



# Analyste Programmeur en Automatisation, Robotique et Informatique Industrielle TS ARII

## **Module MF 3.1**

***MF 3.1 Régler et mettre au point un S.A.P.***

***Programmer avec Ecostruxure***

**Patrick MONASSIER – année 2019-2020**

## **Module MF 3.1**

### ***Régler et mettre au point un Système Automatisé de Production (SAP)***

#### Compétences

Régler et mettre au point les séquences.

#### Objectifs

Exploiter un atelier logiciel d'un point de vue diagnostic

#### Contenu

- Outils de diagnostic du logiciel et du matériel
- Références croisées
- Piles de défauts
- Interprétation d'un listing
- Mode de visualisation
- Diagnostic matériel
- Exemples liés à la robotique

# Formation sur EcoStruxure

Présentation générale et organisation des programmes

*Description de la documentation et utilisation*



**EcoStruxure Machine Expert - Basic** prend en charge les langages de programmation CEI-61131- 3 suivants :

- Schéma à contacts
- Liste d'instructions
- Grafcet (liste)
- Grafcet (SFC)

**EcoStruxure Machine Expert - Basic** peut fonctionner dans les modes suivants :

- Mode hors ligne
- Mode en ligne
- Mode simulateur



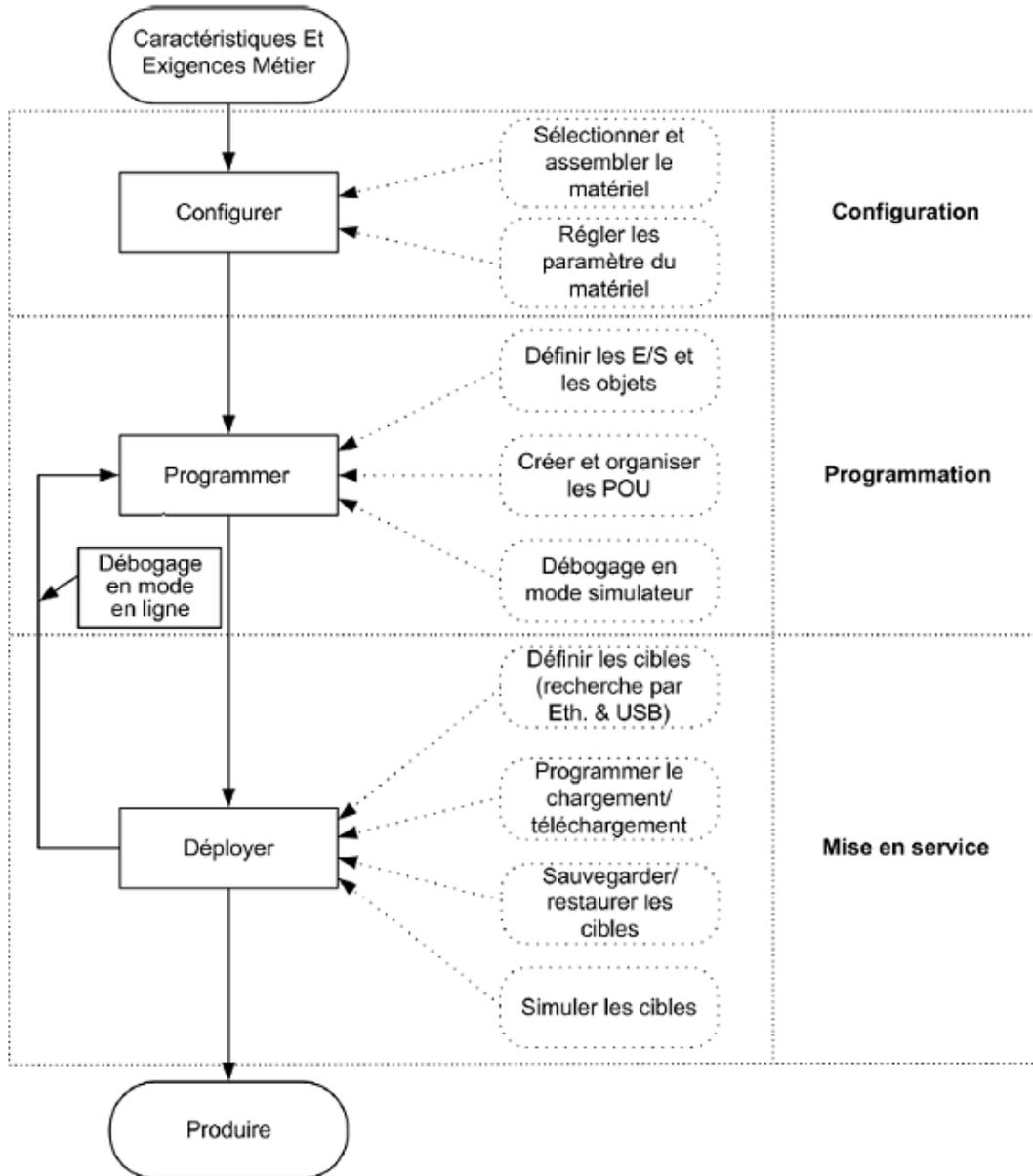
## Présentation

EcoStruxure Machine Expert - Basic est un outil de programmation graphique conçu pour faciliter la configuration, le développement et la mise en service de programmes pour des Logic Controller.

### Termes clés :

EcoStruxure Machine Expert - Basic utilise la terminologie suivante :

- **Projet** : un projet EcoStruxure Machine Expert - Basic contient des détails sur le développeur et le but du projet, la configuration du Logic Controller et des modules d'extension associés ciblés par le projet, le code source d'un programme, les symboles, les commentaires, la documentation et d'autres informations.
- **Application** : contient toutes les parties du projet qui sont chargées dans le contrôleur logique, notamment le programme compilé, les informations de configuration matérielle et les données étrangères au programme (propriétés du projet, symboles et commentaires).
- **Programme** : code source compilé exécuté sur le Logic Controller.
- **POU** (*program organization unit, unité organisationnelle de programme*) : objet réutilisable contenant une déclaration de variables et un ensemble d'instructions utilisées dans un programme.



- **Propriétés** : Configurez les propriétés du projet.
- **Configuration** : Définissez la configuration matérielle du Logic Controller et des modules d'extension associés.
- **Programmation** : Développez votre programme dans l'un des langages de programmation pris en charge - Afficheur : Créez une interface opérateur pour le module Afficheur graphique déporté TMH2GDB.
- **Mise en service** : Gérez la connexion entre EcoStruxure Machine Expert - Basic et le contrôleur logique - chargez/téléchargez des applications, testez et mettez en service l'application.

Icône	Description
	Démarrer le Logic Controller (CTRL+M). Disponible uniquement en mode en ligne et lorsque le contrôleur n'est pas déjà à l'état <code>RUNNING</code> .
	Arrêter le Logic Controller (CTRL+L). Disponible uniquement en mode en ligne et lorsque le contrôleur est à l'état <code>RUNNING</code> .
	Initialiser le contrôleur logique (uniquement en mode en ligne).
	Compiler le programme.
	Connexion au contrôleur sélectionné (CTRL+G) ou déconnexion du contrôleur sélectionné (CTRL+H). <b>NOTE</b> : Le nom du contrôleur sélectionné s'affiche à gauche de ce bouton.
	Lancer (CTRL+B) ou arrêter (CTRL+W) le simulateur EcoStruxure Machine Expert - Basic ( <i>voir page 278</i> ).



- 1 **Etat du programme :**  
Indique si des erreurs ont été détectées pour le programme.
- 2 **Etat de la connexion :**  
Indique l'état de la connexion entre EcoStruxure Machine Expert - Basic et soit le contrôleur logique soit le contrôleur logique simulé.
- 3 **Etat du contrôleur :**  
Indique l'état actuel du contrôleur logique (RUNNING, STOPPED, HALTED, etc.).
- 4 **Temps de scrutation :**  
Indique la durée de la dernière scrutation.
- 5 **Dernière erreur du contrôleur :**  
Indique la dernière erreur détectée. Les informations sont extraites des bits système et des mots système si le contrôleur logique est à l'état STOPPED ou HALTED.

## Imprimer des rapports

### Présentation

Vous pouvez générer des rapports personnalisables en vue de leur impression ou de leur enregistrement au format PDF sur le PC.

Le bouton **Imprimer** dispose des options suivantes :

- **Imprimer** pour imprimer un rapport personnalisé pouvant fournir la liste des composants matériels, l'architecture de l'application ainsi que le contenu du projet, du programme et de l'application.
- **Imprimer la nomenclature** pour imprimer la liste des composants matériels utilisés dans la configuration du projet.
- **Paramètres** pour personnaliser le rapport de projet en sélectionnant les éléments à inclure et la mise en page.

# Chapitre 5

## Programmation

### Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
5.1	Présentation de l'espace de travail de programmation	68
5.2	Fonctions spéciales	69
5.3	Configuration du comportement et des tâches du programme	80
5.4	Gestion des POU	89
5.5	Fonctions définies par l'utilisateur	103
5.6	Blocs fonction définis par l'utilisateur	113
5.7	Tâche maître	122
5.8	Chaînes	126
5.9	Tâche périodique	133
5.10	Tâche d'événement	137
5.11	Utilisation des outils	144
5.12	Programmation en langage Schéma à contacts	176
5.13	Programmation en langage Liste d'instructions	201
5.14	Programmation en Grafcet (liste)	213
5.15	Programmation en Grafcet (SFC)	222
5.16	Débogage en mode en ligne	237

## Présentation

## Programmation

## Objets

Dans EcoStruxure Machine Expert - Basic, le terme *objet* représente une zone mémoire d'un Logic Controller réservée à l'usage d'une application. Exemples d'objets :

- Variables logicielles simples (bits et mots mémoire par exemple)
- Adresses d'entrées et de sorties numériques ou analogiques
- Variables internes du contrôleur (mots et bits système par exemple)
- Fonctions système ou blocs fonction prédéfinis (temporisateurs ou compteurs par exemple)

La mémoire du contrôleur est soit pré-allouée à certains types d'objets, soit allouée automatiquement lorsqu'une application est téléchargée dans le Logic Controller.

Les objets ne peuvent être adressés dans un programme qu'après allocation de mémoire.

L'adressage des objets utilise le préfixe `%`. Par exemple, `%MW12` est l'adresse d'un mot mémoire, `%Q0.3` est l'adresse d'une sortie numérique intégrée et `%TMO` est l'adresse d'un bloc fonction Timer.

## Introduction

## Allocation de mémoire

EcoStruxure Machine Expert - Basic vous permet de préallouer (réserver) des blocs de mémoire du Logic Controller à certains types d'objet utilisés dans un programme, notamment des objets simples (mots mémoire, mots constants) et des objets logiciels (blocs fonction).

**En mode Automatique ou en mode Manuel**

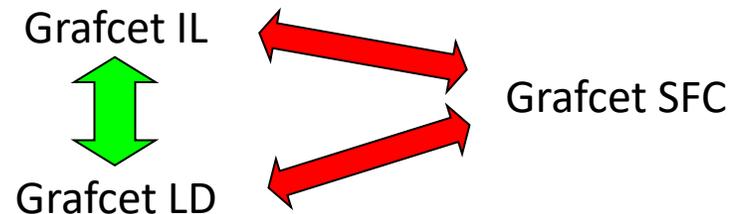
## Introduction

## Réversibilité Schéma à contacts/Liste

EcoStruxure Machine Expert - Basic prend en charge la conversion des réseaux Schéma à contacts (LD) en Liste d'instructions (IL) et inversement. On parle de *réversibilité de programme*.

Dans EcoStruxure Machine Expert - Basic, vous pouvez convertir des réseaux dans différents langages de programmation, à tout moment et autant de fois que vous le souhaitez. Par exemple, vous pouvez afficher un programme avec des réseaux en langage LD (Schéma à contacts) et d'autres réseaux en langage IL (Liste d'instructions).

**NOTE** : la conversion d'un programme LD ou IL en Grafcet (SFC), Grafcet (SFC) en LD ou IL, ou Grafcet (IL) en Grafcet (SFC) est impossible.



## Présentation Comportement de l'application

Vous pouvez configurer les aspects suivants du comportement de l'application avec l'automate logique :

- Démarrage (*voir page 81*)
- Chien de garde (*voir page 83*)
- Comportement de repli (*voir page 84*)
- Niveaux fonctionnels (*voir page 84*)

### Configuration du comportement de l'application

Pour configurer le comportement de l'application, procédez comme suit :

Étape	Action
1	Cliquez sur l'onglet <b>Tâches</b> dans la partie gauche de la fenêtre <b>Programmation</b> .
2	Sélectionnez l'élément <b>Comportement</b> . <b>Résultat</b> : les propriétés de <b>Comportement</b> s'affichent dans la partie centrale inférieure de la fenêtre <b>Programmation</b> .
3	Modifiez ces propriétés.
4	Cliquez sur <b>Appliquer</b> pour enregistrer les modifications.

### Démarrage

Déterminez le comportement du programme après un redémarrage de l'automate logique :

- **Démarrer avec l'état précédent** : l'automate logique démarre avec l'état qu'il avait avant son arrêt.
- **Démarrer en mode d'arrêt** : l'automate logique ne démarre pas automatiquement l'exécution de l'application.
- **Démarrer en mode Run** (par défaut) : l'automate logique démarre automatiquement l'exécution de l'application lorsque certains critères d'exécution, tels que la présence et la charge d'une batterie, sont satisfaits.
- **Démarrage inconditionnel en mode Run** : l'automate logique démarre automatiquement l'exécution de l'application même en cas d'absence ou de décharge de la batterie du contrôleur.

### Chien de garde

Un chien de garde est un temporisateur spécial qui permet aux programmes de ne pas dépasser le temps de scrutation qui leur est alloué.

La valeur par défaut du temporisateur du chien de garde est de 250 ms. Indiquez la durée de la tâche de scrutation du chien de garde. Elle doit être comprise entre 10 et 500 ms.

## Tâches et modes de scrutation

### Présentation

EcoStruxure Machine Expert - Basic propose différents modes de scrutation pour la tâche maître :

- **Mode normal**

Mode de scrutation cyclique continu (Roue libre) ; la nouvelle scrutation débute à l'issue de la précédente.

- **Mode périodique**

Mode de scrutation cyclique périodique ; la nouvelle scrutation ne débute qu'après l'expiration du temps configuré pour la scrutation précédente. Chaque scrutation a donc la même durée.

EcoStruxure Machine Expert - Basic propose les types de tâche suivants :

- **Tâche maître** : tâche principale de l'application.

La tâche maître est commandée par la scrutation cyclique continue (en mode normal) ou en spécifiant une période de scrutation comprise entre 1 et 150 ms (100 ms par défaut) en mode périodique.

- **Tâche périodique** : sous-programme de courte durée, exécuté à intervalles réguliers.

Ce type de tâche est configuré en définissant une période de scrutation comprise entre 1 et 255 ms (255 ms par défaut).

- **Tâche d'événement** : sous-programme de très courte durée, permettant de diminuer le temps de réponse de l'application.

Les tâches d'événement sont déclenchées par les entrées physiques ou les blocs fonction HSC. Ces événements sont associés à des entrées numériques intégrées (%I0.2 à %I0.5) (front montant, front descendant ou les deux) ou aux compteurs rapides (%HSC0 et %HSC1) (lorsque le compteur atteint son seuil). Vous pouvez configurer 2 événements pour chaque bloc fonction HSC (High Speed Counter).

## Priorités des tâches

Ce tableau récapitule les types de tâche et leurs priorités :

Type de tâche	Mode de scrutation	Condition de déclenchement	Plage configurable	Nombre maximum de tâches	Priorité
Maître	Normal	Normal	Non applicable	1	La plus basse
	Périodique	Temporisateur logiciel	1 à 150 ms <sup>1</sup>		
Périodique	Périodique	Temporisateur logiciel	1 à 255 ms <sup>1</sup>	1	Supérieure à celle de la tâche maître et inférieure à celle des tâches d'événement
Evénement	Périodique	Entrées physiques	%I0.2 à %I0.5	4	La plus haute
		Blocs fonction %HSC	2 événements par objet %HSC		
<sup>1</sup> L'application doit être configurée avec un niveau fonctionnel ( <i>voir page 84</i> ) au moins équivalent au Niveau 5.0 pour pouvoir configurer une valeur minimum d'1 ms. Sinon, la valeur minimum est de 2 ms.					

## Sous-chapitre 5.4

### Gestion des POU

#### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
POU	90
Gestion de POU avec des tâches	91
Gestion des réseaux	94
Gestion des POU Grafcet (SFC)	97
POU libres	99

## Présentation

## POU

Un POU (ou unité d'organisation de programme) est un objet réutilisable dans un programme. Chaque POU comprend une déclaration de variable et un jeu d'instructions dans le code source d'un langage de programmation pris en charge.

Il existe toujours un POU lié à la tâche maître du programme. Celui-ci est appelé automatiquement à chaque démarrage du programme.

Vous pouvez créer d'autres POU contenant d'autres objets (fonctions ou blocs fonction, par exemple).

Lors de sa création, un POU peut être au choix :

- associé à une tâche (*voir page 91*) ;
- un POU libre (*voir page 99*). Un Free POU n'est pas associé à une tâche ou un événement. Il peut, par exemple, contenir des fonctions de bibliothèque gérées indépendamment du programme principal. Les POU libres sont appelés par des sous-programmes ou des sauts d'un programme. Une tâche périodique (*voir page 134*) est un sous-programme mis en œuvre en tant que POU libre.

Etape	Action
1	Cliquez sur l'onglet <b>Tâches</b> dans la partie gauche de la fenêtre <b>Programmation</b> .
2	<p>Ajoutez un réseau dans un POU en procédant au choix comme suit :</p> <ul style="list-style-type: none"> <li>● A l'aide du bouton droit de la souris, cliquez sur le POU souhaité, puis sélectionnez <b>Ajouter un réseau</b> dans le menu contextuel qui s'affiche.</li> <li>● Sélectionnez un POU et cliquez sur  (bouton <b>Ajouter un réseau</b>) dans la barre d'outils en haut de l'onglet <b>Tâches</b>.</li> <li>● Sélectionnez un POU et cliquez sur  (bouton <b>Créer un nouveau réseau</b>) dans la barre d'outils en haut de l'espace de travail de programmation.</li> </ul> <p><b>Résultat</b> : un nouveau réseau est ajouté à la structure du programme, juste après le dernier réseau.</p>
	<p>Pour repositionner un réseau dans un POU, sélectionnez-le et cliquez sur le bouton <b>HAUT</b> ou <b>BAISSER</b> de la barre d'outils située en haut de l'onglet <b>Tâches</b>.</p>
	<p>...ent de séquence, tel que Réseau0 (Rung0). Pour faciliter ...</p> <p>... ajouter un commentaire en cliquant sur l'en-tête.</p>
	<p>... contacts). Pour changer le langage ...</p>
	<p>... et sur</p>

## Création d'un POU Grafcet

## Gestion des POU Grafcet (SFC)

Etape	Action
1	Cliquez sur l'onglet <b>Tâches</b> dans la partie gauche de la fenêtre <b>Programmation</b> .
2	<p>Ajoutez un POU Grafcet en procédant au choix comme suit :</p> <ul style="list-style-type: none"> <li>● A l'aide du bouton droit de la souris, cliquez sur l'élément <b>Tâche maître</b>, puis sélectionnez <b>Ajouter un POU Grafcet</b> dans le menu contextuel qui s'affiche.</li> <li>● Cliquez sur le bouton  (<b>Ajouter un POU Grafcet</b>) dans la barre d'outils située en haut de l'onglet <b>Tâches</b>.</li> </ul> <p><b>Résultat</b> : un nœud <math>n</math>- <b>Grafcet</b> s'affiche sous le nœud <b>Tâche maître</b> (<math>n</math> étant un entier incrémenté à chaque création d'un POU Grafcet).</p>

## Insertion d'un POU Grafcet

Etape	Action
1	<p>Cliquez sur l'onglet <b>Tâches</b> dans la partie gauche de la fenêtre <b>Programmation</b>.</p> <p>Placez le curseur de la souris sur le POU Grafcet au-dessus duquel vous souhaitez insérer le nouveau POU Grafcet.</p> <p>Appuyez sur le bouton droit de la souris sur le POU sélectionné, puis sélectionnez <b>Insérer</b></p> <p>Appuyez sur le bouton <b>Insérer</b> et cliquez</p>

## Introduction POU libres

Dans EcoStruxure Machine Expert - Basic, un POU libre est un type spécial de POU non explicitement associée à une tâche :

- [-] POU libres
  - [-] POU\_0 libre (SR2)
    - Réseau 0
    - Réseau 1
  - [-] POU\_1 libre (SR3)
    - Réseau 0
  - [-] POU\_2 libre (SR4)
    - Réseau 0
    - Réseau 1
    - Réseau 2

Chaque POU libre est mis en œuvre sous la forme d'un sous-programme et est constitué d'un ou de plusieurs réseaux écrits en langage de programmation LD ou IL.

NOTE : les POU Grafset ne peuvent pas être de type POU libre.

# Sous-chapitre 5.5

## Fonctions définies par l'utilisateur

### Présentation

Une fonction définie par l'utilisateur permet de créer des fonctions avec des paramètres d'entrée, des variables locales et une valeur de retour. Les fonctions définies par l'utilisateur sont stockées dans le projet EcoStruxure Machine Expert - Basic.

Vous pouvez appeler des fonctions définies par l'utilisateur dans :

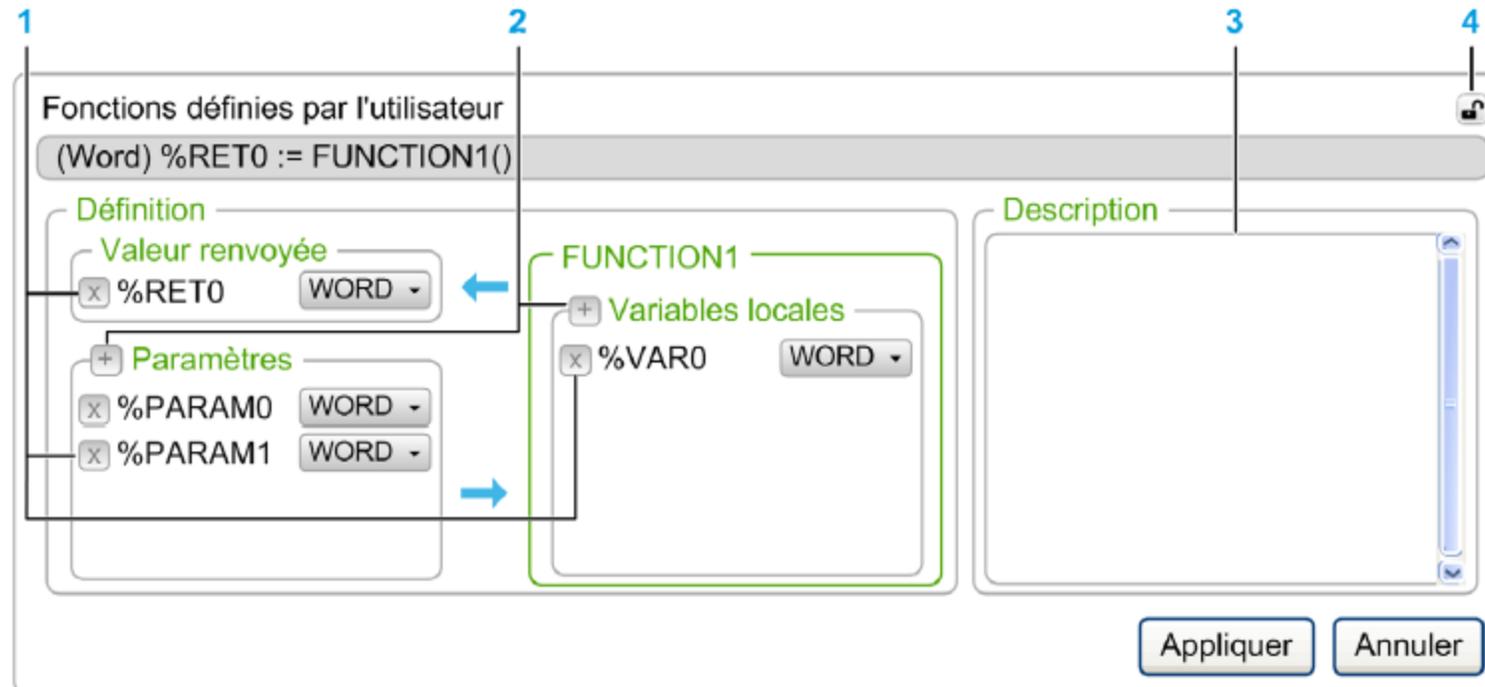
- La tâche maître
- Les tâches périodiques
- Les événements
- Les POU libres

Sujet	Page
Création d'une fonction définie par l'utilisateur	104
Définition d'une fonction définie par l'utilisateur	105
Gestion des fonctions définies par l'utilisateur	110

## Présentation

# Définition d'une fonction définie par l'utilisateur

L'illustration suivante montre les actions disponibles dans la vue **Propriétés** de la fonction définie par l'utilisateur :



- 1 Supprime la **valeur de retour**, un **paramètre d'entrée** ou une **variable locale**
- 2 Ajoute une **valeur de retour**, un **paramètre d'entrée** ou une **variable locale**
- 3 Ecrit éventuellement une description de l'objet de la fonction définie par l'utilisateur. Cette description apparaît dans une bulle d'aide lorsque vous utilisez la fonction définie par l'utilisateur dans un **bloc opération** ou un **bloc comparaison**.
- 4 Détache la vue des propriétés

---

## Sous-chapitre 5.6

### Blocs fonction définis par l'utilisateur

#### Présentation

Un bloc fonction défini par l'utilisateur permet de créer des blocs fonction avec un ou plusieurs paramètres d'entrée et de sortie et des variables locales. Les blocs fonction définis par l'utilisateur sont stockés dans le projet EcoStruxure Machine Expert - Basic.

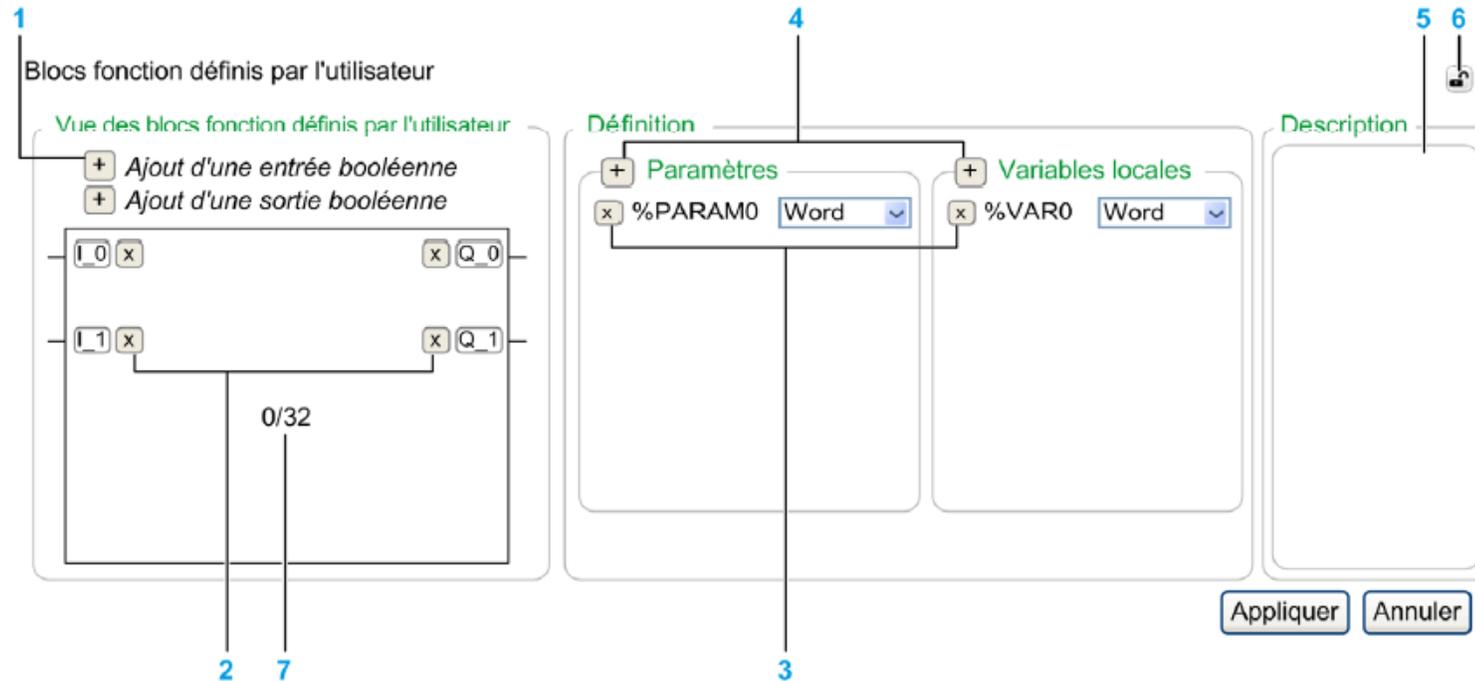
Vous pouvez appeler des blocs fonction définis par l'utilisateur dans :

- La tâche maître
- Les tâches périodiques
- Les événements
- Les POU libres

Sujet	Page
Création d'un bloc fonction défini par l'utilisateur	114
Définition d'un bloc fonction défini par l'utilisateur	115
Gestion des blocs fonction définis par l'utilisateur	119

## Présentation Définition d'un bloc fonction défini par l'utilisateur

L'illustration suivante montre les actions disponibles dans la vue **Propriétés** de la fonction définie par l'utilisateur :



- 1 Ajoute une **entrée** ou une **sortie**
- 2 Supprime une **entrée** ou une **sortie**
- 3 Supprime le **paramètre** ou la **variable locale**
- 4 Ajoute un **paramètre** ou une **variable locale**
- 5 Ecrit éventuellement une description de l'objet du bloc fonction défini par l'utilisateur. Cette description apparaît dans une bulle d'aide lorsque vous utilisez le bloc fonction défini par l'utilisateur dans un **bloc opération**.
- 6 Détache la vue des propriétés
- 7 Nombre d'instances dans la logique utilisateur pour cette définition de bloc fonction

## Sous-chapitre 5.7

### Tâche maître

#### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Description de la tâche maître	123
Configuration de la tâche maître	124

#### Présentation

#### Description de la tâche maître

La tâche maître représente la principale tâche du programme de l'application. Obligatoire, elle est créée par défaut. La tâche maître est constituée de sections et de sous-programmes représentés dans des POU (Program Organizational Unit). Chaque POU de la tâche maître peut être programmé dans l'un des langages de programmation pris en charge.

Pour	Section à consulter
Créer un POU dans la tâche maître	Création d'un POU associé à une tâche ( <i>voir page 91</i> )
Renommer un POU dans la tâche maître	Renommage d'un POU ( <i>voir page 93</i> )
Supprimer un POU de la tâche maître	Suppression d'un POU ( <i>voir page 93</i> )

## Propriétés de la tâche maître

### Mode de scrutation

Choisissez le mode de scrutation à utiliser pour le programme :

- **Normal** : lorsqu'un Logic Controller est en mode de scrutation normal (roue libre), la nouvelle scrutation débute immédiatement à l'issue de la précédente.
- **Périodique** : en mode de scrutation périodique, le Logic Controller attend que la durée de scrutation configurée soit écoulée pour lancer une nouvelle scrutation. Chaque scrutation a donc la même durée.

Spécifiez la **Période** du mode de scrutation périodique (comprise entre 1 et 150 ms).

### Bits et mots système contrôlant la tâche maître

La tâche maître peut être contrôlée par les bits système (%S) et les mots système (%SW) :

Ce tableau répertorie les bits système :

Bits système	Description
%S11	Dépassement du chien de garde
%S19	Dépassement de la période de scrutation (mode de scrutation périodique)

Ce tableau répertorie les mots système :

Mots système	Description
%SW0	Période de scrutation du Logic Controller (mode de scrutation périodique).
%SW30, %SW70	Durée de la dernière scrutation. Indique la durée du dernier cycle de scrutation du contrôleur,
%SW31, %SW71	Durée de scrutation maximum. Indique la durée de la plus longue scrutation du Logic Controller depuis le dernier démarrage à froid de ce dernier.
%SW32, %SW72	Durée de scrutation minimum. Indique la durée de la plus courte scrutation du Logic Controller depuis le dernier redémarrage à froid de ce dernier.

# Sous-chapitre 5.8

## Chaînes

### Présentation

Les chaînes sont une séquence d'octets contenant des caractères ASCII que vous pouvez stocker dans les objets mémoire suivants :

- Mots mémoire `%MW`
- Mots constants `%KW`

Sujet	Page
Configuration de chaînes en mots constants	128
Affectation de chaînes dans des mots mémoire	129
Gestion des chaînes	130

Suite page suivante....

---

## Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

# Sous-chapitre 5.9

## Tâche périodique

Sujet	Page
Création d'une tâche périodique	134
Configuration de la durée de scrutation de la tâche périodique	136

### Présentation **Création d'une tâche périodique**

Une tâche périodique est un sous-programme, généralement de courte durée, exécuté à intervalles réguliers. Dans EcoStruxure Machine Expert - Basic, ce sous-programme est mis en œuvre sous la forme d'un POU libre (*voir page 99*). Il peut être écrit dans l'un des langages de programmation pris en charge par EcoStruxure Machine Expert - Basic.

## Sous-chapitre 5.10

### Tâche d'événement

#### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Présentation des tâches d'événement	138
Sources d'événement	139
Priorités des événements	140
Affichage de tâches d'événement	141

#### Priorités des événements

Les événements peuvent avoir une des huit priorités disponibles : de 7 (la plus faible) à 0 (la plus élevée).

Attribuez une priorité à chaque source d'événements. Deux événements ne peuvent pas avoir la même priorité. L'ordre d'exécution des événements dépend des priorités de chaque événement et de leur ordre de détection.

#### Affectation d'un POU libre à une source d'événement

Pour affecter un POU libre à une source d'événements configurée, procédez comme suit : ....

## Introduction

## Présentation des tâches d'événement

Une tâche d'événement :

- est une portion de programme exécutée à une condition donnée (source d'événement) ;
- possède une priorité supérieure à celle du programme principal ;
- a un temps de réponse rapide, qui permet au système d'être très réactif.

## Description d'un événement

Un événement se compose :

- d'une *source d'événement* : condition logicielle ou matérielle qui interrompt le programme lorsque l'événement est déclenché ;
- d'un *POU* : entité de programme indépendante (sous-programme) associée à un événement ;
- d'un *niveau de priorité* : priorité affectée aux événements pour déterminer leur ordre d'exécution.

## Présentation Sources d'événement

8 sources d'événement sont disponibles :

- 4 sont liées aux entrées physiques sélectionnées du Logic Controller ;
- 4 sont liées aux seuils du bloc fonction HSC

Une source d'événement est toujours associée à un seul événement. Lorsqu'un événement est déclenché, il interrompt le contrôleur, qui exécute ensuite le sous-programme associé.

## Sous-chapitre 5.11

### Utilisation des outils

#### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Messages	145
Tables d'animation	148
Objets mémoire	154
Objets système	159
Objets d'E/S	160
Objets de réseau	161
Objets logiciels	162
Objets PTO	163
Objets de variateur	164
Objets de communication	165
Rechercher et remplacer	166
Référence croisée	168
Liste de symboles	169
Vue de l'utilisation de la mémoire	174

## Tables d'animation

Vous pouvez ajouter des objets dans des tables d'animation pour :

- afficher les symboles et commentaires associés à ces objets ;
- afficher et modifier les valeurs en temps réel de certains types d'objet lorsque EcoStruxure Machine Expert - Basic est connecté au contrôleur logique (mode en ligne) ;
- sélectionner les objets à afficher dans la fenêtre (*voir page 238*) Trace.

## Objets mémoire

Les objets mémoire incluent :

- Bits mémoire
- Mots mémoire
- Mots constants

## Objets d'E/S

Les types d'objet ci-dessous sont propres au matériel indiqué et dépendent du Logic Controller utilisé :

- Entrées et sorties logiques
- Entrées et sorties analogiques
- Blocs fonction avancés, tels que les compteurs rapides et les générateurs d'impulsion.

## Objets de réseau

Les objets de réseau permettent de communiquer via EtherNet/IP, Modbus TCP ou Scrutateur d'E/S Modbus série.

## Objets de communication

Les objets de communication sont utilisés pour communiquer avec les appareils Modbus, envoyer/recevoir des messages en mode caractère (ASCII) et envoyer/recevoir des messages SMS.

## Sous-chapitre 5.12

Contenu de ce sous-chapitre

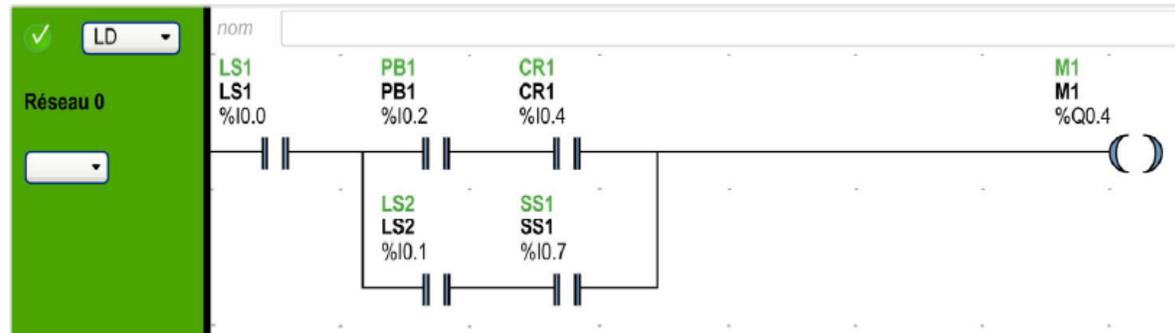
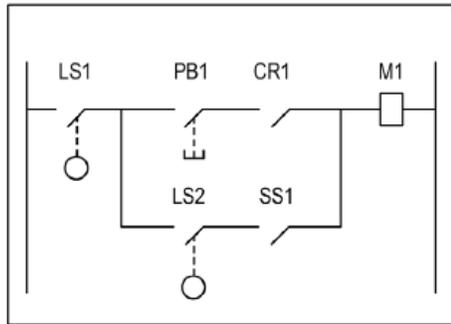
### Programmation en langage Schéma à contacts

Sujet	Page
Introduction aux schémas à contacts	177
Principes de programmation des schémas à contacts	180
Codes couleur des réseaux	182
Éléments graphiques des schémas à contacts	184
Blocs comparaison	191
Blocs opération	192
Ajout de commentaires	196
Bonnes pratiques en matière de programmation	197

## Introduction aux schémas à contacts

Les schémas à contacts utilisent la même représentation graphique que celle des circuits de relais en logique programmée. A ceci près que, dans un schéma à contacts :

- Toutes les entrées et tous les bits logiques binaires sont représentés par des symboles 
- Toutes les sorties et tous les bits logiques binaires sont représentés par des symboles 
- Les opérations numériques proviennent du jeu d'instructions graphiques des schémas à contacts.



## Réseaux de schéma à contacts

Un programme en langage Schéma à contacts est composé de réseaux, c'est-à-dire d'ensembles d'instructions graphiques placées entre deux barres verticales. Les réseaux sont exécutés de manière séquentielle par le Logic Controller.

L'ensemble des instructions graphiques représente les fonctions suivantes :

- Entrées/sorties du contrôleur (boutons-poussoir, capteurs, relais, voyants, etc.)
- Fonctions du contrôleur (temporisateurs, compteurs, etc.)
- Opérations mathématiques et logiques (addition, division, AND, XOR, etc.)
- Opérateurs de comparaison et autres opérations numériques ( $A < B$ ,  $A = B$ , décalage, rotation, etc.)
- Variables internes dans le contrôleur (bits, mots, etc.)

Ces instructions sont disposées graphiquement selon des connexions verticales et horizontales, débouchant éventuellement sur une ou plusieurs sorties et/ou actions. Un réseau ne peut pas contenir plus d'un groupe d'instructions liées.

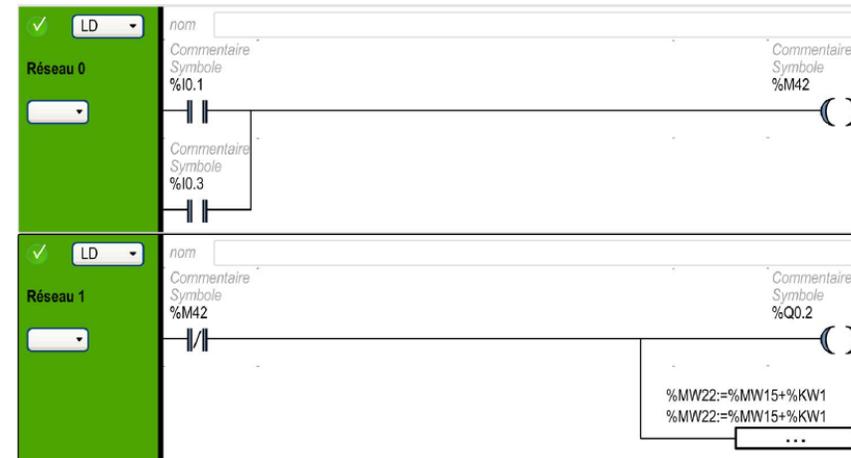
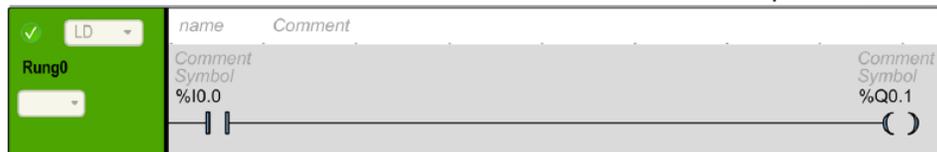
### Mode en ligne

En mode en ligne :

- Les réseaux non modifiés s'affichent sur un fond vert.
- Les réseaux ajoutés ou modifiés en mode en ligne s'affichent sur un fond orange :



- Les réseaux sans éléments modifiables sont verrouillés. Leur avant-plan est gris :



## Bonnes pratiques en matière de programmation

### Gestion des sauts de programme

Utilisez les sauts de programme avec précaution, car ils peuvent générer des boucles qui ralentissent considérablement les opérations de scrutation. Evitez d'insérer des sauts pointant vers des instructions situées en amont.

### Programmation des sorties

Les sorties physiques et les bits logiques ne doivent être modifiés qu'une seule fois dans le programme. Pour les sorties physiques, seule la dernière valeur scrutée est prise en compte lors de leur mise à jour.

### Utilisation de capteurs d'arrêt d'urgence à liaison directe

Les capteurs utilisés directement en cas d'arrêt d'urgence ne doivent pas être gérés par le Logic Controller. Ils doivent être connectés directement aux sorties correspondantes et mise en œuvre conformément aux réglementations locales, nationales et/ou internationales.

### Gestion des reprises secteur

Après une coupure de courant, la reprise secteur doit se faire par une opération manuelle. Un redémarrage automatique de l'installation peut entraîner un fonctionnement inattendu de la part de l'équipement (utilisez les bits système %S0, %S1 et %S49). D'autres bits et mots système peuvent également faciliter la gestion des reprises après des coupures de courant. Reportez-vous aux sections Bits système (%S) et Mots système (%SW) (*voir Modicon M221, Contrôleur logique, Guide de programmation*).

### Gestion de l'heure et des blocs horodateurs

Il est nécessaire de vérifier l'état du bit système %S51 qui signale toute erreur détectée de l'horodateur.

## Sous-chapitre 5.13

### Programmation en langage Liste d'instructions

#### Contenu de ce sous-chapitre

Sujet	Page
Présentation des programmes en langage Liste d'instructions	202
Fonctionnement des instructions en langage Liste d'instructions	205
Instructions en langage Liste d'instructions	206
Utilisation des parenthèses	210

#### Présentation des programmes en langage Liste d'instructions

Un programme en langage Liste d'instructions consiste en une série d'instructions exécutées de manière séquentielle par le Logic Controller. Chaque instruction est représentée par une ligne de code et se compose des éléments suivants :

- Numéro de la ligne
- Valeur courante (uniquement en mode en ligne)
- Opérateur d'instruction
- Opérande(s)
- Commentaire (facultatif)

## Exemple d'un programme en langage Liste d'instructions

IL		nom		
0000	LD	%M1		Charger le bit 1
0001	AND (	%I0.1		Créer une branche et charger le bit d'entrée 1
0002	OR (	%I0.2		Charger le bit d'entrée 2
0003	ANDN	%I0.3		Charger le bit d'entrée 3 et inverser
0004	)			Commentaire
0005	)			Commentaire
0006	ST	%Q0.0		Régler le bit de sortie 0

Avec les opérateurs logiques `AND` et `OR`, les parenthèses permettent d'imbriquer les instructions logiques. Ce faisant, elles créent des branches dans l'éditeur de schéma à contacts. Les parenthèses sont associées à des instructions, de la manière suivante :

- L'ouverture de parenthèses est associée à l'opérateur `AND` ou `OR`.
- La fermeture de parenthèses est une instruction (opérateur sans opérande) requise pour chaque parenthèse ouverte.

## Sous-chapitre 5.14

### Programmation en Grafcet (liste)

#### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Description de la programmation en Grafcet (liste)	214
Structure du programme Grafcet (liste)	216
Utilisation des instructions Grafcet (liste) dans un programme EcoStruxure Machine Expert - Basic	220

## Sous-chapitre 5.15

### Programmation en Grafcet (SFC)

#### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Introduction à la programmation en Grafcet (SFC)	223
Utilisation de l'Editeur graphique Grafcet (SFC)	226
Divergences	231
Bonnes pratiques relatives à la programmation	235

## Introduction à la programmation en Grafcet (SFC)

Le langage Grafcet (SFC) est un langage de programmation graphique qui décrit l'ordre chronologique d'exécution de tâches numériques, appelées *étapes*. L'ordre d'exécution des étapes dépend des *transitions* qui les relient.

### Composition d'un POU Grafcet (SFC)

Un POU Grafcet (SFC) est constitué des éléments suivants :

- **Etape** : exécute une série d'actions définies dans un ou plusieurs réseaux écrits en langage de programmation LD ou IL. On distingue plusieurs types d'étapes :
  - **Etape initiale** : exécutée au début du programme ou suite au redémarrage du contrôleur. Cette étape est représentée par une cellule à double bordure.
  - **Etape normale** : exécutée sous condition une fois l'étape initiale totalement achevée.
- **Transition** : expression booléenne évaluée entre les étapes. La transition permet de faire la liaison entre deux étapes ou plus. L'expression booléenne est définie dans un seul réseau de transitions écrit en langage de programmation LD ou IL.

### Règles applicables aux POU Grafcet (SFC)

Les POU Grafcet peuvent être créés uniquement dans la tâche maître d'un programme. Il est possible de créer plusieurs POU Grafcet.

## Traitement des Grafcet (SFC)

Le Logic Controller applique les règles d'exécution suivantes aux Grafcet (SFC) :

- Le cycle de la tâche maître démarre.
- Les POU précédant la première étape Grafcet (SFC) sont exécutés de façon séquentielle.
- La première étape Grafcet (SFC) lance le **moniteur de Grafcet**.
- Une fois le **moniteur de Grafcet** terminé, le POU qui suit immédiatement la dernière étape Grafcet (SFC) est appelé.

## Sous-chapitre 5.16

### Débogage en mode en ligne

#### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Fenêtre Trace	238
Modification de valeurs	241
Forçage de valeurs	242
Modifications en mode en ligne	243

## Chapitre 6

### Mise en service

#### Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
6.1	Présentation de la fenêtre Mise en service	252
6.2	Connexion à un contrôleur logique	253
6.3	Mise à jour du contrôleur	266
6.4	Gestion de la mémoire	267
6.5	Info contrôleur	273
6.6	Gestion RTC	275

---

## Sous-chapitre 6.2

### Connexion à un contrôleur logique

#### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Connexion à un Logic Controller	254
Chargement et téléchargement d'applications	261

## Chapitre 7

### Simulateur

#### Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Présentation du simulateur EcoStruxure Machine Expert - Basic	278
Fenêtre du gestionnaire d'E/S du simulateur EcoStruxure Machine Expert - Basic	280
Fenêtre <b>Gestion du temps</b> du Simulateur EcoStruxure Machine Expert - Basic	282
Modification de valeurs à l'aide du simulateur EcoStruxure Machine Expert - Basic	285
Utilisation du simulateur EcoStruxure Machine Expert - Basic	291
Lancement de la simulation dans Vijeo-Designer	292

#### Présentation du simulateur EcoStruxure Machine Expert - Basic

Le simulateur EcoStruxure Machine Expert - Basic vous permet :

- de simuler une connexion entre le PC, le contrôleur logique et les éventuels modules d'extension ;
- d'exécuter et de tester un programme sans contrôleur logique ni module d'extension, via une connexion physique au PC.

Le simulateur réplique le comportement du contrôleur logique et joue le rôle d'un contrôleur logique virtuel auquel vous vous connectez à l'aide de EcoStruxure Machine Expert - Basic.

## Fenêtres du simulateur EcoStruxure Machine Expert - Basic

Le simulateur EcoStruxure Machine Expert - Basic comporte les deux fenêtres suivantes :

- **Fenêtre de gestion du temps du simulateur**  
Vous permet de contrôler l'horodateur du contrôleur pour simuler le passage du temps et son impact sur les constructions logiques affectées par l'horodateur.
- **Fenêtre du gestionnaire d'E/S du simulateur**  
Permet de gérer l'état des E/S du contrôleur et des modules d'extension.

Une fois la connexion établie entre le PC et le contrôleur logique virtuel (consultez la section Utilisation du simulateur EcoStruxure Machine Expert - Basic (*voir page 291*)), la fenêtre du gestionnaire d'E/S du simulateur EcoStruxure Machine Expert - Basic s'affiche :

## Fenêtre du gestionnaire d'E/S du simulateur EcoStruxure Machine Expert - Basic

La fenêtre du gestionnaire d'E/S du simulateur comprend les éléments suivants :

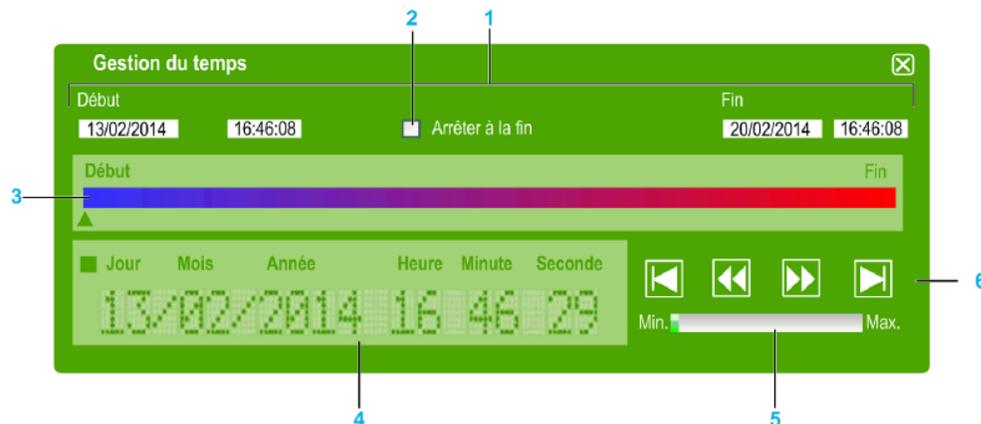
- **Etat des voyants :**  
Pour surveiller l'état des voyants d'un contrôleur simulé.
- **Etat des E/S :**  
Pour contrôler les entrées et sorties lors de l'exécution du programme.

## Fenêtre Gestion du temps du Simulateur EcoStruxure Machine Expert - Basic

La fenêtre **Gestion du temps** du Simulateur contient les éléments suivants :

- Plage de simulation de date/heure pour l'exécution du programme dans le simulateur :
  - Date et heure de **Début**
  - Date et heure de **Fin**
  - Case à cocher **Arrêter à la fin** (pour arrêter l'exécution du programme lorsque la date et l'heure de **Fin** sont atteintes)
- Barre de contrôle du temps :
  - Pour déplacer la simulation de l'écoulement du temps vers l'avant ou vers l'arrière
- Affichage de la date et de l'heure :
  - Date et heure de l'horodateur simulé du simulateur
- Boutons de commande :
  - Pour réinitialiser, reculer, avancer ou arrêter la gestion du temps associée à l'horodateur
- Barre d'incrémentatation :
  - Pour contrôler la vitesse d'écoulement simulé du temps par rapport au temps réel

La fenêtre **Gestion du temps** du simulateur est représentée ci-dessous :



- 1 Plage de simulation de date/heure (Début – Fin)
- 2 Case à cocher Arrêter à la fin (de la plage de date/heure)
- 3 Barre de contrôle du temps
- 4 Date et heure de l'horodateur
- 5 Barre d'incrémentatation
- 6 Boutons de commande du temps écoulé

Analyste Programmeur en Automatisation, Robotique et  
Informatique Industrielle  
TS ARII

***Module MF 3.1***

***MF 3.1 Régler et mettre au point un S.A.P.***

***Programmer avec Ecostruxure***

**Fin de Présentation**