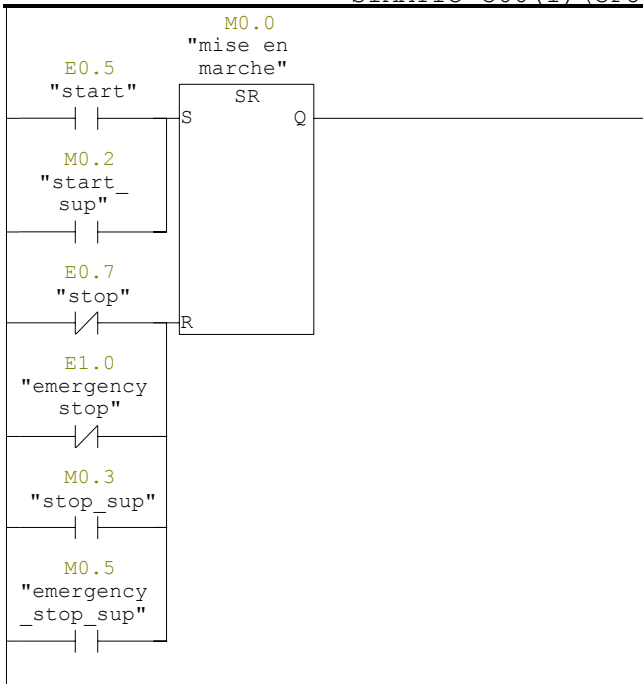
Famille :
Version : 0.1
Version de bloc : 2
01/06/2024 11:14:10
15/02/1996 16:51:12
00402 00242 00022

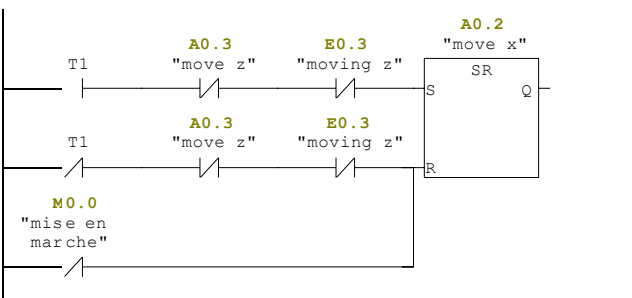
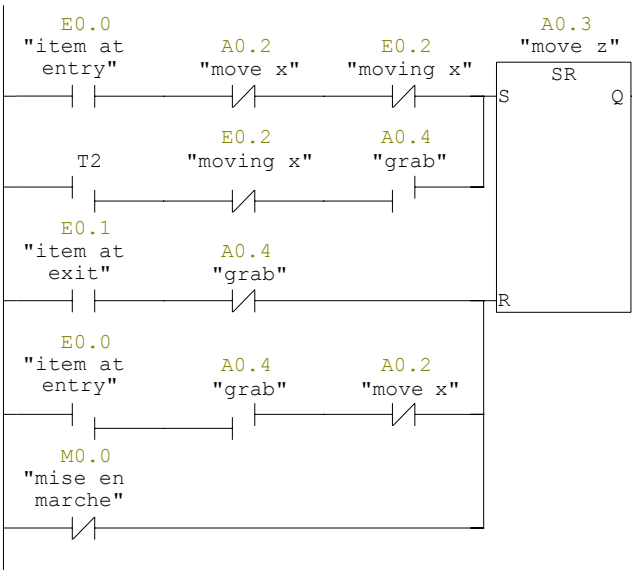
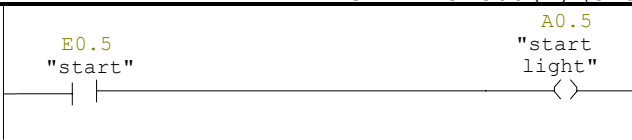
Nom	Type de données	Adresse	Commentaire
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

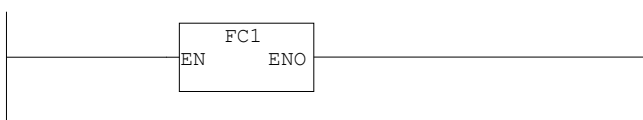
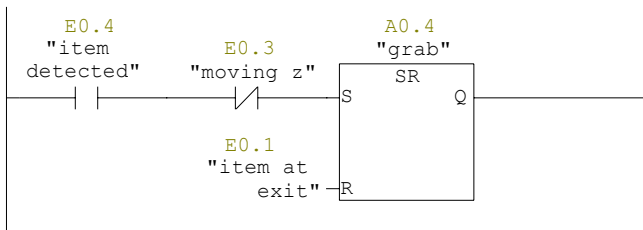
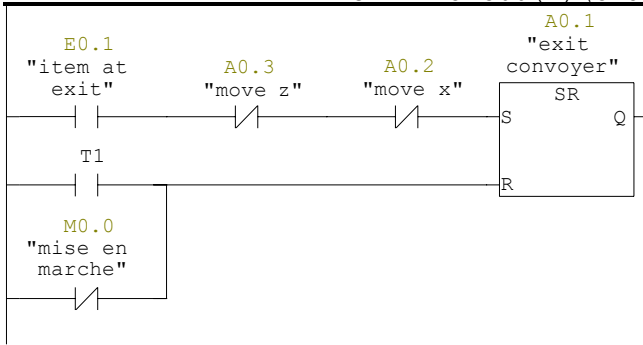
Bloc : OB1 "Main Program Sweep (Cycle)"

Réseau : 1









FC1 - <hors ligne>

""

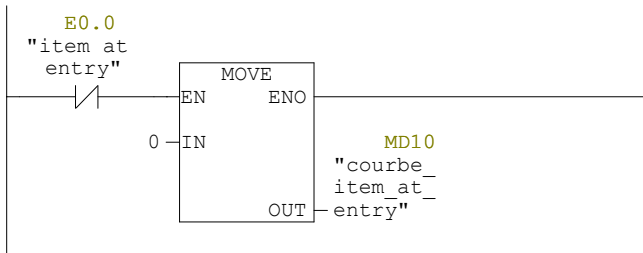
Nom :
Auteur :
Horodatage Code :
Interface :
Longueur (bloc/code /données locales) :

Famille :
Version : 0.1
Version de bloc : 2
 01/06/2024 11:13:55
 01/06/2024 11:12:13
 00200 00086 00000

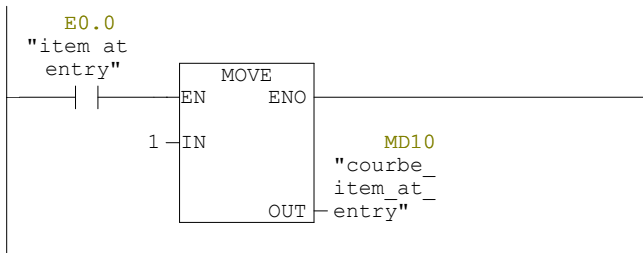
Nom	Type de données	Adresse	Commentaire
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Bloc : FC1

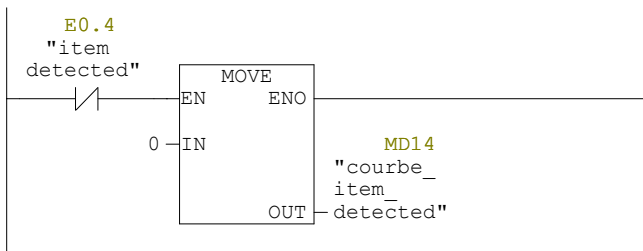
Réseau : 1



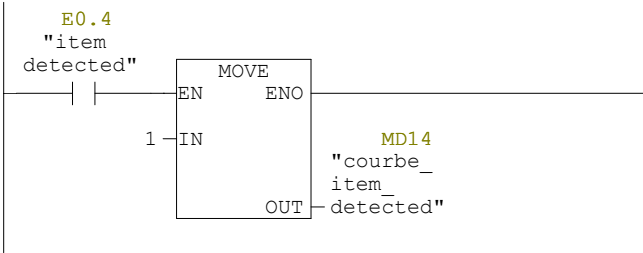
Réseau : 2



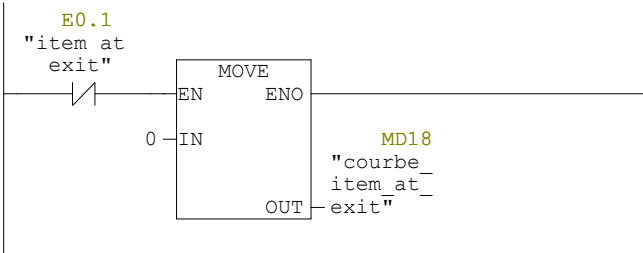
Réseau : 3



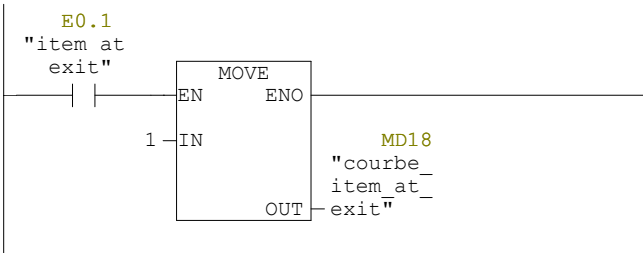
Réseau : 4



Réseau : 5



Réseau : 6



Automatisation et supervision de l'unité de tri de pièces, sous Factory I/O : Approche de modélisation par réseau de Petri.

Résumé :

Nous avons débuté par une présentation du logiciel de simulation 3D Factory I/O et l'analyse de plusieurs stations de tri basées sur la hauteur, le poids et la couleur dans le premier chapitre. Le deuxième chapitre est consacré à la modélisation des stations de tri à l'aide des RDP's et nous avons également exploré la transcription des modèles de RDP vers les modèles GRAFCET et comparé les réseaux de Petri au langage Ladder, en terminant par la transcription des réseaux de Petri en langage Ladder. Dans le troisième et quatrième chapitre nous avons présenté les outils de programmation (STEP7 et WinCC) et développé une solution de commande supervisée pour la station pick&place.

Mots clés :

Logiciel de simulation 3D Factory I/O

Stations de tri

Tri par hauteur

Tri par poids

Tri par couleur

Réseaux de Petri

Théorie des graphes

Méthodes d'analyse

Validation des modèles

Modèles GRAFCET

Langage Ladder

Conversion RdP-GRAFCET

Comparaison RdP-Ladder

Automates programmables industriels (API)

Programmation STEP7

WinCC flexible

WinCC Runtime