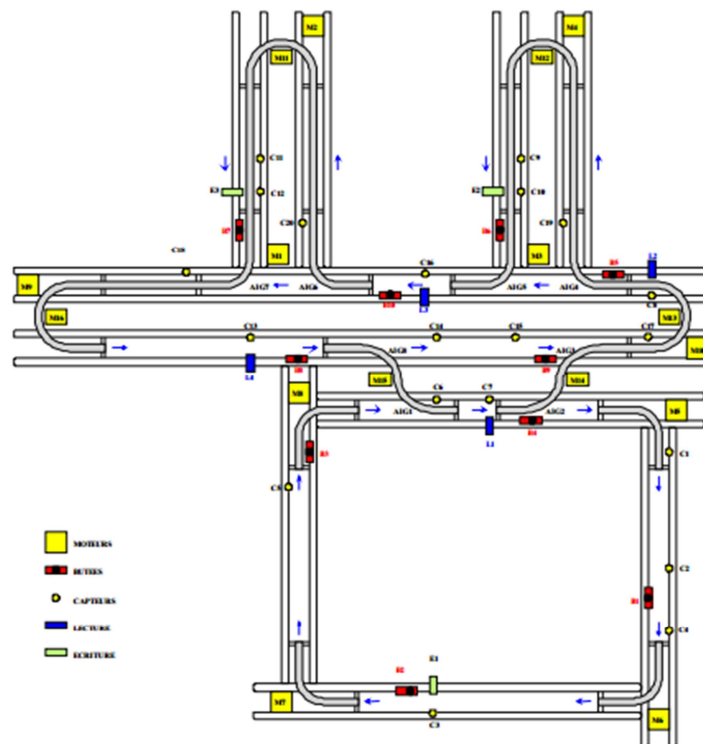


# Rapport de Projet – EI4 AGI

## ROBOTISATION DU SYSTEME TRANSITIQUE



### ***Projet réalisé par :***

Younes Lachraf  
Emeric Leclair

### ***Projet Encadré par :***

Laurent Hardouin

### ***Participation de :***

Vinicius Mariano Gonçalves

# REMERCIEMENTS

Nous tenons tout d'abord à remercier notre tuteur, Monsieur Laurent Hardouin pour avoir proposé ce projet à notre promotion et pour nous avoir suivis durant le projet. Monsieur Hassan Bouljroufi pour avoir effectué quelques réglages et modifications sur le système afin d'avoir une partie opérative opérationnelle. Monsieur Vinicius Mariano Gonçalves pour sa contribution à la modélisation et l'optimisation de la ligne transitive. Et enfin, l'ISTIA pour nous avoir fourni du matériel fonctionnel et proche de la réalité industrielle afin de mener à bien notre projet tuteuré.

# TABLE DES MATIERES

INTRODUCTION .....	5
1. PRESENTATION DU PROJET .....	6
1.1. La ligne transitique .....	6
1.2. Description du système .....	7
1.2.2. Les palettes.....	8
1.2.3. Le convoyeur .....	8
1.2.4. Les actionneurs.....	8
1.2.5. Les capteurs.....	9
1.2.7. Les automates programmables industriels .....	9
1.2.8. PC de supervision .....	9
1.2.9. La communication des éléments.....	10
2. OBJECTIFS DU PROJET .....	11
2.1. Spécifications techniques .....	11
2.1.1. Spécifications de l'automatisation du système transitique (voir Figure 2).....	11
2.1.2. Spécifications de la robotisation .....	12
2.1.3. Spécifications de la supervision locale .....	13
2.1.4. Spécifications de la supervision à distance .....	13
2.2. Travail à réaliser .....	13
2.2.1. Automatisation du système transitique.....	13
2.2.2. Robotisation .....	14
2.2.3. Supervision .....	14
3. TRAVAIL EFFECTUE .....	15
3.1 Automatisation de la ligne transitique.....	15
3.1.1. La table de données globales .....	15
3.1.2. Programmation de la station 4 (Hippodrome).....	16
3.1.3. Programmation de la station 5 (Magasin).....	18
3.1.4. Programmation de la station 6 (Epis).....	19
3.1.4. Programmation du robot .....	20
3.1.5. Changement du cahier des charges .....	21
3.1.6. Nouvelle programmation des automates .....	21
3.1.7. Le programme informatique de gestion .....	21

4. BILAN DU PROJET .....	22
4.1. : Organisation du projet .....	22
4.1.1. : Planning final .....	22
4.1.2. : Répartition des activités.....	22
4.1.3. : Répartition des tâches.....	22
4.2. : Ce qui n’a pas pu être réalisé .....	23
4.3. : Solution finale.....	23
4.4. : Bilan économique.....	23
CONCLUSION .....	24
RESUME .....	25
MOTS-CLES .....	25
ABSTRACT .....	25
ANNEXES.....	26
Annexe 1. : GRAFCETS .....	26
Annexe 1.1. : GRAFCETS Station 4.....	26
Annexe 1.2. : GRAFCETS Station 5 .....	33
Annexe 1.3. : GRAFCETS Station 6 .....	39
Annexe 2. : PROGRAMME ROBOT.....	44
Annexe 2.1. : Programme principal.....	44
Annexe 2.2. : Programme d’appels .....	44
Annexe 2.3. : Sous-programmes .....	45
Annexe 3. : Diagramme GANTT .....	47

## INTRODUCTION

De nos jours, les systèmes de production automatisés sont nécessaires pour assurer une productivité élevée, ils permettent aussi une adaptabilité aux changements du marché et aux besoins des clients. Les systèmes de convoyage sont très présents dans l'industrie agroalimentaire, les centres de tris ou les aéroports afin de faciliter la manutention des éléments au sein d'une infrastructure. En amont et en aval de ces systèmes de convoyages certaines entreprises optent pour des systèmes automatisés. En effet des machines telles que des robots ou des palettiseurs permettent le chargement ou le déchargement des éléments sur le convoyeur.



Figure 1 : Exemple de ligne transitive dans l'industrie agroalimentaire, source : <http://www.synapse-robotics.com>

La ligne transitive située en salle 216 à l'ISTIA, est en adéquation avec les lignes de production automatisées que nous pouvons trouver dans l'industrie. Elle est équipée d'un système de convoyage muni de tapis roulants, d'aiguillage et d'éléments de stoppage. Elle est aussi équipée d'un bras robotisé industriel permettant la manutention des éléments en amont et en aval de la ligne de production. Notre projet consiste à prendre en charge l'automatisation et la supervision du système transitive permettant le transport de palettes. La partie opérative et le câblage des différents organes du système sont déjà réalisés et nous ne développerons pas ces aspects durant le projet.

# 1. PRESENTATION DU PROJET

## 1.1. La ligne transitive

La ligne transitive est composée de deux organes principaux, la ligne de convoyage qui fonctionne en permanence et un robot 6 axes permettant le chargement et le déchargement des palettes. La ligne de convoyage est bouclée (une palette déposée à un endroit de la ligne reviendra forcément à son point de départ). Le contrôle de la direction des palettes au fur et à mesure de leur avancement est effectué grâce à des aiguillages disposés sur la ligne. Par conséquent, la seule solution pour l'ajout ou l'extraction des palettes est un bras robotisé situé à proximité d'une partie de la ligne, le magasin rectangulaire. Les palettes seront insérées grâce au robot dans le magasin puis circuleront dans celui-ci pour être orientées vers la partie appelée hippodrome. Une fois dans l'hippodrome, les palettes seront orientées vers une des deux parties de la ligne permettant l'usinage des pièces situées sur les palettes. Les deux parties permettant l'usinage sont appelées les épis et seront utilisées alternativement lors de l'arrivée de plusieurs palettes, cela permettra une meilleure fluidité de la ligne. Une fois les pièces usinées, les palettes retourneront dans l'hippodrome puis dans le magasin pour être déchargées par le robot. Pour illustrer cela, voici ci-dessous l'architecture de la partie opérative :

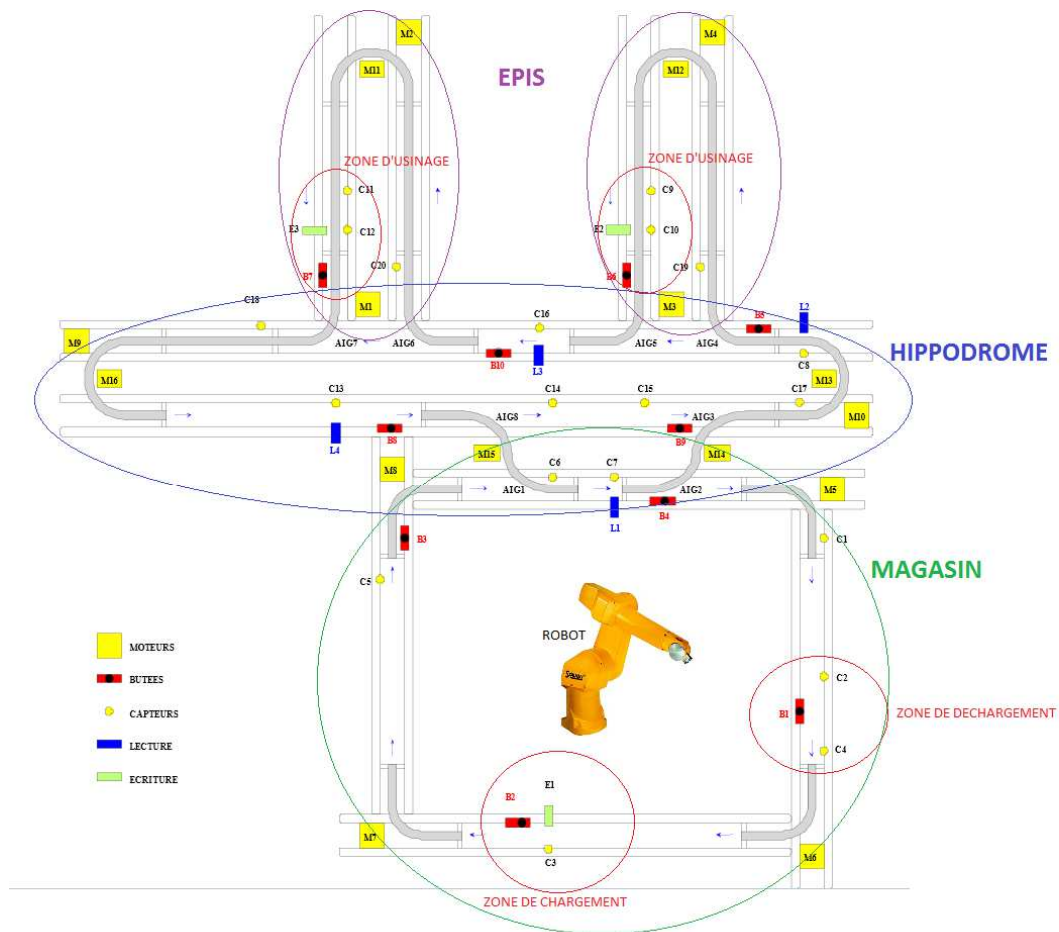


Figure 2 : Partie opérative et ses éléments

## 1.2. Description du système

La plupart des systèmes industriels sont constitués d'éléments mécaniques statiques ou dynamiques afin d'effectuer une ou des tâches précises. Les éléments mécaniques pouvant être mis en mouvement sont entraînés par des actionneurs, cela permet de modifier les caractéristiques mécaniques afin de faire évoluer le système. Cependant ces actionneurs doivent être commandés afin de faire évoluer le système selon des règles que le concepteur et/ou l'utilisateur ont établi. Pour cela des éléments intelligents de commande tel que des automates programmables permettent de définir ces règles et de commander les actionneurs par l'intermédiaire de pré-actionneurs en fonction de l'état du système en temps réel. Par conséquent, il faut des éléments permettant de connaître l'état du système, pour cela des capteurs sont disposés à des endroits stratégiques de la partie opérative. De plus, l'évolution du système doit être transmise en temps réel à l'utilisateur, pour cela la mise en place d'une interface « homme-machine » est indispensable pour la supervision du système. Tous ces éléments constituent le synoptique d'un système automatisé moderne et permettent d'effectuer une infinité de tâches complexes. La ligne transiquite est constituée selon ce schéma (voir Figure 3) que nous allons décrire plus précisément.

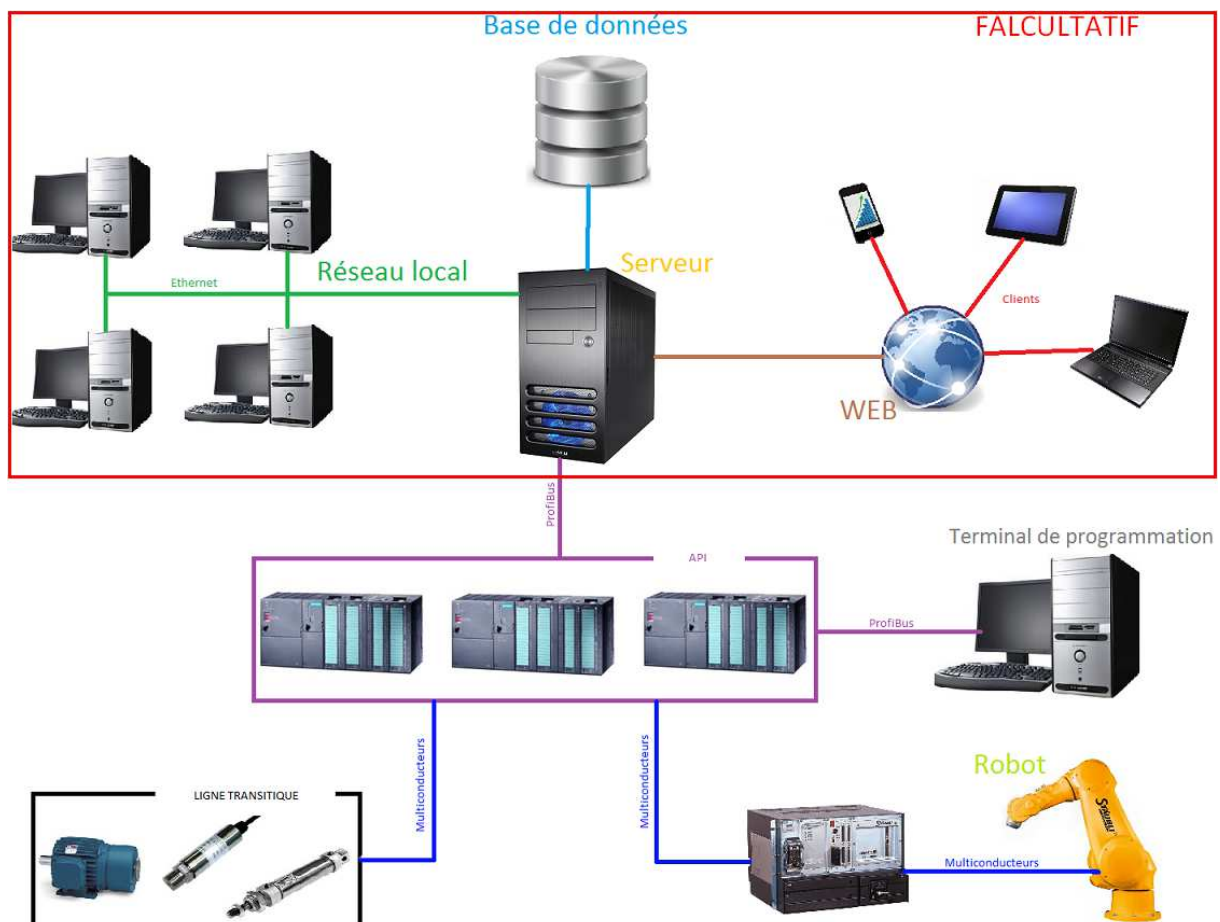


Figure 3 : Synoptique de la ligne transiquite

### 1.2.2. Les palettes

Les palettes permettent le transport des éléments à traiter sur la ligne, elles sont équipées de **marques mécaniques(1)** (voir Figure 4) permettant de savoir si la pièce a été traitée ou non. L'état de ces marques pourra être lu ou modifié par le système. Elles sont aussi équipées de 2 **roues de guidage(2)** dont le rôle sera défini ultérieurement.

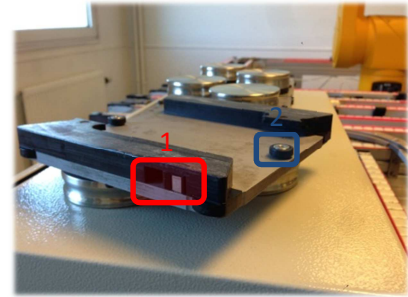


Figure 4 : Une palette

### 1.2.3. Le convoyeur

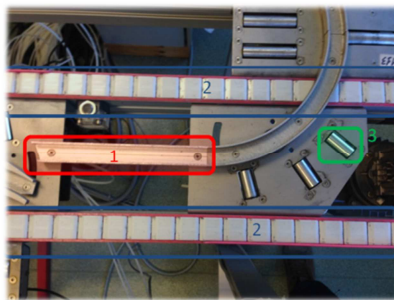


Figure 5 : Le convoyeur

Le convoyage est assuré par deux éléments, l'un fixe appelé **le rail(1)** et l'autre mobile appelé **le tapis roulant(2)**. Le premier permet le guidage des palettes sur la ligne car les roues de guidage de celle-ci viennent se placer dans le rail. Le deuxième est divisé en deux parties (voir Figure 5) et permet le déplacement des palettes grâce la translation des bandes qui sont en contact. **Des rouleaux(3)** sont aussi situés dans les virages afin de faciliter le pivotement des palettes.

### 1.2.4. Les actionneurs

Plusieurs actionneurs permettent de faire évoluer dynamiquement le système, il s'agit soit d'actionneurs électriques, soit d'actionneurs pneumatiques.

#### 1.2.4.1. Les moteurs

La ligne transitive est équipée de 11 moteurs asynchrones triphasés et de 4 moteurs monophasés. Les moteurs triphasés permettent l'entraînement des tapis roulants pour chaque portion de la ligne. Les moteurs monophasés ayant une puissance inférieure entraînent des rouleaux placés dans les virages en « épingle » du système.

#### 1.2.4.2. Les butées

La ligne transitive est équipée de 9 butées disposées aux endroits où les palettes sont susceptibles d'être bloquées (positionnées à la hauteur du rail). Elles sont commandées pneumatiquement et sont équipées d'**une pêne métallique** pouvant sortir ou se rétracter (voir Figure 6).

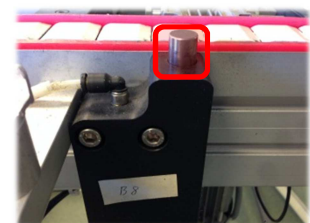


Figure 6 : Une Butée



#### 1.2.4.3 Les aiguillages

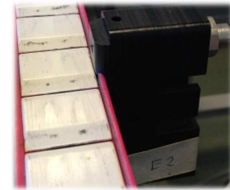
Les aiguillages sont disposés derrière certaines butées et comme leur nom l'indique ils permettent l'aiguillage des palettes dans différentes directions. Ils peuvent avoir 2 positions, lorsqu'ils sont en position



« ouvert » les palettes vont tout droit et lorsqu'ils sont fermés la palette change de direction.

### 1.2.4.3. Les marqueurs

Les marqueurs sont situés au niveau de certaines butées et permettent de changer l'état des marques des palettes.



### 1.2.5. Les capteurs

Plusieurs capteurs permettent de connaître l'état du système en temps réel, il s'agit de capteurs inductifs permettant de détecter des surfaces métalliques.

#### 1.2.5.1. Les capteurs de présence



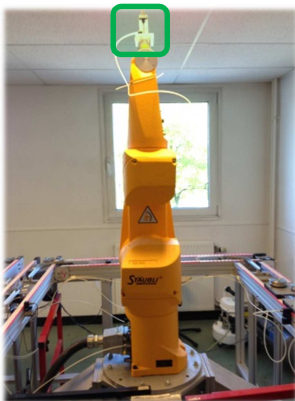
Les capteurs de présence (14 sur le système) sont situés pour la plupart au niveau des butées et permettent de détecter la présence des palettes.

#### 1.2.5.2. Les capteurs de marques

Les capteurs de marques (3 sur la ligne) sont situés pour la plupart au niveau des butées et permettent de connaître l'état des palettes (usinées ou non).



### 1.2.6. Le robot



Le système transitique est équipé d'un bras articulé 6 axes de marque Staubli, modèle RX 90. Ce robot est capable de prendre et déposer des pièces grâce à une pince pneumatique située à son extrémité. Il peut transporter des charges allant jusqu'à 12 kg avec une répétabilité de +/- 0.02mm. Ce robot est commandé par une centrale équipée d'un terminal et d'une console. Il est programmable avec le langage V+, celui-ci est très utilisé dans la robotique industrielle.

### 1.2.7. Les automates programmables industriels

Les actionneurs de la ligne transitique sont commandés par 3 automates Siemens S7-300 (1 pour le magasin, 1 pour l'hippodrome et le dernier pour le contrôle des épis), ils sont composés de cartes d'entrées et de sorties et d'un processeur réalisant l'intelligence du système. Le logiciel de programmation des automates est SIMATIC Manager de Siemens et le langage de programmation privilégié est le langage LIST créé par Siemens et utilisé sur leurs équipements.

### 1.2.8. PC de supervision

Un ordinateur situé à proximité des API permet de réaliser la supervision du système grâce à des logiciels adaptés.

### ***1.2.8.1. Le serveur OPC Applicom***

Une carte de communication Applicom Profibus permet à l'ordinateur d'être inclus dans le réseau industriel du système. Ainsi grâce au logiciel Applicom Woodhead et son serveur OPC (**O**bject **L**inking and **E**mbedding – for **P**rocess **C**ontrol), nous pouvons récupérer les différentes informations du système ou même modifier certaines variables.

### ***1.2.8.2. Programme en langage C***

Une bibliothèque Applicom est disponible pour réaliser des applications en langage C. Ainsi les possibilités d'application communicant avec le système transistrique sont conséquentes (supervision, enregistrement des évènements, traçabilité,...).

### ***1.2.8.3. La base de données***

Une base de données peut également être mise en œuvre (exemple : MySQL) afin d'enregistrer ou consulter les différentes données du système. Ainsi le développement d'application web permettant d'interagir avec le système est possible.

## **1.2.9. La communication des éléments**

Afin de permettre le fonctionnement des éléments au sein du même système, une communication est nécessaire entre tous les organes. Pour cela un réseau industriel est mis en œuvre pour faire communiquer les automates et l'ordinateur équipé du serveur OPC (Voir Figure 3). Il s'agit d'un réseau de type Profibus utilisant le protocole MPI proposé nativement sur les automates Siemens. Le robot par contre communique avec l'automate du magasin directement en câblé. En effet, certaines sorties de l'automate sont reliées aux entrées du robot qui peuvent être ensuite interprétées par le programme de ce dernier. A l'inverse les signaux de sortie du robot sont reliés à certaines entrées de l'automate et peuvent être interprétées par le programme de l'automate.

## 2. OBJECTIFS DU PROJET

L'objectif du projet est d'automatiser la ligne transitive et de programmer le robot afin d'obtenir un fonctionnement optimale et une cadence élevée tout en ayant une gestion et une signalisation des différents défauts. Les défauts devront être signalés à proximité de la ligne afin que les opérateurs soient prévenus rapidement. Une application permettant la supervision du système sera développée afin de superviser le système depuis les postes du réseau informatique de la salle. L'application devra aussi être consultable depuis l'internet afin de visualiser les défauts et le statut de la ligne de production à distance. Une solution sur système d'exploitation mobile (Android) pourra aussi être développée afin de consulter les informations depuis des appareils mobiles (tablettes tactiles, Smartphone,...).

### 2.1. Spécifications techniques

#### 2.1.1. Spécifications de l'automatisation du système transitive (voir Figure 2)

- Les tapis tourneront en permanence permettant le transport des palettes (Moteur M1 à M16).
- Le contrôle des palettes sera assuré par les butées (B1 à B10) qui bloquent ou autorisent le passage.
- La détection des palettes sera assurée par les capteurs C1 à C15.
- Chaque palette est dotée d'une marque binaire, l'écriture à 0 ou à 1 de cette marque sera assurée par les marqueurs mécaniques (E1, E2 et E3).
- La lecture de l'état d'une palette sera assurée par les capteurs de marques magnétiques (L1 à L4).

#### 2.1.1.2. Gestion entre l'hippodrome et le magasin

- Le lecteur L4 permettra de décider si la palette située en C13 transporte des pièces traitées (L4=1) ou à traiter (L4=0). La lecture de l'état d'une palette ne sera considérée valide que lorsqu'une palette sera détectée par C13 (pour L4) ou C7 (pour L1) depuis 500 ms.
- Les palettes provenant de l'hippodrome seront prioritaires par rapport à celles provenant du magasin.
- Les aiguillages seront gérés par des vérins bistables (2 sorties automate par vérins, une impulsion de 500 ms permettra le changement d'état).
- Les capteurs C6 et C14 permettront de s'assurer que les palettes parties de B3 ou B8, seront bien arrivées au-delà de l'aiguillage. La butée devra rester baisser 500 ms pour laisser passer une palette.
- Les palettes seront libérées par une butée lorsque la précédente sera sortie de la zone d'aiguillage et si les zones de stockage en aval peuvent accueillir la palette. Les capteurs C1, C6, C14 et C17 joueront le rôle de capteurs de trop plein de ces zones.

### **2.1.1.3. Gestion entre l'hippodrome et les épis**

-Le lecteur L2 permettra de décider si la palette située en C8 transporte des pièces traitées (L2=1) ou à traiter (L2=0). Le lecteur L3 permettra de décider si la palette située en C16 transporte des pièces traitées (L3=1) ou à traiter (L3=0). La lecture de l'état d'une palette ne sera considérée valide que lorsqu'une palette sera détectée par C8 (pour L2) ou C16 (pour L3) depuis 500 ms.

-Les aiguillages seront gérés par des vérins bistables (2 sorties automate par vérins, une impulsion de 500 ms permettra le changement d'état).

-Le capteur C19 (ou C20) permettra de s'assurer que la palette provenant de B5 (ou B10) est arrivée au-delà de l'aiguillage. La butée devra rester baissée 500 ms pour laisser passer une palette.

-La condition d'entrée (au niveau de l'aiguillage 4) d'une palette dans le premier épi (n°1) sera la suivante : la palette transporte de la matière première non traitée et ((il n'y pas de palette en C10) ou (il n'y a pas de palette en C9 et une palette en C12)). Lorsque la condition ne sera pas satisfaite, la palette poursuivra son parcours sur l'hippodrome.

-La condition d'entrée (au niveau de l'aiguillage 6) dans le second épi (n°2) sera la suivante : la palette transporte de la matière première non traitée et (il n'y a pas de palette en C11). Lorsque la condition ne sera pas satisfaite, la palette attendra 3 secondes en C16 puis poursuivra son parcours sur l'hippodrome.

-Lorsqu'une opération de transformation sera terminée (5 secondes passées au niveau des capteurs C10 ou C12), les modules d'écriture E2 (ou E3) marqueront la palette (action durant 500ms sur la sortie E2 ou E3). Ensuite la palette chargée de matière transformée sera évacuée vers l'hippodrome puis le magasin.

### **2.1.1.4. Gestion dans le magasin**

-Lorsqu'une palette arrivera en C2, le robot déchargera la palette. Une fois déchargée la palette continuera jusqu'au capteur C1 où elle sera chargée par le robot. Après avoir été chargée elle sera orientée vers la butée B3 pour accéder à l'hippodrome.

-Le chargement sera prioritaire par rapport au déchargement (comptage du nombre de palette en attente de chargement géré par l'automate).

## **2.1.2. Spécifications de la robotisation**

-Lorsque l'opérateur démarrera le cycle, le robot se mettra en position de repos (situé entre les positions de chargement et de déchargement).

-Lors d'un arrêt d'urgence du système, le robot finira son mouvement puis reviendra en position initiale (position de référence entrée par le constructeur du robot).

-La pince pneumatique située à l'extrémité du robot permettra d'agripper ou relâcher les pièces disposées sur les palettes.

### 2.1.3. Spécifications de la supervision locale

L'application devra permettre de:

- Sélectionner le nombre de pièces à usiner.
- Afficher l'historique de production
- Afficher l'état des voyants (Initialisation, Défaut et Mode Auto/Manu)
- Identifier les différents défauts.
- Commander le départ du cycle, le changement de mode de fonctionnement et l'arrêt du système.
- Avoir une vision globale du système (représentation graphique) en temps réel avec l'avancement des palettes → **Facultatif**

### 2.1.4. Spécifications de la supervision à distance

L'application devra permettre de:

- Sélectionner le nombre de pièces à usiner.
- Afficher l'historique de production
- Afficher l'état des voyants (Initialisation, Défaut et Mode Auto/Manu)
- Identifier les différents défauts.

## 2.2. Travail à réaliser

### 2.2.1. Automatisation du système transitique

#### 2.2.1.1. Automatisation sans détection de défaut

Dû à son architecture, le système transitique nécessite une gestion complexe et synchrone de tous les éléments pour obtenir un fonctionnement optimale. Le but sera d'élaborer un programme qui permettra la gestion et le transport des palettes sur le système transitique. Nous aurons aussi à établir la communication entre les trois automates. Afin de faciliter la tâche de programmation nous réaliserons au préalable des Grafquets permettant de définir le fonctionnement précis du système dans une forme simple et universelle. Il faudra ensuite traduire ces Grafquets en un langage interprétable par le logiciel de programmation des automates Siemens. Dans notre cas, nous utiliserons le langage d'instruction spécifique à Siemens, le LIST. Ce langage est facilement interprété par le compilateur car il est très proche du langage assembleur qui travaille au plus près du processeur. Il faudra aussi définir les mots réseaux dans la table de données globales partagée entre les stations du système sur le réseau Profibus. La programmation concernant la gestion des épis a été

faite dans le cadre de notre formation durant le TP de réseaux industriels, nous prendrons donc ce programme comme base.

### ***2.2.1.2. Automatisation avec détection de défaut***

Une fois que le fonctionnement de la ligne sera optimal, il sera possible d'ajouter au programme des Graficets de gestion des défauts. Ils devront par exemple, signaler les défauts produits par le blocage des aiguillages ou d'une palette sur la ligne. Le programme proposera ensuite à l'opérateur d'acquiescer le défaut pour permettre un « dialogue » entre l'utilisateur et le programme.

## **2.2.2. Robotisation**

### ***2.2.2.1. Programmation du robot sans communication***

Dans cette étape, nous aurons à nous familiariser avec l'utilisation et la programmation du robot Staubli. Nous devons définir les différents mouvements à exécuter par le bras afin qu'il effectue les tâches souhaitées. Il faudra ensuite programmer ses mouvements en VB+ à partir du logiciel de programmation. La finalité sera d'obtenir un cycle complet de chargement et déchargement des palettes.

### ***2.2.2.2. Programmation du robot avec communication***

Dans cette partie, nous pourrions faire communiquer le robot avec les automates afin que ces tâches soient synchronisées avec le cycle de la ligne.

## **2.2.3. Supervision**

Les fonctions de cette partie se résument en deux axes principaux :

### ***2.2.3.1. Supervision local***

La supervision locale du système de transport de palettes sera faite à l'aide du réseau MPI et du serveur OPC Applicom qui nous permettra de récupérer l'état des différents capteurs/actionneurs et de les enregistrer dans une base de données locale. Ensuite c'est avec une interface graphique développée avec l'environnement de programmation C++ et constituée de formulaires, boutons, boutons, labels et TextBox qu'on pourra faire interagir l'utilisateur avec le système.

### ***2.2.3.2. Supervision à distance***

La supervision à distance consistera à développer un modèle client/serveur : Constitué d'une simple page d'accueil internet qui sera réalisée grâce aux langages de programmation web PHP, HTML et CSS. La communication sera effectuée à l'aide du protocole TCP/IP afin d'établir la connexion à distance. Cette page d'accueil affichera les informations utiles pour la supervision.

### 3. TRAVAIL EFFECTUE

Dans ce chapitre nous allons vous présenter le travail réalisé à l'aide de diagrammes tels que des grafjets, des schémas et du code. Ces éléments seront exposés en annexe de ce rapport mais seront commentés dans cette partie.

#### 3.1 Automatisation de la ligne transitique

Nous allons détailler les grafjets et leur programmation pour chaque station et leurs rôles dans le fonctionnement du système.

*Information 1 :* Nous avons choisi de nommer les étapes des grafjets selon le bit interne utilisé dans le programme des automates, par exemple, l'étape 150 correspondra au bit interne M15.0 du programme. Cela a permis de faciliter la tâche de traduction et de programmation.

*Information 2 :* Étant donné que le système nécessite une synchronisation entre les différentes opérations, notre conception des grafjets est composée de beaucoup de sémaphore. Cela consiste à demander l'autorisation à un automate distant d'effectuer une action, cet automate distant va recevoir la demande, la traiter puis donner l'autorisation si les conditions sont respectées.

##### 3.1.1. La table de données globales

Lorsque plusieurs automates Siemens nécessitent une communication entre eux au sein d'un réseau industriel, une table de données globales doit être définie. Cette table définit les mots qui communiqueront entre les stations et attribut à chaque stations l'accès en lecture ou en écriture à chacun des mots. Dans notre cas nous avons définis 4 mots en écriture et 8 mots en lecture pour chaque station. Ci-dessous la table de données globales de notre projet :

Données globales du sous-réseau 'lnit-pgbase\MPI(1)'				
	Identificateur GD	Station epis\ CPU314	Station hippodrome\ CPU314(1)	Station magasin\ CPU314
1	GD 1.1.1	>MB124:4	MB124:4	MB124:4
2	GD 2.1.1	MB120:4	>MB120:4	MB120:4
3	GD 3.1.1	MB116:4	MB116:4	>MB116:4
4	GD			
5	GD			
6	GD			
7	GD			
8	GD			

Figure 7 : Table de données globales, source : Enoncé de TP Profibus (Ei4)

## 3.1.2. Programmation de la station 4 (Hippodrome)

### 3.1.2.1. GRAFCETS de la station 4

-Annexe 1.1.1. : Ce grafcet permet de définir le mode de fonctionnement du système en fonction de la sélection de l'opérateur.

-Annexe 1.1.2. : Ce grafcet permet l'arrêt du système en cas d'arrêt d'urgence.

-Annexe 1.1.3. : Ce grafcet permet de gérer la synchronisation des stations lors de l'initialisation du système.

-Annexe 1.1.4. : Ce grafcet permet le démarrage asynchrone des moteurs lors du démarrage du système. Ce choix de démarrage permet de limiter l'appel de courant en démarrant les moteurs les uns après les autres.

-Annexe 1.1.5. : Ce grafcet reçoit les demandes de passage depuis la butée B5 (Annexe 1.1.6.), il demande aux aiguillages 4 et 5 (Annexe 1.3.4.) de changer d'état.

-Annexe 1.1.6. : Ce grafcet permet de choisir la direction lorsqu'une palette arrive à la butée B5, il contrôle d'abord l'état de la palette (capteur L2). Puis en fonction de la disponibilité des épis, il demande l'autorisation de passage (Annexe 1.1.5.). Une fois l'autorisation reçue il baisse la butée B5.

-Annexe 1.1.7. : Ce grafcet permet de choisir la direction lorsqu'une palette arrive à la butée B10, il contrôle d'abord l'état de la palette (capteur L3). Il demande l'autorisation de passage (Annexe 1.1.8.). Une fois l'autorisation reçue il baisse la butée B10.

-Annexe 1.1.8. : Ce grafcet permet de faire le choix entre les demandes de passages au niveau des butées B7 et B10. En fonction de la demande, il va demander le changement d'état des aiguillages 6 (Annexe 1.1.10.) et 7 (Annexe 1.3.11.).

-Annexe 1.1.9. : Ce grafcet permet de renseigner l'état du capteur C18.

-Annexe 1.1.10. : Ce grafcet gère les demandes d'ouverture et de fermeture (Annexe 1.1.8.) de l'aiguillage 6.

-Annexe 1.1.11. : Ce grafcet permet de contrôler la butée B3 lorsqu'une palette arrive. Il demande l'autorisation de passage vers l'hippodrome (Annexe 1.1.13.). Une fois l'autorisation reçue il baisse la butée B3.

-Annexe 1.1.12. : Ce grafcet permet de choisir la direction lorsqu'une palette arrive à la butée B8, il contrôle d'abord l'état de la palette (capteur L4). Puis il demande l'autorisation de passage (Annexe 1.1.14. ou Annexe 1.1.15.). Une fois l'autorisation reçue il baisse la butée B8.

-Annexe 1.1.13. : Ce grafcet reçoit la demande de passage vers C6 (Annexe 1.1.11.) puis il vérifie les conditions et qu'il n'y a pas de palette en C6 (Annexe 1.2.8.). Il change l'état des aiguillages puis il autorise le passage jusqu'à ce que la palette atteigne C6 (Annexe 1.2.10.).

-Annexe 1.1.14. : Ce grafcet reçoit la demande de passage vers C6 (Annexe 1.1.12.) puis il vérifie les conditions et qu'il n'y a pas de palette en C6 (Annexe 1.2.8.). Il change l'état des aiguillages puis il autorise le passage jusqu'à ce que la palette atteigne C6 (Annexe 1.2.10.).



-Annexe 1.1.15. : Ce grafctet reçoit la demande de passage vers C14 (Annexe 1.1.12.) puis il vérifie les conditions et qu'il n'y a pas de palette en C14 (Annexe 1.2.9.). Il change l'état des aiguillages puis il autorise le passage jusqu'à ce que la palette atteigne C14 (Annexe 1.2.11.).

### 3.1.2.2. Les entrées/sorties de la station 4

Dans le tableau ci-dessous, nous pouvons observer la liste des entrées et des sorties de la station gérant l'hippodrome et leurs adresses correspondantes.

Appellation	Adresse	Appellation	Adresse
Aig1_ferme	A5.0	Moteur9	A4.0
Aig1_ouvert	A5.4	Moteur10	A4.1
Aig4_ouvert	A5.5	Moteur13	A4.2
Aig4_ferme	A5.1	Moteur16	A4.3
Aig6_ouvert	A5.6	C13	E0.5
Aig6_ferme	A5.2	C16	E0.6
Aig8_ouvert	A5.7	C18	E1.3
Aig8_ferme	A5.3	C5	E0.3
B10	A4.5	C8	E0.2
B3	A4.7	L2	E0.7
B5	A4.4	L3	E1.0
B8	A4.6	L4	E1.2
		ARU	E0.0
		DCY	E0.1

Figure 8 : Table d'adressage des E/S station 4, source : Enoncé de TP Profibus (Ei4)

### 3.1.2.3. Exemples de traduction en langage LIST

La tâche la plus contraignante fut de traduire les grafctets en langage LIST afin de réaliser le programme qui sera entré dans l'automate. Ci-dessous nous avons développé la traduction du grafctet de gestion de la butée B3 (Annexe 1.1.11.).

```

U M12.0          //étape 120
U E0.3           //C5
UN M123.5       //Demande C5 C6 (bit de la table de données globales)
R M12.0         //étape 120
S M12.1         //étape 121

U M12.1         //étape 121
U M123.5       //Demande C5 C6 (bit de la table de données globales)
R M12.1         //étape 121
S M12.2         //étape 122

U M12.2         //étape 122
U T110          //temporisation de 500 ms
R M12.2         //étape 122
S M12.0         //étape 120

```

### 3.1.3. Programmation de la station 5 (Magasin)

#### 3.1.3.1. GRAFCETS de la station 5

-Annexe 1.2.1. : Ce grafcet permet le démarrage asynchrone des moteurs lors du démarrage du système.

-Annexe 1.2.2. : Ce grafcet permet l'arrêt du système en cas d'arrêt d'urgence.

-Annexe 1.2.3. : Ce grafcet permet de choisir la direction lorsqu'une palette arrive à la butée B4, il contrôle d'abord l'état de la palette (capteur L1). Puis il demande l'autorisation de passage (Annexe 1.2.5. ou Annexe 1.2.7.). Une fois l'autorisation reçue il baisse la butée B4.

-Annexe 1.2.4. : Ce grafcet permet de contrôler la butée B9 lorsqu'une palette arrive. Il demande l'autorisation de passage vers l'hippodrome (Annexe 1.2.6.). Une fois l'autorisation reçue il baisse la butée B9.

-Annexe 1.2.5. : Ce grafcet reçoit la demande de passage vers C1 (Annexe 1.2.3.) puis il vérifie les conditions. Il change l'état des aiguillages puis il autorise le passage jusqu'à ce que la palette atteigne C1.

-Annexe 1.2.6. : Ce grafcet reçoit la demande de passage vers C17 (Annexe 1.2.4.) puis il vérifie les conditions. Il change l'état des aiguillages puis il autorise le passage jusqu'à ce que la palette atteigne C17.

-Annexe 1.2.7. : Ce grafcet reçoit la demande de passage vers C17 (Annexe 1.2.3.) puis il vérifie les conditions. Il change l'état des aiguillages puis il autorise le passage jusqu'à ce que la palette atteigne C17.

-Annexe 1.2.8. & 1.2.10. : Ce grafcet permet de renseigner l'état du capteur C6.

-Annexe 1.2.9. & 1.2.11. : Ce grafcet permet de renseigner l'état du capteur C14.

-Annexe 1.2.12. : Ce grafcet permet de contrôler la butée B1 lorsqu'une palette arrive. Il demande le déchargement de la palette (Annexe 1.2.14.). Une fois le déchargement terminé il baisse la butée B1.

-Annexe 1.2.13. : Ce grafcet permet de contrôler la butée B2 lorsqu'une palette arrive. Il demande le chargement de la palette (Annexe 1.2.14.). Une fois le chargement terminé il change l'état de la palette et baisse la butée B2.

-Annexe 1.2.14. : Ce grafcet permet la commande du robot en fonction des palettes présentes dans le magasin. Il demande la mise en position repos du robot lors du démarrage du cycle. Lorsque la position est atteinte il attend soit une demande de chargement (Annexe 1.2.13.) soit une demande de déchargement (Annexe 1.2.12.). Il demande ensuite un déplacement vers la position demandée, une fois la position atteinte il demande l'exécution de l'action (chargement ou déchargement). Dès que l'action est terminée, il attend une nouvelle demande (Annexe 1.2.12. ou Annexe 1.2.13.). Si une demande de déchargement est effectuée alors que le robot est du côté du chargement, il vérifie qu'il n'y a pas de palettes vides (Annexe 1.2.15.). Sinon aucune vérification n'est effectuée pour les autres situations.

-Annexe 1.2.15. : Ce grafcet réalise le comptage des palettes entre les butées B1 et B2.

### 3.1.3.2. Les entrées/sorties de la station 5

Dans le tableau ci-dessous, nous pouvons voir la liste des entrées et des sorties de la station gérant le magasin et leurs adresses correspondantes.

Appellation	Adresse	Appellation	Adresse
Aig2_ferme	A4.0	C1	E0.1
Aig2_ouvert	A4.2	C2	E0.2
Aig3_ouvert	A4.3	C3	E0.3
Aig3_ferme	A4.1	C4	E0.4
B1	A4.4	C6	E0.6
B2	A4.5	C7	E0.7
B4	A4.6	C14	E1.4
B9	A4.7	C15	E1.5
E1	A5.6	C17	E1.7
Moteur5	A5.0	L1	E0.0
Moteur6	A5.1		
Moteur7	A5.2		
Moteur8	A5.3		
Moteur14	A5.4		
Moteur15	A5.5		

Figure 9 : Table d'adressage des E/S station 5, source : Enoncé de TP Profibus (Ei4)

### 3.1.4. Programmation de la station 6 (Epis)

#### 3.1.4.1. GRAFCETS de la station 6

-Annexe 1.3.1. : Ce grafcet permet le démarrage asynchrone des moteurs lors du démarrage du système.

-Annexe 1.3.2. : Ce grafcet permet l'arrêt du système en cas d'arrêt d'urgence.

-Annexe 1.3.3. : Ce grafcet permet de contrôler la butée B6 lorsqu'une palette arrive. Il demande l'autorisation de passage. Puis il attend 5s puis il change l'état de la palette et baisse la butée B6.

-Annexe 1.3.4. : Ce grafcet gère les demandes d'ouverture (Annexe 1.1.5.) et de fermeture (Annexe 1.3.3.) de l'aiguillage 5.

-Annexe 1.3.5. : Ce grafcet permet de renseigner l'état du capteur C9.

-Annexe 1.3.6. : Ce grafcet permet de renseigner l'état du capteur C10.

-Annexe 1.3.7. : Ce grafcet permet de renseigner l'état du capteur C12.

-Annexe 1.3.8. : Ce grafcet permet de contrôler la butée B7 lorsqu'une palette arrive. Il demande l'autorisation de passage (Annexe 1.1.8.). Puis il attend 5s puis il change l'état de la palette et baisse la butée B6.

-Annexe 1.3.9. : Ce grafcet permet de renseigner l'état du capteur C11.

-Annexe 1.3.10. : Ce grafcet permet de renseigner l'état du capteur C20.

-Annexe 1.3.11. : Ce grafcet gère les demandes d'ouverture et de fermeture (Annexe 1.1.8.) de l'aiguillage 7.

### 3.1.4.2. Les entrées/sorties de la station 6

Dans le tableau ci-dessous, nous pouvons voir la liste des entrées et des sorties de la station gérant les épis et leurs adresses correspondantes.

Appellation	Adresse	Appellation	Adresse
Aig5_ferme	A5.0	C9	E0.0
Aig5_ouvert	A5.2	C10	E0.1
Aig7_ferme	A5.1	C11	E0.2
Aig7_ouvert	A5.3	C12	E0.3
B6	A5.5	C19	E0.4
B7	A5.7	C20	E0.5
E2	A5.4		
E3	A5.6		
Moteur1	A4.0		
Moteur2	A4.1		
Moteur3	A4.2		
Moteur4	A4.3		
Moteur11	A4.4		
Moteur12	A4.5		

Figure 10 : Table d'adressage des E/S station 6, source : Enoncé de TP Profibus (Ei4)

### 3.1.4. Programmation du robot

Une fois le grafcet de gestion du robot effectué (Annexe 1.2.14.), nous avons procédé à la programmation du robot. Cette programmation est effectuée en 3 étapes. D'abord nous codons la boucle principale qui analyse en permanence l'état des actions du grafcet de gestion et appelle les sous-programmes correspondants. Ensuite nous pouvons définir pour chaque sous-programme les mouvements effectués chronologiquement par le robot. La dernière étape est de définir les coordonnées à atteindre pour chaque mouvement.

Finalement nous avons obtenu un fonctionnement avec un programme principal fonctionnant même en cas d'ARU (Annexe 2.1.), un programme d'appel (Annexe 2.2.) et 6 sous-programmes (Annexe 2.3.). Dans les sous-programmes les mouvements sont opérés axe par axe, ci-dessous, le schéma représentant les différents axes du robot :

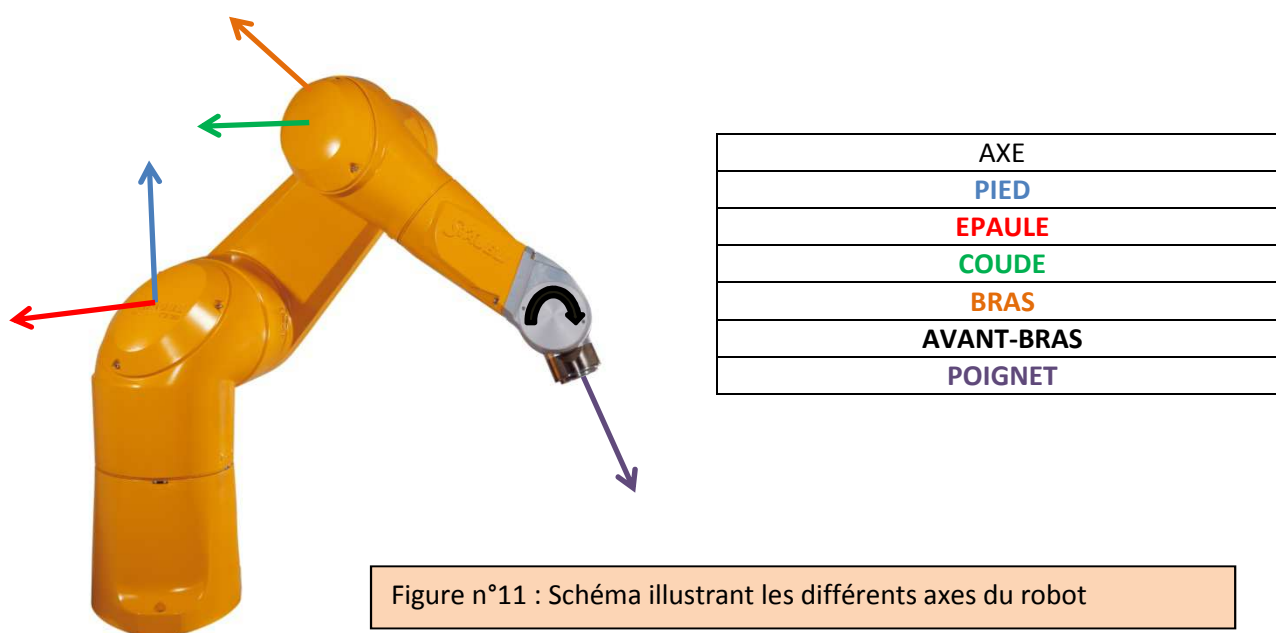


Figure n°11 : Schéma illustrant les différents axes du robot

### 3.1.5. Changement du cahier des charges

A la mi-Février, Vinicius, un doctorant d'origine brésilienne, est venu contribuer à notre projet en réalisant une recherche sur la modélisation et l'optimisation de la ligne transitaire. Le cahier des charges a donc été modifié afin de lui permettre de réaliser ces tests et ces mesures sur le système à l'aide d'un programme informatique. Vinicius avait réalisé au préalable le réseau de Pétri correspondant à notre système, il souhaitait donc réaliser ces essais dans l'objectif de valider son modèle. Pour cela, il nous a demandé de faire fonctionner le système de la manière suivante :

-Le système fonctionne maintenant avec deux boucles distinctes, la première est la boucle magasin (les aiguillages 1 et 2 sont ouverts en permanence) et la deuxième boucle est l'hippodrome et les épis (les aiguillages 4, 5, 6 et 7 sont fermés en permanence).

-Le passage des palettes au niveau des butées B4 et B9 sera synchronisé (exemple : une palette arrivée en B4 attend qu'une palette arrive en B9, lorsque la condition est respectée, les deux butées se baissent).

-Seulement 2 palettes doivent être présentes dans chaque épi.

-Seulement 3 palettes doivent être présentes en les butées B4 et B1.

Pour effectuer les mesures de manière optimum, ce sera le programme informatique qui réalisera la lecture des capteurs et la commande des butées en temps réel.

### 3.1.6. Nouvelle programmation des automates

Nous avons revu tous nos graficets afin d'obtenir le programme voulu par Vinicius. Son souhait était que les automates permettent uniquement d'effectuer les « relais » entre son programme informatique et le système. Ainsi pour chaque butée, le programme informatique écrit dans un mot en commun avec l'automate, ce mot sera lu par l'automate et il changera l'état de la butée (voir Figure 12).



Figure n°12 : Exemple en Ladder de la commande d'une butée

Ici, nous testons la valeur du mot commun (%MW98) et si les conditions sont bonnes, le programme commande la butée.

### 3.1.7. Le programme informatique de gestion

Le programme est effectué sur l'ordinateur de la salle équipée de la carte réseau et du serveur OPC. Ce programme réalisé en langage C, utilise une bibliothèque Applicom afin de lire et écrire dans les variables du serveur OPC (relié sur le réseau industrielle par la carte sur le réseau). Il permet lorsque le système transitaire est en fonctionnement de lire l'état des capteurs de présence,

d'enregistrer un événement correspondant (fichier ou base de données) lorsqu'une palette arrive sur un capteur et de demander l'activation de la butée à l'automate (voir Figure 12) et tout cela en temps réel (voir programme « .c » disponible sur Dropbox, section « Codes Supervision »).

## 4. BILAN DU PROJET

### 4.1. : Organisation du projet

#### 4.1.1. : Planning final

Nous avons réalisé un diagramme Gant montrant comment les différentes activités devront se répartir au fil des semaines de projet (voir Annexe 3).

#### 4.1.2. : Répartition des activités

Nous avons réalisé un diagramme circulaire montrant la part de ces différentes activités dans le projet (voir Figure 13).

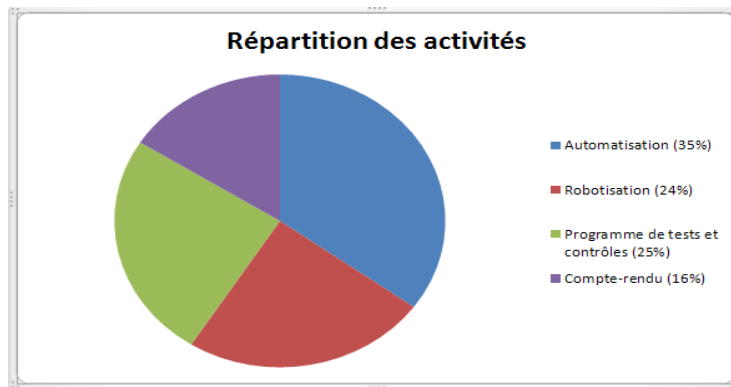


Figure n°13 : Schéma illustrant les différents axes du robot

#### 4.1.3. : Répartition des tâches

Nous avons réalisé plusieurs diagrammes circulaires montrant nos parts de travail selon les différentes activités du projet (voir Figure 14).

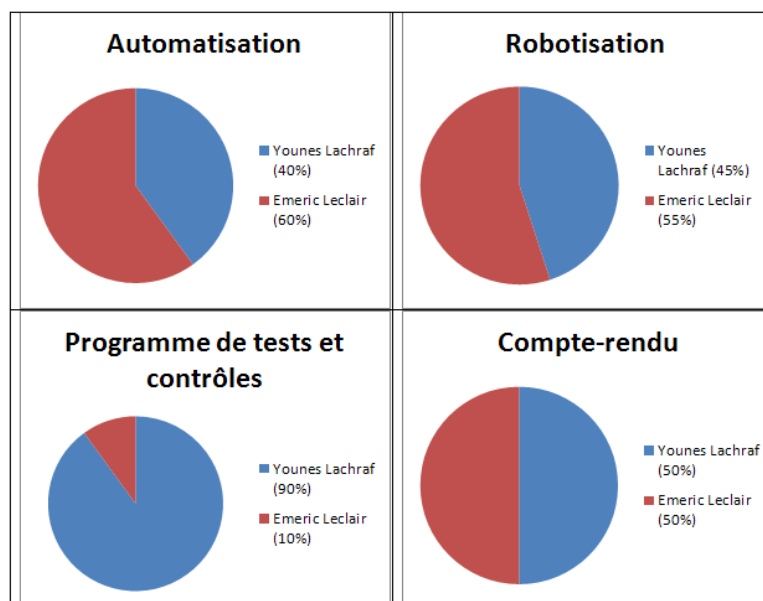


Figure n°14 : Schéma illustrant la répartition des tâches

## 4.2. : Ce qui n'a pas pu être réalisé

Etant donné que le cahier des charges a été modifié, nous n'avons pas pu effectuer la supervision locale et la supervision à distance du système transitique. Tous les défauts n'ont pas pu être gérés sur le système transitique. La base de données enregistrant les événements se produisant sur le système n'a aussi pas pu être réalisée faute de temps.

## 4.3. : Solution finale

Le cahier des charges ayant changé, deux solutions finales ont été obtenus, une permettant le fonctionnement de la ligne avec un robot de manutention et l'autre faisant fonctionner le système de manière simple mais avec un système d'acquisition de données en temps réel. Ces deux solutions présentent deux aspects différents mais relativement intéressants. En effet, nous pouvons caractériser la première solution comme semblable à un système de manutention que nous pourrions trouver dans l'industrie. Elle est donc intéressante pour nous, futurs ingénieurs, car elle permet d'anticiper sur la gestion d'un projet semblable à ce que nous serions susceptibles de faire une fois dans le monde du travail. La deuxième solution à un caractère plus expérimentale et a pour but de valider des travaux de recherches, elle permet donc de prendre du recul sur le fonctionnement de notre système transitique dans le but de l'optimiser.

## 4.4. : Bilan économique

Notre projet n'ayant nécessité aucun achat de matériel, nous avons estimé le coût de notre prestation sur un tel projet (voir Figure 15). Le tarif a été défini à partir de grille de tarif de prestation d'un ingénieur d'études (*source : <http://tarifs.freelance-info.fr/ingenieur-etudes>*).

Mission	Heures de travail	Tarif horaire	Tarif en €
Automatisation	50	40€	2000
Robotisation	35		1400
Programme de tests et contrôles	37		1480
<b>TOTAL</b>			<b>4880</b>

Figure n°15 : Tableau récapitulatif des coûts estimés sur ce projet

## CONCLUSION

Malgré un changement d'objectifs en cours de projet, nous avons su nous réorganiser, et répondre aux besoins de Vinicius. Ainsi nous avons réussi à mettre en place une solution fonctionnelle et optimisée. Le travail en équipe avec la présence de Vinicius a été enrichissant pour chacun d'entre nous.

Bien que nous ne soyons pas parvenus à remplir totalement les objectifs qui étaient de réaliser la supervision locale et à distance, en plus de la commande du système transitique. Nous avons pu mettre en place une solution qui répond en grande partie des attentes du cahier des charges.

Ce projet nous a permis d'acquérir des méthodes et des réflexes essentiels dans la recherche et la mise en place de solutions adaptées. Nous avons également acquis un certain nombre de connaissances sur la gestion d'un système transitique, sur la programmation industrielle et informatique, ainsi que sur la programmation d'un robot industriel, sur l'acquisition de données sur un système industriel et plus globalement sur la gestion d'un projet. Ce projet nous a permis de mettre en pratique nos cours théoriques et de développer notre capacité à la gestion de projet.



# ROBOTISATION DU SYSTEME TRANSITIQUE

**Projet réalisé par :** Younes Lachraf  
Emeric Leclair

**Projet encadré par :** Laurent Hardouin

## RESUME

Ce rapport rend compte de notre travail réalisé lors de notre projet tutoré effectué en Ei4 à l'ISTIA. Notre projet consiste à réaliser l'automatisation, la robotisation et la réalisation d'un programme de tests et de contrôles sur une ligne transitive présente dans une salle de notre établissement. Dans ce document, nous décrivons le système et exposons les solutions techniques utilisées pour parvenir à une solution optimale. Deux solutions finales ont été obtenues, une permettant le fonctionnement de la ligne avec un robot de manutention et l'autre faisant fonctionner le système de manière simple mais avec un système d'acquisition de données en temps réel.

## MOTS-CLES

Transitive, Automatisation, Robotisation, GRAFCET, langage C, langage V+, sémaphores.

## ABSTRACT

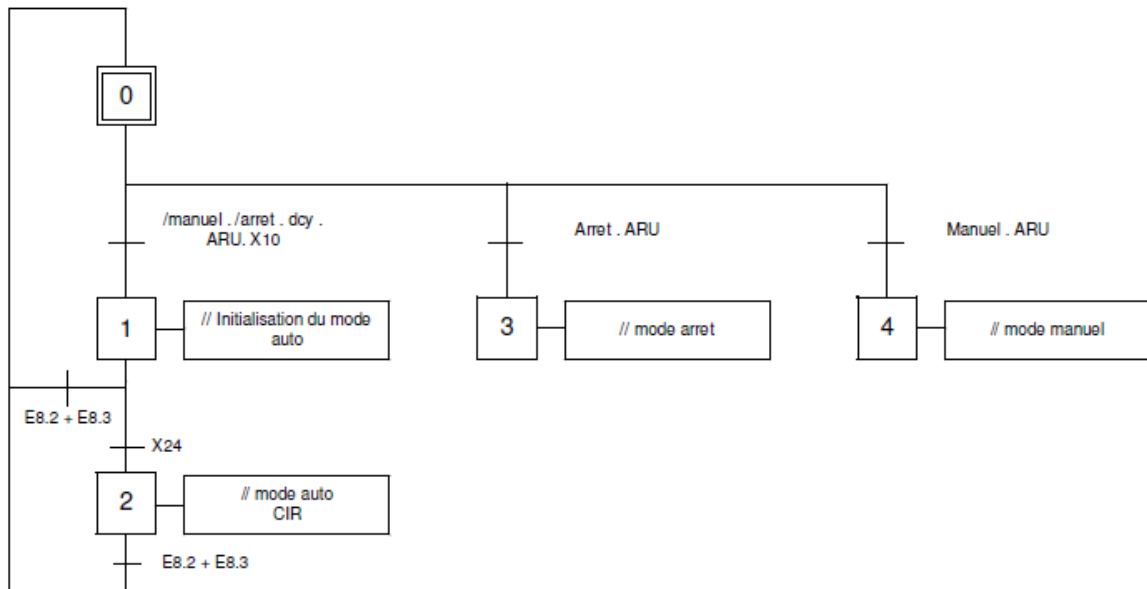
This report talk about our work done during our tutored project carried out in Ei4 at ISTIA. Our project was to achieve automation, robotics and implementation of a testing program and control on a chain conveyor, available in our school. In this paper, we describe the system and expose the technical solutions used to achieve an optimal solution. Two final solutions have been obtained, one of both is the operation of the line with a handling robot and the other is to operating the system in a simple way but with a data acquisition system in real time.

# ANNEXES

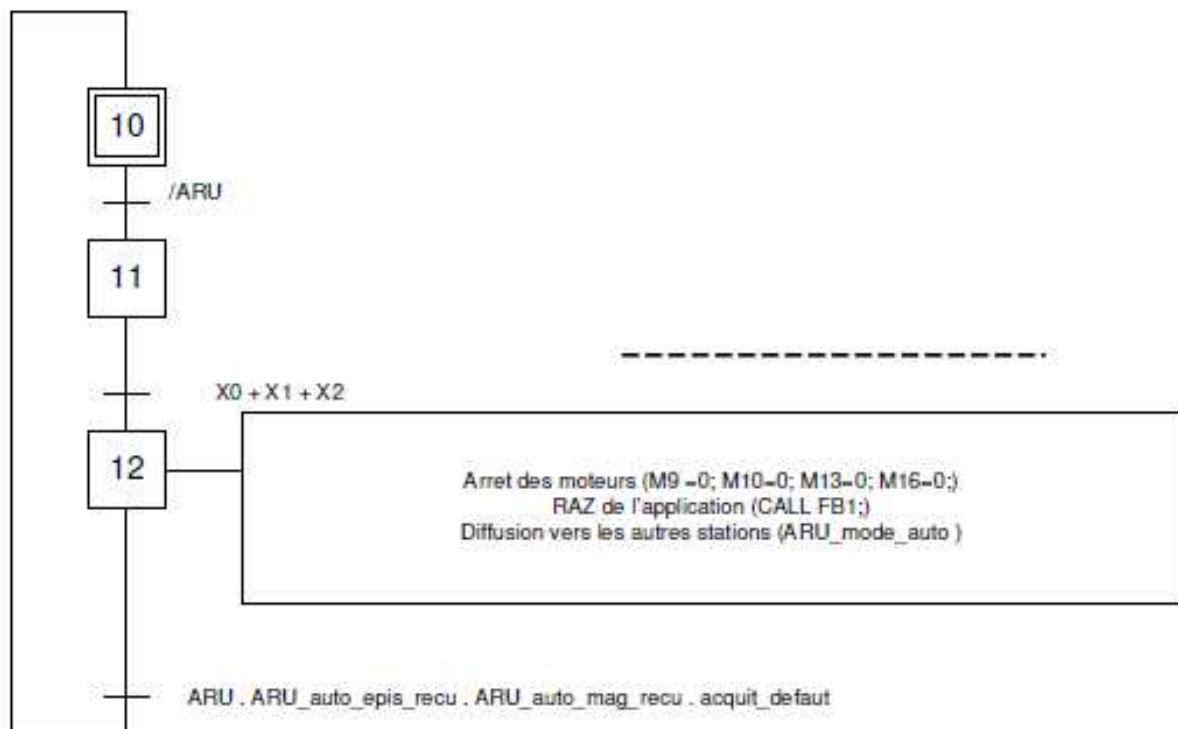
## Annexe 1. : GRAFCETS

### Annexe 1.1. : GRAFCETS Station 4

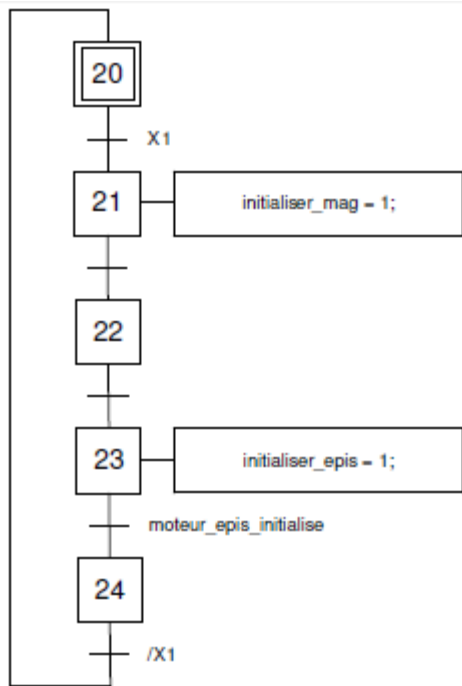
#### Annexe 1.1.1. : Mode de fonctionnement



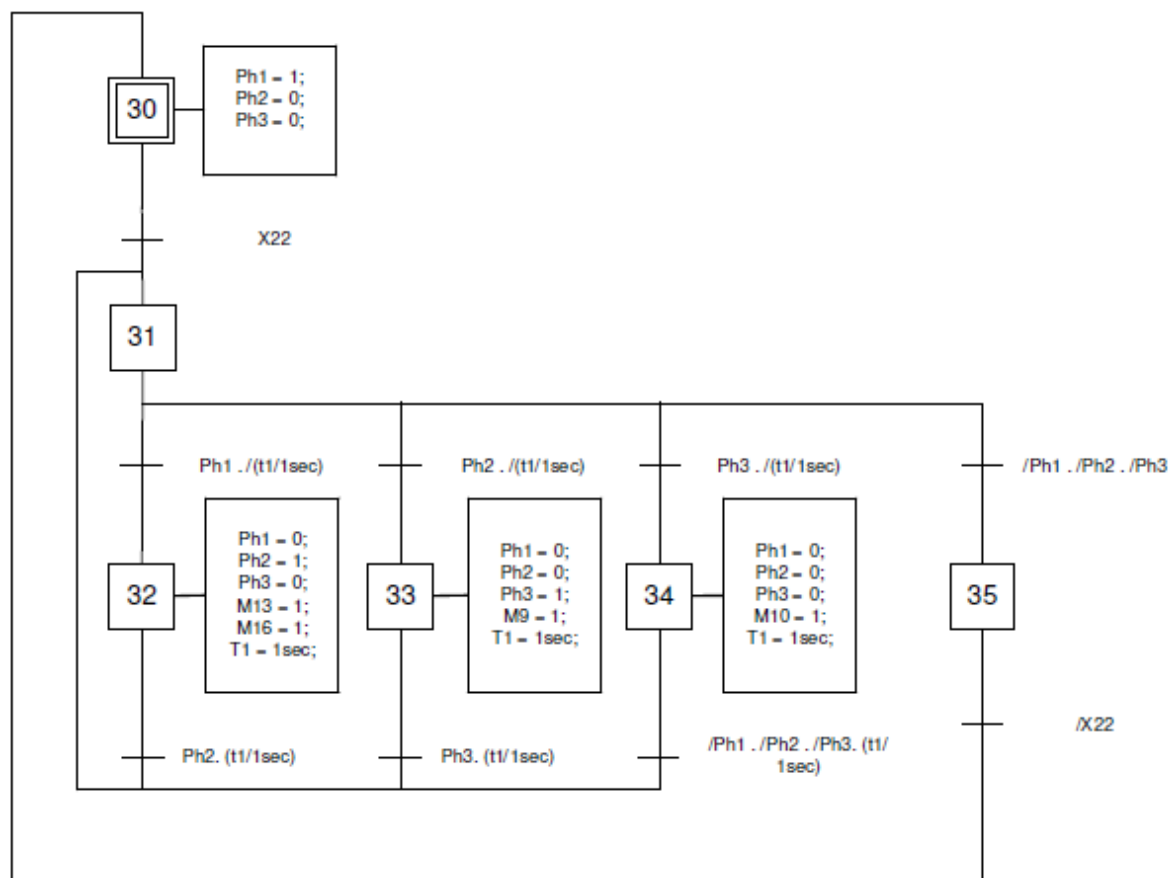
#### Annexe 1.1.2. : Gestion de l'ARU



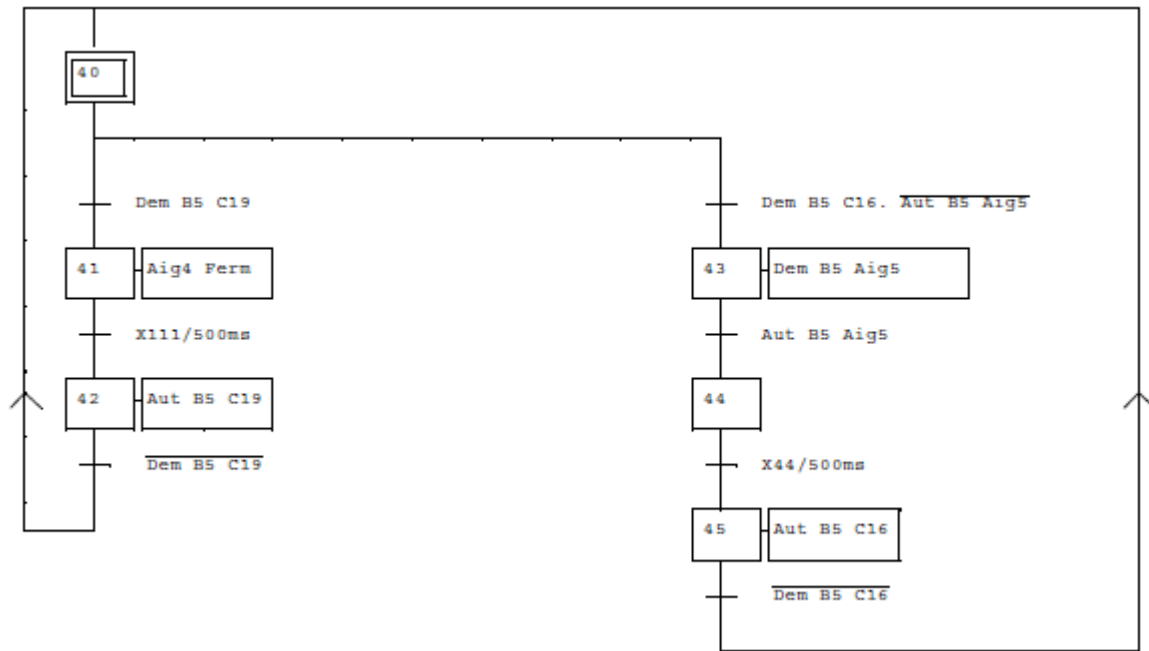
**Annexe 1.1.3. : Gestion de l'initialisation (communication entre les 3 stations)**



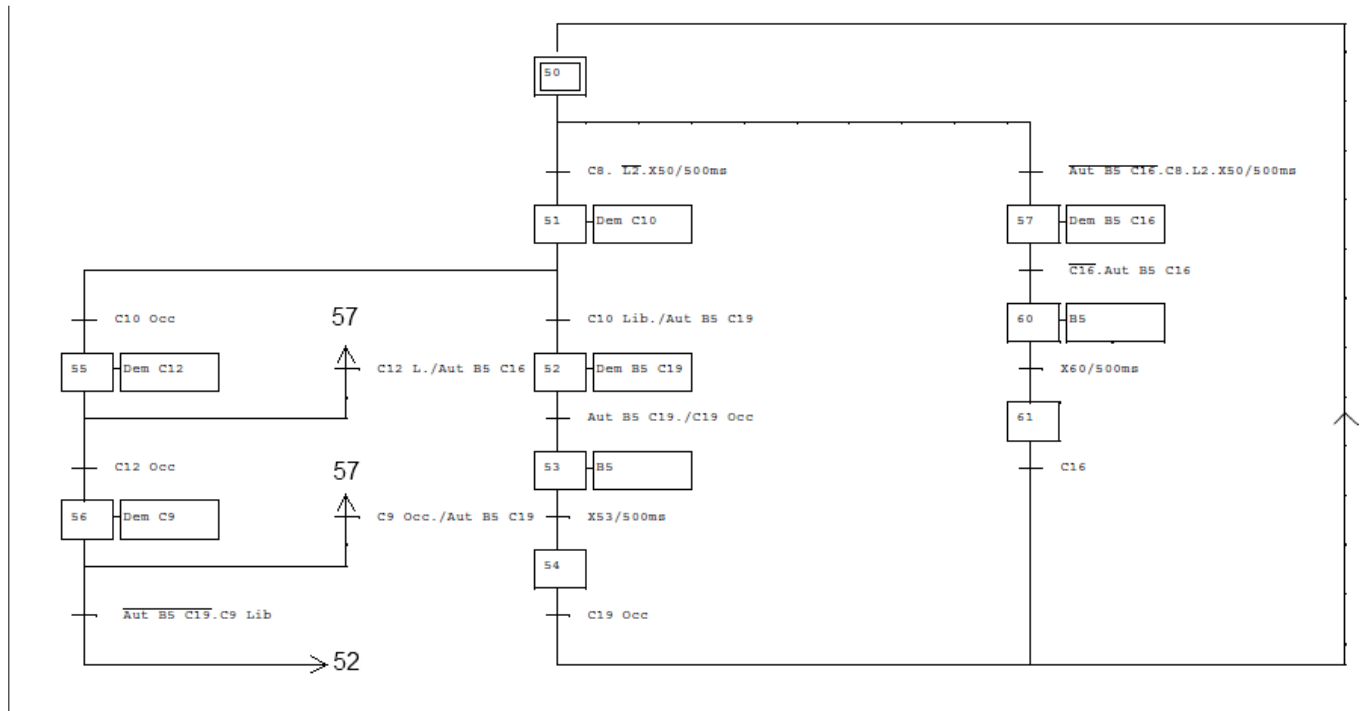
**Annexe 1.1.4. : Démarrage des moteurs de l'hippodrome**



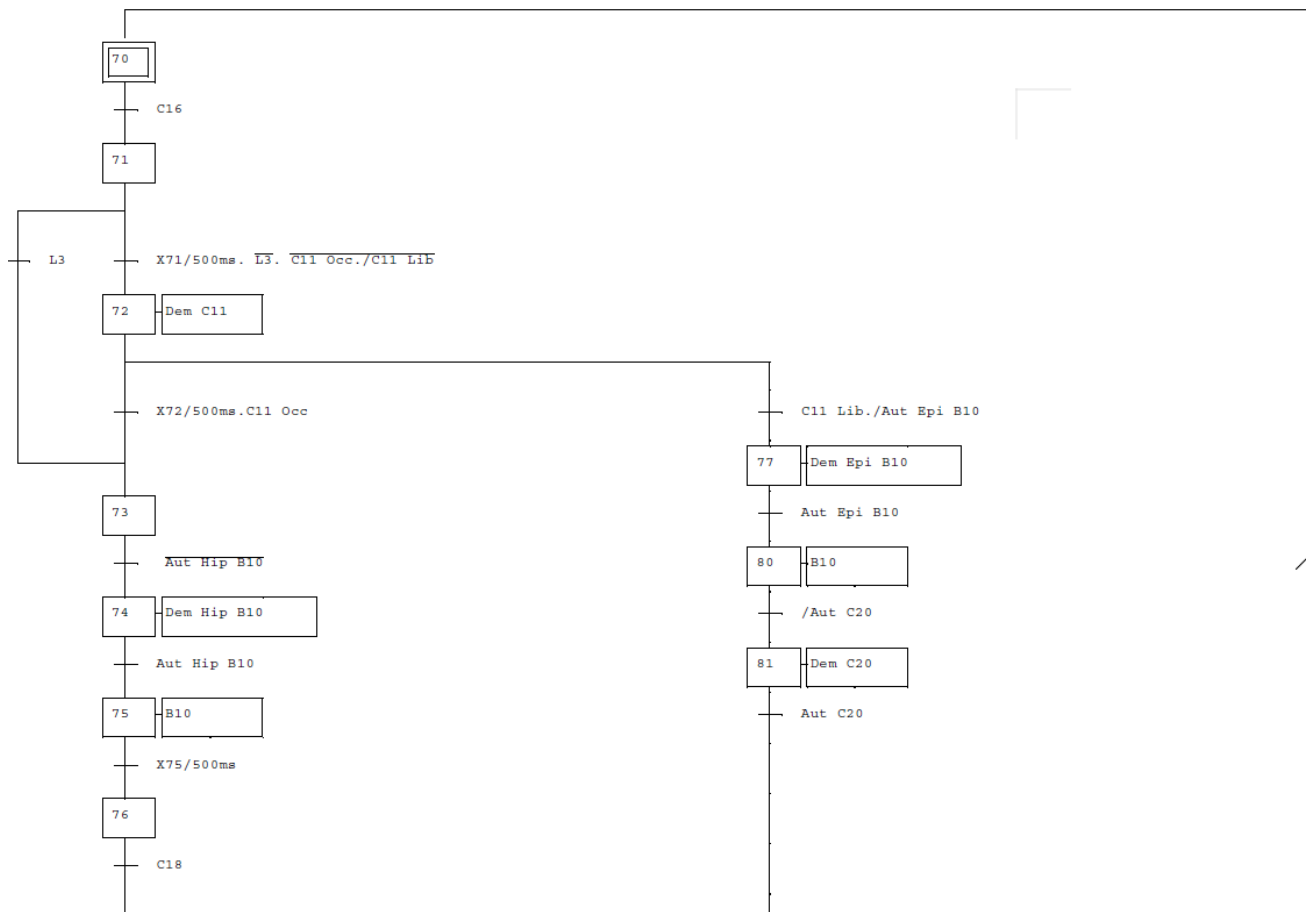
### Annexe 1.1.5. : Gestion de l'aiguillage 4



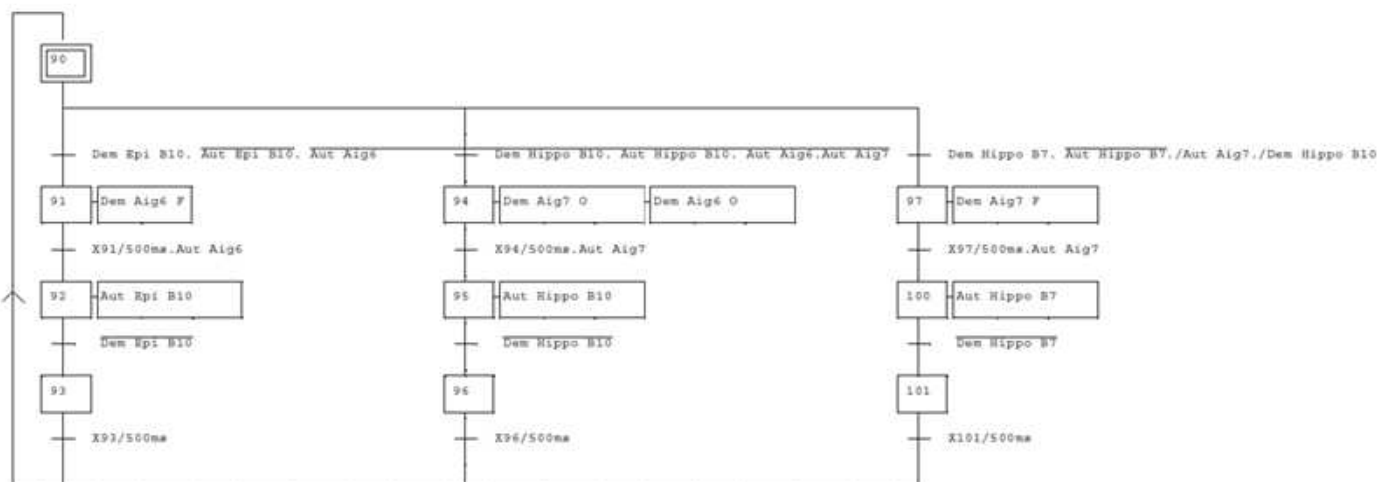
### Annexe 1.1.6. : Gestion de la butée 5



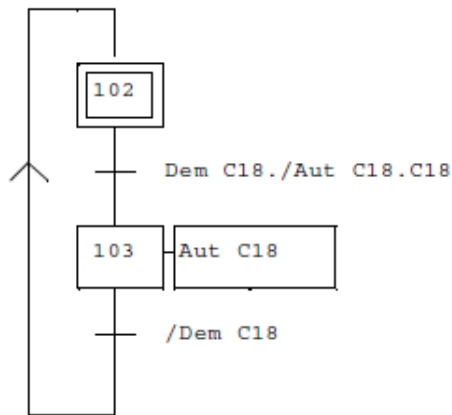
### Annexe 1.1.7. : Gestion de la butée B10



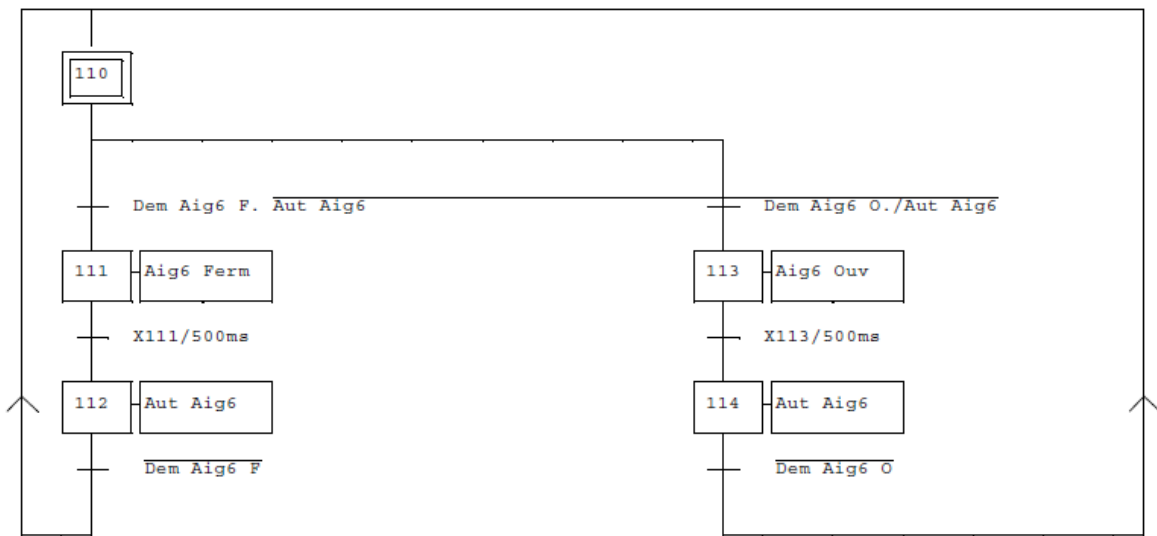
### Annexe 1.1.8. : Gestion des accès (B7 et B10)



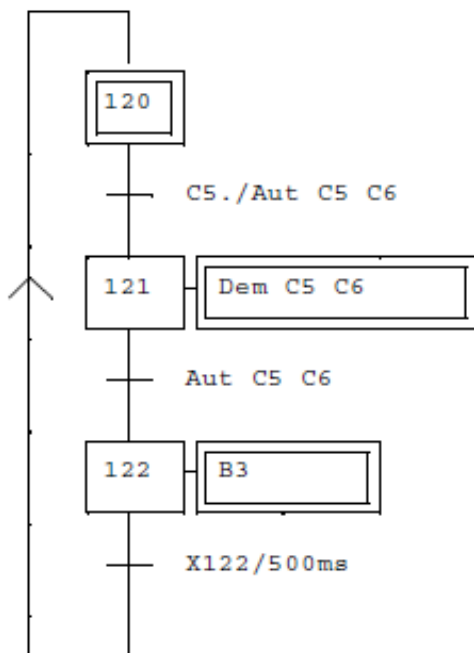
**Annexe 1.1.9. : Gestion du capteur C18**



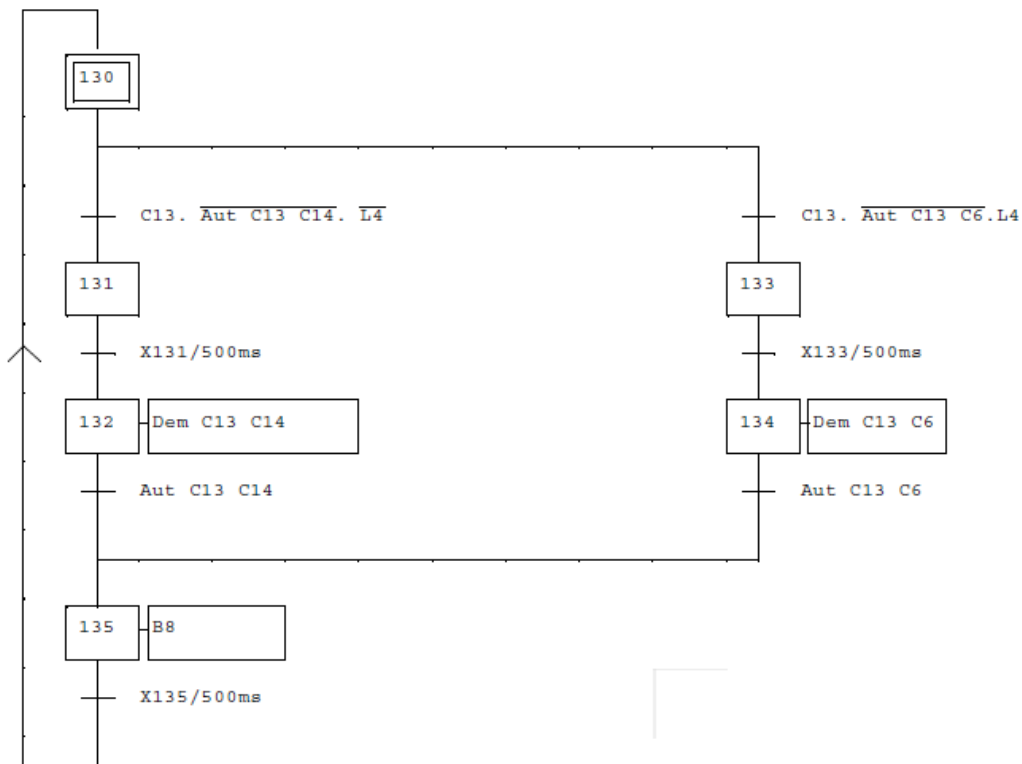
**Annexe 1.1.10. : Gestion de l'aiguillage 6**



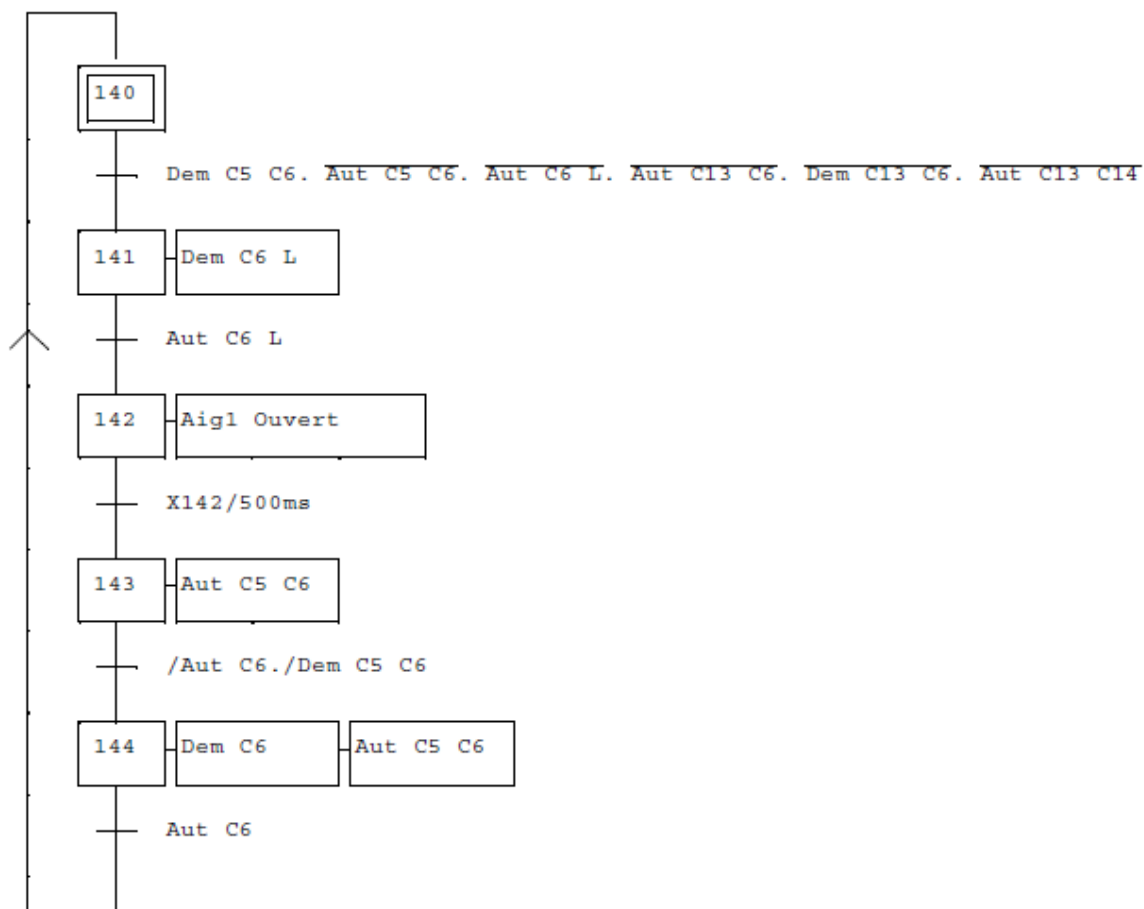
**Annexe 1.1.11. : Gestion de la butée B3**



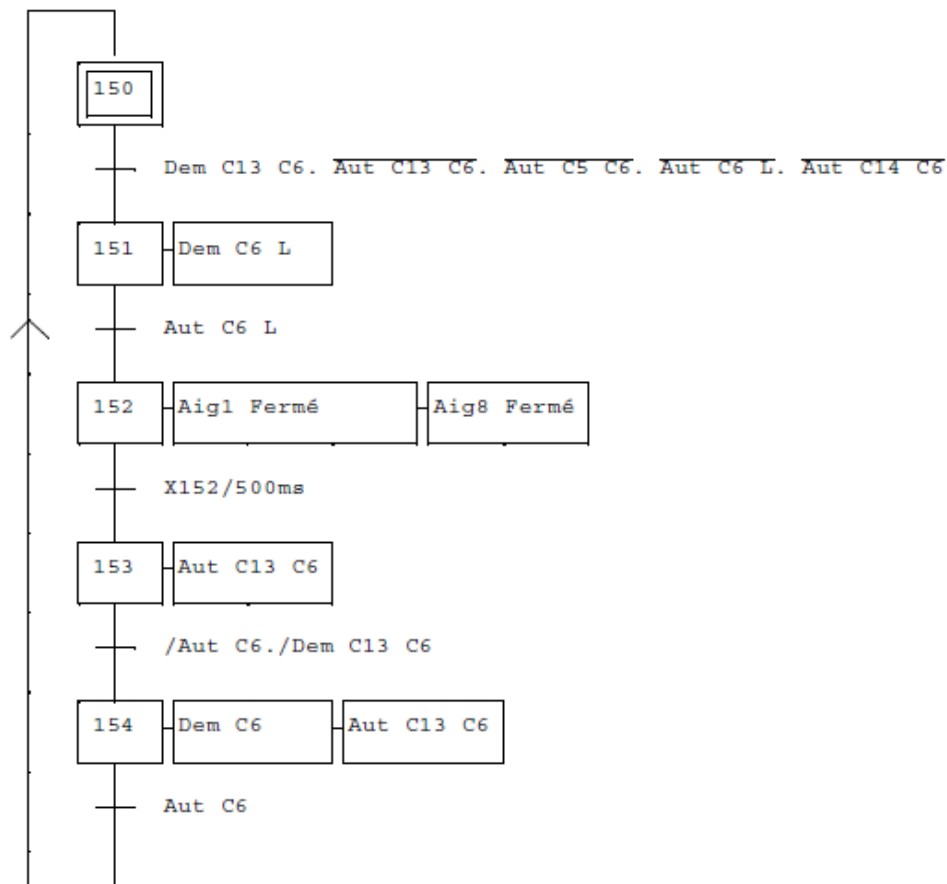
**Annexe 1.1.12. : Gestion de la butée B8**



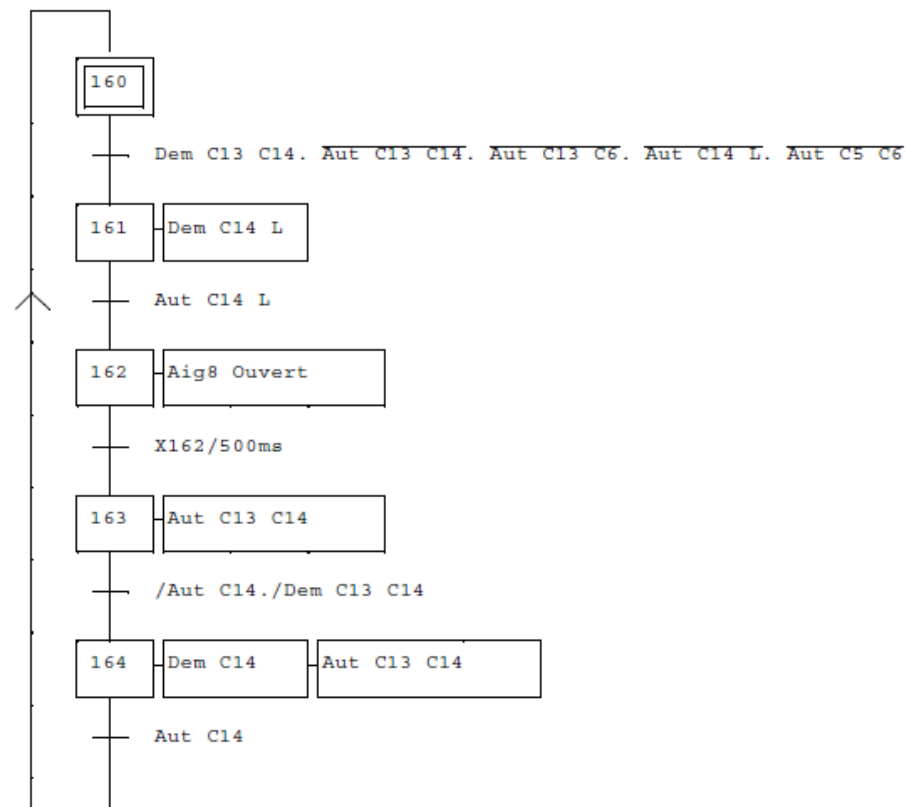
**Annexe 1.1.13. : Gestion accès C6 depuis C5**



**Annexe 1.1.14. : Gestion accès C6 depuis C13**



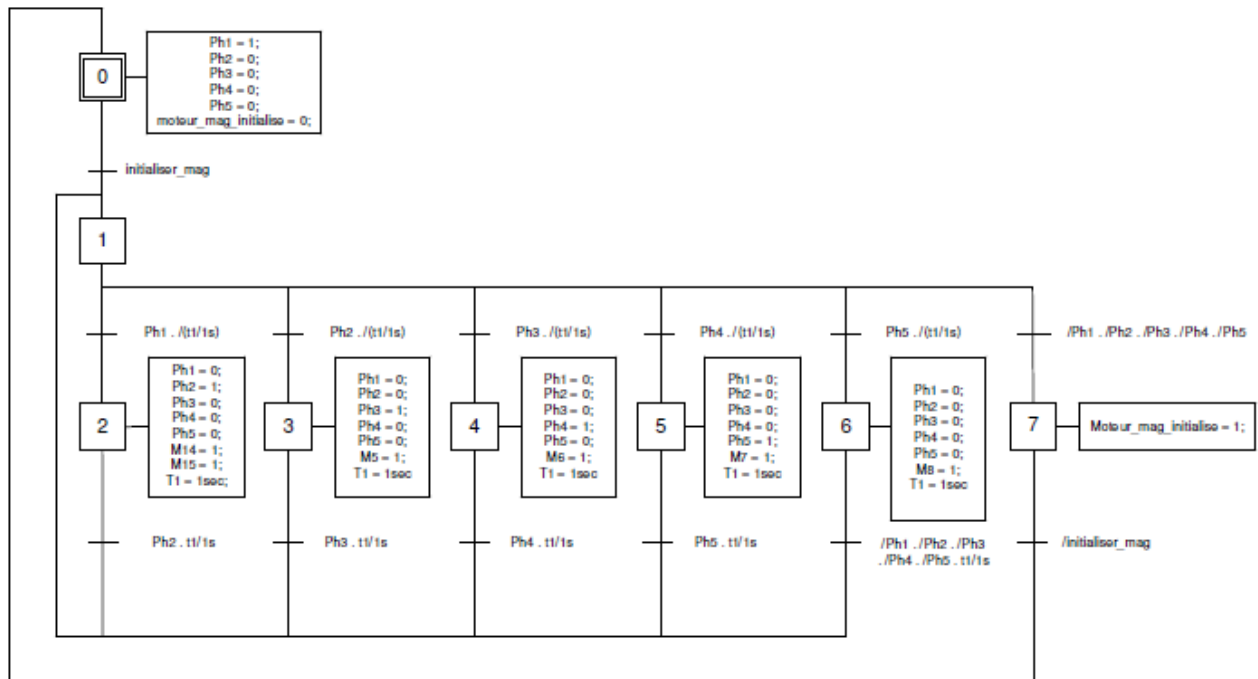
**Annexe 1.1.15. : Gestion accès C14 depuis C13**



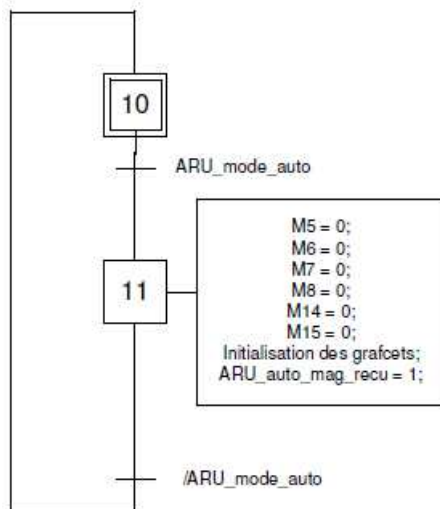


## Annexe 1.2. : GRAFCETS Station 5

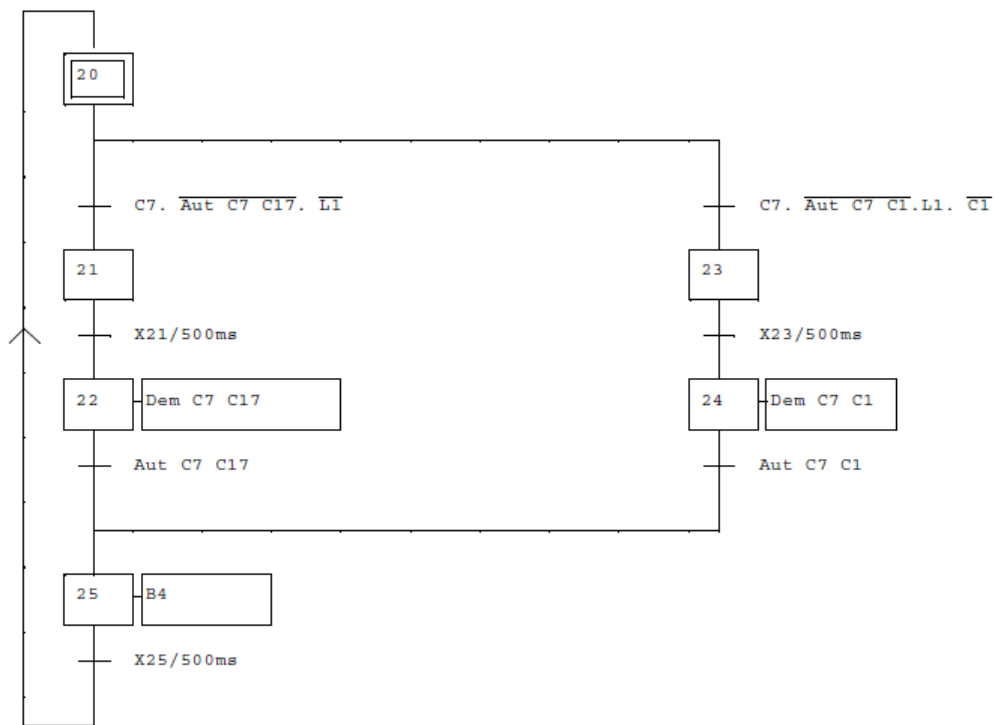
### Annexe 1.2.1. : Démarrage moteurs



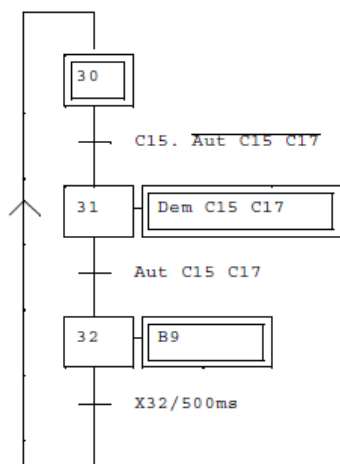
### Annexe 1.2.2. : Gestion ARU



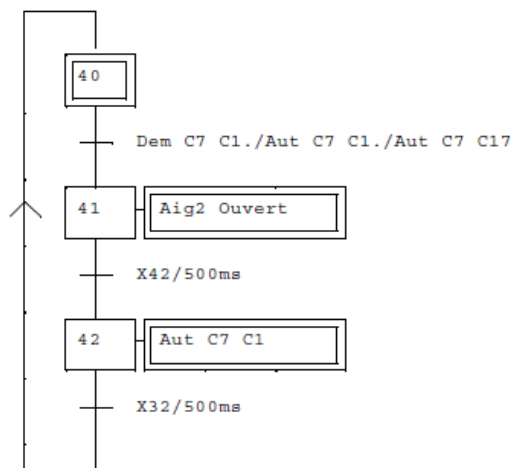
**Annexe 1.2.3. : Gestion de la butée B4**



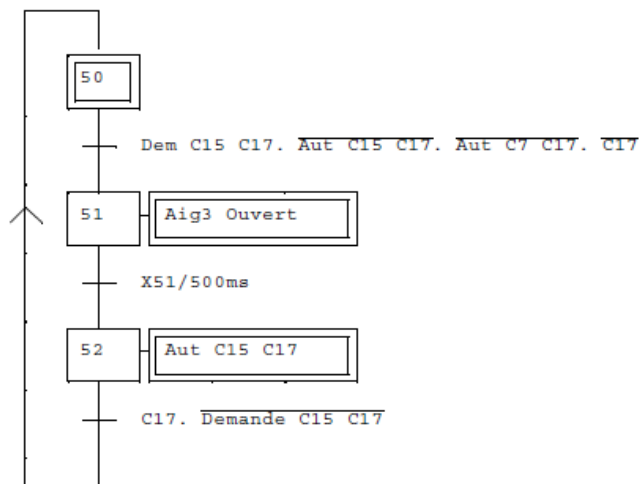
**Annexe 1.2.4. : Gestion de la butée B9**



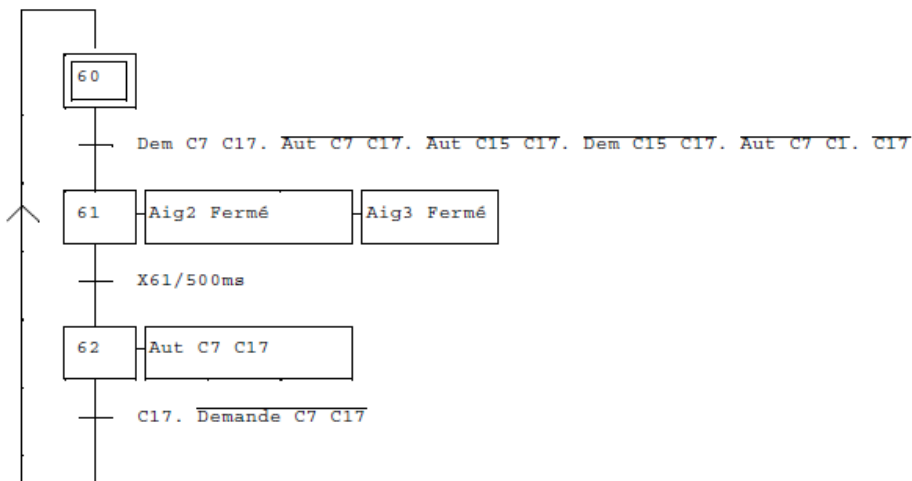
**Annexe 1.2.5. : Gestion accès C1 depuis C7**



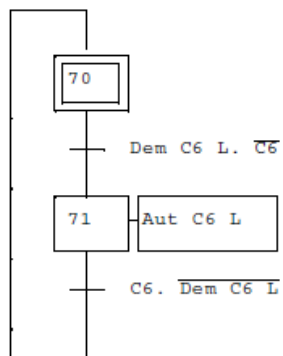
**Annexe 1.2.6. : Gestion accès C17 depuis C15**



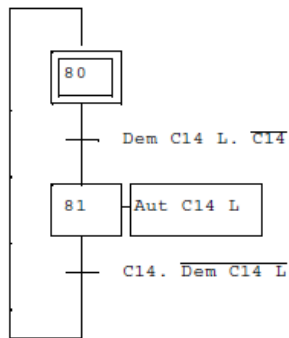
**Annexe 1.2.7. : Gestion accès C17 depuis C7**



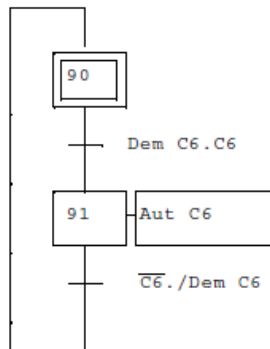
**Annexe 1.2.8. : Gestion du capteur C6 (Libre)**



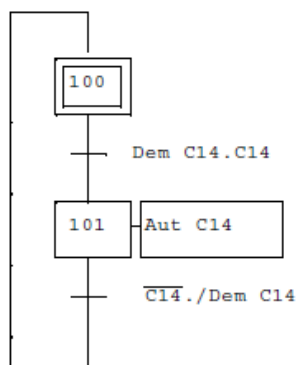
**Annexe 1.2.9. : Gestion du capteur C14 (Libre)**



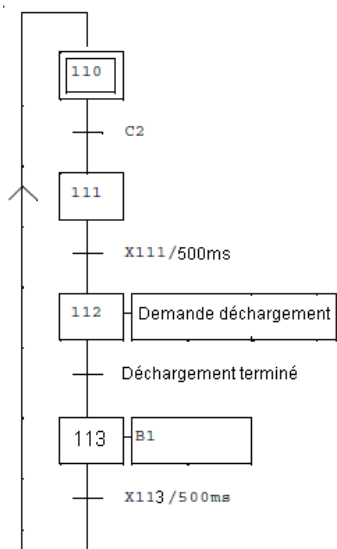
**Annexe 1.2.10. : Gestion du capteur C6(Occupé)**



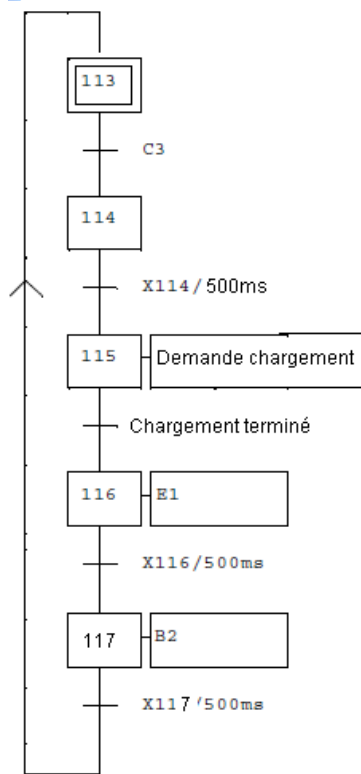
**Annexe 1.2.11. : Gestion du capteur C14(Occupé)**



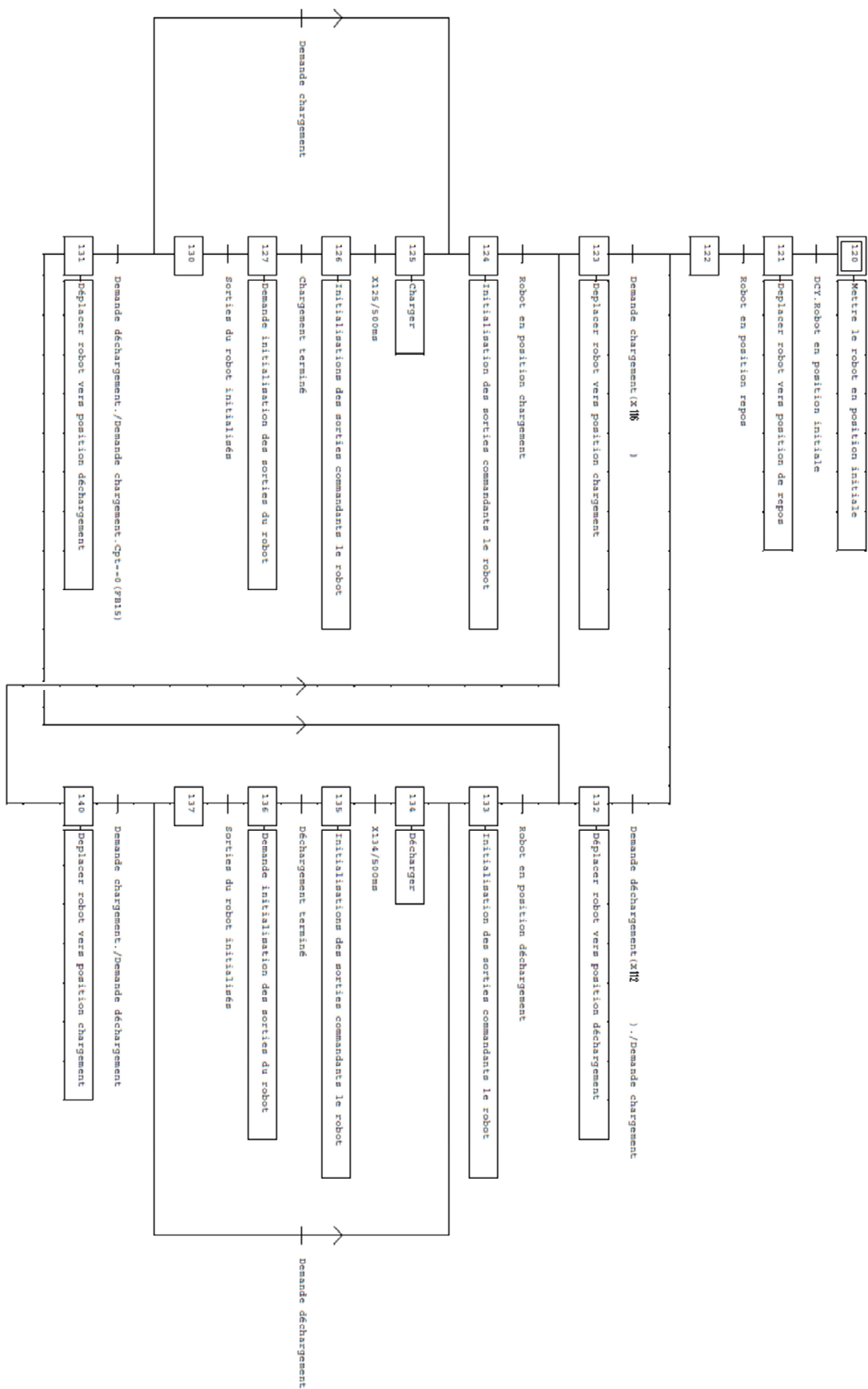
**Annexe 1.2.12. : Gestion de la butée B1**



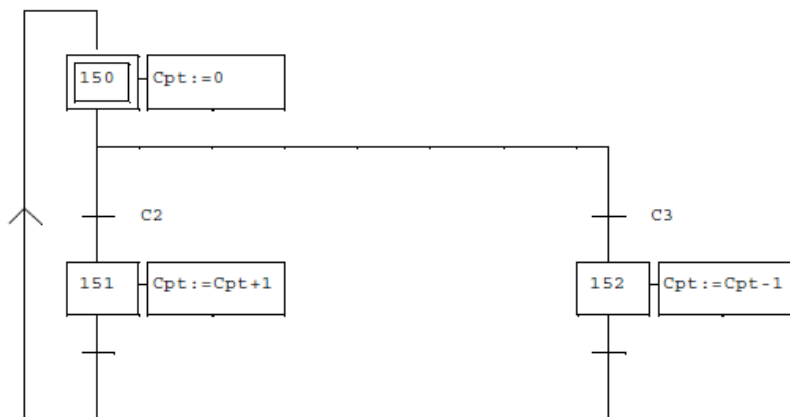
**Annexe 1.2.13. : Gestion de la butée B2**



## Annexe 1.2.14. : Gestion du robot

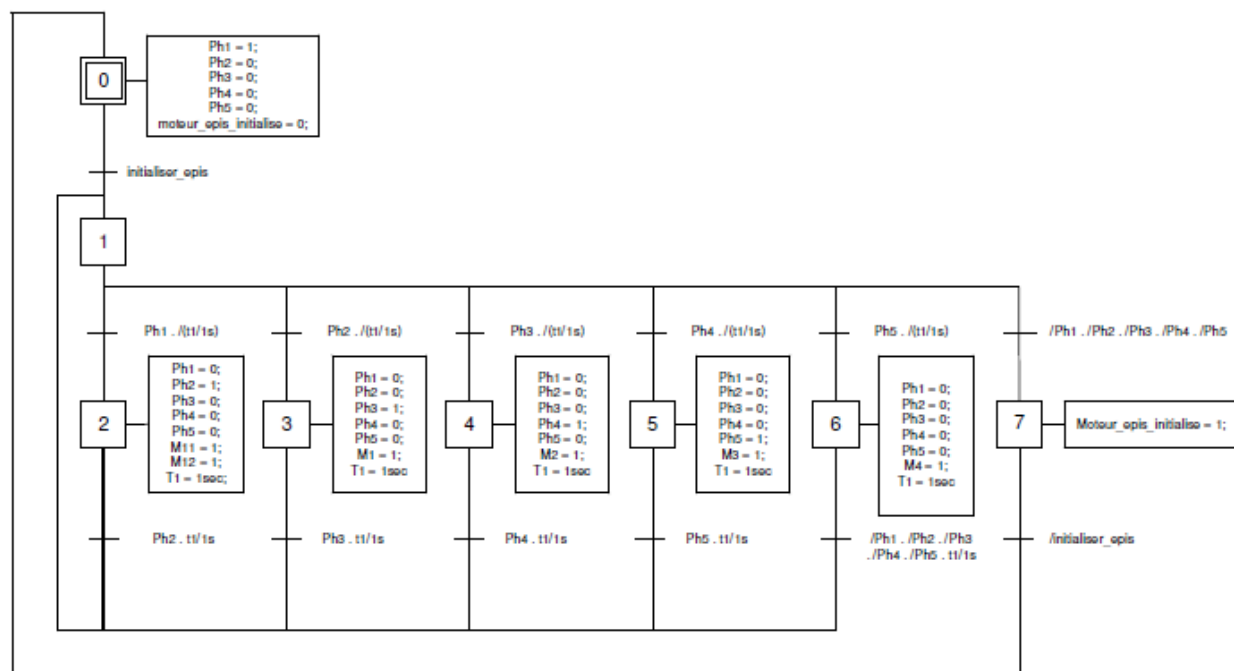


### Annexe 1.2.15. : Compteur de palettes vides

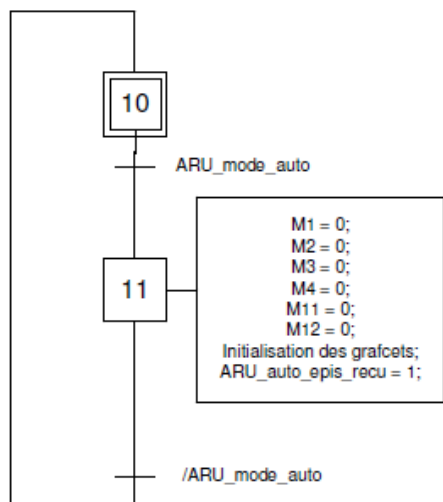


### Annexe 1.3. : GRAFCETS Station 6

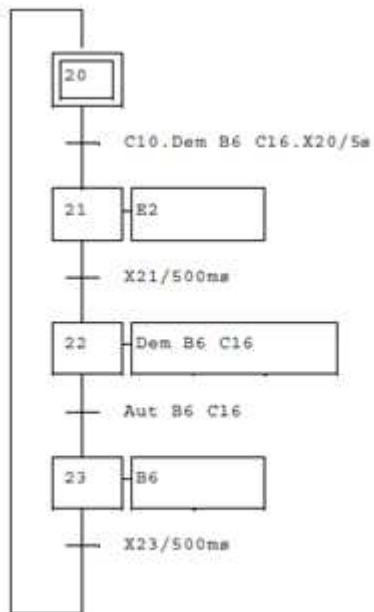
#### Annexe 1.3.1. : Démarrage moteurs



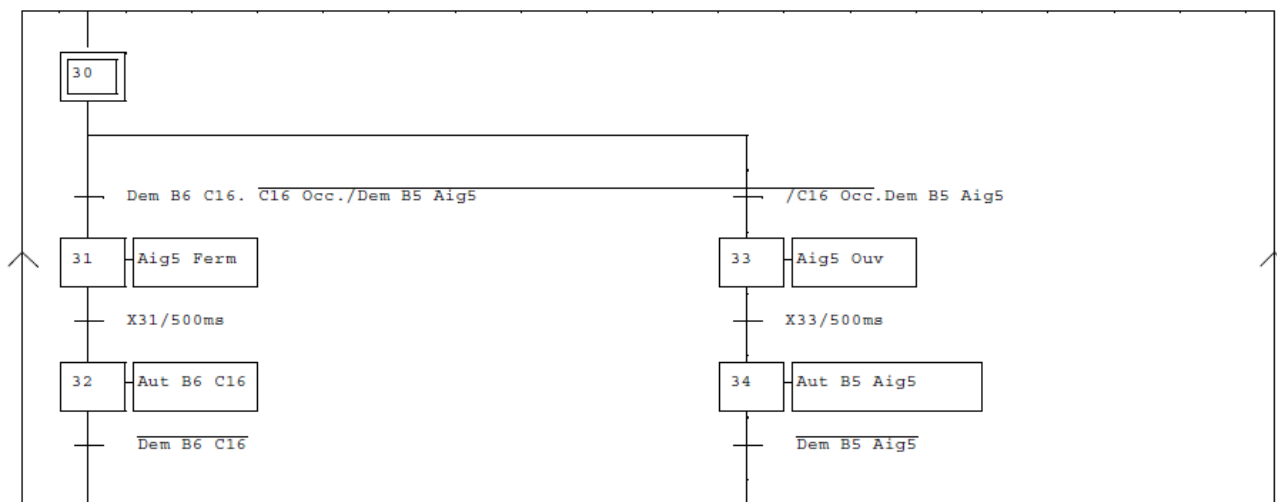
#### Annexe 1.3.2. : Gestion ARU



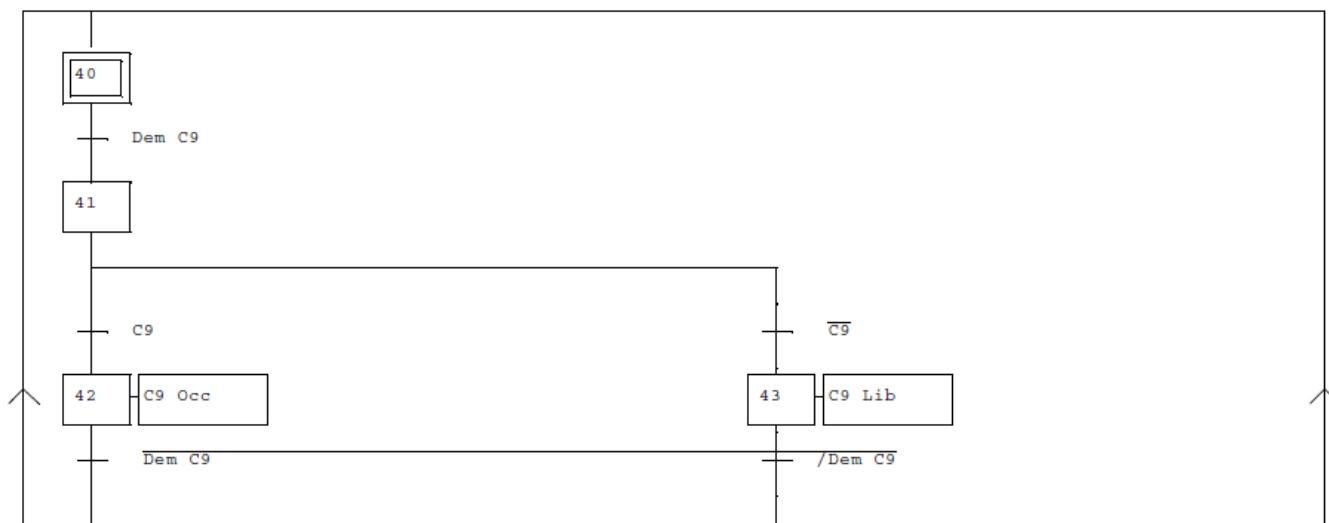
**Annexe 1.3.3. : Gestion de la butée B6**



**Annexe 1.3.4. : Gestion de l'aiguillage 5**

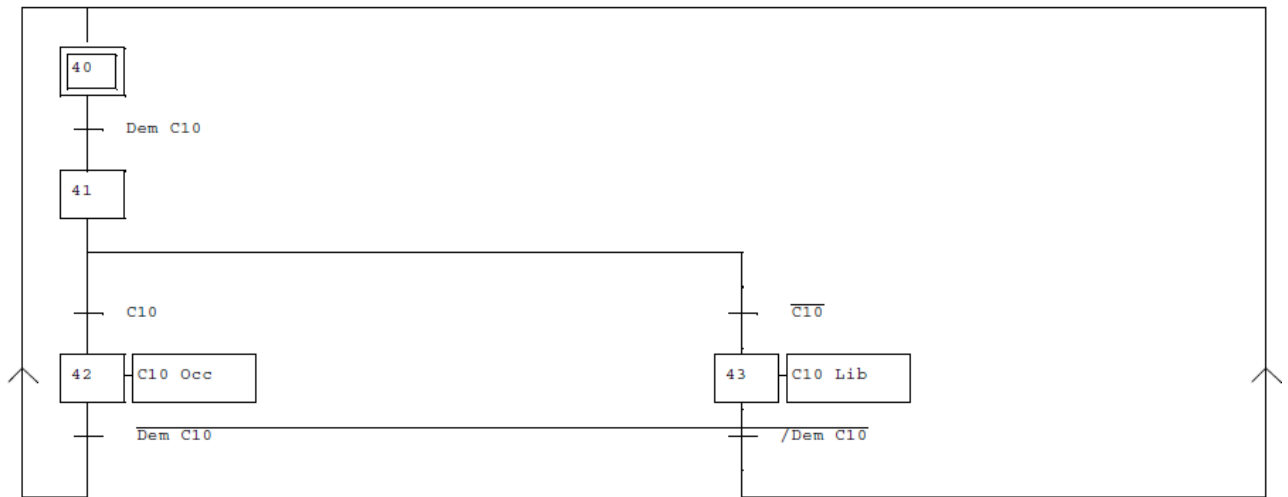


**Annexe 1.3.5. : Gestion du capteur C9**

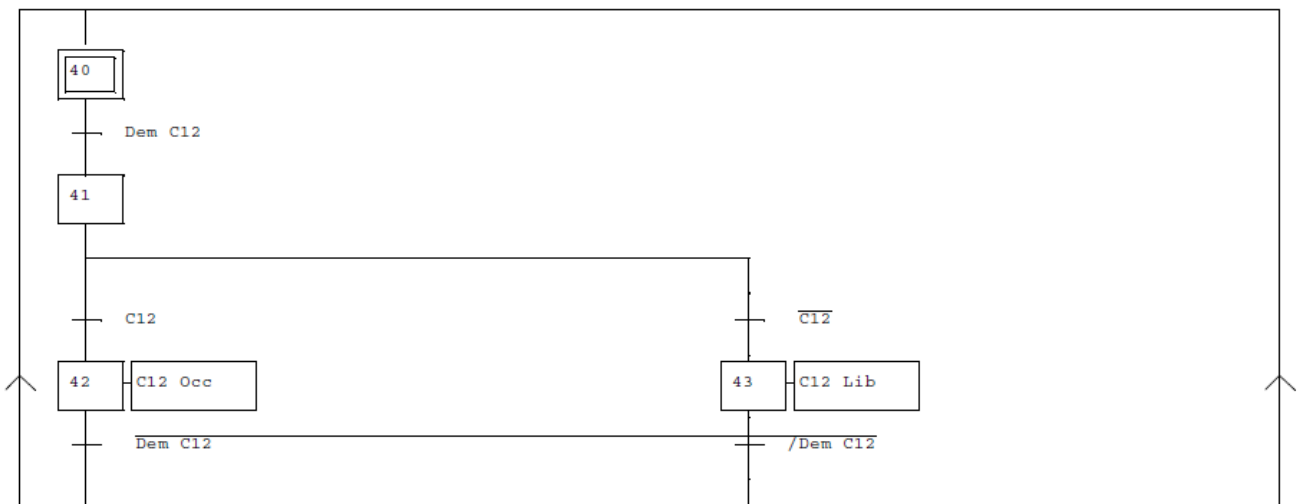




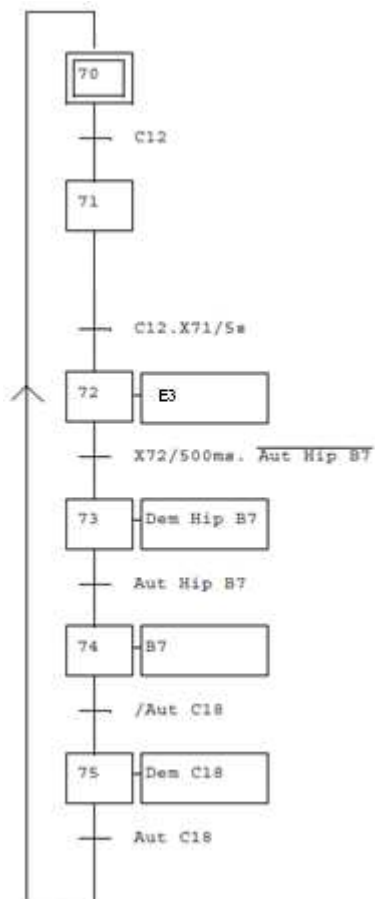
### Annexe 1.3.6. : Gestion du capteur C10



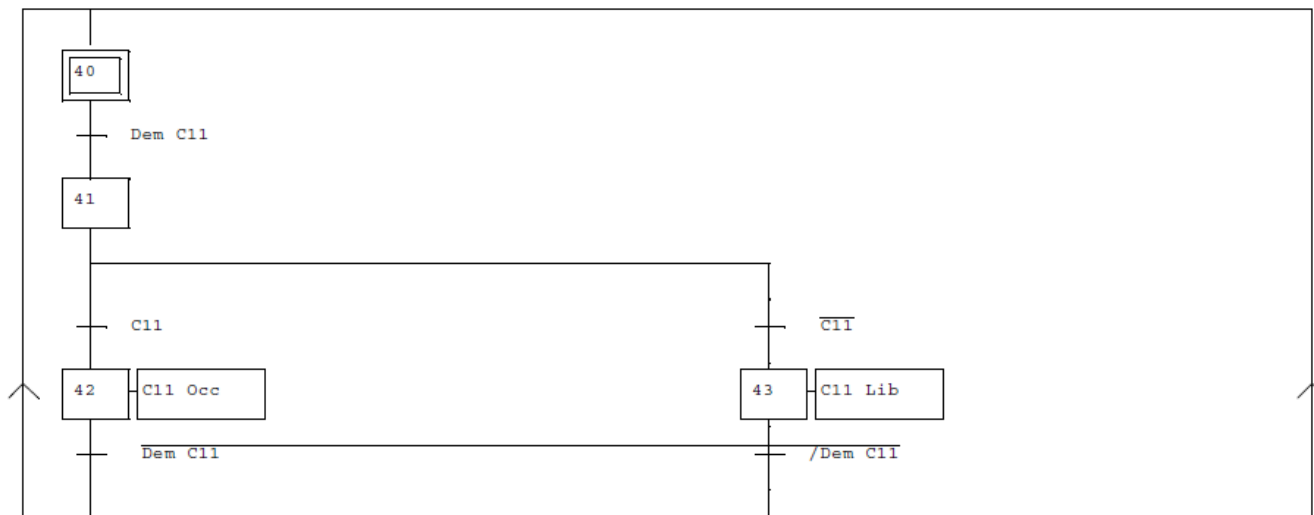
### Annexe 1.3.7. : Gestion du capteur C12



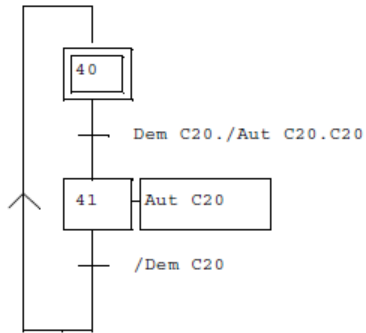
**Annexe 1.3.8. : Gestion de la butée B7**



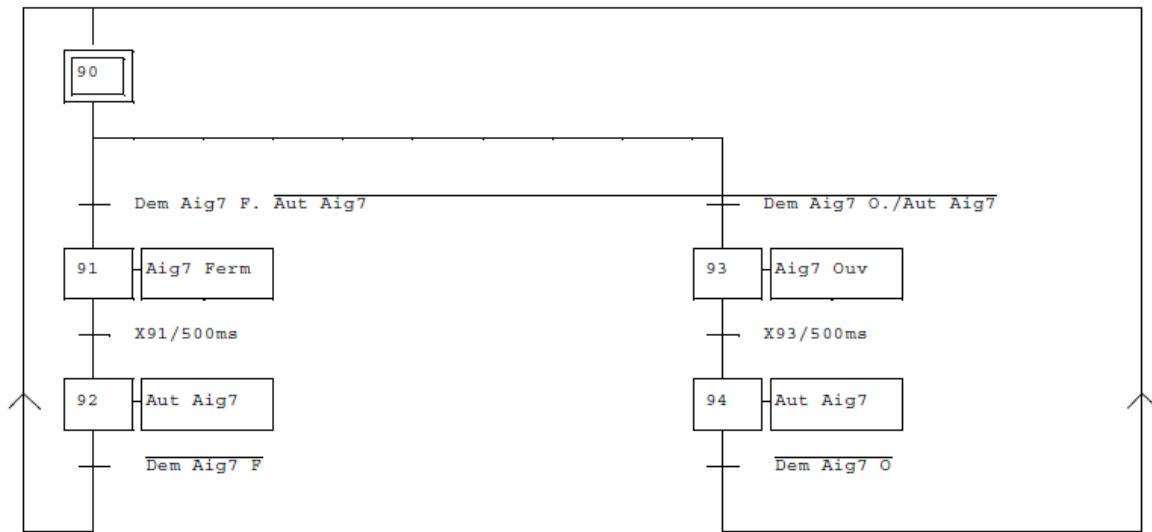
**Annexe 1.3.9. : Gestion du capteur C11**



**Annexe 1.3.10. : Gestion du capteur C20**



**Annexe 1.3.11. : Gestion de l'aiguillage 7**



## Annexe 2. : PROGRAMME ROBOT

### Annexe 2.1. : Programme principal

```
.PROGRAM boucle.pg()
  WHILE (1) DO
    CALL main.pg() //appel du programme main.pg
  END

.END
```

### Annexe 2.2. : Programme d'appels

```
.PROGRAM main.pg()
  WAIT SIG(1001,-1002,-1003,-1004) //attend demande mise en position initiale
  READY //mise en position initiale
  SIGNAL 1, -2, -3, -4 //renseigne que le robot est en position initiale
  WAIT SIG(-1001,1002,-1003,-1004) //attend demande mise en position repos
  CALL program1.pg() //appel du programme correspondant
  SIGNAL -1, 2, -3, -4 //renseigne que le robot est en position repos
  WHILE (SIG(1005)) DO //****tant qu'il n'y pas d'arret d'urgence***
    IF SIG(1001,1002,-1003,-1004) THEN //demande de mise en position chargement depuis la position repos
      CALL repchar.pg() //appel du programme correspondant
      SIGNAL 1, 2, -3, -4 //renseigne que le robot est en position chargement
    END
    IF SIG(-1001,-1002,1003,-1004) THEN //demande de chargement
      CALL charg.pg() //appel du programme correspondant
      SIGNAL -1, -2, 3, -4 //renseigne que le robot à fini le chargement
      WAIT SIG(1001,-1002,-1003,1004) //demande d'initialisation des sorties du robot
      SIGNAL -1, -2, -3, -4 //initialisation des sorties du robot
    END
    IF SIG(1001,-1002,1003,-1004) THEN //demande de mise en position d'@chargement depuis la position repos
      CALL repdec.pg() //appel du programme correspondant
      SIGNAL 1, -2, 3, -4 //renseigne que le robot est en position d'@chargement
    END
    IF SIG(-1001,1002,1003,-1004) THEN //demande de d'@chargement
      CALL dech.pg() //appel du programme correspondant
      SIGNAL -1, 2, 3, -4 //renseigne que le robot à fini le d'@chargement
      WAIT SIG(-1001,1002,-1003,1004) //demande d'initialisation des sorties du robot
      SIGNAL -1, -2, -3, -4 //initialisation des sorties du robot
    END
    IF SIG(1001,1002,1003,-1004) THEN //demande de mise en position d'@chargement depuis la position chargement
      CALL charde.pg() //appel du programme correspondant
      SIGNAL 1, -2, 3, -4 //renseigne que le robot est en position d'@chargement
    END
    IF SIG(-1001,-1002,-1003,1004) THEN //demande de mise en position chargement depuis la position d'@chargement
      CALL deccha.pg() //appel du programme correspondant
      SIGNAL 1, 2, -3, -4 //renseigne que le robot est en position chargement
    END
  END
  END //si arret d'urgence
  WAIT SIG(1005) //attendre d'@sactivation de l'arr@t d'urgence
  READY //mise en position initiale
.END
```

## Annexe 2.3. : Sous-programmes

```
.PROGRAM program1.pg() //programme de mise en position repos
SET #mov1 = #PPOINT(47,-90,90,0,0,0)
SET #mov2 = #PPOINT(47,-55,90,0,0,0)
SET #mov3 = #PPOINT(47,-55,170,0,0,0)
SET #mov4 = #PPOINT(47,-55,170,0,63,0)
MOVE #mov1 //rotation du pied
MOVE #mov2 //rotation de l'axe@paule
MOVE #mov3 //rotation du coude
MOVE #mov4 //rotation de l'avant bras

.END
*****
.PROGRAM repchar.pg() //programme de mise en position chargement depuis la position repos
SET #mov1 = #PPOINT(47,-55,121.801,0,63,0)
SET #mov2 = #PPOINT(-12.463,-55,121.801,0,63,0)
SET #mov3 = #PPOINT(-12.463,-55,121.801,0,63,-6.464)
SET #mov4 = #PPOINT(-12.463,-55,121.801,0,88.055,-6.464)
SET #mov5 = #PPOINT(-12.463,-27.33,121.801,0,88.055,-6.464)
OPENI
MOVE #mov1 //rotation du coude
MOVE #mov2 //rotation du pied
MOVE #mov3 //rotation du poignet
MOVE #mov4 //rotation de l'avant bras
MOVE #mov5 //rotation de l'axe@paule
BREAK

.END

.PROGRAM charg.pg() //programme de chargement
SET #mov0 = #PPOINT(-12.463,-27.33,121.801,0,88.055,-6.454)
SET #mov1 = #PPOINT(-12.463,-55,121.801,0,88.055,-6.454)
SET #mov2 = #PPOINT(-12.174,-55,121.801,0,88.055,-6.454)
SET #mov3 = #PPOINT(-12.174,-55,121.801,0,88.055,-13)
SET #mov4 = #PPOINT(-12.174,-55,121.801,0,63,-13)
SET #mov5 = #PPOINT(-12.174,-55,165,0,63,-13)
SET #mov6 = #PPOINT(-12.174,-45.262,165,0,63,-13)
CLOSEI //fermer la pince
MOVE #mov1 //rotation de l'axe@paule
MOVE #mov2 //rotation du pied
MOVE #mov3 //rotation du poignet
MOVE #mov4 //rotation de l'avant bras
MOVE #mov5 //rotation du coude
MOVE #mov6 //rotation de l'axe@paule
BREAK
OPENI //ouvrir pince
DELAY (2) //attendre 2 secondes
MOVE #mov5 //rotation de l'axe@paule
MOVE #mov4 //rotation du coude
MOVE #mov3 //rotation de l'avant bras
MOVE #mov2 //rotation du poignet
MOVE #mov1 //rotation du pied
MOVE #mov0 //rotation de l'axe@paule

.END
*****
.PROGRAM repdec.pg() //programme de mise en position d'axe@chargement depuis la position repos
SET #mov1 = #PPOINT(102.672,-55,170,0,63,0)
SET #mov2 = #PPOINT(102.672,-55,170,0,63,13.382)
SET #mov3 = #PPOINT(102.672,-55,170,0,81.502,13.382)
SET #mov4 = #PPOINT(102.672,-55,120.059,0,81.502,13.382)
OPENI
MOVE #mov1 //rotation du pied
MOVE #mov2 //rotation du poignet
MOVE #mov3 //rotation de l'avant bras
MOVE #mov4 //rotation du coude
BREAK

.END
```

```

*****
.PROGRAM dech.pg() //programme de d  chargement
SET #mov0 = #PPOINT(76,-48.054,164.695,0,61.877,-6.454)
SET #mov1 = #PPOINT(76,-55,164.695,0,61.877,-6.454)
SET #mov2 = #PPOINT(102.672,-55,164.695,0,61.877,-6.454)
SET #mov3 = #PPOINT(102.672,-55,164.695,0,61.877,13.382)
SET #mov4 = #PPOINT(102.672,-55,164.695,0,81.502,13.382)
SET #mov5 = #PPOINT(102.672,-55,120.059,0,81.502,13.382)
SET #mov6 = #PPOINT(102.672,-20.109,120.059,0,81.502,13.382)
OPENI //ouvrir pince
MOVE #mov6 //rotation de l'  paule
BREAK
DELAY (2) //attendre 2 secondes
CLOSEI //fermer pince
MOVE #mov5 //rotation de l'  paule
MOVE #mov4 //rotation du coude
MOVE #mov3 //rotation de l'avant bras
MOVE #mov2 //rotation du poignet
MOVE #mov1 //rotation du pied
MOVE #mov0 //rotation de l'  paule
BREAK
OPENI //ouvrir pince
DELAY (2) //attendre 2 secondes
MOVE #mov1 //rotation de l'  paule
MOVE #mov2 //rotation du pied
MOVE #mov3 //rotation du poignet
MOVE #mov4 //rotation de l'avant bras
MOVE #mov5 //rotation du coude
BREAK
OPENI //ouvrir pince

.END
*****
.PROGRAM charde.pg() //programme de mise en position d  chargement depuis la position chargement
SET #mov1 = #PPOINT(-12.463,-55,121.801,0,88.055,-6.454)
SET #mov2 = #PPOINT(-12.463,-55,121.801,0,88.055,13.382)
SET #mov3 = #PPOINT(-12.463,-55,121.801,0,81.502,13.382)
SET #mov4 = #PPOINT(102.672,-55,121.801,0,81.502,13.382)
SET #mov5 = #PPOINT(102.672,-55,120.059,0,81.502,13.382)

MOVE #mov1 //rotation de l'  paule
MOVE #mov2 //rotation du poignet
MOVE #mov3 //rotation de l'avant bras
MOVE #mov4 //rotation du pied
MOVE #mov5 //rotation du coude
BREAK

.END
*****
.PROGRAM deccha.pg() //programme de mise en position chargement depuis la position d  chargement
SET #mov1 = #PPOINT(102.672,-55,120.059,0,88.055,-6.454)
SET #mov2 = #PPOINT(-12.463,-55,120.059,0,88.055,-6.454)
SET #mov3 = #PPOINT(-12.463,-55,121.801,0,88.055,-6.454)
SET #mov4 = #PPOINT(-12.463,-27.33,121.801,0,88.055,-6.454)
MOVE #mov1 //rotation du poignet
MOVE #mov2 //rotation du pied
MOVE #mov3 //rotation du coude
MOVE #mov4 //rotation de l'  paule
BREAK

.END

```

### Annexe 3. : Diagramme GANTT

