



Université Mohamed BOUDIAF - M'sila
Faculté de technologies
Département Génie Electrique



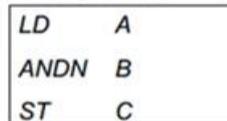
Module : **Automate Programmable Industriel** Enseignant : **O. Zouggar**

Classe : **Master**

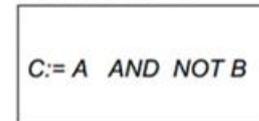
Programmation d'un automate et langage normalisé IEC 1131-3

The 5 Languages of IEC 1131-3

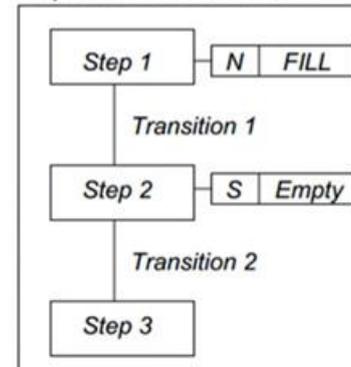
Instruction List



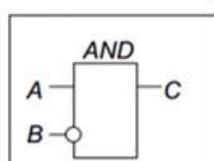
Structured Text



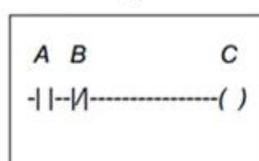
Sequential Function Chart



Function Block Diagram



Ladder Diagram



- I. C'est quoi un programme ? 3
- II. Définition : 3
- III. Langages de programmation 3
 - III.1. Les langages graphiques : 4
 - a. Langage Ladder : 4
 - b. Langage FBD (Function Block Diagram ou Logigramme) : 7
 - c. Langage SFC (Sequential function char) : Le Grafcet : 9
 - III.2. Les langages textuels : 10
 - a. Langage IL (LIST) : 10
 - b. SCL (Structured Control Langage) : 11

Figure 3-1: Les cinq langages de programmation.....	4
Figure 3-2: Temporisation	5
Figure 3-3: Compteur incrémental.....	6
Figure 4 : Exemple d'une structure d'un réseau de contacts	6
Figure 5 : Temporisation (Siemens).....	8
Figure 6 : Compteurs (Siemens)	8
Figure 3-7: exemple d'un programme GRAFCET du fonctionnement d'un vérin.....	10
Figure 3-8: Exemple d'allumage d'une LED sous langage à List	11

I. C'est quoi un programme ?

Un programme est une unité logique de programmation qui décrit des opérations entre les variables de l'application. Un programme décrit des opérations séquentielles ou cycliques. Un programme cyclique est exécuté systématiquement à chaque cycle automate. L'exécution d'un programme séquentiel respecte les règles d'évolution du langage SFC.

II. Définition :

La norme IEC 1131-3, est une norme industrielle de la Commission électrotechnique internationale (**International Electrotechnical Commission** en anglais) (IEC). Publiée la première fois en 1993, sa dernière édition date de septembre 2013.

La norme de programmation IEC 1131-3 est devenue obligatoire pour l'ensemble des constructeurs depuis 1999. Actuellement, elle est la référence sur le marché de l'automatisme. La maîtrise de cette norme est nécessaire pour réaliser les applications portables et réutilisables sur les différentes plateformes de l'automatisation.

III. Langages de programmation

La norme IEC 1131-3 définit entre autres choses, cinq langages qui peuvent être utilisés pour la programmation d'applications d'automatisme. Les cinq langages sont :

- SFC (« sequential function chart ») : issu du langage GRAFCET, ce langage, de haut niveau, permet la programmation aisée de tous les procédés séquentiels ;
- FBD (« function block diagram », ou schéma par blocs) : ce langage permet de programmer graphiquement à l'aide de blocs, représentant des variables, des opérateurs ou des fonctions. Il permet de manipuler tous les types de variables ;
- LD (« ladder diagram », ou schéma à relais) : ce langage graphique est essentiellement dédié à la programmation d'équations booléennes (true/false) ;
- ST (« structured text » ou texte structuré) : ce langage est un langage textuel de haut niveau. Il permet la programmation de tout type d'algorithme plus ou moins complexe ;
- IL (« instruction list », ou liste d'instructions) : ce langage textuel de bas niveau est un langage à une instruction par ligne. Il peut être comparé au langage assembleur.

The 5 Languages of IEC 1131-3

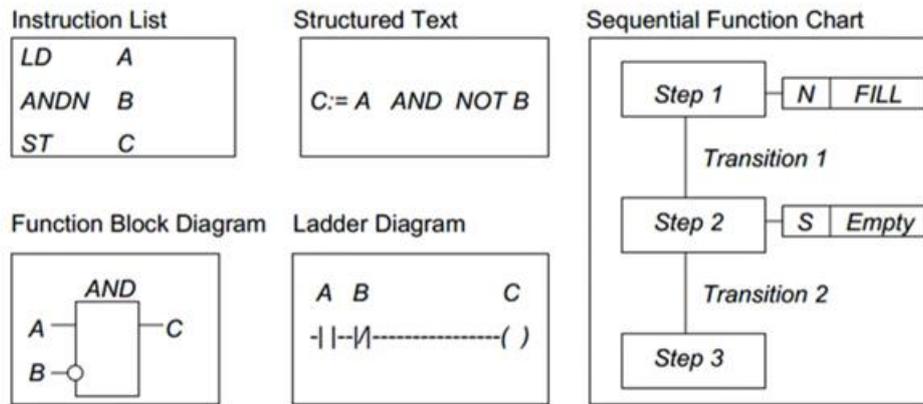


Figure III-1: Les cinq langages de programmation

III.1. Les langages graphiques :

a. Langage Ladder :

Le langage LD (ladder diagram) est une représentation graphique d'équations booléennes combinant des contacts (les entrée) et des relais (les sortie). Il permet la manipulation de données booléennes, à l'aide de symboles graphiques organisés dans un diagramme comme les éléments d'un schéma électrique à contacts.

L'idée initiale du Ladder est la représentation de fonction logique sous la forme de schémas électriques. Cette représentation est originalement matérielle : quand l'Automate Programmable Industriel n'existait pas, les fonctions étaient réalisées par des câblages. Par exemple, pour réaliser un ET logique avec des interrupteurs, il suffit de les mettre en série. Pour réaliser un OU logique, il faut les mettre en parallèle. Partant de ces principes, le Ladder a été créé et normalisé dans la norme IEC 1131-3. Il est, depuis, très utilisé dans la programmation des Automates Programmables Industriels. Un programme Ladder se lit de haut en bas et l'évaluation des valeurs se fait de gauche à droite.

Les valeurs correspondent en fait, si on le compare à un schéma électrique, à la présence ou non d'un potentiel électrique a chaque nœud de connexion. En effet, le Ladder est basé sur le principe d'une alimentation en tension représentée par deux traits verticaux reliée horizontalement par des bobines, des contacts et des blocs fonctionnels. Il est composé de réseaux lus les uns à la suite des autres par l'automate. Ces réseaux sont constitués de divers symboles représentant les entrées/sorties de l'automate, les opérateurs séquentiels (temporisations, compteurs, ...), les opérations, ainsi que les bits systèmes internes à l'automate

Chapitre 2 : Programmation d'un automate et langage normalisé IEC 1131-3

Les composants du langage

Il existe 3 types d'élément de langage :

- Les entrées (ou contact), qui permettent de lire la valeur d'une variable booléenne,
- Les sorties (ou bobines) qui permettent d'écrire la valeur d'une variable booléenne,
- Les blocs fonctionnels qui permettent de réaliser des fonctions avancées.

Le tableau ci-dessus donne les principaux éléments (contacte et bobines) d'un réseau LD.

Object graphique API	Nom
- -	Contact normalement ouvert
- / -	Contact normalement fermé
- P -	Contact fermé au front montant
- N -	Contact fermé au front descendant
-()-	Bobine normalement ouverte
-(/)-	Bobine normalement fermée
-(S)- ou -(L)-	Le bit interne maintenu à 1 une fois actionné
-(R)- ou -(U)-	Le bit interne remise à 0 une fois actionné

Tableau 1: les principaux éléments (contacte et bobines) d'un réseau LD.

- Les blocs fonctionnels qui permettent de réaliser des fonctions avancées (temporisation, comptage, etc.).

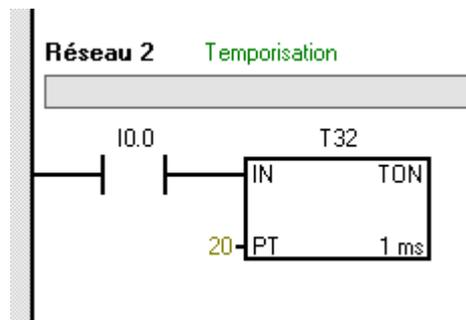


Figure III-2: Temporisation

L'opération Démarrer temporisation sous forme de retard à la montée s'écoule jusqu'à la valeur maximale lorsqu'elle est activée. Lorsque la valeur en cours « T » est supérieure ou égale à la valeur prédéfinie PT, le bit de temporisation T est activé. La temporisation « retard à la montée » est remise à zéro lors de sa désactivation. Cette temporisation s'arrête lorsque sa valeur maximale est atteinte.

Chapitre 2 : Programmation d'un automate et langage normalisé IEC 1131-3

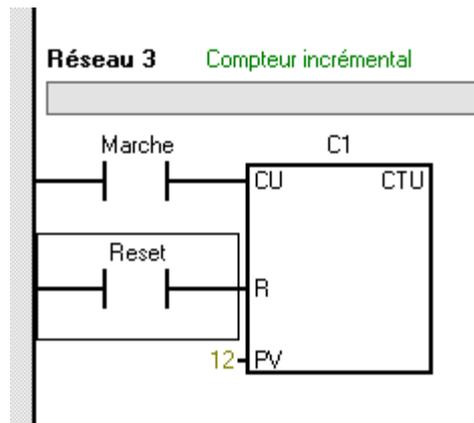


Figure III-3: Compteur incrémental

L'opération Compteur incrémental incrémente jusqu'à la valeur maximale en cas de front montant à l'entrée d'incrémementation CU. Lorsque la valeur en cours « C » est supérieure ou égale à la valeur prédéfinie PV, le bit de compteur C est activé. Le compteur est remis à zéro lorsque l'entrée de remise à zéro R est activée.

Structure d'un réseau de contacts :

Un réseau de contacts se compose de la manière suivante : étiquette (ou titre) + Commentaire + réseau graphique (zone test + zone action).

La zone de test accueille :

- les contacts.
- les blocs fonction (temporisations, compteurs, ...).
- les blocs comparaison.

La zone action accueille :

- Les bobines.
- Les blocs opérations.

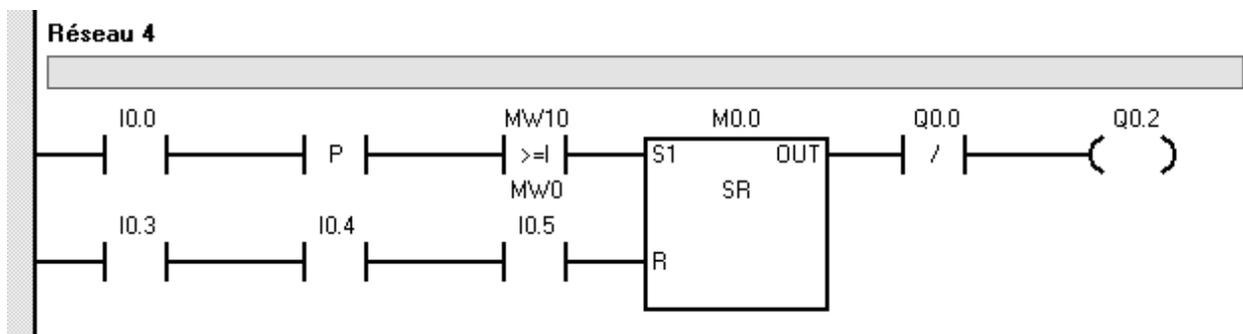


Figure 4 : Exemple d'une structure d'un réseau de contacts

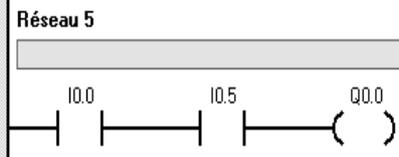
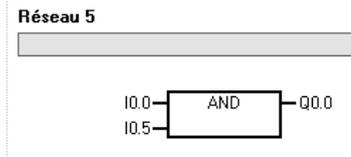
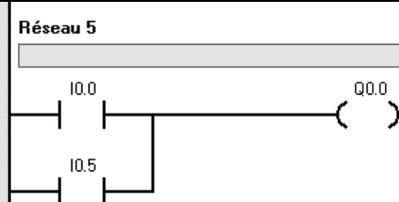
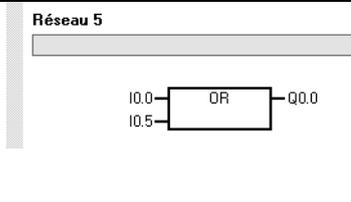
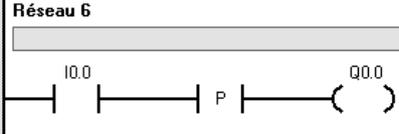
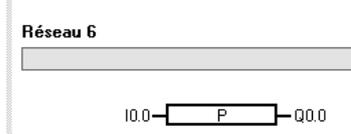
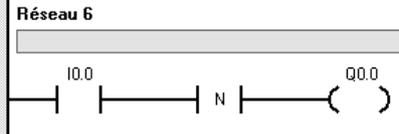
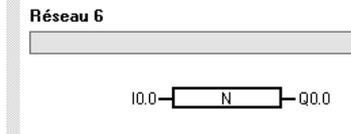
Chapitre 2 : Programmation d'un automate et langage normalisé IEC 1131-3

b. Langage FBD (Function Block Diagram ou Logigramme) :

Langage FBD est un langage de programmation graphique qui utilise les blocs de l'algèbre de Boole pour représenter les opérations logiques. Un réseau LOG est composé d'une ou plusieurs boîtes d'opérations LOG. Au lieu d'utiliser des contacts, on affecte une ou plusieurs valeurs binaires comme entrées à une boîte d'opération LOG. Vous utilisez les sorties de l'opération pour connecter cette dernière à une opération consécutive ou pour achever le réseau. Ainsi, une seule opération LOG peut représenter la même fonction qu'un ensemble de contacts, bobines ou boîtes en schéma à contacts. Le réseau est achevé lorsque vous avez procédé à l'affectation de tous les paramètres de l'opération ou que vous les avez connectés à une autre opération

Les symboles utilisés :

Les opérations sur bits :

Langage LD	Langage LOG	Fonction
<p>Réseau 5</p> 	<p>Réseau 5</p> 	Cette boîte représente la fonction ET en associant deux bits que l'on peut inverser à l'entrée de la boîte.
<p>Réseau 5</p> 	<p>Réseau 5</p> 	Cette boîte représente la fonction OU en associant deux bits que l'on peut inverser à l'entrée de la boîte.
<p>Réseau 6</p> 	<p>Réseau 6</p> 	Contact à front montant
<p>Réseau 6</p> 	<p>Réseau 6</p> 	Contact à front descendant.

Les circuits séquentiels :

- **Temporisation** : Il existe trois types de temporisations :
 - TON (retard à la montée),
 - TONR (retard à la montée temporisé),
 - TOF (retard à la descente).

Chapitre 2 : Programmation d'un automate et langage normalisé IEC 1131-3

- IN est l'entrée de validation et PT le temps prédéfini.
- La base de temps dépend du numéro de la temporisation.

Réseau 2 Temporisation

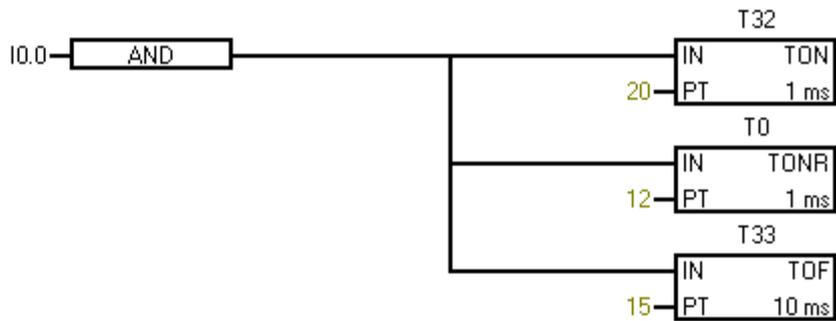
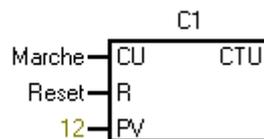


Figure 5 : Temporisation (Siemens)

- **Compteur** : Il existe trois principaux types de compteurs/décompteurs :
 - CTU (compteur incrémental),
 - CTD (compteur décrémental),
 - CTUD (compteur incrémental/décrémental).
 - PV est la valeur prédéfinie et R la remise à zéro.
 - L'entrée d'incrémentation s'appelle CU et l'entrée de décrémentation CD.

Réseau 5 Compteur incrémental



Mnémorique	Adresse	Commentaire
Marche	I0.1	Marche compteur
Reset	I0.2	mise à zero la valeur du compteur

Réseau 6 Compteur décrémental

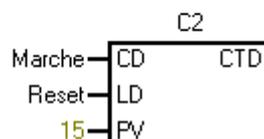


Figure 6 : Compteurs (Siemens)

Les blocs comparaison :

Ces blocs permettent de comparer des nombres, des bits, des octets ou des mots en supériorité, infériorité ou égalité.

Chapitre 2 : Programmation d'un automate et langage normalisé IEC 1131-3

Les opérations d'exécution :

Ces blocs regroupent plusieurs fonctions établissant l'ordre d'exécution du programme : saut de programme, retour de sous-programme, fin de programme, ...

c. Langage SFC (Sequential function char) : Le Grafset :

Le GRAFCET c'est un outil graphique de description du comportement déterministe de la Partie Commande, il décrit les interactions informationnelles à caractère déterministe à travers la frontière d'isolement entre la partie commande et la partie opérative d'un système isolé, il est utilisé pour la description de procédures séquentielles à séquences alternatives ou parallèles.

Les procédures se configurent et se programment clairement et rapidement dans un mode de visualisation standardisé, le processus se décrit graphiquement en se décomposant en différentes étapes à fonctions élémentaires.

- Structure flexible des chaînes séquentielles : Branchements simultanés ou alternatifs, sauts à l'intérieur des chaînes séquentielles, activation et désactivation d'étapes.
- Exécution sélective des étapes : Le temps d'exécution d'une chaîne séquentielle est ainsi fonction du nombre d'étapes.
- Synchronisation des modes automatique et manuel : le processus n'est plus alors synchronisé quand il a été amené dans un autre état en mode manuel. GRAPH supporte la recherche de points de synchronisation en vue de la reprise du mode automatique. Les étapes pertinentes sont à cet effet repérées. Les critères définissables peuvent être des transitions ou des verrouillages.
- Visualisation étape par étape donnant un aperçu de tous les détails d'une étape Gain de temps important par rapport à la programmation en CONT/LOG/LIST.

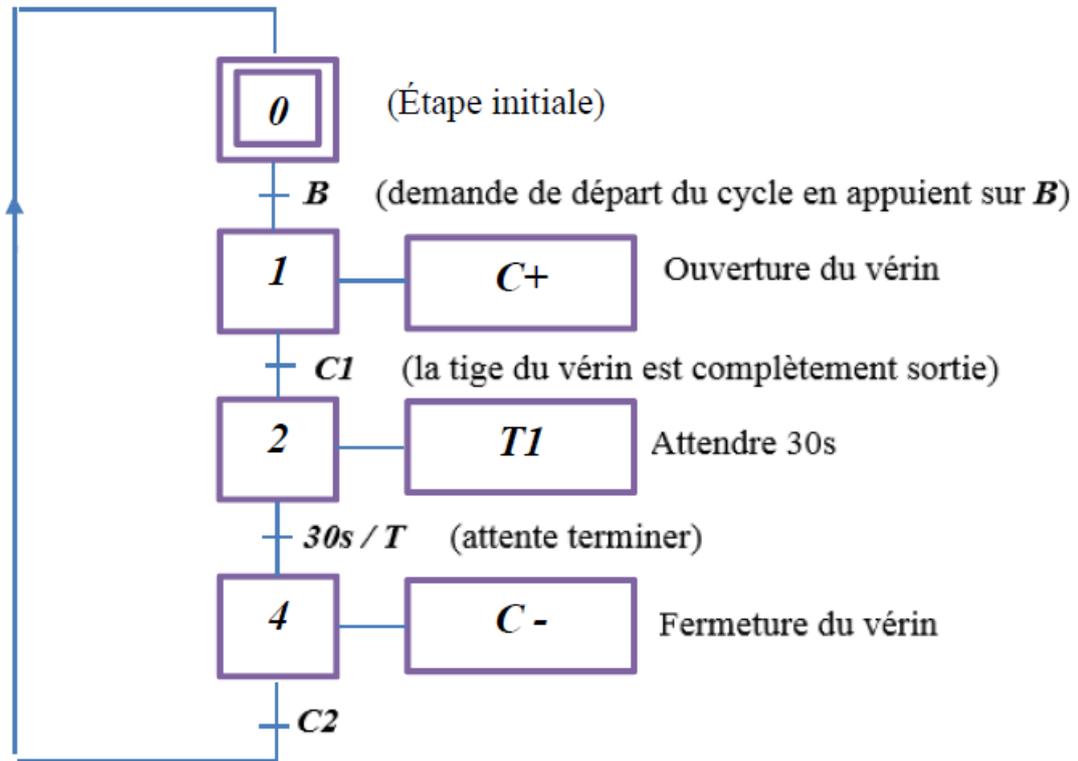


Figure III-7: exemple d'un programme GRAFCET du fonctionnement d'un vérin

III.2. Les langages textuels :

a. Langage IL (LIST) :

Définition :

Le langage de programmation textuelle LIST permet de créer des programmes d'applications à un niveau proche du matériel et en optimisant le temps d'exécution et la place en mémoire.

Il est l'abréviation de liste d'instructions. C'est un langage de liste d'instructions (**LIST**) est un langage de programmation textuel proche de la machine.

Dans un programme **LIST**, les différentes instructions correspondent, dans une large mesure, aux étapes par lesquelles la **CPU** traite le programme. Pour faciliter la programmation, **LIST** a été complété par quelques structures de langage évolué (comme par exemple, des paramètres de blocs et accès structurés aux données).

Tout programme écrit en CONT ou en LOG peut être réécrit en LIST.

Chapitre 2 : Programmation d'un automate et langage normalisé IEC 1131-3

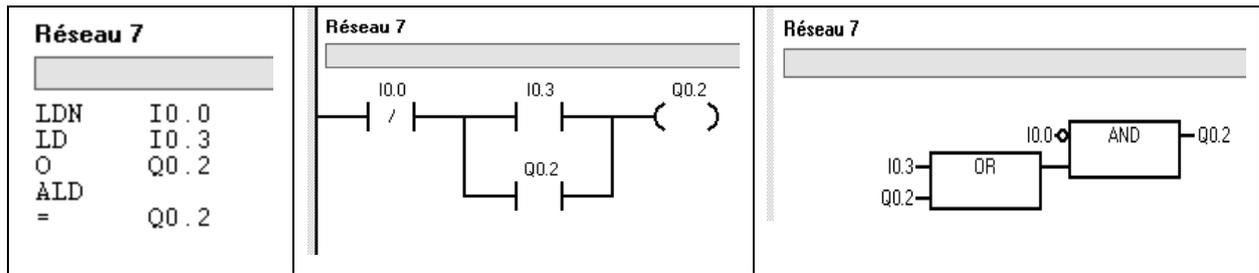


Figure III-8: Exemple d'allumage d'une LED sous langage à List

Les instructions de base :

Instructions de test :

- LD : contact normalement ouvert.
- LDN : contact normalement fermé.
- LDR ou EU : contact à front montant.
- LDF ou ED : contact à front descendant.
- AND : liaison série (ET) à un contact normalement ouvert.
- ANDN : liaison série (ET) à un contact normalement fermé.
- ANDR : liaison série (ET) à un contact à front montant.
- ANDF : liaison série (ET) à un contact à front descendant.
- OR : liaison parallèle (OU) à un contact normalement ouvert.
- ORN : liaison parallèle (OU) à un contact normalement fermé.
- ORR : liaison parallèle (OU) à un contact à front montant.
- ORF : liaison parallèle (OU) à un contact à front descendant.

Instructions d'action :

- ST : bobine directe. L'objet bit associé prend la valeur du résultat de la zone test.
- STN : bobine inversée. L'objet bit associé prend la valeur inversée du résultat de la zone test.
- S : bobine d'enclenchement. L'objet bit associé est mis à 1 lorsque la valeur du résultat de la zone test est à 1.
- R : bobine de déclenchement. L'objet bit associé est mis à 0 lorsque la valeur du résultat de la zone test est à 1.

b. SCL (Structured Control Langage) :

Programmation d'algorithmes complexes

Ce langage est un langage textuel de haut niveau ST (Structured Text), dédié aux applications d'automatisme. SCL convient notamment à la programmation rapide d'algorithmes complexes et de fonctions mathématiques ou à des missions relevant du domaine du traitement des données difficilement modélisables avec les langages graphiques.

Chapitre 2 : Programmation d'un automate et langage normalisé IEC 1131-3

Le code SCL est plus simple, plus court et plus clair, ce qui en facilite et en accélère l'écriture et la manipulation.