Chapitre 102

API(PLC)

Objectifs

1. Expliquer comment fonctionne une API
2. Expiquer les pricipes du grafcet

Dans ce chapitre nous allons traiter du fonctionnement du Programmable Logic Controller. Ce qui signifie que nous allons regarder uniquement le hardware.Cela signifie plus concrètement que nous allons étudier le fonctionnement d'un micro-ordinateur. En plus, nous allons aussi regarder comment fonctionne la communication entre les diffèrentes parties d'une telle machine et entre des machines muttuellement, c'est ce qu'on appelle les systèmes de bus. Le programmation d'un PLC (le logiciel) est fait en laboratoire.Le logiciel est évoqué dans ce chapitre seulement où il est nécessaire de l'évoquer. Nous allons étudier les principes de base de programmation ce qu'on apelle le grafcet.

102.1 Déscription générale

En fait un PLC fait exactement la même chose qu'une commande par relais,c'est à dire :

1. Prendre l'information d'entrée par le panneau de commande
2. Combiner l'information d'entrée par une logique concrete,qui est mise dans le programme
3. Déduire de ceci les commandes de sortie et les diriger dehors

La figure ci dessous vous donne une représentation d'un PLC.

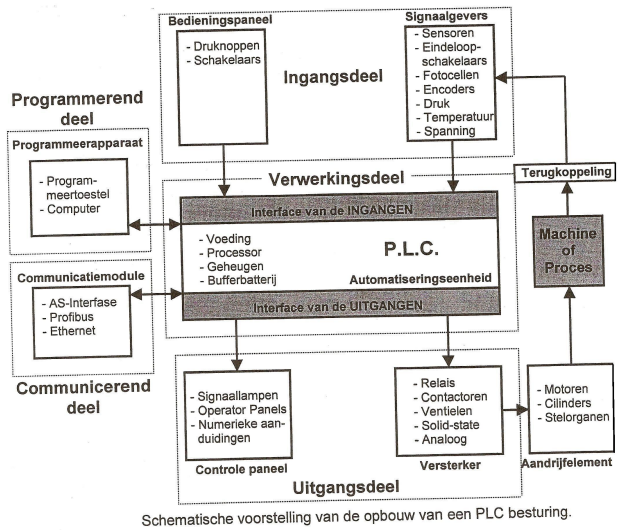


FIGURE 102.1 – source : cursus PLC, APHogeschool

Les flèches donnent la direction de l'échange d'information entre la partie d'entrée, la partie de sortie, la partie de commande et la partie de programmation.

102.1.1 Propriétés générales

1. La partie d'entrée :C'est par ici que passe le signal d'enregistrement de l'information de la console de commande ainsi que des detecteurs de la machine qui sont connecté aux bornes des modules d'entrée.Ceci est aussi valable pour toute autre information d'entrée comme les informations de mesure où les alarmes.
2. La partie de traitement :Le traitement passe par un processeur. Celui-ci est composé d'un centre de traitement (CPU), une mémoire de système interne (ROM), une mémoire de données et d'une mémoire de programmes (RAM,EPROM,...).

   La mémoire du système interne contient le logiciel du système, comme le set d'instructions, qui sont nécessaire pour le fonctionnement du système.

   La mémoire du programme contient le programme de conduite spécifique qui a été écrit par l'utilisateur et dans lequel le logique de la machine ou le processus a été mis.Le CPU prend un régle du programme de la mémoire et sait ainsi comment traiter l'information de l'entrée afin de recevoir les commandes souhaitéess aux sorties.

La mémoire des données est utilisée pour le mémorisation des données temporaires.

3. La partie de communication :Le module de communication est le centre du réseau. Il transporte l'information d'un participant à l'autre. Ces communications sont bidirectionel.Une liaison entre deux participants est nommée communication point à point (point-to-point :PPI) et une communication entre plusieurs participants est nommée communication multipoints (multipoint :MPI).
4. La partie programmation :Le programme d'utilisateurs est écrit à l'aide un appareil de programmation.

La figure ci dessous vous laisse voir la partition de mémoire d'un PLC.

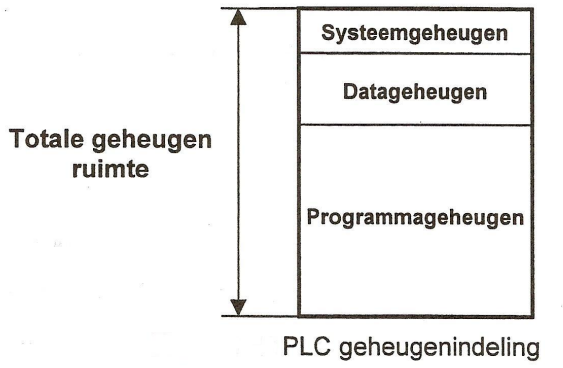


FIGURE 102.2 – source : cursus PLC, APHogeschool

102.1.2 Structure générale

La figure ci dessous vous donne un schema de la construction d'un PLC

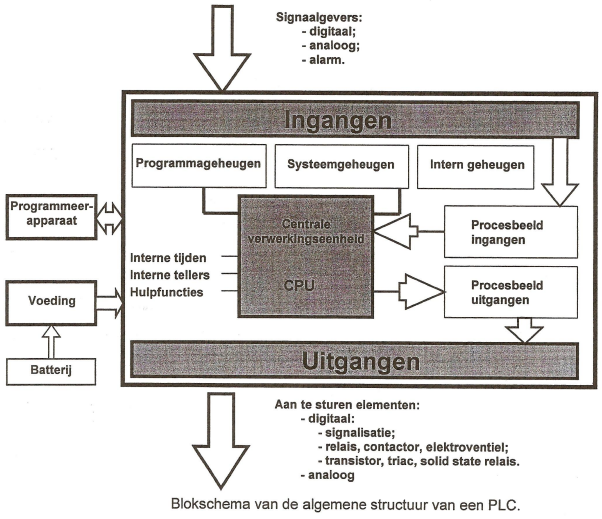


FIGURE 102.3 – source : cursus PLC, APHogeschool

Examinons la différence entre API modulaire et non modulaire API's.

- API non-modulaire :ils ont un nombre limité d'entrées et sorties et sont construit d'un ensemble.Le désavantage est que ce type ci est difficile à élargir.
- API modulaire :Chaque partie est un bloc séparé. Le type et le nombre de bloc peuvent être choisis.Chaque bloc est réalisé par un module spécifique.La API est donc réalisé par un nombre de blocs fonctionels qui sont construits autour un canal de communication : *un bus interne*.
  Ce bus est composé de plusieurs fils de signal parallèle.
  - Bus d'adresse :Ce bus va communiquer avec toutes les différentes adresses des différentes cartes.
  - Bus de données :Ce bus va transmettre toutes les données des cartes d'entrée ou de sortie.
  - Bus de commande :Ce bus va transmettre tous les signaux de commande ou d'alarme.

102.2 Fonctionnement

102.2.1 Général

La figure ci dessous donne une image plus détaillé d'un PLC.Cela permet de comprendre le fonctionnement du PLC.

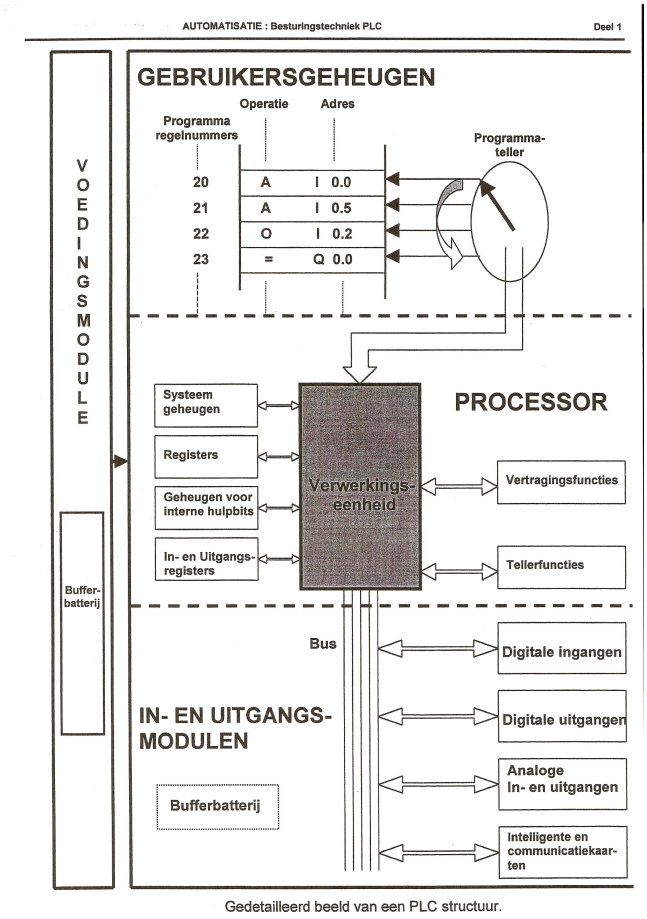


FIGURE 102.4 – source : cursus PLC, APHogeschool

Le programme utilisateur contient un nombre de régles qui sont mises dans la mémoire utilisateur par l'appareil de programmation.

Chaque régle contient une instruction à diviser en :
- Un *code d'operation* qui dit quel traitement il faut faire.
- L' *operand* ou *adresse* qui dit quelle données il faut utiliser pour ce traitement.

Le processeur contient les fonctions suivantes
- L' *unité de commande* qui régle le traitement de l'information basé sur les instructions qui sont écrites dans le programme d'utilisateur
- Les *accumulateurs* ou *registres* dans lequel les résultats intermédiares des traitements sont écrits.Quand l'execution suivante est faite, le contenu est transcrit par le nouveau résultat.
- La *mémoire* pour les bits d'assistance internes (marqueurs) qui sont utilisés par l'unité de commande pour mémoriser les résultats de traitement.
- Le *registre d'entrée et de sortie* qui mémorise la situation des entrés et des sorties pendant un cycle de programme.Il est en même temps une mémoire.
- Les *fonctions de temps et compteurs* ou *counters* sont des fonctions de logiciel qui sont fournis par le producteur et qui ont pour but de disposer de moyens qui ont la même fonction comme les fonctions de temps et des compteurs de la technologie hardware.

Le producteur fourni deux types de mémoires

1. Remanent :la situation logique d'avant l'interruption de tension est la même qu' *après* l'interruption.
2. Non remanent :la situation logique est mis à zèro après l'interruption de tension.

102.2.2 Fonctionnement du système

Après la mise sous tension du réseau, l'unité de commande donne un pulse qui remet à zèro les fonctions de temps non remanent, compteurs et marqeurs,les accu's et les registres d' entré et sortie.Nous supposons un progrès cyclique.Maintenant l'unité de commande lit les situations des signaux qui se trouvent dans le registre d'entrée.Le programme est parcouru, règle par règle,et traité en fonction des instructions. Sur base de ce programme et de la situation échantilloné des entrées, une sortie devient haute et l'unité de commande va fixer cette situation dans le registre de sortie. A la fin du cycle, l'unité de commande transfère le contenu complet par les modules de sortie.
Cette procedure est interrompue seulement par une erreur ou à l'aide de la fonction stop du API.

102.2.3 Temps de cycle

Le temps de cycle est la durée dont le processeur a besoin pour traiter completement le programme d'utilisateur. Ce durée contient aussi l'échantillonage des entrées et des sorties.

Trois éléments sont importants :

1. Tâches de gestion :Le logiciel du système doit soigner l'organisation des différentes unités et la communication.
2. Temps de traitemant du programme :Le temps nécessaire pour traiter le logique du programme. Il faut remarquer qu'il y a une différence entre le temps de traitement d'instructions d'un bit et d'un mot.
3. Commander les entrées et les sorties :La durée nécessaire pour lire les entrées et actualiser les sorties.

La figure ci dessous vous donne une représentation de la durée d'un cycle. Une valeur normale pour cette durée est de l'ordre de la microsecondes.

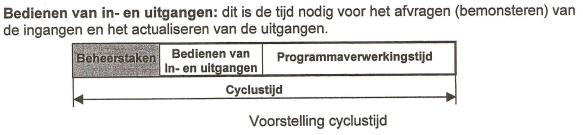


FIGURE 102.5 – source : cursus PLC, APHogeschool

102.3 Traitement du programme

102.3.1 Traitement du programme synchrone

Un cycle de programme contient trois phases :

1. Tous les signaux d'entrée sont écrit dans la mémoire d'entrée.
2. Le programme est traité avec l'information du régistre d'entrée
3. Pendant le traitement du programme d'utilisateur les signaux de commande pour les sorties sont écrit dans un régistre de sortie.

Seulement à la fin du cycle le contenu du régistre de la sortie est envoié au modules de sortie. Le programme est parcouru de façon cyclique et donc il faut en tenir compte pour l'ordre des instructions.

102.3.2 Traitement de programme asynchrone

Dans ce cas il n'y a pas un rapport entre la lecture des entrées, actualiser les sorties et le traitement du programme. **Pour programmer d'une façon fiable et optimale il faut avoir une bonne connaissance de la manière dont un programme est traité et du système employé.**

102.3.3 Temps de réaction

Le temps de réaction d'un API peut être défini comme la durée entre le changement de l'entrée, qui est mésuré par un capteur à l'entrée, et le changement du signal correspondant de la sortie sur un actuator.

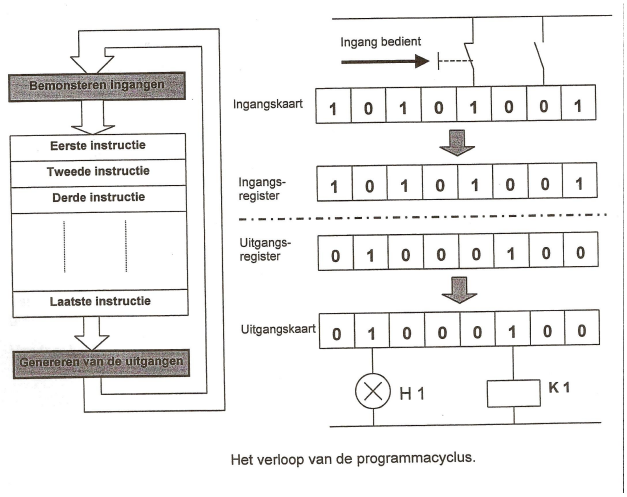


FIGURE 102.6 – source : cursus PLC, APHogeschool

Temps de réaction pratique

En pratique le temps de réaction est l'addition de

1. Temps propre du capteur
2. Retard du module d'entrée
3. Temps du système du API
4. Temps de cycle du API
5. Retard du module de la sortie
6. Temps propre de l'actuator

Un PLC a un système de surveillance qui surveille si le temps de cycle ne dépasse pas une certaine valeur.Si cette valeur est dépassée toutes les sorties sont mises à zèro.

102.4 Composants

Nous pouvons retrouver les composants suivants
– Module d'alimentation
– Unité centrale
– Module digitale d'importation et d'exportation

102.4.1 Module d'alimentation

La fonction la plus importante est de transformer la tension nécessaire pour le fonctionnement des composants du PLC.
En général il y a deux circuits

– Le circuit d'alimentation du processeur et les modules d'expansion
– Le circuit pour la charge externe

102.4.2 CPU

Le CPU est le coeur du PLC. Il contient les mémoires et l'unité centrale de traitement. Afin de programmer le CPU on utilise plusieurs connections pour programmer l'appareil ou même d'autres CPU ou des bus champs.
La mémoire est construite de plusieurs lieux de mémoire qui peuvent tous mémoriser une situation 0 ou 1.Les composés de mémoires les plus appliqués sont RAM,ROM,PROM,EPROM,EEPROM,FEPROM.

102.4.3 Modules de signaux

Les modules de signaux sont l'interface entre le processus à commander et le PLC.Il y a des modules digitaux que analogue.
Les deux circuits peuvent être construits de différentes manières en fonction de la connection de potentiel entre les deux circuits.
– Modules connectés en potentiel :il y a une connection à la terre centrale et une masse centrale.
– Modules sans potentiel : il n'y a pas de connection de potentiel entre les différents modules.

102.5 Techniques de programmation

Il existe trois méthodes d'écriture des solutions, c'est à dire
– Expression de Boole
– Diagramme échelle
– Schemas logiques
Si les problèmes sont plus difficiles on emploie le GRAFCET. Dans la norme IEC, il existe cinq methodes de programmation
– Liste d'instruction(STL of AWL)
– Diagramme d'échelle(LD of KOP)
– Diagramme de bloques de fonctions(FDB of FUP)
– Texte structurée(ST of SCL)
– Diagramme de fonction sequentiel(SFC)
Il faut tenir compte qu'il existe une différence entre des bits, bytes, mots et double mots à utiliser dans le programme.
– bit :l'unité la plus petite dans le système binaire, 0 of 1.
– byte :8 bits
– mot :16 bits ou deux bytes
– double mot :32 bits
La figure ci dessous reflète ces notions

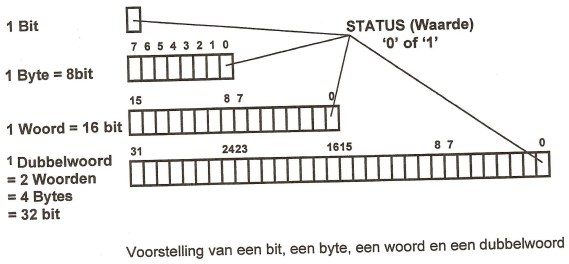


FIGURE 102.7 – source : cursus PLC, APHogeschool

102.5.1 Instructions

Dans un programme les entrées sont lues et les sorties sont commandées par moyen d'instructions.Une instruction renvoi aux entrées et sorties.Donc chaque entrée et sortie doit être nommée. Ceci est fait par allouer un paramétre aux entrées et sorties. Faites attention parce qu'il faut distinguer les adresses d'un bit,byte,mot ou double mot.

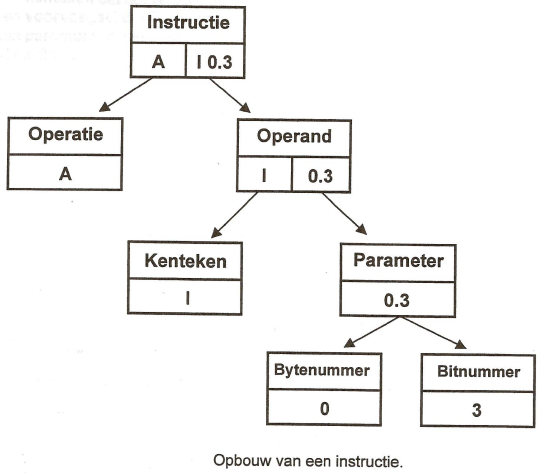


FIGURE 102.8 – source : cursus PLC, APHogeschool

Un **marqueur** est un relais d'assistance interne qui peut servir comme résultat intermédiaire. Le marqueur prend une valeur constante pendant un cycle complet et est disponible pour les connections suivantes. Un marqueur peut servir dans un programme pour ouvrir ou fermer.
L'adresse existe de
– Une caractèristique qui indique la différence entre l'entré et la sortie.
– Un préfixe qui indique la largeur de la caractéristique.

– Un parametère qui indique une combination de chiffres pour le bit d'entrée ou de sortie et qui est séparé par un point.

Soort	Type	Opmerking
I	Ingang	Procesbeeld van de digitale ingangen van de PLC
Q	Uitgang	Procesbeeld van de digitale uitgangen van de PLC
M	Geheugen	Geheugengebied dat door de PLC kan gebruikt worden voor tussen resultaten

Kenteken aanduiding	Grootte (bit)	Opmerking
X	1	Bit
(geen)	1	Bit
B	8	Byte
W	16	Woord
D	32	Dubbelwoord
L	64	Lang woord

Overzicht van de gebruikte kentekens en voorvoegsels

FIGURE 102.9 – source : cursus PLC, APHogeschool

Exemples
– I0.0 :adresse du bit d'entrée
– Q4.8 :adresse du bit de sortie
Comme ça on a aussi
– Adresser un byte :IB0(byte d'entrée) of QB6(byte de sortie)
– Adresser un mot :IW0(mot d'entrée) of QW5(mot de sortie)
– Double mot :IDW0 of QDW6
Dans un PLC, on peut aussi adresser des bits d'assistance interne. Ils sont indiqués par M.Exemple M0.4
En anglais cela devient F de flag.Un bit d'assistance n'est qu'un nombre de bits dans la mémoire RAM.
Comme ça on a aussi des bytes d'assistance internes et (double)mots qui sont indiqués respectivement MBx et MWy(MDWy) avec x et y respectivement chiffre de byte et de mot. Exemples MB30,MW32.

102.6 Systèmes de communication (bus)

Le but d'un bus d'ordinateur est de tranferer des données entre l'ordinateur et l'appareillage périphérique.
L'appareillage périphérique direct sont les cartes que l'on peut tamponner dans un rack d'ordinateur.
Des cartes possibles

1. Extension de mémoire
2. Carte video
3. Carte disque ou tape
4. Carte de réseau
5. Carte paralléle ou en série
6. Carte de mesure digitale ou analogique

Comment des données peuvent elles être traitées ? Il y a quatre possibilités

1. Input/output programmé :Le programme va lire le régistre d'état de la carte I/O et si nécessaire exécuter les traitements I/O.
2. Interrupts :Une ou plusieurs lignes d'interrupt sont prévues sur un bus.Un I/O activera, si nécessaire, ces lignes. Le processeur interrompera le programme qui est occupé, gardera le nécessaire et aprés l'interrupt continuera le programme.
3. DMA :Une carte I/O contient un controlleur DMA qui sait ou il faut sauvegarder ou chercher les données.Dans ce cas le processeur n'est pas chargé.
4. I/Oprocesseur :Ce processeur exécute les tâches I/O.

Il y a deux types de bus,en série ou en le parallèle.
– Parallèle :Chaque bit a sa ligne physique.On a besoin de beaucoup de lignes pour exécuter la communication mais la vitesse de transfer est élevée.
– En série :Maintenant on n'a pas besion de beaucoup de cable mais la vitesse de transfer est basse.

102.6.1 Bus d'ordinateur

Dans ce cours, on ne donne qu'un petit échantillon de ce qui existe comme types de bus fréquemment utilisé.
– RS-232-C
– RS485/422
– Inter C bus
– IEEE-488bus

102.6.2 Bus de champ

Dans ce paragraphe nous parlerons des bus de champs, c'est à dire la communication entre des ordinateurs (PLC) et des capteurs/transmetteurs.

- ASIbus :Actuator Sensor Interface.C' est un cable à deux âmes avec des connections spécifique.Ce cable envoie des message de commande et de données.C'est la puce interface ASI qui contient la force de ce système.Dans cette puce, les données parallèle des capteurs sont transformées dans des données en série, qui arrivent au gerant ASI (Ceci est en fait le PLC).
- Bitbus :Le transfert est très rapide avec une connection RS-485 avec protocol SDLC.Ce bus est bien approprié pour la communication dans des applications industrielles.
- CANbus :Controller Area Network,est appliqué dans l'industrie automobil.Le protocol est de type CSMA/CD-BA.
- FIPbus :Factory Instrumentation Protocol ; Pour l'automatisation industrielle,on utilise la fibre de verre ou cable twisté.Le protocol est basé sur le fait que les applications ont un comportement cyclique.
- Inter C bus :Le principe est basé sur la communication synchrone et pour cela on a besoin de deux signaux : signal de données(SDA) et clock(CLK).Ce système est facile à utiliser pour de grandes distances.
- Interbus-S :La communication des données est hierarchisée.
- Profibus :Process Field Bus modélisé OSI.

102.7 Grafcet

102.7.1 Introduction

Parce que l'évolution technique a rendu l'automatisation des processus plus difficile, on a dû trouver une manière de déterminer les specifications techniques et fonctionelles clairement et sans ambiguité. Cette manire de travail a t dvloppe pendant les annes 70 et 80 en Allemagne et en France et est appele GRAFCET. Cette acronyme signifie Graphe Fonctionnel de Commande Etape/Transition. En Allemagne cette méthode est appelée Functionsplan et en Belgique et aux Pays-Bas functiediagram. La norme est le **NEN-EN-IEC 60848**.

102.7.2 Principes généralles

Un systme automatisé existe en deux parties qui dépendent les unes des autres.

1. Système control

   La principe de programmation est comme écrire un flowchart de n'importe quel autre programme. Les étapes sont

   (a) Analyse du problème et donc partager du processus en {etappes

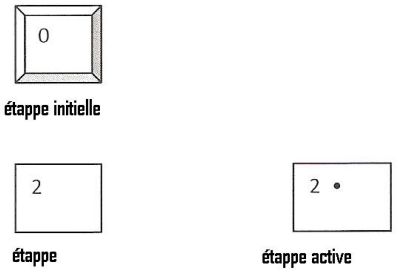
(b) A chaque étape on couple les actions nécessaires

(c) Une étape devient active si l'étappe précédente stap est active et les conditions de transition sont repectées.

(d) La première étape est l'initialisation

102.7.3 Symboles

a) L'étappe



b)La connexion ciblée

L'evolution du processus est marqué par des connexions ciblées verticales.

c)La transition

La transition est la condition pour passer d'une étape á l'étape suivante. La transition ne peut qu'avoir lieu si les étapes précédentes sont actives. On travaille donc avec des structures Boolean.

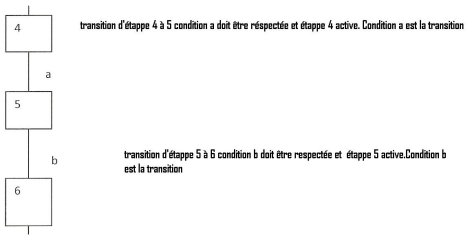
Donc deux étapes sont sépar'ees par une transition et deux transitions sont séparées par une étape.

Remarque :Suppose qu'il faut allumer une lampe avec un bouton poussoir.L'extinction de cette lampe se produit avec ce même bouton. Ceci se produit tellement vite que la lampe s'allume avant qu'on a lâché le bouton. Donc on arrive de l'étape 1 'lampe éteint' á l'étape 2 'lampe allumée' mais parce que le bouton n'est pas encore lâché on arrivera tout de suite en étape 1 de nouveau 'lampe éteint'. Ceci est impossible parce qu'on arrive dans une situation indécis. Il faut donc utiliser un controle de flanc. Flanc montant lampe allumée, flanc descend lampe éteint

102.7.4 Succesion

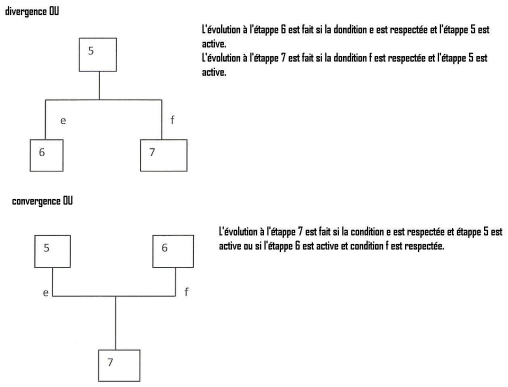
Succesion unique

Appelé aussi succesion lineaire.

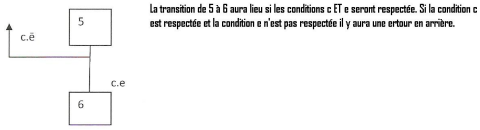


Succesion de choix

Ici on applique la condition OU

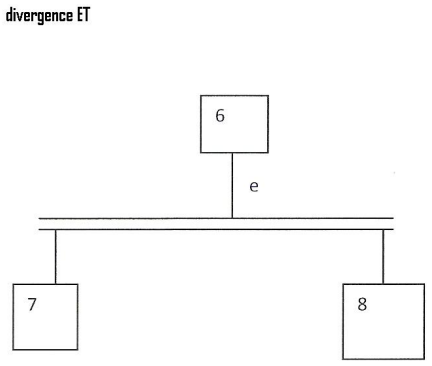


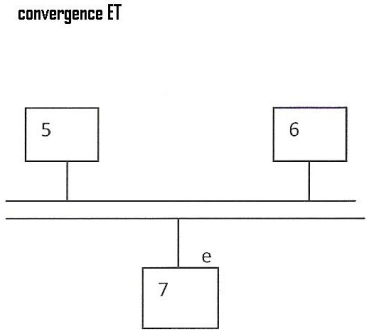
On peut programmer dans ce cas ci aussi un saut conditionel 'retour'.



Succesion simultané

On utilise ici la condition ET





Commande's

Il existe quatre types de commande's

(a) S :commande est memorisée(S :Stored)

(b) D :commande est retardé(D :Delay)

(c) L :commande est limité dans le temps(L :Limited)

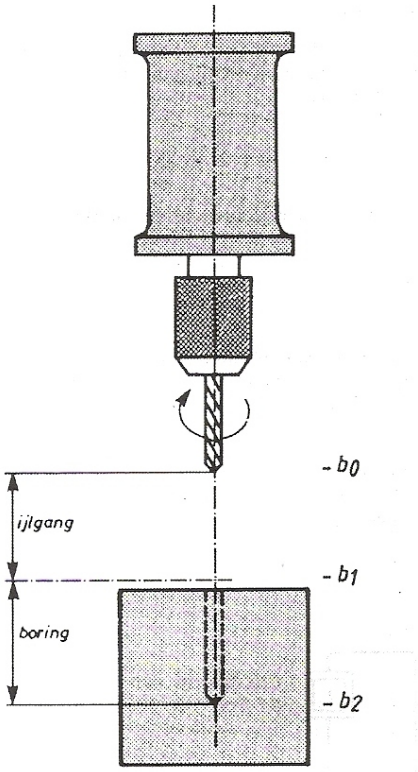
(d) C :commande est conditionel(C :Conditional)

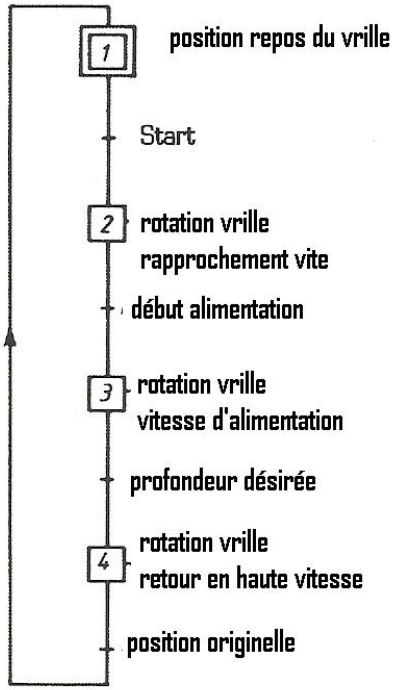
Si on programme les commande's il faut faire attention de la succesion dans laquelle on les programme. Suppose que la commande est programmé de manière suivante

- CLS :l'étape est memorisée et limitée dans le temps et n'devient active avant qu'une condition est respectée
- SCL :l'étape sera memorisée mais la limite dans le temps et activation dependent d'une condition.

Les deux programmes semblent le même en point de vue programmatoire mais se sont deux actions totallement différentes.

102.8 Exemple




1
position repos du vrille
Start
2
rotation vrille
rapprochement vite
début alimentation
3
rotation vrille
vitesse d'alimentation
profondeur désirée
4
rotation vrille
retour en haute vitesse
position originelle