

# Compte Rendu de Travaux pratiques

Dans le cadre du cours de Modèle pour le parallélisme de la formation Systèmes Robotique et Interactifs (SRI) de l'école d'ingénieur UPSSITECH

## Table des matières

1	Introduction.....	2
2	Présentation du système.....	2
3	Cahier des charges 1 – prise en main.....	4
3.1	Objectif.....	4
3.2	Notre solution.....	4
3.3	Description détaillée des étapes.....	4
4	Cahier des charges 2 – commande d'un aiguillage 2-en-1.....	5
4.1	Objectif.....	5
4.2	Notre solution.....	5
4.3	Description détaillée des étapes.....	5
5	Cahier des charges 3 – commande d'un aiguillage 1-en-2.....	6
5.1	Objectif.....	6
5.2	Notre solution.....	6
5.3	Description détaillée des étapes.....	7
5.3.1	Grafcet de l'aiguillage A1.....	7
5.3.2	Grafcet de l'aiguillage A2 (et A11).....	7
5.3.3	Grafcet de l'aiguillage A12.....	8
5.3.4	Grafcet de la zone de transfert 2.....	8
6	Cahier des charges 4 – Machine de Petri en langage ST.....	9
6.1	Objectif.....	9
6.2	Notre solution.....	9
6.3	Déclaration du Timer.....	10
6.4	Codage du réseau de pétri Figure 5 en langage ST.....	10
6.5	Description détaillée du programme.....	11

## 1 Introduction

Ces TP ont pour but d'illustrer différentes manières de commander un système de transport industriel. La cellule flexible des espaces robotiques de la Maison de la Formation Jacqueline Auriol (MFJA) est considérée et différents cahiers des charges sont proposés, du plus simple au plus complexe. Dans un premier temps, une commande Grafset du système de transport sera réalisée. On mettra ensuite en œuvre une commande modélisée par réseaux de Petri et implémentée en langage ST.

## 2 Présentation du système

On considère un système de transport composé de sections et d'aiguillages sur lesquels circulent des navettes autonomes. Un automate permet de contrôler les parties du système représentées sur la figure suivante.

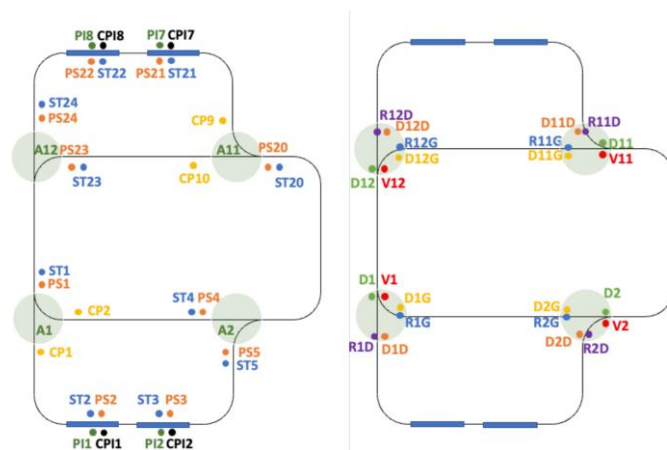


Figure 1 : Schéma représentatif du système

Les navettes transportent des produits d'un poste de travail en autre (représenté par les segments bleus sur la figure). Les aiguillages permettent de dériver les navettes vers des postes de travail robotisés. Les navettes se déplacent dans le sens inverse des aiguilles d'une montre dès le système mis sous tension.

Soit les différents capteurs et actionneurs présentés sur la figure si dessus tel que :

- Sur les **rails** sont positionnés :

<b>PSi</b>	1, 2, 11, 12	Capteur de présence	Capteurs disposés en amont des aiguillages <b>Ai</b>
<b>CPi</b>	1, 2, 9, 10	Capteur de présence	Capteurs disposés en aval des aiguillages <b>Ai</b>
<b>STi</b>	1, 2, 11, 12	Actionneur pneumatique	Libère le frein pneumatique et permet à la navette de démarrer
<b>PIi</b>	1, 2, 7, 8	Actionneur pneumatique	Active le frein pneumatique devant le poste de travail
<b>CPIi</b>	1, 2, 7, 8	Capteur de présence	Indique l'état de l'indexage

- Sur les **aiguillages Ai** sont positionnés :

<b>Rid</b>	1, 2, 11, 12	Actionneur pneumatique	Commande une rotation à droite de l'aiguillage
<b>Rig</b>	1, 2, 11, 12	Actionneur pneumatique	Commande une rotation à gauche de l'aiguillage
<b>Di</b>	1, 2, 11, 12	Actionneur pneumatique	Déverrouille l'aiguillage
<b>Vi</b>	1, 2, 11, 12	Actionneur pneumatique	Verrouille l'aiguillage
<b>Did</b>	1, 2, 11, 12	Capteur de position	Actif si positionnement de l'aiguillage à droite
<b>Dig</b>	1, 2, 11, 12	Capteur de position	Actif si positionnement de l'aiguillage à gauche

**Note** : 2 navettes peuvent éventuellement se rattraper mais une détection de proximité évite les collisions.

Une commande rotation ne peut être exécutée que si l'aiguillage est déverrouillé (et s'il n'est pas déjà dans la configuration finale souhaitée). Après chaque rotation, l'aiguillage doit de nouveau être verrouillé pour empêcher le déraillement des navettes qui le traversent. De façon générale, il faut interdire toute rotation de l'aiguillage

pendant le passage d'une navette. On suppose que la configuration initiale au moment de la mise sous tension sera contrôlée humainement pour assurer qu'elle soit sans risque pour le matériel.

La manipulation consiste à satisfaire les cahiers des charges suivants :

- Le Cahier des Charges 1 permet une prise en main du matériel.
- Les autres cahiers des charges sont incrémentaux et seront réalisés au sein d'un même projet par ajout successif de programmes.

Des projets UnityPro (Schneider Electric) préconfigurés ont été fournis permettant en particulier de faire le lien entre les noms symboliques des capteurs et actionneurs et leur adresse sur l'automate.

Name	Type	Value	Comment	Alias	Alias of	Address	HMI variable	R/W Rights of Referenced Variable
CapPosD...	BOOL							
CapPosG...	BOOL							
CP1	EBBOOL					%I0.2.7		
CP2	EBBOOL					%I0.2.12		
CP9	EBBOOL					%I0.1.0		
CP10	EBBOOL					%I0.1.12		
CP11	EBBOOL					%I0.1.8		
CP12	EBBOOL					%I0.1.9		
CP17	EBBOOL					%I0.2.8		
CP18	EBBOOL					%I0.2.9		
cpt	UINT							
D1	EBBOOL					%Q0.4.3		
D10	EBBOOL					%I0.2.5		
D1G	EBBOOL					%I0.2.6		
D2	EBBOOL					%Q0.4.7		
D20	EBBOOL					%I0.2.10		
D2G	EBBOOL					%I0.2.11		
D11	EBBOOL					%Q0.3.3		
D11D	EBBOOL					%I0.1.5		
D11G	EBBOOL					%I0.1.6		
D12	EBBOOL					%Q0.3.7		
D12D	EBBOOL					%I0.1.10		
D12G	EBBOOL					%I0.1.11		
FTempo	BOOL							
P1	UINT	1						
P2	UINT							
P3	UINT							
P4	UINT							
P5	UINT							
P6	UINT							
P7	UINT							
P8	UINT							
P9	UINT							
P10	UINT							
P11	UINT							
P12	UINT							
P13	UINT							
P14	UINT							
P15	UINT							
P16	UINT							
P17	UINT							
P18	UINT							
P19	UINT							
P20	UINT							
P21	UINT							
P22	UINT							
PI1	EBBOOL					%Q0.3.13		
PI2	EBBOOL					%Q0.3.14		
PI7	EBBOOL					%Q0.4.13		
PI8	EBBOOL					%Q0.4.14		
Presence	BOOL							
PS1	EBBOOL					%I0.1.13		
PS2	EBBOOL					%I0.2.1		
PS3	EBBOOL					%I0.2.2		
PS4	EBBOOL					%I0.2.4		
PS4andn...	BOOL							
PS5	EBBOOL					%I0.2.3		
PS20	EBBOOL					%I0.1.7		
PS21	EBBOOL					%I0.1.1		
PS22	EBBOOL					%I0.1.2		
PS23	EBBOOL					%I0.1.4		
PS24	EBBOOL					%I0.1.3		
R1D	EBBOOL					%Q0.4.0		
R1G	EBBOOL					%Q0.4.1		
R2D	EBBOOL					%Q0.4.4		
R2G	EBBOOL					%Q0.4.5		
R11D	EBBOOL					%Q0.3.0		
R11G	EBBOOL					%Q0.3.1		
R12D	EBBOOL					%Q0.3.4		
R12G	EBBOOL					%Q0.3.5		
ST1	EBBOOL					%Q0.4.8		
ST2	EBBOOL					%Q0.4.9		
ST3	EBBOOL					%Q0.4.10		
ST4	EBBOOL					%Q0.4.12		
ST5	EBBOOL					%Q0.4.11		
ST20	EBBOOL					%Q0.3.8		
ST21	EBBOOL					%Q0.3.9		
ST22	EBBOOL					%Q0.3.10		
ST23	EBBOOL					%Q0.3.12		
ST24	EBBOOL					%Q0.3.11		
Tempo	BOOL							
V1	EBBOOL					%Q0.4.2		
V2	EBBOOL					%Q0.4.6		
V11	EBBOOL					%Q0.3.2		
V12	EBBOOL					%Q0.3.6		

Figure 2 : Variables élémentaires

### 3 Cahier des charges 1 - prise en main

#### 3.1 Objectif

Créer un Grafcet où tous les aiguillages basculent à droite, puis aussitôt à gauche.

Libérer alors les stops de sorte que les navettes présentes dans le système circulent de façon ininterrompue, dans la boucle interne (on suppose qu'aucune navette n'est localisée sur un aiguillage au moment de leur rotation).

#### 3.2 Notre solution

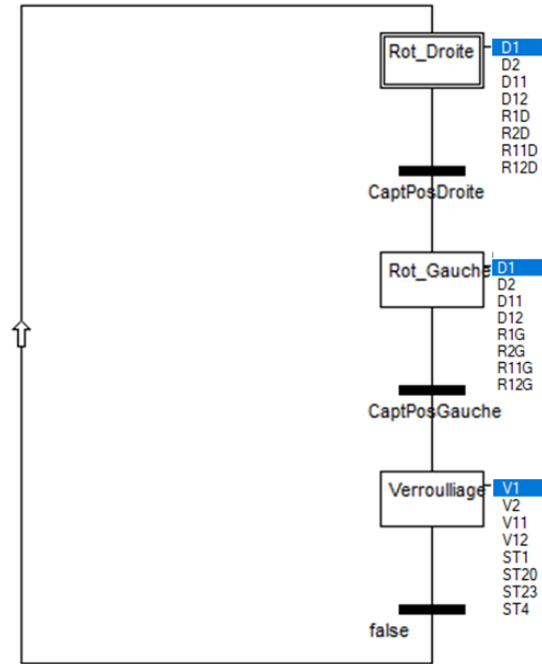


Figure 3 : Grafcet satisfaisant le cahier des charges 1

Conditions sur transitions décrite en langage ST

```

CaptPosGauche:=D1G and D2G and D11G and D12G;
CaptPosDroite:=D1D and D2D and D11D and D12D;
PS4andnotPS5:=PS4 and (not PS5);
  
```

#### 3.3 Description détaillée des étapes

1. A l'initialisation du grafcet, dans l'état *Rot\_Droite*, les aiguillages sont déverrouillés (D1, D2, D11, D12) puis effectues une rotation à droite (R1D, R2D, R11D, R12D).
2. Lorsque la condition *CaptPosDroite* est satisfaite, c'est-à-dire que tous les capteurs (D1D, D2D, D11D, D12D) ont détecté l'aiguillage bien positionné à droite on passe à l'état *Rot\_Gauche*.
3. Dans l'état *Rot\_Gauche*, les aiguillages sont déverrouillés (D1, D2, D11, D12) puis effectues une rotation à gauche (R1G, R2G, R11G, R12G).
4. Lorsque la condition *CaptPosGauche* est satisfaite, c'est-à-dire que tous les capteurs (D1G, D2G, D11G, D12G) ont détecté l'aiguillage bien positionné à droite on passe à l'état *Verrouillage*
5. Dans l'état *Verrouillage* tous les aiguillages sont verrouillés (V1, V2, V3, V4) et les freins des navettes sont déverrouillés (ST1, ST20, ST23, ST4) afin qu'elles puissent naviguer librement.
6. Une condition *false* a été ajouté afin de rester bloqué dans l'état *Verrouillage* et ainsi satisfaire le cahier des charges.

## 4 Cahier des charges 2 - commande d'un aiguillage 2-en-1

### 4.1 Objectif

Proposer et tester un Grafcet pour commander l'aiguillage A2 afin de permettre un passage sans collision des navettes arrivant par l'un ou l'autre des entrées de l'aiguillage, quel que soit le nombre de navettes présentes dans le système.

### 4.2 Notre solution

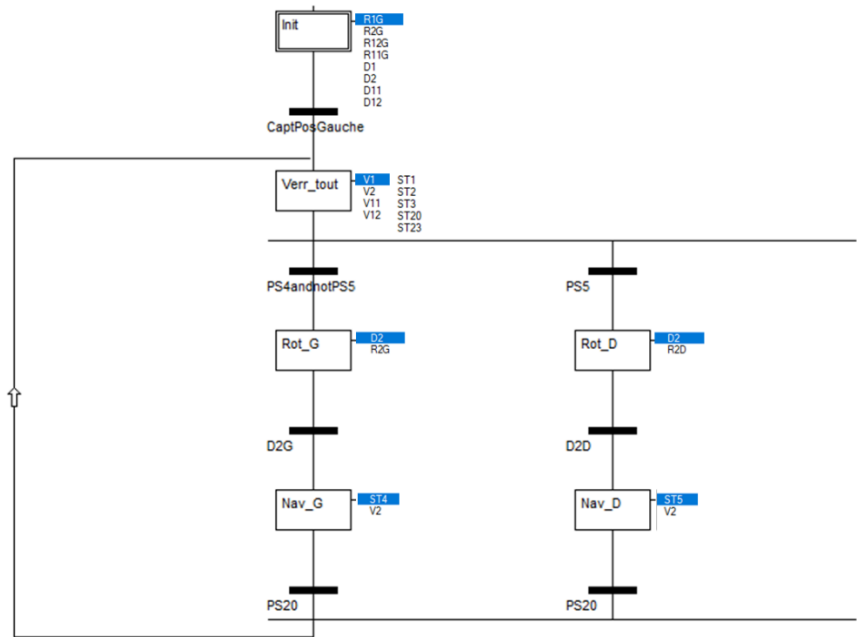


Figure 4 : Grafcet satisfaisant le cahier des charges 2

Conditions sur transitions décrite en langage ST

CaptPosGauche:=D1G and D2G and D11G and D12G;  
PS4andnotPS5:=PS4 and (not PS5);

### 4.3 Description détaillée des étapes

1. A l'initialisation du grafcet, dans l'état *Init*, les aiguillages sont déverrouillés (D1, D2, D11, D12) et effectues une rotation à gauche (R1G, R2G, R11G, R12G).
2. Lorsque la condition *CaptPosGauche* est satisfaite, c'est-à-dire que tous les capteurs (D1G, D2G, D11G, D12G) ont détecté l'aiguillage bien positionné à gauche on passe à l'état *Verr\_tout*.
3. Dans l'état *Verr\_tout*, les aiguillages sont verrouillés (V1, V2, V11, V12) et les freins positionnés sur les rails en lignes droite sont déverrouillés (ST1, ST2, ST3, ST20, ST23) afin que les navettes puissent naviguer librement.
4. Ensuite deux cas sont possibles :
  - a. Soit une navette est détecté à droite (PS5) et on passe alors à l'état *Rot\_D*
    - i. Dans l'état *Rot\_D*, l'aiguillage A2 est déverrouillé (D2) puis effectues une rotation à droite (R2D).
    - ii. Lorsqu'on détecte que la navette a bien passé l'aiguillage (PS20) alors on revient à l'état *Verr\_tout*.
  - b. Soit il n'y a pas de navette droite (NOT PS5) mais il y en a une à gauche (PS4) alors on passe à l'état *Rot\_G* (priorité pour la navette venant de droite)
    - i. Dans l'état *Rot\_G*, l'aiguillage A2 est déverrouillé (D2) puis effectues une rotation à gauche (R2G).
    - ii. Lorsqu'on détecte que la navette a bien passé l'aiguillage (PS20) alors on revient à l'état *Verr\_tout*.

## 5 Cahier des charges 3 - commande d'un aiguillage 1-en-2

## 5.1 Objectif

Proposer et tester un modèle de commande Grafcet permettant d'assurer que l'aiguillage A1 oriente chaque navette se présentant à son entrée alternativement à droite, puis à gauche quel que soit le nombre de navettes présentes dans le système.

Pour répondre au cahier des charges nous avons imaginé 4 graficets en parallèle, soit :

1. 1 grafcet représentant l'aiguillage A1 qui orientera alternativement la navette à droite puis à gauche
2. 1 grafcet représentant l'aiguillage A2 (et gérant également l'aiguillage A1)
3. 1 grafcet représentant l'aiguillage A12 et permettant d'éviter toute collision entre navette au niveau de l'aiguillage A1
4. 1 grafcet représentant la zone de transfert 2 (en bas à droite de la Figure 1) permettant d'éviter toute collision entre navette au niveau de l'aiguillage A2

## 5.2 Notre solution



Figure 5 : Grafcets satisfaisant le cahier des charges 3

### Conditions sur transitions et actions décrite en langage ST

```
CaptPosGauche:=D1G and D2G and D11G and D12G;  
PS4andnotPS5:=PS4 and (not PS5);  
ST2:=True
```

### 5.3 Description détaillée des étapes

#### 5.3.1 Grafcet de l'aiguillage A1

1. A l'initialisation du grafcet, dans l'état *A1\_Rot\_G*, l'aiguillage A1 est déverrouillé (D1) et effectue une rotation à gauche (R1G).
2. Lorsque l'aiguillage est correctement positionné à gauche (D1G) on passe à l'état *A1\_VerrG*.
3. Dans *A1\_VerrG* on verrouille l'aiguillage A1.
4. Lorsqu'une navette arrive en amont de l'aiguillage A1 (PS1) on passe à l'état *A1\_PassG1*.
5. Dans *A1\_PassG1* on désactive le frein pour laisser passer la navette sur l'aiguillage A1 (ST1) pour la guider vers la gauche.
6. Lorsqu'on ne détecte plus la navette en amont de l'aiguillage A1 (NOT PS1) on passe à l'état *A1\_PassG2*.
7. Dans l'état *A1\_PassG2* on attend que la navette se déplace et traverse l'aiguillage A1.
8. Lorsque la navette sort de l'aiguillage et atteint le capteur aval de gauche (CP2) on passe à l'état *A1\_Rot\_D*.
9. Dans l'état *A1\_Rot\_D*, l'aiguillage A1 est déverrouillé (D1) et effectue une rotation à droite (R1D).
10. Lorsque l'aiguillage est correctement positionné à droite (D1D) on passe à l'état *A1\_VerrD*.
11. Dans *A1\_VerrD* on verrouille l'aiguillage A1.
12. Lorsqu'une navette arrive en amont de l'aiguillage A1 (PS1) on passe à l'état *A1\_PassD1*.
13. Dans *A1\_PassD1* on désactive le frein pour laisser passer la navette sur l'aiguillage A1 (ST1) pour la guider vers la droite.
14. Lorsqu'on ne détecte plus la navette en amont de l'aiguillage A1 (NOT PS1) on passe à l'état *A1\_PassD2*.
15. Dans l'état *A1\_PassD2* on attend que la navette se déplace et traverse l'aiguillage A1.
16. Lorsque la navette sort de l'aiguillage et atteint le capteur aval de droite (CPI) on retourne à l'état initial *A1\_Rot\_G*.

#### 5.3.2 Grafcet de l'aiguillage A2 (et A11)

1. A l'initialisation du grafcet, dans l'état *A211\_Init*, les aiguillages sont déverrouillés (D2, D11) et effectuent une rotation à gauche (R2G, R11G).
2. Lorsque la condition *CaptPosGauche* est satisfaite, c'est-à-dire que tous les capteurs (D1G, D2G, D11G, D12G) ont détecté l'aiguillage bien positionné à gauche on passe à l'état *A211\_Verr*.
3. Dans l'état *A211\_Verr*, les aiguillages sont verrouillés (V2, V11) et les freins positionnés sur les rails en lignes droite sont déverrouillés (ST4, ST5) afin que la navette puisse naviguer librement.
4. Ensuite deux cas sont possibles :
  - a. Soit une navette est détectée à droite (PS5) et on passe alors à l'état *A2\_Rot\_D*
    - i. Dans l'état *A2\_Rot\_D*, l'aiguillage A2 est déverrouillé (D2) puis effectue une rotation à droite (R2D).
    - ii. Lorsqu'on détecte que l'aiguillage est bien à droite (D2D) alors on passe à l'état *A2\_PassD*.
    - iii. Dans l'état *A2\_PassD*, l'aiguillage est verrouillé (V2) et les freins sur les rails en lignes droites sont déverrouillés (ST3, ST5) afin que la navette puisse naviguer librement.
    - iv. Lorsque le capteur de position amont de l'aiguillage A2 se désactive (NOT PS5), on passe à l'état *A2\_Pass*.
  - b. Soit il n'y a pas de navette droite (NOT PS5) mais il y en a une à gauche (PS4) alors on passe à l'état *A2\_Rot\_G* (priorité pour la navette venant de droite)
    - i. Dans l'état *A2\_Rot\_G*, l'aiguillage A2 est déverrouillé (D2) puis effectue une rotation à gauche (R2G).
    - ii. Lorsqu'on détecte que l'aiguillage est bien à gauche (D2G) alors on passe à l'état *A2\_PassG*.
    - iii. Dans l'état *A2\_PassG*, l'aiguillage est verrouillé (V2) et les freins sur les rails en lignes droites sont déverrouillés (ST4) afin que la navette puisse naviguer librement.

- iv. Lorsque le capteur de position amont de l'aiguillage A2 se désactive (NOT PS4), on passe à ce moment à l'état *A2\_Pass*.
- 5. Dans l'état *A2\_Pass*, on attend que la navette traverse l'aiguillage A2.
- 6. Lorsque la navette est détectée au niveau du capteur de position amont de l'aiguillage A1 (PS20), on passe à l'état *A1\_Pass*.
- 7. Dans l'état *A1\_Pass*, les freins positionnés sur les rails en lignes droite sont déverrouillés (ST20) afin que la navette puisse naviguer librement.
- 8. Lorsque le capteur de position amont de l'aiguillage A1 se désactive (NOT PS20), c'est-à-dire, que la navette c'est bien déplacé, on retourne à l'état *A2\_Verr*.

### 5.3.3 Grafcet de l'aiguillage A12

Ce grafcet permet d'éviter toute collision entre navette au niveau de l'aiguillage A1

- 1. A l'initialisation du grafcet, dans l'état *A12\_Init*, les aiguillages sont déverrouillés (D12) puis effectues une rotation à gauche (R12D).
- 2. Lorsque le capteur d'orientation de l'aiguillage A12 détecte le rail a gauche (D12G) on passe à l'état *A12\_Pass*.
- 3. Dans l'état *A12\_Pass*, l'aiguillage A12 se verrouille (V12) et les freins positionnés sur les rails sont déverrouillés (ST23) afin que la navette puisse passer.
- 4. Lorsque le capteur de position de l'aiguillage A1 détecte la présence d'une navette (PS1), c'est-à-dire qu'une navette est déjà en attente pour passer sur l'aiguillage A1, on passe alors à l'état *A12\_Att*.
- 5. Dans l'état *A12\_Att*, la navette attend au niveau de l'aiguillage A12 que la place se libère au niveau de l'aiguillage A1
- 6. Lorsque le capteur amont de l'aiguillage A1 est désactivé (NOT PS1), cela signifie que la place c'est libéré, on revient donc à l'état *A12\_Pass*.

### 5.3.4 Grafcet de la zone de transfert 2

Ce grafcet permet d'éviter toute collision entre navette au niveau de l'aiguillage A2

- 1. A l'initialisation du grafcet, dans l'état *ZT\_Pass*, les freins positionnés sur les rails sont déverrouillés (ST3) afin que les navettes puissent naviguer librement.
- 2. Lorsque le capteur de position amont de l'aiguillage A2 (PS5) détecte la présence d'une navette c'est qu'une navette est déjà en attente pour passer sur l'aiguillage A2, alors on passe à l'état *ZT\_Att*.
- 3. Dans l'état *ZT\_Att*, la navette attend au niveau de la zone de transfert 2 (car ST2:=True) que la place se libère au niveau de l'aiguillage A2
- 4. Lorsque le capteur de position amont de l'aiguillage A2 est désactivé (NOT PS5) cela signifie que la place c'est libéré, on revient donc à l'état *ZT\_Pass*.



## 6 Cahier des charges 4 – Machine de Petri en langage ST

### 6.1 Objectif

Les commandes précédentes des aiguillages A1 et A2 sont conservées. On s'intéresse à présent à concevoir et tester une nouvelle commande modélisée par Réseaux de Petri pour les aiguillages A11 et A12 (que l'on implémentera en langage ST).

La commande doit dériver les navettes vers les postes P7 et P8 en assurant qu'il n'y ait jamais plus de deux navettes présentes dans la zone de dérivation (comprise entre les capteurs CP9 et PS24). La première navette entrante est arrêtée devant le poste P8 grâce à l'actionneur ST22 et est indexée (actionneur PI8) pendant 15 secondes, puis libérer pour repartir. Si une deuxième navette est dérivée vers les postes P7 et P8 alors qu'une navette est déjà présente, elle est arrêtée devant le poste P7 via l'actionneur ST21 tant que le poste P8 est occupé, puis acheminée à son tour vers le poste P8 où elle stationne pendant 15 secondes avant de repartir à nouveau.

### 6.2 Notre solution

Plutôt que d'implémenter directement le réseau de pétri complet et que les erreurs soient plus compliquées à trouver en cas de problème, nous avons décidé de commencer par implémenter le cas où nous n'avons qu'une seule navette de présente sur le circuit.

Malheureusement par manque de temps nous n'avons pas pu implémenter le programme en langage ST afin de le tester sur maquette.

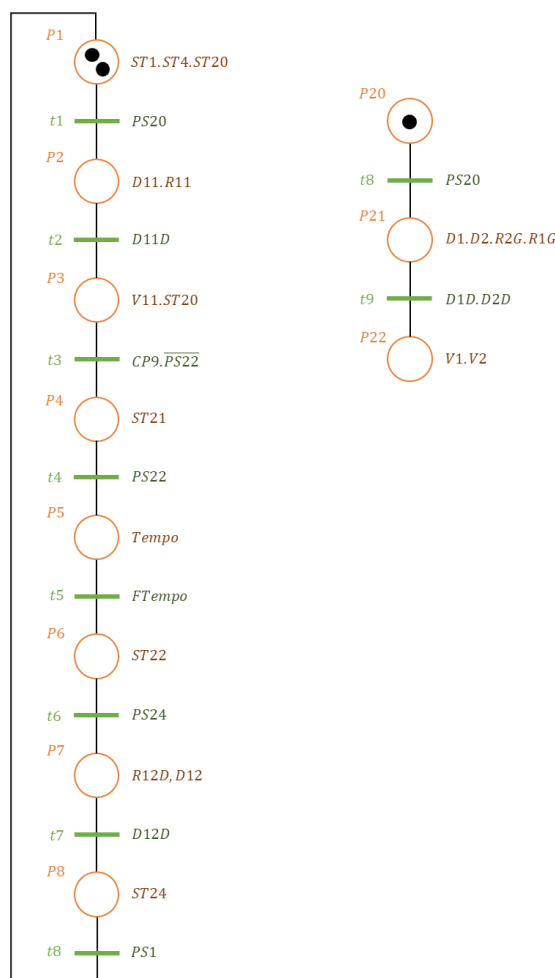


Figure 6 : Réseau de Pétri pour une seule navette

### 6.3 Déclaration du Timer

Le timer *MonTimer* a été déclaré dans la rubrique *Elementary FB Instances* du logiciel

Name	no.	Type	Value	Comment	HMI variable	R/W Rights of Referenced Variable
MonTimer		TON				
<inputs>						
IN	1	BOOL		Start delay		
PT	2	TIME		Preset delay time		
<outputs>						
Q	1	BOOL		Delayed output		
ET	2	TIME		Internal time		
<inputs/outputs>						
<public>						

Figure 7: Déclaration du Timer dans le logiciel

### 6.4 Codage du réseau de pétri Figure 6 en langage ST

```
(*//////////CREATION DU TIMER//////////*)
MonTimer(IN:=(Tempo),
  PT:=(t#15s),
  Q=> FTempo);

(*//////////INITIALISATION//////////*)
IF (inter1) THEN
  P1:=2;
  P2:=0;
  P3:=0;
  P4:=0;
  P5:=0;
  P6:=0;
  P7:=0;
  P8:=0;
  P20:=1;
  P21:=0;
  P22:=0;

ELSE
  (*//////////TRANSITIONS//////////*)

  (*t1*) IF(PS20)THEN P1:=P1-1; P2:=P2+1; END_IF;
  (*t2*) IF(D11D)THEN P2:=P2-1; P3:=P3+1; END_IF;
  (*t3*) IF(CP9 and not PS22)THEN P3:=P3-1; P4:=P4+1; END_IF;
  (*t4*) IF(PS22)THEN P4:=P4-1; P5:=P5+1; END_IF;
  (*t5*) IF(FTempo=1)THEN P5:=P5-1; P6:=P6+1; END_IF;
  (*t6*) IF(PS24)THEN P6:=P6-1; P7:=P7+1; END_IF;
  (*t7*) IF(D12D)THEN P7:=P7-1; P8:=P8+1; END_IF;
  (*t8*) IF(PS1)THEN P8:=P8-1; P1:=P1+1; END_IF;

  (*t9*) IF(PS20)THEN P20:=P20-1; P21:=P21+1; END_IF;
  (*t10*) IF(D1G and D2G)THEN P21:=P21-1; P22:=P22+1; END_IF;

  (*//////////ACTIONS//////////*)

  (*Place 1*) ST1:=(P1>0); ST4:=(P1>0); ST20:=(P1>0);
  (*Place 2*) D11:=(P2>0); R11D:=(P2>0);
  (*Place 3*) V11:=(P3>0); ST20:=(P3>0);
  (*Place 4*) ST21:=(P4>0);
  (*Place 5*) IF (P5>0) THEN set(Tempo); END_IF;
  (*Place 6*) ST22:=(P6>0);
  (*Place 7*) D12:=(P7>0); R12D:=(P7>0);
  (*Place 8*) ST24:=(P8>0);

  (*Place 21*) D1:=(P21>0); D2:=(P21>0); R1G:=(P21>0); R2G:=(P21>0);
  (*Place 22*) V1:=(P22>0); V2:=(P22>0);

END_IF
```

## 6.5 Description détaillée du programme

Dans le programme ci-dessus nous avons rajouté 2 variables élémentaires *Tempo* et *FTempo* de type Booléen. *Tempo* est l'entrée mise à 1 afin de débiter la temporisation, paramétrée pour 15 secondes. Une fois que les 15 secondes sont écoulées, la variable de sortie *FTempo* passera à 1 pour indiquer la fin.

1. Au début du programme si la variable *inter1* est à 1 on initialise le réseau de Petri en indiquant le nombre de jetons présents dans chaque place. Dans notre cas, on a :
  - a. 2 jetons dans la place P1,
  - b. 1 jeton dans la place P20,
  - c. 0 jeton dans les autres places
 L'initialisation étant effectuée, on passe la variable *inter1* à 0 pour faire évoluer le système.
2. Dans la place P1, on désactive les freins (ST1, ST4, ST20) afin que la navette puisse avancer.
3. Lorsque le capteur de position (PS20) détecte la navette, alors on quitte la place P1 (perte du jeton) et on passe dans la place P2 (gain d'un jeton).
4. Dans la place 2, on déverrouille l'aiguillage 11 (D11) et on effectue une rotation de l'aiguillage 11 vers la droite (R11D).
5. Lorsque l'aiguillage est correctement positionné à droite (D1D) on passe à la place P3 (perte du jeton dans P2 et gain du jeton dans P3).
6. Dans la place P3, on verrouille l'aiguillage 11 (V11) et on désactive le frein associé (ST20) pour que la navette puisse se déplacer sur l'aiguillage 11.
7. Lorsque la navette est détectée par le capteur aval de l'aiguillage (CP9) et qu'aucune navette n'est présente sur le capteur de position de la zone de transfert 8 en haut à gauche de la Figure 1 (NOT PS22), alors on passe à la place P4.
8. Dans la place P4, comme aucune navette n'est présente dans la zone de transfert 8 (NOT PS22), on désactive les freins de la zone de transfert 7 (ST21) pour que la navette puisse continuer jusqu'à la zone de transfert 8 car la place est libre.
9. Lorsque le capteur de position amont de la zone de transfert 8 (PS22) détecte une nouvelle présence on passe à la place P5.
10. Dans la place 5 on démarre la temporisation.
11. Lorsque les 15 secondes sont écoulées, la variable *FTempo* du temporisateur *MonTimer* passe à 1 et permet de passer à la place P6.
12. Dans la place P6, on désactive le frein de la zone de transfert 8 (ST22) pour que la navette puisse continuer vers l'aiguillage.
13. Lorsque le capteur de position amont de l'aiguillage A12 (PS24) détecte une présence, on passe à la place P7.
14. Dans la place P7 on déverrouille l'aiguillage A12 (D12) et on effectue une rotation à droite (R12D).
15. Lorsque le capteur d'orientation de l'aiguillage A12 (D12D) détecte l'aiguillage à droite on passe à la place P8.
16. Dans la place P8 on désactive le frein amont de l'aiguillage A12 (ST24) afin de laisser passer la navette.
17. Lorsque la navette a passé l'aiguillage A12 et a été détecté par le capteur de position amont de l'aiguillage A1 (PS1) on se retrouve alors à la fin de cette branche du réseau de pétri représentant une des possibilités du cahier des charges. On revient donc à l'étape 1 pour recommencer un cycle.

Réseau de pétri pour maintenir les aiguillages A1 et A2 à gauche

1. A l'initialisation, dans la place 20, aucune action n'est effectuée.
2. Lorsque le capteur de position aval de l'aiguillage A11 (PS20) détecte une navette, on passe à la place P21.
3. Dans la place P21 on déverrouille les aiguillages A1 et A2 (D1, D2) et on effectue une rotation à gauche (R1G, R2G).
4. Lorsque le capteur d'orientation de l'aiguillage A12 (D1G, D2G) détecte l'aiguillage à gauche on passe à la place P22.

5. Dans la place P22 on verrouille les aiguillages A1 et A2 à gauche (V1, V2).  
Cette place est donc un état puit. Une fois que les aiguillages A1 et A2 sont réglés, on ne s'en occupe plus.