Rapport de Projet 4ème année AGI

Robotisation de la ligne transitique

BOURGET Simon | JAFFRE Cindy

Encadrant tuteur : Mr HARDOUIN Laurent

2015 - 2016

REMERCIEMENTS

Tout d'abord, nous exprimons notre reconnaissance au Directeur des Etudes AGI-4ème année, Mr LHOMMEAU Medhi, et à l'ISTIA pour les salles misent à notre disposition ainsi que le matériel qui nous a permis de réaliser notre projet.

Nous remercions particulièrement Mr HARDOUIN Laurent, notre tuteur de projet, pour sa disponibilité, ses conseils et son aide tout au long de ce projet. Sans qui, ce projet n'aurait pu être réalisé.

Nous tenons également à dire merci à Mr GUYONNEAU Rémi, pour son encadrement au cours des Travaux pratiques d'Automatisme, qui nous a permis de constituer le pilier de notre projet.

Nous voulons adresser nos remerciements au corps enseignant de 4ème année Automatique et Informatique Industriel de l'ISTIA école d'ingénieur de l'université d'Angers, pour leur enseignement.

Pour finir, nous tenons à adresser nos sincères remerciements à toutes les personnes ayant contribué de près ou de loin, à l'élaboration de ce projet.

SOMMAIRE

LISTE DES FIGURES

GLOSSAIRE

API[1] : Automate Programmable Industriel, système capable de commander un processus.

Aiguillage[2] : pièce métallique pivotant grâce à un système pneumatique permettant de diriger les palettes entre deux voies.

BDD[3] : Base De Données, dispositif permettant de stocker des données brutes dans des tables.

Bit[4] : unité de numération pouvant prendre que deux valeurs (0 ou 1)

Butée[5] : dispositif qui arrête les palettes et qui se baisse par un système pneumatique pour les laisser passer

GRAFCET[6] : GRAphe Fonctionnel de Commande des Etapes et Transitions, méthode de représentation d'un automatisme grâce à un système d'étapes et de transitions

Interface web[7] : interface homme - machine constituée de pages web

LIST[8] : langage de programmation automate proche du langage assembleur

Octet[9] : système de codage, constitué 8 bits

PHP[10] : Hypertext Preprocessor, langage de programmation libre principalement utilisé pour les pages web

Profibus[11] : Process Field Bus, bus de terrain et également protocole associé pour la communication entre automates

Requête[12] : élément de base des langages de requêtes utilisant des mots clés (SELECT, UPDATE…)

Sémaphore[13] : méthode de communication entre deux appareils

SQL[14] :Structured Query Language, langage informatique permettant d'exploiter les base de données par le biais de requêtes

Supervision[15] : suivi et pilotage à distance de processus

INTRODUCTION

Au cours de notre 4ème année d'ingénieur Génie des Systèmes Industriels à l'ISTIA école d'ingénieur de l'université d'Angers, nous avons réalisé un projet du 2 décembre 2015 au 27 avril 2016. Ce projet avait pour but de réaliser l'automatisme et la supervision[(15)] d'une ligne transitique en interaction avec un robot.

En outre, le développement de la programmation automate, la programmation du robot a également été développé. De plus, la ligne transitique est constituée de trois Automates Programmables Industriels (API), ainsi la communication réseau entre ces derniers fut indispensable. Une interface Web[(7)] a aussi été mise en place pour fournir un accès à distance depuis Internet.

Dans un premier temps, nous présenterons la maquette du projet. Puis, nous détaillerons la programmation des automates. Ensuite, nous décrirons la supervision. Pour finir, nous étudierons la programmation du robot.

1 Présentation du projet

Ce projet a été réalisé en binôme sur une maquette présente au sein de l'ISTIA. Avant de débuter le projet, nous avons défini - avec notre tuteur - les besoins et le travail à réaliser pour le mener à bien. Cette première phase est importante, elle va permettre de décomposer les tâches à effectuer par la suite et d'organiser le projet.

1.1 Cahier des charges

1.1.1 Présentation de la maquette

La maquette est constituée d'une ligne transitique et d'un robot.

La ligne comporte trois parties. Chacune possède des capteurs (permettant de savoir si une palette est présente et/ou si elle a été usinée), des butées(5) et des aiguillages(2) pneumatiques (permettant de retenir et guider les palettes) afin qu'elles circulent entre les différentes parties de la maquette.

La première partie, nommée « magasin », charge les pièces pour être usinées puis déchargées. Le robot est localisé sur cette partie. La deuxième partie, l' « hippodrome » est une zone de transfert, afin que les pièces soient usinées. La troisième partie, les « épis » permet de simuler l'usinage des pièces.

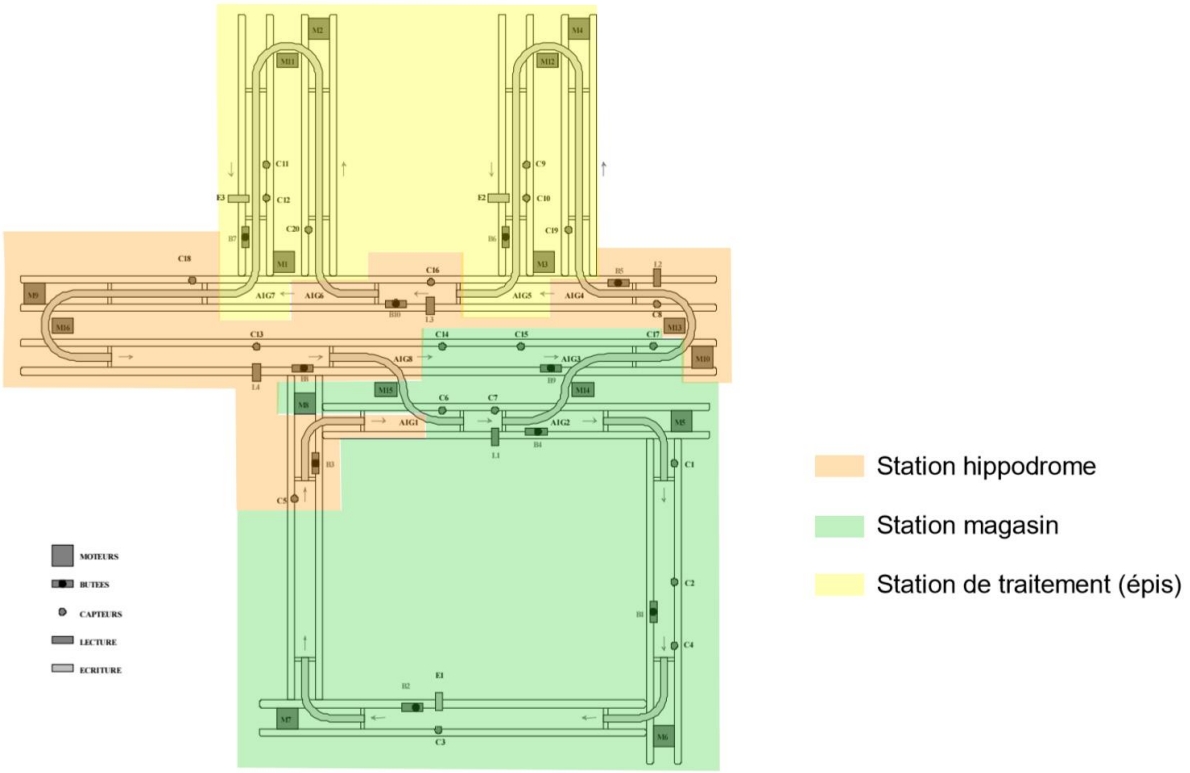


Figure 1 : Schéma de la maquette de la ligne transitique

1.1.2 Présentation des besoins

Ce projet a pour but de superviser et d'automatiser le système de transport de palettes et de programmer le robot Staubli afin qu'il interagisse avec le système.

Un système client/serveur sera chargé de collecter les données du système, ce superviseur renseignera une base de données (BDD) mysql.

Les données collectées seront mises à disposition d'applications légères permettant de suivre en temps réel l'état du système depuis internet, client Java. Ce client enverra également les ordres de production au superviseur. Le client pourra également être sur un système mobile, développement sous android.

Le superviseur sera également chargé d'interagir avec le système afin de réaliser un pilotage en juste à temps du système, c'est à dire que l'accumulation de palettes devra être limitée au maximum.

1.2 Architecture du projet

1.2.1 Architecture réseau

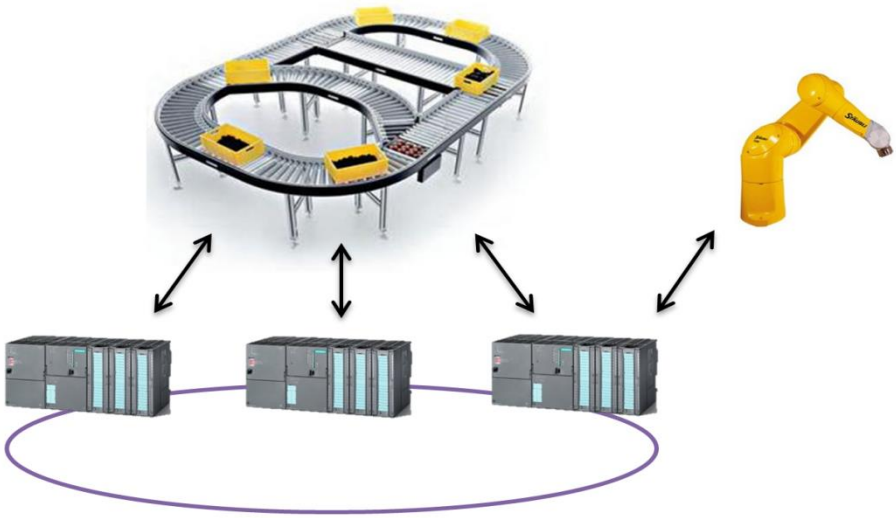


Figure 2 : Architecture réseau de la ligne transitique

Pour pouvoir interagir avec le milieu industriel, le robot doit être connecté à un API(1) (celui de l'hippodrome). Chaque API(1) gère une partie de la ligne *(voir paragraphe 1.1.1. Présentation de la maquette)*.

Les entrées/sorties correspondants aux capteurs, butées(5), aiguillages(2) et moteurs du convoyeur. Ils sont câblés directement sur les API(1) en entrées/sorties.

Les automates communiquent entre eux via le réseau Profibus(11)/DP suivant un système jeton sur anneau.

Les entrées/sorties du robot est câblé directement sur l'API(1) de l'hippodrome.

1.2.2 Architecture de la supervision

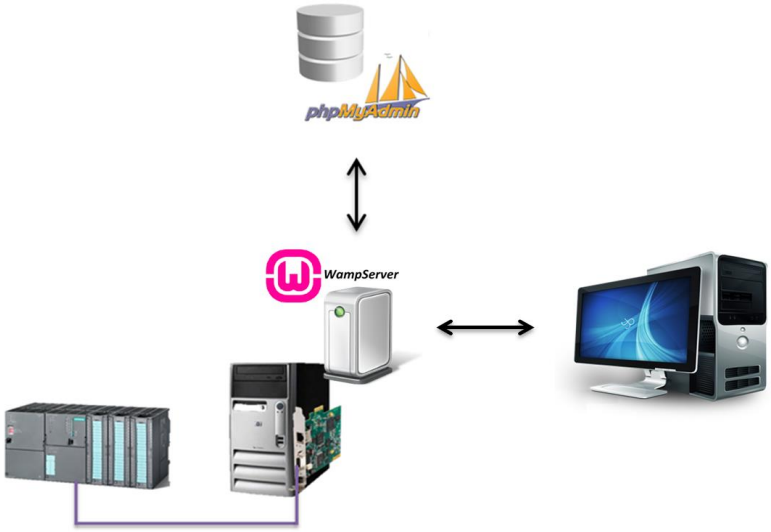


Figure 3 : Architecture de la supervision

L'API(1) communique avec un PC-serveur via une carte applicom (installé sur ce PC-serveur). Les échanges utilisent le protocole de communication Profibus(11) DP – MPI.

Le PC-serveur fait la communication entre la BDD(3) et les API(1).

La BDD(3) est gérée par l'application Web phpMyAdmin. De plus, l'application dispose d'une interface(7) pour gérer les bases de données. La communication avec cette BDD(3) se fait via des requêtes(12) SQL(14).

La plateforme de développement WampServer, installée sur le PC-serveur, permet de faire la communication entre l'utilisateur et la BDD(3). Cette communication est basée sur une architecture client/serveur.

1.3 Gestion du projet

1.3.1 Planning prévisionnel

Après avoir établi le cahier des charges, nous avons réfléchi à un planning comportant les tâches à réaliser. Le plus difficile étant d'attribuer une durée pour chacune des tâches sans dépasser le nombre de séances dont nous disposions.

Nous avons donc réalisé un planning prévisionnel qui comporte les différentes parties du projet (programmation automate, supervision…). Chacune de ces parties décrit les tâches à réaliser avec leur durée, la date de début et la date de fin.

Nom de la tâche	Durée	Début	Fin
Programmation automate	**45 jours**	**Mer 02/12/15**	**Mar 02/02/16**
Programmation du Magasin	6 sm	Mer 02/12/15	Mar 08/12/15
Programmation de l'automate de l'hyppodrome	1 sm	Mer 09/12/15	Mar 15/12/15
Programmation de l'automate des épis	1 sm	Mer 09/12/15	Mar 22/12/15
Programmation de la sécurité	2 sm	Mar 19/01/16	Lun 01/02/16
Programmation des calculs des temps de virage	2 sm	Mer 20/01/16	Mar 02/02/16
Supervision	**45 jours**	**Mar 02/02/16**	**Lun 04/04/16**
Supervision des automates	1 sm	Mar 02/02/16	Lun 08/02/16
Supervision des défauts	1 sm	Mar 29/03/16	Lun 04/04/16
Robotisation	**50 jours**	**Mar 02/02/16**	**Lun 11/04/16**
Manipulation du bras de robot	1 sm	Mar 02/02/16	Lun 08/02/16
Mise en position initiale du robot	1 sm	Mar 09/02/16	Lun 15/02/16
Programmation du robot	3 sm	Mar 16/02/16	Lun 07/03/16
Synchronisation du robot avec la ligne transitique	2 sm	Mer 03/02/16	Mar 16/02/16
Module de communication internet	2 sm	Mer 17/02/16	Mar 01/03/16
Programmation du module internet	1 sm	Mar 05/04/16	Lun 11/04/16
Application Web	**15 jours**	**Mar 29/03/16**	**Lun 18/04/16**
Création de l'application web	2 sm	Mar 29/03/16	Lun 11/04/16
Communication entre le module et l'application web	1 sm	Mar 12/04/16	Lun 18/04/16
Rapport de projet	99 jours	Jeu 10/12/15	Mar 26/04/16
Soutenance de projet	1 sm	Mar 19/04/16	Lun 25/04/16

Figure 4 : Planning prévisionnel

1.3.2 Répartition des taches

Nous nous sommes répartis les tâches en fonction de nos préférences et de nos compétences, ce qui permet de travailler sur deux tâches simultanément.

La répartition de ces tâches doit également suivre un certain ordre car certaines d'entre elles doivent être terminées avant de pouvoir en commencer d'autres, ce qui peut retarder le planning prévisionnel. Par exemple, la programmation des automates doit être finie avant de commencer la supervision[15] mais la programmation du robot peut se faire en parallèle de la supervision.

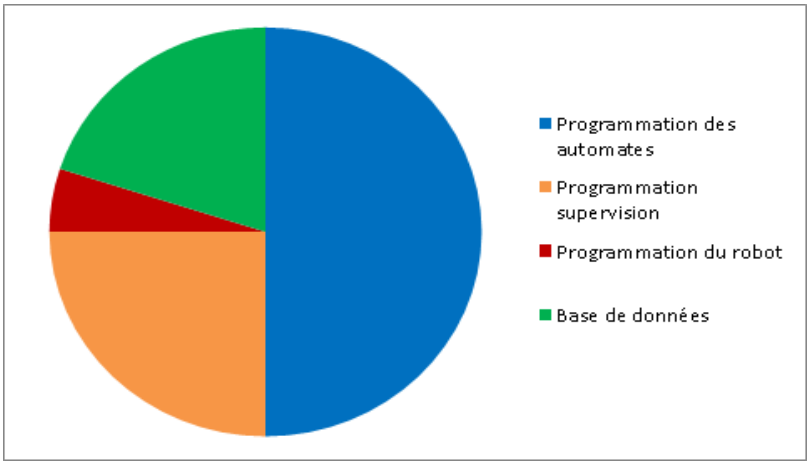


Figure 5 : Diagramme de répartition des tâches globales

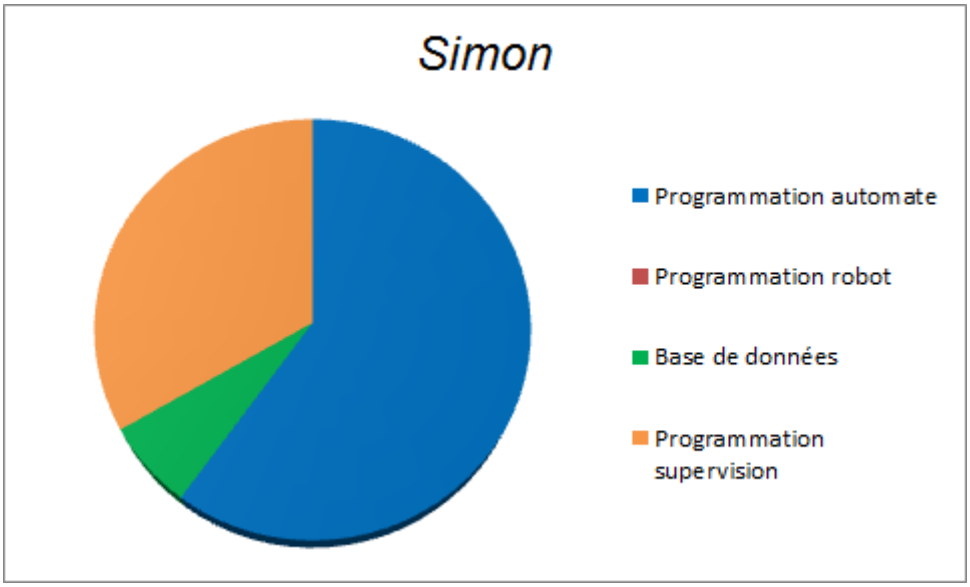


Figure 6 : Diagramme de répartition des tâches accompli par Simon

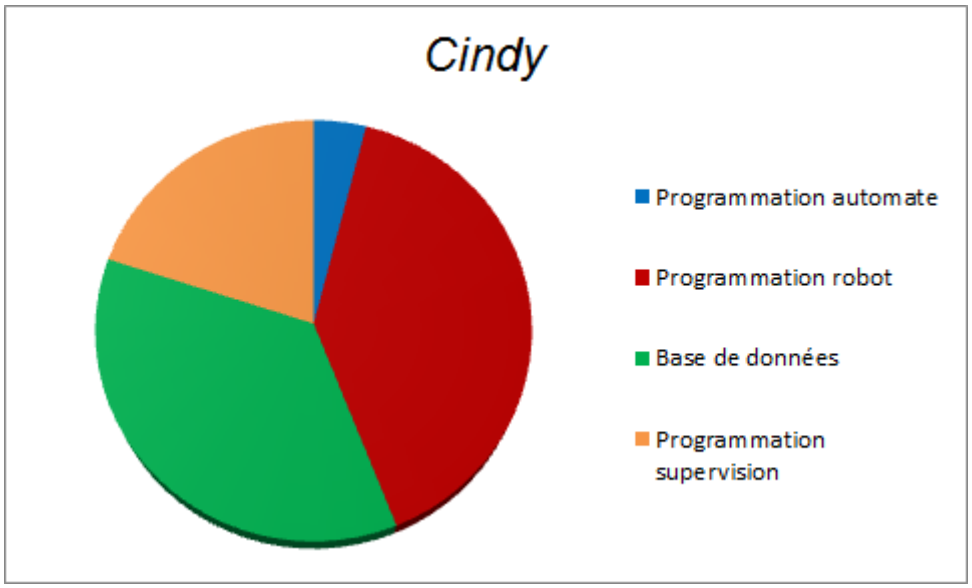


Figure 7 : Diagramme de répartition des tâches accompli par Cindy

2 Programmation de la ligne transitique

La programmation de la ligne fait intervenir à la fois de la programmation automate et le réseau industriel pour la communication entre les API(1). Cette partie du projet constitue la base de celui-ci. En effet, la supervision(15) et la programmation du robot reposent sur l'automatisme.

Nous avons donc commencé par reprendre un TP de réseau industriel de 4ème année. Il utilise la maquette et le protocole Profibus(11).

2.1 Communication réseau entre automates

2.1.1 Présentation des mots réseaux

La ligne transitique, constituée de trois API(1) *(voir paragraphe 1.1.1 Présentation de la maquette),* communique et échange des informations. Ces API(1) sont nécessaires pour la gestion des aiguillages(2) et des butées(5) *(voir paragraphe 1.1.2 Présentation des besoins)*.

Cette communication est réalisée par le biais de mots réseaux communs. Chaque API(1) dispose d'une plage de mots réseaux. Ils peuvent écrire sur la plage leur étant réservée et uniquement lire les plages des autres API(1).

A chaque cycle réseau, l'API(1) lit les mots réseaux des autres l'API(1) et écrit sur les siens *(voir paragraphe 1.2.1 Architecture réseau)*.

Automate du Magasin	Automate de l'Hippodrome	Automate des Épis
> MB 116:4	MB 116:4	MB 116:4
MB 120:4	> MB 120:4	MB 120:4
MB 124:4	MB 124:4	> MB 124:4

Figure 8 : Tableau des plages de mots réseaux

Les cases bleues avec le chevron ">" indique que l'API(1) écrit sur cette plage. Par exemple, > MB 116:4 indique que l'API(1) du magasin écrit à partir du mot M116 jusqu'au mot M119. Chaque mot est composé de 8 bits(4) (M116.0 à M116.7).

2.1.2 Gestion des aiguillages(2) (sémaphore(13))

La gestion des aiguillages(2) assure le guidage des palettes sur la ligne transitique. Cependant les aiguillages(2) peuvent être partagés sur différentes parties de la maquette *(voir paragraphe 1.1.1 Présentation de la maquette)*. Pour assurer et synchroniser la liaison entre ces parties, on utilise des sémaphores(13).

figure 9 : Exemple de sémaphore[13] pour le capteur C6

Premier API[1]		**Deuxième API[1]**
Envoie d'une demande « DemLibre ».	→	Lecture de cette demande.
		↓
Lecture de l'autorisation. Cessation d'émission de la demande.	←	Si la palette est présente sur le capteur, il va envoyer une autorisation « AutoLibre ».

2.2 Programmation automate

2.2.1 Présentation des GRAFCET[6]

Avant de se lancer dans la programmation automate, nous avons écrit les GRAFCET[6] qui régissent le fonctionnement du système *(voir Annexe I. Présentation des GRAFCET).*

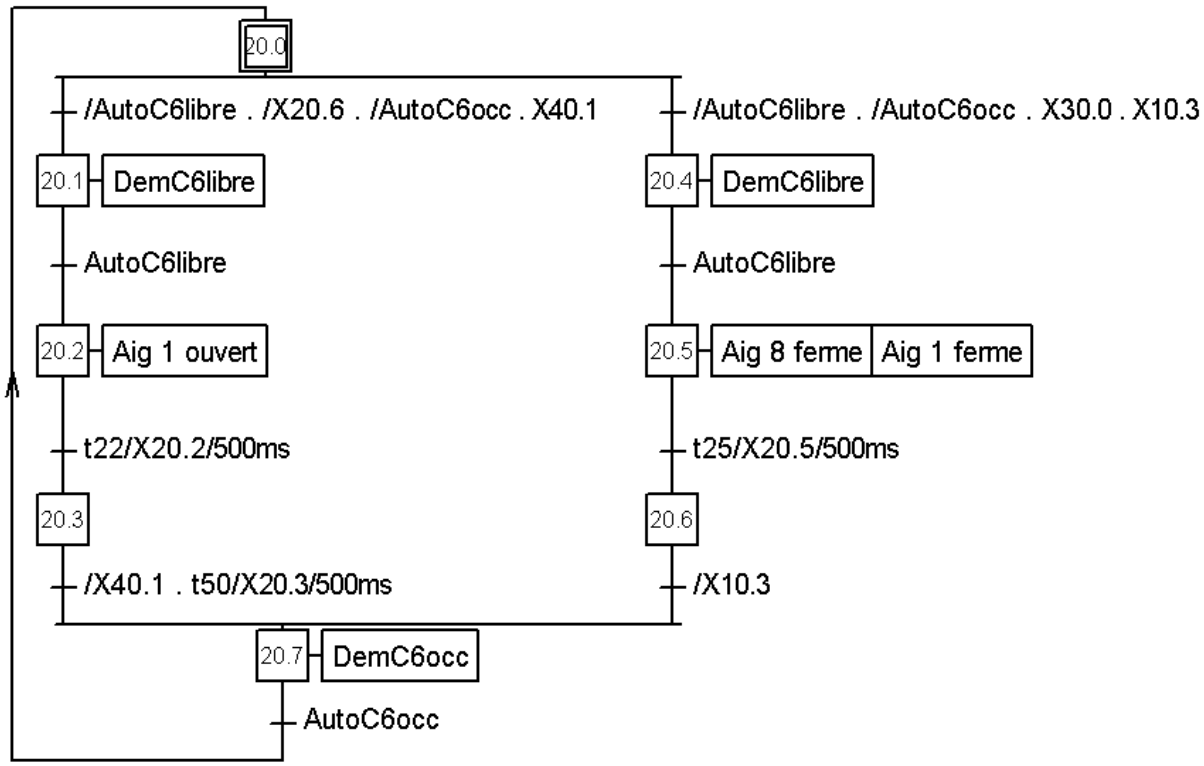


Figure 9 : GRAFCET[6] de l'aiguillage[2] 1

L'étape 20.0 représente l'étape initiale. C'est-à-dire, elle caractérise l'état du système au début du fonctionnement.

La transition pour franchir l'étape 20.0 présente une divergence. Deux choix sont possibles : soit une palette est en C13, soit elle est en C5. Dans le cas où C13 et C15 sont activées au même moment, une priorité arbitraire est donnée.

La transition pour franchir l'étape 20.3 attend que la temporisation t50 (lancée sur activation de cette étape) soit terminée.

2.2.2 Programmation en langage LIST(8)

La programmation des API(1) Siemens est faite en langage LIST(8) via le logiciel SIMATIC. Nous traduisons alors les GRAFCET(6) en langage LIST(8) *(voir Annexe II. Programmation LIST).* Ce langage utilise des commandes simples et équivalentes à la syntaxe des GRAFCET(6).

```
U     M     20.0
U     M     40.1
UN    "AutoC6libre"
UN    M     20.6
UN    "AutoC6Occ"
S     M     20.1
R     M     20.0

U     M     20.1
U     "AutoC6libre"
S     M     20.2
R     M     20.1

U     M     20.2
U     T     22
S     M     20.3
R     M     20.2

U     M     20.3
UN    M     40.1
U     T     50
S     M     20.7
R     M     20.3
```

```
<- Transition entre
les étapes 20.0 et
20.1
Transition entre les
étapes 20.0 et 20.4 ->




Les bits(4) avec un "U"
devant doivent être à
"1" et ceux avec "UN"
à "0", si c'est le cas
on met à "1" les
bits(4) qui suivent un
"S" et à "0" ceux qui
suivent un "R"




<- Temporisation t50
```

```
U     M     20.0
U     M     10.3
U     M     30.0
UN    "AutoC6libre"
UN    "AutoC6Occ"
S     M     20.4
R     M     20.0

U     M     20.4
U     "AutoC6libre"
S     M     20.5
R     M     20.4

U     M     20.5
U     T     25
S     M     20.6
R     M     20.5

U     M     20.6
UN    M     10.3
S     M     20.7
R     M     20.6

U     M     20.7
U     "AutoC6Occ"
S     M     20.0
R     M     20.7
BE
```

Figure 10 : Langage LIST(8) de l'aiguillage(2) 1

On peut voir ci-dessus, la symétrie avec le GRAFCET(6) correspondant présenté au paragraphe précédent *(voir paragraphe 1.1.1 Présentation de la maquette)*.

2.3 Gestion des défauts

2.3.1 Détection des défauts

La circulation des palettes sur la ligne transitique peut présenter quelques problèmes, notamment dans les virages *(voir Annexe IV.I Carte des virages)*. En effet, les palettes ont parfois du mal à passer les virages et peuvent rester bloquées. Un opérateur doit alors intervenir. Pour limiter le temps d'intervention de l'opérateur, on va détecter ces défauts.

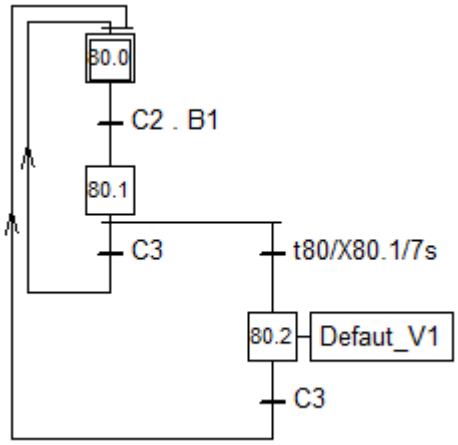


Figure 11 : GRAFCET[6] de défaut du virage 1

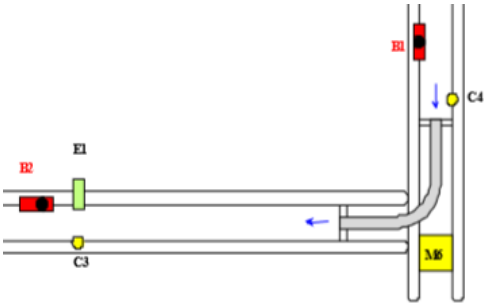


Figure 12 :Image du virage 1

Pour détecter un défaut, on déclenche une temporisation dès que la butée[5] se baisse et libère la palette. Si la temporisation arrive à terme sans que la palette ne soit parvenue au capteur situé après le virage, un bit[4] est mis à "1" pour identifier le défaut. Il sera remis à "0" dès que la palette sera sur le capteur.

La valeur de chacune des temporisations a été ajustée à la suite de plusieurs essais chronométrés.

2.3.2 Calcul des temps de défauts

Les calculs des temps de défauts sont effectués en langage C.

```
if(New_turn_default[c] && !flagCounter[c])
{
    flagCounter[c]=1;
    Time_default[c]=SystemTime();
}
if(!New_turn_default[c] && flagCounter[c])
{
    flagCounter[c]=0;
    Time_default[c]=SystemTime()-Time_default[c]+TimeAPI[c];

    NbDefault[c] ++;
    Counter_default[c] += Time_default[c];
    Counter_average[c] = Counter_default[c] / NbDefault[c];
}
```

Figure 13 : Programmation des temps de défauts

Lorsque l'API(1) détecte un défaut, on enregistre le temps t_1 du système PC. Une fois le défaut disparu de l'API(1), on récupère le nouveau temps t_2 du système PC. On peut alors calculer le temps de défauts « Time_default » du virage : t_2 - t_1 +temps de temporisation de l'API(1).

On veut également connaitre le temps moyen des défauts s'étant présentés dans chaque virage. Pour cela, on additionne le temps de défauts « Counter_default » et compte le nombre de défaut « NbDefault ». On réalise ensuite la division entre ces derniers.

3 Supervision

La supervision[15] de la maquette se fait en deux étapes : la première étape consiste à récupérer les données de l'API[1] pour ensuite les enregistrer dans une BDD[3]. Cette base est accessible par un utilisateur connecté à internet appelé client. Pour interagir plus facilement avec ce client, une interface web[7] est mise en place. La création de cette interface web[7] représente la deuxième étape. *Voir paragraphe 1.2.2. Architecture de la supervision.*

3.1 Présentation de la base de données

3.1.1 Gestion des tables

La structure de notre BDD[3] est organisée sous forme de tables. Les éléments principaux de notre supervision[15] sont la visualisation des défauts (dans les virages) et des présences de palettes (sur les capteurs). Ainsi, notre BDD[3] est constituée de deux tables.

Figure 14 : Tables sous le logiciel WampServer

- La table des capteurs *"sensors"* récupère les informations correspondant à l'état des capteurs de présence d'une palette.
- La table des défauts *"defaults"* correspond l'état d'un défaut dans un virage, la durée du dernier défaut présent, le nombre de détection des défauts et le temps moyen des défauts.

3.1.2 Gestion des utilisateurs

Tout d'abord, il faut savoir que lorsque l'on se connecte à la BDD[3], on le fait avec un utilisateur. Afin de définir les droits des utilisateurs, nous leurs donnons certains "privilèges".

La supervision[15] présente deux utilisateurs :

Utilisateur	Client	Type	Privilèges
production	localhost	global	ALL PRIVILEGES
		spécifique à cette base de données	ALL PRIVILEGES
supervision	localhost	global	SELECT
		spécifique à cette base de données	SELECT

Figure 15 : Utilisateurs sous le logiciel WampServer

- Le gestionnaire de production, appelé “production”, réalise la communication entre l’API(1) et la BDD(3). Celui-ci gère la BDD(3), il a donc tous les droits sur celle-ci “ALL PRIVILEGES”.
- Le gestionnaire de l'interface web(7), appelé “supervision”, réalise la communication entre la BDD(3) et le client. Ce dernier peut uniquement visualiser la BDD(3) (via une page Web), il ne peut donc pas modifier les données de celle-ci. Pour se faire, il utilise uniquement des requêtes(12) “SELECT”.

3.2 Communication à la base de données en langage C

3.2.1 Récupération des données de l’API(1)

Afin de récupérer les informations de l’API(1), il faut tout d'abord établir une connexion avec ce dernier. Elle se fait de manière suivante :

```
iOctet = 0;
neq = 4;
adr = iOctet*8;
readpackibit(&nchan, &neq, &nb, &adr, tabl, &status);
if (!status)
{
    transwordbit(&nb, tabl, tablbit, &status);
}
else
{
    printf("Problème sur l'équipement %d",neq);
}
```

Figure 16 : Code de la connexion à l’API(1) en langage C

- Définition de l’API(1) de lecture “neq”.
- Définition de l’adresse de la première donnée de la liste “adr”. Cette donnée est définie par son numéro d’octet(9) et de bit(4). Ici, 0.0. De plus, un octet(9) est constitué de 8 bits(4). On réalise alors le calcul suivant pour définir l’adresse : **n°octet * 8 + n°bit**.
- Lecture dans l’API(1) “readpackibit” : Lecture de 64 données “nb”, recopie des données dans une variable local “tabl” et “état de la connexion “status”.
- Transfert des données “transwordbit” dans une nouvelle variable “tablbit”, si la connexion s’est bien déroulée (“status” valant 0).

Les données provenant de l'API[1] sont de deux types : les données d'entrée I et les données mémoires M. Les données d'entrées se lisent via la fonction "readpackibit". Les données mémoires se lisent via la fonction "readpackbit".

Une fois la connexion avec l'API[1] réalisée, nous classons les données dans une variable plus explicite pour faciliter leurs utilisations.

```
New_measure_sensor[5]=tablbit[0*8+3];       //E0.3
```

Figure 17 : Code du classement des données de la BD[3]D en langage C

3.2.2 Envoi des données à la base de données

Les données sont envoyées à la BDD[3] via des requêtes[12] SQL[14]. Il faut tout d'abord établir une connexion avec la BDD[3]. Pour cela, il faut inclure une librairie MySQL. Puis, la connexion se fait de manière suivante :

```
MYSQL mysql;
mysql_init(&mysql);
mysql_options(&mysql,MYSQL_READ_DEFAULT_GROUP,"option");

///Si la connexion réussie...
if(mysql_real_connect(&mysql,"127.0.0.1","production","production","projet",0,NULL,0))
```

Figure 18 : Code de la connexion à la BDD[3] en langage

- Déclaration d'un objet de type MySQL.
- Initialisation de l'objet "mysql_init".
- Définition des options "mysql_options".
- Connexion à la BDD[3] via l'utilisateur production *(voir paragraphe 3.1.2 Gestion des utilisateurs)*.

Nous écrivons dans la BDD[3] à chaque changement d'état des données :

```
if((Old_turn_default[c]==0)&&(New_turn_default[c]==1))
{
    sprintf(requete,"UPDATE `defaults` SET `etat`=1 WHERE `nom` = 'V0%d'", c);
    mysql_query(&mysql,requete);
}
```

Figure 19 : Code de l'écriture dans la BDD[3] en langage C

- Détection du changement d'état.
- Recopie de la requête[12] SQL[14] à envoyer dans la BDD[3]. La mise à jour d'une variable est réalisée par la requête[12] "UPDATE".
- Envoie de la requête[12] à la BDD[3] "mysql_query".

3.3 Interface Web[7]

3.3.1 Récupération des données de la base

L'interface web[7] permet à l'utilisateur de connaître en temps réel l'état de la ligne transitique, sans oublier les temps de défauts des virages.

Cette interface[7] a été codée en utilisant plusieurs langages, du HTML essentiellement pour la mise en page (la disposition des tableaux de données, des titres…), du CSS dans la feuille de style pour la mise en forme (couleurs utilisées, taille …) et du PHP[10] pour la récupération des données de la base *(voir Annexe III.II Programme interface web)*.

Pour afficher ces données, nous les récupérons depuis la BDD[3].

D'abord, l'interface[7] doit se connecter à la BDD[3], avec un identifiant et un mot de passe.

```php
<?php
        // Connexion à la base de données
    try
    {
        require("include_bdd.php");
    }
    catch(Exception $e)
    {
        die('Erreur : '.$e->getMessage());
    }
?>
```

Figure 20 : Code d'appel de la page de connexion à la BDD[3] en langage PHP[10]

On fait appelle à une autre page de code PHP[10] « require ». Si des erreurs apparaissent, on écrit l'erreur puis on quitte l'exécution du programme.

```php
<?php
$bdd = new PDO('mysql:host=localhost;dbname=projet', 'supervision', 'supervision');
?>
```

Figure 21 : Code de la connexion à la BDD[3] en langage PHP[10]

- Connexion à la BDD[3] via l'utilisateur *(voir paragraphe 3.1.2 Gestion des utilisateurs)*.
- Accès à la BDD[3] via l'objet « $bdd ».

Après la connexion avec la BDD[3] fonctionnelle, on peut passer à la récupération des données. On interroge les tables de la BDD[3] via des requêtes[12] SQL[14] « SELECT » envoyées par un code PHP[10].

```php
$reqCapteurs = $bdd->query('SELECT * FROM sensors ORDER BY nom');
```

Figure 22 : Code de lecture de la BDD[3] en langage PHP[10]

- Envoie de la requête[12] à la BDD[3] « $bdd->query ».
- Recopie de la sélection de la BDD[3] dans une variable « reqCapteurs ».

Un script en langage Javascript est intégré au fichier “supervision.php“ pour rafraîchir les données toutes les secondes. Il garantit un aperçu en temps réel de la ligne.

3.3.2 Affichage des données

Une fois les données récupérées, on les affiche sur l’interface web[7] de deux façons différentes. *(voir Annexe IV.II Images de l’ interface web)*

La première consiste à lister les données dans des tableaux (un pour les capteurs et un pour les défauts).

```php
<?php
    $reqCapteurs = $bdd->query('SELECT * FROM sensors ORDER BY nom');

    while($donneeCapteurs = $reqCapteurs->fetch())
    {
        echo "<TR>
                <TD align=center>",$donneeCapteurs['nom'],"</TD>
                <TD align=center>",$donneeCapteurs['etat'],"</TD>
            </TR>";          // affichage du nom et de l'état du capteur
    }
?>
```

Figure 23 : Code du tableau des capteurs en langage PHP[10]

- Récupération des données de la BDD[3] triées suivant le « nom ».
- Recopie de la sélection de la BDD[3] dans une variable « reqCapteurs ».
- Parcourt ligne par ligne du contenu de la variable “reqCapteurs” : « fetch ».
    - ❖ Recopie de la ligne dans la variable “donneeCapteurs”.
    - ❖ Affichage de la table « echo ».

La deuxième façon d’afficher l’interface web[7] consiste à afficher une image correspondant à la ligne transitique et représenter des palettes (par superposition d’éléments) lorsque les capteurs et/ou les défauts sont actifs.

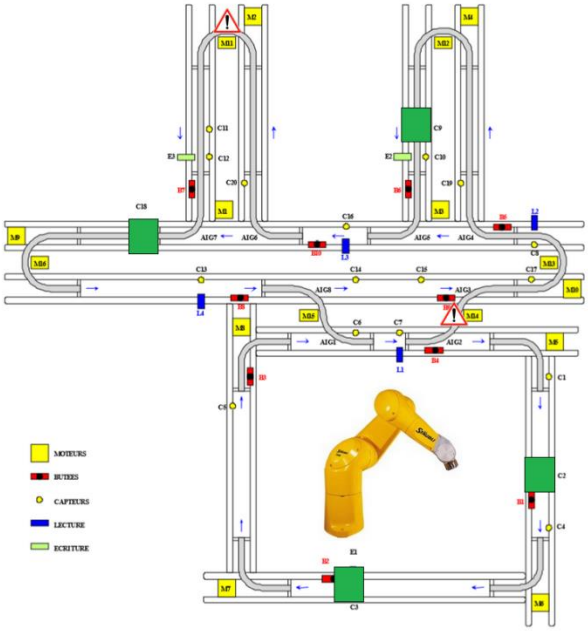


Figure 24 : Interface Web[7] avec l'image de la ligne transitique

Dans un premier temps, on affiche une image de la ligne, puis on superpose des carrés verts où les capteurs sont actifs et des panneaux "attention" où les défauts sont actifs.

```
$i_C = 0;       // indice pour les capteurs
$Pos_image_left_C = [1010,1010,740,1010,590,740,820,1000,835,835,530,530,525,750,835,740,1000,450,910,605];
$Pos_image_top_C = [780,910,1065,990,820,725,725,565,415,470,415,470,650,650,650,575,650,573,495,495];

$reqCapteurs = $bdd->query('SELECT * FROM sensors ORDER BY nom');

while($donneeCapteurs = $reqCapteurs->fetch())
{
    $LEFT = $Pos_image_left_C[$i_C];
    $TOP = $Pos_image_top_C[$i_C];
    if($donneeCapteurs["etat"]=="1")
            echo '<DIV style="position:absolute; left:'.$LEFT.'px; top:'.$TOP.'px; width:136px; height:132px; z-index:2">
            <IMG src="C_on.png"> </DIV>';   // affichage d'un carré vert à la position définie
    $i_C+=1;
}
```

Figure 25 : Code de l'image des capteurs en langage PHP[10]

Les « carrés verts » sont définit par leur position "left" et "top", pré-définit dans des tableaux. Ces positions ont été calculées à partir de l'affichage de l'image ligne.

La position est figée avec "position:absolute". Puis, on fait passer le carré vert devant l'image de la ligne à l'affichage avec "z-index:2". En effet, l'image de la ligne possède comme paramètre "z-index:1" (la valeur la plus élevée se retrouve au premier plan).

4 Programmation du robot

L'intégration du robot dans notre projet fait intervenir plusieurs éléments. Dans un premier temps, la prise en main puis la programmation du robot. Pour associer le robot à la ligne transitique, une modification des GRAFCET(6) est nécessaire. Enfin, en cas de défaillance du robot, deux modes de fonctionnements sont implémentés.

4.1 Présentation du robot

4.1.1 Fonctionnement du robot

Avant toutes manipulations du robot, il faut comprendre son fonctionnement. Un robot se caractérise par un système semblable à un bras humain. Notre robot est poly-articulé à 6 degrés de liberté.

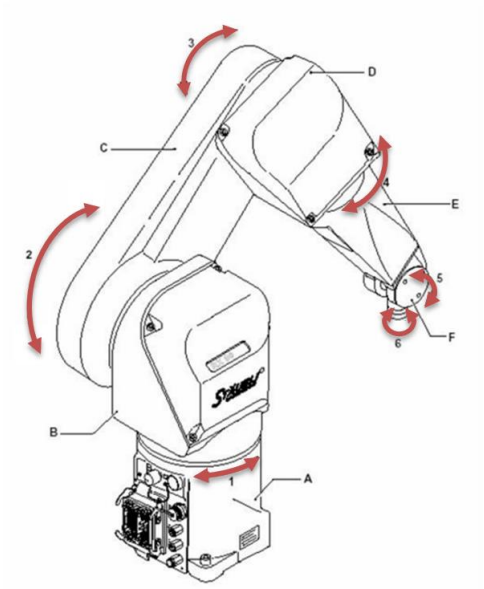


Figure 26 : Image des 6 degrés de libertés du robot

Le robot est constitué d'une baie de commande, d'un teach penant et d'un PC de programmation *(voir Annexe IV.III Composition du robot).*

La programmation du robot consiste à lui indiquer les mouvements qu'il doit réaliser. Il existe deux moyens de programmation :

La programmation manuelle
Nous l'avons utilisé, principalement, pour connaître les cordonnées des positions du robot.

La programmation PC
Nous l'avons utilisé pour réaliser des enchaînements de positions et interagir avec la ligne transitique.

4.1.2 Définition des actions réalisées

La programmation se décompose en deux parties : les programmes de gestion et les programmes de mouvements *(voir Annexe III.III Programme robot).*

Les programmes de gestions sont constitués de deux sous-programmes.

- Le bouclage infini, nommé “boucle”, du programme principal. En effet, le robot ne doit pas s’arrêter après avoir réalisé un déplacement.
- Le programme principal, nommé “main”, appelant un mouvement précis selon les informations provenant de l’API[(1)] : `IF(1001, -1002, -1003)`

  Puis, renvoie à l’API[(1)] la position du robot : `SIGNAL 1, -2, -3`

Les programmes de mouvements sont constitués de cinq sous-programmes.

- La simulation de chargement d’une pièce.
- La simulation de déchargement d’une pièce.
- La simulation d’usinage d’une pièce.
- Le repos du robot après le déchargement.
- Le repos du robot après usinage ou chargement.

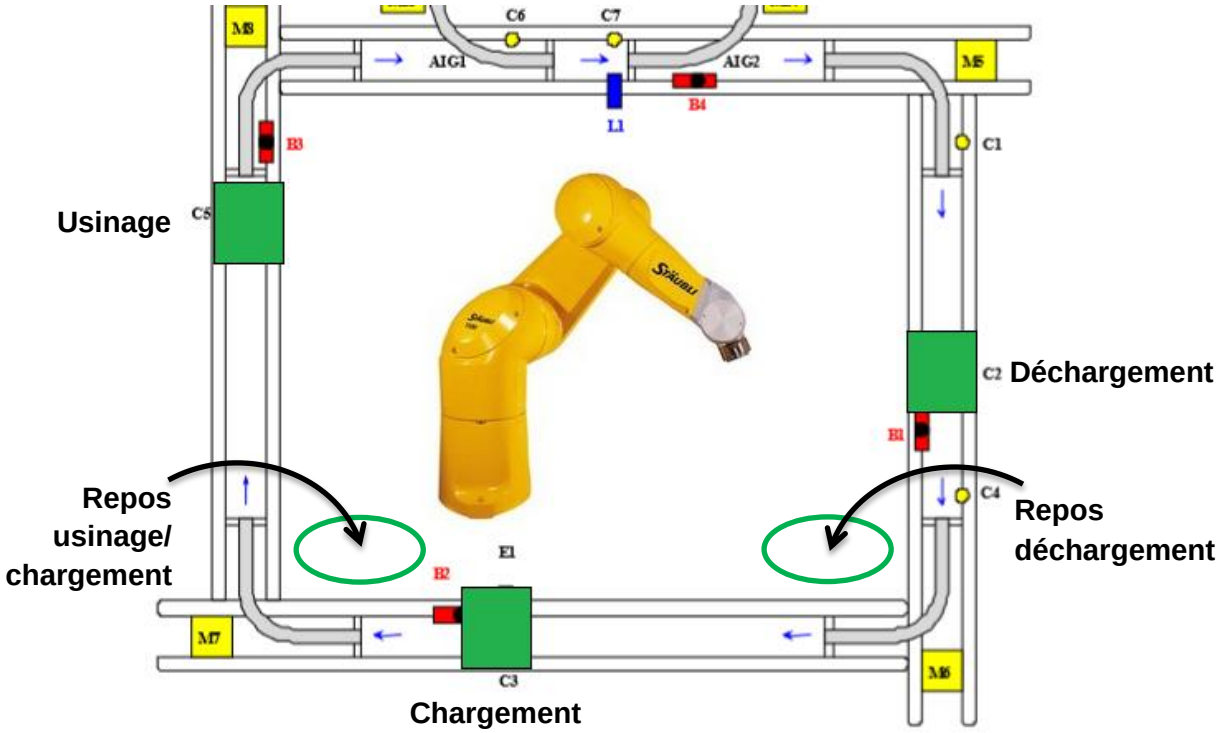


Figure 27 : Image des positions du robot

4.2 Adaptation des GRAFCET(6) automates

4.2.1 Intégration du robot dans le programme automate

Suite à la programmation du robot, nous avons intégré ce dernier au fonctionnement du programme automate. Certains GRAFCET(6) ont donc été modifiés, principalement ceux de la partie "Magasin". En effet, le robot intervient essentiellement sur cette partie.

La communication entre l'API(1) et le robot est réalisée par un câblage en direct *(voir paragraphe 1.2.1 Architecture réseau).* Les GRAFCET(6) doivent donc prendre en compte ces nouvelles informations.

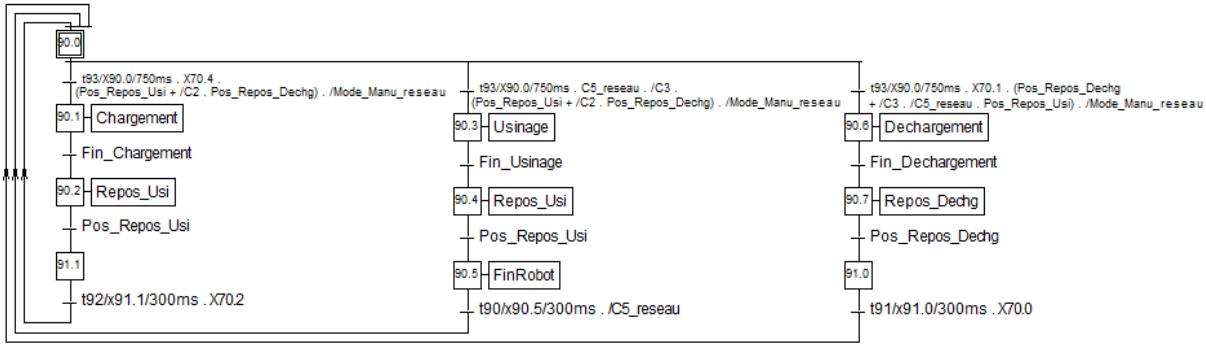


Figure 28 : GRAFCET(6) de la gestion du robot

Les trois branches représentent respectivement le mouvement de chargement, d'usinage et de déchargement. Une fois le mouvement effectué, le robot est mis en position de repos associée.

4.2.2 Définition des priorités

En production, il est fréquent que des palettes se trouvent dans différentes phases simultanément. C'est-à-dire, une palette présente au poste de chargement, de déchargement et/ou d'usinage. Il est alors nécessaire de définir des priorités pour éviter les conflits.

Ces priorités dépendent de la position de repos du robot.
Lorsque le robot est dans la position de repos déchargement "ReposDechg" :

- Il effectue le déchargement d'une pièce en priorité.
- Il effectue le chargement d'une pièce en second.

Lorsque le robot est dans la position de repos usinage/chargement "ReposUsi"

- Il effectue le chargement d'une pièce en priorité.
- Il effectue l'usinage d'une pièce en second.

4.3 Mode d'utilisation « sans robot »

4.3.1 Choix des modes

La ligne transitique peut fonctionner selon deux modes : le mode « avec robot » et le mode « sans robot ». Ce choix s'effectue via un sélecteur auto/manu situé sur l'armoire électrique de la ligne.

Dans le cas où ce sélecteur est en position "manu", le programme automate n'attend aucune information provenant du robot. Le robot n'est pas pris en compte. On appelle ce mode « sans robot ».

Lorsque le sélecteur est en position "auto", le programme automate demande au robot de réaliser les mouvements correspondant. On appelle ce mode « avec robot ».

4.3.2 Modification des GRAFCET(6)

Le robot interagit principalement avec l'API(1). C'est pourquoi seul les GRAFCET(6) du "Magasin" ont été modifiés. Au niveau de la gestion des butées(5), on intègre le sélecteur "Auto/Manu" aux conditions de transition.

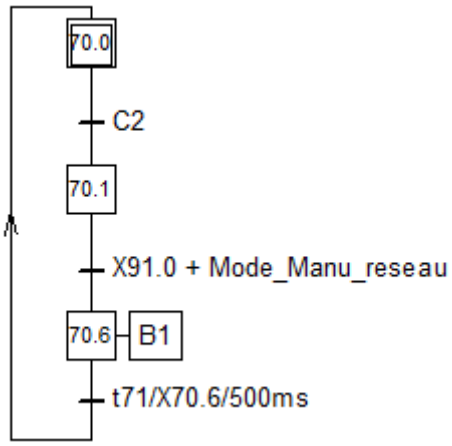


Figure 29 : GRAFCET(6) de la gestion de la butée(5) B1

La modification a lieu au niveau de la transition entre les étapes 70.1 et 70.6. Soit on a le mode « sans robot » "Mode_Manu_reseau" (correspondant au sélecteur "Auto/Manu" sur la position "Manu"). Soit on a le mode « avec robot », l'étape 91.0 du GRAFCET(6) de gestion du robot.

CONCLUSION

Ce projet s'est déroulé sur une période de 5 mois, à durée de 1 jour par semaine. Nous avons pu mettre en pratique nos compétences acquises au cours de notre cursus scolaire. En outre, la programmation d'automates, la supervision(15) d'un système ainsi que la programmation d'un robot.

De plus, ce projet a été l'opportunité d'acquérir de nouvelles aptitudes en matière d'autonomie, de gestion de temps, de travail en équipe et de prise de décisions.

Malheureusement, nous avons manqué de temps pour finir le projet. Nous n'avons pu faire le pilotage de la ligne via la supervision(15) et l'application sur tablette. Deux à trois semaines supplémentaires semblent nécessaires pour pouvoir terminer ce projet.

PERSPECTIVE

De nos jours, le monde industriel utilise des systèmes de production automatisés. Ces technologies présentent de nombreux avantages en augmentant la productivité et le rendement de l'entreprise. Les convoyeurs sont principalement utilisés dans l'industrie agro-alimentaire ou les centres de triage.

De plus en plus utilisés dans l'industrie, les robots présentent une rapidité et une précision impressionnante. Ils sont principalement utilisés dans le nucléaire ou les industries de séries. Et permettent également d'aider l'être humain pour le maniement d'objets lourds.

Des systèmes similaires, à la ligne transitique ou le robot, sont principalement présents dans l'industrie.

BILAN PERSONNEL

Cindy

Arrivée à terme de projet, le bilan s'avère très positif.

En effet, ce projet m'a permis d'avoir une vision plus étendue de l'automatisme et de la supervision, mais également de robot. J'ai acquis de l'expérience et développé mes connaissances.

Le projet m'a également permis de travaillé sur la communication en équipe et la gestion du temps, ce qui, à mon avis, est indispensable pour réussir en entreprise.

Cependant le fait de ne pas avoir terminé ce projet, par manque de temps, fût frustrant. Ne pouvant apprécier au maximum de notre projet et de sa finalisation.

Simon

Le bilan de ce projet est très positif, j'ai pu mettre en pratique mes compétences en automatisme et supervision(15) ainsi que découvrir la robotique. La communication au sein du binôme a été très bonne, mais je suis déçu de ne pas avoir eu le temps de finir le projet dans sa totalité. Ce fut néanmoins une bonne expérience qui donne un aperçu du travail en équipe et du fonctionnement d'un projet.

Cela nous prépare pour notre intégration en entreprise lors du stage.

WEBOGRAPHIE

Téléchargement de la librairie MySQL, OpenClassRoom :
https://openclassrooms.com/courses/utiliser-l-api-mysql-dans-vos-programmes

Utilisation de fichier CSS, OpenClassRoom :
https://openclassrooms.com/courses/apprenez-a-creer-votre-site-web-avec-html5-et-css3/mettre-en-place-le-css

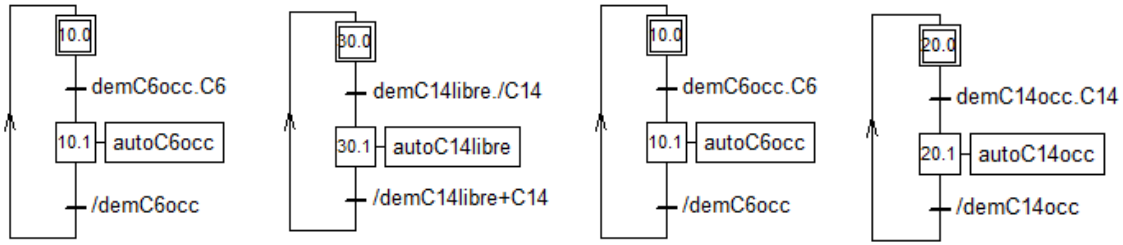
Connexion à une base de données en PHP, OpenClassRoom :
https://openclassrooms.com/courses/concevez-votre-site-web-avec-php-et-mysql/lire-des-donnees-2

ANNEXES

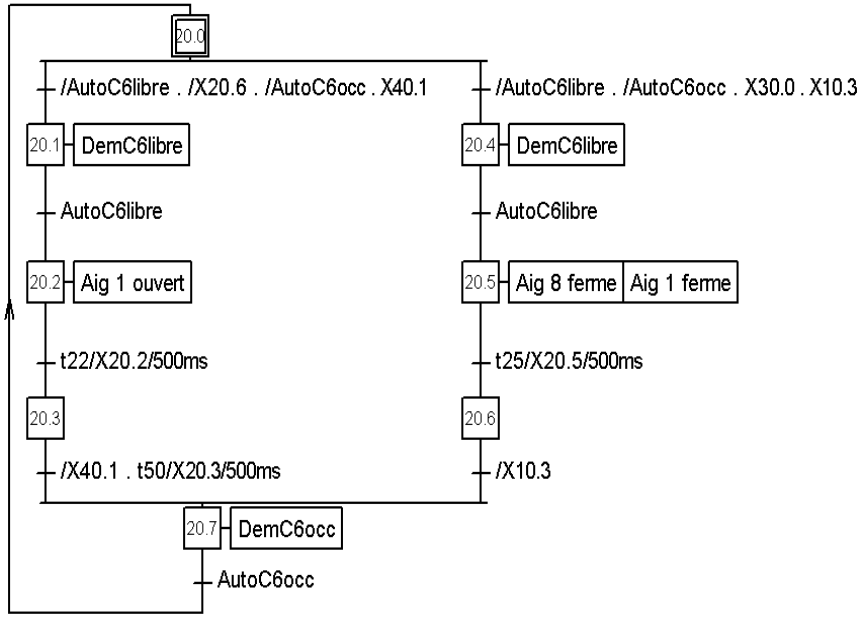
Annexe I. Présentation des GRAFCET

Annexe I.I GRAFCET de la Station 4 « Hippodrome »

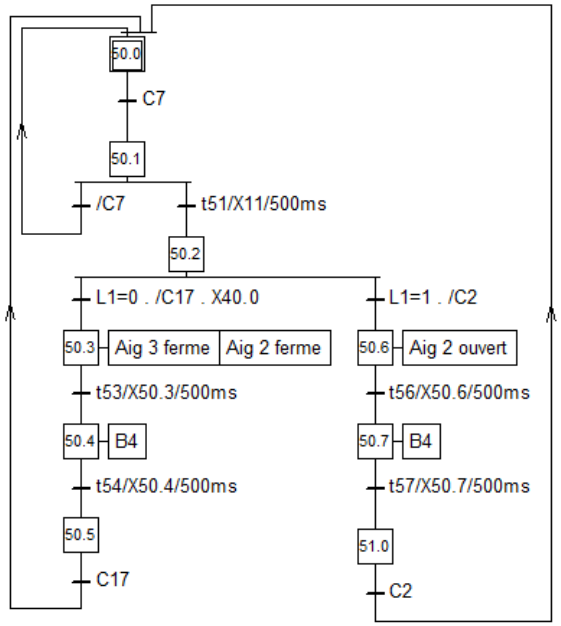
Bloc FB4 :



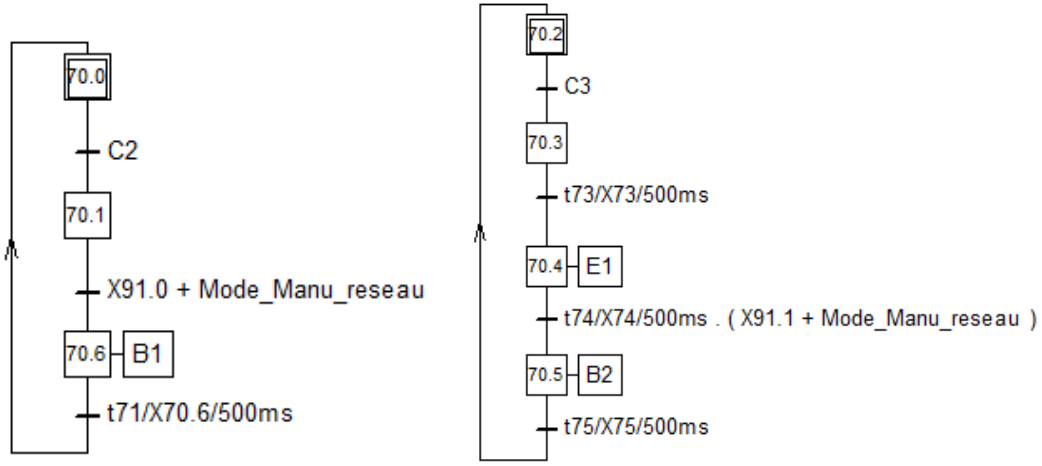
Bloc FB9 :



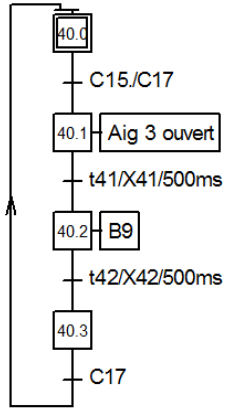
Bloc FB6 :



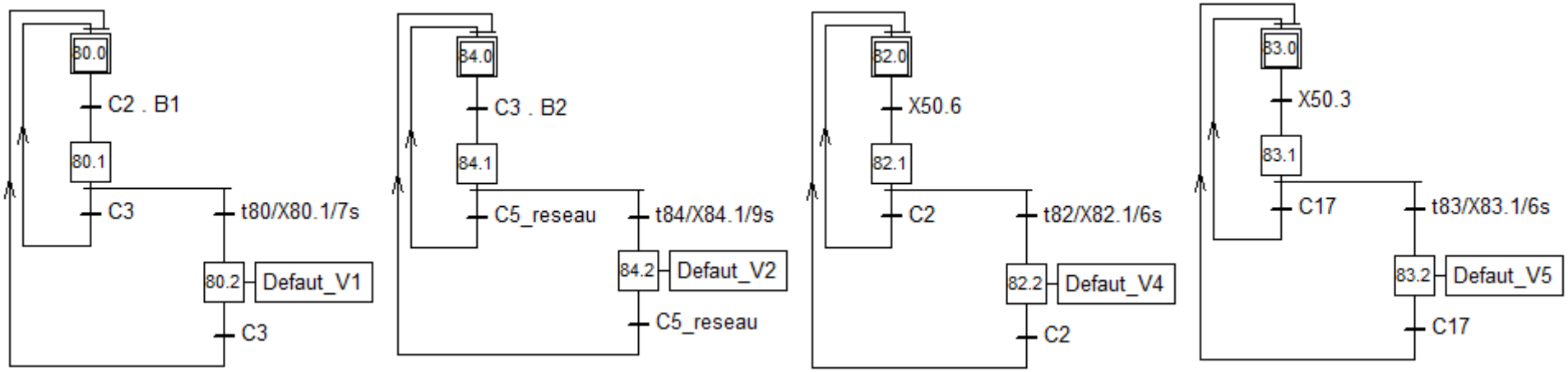
Bloc FB7 :



Bloc FB5 :

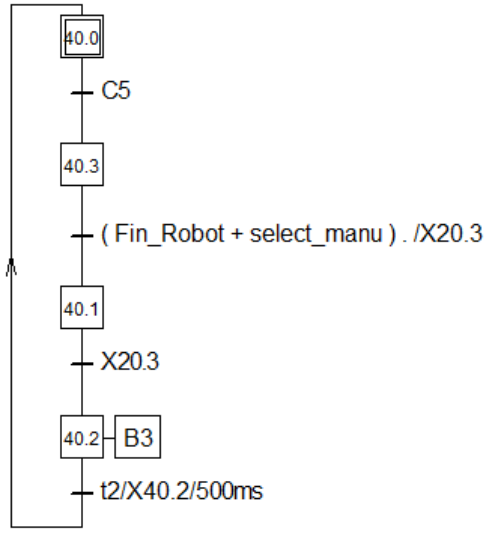


Bloc FB8 :

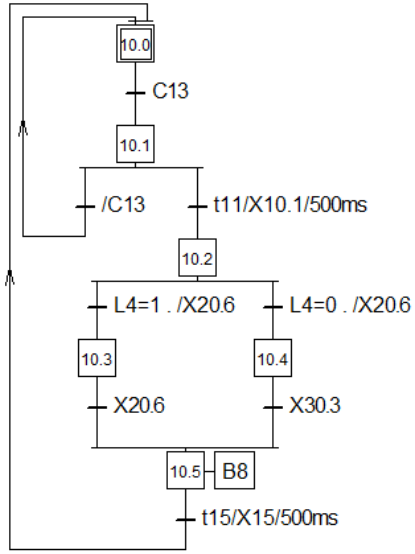


Annexe I.II GRAFCET de la Station 5 « Magasin »

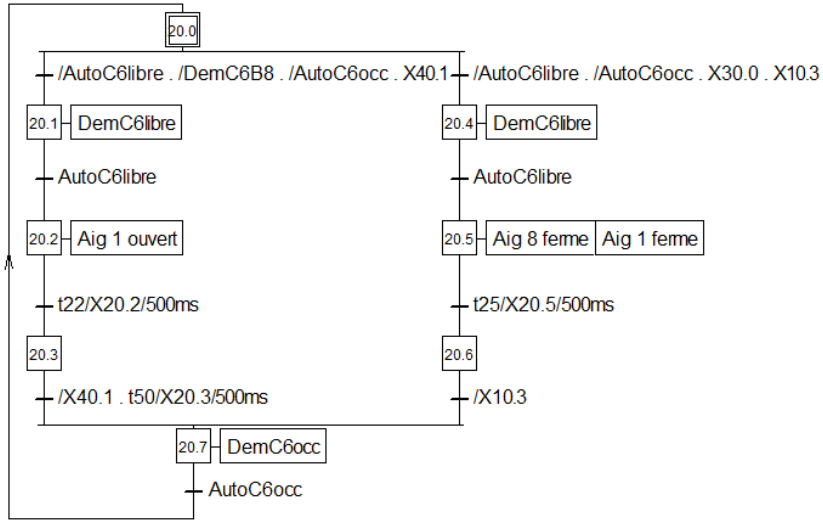
Bloc FB6 :



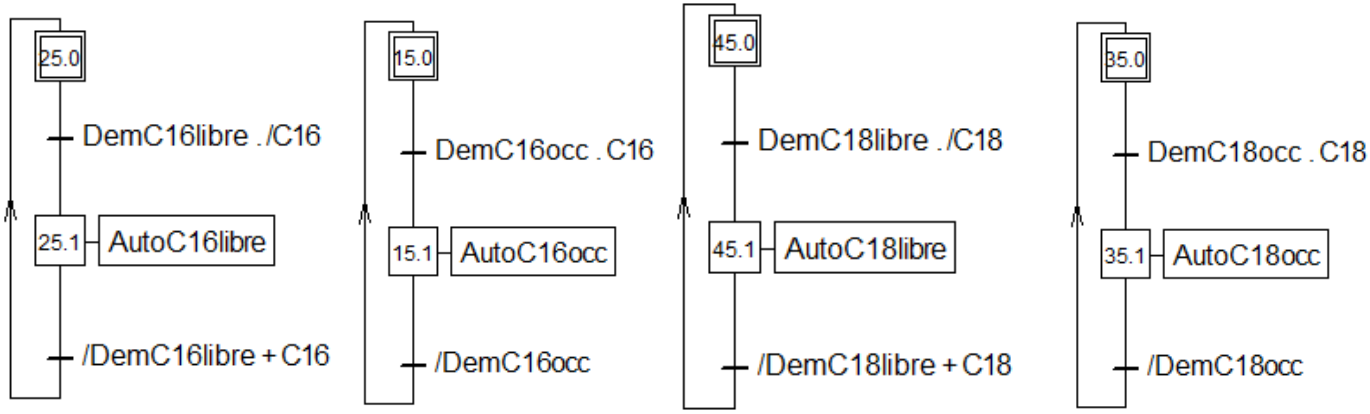
Bloc FB7 :



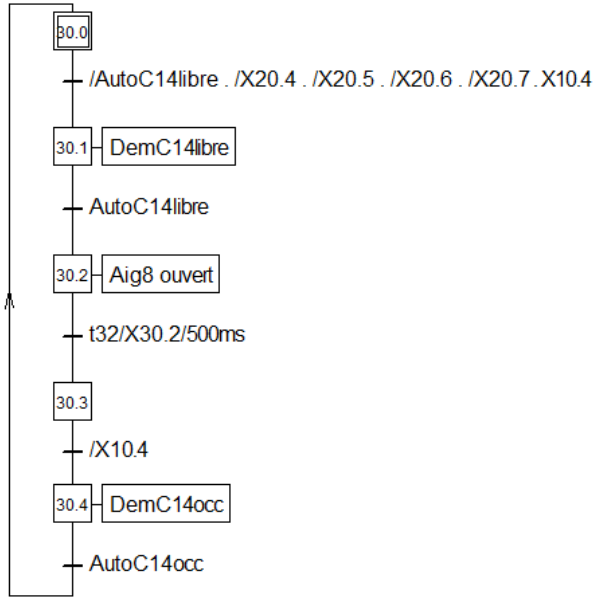
Bloc FB8



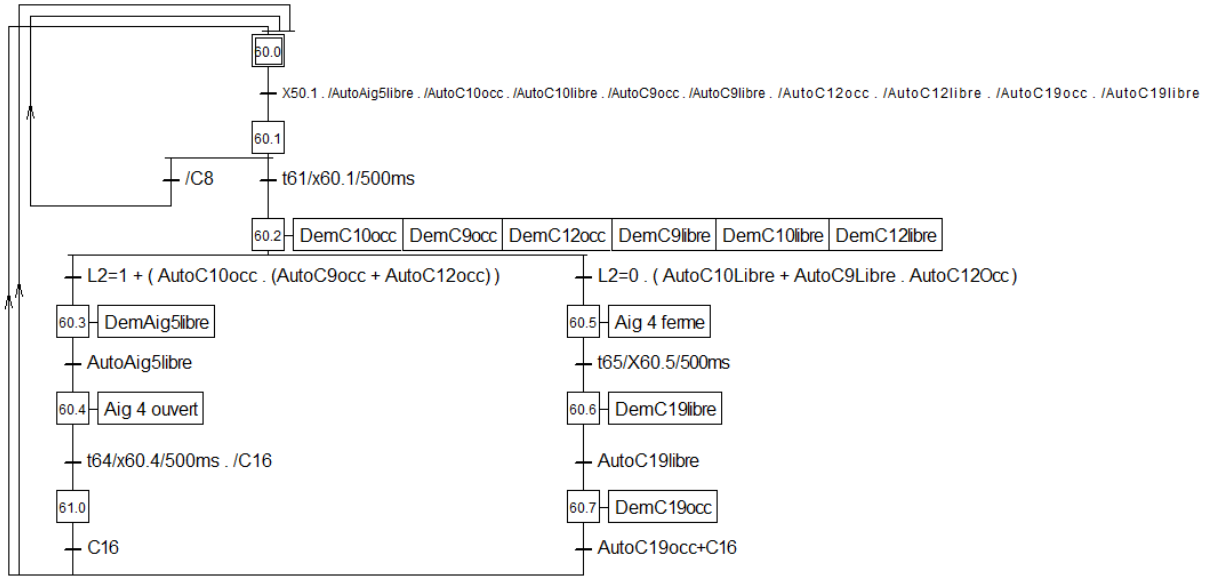
Bloc FB13 :



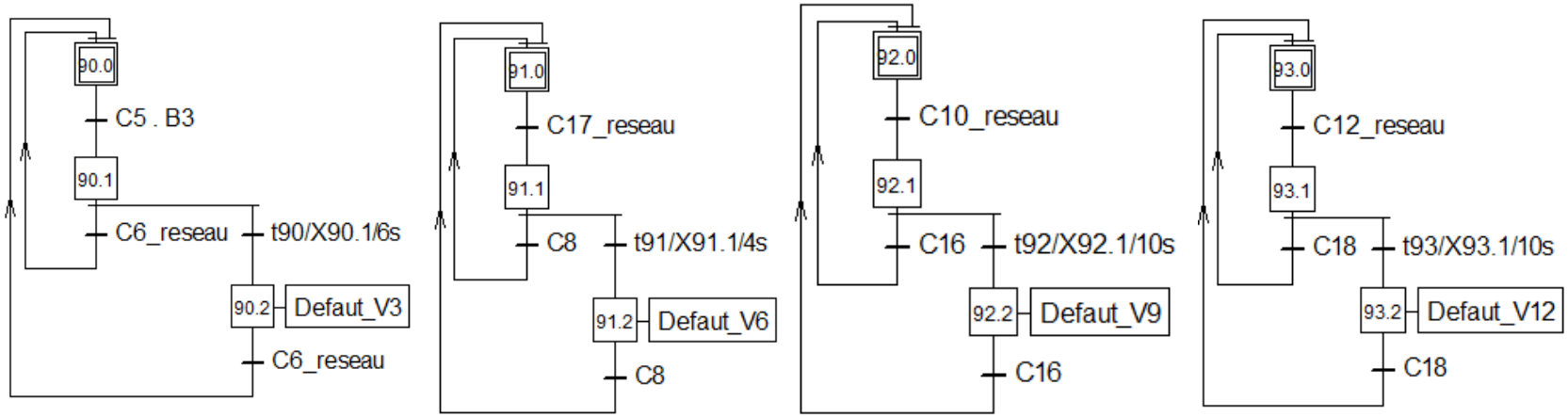
Bloc FB9 :



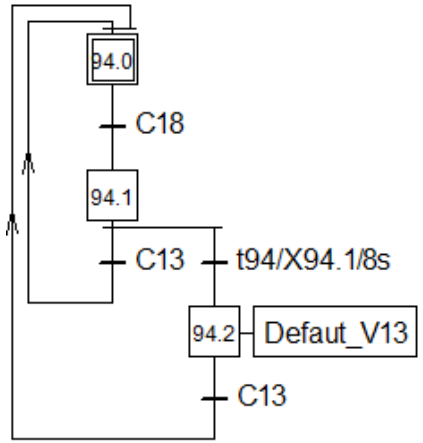
Bloc FB12 :

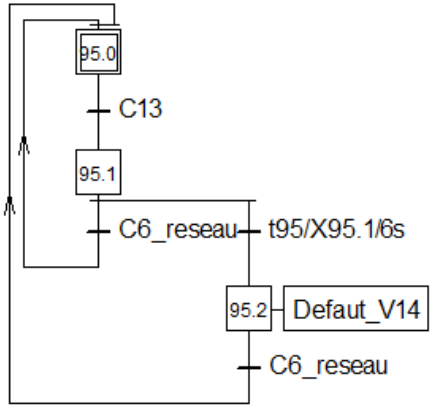


Bloc FB16 :

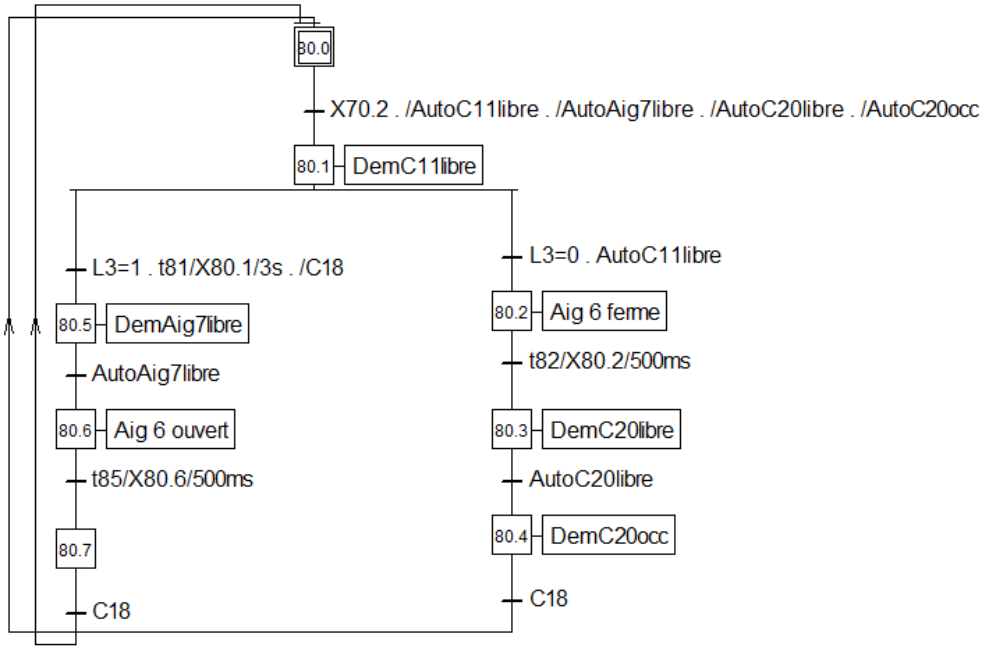


Bloc FB16 :

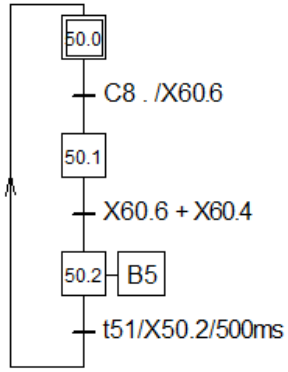




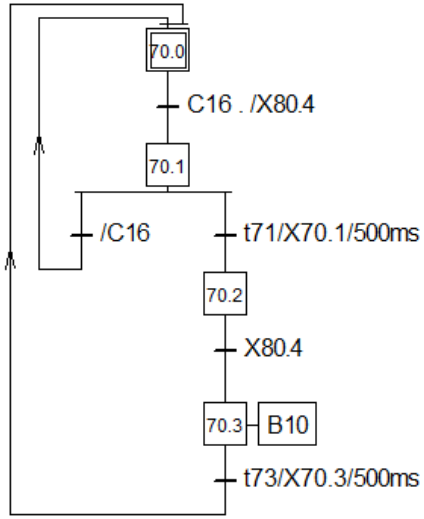
Bloc FB15 :



Bloc FB11 :

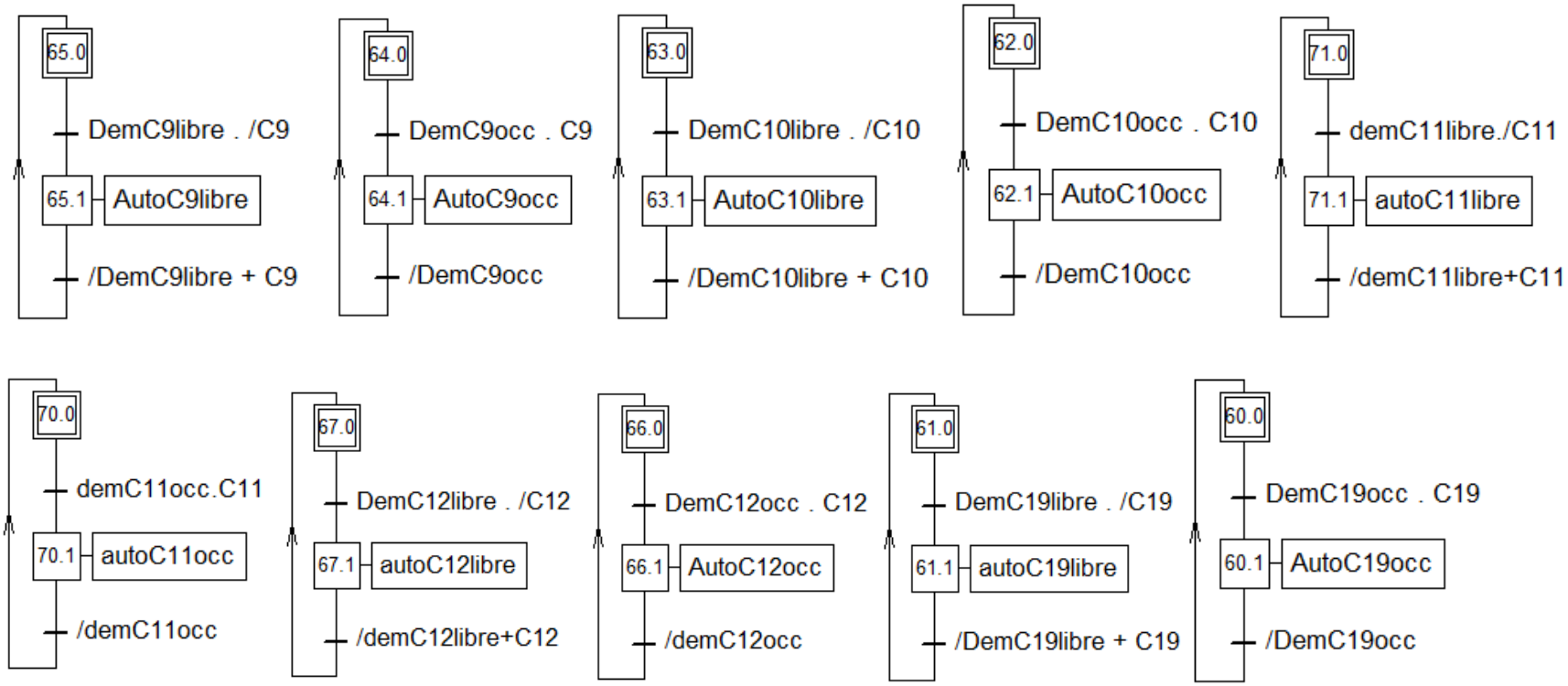


Bloc FB14 :

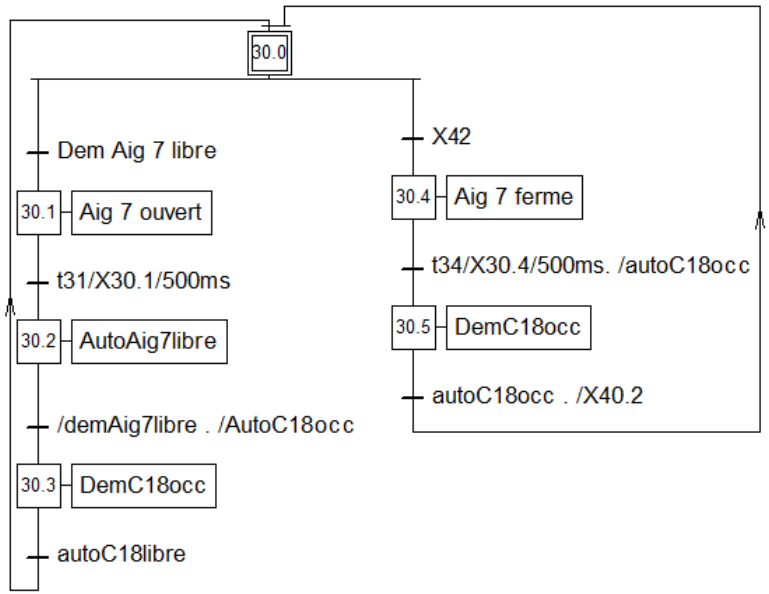


Annexe I.III GRAFCET de la Station 6 « Epis »

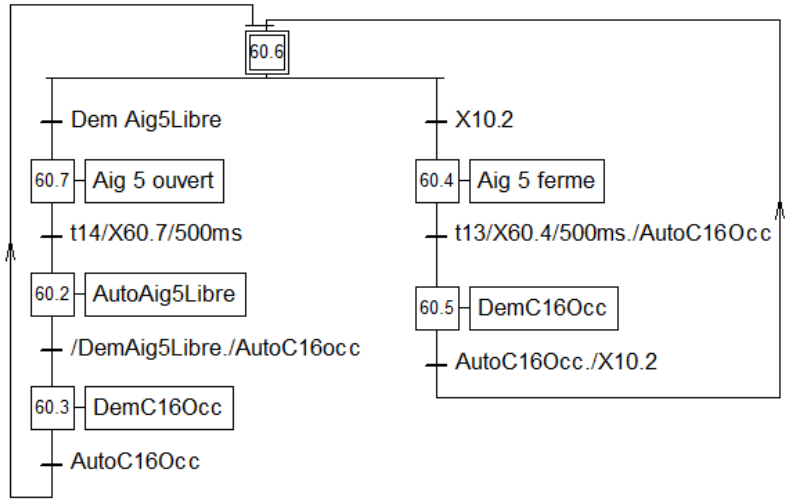
Bloc FB4 :



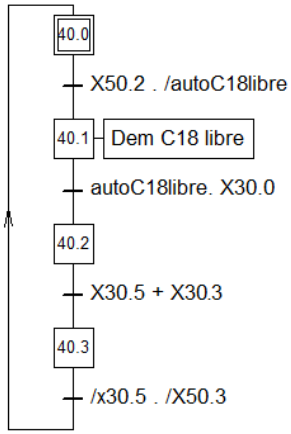
Bloc FB5 :



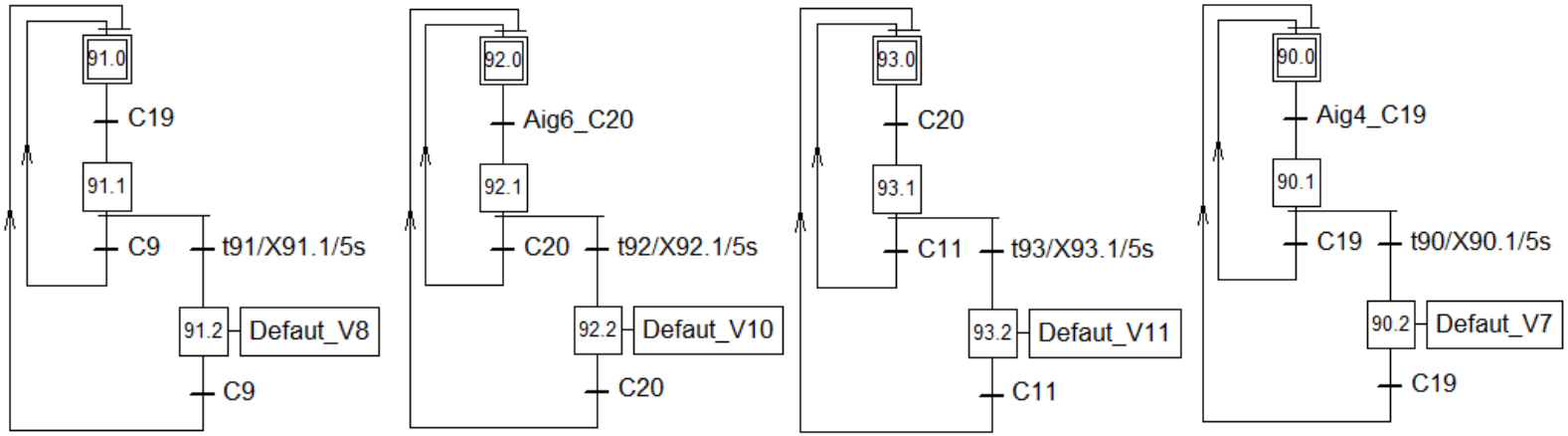
Bloc FB6 :



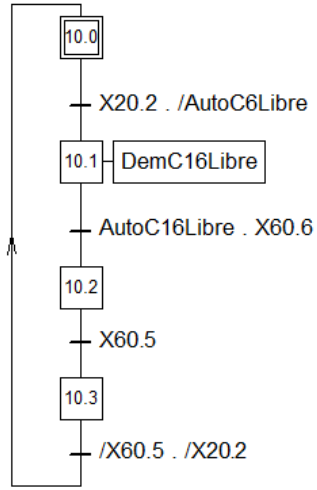
Bloc FB9 :



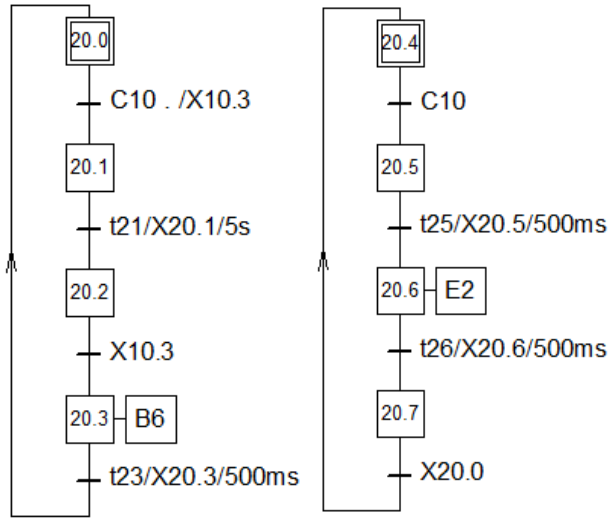
Bloc FB12 :



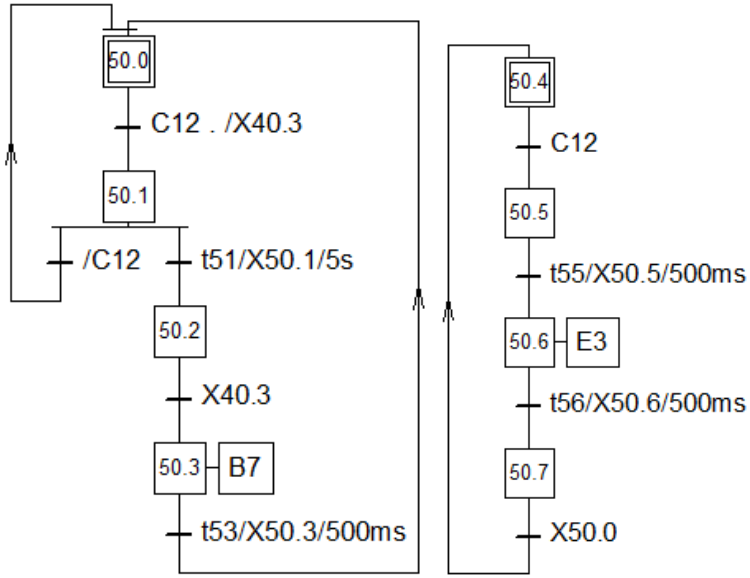
Bloc FB7 :



Bloc FB8 :



Bloc FB11 :



Annexe II. Programmation LIST

Annexe II.I LIST de la Station 4 « Hippodrome »

```
Bloc FB1 :
S       M       40.0
R       M       40.1
R       M       40.2
R       M       40.3
S       M       10.0
R       M       10.1
R       M       10.2
R       M       10.3
R       M       10.4
R       M       10.5
S       M       20.0
R       M       20.1
R       M       20.2
R       M       20.3
R       M       20.4
R       M       20.5
R       M       20.6
R       M       20.7
S       M       30.0
R       M       30.1
R       M       30.2
R       M       30.3
R       M       30.4
S       M       50.0
R       M       50.1
R       M       50.2
S       M       60.0
R       M       60.1
R       M       60.2
R       M       60.3
R       M       60.4
R       M       60.5
R       M       60.6
R       M       60.7
S       M       70.0
R       M       70.1
R       M       70.2
R       M       70.3
S       M       80.0
R       M       80.1
R       M       80.2
R       M       80.3
R       M       80.4
R       M       80.5
R       M       80.6
R       M       80.7
S       M       25.0
R       M       25.1
S       M       15.0
R       M       15.1
S       M       45.0
R       M       45.1
S       M       35.0
R       M       35.1
S       M       90.0
R       M       90.1
R       M       90.2
S       M       91.0
R       M       91.1
R       M       91.2
S       M       92.0
R       M       92.1
R       M       92.2
S       M       93.0
R       M       93.1
R       M       93.2
S       M       94.0
R       M       94.1
R       M       94.2
S       M       95.0
R       M       95.1
R       M       95.2
```

```
Bloc FB10:

// Mots réseaux
défauts
U       "C5"
=       "C5 reseau"
U       "C8"
=       "C8_reseau"
U       "select_manu"
= "Mode_Manu_reseau"
U       "C16"
=       "C16 reseau"
U       M       60.5
=       "Aig4_C19"
U       M       80.2
=       "Aig6_C20"

// Tempo défauts
U       M       90.1
L       S5T#6S
SE      T       90
U       M       91.1
L       S5T#4S
SE      T       91
U       M       92.1
L       S5T#10S
SE      T       92
U       M       93.1
L       S5T#10S
SE      T       93
U       M       94.1
L       S5T#8S
SE      T       94
U       M       95.1
L       S5T#6S
SE      T       95

# Gestion B10
U       M       70.3
=       "B10"
U       M       70.1
L       S5T#500MS
SE      T       71
U       M       70.3
L       S5T#500MS
SE      T       73

# Gestion aig 6
U       M       80.1
=       "DemC11libre"
U       M       80.5
=       "DemAig7libre"
U       M       80.6
=       "Aig6_ouvert"
U       M       80.2
=       "Aig6 ferme"
U       M       80.3
=       "DemC20libre"
U       M       80.4
=       "DemC20Occ"
U       M       80.1
L       S5T#500MS
SE      T       81
U       M       80.6
L       S5T#500MS
SE      T       85
U       M       80.2
L       S5T#500MS
SE      T       82
```

```
//FB5
U       M       50.2
=       "B5"
U       M       50.2
L       S5T#500MS
SE      T       51

// Aig 4
U       M       60.2
=       "DemC10libre"
=       "DemC10Occ"
=       "DemC9libre"
=       "DemC9Occ"
=       "DemC12libre"
=       "DemC12Occ"
U       M       60.3
=       "DemAig5libre"
U       M       60.4
=       "Aig4_ouvert"
U       M       60.5
=       "Aig4_ferme"
U       M       60.6
=       "DemC19libre"
U       M       60.7
=       "DemC19Occ"
U       M       60.1
L       S5T#500MS
SE      T       61
U       M       60.4
L       S5T#500MS
SE      T       64
U       M       60.5
L       S5T#500MS
SE      T       65

// Défauts voyant
U       M       90.2
O       M       91.2
O       M       92.2
O       M       93.2
=       A       12.2

// Mots reseaux
U       M       40.2
L       S5T#500MS
SE      T        2
U       M       40.2
=       "B3"
U       M       10.1
L       S5T#500MS
SE      T       11
U       M       10.5
L       S5T#500MS
SE      T       15
U       M       10.5
=       "B8"
U       M       20.2
L       S5T#500MS
SE      T       22
```

```
U(
O       M       20.1
O       M       20.4)
=       "DemC6libre"
U       M       20.2
=       "Aig1_ouvert"
U       M       20.5
=       "Aig8_ferme"
U       M       20.5
=       "Aig1 ferme"
U       M       20.7
=       "DemC6Occ"
U       M       30.1
=       "DemC14libre"
U       M       30.2
=       "Aig8 ouvert"
U       M       30.2
L       S5T#500MS
SE      T       32
U       M       30.4
=       "DemC14Occ"
U       M       20.3
L       S5T#500MS
SE      T       50
U       M       20.5
L       S5T#500MS
SE      T       25

# C16 occ
U       M       15.1
=       "AutoC16Occ"
# C16 libre
U       M       25.1
=       "AutoC16libre"
# C18 occ
U       M       35.1
=       "AutoC18Occ"
# C18 libre
U       M       45.1
=       "AutoC18libre"
```

```
Bloc FB11 :

U       M       50.0
U       "C8"
UN      M       60.6
S       M       50.1
R       M       50.0
U       M       50.1
U(
O       M       60.6
O       M       60.4)
S       M       50.2
R       M       50.1
U       M       50.2
U       T       51
S       M       50.0
R       M       50.2
```

```
Bloc FB12 :
U      M      60.0
U      M      50.1
UN    "AutoAig5libre"
UN     "AutoC10libre"
UN     "AutoC10Occ"
UN     "AutoC9libre"
UN     "AutoC9Occ"
UN     "AutoC12libre"
UN     "AutoC12Occ"
UN     "AutoC19libre"
UN     "AutoC19Occ"
S      M      60.1
R      M      60.0
U      M      60.1
UN     "C8"
S      M      60.0
R      M      60.1
U      M      60.1
U      T      61
S      M      60.2
R      M      60.1
U      M      60.2
U(
O      "L2"
O(
U      "AutoC10Occ"
U(
O      "AutoC9Occ"
O      "AutoC12Occ"
)
)
)
S      M      60.3
R      M      60.2
```

```
U      M      60.3
U     "AutoAig5libre"
S      M      60.4
R      M      60.3
U      M      60.4
U      T      64
UN     "C16"
S      M      61.0
R      M      60.4
U      M      61.0
U      "C16"
S      M      60.0
R      M      61.0
U      M      60.2
UN     "L2"
U(
O      "AutoC10libre"
O(
U      "AutoC9libre"
U      "AutoC12Occ"
)
)
S      M      60.5
R      M      60.2
U      M      60.5
U      T      65
S      M      60.6
R      M      60.5
U      M      60.6
U      "AutoC19libre"
S      M      60.7
R      M      60.6
U      M      60.7
U      "AutoC19Occ"
S      M      60.0
R      M      60.7
```

```
Bloc FB6 :
// Gestion B3
U      M      40.0
U      "C5"
S      M      40.3
R      M      40.0
U      M      40.3
U(
O      "FinRobot"
O      "select_manu")
UN     M      20.3
R      M      40.3
S      M      40.1
```

```
U      M      40.1
U      M      20.3
S      M      40.2
R      M      40.1
U      M      40.2
U      T       2
S      M      40.0
R      M      40.2
```

```
Bloc FB13 :
// C16 libre
U      M      25.0
U      "DemC16libre"
UN     "C16"
S      M      25.1
R      M      25.0
U      M      25.1
U(
O      "C16"
ON     "DemC16libre"
)
S      M      25.0
R      M      25.1

// C16 occ
U      M      15.0
U      "C16"
U      "DemC16Occ"
S      M      15.1
R      M      15.0
U      M      15.1
UN     "DemC16Occ"
S      M      15.0
R      M      15.1

// C18 libre
U      M      45.0
U      "DemC18libre"
UN     "C18"
S      M      45.1
R      M      45.0
U      M      45.1
U(
O      "C18"
ON     "DemC18libre"
)
S      M      45.0
R      M      45.1

// C18 occ
U      M      35.0
U      "C18"
U      "DemC18Occ"
S      M      35.1
R      M      35.0
U      M      35.1
UN     "DemC18Occ"
S      M      35.0
R      M      35.1
```

```
Bloc FB14 :
U      M      70.0
U      "C16"
UN     M      80.4
S      M      70.1
R      M      70.0
U      M      70.1
UN     "C16"
S      M      70.0
R      M      70.1
U      M      70.1
U      T      71
S      M      70.2
R      M      70.1
U      M      70.2
U(
O      M      80.4
O      M      80.7
)
S      M      70.3
R      M      70.2
U      M      70.3
U      T      73
S      M      70.0
R      M      70.3
```

```
Bloc FB15 :

U      M      80.0
U      M      70.2
UN     "AutoC11libre"
UN    "AutoAig7libre"
UN     "AutoC20libre"
UN     "AutoC20Occ"
S      M      80.1
R      M      80.0
U      M      80.1
U      "L3"
UN     "C18"
U      T      81
S      M      80.5
R      M      80.1
U      M      80.5
U     "AutoAig7libre"
S      M      80.6
R      M      80.5
```

```
Bloc FB9 :
U      M      30.0
UN     "AutoC14libre"
UN     M      20.4
UN     M      20.5
UN     M      20.6
UN     M      20.7
U      M      10.4
S      M      30.1
R      M      30.0
U      M      30.1
U      "AutoC14libre"
S      M      30.2
R      M      30.1
```

```
U      M      30.2
U      T      32
S      M      30.3
R      M      30.2
U      M      30.3
UN     M      10.4
S      M      30.4
R      M      30.3
U      M      30.4
U      "AutoC14Occ"
S      M      30.0
R      M      30.4
```

```
Bloc FB8 :
U      M      20.0
U      M      40.1
UN     "AutoC6libre"
UN     M      20.6
UN     "AutoC6Occ"
S      M      20.1
R      M      20.0
U      M      20.1
U      "AutoC6libre"
S      M      20.2
R      M      20.1
U      M      20.2
U      T      22
S      M      20.3
R      M      20.2
U      M      20.3
UN     M      40.1
U      T      50
S      M      20.7
R      M      20.3
```

```
U      M      20.0
U      M      10.3
U      M      30.0
UN     "AutoC6libre"
UN     "AutoC6Occ"
S      M      20.4
R      M      20.0
U      M      20.4
U      "AutoC6libre"
S      M      20.5
R      M      20.4
U      M      20.5
U      T      25
S      M      20.6
R      M      20.5
U      M      20.6
UN     M      10.3
S      M      20.7
R      M      20.6
U      M      20.7
U      "AutoC6Occ"
S      M      20.0
R      M      20.7
```

```
Bloc FB16 :
// Virage 6
U       M       91.0
U       "C17_reseau"
S       M       91.1
R       M       91.0
U       M       91.1
U       "C8"
S       M       91.0
R       M       91.1
U       M       91.1
U       T       91
S       M       91.2
R       M       91.1
U       M       91.2
U       "C8"
S       M       91.0
R       M       91.2

// Virage 3
U       M       90.0
U       "C5"
U       "B3"
S       M       90.1
R       M       90.0
U       M       90.1
U       "C6_reseau"
S       M       90.0
R       M       90.1
U       M       90.1
U       T       90
S       M       90.2
R       M       90.1
U       M       90.2
U       "C6_reseau"
S       M       90.0
R       M       90.2

// Virage 9
U       M       92.0
U       "C10_reseau"
S       M       92.1
R       M       92.0
U       M       92.1
U       "C16"
S       M       92.0
R       M       92.1
U       M       92.1
U       T       92
S       M       92.2
R       M       92.1
U       M       92.2
U       "C16"
S       M       92.0
R       M       92.2

// Virage 12
U       M       92.0
U       "C10_reseau"
S       M       92.1
R       M       92.0
U       M       92.1
U       "C16"
S       M       92.0
R       M       92.1
U       M       92.1
U       T       92
S       M       92.2
R       M       92.1
U       M       92.2
U       "C16"
S       M       92.0
R       M       92.2

// Virage 13
U       M       94.0
U       "C18"
S       M       94.1
R       M       94.0
U       M       94.1
U       "C13"
S       M       94.0
R       M       94.1
U       M       94.1
U       T       94
S       M       94.2
R       M       94.1
U       M       94.2
U       "C13"
S       M       94.0
R       M       94.2

// Virage 14
U       M       95.0
U       "C13"
U       "B8"
S       M       95.1
R       M       95.0
U       M       95.1
U       "C6_reseau"
S       M       95.0
R       M       95.1
U       M       95.1
U       T       95
S       M       95.2
R       M       95.1
U       M       95.2
U       "C6_reseau"
S       M       95.0
R       M       95.2
```

```
Bloc FB7 :

U       M       10.0
U       "C13"
S       M       10.1
R       M       10.0
U       M       10.1
UN      "C13"
S       M       10.0
R       M       10.1
U       M       10.1
U       T       11
S       M       10.2
R       M       10.1
U       M       10.2
U       "L4"
UN      M       20.6
S       M       10.3
R       M       10.2
U       M       10.2
UN      "L4"
UN      M       20.6
S       M       10.4
R       M       10.2
U       M       10.3
U       M       20.6
S       M       10.5
R       M       10.3
U       M       10.4
U       M       30.3
S       M       10.5
R       M       10.4
U       M       10.5
U       T       15
S       M       10.0
R       M       10.5
```

Annexe II.II LIST de la Station 5 « Magasin »

```
Bloc FB1 :

S       M       60.0
R       M       60.1
S       M       10.0
R       M       10.1
S       M       20.0
R       M       20.1
S       M       30.0
R       M       30.1
S       M       40.0
R       M       40.1
R       M       40.2
R       M       40.3
S       M       50.0
R       M       50.1
R       M       50.2
R       M       50.3
R       M       50.4
R       M       50.5
R       M       50.6
R       M       50.7
R       M       51.0
S       M       70.0
R       M       70.1
S       M       70.2
R       M       70.3
R       M       70.4
R       M       70.5
R       M       70.6
S       M       80.0
R       M       80.1
R       M       80.2

S       M       82.0
R       M       82.1
R       M       82.2
S       M       83.0
R       M       83.1
R       M       83.2
S       M       84.0
R       M       84.1
R       M       84.2
S       M       90.0
R       M       90.1
R       M       90.2
R       M       90.3
R       M       90.4
R       M       90.5
R       M       90.6
R       M       90.7
R       M       91.0
R       M       91.1

// Grafcet de gestion
des mots communs
S       "etape10"
R       "etape11"
S       "etape20"
R       "etape21"
S       M       102.2
```

```
Bloc FB7 :

// Gestion C3
U       M       70.2
U       "C3"
S       M       70.3
R       M       70.2
U       M       70.3
U       T       73
S       M       70.4
R       M       70.3
U       M       70.4
U       T       74
U(
O       M       91.1
O "Mode_Manu_reseau"
)
S       M       70.5
R       M       70.4

U       M       70.5
U       T       75
S       M       70.2
R       M       70.5

// Gestion C2
U       M       70.0
U       "C2"
S       M       70.1
R       M       70.0
U       M       70.1
U(
O       M       91.0
O "Mode_Manu_reseau"
)
S       M       70.6
R       M       70.1
U       M       70.6
U       T       71
S       M       70.0
R       M       70.6
```

```
Bloc FB5 :

U       M       40.0
U       "C15"
UN      "C17"
S       M       40.1
R       M       40.0
U       M       40.1
U       T       41
S       M       40.2
R       M       40.1

U       M       40.2
U       T       42
S       M       40.3
R       M       40.2
U       M       40.3
U       "C17"
S       M       40.0
R       M       40.3
```

Bloc FB4 :

```
U     M     60.0
U     "DemC6libre"
UN    "C6"
S     M     60.1
R     M     60.0
U     M     60.1
U(
O     "C6"
ON    "DemC6libre"
)
S     M     60.0
R     M     60.1
U     M     10.0
U     "C6"
U     "DemC6Occ"
S     M     10.1
R     M     10.0
U     M     10.1
UN    "DemC6Occ"
S     M     10.0
R     M     10.1
U     M     20.0
U     "C14"
U     "DemC14Occ"
S     M     20.1
R     M     20.0
U     M     20.1
UN    "DemC14Occ"
S     M     20.0
R     M     20.1
U     M     30.0
UN    "C14"
U     "DemC14libre"
S     M     30.1
R     M     30.0
U     M     30.1
U(
O     "C14"
ON    "DemC14libre"
)
S     M     30.0
R     M     30.1
```

Bloc FB6 :

```
U     M     50.0
U     "C7"
S     M     50.1
R     M     50.0
U     M     50.1
UN    "C7"
S     M     50.0
R     M     50.1
U     M     50.1
U     T     51
S     M     50.2
R     M     50.1
U     M     50.2
UN    "L1"
UN    "C17"
U     M     40.0
S     M     50.3
R     M     50.2
U     M     50.3
U     T     53
S     M     50.4
R     M     50.3
U     M     50.4
U     T     54
S     M     50.5
R     M     50.4
U     M     50.5
U     "C17"
S     M     50.0
R     M     50.5
U     M     50.2
U     "L1"
UN    "C2"
S     M     50.6
R     M     50.2
U     M     50.6
U     T     56
S     M     50.7
R     M     50.6
U     M     50.7
U     T     57
S     M     51.0
R     M     50.7
U     M     51.0
U     "C2"
S     M     50.0
R     M     51.0
```

Bloc FB8 :

```
// Virage 1
U     M     80.0
U     "C2"
U     "B1"
S     M     80.1
R     M     80.0
U     M     80.1
U     "C3"
S     M     80.0
R     M     80.1
U     M     80.1
U     T     80
S     M     80.2
R     M     80.1
U     M     80.2
U     "C3"
S     M     80.0
R     M     80.2

// Virage 4
U     M     82.0
U     M     50.6
S     M     82.1
R     M     82.0
U     M     82.1
U     "C2"
S     M     82.0
R     M     82.1
U     M     82.1
U     T     82
S     M     82.2
R     M     82.1
U     M     82.2
U     "C2"
S     M     82.0
R     M     82.2

// Virage 5
U     M     83.0
U     M     50.3
S     M     83.1
R     M     83.0
U     M     83.1
U     "C17"
S     M     83.0
R     M     83.1
U     M     83.1
U     T     83
S     M     83.2
R     M     83.1
U     M     83.2
U     "C17"
S     M     83.0
R     M     83.2

// Virage 2
U     M     84.0
U     "C3"
U     "B2"
S     M     84.1
R     M     84.0
U     M     84.1
U     "C5 reseau"
S     M     84.0
R     M     84.1
U     M     84.1
U     T     84
S     M     84.2
R     M     84.1
U     M     84.2
U     "C5_reseau"
S     M     84.0
R     M     84.2
```

Bloc FB10 :

```
// Mots réseau de
défauts
U     "C3"
=     "C3 reseau"
U     "C17"
=     "C17_reseau"
U     "C6"
=     "C6_reseau"

// B1 et B2
U     M     70.6
=     "B1"
U     M     70.4
=     "E1"
U     M     70.5
=     "B2"
U     M     70.6
L     S5T#500MS
SE    T     71
U     M     70.3
L     S5T#500MS
SE    T     73
U     M     70.4
L     S5T#500MS
SE    T     74
U     M     70.5
L     S5T#500MS
SE    T     75
U     M     90.5
L     S5T#300MS
SE    T     90
U     M     91.0
L     S5T#300MS
SE    T     91
U     M     91.1
L     S5T#300MS
SE    T     92
U     M     90.0
L     S5T#750MS
SE    T     93

// Capteurs
U     M     60.1
=     "AutoC6libre"
U     M     10.1
=     "AutoC6Occ"
U     M     20.1
=     "AutoC14Occ"
U     M     30.1
=     "AutoC14libre"
# B9
U     M     40.1
=     "Aig3_ouvert"
U     M     40.2
=     "B9"
U     M     40.1
L     S5T#500MS
SE    T     41
U     M     40.2
L     S5T#500MS
SE    T     42

# B4
U     M     50.1
L     S5T#500MS
SE    T     51
U     M     50.3
=     "Aig3 ferme"
U     M     50.3
=     "Aig2_ferme"
U(
O     M     50.4
O     M     50.7
)
=     "B4"
U     M     50.3
L     S5T#500MS
SE    T     53
U     M     50.4
L     S5T#500MS
SE    T     54
U     M     50.6
=     "Aig2 ouvert"
U     M     50.6
L     S5T#500MS
SE    T     56
U     M     50.7
L     S5T#500MS
SE    T     57
```

```
Bloc FB9 :

// Chargement
U     M     90.0
U     T     93
U     M     70.4
UN
"Mode_Manu_reseau"
U(
O(
U     "R 3"
UN    "R_2"
UN    "R_1"
)
O(
UN    "C2"
U     "R 1"
U     "R_3"
UN    "R_2"
)
)
S     M     90.1
R     M     90.0
U     M     90.1
U     "R_1"
UN    "R_2"
UN    "R 3"
S     M     90.2
R     M     90.1
U     M     90.2
UN    "R_1"
UN    "R_2"
U     "R 3"
S     M     91.1
R     M     90.2
U     M     91.1
U     M     70.2
U     T     92
S     M     90.0
R     M     91.1

// Usinage
U     M     90.0
U     T     93
U     "C5_reseau"
UN "Mode_Manu_reseau"
UN    "C3"
U(
O(
U     "R 3"
UN    "R_2"
UN    "R_1"
)
O(
UN    "C2"
U     "R 1"
U     "R_3"
UN    "R_2"
)
)
S     M     90.3
R     M     90.0
U     M     90.3
U     "R_2"
UN    "R_1"
UN    "R 3"
S     M     90.4
R     M     90.3
U     M     90.4
U     "R_3"
UN    "R_2"
UN    "R 1"
S     M     90.5
R     M     90.4
U     M     90.5
UN    "C5 reseau"
U     T     90
S     M     90.0
R     M     90.5

// Dechargement
U     M     90.0
U     T     93
U     M     70.1
UN "Mode_Manu_reseau"
U(
O(
U     "R 3"
U     "R 1"
UN    "R_2"
)
O(
UN    "C3"
UN    "C5 reseau"
U     "R 3"
UN    "R_2"
UN    "R_1"
)
)
S     M     90.6
R     M     90.0
U     M     90.6
U     "R_2"
U     "R_1"
UN    "R 3"
S     M     90.7
R     M     90.6
U     M     90.7
U     "R_1"
UN    "R_2"
U     "R 3"
S     M     91.0
R     M     90.7
U     M     91.0
U     M     70.0
U     T     91
S     M     90.0
R     M     91.0
```

Annexe II.III LIST de la Station 6 « Epis »

```
Bloc FB 1 :

S     M     30.0
R     M     30.1
R     M     30.2
R     M     30.3
R     M     30.4
R     M     30.5
S     M     40.0
R     M     40.1
R     M     40.2
R     M     40.3
S     M     50.0
R     M     50.1
R     M     50.2
R     M     50.3
S     M     50.4
R     M     50.5
R     M     50.6
S     M     60.6
R     M     60.7
R     M     60.2
R     M     60.3
R     M     60.4
R     M     60.5

S     M     10.0
R     M     10.1
R     M     10.2
R     M     10.3
S     M     20.0
R     M     20.1
R     M     20.2
R     M     20.3
S     M     20.4
R     M     20.5
R     M     20.6
R     M     20.7
S     M     60.0
R     M     60.1
S     M     61.0
R     M     61.1
S     M     62.0
R     M     62.1
S     M     63.0
R     M     63.1
S     M     64.0
R     M     64.1
S     M     65.0
R     M     65.1

S     M     66.0
R     M     66.1
S     M     67.0
R     M     67.1
S     M     68.0
R     M     68.1
S     M     69.0
R     M     69.1
S     M     70.0
R     M     70.1
S     M     71.0
R     M     71.1
S     M     90.0
R     M     90.1
R     M     90.2
S     M     91.0
R     M     91.1
R     M     91.2
S     M     92.0
R     M     92.1
R     M     92.2
S     M     93.0
R     M     93.1
R     M     93.2
```

```
Bloc FB7 :

U     M     10.0
U     M     20.2
UN    "AutoC16libre"
S     M     10.1
R     M     10.0
U     M     10.1
U     M     60.6
U     "AutoC16libre"
S     M     10.2
R     M     10.1
U     M     10.2
U     M     60.5
S     M     10.3
R     M     10.2
U     M     10.3
UN    M     60.5
UN    M     20.2
S     M     10.0
R     M     10.3
```

```
Bloc FB10 :

// Mots réseaux
U       "C10"
=       "C10_reseau"
U       "C12"
=       "C12_reseau"

# Aig 5 libre
U       M       60.7
=       "Aig5_ouvert"
U       M       60.2
=     "AutoAig5libre"
U(
O       M       60.3
O       M       60.5
)
=       "DemC16Occ"
U       M       60.4
=       "Aig5 ferme"
U       M       60.7
L       S5T#500MS
SE      T       14
U       M       60.4
L       S5T#500MS
SE      T       13

# Gestion aig 5
U       M       10.1
=       "DemC16libre"

# Gestion B6
U       M       20.3
=       "B6"
U       M       20.6
=       "E2"
U       M       20.1
L       S5T#5S
SE      T       21
U       M       20.3
L       S5T#500MS
SE      T       23
U       M       20.5
L       S5T#500MS
SE      T       25
U       M       20.6
L       S5T#500MS
SE      T       26

# C19 Occ
U       M       60.1
=       "AutoC19Occ"

# C19 libre
U       M       61.1
=       "AutoC19libre"

# C10 Occ
U       M       62.1
=       "AutoC10Occ"

# C10 libre
U       M       63.1
=       "AutoC10libre"

# C9 Occ
U       M       64.1
=       "AutoC9Occ"

# C9 libre
U       M       65.1
=       "AutoC9libre"

# C12 Occ
U       M       66.1
=       "AutoC12Occ"

# C12 libre
U       M       67.1
=       "AutoC12libre"

# C20 Occ
U       M       68.1
=       "AutoC20Occ"

# C20 libre
U       M       69.1
=       "AutoC20libre"

# C11 Occ
U       M       70.1
=       "AutoC11Occ"

# C11 libre
U       M       71.1
=       "AutoC11libre"

// Aig 7 libre
U       M       30.1
=       "Aig7_ouvert"
U       M       30.2
=     "AutoAig7libre"
U(
O       M       30.3
O       M       30.5
)
=       "DemC18Occ"
U       M       30.4
=       "Aig7 ferme"
U       M       30.1
L       S5T#500MS
SE      T       31
U       M       30.4
L       S5T#500MS
SE      T       34

# Gestion aig 7
U       M       40.1
=       "DemC18libre"

# Gestion B7
U       M       50.3
=       "B7"
U       M       50.6
=       "E3"
U       M       50.1
L       S5T#5S
SE      T       51
U       M       50.3
L       S5T#500MS
SE      T       53
U       M       50.5
L       S5T#500MS
SE      T       55
U       M       50.6
L       S5T#500MS
SE      T       56
```

```
Bloc FB4 :

# C19 occ
U       M       60.0
U       "C19"
U       "DemC19Occ"
S       M       60.1
R       M       60.0
U       M       60.1
UN      "DemC19Occ"
S       M       60.0
R       M       60.1

# C19 libre
U       M       61.0
UN      "C19"
U       "DemC19libre"
S       M       61.1
R       M       61.0
U       M       61.1
U(
O       "C19"
ON      "DemC19libre"
)
S       M       61.0
R       M       61.1

# C10 occ
U       M       62.0
U       "C10"
U       "DemC10Occ"
S       M       62.1
R       M       62.0
U       M       62.1
UN      "DemC10Occ"
S       M       62.0
R       M       62.1

# C10 libre
U       M       63.0
UN      "C10"
U       "DemC10libre"
S       M       63.1
R       M       63.0
U       M       63.1
U(
O       "C10"
ON      "DemC10libre"
)
S       M       63.0
R       M       63.1

# C9 occ
U       M       64.0
U       "C9"
U       "DemC9Occ"
S       M       64.1
R       M       64.0
U       M       64.1
UN      "DemC9Occ"
S       M       64.0
R       M       64.1

# C9 libre
U       M       65.0
UN      "C9"
U       "DemC9libre"
S       M       65.1
R       M       65.0
U       M       65.1
U(
O       "C9"
ON      "DemC9libre"
)
S       M       65.0
R       M       65.1

# C12 occ
U       M       66.0
U       "C12"
U       "DemC12Occ"
S       M       66.1
R       M       66.0
U       M       66.1
UN      "DemC12Occ"
S       M       66.0
R       M       66.1

# C12 libre
U       M       67.0
UN      "C12"
U       "DemC12libre"
S       M       67.1
R       M       67.0
U       M       67.1
U(
O       "C12"
ON      "DemC12libre"
)
S       M       67.0
R       M       67.1

# C20 occ
U       M       68.0
U       "C20"
U       "DemC20Occ"
S       M       68.1
R       M       68.0
U       M       68.1
UN      "DemC20Occ"
S       M       68.0
R       M       68.1

# C20 libre
U       M       69.0
UN      "C20"
U       "DemC20libre"
S       M       69.1
R       M       69.0
U       M       69.1
U(
O       "C20"
ON      "DemC20libre"
)
S       M       69.0
R       M       69.1

# C11 occ
U       M       70.0
U       "C11"
U       "DemC11Occ"
S       M       70.1
R       M       70.0
U       M       70.1
UN      "DemC11Occ"
S       M       70.0
R       M       70.1

# C11 libre
U       M       71.0
UN      "C11"
U       "DemC11libre"
S       M       71.1
R       M       71.0
U       M       71.1
U(
O       "C11"
ON      "DemC11libre"
)
S       M       71.0
R       M       71.1
```

```
Bloc FB11 :

U       M       50.0
U       "C12"
UN      M       40.3
S       M       50.1
R       M       50.0
U       M       50.1
UN      "C12"
S       M       50.0
R       M       50.1
U       M       50.1
U       T       51
S       M       50.2
R       M       50.1
U       M       50.2
U       M       40.3
S       M       50.3
R       M       50.2
U       M       50.3
U       T       53
S       M       50.0
R       M       50.3

U       M       50.4
U       "C12"
S       M       50.5
R       M       50.4
U       M       50.5
U       T       55
S       M       50.6
R       M       50.5
U       M       50.6
U       T       56
S       M       50.7
R       M       50.6
U       M       50.7
U       M       50.0
S       M       50.4
R       M       50.7
```

```
Bloc FB6 :

U       M       60.6
U       "DemAig5libre"
U       M       10.0
S       M       60.7
R       M       60.6
U       M       60.7
U       T       14
S       M       60.2
R       M       60.7
U       M       60.2
UN      "DemAig5libre"
UN      "AutoC16Occ"
S       M       60.3
R       M       60.2
U       M       60.3
U       "AutoC16Occ"
S       M       60.6
R       M       60.3
U       M       60.6
U       M       10.2
S       M       60.4
R       M       60.6
U       M       60.4
U       T       13
UN      "AutoC16Occ"
S       M       60.5
R       M       60.4
U       M       60.5
U       "AutoC16Occ"
UN      M       10.2
S       M       60.6
R       M       60.5
```

```
Bloc FB12 :

// Virage 7
U       M       90.0
U       "Aig4_C19"
S       M       90.1
R       M       90.0
U       M       90.1
U       "C19"
S       M       90.0
R       M       90.1
U       M       90.1
U       T       90
S       M       90.2
R       M       90.1
U       M       90.2
U       "C19"
S       M       90.0
R       M       90.2

// Virage 8
U       M       91.0
U       "C19"
S       M       91.1
R       M       91.0
U       M       91.1
U       "C9"
S       M       91.0
R       M       91.1
U       M       91.1
U       T       91
S       M       91.2
R       M       91.1
U       M       91.2
U       "C9"
S       M       91.0
R       M       91.2

// Virage 10
U       M       92.0
U       "Aig6 C20"
S       M       92.1
R       M       92.0
U       M       92.1
U       "C20"
S       M       92.0
R       M       92.1
U       M       92.1
U       T       92
S       M       92.2
R       M       92.1
U       M       92.2
U       "C20"
S       M       92.0
R       M       92.2

// Virage 11
U       M       93.0
U       "C20"
S       M       93.1
R       M       93.0
U       M       93.1
U       "C11"
S       M       93.0
R       M       93.1
U       M       93.1
U       T       93
S       M       93.2
R       M       93.1
U       M       93.2
U       "C11"
S       M       93.0
R       M       93.2
```

```
Bloc FB5 :

U       M       30.0
U       "DemAig7libre"
S       M       30.1
R       M       30.0
U       M       30.1
U       T       31
S       M       30.2
R       M       30.1
U       M       30.2
UN      "DemAig7libre"
UN      "AutoC18Occ"
S       M       30.3
R       M       30.2
U       M       30.3
U       "AutoC18Occ"
S       M       30.0
R       M       30.3
U       M       30.0
U       M       40.2
S       M       30.4
R       M       30.0
U       M       30.4
U       T       34
UN      "AutoC18Occ"
S       M       30.5
R       M       30.4
U       M       30.5
U       "AutoC18Occ"
UN      M       40.2
S       M       30.0
R       M       30.5
```

```
Bloc FB8 :

U       M       20.0
U       "C10"
UN      M       10.3
S       M       20.1
R       M       20.0
U       M       20.1
U       T       21
S       M       20.2
R       M       20.1
U       M       20.2
U       M       10.3
S       M       20.3
R       M       20.2
U       M       20.3
U       T       23
S       M       20.0
R       M       20.3
U       M       20.4
U       "C10"
S       M       20.5
R       M       20.4
U       M       20.5
U       T       25
S       M       20.6
R       M       20.5
U       M       20.6
U       T       26
S       M       20.7
R       M       20.6
U       M       20.7
U       M       20.0
S       M       20.4
R       M       20.7
```

```
Bloc FB9 :

U       M       40.0
U       M       50.2
UN      "AutoC18libre"
S       M       40.1
R       M       40.0
U       M       40.1
U       M       30.0
U       "AutoC18libre"
S       M       40.2
R       M       40.1
U       M       40.2
U(
O       M       30.5
O       M       30.3
)
S       M       40.3
R       M       40.2
U       M       40.3
UN      M       30.5
UN      M       50.3
S       M       40.0
R       M       40.3
```

Annexe III. Programmes de la supervision et du robot

Annexe III.I Programme en langage C

```
#include <winsock2.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "Applicom\Applicom.h"
#include "MySQL\mysql.h"
#include <conio.h>
#include <time.h>

long int SystemTime();
long int initTime;
short int  status;                    /* Status */

int main()
{
    int iOctet = 0;
    initTime=time(NULL);    // heure  courante current time
    short int nchan ;      /* Numéro de canal */
    short int nb ;                    /* Nombre de variables */   //32+32 dans une seule API
    short int neq ;                     /* Numéro d'équipement */
    short int  status;                /* Status */
    long adr = 3;                     /* Adresse de la première variable */
    short int tabl[64];               /* Table recevant les données */
    short int tablbit[64];            /* Table contenant les valeurs éclatées */
    short int c=0;
    ///New measures of the sensors
    short int  New_measure_sensor[32];
    short int  Old_measure_sensor[32];
    ///New measures default
    short int  New_turn_default[32];
    short int  Old_turn_default[32];
    ///Counter default
    long int  Time_default[16];
    int  flagCounter[16];
    int  TimeAPI[16];
    int  NbDefault[16];
    long int  Counter_default[16];
    long int  Counter_average[16];
    ///Mode
    short int  New_ARU=0;
    short int  Old_ARU=0;

    ///SQL Server
    char requete[512];
    MYSQL mysql;
    mysql_init(&mysql);
    mysql_options(&mysql,MYSQL_READ_DEFAULT_GROUP,"option");

    ///Initialize the variables
    for(c=0;c<=20;c++)
    {
        New_measure_sensor[c]=0;
        Old_measure_sensor[c]=0;

        //Initialize SQL Server
        MYSQL mysql;
        mysql_init(&mysql);
        mysql_options(&mysql,MYSQL_READ_DEFAULT_GROUP,"option");
        if(mysql_real_connect(&mysql,"127.0.0.1","production","production","projet",0,NULL,0))
        {
            if(c < 10){
                sprintf(requete,"UPDATE `sensors` SET `etat` =0 WHERE `nom` = 'C0%d'", c);
            }
            else{
                sprintf(requete,"UPDATE `sensors` SET `etat` =0 WHERE `nom` = 'C%d'", c);
            }
            mysql_query(&mysql,requete);
        }
        else
        {
            printf("Une erreur s'est produite lors de la connexion à la BDD (pour l'initialisation des capteurs)!");
        }
        mysql_close(&mysql);        //deallocates the connection handle pointed to by mysql
                                    //(if the handle was allocated automatically by mysql_init() or mysql_connect())
}
```

```
///Intialize Default Counter
for(c=0;c<=14;c++) {
    New_turn_default[c]=0;
    Old_turn_default[c]=0;
    Time_default[c]=0;
    flagCounter[c]=0;
    NbDefault[c]=0;
    Counter_default[c]=0;
    Counter_average[c]=0;

    //Initialize SQL Server
    MYSQL mysql;
    mysql_init(&mysql);
    mysql_options(&mysql,MYSQL_READ_DEFAULT_GROUP,"option");
    if(mysql_real_connect(&mysql,"127.0.0.1","production","production","projet",0,NULL,0)) {
        if(c < 10){
            sprintf(requete,"UPDATE `defaults` SET `etat`=0,`duree`='00:00:00', `nb_defaut`=0,
            `temps`='00:00:00' WHERE `nom` = 'V0%d'", c);
        }
        else{
            sprintf(requete,"UPDATE `defaults` SET `etat`=0,`duree`='00:00:00', `nb_defaut`=0,
            `temps`='00:00:00' WHERE `nom` = 'V%d'", c);
        }
        mysql_query(&mysql,requete);
    }
    else {
        printf("Une erreur s'est produite lors de la connexion à la BDD (pour l'initialisation des virages)!");
    }
    mysql_close(&mysql);
}

///Intialize time detection Default API
TimeAPI[1]=7;
TimeAPI[2]=9;
TimeAPI[3]=6;
TimeAPI[4]=4;
TimeAPI[5]=6;
TimeAPI[6]=4;
TimeAPI[7]=5;
TimeAPI[8]=5;
TimeAPI[9]=10;
TimeAPI[10]=5;
TimeAPI[11]=5;
TimeAPI[12]=10;
TimeAPI[13]=8;
TimeAPI[14]=6;
```

```
/// Main loop ///
    do
    {

        /// * * * * * * * * * * * * * * * * * * * * * ///
        /// Read the information of all the sensors ///
        /// * * * * * * * * * * * * * * * * * * * * * ///
        initbus(&status);
        nchan = 0;
        nb = 64;
        /// ***** HIPPODROME ***** ///
        //printf("\nHippodorme");

        /*Bits entrées*/
        iOctet = 0;
        neq = 4;
        adr = iOctet*8;
        readpackibit(&nchan, &neq, &nb, &adr, tabl, &status);
        if (!status) {
            transwordbit(&nb, tabl, tablbit, &status);
        }
        else{
            printf("Problème sur l'équipement %d",neq);
        }

        //*tablbit = ReadAPI(0, 4);

        New_measure_sensor[5]=tablbit[0*8+3];      //E0.3
        New_measure_sensor[8]=tablbit[0*8+2];      //E0.2
        New_measure_sensor[13]=tablbit[0*8+5];     //E0.5
        New_measure_sensor[16]=tablbit[0*8+6];     //E0.6
        New_measure_sensor[18]=tablbit[1*8+3];     //E1.3

        /*Bits mémoires*/
        iOctet = 90;
        neq = 4;
        adr = iOctet*8;
        readpackbit(&nchan, &neq, &nb, &adr, tabl, &status);
        if (!status) {
            transwordbit(&nb, tabl, tablbit, &status);
        }
        else {
            printf("Problème sur l'équipement %d",neq);
        }
```

```
New_turn_default[3]=tablbit[0*8+2];   //M90.2
New_turn_default[6]=tablbit[1*8+2];   //M91.2
New_turn_default[9]=tablbit[2*8+2];   //M92.2
New_turn_default[12]=tablbit[3*8+2];  //M93.2
New_turn_default[13]=tablbit[4*8+2];  //M94.2
New_turn_default[14]=tablbit[5*8+2];  //M95.2

/// ***** MAGASIN ***** ///
//printf("\nMagasin");

/*Bits entrées*/
iOctet = 0;
neq = 5;
adr = iOctet*8;
readpackibit(  &nchan,&neq,&nb,&adr, tabl,&status);
if (!status) {
   transwordbit( &nb,tabl,tablbit, &status);
}
Else {
   printf("Problème sur l'équipement %d",neq);
}
New_measure_sensor[1]=tablbit[0*8+1];      //C1
New_measure_sensor[2]=tablbit[0*8+2];      //C2
New_measure_sensor[3]=tablbit[0*8+3];      //C3
New_measure_sensor[4]=tablbit[0*8+4];      //C4
New_measure_sensor[6]=tablbit[0*8+6];      //C6
New_measure_sensor[7]=tablbit[0*8+7];      //C7
New_measure_sensor[14]=tablbit[1*8+4];     //C14
New_measure_sensor[15]=tablbit[1*8+5];     //C15
New_measure_sensor[17]=tablbit[1*8+7];     //C17

/*Bits mémoires*/
iOctet = 80;
neq = 5;
adr = iOctet*8;
readpackbit(&nchan, &neq, &nb, &adr, tabl, &status);
if (!status) {
   transwordbit(&nb, tabl, tablbit, &status);
}
else {
   printf("Problème sur l'équipement %d",neq);
}
New_turn_default[1]=tablbit[0*8+2];   //M80.2
New_turn_default[2]=tablbit[4*8+2];   //M84.2
New_turn_default[4]=tablbit[2*8+2];   //M82.2
New_turn_default[5]=tablbit[3*8+2];   //M83.2

/// ***** EPIS ***** ///
//printf("\nEpis");

/*Bits entrées*/
iOctet = 0;
neq = 6;
adr = iOctet*8;
readpackibit(&nchan, &neq, &nb, &adr, tabl, &status);
if (!status) {
   transwordbit(&nb, tabl, tablbit, &status);
}
else{
   printf("Problème sur l'équipement %d",neq);
}
New_measure_sensor[9]=tablbit[0*8+0];      //C9
New_measure_sensor[10]=tablbit[0*8+1];     //C10
New_measure_sensor[11]=tablbit[0*8+2];     //C11
New_measure_sensor[12]=tablbit[0*8+3];     //C12
New_measure_sensor[19]=tablbit[0*8+4];     //C19
New_measure_sensor[20]=tablbit[0*8+5];     //C20

/*Bits mémoires*/
iOctet = 90;
neq = 6;
adr = iOctet*8;
readpackbit(&nchan, &neq, &nb, &adr, tabl, &status);
if (!status) {
   transwordbit(&nb, tabl, tablbit, &status);
}
else {
   printf("Problème sur l'équipement %d",neq);
}
New_turn_default[7]=tablbit[0*8+2];   //M90.2
New_turn_default[8]=tablbit[1*8+2];   //M91.2
New_turn_default[10]=tablbit[2*8+2];  //M92.2
New_turn_default[11]=tablbit[3*8+2];  //M93.2
```

```
/// Default Calcul ///
for( c=1;c<15;c++){
   //printf("\nDef%d %d - Flag %d",c,New_turn_default[c],flagCounter[1]);
   if(New_turn_default[c] && !flagCounter[c]) {
      flagCounter[c]=1;
      Time_default[c]=SystemTime();
      printf("\nDefault %d",c);
   }
   if(!New_turn_default[c] && flagCounter[c]) {
      flagCounter[c]=0;
      Time_default[c]=SystemTime()-Time_default[c]+TimeAPI[c];
      NbDefault[c] ++;
      Counter_default[c] += Time_default[c];
      Counter_average[c] = Counter_default[c] / NbDefault[c];
      printf("\tTime : %d s\t",Time_default[c]);
   }
}



/// * * * * * * ///
///  SQL Server ///
/// * * * * * * ///


MYSQL mysql;
mysql_init(&mysql);
mysql_options(&mysql,MYSQL_READ_DEFAULT_GROUP,"option");

///Si la connexion réussie...
if(mysql_real_connect(&mysql,"127.0.0.1","production","production","projet",0,NULL,0)) {
   printf("\n\tCAPTEURS\n\n");
   for( c=1;c<=20;c++){
      ///Detect a new event Sensor
      printf("C%d : %d\n",c,New_measure_sensor[c]);
      if((Old_measure_sensor[c]==0)&&(New_measure_sensor[c]==1)) {
         if(c < 10){
            sprintf(requete,"UPDATE `sensors` SET `etat`=1 WHERE `nom` = 'C0%d'", c);
         }
         else{
            sprintf(requete,"UPDATE `sensors` SET `etat`=1 WHERE `nom` = 'C%d'", c);
         }
         mysql_query(&mysql,requete);
      }
      if((Old_measure_sensor[c]==1)&&(New_measure_sensor[c]==0)) {
         if(c < 10){
            sprintf(requete,"UPDATE `sensors` SET `etat`=0 WHERE `nom` = 'C0%d'", c);
         }
         else{
            sprintf(requete,"UPDATE `sensors` SET `etat`=0 WHERE `nom` = 'C%d'", c);
         }
         mysql_query(&mysql,requete);
      }
      Old_measure_sensor[c]=New_measure_sensor[c];
   }

   printf("\n\n\tVIRAGES\n\n");
   for( c=1;c<=14;c++) {
      printf("V%d : %d - time:%d - nb:%d - moyenne:%d\n",c, New_turn_default[c],
      Time_default[c],NbDefault[c],Counter_average[c]);
      if((Old_turn_default[c]==0)&&(New_turn_default[c]==1)) {
         if(c < 10){
            sprintf(requete,"UPDATE `defaults` SET `etat`=1 WHERE `nom` = 'V0%d'", c);
         }
         else{
            sprintf(requete,"UPDATE `defaults` SET `etat`=1 WHERE `nom` = 'V%d'", c);
         }
         mysql_query(&mysql,requete);
      }
      if((Old_turn_default[c]==1)&&(New_turn_default[c]==0)) {
         if(c < 10){
            sprintf(requete,"UPDATE `defaults` SET `etat`=0,`duree`='%d:%d:%d',
            `nb_defaut`=%d, `temps`='%d:%d:%d' WHERE `nom` = 'V0%d'",
            Time_default[c]/3600, (Time_default[c]%3600)/60, Time_default[c]%60, NbDefault[c],
            Counter_average[c]/3600, (Counter_average[c]%3600)/60,
            Counter_average[c]%60,c);
         }
         else{
            sprintf(requete,"UPDATE `defaults` SET `etat`=0,`duree`='%d:%d:%d',
            `nb_defaut`=%d, `temps`='%d:%d:%d' WHERE `nom` = 'V%d'",
            Time_default[c]/3600, (Time_default[c]%3600)/60, Time_default[c]%60, NbDefault[c],
            Counter_average[c]/3600, (Counter_average[c]%3600)/60,
            Counter_average[c]%60,c);
         }
         mysql_query(&mysql,requete);
      }
      Old_turn_default[c]=New_turn_default[c];
   }
```

```
printf("\n\n\tMODE\n\n");
        printf("ARU : %d\n",New_ARU);
        if((Old_ARU==0)&&(New_ARU==1)) {
           sprintf(requete,"UPDATE `mode` SET `etat`=0 WHERE `nom` = 'ARU'");
           mysql_query(&mysql,requete);
        }
        if((Old_ARU==1)&&(New_ARU==0)) {
           sprintf(requete,"UPDATE `mode` SET `etat`=1 WHERE `nom` = 'ARU'");
           mysql_query(&mysql,requete);
        }

        mysql_close(&mysql);
    }
    else
    {
        printf("Une erreur s'est produite lors de la connexion à la BDD!");
        mysql_close(&mysql);
    }

  }while(1);

  return (0);
}

long int SystemTime()
{
  ///Get the current system time, in seconds

  long int finalTime=time(NULL);
  long int resultTime;
  resultTime=finalTime-initTime;
  return resultTime;
}
```

Annexe III.II Programme interface web

Programme include_BDD.php :

```
<?php
$bdd = new PDO('mysql:host=localhost;dbname=projet', 'supervision', 'supervision');
?>
```

Programme description.php:

```
<HTML>
        <HEAD>
                <!-- Librairie JQuery -->
                <script type="text/javascript" src="jquery-1.12.1.min.js"></script>
                <!-- Raffraichissement de la bdd -->
                <script type="text/javascript">

                        $(document).ready(function(){
                          refreshTable();
                        });

                        function refreshTable(){
                                $('#tableBDD').load('TableDescription.php', function(){
                                  setTimeout(refreshTable, 1000);
                                });
                        }
                </script>
        </HEAD>
        <BODY>

                <?php require ("header.php"); ?>
                <DIV id="tableBDD"></DIV>

                <div id="home">
                        <a href="supervision.php"><img src="home.png"/></a>
                </div>

                <?php require ("footer.php"); ?>

        </BODY>
</HTML>
```

Programme TableDescription.php:

```
<?php
	// Connexion à la base de données
	try	{
		require("include_bdd.php");
	}
	catch(Exception $e){
		die('Erreur : '.$e->getMessage());
	}
?>
<div id="tableaux">
	<TABLE>
		<TR>
			<TD width=40></TD>
		</TR>
	</TABLE>
</div>
<div id="tableaux">
	<div id="left">
		<TABLE align=center id="colonne">
			<TR align=center>
				<TD colspan=5 id="titre">DEFAUTS</TD>
			</TR>
			<TR>
				<TD height=20></TD>
			</TR>
			<TR align=center style="font-weight:bold">
				<TD width=80> Nom </TD>
				<TD width=80> Etat </TD>
				<TD width=80> Duree du defaut </TD>
				<TD width=80> Nombre de defauts </TD>
				<TD width=80> Moyenne des defauts </TD>
			</TR>
			<TR>
				<TD height=10></TD>
			</TR>
			<?php
				$reqDefauts = $bdd->query('SELECT * FROM defaults
				ORDER BY nom');
				// requete sql pour afficher les élements de la table defauts

				while($donneeDefauts = $reqDefauts->fetch()){
				// boucle qui parcours la table tant qu'il y a des élements
					echo "<TR>

<TD align=center>",$donneeDefauts['nom'],"</TD>
```

```
                                                <TD align=center>",$donneeDefauts['etat'],"</TD>
                                                <TD align=center>",$donneeDefauts['duree'],"</TD>
                                                <TD align=center>",$donneeDefauts['nb_defaut'],"</TD>
                                                <TD align=center>",$donneeDefauts['temps'],"</TD>
                                        <TR>";                          // affichage du nom de la date d'apparition, d'acquittement et la duree du defaut
                                }
                        ?>
                </TABLE>
        </div>
</div>

<div id="tableaux">
        <div id="right">
                <TABLE align=center id="colonne">
                        <TR align=center>
                                <TD colspan=2 id="titre">CAPTEURS</TD>
                        </TR>
                        <TR>
                                <TD height=20></TD>
                        </TR>
                        <TR align=center style="font-weight:bold">
                                <TD width=80> Nom </TD>
                                <TD width=80> Etat </TD>
                        </TR>
                        <TR>
                                <TD height=10></TD>
                        </TR>
                        <?php
                                $reqCapteurs = $bdd->query('SELECT * FROM sensors ORDER BY nom');               // requete sql pour affichier les élements de la table capteurs

                                while($donneeCapteurs = $reqCapteurs->fetch()){
                                // boucle qui parcours la table tant qu'il y a des élements
                                        echo "<TR>
                                                <TD align=center>",$donneeCapteurs['nom'],"</TD>
                                                <TD align=center>",$donneeCapteurs['etat'],"</TD>
                                        </TR>";                         // affichage du nom et de l'état du capteur
                                }
                        ?>
                </TABLE>
        </div>
</div>
```

Programme TableBDD.php:

```
<?php
	// Connexion à la base de données
	try {
		require("include_bdd.php");
	}
	catch(Exception $e){
		die('Erreur : '.$e->getMessage());
	}
?>

<TABLE align=center>
	<?php

		$Pos_image_left_D = [1000,610,610,1000,890,1050,925,875,825,605,570,520,
		275,720]; // Tableau avec les valeur left des défauts
		$Pos_image_top_D = [1070,1070,725,725,690,610,550,275,550,550,275,550,
		610,690]; // Tableau avec les valeur top des défauts
		$i_D = 0;		// indice pour les défauts


		$reqDefauts = $bdd->query('SELECT * FROM defaults ORDER BY nom');
		 // requete sql pour afficher les élements de la table defauts


		while($donneeDefauts = $reqDefauts->fetch())
		// boucle qui parcours la table tant qu'il y a des élements
		{
			$LEFT = $Pos_image_left_D[$i_D];
			$TOP = $Pos_image_top_D[$i_D];
			if($donneeDefauts["etat"]=="1")
				echo '<DIV style="position:absolute; left:'.$LEFT.'px;
				top:'.$TOP.'px; width:136px; height:132px; z-index:2">
				<IMG src="D_on.png"> </DIV>';
				// affichage d'un panneau attention à la position définie


			$i_D+=1;
		}
	?>
</TABLE>
```

```
<TABLE align=center>
	<?php

		$i_C = 0;		// indice pour les capteurs
		$Pos_image_left_C = [1010,1010,740,1010,590,740,820,1000,835,835,530,
		530,525,750,835,740,1000,450,910,605];
		// Tableau avec les valeur left des capteurs

		$Pos_image_top_C = [780,910,1065,990,820,725,725,565,415,470,415,470,
		650,650,650,575,650,573,495,495]; // Tableau avec les valeur top des capteurs


		$reqCapteurs = $bdd->query('SELECT * FROM sensors ORDER BY nom');
		// requete sql pour affichier les élements de la table capteurs


		while($donneeCapteurs = $reqCapteurs->fetch())
		// boucle qui parcours la table tant qu'il y a des élements
		{
			$LEFT = $Pos_image_left_C[$i_C];
			$TOP = $Pos_image_top_C[$i_C];
			if($donneeCapteurs["etat"]=="1")
				echo '<DIV style="position:absolute;
				left:'.$LEFT.'px; top:'.$TOP.'px; width:136px;
				height:132px; z-index:2">
				<IMG src="C_on.png"> </DIV>';
				// affichage d'un carré vert à la position définie
			$i_C+=1;
		}
	?>
</TABLE>
```

Programme supervision.php:

```
<HTML>
	<HEAD>
		<!-- Librairie JQuery -->
		<script type="text/javascript" src="jquery-1.12.1.min.js"></script>

		<!-- Raffraichissement de la bdd -->
		<script type="text/javascript">

			$(document).ready(function(){
				refreshTable();
			});
			function refreshTable(){
				$('#tableBDD').load('TableBDD.php', function(){
				  setTimeout(refreshTable, 1000);
				});
			}
		</script>
	</HEAD>

	<BODY>
		<?php require ("header.php"); ?>
		<DIV style="position:absolute; left:200px; top:250px; width:136px; height:132px; z-index:1">
			<IMG src="Ligne.png" >
		</DIV>
		<DIV id="tableBDD"></DIV>
		<div id="description">
			<a href="description.php"><img src="fleche.png" width=30%/></p>Pour plus de pr&eacute;cision ...</a>
		</div>
		<DIV style="position:absolute; left:720px; top:810px; z-index:1">
			<IMG src="Staubli.jpg" >
		</DIV>
		<DIV style="height: 950;">
		</DIV>
		<?php require ("footer.php"); ?>

	</BODY>
</HTML>
```

Annexe III.III Programme robot

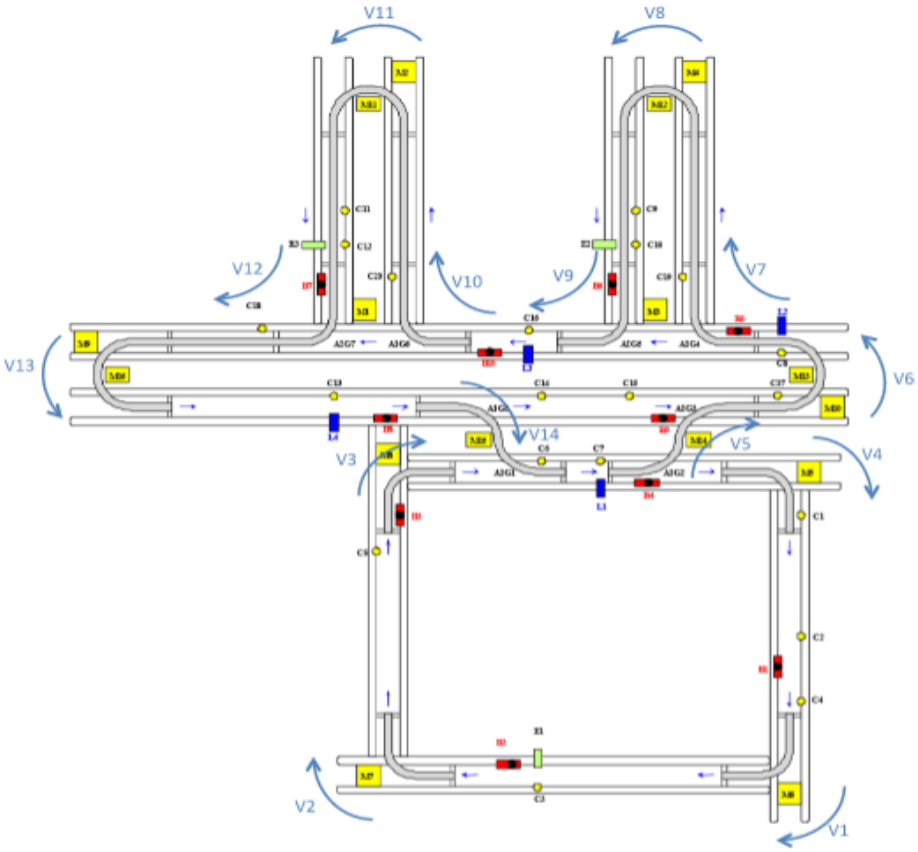
```
.PROGRAM boucle.pg()
        READY  //position initial
        BREAK

        WHILE (1) DO
                CALL main.pg()
        END
********************************
.PROGRAM main.pg()

        IF(1001, -1002, -1003)
                CALL charge.pg()
                SIGNAL 1, -2, -3
        END
        IF(1001, 1002, -1003)
                CALL decharge.pg()
                SIGNAL 1, 2, -3
        END
        IF(-1001, 1002, -1003)
                CALL usinage.pg()
                SIGNAL -1, 2, -3
        END
        IF(-1001, -1002, 1003)
                CALL repusi.pg()
                SIGNAL -1, -2, 3
        END
        IF(1001, -1002, 1003)
                CALL repch.pg()
                SIGNAL 1, -2, 3
        END
```

```
.PROGRAM charge.pg()

        SET #movpos = #PPOINT(-13.3,-3.8,177.8,0,42.2,0)
        SET #movrep = #PPOINT(-13.3,-9.6,177.8,0,42.2,0)

        CLOSEI
        DELAY(2)

        MOVE #movrep
        BREAK

        MOVE #movpos
        BREAK

        OPENI
        DELAY(2)

        MOVE #movrep
        BREAK

        CLOSEI
        DELAY(2)
*********************************************************
.PROGRAM decharge.pg()

        SET #movpos = #PPOINT(102.6,-7.9,129.7,0,72.3,0)
        SET #movrep = #PPOINT(102.6,-2.1,129.7,0,72.3,0)

        OPENI
        DELAY(2)

        MOVE #movrep
        BREAK

        MOVE #movpos
        BREAK

        CLOSEI
        DELAY(2)

        MOVE #movrep
        BREAK

        OPENI
        DELAY(2)
```

```
.PROGRAM usinage.pg()

        SET #movposa = #PPOINT(-116.08,-27.34,160.96,-6.545,
                                -47.11,8.62)
        SET #movposr = #PPOINT(-116.08,-38.99,184.63,-3.3,-53.75,0)
        SET #movrep = #PPOINT(-116.08,-56.9,197.7,-3.3,-53.75,0)

        OPENI
        DELAY(2)

        MOVE #movrep
        BREAK

        MOVE #movposr
        BREAK

        MOVE #movposa
        BREAK

        CLOSEI
        DELAY(5)

        MOVE #movposr
        BREAK

        MOVE #movrep
        BREAK

        OPENI
        DELAY(2)
*********************************************************
.PROGRAM repusi.pg()

        SET #movpos = #PPOINT(-52,-40,130,0,72,0)

        MOVE #movposr
        BREAK
*********************************************************
.PROGRAM repch.pg()

        SET #movpos = #PPOINT(-49,-40,130,0,72,0)

        MOVE #movposr
        BREAK
```

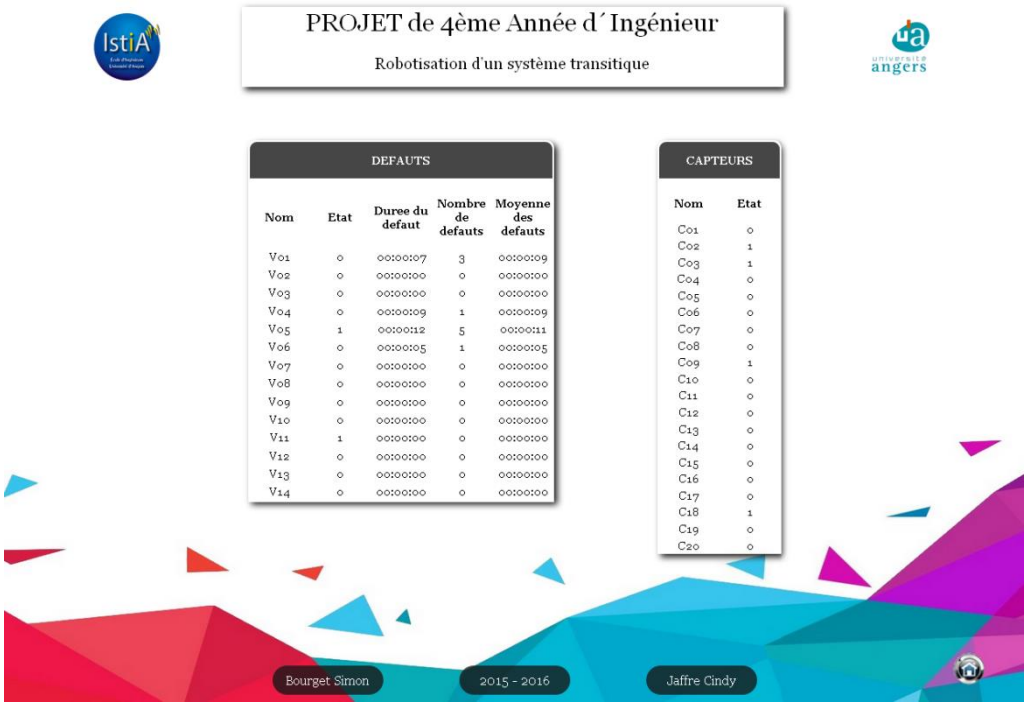
Annexe IV. Images liées au projet

Annexe IV.I Carte des virages

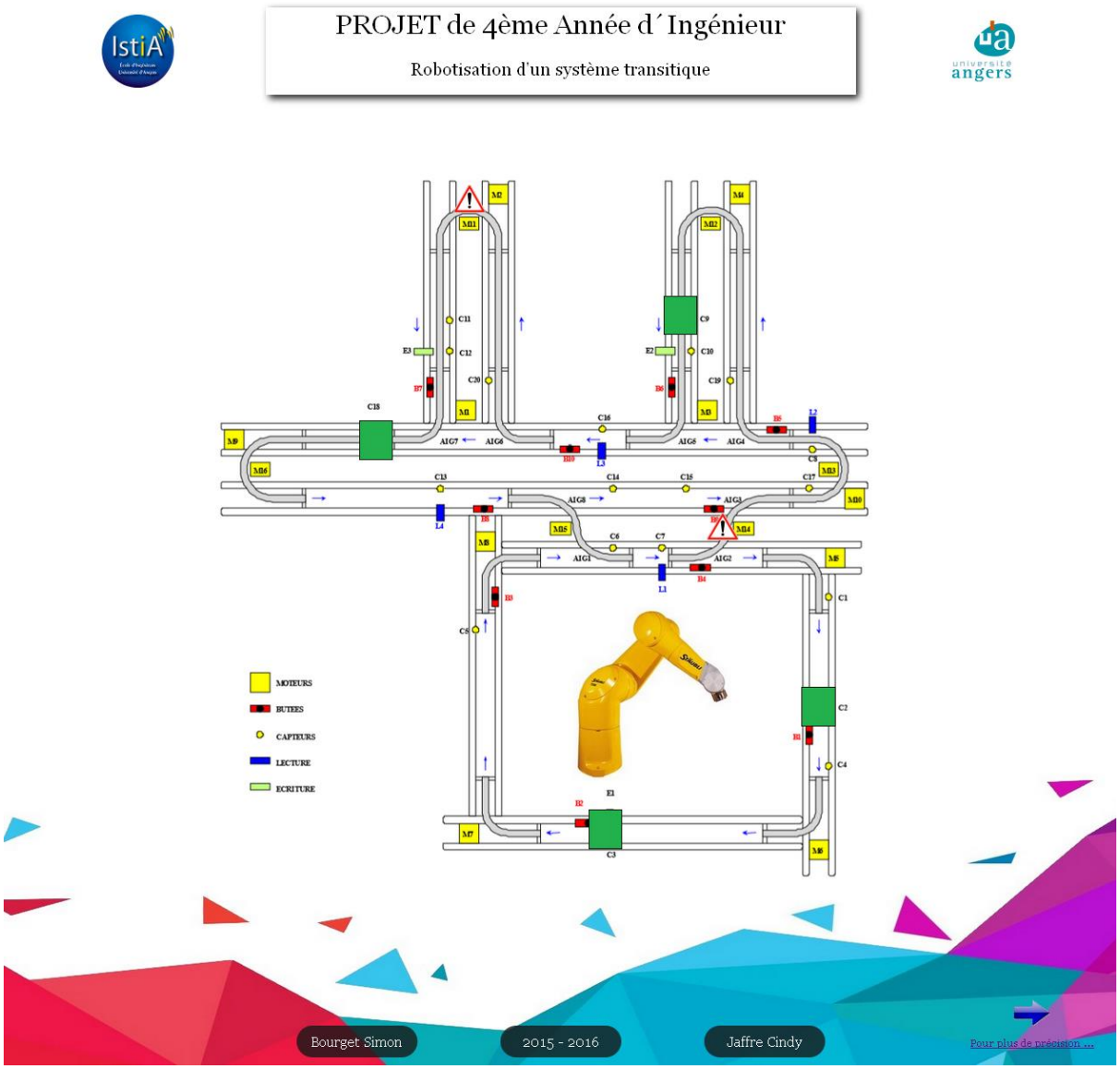


Annexe IV.II Images de l'interface web

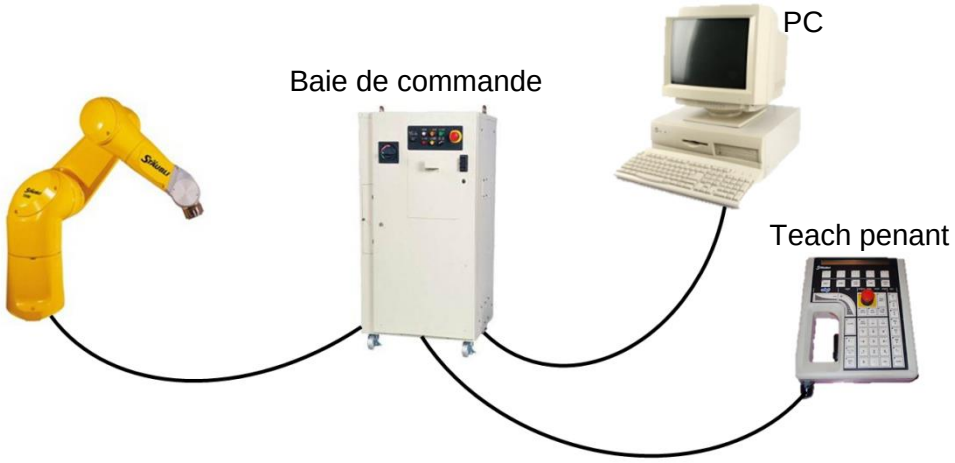
Interface description.php :



Interface supervision.php :



Annexe IV.III Composition du robot



ABSTRACT

This project is based on a practical course of industrial network of the 4th year of automation. The aim is combine different competences like web programmation, supervision, automation and robotic on the same system of production. We work in pair and we split the different tasks between us according to our skills.

The first part of the project, is the programmation of the three automata which manage the conveyors, the sensors, the stops and the switchs to drive the palettes. The second part consists of the monitoring of the process using a data base to show on a web page the sensors and the defaults. The last part, concerns the programmation of the robot, with three different positions, one to load, a second to work on a piece and the other to unload.

RESUME

Ce projet est basé sur un TP de réseau industriel de 4ème année d'automatique. Le but est de combiner différentes compétences comme la programmation web, la supervision, l'automatisme et la robotique sur un même système de production. Nous avons travaillé en binôme et nous avons séparé les différentes tâches à effectuer suivant nos compétences.

La première partie du projet est la programmation des trois Automates Programmables Industriels (API) qui gèrent les convoyeurs, les capteurs, les butées et les aiguillages qui dirigent les palettes. La deuxième, consiste à superviser le système via une base de données pour afficher sur une page web, l'état des capteurs et les défauts. La dernière partie concerne la programmation du robot, avec trois positions différentes, une pour le chargement, la deuxième pour l'usinage et l'autre pour le déchargement.

MOTS-CLES

Automate Programmable Industriel, Robot Staubli, convoyeur, supervision, base de données, GRAFCET, LIST, communication réseau.