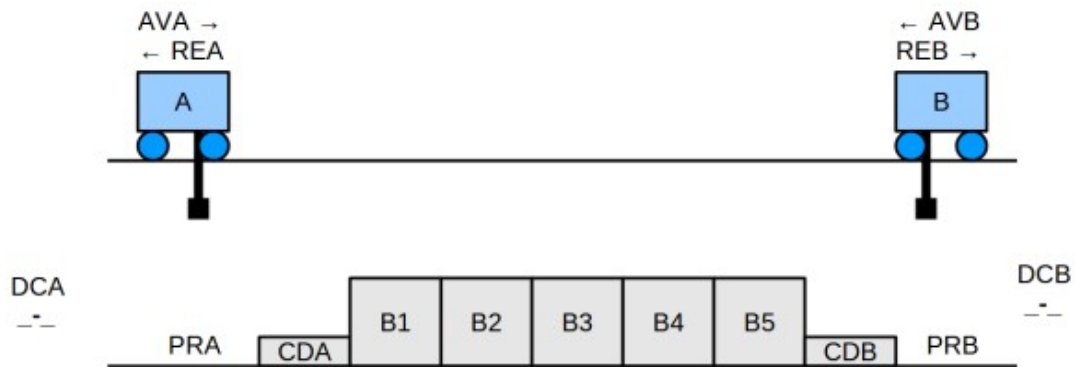


TP2 AUTOMATISME : Synchronisation avancée par Grafcet

Système des chariots

Niveau : MASTER 1 MESI



Auteur : Leroy Makita Loubaki & Daniel BAL
Enseignant : Pierre Paul Zeil

Table des matières

1	Introduction	2
2	Table des variables	3
3	Grafjets synchronisés du système	5
3.1	Grafjet bac de chargement	5
3.2	Grafjet bac de déchargement	6
3.3	Grafjet BAC 1	7
3.4	Grafjet BAC 3	8
3.5	Grafjet BAC 5	9
3.6	Grafjet Chariot A	10
3.7	Grafjet Chariot B	11
4	Interface HMI	12
4.1	Présentation générale	12
4.2	Description des éléments affichés	12
4.3	Animation des flèches de déplacement	14
4.4	Configuration des boutons de départ cycle	15
4.5	Animation des flèches du crochet	16
4.6	Affichage de la position des chariots	17
4.6.1	Liaison à la variable de position	17
4.6.2	Liste de textes <code>Liste_position_chariot</code>	17
4.7	Animation couleur des bacs	19
4.7.1	Liaison à la variable d'état	19
4.8	Animation du déplacement des chariots — Scripts VB	20
4.8.1	<code>VBFunction_1</code> — Chariot A	20
4.8.2	<code>VBFunction_2</code> — Chariot B	20
5	Supervision complémentaire — Node-RED	22
5.1	Architecture du flux Node-RED	22
5.2	Interface de supervision web	23
5.3	Description du dashboard	23
5.4	Avantages de la supervision Node-RED	24
6	Conclusion	25

1. Introduction

Dans le cadre du module d'automatisme de Master 1, ce travail pratique porte sur la synchronisation avancée de systèmes automatisés à l'aide du formalisme Grafset. Le système étudié est un poste de trempage composé de deux chariots motorisés, notés **A** et **B**, se déplaçant sur un rail commun au-dessus de cinq bains de traitement.

L'objectif principal est de concevoir la commande complète de ce système en garantissant un acheminement ordonné des bacs depuis la zone de chargement (CDA) jusqu'à la zone de déchargement (CDB), en passant successivement par les cinq bains de trempage (B1 à B5), chaque trempage ayant une durée minimale de deux secondes.

La contrainte centrale de ce système réside dans le fait que les deux chariots partagent le même rail : toute collision ou croisement doit être rigoureusement évité. La solution retenue repose sur une architecture modulaire de **7 Grafsets indépendants** un par chariot et un par bac dont la coordination est assurée par des variables d'échange et des signaux de disponibilité.

Ce compte rendu présente la démarche adoptée : décomposition fonctionnelle du système, conception et programmation des Grafsets sous *TIA Portal*, stratégie de synchronisation inter-Grafsets, ainsi que la réalisation d'une interface HMI permettant de superviser l'ensemble du cycle en temps réel.

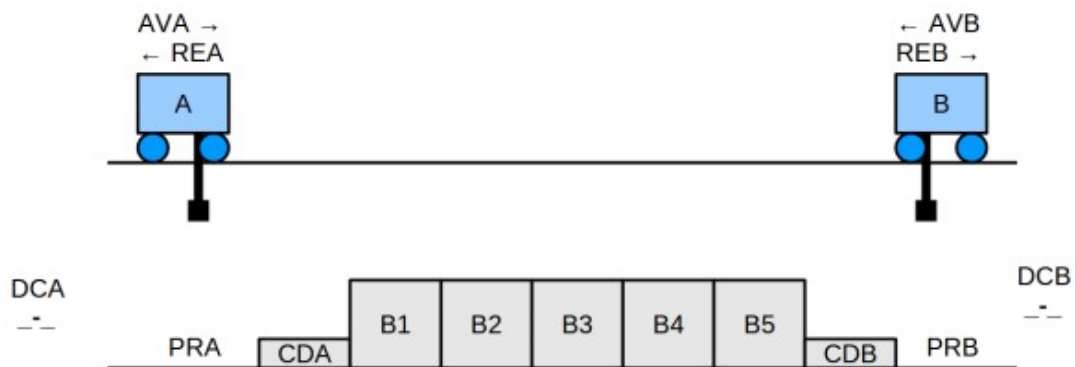


FIGURE 1 – Schéma du système

2. Table des variables

L'ensemble des entrées/sorties physiques ainsi que les variables internes du programme sont déclarées dans la table de tags du projet TIA Portal. Le tableau 1 en présente le contenu complet.

Remarque : certains capteurs partagent la même adresse physique pour les deux chariots (par exemple B2A/B1B en %I0.3), car ils correspondent à des positions communes sur le rail vues différemment selon le chariot. Les variables entières MB*_Etat encodent l'état de chaque bac (libre, en traitement, terminé), tandis que X_chariot_A et X_chariot_B mémorisent la position courante de chaque chariot pour la gestion des conflits.

TABLE 1 – Table des variables PLC

Nom	Type	Adresse	Description
<i>Sorties — Actionneurs</i>			
MoB	Bool	%Q0.0	Monter bac chariot B
DeB	Bool	%Q0.1	Descendre bac chariot B
ReB	Bool	%Q0.2	Reculer chariot B
AvB	Bool	%Q0.3	Avancer chariot B
MoA	Bool	%Q1.2	Monter bac chariot A
DeA	Bool	%Q1.3	Descendre bac chariot A
AvA	Bool	%Q1.4	Avancer chariot A
ReA	Bool	%Q1.5	Reculer chariot A
<i>Entrées — Capteurs chariot A</i>			
PRA	Bool	%I0.0	Position reculée chariot A
CDA	Bool	%I0.1	Zone de chargement chariot A
B1A	Bool	%I0.2	Chariot A au-dessus du bac 1
B2A/B1B	Bool	%I0.3	Chariot A/bac 2 — Chariot B/bac 1
B3A/B2B	Bool	%I0.4	Chariot A/bac 3 — Chariot B/bac 2
B4A/B3B	Bool	%I0.5	Chariot A/bac 4 — Chariot B/bac 3
B5A/B4B	Bool	%I0.6	Chariot A/bac 5 — Chariot B/bac 4
B5B	Bool	%I0.7	Chariot B au-dessus du bac 5
<i>Entrées — Capteurs chariot B et crochets</i>			
CHA	Bool	%I1.2	Crochet en haut chariot A
CBA	Bool	%I1.3	Crochet en bas chariot A
CHB	Bool	%I1.4	Crochet en haut chariot B
CBB	Bool	%I1.5	Crochet en bas chariot B
DCA	Bool	%I1.6	Départ cycle chariot A
DCB	Bool	%I1.7	Départ cycle chariot B
CDB	Bool	%I1.0	Zone de déchargement chariot B
PRB	Bool	%I1.1	Position reculée chariot B
<i>Variables internes — États des bacs et chariots</i>			
MB1_Etat	Int	%MW0	État du bac B1
MB2_Etat	Int	%MW1	État du bac B2
MB3_Etat	Int	%MW2	État du bac B3
MB4_Etat	Int	%MW3	État du bac B4
MB5_Etat	Int	%MW4	État du bac B5
MCDB_Etat	Int	%MW5	État de la zone de déchargement
MCDA_Etat	Int	%MW6	État de la zone de chargement
X_chariot_A	Int	%MW8	Position courante du chariot A
X_chariot_B	Int	%MW10	Position courante du chariot B

3. Grafcets synchronisés du système

3.1 Grafcet bac de chargement

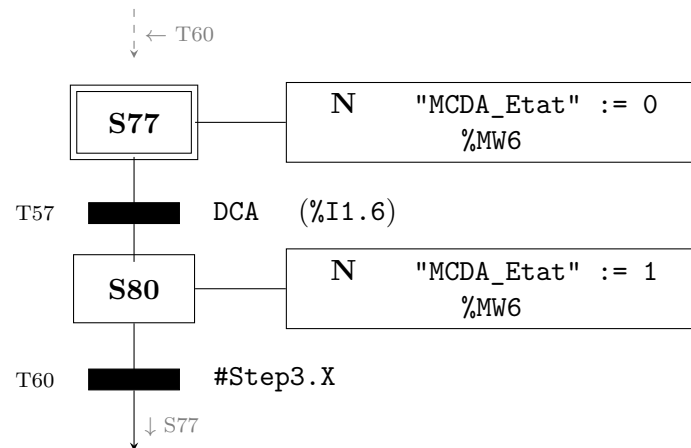


FIGURE 2 – Grafcet *Chargement A* — gestion de la zone CDA

Ce Grafcet ne pilote pas directement d'actionneurs. Son rôle est exclusivement de maintenir à jour la variable entière `MCDA_Etat (%MW6)`, qui encode l'état courant de la zone de chargement CDA.

- **Étape S77** (attente) : `MCDA_Etat := 0` — la zone est libre, aucun bac n'est présent. L'IHM affiche la zone en **jaune**.
- **Étape S80** (bac présent) : `MCDA_Etat := 1` — un bac a été déposé par l'opérateur (signal DCA). L'IHM affiche la zone en **vert**, indiquant qu'un bac est disponible pour le chariot A.

La transition **T57** est franchie sur le signal DCA (`%I1.6`), correspondant à l'action de l'opérateur qui place un bac en zone de chargement. La transition **T60** est franchie lorsque le chariot A a pris en charge le bac (`#Step3.X`), remettant la zone à l'état libre.

L'IHM interroge cycliquement `MCDA_Etat` pour mettre à jour la couleur de la zone CDA sans avoir à lire directement l'état interne du Grafcet.

3.2 Grafcet bac de déchargement

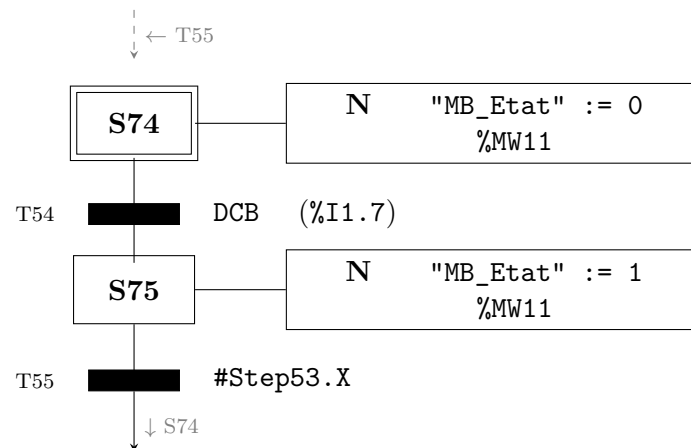


FIGURE 3 – Grafcet *Déchargement B* — gestion de la zone CDB

Ce Grafcet est symétrique au Grafcet Chargement A. Il maintient à jour la variable `MB_Etat (%MW11)`, qui encode l'état de la zone de déchargement CDB.

- **Étape S74** (attente) : `MB_Etat := 0` — la zone est libre. L'IHM affiche la zone en **jaune**.
- **Étape S75** (bac présent) : `MB_Etat := 1` — un bac a été déposé par le chariot B. L'IHM affiche la zone en **vert**, indiquant qu'un bac est disponible pour déchargement par l'opérateur.

La transition **T54** est franchie sur le signal `DCB (%I1.7)`, correspondant au dépôt du bac par le chariot B en zone de déchargement. La transition **T55** est franchie lorsque l'opérateur a retiré le bac (`#Step53.X`), remettant la zone à l'état libre.

Ce grafcet sert aussi à mémoriser l'appui sur le bouton DCB.

3.3 Grafcet BAC 1

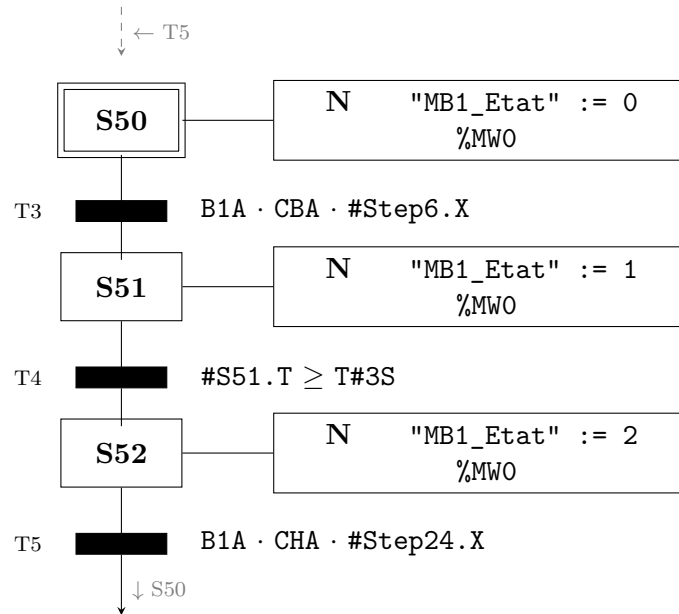


FIGURE 4 – Grafcet *BAC 1* — gestion du bain de trempage B1

Ce Grafcet gère le cycle de trempage du bac B1 et met à jour MB1_Etat (%MW0) pour l'affichage IHM.

- **S50** : bac libre — MB1_Etat := 0, IHM en **vert**.
- **S51** : trempage en cours — MB1_Etat := 1, IHM en **bleu**. La temporisation T#3S est décomptée via #S51.T.
- **S52** : trempage terminé, attente chariot — MB1_Etat := 2, IHM en **rouge**.

La transition **T3** est franchie lorsque le chariot A est positionné au-dessus du bac (B1A), crochet en bas (CBA), et en phase de dépôt (#Step6.X). La transition **T5** est franchie lors de l'enlèvement du bac par le chariot : position B1A, crochet en haut (CHA), et étape de levage active (#Step24.X).

3.4 Grafcet BAC 3

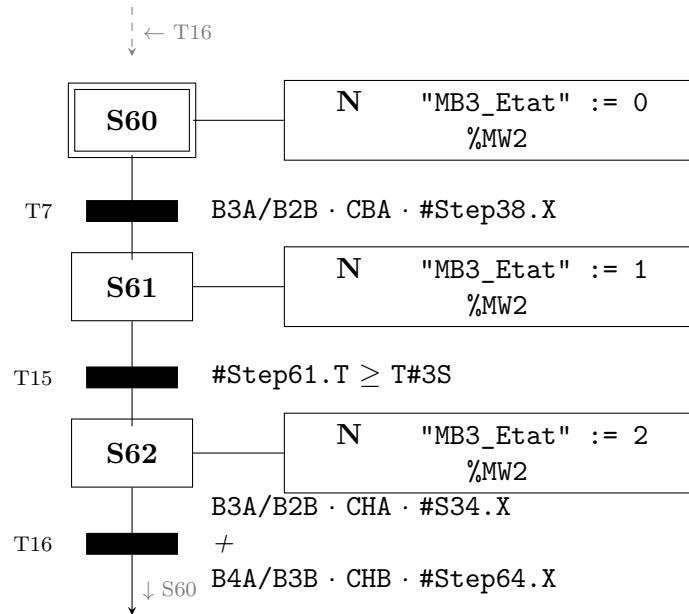


FIGURE 5 – Grafcet *BAC 3* — gestion du bain de trempage B3

Ce Grafcet est identique dans sa structure au Grafcet BAC 1, avec une différence notable au niveau de la transition **T16** : le bac 3 étant accessible depuis les deux côtés du rail, il peut être enlevé indifféremment par le **chariot A** (capteur B3A/B2B, crochet CHA, étape #S34.X) ou par le **chariot B** (capteur B4A/B3B, crochet CHB, étape #Step64.X). La condition de T16 est donc un **OU logique** entre ces deux cas.

3.5 Grafcet BAC 5

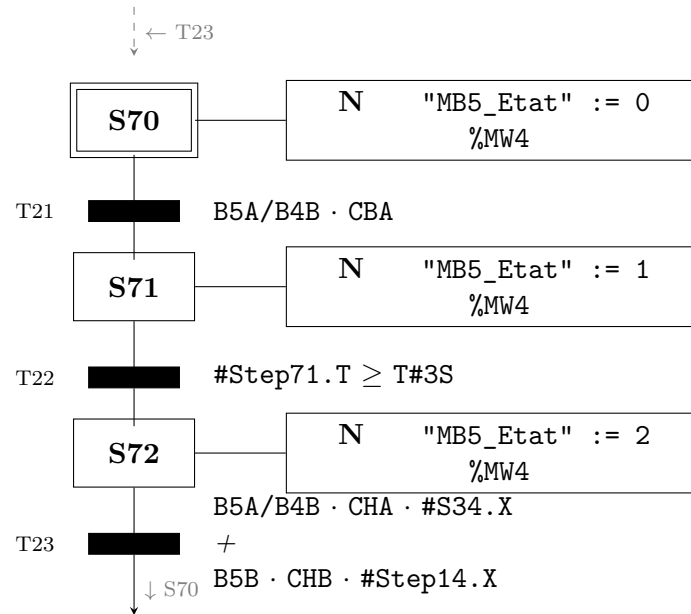


FIGURE 6 – Grafcet *BAC 5* — gestion du bain de trempage B5

Ce Grafcet est structurellement identique au Grafcet BAC 3. La particularité de BAC 5 est sa position en fin de chaîne : il est accessible uniquement par le chariot A (capteur B5A/B4B, %I0.6) pour le dépôt, mais l'enlèvement peut être effectué par le **chariot A** (B5A/B4B · CHA · #S34.X) ou par le **chariot B** (B5B · CHB · #Step14.X), d'où la condition **OU** sur T23.

3.6 Grafcet Chariot A

Le Grafcet du chariot A comporte une **étape initiale S1** et trois branches activées en OU selon la disponibilité des bacs :

- **Cycle 1 (T8)** : prise en CDA, dépôt en B1
- **Cycle 2 (T9)** : prise en B1, dépôt en B3
- **Cycle 3 (T20)** : prise en B3, dépôt en B5

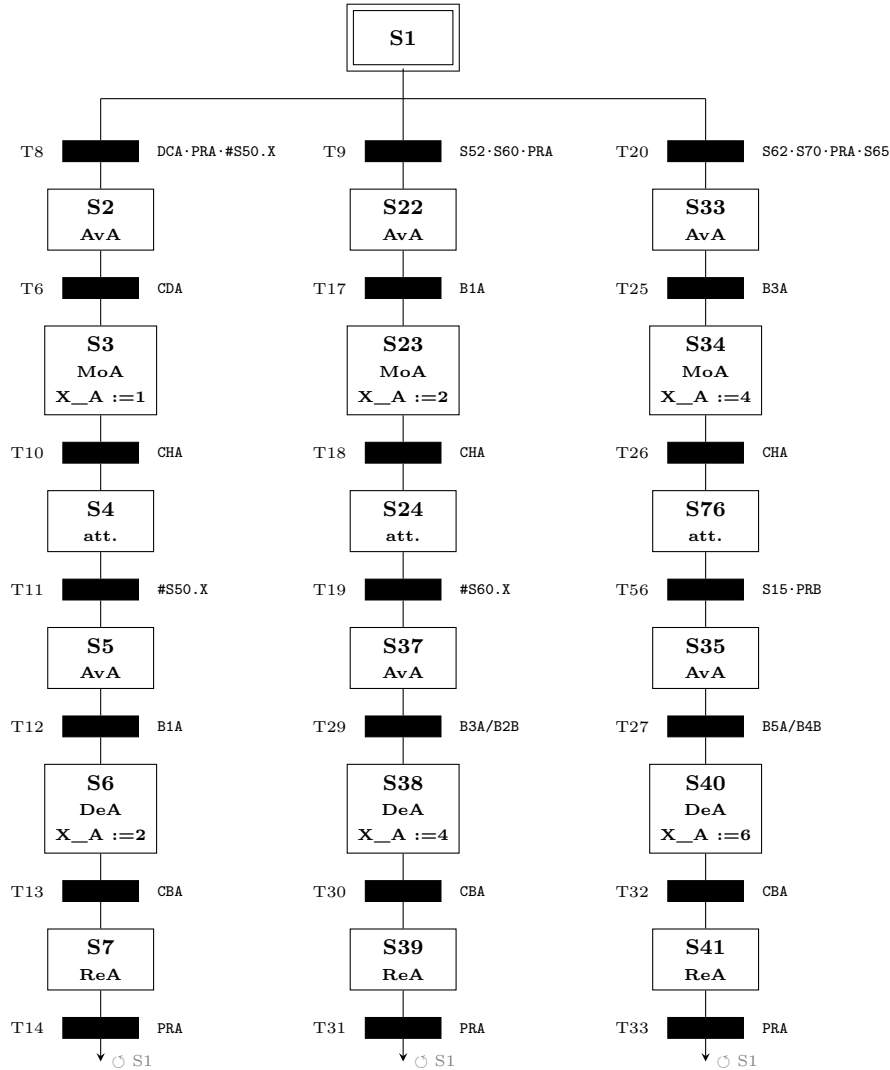


FIGURE 7 – Grafcet *Chariot A* — trois cycles de transport avec mise à jour de `X_chariot_A`

Le chariot A gère le transport des bacs depuis la zone de chargement jusqu'au cinquième bain. Chaque cycle suit le même schéma en quatre phases : **avance** vers la position cible, **prise ou dépôt** du bac (montée ou descente du crochet), **attente** de libération de la position suivante, puis **retour** en position initiale PRA.

Les étapes d'attente (**S4**, **S24**, **S76**) constituent le mécanisme central de **prévention des collisions** : le chariot A ne progresse vers la position suivante que si le Grafcet du bac cible confirme que la position est libre (conditions `#S50.X`, `#S60.X`, `#Step15.X·PRB`).

Les étapes de dépose (**S3**, **S23**, **S34**) et de prise (**S6**, **S38**, **S40**) mettent à jour la variable entière `X_chariot_A` (%MW8). Cette variable encode la position courante du chariot

sur le rail et est interrogée cycliquement par l'IHM pour afficher sa position en temps réel, sans avoir à lire directement l'état interne du Grafcet.

3.7 Grafcet Chariot B

Le Grafcet du chariot B comporte une **étape initiale S10** et deux branches activées en OU :

- **Cycle 1** (T1) : prise en B5, dépôt en CDB
- **Cycle 2** (T44) : prise en B3, dépôt en B5

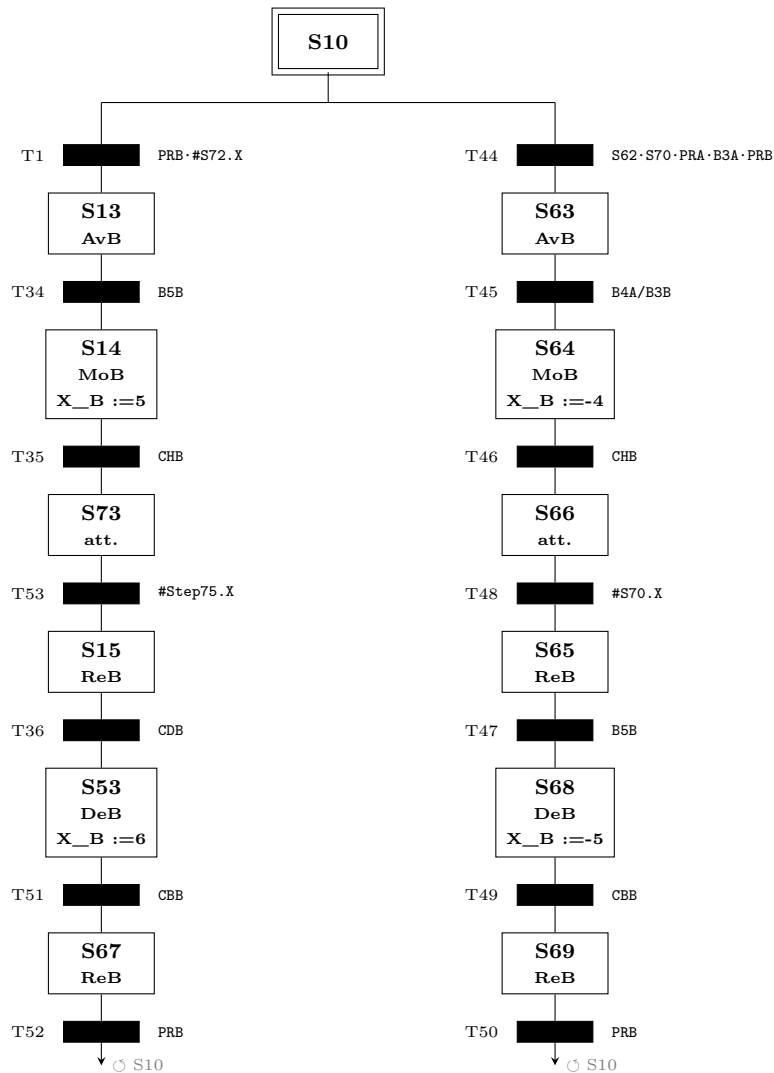


FIGURE 8 – Grafcet *Chariot B* — deux cycles de transport avec mise à jour de X_{chariot_B}

Le chariot B opère en miroir du chariot A depuis sa position d'origine PRB (côté droit du rail). Le **Cycle 1** assure le transfert des bacs traités en B5 vers la zone de déchargement CDB. Le **Cycle 2** prend en charge le transfert inter-bains de B3 vers B5. Comme pour le chariot A, les étapes d'attente (**S73**, **S66**) et la condition T56: $\#Step15.X \cdot PRB$ garantissent qu'aucun croisement ne se produit. La variable X_{chariot_B} (%MW10) est mise à jour à chaque prise ou dépôt pour l'affichage IHM.

4. Interface HMI

4.1 Présentation générale

L'interface HMI a été réalisée sous *TIA Portal WinCC* et déployée sur un pupitre **SIMATIC HMI Touch**. Elle offre une vue synoptique du système en temps réel, permettant à l'opérateur de superviser l'ensemble du cycle sans accéder au code automate.

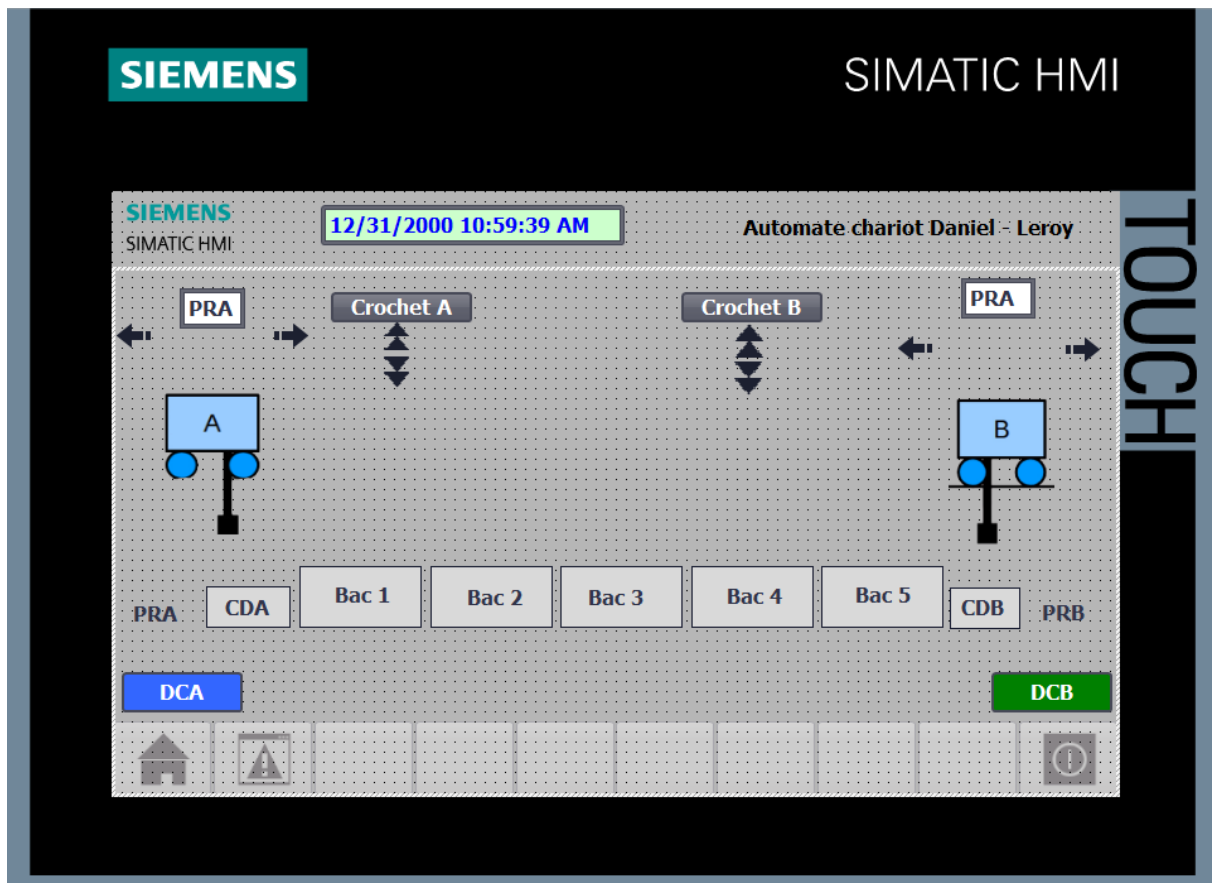


FIGURE 9 – Interface HMI — vue principale du système de trempage

4.2 Description des éléments affichés

Représentation des chariots. Les deux chariots **A** et **B** sont représentés graphiquement sur le rail, avec leur position courante reflétée par leur emplacement dans le synoptique. Chaque chariot affiche son crochet (indicateur *Crochet A* et *Crochet B*) avec des flèches indiquant la direction de déplacement possible. Les indicateurs **PRA** et **PRB** s'allument lorsque les chariots sont en position reculée (position d'origine).

Représentation des bacs. Les cinq bacs de trempage (**Bac 1** à **Bac 5**) ainsi que les zones **CDA** (chargement) et **CDB** (déchargement) sont affichés sous forme de rectangles. Leur couleur reflète l'état de la variable entière correspondante (**MB1_Etat** à **MB5_Etat**, **MCDA_Etat**, **MCDB_Etat**) :

- **Gris / blanc** : bac libre, aucune pièce présente
- **Orange** : bac en cours de trempage (temporisation active)
- **Vert** : trempage terminé, bac prêt à être enlevé

Commandes opérateur. Deux boutons de départ de cycle sont disponibles en bas de l'écran :

- **DCA** (bouton bleu, bas gauche) : signale qu'un bac a été déposé en zone de chargement CDA et lance le cycle du chariot A.
- **DCB** (bouton vert, bas droit) : signale que l'opérateur a retiré le bac de la zone de déchargement CDB.

4.3 Animation des flèches de déplacement

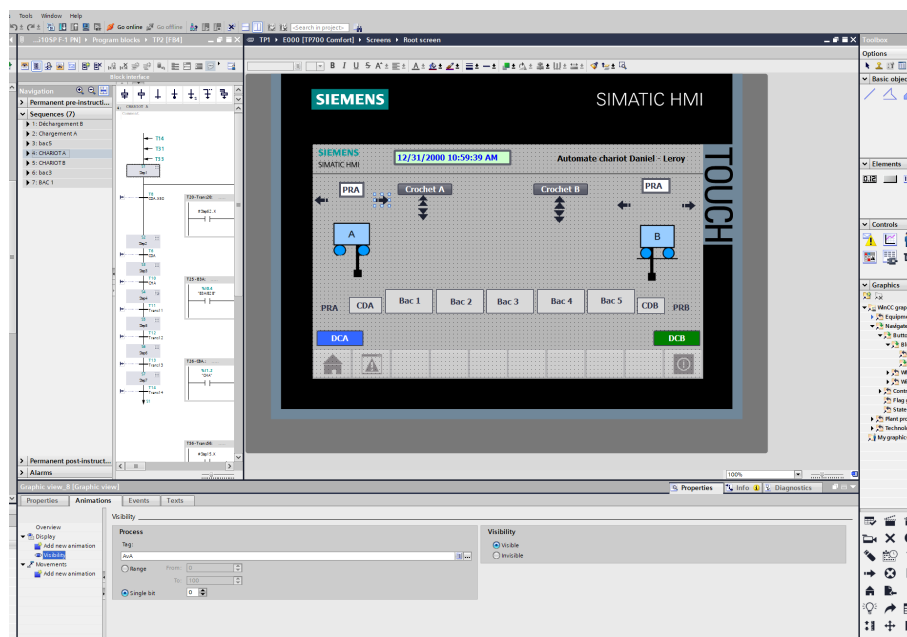


FIGURE 10 – Animation des flèches de déplacement

Les flèches de déplacement des chariots (avance et recul) sont animées via le mécanisme de **Visibility** de WinCC. Chaque flèche est un objet graphique dont la visibilité est liée à un bit d'actionneur :

TABLE 2 – Configuration des animations de visibilité — flèches de déplacement

Objet	Tag associé	Mode	Condition
Flèche avance chariot A	AvA (%Q1.4)	Single bit	Visible si = 1
Flèche recul chariot A	ReA (%Q1.5)	Single bit	Visible si = 1
Flèche avance chariot B	AvB (%Q0.3)	Single bit	Visible si = 1
Flèche recul chariot B	ReB (%Q0.2)	Single bit	Visible si = 1

La configuration s'effectue dans l'onglet **Animations** de chaque objet graphique, sous la propriété *Visibility*. Le tag de sortie correspondant à l'actionneur (ici **AvA**) est sélectionné en mode **Single bit** : l'objet est *Visible* lorsque le bit vaut 1, et *Invisible* lorsqu'il vaut 0.

Cette approche permet d'afficher en temps réel la direction de déplacement effective de chaque chariot, telle qu'elle est commandée par le Grafset, sans variable intermédiaire supplémentaire.

4.4 Configuration des boutons de départ cycle

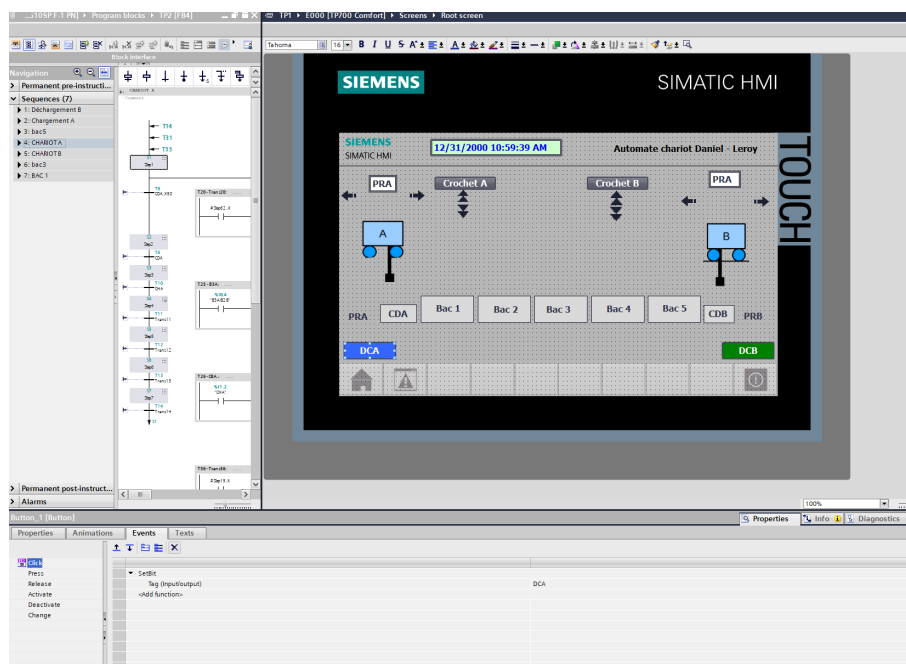


FIGURE 11 – Configuration de l'événement *Click* du bouton **DCA** — fonction **SetBit**

Les boutons **DCA** et **DCB** sont configurés dans l'onglet **Events** de WinCC. Contrairement aux flèches qui utilisent la *Visibility*, ces boutons utilisent une action de type **SetBit** déclenchée sur l'événement *Click* :

TABLE 3 – Configuration des événements — boutons de départ cycle

Bouton	Événement	Fonction	Tag associé	Adresse
DCA (bleu)	Click	SetBit	DCA	%I1.6
DCB (vert)	Click	SetBit	DCB	%I1.7

Lorsque l'opérateur appuie sur le bouton **DCA**, la fonction **SetBit** force le bit **DCA** (%I1.6) à 1, ce qui franchit la transition **T57** du Grafset Chargement A et signale qu'un nouveau bac est disponible en zone de chargement.

De façon symétrique, le bouton **DCB** active **DCB** (%I1.7), franchissant la transition **T54** du Grafset Déchargement B pour indiquer que l'opérateur a retiré le bac de la zone de déchargement.

Cette configuration permet à l'opérateur d'interagir directement avec le cycle automatisé depuis l'IHM, sans intervention sur le programme automate. Les boutons constituent ainsi l'unique point d'entrée opérateur du système.

4.5 Animation des flèches du crochet

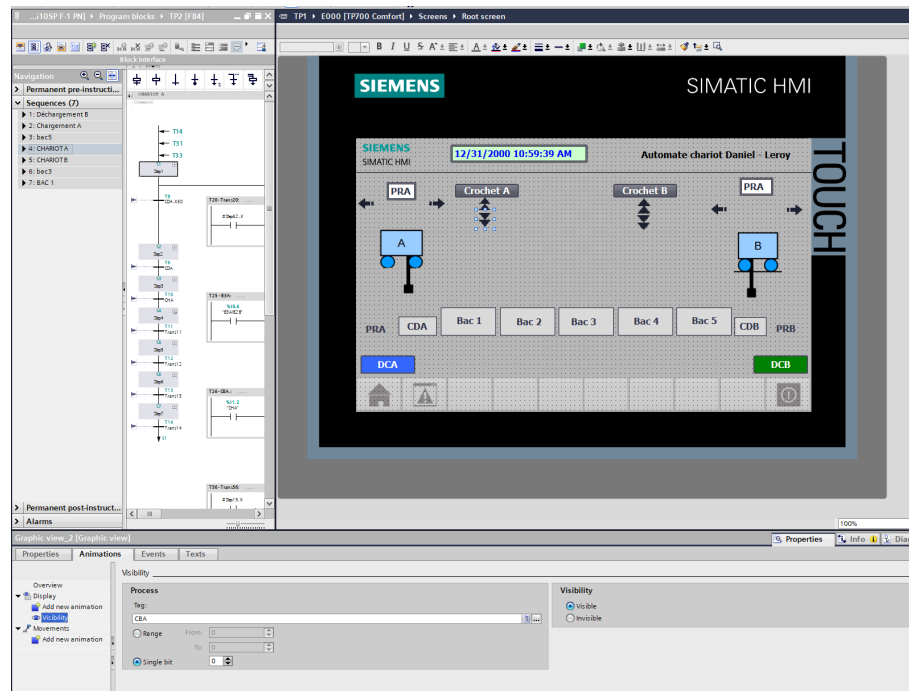


FIGURE 12 – Configuration de la visibilité de la flèche de descente du crochet A — tag CBA

Les flèches verticales associées aux crochets des chariots A et B sont animées de la même façon que les flèches de déplacement, via la propriété **Visibility** en mode **Single bit**. Chaque sens de mouvement (montée / descente) correspond à un capteur de fin de course :

TABLE 4 – Configuration des animations de visibilité — flèches de crochet

Objet	Tag associé	Adresse	Condition
Flèche descente crochet A	CBA	%I1.3	Visible si = 0
Flèche montée crochet A	CHA	%I1.2	Visible si = 0
Flèche descente crochet B	CBB	%I1.5	Visible si = 0
Flèche montée crochet B	CHB	%I1.4	Visible si = 0

La logique retenue est la suivante : la flèche de descente est visible tant que le capteur *crochet en bas* (CBA) n'est pas atteint (bit à 0), indiquant que le crochet est encore en mouvement ou en position haute. De façon symétrique, la flèche de montée est visible tant que CHA n'est pas activé.

Cette configuration permet de visualiser en temps réel le mouvement vertical des crochets : lorsque le crochet est en position basse (CBA = 1), la flèche disparaît, confirmant visuellement l'atteinte de la position de prise ou de dépôt du bac.

4.6 Affichage de la position des chariots

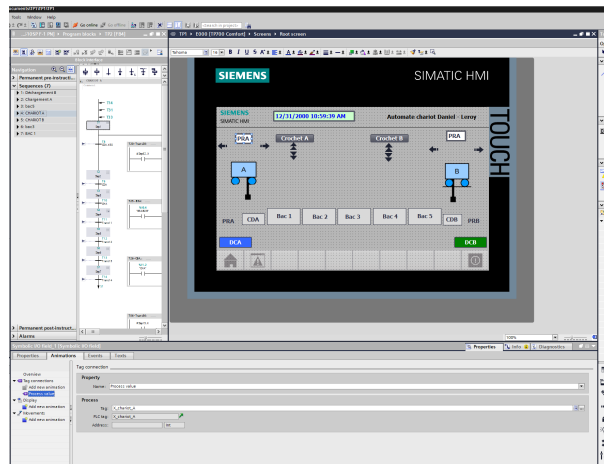


FIGURE 13 – Configuration du champ symbolique de position — tag `X_chariot_A`

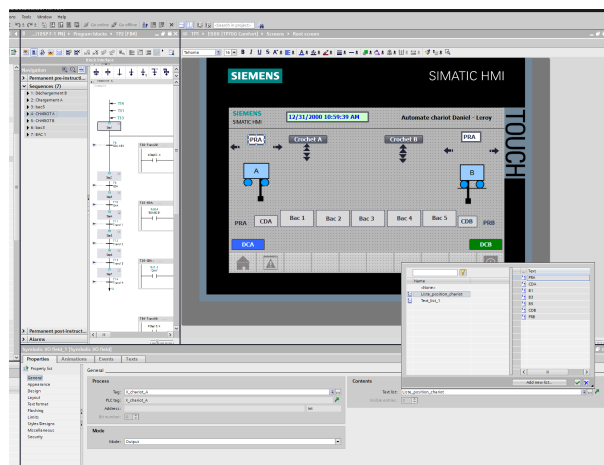


FIGURE 14 – Configuration du champ symbolique de position — tag `X_chariot_A`

La position courante de chaque chariot est affichée sur l'IHM via un **Symbolic I/O field** (champ d'E/S symbolique). Cet objet WinCC convertit la valeur entière de `X_chariot_A` (`%MW8`) en un texte lisible, grâce à une **liste de textes** associée.

4.6.1 Liaison à la variable de position

Dans l'onglet **Animations** → *Tag connections* → *Process value*, le champ est lié au tag `X_chariot_A` de type `Int`. Chaque valeur entière correspond à une position physique sur le rail.

4.6.2 Liste de textes `Liste_position_chariot`

Une liste de textes nommée `Liste_position_chariot` est définie dans WinCC et associée au champ symbolique. Elle établit la correspondance entre les valeurs numériques de `X_chariot_A` et les étiquettes affichées :

TABLE 5 – Liste de textes Liste_position_chariot

Valeur de X_chariot_A	Texte affiché
0	PRA
1	CDA
2	B1
3	B3
4	B5
5	CDB
6	PRB

Le champ est configuré en mode **Output** (lecture seule), ce qui signifie qu'il affiche uniquement la position courante sans permettre de modification depuis l'IHM. Le même principe est appliqué pour le chariot B avec le tag X_chariot_B (%MW10).

Cette approche évite d'afficher des valeurs brutes incompréhensibles pour l'opérateur et garantit une lecture intuitive de la position des chariots en temps réel.

4.7 Animation couleur des bacs

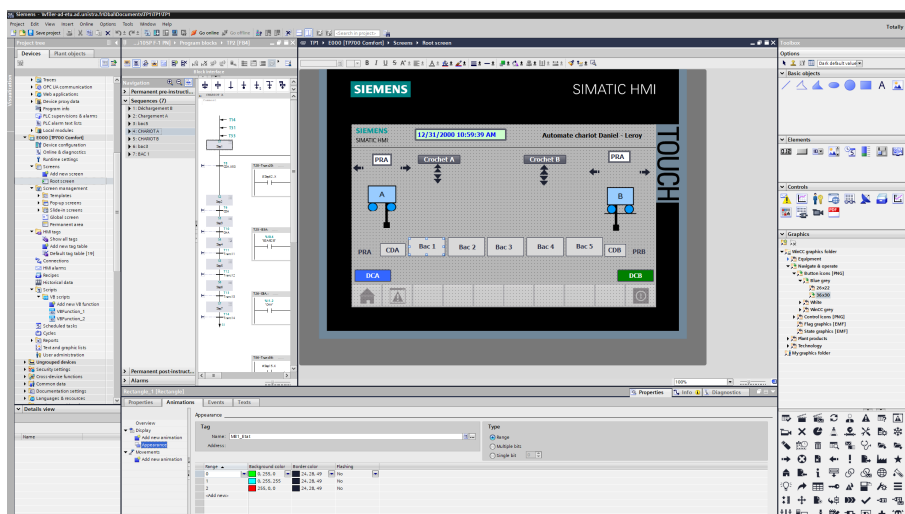


FIGURE 15 – Configuration de l’animation *Appearance* du Bac 1 — tag MB1_Etat, trois plages de couleur

Chaque bac est représenté par un **Rectangle** dont la couleur de fond change dynamiquement selon l’état du bac. La configuration s’effectue dans l’onglet **Animations** → *Display* → *Appearance*, en mode **Range**.

4.7.1 Liaison à la variable d’état

Le rectangle du Bac 1 est lié au tag MB1_Etat (%MW0) de type Int. Trois plages de valeurs définissent trois couleurs distinctes :

TABLE 6 – Configuration des couleurs — animation *Appearance* des bacs

Valeur	État	Couleur fond	Signification
0	Libre	Vert	Bac vide, position disponible
1	En trempage	Cyan/Jaune	Pièce en cours de traitement
2	Terminé	Rouge	Trempage terminé, attente chariot

Le même principe est appliqué à l’ensemble des bacs et zones :

TABLE 7 – Récapitulatif des tags d’état liés aux animations couleur

Élément	Tag	Adresse
Bac 1	MB1_Etat	%MW0
Bac 2	MB2_Etat	%MW1
Bac 3	MB3_Etat	%MW2
Bac 4	MB4_Etat	%MW3
Bac 5	MB5_Etat	%MW4
Zone CDB	MCDB_Etat	%MW5
Zone CDA	MCDA_Etat	%MW6

Cette approche centralise la logique d'état dans l'automate via les Grafjets dédiés, et délègue uniquement l'affichage à l'IHM. L'opérateur peut ainsi suivre en un coup d'œil la progression des pièces dans la chaîne de trempage, sans interpréter de valeurs numériques.

4.8 Animation du déplacement des chariots — Scripts VB

Pour simuler le déplacement physique des chariots sur le synoptique, deux scripts **VBScript** ont été développés dans WinCC. Ces fonctions déplacent horizontalement le groupe graphique représentant chaque chariot en modifiant sa propriété **Left** (position horizontale en pixels) selon la valeur de **X_chariot_A** ou **X_chariot_B**.

4.8.1 VBFunction_1 — Chariot A

Listing 1 – Script de déplacement du chariot A

```
Sub VBFunction_1()  
  
    Dim pos  
    pos = SmartTags("X_chariot_A")  
  
    Select Case pos  
        Case 0 : HmiRuntime.Screens("Root screen") _  
                .ScreenItems("Graphic view_4").Left = 0  
        Case 1 : HmiRuntime.Screens("Root screen") _  
                .ScreenItems("Graphic view_4").Left = 14  
        Case 2 : HmiRuntime.Screens("Root screen") _  
                .ScreenItems("Graphic view_4").Left = 745  
        Case 3 : HmiRuntime.Screens("Root screen") _  
                .ScreenItems("Graphic view_4").Left = 149  
        Case 4 : HmiRuntime.Screens("Root screen") _  
                .ScreenItems("Graphic view_4").Left = 356  
    End Select  
  
End Sub
```

4.8.2 VBFunction_2 — Chariot B

Listing 2 – Script de déplacement du chariot B

```
Sub VBFunction_2()  
  
    Dim pos  
    pos = SmartTags("X_chariot_B")  
  
    Select Case pos  
        Case 4 : HmiRuntime.Screens("Root screen") _  
                .ScreenItems("Graphic view_1").Left = 356  
        Case 5 : HmiRuntime.Screens("Root screen") _  
                .ScreenItems("Graphic view_1").Left = 563  
        Case 6 : HmiRuntime.Screens("Root screen") _  
                .ScreenItems("Graphic view_1").Left = 665  
    End Select  
  
End Sub
```

```

    Case 7 : HmiRuntime.Screens("Root screen") _
              .ScreenItems("Graphic view_1").Left = 700
    End Select

End Sub

```

Ces scripts sont déclenchés cycliquement par le **Scheduler** de WinCC à chaque variation des tags X_chariot_A et X_chariot_B. La correspondance entre valeur et position en pixels est établie par calibration visuelle sur le synoptique :

TABLE 8 – Correspondance valeurs / positions pixel pour l’animation des chariots

Valeur	Position	Left chariot A (px)	Left chariot B (px)
0	PRA	0	—
1	CDA	14	—
2	B1	745	—
3	B3	149	—
4	B5/B3	356	356
5	B5	—	563
6	CDB	—	665
7	PRB	—	700

Remarque : le langage de script utilisé est le *VBScript* intégré à *WinCC Comfort*. La fonction `SmartTags()` permet d’accéder directement à la valeur courante d’un tag HMI sans déclaration préalable de variable intermédiaire.

5. Supervision complémentaire — Node-RED

En complément de l'IHM SIMATIC, une interface de supervision web a été développée sous **Node-RED**, permettant de surveiller le système depuis n'importe quel navigateur sans nécessiter de client WinCC.

5.1 Architecture du flux Node-RED

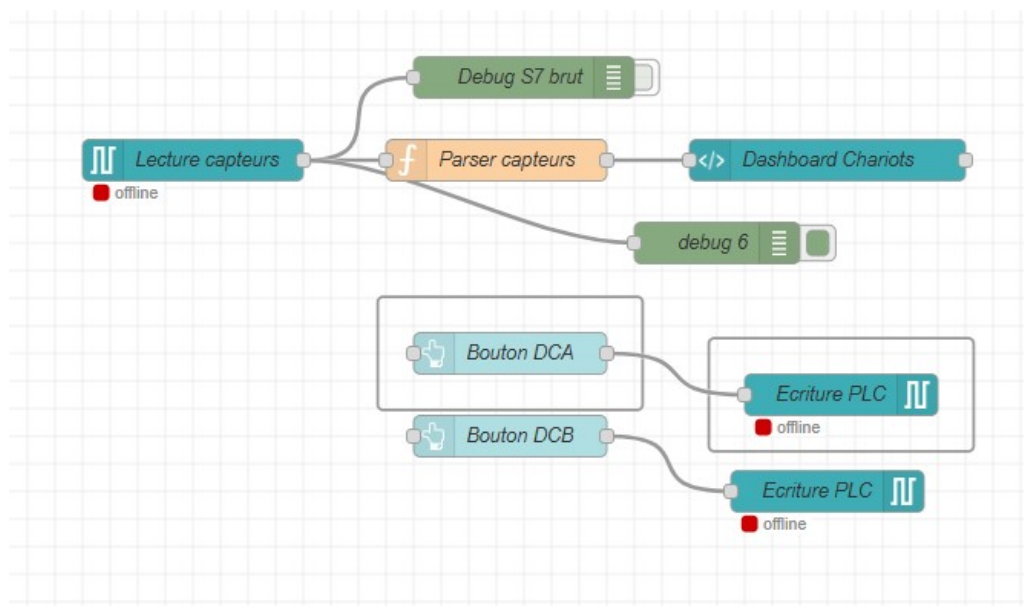


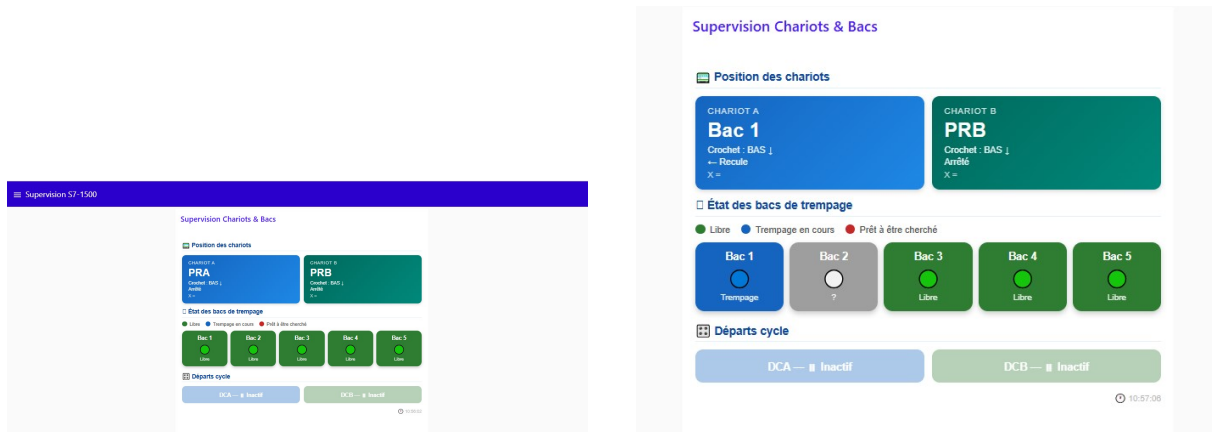
FIGURE 16 – Flux Node-RED — lecture capteurs, parsing et dashboard

Le flux est organisé en deux chaînes parallèles :

Chaîne de supervision (lecture). Un nœud *Lecture capteurs* interroge cycliquement l'automate S7-1500 via le protocole S7. Les données brutes sont transmises à un nœud de traitement *Parser capteurs* (fonction JavaScript) qui interprète les valeurs des tags (MB*_Etat, X_chariot_A, X_chariot_B) et alimente le *Dashboard Chariots* en données formatées. Un nœud *Debug S7 brut* permet de vérifier les trames reçues.

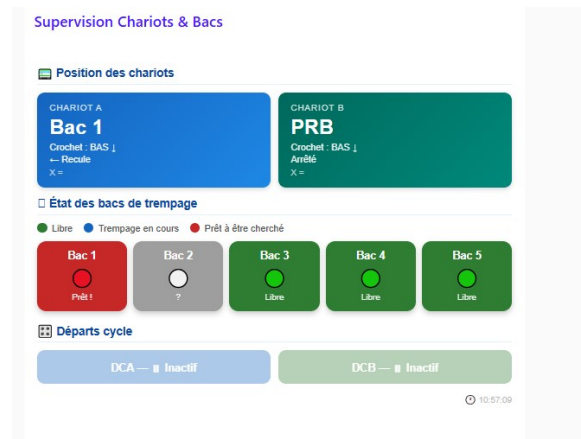
Chaîne de commande (écriture). Deux boutons *Bouton DCA* et *Bouton DCB* sont exposés sur le dashboard. Un clic déclenche un nœud *Écriture PLC* qui écrit directement le bit correspondant (%I1.6 ou %I1.7) dans l'automate, reproduisant le comportement des boutons de l'IHM SIMATIC.

5.2 Interface de supervision web



(a) État initial — tous les bacs libres, chariots en PRA/PRB

(b) Cycle en cours — Bac 1 en trempage, chariot A sur Bac 1



(c) Bac 1 prêt à être cherché (rouge), Bac 2 inconnu (gris)

FIGURE 17 – Captures de l'interface de supervision Node-RED

5.3 Description du dashboard

Le dashboard est organisé en trois sections :

Position des chariots. Deux cartes colorées affichent en temps réel la position de chaque chariot (issue de `X_chariot_A` et `X_chariot_B`), l'état du crochet (haut/bas), et le mouvement en cours (avance, recule, arrêté). La carte du chariot A est en bleu, celle du chariot B en vert foncé.

État des bacs de trempage. Cinq indicateurs circulaires représentent les bacs B1 à B5. Leur couleur reflète la valeur de `MB*_Etat` selon la légende suivante :

TABLE 9 – Code couleur des bacs sur le dashboard Node-RED

Couleur	État	Valeur MB*_Etat
Vert	Libre	0
Bleu	Trempage en cours	1
Rouge	Prêt à être cherché	2

Départs cycle. Deux boutons **DCA** et **DCB** reproduisent les commandes opérateur. Leur état (Inactif / Actif) est affiché en temps réel.

5.4 Avantages de la supervision Node-RED

Cette supervision complémentaire présente plusieurs avantages par rapport à l'IHM SIMATIC :

- **Accessibilité** : interface consultable depuis n'importe quel navigateur web sur le réseau local, sans logiciel propriétaire.
- **Flexibilité** : le flux Node-RED est modifiable rapidement sans recompilation du projet TIA Portal.
- **Lisibilité** : affichage textuel explicite de la position et du mouvement des chariots, plus intuitif que la position pixel de l'IHM SIMATIC.

6. Conclusion

Ce travail pratique nous a permis de concevoir et d'implémenter une commande complète d'un système de trempage industriel à deux chariots, en appliquant le formalisme **Grafcet** dans un contexte de synchronisation avancée.

L'architecture retenue, basée sur **7 Grafkets indépendants** et communicants, s'est révélée particulièrement adaptée à la complexité du système. La décomposition en entités autonomes un Grafcet par chariot, un par bac et un par zone de transfert a permis de maîtriser la synchronisation inter-Grafkets via des variables d'échange entières (**MB*_Etat**, **X_chariot_A**, **X_chariot_B**), garantissant à la fois la lisibilité du code et la robustesse de la commande.

La contrainte principale du système l'interdiction absolue de croisement entre les deux chariots sur un rail commun a été résolue par l'introduction d'**étapes d'attente** conditionnées par l'état des Grafkets voisins. Cette approche décentralisée évite tout gestionnaire de ressource centralisé et rend le système facilement extensible.

Sur le plan de la supervision, deux interfaces complémentaires ont été développées. L'**IHM SIMATIC WinCC** offre une vue synoptique animée, avec déplacement des chariots piloté par des scripts VBScript, affichage couleur des bacs et commandes opérateur intégrées. La supervision **Node-RED**, accessible depuis n'importe quel navigateur, apporte une couche de supervision web légère et flexible, communiquant directement avec l'automate S7-1500 via le protocole S7.

Ce projet nous a permis de consolider nos compétences en automatisme industriel, notamment la programmation en langage GRAPH sous TIA Portal, la conception d'interfaces HMI et l'intégration de solutions de supervision modernes. Il illustre concrètement les enjeux de la synchronisation dans les systèmes multi-actionneurs partageant des ressources communes, problématique centrale dans l'industrie manufacturière.