

PROGRAMMATION CONTROLEUR PLC – LANGAGE SFC

1. Généralité sur le langage SFC

Le **Sequential Function Chart (SFC)** est un langage graphique de programmation des **Automates Programmables Industriels** défini dans la norme CEI 61131-3.

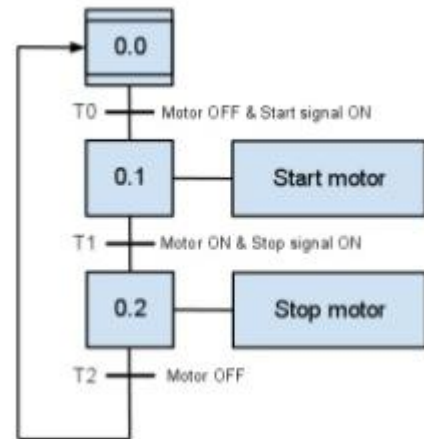
C'est un outil de spécification de la partie séquentielle d'un système automatisé depuis le cahier des charges jusqu'à son exploitation, de haut niveau.

Cet outil de modélisation est maintenant utilisé de manière internationale par toutes les entreprises nécessitant une production automatisée

Le SFC est un dérivé international du GRAFCET (GRAphe Fonctionnel de Commande Étape / Transition) qui est utilisé pour décrire graphiquement une ou des séquences.

Le SFC fait partie des spécifications d'un système séquentiel comme :

- ✓ les chronogrammes,
- ✓ les algorithmes et les organigrammes
- ✓ ou encore les graphes d'état,



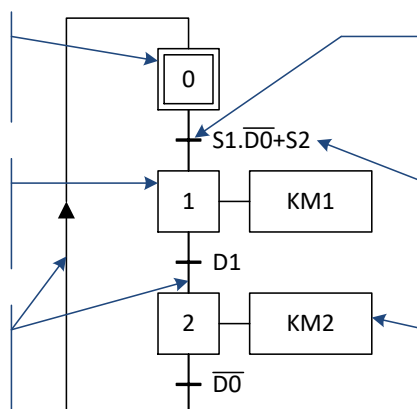
2. La programmation en SFC

2.1. Représentation générale

Étape initiale : correspond, en général, à un état de repos du système

Étape : correspond à un état stable du système.

Liaisons orientées : indiquent le sens d'évolution du SFC.



Transition : indique la possibilité d'évolution d'une étape à l'étape suivante.

Réceptivité : condition logique sur les informations reçues par le système.

Action : action à effectuer lorsque l'étape est active.

2.2. Points de vue

2.2.1. Point de vue « partie opérative » (PO)

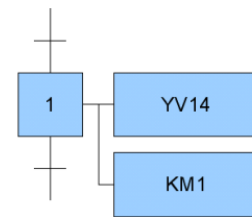
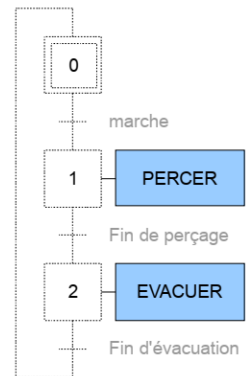
La description du comportement du système est faite par un observateur se situant d'un point de vue interne au système automatisé, sans tenir compte de la partie commande (PC). Le point de vue partie opérative décrit les actions produites par les **effecteurs**.

- ✎ La notation est littérale avec des verbes à l'infinifitif pour les actions.

2.2.2. Point de vue « partie commande » (PC)

La description du comportement du système est faite par un observateur se situant d'un point de vue interne à la partie commande. Ce SFC prend en compte les choix technologiques et l'ensemble des échanges PC ↔ PO (**capteurs et pré-actionneurs**) et PC ↔ Opérateur (**dialogue**).

- ✎ La notation est symbolique (repères du dossier technique).



2.3. Règles

2.3.1. Règle de syntaxe

L'alternance étape-transition et transition-étape doit toujours être respectée quelle que soit la séquence parcourue. Deux étapes ou deux transitions ne doivent jamais être reliées par une liaison orientée. La liaison orientée relie obligatoirement une étape à une transition ou une transition à une étape.

2.3.2. Règles d'évolution

Règle 1 : Situation initiale

Cette règle vise à vérifier que le système est prêt à commencer un cycle. Elle impose donc à la première transition (après l'étape initiale) de vérifier l'état du système (capteurs) et les ordres reçus (dialogue).

Règle 2 : Franchissement d'une transition

Une transition devient franchissable et est alors obligatoirement franchi lorsque la transition est validée (lorsque toutes les étapes immédiatement précédentes reliées à cette transition sont actives) et lorsque la réceptivité associée à cette transition est vraie.

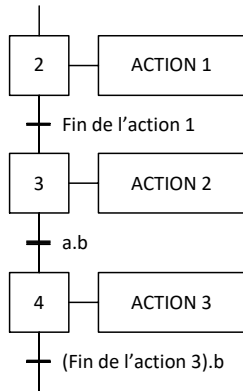
Règle 3 : Evolution des étapes actives

Le franchissement d'une transition entraîne l'activation de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes.

2.4. Structures de bases

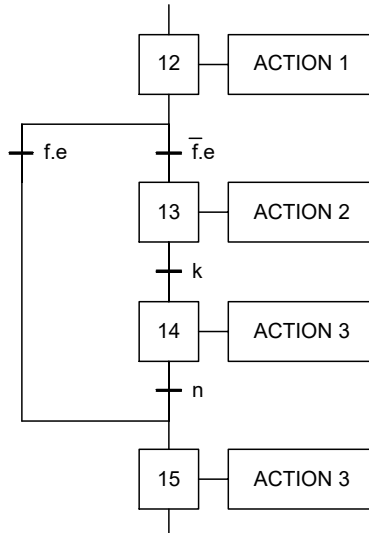
Séquence linéaire :

Une séquence linéaire est composée d'une suite d'étapes s'activant les unes après les autres.



Saut d'étapes :

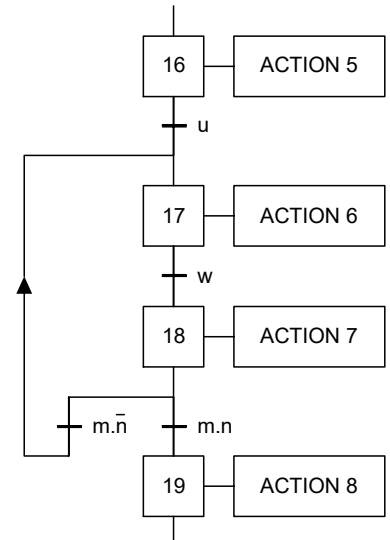
Le saut d'étapes permet de sauter une ou plusieurs étapes.



Ici, les étapes 13 et 14 seront sautées si la réceptivité « f.e » est vraie.

Reprise de séquence :

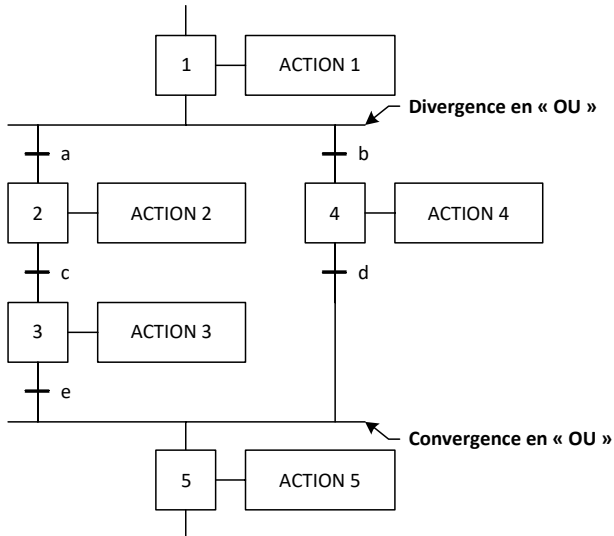
La reprise de séquence permet de recommencer plusieurs fois la même séquence.



Ici, les étapes 17 et 18 se répèteront si la réceptivité « m.n » est vraie.

Sélection de séquences :

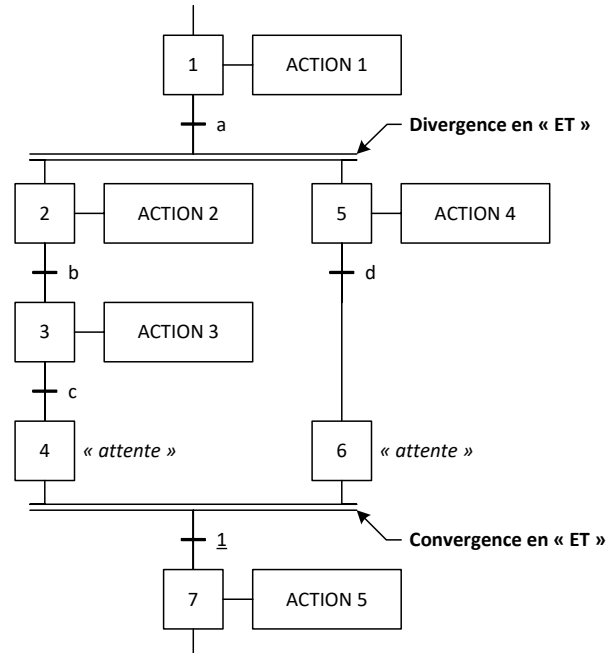
Une sélection de séquence exprime un choix d'évolution entre plusieurs étapes ou séquences.



Elle se représente par autant de transitions validables en même temps qu'il y a d'évolutions possibles.

Séquences simultanées :

Le franchissement d'une transition conduit à activer plusieurs séquences simultanément.

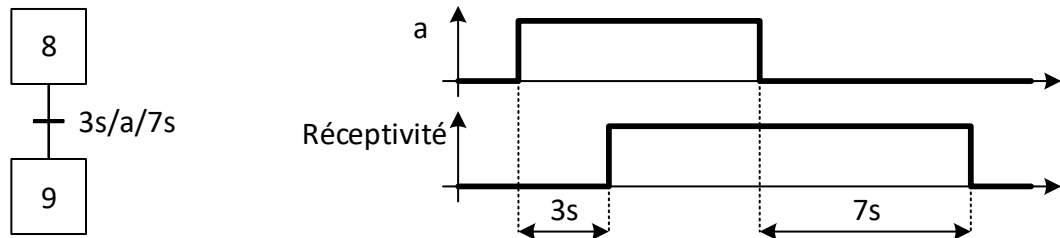


Pour assurer la synchronisation de la désactivation simultanée de plusieurs séquences, des étapes d'attente sont souvent prévues.

2.5. Réceptivité dépendante du temps (temporisation)

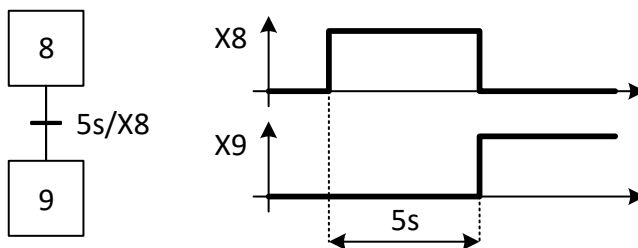
La notation est de la forme « $t1/variable/t2$ ».

Dans l'exemple suivant, la réceptivité n'est vraie que 3 s après que « a » passe de l'état 0 à l'état 1, elle ne redevient fausse que 7 s après que « a » passe de l'état 1 à l'état 0.



Simplification usuelle :

L'utilisation la plus courante est la temporisation de la variable d'étape avec un temps $t2$ égal à zéro :



Dans ce cas la durée d'activité de l'étape 8 est de 5 s.

Remarque : Il est possible d'utiliser cette notation lorsque l'étape temporisée n'est pas l'étape amont de la transition.

3. Raccordement d'un Automate Programmable Industriel

Conformément à la structure de la chaîne d'information et à celle du SFC, sont raccordés à l'API :

Sur les entrées : Tout élément transmettant :

- soit des ordres en provenance de l'opérateur (boutons poussoirs, commutateurs, coups de poings d'arrêt d'urgence...),
- soit des informations en provenance du système lui-même (capteurs, relais thermiques...).

Ces éléments se retrouvent au niveau des **réceptivités** du SFC (associées aux **transitions**).

Sur les sorties : Tout élément transmettant :

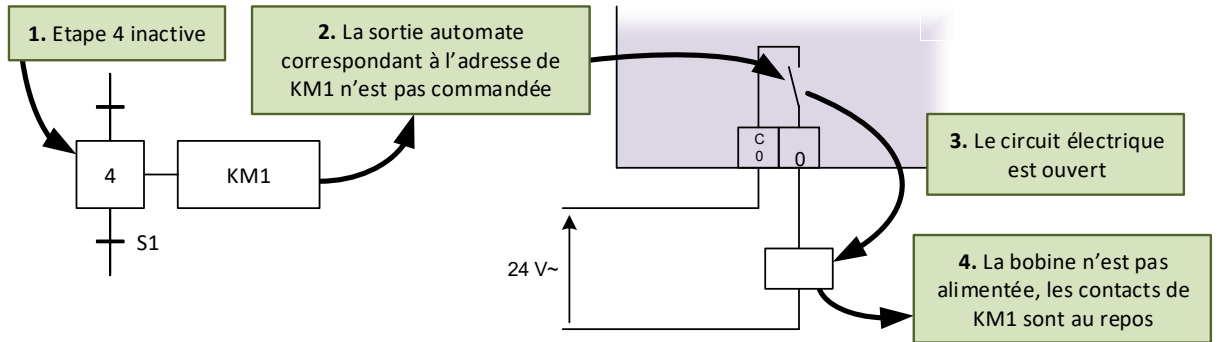
- soit des informations en direction de l'opérateur (voyants, afficheurs...),
- soit des ordres en direction du système lui-même (contacteurs, distributeurs...).

Ces éléments se retrouvent au niveau des **actions** du SFC (associées aux **étapes**).

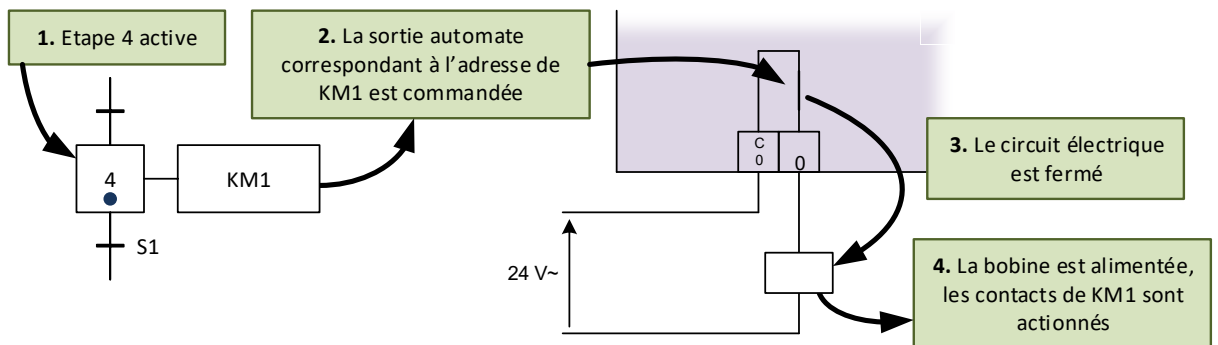
3.1. Raccordement des sorties d'un API

Soit un contacteur KM1, raccordé à la sortie n°0 d'un API :

1^{ère} cas :



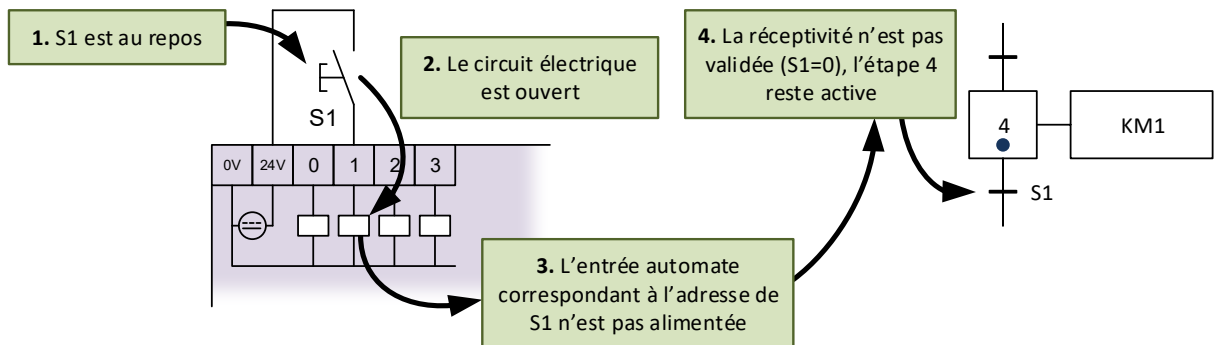
2^{ème} cas :



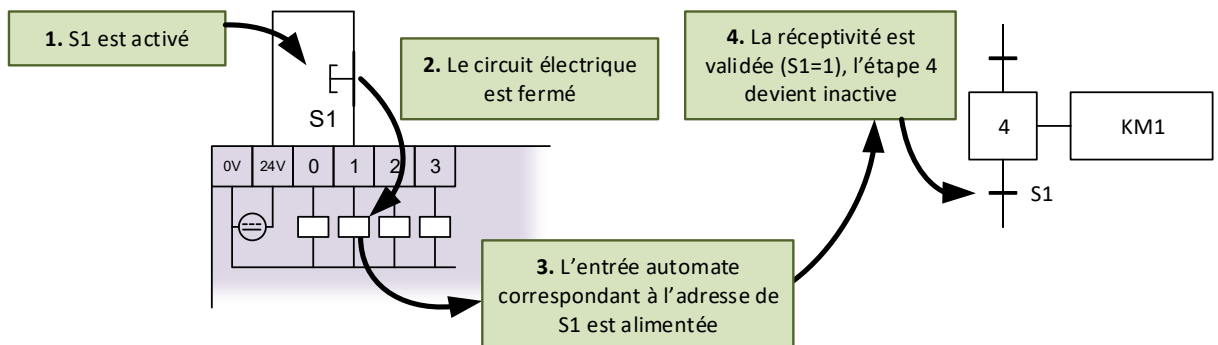
3.2. Raccordement des entrées d'un API

Soit un bouton poussoir S1, raccordé à l'entrée n°1 d'un API :

1^{ère} cas :



2^{ème} cas :



4. Exercice 1 : LANGAGE FBD / LANGAGE SFC

Soit l'exemple d'un bouton-poussoir S1 raccordé à l'entrée n°1 d'un module Zelio d'un voyant H1 raccordé à la sortie n°1. On souhaite que le voyant soit allumé quand le bouton est activé. Autrement dit, la sortie Q1 doit être activée quand l'entrée I1 est alimentée :

1. Programme en FBD :

- S1 est au repos --> I1 n'est pas alimentée --> Q1 est au repos --> H1 est éteint :

- S1 est actionné --> I1 est alimentée --> Q1 est activé --> le voyant est allumé :

2. Programme en SFC :

- S1 est au repos --> I1 n'est pas alimentée --> Etape initiale active --> Q1 est au repos --> H1 est éteint :

- S1 est actionné --> I1 est alimentée --> Etape suivante active --> Q1 est activé --> le voyant est allumé :

5. Exercice 2 : DIAGRAMME D'ETATS-TRANSITIONS / LANGAGE SFC

1. Diagramme d'états-transitions

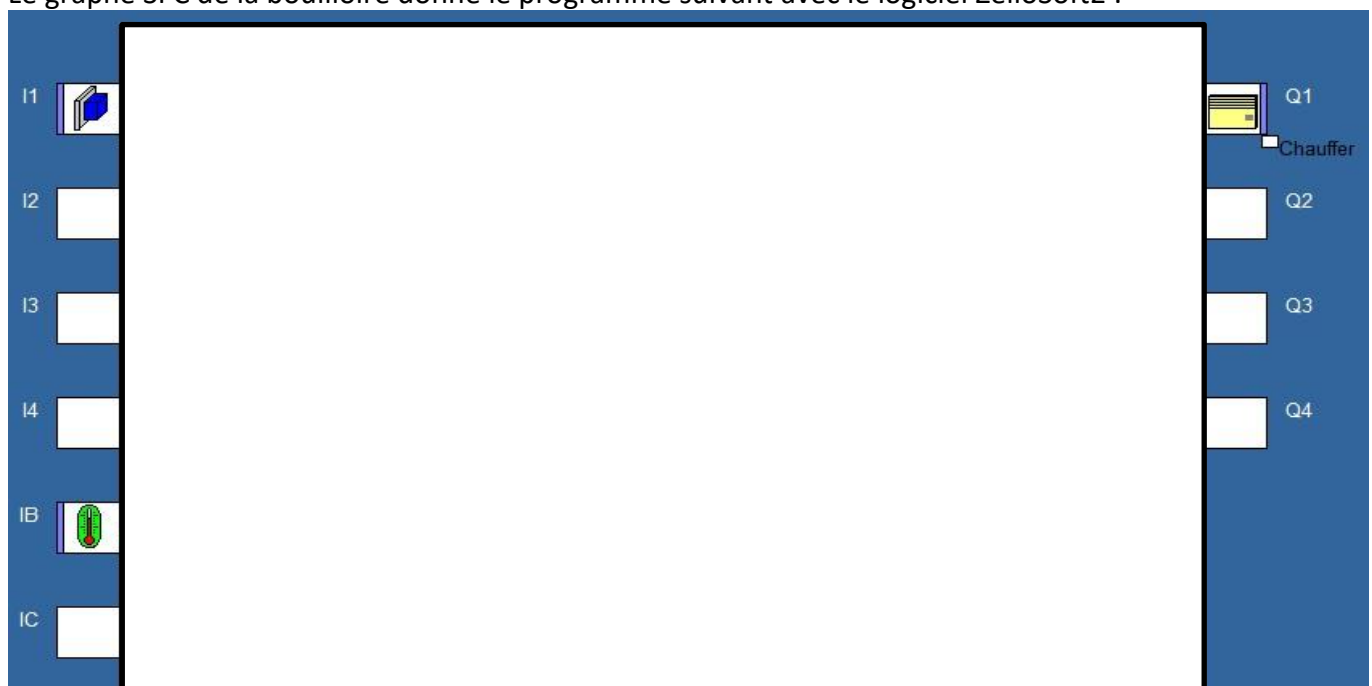
Exe

Lors d'un appui sur un bouton poussoir BP une bouilloire doit se mettre à chauffer l'eau jusqu'à ce que la température de l'eau atteigne 95°C :

2. Langage SFC

Lors d'un appui sur un bouton poussoir BP une bouilloire doit se mettre à chauffer l'eau jusqu'à ce que la température de l'eau atteigne 95°C :

Le graphe SFC de la bouilloire donne le programme suivant avec le logiciel ZelioSoft2 :



6. EXERCICES

EXERCICE 1 — Éclairage automatique

Contexte : On souhaite automatiser l'éclairage d'un couloir à l'aide :

- d'un détecteur de présence P,
- d'un bouton d'arrêt d'urgence AU,
- d'une lampe commandée par l'automate.

Fonctionnement attendu :

1. Lorsque P détecte une présence, la lampe doit s'allumer.
2. Lorsque la présence disparaît, la lampe reste allumée 10 secondes.
3. Si AU est activé, l'éclairage doit s'éteindre immédiatement.

Travail demandé : Réaliser le SFC complet

EXERCICE 2 — Convoyeur simple

Contexte : Un mini-convoyeur transporte une pièce jusqu'à un capteur de fin de course FC.

Fonctionnement attendu :

1. L'opérateur appuie sur Start → le convoyeur démarre.
2. Le convoyeur avance jusqu'à ce que FC = 1.
3. Le convoyeur s'arrête automatiquement.
4. L'opérateur appuie sur Reset pour revenir à l'état initial.

Travail demandé : Réaliser le SFC complet

EXERCICE 3 — Mélangeur automatique

Contexte : Un mélangeur doit effectuer automatiquement un cycle de production.

Fonctionnement attendu :

1. Remplissage pendant 5 s.
2. Mélange pendant 8 s.
3. Vidange jusqu'à ce que le capteur de cuve vide (CV) = 1.
4. Retour à l'état initial.

Travail demandé : Réaliser le SFC complet