

المسيلة في: 19 نوفمبر 2023

الرقم 319/2023

مستخلص اللجنة العلمية للقسم بتاريخ: 2023-11-14

بخصوص مطبوعة الدروس

بخصوص مطبوعة الأعمال التطبيقية المنجزة من طرف الأستاذ عبدو عبد الحق أستاذ محاضر قسم "أ" بجامعة باتنة تحت عنوان: «: TP API et supervision » فقد اطلعت اللجنة على التقارير الواردة من طرف لجنة الخبراء المكونة من الأستاذ بوقرة عبد الرحمان أستاذ بجامعة محمد بوضياف بالمسيلة ، الأستاذ غلاب محمد زين العابدين أستاذ محاضر "أ" بجامعة محمد بوضياف بالمسيلة و الأستاذ لعماري يحيي أستاذ محاضر -أ- بجامعة مصطفى بن بولعيد باتنة 2 والتي كانت كلها ايجابية، لهذا فان اللجنة لا ترى مانعا أن تتخذ سنداً في تدريس طلبة السنة الأولى ماستر آلية وأنظمة في ميدان علوم و تكنولوجيا و أن تعتمد في أي تقييم للمسار العلمي للأستاذ المعني.

رئيس اللجنة العلمية

أ.د بوقرة عبد الرحمان



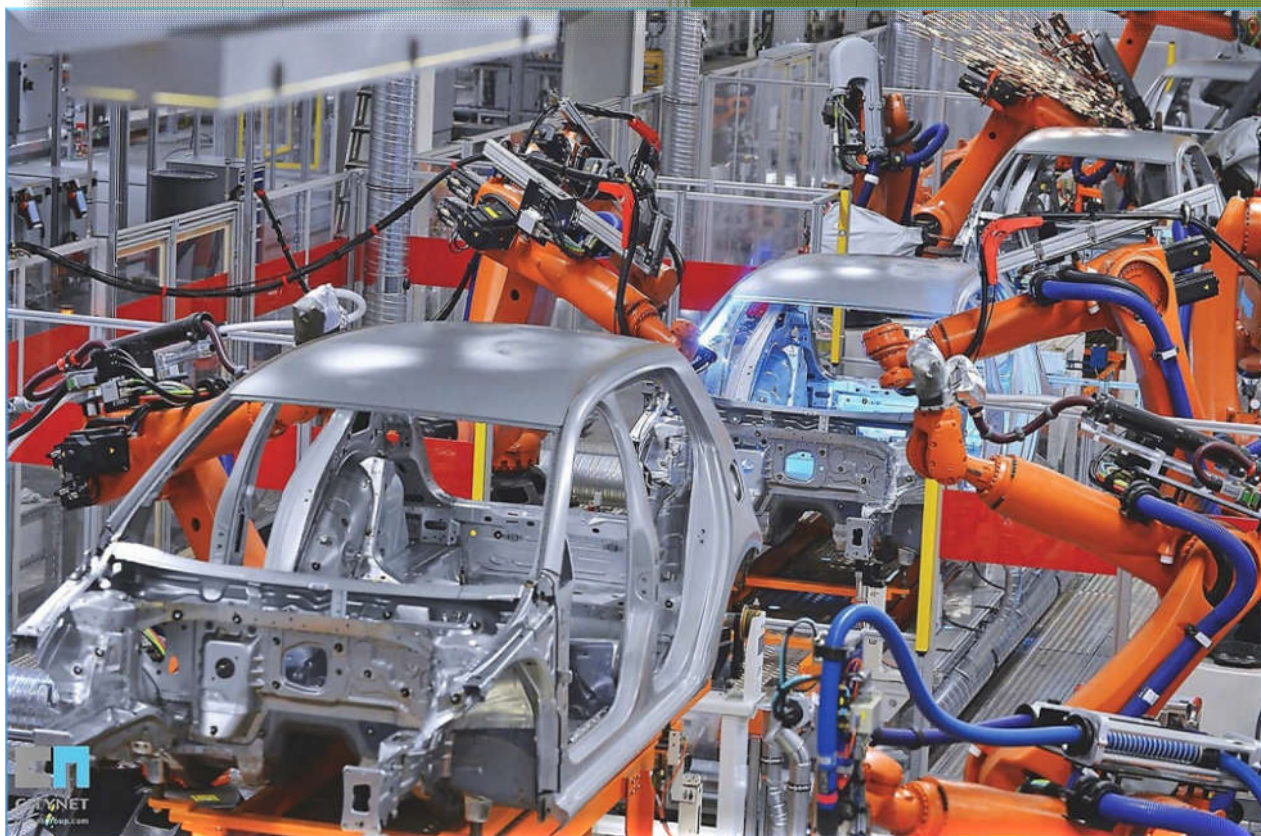


Université Mohamed BOUDIAF - M'sila

Faculté de Technologie
Département de Génie Electrique



Support pédagogique
TP AUTOMATE PROGRAMMABLE INDUSTRIEL
ET SUPERVISION



Préparé par :
Dr . Abdelhak ABDOU

Polycopie de Cours Expertisé par :

Nom & prénom	Grade	Etablissement
LAAMARI Yahia	MCA	Université Batna 2
BOUGUERRA Abderrahmen	Prof	Université de Msila
GHELLAB Mohamed Zinelabidine	MCA	Université de Msila

*Support pédagogique de TP destiné aux étudiants :
1^{ère} année Master Automatique
Spécialité : Automatique et Système*

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique



Université Mohamed BOUDIAF - M'sila



Faculté de Technologie
Département de Génie Electrique

Support Pédagogique

Polycopie

TP API et Supervision

*Support pédagogique de TP destiné aux étudiants :
1^{ère} année Master Automatique
Spécialité : Automatique et Système*

Préparé par :

Dr Abdelhak ABDYOU

abdelhak.abdou@univ-msila.dz



*Ce document constitue un support pédagogique
et ne prétend donc ni à l'originalité, ni à l'exhaustivité.
Ces notes doivent beaucoup aux emprunts faits à de nombreux ouvrages et à
différents travaux de collègues.*

Contenu de la matière

Semestre: 2

Unité d'enseignement: UEM 1.2

Matière: TP API et supervision

VHS: 22h30 (TP: 1h30)

Crédits: 2

Coefficient: 1

Objectifs de l'enseignement:

- Ce TP permettra à l'étudiant la mise en pratique et la consolidation des connaissances acquises dans le module API et supervision.

Connaissances préalables recommandées:

- Cours de la matière API et Supervision.

Contenu de la matière:

Prévoir quelques TPs en relation avec le matériel disponible.

Mode d'évaluation:

- Contrôle continu: 100 %.

Table des matières

Contenu de la matière	i
Table des matières	ii
Avant propos	iii
INITIATION AU LOGICIEL MILLENIUM 3	1
Introduction.....	1
CREATION D'UN NOUVEAU DOCUMENT DE TRAVAIL	2
PROGRAMMATION DE BASE	6
EXERCICES D'APPLICATION 1 : CIRCUIT LOGIQUE	14
INITIATION AU LOGICIEL PL7-PRO	15
I.1 Introduction	15
I.2 Ouverture d'une session	15
I.2.1 Créer un nouveau projet	15
I.2.2 Sélectionner la version d'automate que vous utilisé	16
I.2.3 Ouverture « Navigateur Application »	17
I.2.4 Programmation d'une application	18
2. Exemples et problèmes de programmations	26
TP N° 2 :PROGRAMMATION EN LANGAGE LD, IL et ST	33
2. Exercices d'applications industrielles	33
Exercice 1 : Démarrage d'un moteur asynchrone	33
TP N° 3 : MINI AUTOMATE MILLENIUM 4	39
Disposition de la partie opérative :	40
Tableau des entrées. / sorties :	40
Programmation de l'automate Millénium :	41
Schéma et câblage du module avec son environnement.....	41
Les entrées et les sorties:.....	41
Solutions des Exemples et problèmes.....	42
I.2.1 Exemples	42
I.2.2 Problèmes	47
TP N° 5 :	51
Démarrage un seul sens de rotation.....	51
Travail demandé :	53
TP N° 6 : 58	
Solution TP 6 : Etude et commande par API d'un appareil à plier les tôles	63
Références Bibliographiques.....	67

Avant propos

Description générale

Ce cours est destiné aux étudiants de niveau Master 2 Automatique option Automatique et système semestre 2. Il s'adresse également aux personnes voulant s'initier ou avoir un aperçu général sur systèmes automatisés commandés par API.

Objectifs

Ce support pédagogique de travaux pratique TP à pour objectif de :

- faciliter la tâche des étudiants qui entrent en contact avec les automates pour la première fois.
- L'objectif du programme est de soumettre aux étudiants un ensemble de connaissances indispensables et nécessaires pour la compréhension des systèmes automatisés commandés par API.
- Identification des composantes d'un système automatisé.

Connaissances préalables recommandées :

- Circuits logiques
- automatisme industriel
- Cours de la matière API et Supervision
- Schémas et rôles des composants automatisme industriel

INITIATION AU LOGICIEL MILLENIUM 3

API CROUZET

INTRODUCTION

Le logiciel Millenium 3 sert à programmer l'Automate Programmable Industriel (API) Crouzet.



On nomme **Automate Programmable Industriel**, (**API**, en anglais **Programmable Logic Controller**, **PLC**) un dispositif similaire à un ordinateur, utilisé pour automatiser des processus comme la commande des machines sur une chaîne de montage dans une usine.

Là où les systèmes automatisés plus anciens emploieraient des centaines ou des milliers de relais et de cames, un simple automate suffit. On nomme automaticiens les programmeurs de ces Automates Programmables Industriels.

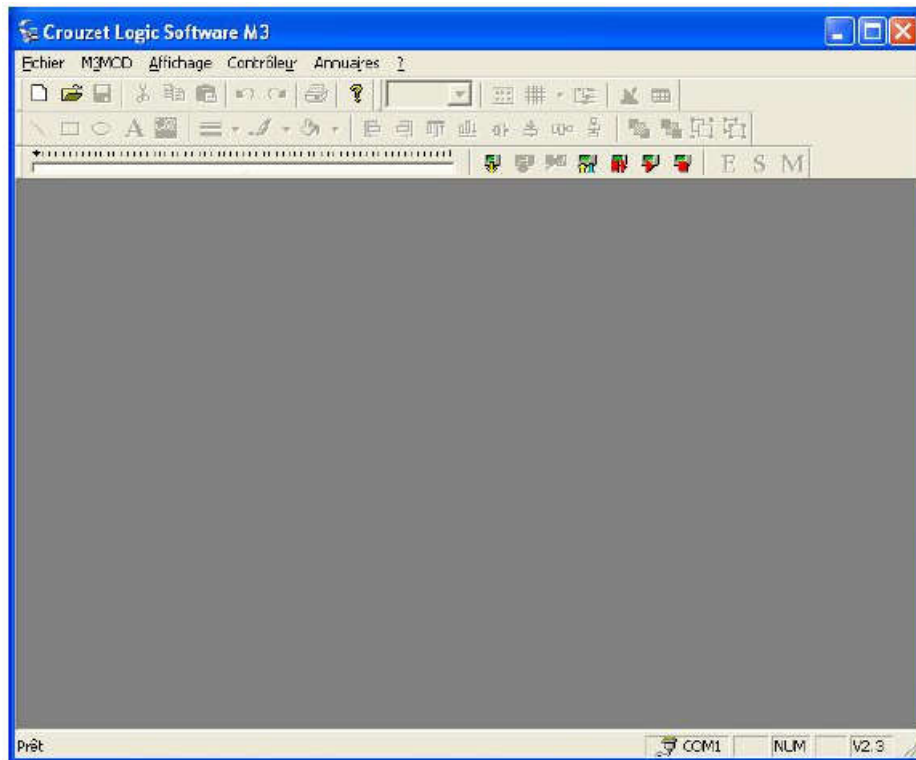
L'API est structuré autour d'une unité de calcul ou processeur (en anglais Central Processing Unit, CPU), d'une alimentation (depuis des tensions alternatives - AC ou continues - DC) et, de modules suivant les besoins de l'application, tel que:

- des cartes d'entrées - sorties (en anglais **Input - Output**, **I/O**) numériques (Tout ou rien)
- ou analogiques
- des cartes d'entrées pour brancher des capteurs (un capteur est un dispositif qui transforme l'état d'une grandeur physique observée en une grandeur utilisable,...), boutons poussoirs, ...
- des cartes de sorties pour brancher des actionneurs, voyants, vannes, ...
- des modules de communication pour dialoguer avec d'autres automates, des entrées/sorties déportées, des supervisions ou autres interfaces homme-machine
- des modules dédiés métiers, tels que de comptage rapide, de pesage...
- des modules d'interface pour la commande de mouvement, dits modules **Motion**, tels que démarreurs progressifs, variateur de vitesse, commande d'axes.

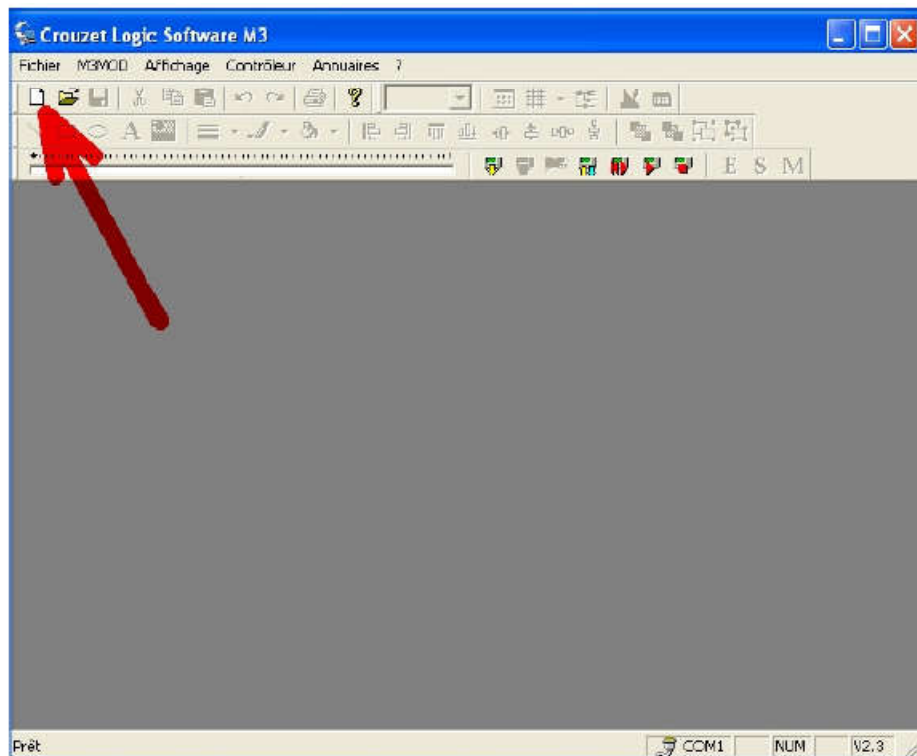
Nous allons étudier le fonctionnement du logiciel de programmation de cet automate. Une fois le programme élaboré et testé sur l'ordinateur, il pourra être transféré dans l'automate pour y être exploité.

CREATION D'UN NOUVEAU DOCUMENT DE TRAVAIL

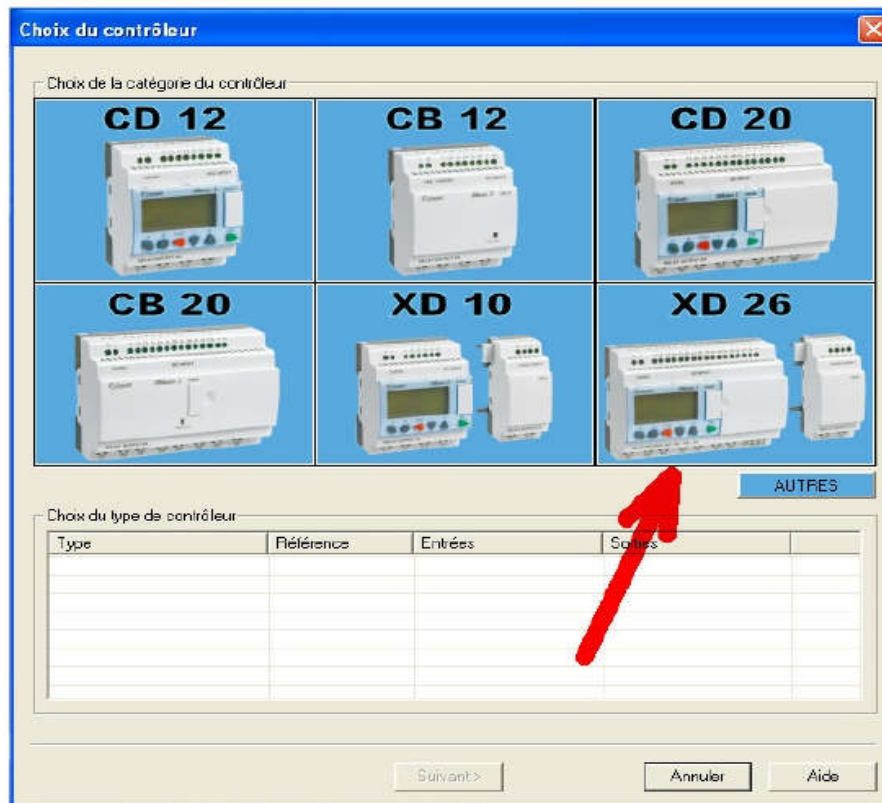
- a. Lancez le logiciel "Millenium 3 AC".



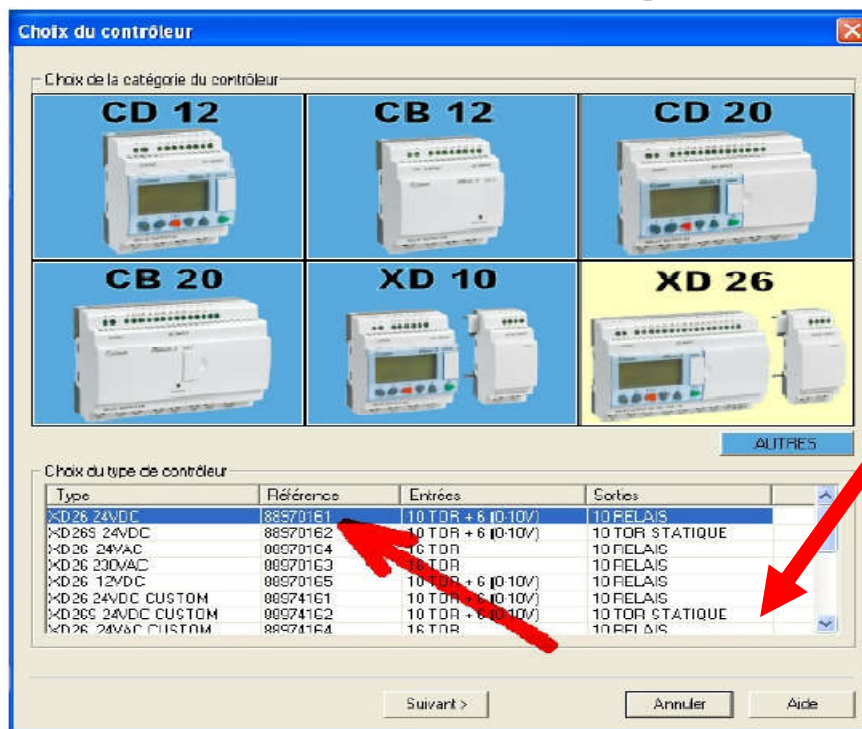
- b. Cliquez sur "Nouveau document".



- c. Vous devez choisir le type d'Automate : Exemple : cliquez sur XD26.



- d. Précisez la référence exacte de l'Automate : Exemple : choisissez "88 970 161".

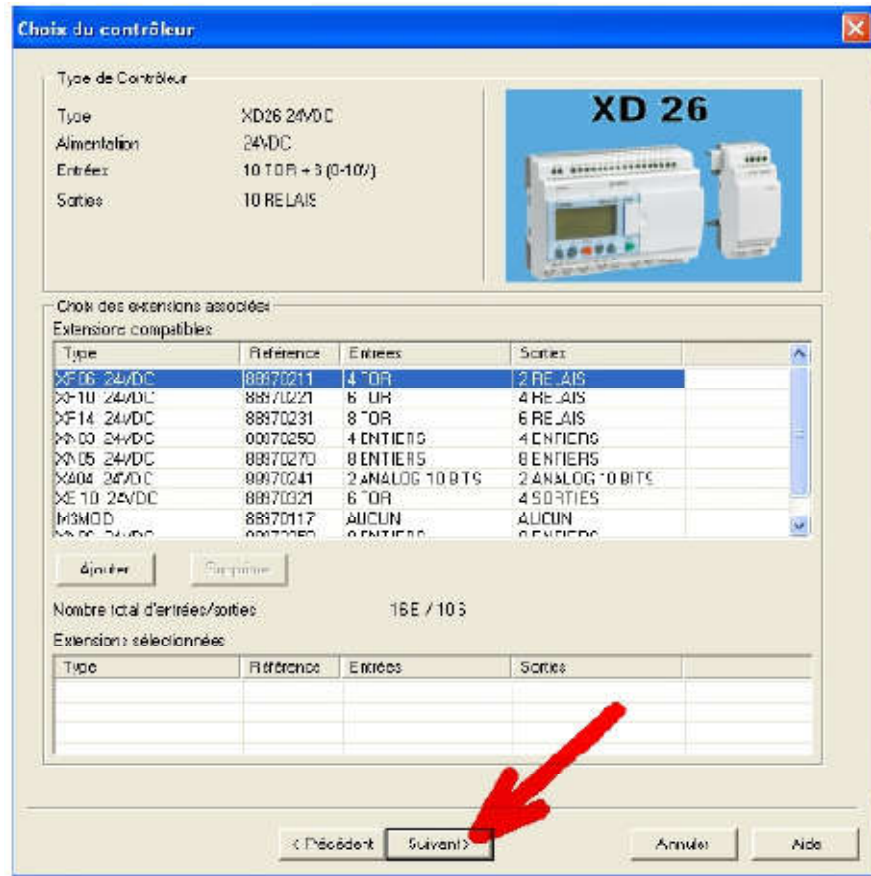


Ce type est caractérisé par :

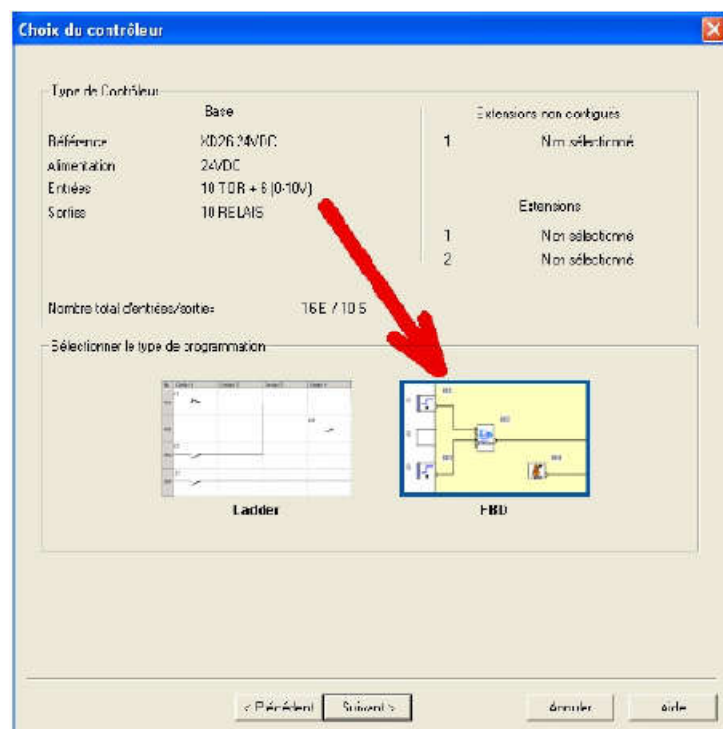
- 10 entrées Tout Ou Rien (TOR) et 6 entrées analogiques,
- 10 sorties Tout Ou Rien (TOR).

- e. Il est possible d'ajouter des extensions sur l'Automatique (communication, entrées, sorties, etc.). Pour cette application, nul besoin d'ajouter d'extension : cliquez sur "Suivant".

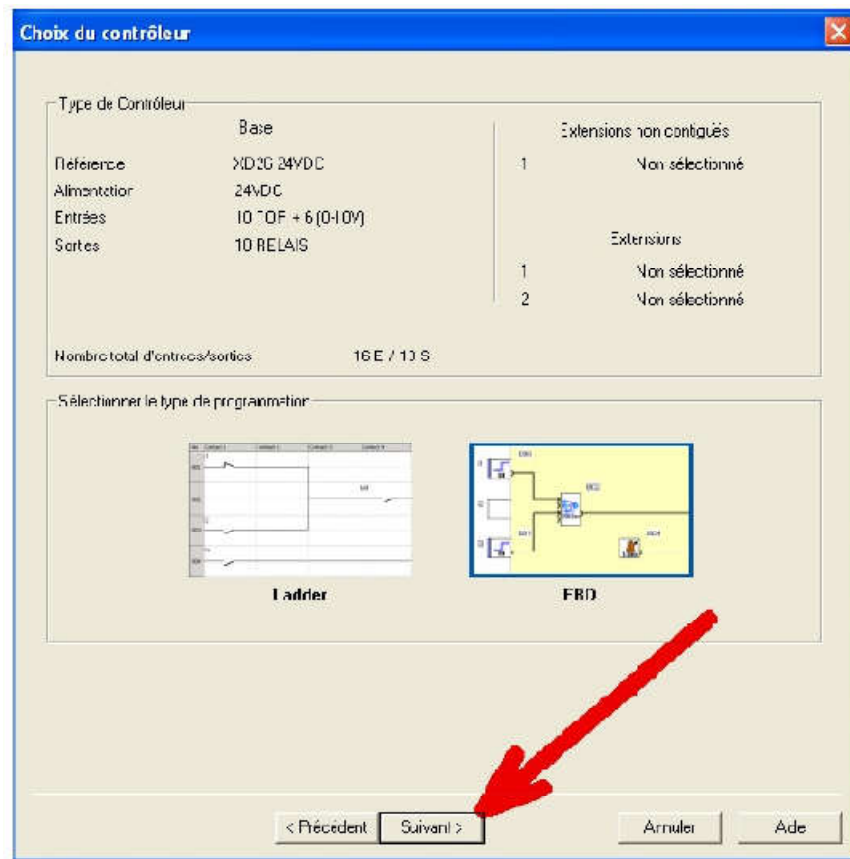
f.



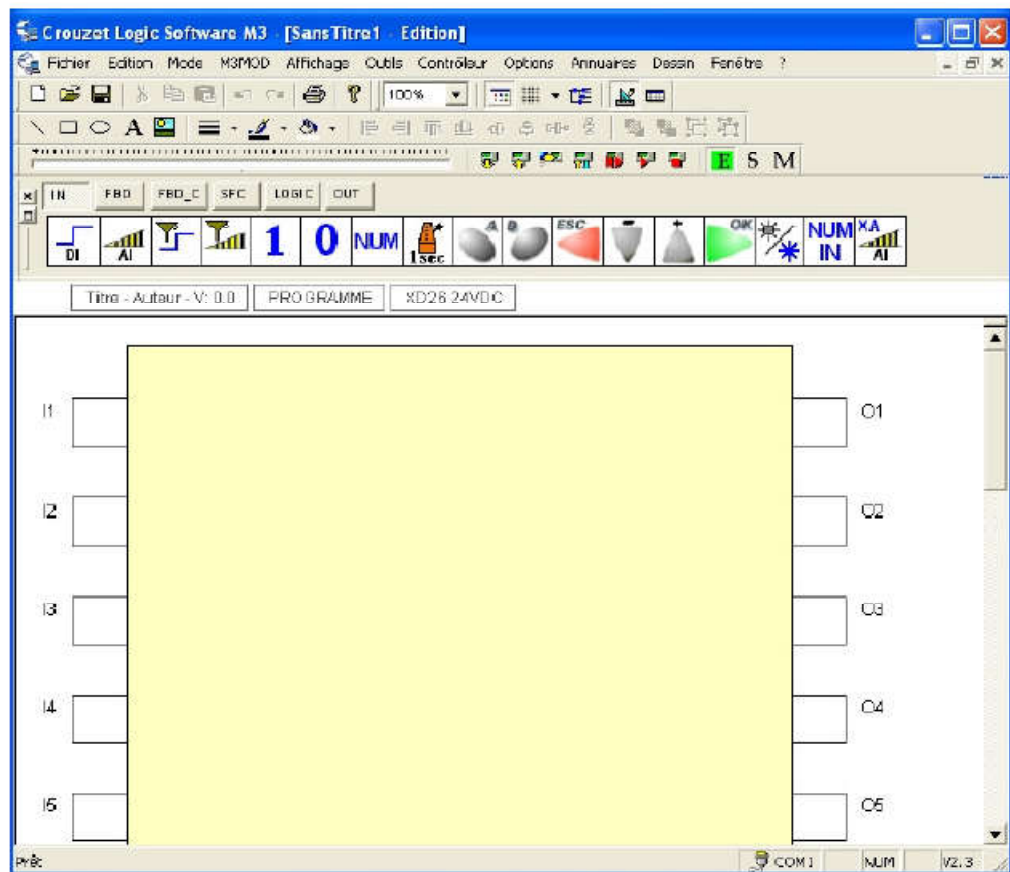
g. Sélectionnez le type de langage de programmation - langage à contact (Ladder) ou FBD (Functional Block Diagram). Pour cette initiation, cliquez sur "FBD".



h. Validez ce choix en cliquant sur "Suivant".

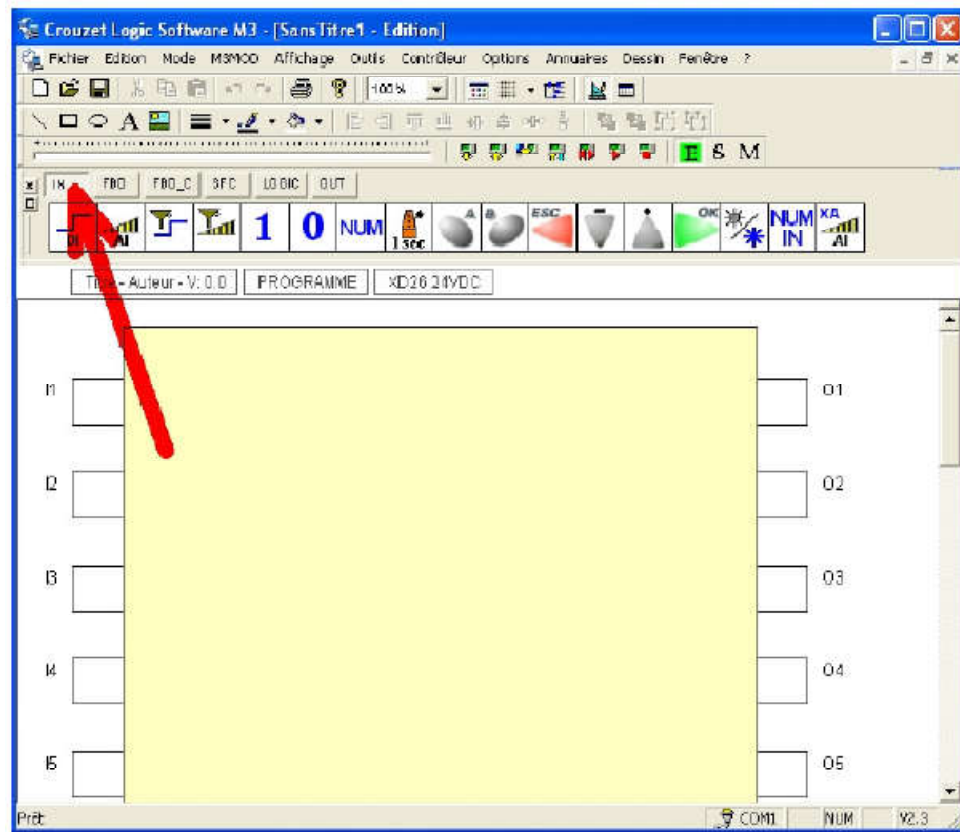


i. Voici le nouveau document configuré pour votre automate.

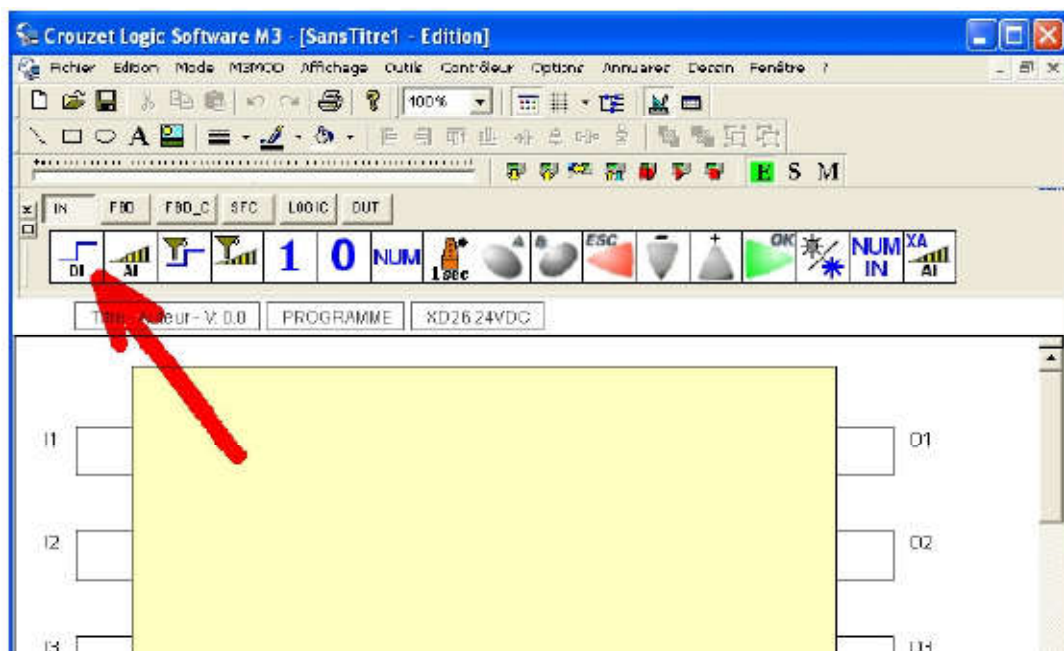


PROGRAMMATION DE BASE

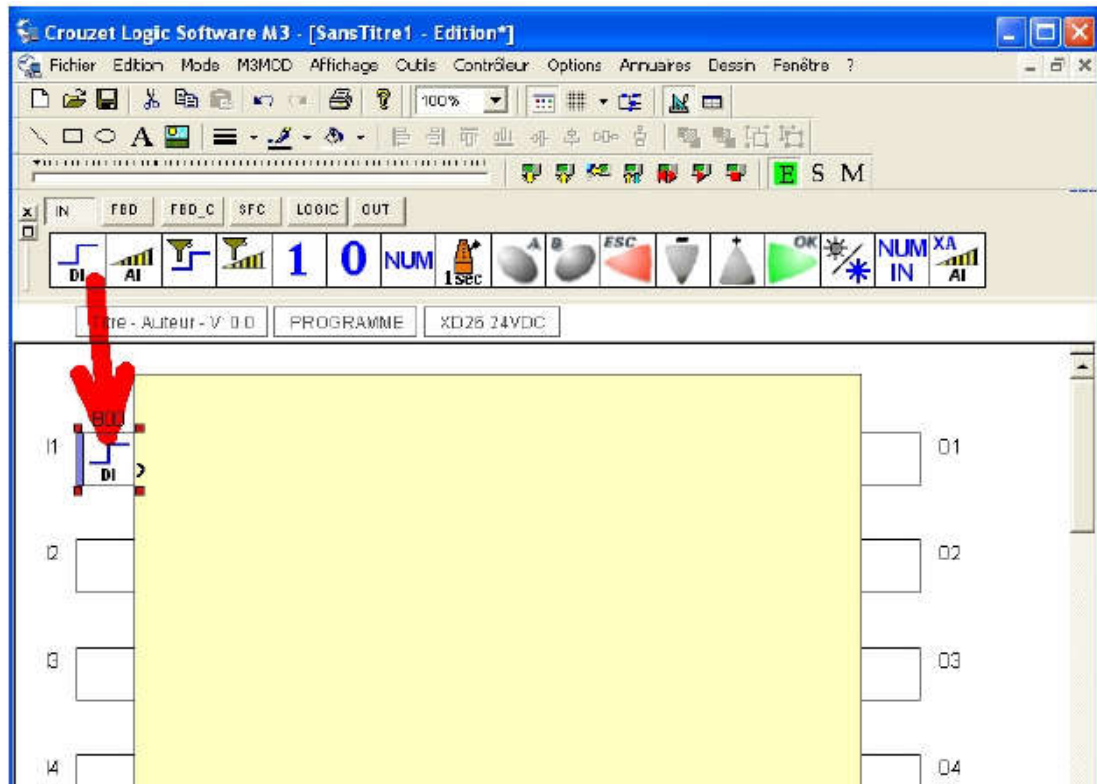
- a. Une barre d'outils nommée "Barre de fonctions" permet d'insérer dans le document des éléments d'entrées (IN), de fonctions (FBD), de grafcet (SFC), de logique (LOGIC) ou de sorties (OUT). Choisissez pour commencer le menu de fonctions "IN".



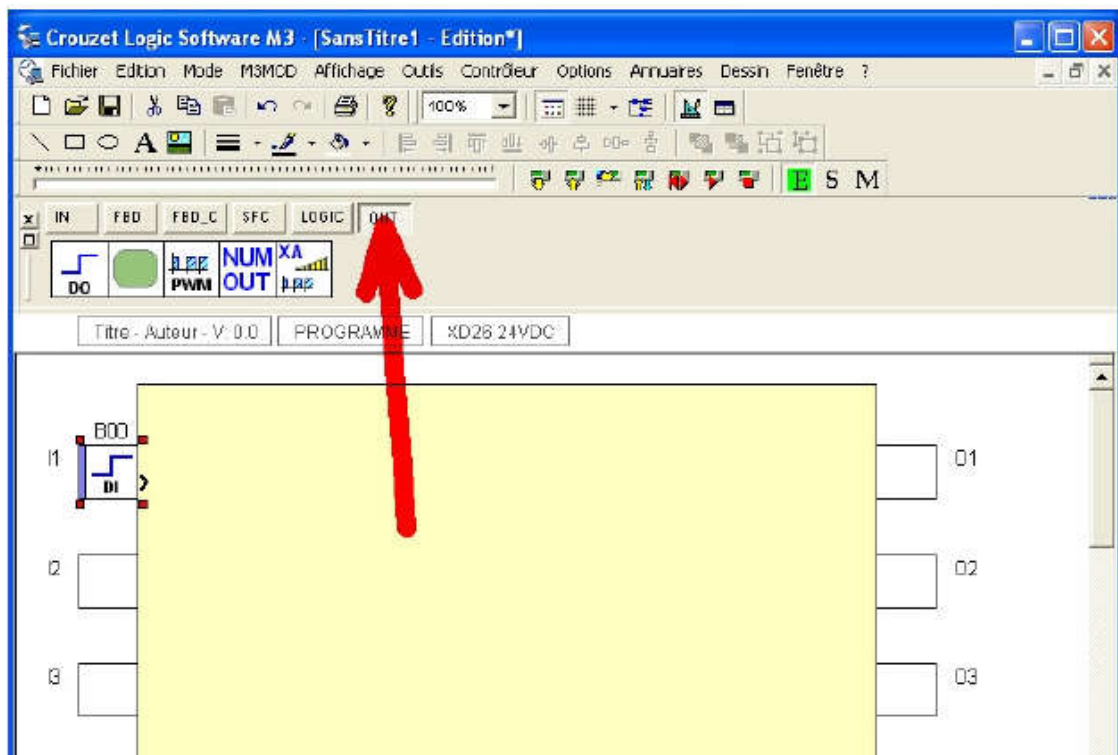
- b. Nous allons placer une entrée binaire (de type Tout Ou Rien - TOR) sur notre document : elle se nomme "DI" (Digital Input).



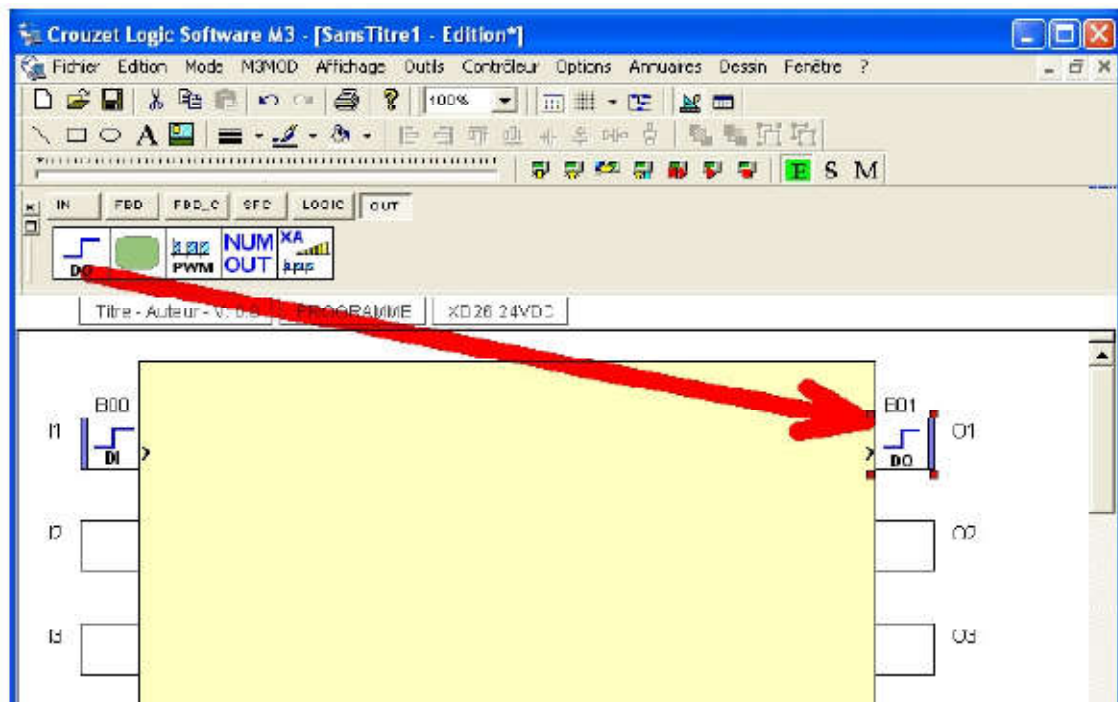
- c. Faites glisser l'icône sur une des bornes d'entrées de l'automate (I1 à IA).



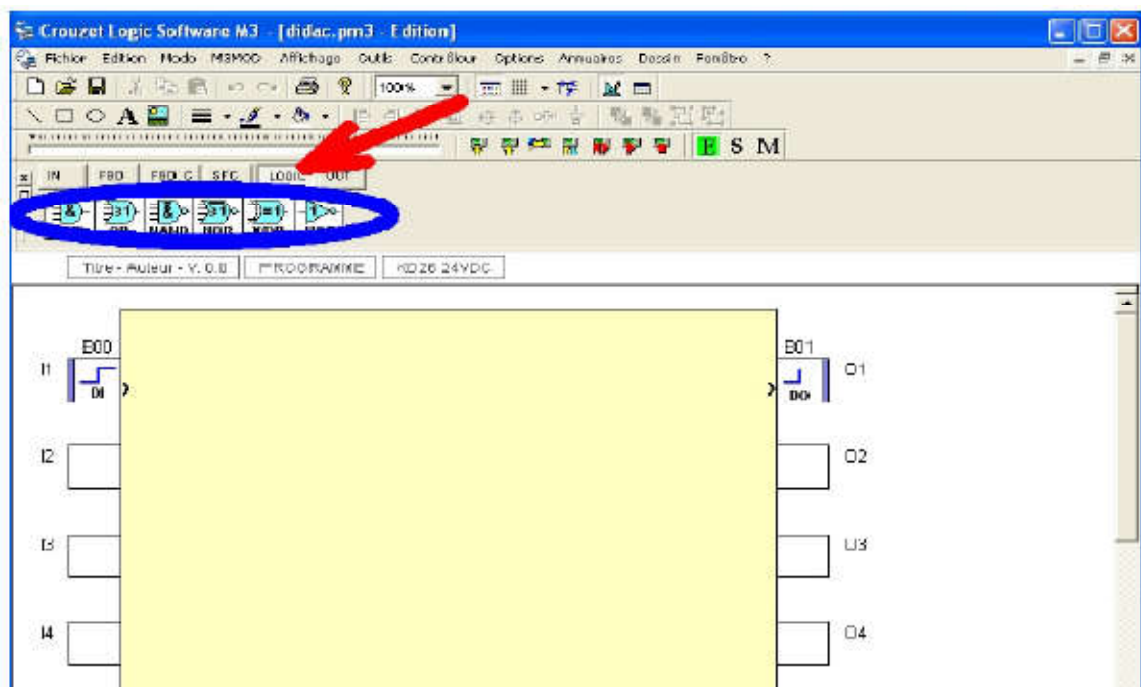
d. Choisissez le menu de fonctions "OUT" (sorties).



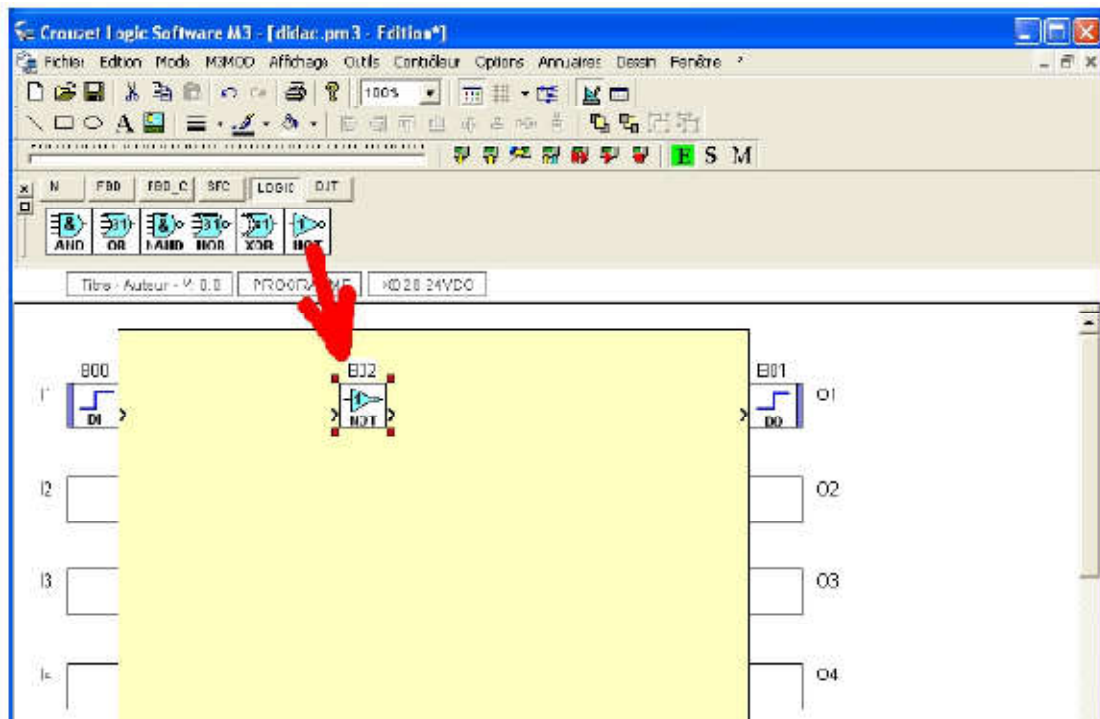
- e. Faites glisser l'icône de sortie "DO" (Digital Output) sur une des bornes de sortie de l'automate (O1 à OA).



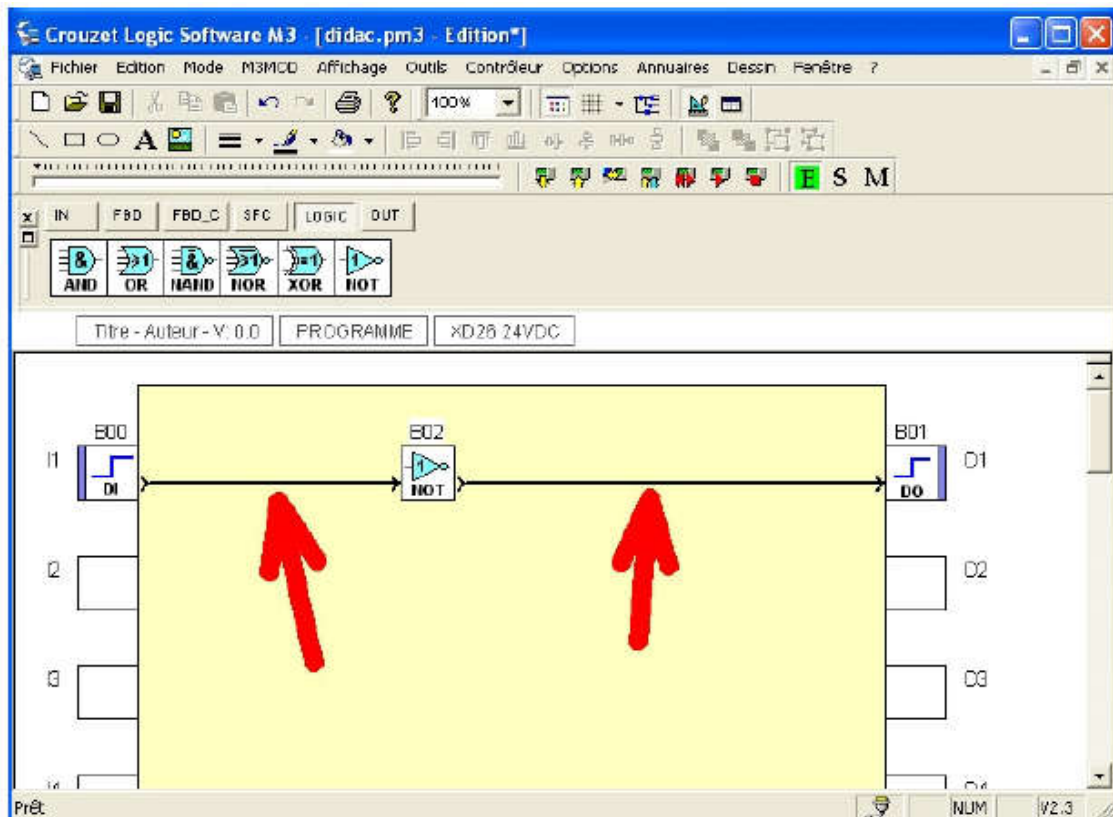
- f. Choisissez le menu de fonctions "LOGIC".



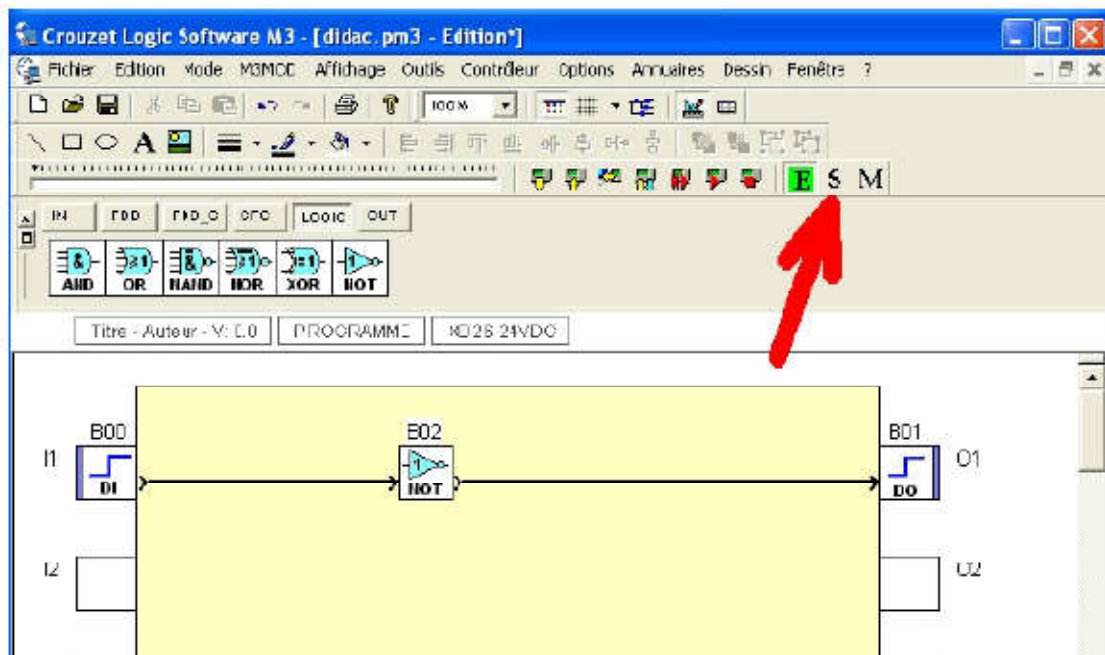
- g. Faites glisser l'icône de l'inverseur logique (NOT) sur l'espace de travail.



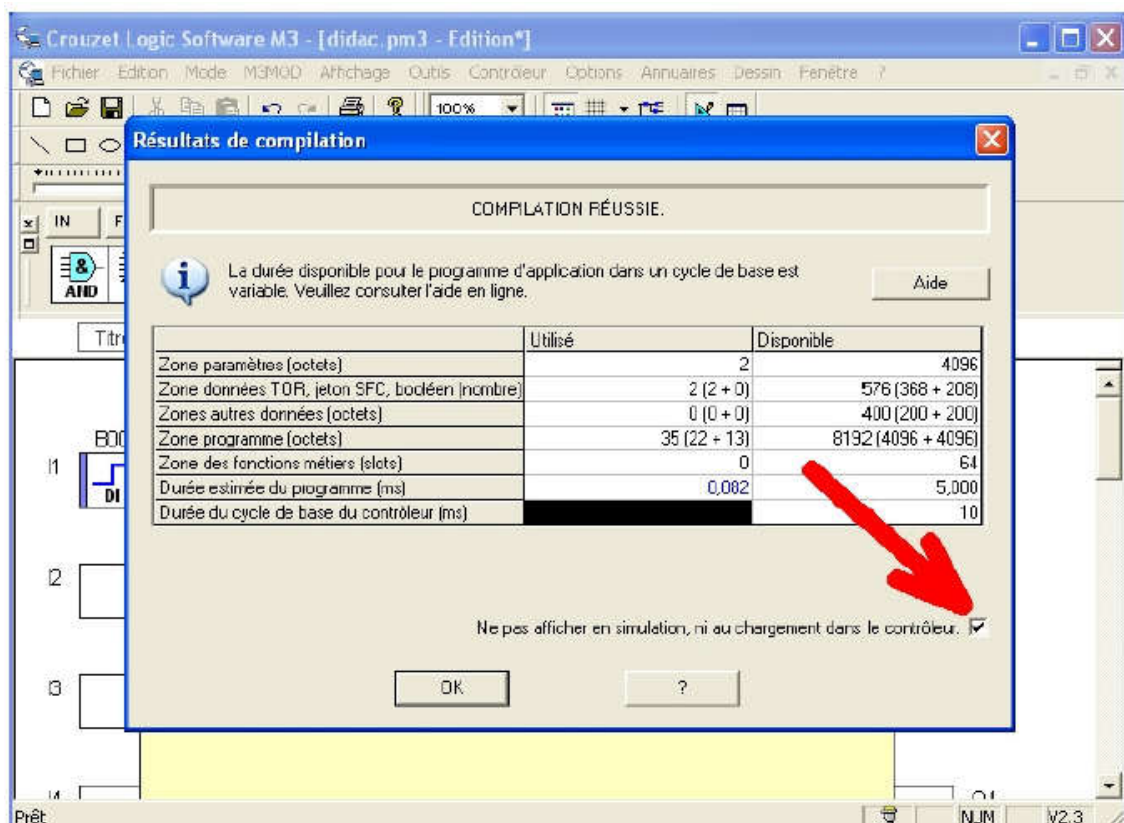
- h. Effectuez les liaisons en cliquant-déplaçant votre souris depuis les bornes d'entrée/sortie vers les connexions de l'inverseur logique.



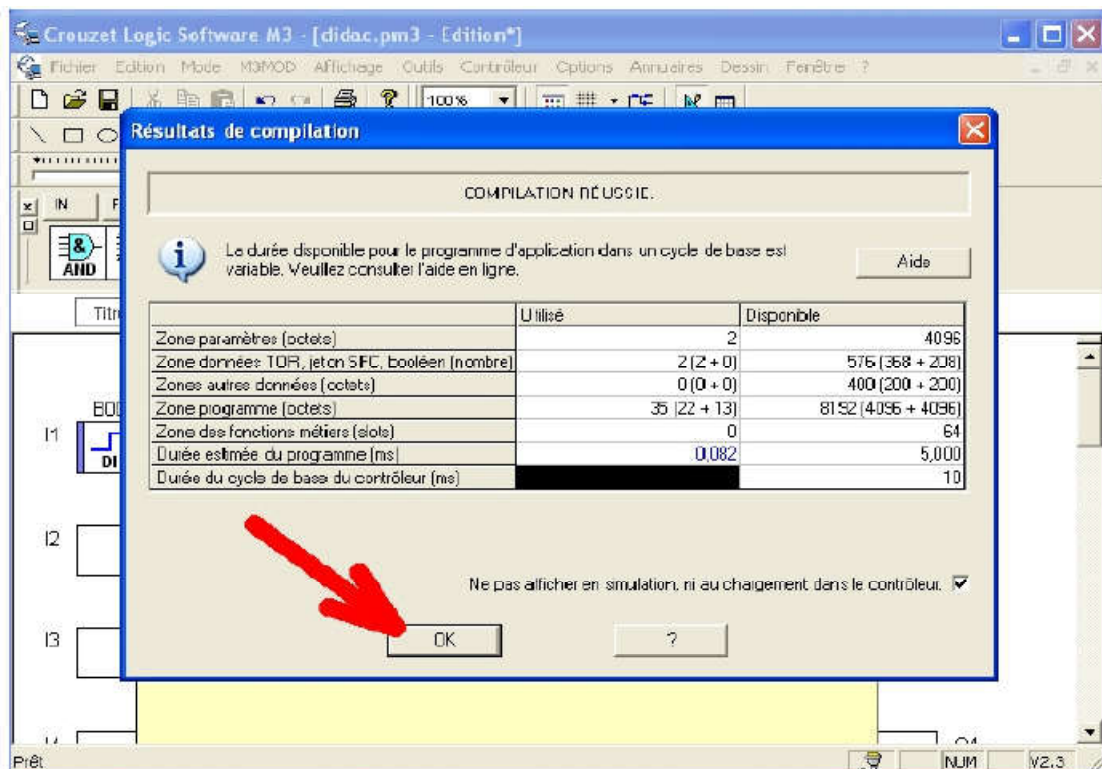
- i. Votre circuit de base est terminé.. Il faut maintenant vérifier son fonctionnement à l'aide du simulateur (bouton "S").



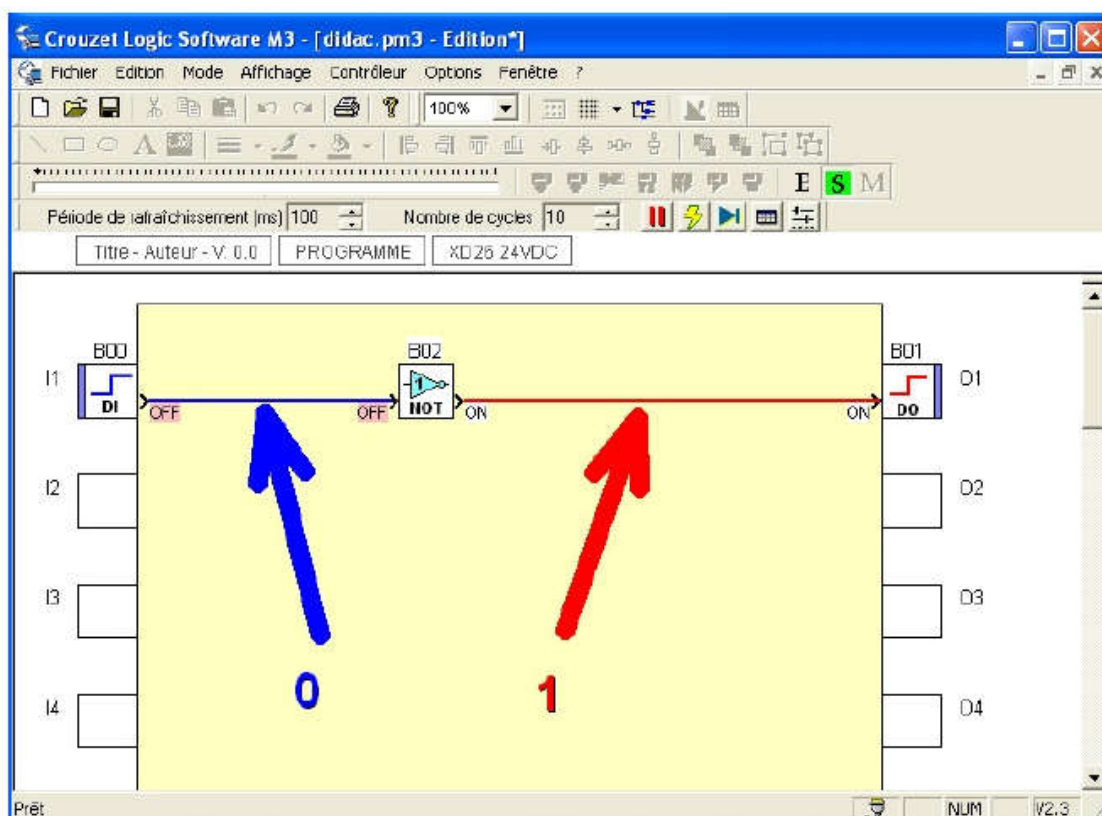
- j. Lorsque vous lancez la Simulation, il se peut que le résultat de la compilation du programme apparaisse : il atteste que le programme écrit a été correctement "traduit" pour être exploité par le simulateur. Si vous ne voulez plus voir cette fenêtre, cochez l'option correspondante :



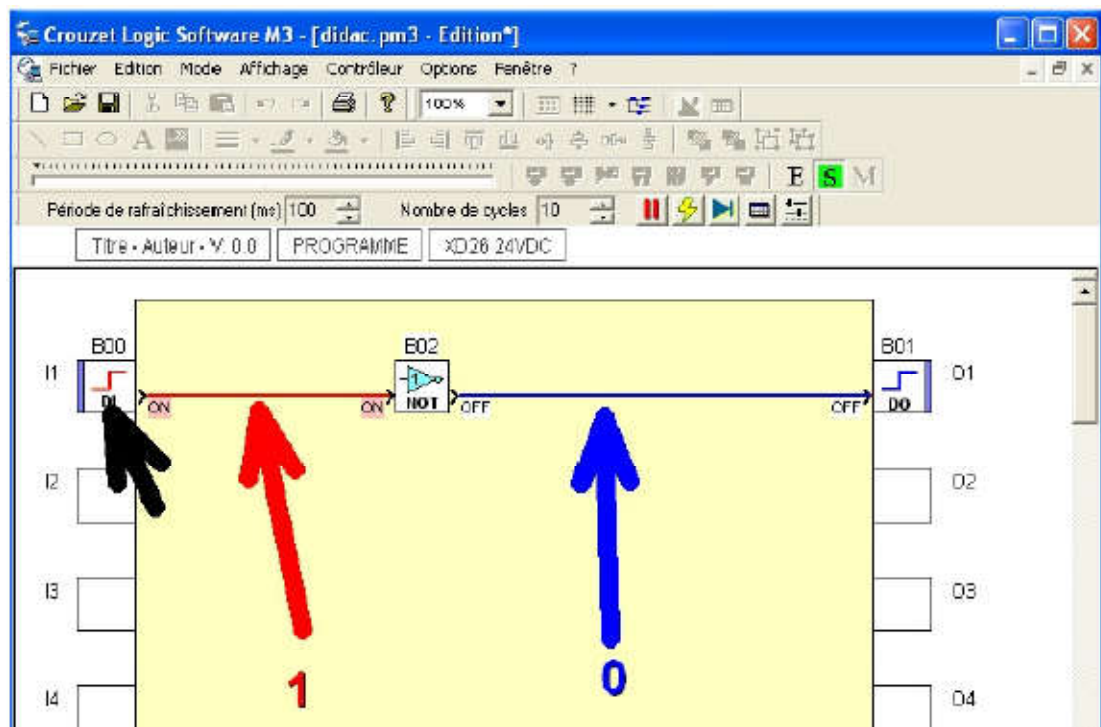
- k. Validez ensuite en cliquant sur "OK".



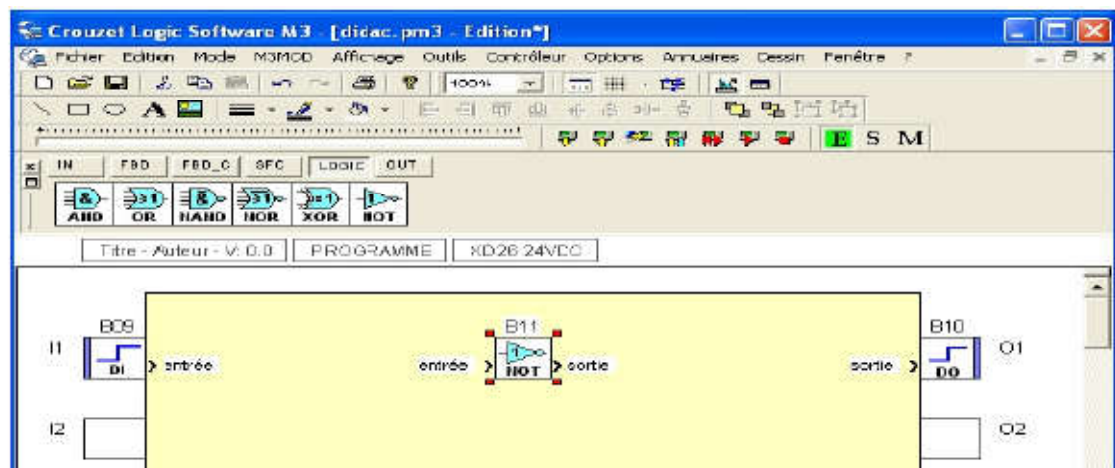
- l. La fenêtre montre alors le circuit dessiné en fonctionnement : une liaison en bleu indique un état logique 0 ; une ligne en rouge indique un état logique 1.



- m. Pour changer l'état d'une entrée, il suffit de cliquer dessus :



- n. 14. On peut aussi utiliser du texte à la place des fils de liaison. Il suffit de cliquer avec le bouton droit sur une liaison et de choisir : "Type de câblage / Texte" et de remplacer dans un second temps le texte par défaut par celui de votre choix (entrée 1, etc.)



EXERCICES D'APPLICATION 1 : CIRCUIT LOGIQUE

On vous demande de réaliser le schéma ci-dessous et de remplir la table de vérité ci-après à partir des essais effectués en mode Simulation.

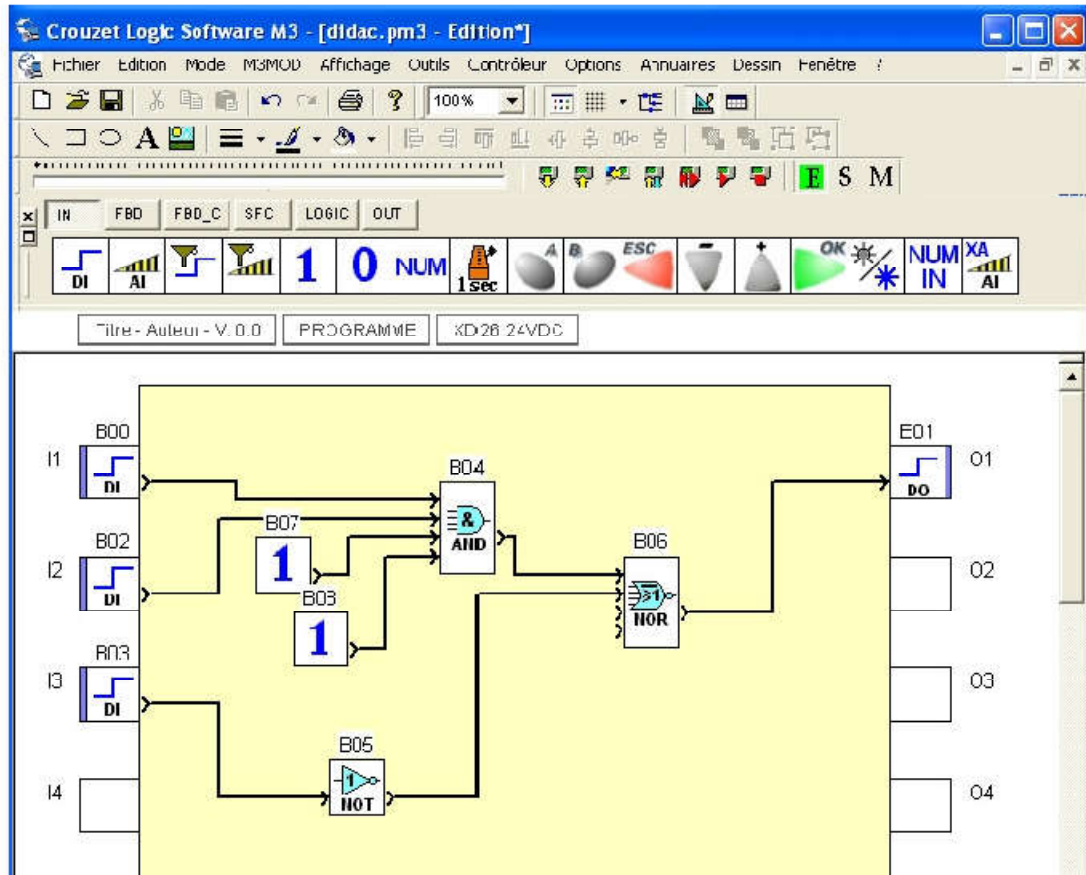


Table de vérité :

I1	I2	I3	Q1
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

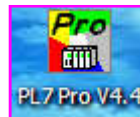
INITIATION AU LOGICIEL PL7-PRO

I.1 Introduction

L'objectif de cette partie est de connaître le minimum requis pour l'utilisation du logiciel PL7 vis à vis d'un TSX 37/57. Les langages PL7 sont des langages graphiques (Langage à Relais, Grafcet) et textuels (Structured Text, Instruction List) destinés à programmer les automates TSX. Ils offrent un environnement de programmation complet permettant à l'utilisateur de développer son application dans un langage graphique simple (éditeur), de la traduire automatiquement en langage automate (compilateur) et enfin de la charger dans la mémoire centrale de l'A.P.I. (téléchargement).

Suivant le TSX dont vous disposez (Micro ou Premium), une certaine quantité de mots et de bits sont disponibles (pour le TSX3722 Micro, on a 128 bits d'étapes %Xi, 256 bits internes %Mi, et un nombre de mots limité par la taille mémoire de l'automate.).

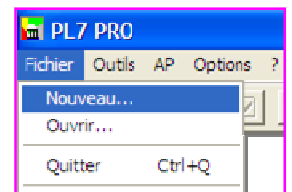
I.2 Ouverture d'une session



Cliquer sur l'icône dans le bureau ou utiliser le menu Démarrer-->Programmes-->Modicon Télémécanique-->PL7 PRO V4.4.

I.2.1 Créer un nouveau projet

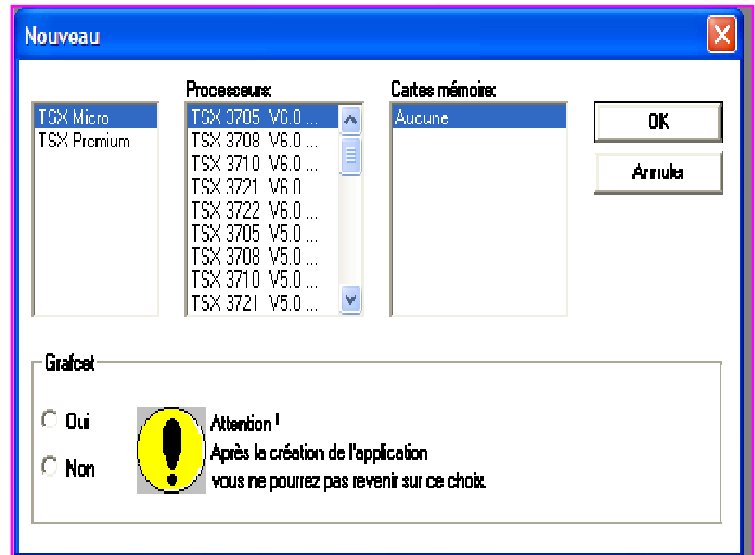
Créer une nouvelle application dans le menu "Fichier", sous menu "Nouveau...".



I.2.2 Sélectionner la version d'automate que vous utilisez

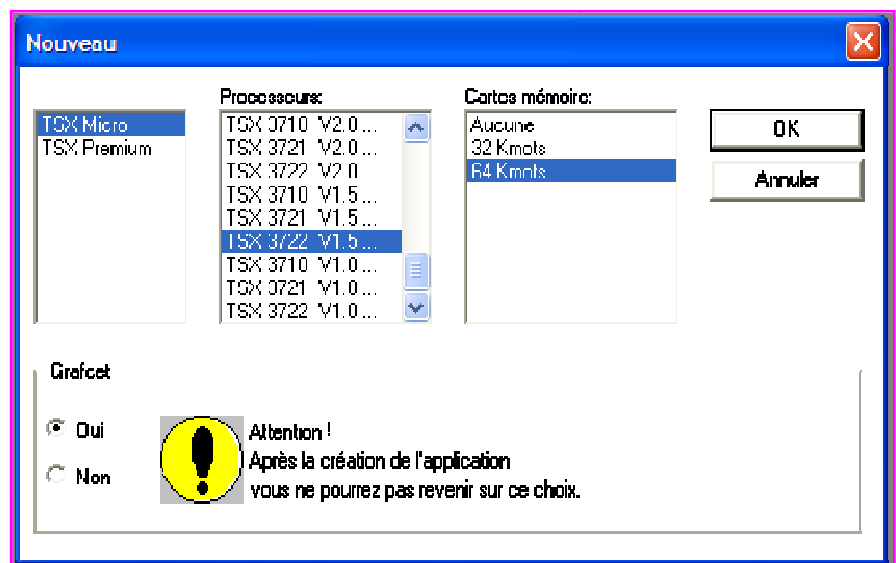
Après création du nouveau projet , on obtient cette fenêtre.

Où on doit choisir en premier le nom de l'automate utilisé soit TSX Pemium (pas Langage Grafcet) ou TSX Micro (avec Langage Grafcet) , puis en second on choisit le type de processeur et enfin la carte mémoire utilisée (si elle existe) choisir la programmation Grafcet oui ou non selon la solution à proposée



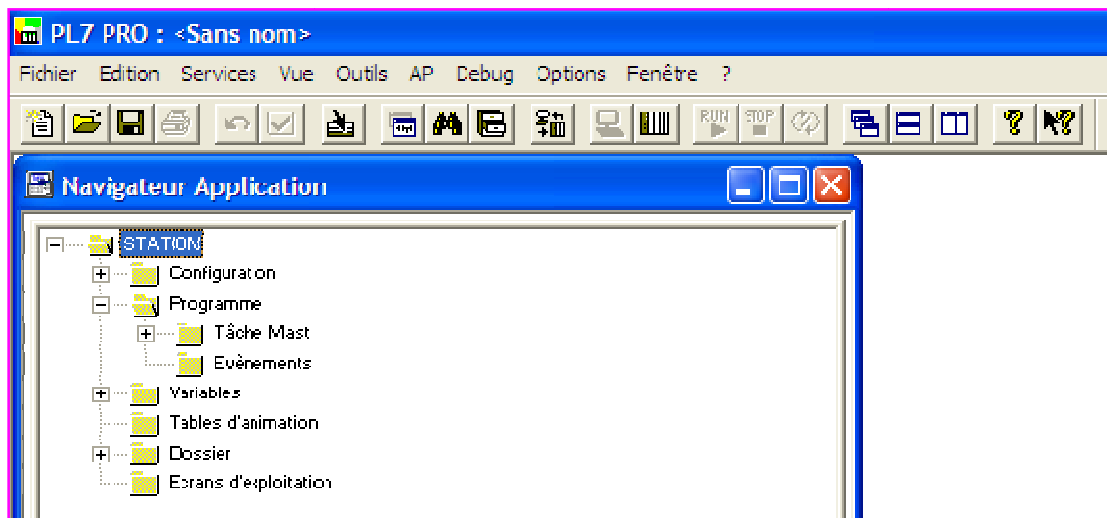
Choisir l'automate utilisé par exemple TSX3722 V3.0 , cartes mémoires « 64 Kmots » et « oui » pour la programmation Grafcet.

Valider votre choix par l'appui sur le bouton "OK".



I.2.3 Ouverture « Navigateur Application »

L'application est créée et la fenêtre "Navigateur Application" s'ouvre alors.



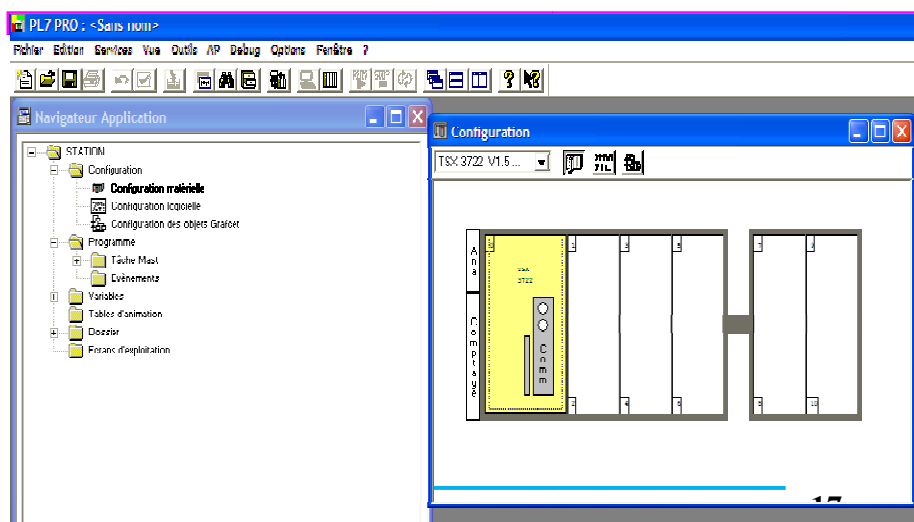
Le Navigateur "Application" présent sur la gauche de l'écran permet d'accéder aux différentes fonctionnalités de l'outil de développement : configuration, programmation, accès aux données, transfert et mise en application du programme, dossier de l'application en cours.

I.2.3.a. Configuration matérielle de l'Automate

1) Configuration (automatique) du processeur

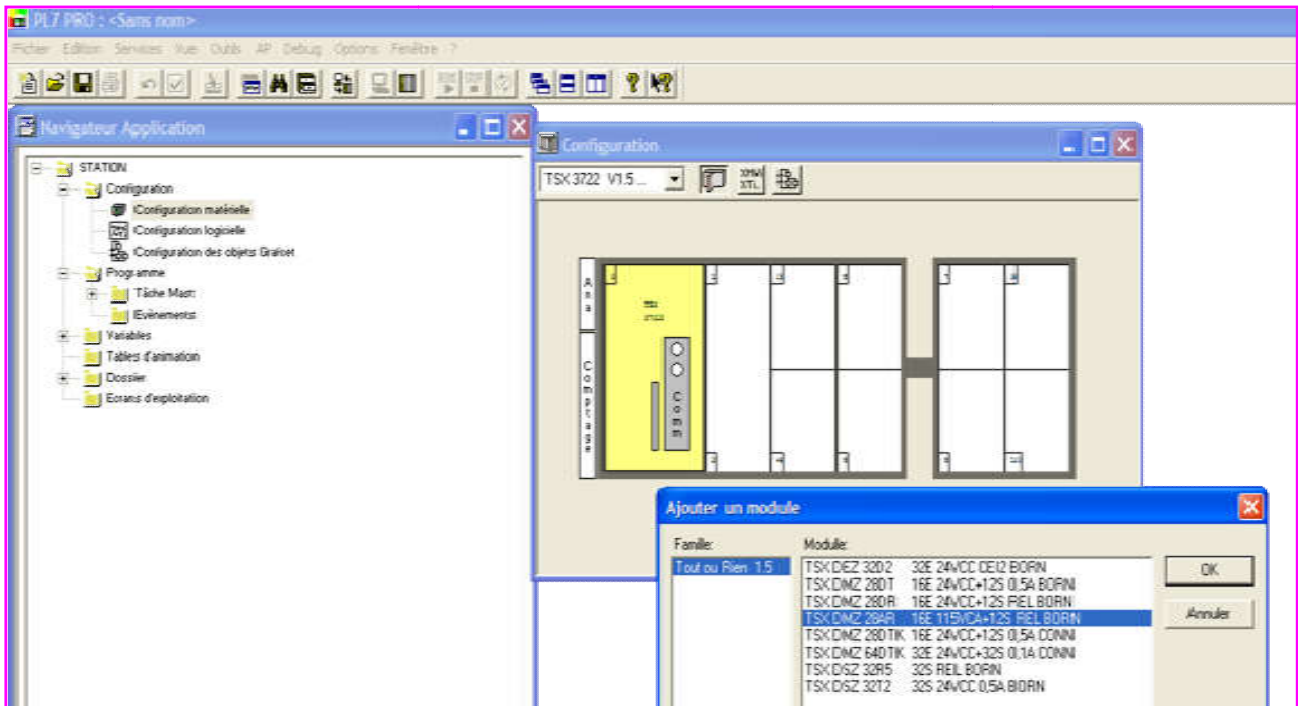
La première étape est de définir sur l'outil de développement la configuration matérielle de l'automate sur lequel le programme sera transféré puis exécuté. Sélectionner dans le dossier "Configuration"--> "Configuration matérielle".

On obtient une image du TSX choisi avec par défaut l'ensemble de ses emplacements vides. Le module de communication "Com" ou module « 0 » est par défaut correctement configuré il correspond au processeur choisit.



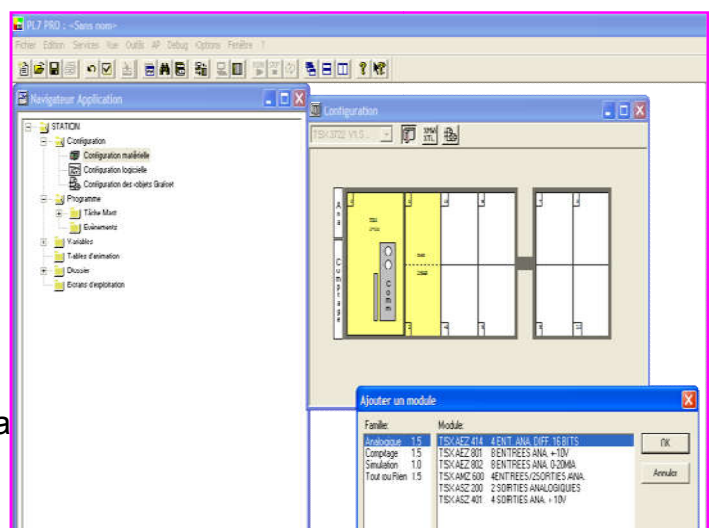
2) Configuration module entrée / sortie

Pour déclarer les entrées sorties du TSX, cliquer sur l'emplacement 2 et ajouter le module TSXDMZ28DR . Ce module comporte 16 entrées TOR sur la position 1 et 12 sorties sur la position 2 du rack. On valide par OK.



On obtient cette fenêtre, si le nombre variable d'entrée (ou le type d'entrée) ou sortie , on peut ajouter d'autre module on double cliquant sur le module 3 et procédé de la même manière précédente

En quittant, valider la "reconfiguration ma

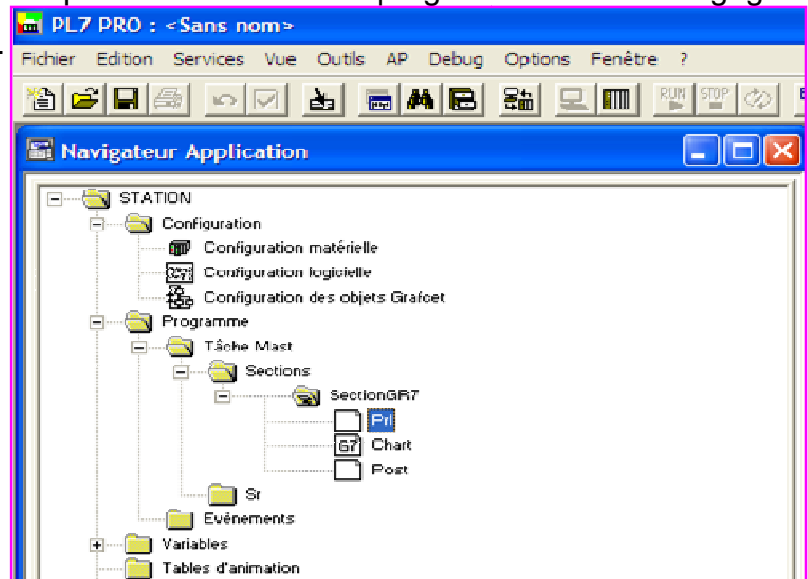


I.2.4 Programmation d'une application

Par défaut, la programmation se fait en mode "déconnecté" ou "local". Vous ne communiquez pas avec le TSX. Le navigateur d'application permet d'accéder à tous les

modules de programme qui constituent l'application complète. Si vous avez choisi le mode "Grafcet" à la création de l'application, le navigateur montre les éléments de programme prédéfinis : le Grafcet en lui-même (Chart) , la partie de programme exécutée avant l'exécution du Grafcet (Pré) et celle exécutée après (Post) . Le langage des sections Pre et Post doit être défini par l'utilisateur lors du premier appel de la section. Vous pouvez rajouter une ou plusieurs sections de programme dans le langage de votre choix à votre application. défaut.

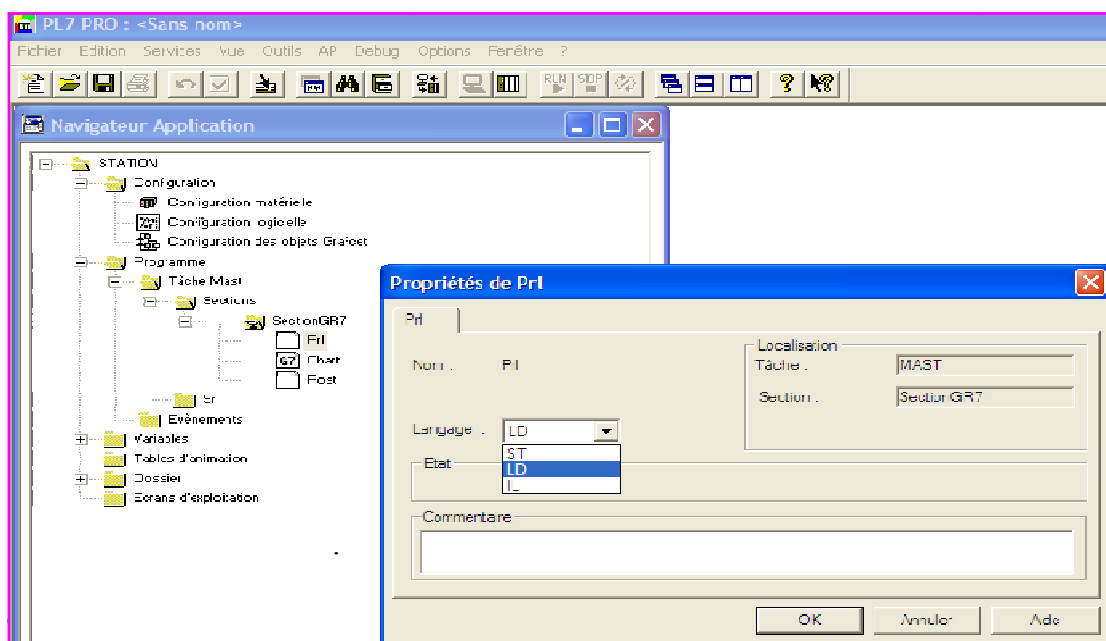
Pour commencer la programmation, on ouvre le dossier Programme, Tâche Maste, Sections, SectionGR7



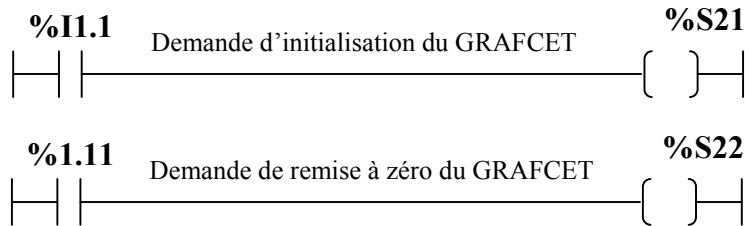
I.2.4.a. Traitement préliminaire

Pour une programmation d'un Grafcet, le traitement préliminaire dépend du cahier de charge.

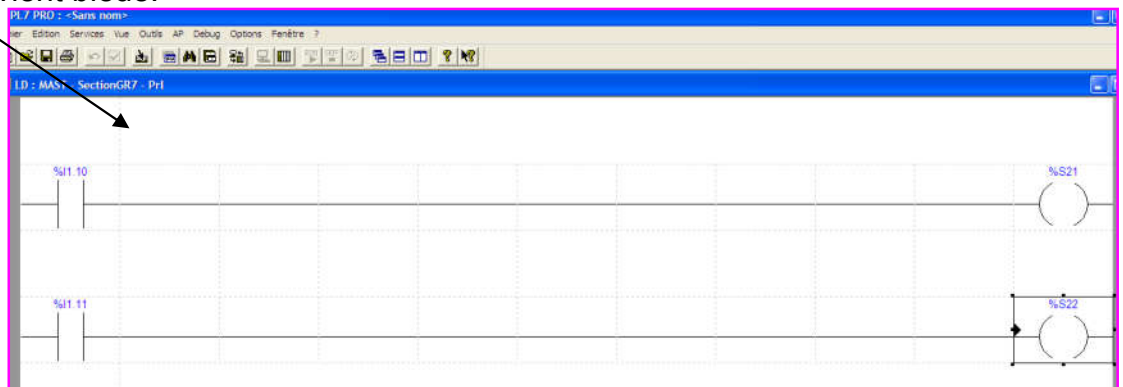
Ouvrir le traitement préliminaire en double-cliquant sur « PRL ». Il peut être programmé en Ladder (LD), structuré (ST), ou instruction list (IL), choisir « LD »



%S21 : initialisation du GRAFCET %S22 : remise à zéro du GRAFCET



Une fois le programme terminé, il faut valider, la couleur des traits et des adresses change et devient bleue.

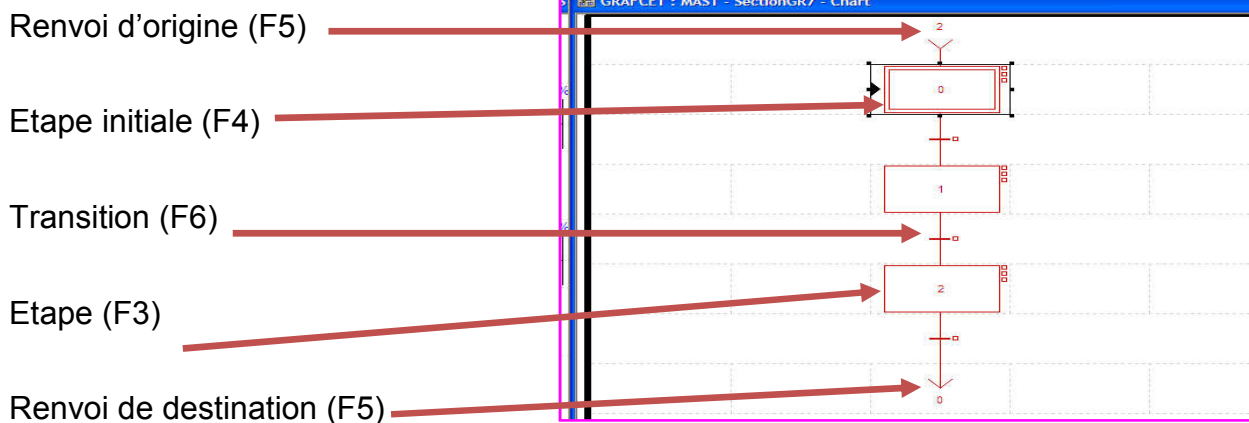


1.2.4.b. Traitement séquentiel

La programmation le traitement séquentiel se fait en deux étapes, la construction de l'architecture du Grafcet et la programmation des transitions.

1) Construction de l'architecture du graphe.

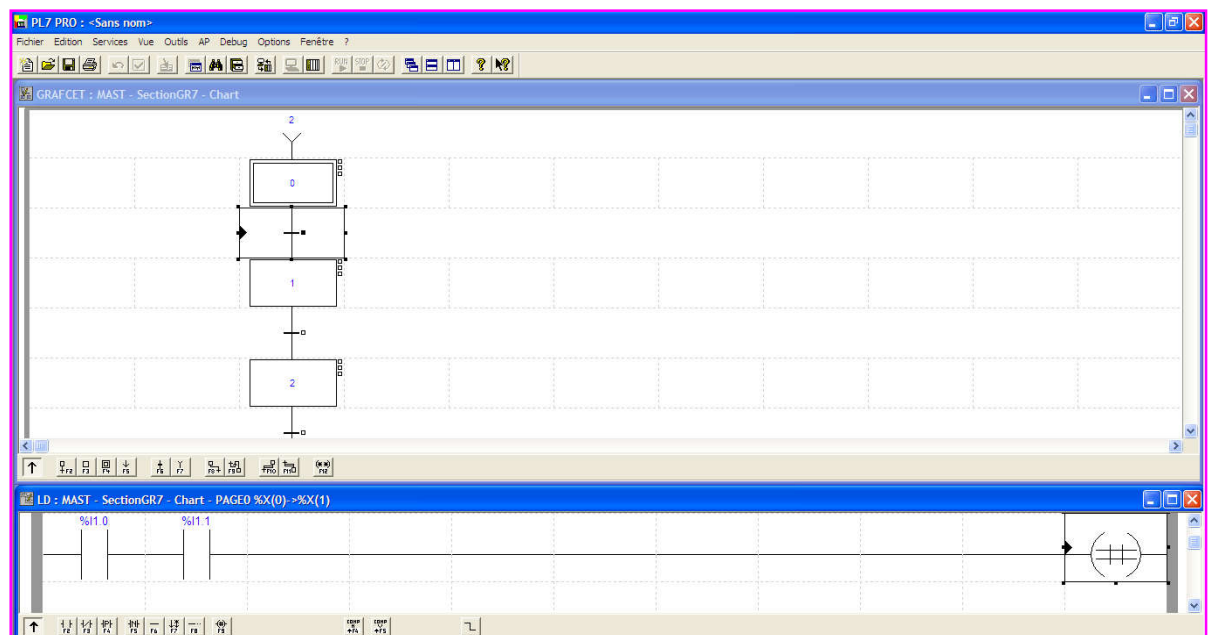
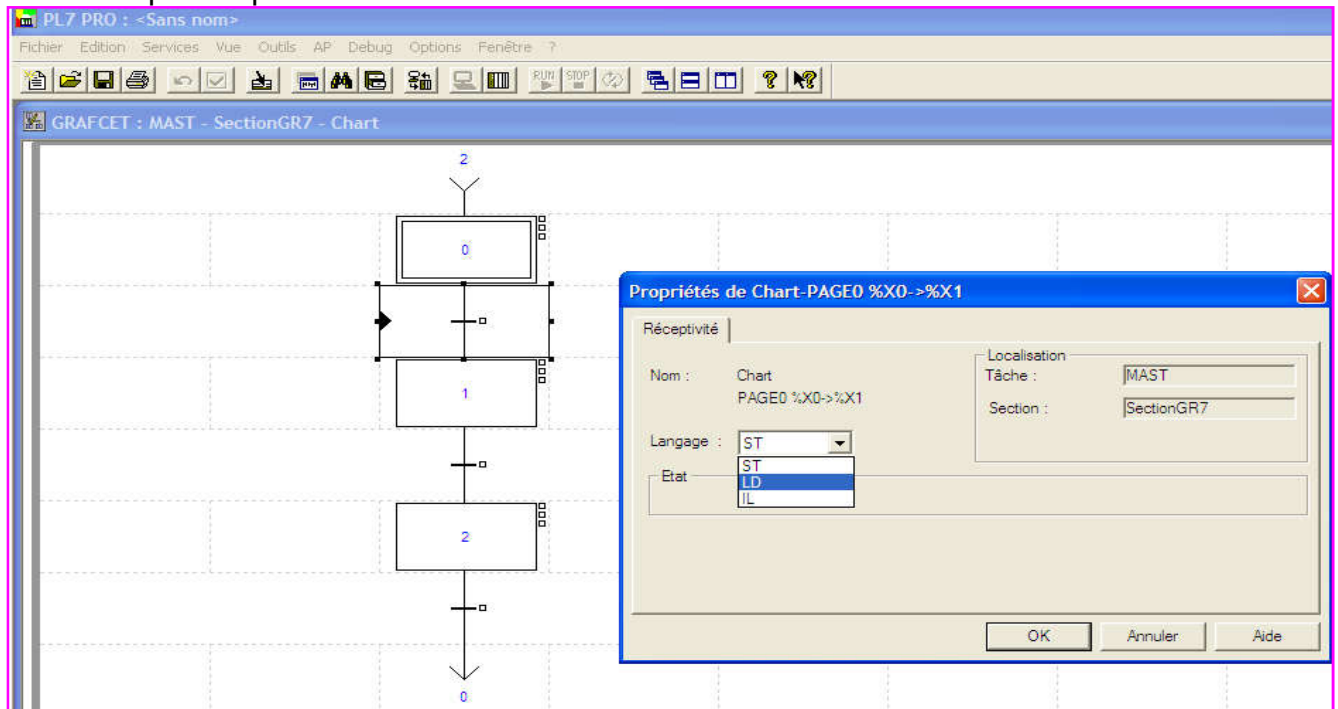
Il faut Cliquer sur **chart** (Le navigateur d'application, Programme, Tâche Maste, Sections, SectionGR7, **chart**), la page suivante s'ouvre. Compléter cette dernière en fonction du grafcet.



Une fois terminé, il faut valider, la couleur du Grafcet devient noire.

2) Programmation des transitions

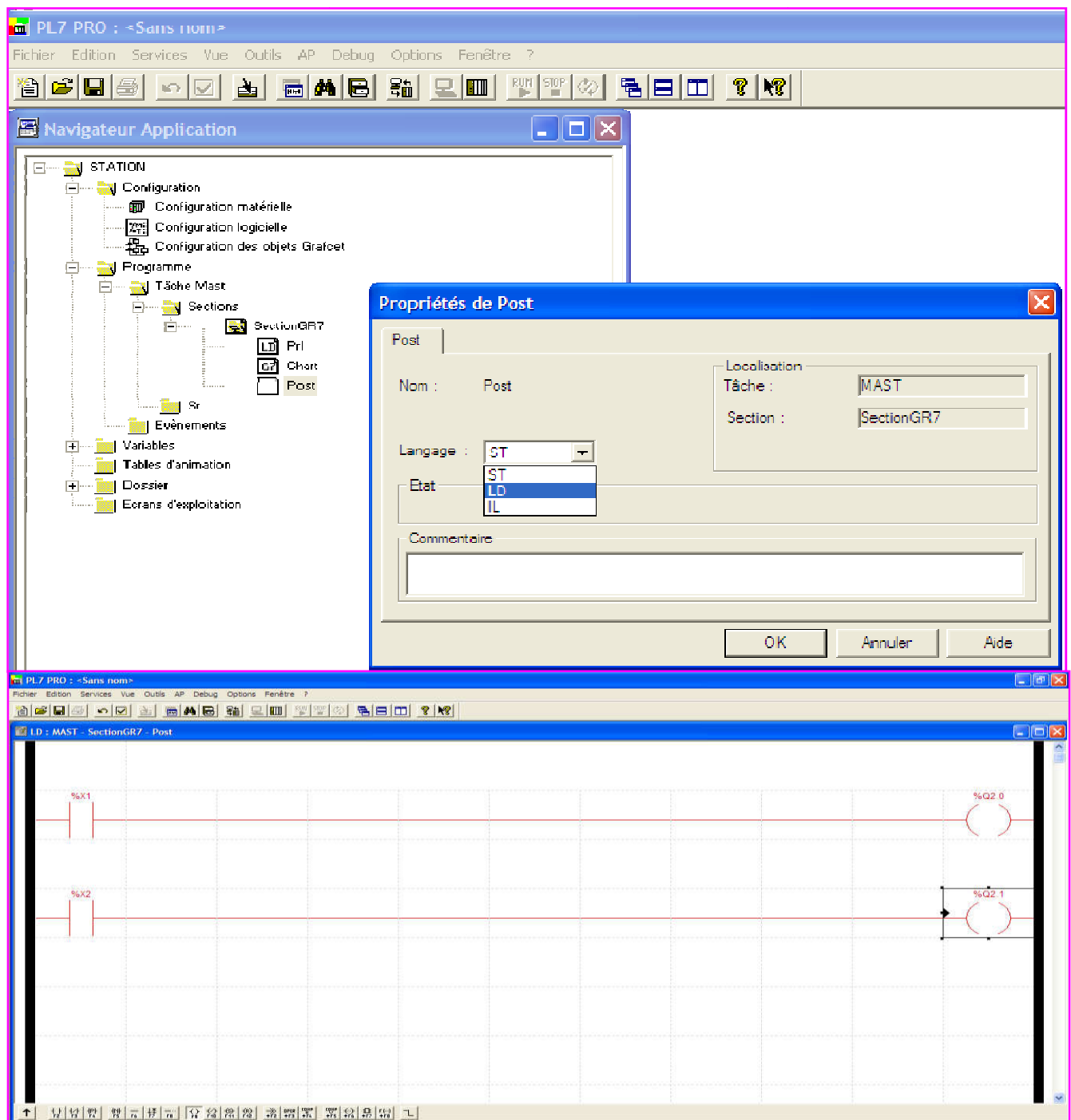
Pour la programmation des transitions, double-cliquez sur celle qui vous intéresse et sélectionnez le langage de spécification formelle de la réceptivité associée (LD pour LADDER) puis éditer la fonction combinatoire de transition (la transition agit sur la sortie implicite "#" qui est la réceptivité de l'étape suivante). Vous validez et fermez la fenêtre de la réceptivité que vous venez de saisir afin d'en saisir une autre de la même manière.



Après validation l'écriture de l'adresse devient bleue et le carré de la transition devient noir complètement.

1.2.4.c. Traitement postérieur

C'est la programmation **actions associées aux étapes** . Pour traiter les actions Grafcet, ouvrir le traitement postérieur Post (Le navigateur d'application, Programme, Tâche Maste, Sections, SectionGR7, **Post**), . Sélectionner le langage de programmation (LD pour LADDER) et écrire les fonctions de sortie en utilisant les activités d'étapes %Xi



Après la programmation postérieure, il faut valider , le réseau devient noir , l'écriture des adresses bleue.

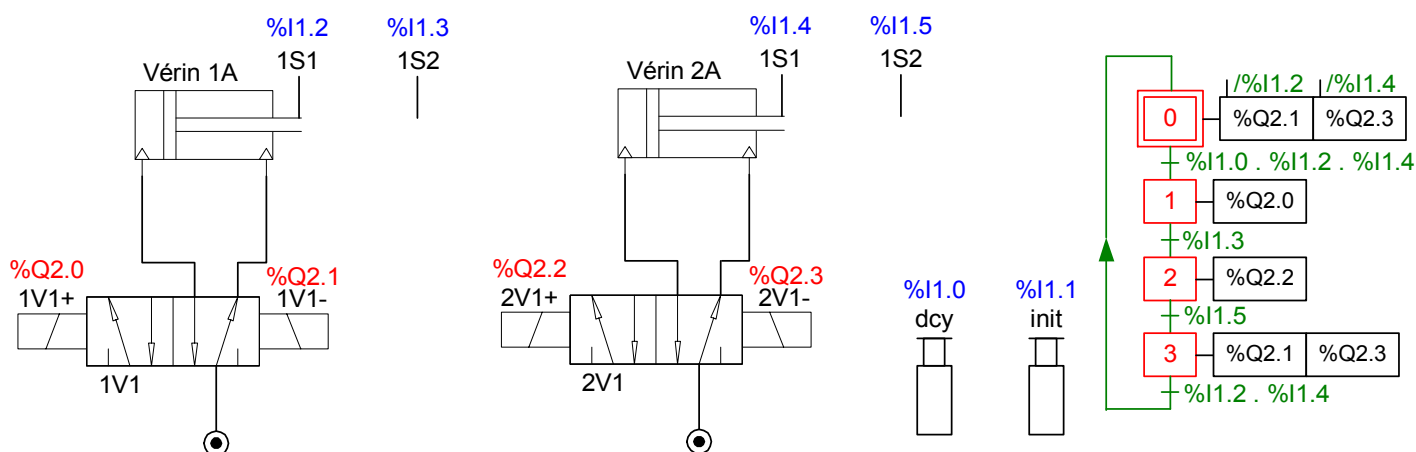
Pour terminer il faut enregistrer le programme.

Exemple 1 : programmation de grafcet

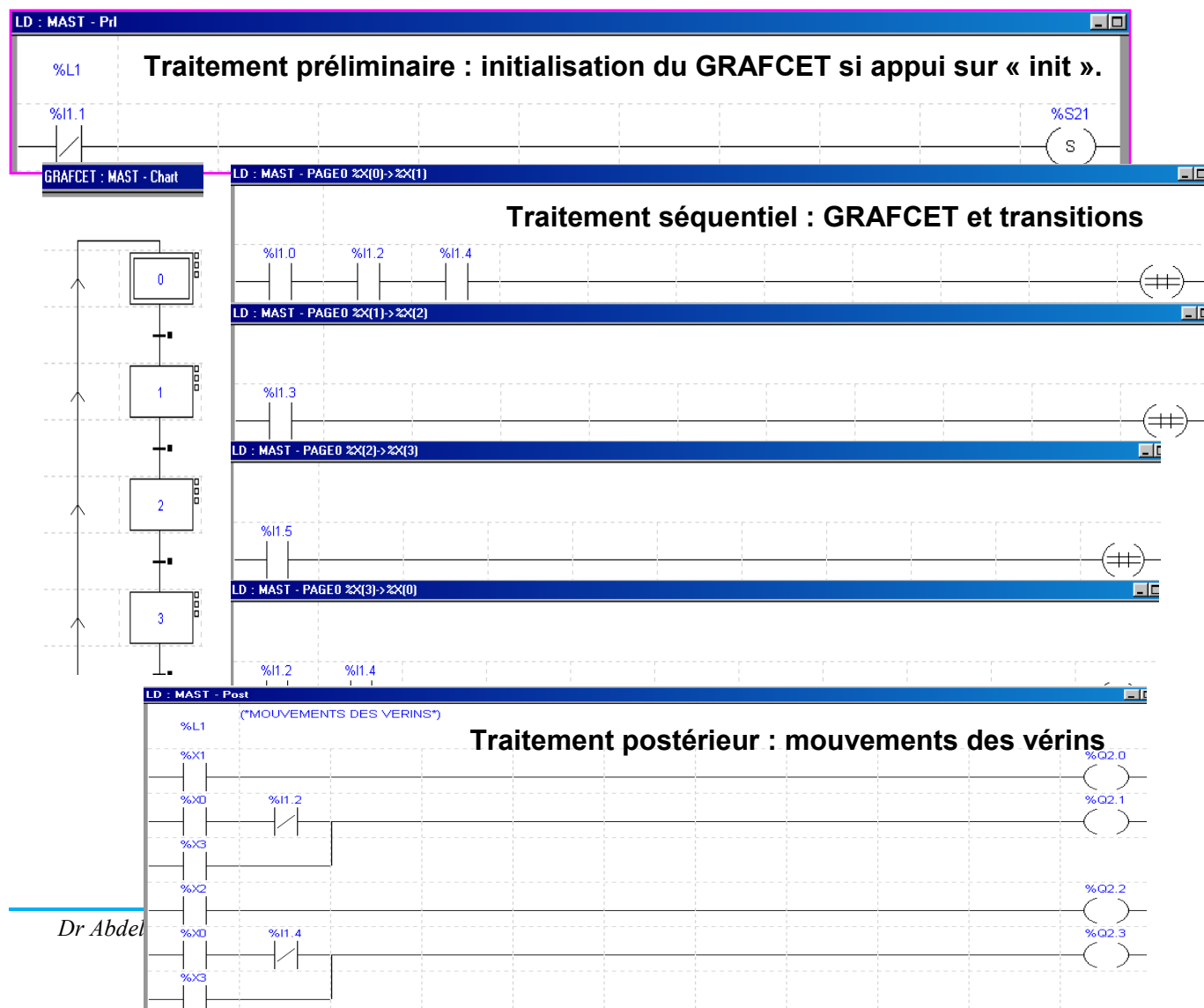
SCHEMA PNEUMATIQUE

GRAFCET

AUTOMATE

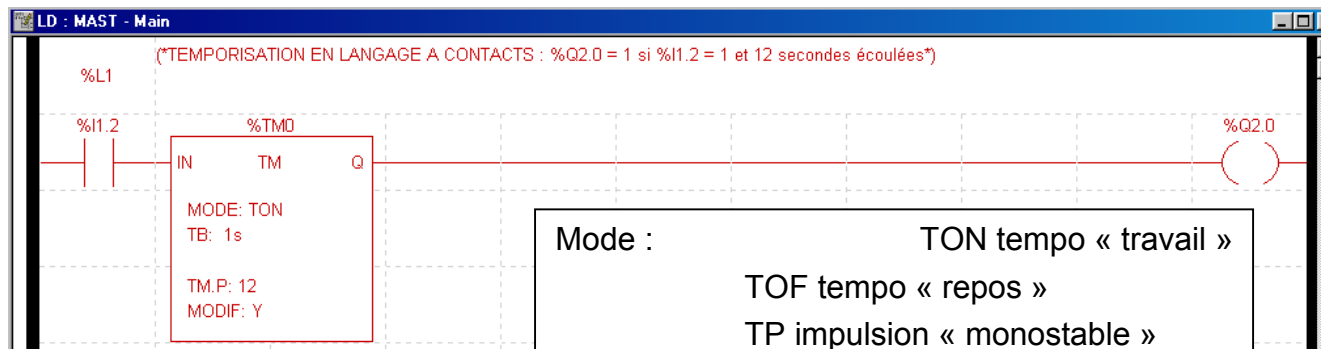


PROGRAMME CORRESPONDANT



Exemple 2 : programmation de temporisation

1/ PROGRAMMATION EN SCHEMA CONTACT



Mode : TON tempo « travail »

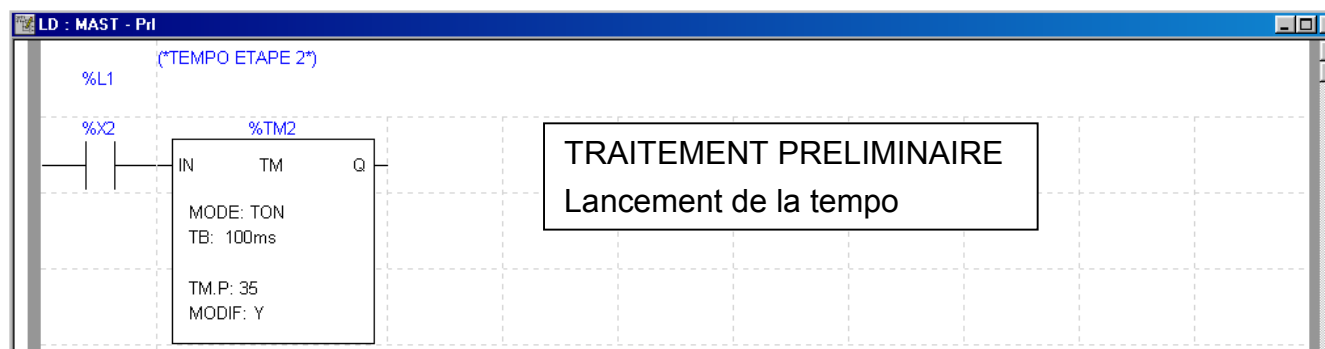
TOF tempo « repos »

TP impulsion « monostable »

TM.P : présélection

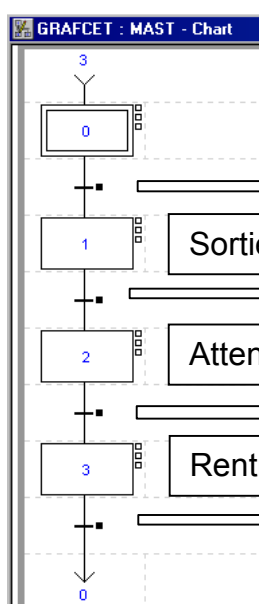
TB : base de temps

2/ PROGRAMMATION EN GRAFCE



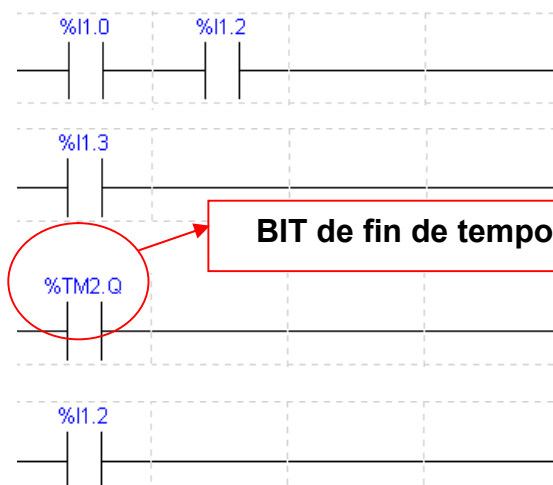
TRAITEMENT PRELIMINAIRE

Lancement de la tempo



TRAITEMENT SEQUENTIEL

$X_2 \rightarrow X_3$ si fin de la tempo TM_2



BIT de fin de tempo = 1 si $TM2.V = 0$

TRAITEMENT POSTERIEUR

Pilotage des mouvements

Fig I. 1 : Raccourci « Automation studio ».

TP N° 1 : **PROGRAMMATION EN LANGAGE LD, IL et ST**

1. Objectifs :

- Se familiariser avec l'automate programmable.,
- Métriser les langages de programmation des automates programmables LD, IL et ST,
- Comprendre l'architecture matérielle des automates programmables industriels
- ,Analyser la structure d'un schéma électrique ,
- Programmer et simuler des circuits électrique sous P17 ou Millénium3.

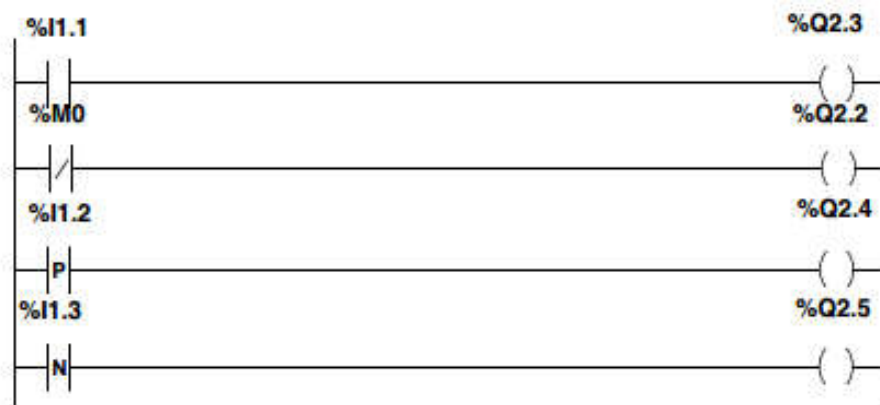
2. Exemples et problèmes de programmations

Exemples

Exemple 1 : Programmation en LD, IL et ST

➤ Exemple en langage Ladder LD :

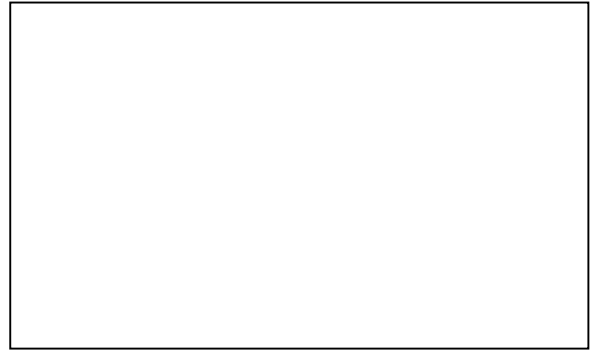
Traduisez le schéma ci-dessous en langage liste d'instruction et texte structuré :



➤ Exemple en langage liste d'instructions IL

N° de ligne	Instruction	Opérande

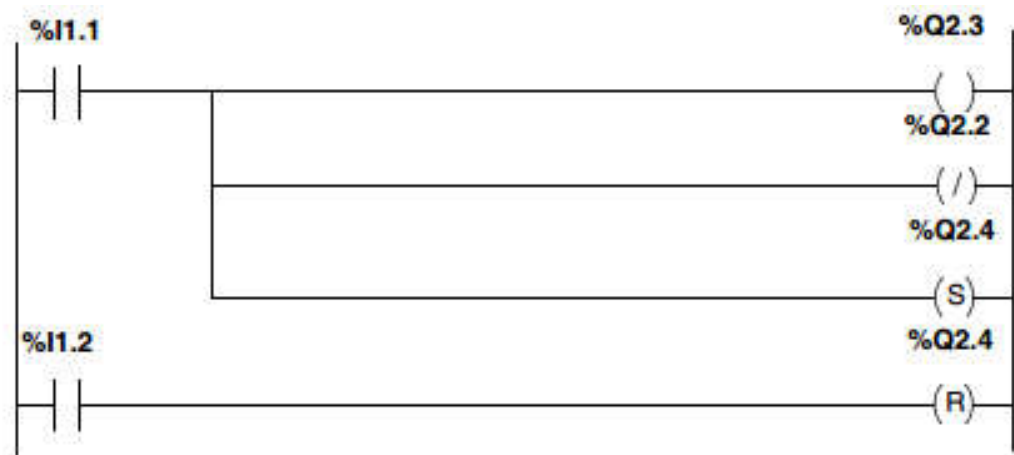
➤ Exemple en langage littéral structuré ST



Exemple 2 : Programmation en LD, IL et ST

➤ Exemple en langage Ladder LD

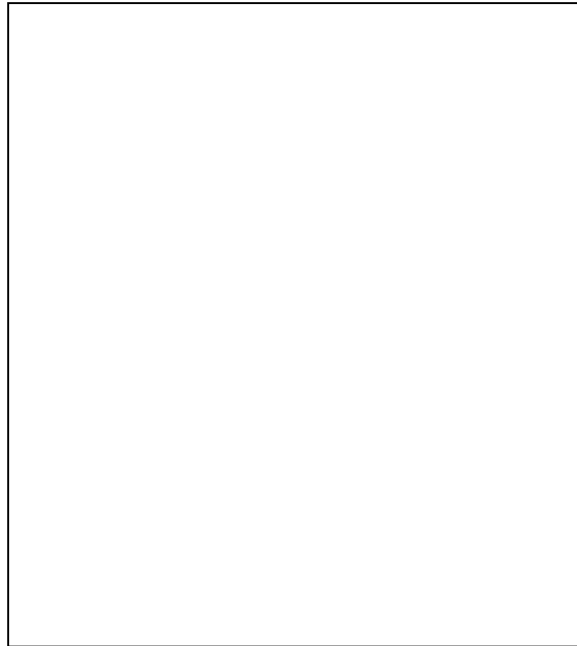
Traduisez le schéma ci-dessous en langage liste d'instruction et texte structuré :



➤ Exemple en langage liste d'instructions IL

N° de ligne	Instruction	Opérande

➤ Exemple en langage littéral structuré ST



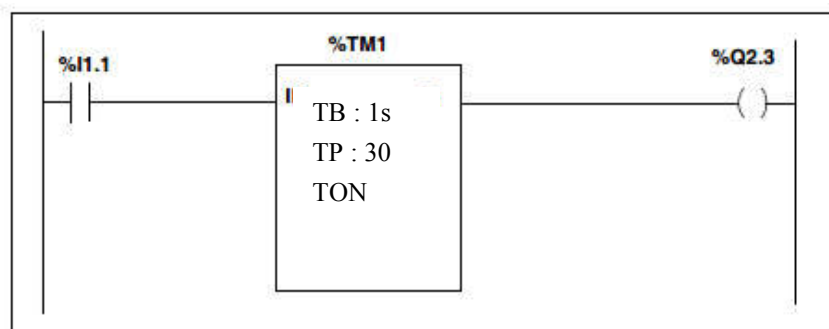
Exemple 2 : Programmation de temporisation

➤ Cahier des charges :

La programmation des blocs fonction temporisateur est identique quel que soit le mode d'utilisation sélectionné. Le choix du fonctionnement T_{Mi} , TON, TB et TP s'effectue dans l'éditeur de variables.

Mode	TON
TB	1s,
TP	30
%T _{Mi}	1

➤ Exemple en langage Ladder LD



➤ Exemple en langage liste d'instructions IL

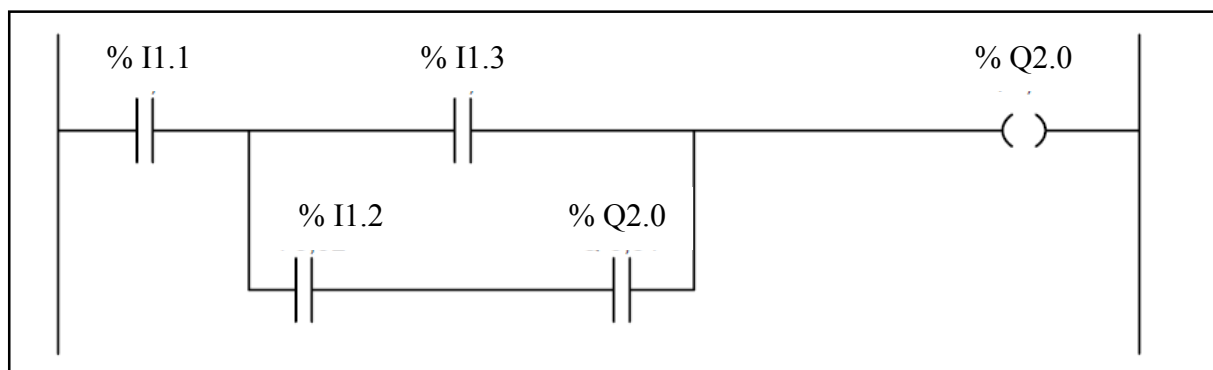
N° de ligne	Instruction	Opérande

➤ Exemple en langage littéral structuré ST

--

- L'instruction START %TMi, génère un front montant sur l'entrée IN du bloc temporisateur.
- L'instruction DOWN %TMi, génère un front descendant sur l'entrée IN du bloc temporisateur.

➤ Schéma en langage LD de la transcription du programme IL.



TP N° 2 :

PROGRAMMATION EN LANGAGE LD, IL ET ST

1. Objectifs :

- Se familiariser avec l'automate programmable.,
- Métriser les langages de programmation des automates programmables LD, IL et ST,
- Comprendre l'architecture matérielle des automates programmables industriels
- Analyser la structure d'un schéma électrique ,
- Programmer et simuler des circuits électrique industriels sous P17 ou Millénium3.

2. Exercices d'applications industrielles

Exercice 1 : Démarrage d'un moteur asynchrone

➤ Cahier des charges :

On se propose de programmer le démarrage d'un moteur asynchrone un seul sens de rotation, dont le circuit de commande est donné par la Figure (II. 2).

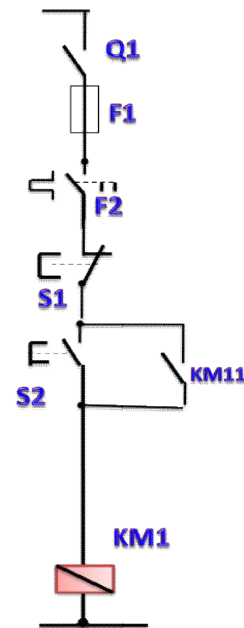


Fig II. 1 : Circuit de commande (logique câblée).

- ✓ Les entrées du système sont : S1 son adresse API est %I1.0
S2 son adresse API est %I1.1
- ✓ La sortie du système est : KM1 son adresse API est %Q2.0 Le contact auxiliaire KM11 d'auto maintien sera pris comme bit mémoire (image) de la sortie KM1, son adresse API est %Q2.0, Le fusible ainsi que le contact Q du sectionneur, le contact du relais

thermique ne seront pas utilisés comme des entrées directes, mais ils seront câblés en série avec le commun de l'alimentation de l'interface de sortie.

Programmation en langage à contact ou Ladder

Le réseau à contact s'inscrit entre deux barres verticales représentant la tension d'alimentation. Exemple de schéma à contact programmable (sous millenium), Figure (II. 3).

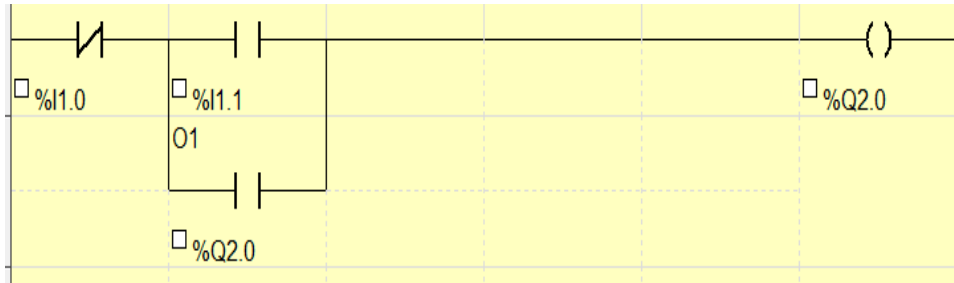


Fig II. 2 : Programme en langage LD du circuit de commande du moteur asynchrone.

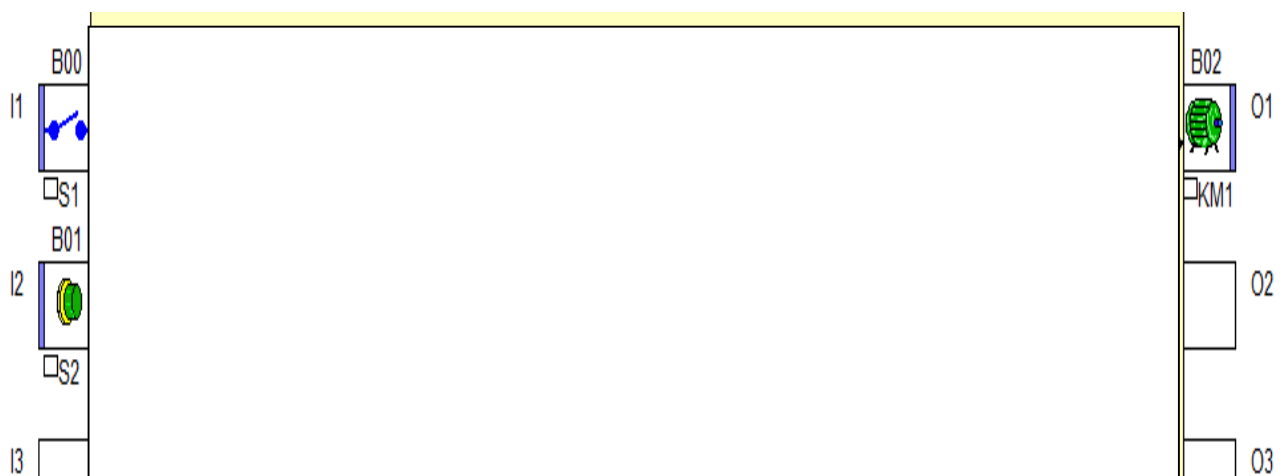
Programmation en langage liste d'instructions

L'adresse ou le code opérande est précédé de %.

<u><i>Exemple d'écriture</i></u> N° de ligne	Instruction	Opérande	Commentaire

Programmation en langage FBD

Il est similaire aux circuits électroniques à base de portes logiques, Figure (II. 4).



TP N° 3 :

PROGRAMMATION EN LANGAGE LD, IL ET ST

1. Objectifs :

- Se familiariser avec l'automate programmable.,
- Métriser les langages de programmation des automates programmables LD et SFC,
- Comprendre l'architecture matérielle des automates programmables industriels,
- Analyser la structure d'un schéma électrique ,
- Programmer et simuler des circuits électrique industriels sous P17 ou Millénium3.

2. Exercice 2 : Etude et commande par API d'un poste de perçage

Cahier des charges

On se propose de programmé, sous langage SFC, la solution GRAFCET d'un système automatisé « Poste de perçage », Figure (II. 5)

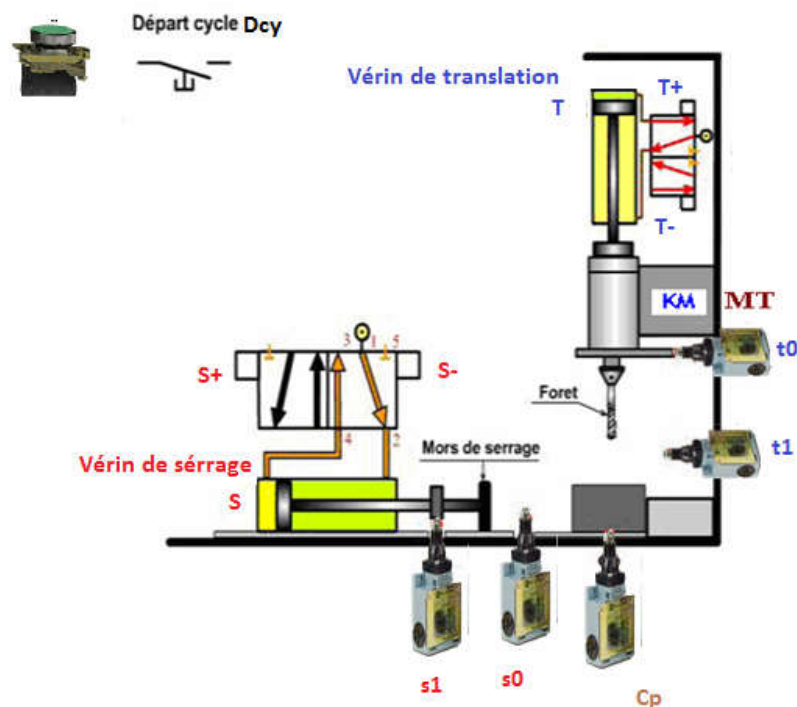


Fig II. 4 : Système de perçage automatique.

➤ Fonctionnement :

L'appui sur le bouton Départ cycle (Dcy) lance le cycle suivant :

- ✓ Serrage de la pièce à percer ;
- ✓ Perçage de la pièce ;
- ✓ Desserrage de la pièce.

▪ Remarque :

Le cycle ne peut être lancé que si la pièce à percer est présente, à l'arrêt les fins de courses **s0** et **t0** sont actionnés.

➤ Fonctionnement en tenant compte des solutions technologiques utilisées :

- Le capteur (**Cp**) indique la présence de la pièce à percer;
- L'appui sur le bouton Départ cycle (**Dcy**) lance le cycle ;
- Le vérin de serrage (**S**) déplace la pièce pour la serrer; le capteur (**s1**) indique que la pièce est serrée ;
- Le moteur supportant le foret (**MT**) commence à tourner et le vérin (**T**) pousse le moteur vers le bas ;
- L'action sur le capteur (**t1**) indique que la pièce est percée, d'où la remonté du vérin (**P**) ;
- Quand le capteur (**t0**) est actionné, le moteur (**MT**) et le vérin (**T**) sont arrêtés , par contre Le vérin (**S**) retourne dans l'autre sens ;
- L'action sur le capteur (**s0**) indique que la pièce est desserrée, et le cycle est terminé (on revient à l'état initial).
- Afin d'améliorer le fonctionnement on doit prévoir le fonctionnement cycle par cycle (Cy/cy)et cycle continu ou automatique (AQ), ainsi qu'un bouton d'initialisation de l'étape initiale.

On demande en premier lieu le GRAFCET du point de commande et par la suite la programmation de la solution trouvée en langage SFC.

Solution graphique par GRAFCET

La solution GRAFCET proposée, Figure ci-dessous :

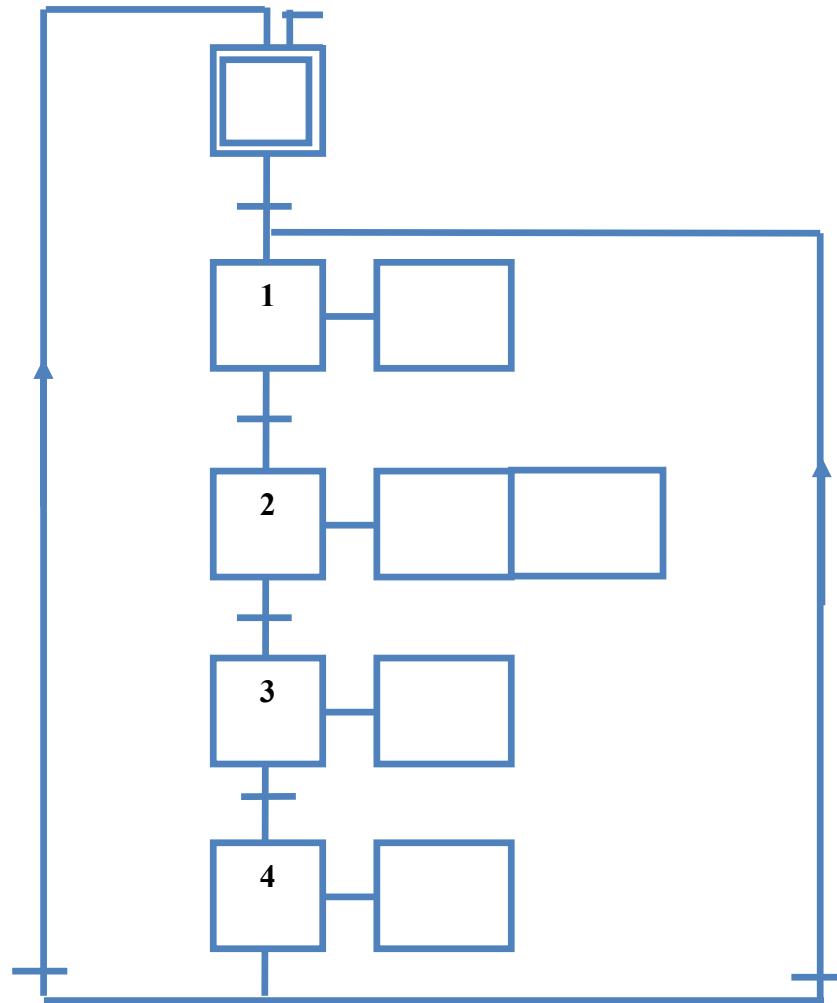


Fig II. 5 : Solution GRAFCET du système de perçage automatique.

Programmation de la solution en langage SFC

La Figure (II.7) illustre la programmation de la solution GRAFCET proposée en langage SFC sous le logiciel Millénium de API Crouzet,

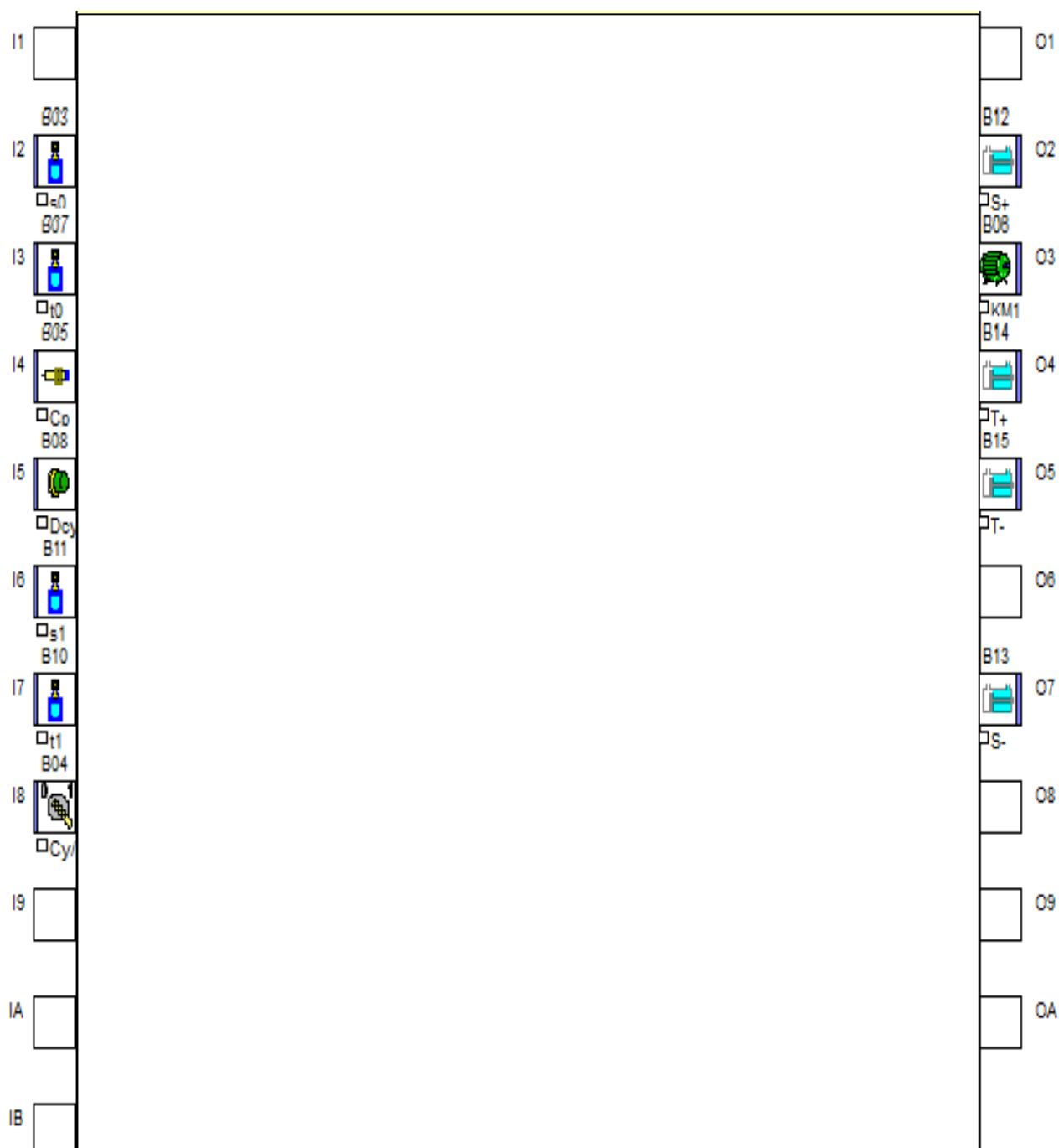


Fig II. 6 : Programmation en SFC de la Solution GRAFCET du système de perçage automatique.

TP N° 4 :

MINI AUTOMATE MILLENIUM 4

1. Objectifs :

- Découvrir et mettre en œuvre le mini automate MILLENIUM et son logiciel de programmation (Crouzet Logic Software).
- Métriser les langages de programmation des automates programmables LD et SFC,
- Comprendre l'architecture matérielle des automates programmables industriels,
- Analyser la structure d'un schéma électrique ,
- Programmer et simuler des circuits électrique industriels sous Millénium3.

2. TRAVAIL DEMANDE :

- 1 Schémas et symboles aux normes
- 2 Ecriture du programme
- 3 Programmation du module
- 4 Schéma de raccordement du module

3. CAHIER DES CHARGES :

Un particulier désire que l'accès à son domicile soit contrôlé par un portail automatisé équipé d'un moteur à double sens de rotation (ouverture ou fermeture).

Ouverture :

Que le portail soit fermé ou qu'il soit en position intermédiaire, le signal de la télécommande provoque l'ouverture complète du portail.

Durant l'ouverture, chaque nouvelle action sur la télécommande stoppe ou relance le moteur.

Dès que le portail est complètement ouvert, une temporisation de 4 secondes retarde sa fermeture.

Fermeture :

Pendant la fermeture, un capteur provoque l'ouverture complète si un passage est détecté ou si la télécommande est actionnée.

Tant que ce détecteur est activé (véhicule arrêté dans le passage par exemple), le portail reste complètement ouvert.

Signalisation :

Une balise clignote durant toute la phase d'ouverture et de fermeture du portail.

Disposition de la partie opérative :

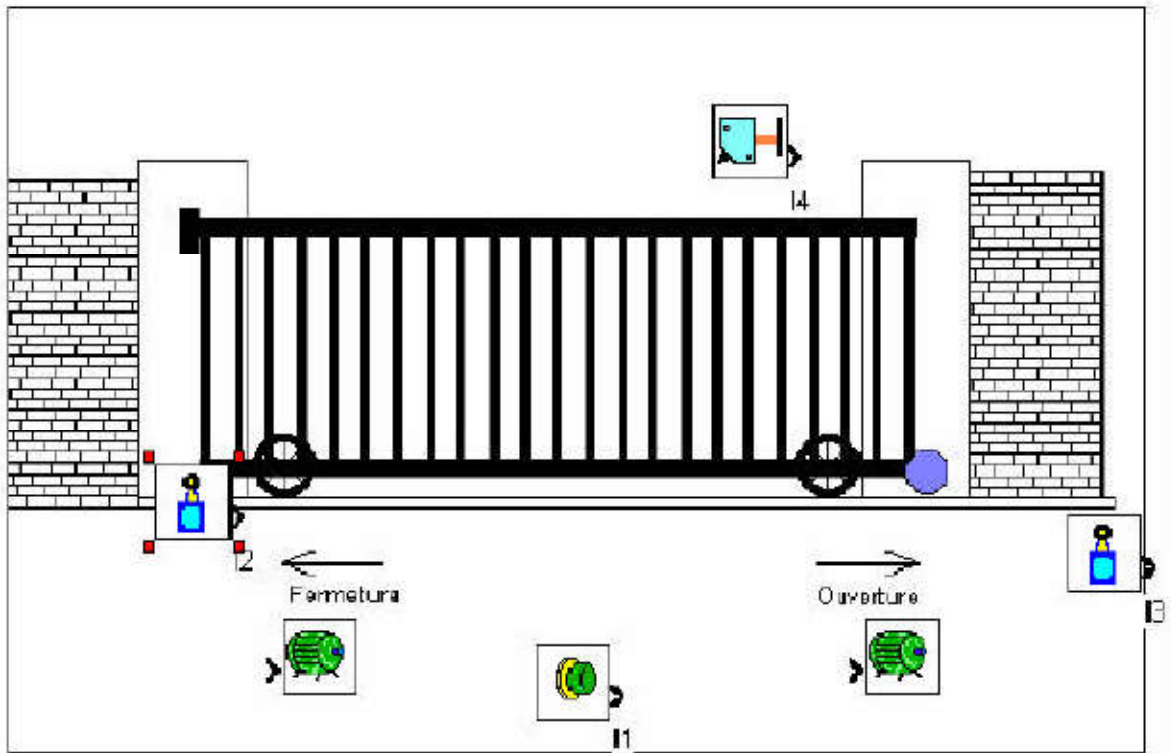


Tableau des entrées. / sorties :

A chaque bouton poussoir il sera affecté une adresse correspondant à une entrée sur le module.

A chaque bobine de contacteur ou voyant il sera affecté une adresse correspondant à une sortie sur le module.

Comme l'indique le tableau ci-contre:

ENTREES		SORTIES	
I1	Télécommande	Q1	Ouverture du portail
I2	Portail position fermé	Q2	Fermeture du portail
I3	Portail position ouvert	Q3	Balise clignotante
I4	Détecteur de passage		

Programmation de l'automate Millénium :

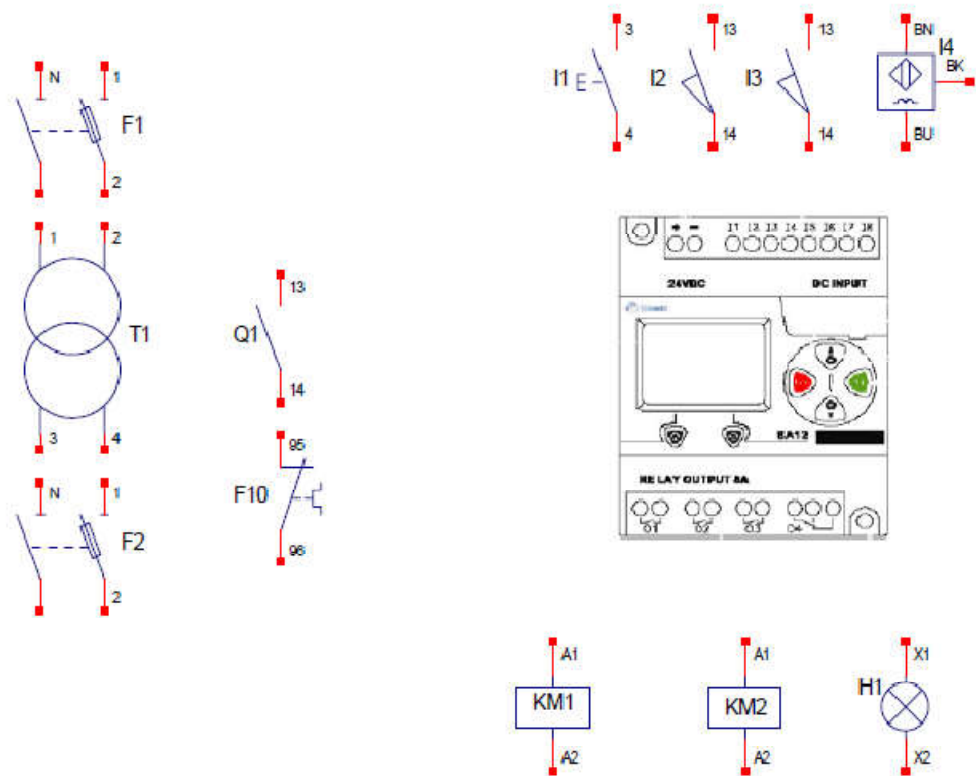
Lancez le logiciel :Crouzet Logic Software, puis vous allez créer un nouveau programme.

Vous devrez ensuite choisir votre automate.

Vous pouvez maintenant programmer. Ce logiciel permet de tester votre programme, il vous faut lancer le mode simulation (S).

SCHÉMA ET CÂBLAGE DU MODULE AVEC SON ENVIRONNEMENT

Complétez les phrases suivantes :



Les entrées et les sorties:

Afin de faciliter la vérification du Câblage, vous devrez utiliser des Cordons bleu pour l'alimentation, **Rouge** pour les entrées et **noir** pour les sorties.

Réalisez le schéma de câblage des entrées sorties sur la feuille ci-dessous en respectant les consignes suivantes :

- Partie alimentation : Les conducteurs d'alimentation seront tracés en bleu.
- Partie entrées : Les entrées seront tracées en rouge.
- Partie sorties: Les sorties seront tracé en noir.

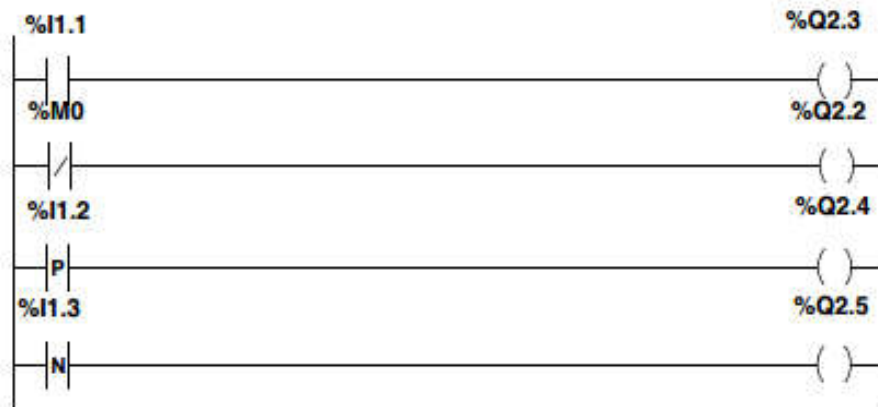
SOLUTIONS DES EXEMPLES ET PROBLÈMES

I.2.1 Exemples

I.2.1.a. Exemple 1 : Programmation en LD, IL et ST

Exemple en langage Ladder LD :

- Traduisez le schéma ci-dessous en langage liste d'instruction et texte structuré :



➤ Exemple en langage liste d'instructions IL

° de ligne	Instruction	Opérande
00	LD	% I1.1
01	ST	% Q2.3
02	LDN	% M0
03	ST	% Q2.2
04	LDR	% I1.2
05	ST	% Q2.4
06	LDF	% I1.3
07	ST	% Q2.5

➤ Exemple en langage littéral structuré ST

% Q2.3 := % I1.1

% Q2.2 := NOT % M0

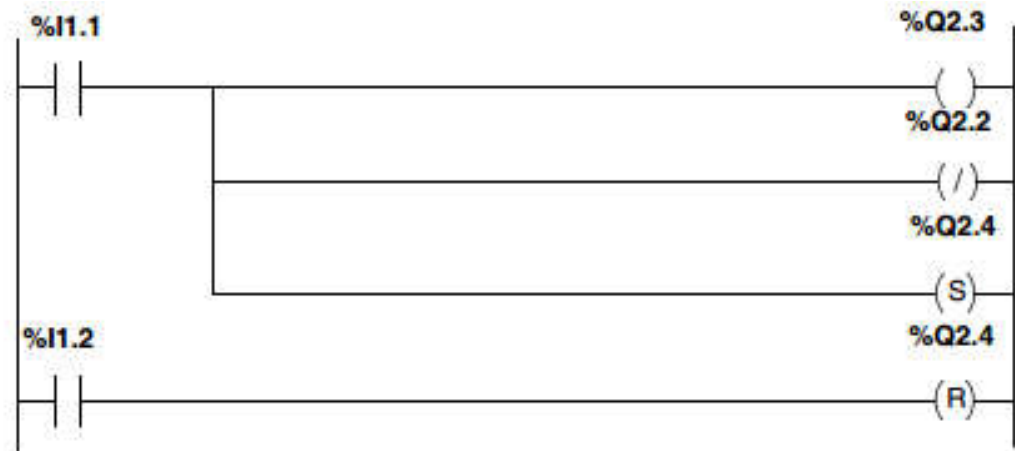
% Q2.4 := RE % I1.2

% Q2.5 := FE % I1.3

1.2.1.b. Exemple 2 : Programmation en LD, IL et ST

➤ Exemple en langage Ladder LD :

- Traduisez le schéma ci-dessous en langage liste d'instruction (PL7) :



➤ Exemple en langage liste d'instructions IL

° de ligne	Instruction	Opérande
00	LD	% I1.1
01	ST	% Q2.3
02	STN	% Q2.2
03	S	% Q2.4
04	LD	% I1.2
05	R	% Q2.4

➤ Exemple en langage littéral structuré ST

```
% Q2.3 := % I1.1;

% Q2.2 :=NOT  % I1.1;

IF % I1.1  THEN

    SET % Q2.4
END_IF;

IF % I1.2  THEN

    RESET % Q2.4;

END_IF;
```

I.2.1.c. Exemple 2 : Programmation de temporisation

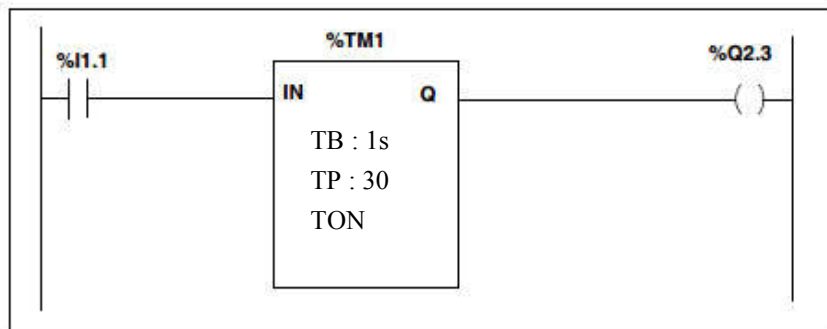
➤ Cahier des charges :

La programmation des blocs fonction temporisateur est identique quel que soit le mode d'utilisation sélectionné.

Le choix du fonctionnement TMi , TON, TB et TP s'effectue dans l'éditeur de variables.

Mode	TON
TB	1s,
TP	30
%TMi	1

➤ Exemple en langage Ladder LD



➤ Exemple en langage liste d'instructions IL

N° de ligne	Instruction	Opérande
00	LD	% I1.1
01	IN	% TM1
02	LD	% TM1.Q
03	ST	% Q2.3

➤ Exemple en langage littéral structuré ST

```

IF RE % I1.1 THEN

    START % TM1;

ELSIF FE % I1.1 THEN

    DOWN % TM1;

    END_IF;

% Q2.3 := % % TM1.Q

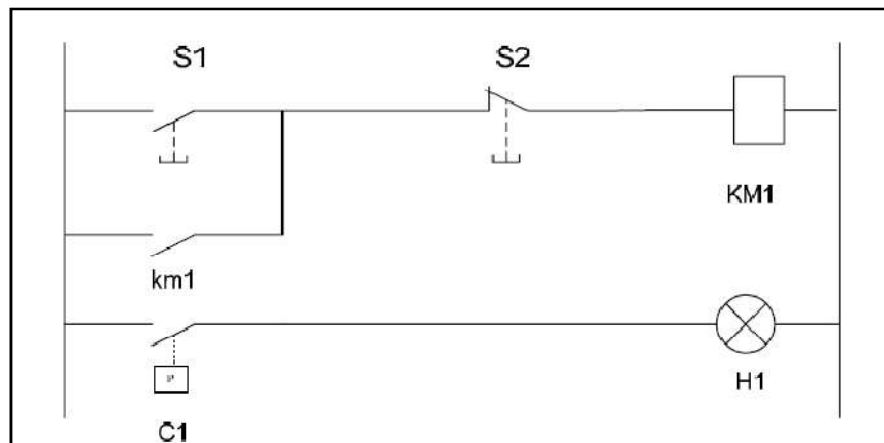
```

- L'instruction START %T_{Mi}, génère un front montant sur l'entrée IN du bloc temporisateur.
- L'instruction DOWN %T_{Mi}, génère un front descendant sur l'entrée IN du bloc temporisateur.

1.2.1.d. Exemple 4 : Commande d'une pompe

➤ Cahier des charges :

- Traduisez le schéma ci-dessous en langage liste d'instruction (PL7) :



➤ Solution :

Soit le tableau des entrées et sortie avec l'adressage API :

Elément d'entrée	désignation	Adressage API	Elément de sortie	désignation	Adressage API
S1	Bouton d'arrêt	% I1.0	KM1	Contacteur de pompe	% Q2.0
S2	Bouton Marche	% I1.1	H1	Voyant de pression	% Q2.1
C1	Capteur de pression	% I1.2			

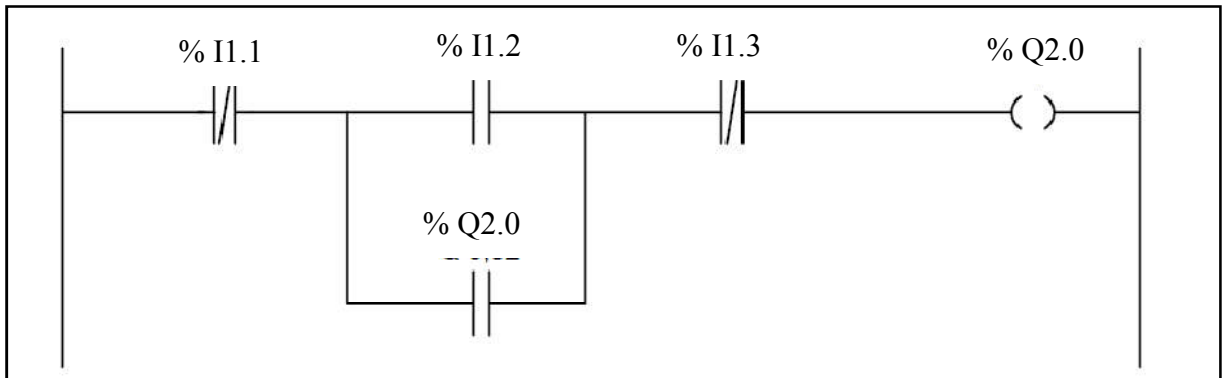
➤ Programme en langage IL du schéma de commande de la pompe

N° de ligne	Instruction	Opérande	Commentaire
00	LD	% I1.0	ter le bouton marche S1
01	OR	% Q2.0	cuter un OU avec KM1
02	AND	% I1.1	cuter un ET avec Le bouton marche S2
03	ST	% Q2.0	iver la sortie du contacteur KM1 (Q2.0)
04	LD	% I1.2	ter la capteur de pression C1
05	ST	% Q2.1	iver la sortie voyant H1

1.2.1.e. Exemple 5 : Transcription du langage Ladder en langage IL

➤ Cahier des charges :

- A partir du programme Ladder ci-dessous déterminer la liste des instructions.



➤ Programme en langage IL

° de ligne	Instruction	Opérande	Commentaire
00	LDN	% I1.1	Lire l'entrée inverse
01	AND(	% I1.2	Exécuter un ET , On imbrique une parenthèse.
02	OR	% Q2.0	Exécuter un OU avec la ligne précédente
03	)		On ferme la parenthèse
04	ANDN	% I1.3	Exécuter un NON ET
05	ST	% Q2.0	Activer la sortie

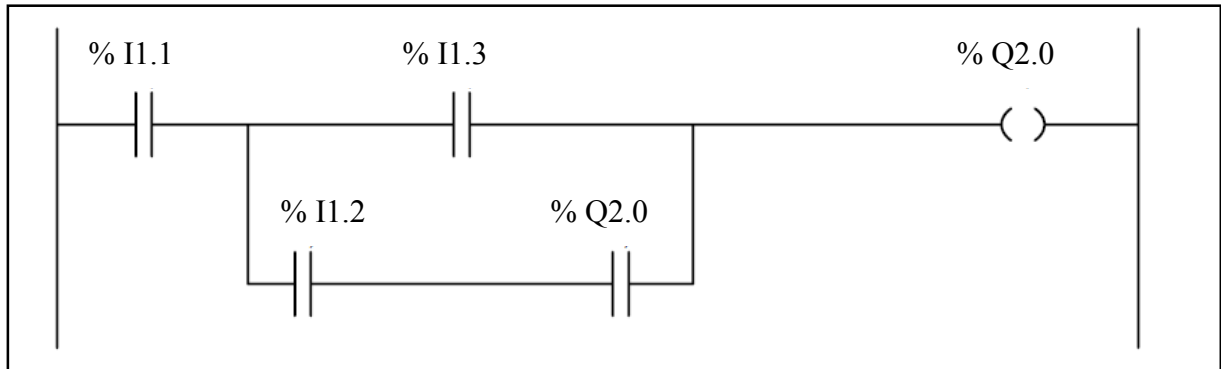
1.2.1.f. Exemple 6 : Transcription du langage IL vers le langage LD

➤ Cahier des charges :

- A partir de la liste des instructions du programme Ladder ci-dessous déterminer.

° de ligne	Instruction	Opérande	Commentaire
00	LD	% I1.1	Lire l'entrée
01	AND(	% I1.3	Exécuter un ET , On imbrique une parenthèse.
02	OR(	% I1.2	Exécuter un OU avec la ligne précédente
03	AND	% Q2.0	Exécuter un NON ET
04	)		On ferme la parenthèse
05	)		On ferme la parenthèse
06	ST	% Q2.0	Activer la sortie

➤ Schéma en langage LD de la transcription du programme IL.



I.2.2 Problèmes

I.2.2.a. Problème 1 : Démarrage d'un moteur asynchrone

- ✓ Les entrées du système sont : S1 son adresse API est %I1.0
S2 son adresse API est %I1.1
- ✓ La sortie du système est : KM1 son adresse API est %Q2.0 Le contact auxiliaire KM1 d'auto maintien sera pris comme bit mémoire (image) de la sortie KM1, son adresse API est %Q2.0, Le fusible ainsi que le contact Q du sectionneur, le contact du relais thermique ne seront pas utilisés comme des entrées directes, mais ils seront câblés en série avec le commun de l'alimentation de l'interface de sortie.

➤ Programmation en langage à contact ou Ladder

Le réseau à contact s'inscrit entre deux barres verticales représentant la tension d'alimentation
Exemple de schéma à contact programmable (sous millenium), Figure (II. 3).

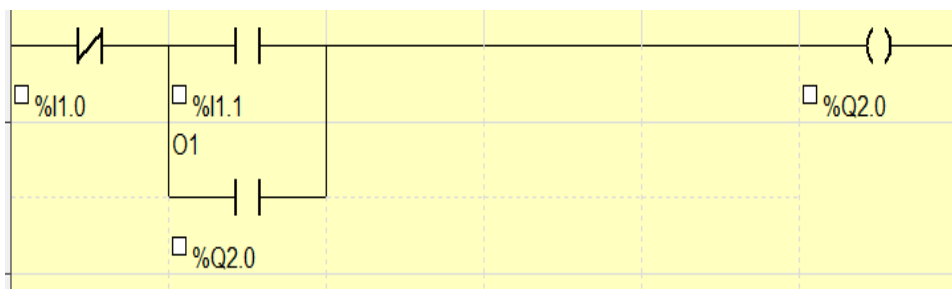


Fig II. 7 : Programme en langage LD du circuit de commande du moteur asynchrone.

➤ Programmation en langage liste d'instructions

L'adresse ou le code opérande est précédé de %.

<u>Exemple d'écriture</u> <u>N° de ligne</u>	<u>Instruction</u>	<u>Opérande</u>	<u>Commentaire</u>
00	LDN	% I1.0	tester le bouton S1

01	AND(	% I1.1	(entrée d'adresse %I1.1) Exécuter un ET avec Le bouton marche S2 (%I1.1) on imbrique une parenthèse .
02		% Q2.0	Exécuter un OU avec KM (mémoire Q2.0)
03			En ferme la parenthèse
04		% O 2.0	Activer la sortie du contacteur KM1 (Q2.0)

➤ Programmation en langage FBD

Il est similaire aux circuits électroniques à base de portes logiques, Figure (II. 4).

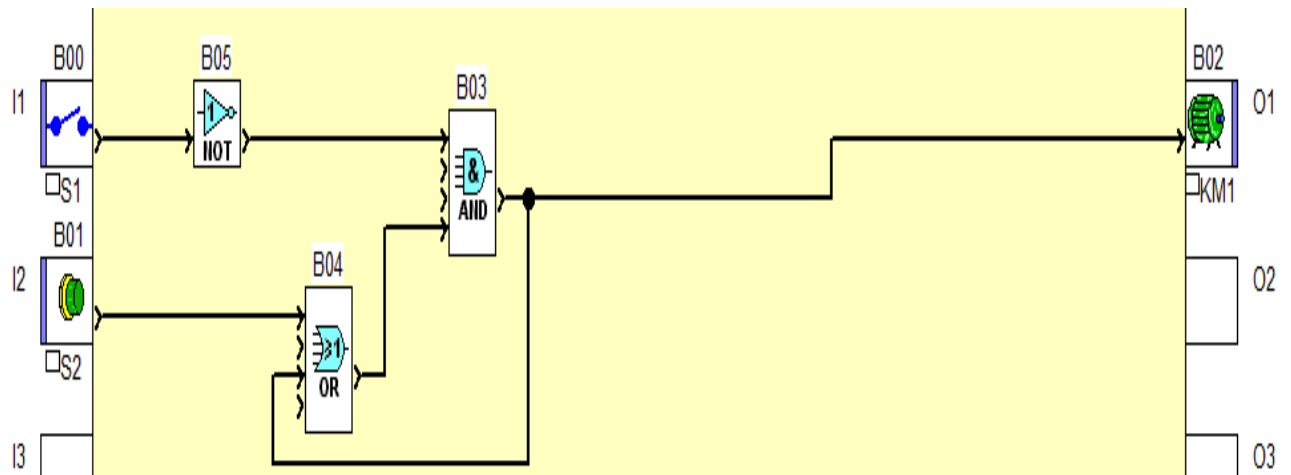


Fig II. 8 : Programme en langage FBD du circuit de commande du moteur asynchrone.

I.2.2.b. Problème 2 : Etude et commande par API d'un poste de perçage

➤ Solution graphique par GRAFCET

La solution GRAFCET proposée, Figure (II.6) :

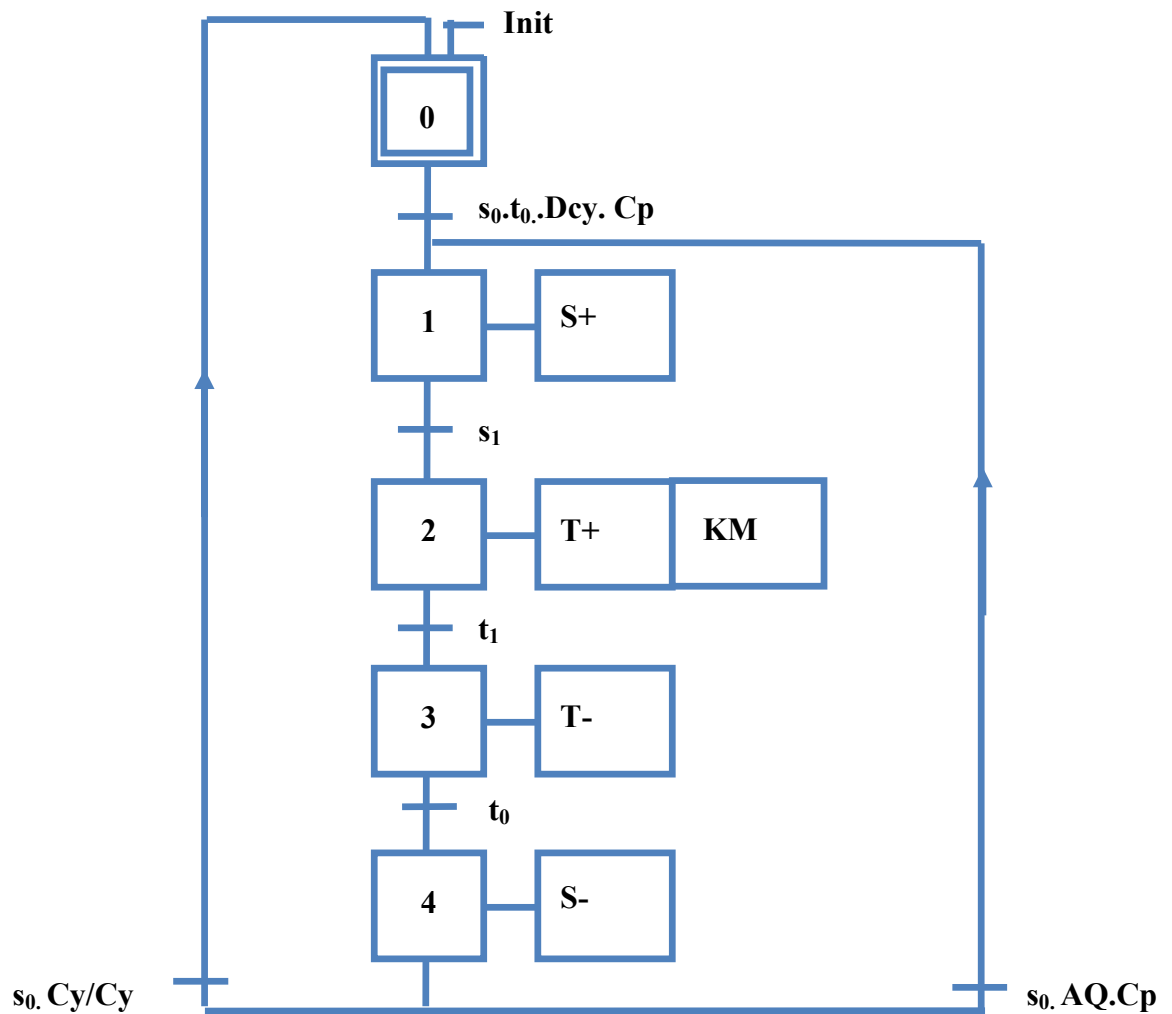


Fig II. 9 : Solution GRAFCET du système de perçage automatique.

➤ Programmation de la solution en langage SFC

La Figure (II.7) illustre la programmation de la solution GRAFCET proposée en langage SFC sous le logiciel Millénium de API Crouzet,

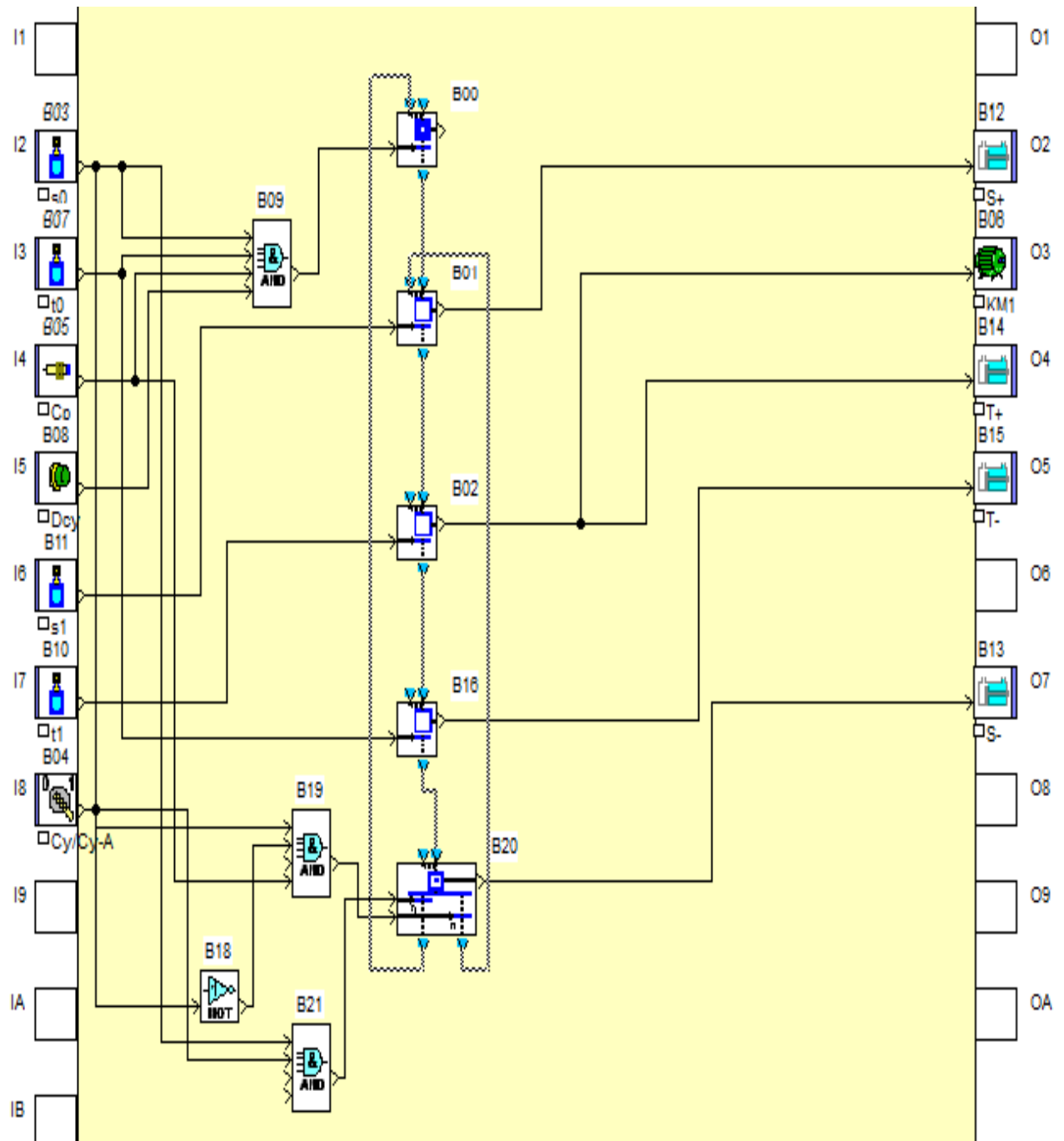


Fig II. 10 : Programmation en SFC de la Solution GRAFCET du système de perçage automatique.

TP N° 5 :**PROGRAMMATION EN LANGAGE LD ET SFC
SOUS LES LOGICIELS MILLINIUM3 ET PL****1. Objectifs :**

- Découvrir et mettre en œuvre le mini automate MILLINIUM et son logiciel de programmation (Crouzet Logic Software).
- Métriser les langages de programmation des automates programmables LD et SFC,
- Comprendre l'architecture matérielle des automates programmables industriels,
- Analyser la structure d'un schéma électrique ,
- Programmer et simuler des circuits électrique industriels sous Millénium3.

2. CAHIER DE CHARGE: COMMANDE D'UN MOTEUR ASYNCHRONE TRIPHASE

On veut programmer le circuit de commande du démarrage d'un moteur asynchrone triphasé, Fig1.

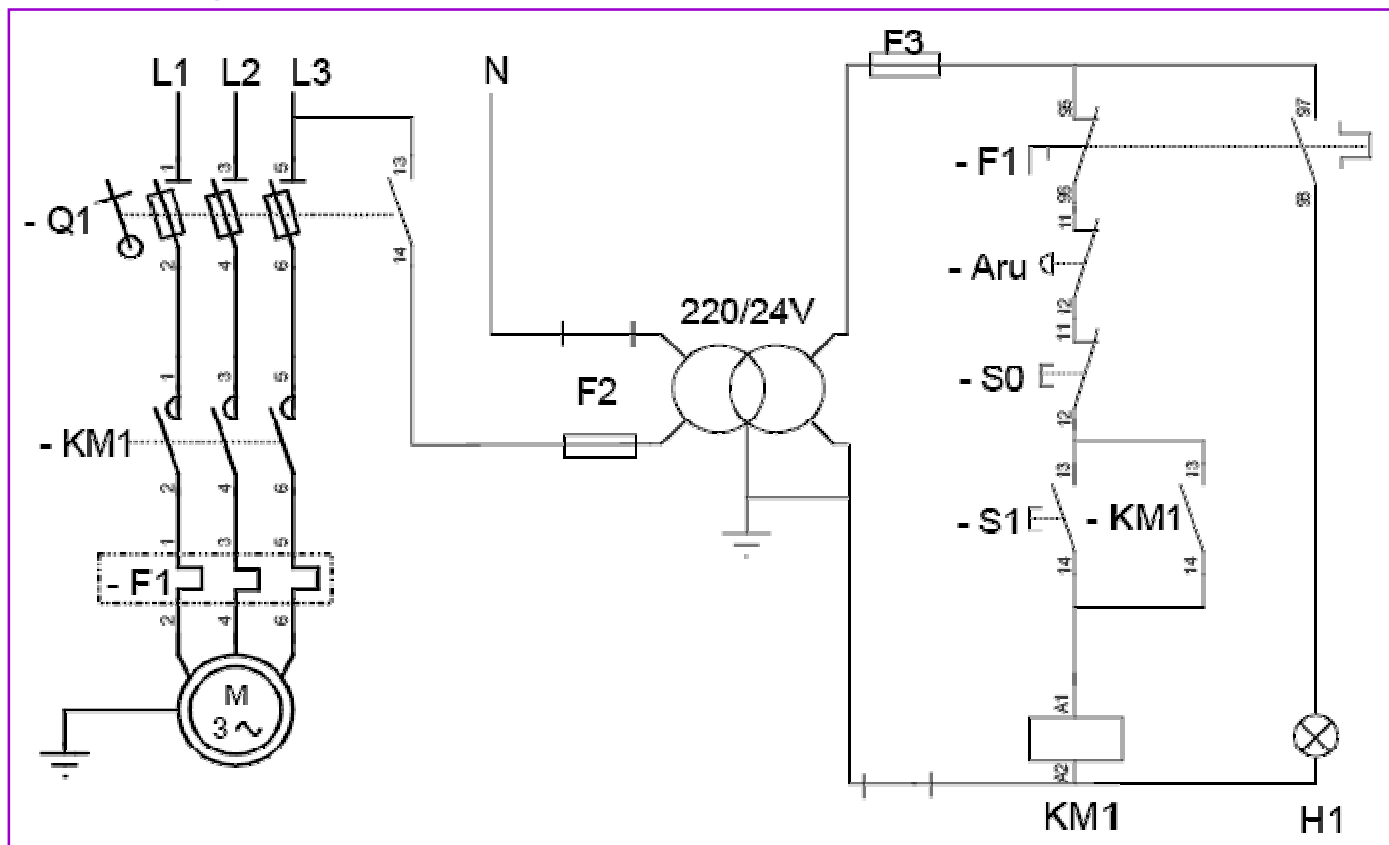
Démarrage un seul sens de rotation

Fig1 Schéma de principe.

Les éléments qui composent les circuits de commande et de puissance sont :

- **L1, L1, L3** : arrivée du réseau triphasé.
- **Q1** : sectionneur porte-fusibles tripolaire équipé de 2 contacts à fermeture
- **KM1** : contacteur tripolaire équipé avec un contact à fermeture
- **F2** : relais de protection thermique
- **F1** : fusible
- **S1, S3** : boutons-poussoirs à ouverture et à retour automatique
- **S2, S4** : boutons-poussoirs à fermeture et à retour automatique
- **S5** : bouton rotatif à fermeture sans retour automatique

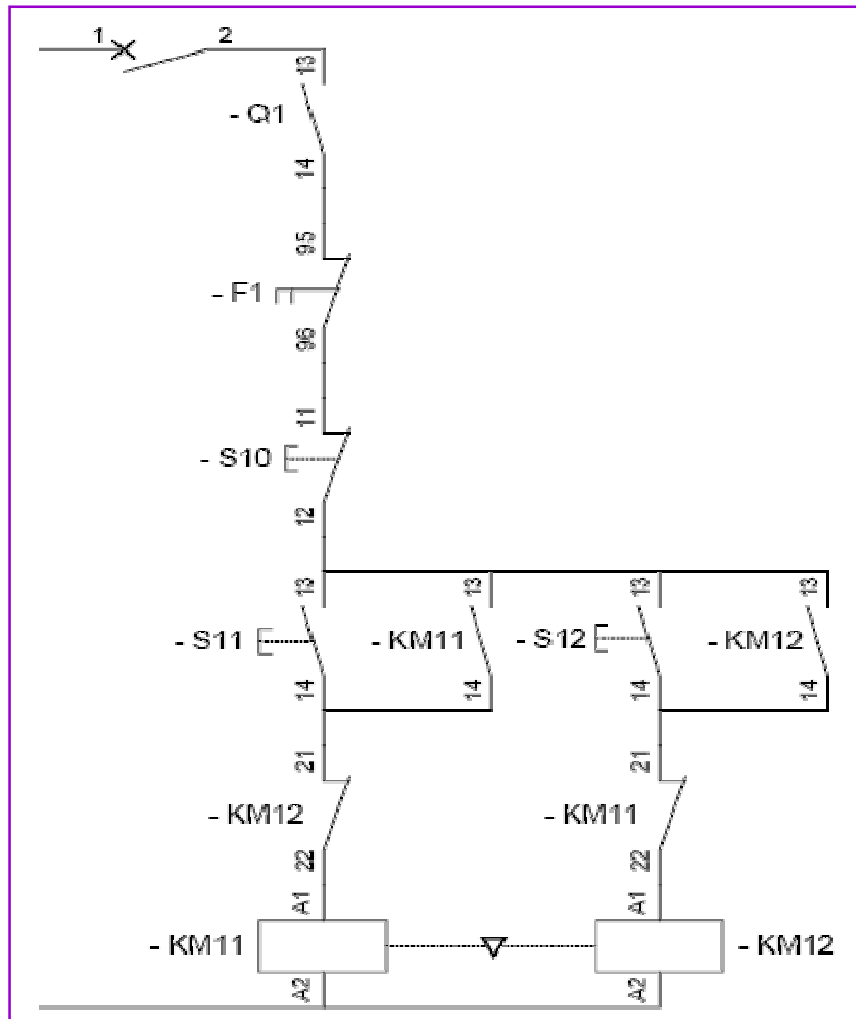
Le circuit de commande est protégé par le fusible F1 et isolé de toute alimentation par le sectionneur **Q1**.

Le circuit avec un seul poste de commande est équipé avec deux boutons-poussoirs:

- **S1** : bouton-poussoir "ARRÊT" (AT)
- **S2** : bouton-poussoir "MARCHE" (MA).

La bobine du contacteur est repérée **KM1**. Le contact KM1est le contact auxiliaire du contacteur KM1 : c'est le contact d'auto-alimentation ou auto-maintien.

- Démarrage deux sens de rotation.



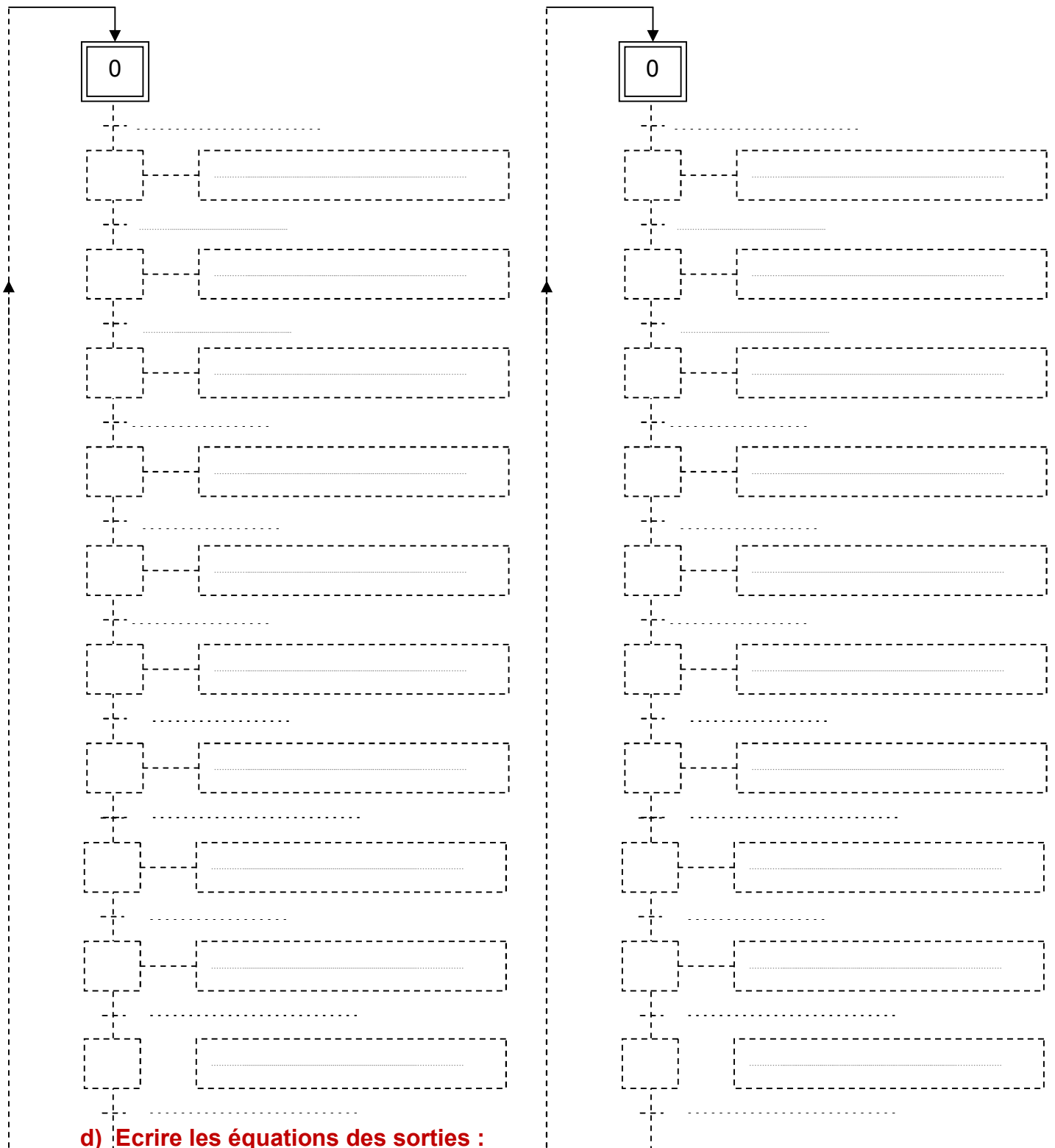
Travail demandé :

- 1) Complétez le tableau des entrées et sorties avec les adresse **API** correspondantes;
- 2) Donnez le **GRAFCET** du point commande et du point de vue automate, fig2 ; fig3;
- 3) Ecrire les équations des sorties (préactionneurs);
- 4) Programmez la solution sous **MILLINIUM3** et sous le **logiciel PL7** : le GRAFCET (**chart**) , les réceptivités et les actions (**programmation postérieur**), ainsi que la **programmation préliminaire** avec **Initialisation de l'étape initiale et remise à zéro du système** ; Complétez le schéma de câblage de l'automate **en logique négative**.

a) Tableau Affectation des entrées / sorties

Désignation	Repère	Affectation adresse API	
.....			Bouton Poussoirs et capteurs
.....			
.....			
.....			
.....			
.....			
.....			
.....			
.....			
.....			
.....			
.....			
.....			Pré actionneurs
.....			
.....			
.....			
.....			
.....			
.....			
.....			
.....			
.....			
.....			

b) GRAFCET point de vue partie commande (PC) : c) GRAFCET point de vue automate (API) :



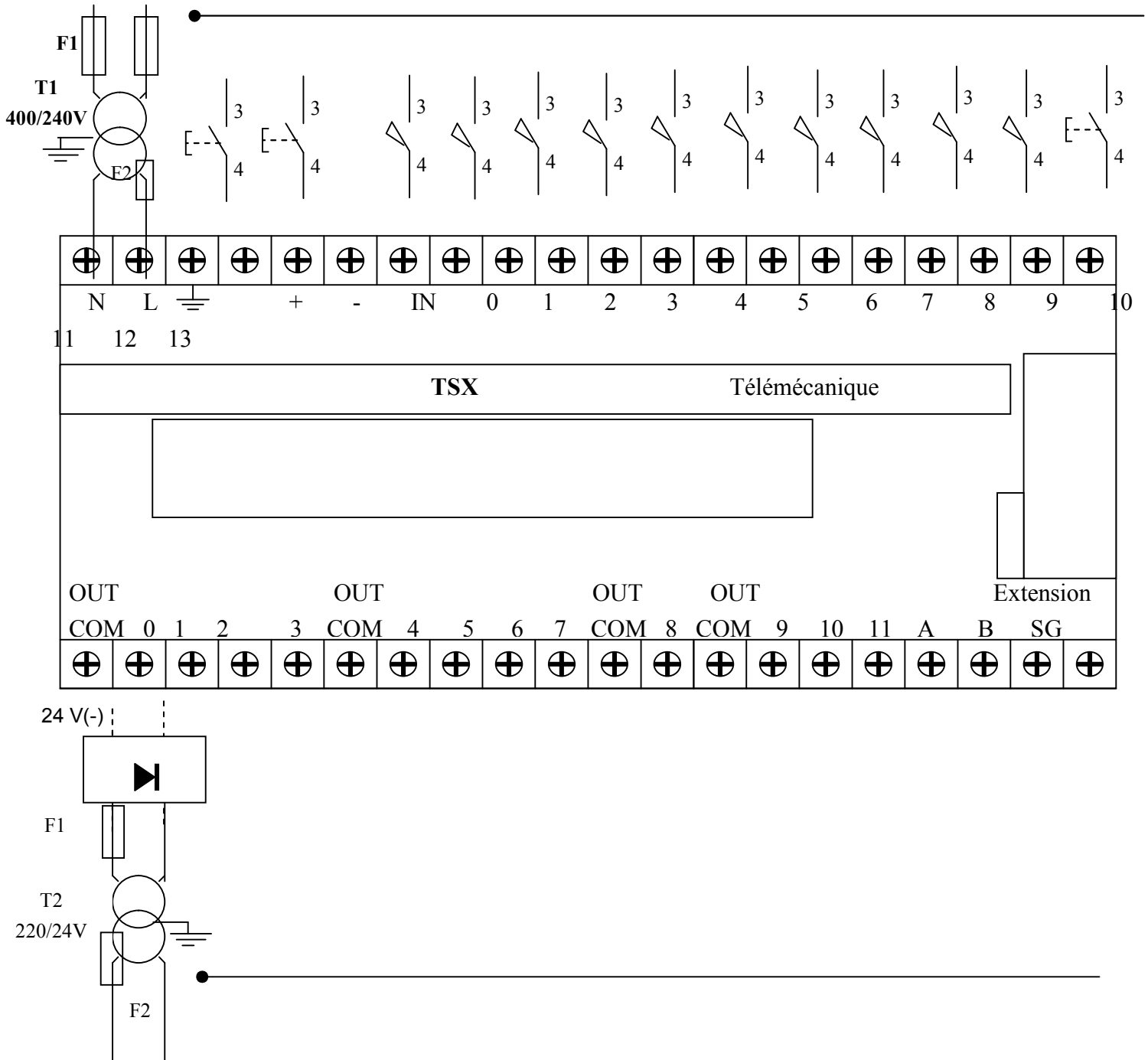
d) Ecrire les équations des sorties :

..... = ...
 = ...
 = ...
 = ...

..... = ...
 = ...
 = ...
 = ...

Programmation Chart		Programmation Préliminaire	
Grafset	Réceptivités (entrées)	Programmation Postérieur	
		Actions (sorties)	
↓	(#)		()
0	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		()
□	(#)		()
+	(#)		

f) Schéma de câblage de l'automate

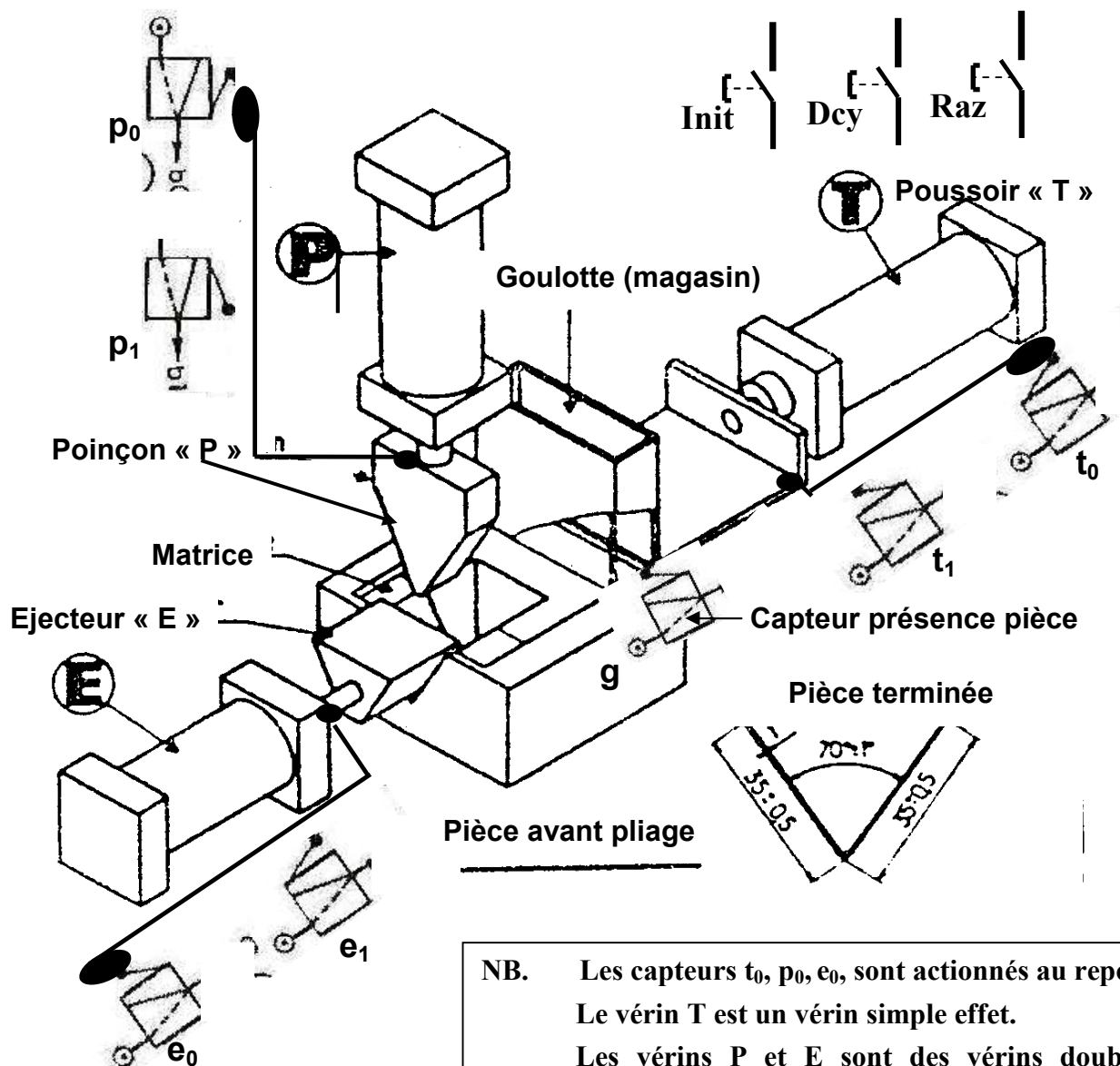


TP N° 6 :

**PROGRAMMATION EN LANGAGE LD ET SFC
SOUS LES LOGICIELS MILLINIUM₃ ET PL7**

1. Objectifs :

- Métriser les langages de programmation des automates programmables LD et SFC,
- Comprendre l'architecture matérielle des automates programmables industriels,
- Analyser la structure cahier de charge industriel et trouver la solution GRAFCET de commande puis la programmée ,
- Programmer et simuler des circuits électrique industriels sous PL7.

2. CAHIER DE CHARGE ETUDE D'UN APPAREIL A PLIER LES TÔLES :

Cet appareil sert à plier des tôles.

Des tôles non pliées sont empilées dans une goulotte (magasin). Un capteur (**g**), détecte la présence de tôles dans la goulotte.

Lorsque l'opérateur actionne le bouton poussoir (**Dcy**), avec présence de pièce (**g**), les actions suivantes se produisent :

- Sortie de la tige du vérin **T** (Le poussoir (**T**) pousse la tôle sous le poinçon).
- Action sur **t₁**, la tige du vérin **P** sort pour plier la tôle.
- Action **p₁**, les tiges des vérins **P** et **T** rentrent.
- Action sur **t₀** et **p₀**, la tige du vérin **E** sort pour éjecter la tôle pliée.
- Action sur **e₁**, la tige du vérin **E** rentre.
- Action sur **e₀**, le cycle se termine.

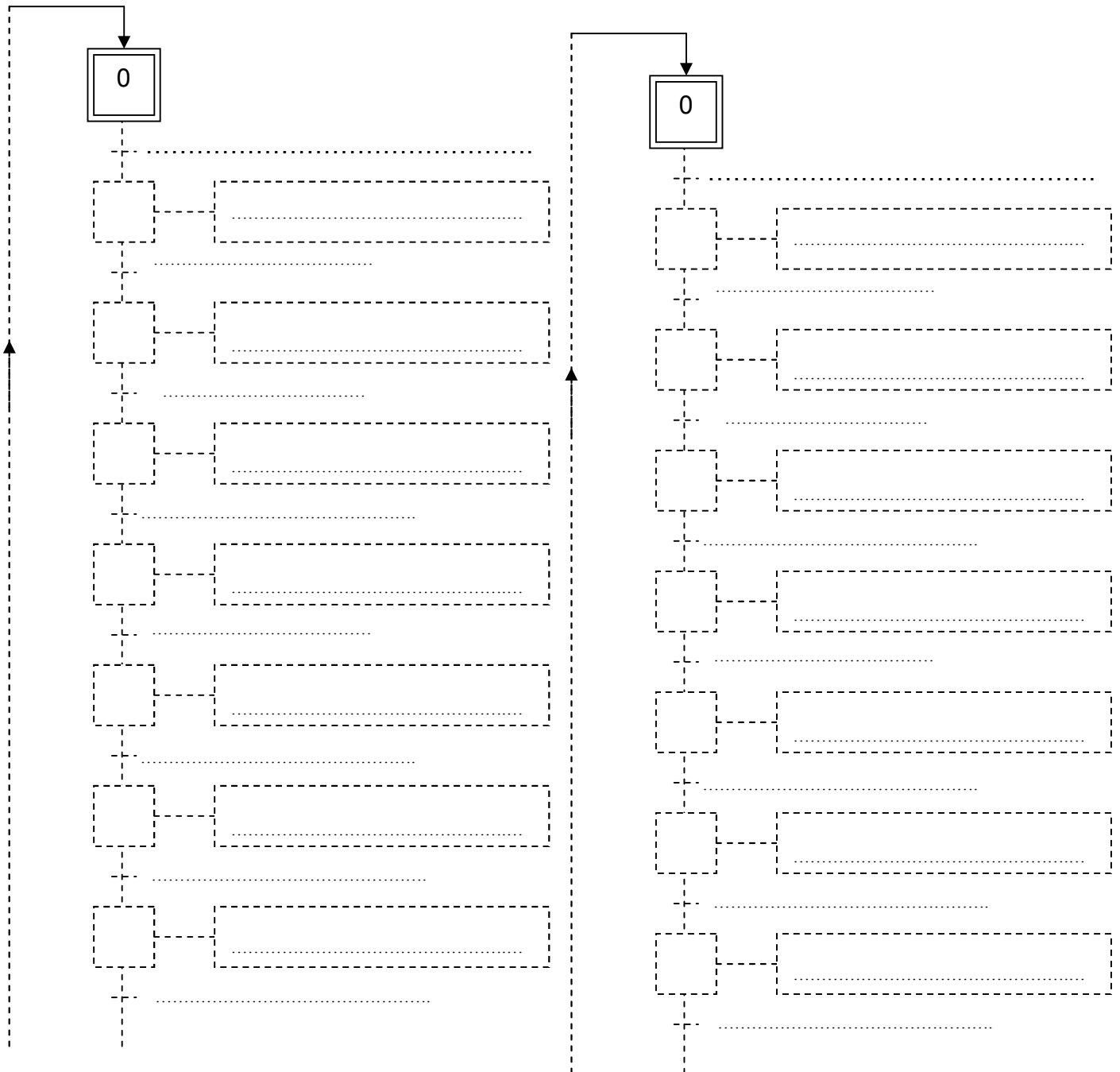
3. Travail demandé :

- 1) Complétez le tableau des entrées et sorties avec les adresse **API correspondantes**;
- 2) Donnez le **GRAFCET** du point commande et du point de vue automate, fig2 ; fig3;
- 3) Ecrire les équations des sorties (préactionneurs);
- 4) Programmez selon le **logiciel PL7** le GRAFCET (**chart**) , les réceptivités et les actions (**programmation postérieur**), ainsi que la **programmation préliminaire** avec Initialisation de l'étape initiale et remise à zéro du système ; Complétez le schéma de câblage de l'automate **en logique positive**.

c) Tableau Affectation des entrées / sorties

Désignation	Repère	Affectation adresse API	
Présence pièce	g		Bouton Poussoirs et capteurs
Bouton Poussoir départ cycle	Dcy		
Fin de course rentrée de tige vérin T	t0		
Fin de course sortie tige vérin T	t1		
Fin de course rentrée.	p0		
.....			
.....			Pré actionneurs
Sortie tige Vérin de poinçonnage P			
.....			
.....			
.....			
.....			

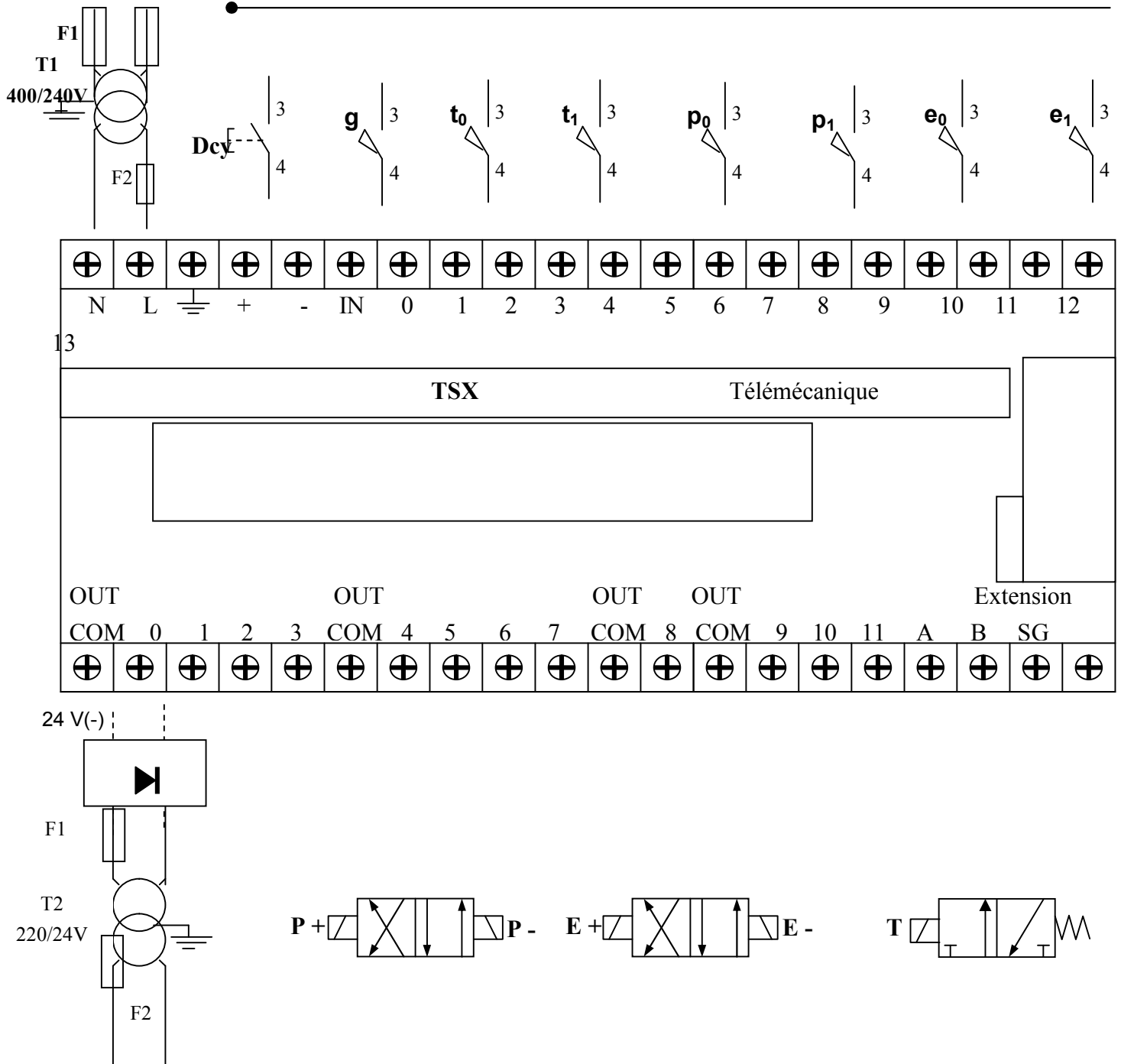
d) GRAFCET point de vue partie commande (PC) : c) GRAFCET point de vue automate (API) :



d) Ecrire les équations des sorties :

..... = ...	 = ...
..... = ...	 = ...
..... = ...	 = ...
..... = ...	 = ...
..... = ...	 = ...

g) Schéma de câblage de l'automate



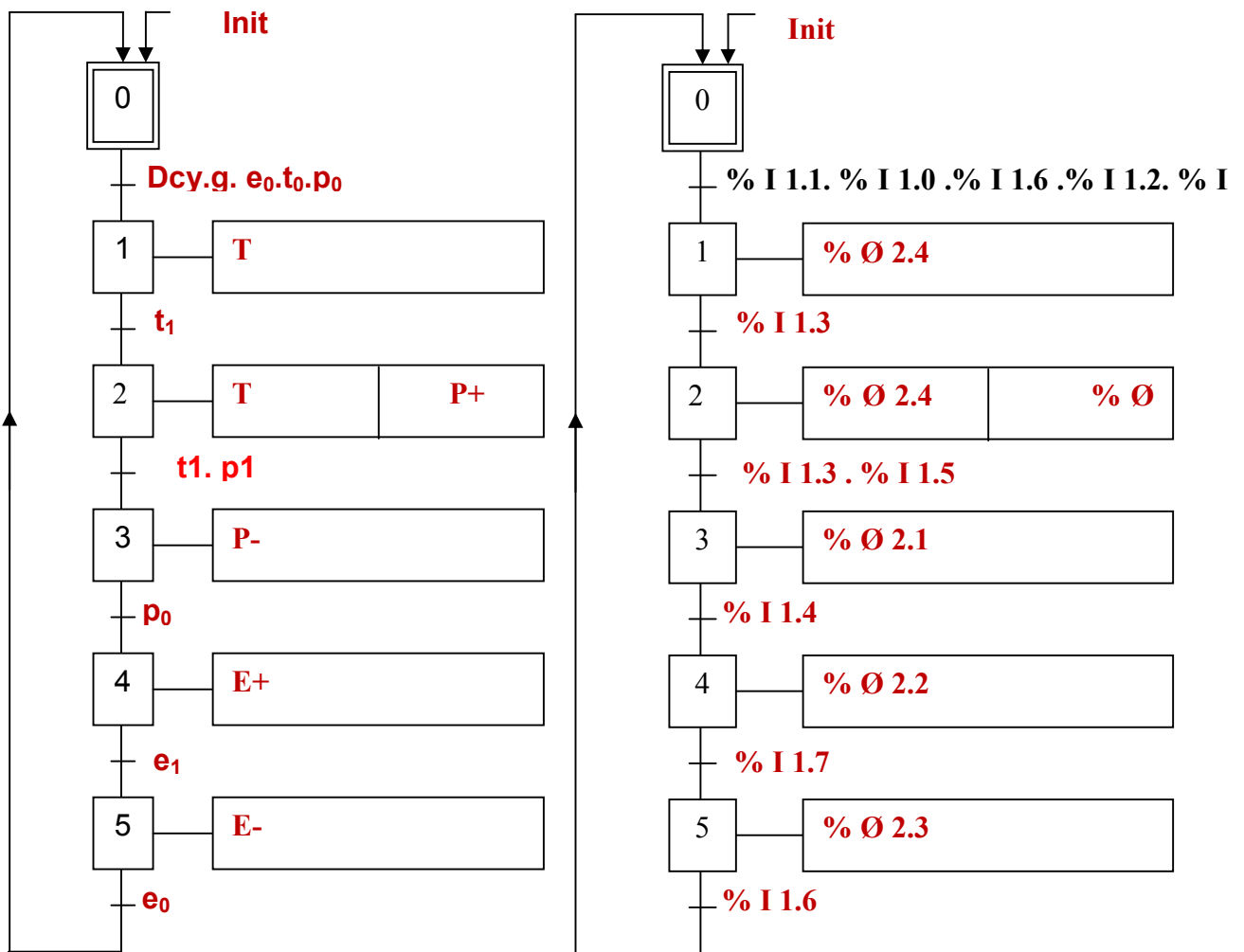
SOLUTION TP 6 : ETUDE ET COMMANDE PAR API D'UN APPAREIL À PLIER LES TÔLES

➤ Solution graphique par GRAFCET

a) Tableau d'adressage des entrées et sorties de l'API.

Désignation	Repère	Affectation adresse API	
Présence pièce	g	% I 1.0	Boutons Poussoirs et capteurs
Bouton Poussoir départ cycle	Dcy	% I 1.1	
Fin de course rentrée de tige vérin T	t₀	% I 1.2	
Fin de course sortie tige vérin T	t₁	% I 1.3	
Fin de course rentrée tige vérin P	p₀	% I 1.4	
Fin de course sortie tige vérin P	p₁	% I 1.5	
Fin de course rentrée tige vérin E	e₀	% I 1.6	
Fin de course sortie tige vérin E	e₁	% I 1.7	
P : Vérin de pliage de la tôle	P+	%Q 2.0	Pré actionneurs
	P-	%Q 2.1	
E : Vérin éjecteur de la tôle pliée	E+	%Q 2.2	
	E-	%Q 2.3	
T : Vérin de translation de la tôle	T	%Q 2.4	

b) GRAFCET point de vue partie commande (PC) : c) GRAFCET point de vue API



d) Ecrire les équations des sorties :

$$P+ . = \%X2$$

$$P- = \%X3$$

$$E+ . = \%X4$$

$$E- = \%X5$$

$$T . = \%X1 + \%X2$$

On peut aussi écrire directement les sortie selon leurs adresse API

$$\% \text{ Ø } 2.0 . = \%X2$$

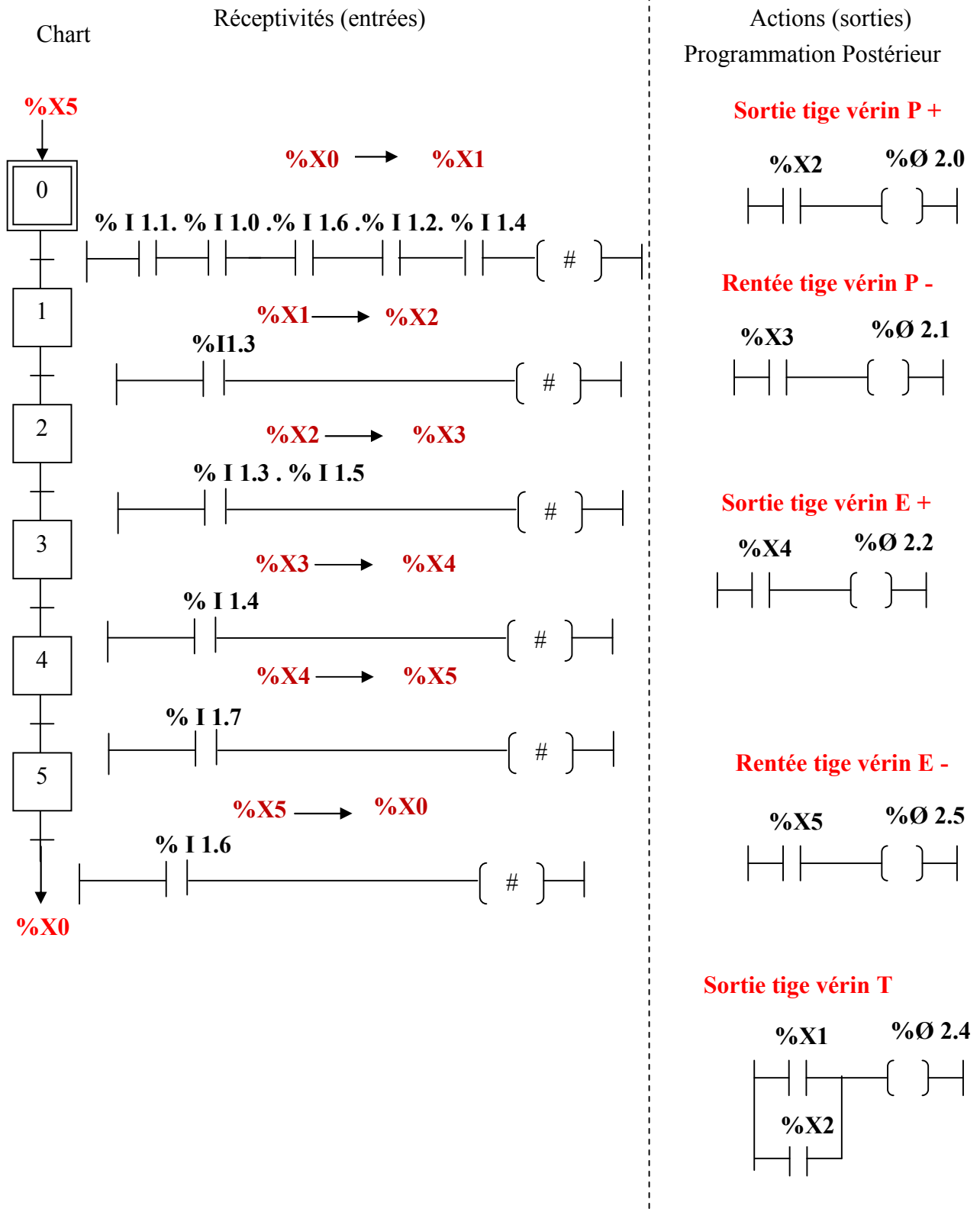
$$\% \text{ Ø } 2.1 = \%X3$$

$$\% \text{ Ø } 2.2 = \%X4$$

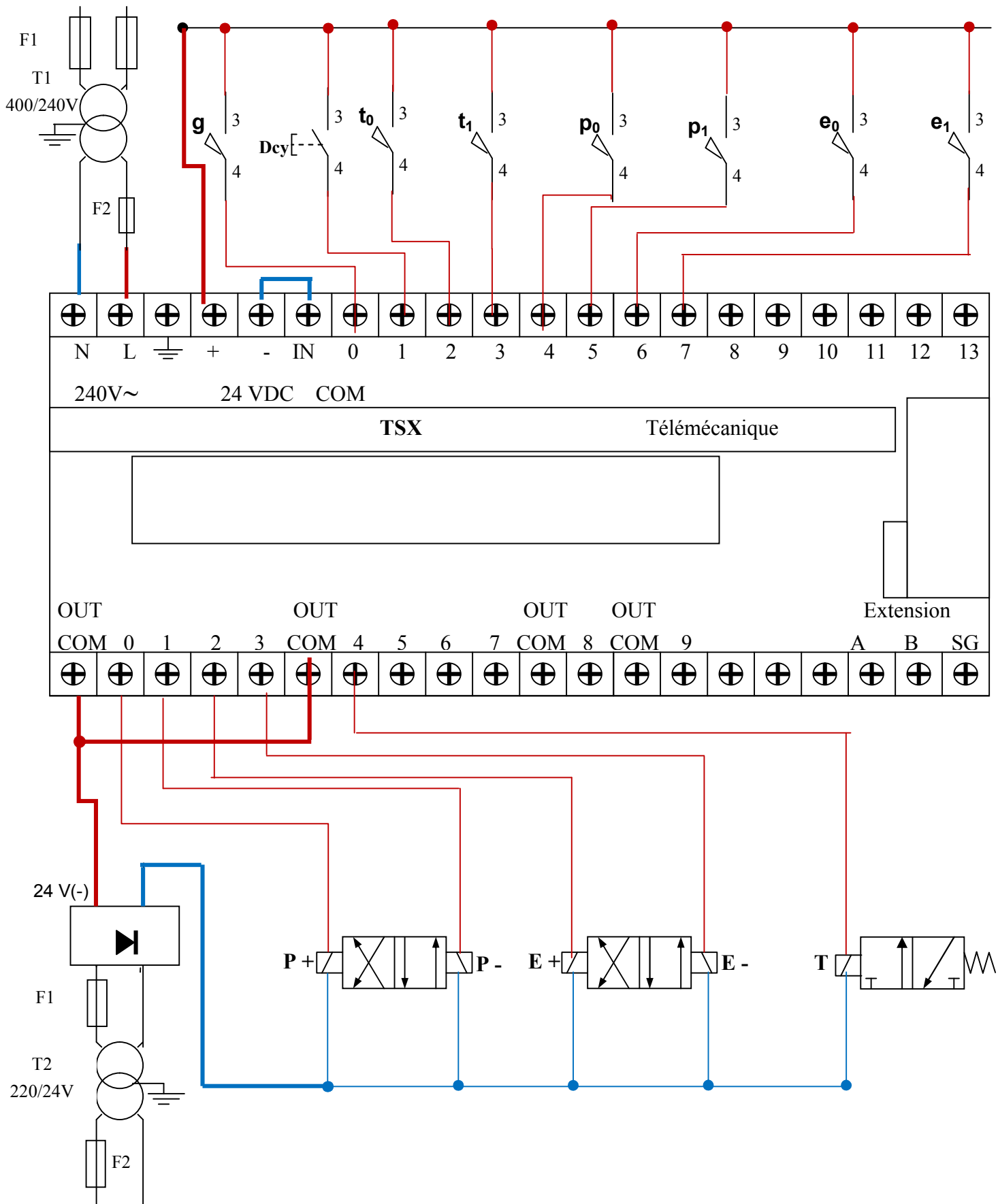
$$\% \text{ Ø } 2.3 = \%X5$$

$$\% \text{ Ø } 2.4 = \%X1 + \%X2$$

e) Document réponse programmation de réceptivités et action en langage GRAFCET sous PL7,



e) Schéma de câblage de l'API en logique positive



Références bibliographiques

- [1] : Henry NEY, Automatique et informatique industriel, éditions Nathan, Paris 1998 ;
- [2] : Pierre BOYE André BIANCIOTTO, Le schéma en électrotechnique, éditions Delagrave Paris 1982 ;
- [3] : Henry NEY, Eléments d'automatismes, éditions Nathan, Paris 1985 ;
- [4] : F DEGOULANGE, R LEMAITRE, D PERRIN, Automatismes , éditions Technique et vulgarisation ;
- [5] : J MONTAGNAC, Cours de schémas automatisme – électricité, éditions Dunod, Paris 1985 ;
- [6] : Philippe RAYMOND, Les Automates Programmables Industriels (API), Notes de cours 2005 ;
- [7] : R. CHAPPERT A. CAMPA L. THIBERVILLE, L'automatique par les problèmes, tome2, éditions Foucher Paris 1979 ;
- [8] : Robert STRANDH Irène DURAND, Architecture de l'ordinateur, éditions Dunod, Paris, 2005 ;
- [9] : T. DUMARTIN ; Architecture des ordinateurs, Note de cours, 2004-2005 ;
- [10] : D. BLIN J. DANIC R. Le GARREC F. TROLE J.C SEITE, Les automatismes, éditions Casteilla, Paris 1984 ;
- [11] : C. BOURBOUNNE J. COJEAN, Les systèmes automatisés de connaissance à la conception, Tome 2, éditions Foucher Paris 1981 ;
- [12] : A. RIDEAU A. BIANCIOTTO P. BOYE, La technologie des systèmes automatisés, éditions Delagrave Paris 1997 ;
- [13] : D. BOUTEILLE N. BOUTEILLE S. CHANTREUIL R. COLLOT A. SFAR, Les automatismes programmables, éditions Cepadues, Toulouse 1988 .