REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMERI de Tizi-Ouzou

Faculté de Génie Electrique et d'Informatique
Département d'Automatique

PROJET DE FIN D'ETUDES

En vue de l'obtention du diplôme

D'INGENIEUR D'ETAT EN AUTOMATIQUE

Thème

Développement d'un Programme de Détection et d'Affichage de Défauts pour Aide à la Maintenance d'une Machine Assembleuse Pose A Plat (PAP)

Proposé par :
M^r^. AOUINI.K

Présenté par :
M^r^. **BELLAL Samir**
M^r^. **TEMZI Lounis**

Dirigé par :
M^r^. DJENNOUN.S

Soutenu le : 14/ 07 /2009 Devant le jury d'examen composé de :

M^r^. **CHARIF.M**
M^r^. **MAIDI.A**
M^r^. **HADDOUCHE.R**

Promotion 2009

Michelin Algérie-Bach Djerrah-Alger

Remerciements

Le travail que constitue cette thèse a été réalisé au sein de

L'entreprise Michelin Algérie

Division maintenance - Secteur MS3

Nous remercions tout le personnel de la direction technique et générale de Michelin Algérie.

Notre gratitude à messieurs : Khider.S, Ouamara.S, Guennoun.K, Aouini.K.

Nous sommes absolument très reconnaissants à Mr Tamani Rabah qui nous a toujours aidés.

De même nous remercions tout le personnel du secteur maintenance MS3.

Nous sommes très reconnaissants à notre promoteur Monsieur Djennoun Said (Professeur à l'UMMTO) pour le suivi et l'évaluation du travail.

Nos remerciements aux membres de jury qui ont accepté de juger notre travail.

Enfin, nous remercions toute personne qui a contribué de prés ou de loin à la réalisation de ce mémoire.

Bellal Samir / Temzi Lounis

Dédicaces

Je dédie ce travail à mes parents mes plus chers, qui ont toujours été à mes côtés, qui m'ont toujours aidé à surmonter mes problèmes.

A mon frère Karim.

A mon frère Hamid et sa femme Linda et ses deux enfants le méchant H'mimi et le petit Samy.

A mes deux sœurs Malika et Nacéra, leurs enfants et leurs maris Karim et Saadi.

A la mémoire de mon frère Ahmed.

A la mémoire de mes grand-parents et grand-mères.

A toute la famille Bellal.

A mes oncles et tantes.

A tous mes amis depuis l'enfance.

A tous mes enseignants depuis le primaire.

Samir Bellal

Dédicaces

Je dédie ce mémoire à mes très chers parents que j'ai trouvé toujours à mes cotés durant un parcours de sacrifices dans mes études mais plein de réussite à qui je dois tout et je ne leur rendrai jamais assez, que dieu les protège.

A mes très chers frères Mourad, Ali, Mouloud et Samir, que j'aime beaucoup.

A mes belles sœurs Assia et Nawel et les petits Chérif,Belaid,Yanis et Nassim.

A ma grand-mère Terkia.

A la mémoire de mes grand-parents.

A tous mes cousins et cousines .

A tous mes oncles et tantes.

A toutes la famille Temzi.

A tous mes amis et ceux qui m'ont aidé de prés ou de loin a la réalisation de ce travail.

A mes professeurs.

Temzi Lounis

SOMMAIRE

-Préambule-

Michelin monde :

C'est au 28 Mai 1889 qu'Edouard Michelin lance la création de la société de fabrication du pneumatique en France (son frère André Michelin ne le rejoindra que bien plus tard) à Clermont Ferrand. Actuellement l'entreprise est une multinationale représentée par :

- Plus de 12 marques différentes.
- 68 sites de production dans 19 pays y compris l'Algérie (Usine d'Alger).
- Un centre de technologie réparti sur 3 continents (Asie, Amérique, Europe).
- Près de 116 000 employés de toutes cultures sur tous les continents dont 4 000 chercheurs en Europe, aux Etats-Unis, en Asie.
- 6 plantations d'hévéa réparties dans 2 pays (Brésil, Nigéria).
- Une présence commerciale dans plus de 170 pays avec 17,7 du marché.

Historique du groupe Michelin en bref :

1889 : Edouard Michelin lance la création de la société de fabrication du pneumatique en France.

1895 : Michelin fait rouler la première voiture sur pneus : l'Eclair.

1898 : **Naissance du bonhomme Michelin, Bibendum.**

1900 : parution du premier guide rouge.

1935 : Michelin prend le contrôle de Citroën, qu'il gardera jusqu'en 1975.

1946 : Michelin dépose le brevet d'un pneu révolutionnaire : Le Radial.

1979 : Associé à Ferrari, le pneu radial Michelin est champion du monde en formule 1.

1981 : Le 1er pneu radial pour avion, Michelin Air X, est développé.

1990 : Michelin absorbe Uniroyal-Goodrich Tire Company au Etats-Unis.

1994 : La nouvelle gamme de pneu '' Energy'' permet d'économiser du carburant.

2001 : Michelin conçoit le plus grand pneu de génie civil du monde.

2005 : Michelin remporte les titres de champion du monde de F1, de Rallye et de GP Moto.

Michelin Algérie :

Michelin a construit son usine algérienne en 1963 et l'intègre dans la SATI (Société Algérienne des Techniques Industrielles créée le 30 juin 1959). La SATI (Désormais Michelin Algérie) est constituée de deux entités : Une société commerciale qui vend tous les types de pneumatique Michelin et l'unité de production qui fabrique des pneus poids lourd. Pour des raisons de sécurité suite aux événements algériens, la SATI met en sommeil son unité de production en 1993. Huit ans plus tard, en novembre 2001 Michelin décide une compagne de redémarrage pour fabriquer des pneus poids lourd correspondant aux besoins du marché local. Le 12 Août 2002, Michelin crée la société ‘’Michelin Algérie ‘’, entreprise de droit algérien.

Michelin Algérie 2009 :

L'année 2008 riche en événements et changements est terminée. C'est aussi une année des progrès réels et significatifs dans tous les domaines : Sécurité, qualité, service aux clients. L'année 2009 commence dans une conjoncture difficile dont l'engagement quotidien et la portance de l'entreprise sont les clefs de la réussite.

En cette année, Michelin concentre ses efforts pour réussir plusieurs défis :

- Pérenniser ses progrès en sécurité par application des méthodes Michelin : PSA, OPS, DRS.
- Améliorer fortement la productivité et la qualité à l'atelier préparation.
- Continuer à réduire la perte pour approcher les standards Michelin.
- Réussir tous les jours ses engagements production grâce à la productivité, la gestion rigoureuse du présentéisme et une disponibilité améliorée de produits de la PREP.

Introduction générale

De nos jours, l'automatisation joue un rôle phénoménal dans le développement de l'industrie vue les améliorations qu'elle apporte et qu'elle ne cessera d'apporter aux différentes industries. Son application s'accroit rapidement et elle s'étend de plus en plus des petits appareils aux grandes installations industrielles.

Michelin Algérie comme toute grande entreprise utilise des machines très complexes, Cette complexité rend la tâche difficile aux maintenanciers, notamment le cas de l'assembleuse pose à plat contrôlée par un automate dont le programme ne traite pas les cas de défaillance (Défauts). Une petite panne d'absence d'un capteur ou un mal fonctionnement d'un vérin peut prendre des heures ou des jours, ou même des fois nécessite l'intervention du bureau d'études pour la localiser.

Pour remédier à ce problème, la direction technique de Michelin Algérie nous a confié le travail d'élaboration d'un programme de détection de défauts, et les afficher sur une interface homme/machine.

Afin de mener à bien notre travail, nous avons adopté la méthodologie suivante :
Dans une première étape, une étude détaillée du fonctionnement de la machine a été menée. Ensuite, nous avons recensé les différents défauts qui peuvent survenir sur la machine.

Nous avons classé les défauts selon trois types, à savoir défauts matériels, défauts liés aux transitions et défauts d'alimentation et défaut tringle.

Dans une seconde étape, il nous a fallu étudier et maitriser le logiciel d'exploitation de l'automate ainsi que l'outil d'affichage.

En dernière étape, nous avons proposé des solutions pour diagnostiquer les défauts et les afficher, ces solutions ont été implantées sous le logiciel RSLogix 500 et PanelBuilder 32.

Notre mémoire est organisé comme suit :
Chapitre I : Description de la machine.
Chapitre II : Présentation de l'automate ALLEN Bradley SLC 500.
Chapitre III : Le logiciel de programmation RSLogix 500.
Chapitre IV : L'interface homme/machine PanelView 550 et son logiciel PanelBuilder 32.
Chapitre V : Solution proposée pour le diagnostic et l'affichage des défauts.

Chapitre I

Description de la machine

Introduction :

Le pneumatique est une enveloppe que l'on fixe à la jante des roues des véhicules pour absorber les irrégularités du sol et favoriser leur déplacement sans glissement.
Un pneumatique assure plusieurs fonctions vitales pour la sécurité de tous les usagers de la route. Il sert à rouler, porter la charge, assurer le guidage, transmettre les efforts freineurs et moteurs, amortir les bruits et les vibrations mécaniques. Pour assurer ces fonctions, le pneumatique doit adhérer, assurer la trajectoire (Comportement routier), résister à l'usure, être endurant, être confortable, faire le moins de bruit possible mais aussi être capable de supporter le poids du véhicule et de ses passagers. Ces caractéristiques sont le résultat d'une haute précision dans la composition du pneumatique. C'est un produit complexe constitué de plusieurs éléments : Un véritable produit de haute technologie. En effet, le pneumatique est soumis à de multiples contraintes, il doit s'adapter à des conditions d'utilisation différentes et doit satisfaire des besoins différents. Chaque élément du pneumatique est conçu pour répondre à ces nécessités.

I. Les principaux composants d'un pneumatique :

GI : Gomme Intérieure, volume de gomme.

RP/T : Renfort de la nappe carcasse.

RIC : Renfort Intérieur Carcasse.

NC : Nappe Carcasse, armature principale de la carcasse, Sert de liaison entre les bourrelets et le sommet.

BT : Bourrage Tringle, Sert à positionner la tringle.

GB : Gomme de Bordure, Sert à enrober les extrémités de la NC ou du raidisseur.

TT : Tringles Tressées, Servent à accrocher l'enveloppe sur la jante.

PS : Pied Sommet, désolidarise les extrémités des nappes sommet de la nappe carcasse.

II. Processus de fabrication d'un pneu Michelin :

La fabrication d'un pneumatique Michelin passe par plusieurs étapes commençant de la préparation des semi-finis jusqu'au contrôle de la qualité selon l'hiérarchie suivante :

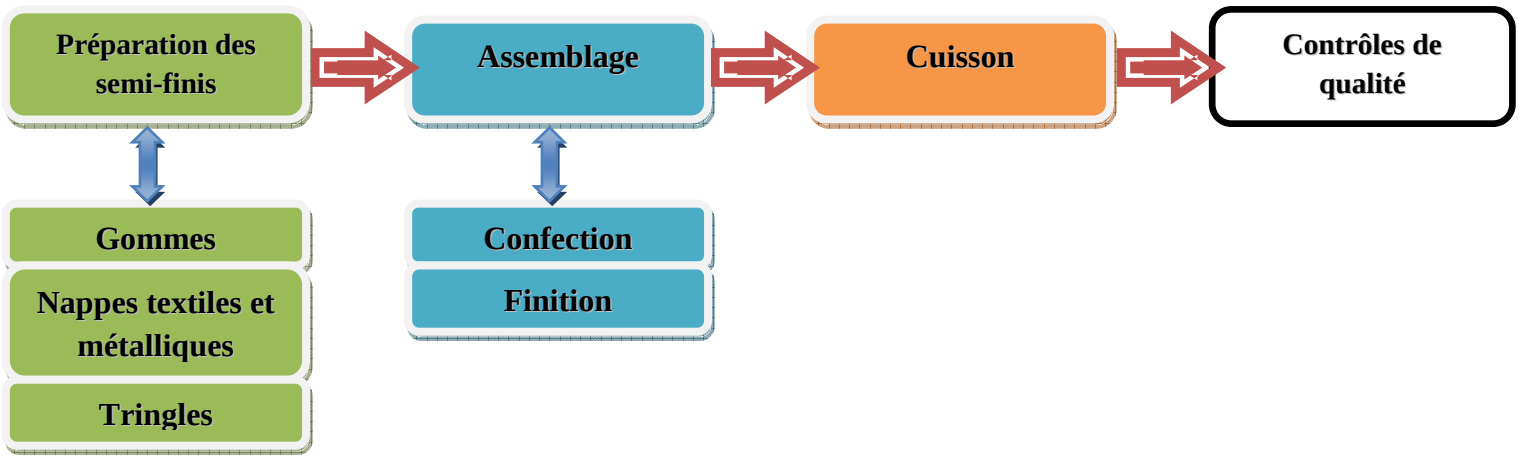


Figure I-1 : Etapes de fabrication d'un pneumatique.

III. L'assembleuse pose à plat (PAP) :

III.1. Description et rôle :

L'assembleuse PAP est une machine qui se trouve sur le secteur d'assemblage, servant à assembler les différents produits semi-finis (Produits plats et profilés, nappes et tringles) nécessaires à la fabrication d'une carcasse à des côtes bien déterminées. Le principe est de superposer, à plat, les produits suivant un ordre d'assemblage et de positionnement bien précis, sur un tambour en rotation. Elle est munie de deux rouletages : galet et piano, utiles pour coller les produits l'un sur l'autre. Les produits sont portés sur 4 pose-produits montés sur une table qui permet par rotation de passer de l'un à l'autre pour faire face à l'opérateur.

La machine est gérée par un automate programmable Allen Bradley gamme SLC 500 avec deux modules d'entrées, et deux modules de sorties TOR.

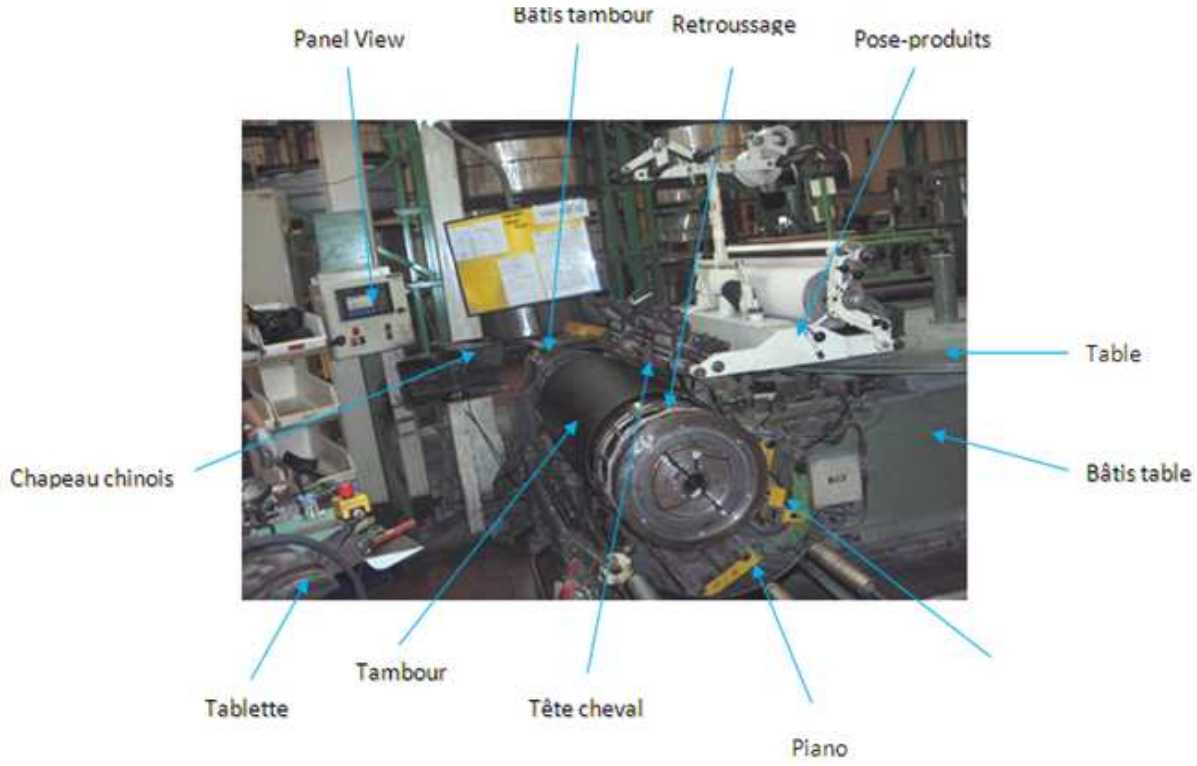


Figure I.2 : Assembleuse PAP.

III.2. Eléments de la machine :

Elle se compose des éléments suivants :

Tambour : Sert de gabarit et de support aux produits nécessaires à la confection, son diamètre est variable à savoir : diamètre minimum, intermédiaire, ou maximum, entrainé par un moteur asynchrone qui assure sa rotation après validation sur une pédale. Il est constitué de :

- deux demi-manchons.
- deux annaux à gorge.
- des ceintures de rappel.
- des éléments du tambour.
- deux paniers de retroussage (côté référence et anti-référence) chacun avec :
    * Un bras de retroussage.
    * Un ressort de retroussage.

Bâti tambour : Sert à supporter le tambour et permettre sa rotation.

Table : Sert à supporter les différents pose-produits.

Bâti table : Sert à supporter la table et permettre sa rotation.

Pose-produits : Au nombre de 4, Servent à supporter et permettre la pose des produits.

Tête de cheval : Élément du bâti table servant à l'entraînement de la nappe carcasse et du complexe raidisseur. Son fonctionnement se fait par un vérin double effet.

Tablette : Support de l'outillage de confection, dispose de deux fours, l'un pour les produits butyles (Gomme intérieure qui renforce la chambre à air), l'autre pour les produits non butyles.

Rouletage piano : Entrainé par un vérin double effet, il assure le rouletage du PT en le serrant contre les produits de la couche inférieure.

Rouletage galet : comme le rouletage piano, il assure le rouletage de la NC.

Lasers : au nombre de 16, placés au dessus de la machine, émettent des faisceaux lumineux en lignes rouges sur le tambour pour guider l'opérateur à poser les produit avec précision.

Chapeau chinois : entrainé par un vérin rotatif lui permettant de basculer entre deux positions : position ''solvant butyle accessible-solvant non butyle inaccessible'' pour les produits ayant besoin d'un solvant butyle, et position ''solvant non butyle accessible-solvant butyle inaccessible'' pour les produits ayant besoin d'un solvant non butyle.

III.3. L'armoire électrique :

Elle contient les éléments suivants :

- Un automate programmable Allen Bradley gamme SLC 500 avec deux modules d'entrées, deux modules de sorties, et une alimentation, imbriqués sur un seul rack.
- Convertisseur RS232/DH.
- 4 transformateurs :
  400V/230V 1600VA.
  400V/230V 6300VA.
  400V/115V 1000VA.
  400V/24V 100VA.
- Une alimentation stabilisée 24V/4.3A.
- Un filtre secteur.
- Des disjoncteurs.

- Des relais.
- Des contacts.
- Deux régulateurs de température à sortie analogique.

III.3.1. Positionnement des éléments sur l'armoire électrique :

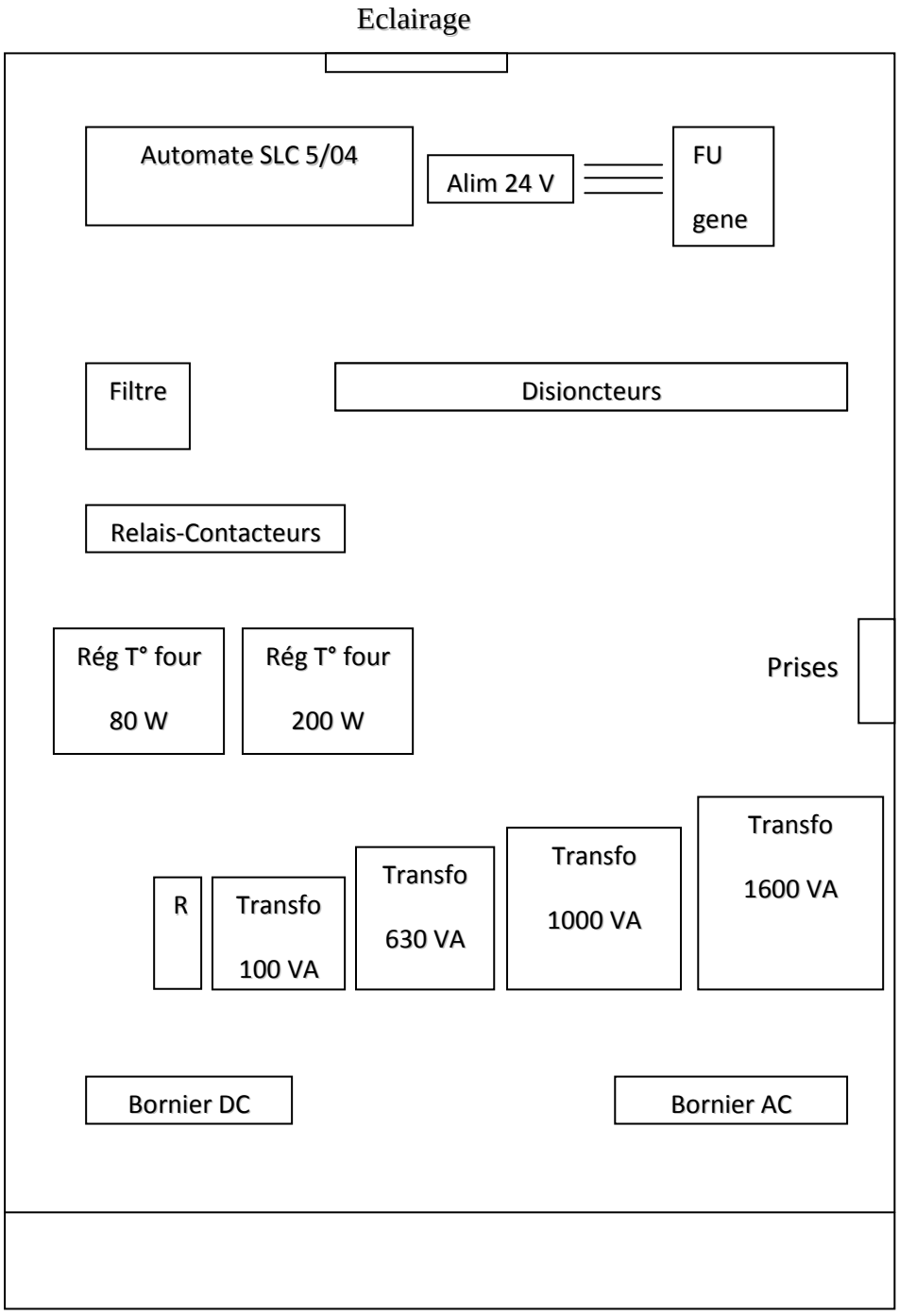


Figure I-3 : L'armoire électrique.

III.4. L'armoire pneumatique :

Elle comprend les éléments suivants :

- Un robinet 2 voies / 3 orifices.
- Un filtre.
- Une vanne remise en pression progressive.
- Des manocontact réglage 0.5-8 Bars.
- Des réducteurs de pression.
- Des manomètres.
- Des distributeurs à terroir différentiel.
- Des embases à raccord inférieure.
- Des électrovannes.
- Des connecteurs antiparasites avec LED.
- Des silencieux réducteurs d'échappement.
- Une soupape d'échappement rapide membrane.
- Des réducteurs de débit unidirectionnel.

III.5. Pupitre opérateur (PanelView 550) :

Pour que l'opérateur puisse avoir accès aux entrées/sorties et variables internes de l'automate commandant la machine, un IHM ''PanelView 550'' est mis à sa disposition. Ce pupitre permet de donner des commandes par des touches fonction et de visualiser l'état des entrées et sorties de l'automate ainsi que l'évolution du cycle machine.

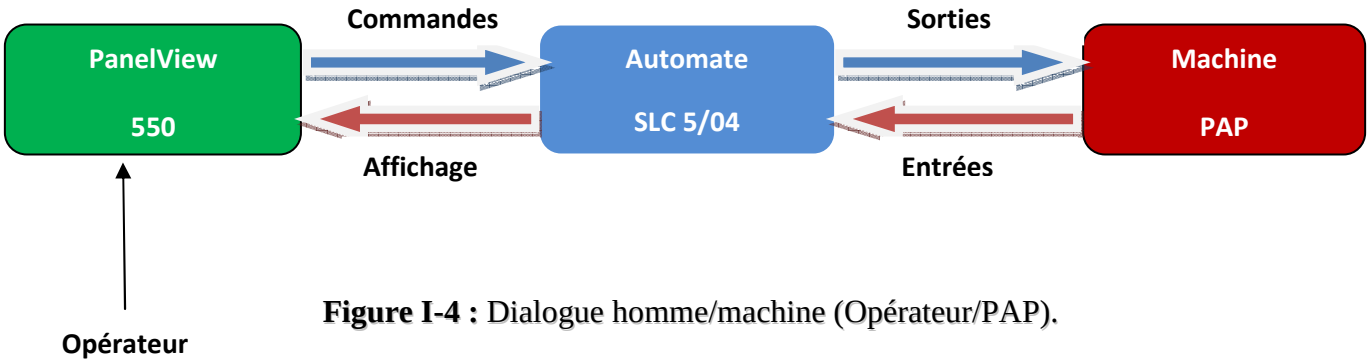


Figure I-4 : Dialogue homme/machine (Opérateur/PAP).

III.6. Les Capteurs :

La machine est équipée de 13 capteurs de 4 types différents selon le tableau suivant :

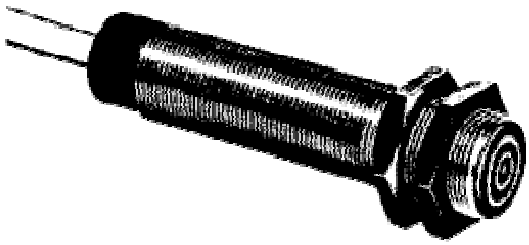
Notation Capteur	Type	Rôle	Fournisseur
2SP	Contact à pression	Détection gonflage tambour	Télémécanique
39SQ13	Contact mécanique	Détection présence tringles	Télémécanique
42SQ33	Contact mécanique	Détection tête cheval position basse	Télémécanique
34SP	Contact à pression	Détection tête cheval position haute	Télémécanique
42SQ21	Inductif	Fin de course lance table	Télémécanique
42SQ9	Cellule photoélectrique	Détection rotation tourelle	Jay
42SQ5	Inductif	Détection piano position basse	Télémécanique
42SQ4	Inductif	Détection piano position haute	Télémécanique
42SQ37	Inductif	Détection galet position basse	Télémécanique
42SQ36	Inductif	Détection galet position haute	Télémécanique
42SQ18	Cellule photoélectrique	Détection position tourelle pour rotation chapeau chinois	Télémécanique
42SQ2	Cellule photoélectrique	Détection retroussage position basse	Jay
42SQ1	Cellule photoélectrique	Détection retroussage position haute	Jay

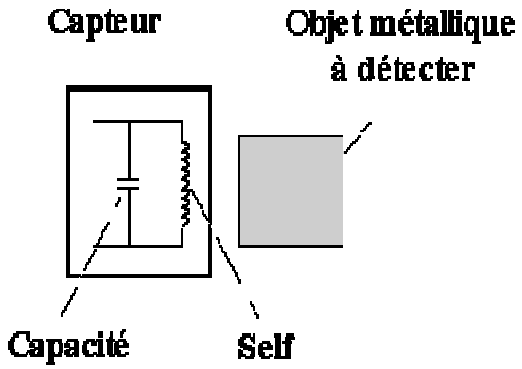
Tableau I.1 : Capteurs de la machine.

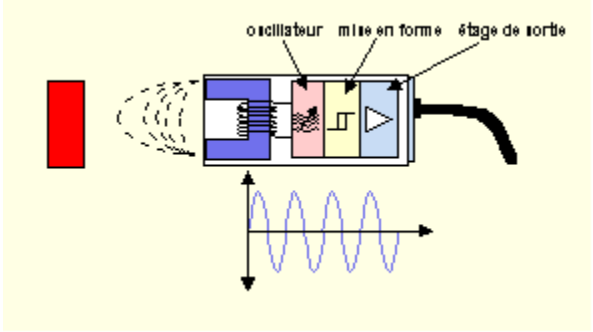
III.6.1. Notions sur les capteurs :

III.6.1.a. Capteurs inductifs :

Les capteurs inductifs produisent à l'extrémité leur tête de détection un champ magnétique oscillant. Ce champ est généré par une self et une capacité montée en parallèle. Lorsqu'un objet métallique pénètre dans ce champ, il y a perturbation de ce champ puis atténuation du champ oscillant. Cette variation est exploitée par un amplificateur qui délivre un signal de sortie, le capteur commute.





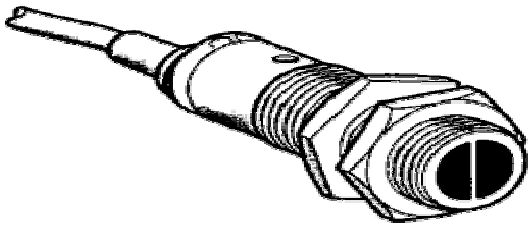


Oscillateur

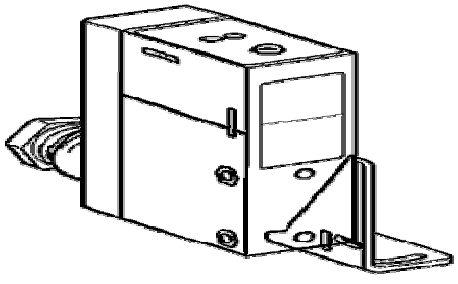
III.6.1.b. Capteurs optiques :

Un capteur photoélectrique est un capteur de proximité. Il se compose d'un émetteur de lumière associé à un récepteur. La détection d'un objet se fait par coupure ou variation d'un faisceau lumineux. Le signal est amplifié pour être exploité par la partie commande.

Exemples :

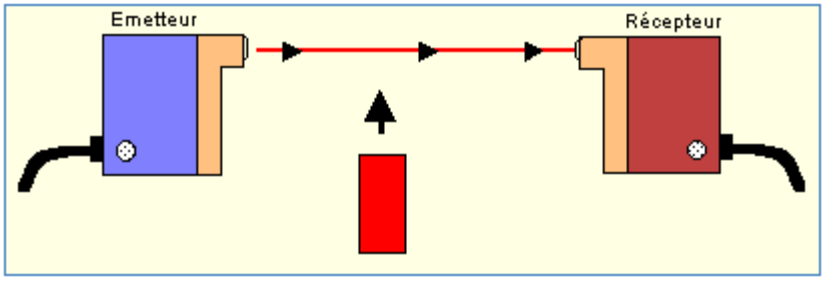


Détecteur photoélectrique cylindrique

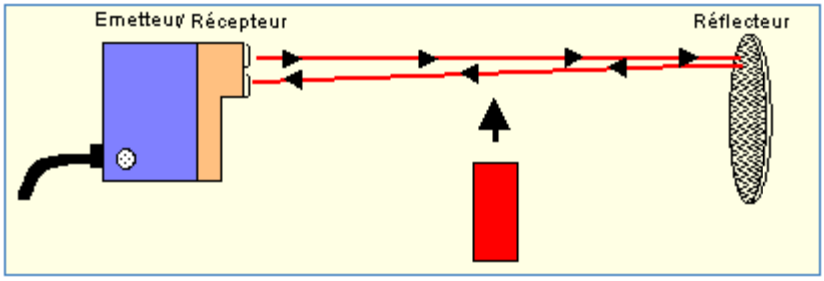


Détecteur photoélectrique avec signal de sortie analogique

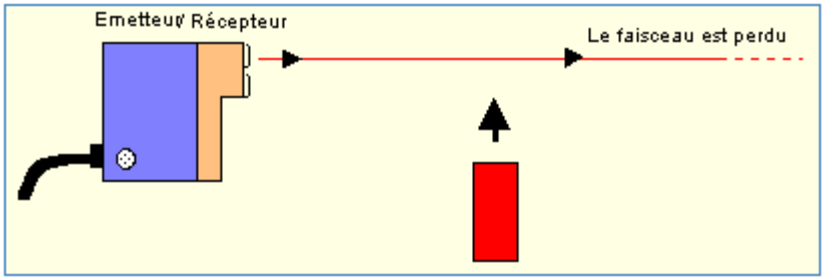
. Différents types de détection :



Système barrage



Système reflex



Système de proximité
(réflexion directe)

IV. Principe de fonctionnement de l'assembleuse PAP :

La mise en marche de la machine passe par les étapes suivantes :

a. Mise en air :

On ouvre l'arrivée générale d'air par l'intermédiaire d'un robinet situé sur l'armoire pneumatique.

b. Mise sous tension :

On met la machine sous tension par l'intermédiaire d'un interrupteur général situé sur l'armoire électrique. Un voyant « POWER ON » s'allume sur l'armoire ainsi qu'un voyant « CONTROL OFF » sur le pupitre opérateur.

c. Réarmement de la machine :

On appuie sur le bouton poussoir « REARMEMENT » du pupitre opérateur, le voyant « CONTROL OFF » doit s'éteindre. Sur l'écran de panelView de la machine s'affiche trois sélections :

. Machine setting :

Ecran de paramétrage permettant de :

- Modifier les opérations du cycle.

- Modifier les présélections des temporisations.

- Modifier les valeurs des pressions du rouletage.

Son accès se fait par l'interrupteur à clef situé sur le pupitre opérateur.

. Manual commands :

Ecran principal des commandes manuelles. Il permet de sélectionner le sous ensemble mécanique sur lequel va porter les commandes manuelles. Soit pour tester les actionneurs, soit pour exécuter des cycles élémentaires, soit pour se dégager après incident et repartir en cycle, etc…

Cet écran est accessible en permanence dés que la machine est en énergie en actionnant l'interrupteur à clé situé sur le pupitre opérateur.

. Star cycle (validation pédale) :

Permet d'aller dans le cycle prédéterminé de la pose des produits et des différentes opérations à faire par l'opérateur.

L'accès à cette fonction se fait par une impulsion sur la pédale de validation (si une tringle est présente sur son support côté bâtis).

Le cycle automatique est alors enclenché.

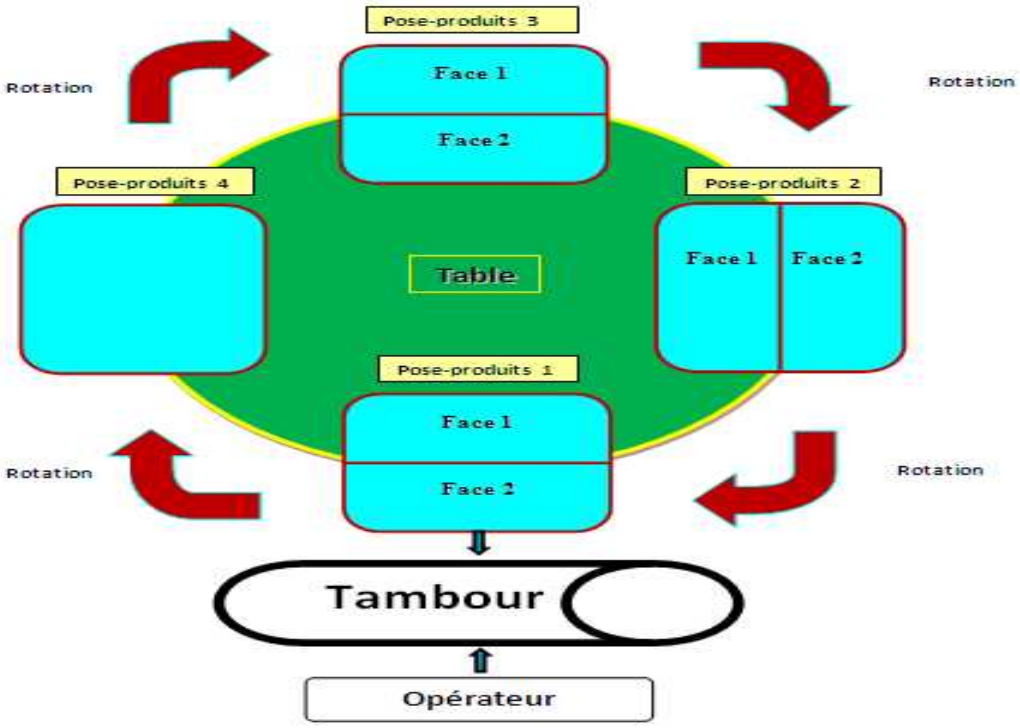


Figure I.5 : Illustration du cycle de production.

Les opérations réalisées pour chaque face :

Pose-produits 1 :

Face 1 :

1- Pose de la gomme intérieure par l'opérateur.
2- Rotation du chapeau chinois.
3- Retour du chapeau chinois à sa place.
4- Déverrouillage et rotation du pose-produits.

Face 2 :

1- Pose du RPT et le RIC par l'opérateur.
2- Déverrouillage et rotation (lancement) de la table.

Pose-produits 2 :

Face 1 :

1- Montée de la tête cheval.
2- Déroulement de la nappe carcasse.
3- Descente de la tête cheval.
4- Pose du GB.
5- Montée du piano.
6- Descente du piano après un temps de rouletage
7- Déblocage et rotation du pose-produits.

Face 2 :

1- Pose du complexe BT par l'opérateur.
2- Montée du piano.
3- Descente du piano après un temps de rouletage.
4- Pose des tringles par l'opérateur.
5- Déverrouillage et rotation (lancement) de la table.

Pose-produits 3 :

Face 1 :

1- Gonflage du tambour.
2- Montée retroussage.
3- Pose PS.
4- Descente retroussage.
5- Montée du piano.
6- Descente du piano.
7- Déblocage et rotation du pose-produits.

Face 2 :

1- Pose du complexe raidisseur par l'opérateur.
2- Montée du galet.
3- Descente du galet.
4- Déblocage et rotation de la table.

Pose-produits 4 :

1- Pose du PT par l'opérateur.
2- Pose du FE par l'opérateur.
3- Montée du galet.
4- Descente du galet.
5- Dégonflage du tambour au diamètre minimum (diamètre sortie carcasse).
6- Sortie carcasse par l'opérateur.
7- Déverrouillage et rotation (lancement) de la table.

Cette dernière étape ramène vers l'étape de départ pose-produits 1 / Face 1 pour démarrer un nouveau cycle.

Conclusion :

Dans ce chapitre, nous avons décrit la machine, son fonctionnement, ainsi que l'ensemble des éléments la constituant, là où on a dit qu'elle est gérée par un automate Allen Bradley gamme SLC 500 qui fera l'objet du chapitre suivant.

Chapitre II

Présentation de l'automate ALLEN Bradley SLC 500

Introduction :

L'automate programmable industriel API (ou Programmable Logic Controller PLC) est aujourd'hui le constituant le plus répandu des automatismes. On le trouve non seulement dans tous les secteurs de l'industrie, mais aussi dans les services (gestion de parkings, accès à des bâtiments) et dans l'agriculture (composition et délivrance de rations alimentaires dans les élevages). Il répond aux besoins d'adaptation et de flexibilité des activités économiques actuelles.

I. Structure d'un système automatisé :

I.1. Partie Opérative :

C'est elle qui opère ou agit sur la matière d'œuvre ou le produit. Elle comporte en général des actionneurs, des outillages, des éléments mécaniques permettant le processus de production.

I.2. Partie Commande :

Elle reçoit des informations de la partie opérative avec des interfaces adaptées aux informations logiques, analogiques, ou numériques et émet des ordres en retour par des cartes de sorties agissant sur les pré-actionneurs et les actionneurs afin de coordonner les actions.

Réalisée auparavant avec de la technologie câblée, elle est maintenant conçue autour de la technologie de traitement programmable.

La partie commande est le centre de traitement de trois dialogues qu'elle coordonne :

a. Dialogue avec la machine :

Prise en compte des signaux par les détecteurs et les capteurs qui informent sur l'évolution de la machine.

Commande des actionneurs (Vérins, moteurs) par l'intermédiaire des pré-actionneurs (contacteurs, distributeurs, variateurs…).

b. Dialogue homme machine :

Pour exploiter, régler, dépanner la machine, le personnel renseigne et reçoit des informations en retour.

c. Le dialogue avec d'autres machines :

Plusieurs machines peuvent se compléter pour assurer une même production. Leur coordination est réalisée par des échanges d'information entre leurs parties commandes.

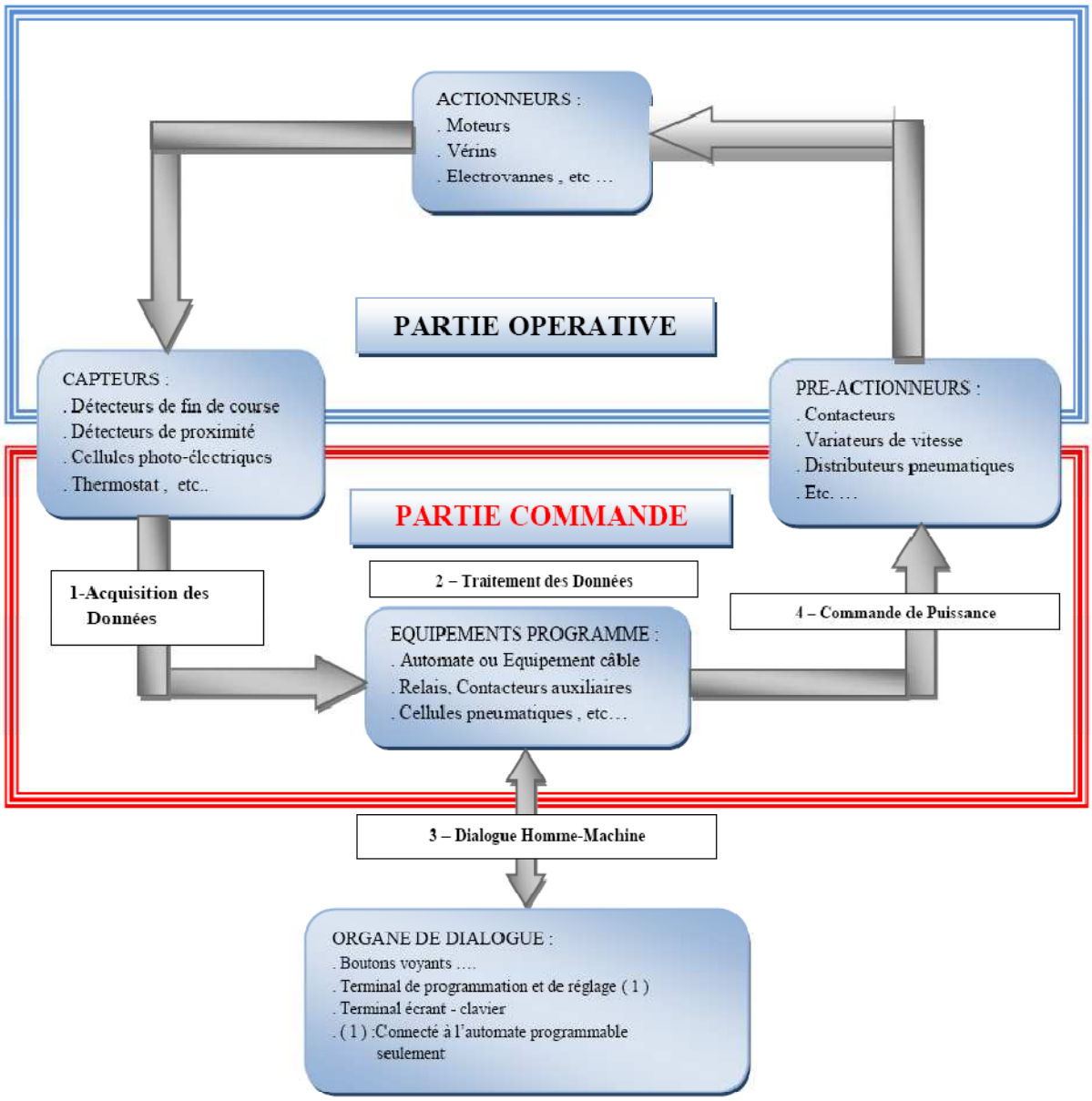


Figure II-1: Structure d'un système automatisé**.**

II. Architecture des automates programmables industriels :

La structure interne d'un API peut se représenter comme suit :

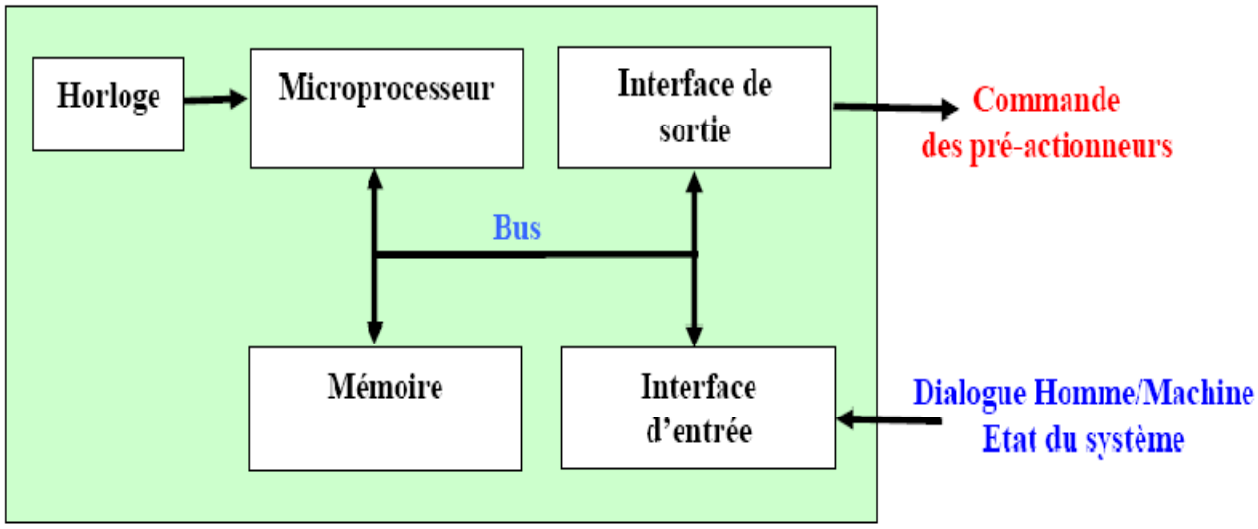


Figure II.2 : Architecture interne des API.

L'automate programmable reçoit les informations relatives à l'état du système pour ensuite commander les pré-actionneurs suivant le programme inscrit dans sa mémoire.

II.1. L'alimentation :

Elle fournie les tensions nécessaires à l'électronique de l'automate à partir des tensions usuelles (110 / 220 V alternatif ou 24 V continu)

II.2. Le Bus :

Ensemble de liaisons électriques parallèles (circuit imprimé ou câble multiconducteurs). Le nombre de fils constituant le bus dépend de l'information à véhiculer.

II.3. Le processeur :

C'est le cerveau de l'automate, il réalise toutes les fonctions logiques : ET, OU, les fonctions de temporisation, de comptage, de calcul... à partir d'un programme contenu dans sa mémoire.

II.4. La zone mémoire :

Permet de recevoir les informations issues du système et celles gérées par le processeur destinées à la commande des sorties. Les mémoires utilisées sont :

RAM (Random Acces Memory) : Mémoire vive sur laquelle on peut écrire, lire et effacer (contient le programme).

ROM (Read Only Memory) : Mémoire morte sur laquelle on ne peut que lire.

EPROM : Mémoire morte reprogrammable effaçable par rayons ultraviolets.

EEPROM : Mémoire morte reprogrammable effaçable électriquement.

II.5. Les interfaces d'entrées/sorties :

Les interfaces E/S se présentent comme des modules standards directement utilisables pour relier les capteurs (entrées) et actionneurs (sorties) au contrôleur.

II.5.1.Modules d'entrées/sorties TOR :

a. Modules d'entrées :

Un module d'entrées doit permettre à l'unité centrale de l'automate d'effectuer une ''lecture'' de l'état logique des capteurs qui lui sont associés (Module 4, 8, 16 ou 32 entrées). A chaque entrée correspond une voie qui traite le signal électrique pour élaborer une information binaire, le bit d'entrée qui est mémorisé.

b. Modules de sorties :

Un module de sorties permet à l'automate programmable d'agir sur les actionneurs. Il réalise la correspondance : état logique⟶ Signal électrique. Périodiquement le processeur adresse le module et provoque l'écriture des bits d'un mot mémoire sur les voies de sorties du module.

II.5.2. Modules d'entrées/sorties analogiques :

Le module analogique permet d'établir la correspondance entre valeurs numériques et grandeurs analogiques (Courant ou tension). La résolution (Plus petit échelon de courant ou tension) est fonction du nombre de bits utilisés pour le codage numérique. La rapidité de conversion est également une caractéristique du module.

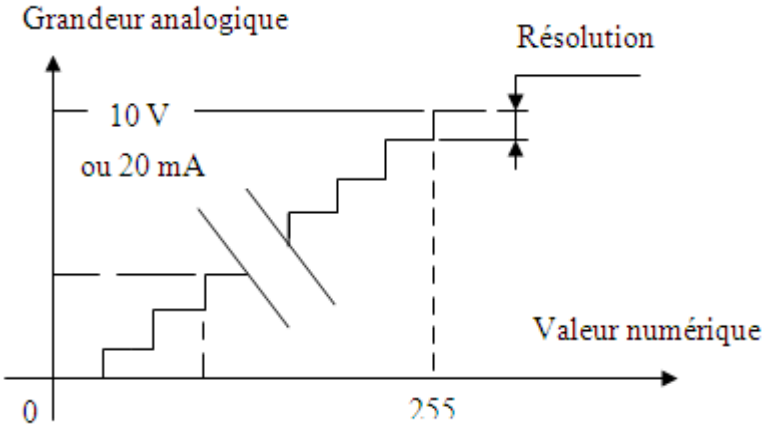


Figure II.3 : Conversion grandeur analogique en valeur numérique.

a. Modules d'entrées :

Il existe deux types de modules d'entrées analogiques :

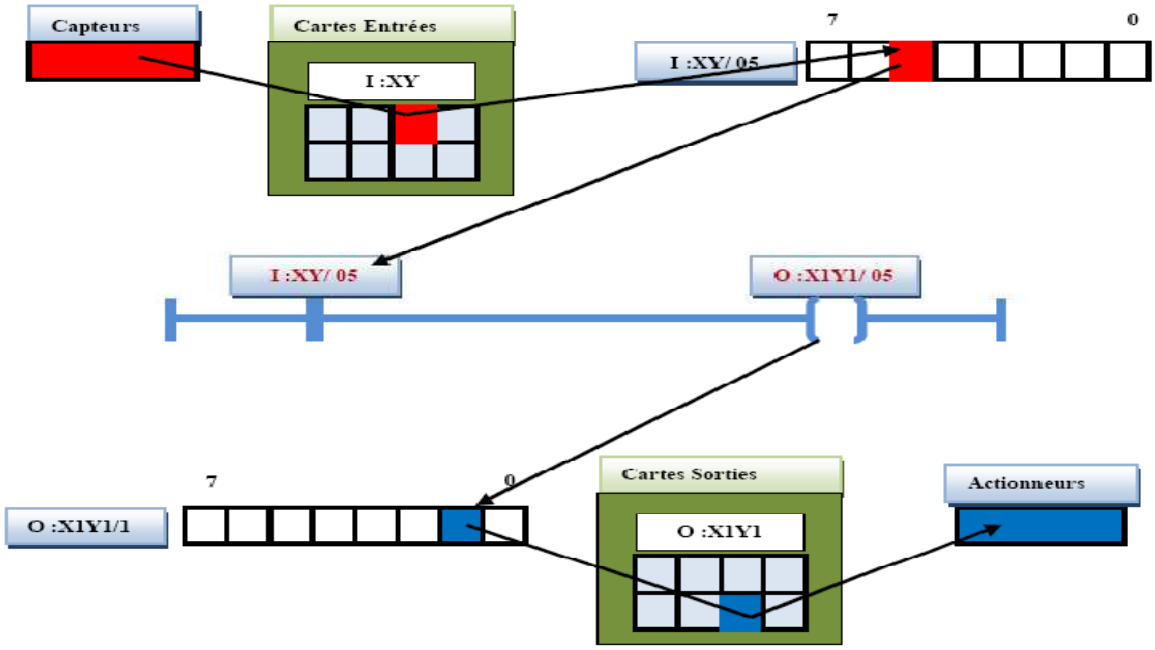
- Les entrées de détection de seuil.
- Les entrées de mesure analogique (Conversion analogique/numérique).

Un réglage d'échelle est généralement possible, permettant d'élargir les possibilités de mesure. On utilise couramment un tel module pour la mesure de température : La sonde résistive est reliée directement au module, lequel réalise ou non certaines opérations de linéarisation du signal délivré par le capteur avant écriture du mot (De n bits dans la mémoire).

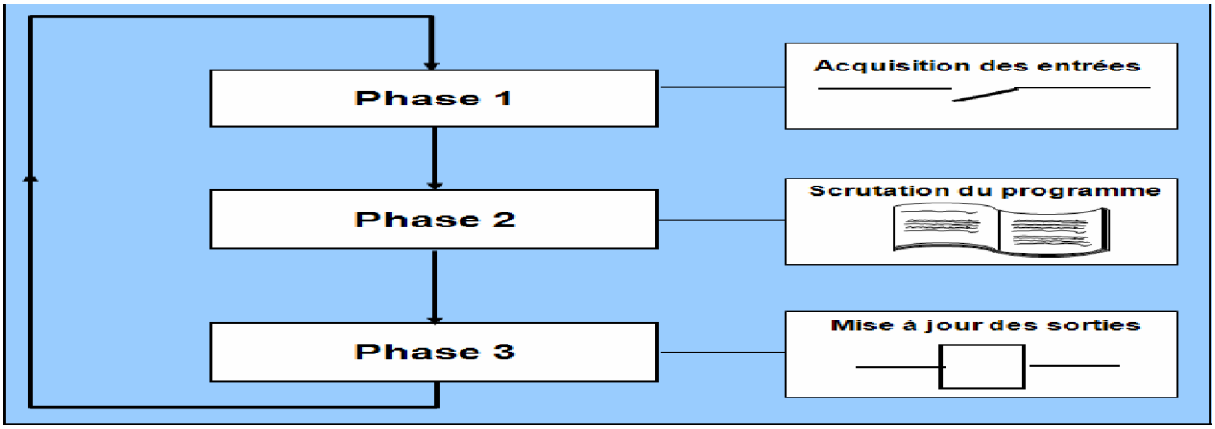
b. Modules de sorties :

Chaque sortie est l'image analogique de la valeur numérique codée sur une chaine de bits (En général de 8 à 12 bits) définie par programme. Les modules analogiques de sorties permettent, associés à des pré-actionneurs (Gradateurs de puissance, variateurs de vitesse…), de réaliser des fonctions de commande et régulation. Chaque sortie est définie par la nature du courant délivré et par ses limites (0-10 V, 4-20 mA).

III. Connexion Entrées/Sorties entre l'API et l'Automatisme piloté :



IV. Notion de temps de cycle :



Le temps de cycle (ou scrutation) est le temps qui s'écoule entre deux prises en compte d'une entrée physique (électrique).

V. Les automates programmables Allen Bradley :

L'entreprise Allen Bradley offre une gamme d'automates très large adaptée au service de la sécurité de conception et d'utilisation de machines et d'équipements dans de nombreux secteurs industriels.

On peut classer ces automates en deux types selon qu'ils soient modulaires ou compacts.

. Les automates modulaires :

Composés de différents modules (très flexibles), on distingue :

Les SLC 500.

Les PLC 5.

Les Logix 5000 : - CompactLogix.
- ControlLogix.
- FlexLogix.

Les nano-automates Pico : - Module Pico.
- Module Pico GFX.

. Les automates compacts :

Tous les composants sont regroupés en un seul bloc.

Les MicroLogix 1000 **(**SLC compact) : - MicroLogix 1100.
- MicroLogix 1200.
- MicroLogix 1500.

V.1. L'automate SLC 500 :

V.1.1. Présentation :

Le SLC 500 est une plate-forme adaptable, le plus ancien de la gamme des automates Allen Bradley reste pourtant une référence avec une offre très large de modules d'entrés/sorties classiques format **1746**, bâtie autour de deux options matérielles : une version bloc avec possibilité d'extension en utilisant un châssis à 2 emplacements, et une version modulaire comprenant jusqu'à 960 points d'E/S.

La gamme est composée de cinq modèles de processeurs : les 5/01, 5/02, 5/03, 5/04 et 5/05. Grâce à cette gamme de processeurs, le SLC 500 peut commander des machines simples

ou des procédés complexes (petits ou grands) : ces processeurs ont l'évolutivité dont ont besoin les diverses applications.

V.1.2. Caractéristiques et avantages :

Caractéristiques :

- Quatre unités centrales.
- Quatre différentes tailles de châssis (4, 7, 10 et 13).
- Modules d'E/S 1746.
- Quatre types d'alimentations.
- Options de communication.

Avantages :

- Répond à un grand nombre de besoins d'E/S et de fonctionnalités.
- Souplesse de montage des E/S et des options d'extension.
- Plus de 48 modules différents pour répondre aux besoins des applications.
- Quatre tailles différentes, supportant les alimentations c.a. et c.c.
- Par liaisons DH-485, RS-232 et DH+.

V.1.3. Options de communication :

Le tableau ci-dessous résume les options de communication des processeurs de la famille SLC 500.

Option	Communication	Type de processeur			
		SLC 5/01	SLC 5/02	SLC 5/03	SLC 5/04
DH 485	DH-485	(réception)	(réception ou émission)	(réception ou émission)	(réception ou émission)
RS-232	DH-485			•	•
	DF1 (4)	• (1)	• (1)	•	•
	ASCII			•	•
Data Highway Plus	DH+	• (2) (3)	• (2) (3)	• (2)	•

(1) Module 1747-KE nécessaire.

(2) Module 1785-KA5 nécessaire.

(3) Opération de réception uniquement par l'intermédiaire du module 1785-KA5.

(4) Esclave full ou half-duplex.

V.1.4. Processeur SLC 5/04 :

Le processeur SLC 5/04 possède les fonctionnalités du processeur SLC 5/03 auxquelles s'ajoutent les communications DH+. Ces communications sont de trois à douze fois plus rapides que les communications DH-485, offrant ainsi de meilleures performances. En outre, le processeur SLC 5/04 fonctionne environ 15 % plus vite que le processeur SLC 5/03.

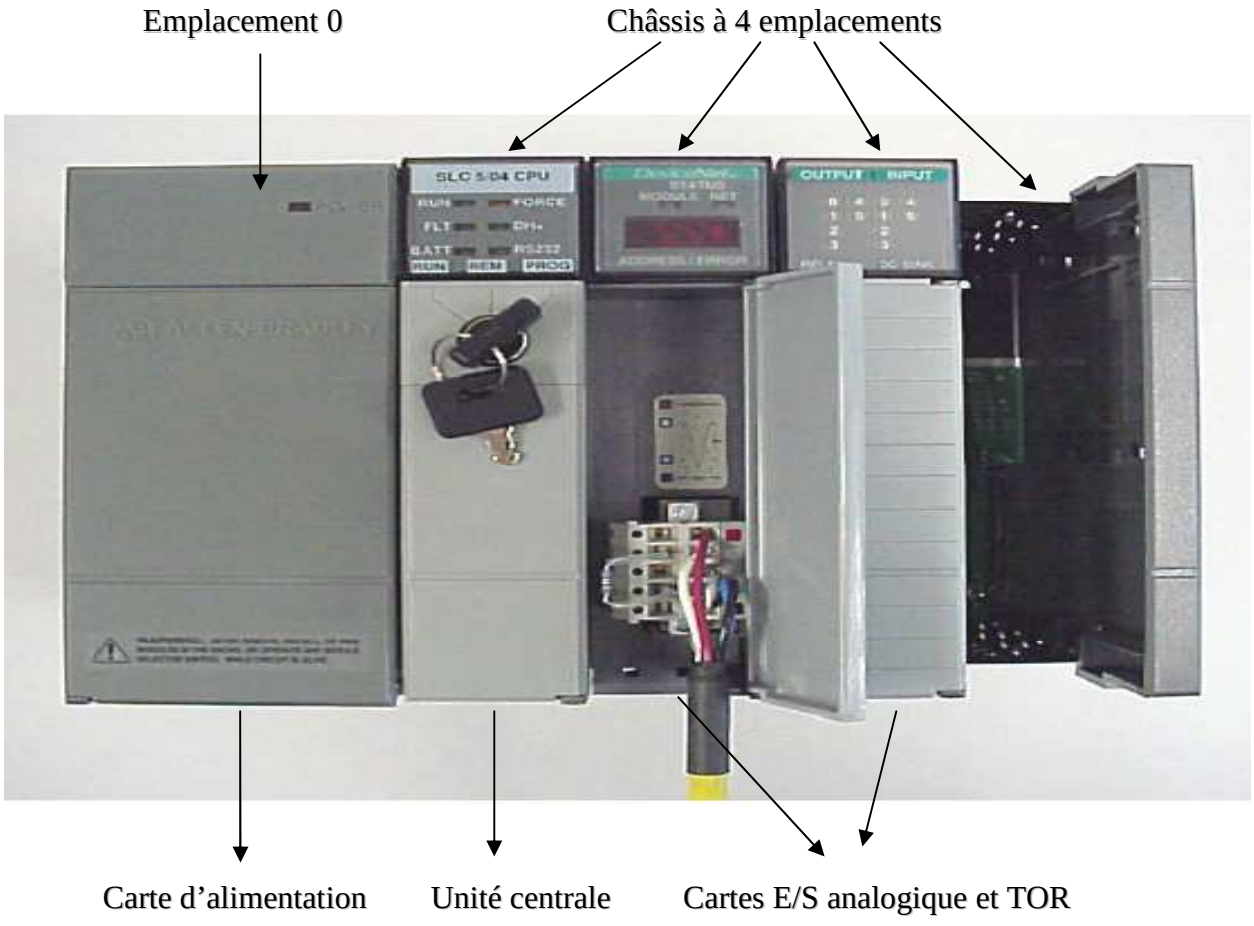


Figure II-4 : L'automate programmable Allen Bradley.

V.1.4.1. Caractéristiques du processeur SLC 5/04 :

- Tailles de mémoire programme de 16, 32 ou 64 K.
- Commande jusqu'à 4096 points d'entrée et de sortie.
- Programmation en ligne (inclut l'édition runtime).
- Voie DH+ intégrée, prenant en charge :
  - Les communications rapides (57,6K, 115,2 et 230,4 Kbauds)
  - Les capacités de messagerie avec les processeurs SLC 500, PLC-2, PLC-5 et PLC-5/250
- Voie RS-232 intégrée, prenant en charge :
  - Le DF1 Full-Duplex pour les communications point à point ; les connexions décentralisées via modem ou les connexions directes pour les dispositifs de programmation ou d'interface opérateur.
  - Le DF1 Half-Duplex maître/esclave pour les communications de type SCADA (point-à-multipoint).
  - Le DH-485 pour se connecter au réseau DH-485.
  - Les E/S ASCII pour la connexion aux autres dispositifs ASCII, tels que les lecteurs de codes à barres, les imprimantes série ou les balances.
- Capacités de "passthrough" voie à voie (DH+ vers DH-485) vers les interfaces opérateur.
- "Passthrough" voie à voie (DF1 Full-Duplex vers DH+) (OS401 et ultérieur uniquement).
- "Passthrough" RIO.
- Horloge/calendrier intégré.
- Interruption temporisée programmable de 1 ms (STI).
- Interruption d'entrée TOR de 0,50 ms (DII).
- Fonctions mathématiques évoluées - Instructions trigonométriques, PID, exponentielles, à virgule flottante et instructions de calcul.
- Adressage indirect.
- La PROM flash assure les mises à niveau du firmware sans changement physique d'EPROM.
- Module mémoire EPROM flash disponible en option.

- Commutateur - RUN, REM, PROG (Effacement des défauts).
- . RAM sauvegardée par pile.

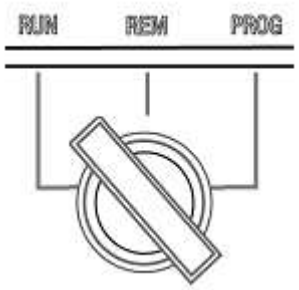
V.1.4. 2. Mode de fonctionnement du SLC 5/04 :

Utilisez le commutateur à clé situé à l'avant de l'automate pour sélectionner le mode de fonctionnement de l'automate.

a. Mode RUN :

Impossibilité de créer ou effacer des tâches, des programmes ou des sous-programmes.

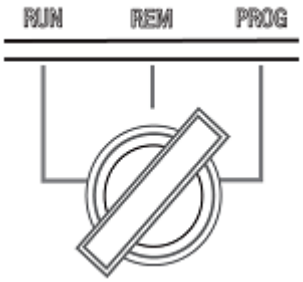
- Exécuter le programme.
- Activer les sorties.



b. Mode PROGRAM :

L'automate n'exécute pas de tâches (pas de scrutation).

- Créer, modifier et effacer des tâches, des programmes ou des sous-programmes.
- Transférer des projets.

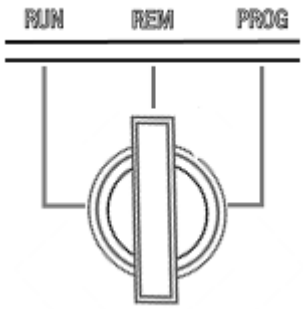


c. Mode REMOTE :

Passer aux modes programmation à distance, test à distance et fonctionnement à distance par le logiciel de programmation.

. Programmation à distance (Remote Program) :

- Désactivation des sorties.
- Création, modification et effacement de tâches.
- Transfert de projets.



. Test à distance (Remote Test) :

- Exécution de tâches avec sorties désactivées.
- Modification en ligne (limitée).

. Fonctionnement à distance (Remote Run) :

- Activation des sorties.
- Modification en ligne.

Conclusion :

Dans ce chapitre, nous avons présenté les généralités sur les automates programmables industriels, puis nous somme rentrés dans la gamme d'automates Allen Bradley, pour ensuite décrire l'automate SLC 500 (processeur 5/04) qui gère la machine pose à plat PAP sur laquelle s'effectue notre travail. Cet automate est programmé par le langage de programmation RSLogix 500 que nous allons voir au chapitre qui suit.

Chapitre III

Le logiciel de programmation RSLogix 500

Introduction :

RSLogix 500 est un logiciel de programmation de logique à relais 32 bits sous Windows pour les processeurs SLC 500 et MicroLogix de la gamme d'automate Allen Bradley. Il permet la création et la gestion de projets, la configuration et le paramétrage du matériel et de la communication, la gestion des mnémoniques, la création des programmes.

I. Les applications du logiciel RSLogix 500 :

1. **Un gestionnaire de projet :** il gère toutes les données relatives à un projet d'automatisation.
2. **Configuration matérielle :** il permet de configurer et paramétrer le matériel d'un projet d'automatisation en sélectionnant les châssis (Racks) ainsi que le nombre d'emplacements dans chaque rack, puis affecter les modules nécessaires aux emplacements souhaités dans les racks. De plus il permet le paramétrage de la CPU (comportement à la mise en route) et du protocole de communication et l'alimentation (premier emplacement automatique).
3. **Editeur de mnémoniques :** il permet de gérer toutes les variables globales. C'est-à-dire la définition de désignations symboliques et de commentaires pour les signaux du processus (entrées/sorties), mémentos, l'importation et l'exportation avec d'autres programmes Windows.
4. **Langages de programmation :** deux langages de programmation sont inclus dans le logiciel de base : LAD (Ladder Diagram), ASCII (Assembleur).
5. **Diagnostic du matériel :** il fournit un aperçu de l'état du système d'automatisation. Dans une représentation d'ensemble, un symbole permet de préciser pour chaque module s'il est défaillant ou pas.
6. **RSLinx :** il permet le transfert de données via un réseau de communication DH+, DH485, Ethernet, tout en offrant les possibilités de choisir les participants à la communication.

II. Les fonctions du logiciel RSLogix 500 :

1. Un éditeur de logique à relais à structure libre qui permet de se concentrer sur la logique de l'application plutôt que sur la syntaxe pendant l'écriture du programme.
2. Un vérificateur de projet puissant qui sert à créer une liste d'erreurs.
3. Une fonction d'édition "glisser-déplacer" pour déplacer rapidement des éléments de table de données d'un fichier de données à un autre, des lignes d'un sous-programme ou d'un projet à un autre ou des instructions d'une ligne à une autre dans un même projet.

4. Des bibliothèques SLC servant à stocker et à récupérer des portions de la logique à relais pour les réutiliser dans l'un des logiciels de programmation de SLC.

5. Un utilitaire de comparaison permettant de visualiser les différences entre deux projets.

III. Les langages de programmation :

Cinq langages de programmation peuvent être utilisés pour la programmation des automates programmables industriels (API). Ces langages peuvent être divisés en deux grandes catégories :

III.1. Les langages graphiques :

III.1.1. Le GRAFCET :

L'acronyme GRAFCET signifie : **GRA**phe **F**onctionnel de **C**ommande **E**tape **T**ransition. C'est une méthode de représentation graphique permettant de décrire le cahier de charge d'un automatisme. Il est adapté aux systèmes à évolution séquentielle, défini par un ensemble d'éléments graphiques de base traduisant le comportement de la partie commande vis-à-vis de ses entrées et de ses sorties.

Un programme GRAFCET décrit un procédé comme une suite d'étapes, reliées entre elles par des transitions. A chaque transition est associée une réceptivité, celle-ci est une condition logique qui doit être vraie pour franchir la transition et passer à l'étape suivante. Des actions sont associées aux étapes du programme.

Le format graphique d'un programme GRAFCET est le suivant :

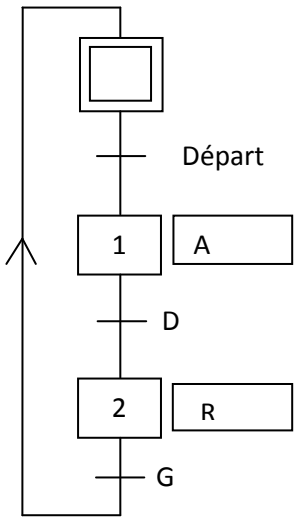


Figure III-1 : Exemple d'un GRAFCET.

Une ETAPE correspond à une phase durant laquelle on effectue une ACTION pendant une certaine DUREE. L'action doit être stable, c'est à dire que l'on fait la même chose pendant toute la durée de l'étape, en particulier composition de plusieurs actions, ou à l'opposé l'inaction (étape dite d'attente).

On représente chaque étape par un carré, l'action est représentée dans un rectangle à droite, l'entrée se fait par le haut et la sortie par le bas. On numérote chaque étape par un entier positif, mais pas nécessairement croissant par pas de 1.

Une TRANSITION est une condition de passage d'une étape à une autre. Elle n'est que logique, sans notion de durée. La condition est définie par une RECEPTIVITE qui est généralement une expression booléenne de l'état des CAPTEURS.

On représente une transition par un trait horizontal sur une liaison verticale. On note à droite la réceptivité double barre. Dans le cas de plusieurs liaisons arrivant sur une transition, on les fait converger sur une grande.

III.1.2. Le Ladder (LADDER Diagram) :

Le LD est une représentation graphique qui traduit directement des équations booléennes en un circuit électrique en combinant des contacts et des relais à l'aide des connexions horizontales et verticales, les contacts représentent les entrées (contacts normalement ouverts, contacts normalement fermés,…), et les relais représentent les sorties (relais directs, relais inversés,…). les diagrammes LD sont limités sur la gauche par une barre d'alimentation et par la masse sur la droite.

Exemple : Programmation d'une fonction logique : $x = (a + b) . (e.\overline{c} + \overline{d})$.

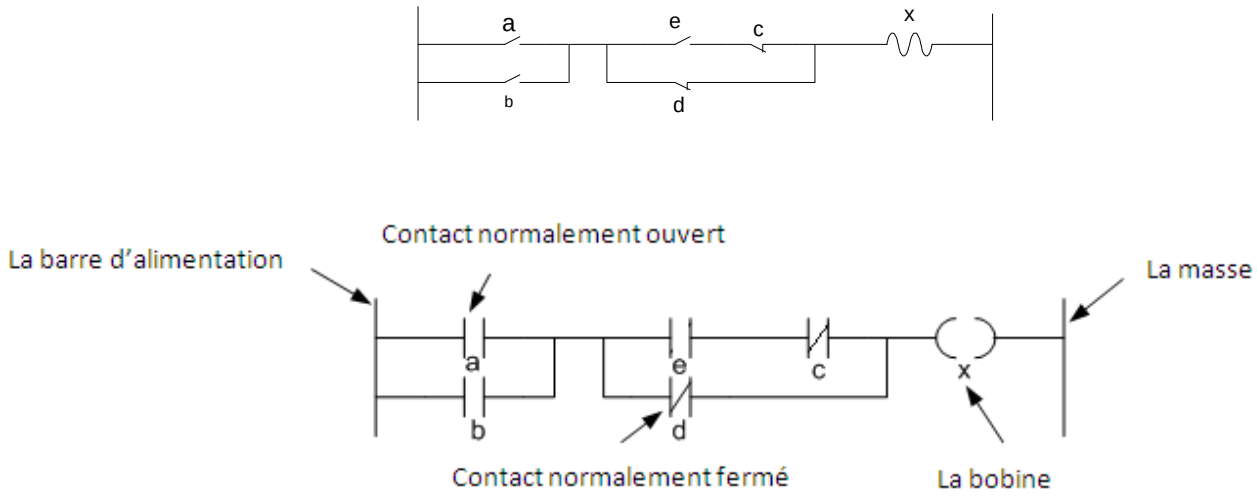


Figure III.2 : Programme LADDER.

Le langage Ladder propose d'autres fonctions telles que les fonctions de comptage et de temporisation, les fonctions arithmétiques, logiques et de conversion, les fonctions de comparaison et de transfert.

III.1.3. Le Bloc de fonction (FBD) :

C'est un langage graphique qui permet la construction d'équations complexes à partir des opérateurs standard, ou de blocs fonctionnels. Il se compose de réseaux de fonctions préprogrammées ou non, représentées par des rectangles qui sont connectés entre eux par des lignes. La programmation avec le FBD est très souple et facile à apprendre, la plupart des fonctions nécessaires (les fonctions arithmétiques et logiques, les fonctions de temporisation, les blocs fonctionnels PID…) sont déjà disponibles dans la bibliothèque, il suffit juste de les connecter et bien paramétrer les entrées et les sorties, c'est-à-dire respecter le type de variables lors de la connexion.

Par exemple pour réaliser l'opération arithmétique 20.(x+y)/z = w, on aura besoin de trois blocs : Un pour l'addition, un pour la multiplication et un autre pour la division.

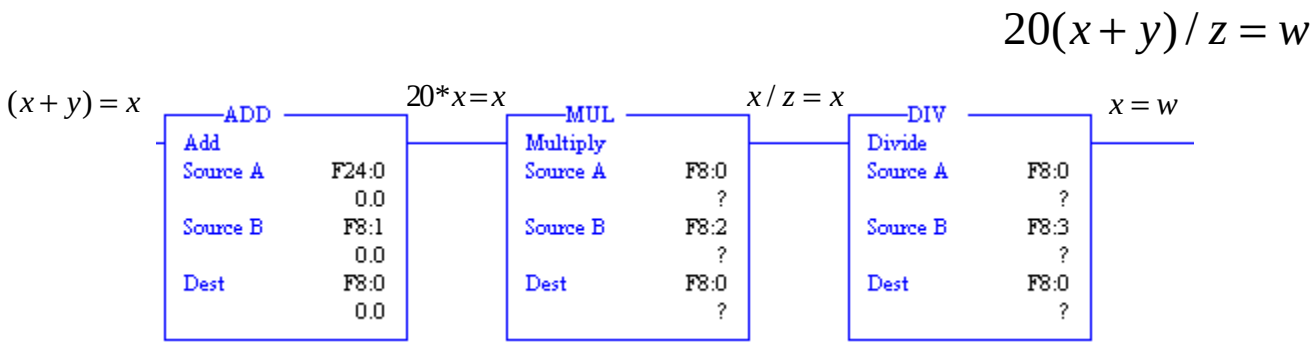


Figure III.3 : ligne d'une opération arithmétique.

III.2. Les langages textuels :

III.2.1. Le texte structuré (ST) :

Le langage ST (Structured Text) est un langage de programmation textuel de haut niveau dédié aux applications d'automatisation, il est utilisé principalement pour décrire les procédures complexes et difficilement modélisables avec les langages graphiques, il peut aussi être utilisé en tant que sous programme avec d'autres langages de programmation. Il utilise les mêmes énoncés que les langages de programmation évolués (Pascal, C, C++…), comme : les assignations, les

appels de fonctions, les énoncés de contrôle (IF, THEN, ELSE, CASE) ou d'itération (FOR, WHILE, REPEAT) en plus des opérations arithmétiques et logiques.

III.2.2. La liste d'instructions (IL) :

Le langage IL est un langage textuel de bas niveau (proche du langage assembleur), qui utilise un jeu d'instructions simple, il trouve sa puissance dans les applications de petites tailles, et dans la création de sous programmes ou procédures, car il permet un contrôle totale et une optimisation parfaite du code, par contre dans les grandes applications il est très difficile de programmer avec le IL. Les programmes dans ce langage peuvent être traduits des autres langages. Le IL a la même structure que l'assembleur, il utilise un ou plusieurs registres de travail. Les valeurs intermédiaires nécessaires pour l'exécution d'une instruction donnée seront mémorisées dans ces registres le temps de leur utilisation. Il possède un jeu d'instructions assez riche qui peut réaliser toutes les opérations arithmétiques et logiques, les opérations de comptage et temporisations, la comparaison et le transfert…

IV. Création d'un projet RSLogix 500 :

RSLogix 500 est basé sur l'utilisation de projets. Un projet est un ensemble complet de fichiers associés à un programme de logique. Il comprend essentiellement deux données : les fichiers programmes et la configuration matérielle.
On crée un projet à partir du menu Fichier, RSLogix 500 invite à définir le type de processeur avec lequel il sera communiqué et créer un contrôle d'arborescence du projet. Cette arborescence du projet est un point d'accès au programme, à la table de données et aux fichiers de base de données.

Les types de processeurs qui peuvent être programmés par RSLogix500 sont les SLC 500 et les MicroLogix de la gamme automate Allen Bradley.

Exemple : choix d'un automate SLC 500, CPU SLC 5/04 et une mémoire de 64 K mots.

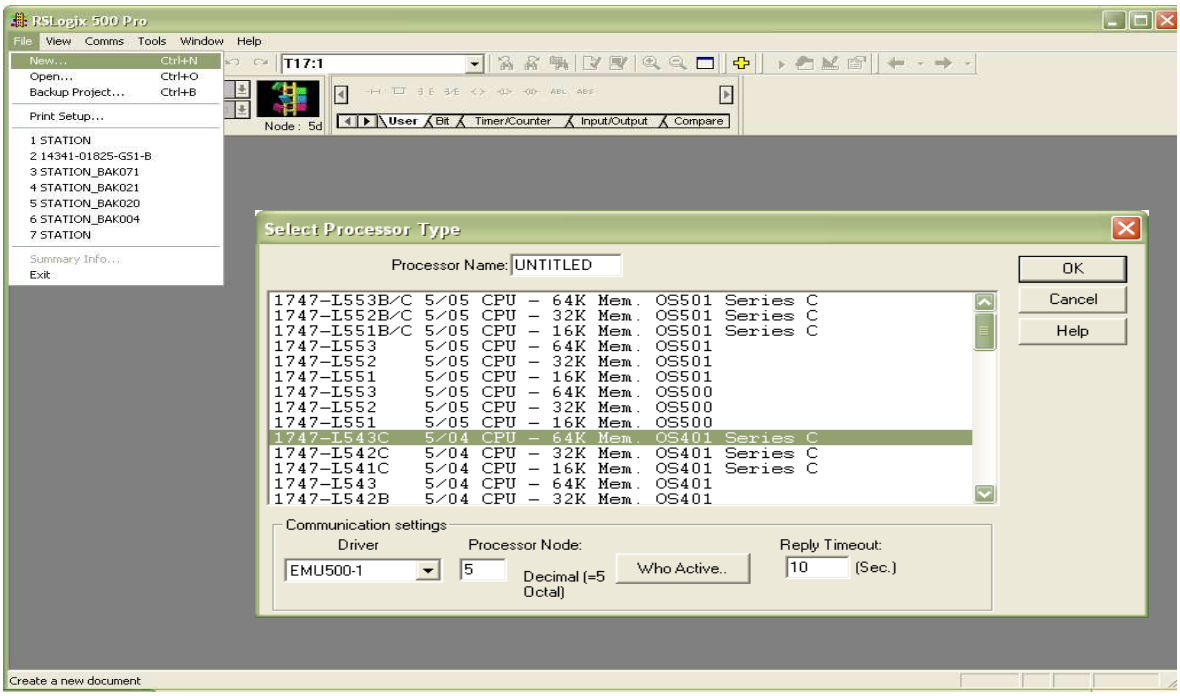


Figure III.4 : Choix de la CPU de travail.

Comment un processeur d'un automate Allen Bradley est référencié :

. CPU : L'Unité Centrale, noté : **1747-Lxxx 5/0x CPU - xxK Mem. OSxxx** :

- 1747-Lxxx : La référence des automates SLC.
- 5/0x CPU : Le type d'automate.
- xxK Mem : La taille mémoire de l'automate donnée en Kilo d'instructions.
- OSxxx : La série de la CPU d'automate.

IV.1. Configuration des communications système :

IV.1. 1. Communications système et communications de l'automate :

Configurer toutes les communications avant de commencer un nouveau projet.

Deux méthodes permettant de définir les paramètres de communication :

➢ Utilisation de la boîte de dialogue ''Communications du système'' (accessible à partir du menu Communications) pour indiquer la configuration de communication du processeur auquel on veut se connecter. Cette méthode de communication n'est pas reliée au projet.

➢ Utiliser la boîte de dialogue ''Communications de l'automate'' (accessible à partir du

menu Propriétés de l'automate dans l'arborescence du projet), dans le cas où on veut que les paramètres du driver et de la station entrés restent dans le projet. D'autre terme, si on veut que les

informations du driver et de la station qu'on a définies pour notre projet écrasent les paramètres de communication du système au chargement du projet par la suite sur un processeur spécifique.

Sélectionner ensuite un driver de Communication. Si la liste de drivers est vide, on lance RSLinx pour configurer un driver.

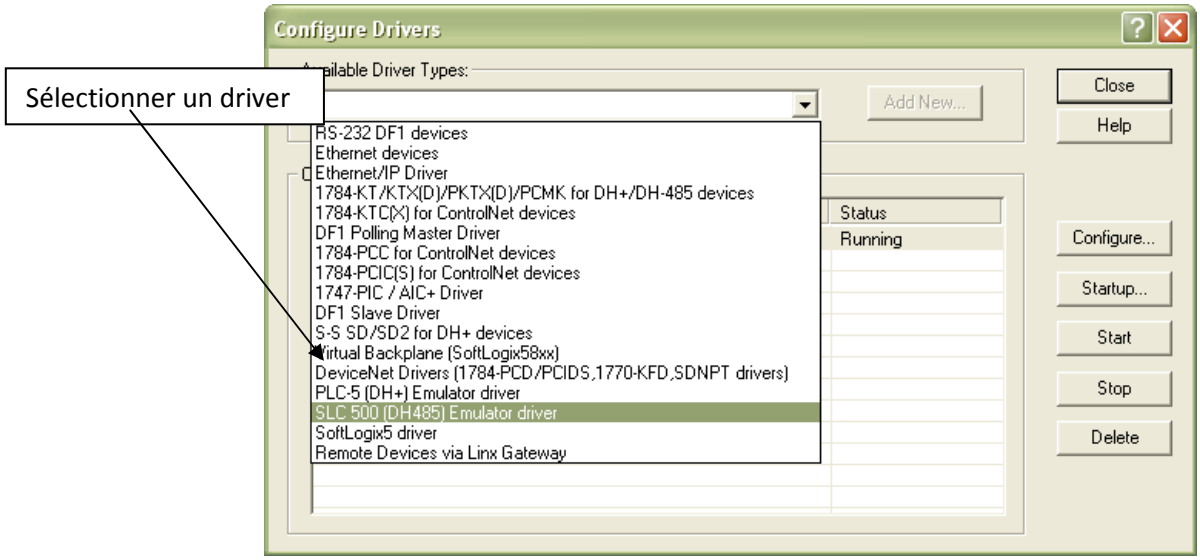


Figure III.5 : Configuration d'un driver.

Dans le menu des drivers on trouve les drivers des protocoles de communication des automates Allen Bradley :

Le port DH-485 existe en standard sur les SLC 5/01, 5/02 et 5/03.

Le port DH + existe en standard sur les SLC 5/04.

Le port Ethernet existe en standard sur les SLC 5/05.

IV.1.2. Les différents protocoles de communication :

EtherNet/IP : Communique avec les PLC-5E, SLC-5/05, et Automates Logix 5000. Le réseau EtherNet est un réseau local avec un débit de transmission en bande de base de 10 Mbits/s.

DeviceNet : Réseau de communication ouvert aux normes industrielles. Ce réseau a été conçu pour fournir une interface à des différents dispositifs, tels que les capteurs, les boutons-poussoirs, les

démarreurs de moteurs, les interfaces opérateur simples et les variateurs, via un seul câble partant du processeur d'un contrôleur programmable.

ControlNet : Réseau de commande ouvert utilisant le modèle Producteur/ Consommateur pour combiner les fonctionnalités d'un réseau d'I/O et d'un réseau d'égal à égal tout en garantissant de hautes performances aux deux types de fonctions. Le terminal ControlNet communique avec multiples contrôleurs sur un réseau ControlNet.

Data Highway Plus : Le réseau DH+ est une liaison en bande de base à passage de jeton d'Allen-Bradley conçue pour un réseau local. Le terminal DH+ communique avec un contrôleur PLC seul, SLC 5/04, système ControlLogix, ou multiples automates Allen-Bradley sur le réseau DH+. Ces dispositifs sont appelés stations. Il donne accès direct aux fichiers de données du contrôleur. Il aide à fournir un temps-efficace fiable pou le transport de données. On Peut connecter au maximum 32 stations à une seule liaison DH+.

Remote I/O (RIO) : Communique avec automates PLC ou SLC 500, ou système ControlLogix sur le module 1771 Remote I/O network, Il supporte les deux types de transfert discret et bloc de données.

DH-485 : Le DH-485 est un protocole de communication qui passe les informations entre différentes étapes de l'installation. Le réseau permet de contrôler le processus et les différents paramètres, différents états, et programmes d'application, surveillance de données, chargement du programme, et maîtrise de la surveillance. Le terminal DH-485 permet la communication avec automate Allen-Bradley SLC 500 seul ou multiple ou MicroLogix sur le réseau DH-485. Il supporte le point à point ou les transferts du réseau.

RS-232 (DH-485 protocole) : Communique avec automates MicroLogix et SLC qui utilisent le protocole DH-485 point à point.

RS-232 (DF1 protocole) : Communique avec automate SLC 500 seul, PLC ou MicroLogix sur point-à-point liaison DF1.

Profibus : Une communication utilise le Profibus DP standard pour des grandes-vitesses (jusqu'à 1.5M baud) de transmission des données à Siemens et autres contrôleurs qui supportent le protocole Profibus DP.

Modbus terminal : Utilise un transfert half-duplex, et est un protocole de communication maître/esclave d'architecture contrôleur Rockwell Automation et d'autres stations d'automatisation.

IV.1.3. Qui actif :

La fonction Qui actif est utilisée pour indiquer les stations connectées au réseau SLC. Ces informations permettent de sélectionner les stations à partir desquelles transférer, vers lesquelles charger ou lesquelles surveiller en ligne. Elle peut également afficher les statistiques sur le rendement de ces communications.

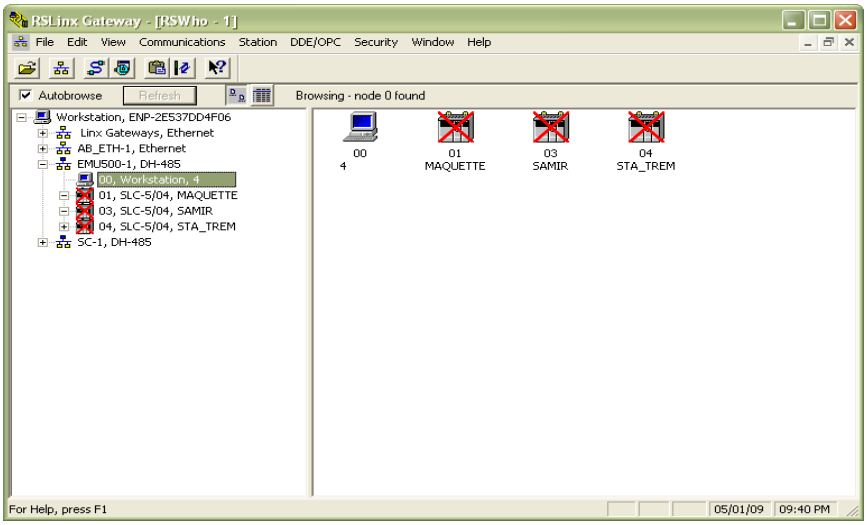


Figure III.6 : Fonction Qui Actif.

IV.2. Configuration matérielle :

IV.2.1. Installation du châssis et des modules d'E/S :

Après avoir ouvert un projet, il faut définir un châssis, identifier les cartes d'E/S en indiquant leur position dans le rack du processeur et sélectionner le type d'alimentation nécessaire pour chacun des racks de la configuration. Une application réelle peut comprendre jusqu'à trois racks et plusieurs modules d'E/S.

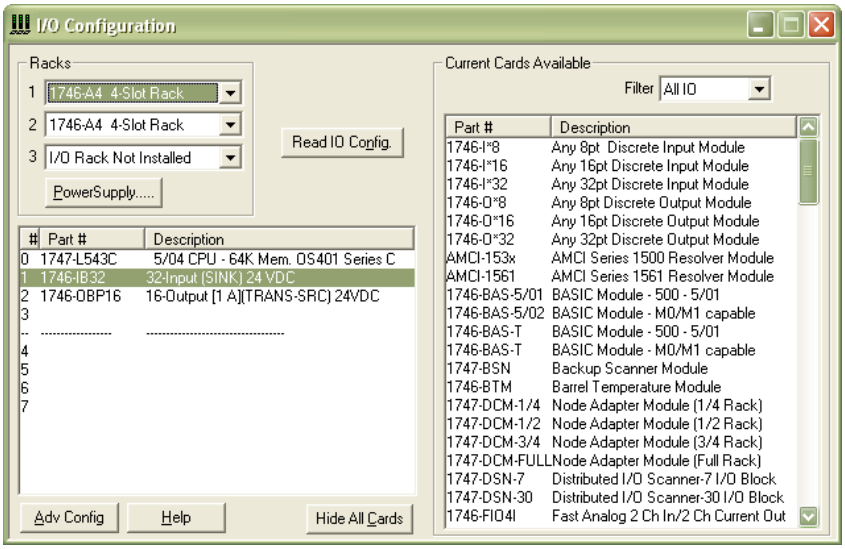


Figure III.7 : Configuration matérielle.

La boîte de dialogue Configuration des E/S permet également d'effectuer d'autres tâches :

- De savoir si la source d'alimentation que nous avons l'intention d'utiliser fournira assez de courant aux modules placés dans le rack.
- De configurer les modules analogiques et les autres modules spécialisés.
- Lecture automatique de la configuration existante des E/S d'une station processeur sur le réseau.

IV.2.2 Consommation électrique :

La boîte de dialogue Consommation électrique est en lecture seule. On ne peut définir aucun paramètre. Elle est utilisée pour examiner la consommation d'un rack selon la configuration des modules qu'on a sélectionnés.

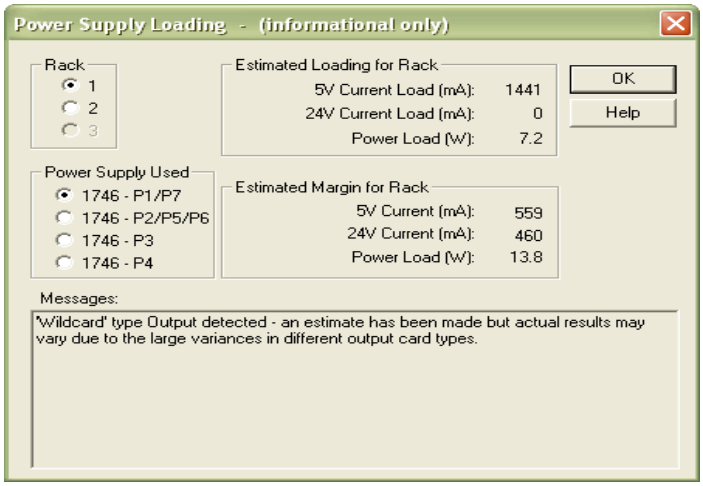


Figure III.8 : Consommation électrique.

V. Utilisation des instructions de logique à relais :

V.1 Palette d'instructions flottante :

RSLogix 500 permet d'afficher une palette d'instructions flottante qui facilite la sélection des instructions, elle est accessible soit à partir du menu affichage, soit en cliquant sur son icône.

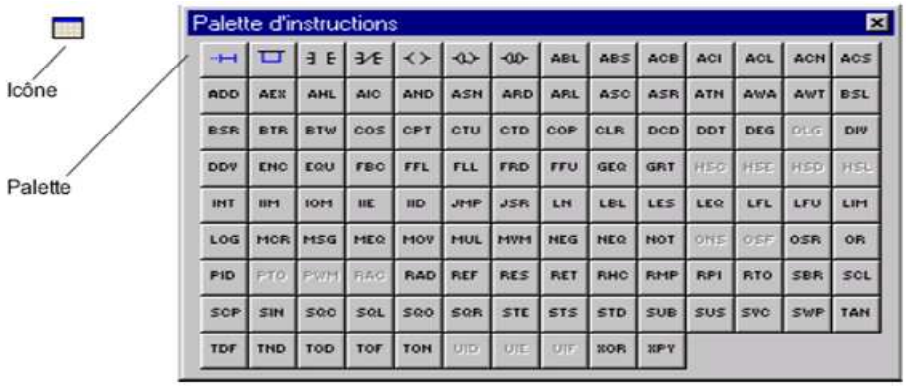


Figure III.9 : palette d'instructions flottante.

Pour choisir l'instruction désirée, soit on clique directement sur son icône à partir de la palette flottante, soit en cliquant sur le bouton droit de la souris au début de la ligne sur laquelle on veut l'insérer, une fenêtre d'instructions s'ouvre après avoir sélectionné « Append instruction », et on choisit l'instruction désirée.

V.2. Barre d'instructions :

Pour choisir les instructions selon leur type (exemple : User, Bit, Timer/Counter…), on utilise la barre d'instructions située au dessus de la zone d'édition du programme.

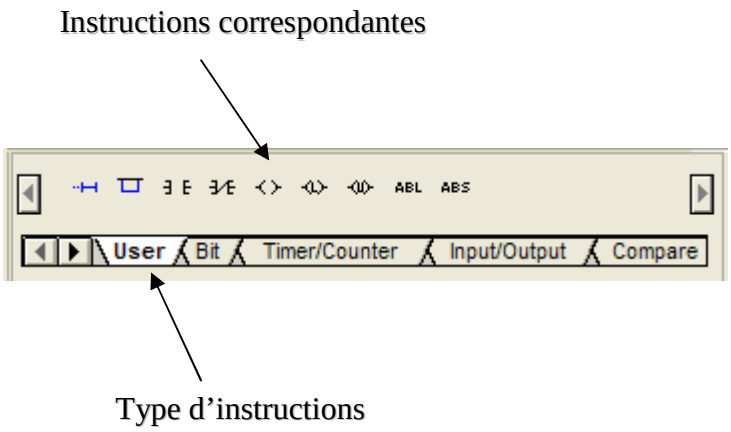


Figure III.10 : Barre d'instructions.

V.3. Entrée rapide des instructions :

Pour accélérer les tâches de programmation, RSLogix 500 permet d'affecter à chaque touche alphabétique (A à Z) du clavier une instruction de programmation de logique à relais. Pour y accéder à la fenêtre de sélection rapide, on va sur le menu Afficher, puis on clique sur Propriétés.

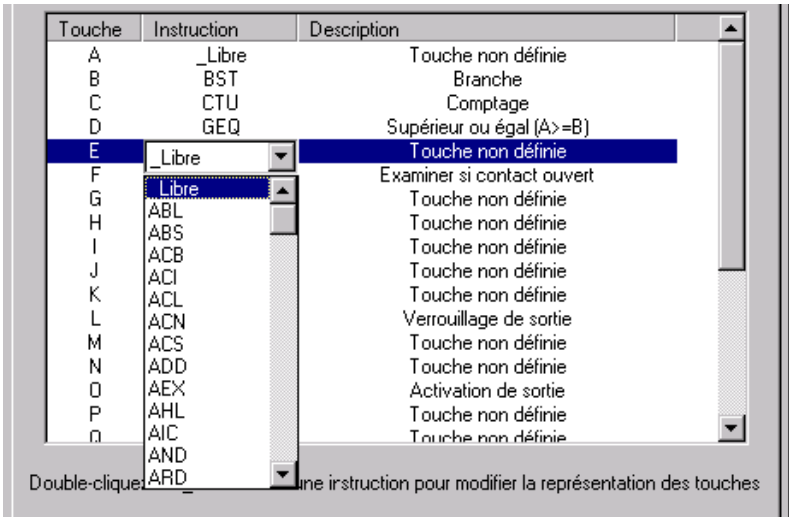


Figure III.11 : Entrée rapide des instructions.

La fenêtre suivante montre la zone d'édition de programmes LADDER :

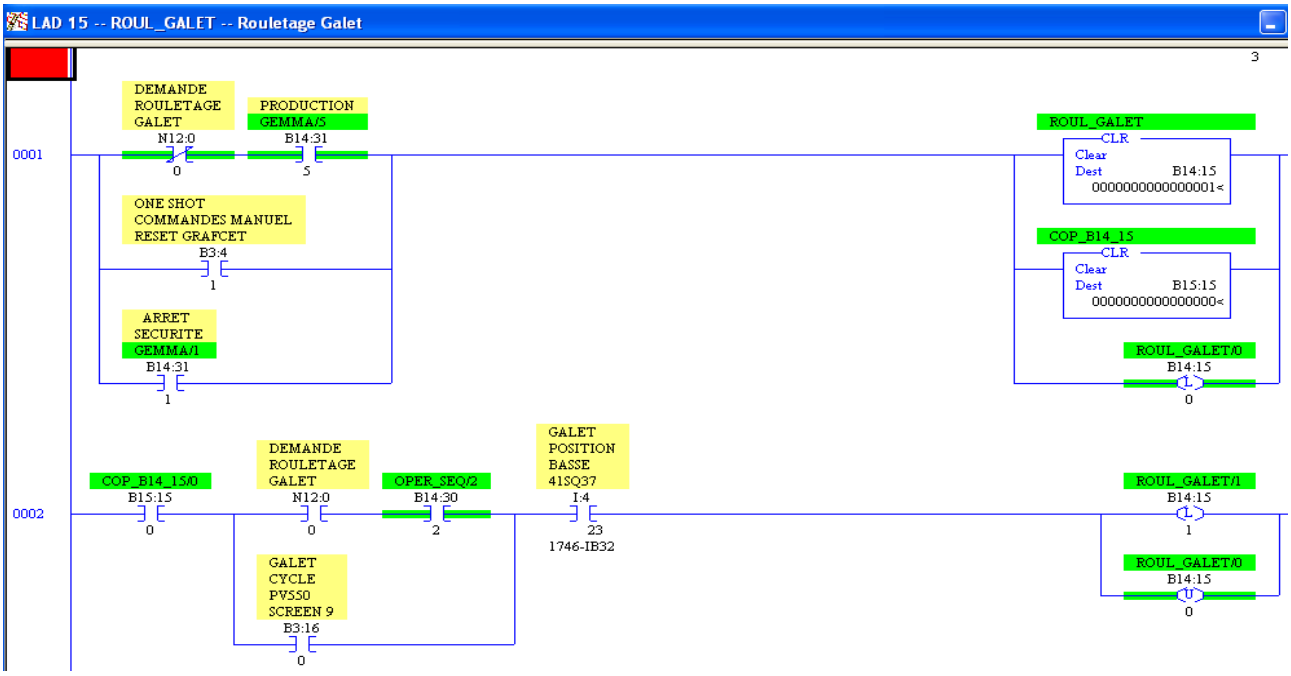


Figure III.12 : La zone d'édition d'un programme LADDER.

VI. Introduction à la programmation :

VI.1. Organisation de la mémoire programme :

Il peut y avoir 256 fichiers programme (Fichiers LADDER numérotés de 0 à 255) pour les SLC : Un programme principal et des sous-programmes.

Fichier 0 : Fichier réservé au système.

Fichier 1 : Fichier non utilisé.

Fichier 2 à 255 :

- Fichier 2 : Programme principal.
- Fichiers de 3 à 255 : Sous-programmes :
    - Fonctionnement.
    - Défauts.
    - Temps (STI)
    - Entrées interruptives (DII)…

VI.2. Les types de variables :

Les données programme sont organisées en fichiers programme selon leur type, on distingue :

- ASCII : Caractère ASCII.
- BIT : Elément binaire.
- COUNTER : Compteur.
- FLOATING : Nombre flottant.
- INPUT : Entrée.
- INTEGER : Nombre entier.
- OUTPUT : Sortie.
- PID : Registre de contrôle PID.
- REGISTER : Registre de contrôle.
- STATUS : Etat.
- STRING : Chaine de caractères ASCII.
- TIMER : Temporisation.

VI.3. L'adressage des entrées/sorties :

L'adressage des entrées et des sorties se fait de la même manière : **I:e.s/b** et **O:e.s/b.**

- **I** = entrée.
- **O** = sortie.
- **:** = Délimiteur d'élément.
- **e** = Numéro d'emplacement (décimal).
- **.** = Délimiteur de mot.
- **s** = Numéro de mot.
- **/** = Délimiteur de bit.
- **b** = Numéro de bit (de borne) en décimal.

- Principe de base :

- L'emplacement 0 est réservé à l'UC.
- Les emplacements sont numérotés en suivant, de châssis en châssis de 1 à 30 pour les SLC modulaires répartis sur 3 châssis au maximum.

- Adressage d'un automate SLC Modulaire :

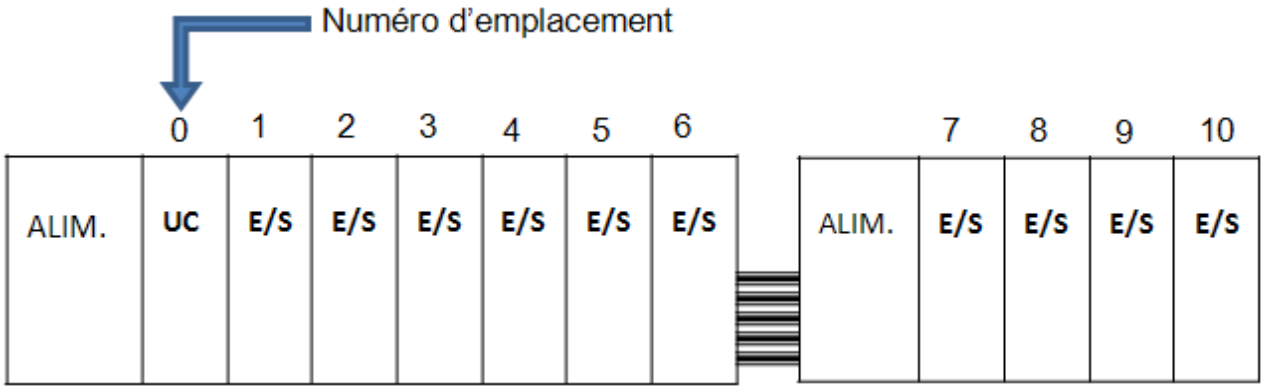


Figure III-13 : Adressage d'un automate SLC modulaire.

VI.4. La table de données :

On peut avoir jusqu'à 256 sections numérotées de 0 à 255.

Les sections de 0 à 8 contiendront toujours :

0 = Output ; 1 = Input ; 2 = Status ; 3 = Binary ; 4 = Timer ; 5 = Counter ;

6 = Register ; 7 = Integer ; 8 = Floating.

Il est conseillé de réserver la section n^0 9 pour le transfert DH-485 vers autre automate que SLC 500.

Numéro de la Section	Nombre Maxi d'éléments	Type d'élément	Adresses
0	30	OUTPUT Table images des sorties	O:01 O:30
1	30	INPUT Table images des entrées	I:01 I:30
2	Selon UC	STATUS Fichier état du processeur	S:0 S:xx
3	256	BIT Bits internes	B3:0 B3:255
4	256	TIMER Temporisations	T4:0 T4:255
5	256	COUNTER Compteurs	C5:0 C5:255
6	256	CONTROL REGISTER Registres de contrôle	R6:0 R6:255
7	256	INTEGER Nombres entiers	N7:0 N7:255
8	256	FLOATING Nombres flottants	F8:0 F8:255
9	256	Utilisé pour le DH-485	
N	256	ASCII Caractères ASCII	A25:0 A25:255
N	256	STRING Chaîne de caractères	ST32:0 ST32:255

Tableau III.1 : Les types de sections.

VI.5. Adressage direct de la table de données :

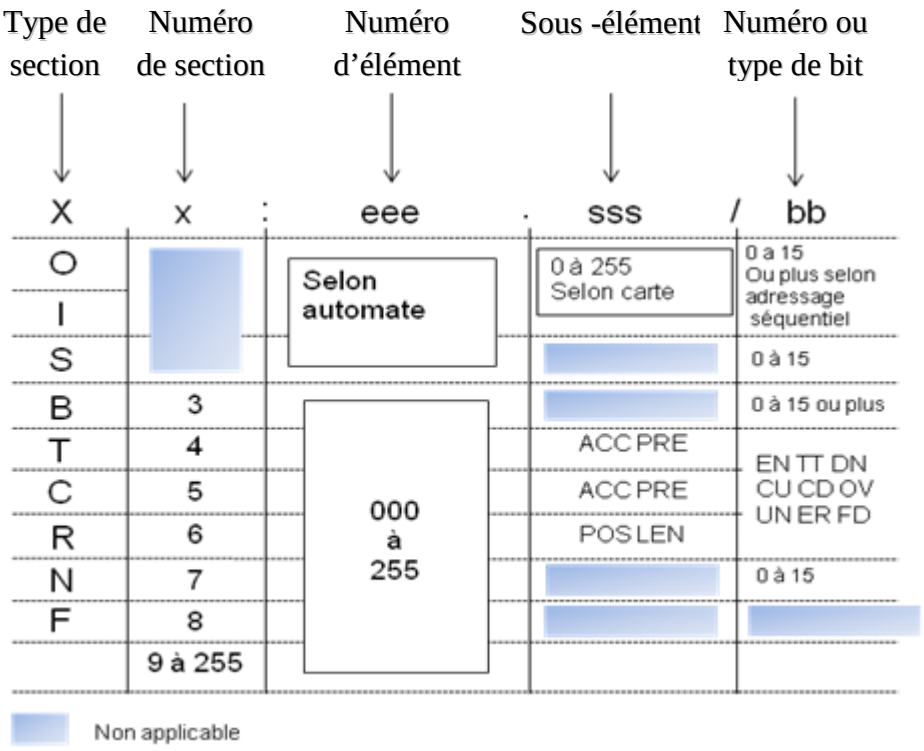


Figure III.14 : Adressage direct de la table de données.

VI.6. Adressage des sections de type B :

Les variables de type B peuvent être adressées de deux façons différentes :

En adressage mot et bit, Exemple : B3:3/14 ; B3:252/0.

En adressage séquentiel, Exemple : B3/62 ; B3/4032.

VI.7. La scrutation Cyclique :

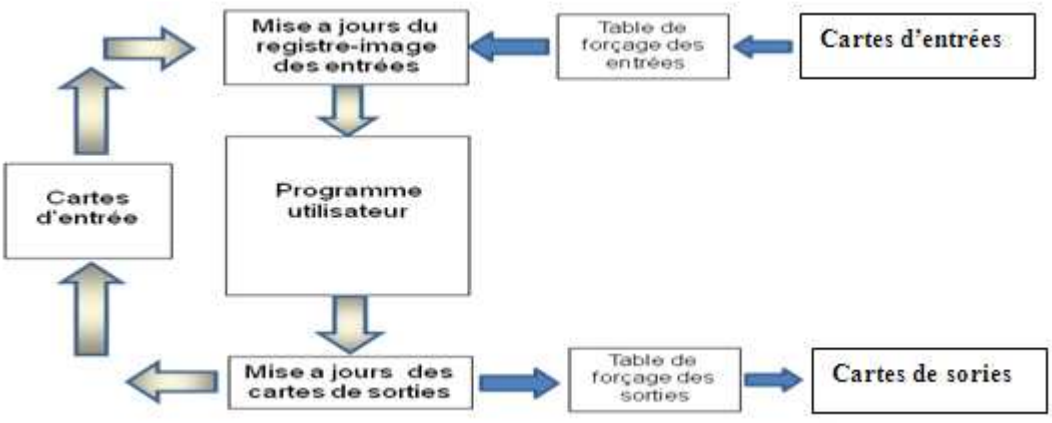


Figure III.15 : Diagramme de scrutation**.**

Remarque : La représentation ci-dessus ne prend pas en considération les interruptions.

1) - Le programme utilisateur peut comporter des instructions de rafraichissement immédiat de mots d'entrée ou de sortie.

2) - Si elles sont déclarées, des interruptions au temps ou sur entrées interruptives peuvent se produire à n'importe quel point du cycle.

VI.8. Les instructions de base :

VI.8.1. Généralités sur le langage à relais :

Le langage LADDER est dit langage à relais, c'est un langage de programmation inspiré des schémas électriques classiques.

Exemple :

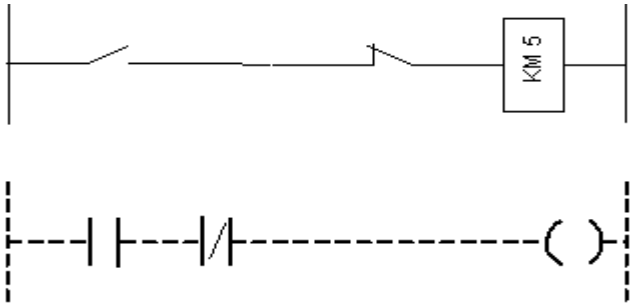
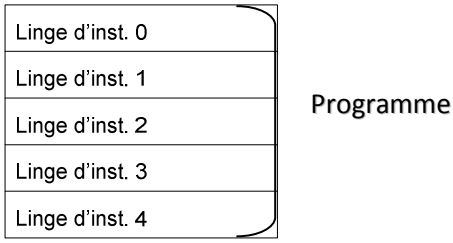


Figure III.16 : Programme de langage à relais.

Un programme en langage RELAIS est une suite de lignes d'instructions (RUNGS).



Chaque ligne d'instructions est composée d'une ou plusieurs actions.

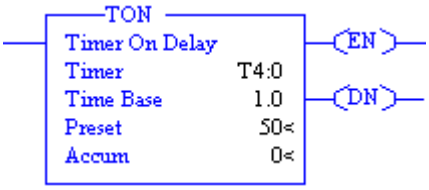
VI.8.2. Les instructions de type relais :

Fonction	Mnémonique	Représentation
Test a 1	XIC	
Test a 0	XIO	
Activation d'un bit	OTE	
Accrochage d'un bit	OTL	
Décrochage d'un bit	OTU	
Impulsion front montant	OSR	
Branches		

VI.8.3. Les temporisateurs :

On distingue les temporisations :

- TON : Temporisation au travail.
- TOF : Temporisation au repos.
- RTO : Temporisation à mémoire.



L'élément temporisation T (section):(élément)

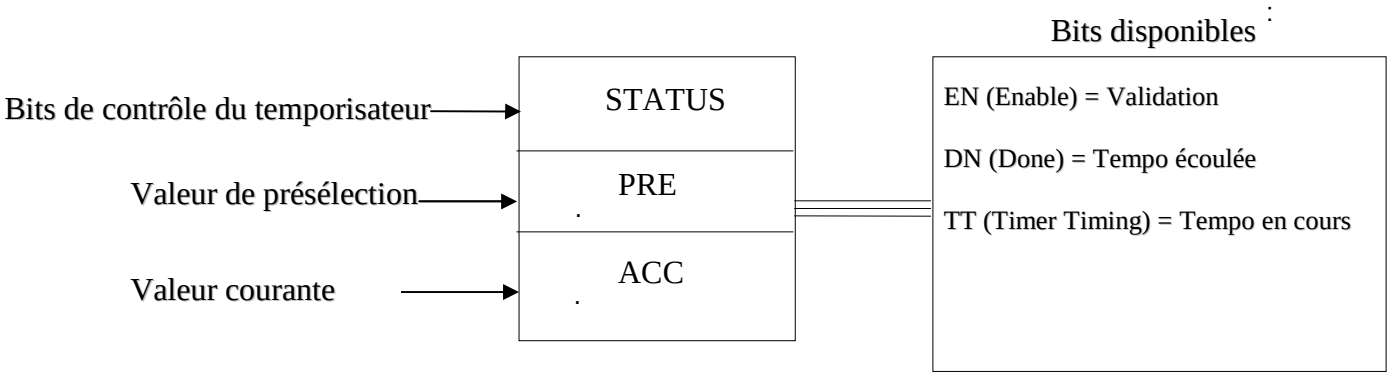


Figure III-17 : Instruction de temporisation.

L'adresse du temporisateur est obligatoirement un élément de type T.

La présélection est un entier signé, dans la pratique de 1 à 32767.

La valeur courante sera normalement comprise entre 0 et PRE.

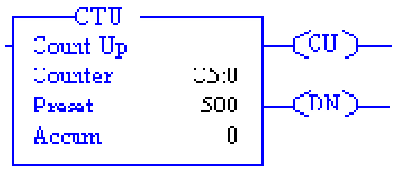
VI.8.4. Les compteurs :

Les opérations de comptage mettent en œuvre deux instructions et un type de variable :

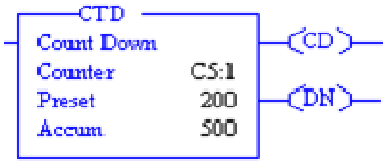
1. L'instruction CTU (Counter *UP*) : permet de compter des impulsions, c'est une action d'incrémentation de 1.
2. L'instruction CTD (Counter *DOWN*) : permet de décompter des impulsions, c'est une action de décrémentation de 1.

Les incrémentations et (ou) décrémentations sont totalisées dans une variable de type COMPTEUR [C (section):(élément)].

Comptage :



Décomptage :



Le comptage :

- L'adresse du compteur est obligatoirement un élément de type C.
- La présélection est un mot entier signé de -32768 à 32767.
- La valeur accumulée est un mot entier signé de -32768 à 32767.
- Les bits CU et CD servent à détecter les fronts d'impulsion.
- Le bit DN indique que la consigne est atteinte ou dépassée par le haut.
- Le bit OV (overflow) indique que l'accumulateur a dépassé vers le haut la valeur 32767.
- Le bit UN (underflow) indique que l'accumulateur a dépassé vers le bas la valeur -32767.

Remarque : les sous-éléments d'un compteur sont adressables (DN, UN, ACC, PRE).

VI.8.5. Les fonctions et modules de comptage rapide :

Les fonctions de comptage rapide permettent de compter des trains d'impulsions dont la fréquence est importante et que le temps de cycle de l'automate ne permet pas de compter directement.

Pour cela, des modules spécifiques sont disponibles, comme Le module 1746-HSCE du SLC modulaire qui :

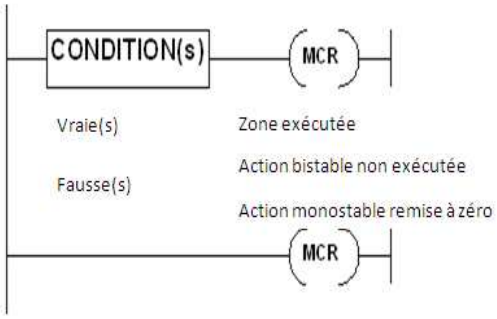
- Compte les impulsions (trains d'impulsions simples ou incrément ou décrément ou impulsions en quadratique).
- Communique le résultat de comptage à l'unité centrale.
- Gère ses propres sorties.

VII. L'organisation et la structure d'un programme :

VII.1. Le Contrôle de zone MCR et le saut JMP :

VII.1.1. MCR (Relais de Contrôle Maître) :

Zone de programme contrôlée par un relais maître :

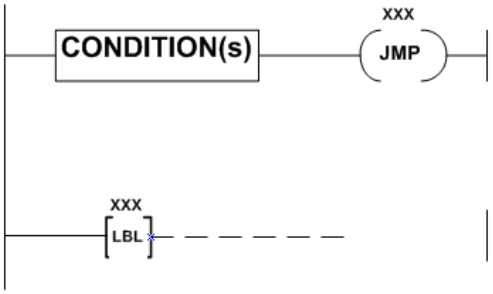


VII.1.2. Saut à une étiquette JMP :

JMP (JuMP) : Saut vers un sous-programme (étiquette).

LBL (LaBeL) : Etiquette.

XXX : Numéro de l'étiquette de 0 à 999.



Remarque : LBL peut être avant ou après JMP, on peut faire plusieurs JMP pour la même étiquette LBL dont le nombre maximum par fichier programme est 256.

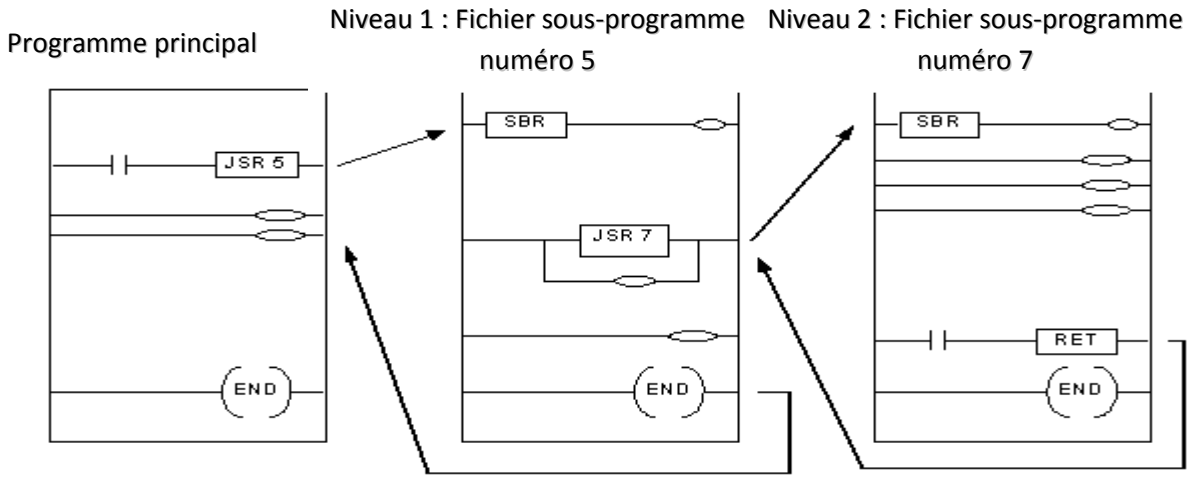
VII.2. L'appel de sous-programme JSR SBR RET :

Lorsque la ligne est vraie, l'instruction JSR provoque un saut à la première ligne du fichier sous-programme désigné. Dans le sous-programme, l'instruction RET provoque le retour de la scrutation à l'instruction JSR appelant.

On peut imbriquer jusqu'à 8 sous programmes pour les SLC 5/02 à 5/05.

- **JSR xxx :** (Jump to SubRoutine) saut à un sous-programme.
- **SBR :** (SuBRoutine) Sous-programme.
- **RET :** (RETurn) Retour.
- **xxx :** Numéro du sous-programme compris entre 3 et 255.

.Exemple de sous-programmes imbriqués à 2 niveaux :



Remarque : Sans l'instruction RET, l'instruction END (toujours présente dans le sous-programme) permet de revenir automatiquement à la ligne de l'instruction JSR correspondante. Les instructions SBR et RET sont facultatives, mais il est conseillé de les utiliser.

VII.3. La fin temporaire TND :

TND (Temporary eND) est une instruction temporaire, lorsque la ligne est vraie, l'instruction stoppe la scrutation du programme ou sous-programme en cours, rafraîchit les E/S, et retourne à la première ligne du programme principal.

VII.4. L'interruption du Programme SUS :

SUS est une instruction temporaire : lorsque la ligne est vraie, l'instruction arrête la scrutation du programme.

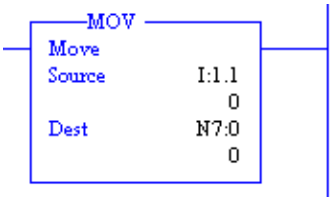
Le numéro du fichier interrompu est indiqué en S:8.

Le numéro de l'interruption SUS est indiqué en S:7.

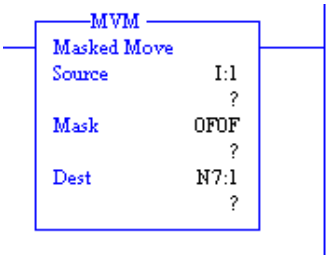
VIII. Les instructions de transfert, calcul et test :

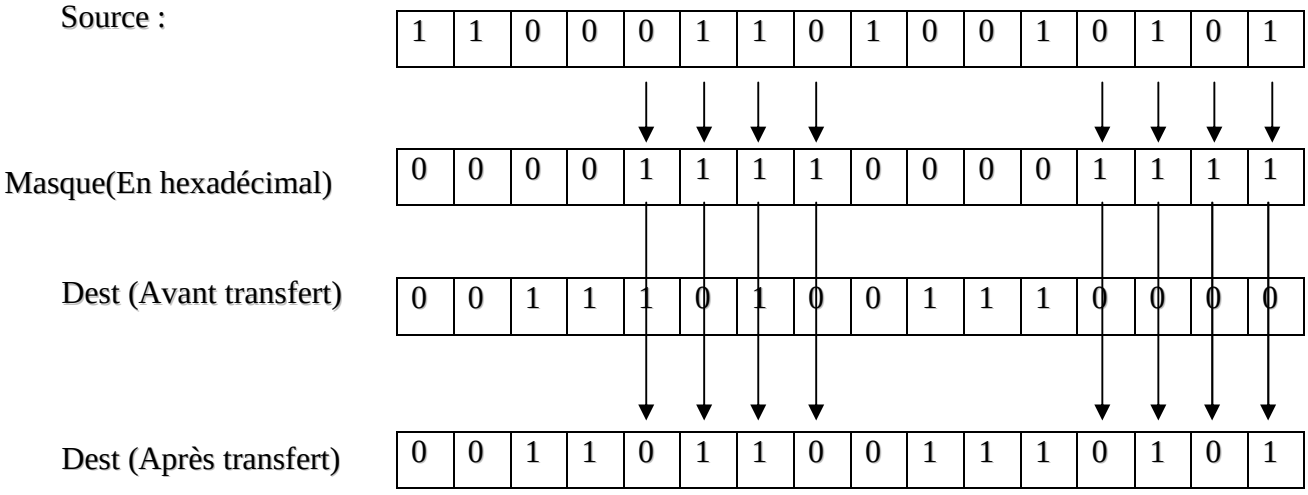
VIII.1. Les transferts :

a)- Le transfert simple : recopie une valeur (entière ou flottante) de la source vers la destination.



b)- Le transfert avec masque : recopie un mot de 16 bits de la source vers la destination à travers un masque.





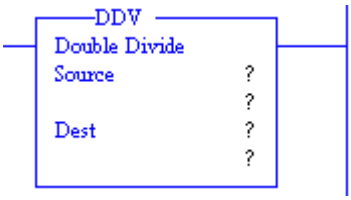
VIII.2. Les instructions de calcul :

VIII.2.1. Les opérations Arithmétiques :

Opération	Instruction	Opération	Instruction
Addition	ADD	Soustraction	SUB
Multiplication	MUL	Division	DIV
Division double	DDV	Racine carrée	SQR
Logarithme naturel	LN	Logarithme base 10	LOG
X à la puissance Y	XPY	Valeur absolue	ABS
Changement de signe	NEG	Remise à zéro	CLR
Permutation	SWP	Mise à l'échelle	SCP
Mise à l'échelle	SCL	Calcul	CPT
Rampe	RMP		

- L'instruction DDV (Division sur 32 bits):

L'instruction DDV permet d'effectuer en format MOT des calculs (en particulier des règles de trois) avec une grande précision.

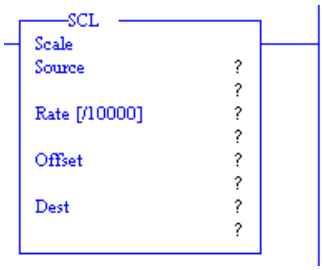


Cette DDV effectue la division du registre arithmétique (sur double mot) par la valeur source. Le quotient arrondi est chargé sur l'adresse de destination.

Le quotient non arrondi est placé dans le mot S:13.

Le reste de la division est placé dans le mot S:14.

- L'instruction SCL (Mise à l'échelle) :

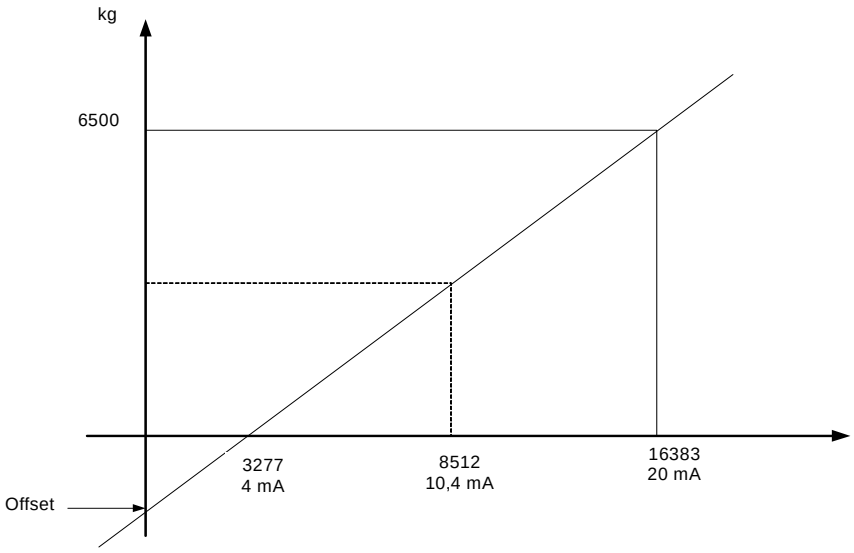


Les cartes analogiques du SLC 500 n'effectuent pas la mise à l'échelle. Cette dernière peut être réalisée avec une grande précision en format mot entier.

L'opération arithmétique réalisée par cette instruction peut s'exprimer sous la forme suivante :

Dest = (Source x Coef) + Offset.

Exemple de mise à l'échelle d'une entrée analogique :



Problème : sur une entrée analogique 4 à 20 mA de mesure de poids :

4 mA (3277 points) correspondent à 0 kg.

20 mA (16383 points) correspondent à 6500 kg.

Recherche du poids correspondant à 8512 points, soit 10,4 mA :

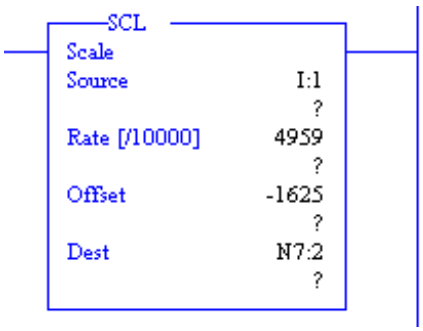
Calcul des paramètres de mise à l'échelle :

Source = 8512,

Coéf = (Dest Maxi – Dest Mini) / (Source Maxi – Source Mini) = (6500-0)/(16383-3277).

= 0,4959.

Offset = Dest Mini – (Source Mini × Coéf) = 0-(3277*0,4959) = -1625.



- L'instruction SCP (Mise à l'échelle avec paramètres) : Cette mise à l'échelle supporte les valeurs de données de type entier ou flottant. Elle permet une linéarisation entre les valeurs minimales et les valeurs maximales.

Exemple : Sur une entrée analogique 4 à 20 mA de mesure de poids :

4 mA (3277 points) correspondent à 0 kg.

20 mA (16383 points) correspondent à 6500 kg.

Pour I:1.0 = 8512 points ==> F8 :20 = 2596,330

- L'instruction RMP (Rampe) : Le bloc de contrôle est constitué de 7 mots. Et la destination est un entier.

VIII.2.2. Les opérations Trigonométriques :

Opération	Instruction	Opération	Instruction
Sinus	SIN	Cosinus	COS
Tangente	TAN	Arc sin	ASN
Arc cosinus	ACS	Arc tangente	ATN

VIII.2.3. Les opérateurs logiques et les conversions :

Opération	Instruction	Conversion	Instruction
ET logique	AND	Conversion binaire vers BCD	TOD
OU logique	OR	Conversion DCB vers Binaire	FRD
OU exclusif	XOR	Conversion radians vers degrés	DEG
Complément logique	NOT	Conversion degrés vers radians	RAD
Décodage sur 16 bits	DCD		

Si une valeur DCB dépasse 9999 ou est non exprimable, le bit arithmétique de dépassement est positionné à 1 ainsi que le bit S:5/0 d'erreur mineure.

- **Indications arithmétiques :** pour toutes les instructions arithmétiques, trigonométriques, logiques et de transfert, les indications sont mises à jour à l'issue de l'opération :

S:0/0 Retenue(Carry).

S:0/1 Dépassement (overflow). Attention **S:5/0** est également positionné.

S:0/2 Résultat égal à zéro (Zéro)

S:0/3 Signe : 0 = positif 1 = négatif.

Ces bits sont à interpréter immédiatement après l'instruction.

Le bit **S:5/0** indique une faute mineure (Dépassement) qui se transformera en faute majeure à la fin de la scrutation si elle n'a pas été traitée par le programme utilisateur.

VIII.3. Les tests sur mot et sur valeur :

Définition :

MOT désigne un mot entier de 16 bits.

VALEUR désigne une valeur numérique pouvant être exprimée suivant le type du SLC, sur mot de 16 bits ou sur flottant de 32 bits.

Opération	Instruction	Opération	Instruction
Egal	EQU	Différence	NEQ
Plus petit que	LES	Plus petit que ou égal	LEQ
Plus grand que	GRT	Plus grand que ou égal	GEQ
Comparaison masquée Sur MOT uniquement	MEQ	Test sur Limites	LIM

IX. Le fichier d'état :

IX.1. Généralités :

Le fichier d'état (Status) se trouve dans la section 2.

Il contient les informations concernant le diagnostic du processeur et permet de le configurer.

Pour accéder à la visualisation des écrans de ce fichier, deux chemins sont possibles :

1. Visu Données et demande S:xx, xx étant une adresse valide.
2. Etat processeur.

Les mots S: 0 à S:15 sont communs à tous les SLC.

Les mots S:16 à S:32 sont utilisés par le 5/02, le 5/03, le 5/04, le 5/05 et les MicroLogix.

Les mots S:33 à S:82 sont utilisés par le 5/03, le 5/04, le 5/05 et les MicroLogix.

Les mots au delà sont réservés au 5/04 (jusqu'à S:163).

Remarque :

Les pages de STATUS sont différentes d'un automate à un autre, et parfois pour un même automate d'un système d'exploitation à un autre.

IX.2. Fichier d'état sur RSLogix500 :

Les informations et réglages sont classés et rangés par catégories.

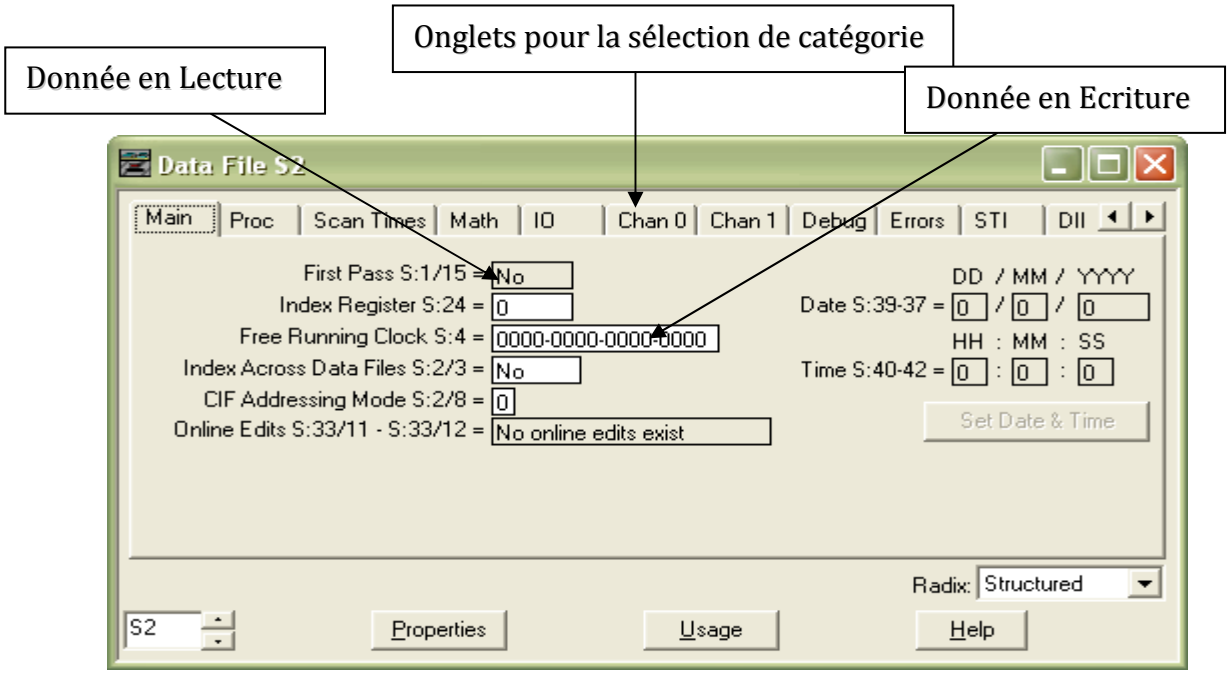


Figure III-18 : fichier d'état.

Liste des onglets :	
Main	: Infos générales
Proc	: Infos Processeur
Scan Times	: Temps de scrutation
Math	: Registres et Bits Arithmétiques
I/O	: Gestion des emplacements E/S
Chan 0	: Etat et configuration port 0 en RS232
Chan 1	: Etat et configuration port 0 en RS232
Debug	: Paramètres de mise au point.
Errors	: Bits et codes de défauts
STI	: Etat et configuration de la STI
DII	: Etat et configuration de la SDII
Protection	: Bit de protection
Module Mémoire	: Bits de gestion de la mémoire
Forces	: Etat des forçages.
Global Data	: Données globales sur DH+

IX.3. Informations importantes :

1. Indication de forçage(s) :

S:1/6 = 1 : Forçage (s) declaré (s).

S:1/5 = 1 : Forçage(s) activés.

Remarque : Le programme utilisateur ignore quelle adresse est forcée.

2. Indication de défaut pile :

S:5/11 = 1 : Défaut pile

Remarque : Tous les SLC possèdent un condensateur de sauvegarde.

3. DH-485 actif sur canal 1 :

Si S:1/7 = 1 : Au moins une autre station est présente sur DH-485 sur canal 1.

Remarque :
- Détail des stations présentes sur S:9 et S:10 : 1 bit par station.
- S:33/4 : Même rôle sur canal 0.

4. Dépassement mathématique :

Si S:5/0 = 1 : faute mineure se transformant en faute majeure en fin de scrutation.

5. Durée de la dernière scrutation :

S:3 : En base 10 ms.

S:35 : En base 1 ms (5/03, 5/04 et 5/05).

6. Code d'erreur :

S:6 (En hexadécimal) : code d'erreur.

S:21 : Dernier fichier traité, S:20 : Dernière ligne traitée.

7. Edition en ligne :

S:33 : Bits 11 et 12 : état des éditions en ligne. (Modifications de programme en cours).

8. Présents DH+ sur canal 1 du SLC 5/04 :

S:34/1 = 1 : Mise à jour active de la table des présents DH+, ces derniers sont indiqués par les fichiers de S:83 à S:86 (1 bit par station).

Remarque : Le réglage par défaut est sans mise à jour.

9. Augmentation du débit des communications :

Canal 1 : S:2/15 et voir S:33/7.

Canal 0 : S:33/5 et S:33/6.

IX.4. Les modes de redémarrage:

Le mode de redémarrage d'un automate SLC 500 dépend des facteurs suivants :

- Le type de SLC (5/01, 5/02, 5/03, 5/04, 5/05).
- Eventuellement la série de l'automate.
- Le paramétrage des bits de configuration du fichier d'état.
- La présence d'un module EEPROM.

1. Exécution du défaut majeur à la mise sous tension :

Si S:1/8 = 1 : Pour tous les SLC.

Conséquence : S:1/13 et S :5/0 à S :5/7 remise à zéro.

2. Exécution du sous-programme d'erreurs :

S:1/9 = 1 : SLC 5/02 à SLC 5/05.

Conséquence : exécution systématique du sous-programme d'erreurs à la mise sous tension.

3. Transfert de l'EEPROM => RAM sur défaut mémoire :

Si S:1/10 = 1.

Conséquence : lecture de l'EEPROM (Programme et données) en cas de défaut de RAM.

4. Transfert inconditionnel de l'EEPROM vers la RAM :

Si S:1/11 =1.

Conséquence : lecture de l'EEPROM (Programme et données) à chaque retour secteur.

5. Mise en mode EXECUTION :

Si S:1/12 = 1.

Conséquence : Mise en RUN après la lecture de l'EEPROM à chaque retour secteur.

Conclusion :

L'étude précédente nous a permis de connaitre les principes de base pour l'exploitation du langage de programmation RSLogix 500 montrant la procédure de création d'un projet ainsi que le rôle et l'utilisation des différentes instructions. Ceci est le point de départ pour accomplir la première phase de notre travail qui est la détection des défauts de l'assembleuse pose à plat, pour pouvoir ensuite les afficher sur le pupitre opérateur à l'aide du logiciel PanelBuilder 32 que nous allons décrire au chapitre suivant.

Chapitre IV

L’interface homme/machine PanelView 550 et son progiciel PanelBuilder 32

Introduction :

L'interface homme machine a connu une évolution très importante. Ainsi, dans les années 1950, il fallait recourir à des tableaux de connexion, sur lesquels on enfichait des câbles reliant deux opérateurs, pour programmer des opérations mathématiques sur les tabulatrices électromécaniques, lointains ancêtres de nos calculatrices programmables. Suite à l'automatisation industrielle, l'opérateur humain a été contraint de conduire ou de superviser des machines automatisées, en réduisant les prises d'information et les actions directes sur le processus, ce qui conduit à l'élaboration d'interface d'interaction Homme/Machine, flexible et aussi lisible pour un simple opérateur. Le dialogue est d'autant plus facile que l'écran comporte des images avec des synoptiques, des graphes, des bargraphes…etc.

La famille d'interfaces opérateur standard PanelView Allen-Bradley offre des affichages haute luminosité en couleur, en niveaux de gris ou monochrome, avec des tailles d'écran allant de 5 a 14 pouces, et possèdent des fonctions performantes d'interfaçage avec traitement évolué des alarmes, sécurité des écrans, indicateurs analogiques, carte mémoire ATA PC, support multilingue et impression en ligne.

Le progiciel PanelBuilder32 est le logiciel le plus approprié pour la conception de ce type d'interface de commande/diagnostic pour les panels view suivant : PV900 Mono, PV900 Couleur, PV 600, PV 550, PV 300, PV 300 Micro, PV 1400, PV 1000 Couleur, PV 1000 gris.

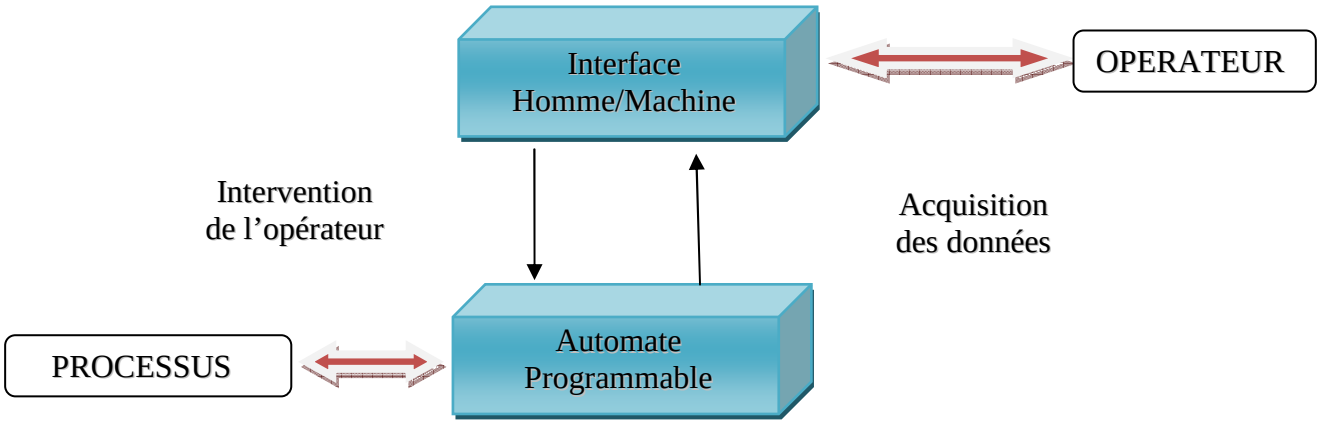


Figure IV.1 : L'interface Homme/Machine dans un processus automatisé

I. Les IHM PanelView :

On peut utiliser les terminaux opérateur PanelView pour une large variété d'applications de contrôle et de surveillance de machines. Ils fonctionnent avec des applications à conception personnalisée. La première fois que le terminal est mis sous tension (sans fichier d'application chargé), il affiche le menu du Mode de Configuration.

Remarques :

- Les terminaux RIO ont une application "prête à l'emploi" pour la configuration des paramètres de la communication RIO.
  Si une application est chargée, le terminal affiche la vue de démarrage de l'application.
- La façon dont le terminal fonctionne dépend de l'application et du type de terminal (à écran tactile ou à touches).

- **Terminaux à écran tactile :**

Les applications créées pour terminaux à écran tactile sont contrôlées en touchant des objets sur l'écran.

- **Terminaux à touches :**

Les applications créées pour terminaux à touches sont contrôlées en pressant les touches de fonction correspondant aux objets de la vue. Les données sont entrées manuellement à l'aide des touches d'entrée numérique.

I.1. Menu du mode de Configuration :

Depuis le menu du Mode de configuration des terminaux on peut :

• sélectionner une langue pour les affichages du terminal.
• charger/transférer des applications avec une carte mémoire.
• configurer ou afficher les paramètres de communication.
• sélectionner les valeurs de présélection.
• obtenir des informations sur le terminal et l'application.
• configurer l'écran.
• faire les réglages de l'heure et de la date.

• configurer les paramètres de l'imprimante (pour les terminaux ayant un port imprimante RS-232).

• repasser en mode Run.

-

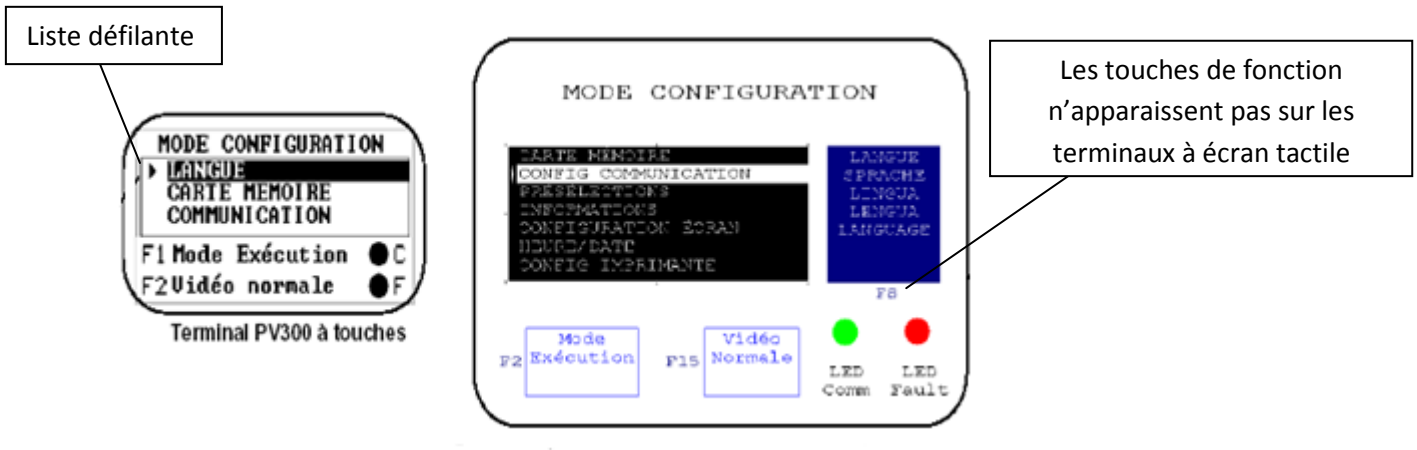


Figure IV.2 : Mode configuration.

I.2. Messages du terminal :

Les messages du terminal :

• affichent l'état d'une opération.

• indiquent les défauts mineurs, les erreurs, ou les fautes d'entrées numériques.

• affichent des invites opérateur.

I.3. Impression :

Les terminaux PanelView munis d'un port imprimante RS-232 peuvent imprimer :

• les messages déclenchés sur un afficheur de messages.

• les états déclenchés d'un voyant multi-états.

• les messages d'alarmes.

• la liste d'alarmes.

Les attributs d'impression des objets sont définis dans l'application.

I.4. Liste d'alarmes :

Le terminal PanelView stocke les informations sur les alarmes déclenchées dans une liste d'alarmes. Cette liste enregistre les informations sur autant d'alarmes (jusqu'à 100) que

le terminal peut maintenir dans la mémoire RAM non volatile. Le nombre des alarmes stockées dans la liste est configuré à l'aide du logiciel PanelBuilder32.

La liste d'alarmes enregistre, pour chaque alarme, les données suivantes :

• l'indicateur d'acquittement.
• la date et heure de l'alarme.
• la date et heure de l'acquittement.
• la valeur de déclenchement de l'alarme.
• le texte de l'alarme, variables comprises.

La liste d'alarmes est effacée :

• quand une application est chargée dans le terminal.
• quand le terminal est réinitialisé ou quand l'alimentation est coupée puis rétablie.

L'objet Liste d'alarmes peut apparaître sur le bandeau d'alarmes ou sur d'autres vues de l'application. Les données qui s'affichent dans la liste d'alarmes sont configurées à l'aide du logiciel PanelBuilder32.

I.5. Réinitialisation du terminal :

Cette fonction réinitialise le terminal PanelView (comme si l'alimentation avait été coupée puis rétablie). La manière de cette réinitialisation diffère d'un terminal à un autre :

- Soit en utilisant les touches curseur.
- Soit en pressant le bouton Reset se trouvant sur la face arrière du terminal par un objet non conducteur.

I.6. Configuration de la communication :

Pour modifier ou afficher les paramètres du terminal PanelView, on Sélectionne ‘’Config Communication’’ dans le menu du mode Configuration. La vue qui apparaît dépend du protocole de communication du terminal.

I.7. PanelView 550 :

L'interface Homme/Machine PanelView 550 d'Allen Bradley est un terminal à écran à touches très robuste utilisé dans une grande part d'installations industrielles, notamment à

l'usine Michelin Algérie, et en particulier sur la machine assembleuse pose à plat (PAP) sur laquelle s'effectue notre travail. On l'utilise pour le contrôle et la visualisation des entrées/sorties et les différentes opérations programmées et exécutées par l'automate auquel il est relié.

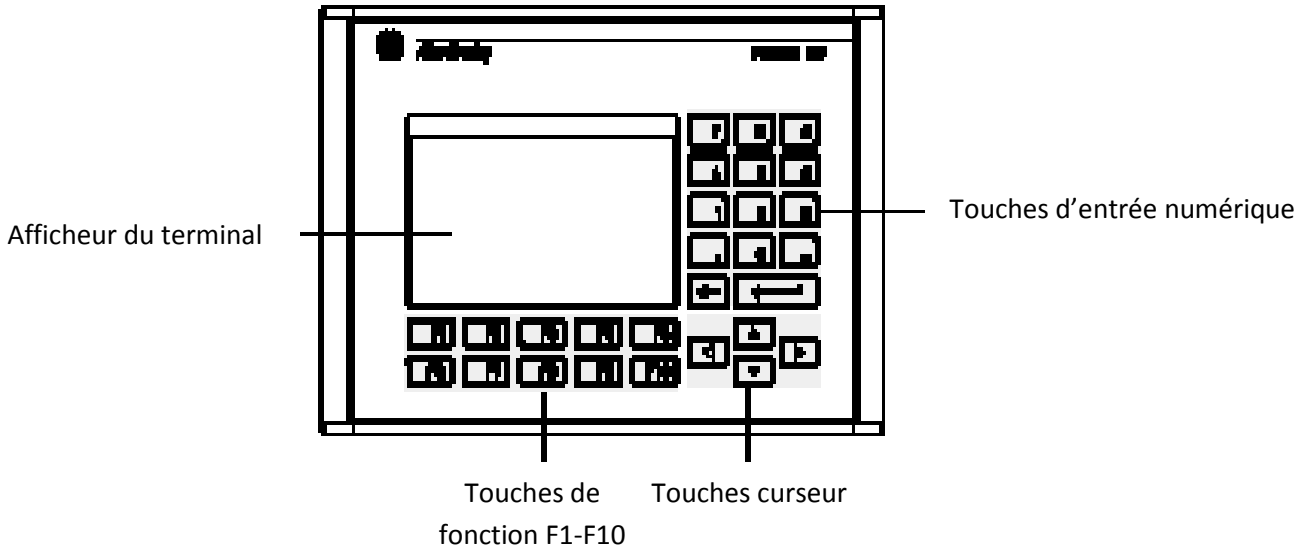


Figure IV.3 : Face avant du PV 550.

I.7.1. Ces caractéristiques :

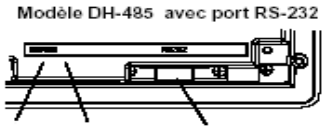
Il présente les caractéristiques rassemblées dans le tableau suivant :

Caractéristique	Description
1 - Borne de connexion d'alimentation	Connecte à une source d'alimentation externe.
2 - Logement de carte mémoire	Reçoit une carte mémoire pour stocker des applications.
3 - Port de communication DH-485	Connecte à un automate SLC ou MicroLogix, au réseau DH-485 ou à une alimentation de panneau.
4 - Connecteur de programmation	Connecte à un convertisseur d'interface pour

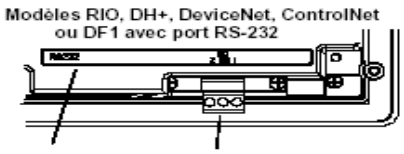
DH-485.	PC (référence 1747-PIC) pour le transfert d'applications. Connecte aussi à un programmateur SLC, tel qu'un terminal portatif.
5 - Port de communication RS-232	Connecte au port Canal 0 d'un automate SLC 5/03 ou 5/04 pour la communication DH-485 point-à-point ; connecte à un automate MicroLogix, connecte aussi au port série RS-232 d'un ordinateur pour le transfert d'applications.
6 - Port RIO	Connecte à un scrutateur ou à un sous-scrutateur d'automate sur un réseau RIO.
7 - Port de communication DH+	Connecte au port DH+ d'un automate PLC ou SLC 5/04 sur une liaison DH+.
8 - Connecteur DeviceNet	Connecte au réseau DeviceNet.
9 - Connecteur ControlNet	Connecte au réseau ControlNet .
10 - Port de communication DF1	Connecte au port DF1 d'un automate PLC, SLC ou MicroLogix.
11 - Port imprimante/ transfert de fichiers RS-232	Connecte à une imprimante. Sur les terminaux RIO, DH+ et DeviceNet ce port connecte aussi au port série RS-232 d'un ordinateur pour le transfert d'applications.

Tableau IV.I : Caractéristique de PanelView 550.

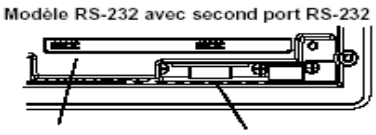
Parmi les modèles d'IHM PanelView 550 on a



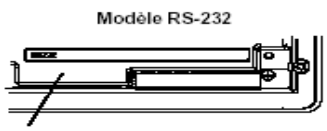
4 7 11



11 6, 7, 8, 9 ou 10



5 11



5

I.7.2. PanelView 550 modèle RIO, DH+ avec port RS-232 :

Ce type de terminal est doté des ports suivants :

- Voie DH+ intégrée, prenant en charge :
  - Les communications DH+.
  - Les communications DH-485.
  - Les communications RIO.
- Voie RS-232 intégrée, prenant en charge :
  - Le DF1 Full-Duplex.
  - Le DF1 Half-Duplex.
  - Le DH-485.
  - Les E/S ASCII.

II. Le progiciel de conception et de configuration d'interface PanelBuilder32 :

II.1. Présentation générale :

Le logiciel PanelBuilder32 permet de créer des applications de panneaux de commande pour la gamme de terminaux PanelView standard. Ces terminaux offrent des options de communication souples qui permettent d'exécuter les applications dans une grande variété d'environnements, dont : DH+, RIO, DH-485, RS-232 (DH-485), DeviceNet, ControlNet, DF1, Protocoles Modbus et autres.

PanelBuilder32 contient les barres d'outils (donnant accès aux commandes et boîtes de dialogue) suivantes :

- **Barre des menus :** La barre des menus contient toutes les commandes nécessaires à l'utilisation de PanelBuilder32
- **Zone de travail :** La zone de travail sert à configurer les vues, de façon qu'elles soient compréhensibles par l'utilisateur, et très faciles à manipuler et à consulter les résultats.
- **Boîte à outils :** La fenêtre des outils propose un choix d'objets simples ou complexes qu'on insère dans les vues, par exemple des objets graphiques et éléments de commande.

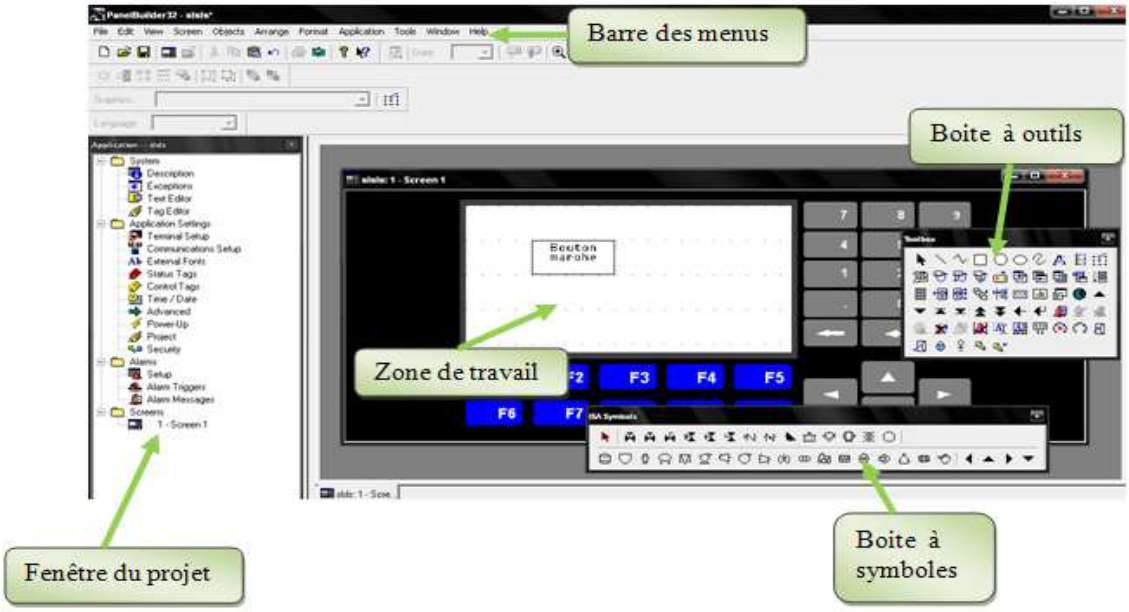


Figure IV.4 : Vue d'ensemble du progiciel PanelBuilder32.

II.2. Les fonctions de Panel Builder 32 :

II.2.1. Editeurs tableurs :

Les tableurs sont utilisés pour simplifier de nombreuses opérations d'édition, comme:

- Editer les états d'objets multi-états ou listes.
- Editer les textes d'une application.
- Editer les alarmes et les déclenchements d'alarmes.

La plupart des options de formatage pour le texte et les objets peuvent être configurées directement dans les cellules du tableur.

II.2.2. Objets et graphiques prédéfinis :

Pour simplifier le développement d'applications, PanelBuilder32 fournit un ensemble d'objets prédéfinis (tels que boutons-poussoirs, bargraphes…). Des graphiques supplémentaires sont disponibles pour créer des dessins particuliers ou améliorer les vues. On peut aussi importer des graphiques bitmap créés avec d'autres programmes. De plus, différentes options de format sont utilisables pour modifier l'aspect des objets et du texte.

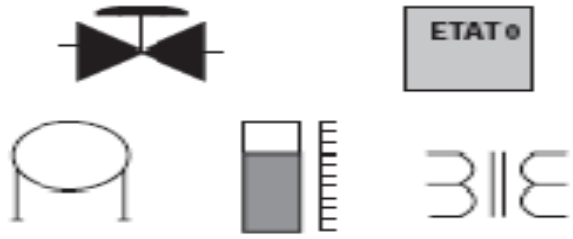


Figure IV.5 : Exemple d'objets graphiques

II.2.3. Objets globaux :

Un objet global fait référence à un objet qui peut apparaître plusieurs fois dans une application. Tout objet non-graphique peut être désigné comme objet global. On peut accéder à un objet global à partir de n'importe quelle vue. Quand on modifie un objet global, PanelBuilder32 met automatiquement à jour toutes ses reproductions.
Le terminal PanelView ne stocke qu'une copie d'un objet global quel que soit le nombre de reproductions qui lui sont associées.

II.2.4. Jauge analogique :

La jauge analogique utilise une aiguille tournante pour afficher une variable de procédé telle que la vitesse, la température ou la pression. Elle comprend une échelle circulaire graduée, **1** à **4** aiguilles et une étiquette avec un texte intérieur statique ou des données variables. L'échelle circulaire fait partie de la jauge mais peut être créée séparément. Pour accéder à la jauge, on sélectionne **Objets>Indicateurs graphiques>Jauge**.

II.2.5. Echelle circulaire :

L'échelle circulaire existe avec ou sans graduations principales/secondaires à l'intérieur ou à l'extérieur de l'échelle. Ses angles minimum/maximum sont réglables. Elle peut être désactivée afin que seules les graduations soient visibles. Pour y accéder, on clique sur **Objets>Indicateurs graphiques>Echelles>Circulaire**. Utilisée avec une jauge analogique, l'échelle circulaire peut servir d'échelle secondaire.

II.2.6. Incrémentation/décrémentation numérique :

L'objet Incr/Décr est similaire au pointeur d'entrées numériques, sauf qu'il utilise des incréments prédéfinis pour mettre à jour une valeur numérique. Pour y accéder, on clique sur **Objets>Entrée numérique> Incr/Décr**.

II.2.7. Editeur de points :

L'éditeur de points est utilisé pour entrer, mettre à jour, imprimer, importer/exporter des points de l'application. Chaque point a des attributs qui définissent comment un objet interagit avec une adresse d'automate. On peut entrer tous les points en même temps à l'aide du tableau ou un seul point à la fois à l'aide du masque d'édition de points.

II.2.8. Protection des vues :

PanelBuilder32 offre des fonctions de protection par mot de passe, permettant de limiter l'accès des vues et du terminal (écrans de configuration inclus) aux opérateurs autorisés. On peut également créer des vues pour permettre aux opérateurs habilités de sélectionner et de modifier les mots de passe des autres opérateurs.

II.3. Création d'un projet PanelBuilder32 :

II.3.1. Vue de démarrage :

Au lancement de Panel Buider32, une boîte de dialogue s'ouvre indiquant les opérations qu'on peut effectuer : création, ouverture, chargement ou transfert d'application. Les options de cette boîte de dialogue correspondent à certaines commandes du menu **Fichier.**

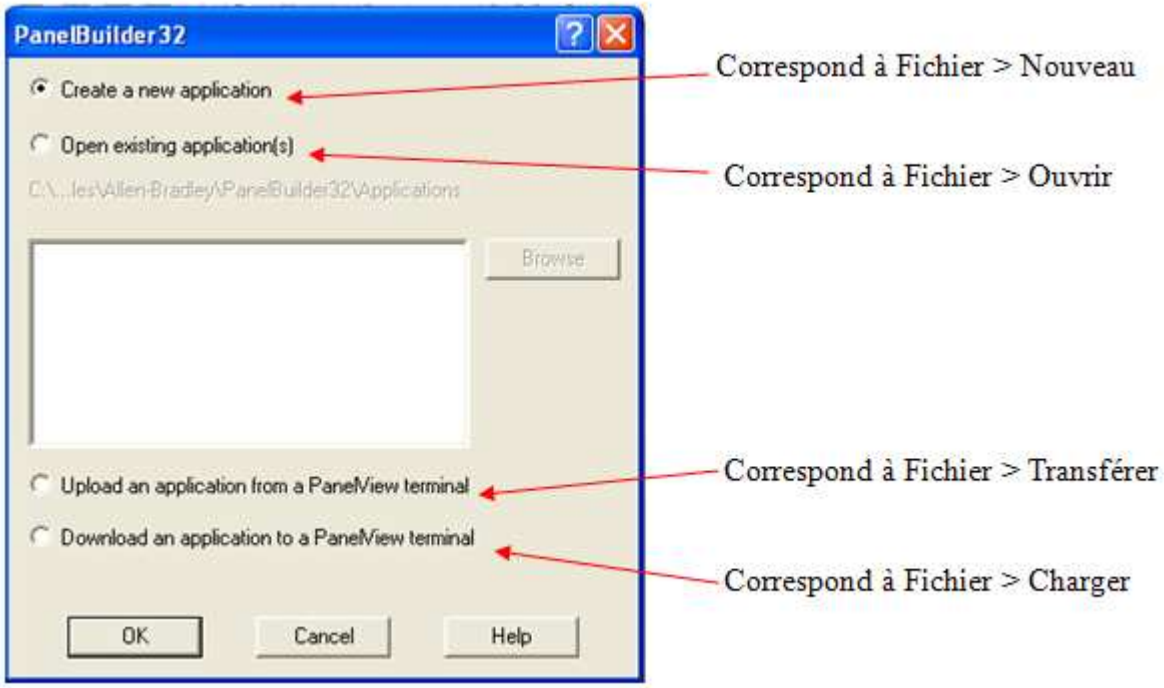


Figure IV.6 : Vue de démarrage

II.3.2. Fenêtre d'application :

La fenêtre Application, qui s'affiche à l'ouverture d'une nouvelle application ou d'une application existante, offre une vue générale de l'application. Elle contient des dossiers sous forme d'arborescence permettant un accès rapide aux composants principaux de l'application. Il suffit de cliquer deux fois sur une icône du dossier pour ouvrir une vue, une boîte de dialogue ou un tableur. Les options de la fenêtre d'application sont accessibles également depuis les menus.

II.3.3. Fenêtre de vue :

Cette fenêtre (pareille au panneau avant d'un terminal) apparaît à l'ouverture d'une nouvelle vue ou d'une vue existante. La barre de titre indique le nom et le numéro de la vue. Les vues peuvent être déplacées et dimensionnées comme les autres fenêtres.

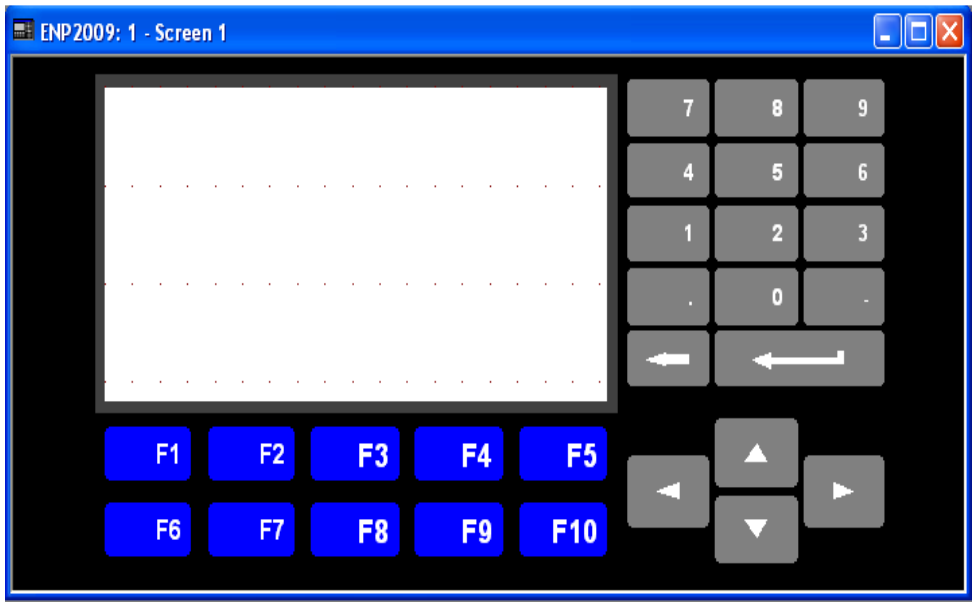


Figure IV.7 : Fenêtre de vue PanelBuilder32.

II.3.4. Création d'une nouvelle vue :

Pour créer une nouvelle vue, on clique sur le bouton ''New screen'' de la barre d'outils, la fenêtre de dialogue suivante s'ouvre, sur laquelle on peut modifier ses paramètres.

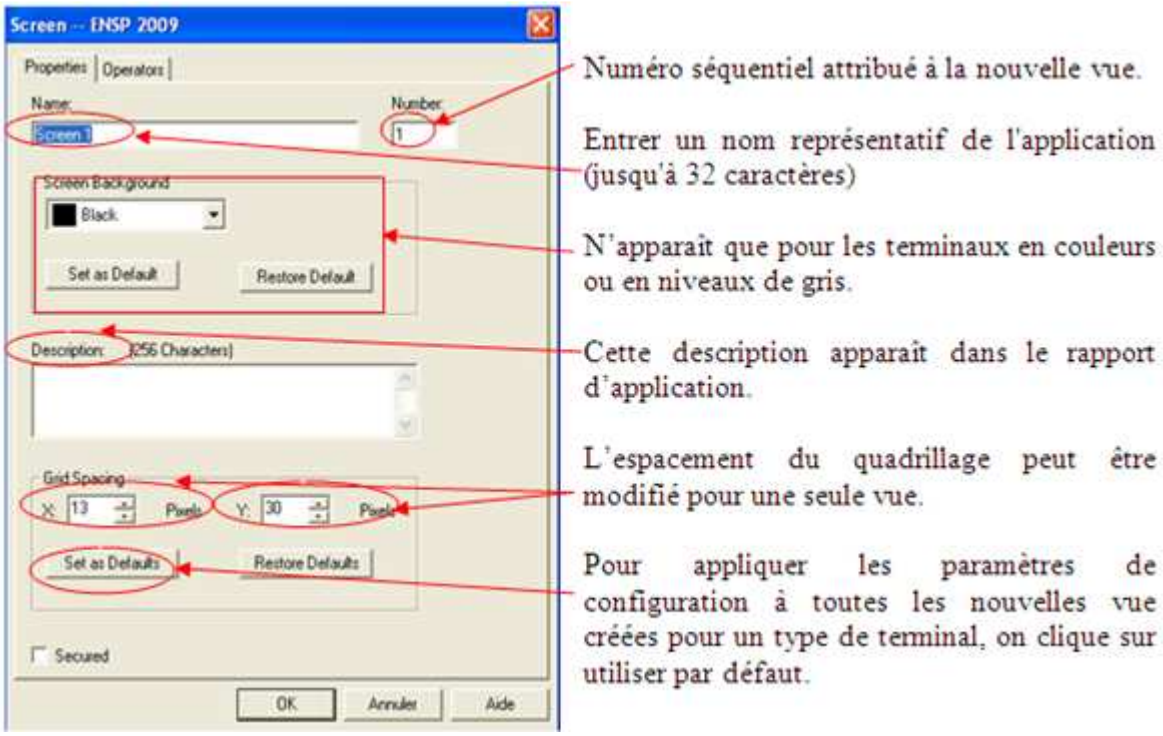


Figure IV.8 : Fenêtre de création d'une nouvelle vue.

II.4. Objets :

II.4.1. Présentation des objets :

Les objets sont des images graphiques qui s'affichent sur le terminal PanelView. Un objet peut être une ligne de texte, un graphique, un contrôle ou un bandeau d'alarmes. Certains objets sont statiques : ils fournissent des informations visuelles mais ne possèdent aucune fonction de contrôle. D'autres, tels que les sélecteurs listes de contrôle, sont dynamiques et interagissent avec l'opérateur du terminal et avec l'automate. Les objets dynamiques sont reliés à l'automate par un point. Celui-ci contient l'adresse de l'objet ainsi que d'autres paramètres tels que le type de données et permet la lecture et l'écriture des données d'un automate.

Exemples d'objets :

Les objets	**Types**
Boutons-poussoirs	- Momentané - Maintenu - Maintenu - Avec bit de verrouillage - Multi-états
Sélecteurs Listes de contrôle	Standard Pilotée
Objets d'entrée numérique	Pilotée Validation de clavier
Sélecteurs de vues	- Bouton Aller à -Bouton Aller à Config. - Bouton Retour - Sélecteur Liste de vues
Touches de liste	- Monter, Descendre -Page précédente, suivante - Origine, Fin - Retour arrière - Entrée
Alarmes	- Bandeau d'alarmes - Bouton Acquitter - Bouton Supprimer - Bouton Imprimer - Bouton Tout acquitter - Liste d'alarmes - Bouton Imprimer la liste d'alarmes - Bouton Effacer la liste d'alarmes
Graphiques et Texte	- Ligne, Rectangle, Cercle, Ellipse - Forme libre - Symboles ISA - Graphiques importés - Texte d'arrière-plan

Tableau IV.II : Exemples d'objets.

II.4.2. Configuration des propriétés des objets :

La plupart des objets, sauf les graphiques et le texte, ont des propriétés qui définissent leur fonctionnement et leur interaction avec le terminal PanelView et l'automate.

Pour accéder à ses attributs, soit on clique deux fois sur un objet, soit une seule fois avec le bouton gauche de la souris puis on sélectionne Edition>Propriétés de l'objet.

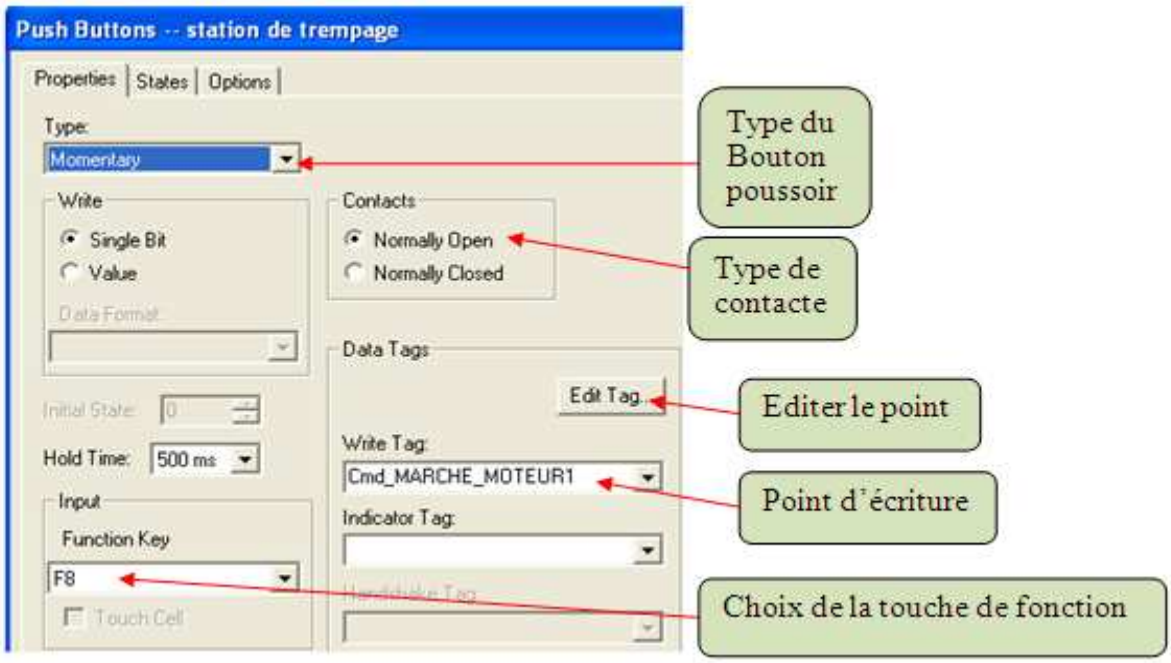


Figure IV.9 : Configuration des propriétés des objets.

Les attributs sont particuliers à chaque objet. Pour les objets de contrôle, le champ Entrée permet de déterminer si l'objet est activé par une touche de fonction ou par une cellule tactile sur le terminal PanelView. Pour les terminaux à écran tactile, la cellule tactile est automatiquement activée.

Point d'écriture : définit l'adresse d'automate où le terminal écrit les données de l'objet. Les objets peuvent être associés à plus d'un point.

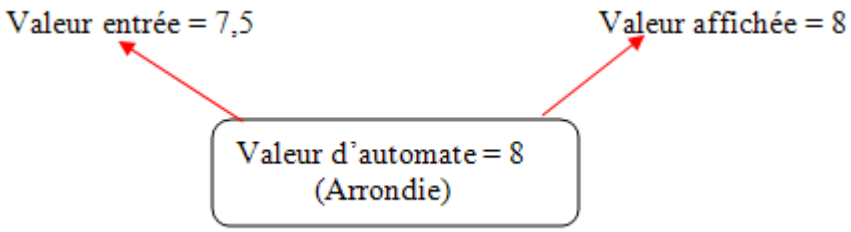
Point de lecture : définit l'adresse d'automate que le terminal lit pour afficher les données.

II.4.3. Mise à l'échelle des données :

Les données entrées par l'opérateur peuvent être mises à l'échelle à l'aide de la formule suivante :

$$Valeur\ stockée\ dans\ l'automate = \frac{Valeur\ entrée - Offset}{Echelle}$$

Les valeurs d'échelle et d'offset sont entrées dans l'éditeur de points. Si une valeur mise à l'échelle n'appartient pas à la gamme du type de données sélectionné, le terminal affiche un message d'erreur.

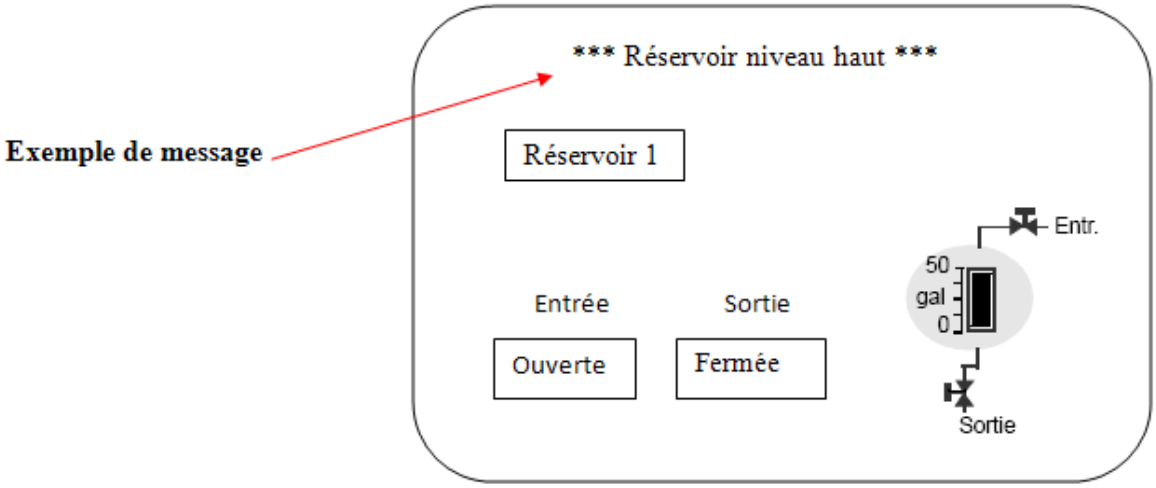


Précision et arrondi : Sauf pour les valeurs à virgule flottante, l'automate stocke les valeurs sous forme d'entiers. Les chiffres à droite d'une virgule décimale sont arrondis. Toutefois, le terminal peut mettre à l'échelle des valeurs décimales en/à partir de nombres entiers**.**

II.4.4. Affichages de messages :

Les affichages de messages donnent des informations d'état ou des instructions à un opérateur. Ils peuvent afficher différentes informations :

- état d'un procédé.
- invites opérateur.
- heure / date.
- variable numérique ou ASCII.



Les affichages de messages peuvent être déclenchés par :

• Bit unique.

• Bit de poids faible.

• Valeur.

II.4.5. Alarmes :

II.4.5.1. Présentation du système d'alarmes :

Les alarmes avertissent l'opérateur, au moyen d'une mise en garde visuelle ou imprimée, qu'une valeur de déclenchement a été atteinte ou dépassée. Cette valeur de déclenchement peut être associée via l'automate à différents capteurs ou programmes de surveillance du contrôle de production. Toutefois, certaines considérations de sécurité doivent être prises en compte :

Liste d'alarmes : Quand les alarmes sont déclenchées, elles sont ajoutées à la liste d'alarmes. Celle-ci permet à l'opérateur de visualiser et de gérer plusieurs alarmes.

Bandeau d'alarmes : Définit la zone de la vue où l'opérateur peut visualiser les alarmes et les boutons de contrôle et y accéder. Quand une alarme est déclenchée, le bandeau d'alarmes s'affiche, quelle que soit la vue activée.

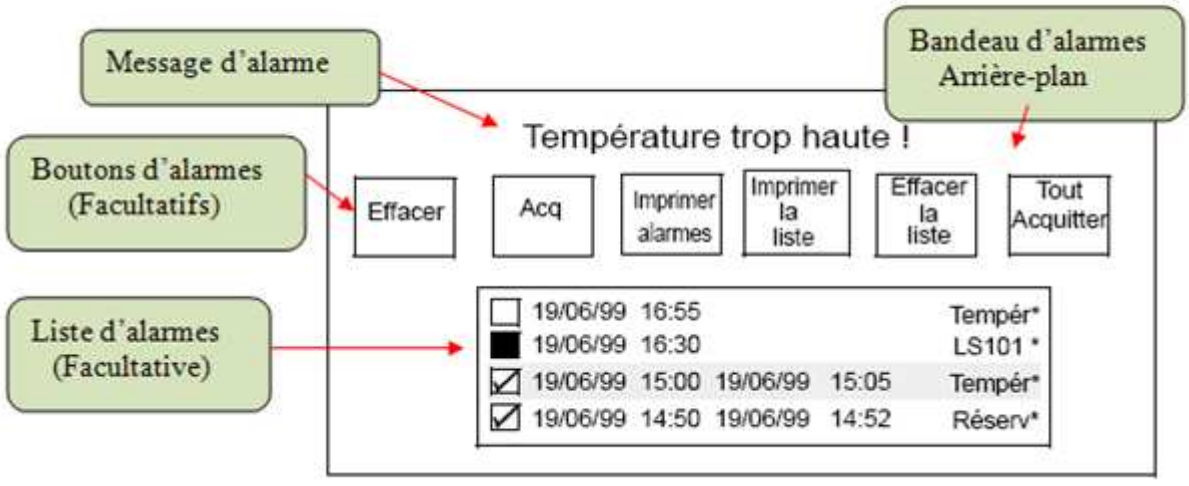


Figure VI.10 : Exemple de bandeau d'alarmes.

Le bandeau d'alarmes est un affichage global créé une seule fois dans une application mais apparaissant au même endroit sur chaque vue. Il n'apparaît que lorsqu'une alarme est déclenchée. Puisque le bandeau d'alarmes s'affiche par dessus les objets de contrôle et

d'affichage, on peut spécifier si on veut que tous les objets de la vue ou seuls les objets recouverts soient désactivés pendant l'affichage du bandeau.

Le bandeau disparaît de la vue lorsque :

- L'opérateur appuie sur les boutons Acquitter, Effacer, ou Tout acquitter.
- L'automate supprime ou acquitte l'alarme.
- Le terminal passe en mode de Configuration.

II.4.5.2. Définir les déclenchements des alarmes :

Chaque alarme est associée à un déclenchement. Pour chaque déclenchement, il faut définir :

- **Un point de déclenchement :** définit l'adresse d'automate surveillée par le terminal pour une valeur ou un bit de déclenchement d'alarme. La plupart des applications utilisent un seul point de déclenchement.

- **Un type de déclenchement :** définit quel type de données déclenchera l'alarme (bit, valeur, bit de poids faible).

Les déclenchements sont attribués aux alarmes individuelles dans la section **Messages d'alarme**s de la boîte de dialogue **Alarmes**.

Pour définir les déclenchements d'alarmes :

- Ouvrir le dossier **Alarmes** dans la fenêtre d'application et cliquer deux fois sur l'icône **Déclenchement d'alarmes** ou
- Sélectionner **Application>Configuration des alarmes** puis sélectionner la section **Déclenchement d'alarmes**.

II.4.5.3. Exemples de déclenchement d'alarmes sur Bit ou LSBit :

Cette section présente un exemple de 3 alarmes utilisant un déclenchement de type **Bit**. La plage des données pour un déclenchement sur bit va de 0 à 255.

Dans l'exemple ci-dessous, 3 alarmes (A, B, C) sont attribuées au déclenchement d'alarmes point_alm. C'est un déclenchement de type Bit. Dans ce cas, le champ Valeur/Bit est un offset de bit depuis l'adresse du Point de déclenchement (et non une valeur).

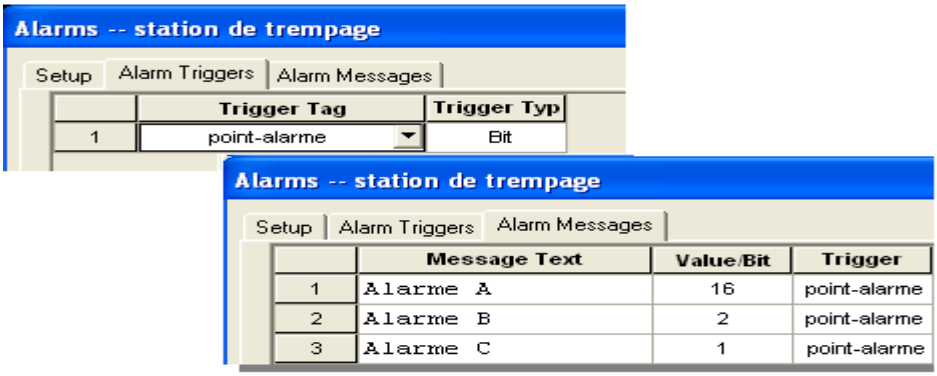


Figure IV.11 : Fenêtre des propriétés des alarmes.

Dans l'éditeur de points, la définition du point point_alm est :

Nom de point	Type de données	Adresse de point	Nom de station
point_alm	Bit	N15:0/0	SLC_1

- Le point gère au total 256 bits/alarmes consécutifs (0 à 255), N:15:0/0 à N15:15/15.

L'adresse du bit de déclenchement d'alarmes est le nombre qui se trouve dans le champ Valeur/Bit ajouté à celui de l'adresse du point de déclenchement.

Adresse du point de déclenchement + Valeur/Bit = Adresse du bit de déclenchement

Par exemple, l'alarme A se déclenche lorsque N15:1/0 passe de 0 à 1.

16 + N15:0/0 = N15:1/0

L'alarme B se déclenche lorsque N15:0/2 passe de 0 à 1.

2 + N15:0/0 = N15:0/2

Pour le déclenchement sur **LSBit**, l'alarme se déclenche lorsque le bit de déclenchement passe de 0 à 1 alors que tous les autres bits entre l'adresse du point de déclenchement et celle du bit de déclenchement sont à zéro.

L'adresse du point de déclenchement étant N7:12/4 et la Valeur/Bit 6, une alarme est déclenchée lorsque N7:12/10 passe de 0 à 1. C'est le seul bit configuré entre N7:12/4 et N7:12/10.

Si N7:12/4 est à 1 lorsque N7:12/10 passe à 1, l'alarme correspondant à N7:12/10 ne sera déclenchée qu'après la remise à 0 de N7:12/4.

II.4.6. Graphiques :

PanelBuilder32 fournit des outils graphiques pour créer des lignes, des formes et des dessins. Ces outils sont accessibles depuis le menu **Objets>Graphiques** ou depuis la **boîte à outils.** Contrairement aux objets de contrôle et d'affichage, les graphiques peuvent se chevaucher sur la vue.

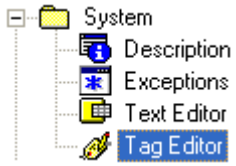
II.5. Editeur de points :

II.5.1. Définition :

L'éditeur de points est utilisé pour entrer des points dans l'application. Les points associent des objets à des adresses d'une base de données. Chaque point définit une adresse et des attributs de données que le terminal PanelView utilise pour lire ou écrire des données. Par exemple, un point de lecture est attribué à un bargraphe. Le terminal lit les données à l'adresse du point de lecture pour remplir le bargraphe.

Pour ouvrir l'éditeur de points :

- Cliquer deux fois sur l'icône **Base de données de points** dans le dossier **Système**
- Ou cliquer sur **Outils>Editeur de points**



II.5.2. Fiche de configuration de l'éditeur de points :

L'éditeur de points utilise des fiches de configuration similaires aux tableurs Excel pour l'entrée et l'édition de points. L'onglet de la fiche de configuration identifie le protocole pour lequel les points sont créés. Certains protocoles supportant plusieurs options, tels que

DeviceNet ou ControlNet, disposeront de plusieurs onglets. L'éditeur de points possède ses propres barres de menus et barres d'outils qui sont activées lorsqu'il est ouvert.

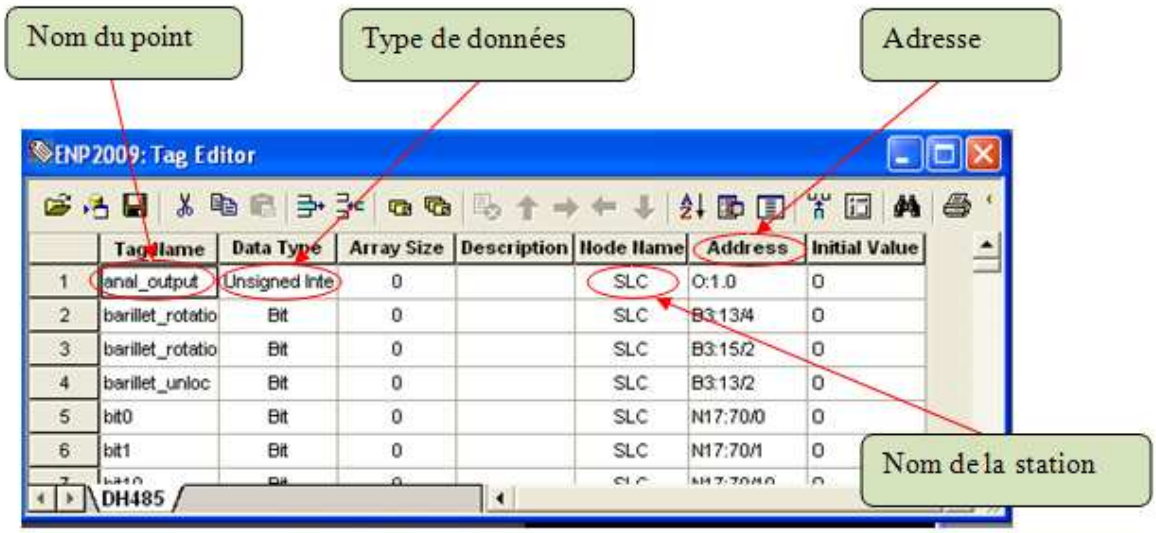


Figure IV.12 : Fenêtre de vue Editeur de points.

II.6. Configuration de l'application :

II.6.1. Configuration de la communication :

Le terminal PanelView communique avec les automates et les appareils décentralisés sur différents réseaux, tels que DH-485, DH+, RIO, DF1, DeviceNet, ControlNet, Modbus et autres protocoles.

Chaque terminal prend en charge un protocole de communication spécifique sélectionné au moment de la création de l'application. Avant de charger une application, il faut configurer le terminal et l'automate pour permettre la communication.

Pour configurer la communication :

- Cliquer deux fois sur l'icône **Configuration de la communication** dans le dossier **Configuration de l'application** de la fenêtre d'application, ou
- Sélectionner **Application>Configuration** puis cliquer sur le bouton **Configuration de la communication**.

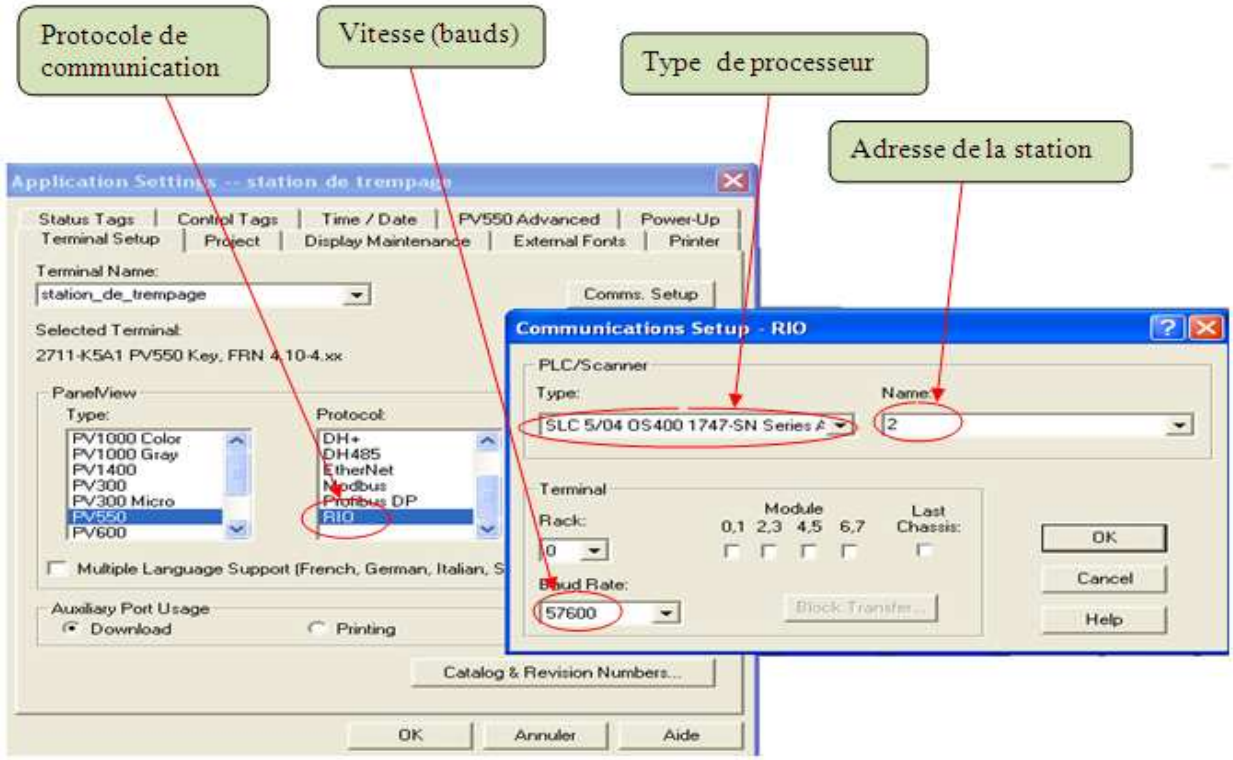


Figure IV.13 : Fenêtre de configuration de la communication.

Les informations d'automate (nom de station, adresse de station, type de station) sont généralement entrées de la même façon pour la plupart des protocoles.

II.7. Transfert d'applications :

II.7.1. Comment transférer des applications :

On peut transférer des applications entre un ordinateur et un terminal PanelView ou une carte mémoire en utilisant :

- Les commandes **Fichier>Charger** ou **Fichier>Transférer** de PanelBuilder32
- L'utilitaire de transfert de fichiers hors PanelBuilder32

Si on utilise l'utilitaire de transfert de fichiers, on doit d'abord enregistrer l'application comme fichier .PVA avec la commande **Fichier>Enregistrer sous**.
L'utilitaire de transfert de fichiers ne transfère que les fichiers .PVA du terminal PanelView.

Le driver RSLinx approprié doit être configuré et exécuté sur l'ordinateur. Pour transférer une application, sélectionner **Fichier>Transférer**.

II.7.2. Chargement d'une application dans un terminal :

Cette section explique comment charger une application depuis le port COM série de l'ordinateur dans le port RS-232/DF1 du terminal PanelView à l'aide d'une connexion point à point. Le chargement utilise le driver DF1 interne du port COM1- COM9 de l'ordinateur. On utilise le câble 2711-NC13 (connecteur 9 broches) pour la connexion point à point. On doit Vérifier les connexions de câble avant de lancer le chargement.

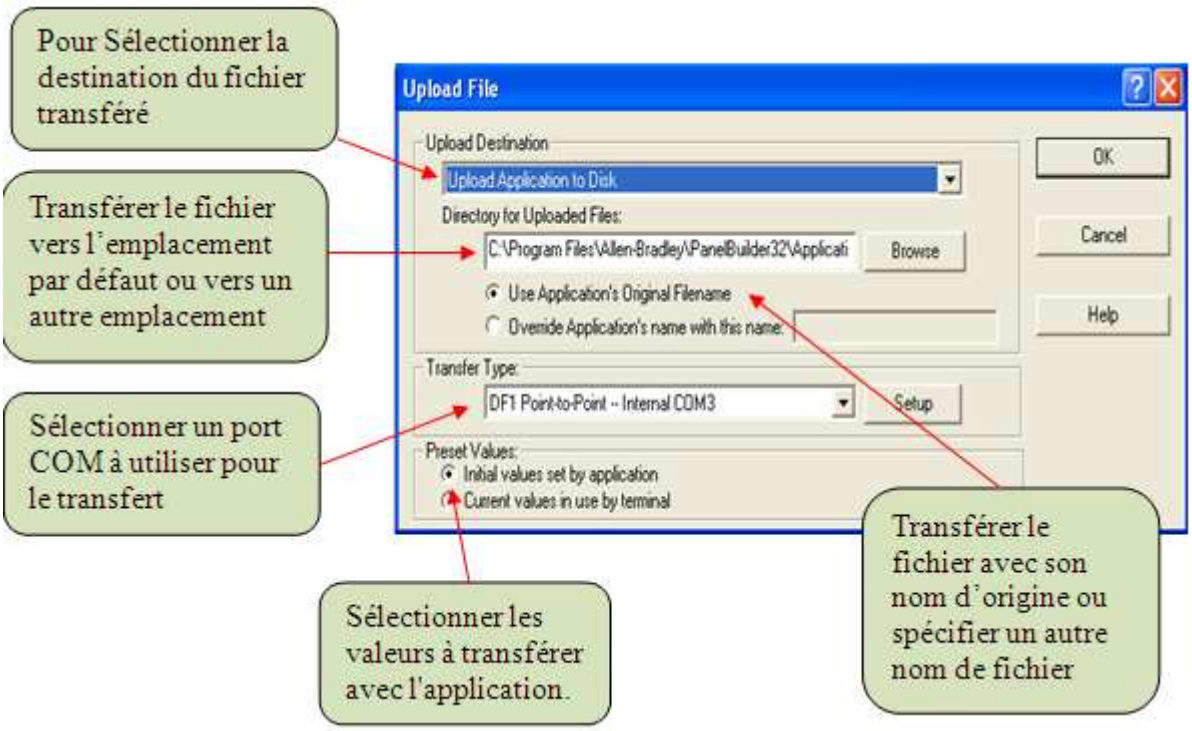


Figure IV.14 : Fenêtre de Chargement.

Quand le transfert est terminé, l'application s'ouvre sous PanelBuilder32 à moins qu'on n'ait transféré l'application dans un fichier sur disque.

II.7.3. Utilisation de l'utilitaire de transfert de fichiers :

On utilise l'utilitaire de transfert de fichiers pour transférer des applications entre un terminal et un ordinateur sous Windows sans ouvrir PanelBuilder32.

Pour utiliser l'utilitaire de transfert de fichiers, on sélectionne Programmes>PanelBuilder32>WinPFT depuis le menu Démarrer.

Figure IV-15 : Fenêtre de l'utilitaire de transfert de fichiers.

Charge un fichier .PVA dans un terminal ou sur une carte mémoire.

II.7.4. Vérification d'une application :

Pour vérifier une application on clique sur Vérification dans la boîte de dialogue de l'utilitaire de transfert de fichiers pour comparer l'application à charger à une application déjà chargée dans un terminal PanelView ou sur un fichier disque. Il est bon de s'assurer que des différences existent avant de charger une application dans un terminal. On a aussi l'option de charger l'application si l'application de référence est différente de celle déjà chargée dans le terminal.

Conclusion :

La description précédente de l'outil PanelBuilder32 montre les différentes étapes de la création d'une application et explique comment charger et exécuter une application dans un terminal PanelView. Cette étude est une démarche inévitable pour pouvoir mener à bien la deuxième phase de notre travail qui est l'affichage des défauts de l'assembleuse pose à plat après les avoir détectés par le logiciel de programmation RSLogix 500.

Chapitre V

Solution proposée pour le diagnostic et l'affichage des défauts

Introduction :

Après avoir étudié et décrit le langage de programmation RSLogix 500 et le Progiciel de conception et de configuration d'interface PanelBuilder32 dans les chapitres précédents, notre travail consiste à élaborer un programme de détection de défauts pouvant se produire sur la machine d'assemblage pose à plat (PAP), et de créer une application de panneau de commande servant à les afficher sur le terminal PanelView 550.

Pour cela, nous avons procédé d'abord à établir une liste de défauts traitant le maximum de cas possibles. Ensuite, nous avons utilisé l'outil grafcet pour modéliser les solutions apportées qui sont après traduites en langage LADDER.

Afin de valider la solution proposée, des simulations sur PC ont été effectuées avec le logiciel RSLogix 500.

I. Classification des défauts :

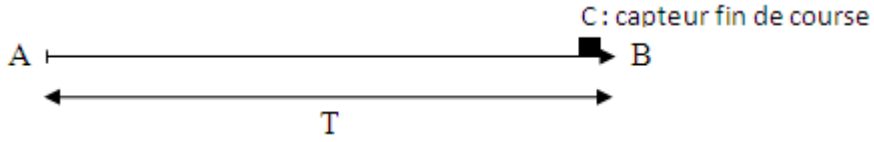
Nous avons classé les défauts en trois types **:**

I.1. Défauts matériels :

Comme tous les mouvements des éléments de la machine se font par des vérins à l'exception du gonflage tambour, nous avons classé les défauts matériels comme suit :

a. Défauts liés aux vérins (Plusieurs) :

Pour qu'un vérin passe d'un état A (entrée ou sortie) vers un état B (sortie ou entrée), cela prend une certaine durée de temps T. Si la durée T est achevée et que le capteur en position B n'est pas activé, alors le vérin n'a pas effectué sa course ou qu'il est quelque part entre les points A et B, ceci est un défaut matériel.



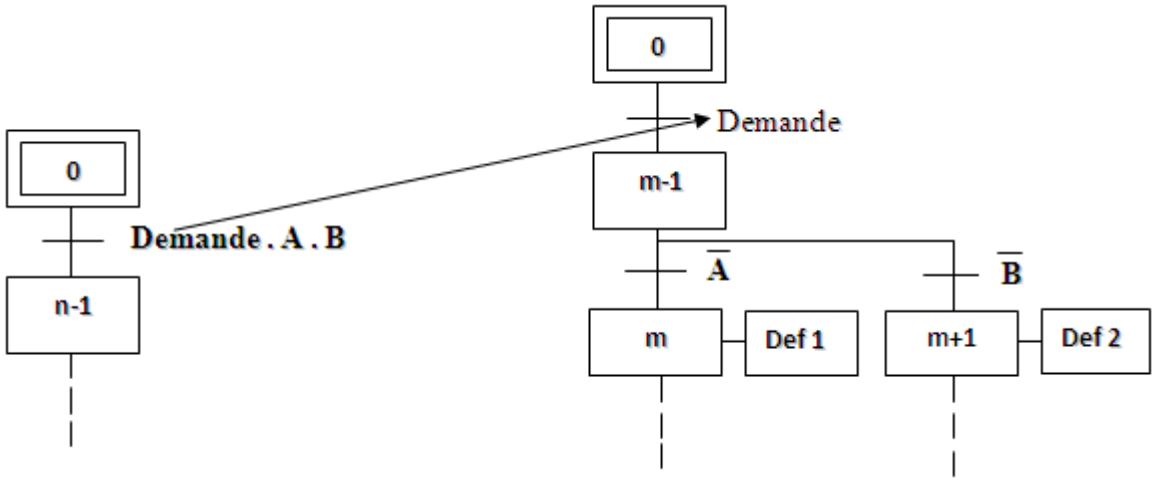
Bit défaut = $\overline{C}$. FIN T.

b. Défaut lié au tambour :

Idem pour le tambour qui passe par gonflage d'un diamètre minimum à un diamètre maximum.

I.2. Défauts liés au franchissement des transitions :

Afin d'éviter les chevauchements entre les différents éléments de la machine pouvant provoquer des dégâts, chaque mouvement de l'un est conditionné par la présence des autres sur les positions ne provoquant pas ces accidents. Ces positions sont données par des capteurs associés à chaque élément.



Grafcet fonctionnement **Grafcet Défauts**

Etant l'étape n-1 celle actionnant l'entrée ou la sortie d'un vérin V1, et A, B les états des capteurs associés aux vérins V2 et V3, pour que le vérin V1 effectue sa course sans qu'il y'ait interaction avec V2 et V3, il faut que V2 soit à la position A=1 et V3 à la position B=1.

Dans le cas contraire, A=0 et/ou B=0, le vérin V1 n'effectue pas sa course, donc défaut.

Pour savoir lequel des vérins V2 ou V3 n'étant pas à la bonne position, on appelle le grafcet des défauts par le bit de demande d'entrée ou de sortie du vérin V1 comme c'est illustré sur la figure ci-dessus.

Une fois le grafcet des défauts est appelé (l'étape m-1 activée) :

Si A=0 $\Longrightarrow$ $\overline{A}$=1 $\Longrightarrow$ L'étape m est activée, donc défaut absence capteur A (Def 1).

Si B=0 $\Longrightarrow$ $\overline{B}$=1 $\Longrightarrow$ L'étape m+1 est activée, donc défaut absence capteur B (Def 2).

I.3. Défauts alimentation et défaut tringle :

a. Défauts alimentation :

Chaque alimentation (24 V, 110 V, alimentation four…) passe par un disjoncteur. La sortie de chaque disjoncteur va vers un système relais dont le contact à fermer ou à ouvrir fait passer une tension de 24 V comme entrée vers l'automate. Cette information est exploitée pour s'assurer de la présence de l'alimentation comme suit :

Entrée = 24 V (1 logique) $\Longrightarrow$ Remise à zéro d'un bit défaut.

Entrée = 0 V (0 logique) $\Longrightarrow$ Mise à 1 du bit défaut.

b. Défaut tringle :

Chaque carcasse doit contenir deux tringles (Cerceaux), chacune sur un côté. On ne peut faire passer les tringles que par le côté droit du tambour, donc il faut s'assurer de la présence d'une tringle sur le côté gauche avant même de commencer un cycle de production, parce qu'une fois le tambour est gonflé on ne peut plus la faire passer, sachant que la pose des tringles se fait sur un diamètre maxi du tambour (Tambour gonflé).

La présence de la tringle est donnée par un capteur à contact mécanique, cette information est une entrée automate que nous avons exploitée dans le programme de détection des défauts de la façon suivante :

Entrée capteur tringle = 0 V (0 logique) $\Longrightarrow$ Mise à 1 d'un bit défaut.

Entrée capteur tringle = 24 V (1 logique) $\Longrightarrow$ Remise à 0 du bit défaut.

II. Liste des défauts :

Au total, nous avons recensé les défauts suivants :

Défaut montée tête de cheval.

Défaut descente tête de cheval.

Défaut montée piano.

Défaut descente piano.

Défaut montée galet.

Défaut descente galet.

Défaut gonflage tambour.

Défaut lance table.

Défaut montée retroussage.

Défaut descente retroussage.

Défaut matériels

Défauts sécurité galet.

Défauts sécurité piano.

Défauts sécurité retroussage.

Défaut sécurité tourelle.

Défaut sécurité lance table.

Défauts liés aux transitions

Défaut absence 24 V.

Défaut absence 110 V.

Défaut alimentation four.

Défaut télécommande.

Défaut tringle.

Défauts alimentation et défaut tringle

III. Solution proposée :

III.1 Modélisation par Grafcet :

La modélisation par l'outil grafcet décrit au chapitre III est la méthode adoptée pour faire des programmes avec le langage de programmation RSLogix 500.

III.1.1. Grafcets pour défauts matériels :

Sur les grafcets ci-dessous, la réceptivité associée à la transition de sortie de l'étape initiale est un bit d'action d'entrée ou de sortie d'un vérin ou gonflage du tambour, si ce bit est activé, l'étape initiale étant à 1, l'étape suivante (m) sera activée, enclenchant une temporisation nécessaire pour effectuer la course.

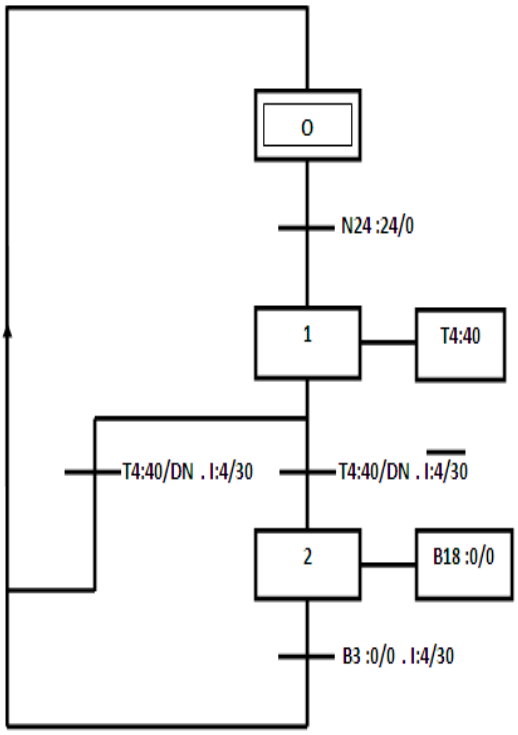
Si le délai de cette temporisation est écoulé et que le capteur de fin de course n'émet pas de signal (Réceptivités associées à la transition de sortie de l'étape m), alors l'étape m+1 est activée, donc le bit défaut est mis à 1, sinon on revient directement à l'étape initiale dans le cas où le capteur émet un signal.

. Acquittement du défaut (B3:0/0) :

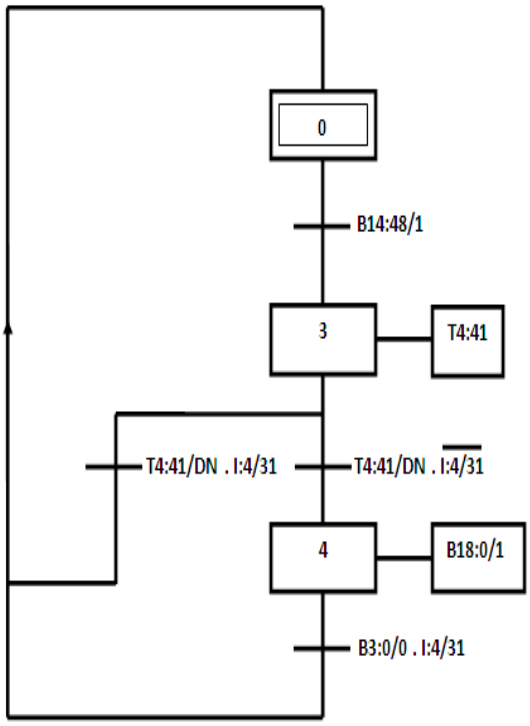
Afin d'acquitter le défaut, nous avons programmé un bouton d'acquittement sur le pupitre opérateur.

L'acquittement ne peut se faire qu'après réparation de la panne et forçage du vérin ou tambour sur la position qi'il faut (Capteur activé), cela remet le grafcet à l'étape initiale.

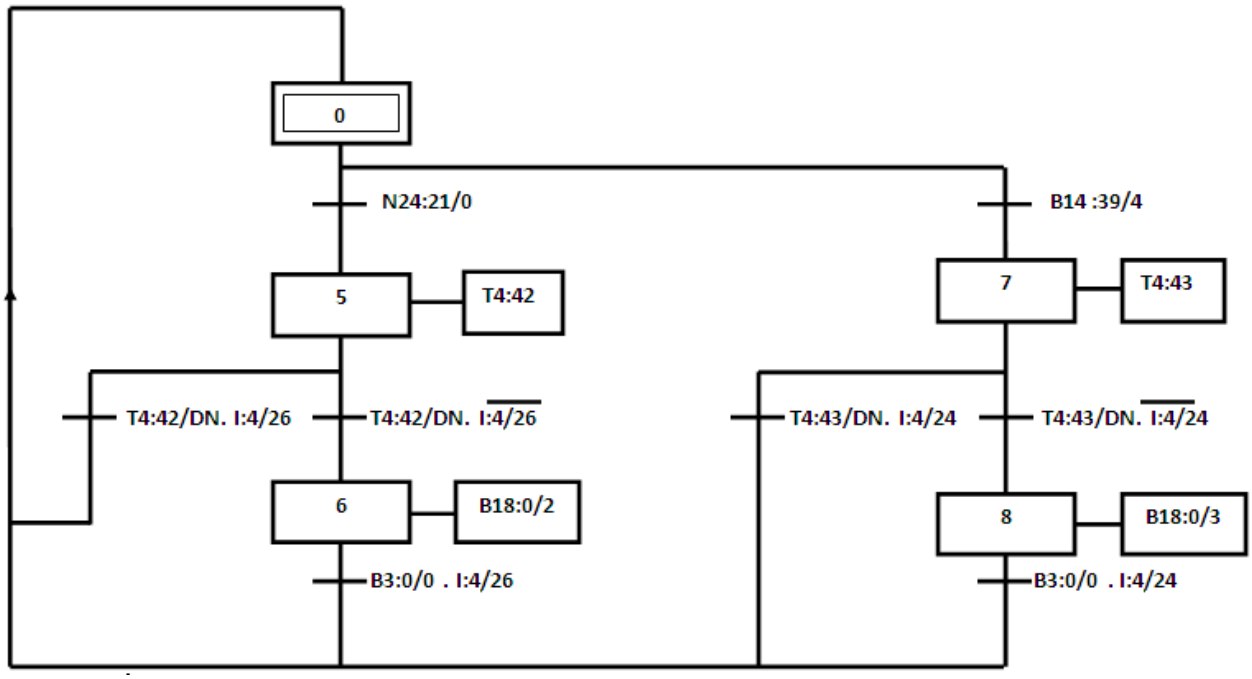
Défaut montée tête de cheval :



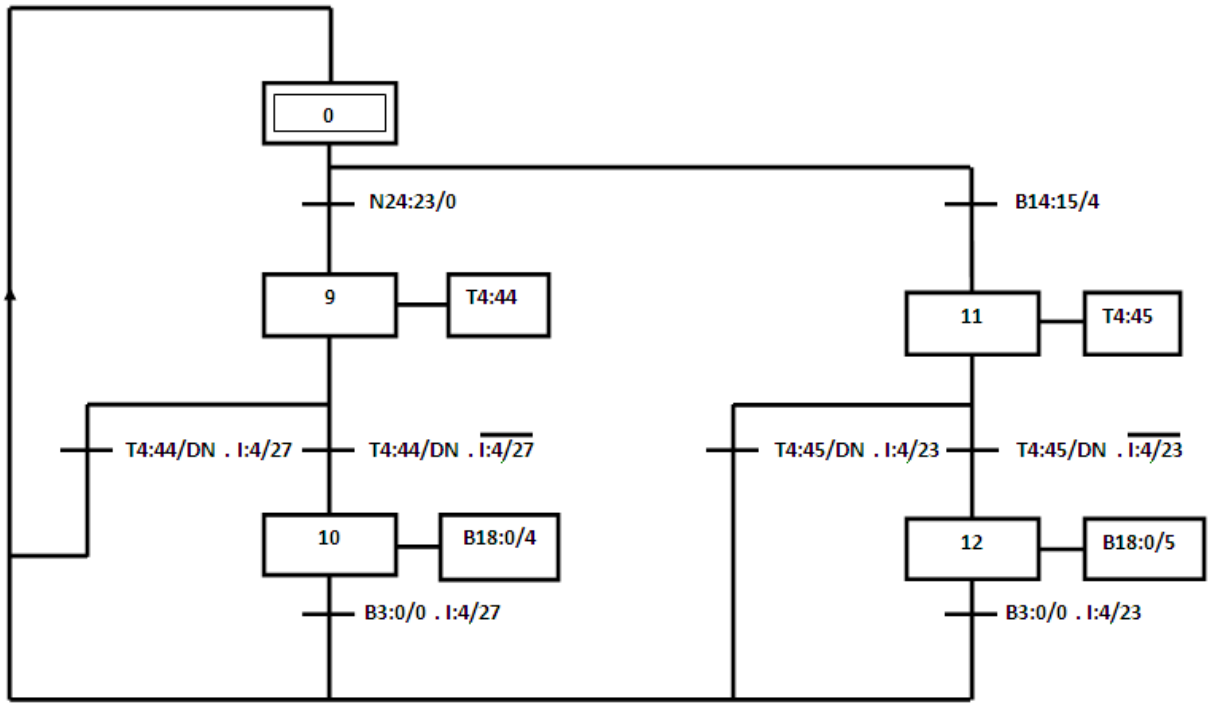
Défaut descente tête de cheval :



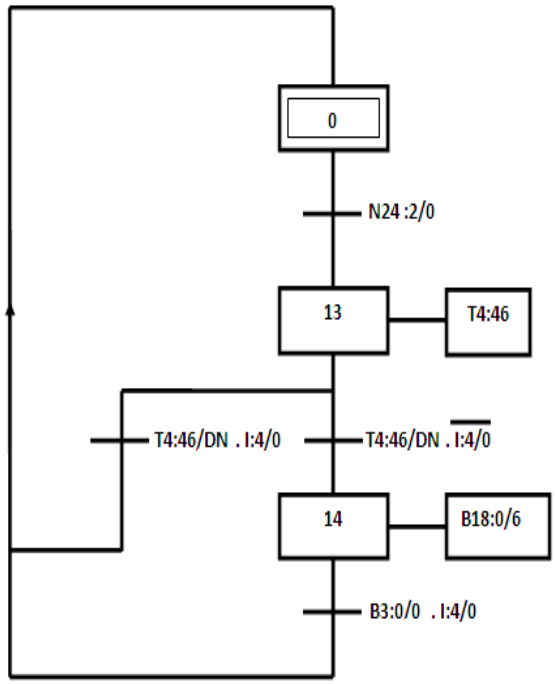
Défaut montée et descente piano :



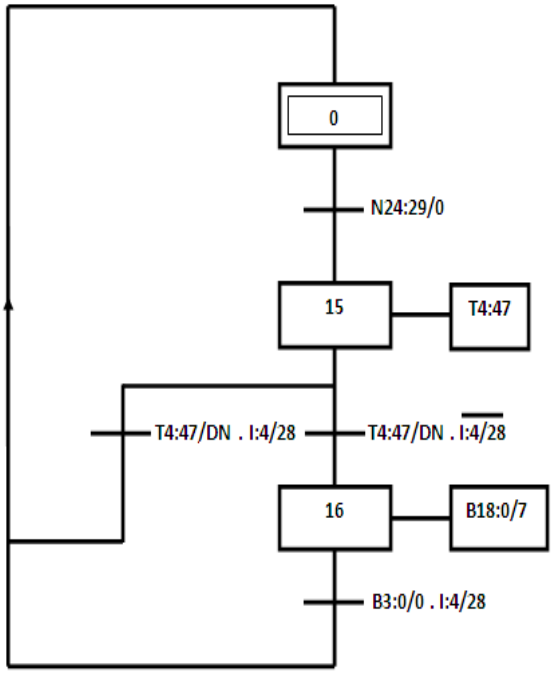
Défaut montée et descente galet :



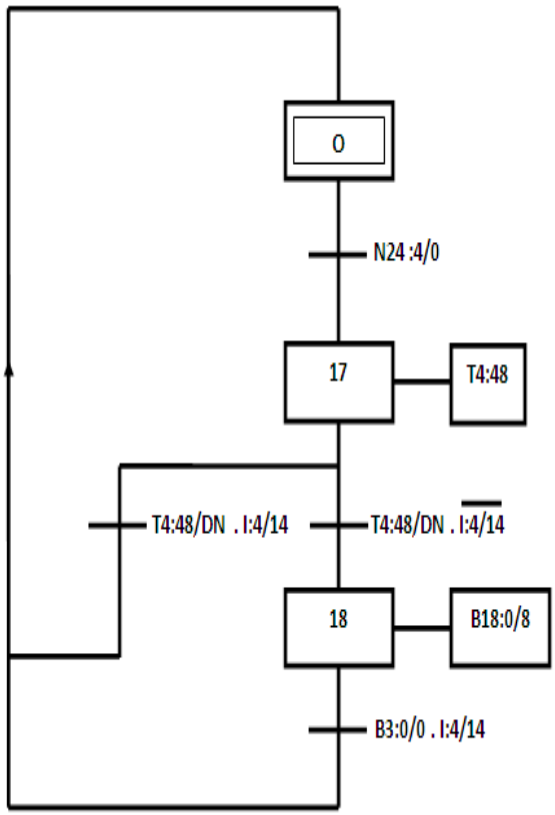
Défaut gonflage tambour :



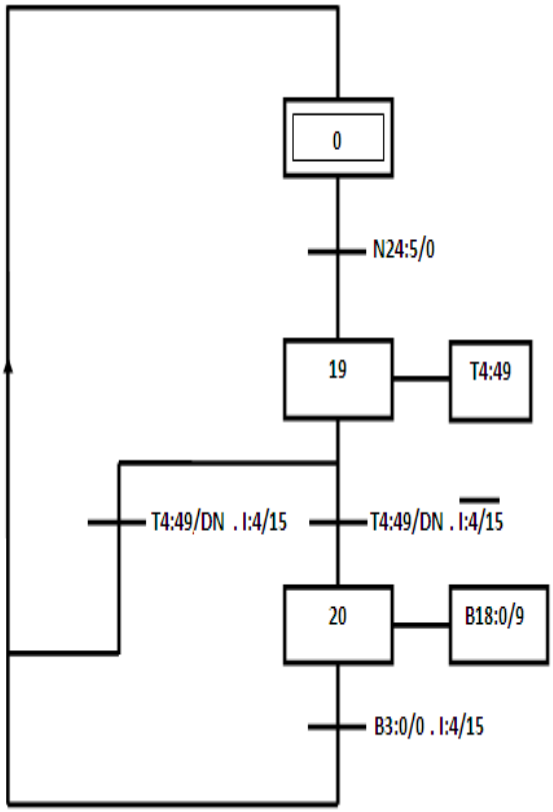
Défaut lance table :



Défaut montée retroussage :



Défaut descente retroussage :

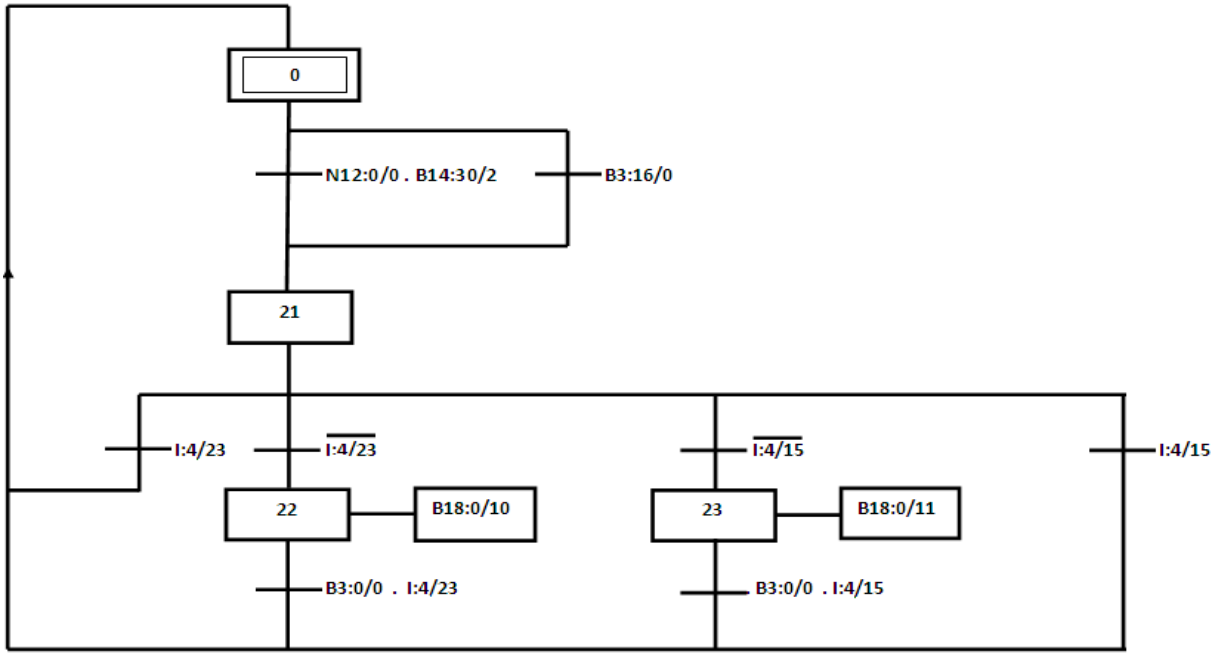


III.1.2. Grafcets pour défauts liés aux transitions :

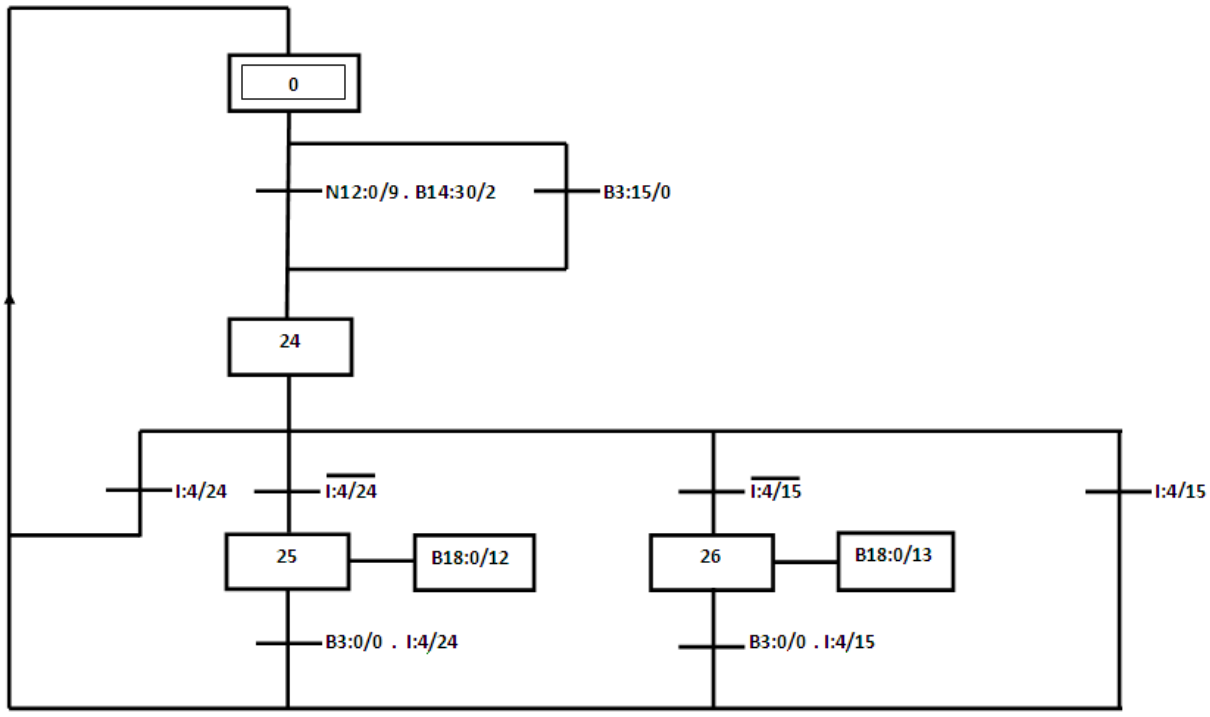
Le passage de l'étape initiale à l'étape suivante (m) se fait soit par le bit de demande de sortie ou entrée d'un vérin V programmé dans le cycle automatique ou par le bit de commande manuelle. Les réceptivités associées aux transitions de sortie de l'étape m sont des états des capteurs des éléments qui doivent être sur des positions programmées afin d'éviter tout interaction avec le vérin V.

Si le bit d'état de l'un de ces capteurs n'est pas à 1, alors activation de l'étape suivante : m+1 ou m+2 auxquelles nous avons associé chacune l'action d'activation d'un bit de défaut.

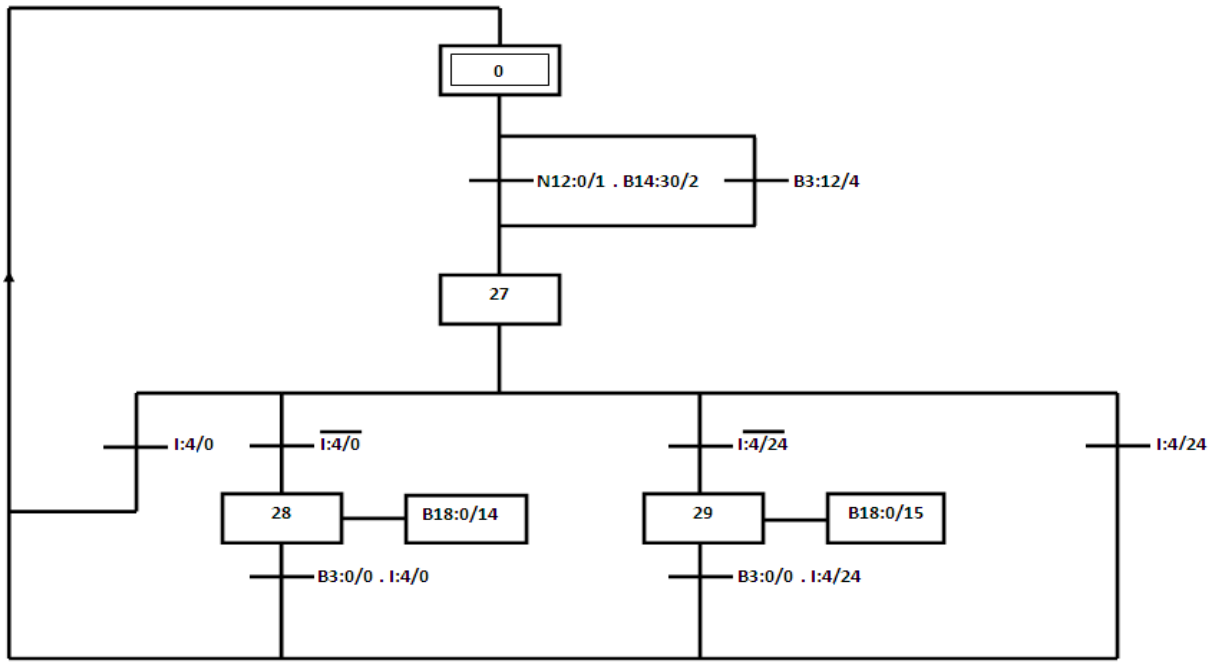
Défaut sécurité galet :



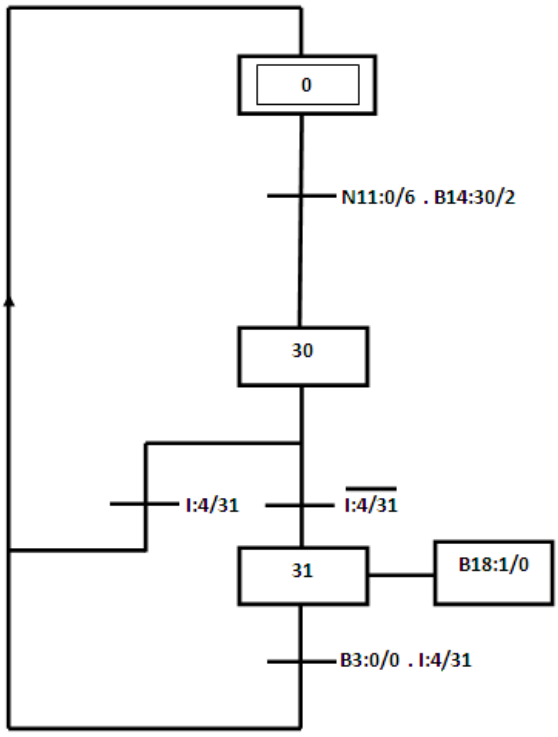
Défaut sécurité piano :



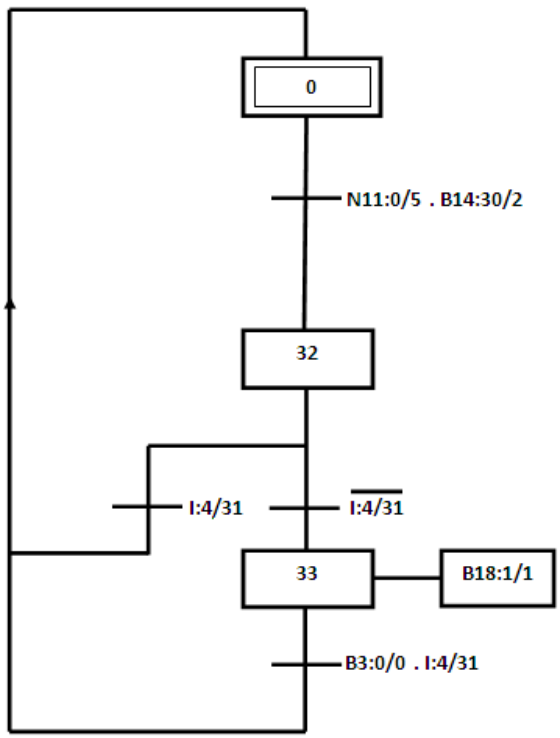
Défaut sécurité retroussage :



Défaut sécurité tourelle :



Défaut sécurité lance table :



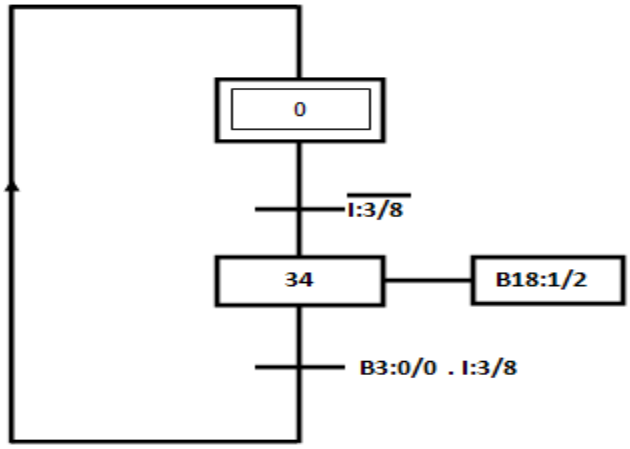
III.1.3. Grafcets pour défauts alimentation et défaut tringle :

a. Défauts alimentations :

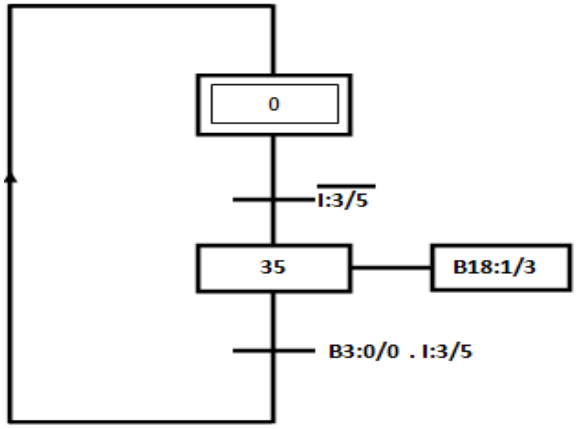
La réceptivité associée à la transition de sortie de l'étape initiale est un bit d'entrée automate informant si une alimentation est présente ou non.

Dans le cas d'absence d'une alimentation, la transition est franchie pour activer l'étape suivante qui à son tour active un bit défaut.

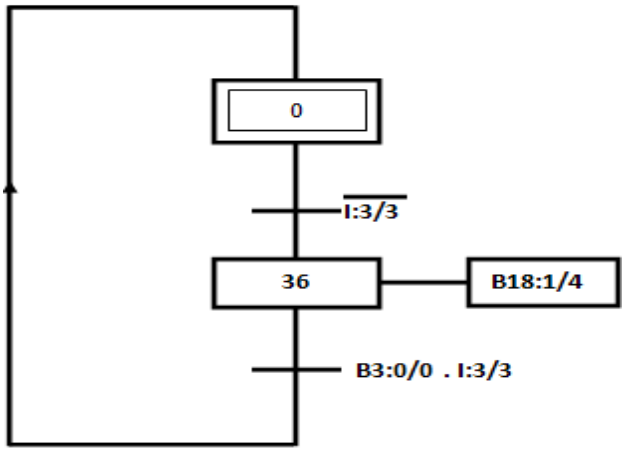
Défaut absence 24



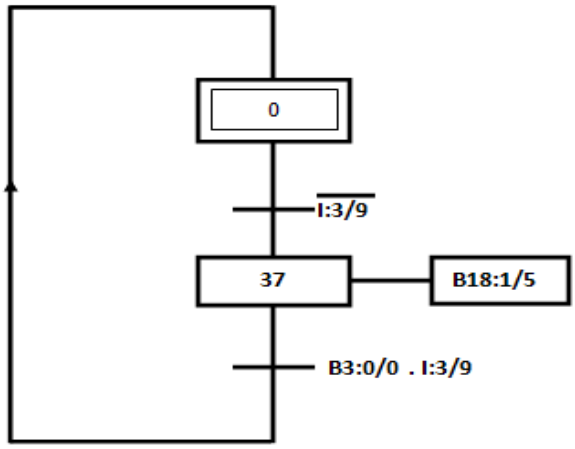
Défaut absence 110



Défaut alimentation four :



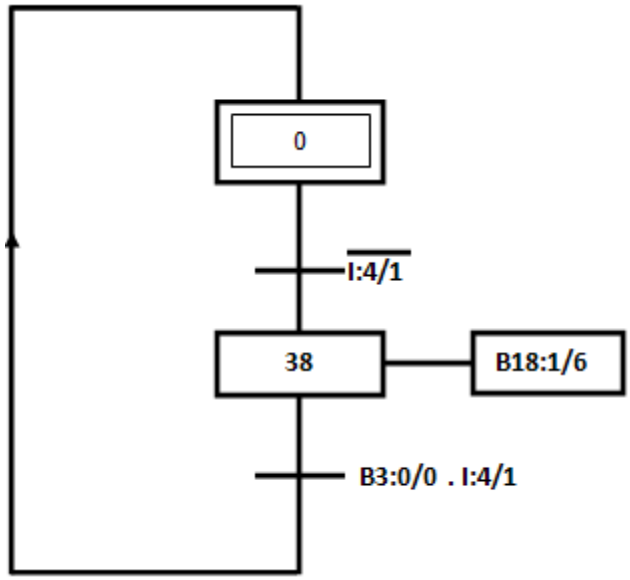
Défaut télécommande :



b. Défaut tringle :

A l'absence d'une tringle I:4/1 = 0 $\Longrightarrow$ $\overline{\text{I:4/1}}$ = 1 : la transition est franchie, l'étape 38 est activée, donc le bit défaut B18:1/6 est mis à 1**.**

Défaut absence tringle :



- **Désignation des bits et temporisations utilisés dans les grafcets des défauts :**

Bit	Désignation
N24:24/0	Bit action montée tête cheval
N24:21/0	Bit action montée piano
N24:23/0	Bit action montée galet
N24:2/0	Bit action gonflage tambour
N24:29/0	Bit action lance table
N24:4/0	Bit action montée retroussage
N24:5/0	Bit action descente retroussage
N12:0/0	Bit demande montée galet
N12:0/9	Bit demande montée piano
N12:0/1	Bit demande montée retroussage
N11:0/6	Bit demande déverrouillage tourelle
N11:0/5	Bit demande lance table

Bit	Désignation
I:4/30	Capteur tête cheval position haute 42SQ
I:4/31	Capteur tête cheval position basse
I:4/26	Capteur piano position haute
I:4/24	Capteur piano position basse
I:4/27	Capteur galet position haute
I:4/23	Capteur galet position basse
I:4/0	Capteur tambour gonflé
I:4/28	Fin de course lance table
I:4/14	Capteur retroussage position haute
I:4/15	Capteur retroussage position basse
I:3/8	Alimentation 24 V
I:3/5	Alimentation 110 V
I:3/3	Alimentation four
I:3/9	Télécommande
I:4/1	Capteur Tringle

Bit	Désignation
B14:48/1	Bit étape descente tête cheval
B14:39/4	Bit étape descente piano
B14:15/4	Bit étape descente galet
B14:30/2	Bit étape 2 opération séquentielle
B3:16/0	Bit demande montée galet sur PV 550 écran 9
B3:15/0	Bit demande montée piano sur PV 550 écran 9
B3:12/4	Bit demande montée retroussage sur PV 550 écran 8
B3:0/0	Bit d'acquittement de défauts sur PV 550 écran Alarm Banner

Bit	Désignation
B18:0/0	Bit défaut absence capteur tête cheval position haute 34SP
B18:0/1	Bit défaut absence capteur tête cheval position basse 42SQ33
B18:0/2	Bit défaut absence capteur piano position haute 42SQ4
B18:0/3	Bit défaut absence capteur piano position basse 42SQ5
B18:0/4	Bit défaut absence capteur galet position haute 42SQ36
B18:0/5	Bit défaut absence capteur galet position basse 42SQ37
B18:0/6	Bit défaut absence capteur tambour gonflé 2SP
B18:0/7	Bit défaut absence capteur lance table 42SQ21
B18:0/8	Bit défaut absence capteur retroussage position haute 42SQ1
B18:0/9	Bit défaut absence capteur retroussage position basse 42SQ2
B18:0/10	Bit défaut absence capteur galet position basse 42SQ37
B18:0/11	Bit défaut absence capteur retroussage position basse 42SQ2
B18:0/12	Bit défaut absence capteur piano position basse 42SQ5
B18:0/13	Bit défaut absence capteur retroussage position basse 42SQ2
B18:0/14	Bit défaut absence capteur tambour gonflé 2SP
B18:0/15	Bit défaut absence capteur piano position basse 42SQ5

B18:1/0	Bit défaut absence capteur tête cheval position basse 42SQ33
B18:1/1	Bit défaut absence capteur tête cheval position basse 42SQ33
B18:1/2	Bit défaut absence 24 V
B18:1/3	Bit défaut absence 110 V
B18:1/4	Bit défaut absence alimentation four
B18:1/5	Bit défaut télécommande
B18:1/6	Bit défaut absence tringle

Tempo	Désignation
T4:40	Temporisation montée tête cheval
T4:41	Temporisation descente tête cheval
T4:42	Temporisation montée piano
T4:43	Temporisation descente piano
T4:44	Temporisation montée galet
T4:45	Temporisation descente galet
T4:46	Temporisation gonflage tambour
T4:47	Temporisation lance table
T4:48	Temporisation montée retroussage
T4:49	Temporisation descente retroussage

IV. Programmme de détection des défauts par RSLogix 500 :

IV.1 Méthode de conversion d'un grafcet en un programme LADDER :

La méthode permet de traduire tout grafcet, aussi complexe soit-il, à partir des instructions de base suivantes :

Le principe de cette méthode repose sur :

L'exploitation de deux fichiers binaires par entité logicielle :

BXX4 : fichier des étapes ACTIVES grafcet.

BXX5 : fichier des étapes VALIDANT LES TRANSITIONS (évolution grafcet).

BXX4 représente les fichiers d'étapes actives utilisés par la partie commune et /ou les différentes entités logicielles (par exemple, B14, B64, B114...). Idem pour BXX5.

A l'étape N seront donc associés deux bits de même rang : BXX4/N et BXX5/N.

- Le fichier BXX4 traduit l'état des grafcets.

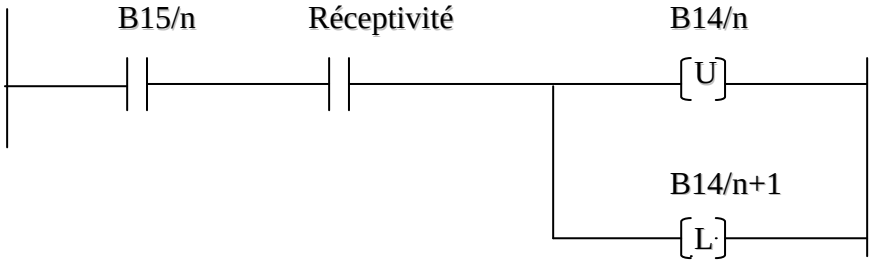
- Le fichier BXX5 permet de limiter l'évolution des grafcets à une étape au plus, à chaque scrutation.

La méthode proposée consiste en une traduction mécanique des principes de franchissement des transitions et d'évolution des étapes actives. Il s'agit donc, transition par transition, de traduire en schéma échelle la phrase suivante :

Si étape (s) amont ACTIVE (S) et réceptivité vraie

Alors désactiver étape (s) amont et activer étape (s) aval.

La règle précédente, dans le cas de structures linéaires, devient :



Le réseau ci- dessus ne fait que préparer le franchissement de la transition et l'évolution du Grafcet. Il faut faire, en amont de ce réseau, une copie des fichiers des étapes actives (B14, B64,…) vers les fichiers des étapes validant les transitions (B15, B65 …). Cette copie s'effectuera, pour chaque entité, dans le fichier contenant le graphe directeur.

La traduction d'un grafcet comprendra, dans le même fichier, la description complète de la structure à l'aide de la méthode ci-dessus, suivie de la description des actions sur étapes.

IV.2 Le programme :

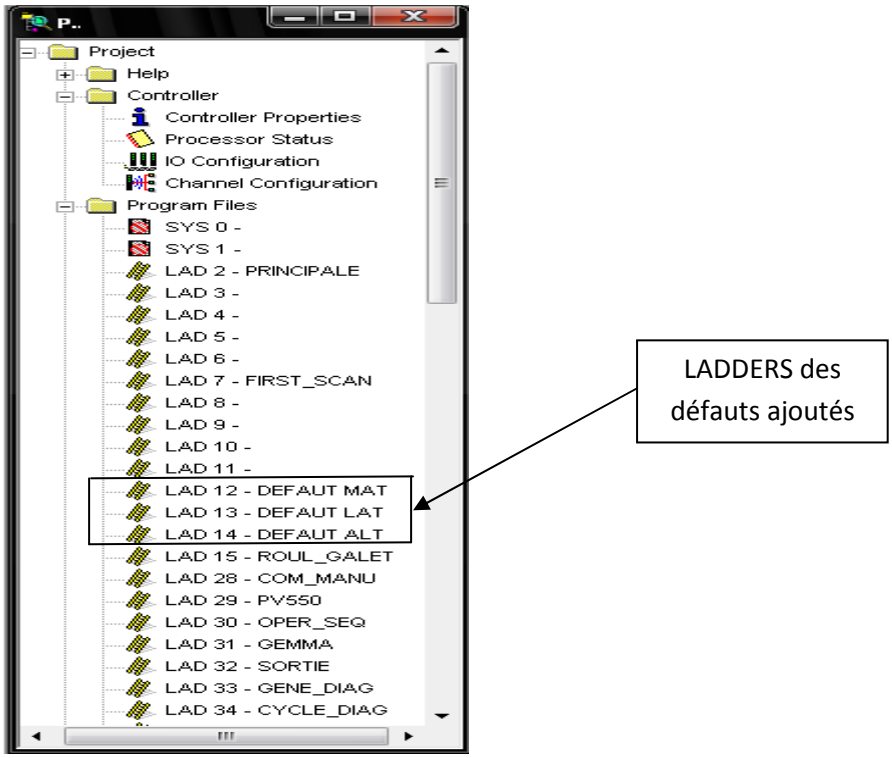
IV.2.1. Création des fichiers LADDER pour les défauts :

Nous avons créé un fichier LADDER pour chaque type de défauts :

Défauts matériels ⟶ **LAD 12-DEFAUT MAT.**

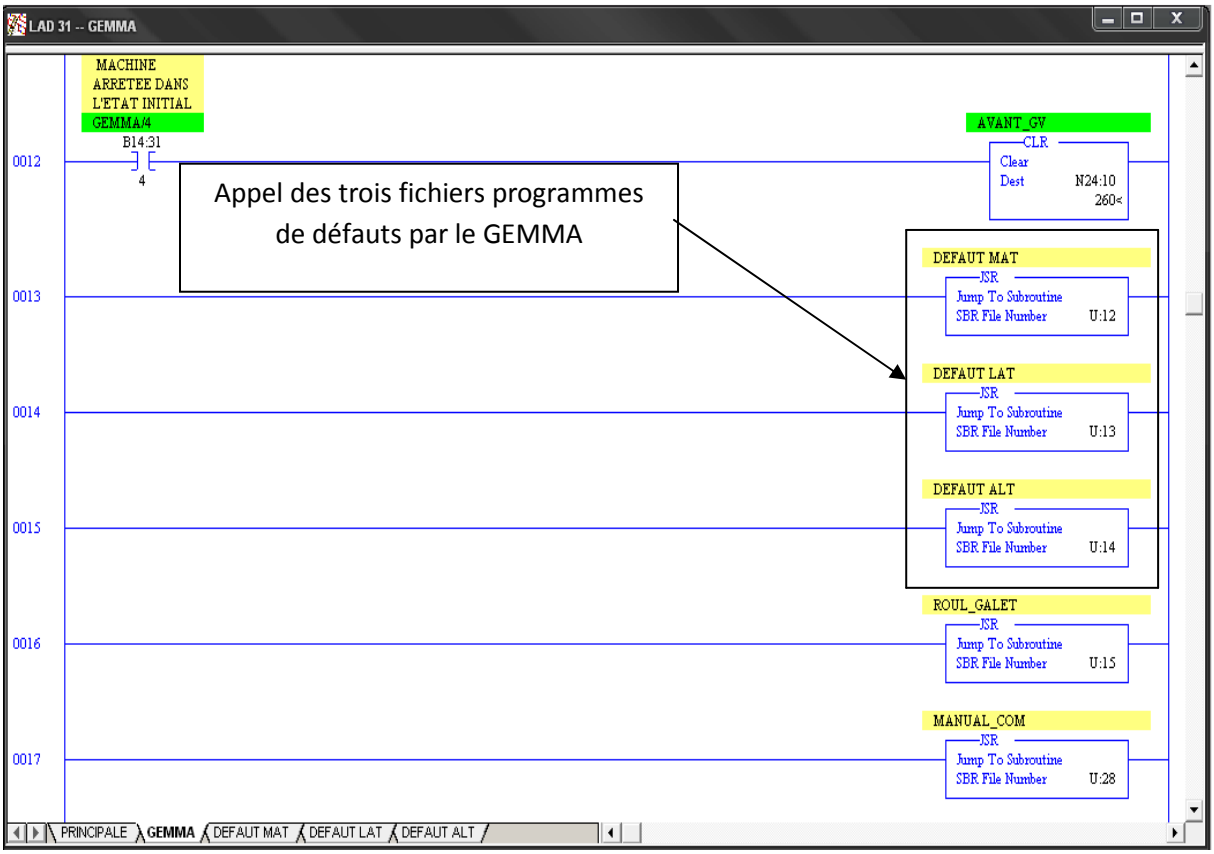
Défauts liés aux transitions ⟶ **LAD 13-DEFAUT LAT.**

Défauts alimentation et défaut tringle ⟶ **LAD 14-DEFAUT ALT.**



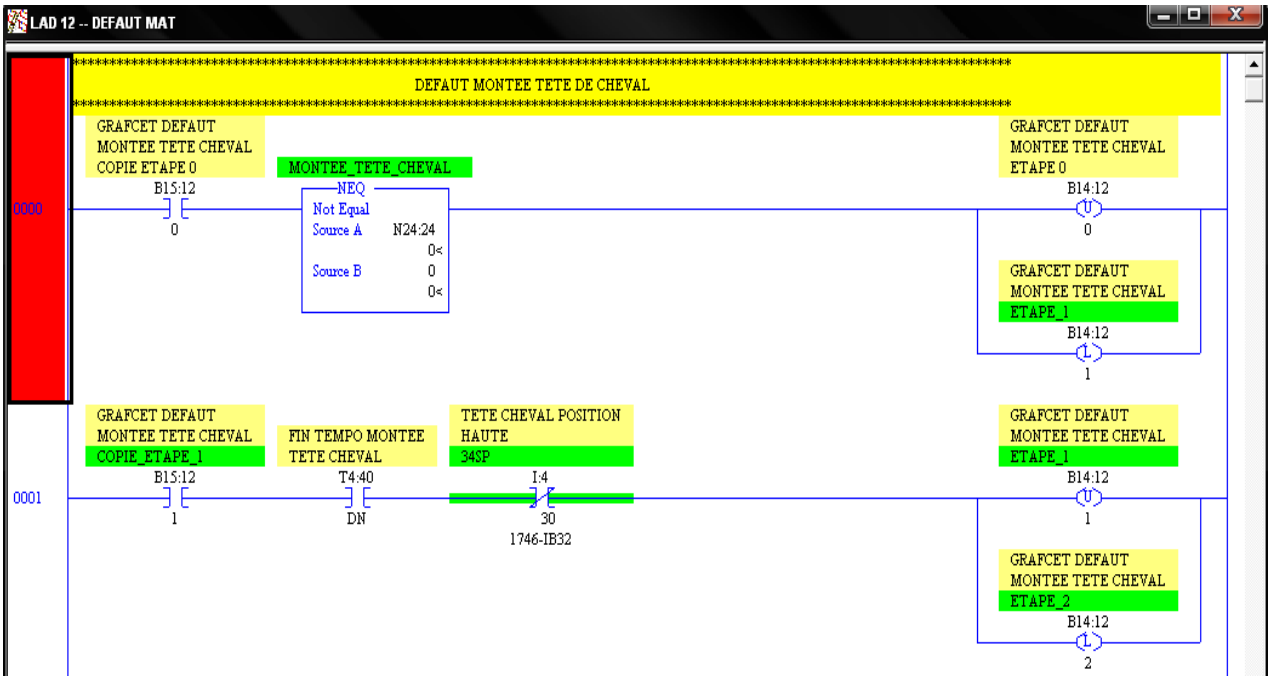
IV.2.2. Appel des LADDER défauts par le LADDER GEMMA :

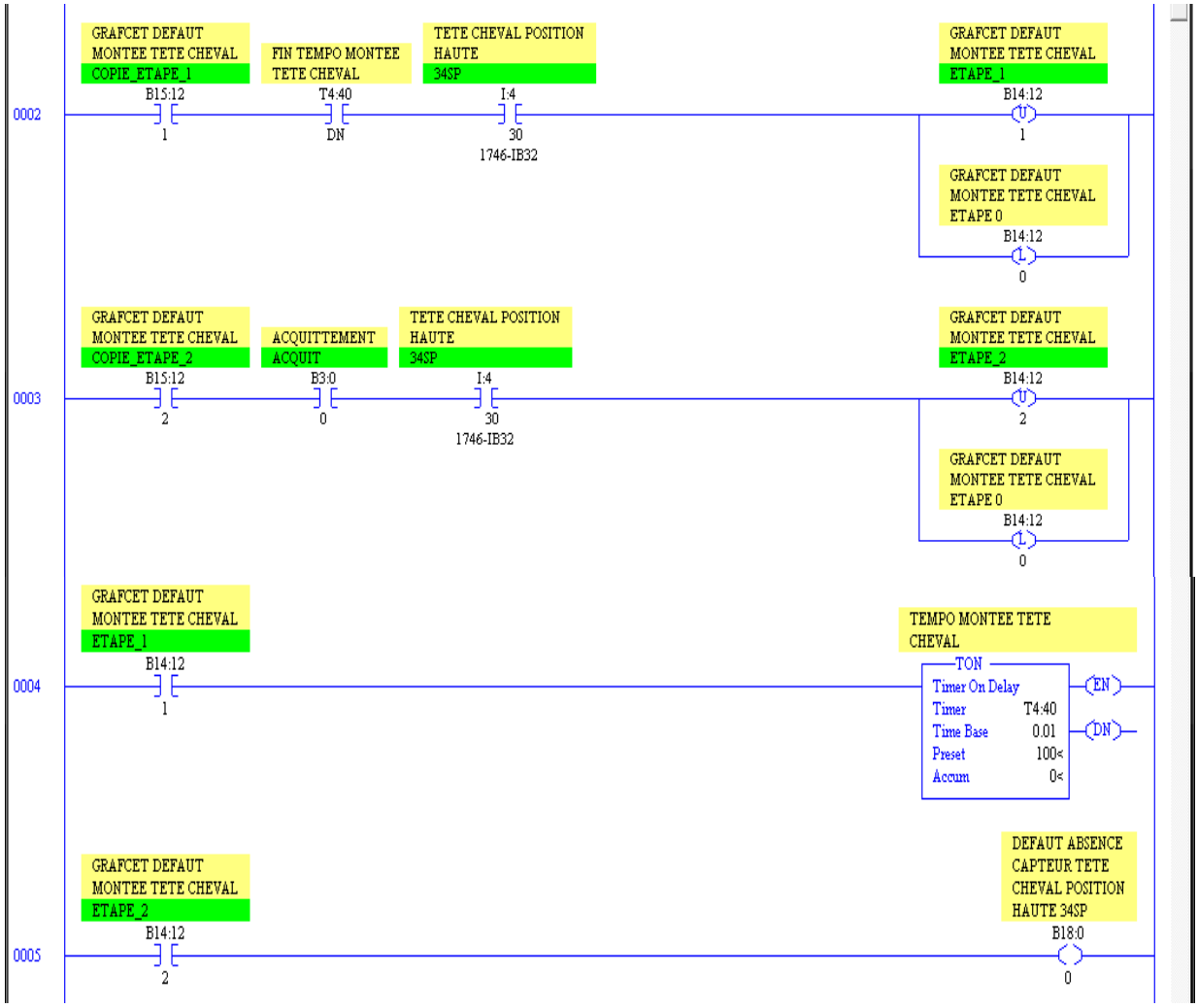
Les LADDER des défauts sont appelés par le GEMMA à l'aide de l'instruction JSR (Jump to SubRoutine) comme le montre la figure suivante :



IV.2. 3. Exemples de programmes :

a. **Programme défauts matériels** (Défaut montée tête de cheval du fichier LAD 12-DEFAUT MAT) :

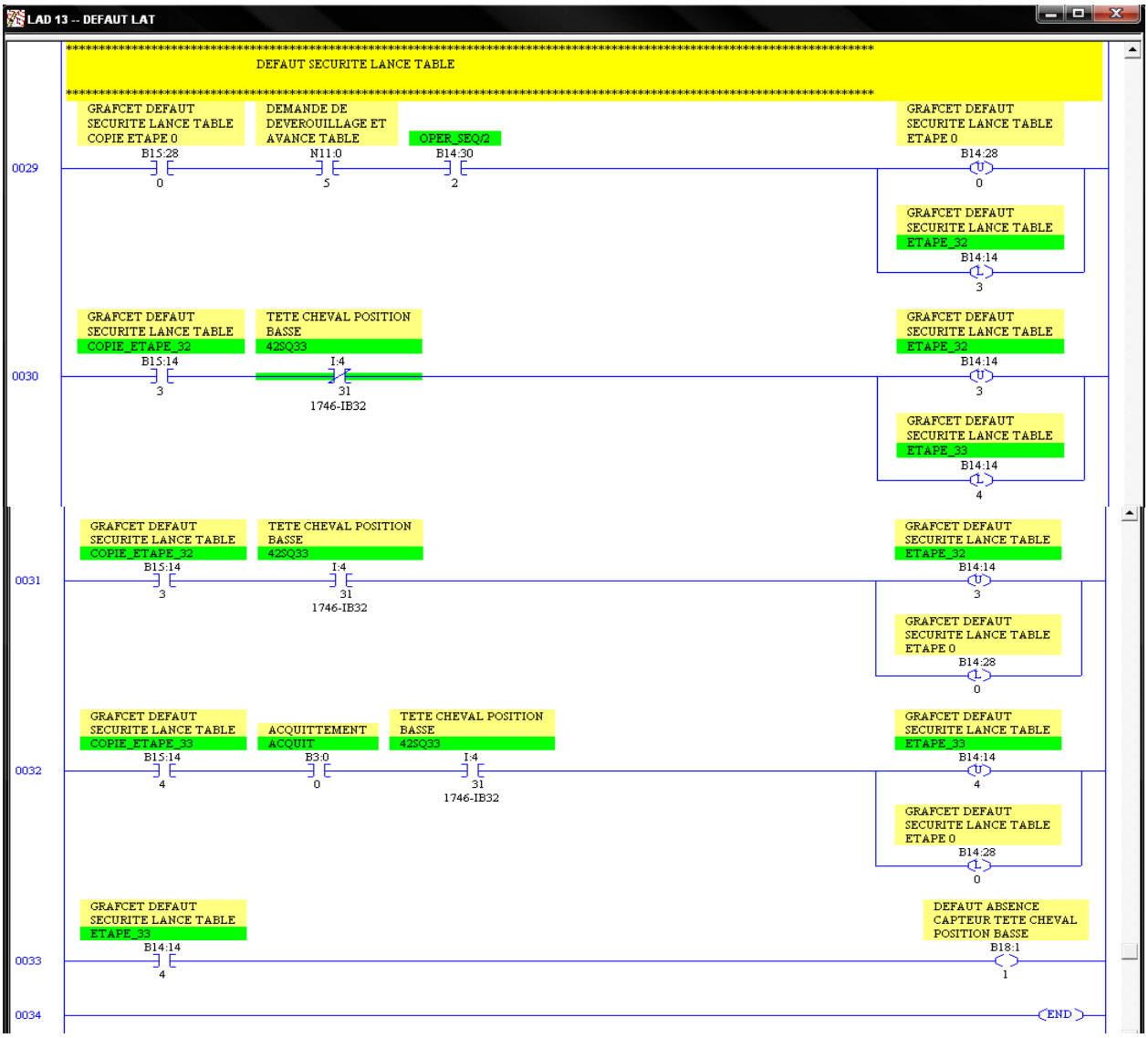




Le BIT associé à l'action de montée de la tête cheval est le N24:24/0, ce même bit est exploité dans le programme de vérification de défaut pour enclencher la temporisation (T4:40) nécessaire au vérin pour atteindre le fin de course, alors à la fin de cette temporisation, si le capteur de fin de course (tête de cheval position haute : I4:30) n'est pas activé, cela met à 1 le bit défaut B18:0/0 qui sera utilisé sur PanelBuilder 32 pour l'affichage.

Une fois la panne est réglée, donc l'état du capteur I:4/30 = 1, on acquitte le défaut par activation du bit B3:0/0 à partir d'un bouton programmé sur le pupitre opérateur.

b. Programme défauts liés aux transitions (Défaut sécurité lance table du fichier LAD 13-DEFAUT LAT) :

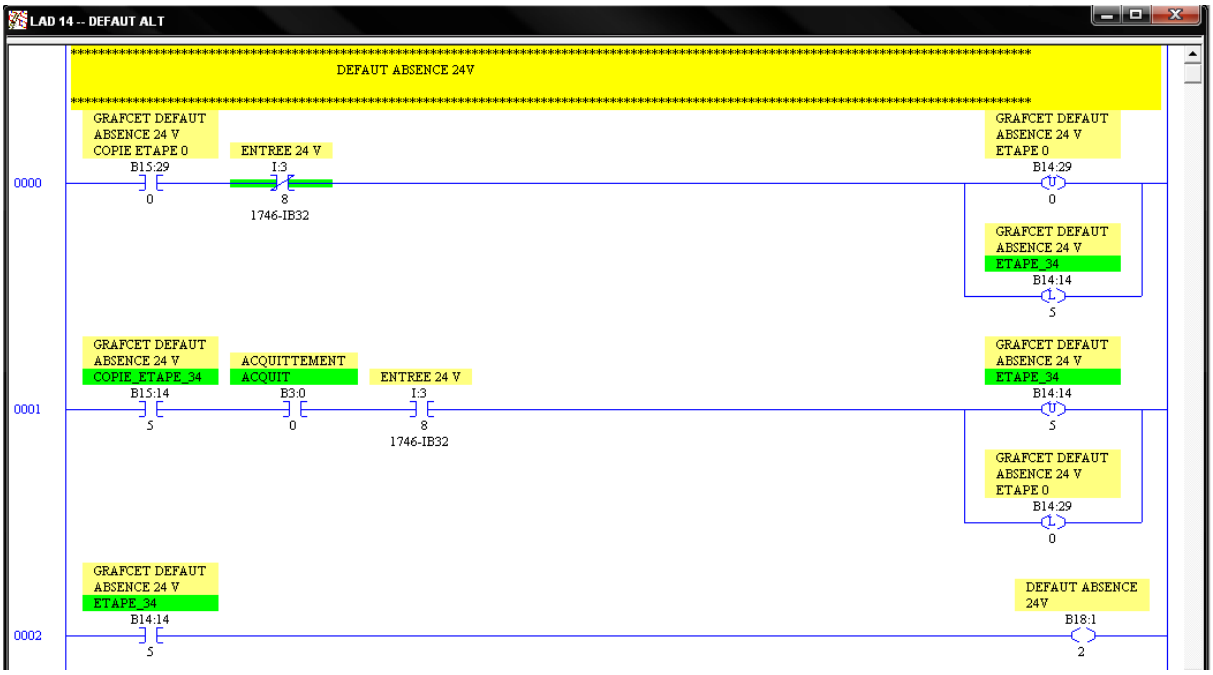


Le bit associé à la demande lance table est le N11:0/5, il est également utilisé dans le programme des défauts pour activer une étape d'attente B14:14/3. Une fois cette étape est activée on vérifie l'état du capteur tête cheval position basse qui doit être activé.

Alors si l'état de ce capteur est à 0, l'étape suivante B14:14/4 sera activée pour activer à son tour le bit défaut B18:1/1.

Sinon on revient directement à l'étape initiale B14:28/0.

c. **Programme défauts alimentation** (Défaut absence 24 V du fichier LAD 14-DEFAUT ALT) :



Etant I:3/8 l'entrée informant sur la présence ou l'absence de l'alimentation 24 V, alors si I:3/8 = 0, l'étape B14:14/5 est activée, donc activation du bit défaut B18:1/2.

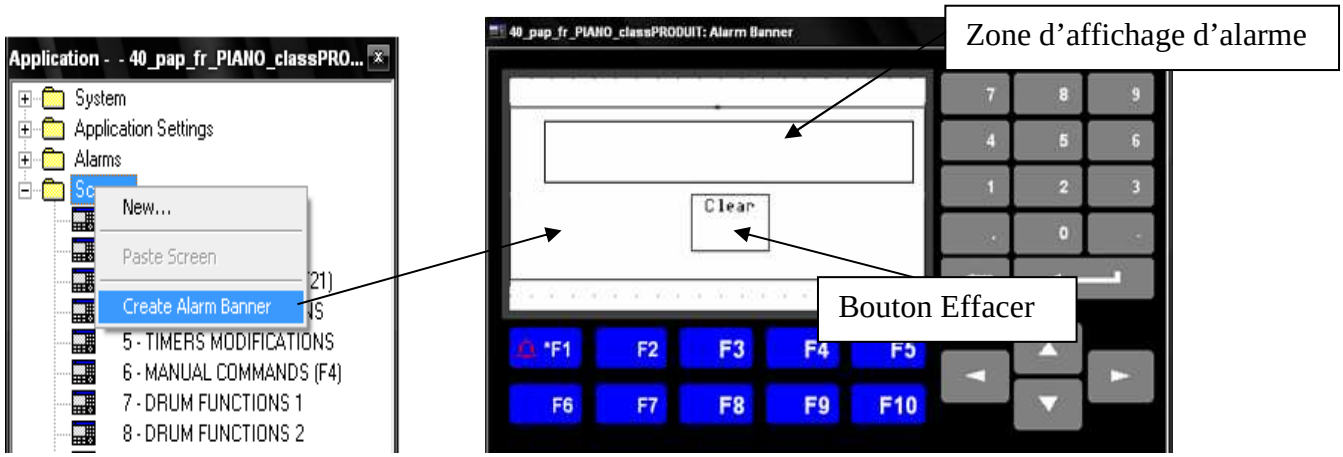
V. Affichage des défauts avec PanelBuilder 32 :

Après avoir fait le programme de détection des défauts sur RSLogix 500, il reste à les afficher sur le PV550 à l'aide du logiciel PanelBuilder 32. Pour cela nous avons utilisé la fonction alarmes.

Nous avons procédé de la manière suivante :

V.1. Création du bandeau d'alarmes :

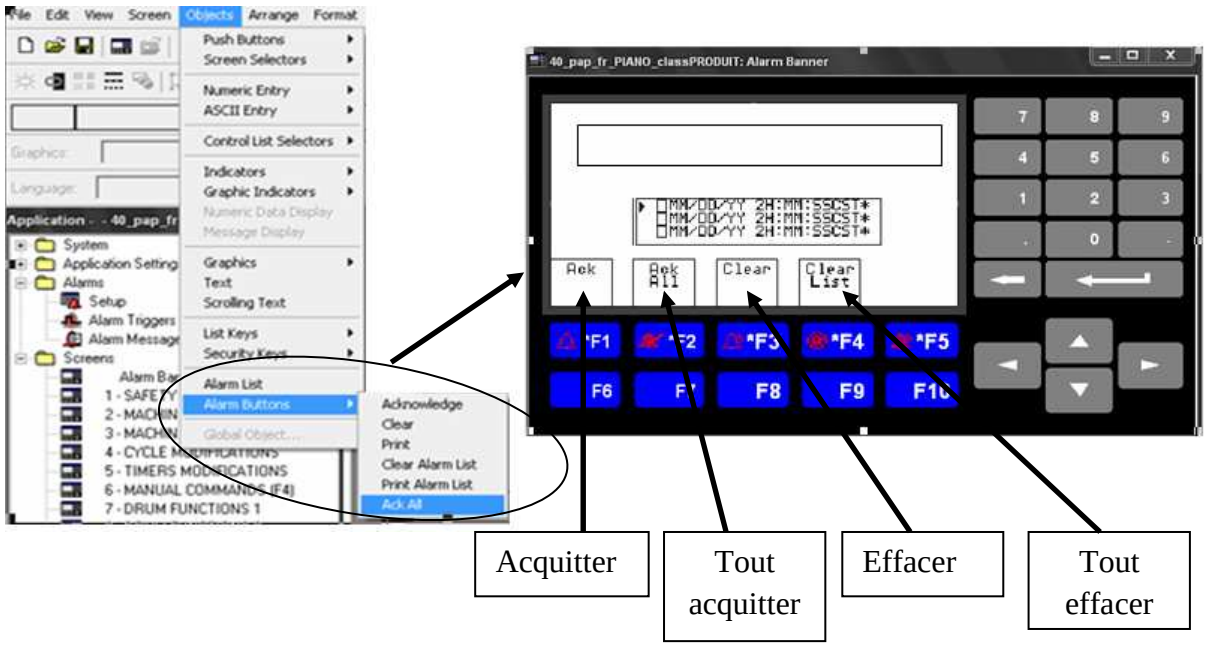
Cela se fait en cliquant avec le bouton droit sur le dossier ''Screens'' de la fenêtre d'application, et cliquer sur ''Creat Alarm Banner''.



La vue ‘’Alarm Banner’’ s’ouvre avec la zone d’affichage de l’alarme enclenchée et le bouton ‘’Clear’’ par défaut.

V.2. Insertion de la fonction liste d’alarmes et des boutons Acquitter, Tout acquitter, Effacer et Tout effacer :

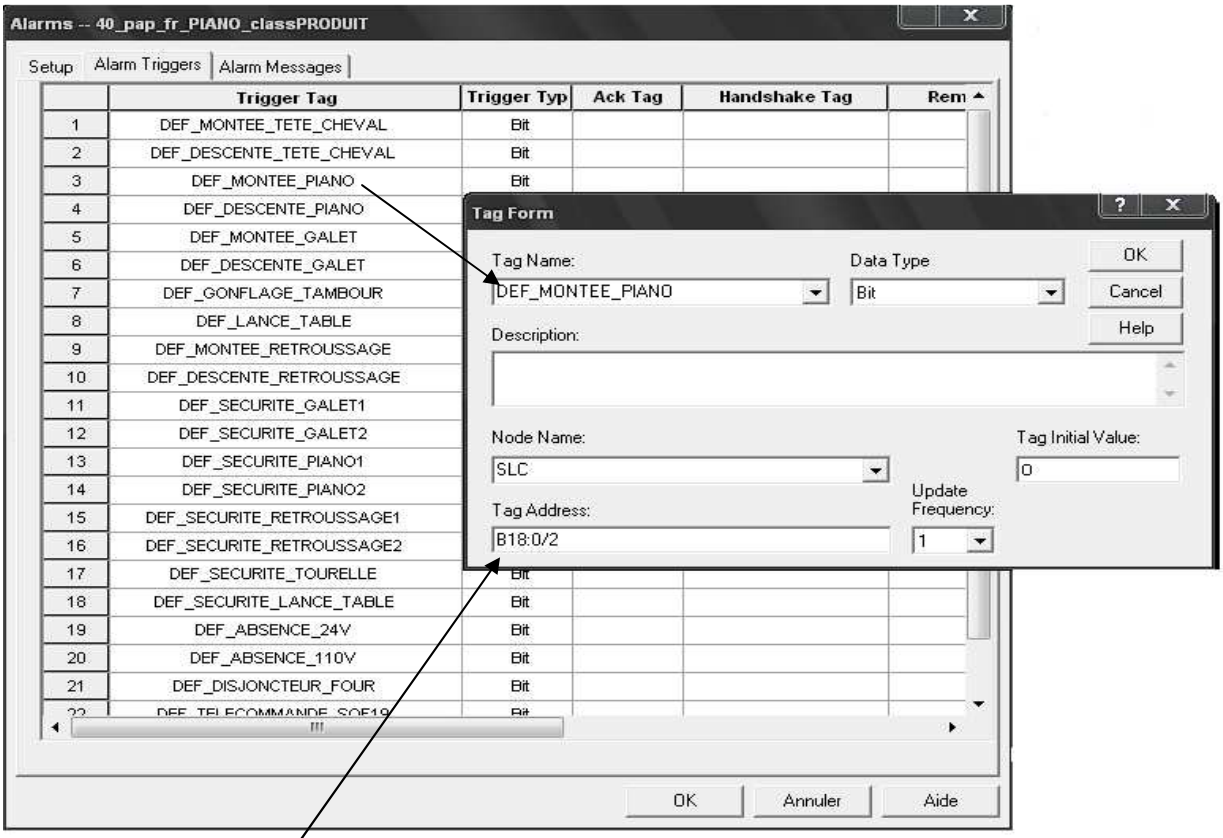
Pour se faire, on clique sur le menu ‘’Objects’’ de la barre de menus, et on prend les boutons désirés.



V.3. Configuration de la liste d'alarmes :

V.3.1. Définir les points et type de déclenchement d'alarmes :

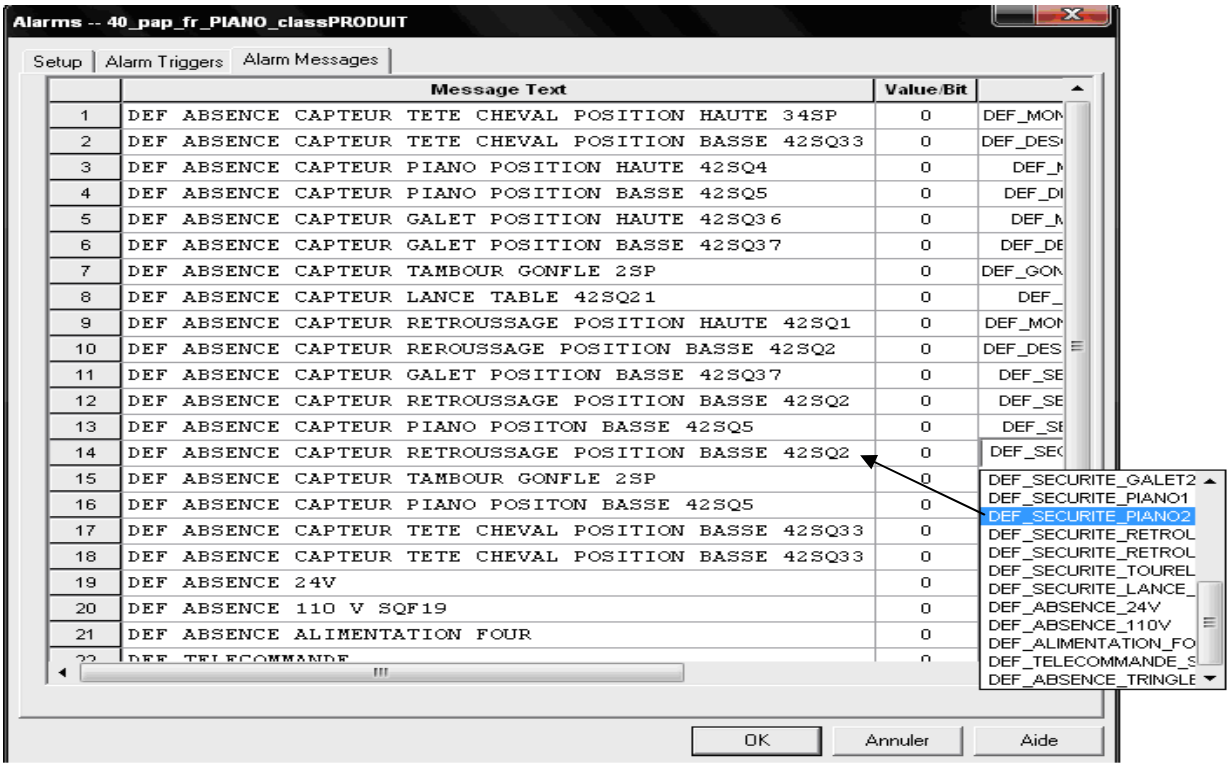
L'accès à cette fenêtre se fait par double-clique sur le lien ''Alarm Triggers'' propre à la liste d'alarmes sur la fenêtre d'application.



Point de déclenchement d'alarme

Cette figure montre comment créer les points de déclenchement pour les associer ensuite chacun à une alarme.

V.3.2. Affectation du point de déclenchement et du champ Value/Bit pour chaque alarme :



Cette étape consiste à faire entrer la liste d'alarmes et associer un point de déclenchement et la valeur ''Value/Bit'' pour chacune d'elles pour obtenir les bits de déclenchement d'alarmes qui sont les bits de défauts sur le programme RSLogix 500.

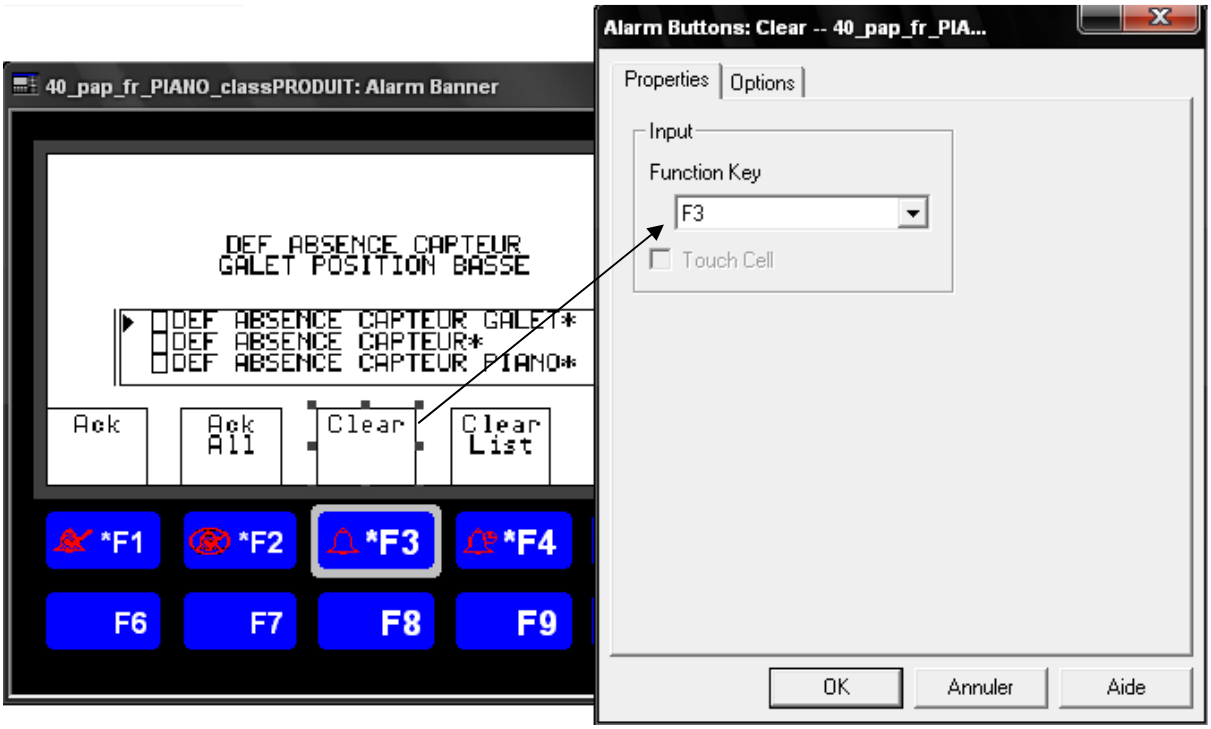
Comme c'est expliqué au chapitre IV :

Adresse du point de déclenchement + Valeur/Bit = Adresse du bit de déclenchement.

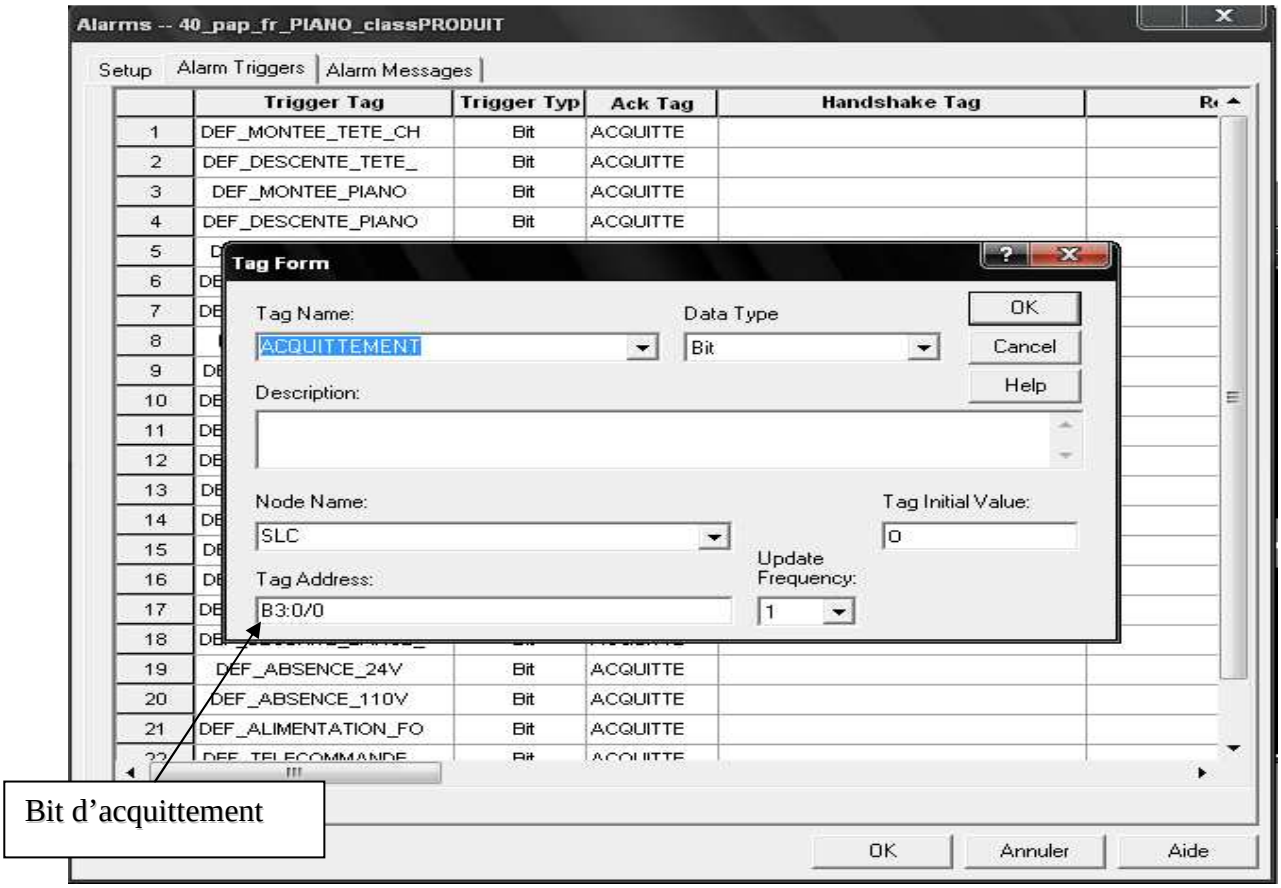
Dans notre cas, nous avons créé un point de déclenchement égal au bit de déclenchement pour chaque alarme, et donc le champ Valeur/Bit = 0 pour toutes les alarmes.

V.3.3. Configuration des boutons d'alarmes :

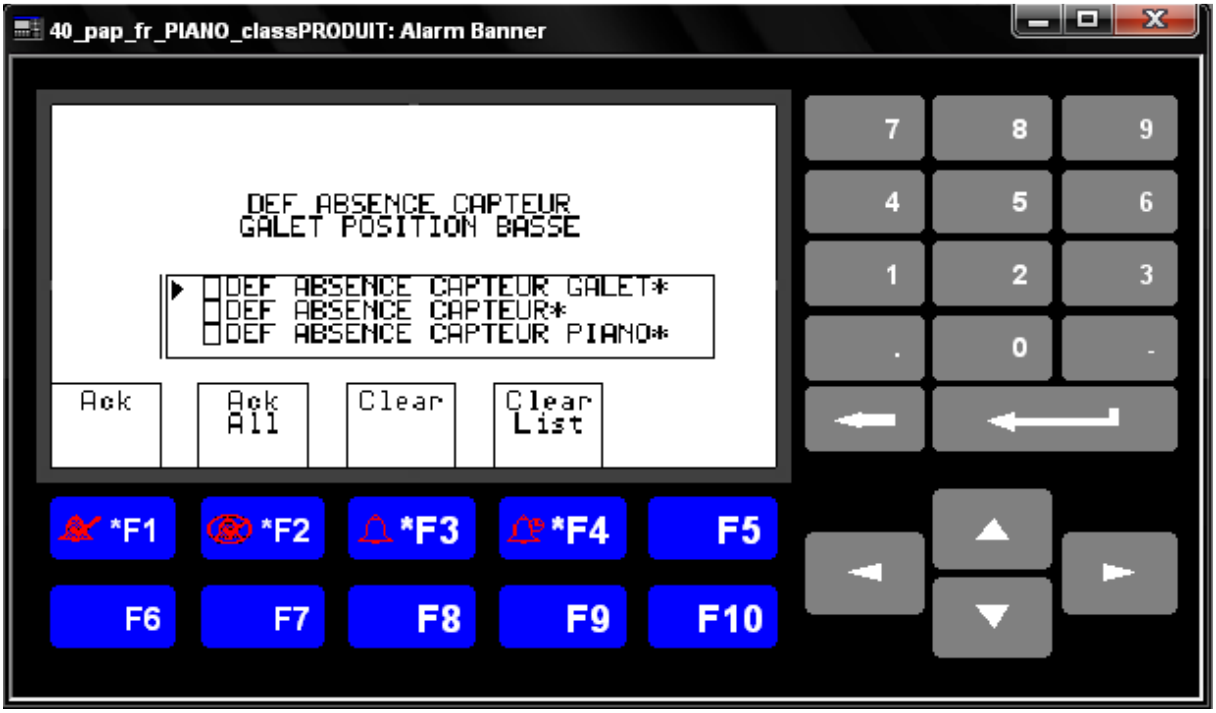
Pour chaque bouton, on double-clique dessus pour afficher la fenêtre de configuration et on lui affecte une touche fonction : F1 pour ''Acquitter'', F2 pour ''Tout acquitter'', F3 pour ''Effacer'' et F4 pour ''Tout effacer''.



Pour le bouton d'acquittement, nous lui avons associé le bit B3:0/0 afin d'acquitter les défauts sur le programme de détection comme le montre la figure suivante :



- **L'écran final :**



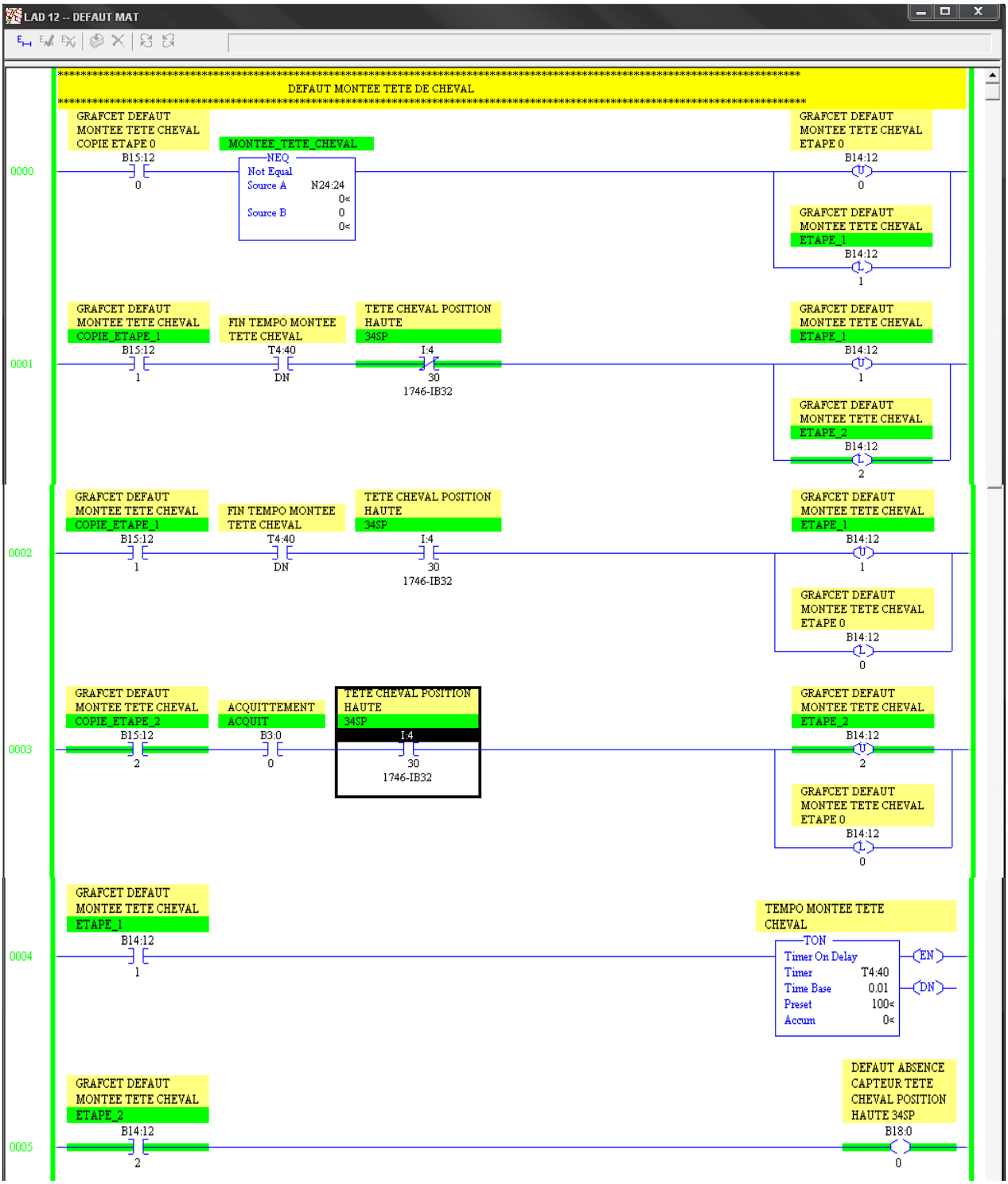
L'écran final contient **:**

- La zone d'affichage du défaut enclenché.
- La liste d'alarmes pour afficher l'historique des défauts.
- Les boutons ''Ack'' et ''Ack All'' pour l'acquittement des défauts.
- Les boutons ''Clear'' et ''Clear List'' pour l'effacement des défauts.

VI. Simulation :

Afin de valider la solution apportée, nous avons effectué une série de simulations sur PC avec le logiciel RSLogix 500.

. Exemple de simulation d'un défaut matériel (Défaut montée galet) :



Cet exemple de simulation montre la détection du défaut absence capteur tête de cheval position haute. Il suffit d'activer le bit N24:24/0 d'action de montée de la tête cheval et de laisser le bit état capteur tête cheval position haute I:4/30 à 0 pour activer le bit défaut B18:0/0 après un délai de temps T4:40 nécessaire à la montée de la tête de cheval.

Conclusion :

Dans ce chapitre, nous avons apporté les solutions nécessaires à la détection et l'affichage des différents défauts pouvant survenir sur la machine assembleuse pose à plat (PAP) au cours d'un cycle de production, les défauts ont été modélisés par le formalisme GRAFCET, puis traduits en langage LADDER par le logiciel de programmation RSLogix 500 pour être affichés ensuite sur l'IHM PanelView 550 à l'aide du progiciel PanelBuilder 32.

Conclusion générale

Ce projet a nécessité la mise en œuvre de plusieurs aspects notamment l'étude du fonctionnement de la machine, la modélisation et la programmation.

Notre travail a consisté en un développement d'un programme de détection et d'affichage de défauts pouvant survenir sur la machine assembleuse pose à plat (PAP). Nous avons pu recenser le maximum de défauts, les classer, concevoir les modèles grafcet pour les différents types et faire le programme.

Nous avons apprécié d'avoir à travailler en relation directe avec le monde industriel et nous avons été très attirés par notre sujet de stage. Nous avons également pu nous rendre compte des problèmes réels rencontrés par les ingénieurs en milieu industriel.

Cette expérience a été très enrichissante et a confirmé notre envie de travailler dans le domaine de l'automatique.

Enfin, ce travail non seulement, nous a permis de connaitre l'automate et l'interface Homme/Machine Allen Bradley, mais aussi comprendre que la performance d'une machine automatique dépend essentiellement de la qualité et de la technologie qu'elle comporte.

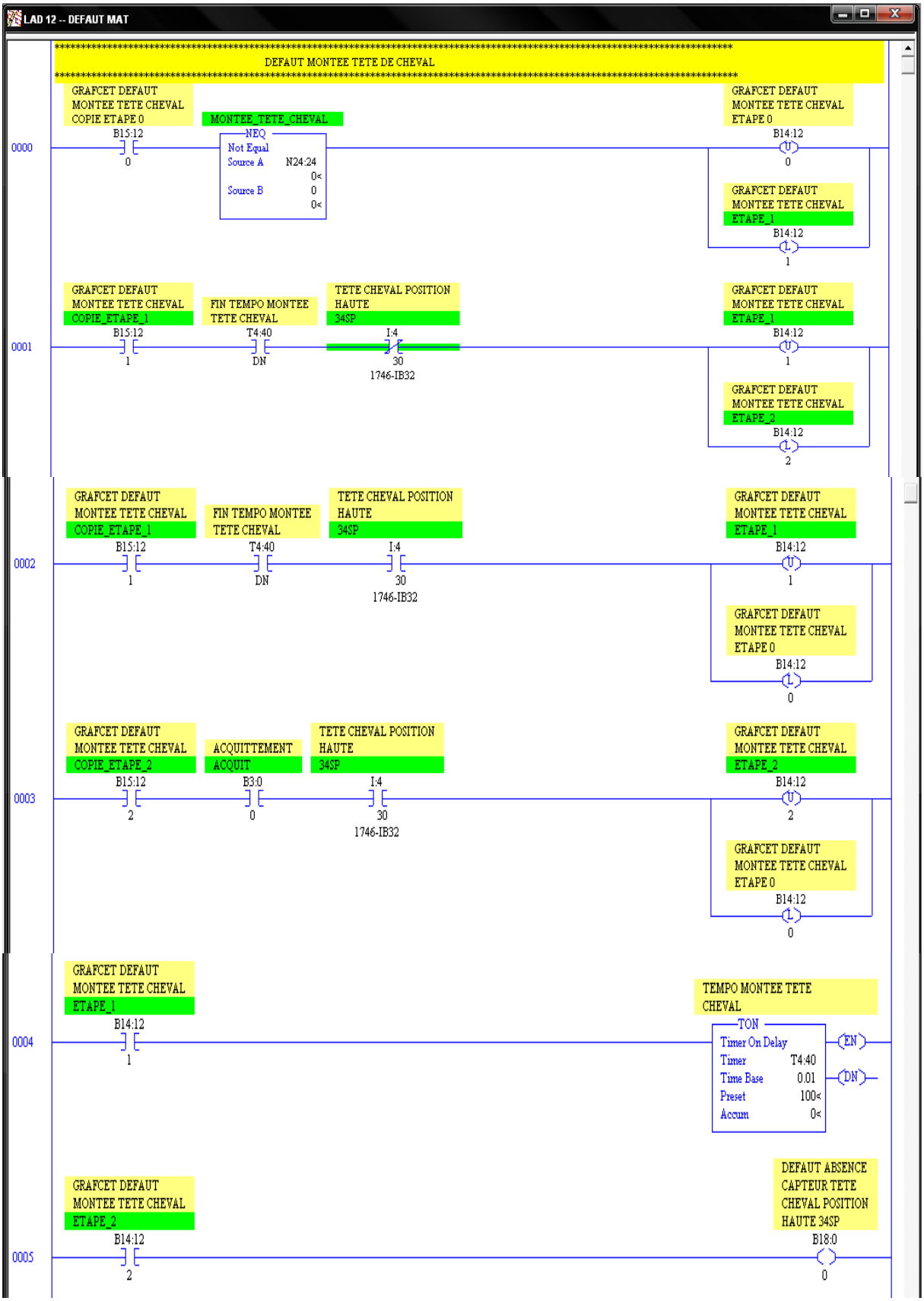
Vue la production continue (24h/24) à l'usine de Michelin Algérie, et l'importance que représente la machine pose à plat (PAP), il nous a pas été possible d'interrompre la production sur cette machine pour injecter le programme de détection sur l'automate et celui d'affichage sur le pupitre opérateur. Cependant, vue les avantages que va apporter notre travail, essentiellement en terme de facilité de la maintenance, la direction technique a prévu une journée lors du prochain arrêt afin de mettre en marche la solution que nous avons proposée.

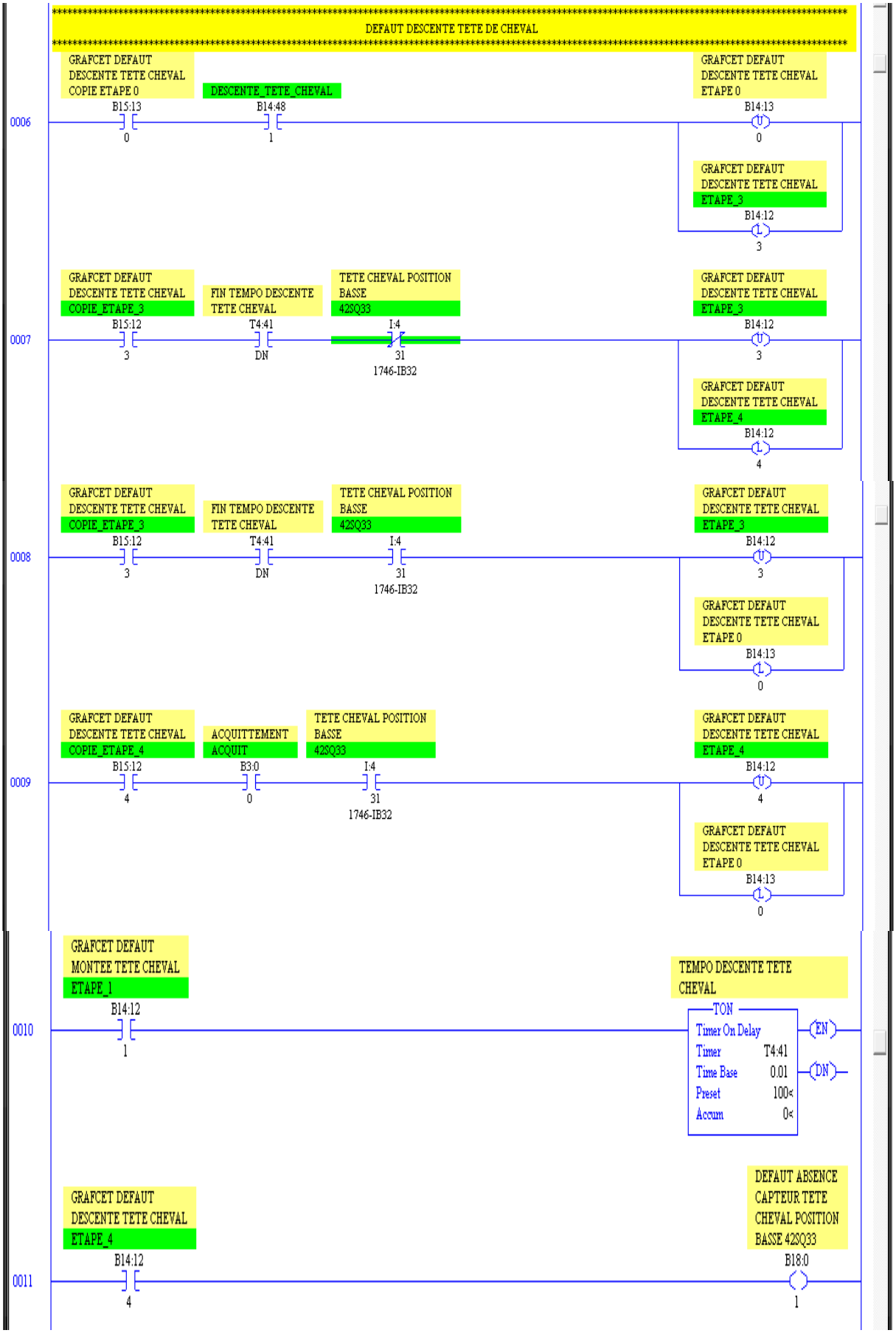
Nous espérons que notre travail pourra servir de support aux promotions à venir.

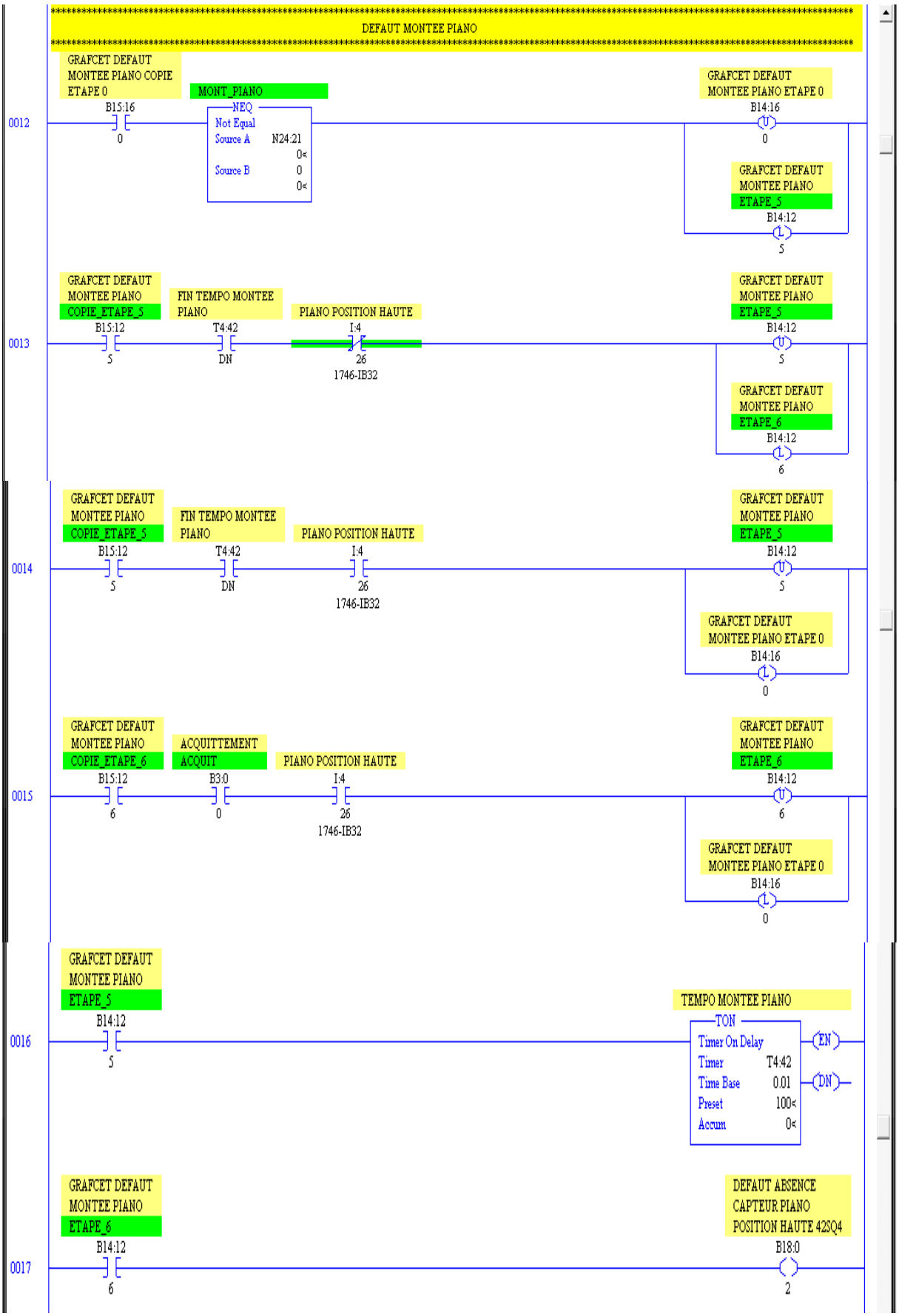
Annexe

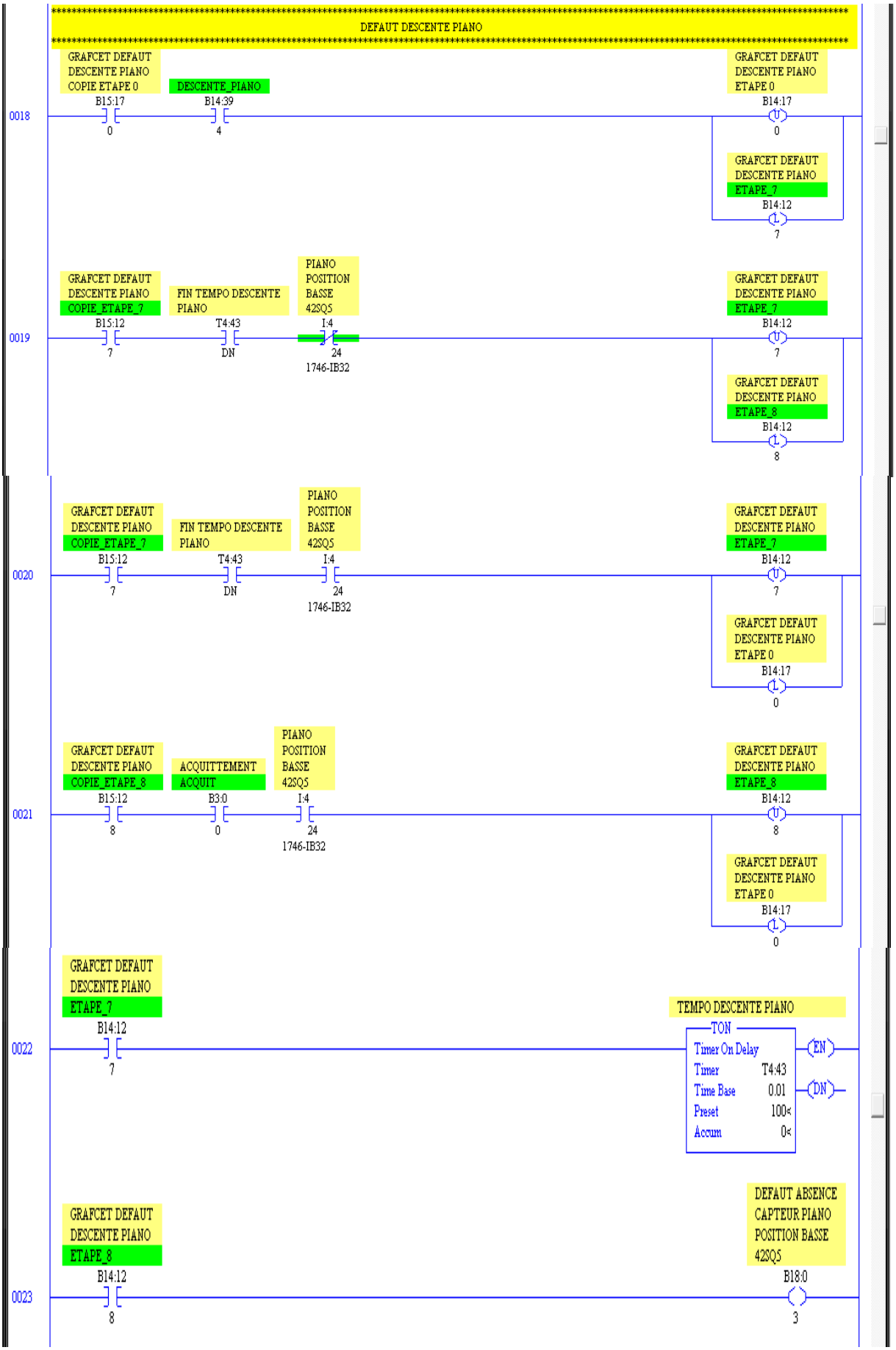
Programme de détection des défauts

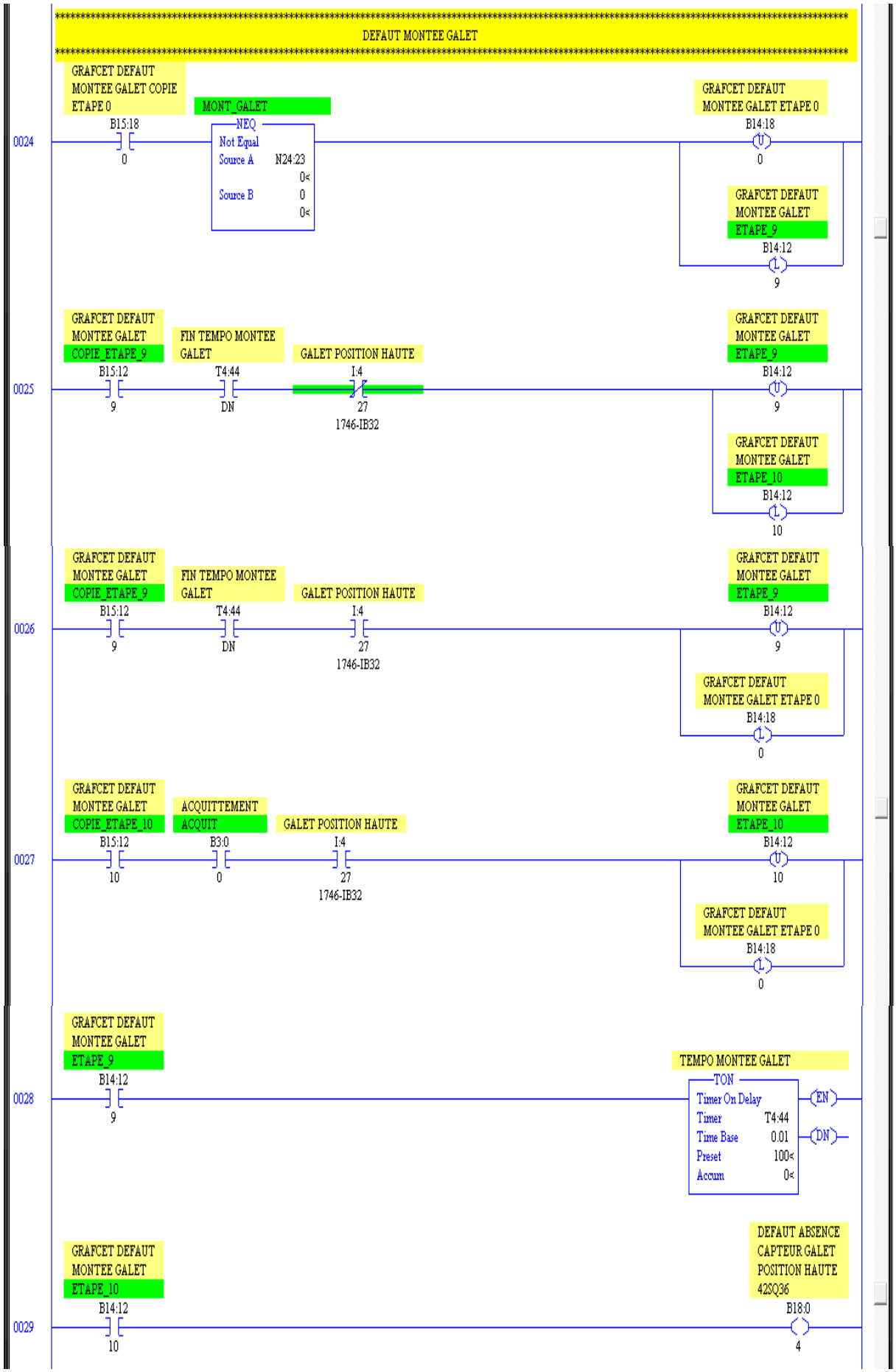
- **Programme pour la détection des défauts matériels :**

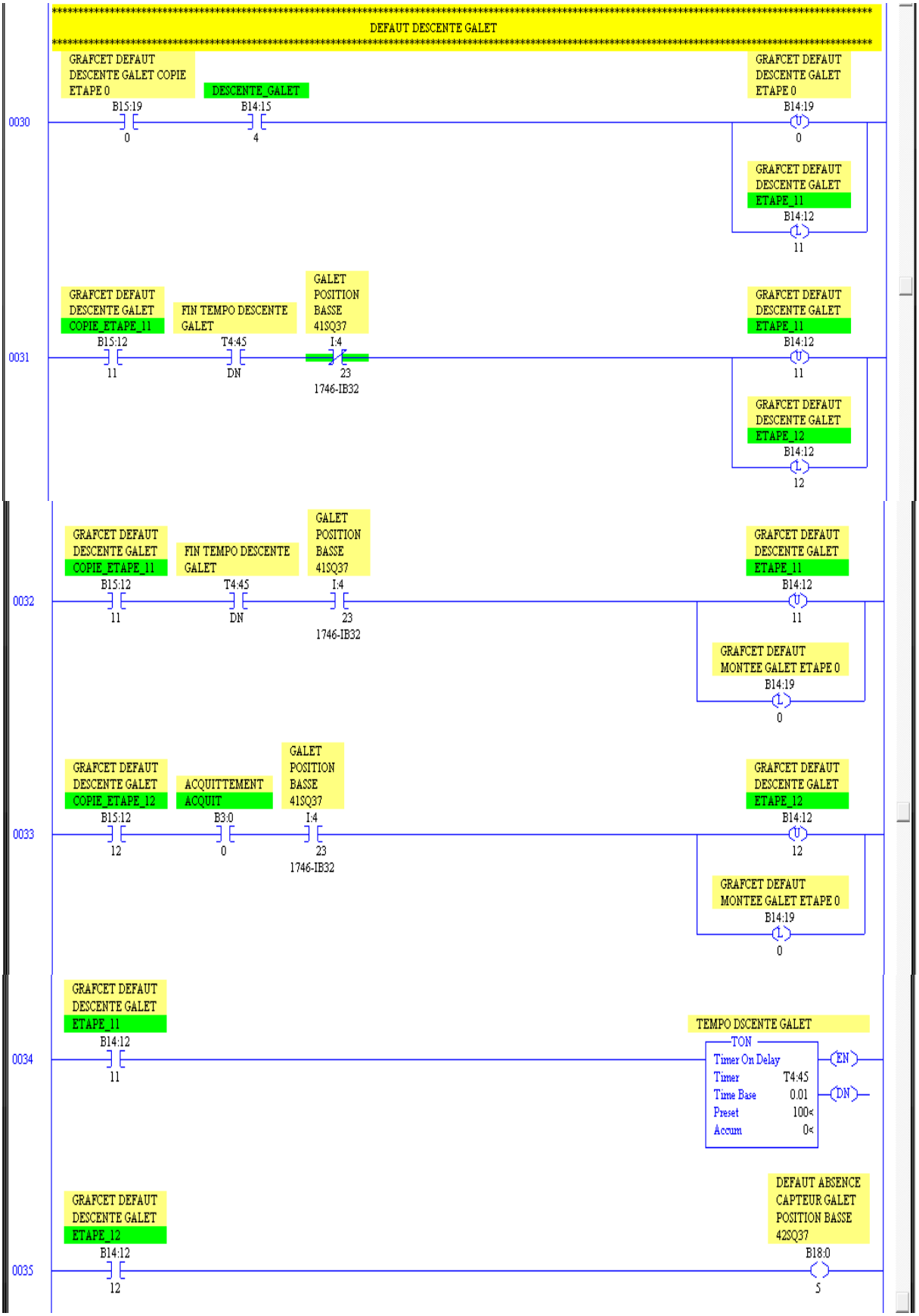


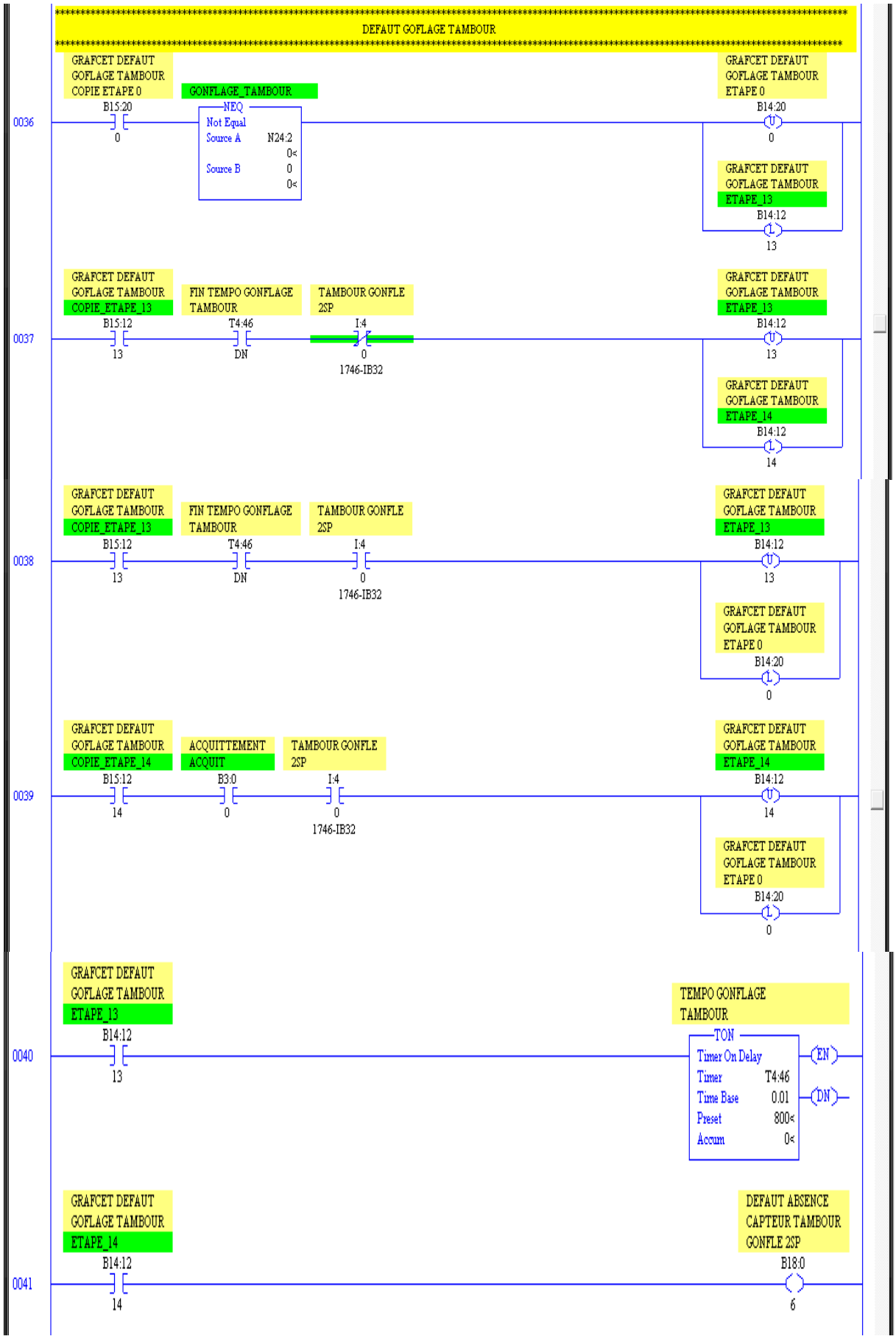

DEFAUT DESCENTE TETE DE CHEVAL
GRAFCET DEFAUT DESCENTE TETE CHEVAL COPIE ETAPE 0
B15:13
0
DESCENTE_TETE_CHEVAL
B14:48
1
GRAFCET DEFAUT DESCENTE TETE CHEVAL ETAPE 0
B14:13
0
GRAFCET DEFAUT DESCENTE TETE CHEVAL ETAPE_3
B14:12
3
0006
GRAFCET DEFAUT DESCENTE TETE CHEVAL COPIE_ETAPE_3
B15:12
3
FIN TEMPO DESCENTE TETE CHEVAL
T4:41
DN
TETE CHEVAL POSITION BASSE 42SQ33
I:4
31
1746-IB32
GRAFCET DEFAUT DESCENTE TETE CHEVAL ETAPE_3
B14:12
3
GRAFCET DEFAUT DESCENTE TETE CHEVAL ETAPE_4
B14:12
4
0007
GRAFCET DEFAUT DESCENTE TETE CHEVAL COPIE_ETAPE_3
B15:12
3
FIN TEMPO DESCENTE TETE CHEVAL
T4:41
DN
TETE CHEVAL POSITION BASSE 42SQ33
I:4
31
1746-IB32
GRAFCET DEFAUT DESCENTE TETE CHEVAL ETAPE_3
B14:12
3
GRAFCET DEFAUT DESCENTE TETE CHEVAL ETAPE 0
B14:13
0
0008
GRAFCET DEFAUT DESCENTE TETE CHEVAL COPIE_ETAPE_4
B15:12
4
ACQUITTEMENT ACQUIT
B3:0
0
TETE CHEVAL POSITION BASSE 42SQ33
I:4
31
1746-IB32
GRAFCET DEFAUT DESCENTE TETE CHEVAL ETAPE_4
B14:12
4
GRAFCET DEFAUT DESCENTE TETE CHEVAL ETAPE 0
B14:13
0
0009
GRAFCET DEFAUT MONTEE TETE CHEVAL ETAPE_1
B14:12
1
TEMPO DESCENTE TETE CHEVAL
TON
Timer On Delay
Timer T4:41
Time Base 0.01
Preset 100
Accum 0
EN
DN
0010
GRAFCET DEFAUT DESCENTE TETE CHEVAL ETAPE_4
B14:12
4
DEFAUT ABSENCE CAPTEUR TETE CHEVAL POSITION BASSE 42SQ33
B18:0
1
0011

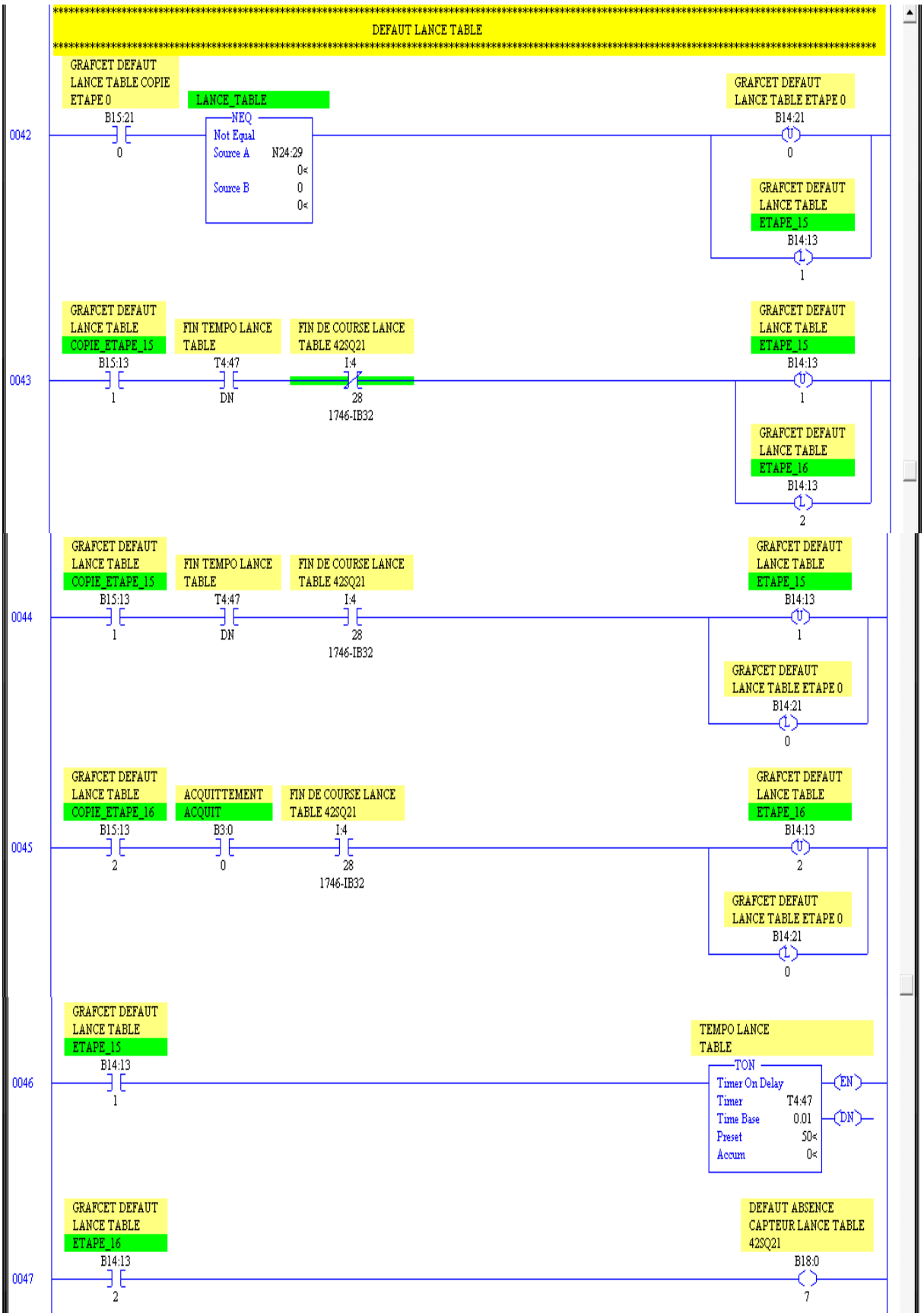

DEFAUT MONTEE PIANO
GRAFCET DEFAUT MONTEE PIANO COPIE ETAPE 0
B15:16
0
MONT_PIANO
NEQ
Not Equal
Source A N24:21
0<
Source B 0
0<
GRAFCET DEFAUT MONTEE PIANO ETAPE 0
B14:16
U
0
GRAFCET DEFAUT MONTEE PIANO ETAPE_5
B14:12
L
5
0012
GRAFCET DEFAUT MONTEE PIANO COPIE_ETAPE_5
B15:12
5
FIN TEMPO MONTEE PIANO
T4:42
DN
PIANO POSITION HAUTE
I:4
26
1746-IB32
GRAFCET DEFAUT MONTEE PIANO ETAPE_5
B14:12
U
5
GRAFCET DEFAUT MONTEE PIANO ETAPE_6
B14:12
L
6
0013
GRAFCET DEFAUT MONTEE PIANO COPIE_ETAPE_5
B15:12
5
FIN TEMPO MONTEE PIANO
T4:42
DN
PIANO POSITION HAUTE
I:4
26
1746-IB32
GRAFCET DEFAUT MONTEE PIANO ETAPE_5
B14:12
U
5
GRAFCET DEFAUT MONTEE PIANO ETAPE 0
B14:16
L
0
0014
GRAFCET DEFAUT MONTEE PIANO COPIE_ETAPE_6
B15:12
6
ACQUITTEMENT
ACQUIT
B3:0
0
PIANO POSITION HAUTE
I:4
26
1746-IB32
GRAFCET DEFAUT MONTEE PIANO ETAPE_6
B14:12
U
6
GRAFCET DEFAUT MONTEE PIANO ETAPE 0
B14:16
L
0
0015
GRAFCET DEFAUT MONTEE PIANO ETAPE_5
B14:12
5
TEMPO MONTEE PIANO
TON
Timer On Delay
Timer T4:42
Time Base 0.01
Preset 100<
Accum 0<
EN
DN
0016
GRAFCET DEFAUT MONTEE PIANO ETAPE_6
B14:12
6
DEFAUT ABSENCE CAPTEUR PIANO POSITION HAUTE 42SQ4
B18:0
2
0017

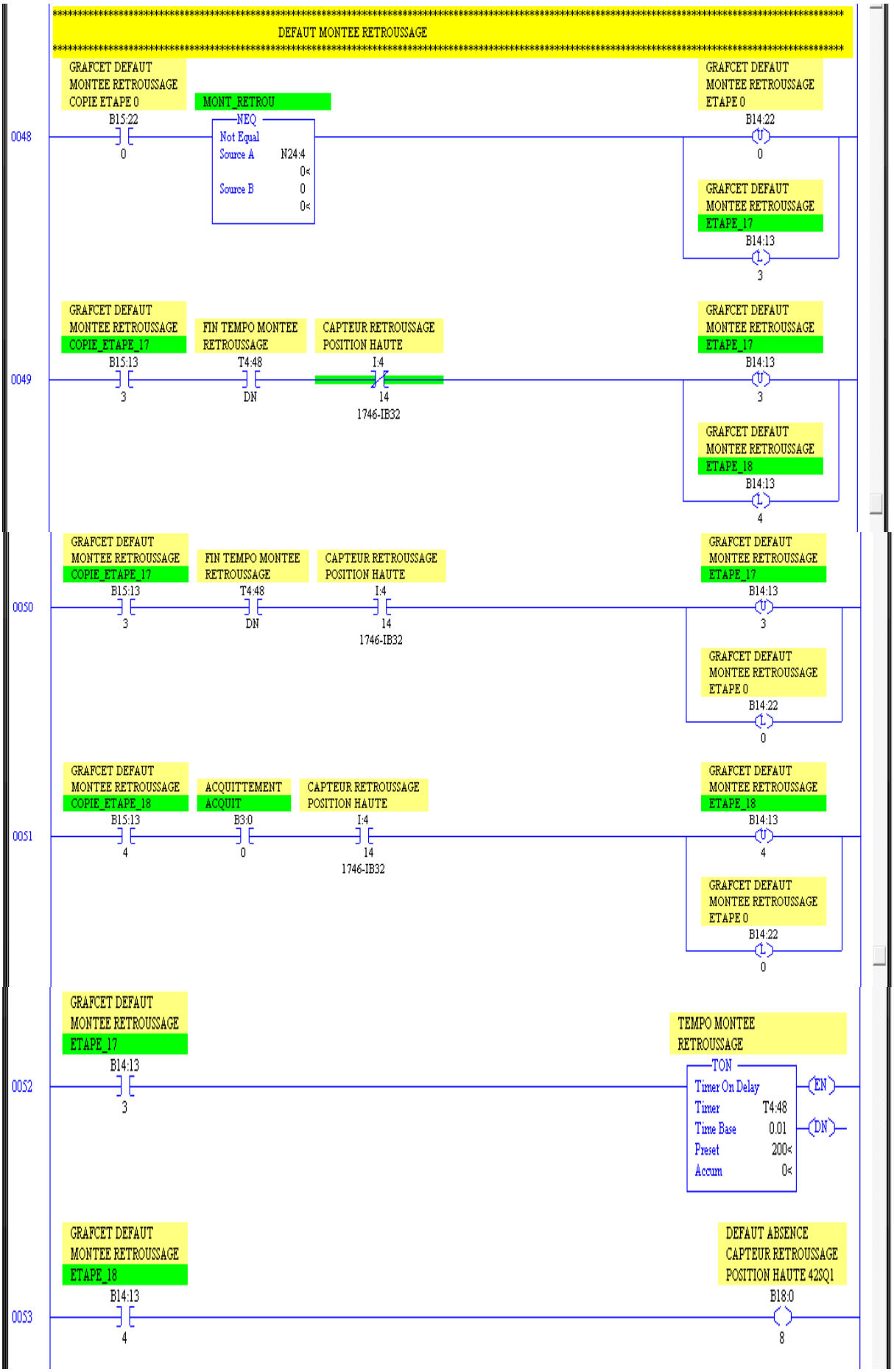

DEFAUT DESCENTE PIANO
0018
GRAFCET DEFAUT DESCENTE PIANO COPIE ETAPE 0
B15:17
0
DESCENTE_PIANO
B14:39
4
GRAFCET DEFAUT DESCENTE PIANO ETAPE 0
B14:17
U
0
GRAFCET DEFAUT DESCENTE PIANO ETAPE_7
B14:12
L
7
0019
GRAFCET DEFAUT DESCENTE PIANO COPIE_ETAPE_7
B15:12
7
FIN TEMPO DESCENTE PIANO
T4:43
DN
PIANO POSITION BASSE 42SQ5
I:4
24
1746-IB32
GRAFCET DEFAUT DESCENTE PIANO ETAPE_7
B14:12
U
7
GRAFCET DEFAUT DESCENTE PIANO ETAPE_8
B14:12
L
8
0020
GRAFCET DEFAUT DESCENTE PIANO COPIE_ETAPE_7
B15:12
7
FIN TEMPO DESCENTE PIANO
T4:43
DN
PIANO POSITION BASSE 42SQ5
I:4
24
1746-IB32
GRAFCET DEFAUT DESCENTE PIANO ETAPE_7
B14:12
U
7
GRAFCET DEFAUT DESCENTE PIANO ETAPE 0
B14:17
L
0
0021
GRAFCET DEFAUT DESCENTE PIANO COPIE_ETAPE_8
B15:12
8
ACQUITTEMENT ACQUIT
B3:0
0
PIANO POSITION BASSE 42SQ5
I:4
24
1746-IB32
GRAFCET DEFAUT DESCENTE PIANO ETAPE_8
B14:12
U
8
GRAFCET DEFAUT DESCENTE PIANO ETAPE 0
B14:17
L
0
0022
GRAFCET DEFAUT DESCENTE PIANO ETAPE_7
B14:12
7
TEMPO DESCENTE PIANO
TON
Timer On Delay
Timer T4:43
Time Base 0.01
Preset 100
Accum 0
EN
DN
0023
GRAFCET DEFAUT DESCENTE PIANO ETAPE_8
B14:12
8
DEFAUT ABSENCE CAPTEUR PIANO POSITION BASSE 42SQ5
B18:0
3

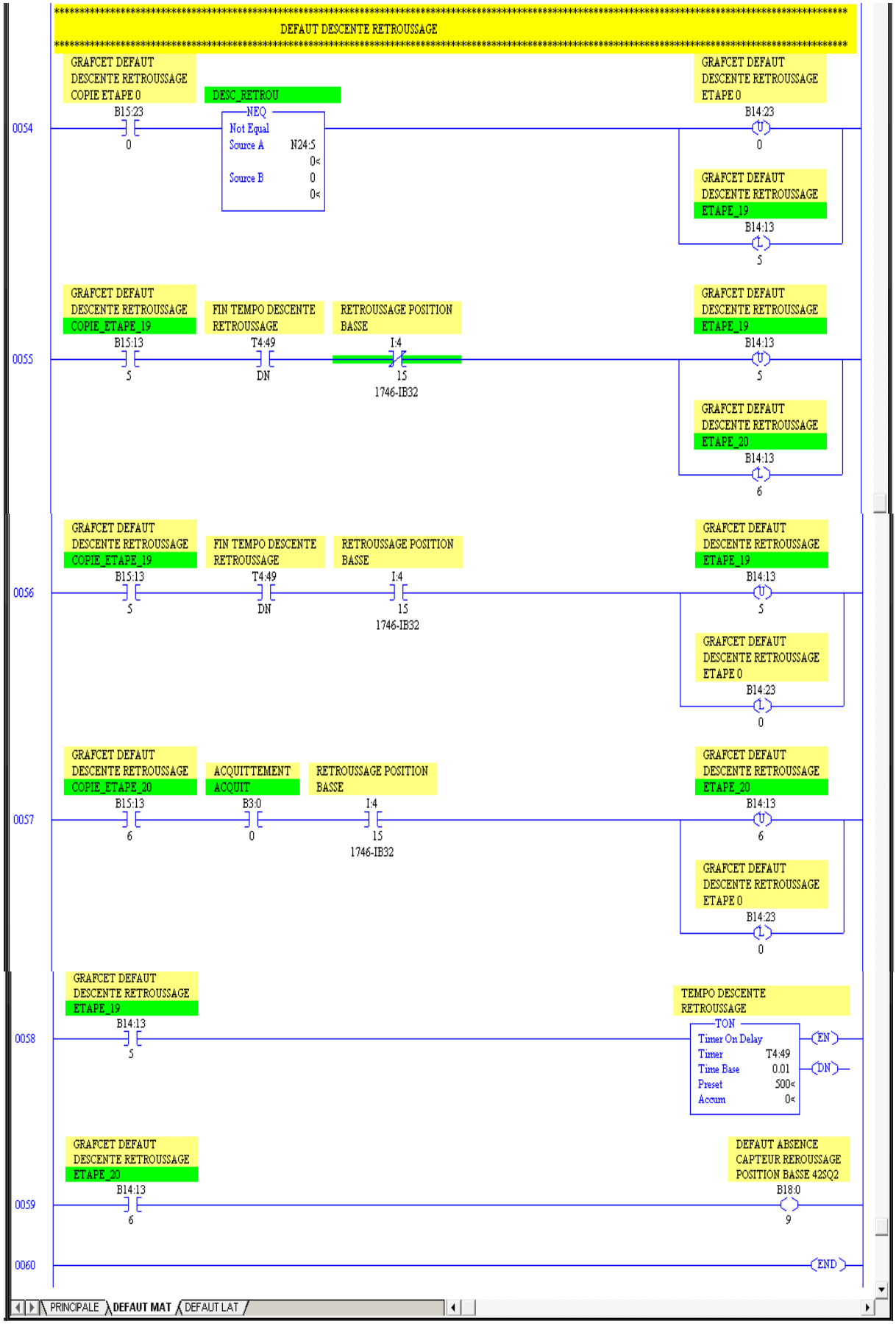

DEFAUT MONTEE GALET
GRAFCET DEFAUT MONTEE GALET COPIE ETAPE 0
MONT_GALET
B15:18
0
NEQ
Not Equal
Source A N24:23
Source B 0
GRAFCET DEFAUT MONTEE GALET ETAPE 0
B14:18
0
GRAFCET DEFAUT MONTEE GALET ETAPE_9
B14:12
9
0024
GRAFCET DEFAUT MONTEE GALET COPIE_ETAPE_9
B15:12
9
FIN TEMPO MONTEE GALET
T4:44
DN
GALET POSITION HAUTE
I:4
27
1746-IB32
GRAFCET DEFAUT MONTEE GALET ETAPE_9
B14:12
9
GRAFCET DEFAUT MONTEE GALET ETAPE_10
B14:12
10
0025
GRAFCET DEFAUT MONTEE GALET COPIE_ETAPE_9
B15:12
9
FIN TEMPO MONTEE GALET
T4:44
DN
GALET POSITION HAUTE
I:4
27
1746-IB32
GRAFCET DEFAUT MONTEE GALET ETAPE_9
B14:12
9
GRAFCET DEFAUT MONTEE GALET ETAPE 0
B14:18
0
0026
GRAFCET DEFAUT MONTEE GALET COPIE_ETAPE_10
B15:12
10
ACQUITTEMENT
ACQUIT
B3:0
0
GALET POSITION HAUTE
I:4
27
1746-IB32
GRAFCET DEFAUT MONTEE GALET ETAPE_10
B14:12
10
GRAFCET DEFAUT MONTEE GALET ETAPE 0
B14:18
0
0027
GRAFCET DEFAUT MONTEE GALET ETAPE_9
B14:12
9
TEMPO MONTEE GALET
TON
Timer On Delay
Timer T4:44
Time Base 0.01
Preset 100
Accum 0
EN
DN
0028
GRAFCET DEFAUT MONTEE GALET ETAPE_10
B14:12
10
DEFAUT ABSENCE CAPTEUR GALET POSITION HAUTE 42SQ36
B18:0
4
0029


DEFAUT DESCENTE GALET
GRAFCET DEFAUT DESCENTE GALET COPIE ETAPE 0
B15:19
0
DESCENTE_GALET
B14:15
4
GRAFCET DEFAUT DESCENTE GALET ETAPE 0
B14:19
0
GRAFCET DEFAUT DESCENTE GALET ETAPE_11
B14:12
11
0030
GRAFCET DEFAUT DESCENTE GALET COPIE_ETAPE_11
B15:12
11
FIN TEMPO DESCENTE GALET
T4:45
DN
GALET POSITION BASSE 41SQ37
I:4
23
1746-IB32
GRAFCET DEFAUT DESCENTE GALET ETAPE_11
B14:12
11
GRAFCET DEFAUT DESCENTE GALET ETAPE_12
B14:12
12
0031
GRAFCET DEFAUT DESCENTE GALET COPIE_ETAPE_11
B15:12
11
FIN TEMPO DESCENTE GALET
T4:45
DN
GALET POSITION BASSE 41SQ37
I:4
23
1746-IB32
GRAFCET DEFAUT DESCENTE GALET ETAPE_11
B14:12
11
GRAFCET DEFAUT MONTEE GALET ETAPE 0
B14:19
0
0032
GRAFCET DEFAUT DESCENTE GALET COPIE_ETAPE_12
B15:12
12
ACQUITTEMENT
ACQUIT
B3:0
0
GALET POSITION BASSE 41SQ37
I:4
23
1746-IB32
GRAFCET DEFAUT DESCENTE GALET ETAPE_12
B14:12
12
GRAFCET DEFAUT MONTEE GALET ETAPE 0
B14:19
0
0033
GRAFCET DEFAUT DESCENTE GALET ETAPE_11
B14:12
11
TEMPO DSCENTE GALET
TON
Timer On Delay
Timer T4:45
Time Base 0.01
Preset 100
Accum 0
EN
DN
0034
GRAFCET DEFAUT DESCENTE GALET ETAPE_12
B14:12
12
DEFAUT ABSENCE CAPTEUR GALET POSITION BASSE 42SQ37
B18:0
5
0035

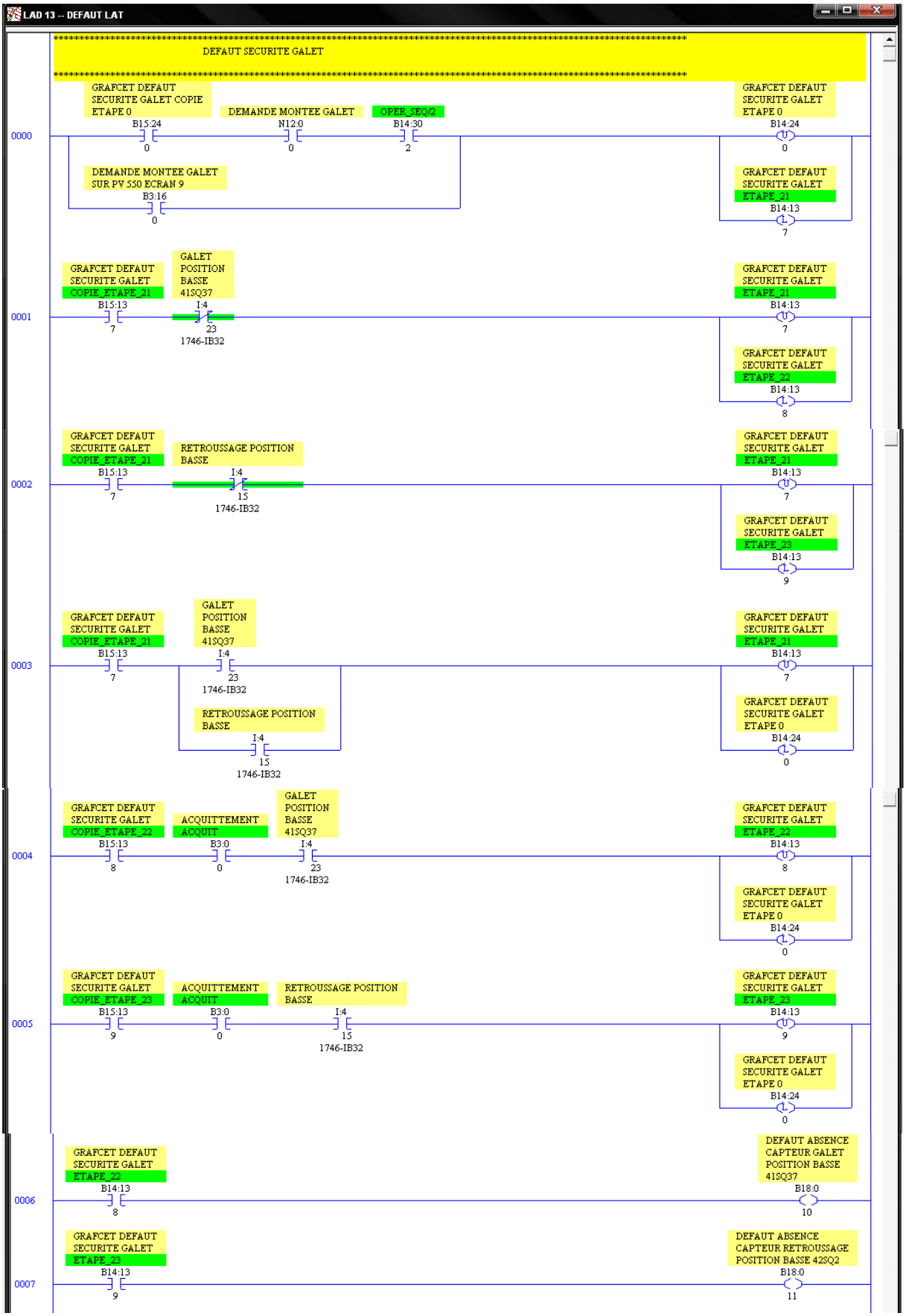

DEFAUT GOFLAGE TAMBOUR
GRAFCET DEFAUT GOFLAGE TAMBOUR COPIE ETAPE 0
B15:20
0
GONFLAGE_TAMBOUR
NEQ
Not Equal
Source A N24:2 0
Source B 0 0
GRAFCET DEFAUT GOFLAGE TAMBOUR ETAPE 0
B14:20
0
GRAFCET DEFAUT GOFLAGE TAMBOUR ETAPE_13
B14:12
13
0036
GRAFCET DEFAUT GOFLAGE TAMBOUR COPIE_ETAPE_13
B15:12
13
FIN TEMPO GONFLAGE TAMBOUR
T4:46
DN
TAMBOUR GONFLE 2SP
I:4
0
1746-IB32
GRAFCET DEFAUT GOFLAGE TAMBOUR ETAPE_13
B14:12
13
GRAFCET DEFAUT GOFLAGE TAMBOUR ETAPE_14
B14:12
14
0037
GRAFCET DEFAUT GOFLAGE TAMBOUR COPIE_ETAPE_13
B15:12
13
FIN TEMPO GONFLAGE TAMBOUR
T4:46
DN
TAMBOUR GONFLE 2SP
I:4
0
1746-IB32
GRAFCET DEFAUT GOFLAGE TAMBOUR ETAPE_13
B14:12
13
GRAFCET DEFAUT GOFLAGE TAMBOUR ETAPE 0
B14:20
0
0038
GRAFCET DEFAUT GOFLAGE TAMBOUR COPIE_ETAPE_14
B15:12
14
ACQUITTEMENT ACQUIT
B3:0
0
TAMBOUR GONFLE 2SP
I:4
0
1746-IB32
GRAFCET DEFAUT GOFLAGE TAMBOUR ETAPE_14
B14:12
14
GRAFCET DEFAUT GOFLAGE TAMBOUR ETAPE 0
B14:20
0
0039
GRAFCET DEFAUT GOFLAGE TAMBOUR ETAPE_13
B14:12
13
TEMPO GONFLAGE TAMBOUR
TON
Timer On Delay
Timer T4:46
Time Base 0.01
Preset 800
Accum 0
EN
DN
0040
GRAFCET DEFAUT GOFLAGE TAMBOUR ETAPE_14
B14:12
14
DEFAUT ABSENCE CAPTEUR TAMBOUR GONFLE 2SP
B18:0
6
0041

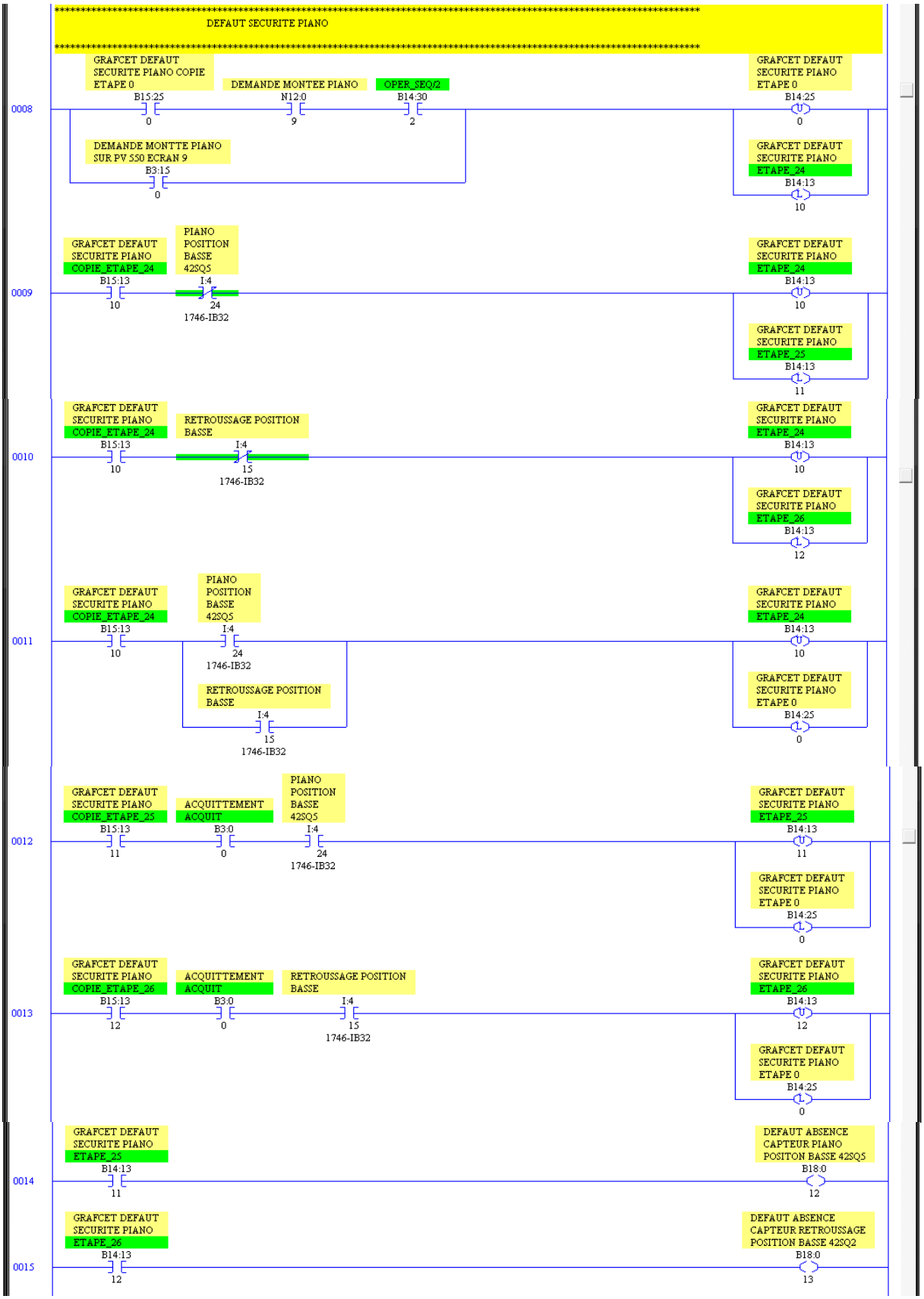

DEFAUT LANCE TABLE
GRAFCET DEFAUT LANCE TABLE COPIE ETAPE 0
B15:21
0
LANCE_TABLE
NEQ
Not Equal
Source A N24:29
0
Source B 0
0
GRAFCET DEFAUT LANCE TABLE ETAPE 0
B14:21
0
GRAFCET DEFAUT LANCE TABLE ETAPE_15
B14:13
1
0042
GRAFCET DEFAUT LANCE TABLE COPIE_ETAPE_15
B15:13
1
FIN TEMPO LANCE TABLE
T4:47
DN
FIN DE COURSE LANCE TABLE 42SQ21
I:4
28
1746-IB32
GRAFCET DEFAUT LANCE TABLE ETAPE_15
B14:13
1
GRAFCET DEFAUT LANCE TABLE ETAPE_16
B14:13
2
0043
GRAFCET DEFAUT LANCE TABLE COPIE_ETAPE_15
B15:13
1
FIN TEMPO LANCE TABLE
T4:47
DN
FIN DE COURSE LANCE TABLE 42SQ21
I:4
28
1746-IB32
GRAFCET DEFAUT LANCE TABLE ETAPE_15
B14:13
1
GRAFCET DEFAUT LANCE TABLE ETAPE 0
B14:21
0
0044
GRAFCET DEFAUT LANCE TABLE COPIE_ETAPE_16
B15:13
2
ACQUITTEMENT ACQUIT
B3:0
0
FIN DE COURSE LANCE TABLE 42SQ21
I:4
28
1746-IB32
GRAFCET DEFAUT LANCE TABLE ETAPE_16
B14:13
2
GRAFCET DEFAUT LANCE TABLE ETAPE 0
B14:21
0
0045
GRAFCET DEFAUT LANCE TABLE ETAPE_15
B14:13
1
TEMPO LANCE TABLE
TON
Timer On Delay
Timer T4:47
Time Base 0.01
Preset 50
Accum 0
EN
DN
0046
GRAFCET DEFAUT LANCE TABLE ETAPE_16
B14:13
2
DEFAUT ABSENCE CAPTEUR LANCE TABLE 42SQ21
B18:0
7
0047

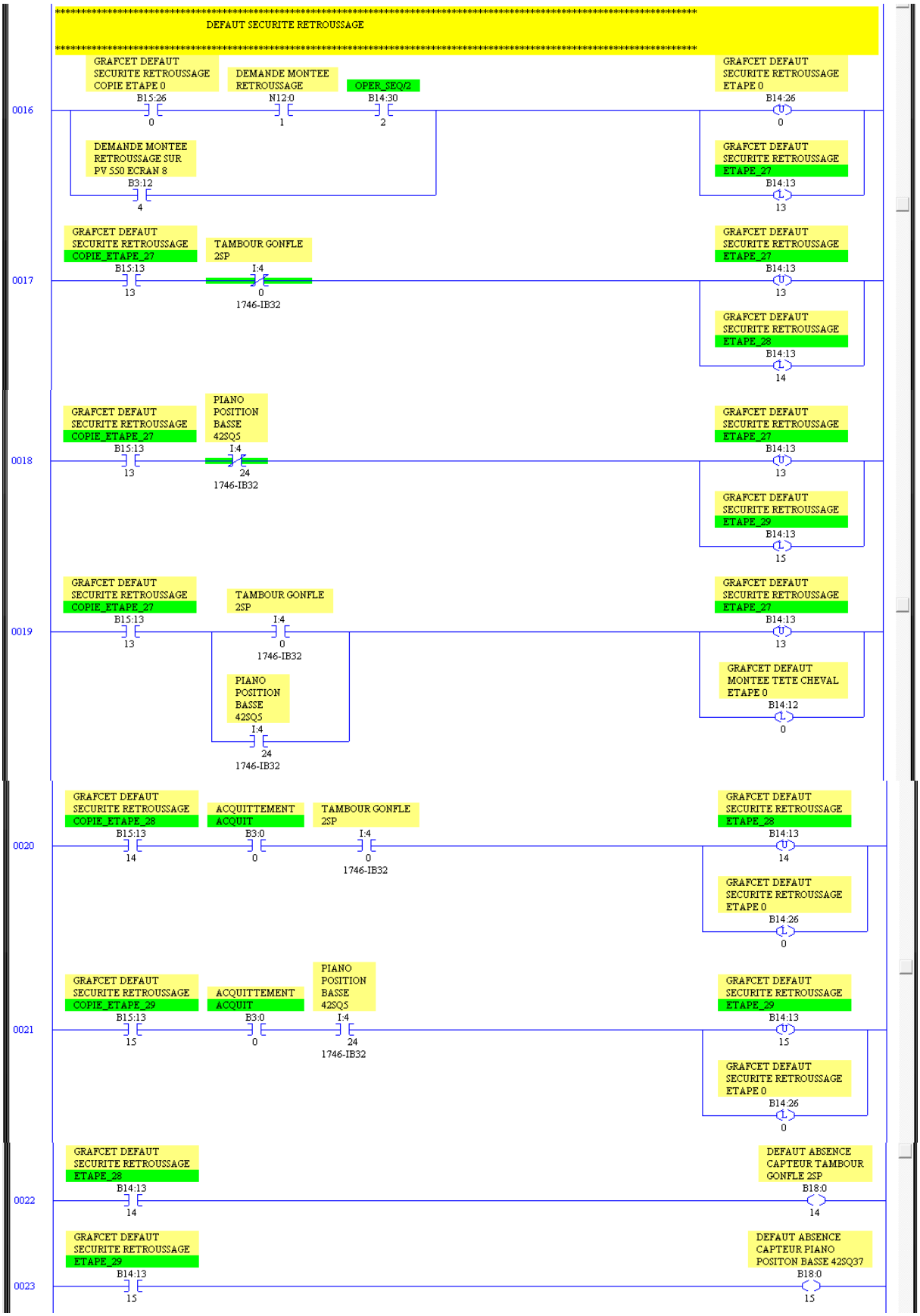

DEFAUT MONTEE RETROUSSAGE
GRAFCET DEFAUT MONTEE RETROUSSAGE COPIE ETAPE 0
B15:22
0
MONT_RETROU
NEQ
Not Equal
Source A N24:4
Source B 0
GRAFCET DEFAUT MONTEE RETROUSSAGE ETAPE 0
B14:22
0
GRAFCET DEFAUT MONTEE RETROUSSAGE ETAPE_17
B14:13
3
0048
GRAFCET DEFAUT MONTEE RETROUSSAGE COPIE_ETAPE_17
B15:13
3
FIN TEMPO MONTEE RETROUSSAGE
T4:48
DN
CAPTEUR RETROUSSAGE POSITION HAUTE
I:4
14
1746-IB32
GRAFCET DEFAUT MONTEE RETROUSSAGE ETAPE_17
B14:13
3
GRAFCET DEFAUT MONTEE RETROUSSAGE ETAPE_18
B14:13
4
0049
GRAFCET DEFAUT MONTEE RETROUSSAGE COPIE_ETAPE_17
B15:13
3
FIN TEMPO MONTEE RETROUSSAGE
T4:48
DN
CAPTEUR RETROUSSAGE POSITION HAUTE
I:4
14
1746-IB32
GRAFCET DEFAUT MONTEE RETROUSSAGE ETAPE_17
B14:13
3
GRAFCET DEFAUT MONTEE RETROUSSAGE ETAPE 0
B14:22
0
0050
GRAFCET DEFAUT MONTEE RETROUSSAGE COPIE_ETAPE_18
B15:13
4
ACQUITTEMENT ACQUIT
B3:0
0
CAPTEUR RETROUSSAGE POSITION HAUTE
I:4
14
1746-IB32
GRAFCET DEFAUT MONTEE RETROUSSAGE ETAPE_18
B14:13
4
GRAFCET DEFAUT MONTEE RETROUSSAGE ETAPE 0
B14:22
0
0051
GRAFCET DEFAUT MONTEE RETROUSSAGE ETAPE_17
B14:13
3
TEMPO MONTEE RETROUSSAGE
TON
Timer On Delay
Timer T4:48
Time Base 0.01
Preset 200
Accum 0
EN
DN
0052
GRAFCET DEFAUT MONTEE RETROUSSAGE ETAPE_18
B14:13
4
DEFAUT ABSENCE CAPTEUR RETROUSSAGE POSITION HAUTE 42SQ1
B18:0
8
0053

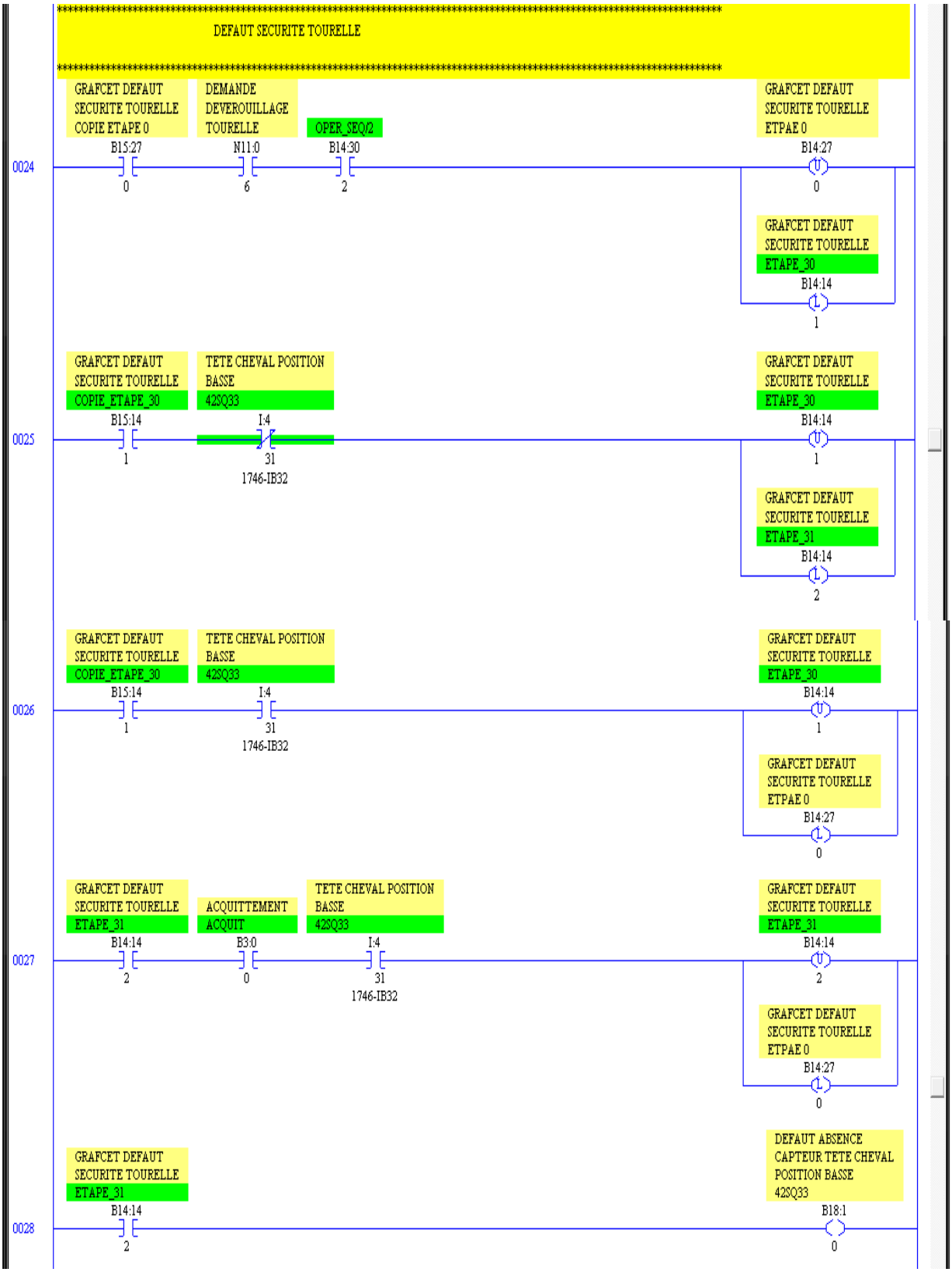

DEFAUT DESCENTE RETROUSSAGE
GRAFCET DEFAUT DESCENTE RETROUSSAGE COPIE ETAPE 0
B15:23
0
DESC_RETROU
NEQ
Not Equal
Source A N24:5
Source B 0
GRAFCET DEFAUT DESCENTE RETROUSSAGE ETAPE 0
B14:23
0
GRAFCET DEFAUT DESCENTE RETROUSSAGE ETAPE_19
B14:13
5
0054
GRAFCET DEFAUT DESCENTE RETROUSSAGE COPIE_ETAPE_19
B15:13
5
FIN TEMPO DESCENTE RETROUSSAGE
T4:49
DN
RETROUSSAGE POSITION BASSE
I:4
15
1746-IB32
GRAFCET DEFAUT DESCENTE RETROUSSAGE ETAPE_19
B14:13
5
GRAFCET DEFAUT DESCENTE RETROUSSAGE ETAPE_20
B14:13
6
0055
GRAFCET DEFAUT DESCENTE RETROUSSAGE COPIE_ETAPE_19
B15:13
5
FIN TEMPO DESCENTE RETROUSSAGE
T4:49
DN
RETROUSSAGE POSITION BASSE
I:4
15
1746-IB32
GRAFCET DEFAUT DESCENTE RETROUSSAGE ETAPE_19
B14:13
5
GRAFCET DEFAUT DESCENTE RETROUSSAGE ETAPE 0
B14:23
0
0056
GRAFCET DEFAUT DESCENTE RETROUSSAGE COPIE_ETAPE_20
B15:13
6
ACQUITTEMENT ACQUIT
B3:0
0
RETROUSSAGE POSITION BASSE
I:4
15
1746-IB32
GRAFCET DEFAUT DESCENTE RETROUSSAGE ETAPE_20
B14:13
6
GRAFCET DEFAUT DESCENTE RETROUSSAGE ETAPE 0
B14:23
0
0057
GRAFCET DEFAUT DESCENTE RETROUSSAGE ETAPE_19
B14:13
5
TEMPO DESCENTE RETROUSSAGE
TON
Timer On Delay
Timer T4:49
Time Base 0.01
Preset 500
Accum 0
EN
DN
0058
GRAFCET DEFAUT DESCENTE RETROUSSAGE ETAPE_20
B14:13
6
DEFAUT ABSENCE CAPTEUR REROUSSAGE POSITION BASSE 42SQ2
B18:0
9
0059
END
0060
PRINCIPALE
DEFAUT MAT
DEFAUT LAT

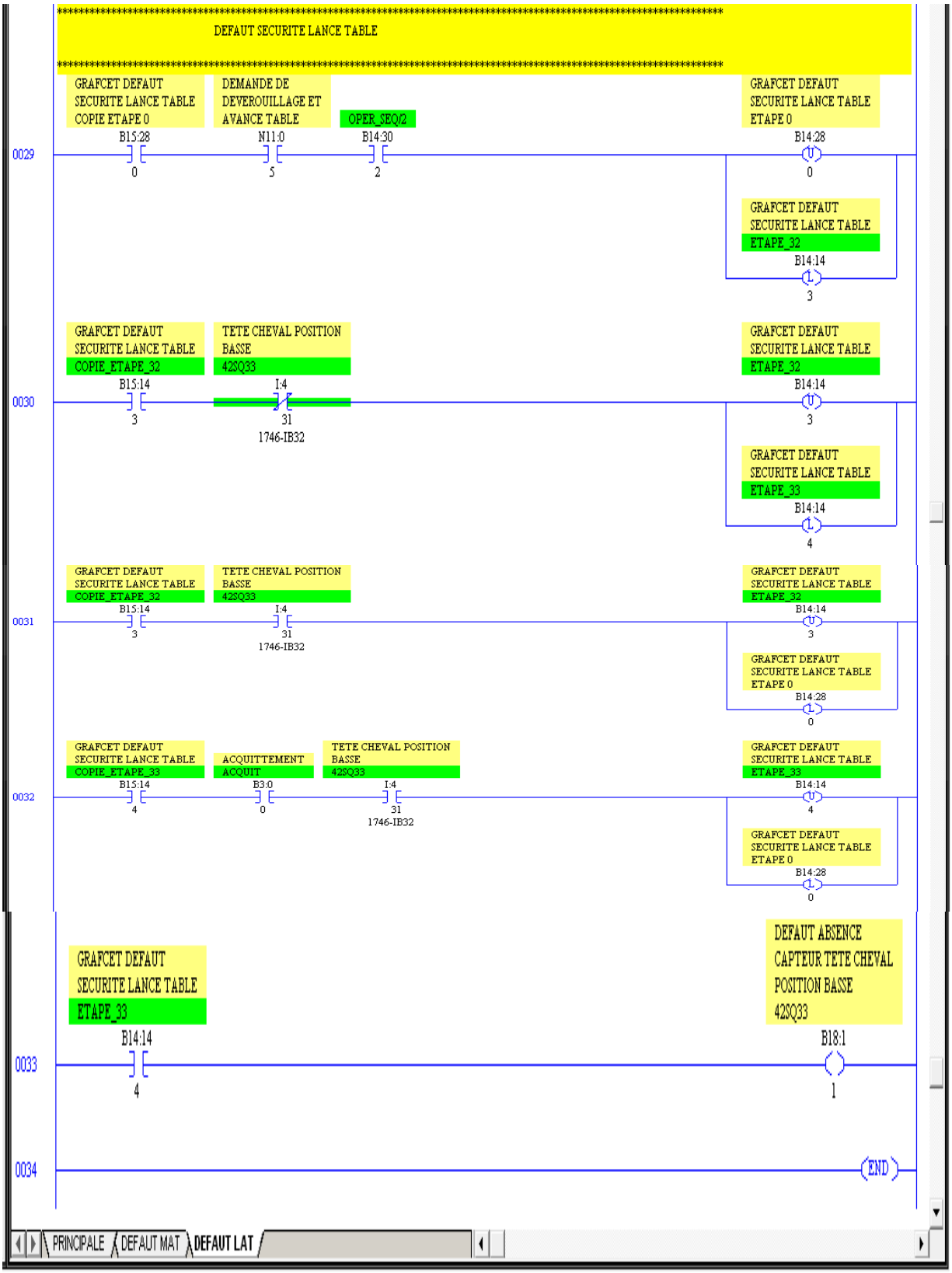
- **Programme pour la détection des défauts liés aux transitions :**



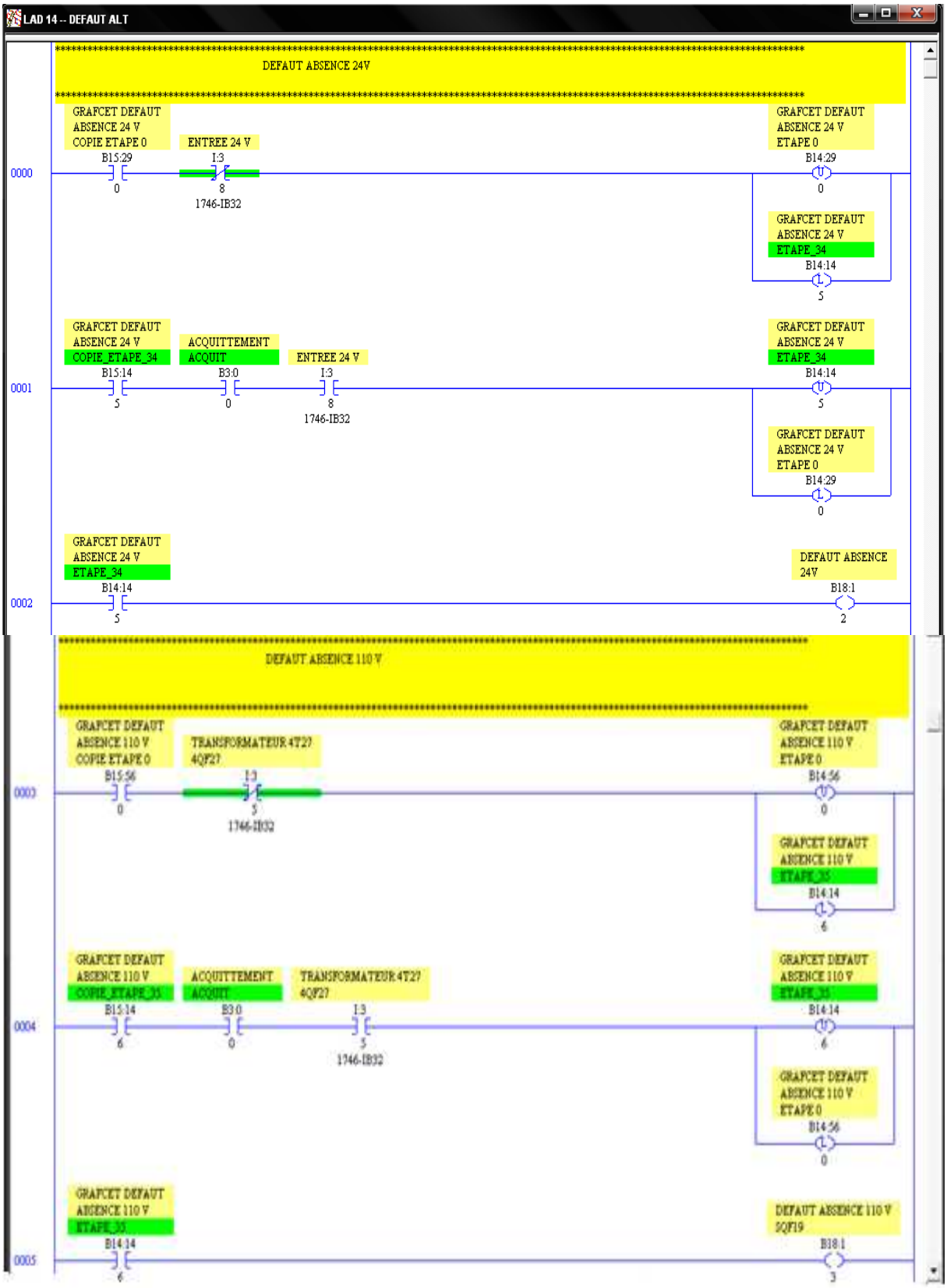

DEFAUT SECURITE PIANO
0008
GRAFCET DEFAUT SECURITE PIANO COPIE ETAPE 0
B15:25 0
DEMANDE MONTEE PIANO
N12:0 9
OPER_SEQ/2
B14:30 2
DEMANDE MONTTE PIANO SUR PV 550 ECRAN 9
B3:15 0
GRAFCET DEFAUT SECURITE PIANO ETAPE 0
B14:25 U 0
GRAFCET DEFAUT SECURITE PIANO ETAPE_24
B14:13 L 10
0009
GRAFCET DEFAUT SECURITE PIANO COPIE_ETAPE_24
B15:13 10
PIANO POSITION BASSE 42SQ5
I:4 24 1746-IB32
GRAFCET DEFAUT SECURITE PIANO ETAPE_24
B14:13 U 10
GRAFCET DEFAUT SECURITE PIANO ETAPE_25
B14:13 L 11
0010
GRAFCET DEFAUT SECURITE PIANO COPIE_ETAPE_24
B15:13 10
RETROUSSAGE POSITION BASSE
I:4 15 1746-IB32
GRAFCET DEFAUT SECURITE PIANO ETAPE_24
B14:13 U 10
GRAFCET DEFAUT SECURITE PIANO ETAPE_26
B14:13 L 12
0011
GRAFCET DEFAUT SECURITE PIANO COPIE_ETAPE_24
B15:13 10
PIANO POSITION BASSE 42SQ5
I:4 24 1746-IB32
RETROUSSAGE POSITION BASSE
I:4 15 1746-IB32
GRAFCET DEFAUT SECURITE PIANO ETAPE_24
B14:13 U 10
GRAFCET DEFAUT SECURITE PIANO ETAPE 0
B14:25 L 0
0012
GRAFCET DEFAUT SECURITE PIANO COPIE_ETAPE_25
B15:13 11
ACQUITTEMENT ACQUIT
B3:0 0
PIANO POSITION BASSE 42SQ5
I:4 24 1746-IB32
GRAFCET DEFAUT SECURITE PIANO ETAPE_25
B14:13 U 11
GRAFCET DEFAUT SECURITE PIANO ETAPE 0
B14:25 L 0
0013
GRAFCET DEFAUT SECURITE PIANO COPIE_ETAPE_26
B15:13 12
ACQUITTEMENT ACQUIT
B3:0 0
RETROUSSAGE POSITION BASSE
I:4 15 1746-IB32
GRAFCET DEFAUT SECURITE PIANO ETAPE_26
B14:13 U 12
GRAFCET DEFAUT SECURITE PIANO ETAPE 0
B14:25 L 0
0014
GRAFCET DEFAUT SECURITE PIANO ETAPE_25
B14:13 11
DEFAUT ABSENCE CAPTEUR PIANO POSITON BASSE 42SQ5
B18:0 12
0015
GRAFCET DEFAUT SECURITE PIANO ETAPE_26
B14:13 12
DEFAUT ABSENCE CAPTEUR RETROUSSAGE POSITION BASSE 42SQ2
B18:0 13

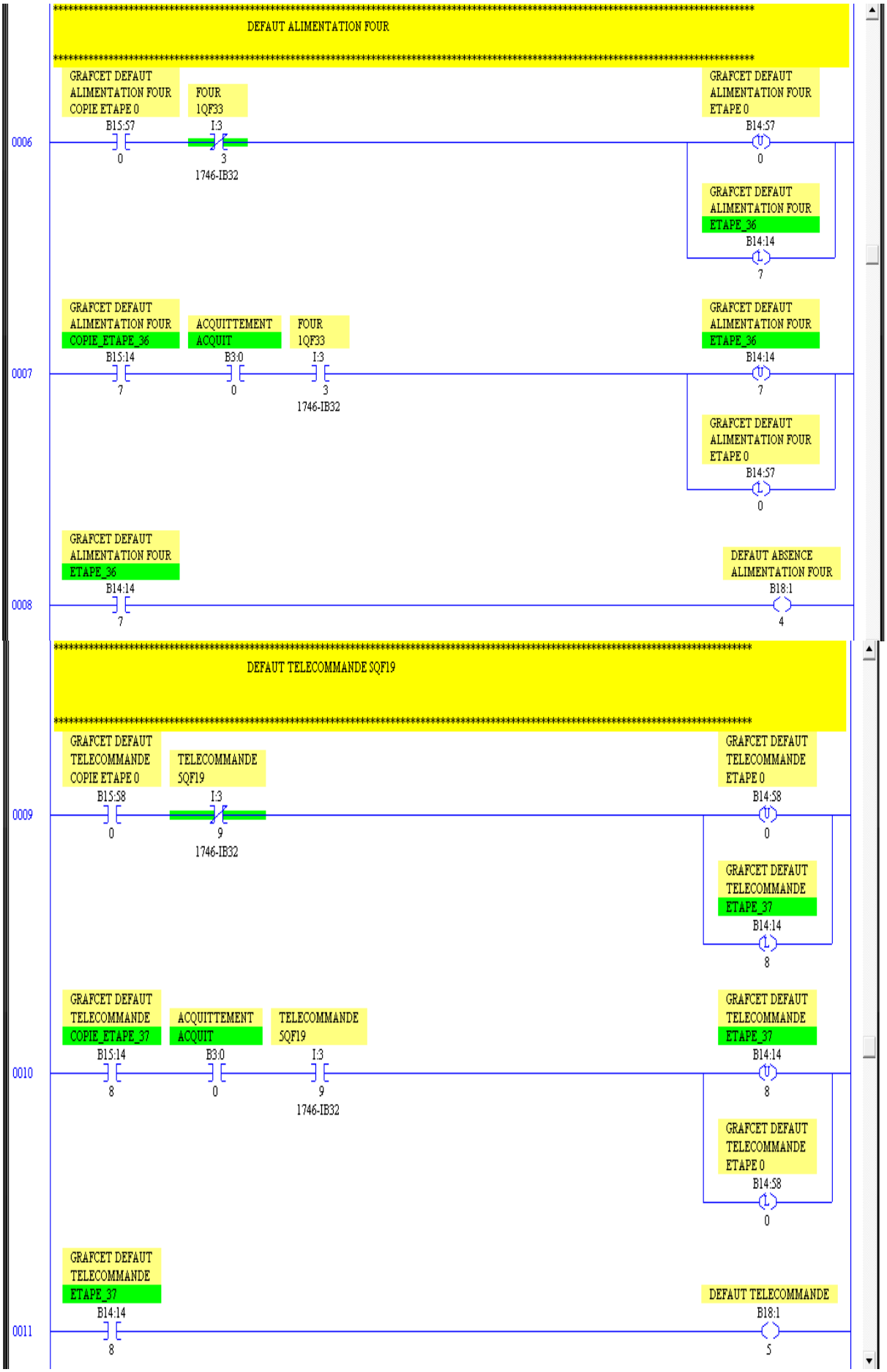

DEFAUT SECURITE RETROUSSAGE
0016
GRAFCET DEFAUT SECURITE RETROUSSAGE COPIE ETAPE 0
B15:26
0
DEMANDE MONTEE RETROUSSAGE
N12:0
1
OPER_SEQ/2
B14:30
2
DEMANDE MONTEE RETROUSSAGE SUR PV 550 ECRAN 8
B3:12
4
GRAFCET DEFAUT SECURITE RETROUSSAGE ETAPE 0
B14:26
0
GRAFCET DEFAUT SECURITE RETROUSSAGE ETAPE_27
B14:13
13
0017
GRAFCET DEFAUT SECURITE RETROUSSAGE COPIE_ETAPE_27
B15:13
13
TAMBOUR GONFLE 2SP
I:4
0
1746-IB32
GRAFCET DEFAUT SECURITE RETROUSSAGE ETAPE_27
B14:13
13
GRAFCET DEFAUT SECURITE RETROUSSAGE ETAPE_28
B14:13
14
0018
GRAFCET DEFAUT SECURITE RETROUSSAGE COPIE_ETAPE_27
B15:13
13
PIANO POSITION BASSE 42SQ5
I:4
24
1746-IB32
GRAFCET DEFAUT SECURITE RETROUSSAGE ETAPE_27
B14:13
13
GRAFCET DEFAUT SECURITE RETROUSSAGE ETAPE_29
B14:13
15
0019
GRAFCET DEFAUT SECURITE RETROUSSAGE COPIE_ETAPE_27
B15:13
13
TAMBOUR GONFLE 2SP
I:4
0
1746-IB32
PIANO POSITION BASSE 42SQ5
I:4
24
1746-IB32
GRAFCET DEFAUT SECURITE RETROUSSAGE ETAPE_27
B14:13
13
GRAFCET DEFAUT MONTEE TETE CHEVAL ETAPE 0
B14:12
0
0020
GRAFCET DEFAUT SECURITE RETROUSSAGE COPIE_ETAPE_28
B15:13
14
ACQUITTEMENT ACQUIT
B3:0
0
TAMBOUR GONFLE 2SP
I:4
0
1746-IB32
GRAFCET DEFAUT SECURITE RETROUSSAGE ETAPE_28
B14:13
14
GRAFCET DEFAUT SECURITE RETROUSSAGE ETAPE 0
B14:26
0
0021
GRAFCET DEFAUT SECURITE RETROUSSAGE COPIE_ETAPE_29
B15:13
15
ACQUITTEMENT ACQUIT
B3:0
0
PIANO POSITION BASSE 42SQ5
I:4
24
1746-IB32
GRAFCET DEFAUT SECURITE RETROUSSAGE ETAPE_29
B14:13
15
GRAFCET DEFAUT SECURITE RETROUSSAGE ETAPE 0
B14:26
0
0022
GRAFCET DEFAUT SECURITE RETROUSSAGE ETAPE_28
B14:13
14
DEFAUT ABSENCE CAPTEUR TAMBOUR GONFLE 2SP
B18:0
14
0023
GRAFCET DEFAUT SECURITE RETROUSSAGE ETAPE_29
B14:13
15
DEFAUT ABSENCE CAPTEUR PIANO POSITON BASSE 42SQ37
B18:0
15

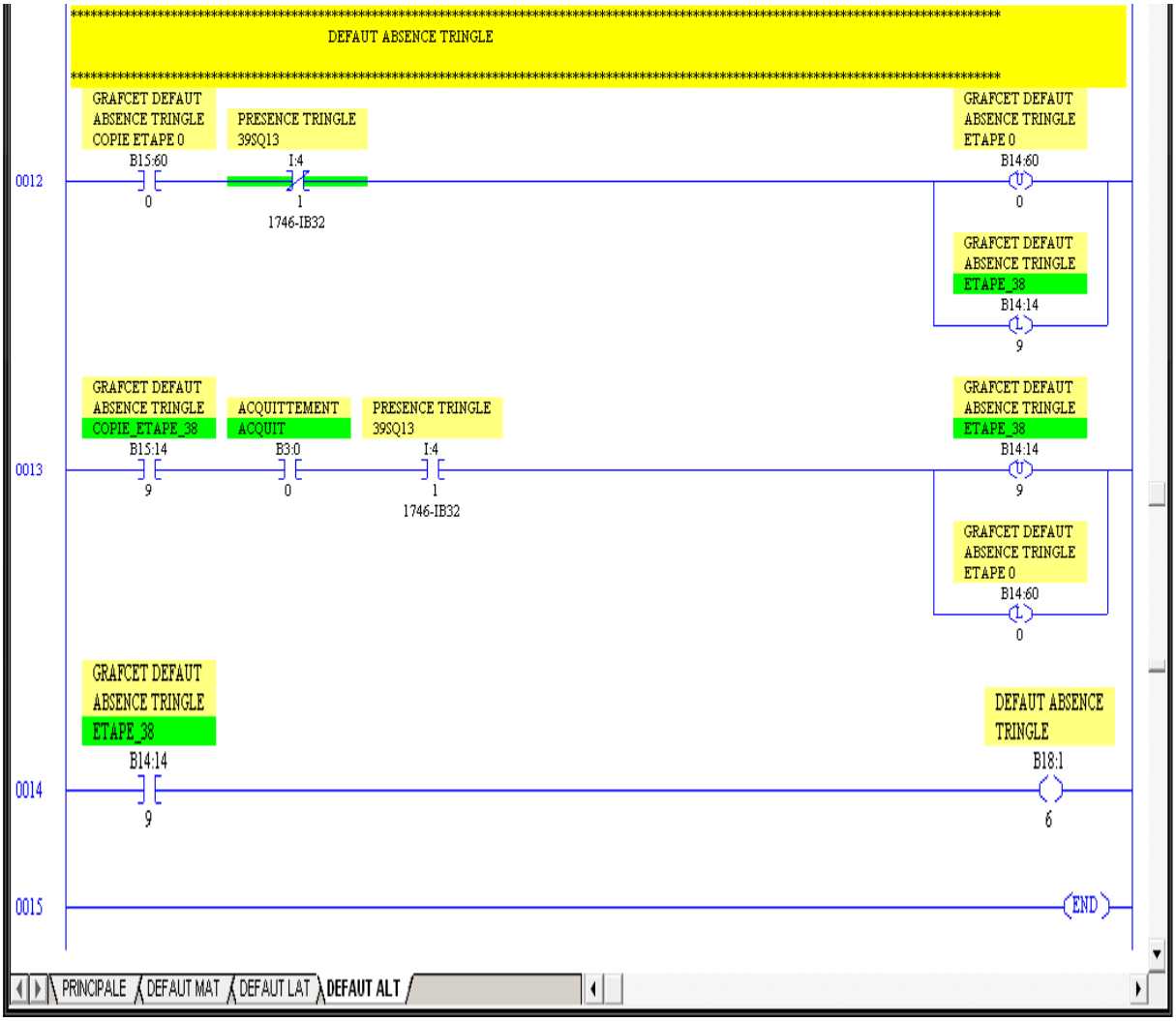

DEFAUT SECURITE TOURELLE
0024
GRAFCET DEFAUT SECURITE TOURELLE COPIE ETAPE 0
B15:27
0
DEMANDE DEVEROUILLAGE TOURELLE
N11:0
6
OPER_SEQ/2
B14:30
2
GRAFCET DEFAUT SECURITE TOURELLE ETPAE 0
B14:27
0
GRAFCET DEFAUT SECURITE TOURELLE ETAPE_30
B14:14
1
0025
GRAFCET DEFAUT SECURITE TOURELLE COPIE_ETAPE_30
B15:14
1
TETE CHEVAL POSITION BASSE 42SQ33
I:4
31
1746-IB32
GRAFCET DEFAUT SECURITE TOURELLE ETAPE_30
B14:14
1
GRAFCET DEFAUT SECURITE TOURELLE ETAPE_31
B14:14
2
0026
GRAFCET DEFAUT SECURITE TOURELLE COPIE_ETAPE_30
B15:14
1
TETE CHEVAL POSITION BASSE 42SQ33
I:4
31
1746-IB32
GRAFCET DEFAUT SECURITE TOURELLE ETAPE_30
B14:14
1
GRAFCET DEFAUT SECURITE TOURELLE ETPAE 0
B14:27
0
0027
GRAFCET DEFAUT SECURITE TOURELLE ETAPE_31
B14:14
2
ACQUITTEMENT ACQUIT
B3:0
0
TETE CHEVAL POSITION BASSE 42SQ33
I:4
31
1746-IB32
GRAFCET DEFAUT SECURITE TOURELLE ETAPE_31
B14:14
2
GRAFCET DEFAUT SECURITE TOURELLE ETPAE 0
B14:27
0
0028
GRAFCET DEFAUT SECURITE TOURELLE ETAPE_31
B14:14
2
DEFAUT ABSENCE CAPTEUR TETE CHEVAL POSITION BASSE 42SQ33
B18:1
0


DEFAUT SECURITE LANCE TABLE
GRAFCET DEFAUT SECURITE LANCE TABLE COPIE ETAPE 0
B15:28
0
DEMANDE DE DEVEROUILLAGE ET AVANCE TABLE
N11:0
5
OPER_SEQ/2
B14:30
2
GRAFCET DEFAUT SECURITE LANCE TABLE ETAPE 0
B14:28
0
GRAFCET DEFAUT SECURITE LANCE TABLE ETAPE_32
B14:14
3
0029
GRAFCET DEFAUT SECURITE LANCE TABLE COPIE_ETAPE_32
B15:14
3
TETE CHEVAL POSITION BASSE 42SQ33
I:4
31
1746-IB32
GRAFCET DEFAUT SECURITE LANCE TABLE ETAPE_32
B14:14
3
GRAFCET DEFAUT SECURITE LANCE TABLE ETAPE_33
B14:14
4
0030
GRAFCET DEFAUT SECURITE LANCE TABLE COPIE_ETAPE_32
B15:14
3
TETE CHEVAL POSITION BASSE 42SQ33
I:4
31
1746-IB32
GRAFCET DEFAUT SECURITE LANCE TABLE ETAPE_32
B14:14
3
GRAFCET DEFAUT SECURITE LANCE TABLE ETAPE 0
B14:28
0
0031
GRAFCET DEFAUT SECURITE LANCE TABLE COPIE_ETAPE_33
B15:14
4
ACQUITTEMENT ACQUIT
B3:0
0
TETE CHEVAL POSITION BASSE 42SQ33
I:4
31
1746-IB32
GRAFCET DEFAUT SECURITE LANCE TABLE ETAPE_33
B14:14
4
GRAFCET DEFAUT SECURITE LANCE TABLE ETAPE 0
B14:28
0
0032
GRAFCET DEFAUT SECURITE LANCE TABLE ETAPE_33
B14:14
4
DEFAUT ABSENCE CAPTEUR TETE CHEVAL POSITION BASSE 42SQ33
B18:1
1
0033
0034
END
PRINCIPALE
DEFAUT MAT
DEFAUT LAT

- **Programme pour la détection des défauts alimentation et défaut tringle :**




DEFAUT ALIMENTATION FOUR
GRAFCET DEFAUT ALIMENTATION FOUR COPIE ETAPE 0
B15:57
0
FOUR 1QF33
I:3
3
1746-IB32
GRAFCET DEFAUT ALIMENTATION FOUR ETAPE 0
B14:57
0
GRAFCET DEFAUT ALIMENTATION FOUR ETAPE_36
B14:14
7
0006
GRAFCET DEFAUT ALIMENTATION FOUR COPIE_ETAPE_36
B15:14
7
ACQUITTEMENT ACQUIT
B3:0
0
FOUR 1QF33
I:3
3
1746-IB32
GRAFCET DEFAUT ALIMENTATION FOUR ETAPE_36
B14:14
7
GRAFCET DEFAUT ALIMENTATION FOUR ETAPE 0
B14:57
0
0007
GRAFCET DEFAUT ALIMENTATION FOUR ETAPE_36
B14:14
7
DEFAUT ABSENCE ALIMENTATION FOUR
B18:1
4
0008
DEFAUT TELECOMMANDE SQF19
GRAFCET DEFAUT TELECOMMANDE COPIE ETAPE 0
B15:58
0
TELECOMMANDE SQF19
I:3
9
1746-IB32
GRAFCET DEFAUT TELECOMMANDE ETAPE 0
B14:58
0
GRAFCET DEFAUT TELECOMMANDE ETAPE_37
B14:14
8
0009
GRAFCET DEFAUT TELECOMMANDE COPIE_ETAPE_37
B15:14
8
ACQUITTEMENT ACQUIT
B3:0
0
TELECOMMANDE SQF19
I:3
9
1746-IB32
GRAFCET DEFAUT TELECOMMANDE ETAPE_37
B14:14
8
GRAFCET DEFAUT TELECOMMANDE ETAPE 0
B14:58
0
0010
GRAFCET DEFAUT TELECOMMANDE ETAPE_37
B14:14
8
DEFAUT TELECOMMANDE
B18:1
5
0011


DEFAUT ABSENCE TRINGLE
GRAFCET DEFAUT ABSENCE TRINGLE COPIE ETAPE 0
B15:60
0
PRESENCE TRINGLE 39SQ13
I:4
1
1746-IB32
GRAFCET DEFAUT ABSENCE TRINGLE ETAPE 0
B14:60
0
GRAFCET DEFAUT ABSENCE TRINGLE ETAPE_38
B14:14
9
0012
GRAFCET DEFAUT ABSENCE TRINGLE COPIE_ETAPE_38
B15:14
9
ACQUITTEMENT ACQUIT
B3:0
0
PRESENCE TRINGLE 39SQ13
I:4
1
1746-IB32
GRAFCET DEFAUT ABSENCE TRINGLE ETAPE_38
B14:14
9
GRAFCET DEFAUT ABSENCE TRINGLE ETAPE 0
B14:60
0
0013
GRAFCET DEFAUT ABSENCE TRINGLE ETAPE_38
B14:14
9
DEFAUT ABSENCE TRINGLE
B18:1
6
0014
END
0015
PRINCIPALE
DEFAUT MAT
DEFAUT LAT
DEFAUT ALT

Bibliographie

-Bibliographie-

Livres :

Chevalier, Guide du dessinateur industriel (www.hachette-education.com).

Buissoux, J. Montagnac, Cours de schémas Automatisme-Electricité 1984.

Publications :

SLC 500™ Power Supplies, Publication 1746-IN004A-ML-P, janvier 2000.

Automates programmables SLC 500, Publication 1747-BR017A-FR-P, Août 2004.

PLC-5 programmable controllers, Publication 1785-SG001B-EN-P, Juin 2006.

Automates programmables PLC-5, Publication 1785-10.2FR, Avril 1996.

Processeurs SLC 500 sur châssis, Publication 1747-2.39FR, Décembre 1997.

La famille des automates programmables SLC 500, Publication 1747-2.30FR, Janvier 1995

Automates programmables MicroLogix 1000, Publication 1761-TD001B-FR-P, Mai 2003.

MicroLogix Programmable Controllers, Publication 1761-SG001C-EN-P, Juillet2008.

Logix5000, Publication 1756-RM094A-FR-P, Mai 2004.

ControlLogix Selection Guide 1756-SG001K-EN-P, Avril 2008.

Module Pico™ Publication 1760-UM001B-FR-P, Juillet 2000.

RSLogix 500TM, programming for the SLC500TM and MicroLogixTM families

Technique de l'ingénieur : LE GRAFCET, Langage de programmation pour API

Norme IEC 1131-3.

Biblio : help RSLogix 500.

Webliographie

-Webliographie-

www.rockwellautomation.fr

www.rockwellautomation.com

www.ab.com